MAJOR PROJECT-2

SYNOPSIS

ON

Topic Name: Stock Market Analysis

Submitted By

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

Under the guidance of

Ravi Shankar Singhal

Industry Fellow

Department of Cybernetics

Department of Cybernetics

School of Computer Science

UNIVERSITY OF PETROLEUM AND ENERGY STUDIES

Dehradun-248007

Jan-June, 2023

Index

Chapter 1: Abstract

The stock market analysis project in Python is a comprehensive and scalable project that allows users to analyze and visualize stock data. The project involves several steps such as acquiring and cleaning data, exploratory data analysis, feature engineering, model selection, model training, model validation, and back testing. The project allows users to input a stock ticker, start and end dates, and select the chart type through a graphical user interface (GUI).

After acquiring and cleaning the data, the project uses various exploratory data analysis techniques to gain insights into the data. Feature engineering is then performed to create new features from the existing data to improve the accuracy of the models. The user can then select the type of modeling they want to perform, and the model is trained using the training data. The model is then validated to ensure that it performs well on unseen data.

The back testing step involves testing the performance of the model on historical data to see how well it would have performed in the past. The project uses a lookback period of 14 days to calculate the returns and plots the actual vs. predicted returns to evaluate the model's performance. The project is scalable and can be used to analyze data for multiple stocks by allowing the user to select the stock ticker from a dropdown and saving the data as a CSV file.

Overall, the project provides a comprehensive and scalable approach to analyzing stock market data, allowing users to gain insights into the data and make informed investment decisions.

Chapter 2: Introduction

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

The stock market analysis project in Python is a comprehensive and scalable project that aims to help users analyze and visualize stock market data. The project involves several steps, including acquiring and cleaning data, exploratory data analysis, feature engineering, model selection, model training, model validation, and back testing. The project provides a graphical user interface (GUI) that allows 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 is 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 aims to help users make informed investment decisions. The project is 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 provides 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 is 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 offer 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 is to provide investors and traders with a comprehensive and scalable approach to analyzing stock market data. The project aims to help users visualize stock market data and make informed investment decisions based on data-driven analysis. The project also aims to provide users with a user-friendly interface that is easy to use, even for users with limited technical expertise.

The project addresses several challenges, including data cleaning and preprocessing, exploratory data analysis, feature engineering, model selection, model training, model validation, and backtesting. By addressing these challenges, the project aims to provide users with a comprehensive and scalable approach to analyzing stock market data. The project also aims 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 aims 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 is a complex and dynamic environment that has long been the subject of research in various fields, including finance, economics, and computer science. Researchers have attempted to develop models that can predict stock prices using a variety of data analysis techniques, including fundamental analysis, technical analysis, and machine learning.

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

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

Recent research has 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 have shown promise in analyzing large amounts of time-series data and identifying complex patterns in stock prices.

In conclusion, the stock market has long been a subject of research in various fields, and advancements in data analysis and machine learning techniques offer new opportunities for predicting stock prices. The literature review highlights 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 is to develop a comprehensive and scalable stock market analysis system using Python. The system aims to provide users with an intuitive graphical user interface (GUI) that allows them to input a ticker, start and end date, and select different chart types. The system will then acquire and clean the necessary data from Yahoo Finance and save it as a CSV file for later use.

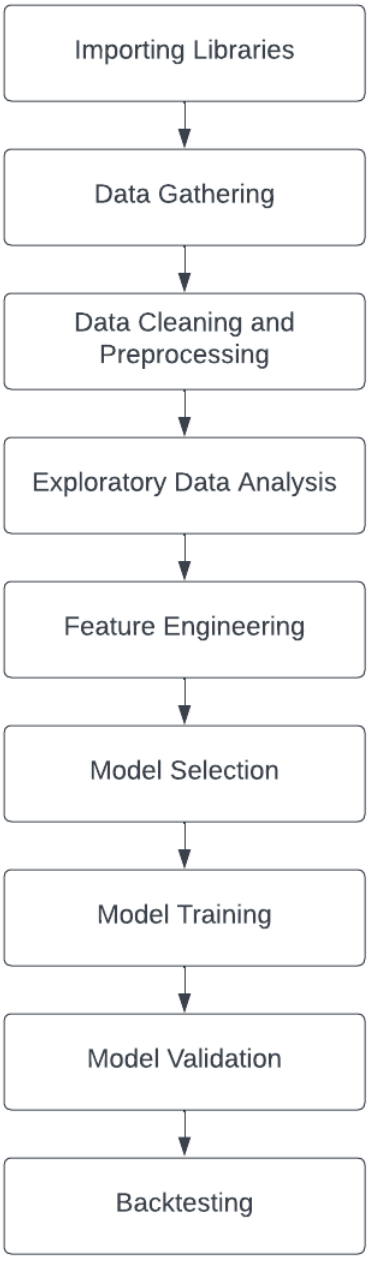
The system will then perform exploratory data analysis (EDA) to visualize the data and identify trends and patterns. It will also perform feature engineering to extract relevant features from the data that can be used to train machine learning models.

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

Finally, the system will perform model validation and back testing to evaluate the performance of the selected machine learning model. The system will generate 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 is to provide users with a comprehensive and scalable stock market analysis system that utilizes the latest data analysis and machine learning techniques to make accurate predictions of stock prices. The system will be 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 or any Python IDE
- 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 9: 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