# PHASE- III

# STOCK PRICE PREDICTION

## ***DATASET***

A data set is an ordered collection of data. As we know, a collection of information obtained through observations, measurements, study, or analysis is referred to as data. I could include information such as facts, numbers, figures, names, or even basic descriptions.

https://www.kaggle.com/datasetst f objects.

## Column description

- Dataset_name - Name of the dataset
- Author_name - Name of the author
- Author_id - Kaggle id of the author
- No_of_files - Number of files the author has uploaded
- size - Size of all the files
- Type_of_file - Type of the files such as csv, json etc.
- Upvotes - Total upvotes of the dataset
- Medals - Medal of the dataset
- Usability - Usability of the dataset
- Date - Date in which the dataset is uploaded
- Day - Day in which the dataset is uploaded
- Time - Time in which the dataset is uploaded
- Dataset_link - Kaggle link of the dataset How do you implement a dataset?

## What is data set with example?

- A data set is a collection of numbers or values that relate to a particular subject. For example, the test scores of each student in a particular class is a data set. The number of fish eaten by each dolphin at an aquarium is a data set. Other types of data sets.
- Numerical data, also known as quantitative data, is expressed in numbers instead of in what we know as natural language.
- Bavarians data sets contain only two variables.
- Multivariate data sets contain at least three variables that are somehow related.

| Dataset index | Dataset | Number of molecules | Number of descriptors |
|---|---|---|---|
| ACT1 | 3A4 | 50,000 | 9491 |
| ACT2 | CB1 | 11640 | 5877 |
| ACT3 | DPP4 | 8327 | 5203 |
| ACT4 | HIVINT | 2421 | 4306 |
| ACT5 | HIVPROT | 4311 | 6274 |
| ACT6 | LOGD | 50,000 | 8921 |
| ACT7 | METAB | 2092 | 4595 |
| ACT8 | NK1 | 13482 | 5803 |
| ACT9 | OX1 | 7135 | 4730 |
| ACT10 | OX2 | 14875 | 5790 |
| ACT11 | PGP | 8603 | 5135 |
| ACT12 | PPB | 11622 | 5470 |
| ACT13 | RAT_F | 7821 | 5698 |
| ACT14 | TDI | 5559 | 5945 |
| ACT15 | THROMBIN | 6924 | 5552 |

## Steps to Constructing Your Data set

1. Collect the raw data.

2. Identify feature and label sources.

3. Select a sampling strategy.

4. Split the data.

Steps to Constructing Your Dataset

1. Collect the raw data.

2. Identify feature and label sources.

3. Select a sampling strategy.

4. Split the data.

# BEGIN BUILDING THE PROJECT BY LOAD THE DATASET:

## Data set preparation is sometimes a DIY project:

If you were to consider a spherical machine-learning cow, all data preparation should be done by a dedicated data scientist. And that's about right. If you don't have a data scientist on board to do all the cleaning, well… you don't have machine learning. But as we discussed in our story on data science team structures, life is hard for companies that can't afford data science talent and try to transition existing IT engineers into the field. Besides, dataset preparation isn't narrowed down to a data scientist's competencies only. Problems with machine learning datasets can stem from the way an organization is built, workflows that are established, and whether instructions are adhered to or not among those charged with recordkeeping.

## 1. Articulate the problem early

Knowing what you want to predict will help you decide which data may be more valuable to collect. When formulating the problem, conduct data exploration and try to think in the categories of classification, clustering, regression, and ranking that we talked about in our whitepaper on business application of machine learning. In layman's terms, these tasks are differentiated in the following way:

## 2. Establish data collection mechanisms

Creating a data-driven culture in an organization is perhaps the hardest part of the entire initiative. We briefly covered this point in our story on machine learning strategy. If you aim to use ML for predictive analytics, the first thing to do is combat data fragmentation.

## 3. Check your data quality

The first question you should ask — do you trust your data? Even the most sophisticated machine learning algorithms can't work with poor data. We've talked in detail about data quality in a separate article, but generally you should look at several key things.

## 4. Format data to make it consistent

Data formatting is sometimes referred to as the file format you're using. And this isn't much of a problem to convert a dataset into a file format that fits your machine learning system best. We're talking about format consistency of records themselves.

## 5. Reduce data

It’s tempting to include as much data as possible, because of… well, big data! That’s wrong-headed. Yes, you definitely want to collect all data possible. But if you’re preparing a dataset with particular tasks in mind, it’s better to reduce data.

## 6. Complete data cleaning

Since missing values can tangibly reduce prediction accuracy, make this issue a priority. In terms of machine learning, assumed or approximated values are “more right” for an algorithm than just missing ones. Even if you don’t know the exact value, methods exist to better “assume” which value is missing or bypass the issue. How to clean data? Choosing the right approach also heavily depends on data and the domain you have:

## 7. Create new features out of existing ones

Some values in your data set can be complex and decomposing them into multiple parts will help in capturing more specific relationships. This process is actually the opposite to reducing data as you have to add new attributes based on the existing ones.

## 8. Join transactional and attribute data

Transactional data consists of events that snapshot specific moments, e.g. what was the price of the boots and the time when a user with this IP clicked on the Buy now button? Attribute data is more static, like user demographics or age and doesn’t directly relate to specific events. You may have several data sources or logs where these types of data reside. Both types can enhance each other to achieve greater predictive power. For instance, if you’re tracking machinery sensor readings to enable predictive maintenance, most likely you’re generating logs of transactional data, but you can add such qualities as the equipment model, the batch, or its location to look for dependencies between equipment behavior and its attributes.

CODING:

```
def load_csv(filepath):
data = []
col = []
checkcol = False
with open(filepath) as f:
for val in f.readlines():
val = val.replace(&quot;\n&quot;,&quot;&quot;)
val = val.split(&#39;,&#39;)
if checkcol is False:
col = val
checkcol = True
else:
data.append(val)
df = pd.DataFrame(data=data, columns=col)

return df
Output:
myData = load_csv(&#39;100 Sales Record.csv&#39;)
print(myData.head())
```

## Example:

from numpy import loadtxt

path = r"C:\pima-indians-diabetes.csv"

datapath= open(path, 'r')

data = loadtxt(datapath, delimiter=",")

print(data.shape)

print(data[:3])

Output:

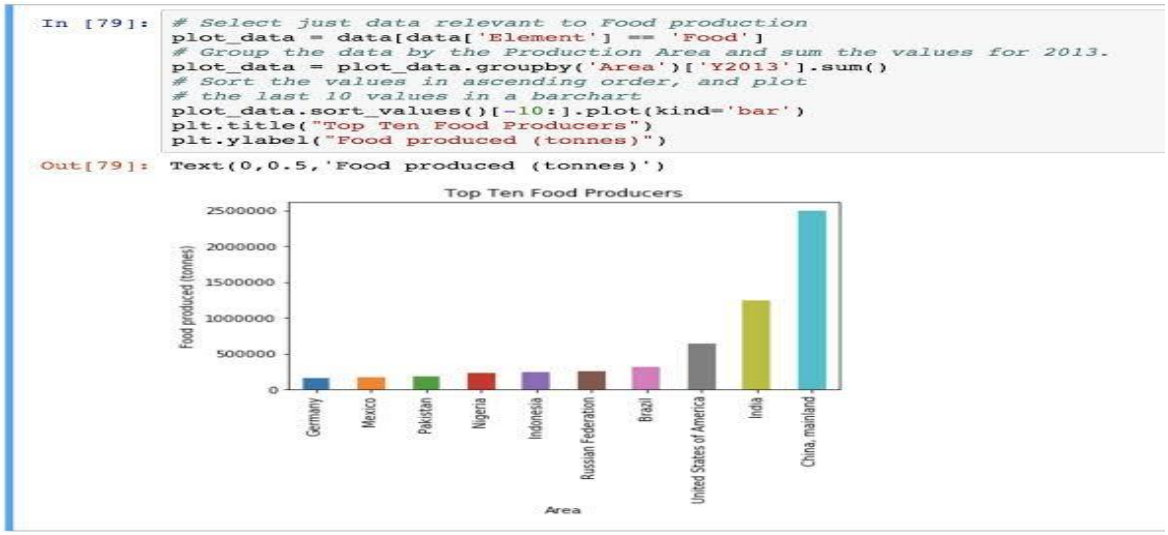
(768, 9)

[ [ 6. 148. 72. 35. 0. 33.6 0.627 50. 1.]

[ 1. 85. 66. 29. 0. 26.6 0.351 31. 0.]

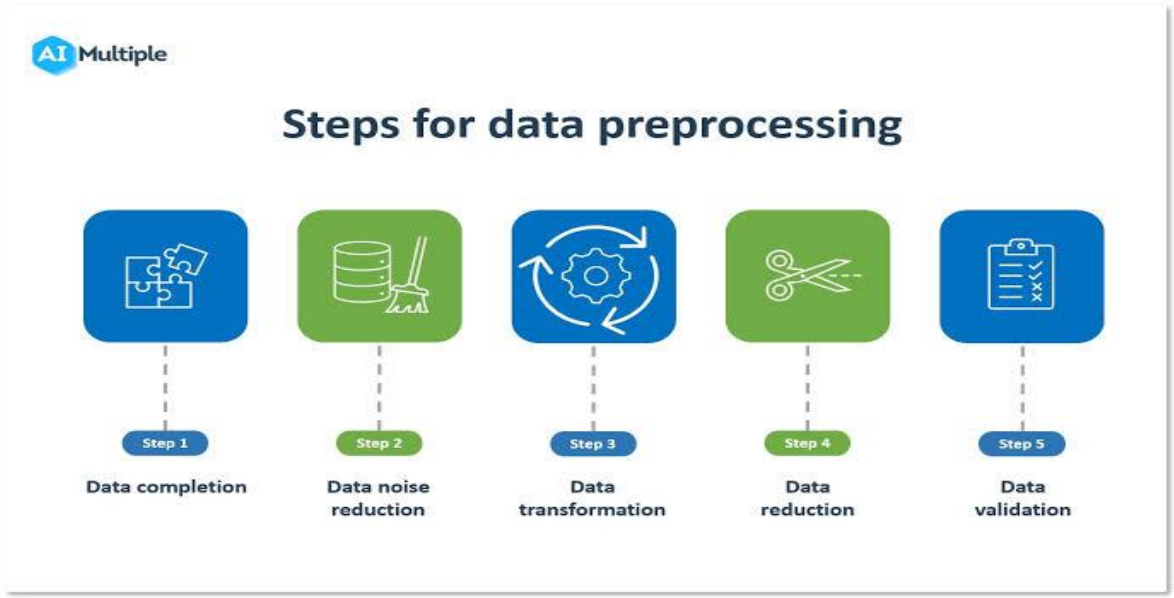
[ 8. 183. 64. 0. 0. 23.3 0.672 32. 1.]

```
In [79]: # Select just data relevant to Food production
         plot_data = data[data['Element'] == 'Food']
         # Group the data by the Production Area and sum the values for 2013.
         plot_data = plot_data.groupby('Area')['Y2013'].sum()
         # Sort the values in ascending order, and plot
         # the last 10 values in a barchart
         plot_data.sort_values()[-10:].plot(kind='bar')
         plt.title("Top Ten Food Producers")
         plt.ylabel("Food produced (tonnes)")
Out[79]: Text(0,0.5,'Food produced (tonnes)')
```



# ***PREPROCESS DATASHEETS:***

At the heart of Machine Learning is to process data. Your **machine learning tools are as good as the quality of your data**. This blog deals with the various steps of **cleaning data**. Your data needs to go through a few steps before it is could be used for making predictions.




.

## ***Steps involved in data preprocessing*** :

1. Importing the required Libraries
2. Importing the data set
3. Handling the missing data.
4. Encoding Categorical Data.
5. Splitting the data set into test set and training set.
6. Feature Scaling.

So let us look at these steps one by one.

### Step 1: Importing the required Libraries

To follow along you will need to download this dataset : **Data.csv**
Every time we make a new model, we will require to import Numpy and Pandas. Numpy is a Library which contains Mathematical functions and is used for scientific computing while Pandas is used to import and manage the data sets.

```
import pandas as pd
import numpy as np
```

Here we are importing the pandas and Numpy library and assigning a shortcut “pd” and “np” respectively.

## Step 2: Importing the Dataset

Data sets are available in .csv format. A CSV file stores tabular data in plain text. Each line of the file is a data record. We use the read_csv method of the pandas library to read a local CSV file as a **dataframe**.

```
dataset = pd.read_csv('Data.csv')
```

After carefully inspecting our dataset, we are going to create a matrix of features in our dataset (X) and create a dependent vector (Y) with their respective observations. To read the columns, we will use iloc of pandas (used to fix the indexes for selection) which takes two parameters — [row selection, column selection].

```
X = dataset.iloc[:, :-1].values
y = dataset.iloc[:, 3].values
```

## Step 3: Handling the Missing Data

An example of Missing data and Imputation

The data we get is rarely homogenous. Sometimes data can be missing and it needs to be handled so that it does not reduce the performance of our machine learning model.

To do this we need to replace the missing data by the Mean or Median of the entire column. For this we will be using the sklearn.preprocessing Library which contains a class called Imputer which will help us in taking care of our missing data.

```
from sklearn.preprocessing import Imputer
imputer = Imputer(missing_values = "NaN", strategy = "mean", axis = 0)
```

Our object name is **imputer.** The Imputer class can take parameters like :

**missing_values** : It is the placeholder for the missing values.

1. imputer = imputer.fit(X[:, 1:3]) All occurrences of missing_values will be imputed. We can give it an integer or “NaN” for it to find missing values.
2. **strategy** : It is the imputation strategy — If “mean”, then replace missing values using the mean along the axis (Column). Other strategies include “median” and “most_frequent”.
3. **axis** : It can be assigned 0 or 1, 0 to impute along columns and 1 to impute along rows.

Now we fit the imputer object to our data.

Now replacing the missing values with the mean of the column by using transform method.

```
X[:, 1:3] = imputer.transform(X[:, 1:3])
```

## Step 4: Encoding categorical data

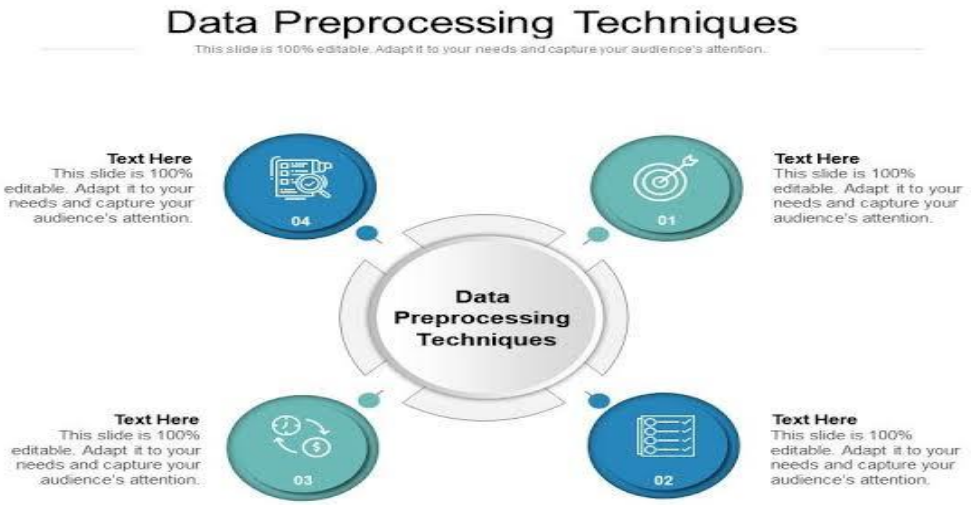
Converting Categorical data into dummy variables. Any variable that is not quantitative is categorical. Examples include Hair color, gender, field of study, college attended, political affiliation, status of disease infection.

But why encoding ?

We cannot use values like “Male” and “Female” in mathematical equations of the model so we need to encode these variables into numbers.

To do this we import “LabelEncoder” class from “sklearn.preprocessing” library and create an object labelencoder_X of the LabelEncoder class. After that we use the fittransform method on the categorical features.

After Encoding it is necessary to distinguish between between the variables in the same column, for this we will use OneHotEncoder class from sklearn.preprocessing library.




**One-Hot**

## Encoding

One hot encoding transforms categorical features to a format that works better with classification and regression algorithms. from sklearn.preprocessing import LabelEncoder, OneHotEncoder labelencoder_X = LabelEncoder()
X[:, 0] = labelencoder_X.fit_transform(X[:, 0])
onehotencoder = OneHotEncoder(categorical_features = [0])
X = onehotencoder.fit_transform(X).
toarray()labelencoder_y = LabelEncoder()
y = labelencoder_y.fit_transform(y)

## Step 5: Splitting the Data set into Training set and Test Set

Now we divide our data into two sets, one for training our model called the **training set** and the other for testing the performance of our model called the **test set**. The split is

generally 80/20. To do this we import the “train_test_split” method of “sklearn.model_selection” library.

```
from sklearn.model_selection import train_test_split
```

Now to build our training and test sets, we will create 4 sets —

1. **X_train** (training part of the matrix of features),
2. **X_test** (test part of the matrix of features),
3. **Y_train** (training part of the dependent variables associated with the X train sets, and therefore also the same indices) ,
4. **Y_test** (test part of the dependent variables associated with the X test sets, and therefore also the same indices).

We will assign to them the test_train_split, which takes the parameters — arrays (X and Y), test_size (Specifies the ratio in which to split the data set).

```
X_train, X_test, Y_train, Y_test = train_test_split( X , Y , test_size = 0.2, random_state = 0)
```

## Step 6: Feature Scaling

Most of the machine learning algorithms use the **Euclidean distance** between two data points in their computations . Because of this, **high magnitudes features will weigh more** in the distance calculations **than features with low magnitudes**. To avoid this Feature standardization or Z-score normalization is used. This is done by using “StandardScaler” class of “sklearn.preprocessing”.

```
from sklearn.preprocessing import StandardScaler
sc_X = StandardScaler()
```

## What Are the Different Types of Data Analysis?

Data analysis is an aspect of data science and data analytics that is all about analyzing data for different kinds of purposes. The data analysis process involves inspecting, cleaning, transforming and modeling data to draw useful insights from it.

 Descriptive analysis.

 Diagnostic analysis.

 Exploratory analysis.

 Inferential analysis.

 Predictive analysis.

 Causal analysis.

 Mechanistic analysis.

 Prescriptive analysis.

## DESCRIPTIVE ANALYSIS

 Descriptive analysis is the very first analysis performed in the data analysis process.

 It generates simple summaries about samples and measurements.

 It involves common, descriptive statistics like measures of central tendency, variability, frequency and position.

## DIAGNOSTIC ANALYSIS

Diagnostic analysis seeks to answer the question “Why did this happen?” by taking a more in-depth look at data to uncover subtle patterns. Here’s what you need to know:

 Diagnostic analysis typically comes after descriptive analysis, taking initial findings and investigating why certain patterns in data happen.

 Diagnostic analysis may involve analyzing other related data sources, including past data, to reveal more insights into current data trends.

 Diagnostic analysis is ideal for further exploring patterns in data to explain anomalies.

## EXPLORATORY ANALYSIS (EDA):

Exploratory analysis involves examining or exploring data and finding relationships between variables that were previously unknown. Here's what you need to know:

 EDA helps you discover relationships between measures in your data, which are not evidence for the existence of the correlation, as denoted by the phrase, "Correlation doesn't imply causation".

 It's useful for discovering new connections and forming hypotheses. It drives design planning and data collection.

## INFERENTIAL ANALYSIS

Inferential analysis involves using a small sample of data to infer information about a larger population of data.

The goal of statistical modeling itself is all about using a small amount of information to extrapolate and generalize information to a larger group. Here's what you need to know:

 Inferential analysis involves using estimated data that is representative of a population and gives a measure of uncertainty or standard deviation to your estimation.

 The accuracy of inference depends heavily on your sampling scheme. If the sample isn't representative of the population, the generalization will be inaccurate. This is known as the central limit theorem.

## PREDICTIVE ANALYSIS

Predictive analysis involves using historical or current data to find patterns and make predictions about the future. Here's what you need to know:

 The accuracy of the predictions depends on the input variables.

 Accuracy also depends on the types of models. A linear model might work well in some cases, and in other cases it might not.

 Using a variable to predict another one doesn't denote a causal relationship.

## CAUSAL ANALYSIS

Causal analysis looks at the cause and effect of relationships between variables and is focused on finding the cause of a correlation. Here's what you need to know:

 To find the cause, you have to question whether the observed correlations driving your conclusion are valid. Just looking at the surface data won't help you discover the hidden mechanisms underlying the correlations.

 Causal analysis is applied in randomized studies focused on identifying causation.

 Causal analysis is the gold standard in data analysis and scientific studies where the cause of phenomenon is to be extracted and singled out, like separating wheat from chaff.

 Good data is hard to find and requires expensive research and studies. These studies are analyzed in aggregate (multiple groups), and the observed relationships are just average effects (mean) of the whole population. This means the results might not apply to everyone.

## MECHANISTIC ANALYSIS

Mechanistic analysis is used to understand exact changes in variables that lead to other changes in other variables. Here's what you need to know:

 It's applied in physical or engineering sciences, situations that require high precision and little room for error, only noise in data is measurement error.

 It's designed to understand a biological or behavioral process, the pathophysiology of a disease or the mechanism of action of an intervention.

## PRESCRIPTIVE ANALYSIS

Prescriptive analysis compiles insights from other previous data analyses and determines actions that teams or companies can take to prepare for predicted trends. Here's what you need to know:

 Prescriptive analysis may come right after predictive analysis, but it may involve combining many different data analyses.

 Companies need advanced technology and plenty of resources to conduct prescriptive analysis. AI systems that process data and adjust automated tasks are an example of the technology required to perform prescriptive analysis.

## Numeric data:

Data = [1, 3, 4, 5, 6, 2, 9]

# Creating series with default index values

s = pd.Series(Data)

# predefined index values

Index = ['a', 'b', 'c', 'd', 'e', 'f', 'g']