**T.C.**
**ESKİŞEHİR OSMANGAZİ ÜNİVERSİTESİ**
**SAĞLIK BİLİMLERİ ENSTİTÜSÜ**
**BİYOİSTATİSTİK ANABİLİM DALI**

# RANDOM FORESTS YÖNTEMİNDE KAYIP VERİ PROBLEMİNİN İNCELENMESİ VE SAĞLIK ALANINDA BİR UYGULAMA

**YÜKSEK LİSANS TEZİ**

**HÜLYA YILMAZ**

**DANIŞMAN**
**YRD. DOÇ. DR. CENGİZ BAL**

**OCAK-2014**

**T.C.**
**ESKİŞEHİR OSMANGAZİ ÜNİVERSİTESİ**
**SAĞLIK BİLİMLERİ ENSTİTÜSÜ**
**BİYOİSTATİSTİK ANABİLİM DALI**

# RANDOM FORESTS YöNTEMiNDE KAYIP VERi PROBLEMiNiN iNCELENMESi VE SAĞLIK ALANINDA BiR UYGULAMA

**YÜKSEK LİSANS TEZİ**

**HÜLYA YILMAZ**

**DANIŞMAN**
**YRD. DOÇ. DR. CENGİZ BAL**

## KABUL VE ONAY SAYFASI

HÜLYA YILMAZ'ın Yüksek Lisans Tezi olarak hazırladığı "RANDOM FORESTS YÖNTEMİNDE KAYIP VERİ PROBLEMİNİN İNCELENMESİ VE SAĞLIK ALANINDA BİR UYGULAMA" başlıklı bu çalışma Eskişehir Osmangazi Üniversitesi Lisansüstü Eğitim ve Öğretim Yönetmeliği'nin ilgili maddesi uyarınca değerlendirilerek "KABUL" edilmiştir.

17.01.2014

Üye: Prof. Dr. Kazım ÖZDAMAR

Üye: Doç. Dr. Setenay ÖNER

Üye: Doç. Dr. Fezan MUTLU

Üye: Doç. Dr. Canan BAYDEMİR

Üye: Yrd. Doç. Dr. Cengiz BAL

Eskişehir Osmangazi Üniversitesi Sağlık Bilimleri Enstitüsü Yönetim Kurulu'nun 24./01./2014. tarih ve 987/.4582 sayılı kararı ile onaylanmıştır.

Prof.Dr. KAZIM ÖZDAMAR

Enstitü Müdürü

# ÖZET

## RANDOM FORESTS YÖNTEMİNDE KAYIP VERİ PROBLEMİNİN İNCELENMESİ VE SAĞLIK ALANINDA BİR UYGULAMA

Bu tez çalışmasında, kayıp verili sınıflandırma probleminde kullanılan Random Forests (RF) yönteminin kayıp değer atama algoritmasıyla, K En Yakın Komşu (KNN) ile kayıp değer atama yönteminin karşılaştırılması amaçlanmaktadır.

Karşılaştırmalar iki aşamada gerçekleştirilmiştir. İlk aşamada benzetim çalışmaları yapılmıştır. (100000/n) Monte Carlo benzetim tekniği örneklem hacimlerine (n=100, 200, 500, 1000) ve tekrar sayılarına (s=1000, 500, 200, 100) karar vermek için kullanılmıştır. Çok değişkenli standart normal dağılımdan, önemli değişkenlerinin birbirleri ile düşük, orta ve yüksek (r=0.1, 0.5, 0.9) derecede ilişkili olduğu veri setleri türetilmiştir. Bu veri setlerinin iki değişkeni üzerinde aynı anda ve aynı yüzdeliklerde ( %5, %10, %15, %20, %25) kayıp değerler oluşturulmuştur. Kayıp değerler RF'nin atama algoritması ve farklı komşuluk değerli (k=5, 10, 15, 20) KNN ile kayıp değer atama yöntemleri tarafından ayrı ayrı tamamlandıktan sonra farklı veri setleri elde edilmiştir. Atanmış farklı veri setleri aynı RF algoritmasına ayrı ayrı yerleştirilerek sınıflandırma sonuçları gözlemlenmiştir. Doğru sınıflandırma oranları (DSO) kullanılarak atama yöntemleri karşılaştırılmıştır. İkinci aşamada ise sağlık alanına ait kayıp değerli bir veri seti, atama yöntemlerini uygulamak ve elde edilen sonuçları benzetim çalışmalarıyla ilişkilendirmek için kullanılmıştır.

Benzetim çalışmalarında atama yöntemleri benzer DSO sonuçları sunmaktadır. Örneklem hacimleri ve değişkenler arasındaki ilişki arttıkça DSO artmakta, ama kayıp değer yüzdesi arttıkça DSO azalmaktadır. Orta ve düşük derecede ilişkili veri setlerinde KNN ile kayıp değer atama yöntemi, yüksek derecede ilişkili veri setlerinde ise RF'nin kayıp değer atama algoritması üstün sonuçlar vermiştir. En

yüksek DSO tahmin değeri örneklem hacminin 1000'e eşit olduğu %5 kayıp değerli yüksek derecede ilişkili (r=0.9) veri setlerinde RF'nin atama algoritması tarafından %95.66 olarak bulunmuştur. En düşük DSO tahmin değeri ise örneklem hacminin 100'e eşit olduğu %25 kayıp değerli düşük derecede ilişkili (r=0.1) veri setlerinde RF'nin atama algoritması tarafından %78.27 olarak bulunmuştur. Sağlık alanına yönelik yapılan uygulama, benzetim çalışması ile uyumlu sonuçlar vermiştir.

Bu çalışma; bir sınıflandırma probleminde, kayıp değerli veri setlerine atama yapmak için her iki yöntemin de kullanılabileceğini göstermektedir; ancak veri setinin ilişki yapısına göre en uygun atama yönteminin seçilmesi önerilmektedir. Düşük ve orta derecede ilişkili veri setlerinde komşuluk değerinin k=10, 15 ya da 20'e eşit olduğu KNN ile kayıp değer atama yöntemi kullanılmalıdır. Yüksek derecede ilişkili veri setlerinde ise RF'nin atama algoritması tercih edilmelidir.

**Anahtar Kelimeler**: Random Forests, Kayıp veri analizi, K en yakın komşu ile kayıp değer atama yöntemi

## SUMMARY

### STUDYING THE MISSING DATA PROBLEM IN RANDOM FORESTS METHOD AND AN APPLICATION IN HEALTH FIELD

In this thesis study, it's aimed to compare the missing data imputation algorithm of Random Forests (RF) and the K Nearest Neighbourhood (KNN) imputation method in a classification problem with missing data.

Comparisons were made in two steps. At the first step simulation studies were done. (100000/n) Monte Carlo Simulation Technique was used to determine sample sizes (n=100, 200, 500, 1000) and the number of repetitions (s=1000, 500, 200, 100). Data sets, whose important variables are low, middle, and high (r=0.1, 0.5, 0.9) correlated with each other, were generated from multivariate standard normal distribution. Missing values were created on two important variables with using same percentage (5%, 10%, 15%, 20%, 25%) simultaneously. Different datasets were obtained after having imputed the missing values seperately by RF's imputation algorithm and KNN imputation method with different neighbourhood values (k=5, 10, 15, 20). Classification results were observed by putting the different imputed datasets in the same RF model one by one. Imputation methods were compared by their true classification rates (TCR). At the second step, a dataset with missing values in health field was used to apply the imputation methods and associate the obtained results with simulation studies.

In simulation studies, imputation methods present simular TCR results. As the sample sizes and the correlation between variables increase, TCR increases, but as the percentage of missing value increases, TCR decreases. In low and middle correlated datasets KNN imputation, in high correlated datasets RF's imputation

algorithm gave better results. The highest TCR value was found 95.66% by RF's imputation algorithm in high corralated (r=0.9) datasets with 5% missing value when the sample size is equal to 1000. The lowest TCR was found found 78.27% by RF's imputation algorithm in low (r=0.1) corralated datasets with 25% missing value when the sample size is equal to 100. The application in health field gave matching results with simulation studies.

This study shows both methods can be used to impute a dataset with missing values in a classification problem, but it is suggested to choose the most suitable imputation method according to the the correlation structure of the dataset. In low and middle correlated datasets, KNN imputation method with the neighbourhood value is equal to 10, 15 or 20 should be used. In high correlated data sets RF's imputation algorithm should be prefered.

**Keywords:** Random Forests, Missing data analysis, KNN imputation method

# İÇİNDEKİLER

# TABLO DİZİNİ

**Sayfa**

# ŞEKİL DİZİNİ

**Sayfa**

## SİMGE VE KISALTMALAR DİZİNİ

| | |
|---|---|
| **CART** | Sınıflandırma ve Regresyon Karar Ağacı |
| **DSO** | Doğru Sınıflandırma Oranı |
| **TCR** | True Classification Rate |
| **KNN** | K En Yakın Komşu |
| **L** | Eğitim Veri Seti |
| **MAR** | Rasgele Olarak Kayıp |
| **MI** | Çoklu Atama Yöntemi |
| **MCAR** | Tamamen Rasgele Olarak Kayıp |
| **MNAR** | Rasgele Olmayan Kayıp |
| **OOB** | Out of Bag Veri Seti |
| **RF** | Random Forests |
| **CIF** | Conditional Inference Forests |

# 1.GİRİŞ

Günümüzde teknolojinin ilerlemesi buna paralel olarak bilgisayarların hız ve kapasitelerinin artması sonucunda oldukça büyük veri yığınları elde edilip depolanmaktadır. Depolanan bu verilerin doğru biçimde değerlendirilmesi, ilişkilendirilmesi ve analizinin yapılması gerekmektedir. Son zamanlarda hızla gelişen Veri Madenciliği (Data Mining) bu konu üzerine çalışmaktadır. Veri Madenciliği, elde edilen veriyi değişik yönleriyle incelemekte ve farklı modeller kurarak veriyi bilgiye dönüştürmektedir. İstatistik, makine öğrenmesi (machine learning) gibi pek çok dal ile ilişki içerisinde bulunan Veri Madenciliği, karar ağaçları (decision trees), birliktelik kuralları oluşturma (association rules), sınıflama (classification), kümeleme (clustering) gibi pek çok algoritmayı kullanarak sonuçlar elde etmektedir.

Veri Madenciliğinde tahmin modelleri oluşturma ve sınıflandırma konusunda yaygın olarak kullanılan yöntemlerden biri karar ağaçlarıdır. Kurulan farklı algoritmalar sonucunda farklı karar ağaçları oluşturulmuştur. Fakat, elde edilen tekil karar ağaç (single decision tree) algoritmalarının genellikle aşırı uyum (overfitting) sergilemeleri başka arayışlara neden olmuştur. Günümüzde bilgisayar programlarının da gelişimi ile tekil karar ağaçları yerine, topluluk (ensemble) halinde kurulan karar ağaçları algoritmaları elde edilmiştir. Bunlar içerisinde en yaygın kullanılanları Bagging, Boosting ve Random Forests yöntemleridir.

Random Forests (RF), bootstrap yöntemi ile elde edilen alt örneklemler ile belirli bir tahmin kuralına göre sınıflandırma yapmaktadır. Genetik, biyoinformatik gibi pek çok alanda tercih edilen bir sınıflandırma ve regresyon algoritmasıdır. RF, çok boyutlu ve karmaşık veri setlerinde oldukça iyi çözümler vermektedir. Bu yöntem; değişken önemliliği, genelleştirilmiş hata ve gözlemler arası ilişkiyi gösteren uzaklık (proximity) matrisini hesaplamaktadır. Bu uzaklık matrisini kullanarak geliştirdiği algoritma ile veri setinde yer alan kayıp değerli gözlemlere

atama yapmaktadır. Böylece veri setinde yer alan tüm gözlemleri kullanarak sınıflandırma ya da regresyon karar ağaçları oluşturmaktadır.

Kayıp değerli veri setleri pek çok istatistiksel analizde ve veri madenciliği algoritmalarının kurulmasında büyük problemler oluşturmaktadır. Kayıp değerli gözlemlerin veri setinden çıkarılması, örneklem hacminin küçülmesine ve yapılan analizlerin istatistiksel gücünün azalmasına neden olmaktadır. Bu nedenle kayıp değerli gözlemlerin veri setinden çıkarılması yerine alternatif çözümler aranmıştır. Kayıp değerler için farklı yöntemler geliştirilerek yeni değer ataması yapılmıştır. Bunlara örnek olarak; tekil atama, çoklu atama, benzerlik ve tahmin fonksiyonu yöntemleri, Bayesgil ve çoklu atama, k en yakın komşu algoritması ile atama, vb. algoritmalar gösterilebilir.

K en yakın komşu algoritması ile kayıp değer atama yöntemi, gözlemlerin birbirlerine olan yakınlıkları üzerine kuruludur. Özellikle mikrodizilim (microarray) veri setleri gibi çok boyutlu setlerde kullanımı ön plana çıkmaktadır (25,26). Kayıp değerler, ait oldukları gözlemlerin k en yakın komşuları ile ilişkilendirilerek tahmin edilmektedir. Bu ilişkilendirme Öklid gibi bir uzaklık fonksiyonundan faydalanılarak yapılmaktadır. Uzaklık değerleri ile ağırlıklı ortalama hesaplanarak atanacak değer belirlenmektedir.

Bu tez çalışmasında, RF yönteminin kayıp değerli veri setinde uygulanması durumunda kullandığı algoritmanın, k en yakın komşu ile kayıp değer atama yöntemiyle karşılaştırılması amaçlanmaktadır. Bu karşılaştırmalar iki aşamada gerçekleştirilmiştir. İlk aşamada benzetim çalışmalarından faydalanılmıştır. Benzetim yolu ile yapılan karşılaştırmalar;

- Farklı örneklem hacimleri ve tekrar sayıları,
- Veri setlerindeki düşük, orta ve yüksek dereceli korelasyon yapıları
- Değişkenlerdeki farklı kayıp değer yüzdeleri,
- Farklı k en yakın komşuluk sayıları,

göz önünde bulundurularak yapılmıştır. Elde edilen sonuçlar, sınıfların doğru tahmin edilme oranlarının ortalamasına göre değerlendirilmiştir. İkinci aşamada ise sağlık alanına ait kayıp değerli bir veri seti üzerinde uygulama yapılmıştır. Atama yöntemleri uygulandıktan sonra elde edilen sonuçlar, benzetim çalışması sonuçları ile ilişkilendirilmiştir.

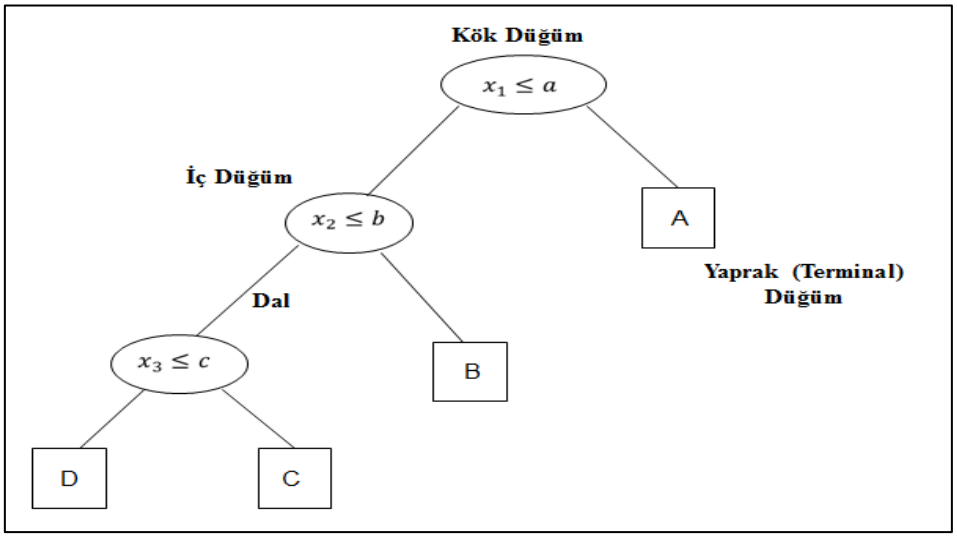
# 2. GENEL BİLGİLER

Bu tez çalışmasında kullanılan yöntemler aşağıda açıklanmaktadır. Öncelikle Karar ağaçları, RF'nin alt yapısını oluşturan Sınıflandırma ve Regresyon Karar Ağaçları, Topluluk Öğrenme Yöntemleri, RF'nin algoritması ve özelliklerinden bahsedilmektedir. Daha sonra ise Kayıp Veri Analizi hakkında bilgi verilip, K En Yakın Komşu ile Kayıp Değer Atama yöntemi anlatılmaktadır.

## 2.1. Karar Ağaçları

Sınıflama, makine öğrenmesinin (machine learning) önemli araçlarından biridir. Tümevarımsal öğrenmenin temel çerçevesi eğitim örneklerinden oluşan bir eğitim kümesi ile test örneklerinden oluşan bir test kümesini içerir. Sınıflama iki adımda gerçekleşir. Bunlar eğitim kümesi ile modelin kurulması ve test kümesi ile de bu modelin test edilmesidir. Modellerin kesinliğinin belirlenmesi için test örneklerinin bilinen sınıfları ile model tarafından tahmin edilen sınıflar karşılaştırılır. Test örneklerinin model tarafından doğru olarak sınıflanma oranı, kesinlik oranını verir (12).

Karar ağaçları bir örüntü sınıflandırma (pattern classification) türüdür. Bir bağımlı ve birden çok bağımsız değişkenden oluşan veri setinde, bağımlı değişken kategorik yapıda ise kurulan ağaç *sınıflandırma ağacı*, sürekli yapıda ise de *regresyon ağacı* olarak adlandırılır. Sınıflandırma ağaçlarında sınıf daha önceden atanmış bir değerdir. Ağaç, dallardan ve düğümlerden oluşur. İç düğüm (parent ya da internal), alt iki dala ayrılırken yaprak (terminal) düğüm en son noktadır ve hiç alt düğümü yoktur. Yaprak düğüm bir sınıf değeri taşır ve bu düğüme gelen gözlem artık o sınıfın bir elemanıdır.



Şekil 2.1. Karar ağacı şeması

Karar ağaçları kurulma biçimlerine göre farklılıklar gösterirler. Çünkü kullanılan algoritmaya göre oluşturulan ağacın şekli değişebilir. Değişik ağaç yapıları da farklı sınıflandırma sonuçları verir.

Karar ağaçları genellikle üç aşamada oluşturulur:

1) Eğitim ve test veri setleri belirlendikten sonra, eğitim veri seti ile oldukça büyük bir ağaç oluşturulur. Ağaç bölünmeye, tüm gözlemleri barındıran kök (root) düğümden başlar. Tahminci olan bağımsız değişken sürekli ise, dallara ayrılacak bölünme noktası tek bir değer olarak belirlenir. Tahminci değişkene ait değerler bölünme noktası değerinden küçük ise karar sol düğüme, tersi durumda ise sağ düğüme geçer. Eğer tahminci olan bağımsız değişken kategorik yapıda ise bölünme kuralı kategorilerin bir alt kümesini sol düğüme kalanının ise sağ düğüme gönderir. Kök düğümün bölünmesi için kullanılacak bu kural ağaç oluşumunda kullanılacak tüm tahminci değişkenler üzerindeki mümkün bölünme noktalarının belirlenmesinden sonra, bu noktalar içerisinden en iyi ayrımı sağlayanın seçilmesi ile elde edilir. Kök düğüm alt dallara ayrıldıktan sonra elde edilen düğümler aynı kurallar ile tekrarlı olarak bölünmeye devam eder. Bu bölünme belirli bir durdurma kuralına kadar devam eder. Bu kurallar genellikle, maksimum ağaç derinliği, bir düğümde bölünme için ele alınan minumum eleman sayısı ve yeni bir düğümde olması gereken

minumum eleman sayısı gibi çeşitli faktörlere dayanır. Burada ağaç derinliği, kök düğümden en uzaktaki yaprak düğüme kadar olan düğüm sayısını ifade eder (12).

Ağaç oluşumunda kullanılan bölünme kriterlerine örnek olarak;

- Karışıklık (impurity) ölçütü,
- Bilgi kazancı,
- Gini indeksi,
- Benzerlik-Oran Ki kare istatistikleri,
- Twoing ölçütü,
- Kolmogorov-Smirnov ölçütü gösterilebilir.

2) Oluşturulan karar ağacına budama (pruning) işlemi uygulanır. Budama işlemi, aşırı uyum (overfitting) problemini engellemek ve yanlış sınıflandırma hatasını minimize etmek için ağacın gerçek büyüklüğüne karar vermede kullanılır. Belirli düğümlerin çıkarılması ile elde edilen alt ağaç gruplarından uygun olanı seçilir. Budama türlerine örnek olarak;

- Maliyet-karmaşıklık budama (Cost-complexity pruning)
- İndirgenmiş hata budaması (Reduced error pruning)
- En düşük hata budaması (Minimum error pruning),
- Kötümser budama (Pessimistic pruning) gösterilebilir (17).

3) Bazı test yöntemleri (çapraz geçerlilik (cross validation), bağımsız test örneklemi (independent test sample), vb.) kullanılarak, budama sonucu elde edilen alt ağaç kümesinden en iyi olan ağaç seçilir.

Ağaç tabanlı yöntemlerin temelini oluşturan karar ağaçları modellerinin ilk uygulamaları AID (Automatic Interaction Detector) algoritması ile yapılmıştır ve çeşitli algoritmalar ile sürdürülmüştür. Geliştirilen bu algoritmalar içerisinde CHAID (Chi-Squared Automatic Interaction Detector;

G.V. Kass; 1980), CART (Classification and Regression Trees; Breiman, Friedman, Olshen ve Stone; 1984), ID3 (Quinlan; 1986), C4.5 (Quinlan; 1986), C5.0 (Quinlan), SLIQ (Supervised Learning in Quest; Mehta, Agarwal ve Rissanen), SPRINT (Scalable Parallelizable Induction of Decision Trees; Shafer, Agrawal ve Mehta) bunların başlıcalarıdır (2).

CHAID; ki-kare testlerini kullanarak bölme işlemini gerçekleştirir. Dalların sayısı iki ile tahmin edicinin kategori sayısı arasında değişir. CART; Gini'ye dayalı ikili bölme işlemini içerir. Son veya uç olmayan her bir düğümde iki adet dal bulunmaktadır. Budama işlemi ağacın karmaşıklık ölçüsüne dayanır. Sınıflandırma ve regresyon destekleyici bir yapıdadır. C4.5 ve C5.0; ID3 karar ağacı algoritmasının ileri versiyonlarıdır. Her düğümden çıkan çoklu dallar ile ağaç oluşturulur. Dalların sayısı tahmin edicinin kategori sayısına eşittir. Tek bir sınıflayıcıda birden çok karar ağacını birleştirir. Bölünme işlemi için bilgi kazancı yöntemi kullanılır. Budama işlemi her yapraktaki hata oranına dayanır. SLIQ; hızlı ölçeklenebilir bir sınıflayıcıdır. Hızlı ağaç budama algoritması mevcuttur. SPRINT; büyük veri kümeleri için idealdir. Bölme işlemi tek bir değişkenin değerine dayanır (12).

Karar ağaçlarının kullanım alanlarına örnek olarak; tıp alanında hastalıkların sınıflandırılması, NASA tarafından hava durumu tahmininin belirlemesi, finans sektöründe ise kredi skorlaması, portföy yöneticiliği ve sigortacılık alanlarındaki kullanımları gösterilebilir (4).

Karar ağaçları kullanımı oldukça yaygın bir yöntemdir. Bu yöntemin tercih edilmesindeki önemli etkenler şu şekilde sıralanabilir:

- Değişkenler arasındaki etkileşimleri belirleyebilir.
- Sürekli ve kategorik tahminci değişkenlerin her ikisini de yapısında bulundurabilir.
- Tahminci değişkenlerdeki kayıp veri problemine karşı duyarlıdır.

- Tahminci değişkenlerdeki sapan değerlerden etkilenmezler.
- Tahminci değişkenlerin monoton değişimlerinden etkilenmez.
- Büyük örneklem hacimleri için iyi ölçekleme yapmaktadır.

Bu özelliklerinin yanında karar ağaçları,

- Tahminci değişkenlerin doğrusal birleşmeleri tespit etmede başarılı değildir.
- Eğer veri setinin yapısı belirli bir oranda bozulursa, bu durum ağacın yapısını oldukça etkileyebilir. Bu nedenle karar ağaçları sabit olmayan bir yapıya sahip olarak bilinirler.

Karar ağaçlarının en büyük dezavantajı, son zamanlarda geliştirilen sınıflandırma ve regresyon yöntemleri kadar kesin sonuçlar verememesidir. Özellikle ağaç tabanlı topluluk yöntemler (tree-based ensembles medhods) olarak bilinen algoritmalar, uygun bir şekilde seçilmiş ağaçları birleştirerek daha kesin sonuçlar elde ederler. Bu topluluk yöntemler; Bagging, Boosting ve RF'dir (11).

## 2.2. Sınıflandırma ve Regresyon Karar Ağaçları

Sınıflandırma ve Regresyon karar ağaçları (Classification and Regression Trees; CART) 1984 yılında Leo Breiman, Jerome Friedman, Richard Olshen ve Charles Stone tarafından geliştirilen bir yöntemdir. Bu yöntem veri madenciliğinde oldukça önemli bir yere sahiptir (7).

CART parametrik olmayan bir yöntemdir. Kümeleme ve diskriminant analizlerinden oldukça farklı bir sınıflandırma algoritmasına sahiptir. Bu yöntemde sınıf sayısı önceden belirlidir ve fonksiyonel bir form tanımlamayı gerektirmemektedir. Karışık yapılı veri setleri için oldukça elverişlidir.

Kategorik ve sürekli değişkenlerin herhangi bir birleşiminden oluşan veri setlerini kullanabilir. Sapan değerlere karşı oldukça duyarlıdır (4).

CART üç aşamalı bir süreç ile oluşturulur. İlk aşama, optimal bölünmeyi sağlayacak kriterler kullanılarak aşırı büyük bir ağaç oluşturulur. Oluşturulan bu büyük ağaçların eğitim setine çok uyduğu görülmektedir. Bu nedenle herhangi bir genelleme yapamazlar. Yeni bir örüntüyü doğru sınıflandırma oranı oldukça düşük olacaktır. Bu problem için önerilen çözüm yolu, ikinci aşama olarak belirtilen, maliyet karmaşıklığı (cost complexity) yöntemi ile budama yaparak alt ağaç türleri oluşturmaktır. En son aşama ise oluşturulan bu alt ağaç türlerinden doğru büyüklükte olanın seçilmesidir. Bunun için çapraz geçerlilik (cross validation) ya da bağımsız test örneği (independent test sample) yöntemleri kullanılır (20).

Kayıp veri bulunduran veri setlerinde CART kullanılabilir. CART, kendi yapısında yer alan bir algoritma ile kayıp veri problemine bir çözüm getirmiştir. Bu algoritma yedek (surrogate) değişkenler üzerine kuruludur. Yedek değişkenler belirli bir birliktelik (association) puanına göre hesaplanırlar. Bölünmenin gerçekleştiği düğümde; ayırımı sağlayan tahminci değişkenin yedek değişkenine göre sol ya da sağ alt düğüme yerleştirme yapılır. Eğer aynı gözlemin, belirlenen ilk yedek değişken üzerinde de kayıp verisi var ise ayrım için ikinci yedek değişken kullanılır. Eğer tüm yedek değişkenler kayıp veri içeriyorsa, bu gözlem değeri sol ve sağ alt düğümden en kalabalık olanına yerleştirilir (17).

### 2.2.1. Ağacın oluşturulması

İkili (binary) sınıflandırma ağaçlarındaki temel düşünce, d boyutlu uzayı oldukça küçük parçalara ayırarak sınıf üyelikleri daha saf olacak şekilde bölümler oluşturmaktır. Bir başka ifadeyle, gözlemlerin büyük bir

çoğunluğunun bir sınıfa ait olabileceği parçalar araştırılmaktadır. Breiman ve arkadaşlarına göre ikili bölünmeler, çoklu bölünmelere göre daha çok tercih edilir. Çünkü;

- İkili bölmeler verileri çoklu bölmelere göre daha yavaş parçalara ayırır.
- Aynı değişken üzerinde tekrarlı bölünmelere izin verilir. Bir başka ifadeyle eğer gerekli ise bir değişken çok sayıda bölünme gerçekleştirebilir.

Ağaç oluşturulurken düğümlerin nasıl ayrılacağına karar vermek için bazı kriterlere ihtiyaç duyulur. Ayrıca, düğümlere ayırmayı ya da bir başka ifadeyle ağaç oluşturmayı sonlandıracağımız bir durdurma kuralı da belirlenmesi gerekebilir. Oldukça büyük bir ağaç oluşturduğumuz için durdurma kuralı basit olabilir. Durdurma kuralında bir seçenek, yaprak düğümler sadece bir sınıfa ait gözlem değerleri içerene kadar bölünmeye devam edilmesi iken bir başka seçenek de düğümde kalması beklenen gözlemlerin maksimum sayısını belirlemektir. Yaprak düğümde kalan maksimum gözlem sayısının 1 ile 5 arasında olması önerilmektedir.

Breiman ve arkadaşları; ağaçların sınıflandırılması için dört bölme kuralını (Gini, Twoing, Ordered Twoing, Symetric Gini) kullanarak örnekler ele almaktadırlar. Fakat son zamanlarda CART ile çalışan pek çok kişi Gini üzerine odaklanmaktadır (17).

İkili sınıflandırma ağaçlarında olduğu gibi CART'da da her iç düğüm, kendisine ait bir bölünme kuralına göre gözlemleri sol ve sağ alt düğüm olmak üzere ikiye ayırır. Sürekli tahminci değişkenler ($x_i$) için bölünme, $b$ bölünme kuralına bağlıdır. Bu kurala göre $(x_i \leq b)$ olan gözlemler sol alt düğüme ve $(x_i > b)$ olanlar ise sağ alt düğüme atanır. Kategorik tahminci değişkenler için ise bölünme kuralı, $C$ kategori altkümesine bağlıdır. $(x_i \in C)$ olan gözlemler sol alt düğüme atanırken $(x_i \notin C)$ koşuluna uyan gözlemler ise sağ alt düğüme yerleştirilir.

Ağaç oluşumunda düğümleri bölerek alt düğümler oluşturmada en çok kullanılan kurallar aşağıdaki gibidir:

**Entropi ve Bilgi Kazancı:** Entropi; bir sistemdeki düzensizliğin ya da belirsizliğin ölçüsüdür. Tek değişkenli karar ağaçlarında örneğin ID3 algoritması bilgi kazancı yaklaşımını kullanmaktadır. Bu algoritmanın geliştirilmiş hali olan C4.5 algoritması bölünme bilgisi kavramı ile bilgi kazancından yararlanarak hesaplanan kazanç oranı yaklaşımını kullanmaktadır. Entropi 0 ve 1 aralığında değerler alır ve 1 değerine yaklaştıkça belirsizlik artar.

Sınıf olasılık dağılımları P $(p_1, p_2, \ldots . p_m)$ olan bir D veri seti olsun. $p_i$; D veri setindeki i sınıfının olasılığıdır ve i sınıfına düşen örnek sayısının tüm veri setindeki toplam örnek sayısına bölünmesi ile elde edilir. Bu bilgiler altında Entropi şu şekilde hesaplanır:

$$E(D) = -\sum_{k=1}^{m} p_i \log_2(p_i) \qquad (2.1)$$

Eğer D veri seti, n tane alt bölüme X değişkeninden bölünecekse X'e ait bilgi kazancı şu şekilde hesaplanır:

$$Bilgi\ Kazancı(D, X) = E(D) - \sum_{k=1}^{n} p(D_i)\, E(D_i) \qquad (2.2)$$

Burada E(D); veri setinin X üzerinden bölünmeden önceki entropisini, $E(D_i)$; i alt bölümünün X üzerinden bölünme olduktan sonraki entropisini ve $p(D_i)$ ise i alt bölümünün X üzerinden bölünme olduktan sonraki olasılığıdır.

Bilgi kazancının en yüksek olduğu değişken en iyi dallara ayırma kriteri olarak seçilir ve bölünmeye o değişkenden başlanılır (16).

**Gini İndeksi:** Bir düğüm bölündüğünde amaç karışıklığı (impurity) en iyi şekilde azaltacak bölünme değerini bulmaktır. Bu nedenle t düğümü için bir *i(t)* karışıklık ölçüsü (measure of impurity) değerine ihtiyaç duyulur. Bu ölçü Gini indeksi ile hesaplanır. Bir sınıflandırma ağacında j tane sınıf olsun ve $p(w_j|t)$; t düğümünde yer alan bir gözlemin j sınıfında olma olasılığını göstersin. Bu durumda karışıklık ölçütü, Gini ile şu şekilde hesaplanır:

$$i(t) = 1 - \sum_{j=1}^{j} p^2(w_j|t) \tag{2.3}$$

Formül (2.3) sonucu elde edilen t düğümündeki karışıklığın, $b$ bölünme kuralından sonra ne miktarda azaldığı ise (2.4) ile bulunur:

$$\Delta i(b,t) = i(t) - p_R i(t_R) - p_L i(t_L) \tag{2.4}$$

$p_R$ ve $p_L$; $b$ ayrımından sonra sırasıyla sağ ve sol alt düğüme gönderilen veri seti oranlarıdır. $\Delta i(b,t)$'nin büyüklüğü; $b$ ayrımının ne derece iyi olduğunu gösterir.

Bağımsız değişken sayısının d olduğu bir eğitim veri seti ile sınıflandırma ağacı oluşturmak istensin. Bir ağaç kök düğümden başlayarak bölünmeye başladığında, her bağımsız değişkenden en iyi ayrımı verecek olan bir başka deyişle $\Delta i(b,t)$ değeri en yüksek olan değişkenler araştırılır. Bölünmeye aday d mümkün bağımsız değişken olsa bile sürekli bağımsız değişkenlerin varlığı sınırsız sayıda mümkün bölünme noktası sunar. Sınırsız sayıda mümkün bölünme noktası, ağaç oluşum süresini uzatarak büyük bir sorun yaratmaktadır. Bu sorun şu şekilde çözülebilir: her bir bağımsız sürekli değişken kendi içerisinde küçükten büyüğe sıralanır. Sıralı arka arkaya gelen değerlerin orta noktaları hesaplanır. Elde edilen bu orta noktalardan karışıklığı en fazla azaltan ($\Delta i(b,t)$ değeri en büyük olan) en iyi bölünme noktasıdır. En iyi bölünmeyi sunan bu değişken ile düğüm bölünmeye başlar ve aynı kural ile ağaç alt düğümlere ayrılarak bölünmeye devam eder.

Her yaprak düğümde durdurma kuralı sağlandıktan sonra ağaç bölünmeyi sonlandırır. Elde edilen yaprak düğümlere sınıf etiketi ataması yapılarak ağaç modeli kurulumu tamamlanmış olur (20).

Regresyon ağaçlarında ise bilinen bir algoritma ile her $R_m$ bölgesi için sabir bir $c_m$ değeri hesaplanır. Böylece tüm ağaç modeli şu şekilde gösterilebilir:

$$f(x) = \sum_{m=1}^{b} c_m \, I(X \in R_m) \tag{2.5}$$

$R_m$ , ($m$=1,2,…,$b$) tahminci değişkenler uzayının parçalarını temsil etmektedir. Aynı zamanda $b$ adet yaprak düğüm uzayını da temsil etmektedir. En iyi ayrımı belirlemek için $\sum\left(y_i - f(X_i)\right)^2$ kareler toplamı fonksiyonunu minumum yapan değer araştırılır. Bölünme gerçekleştikten sonra $c_m$ değeri ise $R_m$ bölgesinde yer alan $y_i$ lerin ortalaması ile tahmin edilir.

$$\hat{c}_m = ortalama(y_i | X_i \in R_m) = \frac{1}{N_m} \sum_{X_i \in R_m} y_i \tag{2.6}$$

$N_m$; m düğümüne düşen gözlemlerin sayısıdır.

Bu hesaplamaların ardından hesaplanacak olan hata kareler toplamı regresyon ağaçları için karışıklık ölçüsü olarak kullanılır (17).

$$Q_m(T) = \frac{1}{N_m} \sum_{X_i \in R_m} (y_i - \hat{c}_m)^2 \tag{2.7}$$

### 2.2.2. Ağacın budanması

Ağaç tabanlı algoritmalar genellikle aşırı uyum (overfitting) oluştururlar. Ağacın büyüklüğü arttıkça test veri setinin hata oranı yükselmekte ve ağacın doğruluğu azalmaktadır. Bu noktada yapılması gereken ise ağaç budama işlemidir. Budama ile eğitim veri setine özgü bazı kurallar silinerek test veri setine ilişkin hata oranının düşük bulunması sağlanır.

CART modellerinde budama genellikle maliyet karmaşıklığı (cost complexity) budama yöntemi ile yapılmaktadır. Öncelikle oldukça büyük bir ağaç ($T_{max}$) elde edilir. Daha sonra ağacın karmaşıklığı için belirlenen maliyet ile yanlış sınıflandırma oranları kullanılarak budama işlemine başlanılır. Ağacın karmaşıklığı alt ağaçtaki ya da daldaki yaprak düğüm sayısına bağlıdır. Maliyet karmaşıklığı ölçüsü ($R_\alpha(T)$) aşağıdaki şekilde tanımlanmaktadır:

$$r(t) = 1 - \max\{\mathrm{p}(w_j|t)\} \tag{2.8}$$

$$R(t) = r(t)p(t) \tag{2.9}$$

$$R(T) = \sum_{t\in\tilde{T}} R(t) \tag{2.10}$$

$$R_\alpha(T) = R(T) + \alpha|\tilde{T}| \tag{2.11}$$

r(t) : t düğümünde yer alan j. sınıfa ait gözlemin tekrar atama sonucu yanlış sınıflandırılma oranı tahmini

p(t) : bir gözlemin t düğümüne düşme olasılığı

R(t) : t düğümü için tekrar atama sonrasındaki risk tahmini

$R(T)$: tüm ağaç üzerindeki tekrar atama sonucunda yanlış sınıflandırma oranı

$\tilde{T}$ : ağaçta yer alan yaprak düğümlerin tümü

$\alpha$ : karmaşıklık parametresi

Eğer her yaprak düğümün sadece bir sınıfa ait gözlem değerlerini içerdiği oldukça büyük bir ağaç elde edersek $R(T)$ sıfıra eşit olur, ama karmaşıklık nedeniyle maliyet karmaşıklığı ölçüsü (2.12)'deki gibi hesaplanır:

$$R_\alpha(T) = \alpha\left|\tilde{T}\right| \tag{2.12}$$

Eğer $\alpha$ değeri küçük ise, karmaşık bir ağaç oluşturmanın cezası daha küçük ve oluşan alt ağaç daha büyük olacaktır. $R_\alpha(T)$ değerini minimize edecek olan ağaç, az sayıda düğüme ve büyük $\alpha$ değerine sahip olur.

$T_t$ ; T ağacının t düğümünden itibaren olan dalını ve bu dalın ardından gelen düğümleri temsil etsin. Eğer $T_t$ dalı budanırsa ya da silinirse, yalnızca t düğümü bırakılarak onun ardından gelen düğümler silinmiş olur.

En küçük karmaşıklık budaması; en zayıf bağlantıya sahip olan dalı araştırır. Budama süreci daha az yaprak düğüme ve azalan karmaşıklığa sahip ardışık alt ağaçlar kümesi oluşturur. En büyük ağaç $T_{max}$ ve kök düğüm ise $t_1$ olsun. Bu durumda ardışık ağaçlar serisi (2.3)'deki gibi gösterilir:

$$T_{max} > T_1 > T_2 > \cdots > T_k = \{t_1\} \tag{2.13}$$

Bu ardışıklığın başlangıç noktası $T_1$ ağacı olup diğer ardışık alt ağaçlardan farklı bir şekilde elde edilmektedir. $T_{max}$'da yer alan yaprak düğüm çiftlerinde (aynı düğümden ayrılıp yaprak düğüm olarak sonuçlanmış düğüm çiftleri) yanlış sınıflandırma oranlarına bakılır.

$$R(t) \geq R(t_L) + R(t_R) \tag{2.14}$$

Yukarıdaki eşitsizlik, t düğümündeki yanlış sınıflandırma oranının, bu düğüme ait alt düğümlerdeki yanlış sınıflandırma oranları toplamından büyük ya da eşit olduğunu gösterir. İlk aşamada $T_{max}$'da yer alan ve aşağıdaki koşulu

sağlayan yaprak düğüm çiftleri bulunarak budanır. Bu aşama tamamlandıktan sonra $T_1$ elde edilir.

$$R(t) = R(t_L) + R(t_R) \tag{2.15}$$

Karmaşıklık parametresi $\alpha$ için büyük bir değer kümesi vardır, ama verilen bir $\alpha$ değeri için $R_\alpha(T)$'yi minimize edecek T($\alpha$) ağaçları elde etmek amaçlanmaktadır. Bu nedenle, her seviyede karmaşıklık maliyetini minimize edecek ardışık $\alpha$ değerleri araştırılır. $T_1$ ağacı elde edildikten sonra, ağaç dallarını budamak için en zayıf bağlantıya ihtiyaç duyulur. Bu en zayıf bağlantı aşağıdaki fonksiyon ile elde edilir:

$$g_k(t) = \frac{R(t)-R(T_{kt})}{|\tilde{T}_{kt}|-1} \tag{2.16}$$

Burada t bir iç düğümü, $T_{kt}$; t düğümünden oluşan alt $T_k$ ağacına bağlanan dalı, $|\tilde{T}_{kt}|$; $T_{kt}$'ye ait yaprak düğüm sayısını gösterir. $T_k$ ağacında yer alan en zayıf bağlantı $t_k{}^*$; $g_k(t)$'yi minimize eden t iç düğümüdür.

$$g_k(t_k{}^*) = min_t\{g_k(t)\} \tag{2.17}$$

Bu düğüm belirlendikten sonra budanarak, $T_{k+1}$ ardışık ağacı elde edilir. Daha sonra ise $\alpha$ karmaşıklık parametresine aşağıdaki atama yapılır:

$$\alpha_{k+1} = g_k(t_k{}^*) \tag{2.18}$$

Budama süreci bu şekilde devam ederek gittikçe küçülen alt ağaç kümeleri elde edilir. Bu süreçte karmaşıklık parametresi $\alpha$ da artan bir eğilim gösterir.

$$T_{max} > T_1 > T_2 > \cdots > T_k = \{t_1\} \tag{2.19}$$

$$0 = \alpha_1 < \cdots < \alpha_k < \alpha_{k+1} < \cdots < \alpha_K \tag{2.20}$$

Elde edilen ardışık alt ağaçlar serisinden en iyi ağaç seçimi için bir koşul tanımlanır; $k \geq 1$ olacak şekilde $T_k$ ağacı ; $\alpha_k < \alpha < \alpha_{k+1}$ aralığı için en küçük maliyet karmaşıklığı ağacıdır (20).

$$T(\alpha) = T(\alpha_k) = T_k \tag{2.20}$$

### 2.2.3. En iyi ağacın seçilmesi

En iyi ağacın seçilmesi için kullanılan yöntemlerin başında çapraz geçerlilik (cross-validation) algoritması gelmektedir. Bu algoritma ile ağaçlar için sınıflandırma hatası tahmin edilir. Elimizde bulunan eğitim veri setini ($L$) birkaç defa farklı biçimde eğitim ve test veri seti olarak parçalara ayrılır. Daha sonra eğitim veri setleri kullanılarak ardışık ağaçlar elde edilir ve test setleri ile sınıflandırma hataları tahmin edilir.

$L_v$ ; L eğitim setinin v adet alt parçaya ayrıldıktan sonraki herhangi bir bölümü olsun. Bu parçanın dışındaki tüm veri seti de $L^{(v)}$ ile gösterilsin.

$$L^{(v)} = L - L_v \; ; \quad v=1,2,\ldots,V \tag{2.21}$$

$T_k{}^{(v)}$; $L^{(v)}$ kullanılarak oluşturulan ağacı, $\alpha_k{}^{(v)}$; bu ağaç oluşumunda kullanılan karmaşıklık parametresini ve $\hat{R}^{CV}(T)$ ağaç için bekleyen yanlış sınıflandırma maliyeti tahminini gösterir.

Bu yöntemi kullanırken öncelikle $L$ eğitim seti v adet alt parçaya ayrılır. Breiman bu ayrımda kullanmak üzere V=10 değerini önermiştir. $L^{(v)}$ kullanılarak ardışık alt ağaç kümeleri elde edilir.

$$T_{max}{}^{(v)} > T_1{}^{(v)} > \cdots > T_k{}^{(v)} > T_{k+1}{}^{(v)} > \cdots > T_K{}^{(v)} \tag{2.22}$$

Her eğitim seti parçası için bu alt ağaç toplulukları oluşturulur. Oluşturulan $T_k^{(v)}$ ağaçları, en başta tüm eğitim veri seti $L$ kullanılarak elde edilen ardışık alt ağaç kümesindeki $T_k$'ların sınıflandırma performansını değerlendirmek için kullanılır. Ayrıca her bir ardışık alt ağaç kümelerinde hesaplanmak üzere bu setler ile eşleştirilmiş karmaşıklık parametresi setleri de vardır:

$$0 = \alpha_1^{(v)} < \cdots < \alpha_k^{(v)} < \alpha_{k+1}^{(v)} < \cdots < \alpha_K^{(v)} \tag{2.23}$$

Bu durumda V+1 tane ardışık alt ağaç kümesi ve karmaşıklık parametresi seti olacaktır.

$T_k^{(v)}$ ağacında $L_v$ test örneği kullanılarak $T_k$ alt ağaçlarındaki sınıflandırma hatasına karar verilir. Bunu başarmak için, $T_k^{(v)}$ ağaçları ardışıklığından $T_k$'ya denk karmaşıklıkta ağaçlar bulunmalıdır. $T_k$; $\alpha_k < \alpha < \alpha_{k+1}$ aralığı için en küçük maliyet karmaşıklığı ağacıdır. Bu tanımlı aralık kullanılarak temsilci karmaşıklık parametresi geometrik ortalama ile hesaplanır:

$$\alpha'_k = \sqrt{\alpha_k \alpha_{k+1}} \tag{2.24}$$

Daha sonra yanlış sınıflandırma oranı aşağıdaki formülle elde edilir:

$$\hat{R}^{CV}(T) = \hat{R}^{CV}(T(\alpha'_k)) \tag{2.25}$$

En iyi alt ağacı seçmek için yanlış sınıflandırma hatası $\hat{R}^{CV}(T)$'nın standart hatası için bir ifadeye ihtiyaç duyulur. $L_v$'den gelen test veri seti ağaca sunulduğunda her doğru sınıflandırma için sıfır, yanlış sınıflandırma için ise bir değeri kaydedilir. Bu durumda $\hat{R}^{CV}(T)$ tahmin değeri, sıfırların ve birlerin ortalamasına eşit olur. Standart hata ise bu durumda (2.26)'daki gibi elde edilir:

$$\widehat{SE}\left(\hat{R}^{CV}(T_k)\right) = \sqrt{\frac{s^2}{n}} \qquad (2.26)$$

$s^2$; birler ve sıfırlardan oluşan örneklemin varyansı ile (n-1)/n değerinin çarpımına eşittir.

Yukarıdaki bilgiler göz önünde bulundurularak çapraz geçerlilik yöntemi ile yanlış sınıflandırma hatasının tahmin yöntemi algoritması sonucu en uygun ağaç seçilir.

Bu algoritma aşağıdaki gibidir:

1. $L$ öğrenim veri seti kullanılarak $T_k$ alt ardışık ağaçları elde edilir.
2. Her $T_k$ ağacı için $\alpha'_k$ maliyet karmaşıklığı parametresi belirlenir.
3. $L$, V parçaya ayrılarak $L_v$'ler elde edilir ve bunlar ağaçları test etmede kullanılır.
4. Her $L_v$ için, $L^{(v)}$' ler kullanılarak ardışık alt ağaçlar elde edilir. Bu durumda V+1 tane ardışık alt ağaç kümesi elde edilmiş olur.
5. Tahmin edilmiş yanlış sınıflandırma hatası $\hat{R}^{CV}(T_k)$ bulunur. $T_k$'ya karşılık gelen her $\alpha'_k$ için tüm denk $T_k^{(v)}$'ler elde edilir. $\alpha'_k \epsilon \left[\alpha_k^{(v)}, \alpha_{k+1}^{(v)}\right)$ koşulundaki $T_k^{(v)}$ seçilir.
6. Her $L_v$ için test verisi 5. Adımda bulunan $T_k^{(v)}$'ye sunulur. Doğru sınıflandırma 0, yanlış sınıflandırma 1 ile kaydedilir. Bunlar sınıflandırma maliyetidir.
7. Kaydedilen sıfırların ve birlerin ortalaması alınarak $\hat{R}^{CV}(T_k)$ elde edilir.
8. Standart hatası hesaplanır.
9. 5'den 8'e kadar tüm adımlar kullanılarak her alt $T_k$ ağacı için yanlış sınıflandırma maliyeti hesaplanır.
10. En küçük hata bulunur.

$$\hat{R}^{CV}{}_{min} = min_k\{\hat{R}^{CV}(T_k)\} \quad (2.27)$$

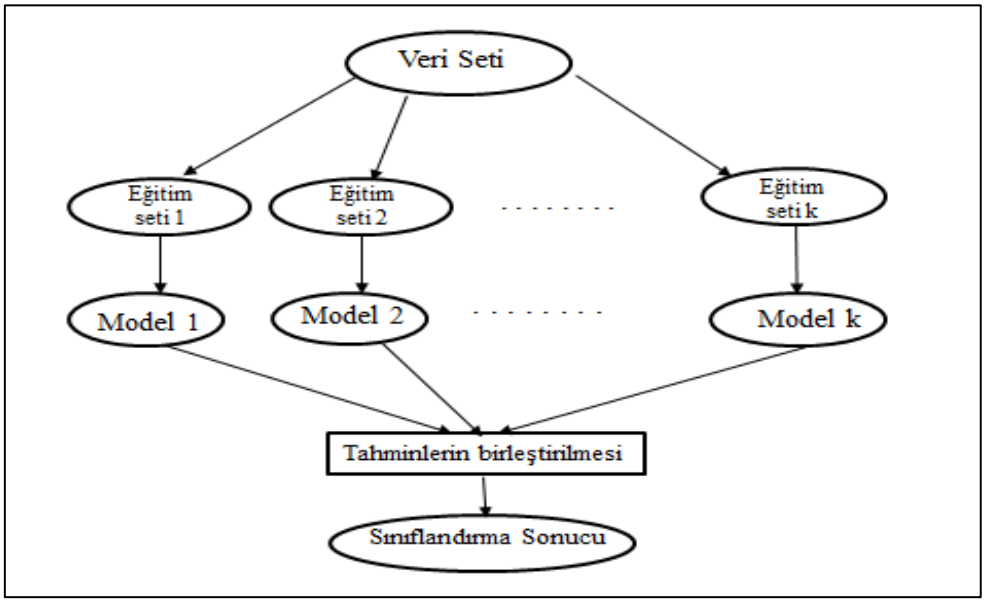
11. $\hat{R}^{CV}{}_{min} + \widehat{SE}(\hat{R}^{CV}{}_{min})$ değeri hesaplanır.
12. 11. Adımda hesaplanan değerden küçük olan en büyük k değerine ya da en az sayıda düğüme sahip olan ağaç, sınıflandırma ağacı olarak seçilir (20).

## 2.3. Ağaç Tabanlı Topluluk Yöntemler

Ağaç tabanlı topluluk yöntemler, birleştirilmiş bir tahmin vermek için çok sayıda farklı ağacın tahminlerini bir araya getirir. Farklı ağaçlar elde etmek için bazı topluluk yöntemler rasgeleliği kullanırken, diğerleri veri setinin farklı türleri için rasgele olmayan ağaçlar oluşturur. Bazı yöntemler ise her iki stratejiyi birden uygular. Bu yöntemler ayrıca tahminci değişkenlerin birleştirilmesi açısından da farklılık gösterirler. Regresyon ağaçlarında birleştirilmiş tahmin yöntemine, tekil ağaçlardan elde edilen tahminlerin ortalamaları örnek olarak gösterilebilir. Sınıflandırma ağaçlarında ise kullanılan en basit tahmin yöntemi oylama (voting)'dır. Topluluktaki ağaçların oylanması sonucu bir gözlem için en çok elde edilen sınıf değeri, o gözlemin tahmin edilen sınıf değeri olur. Şekil 2.2'de ağaç tabanlı topluluk yöntemlerin oluşturulmasına yönelik bir şema yer almaktadır.

Topluluk yöntemler tekil ağaçları kullanarak geliştirilmiş tahmin kesinliği verebilir. Topluluk sınıflandırıcılarının tahmin kesinliği arkasındaki düşünce, eğer tekil ağaçlar tahminci değişkenler uzayının farklı bölgelerinde tahmin hatası oluşmasına eğilimli olurlarsa, bu tekil ağaçlar doğru olan diğerlerinin oy üstünlüğü sonucunda dışlanırlar. Topluluk yöntemlerin daha kesin tahminler vermesindeki bir başka neden ise tekil ağaçları düzelten bir yol kullanarak daha düzgün regresyon ya da sınıflandırma için daha düzgün sınırlar belirlemesidir (11).

Son zamanlarda ağaç tabanlı topluluk yöntemlere olan ilgi oldukça artmıştır. Bu yöntemlerden en bilinen ve kullanılanlar Bagging, Boosting ve RF algoritmalarıdır.




Şekil 2.2. Topluluk yöntemlerin oluşturulması

### 2.3.1. Bagging

Bagging (bootstrap aggregating) yöntemi 1996 yılında Breiman tarafından geliştirilmiştir. Orijinal veri setinden elde edilen bootstrap örneklerine tahminciler uygulanarak bir topluluk oluşturulur. Burada bootstrap uygulaması, iadeli rasgele seçim yapıp alt örneklemler oluşturmak için kullanılır. Oluşturulan alt örneklemler orijinal veri setindeki sayı ile aynı olacaktır. Bu nedenle bazı gözlemler bootstrap sonucunda oluşturulan örneklemlerde yer almazken bazıları iki veya daha fazla defa görülebilir. Tahminlerin birleştirilmesi aşamasında regresyon ağaçları için ortalama alınırken sınıflandırma ağaçlarında sonuçlar oylama ile belirlenir.

Bagging, tutarsız bir tahminci değişkenin tahmin geçerliliğini de arttırabilir. Düşük yanlılık miktarına sahip ama yüksek varyanslı olan değişkenleri kullanarak onları daha elverişli hale getirir. Ayrıca deneysel sonuçlara göre Bagging yöntemi, tekil ağaçlara göre daha etkin sonuçlar vermektedir.

Basit bir biçimde yorumlanabilen bir yöntem değildir. Bu yöntemde farklı ağaçların oluşmasındaki tek neden farklı bootstrap örneklemlerinin kullanılmasıdır. Sezgisel olarak bu durumda birbirine benzeyen ağaçlar benzer hatalar yapmaya eğilimli olacaktır (2,10,11).

### 2.3.2. Boosting

Boosting yöntemindeki temel fikir, veri setine farklı ağırlıklar verilmesi sonucu elde edilen ağaçlar topluluğundan çıkarsamalar yapılmasıdır. Başlangıçta tüm gözlemler eşit ağırlığa sahiptir. Ağaç topluluğu büyümeye başladıkça, problem bilgisine kurulu olarak ağırlıklandırmalar düzenlenir. Yanlış sınıflandırılan gözlemlerin ağırlığı arttırılırken, nadiren yanlış sınıflandırılan gözlemlerin ağırlığı azaltılır. Bu sayede ağaçlar zor durumlar karşısında kendini düzenleyebilme yeteneği kazanır.

Boosting yönteminde birden fazla algoritma geliştirilmiştir. Bu algoritmalar;

- Tekil ağaç,
- Ağırlıklandırmaları değiştirme yöntemi,
- En son tahmini verirken kullanılan birleştirici yöntem

seçimlerindeki farklılıklardan dolayı birbirlerinden ayrılırılar. Bu algoritmalardan en çok bilinenleri Freund ve Schapire tarafından 1996 yılında geliştirilen Adaboost ve 2001 yılında Freidman tarafından geliştirilen Gradient Boosting Makinalarıdır (10,11).

## 2.4. Random Forests

### 2.4.1. Tanımı ve algoritması

RF yöntemi 2001 yılında Leo Breiman tarafından geliştirilmiştir. Breiman, kendisinin 1996 yılında geliştirdiği Bagging yöntemi ile Ho tarafından 1998'de önerilen ve rasgele alt gruplar seçmek için kullanılan The Random Subspace tekniğini birleştirerek yeni bir yöntem oluşturmuştur. Breiman bu yöntemi geliştirirken; Amit ve Geman tarafından 1997'de tanımlanan, her düğüm için en iyi ayrımın rasgele bir seçim üzerinden belirlendiği belirtilen bir çalışmadan da etkilenmiştir (8).

RF bir topluluk öğrenme yöntemidir. Birbirinden farklı olarak kurulan sınıflama ve regresyon karar ağaçları (CART) karar ormanı topluluğunu oluşturur. Karar ormanı oluşumu sırasında elde edilen sonuçlar bir araya getirilerek en son tahmin yapılır. RF yönteminde ağaçlar, seçilen bootstrap örneklemleri ve her düğüm ayrımında rasgele seçilen *m* adet tahminci ile oluşturulur. *m* adet tahmincinin toplam tahminci sayısından oldukça küçük olmasına dikkat edilir. Oluşturulan her bir karar ağacı en geniş haliyle bırakılır ve budanmaz. Sınıflandırma için ağaçlar; her yaprak düğümü sadece bir sınıfın üyelerini içerecek şekilde oluşturulurlar. Regresyon için ise; yaprak düğümde az sayıda birim kalana kadar ağaçlar bölünmeye devam ederler (11).

RF yöntemi bilinen makine öğrenme yöntemleri içerisinde eşsiz bir tahmin geçerliliği ve model yorumlanabilirliği sağlar. Rasgele örneklemeyi ve topluluk yöntemlerdeki tekniklerin iyileştirilmiş özelliklerini içermesi nedeniyle RF yöntemi daha iyi genellemeler sunar ve geçerli tahminlerde bulunur (23). RF yönteminin tahminlerinin kesinliğinin nedenleri yanlılığı düşük sonuçlar vermesi ve ağaçlar arasındaki düşük korelasyondur. Düşük yanlılık miktarı, oldukça büyük ağaçların oluşturulması sonucu elde edilir. Mümkün olduğunca birbirinden farklı ağaçlar oluşturarak da düşük korelasyon yapısında bir topluluk elde edilir.

Sınıflandırma makinesi olarak RF;

1) Mükemmel bir geçerlilik sunar. Pek çok veri seti için Adaboost ve Destek Vektör Makinalarından (Support Vector Machines) daha kesin sonuçlara sahiptir.

2) Oldukça kısa sürede sonuç verir. 100 değişkenli 100 ağaçlık bir karar ormanı, arka arkaya kurulan 3 tekil CART ile aynı sürede oluşturulur.

3) Binlerce değişkene ve fazla sayıda sınıf etiketine sahip kategorik değişken içeren, kayıp verili veya dengesiz bir dağılım sergileyen veri setlerini kullanarak sonuçlar verir.

4) Topluluğa ağaçlar eklendikçe, test setine ait hata tahmini için yanlılığı düşük sonuçlar vermeye başlar.

5) Aşırı uyum sergilemez (9).

RF modeli 2 parametre üzerine kuruludur. Bu parametreler; oluşturulacak olan ağaç sayısı (B) ve her düğüm ayrımında rasgele seçilecek olan tahminci sayısıdır (m). Her karar ağacı oluşturulurken, orijinal veri setindeki gözlem sayısı (n) ile aynı ölçüde olacak şekilde bootstrap yöntemi ile örneklem oluşturulur. Bu örneklemin 2/3'ü ağacı oluşturmak için kullanılan eğitim veri seti (inBag) ve geriye kalan 1/3'ü ise kurulan modelin iç hata oranını test etmek için test veri seti (out of bag veya OOB) olacak şekilde ikiye ayrılır.

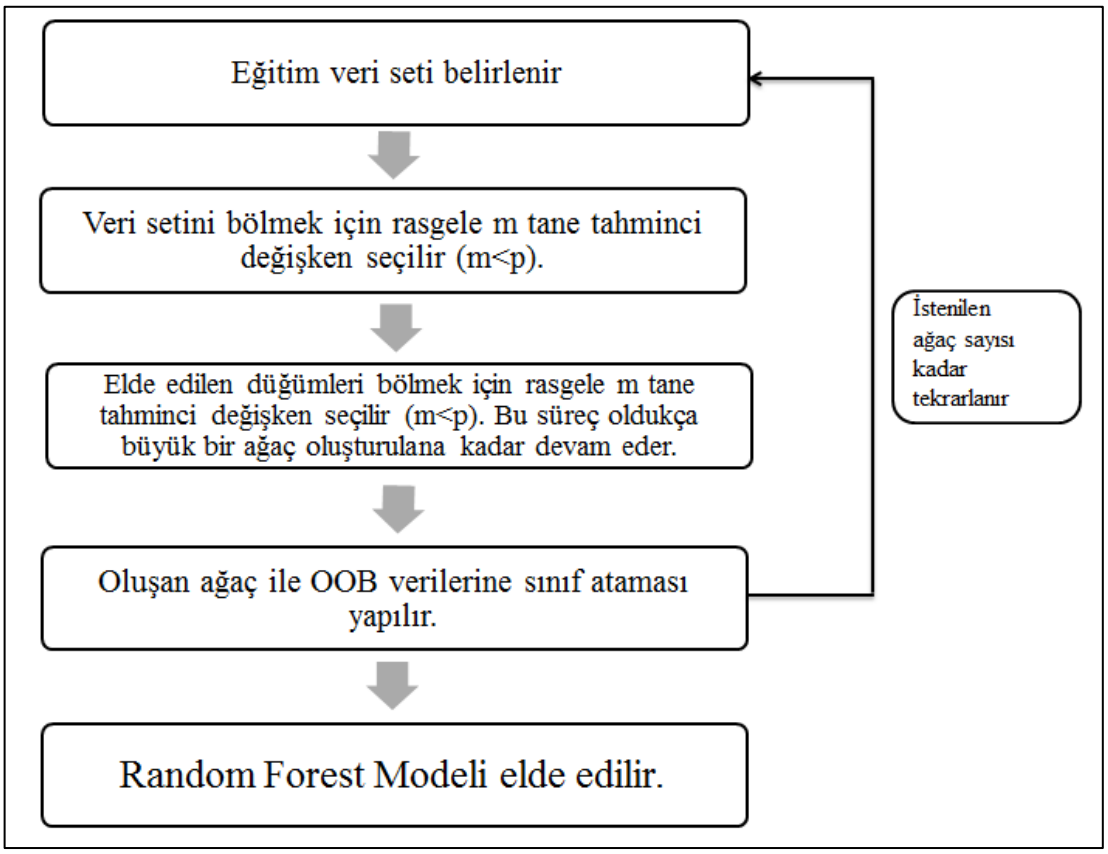
RF algoritması aşağıdaki şekilde kurulur:

1) Bootstrap yöntemi ile n hacimli veri seti seçilir. Bu veri seti, eğitim veri seti (inBag) ve test veri seti (OOB) olarak ikiye ayrılır.

2) Eğitim veri seti (inBag) ile en büyük genişlikte bir karar ağacı (CART) oluşturulur ve elde edilen bu karar ağacı budanmaz. Bu ağaç oluşturulurken her düğümün bölünmesinde toplam p tane tahminci değişkenden m tanesi rasgele seçilir. Burada m<p koşulu sağlanmalıdır. Çünkü ağacın aşırı büyümesi ve aşırı uyum gözlenmesi istenmemektedir. Seçilen bu m tane tahminciden bilgi kazancı en yüksek olan ile dallara ayrılma gerçekleşir. Belirlenen bu

değişkenin hangi değerine göre ayrımın olacağına Gini indeksi ile karar verilir. Bu işlem her düğüm için yeni oluşturulacak dal kalmayıncaya kadar tekrar edilir.

3) Her yaprak düğüme bir sınıf atanır. Daha sonra test veri seti (OOB) ağacın en tepesinden bırakılır ve bu veri setinde yer alan her gözleme atanan sınıf kaydedilir.
4) 1.'den 3. adıma kadar tüm aşamalar B defa tekrar edilir.
5) Ağaç oluşturulurken kullanılmayan gözlemler (OOB) ile bir değerlendirme yapılır. İncelenen bir gözlemin hangi kategorilerde kaç defa sınıflandırıldığı sayılır.
6) Her gözleme, ağaç setleri üzerinden belirlenen bir oy çoğunluğu ile sınıf ataması yapılır. Örneğin 2 kategoriye sahip bir sınıflandırma modelinde, bir gözlem tüm ağaçlar üzerinden en az %51 oy çoğunluğunu aldığı sınıfın etiketini taşır ve bu sınıf onun tahmin edilmiş sınıf değeri olur.




Şekil 2.3. Random Forests modeli oluşturma algoritması

### 2.4.2. Random Forests yönteminin özellikleri

#### 2.4.2.1. Genelleme hatası (Generalization error)

Veri setinden bir bootstrap örneklemi seçildiğinde, bazı gözlemler ağaç oluşturma aşamasında yer almaz. OOB olarak adlandırdığımız bu gözlemler ile genelleme hatasına yönelik bir iç tahmin yapılır. OOB hata oranını elde etmek için, her ağaç OOB veri seti için bir sınıf değeri tahmin eder ve bu tahminler kaydedilir. Herhangi bir noktada, her bir gözlem için OOB olduğu ağaçlardaki hata oranı tahminlerinin ortalaması alınarak genelleme hatası hesaplanabilir. Genel bir hata oranı ise tüm gözlemlerin ortalaması alınarak hesaplanabilir (10, 11).

#### 2.4.2.2. Parametreleri ayarlama (Tunning parameters)

RF yönteminde karar ormanı oluşturulurken belirlenmesi gereken 2 parametre vardır; her düğümde rasgele seçilecek olan değişken sayısı ($m$) ve oluşturulacak ağaç sayısıdır (B). RF bu parametrelerin seçiminde hassas bir yapı sergilemez. Breiman, bu parametrelerin seçimi için bazı önerilerde bulunmuştur. Breiman'a göre 500 adet ağaçtan oluşacak bir karar ormanı yeterli sayılabilir. Pek çok sınıflandırma problemi için her düğümde rasgele seçilecek olan değişken sayısı $m = \sqrt{p}$ eşitliği ile hesaplanmaktadır. Burada $p$; veri setindeki tahminci değişkenleri sayısını göstermektedir. Regresyon ağaçlarında ise $m$ parametresi $m=p/3$ olarak elde edilir.

RF oluşturulurken ormana daha fazla sayıda ağaç eklemek aşırı uyumun oluşmasına neden olmamaktadır. Ağaçların sayısı için ilgilenilen önemli nokta yeterli büyüklükte olmasıdır. Bu sayı, OOB hata oranı kullanılarak kontrol edilir. Şekil 2.4'de görüldüğü gibi OOB hata oranı belirli bir ağaç sayısından sonra sabit bir değere yakınsar.

Şekil 2.4. Oluşturulan ağaç sayısına göre hata oranı değişimi

Bazı kısıtların önceden tanımlandığı özelleştirilmiş problemlerde farklı parametreler için ayarlamalar yapılabilir. Örneğin regresyon problemlerinde ağaçların derinliğinin ya da yaprak düğümlerde kalacak olan minimum gözlem sayısının kontrol edilmesi gereklidir (10, 11).

#### 2.4.2.3. Değişken önemliliği (Variable Importance)

Değişken önemliliği, bir değişkenin tahmin ediciliğini ölçer. Tahminci değişkenlerin önemliliğinin ölçümü, değişken seçimi ve kurulmuş ormanı yorumlamak için kullanışlıdır. Bazı istatistiksel analizler uygulanmadan önce, yüksek boyutlu veri setini indirgemek için temel bileşenler analizi kullanılsa da, bu yöntem tahmin için önemli bilgileri yakalayamamaktadır. Bu durumda değişken önemliliğini direk algoritmadan gözlemlemek ve önemli değişkenler kullanılarak model kurmak daha çok tercih edilen bir durumdur (10, 11).

RF sınıflandırma kuralları oluşturulurken doğrudan değişken seçimini gerçekleştirir. Değişken önemliliğinin hesaplanmasındaki en önemli amaçlar; model performansını geliştirerek aşırı uyumu engellemek ve veri setini türeten sürecin altında yatan kavramı daha derinden anlamaktır (23).

Değişken önemliliği birbirine paralel sonuçlar veren iki yöntem ile hesaplanabilir. Bunlar; Gini önemliliği ve permütasyona dayalı değişken önemliliğidir.

#### 2.4.2.3.1 Gini önemliliği:

Gini önemliliği, doğrudan RF ağaçları oluşturulurken kullanılan Gini indeksinden elde edilir. Gini indeksi bir düğüme atanmış örneklemin karışıklık ya da eşitsizlik seviyesini ölçer. Örneğin, iki sınıflı bir sınıflandırma probleminde p; k düğümünde yer alan pozitif gözlemlerin oranını ve 1-p de negatif gözlemlerin oranını göstersin. Bu durumda k düğümünde yer alan Gini indeksi aşağıdaki gibi hesaplanır:

$$G_k = 2p(1-p) \tag{2.28}$$

Bir düğüm ne kadar saflaştırılırsa, Gini değeri de o kadar küçülür. Bir düğümde *v* değişken üzerinden bölünme gerçekleştiğinde elde edilen yeni iki düğümün Gini değeri, bölünen düğümün Gini değerinden daha küçük olur. Her bir tekil ağaç için *v* değişkeninin Gini önemlilik değeri bu iki değer arasındaki fark hesaplanarak elde edilir. Ormandaki tüm ağaçlar oluşturulduktan sonra, *v* değişkenin yer aldığı ağaçlardaki Gini önemlilikleri toplanarak *v* değişkenine ait önem derecesi belirlenmiş olur (10, 11).

#### 2.4.2.3.2 Permütasyona dayalı değişken önemliliği

RF yönteminde *v* değişkeninin önem derecesi aşağıdaki sıralama ile bulunur. Öncelikle OOB gözlemleri ağaçtan aşağı bırakılır ve tahmin edilen değerler belirlenir. Daha sonra ise OOB' de yer alan diğer tahminci değişkenler sabit olmak koşulu ile *v* değişkenine ait gözlem değerleri rasgele karıştırılır.

Elde edilen yeni OOB veri seti ağaçtan aşağı bırakılır ve tahmin edilen değerler belirlenir. Bu işlem sonucunda her gözlem için iki tane tahmin değeri elde edilmiş olur. Orijinal OOB ile elde edilen doğru tahmin sayısından, değiştirilmiş OOB ile elde edilen doğru tahmin sayısı çıkartılarak bir fark elde edilir. Bu işlem tüm ormana uygulanarak ormandaki ağaç sayısı kadar fark elde edilir ve bu farkların ortalaması hesaplanır. Tüm ağaçların birbirinden bağımsız olduğu ve elde edilen fark değerlerinin normal dağıldığı varsayımı altında *v* değişkeni için z skor değeri hesaplanır. Bu skor değeri; farklar ortalamasının farkların standart hatasına oranlanması ile hesaplanır. Ağaçta yer alan her *v* değişkeni için skor değerler elde edilir. Elde edilen skor değerlerine göre değişkenlerin önemlilik dereceleri kıyaslanarak bir sıralama belirlenmiş olur (2, 10, 11).

#### 2.4.2.4. Farklı sınıf büyüklükleri (Unequal class sizes)

Sınıflara ait gözlem sayılarının birbirinden farklı olduğu dengesiz veri setleri pek çok sınıflandırıcı için sorun oluşturmaktadır. Saf bir sınıflandırıcı gözlem sayısı büyük sınıflara odaklanacağı için bu sınıflar üzerinden büyük bir hata oranına sebep olacaktır. RF, dengesiz veri setlerinde dengeli sonuçlar vermek için etkin bir yöntem ile sınıfları ağırlıklandırır. Bunu yapmasındaki önemli bir sebep, yöntemin gözlem sayısı küçük olan sınıflara daha fazla dikkat etmesi sonucunda önemli tahminci değişkenlerde farklılıklar görebilmesidir. Dengeli olan veri setlerinde bile, yüksek derecede yanlış sınıflandırma maliyetine sahip kararlara daha düşük hata oranları vermek için ağırlıklandırmalarda düzenleme yapılabilir (10, 11).

#### 2.4.2.5. Örnekler arası uzaklık (Proximity)

Yüksek boyutlu veri analizlerindeki en çok karşılaşılan zorluklardan biri, veri setinin tutarlı olup olmadığını net bir şekilde gözlemleyememektir. Bilinen sınıflarda alt grup oluşumu ya da buna benzer örüntüler var mıdır? Sapan değerler var mıdır? Çok sınıflı durumlarda bazı gözlemler birbirleri ile örtüşürken, bazıları birbirinden ayrı mıdır? RF bu soruların iç yüzünü anlamak için veri setine bir bakış açısı sunar. Bunu, gözlem çiftleri arasında uzaklık ölçüsü (proximity measure) hesaplayarak yapar. İki gözlem arasındaki uzaklık, aynı yaprak düğümde sonlanma oranlarına eşittir. Bu oran ormandaki ağaçlar üzerinden hesaplanır. RF bu uzaklık ölçüsünü kullanarak bir uzaklık matrisi (proximity matrix) oluşturur.

Uzaklık matrisi *nxn* boyutlarında ve simetriktir. Burada *n*; ağaç oluşumunda kullanılan veri setindeki tüm gözlemlerin sayısıdır. Veri setinin tümü (inBag ve OOB) ağaçtan aşağı bırakılır. Eğer *i.* ve *j*. gözlemleri aynı yaprak düğümde sonlanırsa aralarındaki uzaklık 1 arttırılır. Veri seti ormandaki bütün ağaçlara yerleştirilip uzaklıklar elde edildikten sonra ortaya çıkan matrisin her bir gözesi, ormandaki ağaç sayısına bölünür. Böylece uzaklık oranları elde edilmiş olur. Eğer iki gözlem değeri her zaman aynı yaprak düğümde sonlanırsa uzaklıkları 1'e, hiçbir zaman aynı yaprak düğümde olmazlar ise de 0'a eşit olur. Uzaklık oranları oldukça yüksek olan gözlemler birbirlerine daha benzer bir yapı gösterirlerken, diğer gözlemlerle arasındaki uzaklık oranı oldukça düşük olanlar sapan değer (outlier) şüphesi taşırlar (2, 8, 10, 11).

#### 2.4.2.6. Kayıp değer atama (Missing value imputation)

Kayıp değer pek çok veri setinde ortaya çıkan bir problemdir. RF, kayıp değerleri olan gözlemleri veri setinden dışlamak yerine kendi içinde geliştirdiği

bir algoritma ile onların veri setinde kalmasına olanak sağlar. Bu algoritmanın temelini bölüm 2.4.2.5 de açıklanan uzaklık ölçüsü oluşturmaktadır.

Kayıp değer atama algoritması aşağıdaki şekilde ifade edilmektedir. Öncelikle veri setindeki kayıp veriler tespit edilir. Kayıp verinin ait olduğu değişken sürekli ise, bu değişkene ait eksiksiz verilerin medyan değeri bulunarak kayıp veriye atama yapılır. Eğer kayıp verinin ait olduğu değişken kategorik ise, eksiksiz verilerden en yüksek frekans değerine sahip olan kategori ile atama yapılır. Elde edilen tamamlanmış veri seti ile bir RF modeli kurulur. Bu modelden bir uzaklık matrisi elde edilir. Elde edilen bu matristeki uzaklıklar ağırlıklandırma ölçüsü olarak kullanılır. Sürekli bir değişkene ait kayıp değerler için, eksiksiz verilerin uzaklık ölçüleri kullanılarak ağırlıklı ortalaması hesaplanır. Elde edilen değer kayıp veriye atanır. Kategorik kayıp verilere ise, eksiksiz verilerden uzaklık oranı en yüksek olanın kategori değeri atanır. Yeni atama işlemleri tamamlandıktan sonra elde edilen yeni veri seti ile tekrardan bir RF modeli kurulur ve yeni bir uzaklık matrisi elde edilir. Aynı kurallar çerçevesinde kayıp değerlere tekrardan yeni atamalar yapılır.

Uzaklık matrisi kullanılarak kayıp değer atamasının yapıldığı bu süreç tutarlı bir sonuç belirlemek için 5 defa tekrar edilir. Bu yöntem bir tür uzaklık tabanlı en yakın komşuluk yöntemi olduğu için, kayıp verilerin rasgele olduğu durumlarda geçerli olacaktır (10, 11, 14).

RF yönteminin, ağaç tabanlı topluluk yöntemler içerisinde üstün olmasını sağlayan özellikleri aşağıdaki şekilde özetlenebilir:

- Her düğüm ayrımında rasgele tahminci değişkenler ile çalıştığı için yerleştirilen veriler açısından ağaçlar birbirinden bağımsızdır.
- Genellikle regresyon analizinde tahminci sayısının veri setindeki gözlem sayısından küçük olması gerekmektedir. RF yönteminde böyle bir zorunluluk yoktur.

- Çok sayıda ağacın kullanılması, RF uygulama fonksiyonunu CART uygulama fonksiyonundan daha karışık hale getirir. Ama bunun yanı sıra da model performansını değerlendirmede OOB veri setini kullanarak iç hata oranı hesaplar. Böylece CART için hassas bir problem olan aşırı uyumu telafi eder.
- Pek çok sınıflandırıcıya göre doğruluk payı oldukça yüksektir.
- Orman oluşturma sürecinde yansız genelleştirilmiş hata tahmini yapar.
- Kayıp veri tahmininde etkin bir yöntemdir.
- Dengesiz sınıflandırılmış toplum veri setlerinde hatayı dengeleyen bir yöntemdir.
- Başka veri setlerinde kullanmak için türetilen ormanlar kaydedilebilir.
- Sınıflandırmada hangi değişkenin önemli olduğuna dair tahminler verir.
- Kümeleme, sapan değerleri belirleme ya da ölçekleme için gözlem çiftleri arasındaki uzaklıkları hesaplar (14).

RF yönteminin üstün özelliklerinin dışında bazı kısıtları da bulunmaktadır. Bunlar aşağıdaki şekilde sıralanabilir:

- Tek bir karar ağacında olduğu gibi ortaya çıkan sonuç, ağaç yapısında görsel olarak görülmez.
- Ortaya çıkan sonuç için bir güven aralığı veremez (2).

## 2.5. Kayıp Veri Analizi

Kayıp veri, istatistiksel analizlerin uygulanmasında oldukça büyük bir problemdir. Hastaların ölümü, araç gereç arızası, anket çalışmalarında katılımcıların bazı soruları cevaplamayı reddetmesi gibi nedenlerden dolayı veri setlerinde kayıp veri problemi ortaya çıkmaktadır.

Makine öğrenme yöntemleri, temel bileşenler analizi, kümeleme gibi doğrudan veri setinden alınan bilgi doğrultusunda kurulan algoritmalar için

kullanılacak verinin kalitesi oldukça önemlidir. Kayıp veri kaliteyi düşüren önemli bir faktördür. Ayrıca, örneklem hacminde azalma olması nedeniyle istatistiksel güç de oldukça düşük olacaktır.

Kayıp veri ile ortaya çıkan bu problemlere çözüm olarak 1987 yılında Little ve Rubin tarafından yayınlanan ve 2002 yılında yeniden düzenlenen "Statistical Analysis with Missing Data" kitabı yayınlanmıştır (19). Bu kitap günümüzde kayıp veri için geliştirilen pek çok yöntemin ve bilgisayar programlarının temelini oluşturmaktadır. Kayıp veri yapıları için belirli mekanizma yapılarının olduğu belirtilen bu kitapta ayrıca kayıp veri yerine yeni bir değer atamak için önerilen yöntemler ele alınmıştır.

### 2.5.1. Kayıp veri mekanizmaları

Kayıp veri problemini gidermek için doğru atama yönteminin seçimini yapmadan önce, bir kayıp veri mekanizması belirlenmesi gerekmektedir. Little ve Rubin'in yaptıkları tanımlamalar doğrultusunda 3 farklı kayıp veri mekanizması bulunmaktadır (19).

#### 2.5.1.1. Tamamen rasgele olarak kayıp (Missing completely at random; MCAR)

Tamamen rasgele olarak kayıp veri mekanizması, rasgeleliğin en yüksek olduğu durumdur. Verilerin kayıp olma durumu gözlenen veya kayıp olan diğer verilerden bağımsızdır.

#### 2.5.1.2. Rasgele olarak kayıp (Missing at random;MAR)

Rasgele olarak kayıp veri mekanizması, MCAR'ın sınırlandırılmış farklı bir türüdür. Kayıp veri mekanizması kayıp verilere değil gözlenen verilere bağlıdır.

#### 2.5.1.3. Rasgele olmayan kayıp (Missing not at random; MNAR)

Diğer adı İhmal Edilemez (Non-Ignorable) Kayıp Veri mekanizmasıdır. Bu mekanizma kayıp verilere bağlıdır. Kayıp veri oluşumu araştırmacının birimleri ya da olayları ölçmemesine dayanır. Bu durum veri analizine oldukça zarar verebilir.

#### 2.5.1.4. Little'ın MCAR testi (Little's MCAR test)

Günümüzde kayıp değerli veri setleri için çok sayıda farklı atama yöntemleri geliştirilmiştir. Ama bunun yanında, veri setindeki kayıp veri mekanizmasını belirlemek amacıyla az sayıda çalışma yapılmıştır. Bu çalışmalar içerisinde en sık kullanılanı Little tarafından geliştirilen MCAR Testidir. Testin varsayımı, veri setinin MCAR yapıya sahip olduğudur. Bu varsayım altında Little, çok değişkenli kikare testi geliştirmiştir.

Bu testin gerçekleştirilmesi üç aşamada özetlenebilir:

1) EM (Expectation-Maximization) yaklaşımı kullanılarak tahmini ortalama ($\hat{\mu}$) ve varyans-kovaryans matrisleri ($\hat{\Sigma}$) belirlenir.
2) Kayıp örüntüsüne göre gözlemler k farklı yapı olarak gruplandırılır ve her grup için gözlemlerin ortalaması hesaplanır.

3) Gözlenen ve tahmin edilen ortalama değerlerinin farkı alınır. Bu fark, tahmin edilen varyans-kovaryans matrisi ve gruplardaki gözlem sayıları kullanılarak ağırlıklandırılır. Bunun sonucunda, yaklaşık olarak kikare dağılımına uyan bir istatistik elde edilir. Son olarak ise elde edilen istatistiğe göre varsayım test edilir.

Bu istatistik eşitlik 2.29 ile hesaplanır:

$$D^2 = \sum_{k=1}^{K} N^{(r_k)} \left(\bar{y}_{gözlenen.k} - \hat{\mu}_{gözlenen.k}\right)^T \hat{\Sigma}^{-1}_{gözlenen.k} \left(\bar{y}_{gözlenen.k} - \hat{\mu}_{gözlenen.k}\right) \quad (2.29)$$

Burada, $N^{(r_k)}$; k. örüntüdeki gözlenen örnek sayısını, $\bar{y}_{gözlenen.k}$; k. örüntüdeki yapının ortalama matrisi, $\hat{\mu}_{gözlenen.k}$; k. örüntüdeki yapının tahmini ortalamasıdır. $\hat{\Sigma}_{gözlenen.k}$ ise k. örüntüdeki yapının tahmini varyans-kovaryans matrisidir.

Bu kikare istatistiğinin serbestlik derecesi (sd); tüm yapılardaki gözlenen değişken sayısı toplamından ($p_k$), veri setinde yer alan değişken sayısının (p) çıkarılması ile elde edilir (18).

$$sd = \left(\sum_{k=1}^{K} p_k\right) - p \quad (2.30)$$

### 2.5.2. Kayıp veri analizinde kullanılan başlıca yöntemler

Kayıp veri analizi için kullanılan yaklaşımlar 3 madde altında toplanabilirler:

1. Kayıp verilere yeni değerler atanarak veri seti tamamlanabilir.
2. Kayıp veri mekanizması modellenebilir.
3. Kayıp verileri olan gözlemler veri setinden çıkartılabilir (6).

#### 2.5.2.1. Liste düzeyinde veri silme (Listwise deletion)

Bu yöntemde sadece tam (kayıp veri bulundurmayan) gözlemler kullanılır. Bir ya da daha fazla kayıp verisi olan gözlemler veri setinden çıkartılarak işlem yapılır. Eğer cevap (response) değişkeni kayıp veri içeriyorsa kullanılacak en mantıklı yöntem liste düzeyinde veri silmedir.

Eğer veri seti tamamen rasgele olarak kayıp veri içeriyorsa, bu yöntem uygulandıktan sonra istatistiksel güç daha düşük olacaktır. Eğer veri seti rasgele olmayan kayıp verilere sahip ise bu yöntemin uygulanmasının ardından elde edilen veri seti yanlı sonuçların oluşmasına neden olur (3, 5).

#### 2.5.2.2. Tekil atama yöntemleri (Single imputation)

Tekil atama yöntemleri, örneklemdeki bilgi kullanılarak kayıp verilerin yerine yeni değer atanması mantığı ile geliştirilmiştir. Bu yöntemlerden bazıları aşağıda verilmiştir.

***Ortalama Atama (Mean Imputation)***; en sık kullanılan yöntemlerden biridir. Kayıp verinin ait olduğu değişkenin kayıp olmayan verilerinin ortalaması alınır ve elde edilen sonuç kayıp veriye atanır. Bu durum sonucunda veri setinin ortalaması sabit kalırken, varyansı küçülür. Varyansa gerekli önemi vermemesi, veri setinin korelasyon yapısına negatif yönde yanlılık kazandırması ve yeni değerlerin dağılımının toplum değerlerinin dağılımını yansıtmaması bu atama yönteminin tercih edilmesini olumsuz yönde etkilemektedir (1, 3, 5, 6).

***Hot Deck Atama (Hot Deck Imputation);*** benzer veriler kullanılarak yapılan bir atama türüdür. Veri setindeki tüm gözlemler benzer

karakteristiklere göre gruplara bölünürler. Kayıp veri ataması yapılacak olan verinin yer aldığı gruptan rasgele bir gözlem seçilir. Seçilen bu gözlemdeki değer kayıp veriye atanır. Fazla sayıda alt gruplar oluşturmak atama sonucu yapılan tahminlerin geçerliliğini arttırır, ama bu durumda alt grup örneklem hacimlerinin küçük olmamasına dikkat edilmelidir (3, 5, 6).

***Cold Deck Atama (Cold Deck Imputation);*** dış kaynaktan, genellikle önceden yapılmış benzer çalışmalardan elde edilen sabit bir değerin kayıp veriye atanması ile gerçekleştirilen bir atama yöntemidir. Önceki yöntemlerden tek farkı, atanacak değerin kaynağının farklı bir veri seti olmasıdır. Ortalama atama yöntemi ile benzer olumsuz özelliklere sahiptir (3, 5, 6).

***Kayıp Gözlem ile Tam Gözlemin Yer Değiştirmesi (Case Substitution);*** örnekleme giren ama pek çok hatta tüm değerleri kayıp olan gözlem ile bu gözleme benzeyen ancak örnekleme hiç girmeyen başka bir gözlemin yer değiştirmesidir (3).

***EM (Expectation-Maximization) yaklaşımı;*** veri seti modelinin bilinmeyen parametreleri ile kayıp değerleri arasındaki ilişkiye bağlı çalışan bir en çok benzerlik (maximum likelihood) yöntemidir. Bu yöntem genellikle çok değişkenli normal model varsayımını kullanmaktadır. İki aşamadan oluşmaktadır. İlk aşamada kayıp değerli gözlemler için en iyi tahminler elde edilir. İkinci aşamada ise, atama gerçekleştirildikten sonra ortalama, standart sapma, korelasyon gibi değişkenlere ilişkin tahminlerde bulunulur. Bu iki aşama; elde edilen ardışık tahminler arasındaki fark önemli derece azalıncaya kadar tekrarlanır (3, 5, 6, 15).

***Regresyon atama (Regression Imputation);*** kayıp verilere, tam veriler üzerinden elde edilen bir regresyon modeli ile değer atama yöntemidir. Bu durumda kayıp verinin bulunduğu değişken bağımlı değişkeni, tam verilerin

olduğu değişkenler ise bağımsız değişkenleri oluşturmaktadır. Kayıp verinin (bağımlı değişkenin) türüne göre atama için kullanılacak regresyon modeli seçilir. İkili (binary) veri tipleri için probit ya da lojit modeller, kesikli tamsayı değerler için Poisson regresyon ve diğer sürekli tip veriler için Klasik En Küçük Kareler (OLS) Regresyon Modeli tercih edilebilir. Değişkenler arasındaki ilişkiyi ve bilgiyi kullanarak tahminlerde bulunsa da, bu yöntemin bazı olumsuz yönleri de bulunmaktadır. Atama sonucunda veri setinde zaten var olan ilişki daha da kuvvetlenmiş olur. Bu atama yönteminin aynı veri seti üstünde kullanımı birden fazla sayıda olursa, elde edilen yeni veri seti kendine özgü bir yapıya ulaşır ve genellenebilirliği azalır. Regresyon sonucu elde edilen tahminin sınırları veri setinde var olan sınırların ötesinde bulunabilir. Bu nedenle bazı düzeltme işlemleri yapılmalıdır (3).

Tekil atamaların en büyük kısıtı, tahmin edilen değerler sonucu veri setindeki değişkenliğini azaltması ve bunu önemsememesidir. Bu nedenle Çoklu Atama (Multiple Imputation), K En Yakın Komşu ile Kayıp Değer Atama (KNN Imputation) gibi daha farklı algoritmalar geliştirilmiştir.

#### 2.5.2.3. Çoklu atama (Multiple imputation; MI) yöntemi

Çoklu atama yönteminin temelini, kayıp veriye $D$ $(D \geq 2)$ defa atama yapma oluşturur. Kayıp verinin ait olduğu değişkene ait belirli bir modele bağlı dağılım belirlenir ve bu dağılımdan rasgele seçilen değerler ile atama yapılır. D tane tamamlanmış veri setinden elde edilen sonuçların birleştirilmesi sonucunda bir tahmin elde edilir.

Pek çok paket program birbirinden farklı çok değişkenli modeller kullanmaktadır. Çok değişkenli normal dağılım, loglinear modeller ve genel konum modelleri bunlara örnek olarak verilebilir. Ama gerçek hayat verilerini her zaman bir modele uyarlamak kolay olmamaktadır. Atama modeli

değişkenler arasındaki ilişkiyi yansıtacak bir yapıda olmalıdır. MI'ın pek çok uygulamasında, en azından yaklaşık olarak doğru olacak biçimde atamalar yapılmaktadır. Ayrıca gerçek veri seti ile atanmış veri setini karşılaştırabilecek bir atama modelinin seçilmesi gerekmektedir (3, 5, 14).

#### 2.5.2.4. K En Yakın Komşu (KNN) ile Kayıp değer atama yöntemi

K en yakın komşu (KNN) ile kayıp değer atama yönteminin temelini KNN algoritması oluşturmaktadır. Bu algoritma; bir örneklemde yer alan gözlemlerin her birinin belirli bir gözlem değerine göre uzaklıklarının hesaplanması ve elde edilen en küçük k tane gözlemin seçilmesi ile elde edilir.

KNN algoritmasında uzaklık hesaplamalarında kullanılmak üzere birbirinden farklı fonksiyonlar belirlenmiştir. Bunlara örnek olarak Öklid (Euclidean), Manhattan ve Minkowski Uzaklık Fonksiyonları gösterilebilir. Bu uzaklık fonksiyonları içerisinde kullanımı en yaygın olan Öklid Uzaklık Fonksiyonudur. Bu fonksiyon ile p boyutlu bir uzayda i ve j noktaları arasındaki uzaklık şu şekilde elde edilir:

$$d(i,j) = \sqrt{\sum_{k=1}^{p}\left(x_{ik} - x_{jk}\right)^2} \tag{2.31}$$

Veri setinin ikiden fazla sayıda değişken içerdiği durumlarda Standardize Edilmiş Öklid Uzaklık Fonksiyonu kullanılır. Her değişken kendi içerisinde z dönüşümü uygulanarak standardize edilir ve eşitlik 2.31'e yerleştirilir. Böylece değişkenler arasındaki ölçüm farklılıkları ortadan kalkmış olur.

Bu algoritma kurulurken k değeri seçimine dikkat edilmelidir. Komşuluk değeri k'nın küçük bir sayı olması durumunda baskın gözlemlerin aşırı vurgulanmasına bağlı olarak atama yönteminin tahmin performansında bir

bozulma görülebilir. Diğer yandan, büyük bir k komşuluk değeri, kayıp değerli gözlemden oldukça farklı gözlemlerin tahmin sürecinde yer almasına ve bunun sonucunda tutarsız değerler elde edilmesine neden olabilir.

Son zamanlarda KNN algoritması kullanılarak elde edilen yeni bir yöntem kayıp veri problemlerinin çözümünde oldukça etkin olmuştur. İlk olarak 1979 yılında Dixon, sınıflandırma problemlerinde kayıp veri sorunu için bu tekniğin kullanımına değinmiştir. Daha sonra ise mikrodizilim veri setlerinde ve yapay sinir ağları, veri madenciliği gibi dallarda kayıp veri probleminin çözümlenmesinde KNN Kayıp değer Atama yöntemi kullanılmıştır (3).

KNN ile kayıp değer atama yöntemi, ilgilenilen kayıp verili gözleme benzer diğer gözlem değerleri kullanılarak yapılır. Bir veri matrisinde g satırlardaki gözlemi göstersin. Bu gözlem değerine ait k en yakın komşuları belirlenmek istendiğinde, g gözlem değerine ait kayıp verilerin olduğu sütunlarda, komşu gözlemler mutlaka veri bulundurmalıdır. Uzaklık hesabının yapılacağı diğer gözlemde kayıp değer bulunuyor ise, iki gözlemin kayıp değer bulundurmadığı ortak sütunlar kullanılarak uzaklık hesabı yapılır. Bu hesaplamalar sonucunda en küçük uzaklığa sahip k tane gözlem komşu olarak seçilir. Daha sonra, g gözlemindeki kayıp verinin olduğu sütuna değer atamak için, komşu gözlemlerin o sütundaki değerlerinin ağırlıklı ortalaması kullanılır. Bu ağırlıklar Öklid uzaklık değerlerinin tersi alınarak elde edilir (25).

Literatürde Acuna ve Rodriguez farklı bir atama algoritması daha geliştirmiştir. Bu algoritmada, uzaklık fonksiyonunun hesaplandığı gözlem satırlarının kayıp veri içermesine izin verilmemektedir. Bu kayıp değer atama algoritması şu şekildedir:

1. D veri seti $D_m$ ve $D_c$ olmak üzere iki parçaya ayrılır. $D_m$; en az bir değişkeni kayıp veri içeren gözlemler seti olsun. $D_c$ ise tüm değişkenlerin eksiksiz olduğu geriye kalan veri olsun.
2. $D_m$ 'deki her $g$ gözlem vektörünü gözlenen ($g_o$) ve eksik bölümler ($g_m$) olarak ikiye ayrılır.

$$g = [g_o; g_m]$$

3. $g_o$ ile $D_c$ veri setinde yer alan gözlem vektörleri arasındaki uzaklık hesaplanır. Bu uzaklık hesaplamalarında $D_c$'nin $x_o$ ile kesişiminde yer alan değişkenler kullanılır.
4. Uzaklık hesaplamaları elde edildikten sonra kayıp verili gözleme en yakın olan k tane komşu gözlem belirlenir.
5. Kayıp veri kategorik bir değişken ise en çok oyu alan kategori kayıp veriye atanarak tahmin edilmiş olur. Eğer sürekli bir değişken ise k tane değerin ortalaması elde edilir (1).

Bu tez çalışmasında, R paket programında yer alan "imputation" paketindeki "knnImpute" fonksiyonu kullanılarak değerlendirmeler yapılmıştır. Bu fonksiyon Öklid uzaklık değerlerinin tersini almak yerine, onları aşağıdaki şekilde normalleştirir. En küçük k uzaklık değerlerini en büyük uzaklık değerine böler. Daha sonra ise herbirini birden çıkartarak yeni değerler elde eder. Bu değerleri kullanarak ağırlıklı ortalama hesaplar ve atanacak değer belirlenmiş olur.

Bu kayıp değer atama yönteminin avantajları aşağıdaki gibi sıralanabilir.
1. K en yakın komşu algoritması ile kategorik (k en yakın komşular arasında en sık gözlenen değer) ve sürekli (k en yakın komşuların ağırlıklı ortalaması) değişkenler için tahminler yapılabilir.
2. Kayıp değerli her değişken için bir tahmin modeli kurmaya gerek duyulmamaktadır.
3. Çok sayıda kayıp değerli gözlemlere oldukça kolay bir şekilde müdahale eder.

4. Veri setinin korelasyon yapısını dikkate almaktadır.

Kısıtları ise şu şekilde ifade edilebilir:

1. Uzaklık fonksiyonu seçiminde sorunlar yaşanabilir. Öklid uzaklık fonksiyonu her zaman uygun olmayabilir.
2. Komşu sayısı olan k değerinin seçimi sonuçları önemli derecede etkileyebilir. Bu nedenle dikkatle belirlenmelidir (1).

# 3. GEREÇ VE YÖNTEMLER

Bu tez çalışmasında Random Forests (RF) yönteminin kayıp değer atama yöntemiyle, k en yakın komşu algoritması (KNN) ile kayıp değer atama yöntemi karşılaştırılmıştır. RF yönteminin sınıflandırma algoritması kullanılarak sonuçlar elde edilmiştir. Karşılaştırmalar iki farklı aşamada yapılmıştır. İlk aşamada benzetim yolu ile türetilen veri setleri kullanılmıştır. İkinci aşamada ise sağlık alanına ait kayıp değerli bir veri setinden yararlanılmıştır. Her iki aşamada da da R 3.0.1 istatistik paket programı (The R Project for Statistical Computing) kullanılmıştır.

## 3.1. Benzetim Çalışmaları ve Veri Türetimi

(100000/n) Monte Carlo benzetim tekniği kullanılarak örneklem hacimleri (n) ve tekrar sayıları (s) belirlenmiştir. Birbirinden farklı 4 adet benzetim çalışmasında örneklem hacimleri 100, 200, 500, 1000 ve bunlara karşılık gelen tekrar sayıları ise sırasıyla 1000, 500, 200, 100 olarak belirlenmiştir. Veri türetiminde R programının MASS paketi kullanılmıştır.

Her benzetim çalışmasında tekrar sayısı kadar birbirinden bağımsız veri setleri türetilmiştir. Tüm veri setleri 1 tane cevap (response) ve 30 tane sürekli tahminci değişken olacak biçimde oluşturulmuştur. Cevap değişkeni (y) iki kategorili olarak belirlenmiştir. Tahminci değişkenler ise farklı korelasyon matrisleri kullanılarak türetilmiştir.

Tahminci değişkenlerin veri türetimi iki farklı biçimde yapılmıştır. 30 sürekli değişkenin ilk 5 tanesi (X) her birinin ortalaması 0 olacak biçimde düşük, orta ve yüksek derecede ilişkili korelasyon matrislerine ($R_i$) göre çok değişkenli standart normal dağılımdan türetilmiştir. Geriye kalan 25 değişken ise birbirinden bağımsız

herbiri sıfır ortalamalı olacak biçimde çok değişkenli standart normal dağılımdan elde edilmiştir. Birbirlerinden bağımsız olmaları için korelasyon matrisi olarak birim matris tercih edilmiştir (21).

İlk 5 değişken ($x_j$, j=1,2,3,4,5) için korelasyon matrisleri; düşük ($R_1$), orta ($R_2$) ve yüksek ($R_3$) derecede ilişkili olmak üzere aşağıdaki gibi belirlenmiştir:

$$R_1 = \begin{bmatrix} 1 & 0.1 & 0.1 & 0.1 & 0.1 \\ 0.1 & 1 & 0.1 & 0.1 & 0.1 \\ 0.1 & 0.1 & 1 & 0.1 & 0.1 \\ 0.1 & 0.1 & 0.1 & 1 & 0.1 \\ 0.1 & 0.1 & 0.1 & 0.1 & 1 \end{bmatrix} \quad R_2 = \begin{bmatrix} 1 & 0.5 & 0.5 & 0.5 & 0.5 \\ 0.5 & 1 & 0.5 & 0.5 & 0.5 \\ 0.5 & 0.5 & 1 & 0.5 & 0.5 \\ 0.5 & 0.5 & 0.5 & 1 & 0.5 \\ 0.5 & 0.5 & 0.5 & 0.5 & 1 \end{bmatrix} \quad R_3 = \begin{bmatrix} 1 & 0.9 & 0.9 & 0.9 & 0.9 \\ 0.9 & 1 & 0.9 & 0.9 & 0.9 \\ 0.9 & 0.9 & 1 & 0.9 & 0.9 \\ 0.9 & 0.9 & 0.9 & 1 & 0.9 \\ 0.9 & 0.9 & 0.9 & 0.9 & 1 \end{bmatrix}$$

Cevap (sınıf) değişkeninin üretilmesi için lojistik regresyon fonksiyonundan faydalanılmıştır.

$$P(y = 1|X) = \pi(X) = \frac{exp(X\beta)}{1+exp(X\beta)} \tag{3.1}$$

Lojistik regresyonda bağımsız değişkenler olarak $R_i$ korelasyon matrisli ve sıfır ortalamalı çok değişkenli standart normal dağılıma uyan $x_j$, j=1,2,3,4,5 tahminci değişkenleri kullanılmıştır. $\beta_0 = 0$ ve değişkenler için ise $\beta = [1\ 2\ 3\ 4\ 5]^T$ olacak biçimde $\beta$ katsayıları seçilmiştir. Hesaplanan $\pi(X)$ değerleri Bernoulli kümülatif dağılım fonksiyonunun tersine olasılık olarak aktarılmıştır. Bunun sonucunda ise 0 ve 1 olarak Bernoulli sonuçları gözlemlenmiş ve sınıf değeri olarak atanmıştır.

Özetle, benzetim çalışmalarında veri türetimi iki parametre üzerine kurulu olarak gerçekleştirilmiştir. Bunlar örneklem hacmi (n) ve korelasyon matrsinde köşegen dışında yer alan değerlerdir (r) (Bkz. EK-1).

Benzetim çalışmalarının her tekrar aşamasında öncelikle bir tam veri seti türetilmiştir. Daha sonra bu veri setinde belirli yüzdeliklere göre kayıp değerler

oluşturulmuştur. Tam veri setinde kayıp değer oluşturmak için iadesiz rasgele örnekleme yöntemi kullanılmıştır. Bu yöntem ile seçilen değerlere kayıp değeri gösteren NA ifadesi atanmıştır. Rasgele örnekleme ile kayıp veri oluşturma değişken (sütun) bazında yapılmıştır. Veri seti türetiminde kullanılan ana değişkenlerden X1 ve X5'e, aynı anda ve aynı yüzdelik değerler kullanılarak rasgele kayıp değerler atanmıştır. Kullanılacak olan kayıp değer yüzdelikleri 5, 10, 15, 20, 25 olarak belirlenmiştir.

Kayıp değerli veri setleri 2 farklı biçimde atanmıştır. İlk atama yöntemi RF'nin uzaklık matrisini kullanarak gerçekleştirdiği kayıp değer atama algoritması ile yapılmıştır. Bu atama işleminde iterasyon sayısı 5 olarak belirlenmiştir. Her bir iterasyon için 500 ağaçlık karar ormanı kurulmuştur. Beşinci iterasyon sonunda elde edilen veri seti, bu yöntem ile tamamlanmış veri seti olarak kullanılmıştır. Bu atama işlemi için R programının randomForest paketinde yer alan rfImpute fonksiyonu kullanılmıştır. Diğer atama işlemi ise KNN ile kayıp değer atama yöntemiyle yapılmıştır. Bu atama işlemi için ise R programının Imputation paketinde yer alan knnImpute fonksiyonu kullanılmıştır. Benzetim çalışmasında kullanılacak k komşuluk değerleri 5, 10, 15 ve 20 olarak belirlenmiştir. Bunun sonucunda KNN algoritması ile 4 farklı biçimde tamamlanmış veri seti elde edilmiştir. Yapılan bu işlemlerinin ardından aynı kayıp değerli veri seti kullanılarak beş farklı atanmış veri seti elde edilmiştir.

Karşılaştırmaların aynı koşulda olması için RF algoritmaları tüm veri setlerinde aynı özelliklere göre kurulmuştur. Bunun için R programının randomForest paketinde yer alan randomForest fonksiyonu kullanılmıştır. Kurulan her RF algoritması 500 ağaçtan oluşmaktadır. Breiman'a göre 500 adet ağaçtan oluşan bir karar ormanı oldukça tutarlı sonuçlar vermektedir (8). RF'nin sınıflandırma algoritması kullanıldığı için her ayrımda rasgele seçilecek düğüm sayısı $m = \sqrt{p}$ olarak belirlenmiştir.

İlk türetilen tam veri seti ve atanmış veri setlerinin herbirine aynı RF algoritması uygulanarak 6 adet sınıflandırma matrisi (confusion matrix) gözlemlenmiştir. Bu matris OOB veri setlerine bağlı olarak hesaplanmaktadır.

Tablo 3.1. Sınıflandırma matrisi (confusion matrix)

| | | Tahmin Edilen Sınıf Değerleri | | |
|---|---|---|---|---|
| | | 0 | 1 | Toplam |
| Gerçek Sınıf Değerleri | 0 | a | b | a+b |
| | 1 | c | d | c+d |
| Toplam | | a+c | b+d | n |

Yukarıdaki matriste görüldüğü üzere doğru sınıflandırma oranı (DSO), köşegen üzerindeki değerler toplamının genel toplama oranı ile elde edilir.

$$\mathrm{DSO} = \frac{a+d}{n} \tag{3.2}$$

Böylece her bir tekrarda 5 farklı DSO elde edilmektedir. Her benzetim çalışmasında 5 farklı atama algoritması sonucu için tekrar sayısı kadar DSO sonucu elde edilmiştir. Hesaplanan DSO'ların ortalamaları alınarak birbirleriyle karşılaştırılmıştır (Bkz. EK-2).

### 3.2. Sağlık Alanında Bir Uygulama

Eskişehir Osmangazi Üniversitesi Tıp Fakültesi Hastanesi Endokrinoloji Servis ve Polikliniği'ne 1 Haziran 2011 ve 1 Haziran 2013 tarihleri aralığında başvuran kırk yaş üstü tip 2 diyabetli hastalarda, periferik arter hastalığı varlığı araştırmasına ilişkin veriler toplanmıştır. Belirlenen süre boyunca koşulları sağlayan 110 kişinin Doppler ultrason sonuçlarına bakılarak uzman görüşüyle hasta olup olmadığına karar

verilmiştir. Doppler ultrason sonuçlara göre 25 kişinin periferik arter hastalığına sahip olduğu ve 85 kişinin ise olmadığı belirlenmiştir. Doppler ultrasonunun dışında bir takım biyokimyasal değer ölçümleri ve oran hesapları da incelemeye alınmıştır. Bu tez çalışmasında; periferik arter hastalığı araştırılması sürecinde toplanan sürekli değişkenlerden uygulama amacıyla faydalanmak için çalışma sahiplerinden izin alınmıştır.

Benzetim çalışmalarında kullanılan kayıp değer atama yöntemleri, periferik arter hastalığı verileri kullanılarak da karşılaştırılmıştır. Periferik arter hastalığı ile ilişkilendirilen sürekli değişkenler aşağıda verilmiştir:

- Yaş
- Diyabet süresi
- SolABI1
- SağABI1
- SolABI2
- SağABI2
- Ürikasit
- Homosistein
- Fosfor
- LDL
- TG
- HDL
- Albuminuri
- GFR
- Otuzonbeşoranı
- HbA1c
- HOMA
- Vücut Kitle İndeksi

Bu veri setinde homosistein ve otuzonbeş oranı değişkenlerinde kayıp değerler gözlemlenmiştir. Atama işlemlerinin gerçekleştirilebilmesi için bu veri setinin kayıp veri mekanizması incelenmiştir. Little'ın Tamamen Rasgele Olarak Kayıp testi (Little's MCAR test) ile kayıp değerlerin MCAR bir yapı sergileyip sergilemediği araştırılmıştır. Bunun için R programında yer alan BayloEdPsych paketindeki LittleMCAR fonksiyonu kullanılmıştır. Test sonucuna göre MCAR yapı sergileyen veri setine; RF'nin uzaklık matrisi yöntemiyle ve farklı k değerleri için KNN ile kayıp değer atama yöntemi kullanılarak atamalar yapılıp sonuçlar elde edilmiştir. DSO değerleri kullanılarak karşılaştırmalar yapılmıştır. Elde edilen sonuçlar, benzetim çalışması sonuçları ile ilişkilendirilerek yorumlanmıştır.

# 4. BULGULAR

## 4.1 Benzetim Çalışması Bulguları

Gereç ve yöntemler başlığında anlatılan benzetim çalışmalarına ilişkin sonuçlar aşağıdaki tablolarda yer almaktadır. Satırlarda, değişkenlerde oluşturulan kayıp değer yüzdeleri verilmektedir. Sütunlarda ise tam veri seti ve bu veri setinin iki değişkeninden belirtilen yüzdelere göre eksiltilmiş halini tamamlayan atama yöntemleri belirtilmektedir. Hücrelerde DSO değerleri verilmektedir. Her tabloda tam veri setine ilişkin verilen DSO sonucu, üzerinde herhangi bir eksiltme ve atama işlemi yapılmadığından sabit bir değer olarak gösterilmektedir.

Örneklem hacminin 100 ve tekrar sayısının 1000 olduğu benzetim çalışmasında farklı korelasyon matrislerine göre elde edilen sonuçlar Tablo 4.1, Tablo 4.2 ve Tablo 4.3'de verilmiştir.

Tablo 4.1. $R_1$ korelasyon matrisine göre türetilen n=100 ve s=1000 koşulu ile gerçekleştirilen benzetim çalışması sonuçları

| Veri Yapısı | Tam veri seti DSO | RF Uzaklık Matrisi DSO | KNN ile kayıp değer atama | | | |
|---|---|---|---|---|---|---|
| | | | K=5 DSO | K=10 DSO | K=15 DSO | K=20 DSO |
| N=100 | **0.82738** | - | - | - | - | - |
| %5 Eksik | - | 0.81771 | 0.81757 | 0.81811 | **0.81843** | 0.81840 |
| %10 Eksik | - | 0.81096 | 0.81082 | 0.81043 | 0.81118 | **0.81183** |
| %15 Eksik | - | 0.80173 | 0.80259 | 0.80411 | **0.80446** | 0.80444 |
| %20 Eksik | - | 0.79225 | 0.79330 | 0.79483 | 0.79399 | **0.79488** |
| %25 Eksik | - | 0.78271 | 0.78492 | 0.78508 | 0.78687 | **0.78700** |

Tablo 4.1'de yer alan n=100 birimlik ve düşük korelasyonlu türetilen veri yapılarında tam veri seti DSO'ı olan %82.74 değerine en yakın tahmin %5'lik kayıp değerli veri setlerindeki % 81.843'lük tahmin oranı ile k=15 komşuluk değerindeki KNN atama yöntemi ile elde edilirken, en uzak tahmin %25'lik kayıp değerli veri setlerindeki %78.27'lik oran ile uzaklık matrisi yöntemine aittir.

Tablo 4.1 incelendiğinde, %5'lik eksik veri setlerinde %98.918'lik doğruluk payı ile k=15 komşuluk değerindeki KNN ile kayıp değer atama yönteminin, %10'luk eksik veri setlerinde %98.121'lik doğruluk payı ile k=20 komşuluk değerindeki KNN ile kayıp değer atama yönteminin, %15'lik eksik veri setlerinde %97.23'lük doğruluk payı ile k=15 komşuluk değerindeki KNN ile atama yönteminin, %20'lik ve %25'lik eksik veri setlerinde ise sırasıyla %96.072'lik ve %95.120'lik doğruluk payı ile k=20 komşuluk değerindeki KNN ile atama yönteminin en iyi sonuçları verdiği görülmektedir.

Tablo 4.2. $R_2$ korelasyon matrisine göre türetilen n=100 ve s=1000 koşulu ile gerçekleştirilen benzetim çalışması sonuçları

| Veri Yapısı | Tam veri seti DSO | RF Uzaklık Matrisi DSO | KNN ile kayıp değer atama | | | |
|---|---|---|---|---|---|---|
| | | | K=5 DSO | K=10 DSO | K=15 DSO | K=20 DSO |
| N=100 | **0.90723** | - | - | - | - | - |
| %5 Eksik | - | 0.90273 | 0.90276 | 0.90262 | **0.90332** | 0.90261 |
| %10 Eksik | - | 0.89773 | 0.89730 | 0.89855 | 0.89908 | **0.89947** |
| %15 Eksik | - | 0.89395 | 0.89470 | 0.89601 | **0.89682** | 0.89622 |
| %20 Eksik | - | 0.88936 | 0.88902 | 0.89064 | 0.89159 | **0.89175** |
| %25 Eksik | - | 0.88325 | 0.88443 | 0.88576 | **0.88780** | 0.88706 |

Tablo 4.2'de yer alan n=100 birimlik ve orta korelasyonlu türetilen veri yapılarında tam veri seti DSO'ı olan %90.72 değerine en yakın tahmin %5'lik kayıp değerli veri setlerindeki % 90.33'lük tahmin oranı ile k=15 komşuluk değerindeki KNN atama yöntemi ile elde edilirken, en uzak tahmin %25'lik kayıp değerli veri setlerindeki %88.33'lük oran ile uzaklık matrisi yöntemine aittir.

Tablo 4.2 incelendiğinde, %5'lik eksik veri setlerinde %99.569'luk doğruluk payı ile k=15 komşuluk değerindeki KNN ile kayıp değer atama yönteminin, %10'luk eksik veri setlerinde %99.145'lik doğruluk payı ile k=20 komşuluk değerindeki KNN ile kayıp değer atama yönteminin, %15'lik eksik veri setlerinde %98.853'lük doğruluk payı ile k=15 komşuluk değerindeki KNN ile atama yönteminin, %20'lik eksik veri setlerinde %98.294'lük doğruluk payı ile k=20 komşuluk değerindeki KNN ile atama yönteminin, %25'lik eksik veri setlerinde ise

%97.858'lik doğruluk payı ile k=15 komşuluk değerindeki KNN ile atama yönteminin en iyi sonuçları verdiği görülmektedir.

Tablo 4.3. $R_3$ korelasyon matrisine göre türetilen n=100 ve s=1000 koşulu ile gerçekleştirilen benzetim çalışması sonuçları

| Veri Yapısı | Tam veri seti DSO | RF Uzaklık Matrisi DSO | KNN ile kayıp değer atama | | | |
|---|---|---|---|---|---|---|
| | | | K=5 DSO | K=10 DSO | K=15 DSO | K=20 DSO |
| N=100 | **0.94761** | - | - | - | - | - |
| %5 Eksik | - | 0.94543 | 0.94599 | 0.94615 | **0.94641** | 0.94616 |
| %10 Eksik | - | 0.94407 | 0.94367 | 0.94395 | 0.94445 | **0.94481** |
| %15 Eksik | - | 0.94329 | 0.94229 | 0.94300 | 0.94319 | **0.94375** |
| %20 Eksik | - | 0.94205 | 0.94134 | 0.94110 | 0.94189 | **0.94284** |
| %25 Eksik | - | 0.94035 | 0.93942 | 0.94055 | **0.94132** | 0.94088 |

Tablo 4.3'de yer alan n=100 birimlik ve yüksek korelasyonlu türetilen veri yapılarında tam veri seti DSO'ı olan %94.76 değerine en yakın tahmin %5'lik kayıp değerli veri setlerindeki % 94.64'lük tahmin oranı ile k=15 komşuluk değerindeki KNN atama yöntemi ile elde edilirken, en uzak tahmin %25'lik kayıp değerli veri setlerindeki %93.94'lük oran ile k=5 komşuluk değerindeki KNN atama yöntemine aittir.

Tablo 4.3 incelendiğinde, %5'lik eksik veri setlerinde %99.873'lük doğruluk payı ile k=15 komşuluk değerindeki KNN ile kayıp değer atama yönteminin, %10'luk, %15'lik ve %20'lik eksik veri setlerinde sırasıyla %99.705'lik, %99.593'lük ve %99.497'lik doğruluk payı ile k=20 komşuluk değerindeki KNN ile atama yönteminin, %25'lik eksik veri setlerinde ise %99.336'lık doğruluk payı ile k=15 komşuluk değerindeki KNN ile atama yönteminin en iyi sonuçları verdiği görülmektedir.

Tablo 4.1-3'de kayıp değer yüzdeleri arttıkça, tüm yöntemler için DSO sonuçlarında bir düşüş görülmektedir. Korelasyon matrisindeki değerler bir başka ifadeyle değişkenler arasındaki ilişki arttıkça, DSO sonuçlarında da artış gözlemlenmektedir. Tablolardaki en yüksek DSO değerleri KNN ile kayıp değer

atama yöntemine aittir. Tüm yöntemler birbirlerine oldukça yakın sonuçlar sergilemektedir. Tam veri setlerine en yakın tahmin %0.12'lik hata ile yüksek korelasyonlu veri yapılarında, en uzak tahmin ise %4.47'lik hata ile düşük korelasyonlu veri yapılarında gözlemlenmektedir. Yüksek derecede ilişkili korelasyon matrisi ile türetilen veri setlerinde uzaklık matrisi ile atama yöntemi, k=5 olan KNN ile kayıp değer atama yönteminden daha iyi sonuçlar göstermektedir. Benzetim çalışmaları sonucunda k=15 ve k=20 değerlerinin kullanıldığı KNN ile kayıp değer atama yöntemi, uzaklık matrisi yöntemine göre daha iyi sonuçlar vermektedir.

Örneklem hacminin 200 ve tekrar sayısının 500 olduğu benzetim çalışmasında farklı korelasyon matrislerine göre elde edilen sonuçlar Tablo 4.4, Tablo 4.5 ve Tablo 4.6'da verilmiştir.

Tablo 4.4. $R_1$ korelasyon matrisine göre türetilen n=200 ve s=500 koşulu ile gerçekleştirilen benzetim çalışması sonuçları

| Veri Yapısı | Tam veri seti DSO | RF Uzaklık Matrisi DSO | KNN ile kayıp değer atama | | | |
|---|---|---|---|---|---|---|
| | | | K=5 DSO | K=10 DSO | K=15 DSO | K=20 DSO |
| N=200 | **0.86116** | - | - | - | - | - |
| %5 Eksik | - | 0.85314 | 0.85342 | **0.85376** | 0.85363 | 0.85262 |
| %10 Eksik | - | 0.84550 | 0.84454 | 0.84643 | **0.84731** | 0.84629 |
| %15 Eksik | - | 0.83576 | 0.83790 | 0.83886 | 0.83898 | **0.83917** |
| %20 Eksik | - | 0.82754 | 0.82842 | 0.83029 | **0.83251** | 0.83148 |
| %25 Eksik | - | 0.82082 | 0.82314 | 0.82478 | **0.82586** | 0.82584 |

Tablo 4.4'de yer alan n=200 birimlik ve düşük korelasyonlu türetilen veri yapılarında tam veri seti DSO'ı olan %86.12 değerine en yakın tahmin %5'lik kayıp değerli veri setlerindeki % 85.38'lik tahmin oranı ile k=10 komşuluk değerindeki KNN atama yöntemi ile elde edilirken, en uzak tahmin %25'lik kayıp değerli veri setlerindeki %82.08'lik oran ile uzaklık matrisi yöntemine aittir.

Tablo 4.4 incelendiğinde, %5'lik eksik veri setlerinde %99.141'lik doğruluk payı ile k=10 komşuluk değerindeki KNN ile kayıp değer atama yönteminin,

%10'luk eksik veri setlerinde %98.392'lik doğruluk payı ile k=15 komşuluk değerindeki KNN ile kayıp değer atama yönteminin, %15'lik eksik veri setlerinde %97.446'lık doğruluk payı ile k=20 komşuluk değerindeki KNN ile atama yönteminin, %20'lik ve %25'lik eksik veri setlerinde ise sırasıyla %96.673'lük ve %95.901'lik doğruluk payı ile k=15 komşuluk değerindeki KNN ile atama yönteminin en iyi sonuçları verdiği görülmektedir.

Tablo 4.5. $R_2$ korelasyon matrisine göre türetilen n=200 ve s=500 koşulu ile gerçekleştirilen benzetim çalışması sonuçları

| Veri Yapısı | Tam veri seti DSO | RF Uzaklık Matrisi DSO | KNN ile kayıp değer atama | | | |
|---|---|---|---|---|---|---|
| | | | K=5 DSO | K=10 DSO | K=15 DSO | K=20 DSO |
| N=200 | **0.92070** | - | - | - | - | - |
| %5 Eksik | - | 0.91632 | 0.91582 | 0.91683 | **0.91687** | 0.91600 |
| %10 Eksik | - | 0.91221 | 0.91172 | **0.91264** | 0.91178 | 0.91237 |
| %15 Eksik | - | 0.90914 | 0.90864 | 0.90924 | **0.91000** | 0.90993 |
| %20 Eksik | - | 0.90376 | 0.90224 | 0.90364 | 0.90489 | **0.90507** |
| %25 Eksik | - | 0.89934 | 0.89762 | 0.90042 | 0.90039 | **0.90100** |

Tablo 4.5'de yer alan n=200 birimlik ve orta korelasyonlu türetilen veri yapılarında tam veri seti DSO'ı olan %92.07 değerine en yakın tahmin %5'lik kayıp değerli veri setlerindeki % 91.69'luk tahmin oranı ile k=15 komşuluk değerindeki KNN atama yöntemi ile elde edilirken, en uzak tahmin %25'lik kayıp değerli veri setlerindeki %89.76'lık oran ile k=5 komşuluk değerindeki KNN atama yöntemine aittir.

Tablo 4.5 incelendiğinde, %5'lik eksik veri setlerinde %99.584'lük doğruluk payı ile k=15 komşuluk değerindeki KNN ile kayıp değer atama yönteminin, %10'luk eksik veri setlerinde %99.125'lik doğruluk payı ile k=10 komşuluk değerindeki KNN ile kayıp değer atama yönteminin, %15'lik eksik veri setlerinde %98.838'lik doğruluk payı ile k=15 komşuluk değerindeki KNN ile atama yönteminin, %20'lik ve %25'lik eksik veri setlerinde ise sırasıyla %98.302'lik ve %97.860'lık doğruluk payı ile k=20 komşuluk değerindeki KNN ile atama yönteminin en iyi sonuçları verdiği görülmektedir.

Tablo 4.6. $R_3$ korelasyon matrisine göre türetilen n=200 ve s=500 koşulu ile gerçekleştirilen benzetim çalışması sonuçları

| Veri Yapısı | Tam veri seti DSO | RF Uzaklık Matrisi DSO | KNN ile kayıp değer atama | | | |
|---|---|---|---|---|---|---|
| | | | K=5 DSO | K=10 DSO | K=15 DSO | K=20 DSO |
| N=200 | **0.95111** | - | - | - | - | - |
| %5 Eksik | - | 0.94958 | 0.94903 | 0.94975 | 0.94962 | **0.94982** |
| %10 Eksik | - | **0.94820** | 0.94761 | 0.94764 | 0.94784 | 0.94800 |
| %15 Eksik | - | **0.94771** | 0.94608 | 0.94602 | 0.94683 | 0.94716 |
| %20 Eksik | - | **0.94624** | 0.94453 | 0.94484 | 0.94559 | 0.94563 |
| %25 Eksik | - | **0.94543** | 0.94319 | 0.94370 | 0.94389 | 0.94469 |

Tablo 4.6'da yer alan n=200 birimlik ve yüksek korelasyonlu türetilen veri yapılarında tam veri seti DSO'ı olan %95.11 değerine en yakın tahmin %5'lik kayıp değerli veri setlerindeki % 94.98'lik tahmin oranı ile k=20 komşuluk değerindeki KNN atama yöntemi ile elde edilirken, en uzak tahmin %25'lik kayıp değerli veri setlerindeki %94.32'lik oran ile k=5 komşuluk değerindeki KNN atama yöntemine aittir.

Tablo 4.6 incelendiğinde, %5'lik eksik veri setlerinde %99.864'lük doğruluk payı ile k=20 komşuluk değerindeki KNN ile kayıp değer atama yönteminin, %10'luk, %15'lik, %20'lik ve %25'lik eksik veri setlerinde sırasıyla %99.694'lük, %99.643'lük, %99.488'lik ve %99.403'lük doğruluk payı ile RF'nin uzaklık matrisi ile kayıp değer atama yönteminin en iyi sonuçları verdiği görülmektedir.

Tablo 4.4-6'da da n=100 birimlik benzetim çalışmalarına benzer sonuçlar gözlemlenmektedir. Kayıp değer yüzdeleri arttıkça, tüm yöntemler için DSO sonuçlarında bir düşüş görülmektedir. Değişkenler arasındaki ilişki arttıkça, DSO sonuçlarında da artış gözlemlenmektedir. Tüm yöntemler birbirlerine yakın sonuçlar sergilemektedir. Tam veri setlerine en yakın tahmin %0.13'lük hata ile yüksek korelasyonlu veri yapılarında, en uzak tahmin ise %4.03'lük hata ile düşük korelasyonlu veri yapılarında gözlemlenmektedir. Düşük ve orta derecede ilişkili korelasyon matrisi ile türetilen veri setlerinde k=10, k=15 ve k=20 değerleri kullanılarak yapılan KNN ile kayıp değer atama yöntemi, uzaklık matrisi yöntemine

göre daha iyi sonuçlar vermektedir. Orta derecede ilişkili korelasyon matrisi ile türetilen veri setlerinde, uzaklık matrisi ile atama yöntemi, k=5 olan KNN ile kayıp değer atama yönteminden daha iyi sonuçlar vermektedir. Yüksek derecede ilişkili korelasyon matrisi ile türetilen veri setlerinde ise, değişkenlerdeki kayıp değer sayısının %5'den daha fazla olduğu durumlarda, uzaklık matrisi ile atama yöntemi farklı komşuluk değerlerindeki KNN ile kayıp değer atama yöntemlerinden daha iyi DSO sonuçları ortaya koymaktadır.

Örneklem hacminin 500 ve tekrar sayısının 200 olduğu benzetim çalışmasında ise farklı korelasyon matrislerine göre elde edilen sonuçlar Tablo 4.7, Tablo 4.8 ve Tablo 4.9'da verilmiştir.

Tablo 4.7. $R_1$ korelasyon matrisine göre türetilen n=500 ve s=200 koşulu ile gerçekleştirilen benzetim çalışması sonuçları

| Veri Yapısı | Tam veri seti DSO | RF Uzaklık Matrisi DSO | KNN ile kayıp değer atama | | | |
|---|---|---|---|---|---|---|
| | | | K=5 DSO | K=10 DSO | K=15 DSO | K=20 DSO |
| N=500 | **0.88748** | - | - | - | - | - |
| %5 Eksik | - | 0.87778 | 0.87935 | 0.88008 | **0.88020** | 0.88019 |
| %10 Eksik | - | 0.87133 | 0.87190 | 0.87251 | **0.87286** | 0.87270 |
| %15 Eksik | - | 0.86396 | 0.86452 | 0.86513 | **0.86579** | 0.86462 |
| %20 Eksik | - | 0.85615 | 0.85698 | 0.85744 | 0.85858 | **0.85956** |
| %25 Eksik | - | 0.84774 | 0.85063 | 0.85105 | 0.85068 | **0.85180** |

Tablo 4.7'de yer alan n=500 birimlik ve düşük korelasyonlu türetilen veri yapılarında tam veri seti DSO'ı olan %88.75 değerine en yakın tahmin, %5'lik kayıp değerli veri setlerindeki % 88.02'lik tahmin oranı ile k=15 komşuluk değerindeki KNN atama yöntemi ile elde edilirken, en uzak tahmin %25'lik kayıp değerli veri setlerindeki %84.77'lik oran ile uzaklık matrisi yöntemine aittir.

Tablo 4.7 incelendiğinde, %5'lik, %10'luk ve %15'lik eksik veri setlerinde sırasıyla %99.180'lik, %98.353'lük ve %97.556'lık doğruluk payı ile k=15 komşuluk değerindeki KNN ile kayıp değer atama yönteminin, %20'lik ve %25'lik eksik veri setlerinde sırasıyla %96.854'lük ve %95.980'lik doğruluk payı ile k=20

komşuluk değerindeki KNN ile kayıp değer atama yönteminin en iyi sonuçları verdiği görülmektedir.

Tablo 4.8. $R_2$ korelasyon matrisine göre türetilen n=500 ve s=200 koşulu ile gerçekleştirilen benzetim çalışması sonuçları

| Veri Yapısı | Tam veri seti DSO | RF Uzaklık Matrisi DSO | KNN ile kayıp değer atama | | | |
|---|---|---|---|---|---|---|
| | | | K=5 DSO | K=10 DSO | K=15 DSO | K=20 DSO |
| N=500 | **0.93370** | - | - | - | - | - |
| %5 Eksik | - | 0.92915 | 0.92856 | 0.92927 | **0.92979** | 0.92914 |
| %10 Eksik | - | 0.92384 | 0.92389 | 0.92536 | 0.92557 | **0.92581** |
| %15 Eksik | - | 0.92080 | 0.91942 | 0.92131 | **0.92175** | 0.92166 |
| %20 Eksik | - | 0.91538 | 0.91365 | 0.91627 | **0.91705** | 0.91598 |
| %25 Eksik | - | 0.91327 | 0.91050 | 0.91282 | 0.91298 | **0.91442** |

Tablo 4.8'de yer alan n=500 birimlik ve orta korelasyonlu türetilen veri yapılarında tam veri seti DSO'ı olan %93.37 değerine en yakın tahmin %5'lik kayıp değerli veri setlerindeki % 92.98'lik tahmin oranı ile k=15 komşuluk değerindeki KNN atama yöntemi ile elde edilirken, en uzak tahmin %25'lik kayıp değerli veri setlerindeki %91.05'lik oran ile k=5 komşuluk değerindeki KNN atama yöntemine aittir.

Tablo 4.8 incelendiğinde, %5'lik eksik veri setlerinde %99.581'lik doğruluk payı ile k=15 komşuluk değerindeki KNN ile kayıp değer atama yönteminin, %10'luk eksik veri setlerinde %99.155'lik doğruluk payı ile k=20 komşuluk değerindeki KNN ile kayıp değer atama yönteminin, %15'lik ve %20'lik eksik veri setlerinde sırasıyla %98.720'lik ve %98.217'lik doğruluk payı ile k=15 komşuluk değerindeki KNN ile kayıp değer atama yönteminin ve %25'lik eksik veri setlerinde %97.935'lik doğruluk payı ile k=20 komşuluk değerindeki KNN ile kayıp değer atama yönteminin en iyi sonuçları verdiği görülmektedir.

Tablo 4.9. $R_3$ korelasyon matrisine göre türetilen n=500 ve s=200 koşulu ile gerçekleştirilen benzetim çalışması sonuçları

| Veri Yapısı | Tam veri seti DSO | RF Uzaklık Matrisi DSO | KNN ile kayıp değer atama | | | |
|---|---|---|---|---|---|---|
| | | | K=5 DSO | K=10 DSO | K=15 DSO | K=20 DSO |
| N=500 | **0.95525** | - | - | - | - | - |
| %5 Eksik | - | 0.95392 | 0.95292 | **0.95394** | 0.95393 | 0.95389 |
| %10 Eksik | - | **0.95394** | 0.95210 | 0.95230 | 0.95257 | 0.95366 |
| %15 Eksik | - | **0.95153** | 0.95012 | 0.95130 | 0.95097 | 0.95127 |
| %20 Eksik | - | **0.95112** | 0.94893 | 0.94905 | 0.94972 | 0.95013 |
| %25 Eksik | - | **0.94995** | 0.94702 | 0.94748 | 0.94874 | 0.94912 |

Tablo 4.9'da yer alan n=500 birimlik ve yüksek korelasyonlu türetilen veri yapılarında tam veri seti DSO'ı olan %95.53 değerine en yakın tahmin %5'lik kayıp değerli veri setlerindeki % 95.39'luk tahmin oranı ile k=10 komşuluk değerindeki KNN atama yöntemi ile elde edilirken, en uzak tahmin %25'lik kayıp değerli veri setlerindeki %94.70'lik oran ile k=5 komşuluk değerindeki KNN atama yöntemine aittir.

Tablo 4.9 incelendiğinde, %5'lik eksik veri setlerinde %99.863'lük doğruluk payı ile k=10 komşuluk değerindeki KNN ile kayıp değer atama yönteminin, %10'luk, %15'lik, %20'lik ve %25'lik eksik veri setlerinde ise sırasıyla %99.863'lük, %99.611'lik, %99.568'lik ve %99.445'lik doğruluk payıyla RF'nin uzaklık matrisi ile kayıp değer atama yönteminin en iyi sonuçları verdiği görülmektedir.

Tablo 4.7-9'da da n=100 ve n=200 birimlik benzetim çalışmalarına benzer sonuçlar elde edilmiştir. Kayıp değer yüzdeleri arttıkça, tüm yöntemler için DSO sonuçlarında bir düşüş görülmektedir ve değişkenler arasındaki ilişki arttıkça da DSO sonuçlarında artış gözlemlenmektedir. Bu benzetim çalışmasında da tüm yöntemler birbirlerine yakın sonuçlar sergilemektedir. Tam veri setlerine en yakın tahmin %0.13'lük hata ile yüksek korelasyonlu veri yapılarında, en uzak tahmin ise %3.97'lik hata ile düşük korelasyonlu veri yapılarında gözlemlenmektedir. Orta derecede ilişkili korelasyon matrisi ile türetilen veri setlerinde, kayıp değer sayısının

veri setinin %5'inden fazla olduğu durumlarda uzaklık matrisi ile atama yöntemi, k=5 olan KNN ile kayıp değer atama yönteminden daha iyi sonuçlar vermektedir. Düşük ve orta derecede ilişkili korelasyon matrisi ile türetilen veri setlerinde kayıp değer yüzdeleri arttıkça, k=10, k=15 ve k=20 değerleri kullanılarak yapılan KNN ile kayıp değer atama yöntemi, uzaklık matrisi ile atama yöntemine göre daha iyi sonuçlar sergilemektedir. Yüksek derecede ilişkili korelasyon matrisi ile türetilen veri setlerinde kayıp değer sayısı değişkenin %5'inden daha fazla ise, uzaklık matrisi ile atama yöntemi KNN ile kayıp değer atama yöntemlerinden daha iyi sonuçlar vermektedir.

Örneklem hacminin 1000 ve tekrar sayısının 100 olduğu benzetim çalışmasında da farklı korelasyon matrislerine göre elde edilen sonuçlar Tablo 4.10, Tablo 4.11 ve Tablo 4.12'de verilmiştir.

Tablo 4.10. $R_1$ korelasyon matrisine göre türetilen n=1000 ve s=100 koşulu ile gerçekleştirilen benzetim çalışması sonuçları

| Veri Yapısı | Tam veri seti DSO | RF Uzaklık Matrisi DSO | KNN ile kayıp değer atama | | | |
|---|---|---|---|---|---|---|
| | | | K=5 DSO | K=10 DSO | K=15 DSO | K=20 DSO |
| N=1000 | **0.90039** | - | - | - | - | - |
| %5 Eksik | - | 0.89277 | 0.89184 | 0.89285 | **0.89361** | 0.89327 |
| %10 Eksik | - | 0.88555 | 0.88625 | 0.88569 | **0.88705** | 0.88676 |
| %15 Eksik | - | 0.87845 | 0.87919 | 0.87961 | **0.87973** | 0.87918 |
| %20 Eksik | - | 0.86838 | 0.87082 | **0.87148** | 0.87059 | 0.87045 |
| %25 Eksik | - | 0.86228 | 0.86429 | 0.86612 | **0.86615** | 0.86544 |

Tablo 4.10'da yer alan n=1000 birimlik ve düşük korelasyonlu türetilen veri yapılarında tam veri seti DSO'ı olan %90.04 değerine en yakın tahmin, %5'lik kayıp değerli veri setlerindeki % 89.36'lık tahmin oranı ile k=15 komşuluk değerindeki KNN atama yöntemi ile elde edilirken, en uzak tahmin %25'lik kayıp değerli veri setlerindeki %86.23'lük oran ile uzaklık matrisi yöntemine aittir.

Tablo 4.10 incelendiğinde, %5'lik, %10'luk ve %15'lik eksik veri setlerinde sırasıyla %99.247'lik, %98.518'lik ve %97.705'lik doğruluk payı ile k=15 komşuluk değerindeki KNN ile kayıp değer atama yönteminin, %20'lik eksik veri setlerinde

%96.789'luk doğruluk payı ile k=10 komşuluk değerindeki KNN ile kayıp değer atama yönteminin ve %25'lik eksik veri setlerinde %96.197'lik doğruluk payı ile k=15 komşuluk değerindeki KNN ile kayıp değer atama yönteminin en iyi sonuçları verdiği görülmektedir.

Tablo 4.11. $R_2$ korelasyon matrisine göre türetilen n=1000 ve s=100 koşulu ile gerçekleştirilen benzetim çalışması sonuçları

| Veri Yapısı | Tam veri seti DSO | RF Uzaklık Matrisi DSO | KNN ile kayıp değer atama | | | |
|---|---|---|---|---|---|---|
| | | | K=5 DSO | K=10 DSO | K=15 DSO | K=20 DSO |
| N=1000 | **0.93804** | - | - | - | - | - |
| %5 Eksik | - | 0.93390 | 0.93324 | **0.93436** | 0.93353 | 0.93376 |
| %10 Eksik | - | 0.92994 | 0.93014 | 0.92994 | 0.92987 | **0.93027** |
| %15 Eksik | - | 0.92563 | 0.92468 | 0.92590 | 0.92617 | **0.92684** |
| %20 Eksik | - | **0.92229** | 0.91939 | 0.92220 | 0.92146 | 0.92220 |
| %25 Eksik | - | 0.91870 | 0.91539 | 0.91726 | 0.91824 | **0.91878** |

Tablo 4.11'de yer alan n=1000 birimlik ve orta korelasyonlu türetilen veri yapılarında tam veri seti DSO'ı olan %93.80 değerine en yakın tahmin %5'lik kayıp değerli veri setlerindeki % 93.44'lük tahmin oranı ile k=10 komşuluk değerindeki KNN atama yöntemi ile elde edilirken, en uzak tahmin %25'lik kayıp değerli veri setlerindeki %91.54'lük oran ile k=5 komşuluk değerindeki KNN atama yöntemine aittir.

Tablo 4.11 incelendiğinde, %5'lik eksik veri setlerinde %99.608'lik doğruluk payı ile k=10 komşuluk değerindeki KNN ile kayıp değer atama yönteminin, %10'luk ve %15'lik eksik veri setlerinde sırasıyla %99.172'lik ve %98.806'lık doğruluk payı ile k=20 komşuluk değerindeki KNN ile kayıp değer atama yönteminin, %20'lik eksik veri setlerinde %98.321'lik doğruluk payı ile RF'nin uzaklık matrisi ile kayıp değer atama yönteminin ve %25'lik eksik veri setlerinde %97.947'lik doğruluk payı ile k=20 komşuluk değerindeki KNN ile kayıp değer atama yönteminin en iyi sonuçları verdiği görülmektedir.

Tablo 4.12. $R_3$ korelasyon matrisine göre türetilen n=1000 ve s=100 koşulu ile gerçekleştirilen benzetim çalışması sonuçları

| Veri Yapısı | Tam veri seti DSO | RF Uzaklık Matrisi DSO | KNN ile kayıp değer atama | | | |
|---|---|---|---|---|---|---|
| | | | K=5 DSO | K=10 DSO | K=15 DSO | K=20 DSO |
| N=1000 | **0.95750** | - | - | - | - | - |
| %5 Eksik | - | **0.95657** | 0.95559 | 0.95595 | 0.95653 | 0.95620 |
| %10 Eksik | - | **0.95473** | 0.95374 | 0.95397 | 0.95473 | 0.95463 |
| %15 Eksik | - | **0.95422** | 0.95161 | 0.95239 | 0.95307 | 0.95313 |
| %20 Eksik | - | **0.95282** | 0.95048 | 0.95116 | 0.95175 | 0.95195 |
| %25 Eksik | - | **0.95195** | 0.94878 | 0.94957 | 0.94996 | 0.95037 |

Tablo 4.12'de yer alan n=1000 birimlik ve yüksek korelasyonlu türetilen veri yapılarında tam veri seti DSO'ı olan %95.75 değerine en yakın tahmin %5'lik kayıp değerli veri setlerindeki % 95.66'lık tahmin oranı ile uzaklık matrisi yöntemi ile elde edilirken, en uzak tahmin %25'lik kayıp değerli veri setlerindeki %94.88'lik oran ile k=5 komşuluk değerindeki KNN atama yöntemine aittir.

Tablo 4.12 incelendiğinde, %5'lik, %10'luk, %15'lik, %20'lik ve %25'lik eksik veri setlerinde sırasıyla %99.903'lük, %99.711'lik, %99.657'lik, %99.511'lik ve %99.420'lik doğruluk payıyla RF'nin uzaklık matrisi ile kayıp değer atama yönteminin en iyi sonuçları verdiği görülmektedir.

Tablo 4.10-12'de de n=100, n=200 ve n=500 birimlik benzetim çalışmalarına benzer sonuçlar elde edilmiştir. Kayıp değer yüzdeleri arttıkça, tüm yöntemler için DSO sonuçlarında bir düşüş olduğu görülmektedir ve değişkenler arasındaki ilişki arttıkça da DSO sonuçlarında artış gözlemlenmektedir. Bu benzetim çalışmasında da tüm yöntemler birbirlerine yakın sonuçlar sergilemektedir. Tam veri setlerine en yakın tahmin %0.093'lük hata ile yüksek korelasyonlu veri yapılarında, en uzak tahmin ise %3.81'lik hata ile düşük korelasyonlu veri yapılarında gözlemlenmektedir. Orta derecede ilişkili korelasyon matrisi ile türetilen veri setlerinde, kayıp değer sayısının veri setinin %5'inden fazla olduğu durumlarda uzaklık matrisi ile atama yöntemi, k=5 olan KNN ile kayıp değer atama yönteminden daha iyi sonuçlar vermektedir. Düşük ve orta derecede ilişkili korelasyon matrisi ile

türetilen veri setlerinde, k=10, k=15 ve k=20 değerleri kullanılarak yapılan KNN ile kayıp değer atama yöntemi, uzaklık matrisi ile atama yöntemine göre daha iyi sonuçlar vermektedir. Yüksek derecede ilişkili korelasyon matrisi ile türetilen veri setlerinde ise uzaklık matrisi ile atama yöntemi, KNN ile kayıp değer atama yöntemlerinden daha iyi sonuçlar vermektedir.

Tablo 4.1-12'de belirtilen DSO değerleri ile ilgili olarak aşağıdaki sonuçlar elde edilmiştir:

- Bütün benzetim çalışmalarında atama yöntemlerinden sonra elde edilen DSO değerlerinin, veri setlerinin tam olduğu durumlarda elde edilen DSO değerlerine oldukça yakın olduğu gözlemlenmektedir.
- Tüm yöntemler birbirlerine oldukça yakın sonuçlar sergilemektedir.
- Örneklem hacmi arttıkça, tüm atama yöntemleri sonucunda elde edilen DSO değerleri artmakta, neredeyse tam veri setine yaklaşmaktadır.
- Veri setlerinde yer alan değişkenler arasındaki ilişki arttıkça, tüm atama yöntemleri sonucunda elde edilen DSO değerleri artmaktadır. Tam veri setlerine en yakın tahminler yüksek korelasyonlu veri yapılarında, en uzak tahminler ise düşük korelasyonlu veri yapılarında gözlemlenmektedir.
- Değişkenlerde yer alan kayıp değer yüzdesi arttıkça, DSO değerleri azalmaktadır.
- Düşük ve orta derecede ilişkili korelasyon matrisi ile elde edilen veri setlerinde, k=10, k=15 ve k=20 değerleri kullanılarak yapılan KNN ile kayıp değer atama yöntemi, uzaklık matrisi ile atama yöntemine göre daha iyi sonuçlar vermektedir.
- Değişkenleri arasında yüksek derecede ilişki olan ve örneklem hacminin 100'den büyük olduğu veri setlerinde kayıp değerli gözlem sayısı %5' den fazla ise uzaklık matrisi ile atama yöntemi, KNN ile kayıp değer atama yöntemlerinden daha iyi sonuçlar vermektedir. Bu tür yüksek derecede ilişkili veri setlerinde, uzaklık matrisi yönteminden sonraki en iyi atama yöntemi ise k=20 değerleri kullanılarak yapılan KNN ile kayıp değer atama yöntemidir.

- Orta derecede ilişkili korelasyon matrisi ile elde edilen veri setlerinde örneklem hacminin 100'den büyük olduğu durumlarda, uzaklık matrisi ile atama yöntemi, k=5 olan KNN ile kayıp değer atama yönteminden daha iyi sonuçlar sergilemektedir.
- Düşük derecede ilişkili korelasyon matrisi ile elde edilen veri setlerinde en düşük DSO sonuçları uzaklık matrisi yöntemine ait iken, orta ve yüksek derecede ilişkili veri setlerinde en düşük sonuçlar k=5 olan KNN ile kayıp değer atama yöntemiyle elde edilmektedir.

## 4.2. Sağlık Alanı Uygulaması Bulguları

Periferik arter hastalığının araştırılmak istendiği tip 2 diyabetli 110 hastaya ait veri setinde, homosistein ve otuzonbeş oranı değişkenlerinde kayıp değerler gözlemlenmiştir. Değişkenlere ait kayıp değer sayıları ve yüzdeleri aşağıdaki gibidir.

Tablo 4.13. Değişkenlerin kayıp değer sayıları ve yüzdeleri

| | Kayıp Değer Sayısı | Kayıp Değer Yüzdesi |
|---|---|---|
| Homosistein | 24 | %21.82 |
| Otuzonbeş Oranı | 6 | %5.45 |

Atama işlemlerinin uygulanabilmesi için veri setinin, rasgele kayıp veri mekanizmasına sahip olması gerekmektedir. Bu koşul, Little'ın Tamamen Rasgele Olarak Kayıp Testi (Little's MCAR Test) ile araştırılmıştır. Test sonucunda veri setinin kayıp veri mekanizmasının MCAR bir yapı sergilediği gözlemlenmiştir ($\chi^2 = 39.18497$, Serbestlik derecesi=50, p=0.8650). Böylece bu veri setindeki kayıp değerli gözlemlere değer atama yapılmasına karar verilmiştir.

Uzaklık matrisi ile yapılan atama sonrası kurulan RF algoritması sonucu elde edilen sınıflandırma matrisi Tablo 4.14'de verilmektedir.

Tablo 4.14. Uzaklık matrisi atama yöntemi sonrasında kurulan RF algoritmasının sınıflandırma matrisi

| | | Tahmin Edilen Sınıf Değerleri | | |
|---|---|---|---|---|
| | | HASTA | NORMAL | Toplam |
| Gerçek Sınıf Değerleri | HASTA | 19 | 6 | 25 |
| | NORMAL | 1 | 84 | 85 |
| | Toplam | 20 | 90 | 110 |

Komşuluk değerleri k=5, 10, 15 ve 20 olarak belirlenen KNN ile kayıp değer atama algoritması ile tamamlanmış veri setlerinin RF algoritmasına yerleştirilmesinden sonra elde edilen sınıflandırma matrisleri sırasıyla Tablo 4.15, Tablo 4.16, Tablo 4.17 ve Tablo 4.18'de gösterilmektedir.

Tablo 4.15. k=5 koşulunda KNN ile kayıp değer atama yöntemi sonrasında kurulan RF algoritmasının sınıflandırma matrisi

| | | Tahmin Edilen Sınıf Değerleri | | |
|---|---|---|---|---|
| | | HASTA | NORMAL | Toplam |
| Gerçek Sınıf Değerleri | HASTA | 19 | 6 | 25 |
| | NORMAL | 2 | 83 | 85 |
| | Toplam | 21 | 89 | 110 |

Tablo 4.16. k=10 koşulunda KNN ile kayıp değer atama yöntemi sonrasında kurulan RF algoritmasının sınıflandırma matrisi

| | | Tahmin Edilen Sınıf Değerleri | | |
|---|---|---|---|---|
| | | HASTA | NORMAL | Toplam |
| Gerçek Sınıf Değerleri | HASTA | 21 | 4 | 25 |
| | NORMAL | 1 | 84 | 85 |
| | Toplam | 22 | 88 | 110 |

Tablo 4.17. k=15 koşulunda KNN ile kayıp değer atama yöntemi sonrasında kurulan RF algoritmasının sınıflandırma matrisi

| | | Tahmin Edilen Sınıf Değerleri | | |
|---|---|---|---|---|
| | | HASTA | NORMAL | Toplam |
| Gerçek Sınıf Değerleri | HASTA | 19 | 6 | 25 |
| | NORMAL | 1 | 84 | 85 |
| Toplam | | 20 | 90 | 110 |

Tablo 4.18. k=20 koşulunda KNN ile kayıp değer atama yöntemi sonrasında kurulan RF algoritmasının sınıflandırma matrisi

| | | Tahmin Edilen Sınıf Değerleri | | |
|---|---|---|---|---|
| | | HASTA | NORMAL | Toplam |
| Gerçek Sınıf Değerleri | HASTA | 19 | 6 | 25 |
| | NORMAL | 1 | 84 | 85 |
| Toplam | | 20 | 90 | 110 |

Beş farklı atama sonucunda RF algoritması kurularak elde edilen DSO değerleri Tablo 4.19'da verilmektedir.

Tablo 4.19. Atama yöntemlerine ait DSO değerleri

| | Kayıp değer atama yöntemleri | | | | |
|---|---|---|---|---|---|
| | Uzaklık Matrisi | K=5 | K=10 | K=15 | K=20 |
| DSO | 0.93636 | 0.92727 | 0.95455 | 0.93636 | 0.93636 |

Tablo 4.19'daki sonuçlar incelendiğinde, tüm atama yöntemlerin ardından elde edilen veri setleri ile kurulan RF algoritmalarının birbirlerine yakın sonuçlar verdikleri gözlemlenmiştir. Analizler sonucunda oldukça yüksek DSO değerleri elde edilmiştir. Bu atama yöntemlerinde k=10 ile yapılan kayıp değer ataması sonucunda en büyük DSO değeri elde edilmiştir. Bu sıralamayı uzaklık matrisi ve aynı sonucu veren k=15 ve k=20 ile yapılan atama yöntemi takip etmektedir. DSO oranı en küçük olan atama yöntemi ise k=5 ile yapılan kayıp değer atama yöntemidir.

Uygulama çalışması sonuçlarının benzetim çalışması sonuçları ile karşılaştırılması istenildiğinde, ilk incelenmesi gereken adım değişkenler arasındaki ilişkidir. Benzetim çalışmasında veri setlerinde değişkenler arası ilişki yapısı tanımlanırken, 5 ana değişken kullanılmıştır. Bu değişkenler bir korelasyon matrisi ile ilişkilendirilerek türetilmiştir. Daha sonra ise bu 5 değişkenden 2'si üzerinde kayıp değerler oluşturulmuştur. Bu nedenle uygulama veri setinde yer alan kayıp değerli iki değişkenin yanına üç önemli değişken yerleştirilerek, benzetim çalışmalarında olduğu gibi 5x5'lik bir ilişki matrisi elde edilmelidir. Bu önemli değişkenlerin bulunması için RF'nin bir özelliği olan değişken önemliliği kullanılmıştır. Atama işlemlerinden sonra kurulan RF algoritmalarında Gini ile elde edilen değişken önemliliği değerleri Tablo 4.20'de verilmiştir.

Tablo 4.20. Atama yöntemleri ardından kurulan RF algoritmaları için hesaplanan Gini değişken önemliliği sonuçları

| | | Kayıp Değer Atama Yöntemleri | | | | |
|---|---|---|---|---|---|---|
| | | Uzaklık matrisi | K=5 | K=10 | K=15 | K=20 |
| Değişkenler | Yaş | 1.09857 | 0.82547 | 1.01639 | 0.87869 | 0.95545 |
| | Diyabet Süresi | 1.19046 | 1.09973 | 1.26312 | 1.20017 | 0.93525 |
| | SolABI1 | **8.18258** | **8.67335** | **8.35621** | **8.84732** | **9.12068** |
| | SağABI1 | 5.38296 | 5.56853 | **5.45443** | 5.22714 | 5.43555 |
| | SolABI2 | **7.94684** | **7.86094** | **7.76900** | **7.41407** | **7.02011** |
| | SağABI2 | **5.46397** | **5.70529** | 5.29547 | **5.33835** | **5.69830** |
| | Ürik Asit | 0.40243 | 0.44191 | 0.43014 | 0.39176 | 0.32985 |
| | Homosistein | 1.56812 | 0.96819 | 1.07656 | 1.29021 | 1.18636 |
| | Fosfor | 0.25323 | 0.32164 | 0.30071 | 0.35471 | 0.26996 |
| | LDL | 0.26491 | 0.36549 | 0.34081 | 0.31568 | 0.27201 |
| | TG | 0.37175 | 0.43138 | 0.31159 | 0.44524 | 0.41900 |
| | HDL | 0.64888 | 0.56704 | 0.47121 | 0.47724 | 0.60425 |
| | Albuminuri | 0.49023 | 0.57609 | 0.51955 | 0.62562 | 0.52201 |
| | GFR | 0.43608 | 0.42163 | 0.53067 | 0.49442 | 0.44562 |
| | Otuzonbeş oranı | 0.74717 | 0.90649 | 0.89347 | 1.11672 | 1.00956 |
| | HbA1c | 0.36222 | 0.33973 | 0.33016 | 0.36348 | 0.42712 |
| | VKİ | 0.26584 | 0.25001 | 0.26430 | 0.40341 | 0.29057 |
| | HOMA | 2.28025 | 2.59608 | 3.11848 | 2.94072 | 3.15286 |

Gini değeri büyük olan değişkenler, en önemli değişkenlerdir. Bu durumda Tablo 4.20'deki sonuçlara bakılarak seçilecek olan üç değişken SolABI1, SolABI2

ve SağABI2 olarak belirlenmiştir. Belirlenen önemli değişkenler ile kayıp değerli değişkenler arasındaki ilişki Tablo 4.21'de verilmektedir.

Tablo 4.21. Uygulama veri setinin önemli değişkenlerinin ilişki tablosu

| | SolABI1 | SolABI2 | SağABI2 | Homosistein | Otuzonbeş oranı |
|---|---|---|---|---|---|
| SolABI1 | 1 | 0.952<br>P<0.001 | 0.878<br>P<0.001 | 0.383<br>P<0.001 | 0.337<br>P<0.001 |
| SolABI2 | 0.952<br>P<0.001 | 1 | 0.871<br>P<0.001 | 0.450<br>P<0.001 | 0.314<br>P<0.001 |
| SağABI2 | 0.878<br>P<0.001 | 0.871<br>P<0.001 | 1 | 0.422<br>P<0.001 | 0.351<br>P<0.001 |
| Homosistein | 0.383<br>P<0.001 | 0.450<br>P<0.001 | 0.422<br>P<0.001 | 1 | 0.143<br>P<0.001 |
| Otuzonbeş oranı | 0.337<br>P<0.001 | 0.314<br>P<0.001 | 0.351<br>P<0.001 | 0.143<br>P<0.001 | 1 |

Tablo 4.21'deki korelasyon katsayıları incelendiğinde, kayıp değerli değişkenlerin diğer değişkenler ile yaklaşık olarak orta derecede ilişkili olduğu görülmektedir. Benzetim çalışması sonucunda, orta derecede ilişkili değişkenler için en iyi kayıp değer atama yöntemi k=10, k=15 ve k=20 komşuluk değerleri kullanılarak gerçekleştirilen KNN ile kayıp değer atama yöntemi bulunmuştur. Bu yöntemin ardından, RF'nin uzaklık matrisi ile gerçekleştirdiği atama yönteminin iyi sonuçlar verdiği gözlemlenmiştir. En düşük DSO sonuçlarını ise, k=5 komşuluk değeri kullanılarak gerçekleştirilen KNN ile kayıp değer atama yöntemi vermiştir.

Uygulama bölümünde elde edilen sonuçlara göre yöntemler sıralandığında, benzetim çalışmasındakine benzer bir sıralama elde edilmiştir.

# 5. TARTIŞMA VE SONUÇ

Random Forests; aynı veri setinden bootstrap yöntemiyle örnekler seçilerek oluşturulan karar ağaçlarının sonuçlarını birleştiren, hem sınıflandırma hem de regresyon amacıyla kullanılabilen bir topluluk öğrenme yöntemidir. Karar ağaçları oluşturulurken her düğüm ayrımında tahminci değişkenlerin tümü yerine, bu değişkenlerden rasgele seçilen bir alt grup ile çalışılması hata oranlarını oldukça düşürmektedir. Bu özelliği ile bir boosting yöntemi olan Adaboost'dan daha etkin sonuçlar vermektedir. Bu yöntem, sahip olduğu faklı özellikleri nedeniyle de ön plana çıkmaktadır. OOB ile yapılan iç tahminler, RF kurulumunda kullanılan ağaç sayısı arttıkça elde edilen hata oranının sabit bir değere yaklaştığını göstermektedir. Aynı zamanda iç tahminler yolu ile tahminci değişkenlerin önemlilik dereceleri ölçülebilmektedir. RF'nin bu özelliği çok değişkenli veri setleri için oldukça kullanışlıdır (8).

RF yönteminin önemli özelliklerinden birisi de içerisinde bulundurduğu kayıp değer atama algoritmasıdır. Gözlemler arasındaki uzaklıkları kullanarak kayıp değerli gözlemlere yeni değer ataması yapmaktadır.

Kayıp değerli veri setleri, istatistiksel analizlerde genellikle karşılaşılan bir problemdir. Kayıp değerli gözlemlerin veri setinden çıkarılması sonucunda örneklem hacmi küçülmekte ve istatistiksel güç düşmektedir. Bu nedenle kayıp değerlere, yeni değerler atamak için farklı yöntemler geliştirilmiştir. Bu yöntemlerden biri K En Yakın Komşuluk Algoritması ile Kayıp Değer Atama Yöntemidir. Kayıp değerlerin, ait oldukları gözlemlerin k en yakın komşuları ile ilişkilendirilerek tahmin edildiği bu yöntem özellikle çok boyutlu veri setlerinde tercih edilmektedir (25, 26).

Tronskaya ve ark. 2001 yılında kayıp değerli mikrodizilim veri seti üzerinde yaptıkları çalışmada, KNN ile kayıp değer atama yöntemi, Tekil Değer Ayrışım (Singular Value Decomposition) tabanlı atama yöntemi ve ortalama atama (satır

ortalaması) yöntemi birbirleriyle karşılaştırılmıştır. Bu karşılaştırmalar sonucunda KNN ile kayıp değer atama yöntemi oldukça başarılı bulunmuştur. Kayıp değer yüzdesinin en fazla %20 olarak kullanıldığı bu çalışmada, k komşuluk değerinin çok küçük olduğu durumlarda kullanımı önerilmemiştir. Bu yöntemin en az 4 değişken kullanılarak uygulanması gerektiği belirtilmiştir (26).

Pantanowitz ve Marwala'nın 2008'de yaptıkları çalışmada, kayıp değerli bir HIV veri setine ilişkin sınıflandırma probleminde RF ile Veri Madenciliğinde kullanılan Genetik Algoritmalı Sinir Ağları (Autoassociative Neural Networks), Bulanık Mantığa Dayalı Sinir Yapıları (Neuro-Fuzzy Configurations) ve bunların ikişerli olarak birleştirildiği hibrit modeller karşılaştırılmıştır. Atamalar yapıldıktan sonra yöntemlerin karşılaştırılmasında, istatistiksel analizlerden ve HIV durumu sınıflandırma sonuçlarından faydalanılmıştır. Yapılan bu karşılaştırmalar sonucunda RF, kayıp değerli gözlemlere değer atama açısından diğer yöntemlerden oldukça başarılı bulunmuştur (22).

He 2006'da gerçekleştirdiği çalışmada CART ve RF yöntemlerinin kayıp değer atama algoritmaları incelenmiştir. Bu algoritmaları, parametrik olmayan bootstrap yöntemini kullanarak elde ettiği yeni bir atama yöntemi ile karşılaştırmıştır. Bu yöntem oluşturulurken belirli sayıda bootstrap örneklemi seçilmiştir. Her bootstrap örneği için kayıp değerli nicel değişkenlere medyan, nitel değişkenlere ise mod değeri ataması yapılmıştır. Daha sonra ise kayıp değerli değişkenlerin yapısına uygun regresyon yöntemi (basit doğrusal, poisson, lojistik) ile atama yapılmıştır. Atama işlemlerinin ardından CART ya da RF uygulanarak elde edilen sınıflandırma matrislerinden yalancı pozitif ve yalancı negatif değerleri elde edilmiştir. Bu süreç tüm bootstrap örneklerinde uygulandıktan sonra yalancı pozitif ve yalancı negatif değerleri ve bu değerler için güven aralıkları elde edilmiştir. Parametrik olmayan bootstrap ile kayıp değer atama algoritması ile RF ve CART'ın kayıp değer atama algoritmalarını karşılaştırırken 3 farklı veri seti kullanılmıştır. Parametrik olmayan bootstrap yönteminin daha düşük yalancı pozitif ve yalancı negatif sonuçları verdiği

elde edilmiştir. Herhangi bir benzetim çalışması yapmadan sonuçlandırılan bu çalışmada, parametrik olmayan bootstrap yönteminin kullanımını önerilmiştir (14).

Rieger ve ark.'nın 2010 yılında gerçekleştirdikleri çalışmada kurulumu RF'ye çok benzeyen Conditional Inference Forests (CIF), yönteminin kayıp değer atama algoritmasıyla, KNN ile kayıp değer atama yöntemi karşılaştırılmıştır. CIF yönteminin kayıp değer atama algoritması RF'den farklıdır. Bunun nedeni ise yedek değişkenler kullanarak yeni değer ataması yapmasıdır. Bu çalışmada düşük ve yüksek korelasyon yapılarındaki kayıp değerli veri setlerinde sınıflandırma ve regresyon problemleri incelenmiştir. Benzetim yöntemi kullanılarak karşılaştırmalar yapılmıştır. Örneklem hacmi n=200 ve tekrar sayısı s=100 olarak belirlenmiştir. İki değişkeninin %20'lik ve bir değişkeninin %10'luk oranlarda kayıp değer içerdiği veri setleri oluşturulmuştur. Elde edilen sonuçlara göre yedek değişkenler kullanılarak yapılan kayıp değer atama yöntemi ve KNN ile kayıp değer atama yöntemi birbirlerine üstünlük sağlayamamıştır. Farklı koşullarda yapılan karşılaştırmalarda birbirlerine oldukça yakın sonuçlar elde etmişlerdir. Rieger ve ark. bu çalışmada farklı örneklem hacmi, kayıp değer yüzdesi ve tekrar sayısında denemeler yapmamıştır (24).

Hapfelmeier 2012'de yaptığı çalışmasında yedek değişkenler ile çoklu atama yöntemini (MI) karşılaştırmıştır. Bu nedenle yedek değişkenleri yapısında bulunduran CIF ve CART ile MI ataması ardından tamamlanan veri seti ile kurulan RF yönteminin sonuçları karşılaştırılmıştır. Bu karşılaştırmalarda dört farklı simülasyon çalışması ve sekiz gerçek veri seti kullanılmıştır. Birbirlerine olan göreli etkinlikleri hesaplanarak sonuçlar elde edilmiştir. Bu sonuçlara göre, MI ve yedek değişkenler kayıp değer atama açısından birbirlerine üstünlük sağlayamamıştır. Halpfelmeier her iki yöntemin de kullanılabilir olduğunu belirtmiştir. Ancak MI yönteminin uygulanması daha karmaşık olduğu için yedek değişkenlerin kullanımının daha etkin olabileceğini önermiştir (13).

Bu tez çalışmasında, kayıp değerli bir veri seti için kurulan sınıflandırma probleminde; farklı örneklem hacimleri (n=100, 200, 500, 1000), korelasyon yapıları (r=0.1, 0.5, 0.9) ve tekrar sayılarında (s=1000, 500, 200, 100) çokdeğişkenli standart normal dağılımdan türetilen veri setlerinin önemli değişkenlerinde, farklı yüzdeliklerde (%5, %10, %15, %20, %25) eksik değer oluşturulduktan sonra RF'nin uzaklık matrisiyle atama yöntemi ve farklı komşuluk değerlerindeki (k=5, 10, 15, 20) KNN ile kayıp değer atama yöntemiyle ayrı ayrı tamamlanarak, bu atama yöntemlerinin karşılaştırılması amaçlanmıştır. Bu iki yöntemlerin karşılaştırılması iki aşamalı olarak gerçekleştirilmiştir.

İlk aşamada karşılaştırmalar benzetim yolu ile yapılmıştır. Kayıp değerli veri setleri RF'nin kayıp değer atama yöntemi ve k=5, 10, 15, 20 olmak üzere KNN ile kayıp değer atama yöntemi yoluyla ayrı ayrı tamamlanmıştır. Tamamlanan veri setleri ayrı ayrı, aynı RF algoritmasına yerleştirilmiştir. Elde edilen DSO oranları ile karşılaştırmalar yapılmıştır.

Tablo 4.1-3'de yer alan ve örneklem hacminin n=100 olduğu benzetim çalışmalarında, tam veri setlerine en yakın tahmin %0.12'lik hata ile yüksek korelasyonlu ve %5'lik kayıp değerli veri yapılarında, en uzak tahmin ise %4.47'lik hata ile düşük korelasyonlu ve %25'lik kayıp değerli veri yapılarında gözlemlenmiştir.

Tablo 4.4-6'da yer alan ve örneklem hacminin n=200 olduğu benzetim çalışmalarında, tam veri setlerine en yakın tahmin %0.13'lük hata ile yüksek korelasyonlu ve %5'lik kayıp değerli veri yapılarında, en uzak tahmin ise %4.03'lük hata ile düşük korelasyonlu ve %25'lik kayıp değerli veri yapılarında gözlemlenmiştir.

Tablo 4.7-9'da yer alan ve örneklem hacminin n=500 olduğu benzetim çalışmalarında tam veri setlerine en yakın tahmin %0.13'lük hata ile yüksek

korelasyonlu ve %5'lik kayıp değerli veri yapılarında, en uzak tahmin ise %0.093'lük hata ile düşük korelasyonlu ve %25'lik kayıp değerli veri yapılarında gözlemlenmiştir.

Tablo 4.10-12'de yer alan ve örneklem hacminin n=1000 olduğu benzetim çalışmalarında tam veri setlerine en yakın tahmin %0.093'lük hata ile yüksek korelasyonlu ve %5'lik kayıp değerli veri yapılarında, en uzak tahmin ise %3.81'lik hata ile düşük korelasyonlu ve %25'lik kayıp değerli veri yapılarında gözlemlenmiştir.

Tam veri setlerine en yakın tahminler yüksek korelasyonlu ve %5'lik kayıp değerli veri yapılarında, en uzak tahminler ise düşük korelasyonlu ve %25'lik kayıp değerli veri yapılarında gözlemlenmiştir. Düşük derecede ilişkili korelasyon matrisi ile elde edilen veri setlerinde en düşük DSO sonuçları uzaklık matrisi yöntemine ait iken, orta ve yüksek derecede ilişkili veri setlerinde en düşük sonuçlar k=5 olan KNN ile kayıp değer atama yöntemiyle elde edilmiştir.

Rieger ve ark.'nın çalışmasında olduğu gibi, bu tez çalışmasında da her iki yöntem birbirine oldukça yakın sonuçlar sergilemiştir, ancak birbirlerinden üstün olduğu koşullar gözlemlenmiştir. Örneklem hacimleri ve değişkenler arasındaki ilişki arttıkça, atama yöntemleri sonucunda elde edilen DSO değerleri de artmıştır. Kayıp değer yüzdelerindeki artış ise DSO değerlerinde düşüşe sebep olmuştur. Düşük ve orta derecede ilişkili korelasyon matrisi ile elde edilen veri setlerinde, k=10, k=15 ve k=20 değerleri kullanılarak yapılan KNN ile kayıp değer atama yöntemi, uzaklık matrisi ile atama yöntemine göre daha iyi sonuçlar vermektedir. Orta ve yüksek derecede ilişkili korelasyon matrisi ile elde edilen veri setlerinde örneklem hacminin 100'den büyük olduğu durumlarda, uzaklık matrisi ile atama yöntemi, k=5 olan KNN ile kayıp değer atama yönteminden daha iyi sonuçlar sergilemektedir. Küçük k komşuluk değerinin kullanılması önerilmemektedir. Bu sonuç, Tronskaya ve ark.'nın çalışmasındaki öneri ile de desteklenmektedir. Değişkenleri arasında yüksek derecede ilişki bulunan ve örneklem hacmi 100'den büyük olan veri setlerinde en iyi

atama yöntemi; uzaklık matrisi ile yapılan atama yöntemidir. Değişkenler arasındaki ilişki arttıkça, uzaklık matrisi ile yapılan atama yöntemi daha etkin sonuçlar vermektedir.

İkinci aşamada periferik arter hastalığına ait kayıp değerli veri setinin sınıflandırılmasına yönelik uygulama çalışması yapılmıştır. Bu çalışmada da atama yöntemlerine ilişkin DSO sonuçlarının birbirlerine oldukça yakın olduğu gözlemlenmiştir. Uygulama çalışmasında elde edilen sonuçların, benzetim çalışmasında elde edilen sonuçlara yakın olduğu belirlenmiştir.

Bu tez çalışmasında RF'nin uzaklık matrisi kullanılarak gerçekleştirilen kayıp değer atama algoritması ve KNN ile kayıp değer atama yöntemlerinin oldukça başarılı sonuçlar ortaya koyduğu gözlemlenmiştir. Her iki yöntem de kayıp değer atama amacıyla kullanılabilir. Ancak, daha etkin sonuçlar gözlemlenmek istendiğinde veri setinde yer alan önemli değişkenler arası ilişki incelenmelidir. Değişkenleri arasında düşük veya orta derecede ilişki olan kayıp değerli veri setlerinde KNN ile kayıp değer atama yöntemi kullanılmalıdır. Bu yöntem kullanılırken k=10, k=15 veya k=20 tercih edilmelidir. Daha küçük k komşuluk değerlerinin kullanımını önerilmemektedir. Değişkenleri arasında yüksek derecede ilişki olan kayıp değerli veri setlerinde ise RF'nin uzaklık matrisi kullanarak gerçekleştirdiği kayıp değer atama yöntemi kullanılmalıdır.

## KAYNAKLAR DİZİNİ


1. Acuna, E. and Rodriguez, C.,2004, The treatment of missing values and its effect in the classifier accuracy, "http://academic. uprm.edu/ eacuna/ IFCS04r.pdf", (2013-20-6), 9 p.

2. Akman, M., 2010, Veri madenciliğine genel bakış ve random forests yönteminin incelenmesi: sağlık alanında bir uygulama, Yüksek Lisans Tezi, Ankara Üniversitesi, Sağlık Bilimler Enstitüsü, 82 s.

3. Alpar, R., 2011, Çok değişkenli istatistiksel yöntemler, Detay Yayıncılık, Ankara, 853 s.

4. Andriyashin, A., 2005, Financial applications of classification and regression trees, Yüksek Lisans Tezi, Humboldt Üniversitesi, 41 p.

5. Bal, C., 2003, Çok gruplu veri setlerinde eksik gözlem sorununun çözümlenmesi ve sağlık alanında bir uygulama, Doktora Tezi, Eskişehir Osmangazi Üniversitesi, 130 s.

6. Batista, G. and Monard, M.C., 2003, A study of k-nearest neighbour as an imputation method, "http://citeseerx.ist. psu.edu/viewdoc/ summary? doi= 10.1.1.14.3558", (2013-20-6), 10 p.

7. Breiman, L., Friedman, J.H., Olshen, R.A. and Stone, C.J., 1998, Classification and regression trees, Chapman&Hall, Amerika Birleşik Devletleri, 358 p.

8. Breiman, L., 2001, Random forests, Machine Learning, 45, 5-32 p.

9. Breiman, L. and Cutler, A., 2004, RF tools for predicting and understanding data, "http://www.stat.berkeley.edu/~breiman/ RandomForests /interface04.pdf", (2013-12-5), 62 p.

10. Cutler, A., Cutler, D. R. and Stevens, J. R., 2012, Ensemble machine learning, Springer, New York, 329 p.

## KAYNAK DİZİNİ (devam ediyor)


11. Cutler, A., Cutler, D. R. and Stevens, J. R., Tree-based methods, "https:// www.nescent.org/wg/cart/images/9/91/Chapter6_March20.pdf", (2013-25-4), 21 p.

12. Gökay, G. E. ve Taşkın, Ç., 2005, Veri madenciliğinde karar ağaçları ve bir satış analizi, Eskişehir Osmangazi Üniversitesi Sosyal Bilimler Dergisi, 6, 2, 221-239 s.

13. Hapfelmier, A., 2012, Analysis of missing data with random forests, "http : // edoc.ub.uni-muenchen.de/15058/1/Hapfelmeier_Alexander.pdf", (2013-10-20), 168 p.

14. He, Y., 2006, Missing data imputation for tree-based models, Doktora Tezi, California Üniversitesi, 81 p.

15. Howell, D.C., The treatment of missing data, "http://www. uvm.edu/~ d ho well/StatPages/More_Stuff/Missing_Data/MissingDataFinal.pdf", (2013-9-12), 44 p.

16. Kavzoğlu, T. ve Çölkesen, İ., 2010, Karar ağaçları ile uydu görüntülerinin sınıflandırılması: Kocaeli örneği, Harita Teknolojileri Elektronik Dergisi, 2, 1, 36-45 s.

17. Kuzey, C., 2012, Veri madenciliğinde destek vektör makinaları ve karar ağaçları yöntemlerini kullanarak bilgi çalışanlarının kurum performansı üzerine etkisinin ölçülmesi ve bir uygulama, Doktora Tezi, İstanbul Üniversitesi, Sosyal Bilimler Enstitüsü, 318 s.

18. Lin, J., Extending Little's missing completely at random test for sparse missing data patterns, "https://www.academia.ed u/2034404/Extending_Littles_ Missing_Completely_at_Random_Test", (2013-10-21), 14 p.

19. Little, R.J.A. and Rubin, D.B., 1987, Statistical analysis with missing data, Wiley, New York, 278 p.

20. Martinez, W.L., Martinez, A.R., 2002, Computational statistics handbook with MATLAB, Chapman&Hall Publishers, Florida, 591 p.

## KAYNAK DİZİNİ (devam ediyor)


21. Özdamar, K., 2013, Paket programlar ile istatistiksel veri analizi cilt 2, Nisan Kitabevi, Ankara, 474 s.

22. Pantanowitz, A. and Marwala, T., 2008, Evaluating the impact of missing data imputation through the use of the random forest algorithm, "http://arxiv.org/ftp/arxiv/papers/0812/0812.2412.pdf", (2012-10-12), 11 p.

23. Qi, Y., Random forest for bioinformatics, "http://www.cs. cmu.edu /~qyj /papersA08/11-rfbook.pdf", (2013-20-4), 17 p.

24. Rieger, A., Hothorn, T. and Strobl, C., 2010, Random forests with missing values in the covariates, "http://epub.ub.uni-muenchen. de/ 11481/1/ techreport.pdf", (2012-11-20), 13 p.

25. Scheel, I., Aldrin, M., Glad, I.K., Sorum, R., Lyng, H. and Frigessi, A., 2005, The influence of missing value imputation on detection of differentially expressed genes from microarray data, Bioinformatics, 21, 23, 4272-4279 p.

26. Troyanskaya, O., Cantor, M., Sherlock, G., Brown, P., Hastie, T., Tibshirani, R., Botstein, D. and Altman, R.A., 2001, Missing value estimation mehods for dna microarrays, Bioinformatics, 17, 6, 520-525 p.

**EK-1**

**Veri Seti Türetiminde Kullanılan R Programı Kodları**

```
f=function(n,r){
h=mvrnorm(n,c(0,0,0,0,0),matrix(c(1,r,r,r,r,r,1,r,r,r,r,r,1,r,r,r,r,r,1,r,r,r,r,r,1),nrow=5))
bet=c(1,2,3,4,5)
a=h%*%bet
pi=1/(1+exp(-a))
pred=rbinom(n,1,pi)
for (i in 1:n){
if (pred[i]==0) pred[i]="neg"
else pred[i]="pos"
}
k=mvrnorm(n,matrix(0,25,1),diag(25))
df = data.frame(pred,h,k)
names(df)<-
c("pred","x1","x2","x3","x4","x5","x6","x7","x8","x9","x10","x11","x12","x13",
"x14","x15","x16","x17","x18","x19","x20","x21","x22","x23","x24","x25","x26",
"x27","x28","x29","x30")
list(df)
}
```

**EK-2**

**Benzetim Çalışmalarında Kullanılan R Programı Kodları**

```
fnn=function(n,s,r,yuzde){
orantum<-vector(length=s)
oran.imputed<-vector(length=s)
oran.kbes<-vector(length=s)
oran.kon<-vector(length=s)
oran.konbes<-vector(length=s)
oran.kyirmi<-vector(length=s)

for (m in 1:s){
set.seed(1*m)
a=f(n,r)
set.seed(2*m)
rftum=randomForest(pred~., data=a[[1]],importance=TRUE)
orantum[m]=(rftum$confusion[1,1]+rftum$confusion[2,2])/n
a.na<-a
set.seed(3*m)
for (i in c(2,6)) a.na[[1]][sample(n,n*yuzde),i]<-NA
set.seed(2*m)
a.imputed=rfImpute(pred~.,a.na[[1]],ntree=500,iter=5)
set.seed(2*m)
rf.imputed=randomForest(pred~., data=a.imputed,importance=TRUE)
oran.imputed[m]=(rf.imputed$confusion[1,1]+rf.imputed$confusion[2,2])/n

bbes<-kNNImpute(a.na[[1]][,2:31],5)
a.knnbes= data.frame(a.na[[1]]$pred,bbes$x)
names(a.knnbes)<-
c("pred","x1","x2","x3","x4","x5","x6","x7","x8","x9","x10","x11","x12","x13",
"x14","x15","x16","x17","x18","x19","x20","x21","x22","x23","x24","x25","x26",
"x27","x28","x29","x30")
set.seed(2*m)
rfknnbes=randomForest(pred~., data=a.knnbes,importance=TRUE)
oran.kbes[m]=(rfknnbes$confusion[1,1]+rfknnbes$confusion[2,2])/n

bon<-kNNImpute(a.na[[1]][,2:31],10)
a.knnon= data.frame(a.na[[1]]$pred,bon$x)
names(a.knnon)<-
c("pred","x1","x2","x3","x4","x5","x6","x7","x8","x9","x10","x11","x12","x13",
"x14","x15","x16","x17","x18","x19","x20","x21","x22","x23","x24","x25","x26",
"x27","x28","x29","x30")
```

```
set.seed(2*m)
rfknnon=randomForest(pred~., data=a.knnon,importance=TRUE)
oran.kon[m]=(rfknnon$confusion[1,1]+rfknnon$confusion[2,2])/n

bonbes<-kNNImpute(a.na[[1]][,2:31],15)
a.knnonbes= data.frame(a.na[[1]]$pred,bonbes$x)
names(a.knnonbes)<-
c("pred","x1","x2","x3","x4","x5","x6","x7","x8","x9","x10","x11","x12","x13",
"x14","x15","x16","x17","x18","x19","x20","x21","x22","x23","x24","x25","x26",
"x27","x28","x29","x30")
set.seed(2*m)
rfknnonbes=randomForest(pred~., data=a.knnonbes,importance=TRUE)
oran.konbes[m]=(rfknnonbes$confusion[1,1]+rfknnonbes$confusion[2,2])/n

byirmi<-kNNImpute(a.na[[1]][,2:31],20)
a.knnyirmi= data.frame(a.na[[1]]$pred,byirmi$x)
names(a.knnyirmi)<-
c("pred","x1","x2","x3","x4","x5","x6","x7","x8","x9","x10","x11","x12","x13",
"x14","x15","x16","x17","x18","x19","x20","x21","x22","x23","x24","x25","x26",
"x27","x28","x29","x30")
set.seed(2*m)
rfknnyirmi=randomForest(pred~., data=a.knnyirmi,importance=TRUE)
oran.kyirmi[m]=(rfknnyirmi$confusion[1,1]+rfknnyirmi$confusion[2,2])/n


}
list
(sonuc=cbind(eksiksiz=mean(orantum),uzaklıkmatrisli=mean(oran.imputed),knnbes
=mean(oran.kbes),knnon=mean(oran.kon),knnonbes=mean(oran.konbes),knnyirmi=
mean(oran.kyirmi)))
}
```

## ÖZGEÇMİŞ

**Bireysel Bilgiler**


| | |
|---|---|
| Adı-Soyadı | :Hülya YILMAZ |
| Doğum tarihi ve yeri | :21.06.1987 / İzmir |
| Uyruğu | :T.C. |
| Medeni durumu | :Bekar |
| İletişim adresleri | :hulyayilmaz@ogu.edu.tr |


**Eğitim Durumu :**

Malazgirt İlköğretim Okulu (1993-1998)

Bornova İlköğretim Okulu (1998-2001)

Suphi Koyuncuoğlu Anadolu Lisesi (2001-2005)

Ege Üniversitesi Fen Fakültesi İstatistik Bölümü (2005-2009)

Ege Üniversitesi Fen Bilimleri Enstitüsü İstatistik Yüksek Lisansı (2009-2011)

**Mesleki Deneyim** **:** Araştırma Görevlisi (2011- )

**Yayınlar**:

- Yılmaz, H. ve Sazak, H.S., Bozulmuş normal dağılım altında konum ve dağılım parametre tahmin edicilerinin etkinliklerinin benzetim yoluyla karşılaştırılması, Uluslararası Katılımlı 15. Ulusal Biyoistatistik Kongresi, Aydın, Kongre Bildiri Özetleri Kitabı, 94, 20-23 Ağustos 2013.

- Yılmaz, H. and Sazak, H.S., 2014, Double-looped maximum likelihood estimation for the parameters of the generalized gamma distribution, Mathematics and Computers in Simulation, 98, 18-30 p.