Düzce Üniversitesi Bilim ve Teknoloji Dergisi

Araştırma Makalesi

Öğrencilerin Dersteki Niteliklerinin Makine Öğrenmesi Teknikleri Kullanılarak Sınıflandırılması[1]

Ercüment GÜVENÇ [a,*], Murat SAKAL [a], Gürcan ÇETİN [b], Osman ÖZKARACA [b]

[a] *Enformatik Bölümü, Rektörlük, Muğla Sıtkı Koçman Üniversitesi, Muğla, TÜRKİYE*

[b] *Bilişim Sistemleri Mühendisliği Bölümü, Teknoloji Fakültesi, Muğla Sıtkı Koçman Üniversitesi, Muğla, TÜRKİYE*

* *Sorumlu yazarın e-posta adresi: eguvenc@mu.edu.tr*

DOI: 10.29130/dubited.1017202

ÖZ

Öğrencilerin akademik başarılarını tahmin etme ve eksik oldukları alanları giderme anlamında yapılan bu çalışma, Bilişim Sistemleri Mühendisliğine Giriş dersi alan öğrencilere uygulanmıştır. Bu öğrencilerin dönem başı bilgisayar bilgi düzeylerinin, dönem sonunda elde ettikleri başarı notu üzerine etkisi makine öğrenmesi yöntemleri uygulanarak eğitim kalitesinin arttırılması amaçlanmıştır. Çalışmaya katılan öğrencilere ait veriseti eğitim ve test verisi olmak üzere ayrıldığında veri yetersizliğinden dolayı anlamsız sonuçlar ortaya çıkmıştır. Bu nedenle makine öğrenmesi algoritmalarının başarımını arttırmak için "Sentetik Azınlık Örneklem Arttırma Yöntemi (SMOTE)" çalışmada veri çoğaltma tekniği olarak seçilmiştir. Veri çoğaltma işlemi yapıldıktan sonra, veri seti üzerinde uygulanan K-en yakın komşu (KNN), Destek vektör makinesi (DVM), Lojistik Regresyon (LR), Rasgele Orman (RF), Karar ağaçları (DT) ve Naive Bayes makine öğrenmesi yöntemlerine göre en iyi sonucu en yakın komşuluk- KNN algoritması ile oluşturulmuş model vermiştir. Bu model, eğitim setinden bağımsız 300 öğrenciden oluşan test verisinin sınıflandırma işlemini, %97.66 doğrulukla tahmin etmiştir.

Anahtar Kelimeler: *Makine öğrenmesi, Sınıflandırma, Başarı tahmini, En yakın komşuluk algoritması, Sentetik azınlık örneklem arttırma yöntemi.*

Classification of Students' Course Qualifications Using Machine Learning Techniques

ABSTRACT

This study, which aims to predict the academic success of the students and to eliminate the missing areas, was applied to the students who took the Introduction to Information Systems Engineering course. It is aimed to increase the quality of education by applying machine learning methods to the effect of the computer knowledge level of these students at the beginning of the semester on the success grade they get at the end of the semester. When the dataset of the students participating in the study was separated as training and test data, meaningless results emerged due to the lack of data. For this reason, "Synthetic Minority Sampling Method (SMOTE)" was chosen as the data multiplication technique in the study to increase the performance of machine learning algorithms. After the data replication process is done, according to the K-nearest neighbor (KNN), Support vector machine (DVM), Logistic Regression (LR), Random Forest (RF), Decision trees (DT) and Naive Bayes machine learning methods applied on the data set. The model created with the nearest neighbor-KNN algorithm gave the best result. This model predicted the classification process of the test data consisting of 300 students independent of the training set, with an accuracy of 97.66%.

Keywords: *Machine Learning, Classification, Success prediction, K-nearest Neighbour, SMOTE*

1 ICAIAME 2021, konferansında sunulmuştur.

Geliş: 01/11/2021, Düzeltme: 13/12/2021, Kabul: 20/12/2021

I. GİRİŞ

Lisans düzeyindeki öğrencilerin akademik performanslarının makine öğrenmesi teknikleri kullanılarak tahmin edilmesi ve bu tahminler ışığında ders ve konu temelli yönlendirmelerin yapılması ile ilgili çalışmalar son yıllarda oldukça gelişme göstermiştir [1]. Özellikle eş-zamanlı ve eş-zamansız eğitim yöntemlerinde öğrencilerin takibi ve başarı oranlarının seviyelerinin istenilen düzeylere getirilebilmesi açısından bu tip çalışmaların önemi oldukça fazladır. Bu alanda makine öğrenmesi algoritmalarından sınıflandırma ve kümeleme algoritmaları ile çok sayıda çalışma gerçekleştirilmiştir [2].

Özellikle büyük verilerin sınıflandırılması ile ilgili yapılan çalışmalar, görüntü işleme, doğal dil işleme, optimizasyon, risk analizi ve ekonomi alanında tahminler gibi birçok alana yayılmıştır [3]. Bu alanlarda yapılan çalışmaların çoğunda var olan bir veri setindeki veriler kullanılarak makine öğrenmesi algoritmaları yardımıyla sınıflandırma işlemleri gerçekleştirilmektedir. K-Nearest Neighbors (KNN), Support Vector Machine (SVM), Logistic Regression (LR), Random Forest (RF), Decision Tree (DT), Naive Bayes (NB) gibi algoritmalar makine öğrenmesi tabanlı sınıflandırma algoritmalarının en çok tercih edilenleridir. Bu algoritmaların farklı veri setleriyle elde edilmiş sınıflandırma sonuçlarının başarılı olduğunu gösteren çok sayıda örnek literatürde yer almaktadır [4].

Sınıflandırma işlemi veri setinde yer alan verilerin, niteliklerine göre kategorileştirme işlemidir denilebilir. Makine öğrenmesi alanında sınıflandırma işlemi, örnek veri setindeki giriş değerlerinin sınıf etiketlerinin tahmin edilmesi problemine çözüm olarak ortaya çıkan bir modeldir [5]. Sınıflandırma işlemi yapılırken belirtilen algoritmalar, istatistiksel hesaplamalar yardımıyla verileri benzer özelliklere göre veya birbirilerine olan yakınlıklarına göre gruplama işlemi yapmaktadır. Belirtilen algoritmalar ile sınıflandırma işlemi yapılırken veri setindeki veriler sınıflandırma algoritmaları ile eğitilmeden önce veri hazırlama işlemleri yapılır. Bu veri hazırlama işlemi esnasında eldeki verilerin yeterli olup olmadığı belirlenen algoritmaların verdiği sonuçlara bakılarak ortaya çıkartılır [4].

Son yıllarda dengesiz veri kümelerinde, dengesiz veri gruplarını artırmak ya da veri sayısı az olan ve veri toplamanın zor olduğu konularda sentetik olarak veri üretmek amacıyla kullanılan yöntemlerden birisi de SMOTE olmuştur. Literatürde yapılan çalışmalar incelendiğinde, Yavaş ve arkadaşları [6] Covid-19 hastalığının teşhisinde yapay sinir ağı modelinin başarım oranını arttırmak için SMOTE yöntemini önermişlerdir. Çalışmada kullanılan model ile 0.86 olarak elde edilen doğruluk değeri, SMOTE kullanılarak dengelenen veri kümesinde 0.90'a çıkmıştır. Turhan ve arkadaşları [7] tarafından gerçekleştirilen bir diğer çalışmada ise diyabet tanısının sınıflandırılmasında sınıf dengesizliğinin gidermek amacıyla sentetik veri örnekleme metotları başarılı bir şekilde kullanılmıştır. Wang [8] tarafından gerçekleştirilen çalışmada da benzer şekilde kredi risk değerlendirmesinde, dengesiz veri sınıflarından kaynaklanan başarım problemini çözmek için SMOTE yöntemi kullanılmıştır. Çalışma sonucunda SMOTE yönteminin genel başarımı artırdığı belirtilmiştir.

SMOTE yönteminin veri arttırma ve dengesiz veri sınıflarını dengelemedeki başarımı göz önünde bulundurularak, veri sayısı az olan verisetleri üzerinde ne ölçüde başarılı olabileceği öğrenci başarımını ortaya koyacak bir veriseti için araştırılmıştır. Bu amaçla öncelikle örnek bir veri seti hazırlanmıştır. Bu veri setinin hazırlanmasında, "Bilişim Sistemleri Mühendisliğine Giriş" dersini alan öğrencilere, dersi almadan önce öğrencilerin bilgisayar bilgi düzeylerini ortaya koyacak 68 soru sorulmuş ve 71 öğrenciye ait veriler elde edilmiştir. Bu verilere dönem sonu başarı puanları da eklenerek öğrenciler dört ayrı kategoride (çok iyi, iyi, orta, zayıf) sınıflandırılmıştır. Veri setindeki verilerin %20'si test verisi olacak şekilde ayrılarak altı farklı makine öğrenmesi algoritması ile eğitilmiştir. Çalışma sonuçları, SMOTE yöntemi kullanılarak üretilen sentetik verileri içeren veriseti ve orijinal veriseti üzerinde gerçekleştirilen makine öğrenmesi sınıflandırma işlemleri kesinlik, doğruluk ve f1-skor sonuçlarına göre incelenmiştir.

Çalışmanın bölümleri şu şekilde düzenlenmiştir; 2. Bölümde kullanılan veriseti, SMOTE yöntemi ve kullanılan makine öğrenmesi teknikleri detaylandırılmıştır. 3. Bölümde ise veri çoğaltma sürecine ve

yöntemin başarımına yönelik elde edilen sonuçlar paylaşılmıştır. Çalışma sonuçları, Bölüm 4'te verilmiştir.

II. MATERYAL VE METOD

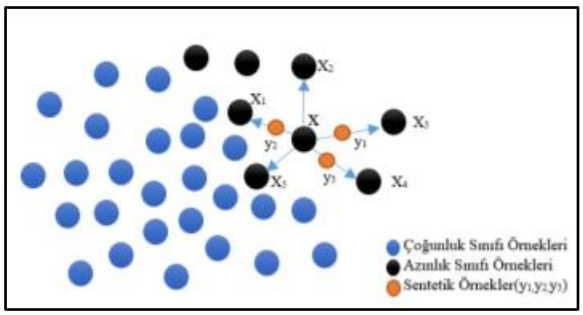
A. VERİ SETİ

Gerçekleştirilen çalışmanın amacı Bilişim Sistemleri Mühendisliğine Giriş dersini seçen öğrencilerin, dönem başlangıcında ders içeriğinde yer alan temel konularla ilgili özelliklerinin belirlenerek, mevcut bilgi düzeyleri ile dönem sonundaki başarı durumunu tahmin etmektir.

Çalışmanın ilk aşaması öğrencilerden veri toplama aşamasıdır. Bu aşamada Bilişim Sistemleri Mühendisliğine Giriş dersinin içeriğinde yer alan haftalık kazanımlar ortaya çıkartılmış ve konu başlıkları dersin veren öğretim görevlisi tarafından seyreltilerek 68 farklı kazanım ortaya çıkarılmıştır. Ortaya çıkarılan bu kazanımların her birisi için, öğrencilerin kendilerini 1-5 puan arasında değerlendirmeleri istenmiştir. 2020-2021 Güz dönemi başında yapılan veri toplama işlemine 71 öğrenci katılmış ve bu şekilde gerçek verilerden oluşan 71x68 boyutundaki ham veri seti ortaya çıkartılmıştır.

B. SMOTE (SENTETİK AZINLIK ÖRNEKLEM ARTTIRMA YÖNTEMİ)

Veri setinde yer alan verilerin yetersiz ve dengesiz olması durumlarında SMOTE (Synthetic Minority Oversampling Technique) olarak adlandırılan yeniden örnekleme yöntemi kullanılmaktadır. 2002 yılında geliştirilen bu algoritma birçok dengesiz veri kümesi problemine uygulanmıştır [9]. SMOTE tekniği, bu gibi durumlarda en yaygın kullanılan ve çoğu zaman en başarılı örnekleme yöntemi olarak literatürde yer almaktadır. Algoritmanın çalışma mantığı Şekil 1'de gösterildiği gibidir [6]. Burada öncelikle, azınlık sınıfına ait her bir gözlemin en yakın k komşusu aranır. Bulunan fark, yeni bir sentetik gözlem oluşturmak için (0,1) aralığında seçilen rastgele bir sayı ile çarpılır. Bu işlem, istenen sayıda sentetik gözlem üretmek için tekrarlanır.



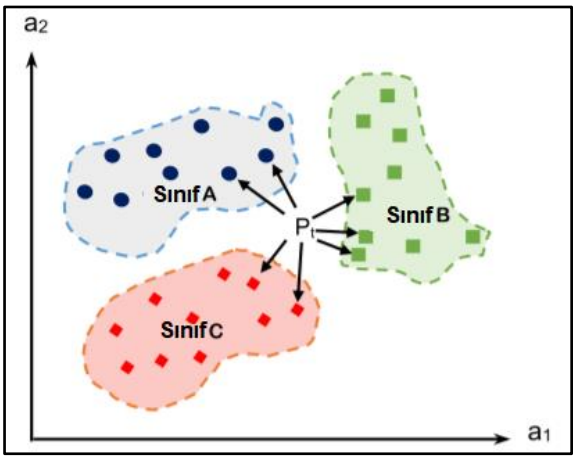
Şekil 1. *SMOTE algoritması [6].*

C. KNN (K-EN YAKIN KOMŞU ALGORİTMASI)

K-NN, sınıflandırılmamış örneğin en yakın komşularının bulunmasına ve benzerliği yüksek sınıflara göre tahminlerde bulunulmasına dayanan bir sınıflandırma yöntemidir [10]. En yakın komşuları bulmak için veri setini tek tek taramak algoritmanın performansını düşürdüğü için tembel öğrenme yöntemi veya vaka tabanlı öğrenme yöntemi olarak adlandırılır [11]. Bu dezavantajdan dolayı, k-NN algoritması, özellikle büyük hacimli verilerde yavaş bir çalışma süresine sahiptir [12].

K-NN algoritması, sınıflandırılmamış bir örneği önceden sınıflandırılmış veri noktalarından herhangi birisine olan uzaklığına göre ilgili sınıfa atama işlemini yapar [13]. Şekil 2'de, bilinmeyen bir veri

noktası olan P, belirlenen k komşuluk değerine göre veri noktasından minimum uzaklığa göre bir sınıfa dahil edilmiştir [14].



Şekil 2. *K-en yakın komşu sınıflandırıcısına bir örnek.*

KNN algoritmasında, veri setindeki sınıfları belli olan kümelerden faydalanılarak, tahminde bulunulacak veri setindeki yeni verilerin, mevcut verilere göre uzaklık değerleri hesaplanır ve ***k*** sayıdaki komşuluklarına bakılır. Komşulukların mesafeleri hesaplanırken sıklıkla kullanılan bir fonksiyon olan Öklid uzaklığı Denklem (1)'de verilmiştir [2].

$$d(x,y) = \sqrt{\sum_{i=1}^{n}(x_i - y_i)^2} \quad (1)$$

Veri setinden yeni bir veriyi sınıflandırmak amaçlı olarak algoritmaya verdiğimizde, eğitilmiş veri setinde yer alan k adet en yakın komşuların sınıf etiketleri alınır. Sonraki adımda, ortaya çıkan sınıf etiketlerinden hangisi çoğunluktaysa işleme alınan veri o kümeye dahil edilir [15].

Ç. SVM (DESTEK VEKTÖR MAKİNESİ ALGORİTMASI)

Destek Vektör Makinesi (SVM), sınıflandırma veya regresyon problemleri için kullanılan bir denetimli makine öğrenmesi algoritmasıdır. Eğer çözülmesi gereken sorun, denetlenen ikili sınıflandırmalardan biri ise SVM(Destek Vektör Makinesi) kullanılabilir [16]. Yani, kategorisi bilinmeyen yeni nesneler, özelliklerine ve daha önceden kategorize edilmiş bilinen örneklerine göre iki ayrı gruba ayırmak istendiğinde SVM algoritması oldukça başarılıdır [17].

D. LR (LOJİSTİK REGRESYON ALGORİTMASI)

Lojistik regresyon, ismine rağmen, regresyon modelinden ziyade bir sınıflandırma modelidir. LR, ikili ve doğrusal sınıflandırma problemleri için basit ve daha verimli bir yöntemdir. Gerçekleştirilmesi çok kolay olan ve lineer olarak ayrılabilir sınıflarla çok iyi performans gösteren bir sınıflandırma modelidir. Endüstride sınıflandırma için yaygın olarak kullanılan bir algoritmadır. Adaline ve perceptron gibi lojistik regresyon modeli, çok sınıflı sınıflandırmaya genelleştirilebilen ikili sınıflandırma için istatistiksel bir yöntemdir [18].

Lojistik regresyonda, bağımlı değişken yalnızca 1 (DOĞRU) veya 0 (YANLIŞ) şeklinde kodlanmış verileri içerir. LR'nin amacı, bağımlı değişken ile bağımsız değişken arasındaki ilişkiyi tanımlamak için en uygun modeli bulmaktır. Lojistik regresyon analizi temelde, bir olayın logaritmik oranını tahmin eder. Matematiksel olarak, lojistik regresyon Denklem (2)'deki gibi tanımlanmış çoklu doğrusal regresyon fonksiyonunu kullanır.

$$\text{logit}_{(p)} = \log\left(\frac{p(y=1)}{1-(p=1)}\right) = \beta_0 + \beta_1 X_1 + \beta_2 X_2 + \beta_3 X_3 + \cdots + \beta_k X_k \quad (2)$$

Bu denklemde p, karakteristik özelliğinin var olma olasılığıdır. Bir diğer önemli değer de odds değeri olarak adlandırılan değerdir. Denklem (3) ile verilen bu değer karakteristik özelliğinin var olma olasılığı (p) değerinin, karakteristik özelliğinin var olmama olasılığı (p-1) değerine bölünmesiyle ortaya çıkan bir değerdir.

$$\text{odds}=\frac{p}{p-1}=\frac{karakteristiğin\ var\ olma\ olasılığı}{karakteristiğin\ var\ olmama\ olasılığı} \quad (3)$$

Burada yer alan odds değeri iki farklı durumun gerçekleşme olasılığını ortaya çıkarır. Gerçekleştirilen çalışmalarda ortaya çıkan bu değerlerin birbirlerine oranı ise iki durumun birbirine göre gerçekleşme olasılıklarını göstermektedir. Örneğin bir hastalığa yakalanma riskinin yaş gruplarına göre olasılıkları hesaplanarak bu hesaplamalar sonucunda hangi yaş aralığındaki kişilerin daha riskli bir yaş grubunda oldukları bulunabilmektedir.

E. RF (RASTGELE ORMAN ALGORİTMASI)

Rastgele ormanlar veya rastgele karar ormanları, eğitim zamanında çok sayıda karar ağacı oluşturarak çalışan, sınıflandırma, regresyon ve diğer görevler için bir topluluk öğrenme yöntemidir. Sınıflandırma görevleri için, rastgele ormanın çıktısı, çoğu ağaç tarafından seçilen sınıftır. Regresyon görevleri için, tek tek ağaçların ortalama veya ortalama tahmini döndürülür [19]. Rastgele orman algoritması genellikle karar ağaçlarından daha iyi performans gösterir, ancak doğruluk değerleri gradyan destekli ağaçlardan daha düşüktür. Ancak, veri setinde yer alan verilerin özellikleri performanslarını etkileyebilir [20].

Rastgele orman algoritmasında yer alan ağaçların her birinin öğrenmesi aşamasında kullanılan genel teknik bootstrap veya bagging yöntemidir. Bu yöntemin işleyişinde, eğitim sonrası, görünmeyen örnekler x' için tahminler, x' üzerindeki tüm bireysel regresyon ağaçlarından gelen tahminlerin ortalaması alınarak aşağıdaki formül ile veya sınıflandırma ağaçları söz konusu olduğunda oy çoğunluğuna bakılarak yapılabilir [21].

$$\hat{f}=\frac{1}{B}\sum_{b=1}^{B} f_b(x^{\iota}) \quad (4)$$

Bu bootstrap prosedürü, bias'ı artırmadan modelin varyansını azalttığı için daha iyi bir performans elde edebilmeyi sağlamaktadır [21].

F. DT (KARAR AĞAÇLARI ALGORİTMASI)

Karar ağaçları algoritması, veri madenciliğinde kullanılan çok sayıda sınıflandırma algoritması olmasına rağmen, uygulamasının ve anlaşılmasının kolay olması sebebiyle literatürde yaygın olarak kullanılan bir sınıflandırma algoritmasıdır [22].

Karar ağacı algoritması, veri setini küçük ve hatta daha küçük parçalara bölerek geliştirilir. Bir karar düğümü bir veya birden fazla dallanma içerebilir. İlk düğüme kök düğüm (root node) denir. Bir karar ağacı hem kategorik hem de sayısal verilerden oluşabilmektedir [23].

Karar ağacı algoritmasının kullanıldığı sınıflandırma problemlerinde, diğer birçok algoritmada olduğu gibi kullanılacak veri seti iki ana parçaya (eğitim-train ve test) ayrılır. Algoritma, eğitim verilerini kullanarak modeli oluşturur. Oluşturulan bu model de test verisi üzerinde uygulanarak modelin problem çözümündeki başarısı hesaplanır. Literatürde en sık kullanılan karar ağacı algoritmaları ID3, C4.5, CHAID, CART algoritmalarıdır.

G. NB (NAİVE BAYES ALGORİTMASI)

İstatistikte Naive Bayes sınıflandırıcıları, özellikler arasında güçlü bağımsızlık varsayımları ile Bayes teoreminin uygulanmasına dayanan basit "olasılıklı sınıflandırıcılar" ailesi olarak tanımlanırlar. En basit Bayes ağ modelleri arasındadırlar [24], fakat çekirdek yoğunluğu tahmini ile birleştiğinde daha yüksek doğruluk seviyelerine ulaşabilirler [25] [26]. Naive Bayes sınıflandırıcıları oldukça ölçeklenebilirdir ve bir öğrenme problemindeki değişkenlerin (features/predictors) sayısında doğrusal bir dizi parametre gerektirir [27].

İstatistik ve bilgisayar bilimi literatüründe, Naive Bayes modelleri, Simple Bayes ve Independent Bayes dahil olmak üzere çeşitli isimler altında bilinir ve bütün bu isimler, sınıflandırıcının karar kuralında Bayes teoreminin kullanımına atıfta bulunur. Ancak Naive Bayes bir Bayes yöntemi değildir [27] [28].

III. VERİ ÇOĞALTMA VE BAŞARI DURUMU TESPİTİ

Veriseti kullanılmadan önce ön işleme adımından geçirilerek veriler eğitime hazır hale getirilmiştir. Ön işleme aşamasında öğrencilerin dönem sonu başarı puanları, ilk adımda oluşturulan veri setindeki öğrenci bilgileriyle eşleştirilerek en son sütuna eklenmiştir. Ayrıca, öğrencilerin başarı puanlarına göre sınıflandırılması işlemi, istatiksel yöntemler kullanılarak ***ÇOK İYİ-İYİ-ORTA-ZAYIF*** şeklinde dört kategoriye ayrılmıştır. Böylece her öğrencinin mevcut niteliklerine göre hangi başarı durumu sınıfında olduğu belirlenmiştir. Öğrencilerden toplanan verilere göre oluşturulan veri setinin ilk hali Şekil 3'te verilmiştir.

	A	C	D	E	F	G	BQ	BR	BS	BU
1	**ÖğrenciNo**	**Nitelik 1**	**Nitelik 2**	**Nitelik 3**	**Nitelik 4**	**Nitelik 5**	**Nitelik 67**	**Nitelik 68**	**BasariNot**	**Durum**
2	201601714	5	3	3	4	3	5	5	31,3	ZAYIF
3	201601021	4	5	3	5	4	5	5	61,9	ÇOK İYİ
4	201601036	4	4	4	4	4	5	5	47,01	ORTA
5	201601065	5	5	4	4	5	5	4	61,08	ÇOK İYİ
6	201601602	3	5	5	5	5	5	5	56,11	İYİ
7	201601007	4	5	5	5	5	5	5	44,53	ORTA
8	201601013	4	4	4	5	3	4	4	55,29	İYİ
9	201601059	5	4	4	5	4	5	4	51,98	İYİ
10	201601025	5	4	4	5	5	5	5	56,11	İYİ
11	201601603	5	5	5	5	5	5	5	47,01	ORTA
12	201601035	5	5	4	5	5	5	5	47,01	ORTA
13	201601061	5	5	5	5	5	5	5	53,63	İYİ
14	201601011	4	5	5	5	5	5	3	54,46	İYİ
15	201601062	3	3	4	2	2	4	3	44,53	ORTA
16	201601070	4	4	3	3	5	5	4	59,42	ÇOK İYİ
17	201601018	5	5	5	5	5	5	5	61,08	ÇOK İYİ
18	201601057	3	3	2	4	1	4	1	40,4	ZAYIF

Şekil 3. *71 Öğrencinin bilgilerine dayalı veriseti.*

Çalışmanın ilk aşamasında, 71 satırlık veri setindeki veriler %20 test ve %80 eğitim verisi olarak ayrılarak, seçilen altı makine öğrenmesi yöntemi ile eğitilmiştir. Bu algoritmalara ait modellerin doğruluk skorları Şekil 4'te verilmiştir.

Algoritma	Doğruluk
Lojistik Regresyon	60.000000
Destek Vektör Makinesi	53.333330
K-en yakın komşu	40.000000
Rastgele Orman	40.000000
Karar Ağacı	33.333333
Naive Bayes	13.333333

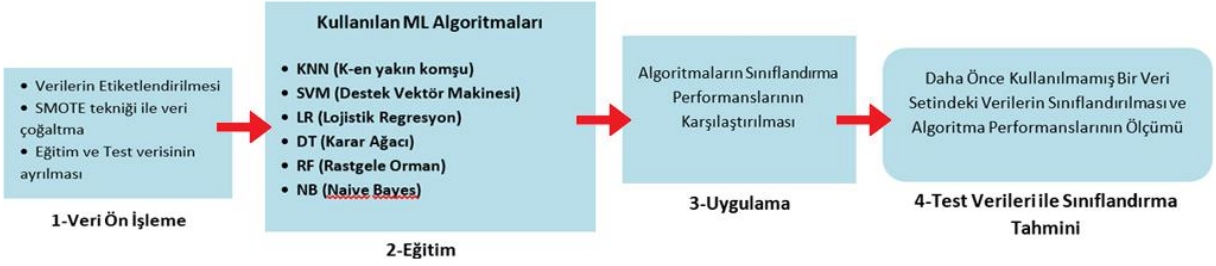
Şekil 4. *Algoritmaların veri setinin ilk hali ile eğitilmesiyle ortaya çıkan doğruluk skor tablosu.*

Şekil 4'de de verildiği gibi ortaya çıkan makine öğrenmesi doğruluk değerleri, eldeki veri seti ile çok başarılı bir sınıflandırmanın yapılamayacağını ortaya koymuştur. Bu nedenle 71 satırlık veri seti içerisindeki veriler kullanılarak, "Sentetik Azınlık Örneklem Arttırma Yöntemi (SMOTE)" ile 640 satırlık eğitim ve 300 satırlık test verisi üretilmiştir. Eğitim veri setindeki verilerin %20 lik bölümü test verisi olarak ayrılmış ve bu veriler daha önce belirtilen altı makine öğrenmesi sınıflandırma algoritması ile eğitilmiştir. Eğitim sonucunda elde edilen modellerin doğruluk tablosu ise Şekil 5'de verilmiştir.

Algoritma	Doğruluk
K-en yakın komşu	96.87500
Destek Vektör Makinesi	96.09375
Lojistik Regresyon	92.96875
Karar Ağacı	90.62500
Rastgele Orman	89.84375
Naive Bayes	59.37500

Şekil 5. *Algoritmaların Eğitim veri setine göre doğruluk skor tablosu.*

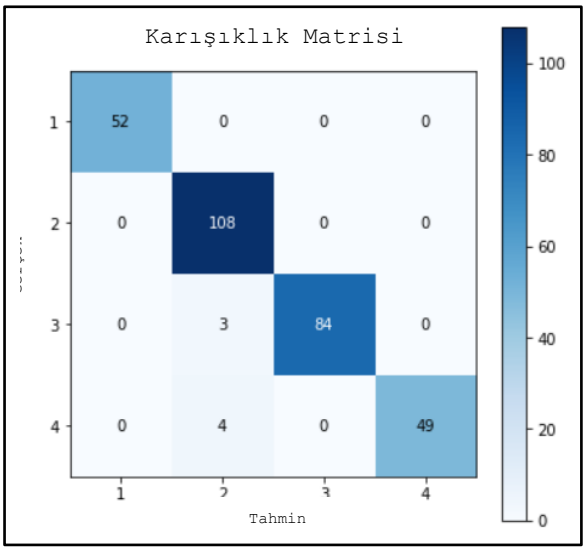
Çalışmada gerçekleştirilen işlem adımları ve bu işlemlere ait bilgiler Şekil 6' da gösterilmiştir. Şekil 6'ya göre birinci adımda veri setindeki verilerin yeterli olup olmadığı test edilmiştir. Daha sonra sentetik veri arttırma tekniği kullanılarak satır sayısı arttırılmış ve eğitim ve test veri seti olarak bölümlere ayrılmıştır. İkinci adımda ise hazırlanan veriler, belirlenen algoritmalar kullanılarak eğitilmiştir. Üçüncü adımda algoritmaların sınıflandırma performansları karşılaştırılarak sıralanmış ve böylece son adımdaki test verileri ile sınıflandırma tahmini yaparken bu tablodan faydalanılmıştır. Şekil 5'deki tabloda da görüldüğü gibi ilk iki sırada yer alan algoritmalar sırasıyla, KNN ve SVM algoritmaları olmuş test verileri ile sınıflandırma tahmini adımında bu iki algoritmanın sınıflandırma performansı incelenmiştir.



Şekil 6. *Yapılan çalışmaya ait işlem adımları.*

A. KNN (K-EN YAKIN KOMŞU ALGORİTMASI)

KNN algoritmasının verimliliği ve performansını belirleyen en önemli faktör komşu sayısının belirlendiği k değeridir. Gerçekleştirilen çalışmada çeşitli k değerlerine göre performans ölçümleri yapılarak en uygun değerin 8 olduğu belirlenmiştir. KNN modelinin optimizasyonu eğitim veri setindeki verilerle sağlandıktan sonra, modelin doğruluğu 300 satırlık test verisi ile ölçülmüştür. KNN algoritmasıyla elde edilen sonuçlara ait karışıklık matrisi Şekil 7'de verilmiştir. Buna göre, KNN algoritması "1 - Çok İyi" ve "2- İyi" sınıflarına ait olan öğrencileri hatasız sınıflandırırken, "3- Orta" sınıfına ait öğrencilerden 3 tanesini "İyi" olarak sınıflandırılmıştır. Ayrıca, "4- Kötü" sınıfına ait öğrencilerden de 4 tanesini "İyi" sınıfına dahil ederken 49 veri doğru sınıflandırılmıştır.



Şekil 7. *KNN Algoritması Karışıklık Matrisi Tablosu.*

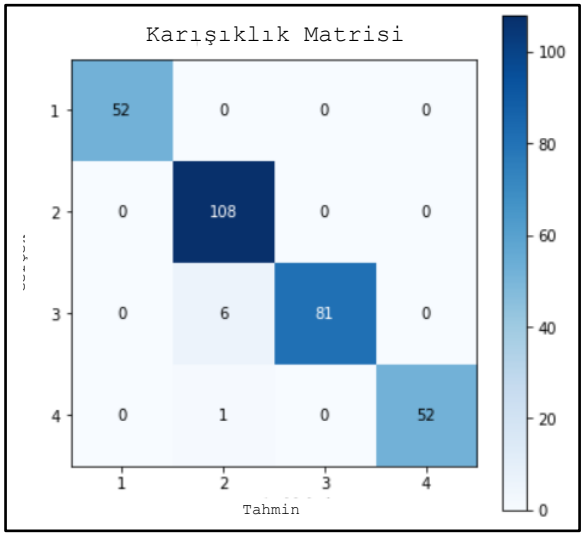
KNN algoritmasının sınıflandırma performansını gösteren değerlendirme kriterleri ve değerleri Şekil 8'deki "Sınıflandırma Raporu" ile verilmiştir. Bu matriste yer alan f1-score puanı, hangi modelin en iyi sınıflandırma performansına sahip olduğuna karar vermek için kullanılmıştır. KNN modeli ile, çalışmadaki test veri setinde yer alan dört farklı sınıfa ait sınıflandırma sonuçlarının f1-score değerlerinden %97.66'lık bir doğruluk değeri elde edilmiştir. Ayrıca matriste görülen kesinlik ve duyarlılık değerlerinin 1'e yakın olması modelin doğruluğunun ne derecede yüksek olduğunu da göstermektedir.

	Kesinlik	Duyarlılık	f1-skor	Destek
1	1.00	1.00	1.00	52
2	0.94	1.00	0.97	108
3	1.00	0.97	0.98	87
4	1.00	0.92	0.96	53
Doğruluk			0.98	300
macro avg	0.98	0.97	0.98	300
weighted avg	0.98	0.98	0.98	300

Şekil 8. *KNN algoritmasına ait sınıflandırma raporu matrisi.*

B. SVM (DESTEK VEKTÖR MAKİNESİ)

Çalışmanın uygulama adımında elde edilen sonuçlar ışığında %96.09'luk doğruluk oranına sahip olan SVM algoritması, test verilerindeki sınıfı belli olmayan satırları Şekil 9'da verilen karışıklık matrisi verilerine göre sınıflandırmıştır. Buna göre SVM algoritması da KNN algoritması gibi "1 - Çok İyi" ve "2 - İyi" sınıflarında yer alan satırları doğru sınıflandırırken, "3 - Orta" sınıfında yer alan 87 kişiden 6 kişiyi "İyi" sınıfına dahil etmiştir. Ayrıca, "4 - Kötü" sınıfında yer alan 53 kişiden 1 tanesi benzer şekilde "İyi" sınıfında tahmin edilmiştir.



Şekil 9. *SVM algoritması karışıklık matrisi tablosu.*

SVM algoritmasının sınıflandırma performansına ait bilgiler Şekil 10'daki Sınıflandırma Raporu ile verilmiştir. KNN algoritması ile elde edilen sonuçlar benzer şekilde SVM algoritması ile de elde edilmiştir. SVM algoritması da %97.66'lık bir doğrulukla sınıflandırma işlemini gerçekleştirmiştir. Bununla birlikte, kesinlik ve duyarlılık değerlerinin KNN de olduğu gibi 1 değerine çok yakın olduğu görülmüştür.

	Kesinlik	Duyarlılık	f1-skor	Destek
1	1.00	1.00	1.00	52
2	0.94	1.00	0.97	108
3	1.00	0.93	0.96	87
4	1.00	0.98	0.99	53
Doğruluk			0.98	300
macro avg	0.98	0.98	0.98	300
weighted avg	0.98	0.98	0.98	300

Şekil 10. *SVM algoritmasına ait sınıflandırma raporu matrisi.*

IV. SONUÇ

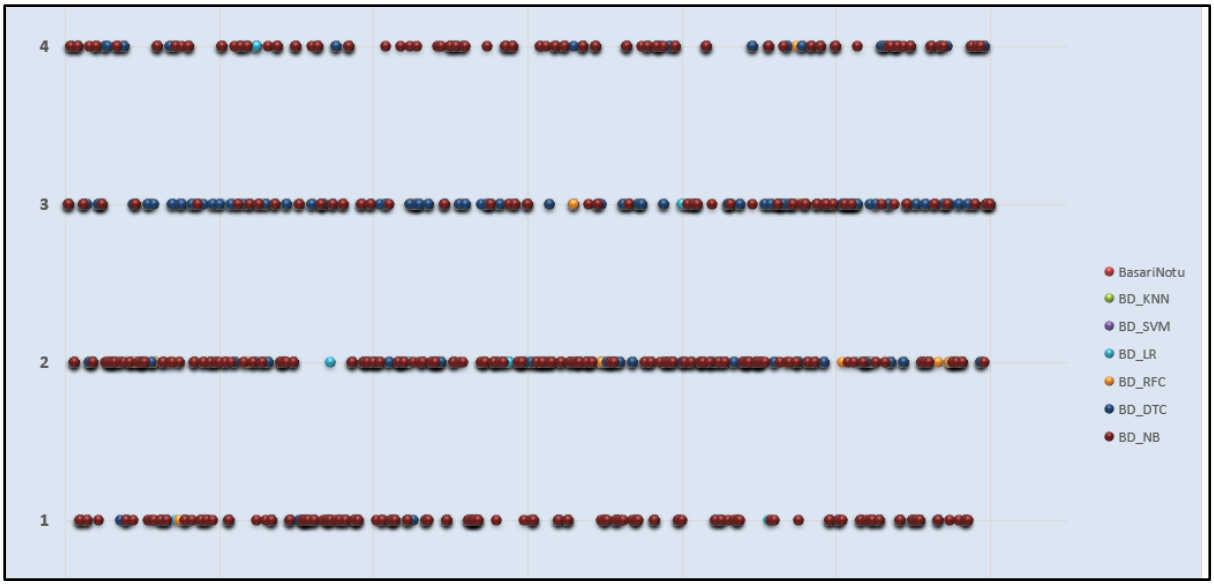
Yapılan çalışmada, eğitim verisinden ayrılan %20'lik veri seti kullanılarak gerçekleştirilen sınıflandırma yöntemlerinin başarısı ortaya konmuştur. Eğitim verisinden ayrılan bu verilerle elde edilen sonuçlar ile 300 satırlık test verisinden elde edilen sonuçlar karşılaştırıldığında en iyi sonuçları veren algoritmaların her iki aşamada KNN ve SVM algoritmaları olduğu görülmüştür. Gerçekleştirilen çalışmada kullanılan algoritmaların test veri seti kullanılarak elde edilmiş sınıflandırma doğruluk skorları Şekil 11'de verilmiştir.

Algoritma	Doğruluk
K-en yakın komşu	97.666667
Destek Vektör Makinesi	97.666667
Lojistik Regresyon	95.333330
Rastgele Orman	94.333330
Karar Ağacı	91.666667
Naive Bayes	68.333333

Şekil 11. *Algoritmaların Test veri setine göre doğruluk skor tablosu.*

KNN ve SVM algoritmalarının, test veri setine uygulanması sonucunda sınıflandırma başarısını gösteren doğruluk değerleri her iki algoritmada da %97.66 olarak ortaya çıkmıştır. KNN ve SVM algoritmalarının test veri setine uygulanmasıyla elde edilen Karışıklık matrisleri incelendiğinde, her iki algoritma da "Çok İyi" ve "İyi" sınıflarını %100 doğrulukla tahmin ederken, KNN algoritması "Orta" sınıfındaki öğrencileri %96.55, SVM algoritması %93.10 doğrulukla tahmin etmiştir. "Kötü" sınıfındaki öğrencileri ise KNN algoritması %92.45, SVM algoritması %98.11 doğrulukla tahmin etmiştir.

Çalışmada kullanılan altı algoritmanın sınıflandırma tahminlerinin gerçekte olması gereken değerlerle karşılaştırıldığı grafik Şekil 12'teki gibidir. Grafikte "Başarı Notu" olarak tanımlanan gösterge, veri setinde yer alan değerlerin gerçekte olması gereken sınıflarını temsil ederken diğer göstergeler ise algoritmaları temsil etmektedir. Şekilden de anlaşıldığı gibi "Başarı Notu" sınıfındaki değerlerden farklı olan sınıflandırma sonuçları ilgili algoritma için belirlenen renk ile gösterilmiştir. Buna göre en farklı sınıflandırmayı Naive Bayes algoritmasının yaptığı grafikte de görülmektedir. Bununla birlikte 3 ncü sınıf olarak tanımlanan "Orta" sınıfında yer alması gereken öğrenciler diğer sınıfların tahminine göre daha düşük oranda tahmin edilmişlerdir.



Şekil 12. *Sınıflandırma tahminleri ve gerçek değerlerin karşılaştırılması.*

Çalışmada kullanılan algoritmaların sınıflandırma tahminleri ile olması gereken sınıflandırma değerleri arasındaki regresyon sonuçları da Tablo 1'de gösterilmiştir. Tabloya göre algoritmaların β katsayıları ve r^2 değerleri incelendiğinde SVM algoritmasının sınıflandırma tahminlerinin, olması gereken tahmin değerlerine en yakın sonuçları üreten doğrusal regresyon modeline sahip olduğu görülmüştür. Ayrıca her bir algoritmanın olasılık değerleri 0.05'in altında olduğundan bütün algoritmalara ait regresyon modelleri anlamlı bulunmuştur.

Tablo 1. *Algoritmalara ait basit doğrusal regresyon analizi sonuçları.*

Algoritma (X)	**Basit Doğrusal Regresyon Modeli**	**R^2**	**Modelin Olasılık Değeri** *(p)*
KNN	Y=0,079+0,983X	0,935	,000
SVM	Y=0,057+0,987X	0,983	,000
LR	Y=0,173+0,950X	0,934	,000
RF	Y=0,138+0,959X	0,944	,000
DT	Y=0,187+0,915X	0,928	,000
NB	Y=1,126+0,591X	0,663	,000

Y: başarı notu, $p<0,05$

Elde edilen bütün sonuçlar karşılaştırıldığında bu gibi veri setlerinde SVM algoritmasının sınıflandırma başarı oranının oldukça yüksek olduğu görülmektedir.

Yapılan çalışmanın ilerleyen aşamalarında Bilişim Sistemleri Mühendisliğine Giriş dersini ilk defa alan öğrencilerin ders ile ilgili nitelikleri, çalışmanın çıkış noktasında olduğu gibi toplanacak, daha sonra yapılan çalışmada ortaya konulan en iyi algoritma ile öğrencilerin niteliklerine göre sınıflandırma tahminleri yapılacaktır. Ortaya çıkan sonuçlar incelenerek periyodik olarak takip edilerek öğrencilere gerekli yönlendirmelerin yapılması planlanmaktadır.

V. KAYNAKLAR

[1] M. Imran, S. Latif, D. Mehmood ve M. S. Shah, "Student academic performance prediction using supervised learning techniques," *International Journal of Emerging Technologies in Learning (iJET)*,vol. 14, pp. 92-105, 2019.

[2] E. Güvenç, G. Çetin ve H. Koçak, "Comparison of KNN and DNN classifiers performance in predicting mobile phone price ranges," *Advances in Artificial Intelligence Research (AAIR),* vol. 1, pp. 19-28, 2021.

[3] A. A. Soofi ve A. Awan, "Classification techniques in machine learning: Applications and issues," *Journal of Basic & Applied Sciences,* no. 13, pp. 459-465, 2017.

[4] B. Abdualgalil ve S. Abraham, "Applications of machine learning algorithms and performance comparison: A Review," *International Conference on Emerging Trends in Information Technology and Engineering,* pp. 1-6, 2020.

[5] J. Brownlee. (2020, Apr 8). *4 Types of Classification Tasks in Machine Learning*. [Online]. Available: https://machinelearningmastery.com/types-of-classification-in-machine-learning/.

[6] M. Yavaş, A. Güran ve M. Uysal, "Covid-19 veri kümesinin SMOTE tabanlı örnekleme yöntemi uygulanarak sınıflandırılması," *Avrupa Bilim ve Teknoloji Dergisi Özel Sayı, ss*. 258-264, 2020.

[7] S. Turhan, Ö. Yüksel, B. S. Yürekli, A. S. Karakülah ve E. Doğu, "Sınıf dengesizliği varlığında hastalık tanısı için kolektif öğrenme yöntemlerinin karşılaştırılması: Diyabet tanısı," *Türkiye Klinikleri Biyoistatistik Dergisi,* c. 12, ss. 16-26, 2020.

[8] L. Wang, "Imbalanced credit risk prediction based on SMOTE and Multi-Kernel FCM improved by particle swarm optimization," *Applied Soft Computing,* vol. 114, pp. 1-14, 2021.

[9] N. V. Chawla, K. W. Bowyer, L. O. Hall ve W. P. Kegelmeyer, "SMOTE: Synthetic minority over-sampling technique," *Journal of Artificial Intelligence Research,* no. 16, pp. 321-357, 2002.

[10] Y.-l. Cai, D. Ji ve D.-f. Cai, *Proceedings of NTCIR-8 Workshop Meeting,* Tokyo, 2010.

[11] P. Cunningham ve S. J. Delany, "k-Nearest neighbour classifiers - A Tutorial," *ACM Computing Surveys, vol*. 6, no. 54, pp. 1-25, 2021.

[12] G. Guo, H. Wang, D. Bell, Y. Bi ve K. Greer, "KNN model-based approach in classification," *On The Move to Meaningful Internet Systems 2003: CoopIS, DOA, and ODBASE. OTM 2003. Lecture Notes in Computer Science,* pp. 986-996, 2003.

[13] T. Cover ve p. Hart, "Nearest neighbor pattern classification," *IEEE Transactions on Information Theory,*vol. 1, no. 13, pp. 21-27, 1967.

[14] D. M. Atallah, M. Badawy, A. El-Sayed ve M. A. Ghoneim, "Predicting kidney transplantation outcome based on hybrid feature selection and KNN classifier," *Multimedia Tools and Asslications,* 2019.

[15] M. Muja ve D. G. Lowe, "Fast approximate nearest neighbors with automatic algorithm configuration," *Proceedings of the Fourth International Conference on Computer Vision Theory and Asslications,* Lisboa, Portugal, pp. 331-340, 2009.

[16] C. Cortes ve V. Vapnik, "Support vector networks," *Machine Leaming,* no. 20, pp. 273-289, 1995.

[17] S. Alay. (2020, 22 Haziran). *AlgoRithm:Destek Vektör Makineleri(Support Vector Machines)(R Kod Örnekli)*. [Çevrimiçi]. Erişim: https://www.datasciencearth.com/algorithmdestek-vektor-makinelerisupport-vector-machinesr-kod-ornekli/.

[18] A. Subasi, Practical machine learning for data analysis using python, *Academic Press*, 2020.

[19] T. K. Ho, "Random decision forests," *Proceedings of the 3rd International Conference on Document Analysis and Recognition, Montreal*, pp. 278-282, 1995.

[20] S. M. Piryonesi ve T. E. El-Diraby, "Role of data analytics in infrastructure asset management: overcoming data size and quality problems," *Journal of Transportation Engineering, Part B: Pavements*, no. 146, pp. 1-17, 2020.

[21] G. James, D. Witten, T. Hastie ve R. Tibshirani, %1 içinde *An Introduction to statistical learning*, Springer, 2013, pp. 316-321.

[22] H. Chauhan ve A. Chauhan, "Implementation of decision tree algorithm c4.5," *International Journal of Scientific and Research Publications*, no. 3, pp. 1-3, 2013.

[23] E. Uzun. (2022) .*Decision Tree (Karar Ağacı): ID3 Algoritması – Classification (Sınıflama)*. [Çevrimiçi]. Erişim: https://erdincuzun.com/makine_ogrenmesi/decision-tree-karar-agaci-id3-algoritmasi-classification-siniflama/.

[24] A. McCallum, *Graphical Models, Lecture2: Bayesian Network Representation*, 2019.

[25] S. M. Piryonesi ve T. E. El-Diraby, "Role of data analytics in ınfrastructure asset management: overcoming data size and quality problems," *Journal of Transportation Engineering, Part B: Pavements*, vol. 2, no. 146, pp. 1-17, 2020.

[26] T. Hastie, R. Tibshirani ve J. H. Friedman, The elements of statistical learning : Data mining, inference, and prediction : With 200 full-color illustrations, New York : Springer, 2001.

[27] R. Stuart ve N. Peter, Artificial intelligence a modern approach, New Jersey: Published by Prentice Hall, 1995.

[28] D. J. Hand ve K. Yu, "Idiot's bayes: Not so stupid after all?," *International Statistical Review*, no., 69, pp. 385-398, 2001.

[29] devhunteryz.wordpress.com. (2018, 20 Eylül). *Rastgele Orman(Random Forest) Algoritması*» [Çevrimiçi]. Erişim: https://devhunteryz.wordpress.com/2018/09/20/rastgele-ormanrandom-forest-algoritmasi/comment-page-1/.