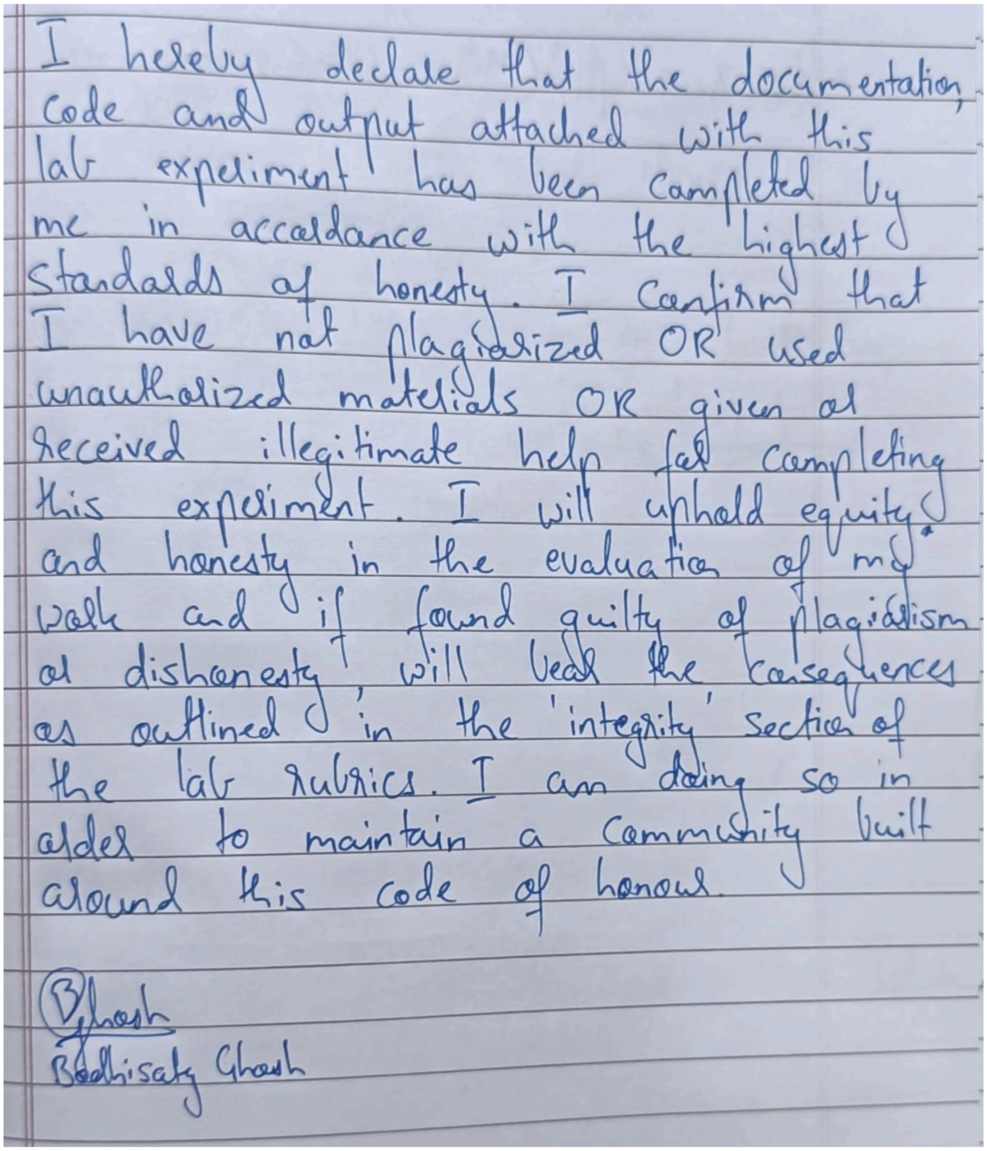
Name	**Bodhisatya Ghosh**
UID no.	**2021700026**

Experiment 8	
HONOUR PLEDGE	 I hereby declare that the documentation, code and output attached with this lab experiment has been completed by me in accordance with the highest standards of honesty. I confirm that I have not plagiarized OR used unauthorized materials OR given or received illegitimate help for completing this experiment. I will uphold equity and honesty in the evaluation of my work and if found guilty of plagiarism or dishonesty, will bear the consequences as outlined in the 'integrity' section of the lab rubrics. I am doing so in order to maintain a community built around this code of honour. B Ghosh Bodhisaty Ghosh
PROBLEM STATEMENT :	**Regression Analysis on Real-time data** 1. Pick up 3 stocks from the S&P500 index (or any other index of interest) and fetch their data from 2010-01-01 to present date into a pandas dataframe

2. Train a regression model using OLS in statsmodels library on 80% of the historic data for each stock, and predict on the recent 20%
3. Print the model summary and explain what do each of the components in the report summary mean
4. Evaluate the fitted model on various statistical metrics for error on 'train' and 'test' 5. Assess the model on metrics that calculate goodness of fit on 'train' and 'test' Add plots of the Actuals, Predictions and Residuals for each of the stocks

THEORY:

What do each term in OLS model summary mean?

The terms are as follows:

• **Dep. Variable (Dependent Variable):** This indicates the variable that is being predicted or explained by the independent variables. In this case, the dependent variable is “Close”.

• **R-squared (uncentered):** R-squared measures the proportion of the variance in the dependent variable that is explained by the independent variables. An R-squared of 1.000 indicates that the model explains all the variance in the dependent variable.

• **Model:** Indicates the type of model used, which is Ordinary Least Squares (OLS) regression.

• **Adj. R-squared (uncentered):** Adjusted R-squared is a modified version of Rsquared that adjusts for the number of predictors in the model. It penalizes the inclusion of unnecessary predictors.

• **Method:** Specifies the method used for fitting the model, which is Least Squares in this case.

• **F-statistic:** The F-statistic tests the overall significance of the regression model. It assesses whether the regression model as a whole is significant.

• **Prob (F-statistic):** This is the p-value associated with the F-statistic. It indicates the probability of observing an F-statistic as extreme as the one calculated, assuming that the null hypothesis is true (i.e., all coefficients are equal to zero). A low p-value suggests that the regression model is significant.

• **Date and Time:** Indicates the date and time when the regression analysis was conducted.

- **No. Observations:** Indicates the number of observations (data points) used in the regression analysis.

- **AIC (Akaike Information Criterion) and BIC (Bayesian Information Criterion):** These are measures of the relative quality of statistical models for a given set of data. Lower values indicate better-fitting models.

- **Df Residuals:** Degrees of freedom of the residuals, which is the difference between the number of observations and the number of predictors in the model.

- **Df Model:** Degrees of freedom of the model, which is the number of predictors (excluding the intercept).

- **Covariance Type:** Specifies the type of covariance estimator used in the regression analysis. In this case, it is “nonrobust.”

- **coef (Coefficients), std err (Standard Error), t (t-value), P>|t| (p-value), [0.025 0.975] (Confidence Intervals):** These columns provide information about the coefficients of the independent variables in the regression model. The coefficients represent the estimated effect of each independent variable on the dependent variable. The standard error indicates the precision of the coefficient estimate. The t-value is the coefficient divided by its standard error and measures the significance of the coefficient. The p-value indicates the probability of observing a t-value as extreme as the one calculated, assuming that the null hypothesis is true (i.e., the coefficient is equal to zero). The confidence intervals provide a range within which the true coefficient is likely to lie.

- **Omnibus:** Tests the overall significance of the model’s residuals. A low p-value suggests that the residuals are not normally distributed.

- **Durbin-Watson:** Tests for autocorrelation in the residuals. Values close to 2 indicate no autocorrelation.

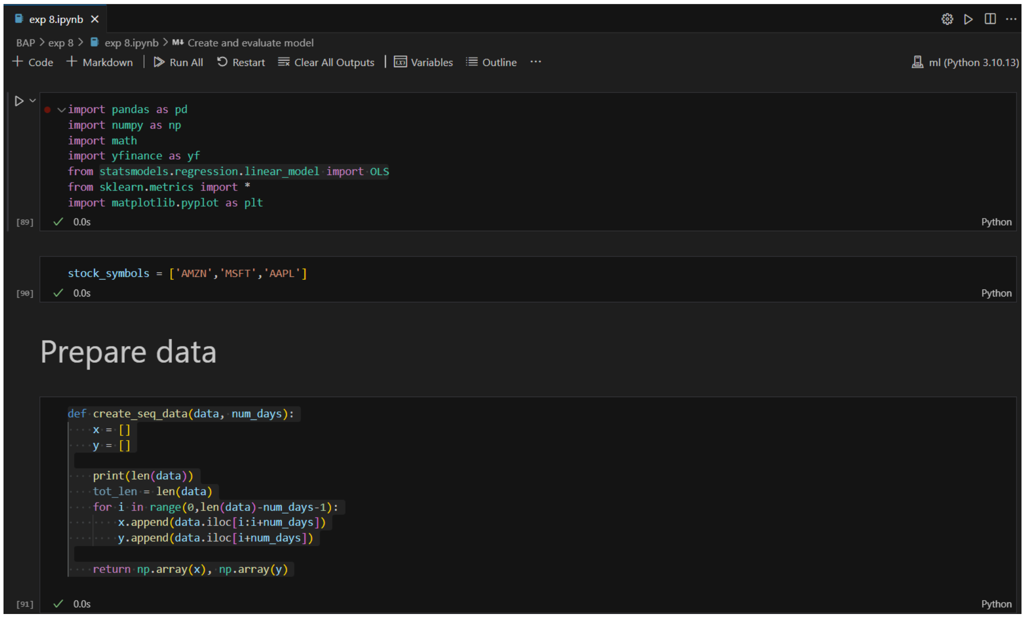
- **Jarque-Bera (JB):** Tests the null hypothesis that the residuals are normally distributed. A low p-value suggests that the residuals are not normally distributed.

- **Skew:** Measures the symmetry of the residuals. A value close to zero indicates that the residuals are symmetrically distributed.

- **Kurtosis:** Measures the peaked-ness of the residuals. A value of 3 indicates a normal distribution.

- **Cond. No. (Condition Number):** Measures multicollinearity in the regression model. Values above 30 suggest multicollinearity.

RESULT:

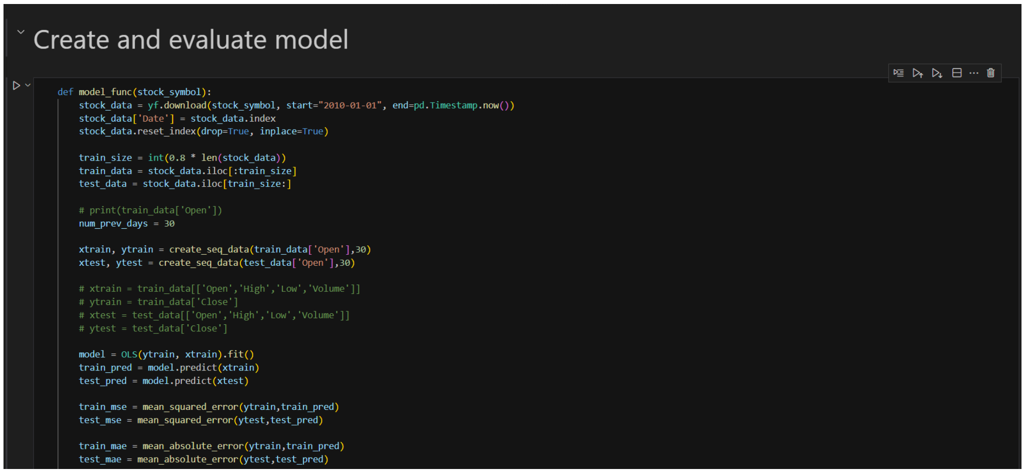


```python
import pandas as pd
import numpy as np
import math
import yfinance as yf
from statsmodels.regression.linear_model import OLS
from sklearn.metrics import *
import matplotlib.pyplot as plt
```

```python
stock_symbols = ['AMZN','MSFT','AAPL']
```

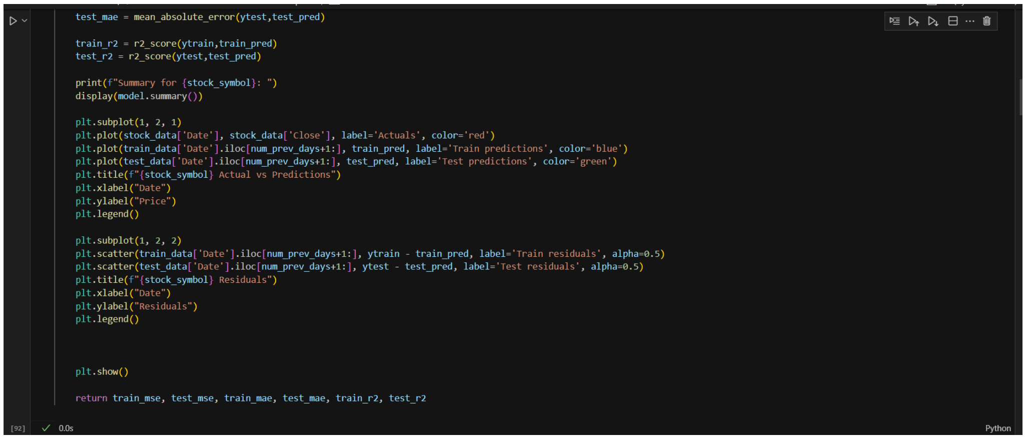
Prepare data

```python
def create_seq_data(data, num_days):
    x = []
    y = []

    print(len(data))
    tot_len = len(data)
    for i in range(0,len(data)-num_days-1):
        x.append(data.iloc[i:i+num_days])
        y.append(data.iloc[i+num_days])

    return np.array(x), np.array(y)
```

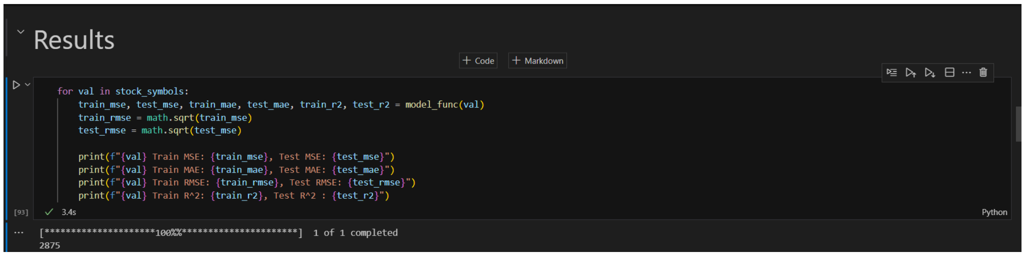


Create and evaluate model

```python
def model_func(stock_symbol):
    stock_data = yf.download(stock_symbol, start="2010-01-01", end=pd.Timestamp.now())
    stock_data['Date'] = stock_data.index
    stock_data.reset_index(drop=True, inplace=True)

    train_size = int(0.8 * len(stock_data))
    train_data = stock_data.iloc[:train_size]
    test_data = stock_data.iloc[train_size:]

    # print(train_data['Open'])
    num_prev_days = 30

    xtrain, ytrain = create_seq_data(train_data['Open'],30)
    xtest, ytest = create_seq_data(test_data['Open'],30)

    # xtrain = train_data[['Open','High','Low','Volume']]
    # ytrain = train_data['Close']
    # xtest = test_data[['Open','High','Low','Volume']]
    # ytest = test_data['Close']

    model = OLS(ytrain, xtrain).fit()
    train_pred = model.predict(xtrain)
    test_pred = model.predict(xtest)

    train_mse = mean_squared_error(ytrain,train_pred)
    test_mse = mean_squared_error(ytest,test_pred)

    train_mae = mean_absolute_error(ytrain,train_pred)
    test_mae = mean_absolute_error(ytest,test_pred)
```



```python
    test_mae = mean_absolute_error(ytest,test_pred)

    train_r2 = r2_score(ytrain,train_pred)
    test_r2 = r2_score(ytest,test_pred)

    print(f"Summary for {stock_symbol}: ")
    display(model.summary())

    plt.subplot(1, 2, 1)
    plt.plot(stock_data['Date'], stock_data['Close'], label='Actuals', color='red')
    plt.plot(train_data['Date'].iloc[num_prev_days+1:], train_pred, label='Train predictions', color='blue')
    plt.plot(test_data['Date'].iloc[num_prev_days+1:], test_pred, label='Test predictions', color='green')
    plt.title(f"{stock_symbol} Actual vs Predictions")
    plt.xlabel("Date")
    plt.ylabel("Price")
    plt.legend()

    plt.subplot(1, 2, 2)
    plt.scatter(train_data['Date'].iloc[num_prev_days+1:], ytrain - train_pred, label='Train residuals', alpha=0.5)
    plt.scatter(test_data['Date'].iloc[num_prev_days+1:], ytest - test_pred, label='Test residuals', alpha=0.5)
    plt.title(f"{stock_symbol} Residuals")
    plt.xlabel("Date")
    plt.ylabel("Residuals")
    plt.legend()



    plt.show()

    return train_mse, test_mse, train_mae, test_mae, train_r2, test_r2
```

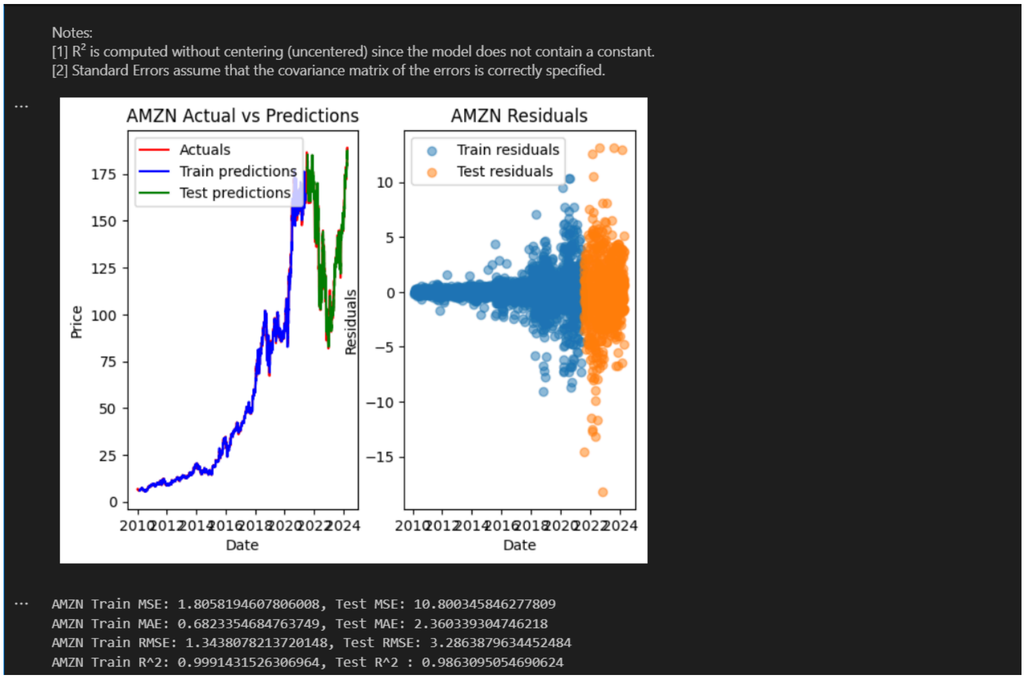


Results

```
for val in stock_symbols:
    train_mse, test_mse, train_mae, test_mae, train_r2, test_r2 = model_func(val)
    train_rmse = math.sqrt(train_mse)
    test_rmse = math.sqrt(test_mse)

    print(f"{val} Train MSE: {train_mse}, Test MSE: {test_mse}")
    print(f"{val} Train MAE: {train_mae}, Test MAE: {test_mae}")
    print(f"{val} Train RMSE: {train_rmse}, Test RMSE: {test_rmse}")
    print(f"{val} Train R^2: {train_r2}, Test R^2 : {test_r2}")
```

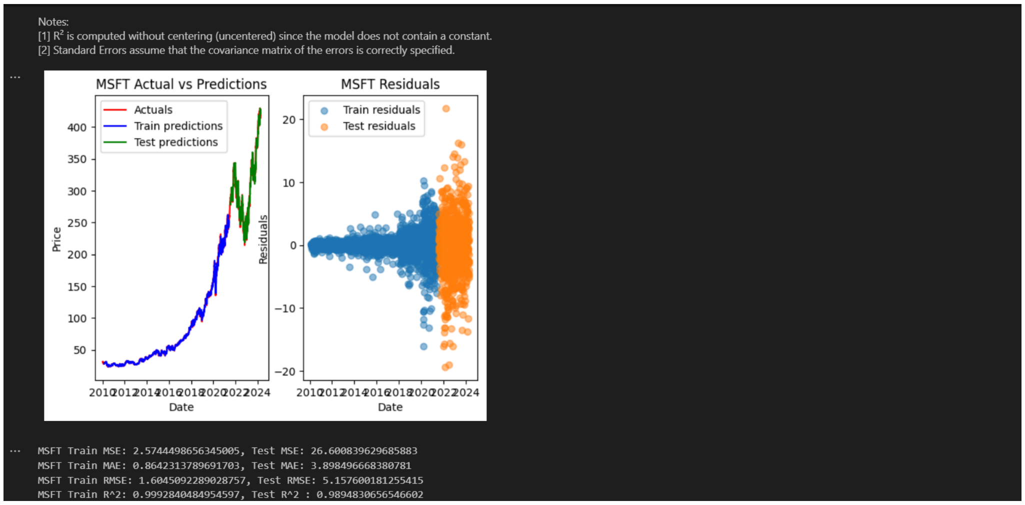
[93] ✓ 3.4s

```
[*********************100%%**********************]  1 of 1 completed
2875
```



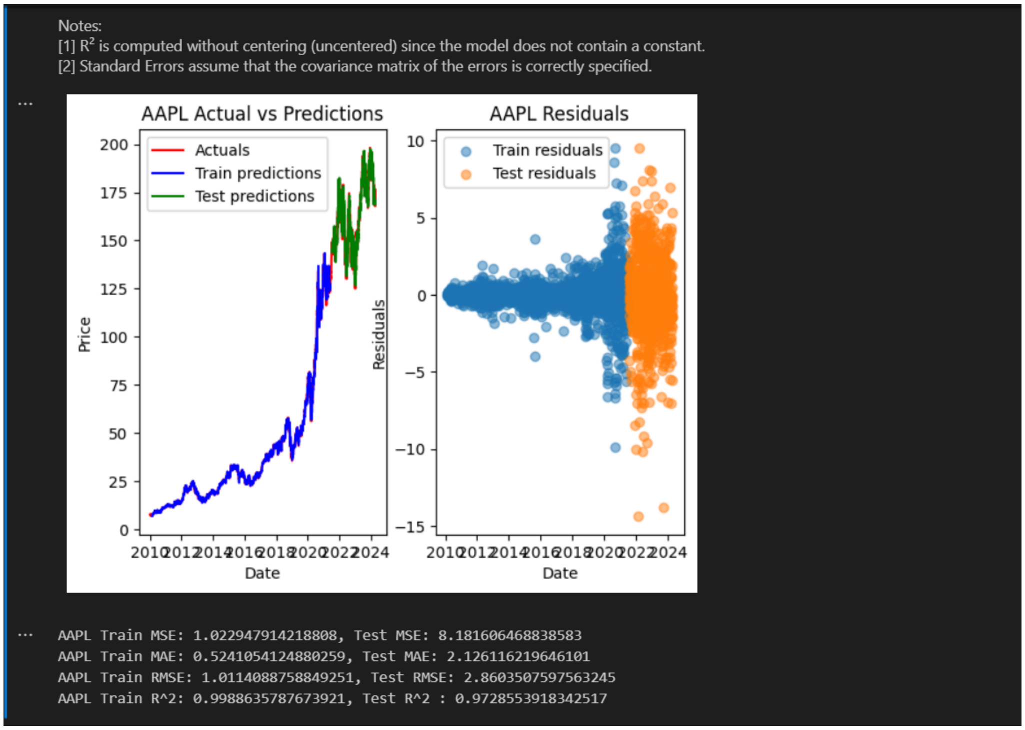
Notes:
[1] R^2 is computed without centering (uncentered) since the model does not contain a constant.
[2] Standard Errors assume that the covariance matrix of the errors is correctly specified.

```
AMZN Train MSE: 1.8058194607806008, Test MSE: 10.800345846277809
AMZN Train MAE: 0.6823354684763749, Test MAE: 2.360339304746218
AMZN Train RMSE: 1.3438078213720148, Test RMSE: 3.2863879634452484
AMZN Train R^2: 0.9991431526306964, Test R^2 : 0.9863095054690624
```



Notes:
[1] R^2 is computed without centering (uncentered) since the model does not contain a constant.
[2] Standard Errors assume that the covariance matrix of the errors is correctly specified.

```
MSFT Train MSE: 2.5744498656345005, Test MSE: 26.600839629685883
MSFT Train MAE: 0.8642313789691703, Test MAE: 3.898496668380781
MSFT Train RMSE: 1.6045092289028757, Test RMSE: 5.157600181255415
MSFT Train R^2: 0.9992840484954597, Test R^2 : 0.9894830656546602
```

<table>
<tr><td></td><td>
Notes:
[1] R^2 is computed without centering (uncentered) since the model does not contain a constant.
[2] Standard Errors assume that the covariance matrix of the errors is correctly specified.
...

...
AAPL Train MSE: 1.022947914218808, Test MSE: 8.181606468838583
AAPL Train MAE: 0.5241054124880259, Test MAE: 2.126116219646101
AAPL Train RMSE: 1.0114088758849251, Test RMSE: 2.8603507597563245
AAPL Train R^2: 0.9988635787673921, Test R^2 : 0.9728553918342517</td></tr>
<tr><td><b>REFERENCES:</b></td><td>JPCSJ17341058.pdf (iop.org)
statsmodels.regression.linear_model.OLS - statsmodels 0.15.0 (+251)</td></tr>
<tr><td colspan="2"><b>CONCLUSION:</b> Studied about statsmodel library and OLS method for model training and prediction.</td></tr>
</table>