APR(CS6103) Assignment-1: Stock Price Prediction using Linear Regression

Author: Abdul Khazmuddin

Roll: 2511CS06

Email: abdul_2511cs06@iitp.ac.in

Phone: 7003532322

Introduction: This assignment demonstrates the application of supervised learning using Linear Regression for predicting stock prices. The goal is to predict the next day's closing price of Apple Inc. (AAPL) stock using historical market data such as Open, High, Low, and Volume.

Dataset & Features - Source: Yahoo Finance (yfinance library)

- Stock: Apple Inc. (AAPL)
- Period: 1 Jan 2020 – 1 Jan 2023
- Features: Open, High, Low, Volume
- Target: Next day’s Close price

Methodology

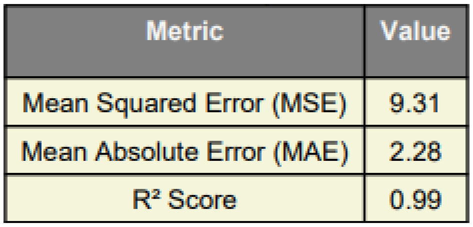
1. Download stock data using yfinance.
2. Create target variable (Close[t+1]).
3. Split dataset into train (80%) and test (20%).
4. Train a Linear Regression model.
5. Evaluate using MSE, MAE, and R² Score

Code:

```
# APR(CS6103) Assignment-1: Stock Price Prediction using Linear Regression
# Author: Abdul Khazmuddin
# Roll: 2511CS06
# Email: abdul_2511cs06@iitp.ac.in
# Phone: 7003532322

import yfinance as yf
import pandas as pd
import numpy as np
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LinearRegression
from sklearn.metrics import mean_squared_error, mean_absolute_error, r2_score

stock_name = "AAPL"
# stock data (example: Apple Inc. - AAPL) from yfinance
data = yf.download(stock_name, start="2020-01-01", end="2023-01-01")

# Feature engineering
# This will predict NEXT day's closing price using today's Open, High, Low, Volume
data['Target'] = data['Close'].shift(-1)  # next day's close
data = data.dropna()

X = data[['Open', 'High', 'Low', 'Volume']]
y = data['Target']

# Spliting into train and test sets
X_train, X_test, y_train, y_test = train_test_split(
    X, y, test_size=0.2, random_state=42
)

# Training Linear Regression model
model = LinearRegression()
model.fit(X_train, y_train)

# Prediction
y_pred = model.predict(X_test)
```

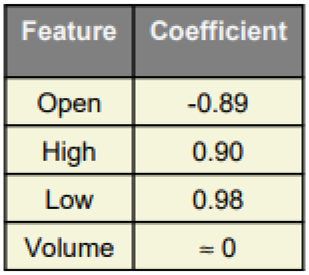
```
# Evaluating model
mse = mean_squared_error(y_test, y_pred)
mae = mean_absolute_error(y_test, y_pred)
r2 = r2_score(y_test, y_pred)

print("\nModel Performance:")
print("Mean Squared Error (MSE):", mse)
print("Mean Absolute Error (MAE):", mae)
print("R² Score:", r2)

# coefficients
coef_df = pd.DataFrame({
   "Feature": X.columns,
   "Coefficient": model.coef_
})
print("\nModel Coefficients:")
print(coef_df.head())
```

Results: The model performance on test data is as follows:



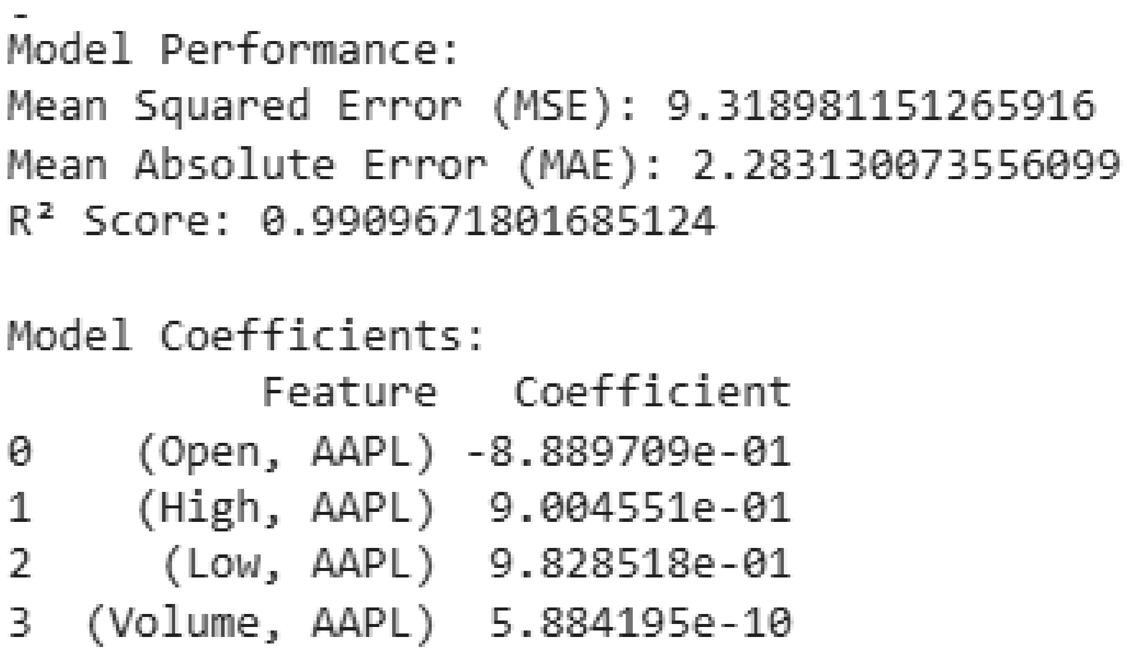
Metric	Value
Mean Squared Error (MSE)	9.31
Mean Absolute Error (MAE)	2.28
R^2 Score	0.99

The coefficients of the Linear Regression model are:



Feature	Coefficient
Open	-0.89
High	0.90
Low	0.98
Volume	≈ 0

Output Screenshot:



```
-
Model Performance:
Mean Squared Error (MSE): 9.318981151265916
Mean Absolute Error (MAE): 2.283130073556099
R² Score: 0.9909671801685124

Model Coefficients:
          Feature   Coefficient
0    (Open, AAPL) -8.889709e-01
1    (High, AAPL)  9.004551e-01
2     (Low, AAPL)  9.828518e-01
3  (Volume, AAPL)  5.884195e-10
```

Conclusion: The model shows very high accuracy with $R^2 = 0.99$, suggesting that the chosen features explain most of the variance in the target variable. However, in real-world scenarios, stock price prediction is far more complex due to market volatility, external news, and non-linear dependencies. This assignment demonstrates the usage of Linear Regression as a simple supervised learning model in the financial domain.