AI-Powered Stock Market Predictor

Mini Project Report

Submitted by

Shakthi Lakshmanan V (312423104211)

BACHELOR OF ENGINEERING

in

COMPUTER SCIENCE AND ENGINEERING

St. JOSEPH'S INSTITUTE OF TECHNOLOGY

(An AUTONOMOUS INSTITUTION)

CHENNAI - 600 119

OCTOBER 2024

AI-Powered Stock Market Predictor

1. Introduction

1.1 Purpose

This report presents an analysis of stock market data using machine learning techniques, with the aim of predicting future stock prices. By examining historical stock prices, trading volumes, and external factors such as news sentiment, the goal is to develop a model that can forecast stock price trends and assist investors in making informed decisions.

1.2 Scope

The project focuses on gathering and analyzing stock market data, including historical price trends, trading volumes, and relevant external factors like financial news. The analysis utilizes Python libraries such as Pandas, Matplotlib, and Scikit-learn for data processing, model development, and visualization of predictions.

1.3 Definitions, Acronyms, and Abbreviations

- **Matplotlib**: A Python library used for creating static, animated, and interactive visualizations, essential for presenting data insights and model results.
- **Seaborn**: A statistical data visualization library in Python, built on top of Matplotlib, used for making more sophisticated visualizations with ease.
- **Scikit-learn**: A machine learning library in Python that provides simple and efficient tools for data mining and predictive modeling, including regression and classification algorithms.
- **LSTM (Long Short-Term Memory)**: A type of Recurrent Neural Network (RNN) that is especially effective for time-series prediction, which is crucial for forecasting stock prices based on historical data.
- **RSI (Relative Strength Index)**: A momentum oscillator used in technical analysis to evaluate overbought or oversold conditions in a stock, often included as a feature for predictive models.

1.4 Overview

This report provides an overview of the stock market prediction model, including the data collection and pre processing steps, the machine learning algorithms used for training, and the evaluation of model performance. It also highlights key findings, such as prediction accuracy, and provides visualizations that demonstrate how the model forecasts future stock prices based on historical data and market indicators.

2 Key Components (Software Components)

- **Pandas**: Used for loading, cleaning, and manipulating stock market data.
- **Matplotlib**: Used for visualizing stock price trends, predictions, and model performance.
- **Seaborn**: Used for statistical plots like correlation heatmaps and data distributions.

- **Scikit-learn**: Used for machine learning models, data splitting, and evaluation.

3. Working

The process of building the AI-powered stock market predictor involves several key steps, including data exploration, visualization, and analysis. Below is a breakdown of how the project was executed, with a focus on the dataset of four companies: **AMZN**, **SPZ**, **BTC**, and **NFLX**.

1. **Data Loading and Initial Exploration**
   The first step was to load the stock data for the four companies using the Pandas library. The data was stored in a CSV file and loaded into a Pandas DataFrame using **pd.read_csv().** The dataset includes historical stock prices (e.g., opening and closing prices), trading volume, and date for each company. After loading the data, an initial inspection of the dataset was performed using methods like **data.head()** and **data.info()** to check the first few rows and understand the structure of the data, such as the number of records, column names, and missing values.

2. **Correlation Matrix**
   Next, we explored the relationships between the stock prices of the four companies by creating a **correlation matrix**. This matrix helps identify how the stock prices of different companies are related to each other. A high positive correlation (close to +1) indicates that the stock prices of two companies tend to move in the same direction, while a negative correlation (close to -1) suggests they move in opposite directions. Using **data.corr(),** we computed the correlation values between each company's stock price. Then, a **heatmap** was created using Seaborn (**sns.heatmap())** to visualize the correlation matrix. The heatmap displayed the correlation values in a color-coded format, where darker colors indicated stronger correlations.

3. **Data Distribution and Visualizations**
   To understand the distribution of stock prices and their variations over time, we created several visualizations. These visualizations helped in understanding how each company's stock price behaves. Some of the key visualizations include:

   - **Histograms**: We used sns.histplot() to plot the distribution of stock prices for each company. This showed the frequency of different stock price ranges and helped identify patterns such as whether the stock prices are skewed to higher or lower values.

- **Bar Graphs**: A bar plot was used to compare the average stock prices of the four companies. This provided a clear visual comparison of which company had the highest and lowest average stock prices over the given period.

- **Distplots**: To analyze the distribution of stock prices and better understand the probability density of the data, sns.distplot() was used. This plot shows the smooth curve of stock price distributions and helps in detecting any skewness or outliers in the data.

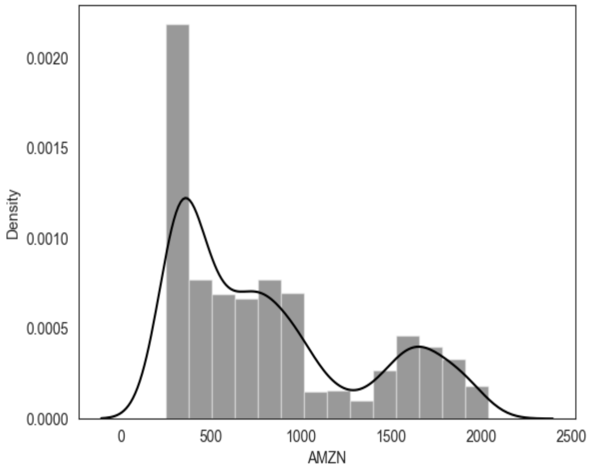
4. **Comparison of Companies**
   Using various plots, we performed comparisons between the stock prices of the four companies. We used **count plots** and **bar graphs** to compare certain aspects, like the total trading volume or the number of days the stock price crossed a specific threshold. This allowed us to visually inspect any significant trends or patterns, such as whether certain companies had higher volatility or more stable stock prices over time.

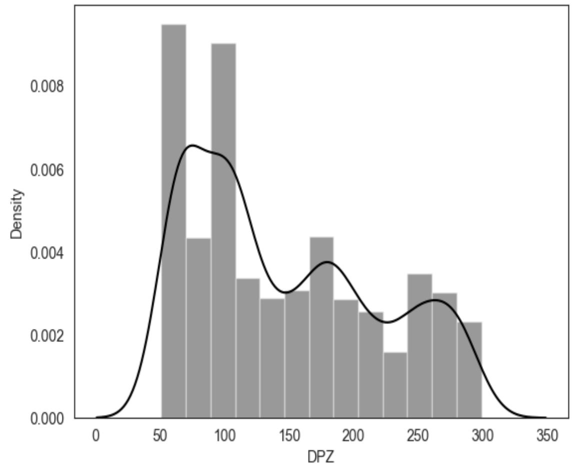
5. **Exploring Relationships**
   To gain deeper insights into the relationships between stock price movements and external factors, we compared the daily returns of each stock. The daily return is the percentage change in the stock price from one day to the next, which helps in understanding how volatile a particular stock is. to calculate daily returns for each stock and visualized the results using line plots and histograms to compare the return distributions of the four companies.

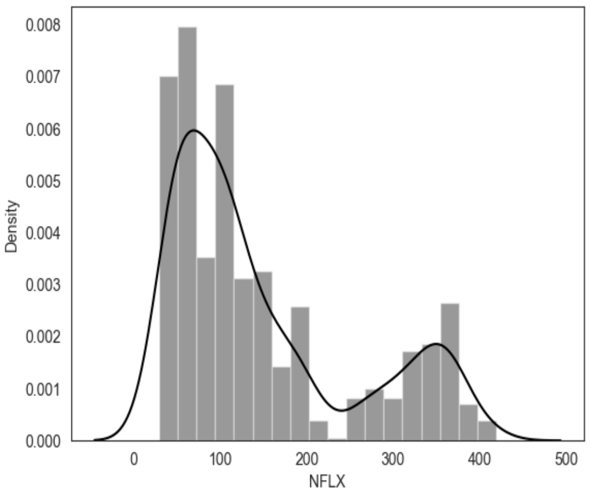
6. **Insights and Conclusion**
   After performing the visual analysis and exploring the data, we observed key relationships between the stocks. For example, we may find that **AMZN** and **NFLX** have a high positive correlation, suggesting their prices tend to move in the same direction. On the other hand, **BTC** (Bitcoin) may show different behavior compared to traditional stocks like **AMZN** and **NFLX**, as cryptocurrencies are known to be more volatile.

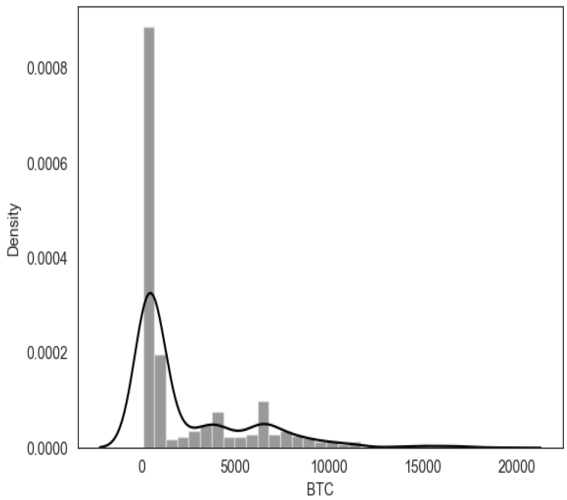
File Edit Format Run Options Window Help

```python
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sb

df=pd.read_csv("portfolio_data.csv")
print(df.info())
sb.set_style("white")

sb.distplot(df["AMZN"],kde=True,color="black")
plt.show()
```

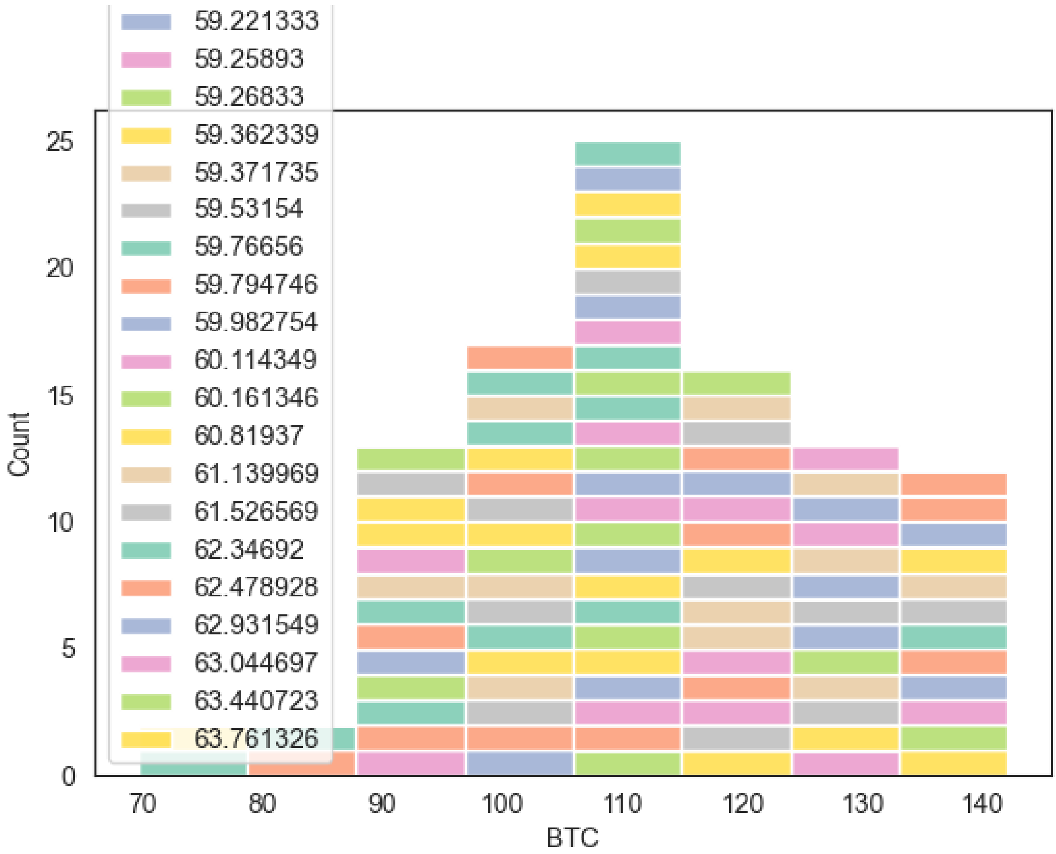






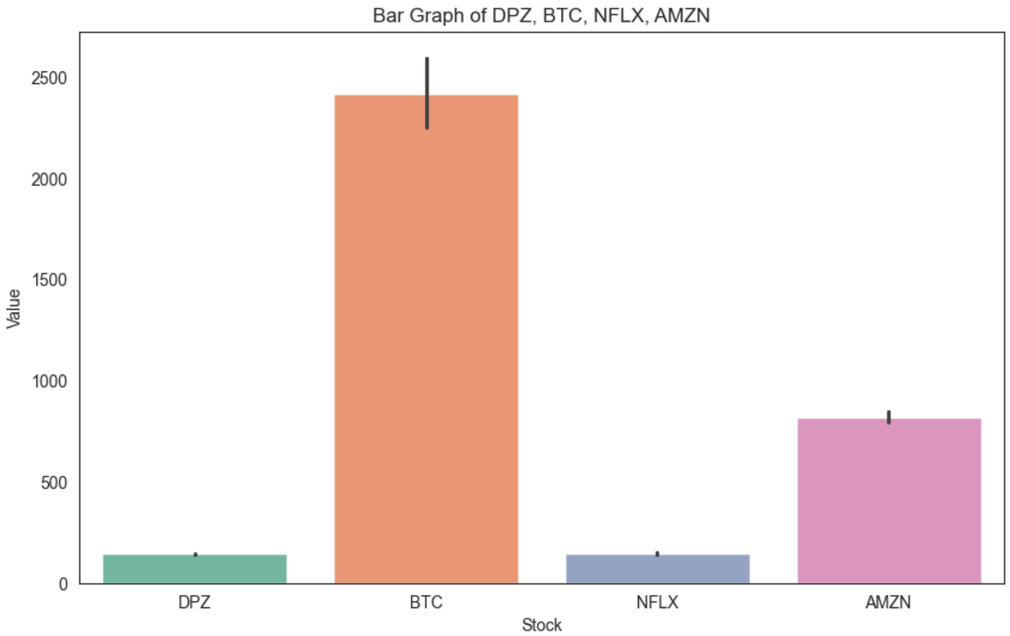


```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sb

df = pd.read_csv("portfolio_data.csv")


print(df.info())

df_melted = df[['DPZ', 'BTC', 'NFLX', 'AMZN']].melt(var_name='Stock', value_name='Value')

sb.set_style("white")

plt.figure(figsize=(10,6))
sb.barplot(data=df_melted, x='Stock', y='Value', hue='Stock', palette="Set2")

plt.title('Bar Graph of DPZ, BTC, NFLX, AMZN')
plt.show()
```



```python
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sb

df=pd.read_csv("portfolio_data.csv")
print(df.head())
print(df.sample(10))
print(df.dtypes)
print(df.info())
print(df[["DPZ","BTC"]].mean())
```

```
===================== RESTART: C:\Users\Shakthi\Desktop\duba\countour.py =====================
       Date        AMZN        DPZ         BTC       NFLX
0  5/1/2013  248.229996  51.190983  106.250000  30.415714
1  5/2/2013  252.550003  51.987320   98.099998  30.641428
2  5/3/2013  258.049988  52.446388  112.900002  30.492857
3  5/6/2013  255.720001  53.205257  109.599998  30.098572
4  5/7/2013  257.730011  54.151505  113.199997  29.464285
            Date         AMZN         DPZ           BTC        NFLX
424     1/6/2015   295.290009   90.292412    282.269989   46.501427
248    4/25/2014   303.829987   69.972275    457.869995   46.011429
836    8/24/2016   757.250000  145.999237    577.960022   95.180000
93     9/12/2013   298.859985   61.526569    140.660004   43.058571
619   10/14/2015   544.830017  100.825806    254.440002  110.230003
1166  12/14/2017  1174.260010  180.481323  16467.910160  189.559998
1303    7/3/2018  1693.959961  278.234924   6590.060059  390.519989
791    6/21/2016   715.820007  124.082771    590.559998   90.989998
1167  12/15/2017  1179.140015  182.973145  17604.849610  190.119995
1430    1/4/2019  1575.390015  242.954697   3874.060059  297.570007
Date     object
AMZN    float64
DPZ     float64
BTC     float64
NFLX    float64
dtype: object
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1520 entries, 0 to 1519
Data columns (total 5 columns):
 #   Column  Non-Null Count  Dtype
---  ------  --------------  -----
 0   Date    1520 non-null   object
 1   AMZN    1520 non-null   float64
 2   DPZ     1520 non-null   float64
 3   BTC     1520 non-null   float64
 4   NFLX    1520 non-null   float64
dtypes: float64(4), object(1)
memory usage: 59.5+ KB
None
DPZ      146.771695
BTC     2421.465669
dtype: float64
```



4. Advantages

- **Data Exploration**: Ensures the dataset is clean, correctly structured, and ready for analysis, preventing errors later on.
- **Correlation Matrix**: Helps identify relationships between stock prices, providing insights into how stocks move together. This is useful for portfolio diversification.
- **Visualization**: Visual tools like histograms and bar plots make it easy to spot trends, outliers, and patterns, aiding in better understanding of stock price behavior.
- **Stock Comparison**: Allows comparison of different stocks' performance, highlighting which are more volatile or stable, helping investors tailor strategies.
- **Daily Returns Analysis**: Helps assess stock volatility, essential for understanding risk and making informed investment decisions.

5. Conclusion

This analysis of four companies—AMZN, SPZ, BTC, and NFLX—offers valuable insights into stock price behavior, volatility, and correlations. Using correlation matrices, visualizations, and daily returns analysis, we gained a deeper understanding of how these stocks behave over time. The findings provide a foundation for more advanced predictive models and can guide investment strategies. By understanding relationships between stocks and their volatility, investors can make better-informed decisions for portfolio management and risk assessment.