

```
In [1]: import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
import pandas as pd
from sklearn.preprocessing import StandardScaler
from IPython.display import display, HTML
import re

In [2]: data = pd.read_excel('arcgis_ngas_usak_v02.xlsx')
df = data.copy()
```

```
In [3]: def getting_primary_info(df):
    print("-----")
    print("Veri setinin şekli", df.shape)
    print("-----")
    print("Veri seti değişken tipleri:\n", df.dtypes)
    print("-----")
    print("Veri setinin ilk 5 satırını")
    display(HTML(df.head().to_html()))
    print("-----")
    print("Veri setinin istatistikli verileri")
    description = df.describe()
    display(HTML(description.to_html()))
    print("-----")
    getting_primary_info(df)

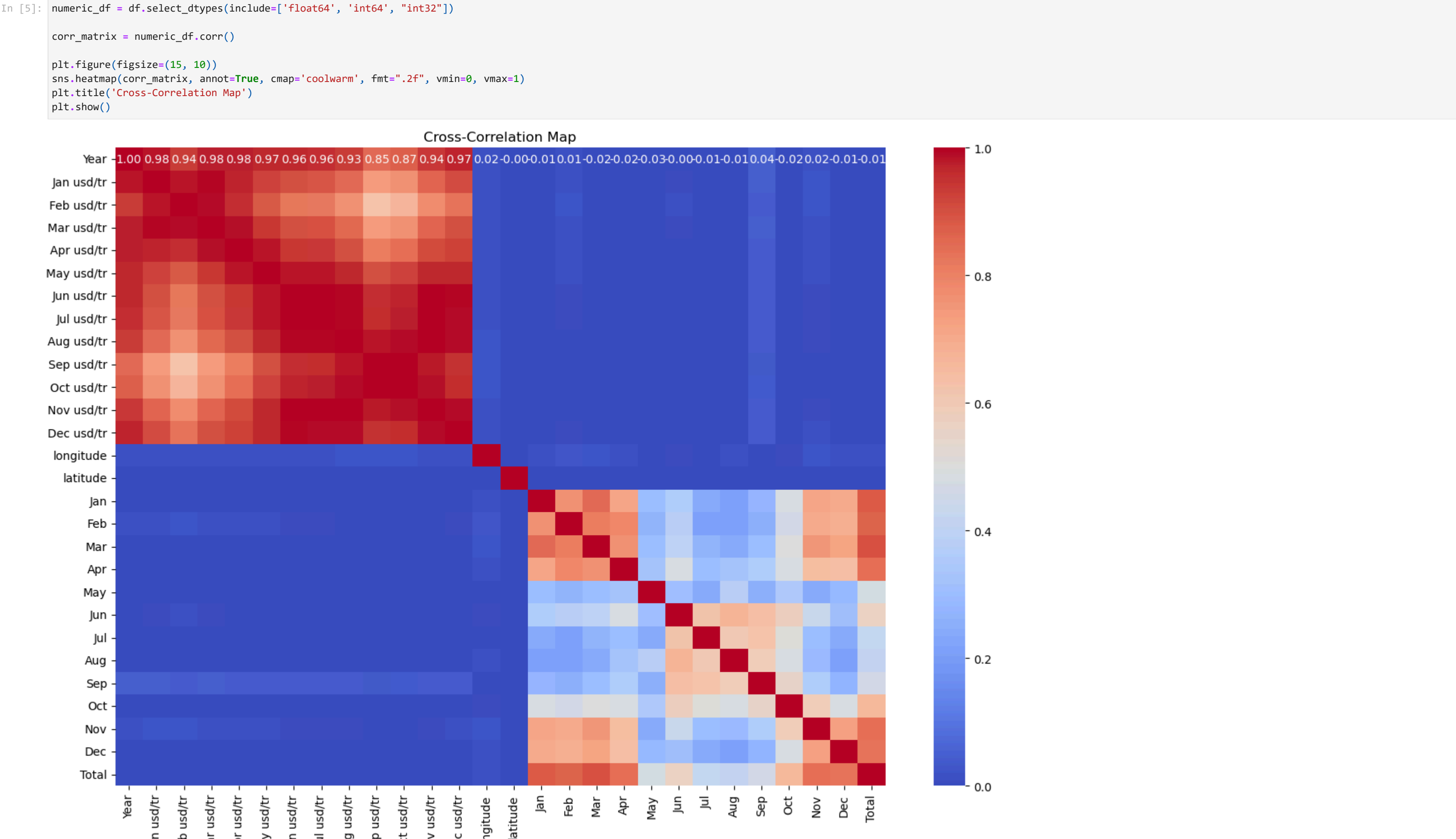
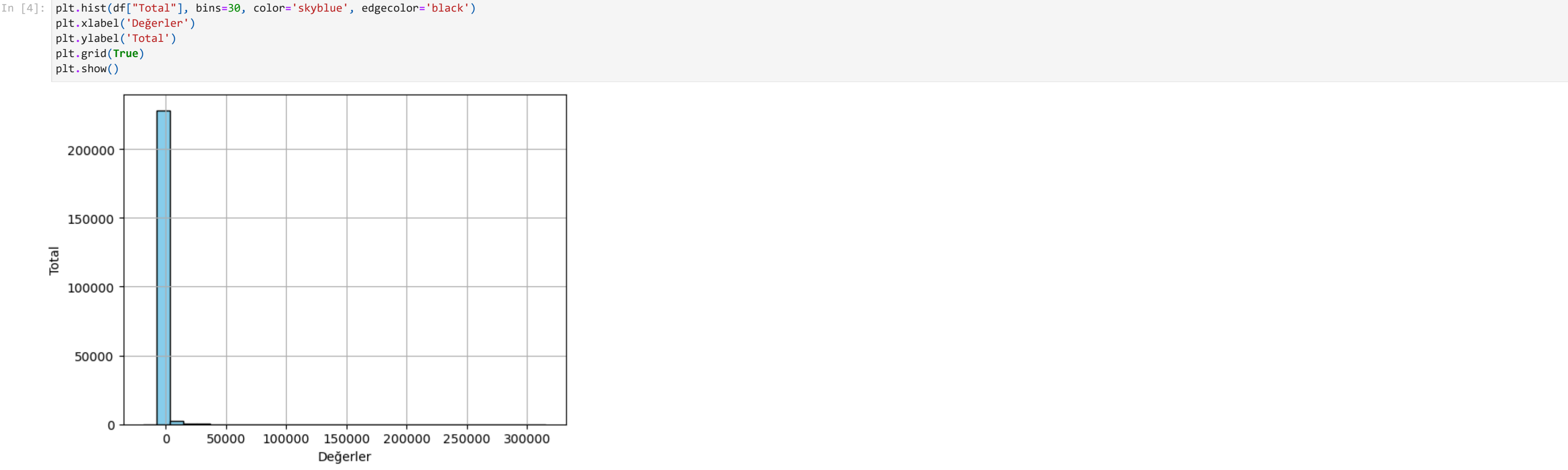
-----
Veri setinin şekli (231623, 28)
-----
Veri seti değişken tipleri:
Year          int64
Jan usd/tr    float64
Feb usd/tr    float64
Mar usd/tr    float64
Apr usd/tr    float64
May usd/tr    float64
Jun usd/tr    float64
Jul usd/tr    float64
Aug usd/tr    float64
Sep usd/tr    float64
Oct usd/tr    float64
Nov usd/tr    float64
Dec usd/tr    float64
longitude     float64
latitude      float64
Jan           float64
Feb           float64
Mar           float64
Apr           float64
May           float64
Jun           float64
Jul           float64
Aug           float64
Sep           float64
Oct           float64
Nov           float64
Dec           float64
Total         float64
dtype: object
-----
Veri setinin ilk 5 satırını
```

	Year	Jan usd/tr	Feb usd/tr	Mar usd/tr	Apr usd/tr	May usd/tr	Jun usd/tr	Jul usd/tr	Aug usd/tr	Sep usd/tr	Oct usd/tr	Nov usd/tr	Dec usd/tr	longitude	latitude	Jan	Feb	Mar	Apr	May	Jun	Jul	Aug	Sep	Oct	Nov	Dec	Total
0	2015	2.3449	2.4236	2.5187	2.6008	2.7109	2.6762	2.6992	2.7736	2.9147	3.0174	2.8035	2.8923	29.435552	38.655245	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
1	2015	2.3449	2.4236	2.5187	2.6008	2.7109	2.6762	2.6992	2.7736	2.9147	3.0174	2.8035	2.8923	29.404893	38.666263	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
2	2015	2.3449	2.4236	2.5187	2.6008	2.7109	2.6762	2.6992	2.7736	2.9147	3.0174	2.8035	2.8923	29.386129	38.678564	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
3	2015	2.3449	2.4236	2.5187	2.6008	2.7109	2.6762	2.6992	2.7736	2.9147	3.0174	2.8035	2.8923	29.416003	38.682545	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
4	2015	2.3449	2.4236	2.5187	2.6008	2.7109	2.6762	2.6992	2.7736	2.9147	3.0174	2.8035	2.8923	29.402495	38.638435	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0

```
-----
Veri setinin istatistikli verileri
```

	Year	Jan usd/tr	Feb usd/tr	Mar usd/tr	Apr usd/tr	May usd/tr	Jun usd/tr	Jul usd/tr	Aug usd/tr	Sep usd/tr	Oct usd/tr	Nov usd/tr	Dec usd/tr	longitude
count	231623.000000	231623.000000	231623.000000	231623.000000	231623.000000	231623.000000	231623.000000	231623.000000	231623.000000	231623.000000	231623.000000	231623.000000	231623.000000	231623.000000
mean	2016.648847	3.218489	3.301640	3.293438	3.326757	3.364186	3.531916	3.530348	3.652933	4.154344	4.03256	3.955177	3.976242	29.442662
std	1.109379	0.538237	0.552588	0.510438	0.558533	0.584172	0.750087	0.769689	0.863947	1.609658	1.27269	1.105374	0.862284	0.504281
min	2015.000000	2.344900	2.423600	2.518700	2.600800	2.710900	2.676200	2.699200	2.773600	2.914700	3.00360	2.803500	2.892300	27.047201
25%	2016.000000	2.942200	2.967100	2.945100	2.819700	2.792800	2.948900	2.884600	2.979700	2.955800	3.01740	3.098100	3.448300	29.392015
50%	2017.000000	3.533800	3.756700	3.631300	3.637500	3.534800	3.531200	3.530500	3.516900	3.426400	3.57570	3.808000	3.932400	29.400218
75%	2018.000000	3.765200	3.774600	3.808200	3.957700	4.120800	4.592200	4.622500	4.922300	6.621300	5.96590	5.570400	5.190600	29.409796
max	2018.000000	3.765200	3.774600	3.808200	3.957700	4.120800	4.592200	4.622500	4.922300	6.621300	5.96590	5.570400	5.190600	42.762756

```
-----
```



```
In [6]: from sklearn.linear_model import LinearRegression
from sklearn.metrics import r2_score
from sklearn.model_selection import train_test_split
from sklearn.ensemble import RandomForestRegressor
from sklearn.ensemble import GradientBoostingRegressor

df = df.dropna()
X , y = df.drop(columns=["Year", "Total"]), df["Total"]

X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=47)
model = LinearRegression()
model.fit(X_train, y_train)

y_pred = model.predict(X_test)

r2 = r2_score(y_test, y_pred)
print("Linear Regression Test R-squared:", r2)

y_pred_train = model.predict(X_train)
r2_train = r2_score(y_train, y_pred_train)
print("Linear Regression Train R^2 Score:", r2_train)
print("-----")

model_random_forest = RandomForestRegressor(n_estimators=20, min_samples_split=2)
model_random_forest.fit(X_train, y_train)

y_pred_rf = model_random_forest.predict(X_test)
r2_rf = r2_score(y_test, y_pred_rf)
print("Random Forest Test R^2 Score:", r2_rf)

y_pred_rf_train = model_random_forest.predict(X_train)
r2_rf_train = r2_score(y_train, y_pred_rf_train)
print("Random Forest Train R^2 Score:", r2_rf_train)

model_gb = GradientBoostingRegressor(n_estimators=80, min_samples_split=2)
model_gb.fit(X_train, y_train)

print("-----")
y_pred_gb = model_gb.predict(X_test)
r2_gb = r2_score(y_test, y_pred_gb)
print("Gradient Boosting Test R^2 Score:", r2_gb)

y_pred_gb_train = model_gb.predict(X_train)
r2_gb_train = r2_score(y_train, y_pred_gb_train)
print("Gradient Boosting Train R^2 Score:", r2_gb_train)

Linear Regression Test R-squared: 1.0
Linear Regression Train R^2 Score: 1.0
-----
```

Random Forest Test R^2 Score: 0.9985486285798618
Random Forest Train R^2 Score: 0.9927714643368053
.....
Gradient Boosting Test R^2 Score: 0.9901525381385884
Gradient Boosting Train R^2 Score: 0.9964645277055614