STOCK PRICE PREDICTION

TEAM MEMBERS

MANIKANDAN.B

KARTHICK.M

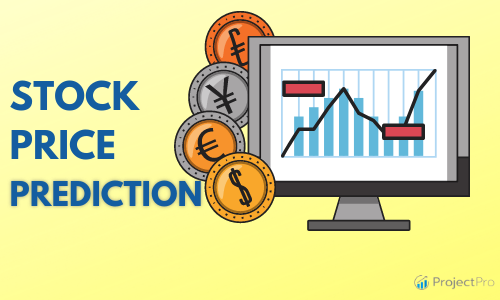
MATHAN.S

SHIBIN.G

PRASANNA.KM

PHASE 3 SUBMISSION DOCUMENT

PHASE 3: DEVELOPMENT PART 1



INTRODUCTION

Stock Price Prediction using machine learning helps you discover the future value of company stock and other financial assets traded on an exchange. The entire idea of predicting stock prices is to gain significant profits. Predicting how the stock market will perform is a hard task to do. There are other factors involved in the prediction, such as physical and psychological factors, rational and irrational behavior, and so on. All these factors combine to make share prices dynamic and volatile. This makes it very difficult to predict stock prices with high accuracy.

DATASET LINK : [Microsoft Historical Dataset (kaggle.com)](https://www.kaggle.com/datasets/prasoonkottarathil/microsoft-lifetime-stocks-dataset)

NECESSARY STEPS TO FOLLOW:

Start by importing libraries:

Pandas:

Pandas is essential for data manipulation and analysis, particularly for loading and handling datasets.

Program:

import pandas as pd

NumPy:

NumPy is used for numerical computations, and it complements Pandas for handling arrays and mathematical operations.

Program:

import NumPy as np

Scikit-Learn (sklearn):

Scikit-Learn provides tools for machine learning, including dataset splitting, preprocessing, and model evaluation. You'll import specific modules as needed for your analysis.

Program:

from sklearn. model selection import train\_test\_split

from sklearn. Preprocessing import StandardScaler # For data scaling (if needed)

LOAD THE DATASET:

To load a dataset for credit card fraud detection, We can use the Pandas library in Python. Here's how we can load a dataset from a CSV file, which is a common data format:

Program:

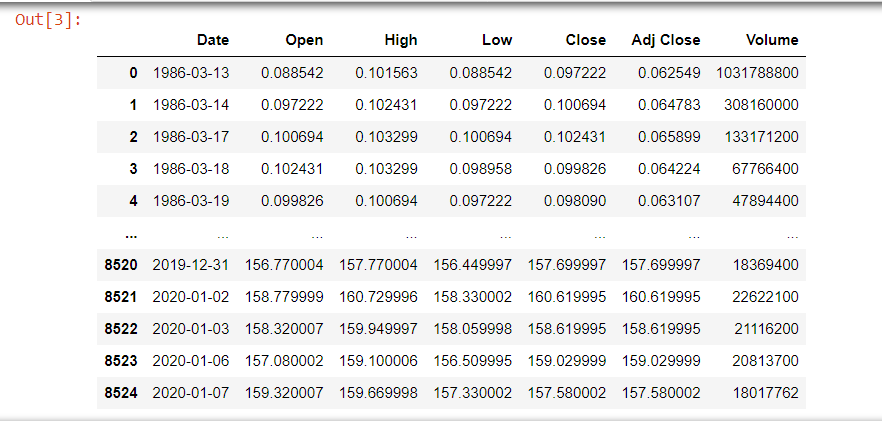
import pandas as pd

# Specify the file path to your dataset

File\_path=(" C:/Users/mithu/Documents/manikandan/MSFT.csv ")

# Use Pandas to read the CSV file into a DataFrame

df = pd.read\_csv("C:/Users/mithu/Documents/manikandan/MSFT.csv")



# Now, 'df' contains your dataset, and you can start working with it.

In the code above:

Import the Pandas library to work with data.

Replace 'your\_dataset.csv' with the actual file path to our dataset. Make sure that the CSV file is in the same directory as our Python script, or provide the full path to the file if it's located elsewhere.

The pd.read\_csv(file\_path) function reads the CSV file and stores its contents in a Pandas DataFrame called df. This DataFrame is a two-dimensional table-like data structure that you can manipulate and analyze.

After loading the dataset into a Data Frame, we can perform various data analysis tasks, such as data exploration, preprocessing, and modelling, depending on your specific objectives in credit card fraud detection.

Data processing :

Data processing occurs when data is collected and translated into usable information. Usually performed by a data scientist or team of data scientists, it is important for data processing to be done correctly as not to negatively affect the end product, or data output.

Basic Summary Statistics:

Use Pandas to obtain summary statistics of the dataset, which can give a quick overview of the data, including counts, means, standard deviations, and percentiles.

Program:

print(df.describe())

Data Shape:

Use to see how many Rows and columns in our Dataset.

df**.**shape

(8525, 7)

Dependent and Independent Variables

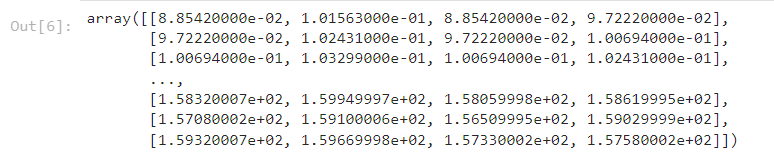
So, y is referred to as dependent feature or variable of total\_vaccinations and x is referred to as independent features or variables of location,date,vaccine,total\_vaccinations. Any predictive mathematical model tends to divide the observations (data) into dependent/ independent features in order to determine the causal effect.

Program:

x**=**df[['Open','High','Low','Close']]**.**values

# x for independent variables

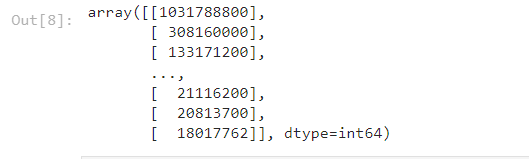
X



# y for dependent variables

y**=**df[['Volume']]**.**values

y



Sklearn.impute

The SimpleImputer class provides basic strategies for imputing missing values. Missing values can be imputed with a provided constant value, or using the statistics (mean, median or most frequent) of each column in which the missing values are located. This class also allows for different missing values encodings.

Program:

**import** sklearn.impute **as** sl

In

**from** sklearn.impute **import** SimpleImputer

In

imputer**=**SimpleImputer(missing\_values**=**np**.**nan,strategy**=**'mean')

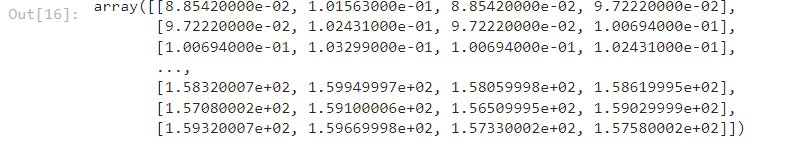
In

imputer**=**imputer**.**fit(x[:,1:2])

In

x

output:



Label encoder

Label encoding is a technique used in machine learning and data analysis to convert categorical variables into numerical format. It is particularly useful when working with algorithms that require numerical input, as most machine learning models can only operate on numerical data.

Program:

**from** sklearn.preprocessing **import** LabelEncoder

In :

label\_encode\_x**=**LabelEncoder()

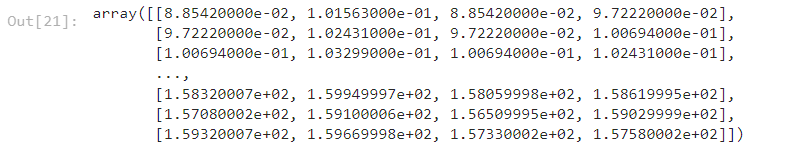
In :

y[:,0]**=**label\_encode\_x**.**fit\_transform(x[:,0])

In :

x

Output:



To use machine learning models, analyse feature importance to understand which attributes play a significant role in covid-19 vaccine analysis

One-hot encoder

One-Hot Encoding is another popular technique for treating categorical variables. It simply creates additional features based on the number of unique values in the categorical feature. Every unique value in the category will be added as a feature. One-Hot Encoding is the process of creating dummy variables.

Program:

**from** sklearn.preprocessing **import** OneHotEncoder

In :

onehotencoder**=**OneHotEncoder()

In :

onehotencoder**.**fit\_transform(df**.**location**.**values**.**reshape(**-**1,1))**.**toarray()

output:

array([[1., 0., 0., ..., 0., 0., 0.],

[0., 0., 0., ..., 0., 0., 0.],

[0., 0., 0., ..., 0., 0., 0.],

...,

[0., 0., 0., ..., 0., 0., 0.],

[0., 0., 0., ..., 0., 0., 0.],

[0., 0., 0., ..., 0., 1., 0.]])

LabelEncoder Fit\_transform

LabelEncoder , we can use the fit\_transform function. This function fits the LabelEncoder object to the input data, and then transforms the data into encoded values. By default, fit\_transform assigns a unique numerical value to each category in the input data.

Program:

labelencoder\_y**=**LabelEncoder()

In :

y**=**labelencoder\_y**.**fit\_transform(y)

y

output:

array([8336, 8325, 8010, ..., 521, 502, 291], dtype=int64)

sklearn model\_selection in train\_test\_split

train\_test\_split is a function in Sklearn model selection for splitting data arrays into two subsets: for training data and for testing data. With this function, you don't need to divide the dataset manually. By default, Sklearn train\_test\_split will make random partitions for the two subsets.

Program:

**from** sklearn.model\_selection **import** train\_test\_split

x\_train,x\_test,y\_train,y\_test**=**train\_test\_split(x,y,test\_size**=**0.2,train\_size**=**0.4,random\_state**=**0)

Input:

x\_train

output:

array([[41.73 , 42. , 41.689999, 41.990002],

[25.1 , 25.219999, 24.92 , 25.120001],

[27.549999, 27.719999, 27.42 , 27.540001],

...,

[25.4 , 25.41 , 25.110001, 25.26 ],

[39.96875 , 41.09375 , 39.65625 , 40.15625 ],

[10.5 , 10.53125 , 10.3125 , 10.328125]])

Input:

x\_test

output:

array([[23.709999, 23.790001, 23.15 , 23.219999],

[27.08 , 27.110001, 26.809999, 26.92 ],

[30.17 , 30.4 , 29.889999, 30.16 ],

...,

[17.40625 , 17.421875, 16.96875 , 17.125 ],

[37.349998, 37.599998, 37.299999, 37.470001],

[ 2.921875, 2.945313, 2.875 , 2.890625]])

Input:

y\_train

output

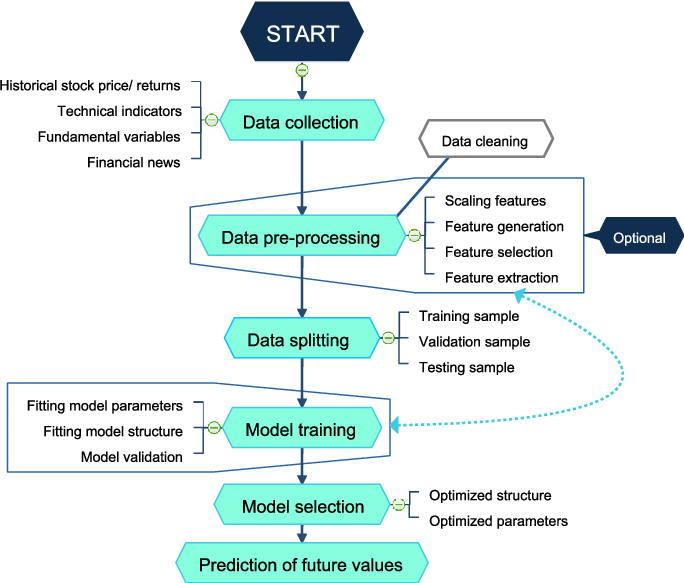
array([ 348, 4449, 4732, ..., 3773, 6198, 1811], dtype=int64)

input:

y\_test

output:

array([7254, 6564, 3283, ..., 4477, 1062, 6332], dtype=int64)



CONCLUSION

The stock market plays a remarkable role in our daily lives. It is a significant factor in a country's GDP growth. In this tutorial, you learned the basics of the stock market and how to perform stock price prediction using machine learning.

Do you have any questions related to this tutorial on stock prediction using machine learning? In case you do, then please put them in the comments section. Our team of experts will help you answer your questions.