Exploratory Data Analysis (EDA) Report - **Stock Price Prediction Challenge**

Data visualizations and Analysis

Initially the data set provided had 11291 rows and 8 feature columns

```
df = pd.read_csv('drive/MyDrive/question4-stock-data.csv')
df.shape
```

```
(11291, 8)
```

```
[95] df.dtypes
```

	0
Unnamed: 0	int64
Date	object
Adj Close	float64
Close	float64
High	float64
Low	float64
Open	float64
Volume	float64

dtype: object

Converted the Date column in order to use in future model training.

```
[96] # Convert Date column to datetime format
     df["Date"] = pd.to_datetime(df["Date"], format = '%Y-%m-%d',errors="coerce")
```

Sumerised numerical features to get an clear idea of how the data is distributed

```
# sumarize numerical features
df.describe()
```

	Unnamed: 0	Date	Adj Close	Close	High	Low	Open	Volume
count	11291.000000	11181	11198.000000	11174.000000	11196.000000	11164.000000	11188.000000	1.114600e+04
mean	5645.000000	2002-08-03 13:57:54.429836288	63.609130	72.026945	72.503100	71.665079	67.999259	2.144157e+05
min	0.000000	1980-03-17 00:00:00	2.259452	3.237711	3.237711	3.237711	0.000000	0.000000e+00
25%	2822.500000	1991-05-17 00:00:00	19.224636	27.500000	27.789255	27.536156	0.000000	1.350000e+04
50%	5645.000000	2002-07-26 00:00:00	50.608900	66.035000	66.724998	65.418751	66.065002	9.032350e+04
75%	8467.500000	2013-10-21 00:00:00	104.723621	114.297503	114.892500	113.639999	114.269997	2.915750e+05
max	11290.000000	2024-12-27 00:00:00	254.770004	254.770004	255.229996	253.589996	255.000000	1.858270e+07
std	3259.575279	NaN	52.266247	51.259828	51.550735	51.011632	55.834401	3.883662e+05

Removed unwanted features like Adj Close and Unnamed 0 first column,
(Because Adj Close and Close contained mostly similar values, and Unnamed 0 contained no valuable data)

```
[99] df = df.drop(columns=['Unnamed: 0', 'Adj Close'])
```

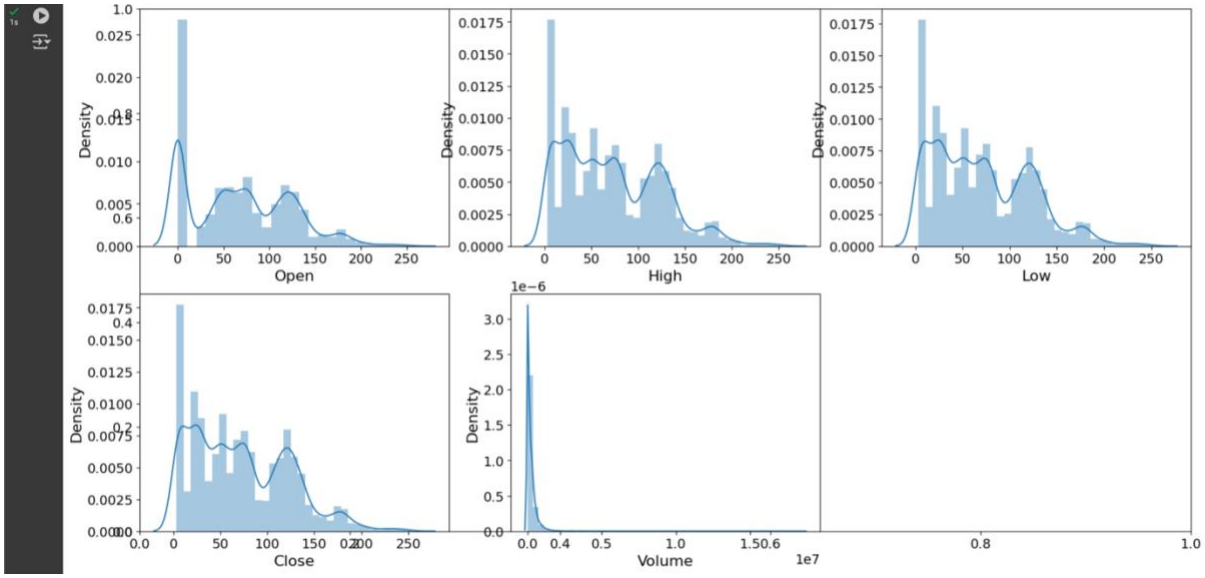
Done a data distribution visualization using histograms on each feature.
Each price showed a uniform distribution with distinguishable 4 peacks

The volume histogram was left skewed with a one high peak

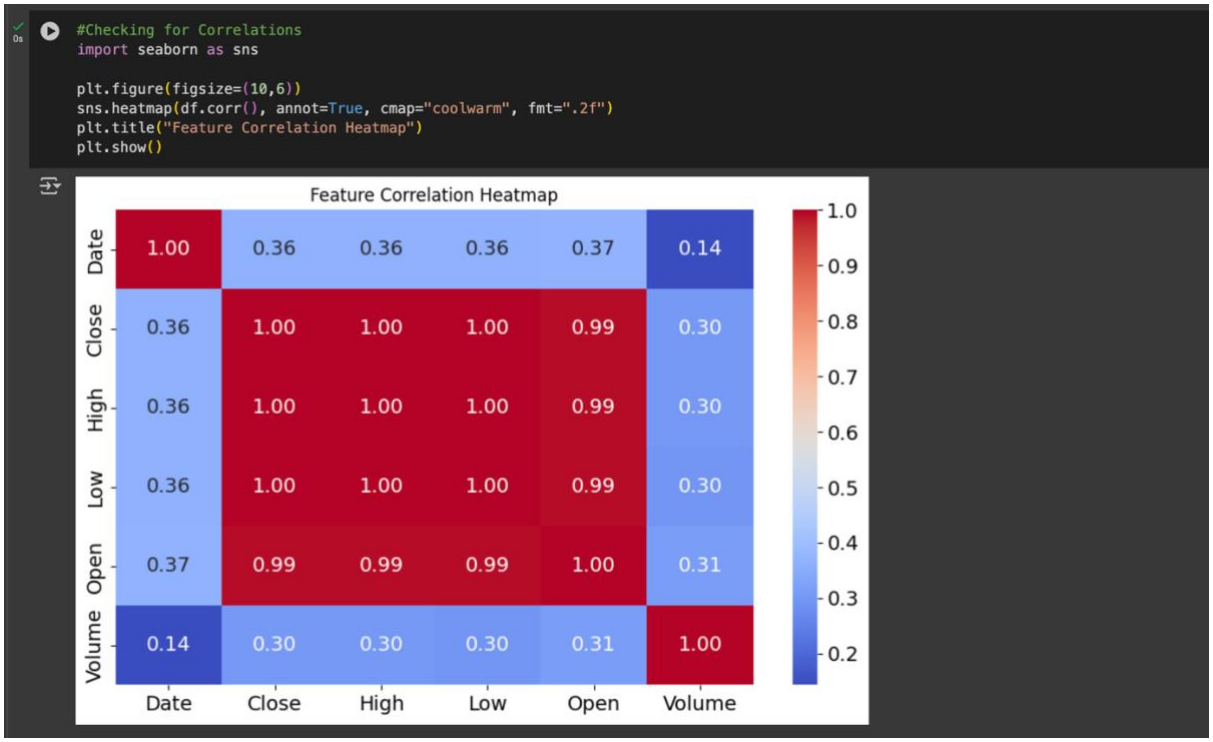
```
#data distribution histograms
features = ['Open', 'High', 'Low', 'Close', 'Volume']

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

for i, col in enumerate(features):
  plt.subplot(2,3,i+1)
  sb.distplot(df[col])
plt.show()
1
```



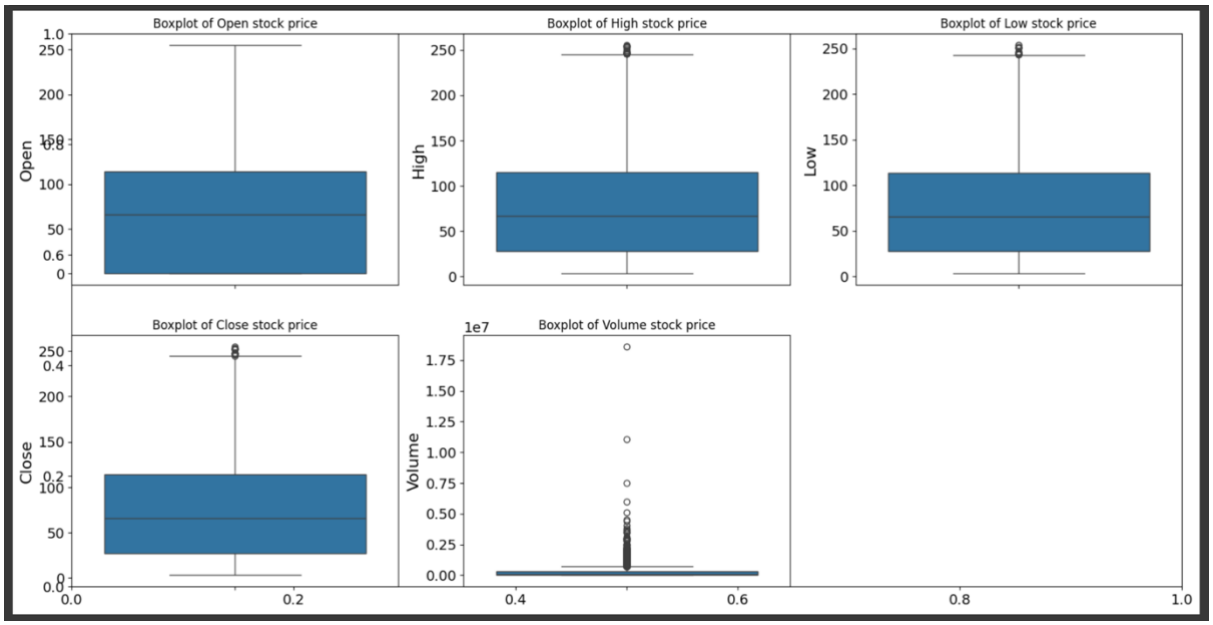
Plotted a feature Correlation heatmap to identify the dependancy of feeatures on each other. According to the heat map it was clearly showed Open, High,Low ,Close were highly correlated. While Date and the Close price was not too co related on each other



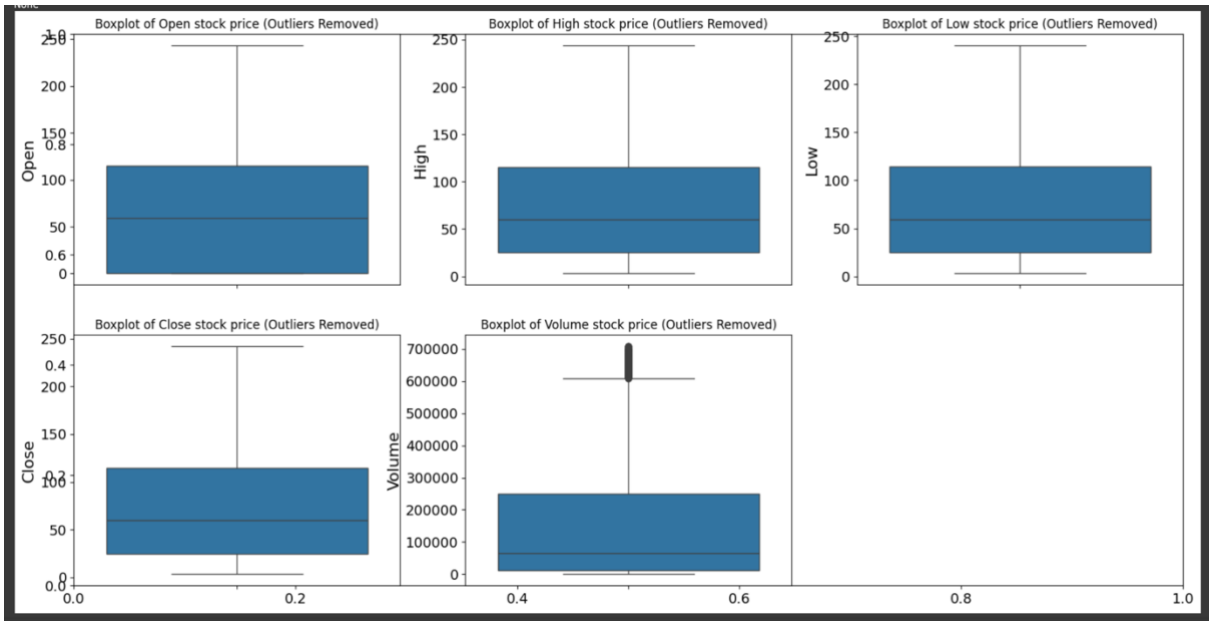
Then after identifying the dependancies among each feature, I tested data set for extreme data points to ensure data quality, improve analytical accuracy, enhance model performance, and uncover critical anomalies

Outlier graph received from the raw data set : High , Low and Close prices showed some mini outliers, while the Volume had some serious outliers

```
#detecting outliers
plt.subplots(figsize=(20,10))
for i, col in enumerate(features):
  plt.subplot(2,3,i+1)
  plt.title(f"Boxplot of {features[i]} stock price")
  sb.boxplot(df[col])
plt.show()
```



Then removed Outliers to have a quality data set to remove annomalies and to improved model accuracy.



Preprocessed data

Sorted data by date bacause tryng to do a Time series analysis, Using random order can corroupt the model training

```
[105] # sort by datetime
      df.sort_values(by='Date', inplace=True, ascending=True)
```

There were no duplicate data found and there were some missing data in each column. So had to drop them.

```
duplicates = df.duplicated().sum()
missing_values = df.isnull().sum()

duplicates, missing_values
```

```
(0,
 Date      110
 Close     117
 High       95
 Low       127
 Open      103
 Volume    145
 dtype: int64)
```

```
[107] # Drop rows with missing values
      df = df.dropna()
```

Plotted the close price history to identify the trend in the past. The stock price was relatively low and stable before **1995**. It shows **rapid growth and fluctuations** from **2000 onward**, with significant peaks and drops.Around **2020-2025**, the stock price reached its highest values before experiencing a noticeable decline.

The stock follows an **upward trend** over time, which may indicate long-term growth.

There are **multiple sharp drops**, suggesting market volatility

The **recent high volatility** (especially post-2020) could be due to economic factors, global events, or company-specific performance.

```
plt.figure(figsize = (16,8))
plt.plot(df['Date'], df['Close'], label = 'Closing Price History')
plt.legend(loc = "upper left")
plt.xlabel('Year')
plt.ylabel('Stock Price ($)')
plt.show()
```