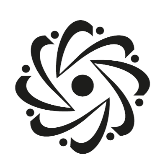
****

**Bursa Teknik Üniversitesi**

**Veri Madenciliğine Giriş**

*Proje Raporu*

**Ad:** Eren

**Soyad:** KÖSE

**Öğrenci Numarası:** 22360859075

**Proje Hakkında**

* **Açıklama:** Öğrencilerin 2 yıllık Matematik, Bilim, İngilizce dersleri için toplanan not verileri, cinsiyetleri, 2017/18 öğretim yılından hesaplanan yaşları, önerilen yıl/sınıfları, okula kabul yılları, önceki okullarında okudukları müfredatları, şu anki müfredatları, önceki okuldaki yıl/sınıfı, okula giriş sınavı matematik notu, okula giriş sınavı bilim notu, okula giriş sınavı ingilizce notu bilgileri bu veri setinde yer almaktadır. Bu veriler kullanılarak öğrencilerin “High”, “Medium”, “Low” şeklinde 3 farklı seviyeye ayrılması amaçlanmıştır. Bu amaç doğrultusunda öğrencilerin not ortalamalarının 85 ve üzeri olması durumu “High” seviye öğrenci, 75 ve 85 aralığı “Medium” seviye öğrenci, 75 altı ise “Low” seviye öğrenci olarak tanımlanmıştır.
* **Kullanılan Veri Seti:** <https://www.sciencedirect.com/science/article/pii/S235234092100192X>
* **Referans Alınan Makale:** <https://link.springer.com/article/10.1007/s43926-022-00023-0>
* **Kullanılan Yöntem:** Random Forest Classifier
* **Github Reposu:** <https://github.com/erennkose/data-mining-project>
* **Tanıtım Videosu:** eren

**Modelim Hakkında Bilgiler**

Projemde geliştirdiğim model öncelikle veri setini içeri aktarır. Bu veri setindeki boşluk, tırnak gibi gereksiz karakterlerin temizliği yapılır. Bu temizlik sayesinde model performansı yükselmektedir. Veri setinden çekilen verilerde kullanılan öznitelikler aşağıdaki tabloda belirtilmiştir.

|  |  |
| --- | --- |
| Öznitelik Numarası | Öznitelik Adı |
| 1 | Gender |
| 2 | Age as of Academic Year 17/18 |
| 3 | Previous Curriculum (17/18)2 |
| 4 | Math20-1 |
| 5 | Science20-1 |
| 6 | English20-1 |
| 7 | Math20-2 |
| 8 | Science20-2 |
| 9 | English20-2 |
| 10 | Math20-3 |
| 11 | Science20-3 |
| 12 | English20-3 |

Veri setinde bulunan eksik değer içeren satırlar dropna() fonksiyonu ile kaldırılmıştır. Modelimin hedef değişkeni ise öğrencilerin kategorik seviyeleridir. Belirtilen öznitelikler kullanılarak öğrenciler belirli seviyelere atanmıştır. Tahmin için kullanılacak bağımsız değişkenler ve bu bağımsız değişkenlere bağlı olacak öğrenci seviyesi belirtilmiştir. Bunların ardından kategorik değişkenler (Gender, Previous Curriculum gibi) one-hot encoding yöntemiyle binary formata çevrilir. Ayrıca öğrencilerin not değerleri yaş gibi değerlere kıyasla değer olarak çok daha baskın olacağından bunu önleyip gerçekçi bir deneyim elde etme amaçlı öğrenci notlarına normalizasyon işlemi uygulandı. Sonrasında ise veri seti % 70 eğitim (train), % 10 doğrulama (validation), %20 test verisi olacak şekilde ayrılır. Stratify parametresi ile sınıfların orantılı dağılması sağlanır. Bu ayrılan veriler kullanılarak 4 farklı model (50, 100, 200 ve 500 ağaçla) eğitilir. Bunlar arasından genel olarak en dengeli sonuçları veren 200 ağaçlık modelin sonuçlarını paylaşacağım. Bu 200 ağaçlık modeli seçme kararını almak için gerekli Random Forest Classifier parametrelerinin olası en iyi sonuç verebilecek değerleri farklı kombinasyonlarla eğitimde denenmiştir. Bu denemeler sonucunda {'criterion': 'gini', 'max\_depth': 15, 'max\_features': 'sqrt', 'min\_samples\_leaf': 1, 'min\_samples\_split': 2, 'n\_estimators': 200} parametre değerlerinin bulunduğu modelin en iyi sonuçları verdiği kararlaştırılmıştır.

metin, yazı tipi, ekran görüntüsü içeren bir resim

Yapay zeka tarafından oluşturulan içerik yanlış olabilir.

Şekil Modellerin Değerleri

**Eğittiğim Modelin Sonuçları**

Projemde yaptığım araştırmalar ve yapay zekalara danışmalarım sonucunda veri setimde Random Forest Classifier yönteminin amacıma daha uygun bir yöntem olduğunu keşfettim ve bu yöntemi seçtim. Bu yöntem sayesinde makalede bulunan değerlerin biraz üstüne çıkmayı başardım. Bunun bir diğer sebebi de gürültü yaratabilecek ve model için mantıksız olan özniteliklerin kullanılmamasıdır. Aşağıda verilen ROC eğrileri sonucunda görüldüğü üzere train verilerinin başarısı kadar test verileri de başarılı şekilde tahmin edilmiş durumda. Bu da overfittingin modelimde söz konusu olmadığını gösteriyor.

Şekil 4’te görüldüğü üzere karmaşıklık matrisinde öğrenciler gayet başarılı şekilde gerekli seviyelere ayrılıyor. Veri seti büyüklüğüne kıyasla ufak seviyede hatalarla karşılaşılıyor. En göze batan hatalı tespit kısmı gerçekte “High” seviyede olan öğrencilerin “Medium” olarak tahmin edilmesidir.

Şekil 5’e baktığımızda da modelimin performans metriklerini görebiliyoruz. Bu performans metriklerinden Accuracy değerleri modelin tahminlerinin genel olarak doğru olduğunu belirtiyor. Bu, tüm sınıflarda modelin başarılı olduğunu ortalama olarak gösteriyor. AUC değerleri ise modelin sınıfları ayırt etme kabiliyetinin çok yüksek olduğunu belirtiyor. ROC eğrisi altında kalan alan çok yüksek olduğu için modelin sınıfları ayrıştırma gücü de o kadar yüksek oluyor. Specifity değeri de AUC ve accuracy gibi yüksek seviyede, yani hatalı sınıflandırma düşük seviyede. Precision, Recall ve F-1 Skoru değerlerinin biraz diğerlerine kıyasla düşük kalmasının sebebi ise veri setimde bulunan verilerin dengesiz dağılımından kaynaklanıyor. High, Medium ve Low seviyede olan öğrencilerin dağılımları eşit sayıda değil. Bundan dolayı Precision, Recall ve F-1 Skoru değerleri, accuracy ve AUC değerlerine kıyasla biraz düşük kalıyor fakat ona rağmen kabul edilebilir durumda.

metin, ekran görüntüsü, çizgi, öykü gelişim çizgisi; kumpas; grafiğini çıkarma içeren bir resim

Yapay zeka tarafından oluşturulan içerik yanlış olabilir.

Şekil ROC Eğrisi (Train)

metin, çizgi, öykü gelişim çizgisi; kumpas; grafiğini çıkarma, ekran görüntüsü içeren bir resim

Yapay zeka tarafından oluşturulan içerik yanlış olabilir.

Şekil ROC Eğrisi (Test)

metin, ekran görüntüsü, diyagram, dikdörtgen içeren bir resim

Yapay zeka tarafından oluşturulan içerik yanlış olabilir.

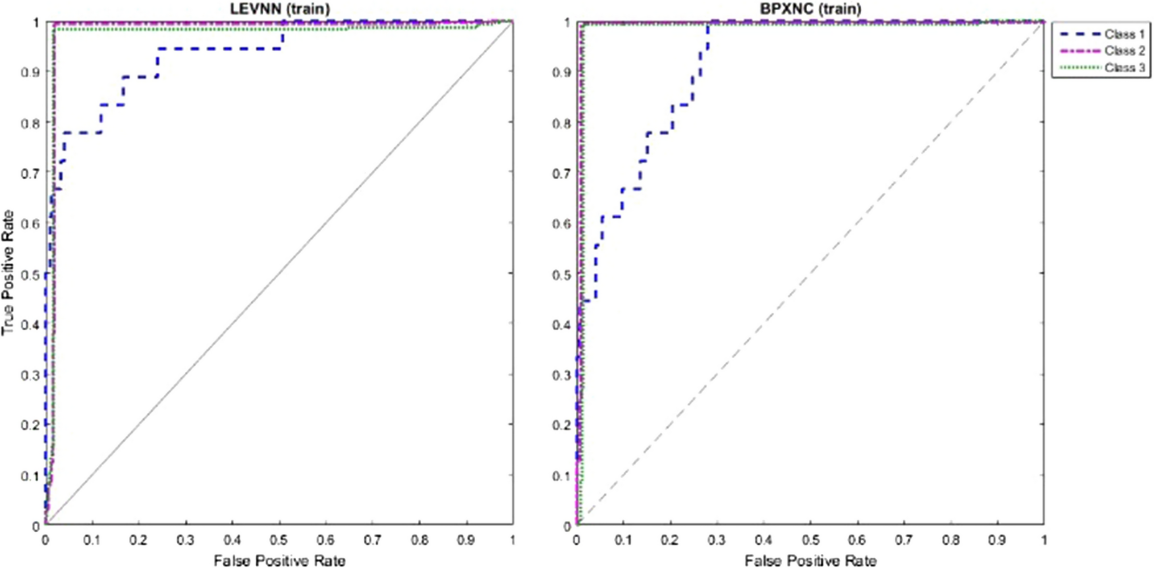
Şekil Karmaşıklık Matrisi

metin, ekran görüntüsü, diyagram, öykü gelişim çizgisi; kumpas; grafiğini çıkarma içeren bir resim

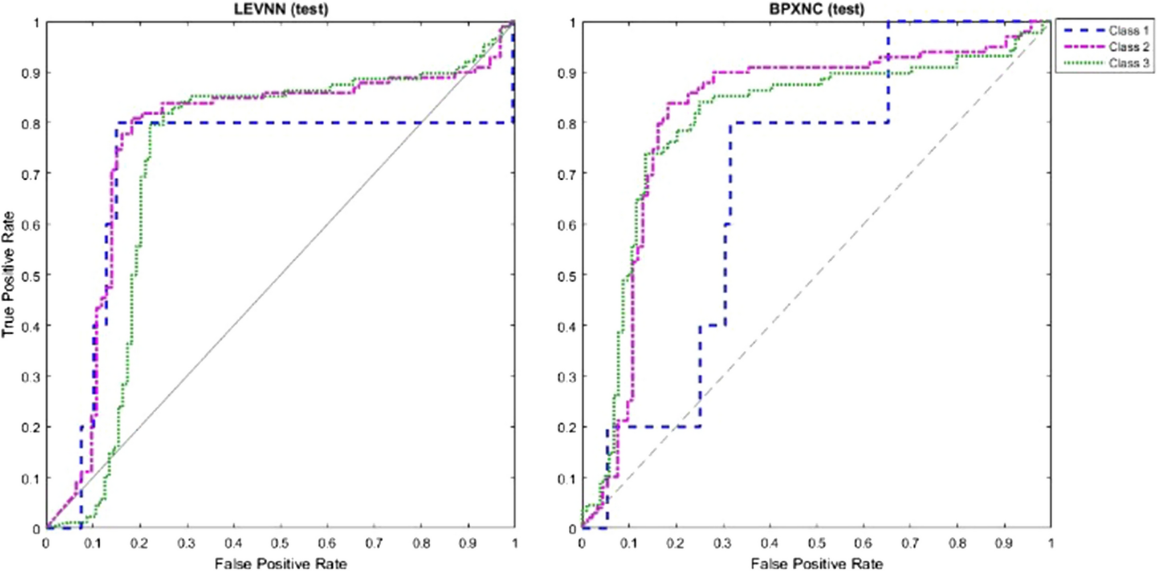
Yapay zeka tarafından oluşturulan içerik yanlış olabilir.

Şekil Performans Metrikleri

**Referans Aldığım Makaledeki Modelin Sonuçları**



Şekil Makale ROC Eğrisi (Train)



Şekil Makale ROC Eğrisi (Test)

**Makale ve Kendi Modelimin Karşılaştırılması**

Makalede alınan değerleri ve kendi modelimin değerleri incelendiğinde;

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Model Adı | Sensitivity (Recall) | Specificity | Precision | F1-Skoru | Accuracy | AUC |
| RFC50 (Makale) | 0.632 | 0.674 | 0.466 | 0.473 | 0.660 | 0.720 |
| LEVNN (Makale) | 0.801 | 0.815 | 0.568 | 0.602 | 0.816 | 0.732 |
| RFC200 (Benim Modelim) | 0.646 | 0.899 | 0.949 | 0.697 | 0,894 | 0.979 |

Bu tablo ele alındığında benim modelim daha iyi ve doğru kararlar alabilen bir model olarak gözüküyor. Bunun birkaç nedenini şöyle sıralayabiliriz;

* Benim yazdığım kodda en iyi model GridSearch ile belirleniyor, parametreler tek tek deneniyor; makaledeki modelde ise sınırlı ayarlama var, çoğunlukla sabit değerlerle çalışılmış.
* Benim modelim için veri setinden çekilen verilerdeki eksik veriler dropna() fonksiyonu ile temizlendi. Makalede bu işlemler manuel yapılmış. Bu da verilerde daha fazla hata ve gürültü olma olasılığını arttırıyor.