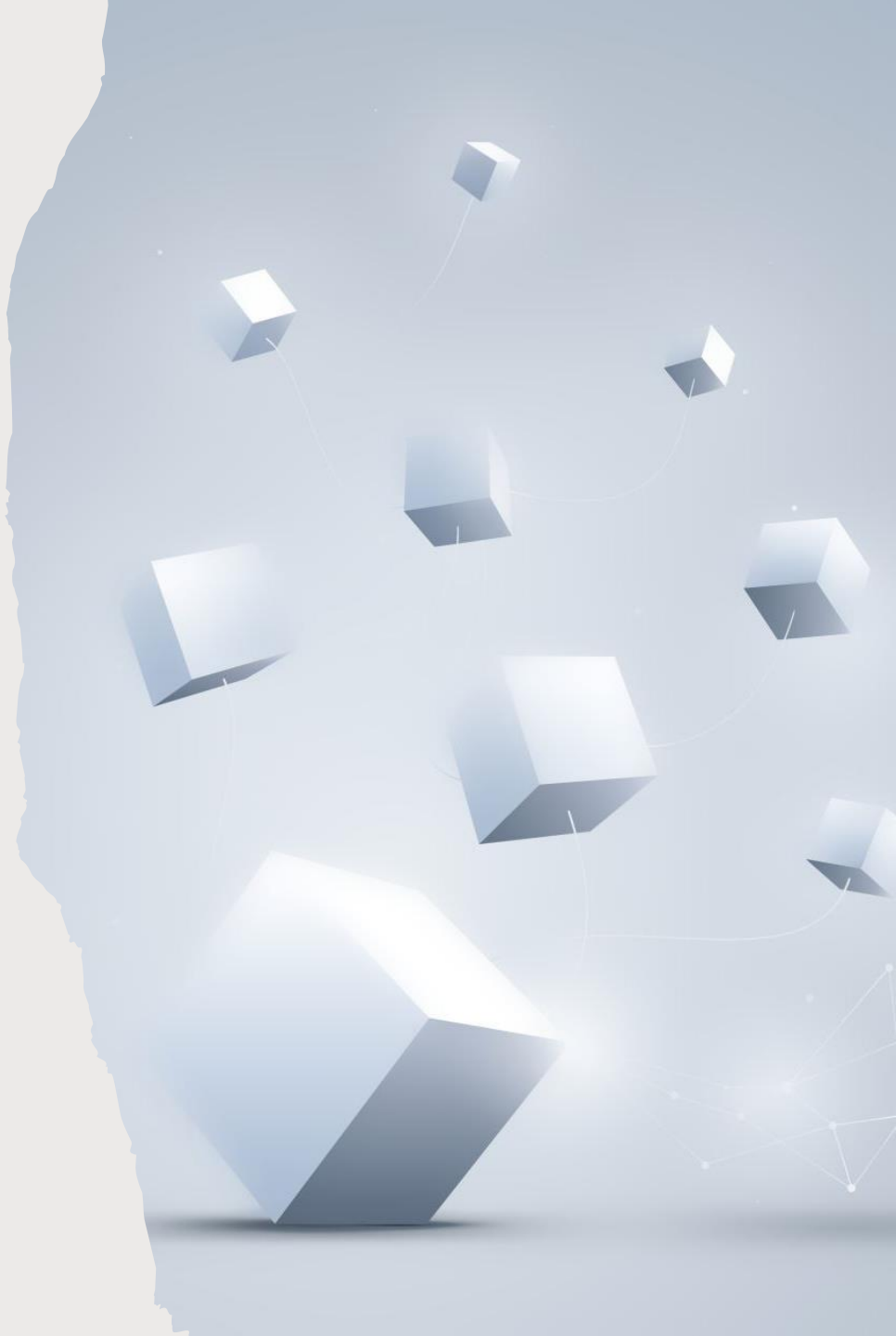
REGRESYON NEDİR ?

Alican Selen



REGRESYON NEDİR

- Veri madenciliğinde regresyon, bağımlı bir değişken ile bir veya daha fazla bağımsız değişken arasındaki ilişkiyi modellemeye yönelik bir tekniktir. Amaç, bağımlı değişkenin (sonuç) bağımsız değişkenler (girdiler) kullanılarak nasıl tahmin edileceğini öğrenmektir. Regresyon, veri analizi ve tahminleme için yaygın olarak kullanılır ve çeşitli türleri bulunmaktadır.

REGRESYON TÜRLERİ

Doğrusal Regresyon

Bağımlı değişkenin bağımsız değişkenlere göre doğrusal bir ilişkisinin olduğu varsayımına dayanan en temel regresyon türüdür.

Lojistik Regresyon

Bağımlı değişkenin ikili (evet/hayır) olduğu durumlarda kullanılan regresyon yöntemidir.

Polinomal Regresyon

Bağımlı değişkenin bağımsız değişkenlere göre doğrusal olmayan, polinomsal bir ilişkisinin olduğu durumlarda kullanılır.

Çoklu Regresyon

Bir bağımlı değişkenin, birden fazla bağımsız değişken tarafından açıklanabildiği durumlarda kullanılır.

REGRESYON ANALİZİ AVANTAJLARI

Regresyon analizleri genellikle hesaplama açısından hızlı ve uygulaması kolaydır.

Modelin açıklanabilirliği ve yorumu genellikle basittir.

REGRESYON ANALİZİ DEZAVANTAJLARI

1

Doğrusal regresyon, bağımsız ve bağımlı değişkenler arasındaki ilişkinin doğrusal olması varsayımına dayanır. Bu varsayım her zaman doğru olmayabilir.

2

Çoklu bağımsız değişkenler arasında yüksek korelasyon (multicollinearity) durumunda, modelin güvenilirliği düşebilir.

LİNEER REGRESYON

- Veri madenciliğinde lineer regresyon, bir bağımlı değişken ile bir veya daha fazla bağımsız değişken arasındaki doğrusal ilişkiyi modellemeye yönelik bir tekniktir. Lineer regresyon, değişkenler arasındaki ilişkinin doğrusal olduğunu varsayar ve bu ilişkiyi en iyi şekilde tanımlayan doğruyu (regresyon doğrusu) bulmayı amaçlar. Temel olarak, iki tür lineer regresyon vardır: basit lineer regresyon ve çoklu lineer regresyon.

BASİT LİNEER REGRESYON

- Basit lineer regresyon, bir bağımlı değişken (Y) ile bir bağımsız değişken (X) arasındaki ilişkiyi modelleyen en temel regresyon türüdür. Model şu şekildedir:
- $Y = a + b\,X + \epsilon$
- Y: Bağımlı değişken (tahmin edilen veya açıklanan değişken)
- X: Bağımsız değişken (açıklayıcı değişken)
- a: Y-intercept (doğrunun Y eksenini kestiği nokta)
- b: Eğim (bağımsız değişkenin katsayısı)
- ϵ: Hata terimi (modelin tahmin hatası)
- Bu denklem, bağımsız değişkendeki bir birimlik değişikliğin, bağımlı değişkende ne kadar değişiklik yaratacağını gösterir.

ÇOKLU LİNEER REGRESYON

- Çoklu lineer regresyon, birden fazla bağımsız değişkenin bir bağımlı değişken üzerindeki etkisini inceleyen regresyon türüdür. Model şu şekildedir:
- Y: Bağımlı değişken
- $X_1, X_2, \cdots, X_n$ X 1 ,X 2 ,⋯,X n : Bağımsız değişkenler
- a: Y-intercept
- $b_1, b_2, \cdots, b_n$ b 1 ,b 2 ,⋯,b n : Bağımsız değişkenlerin katsayıları
- ϵ: Hata terimi

$$y = a + b_1 x_1 + b_2 x_2 + \ldots + b_n x_n + \varepsilon$$

LİNEER REGRESYON AVANTAJLARI

Basitlik: Uygulaması ve yorumlanması genellikle basittir.

Hızlı Hesaplama: Genellikle hızlı bir şekilde hesaplanabilir.

Açıklanabilirlik: Modelin çıktıları kolayca yorumlanabilir ve anlaşılabilir.

LİNEER REGRESYON DEZAVANTAJLARI

Doğrusallık Varsayımı: Modelin doğruluğu, bağımsız ve bağımlı değişkenler arasındaki ilişkinin doğrusal olmasına bağlıdır. Bu her zaman doğru olmayabilir.

Hatalar: Çoklu bağımsız değişkenler arasında yüksek korelasyon olduğunda (multicollinearity), modelin tahmin gücü ve güvenilirliği düşebilir.

Heteroskedastisite: Hata terimlerinin varyansının sabit olmaması durumunda, model hatalı sonuçlar verebilir.

LİNEER REGRESYON KULLANIM ALANLARI

TAHMİN: GELECEKTEKİ OLAYLARIN VEYA DEĞERLERİN TAHMİN EDİLMESİ (ÖRNEĞİN, EV FİYATLARININ TAHMİNİ).

TREND ANALİZİ:VERİ SETLERİNDEKİ EĞİLİMLERİN ANALİZ EDİLMESİ (ÖRNEĞİN, SATIŞ TRENDLERİ).

SEBEP-SONUÇ İLİŞKİLERİ: DEĞİŞKENLER ARASINDAKİ NEDEN-SONUÇ İLİŞKİLERİNİN ANLAŞILMASI (ÖRNEĞİN, REKLAM HARCAMALARININ SATIŞLARA ETKİSİ).

- Lineer regresyon, veri madenciliği ve veri analizi süreçlerinde sıkça kullanılan, güçlü ve etkili bir tekniktir. Hem basit hem de çoklu lineer regresyon modelleri, veriler arasındaki ilişkilerin anlaşılmasına ve tahminlerin yapılmasına yardımcı olur.

LOJİSTİK REGRESYON

- Veri madenciliğinde lojistik regresyon, bağımlı değişkenin kategorik olduğu durumlarda kullanılan bir regresyon analizidir. Özellikle bağımlı değişkenin iki kategoriye (örneğin, evet/hayır, 0/1) ayrıldığı durumlar için uygundur. Lojistik regresyon, bağımlı değişkenin belirli bir kategoriye ait olma olasılığını tahmin etmek için kullanılır.

LOJİSTİK REGRESYON

$$P(Y=1) = \frac{1}{1+e^{-\left(a+b_1X_1+b_2X_2+\ldots+b_nX_n\right)}}$$

P(Y=1): Bağımlı değişkenin 1 (veya "evet") olma olasılığı

e: Doğal logaritmanın tabanı (yaklaşık 2.718)

a :Y-intercept

b 1 , b 2 , ⋯ , b n b 1 ,b 2 ,⋯,b n : Bağımsız değişkenlerin katsayıları

X 1 , X 2 , ⋯ , X n X 1 ,X 2 ,⋯,X n : Bağımsız değişkenler

KULLANIM ALANLARI

Tıp: Hastaların belirli bir hastalığa yakalanma olasılığının tahmin edilmesi.

Pazarlama: Müşterilerin belirli bir ürünü satın alma olasılığının tahmin edilmesi.

Finans: Kredilerin geri ödenmeme olasılığının tahmin edilmesi.

Sosyal Bilimler: Seçmenlerin belirli bir adaya oy verme olasılığının tahmin edilmesi.

AVANTAJLARI

Kategorik Bağımlı Değişken: Bağımlı değişkenin kategorik olduğu durumlar için uygundur.

Olasılık Tahmini: Sonuçları olasılık olarak ifade eder, bu da sonuçların yorumlanmasını kolaylaştırır.

Esneklik: Hem sürekli hem de kategorik bağımsız değişkenlerle çalışabilir.

DEZAVANTAJLARI

Lineer Olmayan İlişki Varsayımı: Doğrusal regresyonun aksine, bağımsız değişkenler ve logit fonksiyonu arasında doğrusal bir ilişki varsayar.

Veri Dengesi: Bağımlı değişkenin kategorileri arasında ciddi bir dengesizlik varsa model performansı düşebilir.

Outlier Hassasiyeti: Aykırı değerler modelin performansını olumsuz etkileyebilir.

PERFORMANS ÖLÇÜTLERİ

- Lojistik regresyonun performansı, çeşitli ölçütler kullanılarak değerlendirilebilir:

Doğruluk (Accuracy): Modelin doğru sınıflandırma oranı.

Kesinlik (Precision): Pozitif sınıflandırmaların doğruluğu.

Duyarlılık (Recall): Gerçek pozitiflerin doğru tanımlanma oranı.

F1 Skoru: Kesinlik ve duyarlılığın harmonik ortalaması.

ROC Eğrisi ve AUC: Modelin sınıflandırma performansını değerlendirir.

Lojistik regresyon, veri madenciliği ve makine öğreniminde önemli bir yer tutar. Bağımlı değişkenin kategorik olduğu durumlarda güçlü ve etkili bir tahmin aracı olarak kullanılır.

LINEER REGRESYON ÖRNEĞI

```python
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LinearRegression
from sklearn.metrics import mean_squared_error, r2_score
```

```python
# Rastgele bir veri seti oluşturma
np.random.seed(0)  # Rastgele sayı üreticisi için sabit bir başlangıç değeri belirleyin
X = 2 * np.random.rand(100, 1)  # 100 adet rastgele X değeri oluşturun (0-2 arası)
y = 4 + 3 * X + np.random.randn(100, 1)  # y = 4 + 3X + rastgele gürültü ekleyin
```

```python
# Veriyi eğitim ve test setlerine ayırma
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)
```

```python
# Lineer regresyon modelini oluşturma
model = LinearRegression()
model.fit(X_train, y_train)  # Modeli eğitim verisiyle eğitin
```

```
LinearRegression()
```

LINEER REGRESYON ÖRNEĞI

```
[ ] # Modeli test verisi ile kullanarak tahmin yapma
    y_pred = model.predict(X_test)  # Test verisi üzerinde tahminler yapın
```

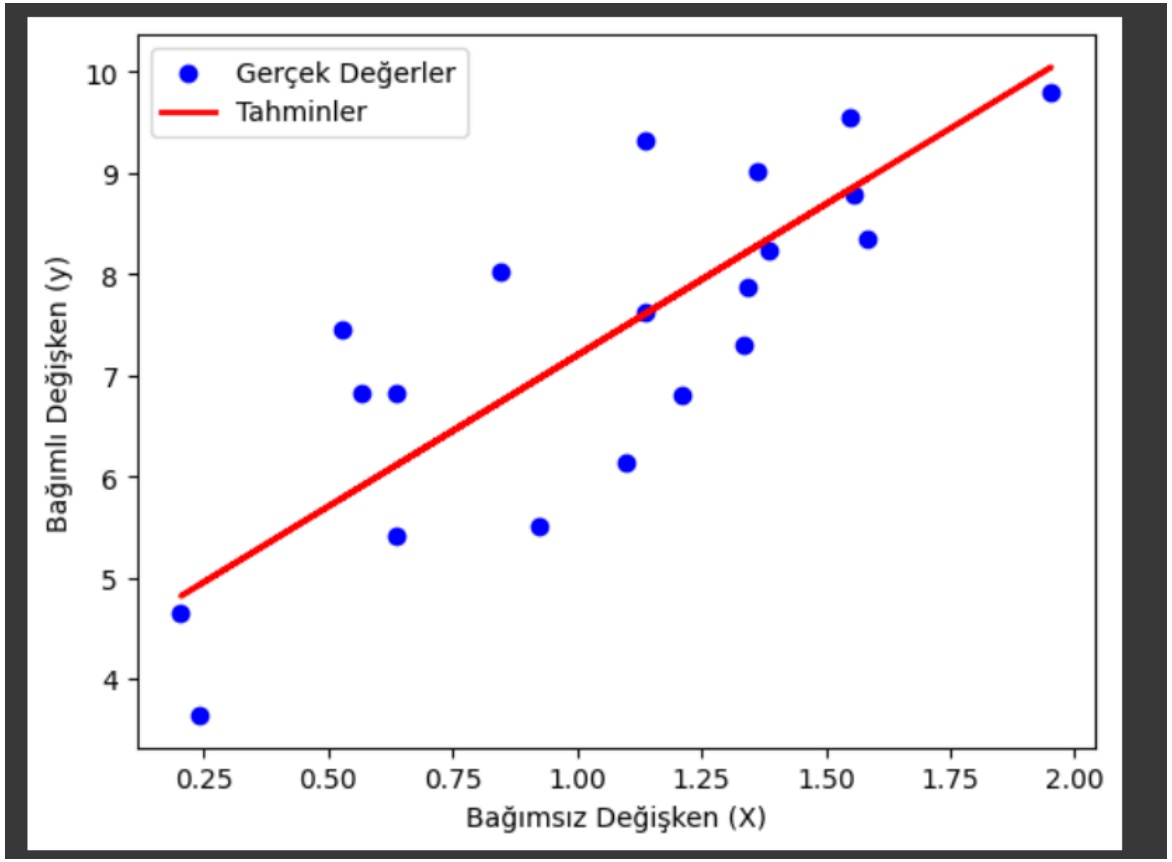
```
[ ] # Modelin performansını değerlendirme
    mse = mean_squared_error(y_test, y_pred)  # Ortalama Kare Hatasını hesaplayın
    r2 = r2_score(y_test, y_pred)  # R^2 skorunu hesaplayın
```

```
    # Performans metriklerini ekrana yazdırma
    print(f"Mean Squared Error: {mse}")
    print(f"R^2 Score: {r2}")
```

```
Mean Squared Error: 0.9177532469714291
R^2 Score: 0.6521157503858556
```

```
[ ] # Sonuçları görselleştirme
    plt.scatter(X_test, y_test, color='blue', label='Gerçek Değerler')  # Test verisinin gerçek değerlerini çizdirin
    plt.plot(X_test, y_pred, color='red', linewidth=2, label='Tahminler')  # Modelin tahminlerini çizdirin
    plt.xlabel('Bağımsız Değişken (X)')  # X eksenini etiketleyin
    plt.ylabel('Bağımlı Değişken (y)')  # Y eksenini etiketleyin
    plt.legend()  # Grafik için bir açıklama ekleyin
    plt.show()  # Grafiği gösterin
```

LINEER REGRESYON ÖRNEĞI



LOJIK REGRESYON ÖRNEĞI

```
import pickle
import pandas as pd
import nltk
nltk.download('stopwords')
from sklearn.linear_model import LogisticRegression
from sklearn.feature_extraction.text import TfidfVectorizer
from sklearn.model_selection import train_test_split
from sklearn.metrics import confusion_matrix
from sklearn.metrics import classification_report
from nltk import word_tokenize
from nltk.corpus import stopwords
import re

dataset = pd.read_csv('/content/drive/MyDrive/Colab Notebooks/dataset.csv')

dataset = dataset.drop(columns='B')

dataset.drop_duplicates(subset="Body" , keep= False , inplace= True)
```

LOJIK REGRESYON ÖRNEĞI

```python
def optimizasyon(dataset):
    dataset = dataset.dropna() #bos veri iceren verileri siler

    stop_words = set(stopwords.words('turkish'))
    noktalamaIsaretleri = ['•', '!', '"', '#', '”', '“', '$', '%', '&', "'", '-', '(', ')', '*', '+', ',', '-', '.', '/', ':', ';', '<', '=', '>', '?', '@', '[', '\'
    stop_words.update(noktalamaIsaretleri)

    for ind in dataset.index:
        body = dataset['Body'][ind]
        body = body.lower()
        body = re.sub(r'http\S+', '', body)#url kaldır
        body = re.sub('\[[^]]*\]', '', body)  # Köşeli parantez içeriğini kaldır
        body = (" ").join([word for word in body.split() if not word in stop_words]) # Stopwords'leri kaldır
        body = "".join([char for char in body if not char in noktalamaIsaretleri])  # Noktalama işaretlerini kaldır
        dataset['Body'][ind] = body # Temizlenmiş metni geri yaz
    return dataset
```

LOJIK REGRESYON ÖRNEĞI

```
[ ]  dataset = optimizasyon(dataset)
```

```
[ ]  # Label değerine göre veri setini ayır

     yorumlar_makina = dataset[dataset['Label']==0]
     yorumlar_insan = dataset[dataset['Label']==1]
```

```
     yorumlar_insan
```

	id	Body	Label
0	1702	-Ayrıca redmi note 9 aldım almaz olsaydım.	1
1	2054	-Cihazın kendisine geldiğimde ise	1
2	2059	-Genel olarak toparlamam gerekirse, cihazın er...	1
3	1701	-Redmi note 8 sağlam telefon.	1
4	1823	-Yanları. Telefon ağır ve kamera çıkıntısı büy...	1
...	...	...	...
2382	1727	Şu an bu üründen yazıyorum yorumu ürün çok iyi 👌	1
2383	1671	Şu ana kadar kullandığım en iyi maskara kesinl...	1
2384	1949	Şu sıkıntılı günlerde geç gelir diye korkmuştu...	1

LOJIK REGRESYON ÖRNEĞI

```
[ ]  tfIdf = TfidfVectorizer(binary=False , ngram_range=(1,3))

     makina_vec = tfIdf.fit_transform(yorumlar_makina['Body'].tolist())
     insan_vec = tfIdf.fit_transform(yorumlar_insan['Body'].tolist())
```

```
[ ]  x = dataset['Body']
     y = dataset['Label']
```

```
[ ]  x_vec = tfIdf.fit_transform(x)
```

LOJIK REGRESYON ÖRNEĞI

```python
# Eğitim ve test veri setlerine ayır
x_egitim_vec, x_test_vec, y_egitim, y_test = train_test_split(x_vec, y, test_size=0.2, random_state=0)

# Lojistik regresyon modeli oluştur ve eğit
lojistikRegresyon = LogisticRegression()
lojistikRegresyon.fit(x_egitim_vec,y_egitim)

# Test veri seti üzerinde tahmin yap
y_tahmin = lojistikRegresyon.predict(x_test_vec)

# Eğitilmiş modeli dosyaya kaydet
pickle.dump(lojistikRegresyon, open("egitilmis_model", 'wb'))
print("Lojistik Regresyon modeli eğitildi ve kayıt edildi !")
```

```
Lojistik Regresyon modeli eğitildi ve kayıt edildi !
```

LOJIK REGRESYON ÖRNEĞI

```
[ ]  # TF-IDF vektörleştirici modelini dosyaya kaydet
     pickle.dump(tfIdf, open("vektorlestirici", 'wb'))
     print("Tf-Idf vektörleştirici modeli kayıt edildi !")
     # Sonuçları yazdır
     print(confusion_matrix(y_test,y_tahmin))
     print(classification_report(y_test,y_tahmin))
     exit()
```

```
Tf-Idf vektörleştirici modeli kayıt edildi !
[[239   4]
 [ 83 153]]
              precision    recall  f1-score   support

           0       0.74      0.98      0.85       243
           1       0.97      0.65      0.78       236

    accuracy                           0.82       479
   macro avg       0.86      0.82      0.81       479
weighted avg       0.86      0.82      0.81       479
```

ALİCAN SELEN