IBM NAAN MUDHALVAN

PHASE-3 PROJECT SUBMISSION

DOMAIN:	Applied Data Science
PROJECT TITLE:	Stock Price Prediction
TEAM MEMBERS:	Ganesh N (420421104019) Sanjay p (420421104067) Tamilselvan A (420421104083) Guganesh R (420421104025)

Project Description:

Analyzing Microsoft Lifetime Stocks,

In this project, we will analyze the "Microsoft Lifetime Stocks Dataset" to gain insights into Microsoft's stock performance over the years. We aim to understand the historical trends, conduct predictive analysis, and make data-driven decisions based on the provided data.

Dataset Information:

Dataset Source: [Microsoft Lifetime Stocks Dataset on Kaggle](https://www.kaggle.com/datasets/prasoonkottarathil/microsoft-lifetime-stocks-dat)

Data Type: CSV

Tools and Technologies Used:

- Python
- Jupyter Notebook
- Pandas
- Matplotlib

Data Loading and Preprocessing

1. Data Loading:

- Describe how the dataset was loaded into your project.

- Specify any challenges encountered during this process.

- **IMPORT LIBRARIES:**

```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
```

- **READ DATASET:**

```
[3]: df=pd.read_csv("MSFT.csv")
     df
```

[3]:

	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
...	...	...	...	...	...	...	...
8520	2019-12-31	156.770004	157.770004	156.449997	157.699997	157.699997	18369400
8521	2020-01-02	158.779999	160.729996	158.330002	160.619995	160.619995	22622100
8522	2020-01-03	158.320007	159.949997	158.059998	158.619995	158.619995	21116200
8523	2020-01-06	157.080002	159.100006	156.509995	159.029999	159.029999	20813700
8524	2020-01-07	159.320007	159.669998	157.330002	157.580002	157.580002	18017762

8525 rows × 7 columns

2. Data Preprocessing:

- Detail the preprocessing steps taken (e.g., handling missing values, data normalization, encoding categorical variables).

- Explain the rationale behind each preprocessing step.

HANDLING MISSING VALUES:

- CHECK THE NULL VALUES IN THE DATA SET

```
: df.isnull()
```

:

	Date	Open	High	Low	Close	Adj Close	Volume
0	False	False	False	False	False	False	False
1	False	False	False	False	False	False	False
2	False	False	False	False	False	False	False
3	False	False	False	False	False	False	False
4	False	False	False	False	False	False	False
...	...	...	...	...	...	...	...
8520	False	False	False	False	False	False	False
8521	False	False	False	False	False	False	False
8522	False	False	False	False	False	False	False
8523	False	False	False	False	False	False	False
8524	False	False	False	False	False	False	False

8525 rows × 7 columns

- ## REMOVE OR REPLACE THE NULL VALUES

```
df.dropna()
```

	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
...	...	...	...	...	...	...	...
8520	2019-12-31	156.770004	157.770004	156.449997	157.699997	157.699997	18369400
8521	2020-01-02	158.779999	160.729996	158.330002	160.619995	160.619995	22622100
8522	2020-01-03	158.320007	159.949997	158.059998	158.619995	158.619995	21116200
8523	2020-01-06	157.080002	159.100006	156.509995	159.029999	159.029999	20813700
8524	2020-01-07	159.320007	159.669998	157.330002	157.580002	157.580002	18017762

8525 rows × 7 columns

- ## DESCRIBE THE DATASET:

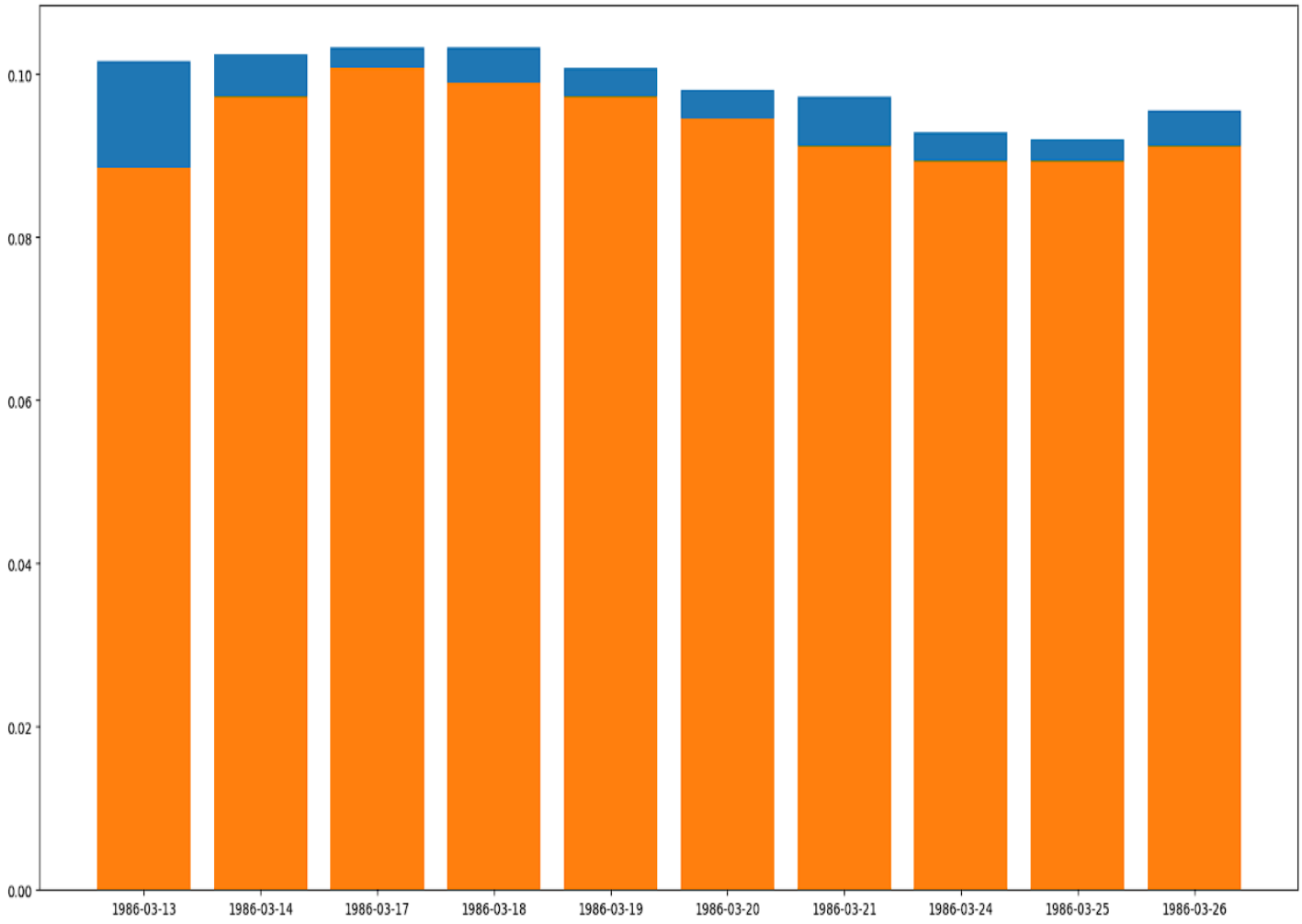
```
df.describe()
```

	Open	High	Low	Close	Adj Close	Volume
count	8525.000000	8525.000000	8525.000000	8525.000000	8525.000000	8.525000e+03
mean	28.220247	28.514473	27.918967	28.224480	23.417934	6.045692e+07
std	28.626752	28.848988	28.370344	28.626571	28.195330	3.891225e+07
min	0.088542	0.092014	0.088542	0.090278	0.058081	2.304000e+06
25%	3.414063	3.460938	3.382813	3.414063	2.196463	3.667960e+07
50%	26.174999	26.500000	25.889999	26.160000	18.441576	5.370240e+07
75%	34.230000	34.669998	33.750000	34.230000	25.392508	7.412350e+07
max	159.449997	160.729996	158.330002	160.619995	160.619995	1.031789e+09

SIMPLE VISUALIZATION OF DATA SET:

BAR PLOT:

```
x=df['Date'].head(10)
y=df['High'].head(10)
y1=df['Low'].head(10)
plt.figure(figsize=(20,10))
plt.bar(x,y ,label='HIGH')
plt.bar(x,y1, label='LOW')
plt.show()
```



LINE PLOT:

```
x=df['Date'].head(10)
y=df['High'].head(10)
y1=df['Low'].head(10)
plt.figure(figsize=(20,10))
plt.plot(x,y ,label='HIGH')
plt.plot(x,y1, label='LOW')
plt.show()
```