Minor Project

G Varshith Reddy

Project Report

Stock Price Prediction

(Using Linear Regression)

Introduction

The goal of this project is to predict the future stock prices of Samsung Electronics using linear regression. Accurate stock price prediction is valuable for investors, traders, and financial analysts as it aids in making informed investment decisions. In this project, we use historical stock data to train a linear regression model and evaluate its performance.

Dataset

The dataset used in this project contains historical stock prices of Samsung Electronics. The data is stored in a ZIP file named `Samsung electronics dataset.zip` and includes daily stock prices over a specified period.

Steps

1. Extracting the Dataset

The ZIP file containing the dataset is extracted to a specified directory using Python's `zipfile` module. The extracted files are then listed to identify the CSV file containing the stock data.

2. Loading the Data

The CSV file is loaded into a Pandas DataFrame for analysis. The first few rows of the dataset are displayed to understand its structure and contents.

3. Preprocessing the Data

- The 'Date' column is converted to a datetime format and set as the index of the DataFrame.

- A 50-day moving average (`50_MA`) of the 'Close' price is calculated to smoothen short-term fluctuations and highlight longer-term trends.

- Rows with missing values are dropped to ensure clean data for training the model.

4. Training the Linear Regression Model

- The feature (`X`) is the 50-day moving average, and the target variable (`y`) is the 'Close' price.

- The data is split into training and testing sets using an 80-20 split.

- A linear regression model is trained on the training set.

5. Evaluating the Model

- The model's predictions on the test set are compared to the actual 'Close' prices.

- Performance metrics such as Mean Absolute Error (MAE) and Root Mean Squared Error (RMSE) are calculated to quantify the model's accuracy.

6. Visualizing the Results

- The actual and predicted stock prices are plotted for visual comparison.

- The plot includes labels, a title, and a legend for clarity.

Minor Project

G Varshith Reddy

Code Used (Python)

```
import zipfile
import os
import pandas as pd
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LinearRegression
from sklearn.metrics import mean_absolute_error, mean_squared_error
import numpy as np
import matplotlib.pyplot as plt

# Step 1: Extract the ZIP file
zip_file_path = 'C:/Users/varsh/OneDrive/Desktop/MInor Project(Corizo)/Samsung electronics dataset.zip'
extracted_path = 'C:/Users/varsh/OneDrive/Desktop/MInor Project(Corizo)/Samsung_dataset/'

with zipfile.ZipFile(zip_file_path, 'r') as zip_ref:
    zip_ref.extractall(extracted_path)

# List the extracted files
extracted_files = os.listdir(extracted_path)
print(f'Extracted files: {extracted_files}')

# Step 2: Load the Data
csv_file_path = os.path.join(extracted_path, extracted_files[0])
data = pd.read_csv(csv_file_path)

# Display the first few rows of the dataset
print(f'First few rows of the dataset:\n{data.head()}')

# Step 3: Preprocess the Data
data['Date'] = pd.to_datetime(data['Date'])
```

```
data.set_index('Date', inplace=True)
data['50_MA'] = data['Close'].rolling(window=50).mean()
data.dropna(inplace=True)

# Display the first few rows of the modified dataset
print(f'Dataset after preprocessing:\n{data.head()}')

# Step 4: Train the Linear Regression Model
X = data[['50_MA']]
y = data['Close']

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)

mae = mean_absolute_error(y_test, y_pred)
rmse = np.sqrt(mean_squared_error(y_test, y_pred))

print(f'Mean Absolute Error: {mae}')
print(f'Root Mean Squared Error: {rmse}')

# Step 5: Plot the Actual vs. Predicted Prices
plt.figure(figsize=(14, 7))
plt.plot(y_test.index, y_test, label='Actual Prices', color='blue', linewidth=2)
plt.plot(y_test.index, y_pred, label='Predicted Prices', color='red', linestyle='--', linewidth=2)
plt.title('Actual vs. Predicted Stock Prices', fontsize=16)
plt.xlabel('Date', fontsize=14)
plt.ylabel('Stock Price', fontsize=14)
```

Minor Project

G Varshith Reddy

plt.legend(fontsize=12)

plt.grid(True)

plt.show()

Results

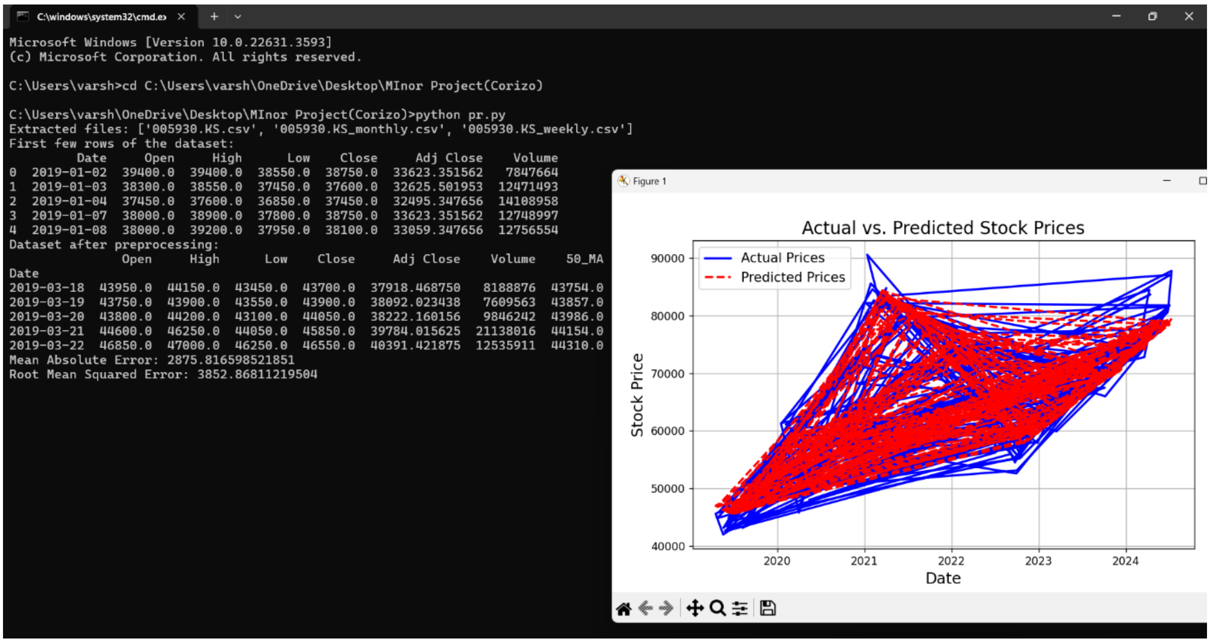
The model's performance is evaluated using MAE and RMSE:

- Mean Absolute Error (MAE): Measures the average magnitude of the errors in a set of predictions, without considering their direction.

- Root Mean Squared Error (RMSE): Measures the square root of the average of the squared differences between predicted and actual values.

The visualization clearly shows the actual vs. predicted stock prices, allowing for easy comparison of the model's performance over time.

Output



```
Microsoft Windows [Version 10.0.22631.3593]
(c) Microsoft Corporation. All rights reserved.

C:\Users\varsh>cd C:\Users\varsh\OneDrive\Desktop\MInor Project(Corizo)

C:\Users\varsh\OneDrive\Desktop\MInor Project(Corizo)>python pr.py
Extracted files: ['005930.KS.csv', '005930.KS_monthly.csv', '005930.KS_weekly.csv']
First few rows of the dataset:
         Date     Open     High      Low    Close     Adj Close    Volume
0  2019-01-02  39400.0  39400.0  38550.0  38750.0  33623.351562   7847664
1  2019-01-03  38300.0  38550.0  37450.0  37600.0  32625.501953  12471493
2  2019-01-04  37450.0  37600.0  36850.0  37450.0  32495.347656  14108958
3  2019-01-07  38000.0  38900.0  37800.0  38750.0  33623.351562  12748997
4  2019-01-08  38000.0  39200.0  37950.0  38100.0  33059.347656  12756554
Dataset after preprocessing:
               Open     High      Low    Close     Adj Close    Volume    50_MA
Date
2019-03-18  43950.0  44150.0  43450.0  43700.0  37918.468750   8188876  43754.0
2019-03-19  43750.0  43900.0  43550.0  43900.0  38092.023438   7609563  43857.0
2019-03-20  43800.0  44200.0  43100.0  44050.0  38222.160156   9846242  43986.0
2019-03-21  44600.0  46250.0  44050.0  45850.0  39784.015625  21138016  44154.0
2019-03-22  46850.0  47000.0  46250.0  46550.0  40391.421875  12535911  44310.0
Mean Absolute Error: 2875.816598521851
Root Mean Squared Error: 3852.86811219504
```

Minor Project

G Varshith Reddy

Conclusion

This project demonstrates how linear regression can be used to predict stock prices based on historical data. The preprocessing steps, including calculating moving averages, are crucial for improving model performance. The results provide valuable insights for investors and can be further refined by exploring additional features and more sophisticated models.