STOCK PRICE PREDICTION

Loading and pre-processing the provided stock price dataset typically involves the following steps:

Data Collection: We need to obtain historical stock price data.as provide from Kaggle website the link for the dataset on which tasks are performed is given below:

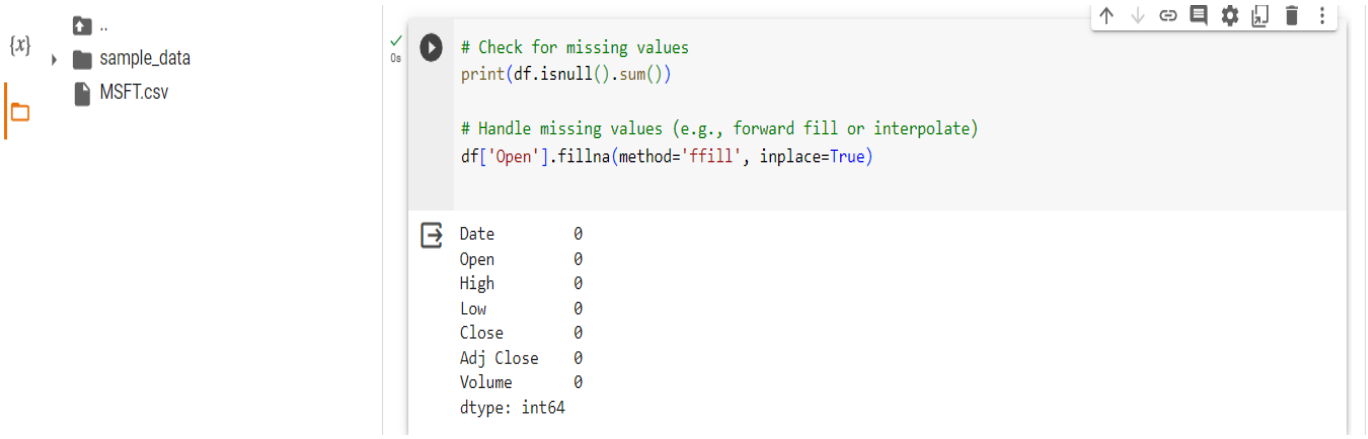
https://www.kaggle.com/datasets/prasoonkottarathil/microsoft-lifetime-stocks-dataset

Data Format: First we should ensure that the data is in a format that you can work with. Typically, this data will be in CSV (Comma-Separated Values) format or some other structured format.

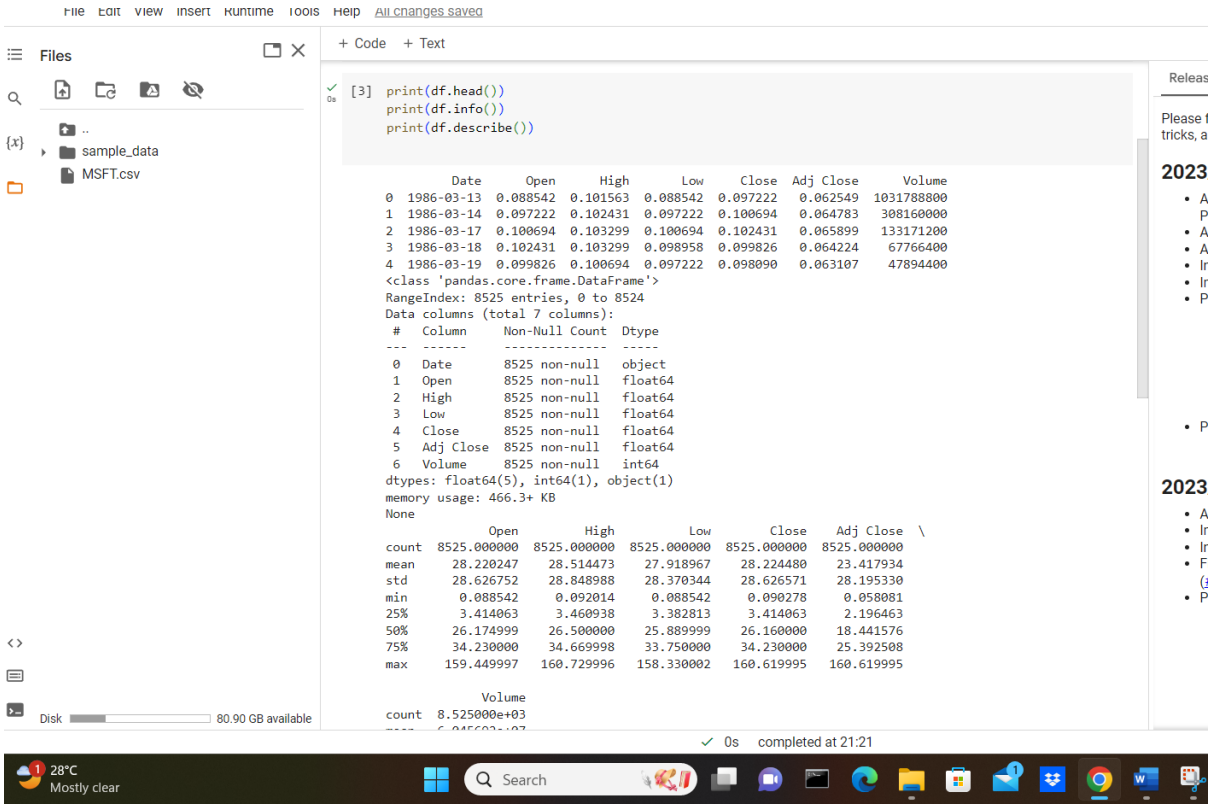
Import Libraries: We will need to import relevant libraries for data manipulation and analysis. Common libraries for this task include ***pandas, numpy, and matplotlib*** for visualization. We may also need libraries to fetch data from the web.

Load the Data: Use pandas to read the dataset into a DataFrame. Here we load a CSV file:

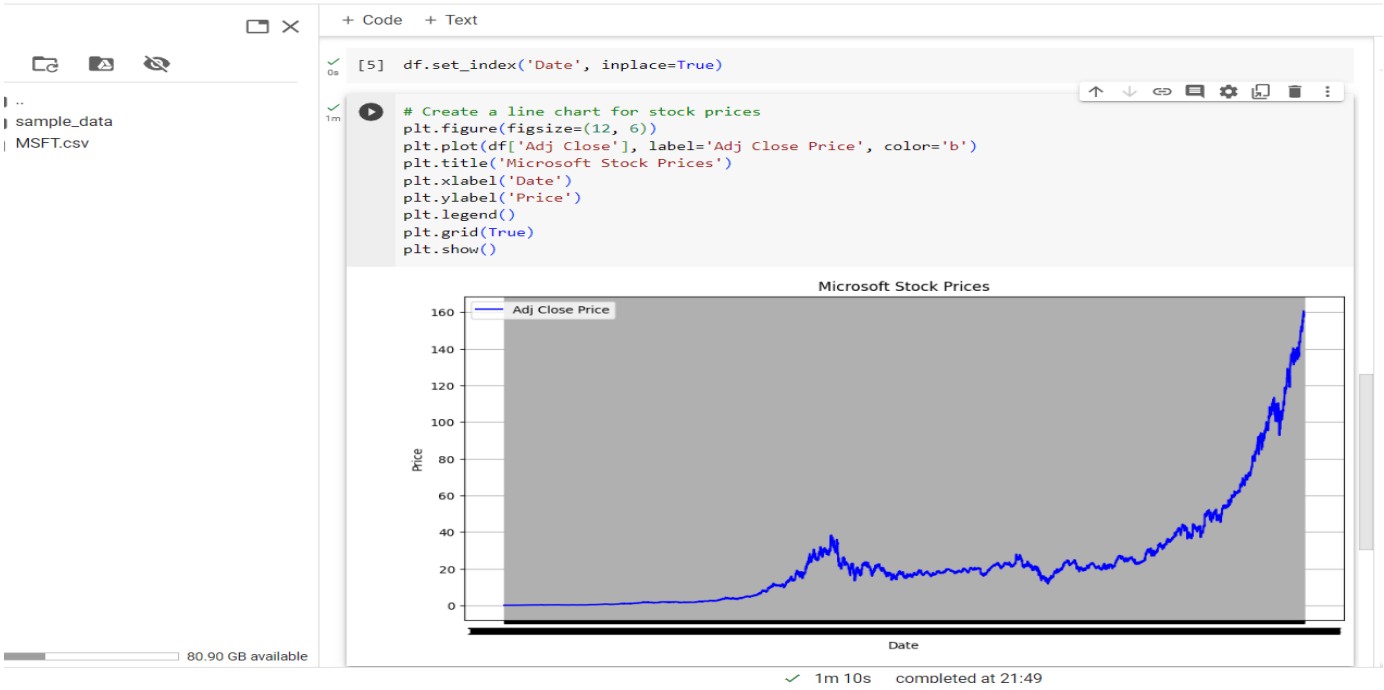
Data Cleaning: Check for missing values, duplicate rows, and outliers in the dataset. We need to handle missing data through imputation, drop duplicates, and handle outliers based on our analysis requirements.

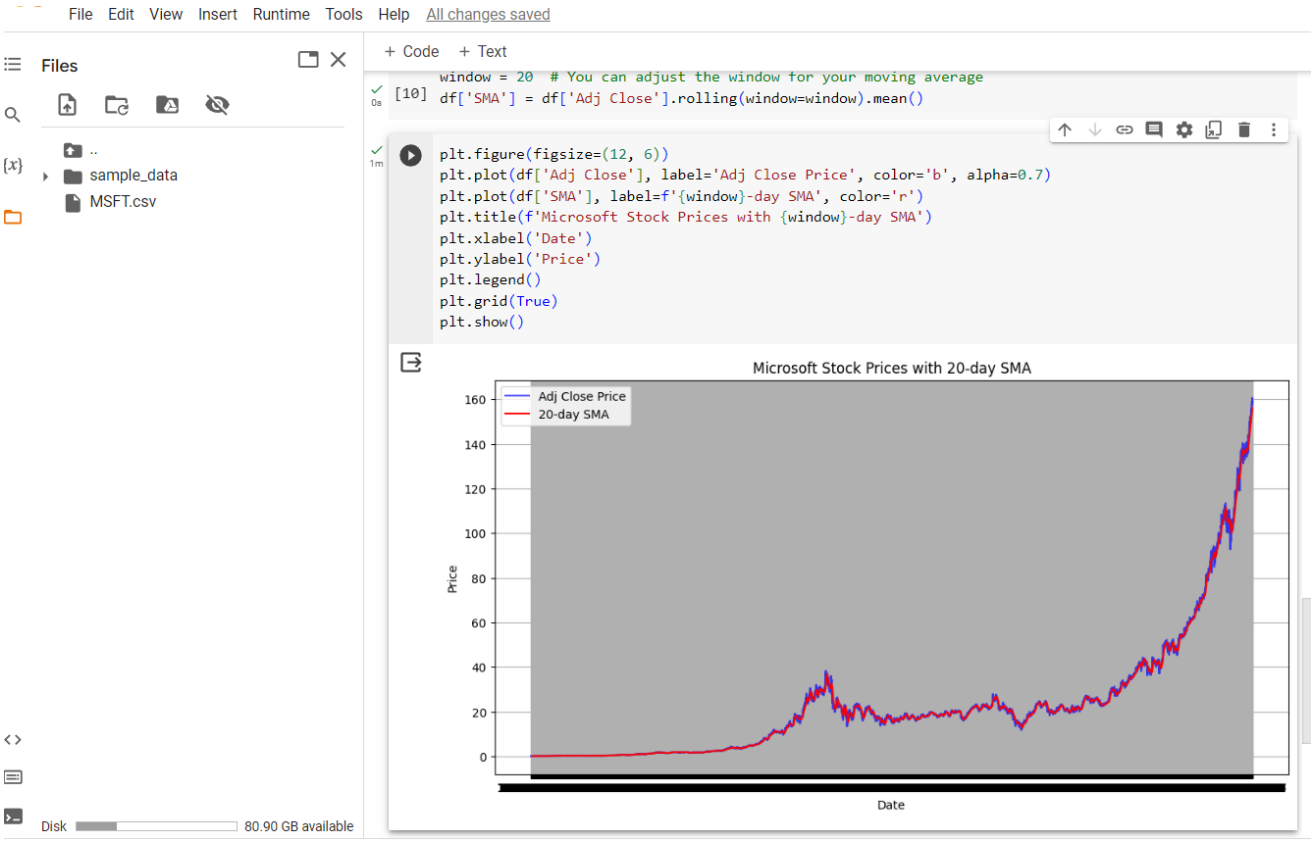


Data Exploration: Explore the dataset to gain insights. We use methods like describe(), info(), and visualizations to understand the data better.



After loading and pre-processing stock price dataset, it's a good practice to create visualizations to gain insights and better understand the data. Common visualizations for financial time series data like stock prices include line charts, candlestick charts, volume plots, and moving averages. Here we can create these visualizations using Python.





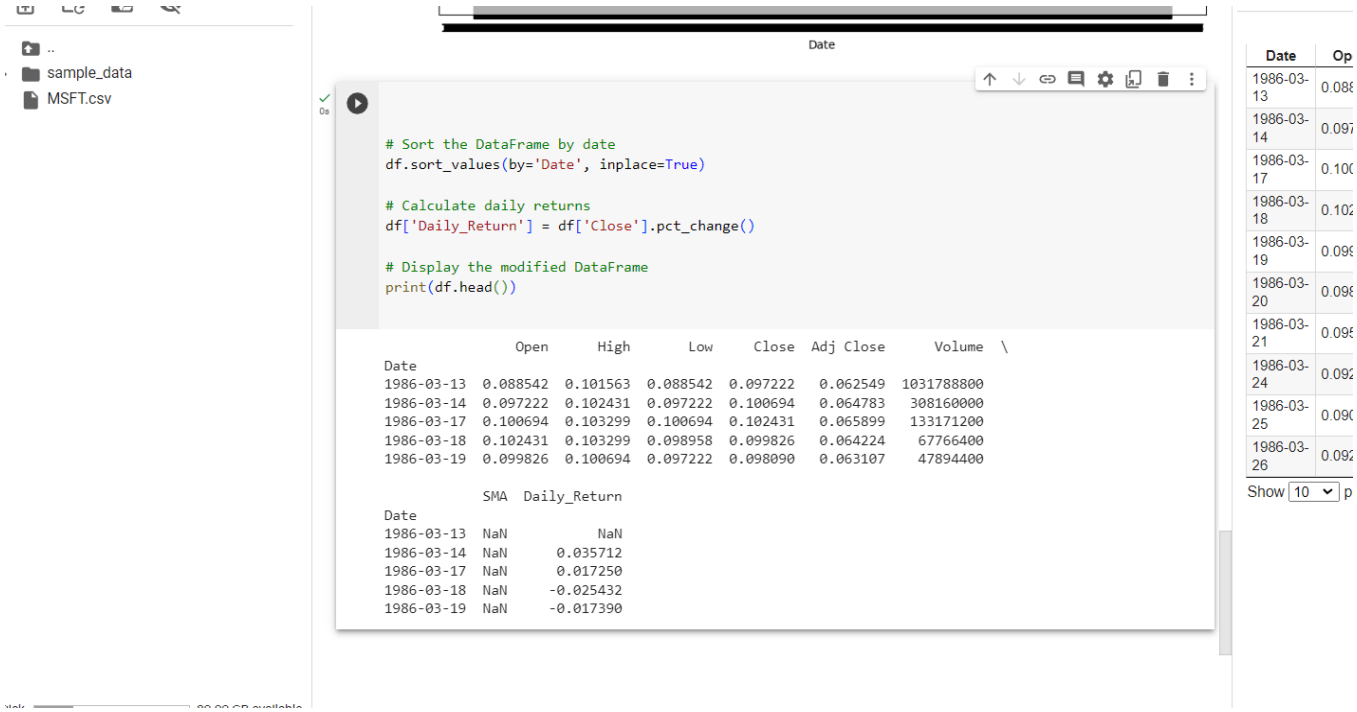
Data Pre-processing:

Date Conversion: Convert the date column to a datetime object if it's not already in that format.

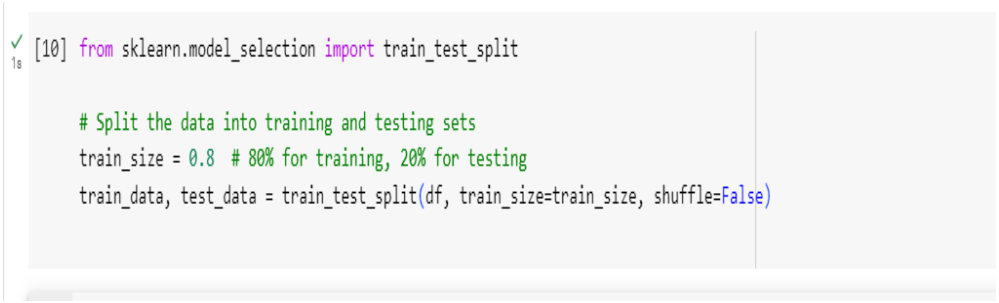
Sorting: Sort the Data Frame by date if it's not already in chronological order.

Feature Engineering: Create additional features if needed, such as moving averages, daily returns, or technical indicators.

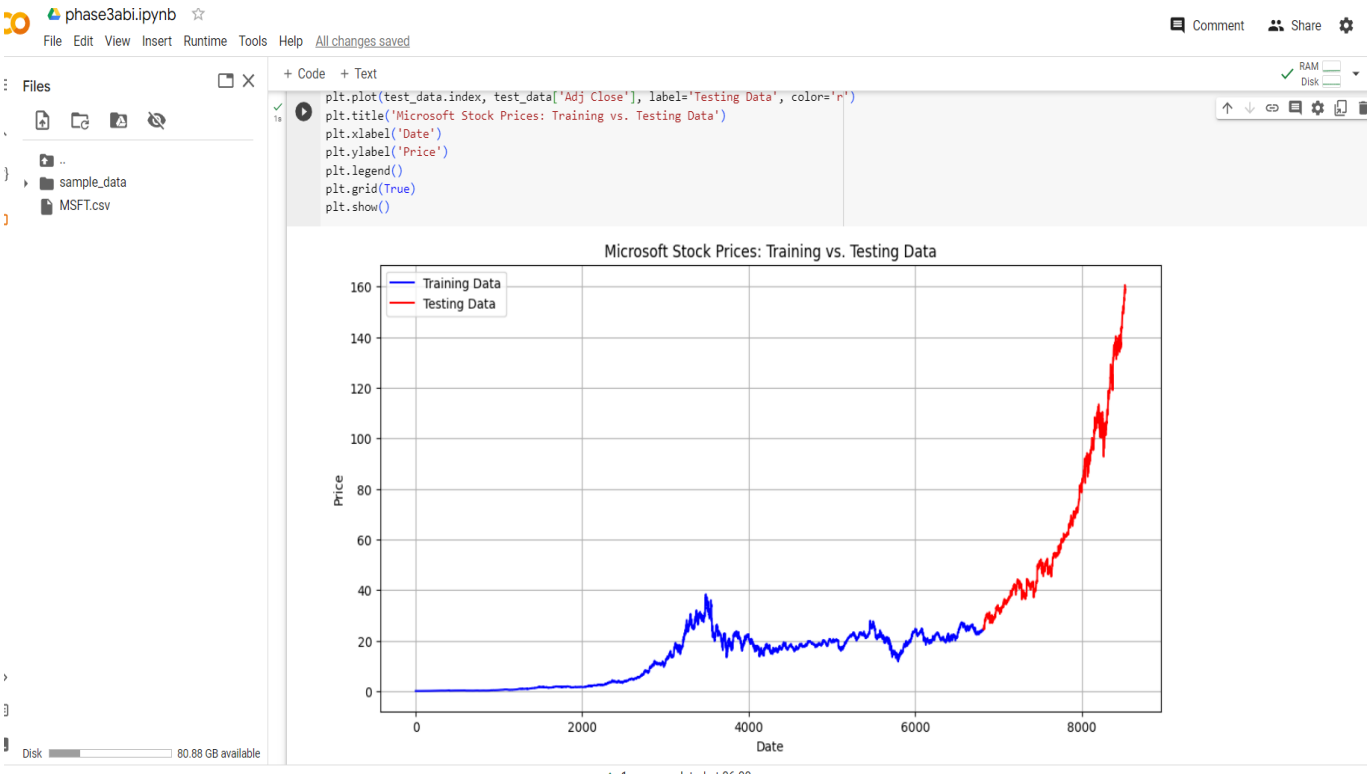
Normalization/Scaling: If you plan to use machine learning models, you might want to normalize or scale the data. Common techniques include Min-Max scaling or standardization.



Data Splitting: If you plan to build a predictive model, split the dataset into training and testing sets. Make sure to maintain the chronological order.



To visualize the training and testing datasets for stock price dataset, we create separate line charts for each dataset to compare them. This helps in understanding how our model's predictions (on the test dataset) compare to the actual stock price movements (from the training dataset).



We load the pre-processed data, containing the 'Adj Close' prices.

We have separate Data Frames for training and testing data, 'train_data' and 'test_data'.

We create line charts for both datasets, with different colours to distinguish between training and testing data.

The x-axis represents the date, and the y-axis represents the adjusted closing price.

This visualization helps compare the stock price movements in the training and testing datasets. It's useful for assessing how well our model generalizes to unseen data. In an ideal scenario, the testing data would closely follow the trends seen in the training data, indicating that our model has learned and generalizes well.

In addition to the basic preprocessing steps mentioned earlier, there are several other advanced preprocessing steps that can be performed

on a stock price dataset. These steps are often used to enhance the quality and relevance of the data for analysis and modelling.