SAĞLAM KISMİ EN KÜÇÜK KARELER REGRESYON ANALİZİNDE YENİ YAKLAŞIMLAR

NEW APPROACHES IN ROBUST PARTIAL LEAST SQUARES REGRESSION ANALYSIS

ESRA POLAT

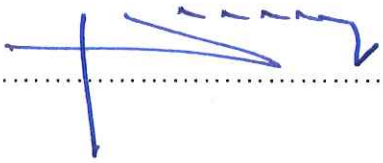
PROF. DR. SÜLEYMAN GÜNAY
Tez Danışmanı

Hacettepe Üniversitesi
Lisansüstü Eğitim-Öğretim ve Sınav Yönetmeliğinin
İstatistik Anabilim Dalı için Öngördüğü
DOKTORA TEZİ olarak hazırlanmıştır.

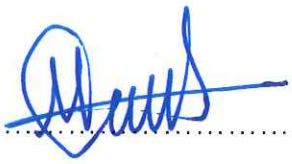
2014

ESRA POLAT'ın hazırladığı "**Sağlam Kısmi En Küçük Kareler Regresyon Analizinde Yeni Yaklaşımlar**" adlı bu çalışma aşağıdaki jüri tarafından **İSTATİSTİK ANABİLİM DALI**'nda **DOKTORA TEZİ** olarak kabul edilmiştir.

Prof. Dr. Hamza Gamgam .....
Başkan

Prof. Dr. Süleyman GÜNAY ..... 
Danışman

Prof. Dr. Gül ERGÜN .....
Üye

Doç. Dr. Meral ÇETİN ..... 
Üye

Doç. Dr. Haydar Demirhan .....
Üye

Bu tez Hacettepe Üniversitesi Fen Bilimleri Enstitüsü tarafından **DOKTORA TEZİ** olarak onaylanmıştır.

Prof. Dr. Fatma SEVİN DÜZ
Fen Bilimleri Enstitüsü Müdürü

ETİK

Hacettepe Üniversitesi Fen Bilimleri Enstitüsü, tez yazım kurallarına uygun olarak hazırladığım bu tez çalışmasında,

- tez içindeki bütün bilgi ve belgeleri akademik kurallar çerçevesinde elde ettiğimi,
- görsel, işitsel ve yazılı tüm bilgi ve sonuçları bilimsel ahlak kurallarına uygun olarak sunduğumu,
- başkalarının eserlerinden yararlanılması durumunda ilgili eserlere bilimsel normlara uygun olarak atıfta bulunduğumu,
- atıfta bulunduğum eserlerin tümünü kaynak olarak gösterdiğimi,
- kullanılan verilerde herhangi bir tahrifat yapmadığımı,
- ve bu tezin herhangi bir bölümünü bu üniversitede veya başka bir üniversitede başka bir tez çalışması olarak sunmadığımı

beyan ederim.

03/03/2014

Esra POLAT

ÖZET

SAĞLAM KISMİ EN KÜÇÜK KARELER REGRESYON ANALİZİNDE YENİ YAKLAŞIMLAR

Esra POLAT

Doktora, İstatistik Bölümü

Tez Danışmanı: Prof. Dr. Süleyman GÜNAY

Mart 2014, 136 Sayfa

Kısmi En Küçük Kareler Regresyon (Partial Least Squares Regression/PLSR), bir gizli değişkenler (Latent Variables/LV) regresyon yöntemidir. Bu yöntemde, ilkin ilişkisiz LV'lerin bir kümesi kestirilir ve daha sonra, bu LV'lerin bağımlı değişken ile olan ilişkisi modellenir. PLSR modelini elde etmek için literatürde en sık kullanılan algoritmalar, NIPALS ve SIMPLS algoritmalarıdır. NIPALS algoritması tek bir bağımlı değişken olduğunda PLS1 ve çoklu Y değişkeni için kullanıldığında ise PLS2 adını alır. Ancak, NIPALS algoritmasında yüklerin, bileşenlerin ve regresyon katsayılarının hesaplanmasında klasik En Küçük Kareler (Least Squares/LS) regresyon adımları kullandığından ve SIMPLS algoritması bağımlı ile bağımsız değişkenler arasındaki varyans-kovaryans matrisine ve LS regresyonuna dayalı olduğundan, veri kümesinde aykırı değerler olduğunda her iki algoritma ile elde edilen sonuçlar da etkilenir. Bu nedenle, literatürde PLS1, PLS2 ve SIMPLS algoritmalarının bazı sağlamlaştırılmış biçimleri olan sağlam PLSR yöntemleri önerilmiştir. Doktora tez çalışmamızda, modelde tek bir bağımlı değişken olması durumunda PLSR modelindeki regresyon katsayılarını kestirmek için kullanılan klasik PLS1 algoritmasının, seçenek tanımındaki kovaryans matrisini sağlam bir şekilde kestirerek literatürde önerilen sağlam PLSR yöntemlerinden yola çıkılarak,

üç yeni sağlam PLSR yöntemi önerilmiştir. *PLS-ARWMCD*, *PLS-Smult* ve *PLS-MMmult* olarak adlandırılan bu üç yeni sağlam PLSR yöntemi, PLS1 algoritmasındaki kovaryans matrisini sırasıyla, ‘ *ilk adımda konum ve kovaryansın sağlam başlangıç kestiricileri olarak En Küçük Kovaryans Determinantı kestiricilerini kullanan, uyarlanabilir yeniden ağırlıklandırılmış bir kovaryans kestiricisi* ‘, ‘ *S-kestiricileri* ’ ve ‘ MM-kestiricileri ’ kullanılarak sağlam bir şekilde kestirerek bulunmuştur. Bu üç yeni sağlam PLSR yöntemi, literatürde var olan klasik PLSR yöntemi ve sağlam PLSR yöntemleri RSIMPLS, Kısmi Sağlam M-Regresyon (PRM), Stahel-Donoho kestiricisine dayalı sağlam PLSR yöntemi (PLS-SD), yansımaların basıklık katsayısına ve Stahel-Donoho kestiricisine dayalı sağlam PLSR yöntemi (PLS-KurSD) ile etkinlik, veriye uyum ve kestirimdeki başarıları açısından benzetim çalışmaları ve gerçek bir veri kümesi üzerinde uygulama ile karşılaştırılmıştır. Yeni önerilen üç sağlam PLSR yöntemi de, küçük boyutlu ve makul düzeyde aykırı değerler tarafından bozulan veri kümelerine uygulandığında özellikle klasik PLSR yönteminden ve bazı aykırı değer türlerinde sağlam PRM, PLS-SD ve PLS-KurSD yöntemlerinden daha sağlam ve etkin sonuçlar verir.

Anahtar Kelimeler: Sağlam Kısmi En Küçük Kareler Regresyonu, RSIMPLS, PLS-SD, PLS-KurSD, PRM, En Küçük Kovaryans Determinantı, sağlam ve etkin uyarlanabilir yeniden ağırlıklandırılmış kovaryans kestiricisi, S-kestiricileri, MM-kestiricileri, uyum iyiliği, kestirim, etkinlik, sağlamlık.

ABSTRACT

NEW APPROACHES IN ROBUST PARTIAL LEAST SQUARES REGRESSION ANALYSIS

Esra POLAT

Doctor of Philosophy, Department of Statistics

Supervisor: Prof. Dr. Süleyman GÜNAY

March 2014, 136 pages

Partial Least Squares Regression (PLSR) method is a Latent Variables (LVs) regression method. Therefore, in this method, firstly, a set of unrelated LVs is predicted and then, the relationship of these LVs with dependent variable is modeled. The most popular algorithms used in literature for obtaining PLSR model are NIPALS and SIMPLS algorithms. NIPALS algorithm called PLS1, when it is used for one dependent variable and called PLS2, when it is used for multiple Y variables. However, classic Least Squares (LS) steps are used in NIPALS algorithm for obtaining loadings, components and regression coefficients and SIMPLS algorithm depends on the covariance matrix between independent and dependent variables and LS regression. Therefore, if there are outliers in the data set, the results obtained with both of the these two algorithms are affected. Hence, the robust PLSR methods, which are the robust versions of PLS1, PLS2 and SIMPLS algorithms, suggested in literature. In our doctoral thesis study, three new robust PLSR methods were suggested based on robust PLSR methods existing in literature that robustly predicts the covariance matrix in the alternative definition of classical PLS1 algorithm used for predicting regression coefficients in PLSR model in case of being one dependent variable in the model. These three new PLSR

methods, called as *PLS-ARWMCD*, *PLS-Smult* and *PLS-MMmult,* found by robustly predicting the covariance matrix in the PLS1 algorithm by using ‘ *an adaptive reweighted estimator of covariance using Minimum Covariance Determinant estimators in the first step as robust initial estimators of location and covariance* ‘, ‘ *S-estimators* ‘ and ‘ *MM-estimators* ‘, respectively. These three new suggested PLSR methods were compared in terms of efficiency, model fit and success in predictions, with classic PLSR method and robust PLSR methods RSIMPLS, Partial Robust M-Regression (PRM), the robust PLSR method based on Stahel-Donoho estimator (PLS-SD) and the robust PLSR method based on kurtosis coefficient of projections and Stahel-Donoho estimator (PLS-KurSD) existing in literature, by using simulation studies and applications on a real data. Three of the new suggested robust PLSR methods, when applied on data sets in low dimensions and contaminated by reasonable level of outliers, give more robust and efficient results than especially classical PLSR method and in some types of outliers than robust PRM, PLS-SD and PLS-KurSD methods.

Keywords: Robust Partial Least Squares Regression, RSIMPLS, PLS-SD, PLS-KurSD, PRM, Minimum Covariance Determinant, robust and efficient adaptive reweighted estimator of covariance, S-estimators, MM-estimators, goodness of fit, prediction, efficiency, robustness.

TEŞEKKÜR

Tez çalışmamın her aşamasında bilgi ve manevi desteği ile her zaman yanımda olan, değerli katkı ve eleştirileri ile bana yol gösteren, emeğini ve zamanını esirgemeyen değerli danışmanım Sayın Prof. Dr. Süleyman GÜNAY'a en içten dileklerimle teşekkür ederim.

Tezin izlenmesi ve değerlendirilmesi aşamalarında değerli yorumları ile bana katkıda bulunan Sayın Prof. Dr. Gül ERGÜN'e, Sayın Prof. Dr. Hamza GAMGAM'a, Sayın Doç. Dr. Meral ÇETİN'e ve Sayın Doç. Dr. Haydar DEMİRHAN'a teşekkür ederim. Bilimsel eğitimime ve doktora tez çalışmama verdiği burs ile destek olan TÜBİTAK'a teşekkür ederim.

Tez çalışmam süresi boyunca manevi desteklerini esirgemedikleri için değerli arkadaşlarım Doç. Dr. Gamze ÖZEL'e, Dr. Nursel KOYUNCU'ya, Dr. Nilgün ÖZGÜL'e ve diğer bütün çalışma arkadaşlarıma içtenlikle teşekkür ederim.

Son olarak çalışmam boyunca gösterdiği anlayış, sabır ve yardımlarından dolayı değerli arkadaşım Dr. Semra TÜRKAN'a ve her zaman gösterdikleri sevgi, anlayış ve güvenle yanımda olarak bana destek olan başta CANIM ANNEM olmak üzere, AİLEM'e içtenlikle teşekkür ederim.

İÇİNDEKİLER

Sayfa

ÖZET……..…..……i
ABSTRACT……....iii
TEŞEKKÜR……..v
İÇİNDEKİLER……..vi
ÇİZELGELER……..ix
SİMGELER VE KISALTMALAR……...x
1.GİRİŞ……1
2. KISMI EN KÜÇÜK KARELER REGRESYONU……9
2.1. Kısmi En Küçük Kareler Regresyon Yöntemi……9
2.2. Doğrusal Olmayan Yinelemeli En Küçük Kareler (NIPALS) Algoritması…….11
2.2.1. Tek Bir Bağımlı Değişken için Dikleştirilmiş PLSR Algoritması……..13
2.2.2. Klasik PLS1 Algoritmasının Seçenek Tanımı……14
2.3. PLS Yönteminin İstatistiksel Olarak Esinlenilmiş Değişikliğinin Basit Bir Uygulaması (SIMPLS) Algoritması……17
3. SAĞLAM KISMİ EN KÜÇÜK KARELER REGRESYONU……21
3.1. Sağlam Kestiricilerin Sahip Olması Gereken Özellikler ve Önemli Sağlamlık Ölçütleri……..21
3.1.1. Kırılma Noktası……...21
3.1.2. Etki Fonksiyonu……..22
3.1.3. İstatistiksel Etkinlik……23
3.2. Sağlam Kısmi En Küçük Kareler Regresyon Yöntemleri……24
3.3. PLS1 ve PLS2 Algoritmalarından Elde Edilen Sağlam PLSR Yöntemleri…...25
3.3.1. Yinelemeli Olarak Yeniden Ağırlıklandırılmış En Küçük Kareler (IRLS) Yöntemi……..25
3.3.2. İç Yinelemeli Yeniden Ağırlıklandırma PLSR Algoritmaları……26
3.3.3. Dış Yeniden Ağırlıklandırma PLSR Algoritması ……..27
3.3.4. Kısmi En Küçük Mutlak Sapmalar (PLAD) Yöntemi……29
3.4. Klasik PLS1 Algoritmasının Seçenek Tanımındaki Kovaryans Matrisini Sağlam Bir Yöntem ile Kestirerek Önerilen Sağlam PLSR Yöntemleri……30
3.4.1. Stahel-Donoho Kestiricisine Dayalı Sağlam PLSR Yöntemi (PLS-SD)……31

3.4.2. BACON Algoritmasına Dayalı Sağlam Kısmi En Küçük Kareler Regresyon Yöntemi……34
3.4.3. Yansımaların Basıklık Katsayısına ve Stahel-Donoho Kestiricisine Dayalı Sağlam PLSR Yöntemi (PLS-KurSD)…… ……35
3.5. SIMPLS Algoritmasının Sağlamlaştırılmasıyla Elde Edilen Sağlam PLSR Yöntemleri……..40
3.5.1. MCD Regresyonunu Kullanılarak Elde Edilen RSIMCD Algoritması……..…42
3.5.2. ROBPCA Regresyonunu Kullanılarak Elde Edilen RSIMPLS Algoritması...44
3.5.3. Kısmi Sağlam M-Regresyon……46
3.5.4. Mekansal İşaret Dönüşümü ile PLSR Yöntemini Sağlamlaştırmak……...55
4. SAĞLAM KISMİ EN KÜÇÜK KARELER REGRESYON ANALİZİNDE YENİ YAKLAŞIMLAR……58
4.1. Önerilen Sağlam PLSR Yöntemi PLS-ARWMCD……58
4.1.1. FAST-MCD Algoritmasında Kullanılan C-adımı ve Dayandığı Teorem……60
4.1.2. H_1 Başlangıç Alt Kümelerinin Oluşturulması……62
4.1.3. Seçmeli Yineleme……..62
4.1.4. İç içe Eklemeler……..63
4.1.5. FAST-MCD Algoritması ve İşleyişi……..64
4.1.6. Sağlam ve Etkin Uyarlanabilir Yeniden Ağırlıklandırılmış Bir Kovaryans Kestiricisi……69
4.2. Önerilen Sağlam PLSR Yöntemleri PLS-Smult ve PLS-MMmult……..73
4.2.1. Çok Değişkenli Konum ve Kovaryans için S-Kestiricileri ile MM kestiricileri……...74
4.2.2. Çok Değişkenli Konum ve Kovaryans için S-kestiricilerini Hesaplayan FastS Algoritması……..……..……76
4.2.3. Çok Değişkenli Konum ve Kovaryans için MM-kestiricilerini Hesaplayan FastMM Algoritması…….....79
5. UYGULAMA……..82
5.1. Benzetim Çalışması……..82
5.1.1. Birinci Benzetim Düzeni……84
5.1.2. İkinci Benzetim Düzeni……..…...93
5.1.3. Aykırı Değerler Tarafından Bozulmayan Temiz Modeldeki Hata Terimlerinin Farklı Dağılımlardan Geldiği Benzetim Düzeni…….……..103

5.1.4. Yeni Önerilen Sağlam PLSR Yöntemlerinin Hesaplama Zamanı ……...108

5.2. Gerçek Veri Kümesi Üzerinde Uygulama……112

6. SONUÇ VE TARTIŞMA……118

KAYNAKLAR……..……124

EKLER……..128

ÖZGEÇMİŞ……..137

ÇİZELGELER

Sayfa

Çizelge 3.1. Adım 1 ve Adım 2 için tek değişkenli yansımalar için kesim değerleri……..39
Çizelge 5.1. n=100 ve p=6, temiz veri kümesi, m=1000 tekrar için benzetim sonuçları…….87
Çizelge 5.2. n=100 ve p=6, gözlemlerin ilk %10'u kötü kaldıraç gözlemleri, m=1000 tekrar için benzetim sonuçları ……88
Çizelge 5.3. n=100 ve p=6, gözlemlerin ilk %10'u dikey aykırı değerler, m=1000 tekrar için benzetim sonuçları…….90
Çizelge 5.4. n=200, p=5 ve k=2, aykırı değer oranı % 10, m=1000 tekrar için benzetim sonuçları……97
Çizelge 5.5. n=200, p=5 ve k=2, aykırı değer oranı % 20, m=1000 tekrar için benzetim sonuçları……99
Çizelge 5.6. Gözlem sayısının (n) ve bağımsız değişken sayısının (p) farklı seçildiği üç örneklem şeması için farklı hata dağılımlarında m=1000 tekrar yapılarak benzetim ile elde edilmiş MSE'ler ……..…105
Çizelge 5.7. Çalışma kümesi 40 ve test kümesi 5 gözlemli balık verisi için GOF ve RMSE değerleri……..114

SİMGELER VE KISALTMALAR

Kısaltmalar

ARWMCD	Uyarlanabilir Yeniden Ağırlıklandırılmış En Küçük Kovaryans Determinantı (Adaptive Reweighted Minimum Covariance Determinant)
BACON	Parçalı Uyarlanabilir Hesaplama Yönünden Etkin Aykırı Gözlem Belirleyicisi (Blocked Adaptive Computationally Efficient Outlier Nominators)
BDP	Kırılma Noktası (Breakdown Point)
CV	Çapraz geçerlik (Cross-validation)
GOF	Uyum iyiliği (Goodness-of-fit)
HBDP	Yüksek Kırılma Noktası (High Breakdown Point)
IF	Etki Fonksiyonu (Influence Function)
IRLS	Yinelemeli Olarak Yeniden Ağırlıklandırılmış En Küçük Kareler (Iteratively Reweighted Least Squares)
LAD	En Küçük Mutlak Sapmalar (Least Absolute Deviations)
LMS	En Küçük Ortanca Kareler (Least Median Squares)
LOOCV	Birini-dışarıda-bırakma çapraz geçerlik (Leave-one-out cross-validation)
LS	En Küçük Kareler (Least Squares)
LV	Gizli Değişken (Latent Variable)
MAD	Ortanca Mutlak Sapma (Median Absolute Deviation)
MCD	En Küçük Kovaryans Determinantı (Minimum Covariance Determinant)
MLR	Çoklu Doğrusal Regresyon (Multiple Linear Regression)
MSE	Hata Kareler Ortalaması (Mean Square Error)
MVE	En Küçük Hacimli Elipsoid (Minimum Volume Ellipsoid)
NIPALS	Doğrusal Olmayan Yinelemeli Kısmi En Küçük Kareler (Non-linear Iterative Partial Least Squares)
NIR	Kızıl Ötesi Yansıyan Spektrumu (Near Infrared Reflectance)
PC	Temel Bileşen (Principal Component)
PCA	Temel Bileşenler Analizi (Principal Component Analysis)

PCR	Temel Bileşenler Regresyonu (Principal Component Regression)
PLAD	Kısmi En Küçük Mutlak Sapmalar (Partial Least Absolute Deviations)
PLS	Kısmi En Küçük Kareler (Partial Least Squares)
PLSR	Kısmi En Küçük Kareler Regresyonu (Partial Least Squares Regression)
PM	Kısmi M-kestiricisi (Partial M-estimator)
PRESS	Kestirim Hata Kareler Toplamı (Prediction Error Sum of Squares)
PRM	Kısmi Sağlam M (Partial Robust M)
RM	Tekrarlı Ortanca (Repeated Median)
RMCD	Yeniden Ağırlıklandırılmış En Küçük Kovaryans Determinantı (Reweighted Minimum Covariance Determinant)
RMSE	Hata Kareler Ortalamasının Karekökü (Root Mean Squared Error)
SDE	Stahel-Donoho Kestiricisi (Stahel-Donoho Estimator)
SIMPLS	PLS yönteminin İstatistiksel olarak Esinlenilmiş Değişikliğinin Basit bir Uygulaması (Straightforward Implementation of a Statistically Inspired Modification of the PLS method)
SPD	Simetrik Pozitif Tanımlı (Symmetric Positive Definite)
SS-PP	Mekansal İşaret Ön İşleme (Spatial Sign Preprocessing)
SVD	Tekil Değer Ayrışımı (Singular Value Decomposition)

1. GİRİŞ

Aykırı gözlemler, veri kümesinin çoğunluğunun sahip olduğu dağılımdan farklı bir dağılıma ya da aynı dağılıma ancak farklı parametrelere sahip oldukları düşünülen ve veri kümesinin çoğunluğundan uzakta bulunan gözlemlerdir [39]. Bir regresyon modelinde üç tür aykırı değer vardır: Birincisi, bağımsız değişkenlere ilişkin **X** matrisinde gözlemlerin çoğunun bulunduğu ana bölümden uzakta bulunan, ancak o gözlemler ile aynı regresyon modeline sahip 'iyi kaldıraç' gözlemleridir. İkincisi, **X** matrisinin ana kısmından uzakta bulunan ve regresyon modelinden önemli bir şekilde sapan 'kötü kaldıraç' gözlemleridir. Üçüncüsü ise kaldıraç gözlemleri olmayan, ancak regresyonda büyük kestirilen **y** artıklarına sahip olan, bu nedenle de 'dikey aykırı' değerler olarak adlandırılanlardır [23].

Genellikle, regresyon parametre kestirimlerinde aykırı gözlemlerin etkisinin yüksek oranda olduğu bilinmektedir. Sağlam yöntemlerde amaç, aykırı değerlerin etkisini azaltmak ya da gidermek ve geriye kalan gözlemlerin büyük bir çoğunlukla sonuçları belirlemesine olanak sağlamaktır. Bir kestirici, veri kümesinde bulunan aykırı gözlemlerin varlığından etkilenmiyor ise o kestirici sağlam, etkileniyor ise sağlam olmayan kestiricidir. Daha genel bir ifade ile veri kümesinde küçük değişimler ya da genel olarak varsayımlarda küçük sapmalar olduğunda bile iyi performans gösteren kestiriciler, 'sağlam kestiriciler' olarak adlandırılır. Benzer bir yaklaşım ile kestirim yöntemi de, sağlam ve sağlam olmayan yöntem şeklinde isimlendirilir [23, 36, 39]. Sağlam kestiriciler ilk olarak 1960 yılında, alanının öncülerinden olan Tukey tarafından tanıtılmıştır. Daha sonra ilerleyen yıllarda sağlam regresyon çözümlemesiyle ilgili yapılan birçok çalışma sonucunda bulunan en dikkat çekici sonuç, hatalar normal bir dağılıma sahip olmadığında ya da veri kümesinde aykırı değerler olduğunda, Klasik En Küçük Kareler (Least Squares/LS) kestiricisinin sağlam olmayan kestirimler vermesidir. Bu nedenle, veri kümesi normallikten sapmalar gösterdiğinde ya da aykırı değerler içerdiğinde, sağlam kestirimler verecek kestiriciler türetilmeye çalışılmıştır [18, 36].

Kısmi En Küçük Kareler Regresyonu (Partial Least Squares Regression/PLSR), bir ya da daha fazla bağımlı değişkeni çok sayıda bağımsız değişken ile ilişkilendirmek için kullanılan bir yöntemdir. PLSR, bağımlı ve bağımsız değişkenler arasındaki doğrusal ilişkiyi dikkate alan bir veri indirgeme yöntemidir. PLSR yöntemi ile bağımlı değişkenleri kesin bir anlamda açıklayacak bileşenler seçilir. PLSR, bir gizli değişkenler (Latent Variables/LV) regresyon yöntemidir. Bu yöntemde, ilkin ilişkisiz LV'lerin bir kümesi kestirilir ve daha sonra, bu LV'lerin bağımlı değişken ile olan ilişkisi modellenir. PLSR yöntemi normal olmama, gözlemlerin bağımsız olmaması ve çoklubağlantı gibi bozulumlara karşı sağlam bir yöntemdir. Ölçümlerin dağılımı ne olursa olsun, PLSR kullanılabilir. Bu nedenle, PLSR özellikle normallik varsayımının sağlanmadığı veriler için uygun bir yöntemdir [11, 21].

PLSR modelini elde etmek için literatürde en sık kullanılan yaklaşımlar, Doğrusal Olmayan Yinelemeli Kısmi En Küçük Kareler (Non-linear Iterative Partial Least Squares/NIPALS) ve PLS yönteminin İstatistiksel olarak Esinlenilmiş Değişikliğinin Basit bir Uygulaması (Straightforward Implementation of a Statistically Inspired Modification of the PLS method/SIMPLS) algoritmalarıdır. NIPALS algoritması, tek bir **y** değişkeni olduğunda 'PLS1' ve çoklu **Y** değişkeni için kullanıldığında ise 'PLS2' adını alır. Ancak, NIPALS algoritmasında yüklerin, bileşenlerin ve regresyon katsayılarının hesaplanmasında klasik LS regresyon adımları kullandığından ve SIMPLS algoritması bağımlı ile bağımsız değişkenler arasındaki varyans-kovaryans matrisine ve LS regresyonuna dayalı olduğundan, veri kümesinde aykırı değerler olduğunda her iki algoritma ile elde edilen sonuçlar da etkilenir. Bu nedenle, literatürde PLS1, PLS2 ve SIMPLS algoritmalarının bazı sağlamlaştırılmış biçimleri olan sağlam PLSR algoritmaları önerilmiştir. Sağlam PLSR yapmak için literatürde iki ana yöntem vardır: Birincisi, PLSR modelini elde etmek için kullanılan algoritmalarda yer alan regresyon adımlarında LS yerine sağlam regresyon yöntemlerini kullanarak aykırı değerlerin ağırlığını azaltmak ve ikincisi, bu algoritmalarda kullanılan kovaryans matrisini sağlam bir regresyon yöntemi ile sağlam kestirmektir. İlk yöntem ile oluşturulan sağlam regresyon yöntemleri, yarı-sağlam (semi-robust) olarak nitelendirilir. Çünkü bu yöntemler, ya

sağlam olmayan başlangıç ağırlıklarına sahiptir ya da ağırlıklar kaldıraç gözlemlerine karşı dirençli değildir [11, 15, 21].

Literatürde PLS1 ve PLS2 algoritmalarında bazı değişiklikler yapılarak, önerilen sağlam PLSR yöntemleri vardır. İlk olarak, Wakeling ve Macfie [37] PLS2 algoritmasındaki **X** değişkenine ilişkin **w** ağırlıklarının ve **Y** değişkenine ilişkin **c** ağırlıklarının hesaplandığı tek değişkenli regresyon adımlarını sağlamlaştırılmış biçimleri ile değiştirerek, ilk sağlam PLSR algoritmasını geliştirmiştir. Bu amaç ile yinelemeli olarak yeniden ağırlıklandırılmış en küçük kareler (iteratively reweighted least squares/IRLS) algoritmasından faydalanmıştır. Bu sağlam algoritma, bağımlı **Y** değişkeninde rasgele aykırı değerler olduğunda etkin bulunmuştur [37]. Griep vd. [11] ise, PLS1 algoritmasında **X** değişkenine ilişkin **w** ağırlıklarının hesaplandığı ilk adımında klasik LS yöntemi yerine En Küçük Ortanca Kareler (Least Median Squares/LMS), Siegel'in Tekrarlı Ortanca'sı (Repeated Median/RM) ve IRLS gibi sağlam yöntemler kullanmıştır. Bu çalışmaya göre, daha küçük boyutlarda bu algoritma, aykırı değerler kaç tane ve ne büyüklükte olursa olsun etkin sonuçlar verdiğinden en iyi yöntemdir [11]. Cummins ve Andrews [1] ise, yinelemeli olarak yeniden ağırlıklandırılmış regresyonu, PLS1 algoritmasına genelleştirmeyi önermiştir. Mantık, klasik IRLS'deki ile aynıdır. Ancak, bu algoritmada IRLS'den farklı olarak Çoklu Doğrusal Regresyon'un (Multiple Linear Regression/MLR) artıkları yerine PLSR'nin artıkları kullanılır. Klasik bir Kısmi En Küçük Kareler (Partial Least Squares/PLS) uyguladıktan sonra, ağırlıklar hesaplanır ve bir sonraki PLS algoritması için kullanılır. Bu algoritma, sadece bir tane bağımlı değişken olan model için geçerlidir ve kaldıraç gözlemlerine karşı dirençli değildir [1]. Dodge vd. [4] tarafından geliştirilen Kısmi En Küçük Mutlak Sapmalar (Partial Least Absolute Deviations/PLAD) regresyonu algoritması, PLS1 algoritmasındaki bileşenlerin diklik özelliklerini koruyarak, bağımlı **y** değişkenini elde edilen bileşenlerden En Küçük Mutlak Sapmalar (Least Absolute Deviations/LAD/L_1) regresyon yöntemini kullanarak kestirir. r_i, i. gözlem için artığı göstermek üzere LAD, $\min_{\hat{\boldsymbol{\beta}}} \sum_{i=1}^{n} |r_i|$ şeklinde artıkların mutlak değerlerinin toplamlarını en küçük yapar [4].

Literatürde, PLS1 algoritmasının seçenek bir tanımı olan algoritmadaki kovaryans matrisini sağlam bir yöntem ile kestirerek önerilen sağlam PLSR yöntemleri de mevcuttur. PLS1 algoritmasının seçenek bir tanımı olan bu algoritmada, Helland (1988)'de verilen yaklaşımı kullanarak PLSR bileşenlerini hesaplamadan ve doğrudan algoritmada kullanılan ağırlık matrisini hesaplayarak, sadece üç adımda PLSR kestirimleri elde edilir [10]. Algoritmanın her bir adımında ağırlıklar, sadece daha önce hesaplanan ağırlık vektörlerine ve $\mathbf{S}_{\mathbf{x}}$, $\mathbf{s}_{\mathbf{y,x}}$ kovaryanslarına dayalıdır. Bu nedenle, $\mathbf{z} = (\mathbf{y}, \mathbf{X})'$ şeklinde **X** ve **y** değişkenlerinden oluşan birleşik veri kümesinin kovaryans matrisi sağlamlaştırıldığında da, sürecin sağlamlaştırılacağı düşünülür. Bu amaçla, algoritmanın bu biçimine ilişkin literatürde çeşitli sağlam PLSR yöntemleri önerilmiştir. İlk olarak Gil ve Romera [9], **X** değişkenlerinin örneklem kovaryans matrisi $\mathbf{S}_{\mathbf{x}}$ ile **X** ve **y** değişkenleri arasındaki kovaryans matrisi $\mathbf{s}_{\mathbf{y,x}}$ 'i Stahel-Donoho kestiricisi (Stahel-Donoho estimator/SDE) ile sağlamlaştırarak, sağlam bir PLS1 yöntemi elde etmiştir. Bu yöntem, sadece gözlem sayısının değişken sayısından büyük olduğu (n>p) veriye uyar ve hesaplanması için açık bir algoritma elde edilememiştir. Hesaplanması çok zaman alacağından, SDE her zaman tam biçimi ile hesaplanamamaktadır. Genelde alt örnekleme süreçleri ile birlikte, yakınsama yöntemleri kullanılmaktadır. Bu sağlam PLSR algoritmasını, González vd. [10], 'PLS-SD' olarak adlandırmıştır [7, 9, 10, 15]. Gil ve Romera [9] çalışmasına benzer bir şekilde Kondylis ve Hadi [19], varyans-kovaryans matrisini sağlam bir şekilde kestirmek ve sağlam PLSR analizi yapmak için Billor vd. (2000) tarafından önerilen BACON algoritmasını kullanmıştır. Kondylis ve Hadi [19], önerdikleri sağlam PLSR algoritmasını 'BACON-PLSR (BPLSR)' olarak adlandırmıştır. Hem gerçek veriden hem de benzetim çalışmasından elde edilen sonuçlar, BPLSR algoritmasının aykırı değerlere karşı dirençli olduğunu gösterir. Regresyon kestirimleri, en azından aykırı değerlerin varlığı ile makul düzeylerde bozulmuş veri kümelerinde, aykırı değerlerden fazla etkilenmemektedir. Sadece yüksek bozulma düzeylerinde ve veri kümelerinin göreceli olarak yüksek boyutlularında (p>n), BPLSR'nin kestirim başarısı düşmektedir. González vd.'de [10] ise 'PLS-KurSD' olarak adlandırılan, sağlam bir PLSR algoritması önerilmiştir. Bu algoritma, çok değişkenli aykırı değerlerden temizlenmiş bir kovaryans matrisini elde etmek için tasarlanmıştır. Algoritma, üç adıma sahiptir. İlk adımda, yansımaların (projections) basıklık

katsayısını en büyük ve en küçük yaparak iki özel yön üretilir ve bu yönlerde aykırı değerler için tek değişkenli araştırma yapılır. İkinci adımda, Pena ve Prieto [25]'de söz edilen tabakalı örneklemeye ilişkin süreç takip edilerek rasgele yönler üretilir ve yeniden aykırı değerler tespit edilir. Üçüncü adımda, tüm şüpheli gözlemler örneklemden geçici olarak silinir ve geriye kalan verinin ortalaması ve kovaryans matrisi hesaplanır. Daha sonra, Mahalanobis uzaklığını kullanarak tüm şüpheli gözlemler kontrol edilir. Aykırı gözlemler olarak belirlenen gözlemler örneklemden silinir ve daha fazla aykırı değer bulunmayana kadar, sürecin üç adımı yeni temizlenmiş örnekleme yeniden uygulanır. Yukarıda anlatılan üç adım, daha fazla aykırı değer bulunmayana kadar tekrarlanır. Böylece elde edilen sağlam kovaryans matrisi, PLS1'nin seçenek tanımında kullanılarak sağlam PLSR yöntemi elde edilmiş olur.

Literatürde, klasik SIMPLS algoritmasını sağlamlaştırarak önerilen sağlam PLSR yöntemleri de bulunmaktadır. Hubert ve Vanden Branden [15], çok değişkenli regresyon yöntemi olan En Küçük Kovaryans Determinantı (Minimum Covariance Determinant/MCD) regresyon ve ROBPCA regresyon yöntemlerini kullanarak, algoritmanın sağlamlaştırılmış biçimleri olan, sırasıyla RSIMCD ve RSIMPLS algoritmalarını geliştirmiştir. RSIMPLS ve RSIMCD algoritmalarının her ikisi de, tek ya da birden çok bağımlı değişkenin olduğu durum için de kullanılabilir. Hubert ve Vanden Branden [15], RSIMCD'den iki kat daha hızlı olduğu için RSIMPLS algoritmasının kullanılmasını tavsiye etmiştir. RSIMPLS algoritması hesaplama açısından çok hızlı olduğu için bağımsız değişken sayısının gözlem sayısından fazla olduğu yüksek boyutlu veri kümelerinde kolaylıkla kullanılır [15]. Serneels vd. [31] ise modelde tek bir bağımlı değişken olduğunda M regresyon kestiricilerinin 'kısmi' bir biçimini önermiş ve bu kestirici, Kısmi Sağlam M (partial robust M/PRM) regresyon kestiricisi olarak adlandırılmıştır. Aykırı değerlerin ağırlığı azaltılarak oluşturulan PRM regresyonunda, **y** ve **x** değişkenler uzayındaki aykırı gözlemlerin etkisini azaltmak için yinelemeli bir şemada sıfır ile bir arasında değişen ağırlıklar hesaplanır [21]. Sağlam PLSR'ye ilişkin daha önceki çalışmalarda PLSR kestiricilerinin sağlam biçimlerini geliştirmeye odaklanılmışken, Serneels vd. [31] çalışmasında sağlam bir kestiricinin kısmi bir biçimini önermiştir. Bu şekilde oluşturulan bir başka kestirici de, PLAD kestiricileridir. Ancak, PRM yöntemi

hesaplama açısından daha kolay olduğundan, literatürde daha çok ilgi çekmiştir [7, 31]. Önerilen biçimine yönelik yapılan benzetim çalışmaları, yöntemin iyi bir etkinlik ve yüksek bir sağlamlık özelliğine sahip olduğunu gösterir. PRM hesaplama açısından çok hızlıdır ve yüksek boyutlu veri kümeleri için kullanılabilir [7]. Serneels vd. [32] ise, SIMPLS algoritmasındaki kovaryans matrisi için 'mekansal işaret kovaryans matrisi' olarak adlandırılan, kovaryans matrisinin sağlam bir şeklini kullanılarak sağlam bir PLSR algoritması elde etmiştir. Bu yöntem, veriye mekansal işaret ön işleme (spatial sign preprocessing/SS-PP) ve akabinde standart bir PLS algoritmasının uygulanmasına eş olduğu için 'SS-PP+PLS' olarak adlandırılır [7, 32]. SS-PP kavramsal olarak basittir ve kolay hesaplanır. Çünkü ayarlanması gereken herhangi bir parametre ya da fonksiyon içermez [32]. SS-PP+PLS yönteminin hesaplanmasının kolay, normal modelde kabul edilir bir şekilde etkin ve hata terimleri Cauchy, Laplace gibi normal olmayan bazı dağılımlardan gelen modellerde iyi sonuçlar verdiği ve veri kümesinde aykırı değerlerden kaynaklanan çok fazla bozulmaya karşı kısmen sağlam olduğu sonuçlarına ulaşılmıştır. Mekansal işaret dönüşümünün yan özellikleri RSIMPLS'nin yan özelliklerinden daha az tercih edilebilir olduğundan, mekansal işaret dönüşümünün sadece hesaplama zamanının bir problem olduğu durumlarda kullanılması gerektiği belirtilmiştir [32].

Bu doktora tez çalışmasında, literatürde önerilen sağlam PLSR yöntemleri ayrıntılı olarak incelenmiştir. Literatürde var olan sağlam PLSR yöntemlerden özellikle yukarıda söz edilen PLS1 algoritmasının seçenek tanımındaki kovaryans matrisini sağlam bir kovaryans kestirim yöntemi ile kestirerek önerilen yöntemlerin uygulanış kolaylığı göz önüne alınarak, üç yeni sağlam PLSR yöntemi önerilmiştir. Önerilen bu üç yeni sağlam PLSR yöntemi, dik yükler algoritması olarak da adlandırılan PLS1 algoritması ile hesaplanan klasik PLSR yöntemi ve literatürdeki dört sağlam PLSR yöntemi RSIMPLS, PRM, PLS-SD ve PLS-KurSD ile etkinlik, veriye uyum ve kestirimdeki başarıları açısından benzetim çalışmaları ve gerçek bir veri kümesi üzerinde yapılan uygulamalar ile karşılaştırılmıştır.

Bu çalışma altı bölümden oluşmaktadır. Birinci bölüm olarak ele alınan giriş bölümünde tezin konusu, önemi, bu konuda yapılan önceki çalışmalar, tezin içeriği ve izlenecek düzen ana hatları ile verilmiştir.

İkinci Bölüm'de, klasik PLSR yöntemi hakkında genel bilgi verilmiş ve bu yöntem ile parametre kestirimlerini elde etmek için kullanılan yinelemeli klasik PLSR algoritmaları 'NIPALS (PLS1 ve PLS2)' ile 'SIMPLS' algoritmaları incelenmiştir.

Üçüncü Bölüm'de, ilk olarak sağlam bir kestiricinin sahip olması gereken özellikler ve önemli sağlamlık ölçütlerinden kısaca söz edilmiştir. Daha sonra, literatürdeki sağlam PLSR yöntemleri detaylı olarak tanıtılmıştır.

Dördüncü Bölüm'de, Helland (1988)'de verilen yaklaşımı kullanarak PLSR bileşenlerini hesaplamadan ve doğrudan algoritmada kullanılan ağırlık matrisini hesaplayarak, PLSR kestirimlerinin sadece üç adımda elde edildiği PLS1 algoritmasının seçenek tanımındaki kovaryans matrisini sağlam bir şekilde kestirerek önerilen üç yeni sağlam PLSR yöntemi tanıtılmıştır. *PLS-ARWMCD*, *PLS-Smult* ve *PLS-MMmult* olarak adlandırılan bu üç yeni sağlam PLSR yöntemi sırasıyla, ' *ilk adımda konum ve kovaryansın sağlam başlangıç kestiricileri olarak En Küçük Kovaryans Determinantı kestiricilerini kullanan, uyarlanabilir yeniden ağırlıklandırılmış bir kovaryans kestiricisi* ', ' *S-kestiricileri* ' ve ' *MM-kestiricileri* ' olarak bilinen üç ayrı sağlam kovaryans kestirim yöntemini kullanarak önerilmiştir.

Beşinci Bölüm'de ise, yapılan benzetim çalışmaları ve gerçek bir veri kümesi üzerindeki uygulama ile yeni önerilen üç sağlam PLSR yöntemi *PLS-ARWMCD*, *PLS-Smult*, *PLS-MMmult* ile klasik PLSR yöntemi ve literatürde var olan dört sağlam PLSR yöntemi RSIMPLS, PRM, PLS-SD, PLS-KurSD karşılaştırılmıştır. Yeni önerilen üç sağlam PLSR yöntemi de küçük boyutlu ve makul düzeyde aykırı değerler tarafından bozulan veri kümelerine uygulandığında, özellikle klasik PLSR yönteminden ve bazı durumlarda literatürde var olan sağlam RSIMPLS, PRM,

PLS-SD ve PLS-KurSD yöntemlerinden daha sağlam ve etkin sonuçlar vermektedir.

Altıncı Bölüm'de ise, beşinci bölümde benzetim çalışmaları ve gerçek bir veri kümesi üzerindeki uygulamalar ile elde edilen sonuçlar özetlenmiş ve sonuçlar tartışılmıştır.

2. KISMI EN KÜÇÜK KARELER REGRESYONU

Herman Wold tarafından 1960'lı yıllarda PLSR, bağımsız değişkenlerin çok sayıda ve ilişkili olması durumunda kestirime yönelik modeller elde etmek için yeni bir yöntem olarak geliştirilmiştir. PLSR yönteminde amaç, bağımsız **X** değişkenleri ve bir ya da birden fazla bağımlı **Y** değişkenlerinin oluşturduğu iki veri bloğu için elde edilmiş gizli değişkenler (Latent Variables/LV) kullanılarak, bu iki veri bloğu arasında ilişki bulmaktır. Bu yöntem ilk olarak Herman Wold tarafından 1966 yılında, ekonomik ve sosyal olayları modellemek için kullanılmıştır. PLSR yöntemi kimya bilim alanında Kowalski vd. tarafından 1979 yılında yapılan bir başlangıç çalışmasından sonra kullanılmaya başlanmıştır. 1980 yılından bu zamana kadar, birbirleri ile doğrusal ilişkili **X** ve **Y** değişkenlerinden oluşan iki bloklu özgün PLSR modeli, bilim ve teknoloji alanındaki verileri daha iyi çözümlemek için geliştirilmiştir. Böylece PLSR yöntemi, klasik regresyonun uygulanmasının zor olduğu karmaşık veri kümelerini çözümlemek için daha yararlı olmuştur [26]

Bu bölümde, ilk olarak klasik PLSR yöntemi tanıtılacak ve kullanıldığı alanlar ile kullanım amaçlarından kısaca bahsedilecektir. İkinci olarak, PLSR parametre kestirimlerini elde etmek için kullanılan klasik PLSR algoritmaları PLS1, PLS2 ve SIMPLS tanıtılacaktır.

2.1. Kısmi En Küçük Kareler Regresyon Yöntemi

Çok emek ve uzun süre gerektiren ancak doğru sonuç veren ölçme yöntemlerinin, ucuz, hızlı ve daha az doğruluğa sahip dolaylı ölçme yöntemleriyle yer değiştirmesi, 'ayarlama' olarak tanımlanır. Ayarlama, bir ya da birden çok bağımlı ve birden çok bağımsız değişken olduğunda, 'çok değişkenli ayarlama (multivariate calibration)' olarak adlandırılır. Çok değişkenli ayarlama çalışmaları, Kemometri bilim alanının en yaygın konularından biridir. Kemometri; istatistik, matematik ve bilgisayar kullanılarak kimyasal verilerin işlenmesini kapsayan kimya alanında bir bilim dalıdır. $\hat{\mathbf{Y}} = \mathrm{f}(\mathbf{X})$ şeklindeki kestiricileri veren ayarlama

modellerinden biri de; $\mathbf{T}=h_1(\mathbf{X})$, $\mathbf{Y}=h_2(\mathbf{T})+\mathbf{F}$ ve $\mathbf{X}=h_3(\mathbf{T})+\mathbf{E}$ şeklindeki LV'ler üzerinden regresyondur. Burada **T** özgün değişkenlerden daha az sayıdaki bileşeni temsil ederken, **E** ve **F**, artıkları temsil etmektedir. Bu türün temsilcisi olan PLSR yöntemi, Temel Bileşenler Regresyonu (Principal Component Regression/PCR) yöntemi gibi ayarlama modellemesi yapmak için kullanılan veri sıkıştırma yöntemlerinden bilineer (ikidoğrusal) yöntemlere girmektedir [26].

İlk olarak Kemometri biliminde yeni bir kestirim tekniği olarak ortaya çıkan PLSR yöntemi, daha sonra çok değişkenli ölçümler arasındaki doğrusal ilişkileri modellemek için yerleşmiş bir araç haline gelmiştir. Martens ve Jensen tarafından 1983 yılında, Kemometri bilim dalında tek bir **Y** değişkeninin olduğu veri için daha iyi kestirimler elde etmek amacıyla PLS yöntemi geliştirilmiştir. Wold vd. tarafından ise 1983 yılında, çok değişkenli **Y** için daha iyi kestirimler elde etmek amacıyla PLS yöntemi geliştirilmiştir. Günümüzde PLSR, birçok bilimsel ve teknolojik uygulamalarda sıklıkla kullanılan bir yöntem haline gelmiştir. Özellikle çoklubağlantı durumunda MLR uygulamalarında, kestirim bakımından daha başarılı ve daha az bileşenli modeller elde etmek için kullanılmaya başlanmıştır [26].

PLSR modelinin bulduğu ve bağımlı değişkenleri kesin bir anlamda açıklayacak olan LV'ler, $\mathbf{t}_a(a=1,2,\cdots,k)$ ile gösterilebilir. Burada 'k', modelde kalacak ideal bileşen sayısıdır. En son PLSR modelinde kalacak ideal bileşen sayısı, genellikle en küçük Hata Kareler Ortalamasının Karekökü (Root Mean Squared Error/RMSE) değerini veren bileşen sayısı olarak ya da bir çapraz geçerlik (cross-validation/CV) süreciyle kestirilen tahmin hatasını minimize ederek seçilir [5, 26]. **X** ve **Y**'nin, aynı LV tarafından modellendiği varsayılır. Wold vd. [38], PLSR ve PCR yöntemlerinin benzer ve farklı yönlerini anlatmıştır. Her iki yöntemde de, **y** değişkenlerini kestirmek için **x** değişkenlerinin sayısından daha az bileşen kullanılarak regresyon probleminin boyutu azaltılır. Her bir bileşen, $\mathbf{X}_1,\cdots,\mathbf{X}_p$'nın doğrusal bir birleşimidir. İki yöntem arasındaki temel fark, PCR'de temel bileşenler (Principal Components/PC) bağımlı değişkenleri referans almadan belirlenirken, PLSR'de PC'ler belirlenirken gözlemlenen bağımlı değişkenler önemli rol oynar. PLSR

modelini elde etmek için, literatürde birçok algoritma tanıtılmıştır. Bu amaçla tanıtılan ilk algoritma, klasik NIPALS algoritmasıdır. NIPALS algoritmasına iyi bir seçenek olarak en çok kullanılan algoritma ise SIMPLS algoritmasıdır [26]. Bir sonraki alt bölümlerde, bu algoritmalar detaylı olarak anlatılacaktır.

2.2. Doğrusal Olmayan Yinelemeli En Küçük Kareler (NIPALS) Algoritması

Veri kümesinde tek bir **y** değişkeni olduğunda NIPALS algoritması, 'PLS1' ve çoklu **Y** değişkeni için kullanıldığında ise, 'PLS2' adını alır. Birden çok **y** değişkeni olduğunda, PLS1 yöntemi ile elde edilen modeller her zaman açıklanan varyans bakımından daha iyi modeller vereceğinden, PLS2 modellerine tercih edilir. Ancak, bağımlı değişkenler ilişkili olduğunda PLS2 yönteminin uygulanması daha uygundur. Bu algoritma isteğe bağlı olarak, ölçeklendirilmiş ve merkezileştirilmiş sırasıyla bağımsız ve bağımlı değişkenlerin yer aldığı **X** ve **Y** matrisleri ile başlar ve daha sonra, aşağıdaki adımlarla gösterilen şekilde devam eder. Tek bir **y** değişkeni olduğunda ise, PLS1 adını alan algoritma yinelemeli değildir. Algoritmadan da görüldüğü üzere, bir sonraki yineleme bir önceki yinelemeden elde edilen **X** ve **Y** artık matrisleri ile başlar. Yinelemelere, bir durdurma ölçütü kullanılana kadar ya da **X** sıfır matrisi olana kadar devam edilebilir. PLS2 algoritmasının işleyişi, aşağıdaki adımlar ile gösterilebilir [11, 26, 37, 38].

Adım 1: Yinelemeli algoritmaya, **u** skor vektörü için bir başlangıç değeri olarak, genellikle **Y**'nin ilk sütunu ya da tüm sütunların ortalaması seçilerek başlanır. Tek bir y değişkeni olduğunda, **u**=**y**'dir.

Adım 2: **X** ağırlıkları $\mathbf{w} = \mathbf{X}'\mathbf{u}/\mathbf{u}'\mathbf{u}$ şeklinde, **X**'in **y** skor vektörü **u**'nun üzerine regresyonu ile bulunur. Böylece **Xw** doğrusal birleşimi ile **y** arasında maksimum kovaryans elde edilir. $\|\mathbf{w}\|$ Öklid normunu göstermek üzere, ağırlıklar $\mathbf{w} = \mathbf{w}/\|\mathbf{w}\|$ şeklinde normalleştirilir.

Adım 3: **X** skorları olan **t**, **X**'in satırlarının **w** üzerine yansıması gibi bulunur: $\mathbf{t} = \mathbf{Xw}$.

Adım 4: Daha sonra **c** ile gösterilen **Y** ağırlıkları, $\mathbf{c} = \mathbf{Y't}/\mathbf{t't}$ şeklinde bulunur. **c** ağırlıkları da, $\mathbf{c} = \mathbf{c}/\|\mathbf{c}\|$ şeklinde normalleştirilir.

Adım 5: Son olarak, **Y** skorlarının güncellenmiş bir kümesi, $\mathbf{u} = \mathbf{Yc}$ şeklinde bulunur.

Adım 6: **t**'deki değişimden yararlanılarak, yakınsaklık denetlenir. Örneğin, $\|\mathbf{t}_{\text{eski}} - \mathbf{t}_{\text{yeni}}\|/\|\mathbf{t}_{\text{yeni}}\| < \varepsilon$'dır. Burada ε, 10^{-6} ya da 10^{-8} arasında küçük bir değerdir. Eğer yakınsaklık sağlanmaz ise Adım 2'ye dönülür, sağlanır ise Adım 7 ile ve daha sonra tekrar Adım 1 ile devam edilir. Eğer tek bir **y** değişkeni varsa Adım 4'deki normalleştirilmiş c, 1'e eşit olur ve süreç tek bir yinelemede yakınsar. Daha sonra ise, doğrudan Adım 7 ile devam eder.

Adım 7: Eğer yakınsaklık sağlanır ise **X** ve **Y** matrislerinden, elde edilen bileşen çıkarılır. Bu indirgenmiş yeni artık matrisleri ise, bir sonraki bileşenin elde edilmesinde yeni **X** ve **Y** matrisleri olarak kullanılır. **X** ve **Y** matrisleri indirgenmeden önce **X** yükleri, **Y** yükleri ve b regresyon katsayısı aşağıdaki gibi hesaplanır.

X yükleri $\mathbf{p} = \mathbf{X't}/(\mathbf{t't})$

Y yükleri: $\mathbf{q} = \mathbf{Y'u}/(\mathbf{u'u})$

Regresyon (**t** üzerine **u**'nun): $\mathrm{b} = \mathbf{u't}/(\mathbf{t't})$

Daha sonra, bir sonraki bileşenin hesaplanması için yeni artık matrisleri, $\mathbf{X} \to \mathbf{X} - \mathbf{tp}'$ ve $\mathbf{Y} \to \mathbf{Y} - \mathrm{b}\mathbf{tc}'$ şeklinde elde edilir.

Adım 8: Son olarak, CV, **X** matrisinde **Y** hakkında daha fazla önemli bilgi olmadığını gösterene kadar bir sonraki bileşenin hesaplanmasıyla devam edilir (Adım 1'e geri dönülür) [26].

PLS2 algoritması için Adım 2 ve Adım 4 incelendiğinde, bu adımlar regresyon adımları olarak görülebilir. Örneğin, Adım 2'deki normalleştirilmemiş **w** vektörü, **u** üzerine **X** değişkenin regresyon katsayısı gibidir [37]. PLS1 algoritmasının adımlarını veren, iki benzer biçimi ise aşağıdaki gibi gösterilebilir. Aşağıdaki adımlardan görüldüğü üzere, algoritmada dört farklı regresyon adımı kullanılmaktadır [9, 11]:

Klasik Biçim	*Regresyonlar Türünden Biçimleri*	
$\mathbf{w} = \mathbf{X}'\mathbf{y}$	$\mathbf{w} = \mathbf{X}'\mathbf{y} / \mathbf{y}'\mathbf{y}$	Regresyon
$\mathbf{w} = \mathbf{w} / \lVert\mathbf{w}\rVert$	$\mathbf{w} = \mathbf{w} / \lVert\mathbf{w}\rVert$	Normalleştirme
$\mathbf{t} = \mathbf{X}\mathbf{w}$	$\mathbf{t} = (\mathbf{X}')' \mathbf{w} / \mathbf{w}'\mathbf{w}$	Regresyon
$\mathbf{p} = \mathbf{X}'\mathbf{t} / \mathbf{t}'\mathbf{t}$	$\mathbf{p} = \mathbf{X}'\mathbf{t} / \mathbf{t}'\mathbf{t}$	Regresyon
$b = \mathbf{y}'\mathbf{t} / \mathbf{t}'\mathbf{t}$	$b = \mathbf{y}'\mathbf{t} / \mathbf{t}'\mathbf{t}$	Regresyon
$\mathbf{X} = \mathbf{X} - \mathbf{t}\mathbf{p}'$	$\mathbf{X} = \mathbf{X} - \mathbf{t}\mathbf{p}'$	**X** artık matrisi
$\mathbf{y} = \mathbf{y} - b\mathbf{t}$	$\mathbf{y} = \mathbf{y} - b\mathbf{t}$	**y** artık vektörü

2.2.1. Tek Bir Bağımlı Değişken için Dikleştirilmiş PLSR Algoritması

Kestirim yapmak için PLSR literatüründe kullanılan iki benzer algoritma vardır. Bunlar, Wold vd. (1983)'te tanıtılan dik skorlar algoritması ile Martens ve Naes [22]'te tanıtılan ve daha sonra Denham [3]'te yeniden bahsedilen dik yükler algoritmasıdır. Dik yükler algoritması, kısaca aşağıdaki gibi tanımlanır. Bu

algoritma da, bir önceki bölümde söz edilen algoritma ile aynı sonucu veren bir PLS1 algoritmasıdır [3, 19, 22].

İlk olarak, $\mathbf{X}$ matrisi ve $\mathbf{y}$ vektörü sırasıyla, $\mathbf{X}_0 = \mathbf{X} - \mathbf{1}\bar{\mathbf{x}}'$ ve $\mathbf{y}_0 = \mathbf{y} - \mathbf{1}\bar{y}$ şeklinde merkezileştirilir. İdeal bileşen sayısı 'k' belirlenir. Daha sonra, $a = 1, \cdots, k$ için [3, 22];

- Ağırlıklar hesaplanır: $\mathbf{w}_a = \mathbf{X}'_{a-1}\mathbf{y}_{a-1}$
- Ağırlıklar için ölçeklendirme faktörü hesaplanır: $c = \left(\mathbf{w}'_a\mathbf{w}_a\right)^{-1/2}$
- $\mathbf{w}_a$ ağırlıklarının uzunlukları 1'e ölçeklendirilir: $\mathbf{w}_a = c\mathbf{w}_a$
- Skorlar hesaplanır: $\mathbf{t}_a = \mathbf{X}_{a-1}\mathbf{w}_a$
- Yükler için ölçeklendirme faktörü hesaplanır: $c = \mathbf{t}'_a\mathbf{t}_a$
- $\mathbf{X}$ yükleri hesaplanır: $\mathbf{p}_a = \mathbf{X}'_{a-1}\mathbf{t}_a / c$
- $\mathbf{Y}$ yükleri hesaplanır: $\mathbf{q}_a = \mathbf{y}'_{a-1}\mathbf{t}_a / c$
- $\mathbf{X}$ artıkları hesaplanır: $\mathbf{X}_a = \mathbf{X}_{a-1} - \mathbf{t}_a\mathbf{p}'_a$
- $\mathbf{Y}$ artıkları hesaplanır: $\mathbf{y}_a = \mathbf{y}_{a-1} - \mathbf{t}_a\mathbf{q}_a$

k tane bileşen için bu adımlar hesaplandıktan sonra, en son olarak $\mathbf{b} = \mathbf{W}\left(\mathbf{P}'\mathbf{W}\right)^{-1}\mathbf{q}$ şeklinde regresyon katsayıları elde edilir. Burada, $\mathbf{W}_k = \left(\mathbf{w}_1, \mathbf{w}_2, \cdots, \mathbf{w}_k\right)$, $\mathbf{P}_k = \left(\mathbf{p}_1, \mathbf{p}_2, \cdots, \mathbf{p}_k\right)$ ve $\mathbf{Q}_k = \left(\mathbf{q}_1, \mathbf{q}_2, \cdots, \mathbf{q}_k\right)$'dur [3, 22].

2.2.2. Klasik PLS1 Algoritmasının Seçenek Tanımı

$\mathbf{z} = \left(\mathbf{y}, \mathbf{X}\right)'$, örneklem boyutu n olan 1+p boyutlu bir vektör olsun. Bu vektör, p tane bağımsız değişkenin bir kümesi ve bir tek bağımlı $\mathbf{y}$ değişkeni olarak ayrıştırılabilir. Vektörler, bir boyutlu sütun matrisleri olarak düşünülür. $\mathbf{S}_\mathbf{z}$, Eş. (2.1)'de gösterilen elemanlara sahip bir örneklem kovaryans matrisi olsun [10].

$$\mathbf{S_z} = \begin{pmatrix} \mathbf{s_y^2} & \mathbf{s'_{y,x}} \\ \mathbf{s_{y,x}} & \mathbf{S_x} \end{pmatrix} \tag{2.1}$$

Burada $\mathbf{s_{y,x}}$, $\mathbf{y}$ ve $\mathbf{x}$ değişkenleri arasındaki kovaryansların $p \times 1$ boyutlu vektörüdür. Amaç, $\hat{\mathbf{y}} = \hat{\boldsymbol{\beta}}'\mathbf{x}$ 'i kestirmek olduğunda bağımlı değişkenin, $\mathbf{x}$ değişkenlerinin doğrusal fonksiyonları olan $(\mathbf{t}_1, \cdots, \mathbf{t}_k), k \leq p$ bileşenlerinin bir kümesi tarafından doğrusal olarak açıklanabileceği varsayılır. 'k' modelde kalacak ideal bileşen sayısı olduğuna göre, $n \times p$ boyutlu bağımsız değişkenler matrisi $\mathbf{X}$'in i. satırı $\mathbf{x}_i'$ ile gösterildiğinde Eş. (2.2) ve Eş. (2.3)'deki modeller yazılır [10].

$$\mathbf{x}_i = \mathbf{P}\mathbf{t}_i + \boldsymbol{\varepsilon}_i \tag{2.2}$$

$$\mathbf{y}_i = \mathbf{q}'\mathbf{t}_i + \boldsymbol{\eta}_i \tag{2.3}$$

Eş. (2.2)'deki $\mathbf{P}$, $\mathbf{t}_i = (t_{i1}, \cdots, t_{ik})'$ vektörlerinin $p \times k$ boyutlu yükleri ve $\mathbf{q}$, y yüklerinin k boyutlu vektörüdür. $\boldsymbol{\varepsilon}_i$ ve $\boldsymbol{\eta}_i$ hata vektörleri; sıfır ortalamaya sahip, normal dağılımlı ve ilişkisizdir. $\mathbf{T} = (\mathbf{t}_1, \cdots, \mathbf{t}_k)'$ bileşen matrisi, doğrudan gözlemlenmemiş ve kestirilmelidir. Buna göre, $\mathbf{T}$ matrisinin en çok olabilirlik kestirimi Eş. (2.4)'deki gibidir [10].

$$\mathbf{T} = \mathbf{X}\mathbf{W}_k \tag{2.4}$$

Eş. (2.4)'deki $\mathbf{W}_k = [\mathbf{w}_1, \mathbf{w}_2, \cdots, \mathbf{w}_k]$ ağırlık matrisi, $p \times k$ boyutlu katsayılar matrisidir ve $\mathbf{w}_i, 1 \leq i < k$ 'ler Eş. (2.6)'daki kısıtlar altında $\mathbf{w}_1 \; \alpha \; \mathbf{s_{y,x}}$ olmak üzere Eş. (2.5)'i sağlayacak şekilde elde edilir [10].

$$\mathbf{w}_i = \arg\max_{\mathbf{w}} \operatorname{cov}^2(\mathbf{X}\mathbf{w}, \mathbf{y}) \tag{2.5}$$

$$\mathbf{w}'\mathbf{w} = 1 \text{ ve } \mathbf{w}_i'\mathbf{S_x}\mathbf{w}_j = 0, \quad 1 \leq j < i \tag{2.6}$$

Böylece, elde edilen $(\mathbf{t}_1,\cdots,\mathbf{t}_k)$ bileşenleri diktir ve $\mathbf{w}_i$ vektörleri, Eş. (2.7)'deki matrisin en büyük özdeğerlerine ilişkin özvektörler olarak bulunur. Burada $\mathbf{P}_x(i)$, $\mathbf{S}_x\mathbf{W}_i$ tarafından kapsanan uzayın üzerindeki projeksiyon matrisidir ve Eş. (2.8)'deki gibi gösterilir [10].

$$(\mathbf{I}-\mathbf{P}_x(i))\mathbf{s}_{y,x}\mathbf{s}'_{y,x} \tag{2.7}$$

$$\mathbf{P}_x(i)=(\mathbf{S}_x\mathbf{W}_i)\left[(\mathbf{S}_x\mathbf{W}_i)'(\mathbf{S}_x\mathbf{W}_i)\right]^{-1}(\mathbf{S}_x\mathbf{W}_i)' \tag{2.8}$$

Sonuç olarak, $\mathbf{w}_i$ vektörleri Eş. (2.9) ve Eş. (2.10)'daki gibi yinelemeli olarak hesaplanır [10].

$$\mathbf{w}_1 \,\alpha\, \mathbf{s}_{y,x} \tag{2.9}$$

$$\mathbf{w}_{i+1} \,\alpha\, \mathbf{s}_{y,x}-\mathbf{S}_x\mathbf{W}_i(\mathbf{W}_i'\mathbf{S}_x\mathbf{W}_i)^{-1}\mathbf{W}_i'\mathbf{s}_{y,x}\,,1\le i<k \tag{2.10}$$

Eş. (2.9) ve Eş. (2.10) kullanıldığında, PLS bileşenleri $\mathbf{t}_i$ 'leri hesaplamak gerekmemektedir. Algoritmanın her bir adımında $\mathbf{w}_{i+1}$, sadece i tane önceki vektörler $\mathbf{w}_1,\mathbf{w}_2,\cdots,\mathbf{w}_i$'ye, $\mathbf{S}_x$ ve $\mathbf{s}_{y,x}$'e dayalıdır. Ayrıca, $\mathbf{w}_1$'in hesaplanması da sadece $\mathbf{s}_{y,x}$'e dayalı olduğundan, sonuç olarak $\mathbf{W}$'nin hesaplanması $\mathbf{S}_x$ ve $\mathbf{s}_{y,x}$'in değerleri ile sabitlenmiştir. En son olarak, $\mathbf{t}$ değişkenlerinin ilişkisiz olmasından dolayı Eş. (2.3)'deki $\mathbf{q}$ ile gösterilen regresyon katsayıları ilişkisiz olduğu için, özgün değişkenler için regresyon katsayıları Eş. (2.11)'deki gibi hesaplanır [10].

$$\hat{\boldsymbol{\beta}}_k^{PLS}=\mathbf{W}_k(\mathbf{W}_k'\mathbf{S}_x\mathbf{W}_k)^{-1}\mathbf{W}_k'\mathbf{s}_{y,x} \tag{2.11}$$

Böylece, PLSR regresyon katsayıları Helland (1988)'de verilen algoritma kullanılarak doğrudan hesaplanmış olur [13].

2.3. PLS Yönteminin İstatistiksel Olarak Esinlenilmiş Değişikliğinin Basit Bir Uygulaması (SIMPLS) Algoritması

PLSR regresyon katsayılarını elde etmek için De Jong [2] SIMPLS algoritmasını önermiştir. SIMPLS algoritması NIPALS algoritmasından farklı olarak, bileşenleri indirgenmiş **X** matrisi yerine özgün **X** matrisinin doğrusal birleşimleri olarak tanımlamayı amaçlar. PLS bileşenleri belirli diklik ve normalleştirme kısıtlarına uyarak, bir kovaryans ölçütünü maksimize edecek şekilde belirlenir. SIMPLS, klasik PLSR algoritması olarak da bilinen NIPALS ile her zaman aynı modeli vermemektedir [26]. Bu bölümde SIMPLS algoritması, Hubert ve Vanden Branden [15] makalesinde verilen biçimiyle anlatılmıştır. SIMPLS yönteminde **x** ve **y** değişkenlerinin, Eş. (2.12) ve Eş. (2.13)'deki gibi iki tane doğrusal model ile bağlı olduğu varsayılır [15].

$$\mathbf{x}_i = \bar{\mathbf{x}} + \mathbf{P}_{p,k}\tilde{\mathbf{t}}_i + \mathbf{g}_i \tag{2.12}$$

$$\mathbf{y}_i = \bar{\mathbf{y}} + \mathbf{A}'_{q,k}\tilde{\mathbf{t}}_i + \mathbf{f}_i \tag{2.13}$$

Bu modellerdeki $\tilde{\mathbf{t}}_i$'ler, k boyutlu skorları ve $\mathbf{P}_{p,k}$, **x** yükleri matrisini gösterir. Her bir denklemin artıkları sırasıyla, $\mathbf{g}_i$ ve $\mathbf{f}_i$ ile gösterilir. $\mathbf{A}_{k,q}$, $\mathbf{y}_i$'nin $\tilde{\mathbf{t}}_i$ üzerinden elde edilen regresyon katsayılarını gösterir. PLS'ye ilişkin literatürde, genellikle $\mathbf{A}_{k,q}$ matrisi $\mathbf{Q}'_{k,q}$ olarak gösterilir ve $\mathbf{Q}_{q,k}$'nın kolonları ise, **y** yükleri $\mathbf{q}_a$ ile gösterilir. Ancak Hubert ve Vanden Branden [15], $\mathbf{q}_a$'yı PLS ağırlık vektörünü göstermek için kullandığından, burada **Q** yerine **A** gösterimini tercih etmiştir [15].

Eş. (2.12) ve Eş. (2.13), iki adımlı algoritma anlamına gelir. Veri kümesi merkezileştirildikten sonra, SIMPLS ilk olarak k tane LV, $\tilde{\mathbf{T}}'_{n,k} = \left(\tilde{\mathbf{t}}_1, \cdots, \tilde{\mathbf{t}}_n\right)'$'yi oluşturur ve ikinci olarak, bu k tane LV üzerinden bağımlı değişkenler için regresyon yapılır. $\tilde{\mathbf{T}}_{n,k}$'nın kolonları, bileşenler olarak adlandırılır. $\tilde{\mathbf{x}}_i = \mathbf{x}_i - \bar{\mathbf{x}}$ ve $\tilde{\mathbf{y}}_i = \mathbf{y}_i - \bar{\mathbf{y}}$ olmak üzere, $\tilde{\mathbf{X}}_{n,p}$ ve $\tilde{\mathbf{Y}}_{n,q}$ merkezileştirilmiş veri matrislerini göstersin.

Buna göre, normalleştirilmiş PLS ağırlık vektörleri $\mathbf{r}_a$ ve $\mathbf{q}_a$, her bir $a = 1,\cdots,k$ bileşen için Eş. (2.14)'deki kovaryansı en büyük yapan vektörler olarak tanımlanır [15].

$$\mathrm{Cov}\left(\tilde{\mathbf{Y}}_{n,q}\mathbf{q}_a, \tilde{\mathbf{X}}_{n,p}\mathbf{r}_a\right) = \mathbf{q}_a' \frac{\tilde{\mathbf{Y}}_{q,n}'\tilde{\mathbf{X}}_{n,p}}{n-1}\mathbf{r}_a = \mathbf{q}_a'\mathbf{S}_{yx}\mathbf{r}_a \tag{2.14}$$

Bu ifadede $\mathbf{S}_{yx}' = \mathbf{S}_{xy} = \frac{\tilde{\mathbf{X}}_{p,n}'\tilde{\mathbf{Y}}_{n,q}}{n-1}$, **x** ve **y** değişkenleri arasındaki kovaryans matrisidir. Buna göre $\tilde{\mathbf{t}}_i$ skorlarının elemanları, merkezileştirilmiş $\tilde{\mathbf{t}}_{ia} = \tilde{\mathbf{x}}_i'\mathbf{r}_a$ verisinin doğrusal birleşimleri olarak ya da benzer şekilde $\mathbf{R}_{p,k} = (\mathbf{r}_1,\cdots,\mathbf{r}_k)$ olmak üzere, $\tilde{\mathbf{T}}_{n,k} = \tilde{\mathbf{X}}_{n,p}\mathbf{R}_{p,k}$ şeklinde tanımlanır. Birden fazla çözüm elde etmek için, $\tilde{\mathbf{t}}_j = \tilde{\mathbf{X}}\mathbf{r}_j$ birleşenleri Eş. (2.15)'de gösterildiği gibi dik olmalıdır [15].

$$\mathbf{r}_j'\tilde{\mathbf{X}}'\tilde{\mathbf{X}}\mathbf{r}_a = \tilde{\mathbf{t}}_j'\mathbf{t}_a = \sum_{i=1}^{n}\tilde{\mathbf{t}}_{ij}\tilde{\mathbf{t}}_{ia} = 0, \quad a > j \tag{2.15}$$

Eş. (2.15)'deki koşulu sağlamak için ise, Eş. (2.16)'daki gibi hesaplanan, **x** değişkenleri ve j. bileşen $\tilde{\mathbf{X}}\mathbf{r}_j$ arasındaki doğrusal ilişkiyi tanımlayan **x** yükü $\mathbf{p}_j$ tanıtılır [15].

$$\mathbf{p}_j = \left(\mathbf{r}_j'\tilde{\mathbf{X}}'\tilde{\mathbf{X}}\mathbf{r}_j\right)^{-1}\tilde{\mathbf{X}}'\tilde{\mathbf{X}}\mathbf{r}_j = \left(\mathbf{r}_j'\mathbf{S}_x\mathbf{r}_j\right)^{-1}\mathbf{S}_x\mathbf{r}_j \tag{2.16}$$

Buradaki $\mathbf{S}_x$, **x** değişkenlerinin kovaryans matrisidir. Eş. (2.16)'daki tanım, a>j için $\mathbf{p}_j'\mathbf{r}_a = 0$ olduğu zaman Eş. (2.15)'in gerçekleştiği anlamına gelir. Bu nedenle, PLS ağırlık vektörü $\mathbf{r}_a$, tüm önceki **x** yükleri $\mathbf{P}_{a-1} = [\mathbf{p}_1,\cdots,\mathbf{p}_{a-1}]$'e dik olmalıdır. Sonuç olarak $\mathbf{r}_a$ ve $\mathbf{q}_a$, $\mathbf{P}_{a-1}$'e dik olan bir alt uzaya yansıtılan $\mathbf{S}_{xy}$'nin ilk sol ve sağ tekil vektörleri olarak hesaplanır. Bu yansıma ise, $\{\mathbf{p}_1,\cdots,\mathbf{p}_{a-1}\}$'nin birimdik bir temeli

$\{\mathbf{v}_1,\cdots,\mathbf{v}_{a-1}\}$'yi oluşturarak gerçekleştirilir. Daha sonra $\mathbf{S}_{xy}^{a-1}$, Eş. (2.17)'deki gibi indirgenir ve $\mathbf{r}_a$ ile $\mathbf{q}_a$, $\mathbf{S}_{xy}^{a}$'nın ilk sol ve sağ tekil vektörleri olur [15].

$$\mathbf{S}_{xy}^{a} = \mathbf{S}_{xy}^{a-1} - \mathbf{v}_a\left(\mathbf{v}_a'\mathbf{S}_{xy}^{a-1}\right) \tag{2.17}$$

Bu yinelemeli algoritmaya $\mathbf{S}_{xy} = \mathbf{S}_{xy}^{1}$ ile başlanır ve k tane bileşen elde edilinceye kadar süreç tekrar edilir. Algoritmanın ikinci aşamasında, bağımlı değişkenler bu bileşenlerden kestirilir. Bu nedenle, ilgilenilen regresyon modeli Eş. (2.18)'deki gibidir [15].

$$\mathbf{y}_i = \boldsymbol{\alpha}_0 + \mathbf{A}_{q,k}'\tilde{\mathbf{t}}_i + \mathbf{f}_i \tag{2.18}$$

Bu eşitlikteki $\mathbf{f}_i$ için, $E(\mathbf{f}_i) = 0$ ve $\mathrm{Cov}(\mathbf{f}_i) = \boldsymbol{\Sigma}_f$'dir. MLR yöntemi ile Eş. (2.19), Eş. (2.20) ve Eş. (2.21)'deki kestirimler elde edilir [15].

$$\begin{aligned}\hat{\mathbf{A}}_{k,q} &= \left(\mathbf{S}_t\right)^{-1}\mathbf{S}_{ty} = \left(\mathbf{T}_{k,n}'\mathbf{T}_{n,k}\right)^{-1}\mathbf{T}_{k,n}'\mathbf{Y}_{n,q} \\ &= \left(\mathbf{R}_{k,p}'\mathbf{X}_{p,n}'\mathbf{X}_{n,p}\mathbf{R}_{p,k}\right)^{-1}\mathbf{R}_{k,p}'\mathbf{X}_{p,n}'\mathbf{Y}_{n,q} = \left(\mathbf{R}_{k,p}'\mathbf{S}_x\mathbf{R}_{p,k}\right)^{-1}\mathbf{R}_{k,p}'\mathbf{S}_{xy}\end{aligned} \tag{2.19}$$

$$\hat{\boldsymbol{\alpha}}_0 = \bar{\mathbf{y}} - \hat{\mathbf{A}}_{q,k}'\bar{\tilde{\mathbf{t}}} \tag{2.20}$$

$$\mathbf{S}_f = \mathbf{S}_y - \hat{\mathbf{A}}_{q,k}'\mathbf{S}_t\hat{\mathbf{A}}_{k,q} = \mathbf{Y}_{q,n}'\mathbf{Y}_{n,q} - \hat{\mathbf{A}}_{q,k}'\mathbf{T}_{k,n}'\mathbf{T}_{n,k}\hat{\mathbf{A}}_{k,q} \tag{2.21}$$

$\mathbf{S}_y$ ve $\mathbf{S}_t$, sırasıyla $\mathbf{y}$ değişkenlerinin ve $\mathbf{t}$ bileşenlerinin kovaryans matrisini gösterir. Burada MLR, birden fazla $\mathbf{x}$ değişkeni ile yapılan ve q>1 olduğunda ise çok değişkenli çoklu regresyon olarak da bilinen klasik LS regresyonunu gösterir. $\bar{\tilde{\mathbf{t}}} = 0$ olduğu için, $\boldsymbol{\alpha}_0$ sabiti $\bar{\mathbf{y}}$ ile kestirilir [15].

$\tilde{\mathbf{t}}_i = \mathbf{R}'_{k,p}(\mathbf{x}_i - \bar{\mathbf{x}})$'yi Eş. (2.13)'de yerine koyarak Eş. (2.22)'deki asıl model için kestirimler, Eş. (2.23)'deki gibi elde edilir. Eş. (2.22)'deki $\mathbf{e}_i$ hata terimleri için $E(\mathbf{e}_i) = 0$ ve $\text{cov}(\mathbf{e}_i) = \mathbf{\Sigma}_e$'dir. Burada $\mathbf{\Sigma}_e$, q boyutlu kovaryans matrisidir [15].

$$\mathbf{y}_i = \boldsymbol{\beta}_0 + \mathbf{B}'_{q,p}\mathbf{x}_i + \mathbf{e}_i \tag{2.22}$$

$$\hat{\mathbf{B}}_{p,q} = \mathbf{R}_{p,k}\hat{\mathbf{A}}_{k,q} \text{ ve } \hat{\boldsymbol{\beta}}_0 = \bar{\mathbf{y}} - \hat{\mathbf{B}}'_{q,p}\bar{\mathbf{x}} \tag{2.23}$$

En son olarak, $\mathbf{S}_f$'yi özgün parametreler cinsinden yazarak, $\mathbf{\Sigma}_e$'nin bir kestirimi Eş. (2.24)'deki gibi elde edilir. q=1 şeklinde tek bir bağımlı değişken olduğunda, $\hat{\mathbf{B}}_{p,1}$ parametre kestirimleri $\hat{\boldsymbol{\beta}}$ vektörü şeklinde yeniden yazılabilir ve $\mathbf{S}_e$ hata varyansı, $\hat{\sigma}_e^2 = s_e^2$'ye sadeleşir [15].

$$\begin{aligned}\mathbf{S}_e &= \mathbf{S}_y - \hat{\mathbf{A}}'_{q,k}\mathbf{S}_t\hat{\mathbf{A}}_{k,q} = \mathbf{S}_y - \hat{\mathbf{A}}'_{q,k}\mathbf{T}'_{k,n}\mathbf{T}_{n,k}\hat{\mathbf{A}}_{k,q} \\ &= \mathbf{S}_y - \hat{\mathbf{A}}'_{q,k}\mathbf{R}'_{k,p}\mathbf{X}'_{p,n}\mathbf{X}_{n,p}\mathbf{R}_{p,k}\hat{\mathbf{A}}_{k,q} = \mathbf{S}_y - \hat{\mathbf{B}}'_{q,p}\mathbf{S}_x\hat{\mathbf{B}}_{p,q}\end{aligned} \tag{2.24}$$

Standart NIPALS algoritmasına göre, SIMPLS algoritmasının birkaç avantajı vardır. İlk olarak, bileşenler doğrudan özgün veri matrisi cinsinden hesaplandığından, **R** ağırlıkları alışılmış **W** ağırlıklarından daha basit bir yoruma sahiptir. İkinci olarak, özgün değişkenlerin basit doğrusal birleşimleri olarak bileşenleri yorumlamak daha kolay olur. Bir başka avantajı ise, bileşenler özgün değişkenlerin doğrusal birleşimleri olarak ifade edildiğinde, en son PLSR modelinin kolayca elde edilebilmesidir. SIMPLS algoritmasında, NIPALS algoritmasında olduğu gibi, **X** ve **Y** veri matrislerini indirgemek gerekmemektedir. Böylece, hesaplama daha hızlı olur ve daha az hafıza gerekir. SIMPLS tek değişkenli **y** için tamamen PLS1'e benzerken, çok değişkenli **Y** için PLS2'den farklılık gösterir [2].

3. SAĞLAM KISMİ EN KÜÇÜK KARELER REGRESYONU

Bu bölümde, ilk olarak Alt Bölüm 3.1'de sağlam bir kestiricinin sahip olması gereken özellikler ve önemli sağlamlık ölçütlerinden söz edilecektir. Daha sonra, Alt Bölüm 3.2'de sağlam PLSR yöntemlerinin elde ediliş nedeni ve hangi yöntemler ile hangi klasik PLSR algoritmalarından elde edildikleri hakkında kısaca bilgi verilecektir. Alt Bölüm 3.3 ve daha sonraki alt bölümlerde ise, literatürdeki sağlam PLSR yöntemleri daha detaylı bir biçimde tanıtılacaktır.

3.1. Sağlam Kestiricilerin Sahip Olması Gereken Özellikler ve Önemli Sağlamlık Ölçütleri

Sağlam bir kestirici için istenen önemli bir özellik, 'dirençlilik' olarak ifade edilebilir. Eğer bir kestirici az sayıdaki büyük hatadan ya da herhangi bir miktardaki yuvarlama-gruplama hatalarından sınırlı miktarda etkileniyor ise, o kestirici dirençlidir denir [18]. Sağlam kestiriciler, aykırı değerlerin büyük oranına karşı ya da varsayımlardan sapmalara karşı dirençli olmalıdır. Farklı koşullar için farklı sağlam yöntemleri karşılaştırabilmek için, bu yöntemlere ilişkin sağlamlık ölçütleri gereklidir [23]. Bu alt bölümde, bir kestiricinin sağlamlık özelliklerini değerlendirmek için literatürde en çok kullanılan iki ölçüt tanıtılacaktır: kırılma noktası (breakdown point/BDP) ve etki fonksiyonu (influence function/IF).

3.1.1. Kırılma Noktası

Bir kestirici için BDP, gözlemlerin ne kadarı bozulduğunda kestiricinin kırılmayacağını, yani kestirimlerde sağlam sonuçlar elde edeceğini gösteren bir ölçüttür. Bu ölçüt, gözlem sayısına dayalı olarak (1/n) ya da $n \to \infty$ için sınırlandırılmış bir değer olarak (% 0) verilebilir. Eğer bir kestiricinin BDP'si sıfırdan büyük ise, bu kestirici dirençlidir denir [18]. Bir kestiricinin BDP değerinin sıfır olması, tek bir aykırı değerin varlığının bile modeli değiştirebileceği anlamına gelir. Örneğin, LS kestiricisinin BDP değeri sıfırdır. Bir kestiriciye ilişkin en yüksek BDP değeri, % 50 olabilir. Çünkü verideki aykırı değerlerin yüzdesi, % 50'den fazla olur

ise verinin, iyi ve kötü kısımlarını ayırmak imkansız hale gelir [36]. BDP değeri % 50 olan ve yüksek dayanıklılık gösteren kestiriciler, yüksek kırılma noktasına (high breakdown point/HBDP) sahip kestiriciler olarak adlandırılır [23].

$T(\mathbf{Z}) = T(\mathbf{X}, \mathbf{y})$ regresyon parametre kestiricisini göstermek üzere, bu kestiricinin BDP'si sonlu örneklem için Eş. (3.1)'deki gibi ifade edilebilir [39].

$$\varepsilon_n^*(\mathbf{T}, \mathbf{Z}) = \min\left\{\frac{m}{n}; \sup\|T(\mathbf{Z}')\| = \infty\right\} \tag{3.1}$$

Burada $\mathbf{Z}' = (\mathbf{X}', \mathbf{y}')$, 'm' aykırı gözlem sayısını göstermek üzere, $\mathbf{Z} = (\mathbf{X}, \mathbf{y})$ verisinde m tane gözlemin keyfi olarak başka gözlemlerle değiştirilmesi ile elde edilir [39]. Eş. (3.1)'e göre BDP, kestiricinin parametre değerinden uzaklaşmasına neden olacak aykırı gözlem sayısının toplam gözlem sayısına oranının supremumu (sup) olarak tanımlanabilir. Staudte ve Sheather (1990), kestiricilerin BDP'sinin, kestiricideki değişim için bir sınır olarak düşünüldüğünde, bu sınırın aşılmasına neden olan en yüksek aykırı gözlem oranının BDP'yi vereceğini ifade etmiştir [7].

3.1.2. Etki Fonksiyonu

1986 yılında Hampel tarafından etki fonksiyonu (influence function/IF) tanıtılmıştır. IF, veride bulunan ölçülemeyecek kadar küçük bir bozulmanın, bir kestirici üzerindeki etkisini ölçmektedir. IF ile farklı sağlam yöntemlerin tek bir aykırı değer altında, daha detaylı niteliksel bir karşılaştırılması yapılır. Eğer tek bir aykırı değer bile kestiricinin kırılmasına neden olur ise, IF sınırlandırılmış olur. IF'yi değerlendirmek için, veriye ilişkin dağılımsal varsayımlar yapılmalıdır. Bu durum ise analizi genellikle daha karmaşık bir hale getirir ve özellikle n<p için, deneysel karşılaştırmalar gerektirebilir. Bir G dağılımı için bir T kestiricisinin IF'si Eş. (3.2)'deki gibi tanımlanır [7, 23].

$$IF(\mathbf{z}, T, G) = \lim_{\varepsilon \to 0} \frac{T[(1-\varepsilon)G + \varepsilon \delta_{\mathbf{z}}] - T(G)}{\varepsilon} \quad (3.2)$$

Burada ε, aykırı değerler tarafından verideki bozulmuş kısımdır ve $\delta_{\mathbf{z}}$, tüm ağırlığı **z**'ye koyan bir olasılık ölçüsüdür. **z**, p boyutlu uzayda herhangi bir nokta olabilir. Bir veri kümesindeki aykırı değerin kestirici üzerindeki etkisi, aykırı değere ilişkin IF'yi değerlendirerek ölçülebilir. Sağlam olmayan kestiriciler için aykırı değerde IF'nin değerlendirilmesi, normal bir veride IF'yi değerlendirme ile karşılaştırıldığında farklı sonuçlar verecektir. Ancak, sağlam bir kestirici için etki kısıtlı olacaktır [7]. Tek bir aykırı değere karşı kestirimin sağlamlığı IF tarafından, verilerde daha çok bozulmaya karşı kestirimin sağlamlığı ise, kestiricinin sağlıklı olmasını engelleyen aykırı değerlerin en küçük oranı olan BDP tarafından ölçülmektedir [36].

Sağlam kestiriciler tasarlanırken sadece sağlamlık özelliklerini araştırmak değil, aynı zamanda etkinliklerini araştırmak da önemlidir [18]. Bu nedenle, bir sonraki alt bölümde istatistiksel etkinlik hakkında kısaca bilgi verilecektir.

3.1.3. İstatistiksel Etkinlik

Sağlam bir kestirici için istenen diğer önemli bir özellik 'istatistiksel etkinlik' olarak ifade edilebilir. Bir kestiricinin varyansı her bir dağılım için minimuma yakın bir değer alıyor ise, bu kestirici 'yüksek etkinliğe' sahiptir denir. Yanlı kestiriciler için, varyans yerine Hata Kareler Ortalamasına (Mean Square Error/MSE) bakılır. Yüksek etkinlik özelliği, bir kestiricinin dağılımın belli olmadığı durumlarda tekrarlı olarak alınan örneklemler için de iyi sonuçlar verdiğini garanti altına alır. Örneklemin bozulduğu durumlarda, kestirimlerin çok az miktarda değişim göstermesi önemli bir durumdur [1, 7, 18].

3.2. Sağlam Kısmi En Küçük Kareler Regresyon Yöntemleri

Buraya kadar olan kısımda, sağlam bir kestiricinin sahip olması gereken özellikler hakkında kısaca bilgi verilmiştir. Bu bilgiler verildikten sonra bu alt bölümde, sağlam PLSR yöntemlerinin ortaya çıkış nedeni ve hangi klasik PLSR algoritmalarından faydalanarak hangi sağlam regresyon ya da sağlam kovaryans kestirim yöntemlerini kullanarak bulundukları hakkında kısaca bilgi verilecektir.

Klasik PLSR, çok değişkenli veri analizinde kullanılan iyi bir yöntemdir. PLSR yöntemi normal olmama, gözlemlerin bağımsız olmaması ve çoklubağlantı gibi bozulumlara ve gözlemlere göre çok fazla değişken olması durumuna karşı sağlam bir yöntemdir. Ölçümlerin dağılımı ne olursa olsun, PLSR kullanılabilir. Bu nedenle, PLSR özellikle normallik varsayımlarının sağlanmadığı kimyasal veriler için uygun bir yöntemdir. Ancak PLSR yönteminde yüklerin, bileşenlerin ve regresyon katsayılarının hesaplanmasında klasik LS regresyon adımları kullandığından, klasik PLS yöntemi de LS yöntemi gibi veri kümesindeki aykırı değerlerin varlığından etkilenir. Bu nedenle, literatürde sağlam PLSR yöntemleri önerilmiştir. Sağlam PLSR analizi yapmak için literatürde iki ana yöntem vardır: Birincisi, PLSR modelini elde etmek için kullanılan algoritmalarda yer alan regresyon adımlarında LS yerine sağlam regresyon yöntemlerini kullanarak aykırı değerlerin ağırlığını azaltmak ve ikincisi, sağlam bir regresyon yöntemi ile kovaryans matrisini sağlam kestirmektir. İlk yöntem ile oluşturulan sağlam regresyon yöntemleri, yarı-sağlam (semi-robust) olarak nitelendirilir. Çünkü bu yöntemler, ya sağlam olmayan başlangıç ağırlıklarına sahiptir ya da ağırlıklar kaldıraç gözlemlerine karşı dirençli değildir [11, 21].

Literatürde PLSR modelini elde etmek için en çok kullanılan algoritmalar olan NIPALS ve SIMPLS algoritmaları, veri kümesindeki aykırı değerlere karşı çok hassastır. NIPALS algoritması tek bir bağımlı değişken için kullanıldığında ‘PLS1’ ve birden çok bağımlı değişken için kullanıldığında ise ‘PLS2’ adını alır. PLS1, PLS2 ve SIMPLS algoritmalarının bazı sağlamlaştırılmış biçimleri önerilmiştir. Bir sonraki alt bölümde ilk olarak, PLS1 ve PLS2 algoritmalarının çeşitli adımlarında

yapılan değişiklikler ile elde edilen sağlam PLSR yöntemleri hakkında bilgi verilecektir.

3.3. PLS1 ve PLS2 Algoritmalarından Elde Edilen Sağlam PLSR Yöntemleri

İlk olarak, yeniden ağırlıklandırma yöntemlerine dayalı olarak sağlam PLSR kestirimleri veren sağlam PLS1 ve PLS2 algoritmaları hakkında bilgi vermeden önce, kısaca Yinelemeli Olarak Yeniden Ağırlıklandırılmış En Küçük Kareler (Iteratively Reweighted Least Squares/IRLS) yöntemi hakkında bilgi vermek gerekir.

3.3.1. Yinelemeli Olarak Yeniden Ağırlıklandırılmış En Küçük Kareler (IRLS) Yöntemi

y, Çoklu Doğrusal Regresyon (Multiple Linear Regression/MLR) yöntemini kullanarak **X** matrisinden kestirilsin. Buna göre, IRLS yöntemi aşağıdaki adımlar ile tanımlanır [9].

1. $\hat{\boldsymbol{\beta}} = (\mathbf{X}'\mathbf{X})^{-1}\mathbf{X}'\mathbf{y}$ regresyon katsayısının başlangıç değeri hesaplanır.
2. Regresyonun $\mathbf{r} = \mathbf{y} - \mathbf{X}\hat{\boldsymbol{\beta}}$, artıkları hesaplanır.
3. İlgili artıklara göre, her bir gözlem için ağırlıklar hesaplanır. Küçük artıklara sahip gözlemler, büyük ağırlıklara (bire yakın) ve büyük artıklara sahip gözlemler, küçük ağırlıklara (sıfıra yakın) sahiptir. Ağırlık vektöründen, köşegen bir Φ matrisi oluşturulur.
4. Ağırlıklar kullanılarak yeni bir regresyon katsayısı hesaplanır: $\hat{\boldsymbol{\beta}} = (\mathbf{X}'\boldsymbol{\Phi}'\boldsymbol{\Phi}\mathbf{X})^{-1}\mathbf{X}'\boldsymbol{\Phi}'\boldsymbol{\Phi}\mathbf{y}$
5. Bir yakınsama kriteri seçilir ve kriter sağlanana kadar, 2 ile 4 arasındaki adımlar tekrarlanır.

IRLS yönteminden kısaca bahsettikten sonra, bir sonraki iki alt bölümde bu yöntemi PLS1 ve PLS2 algoritmalarında kullanarak elde edilen sağlam PLSR yöntemleri tanıtılacaktır.

3.3.2. İç Yinelemeli Yeniden Ağırlıklandırma PLSR Algoritmaları

Tüm süreci tamamen sağlam yapacak şekilde, PLS1 algoritmasındaki tüm adımları sağlam süreçler ile yer değiştirmek mümkündür. Ancak, adımların hepsini sağlam yapmak yerine, bir ya da iki seçilmiş adım sağlamları ile yer değiştirildiğinde, algoritma aykırı değerler ile baş etmek için başarılı olabilir. Bu tür yarı sağlam süreçler, sağlam yöntemler PLS algoritmasının içerisine uygulandığından, 'İç Yinelemeli Yeniden Ağırlıklandırma (Internal Iterative Reweighting: PLSIR)' algoritmaları olarak da bilinir [9, 11]. Wakeling ve Macfie [37] ve Griep vd. [11] tarafından önerilen sağlam PLSR algoritmaları, PLSIR algoritmalarıdır.

Wakeling ve Macfie [37], PLS2 algoritmasındaki **X** değişkenlerine ilişkin **w** ağırlıklarının ve **Y** değişkenlerine ilişkin **c** ağırlıklarının hesaplandığı tek değişkenli regresyon adımlarını sağlamlaştırılmış biçimleri ile değiştirerek, ilk sağlam algoritmayı geliştirmiştir. Bu amaçla, IRLS algoritmasından faydalanmıştır. Bu sağlam algoritma, **Y** matrisinde rasgele aykırı değerler olduğunda etkin bulunmuştur [37].

Griep vd. [11] ise, PLS1 algoritmasında **X** değişkenine ilişkin **w** ağırlıklarının hesaplandığı ilk adımında klasik LS yöntemi yerine En Küçük Ortanca Kareler (Least Median Squares/LMS), Siegel'in Tekrarlı ortanca (Repeated median/RM) ve IRLS gibi sağlam yöntemler kullanmıştır. Bu çalışmaya göre, daha küçük boyutlarda IRLS, aykırı değerler kaç tane ve ne büyüklükte olursa olsun etkin sonuçlar verdiğinden en iyi yöntemdir. Gerçekten de PLS1 algoritmasının ilk adımı, sadece bir regresyon değildir. Ancak, her bir $\mathbf{x}_i$ değişkenin **y** değişkeni üzerine basit regresyonlarından oluşmuştur. Bu nedenle bu ilk adımda IRLS'nin uygulanması, her bir basit regresyona IRLS'nin uygulanması anlamına gelir.

Ancak, en büyük boyutlarda IRLS etkinliğini kaybeder. Bu nedenle, Griep vd. [11] çalışmasına ilişkin bir eleştiri, veri $[\mathbf{x}_1, \mathbf{y}], [\mathbf{x}_2, \mathbf{y}], \cdots, [\mathbf{x}_p, \mathbf{y}]$ düzlemlerine yansıtılıp aykırı değerler aranırken, verinin çok değişkenli doğasının unutulmasıdır. Gerçekten de büyük boyutlarda, bu düzlemlere yansımalar yapıldığında, teşhis edilemeyecek aykırı değerler var olabilir. Ayrıca Griep vd. [11], tek bir veri kümesi için elde ettikleri sonuçların, sağlam PLSR'nin uygulanabileceği tüm veri kümelerine genellenemeyeceğini ve daha çok veri kümesi için, sağlam PLS1'in hem modelleme hem de kestirim için başarısının değerlendirilmesi gerektiğini belirtmiştir [9, 11].

3.3.3. Dış Yeniden Ağırlıklandırma PLSR Algoritması

Cummins ve Andrews [1], yinelemeli olarak yeniden ağırlıklandırılmış regresyonu, PLS1 algoritmasına genelleştirmeyi önermiştir. Mantık, IRLS'deki ile aynıdır. Ancak, bu kez PLSR'nin artıkları kullanılır. Klasik bir PLSR uyguladıktan sonra, ağırlıklar hesaplanır ve bir sonraki PLSR algoritması için kullanılır. Bu nedenle bu algoritma, bir 'Dış Yeniden Ağırlıklandırma (External Iterative Reweighting: IRPLS)' algoritması olarak da bilinir. Adımlar, aşağıdaki gibi olur [1, 9].

0. Bir ağırlık fonksiyonu seçilir.
1. Klasik bir PLSR analizi yapılır.
2. 1. adımdaki regresyon artıkları, ağırlık fonksiyonuna geçirilir.
3. Bu elde edilen ağırlıklar ile ağırlıklandırılmış PLSR yapılır.
4. 3. adımdan elde edilen artıklar, seçilen ağırlık fonksiyonuna koyulur. Ağırlık fonksiyonuna klasik ağırlıkları koymaktan ise kestirilmiş artıkları koyarak, daha iyi sonuçların elde edildiği bulunmuştur.
5. Eğer yakınsaklık kriterine ulaşılır ise durdurulur, aksi takdirde 3. adıma gidilir.

Uygulamada PLSR için ideal bileşen sayısı bilinmez ve tahmin edilmesi gerekir. Bu amaçla, genelde birini-dışarıda-bırakma çapraz geçerlik (leave-one-out cross-validation/LOOCV) yöntemi kullanılır. y_i , **y** vektörünün i. elemanı ve $\hat{y}_{(i),k}$, i. gözlem çıkarıldıktan sonra k bileşenli PLSR için **y**'nin kestirimi olsun. Buna göre, kestirim hata kareler toplamı (prediction error sum of squares/PRESS) Eş. (3.3)'deki gibi hesaplanır [9].

$$PRESS_k = \sum_{i=1}^{n} \frac{\left(y_i - \left(\hat{y}_{(i),k}\right)_i\right)^2}{n} \tag{3.3}$$

PRESS istatistiği, modelin geçerliğinin ve kestirimdeki başarısının bir ölçüsü olarak kullanılır ve en küçük PRESS değerini veren bileşen sayısı, modelde kullanılmak istenir. PLSR modelinde kalacak ideal bileşen sayısını seçmek için kullanılan diğer bir kriter, çapraz geçerlik R^2'yi (cross-validated R^2/R^2_{CV}) en büyük yapmaktır. R^2_{CV} değeri, Eş. (3.4)'deki gibi hesaplanır [1, 9].

$$R^2_{CV} = 1 - \frac{PRESS}{KT_y} \tag{3.4}$$

Yukarıdaki algoritmayı bir durdurma kuralı, şu şekildedir: 'Eğer mevcut yinelemedeki R^2_{CV} değeri, bir önceki yinelemedeki R^2_{CV} değeri ile karşılaştırıldığında % 5'den daha az değişmiş ise dur'. Böylece yakınsaklığın çok hızlı bir şekilde gerçekleştiği bulunmuştur. 3. adımdaki ağırlıklandırılmış PLSR ilk olarak, ideal bileşen sayısını seçmek için CV ile yapılmıştır. Daha sonra, bu adım için en son bir çalıştırma, ideal bileşen sayısını kullanarak yapılmıştır. Ancak, bu yönteme ilişkin problem, her bir gözlem için artıkların güçlü bir şekilde ideal bileşen sayısı k'ya dayalı olmasıdır. Bu nedenle, k'yı seçmek için farklı bir kriter kullanılması, her bir gözlem için farklı bir ağırlığa neden olacağından yakınsama problemleri olabilir. Ayrıca bu algoritma, sadece bir tane bağımlı değişken olan model için geçerlidir ve kaldıraç gözlemlerine karşı dirençli değildir [1, 9, 15].

İç yinelemeli ve dış yeniden ağırlıklandırma sağlam PLSR algoritmaları hakkında bilgi verdikten sonra, bir sonraki alt bölümde En Küçük Mutlak Sapmalar (Least Absolute Deviations/LAD) regresyon yöntemini PLS1 algoritmasında kullanarak elde edilen ve 'PLAD' şeklinde isimlendirilen sağlam PLSR yöntemi tanıtılacaktır.

3.3.4. Kısmi En Küçük Mutlak Sapmalar (PLAD) Yöntemi

r_i, i. gözlem için artık değerini göstermek üzere LAD, $\min_{\hat{\boldsymbol{\beta}}} \sum_{i=1}^{n} |r_i|$ şeklinde artıkların mutlak değerlerinin toplamını en küçük yapar. Dodge vd. [4], PLS1 algoritmasındaki bileşenlerin diklik özelliklerini koruyarak, bağımlı **y** değişkenini elde edilen bileşenlerden LAD regresyon yöntemini kullanarak kestiren, PLAD regresyonu algoritmasını önermiştir. PLS1 algoritmasında $p \times k$ boyutlu ağırlık ya da yükler matrisi olarak adlandırılan **w**, $\mathbf{w}_{jk} = \mathrm{Cov}(\mathbf{x}_{jk}, \mathbf{y}_k), j = 1, \cdots, p$ şeklinde belirlenir. PLAD algoritmasında ise **w**, Eş. (3.5)'deki kovaryans kestirimini kullanılarak elde edilir [4].

$$\mathbf{w} = \mathrm{cov}_{MAD}(\mathbf{x}_j, \mathbf{y}) = \frac{1}{4}\left(MAD(\mathbf{x}_j + \mathbf{y}) - MAD(\mathbf{x}_j - \mathbf{y})\right) \tag{3.5}$$

Buradaki MAD kısaltması, $MAD(\mathbf{y}) = \mathrm{ortanca}\left(|\mathbf{y} - \mathrm{ortanca}(\mathbf{y})|\right)$ şeklinde hesaplanan, ortanca mutlak sapmayı (median absolute deviation/MAD) gösterir. Böylece PLAD bileşenleri, PLS1'in aksine ortalama yerine ortancayı ve varyansın MAD kestiricisini kullanarak elde edilmiştir. Ortanca kullanıldığı için, PLAD bileşenlerinin aykırı değerlere karşı sağlam olması beklenir [4].

PLAD algoritmasındaki temel özellik, elde edilen bileşenleri kullanarak **y**'yi kestirirken LAD kestirimlerinin kullanılmasıdır. Dodge vd. [4], gözlem sayısı bağımsız değişken sayısından küçük olan (n<p) ve bağımsız değişkenleri arasında çoklubağlantı olan iki veri kümesi üzerinde, PLAD regresyonunu PLS1 ile karşılaştırmıştır. Gerçek veri kümeleri üzerindeki uygulamalar sonucunda, LAD kestirimlerinin aykırı değerler içeren küçük veri kümelerinde sıklık ile görülen uzun

kuyruklu dağılımlara ya da simetrik olmayan hata dağılımlarına karşı tam tamına iyi uyduğu görülmüştür. Ayrıca, PLAD regresyon algoritmasının veride aykırı değerler olduğunda, modelde kalacak ideal bileşen sayısını da doğru bir şekilde seçtiği görülmüştür. PLAD regresyon algoritmasının, yüzlerce bağımsız değişkenli ve değişken sayısına kıyasla orta derecede gözlem sayısına sahip veri kümelerine etkin bir şekilde uygulanabildiği belirtilmiştir [4].

Alt Bölüm 3.4'de ise, Alt Bölüm 2.2.2'de bahsedilen ve klasik PLS1 algoritmasının seçenek tanımı olan algoritmadaki kovaryans matrisini, sağlam bir kovaryans kestiricisi ile kestirerek önerilen literatürdeki sağlam PLSR algoritmaları incelenecektir.

3.4. Klasik PLS1 Algoritmasının Seçenek Tanımındaki Kovaryans Matrisini Sağlam Bir Yöntem ile Kestirerek Önerilen Sağlam PLSR Yöntemleri

Daha önce Alt Bölüm 2.2.2'de, PLS1 algoritmasındaki bileşenleri hesaplamadan sadece doğrudan $\mathbf{W}_k$ ağırlık matrisini hesaplayarak PLSR kestirimlerinin elde edilebildiği gösterilmiştir. Böylece algoritmanın uygulanışı iki adım süreci gibi düşünülür: İlk adımda, yeni dik bağımsız $\mathbf{t}_i$ değişkenlerini tanımlayan $\mathbf{w}_i$ ağırlıkları, gözlemlerin kovaryans matrisini kullanarak Eş. (2.9) ve Eş. (2.10) ile hesaplanır. İkinci adımda, $\mathbf{q}_i$ regresyon katsayıları, bağımlı $\mathbf{y}$ ve bağımsız $\mathbf{t}_i$ arasındaki basit bir regresyondan hesaplanır. Böylece, bileşenleri hesaplamadan sadece doğrudan $\mathbf{W}_k$ ağırlık matrisi hesaplanarak PLSR kestirimleri elde edilir [10]. Ancak, Eş. (2.11)'de gösterildiği üzere bu iki adım sadece gözlemlerin kovaryans matrisine dayalı olduğundan ve literatürde Eş. (2.9), Eş. (2.10) ve Eş. (2.11)'in üçü birlikte PLS1 algoritmasının seçenek bir tanımı olarak kullanıldığından, kovaryans matrisi sağlamlaştırıldığında da sürecin sağlamlaştırılacağı düşünülür. Bu yaklaşımdan yola çıkılarak, bir tane bağımlı değişken olduğu model için sadece Eş. (2.1)'deki kovaryans matrisini sağlam bir yöntem ile kestirerek ve daha sonra, Eş. (2.9) ve Eş. (2.10)'u kullanarak sağlam ağırlıklar elde edip, bu ağırlıkları Eş. (2.11)'de kullanıp sağlam PLSR regresyon katsayılarının elde edildiği sağlam PLSR yöntemleri önerilmiştir. Bu yöntemlerden Gil ve Romera [9] tarafından

önerilen PLS-SD, Kondylis ve Hadi [19] tarafından önerilen BACON-PLSR (BPLSR) ve González vd. [10] tarafından önerilen PLS-KurSD hakkında, bir sonraki alt bölümlerde detaylı olarak bilgi verilecektir. .

3.4.1. Stahel-Donoho Kestiricisine Dayalı Sağlam PLSR Yöntemi (PLS-SD)

Gil ve Romera [9], **x** değişkenlerinin örneklem kovaryans matrisi ile **x** ve **y** değişkenleri arasındaki kovaryans matrisini Stahel-Donoho Kestiricisi (Stahel-Donoho Estimator/SDE) ile sağlamlaştırarak, sağlam bir PLS1 yöntemi elde etmiştir. Bu yönteme ilişkin tüm PLSR sürecinin sağlamlık özellikleri bilinmemektedir. Sadece, doğal olarak IF'sinin SDE'nin IF'sine benzer olması beklenir. Bu yöntemin istatistiksel etkinliği ve BDP'si araştırılmamıştır. Yöntemin temel dezavantajı, sadece n>p olan veriye uyması ve açık bir algoritmanın elde edilememesidir [7, 9, 15].

Çok değişkenli konum ve kovaryans için geliştirilen Stahel-Donoho kestiricisinin adı, Stahel (1981) ve Donoho (1982)'nun bağımsız çalışmalarından ortaya çıkmıştır. Yöntemin temel fikri, çok değişkenli konum ve kovaryansın klasik kestiriminde aykırı gözlemlerin ağırlığını azaltmaktan ortaya çıkmıştır. Çok değişkenli aykırı değerler, çok değişkenli uzayda saklanabilir. Çok değişkenli aykırı değerleri bulmak, bir anlamda çok değişkenli konum ve kovaryans kestirimi ile ilişkilidir. Çünkü güvenilir kestirimler elde edildikten sonra, Mahalanobis uzaklıkları hesaplanabilir ve büyük Mahalanobis uzaklıklarına sahip gözlemler potansiyel çok değişkenli aykırı değerler olarak düşünülebilir. SDE, çok değişkenli aykırı değerleri çok basit bir yöntem ile belirler. Bu yöntemde her bir gözlem bir boyutlu uzaya yansıtılır ve aykırılığın bir ölçüsü hesaplanır. Çok değişkenli uzaydan bir boyutlu uzaya sonsuz sayıda fazla mümkün yansıma yönü olduğu için, her bir gözlem için aykırılığın sonsuz sayıda ölçüsü elde edilir. Bu nedenle amaç, mümkün tüm yansıma yönlerinden supremumu belirlemektir [7].

Stahel-Donoho yöntemi ile elde edilen konum ve kovaryans kestiricileri, $\mathbf{z}_i' = (y_i, \mathbf{x}_i)', i = 1, \cdots, n$ için sırasıyla Eş. (3.6) ve Eş. (3.7)'deki gibi tanımlanır [20].

$$\hat{\boldsymbol{\mu}}_{SDE}(\mathbf{Z}) = \frac{\sum_{i=1}^{n} d_i \mathbf{z}_i}{\sum_{i=1}^{n} d_i} \tag{3.6}$$

$$\hat{\boldsymbol{\Sigma}}_{SDE}(\mathbf{Z}) = \frac{\sum_{i=1}^{n} d_i (\mathbf{z}_i - \hat{\boldsymbol{\mu}}_{SDE}(\mathbf{Z}))(\mathbf{z}_i - \hat{\boldsymbol{\mu}}_{SDE}(\mathbf{Z}))'}{\sum_{i=1}^{n} d_i} \tag{3.7}$$

Bu eşitliklerdeki $d_1, \cdots, d_n$ parametreleri, verideki aykırı değerlerin ağırlığını azaltmak için hesaplanan ağırlıklardır. $d(.)$ ağırlık fonksiyonunu göstermek üzere, ağırlıklar $d_i = d(r(\mathbf{z}_i, \mathbf{Z}))$ şeklinde hesaplanır. Buradaki $r(.)$ ise, Eş. (3.8)'deki gibi elde edilir [20].

$$r(\mathbf{z}_i, \mathbf{Z}) = \sup_{\|\delta\|=1} \left\{ \frac{|\boldsymbol{\delta}'\mathbf{z}_i - \mu(\boldsymbol{\delta}'\mathbf{Z})|}{\sigma(\boldsymbol{\delta}'\mathbf{Z})} \right\} \tag{3.8}$$

Eş. (3.8)'deki δ, gözlemlerin üzerine yansıtıldığı p+q boyutlu bir birim (unity) vektördür. Buradaki q, bağımlı değişken sayısını gösterir. δ'nin doğru seçimi aykırı değerleri ortaya çıkarmadaki başarıyı belirler. Ancak, δ'nin belirlenmesi zordur ve hesaplanmasını kolaylaştırmak literatürdeki çalışmalarda tartışılmıştır. $\boldsymbol{\delta}'\mathbf{z}_i$, kordinat sisteminin merkezi ve $\mathbf{z}_i$'nin δ'nin üzerine yansıması arasındaki uzaklıktır. $\mu(\boldsymbol{\delta}'\mathbf{z})$ ve $\sigma(\boldsymbol{\delta}'\mathbf{z})$ sırasıyla, $\mathbf{Z}$'deki gözlemlerin δ'nin üzerine yansımalarının ortanca ve ortanca mutlak sapmalarıdır. $r(\mathbf{z}_i, \mathbf{Z})$'nin mümkün en büyük değeri, o gözlem için 'aykırılık' olarak adlandırılır [20].

Stahel-Donoho kestiricisinin dezavantajı, yakınsak algoritmaların önerilmesine karşın hesaplanmasının uzun süre almasıdır. Bu nedenle de, her zaman tam biçimi ile hesaplanamamaktadır. Genel olarak, alt örnekleme süreçleri ile birlikte yakınsama yöntemleri kullanılır. Gil ve Romera [9], aşağıdaki biçime sahip seçenek bir alt örnekleme yöntemi kullanmıştır [9, 20].

Adım 1: Rasgele olarak her biri p+q+2 tane gözlem içeren, N tane alt örneklem seçilir.

Adım 2: Her bir alt örneklemden, en büyük Mahalanobis uzaklığına sahip gözlem çıkarılır.

Adım 3: Bir alt kümede geriye kalan p+q+1 gözlemden, p+q tane gözlemin p+q+1 farklı kombinasyonu vardır. Her bir p+q gözlemli alt küme tarafından karışlanan hiper düzleme dik olan, bir δ vektörü elde edilebilir. Bu nedenle, her bir alt küme için p+q+1 tane δ vektörü oluşturulur.

Adım 4: Eş. (3.8)'i kullanarak her bir gözleme ilişkin 'aykırılık' değerini hesaplamak için, her bir gözleme her bir δ vektörü uygulanır.

Gil ve Romera [9], Eş. (3.7)'deki sağlam kovaryans kestirimini PLS1 algoritmasının Alt Bölüm 2.2.2'deki tanımındaki Eş. (2.9), Eş. (2.10) ve Eş. (2.11) eşitliklerinde kullanarak, PLS-SD olarak isimlendirilen sağlam PLSR algoritmasını önermiştir. Alt örnekleme yönteminden de anlaşılacağı üzere, yöntemin temel dezavantajı sadece gözlem sayısının değişken sayısından fazla olduğu veri kümelerine uygulanabilmesidir [7].

3.4.2. BACON Algoritmasına Dayalı Sağlam Kısmi En Küçük Kareler Regresyon Yöntemi

Billor vd. (2000) tarafından önerilen 'Parçalı Uyarlanabilir Hesaplama Yönünden Etkin Aykırı Gözlem Belirleyicisi (Blocked Adaptive Computationally Efficient Outlier Nominators/BACON)' algoritması, hem çok değişkenli veriye hem de regresyon problemlerine uygulanabilir. BACON algoritması çok değişkenli veri için uygulandığında, aykırı değerleri belirler ve **X** bağımsız değişkenlerinin varyans-kovaryans matrisinin sağlam kestirimlerini verir. Algoritma regresyon için kullanıldığında ise, çok değişkenli durumda elde edilen sağlam kestirimleri kullanır ve bağımlı bir **y** değişkeni için $\hat{\boldsymbol{\beta}}$ regresyon parametre vektörünü sağlam bir şekilde kestirir. BACON algoritması, 'm' gözlemden oluşan ve büyük olasılıkla aykırı değer içermeyen, bir temel başlangıç alt kümesi oluşturarak başlar. Buradaki m gözlem sayısı, verinin analizini yapan kişi tarafından belirlenir. Daha sonra başlangıç temel alt kümesine uyan gözlemler, bu alt kümeye eklenir. Eğer tüm gözlemler temel alt kümeye eklenirse, veride aykırı değerlerin olmadığı kesinleşir. Ancak, temel altkümeye uymayan gözlemler çok değişkenli aykırı değerler olarak adlandırılır [19].

Kondylis ve Hadi [19], Billor vd. (2000) tarafından önerilen BACON algoritmasını PLS1 algoritmasının Alt Bölüm 2.2.2'deki tanımındaki Eş. (2.9), Eş. (2.10) ve Eş. (2.11) eşitliklerinde kullanarak tek bir bağımlı değişkenin olduğu model için PLSR yöntemini sağlamlaştırmıştır. Kondylis ve Hadi [19] tarafından önerilen algoritma, 'BACON-PLSR (BPLSR)' olarak isimlendirilmiştir. Burada 'B' harfi, BACON varyans-kovaryans matrisinin kullanıldığını vurgular. Kondylis ve Hadi [19], iki nedenle BACON kestirimlerini kullanmıştır: Birincisi, BACON'un hesaplama açısından çok etkin olması ve ikincisi, aykırı değerleri belirlemede aşırı derecede etkin olmasıdır. Örneğin, BACON algoritması 3-5 yinelemede yakınsar ve aykırı değerlerden kaynaklanan % 20'ye kadar olan bozulmayı hoş görür. Hem gerçek veri kümesi üzerindeki uygulama hem de benzetim çalışmasından elde edilen sonuçlar, BPLSR algoritmasının aykırı değerlere karşı dirençli olduğunu gösterir. Regresyon kestirimleri, en azından aykırı değerlerin varlığı ile makul düzeylerde bozulmuş veri kümelerinde, aykırı değerlerden fazla etkilenmemektedir. Sadece

aykırı değerler tarafından yüksek bozulma düzeylerinde ve veri kümelerinin göreceli olarak yüksek boyutlularında BPLSR'nin kestirim başarısı düşer [19].

3.4.3. Yansımaların Basıklık Katsayısına ve Stahel-Donoho Kestiricisine Dayalı Sağlam PLSR Yöntemi (PLS-KurSD)

Pena ve Prieto [25], klasik SDE yönteminden daha etkin bir şekilde elde edilen 'rasgele yönler' ile yansıtılan verinin basıklık katsayısını en büyük ve en küçük yaparak elde edilen 'özel yönlerin' bir birleşimini kullanarak, çok değişkenli veri kümesinde aykırı değerleri araştıran bir yöntem önermiştir. Bu 'rasgele yönler' ve 'özel yönler' olmak üzere iki tür yönlerin birleşimi, faydalı teorik özellikleri olan ve iyi bir performansı olan bir sürecin elde edilmesini sağlamıştır. Böylece bu yeni süreç, SDE'nin iyi teorik özelliklerine sahipken, aynı zamanda basıklık sürecinin büyük kaldıraç kümelenmiş aykırı değerleri (high leverage concentrated outliers) bulmak için sahip olduğu iyi özelliklere de sahip olur [10]. Pena ve Prieto [25], bu yöntemi çok değişkenli aykırı değerleri ortaya çıkarmak için önermiştir. González vd. [10] ise bu yöntemi, PLS-SD algoritmasına benzer biçimde Alt Bölüm 2.2.2'de verilen PLS1 algoritmasında kullanılan $\mathbf{S}_z$ kovaryans matrisini sağlam bir biçimde kestirmek amacıyla kullanmış ve böylece, sağlam PLSR kestirimleri elde etmiştir. González vd. [10], bu sağlam PLSR algoritmasını 'PLS-KurSD' olarak isimlendirmiştir.

González vd. [10] tarafından önerilen algoritma, çok değişkenli aykırı değerlerden temizlenmiş bir kovaryans matrisini elde etmek için tasarlanmıştır ve aşağıda anlatılan üç adımı kullanarak işler. Genellemeyi kaybetmeden, özgün verinin sıfır ortalamaya ve $\mathbf{S}_z$ kovaryansına sahip olduğu varsayılır. Gözlemler, Eş. (3.9)'u kullanarak dönüştürülür [10].

$$\tilde{\mathbf{z}}_i = \mathbf{S}_z^{-1/2}\mathbf{z}_i, i = 1,\cdots,n \tag{3.9}$$

Algoritma, üç adıma sahiptir. İlk adımda, yansımaların (projections) basıklık katsayısını en büyük ve en küçük yaparak iki özel yön üretilir ve bu yönlerde,

aykırı değerler için tek değişkenli araştırma yapılır. İkinci adımda, Pena ve Prieto [25] çalışmasında bahsedilen tabakalı örneklemeye ilişkin süreç takip edilerek rasgele yönler üretilir ve yeniden aykırı değerler tespit edilir. Üçüncü adımda, tüm şüpheli gözlemler örneklemden geçici olarak silinir ve geriye kalan verinin ortalaması ile kovaryans matrisi hesaplanır. Daha sonra, Mahalanobis uzaklığını kullanarak tüm şüpheli gözlemler kontrol edilir. Aykırı gözlemler olarak belirlenen gözlemler örneklemden silinir ve daha fazla aykırı değer bulunmayana kadar, sürecin üç adımı yeni temizlenmiş örnekleme yeniden uygulanır. Bu adımların detayları, aşağıda anlatılacaktır [10].

Adım 1: Yansımanın basıklık katsayısını, en büyük ve en küçük yapan yönler ve aynı zamanda bu iki yönde, verinin normalleştirilmiş tek değişkenli uzaklıkları hesaplanır. Yansımanın basıklık katsayısını en büyük yapan yön, $\boldsymbol{\delta}'\boldsymbol{\delta} = 1$ kısıtı altında Eş. (3.10)'daki problemin çözümü olarak elde edilir [10].

$$\boldsymbol{\delta}_1 = \arg\max_{\boldsymbol{\delta}} \frac{1}{n}\sum_{i=1}^{n}\left(\boldsymbol{\delta}'\tilde{\mathbf{z}}_i\right)^4 \tag{3.10}$$

Aynı süreç, basıklık katsayısının en küçük yapan yönün hesaplanmasına uygulanmıştır. $\boldsymbol{\delta}_2$, bu ikinci yön ve $p_i^{(j)} = \boldsymbol{\delta}'\tilde{\mathbf{z}}_i, j = 1,2$ bu iki yöne yansıyan değerler olsun. Bu yansımış değerler için $r_i^{(j)}$ normalleştirilmiş tek değişkenli uzaklıklar, Eş. (3.11)'deki gibi hesaplanır. Bu eşitlikteki MAD ise, Eş. (3.12)'deki gibi hesaplanır [10].

$$r_i^{(j)} = \frac{1}{\boldsymbol{\beta}_p}\frac{\left|p_i^{(j)} - \text{ortanca}_i\left(p_i^{(j)}\right)\right|}{MAD_i\left(p_i^{(j)}\right)},\ j = 1,2 \tag{3.11}$$

$$MAD_i\left(p_i^{(j)}\right) = \text{ortanca}_i\left|p_i^{(j)} - \text{ortanca}_i\left(p_i^{(j)}\right)\right| \tag{3.12}$$

Eş. (3.11)'deki $\boldsymbol{\beta}_p$ ise, p boyutuna dayalı önceden belirlenmiş referans değeridir ve 0.05'e eşit I. hatayı elde etmek için Monte Carlo ile elde edilir [10]. $\boldsymbol{\beta}_p$, basıklık

katsayısını en küçük ya da en büyük yapan yönler üzerine gözlemlerin yansımalarından aykırı değerleri belirlemek için bir eşik değeri gibi davranır [25]. Pena ve Prieto [25] çalışmasında gösterildiği üzere bu adım, $r_i^{(j)} > 1$ olan gözlemlerden oluşan kümelerin oluşturduğu aykırı değerleri belirleyecektir. Eğer aykırı değerler grubunun boyutu küçük ise (kabaca % 20'den daha küçük ise) aykırı değerleri belirlemek için faydalı yön $\boldsymbol{\delta}_1$ olacaktır. Ancak boyut büyük olur ise, faydalı yön $\boldsymbol{\delta}_2$ olacaktır [10].

Adım 2: Bir tabakalı örnekleme sürecinden, rasgele yönler hesaplanır. Daha sonra, bu yönlerdeki aykırı değerler için araştırma yapılır. Pena ve Prieto [25] çalışmasında önerilen süreç tarafından üretilen bu rasgele yönler, aykırı değerleri belirlemek için kullanılan standart SDE yönteminden daha etkindir. Her bir yön, iki aşamada üretilmiştir. İlk aşamada, örneklemden iki gözlem rasgele seçilir, bu iki gözlem tarafından tanımlanan yön hesaplanır ve daha sonra, gözlemler bu yön üzerine yansıtılır. Bu işlem, $l = 1, \cdots, L$ için tekrar edilir. L, rasgele yönlerin sayısıdır. İkinci aşama, her bir l için K tane tabakalı örneklemi şu şekilde oluşturur: Yansımalar sıralanır ve n/K boyutlu K tane aralığa bölünür. K, her bir rasgele yön için önceden belirlenmiş aralık sayısıdır $1 \le k \le K$ olmak üzere, bu k aralıktan her birinden p gözlemden oluşan bir alt örneklem yerine koymadan seçilir ve bu p tane gözlem tarafından üretilip, hiper düzleme (hyperplane) dik olan $\tilde{\boldsymbol{\delta}}_j$ yönü hesaplanır. Adım 1'de olduğu gibi $\tilde{\boldsymbol{\delta}}_j$ yönü, aykırı değerleri araştırmak için kullanılır. Bu yöndeki $\tilde{p}_i^{(j)} = \tilde{\boldsymbol{\delta}}_j \tilde{\mathbf{z}}_i$ yansımaları, Eş. (3.13)'teki normalleştirilmiş tek değişkenli $\tilde{r}_i^{(j)}$ uzaklıklarını verir [10].

$$\tilde{r}_i^{(j)} = \frac{1}{\boldsymbol{\beta}_p} \frac{\left| \tilde{p}_i^{(j)} - \text{ortanca}_i\left(\tilde{p}_i^{(j)}\right) \right|}{\text{MAD}_i\left(\tilde{p}_i^{(j)}\right)}, \; j = 1, \cdots, LK \tag{3.13}$$

Adım 3: Her bir i gözlemine ilişkin normalleştirilmiş aykırılık ölçüsü, Eş. (3.14)'deki gibi elde edilir [10].

$$r_i = \max\left\{r_i^{(1)}, r_i^{(2)}, \tilde{r}_i^{(1)}, \cdots, \tilde{r}_i^{(LK)}\right\} \tag{3.14}$$

$r_i > 1$ olan gözlemler aykırı değerler olarak etiketlenir ve eğer aykırı gözlem sayısı $n - [(n+1+p)/2]$'den daha az ise örneklemden çıkartılır. Ancak, aykırı değer sayısı bu değerden fazla ise sadece en büyük r_i değerlerine sahip olan $n - [(n+1+p)/2]$ tane gözlem aykırı değer olarak etiketlenir [10].

Son olarak U, aykırı değer olarak etiketlenmeyen tüm gözlemlerin bir kümesini göstersin. Aykırı olmayan gözlemlere ilişkin konum ve kovaryans sırasıyla, Eş. (3.15) ve Eş. (3.16)'daki gibi hesaplanır [10].

$$\tilde{\mathbf{m}} = \frac{1}{|U|}\sum_{i \in U} \mathbf{z}_i \tag{3.15}$$

$$\tilde{\mathbf{S}}_Z = \frac{1}{|U|-1}\sum_{i \in U} (\mathbf{z}_i - \tilde{\mathbf{m}})(\mathbf{z}_i - \tilde{\mathbf{m}})' \tag{3.16}$$

Algoritma, aykırı değerler olarak etiketlenen özgün gözlemlerin aykırı olmayan 'iyi gözlemlere' göre Mahalanobis uzaklığını Eş. (3.17)'deki gibi hesaplar [10].

$$\tilde{v}_i = (\mathbf{z}_i - \tilde{\mathbf{m}})' \tilde{\mathbf{S}}_z^{-1} (\mathbf{z}_i - \tilde{\mathbf{m}}), \quad \forall i \notin U \tag{3.17}$$

$p' = p + 1$ olmak üzere, $i \notin U$ olan gözlemlerden $\tilde{v}_i < \chi^2_{p'-1,0.99}$ şeklinde Mahalanobis uzaklıkları eşik değerinden küçük olanların aykırı değer olmadığı düşünülür ve bu gözlemler de U kümesinde yer alır [10].

Yukarıda anlatılan üç adım, daha fazla aykırı değer bulunmayana kadar (ya da U, tüm gözlemlerin kümesi haline gelene kadar) tekrarlanır. Pena ve Prieto [25] çalışmasındaki gibi, en son elde edilen kovaryans matrisinin tutarlılık için ölçeklendirilmesi gerekmez. Çünkü elde edilen PLSR katsayıları ve yönleri, ölçek değişikliklerine karşı sabittir. Bu algoritma, belirli sayıda parametre içerir.

Uygulamada bu parametrelere atanan değerler, yeterli teorik ve etkinlik özelliklerini sağlamak için Pena ve Prieto [25] çalışmasında tavsiye edildiği gibi seçilir. $\boldsymbol{\beta}_p$ parametresi, I. tür hataların makul bir değerini sağlamak için seçilir ve örneklem uzayı boyutu p'ye dayalıdır. Pena ve Prieto [25] çalışmasından alınan Çizelge 3.1, birkaç örneklem uzay boyutu için kullanılan değerleri gösterir. Diğer boyutlar için değerler, $\log\boldsymbol{\beta}_p$'yi doğrusal olarak logp'ye interpole ederek elde edilir. Adım 2'deki aralık sayısı K, p'ye dayalı olarak 3 ya da 5 olarak seçilir ve L, 10p'ye eşittir [10].

Çizelge 3.1. Adım 1 ve Adım 2 için tek değişkenli yansımalar için kesim değerleri.

Örneklem uzayı boyutu p	5	10	20
Kesim değeri β_p	3.46	3.86	4.67

Sonuç olarak, önerilen sağlam PLSR algoritması PLS-KurSD sağlam kovaryans matrisi $\tilde{\mathbf{S}}_z$'ye dayalıdır. Bu sağlam PLSR algoritmasında $\mathbf{w}_1$, $\tilde{\mathbf{s}}_{y,x}$ vektörünün normalleştirilmesi olarak düşünülür ve böylece, $\tilde{\mathbf{S}}_z$'nin ilk kolonu ilk elemanı oluşturur. Aynı zamanda $\mathbf{w}_i$'nin sonraki değerleri ise, Eş. (3.18)'i kullanarak sağlam bir biçimde hesaplanır [10].

$$\mathbf{w}_{j+1} \alpha \tilde{\mathbf{s}}_{y,x} - \tilde{\mathbf{S}}_x \mathbf{W}_j \left(\mathbf{W}_j' \tilde{\mathbf{S}}_x \mathbf{W}_j\right)^{-1} \mathbf{W}_j' \tilde{\mathbf{s}}_{y,x}, \quad 1 \le j < a \tag{3.18}$$

Bu önerilen sağlam PLSR algoritmasında, her bir gözlem için sağlam Mahalanobis uzaklığını hesaplamak ve veriyi aykırı değerlerden temizlemek için, aykırılığın önerilen ölçüsü kullanılır [10].

Buraya kadar olan alt bölümlerde, klasik PLS1 ve PLS2 algoritmalarında yapılan değişikler ile elde edilen sağlam PLSR yöntemleri hakkında bilgiler verilmiştir. Alt Bölüm 3.5'te ise, SIMPLS algoritmasında yapılan değişiklikler ile elde edilen sağlam PLSR yöntemleri incelenecektir.

3.5. SIMPLS Algoritmasının Sağlamlaştırılmasıyla Elde Edilen Sağlam PLSR Yöntemleri

SIMPLS algoritması, PLS1 ve PLS2'den daha hızlı olduğu ve sonuçlarının yorumlanması daha kolay olduğundan, PLSR kestirimlerini elde etmek için en sık kullanılan PLSR algoritmasıdır. Ancak, SIMPLS algoritması bağımlı ve bağımsız değişkenler arasındaki varyans-kovaryans matrisine ve LS regresyonuna dayalı olduğundan, sonuçlar aykırı gözlemlerden etkilenir. Bu nedenle, Hubert ve Vanden Branden [15], sırasıyla çok değişkenli regresyon yöntemi olan MCD regresyon ve ROBPCA regresyon yöntemlerini kullanarak algoritmanın sağlamlaştırılmış biçimleri olan RSIMCD ve RSIMPLS algoritmalarını geliştirmiştir [15].

RSIMCD ve RSIMPLS algoritmalarını anlatmadan önce, p>n olan büyük boyutlu veri kümelerinde sağlam kovaryans kestiriminin nasıl yapılacağı ve ROBPCA yöntemini kullanarak sağlam bileşenlerin elde edilmesi anlatılmalıdır. Buna göre, $\mathbf{Z}_{n,m} = (\mathbf{X}_{n,p}, \mathbf{Y}_{n,q})$, birleştirilmiş kümeyi göstersin. SIMPLS algoritmasındaki $\tilde{\mathbf{t}}_i$ bileşenleri, $\mathbf{S}_{xy}$ ve $\mathbf{S}_x$ kovaryans matrislerinden hesaplanır. Hem bu iki kovaryans hem de SIMPLS algoritmasının ikinci aşamasında uygulanan LS regresyonu, aykırı değerlere karşı çok hassastır. Hubert ve Vanden Branden [15], $\mathbf{S}_{xy}$ ve $\mathbf{S}_x$'i sırasıyla, $\mathbf{\Sigma}_x$ ve $\mathbf{\Sigma}_{xy}$'nin sağlamlaştırılmış kestirimleri ile yer değiştirerek ve MLR yerine sağlam bir regresyon yöntemi uygulayarak, SIMPLS algoritmasının iki tane sağlamlaştırılmış biçimi olan RSIMPLS ve RSIMCD algoritmalarını önermiştir. Bu sağlam kovaryans matrislerini elde etmek için, Huber vd. (2003) tarafından önerilen ve sağlam bir Temel Bileşenler Analizi (Principal Component Analysis/PCA) yöntemi olan ROBPCA kullanılmıştır RSIMPLS ve RSIMCD algoritmalarının her ikisi de, bir ya da birden çok bağımlı değişkenin olduğu model için de kullanılabilir. Ancak, çalışmada sadece çok değişkenli durum için gösterimlere yer verilmiştir [15].

MCD kestiricisinin temeli, [n/2]<h<n olmak üzere h gözlemin örneklem kovaryans matrisinin determinantını en küçük yapmaktır. MCD kestiricisi sadece, n>p olur ise

uygulanabilir; çünkü p>n olur ise aynı zamanda p>h olur ve herhangi bir h gözlemin kovaryans matrisi her zaman tekil olacağından elde edilen determinant da sıfır olacaktır. Bu nedenle, h gözlemin her bir alt kümesi, mümkün en küçük determinantı verecektir ve tek bir sonuç elde edilemeyecektir [7, 15]. Bu nedenle, m=p+q<n olmak üzere verinin boyutu küçük olur ise SIMPLS algoritmasındaki kovaryans matrislerini kestirmek için konum ve kovaryansın sağlam kestiricilerinden MCD kestiricisi kullanılabilir. Böylece küçük boyutlu örneklemlerde MCD kestiricisi kullanılarak $\mathbf{z}_i$'nin ortalaması $\bar{\mathbf{z}}_h$ ve varyansı, en uygun h altkümenin kovaryans matrisi $\mathbf{S}_h$'ın bir tutarlılık faktörü ile çarpılması ile kestirilir. İstatistiksel etkinliği arttırmak için, yeniden ağırlıklandırma adımı eklenebilir. Eğer bir gözlemin sağlam karesel uzaklığı $\left(\mathbf{z}_i - \bar{\mathbf{z}}_h\right)' \mathbf{S}_h^{-1} \left(\mathbf{z}_i - \bar{\mathbf{z}}_h\right)$, $\chi^2_{m,0.975}$'yi aşar ise, o gözlem sıfır ağırlık alır. Buna göre, Yeniden Ağırlıklandırılmış En Küçük Kovaryans Determinantı (Reweighted Minimum Covariance Determinant /RMCD) kestiricisi, bir ağırlığına sahip gözlemlerin klasik ortalama ve kovaryans matrisi olarak tanımlanır. Ancak, MCD kestiricisi m>n olan yüksek boyutlu veri kümelerine uygulanamadığı için, projeksiyon izleme (projection pursuit) algoritmaları geliştirilmiştir. ROBPCA iki yaklaşımı birleştirir: Donoho (1982) ve Stahel (1981)'deki projeksiyon izleme fikirlerini kullanarak her bir gözlemin aykırılığını hesaplar ve en küçük aykırılığa sahip h tane gözlemin kovaryans matrisini inceler. Daha sonra veri, bu kovaryans matrisinin k_0<m tane baskın özvektörü tarafından taranan K_0 alt uzayına yansıtılır. Daha sonra, bu küçük boyutlu uzayda verinin ortalamasını ve varyansını kestirmek için, MCD yöntemi uygulanır. Son olarak, bu kestirimler özgün uzaya yeniden dönüştürülür ve $\mathbf{Z}_{n,m}$'nin $\hat{\boldsymbol{\mu}}_z$'si ve $\hat{\boldsymbol{\Sigma}}_z$'sinin sağlam bir kestirimi elde edilir. $\mathbf{P}^z_{m,k_0}$, sağlam Z özvektörlerini ve köşegen(L_{k_0,k_0}), sağlam Z özdeğerlerini göstermek üzere, varyans-kovaryans matrisi $\hat{\boldsymbol{\Sigma}}_z = \mathbf{P}^z \mathbf{L}^z \left(\mathbf{P}^z\right)'$ şeklinde ayrıştırılabilir. $\mathbf{L}^z$ köşegen matrisi, azalan sırada $\hat{\boldsymbol{\Sigma}}_z$'nin k_0 en büyük özdeğerini içermektedir. **Z** bileşenleri ise $\mathbf{T}^z = \left(\mathbf{Z} - \mathbf{1}_n \hat{\boldsymbol{\mu}}'_z\right)\mathbf{P}^z$ şeklinde elde edilir [15].

Sağlam PLSR analizi yapmak için, ilk önce sağlam bileşenler elde edilmelidir. Sağlam bileşenleri elde etmek için ise, ilk önce $\mathbf{Z}_{n,m} = \left(\mathbf{X}_{n,p}, \mathbf{Y}_{n,q}\right)$ üzerinde ROBPCA uygulanır. ROBPCA yönteminin uygulanmasının sonucunda, **Z**'nin

sağlam kestiricileri sırasıyla $\hat{\boldsymbol{\mu}}_z = \left(\hat{\boldsymbol{\mu}}_x', \hat{\boldsymbol{\mu}}_y'\right)'$ ve $\hat{\boldsymbol{\Sigma}}_z$ şeklinde elde edilir. $\hat{\boldsymbol{\Sigma}}_z$, Eş. (3.19)'daki gibi ayrıştırılır [15].

$$\hat{\boldsymbol{\Sigma}}_z = \begin{pmatrix} \hat{\boldsymbol{\Sigma}}_x & \hat{\boldsymbol{\Sigma}}_{xy} \\ \hat{\boldsymbol{\Sigma}}_{yx} & \hat{\boldsymbol{\Sigma}}_y \end{pmatrix} \tag{3.19}$$

$\boldsymbol{\Sigma}_{xy}$ kovaryans matrisi, $\hat{\boldsymbol{\Sigma}}_{xy}$ ile kestirilir ve PLSR ağırlık vektörleri $\mathbf{r}_a$'lar, SIMPLS algoritmasındaki gibi hesaplanır. Ancak SIMPLS algoritmasından farklı olarak, $\mathbf{S}_{xy}$ yerine $\hat{\boldsymbol{\Sigma}}_{xy}$ ile başlanır. $\mathbf{x}$ yükleri olan $\mathbf{p}_j$, Eş. (2.16)'ya benzer olarak $\mathbf{p}_j = \left(\mathbf{r}_j' \hat{\boldsymbol{\Sigma}}_x \mathbf{r}_j\right)^{-1} \hat{\boldsymbol{\Sigma}}_x \mathbf{r}_j$ şeklinde tanımlanır. Daha sonra $\hat{\boldsymbol{\Sigma}}_{\mathbf{xy}}^{\mathbf{a}}$ kovaryans matrisi, SIMPLS algoritmasındaki gibi indirgenir. Her bir adımda sağlam bileşenler Eş. (3.20)'deki gibi hesaplanır [15].

$$t_{ia} = \breve{\mathbf{x}}_i' \mathbf{r}_a = \left(\mathbf{x}_i - \hat{\boldsymbol{\mu}}_x\right)' \mathbf{r}_a \tag{3.20}$$

Bu eşitlikteki $\breve{\mathbf{x}}_i$'ler, sağlam bir şekilde merkezileştirilmiş gözlemlerdir. Sağlam bileşenler elde edildikten sonra, sağlam bir regresyon yapılabilir. Sağlam $\mathbf{t}_i$ bileşenlerine dayalı regresyon modeli, Eş. (3.21)'deki gibi yazılır [15].

$$\mathbf{y}_i = \boldsymbol{\alpha}_0 + \mathbf{A}_{q,k}' \mathbf{t}_i + \breve{\mathbf{f}}_i \tag{3.21}$$

3.5.1. MCD Regresyonunu Kullanılarak Elde Edilen RSIMCD Algoritması

RSIMCD algoritmasında Eş. (3.21)'de verilen regresyon modelinin klasik MLR kestirimleri, $(\mathbf{t}, \mathbf{y})$ birleştirilmiş değişkenlerin ortalaması ve kovaryansı cinsinden Eş. (3.22)'deki gibi yazılır [15].

$$\boldsymbol{\mu} = \begin{pmatrix} \boldsymbol{\mu}_t \\ \boldsymbol{\mu}_y \end{pmatrix}, \quad \boldsymbol{\Sigma} = \begin{pmatrix} \boldsymbol{\Sigma}_t & \boldsymbol{\Sigma}_{ty} \\ \boldsymbol{\Sigma}_{yt} & \boldsymbol{\Sigma}_y \end{pmatrix} \tag{3.22}$$

$\boldsymbol{\mu}$ ortalaması, $\left(\bar{\mathbf{t}}', \bar{\mathbf{y}}'\right)'$ örneklem ortalaması ve $\boldsymbol{\Sigma}$ kovaryansı, $(\mathbf{t}, \mathbf{y})$'nin örneklem kovaryans matrisi ile kestirilir, Eş. (2.20)'deki $\tilde{\bar{\mathbf{t}}}$ ise $\bar{\mathbf{t}}$ ile yer değiştirilir ise klasik kestirimler Eş. (2.19)-(2.21) eşitliklerini sağlar. $(\mathbf{t}, \mathbf{y})$'nin klasik ortalama ve kovaryans matrisini, RMCD kestirimleri ile yer değiştirerek sağlam regresyon kestirimleri elde edilir. Ayrıca, etkinliği arttırmak için bu başlangıç regresyon kestirimlerinin yeniden ağırlıklandırıması tavsiye edilir. $r_{i(k)}$, k bileşen için hesaplanan başlangıç kestirimlerine ilişkin i. gözlemin artığı olsun. Eğer $\hat{\boldsymbol{\Sigma}}_{\tilde{f}}$, hataların kovaryansı için başlangıç kestirimi ise, artıkların sağlam uzaklığı Eş. (3.23)'deki gibi tanımlanır. $c_{i(k)}$ ağırlıkları ise, I gösterge fonksiyonu olmak üzere Eş. (3.24)'deki gibi hesaplanır [15].

$$RD_{i(k)} = \left(r'_{i(k)} \hat{\boldsymbol{\Sigma}}_{\tilde{f}}^{-1} r_{i(k)}\right)^{1/2} \tag{3.23}$$

$$c_{i(k)} = I\left(RD_{i(k)}^2 \le \chi^2_{q,0.975}\right) \tag{3.24}$$

En son regresyon kestirimleri ise, sadece $c_{i(k)}$ ağırlıkları bire eşit olan gözlemler için klasik MLR'deki gibi hesaplanır. $RD_{i(k)}$ sağlam artık uzaklıkları ise, Eş. (3.23)'deki gibi hesaplanır ve aynı zamanda $c_{i(k)}$ ağırlıkları da uyarlanır. Eş. (2.22)'deki asıl model için sağlam kestirimler ise, Eş. (3.25)'deki gibi verilir [15].

$$\hat{\mathbf{B}}_{p,q} = \mathbf{R}_{p,k} \hat{\mathbf{A}}_{k,q}, \quad \hat{\boldsymbol{\beta}}_0 = \hat{\boldsymbol{\alpha}}_0 - \hat{\mathbf{B}}'_{q,p} \hat{\boldsymbol{\mu}}_x, \quad \hat{\boldsymbol{\Sigma}}_e = \hat{\boldsymbol{\Sigma}}_{\tilde{f}} \tag{3.25}$$

RSIMCD'nin hesaplama zamanı, (**x**,**y**) değişkenleri üzerinde ROBPCA ve (**t**,**y**) değişkenleri üzerinde MCD uygulanarak belirlenmiştir [15].

3.5.2. ROBPCA Regresyonunu Kullanılarak Elde Edilen RSIMPLS Algoritması

Hubert ve Vanden Branden [15], $(\mathbf{t},\mathbf{y})$ üzerinde MCD hesaplamaktan kaçınan ikinci bir algoritma olarak RSIMPLS'yi önermiştir. MCD regresyon yöntemi, $(k+q)$ boyutlu $(\mathbf{t},\mathbf{y})$ uzayındaki aykırı olmayan gözlemlerin ortalaması **μ** ve kovaryansı **Σ**'nın sağlam kestirimlerini elde etmek için, RMCD kestiricisinin uygulanması ile başlar. Sağlam $\mathbf{t}_i$ bileşenlerini elde etmek için ilk önce $(\mathbf{x},\mathbf{y})$ değişkenlerine ROBPCA yöntemi uygulanır ve bu değişkenleri iyi temsil eden, k_0 boyutlu K_0 alt uzayı elde edilir. Bileşenler **x** değişkenlerine ilişkin en önemli bilgiyi özetlemek için oluşturulduğundan, k_0 boyutlu alt uzaya ilişkin aykırı değerlerin aynı zamanda, $(\mathbf{t},\mathbf{y})$ uzayında aykırı değer olduğu düşünülür. Bu nedenle, $(\mathbf{t},\mathbf{y})$ değişkenlerinin ortalaması ve kovaryansı sırasıyla, Eş. (3.26) ve Eş. (3.27)'deki gibi kestirilir [15].

$$\hat{\boldsymbol{\mu}} = \begin{pmatrix} \hat{\boldsymbol{\mu}}_t \\ \hat{\boldsymbol{\mu}}_y \end{pmatrix} = \sum_{i=1}^{n} w_i \begin{pmatrix} \mathbf{t}_i \\ \mathbf{y}_i \end{pmatrix} \Big/ \left(\sum_{i=1}^{n} w_i \right) \tag{3.26}$$

$$\hat{\boldsymbol{\Sigma}} = \begin{pmatrix} \hat{\boldsymbol{\Sigma}}_t & \hat{\boldsymbol{\Sigma}}_{ty} \\ \hat{\boldsymbol{\Sigma}}_{yt} & \hat{\boldsymbol{\Sigma}}_y \end{pmatrix} = \sum_{i=1}^{n} w_i \begin{pmatrix} \mathbf{t}_i \\ \mathbf{y}_i \end{pmatrix} \begin{pmatrix} \mathbf{t}_i' & \mathbf{y}_i' \end{pmatrix} \Big/ \left(\sum_{i=1}^{n} w_i - 1 \right) \tag{3.27}$$

Bu kestirimler, K_0 alt uzayında $(\mathbf{x}_i,\mathbf{y}_i)$ değerleri merkezden uzakta bulunmayan $(\mathbf{t}_i,\mathbf{y}_i)$'lerin, ağırlıklandırılmış ortalama ve kovaryans matrisidir. Burada $(\mathbf{x},\mathbf{y})$ üzerine ROBPCA uygulandığında i gözlemi aykırı değer olarak tanımlanmıyor ise ağırlıklar $w_i = 1$, aksi takdirde $w_i = 0$ olur [15].

ROBPCA uygulandığında, K_0 uzayının içinde merkezden uzakta kalan ya da K_0'dan uzakta kalanlar olmak üzere, iki tür aykırı değer tanımlanabilir. İlk aykırı değer türü, $D_{i(k_0)} = \sqrt{\left(\mathbf{t}_i^z\right)' \left(\mathbf{L}^z\right)^{-1} \mathbf{t}_i^z}$ sağlam uzaklıkları $\sqrt{\chi^2_{k_0,0.975}}$'yi aşan gözlemler olarak tanımlanabilir. İkinci tür aykırı değerlere karar vermek için, her bir gözlemin K_0 alt uzayına olan $OD_i = \left\| \left(\mathbf{z}_i - \hat{\boldsymbol{\mu}}_z\right) - \mathbf{P}^z \mathbf{t}_i^z \right\|$ dik uzaklığı incelenir. Merkezi limit

teoreminden dolayı, karesi alınmış bu dik uzaklıkların yaklaşık olarak normal dağıldığı kabul edilir. Bu nedenle, ortalamaları ve varyansları tek değişkenli MCD yöntemini kullanarak sırasıyla, $\hat{\mu}_{od^2}$ ve $\hat{\sigma}^2_{od^2}$ .olarak kestirilir. Eş. (3.28) geçerli olduğunda, $w_i = 0$ alınır [15].

$$OD_i > \sqrt{\hat{\mu}_{od^2} + \hat{\sigma}_{od^2} z_{0.975}} \tag{3.28}$$

Bir ağırlığına sahip gözlemler belirlendikten sonra, Eş. (3.26) ve Eş. (3.27)'yi kullanarak $\hat{\boldsymbol{\mu}}$ ve $\hat{\boldsymbol{\Sigma}}$ hesaplanır. Daha sonra, MCD regresyon yöntemindeki gibi devam edilir. Bu kestirimler Eş. (2.19)-(2.21)'deki eşitliklerde yerine koyulur, Eş. (3.23)'deki gibi artık uzaklıkları hesaplanır ve yeniden ağırlıklandırılmış MLR yapılır. Bu yeniden ağırlıklandırma adımının avantajı, aykırı değer olmayan $w_i = 0$ ağırlığına sahip gözlemler yeniden hesaba katılır. Bu algoritma, 'RSIMPLS' olarak isimlendirilir [15].

Hubert ve Vanden Branden [15], klasik SIMPLS, sağlam RSIMCD ve RSIMPLS algoritmalarını karşılaştırmak için farklı n, p, q, k ve $\hat{\boldsymbol{\Sigma}}_t$ değerleri seçmiştir. Aykırı değer içermeyen 'temiz' veri kümesi, n tane gözlemin % 10'nunu rasgele değiştirerek % 10 kötü kaldıraç gözlem, % 10 dikey aykırı değer içeren veri kümeleri türetilmiştir. Her bir benzetim düzeni için k bileşeni kullanarak bu üç algoritma için $\hat{\mathbf{B}}$ 'nın, $\hat{\boldsymbol{\beta}}_0$ 'ın ve artıkların kovaryans matrisi $\hat{\boldsymbol{\Sigma}}_e$ 'nın MSE'leri hesaplanmıştır. Ayrıca, modelde tek bir bağımlı değişken olduğunda (q=1), kestirilen parametre ve gerçek parametre arasındaki 'ortalama açı' hesaplanmıştır. Hubert ve Vanden Branden [15], daha çok parametre kestirimleri üzerinde durmuştur. Benzetim çalışmaları SIMPLS algoritmasının veri kümesi aykırı değerlerden dolayı bozulduğunda tüm MSE'lerin dikkat çekici bir şekilde yükseldiğini ve başarısız olduğunu göstermiştir. SIMPLS'nin aksine RSIMPLS ve RSIMCD'nin her ikisinin de, hem dikey yönlü aykırı değerlere karşı hem de kaldıraç gözlemlerine karşı dirençli ve temiz veri için de performanslarının iyi olduğu bulunmuştur. Ancak Hubert ve Vanden Branden [15], RSIMPLS algoritması RSIMCD'den iki kat daha hızlı olduğu için RSIMCD yerine RSIMPLS

algoritmasının kullanılmasını tavsiye etmiştir [7, 15, 21, 33]. Bu nedenle sağlam PLSR ile ilgili daha sonraki çalışmaların birçoğunda, sağlam PLSR için önerilen yöntemler RSIMPLS ile karşılaştırılmıştır.

Engelen vd. [5], klasik PCR, sağlam bir PCR yöntemi olan RPCR, SIMPLS ve RSIMPLS yöntemlerini etkinlik, uyum iyiliği (goodness-of-fit/GOF), kestirim başarısı ve sağlamlık açısından karşılaştırmıştır. Benzetim çalışması, hem büyük hem de küçük boyutlu veri kümeleri için temiz, % 10 kötü kaldıraç gözlemli, % 10 dikey aykırı değerli veri kümeleri türetilerek yapılmıştır. Her bir durum için m=100 tane veri kümesi türetilmiş ve k=1, 2, 3 bileşen için analiz sonuçları elde edilmiştir. Benzetim çalışması sonuçlarına göre, veri kümesi aykırı değer içermediğinde klasik PCR ve SIMPLS daha başarılı sonuçlar vermiştir. Ancak veri kümesi aykırı değerler içerdiğinde ise klasik PCR ve SIMPLS'nin aksine sağlam RSIMPLS ve RPCR daha dirençli sonuçlar vermiştir. Ayrıca, modelde daha az bileşen olduğunda RSIMPLS'nin RPCR'ye tercih edilmesi gerektiği belirtilmiştir [5].

3.5.3. Kısmi Sağlam M-Regresyon

PLS ismi, PLS'nin LS regresyon kestiricisinin kısmi bir parçası olabileceğini önerir. Bu ifadeyi yorumlamanın bir yolu, 'kısmi' sözcüğünü, bileşenler tarafından oluşturulan uzaya sınırlandırılan' olarak görmektir. Çünkü, PLSR'de LV'lerin kestirilmesinden sonra, bu bileşenler üzerinden bir ya da birden fazla bağımlı değişken kestirilir. Bu nedenle PLS'yi sağlamlaştırmak için doğru bir yaklaşım, sağlam regresyon kestiricisinin kısmi bir biçimini oluşturmaktır. Serneels vd. [31], M regresyon kestiricilerinin 'kısmi' bir biçimini önermiş ve bu kestiriciyi, Kısmi Sağlam M (partial robust M/PRM) regresyon kestiricisi olarak adlandırmıştır. Aykırı değerlerin ağırlığı azaltılarak oluşturulan PRM regresyonunda, **y** ve **x** değişkenler uzayındaki aykırı gözlemlerin etkisini azaltmak için yinelemeli bir şemada sıfır ile bir arasında değişen sürekli ağırlıklar hesaplanır [21]. Sağlam PLSR yöntemlerine ilişkin daha önceki çalışmalarda PLSR kestiricilerinin sağlam biçimlerini geliştirmeye odaklanılmışken, Serneels vd. [31] çalışmasında sağlam bir kestiricinin kısmi bir biçimini önermiştir [7, 31].

Kısmi M-regresyon kestiricilerini anlatmadan önce, standart regresyon düzenindeki M-kestiricilerinden kısaca söz etmek gerekir. **X**, $n \times p$ boyutlu bağımsız değişkenler matrisi ve **y**, $n \times 1$ boyutlu bağımlı değişken vektörünü göstersin. i. gözleme ilişkin bilgiyi içeren **X** ve **y**'nin i. satırları sırasıyla $\mathbf{x}_i$ ve y_i ile gösterilir. Regresyon modeli Eş. (3.29)'daki gibi tanımlansın [31].

$$y_i = \mathbf{x}_i\boldsymbol{\beta} + \varepsilon_i \tag{3.29}$$

Burada $\boldsymbol{\beta}$, p uzunluğuna sahip sütun vektörünü ve $\varepsilon_i, 1 \le i \le n$ ise hata terimlerini gösterir. $\boldsymbol{\beta}$'nın LS kestiricisi ise Eş. (3.30)'daki gibi tanımlanır [31].

$$\hat{\boldsymbol{\beta}}_{LS} = \underset{\hat{\boldsymbol{\beta}}}{\operatorname{argmin}} \sum_{i=1}^{n} (y_i - \mathbf{x}_i\boldsymbol{\beta})^2 \tag{3.30}$$

ε_i hata terimleri normal dağılıma sahip olduğunda bu kestirici, en iyi kestiricidir (en küçük varyansa sahip olan ve yansız olan). Ancak hata terimleri örneğin ağır kuyruklu (heavy-tailed) dağılımlar gibi başka dağılımlardan gelirse, LS en iyi olma özelliğini kaybeder ve başka tür kestiriciler daha iyi başarı gösterir [31].

En iyi bilinen sağlam kestiriciler, Eş. (3.30)'daki kareli ifadenin daha genel bir kayıp fonksiyonu 'ρ' ile yer değiştirmesi ile elde edilen, Eş. (3.31)'deki M kestiricileridir [31].

$$\hat{\boldsymbol{\beta}}_{M} = \underset{\hat{\boldsymbol{\beta}}}{\operatorname{argmin}} \sum_{i=1}^{n} \rho(y_i - \mathbf{x}_i\boldsymbol{\beta}) \tag{3.31}$$

Huber ilk kez 1964 ve 1972 yıllarında sağlam konum kestiricileri ile ilgili çalışmalarını yayınlamış ve M kestiricilerini geliştirmiştir. Bu eşitlikteki ρ kayıp fonksiyonu, simetrik ve azalmayan (tek en küçük değerini sıfırda alan) bir fonksiyon olmalıdır. Eğer $\rho(u) = u^2$ olur ise, kestirici özel bir durum olarak LS kestiricisine döner. $\rho(u)$ fonksiyonunun farklı seçimleri, hataların farklı dağılımlara sahip olduğunu varsaymak ile eştir. Büyük aykırı değerlere daha az önem vermek

için ρ sınırlandırılmış kayıp fonksiyonu seçilir, böylece LS'den daha sağlam bir kestirici elde edilir. $r_i = y_i - \mathbf{x}_i\boldsymbol{\beta}$, Eş. (3.31)'deki amaç fonksiyonunun artığını göstermek üzere i. gözleme ilişkin ağırlık Eş. (3.32)'deki gibi verilir. Daha sonra Eş. (3.31), Eş. (3.33)'deki gibi yeniden yazılır [23, 31].

$$w_i^r = \rho(r_i)/r_i^2 \tag{3.32}$$

$$\hat{\boldsymbol{\beta}}_M = \underset{\hat{\boldsymbol{\beta}}}{\operatorname{argmin}} \sum_{i=1}^{n} w_i^r (y_i - \mathbf{x}_i\boldsymbol{\beta})^2 \tag{3.33}$$

Eş. (3.33)'deki tanımda M kestiricisi, β'ya dayalı ağırlıklar ile ağırlıklandırılmış LS kestiricisi olarak ifade edilir. Eş. (3.33)'deki formül, M kestiricisinin IRLS algoritması ile hesaplanmasına olanak sağlar. IRLS algoritmasının temelinde, her bir gözlem için regresyon artığının boyutuna bağlı olarak bir ağırlık elde etmek yatmaktadır. Böylece en son modelde bu tür ağırlıklar ile aykırı değerlerin etkisini sınırlandırmak mümkün olur [23, 31].

Aykırı değerler, genellikle regresyon parametre kestirimleri üzerinde büyük etkiye sahiptir. Herhangi bir sağlam yöntemin amacı, aykırı değerlerin etkisini azaltmak ya da gidermek ve geriye kalan gözlemlerin büyük bir çoğunlukla sonuçları belirlemesine olanak sağlamaktır. Regresyonun M-kestiricileri, örneğin hata terimlerindeki aykırı değerler gibi, sadece dikey aykırı değerlere karşı sağlam olmalarından dolayı hep eleştirilmiştir. Bu nedenle kaldıraç gözlemlerine karşı sağlam bir kestirici elde etmek için, Eş. (3.33)'deki w_i^r ağırlığı ikinci bir w_i^x ağırlığı ile çarpılarak Eş. (3.34) elde edilmiştir [31].

$$\hat{\boldsymbol{\beta}}_{RM} = \underset{\hat{\boldsymbol{\beta}}}{\operatorname{argmin}} \sum_{i=1}^{n} w_i^r w_i^x (y_i - \mathbf{x}_i\boldsymbol{\beta})^2 \tag{3.34}$$

Bağımsız değişken uzayındaki veri bulutunun merkezine yakın gözlemler bire yakın ya da eşit w_i^x ağırlığı alırken, kaldıraç gözlemleri sıfıra yakın ağırlık alır.

Serneels vd. [31], Eş. (3.34)'deki M kestiricisinde hem artıklar hem de kaldıraç gözlemleri için ağırlıklar kullanıldığından çalışmalarında bu kestiriciyi 'Sağlam M kestiricisi' olarak isimlendirmiştir. Bu kestiriciler, hem dikey aykırı değerlere hem de kaldıraç gözlemlerine karşı dayanıklıdır [31].

Bu alt bölümde buraya kadar, klasik regresyon modeli için M-kestiricilerinin nasıl elde edildiğinden kısaca bahsedilmiştir. Bundan sonra, LV'ler regresyon modeli için 'kısmi sağlam M-kestiricilerinin' elde edilişi anlatılacaktır. LV'ler regresyon modeli, özellikle veri kümesindeki bağımsız değişken sayısı çok fazla ve özellikle çoklubağlantılı olduğunda, klasik regresyon modeline tercih edilen bir modeldir. LV'ler regresyon modelinde bağımlı değişken, sınırlı sayıda k tane LV ile modellenir. Bu LV'ler, satırları $\mathbf{t}_i, 1 \le i \le n$ vektörlerinden oluşan $n \times k$ boyutlu bileşen matrisi **T**'ye koyulur. LV regresyon modeli Eş. (3.35)'deki gibi verilir [31].

$$y_i = \mathbf{t}_i \boldsymbol{\gamma} + \varepsilon_i \tag{3.35}$$

γ'nin boyutu (k) küçük olduğu için, sağlam M kestiricisi kullanılarak bağımlı değişkeni LV'ler ile açıklayarak kestirilir. Klasik regresyonda kullanılan M kestiricilerinden temel farkı, burada dikey aykırı değerler ile ilgili olan w_i^r ağırlıkları $r_i = y_i - \mathbf{t}_i \boldsymbol{\gamma}$ artıklarından ve kaldıraç gözlemlerinin ağırlığını azaltan w_i^x ağırlıkları $\mathbf{t}_i$ bileşenlerinden hesaplanır. Eğer sadece dikey yöndeki aykırı değerler ile ilgilenmek için büyük artıkların ağırlığı azaltılır ise $w_i = w_i^r$ olur ve Kısmi M-kestiricisi (Partial M-estimator/PM) elde edilir. Ancak, olası kaldıraç gözlemlerine karşı da kestirimlerin korunması isteniyor ise, ağırlıklar Eş. (3.36)'daki gibi alınmalıdır. Bu durumda elde edilen kestirici, PRM olarak adlandırılır. PRM regresyon, katsayılarının kestiriminde 'kaldıraç gözlemleri' ve 'dikey aykırı değerler' olarak bilinen iki aykırı değer türünü de dikkate alarak, iyi sağlamlık özellikleri gösterir [21, 31].

$$w_i = w_i^r w_i^x \tag{3.36}$$

Geriye kalan tek konu, doğrudan gözlemlenmeyen **T** bileşen matrisini elde etmektir. Bunun için ise, aşağıdaki şema takip edilir. Buna göre, $\mathbf{p}_a, a = 1, \cdots, k$ yük vektörleri, Eş. (3.38)'deki kısıt altında Eş. (3.37)'deki gibi ardışık adımlar ile elde edilir [31].

$$\mathbf{p}_a = \underset{\mathbf{p}}{\operatorname{argmax}} \operatorname{Cov}_w(\mathbf{y}, \mathbf{X}\mathbf{p}) \tag{3.37}$$

$$\|\mathbf{p}\| = 1 \text{ ve } \operatorname{Cov}_w(\mathbf{X}\mathbf{p}, \mathbf{X}\mathbf{p}_j) = 0, 1 \le j < a \text{ için} \tag{3.38}$$

u, n uzunluğuna sahip başka bir vektör olmak üzere, $\operatorname{Cov}_w(\mathbf{y}, \mathbf{u}) = \frac{1}{n}\sum_{i=1}^{n} w_i y_i u_i$, ağırlıklandırılmış kovaryansı gösterir. Belirlenen yük vektörleri $p \times k$ boyutlu bir **P** matrisinin sütunlarına yerleştirildikten sonra, $\mathbf{T}_{n\times k} = \mathbf{X}_{n\times p}\mathbf{P}_{p\times k}$ ile gösterilen bileşen matrisi elde edilir. $\hat{\boldsymbol{\gamma}}$ elde edildikten sonra, Eş. (3.29)'daki β için en son kestirici $\hat{\boldsymbol{\beta}} = \mathbf{P}\hat{\boldsymbol{\gamma}}$ şeklinde elde edilir [31].

Tüm w_i ağırlıkları eşit alınır ise, özel bir durum olarak sağlam olmayan klasik PLSR kestirimleri elde edilir. Eğer ağırlıkları sabitmiş gibi düşünürsek, o zaman $\hat{\boldsymbol{\gamma}}$ ağırlıklandırılmış gözlemlerden $\left(\sqrt{w_i}\mathbf{x}_i, \sqrt{w_i}y_i\right)$ hesaplanan PLSR kestiricisi olur. Ancak, yukarıdaki tanımlardaki ağırlıklar bilinmeyen niceliklere sahiptir ve sabit değildir. Bu nedenle, ağırlıklar için uygun bir başlangıç değeri kullanılarak $\hat{\boldsymbol{\gamma}}$ kestiricisinin ilk yakınsaması, sabit ağırlıklarla birlikte PLSR'yi kullanarak hesaplanır. Daha sonra, başlangıç parametre kestirimleri kullanılarak ağırlıklar yeniden hesaplanır ve $\hat{\boldsymbol{\gamma}}$'nın ikinci bir yakınsaması, tekrar ağırlıklandırılmış PLSR uygulanarak elde edilir. Daha sonra, w_i ağırlıkları yeniden hesaplanır ve yineleme sürecine devam edilir. Bu nedenle, $\hat{\boldsymbol{\gamma}}$'yı hesaplamak için bir IRLS algoritması kullanılabilir [21, 31].

Kısmi Regresyon için M kestiricilerinin hesaplanması, bir IRLS algoritması ile tamamlanır. Bu nedenle, PRM regresyonu kestiricisini hesaplamak için daha önce Cummins ve Andrews [1] tarafından bulunan IRPLS yöntemine benzer, bir yinelemeli olarak ağırlıklandırılmış PLS algoritmasını kullanmak yeterli gelir. Ancak PRM için kullanılacak algoritmada, sağlam başlangıç değerleri ve hem artık hem de bileşen uzaylarındaki uzaklıklara dayanan ağırlıklar kullanılır. Böylece yöntem, hem dikey yöndeki aykırı değerlere karşı hem de kaldıraç gözlemlerine karşı sağlam hale gelir. Bu nedenle, PRM için sağlam başlangıç değerlerini ve dikkatli bir şekilde seçilmiş ağırlıkları kullanmak, çok önemli olacaktır. Bu iki husus, asıl IRPLS algoritmasının yer aldığı Cummins ve Andrews [1] çalışmasında gözden kaçırılmıştır. Cummins ve Andrews [1], sadece her bir adımdan sonra elde edilen artıkları kullandıkları için, onların yöntemleri kaldıraç gözlemlerine karşı sağlam değildir [7, 31].

Eş. (3.32)'deki w_i^r 'ler, PRM için hesaplanan algoritmada Eş. (3.39)'daki gibi hesaplanır. Eş. (3.39)'daki $\hat{\sigma}$, artık varyansının bir kestiricisidir ve buradaki f ağırlık fonksiyonu, Eş. (3.40)'daki gibi tanımlanır [31].

$$w_i^r = f\left(\frac{r_i}{\hat{\sigma}}, c\right) \tag{3.39}$$

$$f(z,c) = \frac{1}{\left(1+\left|\frac{z}{c}\right|\right)^2} \tag{3.40}$$

Eş. (3.40)'daki c, bir ayarlama sabitidir ve burada, c=4 alınmıştır. f ağırlık fonksiyonu ise, 'Fair' fonksiyonu olarak adlandırılır ve özgün IRPLS makalesinde verilen birkaç mümkün ağırlık fonksiyonundan biridir. Eğer c ayarlama sabiti sonsuzluğa doğru artar ise, bu durumda ağırlık fonksiyonu daha ve daha çok düz hale gelir ve PRM kestiricisi, daha ve daha çok klasik PLSR'ye benzer [31].

Eş. (3.39)'daki $\hat{\sigma}$ artık varyansı için basit ve sağlam bir kestirici, Eş. (3.41)'de tanımlanan MAD kestiricisidir [31].

$$\hat{\sigma} = \text{MAD}\left(r_1,\cdots,r_n\right) = \underset{i}{\text{ortanca}}\left|r_i - \text{ortanca}(\mathbf{r})\right| \tag{3.41}$$

Her bir $\mathbf{t}_i$ bileşenin ağırlığını ölçen w_i^x ağırlıkları Eş. (3.42)'deki gibi hesaplanır [31].

$$w_i^x = f\left(\frac{\left\|\mathbf{t}_i - \text{ortanca}_{L1}(\mathbf{T})\right\|}{\underset{i}{\text{ortanca}}\left\|\mathbf{t}_i - \text{ortanca}_{L1}(\mathbf{T})\right\|}, c\right) \tag{3.42}$$

Burada $\|.\|$, öklit normunu gösterir. Burada $\text{ortanca}_{L1}(\mathbf{T})$, $\{t_1,\cdots,t_n\}$ bileşen vektörlerinin koleksiyonundan hesaplanan L_1-ortancayı gösterir. L_1-ortanca, k boyutlu bileşen vektörü veri bulutunun merkezinin sağlam bir kestiricisidir. Aynı zamanda mekansal ortanca (spatial median) olarak da adlandırılan L_1-ortanca, örneklem ortancasının çok değişkenli bir türüdür; çok hızlı hesaplanır ve BDP'si % 50'dir. Tek değişkenli ortanca, tüm veri noktalarına olan uzaklıkların toplamını en küçük yapan noktadır. L_1-ortanca ise, çok değişkenli uzayda tüm veri noktalarına olan Öklit uzaklıklarının toplamını en küçük yapan nokta olarak tanımlanır. Çok değişkenli ortancayı kestirmek için bir başka yöntem ise, kordinat tarzı (coordinatewise) ortancadır. Bu ortanca, $\left(\underset{i}{\text{ortanca}}\,\mathbf{x}_{i1}, \underset{i}{\text{ortanca}}\,\mathbf{x}_{i2},\cdots, \underset{i}{\text{ortanca}}\,\mathbf{x}_{ip}\right)$ şeklinde tanımlanır. Her bir j değişkeni için $x_{1j}, x_{2j},\cdots,x_{nj}$ değerleri, n gözlemli bir boyutlu bir veri kümesi olarak düşünülür. Her bir örnekleme tek değişkenli sağlam ortanca kestiricisi uygulanır ve sonuçlar bir p boyutlu kestirimde toplanır. Bu nedenle, bu sürecin BDP'si de % 50'dir. Bağımsız değişkenler yerine bileşenler kullanıldığında ise bu sağlam kestirici, bileşen tarzı (componentwise) ortanca olarak adlandırılır [7, 28, 31].

Kısmi Sağlam M-Kestiricilerini hesaplanması için kullanılan PRM algoritması, aşağıdaki gibi tanımlanır [21, 31]:

PRM algoritmasının ilk ve en önemli adımında, ağırlıklara sağlam bir yöntemle ilk değer verilir. Bu nedenle, **X** matrisine ve **y** vektörüne 'sağlam ölçeklendirme' uygulanır. Genelde ölçeklendirme için ortalama ve standart sapma yerine, onların sağlam karşılıkları olan ortanca ve MAD uygulanır. Veri ortanca ile merkezileştirilir ve daha sonra, MAD ile bölünür [21]. Birçok sayısal deneme bu yinelemeli sürecin sağlam olduğunu ve oldukça hızlı bir şekilde yakınsadığını göstermiştir [31].

Adım 1: $\mathbf{x}_i$ ve y_i verisine ilişkin $w_i = w_i^x w_i^r$ ağırlıkları için, sağlam başlangıç değerleri hesaplanır. Daha sonra başlangıç değerleri Fair ağırlık fonksiyonuna geçirilerek, 0 ve 1 arasındaki değerlere dönüştürülür. Sürekli ağırlıklar seçerek, bir gözlem için aykırı değer ya da değil kararına ilişkin ikilemden kaçılır. Her bir gözleme, aykırılığının derecesine göre ağırlık verilir. Bu ilk adımda w_i^r 'ler hesaplanırken $r_i = y_i - \text{ortanca}(\mathbf{y})$ artıkları ve w_i^x'ler hesaplanırken ise $\mathbf{t}_i, 1 \le i \le n$ bileşen vektörleri yerine $\mathbf{x}_i$'ler kullanılır. Buna göre ilk adımda w_i^r ve w_i^x, Eş. (3.43)'deki biçimde hesaplanır.

$$w_i^r = f\left(\frac{y_i - \text{ortanca}(\mathbf{y})}{\underset{i}{\text{ortanca}}\left|y_i - \text{ortanca}(\mathbf{y})\right|}, c\right) \qquad w_i^x = f\left(\frac{\left\|\mathbf{x}_i - \text{ortanca}_{L1}(\mathbf{X})\right\|}{\underset{i}{\text{ortanca}}\left\|\mathbf{x}_i - \text{ortanca}_{L1}(\mathbf{X})\right\|}, c\right) \tag{3.43}$$

Adım 2: **X** ve **y**'nin her bir kolonunun $\sqrt{w_i}$ ile ağırlıklandırılması ile elde edilen yeniden ağırlıklandırılmış $\breve{\mathbf{X}}$ ve $\breve{\mathbf{y}}$ veri matrisleri üzerinde, SIMPLS algoritması kullanılarak PLSR uygulanır. Bu PLS analizinin sonucunda, $\hat{\boldsymbol{\gamma}}$ ve bileşen matrisi **T**'nin güncellenmiş bir hali elde edilir. Daha sonra, **T** matrisinin her bir satırı $\sqrt{w_i}$ ile bölünerek $\mathbf{t}_i = \mathbf{t}_i / \sqrt{w_i}, 1 \le i \le n$ şeklinde düzeltilir.

Adım 3: $r_i = y_i - \mathbf{t}_i\hat{\boldsymbol{\gamma}}$ formülü kullanılarak, artıklar yeniden hesaplanır. Daha sonra bu artıklar da Adım 1'deki gibi, $rc_i = r_i - ortanca(\mathbf{r})$ şeklinde ölçeklendirilir. Bu artıkları kullanılarak w_i^r ağırlıkları, PLS bileşenlerini kullanarak ise w_i^x ağırlıkları ve böylece, $w_i = w_i^r w_i^x$ toplam ağırlıkları yeniden hesaplanır.

Adım 4: Regresyon katsayıları $\hat{\boldsymbol{\gamma}}$'lar yakınsayana kadar, 2. ve 3. adımlar yinelenir. $\hat{\boldsymbol{\gamma}}$'nın iki ardışık kestirimi arasındaki normdaki göreli fark, 10^{-2} gibi belirli bir eşik değerinden küçük olduğunda yakınsaklık sağlanır.

Adım 5: En son kestirim $\hat{\boldsymbol{\beta}}$, en son ağırlıklandırılmış PLS adımından doğrudan elde edilir.

PRM algoritması, hesaplama açısından çok hızlıdır ve p>n olan yüksek boyutlu veri kümeleri için de kullanılır [7]. Ancak, p>n ise öncelikle $\mathbf{X}'$ matrisi üzerinde tekil değer ayrışımı (singular value decomposition/SVD) yaparak hesaplama hızlandırılır. Böylece bilgi kaybı olmadan, boyutluluk azaltılmış olur. **S**, köşegen elemanları **X** matrisinin n tane özdeğerinden oluşan köşegen bir matris ve **U**, $n \times n$ boyutlu bir dik matris olmak üzere $\mathbf{X}'_{p\times n} = \mathbf{V}_{p\times n}\mathbf{S}_{n\times n}\mathbf{U}'_{n\times n}$ yazılır. Yukarıda söz edilen yineleme şemasını **X** matrisi üzerinde yapmak yerine, bu yinelemeli süreç $n \times n$ boyutlu indirgenmiş $\tilde{\mathbf{X}} = \mathbf{US}$ matrisine uygulanır. Daha sonra elde edilen PRM kestircisi $\tilde{\boldsymbol{\beta}}$'nın, yeniden özgün değişkenler uzayı için $\hat{\boldsymbol{\beta}} = \mathbf{V}\tilde{\boldsymbol{\beta}}$'ya dönüştürülmesi gerekir. Bu şekilde elde edilen kestirim, matematiksel olarak algoritmanın doğrudan tam **X** matrisine uygulanması ile elde edilen kestirime denktir. SVD ile özgün veriye ilişkin herhangi bir bilgi kaybına uğranılmadığına göre, asıl olarak veri kümesini işlenebilir yapan ve hesaplamaları hızlandıran bir ön işlem adımı olarak görülebilir. Birçok algoritma için $n \times n$ boyutu yeterlidir ve özellikle yinelemeli yeniden ağırlıklandırma algoritmaları için hesaplanabilir etkinlikte bir artışa neden olur. Ancak SVD sadece boyut indirgemeyi içerdiğinden, hala sağlam kestirimin yapılması gerekir [7, 31].

PRM regresyon yönteminin teorik sağlamlık özellikleri, henüz bilinmemektedir. Yöntem, değiştirilebilir bir parametreye sahiptir. Önerilen biçimine yönelik yapılan benzetim çalışmaları, yöntemin iyi bir etkinlik ve yüksek sağlamlık özelliklerine sahip olduğunu gösterir [31]. Serneels vd. [31], Fair ağırlık fonksiyonundan başka IRLS ağırlık fonksiyonlarının da kullanılabileceğini ve seçilen ağırlık fonksiyonu için herhangi bir en iyi olma özelliği iddia etmemiştir. Ancak Serneels vd. [31], birçok sayısal uygulamanın önerilen ayarlama sabiti ile kullanılan Fair fonksiyonunun sağlamlık ve istatistiksel etkinlik arasında iyi bir uzlaşma sağladığını da ifade etmiştir.

3.5.4. Mekansal İşaret Dönüşümü ile PLSR Yöntemini Sağlamlaştırmak

Tek değişkenli durumda $\mathrm{sgn}(.)$ ile gösterilen işaret fonksiyonu, Eş. (3.43)'deki gibi tanımlanır [32].

$$\mathrm{sgn}(w)=\begin{cases} w<0 \quad \text{için} \quad -1 \\ w=0 \quad \text{için} \quad 0 \\ w>0 \quad \text{için} \quad 1 \end{cases} \tag{3.43}$$

Bu eşitliği yazmak için başka bir yol, Eş. (3.44)'deki gibidir. Eş. (3.44)'deki $|\cdot|$, mutlak değeri göstermektedir [32].

$$\mathrm{sgn}(w)=\begin{cases} w\neq 0 \quad \text{için} \quad w/|w| \\ w=0 \quad \text{için} \quad 0 \end{cases} \tag{3.44}$$

Tek değişkenli için işaret fonksiyonu bu şekilde tanımlandığında, kolaylıkla Eş. (3.45)'de gösterildiği gibi çok değişkenli düzene genelleştirilebilir. Eş. (3.45)'deki $\|\cdot\|$, Öklit normunu gösterir. Geometrik olarak mekansal işaret fonksiyonu, herhangi bir w noktasının başlangıç noktası yönünde bir birim kürenin üzerine yansımasıdır [32, 35].

$$\text{sgn}(\mathbf{w}) = \begin{cases} \mathbf{w} \neq \mathbf{0} & \text{için} \quad \mathbf{w}/\|\mathbf{w}\| \\ \mathbf{w} = \mathbf{0} & \text{için} \quad \mathbf{0} \end{cases} \tag{3.45}$$

İşaretlerin önemli bir bilgi verebilmesi için ilk önce, değişkenler merkezleştirilir. Bu nedenle mekansal işaret dönüşümünden bahsedilir ise, dolaylı olarak verinin merkezileştirildiği varsayılacaktır. Bir p-değişkenli **x** değişkenin n tane gözleminden oluşan sonlu bir örneklem için mekansal işaret kovaryans matrisi, Eş. (3.46)'daki gibi tanımlanır [32].

$$\hat{\mathbf{\Sigma}} = \frac{1}{n-1}\sum_{i=1}^{n} \text{sgn}(\mathbf{x}_i - \hat{\boldsymbol{\mu}})\,\text{sgn}(\mathbf{x}_i - \hat{\boldsymbol{\mu}})' \tag{3.46}$$

Eş. (3.46)'daki merkezileştirmeyi yapmak için bazı konum kestiricileri kullanılabilir. Serneels vd. [32], bu amaç için L_1 ortancanın (mekansal ortancanın) kullanılmasını önermiştir. Çünkü mekansal ortanca, çok hızlı bir şekilde hesaplanabilir. 'Mekansal işaret kovaryans matrisi', mekansal işaret dönüşümü ile ön işleme tabi tutulan veriye uygulanan klasik kovaryans matrisidir [32].

Mekansal işaret kovaryans matrisine dayalı sağlam PLSR yöntemi, hesaplama bakımından aşırı derecede etkindir. Yöntem, klasik PLSR analizi yapmadan önce verinin dönüştürülmesinden oluştuğu için mekansal işarete yansıma, veri ön işlemenin bir biçimi olarak görülür. SIMPLS algoritmasına bir mekansal işaret kovaryansının koyulması, veriye 'mekansal işaret ön işleme (spatial sign preprocessing/SS-PP)' ve akabinde standart bir PLS algoritmasının uygulanmasına eştir. Bu nedenle bu yöntem, 'SS-PP+PLS' olarak adlandırılır [7, 32]. PLS'yi sağlamlaştırırken mekansal işaret kovaryans matrisini kullanmanın iki tane avantajı vardır: Birincisi, mekansal işaret kovaryans matrisi çok basit bir matematiksel tanıma sahiptir, dolayısıyla algoritmadaki hesaplamalar kolay olur ve ikincisi, sınırlandırılmış bir IF'ye sahip olduğundan elde edilen PLS süreci sağlam olur [7].

SS-PP kavramsal olarak basittir ve kolay hesaplanır. Çünkü ayarlanması gereken herhangi bir parametre ya da fonksiyon içermez [32]. Mekansal işaret kovaryansları, sınırlandırılmış bir IF'ye sahiptir, veri kümesindeki makul orandaki aykırı değerler ile baş edebilir, çok değişkenli normallik varsayımına dayanmamaktadır. SS-PP+PLS yönteminin kırılma ve etkinlik özellikleri, henüz teorik olarak araştırılmamıştır. Ancak, bu özelliklere ilişkin benzetim çalışmaları yapılmıştır. Yöntemin en temel dezavantajı ise, sağlamlık özelliklerinin kullanıcı tarafından seçilememesi, yani yöntemin herhangi bir ayarlanabilir parametreye dayanmamasıdır [7]. Serneels vd. [32], SS-PP+PLS yönteminin hesaplanmasının kolay, normal modelde kabul edilir bir şekilde etkin ve Cauchy, Laplace modelleri gibi normal olmayan birkaç tane modelde iyi sonuçlar verdiği ve veri kümesinde aykırı değerlerden kaynaklanan çok fazla bozulmaya karşı kısmen sağlam olduğu sonuçlarına ulaşmıştır. Mekansal işaret dönüşümünün yan özellikleri karşılaştırıldığı diğer sağlam PLSR yöntemlerinden RSIMPLS ile PRM yöntemlerinin yan özelliklerinden daha az tercih edilebilir olduğundan, mekansal işaret dönüşümünün sadece hesaplama zamanının bir problem olduğu durumlarda kullanılması gerektiği belirtilmiştir [32].

4. SAĞLAM KISMİ EN KÜÇÜK KARELER REGRESYON ANALİZİNDE YENİ YAKLAŞIMLAR

Bu bölümde amaç, Alt Bölüm 2.2.2'te bahsedilen klasik PLS1 algoritmasının seçenek bir tanımı olan algoritmada yapılan değişiklik ile elde edilen ve sağlam kestirimler veren üç yeni sağlam PLSR yöntemini tanıtmaktır. Bu algoritmadaki $\mathbf{S}_{\mathbf{x}}$, $\mathbf{s}_{\mathbf{y},\mathbf{x}}$ kovaryanslarını elde etmek için bağımsız ve bağımlı değişkenlerden oluşan $\mathbf{z}_i^{'} = \left(y_i, \mathbf{x}_i\right)^{'}, i = 1, \cdots, n$ birleştirilmiş veri kümesinin kovaryans matrisi, $\mathbf{S}_z$, aşağıdaki alt bölümlerde detaylı olarak söz edilen üç sağlam kovaryans matrisi kestiricisi ' *ilk adımda konum ve kovaryansın* sağlam *başlangıç kestiricileri olarak En Küçük Kovaryans Determinantı kestiricilerini kullanan, uyarlanabilir yeniden ağırlıklandırılmış bir kovaryans kestiricisi* ', ' *S-kestiricileri* ' ve ' *MM-kestiricileri* ' kullanılarak ayrı ayrı kestirilmiş ve sırasıyla, *PLS-ARWMCD*, *PLS-Smult* ve *PLS-MMmult* olarak isimlendirilen üç yeni sağlam PLSR yöntemi önerilmiştir. Bu bölümde, yeni önerilen bu üç sağlam PLSR yöntemi ayrıntılı olarak incelenecektir.

4.1. Önerilen Sağlam PLSR Yöntemi PLS-ARWMCD

Bu bölümde, sağlam ve etkin uyarlanabilir yeniden ağırlıklandırılmış bir kovaryans kestiricisine dayalı olarak önerdiğimiz yeni sağlam PLSR yöntemi tanıtılacaktır. Bu yöntemde, 'FAST-MCD' algoritmasını kullanarak hesaplanan MCD kestiricisi kullanıldığı için ilk önce, detaylı olarak MCD kestiricisi ve FAST-MCD algoritmasının işleyişi incelenecektir.

Sağlam çok değişkenli konum ve kovaryans kestiricileri, istatistiksel çıkarsamada pratik uygulamada kullanılmak için aykırı değere karşı yüksek direncin yanında, normal model altında kayda değer bir etkinliğe ve kullanışlı asimptotik bir dağılıma sahip olmalıdır. Ancak, En Küçük Hacimli Elipsoid (Minimum Volume Ellipsoid/MVE) ve En Küçük Kovaryans Determinantı (Minimum Covariance Determinant/MCD) kestiricileri bu kategoride değildir. Gervini [8], hem sağlamlık hem de etkinlik aynı anda incelendiğinde, genel olarak en iyi seçimin iki-aşama

(two-stage) süreci olduğunu belirtmiştir. Bu süreçte ilk olarak, yüksek derecede sağlam ancak belki etkin olmayan bir kestirici hesaplanır. Daha sonra hesaplanan bu kestirici, Rousseeuw ve van Zomeren tarafından 1999 yılında yapılan çalışmadaki gibi aykırı değerleri belirlemek ve 'temizlenmiş' veri kümesinin örneklem ortalamasını ve kovaryansını hesaplamak için kullanılır. Rousseeuw ve van Zomeren tarafından önerilen süreçte, Mahalanobis uzaklıkları belirli bir eşik değerini aşan gözlemler analizden çıkartılır. Daha önceki yıllarda yapılan çalışmalarda, bu süreçler için başlangıç kestiricisi olarak genellikle MVE kestiricisi kullanılırdı. Ancak Rousseeuw ve Van Driessen [29], MCD'yi hesaplamak için 'FAST-MCD' algoritmasını önermiştir. FAST-MCD algoritması gözlem sayısının değişken sayısından çok fazla olduğu ($n >> p'$) büyük veri kümelerinde kullanıldığında, MVE ile MCD kestiricilerini hesaplamak için daha önce önerilen algoritmalardan daha hızlıdır. Gervini [8], tüm bu özelliklerine ek olarak $1/\sqrt{n}$ yakınsama oranı da düşünüldüğünde, iki-adım sürecinin başlangıç kestiricisi için FAST-MCD algoritmasını kullanan MCD yönteminin MVE'ye kıyasla mevcut en iyi seçim olduğunu belirtmiştir [8].

Çok değişkenli konum ve kovaryansı kestirmek için kullanılan MVE yöntemi, $n/2 \le h < n$ olmak üzere h tane gözlemi kapsayan en küçük elipsoiti bulmaya çalışır. MVE yönteminin kırılma noktası (breakdown point/BDP), $(n-h)/n$'dir [29]. MVE'ye seçenek olarak 1984 yılında Rousseeuw tarafından önerilen MCD yöntemi ise, n gözlem üzerinden klasik kovaryans matrisinin determinantı en küçük olan h gözlemi bulmaya çalışır. Bu durumda konum ve kovaryansın MCD kestirimleri, sırasıyla bu h gözlemin ortalama ve kovaryans matrisleri olacaktır. MCD kestiricisinin hesaplanması kolay değildir. $\mathbf{z}_i' = (y_i, \mathbf{x}_i)', i = 1, \cdots, n$ şeklinde tanımlanan birleştirilmiş bir veri kümesi için MCD kestiricisi sadece, $n > p+1 = p'$ olan küçük boyutlu (low-dimensional) veri kümelerine uygulanabilir; çünkü $p' > n$ olursa aynı zamanda $p' > h$ olur ve herhangi bir h gözlemin kovaryans matrisi her zaman tekil olacağından elde edilen determinant da sıfır olacaktır. Bu nedenle, h gözlemin her bir alt kümesi mümkün en küçük determinantı verecektir ve tek bir sonuç elde edilemeyecektir [7].

FAST-MCD algoritması gözlem sayısının değişken sayısından çok çok büyük olduğu (on binlerce gözlemin olduğu) veri kümeleri için de kullanılabilir. Gözlem sayısının az olduğu küçük veri kümeleri için FAST-MCD tam (exact) sonucu bulur ve gözlem sayısının çok büyük olduğu daha büyük veri kümeleri için daha önce önerilmiş algoritmalardan daha hızlıdır. Rousseeuw ve Van Driessen [29], hem istatistiksel etkinlik hem de hesaplamadaki hızlılığı göz önüne alındığında konum ve kovaryansı kestirmek için FAST-MCD algoritmasının kullanılmasını önermiştir [28, 29]. FAST-MCD algoritmasında konum ve kovaryansın ham MCD kestiricileri, sonlu örneklem etkinliklerini yeterince arttırmak için yeniden ağırlıklandırıldıklarında, Yeniden Ağırlıklandırılmış En Küçük Kovaryans Determinantı (Reweighted Minimum Covariance Determinant /RMCD) kestiricileri olarak adlandırılır [15, 23].

4.1.1. FAST-MCD Algoritmasında Kullanılan C-adımı ve Dayandığı Teorem

FAST-MCD algoritmasının en önemli adımına göre, MCD için herhangi bir kestirimden başlayarak daha küçük determinanta sahip başka bir kestirim hesaplamak mümkündür. FAST-MCD algoritmasında kullanılan '*C-adım*' sürecinin işleyişi aşağıdaki Teorem 1'de verilmiştir [29].

Teorem 1: $\mathbf{z}_i' = (y_i, \mathbf{x}_i)', i = 1,\cdots,n$ olmak üzere $p' = p+1$ değişkenli $\mathbf{Z}_n = \{\mathbf{z}_1,\cdots,\mathbf{z}_n\}$ şeklinde gösterilen bir veri kümesi olsun. $H_1 \subset \{1,\cdots,n\}$ ve $|H_1| = h$ şeklinde tanımlanan bir gözlemler kümesi olsun. Burada H_1, en küçük determinanta sahip h gözlemli alt kümeyi gösterir. Buna göre, bu h gözlemli alt küme için konum ve kovaryans sırasıyla, $\hat{\boldsymbol{\mu}}_1 := (1/h)\sum_{i\in H_1}\mathbf{z}_i$ ve $\hat{\boldsymbol{\Sigma}}_1 := (1/h)\sum_{i\in H_1}(\mathbf{z}_i - \hat{\boldsymbol{\mu}}_1)(\mathbf{z}_i - \hat{\boldsymbol{\mu}}_1)'$ olmak üzere $\det(\hat{\boldsymbol{\Sigma}}_1) \neq 0$ ise, $d_1(i) := \sqrt{(\mathbf{z}_1 - \hat{\boldsymbol{\mu}}_1)'\hat{\boldsymbol{\Sigma}}_1^{-1}(\mathbf{z}_1 - \hat{\boldsymbol{\mu}}_1)}, i = 1,\cdots,n$ şeklinde ilgili uzaklıklar tanımlanır. Daha sonra, $(d_1)_{1:n} \le (d_1)_{2:n} \le \cdots \le (d_1)_{n:n}$ sıralı uzaklıkları göstermek üzere $\{d_1(i); i \in H_2\} := \{(d_1)_{1:n},\cdots,(d_1)_{h:n}\}$ şeklinde tanımlanan bir H_2 gözlemler kümesi alınır ve H_2'ye dayalı olarak $\hat{\boldsymbol{\mu}}_2$ ve $\hat{\boldsymbol{\Sigma}}_2$ hesaplanır. Buna göre, ancak ve ancak $\hat{\boldsymbol{\mu}}_2 = \hat{\boldsymbol{\mu}}_1$ ve $\hat{\boldsymbol{\Sigma}}_2 = \hat{\boldsymbol{\Sigma}}_1$ eşitlikleri geçerli ise $\det(\hat{\boldsymbol{\Sigma}}_2) \le \det(\hat{\boldsymbol{\Sigma}}_1)$ olur [29].

Eğer $\det(\hat{\mathbf{\Sigma}}_1) > 0$ ise Teorem 1 uygulandığında, $\det(\hat{\mathbf{\Sigma}}_2) \leq \det(\hat{\mathbf{\Sigma}}_1)$ olan bir $\hat{\mathbf{\Sigma}}_2$ matrisi elde edilir. FAST-MCD algoritmasında Teorem1'deki yapı '*C-adım*' olarak adlandırılır. Buradaki '*C*' harfi, $\hat{\mathbf{\Sigma}}_2$ kovaryansı H_2'nin kovaryansını gösterdiği için '*kovaryans (covariance)*' kelimesinin baş harfini ya da en küçük uzaklığa sahip h tane gözleme konsantre olunduğu ve $\hat{\mathbf{\Sigma}}_2$ kovaryansı $\hat{\mathbf{\Sigma}}_1$ kovaryansından daha küçük determinanta sahip olduğundan daha çok konsantre olduğu için 'konsantrasyon (concentration)' sözcüğünün baş harfini temsil eder. Algoritmik ifadeler ile C-adım aşağıdaki gibi tanımlanır [29].

h gözlemden oluşan H_{eski} alt kümesi ya da $(\hat{\boldsymbol{\mu}}_{eski}, \hat{\mathbf{\Sigma}}_{eski})$ çifti verilsin:

- $i = 1, \cdots, n$ için $d_{eski}(i)$ uzaklıkları hesaplanır,
- bu uzaklıklar sıralanır, böylece π dizilimi elde edilir ve $d_{eski}(\pi(1)) \leq d_{eski}(\pi(2)) \leq \cdots \leq d_{eski}(\pi(n))$ sıralaması yapılır.
- $H_{yeni} := \{\pi(1), \pi(2), \cdots, \pi(h)\}$ şeklinde oluşturulur.
- $\hat{\boldsymbol{\mu}}_{yeni} := ortalama(H_{yeni})$ ve $\hat{\mathbf{\Sigma}}_{yeni} := kovaryans(H_{yeni})$ hesaplanır.

C-adımlarını tekrar etmek, bir yineleme süreci ortaya çıkarır. Eğer $\det(\hat{\mathbf{\Sigma}}_2) = 0$ ya da $\det(\hat{\mathbf{\Sigma}}_2) = \det(\hat{\mathbf{\Sigma}}_1)$ ise durulur; aksi halde başka bir C-adımı çalıştırılıp $\det(\hat{\mathbf{\Sigma}}_3)$ elde edilir ve bu şekilde devam edilir. $\det(\hat{\mathbf{\Sigma}}_1) \geq \det(\hat{\mathbf{\Sigma}}_2) \geq \det(\hat{\mathbf{\Sigma}}_3) \geq \cdots$ dizilişi negatif olmadığı için yakınsar. Gerçekten de sınırlı sayıda çok fazla h gözlemli alt kümesi olduğuna göre, öyle bir 'm' indisi olmalı ki $\det(\hat{\mathbf{\Sigma}}_m) = 0$ ya da $\det(\hat{\mathbf{\Sigma}}_m) = \det(\hat{\mathbf{\Sigma}}_{m-1})$ olur ve böylece yakınsamaya ulaşılır. Uygulamada genellikle m indisi, 10'nun altındadır. Yakınsamaya ulaşıldıktan sonra, $(\hat{\boldsymbol{\mu}}_m, \hat{\mathbf{\Sigma}}_m)$ üzerinde C-adımı çalıştırmak determinantı daha fazla indirgemez. Bu, $\det(\hat{\mathbf{\Sigma}}_m)$'nin MCD amaç fonksiyonunun genel minimumu olması için yeterli değildir, ancak gerekli bir şarttır. Bu nedenle, Teorem 1 bir algoritma için kısmi bir fikir verir: ' *H_1 alt kümesi için birçok başlangıç*

seçimi yap ve yakınsayana kadar her birine C-adımı uygula ve en küçük determinanta sahip sonucu sakla '. Ancak bu fikri işlevsel yapabilmek için başlangıçta kullanılacak H_1 alt kümeleri nasıl oluşturulacak, kaç tane H_1 alt kümesine ihtiyaç var, birkaç H_1 alt kümesi aynı sonucu verebileceğine göre tekrardan nasıl kaçılacak, daha az C-adımı ile yapılamaz mı, büyük örneklemlerde ne olacak ve bunun gibi bazı sorulara cevap verilmesi gerekir. Bu sorular, sonraki bölümlerde tartışılacaktır [29].

4.1.2. H_1 Başlangıç Alt Kümelerinin Oluşturulması

Bir önceki alt bölümde bir algoritma oluşturmak için verilen kısmi fikri uygulayabilmek için ilk önce, H_1 alt kümelerinin nasıl oluşturulacağına karar verilmelidir. Bu amaçla Rousseeuw ve Van Driessen [29] çalışmasında verilen yönteme göre, ilk olarak $(p'+1)$ gözlemli bir J alt kümesi seçilmelidir ve daha sonra $\hat{\boldsymbol{\mu}}_0 := \text{ortalama}(J)$ ve $\hat{\boldsymbol{\Sigma}}_0 := \text{kovaryans}(J)$ hesaplanır. Eğer $\det(\hat{\boldsymbol{\Sigma}}_0) = 0$ ise J alt kümesine rasgele başka bir gözlem eklenerek genişletilir ve $\det(\hat{\boldsymbol{\Sigma}}_0) > 0$ olana kadar gözlem eklenmeye devam edilir. Daha sonra, $i = 1,\cdots,n$ için $d_0^2(i) := (\mathbf{z}_i - \hat{\boldsymbol{\mu}}_0)' \hat{\boldsymbol{\Sigma}}_0^{-1} (\mathbf{z}_i - \hat{\boldsymbol{\mu}}_0)$ uzaklıkları hesaplanır. Bu uzaklıklar $d_0(\pi(1)) \le \cdots \le d_0(\pi(n))$ şeklinde sıralanır ve $H_1 := \{\pi(1),\cdots,\pi(h)\}$ oluşturulur. Rousseeuw ve Van Driessen [29], $p'+1$'den daha az gözlem seçildiğinde $\hat{\boldsymbol{\Sigma}}_0$ her zaman tekil olacağı için, bu sayıdan daha az gözlem seçilmemesi gerektiğini belirtmiştir [29].

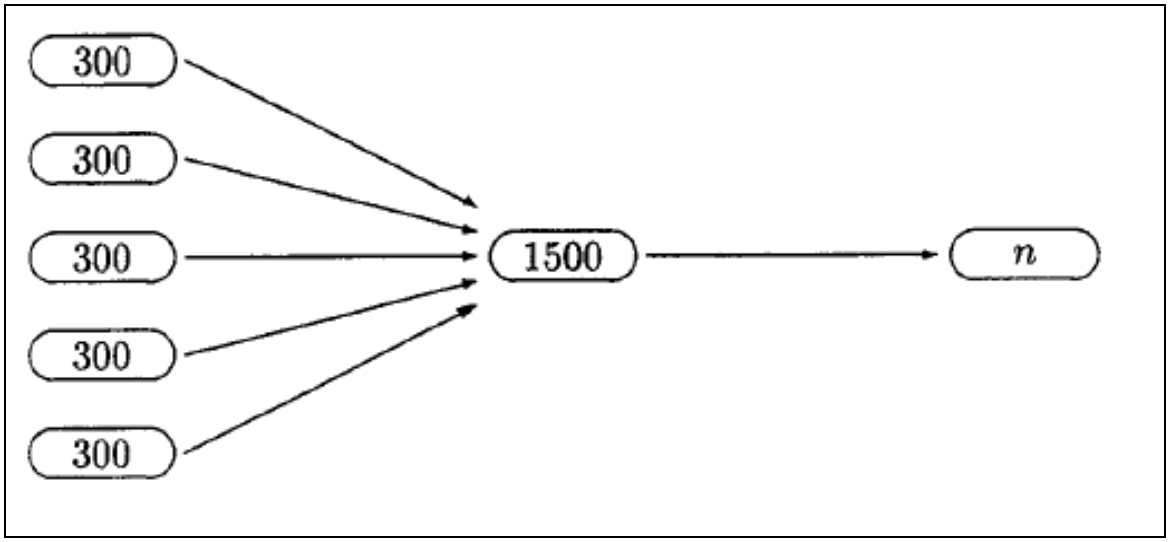
4.1.3. Seçmeli Yineleme

Her bir C-adımı, bir kovaryans matrisi, bu kovaryans matrisinin determinantını ve tüm ilgili uzaklıklarını hesaplar. Bu nedenle, C-adımlarının sayısını azaltmak hızı arttıracaktır. Rousseeuw ve Van Driessen [29], sağlam sonuçlar ve sağlam olmayan sonuçlar arasındaki ayrımın genellikle iki ya da üç C-adımında ortaya çıktığını ifade etmiştir. Ayrıca, her bir başlangıç H_1 alt kümesi için sadece iki

C-adımı yapılması, en küçük determinanta sahip 10 farklı H_3 alt kümesinin seçilmesi ve sadece bu 10 alt küme için yakınsayana kadar C-adımlarının yapılmasını önermiştir [29].

4.1.4. İç içe Eklemeler

n gözlem sayısı küçük olduğunda, Alt Bölüm 4.1.1'de C-adımı için bahsedilen algoritma çok zaman almayacaktır. Ancak, n arttığı zaman özellikle her seferinde hesaplanması gereken uzaklıklar yüzünden hesaplama zamanı artar. Veri kümesinin tamamı üzerinde tüm hesaplamaları yapmaktan kaçınmak için, Rousseeuw ve Van Driessen [29] özel bir yapı kullanmıştır. n>1500 olduğunda, algoritma Şekil 4.1'de gösterildiği üzere alt kümelerin bir iç içe sistemini üretir. Şekil 4.1'deki oklar ' …*'nın alt kümesi* ' anlamına gelir [29].



Şekil 4.1. FAST-MCD Algoritması Tarafından Üretilen Alt Kümelerin İç İçe Sistemi.

Şekil 4.1'deki 300 gözlemli beş alt küme çakışmaz ve birlikte 1500 gözlemli birleşmiş bir veri kümesini oluşturur. Bu birleşmiş veri kümesi ise, n boyutlu veri kümesinin gerçek alt kümesidir. Şekil 4.1'de gösterilen yöntem iki aşama ile çalıştığı için, '*iç içe*' adı kullanılır. Şekil 4.1'yi oluşturmak için algoritma 1500 gözlemi yerine koymadan tek tek çeker. İlk seçilen 300 gözlem, ilk alt kümeye koyulur ve diğer alt kümeler de böyle oluşturulur. Bu sistem yüzünden, her bir 300 gözlemli alt küme tüm veri kümesini kabaca temsil eder ve 1500 gözlemli birleşmiş

veri kümesi ise tüm veri kümesini bu 300 gözlemli alt kümelerden daha iyi temsil eder. n<600 ise, algoritma Alt Bölüm 4.1.1'deki gibi işler. Ancak, $n \geq 1500$ olduğu zaman Şekil 4.1 kullanılır [29].

4.1.5. FAST-MCD Algoritması ve İşleyişi

Önceki alt bölümlerde söz edilen tüm bileşenler birleştirilerek, FAST-MCD algoritması elde edilir. $p' = p + 1$ boyutlu $\mathbf{z}_i' = (y_i, \mathbf{x}_i)', i = 1, \cdots, n$ birleştirilmiş veri kümesi için algoritmanın adımları ve işleyişi aşağıdaki gibidir [29].

Adım 1: MCD kestiricisi $(n - h)$'ye kadar olan aykırı gözlem sayısına karşı sağlam sonuçlar verebilir. Bu nedenle, h sayısı (ya da benzer bir şekilde $\alpha = h/n$ oranı) kestiricinin sağlamlığını belirler. FAST-MCD algoritmasında varsayılan h değeri, $[(n + p' + 1)/2]$'dir ve bu değer için aykırı değerler tarafından bozulmaya karşı en yüksek dirençlilik sağlanır. Ancak, $[(n + p' + 1)/2] \leq h < n$ aralığında herhangi bir h tamsayısı da seçilebilir. Eğer verideki aykırı değer oranının büyük olduğu tahmin ediliyorsa, α=0.5 alınarak $h = [0.5n]$ şeklinde seçilmelidir. Ancak aksi durumda, genellikle beklenen bir durum olarak verideki aykırı değer yüzdesi verinin % 25'inden daha az ise, $h = [0.75n]$ alınarak BDP değeri ve istatistiksel etkinlik arasında iyi bir uzlaşma elde edilir [29, 34].

Adım 2: $h < n$ ve $p' \geq 2$ olsun. Eğer n küçük ise (örneğin, n<600) o zaman,

- 500 defa tekrar edilir:
- ✓ Alt bölüm 4.1.2'deki yöntemi kullanarak H_1 ile gösterilen bir başlangıç alt kümesi oluşturulur, örneğin, rasgele bir $(p' + 1)$ gözlemli alt kümeden başlayarak
- ✓ Alt bölüm 4.1.1'de bahsedilen C-adımı iki kez uygulanır

- en küçük $\det(\hat{\mathbf{\Sigma}}_3)$'e sahip 10 sonuç için:
  - ✓ yakınsayana kadar C-adımları uygulanır

- en küçük $\det(\hat{\mathbf{\Sigma}})$'a sahip çözüm; $(\hat{\boldsymbol{\mu}}, \hat{\mathbf{\Sigma}})$ verilir

Adım 3: Eğer n büyük ise (örneğin, n≥600) o zaman

- Alt bölüm 4.1.4'de anlatıldığı gibi n_{alt} boyutlu beş tane ayrık rasgele alt küme oluşturulur (örneğin, $n_{alt} = 300$ gözlemli 5 tane alt küme)

- her bir alt kümenin içinde, 500/5=100 kez aşağıdakiler tekrar edilir:
  - ✓ $h_{alt} = [n_{alt}(h/n)]$ boyutlu bir başlangıç alt kümesi H_1 oluşturulur
  - ✓ n_{alt} ve h_{alt}'ı kullanarak iki kez C-adımı yapılır
  - ✓ en iyi 10 sonuç $(\hat{\boldsymbol{\mu}}_{alt}, \hat{\mathbf{\Sigma}}_{alt})$ saklanır

- alt kümeler bir arada toplanır ve birleşmiş bir veri kümesi oluşturulur (örneğin, $n_{birleşmiş}$=1500)

- birleşmiş kümede, 50 tane $(\hat{\boldsymbol{\mu}}_{alt}, \hat{\mathbf{\Sigma}}_{alt})$ çözümün her biri için aşağıdakiler tekrar edilir:
  - ✓ $n_{birleşmiş}$ ve $h_{birleşmiş}=[n_{birleşmiş}(h/n)]$'i kullanarak iki kez C-adım gerçekleştirilir
  - ✓ en iyi 10 tane ($\hat{\mathbf{\Sigma}}_{birleşmiş}$, $\hat{\boldsymbol{\mu}}_{birleşmiş}$) sonucu korunur

- tam veri kümesinde, en iyi m_{tam} sonuç için tekrar edilir:

- ✓ n ve h'yi kullanarak birkaç kez C-adımı yapılır
- ✓ en iyi son 10 tane $\left(\hat{\boldsymbol{\mu}}_{tam}, \hat{\boldsymbol{\Sigma}}_{tam}\right)$ sonucu saklanır

Burada m_{tam} ve C-adımının kaç kez yapılacağı (tercihen, yakınsayana kadar) veri kümesinin ne kadar büyük olduğuna dayalıdır [29].

Bu algoritma, FAST-MCD olarak adlandırılır. Bu kestirici, uyum eşdeğişkenlidir (affine equivariant). Bu nedenle, veri dönüştürüldüğünde ya da doğrusal dönüşüme tabi tutulduğunda, sonuç da $\left(\hat{\boldsymbol{\mu}}_{tam}, \hat{\boldsymbol{\Sigma}}_{tam}\right)$ buna uygun olarak dönüşecektir. Kolaylık sağlamak için, iki ilave adım daha içerir [29].

Adım 4: Veri kümesi çok değişkenli normal bir dağılımdan geldiğinde tutarlılığı elde etmek için $\hat{\boldsymbol{\mu}}_{MCD} = \hat{\boldsymbol{\mu}}_{tam}$ ve $\hat{\boldsymbol{\Sigma}}_{MCD} = \dfrac{\text{ortanca}_i\, d^2_{\left(\hat{\boldsymbol{\mu}}_{tam}, \hat{\boldsymbol{\Sigma}}_{tam}\right)}(i)}{\chi^2_{p,0.5}} \hat{\boldsymbol{\Sigma}}_{tam}$ olur.

Adım 5: Bir 'bir-adım yeniden ağırlıklandırma' kestirimlerini elde etmek için her bir gözlem Eş. (4.1)'deki gibi ağırlıklandırılır. Böylece, bu ağırlıkları kullanarak RMCD kestiricileri Eş. (4.2)'deki gibi elde edilir.

$$w_i = \begin{cases} 1, & \left(\mathbf{z}_i - \hat{\boldsymbol{\mu}}_{MCD}\right)' \hat{\boldsymbol{\Sigma}}^{-1}_{MCD} \left(\mathbf{z}_i - \hat{\boldsymbol{\mu}}_{MCD}\right) \le \chi^2_{p',0.975} \\ 0, & \text{ö.d.} \end{cases} \tag{4.1}$$

$$\hat{\boldsymbol{\mu}}_{RMCD} = \frac{\sum_{i=1}^{n} w_i \mathbf{z}_i}{\sum_{i=1}^{n} w_i} \text{ ve } \hat{\boldsymbol{\Sigma}}_{RMCD} = \frac{\sum_{i=1}^{n} w_i \left(\mathbf{z}_i - \hat{\boldsymbol{\mu}}_{RMCD}\right)\left(\mathbf{z}_i - \hat{\boldsymbol{\mu}}_{RMCD}\right)'}{\sum_{i=1}^{n} w_i} \tag{4.2}$$

MATLAB LIBRA Toolbox'ta '*mcdcov*' fonksiyonu ile hesaplanan ve bizim de tezimizde kullandığımız FAST-MCD algoritmasının işleyişi şu şekilde özetlenebilir [34]:

- n<600 olduğunda, veri kümesi bir bütün olarak analiz edilir. Veri kümesi bir bütün olarak analiz edildiğinde, veri kümesinden $p'+1$ gözlemli bir alt küme alınır. Bu gözlemlerin, ortalaması ve kovaryansı hesaplanır. En küçük uzaklığa sahip h gözlem, bir sonraki ortalama ve kovaryansı hesaplamak için kullanılır. Bu döngü ise, iki C-adımı kez tekrar eder. FAST-MCD, bir yeniden örnekleme algoritmasıdır. n gözlemli tüm veri kümesinden $p'+1$ gözlemli 500 alt kümeyi rasgele olarak seçer. Daha sonra, en iyi 10 sonuç (konum ve kovaryans) en son yineleme için başlangıç değerleri olarak kullanılır. Rousseeuw ve Van Driessen [29], yüksek bir olasılık ile en azından bir tane temiz alt küme seçmeyi garanti etmek için alt küme sayısını '500' olarak belirlemiştir. Bu yinelemeler, ard arda gelen iki determinant eşit olduğunda durur. En fazla, üç C-adımı yinelemesi yapılır. En küçük determinanta sahip sonuç (konum ve kovaryans) saklanır.

- Ancak, n≥600 olduğunda (n<1500 ya da değil) algoritma hesaplamaları parça parça, kabaca 1500 gözlemden oluşan aynı gözlemlerin yer almadığı en fazla 5 tane alt kümede yapar. Bu durumda algoritma, üç aşamada işler:

✓ Aşama 1: Her bir alt veri kümesindeki H_1 alt örneklemi için C-adımı iki yineleme ile yapılır. Bu aşamada, 5 tane alt küme ve 500 tane alt örneklem seçilir. Her bir alt küme için, en iyi 10 sonuç (konum ve kovaryanslar) saklanır.

✓ Aşama 2: Daha sonra, alt kümelerinin birleşimi olan ve en fazla 1500 gözlemli bir birleşmiş veri kümesi ele alınır. Eğer n büyük ise, birleşmiş veri kümesi tüm veri kümesinin gerçek alt kümesidir. Aşama 1'deki (en fazla 50 tane olan) $\left(\hat{\boldsymbol{\mu}}_{\text{alt}}, \hat{\boldsymbol{\Sigma}}_{\text{alt}}\right)$ en iyi sonuçlardan her biri, birleşmiş veri kümesine

uygulanan C-adımının yinelemeleri için başlangıç değerleri olarak kullanılır. Bu aşamada, bu kez birleşik veri kümesindeki tüm 1500 gözlem kullanılarak her bir $\left(\hat{\boldsymbol{\mu}}_{alt}, \hat{\boldsymbol{\Sigma}}_{alt}\right)$'den başlayarak C-adım yapılmaya devam edilir. İlk aşamada olduğu gibi bu aşamada da, C-adımı iki yineleme ile yapılır. Buradan da, elde edilen en iyi 10 sonuç ($\hat{\boldsymbol{\mu}}_{birleşmiş}, \hat{\boldsymbol{\Sigma}}_{birleşmiş}$) saklanır.

- ✓ Aşama 3: Bu aşama, veri kümesindeki gözlem sayısı n'ye bağlıdır. En son olarak aynı yolla bu 10 sonuçtan her biri için tüm veri kümesi dikkate alınarak C-adım uygulanır ve en iyi $\left(\hat{\boldsymbol{\mu}}_{tam}, \hat{\boldsymbol{\Sigma}}_{tam}\right)$ sonucu elde edilir. En son hesaplamalar tüm veri kümesi üzerinde yapıldığı için, n arttığı zaman daha çok zaman alır. Bu nedenle, Rousseeuw ve Van Driessen [29] n çok büyük olduğunda algoritmayı hızlandırmak için ($\hat{\boldsymbol{\mu}}_{birleşmiş}, \hat{\boldsymbol{\Sigma}}_{birleşmiş}$) başlangıç çözümlerinin sayısının ve/ya da tüm veri kümesindeki C-adımlarının sayısının sınırlandırılabileceği ifade etmiştir [29, 34]. Buna göre, '*mcdcov*' fonksiyonunda varsayılan değerler şöyledir: $n \leq 5000$ ise ilk 10 sonucun hepsi yinelenir. Eğer $n > 5000$ ise, sadece en iyi ilk sonuç yinelenir. $n*p$'ye göre yineleme sayısı 1'e gerileyebilir. Eğer $n*p \leq 100000$ ise, üçüncü aşamada tüm veri kümesinde uygulanan C adımı 100 kez yinelenir, $n*p > 1000000$ ise sadece tek bir yineleme yapılır [34].

Rousseeuw ve Van Driessen [29], 300 gözlemden oluşan 5 alt örneklem, 500 alt örneklem sayısı, 10 en iyi sonuç ve benzeri bazı seçilmiş sayıların çeşitli deneysel denemelere dayandığını belirtmiştir. Bu varsayılan seçimler sayesinde kullanıcı herhangi bir seçim yapmak zorunda olmamasına karşın, bu sayılardan farklı sayıların da seçilebileceği ifade edilmiştir [29].

FAST-MCD algoritması ve bu algoritmayı hesaplayan MATLAB LIBRA Toolbox'ta [34] verilen ' *mcdcov* ' fonksiyonu hakkında bilgiler verildikten sonra, bir sonraki alt bölümde bu algoritmayı da kullanarak Gervini [8] çalışmasında önerilen ' *sağlam ve etkin uyarlanabilir yeniden ağırlıklandırılmış bir kovaryans kestiricisi* ' hakkında bilgi verilecektir.

4.1.6. Sağlam ve Etkin Uyarlanabilir Yeniden Ağırlıklandırılmış Bir Kovaryans Kestiricisi

Doğrusal regresyonda, hem yüksek etkinliği hem de hem de yüksek sağlamlığı aynı anda sağlamak için birçok kestirici önerilmiştir. Genellikle, bu yöntemler de aynı zamanda iki-aşama süreçleridir. Bu konuda bilinen en iyi yöntemler; 1984 yılında Rousseeuw tarafından önerilen yeniden ağırlıklandırma ya da 1987 yılında Jureckova´ ve Portnoy ile 1992 yılında Simpson vd. tarafından önerilen Newton-Raphson adımlarını kullanarak hesaplanan bir-adım kestiricileri, 1987 ve 1988 yılllarında Yohai'nin önerdiği MM kestiricileri ve τ-kestiricileri gibi, ikinci aşamada etkin bir amaç fonksiyonunu en küçük yapan kestiricilerdir. Bu kestiricilerin tümü, en yüksek BDP'ye ve keyfi olarak en yüksek etkinliğe ulaşabilir. Ancak Rousseeuw'un 1994 yılında yayınlanan çalışmasında belirtildiği üzere, etkinlikteki kazanımlar daha yüksek yana neden olabilir. Çünkü, tüm bu yöntemler uyarlanamaz (non-adaptive) ve daha yüksek etkinlik sadece artan ayarlama parametreleri ile elde edilir ve dolayısıyla bu da, veri kümesi aykırı değerler tarafından bozulduğunda yanı etkiler. Gervini [8], temelde 1990 yılında Rousseeuw ve Van Zomeren tarafından önerilen yöntemi geliştirerek bir yöntem önermiştir. Bu yöntem, uyarlanabilir eşik değerlerini kullanan bir yeniden ağırlıklandırma bir-adım kestiricisinden oluşur. Bu 'uyarlanabilir yeniden ağırlıklandırma (adaptive reweighted)' şeması, başlangıç kestiricisinin kırılmadaki aykırı değer dirençliliğini ve yanını korurken, aynı zamanda normal dağılımda % 100 etkinliğe ulaşır. Bu tarz bir uyarlanabilir yeniden ağırlıklandırma yöntemi ilk defa, doğrusal regresyon modeli için 2002 yılında yine Gervini tarafından önerilmiştir [8]. Gervini [8]'de ise, bu yöntem geliştirilerek çok değişkenli konum ve kovaryans kestirimi için uyarlanabilir bir yöntem önerilmiştir.

$p' = p + 1$ olmak üzere $\Re^{p'}$'de verilen $\mathbf{z}_1, \cdots, \mathbf{z}_n$ örneklemi ile konum ve kovaryansın başlangıç sağlam kestiricileri $\left(\hat{\boldsymbol{\mu}}_{0n}, \hat{\boldsymbol{\Sigma}}_{0n}\right)$ için Mahalanobis uzaklıkları Eş. (4.3)'deki gibidir [8].

$$d_i := d\left(\mathbf{z}_i, \hat{\boldsymbol{\mu}}_{0n}, \hat{\boldsymbol{\Sigma}}_{0n}\right) = \left\{\left(\mathbf{z}_i - \hat{\boldsymbol{\mu}}_{0n}\right)' \hat{\boldsymbol{\Sigma}}_{0n}^{-1} \left(\mathbf{z}_i - \hat{\boldsymbol{\mu}}_{0n}\right)\right\}^{1/2} \tag{4.3}$$

Gervini [8], FAST-MCD algoritması ile hesaplanan MCD yönteminin MVE yöntemine iyi bir seçenek olarak geliştirildiği düşünüldüğünde, burada söz edilen 'uyarlanabilir yeniden ağırlıklandırılmış' yöntemde konum ve kovaryansın sağlam başlangıç kestiricileri olarak, MCD kestiricilerinin kullanılabileceğini belirtmiştir. Bir aykırı değer, beklendiği üzere 'iyi' olarak adlandırılan bir gözlemden daha büyük Mahalanobis uzaklığına sahip olacaktır. Normal dağılım varsayımı altında d_i^2 yaklaşık olarak $\chi^2_{p'}$ dağılır ve örneğin, $d_i^2 \geq \chi^2_{p',0.975}$ olan gözlemlerden aykırı değer olarak şüphelenilebilir. Rousseeuw ve van Zomeren tarafından 1990 yılında yayınlanan makalede, bu aykırı gözlemleri analizden çıkartıp geriye kalan verinin örneklem ortalaması ve kovaryans matrisinin bulunması ve böylece, $\left(\hat{\boldsymbol{\mu}}_{1n}, \hat{\boldsymbol{\Sigma}}_{1n}\right)$ şeklinde gösterilen yeni kestiricilerin elde edilmesi önerilmiştir [8].

Doktora tez çalışmamızda, burada söz edilen 'uyarlanabilir yeniden ağırlıklandırılmış' yöntemde konum ve kovaryansın sağlam başlangıç kestiricileri $\left(\hat{\boldsymbol{\mu}}_{0n}, \hat{\boldsymbol{\Sigma}}_{0n}\right)$ olarak MCD kestiricilerini $\left(\hat{\boldsymbol{\mu}}_{MCD}, \hat{\boldsymbol{\Sigma}}_{MCD}\right)$ kullanarak elde edilen $\left(\hat{\boldsymbol{\mu}}_{1n}, \hat{\boldsymbol{\Sigma}}_{1n}\right)$ sağlam konum ve kovaryans kestiricileri, 'Uyarlanabilir Yeniden Ağırlıklandırılmış En Küçük Kovaryans Determinantı (Adaptive Reweighted Minimum Covariance Determinant/ARWMCD)' kestiricileri $\left(\hat{\boldsymbol{\mu}}_{ARWMCD}, \hat{\boldsymbol{\Sigma}}_{ARWMCD}\right)$ olarak adlandırılmıştır.

Gervini [8] çalışmasında söz edilen yeniden ağırlıklandırma adımı ile başlangıç kestiricisinin sağlamlığının çoğu korunurken, aynı zamanda başlangıç kestiricisinin etkinliği arttırılır. Ancak, $\chi^2_{p',0.975}$ eşik değeri keyfi bir değerdir. Bu şekildeki keyfi eşik değerleri kullanıldığı için, özellikle gözlem sayısının değişken sayısından çok fazla olduğu (n>>p) daha büyük veri kümeleri için normal modele sahip olmalarına karşın çok sayıda gözlem analizden çıkartılmak zorunda olabilir. Bu problemden kurtulmanın bir yolu, eşik değerini başka bir keyfi sabit sayıya yükseltmektir. Ancak, bu durumda da yeniden ağırlıklandırılmış kestiricinin yanı etkilenecektir. Bu nedenle, daha iyi seçenek bir yöntem ise gözlem sayısı arttıkça veri kümesi aykırı değer içermeyen 'temiz' bir veri kümesi olduğunda artan, ancak örneklemde aykırı değerler olduğunda sınırlı kalan bir 'ayarlanabilir eşik değeri' kullanmaktır [8].

Gervini [8], böyle ayarlanabilir eşik değerleri oluşturmak için bir yöntem önermiştir. Eş. (4.4), karesi alınmış Mahalanobis uzaklıklarının deneysel dağılımını göstersin.

$$G_n(u) = \frac{1}{n}\sum_{i=1}^{n} I\left(d^2\left(\mathbf{z}_i, \hat{\boldsymbol{\mu}}_{MCD}, \hat{\boldsymbol{\Sigma}}_{MCD}\right) \le u\right) \tag{4.4}$$

$G_{p'}(u)$, $\chi^2_{p'}$ dağılım fonksiyonu olsun. Normal dağılımlı örneklem için, G_n'in $G_{p'}$'ye yakınsaması beklenir. Bu nedenle, aykırı değerleri belirlemenin bir yolu G_n'in kuyruklarını $G_{p'}$'nin kuyrukları ile karşılaştırmaktır. Eğer α=0.025 gibi belirli küçük bir α için $\eta = \chi^2_{p',1-\alpha}$ ise, Eş. (4.5) tanımlanır [8].

$$\alpha_n = \sup_{u \ge \eta} \left\{ G_{p'}(u) - G_n(u) \right\}^+ \tag{4.5}$$

Burada $\{\cdot\}^+$, pozitif kısımı gösterir. α_n, örneklemdeki aykırı değerlerin bir ölçüsü olarak kabul edilebilir. Eş. (4.5)'te, negatif bir fark aykırı değerlerin varlığını göstermeyeceğinden, sadece pozitif farklılıklar hesaba katılmıştır. Eğer $d^2_{(i)}$ karesi alınmış Mahalanobis uzaklıklarının i. sıralı istatistiğini gösteriyor ve $i_0 = \max\left\{i : d^2_{(i)} < \eta\right\}$ ise, Eş. (4.5) Eş. (4.6)'ya indirgenir [8].

$$\alpha_n = \max_{i > i_0} \left\{ G_{p'}\left(d^2_{(i)}\right) - \frac{i-1}{n} \right\}^+ \tag{4.6}$$

En büyük $\lfloor \alpha_n n \rfloor$ uzaklıklarına karşılık gelen gözlemler aykırı değerler olarak ele alınır ve yeniden ağırlıklandırma adımında çıkartılır. Burada $\lfloor a \rfloor$, a'ya eşit ya da daha küçük en büyük tamsayıyı gösterir. Buna göre, kesim değeri (cut-off value) Eş. (4.7)'deki gibi tanımlanır [8].

$$c_n = G_n^{-1}(1 - \alpha_n) \tag{4.7}$$

Eş. (4.7)'de $G_n^{-1}(u) = \min\{s : G_n(s) \geq u\}$'dır. $i_n = n - \lfloor \alpha_n n \rfloor$ olmak üzere $c_n = d^2_{(i_n)}$'dir ve α_n'nın tanımının sonucu olarak, $i_n > i_0$'dır. Bu nedenle, $c_n > \eta$'dır [8].

Yeniden ağırlıklandırılmış kestiriciyi tanımlamak için, Eş. (4.8)'deki biçimdeki ağırlıklar kullanılır [8].

$$w_{in} = w\left(\frac{d^2\left(\mathbf{z}_i, \hat{\boldsymbol{\mu}}_{MCD}, \hat{\boldsymbol{\Sigma}}_{MCD}\right)}{c_n} \right) \tag{4.8}$$

Buradaki ağırlık fonksiyonu şunları sağlar ***(W)*** $w : [0,\infty) \to [0,1]$ artan değil, $w(0) = 1$, $u \in [0,1)$ için $w(u) > 0$ ve $u \in [1,\infty)$ için $w(u) = 0$'dır. ***(W)***'yi sağlayan fonksiyonlar arasından en basit seçim, uygulamada en sık kullanılanlardan biri olan katı ret fonksiyonu (hard-rejection function) $w(u) = I(u < 1)$'dır [8].

Önce Eş. (4.8)'deki ağırlıklar hesaplanır, sonra bir-adım yeniden ağırlıklandırılmış $\left(\hat{\boldsymbol{\mu}}_{ARWMCD}, \hat{\boldsymbol{\Sigma}}_{ARWMCD}\right)$ kestiricileri, Eş. (4.9) ve Eş. (4.10)'daki gibi tanımlanır [8].

$$\hat{\boldsymbol{\mu}}_{ARWMCD} = \sum_{i=1}^{n} w_{in}\mathbf{z}_i \Big/ \sum_{i=1}^{n} w_{in} \tag{4.9}$$

$$\hat{\boldsymbol{\Sigma}}_{ARWMCD} = \sum_{i=1}^{n} w_{in}\left(\mathbf{z}_i - \hat{\boldsymbol{\mu}}_{ARWMCD}\right)\left(\mathbf{z}_i - \hat{\boldsymbol{\mu}}_{ARWMCD}\right)' \Big/ \sum_{i=1}^{n} w_{in} \tag{4.10}$$

Uygun koşullar altında, çok değişkenli normal dağılım atında Eş. (4.7)'deki eşik değerleri sonsuza gider ve o zaman, Eş. (4.9) ile Eş. (4.10) sırasıyla, klasik örneklem ortalaması ve kovaryansa asimptotik olarak eşit olur ve bu nedenle, tamamen asimptotik etkinliğe ulaşır [8].

Sonuç olarak, bu doktora tez çalışmasında ilk olarak Eş. (4.10)'daki $\hat{\mathbf{\Sigma}}_{ARWMCD}$ sağlam kovaryans kestiricisini kullanarak, $\mathbf{S}_z = \begin{pmatrix} \mathbf{s}_y^2 & \mathbf{s}'_{y,x} \\ \mathbf{s}_{y,x} & \mathbf{S}_x \end{pmatrix}$ kovaryansının sağlam kovaryans kestirimi $\hat{\mathbf{S}}_z$ elde edilmiştir. Daha sonra, sağlam kovaryans matrisi $\hat{\mathbf{S}}_z$'yi alt bölüm 2.2.2'de anlatılan ve sadece Eş. (2.9), Eş. (2.10) ve Eş. (2.11)'deki üç adımı kullanarak PLSR katsayılarının hesaplandığı PLS1 algoritmasının seçenek tanımında kullanarak ise, *PLS-ARWMCD* olarak adlandırılan yeni bir sağlam PLSR yöntemi önerilmiştir. Yeni önerilen sağlam *PLS-ARWMCD* algoritmasının adımları, Eş. (4.11)'deki gibidir.

$$
\begin{aligned}
&\mathbf{w}_1 \; \alpha \hat{\mathbf{s}}_{y,x} \\
&\mathbf{w}_{i+1} \; \alpha \; \hat{\mathbf{s}}_{y,x} - \hat{\mathbf{S}}_x \mathbf{W}_i \left(\mathbf{W}_i' \hat{\mathbf{S}}_x \mathbf{W}_i \right)^{-1} \mathbf{W}_i' \hat{\mathbf{s}}_{y,x} \, , 1 \le i < k \\
&\hat{\boldsymbol{\beta}}_k^{PLS-ARWMCD} = \mathbf{W}_k \left(\mathbf{W}_k' \hat{\mathbf{S}}_x \mathbf{W}_k \right)^{-1} \mathbf{W}_k' \hat{\mathbf{s}}_{y,x}
\end{aligned}
\qquad (4.11)
$$

Eş. (4.11)'deki, sağlam $\hat{\mathbf{s}}_{y,x}$ ve $\hat{\mathbf{S}}_X$ kovaryans kestirimleri $\mathbf{z}_i' = (y_i, \mathbf{x}_i)', i = 1, \cdots, n$ birleştirilmiş veri kümesinin ARWMCD kestiricisi ile hesaplanan sağlam kovaryans kestiriminin, $\hat{\mathbf{S}}_z = \begin{pmatrix} \hat{\mathbf{s}}_y^2 & \hat{\mathbf{s}}'_{y,x} \\ \hat{\mathbf{s}}_{y,x} & \hat{\mathbf{S}}_x \end{pmatrix}$ şeklinde ayrıştırılması ile elde edilmiştir.

4.2. Önerilen Sağlam PLSR Yöntemleri PLS-Smult ve PLS-MMmult

Bu alt bölümde de aynı Alt Bölüm 4.1'de olduğu gibi, S-kestiricileri ve MM-kestiricileri anlatılırken $p' = p + 1$ olmak üzere $\mathbf{z}_i = (y_i, \mathbf{x}_i), i = 1, \cdots, n \in \Re^{p'}$ örneklemi üzerinden eşitlikler incelenmiştir. Çünkü, daha sonra bu yöntemleri kullanarak **z**=(**y**,**X**) birleşik veri matrisinin kovaryans matrisi $\mathbf{S}_Z$ sağlam bir şekilde kestirilmiş ve iki tane yeni sağlam PLSR yöntemi, *PLS-Smult* ve *PLS-MMmult* önerilmiştir.

4.2.1. Çok Değişkenli Konum ve Kovaryans için S-kestiricileri ile MM-kestiricileri

1987 yılında Davies ile Rousseeuw ve Leroy [28] tarafından yapılan çalışmalarda ve 1989 yılında Lopuhaä tarafından yapılan makalede, çok değişkenli konum ve kovaryans için S-kestiricileri önerilmiştir. Çok değişkenli konum ve kovaryans için S-kestiricileri, % 50'ye yakın BDP ile çok sağlamdır ve MM-kestiricilerinin hesaplanmasında kullanılır. $\mathbf{C}_z$, bir $p' \times p'$ boyutlu simetrik pozitif tanımlı (symmetric positive definite/SPD) matris ve $m_z \in \Re^{p'}$ olsun. Buna göre, çok değişkenli konum ve kovaryans için bir S-kestiricisi kısaca şöyle tanımlanabilir: $\mathbf{z}_1, \cdots, \mathbf{z}_n \in \Re^{p'}$ örneklemi için bir S-kestiricisi, tüm $(\mathbf{m}_z, \mathbf{C}_z)$'ler için Eş. (4.12)'deki koşul altında, $|\mathbf{C}_z|$'yi en küçük yapan $(\tilde{\boldsymbol{\mu}}_z, \tilde{\boldsymbol{\Sigma}}_z)$ çifti olarak tanımlanır [17].

$$\frac{1}{n}\sum_{i=1}^{n} \rho\left(\sqrt{(\mathbf{z}_i - \mathbf{m}_z)' \mathbf{C}_z^{-1} (\mathbf{z}_i - \mathbf{m}_z)}\right) = b \tag{4.12}$$

Pozitif kırılma kestirimler elde etmek için, ρ -fonksiyonu aşağıdaki süreklilik koşullarını sağlamalıdır [17, 30]:

1. ρ-fonksiyonu; gerçel, sıfır etrafında simetrik ve iki kez sürekli olarak türevlenebilir olmalıdır.

2. Bir $c > 0$ için ρ, $[0,c]$ aralığında tam olarak artan, $[c,\infty)$ aralığında sabit ve $\rho(0) = 0$ olmalıdır.

ρ-fonksiyonu için genellikle, Eş. (4.13)'teki fonksiyon seçilir. Bu fonksiyondaki c, c>0 olan ve kullanıcı tarafından seçilen uygun bir ayarlama sabitidir. Eş. (4.13)'teki fonksiyonun türevi, 'Tukey'in çiftkare (bisquare) fonksiyonu' olarak bilinir ve Eş. (4.14)'teki gibi gösterilir [17, 30].

$$\rho(z)=\begin{cases}\dfrac{z^2}{2}-\dfrac{z^4}{2c^2}+\dfrac{z^6}{6c^4}, & |z|\le c\\ \dfrac{c^2}{6}, & |z|>c\end{cases} \tag{4.13}$$

$$\rho'(z)=\psi(z)=\begin{cases}z\left(1-\left(\dfrac{z}{c}\right)^2\right)^2, & |z|\le c\\ 0, & |z|>c\end{cases} \tag{4.14}$$

Lopuhaä ve Rousseeuw 1991 yılında yayınladıkları makalelerinde, çok değişkenli bir S-kestiricisinin BDP'sinin $\dfrac{b}{\rho(c)}$ olduğunu göstermiştir. Normal modelde tutarlılığı sağlamak için, $F_0=N(\mathbf{0},\mathbf{I_p})$ olmak üzere b sabiti $E_{F_0}\left[\rho(\|z\|)\right]$ biçiminde hesaplanabilir. Burada E_{F_0}, çok değişkenli normal dağılım F_0'a ilişkin beklenen değeri gösterir. Buna göre normal model altında b, Eş. (4.15)'teki gibi hesaplanır [17].

$$b=\frac{(p')}{2}\chi^2_{p'+2}(c^2)-\frac{p'(p'+2)}{2c^2}\chi^2_{p'+4}(c^2)+\frac{p'(p'+2)(p'+4)}{6c^4}\chi^2_{p'+6}(c^2)+\frac{c^2}{6}\left(1-\chi^2_{p'}(c^2)\right) \tag{4.15}$$

Burada $\chi^2_{p'}$, p′ serbestlik dereceli χ^2'nin birikimli dağılım fonksiyonudur. 1984 yılında Rousseeuw ve Yohai tarafından yapılan çalışmada, % 0 ve % 50 arasındaki BDP'lere ilişkin c ayarlama parametre değerleri verilmiştir. Örneğin, BDP'ler 0.50, 0.25, 0.20, 0.15 iken c'ler sırasıyla, 1.5476, 2.937, 3.42, 4.00'tür. [17, 30].

Salibian-Barrera ve Yohai'nin 2006 yılında regresyonun S-kestiricileri için geliştirdikleri FastS algoritmasını, aynı yıl Salibian-Barrera vd. [30] konum ve kovaryansın çok değişkenli S-kestiricilerine genişletmiştir [17].

4.2.2. Çok Değişkenli Konum ve Kovaryans için S-kestiricilerini Hesaplayan FastS Algoritması

İlk olarak, $|\mathbf{\Gamma}_z| = 1$ ve $\sigma_z = |\mathbf{\Sigma}_z|^{1/2p'}$ olmak üzere Eş. (4.12)'deki $\mathbf{C}_z$, $\sigma_z^2\mathbf{\Gamma}_z$ biçiminde yazılır. $\mathbf{m}_z \in \Re^{p'}$, $\mathbf{G}_z$ ise $|\mathbf{G}_z| = 1$ ile bir $p' \times p'$ boyutlu SPD matris ve s, bir pozitif skaler olsun. Bu nedenle eşdeğer bir amaç, tüm $(\mathbf{m}_z, \mathbf{G}_z, s)$'ler için Eş. (4.16)'daki kısıt altında s'yi en küçük yapan $(\tilde{\boldsymbol{\mu}}_z, \tilde{\mathbf{\Gamma}}_z, \tilde{\sigma}_z)$ üçlüsünü bulmaktır. Böylece, konum ve kovaryansın kestirimleri $(\tilde{\boldsymbol{\mu}}_z, \tilde{\sigma}_z^2\tilde{\mathbf{\Gamma}}_z)$ olur [17].

$$\frac{1}{n}\sum_{i=1}^{n}\rho\left(\frac{\sqrt{(\mathbf{z}_i - \mathbf{m}_z)'\mathbf{G}_z^{-1}(\mathbf{z}_i - \mathbf{m}_z)}}{s}\right) = b \tag{4.16}$$

Algoritma, determinantı sıfır olmayan bir kovaryansa sahip olan $p'+1$ boyutlu N tane rasgele alt kümenin çekilmesiyle ve *l*. alt kümenin klasik ortalaması $\tilde{\boldsymbol{\mu}}_l^{(0)}$ ile kovaryans matrisi $\tilde{\mathbf{\Sigma}}_l^{(0)}$'yı hesaplayarak elde edilen, N tane başlangıç kestirimi $(\tilde{\boldsymbol{\mu}}_1^{(0)}, \tilde{\mathbf{\Gamma}}_1^{(0)}, \tilde{\sigma}_1^{(0)}), \cdots, (\tilde{\boldsymbol{\mu}}_N^{(0)}, \tilde{\mathbf{\Gamma}}_N^{(0)}, \tilde{\sigma}_N^{(0)})$ ile başlar. $l = 1, \cdots, N$ için $\tilde{\mathbf{\Gamma}}_l^{(0)} = |\tilde{\mathbf{\Sigma}}_l^{(0)}|^{-1/p'}\tilde{\mathbf{\Sigma}}_l^{(0)}$ ve $\tilde{\sigma}_l^{(0)} = \text{ortanca}_{i=1}^{n}\sqrt{(\mathbf{z}_i - \tilde{\boldsymbol{\mu}}_l^{(0)})'(\tilde{\mathbf{\Gamma}}_l^{(0)})^{-1}(\mathbf{z}_i - \tilde{\boldsymbol{\mu}}_l^{(0)})}$ şeklinde elde edilir. Bundan sonra, k tane *l*-adımı yerine getirerek bu kestirimler yeniden iyileştirilir (refined) ve sonuç olarak, Eş. (4.17) elde edilir [17].

$$(\tilde{\boldsymbol{\mu}}_1^{(k)}, \tilde{\mathbf{\Gamma}}_1^{(k)}, \tilde{\sigma}_1^{(k)}), \cdots, (\tilde{\boldsymbol{\mu}}_N^{(k)}, \tilde{\mathbf{\Gamma}}_N^{(k)}, \tilde{\sigma}_N^{(k)}) \tag{4.17}$$

$(\tilde{\boldsymbol{\mu}}_l^{(j-1)}, \tilde{\mathbf{\Gamma}}_l^{(j-1)}, \tilde{\sigma}_l^{(j-1)})$ kestirimini iyileştirmek için J. *l*-adımı aşağıdaki gibi tanımlanır [17]:

1. Ölçek iyileştirilir: $\tilde{\sigma}_l^{(j)} = \tilde{\sigma}_l^{(j-1)}\sqrt{\frac{1}{nb}\sum_{i=1}^{n}\rho\left(\frac{(\mathbf{z}_i - \tilde{\boldsymbol{\mu}}_l^{(j-1)})'(\tilde{\mathbf{\Gamma}}_l^{(j-1)})^{-1}(\mathbf{z}_i - \tilde{\boldsymbol{\mu}}_l^{(j-1)})}{\tilde{\sigma}_l^{(j-1)}}\right)}$

2. $u = \frac{\sqrt{\left(\mathbf{z}_i - \tilde{\boldsymbol{\mu}}_l^{(j-1)}\right)' \left(\tilde{\boldsymbol{\Gamma}}_l^{(j-1)}\right)^{-1} \left(\mathbf{z}_i - \tilde{\boldsymbol{\mu}}_l^{(j-1)}\right)}}{\tilde{\sigma}_l^{(j)}}$ ile $w_i^{(j)} = \frac{\rho'(u)}{u}$ ağırlıklarını hesaplamak için $\tilde{\sigma}_l^{(j)}$ kullanılır.

3. $\tilde{\boldsymbol{\mu}}_l^{(j)}$ ağırlıklandırılmış ortalaması hesaplanır ve $\tilde{\boldsymbol{\Sigma}}_l^{(j)}$ ağırlıklandırılmış kovaryansı hesaplanır ve bu da, $\tilde{\boldsymbol{\Gamma}}_l^{(j)} = \left|\tilde{\boldsymbol{\Sigma}}_l^{(j)}\right|^{-1/p'} \tilde{\boldsymbol{\Sigma}}_l^{(j)}$ biçimindeki iyileştirmeye (refinement) neden olur.

k tane *l*-adımı yapıldıktan sonra, $\tilde{\boldsymbol{\mu}}_l^{(k)}$ ve $\tilde{\boldsymbol{\Gamma}}_l^{(k)}$'yı sabit tutarak, Eş. (4.18)'i yakınsayana kadar yinelemeli olarak çözerek, her bir $\left(\tilde{\boldsymbol{\mu}}_l^{(k)}, \tilde{\boldsymbol{\Gamma}}_l^{(k)}, \tilde{\sigma}_l^{(k)}\right)$ için $\tilde{\sigma}_l^{(k)}$ ölçeği düzeltilir [17].

$$\tilde{\sigma}_l^{(k+1)} = \tilde{\sigma}_l^{(k)} \sqrt{\frac{1}{nb} \sum_{i=1}^{n} \rho\left(\frac{\left(\mathbf{z}_i - \tilde{\boldsymbol{\mu}}_l^{(k)}\right)' \left(\tilde{\boldsymbol{\Gamma}}_l^{(k)}\right)^{-1} \left(\mathbf{z}_i - \tilde{\boldsymbol{\mu}}_l^{(k)}\right)}{\tilde{\sigma}_l^{(k)}} \right)} \tag{4.18}$$

En küçük tamamen yinelenmiş (iterated) ölçeklerin yer aldığı iyileştirilmiş, $\left(\tilde{\boldsymbol{\mu}}_1^{(B)}, \tilde{\boldsymbol{\Gamma}}_1^{(B)}, \tilde{\sigma}_1^{(B)}\right), \cdots, \left(\tilde{\boldsymbol{\mu}}_\nu^{(B)}, \tilde{\boldsymbol{\Gamma}}_\nu^{(B)}, \tilde{\sigma}_\nu^{(B)}\right)$ kestirimleri korunur. Tüm $\tilde{\sigma}_l^{(k)}, l = 1, \cdots, N$ kestiricilerinin, Eş. (4.18)'i çözerek hesaplanması gerekmemektedir. Her zaman, ilk $\tilde{\sigma}_l^{(k)}, l = 1, \cdots, \nu$ ölçek hesaplanır. Ancak, $l > \nu$ için *l*. ölçek sadece Eş. (4.19)'daki eşitsizlik sağlanır ise hesaplanır. Burada A, şimdiye kadar tamamen yinelenen ν tane ölçeğin maksimumudur [17].

$$\frac{1}{n} \sum_{i=1}^{n} \rho\left(\frac{\sqrt{\left(\mathbf{z}_i - \tilde{\boldsymbol{\mu}}_l^{(k)}\right)' \left(\tilde{\boldsymbol{\Gamma}}_l^{(k)}\right)^{-1} \left(\mathbf{z}_i - \tilde{\boldsymbol{\mu}}_l^{(k)}\right)}}{A} \right) < b \tag{4.19}$$

Bu düşünce ilk olarak, 1991 yılında Yohai ve Zamar tarafından geliştirilmiştir. Yukarıda tanımlandığı gibi *l*-adımı kullanarak, en küçük ölçeklere sahip ν tane

$\left(\tilde{\boldsymbol{\mu}}_1^{(B)}, \tilde{\boldsymbol{\Gamma}}_1^{(B)}, \tilde{\sigma}_1^{(B)}\right), \cdots, \left(\tilde{\boldsymbol{\mu}}_\nu^{(B)}, \tilde{\boldsymbol{\Gamma}}_\nu^{(B)}, \tilde{\sigma}_\nu^{(B)}\right)$ kestirimin, yakınsayana kadar yeniden iyileştirilmesi gereklidir ve $\left(\tilde{\boldsymbol{\mu}}^{(F)}, \tilde{\boldsymbol{\Gamma}}^{(F)}, \tilde{\sigma}^{(F)}\right)$ biçiminde gösterilen son kestirim (final estimate) ise tam iyileştirmeden sonra en küçük ölçeğe sahip olan kestirimdir. $\boldsymbol{\Sigma}$ kovaryans matrisi için en son kestirim ise, $\tilde{\boldsymbol{\Sigma}}^{(F)} = \left(\tilde{\sigma}^{(F)}\right)^2 \tilde{\boldsymbol{\Gamma}}^{(F)}$'dir. FastS algoritması Riani vd. [27] tarafından geliştirilen MATLAB FSDA Toolbox'da 'Smult' adı ile vardır [17].

MATLAB FSDA Toolbox'da '*Smult*' fonksiyonu ile hesaplanan ve bizim de tezimizde kullandığımız FastS algoritmasına ilişkin bazı varsayılan değerler vardır. Bu değerler algoritmayı daha hızlı işletmek için, Riani vd. [27] tarafından belirlenen değerlerdir. Buna göre, *Smult* fonksiyonunda varsayılan 'BDP' değeri 0.50'dir. 'N' alt küme sayısı, Hubert vd. [17] çalışmasından yola çıkılarak, 500 olarak seçilir. Her bir çekilen alt küme için, varsayılan 'iyileştirme yinelemelerinin (*I* adımları)' sayısı 3'tür. Her bir çekilen alt kümeye ilişkin bu iyileştirme adımlarının yakınsaması için varsayılan 'hoşgörü payı (tolerance)' 1e-6'dır. Alt kümelerden saklanması gereken varsayılan 'en iyi konum' sayısı 5'tir. En iyi alt kümeler için varsayılan 'iyileştirme yinelemelerinin (*I* adımları)' sayısı 50'dir. En iyi alt kümelere ilişkin iyileştirme adımlarının yakınsaması için varsayılan 'hoşgörü payı' 1e-8'dir. Her bir çekilen alt kümeye ve her bir en iyi alt kümeye ilişkin ölçeğin en küçük değerini bulmak için yapılan yineleme sürecine ilişkin varsayılan 'hoşgörü payı' 1e-7'dir [17].

Sonuç olarak, bu doktora tez çalışmasında ilk olarak Alt Bölüm 4.2.2'de bahsedilen FastS algoritması kullanılarak elde edilen $\tilde{\boldsymbol{\Sigma}}^{(F)}$ sağlam kovaryans kestiricisini kullanarak, $\mathbf{S_z} = \begin{pmatrix} \mathbf{s}_y^2 & \mathbf{s}'_{y,x} \\ \mathbf{s}_{y,x} & \mathbf{S_x} \end{pmatrix}$ kovaryansının sağlam kovaryans kestirimi $\tilde{\mathbf{S}}_z$ elde edilmiştir. Daha sonra, sağlam kovaryans matrisi $\tilde{\mathbf{S}}_z$'yi alt bölüm 2.2.2'de anlatılan ve sadece Eş. (2.9), Eş. (2.10) ve Eş. (2.11)'deki üç adımı kullanarak PLSR katsayılarının hesaplandığı PLS1 algoritmasının seçenek tanımında kullanarak, *PLS-Smult* olarak adlandırılan yeni bir sağlam PLSR yöntemi önerilmiştir. Yeni önerilen sağlam *PLS-Smult* algoritmasının adımları, Eş. (4.20)'deki gibidir.

$$\mathbf{w}_1 \propto \tilde{\mathbf{s}}_{y,x}$$
$$\mathbf{w}_{i+1} \propto \tilde{\mathbf{s}}_{y,x} - \tilde{\mathbf{S}}_x \mathbf{W}_i \left(\mathbf{W}_i' \tilde{\mathbf{S}}_x \mathbf{W}_i\right)^{-1} \mathbf{W}_i' \tilde{\mathbf{s}}_{y,x}, 1 \le i < k \quad (4.20)$$
$$\hat{\boldsymbol{\beta}}_k^{PLS-Smult} = \mathbf{W}_k \left(\mathbf{W}_k' \tilde{\mathbf{S}}_x \mathbf{W}_k\right)^{-1} \mathbf{W}_k' \tilde{\mathbf{s}}_{y,x}$$

Eş. (4.20)'deki, sağlam $\tilde{\mathbf{s}}_{y,x}$ ve $\tilde{\mathbf{S}}_x$ kovaryans kestirimleri $\mathbf{z}_i' = (y_i, \mathbf{x}_i)', i = 1, \cdots, n$ birleştirilmiş veri kümesinin S-kestiricisi ile hesaplanan sağlam kovaryans kestiriminin, $\tilde{\mathbf{S}}_z = \begin{pmatrix} \tilde{s}_y^2 & \tilde{\mathbf{s}}_{y,x}' \\ \tilde{\mathbf{s}}_{y,x} & \tilde{\mathbf{S}}_x \end{pmatrix}$ şeklinde ayrıştırılması ile elde edilmiştir.

4.2.3. Çok Değişkenli Konum ve Kovaryans için MM-kestiricilerini Hesaplayan FastMM Algoritması

S-kestiricileri, iyi sağlamlık özelliklerine sahip olmalarına karşın çok etkin değildir. Bu nedenle, bir S-kestiricisi genellikle çok değişkenli bir MM-kestiricisini hesaplamak için bir başlangıç kestiricisi olarak kullanılır. MM-kestiricileri, başlangıç S-kestiricisinin BDP'sini korur ancak, ondan daha yüksek bir etkinliğe sahiptir. Yohai 1987 yılında, regresyon için MM-kestiricilerini önermiştir. Tatsuoka ve Tyler ise 2000 yılında, bu kestiricilerin çok değişkenli karşılıklarını önermiştir. $n \ge p' + 1$ için $\mathbf{z}_1, \cdots, \mathbf{z}_n \subset \Re^{p'}$ olmak üzere $\left(\hat{\boldsymbol{\mu}}_z, \hat{\boldsymbol{\Sigma}}_z\right)$ ile gösterilen konum ve kovaryansın çok değişkenli bir MM-kestiricisi aşağıdaki biçimde iki adımda hesaplanır [17, 30]:

1. Bir ρ-fonksiyonu ρ_0 için Eş. (4.12)'deki gibi çok değişkenli bir S-kestiricisi hesaplanır. Bunun sonucunda, $\left(\tilde{\boldsymbol{\mu}}_z, \tilde{\boldsymbol{\Sigma}}_z\right)$ ve $\tilde{\sigma}_z = \left|\tilde{\boldsymbol{\Sigma}}_z\right|^{1/2p'}$ elde edilir.

2. $\mathbf{m}_z \in \Re^{p'}$, $\mathbf{G}_z$ ise $|\mathbf{G}_z| = 1$ ile bir $p' \times p'$ boyutlu SPD matris olmak üzere tüm $(\mathbf{m}_z, \mathbf{G}_z)$'ler için Eş. (4.21)'i en küçük yapan $\left(\hat{\boldsymbol{\mu}}_z, \hat{\boldsymbol{\Gamma}}_z\right)$ bulunur. Bunu yapmak için, $\left(\tilde{\boldsymbol{\mu}}, \tilde{\sigma}^{-2} \tilde{\boldsymbol{\Sigma}}\right)$'den başlayan yinelemeli olarak yeniden ağırlıklandırılmış en küçük kareler

adımları kullanılır, sonuçta $\left(\widehat{\boldsymbol{\mu}}_z, \widehat{\boldsymbol{\Gamma}}_z\right)$ elde edilir. Buradan da konum ve kovaryansın MM-kestiricileri sırasıyla, $\widehat{\boldsymbol{\mu}}_z$ ve $\widehat{\boldsymbol{\Sigma}}_z = \hat{\sigma}_z^2 \widehat{\boldsymbol{\Gamma}}_z$ olur.

$$\frac{1}{n}\sum_{i=1}^{n} \rho_1 \left(\frac{\sqrt{\left(\mathbf{z}_i - \mathbf{m}_z\right)' \mathbf{G}_z^{-1} \left(\mathbf{z}_i - \mathbf{m}_z\right)}}{\tilde{\sigma}_z} \right) \tag{4.21}$$

ρ_0 ve ρ_1 kayıp fonksiyonlarının her ikisi de, Alt Bölüm 4.2.1'de verilen iki koşulu sağlamalıdır. S-kestiricileri MM-kestiricilerinin özel bir biçimi olarak görülebilir. Çünkü, $\rho_1 = \rho_0$ olduğu zaman MM-kestiricisi, başlangıç S-kestiricisini verir. Salibian-Barrera vd. [30], tüm $s \in \Re$ için $\rho_1(s) \le \rho_0(s)$ ve $\rho_1(\infty) = \rho_0(\infty)$ geçerli olduğunda, MM-kestiricisinin sonlu örneklem BDP'sinin başlangıç S-kestiricisinin BDP'sine eşit ya da daha büyük olduğunu göstermiştir. Çok değişkenli konum ve kovaryansın MM-kestiricilerini hesaplayan FastMM algoritmasının MATLAB program kodları, '*MMmult*' adıyla MATLAB FSDA Toolbox'da mevcuttur [17, 30].

Çok değişkenli MM-kestiricileri, Tatsuoka ve Tyler tarafından 2000 yılında önerilen 'yardımcı ölçekli çok değişkenli M-kestiricileri' olarak adlandırılan geniş bir kestirici sınıfındandır. Bu yöntemdeki düşünce, ölçeği çok sağlam bir S-kestiricisi ile kestirerek, daha sonra konum ($\boldsymbol{\mu}$) ve şekli ($\boldsymbol{\Gamma}$) merkezi modelde daha iyi bir etkinlik verecek şekilde kestirmektir. Konum ve şekil kestirimleri, yardımcı ölçek ile aynı BDP'ye sahip olur. Konum ve kovaryansın MM-kestiricileri hesaplanırken, genellikle çok iyi bilinen Eş. (4.13)'teki gibi verilen 'Tukey'in çiftağırlık (biweight) fonksiyonları' ailesindeki kayıp fonksiyonlar kullanılır. Salibian-Barrera vd. [30], $p' \times p'$ boyutlu $\mathbf{G}_z$ matrisinden bir 'şekil matrisi' ya da 'şekil kestiricisi' olarak söz ettiklerinde $\mathbf{G}_z$'nin, $\mathbf{G}_z = \left|\mathbf{C}_z\right|^{-1/p'} \mathbf{C}_z$ biçiminde elde edilmesinden dolayı $\mathbf{C}_z$ ile gösterilen bir kovaryans matrisine (ya da kestiricisine) karşılık geldiğini belirtmiştir. Öyle ki, $\left|\mathbf{G}_z\right| = 1$ olmak üzere $\mathbf{G}_z$, bir SPD matristir [30].

MATLAB FSDA Toolbox'da *MMmult* fonksiyonunda kullanılan başlangıç S-kestiricisi *Smult* fonksiyonu ile hesaplanır. Burada kullanılan *Smult* fonksiyonu

için varsayılan değerler, Alt Bölüm 4.2.2'de verilenlerle aynıdır. Sadece farklı olarak hesaplama zamanını kısaltmak amacıyla burada varsayılan 'alt küme sayısı', 20 olarak belirlenmiştir [17].

Sonuç olarak, bu doktora tez çalışmasında ilk olarak Alt Bölüm 4.2.3'de bahsedilen FastMM algoritması kullanılarak elde edilen $\widehat{\mathbf{\Sigma}}_z$ sağlam kovaryans kestiricisini kullanarak, $\mathbf{S}_z = \begin{pmatrix} s_y^2 & \mathbf{s}'_{y,x} \\ \mathbf{s}_{y,x} & \mathbf{S}_x \end{pmatrix}$ kovaryansının sağlam kovaryans kestirimi $\widehat{\mathbf{S}}_z$ elde edilmiştir. Daha sonra, sağlam kovaryans matrisi $\widehat{\mathbf{S}}_z$'yi alt bölüm 2.2.2'de anlatılan ve sadece Eş. (2.9), Eş. (2.10) ve Eş. (2.11)'deki üç adımı kullanarak PLSR katsayılarının hesaplandığı PLS1 algoritmasının seçenek tanımında kullanarak ise, *PLS-MMmult* olarak adlandırılan yeni bir sağlam PLSR yöntemi önerilmiştir. Yeni önerilen sağlam *PLS-MMmult* algoritmasının adımları, Eş. (4.22)'deki gibidir.

$$
\begin{aligned}
&\mathbf{w}_1 \; \alpha \widehat{\mathbf{s}}_{y,x} \\
&\mathbf{w}_{i+1} \; \alpha \; \widehat{\mathbf{s}}_{y,x} - \widehat{\mathbf{S}}_x \mathbf{W}_i \left(\mathbf{W}_i' \widehat{\mathbf{S}}_x \mathbf{W}_i \right)^{-1} \mathbf{W}_i' \widehat{\mathbf{s}}_{y,x} \,, 1 \le i < k \\
&\hat{\boldsymbol{\beta}}_k^{PLS-MMmult} = \mathbf{W}_k \left(\mathbf{W}_k' \widehat{\mathbf{S}}_x \mathbf{W}_k \right)^{-1} \mathbf{W}_k' \widehat{\mathbf{s}}_{y,x}
\end{aligned}
\tag{4.22}
$$

Eş. (4.22)'deki, sağlam $\widehat{\mathbf{s}}_{y,x}$ ve $\widehat{\mathbf{S}}_x$ kovaryans kestirimleri $\mathbf{z}_i' = (y_i, \mathbf{x}_i)', i = 1, \cdots, n$ birleştirilmiş veri kümesinin MM-kestiricisi ile hesaplanan sağlam kovaryans kestiriminin, $\widehat{\mathbf{S}}_z = \begin{pmatrix} \widehat{s}_y^2 & \widehat{\mathbf{s}}'_{y,x} \\ \widehat{\mathbf{s}}_{y,x} & \widehat{\mathbf{S}}_x \end{pmatrix}$ şeklinde ayrıştırılması ile elde edilmiştir.

5. UYGULAMA

Tezin dördüncü bölümünde önerilen ve *PLS-ARWMCD*, *PLS-Smult*, *PLS-MMmult* şeklinde isimlendirilen üç yeni sağlam PLSR yöntemi elde edilirken, ilk önce Alt Bölüm 2.2.2'de verilen klasik PLS1 algoritmasının seçenek tanımındaki Eş. (2.1)'deki kovaryans matrisi, sırasıyla ' *ilk adımda konum ve kovaryansın sağlam başlangıç kestiricileri olarak En Küçük Kovaryans Determinantı kestiricilerini kullanan, uyarlanabilir yeniden ağırlıklandırılmış bir kovaryans kestiricisi* ', ' *S-kestiricileri* ' ve ' *MM-kestiricileri* ' kullanılarak sağlam bir şekilde kestirilir. Bu üç yöntemden herhangi birini kullanarak sağlam bir şekilde kestirilen $\mathbf{S}_{\mathbf{x}}$ ve $\mathbf{s}_{\mathbf{y},\mathbf{x}}$ kovaryansları, daha sonra Eş. (2.9) ve Eş. (2.10)'da kullanılarak sağlam ağırlıklar elde edilir ve en son olarak bu ağırlıklar, Eş. (2.11)'de kullanılarak sağlam PLSR regresyon katsayıları elde edilir.

Bu bölümde, yeni önerilen üç sağlam PLSR yöntemi etkinlik, veriye uyum ve kestirimdeki başarıları bakımından Alt Bölüm 2.2.1'de verilen dik yükler algoritması olarak da bilinen PLS1 algoritması ile hesaplanan klasik PLSR yöntemi ve ayrıca literatürde yer alan diğer dört sağlam PLSR yöntemi RSIMPLS, PRM, PLS-SD, PLS-KurSD ile karşılaştırılacaktır. Bu amaçla, benzetim çalışmaları ve gerçek bir veri kümesi üzerinde uygulamalar yapılacaktır. Modelde tek bir bağımlı değişken olduğunda NIPALS (PLS1) ile SIMPLS algoritmaları aynı sonucu vermektedir. Bu nedenle, uygulamada sadece PLS1 algoritmasına ilişkin sonuçlar verilmiştir.

5.1. Benzetim Çalışması

Bu alt bölümde Engelen vd. [5], González vd. [10], Serneels vd. [31] makalelerinden yola çıkılarak kurulan üç benzetim düzeninden söz edilecektir. Bu benzetim düzenlerinden, aşağıdaki alt bölümlerde detaylı olarak bahsedilecektir. İlk olarak, bu benzetim düzenlerinde yöntemleri karşılaştırırken kullanılan ölçütler tanımlanır. Yöntemlerin etkinliği Eş. (5.1)'deki gibi tanımlanan $\hat{\boldsymbol{\beta}}$ kestirilen regresyon parametrelerinin MSE'lerini hesaplayarak değerlendirilir. Burada $\hat{\boldsymbol{\beta}}_k^{(l)}$, l.

benzetimdeki k tane bileşene dayalı kestirilen parametreyi gösterir. MSE, regresyon parametresinin ne derecede doğru bir şekilde kestirildiğini gösterir. Bu nedenle amaç, sıfıra yakın bir MSE değeri elde etmektir [5].

$$MSE_k\left(\hat{\boldsymbol{\beta}}\right)=\frac{1}{m}\sum_{l=1}^{m}\left\|\hat{\boldsymbol{\beta}}_k^{(l)}-\boldsymbol{\beta}\right\|^2 \tag{5.1}$$

Yöntemlerin, aykırı olmayan düzgün gözlemlere ne kadar iyi uyduğu da araştırılmak istenir. Benzetim düzenleri yüzünden, aykırı olmayan gözlemlerin indisleri tam olarak bilinir ve bu gözlemler, G_r kümesinde depolanır. Daha sonra, Uyum İyiliği (Goodness-of-Fit/GOF) ölçütü Eş. (5.2)'deki gibi tanımlanır [5].

$$GOF_k=1-\frac{\underset{i\in G_r}{var}\left(r_i,k\right)}{\underset{i\in G_r}{var}\left(y_i\right)} \tag{5.2}$$

Burada $r_{i,k}$, k tane bileşen hesaplandığında i. gözlemin artığını gösterir. Amaç, 1'e yakın bir GOF değeri elde etmektir [5].

Yöntemlerin kestirim yeteneği ise, Hata Kareler Ortalamasının Karekökü (Root Mean Squared Error/RMSE) kullanılarak ölçülür. Bu ölçütü hesaplamak için ilk önce, n_t tane aykırı değer olmayan gözlemden oluşan temiz bir G_t test kümesi türetilir. Daha sonra, Eş. (5.3) hesaplanır. Burada $\hat{y}_{i,k}$, regresyon parametre kestirimleri modeli oluşturan n boyutlu (X, Y) çalışma kümesine dayalıyken ve modelde k tane bileşen kaldığında, test kümesindeki i gözleminin kestirilen y değeridir. Modelde kalacak ideal bileşen sayısı genellikle, RMSE değerinin en küçük olduğu bileşen sayısı 'k' olarak seçilir [5].

$$RMSE_k=\sqrt{\frac{1}{n_t}\sum_{i=1}^{n_t}\left(y_i-\hat{y}_{i,k}\right)^2} \tag{5.3}$$

Yeni önerilen sağlam *PLS-ARWMCD*, *PLS-Smult*, *PLS-MMmult* yöntemlerinde kullanılan sırasıyla FAST-MCD, FastS ve FastMM algoritmaları yeniden örnekleme algoritmalarıdır. Her üç algoritmada da, n gözlemli tüm veri kümesinden $p'+1$ gözlemli 'N' tane alt küme (alt örneklem) rasgele olarak seçilir. Tezin dördüncü bölümünde de bahsedildiği üzere bu algoritmalara ilişkin MATLAB LIBRA Toolbox ve MATLAB FSDA Toolbox'ta varsayılan alt küme sayıları mevcuttur. Doktora tezimizde, ilk olarak her üç algoritma için de alt küme sayısını N=500 olarak seçtik. Çünkü, özellikle FAST-MCD algoritması için olmak üzere, her üç algoritma için de bu alt küme sayısı literatürdeki makalelerde sıklıkla tercih edilmiştir. İkinci olarak, MATLAB FSDA Toolbox'ta MM-kestiricilerini hesaplayan '*MMmult'* fonksiyonunu hızlandırmak için, bu fonksiyonda kullanılan ve başlangıç S-kestiricisini hesaplayan '*Smult*' fonksiyonu için varsayılan alt küme sayısı N=20 olduğu için bu alt küme sayısı seçilmiştir. Sonuç olarak, çizelgelerde her iki alt küme sayısı için de sonuçlar verilmiştir. N=20 için sonuçlar çizelgelerde parantez içinde ve koyu olarak yazılmıştır. Ancak, her iki alt küme sayısı için de aynı sonuçlar elde edildiğinde ya da önemsenmeyecek kadar küçük farklar olduğunda, bu değerler çizelgelerde yazılmamıştır. Böylece, seçilen alt küme sayılarının yeni önerilen üç sağlam PLSR yönteminin etkinlik, veriye uyum ve kestirimdeki başarılarını ne kadar etkilediği incelenmiştir. Ayrıca, alt küme sayısını az ya da daha fazla seçmenin yeni önerilen üç sağlam PLSR yönteminin hızını nasıl etkilediği araştırılmış ve yeni önerilen üç sağlam yöntem literatürde var olan dört sağlam yöntem ile hız bakımından da karşılaştırılmıştır.

5.1.1. Birinci Benzetim Düzeni

Engelen vd. [5] benzetim çalışmasından yola çıkılarak oluşturulan benzetim düzenleri için modeller, Eş. (5.4)'deki gibi verilmiştir. Burada, $\mathbf{0}_2 = (0,0)'$, $i = j$ için $(\mathbf{I}_{k,p})_{i,j} = 1$ ve $i \neq j$ için $(\mathbf{I}_{k,p})_{i,j} = 0$, $\mathbf{A}_{2,1} = (1,1)'$, $\mathbf{I}_p$ $p \times p$ boyutlu birim matris ve $\mathbf{T}$ $n \times 2$ boyutlu bileşen matrisidir. Benzetim düzenleri için, yöntemlerin etkinlikteki performanslarını karşılaştırmak için hesaplanan MSE ölçütündeki gerçek parametre vektörü ise $\boldsymbol{\beta}_{p\times 1} = \mathbf{I}'_{p\times 2}\mathbf{A}_{2\times 1}$ şeklinde belirlenmiştir.

$$\begin{aligned}&\mathbf{T} \sim \mathrm{N}_2\left(\mathbf{0}_2, \mathbf{\Sigma}_t\right)\\&\mathbf{X}=\mathbf{T}\,\mathbf{I}_{2,\mathrm{p}}+\mathrm{N}_{\mathrm{p}}\left(\mathbf{0}_{\mathrm{p}}, 0.1\,\mathbf{I}_{\mathrm{p}}\right)\\&\mathbf{y}=\mathbf{T}\mathbf{A}_{2,1}+\mathrm{N}(0,1)\end{aligned} \tag{5.4}$$

Eş. (5.4)'e göre modeller oluşturulduktan sonra, gözlemlerin ilk % 10'u aşağıda söz edilen aykırı değer türleri ile değiştirilerek veri kümesi aykırı değerler ile bozulmuştur. $\in$, verinin aykırı değerler tarafından bozulan yüzdesini temsil eder. Buna göre $\mathbf{T}_{\in}$, $\mathbf{X}_{\in}$ ve $\mathbf{Y}_{\in}$ verideki aykırı değerlerce bozulmuş kısımları gösterir. Burada ilk olarak aykırı değerlerin olmadığı temiz veri kümesi için, daha sonra kötü kaldıraç gözlemleri ve dikey aykırı değerler türetilerek bozulan aynı düzen için m=1000 tane veri kümesi üretilmiş ve k=1, 2, 3 tane bileşen içeren her bir model için analiz yapılmıştır. Ayrıca, aynı düzende yöntemlerin kestirim yeteneğini belirlemek amacıyla her bir veri kümesi için Eş. (5.4)'deki modele göre n_t=50 gözlemli temiz bir veri kümesi de üretilmiştir.

1. Kötü Kaldıraç Gözlemlerinin Oluşturulması: Eş. (5.4)'deki **X** için yazılan modelde gözlemlerin ilk % 10'u için $\mathbf{T}_{\in} \sim \mathrm{N}_2\left((15,15)', \mathbf{\Sigma}_t\right)$ yerine koyularak kötü kaldıraç gözlemleri oluşturulur: $\mathbf{X}_{\in}=\mathbf{T}_{\in}\,\mathbf{I}_{2,\mathrm{p}}+\mathrm{N}_{\mathrm{p}}\left(\mathbf{0}_{\mathrm{p}}, 0.1\mathbf{I}_{\mathrm{p}}\right)$. Ancak, bu gözlemlere ilişkin bağımlı değişken değerleri değiştirilmemiştir.

2. Dikey Aykırı Değerlerin Oluşturulması: Eş. (5.4)'deki gözlemlerin ilk % 10'u için **y**'ye ilişkin hata terimini ayarlayarak bağımlı değişken değerlerini değiştirerek, dikey aykırı değerler oluşturulur: $\mathbf{Y}_{\in}=\mathbf{T}\mathbf{A}_{2,1}+\mathrm{N}(15,0.1)$. Ancak, bu gözlemlere ilişkin bağımsız değişken değerleri bozulmamıştır.

Bu benzetim düzenini kullanarak yeni önerilen üç sağlam PLSR yöntemi *PLS-ARWMCD, PLS-Smult* ve *PLS-MMmult*'u, literatürde var olan dört sağlam PLSR yöntemi ve klasik PLSR yöntemi ile karşılaştırmak için iki farklı n, p ve $\mathbf{\Sigma}_t$ değerleri seçilmiştir. Engelen vd. [5] çalışmasından yola çıkılarak $\mathbf{\Sigma}_t=\begin{pmatrix}4 & 0\\0 & 2\end{pmatrix}$,

n=100 ve p=6 olarak alınan ilk benzetim düzeni için türetilmiş temiz veri, gözlemlerin ilk % 10'nun kötü kaldıraç gözlemleri ve dikey aykırı değerler ile yer değiştirdiği veri kümeleri için sonuçlar sırasıyla Çizelge 5.1, Çizelge 5.2 ve Çizelge 5.3'te verilmiştir.

Çizelge 5.1 incelendiğinde, temiz veri kümesi için klasik PLSR yöntemine ilişkin elde edilen sonuçlar, literatürde daha önce önerilen ve bizim önerdiğimiz sağlam PLSR yöntemleri ile karşılaştırıldığında, modelde k=2 ya da k=3 bileşen olduğunda klasik yöntemin sağlam yöntemlerden etkinlik bakımından daha başarılı olduğunu, veriye uyum ve kestirim açısından ise sağlam yöntemler ile yakın bir performansa sahip olduğunu söyleyebiliriz. Modelde k=1 bileşen olduğunda ise, literatürde var olan sağlam RSIMPLS yöntemi etkinlik bakımından kısmen daha iyi olmasına karşın, yeni önerdiğimiz üç yöntemin de dahil olduğu diğer tüm sağlam PLSR yöntemlerinin klasik PLSR yönteminin etkinlik, veriye uyum ve kestirimdeki başarısını yakaladığını söyleyebiliriz.

Genel olarak, veri kümesi temiz olduğu için klasik PLSR yönteminin sağlam yöntemlerden iyi sonuçlar vermesi karşılaşılan bir durumdur. Ancak, temiz veri kümesi için sağlam yöntemlerin klasik yöntemin performansını yakalaması da arzu edilir. Temiz veri kümesi için, yeni önerilen sağlam *PLS-Smult* ile *PLS-MMmult* yöntemleri, alt küme sayıları N=20 ya da N=500 seçildiğinde aynı sonuçları vermiştir. *PLS-ARWMCD* yönteminin ise kısmen farklı olmasına karşın, her iki alt küme sayısı için de çok yakın sonuçlar verdiği görülür.

Çizelge 5.1. n=100 ve p=6, temiz veri kümesi, m=1000 tekrar için benzetim sonuçları.

Modeldeki Bileşen Sayısı		PLSR	RSIMPLS	PRM	PLS-SD	PLS-KurSD	***PLS-ARWMCD***	***PLS-Smult***	***PLS-MMmult***
k=1	MSE	0.2319	0.2232	0.2367	0.2334	0.2349	0.2376 **(0.2379)**	0.2342	0.2325
	GOF	0.7723	0.7738	0.7709	0.7717	0.7712	0.7707 **(0.7706)**	0.7716	0.7721
	RMSE	1.2669	1.2607	1.2698	1.2676	1.2676	1.2697 (**1.2698)**	1.2682	1.2672
k=2	MSE	0.0167	0.0221	0.0183	0.0202	0.0219	0.0234 **(0.0227)**	0.0211	0.0179
	GOF	0.8301	0.8292	0.8293	0.8293	0.8292	0.8286 **(0.8287)**	0.8294	0.8298
	RMSE	1.0993	1.1020	1.1004	1.1015	1.1017	1.1036 **(1.1032)**	1.1010	1.0998
k=3	MSE	0.4977	0.5255	0.5313	0.5895	0.6165	0.6999 **(0.6800)**	0.5756	0.5263
	GOF	0.8361	0.8333	0.8344	0.8338	0.8333	0.8314 **(0.8318)**	0.8341	0.8353
	RMSE	1.1214	1.1254	1.1238	1.1282	1.1300	1.1361 **(0.1348)**	1.1272	1.1235

Çizelge 5.2 incelendiğinde, veri kümesinde kötü kaldıraç gözlemlerinin varlığında modeldeki bileşen sayısı 1, 2 ya da 3 olduğu her üç model için de klasik PLSR yönteminin sağlam yöntemlere karşın etkinlik, veriye uyum ve kestirimdeki başarısızlığı açık bir şekilde görülür. Klasik PLSR yöntemi için regresyon parametre kestirimlerinin MSE değerleri, modeldeki bileşen sayısı arttıkça artar ve bu nedenle, k=1 olan modelde en küçük değerlerine sahiptir. Her üç bileşen sayısı için de GOF değerleri incelendiğinde, beklendiği üzere klasik yöntemin GOF değerleri çok düşüktür. Bu da, klasik PLSR yöntemi için temiz verinin modele kötü bir şekilde uyduğunu gösterir. Yeni önerdiğimiz üç sağlam PLSR yöntemini incelediğimizde ise, modelde k=1 bileşen olduğunda *PLS-ARWMCD*, *PLS-Smult* ve *PLS-MMmult* yöntemlerinin üçünün de özellikle etkinlik ve kestirim açısından literatürde var olan sağlam PRM ve PLS-SD yöntemlerinden daha başarılı olduğunu söyleyebiliriz. Modelde k=2 bileşen olduğunda, yeni önerdiğimiz üç sağlam PLSR yöntemi de PLS-SD ile PRM yöntemlerinden etkinlik ve kestirim açısından daha başarılıdır. k=2 bileşenli model için yeni önerilen üç sağlam PLSR yöntemi de, özellikle etkinlikteki başarıları ile öne çıkan yöntemler olmuştur. Modelde k=3 bileşen olduğunda ise, yeni önerilen sağlam *PLS-MMmult* yönteminin

etkinlik açısından hem klasik yöntemden hem de diğer sağlam yöntemlerden daha başarılı olduğunu söyleyebiliriz. k=3 bileşenli model için, yeni önerilen sağlam *PLS-ARWMCD*, *PLS-Smult* ve *PLS-MMmult* yöntemlerinin üçü de literatürde var olan sağlam PRM, PLS-SD ve PLS-KurSD yöntemlerinden etkinlik ve kestirim açısından daha başarılı bulunmuştur. Sonuç olarak, bu benzetim düzeni için veri kümesinde %10 oranında kötü kaldıraç gözlemleri olduğunda yeni önerilen üç sağlam PLSR yönteminin de etkinlik ve kestirim açısından literatürde var olan PRM ve PLS-SD sağlam yöntemlerine karşın üstünlüğü açık bir şekilde görülür. Çizelge 5.2'den görüldüğü üzere, veri kümesinde % 10 oranında kötü kaldıraç gözlemleri olduğunda, N=20 ve N=500 alt küme sayıları için *PLS-Smult* ile *PLS-MMmult* yöntemlerinin verdiği sonuçlar tamamen aynıdır. Bu alt küme sayıları için *PLS-ARWMCD* yöntemi için de, önemsenmeyecek kadar küçük bir farkla benzer sonuçlar elde edilmiştir.

Çizelge 5.2. n=100 ve p=6, gözlemlerin ilk %10'u kötü kaldıraç gözlemleri, m=1000 tekrar için benzetim sonuçları.

Modeldeki Bileşen Sayısı		PLSR	RSIMPLS	PRM	PLS-SD	PLS-KurSD	***PLS-ARWMCD***	***PLS-Smult***	***PLS-MMmult***
k=1	MSE	1.7384	0.2203	0.2897	0.3081	0.2488	0.2445 **(0.2449)**	0.2450	0.2438
	GOF	0.1301	0.7756	0.7616	0.7568	0.7690	0.7703 **(0.7701)**	0.7702	0.7705
	RMSE	2.4541	1.2603	1.3005	1.3134	1.2798	1.2768 **(1.2773)**	1.2771	1.2765
k=2	MSE	1.9761	0.0241	0.0500	0.1249	0.0242	0.0214	0.0212	0.0196
	GOF	0.1840	0.8286	0.8233	0.8035	0.8284	0.8287	0.8289	0.8292
	RMSE	2.4050	1.1009	1.1178	1.1816	1.1008	1.0998	1.0993	1.0984
k=3	MSE	4.4312	0.6004	2.1928	0.8670	0.7326	0.6579 **(0.6584)**	0.6205	0.5842
	GOF	0.2126	0.8337	0.6081	0.8070	0.8326	0.8339 **(0.8341)**	0.8347	0.8356
	RMSE	2.4594	1.1278	1.6798	1.2205	1.1338	1.1294 **(1.1293)**	1.1269	1.1245

Çizelge 5.3 incelendiğinde, veri kümesinde dikey aykırı değerlerin varlığında modeldeki bileşen sayısı 1, 2 ya da 3 olduğu her üç model için de etkinlik, veriye uyum ve kestirim bakımından klasik PLSR yöntemi yeni önerilen üç sağlam yöntem *PLS-ARWMCD*, *PLS-Smult* ve *PLS-MMmult*'a göre daha başarısızdır. Özellikle k=3 bileşenli model incelendiğinde, klasik yönteme ilişkin MSE değerinin (8.6689), yeni önerdiğimiz üç sağlam yöntem ve literatürde var olan dört sağlam yöntemin MSE değerlerinden çok büyük olduğu görülür. Modelde k=1 bileşen olduğunda, en etkin ve kestirim bakımından en başarılı yöntem literatürde var olan sağlam RSIMPLS yöntemidir. k=1 bileşenli model için yeni önerilen üç sağlam PLSR yöntemi de literatürde var olan diğer üç sağlam yöntem ile çok yakın bir performansa sahiptir. Modelde k=2 bileşen olduğunda, literatürde var olan dört sağlam yöntem ve yeni önerdiğimiz üç sağlam yöntem çok yakın bir başarı gösterir. k=3 bileşenli model incelendiğinde ise, RSIMPLS yönteminden sonra en etkin yöntemlerin sırasıyla, yeni önerilen sağlam *PLS-MMmult* ve *PLS-Smult* yöntemleri olduğu görülür. Ayrıca, k=3 bileşenli model için RSIMPLS, *PLS-MMmult* ve *PLS-Smult* yöntemleri, kestirim açısından diğer dört sağlam yöntemden daha başarılıdır. Çizelge 5.3'ten görüldüğü üzere, veri kümesinde % 10 oranında dikey aykırı değerler olduğunda, N=20 ve N=500 alt küme sayıları için *PLS-Smult* ile *PLS-MMmult* yöntemlerinin verdiği sonuçlar tamamen aynıdır. *PLS-ARWMCD* yöntemi için de, önemsenmeyecek kadar küçük bir farkla benzer sonuçlar elde edilmiştir.

Çizelge 5.3. n=100 ve p=6, gözlemlerin ilk %10'u dikey aykırı değerler, m=1000 tekrar için benzetim sonuçları.

Modeldeki Bileşen Sayısı		PLSR	RSIMPLS	PRM	PLS-SD	PLS-KurSD	***PLS-ARWMCD***	***PLS-Smult***	***PLS-MMmult***
k=1	MSE	0.3121	0.2135	0.2356	0.2416	0.2431	0.2414 **(0.2415)**	0.2403	0.2389
	GOF	0.7344	0.7767	0.7713	0.7696	0.7695	0.7700	0.7703	0.7707
	RMSE	1.3542	1.2543	1.2704	1.2741	1.2758	1.2742 **(1.2744)**	1.2732	1.2722
k=2	MSE	0.3086	0.0249	0.0222	0.0238	0.0239	0.0229 **(0.0222)**	0.0215	0.0202
	GOF	0.7703	0.8290	0.8286	0.8291	0.8289	0.8293 **(0.8294)**	0.8296	0.8299
	RMSE	1.2672	1.1044	1.1054	1.1045	1.1047	1.1034 **(0.1035)**	1.1029	1.1018
k=3	MSE	8.6689	0.5002	0.6439	0.6746	0.6822	0.6317 **(0.6305)**	0.5949	0.5568
	GOF	0.6710	0.8333	0.8333	0.8337	0.8330	0.8342 **(0.8344)**	0.8351	0.8359
	RMSE	1.5603	1.1268	1.1341	1.1343	1.1351	1.1316 **(1.1317)**	1.1291	1.1266

$\boldsymbol{\Sigma}_t = \begin{pmatrix} 8 & 0 \\ 0 & 2 \end{pmatrix}$, n=50 ve p=10 olarak alınan ikinci benzetim düzeni için türetilmiş temiz veri, gözlemlerin ilk % 10'nun kötü kaldıraç gözlemleri ve dikey aykırı değerler ile yer değiştirdiği veri kümeleri için sonuçlar sırasıyla EK 1, EK 2 ve EK 3'teki çizelgelerde verilmiştir. Böylece farklı bir örneklem şeması kullanılarak, yeni önerilen üç sağlam PLSR yöntemi karşılaştırılmış olur.

EK 1'deki çizelge incelendiğinde, temiz veri kümesi için modelde k=1 bileşen olduğunda, literatürde var olan sağlam RSIMPLS yöntemi etkinlik ve kestirim bakımından kısmen daha başarılı olmasına karşın, yeni önerdiğimiz üç yöntemin de dahil olduğu diğer tüm sağlam PLSR yöntemleri klasik PLSR yönteminin etkinlik, veriye uyum ve kestirimdeki başarısını yakalar. Modelde k=2 bileşen olduğunda, literatürde var olan sağlam PRM yöntemi ile yeni önerilen sağlam *PLS-Smult* ile *PLS-MMmult* yöntemlerinin klasik yöntemin etkinlik ve kestirimdeki başarısını yakaladığını söyleyebiliriz. Modelde k=3 bileşen olduğunda ise, RSIMPLS ve PRM yöntemleri etkinlik bakımından klasik PLSR yönteminden daha

iyi ve kestirim açısından klasik yönteme yakın bir başarıya sahip olmalarına karşın, klasik yöntemin yeni önerdiğimiz üç yöntemin de dahil olduğu diğer tüm sağlam PLSR yöntemlerinden etkinlik ve kestirim açısından daha başarılı olduğunu söyleyebiliriz.

EK 2'teki çizelge incelendiğinde, veri kümesinde kötü kaldıraç gözlemlerinin varlığında modeldeki bileşen sayısı 1, 2 ya da 3 olduğu her üç model için de klasik PLSR yöntemi sağlam yöntemlere karşın etkinlik, veriye uyum ve kestirim açısından başarısızdır. Yeni önerdiğimiz üç sağlam PLSR yöntemini incelediğimizde ise, modelde k=1 bileşen olduğunda *PLS-ARWMCD*, *PLS-Smult* ve *PLS-MMmult* yöntemlerinin üçünün de etkinlik, veriye uyum ve kestirim açısından literatürde var olan sağlam PRM, PLS-SD ve PLS-KurSD yöntemlerinden daha başarılı olduğunu söyleyebiliriz. Modelde k=2 bileşen olduğunda, yeni önerdiğimiz üç sağlam PLSR yöntemi de PRM, PLS-SD ve PLS-KurSD yöntemlerinden etkinlik ve kestirim açısından daha başarılıdır. Modelde k=3 bileşen olduğunda ise, yeni önerilen sağlam *PLS-ARWMCD*, *PLS-Smult* ve *PLS-MMmult* yöntemlerinin üçü de PRM, PLS-SD ve PLS-KurSD yöntemlerinden etkinlik, veriye uyum ve kestirim açısından daha başarılıdır. Sonuç olarak, kötü kaldıraç gözlemlerinin varlığında yeni önerilen üç sağlam PLSR yöntemi de, özellikle etkinlik ve kestirim açısından literatürde var olan sağlam PRM, PLS-SD ve PLS-KurSD yöntemlerinden daha iyi sonuçlar vermiştir.

EK 3'teki çizelge incelendiğinde, veri kümesinde dikey aykırı değerlerin varlığında modeldeki bileşen sayısı 1, 2 ya da 3 olduğu her üç model için de etkinlik, veriye uyum ve kestirim bakımından klasik PLSR yöntemi yeni önerilen üç sağlam PLSR yöntemi *PLS-ARWMCD*, *PLS-Smult* ve *PLS-MMmult* ile literatürde var olan dört sağlam PLSR yöntemine göre daha başarısızdır. Modelde k=1 bileşen olduğunda, en başarılı yöntem literatürde var olan sağlam RSIMPLS yöntemidir ve yeni önerilen üç sağlam PLSR yöntemi de literatürde var olan diğer üç sağlam PLSR yöntemi ile yakın bir performansa sahiptir. Modelde k=2 bileşen olduğunda, yeni önerdiğimiz sağlam *PLS-ARWMCD*, *PLS-Smult* ve *PLS-MMmult* yöntemleri literatürde var olan sağlam PLS-KurSD yönteminden etkinlik, veriye uyum ve

kestirim açısından daha başarılıdır. k=2 bileşenli model için, yeni önerilen sağlam *PLS-Smult* ile *PLS-MMmult* yöntemleri ve literatürde var olan sağlam RSIMPLS ile PRM yöntemleri özellikle etkinlik ve kestirimdeki başarıları ile öne çıkmıştır. Modelde k=3 bileşen olduğunda, yeni önerilen sağlam *PLS-ARWMCD, PLS-Smult* ile *PLS-MMmult* yöntemleri, etkinlik ve kestirim açısından literatürde var olan sağlam PLS-SD ile PLS-KurSD yöntemlerinden daha başarılıdır.

Genel olarak EK 1, EK 2 ve EK 3'teki sonuçlar incelendiğinde, temiz veri kümesi ve kötü kaldıraç gözlemleri tarafından bozulan veri kümeleri için alt küme sayılarının N=20 ya da N=500 olarak seçilmesinin *PLS-Smult* ile *PLS-MMmult* yöntemlerinin sonuçlarını nerdeyse hiç etkilemediğini söyleyebiliriz. Ancak, veri kümesinde dikey aykırı değerler varlığında ve modeldeki bileşen sayısı arttığında *PLS-Smult* ile *PLS-MMmult* yöntemlerinin alt küme sayıları N=500 olarak seçildiğinde, bu iki yöntemin etkinlik açısından daha iyi sonuçlar verdiği görülür. Temiz veri kümesi, kötü kaldıraç gözlemleri ya da dikey aykırı değerler tarafından bozulan veri kümeleri incelendiğinde, genel olarak, *PLS-ARWMCD* yöntemi için seçilen alt küme sayısının az ya da çok olmasının sonuçları kısmen etkilediği, ancak, modeldeki bileşen sayısı arttıkça seçilen alt küme sayısının az olmasının etkinliği nispeten arttırdığı görülür.

RMSE ile GOF ölçütlerinin her ikisi de, modelde kalacak ideal bileşen sayısını (k_{opt}) seçmek için ideal bir ölçüt gibi görülür [5]. Bu benzetim çalışmasında klasik yöntem ve sağlam yöntemlere ilişkin GOF_3-GOF_2 farkları, GOF_2-GOF_1 farkları ile karşılaştırıldığında daha küçüktür. Ancak, Engelen vd. [5] çalışmasında da belirtildiği üzere, 'GOF_k'nın en büyük olduğu bileşen sayısı k_{opt} olarak seçilir' sonucuna ulaşılamaz. Her iki benzetim düzeni için de RMSE değerlerine bakıldığında ise, temiz veri kümesi, kötü kaldıraç gözlemleri ve dikey aykırı değerler tarafından bozulan veri kümeleri için, modelde k=2 bileşen olduğunda en küçük RMSE değerleri elde edilir. Bu nedenle, literatürdeki çalışmalardan yola çıkılarak $RMSE_k$'nın en küçük olduğu bileşen sayısının ideal bileşen sayısı olarak seçildiği bilindiğinden, k_{opt}=2 olarak bulunur. Buna göre Çizelge 5.2 ile Çizelge 5.3

birlikte incelendiğinde; n=100, p=6 ve $\mathbf{\Sigma}_t = \begin{pmatrix} 4 & 0 \\ 0 & 2 \end{pmatrix}$ olan veri kümesi % 10 oranında kötü kaldıraç gözlemleri ya da dikey aykırı değerler tarafından bozulduğunda ve modelde kalacak ideal bileşen sayısı k_{opt}=2 olarak seçildiğinde, sırasıyla yeni önerdiğimiz sağlam *PLS-MMmult*, *PLS-Smult* ve *PLS-ARWMCD* yöntemlerinin en etkin sağlam yöntemlerden olduğu görülmüştür. Ayrıca, veri kümesinde kötü kaldıraç gözlemleri olduğunda k_{opt}=2 için yeni önerdiğimiz üç sağlam PLSR yöntemi de, literatürde var olan sağlam PRM ve PLS-SD yöntemlerinden kestirim açısından daha başarılıdır. EK 2 ile EK 3'teki çizelgeler birlikte incelendiğinde; n=50, p=10 ve $\mathbf{\Sigma}_t = \begin{pmatrix} 8 & 0 \\ 0 & 2 \end{pmatrix}$ olan veri kümesi % 10 oranında kötü kaldıraç gözlemleri ya da dikey aykırı değerler tarafından bozulduğunda ve k_{opt}=2 olarak seçildiğinde, sırasıyla yeni önerdiğimiz sağlam *PLS-MMmult*, *PLS-Smult* ve *PLS-ARWMCD* yöntemlerinin en etkin sağlam yöntemlerden olduğu görülmüştür. Bu benzetim düzeni için veri kümesinde kötü kaldıraç gözlemleri olduğunda ve k_{opt}=2 olarak belirlendiğinde, yeni önerdiğimiz üç sağlam PLSR yöntemi de literatürde var olan sağlam PRM, PLS-SD ve PLS-KurSD yöntemlerinden kestirim açısından daha başarılı bulunmuştur. Ayrıca, bu benzetim düzeni için dikey aykırı değerlerin varlığında k_{opt}=2 olarak seçildiğinde ise, yeni önerdiğimiz üç sağlam PLSR yöntemi de literatürde var olan sağlam PLS-KurSD yönteminden kestirim açısından çok daha başarılıdır.

5.1.2. İkinci Benzetim Düzeni

González vd. [10] benzetim çalışmasından yola çıkılarak oluşturulan benzetim düzeninde, bir önceki alt bölümdeki benzetim düzenine benzer bir düzen kullanmış ve $\mathbf{\Sigma}_t = \begin{pmatrix} 4 & 0 \\ 0 & 2 \end{pmatrix}$ olarak almıştır. Bu benzetim düzenindeki temiz veri kümeleri de, Eş. (5.4)'deki modelleri kullanarak türetilmiştir. Bu benzetim düzeninin farkı, beş tane aykırı değer türü tanımlanmıştır. Ayrıca, ideal bileşen sayısı k_{opt}=2 olarak alınıp, sadece bu bileşen sayısı için modeller oluşturulmuş ve test kümesi n_t=100 gözlemli olarak belirlenmiştir. Ayrıca, bu benzetim düzeninde bir önceki benzetim çalışmasındaki ölçütlere ilave olarak gerçek parametre vektörü β ile kestirilen

parametre vektörü $\hat{\boldsymbol{\beta}}_{[\mathbf{y}_\in,\mathbf{X}_\in],\mathbf{k}}$ arasındaki m=1000 tekrardan sonra elde edilen 'ortalama açı' da hesaplanmıştır. Böylece yeni önerilen üç sağlam yöntem ile gerçek parametre vektörünün ne kadar başarılı bir şekilde kestirildiği araştırılmıştır. İdeal bir sağlam yöntem için, bu değerin sıfıra yakın olması beklenir [9, 10].

1. Kötü kaldıraç gözlemleri, regresyon hiper düzlemi üzerine yansımaları gözlemlerin çoğunun (temiz gözlemler) yansıdığı bölgenin dışına düşen ve regresyon hiper düzleminden uzak gözlemlerdir.

Kötü Kaldıraç Gözlemleri:
$$\begin{aligned}\mathbf{T}_\in &\sim \mathrm{N}_2(\mathbf{10}_2,\boldsymbol{\Sigma}_\mathrm{t})\\ \mathbf{X}_\in &= \mathbf{T}_\in\mathbf{I}_{2,\mathrm{p}} + \mathrm{N}_\mathrm{p}(\mathbf{0}_\mathrm{p},0.1\mathbf{I}_\mathrm{p})\end{aligned}$$

2. Dikey (vertical) aykırı değerler, regresyon hiper düzleminden çok uzak olan, ancak gözlemlerin çoğunun yansıdığı bölgeye yansıyan gözlemlerdir.

Dikey Aykırı Değerler: $\mathbf{y}_\in = \mathbf{TA}_{2,1} + \mathrm{N}(10.0.1)$

3. İyi kaldıraç gözlemleri, hiper düzlem civarında bulunan, ancak gözlemlerin büyük çoğunluğunun bulunduğu kümeden uzak gözlemlerdir.

İyi Kaldıraç Gözlemleri:
$$\begin{aligned}\mathbf{T}_\in &\sim \mathrm{N}_2(\mathbf{10}_2,\boldsymbol{\Sigma}_\mathrm{t})\\ \mathbf{X}_\in &= \mathbf{T}_\in\mathbf{I}_{2,\mathrm{p}} + \mathrm{N}_\mathrm{p}((\mathbf{0}_2,\mathbf{10}_{\mathrm{p}-2}),0.1\mathbf{I}_\mathrm{p})\\ \mathbf{y}_\in &= \mathbf{TA}_{2,1} + \mathrm{N}(10,0.1)\end{aligned}$$

4. Kümelenmiş (concentrated) aykırı değerler, kötü kaldıraç gözlemleri kümeleridir.

Kümelenmiş Aykırı Değerler:
$$\begin{aligned}\mathbf{T}_\in &\sim \mathrm{N}_2(\mathbf{10}_2,\boldsymbol{\Sigma}_\mathrm{t})\\ \mathbf{X}_\in &= \mathbf{T}_\in\mathbf{I}_{2,\mathrm{p}} + \mathrm{N}_\mathrm{p}(\mathbf{10}_\mathrm{p},0.001\mathbf{I}_\mathrm{p})\end{aligned}$$

5. Dik (orthogonal) aykırı değerleri, ilk olarak Huber ve Vanden Branden [15] kullanmıştır. Bu aykırı değer türünde de, y değerleri değiştirilmemiştir. Bu gözlemler, t-uzayından (bileşenlerin uzayı) uzakta bulunur. Ancak, t-uzayına

yansıtıldıktan sonra düzgün gözlemlere (aykırı olmayan gözlemlere) dönüşür. Bu nedenle, regresyon parametrelerinin tahminlerini çok fazla etkilemezler. Ancak, yüklerin hesaplanmasını etkileyebilir [15].

Dik Aykırı Değerler: $\mathbf{X}_{\in} = \mathbf{TI}_{2,p} + \mathrm{N}_p\left((\mathbf{0}_2, \mathbf{10}_{p-2}), 0.1\,\mathbf{I}_p\right)$

González vd. [10] çalışmasındaki benzetim düzenine göre temiz veri kümesi, gözlemlerin ilk % 10'unun ve % 20'sinin kötü kaldıraç gözlemleri, dikey aykırı değerler, iyi kaldıraç gözlemleri, kümelenmiş aykırı değerler ve dik aykırı değerler ile yer değiştirdiği veri kümeleri için sonuçlar elde edilmiştir. Çizelge 5.4'te n=200, p=5, k=2 için gözlemlerin ilk % 10'u aykırı değerler tarafından bozulduğunda elde edilen benzetim sonuçları verilmiştir. Çizelge 5.5'te ise aynı düzen için gözlemlerin ilk % 20'si aykırı değerler tarafından bozulduğunda elde edilen benzetim sonuçları verilmiştir.

Çizelge 5.4 incelendiğinde, temiz veri kümesi için yeni önerdiğimiz üç sağlam yöntemin ve literatürde var olan dört sağlam yöntemin etkinlik, veriye uyum ve kestirim açısından klasik yönteme yakın bir performansa sahip oldukları görülür. Veri kümesi çeşitli aykırı değer türleri tarafından bozulduğunda ise, literatürde var olan dört sağlam yöntemin ve yeni önerilen üç sağlam yöntemin klasik yönteme karşı özellikle etkinlik ve kestirim açısından üstünlükleri açık bir şekilde görülür. Özellikle veri kümesinde kötü kaldıraç gözlemleri ve kümelenmiş aykırı değerler olduğunda, klasik yöntemin etkinlik, veriye uyum ve kestirimdeki başarısı yeni önerilen üç sağlam yönteme göre çok düşüktür. Bu aykırı değer türlerinde klasik PLSR yöntemi için gerçek ve kestirilen parametre vektörleri arasındaki ortalama açı değerleri de, sağlam yöntemlere göre çok büyüktür.

Çizelge 5.4'teki kötü kaldıraç gözlemleri ve kümelenmiş aykırı değerler dışındaki aykırı değer türlerine bakıldığında, veriye uyum bakımından klasik yöntem ile yeni önerilen üç yeni yöntemin de dahil olduğu sağlam yöntemler arasında çok büyük farklar yoktur. Bu benzetim düzeni için yeni önerilen *PLS-ARWMCD*, *PLS-Smult* ve *PLS-MMmult* sağlam PLSR yöntemlerinin ve literatürde var olan sağlam

RSIMPLS ile PLS-KurSD yöntemlerinin tüm aykırı değer türlerinde özellikle etkinlikteki başarıları bakımından öne çıktığı belirtilmelidir. Genel olarak, dikey aykırı değerler dışındaki tüm aykırı değer türleri için yeni önerilen üç sağlam PLSR yöntemi de literatürde var olan sağlam PRM yönteminden etkinlik, veriye uyum ve kestirim açısından daha başarılı sonuçlar vermiştir. Çizelge 5.4 incelendiğinde veri kümesi % 10 oranında aykırı değerler tarafından bozulduğunda, yeni önerilen sağlam *PLS-ARWMCD*, *PLS-Smult* ve *PLS-MMmult* yöntemleri için alt küme sayılarının N=20 ya da N=500 seçilmesinin sonuçları etkilemediğini söyleyebiliriz.

Çizelge 5.4. n=200, p=5 ve k=2, aykırı değer oranı % 10, m=1000 tekrar için benzetim sonuçları.

	PLSR	RSIMPLS	PRM	PLS-SD	PLS-KurSD	***PLS-ARWMCD***	***PLS-Smult***	***PLS-MMmult***
Temiz Veri								
MSE	0.0092	0.0111	0.0101	0.0104	0.0100	0.0105	0.0106	0.0096
GOF	0.8312	0.8308	0.8308	0.8308	0.8309	0.8308	0.8308	0.8310
RMSE	1.0961	1.0974	1.0969	1.0973	1.0968	1.0974	1.0974	1.0966
Açı $\left(\beta;\hat{\beta}_{[y_c,x_c]k}\right)$	0.0446	0.0519	0.0462	0.0492	0.0477	0.0491	0.0497	0.0466
Kötü Kaldıraç Gözlemleri								
MSE	1.7184	0.0115	0.0688	0.0969	0.0109	0.0104	0.0108	0.0102
GOF	0.2585	0.8306	0.8177	0.8098	0.8307	0.8309	0.8308	0.8309
RMSE	2.2892	1.0996	1.1413	1.1654	1.0996	1.0989	1.0993	1.0987
Açı $\left(\beta;\hat{\beta}_{[y_c,x_c]k}\right)$	1.1403	0.0515	0.0796	0.0943	0.0496	0.0478	0.0493	0.0473
Dikey Aykırı Değerler								
MSE	0.0489	0.0107	0.0121	0.0118	0.0113	0.0106	0.0110	0.0104
GOF	0.8170	0.8295	0.8294	0.8296	0.8299	0.8300	0.8299	0.8301
RMSE	1.1384	1.0989	1.0998	1.0998	1.0987	1.0981	1.0987	1.0980
Açı $\left(\beta;\hat{\beta}_{[y_c,x_c]k}\right)$	0.1130	0.0467	0.0516	0.0526	0.0507	0.0485	0.0502	0.0481
İyi Kaldıraç Gözlemleri								
MSE	0.5650	0.0120	0.5236	0.0159	0.0112	0.0105	0.0109	0.0103
GOF	0.8094	0.8293	0.8128	0.8292	0.8294	0.8296	0.8294	0.8296
RMSE	1.1556	1.0955	1.1467	1.0953	1.0950	1.0944	1.0949	1.0942
Açı $\left(\beta;\hat{\beta}_{[y_c,x_c]k}\right)$	0.5269	0.0527	0.5037	0.0757	0.0498	0.0477	0.0490	0.0471
Kümelenmiş Aykırı Değerler								
MSE	1.9646	0.0118	1.6318	0.0300	0.0109	0.0104	0.0108	0.0102
GOF	0.5093	0.8307	0.7503	0.8281	0.8307	0.8309	0.8308	0.8309
RMSE	1.8671	1.0996	1.3228	1.1078	1.0996	1.0989	1.0993	1.0987
Açı $\left(\beta;\hat{\beta}_{[y_c,x_c]k}\right)$	1.1031	0.0529	0.6964	0.0707	0.0496	0.0478	0.0493	0.0473
Dik Aykırı Değerler								
MSE	0.1815	0.0137	0.1341	0.0107	0.0109	0.0103	0.0108	0.0102
GOF	0.7847	0.8295	0.7988	0.8298	0.8298	0.8300	0.8299	0.8300
RMSE	1.2316	1.1002	1.1917	1.0996	1.0997	1.0990	1.0996	1.0990
Açı $\left(\beta;\hat{\beta}_{[y_c,x_c]k}\right)$	0.2821	0.0575	0.2323	0.0494	0.0503	0.0488	0.0498	0.0480

Çizelge 5.5 incelendiğinde, veri kümesindeki aykırı değer yüzdesi arttığında sağlam yöntemlerden PRM yönteminin dikey aykırı değerler dışındaki tüm aykırı değerler türlerinde özellikle etkinlik ve kestirimdeki başarısının daha da düştüğü görülür. Ayrıca, bu aykırı değer türlerinde PRM yönteminin gerçek ve kestirilen parametre vektörleri arasındaki ortalama açı değerleri de diğer sağlam yöntemlere

göre çok büyük çıkmıştır. Özellikle veri kümesinde iyi kaldıraç gözlemleri, kümelenmiş aykırı değerler ya da dik aykırı değerler varlığında PRM yöntemi MSE, RMSE ve ortalama açı ölçütleri bakımından klasik yöntemden bile daha kötü sonuçlar vermiştir. Çizelge 5.5 incelendiğinde, veri kümesi aykırı değerler tarafından % 20 oranında bozulduğunda yeni önerilen sağlam *PLS-ARWMCD* yöntemi için alt küme sayısının N=20 ya da N=500 olarak seçilmesi sonuçları etkilememiştir. Ancak, veri kümesindeki aykırı değer yüzdesi % 20 gibi bir seviyeye çıktığında yeni önerilen sağlam *PLS-Smult* ile *PLS-MMmult* yöntemleri için elde edilen sonuçlar, seçilen alt küme sayılarından önemli bir şekilde etkilenmiştir. Veri kümesindeki aykırı değer yüzdesi arttığında alt küme sayısı N=20 yerine N=500 gibi büyük bir sayı olarak belirlendiğinde, *PLS-Smult* ile *PLS-MMmult* yöntemlerinin özellikle etkinlikteki başarılarının arttığı görülür. Özellikle dikey aykırı değerler varlığında, bu iki yeni sağlam PLSR yöntemi için de, alt küme sayılarını daha büyük seçmenin özellikle etkinlik ve kestirimdeki başarıları üzerindeki etkileri daha açık bir şekilde görülür. Bu nedenle, N=20 ya da N=500 olarak seçildiğinde *PLS-Smult* ve *PLS-MMmult* yöntemleri için elde edilen sonuçlar klasik yöntemden daha iyi olmasına karşın, literatürde var olan diğer dört sağlam yönteme özellikle etkinlik açısından seçenek iyi yöntemler olmaları amacıyla, bu iki yeni sağlam PLSR yöntemi için de N=500 olarak daha fazla alt kümenin seçilmesi tercih edilir.

Veri kümesinde kötü kaldıraç gözlemleri olduğunda, yeni önerilen sağlam *PLS-ARWMCD*, *PLS-Smult* ve *PLS-MMmult* yöntemlerinin literatürde var olan sağlam PLS-KurSD ve RSIMPLS yöntemleri ile birlikte etkinlik ve kestirim açısından en başarılı yöntemlerden olduğu görülür. Veri kümesinde dikey aykırı değerler bulunduğunda ise, yeni önerilen sağlam *PLS-ARWMCD* ile literatürde var olan sağlam RSIMPLS ve PLS-KurSD yöntemlerinin etkinlik ve kestirim açısından daha başarılı sonuçlar verdiği görülür. Ayrıca, veri kümesinde dikey aykırı değerlerin varlığında bu üç sağlam yöntem için gerçek ve kestirilen parametre vektörü arasındaki ortalama açıların da, diğer sağlam yöntemlerden çok daha küçük olduğu görülür. Kümelenmiş aykırı değer türü, başa çıkması en zor aykırı değer türüdür. Veri kümesinde % 20 oranında kötü kaldıraç gözlemleri ya da kümelenmiş aykırı değerler olduğunda, yeni önerilen üç sağlam PLSR yönteminin

de literatürde var olan sağlam PLS-SD ve PRM yöntemlerinden etkinlik, veriye uyum, kestirim ve ortalama açı ölçütleri bakımından çok daha iyi sonuçlar verdiği görülmüştür. Veri kümesindeki aykırı değer yüzdesi % 20 gibi bir seviyeye çıktığında yeni önerilen *PLS-ARWMCD*, *PLS-Smult* ve *PLS-MMmult* sağlam PLSR yöntemleri klasik yöntemle kıyaslandıklarında tüm aykırı değer türleri için etkinlik, veriye uyum ve kestirimdeki başarısını korumaktadır. Ancak, yeni önerilen sağlam *PLS-ARWMCD* yöntemi, veri kümesindeki aykırı değer yüzdesi % 20 gibi bir seviyeye çıktığında seçilen alt küme sayısından etkilenmemiş ve özellikle dikey aykırı değerler varlığında *PLS-Smult* ve *PLS-MMmult* yöntemlerinden etkinlik ve kestirim açısından daha iyi sonuçlar vermiştir.

Çizelge 5.5. n=200, p=5 ve k=2, aykırı değer oranı % 20, m=1000 tekrar için benzetim sonuçları.

	PLSR	RSIMPLS	PRM	PLS-SD	PLS-KurSD	***PLS-ARWMCD***	***PLS-Smult***	***PLS-MMmult***
Temiz Veri								
MSE	0.0092	0.0111	0.0101	0.0104	0.0100	0.0105	0.0106	0.0096
GOF	0.8312	0.8308	0.8308	0.8308	0.8309	0.8308	0.8308	0.8310
RMSE	1.0961	1.0974	1.0969	1.0973	1.0968	1.0974	1.0974	1.0966
Açı $\left(\beta;\hat{\beta}_{[y_c,x_c]k}\right)$	0.0446	0.0519	0.0462	0.0493	0.0477	0.0491	0.0497	0.0466
Kötü Kaldıraç Gözlemleri								
MSE	1.8946	0.0122	1.7726	0.4134	0.0121	0.0109	0.0114 **(0.0315)**	0.0111 **(0.0183)**
GOF	0.1858	0.8309	0.2395	0.7143	0.8310	0.8313	0.8312 **(0.8245)**	0.8313 **(0.8289)**
RMSE	2.4002	1.1012	2.3205	1.4282	1.1011	1.0998	1.1005 **(1.1142)**	1.1000 **(1.1040)**
Açı $\left(\beta;\hat{\beta}_{[y_c,x_c]k}\right)$	1.3018	0.0537	1.1833	0.2467	0.0540	0.0500	0.0520 **(0.0648)**	0.0507 **(0.0555)**
Dikey Aykırı Değerler								
MSE	0.0791	0.0115	0.0174	0.0176	0.0126	0.0112	0.0334 **(0.0684)**	0.0304 **(0.0705)**
GOF	0.8057	0.8278	0.8265	0.8267	0.8282	0.8286	0.8219 **(0.8107)**	0.8224 **(0.8092)**
RMSE	1.1681	1.1002	1.1060	1.1063	1.1003	1.0989	1.1209 **(1.1552)**	1.1179 **(1.1584)**
Açı $\left(\beta;\hat{\beta}_{[y_c,x_c]k}\right)$	0.1437	0.0471	0.0632	0.0656	0.0540	0.0503	0.0804 **(0.1302)**	0.0773 **(0.1318)**
İyi Kaldıraç Gözlemleri								
MSE	0.5847	0.0126	0.6306	0.0718	0.0120	0.0108	0.0113 **(0.0181)**	0.0109 **(0.0180)**
GOF	0.8072	0.8294	0.8028	0.8277	0.8294	0.8297	0.8296 **(0.8293)**	0.8297 **(0.8294)**

RMSE	1.1639	1.0985	1.1767	1.1032	1.0983	1.0972	1.0978 **(1.0986)**	1.0974 **(1.0982)**
Açı $\left(\beta;\hat{\beta}_{[y_c,X_c]k}\right)$	0.5377	0.0549	0.5608	0.1823	0.0527	0.0489	0.0502 **(0.0560)**	0.0491 **(0.0553)**
Kümelenmiş Aykırı Değerler								
MSE	1.8527	0.0128	1.9260	0.1628	0.0121	0.0109	0.0114 **(0.0153)**	0.0111 **(0.0205)**
GOF	0.4929	0.8310	0.4850	0.8107	0.8310	0.8313	0.8312 **(0.8305)**	0.8313 **(0.8297)**
RMSE	1.8946	1.1012	1.9104	1.1648	1.1011	1.0998	1.1005 **(1.1018)**	1.1000 **(1.1038)**
Açı $\left(\beta;\hat{\beta}_{[y_c,X_c]k}\right)$	1.1091	0.0569	1.1119	0.2307	0.0539	0.0500	0.0520 **(0.0541)**	0.0507 **(0.0560)**
Dik Aykırı Değerler								
MSE	0.1987	0.0176	0.2332	0.0108	0.0115	0.0104	0.0109 **(0.0131)**	0.0105 **(0.0128)**
GOF	0.7806	0.8310	0.7718	0.8319	0.8319	0.8322	0.8321 **(0.8315)**	0.8322 **(0.8314)**
RMSE	1.2488	1.1026	1.2739	1.1007	1.1010	1.0999	1.1005 **(1.1022)**	1.1001 **(1.1019)**
Açı $\left(\beta;\hat{\beta}_{[y_c,X_c]k}\right)$	0.2982	0.0660	0.3247	0.0504	0.0519	0.0491	0.0506 **(0.0533)**	0.0494 **(0.0525)**

EK 4 ve EK 5'teki çizelgelerde ise n=1000, p=5, k=2 için sırasıyla, verilerin ilk % 10'u ve % 20'si aykırı değerler tarafından bozulduğunda elde edilen benzetim sonuçlarına ilişkin çizelgeler verilmiştir. Böylece, çok büyük bir örneklem seçildiğinde, yeni önerilen üç sağlam PLSR yöntemi ve literatürde var olan dört sağlam PLSR yöntemi incelenmiştir.

EK 4'teki çizelge incelendiğinde büyük örneklem için de, veri kümesi % 10 oranında aykırı değerler tarafından bozulduğunda, yeni önerilen sağlam *PLS-ARWMCD*, *PLS-Smult* ve *PLS-MMmult* yöntemleri için alt küme sayılarının N=20 ya da N=500 seçilmesinin sonuçları etkilemediği görülür. EK 4'teki çizelgeden temiz veri kümesi için, klasik ve yeni önerilen üç yöntemin de dahil olduğu sağlam PLSR yöntemlerinin etkinlik, veriye uyum ve kestirimdeki başarıları açısından çok yakın sonuçlar verdiği görülür. Böylece örneklem büyüklüğü arttıkça beklendiği üzere, temiz veri kümesi için sağlam yöntemler klasik yönteme çok daha yakın sonuçlar vermiştir. Veri kümesindeki gözlem sayısı arttığında ve veri kümesinde % 10 oranında özellikle kötü kaldıraç gözlemleri ya da kümelenmiş aykırı değerler olduğunda klasik PLSR yönteminin sağlam yöntemlere karşın verdiği kötü sonuçlar daha açık bir şekilde görülür. Veri kümesindeki gözlem sayısı

arttığında, tüm aykırı değer türleri için yeni önerilen üç sağlam PLSR yönteminin de, etkinlik, veriye uyum ve kestirim açısından çok yakın sonuçlarıyla en başarılı sağlam yöntemlerden olduğu ve literatürde var olan sağlam RSIMPLS ve PLS-KurSD yöntemlerinin başarısını yakaladığı görülür.

Çizelge 5.4 ve EK 4'te verilen çizelgedeki sonuçlar birlikte incelendiğinde, veri kümesinde % 10 oranında aykırı değer olduğunda yeni önerilen üç sağlam PLSR yöntemi *PLS-ARWMCD, PLS-Smult* ve *PLS-MMmult'*un örneklem büyüklüğü arttıkça özellikle MSE değerlerinin küçüldüğü görülür. Ayrıca, veri kümesinde özellikle kötü kaldıraç gözlemleri, iyi kaldıraç gözlemleri ve kümelenmiş aykırı değerler varlığında klasik PLSR yöntemine ilişkin gerçek ve kestirilen parametre vektörleri arasındaki ortalama açı değerleri neredeyse aynı kalırken, yeni önerilen üç sağlam PLSR yöntemine ve literatürde var olan sağlam RSIMPLS ve PLS-KurSD yöntemlerine ilişkin ortalama açı değerleri yarı yarıya düşmüştür. Böylece, gözlem sayısı arttıkça yeni önerilen üç sağlam PLSR yöntemi ile gerçek parametre vektörünün daha başarılı bir şekilde kestirildiğini söyleyebiliriz.

EK 5'teki çizelge incelendiğinde aynı orta büyüklükteki örneklemde olduğu gibi büyük örneklem için de veri kümesindeki aykırı değer yüzdesi arttığında alt küme sayısı N=20 yerine N=500 gibi büyük bir sayı olarak belirlendiğinde, genel olarak *PLS-Smult* ile *PLS-MMmult* yöntemlerinin özellikle etkinlikteki başarılarının arttığı görülür. Ancak yeni önerilen sağlam *PLS-ARWMCD* yönteminin verdiği sonuçlar, alt küme sayısının değişmesinden etkilenmemiştir. Bu nedenle, aynı orta büyüklükteki örneklem için söz edilen nedenlerden dolayı büyük örneklemde de bu iki yeni sağlam PLSR yöntemi için N=500 olarak daha fazla alt kümenin seçilmesi tercih edilir.

EK 5'teki çizelgeden, aynı orta büyüklükteki örneklemde olduğu gibi büyük örneklem için de veri kümesinde % 20 gibi yüksek bir oranda dikey aykırı değerler dışında diğer aykırı değerler türleri olduğunda, sağlam yöntemlerden PRM yönteminin etkinlik ve kestirimdeki başarısı çok açık bir şekilde düşmüştür. Veri kümesinde kötü kaldıraç gözlemleri olduğunda, yeni önerilen sağlam

PLS-ARWMCD, *PLS-Smult*, *PLS-MMmult* yöntemlerinin literatürde var olan sağlam PLS-KurSD ve RSIMPLS yöntemleri ile birlikte etkinlik, veriye uyum ve kestirim açısından en başarılı yöntemlerden olduğu görülür. Veri kümesinde dikey aykırı değerler bulunduğunda, yeni önerilen sağlam *PLS-ARWMCD* yöntemi literatürde var olan sağlam RSIMPLS ve PLS-KurSD yöntemleri ile çok yakın bir başarı göstermiştir. Veri kümesinde iyi kaldıraç gözlemleri bulunduğunda ise, yeni önerilen üç sağlam PLSR yöntemi de literatürde var olan sağlam RSIMPLS ile PLS-KurSD yöntemleri ile beraber özellikle etkinlikteki başarıları bakımından öne çıkar. Veri kümesinde kümelenmiş aykırı değerler olduğunda, yeni önerilen sağlam *PLS-ARWMCD*, *PLS-Smult*, *PLS-MMmult* yöntemleri etkinlik, veriye uyum ve kestirimdeki başarıları açısından literatürde var olan sağlam RSIMPLS ile PLS-KurSD yöntemlerini yakalamıştır. Veri kümesinde dik aykırı değer varlığında ise, yeni önerilen üç sağlam PLSR yöntemi de klasik yöntem ile literatürde var olan sağlam PRM yönteminden etkinlik, veriye uyum ve kestirim açısından daha başarılıdır.

Çizelge 5.5 ve EK 5'teki sonuçlar birlikte incelendiğinde, orta büyüklükteki örneklemde veri kümesindeki aykırı değer yüzdesi arttığında, % 20 oranında dikey aykırı değer var olduğunda, N=20 ya da N=500 olarak seçildiğinde *PLS-Smult* ile *PLS-MMmult* yöntemleri diğer sağlam yöntemler ile karşılaştırıldığında kestirim açısından daha başarısızken, her iki yöntem de örneklem büyüklüğü arttığında diğer sağlam yöntemlerin kestirimdeki başarısını yakalamıştır. Genel olarak, n=200 gibi orta büyüklükteki bir örneklemde veri kümesindeki aykırı değer yüzdesi arttığında, *PLS-Smult* ile *PLS-MMmult* yöntemleri için çekilen alt küme sayısındaki artışın bu iki yöntemin etkinlik ve sağlamlığına ilişkin performansları üzerinde olumlu etkisi daha çok görülür. n=1000 gibi büyük bir örneklemde ise, bu yeni önerilen iki sağlam yöntemin özellikle kestirimdeki performansı çekilen alt küme sayısından daha az etkilenmiştir.

Sonuç olarak, Alt Bölüm 5.1.1 ve Alt Bölüm 5.1.2'deki benzetim düzenlerinden elde edilen sonuçlar birlikte incelendiğinde; yeni önerilen sağlam *PLS-ARWMCD*, *PLS-Smult* ve *PLS-MMmult* yöntemlerinin literatürde var olan dört sağlam PLSR

yöntemine seçenek birer sağlam PLSR yöntemi olarak önerilebileceği söylenebilir. Bu üç sağlam PLSR yönteminden özellikle *PLS-Smult* ile *PLS-MMmult* yöntemleri için veri kümesindeki aykırı değer yüzdesi arttığında daha etkin ve sağlam sonuçlar elde etmek için, bu yöntemlerde kullanılan FastS ve FastMM algoritmalarındaki alt küme sayılarının (N) arttırılması tercih edilebilir.

Kestiricileri sadece normal dağılım altında incelemek yeterli değildir. Normal dağılıma ek olarak, daha ağır kuyruklu dağılımlar da incelenmelidir [18]. Sağlam kestiriciler tasarlanırken sadece sağlamlık özelliklerini araştırmak değil, aynı zamanda etkinliklerini araştırmak da önemlidir. Bir kestiricinin varyansı her bir dağılım için minimuma yakın bir değer alıyor ise, bu kestirici 'yüksek etkinliğe' sahiptir denir. Yanlı kestiriciler için, varyans yerine MSE'ye bakılır. Yüksek etkinlik özelliği, bir kestiricinin dağılımın belli olmadığı durumlarda tekrarlı olarak alınan örneklemler için de iyi sonuçlar verdiğini garanti altına alır. Örneklemin bozulduğu durumlarda, kestirimlerin çok az miktarda değişim göstermesi önemli bir durumdur [1, 7, 18]. Alt bölümler 5.1.1 ve 5.1.2'de, modeldeki hata terimleri sadece normal dağılımdan geldiğinde ve model çeşitli aykırı değerler tarafından bozulduğunda elde edilen sonuçlar tartışılmıştır. Bir sonraki alt bölümde ise, aykırı değerler tarafından bozulmayan temiz modeldeki hata terimleri standart normal dağılımla birlikte farklı altı dağılımdan geldiğinde özellikle yeni önerilen üç sağlam PLSR yönteminin ve literatürde var olan diğer dört sağlam PLSR yönteminin etkinlikteki başarısı incelenecektir.

5.1.3. Aykırı Değerler Tarafından Bozulmayan Temiz Modeldeki Hata Terimlerinin Farklı Dağılımlardan Geldiği Benzetim Düzeni

Bu alt bölümde de, bir önceki alt bölümdeki benzetim düzenine benzer bir düzen kullanılmıştır. Bu benzetim çalışmasındaki bağımsız değişkenler matrisi $\mathbf{X}$, $\boldsymbol{\Sigma}_t = \begin{pmatrix} 4 & 0 \\ 0 & 2 \end{pmatrix}$ olmak üzere önceki alt bölümlerdeki benzetim düzenleri ile aynı şekilde Eş. (5.4)'teki gibi elde edilmiştir. Burada, $\boldsymbol{\beta}_{p\times 1} = \mathbf{l}'_{p\times 2}\mathbf{A}_{2\times 1}$ olmak üzere $\mathbf{y} = \mathbf{X}\boldsymbol{\beta} + \boldsymbol{\varepsilon}$ modelindeki ε hata terimlerinin $N(0,1)$ normal dağılımdan farklı

dağılımlardan geldiğinde yeni önerilen üç sağlam PLSR yönteminin etkinliği MSE ölçütü kullanılarak incelenmiştir. Serneels vd. [31] çalışmasından yola çıkılarak, hata terimleri için 'Standart Normal', 'Laplace', 5 ve 2 serbestlik dereceli 't-dağılımları' ile 'Cauchy' ve 'Slash' gibi ağır kuyruklu dağılımlar kullanılmıştır. Slash dağılımı, $N(0,1)$ dağılımının $(0,1)$ aralığındaki Uniform dağılıma bölünmesi olarak tanımlanmıştır. Cauchy ve Slash dağılımları, oldukça büyük hata terimleri yaratabilir ve bu hata terimleri, dikey aykırı değerler olarak adlandırılır [31]. Ancak, bu hata dağılımlarının yarattığı aykırı değer yüzdesi bilinemez. Bu nedenle, bu allt bölümde, özellikle *PLS-Smult* ile *PLS-MMmult* yöntemleri için daha etkin sonuçlar elde etmek amacıyla yeni önerilen üç sağlam PLSR yöntemi için de alt küme sayısı N=500 olarak seçilmiştir.

Bu benzetim düzeninde ilk olarak, literatürdeki çalışmalardan yola çıkılarak farklı örneklem şemaları ile çalışılmıştır. $n \times p$, **X** matrisinin boyutunu göstermek üzere; İlk örneklem şeması için 30×10, ikinci örneklem şeması için 60×15, üçüncü örneklem şeması için ise 100×20 seçilmiştir. Bu üç örneklem şeması için de, k=2 olarak belirlenmiştir. Her bir durum için, m=1000 tekrar yapılmıştır. Bu benzetim düzenine ilişkin sonuçlar, Çizelge 5.6'da verilmiştir.

Çizelge 5.6. Gözlem sayısının (n) ve bağımsız değişken sayısının (p) farklı seçildiği üç örneklem şeması için farklı hata dağılımlarında m=1000 tekrar yapılarak benzetim ile elde edilmiş MSE'ler.

		N(0.1)	**Laplace**	**t_5**	**t_2**	**Cauchy**	**Slash**
n=30, p=10, k=2	PLS	0.0578	0.0953	0.0854	1.8796	5340.0158	1710.6507
	RSIMPLS	0.0962	0.1150	0.1143	0.1408	0.3003	0.6315
	PRM	0.0642	0.0772	0.0749	0.1002	0.1606	0.3989
	PLS-SD	0.1090	0.1768	0.1594	0.2517	1.3591	4.9829
	PLS-KurSD	0.1332	0.2773	0.2126	0.6126	4.3038	7.6289
	PLS-ARWMCD	0.1136	0.1575	0.1466	0.2435	1.3506	3.3354
	PLS-Smult	0.0705	0.1119	0.0938	0.1503	0.9036	2.4188
	PLS-MMmult	0.0675	0.1050	0.0904	0.1503	0.8291	2.5045
n=60, p=15, k=2	PLS	0.0456	0.0799	0.0691	2.6651	53800.5323	35444.3967
	RSIMPLS	0.0956	0.1062	0.0970	0.1077	0.1660	0.4237
	PRM	0.0498	0.0577	0.0566	0.0728	0.1207	0.2506
	PLS-SD	0.0559	0.0817	0.0725	0.1233	0.5135	1.3528
	PLS-KurSD	0.1054	0.1750	0.1506	0.2753	2.4771	4.1888
	PLS-ARWMCD	0.0914	0.1348	0.1144	0.1614	0.5048	1.1657
	PLS-Smult	0.0469	0.0728	0.0632	0.1107	0.6856	1.3887
	PLS-MMmult	0.0481	0.0735	0.0640	0.1086	0.4896	1.1905
n=100, p=20, k=2	PLS	0.0269	0.0408	0.0363	0.5078	7622.5505	132730.7680
	RSIMPLS	0.0378	0.0387	0.0397	0.0411	0.0495	0.0814
	PRM	0.0307	0.0339	0.0342	0.0398	0.0523	0.0907
	PLS-SD	0.0314	0.0412	0.0378	0.0549	0.1299	0.3429
	PLS-KurSD	0.0476	0.0641	0.0586	0.0824	0.2412	0.7020
	PLS-ARWMCD	0.0431	0.0576	0.0521	0.0714	0.1552	0.3756
	PLS-Smult	0.0308	0.0421	0.0380	0.0570	0.1763	0.4435
	PLS-MMmult	0.0337	0.0446	0.0407	0.0571	0.1342	0.2808

Çizelge 5.6 incelendiğinde, hatalar standart normal dağıldığında her üç örneklem şeması için de en etkin yöntem, beklendiği üzere klasik PLSR yöntemidir. Ancak, ikinci örneklem şeması için bu hata dağılımında yeni önerilen sağlam *PLS-Smult* ve *PLS-MMmult* yöntemleri, literatürde var olan sağlam PRM yöntemi ile birlikte klasik yöntemin etkinlikteki başarısını yakalamıştır. Üçüncü örneklem şeması için ise bu hata dağılımında, literatürde var olan sağlam PRM ve PLS-SD yöntemleri ile birlikte yeni önerilen sağlam *PLS-Smult* ile *PLS-MMmult* yöntemleri klasik yöntemin etkinlikteki başarısını yakalar. Laplace dağılımına ilişkin elde edilen MSE'ler incelendiğinde ise, birinci örneklem şeması için en etkin yöntem PRM yöntemidir ve yeni önerdiğimiz üç sağlam yöntemin de dahil olduğu diğer tüm

sağlam yöntemler klasik yönteme karşın etkinlikteki başarısını kaybetmiştir. İkinci örneklem şeması incelendiğinde, bu hata dağılımı için PRM yöntemi en etkin yöntem olmasına karşın, yeni önerilen sağlam *PLS-Smult* ile *PLS-MMmult* yöntemleri klasik PLSR yönteminin etkinlikteki başarısını yakalamıştır. Üçüncü örneklem şeması için bu hata dağılımında PRM ve RSIMPLS yöntemleri etkinlikteki başarıları ile öne çıkar ve PLS-SD yöntemi ise yeni önerilen sağlam *PLS-Smult* ile *PLS-MMmult* yöntemleri ile birlikte klasik yöntemin başarısını yakalar. t_5 dağılımına ilişkin MSE'ler incelendiğinde ise, her üç örneklem şeması için de PRM yöntemi en etkin yöntemdir. Birinci örneklem şeması incelendiğinde bu hata dağılımı için, yeni önerilen üç sağlam PLSR yöntemi ve PRM hariç literatürde var olan diğer üç sağlam PLSR yöntemi klasik yönteme karşı etkinlikteki başarısını kaybetmiştir. İkinci ve üçüncü örneklem şemaları incelendiğinde, bu hata dağılımı için yeni önerilen sağlam *PLS-ARWMCD* yöntemi klasik yönteme karşı etkinlikteki başarını kaybetmesine karşın, yeni önerilen sağlam *PLS-Smult* ile *PLS-MMmult* yöntemlerinin etkinlik açısından klasik yöntemi yakaladığı söylenebilir. t_2 dağılımına ilişkin elde edilen sonuçlardan, her üç örneklem şeması için de yeni önerilen üç sağlam PLSR yönteminin ve literatürde var olan dört sağlam PLSR yönteminin klasik PLSR yöntemine karşın etkinlikteki üstünlükleri açık bir şekilde görülür. Bu hata dağılımında, yeni önerilen sağlam *PLS-ARWMCD*, *PLS-Smult* ve *PLS-MMmult* yöntemleri her üç örneklem şeması için de literatürde var olan sağlam PLS-KurSD yönteminden daha etkin sonuçlar vermiştir. Ayrıca, t_2 dağılımı için birinci ve ikinci örneklem şemalarında, yeni önerilen sağlam *PLS-Smult* ve *PLS-MMmult* yöntemleri, PLS-SD yöntemin daha etkin sonuçlar vermiştir. Cauchy ve Slash dağılımlarına ilişkin elde edilen MSE'ler incelendiğinde, üç farklı örneklem şeması için de klasik PLSR yönteminin etkinlik açısından yeni önerilen sağlam *PLS-ARWMCD*, *PLS-Smult* ve *PLS-MMmult* yöntemleri ile literatürde var olan dört sağlam PLSR yöntemine karşı çok başarısız olduğu görülür. Cauchy ve Slash dağılımlarından elde edilen sonuçlar, her bir örneklem şeması için incelenebilir. Buna göre, Cauchy dağılımı için ilk örneklem şeması incelendiğinde, yeni önerilen sağlam *PLS-Smult* ile *PLS-MMmult* yöntemleri, PLS-SD ile PLS-KurSD yöntemlerinden daha etkin bulunmuştur. İkinci ve üçüncü örneklem şemaları için ise, bu hata dağılımında yeni önerilen üç sağlam yöntemden *PLS-MMmult* yöntemi öne çıkmasına karşın, yeni önerilen üç sağlam PLSR yöntemi de literatürde var olan sağlam PLS-KurSD yönteminden daha etkin

bulunmuştur. Slash dağılımı için ilk örneklem şeması incelendiğinde, yeni önerilen üç sağlam PLSR yöntemi de, PLS-SD ile PLS-KurSD yöntemlerinden daha etkin bulunmuştur. İkinci ve üçüncü örneklem şemaları için bu hata dağılımında, yeni önerilen üç sağlam PLSR yöntemi de PLS-KurSD yönteminden daha etkin bulunmuştur. Genel olarak, her üç örneklem şemasından da t_2, Cauchy ve Slash gibi daha çok aykırı değer yaratan hata dağılımlarında yeni önerilen sağlam *PLS-ARWMCD*, *PLS-Smult* ve *PLS-MMmult* yöntemlerinin literatürde var olan sağlam PLS-KurSD yönteminden daha etkin olduğu söylenebilir.

Bu benzetim düzeninde, EK 6'daki çizelgede gösterildiği üzere ikinci olarak, p=5 ve k=2 olarak alınmıştır. n ise sırasıyla, 20, 200 ve 1000 olarak seçilmiştir. Böylece veri kümesindeki gözlem sayısı arttıkça, öncelikle yeni önerilen üç sağlam PLSR yöntemi olmak üzere tüm sağlam PLSR yöntemlerinin etkinliklerindeki değişimler incelenmiştir. Her bir durum için, m=1000 tekrar yapılmıştır.

EK 6'daki çizelge incelendiğinde, hatalar standart normal dağıldığında her üç örneklem büyüklüğü için de en etkin yöntem, klasik PLSR yöntemidir. Ancak, bu hata dağılımında örneklem büyüklüğü arttıkça sağlam yöntemlerin klasik yönteme çok yakın sonuçlar verdikleri ve dolayısıyla klasik yöntem kadar etkin oldukları görülür. Laplace ve t_5 dağılımları için elde edilen MSE'ler incelendiğinde ise, n=20 gibi küçük bir örneklemde PRM yöntemi ve yeni önerilen sağlam *PLS-MMmult* yöntemi hariç, literatürde var olan diğer üç sağlam yöntemin ve yeni önerilen sağlam *PLS-ARWMCD* ile *PLS-Smult* yöntemlerinin klasik PLSR yöntemi ile karşılaştırıldıklarında etkinlikte önemli bir kayba uğradıkları görülür. Bu iki hata dağılımı için de örneklem büyüklüğü arttıkça, yeni önerilen üç sağlam PLSR yöntemi ile birlikte literatürde var olan dört sağlam PLSR yönteminin de, klasik yönteme karşı etkinlikte bir kayıba uğramadıkları görülür. t_2 dağılımı için elde edilen sonuçlar incelendiğinde ise, her üç örneklem büyüklüğü için de yeni önerilen üç sağlam PLSR yönteminin ve literatürde var olan dört sağlam PLSR yönteminin klasik PLSR yöntemine karşın etkinlikteki üstünlükleri açık bir şekilde görülür. n=20 büyüklüğündeki küçük örneklemde hatalar t_2 dağılımına sahip olduğunda, yeni önerilen sağlam *PLS-MMmult* yönteminin literatürde var olan

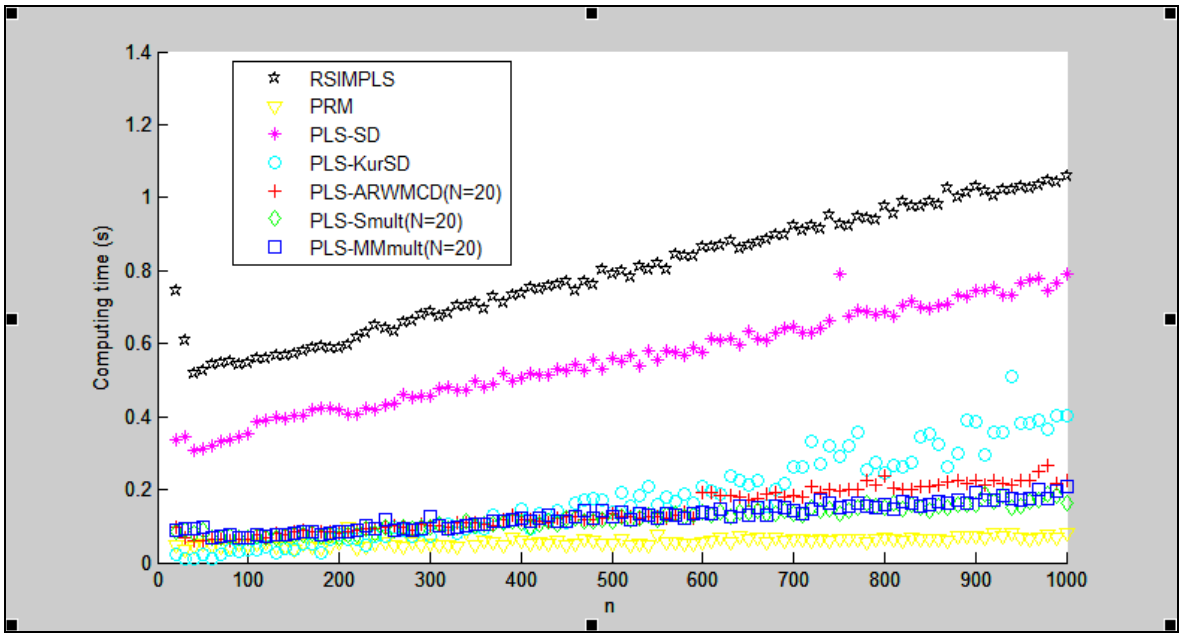
sağlam RSIMPLS, PLS-SD ve PLS-KurSD yöntemlerinden çok daha etkin olduğu izlenmiştir. Ayrıca, bu küçük örneklemde bu hata dağılımı için yeni önerilen üç sağlam PLSR yöntemi de, literatürde var olan sağlam RSIMPLS ile PLS-KurSD yöntemlerinden çok daha etkin bulunmuştur. Cauchy ve Slash dağılımları için elde edilen MSE'ler incelendiğinde, her üç örneklem büyüklüğü için de klasik PLSR yönteminin etkinlik bakımından yeni önerilen sağlam *PLS-ARWMCD*, *PLS-Smult* ve *PLS-MMmult* yöntemleri ile literatürde var olan dört sağlam PLSR yöntemine karşı çok başarısız olduğu görülür. Cauchy ve Slash dağılımlarında elde edilen sonuçlar, her bir örneklem büyüklüğü için incelenebilir. Buna göre, örneklem büyüklüğü n=20 olduğunda her iki hata dağılımı için de, yeni önerilen sağlam *PLS-Smult* ile *PLS-MMmult* yöntemlerinin her ikisi de literatürde var olan sağlam PLS-KurSD yönteminden daha etkin bulunmuştur. Örneklem büyüklüğü n=200 ve n=1000 olduğunda ise her iki hata dağılımı için de, yeni önerilen üç sağlam yöntemden *PLS-Smult* yöntemi öne çıkmış ve literatürde var olan sağlam PLS-KurSD yönteminden kısmen etkin bulunmuştur. Genel olarak, tüm hata dağılımları için örneklem büyüklüğü arttıkça yeni önerilen üç sağlam PLSR yönteminin de birbirine ve literatürde var olan diğer dört sağlam PLSR yöntemine çok daha yakın sonuçlar verdiğini söyleyebiliriz.

Buraya kadar yaptığımız benzetim çalışmalarında, yeni önerilen sağlam *PLS-ARWMCD*, *PLS-Smult* ve *PLS-MMmult* yöntemlerinin üçü için de, N=20 ve N=500 gibi az ve daha çok sayıda alt kümeler seçilmiştir. Bu üç yöntem için de seçilecek alt küme sayılarına göre, algoritmaların hızı değişmektedir. Bu nedenle, yeni önerilen üç sağlam PLSR yöntemlerinin hesaplama zamanları dolayısıyla hızları, bir sonraki alt bölümde literatürde var olan diğer dört sağlam PLSR yöntemi ile karşılaştırılmıştır.

5.1.4. Yeni Önerilen Sağlam PLSR Yöntemlerinin Hesaplama Zamanı

Bu alt bölümde yeni önerdiğimiz üç sağlam PLSR yönteminin ve literatürde var olan dört sağlam PLSR yönteminin MATLAB programında tek bir döngü çalıştırılmaları için gerekli CPU hesaplama zamanları, işlemcisi *Intel (R) Core*

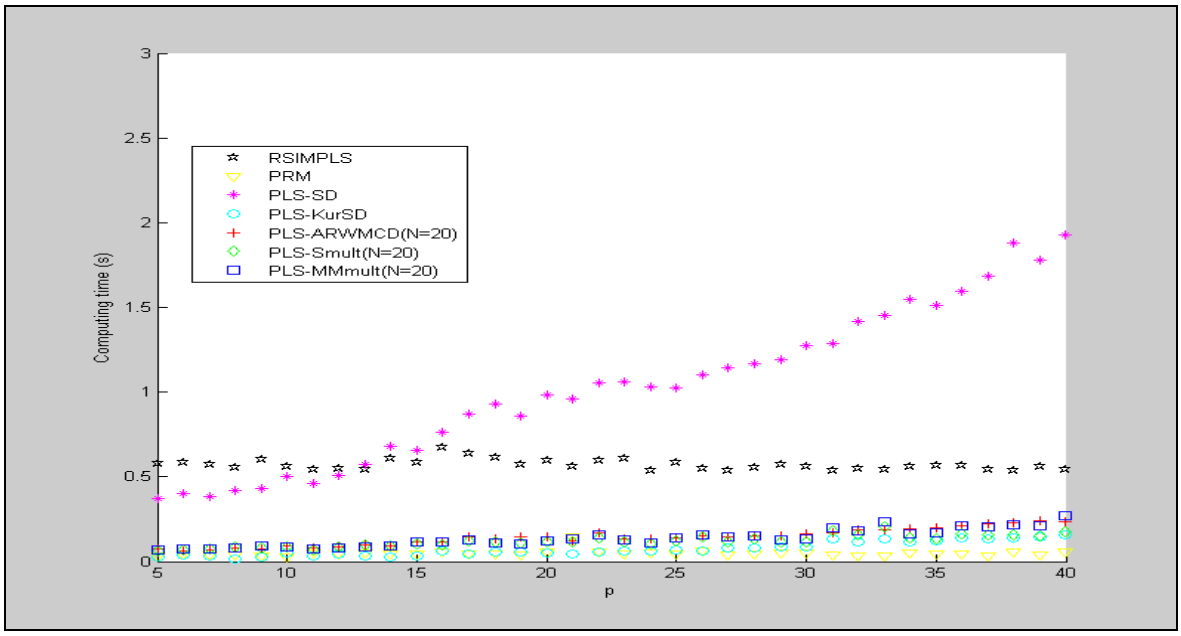
(TM)2 Duo CPU 2.33 Ghz, yüklü belleği (RAM) *2.00 GB*, sistem türü *32 bit işletim sistemi* olan bilgisayarda *MATLAB 2012b* versiyonunda karşılaştırılmıştır. Bu alt bölümde de bağımsız değişkenler matrisi $\mathbf{X}$, $\mathbf{\Sigma}_t = \begin{pmatrix} 4 & 0 \\ 0 & 2 \end{pmatrix}$ olmak üzere önceki alt bölümlerdeki benzetim düzenleri ile aynı şekilde Eş. (5.4)'teki gibi elde edilmiştir. Burada, $\boldsymbol{\beta}_{p\times 1} = \mathbf{I}'_{p\times 2}\mathbf{A}_{2\times 1}$ olmak üzere ε hata terimlerinin $N(0,1)$ normal dağılımdan geldiği $\mathbf{y} = \mathbf{X}\boldsymbol{\beta} + \boldsymbol{\varepsilon}$ modelini kullanarak, veri kümeleri dizisi üretilmiştir. Bu düzenden de anlaşılacağı üzere, modeldeki bileşen sayısı k=2 olarak belirlenmiştir. Sağlam yöntemlerin hesaplama zamanlarını dolayısıyla hızlarını karşılaştırmak için iki deney yapılmıştır. İlk olarak bağımsız değişken sayısının p=5 olarak sabitlendiği ve örneklem büyüklüğünün 20'den 1000'e kadar arttırıldığı düzen için sonuçlar, Şekil 5.1'de verilmiştir. Bu şekilden yararlanarak, alt örneklem sayıları N=20 olarak seçildiğinde yeni önerilen sağlam *PLS-ARWMCD*, *PLS-Smult* ve *PLS-MMmult* yöntemlerinin ve literatürde var olan dört sağlam PLSR yönteminin örneklem büyüklüğü arttıkça hesaplama zamanlarındaki değişimler incelenmiş ve performansları birbirleriyle karşılaştırılmıştır.



Şekil 5.1. Bağımsız değişken sayısı p=5 ve gözlem sayısı 1'den 1000'e kadar olan benzetim verisi için RSIMPLS, PRM, PLS-SD, PLS-KurSD, *PLS-ARWMCD(N=20)*, *PLS-Smult(N=20)*, *PLS-MMmult(N=20)* yöntemlerinin saniye türünden hesaplama zamanları.

Şekil 5.1'den görüldüğü üzere, alt küme sayıları N=20 olarak seçilen yeni önerilen sağlam *PLS-ARWMCD(N=20), PLS-MMmult(N=20)* ve *PLS-Smult(N=20)* yöntemlerinin hesaplama zamanları literatürde var olan sağlam RSIMPLS ve PLS-SD yöntemlerinden daha azdır. Şekil 5.1'den, literatürde var olan sağlam RSIMPLS ile PLS-SD yöntemleri için veri kümesindeki gözlem sayısı arttıkça hesaplama zamanlarının da doğrusal olarak arttığı izlenmiştir. Ancak, PLS-SD için bu artışlar daha düşük bir oranda olmuştur. Literatürde var olan sağlam PRM yönteminin hesaplama zamanları, gözlem sayısı artışından çok etkilenmemiş ve neredeyse sabit kalmıştır. Literatürde var olan sağlam PLS-KurSD yötemi ile yeni önerilen sağlam *PLS-ARWMCD(N=20), PLS-Smult(N=20)* ve *PLS-MMmult(N=20)* yöntemlerinin hesaplama zamanları ise, gözlem sayısı artışından çok az etkilenmiştir.

İlkinci olarak gözlem sayısının n=100 olarak sabitlendiği ve bağımsız değişken sayısının 5'ten 40'a kadar arttırıldığı düzen için sonuçlar, Şekil 5.2'de verilmiştir. Bu şekilden yararlanarak, alt örneklem sayıları N=20 olarak seçildiğinde yeni önerilen sağlam *PLS-ARWMCD*, *PLS-Smult* ve *PLS-MMmult* yöntemlerinin ve literatürde var olan dört sağlam PLSR yönteminin veri kümesindeki bağımsız değişken sayısı arttıkça hesaplama zamanlarındaki değişimler incelenmiş ve performansları birbirleriyle karşılaştırılmıştır.



Şekil 5.2. Gözlem sayısı n=100 ve bağımsız değişken sayısı 1'den 40'a kadar olan benzetim verisi için RSIMPLS, PRM, PLS-SD, PLS-KurSD, *PLS-ARWMCD(N=20)*, *PLS-Smult(N=20)*, *PLS-MMmult(N=20)* yöntemlerinin saniye türünden hesaplama zamanları.

Şekil 5.2'den görüldüğü üzere, bağımsız değişken sayısı arttıkça yeni önerilen sağlam *PLS-ARWMCD(N=20),* *PLS-MMmult(N=20)* ve *PLS-Smult(N=20)* yöntemlerinin ve literatürde var olan sağlam PRM ile PLS-KurSD yöntemlerinin hesaplama zamanları neredeyse sabit kalmıştır. Aynı şekilde, literatüde var olan sağlam RSIMPLS yönteminin de hesaplama zamanı nerdeyse sabit kalmış olmasına karşın, RSIMPLS yönteminin hesaplanması bu yöntemlerden daha fazla zaman gerektirmektedir. Literatürde var olan sağlam PLS-SD yönteminin ise bağımsız değişken sayısı arttıkça hesaplama zamanlarının da doğrusal olarak arttığı ve bu yöntemin diğer sağlam yöntemlerden daha yavaş olduğu izlenmiştir.

EK 7'de, *PLS-ARWMCD*, *PLS-Smult* ve *PLS-MMmult* yöntemlerine ilişkin alt örneklem sayıları N=500 olarak seçildiğinde, yeni önerilen bu üç sağlam PLSR yönteminin ve literatürde var olan dört sağlam PLSR yönteminin hesaplama zamanlarının karşılaştırıldığı şekiller verilmiştir. EK 7'deki şekiller incelendiğinde, N=20 için elde edilen sonuçların aksine N=500 olarak seçildiğinde, yeni önerilen sağlam *PLS-ARWMCD(N=500)*, *PLS-Smult(N=500)*, PLS-*MMmult(N=500)*

yöntemlerinin veri kümesindeki gözlem sayısı ya da bağımsız değişken sayısı arttıkça hesaplama zamanlarının doğrusal olarak arttığı ve hesaplanmalarının diğer dört sağlam PLSR yönteminden daha fazla zaman aldığı görülür. Gözlem sayısı arttıkça, *PLS-Smult(N=500)* ile *PLS-MMmult(N=500)* yöntemlerinin hesaplama zamanlarındaki artış oranı *PLS-ARWMCD(N=500)* yönteminden daha fazlayken, bağımsız değişken sayısı arttıkça tam aksine *PLS-ARWMCD(N=500*) yönteminin hesaplama zamanlarındaki artış bu iki yöntemden daha fazladır. Tüm bu sonuçlardan genel olarak, seçilen alt örneklem sayılarına göre yeni önerilen üç sağlam PLSR yönteminin hesaplama zamanının değiştiğini söyleyebiliriz.

5.2. Gerçek Veri Kümesi Üzerinde Uygulama

Bu alt bölümde, aykırı değer içeren gerçek bir veri kümesi üzerinde önerilen üç yeni sağlam PLSR yöntemi klasik yöntem ve literatürde var olan dört sağlam PLSR yöntemi ile veriye uyum ve kestirimdeki başarıları bakımından karşılaştırılacaktır. Bu amaçla, Naes [24] çalışmasında söz edilen balık isimli veri kümesi kullanılacaktır. Bu veri kümesindeki bağımlı ve bağımsız değişkenleri tanımlamadan önce, ilk olarak tezin ikinci bölümünde söz edilen ‘ayarlama’ sözcüğü bu örnekten yola çıkılarak tekrar açıklanır. Ayarlama, bir aletin verdiği tepkiyi örneklerin (gözlemlerin) özelliklerine bağlamak için bir matematiksel model kurma sürecidir. Kestirim ise, modeli kullanarak verilen bir alet tepkisinde bir örneğin özelliklerini bulma sürecidir. Ayarlamada çoğunlukla ‘spektrum (spectrum)’ ile ‘konsantrasyon’ terimleri, sırasıyla ‘alet tepkisini’ ve ‘örnek özelliğini’ belirtmekte ve tartışmayı kolaylaştırmak için kullanılmaktadır. Çok değişkenli ayarlama ise bir aletten elde edilen birçok tepkiyi, bir örneğin özellik ya da özelliklerine bağlama sürecidir. X değişkenleri, spektroskopik (spectroscopic) ölçümler ve Y değişkenleri, örneğin konsantrasyon miktarları iken genelde çok değişkenli ayarlama, çok değişkenli regresyon çözümlemesinin bir uygulama alanı olarak görülebilir. Spektroskopinin (spectroscopy) amaçlardan biri, örneğin bir tahıl ya da etin kızıl ötesi yansıyan spektrumu (Near Infrared Reflectance/NIR) gibi kimyasal birleşimini tahmin etmektir [26].

Bu örnekte bağımlı değişken 45 tane balığın (gökkuşağı alabalığının) her birinin yağ konsantrasyonudur (%). Bağımsız değişkenler ise, örneklem homojenleştirildikten sonra bir NIR aletinde ölçülen 9 tane dalga boyundaki spektrumlardır. Bu veri kümesini kullanarak yapılan daha önceki çalışmalardan bağımsız değişkenlerin, yüksek derecede çoklubağlantılı olduğu biliniyor. Bu veri kümesine ilişkin analizin amacı, tek bir bağımlı değişken yağ konsantrasyonu ve bu spektrumlar arasındaki ilişkiyi modellemektir. Naes [24], 39-45 arasındaki son 7 gözlemin aykırı değer olduğunu belirtmiştir. Balık veri kümesi, EK 8'teki çizelgede verilmiştir [9, 12, 15, 24]. Bu veri kümesi, sağlam PLSR yöntemlerine ilişkin makalelerde sıklıkla kullanılmıştır.

Sağlam PLSR yöntemi PLS-SD'nin önerildiği Gil ve Romera [9] çalışmasında ise, balık veri kümesi iki parçaya ayrılmıştır. Sırasıyla, ilk 5, 10 ve 20 gözlemin test kümesi olarak ve geriye kalan 40, 35 ve 25 tane gözlemin çalışma kümesi olarak kullanıldığı 3 farklı veri kümesi oluşturulmuştur. Bu çalışmadan yola çıkılarak doktora tez çalışmamızda, bu 3 farklı veri kümesi üzerinde yeni önerilen üç sağlam PLSR yöntemi *PLS-ARWMCD*, *PLS-Smult*, *PLS-MMmult* ile klasik PLSR yöntemi ve literatürde var olan dört sağlam PLSR yöntemi RSIMPLS, PRM, PLS-SD, PLS-KurSD veriye uyum ve kestirim açısından sırasıyla Eş. (5.2) ve Eş. (5.3)'ü kullanarak karşılaştırılmıştır. Benzetim çalışmalarımızda da yaptığımız gibi, GOF değerleri hesaplanırken her bir çalışma kümesindeki 7 aykırı değer analizden çıkartılır. RMSE ölçütü hesaplanırken ise, ilk olarak 7 aykırı değerin de yer aldığı çalışma kümelerini kullanarak her bir yöntem için modeller oluşturulur. Daha sonra, bu modellerdeki regresyon katsayılarını kullanarak aykırı değerlerin yer almadığı test kümeleri için kestirimler elde edilir ve böylece önerdiğimiz üç yeni yöntemin de özellikle klasik yönteme karşın kestirim başarısı incelenmiş olur.

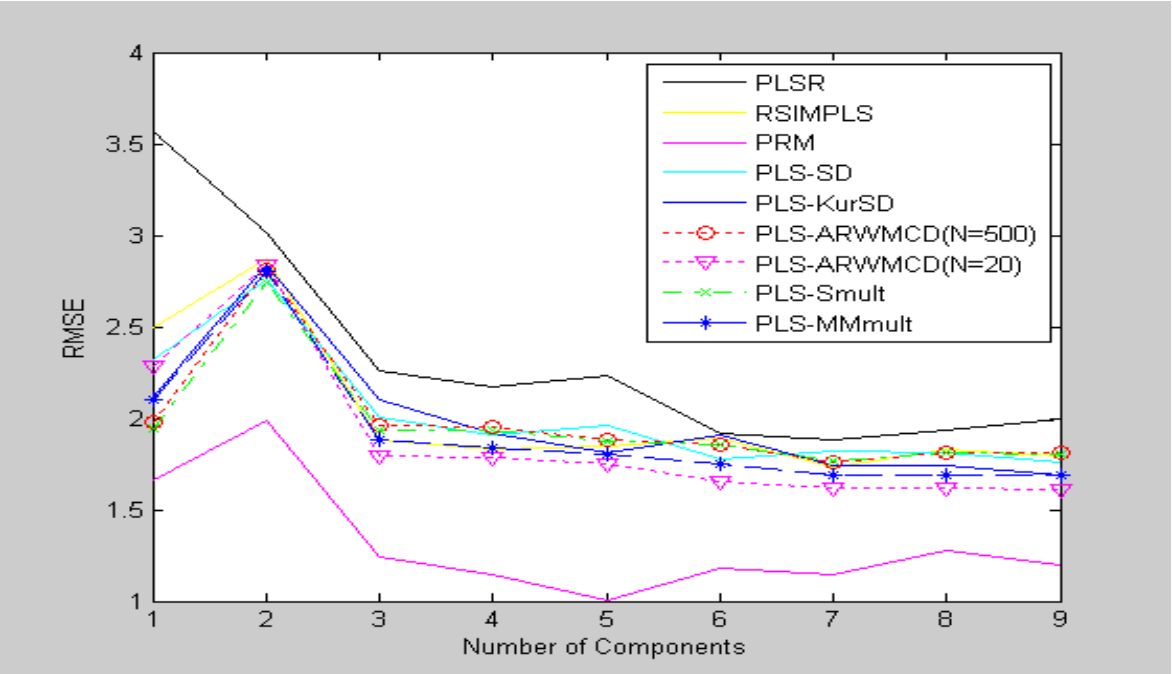
İlk veri kümesi için sonuçlar, Çizelge 5.7'de verilmiştir. Bu veri kümesi için alt küme sayısının N=20 ya da N=500 alınması sadece, *PLS-ARWMCD* yöntemine ilişkin sonuçları etkilemiştir. Bu nedenle, Çizelge 5.7'de bu iki alt küme sayısı için de sonuçlar verimiştir. *PLS-ARWMCD(N=20)* için elde edilen sonuçlar, parantez içinde koyu olarak yazılmıştır.

Çizelge 5.7. Çalışma kümesi 40 ve test Kümesi 5 gözlemli balık verisi için GOF ve RMSE değerleri.

Modeldeki Bileşen Sayısı		PLSR	RSIMPLS	PRM	PLS-SD	PLS-KurSD	***PLS-ARWMCD***	***PLS-Smult***	***PLS-MMmult***
k=1	GOF	0.4012	0.5407	0.4582	0.5420	0.5388	0.5306 **(0.5430)**	0.5263	0.5383
	RMSE	3.5624	2.4939	1.6635	2.3197	2.1207	1.9814 **(2.2854)**	1.9402	2.0983
k=2	GOF	0.7733	0.8333	0.4228	0.8556	0.8652	0.8286 **(0.8395)**	0.8431	0.8422
	RMSE	3.0175	2.8755	1.9905	2.7618	2.8415	2.8075 **(2.8379)**	2.7439	2.8006
k=3	GOF	0.9240	0.9624	0.6813	0.9603	0.9502	0.9608 **(0.9624)**	0.9608	0.9625
	RMSE	2.2604	1.8794	1.2443	2.0029	2.1007	1.9591 **(1.7896)**	1.9382	1.8773
k=4	GOF	0.9291	0.9621	0.6787	0.9583	0.9598	0.9591 **(0.9604)**	0.9597	0.9645
	RMSE	2.1734	1.8312	1.1445	1.9081	1.9179	1.9488 **(1.7890)**	1.9307	1.8419
k=5	GOF	0.9337	0.9668	0.6793	0.9654	0.9685	0.9669 **(0.9679)**	0.9669	0.9702
	RMSE	2.2326	1.8506	1.0011	1.9618	1.8130	1.8812 **(1.7541)**	1.8737	1.8022
k=6	GOF	0.9377	0.9633	0.8048	0.9695	0.9628	0.9672 **(0.9711)**	0.9666	0.9726
	RMSE	1.9128	1.8871	1.1785	1.7740	1.9087	1.8542 **(1.6556)**	1.8558	1.7497
k=7	GOF	0.9407	0.9679	0.8135	0.9670	0.9405	0.9687 **(0.9701)**	0.9674	0.9733
	RMSE	1.8834	1.7305	1.1420	1.8190	1.7419	1.7585 **(1.6180)**	1.7700	1.6891
k=8	GOF	0.9432	0.9600	0.8187	0.9686	0.9370	0.9650 **(0.9702)**	0.9646	0.9731
	RMSE	1.9318	1.8257	1.2743	1.8119	1.7429	1.8091 **(1.6205)**	1.8099	1.6871
k=9	GOF	0.9424	0.9662	0.8186	0.9690	0.9478	0.9658 **(0.9707)**	0.9654	0.9734
	RMSE	1.9935	1.7920	1.1997	1.7623	1.6864	1.8082 **(1.6115)**	1.8023	1.6871

Bu veri kümesi için modelde kalacak ideal bileşen sayısına karar verirken GOF ya da RMSE değerlerine bakabiliriz. GOF değerlerinin daha fazla değişmediği bileşen sayısı ideal bileşen sayısı gibi seçilebilir. Ancak, ilk benzetim çalışmamızda da

belirttiğimiz üzere ideal bileşen sayısına RMSE değerlerine göre karar vermek daha uygundur. Modelde kalacak ideal bileşen sayısına karar verirken RMSE'nin mümkün en küçük değeri almasından çok, modele ilave edilen bileşenin RMSE'de önemli bir düşüş yaratıp yaratmadığı da dikkate alınmalıdır. Böylece, gereksiz bir bileşen modele eklenmemiş ve veri indirgeme amacından da sapılmamış olur. Şekil 5.3'te bileşen sayısına karşın RMSE değerlerinin grafiği çizilmiştir. Bu şekil incelendiğinde, modelde kalacak ideal bileşen sayısının üç olarak seçilmesinin doğru olduğu görülür. Çünkü, şekilden görüldüğü üzere modele üçüncü bileşenin eklenmesi tüm yöntemler için RMSE değerlerinde önemli bir düşüşe neden olmuştur. Çizelge 5.7'den görüldüğü üzere, modelde üç bileşen olduğunda tüm yöntemler için veriye uyum da çok iyidir.



Şekil 5.3. Çalışma kümesi 40 ve test kümesi 5 gözlemli balık verisi için modeldeki bileşen sayısına karşı RMSE değerleri.

İlk veri kümesi için modelde kalacak ideal bileşen sayısı k=3 olarak seçildiğinde, Çizelge 5.7'den yeni önerilen sağlam *PLS-ARWMCD(N=500)*, *PLS-ARWMCD(N=20)*, *PLS-Smult*, *PLS-MMmult* yöntemlerinin hem klasik PLSR yönteminden hem de literatürde var olan sağlam PLS-SD ve PLS-KurSD yöntemlerinden kestirim açısından daha iyi sonuçlar verdiğini söyleyebiliriz. Bu üç bileşenli model için yeni önerilen sağlam *PLS-MMmult* yöntemi, literatürde var olan sağlam RSIMPLS yönteminin kestirimdeki başarısını yakalamıştır. Ayrıca, yeni önerilen bu üç sağlam yöntemin veriye uyumları da literatürde var olan sağlam PRM yönteminden çok daha iyidir. Çizelge 5.7 incelendiğinde genel olarak, *PLS-ARWMCD* yöntemi için N=500 yerine N=20 olacak şekilde daha az sayıda alt küme seçildiğinde, yöntemin veriye uyum ve kestirimdeki başarısının arttığı söylenebilir.

Çalışma kümesi 35 ve test kümesi 10 gözlemli olan ikinci veri kümesi için sonuçlar, EK 9'taki çizelgede verilmiştir. Bu veri kümesi için alt küme sayısının N=20 ya da N=500 alınması, yeni önerilen sağlam *PLS-AWMCD*, *PLS-Smult* ve *PLS-MMmult* yöntemlerine ilişkin GOF ve RMSE değerlerini, dolayısıyla veriye uyum ve kestirimdeki başarılarını etkilememiştir. Bu veri kümesi için de ideal bileşen sayısı, bir önceki veri kümesi ile benzer nedenlerden dolayı üç olarak belirlenmiştir. EK 9'taki RMSE değerlerine ilişkin şekil incelendiğinde, modele üçüncü bileşen eklendiğinde tüm yöntemler için RMSE değerlerinde önemli bir düşüş yaşandığı görülür. Modelde kalacak ideal bileşen sayısı k=3 olarak seçildiğinde, yeni önerilen sağlam *PLS-Smult* ile *PLS-MMmult* yöntemleri, klasik PLSR yönteminden ve literatürde var olan sağlam PRM yönteminden hem veriye uyum hem de kestirim açısından daha başarılı sonuçlar vermiştir. Ayrıca, yeni önerilen sağlam *PLS-Smult* ile *PLS-MMmult* yöntemlerinin literatürde var olan sağlam PLS-SD yönteminden de, kestirim açısından daha iyi sonuçlar verdiği görülür.

Çalışma kümesi 25 ve test kümesi 20 gözlemli olan üçüncü veri kümesi için sonuçlar ise, EK 10'daki çizelge ve şekilde verilmiştir. Bu veri kümesi için de, ideal bileşen sayısı önceki iki veri kümesi ile aynı nedenle üç olarak belirlenmiştir.

Modelde kalacak ideal bileşen sayısı k=3 olarak seçildiğinde, yeni önerilen sağlam *PLS-Smult* ile *PLS-MMmult* yöntemleri bu veri kümesi için, klasik PLSR yönteminden ve öncelikle PRM olmak üzere literatürde var olan sağlam RSIMPLS ve PLS-KurSD yöntemlerinden kestirim açısından daha başarılı sonuçlar vermiştir. Üç bileşenli model için, yeni önerilen sağlam *PLS-ARWMCD* yönteminin ise literatürde var olan sağlam PRM yönteminden veriye uyum ve kestirim açısından daha iyi sonuçlar verdiği görülür.

6. SONUÇ VE TARTIŞMA

Bu çalışmada, tek bir bağımlı değişkenin olduğu doğrusal regresyon modelinde veri kümesi aykırı değerler tarafından bozulduğunda sağlam kestirimler veren üç yeni sağlam PLSR yöntemi önerilmiştir.

Çalışmanın birinci bölümünde kısaca sağlam PLSR yöntemlerinin ortaya çıkışı ve hangi yöntemleri kullanarak geliştirildiği, literatürde bu konuda yapılan önceki çalışmalar, tezin içeriği ve tezde izlenecek düzen ana hatları ile verilmiştir. İkinci bölümde, klasik PLSR yöntemi hakkında genel bilgi verilmiş ve bu yöntem ile parametre kestirimlerini elde etmek için en yaygın kullanılan yinelemeli klasik PLSR algoritmaları incelenmiştir. Üçüncü bölümde, literatürde var olan sağlam PLSR yöntemleri ayrıntılı olarak incelenmiştir. Dördüncü bölümde, doktora tezimizde önerilen üç yeni sağlam PLSR yöntemi tanıtılmıştır.

Bu çalışmanın beşinci bölümünde ise, yeni önerilen üç sağlam PLSR yöntemi *PLS-ARWMCD, PLS-Smult, PLS-MMmult* ile klasik PLSR ve literatürde var olan dört sağlam PLSR yöntemi RSIMPLS, PRM, PLS-SD, PLS-KurSD'yi etkinlik, veriye uyum ve kestirimdeki başarıları bakımından karşılaştırmak amacıyla üç benzetim çalışması yapılmıştır. Ayrıca, veri kümesindeki bağımsız değişken sayısı sabitken gözlem sayısı arttığında ya da gözlem sayısı sabitken bağımsız değişken sayısı arttığında, yeni önerilen üç sağlam PLSR yönteminin hesaplama zamanlarındaki değişimler literatürde var olan diğer dört sağlam PLSR yöntemi ile karşılaştırılmıştır. Bu benzetim çalışmalarına ilave olarak, literatürde çok sık kullanılan gerçek bir veri kümesi üzerinde de, bu yöntemler veriye uyum ve kestirimdeki başarıları bakımından karşılaştırılmıştır.

İlk benzetim çalışmasında orta büyüklükteki bir örneklem (n=100, p=6) ve nispeten daha küçük bir örneklem (n=50, p=10) için temiz veri kümesi, % 10 oranında kötü kaldıraç gözlemleri ve dikey aykırı değerler varlığında yeni önerilen üç sağlam PLSR yöntemi, klasik PLSR yöntemi ve literatürde var olan dört sağlam PLSR

yöntemi RSIMPLS, PRM, PLS-SD ve PLS-KurSD ile etkinlik, veriye uyum ve kestirimdeki başarısı bakımından karşılaştırılmıştır. Ayrıca bu benzetim düzeninde, RMSE ölçütünden faydalanarak modelde kalacak ideal bileşen sayısının nasıl belirlenebileceği hakkında da bilgi verilmiştir. Bu benzetim düzeni için elde edilen sonuçlara göre, temiz veri kümesi için genel olarak klasik PLSR yöntemi daha iyi bir performansa sahip olmasına karşın, k_{opt}=2 için yeni önerilen sağlam *PLS-Smult* ile *PLS-MMmult* yöntemlerinin etkinlik, veriye uyum ve kestirim açısından klasik PLSR yöntemine yakın bir performansa sahip olduğu görülmüştür. Ancak, veri kümesinde kötü kaldıraç gözlemleri ya da dikey aykırı değerler olduğunda yeni önerilen üç sağlam PLSR yönteminin, her üç ölçüt bakımından da, klasik PLSR yönteminden daha iyi bir performansa sahip olduğu izlenmiştir. Her iki benzetim düzeninden elde edilen sonuçlar birlikte yorumlandığında; kötü kaldıraç gözlemleri varlığında ve modelde kalacak ideal bileşen sayısı k_{opt}=2 olarak seçildiğinde, en etkin yöntemlerin yeni önerdiğimiz sağlam *PLS-ARWMCD*, *PLS-MMmult*, *PLS-Smult* yöntemleri ile literatürde var olan sağlam RSIMPLS yöntemi olduğu görülmüştür. Ayrıca, k_{opt}=2 bileşenli model için kötü kaldıraç gözlemleri varlığında yeni önerilen üç sağlam yöntemin de, literatürde var olan sağlam PRM ve PLS-SD yöntemlerinden kestirim açısından daha başarılı olduğu bulunmuştur. Veri kümesinde dikey aykırı değerler olduğunda ve modelde kalacak ideal bileşen sayısı k_{opt}=2 olarak seçildiğinde en etkin yöntemler, yeni önerdiğimiz sağlam *PLS-MMmult* ve *PLS-Smult* yöntemleri ile PRM yöntemidir. Bu benzetim düzeni için elde edilen sonuçlar incelendiğinde, genel olarak yeni önerilen üç sağlam PLSR yöntemindeki algoritmalarda kullanılan alt küme sayılarının (N) farklı seçilmesi sonuçları kısmen etkilemiştir. Ancak, bu etki yöntemler için çok büyük farklılıklara yol açmamıştır.

İkinci benzetim çalışmasında ise temiz veri kümesi, kötü kaldıraç gözlemleri, dikey aykırı değerler, iyi kaldıraç gözlemleri, kümelenmiş aykırı değerler ve dik aykırı değerler varlığında yeni önerilen üç sağlam PLSR yöntemi klasik PLSR yöntemi ve literatürde var olan dört sağlam PLSR yöntemi ile etkinlik, veriye uyum ve kestirim başarısı bakımından karşılaştırılmıştır. Bu benzetim düzeninde ideal bileşen sayısı ise, başlangıçta k=2 olarak alınmıştır. Ayrıca, örneklem büyüklüğünün etkisini görmek amacıyla sırasıyla, n=200 ve n=1000 olmak üzere orta ve büyük

örneklemler ile çalışılmıştır. Bu iki örneklem için de veri kümesinin sırasıyla, %10 ve % 20'si aykırı değerler ile yer değiştirilmiştir. Böylece veri kümesindeki aykırı değer yüzdesi arttığı zaman, yeni önerilen üç sağlam PLSR yöntemi ve literatürde var olan dört sağlam yönteminin performanslarının nasıl etkilendiği incelenmiştir. Buna göre, orta büyüklüklü örneklem için veri kümesinin % 10'u çeşitli aykırı değerler tarafından bozulduğunda yeni önerilen üç sağlam PLSR yöntemi *PLS-ARWMCD*, *PLS-Smult* ve *PLS-MMmult* ile literatürde var olan dört sağlam PLSR yönteminin klasik yönteme karşı etkinlik ve kestirim açısından üstünlükleri açık bir şekilde görülür. Yeni önerilen üç sağlam PLSR yöntemi de her bir aykırı değer türünde etkinlik, veriye uyum ve kestirimde gösterdikleri yakın başarı ile literatürde var olan sağlam PRM ve PLS-SD yöntemlerine iyi birer seçenek yöntem olarak öne çıkmıştır. Ayrıca, tüm aykırı değer türlerinde bu üç yeni sağlam PLSR yöntemi, literatürde var olan sağlam RSIMPLS ve PLS-KurSD yöntemleri ile veriye uyum, kestirim ve ortalama açı ölçütleri bakımından yakın bir performans göstermiştir. Veri kümesindeki aykırı değer yüzdesi % 20 gibi yüksek bir düzeye çıktığında literatürde var olan sağlam PRM yönteminin, dikey aykırı değerler hariç diğer tüm aykırı değer türleri için etkinlik, veriye uyum ve kestirimdeki başarısının düştüğü görülür. Sağlam PRM yöntemi, veri kümesinde % 20 oranında iyi kaldıraç gözlemleri, kümelenmiş aykırı değerler ya da dik aykırı değerler olduğunda, klasik PLSR yöntemine karşın başarısını tamamen kaybetmiştir. Literatürde var olan sağlam PLS-SD yönteminin de, veri kümesinde yüksek bir oranda kötü kaldıraç gözlemleri ya da kümelenmiş aykırı değerler olduğunda, PRM hariç diğer sağlam yöntemlere göre etkinlik ve kestirimdeki başarısının düştüğü görülmüştür. Yeni önerilen üç sağlam PLSR yöntemi ise, tüm aykırı değer türleri için başarılarını korur. Büyük örneklem için aykırı değer yüzdesi % 10 iken elde edilen sonuçlar incelendiğinde, genel olarak tüm aykırı değer türleri için yeni önerilen sağlam *PLS-ARWMCD*, *PLS-Smult*, *PLS-MMmult* yöntemlerinin literatürde var olan sağlam RSIMPLS ve PLS-KurSD yöntemleri ile beraber etkinlik, veriye uyum ve kestirim açısından çok yakın sonuçlarıyla en başarılı yöntemler olduğu görülür. Büyük örneklem için veri kümesindeki aykırı değer yüzdesi arttığında, sağlam PRM ve PLS-SD yöntemleri için orta büyüklükteki örnekleme benzer sonuçlar elde edilmiştir. Genel olarak, gözlem sayısı arttıkça yeni önerilen üç sağlam PLSR yöntemi ile gerçek parametre vektörünün daha başarılı bir şekilde kestirildiğini söyleyebiliriz. Büyük örneklem için veri kümesindeki aykırı değer yüzdesi

arttığında, yeni önerilen sağlam PLSR yöntemleri PLS-ARWMCD, *PLS-Smult* ve *PLS-MMmult* her bir aykırı değer türünde etkinlik ve sağlamlıktaki başarılarını korumuştur. Orta büyüklükteki örneklemde % 20 oranında dikey aykırı değer var olduğunda, yeni önerilen sağlam *PLS-Smult* ve *PLS-MMmult* yöntemleri diğer sağlam yöntemler ile karşılaştırıldığında kestirim açısından daha başarısızken, her iki yöntem de örneklem büyüklüğü arttığında diğer sağlam yöntemlerin kestirimdeki başarısını yakalamıştır. Bu benzetim düzeni için elde edilen sonuçlar incelendiğinde, genel olarak yeni önerilen üç sağlam PLSR yöntemindeki algoritmalarda kullanılan alt küme sayılarının (N) az ya da çok seçilmesi veri kümesindeki aykırı değer yüzdesi % 10 olduğunda sonuçları etkilememiştir. Ancak, veri kümesindeki aykırı değer yüzdesi % 20 gibi yüksek bir düzeye çıktığında yeni önerilen sağlam *PLS-Smult* ile *PLS-MMmult* yöntemlerine ilişkin sonuçlar N=20 gibi küçük bir alt küme sayısı seçildiğinde olumsuz etkilenmiştir. Özellikle veri kümesinde % 20 oranında dikey aykırı değerler olduğunda, *PLS-Smult* ile *PLS-MMmult* yöntemleri için seçilen alt küme sayılarının çok az olmasının, yöntemlerin etkinlik ve sağlamlıktaki performanslarını olumsuz yönde çok belirgin bir şekilde etkilediği görülür. Bu nedenle, veri kümesinde % 20 oranında aykırı değerler olduğunda *PLS-Smult* ile *PLS-MMmult* yöntemleri için N=500 olmak üzere seçilen alt küme sayısı arttırıldığında daha iyi sonuçlar elde edildiği ve bu iki yeni sağlam PLSR yönteminin literatürdeki sağlam RSIMPLS ile PLS-KurSD yöntemlerinin performanslarını yakaladığı görülmüştür.

Üçüncü benzetim çalışmasında, hata terimleri normal dağılımdan farklı bir dağılımdan geldiğinde özellikle yeni önerilen üç sağlam PLSR yönteminin etkinlikteki başarısı incelenmiştir. Bu amaçla, ‘Standart Normal’ dağılımdan farklı olarak ‘Laplace’, 5 ve 2 serbestlik dereceli ‘t-dağılımları’ ile ‘Cauchy’ ve ‘Slash’ gibi ağır kuyruklu dağılımlardan gelen hata terimleri için modeller oluşturulmuştur. Bu benzetim düzeninde ilk olarak, literatürdeki çalışmalardan yola çıkılarak gözlem sayısının ve bağımsız değişken sayısının farklı seçildiği üç örneklem şeması ile çalışılmıştır. Sonuç olarak, her üç örneklem şeması için de t_2, Cauchy ve Slash gibi aykırı değer daha çok yaratan dağılımlara ilişkin sonuçlar incelendiğinde, yeni önerilen sağlam *PLS-ARWMCD, PLS-Smult* ve *PLS-MMmult* yöntemlerinin klasik PLSR yöntemi ile literatürde var olan sağlam PLS-KurSD yönteminden etkinlik

bakımından daha başarılı olduğu görülmüştür. Bu benzetim düzeninde ikinci olarak, p=5 ve k=2 olarak belirlenmiş ve gözlem sayısı sırasıyla 20, 200 ve 1000 olarak seçilmiştir. Sonuç olarak, küçük, orta ve büyük her üç örneklem için de özellikle t_2, Cauchy ve Slash gibi aykırı değer daha çok yaratan dağılımlara ilişkin sonuçlar incelendiğinde, yeni önerilen üç sağlam PLSR yönteminin ve literatürde var olan sağlam PLSR yöntemlerinin klasik PLSR yöntemine karşın etkinlikteki üstünlüğü açık bir şekilde görülmüştür.

Gerçek veri kümesi için elde edilen sonuçlar incelendiğinde ise, ilk olarak 45 tane balığın yağ konsantrasyonunun ölçüldüğü veri kümesinden ilk 5, 10 ve 20 gözlemin test kümesi ve geriye kalan 40, 35 ve 25 tane gözlemin çalışma kümesi olarak kullanıldığı 3 farklı veri kümesi oluşturulmuştur. Daha sonra, her üç veri kümesi için de RMSE değerlerinden yararlanılarak ideal bileşen sayısı k=3 olarak seçilmiştir. Bu veri kümelerinde yapılan uygulamalardan modelde kalacak ideal bileşen sayıs k=3 olarak seçildiğinde, yeni önerilen sağlam *PLS-ARWMCD*, *PLS-Smult* ve *PLS-MMmult* yöntemleri için elde edilen GOF değerlerinin hem klasik PLSR yönteminden hem de literatürde var olan sağlam PRM yönteminden daha büyük olduğu görülür. Bu yeni önerilen üç sağlam yöntem için RMSE değerlerinin ise, her üç veri kümesi için de k=3 olarak belirlendiğinde klasik PLSR yönteminden daha küçük olduğu görülür. Genel olarak, modelde kalacak ideal bileşen sayısından bağımsız olarak yeni önerilen üç sağlam PLSR yönteminin de veriye uyum ve kestirim açısından klasik PLSR yöntemine seçenek daha iyi modeller verdiğini söyleyebiliriz.

Sonuç olarak, benzetim çalışmalarından yola çıkılarak, yeni önerilen üç sağlam PLSR yöntemi *PLS-ARWMCD*, *PLS-Smult* ve *PLS-MMmult*'un küçük boyutlu ve makul düzeyde aykırı değerler tarafından bozulan veri kümelerine uygulandığında özellikle klasik PLSR yönteminden daha sağlam ve etkin sonuçlar verdiği görülmüştür. Genel olarak, veri kümesinde % 10 ya da % 20 oranında kötü kaldıraç gözlemleri olduğunda yeni önerilen üç sağlam PLSR yöntemi de literatürde var olan sağlam PRM ve PLS-SD yöntemlerinden etkinlik ve kestirim açısından daha başarılıdır. Veri kümesinde % 10 oranında dikey aykırı değerler

bulunduğunda ise, yeni önerilen üç sağlam PLSR yöntemi etkinlik ve kestirimdeki performansları açısından literatürde var olan dört sağlam yönteme yakın bir başarıya sahiptir. Veri kümesinde % 10 ya da % 20 oranında iyi kaldıraç gözlemleri olduğunda, yeni önerilen üç sağlam PLSR yöntemi de PRM yönteminden etkinlik ve kestirim açısından, PLS-SD yönteminden ise sadece etkinlik açısından daha başarılıdır. Veri kümesinde % 10 oranında kümelenmiş aykırı değerler varlığında ise, yeni önerilen üç sağlam PLSR yöntemi de literatürde var olan sağlam PRM yönteminden etkinlik ve kestirim açısından, PLS-SD yönteminden ise etkinlik açısından daha başarılıdır. Veri kümesinde % 20 oranında kümelenmiş aykırı değerler bulunduğunda ise yeni önerilen üç sağlam PLSR yöntemi de literatürde var olan sağlam PRM ve PLS-SD yöntemlerinin her ikisinden de etkinlik ve kestirim açısından daha başarılıdır. Veri kümesinde % 10 ya da % 20 oranında dik aykırı değerler olduğunda ise, yeni önerilen üç sağlam PLSR yöntemi de PRM yönteminden etkinlik, veriye uyum ve kestirim açısından daha başarılıdır.

KAYNAKLAR

[1] Cummins, D. J., Andrews, C. W., Iteratİvely Reweighted Partial Least Squares: A Performance Analysis By Monte Carlo Simulation, Journal of Chemometrics, 9, 489-507, **1995**.

[2] De Jong, S., SIMPLS: An alternative approach to partial least squares regression, *Chemometrics and Intelligent Laboratory Systems*, 18, 251-263, **1993**.

[3] Denham, M. C., Implementing Partial Least Squares, *Statistics and Computing*, 5, 191-202, **1995**.

[4] Dodge, Y.; Kondylis, A.; Whittaker, J., Extending PLS1 to PLAD regression and the use of the L1 norm in soft modelling, *In: Compstat 2004: Proceedings in Computational Statistics*, Antoch, J., Ed., Springer-Verlag, Heidelberg, pp. 935-942, **2004**.

[5] Engelen, S., Hubert, M., Vanden Branden, K., Verboven, S., Robust PCR and Robust PLSR: A comparative study, *In Theory and Applications of Recent Robust Methods* (M. Hubert, G. Pison, A. Struyf and S. V. Aelst, eds.), Birkhäuser, Basel, 105–117, **2004**.

[6] Ergül, B., *Robust Regresyon ve Uygulamaları*, Yüksek Lisans Tezi, Eskişehir Osmangazi Üniversitesi Fen Bilimleri Enstitüsü İstatistik Anabilim Dalı, Eskişehir, **2006**.

[7] Filzmoser P., Serneels S., Maronna R., Van Espen, P. J., *Robust multivariate methods in chemometrics*, In B. Walczak, R.T. Ferre, and S. Brown, editors, Comprehensive Chemometrics, 681-722, **2009**.

[8] Gervini, D., A robust and efficient adaptive reweighted estimator of multivariate location and scatter, *Journal of Multivariate Analysis*, 84,116–144, **2003**.

[9] Gil, J. A.; Romera, R., On robust partial least squares (PLS) methods, *Journal of Chemometrics*, 12, 365-378, **1998**.

[10] González, J., Peña, D., Romera, R., A robust partial least squares regression method with applications, *Journal of Chemometrics*, 23, 78–90, **2009**.

[11] Griep, M. I., Wakeling, I. N.,Vankeerberghen, P., Massart, D. L., Comparison of semirobust and robust partial least squares procedures, *Chemometrics and Intelligent Laboratory Systems*, 29, 37-50, **1995**.

[12] Hardy, A. J., MacLaurin, P., Haswell, S. J., De Jong, S., Vandeginste, B. G. M., Double-case diagnostic for outliers identification, *Chemometrics and Intelligent Laboratory Systems***,** 34, 117-129**, 1996**.

[13] Helland, I.S., Partial least squares regression and statistical methods, *Scandinavian Journal of Statistics,* 17, 97-114,**1990**.

[14] Holland, P. W., Welsch, R. E., Robust Regression Using Iteratively Reweighted Least-Squares, *Communications in Statistics - Theory and Methods*, Volume 6, Issue 9, 813-827, **1977**.

[15] Hubert, M., Vanden Branden, K., Robust methods for Partial Least Squares Regression, *Journal of Chemometrics*, 17, 537-549, **2003**.

[16] Huber, P., Ronchetti, E., M., *Robust Statistics*. Wiley, 2nd edition, **2009**.

[17] Hubert, M., Rousseeuw, P.J., Verdonck, T., A deterministic algorithm for robust location and scatter, *Journal of Computational and Graphical Statistics*, 21, 618-637, **2012**.

[18] Kayhan, Y., *Regresyon Çözümlemesinde Sağlam Konum Kestiricilerinin Karşılaştırılması*, Yüksek Lisans Tezi, Hacettepe Üniversitesi Fen Bilimleri Enstitüsü İstatistik Anabilim Dalı, Ankara, **2006**.

[19] Kondylis, A., Hadi, A. S., Derived components regression using the BACON algorithm, *Computational Statistics & Data Analysis*, 51(2), 556-569, **2006**.

[20] Kruger, U., Zhou, Y., Wang, X., Rooney, D., Thompson, J., Robust partial least squares regression: part I, algorithmic developments, *Journal of Chemometrics*, 22,1–13, **2008**.

[21] Liebmann, B., Filzmoser, P., Varmuza, K., Robust and Classical PLS Regression Compared, *Journal of Chemometrics*, 24 (3-4), 111-120, **2010.**

[22] Martens, H., Naes, T., *Multivariate Calibration*, John Wiley & Sons, Chichester, New York, Brisbane, Toronto, Singapore, **1989**.

[23] Moller, S. F., von Frese, J., Bro, R., Robust Methods for Multivariate Data Analysis, *Journal of Chemometrics*, 19(10), 549–563, **2005.**

[24] Naes, T., Multivariate calibration when the error covariance matrix is structured, *Technometrics*, 27:3, 301-311, **1985**.

[25] Pena, D., Prieto, J., Combining random and specific directions for outlier detection and robust estimation of high-dimensional multivariate data, *Journal of Computational and Graphical Statistics*, Volume 16, 228–254, **2007**.

[26] Polat, E., *Kısmi En Küçük Kareler Regresyonu*, Yüksek Lisans Tezi, Hacettepe Üniversitesi Fen Bilimleri Enstitüsü İstatistik Anabilim Dalı, Ankara, **2009**.

[27] Riani, M., Perrotta, D., Torti, F., FSDA: A MATLAB toolbox for robust analysis and interactive data exploration, *Chemometrics and Intelligent Laboratory Systems*, 116, 17–32, **2012**.

[28] Rousseeuw, P. J., Leroy, A., *Robust Regression and Outlier Detection*, New York: John Wiley, **1987**.

[29] Rousseeuw, P.J., Van Driessen, K., A fast algorithm for the minimum covariance determinant estimator, *Technometrics*, 41, 212–224, **1999**.

[30] Salibian-Barrera, M., Van Aelst, S. and Willems, G., PCA based on multivariate MM-estimators with fast and robust bootstrap, *Journal of the American Statistical Association*, 101, 1198-1211, **2006**.

[31] Serneels, S., Croux, C., Filzmoser, P., Van Espen, P. J., Partial Robust M-regression, *Chemometrics and Intelligent Laboratory Systems*, 79, 55-64, **2005**.

[32] Serneels, S., De Nolf, E., Van Espen, P. J., Spatial sign pre-processing: a simple way to impart moderate robustness to multivariate estimators, *Journal of Chemical Information and Modeling*, 46, 1402-1409, **2006**.

[33] Vanden Branden, K., Hubert, M., Robustness properties of a robust partial least squares regression method, *Analytica Chimica Acta*, 515, 229—241, **2004**.

[34] Verboven, S., Hubert, M., LIBRA: a MATLAB library for robust analysis, http://www.google.com.tr/url?sa=t&rct=j&q=&esrc=s&source=web&cd=2&ved=0CDAQFjAB&url=http%3A%2F%2Fwww.researchgate.net%2Fpublication%2F24502

3640_LIBRA_a_MATLAB_library_for_robust_analysis%2Ffile%2F3deec51d5b0c86dceb.pdf&ei=sS0RUp8fzc2zBvmzgJAO&usg=AFQjCNG847MijHcfD66QpNModVXTrIl73g&bvm=bv.50768961,d.Yms (Ağustos, **2013**).

[35] Visuri, S., Koivunen, V., Oja, H., Sign and rank covariance matrices, *Journal of Statistical Planning and Inference*, 91, 557-575, **2000**.

[36] Vural, A., *Aykırı Değerlerin Regresyon Modellerine Etkileri ve Sağlam Kestiriciler*, Yüksek Lisans Tezi, Marmara Üniversitesi Sosyal Bilimler Enstitüsü Ekonometri Anabilim Dalı Ekonometri Bilim Dalı, İstanbul, **2007**.

[37] Wakeling, I. N., Macfie, H. J. H., A Robust PLS Procedure, *Journal of Chemometrics*, 6, 189-198, **1992**.

[38] Wold, S., Sjöström, M., Eriksson, L., PLS-regression: a basic tool of chemometrics, *Chemometrics and Intelligent Laboratory Systems*, 58, 109-130, **2001**.

[39] Yorulmaz, Ö., *Dayanıklı Regresyon Yöntemi ve Çeşitli Sosyal Veriler Üzerinde Aykırı Gözlemlerin Teşhisi*, Balıkesir Üniversitesi Sosyal Bilimler Enstitüsü Dergisi, 21, 76-88 s, **2009**.

EKLER

EK 1: n=50 ve p=10, TEMİZ VERİ KÜMESİ, m=1000 TEKRAR İÇİN BENZETİM SONUÇLARINA İLİŞKİN ÇİZELGE

Modeldeki Bileşen Sayısı		PLSR	RSIMPLS	PRM	PLS-SD	PLS-KurSD	***PLS-ARWMCD***	***PLS-Smult***	***PLS-MMmult***
k=1	MSE	0.5526	0.5352	0.5597	0.5532	0.5645	0.5563 **(0.5539)**	0.5526	0.5527
	GOF	0.7982	0.7997	0.7960	0.7970	0.7930	0.7952 **(0.7958)**	0.7977	0.7978
	RMSE	1.4925	1.4845	1.4977	1.4949	1.5085	1.5000 **(1.4986)**	1.4935	1.4933
k=2	MSE	0.0338	0.0519	0.0357	0.0432	0.0730	0.0601 **(0.0566)**	0.0379	0.0369
	GOF	0.8935	0.8916	0.8921	0.8922	0.8883	0.8898 **(0.8902)**	0.8930	0.8931
	RMSE	1.1135	1.1242	1.1165	1.1200	1.1380	1.1309 **(1.1296)**	1.1162	1.1155
k=3	MSE	1.7041	1.4097	1.5228	1.9331	3.1495	2.7806 **(2.6758)**	1.8428	1.8075
	GOF	0.9082	0.9007	0.9038	0.9029	0.8832	0.8905 **(0.8923)**	0.9057	0.9063
	RMSE	1.1883	1.1860	1.1829	1.2021	1.2760	1.2562 **(1.2506)**	1.1973	1.1951

EK 2: n=50 ve p=10, GÖZLEMLERİN İLK %10'U KÖTÜ KALDIRAÇ GÖZLEMLERİ, m=1000 TEKRAR İÇİN BENZETİM SONUÇLARINA İLİŞKİN ÇİZELGE

Modeldeki Bileşen Sayısı		PLSR	RSIMPLS	PRM	PLS-SD	PLS-KurSD	***PLS-ARWMCD***	***PLS-Smult***	***PLS-MMmult***
k=1	MSE	1.5875	0.5107	0.6115	0.5867	0.5694	0.5577 **(0.5588)**	0.5561	0.5561 **(0.5570)**
	GOF	0.2874	0.8087	0.7894	0.7955	0.7967	0.8010	0.8023	0.8024 **(0.8018)**
	RMSE	2.7942	1.4646	1.5389	1.5145	1.5079	1.4961 **(1.4964)**	1.4931	1.4931 **(1.4938)**
k=2	MSE	2.3084	0.0508	0.1136	0.2354	0.0891	0.0532 **(0.0521)**	0.0404	0.0398
	GOF	0.3992	0.8937	0.8827	0.8639	0.8872	0.8927 **(0.8930)**	0.8944	0.8945
	RMSE	2.6040	1.1226	1.1688	1.2607	1.1470	1.1253 **(1.1240)**	1.1162	1.1158
k=3	MSE	12.5550	1.6954	5.7283	3.1187	3.0885	2.5762 **(2.4446)**	1.9656	1.9312
	GOF	0.4713	0.9046	0.7351	0.8673	0.8882	0.8996 **(0.9014)**	0.9087	0.9092
	RMSE	2.7935	1.1932	1.9097	1.4121	1.2794	1.2376 **(1.2309)**	1.2008	1.1990

EK 3: n=50 ve p=10, GÖZLEMLERİN İLK %10'U DİKEY AYKIRI DEĞERLER, m=1000 TEKRAR İÇİN BENZETİM SONUÇLARINA İLİŞKİN ÇİZELGE

Modeldeki Bileşen Sayısı		PLSR	RSIMPLS	PRM	PLS-SD	PLS-KurSD	***PLS-ARWMCD***	***PLS-Smult***	***PLS-MMmult***
k=1	MSE	0.6133	0.4910	0.5419	0.5487	0.5543	0.5518 **(0.5533)**	0.5465 **(0.5463)**	0.5465
	GOF	0.7613	0.8126	0.8027	0.8019	0.7984	0.8016 **(0.8014)**	0.8034	0.8035
	RMSE	1.6175	1.4485	1.4860	1.4886	1.4956	1.4884 **(1.4883)**	1.4840 **(1.4839)**	1.4840
k=2	MSE	1.1477	0.0517	0.0428	0.0549	0.2505	0.0528 **(0.0504)**	0.0406 **(0.0443)**	0.0400 **(0.0411)**
	GOF	0.8132	0.8942	0.8932	0.8926	0.8868	0.8931 **(0.8936)**	0.8950 **(0.8949)**	0.8951 **(0.8950)**
	RMSE	1.4543	1.1198	1.1189	1.1270	1.1460	1.1208 **(1.1198)**	1.1135 **(1.1143)**	1.1131 **(1.1138)**
k=3	MSE	29.0442	1.6097	1.8694	2.6334	4.6291	2.4799 **(2.3195)**	1.8839 **(1.9818)**	1.8565 **(1.9517)**
	GOF	0.6241	0.9050	0.9043	0.9000	0.8723	0.8998 **(0.9019)**	0.9089 **(0.9077)**	0.9094 **(0.9087)**
	RMSE	2.2290	1.1933	1.2033	1.2438	1.3276	1.2358 **(1.2277)**	1.1998 **(1.2053)**	1.1983 **(1.2017)**

EK 4: n=1000, p=5, k=2, AYKIRI DEĞER ORANI % 10, m=1000 TEKRAR İÇİN BENZETİM SONUÇLARINA İLİŞKİN ÇİZELGE

	PLSR	RSIMPLS	PRM	PLS-SD	PLS-KurSD	***PLS-ARWMCD***	***PLS-Smult***	***PLS-MMmult***
Temiz Veri								
MSE	0.0042	0.0046	0.0047	0.0044	0.0044	0.0043	0.0045	0.0043
GOF	0.8298	0.8297	0.8297	0.8297	0.8297	0.8297	0.8297	0.8297
RMSE	1.0926	1.0927	1.0927	1.0927	1.0927	1.0926	1.0927	1.0926
Açı $\left(\beta;\hat{\beta}_{[y_c,x_c]k}\right)$	0.0217	0.0243	0.0222	0.0229	0.0226	0.0222	0.0234	0.0223
Kötü Kaldıraç Gözlemleri								
MSE	1.6967	0.0046	0.0610	0.0944	0.0045	0.0044	0.0046	0.0044
GOF	0.2586	0.8293	0.8165	0.8068	0.8293	0.8294	0.8293	0.8293
RMSE	2.2786	1.0912	1.1324	1.1621	1.0911	1.0910	1.0910	1.0910
Açı $\left(\beta;\hat{\beta}_{[y_c,x_c]k}\right)$	1.1402	0.0244	0.0659	0.0846	0.0242	0.0233	0.0243	0.0236
Dikey Aykırı Değerler								
MSE	0.0123	0.0043	0.0050	0.0046	0.0044	0.0042	0.0044	0.0043
GOF	0.8271	0.8296	0.8296	0.8296	0.8297	0.8297	0.8297	0.8297
RMSE	1.0991	1.0910	1.0910	1.0910	1.0908	1.0907	1.0907	1.0907
Açı $\left(\beta;\hat{\beta}_{[y_c,x_c]k}\right)$	0.0509	0.0222	0.0249	0.0250	0.0241	0.0231	0.0242	0.0235
İyi Kaldıraç Gözlemleri								
MSE	0.5603	0.0046	0.5239	0.0098	0.0045	0.0043	0.0044	0.0043
GOF	0.8088	0.8294	0.8119	0.8292	0.8294	0.8295	0.8294	0.8294
RMSE	1.1530	1.0905	1.1441	1.0909	1.0903	1.0902	1.0904	1.0902
Açı $\left(\beta;\hat{\beta}_{[y_c,x_c]k}\right)$	0.5264	0.0257	0.5065	0.0647	0.0241	0.0230	0.0243	0.0235
Kümelenmiş Aykırı Değerler								
MSE	1.8930	0.0047	1.8442	0.0247	0.0045	0.0044	0.0046	0.0044
GOF	0.4978	0.8293	0.7768	0.8263	0.8293	0.8294	0.8293	0.8293
RMSE	1.8739	1.0912	1.2449	1.1009	1.0911	1.0910	1.0910	1.0910
Açı $\left(\beta;\hat{\beta}_{[y_c,x_c]k}\right)$	1.1011	0.0257	0.6977	0.0549	0.0241	0.0233	0.0243	0.0236
Dik Aykırı Değerler								
MSE	0.1602	0.0049	0.1269	0.0044	0.0045	0.0043	0.0045	0.0044
GOF	0.7883	0.8293	0.7982	0.8293	0.8293	0.8294	0.8293	0.8294
RMSE	1.2136	1.0909	1.1836	1.0908	1.0909	1.0907	1.0909	1.0908
Açı $\left(\beta;\hat{\beta}_{[y_c,x_c]k}\right)$	0.2646	0.0261	0.2258	0.0237	0.0241	0.0231	0.0241	0.0234

EK 5: n=1000, p=5, k=2, AYKIRI DEĞER ORANI % 20, m=1000 TEKRAR İÇİN BENZETİM SONUÇLARINA İLİŞKİN ÇİZELGE

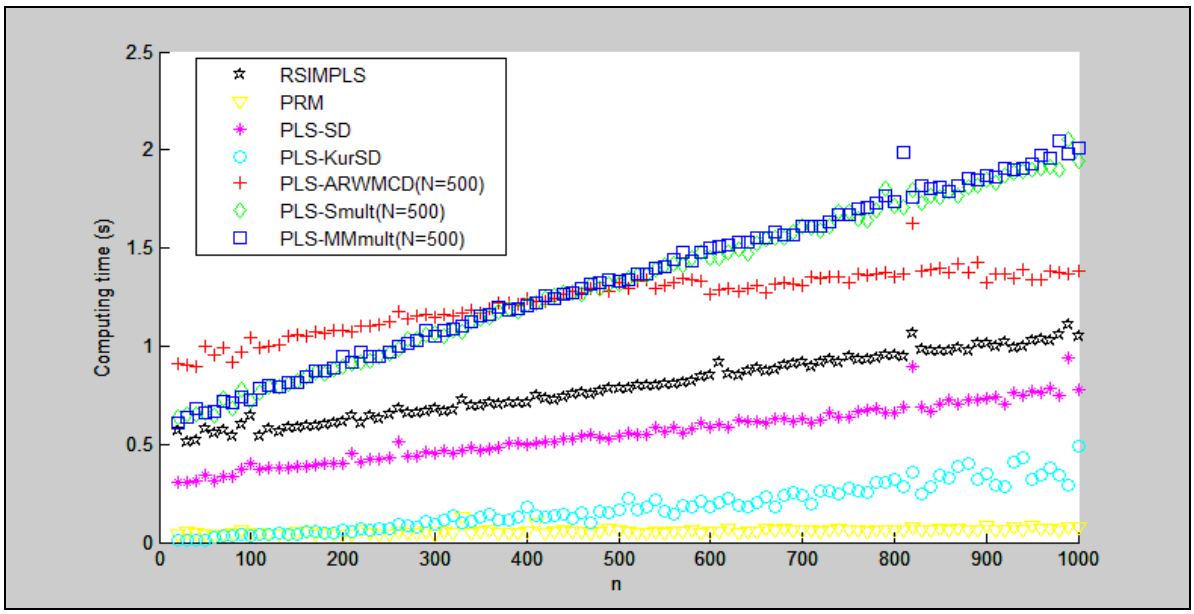
	PLSR	RSIMPLS	PRM	PLS-SD	PLS-KurSD	***PLS-ARWMCD***	***PLS-Smult***	***PLS-MMmult***
Temiz Veri								
MSE	0.0042	0.0046	0.0047	0.0044	0.0044	0.0043	0.0045	0.0043
GOF	0.8298	0.8297	0.8297	0.8297	0.8297	0.8297	0.8297	0.8297
RMSE	1.0926	1.0927	1.0927	1.0927	1.0926	1.0926	1.0927	1.0926
Açı $\left(\beta;\hat{\beta}_{[y_c,x_c]k}\right)$	0.0217	0.0243	0.0222	0.0229	0.0226	0.0222	0.0234	0.0223
Kötü Kaldıraç Gözlemleri								
MSE	1.8767	0.0046	1.7565	0.4018	0.0046	0.0044	0.0046 **(0.0177)**	0.0045 **(0.0082)**
GOF	0.1802	0.8299	0.2344	0.7087	0.8299	0.8300	0.8300 **(0.8255)**	0.8300 **(0.8287)**
RMSE	2.3973	1.0912	2.3172	1.4306	1.0911	1.0910	1.0911 **(0.1012)**	1.0910 **(1.0936)**
Açı $\left(\beta;\hat{\beta}_{[y_c,x_c]k}\right)$	1.3135	0.0242	1.1953	0.2285	0.0248	0.0238	0.0245 **(0.0330)**	0.0241 **(0.0266)**
Dikey Aykırı Değerler								
MSE	0.0178	0.0044	0.0057	0.0056	0.0046	0.0044	0.0077 **(0.0161)**	0.0082 **(0.0171)**
GOF	0.8252	0.8297	0.8295	0.8295	0.8299	0.8299	0.8288 **(0.8261)**	0.8286 **(0.8256)**
RMSE	1.1089	1.0948	1.0959	1.0957	1.0946	1.0945	1.0979 **(1.1068)**	1.0981 **(1.1082)**
Açı $\left(\beta;\hat{\beta}_{[y_c,x_c]k}\right)$	0.0634	0.0223	0.0290	0.0297	0.0250	0.0239	0.0338 **(0.0590)**	0.0345 **(0.0608)**
İyi Kaldıraç Gözlemleri								
MSE	0.5808	0.0048	0.6285	0.0642	0.0045	0.0044	0.0045 **(0.0075)**	0.0044 **(0.0095)**
GOF	0.8073	0.8295	0.8028	0.8279	0.8296	0.8296	0.8296 **(0.8295)**	0.8296 **(0.8294)**
RMSE	1.1604	1.0915	1.1739	1.0965	1.0913	1.0912	1.0912 **(1.0916)**	1.0912 **(1.0917)**
Açı $\left(\beta;\hat{\beta}_{[y_c,x_c]k}\right)$	0.5373	0.0267	0.5615	0.1769	0.0250	0.0241	0.0247 **(0.0273)**	0.0243 **(0.0289)**
Kümelenmiş Aykırı Değerler								
MSE	1.7880	0.0048	1.8592	0.1249	0.0046	0.0044	0.0046 **(0.0105)**	0.0045 **(0.0084)**
GOF	0.4805	0.8299	0.4724	0.8099	0.8299	0.8300	0.8300 **(0.8289)**	0.8300 **(0.8294)**
RMSE	1.9104	1.0912	1.9262	1.1548	1.0911	1.0910	1.0911 **(1.0932)**	1.0910 **(1.0925)**
Açı $\left(\beta;\hat{\beta}_{[y_c,x_c]k}\right)$	1.1062	0.0262	1.1100	0.2047	0.0248	0.0238	0.0245 **(0.0278)**	0.0241 **(0.0262)**
Dik Aykırı Değerler								
MSE	0.1894	0.0055	0.2179	0.0047	0.0048	0.0045	0.0047 **(0.0066)**	0.0046 **(0.0064)**
GOF	0.7806	0.8296	0.7733	0.8297	0.8297	0.8297	0.8297 **(0.8292)**	0.8297 **(0.8293)**
RMSE	1.2386	1.0921	1.2596	1.0920	1.0922	1.0919	1.0920 **(1.0939)**	1.0920 **(1.0933)**
Açı $\left(\beta;\hat{\beta}_{[y_c,x_c]k}\right)$	0.2949	0.0285	0.3173	0.0251	0.0259	0.0246	0.0255 **(0.0283)**	0.0250 **(0.0277)**

EK 6: p=5, k=2 VE KÜÇÜK, ORTA, BÜYÜK ÖRNEKLEMLER İÇİN FARKLI HATA DAĞILIMLARINDA m=1000 TEKRAR YAPILARAK BENZETİM İLE ELDE EDİLMİŞ MSE'LERE İLİŞKİN ÇİZELGE

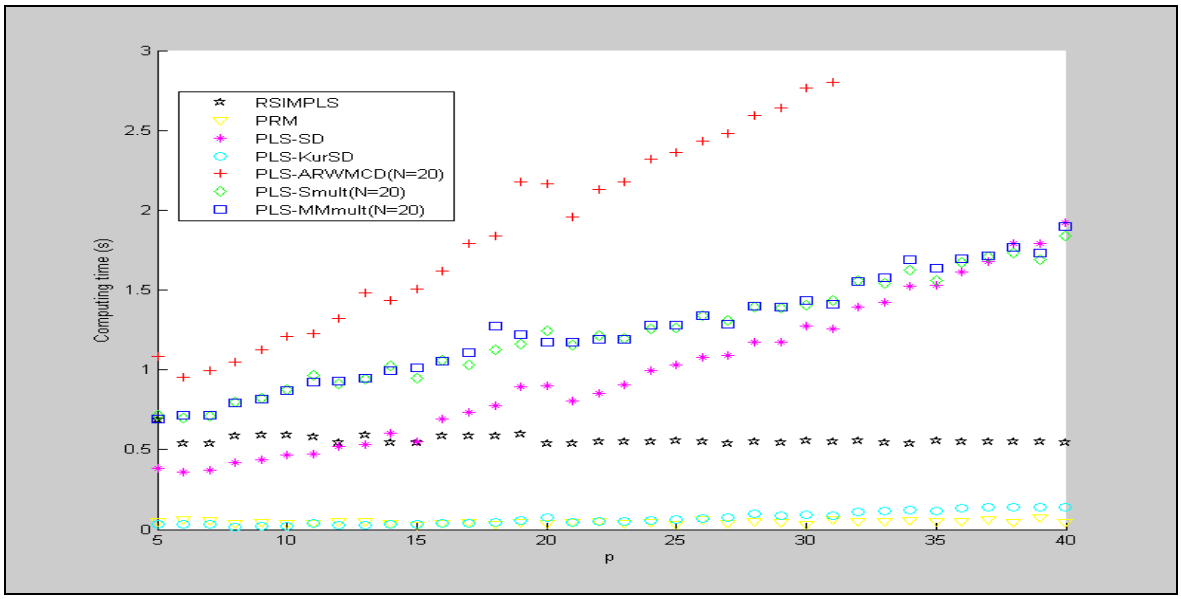
		N(0.1)	Laplace	t_5	t_2	Cauchy	Slash
n=20,p=5,k=2	PLSR	0.0468	0.0928	0.0756	1.4807	35365.4979	366482.9991
	RSIMPLS	0.1462	0.2434	0.2026	0.3726	0.8726	1.2921
	PRM	0.0552	0.0697	0.0702	0.0969	0.2595	0.5249
	PLS-SD	0.0850	0.1524	0.1109	0.1704	1.0914	1.4504
	PLS-KurSD	0.2389	0.3441	0.2807	0.5629	1.4319	4.2582
	PLS-ARWMCD	0.1228	0.1997	0.1789	0.2535	1.1486	1.4430
	PLS-Smult	0.0782	0.1185	0.1004	0.1613	0.6083	1.1004
	PLS-MMmult	0.0601	0.0985	0.0778	0.1335	0.8432	1.5248
n=200,p=5,k=2	PLSR	0.0055	0.0096	0.0085	0.1946	791.0193	2181.4032
	RSIMPLS	0.0069	0.0082	0.0082	0.0096	0.0141	0.0266
	PRM	0.0060	0.0067	0.0075	0.0093	0.0138	0.0282
	PLS-SD	0.0061	0.0077	0.0078	0.0102	0.0166	0.0331
	PLS-KurSD	0.0061	0.0092	0.0085	0.0121	0.0208	0.0393
	PLS-ARWMCD	0.0061	0.0089	0.0085	0.0109	0.0190	0.0356
	PLS-Smult	0.0061	0.0082	0.0080	0.0104	0.0179	0.0337
	PLS-MMmult	0.0056	0.0081	0.0076	0.0109	0.0229	0.0408
n=1000,p=5,k=2	PLSR	0.0010	0.0020	0.0016	0.0105	766.1796	67243.1547
	RSIMPLS	0.0015	0.0018	0.0017	0.0020	0.0027	0.0049
	PRM	0.0012	0.0014	0.0014	0.0019	0.0026	0.0053
	PLS-SD	0.0012	0.0016	0.0015	0.0020	0.0032	0.0061
	PLS-KurSD	0.0012	0.0020	0.0016	0.0026	0.0044	0.0079
	PLS-ARWMCD	0.0011	0.0018	0.0016	0.0022	0.0039	0.0070
	PLS-Smult	0.0013	0.0017	0.0016	0.0022	0.0034	0.0063
	PLS-MMmult	0.0011	0.0017	0.0015	0.0022	0.0042	0.0075

EK 7:

BAĞIMSIZ DEĞİŞKEN SAYISI p=5 VE GÖZLEM SAYISI 1'DEN 1000'E KADAR OLAN BENZETİM VERİSİ İÇİN RSIMPLS, PRM, PLS-SD, PLS-KurSD, *PLS-ARWMCD(N=500)*, *PLS-Smult(N=500)*, *PLS-MMmult(N=500)* YÖNTEMLERİNİN SANİYE TÜRÜNDEN HESAPLAMA ZAMANLARI



GÖZLEM SAYISI N=100 VE BAĞIMSIZ DEĞİŞKEN SAYISI 1'DEN 40'A KADAR OLAN BENZETİM VERİSİ İÇİN RSIMPLS, PRM, PLS-SD, PLS-KurSD, *PLS-ARWMCD(N=500)*, *PLS-Smult(N=500)*, *PLS-MMmult(N=500)* YÖNTEMLERİNİN SANİYE TÜRÜNDEN HESAPLAMA ZAMANLARI

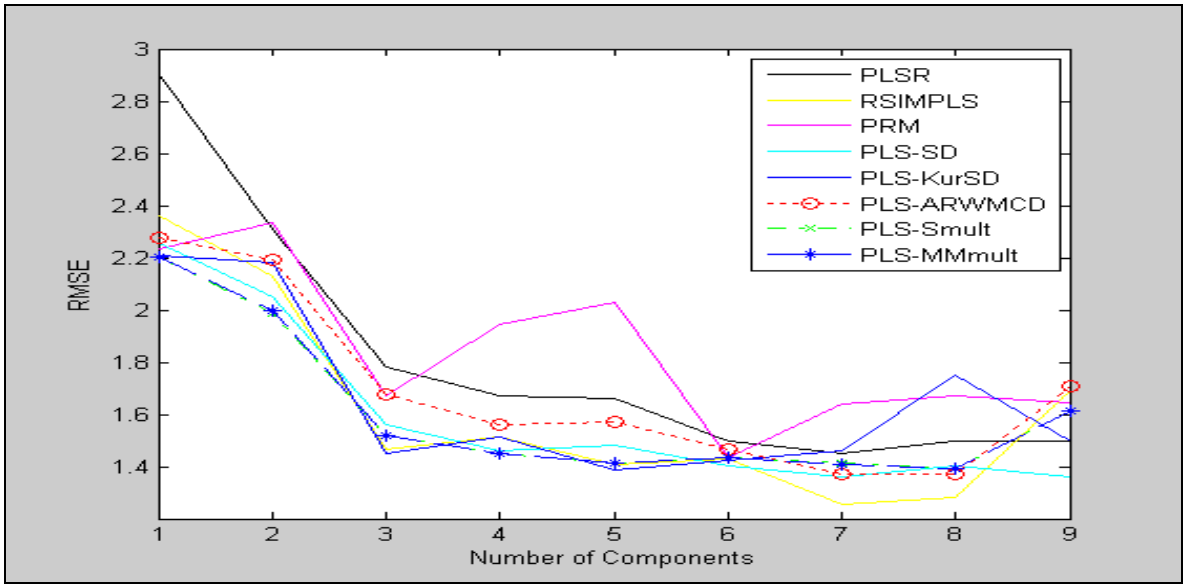


EK 8: BALIK VERİ KÜMESİ

Gözlem	Yağ (%)	x1	x2	x3	x4	x5	x6	x7	x8	x9
1	30.4	1.13980	1.05685	0.90927	0.87793	0.71833	0.58104	0.61953	0.64722	0.37794
2	41.8	1.44555	1.32580	1.13550	1.10678	0.92133	0.78349	0.83632	0.86616	0.50157
3	44.1	1.46065	1.34130	1.15195	1.12160	0.93330	0.79759	0.85026	0.88070	0.50999
4	42.7	1.56370	1.43638	1.23345	1.20193	0.99419	0.85343	0.91167	0.94403	0.53703
5	38.7	1.35975	1.25613	1.08188	1.04878	0.86141	0.72751	0.77614	0.80603	0.46438
6	39.9	1.44000	1.32990	1.14793	1.11545	0.92487	0.77911	0.83033	0.86142	0.50360
7	35.9	1.45343	1.34338	1.16035	1.12573	0.92252	0.76773	0.81970	0.85291	0.48782
8	40.8	1.56753	1.44003	1.24345	1.21078	1.01828	0.86513	0.92025	0.95178	0.56443
9	38.6	1.35713	1.24753	1.07235	1.04098	0.85951	0.71805	0.76632	0.79646	0.46192
10	41.6	1.43913	1.31608	1.12620	1.09798	0.92976	0.76965	0.82151	0.85076	0.50164
11	44.8	1.57905	1.44315	1.23393	1.20498	1.00268	0.86912	0.92803	0.95886	0.55048
12	44.8	1.78915	1.65938	1.45265	1.41970	1.19675	1.03895	1.10350	1.13790	0.66043
13	43.6	1.61793	1.49213	1.29105	1.26025	1.05700	0.91433	0.97399	1.00564	0.58153
14	43.1	1.56153	1.43430	1.23433	1.20270	1.00009	0.86579	0.92326	0.95428	0.55037
15	39.6	1.40280	1.28250	1.09443	1.06215	0.87491	0.73496	0.78646	0.81739	0.46870
16	45.2	1.54388	1.42820	1.24168	1.21508	1.02900	0.89669	0.95201	0.98111	0.58229
17	41.8	1.54553	1.42563	1.23093	1.20180	0.99860	0.85541	0.91213	0.94326	0.54617
18	43.3	1.61078	1.48515	1.28715	1.25690	1.05455	0.91132	0.96786	0.99921	0.58882
19	41.6	1.44988	1.34105	1.16500	1.13613	0.95728	0.81291	0.86600	0.89610	0.51940
20	31.6	1.28340	1.18940	1.02460	0.98946	0.80344	0.64830	0.69260	0.72401	0.41221
21	43.0	1.40150	1.28305	1.09620	1.06780	0.87955	0.75973	0.81099	0.83945	0.48086
22	35.9	1.36360	1.26308	1.09253	1.06185	0.88683	0.73139	0.77900	0.80890	0.47737
23	36.0	1.39218	1.28633	1.10810	1.07515	0.88763	0.73845	0.78682	0.81735	0.47821
24	42.3	1.44163	1.32118	1.12598	1.09385	0.89453	0.76510	0.81861	0.84923	0.48080
25	43.3	1.49383	1.37445	1.18933	1.16123	0.98820	0.83752	0.89124	0.92061	0.54660
26	45.4	1.49850	1.36700	1.16715	1.13993	0.95004	0.82147	0.87655	0.90596	0.52645
27	40.7	1.61163	1.48863	1.28673	1.25180	1.03677	0.88583	0.94401	0.97807	0.56024
28	40.4	1.47875	1.35653	1.16795	1.13790	0.95832	0.80477	0.85708	0.88707	0.52769
29	44.5	1.66145	1.52728	1.32440	1.29335	1.09613	0.95084	1.01070	1.04128	0.61161
30	46.3	1.56013	1.43485	1.24450	1.21865	1.04403	0.90562	0.96207	0.98956	0.59218
31	39.1	1.53533	1.41648	1.22418	1.19233	0.99704	0.84299	0.89836	0.93028	0.54187
32	37.6	1.38763	1.28040	1.10125	1.06798	0.87580	0.73564	0.78512	0.81573	0.47067
33	37.1	1.28400	1.18258	1.01315	0.98289	0.80005	0.67085	0.71607	0.74502	0.42770
34	39.4	1.40040	1.28623	1.10198	1.06965	0.88261	0.74443	0.79510	0.82521	0.47336
35	41.6	1.36028	1.25598	1.08925	1.06010	0.88576	0.75780	0.80684	0.83544	0.48995
36	36.4	1.38418	1.28073	1.10590	1.07248	0.88419	0.73839	0.78686	0.81754	0.47667
37	40.7	1.52028	1.39388	1.19088	1.15688	0.95552	0.80885	0.86418	0.89560	0.51390
38	36.3	1.35128	1.25043	1.07730	1.04448	0.85732	0.70904	0.75800	0.78910	0.44729
39	48.7	1.90523	1.76165	1.53818	1.50960	1.27988	1.13533	1.20703	1.24058	0.70615
40	48.2	1.94215	1.78618	1.55443	1.52023	1.28218	1.12788	1.20123	1.23478	0.70206
41	48.9	1.84348	1.69655	1.47178	1.44495	1.22290	1.09688	1.16533	1.19568	0.69323
42	43.3	1.52323	1.40215	1.21343	1.18808	1.02745	0.86285	0.91723	0.94593	0.57550
43	46.8	2.26510	2.07033	1.78608	1.75028	1.46640	1.29485	1.38043	1.41923	0.78525
44	40.7	2.25588	2.08190	1.80888	1.76018	1.46528	1.21893	1.30105	1.34415	0.74531
45	42.7	2.09810	1.95125	1.71603	1.67763	1.42225	1.21425	1.29205	1.33018	0.76022

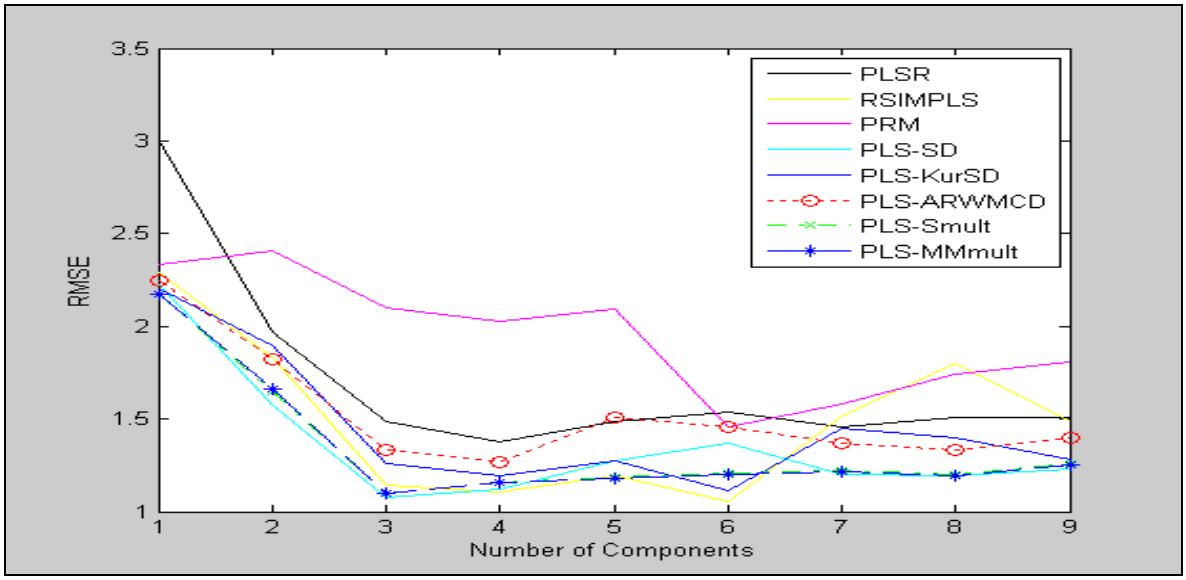
EK 9: ÇALIŞMA KÜMESİ 35 VE TEST KÜMESİ 10 GÖZLEMLİ OLAN BALIK VERİSİ İÇİN GOF VE RMSE DEĞERLERİNİN VERİLDİĞİ ÇİZELGE VE BİLEŞEN SAYISINA KARŞI RMSE DEĞERLERİNİN ŞEKLİ

Modeldeki Bileşen Sayısı		PLSR	RSIMPLS	PRM	PLS-SD	PLS-KurSD	***PLS-ARWMCD***	***PLS-Smult***	***PLS-MMmult***
k=1	GOF	0.4093	0.5639	0.4953	0.5696	0.5138	0.5706	0.5475	0.5486
	RMSE	2.9002	2.3593	2.2321	2.2572	2.2062	2.2778	2.2013	2.2018
k=2	GOF	0.7745	0.8397	0.4621	0.8665	0.8834	0.8311	0.8519	0.8497
	RMSE	2.3099	2.1272	2.3355	2.0504	2.1840	2.1943	1.9874	1.9953
k=3	GOF	0.9306	0.9741	0.7045	0.9720	0.9750	0.9622	0.9740	0.9740
	RMSE	1.7827	1.4642	1.6713	1.5598	1.4495	1.6787	1.5168	1.5185
k=4	GOF	0.9335	0.9761	0.7107	0.9697	0.9662	0.9598	0.9699	0.9699
	RMSE	1.6700	1.5108	1.9448	1.4594	1.5155	1.5612	1.4488	1.4509
k=5	GOF	0.9365	0.9763	0.7192	0.9703	0.9766	0.9577	0.9749	0.9749
	RMSE	1.6610	1.4070	2.0277	1.4795	1.3875	1.5692	1.4127	1.4132
k=6	GOF	0.9395	0.9702	0.8253	0.9691	0.9718	0.9611	0.9727	0.9730
	RMSE	1.4951	1.4280	1.4348	1.4029	1.4243	1.4659	1.4349	1.4330
k=7	GOF	0.9459	0.9812	0.8341	0.9722	0.9641	0.9665	0.9713	0.9717
	RMSE	1.4486	1.2558	1.6385	1.3613	1.4585	1.3715	1.4143	1.4105
k=8	GOF	0.9461	0.9819	0.8412	0.9710	0.9075	0.9664	0.9731	0.9734
	RMSE	1.4995	1.2808	1.6733	1.4019	1.7483	1.3732	1.3928	1.3900
k=9	GOF	0.9459	0.9428	0.8412	0.9709	0.9518	0.9418	0.9573	0.9577
	RMSE	1.4973	1.6930	1.6434	1.3588	1.4986	1.7064	1.6147	1.6118



EK 10: ÇALIŞMA KÜMESİ 25 VE TEST KÜMESİ 20 GÖZLEMLİ OLAN BALIK VERİSİ İÇİN GOF VE RMSE DEĞERLERİNİN VERİLDİĞİ ÇİZELGE VE BİLEŞEN SAYISINA KARŞI RMSE DEĞERLERİNİN ŞEKLİ

Modeldeki Bileşen Sayısı		PLSR	RSIMPLS	PRM	PLS-SD	PLS-KurSD	***PLS-ARWMCD***	***PLS-Smult***	***PLS-MMmult***
k=1	GOF	0.2912	0.4335	0.3777	0.4417	0.4444	0.4397	0.4438	0.4445
	RMSE	3.0001	2.2937	2.3307	2.2274	2.2029	2.2487	2.1759	2.1768
k=2	GOF	0.6927	0.7421	0.2713	0.7853	0.6948	0.7605	0.7988	0.7933
	RMSE	1.9715	1.8293	2.4072	1.5730	1.8935	1.8234	1.6459	1.6602
k=3	GOF	0.8820	0.9687	0.6166	0.9665	0.9594	0.9579	0.9681	0.9681
	RMSE	1.4861	1.1401	2.0993	1.0797	1.2590	1.3322	1.0974	1.0985
k=4	GOF	0.8987	0.9710	0.6277	0.9737	0.9447	0.9662	0.9717	0.9718
	RMSE	1.3742	1.1089	2.0237	1.1198	1.1924	1.2646	1.1605	1.1568
k=5	GOF	0.9113	0.9790	0.6782	0.9716	0.9777	0.9713	0.9797	0.9797
	RMSE	1.4874	1.1918	2.0921	1.2705	1.2708	1.5054	1.1872	1.1821
k=6	GOF	0.9231	0.9825	0.7705	0.9796	0.9854	0.9816	0.9880	0.9879
	RMSE	1.5348	1.0543	1.4578	1.3727	1.1129	1.4545	1.2097	1.2022
k=7	GOF	0.9299	0.9829	0.7806	0.9714	0.9865	0.9862	0.9892	0.9892
	RMSE	1.4553	1.5190	1.5835	1.2033	1.4528	1.3679	1.2230	1.2168
k=8	GOF	0.9463	0.9768	0.8063	0.9769	0.9868	0.9861	0.9893	0.9893
	RMSE	1.5056	1.7989	1.7409	1.1925	1.4019	1.3300	1.2032	1.1972
k=9	GOF	0.9463	0.9851	0.8087	0.9812	0.9798	0.9870	0.9897	0.9897
	RMSE	1.5052	1.4874	1.8095	1.2338	1.2843	1.3990	1.2572	1.2550



ÖZGEÇMİŞ

Kimlik Bilgileri

Adı Soyadı : Esra Polat
Doğum Yeri : Ayancık
Medeni Hali : Bekar
E-posta : espolat@hacettepe.edu.tr
Adresi : Hacettepe Üniversitesi, Fen Fakültesi, İstatistik Bölümü, 06800 Beytepe, ANKARA.

Eğitim

Lise: 1999-2002 Yıldırım Beyazıt Anadolu Lisesi
Lisans: 2002-2006 Hacettepe Üniversitesi, Fen Fakültesi, İstatistik Bölümü
Yüksek Lisans : Hacettepe Üniversitesi, F.B.E., İstatistik A.D.
Doktora : Hacettepe Üniversitesi, F.B.E., İstatistik A.D.

Yabancı Dil ve Düzeyi

İngilizce
Kamu Personeli Yabancı Dil Bilgisi Seviye Tespit Sınavı (KPDS)- 92 Puan

İş Deneyimi

2007- … : Hacettepe Üniversitesi, Fen Fakültesi, İstatistik Bölümü- Araştırma Görevlisi

Deneyim Alanları

-

Tezden Üretilmiş Projeler ve Bütçesi

-

Tezden Üretilmiş Yayınlar

-

Tezden Üretilmiş Tebliğ ve/veya Poster Sunumu ile Katıldığı Toplantılar

-