STOCK PRICE PREDICTION

INTRODUCTION:

Stock price prediction is a vital and challenging task in the field of finance and data science. Stock price prediction is the process of using historical data, mathematical models and often advanced algorithms to estimate the future value of the particular stock or financial asset. By here we can endeavor to forecast the future values of a particular stock or financial gains, but they are also inherently uncertain due to the myriad factors including financial markets.

OBJECTIVE:

1.Historical Data:

Historical stock price data, which includes attributes like open, low, high, closing prices, and trading volume, accurate stock prices, etc ,…analyzing this data allows for the identification of trends and patterns.

2.Market Dynamics:

Stock prices are influenced by market dynamics, news events, investor sentiment, and a host of external factor such as economics reports , and accurate prices.

3.Model Complexity:

There are a multitude of predictive modeling techniques ranging from traditional statistical models to advanced deep learning and machine learning algorithms. complex models can capture nuanced relationship in data, but they also require vast amount of historical data and computational resources

4.Dataset:

Kaggle is an online platform and community for data science and machine learning enthusiasts. It provides access to a vast repository of datasets, tools for running code and analyses, opportunities to compete in data science challenges, and a platform for sharing knowledge and collaborating with a global community of data scientists. Kaggle is a valuable resource for learning, practicing, and advancing skills in data science and machine learning.

5.Details about columns:

In stock price prediction dataset, you typically encounter several columns that provide various pieces of information about a particular stock or financial

asset. While the exact columns may vary depending on the data source and how it's structured. Here is a common set of columns you can find in such a dataset: Date, Open, High, Low, Close, Volume, Adjusted close(Adj Close), Dividends, Stock Splits, Material Capitalization, Earning per share, etc,… In my project I was used the columns are Date, Open, Close, Adj close and volume.

6. Details about libraries used in project:

To work on my project , I just used python libraries for data manipulation, analysis, machine learning, and visualization. Here's a list of commonly used libraries along with the way to download and install them using pip.

Several python libraries are used in my project that is NumPy (Numerical Python), pandas, Matplotlib, seaborn, Scikit- Learn , TensorFlow or PyTorch.

7. How to train and test:

Training and testing a stock price prediction model involves several key steps to ensure the model's accuracy and generalization.

Data Collection: Obtain historical stock price data, typically in the form of OHLCV (Open, High, Low, Close, Volume) and other relevant features. You may also collect news sentiment data or other external factors that could influence stock prices.

Data Preprocessing: Clean the data by handling missing values, outliers, and any inconsistencies in the dataset. Normalize or scale the data to ensure all features are on a similar scale. This is crucial for many machine learning models.

Data Splitting: Split the data into training, validation, and test sets. The training set is used to train the model, the validation set helps tune hyperparameters and prevent overfitting, and the test set is for evaluating the model's performance.

Model Selection: Choose an appropriate model for stock price prediction. Common choices include linear regression, time series models, machine learning models (e.g., Random Forest, Support Vector Machines), or deep learning models (e.g., LSTM or CNN-LSTM).

Model Training: Train the selected model using the training dataset. The model learns to make predictions based on historical data.

Testing: After tuning the model using the validation set, evaluate its performance on the test dataset to assess how well it generalizes to new, unseen data.

Visualization: Create visualizations to compare the model's predictions with actual stock prices. This helps in understanding how well the model performs.

Code and Output:

Importing the libraries:

```
from mpl_toolkits.mplot3d import Axes3D
from sklearn.preprocessing import StandardScaler
import matplotlib.pyplot as plt
import numpy as np
import os
import pandas as pd
```

```
for dirname, _, filenames in os.walk('/kaggle/input'):
    for filename in filenames:
        print(os.path.join(dirname, filename))
```

output:

```
/kaggle/input/MSFT.csv
```

```
nRowsRead = 1000 # specify 'None' if want to read whole file
# MSFT.csv may have more rows in reality, but we are only loading/preview
ing the first 1000 rows
df1 = pd.read_csv('/kaggle/input/MSFT.csv', delimiter=',', nrows = nRow
sRead)
df1.dataframeName = 'MSFT.csv'
nRow, nCol = df1.shape
print(f'There are {nRow} rows and {nCol} columns')
```

Output:

```
There are 1000 rows and 7 columns
```

```
df1.head(5)
```

Output:

	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

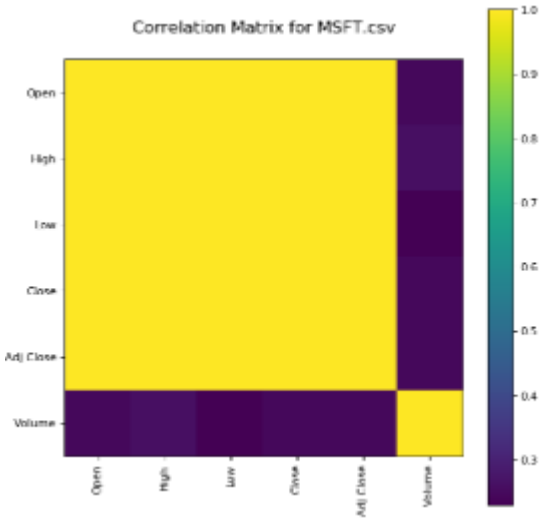
```
plotPerColumnDistribution(df1, 10, 5)
```

Output:

```
<Figure size 2400x512 with 0 Axes>
```

```
plotCorrelationMatrix(df1, 8)
```

Output:



```
plotScatterMatrix(df1, 18, 10)
```

Output: