# STOCK PRICE PREDICTION USING APPLIED DATA SCIENCE

## INTRODUCTION

The stock market is a dynamic and complex system influenced by a multitude of factors,including economic indicators, news events, investor sentiment, and market trends. Predicting its movements has been a subject of great interest for investors, traders, and researchers alike.Applied Data Science provides a powerful toolkit to analyse historical data, extract meaningful patterns, and build models to make informed predictions about future market behaviour.The goals of this project have been outlined, a data analysis plan has been developed, useful visualisation types have been identified, and Python and data visualisation libraries are being used for the analysis in “Jupyter Notebook”.

## DESCRIPTION:

This stage focuses on prepping the provided CSV file in order to prepare it for other operations including analysis, exploratory data analysis, and dataset visualisation. Using the "ANALYSE Tool" to visualise the data is part of phase 3 of the study "STOCK PRICE PREDICTION". This document includes a subplot, bar chart, line chart, and boxplot among other charts

## Dataset

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

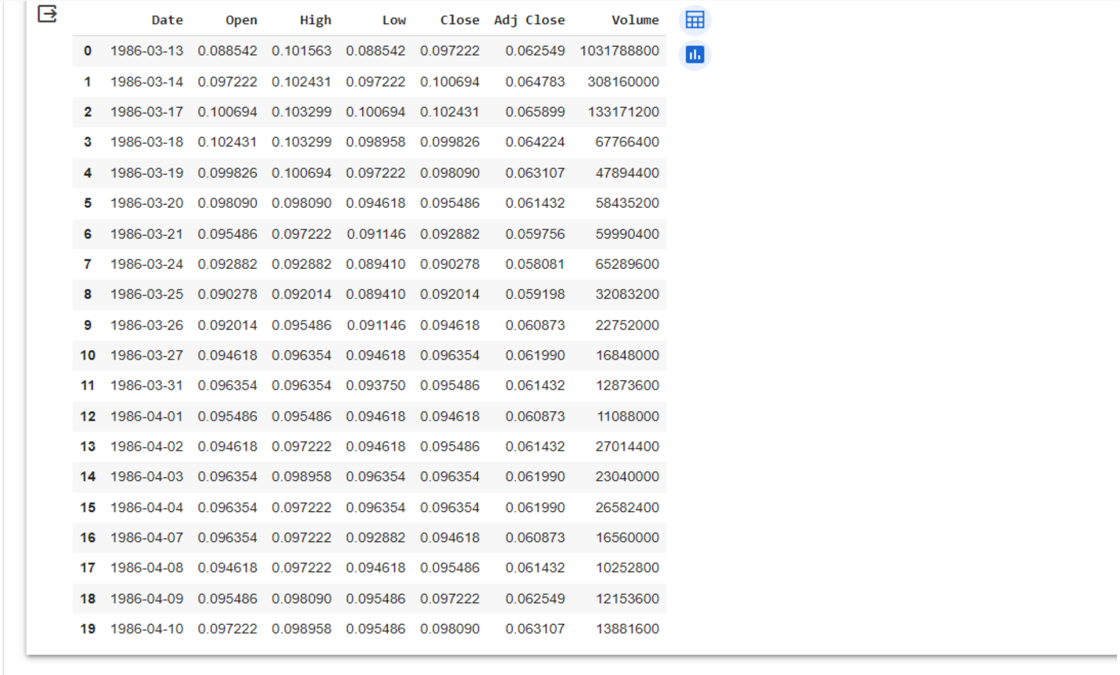
## Implementation

## Load the dataset

Load the dataset a csv file from kaggle and using Pandas package load them

```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sb
warnings.filterwarnings('ignore')
df = pd.read_csv('stock prediction.csv')
df.head(20)
```

## 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 |
| 5 | 1986-03-20 | 0.098090 | 0.098090 | 0.094618 | 0.095486 | 0.061432 | 58435200 |
| 6 | 1986-03-21 | 0.095486 | 0.097222 | 0.091146 | 0.092882 | 0.059756 | 59990400 |
| 7 | 1986-03-24 | 0.092882 | 0.092882 | 0.089410 | 0.090278 | 0.058081 | 65289600 |
| 8 | 1986-03-25 | 0.090278 | 0.092014 | 0.089410 | 0.092014 | 0.059198 | 32083200 |
| 9 | 1986-03-26 | 0.092014 | 0.095486 | 0.091146 | 0.094618 | 0.060873 | 22752000 |
| 10 | 1986-03-27 | 0.094618 | 0.096354 | 0.094618 | 0.096354 | 0.061990 | 16848000 |
| 11 | 1986-03-31 | 0.096354 | 0.096354 | 0.093750 | 0.095486 | 0.061432 | 12873600 |
| 12 | 1986-04-01 | 0.095486 | 0.095486 | 0.094618 | 0.094618 | 0.060873 | 11088000 |
| 13 | 1986-04-02 | 0.094618 | 0.097222 | 0.094618 | 0.095486 | 0.061432 | 27014400 |
| 14 | 1986-04-03 | 0.096354 | 0.098958 | 0.096354 | 0.096354 | 0.061990 | 23040000 |
| 15 | 1986-04-04 | 0.096354 | 0.097222 | 0.096354 | 0.096354 | 0.061990 | 26582400 |
| 16 | 1986-04-07 | 0.096354 | 0.097222 | 0.092882 | 0.094618 | 0.060873 | 16560000 |
| 17 | 1986-04-08 | 0.094618 | 0.097222 | 0.094618 | 0.095486 | 0.061432 | 10252800 |
| 18 | 1986-04-09 | 0.095486 | 0.098090 | 0.095486 | 0.097222 | 0.062549 | 12153600 |
| 19 | 1986-04-10 | 0.097222 | 0.098958 | 0.095486 | 0.098090 | 0.063107 | 13881600 |

## DATA PREPROCESSING

Check the data reductance and null or missing value in the given dataset and preprocess them and check the null value and remove them if any.

```
df.isnull().sum()
```

OUTPUT

```
Date         0
Open          0
High         0
Low          0
Close        0
Adj Close    0
Volume       0
dtype: int64
```

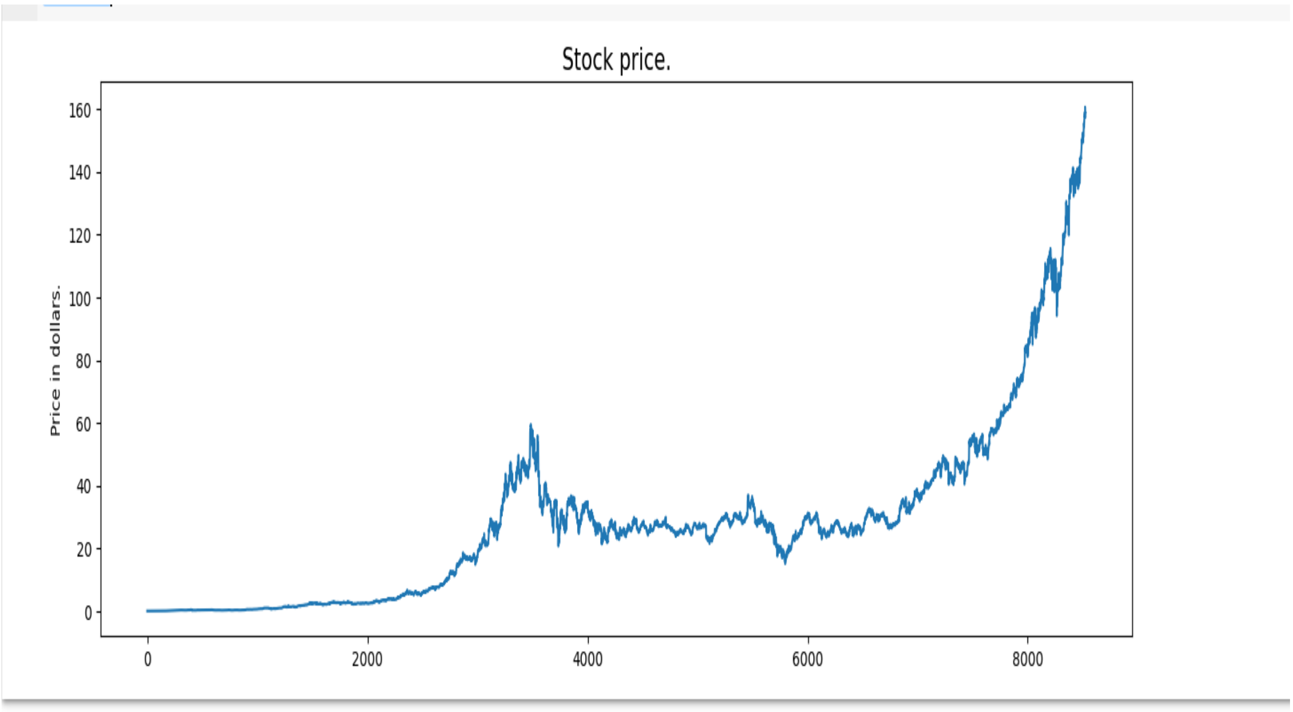
# ANALYSIS

1.PLOT FUNCTION -For plotting the stock price in dollar which are close in the given dataset

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

df = pd.read_csv('stock prediction.csv')
plt.figure(figsize=(15,5))
plt.plot(df['Close'])
plt.title('Stock price.', fontsize=15)
plt.ylabel('Price in dollars.')
plt.show()
```

OUTPUT

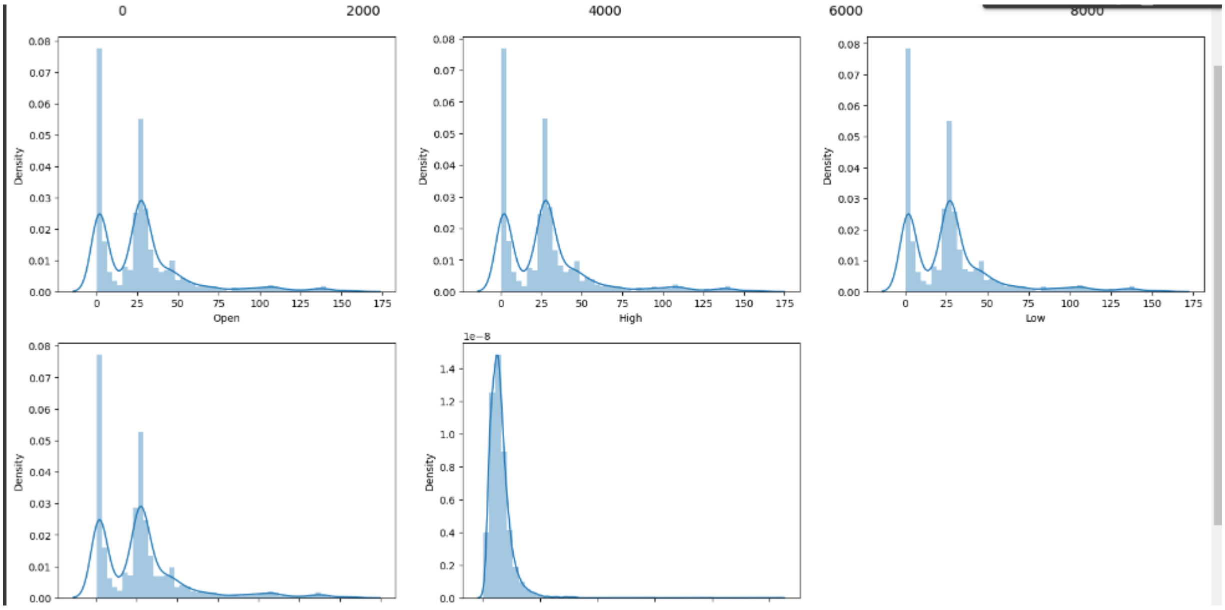


## 2.Displot &Subplot

```
plt.subplots(figsize=(20,10))

features = ['Open', 'High', 'Low', 'Close', 'Volume']
for i,col in enumerate(features):
  plt.subplot(2,3,i+1)
  sb.distplot(df[col])
plt.show()
```

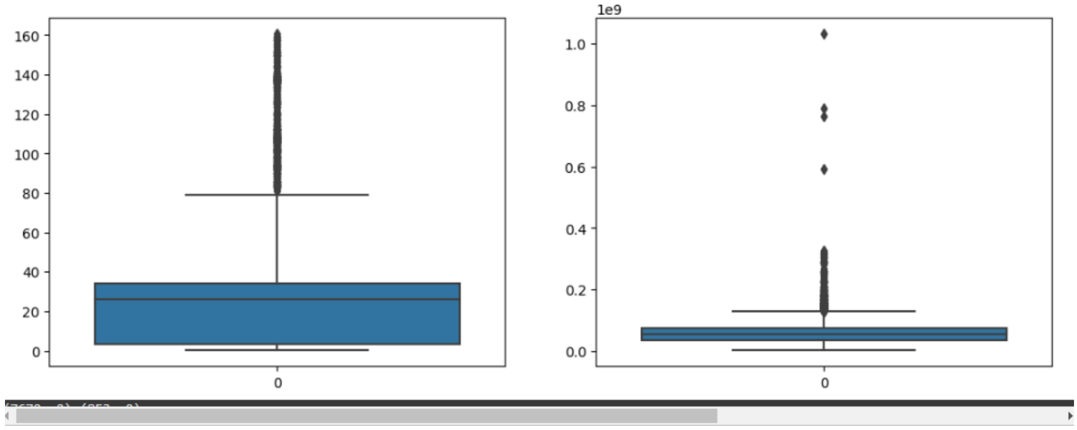
OUTPUT



## 3.BOXPLOT

```
plt.subplots(figsize=(20,10))
for i, col in enumerate(features):
  plt.subplot(2,3,i+1)
  sb.boxplot(df[col])
plt.show()
```

## OUTPUT

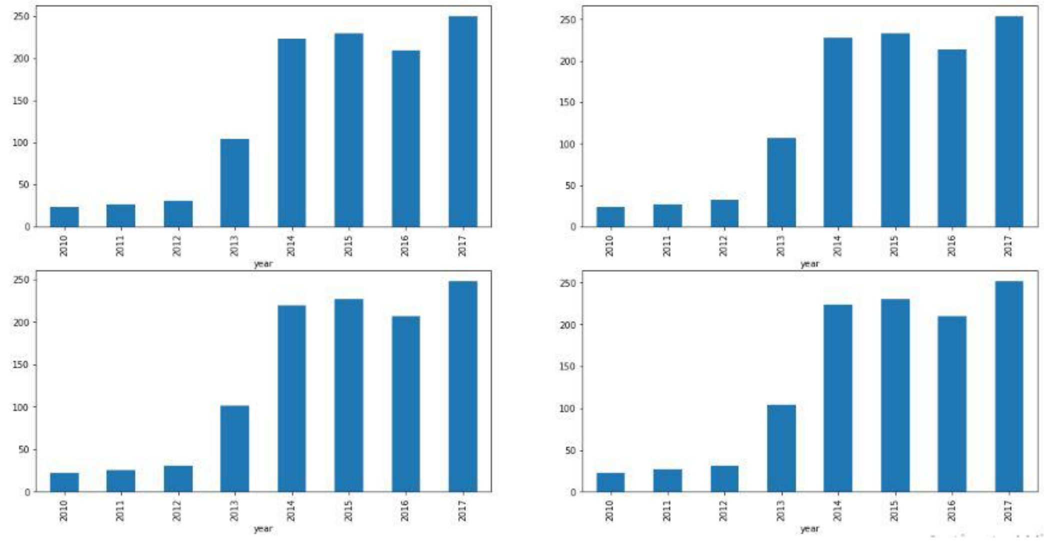


## 4.BAR CHART

```
data_grouped = df.groupby('year').mean()
plt.subplots(figsize=(20,10))

for i, col in enumerate(['Open', 'High', 'Low', 'Close']):
  plt.subplot(2,2,i+1)
  data_grouped[col].plot.bar()
plt.show()
```

## OUTPUT


year