

PROJECT TITLE :STOCK PRICE PREDICTION

PHASE 3:Development part 1

Collect Historical Stock Market Data: You can obtain historical stock market data from various sources such as financial APIs (e.g., Alpha Vantage, Yahoo Finance), financial data providers, or by web scraping.

Make sure to gather data for the stock you want to predict, including features like date, open price, high price, low price, close price, trading volume, and any other relevant data.

Load the Dataset:

Use a programming language like Python and libraries like Pandas to load the dataset into a DataFrame.

Import pandas as pd

Load the dataset

```
Df = pd.read_csv('stock_data.csv') # Replace 'stock_data.csv' with  
your dataset file path
```

Data Preprocessing:

Preprocess the data to make it suitable for model training. This may include handling missing values, converting date columns to the appropriate data type, and scaling/normalizing numerical features.

Data preprocessing example (assuming you have columns 'Date', 'Open', 'High', 'Low', 'Close', 'Volume')

```
Df['Date'] = pd.to_datetime(df['Date'])
```

```
Df.set_index('Date', inplace=True) # Set 'Date' as the index
```

```
Df.sort_index(ascending=True, inplace=True) # Sort data by date in  
ascending order
```

Perform other data preprocessing tasks as needed

Feature Engineering:

You can create additional features like moving averages, technical indicators (e.g., RSI, MACD), and other relevant metrics that might help your prediction model.

Train-Test Split:

Split your data into training and testing sets. Typically, you reserve a portion of the data (e.g., the most recent data) for testing the model's performance.

```
From sklearn.model_selection import train_test_split
```

```
# Split data into training and testing sets
```

```
Train_size = 0.8 # Adjust the split ratio as needed
```

```
Train_data, test_data = train_test_split(df, train_size=train_size,  
shuffle=False)
```