NAAN MUDHALVAN
PHASE 5 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

Stock price prediction refers to the process of using various techniques, algorithms, and models to forecast the future price movements of stocks in the financial markets. Traders, investors, and analysts use stock price predictions to make informed decisions about buying, selling, or holding stocks. Predicting stock prices accurately is a complex task, and it involves analyzing historical price data, market trends, and other relevant factors.

PROBLEM DEFINITION

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.

PROJECT OBJECTIVES

The objective 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.

CHALLENGES IN STOCK PRICE PREDICTION

- Stock prices are influenced by a wide range of factors, including economic indicators, geopolitical events, and market sentiment, making prediction challenging.

- Random Events Unforeseeable events, such as natural disasters or political unrest, can significantly impact stock prices, making predictions difficult.
- Predictions heavily rely on the quality of historical and real-time data. Inaccurate or incomplete data can lead to unreliable predictions.
- Investments should be made cautiously, and predictions should be considered alongside other forms of analysis and expert opinions

PYTHON DEPENDENCIES OF THE PROJECT

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:

To Install NumPy:

Open a terminal or command prompt and 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 labelled 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.

To Install 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.

To Install Matplotlib:

Open your command prompt or terminal 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.

To Install Scikit-learn:

Open your command prompt or terminal 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:

(1) Keras is tightly integrated with TensorFlow and is available as part of TensorFlow as tf.keras. This integration made it easier to use Keras

for deep learning projects because you could install TensorFlow and get Keras as a part of it.

In the command prompt run:

pip install tensorflow

(2) Keras can also be installed by running:

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.

To install Seaborn:

Open your command prompt or terminal and run the following command

pip install seaborn

TABLE OF CONTENTS

1. DATA COLLECTION:

1.1 DATACARD DETAILS:

1. The details include previous close, open, bid, ask, day's range, 52-week range, volume, average volume, market cap, enterprise value, beta, PE ratio, EPS, earnings date, forward dividend and yield, ex-dividend date, and 1-year target estimate.

2. This data card discusses five valuation metrics for analysing a company's stock price. Market cap is the total market value of outstanding shares of stock, enterprise value is a comprehensive measure of total value, trailing P/E is calculated by multiplying current stock price by total outstanding shares.

3. PEG ratio is a valuation metric that takes into account expected earnings growth rate over the next five years. Price/Sales ratio measures how much investors are willing to pay for each dollar of revenue. P/B ratio measures net asset value.

4. EV/Revenue compares enterprise value to total revenue, EV/EBITDA divides enterprise value by earnings before interest, taxes, depreciation, and amortization.

Trading information includes stock price history, share statistics, dividends and splits, earnings estimates, and revenue estimates. Earnings estimates are predictions of a company's future earnings, often expressed in terms of EPS. EPS revisions can provide insights into market sentiment and the company's growth prospects. ESG

performance refers to how well a company performs in terms of Environmental, Social, and Governance factors. These factors are used by investors, analysts, and stakeholders to assess a company's sustainability and ethical practices.

1.2 DATASET LINK:

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.

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

1.3 GIVEN FEATURES:

Date of stock, Opening, High, Low, Closing, Adj Close, and 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 data set is below:

	Date	Open	High	Low	Close	Adj Close	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

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

2.1 DATA CLEANING:

Forward and backward fill are techniques used in data science and data pre-processing, particularly when dealing with time series data, to handle missing values. These methods help to impute or fill in missing data points in a dataset.

2.1.1 Forward Fill (FFill):

- In the forward fill method, missing values are filled with the most recent preceding value in the dataset.
- This method assumes that the missing data points can be reasonably estimated by carrying forward the last observed value.

- It is often used when dealing with time series data where values are expected to be continuous or slowly changing.

2.1.2 Backward Fill (BFill):

- In the backward fill method, missing values are filled with the next available value in the dataset.
- This method assumes that the missing data points can be estimated by carrying backward the next observed value.
- It can be useful when working with data where future values are more indicative of the missing points.

2.2 CHANGING FEATURE NAMES:

Before adding features, it is advised to change the columns' name. It makes it easier for the user to understand the dataset. So, the column names are changed as follows:

- **Existing Features**: Date, Open, High, Low, Close, Adj Close and Volume.
- **Updated Features**: Date, Opening Price, Highest Price, Lowest Price, Closing Price, Adjusted Close Price and Volume.

The head of the updated dataset (The updated names of Features)

	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

2.3 DATA ANALYSIS BEFORE FATURE ENGINEERING:

Histograms and correlation matrices are essential tools in data analysis for understanding the distribution of data and relationships between variables.

i. Histograms:

A histogram is a graphical representation of a dataset's distribution, showing the frequency of data points within specific intervals. It is useful for understanding the shape, central tendency, and spread of a dataset. Histograms can be used to visually inspect data distribution, identify patterns like normal distribution, skewed distribution, or multiple modes, and reveal outliers or extreme values. Understanding data distribution can guide data preprocessing decisions, such as transforming data if the histogram suggests non-normality.

ii. Correlation Matrix:

A correlation matrix is a table that displays the pairwise correlations between different variables in a dataset. It measures the strength and direction of a linear relationship between two variables and is useful for identifying associations between variables. Positive correlations indicate that when one variable increases, the other tends to increase as well, while negative correlations suggest the opposite. In machine learning and predictive modelling, a correlation matrix helps identify highly correlated features, reducing the dimensionality of the dataset, improving model performance, and reducing overfitting. It also

provides insights into potential cause-and-effect relationships, but does not necessarily imply causation. It can also help address multicollinearity issues in linear regression models.

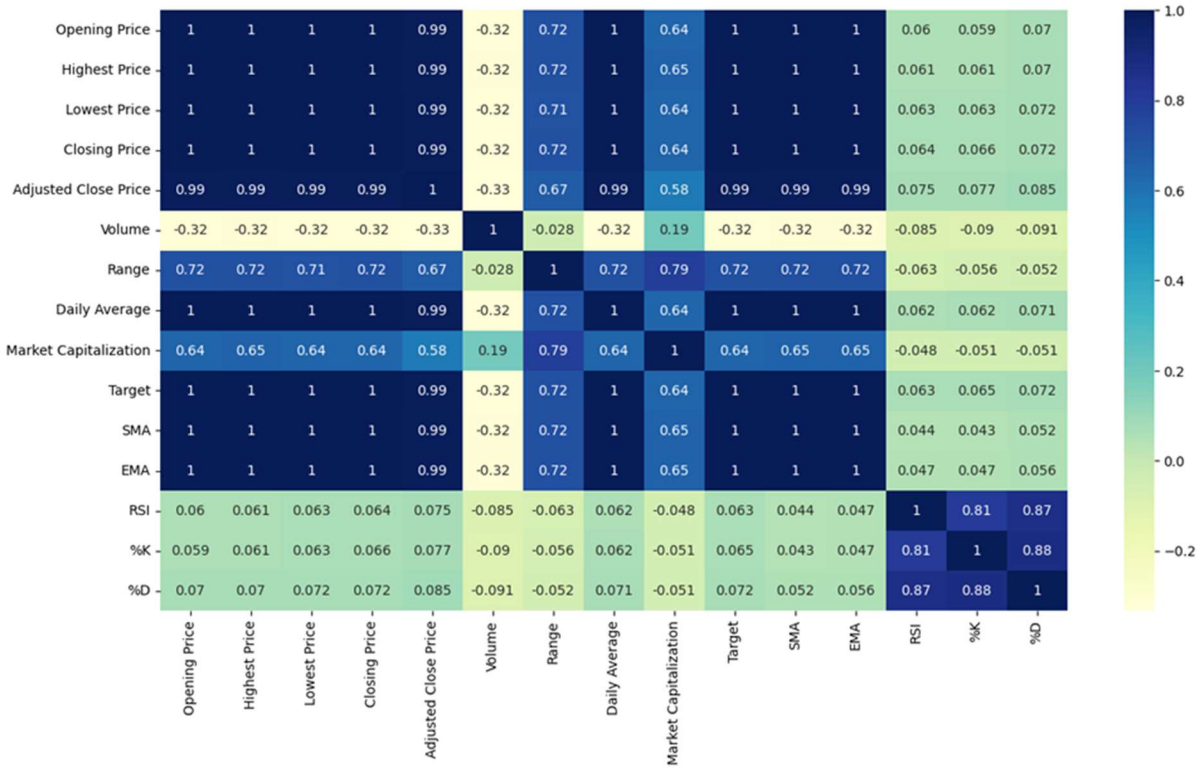
Histograms help us understand the distribution of individual variables, while correlation matrices reveal relationships between pairs of variables.

iii. Heat map:

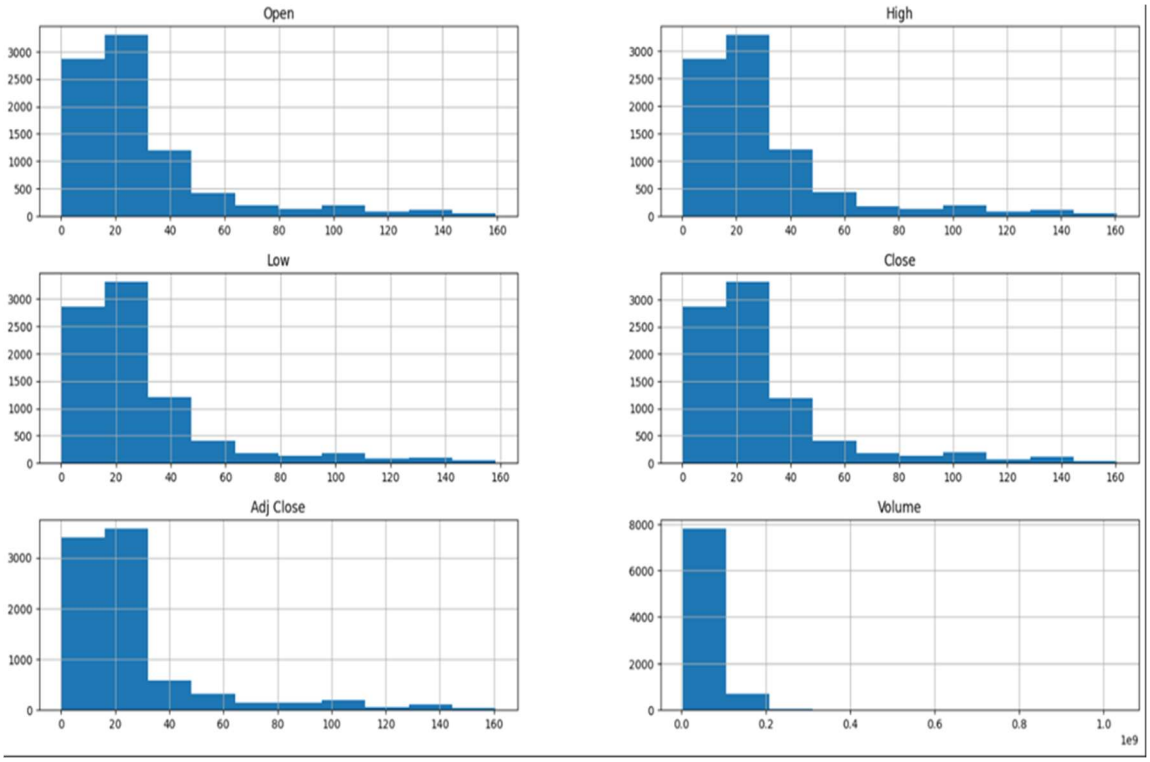
Using Heat Maps to visualize complex data patterns, such as identifying highly correlated variables in a correlation matrix.

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

2.3.1 HEAT MAP FOR THE GIVEN DATA SET:



2.3.2 HISTOGRAM FOR THE GIVEN DATASET:



3. FEATURE ENGINEERING

Feature engineering is a critical step in the process of preparing data for machine learning or predictive modelling. It involves creating new features from the existing data or transforming the data to improve the model's performance. Good feature engineering can lead to more accurate models and better insights.

ADDING FEATURES:

The following features are added in addition to the available features:

- **Range:** The "range" typically refers to the difference between the highest and lowest stock prices during a specific time period. For example, the daily range might be the difference between the highest and lowest prices during a trading day.
- **Daily Average:** Calculate the average price for the day by averaging the opening, high, low, and closing prices
- **Market Capitalization**: Market capitalization, often abbreviated as "market cap," is the total value of a publicly traded company's outstanding shares of stock. Multiply the closing price by the total number of outstanding shares to get the market capitalization
- **Target:** the "target" is the variable we aim to predict. It is typically the stock's future price. This is what our model will try to predict based on historical data and features.
- **SMA (Simple Moving Average):** The Simple Moving Average is a technical indicator that calculates the average price of a stock over a

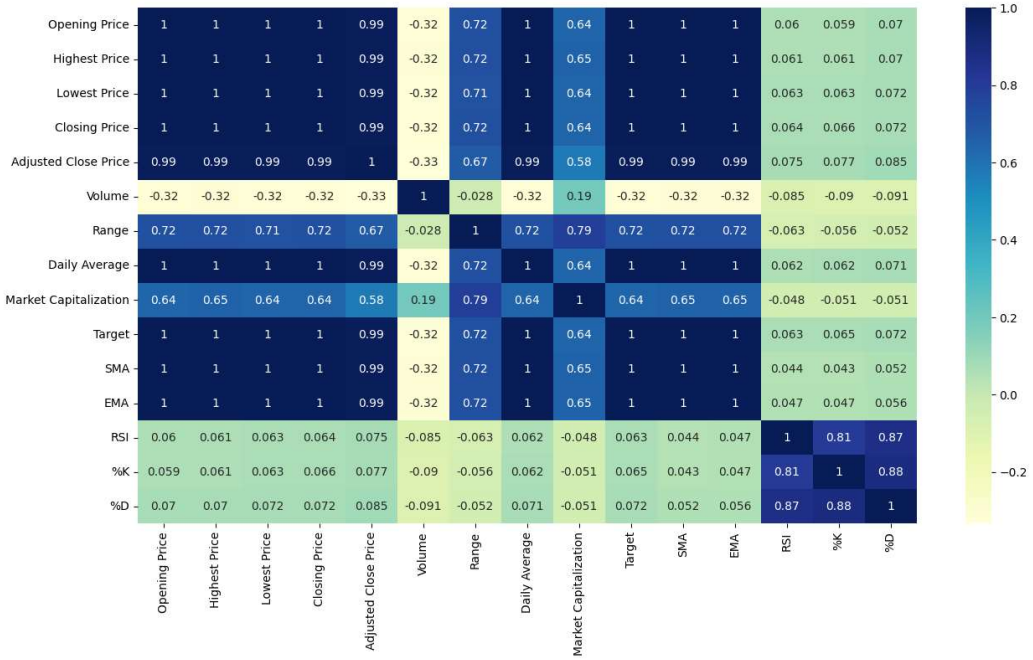
specific period, typically the closing prices. It is used to smooth out price data and identify trends.

- **EMA (Exponential Moving Average):** The Exponential Moving Average is another moving average, but it gives more weight to recent prices. It is calculated using an exponential smoothing formula and is designed to respond more quickly to price changes than the SMA.
- **RSI (Relative Strength Index):** RSI is a momentum oscillator that measures the speed and change of price movements. It ranges from 0 to 100 and is used to identify overbought and oversold conditions in the market. An RSI value above 70 is often considered overbought, while a value below 30 is considered oversold.
- **%K and %D (Stochastic Oscillator):** The Stochastic Oscillator is a momentum indicator that compares the closing price of a stock to its price range over a specific period. %K is the raw measure of momentum, and %D is a smoothed version of %K. It helps traders identify potential reversal points in the market.

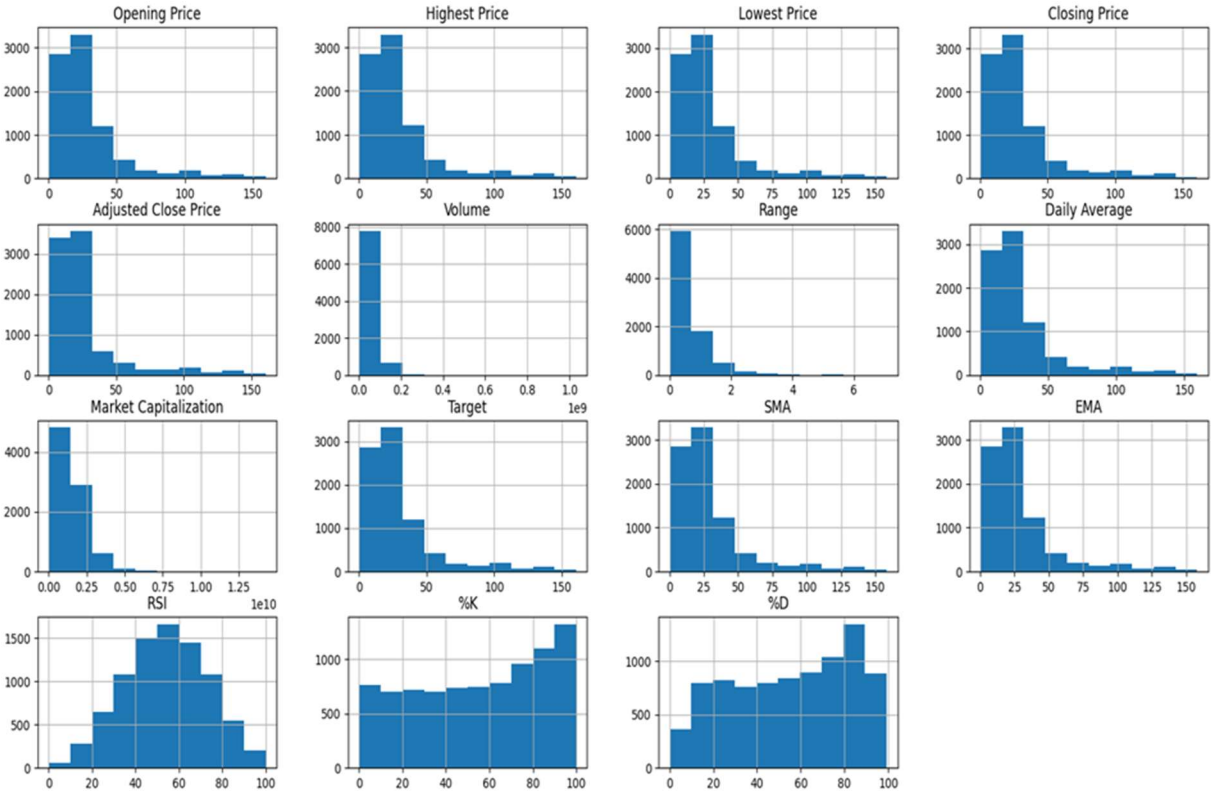
The head of the Updated Data set with additional features after feature Engineering:

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

3.1 HEAT MAP AFTER FEATURE ENGINEERING:

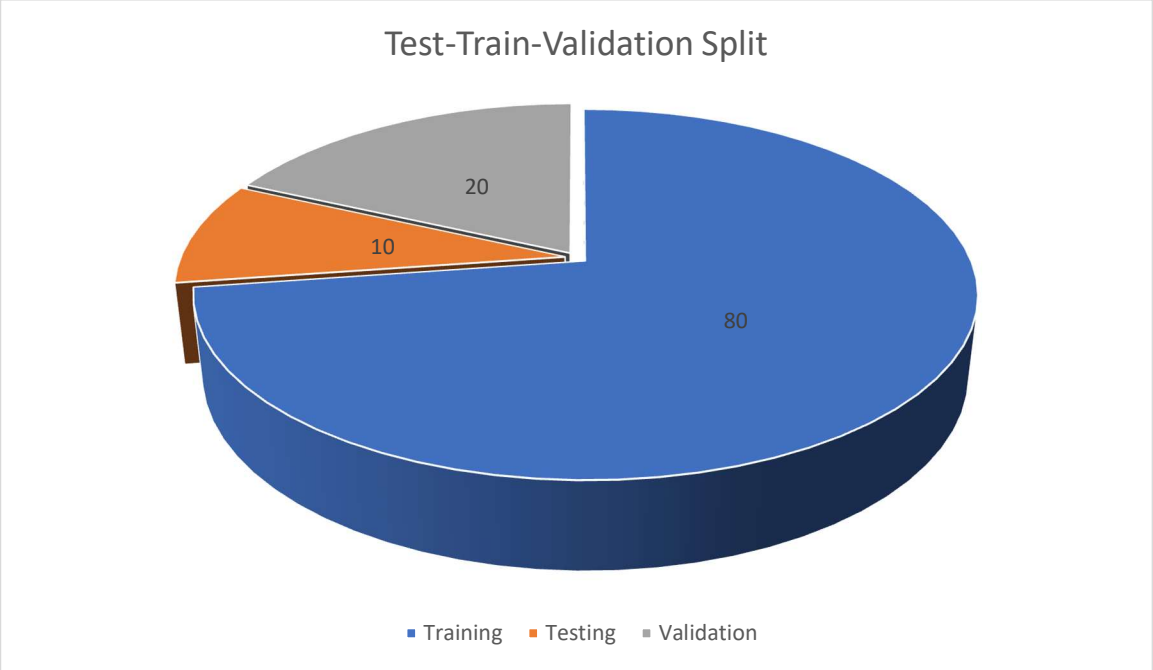


3.2 HISTOGRAM AFTER FEATURE ENGINEERING:



4.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.



5.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.

Random Forest: Random Forest is an ensemble machine learning technique that is widely used for both classification and regression tasks. It is based on decision tree algorithms and combines multiple decision trees to make more accurate predictions.

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.

5.1 LSTM:

- Long Short-Term Memory (LSTM) networks are a type of recurrent neural network (RNN) commonly used in stock price prediction.

- In stock prediction, LSTMs process historical price and volume data, learning from past trends to make future price forecasts.
- They can capture both short-term fluctuations and long-term trends, making them suitable for various investment horizons. LSTMs require careful hyper parameter tuning and feature engineering.
- **STEPS TO APPROACH USING LSTM:**
    - Import the Libraries.
    - Load the Training Dataset.
    - Use the Open Stock Price Column to Train Your Model.
    - Reshape the Data.
    - Building the Model by Importing the Crucial Libraries and Adding Different Layers to LSTM.

5.2 RANDOM FOREST:

- Random Forest is a machine learning algorithm that uses decision trees to make stock price predictions.
- It has several advantages, including ensemble learning, feature importance, non-linearity handling, robustness, flexibility, reduced risk of overfitting, parameter tuning, and prediction uncertainty estimation.
- It combines predictions from multiple decision trees, enhancing accuracy and mitigating overfitting issues.

- It ranks features based on their contribution to prediction, providing insights into factors significantly influencing stock prices.
- Random Forest is also robust against outliers and noise in data, making it suitable for real-world financial data.
- It also offers parameter tuning for optimal performance.

WHAT ADVANTAGES LSTM OFFER?

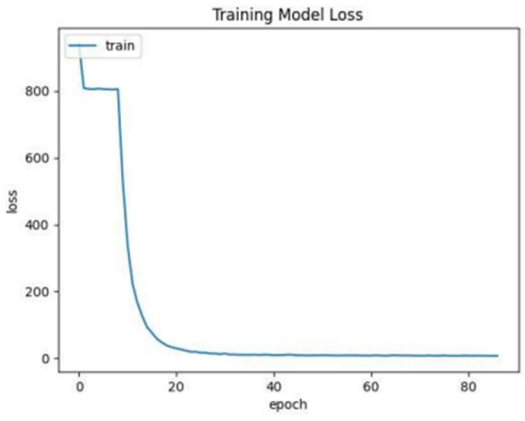
Long Short-Term Memory (LSTM) networks are recurrent neural networks used for stock price prediction due to their sequential data handling, memory of past information, ability to model non-linear relationships, adaptability to irregular time intervals, feature learning, multivariate data handling, robustness to noisy data, long-term predictions, regularization techniques, and model interpretability.

These models are essential for various trading strategies.

6.MODEL TRAINING

In the context of stock price prediction, model training involves preparing historical stock data for machine learning or deep learning algorithms. The process encompasses selecting appropriate features, splitting the data into training and testing sets, and training the model on the training set. This phase also involves tuning model hyperparameters to enhance performance and validating the model's effectiveness in predicting stock prices accurately.

It splits the stock data into three sets: training, validation, and testing. It employs the train_test_split function from the scikit-learn library. Initially, it divides the dataset into training (70%) and a combined temporary set (30%). Then, it further divides the temporary set into validation (1/3 of the initial data) and testing (2/3 of the initial data). The use of random_state=42 ensures reproducibility of the split.



Training Model Loss

6.1 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):

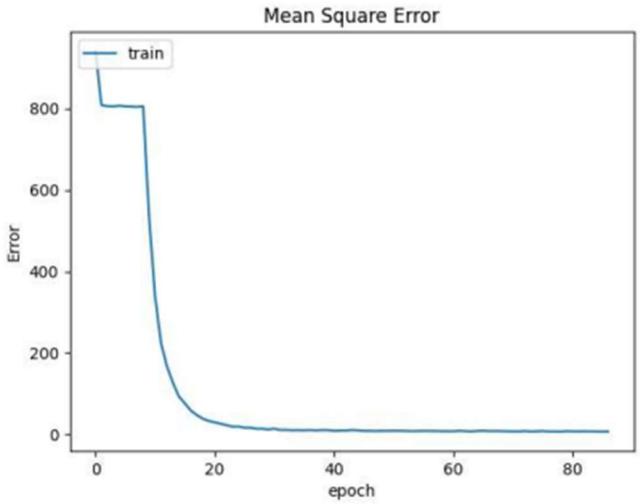
- The optimizer is a key component in training machine learning models.
- Adam (short for Adaptive Moment Estimation) is a popular optimization algorithm used to adjust the model's weights iteratively to minimize the loss function, making the model converge faster and possibly reach a better solution.

7.MODEL EVALUATION

In stock price prediction, model evaluation involves assessing the trained model's accuracy and performance. It includes testing the model on unseen data to measure its predictive capability using metrics like Mean Squared Error (MSE), Root Mean Squared Error (RMSE), or accuracy. Backtesting against historical data also validates the model's effectiveness in a real-world scenario. Comparison with benchmarks or alternative models helps verify the model's predictive power. Fine-tuning and iterating based on evaluation results are essential to ensure the model's reliability before deploying it for making stock price predictions.

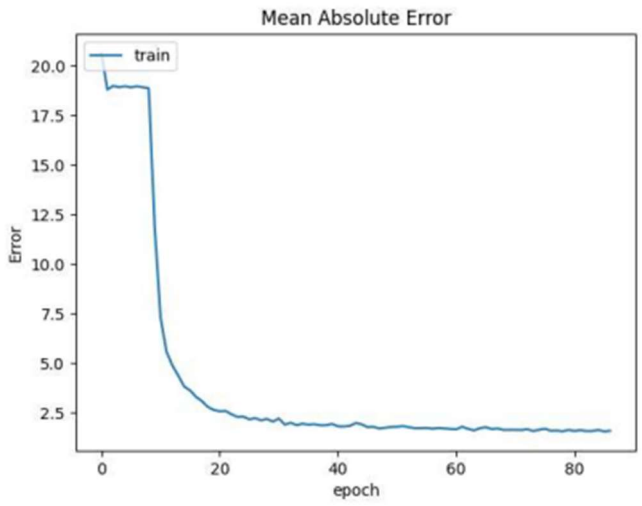
7.1 MEAN SQUARED ERROR(MSE)

- Mean Squared Error (MSE) quantifies the average squared difference between predicted and observed values.
- By calculating the squared differences and averaging them, MSE offers a measure of a model's accuracy in prediction.
- It emphasizes larger errors due to the squaring effect, making it sensitive to outliers or significant deviations.
- Lower MSE values indicate a better fit between the model's predictions and the actual data, making it a widely used metric in evaluating the performance of predictive models in various fields, including finance, machine learning, and statistical analysis



7.2 MEAN ABSOLUTE ERROR(MAE):

- Mean Absolute Error (MAE) calculates the average of absolute differences between predicted and observed values.
- MAE offers a straightforward measure of a model's prediction accuracy, giving equal weight to all errors.
- This metric is easy to interpret and is suitable for situations where large errors should not be overly emphasized.
- MAE is commonly used in various fields, including finance, machine learning, and statistics, to assess the performance of predictive models and understand the average magnitude of errors between predictions and actual values.



7.3 WHY MSE AND MAE?

Mean Squared Error (MSE):

Sensitivity to Large Errors: MSE penalizes large errors more significantly due to the squaring effect, which can be helpful in certain contexts to focus on significant deviations.

Differentiable and Mathematical Convexity: Its differentiable nature is advantageous in optimization algorithms and model training. Also, the convexity of MSE makes it easier to find the minimum.

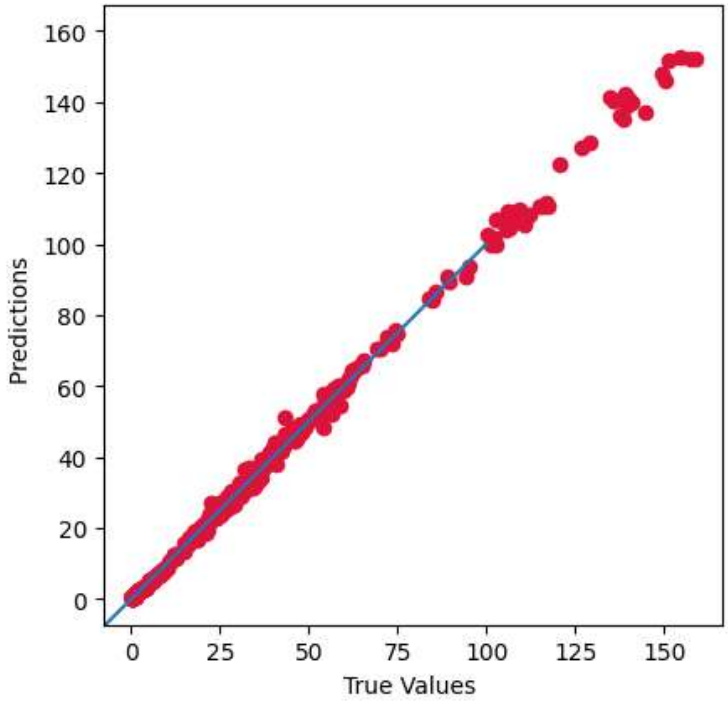
Mean Absolute Error (MAE):

Robustness to Outliers: MAE is less sensitive to outliers compared to MSE because it measures absolute errors. This can make it more robust in situations where outliers are present in the data.

Simplicity and Interpretability: MAE is more straightforward to interpret as it directly measures the average magnitude of errors, making it more intuitive and user-friendly in some contexts.

Both metrics have their advantages, with MSE emphasizing larger errors and being mathematically convenient, while MAE offers robustness against outliers and simplicity in interpretation. The choice between them often depends on the specific characteristics of the data and the objectives of the modeling task.

OUTPUT:



CONCLUSION:

In the realm of stock price prediction, the project involved training a neural network model using historical stock data. The meticulous optimization of parameters aimed at minimizing error was evident.

Visual analysis depicted a consistent reduction in loss over training epochs, indicating improving convergence.Both Mean Absolute Error (MAE) and Mean Squared Error (MSE) plots revealed a declining trend, reflecting the model's capacity to diminish prediction errors gradually.

However, to ensure its applicability in real-world scenarios, further validation with unseen data and continuous refinement are imperative. These measures will bolster the model's predictive accuracy, thereby fortifying its reliability for precise stock price forecasting.