Van Yüzüncü Yıl Üniversitesi Eğitim Bilimleri Enstitüsü
Eğitim Bilimleri Bölümü Anabilim Dalı
Ölçme ve Değerlendirme Bilim Dalı

PISA 2015 VERİ SETİNDE OVA VE OVO STRATEJİLERİ ÇERÇEVESİNDE BAZI TEMEL SINIFLANDIRICILARIN PERFORMANSLARININ KARŞILAŞTIRILMASI

Hümeyra DEMİR

Yüksek Lisans Tezi

Van, 2018

PISA 2015 VERİ SETİNDE OVA VE OVO STRATEJİLERİ ÇERÇEVESİNDE BAZI TEMEL SINIFLANDIRICILARIN PERFORMANSLARININ KARŞILAŞTIRILMASI

Hümeyra DEMİR

Danışman
Dr. Öğr. Üyesi Gürol Zırhlıoğlu

Van Yüzüncü Yıl Üniversitesi Eğitim Bilimleri Enstitüsü

Eğitim Bilimleri Bölümü Anabilim Dalı

Ölçme ve Değerlendirme Bilim Dalı

Yüksek Lisans Tezi

Van, 2018

KABUL VE ONAY

Hümeyra Demir tarafından hazırlanan "PISA 2015 Veri Setinde OVA ve OVO Stratejileri Çerçevesinde Bazı Temel Sınıflandırıcıların Performanslarının Karşılaştırılması" başlıklı bu çalışma, 19.12.2018 tarihinde yapılan savunma sınavı sonucunda başarılı bulunarak jürimiz tarafından yüksek lisans tezi olarak kabul edilmiştir.

Prof. Dr. Selahattin GELBAL (Başkan)

Dr. Öğr. Üyesi Gürol ZIRHLIOĞLU (Danışman) (Üye)

Doç. Dr. Hayati ÇAVUŞ (Üye)

Yukarıdaki imzaların adı geçen öğretim üyelerine ait olduğunu onaylarım.

Doç. Dr. Fuat TANHAN

Enstitü Müdürü

BİLDİRİM

Hazırladığım tezin/~~raporun~~ tamamen kendi çalışmam olduğunu ve her alıntıya kaynak gösterdiğimi taahhüt eder, tezimin/~~raporumun~~ kâğıt ve elektronik kopyalarının Van Yüzüncü Yıl Üniversitesi Eğitim Bilimleri Enstitüsü arşivlerinde aşağıda belirttiğim koşullarda saklanmasına izin verdiğimi onaylarım:

☒ Tezimin/~~Raporumun~~ tamamı her yerden erişime açılabilir.

☐ Tezim/~~Raporum~~ sadece Van Yüzüncü Yıl Üniversitesi yerleşkesinden erişime açılabilir.

☐ Tezimin/~~Raporumun~~ … yıl süreyle erişime açılmasını istemiyorum. Bu sürenin sonunda uzatma için başvuruda bulunmadığım takdirde, tezimin/~~raporumun~~ tamamı her yerden erişime açılabilir.

19.12.2018

Hümeyra Demir

TEŞEKKÜR

Tez yazma sürecim boyunca yaptığı katkılar sebebiyle danışmanım Dr. Öğr. Üyesi Gürol ZIRHLIOĞLU'na,

Verdikleri katkılar sebebiyle tez jürimde bulunarak beni onurlandıran Prof. Dr. Selahattin GELBAL ve Doç Dr. Hayati ÇAVUŞ hocalarıma,

Bilimsel Araştırma Yöntemleri derslerinde akademik hayatım süresince kullanacağım bilgi ve tecrübelerini aktarması sebebiyle müstefid olduğum değerli hocam Prof. Dr. Hasan Basri MEMDUHOĞLU'na,

Tezimin her aşamasında istişare ettiğim kıymetli öğretmen arkadaşım Mehmet Ata OKUYUCU'ya,

Ve canım aileme verdikleri tüm emekler sebebiyle teşekkür ederim…

ÖZET

DEMİR, Hümeyra. *PISA 2015 Veri Setinde OVA ve OVO Stratejileri Çerçevesinde Bazı Temel Sınıflandırıcıların Performanslarının Karşılaştırılması,* Yüksek Lisans Tezi, Van, 2018.

Bu çalışmanın amacı hem çok sınıflı verilerin sınıflandırılması için kullanılan OVA ve OVO stratejilerinin hem de bu stratejiler altında uygulanan bazı ikili sınıflandırıcıların performanslarının karşılaştırılmasıdır. Çalışmanın yöntemi betimsel araştırmadır. PISA 2015 Türkiye uygulamasının fen başarı testi ve anket sonuçları veri seti olarak kullanılmıştır. PISA 2015 Türkiye uygulamasına 61 ilden 187 okul ve 5895 öğrenci katılmıştır. Okullar belirlenirken tabakalı seçkisiz örnekleme yöntemi kullanılmış daha sonra bu okullardan seçilen öğrenciler yine seçkisiz yöntemle belirlenmiştir. Okullar, istatistiki bölge birimleri sınıflamasına göre belirlenen 12 bölge içinden, eğitim türü, okul türü, okulların bulundukları yer ve okulların idari biçimleri dikkate alınarak oluşturulmuştur. 5895 örnek içeren veri setinden, boş veri içeren örnekler silindiğinde elde edilen 3459 örnekli veri seti çalışmada kullanılmıştır. 26 bağımsız 1 bağımlı değişkenden oluşan veri setinde bağımsız değişkenler kategorik olarak tanımlanmıştır. Veri dosyası arff formatına dönüştürülerek WEKA experimenter tezgâhında analizler gerçekleştirilmiştir. OVA ve OVO stratejileri altında belirlenen beş farklı algoritma veri setine uygulanmıştır. Test seçeneklerinden 10 katlı çapraz geçerleme, birleştirme stratejilerinden de oy verme tekniği kullanılmıştır. Analiz sonucunda OVA ve OVO stratejilerinin her ikisi altında en başarılı algoritmalar LR ve NB algoritmaları iken en başarısız algoritma KNN algoritmasıdır. En az örnek içeren sınıfı tahminleyebilme başarısı bakımından her iki strateji altında da en başarılı algoritma KNN algoritmasıdır. Algoritmalar doğruluk değeri ve hata ortalaması bakımından OVA stratejisi altında F metriği bakımından ise OVO stratejisi altında daha başarılıdır. Farklı algoritmaların performansları farklı ve daha çok sayıda veri seti üzerinde, farklı ayrıştırma ve birleştirme stratejileri ile, farklı test seçenekleri, farklı performans metrikleri, farklı algoritma parametreleriyle ya da yapılacak farklı önişlemler ile denenebilir.

Anahtar Sözcükler

Ayrıştırma stratejileri, Çok sınıflı sınıflandırma, Sınıflandırma algoritmaları, Performans karşılaştırma.

ABSTRACT

DEMIR, Humeyra. *Comparing the Performance of Some Basic Classifiers Within The Frameork of OVA and OVO Strategies in PISA 2015 Dateset*, Master Thesis, Van, 2018.

The aim of this study is to compare the OVA and OVO strategies used for the classification of multi-class data as well as the performance of some binary classifiers under these strategies. The method of the study is descriptive research. Test data and survey results of 5895 Turkish students who participated to PISA-2015 were used. PISA 2015 science achievement test and survey results obtained from Turkey were used as data set. 5895 students from 187 schools and 61 provinces joined PISA 2015 in Turkey while determining the schools, stratified random sampling method was used and then the students selected from these schools were determined by random method. The schools were formed by taking into account the types of education, type of school, the location of the schools and the administrative forms of the schools within the 12 regions determined by the classification of statistical district units.
In the data set containing 5895 samples, the 3459 sample data set obtained when the samples containing missing data were deleted was used in the study.
In the data set consisting of 26 independent and 1 dependent variables, independent variables were categorically defined. The data file was converted to arff format and analyzed at WEKA experimenter. It was applied five different algorithms to data set under OVA and OVO strategies. 10-fold cross validation from the test options and voting techniques from the joining strategies were used. The most successful algorithms under both OVA and OVO strategies are LR and NB algorithms, while the most failing algorithm is the KNN algorithm. The most successful algorithm under both strategies is the KNN algorithm for the success of estimating the class with the least sample. Algorithms in terms of accuracy and mean error metrics are more successful under the OVA strategy and F metric for the OVO strategy. The performance of the different algorithms can be tested on different and more data sets, with different decomposition and combining strategies, test options, performance metrics, algorithm parameters or pre processes to be made.

Key Words

Multiclass classification, Binarization techniques, Classification algorithms, Comparing performance.

İÇİNDEKİLER

KISALTMALAR

DVM: Destek Vektör Makinesi Algoritması

IEDMS: Uluslararası Eğitimsel Veri Madenciliği Topluluğu

IEEE: Elektrik ve Elektronik Mühendisleri Enstitüsü

KNN: K- En Yakın Komşuluk Algoritması

LR: Lojistik Regresyon Algoritması

MAE: Ortalama Mutlak Hata

MMH: Maksimum Marjinli Hiperdüzlem

NB: Naive Bayes Algoritması

OVA: One vs All Ayrıştırma Stratejisi

OVO: One vs One Ayrıştırma Stratejisi

PISA: Uluslararası Öğrenci Değerlendirme Programı

RAE: Rölatif Mutlak Hata

RMSE: Ortalama Karesel Hatanın Karekökü

RRSE: Rölatif Karesel Hatanın Karekökü

YSA: Yapay Sinir Ağları

TABLOLAR DİZİNİ

ŞEKİLLER DİZİNİ

1. BÖLÜM

GİRİŞ

Veri tabanı ve bilgisayar teknolojilerindeki gelişmeyle beraber günlük yaşantımızda yaptığımız birçok işlem kayıt altına alınmaya başlanmış, bu kayıt altına alınan veriler büyük veri yığınları oluşturmuştur. Kayıt altına alınan bu büyük miktardaki veriden anlamlı, işe yarar bilgiler ortaya çıkarma ve geleceğe yönelik tahminler yapabilme isteği veri madenciliği kavramını ortaya çıkarmıştır (Han, Kamber ve Pei, 2011).

Veri madenciliği; büyük miktardaki veri içerisinden gizli kalmış, işe yarar, güvenilir, potansiyel olarak kullanışlı ve daha önceden bilinmeyen bilgilerin, örüntülerin, kuralların, bağıntıların çeşitli algoritmalar, istatistiksel ve matematiksel teknikler, desen tanımlayıcı teknolojiler ve bilgisayar programları kullanılarak çıkarılması işlemidir (Akpınar, 2000; Babadağ, 2006; Fayyad, Piatetsky-Shapiro, Smyth ve Uthurusamy, 1996; Hung, Yen ve Wang, 2006). Elde edilen bilgiler herhangi bir konudaki karar verme aşamasında önemli rol oynamaktadır. Veri madenciliği kendisine çok çeşitli disiplinler içinde uygulama alanı bulmaktadır. Bu uygulama alanları arasında bankacılık, sigortacılık, telekomünikasyon, pazarlama, tıp, mühendislik, astronomi, gibi birçoğu sayılabilir. Eğitim alanı da veri madenciliği tekniklerinin uygulandığı önemli disiplinlerden biridir ve eğitim alanındaki uygulamalar eğitimde veri madenciliği alt disiplinini ortaya çıkarmıştır.

Eğitimsel ya da eğitimde veri madenciliği, eğitim ortamlarından gelen büyük boyutlardaki veriden bilgiler çıkarmak için yöntemler geliştiren ve bu yöntemler sayesinde öğrencileri ve öğrenme ortamlarını daha iyi anlamaya çalışan bir disiplindir (Uluslararası Eğitimsel Veri Madenciliği Topluluğu (IEDMS), 2016). Öğrencilere ait kişisel bilgiler, başarı durumları ve notlar gibi birçok bilginin veri tabanlarında kayıt altına alınması eğitimde veri madenciliğinin ortaya çıkma sebebi olarak düşünülebilir. Kayıt altına alınan bu veriler üzerine veri madenciliği süreçleri uygulanmakta ve elde edilen bilgiler, eğitimdeki problemlerin tespitinde, eğitim kalitesinin artırılmasında kullanılmaktadır (Özbay, 2015).

Bir veri setindeki örneklerin belirli kategorilere dahil edilmesi bir diğer deyişle sınıflandırma, birçok disiplinde ihtiyaç duyulan önemli bir uygulamadır. Sınıflandırma

problemlerinin uygulandığı veri setleri iki sınıf içerebildiği gibi ikiden fazla sınıf da içerebilir. Veri setinde ikiden fazla sınıf bulunduran sınıflandırma problemleri çok sınıflı sınıflandırma problemleri olarak adlandırılır. Çok sınıflı veriler günlük hayatta sıkça karşımıza çıkan veri tipidir. Benzer şekilde eğitim alanında da bağımlı değişkenin çok sınıflı olduğu veya çok sınıflı olarak ifade edildiği birçok durum karşımıza çıkmaktadır.

Çok sınıflı verilerde sınıflandırma problemi birçok şekilde ele alınabilir. En popüler tekniklerden biri, orijinal veri kümesini analiz edilmesi daha kolay olan iki sınıflı alt kümeye ayırmak ve her bir alt küme için farklı bir ikili model oluşturmaktır. Bu teknikler ayrıştırma teknikleri olarak bilinir (Galar, Fernández, Barrenechea, Bustince ve Herrera; 2011).

İki sınıflı sınıflandırma problemleri için geliştirilen algoritmaları çok sınıflı veriler için kullanmak geçersiz sonuçlar verebilir. Bu sebeple geliştirilen ayrıştırma tekniklerinden olan OVO (One vs One) ve OVA (One vs All) teknikleri mevcut veriyi iki sınıflı alt problemlere ayırır. m sınıflı bir veri için OVO stratejisinde m(m-1)/2 tane karşılaştırma yapılırken, OVA stratejisinde m tane karşılaştırma yapılır. Çünkü OVO stratejisi her bir sınıfı diğeriyle bir-e-bir olarak karşılaştırır. OVA stratejisinde ise karşılaştırma yapılırken her bir sınıf, diğer tüm sınıfların toplanmasıyla elde edilen sınıfa karşı indüklenerek, bir-e-hepsi şeklinde tarif edilebilecek bir yol izlenir (Lorena, De Carvalho ve Gama; 2008).

Algoritmaların performanslarının karşılaştırılmasında ortalama doğruluk, kesinlik, duyarlılık, AUC, F-ölçüsü gibi birçok değerlendirme ölçüsü kullanılmaktadır (Silahtaroğlu, 2016). Ayrıca performans sonuçları arasındaki farkların anlamlı olup olmadığını sınayan parametrik olmayan istatistiksel testlerden de yararlanılmaktadır (Garcia, Fernandez, Luengo ve Herrera, 2009). Bunların yanında modellerin hata miktarları ya da algoritmaların verilen işlemi gerçekleştirme süreleri de veri madenciliği çalışmalarında, elde edilen performansları değerlendirmek amacıyla kullanılabilir.

1.1. Problem

Bu çalışmanın ana problemi "Çok sınıflı verilerin sınıflandırılması için kullanılan çeşitli stratejiler altında, algoritmaların ve kullanılan stratejilerin performansları nasıldır?" şeklindedir.

1.1.1. Alt Problemler

1. OVA stratejisi altında doğruluk, ortalama hata, F ölçüsü ve kappa metrikleri bakımından en yüksek ve en düşük performans gösteren algoritmalar hangileridir?
2. OVA stratejisi altında en az sayıda veri içeren sınıftaki örnekleri doğru tahminleyebilme başarısı bakımından en iyi performans gösteren algoritma hangisidir?
3. OVO stratejisi altında doğruluk, ortalama hata, F ölçüsü ve kappa metrikleri bakımından en yüksek ve en düşük performans gösteren algoritmalar hangileridir?
4. OVO stratejisi altında en az sayıda veri içeren sınıftaki örnekleri doğru tahminleyebilme başarısı bakımından en iyi performans gösteren algoritma hangisidir?
5. Algoritmalardan her birinin performansları OVA ve OVO stratejilerinden hangisinde daha yüksektir?
6. OVA ve OVO stratejilerinden hangisi çok sınıflı veri setini sınıflandırmada daha başarılıdır?
7. J-48 algoritmasının ürettiği karar ağacına göre fen okuryazarlığını etkileyen değişkenler nelerdir?

1.2. Çalışmanın Amacı

Bu çalışmanın amacı hem çok sınıflı verilerin sınıflandırılması için kullanılan OVO ve OVA stratejilerinin hem de bu stratejiler altında uygulanan bazı ikili sınıflandırıcıların performanslarının karşılaştırılmasıdır.

1.3. Çalışmanın Önemi

Yürütülen literatür taramasında Türkiye'de daha önce çok sınıflı eğitim verileri üzerinden ayrıştırma teknikleri kullanılarak yapılan herhangi bir çalışmaya rastlanmamıştır. Çok sınıflı verilere eğitimde oldukça sık rastlanmaktadır ve örnekleri doğru sınıflara dâhil edebilmek çok sınıflı veriler için bir problem olmaktadır. Bu

anlamda çalışma, çok sınıflı eğitim verilerini sınıflandırmak için kullanılabilecek olan OVA ve OVO stratejilerini tanıtmakta ve eğitim verileri üzerinden bir uygulamasını içermektedir.

Çalışma bazı temel sınıflandırıcıların performanslarını değerlendirmek açısından da önemlidir. Algoritmaların örnekleri doğru sınıflandırma performansının yüksek olması değerlendirme yöntemimizin daha az hatalı olduğunu göstermektedir ki bu, etkili bir değerlendirme yapmak açısından oldukça mühimdir. Her ne kadar algoritmaların performansları kullanılan veri setine göre değişiklik gösterse de eğitim sektöründen gelen verilerle denenmesi çalışmanın önemi olarak değerlendirilebilecek bir husustur.

1.4. Varsayımlar

- Kayıp veri içeren örneklerin veri setinden çıkarılmasının algoritmaların performansını anlamlı ölçüde değiştirmeyeceği varsayılmıştır.

1.5. Sınırlılıklar

- Çalışma 2015 PISA sınavına katılan 3459 öğrenciye ait olan veri seti ile,
- Kullanılan Lojistik Regresyon (LR), Naive Bayes (NB), Destek Vektör Makinesi (DVM), J-48, K-En Yakın Komşuluk (KNN), algoritmaları ile,
- Bazı ayrıştırma stratejileri (OVA ve OVO) ile,
- Sonuçları değerlendirmek için kullanılan bazı performans metrikleri (doğruluk, ortalama hata, F- ölçüsü, kappa ile sınırlı olacaktır.

2. BÖLÜM

KURAMSAL ÇERÇEVE

2.1. Veri Madenciliği

Günümüzde veri tabanı ve bilgisayar teknolojilerindeki gelişmeyle beraber günlük yaşantımızda yaptığımız birçok işlem kayıt altına alınmaya başlanmış, bu kayıt altına alınan veriler büyük veri yığınları oluşturmuştur. Bu büyük miktardaki veriden anlamlı, işe yarar bilgiler ortaya çıkarma ve geleceğe yönelik tahminler yapabilme isteği veri madenciliği kavramını ortaya çıkarmıştır (Han vd., 2011).

Veri madenciliği; büyük miktardaki veri içerisinden gizli kalmış, işe yarar, güvenilir, potansiyel olarak kullanışlı ve daha önceden bilinmeyen bilgilerin, örüntülerin, kuralların, bağıntıların çeşitli algoritmalar, istatistiksel ve matematiksel teknikler, desen tanımlayıcı teknolojiler ve bilgisayar programları kullanılarak çıkarılması işlemidir. (Akpınar, 2000; Babadağ, 2006; Fayyad vd., 1996; Hung vd., 2006). Elde edilen bilgiler herhangi bir konudaki karar verme aşamasında önemli rol oynamaktadır.

Veri madenciliği teriminin yanlış bir isimlendirme olduğu söylenebilir. Kaya veya toprağın içinden altın çıkarma işi nasıl ki toprak veya kaya madenciliği olarak değil de altın madenciliği olarak isimlendiriliyorsa, veri madenciliği daha uygun bir şekilde veriden bilgi madenciliği olarak adlandırılmalıydı. Fakat "veriden bilgi madenciliği" terimi görece uzun bir terimdir. Bilgi madenciliği kavramı da madenciliğin, büyük miktarda veri içinden yapıldığı vurgusunu vermemektedir. Yine de canlı bir terim olan madencilik, küçük bir dizi değerli külçeyi büyük miktardaki işlenmemiş materyalden bulma işlemini karakterize ettiğinden hem veri hem de madenciliği içinde bulunduran bu yanlış adlandırma, popüler bir seçim olmuştur. Veriden bilgi madenciliği, bilgi çıkarma, veri/model analizi, veri arkeolojisi, veri tarama gibi terimler de veri madenciliği ile aynı ya da benzer anlamlar taşımaktadır (Han vd., 2011).

2.1.1. Veri Madenciliğinin Tarihçesi

Veri Madenciliği pek çok disiplinle ilişkili olduğundan, ortaya çıkışı da birçok alandaki gelişmeleri takip etmiştir. Tarihsel sürece bakıldığında bilgisayar ve veri tabanı teknolojilerindeki gelişmelerin veri madenciliğini ortaya çıkardığı söylenebilir.

1950'lerde ilk bilgisayarların sayım amacıyla kullanılmasından sonra 1960'larda veri tabanı ve verilerin depolanması kavramları ortaya çıkmış; 1960'ların sonlarında ise çok basit kuralların bilgisayarlar tarafından öğrenilmesi sağlanmıştır (Adriaans ve Zantinge, 1997). 1970'lerde farklı türde veri tabanlarının ortaya çıkmasıyla basit kurallara dayanan makine öğrenmesi gerçekleştirilmiştir. 1980'lerde veri tabanı teknolojisinin daha da gelişmesiyle; büyük miktarda veri içeren veri tabanları oluşturulmuştur. 1990'larda depolanan bu büyük miktardaki veriden, bilgi çıkarımı nasıl yapılır sorusuna cevap arayışları olarak 1989, KDD (IJCAI)-89 Veri Tabanlarında Bilgi Keşfi Çalışma Grubu toplantısı ve 1991, KDD (IJCAI)-89'un sonuç bildirgesi sayılabilecek "Knowledge Discovery in Real Databases: A Report on the IJCAI-89 Workshop" makalesi ortaya konulmuştur. Veri madenciliği ile ilgili temel tanım ve kavramların bu çalışmalarda ortaya konulması ile süreç daha da hızlanmış ve 1992 yılında veri madenciliği için ilk yazılım gerçekleştirilmiştir. 2000'li yıllardan günümüze kadar veri madenciliği sürekli gelişmekte, yeni algoritmalar geliştirilmekte ve veri madenciliği birçok disiplinde uygulama alanı bulmaktadır (Savaş, Topaloğlu ve Yılmaz, 2012).

2.1.2. Veri Madenciliği Uygulama Alanları

Her disiplinde toplanan büyük miktardaki veri, veri madenciliğinin birçok alanda uygulanmasını sağlamıştır. Bankacılık, sigortacılık, telekomünikasyon, pazarlama, tıp, mühendislik, astronomi ve eğitim bu uygulama alanları arasında sayılabilir.

Bu alanlarda karşılaşılan birçok problem veri madenciliği algoritmalarıyla çözülmeye çalışılmaktadır. Örneğin; kredi kartı harcamalarına göre müşteri gruplarının belirlenmesi, sahtekarlık tespiti, riskli müşteri örüntülerinin belirlenmesi, sigorta dolandırıcılıklarının tespiti sepet analizleri, hedef pazar bulma, promosyon ve kuponların etkisini ölçme, mağaza yerleşimi, hastalık teşhisi, ilaç geliştirme, üretim süreçlerinin optimizasyonu, öğrenci başarısı tahminleme, öğrenci verilerinin

tanımlanması, öğrencilerin başarısızlık sebeplerinin belirlenmesi, öğrenci başarılarının artırılması, eğitim-öğretim ortamlarındaki problemlerin saptanması, daha verimli eğitim-öğretim ortamlarının oluşturulması gibi birçok problem veri madenciliği ile çözüme kavuşturulmaya çalışılmaktadır (Akpınar, 2000; Altıntaş, 2006; Kalıkov, 2006; Özbay, 2015).

2.1.3. Veri Madenciliği Süreci

Veri madenciliğini tek bir adım olarak görmek yerine bir süreç olarak değerlendirmek gerekmektedir. Bu süreci veri setinin hazırlanması (veri temizleme ve verileri yeniden yapılandırma), veri madenciliği algoritmasını uygulama ve son aşamada da sonuçları sunma ve değerlendirme olarak özetleyebiliriz (Han vd., 2011).

2.1.3.1. Verilerin Hazırlanması

Veriler, belirli bir aralıkta yer alan değişkenler (interval), ikili değişkenler (binary), ayrık ve sıralı değişkenler (nominal, ordinal, ratio) olarak düzenlenebilir. Uygulamalarda toplanan verilerin yetersiz, tutarsız ya da gürültülü olması genel anlamıyla hata olarak adlandırılır. Hatalı veri toplama gereçleri, veri giriş problemleri, veri girişi sırasında kullanıcıların hatalı yorumları, veri iletimindeki hatalar, teknolojik sınırlamalar ve veri isimlendirmede ya da yapısındaki uyumsuzluklar hata kaynakları olarak sayılabilir. Bu hatalar nedeniyle tekrarlanan kayıtlar, çelişkili ve yetersiz verileri açığa çıkarır. Bu da verinin güvenilirliğini zedeler ve uygulanan veri madenciliği yönteminin yanlış sonuçlar vermesine neden olur (Oğuzlar, 2003).

Günümüzde gelişmiş veritabanlarında birçok türde ve büyük miktarda veri kayıt altına alınmaktadır. Veri madenciliği sürecinde kullanılacak olan bu veriler çoğu zaman gerçek olmayan ya da tutarsız bilgiler içermekte yahut bu verilerden bazıları kayıp veri durumunda olmaktadır. Tüm veriler tutarlı, doğru ya da eksiksiz olsa bile kimi zaman elimizdeki veri kullanacağımız tekniklerle uyum göstermeyebilmektedir (Silahtaroğlu, 2016). İşte bu sebeplerle veri üzerinde belirli işlemler yapılması gerekir. Bu işlemler verilerin hazırlanması ana başlığı ile verilmiştir.

Veri madenciliği uygulamalarını yapmadan önce veri setlerinin bir hazırlık sürecinden geçmesi gerekmektedir. Verilerin bazılarının kayıp veri durumunda olması, gürültülü veriler içermesi, yanlış ya da anlamsız verilerin bulunması, aynı anlama

gelecek birden fazla bilginin olması gibi sebepler bu hazırlık aşamasını gerekli kılmaktadır (Silahtaroğlu, 2016). Örneğin veritabanında kayıtlı birçok kişinin medeni durumu bazı kayıtlarda eksik olabilir, bu tür veriler genel itibariyle kayıp veri olarak adlandırılır. Bunun dışında kayıtlardan bazıları mantıksal hatalar içeren yanlış veriler olabilmektedir. Örneğin bir kişinin doğum tarihinin 1046 olarak girilmesi bu tür bir veridir ve genellenirse bu tür veriler gürültülü veriler olarak adlandırılır (Han vd., 2011). Bunların yanında mantık hatası olduğunu ilk bakışta anlayamayacağımız başka yanlış bilgiler de veritabanında kayıtlı olabilir. Örneğin bir ürüne ait kodun yanlış olarak girilmesi buna örnektir. Bazen de aynı anlama gelen kayıtlar farklı değişkenler şeklinde kayıt altına alınmış olabilir. Örneğin bir kişiye ait hem doğum tarihi hem de yaş verisinin olması bu türdendir ve bu tür verilerin birleştirilerek tek bir değişken gibi işleme alınması mümkündür (Silahtaroğlu, 2016).

Verilerin hazırlanması aşaması, verilerin temizlenmesi ve verilerin yeniden yapılandırılması başlıkları ile; verilerin yeniden yapılandırılması aşaması ise normalizasyon, boyut indirgeme, veri dönüştürme ve veri bütünleştirme başlıkları ile incelenecektir.

2.1.3.1.1. Verinin Temizlenmesi

Verinin temizlenmesi işlemi ile kastedilen gürültülü ya da kayıp verilerin ortadan kaldırılmasıdır. Bunun yanında grup ortalamasından aşırı sapmış olan uç değerlerin ve yanlış olan verilerin kaldırılması da verilerin temizlenmesi başlığı altında incelenebilir (Silahtaroğlu, 2016). Verileri temizlemek için kullanılabilecek yöntemleri aşağıdaki gibi sıralayabiliriz:

1. Kayıp veri içeren kayıtlar silinmek suretiyle veri dosyası temizlenebilir. Bu yöntemi kullanırken dikkatli olmak gerekir. Eğer kayıp verilerin olduğu kayıtlar toplam kayıt sayısı içerisinde büyük bir orana sahipse bu yöntem büyük miktarda bilgi kaybına sebep olabilir (Han vd., 2011).
2. Kayıp değerler için genel bir sabit kullanılabilir.
3. Kayıp değerler yerine ilgili sınıfın ortalaması atanabilir (Çölkesen, 2013).
4. Diğer değişkenlerin yardımı ile kayıp veriler tahmin edilerek doldurulabilir. Değerlerin tahmin edilmesi için regresyon, karar ağaçları, bayesyen

sınıflandırma algoritmaları gibi birçok algoritma kullanılmaktadır (Silahtaroğlu, 2016).

2.1.3.1.2. Verilerin Yeniden Yapılandırılması

Kullanacağımız algoritmalar verilerin her türdeki yapılanmasında çalışmayabilir. Örneğin bazı algoritmalar sürekli değerlere sahip verilerle çalışırken, başkaları kategorik verilerle çalışıyor olabilir. Dolayısıyla elimizdeki veri seti çalışmak istediğimiz algoritmaya uygun hale getirilmelidir. Bazen de çeşitli sebeplerle değişken sayısı ya da kayıt sayısı azaltılmak istenebilir (Silahtaroğlu, 2016). Daha birçok nedenle verilerin yapılandırılması önem arz etmektedir.

2.1.3.1.3. Normalizasyon

Veri madenciliği sürecinde elimizdeki verileri olduğu gibi kullanmamamız gereken durumlar oluşabilmektedir. Örneğin değişkenlerin ortalamalarının ve varyanslarının birbirinden dikkate değer biçimde farklı olması büyük ortalama ve büyük varyansa sahip olan değişkenlerin diğerlerinin rollerini baskılamasına sebep olacaktır. Bunun yanında değişkenlerin içerdiği aşırı değerler (en büyük ve en küçük değerler) analizin sağlıklı bir şekilde yapılmasını engeller. Bu tür sebeplerle verileri normalizasyon işlemine tabi tutmak gerekmektedir (Özkan, 2016).

Eldeki verilerin belirlenen bir aralığa indirgenmesi işlemine normalizasyon denir. min-max normalizasyonu, sıfır-ortalama normalizasyonu, ondalıklı normalizasyon gibi çeşitli normalizasyon yöntemleri bulunmaktadır (Han vd., 2011).

Min-maks normalizasyon yöntemi verileri 0-1 aralığında olacak şekilde yeniden ifade eder. Burada min bir değişkenin alacağı minimum değeri maks ise değişkenin alacağı maksimum değeri ifade eder. Bu yöntem, değerleri normalize etmek için aşağıdaki bağıntıyı kullanmaktadır (Silahtaroğlu, 2016).

$$X^* = \frac{X - X_{min}}{X_{max} - X_{min}}$$

Burada X^* verinin normalize edilmiş değerini, X ise verinin ham değerini ifade etmektedir. Örneğin en yüksek değeri 89 en düşük değeri ise 18 olan bir değişken için veriler 0-1 aralığına normalize edilmek istenirse;

$X^* = \frac{37-18}{89-18} = 0{,}268$ değeri elde edilir.

Sıfır ortalama yönteminde bir diğer adıyla Z-score standartlaştırmasında ortalama ve standart sapma değerleri kullanılarak normalizasyon işlemi yapılır. Bu işlemin matematiksel bağıntısı;

$$X^* = \frac{X - ort}{\sigma}$$

şeklinde ifade edilir. Burada "ort" ilgili değişkenin ortalamasını, σ ise değişkenin standart sapmasını ifade etmektedir (Silahtaroğlu, 2016).

2.1.3.1.4. Verilerde İndirgeme İşlemleri

Veri madenciliği uygulamalarında analiz işleminin daha kısa sürmesi ve sonuçların daha rahat yorumlanabilmesi için veri indirgeme işlemlerine ihtiyaç duyulmaktadır. Veri indirgeme işlemleri farklı şekillerde yapılmaktadır (Han vd., 2011). Veri setinde bulunan tüm değişkenlerin (ya da özniteliklerin) analize katılması gerekli veya uygun olmadığında "öznitelik seçimi" yöntemleri kullanarak boyut azaltma işlemleri yapılabilir (Doğancı vd., 2015).

Veri setindeki kayıt sayısının azaltılarak da veri indirgeme işlemi yapılabilir. Bu sayede daha küçük veri kümeleri ile tüm veri setinin örneklenmesi sağlanır (Özkan, 2016).

2.1.3.1.5. Veri Bütünleştirme

Farklı veri kaynaklarından elde edilen verinin tek bir notasyonla ifade edilmesi için veri bütünleştirme işlemleri yapılır. Veri madenciliği işlemleri için veri ambarı oluşturulurken veri bütünleştirme işlemlerinin yapılması gerekmektedir (Özkan, 2016).

2.1.3.1.6. Veri Dönüştürme

Veri tabanlarında kayıtlı olan birçok veri nicel ve sürekli verilerdir. Veri madenciliği sürecinde kullanacağımız bazı algoritmalar elimizdeki sürekli ve nicel olan bu değişkenlere ait olan ranj aralığındaki tüm değerleri dikkate alarak analizler

yapmaktadır. Bu durum hem işlem zamanını uzatmakta hem de elde edilecek olan sonuçları karmaşıklaştırmaktadır (Silahtaroğlu, 2016).

Veri tabanında bulunan sürekli ve nicel verilerin düşük-orta-yüksek gibi kategorik hale getirilmesi hem elde edilen sonuçların yorumlanmasını kolaylaştıracak hem de işlem zamanını kısaltacaktır. Veri aralıkları belirlenirken üzerinde çalışılan konunun teorik bilgisine ihtiyaç durulmaktadır. Teorik bilginin yanı sıra kategorileri belirlemek için literatürde çeşitli yardımcı yöntemler de bulunmaktadır. Görsel etiketleme yöntemi ve 3-4-5 yöntemi bu yardımcı yöntemlerdendir(Silahtaroğlu, 2016).

Görsel etiketleme yönteminde değişkenin histogramı çizdirilerek verilerin birbirinden ayrıldıkları yerler belirlenir. Histogram karar vermede her zaman yeterli olmamaktadır. (Örneğin verilerin çok dağınık olduğu durumlarda) ya da kullanılacak bir algoritmanın kendisinin etiketleme yapmasının istendiği durumlarda 3-4-5 yöntemi kullanılabilir (Silahtaroğlu, 2016).

2.1.3.2. Model değerlendirme

Veri önişleme basamağından sonra modelleme basamağı ile süreç devam etmektedir. Modelleme kısmında algoritma ve model seçimi yapılır. Daha sonra seçilen algoritma veri seti üzerinde çalıştırılarak elde edilen sonuçlar yorumlanır.

Kullanılacak algoritma belirlendikten sonra, veri kümesi eğitim veri seti ve test veri seti olarak ikiye ayrılır. Eğitim verisi modelin öğrenmesi, test verisi ise modelin geçerliliğinin test edilmesi için kullanılmaktadır. Modelin öğrenmesi gerçekleştirildikten sonra test kümesi ile modelin doğruluk derecesi belirlenir. Modelin doğruluk derecesi belirlenirken bazı yöntemler kullanılmaktadır. Bu yöntemlerden en bilineni k kat çapraz geçerleme yöntemidir. K- Kat çapraz geçerleme yönteminde veri seti rastgele k adet gruba ayrılır. Yapılan çalışmalarda genellikle k değeri 10 olarak seçilmektedir. Önce birinci grup test, diğer gruplar öğrenim için kullanılır. Bu süreç her defasında bir grubun test, diğer grupların öğrenim amaçlı kullanılması ile sürdürülür (Akpınar, 2000).

2.1.3.3. Veri Madenciliği Algoritması Uygulama

Veri önişleme yöntemleri veri setine uygulandıktan ve veriler hazır hale getirildikten sonra amacımıza hizmet edecek algoritmalar veri setine uygulanır. Bu

algoritmalar genel olarak sınıflandırma, kümeleme ve birliktelik kuralları çıkarımı algoritmaları olarak değerlendirilirler.

2.1.3.4. Sonuçları Sunma ve Değerlendirme

Veri madenciliği algoritmaları veriler üzerine uygulandıktan sonra sonuçlar çoğu kez tablolarla, grafiklerle, ağaçlarla görselleştirilerek sunulur. Örneğin bir hiyerarşik kümeleme yöntemi uygulanmış ise sonuçlar dendogram adı verilen özel grafiklerle sunulur (Özkan, 2016).

2.1.4. Veri Madenciliği Modelleri

Veri madenciliğinde kullanılan modeller, tahmin edici ve tanımlayıcı modeller olmak üzere iki ana başlık altında incelenmektedir. Tahmin edici modellerde, sonuçları bilinen verilerden hareket edilerek çeşitli kurallar geliştirilmesi ve bu kurallardan yararlanılarak sonuçları bilinmeyen veri kümeleri için sonuç değerlerin tahmin edilmesi amaçlanmaktadır. Örneğin bir okula ait veritabanında önceki dönemlerde başarısız ya da başarılı olmuş öğrencilere ilişkin birçok veri kayıtlı olabilir. Bu verilerde bağımsız değişkenler öğrencilerin çeşitli özellikleri; bağımlı değişken ise bir ölçüte göre başarılı olunup olunmadığıdır. Bu verilere uygun olarak kurulan model, daha sonra veri tabanına eklenecek olan öğrencilerin başarılı olup olamayacağının tahmininde kullanılmaktadır. Tanımlayıcı modellerde ise karar vermeye rehberlik etmede kullanılabilecek olan bilgiler mevcut verilerdeki örüntülerin tanımlanması yoluyla ortaya çıkarılmaktadır (Akpınar, 2000). Motivasyonu yüksek, günde X saat çalışan, okula ait hisseden öğrenciler ile motivasyonu düşük, günde Y saat çalışan ve okula ait hissetmeyen öğrencilerin başarı örüntülerinin birbirinden farklılık gösterdiğinin belirlenmesi tanımlayıcı modellere bir örnektir.

Veri madenciliği algoritmaları işlevlerine göre 3 grupta toplanabilir:

a. Sınıflama Algoritmaları (Classification)

b. Kümeleme Algoritmaları (Clustering)

c. Birliktelik Kuralları Çıkarımı (Association Rules) Algoritmaları

Sınıflandırma algoritmaları tahminleme işlemlerinde; kümeleme ve birliktelik kuralları çıkarımı algoritmaları ise tanımlama işlemlerinde kullanılmaktadır. Sınıflandırma modeli içerisinde; istatistiğe dayalı algoritmalar, mesafeye dayalı

algoritmalar, karar ağaçları, genetik algoritmalar ve yapay sinir ağları (YSA) bulunmakta; kümeleme modelinde hiyerarşik yöntemler, bölümlemeli yöntemler, yoğunluğa dayalı algoritmalar, grid temelli algoritmalar, genetik algoritmalar ve YSA bulunmakta; bağlantı analizi modelinde ise apriori algoritması, SETM algoritması, apriori TID algoritması ve GRI algoritması bulunmaktadır (Özkan, 2016).

Veri madenciliğindeki algoritmaların denetimli veya denetimsiz olarak öğrenirler. Veri setinde iyi tanımlanmış veya kesin bir hedef varsa denetimli öğrenmeden, elde edilmesi istenen sonuç için özel bir tanımlama yapılmamışsa veya belirsizlik söz konusu ise denetimsiz öğrenmeden bahsedilir (Friedman, Hastie ve Tibshirani, 2001).

Denetimli öğrenen algoritmalar veriyi anlamaya, keşfetmeye yönelik kullanılır. Denetimsiz öğrenen algoritmalar kategorisi bilinmeyen bir örneği daha önce sınıfı bilinen örneklerden elde edilen kuralları kullanır. Denetimsiz öğrenme yöntemiyle elde edilen bir bilgiyi, denetimli öğrenme yöntemiyle kıyaslamak, bulguların doğruluğu ve geçerliliği açısından faydalı olabilir (Koyuncugil ve Özgülbaş, 2009).

2.1.5. Veri Madenciliği ve İstatistik

İstatistik ve veri madenciliği kavramları arasında birtakım benzerlik ve farklılıklar bulunmaktadır. Yıllardır kullanılan istatistiksel yaklaşımlar ile görece daha yeni bir kavram olan veri madenciliği yöntemleri arasındaki benzerlik ve farklılıkların ortaya konulması büyük resmin görülebilmesi açısından önem taşımaktadır. İstatistik ve veri madenciliği alanlarının benzerlikleri aşağıda belirtilen şekilde sıralanabilir:

- Hem istatistiksel yöntemlerin hem de veri madenciliği yöntemlerinin bir veri setine uygulanmasındaki temel amaç veriden işe yarar bilgi çıkarımı yapabilmek, veriyi tanımlayabilmek, elde edilen bilgiyi geleceği tahmin etmede kullanabilmektir. Kısacası her iki alanın da hedefleri büyük benzerlik göstermektedir (Kuonen, 2004; Tüzüntürk, 2010).
- Veri madenciliği sürecinde kullanılan yöntemlerin birçoğunun arka planında istatistiksel yaklaşımlar bulunmaktadır (Kumar ve Bhardwaj, 2011). Kullanılan analizler incelendiğinde; kümeleme, diskriminant, regresyon, korelasyon analizi gibi birçok analiz türünün her iki alanda da kullanılan ortak yöntemler olduğu görülmektedir (Tüzüntürk, 2010).

- Her iki alan için kullanılan yöntemler dışında sahip olunan bir diğer benzerlik ise veri ön işleme aşamasıdır. Veri önişleme aşamasında her iki alan için kullanılan ortak yöntemler bulunmaktadır. Ön işleme aşaması, verinin analize girmeden önce işlenmeye hazır hale getirilmesini sağlamaktadır. Bu işlem analizin ve sonuçların geçerliliği açısından büyük önem taşımaktadır (Tüzüntürk, 2010).

Veri madenciliği ve istatistik alanlarının benzer yönleri olduğu gibi farklılıkları da vardır. Bu farklar, veri seti ve örneklem büyüklüğü, verinin toplanış amacı, hipotez olup olmaması gibi alt başlıklarda ortaya çıkmaktadır. İstatistik ve veri madenciliği alanları arasındaki farklılıklar aşağıdaki şekilde sıralanabilir:

- Veri setindeki veri sayısı, veri çeşidi ya da değişkenlerin sayısı arttığında klasik istatistiksel yaklaşımlar yeterli olmamaktadır. Bu tip durumlarda geliştirilen veri madenciliği algoritmaları veri setinin büyüklüğünden kaynaklanan problemleri çözmede çok daha başarılıdır (Ganesh, 2002).
- İstatistikte evreni temsil eden bir örneklem ile çalışılırken, veri madenciliğinde çoğu zaman evrenin tamamı veri setine dahildir (Meschenmoser, 2004; Akt: Emre ve Erol, 2017).
- İstatistik çalışmaları kapsamındaki veri belirli bir amaç için bir hipotezden yola çıkılarak toplanır. Veri madenciliğinde ise analizden önce tanımlanmış bir hipotezin varlığından söz edilemez (Mannila, 1996). İstatistiksel yaklaşımların kullanıldığı çalışmaların sonucunda hipotezin reddedilmesi veya kabul edilmesi söz konusudur. Veri madenciliği analizlerinde ise çalışma öncesi bir hipotez belirlenmez, analiz sonucunda elde edilen bilgiler yorumlanır (Tüzüntürk, 2010; Zhao ve Vice, 2006).
- Tüzüntürk (2010)'e göre veri madenciliği yöntemleri kullanılarak evrenin tamamının analize dahil olması ile daha özel/yerel bilgiye ulaşılır. Bu durumda veri madenciliği yöntemlerinin tümdengelim yaklaşımıyla ilerlediği söylenebilir. Fakat istatistiksel analizlerde evren içinden seçilen örneklemden elde edilen bilgilerden yola çıkılarak ulaşılan sonuçlar tüm ana kütle/veri seti için genellenir. Bu anlamda istatistiksel yaklaşımların tümevarım anlayışını benimsediği söylenebilir.

2.1.6. Eğitimde Veri Madenciliği

Eğitimsel veri madenciliği, eğitim ortamlarından gelen büyük boyutlardaki veriden bilgi elde etmek amacıyla yöntemler geliştiren ve bu yöntemler sayesinde öğrencileri ve öğrenme ortamlarını daha iyi anlamaya çalışan bir disiplindir (IEDMS, 2016). Günümüzde öğrencilere ait kişisel bilgiler, başarı durumları ve sınav notları gibi birçok veri kayıt altına alınmaktadır. Kayıt altına alınan bu verilerden elde edilecek bilgiler, eğitimdeki problemlerin tespitinde, eğitim kalitesinin artırılmasında kullanılabilir. (Özbay, 2015).

Veri madenciliğinin eğitimde kullanılma sürecinde veriler sınıf ortamlarından toplanabileceği gibi uzaktan eğitim ortamlarından da toplanabilir(Zaiane ve Luo, 2001). Geleneksel sınıf ortamlarında öğrencilerin kişisel farklılıkları dikkate alınmadığından; veri madenciliği süreçleri ile ortaya çıkarılan bilgiler, öğrencileri tanıma ve kişisel farklılıkları dikkate alma bağlamında da faydalı olacaktır.

Geleneksel sınıf ortamındaki veriler öğretmen ya da öğrenciler tarafından oluşturulan dosyalar, öğrencilerin ders ve kurs bilgileri, öğrenci derse katılım durumlarından elde edilirken olabilirken (Romero, Ventura, Espejo ve Hervas, 2008); uzaktan eğitimde ders içerik hazırlama araçları, eşzamanlı ve eşzamansız konferans sistemleri, anket ve kısa sınav bileşenleri, kaynak paylaşımı için sanal çalışma ortamları, beyaz tahta, not raporlama sistemi, günce kitabı, ödev yayınlama gibi pek çok veri elde etme yolları bulunmaktadır (Zaiane ve Luo, 2011).

Eğitimde verileriyle gerçekleştirilecek veri madenciliği çalışmaları ile, öğrencilerin başarılarına etki eden etmenler belirlenebilmekte, öğrenci profilleri belirlenerek gruplama yapılabilmekte, , öğrenci başarı durumları ya da mezuniyet notları tahmin edilebilmekte ve kullanılabilecek birçok başka uygulama ile eğitim kalitesi artırılabilmektedir (Özbay, 2015).

2.2. OVA ve OVO Stratejileri

Günlük hayatta karşımıza çıkan problemlerden bazıları örneklerin farklı sınıflara ya da kategorilere dâhil edilmesini içerir. Bir eğitim veri seti üzerinde makine öğrenmesi algoritmaları aynı alandan yeni verilerin sınıfını tahmin edebilmek amacıyla kullanılabilir. Sadece iki sınıf içeren veri setlerindeki sınıflandırma problemi, ikili sınıflandırma problemi olarak adlandırılır. Bir hastalığın tıbbi teşhisi ikili sınıflandırma

probleminin bir örneğidir. Hastalığın varlığı veya yokluğu sınıfları meydana getirmektedir. Böyle bir sınıflandırma problemi için, eğitilen sınıflandırma algoritması, hastalığı olup olmadığını belirlemek için söz konusu örnekten elde edilen klinik bilgileri kullanır (Lorena vd., 2008). Bununla birlikte, ikiden fazla sınıf içeren veri setleri de mevcuttur ve bu verisetlerindeki problemler çok sınıflı sınıflandırma problemleri olarak adlandırılmaktadır (Lorena vd., 2008).

İkili sınıflandırma tekniklerini kullanarak çok sınıflı problemleri ele almak için benimsenen iki yaklaşım bulunmaktadır. Bunlardan biri kullanılan algoritmanın iç operasyonlarının uyarlanması diğeri çok sınıflı problemin iki sınıflı sınıflandırma problemlerine ayrıştırılmasıdır. Bir ikili sınıflandırma algoritmasının çok sınıflı bir problem için genişletilmesi, bir diğer deyişle iç operasyonlarının uyarlanması bazı durumlarda pratik olmamakta veya gerçekleştirmek kolay olmayabilmektedir (Passerini, Pontil ve Frasconi, 2004). Bu nedenle, çok sınıflı problemleri iki sınıflı alt problemlere ayrıştırmak yani ayrışma stratejilerini kullanmak daha yaygındır (Lorena vd., 2008).

Makine öğrenmesi tekniklerini kullanarak sınıflandırma yapılırken önce (xi, yi) çiftlerinden oluşan bir f(x) fonksiyonu elde edilir burada $y_i \in \{1,..., k\}$ olan değerleri alabilir. k'nin iki olduğu sınıflandırma problemleri ikili sınıflandırma problemleri; k'nin ikiden büyük olduğu sınıflandırma problemleri ise çok sınıflı sınıflandırma problemleri olarak adlandırılır (Lorena vd., 2008).

Çok sınıflı bir sınıflandırma problemi doğal olarak ikili bir sorundan daha karmaşıktır; çünkü kullanılan sınıflandırıcı, verileri daha yüksek sayıda kategoriye ayırabilme kapasitesine sahip olmalıdır, bu da sınıflandırma hatalarını arttırır. Sonuç olarak, sınıf sayısı arttıkça karmaşıklık da artar (Lorena vd., 2008).

İkili sınıflandırma problemi çözme tekniklerinin çok sınıflı problemler için uyarlanmasında kullanılan en yaygın yaklaşım, problemi birkaç iki sınıflı alt probleme ayırmaktır. Her bir alt problem için kullanılan sınıflandırıcılardan elde edilen sonuçlar yapılacak çok sınıflı tahmin için birleştirilir (Lorena vd., 2008).

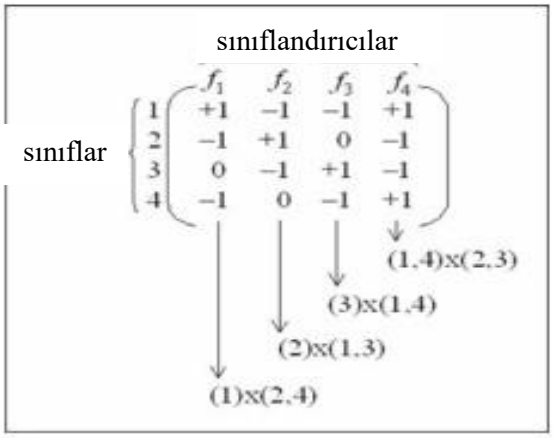
Ayrıştırma stratejilerinin çok sınıflı verilerin çözümlenmesinde kullanılmasının birkaç sebebi vardır. Ayrıştırma yaklaşımı sınıfları belirlemede karmaşıklığı azaltabilir ve aynı zamanda çok sınıflı problemler için genişletilebilen algoritmaların da kullanılabilmesini sağlar. Tüm sınıfları aynı anda ayıran bir sınıflandırıcı kullanmaktansa oluşturulan sınıf çiftleri için kullanılan doğrusal sınıflandırıcılardan

elde edilen sonuçları birleştirmek çok daha basit bir alternatiftir ve bu şekilde yapılmış çalışmalar literatürde mevcuttur (Lorena vd., 2008).

Ayrıştırma yaklaşımlarının kullanılması iki aşamada gerçekleştirilir. Birinci aşamada çok sınıflı problem ikili alt problemlere dönüştürülür ve eğitilecek olan ikili sınıflandırıcılar belirlenir. İkinci adımda ise ikili sınıflandırıcıların çıktıları yeni örneğin sınıfını belirlemek üzere birleştirilir (Mayoraz ve Moreira, 1996).

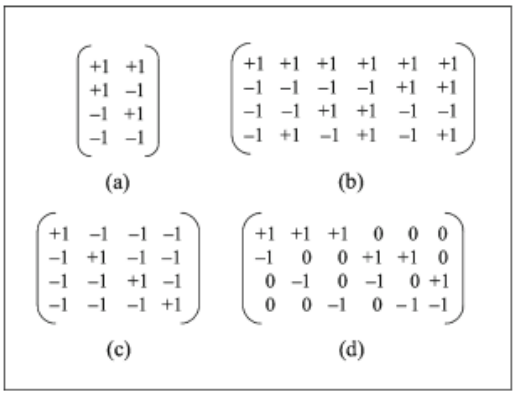
Çok sınıflı problemlerin ikili sınıflandırma problemlerine ayrılması için çeşitli alternatifler kullanılabilir. Genellikle bu ayrışmalar, Allwein, Schapire ve Singer (2000) tarafından önerilen bir kod-matris çerçevesi kullanılarak tarif edilir. Burada kod-matris M ile gösterilir. Bu matrisin satırları her sınıf için atanan kodları içerir. M matrisinin sütunları ise iki sınıflı parçaları tanımlar ve ikili sınıflandırıcıların bu sınıflar için belirlediği etiketlerle temsil edilir. k, çoklu sınıf problemindeki sınıfların sayısını ve l, çok sınıflı problem çözümünde kullanılan ikili sınıflandırıcıların sayısını temsil etmek üzere M matrisi k × l boyutuna sahiptir.

M'nin her elemanı -1, 0 veya +1 değerlerini almaktadır. Herhangi bir eleman +1 değerini alıyorsa; bu f_j sınıflandırıcısı uygulanırken bu i. satıra karşılık gelen sınıf pozitif olarak etiketleniyor demektir. -1 değeri negatif etikete, 0 değeri ise i. sınıftan gelen verinin fj sınıflandırıcısında değerlendirmeye katılmadığını gösterir. İkili sınıflandırıcılar daha sonra M sütunlarında temsil edilen etiketleri öğrenmek için eğitilirler. Şekil 1, dört sınıflı bir problem için oluşturulabilecek bir kod matris örneğini göstermektedir.



Şekil 1. Dört Sınıflı Bir Problem İçin Oluşturulabilecek Kod Matrisi Örneği

k sınıfı olan çok sınıflı bir problemin ayrıştırılmasında kullanılacak olan en kompakt yaklaşım $\log_2 k$ tane ikili sınıflandırıcı kullanılmasıdır (Mayoraz ve Moreira 1996). Şekil 2a'da dört sınıflı bir veri için kompakt bir kod-matris örneği verilmiştir. k sınıflı bir problemde farklı ikili tahmin ediciler toplamı $\frac{1}{2}(3k+1)-2k$ ile hesaplanır burada f ve –f ler eşit kabul edilir çünkü pozitif ve negatif etiketlerin ters çevrilmesiyle aynı ikili sınıflandırıcılar üretilir (Mayoraz ve Moreira 1996). Bunların $(2^{k-1}-1)$tanesi 0 elemanı olmaksızın sadece +1 ve -1 etiketlerini içerir. Dört sınıf için bu son matristen bir örnek şekil 2b de sunulmuştur. Literatürde ayrıştırma stratejileri arasında en yaygın kullanılanlar OVA ve OVO stratejileridir. (Hastie and Tibshirani, 1998; Knerr, Personnaz ve Dreyfus, 1990).



Şekil 2. 4 Sınıflı Bir Veri İçin Bir Kod-Matris Örneği

OVA stratejisinde k sınıflı bir problem verildiğinde k tane ikili sınıflandırıcı $f_i(x)$ üretilir. Bu tahminleyicilerden her biri i. sınıfı kalan sınıflardan ayırmak için eğitilir. OVA ayrışmasının temsili, diyagonaldeki elemanların +1 diğer elemanların -1 olduğu k.k boyutunda bir matris tarafından verilir. Dört sınıflı bir problem için şekil 2c de OVA stratejisi için bir matris verilmiştir. OVA ayrışması bir sınıftaki örneklerin sayısı diğer tüm sınıftan alınan verilerle oluşturulan kümedeki örnek sayısına kıyasla çok düşükse dezavantajlı olabilmektedir. Bu dengesizlik göz önüne alınan sınıf için doğru bir tahmin edici bulmayı zorlaştırabilir (Lorena vd., 2008).

OVO stratejisinde k sınıf için $k.\left(\frac{k-1}{2}\right)$ tane ikili sınıf oluşturulur ve bu sayıda ikili sınıflandırıcı kullanılır. Bu sınıflandırıcıların her biri i ve j sınıf çiftlerini ayırmak için kullanılır. Bu durumda kod matrisi boyutu $k.[k.\frac{(k-1)}{2}]$ dir ve her bir sütun bir çift

sınıf için kullanılan ikili bir sınıflandırıcıya karşılık gelir. i,j çiftini temsil eden bir sütunda i. satıra karşılık gelen elemanın değeri +1 ve j'ye karşılık gelen elemanın değeri -1'dir. Bu sütundaki diğer tüm elemanlar 0 değerindedir ve bu gösterir ki diğer sınıflardan gelen veri bu sınıflandırıcının çıkardığı sonuca dâhil edilmemiştir. Şekil 2d de dört sınıflı bir problemin kod matrisi görülmektedir. OVO stratejisinde üretilen ikili sınıflandırıcılar k^2 sayıda olmasına rağmen her birinin eğitimi yalnızca iki sınıfın verilerini içerir. Bununla birlikte ikili sınıflandırıcıların eğitilmesi için gereken süre genellikle yüksek değildir (Lorena vd., 2008).

OVO stratejisinin bazı dezavantajları vardır bunlardan birkaçı aşağıdaki gibi sıralanabilir:

- Sınıflandırılamayan bölgeler: Her ikili sınıflandırıcının farklı bir sınıf için oy kullanması mümkündür, dolayısıyla böyle bir durumda kazanan belirlenemez. Bu sebeple, bazı ilişki kırma (tie breaking) teknikleri uygulanmalıdır.
- Sınıflandırıcıların sayısı: OVA ile karşılaştırıldığında, OVO'nun daha fazla alt problem oluşturduğu görülür. Dahası, bu kadar çok alt soruna sahip olmanın dezavantajı, çoğunun alakasız olması ve birçok örneğe yanlış cevap vermek zorunda kalmalarıdır, çünkü her sınıflandırıcı, her deseni iki sınıftan birine atamak zorundadır. Eğer bir desen i sınıfına aitse, bu sınıfı ayırt etmek için eğitilmemiş olan tüm sınıflandırıcılar yanlış oyları alacaklardır. Bununla birlikte, OVO her alt problemde daha az örnek kullanmaktadır ve bu nedenle iki sınıf arasında bir karar sınırının yerleştirilmesi bağlamında daha özgürdür.
- Zayıf sınıflandırıcılar: Klasik yol, veri tabanı için en iyi taban sınıflandırıcıyı seçmek ve tüm alt problemleri bu sınıflandırıcı ile sınıflandırmaktır. Çok fazla alt problem olduğu için, bu temel sınıflandırıcıların hepsini ayırt etmekte güçlük çekmesi ve yanlış sonuçlar vermesi mümkündür. Bu, "aynı alt sınıflandırıcı maddenin tüm alt problemlerinde mi kullanılmalı yoksa tüm alt problemler bağımsız olarak mı ayarlanmalı?" sorusunu gündeme getirir (Arruti, Mendialdua, Sierra, Lazkano ve Jauregi, 2014).

Literatürde son sınıfı belirlemek için önerilen bir dizi toplama (birleştirme) stratejisi vardır (Galar, Fernandez, Barrenechea ve Herrera, 2015; Liu, Hao ve Tsang, 2008). Oylama stratejisi (Vote) basit ancak güçlü bir stratejidir. İkili oylama veya maksimum kazançlar kuralı olarak da adlandırılan oylama stratejisinde, tahmin edilen sınıfa ikili sınıflandırıcı tarafından bir oy verilir. Her sınıfın aldığı oylar sayılır ve en fazla oyu alan final sınıfı aşağıdaki şekilde tahmin edilmiş olur:

$$sınıf = \arg maks \sum_{1 \le j \ne i \le m} s_{ij} \qquad i = 1, \dots, m$$

$$s_{ij} = \begin{cases} 1 & r_{ij} > r_{ji} \\ 0 & diğer\ yerlerde \end{cases}$$ (Friedman, 1996)

Ağırlıklandırılmış oylama stratejisinde ise bilinmeyen örnek için tüm temel sınıflandırıcılar belirli bir güven düzeyinde bir tahmin ortaya koyar. En yüksek güven seviyesine sahip sınıf, çıktı sınıfı olarak öngörülür. Bununla birlikte, temel sınıflandırıcıların güven düzeylerine göre tahmin edilen sınıfa karar verilmesi problemli olabilir. Bunun nedeni, temel sınıflandırıcıların tüm sınıflar hakkında sınırlı bilgi içermesidir. Üstelik bağlar, yani eşit oy alan sınıflar ortaya çıkarsa tahmin edilen sınıf bağlı sınıflardan keyfi olarak seçilir (Zhang, Wang ve Liu, 2010).

$$Sınıf = \arg maks \sum_{1 \le j \ne i \le m} r_{ij} \qquad i = 1, \dots, m$$

2.3. Kullanılan Algoritmalar

2.3.1. J-48

J-48 algoritması Ross Quinlan tarafından geliştirilen C4.5 algoritmasının, WEKA programı için geliştirilmiş bir versiyonudur. Açık kaynak kodlu JAVA uygulamasında yazılmış olan bu algoritma veri madenciliğindeki sınıflandırma modeli içerisinde yer alan karar ağacı algoritmalarındandır. Sınıflandırma algoritmaları önceden sınıfları belli olan verileri kullanarak bir model oluşturur. Her bir örnek, veri setinde bulunan değişkenlerin değerleri ile tanımlanır. Sınıflandırma, bir dizi değişken değerinin kullanılarak sınıfı bilinmeyen bir örneği doğru kategoriye yerleştirme görevi olarak algılanabilir.

Karar ağaçları tümevarım yöntemiyle, sınıfları bilinen verileri kullanarak öğrenme işlemini gerçekleştirir. Karar ağaçları hem bilinmeyen örneklerin sınıfını

tahmin etme hem de mevcut veriyi tanımlama amacıyla kullanılmaktadır. Çıktılarının kolay okunabilir olması, veri tabanı sistemleri ile kolayca bütünleştirilebilmeleri, güvenilir olmaları (Vahaplar, 2003), ağaç yapılarının oluşturulmasında kullanılan kuralların anlaşılabilir ve sade olması (Safavian ve Landgrebe, 1991) gibi avantajları karar ağaçlarını çokça kullanılan popüler algoritmalardan yapmıştır.

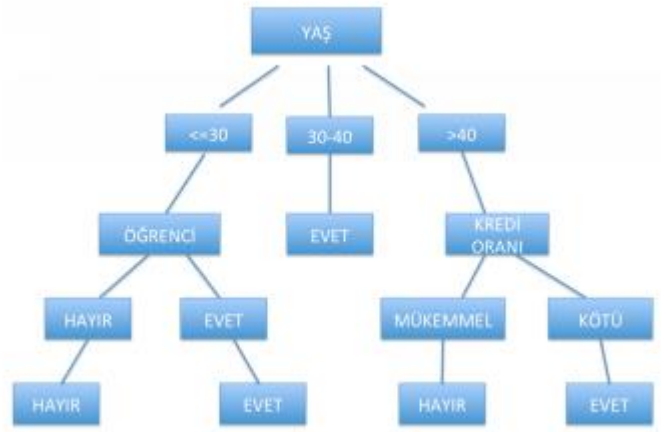
Karar ağaçları düğüm, yaprak ve dallardan meydana gelmektedir (Han vd., 2011). Veri setindeki her bir değişken bir düğümle, yapılan testler dallarla ve veri setindeki sınıflar yapraklarla temsil edilmektedir. Karar ağaçları sınıflandırma işlemini gerçekleştirirken çok aşamalı bir diğer deyişle ardışık ilerleyen bir yaklaşımla karmaşık bir sınıflandırma problemini basitleştirir (Safavian ve Landgrebe, 1991). If-then yapısında kurallar oluşturmak suretiyle kök düğümden yapraklara gerçekleştirilen çeşitli testlerle ağaç yapılandırılır. Yapılan her test ağaçta dallanmalara sebep olur ve bu durum bir yaprakla, yani bir sınıf etiketiyle, sonlanana kadar devam eder. Karar ağaçlarında kök düğümden yapraklara kadar giden bu yapı oluşturulurken sürekli test edilmesinin amacı ağacın, olabilecek en uygun sıralamasını bulmaktır (Pal ve Mather, 2003).

Karar ağaçlarında her düğümde yapılan testler, verilere ilişkin bir dizi sorular sorulmasını ve cevaplar elde edilmesini içerir. Elde edilen cevaplar toplanarak karar kuralları oluşturulur. Ağacın ilk düğümü olan kök düğümünde sorular sorulmaya başlanır ve dalları olmayan düğümler bir diğer deyişle yapraklar elde edilene kadar devam eder. (Pal ve Mather, 2003). Şekil 3'de bir karar ağacı örneği verilmiştir.

YAŞ	GELİR	ÖĞRENCİ	KREDİ-NOTU	BİLGİSAYAR SATIN ALANLAR
<=30	yüksek	Hayır	Kötü	Hayır
<=30	yüksek	Hayır	Mükemmel	Hayır
31-40	yüksek	Hayır	Kötü	Evet
>40	orta	Hayır	Kötü	Evet
>40	düşük	Evet	Kötü	Evet
>40	düşük	Evet	Mükemmel	Hayır
31-40	düşük	Evet	Mükemmel	Evet
<=30	orta	Hayır	Kötü	Hayır
<=30	düşük	Evet	Kötü	Evet
>40	orta	Evet	Kötü	Evet
<=30	orta	Evet	Mükemmel	Evet
31-40	orta	Hayır	Mükemmel	Evet
31-40	yüksek	Evet	Kötü	Evet
>40	orta	Hayır	Mükemmel	Hayır

Şekil 3. Veri Seti Örneği (Şeker, 2016)

Şekil 3’te yaş, gelir, öğrenci olma durumu ve kredi notuna ilişkin çeşitli değerler bulunmakta ve bu değerler baz alınarak örnekler bilgisayar satın alma ya da almama durumuna göre sınıflandırılmaktadır. Örneğin; 1. kişinin yaşının 30’dan küçük veya 30 a eşit, gelirinin yüksek, kredi notunun kötü olduğu ve öğrenci olmadığı bilinmekte ve 1. kişinin bilgisayar almayanlar sınıfına dâhil edildiği görülmektedir. Şekil 3’teki değerlere göre oluşturulmuş olan karar ağacı Şekil
4’te verilmiştir.



Şekil 4. Veri Kümesinin Karar Ağacına Dönüştürülmüş Hali (Şeker, 2016)

Şekil 4’te verilen karar ağacı örneğinde ilk bölünme yaş değişkenine göre yapılmış ve daha sonra yaş değişkeni içinde yer alan kategorilere göre bölünme devam etmiştir. En son ulaşılan “evet” ya da “hayır” şeklindeki yapraklar bilgisayar alıp almama durumunu ifade etmektedir.

Karar ağaçları oluşturulurken ağaçtaki dallanmaların hangi kritere göre yapılacağı bir diğer deyişle düğüm noktalarındaki değişkenlerin sırasının belirlenmesi için geliştirilmiş çeşitli yaklaşımlar bulunmaktadır. Bilgi kazancı ve bilgi kazanç oranı, Gini indeksi, Twoing gibi yaklaşımlar bunların başlıcalarıdır. Bilgi kazancı ve bilgi kazanç oranı yaklaşımları Quinlan (1993) tarafından ortaya atılmıştır.

Karar ağaçlarında ID3 algoritması bilgi kazancı yaklaşımını kullanırken bu algoritmanın geliştirilmiş hali olan C4.5 algoritması bilgi kazanç oranı yaklaşımını kullanmaktadır (Quinlan, 1993).

Karar ağacının hangi değişkenden başlanarak bölüneceği önemlidir. Bu amaçla entropi hesabı yapılmaktadır. Shannon bilgi teorisinde bulunan entropi, bir sistemdeki düzensizliğin ya da belirsizliğin ölçüsüdür (Özkan, 2016).

Eğitim veri setinin örneklerinin herhangi bir sınıfına ait olan örneklerinin değerleri $\{C_1, C_2, \dots, C_k\}$ olmak üzere k tane değer aldığını varsayarsak sınıf içerisindeki herhangi bir i. değere ait olasılık $P_i = \left({}^{C_i}/_{|T|} \right)$ bağıntısıyla, entropi ise $Entropi(T) = -\sum_{i=1}^{n} p_i \, log_2 \, p_i$ bağıntısıyla hesaplanmaktadır (Quinlan, 1993).

Hedef değişken için entropi değeri hesaplandıktan sonra ($Entropi(T) = -\sum_{i=1}^{n} p_i \, log_2 \, p_i \; denklemine \; göre$) hangi değişkenin bilgi kazancına bakılacaksa o değişkenin kategorilerinin hedef değişkenin kategorilerindeki dağılımına bakılır ve bu değişkenin her bir kategorisi için entropi hesaplanır. Daha sonra bu değişken için bilgi kazancının hesaplanması amacıyla

$$Kazanç \; (B,T) = Entropi(T) - \sum_{i=1}^{n} \frac{|T_i|}{|T|} Entropi(T_i)$$

eşitliği kullanılır. Eşitlikte B, veri setinde bulunan B değişkenini göstermektedir ve T sınıf değerlerinin $T_1, T_2, \dots \; T_n$ şeklinde alt kümelere ayrıldığı varsayılmıştır. Değişken seçimi yapılırken kazanç değeri en yüksek olan değişken tercih edilir.

J-48 algoritmasında ağaçlar oluşturulurken veya değişkenler belirlenirken kazanç ölçütü yanında kazanç oranı yaklaşımı da kullanılmaktadır. Özkan (2016)'a göre bu yaklaşımın kullanılmasının sebebi bir veri setindeki herhangi bir değişkenin birbirinden çok farklı değerler içermesi durumunda yanıltıcı sonuçlara sebep olmasıdır.

Kazanç oranının hesaplanmasında bilgi bölünmesi kavramından faydalanılır. Bilgi bölünmesi, kullanılan T veri kümesi için herhangi bir değişkenin değerinin belirlenmesi için gereken bilgi miktarını ortaya koymak üzere geliştirilmiştir. Bu bilgi miktarını hesaplayabilmek için;

$$Bölünme \; bilgisi(B) = -\sum_{i=1}^{k} \frac{|T_i|}{|T|} log_2 \left(\frac{|T_i|}{|T|} \right) eşitliği \; kullanılır.$$

Kazanç oranı için ise,

$$Kazanç\ oranı = \frac{Kazanç\ (B,T)}{Bölünme\ Bilgisi(B)}\ eşitliği\ kullanılmaktadır\ (Özkan, 2016).$$

Kazanç oranı yaklaşımında bu ölçüt kullanılarak, ağacın her bir düğümünde kazanç oranı maksimum olacak şekilde T eğitim kümesi tekrarlı bir şekilde ayrılır. İşleme, her bir yaprak sadece bir sınıfa ait gözlem değerlerini içerene kadar devam edilir (Kavzaoğlu ve Çölkesen, 2010).

Karar ağacı algoritmaları ağaç yapısını oluştururken, bazı düğümlerde ortaya çıkan kategoriler çok az örnek içerebilir. Bu durum, ağaç yapısının karmaşıklaşmasına neden olmakta ve ağacın yorumlanmasını zorlaştırmaktadır. Bu sebeplerle karar ağaçlarında budama işlemi gerçekleştirilir. Budama işlemi, gereksiz dallanmaların, yani sınıflandırma doğruluğunu çok fazla etkilemeyen dallanmaların, ağaçtan çıkarılması işlemidir (Özkan, 2016; Quinlan, 1987). Bu yöntemle daha az karmaşık ve daha kolay anlaşılabilir ve yorumlanabilir bir ağaç elde edilmiş olur. Karar ağaçlarının budanması işleminde genellikle iki yöntem kullanılmaktadır (Breiman, Friedman, Olshen ve Stone, 1984). Bunlardan ilki ağaç oluşturulurken yapılır ve ön budama yöntemi olarak isimlendirilir; diğer yöntemde ise ağaç oluşturulduktan sonra budama işlemi uygulanır ve bu yöntem son budama yöntemi olarak isimlendirilir. C4.5 algoritmasında ön budama yöntemi kullanılır ve bu yöntem, daha az hesaplama içermesi, veri setinin bölünmesi için mümkün olan en iyi yolu araştırması ve elde edilen bilgi kazancını değerlendirmesi gibi avantajlara sahiptir. Bu değerlendirmede belirli bir eşik değerinin altına düşüldüğünde bölünme kabul edilmez ve veri için en uygun yaprak olduğuna karar verilir (Breiman vd., 1984; Pal ve Mather, 2003). Budama işleminin yapılmasının bir diğer amacı da ağaçtaki hata oranını en küçük yapabilmektir. Budama yapılan dalın hata oranını düşürmesi beklenmektedir (Kavzaoğlu ve Çölkesen, 2010).

2.3.2. KNN

Bir veri setindeki örneklerin belirli kategorilere dâhil edilmesi bir diğer deyişle sınıflandırma, birçok disiplinde ihtiyaç duyulan önemli bir uygulamadır (Keller, Gray ve Givens, 1985; Mao, Hu, Wang ve Moore, 2015). KNN algoritması da eski fakat popüler, iyi bilinen ve etkili sınıflandırma algoritmalarındadır (Batista ve Silva, 2009; Bhatia ve Vandana, 2010; Qiu, Kang ve Zhang, 2008). Bu algoritma T.M Cover ve P.E.

Hart tarafından önerilmiştir ve sınıflandırma yaparken ele alınan örneğin k en yakın komşuluğundaki örnekleri dikkate alarak sınıflandırma yapar (Cover ve Hart, 1967).

KNN algoritmasının çeşitli avantajları ve dezavantajları bulunmaktadır. Avantajları arasında eğitilmeye ihtiyaç duymaması, uygulanmasının kolay olması, gürültülü eğitim verilerine karşı dirençli olması (Bhatia ve Vandana, 2010), eğitim veri seti büyük olduğunda da etkin sonuçlar vermesi, ilgisiz değişkenlerin olması durumunda da model oluşturabilmesi (Aha, Kibler ve Goldstone, 1991) gibi özellikler gösterilebilirken dezavantajları arasında yüksek miktarda bellek alanına ihtiyacı olması, sınıfı belirlenmek istenen örneğin k komşuluğundaki bütün uzaklıkların hesaplanıyor olması sebebiyle maliyetinin yüksek oluşu, performansın çeşitli parametrelerden etkileniyor oluşu sayılabilir (Bhatia ve Vandana, 2010; Duda, Hart ve Stork, 2000; Liu ve Zhang, 2012; Shmueli, Patel ve Bruce, 2010).

KNN algoritması, örnek tabanlı öğrenme algoritmalarındandır. Örnek tabanlı öğrenme algoritmalarında, öğrenme işlemi eğitim setinde tutulan verilere dayalı olarak gerçekleştirilmektedir. Yeni örnek sınıflandırılırken, eğitim setinde yer alan örnekler ile arasındaki benzerlik esas alınmaktadır (Mitchell, 1997). KNN algoritmasında her örnek n boyutlu bir uzayda bir noktayı temsil edecek şekilde tutulur. Bir örneğin sınıfı belirlenmek istendiğinde bu örneğe en az uzaklığa sahip olan olan k tane örneğin çoğunlukla seçtiği sınıf bu örneğin sınıfı olarak belirlenir (Han vd., 2011).

KNN algoritmasında sınıflandırılacak olan örneğin k yakınlığındaki komşularına olan uzaklığının nasıl ölçüleceği önemli bir parametredir. Bu uzaklık öklit uzaklığı ile ölçülebileceği gibi başka ölçüler de kullanılmaktadır. KNN algoritması az sayıda parametreye ihtiyaç duyduğundan karmaşık bir yapıya sahip değildir. Sınıflandırmaya k komşuluktaki örneklerin çoğunluk oyuyla karar verilmektedir. Bu durum, veri seti simetrik olmadığında, yeni örneklerin sınıfları belirlenirken çoğunlukta olan sınıfların baskın olmasına sebep olmaktadır (Coomans ve Massart, 1982). Bu sebeple uzaklık ölçütünün etki değerine farklı şekillerde ağırlık atayan yöntemler bulunmaktadır (Gartner, Lloyd ve Flach, 2004).

KNN algoritmasının performansında etkili ve önemli parametreler uzaklık ölçütü, komşu sayısı (k) ve ağırlıklandırma yöntemidir. Uzaklık ölçütü olarak minkowski uzaklığı, öklid uzaklığı, manhattan uzaklığı, chebyschev ve dilca uzaklıkları kullanılmaktadır.

Minkowski Uzaklığı: Öklid uzayında tanımlı bir dizidir. Herhangi iki nokta P ve Q arasındaki Minkowski uzaklığı $P=(x_1, x_2, \ldots, x_n)$ ve $Q=(y_1, y_2, \ldots, y_n)$ olmak üzere, aşağıdaki eşitliğe göre hesaplanmaktadır:

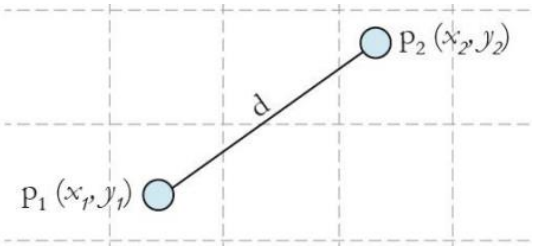
$$\left(\sum_{i=1}^{N}|x_i - y_i|^p\right)^{1/p} \qquad \text{(Cover ve Hart, 1967).}$$

Minkowski uzaklığını hesaplamak için kullanılan yukarıdaki bağıntı genel bir bağıntıdır ve p'nin farklı değerleri için farklı uzaklık ölçütlerinin hesaplandığı bağıntılara dönüştürülebilmektedir. Örneğin bu bağıntıda p=2 alınırsa Öklit uzaklığı, p=1 alınırsa Manhattan uzaklığı ve n→∞ olduğu özel durumda ise Chebyschev uzaklığı hesaplanabilir (Kresse ve Danko, 2012).

Öklit uzaklığı sınıflandırma modelinde en sık başvurulan uzaklık ölçütüdür. Öklit uzaklığı, iki nokta arasındaki doğrusal uzaklıktır. Herhangi iki nokta, $P=(x_1, x_2, \ldots, x_n)$ ve $Q=(y_1, y_2, \ldots, y_n)$ olmak üzere, P ve Q olarak ifade edilirse bu iki nokta arasındaki Öklit uzaklığı, aşağıdaki eşitliğe göre hesaplanır (Kresse ve Danko, 2012):

$$\left(\sqrt{\sum_{i=1}^{n}(x_i - y_i)^2}\right)$$

Burada x_i ve y_i P ve Q noktalarının x ve y boyutlarındaki değerleridir.



Şekil 5. $\boldsymbol{P_1}$ $\boldsymbol{ve}$ $\boldsymbol{P_2}$ Noktaları Arasındaki Öklit Uzaklığı

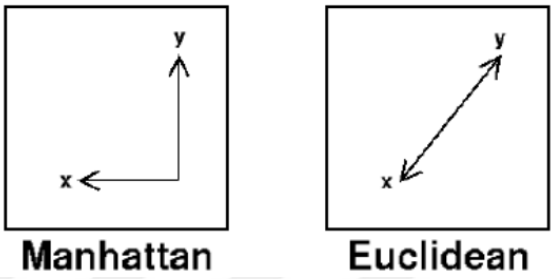
Şekil 5'te P_1 ve P_2 noktaları arasındaki Öklit uzaklığı olan d gösterilmiştir. Burada x_1 ve y_1 P_1 noktasının koordinatları, x_2 ve y_2 ise P_2 noktasının koordinatlarını göstermektedir.

Düzlemsel mesafe olarak da bilinen Manhattan uzaklığı, iki nokta arasındaki uzaklığı koordinat eksenleri üzerindeki çizgi kesitlerini toplayarak hesaplamaktadır. Bir diğer deyişle n boyutlu iki nokta arasındaki farkların mutlak değerlerinin toplamıdır.

Herhangi iki nokta olan P ve Q noktaları arasındaki Manhattan uzaklığı $P=(x_1, x_2, ..., x_n)$ ve $Q=(y_1, y_2, ..., y_n)$ olmak üzere, aşağıdaki eşitliğe göre hesaplanır (Kresse ve Danko, 2012):

$$\left(\sum_{i=1}^{n}|x_i - y_i|\right)$$

Minkovski uzaklığında p=1 alındığında oluşan uzaklık Manhattan uzaklığıdır.



Şekil 6. Manhattan ve Öklid Uzaklığı Hesaplanırken Kullanılan Yollar

Chebyschev uzaklığı (maksimum değer uzaklığı), Minkowski uzaklığının, $n \rightarrow \infty$ olduğu özel durum olup, iki nokta arasındaki farkların mutlak değerlerinin maksimumu olarak tanımlanmaktadır. Herhangi iki nokta, *P ve Q* arasındaki Chebyschev uzaklığı *P=(x1, x2,..., xn)* ve *Q=(y1, y2,..., yn)* olmak üzere, aşağıdaki eşitliğe göre hesaplanır (Xu, Zong ve Yang, 2013).

$$\lim_{p\rightarrow\infty}\left(\sum_{i=1}^{N}|x_i - y_i|^p\right)^{1/p} = max_{i=1}^{n}|x_i - y_i|$$

Aynı zamanda satranç tahtası uzaklığı olarak da bilinir, çünkü satranç oyununda bir kralın bir satranç tahtası üzerindeki bir kareden diğerine gitmesi için gereken minimum hareket sayısı karelerin merkezleri arasındaki Chebyshev mesafesine eşittir (Elektrik ve Elektronik Mühendisleri Enstitüsü (IEEE), 2016).

Komşu Sayısı (k): KNN algoritmasında yeni örneğin sınıfı belirlenirken k parametresine göre karar verilmektedir. k=1 olduğu durumda sadece örneğe uzaklığı en az olan 1 komşunun sınıfı dikkate alınırken, 3 olduğunda 3 örneğin sınıfı dikkate alınmaktadır. Yapılacak oylamaya göre örneğin sınıfına karar verilmektedir (Taşcı ve Onan, 2016).

Ağırlıklandırma: Daha etkili ve doğru sınıflandırmalar yapabilmek adına sınıflandırılacak olan örneğin k komşuluğunda bulunan örnekler uzaklıklarına göre ağırlıklandırılmaktadır. Bu ağırlıklandırma ise d ile temsil edilen uzaklığın $^1/_d$ veya $^1/_{d^2}$ şeklinde alınmasıyla yapılmaktadır (Doad ve Bartere, 2013). Bu şekilde veri seti simetrik olmadığında, yeni örneklerin sınıfları belirlenirken çoğunlukta olan sınıfların baskın olması probleminden kurtulmak amaçlanmaktadır (Coomans ve Massart, 1982).

2.3.3. Lojistik Regresyon Analizi

Neden-sonuç ilişkilerinin araştırıldığı bilimsel araştırmalarda regresyon analizinden faydalanılmaktadır. Yordayıcı ve yordanan değişkenlerin yapısına göre kullanılan regresyon modeli farklılık göstermektedir (Kayri ve Çokluk, 2010). Doğrusal regresyon modellerinde yordayıcı ve yordanan değişkenlerin sürekli değişken olması ve bazı sayıltıların karşılanması gerekirken; LR, bağımlı değişkenin kategorik olduğu durumlarda da kullanılmakta (Mertler ve Vannata, 2005) ve doğrusal regresyondaki sayıltıların kullanılmasını gerektirmemektedir (Kılıç, 2000).

Lojistik regresyonda örneklerin hangi gruba girdiğinin belirlenmesi amacıyla bir regresyon denklemi oluşturulmakta; bu sayede hem sınıflandırılma yapılmakta hem de bağımlı ve bağımsız değişkenler arasındaki ilişkiler ortaya konulmaktadır (Mertler ve Vannata, 2005).

LR, katsayıların kestirilmesi amacıyla en çok olabilirlik yöntemini kullanmaktadır. Bu yaklaşımda amaç bir olayın gerçekleşme olasılığını maksimum yapmaktır (Hair, Black, Babin, Anderson ve Tatham, 2006). Elde edilen model için lojistik regresyonda olasılık, odds oranı ve odds oranının doğal logaritmasının (logit) hesaplanması gerekmektedir. Ψ ya da Exp (β) ile temsil edilen odds oranı bir olayın olma olasılığının olmama olasılığına oranıdır.

$$Odds = \frac{p(x)}{1 - p(x)}$$

Yukarıdaki denklemde p(x) herhangi bir olayın gerçekleşme olasılığının, 1-p(x) ifadesi ise aynı olayın gerçekleşmeme olasılığının matematiksel ifadelerdir (Mertler ve Vannata, 2005).

i. deneğin bağımlı değişkenin kategorilerinden herhangi birinde yer almasına ilişkin kestirilen olasılık $\widehat{Y}_{\text{i}}$,

$$\widehat{Y}_i = \frac{e^u}{1 + e^u}$$

eşitliği ile ifade edilmektedir. Burada e doğal logaritma tabanını, u ise klasik regresyon eşitliğini temsil etmektedir.

$$u = B_0 + B_1X_1 + B_2X_2 + \cdots + B_kX_k$$

Doğrusal regresyon eşitliği herhangi bir örneğin bir grupta bulunmasının diğer grupta bulunması olasılığına bölünmesinin doğal logaritmasıdır. Bu bağlamda doğrusal regresyon eşitliği odds oranının logaritmasını yani logitini oluşturmaktadır ve aşağıdaki eşitlikle ifade edilmektedir:

$$\ln\left(\frac{\hat{Y}}{1-\hat{Y}}\right) = B_0 + B_1X_1 + B_2X_2 + \cdots + B_kX_k$$

Lojistik regresyonda standart ve adımsal yöntemler olmak üzere iki temel yöntemden bahsedilebilir. Standart yöntemde tüm değişkenler regresyon modelinde bir blok halinde yer alır ve her bir blok için parametre kestirimleri yapılır. Adımsal yöntemler ise ileriye doğru yöntemler ve geriye doğru yöntemler olarak sınıflandırılmaktadır. İleriye doğru yöntemlerde sabit terimle başlanan modele diğer değişkenler tek tek eklenerek analize devam edilir. Geriye doğru yöntemlerde ise modele tüm yordayıcı değişkenler dâhil edilir ve model geliştirilirken katkısı en az olan değişkenlerden başlanarak yordayıcı değişkenlerin elenmesiyle analiz devam eder (Field, 2005).

LR, bağımsız değişkenlerin dağılımına ilişkin herhangi bir şartın yerine getirilmesini gerektirmez. Bununla birlikte LR analizini kullanmak için bazı önkoşulların yerine getirilmesi de gerekmektedir:

- Analize giren değişkenler arasında yüksek korelasyon yani çoklu bağlantı bulunmamalıdır.
- Lojistik model uç değerlere duyarlı olduğundan uç değerler konusunda dikkatli olunmalı ve bunlar elenmelidir.

- Yordayıcı değişkenlerin sayısı gruptaki örnek sayısına göre fazlaysa yani bazı hücrelerde çok az örnek bulunuyorsa bu durum hataya neden olacak ve analizin gücünü düşürecektir (Tabachnick ve Fidell, 2005).

2.3.4. NB Algoritması

NB istatistiksel sınıflandırma algoritmalarındandır. Formülasyonları, belirli bir örneğin belirli bir sınıfa ait olma olasılığını hesaplamak amacıyla oluşturulmuştur. NB sınıflama algoritması; Bayes teorisine dayanan, hem tahmin edici hem de tanımlayıcı işlemlerde kullanılabilen, kolay anlaşılan ve hızlı çalışan bir yöntemdir. Bu algoritma tüm değişkenlerin birbirinden bağımsız ve hepsinin aynı öneme sahip olduğu varsayımlarına dayanan bir algoritmadır (Han vd., 2011; Özkan, 2016; Zhang, 2004). NB algoritması, belirli bir sınıftaki bir özellik değerinin etkisinin, diğer özniteliklerin değerlerinden bağımsız olduğunu varsayar. Bu varsayım sınıf koşullu bağımsızlık olarak adlandırılır (Han vd., 2011).

Bağımsızlık varsayımı NB sınıflandırıcısı için bazı avantajlar ve dezavantajlar ortaya çıkarmaktadır. Bağımsızlık varsayımı durumu gerçekte çok nadir görülse de yani bu durum çoğu zaman gerçek dışı olsa da NB algoritmasının sınıflamadaki başarısı ve diğer bazı sınıflama algoritmalarından üstünlüğü çeşitli çalışmalarla ortaya konulmuştur. Bağımsızlık varsayımı her bir değişkenin tek tek öğrenilmesine olanak vermektedir. Böylece, çok değişkene sahip olan verilerde bile sınıflama işleminin hızlı olmasına olanak sağlamaktadır. Fakat gerçek durumlarda değişkenlerin birbirinden bağımsız olması varsayımı çoğunlukla sağlanmadığından bu durum aynı zamanda gerçek dışılığı da beraberinde getirmektedir (Aydoğan, 2008; Zhang, 2004).

Bu algoritma Bayes teoremine dayanır ve NB algoritmasının kullandığı bağıntı aşağıda verilmiştir. Bu bağıntı yardımıyla algoritmanın çalışma prensibi bir örnek üzerinden açıklanacaktır:

$$\mathrm{P(C|X)} = \frac{\mathrm{P(X|C)}.\mathrm{P(C)}}{\mathrm{P(X)}}$$

Burada $P(C|X) = P(X_1|C).P(X_2|C).\ldots.P(X_n|C).P(C)$ şeklinde hesaplanmaktadır (Özkan, 2016). Formülde $\mathrm{P(C|X)}$ X olayının gerçekleşmesi durumunda C olayının ortaya çıkma olasılığını; $\mathrm{P(X|C)}$ ise C olayının gerçekleşmesi

durumunda X olayının ortaya çıkma olasılığını göstermektedir. P(C), C olayının önsel (posterior) olasılığı ve P(X) ise X olayının önsel olasılığıdır.

Tablo 1. *Bir Veri Seti Örneği*

		Başarı Durumu			
		Başarılı		Başarısız	
		Sayı	Olasılık	Sayı	Olasılık
Anne Eğitim Durumu	İlköğretim	1	1/5	2	2/3
	Lise	3	3/5	0	0
	Lisans	1	1/5	1	1/3
Baba Eğitim Durumu	İlköğretim	0	0	2	2/3
	Lise	3	3/5	1	1/3
	Lisans	2	2/5	0	0
Cinsiyet	Kız	3	3/5	1	1/3
	Erkek	2	2/5	2	2/3

Tablo 1'deki veri setinde başarılı ve başarısız olmak üzere iki sınıf bulunmaktadır. 24 örnekli bu veri setinde örneklerin 15 tanesi başarılı, 9 tanesi ise başarısız olarak sınıflandırılmıştır. Veri setinde değişkenler anne eğitim durumu, baba eğitim durumu ve cinsiyet olarak belirlenmiştir. Anne eğitim durumu lisans, baba eğitim durumu lise ve cinsiyeti erkek olan bir örneğin ait olduğu sınıfı NB algoritması aşağıdaki şekilde belirlemektedir (Özkan, 2016):

C_1 başarılı ve C_2 başarısız olan sınıfı X ise sınıflandırılacak örneğin özelliklerini ifade etmektedir. $P(X|C_1).P(C_1)$ ve $P(X|C_2).P(C_2)$ olasılıkları hesaplanmalıdır burada sırasıyla ilk olasılık sınıflandırılacak olanörneğin başarılı grubuna, ikinci olasılık ise başarısız grubuna dahil edilme olasılığıdır.

X'in tüm değerleri için ayrı ayrı koşullu olasılıkları hesaplanmıştır:

$$P(x_1|C_1) = P(anne\ eğitim\ durumu = lisans|başarı\ durumu = başarılı) = \frac{1}{5}$$

$$P(x_2|C_1) = P(baba\ eğitim\ durumu = lise|başarı\ durumu = başarılı) = \frac{3}{5}$$

$$P(x_3|C_1) = P(cinsiyet = erkek|başarı\ durumu = başarılı) = \frac{3}{5}$$

$$P(X|C_1) = \frac{1}{5}.\frac{3}{5}.\frac{3}{5} = \frac{9}{125}$$

$$P(C_1) = P(başarı\ durumu = başarılı) = \frac{5}{8}$$

$P(X|C_1).P(C_1) = 0.045$ olarak hesaplanır.

Aynı şekilde başarısız sınıfı için de benzer işlemler yapıldığında $P(X|C_2).P(C_2) = 0{,}014$ olarak elde edilir. 0.045 daha büyük bir olasılık olduğundan elimizdeki örnek başarılı sınıfına dâhil edilmelidir (Özkan, 2016).

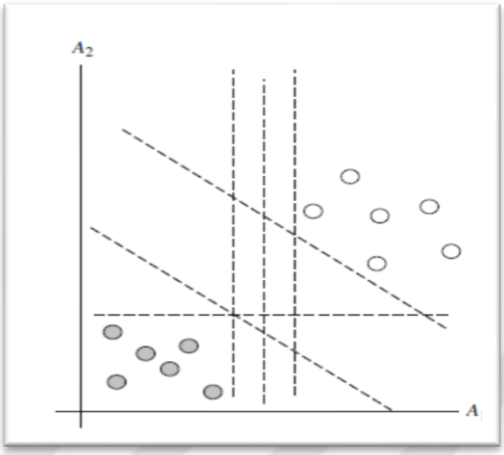
2.3.5. DVM

Orijinal eğitim verilerini daha yüksek bir boyuta dönüştürmek için doğrusal olmayan bir dönüşüm kullanan ve bu yeni boyutta en uygun ayırıcı hiperdüzlemi arayan oldukça basit ve etkili bir sınıflandırma algoritmasıdır. Hiperdüzlem, bir sınıfı diğerinden ayıran bir karar sınırını ifade eder. Yeterince yüksek bir boyutta uygun doğrusal olmayan bir dönüşüm ile iki sınıftan alınan veriler her zaman bir hiperdüzlem ile ayrılabilir. DVM algoritması bu hiperdüzlemi bulmak için destek vektörlerini ve bu destek vektörleri tarafından tanımlanan marjinleri kullanır (Han vd., 2011).

DVM algoritması hakkında ilk çalışma 1992 yılında Vladimir Vapnik ve arkadaşları Bernard Boser ve Isabelle Guyon tarafından sunulmuştur. En hızlı DVM algoritmasında bile eğitim süresi çok uzundur buna rağmen karmaşık ve doğrusal olmayan karar sınırlarını modelleme yetenekleri sebebiyle kullanılmaktadırlar. Diğer algoritmalara göre ezberlemeye daha az eğilimli oldukları düşünülmektedir. DVM algoritması sınıflandırma yanında tahminleme için de kullanılmaktadır (Han vd., 2011).

Verilerin doğrusal bir şekilde ayrılabildiği durumlarda DVM algoritması şu şekilde çalışmaktadır. İki sınıflı bir problemde veri setinin D ile ve D'nin ise ; (x_1,y_1), (x_2,y_2), … , $(x_{|D|},y_{|D|})$ ile temsil edildiği varsayılmaktadır. x_i eğitim verisinde bulunan sınıfların orijinal sınıf etiketlerini ve y_i ise sınıfların – 1 ya da +1 şeklinde ifade edildiği sınıfları göstermektedir. A_1 ve A_2 ile temsil edilen iki değişkene sahip bir veri setinin bulunduğu ve bu veri setindeki örneklerin doğrusal olarak (bir doğru yardımıyla) ayrılabildiği varsayıldığında sınıfları birbirinden ayırabilmek için çizilebilecek sonsuz sayıda çizgi vardır. Çizilebilecek en optimal ya da en uygun çizginin bulunması yani oluşturulan modelin test verilerinde minimum hata oranına sahip olması beklenmektedir. Ele aldığımız veri seti üç boyutlu olursa yani üç değişken bulunursa bu

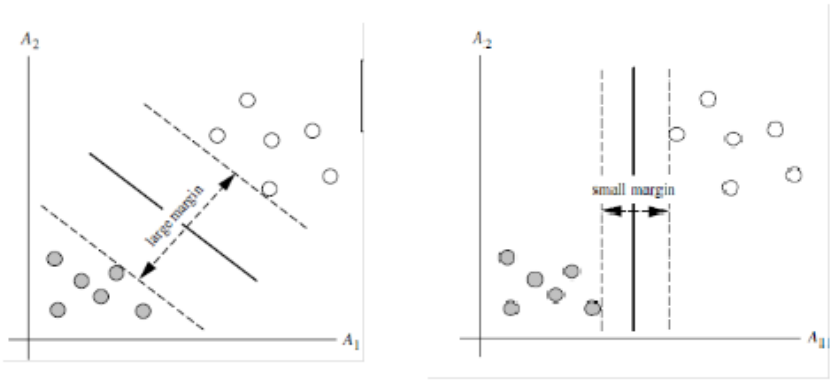
durumda sınıfları en iyi ayıran düzlemin bulunması gerekir. N boyutlu bir veri seti için ise sınıfları en iyi ayıran hiperdüzlem bulunmalıdır. Hiperdüzlem terimi girdi değişkenlerin sayısına bakılmaksızın aranan karar sınırının belirlenmesi için kullanılmaktadır (Han vd., 2011).



Şekil 7. İki Boyutlu Doğrusal Olarak Ayrılabilen Veri Seti İle Mümkün Hiperdüzlemler

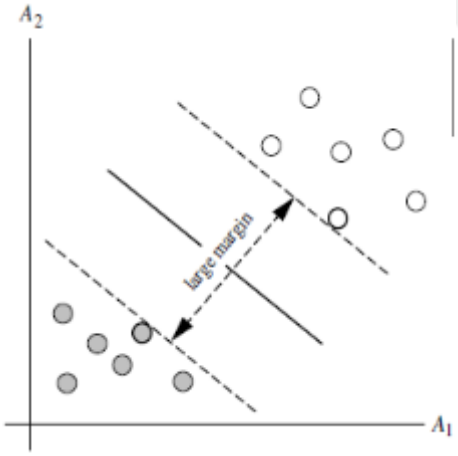
Şekil 7'de verilen veri setinde A_1 ve A_2 isimli iki değişken bulunmakta ve bu veri seti doğrusal olarak ayrılabilmektedir. Veri setini sınıflandırmada kullanılabilecek olası hiperdüzlemler de kesikli çizgilerle gösterilmiştir.

DVM algoritması sınıfları birbirinden ayırırken maksimum marjinli hiperdüzlemi (MMH) bulmaya çalışır. Şekil 8'de A_1 ve A_2 değişkenlerine sahip, iki boyutlu ve iki sınıflı bir veri setindeki sınıfları ayırabilmek için çizilebilecek küçük ve büyük marjinler gösterilmiştir. Olası birçok hiperdüzlem olmasına bağlı olarak, belirlenebilecek olası marjinler de söz konusudur. Her iki hiperdüzlem de verilen örnekleri doğru bir şekilde sınıflandırabilmektedir. Daha büyük kenar boşluğu olan hiperdüzlem daha iyi sınıflama yaptığından DVM algoritması öğrenme ve test aşamalarında en büyük marjine sahip olan hiperdüzlemi arar. Elde edilen bu hiperdüzleme MMH adı verilmektedir. MMH'deki marjin, sınıflar arasındaki en büyük ayrımı verir. Hiperdüzlemden kenar boşluğunun bir tarafına olan en kısa mesafe kenar boşluğunun diğer tarafına olan en kısa mesafeye eşittir ve burada marjinin kenarları hiperdüzleme paraleldir. MMH'de bu uzaklık aslında her iki sınıfın da en yakın örneklerinin birbirine en kısa uzaklığıdır.



Şekil 8. Çizilebilecek Küçük ve Büyük Ölçekte Marjinler ve Hiperdüzlemleri

Ayırıcı bir hiperdüzlem $\vec{W} \cdot \vec{X} + b = 0$ fonksiyonuyla ifade edilir burada $\vec{W}$, n değişkenlerin sayısını belirtmek üzere $\vec{W} = \{w_1, w_2, \ldots, w_n\}$ şeklinde ifade edilen ağırlık vektörüdür ve karar fonksiyonun normalini, $\vec{X}$ ifadesi bu doğru üzerinde bulunan noktaları, b ise eğilim değerini göstermektedir. Amaç $\vec{W}$ ve b' yi eğitim verileri yardımıyla bulmaktır, yani sistemi eğitmektir.



Şekil 9. Destek Vektörleri ve Maksimum Marjine Sahip Hiperdüzlem

Şekil 9'da A_1 ve A_2 ile isimlendirilen iki boyutu bulunan bir test veri seti örneği bulunmaktadır. Görselleştirmenin kolay olması adına veri seti iki boyutlu seçilmiştir. $\vec{X}$ = (x_1, x_2) ile ifade edildiğinde x_1 ve x_2, X için sırasıyla A_1 ve A_2 öznitelik değerleridir. b ek bir ağırlık olarak düşünülürse w_0, ayırıcı hiperdüzlemi w_0 + w_1x_1 + w_2x_2 = 0 olarak yazılır.

Böylece, ayırıcı hiperdüzlemin üzerinde bulunan herhangi bir nokta $w_0 + w_1x_1 + w_2x_2 > 0$ eşitsizliğiyle; benzer şekilde, ayrılan hiperdüzlemin altında bulunan herhangi bir nokta ise $w_0 + w_1x_1 + w_2x_2 < 0$ eşitsizliği ile ifade edilir.

Ağırlıklar ayarlandığında marjinin sınırlarını belirleyen hiperdüzlemler H_1 ve H_2 aşağıdaki şekilde yazılabilir:

$$H_1: w_0 + w_1x_1 + w_2x_2 \geq 1 \; y_i = +1 \text{ için}$$

$$H_2: w_0 + w_1x_1 + w_2x_2 \leq -1 \; y_i = -1 \text{ için}$$

H_1'in üzerinde olan herhangi bir örnek +1 sınıfına ve H_2'nin üzerine veya altına düşen herhangi bir örnek -1 sınıfına aittir. 1 ve 2 deki eşitlikleri birleştirilirse her i için

$$y_i(w_0 + w_1x_1 + w_2x_2) \geq 1 \quad \text{denklemi elde edilir.}$$

H_1 ya da H_2 hiperdüzlemlerinin üzerine düşen herhangi bir örnek, $y_i(w_0 + w_1x_1 + w_2x_2) \geq 1$ denklemi ile sınıflandırılabilir ve destek vektörleri olarak adlandırılır. Destek vektörleri sınıfları eşit olarak ayıran MMH'ye yakındırlar. Şekil 9'da destek vektörleri daha kalın bir sınır ile çevrelenmiş olarak gösterilmiştir. Temel olarak destek vektörleri sınıflanması en zor olan ve sınıflandırmaya ilişkin en çok bilgiyi veren örneklerdir.

Ayırıcı hiperdüzlemden herhangi bir noktaya olan uzaklık $1/\|\vec{W}\|$ ye eşittir burada $\|\vec{W}\|$, $\vec{W}$'nin Öklid formudur yani $\sqrt{(\vec{W}.\vec{W})}$ dir. Tanım olarak bu, H_2'nin üzerindeki herhangi bir noktadan ayırıcı hiperdüzleme olan uzaklığa karşılık geldiğinden maksimum marjin mesafesi $2/\|\vec{W}\|$ ye eşittir. DVM algoritması MMH ve destek vektörlerini bulmak için $y_i(w_0 + w_1x_1 + w_2x_2) \geq 1$ denklemini bazı matematiksel kuralları kullanarak yeni bir forma çevirir ve bu denklemi kısıtlanmış (konveks) ikinci derece optimizasyon problemine dönüştürürek işlem yapar.

Sınıflandırıcının öğrenmesi sırasındaki karmaşıklık, verilerin boyutundan ziyade destek vektörlerinin sayısı ile karakterize edilir. Bu nedenle, DVM'ler diğer bazı yöntemlere göre aşırı öğrenmeye (ezberlemeye) daha az eğilimlidir. Destek vektörleri temel ya da kritik örneklerdir ve bunlar karar sınırının en yakınında bulunurlar. Diğer tüm eğitim örnekleri çıkarılsa ve eğitim tekrar edilse, aynı hiperdüzlem bulunacaktır.

Ayrıca, bulunan destek vektörlerinin sayısı, veri boyutundan bağımsız olan DVM sınıflandırıcısının beklenen hata oranı üzerine hesaplanması için kullanılabilir. Az sayıda destek vektörüne sahip bir DVM, verilerin boyutunun yüksek olduğu durumlarda bile iyi bir genelleştirmeye sahip olabilir.

Veriler doğrusal olarak ayrılabilir değilse yani sınıfları ayıracak düz bir çizgi bulunmuyorsa çalıştığımız doğrusal DVM'ler burada uygun bir çözüm bulamayacaktır. Doğrusal DVM'ler için tarif edilen yaklaşım, doğrusal olarak ayrılmayan verilerin sınıflandırılması için genişletilebilmektedir. Bu gibi DVM'ler, giriş alanında doğrusal olmayan karar sınırlarını (yani, doğrusal olmayan hiper-yüzeyleri) bulabilirler.

Doğrusal DVM'lere uygun olan lineer yaklaşım genişletilerek doğrusal olmayan bir DVM yaklaşımları elde edilmektedir. Bu süreçte iki ana adım bulunmaktadır. İlk adımda, orijinal girdi verileri doğrusal olmayan bir dönüşüm kullanılarak daha yüksek boyutlu bir alana dönüştürülmektedir. Bu adımda bazı ortak doğrusal olmayan dönüşümler kullanılmaktadır. Veriler daha yüksek bir alana dönüştürüldükten sonra, ikinci adımda bu yeni alanda lineer ayırıcı bir hiperdüzlem aranır. Süreç tekrar lineer DVM formülasyonu kullanılarak çözülebilen ikinci dereceden bir optimizasyon problemi ile sonuçlanır. Yeni uzayda bulunan maksimum marjinal hiperdüzlem, orijinal uzayda lineer olmayan bir ayırıcı hiperyüzeye karşılık gelmektedir.

Daha yüksek boyuttaki bir uzaya doğrusal olmayan dönüşüm yapmaya gerek duymadan maksimum marjine sahip ayırıcı hiperdüzlemi bulabilmek için çekirdek fonksiyonları kullanılır. Böylece tüm hesaplamalar potansiyel olarak çok daha düşük boyuta sahip olan giriş uzayında yapılabilir. Kullanılabilecek çekirdek fonksiyonları aşağıda verilmektedir:

Polinom çekirdek fonksiyonu: h: $K(\vec{X}_i, \vec{X}_j) = (\vec{X}_i \cdot \vec{X}_j + 1)^h$

Gaussian radyal tabanlı çekirdek fonksiyonu: $K(\vec{X}_i, \vec{X}_j) = e^{-\|\vec{x}_i - \vec{x}_j\|^2 / 2\sigma^2}$

Sigmoid çekirdek fonksiyonu: $K(\vec{X}_i, \vec{X}_j) = \tan h(\kappa \vec{X}_i \cdot \vec{X}_j - \delta)$

Bunların her biri (orijinal) giriş alanında farklı bir doğrusal olmayan sınıflandırıcı ile sonuçlanır. Hangi kabul edilebilir çekirdeğin DVM algoritmasının

performansını maksimum yapacağı konusunu belirlemek için altın bir kural yoktur. Pratikte seçilen çekirdek genellikle sonuç doğruluğunda büyük bir fark yaratmamaktadır.

DVM sınıflandırıcıları çok sınıflı veriler için birleştirilebilir. Verilen m sınıflı bir veri için en etkili ve basit yaklaşım, her sınıf için bir tane olmak üzere m tane sınıflayıcıyı eğitmektir (j sınıflandırıcısı j sınıfı için pozitif bir değer ve geri kalanı için negatif bir değer vermeyi öğrenir). Bir test örneği, en büyük pozitif mesafeye karşılık gelen sınıfa atanır.

DVM'lerle ilgili çözülmesi gereken problemlerden biri DVM'lerin eğitim ve test etme hızını artırmaktır böylelikle bu algoritma çok büyük veri kümeleri (milyonlarca destek vektörünü içeren) için daha uygun bir seçenek haline gelebilecektir. Ayrıca belirli bir veri seti için hangi çekirdeğin daha iyi olabileceğini ortaya koymak ve çok sınıflı veriler için daha verimli yöntemler geliştirmek de bu algoritma ile ilgili diğer problemler olarak karşımıza çıkmaktadır (Han vd., 2011).

2.4. Performans Metrikleri

Kullanılan algoritmaların performansları belirlenirken ve değerlendirilirken bazı metriklerden yararlanılır. Algoritmaların performans değerlendirmesi için, MUC (Message Understanding Conference) tarafından tavsiye edilen en temel metrikler; doğruluk oranı, kesinlik, duyarlılık ve F ölçütüdür. Ayrıca ROC alanı değeri ve kappa istatistiği de kullanılabilecek metrikler arasındadır (Landis ve Koch, 1977). Bu değerlerin hesaplanabilmesi amacıyla Tablo 2'deki gibi karmaşıklık matrisi oluşturulur. Karmaşıklık matrisi, hem algoritmanın öngörüsünü hem de gerçek sınıf bilgisini içermektedir(Coşkun ve Baykal, 2011).

Tablo 2. *Karmaşıklık Matrisi*

		Öngörülen Sınıf	
		Sınıf 1	Sınıf 0
Doğru Sınıf	Sınıf 1	a	b
	Sınıf 0	c	d

a:Gerçek pozitif (TP) c: Yanlış pozitif (FP)

b: Yanlış negatif (FN) d:Gerçek negatif(TN)

FP 1. Tip hata, FN de 2. tip hata olarak adlandırılır (Güner, 2008).

Doğruluk – Hata oranı: Model başarımının ölçülmesinde kullanılan en popüler ve basit yöntem, modele ait doğruluk oranıdır. Doğru sınıflandırılmış örnek sayısının (TP+TN), toplam örnek sayısına (TP+TN+FP+FN) oranı olarak tanımlanmıştır.

Doğruluk metriği, bütün hata tiplerini dikkate alarak, pozitif ve negatif örnekleri aynı derecede önemsemeyi sağlamakta ve sınıflandırıcının toplam performansını değerlendirmeye yardımcı olmaktadır. Fakat doğruluk ölçütü, veri kümesinde dengesiz dağılım var ise yeterli olmamaktadır (Akbulut, 2006).

Hata oranı ise bu değerin 1'e tamlayanıdır. Diğer bir ifadeyle yanlış sınıflandırılmış örnek sayısının (FP+FN), toplam örnek sayısına (TP+TN+FP+FN) oranıdır.

$$Doğruluk = \frac{TP + TN}{TP + TN + FP + FN}$$

$$Hata\ oranı = \frac{FP + FN}{TP + TN + FP + FN}$$

Kesinlik: Sınıfı 1 olarak tahminlenmiş gerçek pozitif örnek sayısının, sınıfı 1 olarak tahminlenmiş tüm örnek sayısına oranıdır.

$$Kesinlik = \frac{TP}{TP + FP}$$

Duyarlılık: Doğru sınıflandırılmış pozitif örnek sayısının toplam pozitif örnek sayısına oranıdır.

$$\text{Duyarlılık} = \frac{\text{TP}}{\text{TP+FN}}$$ (Coşkun ve Baykal, 2011).

Bu durumda kullanılan kesinlik ve duyarlılık ölçütleri, sırasıyla, pozitif örneklerin negatif olarak sınıflandırılmasından oluşan hatalar ile negatif örneklerin pozitif olarak sınıflandırılmasından oluşan hataları belirtmektedirler (Akbulut, 2006).

F-Ölçütü: Kesinlik ve duyarlılık ölçütleri tek başlarına anlamlı bir karşılaştırma yapmamızda yeterli olmamaktadır. Her iki ölçütü beraber değerlendirmenin daha doğru sonuçlar vereceği düşüncesiyle F-ölçütü tanımlanmıştır. F-ölçütü, kesinlik ve duyarlılık metriklerinin harmonik ortalamasıdır (Coşkun ve Baykal, 2011).

$$\mathrm{F}-\text{ölçütü}=\frac{2\times \text{duyarlılık}\times \text{kesinlik}}{\text{duyarlılık}+\text{kesinlik}}$$

Kappa istatistiği: Veri madenciliğinde kappa istatistiği, sınıflandırma algoritmasının doğruluğu ile tesadüfî tahminde bulunan rastgele bir sınıflandırma algoritmasının doğruluğunun karşılaştırılması amacıyla kullanılmaktadır. Kappa değerinin 1'e yaklaşması doğruluk değerinin bağımlı değişkenin sınıflarındaki örnek sayısındaki yığılmaya bağlı olmadığını yani ortaya çıkan başarının rastgele olmadığını göstermektedir (Kalıpsız ve Cihan, 2015).

Altman (1990)'a göre kappa istatistik değerinin 0 ile 0.20 arasında olması zayıf, 0.20 ile 0.40 arasında olması makul düzeyde, 0.40 ile 0.60 arasında arasında orta düzeyde, 0.60 ile 0.80 arasında olması iyi düzeyde ve 0.80 ile 1 arasında ise mükemmel düzeyde bir uyum olduğunu göstermektedir.

Sınıflandırma yüzdesi ile kappa arasındaki temel fark doğru sınıflandırmaları puanlamaları arasındadır. Sınıflandırma yüzdesinde tüm sınıflar toplu olarak dikkate alınır ve doğru sınıflandırılan tüm örnekler puanlanır. Kappa değerinde ise tüm sınıflar bağımsız olarak düşünülerek puanlanır ve sonuçlar birleştirilir. Kappa değeri bu sebeple her bir sınıfta bulunan farklı sayıdaki örneklerin sebep olduğu rastgeleliğe daha duyarlıdır (Galar vd.,2011).

2.5. Hata Ölçüleri

Hata bir örneğin tahmin edilen değeri ile gerçek değeri arasındaki fark olarak tanımlanmaktadır. Hataların bir kayıp fonksiyonu ile temsil edildiği düşünülürse tüm örnekler üzerinden elde edilen hataların ortalaması test hata ortalaması olarak tanımlanır.

$$Ortalama\ Mutlak\ Hata\ (MAE)=\frac{\sum_{i=1}^{d}|y_i-y_i'|}{d}$$

$$Ortalama\ Karesel\ Hata=\frac{\sum_{i=1}^{d}(y_i-y_i')^2}{d}$$

Denklemlerde bulunan d; örnek sayısını, y_i herhangi bir örneğin gerçek değerini ve y_i' ise tahmini değerini göstermektedir.

Ortalama karesel hata aykırı değerlerin varlığını abartırken ortalama mutlak hata (MAE) böyle değildir. Ortalama karesel hatanın karekökünü alırsak, ortaya çıkan hata ölçüsüne ortalama karesel hatanın karekökü (RMSE) denir. Bu hata türü, ölçülen hatanın, tahmin edilen miktarla aynı büyüklükte olmasına izin vermesi bakımından önemlidir.

Hata $\bar{y}$ cinsinden göreli olarak belirlenmek istendiğinde kullanılan bazı başka hata ölçüleri de bulunmaktadır. Toplam kayıp, her zaman ortalamayı tahmin etmekten kaynaklanan toplam zarara bölünerek normalize edilebilir. Bu göreli hata ölçüleri rölatif mutlak hata (RAE) ve göreli karesel hata (RSE) olarak adlandırılır:

$$RAE = \frac{\sum_{i=1}^{d}|y_i - y_i'|}{\sum_{i=1}^{d}|y_i - \bar{y}|}$$

$$RSE = \frac{\sum_{i=1}^{d}(y_i - y_i')^2}{\sum_{i=1}^{d}(y_i - \bar{y})^2}$$

Denklemlerde $\bar{y}$ eğitim verilerindeki y_i değerlerinin ortalamasıdır. Rölatif karesel hatanın karekökü (RRSE) değerinin elde edilmesi için RSE değerinin karekökü alınır. Böylece ortaya çıkan hata ile öngörülen miktar aynı büyüklükte olur. Pratikte, hata ölçüsü seçimi tahmin model seçimini büyük ölçüde etkilememektedir (Han vd., 2011).

2.6. İlgili Araştırmalar

Lorena, De Carvalho ve Gama (2008) çalışmalarında ikili sınıflandırıcıların çok sınıflı problemlerin çözümü için kullanılması üzerinde durmuşlardır. Çalışmalarının odak noktası çok sınıflı problemi iki sınıflı alt kümelere ayırmak ve nihai karar için de iki sınıflı alt problemlerin çıktılarını birleştirmektir. Çalışmalarında ayrıştırma teknikleri kullanılmazsa ikili sınıflandırıcıların çok sınıflı veriler için tekrar formüle edilmesi gerekliliğinden ve bu yöntemlerin kolay olmadığından ayrıca ayrıştırma tekniklerinin son yıllarda oldukça popüler olduğundan bahsetmişlerdir. Çalışmada bu alandaki temel gelişmelere ve çok sınıflı problemler için uyarlanabilecek tüm ayrıştırma tekniklerinin kullanımına değinilmiştir. Ayrıca çalışmada ikili sınıflandırıcıların çıktılarını birleştirmek için kullanılan tekniklerden de bahsedilmiştir.

Taruna ve Pandey (2014) "öğrencilerin akademik performansını tahminlemek için sınıflandırma tekniklerinin deneysel bir analizi" isimli çalışmalarında öğrencilerin

notlarını tahminlemek amacıyla dört sınıflı bir veri seti üzerinde beş algoritmanın performanslarını kıyaslamışlar ve en başarılı algoritmaların KNN, karar ağacı ve Bayes Net algoritmaları olduğunu belirtmişlerdir.

Sen, Islam, Murase ve Yao (2016) "çok sınıflı sınıflandırmada boosting ve aşırı örnekleme metotlarıyla ayrıştırma" isimli çalışmalarında ayrıştırma tekniğini boosting ve aşırı örnekleme metotlarıyla birleştirmişler ve yeni bir teknik önermişlerdir. Önerdikleri tekniği OVA stratejisine dayalı olarak geliştirmişlerdir. Geliştirilen ve kısaltması BBO şeklinde olan yöntem OVA stratejisinde ortaya çıkan dengesiz sınıf dağılımı problemini çözmek için aşırı örnekleme kullanmaktadır. Bu yeni tekniği birçok algoritma ile test etmişler ve muadilleriyle karşılaştırdıklarında geliştirdikleri yöntemin daha iyi performans gösterdiğini kanıtlanmışlardır.

Galar, Fernández, Barrenechea, Bustince ve Herrera (2011) yaptıkları çalışmada C4.5, knn, ripper gibi bazı çok bilinen sınıflandırma algoritmalarını kullanarak OVO ve OVA stratejileri çerçevesinde farklı toplulaştırma (aggregation) tekniklerini kıyaslamışlardır. Sonuç olarak DVM ve PFDC algoritmalarının OVO stratejisi kullanıldığında daha iyi çalıştığını aynı şekilde C4.5 ve Ripper algoritmalarının da OVO şeması altında orijinal sınıflandırıcıya göre daha yüksek performans gösterdiğini, KNN algoritması kullanılmasının hem OVO hem OVA stratejilerinde orijinal algoritmanın performansını artırdığını fakat bunun istatistiksel olarak anlamlı olmadığını tespit etmişlerdir.

Polat ve Güneş (2007) "çok sınıflı sınıflandırma problemlerinde C4.5 ve OVA yaklaşımına dayalı yeni hibrit bir model" isimli çalışmalarında çok sınıflı problemler için OVA stratejisini ve C4.5 karar ağacı yöntemini birleştirerek sınıflandırma başarısını büyük oranda artırmışlardır. İlk aşamada C4.5 algoritmasını üç veri seti üzerine direkt uygulamışlar ve sırasıyla %84.48, %88.79 ve %80.11 başarı oranları elde etmişlerdir. İkinci aşamada ise veri setleri üzerinde, geliştirilen hibrit yöntemi denemişler; başarı oranlarını %96.71, %95.18 ve %87.95 olarak belirlemişlerdir.

Adnan ve Islam (2015) çalışmalarında bir karar ormanı oluşturma stratejisi olan rastgele ormanlar algoritmasını OVA şemasında temel algoritma olarak kullanarak farklı veri setlerinde kıyaslama yapmışlardır ve rastgele ormanlar algoritmasının OVA şemasında etkili bir yöntem olduğunu belirtmişlerdir.

Caruana ve Niculescu-Mizil (2006) çalışmalarında on tane denetimli öğrenme algoritmasını karşılaştıran büyük ölçüde deneysel bir çalışma tasarlamışlardır. Ayrıca modelleri Platt ölçeklendirmesi (scaling) ve İzotonik Regresyon ile kalibre etmenin algoritmaların performansı üzerindeki etkisini de incelemişlerdir. Platt'in ölçekleme yöntemi veya İzotonik Regresyon, algoritmaların performansı üzerinde oldukça etkilidir. Kalibrasyon; boosted trees, boosted stumps, DVM ve NB algoritmalarının performanslarını önemli ölçüde artırırken; RF (Rastgele Ormanlar) algoritmasının performansını küçük ama dikkate değer ölçüde artırmıştır. Buna karşılık Sinir ağları, bagged trees, hafızaya dayalı metotlar (memory based methods) ve LR algoritmalarının performansları önemli ölçüde artmamıştır. Kullanılan sekiz metriğin tümünde en iyi performans gösteren algoritma kalibre edilmiş boosted trees algoritması iken bu algoritmayı sırasıyla random forests (rastgele ormanlar), kalibre edilmemiş bagged trees, kalibre edilmiş DVM ve kalibre edilmemiş sinir ağları algoritmaları izlemektedir. En az başarı gösteren algoritmalar ise NB, LR, karar ağacı ve boosted stumps algoritmalarıdır.

Varpa, Joutsijoki, Iltanen ve Juhola (2011) çalışmalarında otonörolojik çok sınıflı bir veriyi OVO ve OVA stratejileri kullanarak iki sınıflı alt problemlere bölmüş daha sonra KNN ve DVM algoritmalarını bu alt problemlere uygulamışlardır. Ayrıştırma stratejileri kullanıldığında yeniden formüle etme yöntemine kıyasla, algoritmaların performanslarının daha iyi olduğunu ve genel olarak OVO stratejisinin OVA stratejisinden daha iyi çalıştığını ortaya koymuşlardır. En yüksek sınıflandırma doğruluğunun ise OVO stratejisi ile birlikte kullanılan KNN algoritmasıyla sağlandığını ifade etmişlerdir.

King, Feng ve Sutherland (1995) çalışmalarında algoritmaları StatLog projesindeki çalışmalar üzerinden karşılaştırılmışlardır. Bu algoritmalar sembolik öğrenme algoritmalarından, istatistiksel algoritmalardan ve sinir ağları algoritmalarından seçilmiştir. 12 veri setinden 5 tanesi görüntü analizi, 3 tanesi tıp, 2 tanesi mühendislik ve 2 tanesi finans verilerine aittir. Uygulama sonunda algoritmaların performanslarının veri setine bağlı olduğu bulunmuştur. Veri seti uç dağılımlara sahipse ve değişken sayısının %38 kadarı ikili ve kategorik değişkenler ise bu tür veri setlerinde sembolik öğrenme algoritmalarının daha başarılı olduğu ifade edilmiştir.

Wu vd. (2008) çalışmalarında IEEE tarafından düzenlenen Uluslararası Veri Madenciliği Konferansı'nda belirlenmiş olan en popüler veri madenciliği algoritmalarını ve bunların çalışma prensiplerini tanıtmışlar, etkilerini tartışmışlar ve algoritmalar ile ilgili güncel araştırmalar ile ileride yapılabilecek araştırmaları gözden geçirmişlerdir. Belirlenen ve gözden geçirilen 10 veri madenciliği algoritması; C4.5, KNN, DVM, Apriori, EM, PageRank, AdaBoost, NB, ve CART algoritmaları olarak sıralanabilir.

Sabzevari, Soleymani ve Noorbakhsh (2007) çalışmalarında bir bankanın gerçek kredi puanlama verileri üzerinde çeşitli veri madenciliği modellerini karşılaştırmışlardır. Bu modeller klasik istatistikteki probit ve LR analizleri ile, veri madenciliğindeki CART, bagging ve MARS algoritmalarıdır. Oldukça küçük bir veri kümesi üzerinde yaptıkları bu çalışma sonucunda istatistiksel modeller arasından LR ve veri madenciliği modelleri arasından da bagging algoritmasının daha başarılı sonuçlar ürettiğini ortaya koymuşlardır. Ayrıca çalışmalarına özellik seçimi ile ilgili bir bölüm de dahil etmişler ve bu sayede özellik uzayını daraltmak için otomatik bir teknik önermişlerdir.

Rajavarman ve Rajagopalan (2007) çalışmalarında sınıflandırma problemleri için formüle ettikleri genetik algoritma yaklaşımını geleneksel veri madenciliği yaklaşımlarından olan karar ağacı, sinir ağları ve NB ile karşılaştırmışlardır. Üç farklı veri seti ile yaptıkları değerlendirme sonucuna göre genetik algoritmanın hem standart dağılım bakımından hem de tahminleme başarısı bakımından diğer algoritmalardan daha üstün performans gösterdiğini ifade etmişlerdir.

Bin Othman ve Yau (2007) çalışmalarında WEKA'yı kullanarak göğüs kanseri verileri üzerinde Bayes Network, Radyal Tabanlı Fonksiyon (Radial Basis Function), Pruned Tree,(budanmış ağaç) Single Conjuctive Rule Learner ve Nearest Neighbours (en yakın komşular) algoritmalarını karşılaştırmışlardır. Uygulama sonrasında algoritmalar sınıflandırma doğruluğu bakımından RBF, budanmış karar ağacı (DT with prunning), single conjuctive learner ve KNN olarak sıralanmışlardır. Algoritmalar 175 örnekten 148 ile 157 arasında örneğidoğru sınıflandırmışlardır. Ayrıca hesaplama süresi (computation time) en kısa olan algoritmanın single conjuctive learner algoritması, en uzun olanın ise KNN algoritması olduğunu da belirtmişlerdir. Hesaplama süresi bakımından ikinci sırada yer alan algoritma ise bayes network algoritmasıdır.

Çalışmada, en yüksek hata ortalamasının Rule Conjuctive learner algoritmasına, en düşük hata ortalamasının ise Bayes Network algoritmasına ait olduğu da ifade edilmiştir.

Sharma ve Jain (2013) çalışmalarında WEKA paket programında analiz yapmak için gerekli olan dosya dönüşümünü ve değişken seçimini tanıtmışlar ayrıca WEKA'nın sadece veri madenciliği algoritmalarını değil aynı zamanda genetik algoritmaları kullanmak için de etkili bir araç olduğunu ifade etmişlerdir. Çalışmada sınıflandırma algoritmalarından olan karar ağacı algoritmaları ile bir uygulama gerçekleştirmişler ve karar ağaçlarının performansını yüksek bulmuşlar; bu sebeple başka veri setlerinde de karar ağacı algoritmalarının kullanılabileceğini ortaya koymuşlardır.

Tax ve Duin (2002), "iki sınıflı sınıflandırıcıları çok sınıflı sınıflandırma problemlerinde kullanma" isimli çalışmalarında "voting"(oy verme) ve posterior (önsel) olasılıkların birleştirilmesi yaklaşımıyla iki sınıflı sınıflandırmadan çok sınıflı sınıflandırmaya yapılacak genellemelerin kolaylıkla mümkün olduğunu göstermişlerdir.

Aly (2005), "Çok sınıflı sınıflandırma metodları üzerine bir inceleme" başlıklı çalışmasında çok sınıflı sınıflandırma problemlerini çözmek için geliştirilmiş olan çeşitli yaklaşımları incelemiştir. YSA, DVM, NB, KNN, karar ağaçları gibi algoritmaları çok sınıflı problemler için yeniden formüle etmek; OVA, OVO, ECOC ve genelleştirilmiş kodlama yaklaşımlarını kullanarak çok sınıflı problemi iki sınıflı alt problemlere ayırmak, sınıfları bir karar ağacı yardımıyla oluşturarak ağacın düğümlerinde bir dizi ikili sınıflandırıcı kullanmak suretiyle yaprak düğümlere ulaşmak incelediği yaklaşımlardır.

Arruti, Mendialdua, Sierra, Lazkano ve Jauregi (2014) makalelerinde her bir alt problemde en iyi temel sınıflandırıcının uygulandığı iki yeni OVA ve OVO stratejileri birleşimini sunmuşlardır. OVA ve OVO ayrıştırma stratejileri ile elde edilen çıktıları birleştirdiği için ilk yöntemi OVA + OVO şeklinde isimlendirmişlerdir. İkinci birleşimi ise NOV@ olarak adlandırmışlardır. NOV@ tekniğinin amacı her alt problemde farklı temel sınıflandırıcılar kullanıldığında OVA stratejisinin çözemediği problemleri çözmektir. Önerilen bu iki yöntemi literatürdeki diğer iyi bilinen ayrıştırma stratejileriyle karşılaştırdıkları ampirik bir çalışma yapmışlardır. Bu çalışma UCI (Center of Machine Learning and Intelligent Systems, University of California)

deposundan alınan 20 veri seti üzerinde gerçekleştirilmiş ve sonucunda OVA ve OVO stratejilerinin uyumlu ve birleştirilebilir olduğu belirtilirken yeni önerilen NOV@ metodunun da sınıflandırma süresi bakımından en yeni metotlarla yarışabilecek düzeyde olduğu da ifade edilmiştir. Ayrıca NOV@ metodunun eğitim süresi bakımından diğer metotlarla kıyaslandığında zayıf kaldığı da çalışmada ifade edilen noktalar arasındadır.

Rocha ve Goldenstein (2014) çalışmalarında temel ikili öğrenicilerin korelasyonu ve ortak olasılığı kavramını tanıtmışlardır. Çalışmalarının iki temel amacından biri, çok sınıflı sınıflandırmada gerekli temel öğrenici sayısını azaltmak iken diğeri mevcut veri setini en iyi şekilde tamamlayabilecek yeni temel sınıflandırıcıları bulmaktır. Sonuç olarak herhangi bir veri setinde sınıflandırma sonuçları çok iyi olduğunda optimizasyon metotlarını kullanmanın çok anlamlı olmadığını belirtmişlerdir.

Sharma ve Sahni (2011) çalışmalarında istenmeyen e-posta veri setinde ID3, J48, basit sınıflandırma ve regresyon ağacı (simple CART) ve alternatif karar ağacı olmak üzere dört algoritmayı WEKA ortamında sınıflandırma doğruluğu açısından karşılaştırmışlardır. Simülasyon sonuçlarına göre J48 sınıflandırıcısı %92.7624 doğru sınıflandırma oranıyla, ID3, CART ve alternatif karar ağacını (ADTree) geride bırakmaktadır.

Zhang, Krawczyk, Garcia, Rosales-Perez ve Herrera (2016) çalışmalarında dengesiz çok sınıflı sınıflandırma problemleri için OVO şemasını, ikili kolektif (ensemble) öğrenme yaklaşımlarıyla güçlendirebilmek amacıyla kapsamlı bir ampirik analiz gerçekleştirmişlerdir. Önerilen yaklaşım CART, BPNN ve DVM olmak üzere üç temel sınıflandırıcı ile test edilmiştir. Çalışma göstermiştir ki OVO stratejinin performansı kolektif (ensemble) sınıflandırıcılar ile güçlendirildiğinde önemli ölçüde yükselmektedir ve kolektif (ensemble) öğrenme ile OVO şemasının birlikte kullanılması çok sınıflı dengesiz sınıflandırma problemleri için oldukça etkilidir.

Hassan, Shehab ve Hamed (2016) çalışmalarında bir hastanenin veri tabanında bulunan ve hastaların teşhis ve tedavi için gerekli olan tıbbi ve kişisel bilgilerini içeren bir veri setine on farklı sınıflandırma algoritması uygulamışlar ve en iyi performansa sahip olan algoritmayı çeşitli metrikler yardımıyla belirlemeye çalışmışlardır. Uygulama sonucunda elde edilen 0.987 TP oranı, 0.002 FP oranı, 0.988 kesinlik, 0.987 duyarlılık değerleri ile Bayes Net algoritması en başarılı algoritma olarak değerlendirilmiştir.

Kahraman, Çapar, Ayvacı, Demirel ve Gökmen (2004) çalışmalarında el yazısı tanıma problemi için en uygun sınıflandırıcıyı seçmek amacıyla YSA ve DVM algoritmalarının performanslarını karşılaştırmışlardır. DVM algoritmasının performansını hem eğitilme süresi hem de sınıflandırma başarısı bakımından YSA algoritmasının performansından daha yüksek bulmuşlardır. Bunun sebebi olarak da DVM algoritmasının destek vektörleri kullanmasını ve öznitelik uzayını daha yüksek bir boyuta taşımasını göstermişlerdir. DVM algoritmasının performansının artırılması için yapılacak önişlemlerin geliştirilebileceğini ya da farklı çekirdek tiplerinin denenebileceğini ifade etmişlerdir.

Aydın ve Özkul (2015) "Veri Madenciliği ve Anadolu Üniversitesi Açıköğretim Sisteminde bir Uygulama" isimli çalışmalarında öğrenci başarısını tahminlemek amacıyla C5.0, LR, YSA, Chaid, C&RT, QUEST algoritmalarını kıyaslamışlar ve %82.1 doğru tahminleme yüzdesiyle karar ağacı algoritmalarından C5.0 algoritmasını en başarılı algoritma olarak belirlemişlerdir.

Yurdakul (2015) "Veri Madenciliği ile Lise Öğrenci Performansının Değerlendirilmesi" isimli çalışmalarında lise öğrencilerine uyguladıkları 231 anket verisini kullanarak öğrenci performansına etki eden faktörler ve bu faktörler arasındaki ilişkileri ortaya koymuşlardır. Ayrıca kullandıkları algoritmaların doğru sınıflandırma yüzdelerini de paylaşmışlar ve multilayer perceptron algoritmasının %88.73 doğru tahminleme yüzdesi ile en başarılı algoritma olduğunu NB algoritmasının ise %81.69 doğru sınıflandırma yüzdesi ile performansı en düşük olan algoritma olduğunu belirtmişlerdir. Kullandıkları diğer algoritmalar ise sınıflandırma yüzdeleri bakımından çoktan aza doğru J-Rip, KNN ve J-48 Şeklinde sıralanmışlardır.

Romero, Ventura, Espejo ve Hervas (2008) bazı karar ağacı algoritmalarını bir eğitim veri seti üzerinde karşılaştırmışlardır. Karşılaştırma sonucunda 0.05 saniye uygulama süresi ve %56.25 doğru sınıflandırma yüzdesi ile en başarılı algoritmayı CART algoritması olarak belirlemişlerdir.

Pal ve Pal (2013) çalışmalarında üniversite verileri üzerinde ID3, C4.5 ve Bagging algoritmalarının performanslarını karşılaştırmışlar ve doğru sınıflandırma yüzdesi, ortalama hata miktarı ve Kappa ölçütü gibi metrikler dikkate alındığında ID3 algoritmasının en başarılı algoritma olarak belirlemişlerdir.

Coşkun (2010) “Veri Madenciliği Algoritmaları Karşılaştırılması” isimli tez çalışmasında J-48, NB, LR ve K-Star algoritmalarının performanslarını karşılaştırmış; doğruluk ve F ölçüsü bakımından J-48 algoritmasının diğer algoritmalara göre nispeten daha iyi performans gösterdiğini tespit etmiştir.

Köktürk (2012), KNN, YSA ve karar ağaçları algoritmalarının sınıflandırma performanslarını karşılaştırmak amacıyla yaptığı tez çalışmasında KNN algoritması için %78.3, YSA algoritması %90.8 ve karar ağacı algoritması için ise %82.5 sınıflandırma başarısı belirlemiş ve bu bulgulardan yola çıkarak YSA algoritmasının diğer tekniklere göre daha başarılı olduğunu ifade etmiştir.

Kaur, Singh ve Josan (2015) çalışmalarında Multilayer Perceptron, NB, SMO, J48 ve REPTree algoritmalarını karşılaştırmışlar; %75 başarı performansıyla Multilayer Perceptron algoritmasını en başarılı algoritma olarak belirlemişlerdir. Kullanılan diğer algoritmalardan NB algoritması için doğru sınıflandırma yüzdesi %65.13, SMO algoritması için %68.42, J-48 algoritması için %69.73 ve Rep Tree algoritması için %67.76’dır. Ayrıca Multilayer perceptron algoritmasının %82 F değeriyle de en başarılı algoritma olması yapılan çalışmanın sonuçları arasındadır.

3. BÖLÜM

YÖNTEM

Bu bölümde çalışmada kullanılacak araştırma yöntemi, araştırma grubu, veri toplama araçları, uygulama süreci ve verilerin analizi ile ilgili bilgiler verilmiştir.

3.1. Araştırmanın Yöntemi ve Deseni

Çalışmada PISA fen başarılarına göre sınıflandırma yapmak amacıyla OVA ve OVO ayrıştırma stratejileri çerçevesinde bazı veri madenciliği algoritmalarının performanslarının karşılaştırıldığından bu çalışma betimsel araştırma niteliğindedir.

3.2. Evren ve Örneklem

PISA 2015 Türkiye uygulamasına 61 ilden 187 okul ve 5895 öğrenci katılmıştır. Okullar belirlenirken tabakalı seçkisiz örnekleme yöntemi kullanılmış daha sonra bu okullardan seçilen öğrenciler yine seçkisiz yöntemle belirlenmiştir. Okullar, istatistiki bölge birimleri sınıflamasına göre belirlenen 12 bölge içinden, eğitim türü, okul türü, okulların bulundukları yer ve okulların idari biçimleri dikkate alınarak oluşturulmuştur (MEB, 2016).

3.3. Veri Toplama Araçları

PISA araştırması öğrencilerin hem “fen”, “matematik” ve “okuma” alanlarındaki okuryazarlıklarını belirlemek amacıyla geliştirilmiş başarı testlerini hem de öğrencilerin okul ve aile ortamlarını, motivasyonlarını, öğrenme süreçlerini değerlendirmek üzere geliştirilmiş anketleri içermektedir. Yapılan çalışmada fen başarı testinden elde edilen veriler ile anket sonuçlarından faydalanılmıştır. Her PISA döngüsünde değişmekle birlikte 2015 yılında ağırlıklı alan fen okuryazarlığı alanıdır. Bu sebeple öğrenci anketindeki maddeler de fen okuryazarlığına ilişkindir.

3.4. Uygulama

5895 sayıda örnek içeren 2015 PISA Türkiye verileri içerisinden boş veri içeren örnekler silinmiş ve 3459 sayıda örnek içeren bir veri seti elde edilmiştir. Daha sonra bu veriler içerisinden belirlenen 26 bağımsız ve 1 bağımlı değişken kategorik olarak

tanımlanmıştır. Bağımlı değişken olarak belirlenen fen okuryazarlığı değişkeni için ise 1 ve 2. düzeyler alt yeterlilik düzeyi, 3 ve 4. düzeyler orta yeterlilik düzeyi ve 5 ve 6. düzeyler ise üst yeterlilik düzeyi olarak değerlendirilmiştir. Elde edilen dosya arff formatına dönüştürülerek WEKA paket programında işlenmeye uygun hale getirilmiştir.

3.5. Verilerin Analizi

Verilerin analizinde WEKA paket programı içerisindeki Experimenter tezgahı kullanılmış, OVA ve OVO stratejileri altında belirlenen beş farklı algoritma veri setine uygulanmıştır. Çalışmada algoritmaların performanslarının birbiriyle kıyaslanabilecek şekilde olması ve yanlılık oluşmaması açısından parametre seçimi yapılmamış; tüm algoritmalar için varsayılan parametreler kullanılmıştır. Test seçeneklerinden 10 katlı çapraz geçerleme, birleştirme stratejilerinden de oy verme tekniği kullanılmıştır.

4. BÖLÜM

BULGULAR

Bu bölümde kullanılan veri setine ve analiz sonuçlarına ilişkin bulgular paylaşılmıştır. Veri setine ilişkin bilgilerde her bir tahmin edici değişkenin ve bağımlı değişken olan fen okuryazarlığının kategorilerindeki dağılımlar bir tablo ile verilmiştir. Tabloda rakamla belirtilen kategorilerin hangi değişkene karşılık geldiği tablonun altında belirtilmiştir.

Analiz sonuçlarına ilişkin bulgular çalışmanın ana ve alt problemlerine göre sınıflandırılarak verilmiştir. Bunun için OVA ve OVO stratejileri altında algoritmaların doğruluk, hata, F ve Kappa değerleri bir tablo ile gösterilmiştir. WEKA çıktıları olarak verilen şekillerde ise her bir algoritmanın yine OVA ve OVO stratejileri altında tüm sınıflar için gerçek pozitif oranı (TP rate), yanlış pozitif oranı (FP rate), kesinlik (precision), duyarlılık (recall), F ve MCC değerleri ile ROC ve PRC alanı değerleri verilmiştir. Ayrıca bu WEKA çıktılarında karmaşıklık matrisleri de bulunmaktadır. Son alt problem için J-48 algoritmasının ürettiği karar ağacı üzerinden fen okuryazarlığını etkileyen değişkenler yorumlanmıştır.

Tablo 3. *Değişkenlerin Kategorilerine Göre Dağılımları*

Değişkenler	1	2	3	4	5	6
Bölge	1130	458	571	515	270	515
Baba eğitim durumu	1251	902	503	803		
Anne eğitim durumu	1737	677	861	184		
Sınıf düzeyi	45	701	2675	38		
Cinsiyet	1736	1723				
Odaya sahip olma	2562	897				
Bilgisayara ulaşım imkanı	2468	991				
İnternete ulaşım imkanı	2298	1161				
Kaynak kitaplara ulaşım imkanı	2946	513				
Aile ilgisi	493	2966				
Okullla ilgili kaygı	1068	2391				
Hırslı olma	203	3256				
Takım çalışmasına yatkınlık	464	2995				
Olumsuz öğretmen davranışı	962	2497				
Fen sınıf disiplini	1121	2338				
Sorgulamaya dayalı öğretim	2853	606				
Okula ait hissetme	2791	668				
Öğretmen desteği	798	2661				
Öğretmen odaklı eğitim	2438	1021				
Geribildirim	3082	377				
Uyarlanabilir öğretim	1471	1988				
İçsel motivasyon	2720	739				
Araçsal motivasyon	1954	1505				
Özyeterlilik	2111	1348				
Epistemik inançlar	1814	1645				
Fen etkinliklerine katılım	808	2651				
Fen okuryazarlığı	2482	970	7			

Tablo 3'te değişkenlerin kategorilerine göre dağılımları verilmiştir. Bölge değişkeni için 1:Marmara, 2:Ege, 3:İç Anadolu, 4:Akdeniz, 5: Karadeniz ve 6: Doğu ve Güneydoğu Anadolu Bölgeleri'ni göstermektedir. Baba eğitim durumu ve anne eğitim durumu değişkenleri için 1: okumamış ve ilkokul, 2: ortaokul, 3: lise, 4:üniversite ve üstü mezuniyet durumlarını göstermektedir. Sınıf düzeyi değişkeni için 1: 7. ve 8. sınıfları, 2: 9. sınıfları, 3: 10. sınıfları, 4 ise 11 ve 12. sınıfları göstermektedir. Cinsiyet değişkeninde 1: kızları 2 ise erkekleri göstermektedir. Kendine ait odaya sahip olma, bilgisayar, internet ve kaynak kitaplara ulaşım değişkenleri için 1:var, 2:yok anlamına gelmektedir. Aile ilgisi, sınav kaygısı, hırslı olma, takım çalışmasına yatkınlık, fen derslerindeki sınıf disiplini, olumsuz öğretmen davranışı, içsel motivasyon, epistemik inançlar ve okula ait hissetme değişkenlerinde 1: düşük 2:yüksek olma durumunu ifade etmekte iken araçsal motivasyon ve özyeterlilik değişkenlerinde 1:düşük 2 ise yüksek olma durumunu ifade etmektedir. Sorgulamaya dayalı öğretim ve öğretmen desteği değişkenleri için 1: derslerin çoğunda ya da her ders durumunu belirtirken, 2: bazen ya

da hemen hemen hiç durumunu belirtmektedir. Öğretmen odaklı eğitim, geribildirim verme ve uyarlanabilir öğretim değişkenleri için ise 1: hiç ya da hemen hemen hiç durumunu ifade ederken; 2: derslerin çoğu ya da her ders durumunu ifade etmektedir. Fen etkinlikleri değişkeninde ise 1: düzenli aralıklarla ya da sık sık durumunu belirtirken; 2 ise hiç ya da bazen durumunu belirtmektedir. Bağımlı değişken durumunda olan fen okuryazarlığı değişkeninde ise 1b, 1a ve 2. Seviyede bulunan öğrenciler 1 koduyla; 3 ve 4. seviyedeki öğrenciler 2 koduyla ve 5 ve 6. Seviyedeki öğrenciler ise 3 koduyla verilmiştir.

4.1. Birinci ve İkinci Alt Probleme İlişkin Bulgular

OVA stratejisi altında algoritmaların doğruluk, hata ve Kappa değerleri tablo 4'te gösterilmiştir.

Tablo 4. *OVA Stratejisi Altında Algoritmaların Çeşitli Metrikler Açısından Performans Değerleri*

Metrikler Algoritmalar	Doğruluk (%)	MAE	RAE (%)	RMSE	RRSE (%)	Kappa
LR	74.1255	0.2221	81.9013	0.3365	91.4183	0.275
NB	72.4776	0.2219	81.8183	0.3497	94.99	0.3138
DVM	73.1714	0.1779	65.6084	0.4178	113.4935	0.1249
J-48	72.4198	0.2318	85.4799	0.371	100.7778	0.2474
KNN	67.5629	0.2454	90.4998	0.4372	118.7806	0.1726

Tabloya göre doğruluk değeri en yüksek olan algoritma LR algoritmasıdır. Bu algoritmayı sırasıyla DVM, NB, J-48 ve KNN algoritmaları takip etmektedir. MAE ve RAE hata değerleri bakımından algoritmalar DVM, NB, LR, J-48 ve KNN olarak sıralanmıştır. RMSE ve RRSE hata değerleri açısından ise algoritmalar LR, NB, J-48, DVM ve KNN olarak sıralanmışlardır. Kappa metriği bakımından NB algoritmasını sırasıyla LR, J-48, KNN ve DVM algoritmaları takip etmektedir.

Algoritmaların hata oranlarını değerlendirebilmek için MAE ve RMSE değerlerinin ortalaması çok sık kullanılan bir göstergedir. Alternatif olarak, göreli

hatalar da kullanılabilir fakat ortalama değeri almak akıllıca olmaktadır (Bin Othman ve Yau, 2007).

OVA stratejisi altında en yüksek ortalama hata değerine sahip algoritma 0.3413 değeriyle KNN algoritmasıdır. En düşük ortalama hata değerine sahip olan algoritma ise 0.2793 değeriyle LR algoritmasıdır. Aradaki algoritmalar ise sırasıyla 0.3014 değeriyle j-48 algoritması 2. Sırada, 0.29785 değeriyle SVM algoritması 3. Sırada ve 0.2858 değeriyle NB ise 4. Sıradadır.

OVA stratejisi altında WEKA çıktıları olarak verilen şekillerde ise her bir algoritmanın yine OVA stratejisi altında tüm sınıflar için gerçek pozitif oranı (TP rate), yanlış pozitif oranı (FP rate), kesinlik, duyarlılık, F ve MCC değerleri ile ROC ve PRC alanı değerleri verilmiştir. Ayrıca tüm şekiller algoritmaların her bir sınıftaki örnekleri nasıl yerleştirdiğini gösteren karmaşıklık matrislerini de içermektedir.

```
=== Detailed Accuracy By Class ===

               TP Rate  FP Rate  Precision  Recall   F-Measure  MCC      ROC Area  PRC Area  Class
               0,904    0,656    0,778      0,904    0,836      0,299    0,762     0,889     1
               0,331    0,093    0,580      0,331    0,422      0,291    0,759     0,532     2
               0,000    0,006    0,000      0,000    0,000      -0,004   0,829     0,032     3
Weighted Avg.  0,741    0,497    0,721      0,741    0,718      0,297    0,762     0,787

=== Confusion Matrix ===

   a    b    c   <-- classified as
2243  228   11 |    a = 1
 638  321   11 |    b = 2
   3    4    0 |    c = 3
```

Şekil 10. OVA Stratejisi Altında LR Algoritmasının Her Sınıftaki Performans Değerleri ve Karmaşıklık Matrisi

Şekil 10'da LR algoritmasının OVA stratejisi altında bağımlı değişkenin her bir kategorisine ilişkin değerleri görülmektedir. Ayrıca karmaşıklık matrisinde de görüldüğü üzere LR algoritması 1.sınıfa ait olan örneklerin 2243 tanesini, 2.sınıfa ait olan örneklerin 321 tanesini doğru olarak sınıflandırmıştır. 3. sınıfa ait olan örneklerin hiçbirini doğru olarak sınıflandıramamıştır.

```
=== Detailed Accuracy By Class ===

               TP Rate  FP Rate  Precision  Recall   F-Measure  MCC      ROC Area  PRC Area  Class
               0,817    0,506    0,804      0,817    0,811      0,316    0,745     0,880     1
               0,493    0,182    0,513      0,493    0,503      0,314    0,743     0,509     2
               0,000    0,001    0,000      0,000    0,000      -0,002   0,901     0,035     3
Weighted Avg.  0,725    0,414    0,721      0,725    0,723      0,315    0,745     0,774

=== Confusion Matrix ===

    a    b    c   <-- classified as
 2029  449    4 |    a = 1
  492  478    0 |    b = 2
    2    5    0 |    c = 3
```

Şekil 11. OVA Stratejisi Altında NB Algoritmasının Bağımlı Değişkenin Sınıflarına Ait Performans Değerleri ve Karmaşıklık Matrisi

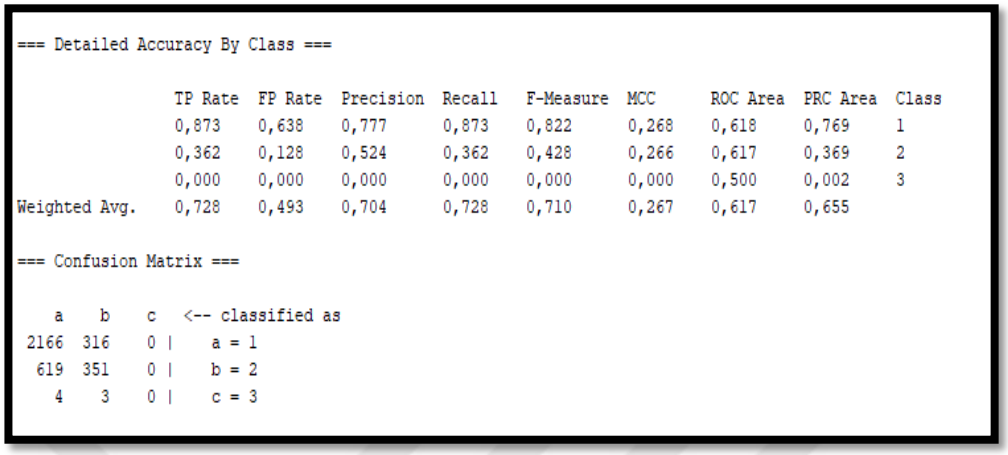
NB algoritmasının OVA stratejisi altında bağımlı değişkenin her bir kategorisine ilişkin değerleri şekil 11'de görülmektedir. Ayrıca karmaşıklık matrisinde de görüldüğü üzere NB algoritması 1.sınıfa ait olan örneklerin 2029 tanesini, 2.sınıfa ait olan örneklerin 478 tanesini doğru olarak sınıflandırmıştır. 3. sınıfa ait olan örneklerin hiçbirini doğru olarak sınıflandıramamıştır.

```
=== Detailed Accuracy By Class ===

               TP Rate  FP Rate  Precision  Recall   F-Measure  MCC      ROC Area  PRC Area  Class
               0,972    0,876    0,738      0,972    0,839      0,189    0,568     0,746     1
               0,123    0,029    0,623      0,123    0,205      0,184    0,567     0,340     2
               0,000    0,000    0,000      0,000    0,000      0,000    0,488     0,002     3
Weighted Avg.  0,732    0,637    0,704      0,732    0,659      0,187    0,567     0,631

=== Confusion Matrix ===

    a    b    c   <-- classified as
 2412   70    0 |    a = 1
  851  119    0 |    b = 2
    5    2    0 |    c = 3
```

Şekil 12. OVA Stratejisi Altında DVM Algoritmasının Bağımlı Değişkenin Sınıflarına Ait Performans Değerleri ve Karmaşıklık Matrisi

DVM algoritmasının OVA stratejisi altında bağımlı değişkenin her bir kategorisine ilişkin değerler şekil 12'de görülmektedir. Ayrıca karmaşıklık matrisinde de görüldüğü üzere DVM algoritması 1.sınıfa ait olan örneklerin 2412 tanesini, 2.sınıfa

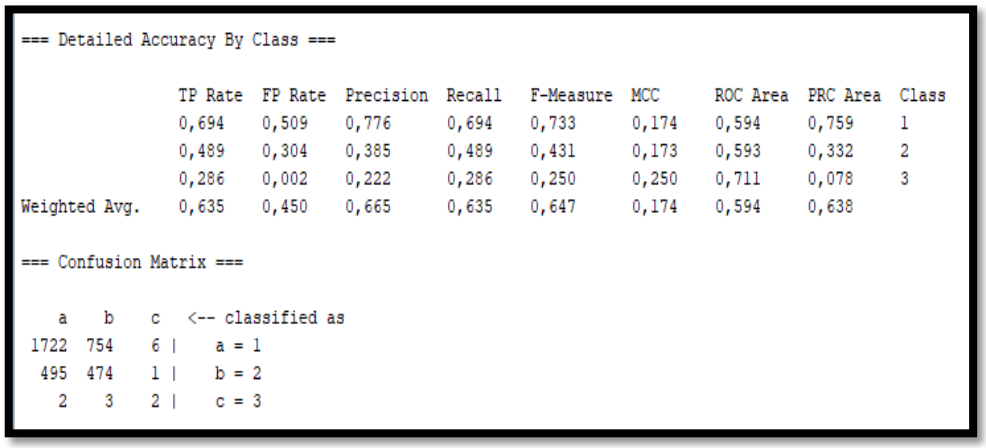
ait olan örneklerin 119 tanesini doğru olarak sınıflandırmıştır. 3. sınıfa ait olan örneklerin hiçbirini doğru olarak sınıflandıramamıştır.



```
=== Detailed Accuracy By Class ===

               TP Rate  FP Rate  Precision  Recall   F-Measure  MCC      ROC Area  PRC Area  Class
               0,873    0,638    0,777      0,873    0,822      0,268    0,618     0,769     1
               0,362    0,128    0,524      0,362    0,428      0,266    0,617     0,369     2
               0,000    0,000    0,000      0,000    0,000      0,000    0,500     0,002     3
Weighted Avg.  0,728    0,493    0,704      0,728    0,710      0,267    0,617     0,655

=== Confusion Matrix ===

    a    b    c   <-- classified as
 2166  316    0 |    a = 1
  619  351    0 |    b = 2
    4    3    0 |    c = 3
```

Şekil 13. OVA Stratejisi Altında J-48 Algoritmasının Bağımlı Değişkenin Sınıflarına Ait Performans Değerleri ve Karmaşıklık Matrisi

J-48 algoritmasının OVA stratejisi altında bağımlı değişkenin her bir kategorisine ilişkin değerleri şekil 13'te görülmektedir. Ayrıca karmaşıklık matrisinde de görüldüğü üzere J-48 algoritması 1.sınıfa ait olan örneklerin 2166 tanesini, 2.sınıfa ait olan örneklerin 351 tanesini doğru olarak sınıflandırmıştır. 3. sınıfa ait olan örneklerin hiçbirini doğru olarak sınıflandıramamıştır.



```
=== Detailed Accuracy By Class ===

               TP Rate  FP Rate  Precision  Recall   F-Measure  MCC      ROC Area  PRC Area  Class
               0,694    0,509    0,776      0,694    0,733      0,174    0,594     0,759     1
               0,489    0,304    0,385      0,489    0,431      0,173    0,593     0,332     2
               0,286    0,002    0,222      0,286    0,250      0,250    0,711     0,078     3
Weighted Avg.  0,635    0,450    0,665      0,635    0,647      0,174    0,594     0,638

=== Confusion Matrix ===

    a    b    c   <-- classified as
 1722  754    6 |    a = 1
  495  474    1 |    b = 2
    2    3    2 |    c = 3
```

Şekil 14. OVA Stratejisi Altında KNN Algoritmasının Bağımlı Değişkenin Sınıflarına Ait Performans Değerleri ve Karmaşıklık Matrisi

KNN algoritmasının OVA stratejisi altında bağımlı değişkenin her bir kategorisine ilişkin değerleri şekil 14'te görülmektedir. Ayrıca karmaşıklık matrisinde

de görüldüğü üzere KNN algoritması 1.sınıfa ait olan örneklerin 1722 tanesini, 2.sınıfa ait olan örneklerin 474 tanesini, 3. sınıfa ait olan örneklerin ise 2 tanesini doğru olarak sınıflandırmıştır.

OVA stratejisi altında sadece KNN algoritması iki örneği doğru sınıfa dahil etmiştir. Diğer algoritmalar en az örneğe sahip olan sınıfın, yani 3 ile temsil edilen sınıfın, hiçbir örneğini doğru sınıfa yerleştirememişlerdir.

Tablo 5. *OVA Stratejisi Altında Algoritmaların Karşılaştırılması*

	Doğruluk	Kesinlik	Duyarlılık	F-ölçüsü
LR	74.1255	0.721	0.741	0.718
NB	72.4776	0.721	0.725	0.723
DVM	73.1714	0.704	0.732	0.659
J-48	72.4198	0.704	0.728	0.710
KNN	67.5629	0.666	0.635	0.647

Tablo 5'teki veriler incelendiğinde OVA stratejisi altında %74.1255 doğruluk değeriyle LR algoritmasının en iyi sonucu ürettiği söylenebilir. Algoritmaların performanslarının değerlendirilmesinde doğruluk ölçütü basit fakat önemli bir kriterdir. Kesinlik ölçütü bakımından 0.721 değeriyle en yüksek performansı gösteren algoritmalar NB ve LR algoritmaları olsa da kesinlik ölçütünün tek başına yorumlanması yanlış değerlendirmelere sebep olabilir. Bu sebeple kesinlik ölçütü duyarlılık ölçütüyle birlikte değerlendirilmelidir. Duyarlılık ölçütüne göre en başarılı algoritma 0.741 değeriyle LR algoritmasıdır. Kesinlik ve duyarlılık metriklerinin bir arada değerlendirilmesi için bu metriklerin harmonik ortalaması olan F ölçüsü göz önüne alınmalıdır. F ölçüsüne göre ise 0.723 değeriyle en başarılı algoritma NB algoritmasıdır.

4.2. Üçüncü ve Dördüncü Alt Probleme İlişkin Bulgular

OVO stratejisi altında tüm algoritmaların doğruluk ve Kappa değerleri ile hata değerleri tablo 6'da verilmiştir.

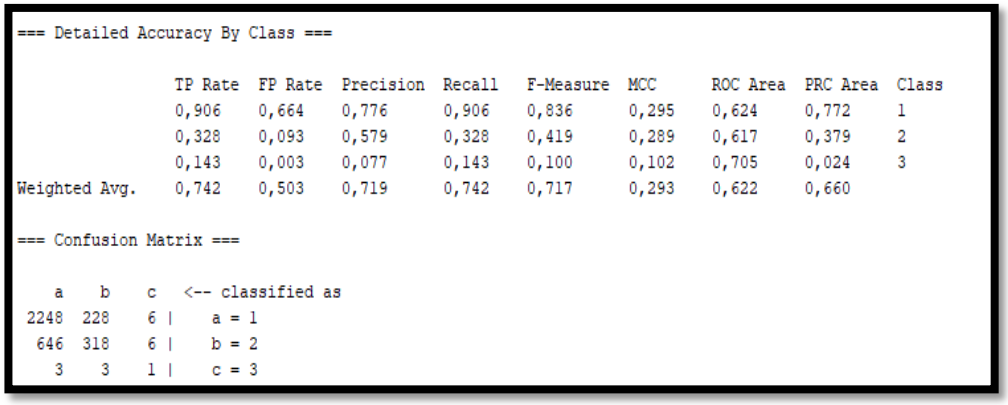
Tablo 6. *OVO Stratejisi Altında Doğruluk, Hata ve Kappa Değerleri*

Metrikler Algoritmalar	Doğruluk (%)	MAE	RAE (%)	RMSE	RRSE (%)	Kappa
LR	74.2122	0.2809	103.6096	0.3641	98.9187	0.2726
NB	72.3041	0.2846	104.9601	0.3694	100.3543	0.311
DVM	73.3738	0.2818	103.9413	0.3656	99.3333	0.146
J-48	72.7667	0.2832	104.4389	0.3675	99.8332	0.2582
KNN	63.5444	0.3039	112.0681	0.3946	107.2075	0.1712

OVO stratejisi altında algoritmaların çeşitli metrikler açısından değerleri tabloda verilmiştir. Bu tabloya göre doğruluk değeri en yüksek olan algoritma LR algoritmasıdır. Bu algoritmayı sırasıyla DVM, NB, J-48 ve KNN algoritmaları takip etmektedir. MAE hata metriği bakımından LR algoritmasını sırasıyla DVM, NB, J-48 ve KNN algoritmaları takip etmektedir. RAE, RMSE ve RRSE metrikleri bakımından ise algoritmalar LR, DVM, J-48, NB, KNN olarak sıralanmaktadır. Kappa ölçütüne göre ise sıralama NB, LR, J-48, KNN ve DVM şeklindedir.

Algoritmaların hata oranlarını değerlendirebilmek için MAE ve RMSE ortalaması çok sık kullanılan bir göstergeyi ifade etmektedir. Alternatif olarak, göreli hatalar da kullanılır. Fakat ortalama değeri almak akıllıca olmaktadır. Düşük hata değerine sahip olan algoritma daha güçlü sınıflandırma yeteneğine sahip olduğundan tercih edilmektedir (Bin Othman ve Yau, 2007).

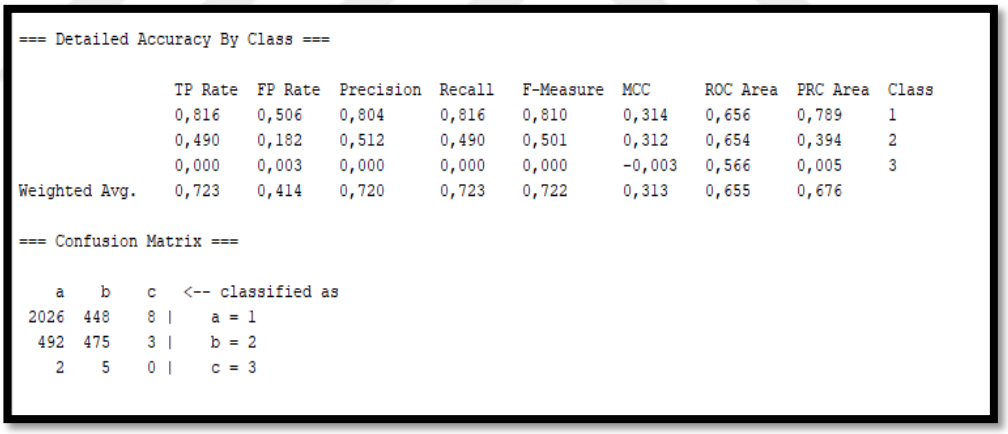
OVO stratejisi altında en yüksek ortalama hata değerine sahip algoritma 0.34925 değeriyle KNN algoritmasıdır. En düşük ortalama hata değerine sahip olan algoritma ise 0.3225 değeriyle LR algoritmasıdır. Aradaki algoritmalar ise sırasıyla 0.327 değeriyle NB algoritması 2. Sırada, 0.32535 değeriyle J-48 algoritması 3. Sırada ve 0.3237 değeriyle DVM ise 4. sırada olacak şekilde sıralanmışlardır.



```
=== Detailed Accuracy By Class ===

               TP Rate  FP Rate  Precision  Recall   F-Measure  MCC      ROC Area  PRC Area  Class
               0,906    0,664    0,776      0,906    0,836      0,295    0,624     0,772     1
               0,328    0,093    0,579      0,328    0,419      0,289    0,617     0,379     2
               0,143    0,003    0,077      0,143    0,100      0,102    0,705     0,024     3
Weighted Avg.  0,742    0,503    0,719      0,742    0,717      0,293    0,622     0,660

=== Confusion Matrix ===

    a    b    c   <-- classified as
 2248  228    6 |    a = 1
  646  318    6 |    b = 2
    3    3    1 |    c = 3
```

Şekil 15. OVO Stratejisi Altında LR Algoritmasının Bağımlı Değişkenin Sınıflarına Ait Performans Değerleri ve Karmaşıklık Matrisi

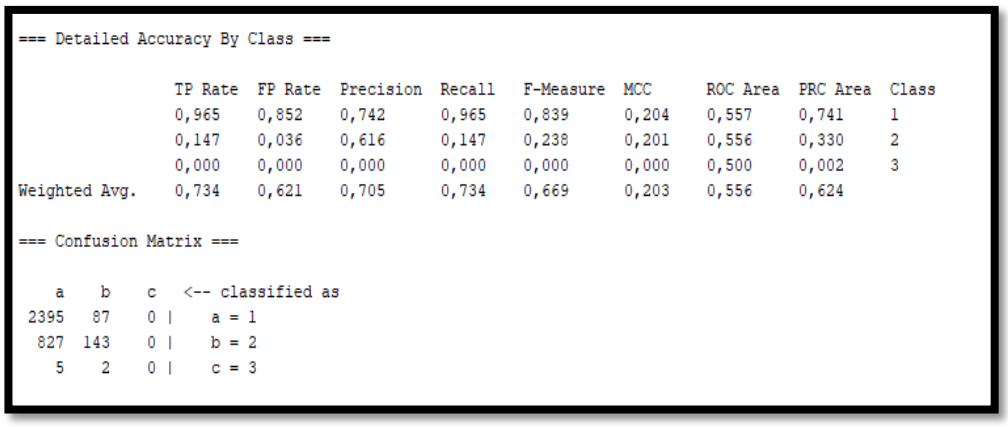
LR algoritmasının OVO stratejisi altında bağımlı değişkenin her bir kategorisine ilişkin değerleri şekil 15'te görülmektedir. Ayrıca karmaşıklık matrisinde de görüldüğü üzere LR algoritması 1.sınıfa ait olan örneklerin 2248 tanesini, 2.sınıfa ait olan örneklerin 318 tanesini, 3. Sınıfa ait olan örneklerin ise 1 tanesini doğru olarak sınıflandırmıştır.



```
=== Detailed Accuracy By Class ===

               TP Rate  FP Rate  Precision  Recall   F-Measure  MCC      ROC Area  PRC Area  Class
               0,816    0,506    0,804      0,816    0,810      0,314    0,656     0,789     1
               0,490    0,182    0,512      0,490    0,501      0,312    0,654     0,394     2
               0,000    0,003    0,000      0,000    0,000      -0,003   0,566     0,005     3
Weighted Avg.  0,723    0,414    0,720      0,723    0,722      0,313    0,655     0,676

=== Confusion Matrix ===

    a    b    c   <-- classified as
 2026  448    8 |    a = 1
  492  475    3 |    b = 2
    2    5    0 |    c = 3
```

Şekil 16. OVO Stratejisi Altında NB Algoritmasının Bağımlı Değişkenin Sınıflarına Ait Performans Değerleri ve Karmaşıklık Matrisi

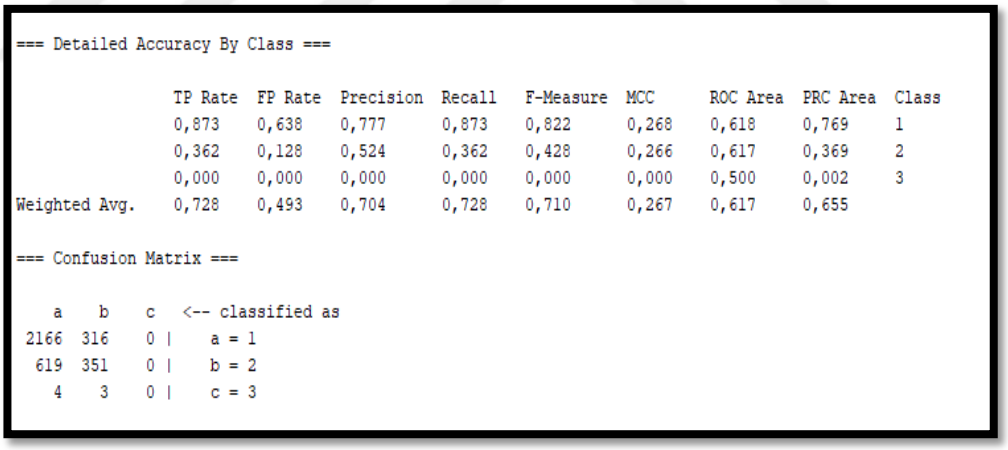
NB algoritmasının OVO stratejisi altında bağımlı değişkenin her bir kategorisine ilişkin değerleri şekil 16'da görülmektedir. Ayrıca karmaşıklık matrisinde de görüldüğü üzere NB algoritması 1.sınıfa ait olan örneklerin 2026 tanesini, 2.sınıfa ait olan örneklerin 475 tanesini doğru olarak sınıflandırmıştır. 3. Sınıfa ait olan örneklerin hiçbirini doğru olarak sınıflandıramamıştır.



```
=== Detailed Accuracy By Class ===

               TP Rate  FP Rate  Precision  Recall   F-Measure  MCC      ROC Area  PRC Area  Class
               0,965    0,852    0,742      0,965    0,839      0,204    0,557     0,741     1
               0,147    0,036    0,616      0,147    0,238      0,201    0,556     0,330     2
               0,000    0,000    0,000      0,000    0,000      0,000    0,500     0,002     3
Weighted Avg.  0,734    0,621    0,705      0,734    0,669      0,203    0,556     0,624

=== Confusion Matrix ===

    a    b    c   <-- classified as
 2395   87    0 |    a = 1
  827  143    0 |    b = 2
    5    2    0 |    c = 3
```

Şekil 17. OVO Stratejisi Altında DVM Algoritmasının Bağımlı Değişkenin Sınıflarına Ait Performans Değerleri ve Karmaşıklık Matrisi

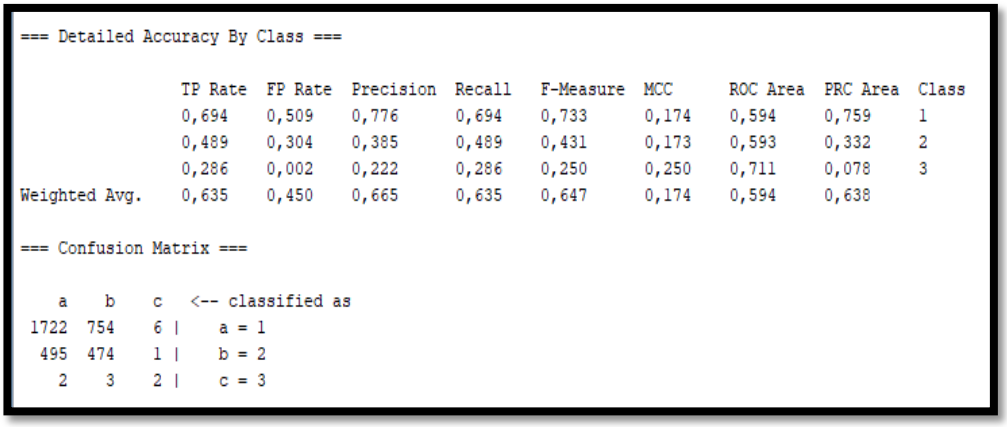
DVM algoritmasının OVO stratejisi altında bağımlı değişkenin her bir kategorisine ilişkin değerleri şekil 17'de görülmektedir. Ayrıca karmaşıklık matrisinde de görüldüğü üzere DVM algoritması 1.sınıfa ait olan örneklerin 2395 tanesini, 2.sınıfa ait olan örneklerin 143 tanesini doğru olarak sınıflandırmıştır. 3. Sınıfa ait olan örneklerin hiçbirini doğru olarak sınıflandıramamıştır.



```
=== Detailed Accuracy By Class ===

               TP Rate  FP Rate  Precision  Recall   F-Measure  MCC      ROC Area  PRC Area  Class
               0,873    0,638    0,777      0,873    0,822      0,268    0,618     0,769     1
               0,362    0,128    0,524      0,362    0,428      0,266    0,617     0,369     2
               0,000    0,000    0,000      0,000    0,000      0,000    0,500     0,002     3
Weighted Avg.  0,728    0,493    0,704      0,728    0,710      0,267    0,617     0,655

=== Confusion Matrix ===

    a    b    c   <-- classified as
 2166  316    0 |    a = 1
  619  351    0 |    b = 2
    4    3    0 |    c = 3
```

Şekil 18. OVO Stratejisi Altında J-48 Algoritmasının Bağımlı Değişkenin Sınıflarına Ait Performans Değerleri ve Karmaşıklık Matrisi

J-48 algoritmasının OVO stratejisi altında bağımlı değişkenin her bir kategorisine ilişkin değerler şekil 18'de görülmektedir. Ayrıca karmaşıklık matrisinde de görüldüğü üzere J-48 algoritması 1.sınıfa ait olan örneklerin 2166 tanesini, 2.sınıfa ait olan örneklerin 351 tanesini doğru olarak sınıflandırmıştır. 3. Sınıfa ait olan örneklerin hiçbirini doğru olarak sınıflandıramamıştır.



```
=== Detailed Accuracy By Class ===

               TP Rate  FP Rate  Precision  Recall   F-Measure  MCC      ROC Area  PRC Area  Class
               0,694    0,509    0,776      0,694    0,733      0,174    0,594     0,759     1
               0,489    0,304    0,385      0,489    0,431      0,173    0,593     0,332     2
               0,286    0,002    0,222      0,286    0,250      0,250    0,711     0,078     3
Weighted Avg.  0,635    0,450    0,665      0,635    0,647      0,174    0,594     0,638

=== Confusion Matrix ===

    a    b    c   <-- classified as
 1722  754    6 |    a = 1
  495  474    1 |    b = 2
    2    3    2 |    c = 3
```

Şekil 19. OVO Stratejisi Altında KNN Algoritmasının Bağımlı Değişkenin Sınıflarına İlişkin Performans Değerleri ve Karmaşıklık Matrisi

KNN algoritmasının OVO stratejisi altında bağımlı değişkenin her bir kategorisine ilişkin değerler şekil 19'da görülmektedir. Ayrıca karmaşıklık matrisinde de görüldüğü üzere KNN algoritması 1.sınıfa ait olan örneklerin 1722 tanesini, 2.sınıfa ait olan örneklerin 474 tanesini, 3. Sınıfa ait olan örneklerin ise 2 tanesini doğru olarak sınıflandırmıştır.

OVO stratejisi altında en az örnekli sınıf için LR algoritması 1 örneği, KNN algoritması ise 2 örneği doğru sınıfa dâhil etmiştir. Diğer algoritmalar ise hiçbir örneğin sınıfını doğru tahminleyememişlerdir.

Tablo 7. *OVO Stratejisi Altında Algoritmaların Doğruluk, Kesinlik, Duyarlılık ve F Ölçüsü Değerleri*

	Doğruluk	Kesinlik	Duyarlılık	F-ölçüsü
LR	74.2122	0.719	0.742	0.717
NB	72.3041	0.720	0.723	0.722
DVM	73.3738	0.705	0.734	0.669
J-48	72.7667	0.704	0.728	0.710
KNN	63.5444	0.665	0.635	0.647

Tablo 7'deki veriler incelendiğinde OVO stratejisi altında %74.2122 doğruluk derecesiyle LR algoritmasının en iyi sonucu ürettiği söylenebilir. Algoritmaların

performanslarının değerlendirilmesinde doğruluk ölçütü basit fakat önemli bir kriterdir. Kesinlik ölçütü bakımından 0.720 değeriyle en yüksek performansı gösteren algoritma NB olsa da kesinlik ölçütünün tek başına yorumlanması yanlış değerlendirmelere sebep olabilir. Kesinlik ölçütü duyarlılık ölçütüyle birlikte değerlendirilmelidir. Duyarlılık ölçütüne göre en başarılı algoritma 0.742 değeriyle LR algoritmasıdır. Kesinlik ve duyarlılık metriklerinin bir arada değerlendirilmesi için bu metriklerin harmonik ortalaması olan F ölçüsüne göre en başarılı algoritma NB algoritmasıdır.

4.3. Beşinci ve Altıncı Alt Probleme İlişkin Bulgular

Algoritmaların her birinin OVA ve OVO stratejileri altındaki tüm metriklerdeki performans değerleri karşılaştırmalı olarak Tablo 8'de verilmiştir. Ayrıca OVA ve OVO stratejilerinden hangisinin veri setini sınıflandırmada daha başarılı olduğu da tablo 8 yardımıyla değerlendirilecektir.

Tablo 8. *Algoritmaların Stratejiler Bazında Karşılaştırılması*

	LR		NB		DVM		J-48		KNN		Ortalama	
	OVA	OVO	OVA	OVO	OVA	OVO	OVA	OVO	OVA	OVO	OVA	OVO
Doğ.	74.1255	74.2122	72.4776	72.3041	73.1714	73.3738	72.4198	72.7667	67.5629	63.5444	71.9514	71.2402
Kes.	0.721	0.719	0.721	0.720	0.704	0.705	0.704	0.704	0.666	0.665	0.7032	0.7026
Duy.	0.741	0.742	0.725	0.723	0.732	0.734	0.728	0.728	0.635	0.635	0.7122	0.7124
F-ölç.	0.718	0.717	0.723	0.722	0.659	0.669	0.710	0.710	0.647	0.647	0.6914	0.693
Kapp.	0.275	0.2726	0.3138	0.311	0.1248	0.146	0.2474	0.2582	0.1726	0.1712	0.22672	0.2318

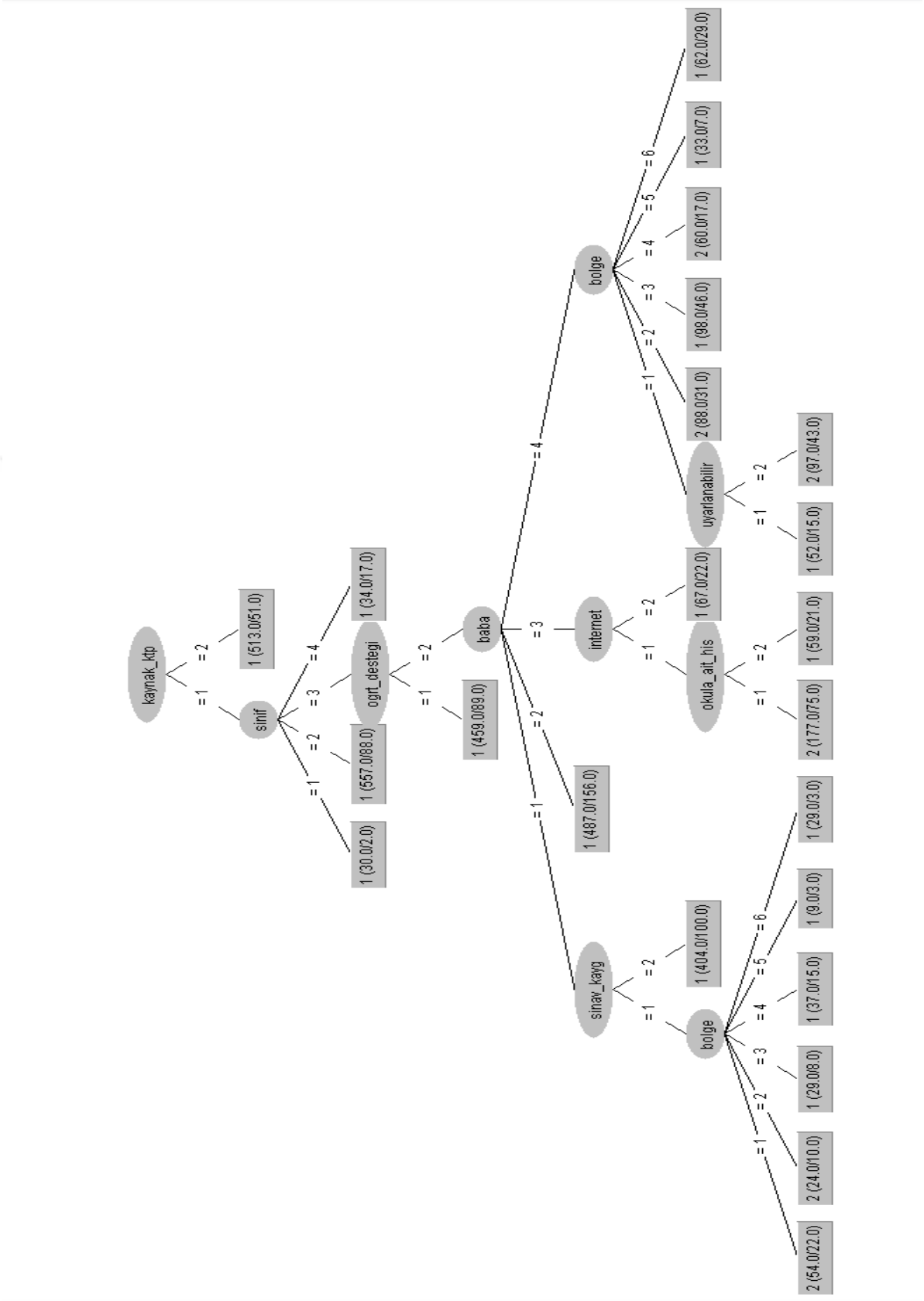
LR algoritmasının doğruluk ve duyarlılık değerleri OVO stratejisinde daha yüksek iken kesinlik, F ölçüsü ve kappa değerleri OVA stratejisinde daha yüksektir. NB algoritmasının tüm değerleri OVA stratejisinde daha yüksektir. DVM algoritmasının

tüm değerleri OVO stratejisi altında daha yüksektir. J-48 algoritmasının doğruluk ve kappa değerleri OVO stratejisi altında daha yüksek iken diğer değerleri her iki stratejide de aynı kalmıştır. KNN algoritmasının doğruluk, kesinlik ve kappa değerleri OVA stratejisi altında daha yüksektir, duyarlılık ve F değerleri ise her iki strateji altında da eşittir. Ortalama değerler incelendiğinde doğruluk ve kesinlik değerleri bakımından OVA stratejisi, duyarlılık, F ölçüsü ve kappa metrikleri bakımından ise OVO stratejisinin başarı ortalamasının daha yüksek olduğu söylenebilir.

Doğruluk değeri bakımından her iki stratejiyi ve tüm algoritmaları göz önüne alırsak en başarılı algoritma OVO stratejisi altında LR algoritmasıdır. Kesinlik değerlerine bakıldığında OVA stratejisi altında LR ve NB algoritmaları en başarılı algoritmalardır. Duyarlılık değerleri bakımından OVO stratejisi altında LR algoritması en başarılı algoritma olarak değerlendirilebilir. F ölçüsü bakımından ise OVA stratejisi altında NB algoritmasının en başarılı algoritma olduğu söylenebilir.

4.7. Yedinci Alt Probleme İlişkin Bulgular

Yedinci alt problem "seçilen değişkenler içinden fen okuryazarlığını etkileyen değişkenler nelerdir?" şeklindedir. Bu sebeple J-48 algoritması ile karar ağacı oluşturulmuş ve düğüm noktaları üzerinden değerlendirme yapılmıştır. Model kurulmadan önce eldeki 26 bağımsız değişken içinden filtreler içerisinde bulunan "attribute selection" özellik seçim filtresiyle bağımsız değişkenlerin sayısı 10 olarak belirlenmiştir. Bu öznitelik seçim yöntemi, J-48 algoritması için, bilgi kazanç oranına göre bağımlı değişken üzerinde en çok etkisi olan değişkenleri belirlemektedir. J-48 algoritmasının veri setine uygulanmasıyla şekil 20'deki karar ağacı elde edilmiştir.



Şekil 20. J-48 Algoritmasının Ürettiği Karar Ağacı

Şekil 20'de görülen yuvarlak şekiller düğüm noktalarını, dikdörtgensel şekiller yaprakları göstermektedir. Yapraklarda parantezin içindeki ilk sayı veri kümesindeki kaç örneğin bu yaprakta doğru olarak sınıflandırıldığını, parantezin içindeki ikinci sayı ise yanlış olarak sınıflandırılan örneklerin sayısını göstermektedir. Şekil 20'de görüldüğü gibi düğüm noktaları yukarıdan aşağıya doğru kaynak kitaba ulaşım, sınıf düzeyi, öğretmen desteği, baba eğitim durumu, sınav kaygısı, internete ulaşım imkânı, bölge, okula ait hissetme ve uyarlanabilir öğretim şeklindedir. Üç sınıf olarak düzenlenen bağımlı değişkenin başarılı olarak kaydedilen sınıfındaki hiçbir örnek doğru olarak sınıflandırılamamıştır. Başka bir deyişle J-48 algoritması düşük seviyedeki ve orta sevideki öğrencileri birbirinden ayırabilmiştir. Bu veriler ışığında J48 karar ağacı algoritmasının sonuçlarını aşağıdaki şekilde sıralamak mümkündür:

1. Birinci dallanma kök düğüm olan kaynak kitaba ulaşım değişkeni ile başlamıştır. Kaynak kitaba ulaşım imkânı olmayan öğrencilerin düşük fen okuryazarlığına sahip olduğu görülmektedir.
2. İkinci dallanma sınıf düzeyine göre oluşmuş ve 10. Sınıfların daha yüksek fen okuryazarlığına sahip olduğu diğer grupların düşük fen okuryazarlığına sahip olduğu görülmüştür.
3. Üçüncü dallanma öğretmen desteği değişkenine göre oluşmuş, öğretmen desteğinin fazla olduğu durumlarda öğrencilerin düşük fen okuryazarlığına sahip olduğu tespit edilmiştir.
4. Dördüncü dallanma baba eğitim durumu değişkenine göre oluşmuştur. Baba eğitim durumları ortaokul olanlar düşük fen okuryazarlığına sahiptir. Baba eğitim durumu okumamış veya ilkokul mezunu olanlar sınav kaygısına göre, lise olanlar internete ulaşım durumuna göre ve üniversite ve üstü olanlar ise bölgeye göre dallanma oluşturmuştur.
5. Beşinci dallanma sınav kaygısı, internete ulaşım ve bölge değişkenlerine göre oluşmuştur. Sınav kaygısı yüksek olan öğrencilerin fen okuryazarlığı düşük iken sınav kaygısı düşük olanlar bölge değişkenine göre yeniden sınıflandırılmıştır. İnternete ulaşım imkânı olmayan öğrenciler, fen okuryazarlığı düşük olan gruba dahil edilmiş, internete ulaşım imkanı bulunanlar ise okula ait hissetme değişkenine göre yeni bir dallanma meydana getirmiştir. İç Anadolu, Karadeniz, Doğu ve Güneydoğu Anadolu

bölgeleri düşük fen okuryazarlığına sahiptir. Ege ve Akdeniz Bölgeleri'ndeki öğrencilerin fen okuryazarlığı daha yüksektir. Marmara Bölgesi için ise uyarlanabilir öğretim değişkenine göre yeni bir dallanma oluşmuştur.

6. Altıncı dallanma bölge, okula ait hissetme ve uyarlanabilir öğretim değişkenlerine göre oluşmuştur. Okula aitlik hissi düşük olan öğrencilerin fen okuryazarlığı daha yüksektir. Derslerin çoğunda uyarlanabilir öğretim kullanıldığında öğrencilerin fen okuryazarlık düzeyi daha yüksektir. Bölge değişkenine göre oluşan dallanmada ise Marmara ve Ege Bölgeleri'ndeki öğrencilerin fen okuryazarlık düzeyinin daha yüksek olduğu gözlenmiştir.

5. BÖLÜM

SONUÇ, TARTIŞMA VE ÖNERİLER

5.1. Sonuç ve Tartışma

Yaşanılan teknolojik gelişmelerle beraber son yıllarda veri madenciliği uygulamaları her sahada gittikçe yaygınlaşmaktadır. Bununla beraber eğitim alanında veri madenciliği uygulamalarının kullanımının istenilen düzeyde olmadığı düşünülmektedir. Eğitim alanının her boyutundan toplanan verileri veri madenciliği yöntemleri ile modellemenin, daha etkili eğitim politikaları oluşturulması, öğrencilere ilişkin biçok sınıflandırma, tahminleme ve kümeleme işlemlerinin yapılması, başarısızlık-okul bırakma gibi durumların önceden tespiti ya da eğitim ve öğrenmeye ilişkin çeşitli kuramların oluşturulması gibi birçok durumda eğitim ve öğretim süreçlerine katkı sağlayabileceği ve verimliliği artırılabileceği öngörülmektedir. Bu anlamda veri madenciliği yöntemlerinin eğitimde ölçme ve değrlendirme süreçlerine de yeni bir boyut kazandıracak potansiyele sahip olduğu söylenebilir. Tüm bu sebeplerle PISA 2015 Türkiye verileri ile yapılan bu çalışmanın hem algoritma performanslarının kıyaslanması hem de çok sınıflı sınıflandırma stratejilerinden olan OVA ve OVO stratejilerinin kullanılması bakımından önemli olduğu düşünülmektedir.

Araştırmada PISA 2015 veri seti üzerinde OVA ve OVO stratejileri altında bazı temel sınıflandırıcıların performanslarını kıyaslamak ve de bu stratejilerden hangisinin daha başarılı performansa sahip olduğunu ortaya koymak amaçlanmıştır. Araştırma temel (kuramsal) araştırma modelinde tasarlanmıştır. PISA 2015 anketleri ve başarı testleri yoluyla toplanan verilerin kullanıldığı çalışmada verilerin analizi WEKA paket programı Experimenter tezgâhı aracılığıyla yapılmıştır. Sonuçların ortaya konulmasında, değerlendirilmesinde ve yorumlanmasında çeşitli performans metriklerinden yararlanılmıştır.

Bu bölümde ise öncelikle çalışmaya ilişkin sonuçlar alt problemler çerçevesinde ortaya konulacak daha sonra sonuçlara ilişkin yorumlar yapılarak literatürden çeşitli çalışmalarla desteklenmiştir.

Birinci ve ikinci alt problemlere ilişkin sonuçlar değerlendirildiğinde; OVA stratejisi altında doğruluk ve ortalama hata değerleri bakımından LR algoritmasının, F değeri bakımından ise NB algoritmasının en başarılı algoritmalar olduğu görülmektedir. Her üç

metrik bakımından da en düşük performansı gösteren algoritma KNN algoritmasıdır. En az sayıda örnek içeren sınıftaki yedi örnekten iki tanesini doğru sınıflandıran KNN algoritması bu değerlendirme ölçüsü bakımından en başarılı algoritmadır. Diğer dört algoritma bu sınıfa ait hiçbir örneği doğru sınıflandıramamıştır. Kappa metriği bakımından OVA stratejisi altında en başarılı algoritmanın NB algoritması olduğu görülmektedir. Bu algoritmayı LR algoritması izlemektedir.

Üçüncü ve dördüncü alt probleme ilişkin sonuçlar değerlendirildiğinde; OVO stratejisi altında doğruluk ve ortalama hata metrikleri bakımından LR algoritmasının ve F metriği bakımından ise NB algoritmasının en yüksek performansa sahip olduğu görülmektedir. En düşük performansa sahip olan algoritma ise her üç metrik açısından da KNN algoritmasıdır. En az sayıda örnek içeren sınıftaki yedi örnekten iki tanesini doğru sınıflandıran KNN algoritması ve bu örneklerden bir tanesini doğru olarak sınıflandıran LR algoritması diğer algoritmalarla kıyaslandığında en başarılı algoritmalar olarak değerlendirilebilir. Kappa ölçüsü bakımından değerlendirildiğinde ise NB algoritması ve bu algoritmayı takiben LR algoritması en yüksek performansı sergilemiştir.

İlk dört alt probleme ilişkin sonuçlar birlikte değerlendirildiğinde doğruluk, hata ve kappa metriklerinin her üçü bakımından da, hem OVA hem de OVO stratejisi altında en başarılı algoritmaların LR ve NB algoritmaları olduğu görülmektedir. Veri setinin dengesiz olması, bağımlı değişkenin üç sınıf olarak bölünmesi ve veri seti ile ilgili diğer parametreler bu sonucu ortaya çıkarmış olabilir. Her iki strateji altında da en düşük performansa sahip algoritmanın KNN algoritması olduğu da görülmektedir. Yapılan çalışmada KNN algoritmasında örnekler arası uzaklıkların öklit uzaklığına bağlı olarak belirlenmesi ve veri setinde çoğunluğu oluşturan örneklerin birbirine yakın bir grup oluşturmaması, bu algoritmanın başarısının düşük olmasının sebebi olabilir. Buna karşılık en az örnekli sınıfı tahmin başarısı bakımından KNN algoritmasının nispeten daha başarılı olduğu da görülmektedir. En az örnek içeren sınıftaki örneklerin bağımsız değişkenlere ait değerlerinin dağılımı birbirine yakındır. Bu sebeple öklit uzaklığı bu sınıf için avantajlı bir parametre olmaktadır. Altman (1990)'a göre her iki strateji altında da DVM ve KNN algoritmalarının kappa değerleri 0.20 değerinin altındadır ve bu sonuçlar zayıf olarak değerlendirilir. Diğer algoritmaların kappa değerleri 0.20 ile 0.40 arasında değiştiğinden makul düzeyde olarak değerlendirilir. Bunun sebebi olarak veri

setinin dengesiz olması gösterilebileceği gibi algoritmanın formülasyonu da gösterilebilir. Algoritmalar daha fazla sayıda örnek içeren sınıflara bağlı olarak öğrenme gerçekleştirdiklerinden, çalışmada kullanılan algoritmaların kappa değerleri düşük çıkmış olabilir. En az sayıda veri içeren sınıfın doğru tahmin edilme performanslarının düşük olması da bu savı güçlendirmektedir. OVO stratejisi az örnekli sınıflarla karşılaştığında zorlanmakta ve algoritmalar ezberlemeye yatkın hale gelmektedir. OVA stratejisinde ise bir sınıf, diğer tüm sınıfların birleşmesiyle oluşturulan gruba karşı indüklendiğinden sınıf dengesizliğinin negatif etkisi artmaktadır. Bu durum her iki stratejide de hem algoritmaların performanslarını düşürmüş hem de en az örnekli sınıfın doğru tahminlenme oranını etkilemiştir.

Beşinci ve altıncı alt problemlere ilişkin sonuçlar incelendiğinde, ortalama hata değeri bakımından tüm algoritmalar OVA stratejisi altında daha başarılıdır. DVM algoritması doğruluk ve F metriklerinin her ikisi bakımından da OVO stratejisi altında daha başarılı iken NB algoritması doğruluk ve F değerleri bakımından OVA stratejisi altında daha başarılıdır. LR algoritması doğruluk değeri bakımından OVO stratejisi altında, F değeri bakımından OVA stratejisi altında daha başarılıdır. J-48 algoritması doğruluk değeri bakımından OVO stratejisi altında daha başarılı iken F metriği bakımından her iki stratejide de performansının aynı olduğu görülmüştür. KNN algoritması doğruluk değeri bakımından OVA stratejisi altında daha başarılı iken F metriği bakımından da her iki stratejide de performansının aynı olduğu gözlenmiştir. Algoritmalar doğruluk değeri ve hata ortalaması bakımından OVA stratejisi altında F metriği bakımından ise OVO stratejisi altında daha başarılıdır. Kappa metriği bakımından LR, NB ve KNN algoritmaları OVA stratejisi altında DVM ve J-48 algoritmaları OVO stratejisi altında daha başarlıdır. Bu metrik bakımından stratejilerin ortalama değerlerine bakıldığında ise OVO stratejisinin daha başarılı olduğu görülmektedir.

Literatürde algoritmaların ve stratejilerin karşılaştırılması amacıyla birçok çalışma yapılmıştır. Kullanılan veri seti, seçilen parametreler, karşılaştırılan teknik ve algoritmalar farklı olduğundan sonuçlar çeşitlilik göstermektedir. Bu çalışmaya paralel sonuçlar ortaya çıktığı gibi farklı sonuçlar ortaya koyan çalışmalara da ulaşılmıştır. Ulaşılan çalışmalardan Bulut (2016), dengesiz veri setleri üzerinde algoritmaları karşılaştırmak üzere yaptığı çalışmada LR algoritmasını en yüksek performansa sahip

olan algoritma olarak değerlendirmiştir. Çalışmasında 13 veri seti kullanmıştır ve LR algoritması bu 13 veri setinin 8'inde AUC metriği bakımından en yüksek performansı sergilemiştir. Çalışmada kullandığı diğer algoritmalar ise J-48, Naive Bayes, KNN ve DVM algoritmalarıdır. Bu çalışmada KNN ve DVM algoritmaları ise sadece birer veri setinde en yüksek performans göstermiş, en başarısız algoritmalar olarak değerlendirilmişlerdir. Sabzevari vd. (2007) kredi skorlaması tahmini üzerine yaptıkları araştırmada istatistiksel tekniklerden LR algoritmasını, veri madenciliği tekniklerinden ise Bagging algoritmasını sınıflandırma doğruluğu metriği bakımından en başarılı bulmuşlardır. Sharma ve Sahni (2011) yürüttükleri çalışmada, çeşitli karar ağacı algoritmalarını karşılaştırmışlar ve doğruluk metriği bakımından en başarılı karar ağacı algoritmasının J48 algoritması olduğunu gözlemlemişlerdir. Coşkun (2010); NB, LR, K-Star ve J-48 algoritmalarını karşılaştırdığı çalışmasında doğruluk ve F metrikleri bakımından en başarılı algoritmanın J-48 algoritması olduğunu belirtmiştir. Hassan vd. (2016) yaptıkları araştırmada, içinde LR, K-star ve J-48 algoritmalarının da bulunduğu 10 farklı algoritmadan Bayes Network algoritmasını, gerçek pozitif oranı, yanlış pozitif oranı, kesinlik, duyarlılık, F değeri, ROC alanı ve hesaplama süresi gibi birçok metrik bakımından en başarılı bulmuşlardır. Kahraman vd. (2004), YSA ile DVM algoritmalarını karşılaştırdıkları çalışmalarında sınıflandırma başarısı ve hesaplama süresi ölçüleri bakımından DVM algoritmasının YSA algoritmasından çok daha yüksek performans gösterdiğini belirtmişlerdir. Akçapınar (2014) yürüttüğü çalışmada, veri setindeki değişkenlerin ifade ediliş türlerini değiştirerek (kategorik, sürekli gibi) ve özellik seçme yöntemlerini kullanarak birçok sınama yapmış, tüm sınamalarda KNN algoritmasının tüm metrikler bakımından başarılı performansa sahip olduğunu gözlemlemiştir. Yurdakul (2015) çalışmasında J-Rip, KNN, J-48 ve NB algoritmalarının performanslarını doğruluk metriğine göre kıyaslamış ve en düşük performans gösteren algoritmanın NB algoritması olduğunu ifade etmiştir. Cong, Yang, Lv ve Xue (2009) çalışmalarında romatoid artrit hastalığı ile ilgili olduğu düşünülen bir enzimi belirlemek üzere makine öğrenmesi algoritmalarının performanslarını sınamış ve KNN algoritması %98.32 başarı performansıyla en başarılı algoritma olarak belirlemiştir. Varpa vd. (2011), KNN ve DVM algoritmalarını OVO ve OVA stratejileri altında kıyasladıkları çalışmalarında OVO stratejisinin OVA stratejisinden daha başarılı

olduğunu ayrıca OVO+KNN birleşiminin en yüksek performansa sahip olduğunu tespit etmişlerdir.

Algoritmaların performanslarının karşılaştırıldığı araştırmalar sonuçları itibariyle çok çeşitlilik göstermektedir. Kullanılan veri seti, karşılaştırılan algoritmalar ve seçilen parametreler farklı olduğundan bu çeşitlilik oldukça normal bir durum olarak değerlendirilmektedir. Fakat bu farklılıklardan dolayı bu çalışmaları eleştiren araştırmacılar da bulunmaktadır. Hand (2006) parametreler, veri seti üzerinde yapılan önişlemler gibi birçok durumun farklı olması sebebiyle bu tür karşılaştırma çalışmalarının illüzyon yarattığını ifade etmiştir. Ayrıca geliştirilen yeni algoritmaların daha başarılı performans göstermesi adına yanlı davranılması ve bu çalışmalarda kullanılan veri setlerinin gerçek veri seti olmaması, performans karşılaştırma çalışmalarının eleştirilmesine sebep olmaktadır.

Fakat tüm bu eleştirilere rağmen algoritmaların karşılaştırılmasına yönelik yapılan çalışmalar literatürde kabul görmüştür ve çalışılan bir alan olmaya devam etmektedir. Veri seti- başarılı algoritma şeklinde ikililerin oluşabileceği düşüncesi bu tür çalışmaların üretilmesine sebep olmakta ve performans karşılaştırma çalışmalarının zamanla veri seti-başarılı algoritma eşleşmelerini belirginleştireceği düşünülmektedir.

Yedinci alt probleme ilişkin sonuçlar değerlendirildiğinde fen okuryazarlığını etkileyen değişkenlerin kaynak kitaba ulaşım, sınıf düzeyi, öğretmen desteği, baba eğitim durumu, sınav kaygısı, internete ulaşım, bölge, okula ait hissetme ve uyarlanabilir öğretim değişkenleri olduğu görülmektedir. Yapılan diğer çalışmalarda da öğrencilerin baba eğitim durumları (Alomar, 2006; Özer ve Anıl, 2011; Schmitt, Sacco, Ramey, Ramey ve Chan, 1999; Turmo, 2004, Taningco ve Pachon, 2008), ulaşabildikleri yardımcı kitapların sayısı (Bos ve Kuiper, 1999; Erbaş, 2005; Özer ve Anıl, 2011; Şaşmazel, 2006), internete erişim sağlayabilmeleri (Christman ve Badgett, 1999; Hativa, 1994, Özer ve Anıl, 2011) değişkenlerinin fen okuryazarlığı üzerinde etkili olduğu görülmüştür. Ayrıca, Yıldırım, Karakurt ve Hacıhasanoğlu (2009), Yurttaş ve Yetkin (2003) çalışmalarında daha üst sınıf düzeylerinde problem çözme becerilerinin düştüğünü tespit etmiş bunu da öğrencilerin öğrendikleri bilgileri problem çözmede kullanamamasına bağlamışlardır. Öğrencinin sosyoekonomik statüsü, ev çevresi, okul çevresi, faklı kültürler, öğretmen niteliği gibi değişkenler eğitimde başarıyı

etkileyen değişkenlerden bazılarıdır (Aydoğdu, 2006). Türkiye'de bölgelere göre sosyoekonomik statü, çevre, kültür gibi değişkenlerin farklılaştığı düşünüldüğünde bölge değişkeninin fen okuryazarlığı üzerinde önemli olması anlaşılabilir bir durum olmaktadır. (Fidan, 1986; aktaran Açıkgöz, 2003) Öğretmenlerin ders işleme niteliğinin öğrenci başarısını etkilediğini ortaya koymuştur. Uyarlanabilir öğretim, dersi sınıfın ihtiyacı ve bilgisine göre uyarlama, bir öğrenci, herhangi bir konuyu veya bir görevi anlamakta güçlük çektiğinde bu öğrenciye özel olarak yardımcı olma, birçok öğrencinin anlamadığı ders veya konularda dersin yapısını değiştirme gibi durumları içermektedir. Bu beceriler öğretmenin nitelikli olması ile yakından ilişkilidir. Bu sebeple, çalışmada ortaya konulan uyarlanabilir öğretimin kullanıldığı sınıflarda öğrencilerin fen okuryazarlığının yüksek olması bulgusu literatür ile uyumludur denilebilir. Burns (2004), Hancock (2001), Culler ve Holahan (1980) çalışmalarında sınav kaygısının akademik performansı olumsuz etkilediğini, yüksek kaygılı öğrencilerin düşük kaygılı öğrencilere göre daha zayıf performans gösterdiklerini ortaya koymuşlardır. Daha önce yapılmış birçok çalışmada okula aitlik hissi ile akademik başarının pozitif yönde ilişkili olduğu gösterilmiştir. Bu çalışmada sınav kaygısına ilişkin bu tip bir sonuç ortaya konulmamıştır.

Oluşturulan karar ağacından elde edilen bulgulardan hareketle fen okuryazarlığına etkisi olan değişkenlerin eğitim öğretim sürecinde daha çok üzerinde durulması gerektiği söylenebilir. Öğrencilerin yardımcı kitaplara ve internete ulaşım imkanları artırılmalı, bölgeler arasındaki farklılıklar giderilmeye çalışılmalı, öğrencilerdeki sınav kaygısını azaltmaya yönelik çalışmalar yapılmalı ve öğretmenler fen derslerinde uyarlanabilir öğretime ilişkin yapılabilecek uygulamalara ağırlık vermelidir.

5.2. Öneriler

1. Aynı veri seti üzerinde farklı algoritmaların (Bagging, boosting, diğer bayesyen, farklı karar ağacı vb.) performansları kıyaslanabilir.
2. Benzer bir çalışma anketle elde edilen eğitim verileri, ÖSYM'nin yaptığı sınavlar, uzaktan eğitim sistemleri veya farklı uluslararası sınavlar (TIMMS, PIRLS vb.) üzerinden toplanan veri setleri ile tekrarlanabilir.

3. Bu çalışmada veri madenciliği alanında en çok kullanılan programlardan biri olan Weka ile çalışılmıştır. Bundan sonra yapılacak çalışmalarda farklı veri madenciliği programları kullanılarak elde edilen sonuçlar karşılaştırılabilir.
4. Eğitim fakültesindeki öğrencilerin türkçe, matematik ve fen derslerindeki başarılarını yordamak amacıyla okulun veri tabanından yararlanılarak; veri madenciliğine dayalı çalışmalar yapılabilir.
5. Elde edilen sonuçları sadece karışıklık matrisinden elde edilen değerlere göre yorumlamak yerine AUC metriği gibi farklı performans metriklerinden yararlanılabilir.
6. Araştırmada kullanılan OVA ve OVO stratejileri dışındaki farklı ayrıştırma stratejileri ve majority voting (çoğunluk oylaması) yöntemi dışındaki sonuçları birleştirme stratejileri dışında başka stratejiler kullanılabilir.
7. Çalışmada test seçeneklerinden 10 katlı çapraz geçerleme yöntemi kullanılmıştır. Bundan sonraki çalışmalarda holdout metodu gibi farklı test seçenekleri kullanılabilir.
8. Çalışmada kullanılan algoritmalar değiştirilebileceği gibi algoritmaların parametreleri de değiştirilebilir. Örneğin KNN algoritmasında uzaklık, Öklit uzaklığı ile değil de Manhattan uzaklığı gibi kullanılabilecek farklı yöntemlerle çalışma tekrarlanabilir. Ya da DVM algoritmasında farklı çekirdek fonksiyonları kullanılabilir.
9. Veri seti üzerinde yapılacak farklı önişlemler ile çalışma tekrarlanabilir. Örneğin oversampling, undersampling gibi verileri dengelemek için geliştirilen yöntemler kullanılabilir. Boş verilerin silinmesi yerine çeşitli yöntemlerle (örneğin karar ağaçları) atama yapılabilir. Değişkenler farklı türlerde düzenlenerek (örneğin kategorik değil de nicel bırakılarak) analiz tekrarlanabilir. Daha çok değişken veri setine dâhil edilebilir ya da, özellik seçimi yöntemleri ile değişkenler azaltılarak denenebilir.

KAYNAKÇA

Açıkgöz, Ü. K. (2003). *Etkili öğrenme ve öğretme.* İzmir: Eğitim Dünyası Yayınları

Adriaans, P. ve Zantinge, D. (1997). *Data mining.* Boston: USA Addison Wesley Longman Publishing.

Adnan, M. N. ve Islam, M. Z. (2015). One-vs-all binarization technique in the context of random forest. *European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning* (ESANN), 22-24 April 2015, Bruges, Belgium.

Aha, D. W., Kibler, D. ve Goldstone, R. L.(1991). Instance-based learning algorithms. *Machine Learning*, *6*(1), 37-66.

Akbulut, S. (2006). *Veri madenciliği teknikleri ile bir kozmetik markanın ayrılan müşteri analizi ve müşteri segmentasyonu.* Gazi Üniversitesi Fen Bilimleri Enstitüsü: Yayımlanmış yüksek lisans tezi.

Akçapınar, G. (2014). *Çevrimiçi öğrenme ortamındaki etkileşim verilerine göre öğrencilerin akademik performanslarının veri madenciliği yaklaşımı ile modellenmesi*. Hacettepe Üniversitesi Eğitim Bilimleri Enstitüsü: Yayımlanmış doktora tezi.

Akpınar, H. (2000). Veri tabanlarında bilgi keşfi ve veri madenciliği. *İstanbul Üniversitesi İşletme Fakültesi Dergisi*, *29*(1), 1-22.

Allwein, E. L., Schapire, R. E. ve Singer, Y. (2000). Reducing multiclass to binary: A unifying approach for margin classifiers. *Journal of Machine Learning Research, 1*, 113-141.

Alomar, B. O. (2006). Personal and family paths to pupil achievement. *Social Behavior and Personality: An international journal, 34*(8), 907-922.

Altıntaş, T. (2006). *Veri madenciliği metotlarından olan kümeleme algoritmalarının uygulamalı etkinlik analizi*. Sakarya Üniversitesi Fen Bilimleri Enstitüsü: Yayımlanmış yüksek lisans tezi.

Altman, D. G. (1990). *Practical statistics for medical research*. USA: CRC press.

Aly, M. (2005). Survey on multiclass classification methods. *Artificial Neural Network, 19*, 1-9.

Arruti, A., Mendialdua, I., Sierra, B., Lazkano, E. ve Jauregi, E. (2014). NewOneVersusOneAll method: NOV@. *Expert Systems with Applications, 41*(14), 6251-6260.

Aydın, S. ve Özkul, A. E. (2015). Veri madenciliği ve anadolu üniversitesi açiköğretim sisteminde bir uygulama. *Journal of Research in Education and Teaching, 4*(3), 36-44.

Aydoğan, E. (2008), *Veri madenciliğinde sınıflandırma problemleri için evrimsel algoritma tabanlı yeni bir yaklasım: Rough-Mep algoritması*. Gazi Üniversitesi Fen Bilimleri Enstitüsü: Yayımlanmamış doktora tezi.

Aydoğdu, B. (2006). *İlköğretim fen ve teknoloji dersinde bilimsel süreç becerilerini etkileyen değişkenlerin belirlenmesi*. Dokuz Eylül Üniversitesi Eğitim Bilimleri Enstitüsü: Yayımlanmış yüksek lisans tezi.

Babadağ, K. (2006). Zeki veri madenciliği: Ham veriden altın bilgiye ulaşma yöntemleri. *Industrial Application Software*, 85-87.

Batista, G. E. A. P. A. ve Silva, D. F. (2009). How k-nearest neighbor parameters affect its performance. *10th Argentine Symposiumon Artificial Intelligence (ASAI 2009),* 24-25 August 2009, Mar Del Plata, Argentina.

Bhatia, N. ve Vandana (2010). Survey of nearest neighbor techniques. *International Journal of Computer Science and Information Security, 8*(2), 302-305.

Bin Othman, M. F. Ve Yau, T. M. S. (2007). Comparison of different classification techniques using WEKA for breast cancer. *3rd Kuala Lumpur International Conference on Biomedical Engineering,* 11-14 December 2006, Kuala Lumpur, Malaysia.

Bos, K. ve Kuiper, W. (1999). Modelling TIMSS data in a European comparative perspective: Exploring influencing factors on achievement in mathematics in grade 8. *Educational Research and Evaluation*, *5*(2), 157-179.

Bounsaythip, C. ve Esa, R. R. (2001). "Overview of data mining for customer behavior modeling. *VTT Information Technology Research Report, 1*, 1-53.

Breiman, L., Friedman J. H., Olshen R. A. ve Stone, C. J. (1984). *Classification and regression trees.* CA: Wadsworth.

Bulut, F. (2016). *Performance evaluations of supervised learners on imbalanced datasets,* 26-27 April 2016, Istanbul, Turkey.

Burns, D. J. (2004). Anxiety at the time of the final exam: Relationships with expectations and performance. *Journal of Education for Business, 80*(2), 119-124.

Büyüköztürk, S., Kılıç-Çakmak. E., Akgün, O. E., Karadeniz, S. ve Demirel, F. (2016). *Bilimsel araştırma yöntemleri* (22.Baskı). Ankara: Pegem Akademi.

Cai, Y. D ve Chou, K. C. (2003). Nearest neighbour algorithm for predicting protein subcellular location by combining functional domain composition and pseudoamino acid composition. *Biochemical and Biophysical Research Communications, 305*(2), 407-411.

Caruana, R. ve Niculescu-Mizil, A. (2006). An empirical comparison of supervised learning algorithms. *Proceedings of the 23rd International Conference on Machine Learning,* 25-29 June 2006, Pittsburgh, Pennsylvania.

Christman, E. ve Badgett, J. (1999). A comparative analysis of the effects of computer-assisted instruction on student achievement in differing science and demographical areas. *Journals of Computers in Mathematics and Science Teaching, 18,* 135-143

Cong, Y., Yang X. G., Lv, W. ve Xue, Y. (2009). Prediction of novel and selective TNF-alpha converting enzyme (TACE) inhibitors and characterization of correlative molecular descriptors by machine learning approaches. *Journal of Molecular Graphics and Modelling, 28*(3), 236-244.

Coomans, D. ve Massart, D. L. (1982). Alternative k-nearest neighbour rules in supervised pattern recognitition: Part 1. K-nearest neighbour classification by using alternative voting rules. *Analytica Chimica Acta, 136,* 15-27.

Coşkun, C. (2010). *Veri madenciliği algoritmalarının karşılaştırılması*. Dicle Üniversitesi Fen Bilimleri Enstitüsü: Yayımlanmış yüksek lisans tezi.

Coşkun, C. ve Baykal, A. (2011). *Veri madenciliğinde sınıflandırma algoritmalarının bir örnek üzerinde karşılaştırılması.* Malatya: Akademik Bilişim.

Cover, T. M. ve Hart, P. E. (1967). Nearest neighbor pattern classification. *IEEE Transactions on Information Theory, 13*(1), 21–27.

Culler, R. E. ve Holahan, C. J. (1980). Test anxiety and academic performance: The effects of study-related behaviors. *Journal of educational psychology*, *72*(1), 16-20.

Çölkesen R. (2013). *Veri yapıları ve algoritmalar.* İstanbul: Papatya Yayıncılık.

Doad, P.K. ve Bartere, M.M. (2013). A review: Study of various clustering techniques. *International Journal of Engineering Research and Technology, 2*(11), 3141-3145.

Doğancı, S., Yıldırım, V., Yeşildal, F., Erol, G., Kadan, M., Özkan, G. Avcu, F. ve Özgürtaş, T. (2015). Comparison of angiogenic and proliferative effects of three commonly used agents for pulmonary artery hypertension (sildenafil, iloprost, bosentan): is angiogenesis always beneficial? *European Review for Medical and Pharmacological Sciences, 19,* 1900-1906

Duda, R. O., Hart, P. E. ve Stork, D. G. (2000). *Pattern classification* (2nd Edition). New Jersey: John Wiley and Sons.

Emre, İ. E. Ve Erol, Ç. S. (2017). Veri analizinde istatistik mi veri madenciliği mi?. *International Journal of Informatics Technologies, 10*(2), 161-167.

Erbaş, K. C. (2005). *Uluslararası öğrenci başarı belirleme programında (PISA) Türkiye'de fen okuryazarlığını etkileyen faktörler*. Ortadoğu Teknik Üniversitesi Fen Bilimleri Enstitüsü: Yayımlanmış yüksek lisans tezi.

Fayyad, U., Piatetsky-Shapiro, G., Smyth, P. ve Uthurusamy, R. (Eds.) (1996). *Advances in knowledge discovery and data mining.* Cambridge: MIT Press/AAAI Press.

Fernández, A., López, V., Galar, M., Del Jesus, M. J. ve Herrera, F. (2013). Analysing the classification of imbalanced data-sets with multiple classes: Binarization techniques and ad-hoc approaches. *Knowledge-Based Systems, 42,* 97-110.

Field, A. (2005). *Discovering statistics using SPSS* (2nd Edition). London: Sage Publications.

Friedman, J. (1996). *Another approach to polychotomous classification.* Technical report, Department of Statistics, Stanford University, Stanford, CA.

Friedman, J., Hastie, T. ve Tibshirani, R. (2001). *The elements of statistical learning*. New York: Springer Series in Statistics.

Galar, M., Fernández, A., Barrenechea, E., Bustince, H. ve Herrera, F. (2011). An overview of ensemble methods for binary classifiers in multi-class problems: Experimental study on one-vs-one and one-vs-all schemes. *Pattern Recognition*, *44*(8), 1761-1776.

Galar, M., Fernández, A., Barrenechea, E. ve Herrera, F. (2015). DRCW-OVO: distance-based relative competence weighting combination for one-vs-one strategy in multi-class problems. *Pattern recognition*, *48*(1), 28-42.

Ganesh, S. (2002). Data mining: Should it be included in the statistics' curriculum?. *The Sixt International Conference on Teaching Statistics*, 7-12 July 2002, Cape Town, South Africa.

Garcia, S., Fernández, A., Luengo, J. ve Herrera, F. (2009). A study of statistical techniques and performance measures for genetics-based machine learning: Accuracy and interpretability. *Soft Computing*, *13*(10), 959-977.

Gartner, T., Lloyd, J. W. ve Flach, P. A. (2004). Kernels and distances for structured data. *Machine Learning*, *57*(3), 205-232.

Güner, E. S. (2008). *Türkçe için derlem tabanlı bir anafor çözümleme çalışması*. Trakya Üniversitesi Fen Bilimleri Enstitüsü: Yayımlanmış yüksek lisans tezi.

Hair, J. F., Black, W. C., Babin, B., Anderson, R. E. ve Tatham, R. L. (2006). *Multivariate data analysis* (6th Edition). Upper Saddle River, NJ: Prentice-Hall.

Han, J., Kamber, M. ve Pei, J. (2011). *Data mining: Concepts and techniques* (3rd Edition). Burlington: Morgan Kaufmann Publishers.

Hancock, D. R. (2001). Effects of test anxiety and evaluative threat on students' achievement and motivation. *The Journal of Educational Research*, *94*(5), 284-290.

Hand, D. J. (2006). Classifier Technology and the Illusion of Progress. *Institute of Mathematical Statistics in Statistical Science, 21*(1), 1-15.

Hassan, M. A., Shehab, M. E. ve Hamed, E. M. R. (2016). A comparative study of classification algorithms in e-health environment. *Sixth International Conference on Digital Information Processing and Communications (ICDIPC),* 21 - 23 April 2016, Beirut, Lebanon.

Hastie, T. ve Tibshirani, R. (1998). Classification by pairwise coupling. *The Annals of Statistics, 26*(2), 451–471.

Hativa, N. (1994). What you design is not what you get (WYDINWYG): Cognitive, affective, and social impacts of learning with ILS—An integration of findings from six-years of qualitative and quantitative studies. *International Journal of Educational Research, 21*(1), 81-111.

Hung, S. Y., Yen, D. C. ve Wang, H. Y. (2006). Applying data mining to telecom churn management. *Expert Systems with Applications, 31*, 515–524.

IEDMS, (2016). Educational data mining. [Çevrim-içi: http://www.educationaldatamining.org/], Erişim Tarihi: 26.12.2017.

IEEE, (2016). *Distances in classification.* [Çevrim-içi: http://www.ieee.ma/uaesb/pdf/distances-in-classification.pdf], Erişim tarihi: 30.09.2018.

Kahraman, F., Çapar, A., Ayvacı, A., Demirel, H. ve Gökmen, M. (2004). Comparison of DVM and ANN performance for handwritten character classification. *Proceedings of the IEEE 12th Signal Processing and Communications Applications Conference,* 30-30 April 2004, Kusadasi, Turkey

Kalıkov, A., (2006), *Veri madenciliği ve bir e-ticaret uygulaması*. Gazi Üniversitesi Fen Bilimleri Enstitüsü: Yayımlanmış yüksek lisans tezi.

Kalıpsız, O. ve Cihan, P. (2015). Öğrenci proje anketlerini sınıflandırmada en iyi algoritmanın belirlenmesi. *Türkiye Bilişim Vakfı Bilgisayar Bilimleri ve Mühendisliği Dergisi*, *8*(1), 41-49.

Karasar, N. (2016). *Bilimsel arastirma yontemi.* Ankara: Nobel Yayin Dağıtım.

Kaur, P., Singh, M. ve Josan, G. S. (2015). Classification and prediction based data mining algorithms to predict slow learners in education sector. *Procedia Computer Science*, *57*, 500-508.

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

Kayri, M. ve Çokluk, Ö. (2010). Using multinomial logistic regression analysis in artificial neural network: An application. *Ozean Journal of Applied Sciences*, *3*(2), 259-268.

Keller, J. M., Gray, M. R. ve Givens, J. A. (1985). A fuzzy K-nearest neighbor algorithm. *IEEE Transactions on Systems, Man and Cybernetics*, *15*(4), 580-585.

Kılıç, S. (2000). *Lojistik regresyon analizi ve pazarlama araştırmalarında bir uygulama.* İstanbul Teknk Üniversitesi Fen Bilimleri Enstitüsü: Yayımlanmamış yüksek lisans tezi.

King, R. D., Feng, C. ve Sutherland, A. (1995). StatLog: Comparison of Classification Algorithms on Large Real-World Problems. *Applied Artificial Intelligence*, *9*(3), 289-333.

Knerr, S., Personnaz, L. ve Dreyfus, G. (1990). Single-layer learning revisited: A stepwise procedure for building and training a neural network. In F. Fogelman-Soulie ve J. Herault (Eds.). *Neurocomputing: Algorithms, architectures and applications, 41–50.* Heidelberg: Springer.

Kuonen, D. (2004). Data mining and Statistics: What is the connection?. *The Data Administration Newsletter*, *30*, 1-6.

Kumar, D., ve Bhardwaj, D. (2011). Rise of data mining: current and future application areas. *International Journal of Computer Science Issues (IJCSI)*, *8*(5), 256.

Koyuncugil, A. ve Özgülbaş, N. (2009). Veri madenciliği: Tıp ve sağlık hizmetlerinde kullanımı ve uygulamaları. *Bilişim Teknolojileri Dergisi*, *2*(2), 21-32.

Köktürk, F. (2012). *K-en yakın komşuluk, yapay sinir ağları ve karar ağaçları yöntemlerinin sınıflandırma başarılarının karşılaştırılması*. Bülent Ecevit Üniversitesi Sağlık Bilimleri Enstitüsü: Yayımlanmış doktora tezi.

Kresse, W. ve Danko, D. M. (2012). *Springer handbook of geographic information.* Berlin: Springer-Verlag.

Landis, J. R. ve Koch, G. G. (1977). The measurement of observer agreement for categorical data. *Biometrics, 33* (1) , 159-174.

Liu, B., Hao, Z. ve Tsang, E. C. (2008). Nesting one-against-one algorithm based on DVMs for pattern classification. *IEEE Transactions on Neural Networks*, *19*(12), 2044-2052.

Liu, H. ve Zhang, S. (2012). Noisy data elimination using mutual k-nearest neighbor for classification mining. *Journal of Systems and Software*, *85*(5), 1067-1074.

Lorena, A. C., De Carvalho, A. C. ve Gama, J. M. P. (2008). A review on the combination of binary classifiers in multiclass problems. *Artificial Intelligence Review*, *30*(1), 19-37.

Mannila, H. (1996). Data mining: Machine learning, statistics, and databases. *SSDBM '96 Proceedings of the Eighth International Conference on Scientific and Statistical Database Management*, 2-9 Temmuz 1996, Stockholm.

Mao, C., Hu, B., Wang, M. ve Moore, P. (2015). Learning from neighborhood for classification with local distribution characteristics. *IEEE International Joint Conference on Neural Networks (IJCNN)*, 12-17 July 2015, Killarney, Ireland.

Mayoraz, E. ve Moreira, M. (1996). On the decomposition of polychotomies into dichotomies. *Proceedings of the Fourteenth International Conference on Machine Learning (ICML)*, 08 – 12 July 1997, Nashville, Tennessee, USA,

MEB (2016). *Uluslararası öğrenci değerlendirme programı (PISA).* [Çevrim-içi: http://pisa.meb.gov.tr/wp-content/uploads/2014/11/PISA2015_UlusalRapor.pdf], Erişim Tarihi: 13.01.2018.

Mertler, C.A. and Vannatta, R.A. (2005) *Advanced and multivariate statistical methods: Practical application and interpretation* (3rd Edition). Los Angeles: Pyrczak.

Mitchell, T. (1997). *Machine Learning.* New York: McGraw Hill.

Oğuzlar, A. (2003). Veri ön işleme. *Erciyes Üniversitesi İktisadi ve İdari Bilimler Fakültesi Dergisi,* 21, 67-76.

Özbay, Ö. (2015). Veri madenciliği kavrami ve eğitimde veri madenciliği uygulamalari. *Uluslararası Eğitim Bilimleri Dergisi*, *4*(3), 262-272.

Özer, Y. ve Anıl, D. (2011). Öğrencilerin fen ve matematik başarılarını etkileyen faktörlerin yapısal eşitlik modeli ile incelenmesi. *Hacettepe Üniversitesi Eğitim Fakültesi Dergisi*, *41*(41), 313-324

Özkan, Y. (2016). *Veri madenciliği yöntemleri* (3. Baskı). İstanbul: Papatya Yayıncılık Eğitim.

Pal, A. K. ve Pal, S. (2013). Analysis and mining of educational data for predicting the performance of students. *International Journal of Electronics Communication and Computer Engineering*, *4*(5), 1560-1565.

Pal, M. ve Mather, P. M. (2003). An assessment of the effectiveness of decision tree methods for land cover classification. *Remote Sensing of Environment*, *86*(4), 554-565.

Passerini, A., Pontil, M. ve Frasconi, P. (2004). New results on error correcting output codes of kernel machines. *IEEE Transactions on Neural Networks and Learning Systems*, *15*(1), 45-54.

Polat, K. ve Güneş, S. (2007). Classification of epileptiform EEG using a hybrid system based on decision tree classifier and fast Fourier transform. *Applied Mathematics and Computation*, *187*(2), 1017-1026.

Qiu, X. Y., Kang, K. ve Zhang, H. X. (2008). Selection of kernel parameters for KNN. *IEEE International Joint Conference on Neural Networks (IJCNN),* 1-8 June 2008, Hong Kong, China

Quinlan, J. R. (1987). Simplifying decision trees. *International Journal of Man-Machine Studies*, *27*(3), 221-234.

Quinlan, J. R. (1993). *C4. 5: Programming for machine learning*. CA: Morgan Kauffmann

Rajavarman, V. N. ve Rajagopalan, S. P. (2007). Comparison between traditional data mining techniques and entropy-based adaptive genetic algorithm for learning classification rules. *International Journal of Soft Computing, 2*(4), 555-561.

Rocha, A. ve Goldenstein, S. K. (2014). Multiclass from binary: Expanding one-versus-all, one-versus-one and ecoc-based approaches. *IEEE Transactions on Neural Networks and Learning Systems, 25* (2), 289-302.

Romero, C., Ventura, S., Espejo, P. G. ve Hervás, C. (2008). Data mining algorithms to classify students. *The 1st International Conference on Educational Data Mining*, 20-21 June 2008, Montréal, Québec, Canada.

Sabzevari, H., Soleymani, M. ve Noorbakhsh, E. (2007). A comparison between statistical and data mining methods for credit scoring in case of limited available data. *Proceedings of the 3rd CRC Credit Scoring Conference*, 22-25 July 2007, Edinburgh, UK.

Safavian S. R. ve Landgrebe D. (1991). A survey of decision tree classifier methodology. *IEEE Transactions on Systems, Man, and Cybernetics, 21*(3), 660-674.

Savaş, S., Topaloğlu, N. ve Yılmaz, M. (2012). Veri madenciliği ve Türkiye'deki uygulama örnekleri. *İstanbul Ticaret Üniversitesi Fen Bilimleri Dergisi, 11*(21), 1-23.

Schmitt, N., Sacco, J. M., Ramey, S., Ramey, C. ve Chan, D. (1999). Parental employment, school climate, and children's academic and social development. *Journal of applied psychology*, *84*(5), 737-753.

Sen, A., Islam, M. M., Murase, K. ve Yao, X. (2016). Binarization with boosting and oversampling for multiclass classification. *IEEE Transactions on Cybernetics*, *46*(5), 1078-1091.

Sharma, A. K. ve Sahni, S. (2011). A comparative study of classification algorithms for spam email data analysis. *International Journal on Computer Science and Engineering*, *3*(5), 1890-1895.

Sharma, T. C. ve Jain, M. (2013). WEKA approach for comparative study of classification algorithm. *International Journal of Advanced Research in Computer and Communication Engineering, 2*(4), 1925-1931.

Shmueli, G., Patel, N. R. ve Bruce, P. C. (2010). *Data mining for business analytics: Concepts, techniques, and applications with XLMiner*. New Jersey: John Wiley and Sons.

Silahtaroğlu, G. (2016). *Veri madenciliği.* İstanbul: Papatya Yayınlar.

Şaşmazel, G. A. (2006). *Uluslararası öğrenci başarı belirleme programında (PISA) Türk öğrencilerin fen bilgisi başarılarını etkileyen faktörler*. Haccettepe Üniversitesi Eğitim Bilimleri Enstitüsü: Yayımlanmamış yüksek lisans tezi.

Şeker, S. E. (2016). *WEKA ile veri madenciliği*. [Çevrim-içi: https://www.kobo.com/tr/tr/ebook/weka-ile-veri-madenciligi], Erişim Tarihi: 30.09.2018.

Şencan, H. (2005). *Sosyal ve davranışsal ölçmelerde güvenirlik ve geçerlik.* Ankara: Sözkesen Matbaacılık.

Tabachnick, B. G. ve Fidell, L. S. (2007). *Using multivariate statistics* (5th Edition). Boston: Allyn & Bacon/Pearson Education.

Taningco, M. T. V. ve Pachon, H. P. (2008). Computer Use, Parental Expectations and Latino Academic Achievement. *Tomas Rivera Policy Institute*.

Taruna, S. ve Pandey, M. (2014). An empirical analysis of classification techniques for predicting academic performance. *IEEE International Advance Computing Conference (IACC),* 21-22 February 2014, Gurgaon, India.

Taşcı, E. ve Onan, A. (2016). K-en yakın komşu algoritması parametrelerinin sınıflandırma performansı üzerine etkisinin incelenmesi. 18. *Akademik Bilişim Konferansı,* 30-5 Şubat 2016, Aydın, Türkiye.

Tax, D. M. ve Duin, R. P. (2002). Using two-class classifiers for multiclass classification. *Proceedings 16th International Conference on Pattern Recognition,* 11-15 August 2002, Quebec City, Canada.

Turmo, A. (2004). Scientific literacy and socio-economic background among 15-year-olds—A Nordic Perspective Scandinavian. *Scandinavian Journal of Educational Research, 48*(3), 287-305.

Tüzüntürk, S. (2010). Veri madenciliği ve istatistik. *Uludağ Üniversitesi İktisadi ve İdari Bilimler Fakültesi Dergisi, 29*(1), 65-90.

Vahaplar, A. (2003). *Bir coğrafi veri madenciliği uygulaması*. Ege Üniversitesi Fen Bilimler Enstitüsü: Yayımlanmamış yüksek lisans tezi.

Varpa, K., Joutsijoki, H., Iltanen, K. ve Juhola, M. (2011). Applying one-vs-one and one-vs-all classifiers in k-nearest neighbour method and support vector machines to an otoneurological multi-class problem. *Studies in Health Technology and Informatics, 169*, 579–583.

Yıldırım, A., Karakurt, P., Hacıhasanoğlu, R. (2009) Comparison of the problem solving skills with feeling and expression of anger in nursing students. *Atatürk Üniversitesi Hemşirelik Yüksekokulu Dergisi, 12*(1), 57-65.

Yurdakul, S. (2015). *Veri madenciliği ile lise öğrenci performanslarının değerlendirilmesi*. Kırıkkale Üniversitesi Fen Bilimleri Enstitüsü: Yayımlanmış yüksek lisans tezi.

Yurttaş, A. ve Yetkin, A. (2003) Sağlık yüksekokulu öğrencilerinin empatik beceri ile problem çözme becerilerinin karşılaştırılması. *Atatürk Üniversitesi Hemşirelik Yüksekokulu Dergisi, 6*(1), 1-13.

Wu, X., Kumar, V., Quinlann, J. R., Ghosh, J., Yang, Q., Motoda, H., McLachlan, G. J., Angus, N., Bing, L., Yu, P. S., Zhou, Z. H., Steinbach, M., Hand, D. J. ve Steinberg, D. (2008). Top 10 algorithms in data mining. *Knowledge of Information Systems, 14*(1), 1-37.

Xu, G., Zong, Y. ve Yang, Z. (2013). *Applied data mining.* New York: CRC Press.

Zaiane, O. R. ve Luo, J. (2001). Towards evaluating learners' behaviour in a web-based distance learning environment. *Proceedings IEEE International Conference on Advanced Learning Technologies (ICALT 2001),* 6-8 August 2001, Madison, WI, USA,

Zhang H. (2004). The optimality of NB. *Proceedings of the Seventeenth International Florida Artificial Intelligence Research Society Conference (FLAIRS 2004),* 25-29 July 2004, Miami Beach, Florida, USA.

Zhang, H., Wang, D. H. ve Liu, C. L. (2010). Keyword spotting from online Chinese handwritten documents using one-vs-all trained character classifier. *12th International Conference on Frontiers in Handwriting Recognition,* 16-18 November 2010, Kolkata, India.

Zhang, Z., Krawczyk, B., Garcìa, S., Rosales-Pérez, A. ve Herrera, F. (2016). Empowering one-vs-one decomposition with ensemble learning for multi-class imbalanced data. *Knowledge-Based Systems*, *106*, 251-263.

Zhao, C. M., & Luan, J. (2006). Data mining: Going beyond traditional statistics. *New Directions for Institutional Research*, *2006*(131), 7-16.

.

.

ÖZGEÇMİŞ

Kişisel Bilgiler:

Ad- Soyad: Hümeyra DEMİR

Doğum Tarihi: 25.08.1991

Doğum Yeri: Ankara

E-posta: humeyrademir8@gmail.com

Eğitim Bilgileri:

Lise: Dr. Şerafettin Tombuloğlu YDA Lisesi (2004-2008)

Lisans: Gazi Üniversitesi, Fizik Öğretmenliği (2008- 2013)

Yüksek Lisans: Van Yüzüncü Yıl Üniversitesi-Eğitimde Ölçme ve Değerlendirme **(**2016-2018)

İş Deneyimi:

Eylül 2014- Mart 2016: MEB, Fizik Öğretmenliği, Van.

Mart 2016- Ocak 2019: Van Yüzüncü Yıl Üniversitesi, Araştırma Görevlisi, Eğitim Bilimleri Bölümü.

Yabancı Dil:

İngilizce, Eylül 2013, YDS, 83.75.

YÜZÜNCÜ YIL ÜNİVERSİTESİ
Eğitim Bilimler Enstitüsü

LİSANSÜSTÜ TEZ ORİJİNALLİK RAPORU

YÜZÜNCÜ YIL ÜNİVERSİTESİ
Eğitim Bilimler Enstitüsü

08./.01./2019

Tez Başlığı / Konusu

PISA 2015 Veri Setinde OVA ve OVO Strateji Gerçeve sinde Bazı Temel Sınıflandırıcıların Performanslarının Karşılaştırılması

Yukarıda başlığı/konusu belirlenen tez çalışmamın Kapak sayfası, Giriş, Ana bölümler ve Sonuç bölümlerinden oluşan toplam ...71..... sayfalık kısmına ilişkin, 08./01./2019 tarihinde şahsım/tez danışmanım tarafından Turnitin...intihal tespit programından aşağıda belirtilen filtreleme uygulanarak alınmış olan orijinallik raporuna göre, tezimin benzerlik oranı % ..8..... (...Sekiz.....) dır.

Uygulanan Filtreler Aşağıda Verilmiştir:
- Kabul ve onay sayfası hariç,
- Teşekkür hariç,
- İçindekiler hariç,
- Simge ve kısaltmalar hariç,
- Gereç ve yöntemler hariç,
- Kaynakça hariç,
- Alıntılar hariç,
- Tezden çıkan yayınlar hariç,
- 7 kelimeden daha az örtüşme içeren metin kısımları hariç (Limit match size to 7 words)

Yüzüncü Yıl Üniversitesi Lisansüstü Tez Orijinallik Raporu Alınması ve Kullanılmasına İlişkin Yönergeyi İnceledim ve bu yönergede belirtilen azami benzerlik oranlarına göre tez çalışmamın herhangi bir intihal İçemediğini; aksinin tespit edileceği muhtemel durumda doğabilecek her türlü hukuki sorumluluğu kabul ettiğimi ve yukarıda vermiş olduğum bilgilerin doğru olduğunu beyan ederim.

Gereğini bilgilerinize arz ederim.

08../.01../2019.
.....Hümeyra....DEMİR.
Adı, Soyadı, İmza

Adı Soyadı	: Hümeyra DEMİR
Öğrenci No	: 169402010067
Anabilim Dalı	: Eğitim Bilimleri
Programı	: Ölçme ve Değerlendirme
Statüsü	: Y. Lisans ☒ Doktora ☐

DANIŞMAN
Dr. Öğr. Üyesi Güral ERHLİOĞLU
08../.01../2019..

ENSTİTÜ ONAYI
UYGUNDUR
08./.01../2019.
Server ÇAM
Enstitü Sekreteri