NAME : SYED RIAZ ALI

EMAIL : syedriazali1997@gmail.com

WHATSAPP : 0300-2502513

1. Simple Linear Regression

In [49]:
```
import pandas as pd

df = pd.read_csv("ml_data_salary.csv")
df
```

Out[49]:

	age	distance	YearsExperience	Salary
0	31.1	77.75	1.1	39343
1	31.3	78.25	1.3	46205
2	31.5	78.75	1.5	37731
3	32.0	80.00	2.0	43525
4	32.2	80.50	2.2	39891
5	32.9	82.25	2.9	56642
6	33.0	82.50	3.0	60150
7	33.2	83.00	3.2	54445
8	33.2	83.00	3.2	64445
9	33.7	84.25	3.7	57189
10	33.9	84.75	3.9	63218
11	34.0	85.00	4.0	55794
12	34.0	85.00	4.0	56957
13	34.1	85.25	4.1	57081
14	34.5	86.25	4.5	61111
15	34.9	87.25	4.9	67938
16	35.1	87.75	5.1	66029
17	35.3	88.25	5.3	83088
18	35.9	89.75	5.9	81363
19	36.0	90.00	6.0	93940
20	36.8	92.00	6.8	91738
21	37.1	92.75	7.1	98273
22	37.9	94.75	7.9	101302
23	38.2	95.50	8.2	113812
24	38.7	96.75	8.7	109431
25	39.0	97.50	9.0	105582
26	39.5	98.75	9.5	116969
27	39.6	99.00	9.6	112635
28	40.3	100.75	10.3	122391
29	40.5	101.25	10.5	121872

In [50]:
```
df = df.drop(["age"], axis = 1)
df.head()
```

Out[50]:

	distance	YearsExperience	Salary
0	77.75	1.1	39343
1	78.25	1.3	46205
2	78.75	1.5	37731
3	80.00	2.0	43525
4	80.50	2.2	39891

In [51]:
```
df = df.drop(["distance"], axis = 1)
df.head()
```

Out[51]:

	YearsExperience	Salary
0	1.1	39343
1	1.3	46205
2	1.5	37731
3	2.0	43525
4	2.2	39891

Step-2 : Splitting dataset into training data and testing data

In [52]:
```
X = df[["YearsExperience"]]
y = df["Salary"]
```

In [53]:
```
X.head()
```

Out[53]:

	YearsExperience
0	1.1
1	1.3
2	1.5
3	2.0
4	2.2

In [54]:
```
y.head()
```

Out[54]:
```
0    39343
1    46205
2    37731
3    43525
4    39891
Name: Salary, dtype: int64
```

In [55]:
```
# import library and split data


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=0)
```

Step-2 : Fit Linear Regression Model

In [56]:
```
from sklearn.linear_model import LinearRegression

model = LinearRegression()
model = model.fit(X_train, y_train)
model
```

Out[56]: `LinearRegression()`

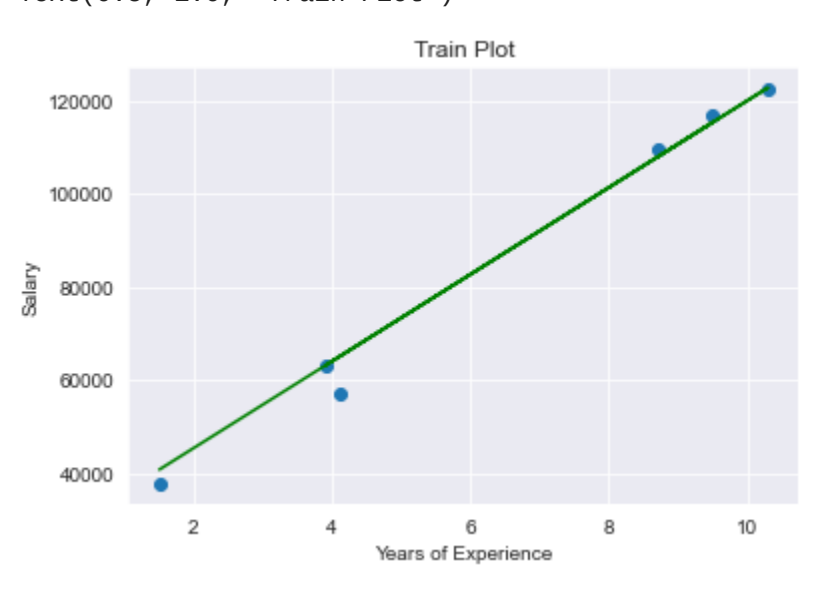
Step-4: Plotting

In [57]:
```
import seaborn as sns
import matplotlib.pyplot as plt

sns.set_style("darkgrid")

plt.scatter(X_train, y_train)
plt.plot(X_train, model.predict(X_train), color ="Green")
plt.xlabel("Years of Experience")
plt.ylabel("Salary")
plt.title("Train Plot")
```

Out[57]: `Text(0.5, 1.0, 'Train Plot')`

In [58]:
```
plt.scatter(X_test, y_test)
plt.plot(X_test, model.predict(X_test), color ="Green")
plt.xlabel("Years of Experience")
plt.ylabel("Salary")
plt.title("Train Plot")
```

Out[58]: `Text(0.5, 1.0, 'Train Plot')`



Step-5 : testing or Evaluating the Model

In [59]:
```
# Model Fitness Checking

model.score(X_test, y_test)
```

Out[59]: `0.988169515729126`

In [60]:
```
model.score(X_train, y_train)
```

Out[60]: `0.9411949620562126`

Step-6 : Prediction of Unknown values

In [61]:
```
model.predict([[5]])
```

```
C:\Users\syedriaz\AppData\Local\Programs\Python\Python39\lib\site-packages\sklearn\base.py:450: UserWarning: X does not have valid feature names, but LinearRegression was fitted with featu
re names
  warnings.warn(
```

Out[61]: `array([73342.97478427])`

In [62]:
```
model.predict([[10]])
```

```
C:\Users\syedriaz\AppData\Local\Programs\Python\Python39\lib\site-packages\sklearn\base.py:450: UserWarning: X does not have valid feature names, but LinearRegression was fitted with featu
re names
  warnings.warn(
```

Out[62]: `array([119905.85041792])`

In [63]:
```
model.predict(X_test)
```

Out[63]:
```
array([ 40748.96184072, 122699.62295594,  64961.65717022,  63099.14214487,
       115249.56285456, 107799.50275317])
```

In [64]:
```
# How to predict value for multiple data

model.predict([[10],[5],[1]])
```

```
C:\Users\syedriaz\AppData\Local\Programs\Python\Python39\lib\site-packages\sklearn\base.py:450: UserWarning: X does not have valid feature names, but LinearRegression was fitted with featu
re names
  warnings.warn(
```

Out[64]: `array([119905.85041792,  73342.97478427,  36092.67427736])`

In [65]:
```
x = ([10],[20],[30],[5])
model.predict(x)
```

```
C:\Users\syedriaz\AppData\Local\Programs\Python\Python39\lib\site-packages\sklearn\base.py:450: UserWarning: X does not have valid feature names, but LinearRegression was fitted with featu
re names
  warnings.warn(
```

Out[65]: `array([119905.85041792, 213031.60168521, 306157.3529525 ,  73342.97478427])`

2. Multiple Linear Regression

In [66]:
```
df = pd.read_csv("ml_data_salary.csv")
df.head()
```

Out[66]:

	age	distance	YearsExperience	Salary
0	31.1	77.75	1.1	39343
1	31.3	78.25	1.3	46205
2	31.5	78.75	1.5	37731
3	32.0	80.00	2.0	43525
4	32.2	80.50	2.2	39891

In [67]:
```
X = df[["age", "distance","YearsExperience"]]
X.head()
```

Out[67]:

	age	distance	YearsExperience
0	31.1	77.75	1.1
1	31.3	78.25	1.3
2	31.5	78.75	1.5
3	32.0	80.00	2.0
4	32.2	80.50	2.2

In [68]:
```
y = df["Salary"]
y.head()
```

Out[68]:
```
0    39343
1    46205
2    37731
3    43525
4    39891
Name: Salary, dtype: int64
```

In [69]:
```
# Create & Fit the model

model = LinearRegression().fit(X, y)
model
```

Out[69]: `LinearRegression()`

In [70]:
```
# Coeffients

model.coef_
```

Out[70]: `array([-2.68055892e+15,  1.06092560e+15,  2.82449143e+13])`

In [71]:
```
model.intercept_
```

Out[71]: `847347429532075.5`

In [72]:
```
model.predict([[31.1,77.75,1.1]])
```

```
C:\Users\syedriaz\AppData\Local\Programs\Python\Python39\lib\site-packages\sklearn\base.py:450: UserWarning: X does not have valid feature names, but LinearRegression was fitted with featu
re names
  warnings.warn(
```

Out[72]: `array([36217.125])`

In [73]:
```
model.score(X, y)
```

Out[73]: `0.9569520722791693`

In [74]:
```
df = pd.read_csv("ml_data_salary.csv")
df.head()
```

Out[74]:

	age	distance	YearsExperience	Salary
0	31.1	77.75	1.1	39343
1	31.3	78.25	1.3	46205
2	31.5	78.75	1.5	37731
3	32.0	80.00	2.0	43525
4	32.2	80.50	2.2	39891

In [75]:
```
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=100)
```

In [76]:
```
from sklearn.linear_model import LinearRegression

model = LinearRegression()
model = model.fit(X_train, y_train)
model
```

Out[76]: `LinearRegression()`

In [77]:
```
X_train.shape
```

Out[77]: `(24, 3)`

In [78]:
```
y_train.shape
```

Out[78]: `(24,)`

In [79]:
```
model.score(X_train, y_train)
```

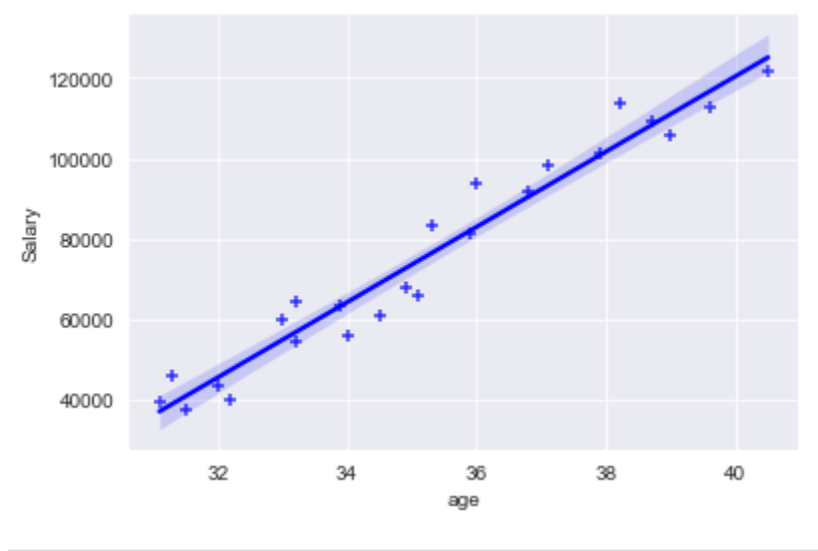
Out[79]: `0.9514648186125461`

In [80]:
```
model.score(X_test, y_test)
```

Out[80]: `0.9726052357502972`

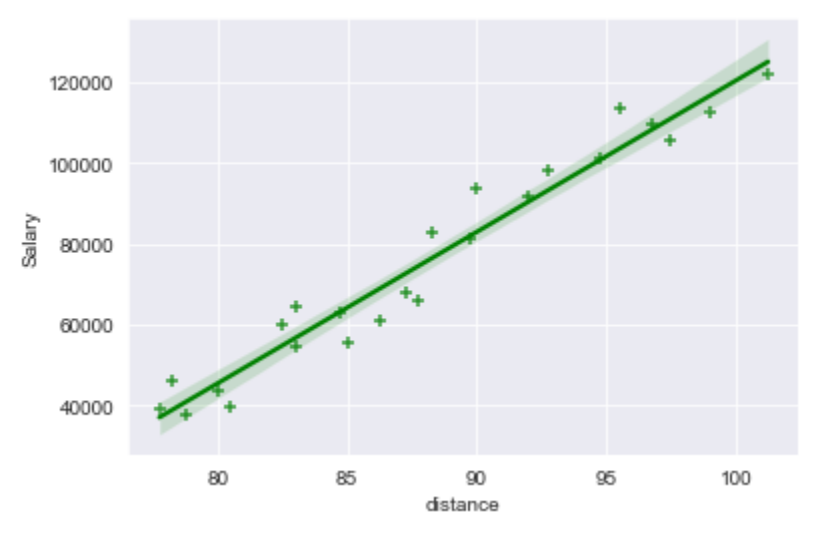
In [81]:
```
sns.regplot(x=X_train["age"], y=y_train, color='blue', marker='+')
```

Out[81]: `<AxesSubplot:xlabel='age', ylabel='Salary'>`



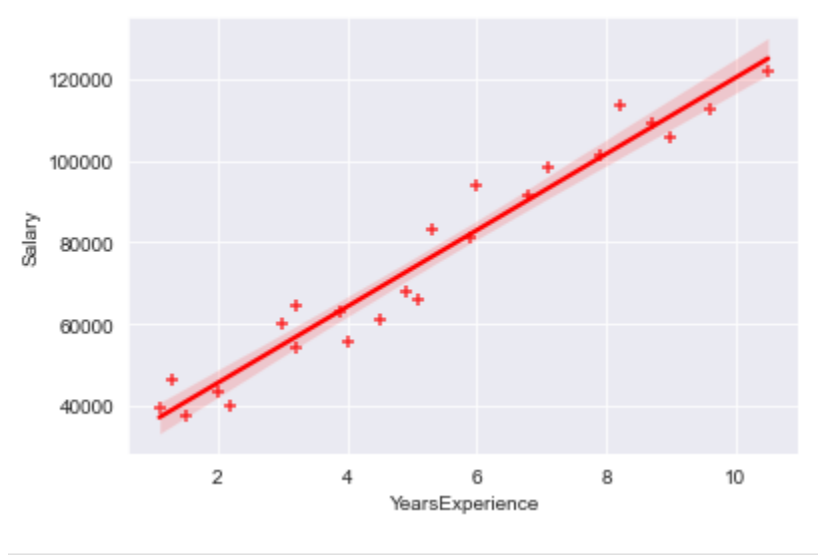
In [82]:
```
sns.regplot(x=X_train["distance"], y=y_train, color='green', marker='+')
```

Out[82]: `<AxesSubplot:xlabel='distance', ylabel='Salary'>`



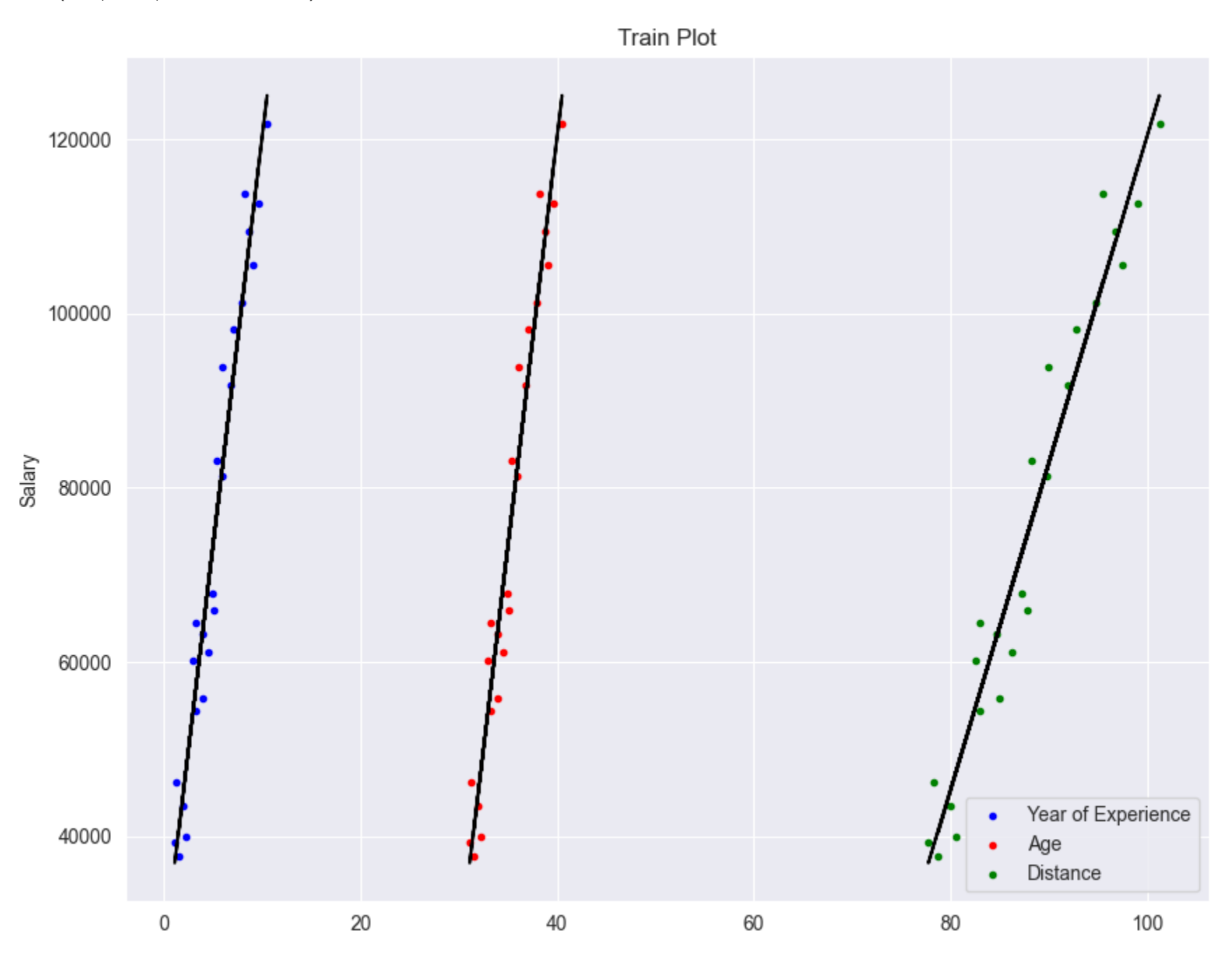
In [83]:
```
sns.regplot(x=X_train["YearsExperience"], y=y_train, color='red', marker='+')
```

Out[83]: `<AxesSubplot:xlabel='YearsExperience', ylabel='Salary'>`

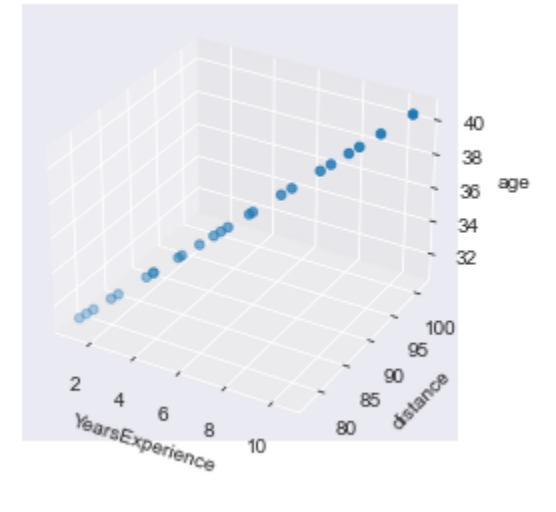


In [95]:
```
plt.rcParams.update({'figure.figsize':(10,8), 'figure.dpi':100})
plt.scatter(X_train["YearsExperience"],y_train,color='blue', marker= '.')
plt.scatter(X_train["age"],y_train,color= 'red', marker='.')
plt.scatter(X_train["distance"],y_train,color= 'green', marker='.')
plt.plot(X_train, model.predict(X_train), color ="Black")
plt.legend(["Year of Experience", "Age","Distance"])
plt.ylabel("Salary")
plt.title("Train Plot")
```

Out[95]: `Text(0.5, 1.0, 'Train Plot')`



In [84]:
```
import matplotlib.pyplot as plt
import pandas as pd
import seaborn as sns
from mpl_toolkits.mplot3d import Axes3D



fig = plt.figure()
ax = fig.add_subplot(111, projection = '3d')



x = X_train["YearsExperience"]
y = X_train["distance"]
z = X_train["age"]

ax.set_xlabel("YearsExperience")
ax.set_ylabel("distance")
ax.set_zlabel("age")

ax.scatter(x, y, z)

plt.show()
```



In [85]:
```
y_predict = model.predict(X_test)
y_predict
```

Out[85]: `array([ 61440., 115776., 123264.,  65152.,  53632.,  64000.])`

In [86]:
```
import matplotlib.pyplot as plt
import numpy as np
from sklearn import datasets, linear_model
from sklearn.metrics import mean_squared_error, r2_score

mean_squared_error(y_test, y_predict, squared = False)
```

Out[86]: `4900.023282257613`

In [87]:
```
r2_score(y_test, y_predict)
```

Out[87]: `0.9726052357502972`

In [ ]: