FUTURE DATA - LETES

OLYMPIC ANALYSIS & PREDICTIONS

Maggie Sun | Anggiela Yupanqui | Sana Momin | Luke Fuller

What are the olympic games ?

- The Olympic Games is an important international event featuring summer and winter sports. Summer Olympic Games and Winter Olympic Games are held every four years. The Olympic Games include 206 countries. Those countries are represented with their best athletes. There are three classes of medals to be won: gold, silver, and bronze, awarded to first, second, and third place, respectively.
- Olympic facts:
  - The U.S. has won a total of 2,960 medals.
  - The Soviet Union sits second for total medals 1,204
  - Germany is third with 1,056 medals.
  - For Team USA , American women won 55.8% of the medals at the 2012 London Olympics.
  - For the past 4 consecutive Summer Olympics the U.S. women have won more medals than U.S. men.
  - For the past 3 consecutive Summer Games women have outnumbered men on the U.S. team.

Predictions :

- *Questions we want to answer:*
  - Can we predict how many medals USA will win in 2020 Tokyo Olympics?
  - Within the medal winning countries, what will be the predicted performance of female and male compared to the overall.

Requirements :

- Produce an analytical model in Python that Script initializes, trains, and evaluates a model, or loads a pre-trained model from hyper-parameter tuning
- Cleans, normalizes, and standardizes input data prior to modeling.
- Utilizes data retrieved from a relational database or big data source
- Demonstrates meaningful predictive power >75% classification accuracy, >80 R-squared

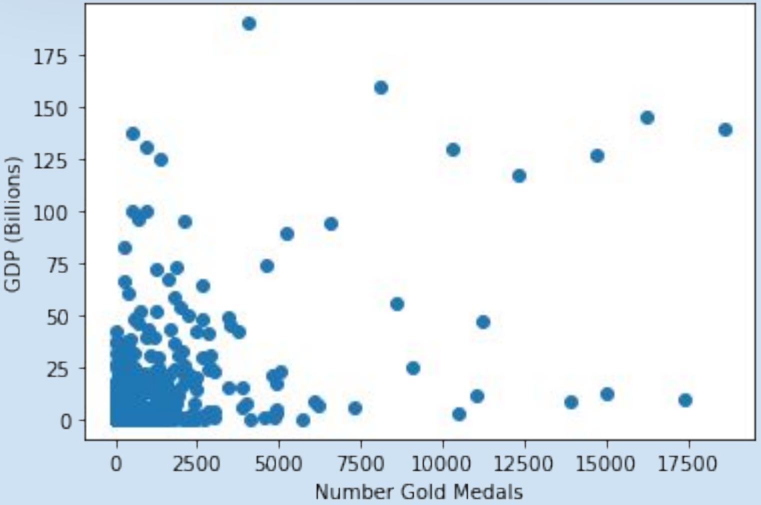
Technology Stack :

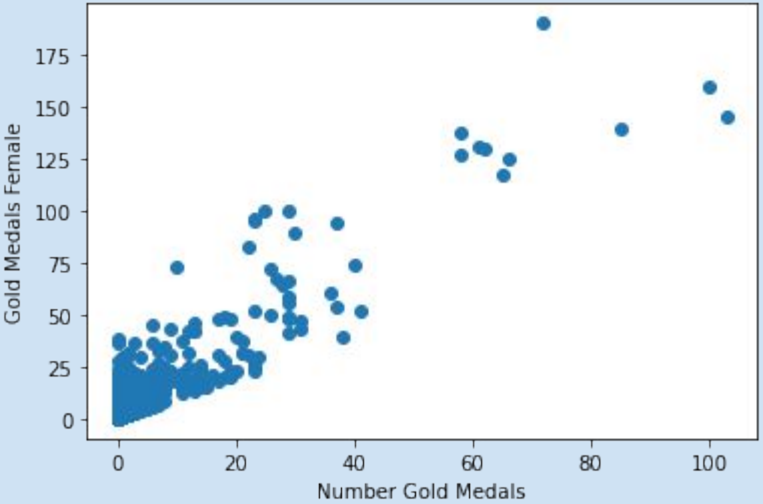
- Excel/CSV file
- PostGres RDB
- Python Flask-Api
- Python - Pandas & Matplotlib
- Python ML Libraries - Scikit

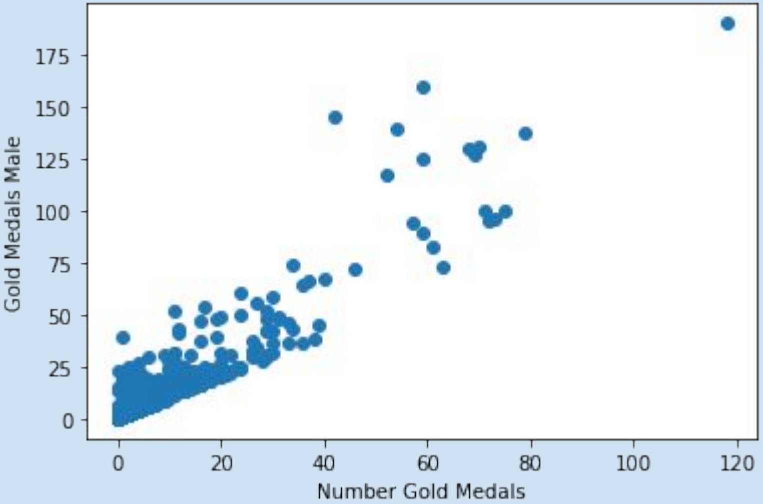
Data Source :

- Data set from Kaggle:
- https://www.kaggle.com/heesoo37/120-years-of-olympic-history-athletes-and-results
- It includes all the Games from Athens 1896 to Rio 2016.
- Contains 271116 rows and 15 columns. Each row corresponds to an individual athlete competing in an individual Olympic event.
- There are 15 columns : ID, Name, Sex, Age, Height, Weight, Team, NOC, Games, Year, Season, City, Sport, Event, and Medal.

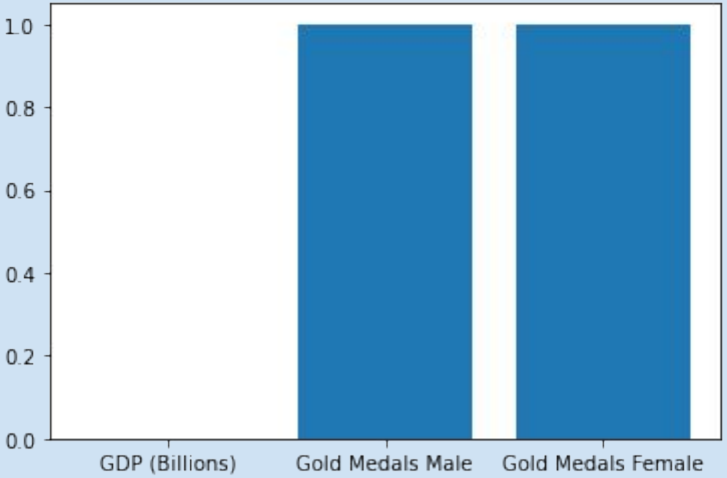
Regression Analysis: Data

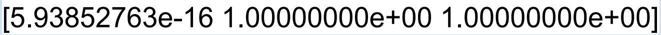






Linear Regression [not successful]

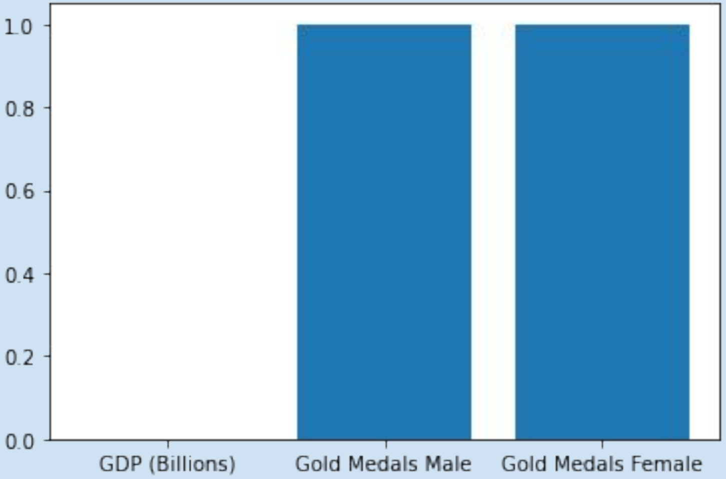




```
[5.93852763e-16 1.00000000e+00 1.00000000e+00]
```

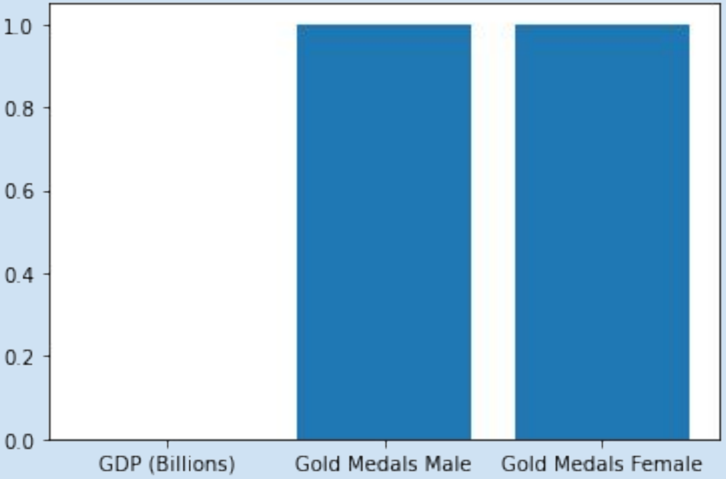
Ridge Regression vs. Lasso Regression

Ridge Regression



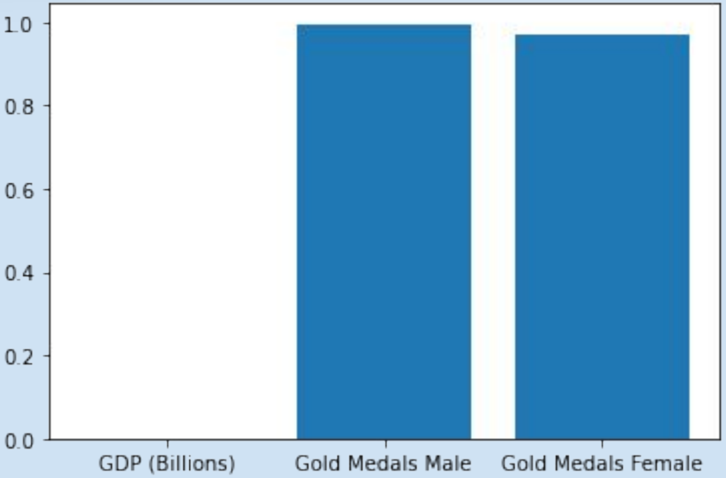
```
[6.11176858e-06 9.99767271e-01 9.98307674e-01]
```

Lasso Regression



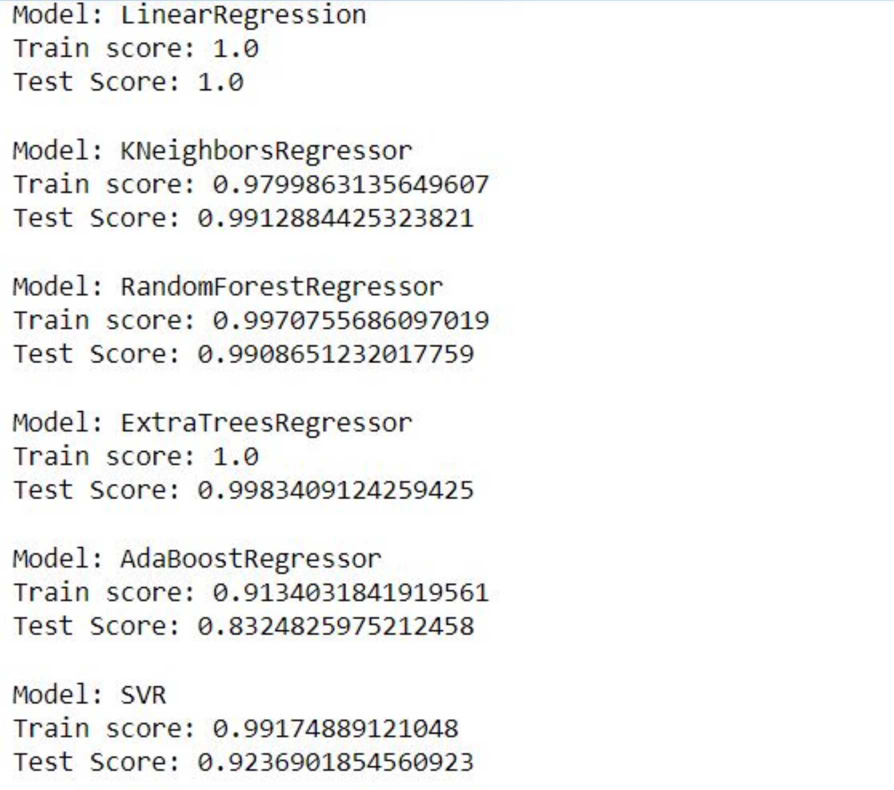
```
[1.09913281e-04 9.95934584e-01 9.69250385e-01]
```

ElasticNet Regression



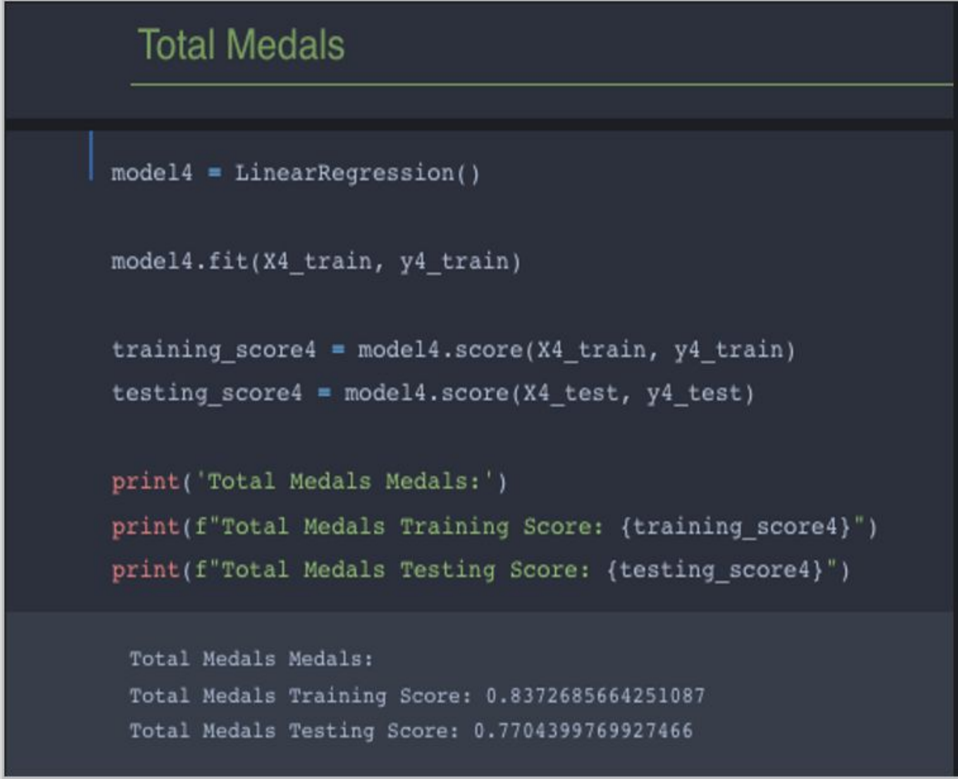
```
[1.08708482e-04 9.95478669e-01 9.70117812e-01]
```

Regression Tests



```
Model: LinearRegression
Train score: 1.0
Test Score: 1.0

Model: KNeighborsRegressor
Train score: 0.9799863135649607
Test Score: 0.9912884425323821

Model: RandomForestRegressor
Train score: 0.9970755686097019
Test Score: 0.9908651232017759

Model: ExtraTreesRegressor
Train score: 1.0
Test Score: 0.9983409124259425

Model: AdaBoostRegressor
Train score: 0.9134031841919561
Test Score: 0.8324825975212458

Model: SVR
Train score: 0.99174889121048
Test Score: 0.9236901854560923
```

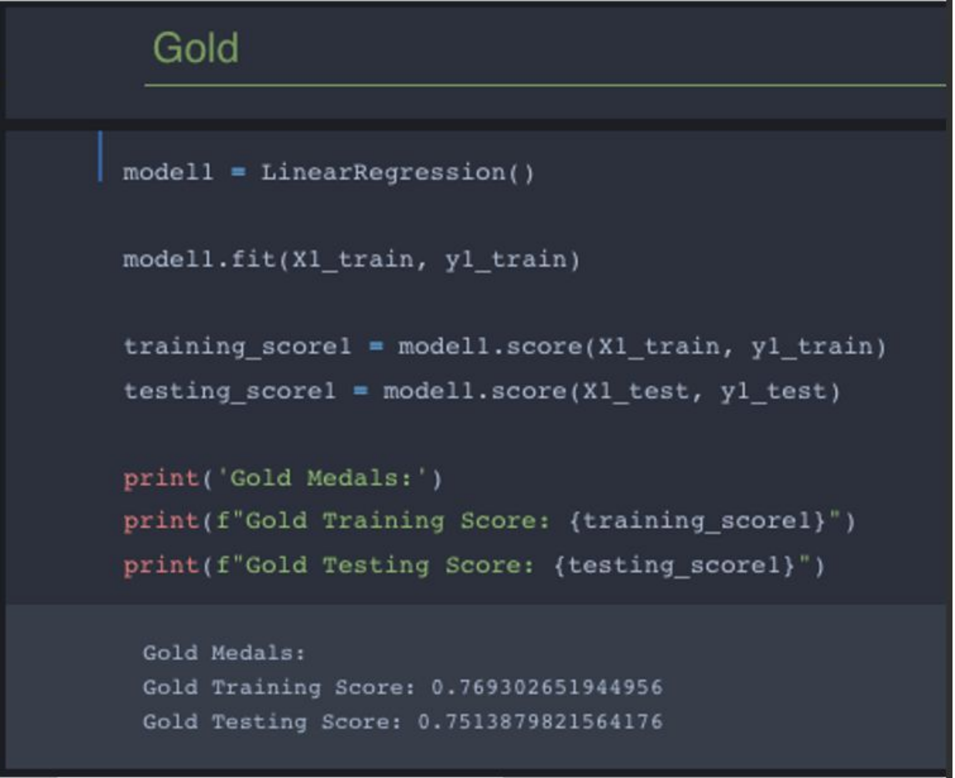
OLYMPIC MEDAL PREDICTIONS



Total Medals

```
model4 = LinearRegression()

model4.fit(X4_train, y4_train)

training_score4 = model4.score(X4_train, y4_train)
testing_score4 = model4.score(X4_test, y4_test)

print('Total Medals Medals:')
print(f"Total Medals Training Score: {training_score4}")
print(f"Total Medals Testing Score: {testing_score4}")
```

```
Total Medals Medals:
Total Medals Training Score: 0.8372685664251087
Total Medals Testing Score: 0.7704399769927466
```

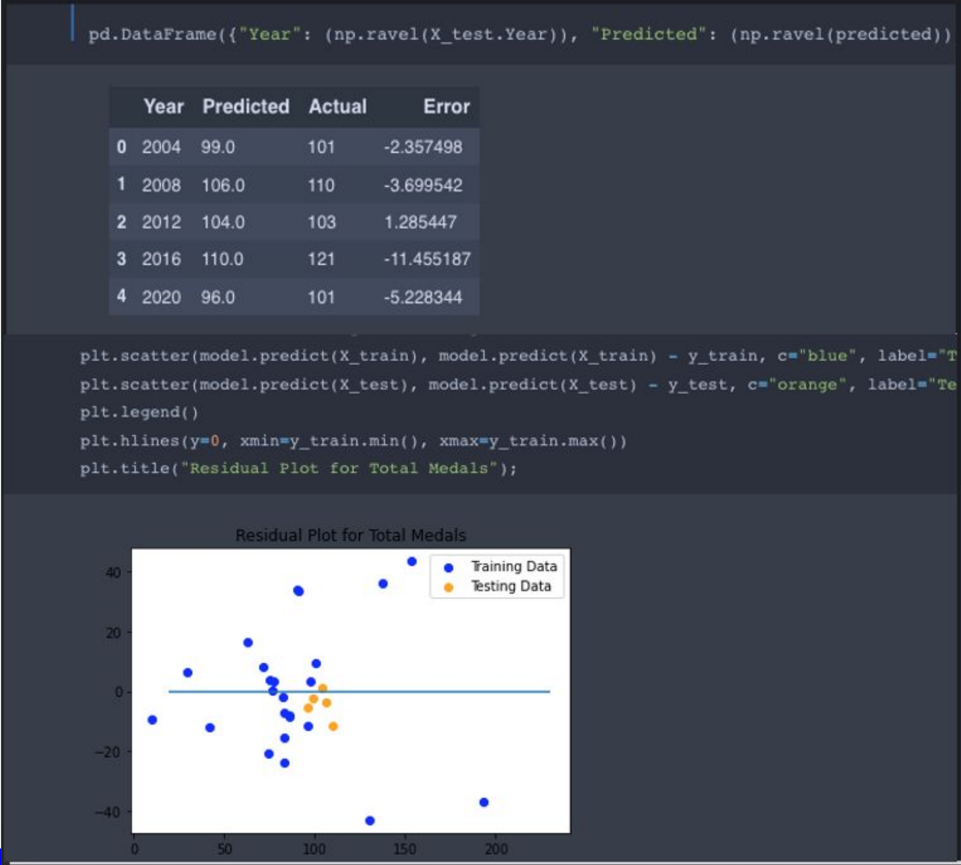


Gold

```
model1 = LinearRegression()

model1.fit(X1_train, y1_train)

training_score1 = model1.score(X1_train, y1_train)
testing_score1 = model1.score(X1_test, y1_test)

print('Gold Medals:')
print(f"Gold Training Score: {training_score1}")
print(f"Gold Testing Score: {testing_score1}")
```

```
Gold Medals:
Gold Training Score: 0.769302651944956
Gold Testing Score: 0.7513879821564176
```

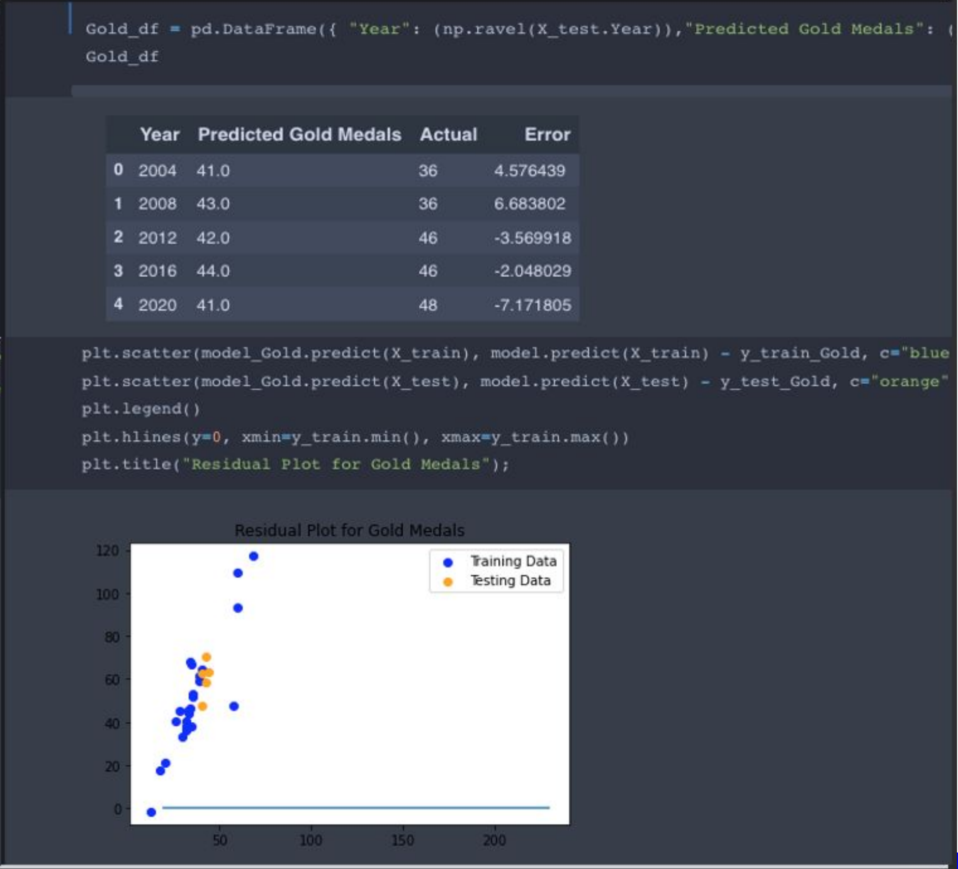
OLYMPIC MEDAL PREDICTIONS



```
pd.DataFrame({"Year": (np.ravel(X_test.Year)), "Predicted": (np.ravel(predicted))
```

	Year	Predicted	Actual	Error
0	2004	99.0	101	-2.357498
1	2008	106.0	110	-3.699542
2	2012	104.0	103	1.285447
3	2016	110.0	121	-11.455187
4	2020	96.0	101	-5.228344

```
plt.scatter(model.predict(X_train), model.predict(X_train) - y_train, c="blue", label="T
plt.scatter(model.predict(X_test), model.predict(X_test) - y_test, c="orange", label="Te
plt.legend()
plt.hlines(y=0, xmin=y_train.min(), xmax=y_train.max())
plt.title("Residual Plot for Total Medals");
```

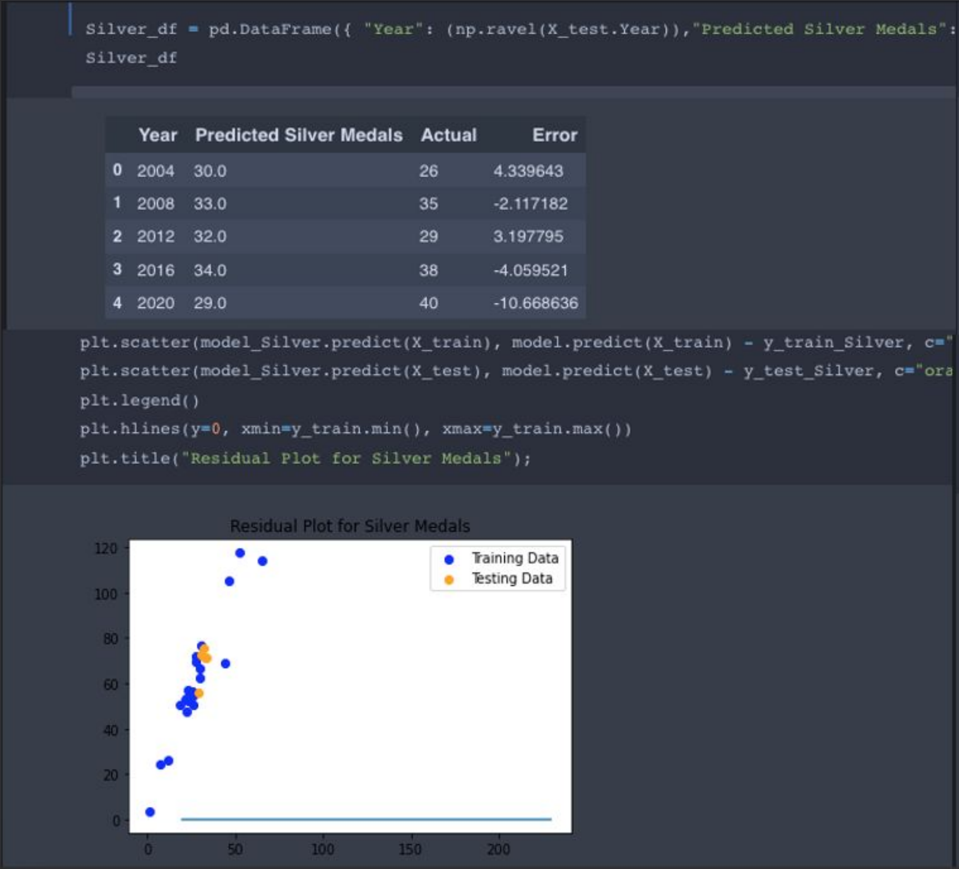


```
Gold_df = pd.DataFrame({ "Year": (np.ravel(X_test.Year)),"Predicted Gold Medals": (
Gold_df
```

	Year	Predicted Gold Medals	Actual	Error
0	2004	41.0	36	4.576439
1	2008	43.0	36	6.683802
2	2012	42.0	46	-3.569918
3	2016	44.0	46	-2.048029
4	2020	41.0	48	-7.171805

```
plt.scatter(model_Gold.predict(X_train), model.predict(X_train) - y_train_Gold, c="blue
plt.scatter(model_Gold.predict(X_test), model.predict(X_test) - y_test_Gold, c="orange"
plt.legend()
plt.hlines(y=0, xmin=y_train.min(), xmax=y_train.max())
plt.title("Residual Plot for Gold Medals");
```

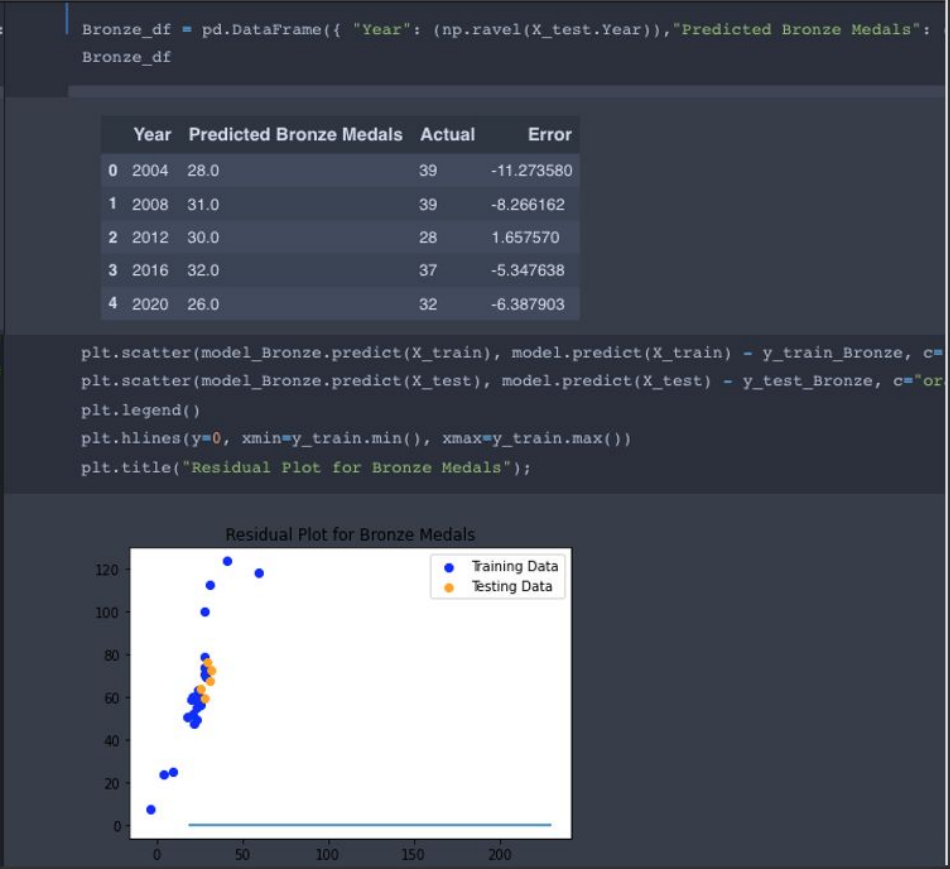
OLYMPIC MEDAL PREDICTIONS



```
Silver_df = pd.DataFrame({ "Year": (np.ravel(X_test.Year)),"Predicted Silver Medals":
Silver_df
```

	Year	Predicted Silver Medals	Actual	Error
0	2004	30.0	26	4.339643
1	2008	33.0	35	-2.117182
2	2012	32.0	29	3.197795
3	2016	34.0	38	-4.059521
4	2020	29.0	40	-10.668636

```
plt.scatter(model_Silver.predict(X_train), model.predict(X_train) - y_train_Silver, c="
plt.scatter(model_Silver.predict(X_test), model.predict(X_test) - y_test_Silver, c="ora
plt.legend()
plt.hlines(y=0, xmin=y_train.min(), xmax=y_train.max())
plt.title("Residual Plot for Silver Medals");
```



```
Bronze_df = pd.DataFrame({ "Year": (np.ravel(X_test.Year)),"Predicted Bronze Medals":
Bronze_df
```

	Year	Predicted Bronze Medals	Actual	Error
0	2004	28.0	39	-11.273580
1	2008	31.0	39	-8.266162
2	2012	30.0	28	1.657570
3	2016	32.0	37	-5.347638
4	2020	26.0	32	-6.387903

```
plt.scatter(model_Bronze.predict(X_train), model.predict(X_train) - y_train_Bronze, c=
plt.scatter(model_Bronze.predict(X_test), model.predict(X_test) - y_test_Bronze, c="or
plt.legend()
plt.hlines(y=0, xmin=y_train.min(), xmax=y_train.max())
plt.title("Residual Plot for Bronze Medals");
```