stl-07

September 15, 2024

1 Question 1:

https://www.kaggle.com/datasets/rohankayan/years-of-experience-and-salary-dataset to an external site.

From the above data: Read the data with pandas and find features and target variables Plot a graph between features and target Find Best fit line using linear regression. FInd MSE, MAE, for test size (20,25)

[7]:

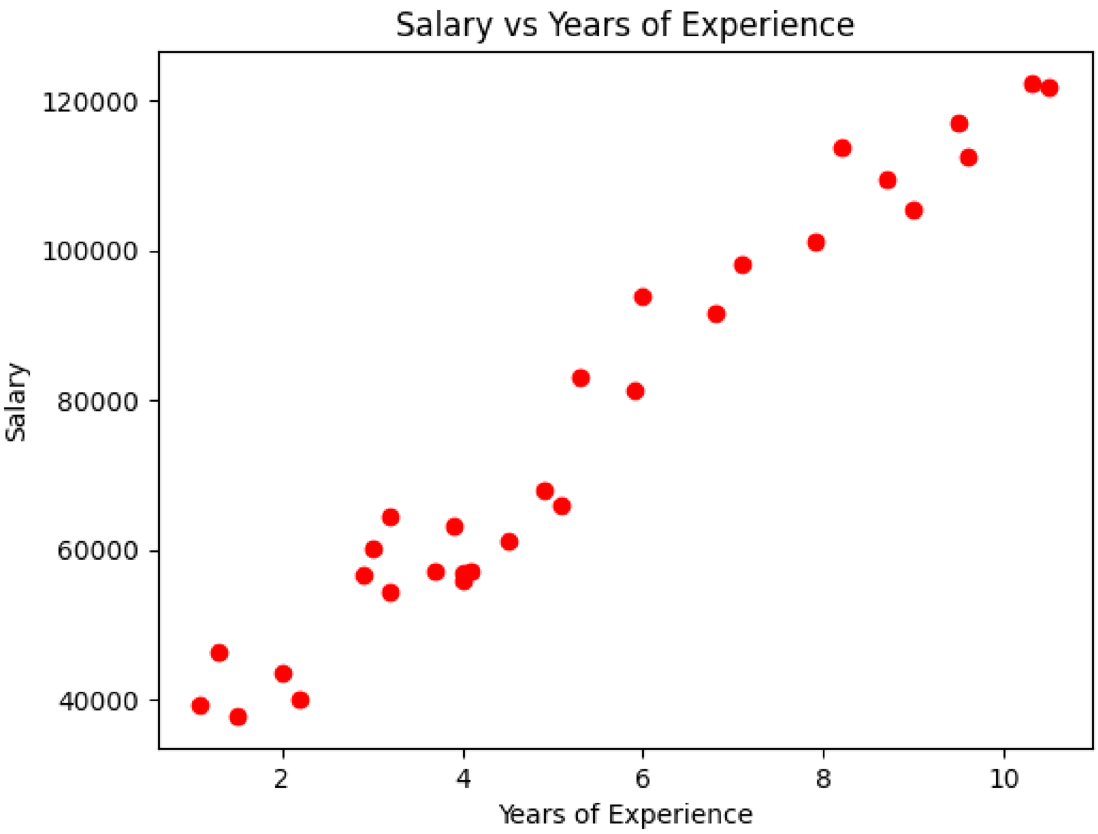
```
import pandas as pd
import matplotlib.pyplot as plt
from sklearn.linear_model import LinearRegression
data = pd.read_csv('/content/Salary_Data.csv')
```

[8]:

```
X = data[['YearsExperience']]
y = data['Salary']
```

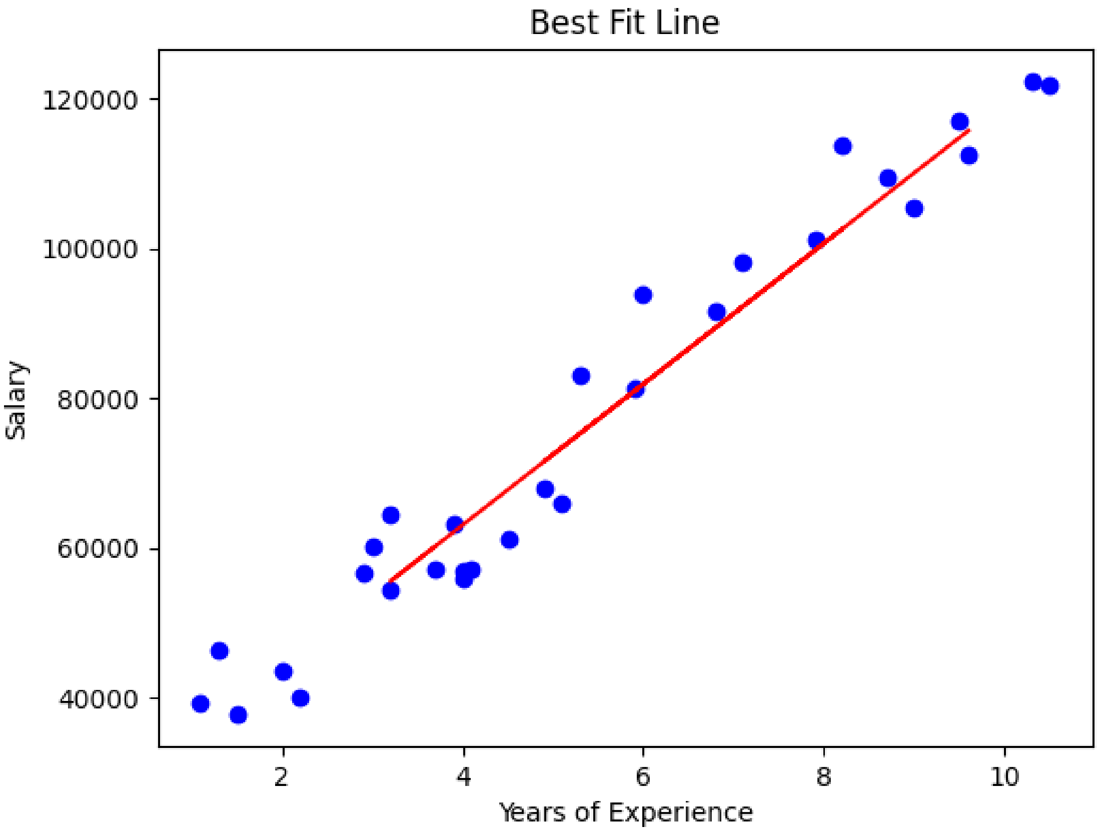
[9]:

```
import matplotlib.pyplot as plt
plt.scatter(X, y, color='Red')
plt.xlabel('Years of Experience')
plt.ylabel('Salary')
plt.title('Salary vs Years of Experience')
plt.show()
```



[10]:

```
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LinearRegression
from sklearn.metrics import mean_squared_error, mean_absolute_error

X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2,␣
  ↪random_state=42)
model = LinearRegression()
model.fit(X_train, y_train)
y_pred = model.predict(X_test)
plt.scatter(X,y,color='blue')
plt.plot(X_test, y_pred, color='red')
plt.xlabel('Years of Experience')
plt.ylabel('Salary')
plt.title('Best Fit Line')
plt.show()
```



[12]:

```
X_train, X_test, y_train, y_test = train_test_split(X,y,test_size=0.
 ↪2,random_state=42)
model.fit(X_train,y_train)
y_pred = model.predict(X_test)
mse_20 = mean_squared_error(y_test,y_pred)
mae_20 = mean_absolute_error(y_test,y_pred)

X_train, X_test,y_train, y_test = train_test_split(X,y,test_size=0.
 ↪25,random_state=42)
model.fit(X_train,y_train)
y_pred = model.predict(X_test)
mse_25 = mean_squared_error(y_test,y_pred)
mae_25 = mean_absolute_error(y_test,y_pred)

print(f'MSE for test size 20%: {mse_20}')
print(f'MAE for test size 20%: {mae_20}')
print(f'MSE for test size 25%: {mse_25}')
print(f'MAE for test size 25%: {mae_25}')
```

```
MSE for test size 20%: 49830096.85590839
```

```
MAE for test size 20%: 6286.453830757749
MSE for test size 25%: 38802588.99247065
MAE for test size 25%: 5056.995466663592
```

2 Questions 2:

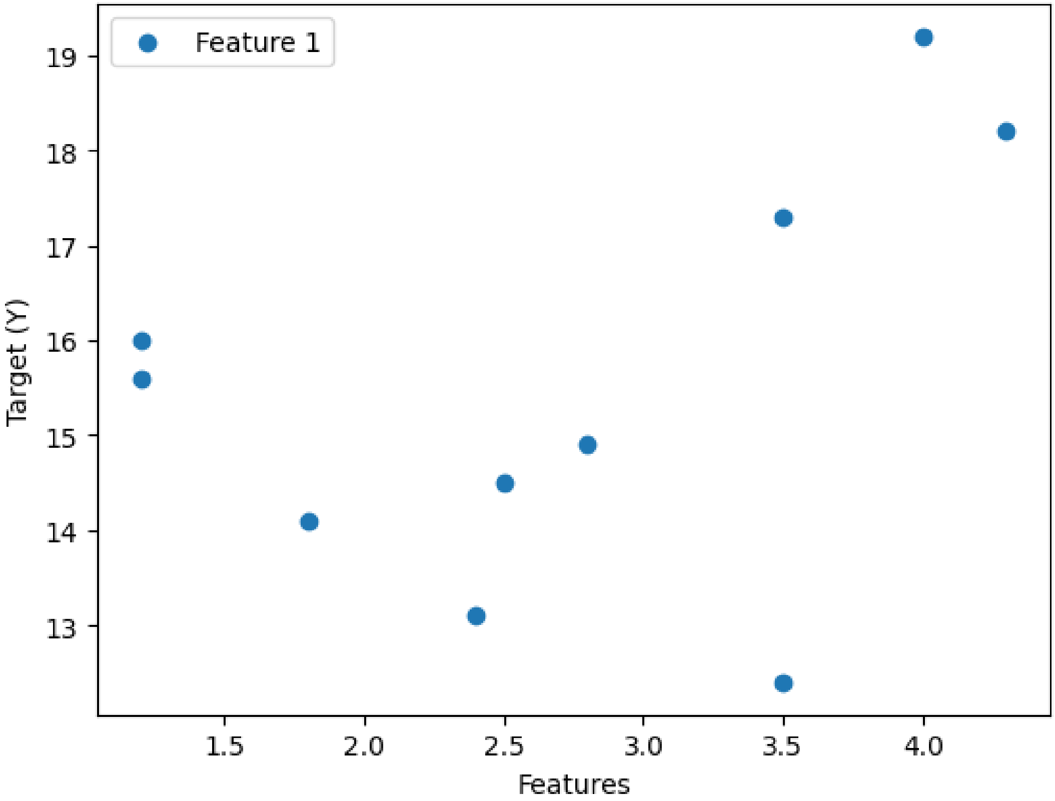
Read the data with pandas and find features and target variables Plot a graph between features and target Find Best fit line using linear regression. Fond MSE , MAE for test size (20,30)

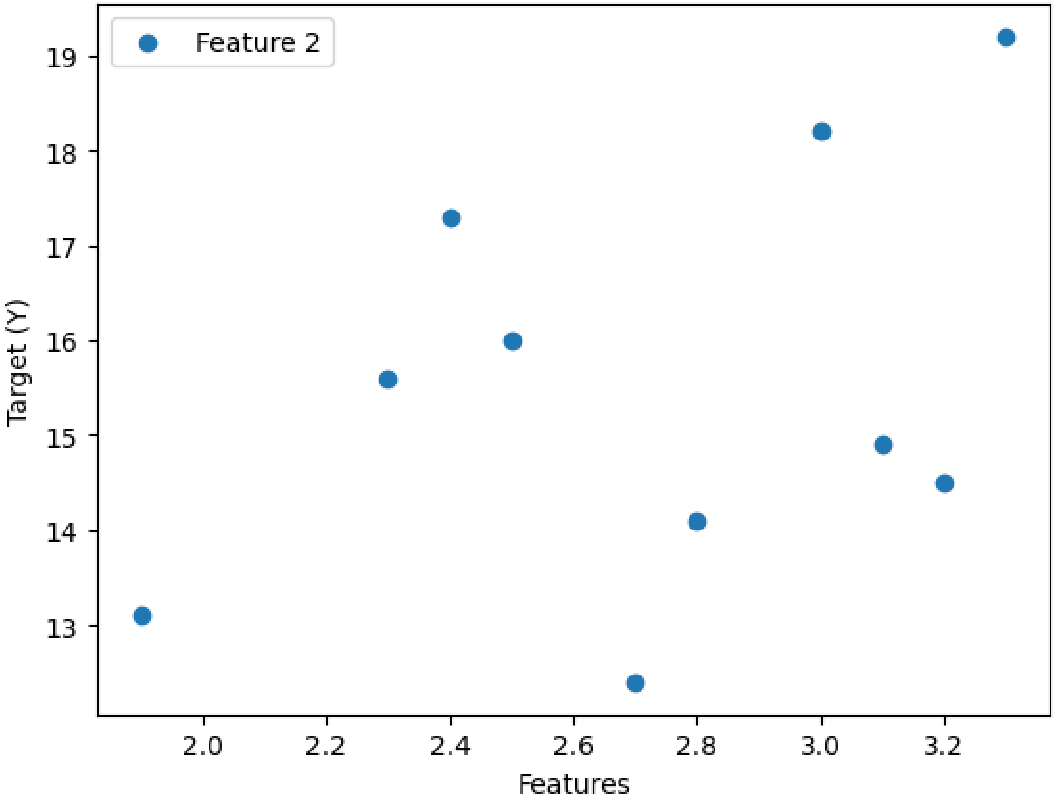
[14]:

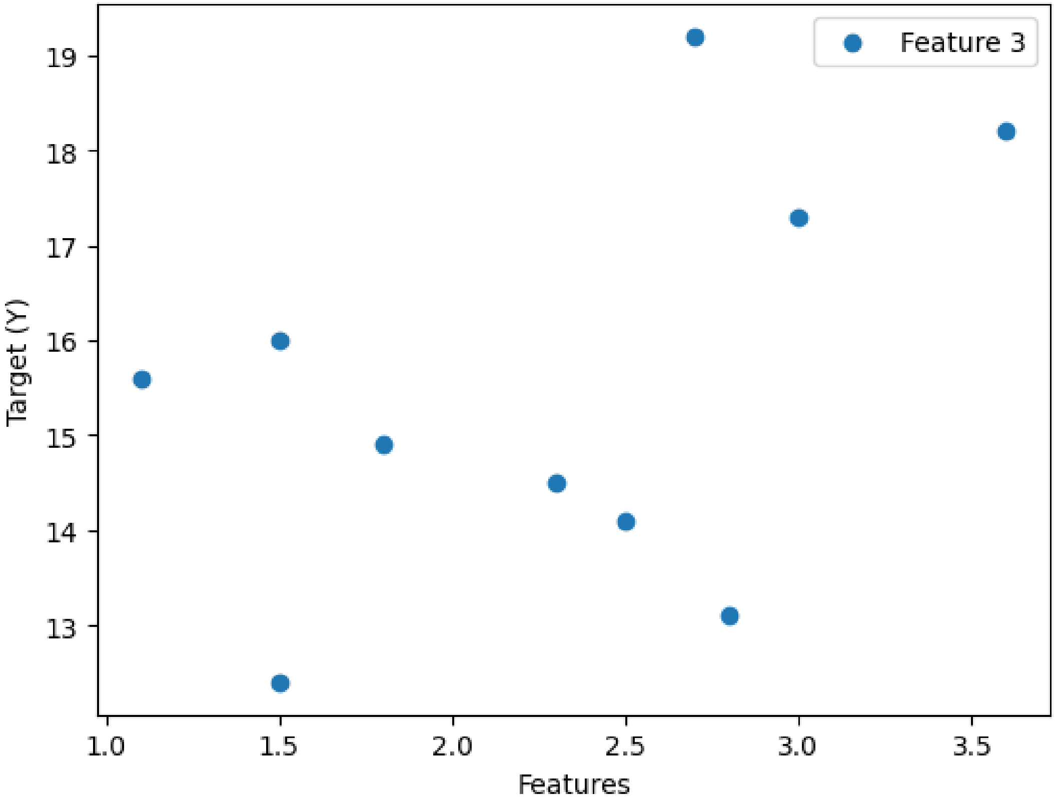
```
import pandas as pd
data = {
    'Feature 1': [1.2, 2.4, 3.5, 4.3, 1.8, 1.2, 2.5, 3.5, 2.8, 4.0],
    'Feature 2': [2.3, 1.9, 2.7, 3.0, 2.8, 2.5, 3.2, 2.4, 3.1, 3.3],
    'Feature 3': [1.1, 2.8, 1.5, 3.6, 2.5, 1.5, 2.3, 3.0, 1.8, 2.7],
    'Feature 4': [4.2, 3.5, 2.9, 4.8, 3.2, 4.0, 4.1, 4.5, 3.6, 4.9],
    'Target (Y)': [15.6, 13.1, 12.4, 18.2, 14.1, 16.0, 14.5, 17.3, 14.9,19.2],
}

df = pd.DataFrame(data)

features  = df[['Feature 1', 'Feature 2', 'Feature 3', 'Feature 4']]
target = df ['Target (Y)']
```

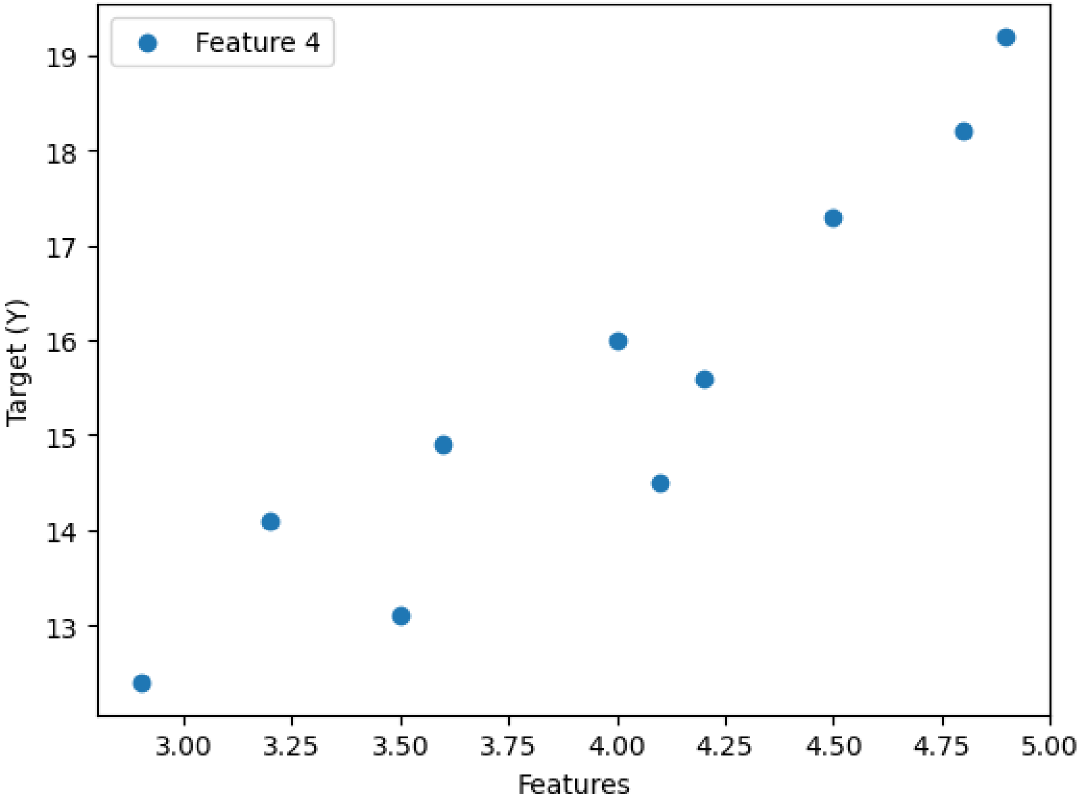
[15]:

```
import matplotlib.pyplot as plt
for column in features.columns:
  plt.scatter(df[column], target, label=column)
  plt.xlabel('Features')
  plt.ylabel('Target (Y)')
  plt.legend()
  plt.show()
```


Feature 1
19
18
17
16
15
14
13
Target (Y)
1.5
2.0
2.5
3.0
3.5
4.0
Features

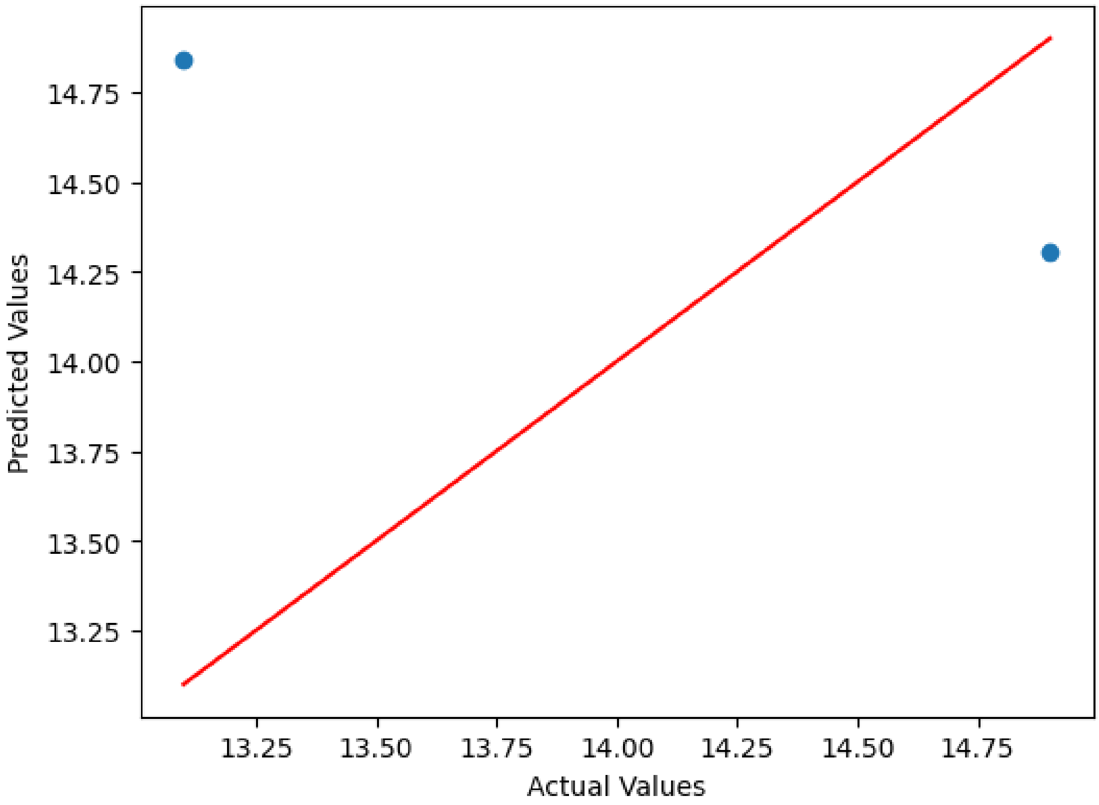

Feature 2
Target (Y)
19
18
17
16
15
14
13
2.0
2.2
2.4
2.6
2.8
3.0
3.2
Features


Feature 3
Target (Y)
19
18
17
16
15
14
13
Features
1.0
1.5
2.0
2.5
3.0
3.5



[16]:

```
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LinearRegression
from sklearn.metrics import mean_squared_error, mean_absolute_error

X_train, X_test, y_train, y_test = train_test_split(features, target,␣
 ↪test_size=0.2, random_state=42)

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

y_pred=model.predict(X_test)
plt.scatter(y_test, y_pred)
plt.xlabel('Actual Values')
plt.ylabel('Predicted Values')
plt.plot([min(y_test), max(y_test)], [min(y_test), max(y_test)], color='red') #␣
 ↪Best fit line
plt.show()
```



```
[23]: def calculate_errors(test_size):
        X_train, X_test, y_train, y_test = train_test_split(features, target,␣
       ↪test_size=test_size,random_state=42)
        model.fit(X_train, y_train)
        y_pred = model.predict(X_test)
        mse = mean_squared_error(y_test, y_pred)
        mae = mean_absolute_error(y_test, y_pred)
        return mse, mae

      mse_20, mae_20 = calculate_errors(0.2)
      print(f'Test Size 20% - MSE: {mse_20}, MAE: {mae_20}')
      mse_30, mae_30  = calculate_errors(0.3)
      print(f'Test Size 30% - MSE: {mse_30}, MAE: {mae_30}')
```

```
Test Size 20% - MSE: 1.686723825329242, MAE: 1.1658259740679746
Test Size 30% - MSE: 1.3045513737291075, MAE: 1.0687852011691732
```