NAAN MUDHALVAN

PHASE 3 - 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.

PHASE OBJECTIVE:

In this part, we aim to begin building our project by loading and pre-processing the dataset by collecting and pre-processing the historical stock market data for analysis.

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 Pre-processing:

- **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**.**

DATA COLLECTION

The data set used for this project is MSFT.csv, which is an open-source dataset available in Kaggle. 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

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

FEATURES IN THE GIVEN DATASET

The given data set had the following features:

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 is 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 is 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

DATA PREPROCESSING

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

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

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.

- **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.
- **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.
  - The choice between forward fill and backward fill depends on the specific context of the data and the problem you are trying to solve. Sometimes, a combination of both methods may be used to fill missing values more effectively. It's important to consider the nature of the data and how these techniques might impact the analysis or modelling process.

DATA ANALYSIS:

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 modeling, 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.

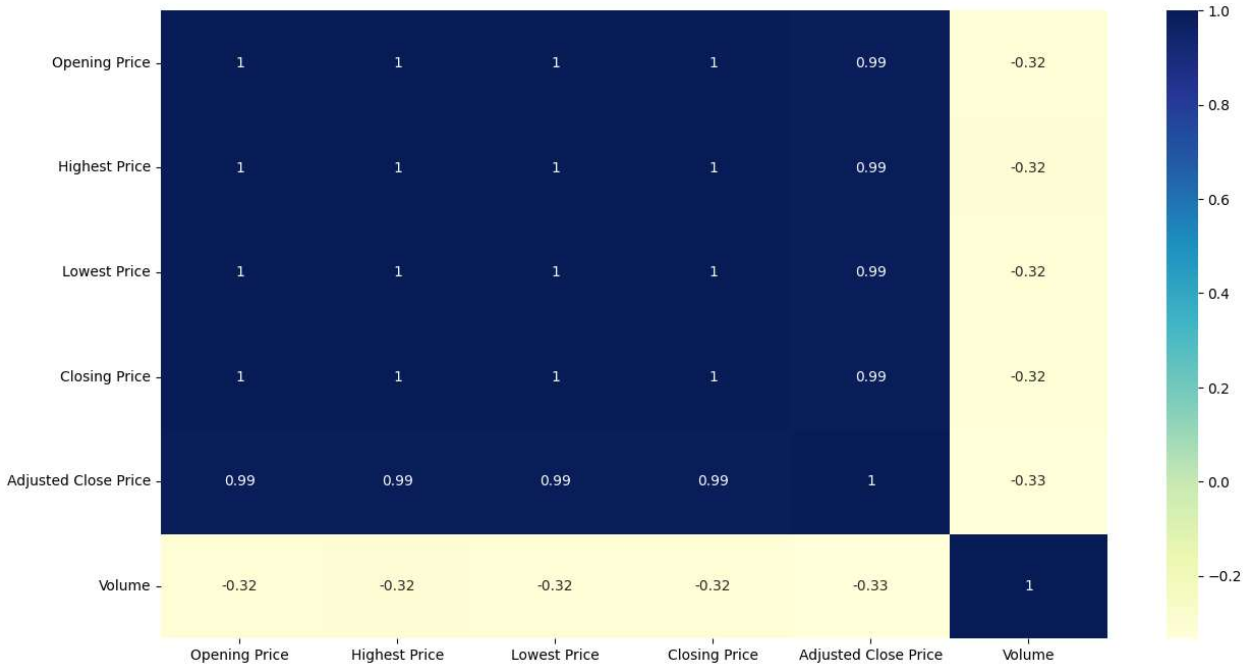
Histograms help us understand the distribution of individual variables, while correlation matrices reveal relationships between pairs of variables.

iii. Heat map:

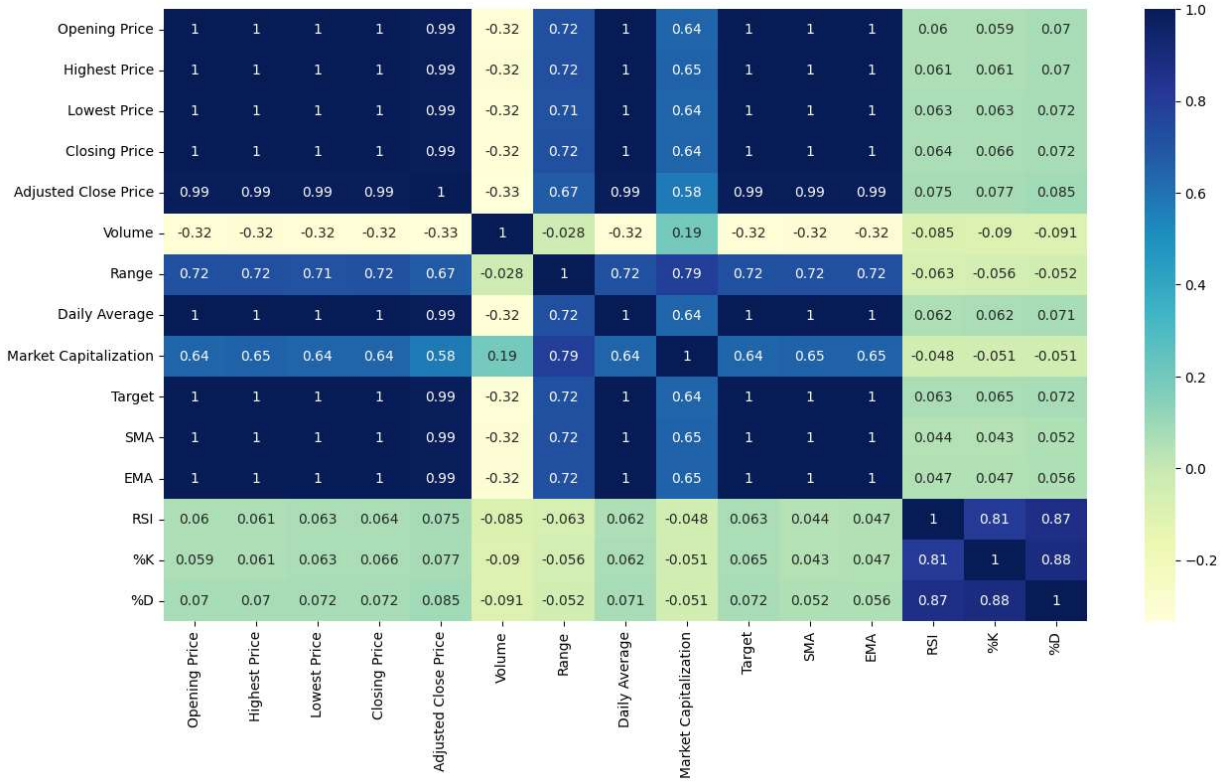
Using Heat Maps to visualize complex data patterns, such as identifying highly correlated variables in a correlation matrix.

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

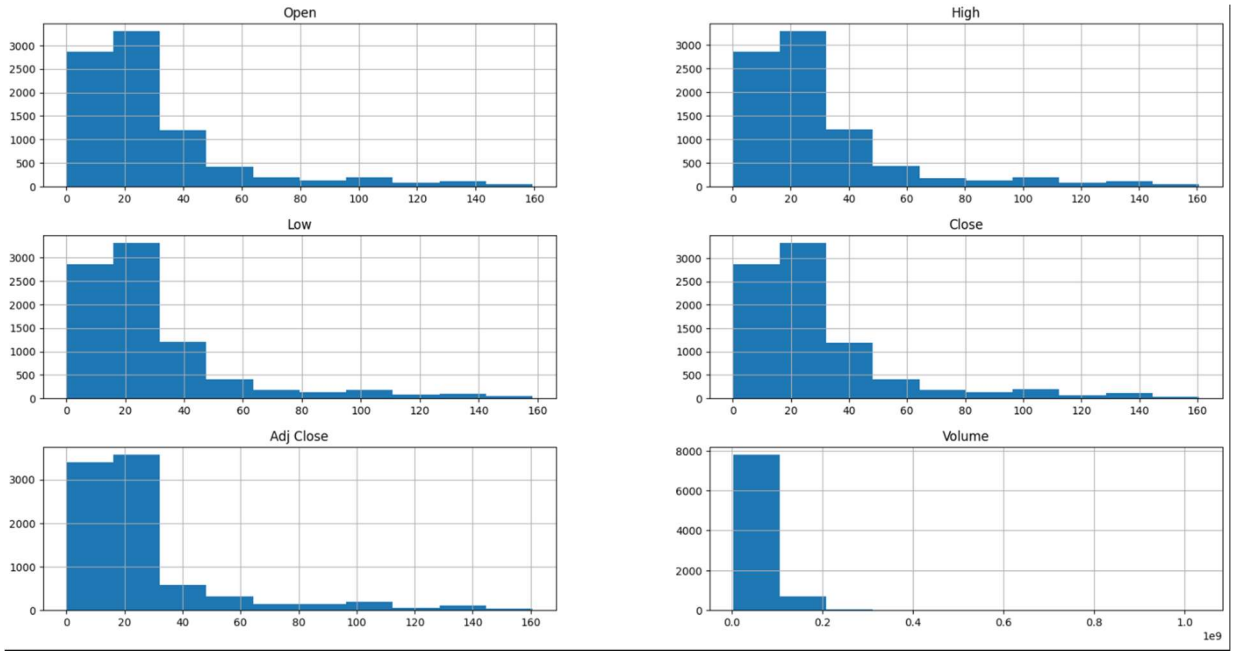
Heat Map for the Given Dataset (Before pre-processing):



Heat Map for the Given Dataset (After pre-processing):



Histogram for the Given Dataset (Before Pre-processing):



Histogram for the Given Dataset (After Pre-processing):