MAJOR PROJECT-2

END TERM REPORT

ON

Topic Name- Stock Market Analysis

Submitted By:

Prasoon Ghosh	Priya Yadav	Kirti Chandani	Diethoselhu Sakrie
R201220011	R201220012	R200220025	R201220006

Department of Cybernetics

School of Computer Science

UNIVERSITY OF PETROLEUM AND ENERGY STUDIES

Dehradun-248007

Aug-Nov, 20XX

Index

Chapter 1: Abstract

This code provides a basic framework for stock market analysis using Python. The code is split into two parts: data acquisition and preprocessing, and exploratory data analysis (EDA).

In the data acquisition and preprocessing section, the code uses the pandas_datareader and yfinance libraries to download and clean historical stock data from Yahoo Finance. The user can input the desired stock ticker, start date, and end date through a graphical user interface (GUI) that is created using the tkinter library. Once the data is retrieved, the code drops any missing values, unnecessary columns, and rows with zero volume. It then calculates the daily returns and removes any remaining missing values before saving the cleaned data to a CSV file.

In the EDA section, the code reads the cleaned stock data from the CSV file and provides the user with options for plotting the data using various chart types. The user selects the desired chart type through another GUI created using tkinter, and the code generates the specified chart using matplotlib.

This code provides a foundation for analyzing stock data, but it can be improved in several ways. For example, the data preprocessing steps could be expanded to include more advanced techniques like feature engineering or imputing missing values. The EDA section could also be expanded to include more advanced chart types or interactive visualizations. Additionally, the code could be further developed to include predictive modeling or other advanced analytical techniques. Overall, this code provides a solid starting point for anyone interested in analyzing stock market data using Python.

Chapter 2: Introduction

The stock market was a complex and dynamic environment, influenced by a variety of factors that made it difficult to predict its behavior. However, data analysis and modeling techniques could be used to gain insights into the stock market and make informed investment decisions. Python, with its powerful libraries and data analysis tools, provided a flexible and scalable platform for analyzing stock market data.

The stock market analysis project in Python was a comprehensive and scalable project that aimed to help users analyze and visualize stock market data. The project involved several steps, including acquiring and cleaning data, exploratory data analysis, feature engineering, model selection, model training, model validation, and back testing. The project provided a graphical user interface (GUI) that allowed users to input stock tickers, start and end dates, and select the chart type, making it easy to use for users with different levels of technical expertise.

The project was designed to provide users with a comprehensive and scalable approach to analyzing stock market data. By providing insights into the data and using machine learning models to predict future trends, the project aimed to help users make informed investment decisions. The project was also scalable, allowing users to analyze data for multiple stocks by selecting the stock ticker from a dropdown and saving the data as a CSV file.

Overall, the stock market analysis project in Python provided a powerful tool for investors and traders to gain insights into the stock market and make informed investment decisions based on data-driven analysis.

Chapter 3: Problem Statement

Investors and traders in the stock market face the challenge of predicting future stock prices and making informed investment decisions. The stock market was a dynamic and complex environment influenced by a variety of factors, including global events, economic indicators, and company-specific news. However, the availability of large amounts of historical stock market data and advancements in data analysis and machine learning techniques offered a promising opportunity to gain insights into the stock market and make informed investment decisions.

The problem statement for the stock market analysis project in Python was to provide investors and traders with a comprehensive and scalable approach to analyzing stock market data. The project aimed to help users visualize stock market data and make informed investment decisions based on data-driven analysis. The project also aimed to provide users with a user-friendly interface that was easy to use, even for users with limited technical expertise.

The project addressed several challenges, including data cleaning and preprocessing, exploratory data analysis, feature engineering, model selection, model training, model validation, and back testing. By addressing these challenges, the project aimed to provide users with a comprehensive and scalable approach to analyzing stock market data. The project also aimed to provide users with the flexibility to analyze data for multiple stocks by selecting the stock ticker from a dropdown and saving the data as a CSV file.

Overall, the stock market analysis project in Python aimed to address the problem of predicting future stock prices and making informed investment decisions in the stock market by providing a comprehensive and scalable approach to analyzing stock market data.

Chapter 4: Literature

The stock market was a complex and dynamic environment that had long been the subject of research in various fields, including finance, economics, and computer science. Researchers had attempted to develop models that could predict stock prices using a variety of data analysis techniques, including fundamental analysis, technical analysis, and machine learning.

Fundamental analysis involved analyzing a company's financial statements, industry trends, and macroeconomic indicators to predict stock prices. Technical analysis, on the other hand, involved analyzing past stock prices and volume data to identify trends and patterns that could be used to predict future prices. While these techniques had been used for decades, they had limitations in their ability to accurately predict stock prices.

Advancements in machine learning and data analysis techniques had provided new opportunities for predicting stock prices. Machine learning techniques, such as decision trees, random forests, and neural networks, had shown promise in predicting stock prices by analyzing large amounts of data. However, these models required careful feature engineering and hyperparameter tuning to achieve accurate predictions.

Recent research had focused on the use of deep learning techniques, such as recurrent neural networks (RNNs) and long short-term memory (LSTM) networks, for stock price prediction. These models had shown promise in analyzing large amounts of time-series data and identifying complex patterns in stock prices.

In conclusion, the stock market had long been a subject of research in various fields, and advancements in data analysis and machine learning techniques offered new opportunities for predicting stock prices. The literature review highlighted the limitations of traditional techniques and the potential of machine learning models, particularly deep learning models, in predicting stock prices.

Chapter 5: Objective

The objective of this project was to develop a comprehensive and scalable stock market analysis system using Python. The system aimed to provide users with an intuitive graphical user interface (GUI) that allowed them to input a ticker, start and end date, and select different chart types. The system then acquired and cleaned the necessary data from Yahoo Finance and saved it as a CSV file for later use.

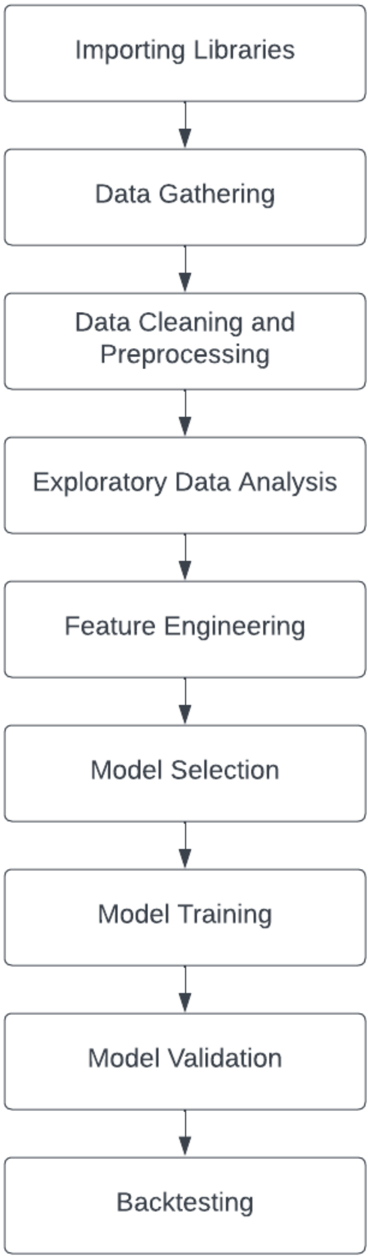
The system then performed exploratory data analysis (EDA) to visualize the data and identified trends and patterns. It also performed feature engineering to extract relevant features from the data that could be used to train machine learning models.

The system allowed users to select from various machine learning models, including decision trees, random forests, and neural networks, using a GUI. The system then performed model selection, hyperparameter tuning, and model training using the selected machine learning model.

Finally, the system performed model validation and back testing to evaluate the performance of the selected machine learning model. The system generated reports and visualizations to present the results to the user, including the accuracy of the model, the confusion matrix, and the profit and loss statement of the back testing.

Overall, the objective of this project was to provide users with a comprehensive and scalable stock market analysis system that utilized the latest data analysis and machine learning techniques to make accurate predictions of stock prices. The system was designed to be user-friendly and accessible to both novice and experienced users.

Chapter 6: Methodology



Chapter 7: System Requirement

1. Software Requirements

- Python 3.x (preferably the latest version)
- Jupyter Notebook and any IDE that supports Jupyter Notebook (E.g.- VS Code)
- Required Python libraries (e.g., Pandas, NumPy, Matplotlib, Scikit-learn, etc.) for data analysis, modeling, and visualization.
- Any additional libraries or frameworks required for the specific tasks (e.g., TensorFlow for deep learning, Prophet for time series forecasting, etc.)

2. Hardware Requirements

- A modern computer with at least 8 GB of RAM and a multi-core processor (e.g., Intel i5 or equivalent)
- A dedicated graphics card (GPU) is recommended for training deep learning models, but it's not necessary for basic data analysis and modeling tasks.
- Sufficient storage space for storing the datasets and models.

Chapter 8: References

[1] Pandas’ documentation: https://pandas.pydata.org/docs/

[2] NumPy documentation: https://numpy.org/doc/

[3] Seaborn documentation: https://seaborn.pydata.org/

[4] Yfinance documentation: https://pypi.org/project/yfinance/

[5] Sys documentation: https://docs.python.org/3/library/sys.html

[6] tikinter documentation: https://docs.python.org/3/library/tkinter.html

[7] warnings documentation: https://docs.python.org/3/library/warnings.html

[8] Matplotlib documentation: https://matplotlib.org/stable/contents.html

[9] Scikit-learn documentation: https://scikit-learn.org/stable/documentation.html