OKU Fen Bilimleri Enstitüsü Dergisi
8(3): 1042-1060, 2025

OKU Journal of The Institute of Science and Technology, 8(3): 1042-1060, 2025

Osmaniye Korkut Ata Üniversitesi Fen Bilimleri Enstitüsü Dergisi

Osmaniye Korkut Ata University Journal of The Institute of Science and Technology

Öğrenci Başarısı Tahmininde Scikit-Learn Kullanımı

Tolga DEMİRHAN[1*]

[1]Trakya Üniversitesi, Tunca MYO, Edirne

[1]https://orcid.org/0000-0001-9840-4457
*Sorumlu yazar: tolgademirhan@trakya.edu.tr

Araştırma Makalesi

Makale Tarihçesi:
Geliş tarihi: 05.09.2024
Kabul tarihi: 03.02.2025
Online Yayınlanma: 16.06.2025

Anahtar Kelimeler:
Scikit-learn
Makine öğrenmesi
Öğrenci başarısı tahmini

ÖZ

Öğrencilerin başarı durumunu öngörmek tüm eğitim kurumları tarafından istenen bir durum olmakla birlikte başarıya etki eden çok faktör olması ve verilerin normalde dengesiz olmasından dolayı oldukça zordur. Bu zorlu süreçte, öğrencinin başarısını tahmin etmek için makine öğrenmesi yaklaşımlarından faydalanılmaktadır. Deneyime dayalı bir yaklaşım olan makine öğrenmesi çalışmalarında; araştırmaya konu olan veri setinin elde edilmesi bu yaklaşımın önemli bir adımıdır. Ancak elde edilen veriler ile işlem ve analizler yapabilmek zordur. Bununla birlikte zaman alıcı ve maliyetli olabilmektedir. Bu sorunun üstesinden gelmek için Scikit-learn gibi ücretsiz python kütüphanelerinden faydalanılabilmektedir. Bu çalışmada; eğitim alanlı veri setlerinde büyük öneme sahip öğrenci performans tahmin işlemlerinin, Scikit-learn kullanılarak analiz edilmesi ve sürecin kullanılan kod betikleri ile birlikte paylaşılarak literatüre ve araştırmacılara katkı sunması amaçlanmıştır. Bu bağlamda çalışmada sırasıyla Naive Bayes (NB), Rastgele Orman (RO), Destek vektör Makinesi (DVM), Karar Ağacı (KA) algoritmaları kullanılmıştır. Geliştirilen tahmin modelinin başarısını ifade etmek için akademik araştırmalarda yaygın olarak kullanılan doğruluk, kesinlik, duyarlılık, F1-puan, kappa ve karışıklık matrisi ölçütleri kullanılmıştır.

Using Scikit-Learn in Student Achievement Prediction

Research Article

Article History:
Received: 05.09.2024
Accepted: 03.02.2025
Published online: 16.06.2025

Keywords:
Scikit-learn
Machine learning
Student achievement prediction

ABSTRACT

Predicting student achievement is desired by all educational institutions, but it is difficult due to the many factors that influence achievement and the normally unstable nature of the data. In this challenging process, machine learning approaches are utilized to predict student achievement. In machine learning studies, which is an experience-based approach, obtaining the data set subject to the research is an important step in this approach. However, it can be difficult, time-consuming and costly to process and analyze the obtained data. To overcome this problem, free python libraries such as Scikit-learn can be utilized. In this study, it is aimed to analyze student performance prediction processes, which are of great importance in educational data sets, using Scikit-learn and to contribute to the literature and researchers by sharing the process with the code scripts used. In this context, Naive Bayes (NB), Random Forest (RF), Support Vector Machine (SVM), Decision Tree (DT) algorithms were used respectively. Accuracy, Precision, Recall, F1-score, kappa and confusion matrix criteria, which are widely used in academic research, were used to express the success of the developed prediction model.

To Cite: Demirhan T. Öğrenci Başarısı Tahmininde Scikit-Learn Kullanımı. Osmaniye Korkut Ata Üniversitesi Fen Bilimleri Enstitüsü Dergisi 2025; 8(3): 1042-1060.

1. Giriş

Öğrencilerin okulda başarısızlık riskinin tahmin edilmesi, birçok faktörün etkisi ve verilerin genellikle dengesiz doğası nedeniyle oldukça karmaşık bir süreçtir. Bu soruna çözüm arayışında, birçok ülke öğrencilerin başarısızlık nedenlerini anlamaya yönelik girişimlerde bulunmaktadır. Bu bağlamda, öğrenci başarısızlığını öngörebilmek ve çözüm yolları geliştirebilmek için makine öğrenimi yaklaşımlarının yaygın biçimde kullanıldığı görülmektedir (Bhawana ve ark., 2014). Makine öğrenimi, veri setleri üzerinde çalışarak verilerden öğrenme ve bu öğrenimi genelleştirme imkânı sunan bir yöntemdir (Namlı ve ark., 2019).

Makine öğreniminde kullanılacak eğitim alanlı veri setleri, sınıf ortamında yüz yüze görüşmeler, anketler veya okul otomasyonlarından elde edilen kayıtlardan oluşturulabilmektedir. (Araque ve ark., 2009). Eğitim alanlı bir veri seti ile yapılacak makine öğrenmesi sürecinde kilit öneme sahip işlem; öğrencinin eğitim kurumlarındaki performansını tahmin edecek öğrenci performans tahmin modellerinin geliştirilmesidir (Khan ve ark., 2019). Veri madenciliği teknikleri ile bu hedefe ulaşılmaya çalışılırken çoğunlukla "sınıflandırma" yöntemi tercih edilmektedir. Sınıflandırma yöntemleri kullanılarak, veriler önceden belirlenmiş gruplara atanır ve bu yöntem yüksek doğruluk oranıyla öğrenci performansını tahmin etmede avantaj sağlar (Guleria ve ark., 2014). Bu tür tahmin modellerinin geliştirilmesi sadece eğitim kurumlarının öğretim kalitesini artırmakla kalmaz, aynı zamanda akademik başarı riski altında olan öğrencilerin erken dönemde tespitine de olanak tanır (Khan ve ark., 2019).

Makine öğrenimi alanında öğrenci performansını anlamaya yönelik yapılan bir dizi çalışma aşağıda özetlenmiştir:

Belachew ve ark. (2017) yaptıkları çalışmada; Wolkite Üniversitesi kayıt ofisinden her bir kayıt için tutulan 34 öznitelikten seçilen 11 tanesi kullanılarak 993 öğrencinin performansını tahmin etmek için bir model geliştirilmiştir. Model oluşturmak için; Neural Nets (MLP), Naive Bayesian, destek vektör makinesi, makine öğrenme algoritmaları kullanılmıştır. Deneyler sonucunda bu algoritmalar içerisinde en başarılı tahmin yapan algoritmanın Naive Bayes olduğu bulunmuştur (Belachew ve ark., 2017).

Soni ve ark. (2018) ise öğrenci performansını, akademik, davranış, müfredat dışı ve yerleştirme olarak adlandırılan dört özellik kategorisi kullanılarak ölçmüştür. Çalışmada farklı üniversitelerden 2.000 öğrenciye uygulanan bir anket kullanılarak veri seti hazırlanmıştır. Karar Ağacı, Naive Bayes ve Destek Vektör Makinesi algoritmalarını deneyen araştırmacılar, en iyi tahmin başarısının Destek Vektör Makinesi algoritmasıyla sağlandığını tespit etmişlerdir (Soni ve ark., 2018).

Khan ve ark. (2019) tarafından yapılan çalışmada ise programlamaya giriş dersinde öğrencilerin yarıyılın erken dönemlerindeki performanslarına bakılarak olası akademik sonuçlar tahmin edilmeye çalışılmıştır. Bu bağlamda çalışmada 11 farklı makine öğrenimi algoritması kullanılmış ve sonuçların karşılaştırması sonunda en doğru tahmini J48 Karar Ağacı algoritmasının sağladığı bulunmuştur (Khan ve ark., 2019).

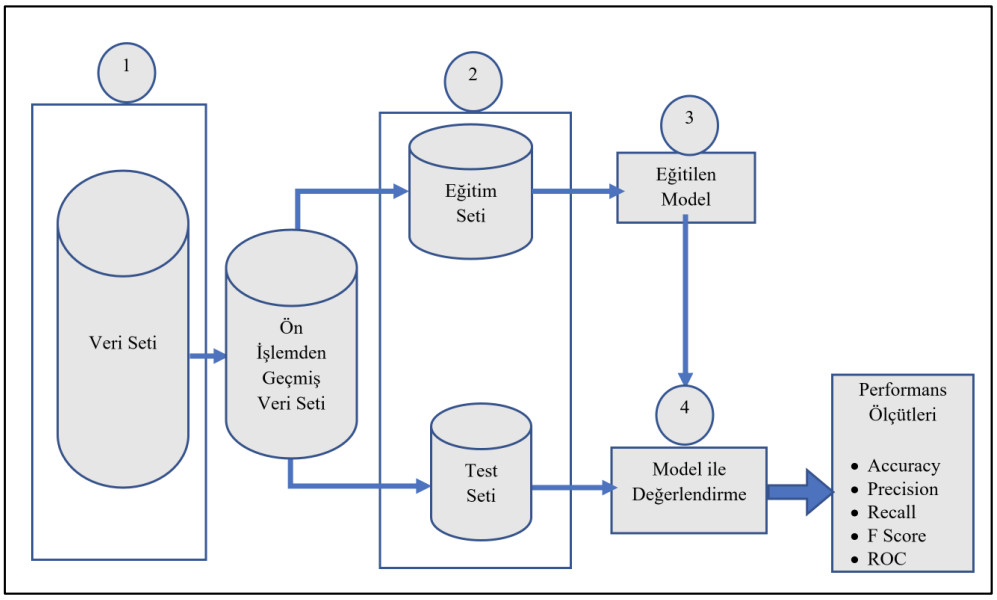
Rony ve ark. (2020) tarafından yapılan çalışmada ise, her biri 25 özniteliğe sahip 522 kayıttan oluşan veri seti üzerinde öğrenci performansını tahmin etmek için Rastgele Orman, K-en yakın komşu (KNN),

Destek vektör sınıflandırma, J48 Karar Ağacı Algoritmalarının kullanıldığı bir çalışma gerçekleştirmiştir. Yapılan deneysel çalışmalar sonucunda Rastgele Orman algoritmasının diğer algoritmalara kıyasla daha başarılı tahmin de bulunduğu tespit edilmiştir (Rahman Rony ve ark., 2020). Günümüzde veriyle çalışan herkesin çok iyi bildiği bir gerçek, bilgi keşfi süreçlerinde makine öğrenmesi ve çeşitli algoritmaların kullanılmasının, özellikle büyük veri setleri üzerinde, artık bir gereklilik haline geldiğidir. Bu bağlamda, makine öğrenmesi araçları önemli bir rol oynamaktadır. Bu tür araçlar ücretli ya da ücretsiz olabilirken, bu çalışmada yıllar içinde daha verimli ve kullanışlı hale gelen, hatta ticari alternatiflerine kıyasla bazı yönlerden daha üstün veya onlarla rekabet edebilir seviyeye ulaşan (Jović ve ark., 2014) Scikit-learn ve Scikit-learn işlemlerine odaklanılmıştır.

Eğitim alanlı veri setleri ile yapılan çalışmalara ait literatür incelendiğinde bu alandaki veri setlerinde Naive Bayes Algoritması, Rastgele Orman (RO), Destek Vektör Makinesi(DVM) ve Karar Ağacı (KA) algoritmalarının en başarılı tahmin sonuçlarını verdiği görülmüştür. Bu kapsamda, çalışmada adı geçen algoritmalar varsayılan ayarlarla Scikit-learn makine öğrenimi aracı üzerinden uygulanmıştır. Veri setinin temizlenmesinden analiz ve başarı değerlendirme aşamalarına kadar tüm süreç boyunca Scikit-learn aracı kullanılmıştır. Veri madenciliğinin her aşamasında Scikit-learn kullanılan ve bu süreçteki kodların paylaşıldığı bir çalışmanın, özellikle bu aracı kullanarak analiz yapmak isteyen araştırmacılar için uygulama odaklı olması sebebiyle değerli bir kaynak olacağı ve literatüre anlamlı bir katkı sağlayacağı düşünülmektedir.

2. Materyal ve Metot

Bu çalışmada, Scikit-learn kütüphanesini kullanmak için eğitim alanlı veri setlerinde en başarılı tahminleri sunan dört algoritma seçilmiş ve başarı tahmini için uygulanmıştır. Bu sürece ait adımlar genel olarak aşağıdaki grafikte paylaşılmıştır. Grafik incelendiğinde birinci adımda veri seti oluşturulup üzerinde önişlemler gerçekleştirilmiştir. İkinci adımda veri seti, %20'si test ve %80'i eğitim olmak üzere iki veri setine ayrılmıştır. Üçüncü adımda eğitim için ayrılan veri seti ile algoritmalar eğitilmiştir. Dördüncü adımda oluşan modeller test veri seti üzerinde çalıştırılmış ve elde edilen tahmin sonuçları, test verisinde gerçek sonuçlar ile karşılaştırılarak performans değerleri ortaya çıkarılmıştır.



Şekil 1. İş akış adımları

Grafikte gösterilen aşamalar, sürecin akışına uygun olarak aşağıda açıklanıp kod betikleri ile verilmiştir.

2.1. Veri Kaynağı

Bu çalışmada kullanılan eğitim alanlı veri seti, 2015 yılında yayınlanmış olan doktora tezi kapsamında gerekli idari izinler alınarak oluşturulup kullanılmış olan veri setidir. Bu nedenle ilgili veri setini kullanmak için çalışma öncesinde etik kurul raporu başvurusu yapılmamıştır. Çalışmaya konu olan veri setindeki kayıtlarda Edirne ili ilköğretim okullarının dördüncü sınıfında okuyan öğrencilere isteğe bağlı olarak uygulanan anket ile konu tarama testi sonuçları bulunmaktadır (Demirhan T., 2015).

Eğitim alanı veri setlerini oluşturan örnekler ve bunlara ait öznitelikler geleneksel olarak sınıf ortamında yüz yüze yapılan görüşmeler veya anketler ile oluşturulabilmektedir (Araque ve ark., 2009).

Çalışmada kullanılan veri seti; her bir öğrenciye ait anket ve ünite testi sonuçlarından oluşmaktadır. Her bir öğrenci, aralarında virgülle ayrılmış 33 öznitelikten oluşan bir kayıt olarak temsil edilmiş ve toplamda 717 öğrencinin bilgisi, her biri ayrı bir satıra denk gelecek şekilde txt uzantılı bir metin dosyasına kaydedilmiştir. Söz konusu metin dosyası, veri kaynağı olarak kullanılmıştır.

2.2. Veri Setini Hazırlama

Veri setini hazırlama, veri analizi sürecinde çok önemli bir adım olup orijinal ham kayıtları kullanılabilir hale getirme sürecidir (Stančin ve ark., 2019). Kayıtlar üzerinde çalışmak için öncelikle verinin çekilip, çalışabilir bir forma dönüştürmek gerekir. Bu kapsamda python'un os ve pandas kütüphaneleri kod betiklerine dahil edilip aşağıdaki işlemler adım adım gerçekleştirilmiştir.

- İlk olarak Os kütüphanesinin import anahtar kelimesi ile python a eklenmesini ve verilerin bulunduğu dosyanın içeriğinin okunarak bir diziye eklenmesi için aşağıdaki kod betiği kullanılmıştır.

```
import os

Icerik=[]

f=open("veriseti.txt",'r',encoding='latin1')
for i in f.readlines():
    Icerik.append(i)
```

Şekil 2. Dosyayı oku

- Belgedeki her bir satır, bir öğrenciye ait özellikleri barındırmaktadır ve satırda yer alan bu özellikler birbirleri ile virgülle ayrılmıştır. Kayıt ve kayıtların her bir özelliğini iki boyutlu bir tabloya yerleştirmek için Split metodu aşağıdaki kod betiği ile kullanılmıştır.

```
kayıtlar=[]

for x in Icerik:
    kayıtlar.append(x.split(","))
```

Şekil 3. Kayda ait özellikleri parçala

- Aşağıdaki kod betiğinde Dataframe de bulunan her bir sütun/özellik için kullanılacak başlıkların tanımlanması yapılmıştır.

```
kayıtlar=[]

for x in Icerik:
    kayıtlar.append(x.split(","))

sutunAdları=["gelir_duzeyi","cinsiyet","yas","nerede_yasiyorsun","kiminle_yasiyorsun","odan_varmi",
             "annen_yasiyormu","annen_calisiyormu","annenin_egitim_durumu","baban_yasiyormu","baban_calisiyormu",
             "babanin_egitim_durumu","kac_kardessiniz","engelli_akraba","odevlerine_kim_yardimci_oluyor",
             "telefon_numarasini_biliyormusun","adresini_biliyormusun","ulasim","okulda_vakit_gecirme",
             "en_coksevdigin_ders","ders_calisma_vakti","odev_yapıyormusun","arakadaslar_ile_vakit_gecirme",
             "yaz_tatili","kis_tatili","kitap_okumayi_seviyormusun","kac_saat_bilgisayar_kullanıyorsun",
             "kac_saat_tv","evcil_hayvan","sira_arakadasindan_memnunumusun","okul_etkinliginde_gorev",
             "uyku_saati","ogretmen_isimlerini_biliyormusun","sontest"]
```

Şekil 4. Başlıkları belirle

- Aşağıdaki kod betiği ile dizi içerisinde bulunan verilerin, satır ve sütundan oluşan iki boyutlu bir DataFrame'e dönüştürülmesi sağlanmıştır.

```
import pandas as pnd
df=pnd.DataFrame(kayıtlar,columns=sutunAdları)
```

Şekil 5. Veriler DataFrame'e dönüştürülüyor

- Oluşturulan DataFrame görünümü aşağıda verilmiştir.

odan_varmi	annen_yasiyormu	annen_calisiyormu	annenin_egitim_durumu
evet	evet	evet	universite
hayir	evet	bos	ortaokul
evet	evet	hayir	universite
evet	evet	evet	bos
evet	evet	evet	universite

Şekil 6. DataFrame Görünümü

Aşağıdaki tablolarda veri setini oluşturulan kayıt ve özellikleri hakkında istatistiği bilgiler paylaşılmıştır.

717 kayıtlı veri setinin sahip olduğu her bir satırdaki 33 öznitelik aşağıdaki tabloda öznitelik adı, açıklamaları ve barındırabileceği değerlerle birlikte verilmiştir.

Tablo 1. Veri setinde kullanılan öznitelikler

Öznitelik Adı	Açıklama	Cevaplar
gelir_duzeyi	Gelir düzeyi bilgisini içerir	'cok_iyi', 'iyi', 'kotu', 'orta'
cinsiyet	Cinsiyet bilgisini içerir	'erkek', 'kiz'
yas	Yaş bilgisini içerir	'10', '11'
nerede_yasiyorsun	Nerede yaşadığı bilgisini içerir	'ailemle evde', 'diğer', 'yurtta'
kiminle_yasiyorsun	Kiminle yaşadığı bilgisini içerir	'ailemle', 'annemle', 'babamla', 'diger'
odan_varmi	Odaya sahip olup olmadığı bilgisini içerir	'evet', 'hayir'
annen_yasiyormu	Anne'nin sağ olup olmadığı bilgisini içerir	'evet', 'hayir'
annen_calisiyormu	Anne'nin çalışıp çalışmadığı bilgisini içerir	'evet', 'hayir'
annenin_egitim_durumu	Anne'nin eğitim bilgisini içerir	'ilkokul', 'lise', 'ortaokul', 'universite'
baban_yasiyormu	Baba'nın sağ olup olmadığı bilgisini içerir	'evet', 'hayir'
baban_calisiyormu	Baba'nın çalışıp çalışmadığı bilgisini içerir	'evet', 'hayir'
babanin_egitim_durumu	Baba'nın eğitim durumu bilgisini içerir	'ilkokul', 'lise', 'ortaokul', 'universite'
kac_kardessiniz	Kardeş sayısı bilgisi içerir	'dort_ve_daha_fazla', 'iki', 'tekim', 'uc'
engelli_akraba	Engelli akraba olup olmadığı bilgisini içerir	'annem', 'babam', 'hicbiri', 'kardesim'
odevlerine_kim_yardimci_oluyor	Ödevlerine yardımcı olan bilgisini içerir	'anne_ve_babam', 'annem', 'babam', 'diger'
telefon_numarasini_biliyormusun	Ev Telefon numarasını bilip bilmediği bilgisini içerir	'evet', 'hayir'
adresini_biliyormusun	Ev adresini bilip bilmediği bilgisini içerir	'evet', 'hayir'
ulasim	Okula nasıl geldiği bilgisini içerir	'araba', 'minibus', 'servis', 'yuruyerek'
okulda_vakit_gecirme	Okulda vakit geçirme bilgisi içerir	'bazen', 'herzaman', 'hicbirzaman', 'siksik'
en_coksevdigin_ders	En çok sevdiği ders bilgisini içerir	'fenveteknoloji', 'hicbiri', 'matematik', 'turkce'
ders_calisma_vakti	Ders çalışma sıklığı bilgisini içerir	'bazen', 'herzaman', 'hicbirzaman', 'siksik'
odev_yapıyormusun	Ödev yapma sıklığı bilgisini içerir	'bazen', 'herzaman', 'hicbirzaman', 'siksik'
arakadaslar_ile_vakit_gecirme	Arkadaşları ile vakit geçirme sıklığı bilgisini içerir	'bazen', 'herzaman', 'hicbirzaman', 'siksik'
yaz_tatili	Yaz tatiline gidilip gidilmediği bilgisini içerir	'bazen', 'herzaman', 'hicbirzaman', 'siksik'
kitap_okumayi_seviyormusun	Kitap okumaya ne kadar vakit ayırdığı bilgisini içerir	'bazen', 'herzaman', 'hicbirzaman', 'siksik'
kis_tatili	Kış tatiline gidilip gidilmediği bilgisini içerir	'bazen', 'herzaman', 'hicbirzaman', 'siksik'
kac_saat_bilgisayar_kullanıyorsun	Bilgisayar karşısında geçirdiği vakit bilgisini içerir	'1_2_saat', '2_4_saat', '4_saaten_fazla', 'kullanmiyorum'
kac_saat_tv	TV karşısında geçirdiği vakit bilgisini içerir	'1_2_saat', '2_4_saat', '4_saaten_fazla', 'kullanmiyorum'
evcil_hayvan	Evcil hayvanı olup olmadığı bilgisini içerir	'evet', 'hayir'
sira_arakadasindan_memnunumusun	Sıra arkadaşından memnun olup olmadığı bilgisi içerir	'bazen memnunum', 'cok memnunum', 'genelde memnunum', 'memnundegilim'
okul_etkinliginde_gorev	Okul etkinliklerinde görev almak sıklığı bilgisi içerir	'bazen', 'herzaman', 'hicbirzaman', 'siksik'
uyku_saati	Akşam saat kaçta uyumaya gittiği bilgisine içerir	'saat10da', 'saat11veyadahagec', 'saat8de', 'saat9da'
ogretmen_isimlerini_biliyormusun	Öğretmen isimlerini öğretneip öğrenmediği bilgisi içerir	'birkacini_biliyorum', 'cogunun_ismini_biliyorum', 'hepsinin_ismini_biliyorum', 'hicbirini_bilmiyorum'
sontest	Testten aldığı sonuç bilgisini içerir	'BASARILI', 'BASARISIZ'

Aşağıdaki tabloda veri setinde yer alan her bir öznitelik için kullanıcılar tarafından en çok tercih edilen değer ve frekans bilgisi ile verilmiştir.

Tablo 2. Veri setinde kullanılan özniteliklerin her biri için en çok tercih edilen değer ve frekansı

Öznitelik Adı	En Çok Seçilen Değer	Değerin Seçim Sıklığı
gelir_duzeyi	orta	278
cinsiyet	kiz	370
yas	10	672
nerede_yasiyorsun	ailemle	701
kiminle_yasiyorsun	ailemle	622
odan_varmi	evet	573
annen_yasiyormu	evet	703
annen_calisiyormu	hayir	385
annenin_egitim_durumu	lise	214
baban_yasiyormu	evet	701
baban_calisiyormu	evet	663
babanin_egitim_durumu	lise	232
kac_kardessiniz	iki	367
engelli_akraba	hicbiri	660
odevlerine_kim_yardimci_oluyor	anne_ve_babam	326
telefon_numarasini_biliyormusun	evet	568
adresini_biliyormusun	evet	630
ulasim	yuruyerek	478
okulda_vakit_gecirme	herzaman	537
en_coksevdigin_ders	matematik	338
ders_calisma_vakti	herzaman	531
odev_yapıyormusun	herzaman	624
arakadaslar_ile_vakit_gecirme	herzaman	587
yaz_tatili	herzaman	382
kis_tatili	bazen	288
kitap_okumayi_seviyormusun	herzaman	413
kac_saat_bilgisayar_kullanıyorsun	1_2_saat	428
kac_saat_tv	1_2_saat	450
evcil_hayvan	hayir	405
sira_arakadasindan_memnunumusun	cokmemnunum	408
okul_etkinliginde_gorev	bazen	357
uyku_saati	saat10da	346
ogretmen_isimlerini_biliyormusun	hepsinin_ismini_biliyorum	359
sontest	BASARILI	592

Yukarıdaki tablo incelendiğinde, sınıflandırma işlemi için kullanılacak sontest isimli öznitelikte 115 adet Başarısız, 592 adet Başarılı değerli kayıt olduğu görülmektedir.

2.3. Veri Önişlem

Önişlem, veri seti üzerinde yapılacak analizin hızını, elde edilen sonuçların güvenilirliğini etkileyen en önemli adımlardan birisidir. Bu kapsamda veri seti üzerinde sırasıyla aşağıdaki adımlar gerçekleştirilmiştir.

- Beş ve daha fazla özelliğin boş olarak geçildiği tespit edilen kayıtlar Dataframe'den çıkarılmıştır. Aşağıda silme işleminin gerçekleştirildiği kod betiği verilmiştir. Silme işleminde sonra veri setindeki kayıt sayısı 716'dan 707'ye inmiştir.

```
dfX.drop(dfX[dfX.isnull().sum(axis=1)>5].index,axis=0,inplace=True)
```

Şekil 7. DataFrame den toplu kayıt silme

- Daha az boş öznitelik barındıran kayıtlar listede bir altında ve bir üstünde bulunan kaydın sahip olduğu öznitelik değeri ile aynı olacak şekilde doldurulmuştur. Bu işlemin yapıldığı kod betiği aşağı verilmiştir.

```
dfX.fillna( method ='ffill', inplace = True)
dfX.fillna( method ='backfill', inplace = True)
```

Şekil 8. Boş özelliklerin doldurulması

- Kayıtlardaki evet/hayır bilgileri barındıran öznitelikler 1/ 0 olacak şekilde sayısallaştırılmıştır. Aşağıda "Evcil hayvanın var mı?" sorusuna verilen Evet/Hayır cevaplarının sayısallaştırılmasını gösteren kod betiği verilmiştir.

```
dfY["e_hayvan"]=dfY["e_hayvan"].apply(lambda x: 0 if x=="hayir" else 1)
```

Şekil 9. Evet/Hayır sayısallaştırıldı

- Kategorik ve aralarında üstünlük kurulan öznitelikler (Eğitim: ilkokul, lise, üniversite) Panda kütüphanesinin factorize metodu kullanılarak 0'dan başlayarak sayısallaştırılmıştır. Kullanılan kod betikleri aşağıda verilmiştir.

```
categoriesEgitimBaba=pnd.Categorical(myDataX['babanin_egitim_durumu'],
                                      categories=["Okuma_yazma_bilmiyor",
                                                  "ilkokul",
                                                  "ortaokul",
                                                  "lise","universite"],
                                      ordered=True)
egitimBaba,unique=pnd.factorize(categoriesEgitimBaba,sort=True)
myDataX['babanin_egitim_durumu']=egitimBaba
```

Şekil 10. Özellikler sayısallaştırıldı

- Kayıtlardaki sınıflandırma öznitelik değeri "Başarısız" değeri için 0, "Başarılı" değeri için 1 olacak şekilde ayarlanmıştır.

```
dfY["sontest"]=dfY["sontest"].apply(lambda x: 0 if x=="BASARISIZ" else 1)
```

Şekil 11. Özellikler sayısallaştırıldı

2.4. Scikit-learn

Scikit-learn; makine öğrenimini akademide veya endüstride herkes için erişilebilir hale getirmeyi amaçlayan açık kaynaklı bir yazılım projesidir. Scikit-learn projesi 2007 yılında David Cournapeau tarafından bir Google Summer of Code projesi olarak başlatılmıştır (Web 4, 2023).

Scikit-learn hem bilim dünyasında geniş çapta benimsenen hem de gelişen bir ekosistem tarafından desteklenen genel amaçlı Python dilinden yararlanmaktadır (Varoquaux ve ark., 2015). Destek vektör makineleri, rastgele ormanlar vb. dahil olmak üzere çeşitli sınıflandırma, gerileme ve kümeleme algoritmalarına sahip olmakla birlikte Python sayısal ve bilimsel kitaplıkları NumPy ve SciPy ile birlikte çalışmak üzere tasarlanmıştır. Scikit-learn sınıfları, yöntemleri ve işlevleri ile kullanılan algoritmaların arka planıyla ilgili belgeler sunmaktadır. Ayrıca, modellerinizi test etmek için kullanabileceğiniz birkaç veri kümesi de sağlamaktadır. Scikit-learn BSD lisansı ile lisanslanmış açık kaynaklı bir pakettir. Python ekosistemindeki çoğu şey gibi, ticari kullanım için bile ücretsizdir (Web 1, 2023).

Scikit-learn'de sınıflandırma algoritması kullanmak için uygulanan iş akışı üç adım içermektedir. İlk adımda, modelin parametreleri belirlenir ve model oluşturulur. İkinci adımda, modelin eğitimi gerçekleştirilir. Son adımda ise öngörülen sınıf değerlerini almak için geliştirilen model, test verileri üzerine uygulanır (Jiangang ve ark.,2019).

```
model=Classifier(hyper-parameters=something)
model.fit(x_train,y_train)
y_predicted=model.predict(x_test)
```

Şekil 12. Sınıflandırma algoritması iş akış kod betiği

Literatür incelendiğinde Scikit-learn kütüphanesinin sahip olduğu esneklik, uyumluluk, verimlilik, performans ve erişilebilirlik gibi özellikleri nedeniyle pek çok alanda araştırmacılar tarafından tercih edilip kullanıldığı görülmektedir.

Ayvaz ve ark. (2017), yaptıkları çalışmada Scikit-learn kütüphanesi kullanarak geliştirdikleri sistem ile öğrencilerin ders boyunca ağırlıklı duygusal durumlarını %97,15 doğrulukla tespit etmişlerdir (Ayvaz ve ark., 2017).

Şenel (2020) yaptığı kayısı iç çekirdeği sınıflandırması başlıklı çalışmasının analiz işlemleri için Scikit-learn kütüphanesinde faydalanmıştır. Gerçekleştirdiği çalışmasında kullandığı algoritmalardan yarısından fazlası ile gerçekleştirdiği analizde yeterli sayıda özniteliğe sahip olduğunda %100 başarı elde ettiğini paylaşmıştır (Şenel, 2020).

Kalaycı (2018) yaptığı çalışmada, bir web sitesinin kimlik hırsızı olup olmadığını tahmin etmek amacıyla Scikit-learn kütüphanesi makine öğrenmesi algoritmalarını kullanmıştır. Elde ettiği analizler sonucunda rastgele orman algoritmasının sınıflandırma başarısının diğerlerinden daha yüksek olduğunu paylaşmıştır (Kalaycı, 2018). Uğuz ve Oral (2019) yaptıkları çalışmada Türkiye'de kurulması planlanan PV santrallerinin üreteceği elektrik gücünün, makine öğrenmesi modelleri ile tahmin edilmesi amacıyla Scikit-learn kütüphanesinden faydalanmıştır (Uğuz ve ark., 2019).

Özen ve ark. (2021), yaptıkları çalışmada Amerika Birleşik Devletleri'ndeki doğrulanmış covid-19 vaka sayılarını kullanarak gelecek günlerdeki vaka tahminlerini yapmak için Scikit-learn kütüphanesinden faydalanmışlardır (Özen, 2021).

2.5. Algoritma Seçimi

Analiz işleminin başlangıcı olan model için kullanılacak algoritmanın seçiminde, analistin sorabileceği en doğal soru, hangi algoritmanın kullanılması gerektiğidir. Bu soru basit bir soru olmasına rağmen ne yazık ki basit bir cevabı yoktur (Caruana ve ark., 2006). Bunun nedeni bu seçimin, olası sonsuz sayıda olasılık arasından hangi makine öğrenme modelinin en iyi performansı gösterdiğini tahmin etme sürecidir (Komer, Bergstra, Eliasmith, 2014).

Model üretmek, geliştirmek için kullanılan algoritmalar faydalı araçlardır. Veri analizi sürecinde kullanılmak üzere geliştirilen pek çok algoritma bulunmaktadır. Bu algoritmaların seçiminde sahip olunan veri seti önem arz etmektedir (Guleria ve ark.,2014). Bu nedenle eğitim alanlı veri setleri ile yapılan çalışmalarda en başarılı sonuçları elde etmiş dört algoritma belirlenerek bu çalışmada kullanılmıştır. Scikit-learn kütüphanesinde bulunan algoritmalar sırasıyla Naive Bayes algoritması, Rastgele Orman (Random Forest) algoritması, Destek Vektör Makinesi (Support Vector Machine) algoritması, Karar Ağacı (Decision Tree) algoritmasıdır. Aşağıda bu algoritmalar ile ilgili kısa açıklamalar ve Python da nasıl kullanılacağı kod betikleri ile verilmiştir.

Naive Bayes Algoritması: Anlaşılabilir ve kolaylıkla uygulanabilir en basit makine öğrenme algoritmalarından birisi olan Naive Bayes (Kaynar ve ark., 2016) sınıflandırıcı; hızlı, doğru ve güvenilir bir algoritmadır. Naive Bayes sınıflandırıcıları, büyük veri kümelerinde yüksek doğruluk ve hıza sahiptir. Bayes sınıflandırıcıları, veri setinde bulunan her bir özniteliğin örnekteki diğer özniteliklerden bağımsız olduğunu varsaymaktadır (Mishra ve ark., 2016).

Örneğin; bir kredi başvurusu sahibinden kefil istenmesinin koşulları kişinin gelirine, önceki kredi ve işlem geçmişine, yaşına ve konumuna bağlı olarak değişmektedir. Bu özellikler birbirine bağımlı olsa bile bağımsız olarak kabul edilmektedir. Bu varsayım, hesaplamayı basitleştirir ve bu yüzden saf olarak kabul edilir. Bu varsayıma sınıf koşullu bağımsızlık denmektedir (Web 2, 2023).

Naive Bayes sınıflandırma algoritmasının Gaussian Naive Bayes, Multinominal Naive Bayes ve Bernoulli Naive Bayes olmak üzere üç türü bulunmaktadır.

- **Gaussian Naive Bayes:** Sonuç özniteliklerinin sürekli değer olması durumunda kullanılır.
- **Multinominal Naive Bayes:** Çok çeşide sahip sonuç özniteliklerinin sınıflandırılmasında tercih edilir.
- **Bernoulli Naive Bayes:** Multinominal Naive Bayes'e benzer şekilde sınıflandırma yapar. Ancak tahminler sadece boolean (ikili) şekildedir (Web 7, 2023).

Çalışmada kullanılan veri setinin özniteliklerinin sahip olduğu değerleri Gaussian Naive Bayes Bayes türüne uygundur. Analiz işlemi Gaussian Naive Bayes ile gerçekleştirilmiştir.

Rastgele Orman (Random Forest) algoritması: Rastgele Orman algoritması (RO), denetimli bir öğrenme algoritmasıdır. Hem sınıflandırma hem de regresyon için kullanılabilir. Aynı zamanda en esnek ve kullanımı kolay algoritmadır. Orman ağaçlardan oluşur. Bir ormanın ne kadar çok ağacı varsa o kadar sağlam olduğu söylenir. Rastgele ormanlar; rastgele seçilen veri örnekleri üzerinde karar ağaçları oluşturur, her ağaçtan tahmin alır ve oylama yoluyla en iyi çözümü seçer. Ayrıca, özelliğin öneminin oldukça iyi bir göstergesini sağlar.

RO, öneri motorları, görüntü sınıflandırma ve özellik seçimi gibi çeşitli uygulamalara sahiptir. Sadık kredi başvuru sahiplerini sınıflandırmak, hileli faaliyetleri belirlemek ve hastalıkları tahmin etmek için kullanılabilir (Web 2, 2023).

Destek Vektör Makinesi (Support Vector Machine) algoritması: Destek vektör makinesi (DVM), algoritması sınıflandırma, regresyon ve aykırı değerlerin tespiti için kullanılan bir dizi denetimli öğrenme yöntemidir. Scikit-learn'deki destek vektör makineleri, girdi olarak hem yoğun (numpy.ndarray ve buna numpy.asarray tarafından dönüştürülebilir) hem de seyrek (herhangi bir scipy.sparse) örnek vektörlerini destekler. Ancak, seyrek veriler için tahminlerde bulunmak üzere bir DVM kullanmak için, bu tür verilere uygun olması gerekir (Web 6,2023).

Karar Ağacıı (Decision Tree): Karar Ağacı; sınıflandırma ve regresyon için kullanılan parametrik olmayan denetimli bir öğrenme yöntemidir. Amaç; veri özelliklerinden çıkarılan basit karar kurallarını öğrenerek bir hedef değişkenin değerini tahmin eden bir model oluşturmaktır. Bir ağaç, parçalı sabit bir yaklaşım olarak görülebilir. Çeşitli karar ağacı algoritmaları vardır. Bunlar ID3, C4.5, C5.0, CART şeklinde sıralanabilir. Scikit-learn, CART algoritmasının optimize edilmiş bir sürümünü kullanır;

CART (Sınıflandırma ve Regresyon Ağaçları), C4.5'e çok benzer. CART, her düğümde en büyük bilgi kazancını sağlayan özelliği ve eşiği kullanarak ikili ağaçlar oluşturur. Weka'daki karşılığı j48 olan algoritma, c4.5 algoritmasıdır (Decision Trees, 2022).

Yukarıda açıklanan algoritmaların Python üzerinde kullanılması için aşağıdaki kod betikleri kullanılır.

```
from sklearn.naive_bayes import GaussianNB
from sklearn.svm import SVC
from sklearn.ensemble import RandomForestClassifier
from sklearn.tree import DecisionTreeClassifier
```

Şekil 13. Algoritmaların projeye dahil edilmesi

Elde edilen veri seti, her bir algoritma için eğitim ve test veri seti olarak kullanılmak üzere iki parçaya ayrılmıştır. Bu işlem için farklı yaklaşımlar kullanılmakla birlikte bu yaklaşımlardan en çok tercih edileni train_test_split() isimli metottur. Aşağıda ilgili metodun nasıl kullanıldığı kod betiği ile verilmiştir. Kod betiğine göre tüm veri setinin %20'si test veri seti, kalan %80 lik kısım ise eğitim veri seti olarak ayrılmıştır.

```
x_train,x_test,y_train,y_test=train_test_split(myDataX,myDataY,test_size = 0.2,random_state=22)
```

Şekil 14. Veri seti eğitim ve test olarak bölmek

Değerlendirme öncesindeki son aşamada ilgili algoritmalar için model oluşturma sürecidir. Bu işlem yukarıda projeye eklenen algoritmalar için aşağıdaki gibidir.

```
modelNaiveBayes=GaussianNB().fit(x_train, y_train)
modelSVM=SVC().fit(x_train, y_train)
modelRF=RandomForestClassifier().fit(x_train, y_train)
modelTree=DecisionTreeClassifier().fit(x_train, y_train)
```

Şekil 15. Algoritma modeli oluşturmak

2.6. Değerlendirme Ölçütleri

Bir veri setinin analizi için oluşturulan modelin, başarısını ifade etmek için kullanılabilecek çok sayıda ölçüt bulunmaktadır. Bu ölçütler, kullanılan veri setinin sahip olduğu örnek sayısı, veri setinin dengeli olup olmaması vb. durumlar için farklılık gösterebilmektedir. Çalışmanın bu bölümünde akademik araştırmalarda yaygın olarak kullanılan ölçütlerden olan Accuracy, Precision, Recall, F1 score, kappa ve confusion matrix ölçütleri hakkında kısa bilgiler verilmiştir. Ayrıca Python da açıklamaları verilen ölçütlerin nasıl kullanılacağı kod betikleri ile gösterilmiştir.

Karmaşıklık Matrisi (Confusion Matrix): Sınıflandırma modellerinin başarımları ile ilgili değerlendirmelerin yapılabilmesi ve ölçütlerin elde edilmesi noktasında karmaşıklık matrisi kullanılmaktadır. İki sınıf değerli/ Classify value/ etiketli (Başarılı/Başarısız) bir karmaşıklık matrisi aşağıdaki gibidir.

Tablo 3. Karmaşıklık Matrisi / TP (True Pozitif)-FN (False Negatif) -FP (False Pozitif)-TN (True Negatif)

		Predict	
		Succesful	Unsuccessful
Actual	Succesful	TP	FN
	Unsuccessful	FP	TN

Doğruluk (Accuracy): Doğruluk, model başarımının ölçülmesinde kullanılan en popüler ve basit yöntemdir (Özhan, 2014). Doğruluk değeri, sınıftan bağımsız olarak tüm test verisi ile ilgili sınıflandırma performansını sunmaktadır (Klement ve ark., 2009). Bir sınıflandırma algoritmasının doğruluğu, Doğruluk değerleri kullanılarak oluşturulan öğrenme eğrileri ile değerlendirilebilir (Russell ve ark., 2010).

$$\textbf{Doğruluk} = \frac{(TP + TN)}{(TP + FP + FN + TN)} \quad (1)$$

Çalışmalarda kullanılan classify value iki sınıflı bir yapıda olduğunda sınıflandırma başarısını sadece doğruluk (Accuracy) değerine bakarak değerlendirmek yanlış olacaktır. Bu nedenle değerlendirme aşamasında F1-Puanı, Kesinlik, Duyarlılık, vb. ölçüt değerlerine de bakılmalıdır (Bulut, 2016).

F1-Puanı (F1-Score): Kesinlik ve duyarlılık değerlerinin harmonik ortalamasıdır. Özellikle eğitim amacıyla kullanılacak verilerin belirlenme aşamasında sınıfların tanı için yeterli olup olmadığını tespit etmek amacıyla kullanılmaktadır. Kabul edilebilir F1 puan değeri genellikle minimum 0,5 olarak alınır (Aydın, 2011).

$$\boldsymbol{F1-Puanı} = \frac{2 * \text{ Kesinlik} * \text{ Duyarlılık}}{\text{Kesinlik} + \text{ Duyarlılık}} \quad (2)$$

Kesinlik (Precision): Kesinlik, sınıfı 1 olarak tahmin edilmiş True Pozitif (TP) örnek sayısının, sınıfı 1 olarak tahmin edilmiş tüm örnek sayısına (TP+FP) oranıdır.

$$\boldsymbol{Kesinlik} = \frac{TP}{TP + FP} \quad (3)$$

Duyarlılık (Recall): Doğru sınıflandırılmış (TP) örnek sayısının gerçek toplam pozitif (TP+FN) örnek sayısına oranıdır.

$$\boldsymbol{Duyarlılık} = \frac{TP}{TP + FN} \quad (4)$$

Kappa Katsayısı: Veri setinde tahmin edilen ve gözlenen sınıflandırmaların karşılaştırmasını nicel olarak ifade etmek için kullanılan bir ölçüttür. KS değeri -1 ve 1 arasında değer almaktadır. -1 değeri tümüyle bir uyumsuzluğu ya da ters yönde bir ilişki olduğunu göstermektedir. 1 ise mükemmel uyumu göstermektedir. KS 0.4 ya da üzeri bir değere sahip ise şansın ötesinde kabul edilebilir bir uyumluluktan söz edilebilir (Aydın, 2011).

ROC Eğrisi (Roc Curve)-AUC (Area Under Curve): Sınıflandırma modelinin performansını değerlendirmek için kullanılan grafiksel bir araçtır. Özellikle ikili sınıflandırma problemlerinde modelin başarısını analiz etmek için kullanılır. ROC eğrisi, True Positive Rate (Gerçek Pozitif Oranı) ile False Positive Rate (Yanlış Pozitif Oranı) arasındaki ilişkiyi gösterir. ROC eğrisi altında kalan alan (AUC) için ideal değer 1'dir. Değer 1'e yaklaştıkça, makine öğrenimi modellerinin sınıfları ayırt etme başarısı artar. AUC'nin 0.5 olması durumunda model tamamen rastgele tahmin yapıyor demektir. 0.5 ile 1 arasındaki değerler, modelin pozitif ve negatif sınıfları belirli bir ölçüde ayırt edebildiğini gösterir. Ancak, 0.5'ten küçük değerler, modelin tahminlerinin ters yönde çalışıyor olabileceğine gösterir. (Web 3,2023).

Değerlendirme ölçütlerini elde etmek için Scikit-learn kütüphanesini kullanalım. Bu aşamada, her bir algoritma için eğitim veri seti kullanılarak eğitilmiş ve oluşturulmuş modeller test veri setlerini kullanarak tahminler üretir. Her bir algoritmanın eğitilen modelinin test veri seti üzerinde tahmin etme işlemini gerçekleştiren kod betiği aşağıda verilmiştir.

```
y_PredictedNaiveBayes=modelNaiveBayes.predict(x_test)
y_PredictedSVM=modelSVM.predict(x_test)
y_PredictedRF=modelRF.predict(x_test)
y_PredictedTree=modelTree.predict(x_test)
```

Şekil 16. Modellere Test veri seti üzerinden tahmin yaptırılması

Naive Bayes algoritması modelinin tahmin başarısı için kullanılacak değerlendirme ölçüt sonuçlarını bulmak için aşağıdaki kod betiği kullanılmıştır.

```
print("Naive Bayes Results")
print("Accuracy Score:{:.2f}".format(metrics.accuracy_score(y_test, y_PredictedNaiveBayes)))
print("Precision Score:{:.2f}".format(metrics.precision_score(y_test, y_PredictedNaiveBayes)))
print("Recall Score:{:.2f}".format(metrics.recall_score(y_test, y_PredictedNaiveBayes)))
print("Kappa Score:{:.2f}".format(metrics.cohen_kappa_score(y_test, y_PredictedNaiveBayes)))
print("F-Measure Score:{:.2f}".format(metrics.f1_score(y_test, y_PredictedNaiveBayes)))
fpr,tpr,threshold=metrics.roc_curve(y_test,y_PredictedNaiveBayes);
roc_auc=metrics.auc(fpr,tpr)
print("ROC - AUC Score:{:.2f}".format(roc_auc))
```

```
Naive Bayes Results
Accuracy Score:0.80
Precision Score:0.90
Recall Score:0.88
Kappa Score:0.06
F-Measure Score:0.89
ROC - AUC Score:0.53
```

```
cmNaiveBayes = confusion_matrix(y_test, y_PredictedNaiveBayes, labels=[0,1])
print("Naive Bayes")
print(cmNaiveBayes)
```

```
Naive Bayes
[[  3  13]
 [ 15 111]]
```

Şekil 17: Değerlendirme Ölçütleri

3. Bulgular

Bu çalışmada, eğitim alanına özgü bir veri seti kullanılarak, scikit-learn kütüphanesi ile öğrenci başarısını tahmin etmeye odaklanılmıştır. Bu amaçla, eğitim alanındaki veri setlerinde en iyi performansı sergileyen dört farklı makine öğrenimi algoritması ilgili veri setinde uygulanmıştır. Süreç boyunca, makine öğrenimi modellerinin oluşturulması, veri ön işleme ve model değerlendirme süreçleri özenle gerçekleştirilmiştir.

Oluşturulan bir modelin başarısını ifade etmek için farklı farklı ölçütlerden faydalanılmaktadır. Çalışmada, model başarı analizi konulu çalışmalarda en çok tercih edilen ölçütlerde olan doğruluk, Kesinlik, Duyarlılık, F1-Puanı, kappa katsayısı, ROC Eğrisi ve Karışıklık matrisi kullanılmıştır. Naive Bayes algoritması, Rastgele Orman algoritması, Destek Vektör Makinesi algoritması, Karar Ağacı algoritması kullanılarak elde edilen değerlendirme ölçüt sonuçları aşağıdaki gibidir.

Tablo 4. Değerlendirme ölçüt sonuçları

Algoritma	Doğruluk	Kesinlik	Duyarlılık	F1-Puanı	Kappa	ROC-AUC
Naive Bayes Algoritması	0,80%	0,90	0,88	0,89	0,06	0,53
Destek Vektör Makinesi algoritması	0,89%	0,89	1	0,94	0	0,50
Rastgele Orman algoritması	0,87%	0,88	0,98	0,93	0,04	0,52
Karar Ağacı	0,80%	0,90	0,87	0,88	0,10	0,56

Naive Bayes algoritması, Rastgele Orman algoritması, Destek Vektör Makinesi algoritması, Karar Ağacı algoritması algoritmalarından elde edilen değerlendirme ölçüt başarı sonuçları ölçüt özelinde şöyledir;

Algoritma başarıları Doğruluk ölçütü bakımından karşılaştırıldığında Destek Vektör Makinesi algoritmasının diğer üç algoritmaya göre daha yüksek bir değere sahip olduğu görülmüştür.

Algoritma başarıları F1-Puan ölçütü bakımından karşılaştırıldığında Destek Vektör Makinesi algoritmasının diğer üç algoritmaya göre daha yüksek bir değere sahip olduğu görülmüştür.
Algoritma başarıları Duyarlılık ölçütü bakımından karşılaştırıldığında Destek Vektör Makinesi algoritmasının diğer üç algoritmaya göre daha yüksek bir değere sahip olduğu görülmüştür.
Algoritma başarıları Precision ölçütü bakımından karşılaştırıldığında Naive Bayes algoritması ile Karar Ağacı algoritmasının diğer iki algoritmaya göre daha yüksek bir değere sahip olduğu görülmüştür.
Algoritma başarıları Kappa ölçütü bakımından karşılaştırıldığında Karar ağacı algoritmasının diğer üç algoritmaya göre daha yüksek bir değere sahip olduğu görülmüştür.
Algoritma başarıları Roc ölçütü bakımından karşılaştırıldığında Karar ağacı algoritmasının diğer üç algoritmaya göre daha yüksek bir değere sahip olduğu görülmüştür.
Oluşturulan modelin başarısını ifade etmek için kullanılan diğer bir metot Karışıklık matrisidir. Aşağıda dört algoritmanın Karışıklık matrisidir sonuçları verilmiştir.

Naive Bayes Algoritması	Destek Vektör Makinesi	Rastgele Orman	Karar Ağaçları
[[3 13] [15 111]]	[[0 16] [0 126]]	[[0 16] [4 122]]	[[3 13] [18 108]]

Şekil 24. Karışıklık matrisi sonuçları

4. Sonuç ve Tartışma

Bu çalışmada, Scikit-learn kütüphanesi kullanılarak her biri 30 özniteliğe sahip toplam 707 kayıttan oluşan eğitim alanlı bir veri seti üzerinde dört farklı denetimli öğrenme sınıflandırma algoritması kullanılarak başarı tahmin analizi gerçekleştirilmiştir. Bu süreçte kullanılan kodlar, resimler ile gösterilip ayrıntılı olarak paylaşılmış, elde edilen sonuçlar aşağıda paylaşılmıştır.
Çalışmada ele alınan Scikit-learn kütüphanesinin açık kaynaklı yapısı ve kapsamlı bir dokümantasyona sahip olması, uygulama geliştirme sürecini önemli ölçüde kolaylaştırmıştır. Orta ölçekli problemlerde başarılı sonuçlar sunduğu belirtilen (Chary, 2020) Scikit-learn kütüphanesi, uygulama geliştirme sürecinde model oluşturma ve test veri setleri üzerindeki tahmin işlemlerinde hızlı yanıt süreleri ile dikkat çekmiştir.
Analizler sonucunda, en yüksek doğru tahmin değerine sahip algoritmanın Destek Vektör Makinesi (DVM) algoritması olduğu belirlenmiştir. DVM algoritması analizinde elde edilen performans ölçütleri ve değerleri, doğruluk: %89, kesinlik: 0,89, duyarlılık: 1, F1-Puanı: 0,94, Kappa katsayısı: 0 ve ROC-AUC: 0,50 olarak bulunmuştur. DVM algoritması model tahmin ölçüt değerleri incelendiğinde;

- Doğruluk ve F1-Puanı değerleri, oluşturulan modelin tahmin/sınıflandırma başarısının yüksek olduğunu göstermektedir.
- Kappa katsayısı ölçütünün 0 olması, modelin tahmin gücünün rastgele bir dağılıma sahip olduğunu göstermektedir (Aydın, 2011).
- Ayrıca ROC-AUC eğrisi ölçütünün 0,5 değerinde bulunması da tahmin gücünün rastgele bir dağılıma sahip olduğunu destekler niteliktedir (Web 3, 2023).

- Kullanılan veri seti dengesizdir: DVM algoritmasının karışıklık matris ölçüt sonuçları değerlendirildiğinde, karışıklık matrisinin sahip olduğu her bir satırındaki değerlerin toplamı her bir sınıfın toplam kayıt sayısını verir. Bu bize model ile yapılan tahmin analizinde kullanılan test veri setinde bulunan 16 kaydın "Başarısız/0", 126 kaydın "Başarılı/1" sınıf değerine sahip olduğunu gösterir. DVM algoritmasının eğitimi için kullanılan veri seti incelendiğinde de veri setindeki 115 kaydın sınıf değeri "Başarısız/0" iken, 592 kaydın sınıf değeri "Başarılı/1" olarak görülür. Veri setindeki bir sınıfa ait kayıt sayısının diğer sınıftaki kayıt sayısından önemli ölçüde az olmazı durumu veri setinin dengesiz bir yapıya sahip olduğunu ifade eder (Kamalov ve ark., 2022).
- Algoritma aşırı uyum (Over Fitting) göstermiştir: Karışıklık matrisi, DVM algoritmasının test veri setindeki performansını şu şekilde özetlemektedir: Sınıf değeri "Başarılı/1" olan 126 kaydın tamamı doğru tahmin edilmiştir, ancak sınıf değeri "Başarısız/0" olan 16 kaydın hiçbiri doğru tahmin edilememiştir. F1-Puanı ölçüt değerinin gösterdiği 0,94 değeri eğitim amacıyla kullanılan kayıtların, test sınıflarının tanısı için yeterli olduğunu göstermiş (Aydın, 2011) bu bağlamda test veri seti üzerindeki başarılı kayıtların tamamını tahmin etmiş ancak başarısız sınıfların hiçbirini tahmin edememiştir, kappa katsayısı ölçütü değerinin 0 olması ve veri setinin dengesiz yapısı ile birlikte değerlendirildiğinde, algoritmanın eğitim verilerine aşırı uyum sağladığı sonucuna varılmıştır. Aşırı uyum durumunda oluşturulan model öğrenme verisi için çok başarılı sınıflama sağlarken, yeni veriler için doğruluk oranı çok düşük olabilmektedir (Demirhan, 2015). Elde edilen sonuç bu duruma bir örnek olarak gösterilebilir. Benzer bir durum, Rastgele Orman algoritmasının performans ölçütleri incelendiğinde de görülmektedir.

Veri setinin dengesiz olma durumu sınıflandırma amaçlı çalışmalarda, sınıflandırmanın başarısı önünde önemli bir engeldir. Bu çalışmada elde edilen analiz sonuçlarında ölçütler arasındaki uyumsuzluğun, veri setindeki sınıf dengesizliğinden kaynaklandığı ifade edilebilir (Alan ve ark., 2020). Dengesiz bir ver setinde sınıflandırma başarısını sadece doğruluk değerine bakarak değerlendirmek yanlış olacaktır (Bulut, F., 2016). Bu nedenle çalışmada, en başarılı algoritmanın DVM olduğunu ifade etmek doğru olmayacaktır. Ayrıca çalışmada kullanılan "Başarısız/0" değerine sahip sınıfların kayıt sayısının "Başarılı/1" sınıfına kıyasla çok yetersiz olması ve analizler de elde edilen sonuçlar seçilmiş dört denetimli algoritmanın da elde ettiği performans sonuçlarının rastlantısal olduğunu bize göstermektedir. Çalışma, Scikit-learn kütüphanesi gibi güçlü bir aracın kullanımıyla dört denetimli öğrenme algoritmasını temel alarak başarı tahmini sürecini detaylı bir şekilde ele almakta ve makine öğrenmesi uygulamalarına dair okuyuculara pratik bilgiler sağlamaktadır. Algoritmaların performansını analiz etme ve yorumlama üzerine yapılan incelemeler, sınıflandırma problemlerinde karşılaşılabilecek zorluklara dair farkındalık kazandırmaktadır. Çalışma ayrıca sınıflandırma performansı değerlendirilirken yalnızca doğruluk gibi tek bir metrik üzerine odaklanmanın yetersiz kalabileceğini belirtmekte ve farklı ölçütlerin bir arada kullanılmasının önemini vurgulamaktadır.

Gelecekteki çalışmalarda, veri setindeki dengesizlik sorununu çözmek amacıyla çeşitli veri ön işleme teknikleri kullanılabilir, kayıt sayısı artırılabilir ve ayrıca kullanılan algoritmaların parametreleri optimize edilerek bu optimizasyonların performansa etkisi detaylı şekilde incelenebilir. Ek olarak, farklı makine öğrenmesi algoritmaları ve derin öğrenme yaklaşımlarını kapsayan daha geniş çaplı bir karşılaştırma yapılması, çalışmanın kapsamını daha da geliştirebilir. Bu tür iyileştirme adımlarının, bir yandan modelin genelleme yeteneğini artırırken diğer yandan uygulamanın gerçek dünya problemlerine daha kolay entegre edilebilir hale gelmesine katkı sağlayacağı değerlendirilmektedir.

Çıkar Çatışması Beyanı

Makale yazarı herhangi bir çıkar çatışması olmadığını beyan eder.

Araştırmacıların Katkı Oranı Beyan Özeti

Yazar makaleye %100 oranda katkı sağlamış olduğunu beyan eder.

Kaynakça

Alan A., Karabatak M. Veri seti- sınıflandırma ilişkisinde performansa etki eden faktörlerin değerlendirilmesi. Fırat Üniversitesi Müh.Mil.Dergisi 2020; 32(2): 531-540

Araque F., Roldan C., Salgu A. Factors influencing university drop out rates. Computers Education 2009; 53(3): 563-574.

Aydın F. Kalp ritim bozukluğu olan hastaların tedavi süreçlerini desteklemek amaçlı makine öğrenmesine dayalı bir sistemin geliştirilmesi, Yüksek Lisans Tezi 2011; 39-41.

Ayvaz U., Gürüler H. Real-time detection of students' emotional states in the classroom. 2017 25th Signal Processing and Communications Applications Conference (SIU) 2017; 1-4.

Belachew E., Gobena F. Student performance prediction model using machine learning approach: The case of Wolkite University. ", International Journal of Advanced Research in Computer Science and Software Engineering 2017; 7(2): 46-50.

Bhawana A., Bharti G. Review on data mining techniques used for educational system. International Journal of Emerging Technology and Advanced Engineering 2014; 4(11): 325-329.

Bulut F. Performance analysis of ensemble methods on imbalanced datasets. Bilişim Teknolojileri Dergisi 2016; 9(2): 153-159.

Caruana R., Niculescu-Mizil A. An empirical comparison of supervised learning algorithms. Proceedings of the 23rd International Conference on Machine learning 2006: 161–168

Decision Trees. scikit-learn.org: https://scikit-learn.org/stable/modules/tree.html adresinden alındı.2022

Deekshith C. Review on advanced machine learning model: Scikit-Learn. Social Science Research Network 2020.

Demirhan T. Makine öğrenmesi algoritmalarının karmaşıklık ve doygunluk analizinin bir veri kümesi üzerinde gerçekleştirilmesi. Yayınlanmış Doktora Tezi, Trakya Üniversitesi Fen Bilimleri Enstitüsü 2015, Edirne.

Guleria P., Sood M. Data mining in education : A review on the knowledge dıscovery perspective. International Journal of Data Mining Knowledge Management Process (IJDKP) 2014; 4(5): 47-60.

Jiangang H., Tin K. Machine learning made easy: A review of scikit-learn package in python programming language. Journal of Educational and Behavioral Statistics 2019; 44(3): 348-361.

Jović A., Brkić K., Bogunović N. An overview of free software tools for general data mining. MIPRO 2014, 26-30 May 2014, Opatija, Croatia 2014: 1112-117.

Kalaycı TE. Kimlik hırsızı web sitelerinin sınıflandırılması için makine öğrenmesi yöntemlerinin karşılaştırılması. Pamukkale Univ Muh Bilim Dergisi 2018; 24(5): 870-878.

Kamalov F., Thabtah F., Leung HH. Feature selection in imbalance Data. Annals of Data Science 2022; 1-15.

Kaynar O., Yıldız M., Görmez Y., Albayrak A. Makine öğrenmesi yöntemleri ile duygu analizi. International Artificial Intelligence and Data Processing Symposium (IDAP'16), 17-18 Eylül, Malatya: IDAP'16 2016; 1-8

Khan I., Sadiri A., Ahmad A., Jabeur N. Tracking student performance in ıntroductory programming by means of machine learning. 4th MEC International Conference on Big Data and Smart City (ICBDSC). Muscat, Oman 2019

Klement W., Wilk S., Michaowski W., Matwin S. Dealing with severely imbalanced data. In: ICEC 2009 Workshop, PAKDD, 2009:1-16

Komer B., Bergstra J., Eliasmith C. Hyperopt-sklearn: Automatic hyperparameter configuration for Scikit-Learn. Proc. of The 13th Python in Science Conf. (SCIPY 2014). 2014.

Mishra T., Kumar, D., Gupta S. Students' employability prediction model through data mining. International Journal of Applied Engineering Research 2016; 11(4): 2279.

Namlı E., Ünlü R., Gül E. Fiyat tahminlemesinde makine öğrenmesi teknikleri ve doğrusal regresyon yöntemlerinin kıyaslanması; Türkiye'de satılan ikinci el araç fiyatlarının tahminlenmesine yönelik bir vaka çalışması. Konya Mühendislik Bilimleri Dergisi 2019; 7(4): 806-821.

Özen NS. COVID-19 Vakalarının makine öğrenmesi algoritmaları ile tahmini: Amerika Birleşik Devletleri Örneği. Avrupa Bilim ve Teknoloji Dergisi 2021; 22: 134-139.

Özhan E. Yapay zeka yöntemleri ile dünya depremlerinin modellene bilirliği. Isıtes2014. Karabük: Isıtes2014.

Rahman Rony A., Tabassum Amity N., Amir M. Student performance prediction using machine learning approach and data mining techniques 2020. Bangladesh: Daffodil International University Dhaka.

Russell S., Norvig P. Artificial ıntelligence a modern approach Third Edition 2010:702-703.

Soni A., Kumar V., Kaur R., Hemavathi D. Predicting student performance using data mining techniques. International Journal of Pure and Applied Mathematics 2018; 119(12): 221-227.

Stančin I., Jović A. An overview and comparison of free Python libraries for data mining and big data analysis. 2019 42nd International Convention on Information and Communication Technology, Electronics and Microelectronics (MIPRO), 2019, Opatija, Croatia. 20-24

Şenel FA. Makine öğrenmesi algoritmaları kullanılarak kayısı iç çekirdeklerinin sınıflandırılması. BEÜ Fen Bilimleri Dergisi 2020; 9(2): 807-815.

Uğuz S., Oral O. PV güç santrallerinden elde edilecek enerjinin makine öğrenmesi metotları kullanılarak tahmin edilmesi. International Journal of Engineering Research and Development 2019; 11(3): 769-779.

Varoquaux G., Buitinck L., Louppe G., Grisel O. Scikit-learn: Machine learning without learning the machinery. GetMobile: Mobile Computing and Communications 2015; 19(1): 29–33.

Web1: https://www.karabayyazilim.com/blog/python/scikit-learn-nedir-2020-02-12-062241. Erişim Tarihi: 09.05.2023.

Web 2: https://www.datacamp.com/community/tutorials/naive-bayes-scikit-learn. Erişim Tarihi, 19.05.2023.

Web 3: https://medium.com/@gulcanogundur/roc-ve-auc-1fefcfc71a14. Erişim Tarihi: 20.05.2023.

Web 4: https://scikit-learn.org/stable/about.html#.Erişim Tarihi: 20.06.2023.

Web 5: https://medium.com/@gulcanogundur/roc-ve-auc-1fefcfc71a14. Erişim Tarihi: 20.05.2023.

Web 6: scikit-learn.org/:https://scikit-learn.org/stable/modules/svm.html. Erişim Tarihi: 20.05.2023.

Web 7: scikit-learn.org/:https://medium.com/kaveai/naive-bayes-ve-uygulamalarıd7d5a56c689b.Erişim Tarihi: 20.05.2023.