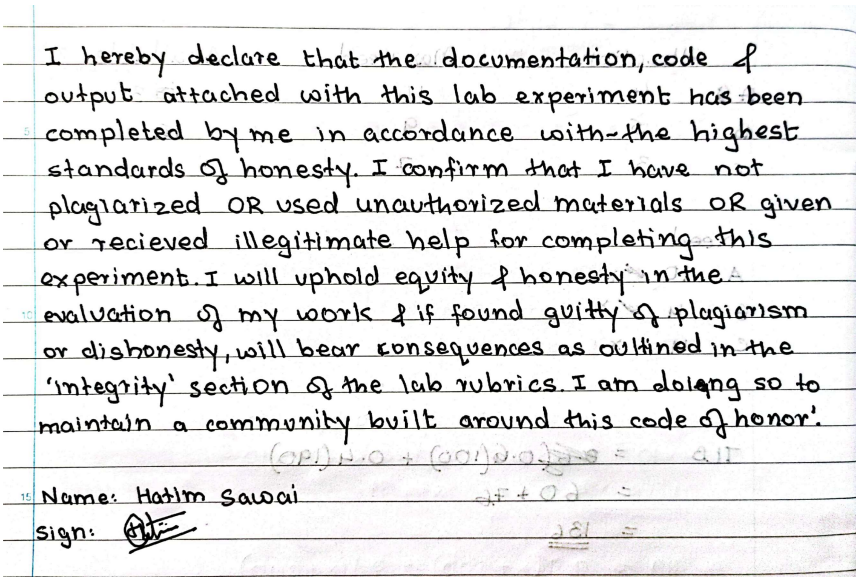
| Name | Hatim Sawai |
|---|---|
| UID No. | 2021300108 |

# Experiment 8

| | |
|---|---|
| HONOR PLEDGE |   |
| PROBLEM STATEMENT | **Regression Analysis on Real-time data:**<br>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<br>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%<br>3. Print the model summary and explain what do each of the components in the report summary mean<br>4. Evaluate the fitted model on various statistical metrics for error on 'train' and 'test'<br>5. Assess the model on metrics that calculate goodness of fit on 'train' and 'test' |
| THEORY | **1.Fetching Stock data:**<br>Stock data is essential for conducting regression analysis in finance. It typically includes attributes such as Open, High, Low, Close prices, and trading volume. Fetching historical stock data allows us to analyze price movements over time and build predictive models.<br>In this task, we select three stocks from a stock index (e.g., S&P 500) and retrieve their historical data using a financial data provider like Yahoo Finance. We store the data in a pandas DataFrame for further analysis. |

**2. Training a regression model using OLS:**
Ordinary Least Squares (OLS) regression is a statistical method used to estimate the relationship between one or more independent variables (features) and a dependent variable (target). It minimizes the sum of squared differences between observed and predicted values.
We split the historical data into training and testing sets. Then, we train an OLS regression model using the training data, where the independent variables are features like Open, High, Low prices, and trading volume, and the dependent variable is the Close price.

**3. Model Summarization:**
The model summary provides detailed information about the regression model's coefficients, statistical significance, goodness of fit, and other relevant statistics. It helps interpret the model's effectiveness and identify potential issues.
We print the summary of the trained regression model using the `summary()` method provided by the statsmodels library**.** The summary includes information such as coefficient estimates, standard errors, t-statistics, p-values, and goodness-of-fit measures like R-squared.

**4. Evaluating model on statistical metrics:**
Evaluating the model involves assessing its performance using statistical metrics to measure the accuracy and reliability of predictions. Common metrics include Root Mean Squared Error (RMSE) and R-squared (R2).
We calculate RMSE and R2 scores for both the training and testing sets. RMSE measures the average deviation of predicted values from actual values, while R2 measures the proportion of variance in the target variable explained by the independent variables. Lower RMSE and higher R2 values indicate better model performance.

**5. Goodness of fit metrics:**
Goodness of fit metrics assess how well the regression model fits the observed data. They quantify the model's ability to capture patterns and variability in the data.
We interpret the RMSE and R2 scores calculated in the previous step as goodness-of-fit metrics. RMSE quantifies the model's prediction error, while R2 indicates the proportion of variance explained by the model. Higher R2 and lower RMSE values suggest a better fit between the model and the data.

## 1. Importing Libraries & Dataset

In [ ]:
```
import pandas as pd
import numpy as np
import yfinance as yf
import statsmodels.api as sm
from sklearn.model_selection import train_test_split
from sklearn.metrics import mean_squared_error, r2_score
import matplotlib.pyplot as plt
```

## 2. Fetching Data

In [ ]:
```
def fetch_stock_data(ticker, start_date, end_date):
    stock_data = yf.download(ticker, start=start_date, end=end_date)
    return stock_data
```

## 3. Training Regression Model using OLS

In [ ]:
```
def train_and_predict_regression(data):
    X = data[["Open", "High", "Low", "Volume"]]
    y = data["Close"]
    X_train, X_test, y_train, y_test = train_test_split(
        X, y, test_size=0.2, shuffle=False
    )

    X_train = sm.add_constant(X_train)
    model = sm.OLS(y_train, X_train).fit()

    X_test = sm.add_constant(X_test)
    y_pred_train = model.predict(X_train)
    y_pred_test = model.predict(X_test)

    return model, y_pred_train, y_pred_test, y_train, y_test
```

## 4. Model Evaluation

In [ ]:
```
def evaluate_model(y_train, y_test, y_pred_train, y_pred_test):
    train_rmse = np.sqrt(mean_squared_error(y_train, y_pred_train))
    test_rmse = np.sqrt(mean_squared_error(y_test, y_pred_test))
    train_r2 = r2_score(y_train, y_pred_train)
    test_r2 = r2_score(y_test, y_pred_test)

    print(f"Train RMSE: {train_rmse}")
    print(f"Test RMSE: {test_rmse}")
    print(f"Train R-squared: {train_r2}")
    print(f"Test R-squared: {test_r2}")

    return train_rmse, test_rmse, train_r2, test_r2
```

## Plots & Visualizations

In [ ]:
```
def plot_results(ticker, y_train, y_test, y_pred_train, y_pred_test):
    plt.figure(figsize=(12, 6))
    plt.plot(y_train.index, y_train, label="Train")
    plt.plot(y_test.index, y_test, label="Test")
    plt.plot(y_train.index, y_pred_train, label="Train Prediction")
    plt.plot(y_test.index, y_pred_test, label="Test Prediction")
    plt.title(f"{ticker} Stock Price Prediction")
    plt.xlabel("Date")
    plt.ylabel("Price")
    plt.legend()
    plt.show()

    # Plot the residuals
    residuals_train = y_train - y_pred_train
    residuals_test = y_test - y_pred_test
    plt.figure(figsize=(12, 6))
    plt.plot(y_train.index, residuals_train, label="Train Residuals")
    plt.plot(y_test.index, residuals_test, label="Test Residuals")
    plt.title(f"{ticker} Stock Price Prediction Residuals")
    plt.xlabel("Date")
    plt.ylabel("Price")
    plt.legend()
    plt.show()
```

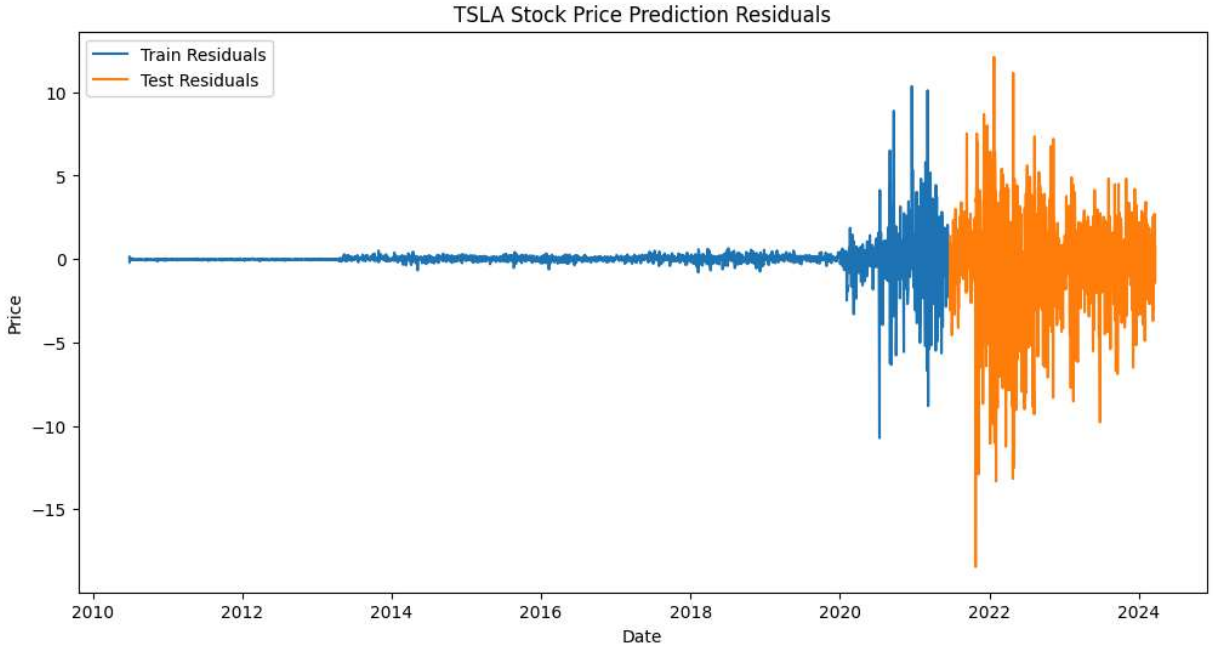
## 3. Output

In [ ]:
```
start_date = "2010-01-01"
end_date = "2024-03-25"
tickers = ["TSLA", "NFLX", "GOOG"]

stock_data = {}
for ticker in tickers:
    stock_data[ticker] = fetch_stock_data(ticker, start_date, end_date)

# Train regression model, print summary, evaluate, and plot results
for ticker, data in stock_data.items():
    model, y_pred_train, y_pred_test, y_train, y_test = train_and_predict_regressio
        data
    )
    print(f"\n\n*** {ticker} ***")
    print(model.summary())
    train_rmse, test_rmse, train_r2, test_r2 = evaluate_model(
        y_train, y_test, y_pred_train, y_pred_test
    )
    plot_results(ticker, y_train, y_test, y_pred_train, y_pred_test)
```

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

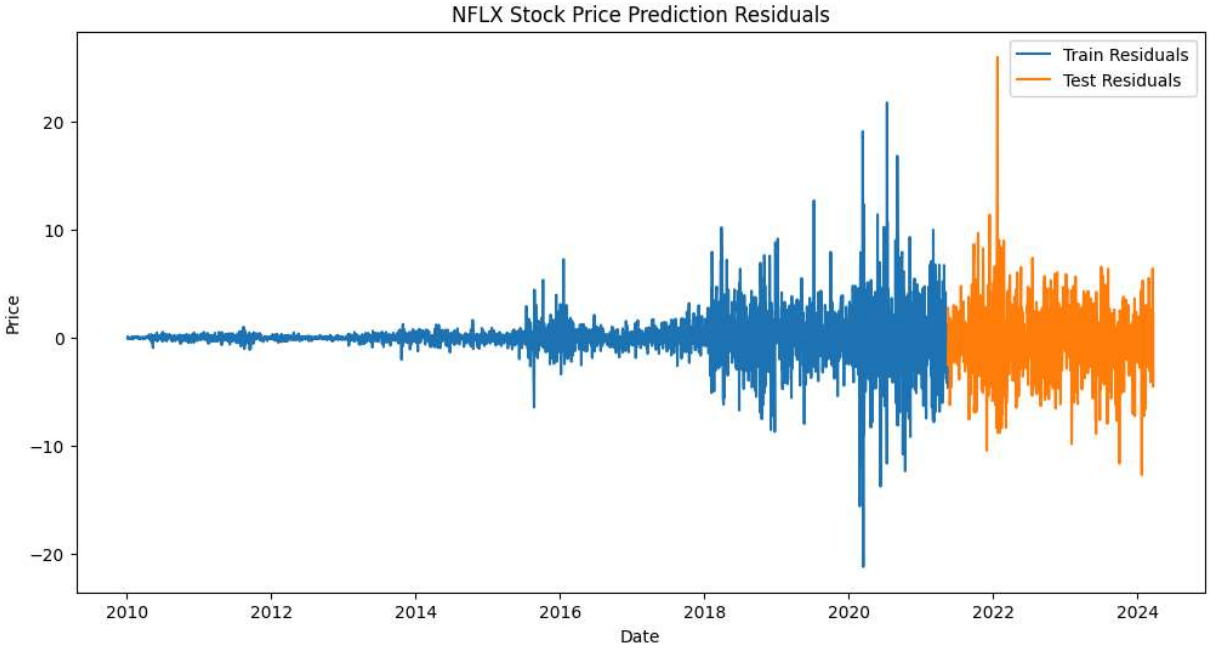
```
*** TSLA ***
                            OLS Regression Results
==============================================================================
Dep. Variable:                  Close   R-squared:                       1.000
Model:                            OLS   Adj. R-squared:                  1.000
Method:                 Least Squares   F-statistic:                 2.701e+06
Date:                Fri, 29 Mar 2024   Prob (F-statistic):               0.00
Time:                        14:02:16   Log-Likelihood:                -3479.1
No. Observations:                2765   AIC:                             6968.
Df Residuals:                    2760   BIC:                             6998.
Df Model:                           4
Covariance Type:            nonrobust
==============================================================================
                 coef    std err          t      P>|t|      [0.025      0.975]
------------------------------------------------------------------------------
const          0.0414      0.026      1.613      0.107      -0.009       0.092
Open          -0.6532      0.015    -44.609      0.000      -0.682      -0.625
High           0.9939      0.014     72.433      0.000       0.967       1.021
Low            0.6542      0.013     51.931      0.000       0.630       0.679
Volume     -5.968e-10   2.05e-10     -2.906      0.004   -9.99e-10   -1.94e-10
==============================================================================
Omnibus:                      931.464   Durbin-Watson:                   2.483
Prob(Omnibus):                  0.000   Jarque-Bera (JB):           268969.966
Skew:                          -0.060   Prob(JB):                         0.00
Kurtosis:                      51.318   Cond. No.                     2.08e+08
==============================================================================

Notes:
[1] Standard Errors assume that the covariance matrix of the errors is correctly spe
cified.
[2] The condition number is large, 2.08e+08. This might indicate that there are
strong multicollinearity or other numerical problems.
Train RMSE: 0.8515773313243238
Test RMSE: 3.4843273675529884
Train R-squared: 0.9997446234797391
Test R-squared: 0.996441859319228
```


TSLA Stock Price Prediction
Train
Test
Train Prediction
Test Prediction
Price
Date
0
50
100
150
200
250
300
350
400
2010
2012
2014
2016
2018
2020
2022
2024

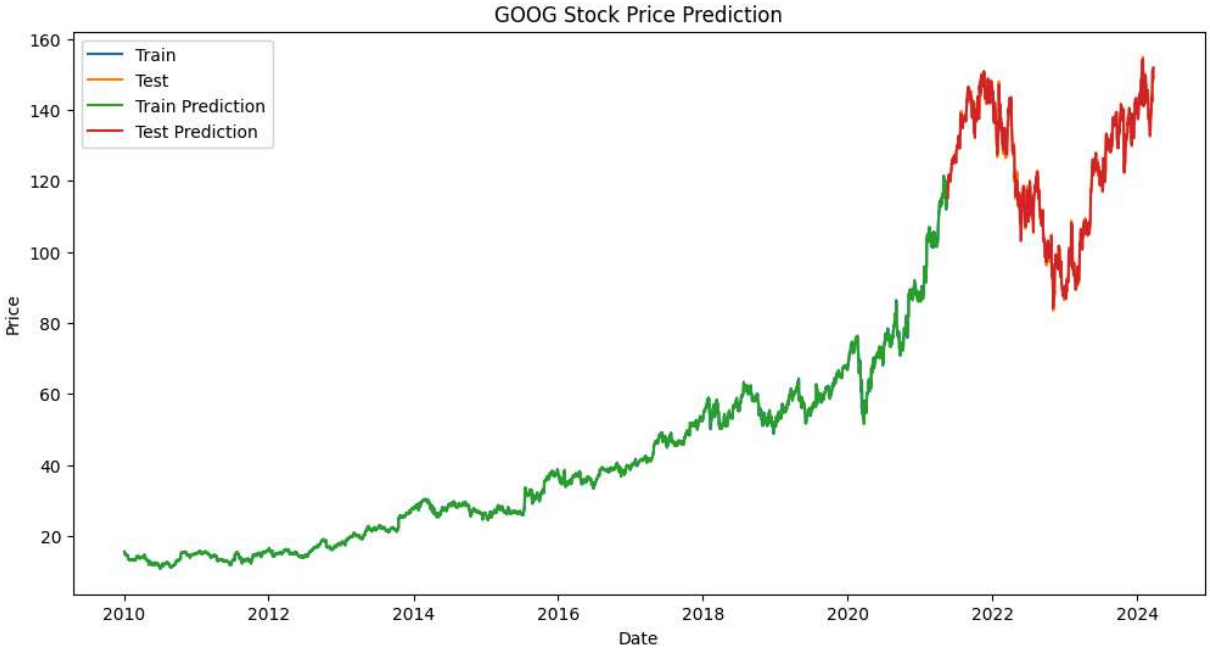

TSLA Stock Price Prediction Residuals
Train Residuals
Test Residuals
Price
Date
−15
−10
−5
0
5
10
2010
2012
2014
2016
2018
2020
2022
2024

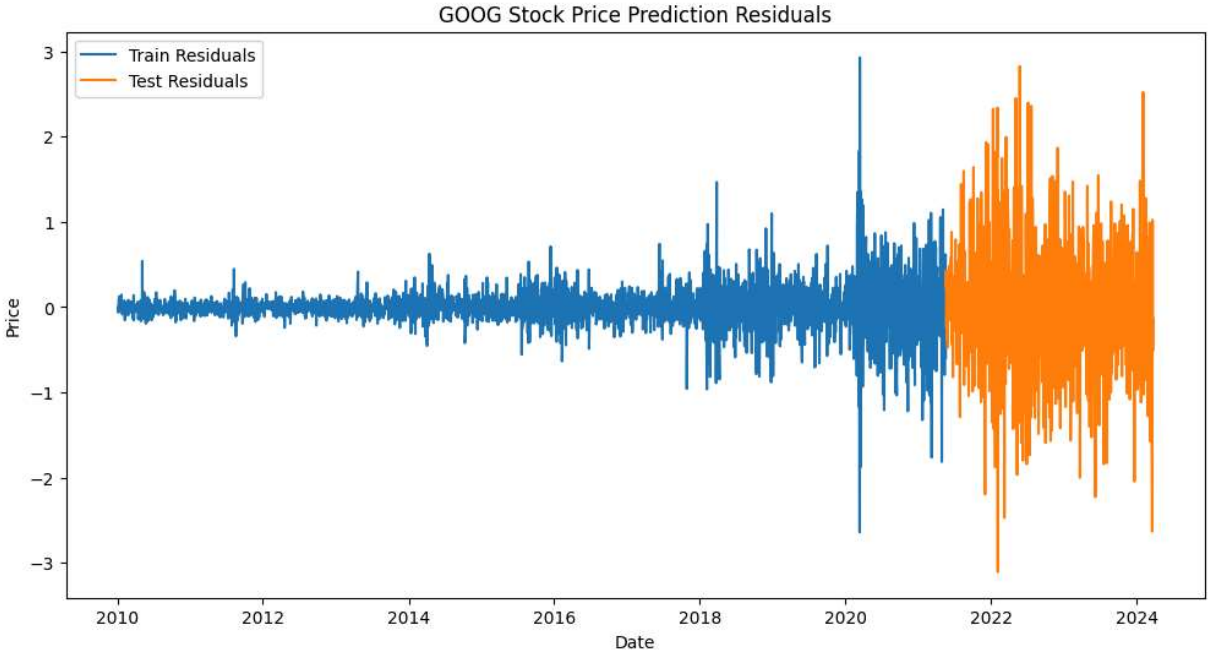
```
*** NFLX ***
                            OLS Regression Results
==============================================================================
Dep. Variable:                  Close   R-squared:                       1.000
Model:                            OLS   Adj. R-squared:                  1.000
Method:                 Least Squares   F-statistic:                 4.061e+06
Date:                Fri, 29 Mar 2024   Prob (F-statistic):               0.00
Time:                        14:02:17   Log-Likelihood:                -6211.8
No. Observations:                2863   AIC:                         1.243e+04
Df Residuals:                    2858   BIC:                         1.246e+04
Df Model:                           4
Covariance Type:            nonrobust
==============================================================================
                 coef    std err          t      P>|t|      [0.025      0.975]
------------------------------------------------------------------------------
const          0.0233      0.084      0.278      0.781      -0.141       0.187
Open          -0.6051      0.015    -40.805      0.000      -0.634      -0.576
High           0.8188      0.013     61.967      0.000       0.793       0.845
Low            0.7865      0.013     58.811      0.000       0.760       0.813
Volume     -6.431e-11   2.13e-09     -0.030      0.976   -4.23e-09     4.1e-09
==============================================================================
Omnibus:                      720.349   Durbin-Watson:                   2.374
Prob(Omnibus):                  0.000   Jarque-Bera (JB):            38198.289
Skew:                           0.297   Prob(JB):                         0.00
Kurtosis:                      20.885   Cond. No.                     6.15e+07
==============================================================================

Notes:
[1] Standard Errors assume that the covariance matrix of the errors is correctly spe
cified.
[2] The condition number is large, 6.15e+07. This might indicate that there are
strong multicollinearity or other numerical problems.
Train RMSE: 2.1185587949421634
Test RMSE: 3.5358668553700507
Train R-squared: 0.9998240848830847
Test R-squared: 0.9993343245125886
```


NFLX Stock Price Prediction
Train
Test
Train Prediction
Test Prediction
Price
Date
0
100
200
300
400
500
600
700
2010
2012
2014
2016
2018
2020
2022
2024


NFLX Stock Price Prediction Residuals
Train Residuals
Test Residuals
Price
Date
−20
−10
0
10
20
2010
2012
2014
2016
2018
2020
2022
2024

```
*** GOOG ***
                            OLS Regression Results
==============================================================================
Dep. Variable:                  Close   R-squared:                       1.000
Model:                            OLS   Adj. R-squared:                  1.000
Method:                 Least Squares   F-statistic:                 5.552e+06
Date:                Fri, 29 Mar 2024   Prob (F-statistic):               0.00
Time:                        14:02:17   Log-Likelihood:                -228.70
No. Observations:                2863   AIC:                             467.4
Df Residuals:                    2858   BIC:                             497.2
Df Model:                           4
Covariance Type:            nonrobust
==============================================================================
                 coef    std err          t      P>|t|      [0.025      0.975]
------------------------------------------------------------------------------
const          0.0297      0.016      1.859      0.063      -0.002       0.061
Open          -0.5518      0.015    -36.859      0.000      -0.581      -0.522
High           0.7578      0.014     55.773      0.000       0.731       0.784
Low            0.7940      0.013     63.064      0.000       0.769       0.819
Volume     -1.095e-10   1.15e-10     -0.951      0.342   -3.35e-10    1.16e-10
==============================================================================
Omnibus:                      641.039   Durbin-Watson:                   2.337
Prob(Omnibus):                  0.000   Jarque-Bera (JB):            29122.231
Skew:                          -0.047   Prob(JB):                         0.00
Kurtosis:                      18.624   Cond. No.                     3.13e+08
==============================================================================

Notes:
[1] Standard Errors assume that the covariance matrix of the errors is correctly spe
cified.
[2] The condition number is large, 3.13e+08. This might indicate that there are
strong multicollinearity or other numerical problems.
Train RMSE: 0.26209242546780065
Test RMSE: 0.8049329471454075
Train R-squared: 0.9998713351539616
Test R-squared: 0.9979055471020055
```




| | |
|---|---|
| CONCLUSION | In this experiment we learned how to perform regression analysis on real-time stock data. We fetched historical stock data for three stocks, trained regression models using OLS, and evaluated the models on statistical metrics. The model summaries provided insights into the relationships between stock prices and other attributes. We also assessed the models' goodness of fit using RMSE and R2 scores. Overall, this experiment demonstrated the application of regression analysis in finance and the importance of model evaluation for predictive accuracy. |