NAAN MUDHALVAN

PHASE 4 PROJECT SUBMISSION

PROJECT 6 - STOCK PRICE PREDICTION

DONE BY:

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
2. Investment Decision Support
3. Risk Management

DATASET DETAIL:

The dataset used for this project is MSFT.csv. **MSFT.csv** contains all the lifetime stock data from 3/13/1986 to 12/10/2019. This dataset contains

seven columns including dates, opening, high, low, closing, adj_close, and volume. LSTMs and Deep Reinforcement Learning agents work well for this dataset.

Link to the Dataset:

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

DETAILS OF THE FEATURES IN THE DATASET:

Date of stock, Opening, High, Low, Closing, Adj_Close, Volume are the columns given in the dataset.

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.

The head of the dataset is below:

	Date	Opening Price	Highest Price	Lowest Price	Closing Price	Adjusted Close Price	Volume
0	1986-03-13	0.088542	0.101563	0.088542	0.097222	0.062549	1031788800
1	1986-03-14	0.097222	0.102431	0.097222	0.100694	0.064783	308160000
2	1986-03-17	0.100694	0.103299	0.100694	0.102431	0.065899	133171200
3	1986-03-18	0.102431	0.103299	0.098958	0.099826	0.064224	67766400
4	1986-03-19	0.099826	0.100694	0.097222	0.098090	0.063107	47894400

ADDITIONAL FEATURES:

The Simple Moving Average (SMA) is a technical indicator that calculates a stock's average price over a specific period, smoothing out data and identifying trends. The Exponential Moving Average (EMA) is another moving average that responds more quickly to price changes. The Relative Strength Index (RSI) measures price movements and can identify overbought or oversold conditions. The Stochastic Oscillator (%K and %D) compares a stock's closing price to its price range, helping traders identify potential reversal points.

Range	Daily Average	Market Capitalization	Target	SMA	EMA	RSI	%K	%D
0.013021	0.093967	1.003126e+08	0.100694	0.096354	0.097222	44.000553	41.17368	44.97231
0.005209	0.099392	3.102986e+07	0.102431	0.096354	0.097853	44.000553	41.17368	44.97231
0.002605	0.101780	1.364086e+07	0.099826	0.096354	0.098686	44.000553	41.17368	44.97231
0.004341	0.101128	6.764849e+06	0.098090	0.096354	0.098893	44.000553	41.17368	44.97231
0.003472	0.098958	4.697962e+06	0.095486	0.096354	0.098747	44.000553	41.17368	44.97231

LIBRARIES USED:

1. **NumPy** is a popular open-source library for numerical and mathematical operations in Python, providing support for working with large, multi-dimensional arrays and matrices of numerical data. It is a fundamental library for scientific and data-intensive computing in Python. To install NumPy, use a package manager like pip or conda.

2. **Pandas** 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.

3. **Matplotlib** is a widely used Python library for creating 2D plots and charts. It allows users 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.

4. **Scikit-learn, also known as sklearn,** is a popular machine learning library in Python that provides a wide range of tools and algorithms for machine learning and data analysis tasks. It 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.

5. **Keras** is an open-source high-level neural networks API written in Python, capable of running on top of other popular deep learning frameworks like TensorFlow and Theano. It provides a userfriendly and modular interface for creating and training deep learning models. Keras can be installed by running. Keras is tightly integrated with TensorFlow and is available as part of TensorFlow as tf.keras. This integration makes it easier to use Keras for deep learning projects because you can install TensorFlow and get Keras as a part of it.

6. **Seaborn** is a popular Python data visualization library built on top of Matplotlib. It provides a high-level interface for creating informative and adaptive statistical graphics, making it particularly useful for visualizing complex datasets and making it easier to understand and interpret data. To install Seaborn, you need to have Python and pip (Python's package manager) installed on your system.

WORKFLOW OF THE CODE:

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:

i) **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.

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

iii) Min Max Scaler: Min-Max scaling is a data preprocessing technique in machine learning that transforms numerical data to a specific range, typically [0, 1]. It maintains relative differences between data points while normalizing them for improved model performance and training.

3. Data Splitting: The test-train split is a common approach in machine learning where a dataset is divided into three subsets: training, validation, and testing. In a 70-20-10 split, 70% of the data is used for training machine learning models, allowing them to learn patterns. The 20% validation set is used for hyperparameter tuning and model evaluation during training, helping to prevent overfitting. The remaining 10% is reserved for testing the final model's performance on unseen data.

4. Model Selection: 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).

LSTM: LSTM is a type of recurrent neural network (RNN) layer in deep learning. It's designed to handle sequences by maintaining memory over long-term dependencies, making it suitable for tasks like natural language processing and time-series forecasting.

Dropout: Dropout is a regularization technique used to prevent overfitting in neural networks. During training, it randomly deactivates a fraction of neurons, forcing the model to learn more robust features and improving its generalization to new data. It helps reduce the risk of the model fitting noise in the training data.

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.

HYPERPARAMETER TUNING

In machine learning, hyperparameter tuning involves adjusting the configuration settings of a model that are not learned from the data but impact the model's performance. It aims to optimize these settings for the best results. For instance, it determines the learning rate, the number of hidden layers, or the dropout rate in neural networks.

Early Stopping: Early stopping is a regularization technique used during training. It halts the training process when a model's performance on a validation dataset stops improving. This helps prevent overfitting, where the model learns the training data too well but doesn't generalize to new data.

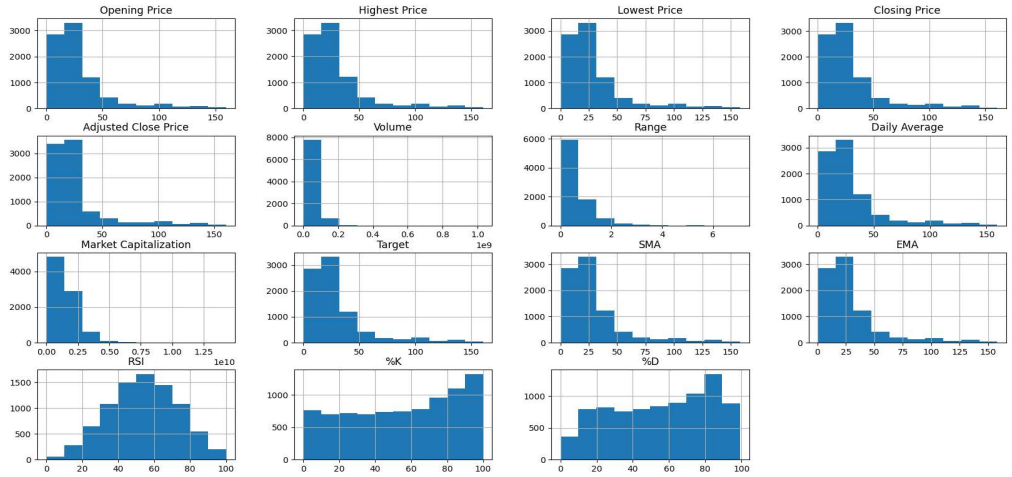
Optimizer (Adam): The optimizer is a key component in training machine learning models. Adam (short for Adaptive Moment Estimation) is a popular optimization algorithm used to adjust the model's weights iteratively to minimize the loss function, making the model converge faster and possibly reach a better solution.

DATA ANALYSIS & VISUALISATION:

Data visualization is the graphical representation of data to reveal patterns, trends, and insights. It simplifies complex information, making it understandable and actionable. Using charts, graphs, and plots, data visualization aids decision-making and communication in various fields, from business analytics to scientific research.

1. **Histogram:**

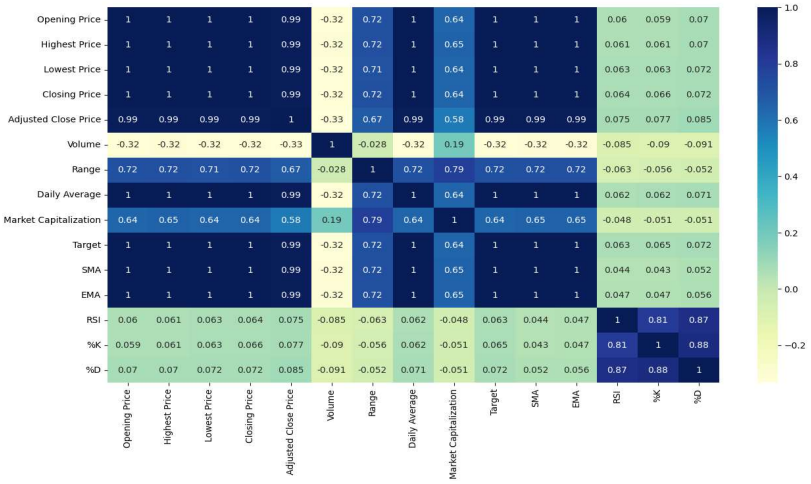
   Histograms visually inspect data distribution, identifying patterns like normal, skewed, or multiple modes. They can reveal outliers or extreme values, and guide preprocessing decisions, such as transforming data if histogram suggests non-normality.

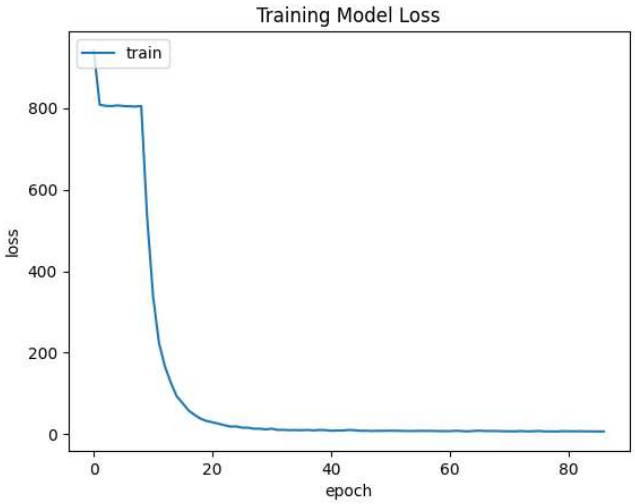


2. **Heat Map:**

   Using Heat Map to visualize complex data patterns, such as identifying highly correlated variables in a correlation matrix.

Dark colours indicates Strong positive correlation. Light colours indicates Weak or no correlation.Red indicates Negative correlation





Training Model Loss

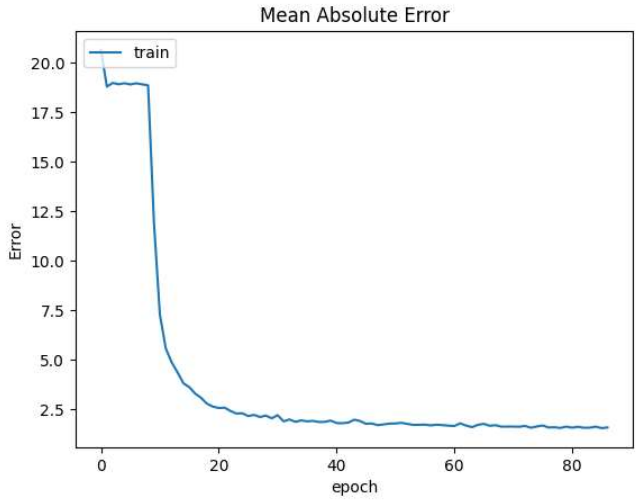
METRICS FOR ACCURACY CHECK

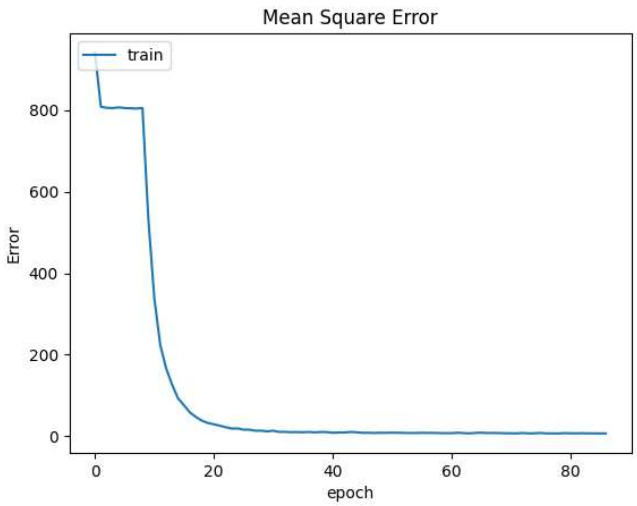
For stock price prediction, commonly used metrics for accuracy evaluation include Mean Absolute Error (MAE), Mean Squared Error (MSE).

Loss Function (MSE and MAE): The loss function quantifies the difference between predicted and actual values during training. Mean Squared Error (MSE) measures the average squared difference, giving higher weight to larger errors. Mean Absolute Error (MAE) measures the average absolute difference, treating all errors equally. They are used to guide the model toward making better predictions.

Validation MAE and Validation MSE: These are the MAE and MSE computed on a separate validation dataset. They help assess the model's performance during training, ensuring it generalizes well to new, unseen data. Lower validation MAE and MSE indicate a better-performing model.

These metrics assess the model's performance in predicting stock prices by measuring the magnitude and direction of prediction errors and the proportion of variance explained**.**





CONCLUSION:

In conclusion, the stock price prediction project has demonstrated the potential for using machine learning techniques to forecast stock prices. Through data preprocessing, model development, and hyperparameter tuning, we have built a predictive model that shows promise.

It's crucial to note that while the model offers predictions based on historical data, the inherent complexity and volatility of financial markets present challenges. Continuous monitoring and refinement are necessary for real-world applications.

This project underscores the importance of data quality, feature engineering, and model evaluation. It serves as a foundation for further research in the field of financial forecasting, offering insights into the possibilities and limitations of stock price prediction.