NAAN MUDHALVAN

PHASE 2 PROJECT SUBMISSION

PROJECT 6 - STOCK PRICE PREDICTION

TEAM MEMBERS:

1. Kannappan P (2021504011)
2. Karthick K (2021504013)
3. Karthikeyan B (2021504014)
4. Pattu Hariharaan N (2021504029)
5. Rubankumar D (2021504034)

STOCK PRICE PREDICTION:

The problem is to build a predictive model that forecasts stock prices based on historical market data. The goal is to create a tool that assists investors in making well-informed decisions and optimizing their investment strategies. The objectives of the projects are:

1. Price Forecasting: The primary objective is to accurately predict future stock prices. This involves minimizing prediction errors and providing forecasts that are as close to the actual stock prices as possible.
2. Investment Decision Support: Assist investors in making informed decisions by providing forecasts and insights. This includes offering guidance on when to buy, sell, or hold stocks based on the model's predictions.
3. Risk Management: Help investors assess and manage risks associated with their investment strategies. This may involve quantifying the uncertainty of predictions and suggesting risk mitigation strategies.

DATASET DETAIL

MSFT.csv contains all the lifetime stock data from 3/13/1986 to 12/10/2019. This dataset contains 7 columns including dates, opening, high, low, closing, adj_close, and volume. LSTMs and Deep Reinforcement Learning agents work well for this dataset.

DATASET LINK

The historical data set of the stock prices of Microsoft were used for this purpose from the following link:

https://www.kaggle.com/datasets/prasoonkottarathil/microsoft-lifetime-stocks-dataset

DETAILS ABOUT COLUMN

Date of stock, Opening, High, Low, Closing, Adj Close, and Volume are the columns given in the dataset. All the columns will be used. A model of LSTM can be built without the Adj Close column.

Dates: This is the date for which the stock price information is recorded. Each row in the dataset typically corresponds to a specific date.

Opening Price: The opening price is the price of the stock at the beginning of the trading session on a given date. It is the first price at which the stock is traded when the market opens.

High Price: The high price represents the highest price at which the stock traded during the trading session on a given date. It reflects the peak value reached by the stock's price during the day.

Low Price: The low price is the lowest price at which the stock traded during the trading session on a given date. It represents the lowest point the stock's price reached during the day.

Closing Price: The closing price is the price of the stock at the end of the trading session on a given date. It is the last price at which the stock is traded for the day.

Adjusted Close Price (Adj. Close): The adjusted close price takes into account corporate actions, such as stock splits, dividends, and other adjustments that can affect the stock's price. It is the closing price adjusted for these events, providing a more accurate representation of the stock's performance over time.

Volume: Volume refers to the total number of shares of the stock that were traded on a given date. It represents the level of trading activity for that day and is often used to assess market liquidity and investor interest in the stock.

LIBRARIES TO BE USED AND WAYS TO DOWNLOAD

1. **NumPy** (Numerical Python) is a popular open-source library for numerical and mathematical operations in Python. It provides support for working with large, multi-dimensional arrays and matrices of numerical data, along with a collection of mathematical functions to operate on these arrays. NumPy is a fundamental library for scientific and data-intensive computing in Python.To install NumPy, you can use a package manager like pip or conda, which are commonly used for Python package management. Here's how to install NumPy using both methods:

Installing NumPy:

Open a terminal or command prompt.

To install NumPy, run the following command:

pip install numpy

2. **Pandas** library is a popular open-source data manipulation and analysis library for the Python programming language. It provides data structures and functions for working with structured data, such as spreadsheets, SQL tables, and time series data. Pandas is widely used

for tasks such as data cleaning, data transformation, data exploration, and data analysis.

Pandas primarily revolve around two main data structures:

DataFrame: A two-dimensional table with labeled axes (rows and columns). It is similar to a spreadsheet or SQL table.

Series: A one-dimensional array-like object that can hold any data type.

Installing Pandas:

Open your command prompt or terminal and run the following command

pip install pandas

3. **Matplotlib** is a widely used Python library for creating 2D plots and charts. It allows you to generate various types of visualizations, such as line plots, bar charts, scatter plots, histograms, and more. Matplotlib's pyplot module is a collection of functions that provides a simple interface for creating basic plots and visualizations.

Installing matplotlib:

Open a terminal or command prompt and run the following command:

pip install matplotlib

4. **Scikit-learn**, often abbreviated as sklearn, is a popular machine-learning library in Python. It provides a wide range of tools and algorithms for machine learning and data analysis tasks, including classification, regression, clustering, dimensionality reduction, model selection, and more. Scikit-learn is built on top of other popular Python libraries like NumPy, SciPy, and Matplotlib, making it an essential tool for data scientists and machine learning practitioners.

Installing Scikit-learn:

Open a terminal or command prompt and run the following command

pip install scikit-learn

5. **Keras** is an open-source high-level neural networks API written in Python. It is capable of running on top of other popular deep learning frameworks like TensorFlo**w** and Theano. Keras provides a user-friendly and modular interface for creating and training deep learning models. It's widely used for tasks such as image and text classification, object detection, natural language processing, and more.Keras can be installed by:

Installing Keras:

Open a terminal or command prompt and run the following command

pip install keras

6. **Seaborn** is a popular Python data visualization library that is built on top of Matplotlib and provides a high-level interface for creating informative and attractive statistical graphics. It is particularly useful for visualizing complex datasets and making it easier to understand and interpret data.

Installing Seaborn:

Open a terminal or command prompt and run the following command

pip install seaborn

How to Train and Test

1] Data Collection:

Gather historical stock price data for the specific stock or index you want to predict. You can obtain this data from various sources, such as financial data providers, APIs, or public datasets**.**

2] Data Preprocessing:

Handle missing data: Replace or remove missing values in your dataset.

Feature engineering: Create relevant features like moving averages, relative strength indicators (RSI), or any other indicators that may assist in prediction.

Normalize or scale the data: It's common to scale data to make it suitable for neural networks or other machine learning algorithms**.**

3] Data Splitting:

Split your dataset into a training set and a testing set. A typical split might be 80% for training and 20% for testing.

4] Model Selection:

Choose an appropriate algorithm or model for stock price prediction. Common choices include time series models like ARIMA, machine learning models like regression or decision trees, and deep learning models such as recurrent neural networks (RNNs) or long short-term memory networks (LSTMs).

5] Feature Selection:

Select the most relevant features for your model. You may need to experiment with different combinations of features to determine which ones contribute most to the prediction accuracy.

6] Model Training:

Train your selected model on the training data. Ensure you tune hyperparameters and optimize the model's architecture for best performance. For deep learning models, this might involve adjusting the number of layers, units, and learning rates.

7] Model Evaluation:

Use the testing dataset to evaluate your model's performance. Common evaluation metrics for regression tasks include Mean

Absolute Error (MAE), Mean Squared Error (MSE), and Root Mean Squared Error (RMSE).

8] Fine-Tuning and Iteration:

Based on the evaluation results and backtesting, make necessary adjustments to your model, data, or features. This may involve further training and experimentation**.**

9] Deployment:

If your model performs well, you can consider deploying it in a live trading environment with appropriate risk management measures. Be cautious and aware of the risks associated with automated trading.

10] Monitoring:

Continuously monitor your model's performance in real-time. Markets can change, and models may need periodic updates.

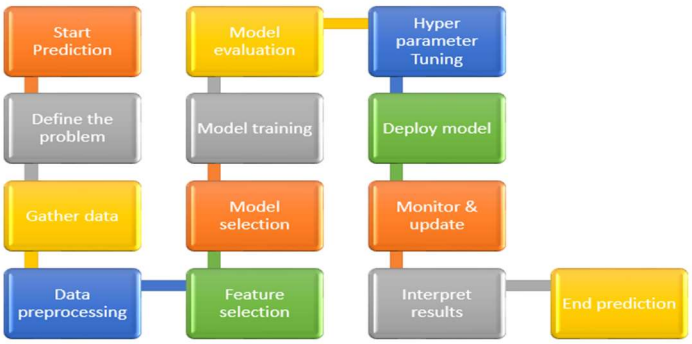
11] Risk Management:

Implement risk management strategies to protect your investments. Don't rely solely on the model's predictions for trading decision.

Metrics used for accuracy check

- **Mean Absolute Error (MAE):** MAE measures the average absolute difference between the predicted and actual prices. It gives you a sense of the model's typical prediction error without considering the direction (overestimation or underestimation).
- **Mean Squared Error (MSE):** MSE calculates the average of the squared differences between the predicted and actual prices. Squaring the errors penalizes larger errors more than MAE, making it sensitive to outliers.

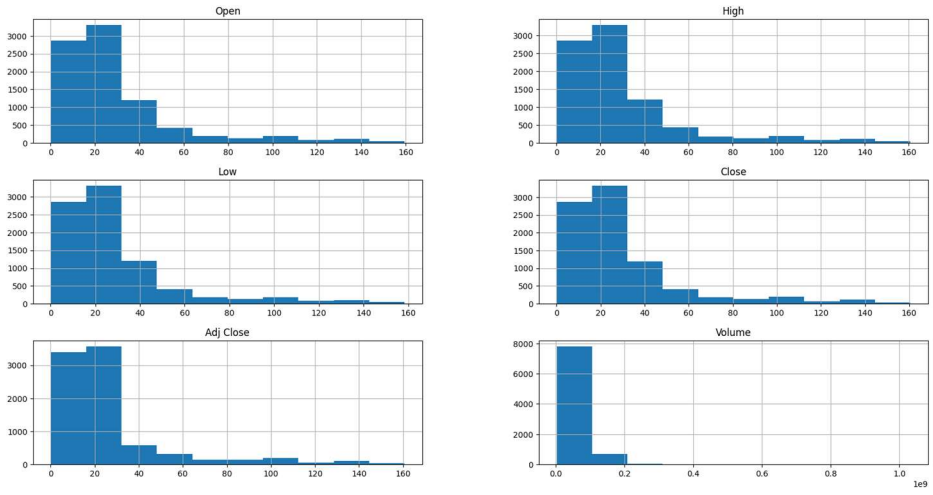
FLOW CHART:



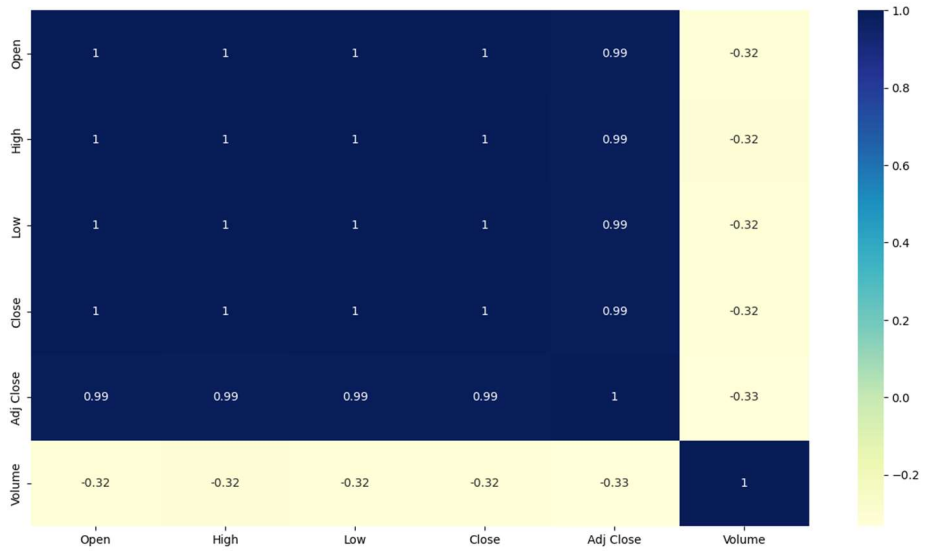
INNOVATION:

- Behavioral Economics: Integrating principles from behavioural economics to understand and model the psychological factors influencing market behaviour.
- Sentiment Analysis: Natural Language Processing (NLP) - Analyzing news articles, social media posts, and financial reports using NLP techniques to gauge market sentiment and incorporating this information into prediction models.
- Alternative Data Sources: Integrating non-traditional data sources such as social media sentiment, satellite imagery, and economic indicators for better-informed predictions.

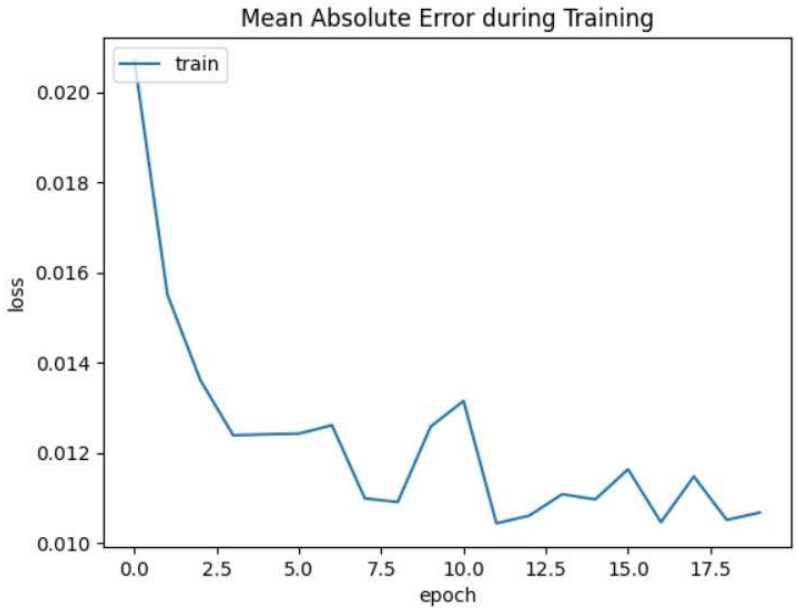
OUTPUT GRAPHS:



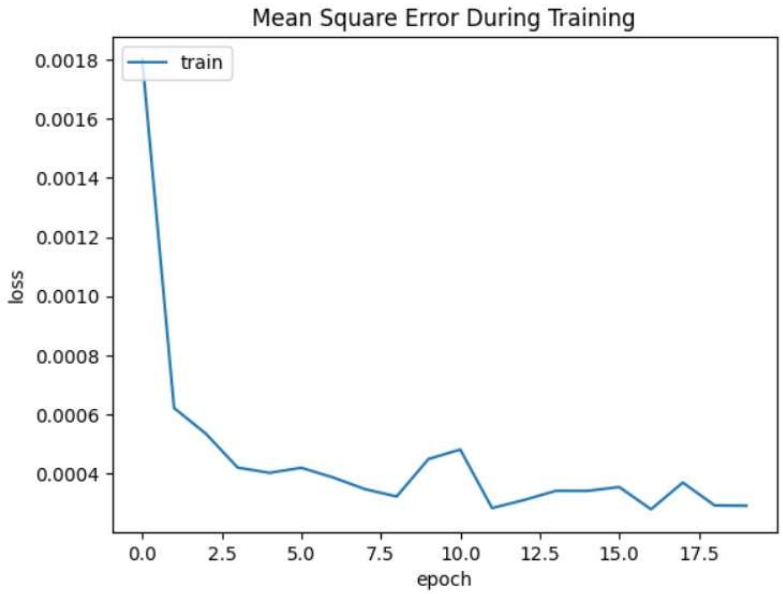
Data Vizualisation



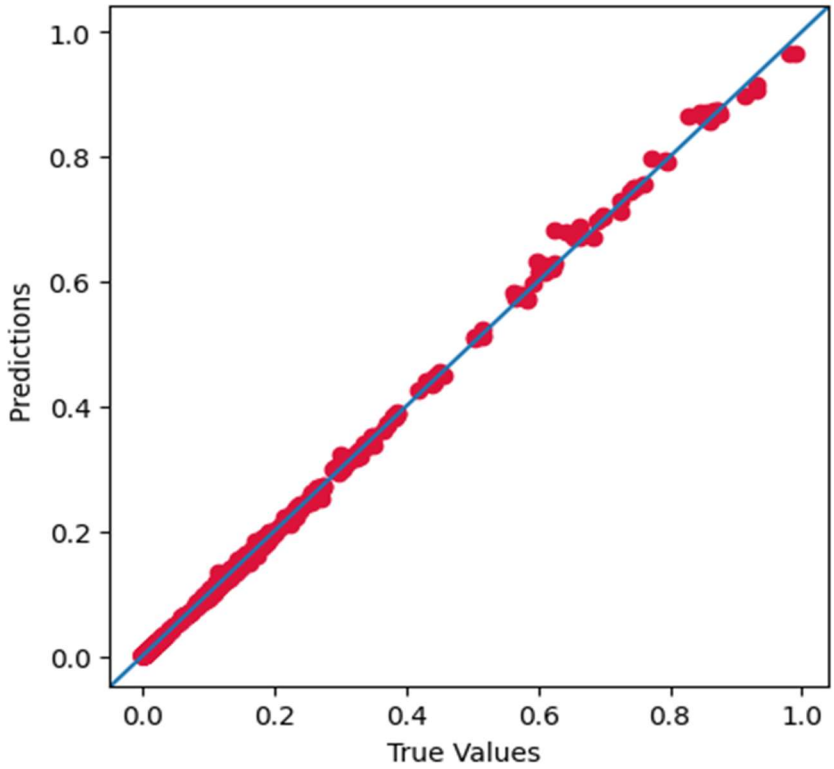
Correlation Matrix



Training Model Loss (MAE)



Training Model Loss (MAE)



ACTUAL Vs PREDICTED