**DEVELOPMENT PART 1**

In this part you will begin building your project by loading and preprocessing the dataset. Begin building the stock price prediction model by loading and preprocessing the dataset. Collect and preprocess the historical stock market data for analysis.

**Dataset Link:** [**https://www.kaggle.com/datasets/prasoonkottarathil/microsoft-lifetime-stocks-dataset**](https://www.kaggle.com/datasets/prasoonkottarathil/microsoft-lifetime-stocks-dataset)

**Data Preprocessing**

Data preprocessing is a crucial step when using Long Short-Term Memory (LSTM) networks for stock price predictions. Here's a general outline of the preprocessing

**STEPS:**

Data preprocessing is a crucial step when using Long Short-Term Memory (LSTM) networks for stock price predictions. Here's a general outline of the preprocessing.

**PROGRAM:**

pip install -q yfinance

pip install pandas\_datareader

**import** pandas **as** pd

**import** numpy **as** np

**import** matplotlib.pyplot **as** plt

**import** seaborn **as** sns

sns**.**set\_style('whitegrid')

plt**.**style**.**use("fivethirtyeight")

**%matplotlib** inline

**from** pandas\_datareader.data **import** DataReader

**import** yfinance **as** yf

**from** pandas\_datareader **import** data **as** pdr

yf**.**pdr\_override()

**from** datetime **import** datetime

fig, axes **=** plt**.**subplots(nrows**=**2, ncols**=**2)

fig**.**set\_figheight(10)

fig**.**set\_figwidth(15)

AAPL[['Adj Close', 'MA for 10 days', 'MA for 20 days', 'MA for 50 days']]**.**plot(ax**=**axes[0,0])

axes[0,0]**.**set\_title('APPLE')

GOOG[['Adj Close', 'MA for 10 days', 'MA for 20 days', 'MA for 50 days']]**.**plot(ax**=**axes[0,1])

axes[0,1]**.**set\_title('GOOGLE')

MSFT[['Adj Close', 'MA for 10 days', 'MA for 20 days', 'MA for 50 days']]**.**plot(ax**=**axes[1,0])

axes[1,0]**.**set\_title('MICROSOFT')

AMZN[['Adj Close', 'MA for 10 days', 'MA for 20 days', 'MA for 50 days']]**.**plot(ax**=**axes[1,1])

axes[1,1]**.**set\_title('AMAZON')

fig**.**tight\_layout()

fig, axes **=** plt**.**subplots(nrows**=**2, ncols**=**2)

fig**.**set\_figheight(10)

fig**.**set\_figwidth(15)

AAPL['Daily Return']**.**plot(ax**=**axes[0,0], legend**=True**, linestyle**=**'--', marker**=**'o')

axes[0,0]**.**set\_title('APPLE')

GOOG['Daily Return']**.**plot(ax**=**axes[0,1], legend**=True**, linestyle**=**'--', marker**=**'o')

axes[0,1]**.**set\_title('GOOGLE')

MSFT['Daily Return']**.**plot(ax**=**axes[1,0], legend**=True**, linestyle**=**'--', marker**=**'o')

axes[1,0]**.**set\_title('MICROSOFT')

AMZN['Daily Return']**.**plot(ax**=**axes[1,1], legend**=True**, linestyle**=**'--', marker**=**'o')

axes[1,1]**.**set\_title('AMAZON')

fig**.**tight\_layout()

plt**.**figure(figsize**=**(12, 9))

**for** i, company **in** enumerate(company\_list, 1):

plt**.**subplot(2, 2, i)

company['Daily Return']**.**hist(bins**=**50)

plt**.**xlabel('Daily Return')

plt**.**ylabel('Counts')

plt**.**title(f'{company\_name[i **-** 1]}')

plt**.**tight\_layout()

fig, axes **=** plt**.**subplots(nrows**=**2, ncols**=**2)

fig**.**set\_figheight(10)

fig**.**set\_figwidth(15)

AAPL[['Adj Close', 'MA for 10 days', 'MA for 20 days', 'MA for 50 days']]**.**plot(ax**=**axes[0,0])

axes[0,0]**.**set\_title('APPLE')

GOOG[['Adj Close', 'MA for 10 days', 'MA for 20 days', 'MA for 50 days']]**.**plot(ax**=**axes[0,1])

axes[0,1]**.**set\_title('GOOGLE')

MSFT[['Adj Close', 'MA for 10 days', 'MA for 20 days', 'MA for 50 days']]**.**plot(ax**=**axes[1,0])

axes[1,0]**.**set\_title('MICROSOFT')

AMZN[['Adj Close', 'MA for 10 days', 'MA for 20 days', 'MA for 50 days']]**.**plot(ax**=**axes[1,1])

axes[1,1]**.**set\_title('AMAZON')

fig**.**tight\_layout()

**for** company **in** company\_list:

company['Daily Return'] **=** company['Adj Close']**.**pct\_change()

fig, axes **=** plt**.**subplots(nrows**=**2, ncols**=**2)

fig**.**set\_figheight(10)

fig**.**set\_figwidth(15)

AAPL['Daily Return']**.**plot(ax**=**axes[0,0], legend**=True**, linestyle**=**'--', marker**=**'o')

axes[0,0]**.**set\_title('APPLE')

GOOG['Daily Return']**.**plot(ax**=**axes[0,1], legend**=True**, linestyle**=**'--', marker**=**'o')

axes[0,1]**.**set\_title('GOOGLE')

MSFT['Daily Return']**.**plot(ax**=**axes[1,0], legend**=True**, linestyle**=**'--', marker**=**'o')

axes[1,0]**.**set\_title('MICROSOFT')

AMZN['Daily Return']**.**plot(ax**=**axes[1,1], legend**=True**, linestyle**=**'--', marker**=**'o')

axes[1,1]**.**set\_title('AMAZON')

fig**.**tight\_layout()

plt**.**figure(figsize**=**(12, 9))

**for** i, company **in** enumerate(company\_list, 1):

plt**.**subplot(2, 2, i)

company['Daily Return']**.**hist(bins**=**50)

plt**.**xlabel('Daily Return')

plt**.**ylabel('Counts')

plt**.**title(f'{company\_name[i **-** 1]}')

plt**.**tight\_layout()

closing\_df **=** pdr**.**get\_data\_yahoo(tech\_list, start**=**start, end**=**end)['Adj Close']

tech\_rets **=** closing\_df**.**pct\_change()

tech\_rets**.**head()

sns**.**jointplot(x**=**'GOOG', y**=**'GOOG', data**=**tech\_rets, kind**=**'scatter', color**=**'seagreen')

sns**.**pairplot(tech\_rets, kind**=**'reg')

return\_fig **=** sns**.**PairGrid(tech\_rets**.**dropna())

return\_fig**.**map\_upper(plt**.**scatter, color**=**'purple')

return\_fig**.**map\_lower(sns**.**kdeplot, cmap**=**'cool\_d')

return\_fig**.**map\_diag(plt**.**hist, bins**=**30)

returns\_fig **=** sns**.**PairGrid(closing\_df)

returns\_fig**.**map\_upper(plt**.**scatter,color**=**'purple')

returns\_fig**.**map\_lower(sns**.**kdeplot,cmap**=**'cool\_d')

returns\_fig**.**map\_diag(plt**.**hist,bins**=**30)

plt**.**figure(figsize**=**(12, 10))

plt**.**subplot(2, 2, 1)

sns**.**heatmap(tech\_rets**.**corr(), annot**=True**, cmap**=**'summer')

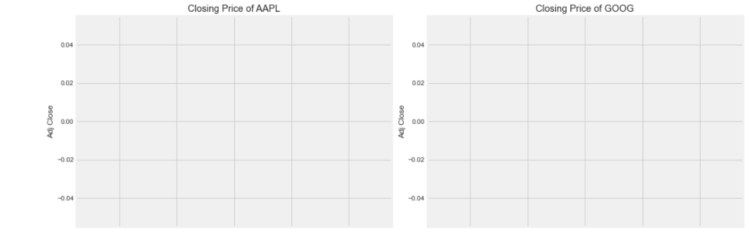
plt**.**title('Correlation of stock return')

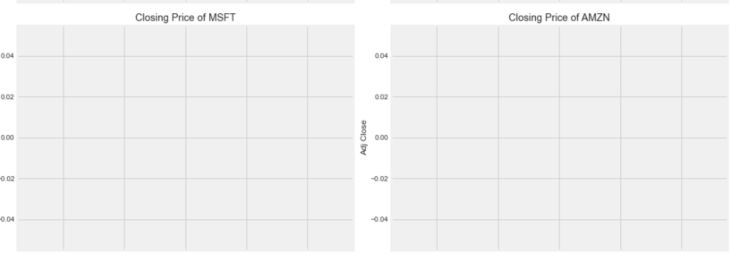
plt**.**subplot(2, 2, 2)

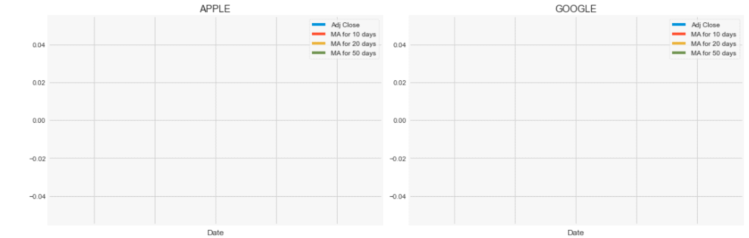
sns**.**heatmap(closing\_df**.**corr(), annot**=True**, cmap**=**'summer')

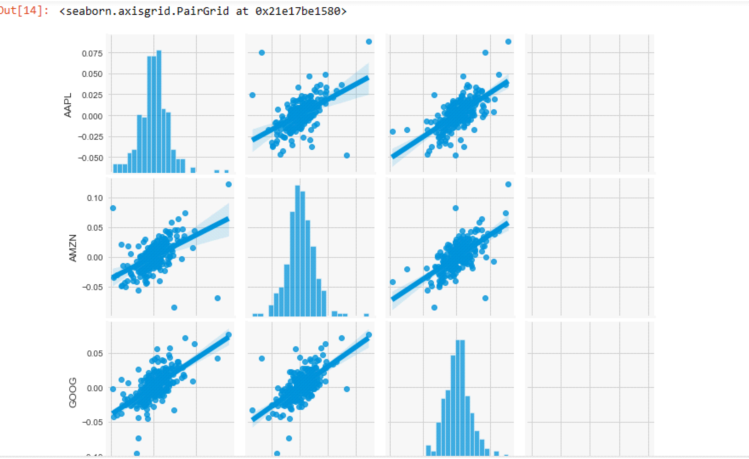
plt**.**title('Correlation of stock closing price')

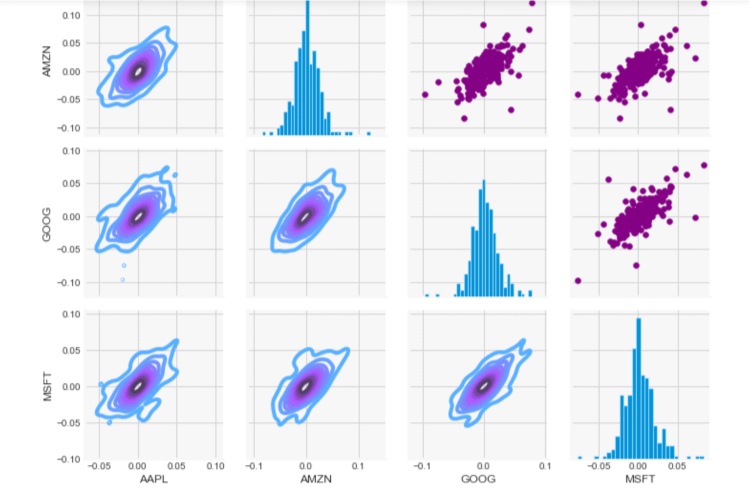
**Output**

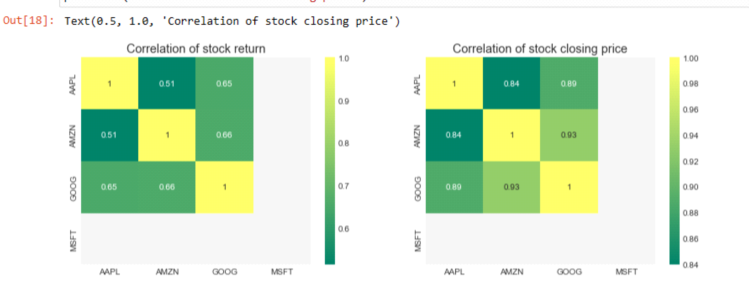














**Feature Engineering:**

Create additional features that might help improve the predictive performance. These features can include technical indicators like moving averages, relative strength index (RSI), and Bollinger Bands, among others.

Calculate any financial metrics or ratios that might be relevant for your analysis, such as price-to-earnings ratio (P/E ratio), earnings per share (EPS), or dividend yield.

**Data Visualization :**

Visualize the data to gain insights and understand any trends, patterns, or anomalies present in the dataset. You can use libraries like Matplotlib or Seaborn for visualization