

# Yıllara Göre Büyüme Oranının Tahmin Edilmesi

17110131006-Mehmet OKYAY

18110131505-Abdourazak ALİ EGUEH

# Projemizin Amacı

- Projemizin amacı 1950-2011 yılları arasında ülkelerini nüfus bilgileri doğrultusunda belirlenmiş bir ülkenin, belirli bir yıla ait dair tahminlerde bulunmasını sağlamak.

# Veri seti Hakkında

- ▶ Veri toplama adımı olarak Kaggle yer alan bir veri seti kullanılmıştır.
- ▶ Veri seti Birleşmiş Milletler(BM) tarafından 1950 yılında itibaren ülkelerin nüfus bilgilerinin yer aldığı bir veri setidir.
- ▶ Veri setinde 228 ülkenin 15109 adet kayıtlık nüfus verisi yer almaktadır.

# Veri Önışleme Adımları(Python Kütüphanelerini Yükleme)

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

# Veri Önışleme Adımları(Veri seti Hakkında)

	Ülke Kodu	Ülke Adı	Yıl	Kaba Doğum Oranı	Gerçek Ölüm Oranı	Net Göç	Doğal Artış Oranı	Büyüme_Oranı
0	SI	Slovenia	2036	7.59	14.65	0.23	-0.706	-0.682
1	SI	Slovenia	2022	7.52	12.28	0.32	-0.476	-0.444
2	SI	Slovenia	2023	7.43	12.41	0.31	-0.498	-0.467
3	SI	Slovenia	2024	7.35	12.55	0.31	-0.520	-0.489
4	SI	Slovenia	2025	7.30	12.68	0.30	-0.538	-0.508
...	...	...	...	...	...	...	...	...
15104	BP	Solomon Islands	2000	34.93	4.53	-2.51	3.040	2.789
15105	BP	Solomon Islands	1999	35.48	4.61	-2.58	3.087	2.829
15106	BP	Solomon Islands	1998	35.98	4.70	-2.45	3.128	2.884
15107	BP	Solomon Islands	1997	36.44	4.79	-2.31	3.165	2.934
15108	BP	Solomon Islands	1995	37.17	4.97	-2.01	3.220	3.019

# Veri Önışleme Adımları(Veri seti Hakkında)

Veri setinde yer alan veri türleri aşığıda görüntülenmektedir. Ayrıca veri tipi ve boş deęer içirip içirmedięi hakkında genel bilgi verilmiřtir.

```
veriseti.info()
```

```
<class 'pandas.core.frame.DataFrame'>  
RangeIndex: 15109 entries, 0 to 15108  
Data columns (total 8 columns):  
#   Column                Non-Null Count  Dtype    
---  ---                  
0   Ülke Kodu              15109 non-null  object   
1   Ülke Adı               15109 non-null  object   
2   Yıl                    15109 non-null  int64    
3   Kaba Doğum Oranı       15109 non-null  float64  
4   Gerçek Ölüm Oranı      15109 non-null  float64  
5   Net Göç                15109 non-null  float64  
6   Doğal Artış Oranı      15109 non-null  float64  
7   Büyüme_Oranı           15109 non-null  float64  
dtypes: float64(5), int64(1), object(2)  
memory usage: 944.4+ KB
```

# Veri Önışleme Adımları(Veri seti Hakkında)

- Verisetindeki temel istatistiki veriler.

```
veriseti.describe()
```

	Yıl	Kaba Doğum Oranı	Gerçek Ölüm Oranı	Net Göç	Doğal Artış Oranı	Büyüme_Oranı
<b>count</b>	15109.000000	15109.000000	15109.000000	15109.000000	15109.000000	15109.000000
<b>mean</b>	2016.748891	20.645532	9.137688	-0.041234	1.150771	1.109136
<b>std</b>	20.220176	11.714597	4.409634	29.716751	1.114461	3.155930
<b>min</b>	1950.000000	3.620000	1.530000	-831.810000	-6.383000	-168.944000
<b>25%</b>	2001.000000	11.160000	6.230000	-2.130000	0.250000	0.223000
<b>50%</b>	2017.000000	16.690000	8.130000	-0.090000	1.038000	0.938000
<b>75%</b>	2034.000000	27.890000	11.050000	1.210000	2.075000	1.989000
<b>max</b>	2050.000000	58.740000	97.080000	1693.010000	4.036000	168.887000

# Veri Önışleme Adımları(Verisetinin düzenleme)

- Sütunların yeniden adlandırılması

```
veriseti=pd.read_csv("dataset.csv")
veriseti=veriseti.rename(columns={"country_code":"Ülke Kodu","country_name":"Ülke Adı","year":"Yıl"})
veriseti=veriseti.rename(columns={"crude_birth_rate":"Kaba Doğum Oranı","crude_death_rate":"Gerçek Ölüm Oranı"})
veriseti=veriseti.rename(columns={"net_migration":"Net Göç","rate_natural_increase":"Doğal Artış Oranı"})
veriseti=veriseti.rename(columns={"growth_rate":"Büyüme Oranı"})
veriseti
```

- Ülke adı ve ülke kodu bilgileri yer alan veri setinde ülke kodu gereksiz olduğu düşünülerek Ülke Kodu sütunu silinmiştir.

```
veriseti.drop('Ülke Kodu', inplace=True, axis=1)
```



# Veri Önışleme Adımları

- ▶ Verilerimizin eğitim ve test verilerinin ayrıştırılması için ilk 6 sütun bağımsız değişken olarak belirlendi ve numpy dizini olarak X değişkeninde hafızaya alındı.
- ▶ Son sütun olan nüfusun büyüme oranı ise bağımlı değişken olarak(y) alındı.

```
X = veriseti.iloc[:, :-1]
```

```
y = veriseti.iloc[:, 6]  
y.head()
```

```
X = veriseti.iloc[:, 1:6].values  
X = np.array(X)  
#y = veriseti.iloc[:, 6].values  
print('X:',X)  
print('Y:',y)
```

```
X: [[ 2.036e+03  7.590e+00  1.465e+01  2.300e-01 -7.060e-01]  
 [ 2.022e+03  7.520e+00  1.228e+01  3.200e-01 -4.760e-01]  
 [ 2.023e+03  7.430e+00  1.241e+01  3.100e-01 -4.980e-01]  
 ...  
 [ 1.998e+03  3.598e+01  4.700e+00 -2.450e+00  3.128e+00]  
 [ 1.997e+03  3.644e+01  4.790e+00 -2.310e+00  3.165e+00]  
 [ 1.995e+03  3.717e+01  4.970e+00 -2.010e+00  3.220e+00]]
```

```
Y: 0      -0.682  
1      -0.444  
2      -0.467  
3      -0.489  
4      -0.508
```

```
...  
15104    2.789  
15105    2.829  
15106    2.884  
15107    2.934  
15108    3.019
```

```
Name: Büyüme_Oranı, Length: 15109, dtype: float64
```

# Modelin Eğitilmesi

R2 hata oranına bakılarak **test\_size = 0,33** ve **random\_state=0** olarak belirlendi.

```
from sklearn.model_selection import train_test_split
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size = 0.33, random_state = 0)

from sklearn.linear_model import LinearRegression
regressor = LinearRegression()
regressor.fit(X_train, y_train)

# Predicting the Test set results
y_pred = regressor.predict(X_test)
```

# Tahmin İşlemleri

- Tahmin işlemleri için linear regresyon modeli kullanıldı.

```
from sklearn.model_selection import train_test_split
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size = 0.33, random_state = 0)
```

```
from sklearn.linear_model import LinearRegression
regressor = LinearRegression()
regressor.fit(X_train, y_train)
```

```
# Predicting the Test set results
y_pred = regressor.predict(X_test)
```

```
regressor.predict([[2036,7.59,14.65,0.23,-0.706]])
array([-0.78097056])
```

```
from sklearn.metrics import r2_score
score=r2_score(y_test,y_pred)
```

```
print(y_pred)
```

```
[-0.70716749  2.10005263  1.10071767 ...  2.34913785  2.44773868
  2.84661336]
```

```
print(score)
```

```
0.09529090418610942
```

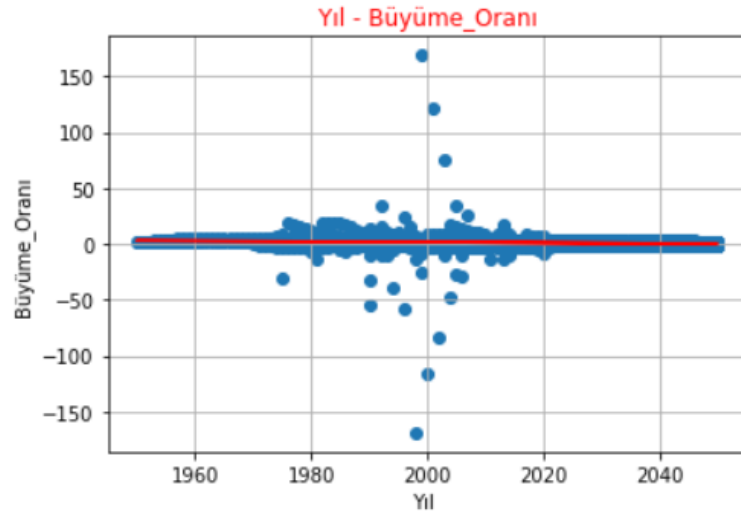
# Veri Görselleştirme

## Basit Doğrusal Regresyon Modeli ile Test/Tahmin/Görselleştirme

Geçen bölümde eğitimini gerçekleştirdiğimiz basit doğrusal regresyon modelimizi kullanarak tahminler yapacağız ve sonuçları görselleştireceğiz.

```
array = veriseti.Yıl.values.reshape(-1,1) # yıla

plt.figure()
plt.scatter(x,y)
y_head = model.predict(array)
plt.plot(array, y_head, color = "red")
plt.xlabel("Yıl ")
plt.ylabel("Büyüme_Oranı")
plt.title("Yıl - Büyüme_Oranı", color = "red")
plt.grid(True)
plt.show()
```



# R2 İle Eğitimin İyileştirilmesi(Geliştirme)

```
print("R2 score hata oranı:",r2_score(y,yeni_tahmin))
```

```
R2 score hata oranı: 0.12861112746857983
```

---