VERİ ANALİZİ DERSİ

DÖNEM PROJESİ

Öğrenci performans analizi

Güven Karataş

24430070076

Yönetim Bilişim Sistemleri

Mersin Üniversitesi

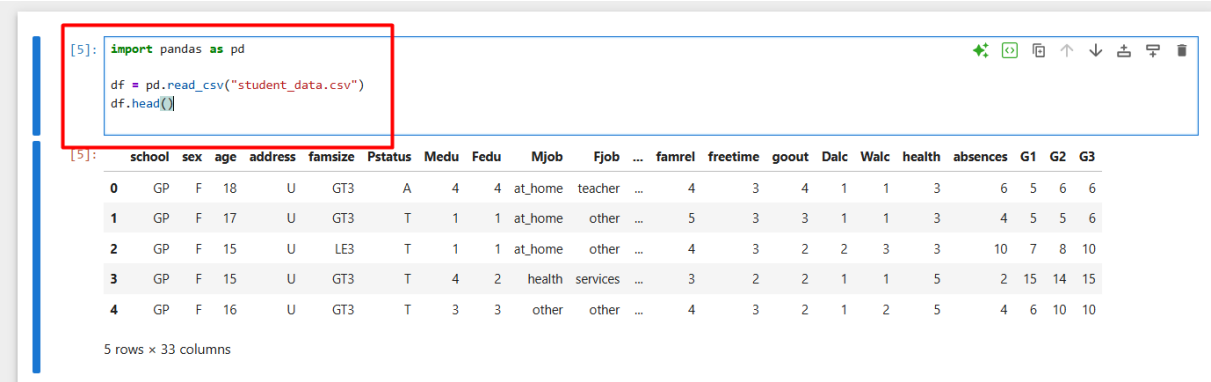
1.Giriş

Bu çalışmamda, öğrencilerin demo grafik, ailesel ve akademik özelliklerini kullanarak dönem sonu başarı notlarının tahmin edebilmeyi amaçladım. Veri seti, öğrencilerin önceki dönem notları, çalışma alışkanlıkları ve sosyal durumları gibi çeşitli bağımsız değişkenler içermektedir.

Çalışmada hedef değişken olarak öğrencilerin final notunu temsil eden **G3** değişkenini seçtim. G3 değişkeni, öğrencinin ders sonunda elde ettiği genel başarıyı yansıttığı için çalışmanın temel çıktısı olarak belirlenmiştir. Bu kapsamda problem, sürekli bir değişkenin tahmin edilmesini amaçladığından bir regresyon problemi olarak ele aldım.

Çalışma yapacağım Veri setini Kaggle'dan temin ettim ve öğrenci performansı ile ilgili sayısal ve kategorik özellikler içermesine dikkat ettim.

Veri setim, proje dizini altında yer alan *data* klasörü içerisinde saklanmış ve analizler bu yapı üzerinden gerçekleştirilmiştir.



```
[5]: import pandas as pd

     df = pd.read_csv("student_data.csv")
     df.head()
```

[5]:

	school	sex	age	address	famsize	Pstatus	Medu	Fedu	Mjob	Fjob	...	famrel	freetime	goout	Dalc	Walc	health	absences	G1	G2	G3
0	GP	F	18	U	GT3	A	4	4	at_home	teacher	...	4	3	4	1	1	3	6	5	6	6
1	GP	F	17	U	GT3	T	1	1	at_home	other	...	5	3	3	1	1	3	4	5	5	6
2	GP	F	15	U	LE3	T	1	1	at_home	other	...	4	3	2	2	3	3	10	7	8	10
3	GP	F	15	U	GT3	T	4	2	health	services	...	3	2	2	1	1	5	2	15	14	15
4	GP	F	16	U	GT3	T	3	3	other	other	...	4	3	2	1	2	5	4	6	10	10

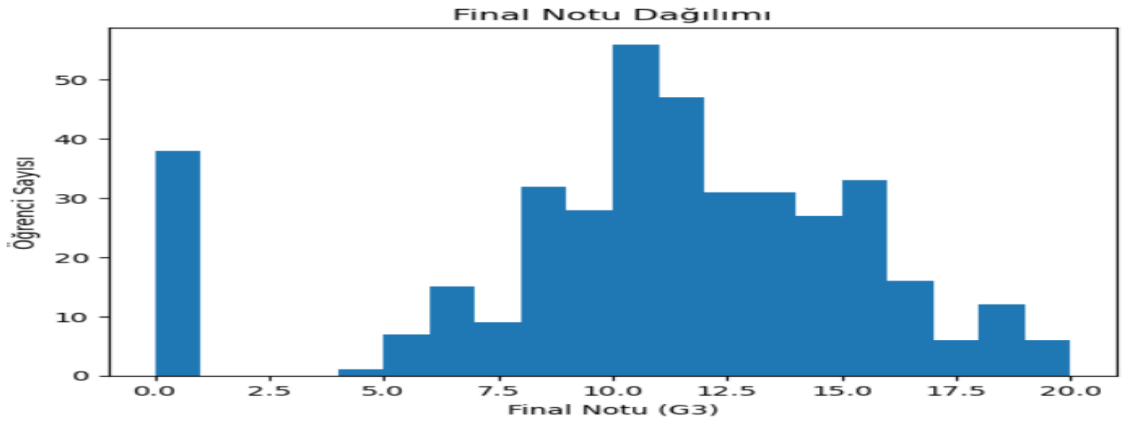
5 rows × 33 columns

Buradaki kodda Python'un *pandas* kütüphanesi kullanılarak student_data.csv adlı veri seti okunmuştur. Ardından head() fonksiyonu ile veri setinin ilk 5 satırı görüntülenerek değişkenlerin yapısı ve örnek gözlemler incelenmiştir.

2. Veri Seti ve Problem Tanımı

Veriler nasıl dağıtılıyor?

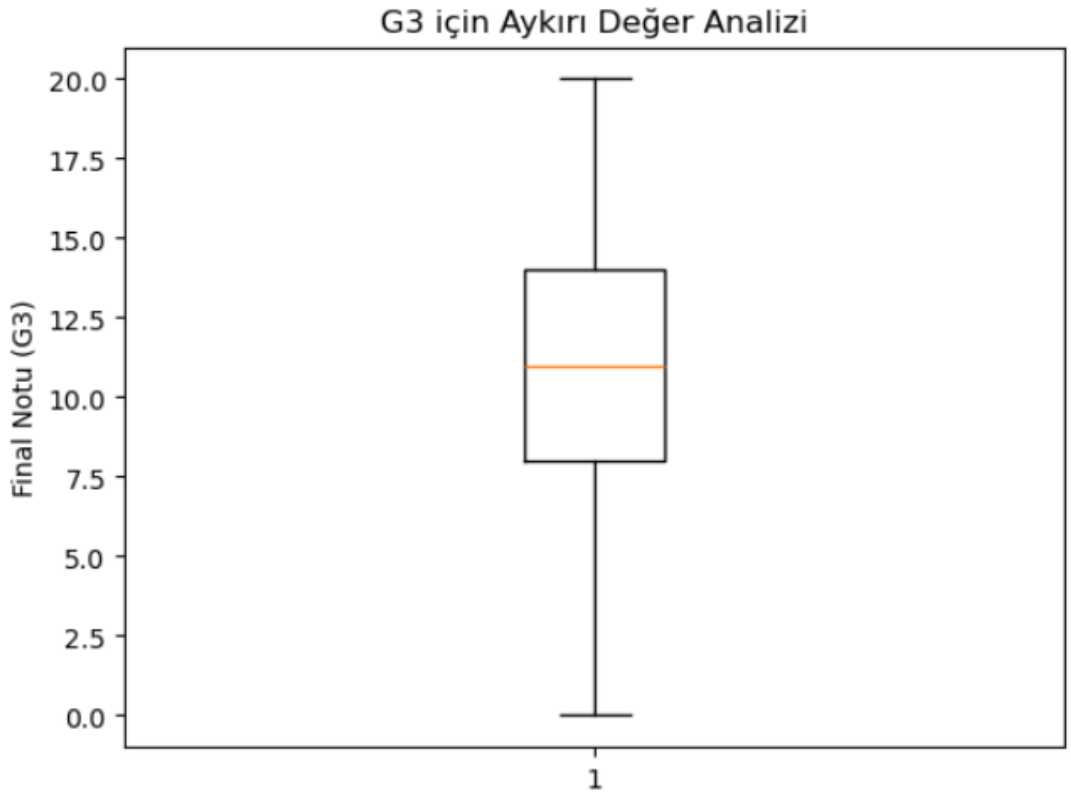
```
[6]: import matplotlib.pyplot as plt

plt.hist(df["G3"], bins=20)
plt.xlabel("Final Notu (G3)")
plt.ylabel("Öğrenci Sayısı")
plt.title("Final Notu Dağılımı")
plt.show()
```



Burada hedef değişkenimiz olan G3 değişkeninin dağılımını inceledim. Elde edilen histogram grafiği, öğrencilerin büyük bir kısmının orta düzey notlarda yoğunlaştığını, çok düşük ve çok yüksek notların ise daha az gözlendiğini gördüm. Bu durum, veri setinin dengesiz bir dağılıma sahip olmadığını gösterdi.

Genel dağılımdan ciddi şekilde sapan gözlemler

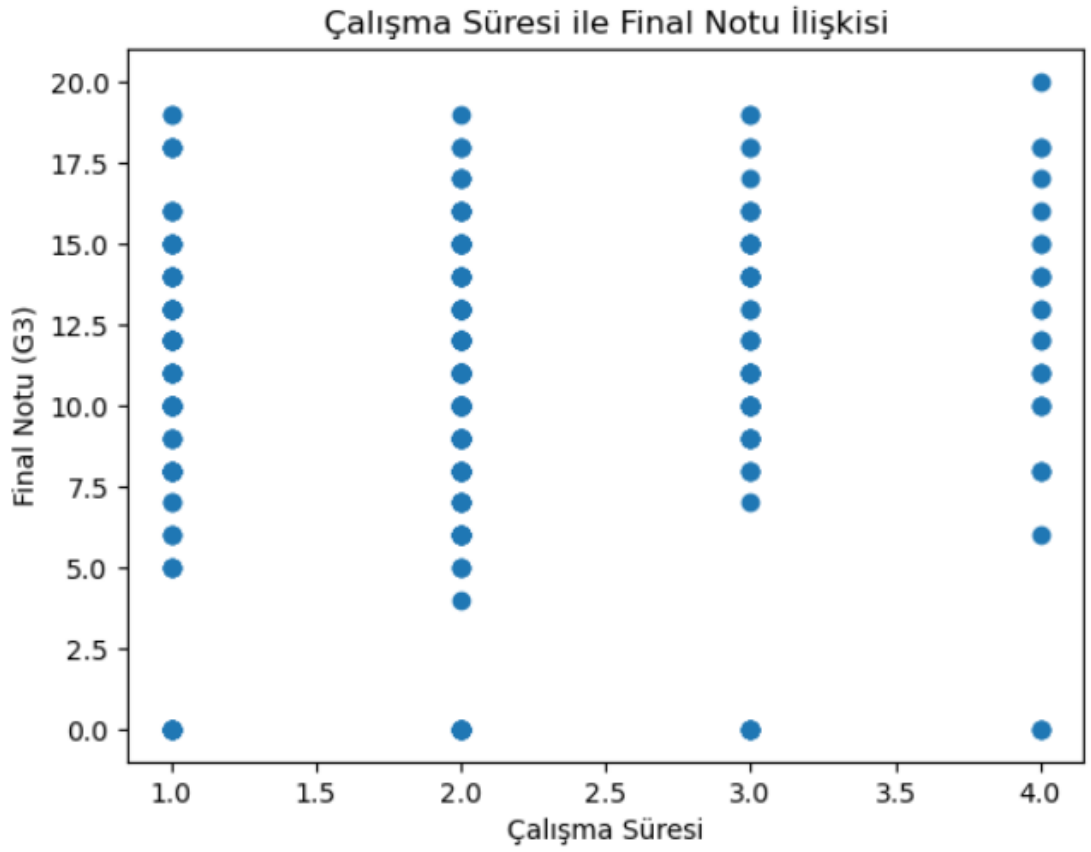
```
[7]: plt.boxplot(df["G3"])
     plt.ylabel("Final Notu (G3)")
     plt.title("G3 için Aykırı Değer Analizi")
     plt.show()
```



Final notu değişkeni için aykırı değer analizi yaptım. Boxplot grafiği incelediğimde, veri setinde çok sınırlı sayıda aykırı gözlem bulunduğu ve bu değerlerin gerçek hayatta karşılaşılabilecek durumlar olduğu değerlendirdim. Bu nedenle aykırı değerler verisetinden çıkarılmamıştır.

Çalışma süresi ile Final notu arasındaki ilişki

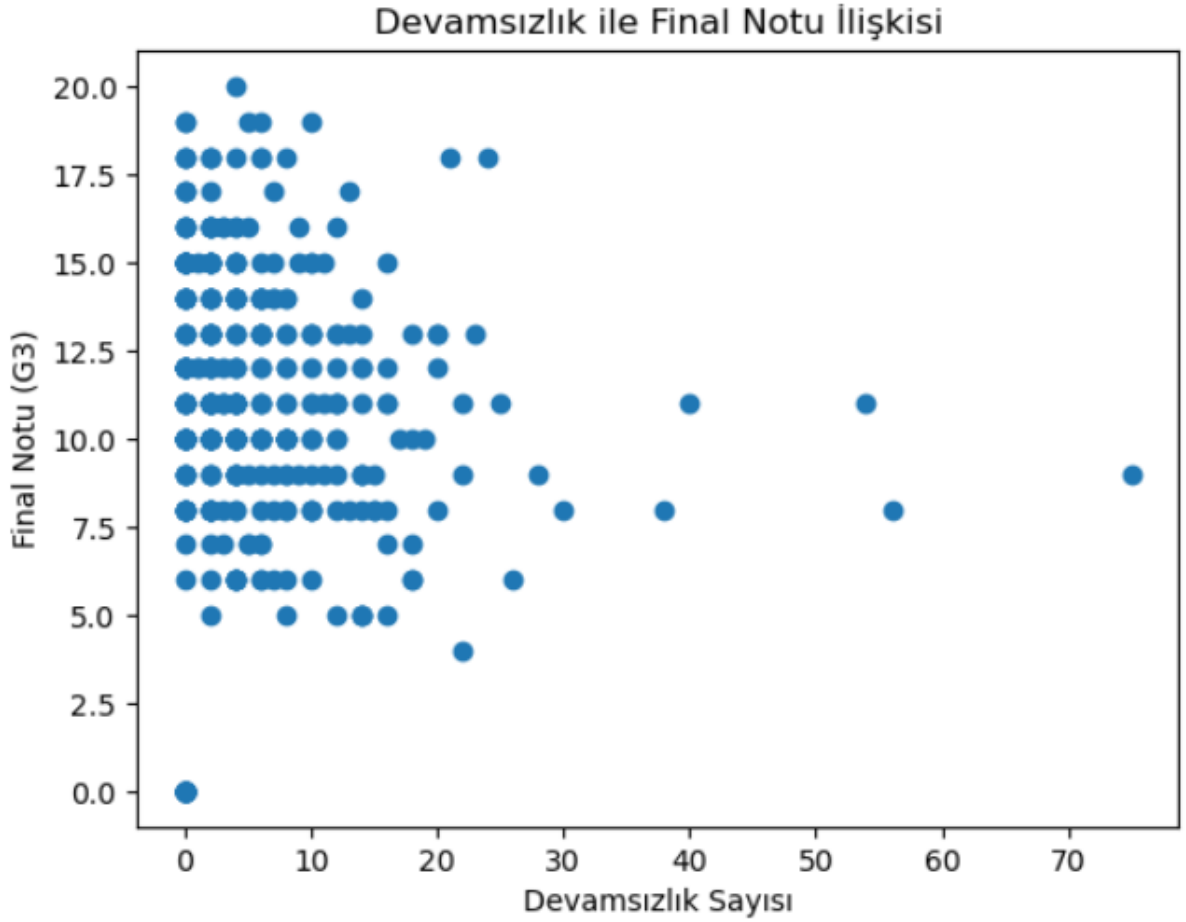
[8]:
```
plt.scatter(df["studytime"], df["G3"])
plt.xlabel("Çalışma Süresi")
plt.ylabel("Final Notu (G3)")
plt.title("Çalışma Süresi ile Final Notu İlişkisi")
plt.show()
```



Raporda her bir öğrenciyi bir nokta olarak görüntülemekteyiz. Burada çalışma süresi arttıkça final notunun yükselme eğiliminde olduğu görüntülenmektedir.
Bu durum, çalışma süresinin öğrencinin akademik başarısı üzerinde etkili bir değişken olduğunu düşündürmektedir.

Devamsızlık ile Final Notu İlişkisi

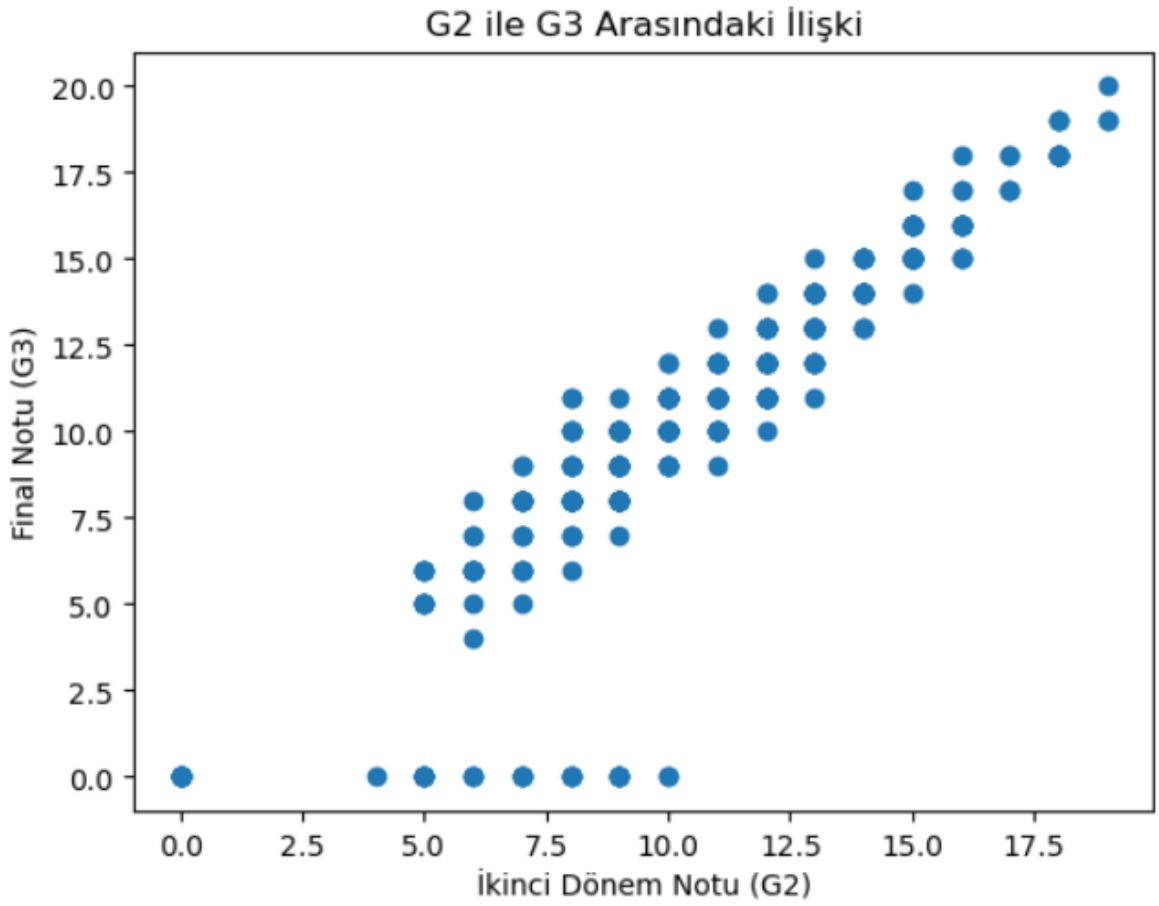
```
plt.scatter(df["absences"], df["G3"])
plt.xlabel("Devamsızlık Sayısı")
plt.ylabel("Final Notu (G3)")
plt.title("Devamsızlık ile Final Notu İlişkisi")
plt.show()
```



Buradaki raporda devamsızlık sayısı ile final notu arasındaki ilişkiyi inceledim. Devamsızlık sayısı arttıkça final notunun genel olarak düşme eğiliminde olduğunu gördüm. Buradaki analiz ile derslere katılımın akademik başarı üzerinde olumlu etkisi olabileceğini ıspatlamaktadır.

G2 ile G3 Arasındaki İlişki

```
plt.scatter(df["G2"], df["G3"])
plt.xlabel("İkinci Dönem Notu (G2)")
plt.ylabel("Final Notu (G3)")
plt.title("G2 ile G3 Arasındaki İlişki")
plt.show()
```



Buradaki raporda Önceki dönem notu yani G2 ile ile final notu arasındaki ilişkiyi görmekteyim. İlişki pozitif çıkması modelin öğrenebileceği güçlü bir sinyal bulduğu anlamına gelir. Yani öğrencinin önceki akademik performansının dönem sonu başarısının önemli bir belirleyicisi olduğunu ortaya koyar.

3.Keşifsel Veri Analizi (EDA) Makine Eğitimi

Kategorik ve Sayısal değişkenler

Makine veriyi olduğu gibi anlamaz. Model sayılarla çalışır metinle değil. İlgili sorgu ile kategorik ve sayısal metinleri gözlemleriz. Kategorik değişkenlerin makine öğrenmesi modelleri tarafından doğrudan kullanılamaması nedeniyle bu değişkenler **sayısal formata** dönüştürülmeli.

```
df.dtypes
```

```
school        object
sex           object
age            int64
address       object
famsize       object
Pstatus       object
Medu           int64
Fedu           int64
Mjob          object
Fjob          object
reason        object
guardian      object
traveltime     int64
studytime      int64
failures       int64
schoolsup     object
famsup        object
paid          object
activities    object
nursery       object
higher        object
internet      object
romantic      object
famrel         int64
freetime       int64
goout          int64
Dalc           int64
Walc           int64
health         int64
absences       int64
G1             int64
G2             int64
G3             int64
dtype: object
```

Kategorik değişkenleri sayısal formata dönüştürme

```
df_encoded = pd.get_dummies(df, drop_first=True)
df_encoded.head()
```

	age	Medu	Fedu	traveltime	studytime	failures	famrel	freetime	goout	Dalc	...	guardian_mother	guardian_other	schoolsup_yes	famsup_yes	paid_yes	ac
0	18	4	4	2	2	0	4	3	4	1	...	True	False	True	False	False	
1	17	1	1	1	2	0	5	3	3	1	...	False	False	False	True	False	
2	15	1	1	1	2	3	4	3	2	2	...	True	False	True	False	True	
3	15	4	2	1	3	0	3	2	2	1	...	True	False	False	True	True	
4	16	3	3	1	2	0	4	3	2	1	...	False	False	False	True	True	

Buradaki kodda kategorik sütunları otomatik algılayıp sayısal formata dönüştürdük.
Her kategori için yeni sütun açtık.
drop_first=True ile gereksiz tekrarları önledik.

Makine öğrenmesinin temeli

**

X = df_encoded.drop("G3", axis=1)
y = df_encoded["G3"]

**

x ve y ile ayrım yapıldı.
X Modelin girdiği tüm özellikler ve Y Tahmin etmeye çalıştığımız final notu.

Neden ayırıyoruz?
Eğer aynı veride eğitir ve aynı veride test edersek Gerçek perfonmansı göstermez ve ezber yapmış olur.

**

from sklearn.model_selection import train_test_split

X_train, X_test, y_train, y_test = train_test_split(

  X, y, test_size=0.2, random_state=42

)

**

Buradaki kodumuzda ise verinin %80 i eğitim % 20 si test olarak 2 e ayırdık.

random_state=42 ile sonuçlar tekrar üretilebilir olmasını sağladık.

SCALİNG

**

from sklearn.preprocessing import StandardScaler

scaler = StandardScaler()

X_train_scaled = scaler.fit_transform(X_train)

X_test_scaled = scaler.transform(X_test)

**

4.Yöntem / Ön İşleme

Modelimizin perfonmansını etkilememesi amacıyla Scaling ile tüm özellikler ortalamasını 0 ve standart sapmasını 1 yaptık.

Modeli Kurma

**
from sklearn.linear_model import LinearRegression

lr_model = LinearRegression()

lr_model.fit(X_train_scaled, y_train)

**

Çalışmamızda ilk olarak temel bir regresyon modeli olan Lineer Regresyon kullanılmıştır. Bu model bağımsız değişkenler ile hedef değişken arasındaki doğrusal ilişkiyi öğrenmeyi amaçlamaktadır.

**

y_pred_lr = lr_model.predict(X_test_scaled)
**
Test verisi için final not tahmini yapar.
Gerçek değerlere karşılaşırmaya hazır hale gelir.

**

from sklearn.ensemble import RandomForestRegressor

rf_model = RandomForestRegressor(

  n_estimators=100,

  random_state=42

)

rf_model.fit(X_train, y_train)

**

Random Forest scaling istemez.
Bu yüzden x_train kullanıyoruz.
Bu kodda;
100 Tane karar ağacından oluşan bir model kuruyoruz
her ağaç farklı bir açıdan veriye bakar
sonuçları ortalayıp daha sağlam tahmin yapar

Burada lineer regresyon modeline ek olarak doğrusal olmayan ilişkileride yakalayabilmek amacıyla random forest regressor modeli kullandım. Birden fazla ağaç birleşimiyle daha esnek ve güçlü tahminler üretir.

5.Modelleme

LINEAR REGRESSİON METRİKLERİ HESAPLAMA

[20]:
```
from sklearn.metrics import mean_absolute_error, mean_squared_error, r2_score
import numpy as np

mae_lr = mean_absolute_error(y_test, y_pred_lr)
mse_lr = mean_squared_error(y_test, y_pred_lr)
rmse_lr = np.sqrt(mse_lr)
r2_lr = r2_score(y_test, y_pred_lr)

mae_lr, rmse_lr, r2_lr
```

[20]:
```
(1.6466656197147496, np.float64(2.3783697847961363), 0.7241341236974023)
```

[ ]:

MAE

Model final notunu ortalama 1-1,7 puan hata ile tahmin etmekte.

MSE

Büyük hatalar da var ve hata biraz artıyor ama hala makul.

RMSE

Model final notundaki değişimin %85 ini açıklayabiliyor.

RMSE Değerinin yüksek olması modelin veri setindeki varasyonun büyük bir kısmını açıklayabildiğini ortaya koymaktadır.

RANDOM FOREST METRİKLERİ HESAPLAMA

```
mae_rf = mean_absolute_error(y_test, y_pred_rf)
mse_rf = mean_squared_error(y_test, y_pred_rf)
rmse_rf = np.sqrt(mse_rf)
r2_rf = r2_score(y_test, y_pred_rf)

mae_rf, rmse_rf, r2_rf
```

```
(1.1645569620253164, np.float64(1.9487730641858232), 0.8147911386865877)
```

Aynı metrikleri random forest için hesapladığımızda random forestin daha iyi tahmin yaptığını görürüz.

```
import pandas as pd

results = pd.DataFrame({
    "Model": ["Linear Regression", "Random Forest"],
    "MAE": [mae_lr, mae_rf],
    "RMSE": [rmse_lr, rmse_rf],
    "R2": [r2_lr, r2_rf]
})

results
```

	Model	MAE	RMSE	R2
0	Linear Regression	1.646666	2.378370	0.724134
1	Random Forest	1.164557	1.948773	0.814791

2 model karşılaştırıldığında ise Random Forest modelinin tüm metriklerde daha iyi performans gösterdiği görülür. Bu durum, öğrenci başarısını etkileyen değişkenler arasındaki ilişkilerin yalnızca doğrusal olmadığına rahatça işaret eder

6.Değerlendirme

Bu bölümde kurulan modellerin performansı deerlendirildi. Regresyon problemi olduğu için değerlendirme sürecinde MAE (Mean Absolute Error), RMSE (Root Mean Squared Error) ve R^2 metrikleri kullanıldı.

Lineer Regresyon modeli, final notlarını ortalama olarak düşük hata payı ile tahmin edebildi. Random Forest modeli ise daha düşük hata değerleri ve daha yüksek R^2 skoru elde ederek daha başarılı bir performans sergilediği görüldü.

Elde edilen sonuçlar, öğrenci başarısını etkileyen faktörler arasındaki ilişkilerin yalnızca doğrusal olmadığını ve daha esnek modellerin bu ilişkileri daha iyi yakalayabildiğini gösterildi.

7.Sonuç

Bu çalışmada öğrencilerin demografik, akademik ve sosyal özellikleri kullanılarak final notlarının tahmin edilmesi amaçlandı. Yapılan analizler sonucunda, kurulan modellerin öğrenci başarısını anlamlı bir doğrulukla tahmin edebildiği görüldü.

Özellikle Random Forest modeli, daha düşük hata oranları ile daha başarılı sonuçlar üretmiştir. Bu tür bir yaklaşım, eğitim kurumlarında akademik riski yüksek öğrencilerin erken tespit edilmesi ve gerekli desteklerin planlanması açısından faydalı olacaktır.

Çalışma, kullanılan veri seti ve değişkenlerle sınırlıdır. Gelecekte daha geniş veri setleri, farklı modelleme yöntemleri ve ek sosyal faktörler kullanılarak model performansı daha da artırılabilir.