Data Science with Python

Introduction to Data Science

Agenda

01 What is Data Science?

02 Why do we need DS?

03 Data Science Process

04 Data Gathering

05 Data Processing

06 Data Analysis

07 Data Cleaning

08 Data Visualization

09 Creating a Model

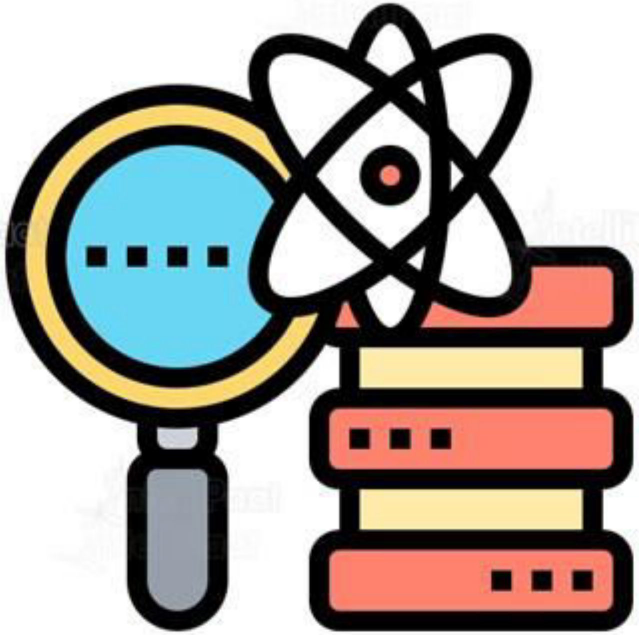
10 Testing the Model

11 Hands-on: Logistic Regression

What is Data Science?

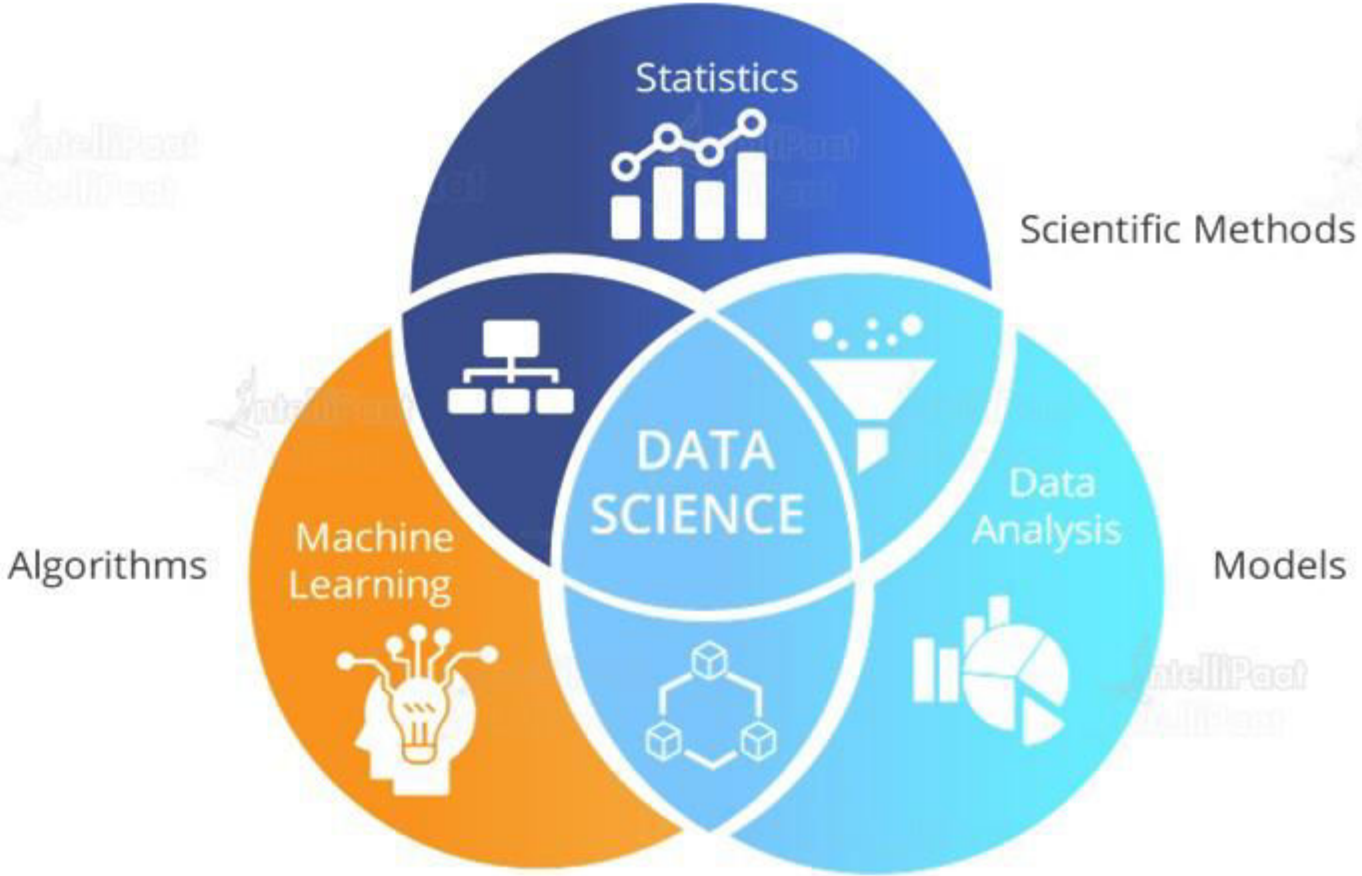
What is Data Science?

Data Science is the process of finding hidden patterns from the raw/unstructured data



What is Data Science?

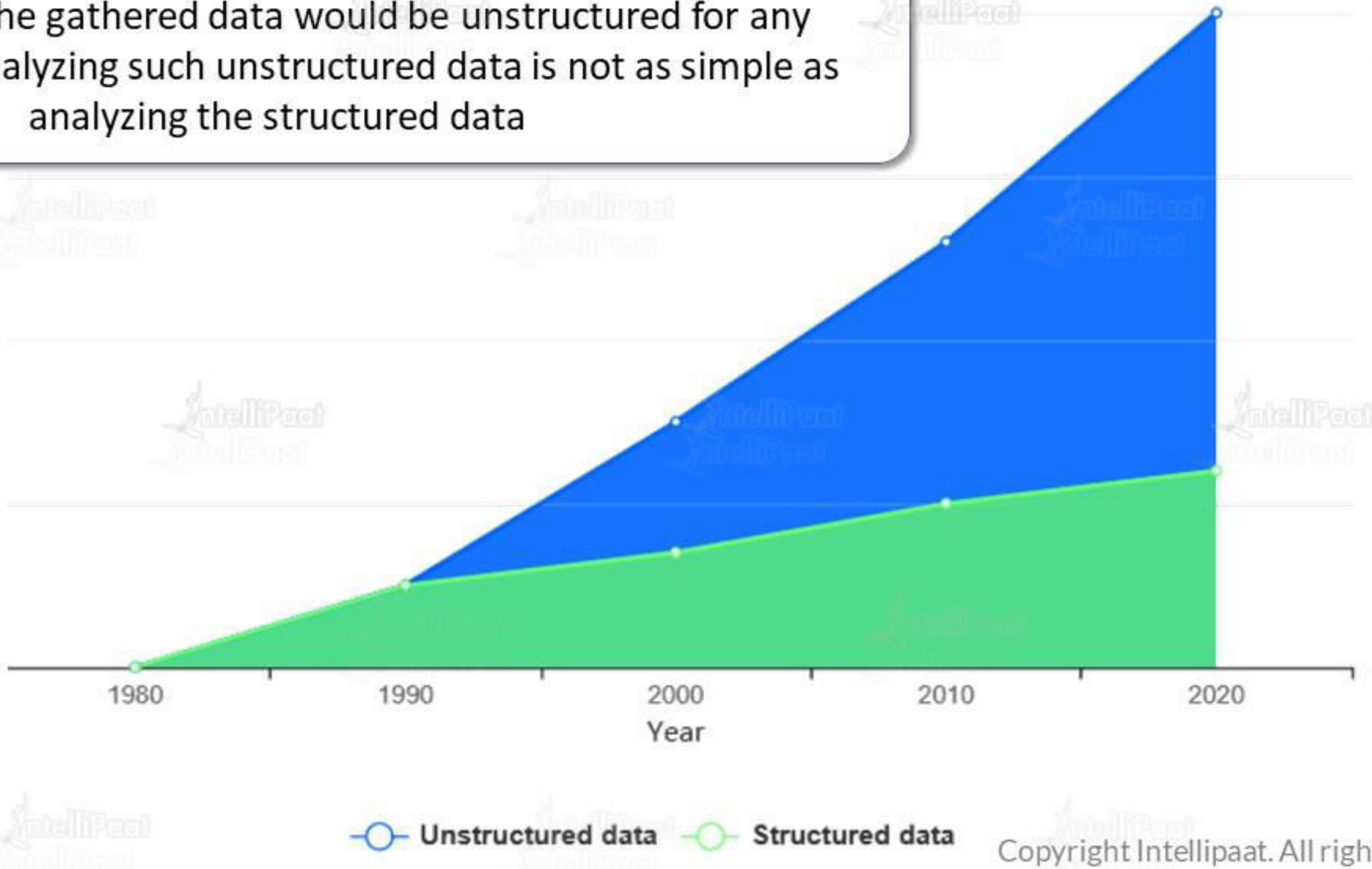
This Venn diagram gives us an idea of Data Science



Why do we need Data Science?

Why do we need Data Science?

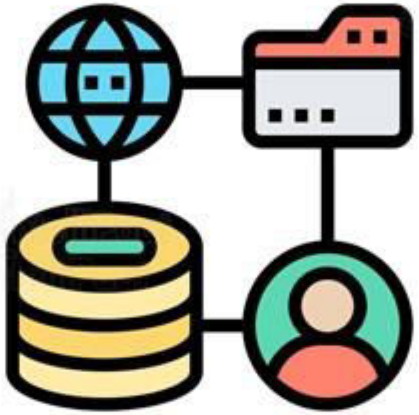
80 percent of the gathered data would be unstructured for any company, and analyzing such unstructured data is not as simple as analyzing the structured data



Why do we need Data Science?

The incoming data is from different data sources, and we can directly put it into a BI tool as we are not capable of handling this variety of data

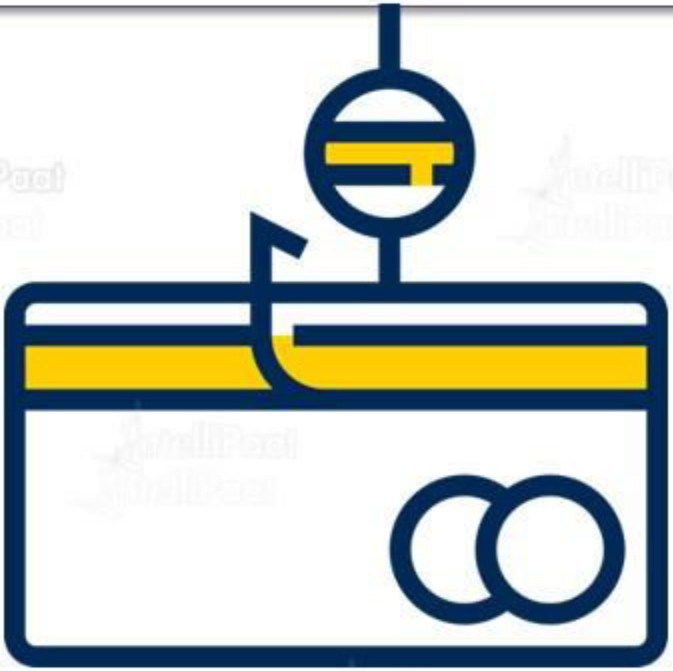
To handle large amounts of data—structured or unstructured—and to draw meaningful trends, we need Data Science



Why do we need Data Science?

Let us take a look at a real-time use case: **Credit Card Fraud Detection**

This model is used to check whether a credit card transaction is fraud or not. The aim is to detect all fraudulent transactions



Why do we need Data Science?

More Data Science Use Cases

01 Social media analytics

02 Predictive analysis

03 Targeted ads

04 Augmented reality

05 Recommendation engines

06 Healthcare imaging

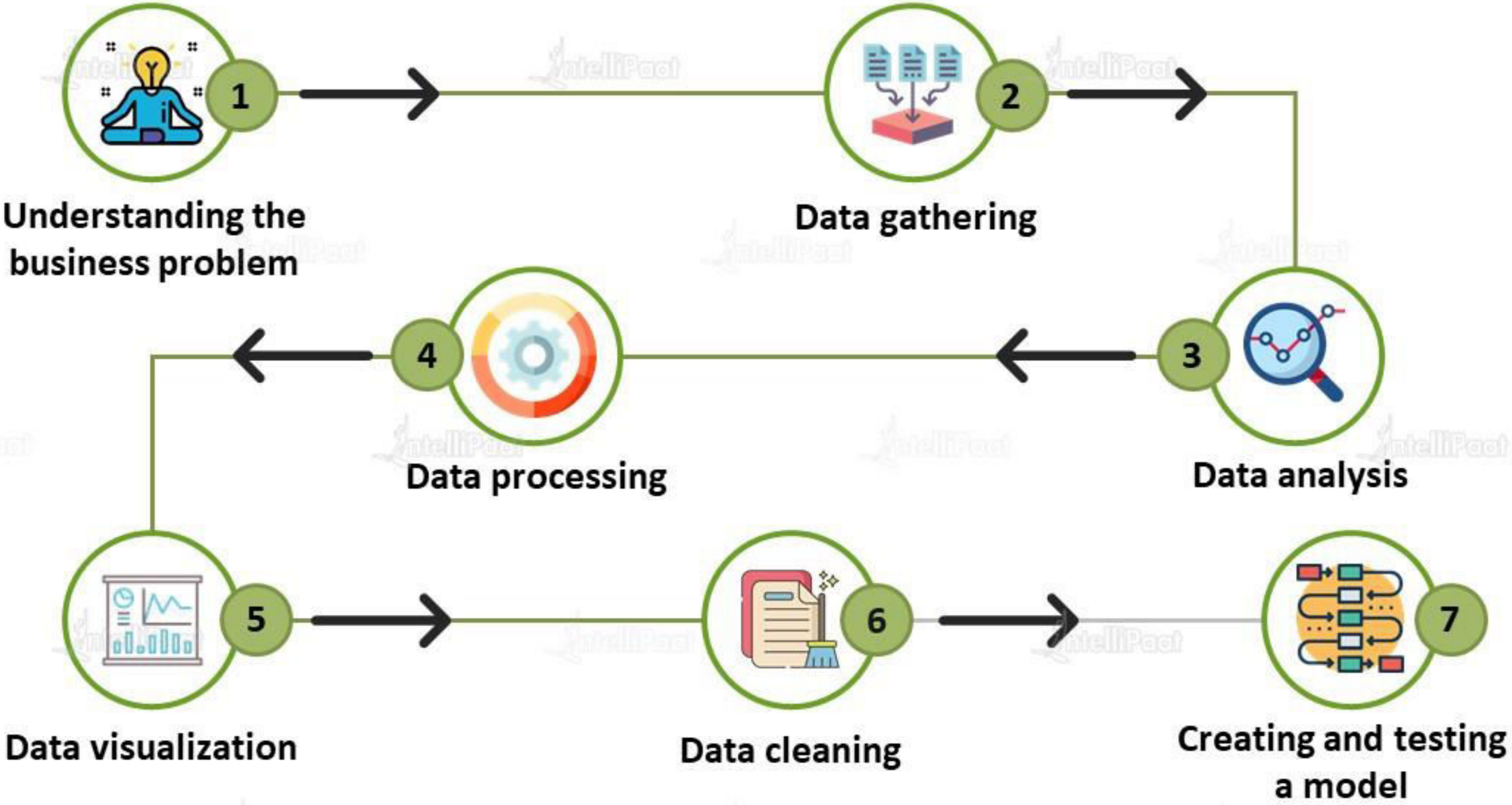
Why do we need Data Science?

Commonly Used DS Algorithms

1. Linear Regression
2. Logistic Regression
3. Decision Tree
4. Naive Bayes
5. KNN (K-nearest neighbors)
6. K-Means Clustering
7. Random Forest

Data Science Process

Data Science Process



Data Science Process

IntelliPaat

The first step in a Data Science solution is understanding the problem
Let us look at some pointers on how to understand the problem

01 First, ask the **Why** questions about the problem

02 Understand the end product required

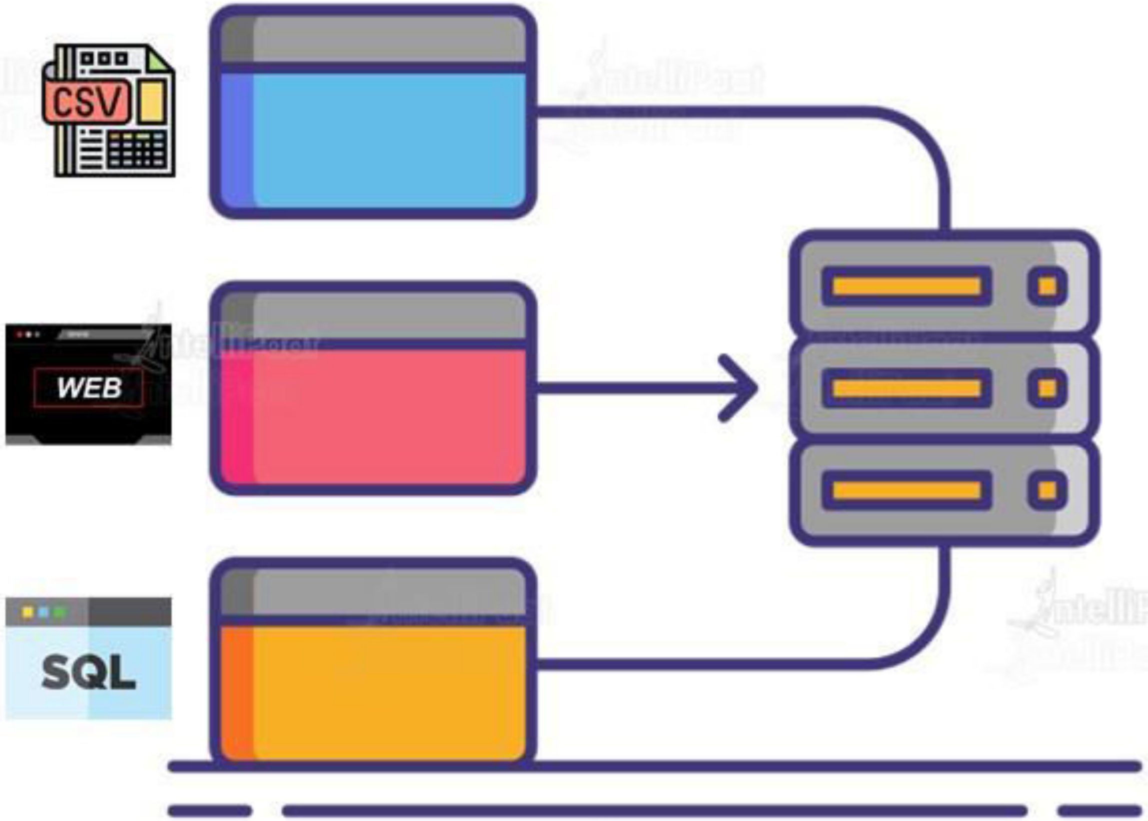
03 Determine the data sources for this problem

04 Gather all information and context required to solve this problem

Data Gathering

Data Gathering

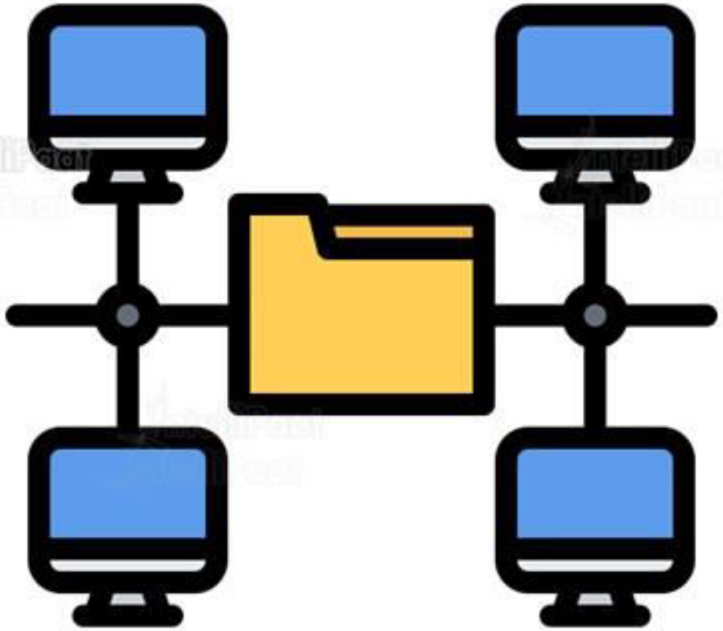
Data extraction is the process of retrieving data from various sources to be used in our Data Science process



Data Gathering

Data extraction is performed in order to gather data from diverse sources and store it in a data repository

This data can later be cleaned and transformed to be used to derive important insights or to make predictions

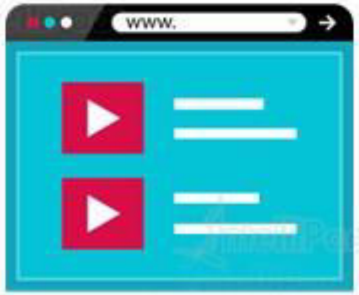


Data Gathering

Data can be extracted from various sources to be used in Data Science for further processing. Some of these sources are:

Databases

Internet



Data Processing

Data Processing

Data processing is the process of converting data into easily readable formats, which are more organized

Below are the popular Python libraries used for data manipulation

Data Processing

01 NumPy

Numerical Python (NumPy) is a very popular Python library. The purpose of the NumPy library is to do scientific computation and apply it to Python applications

02 Pandas

Pandas is a simple yet powerful and open-source data analysis and manipulation tool built on top of Python. We can achieve the same result by writing 1–2 lines in Pandas compared to the native Python

Data Processing

pandas

- Performs better than NumPy for 500k rows or more as it allows us to reorder a complete dataset using data wrangling operations, which are not available in NumPy
- Pandas Series Object is more flexible as we can define our own labeled index

NumPy

- Performs better for 50k rows or less; it suits for recursive or vectorized operations over small sets of data
- Elements in NumPy arrays are accessed by their default indexed position

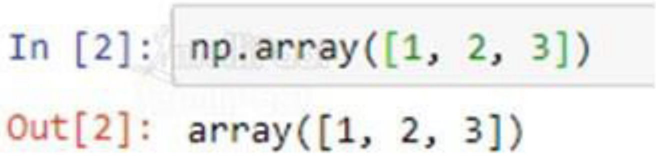
Data Processing

NumPy

The first step to create a NumPy array is to import the NumPy package

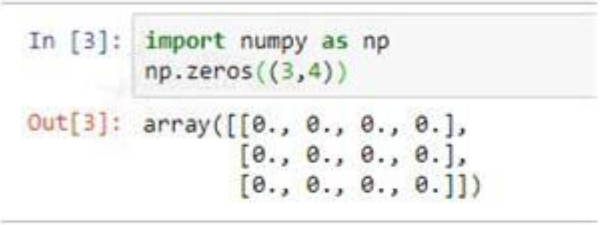
import numpy as np

Different ways to create a NumPy array



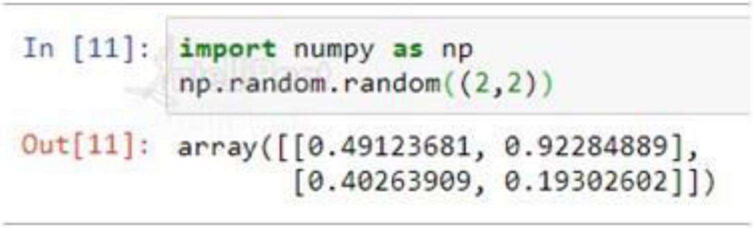
```
In [2]: np.array([1, 2, 3])

Out[2]: array([1, 2, 3])
```



```
In [3]: import numpy as np
        np.zeros((3,4))

Out[3]: array([[0., 0., 0., 0.],
               [0., 0., 0., 0.],
               [0., 0., 0., 0.]])
```



```
In [11]: import numpy as np
         np.random.random((2,2))

Out[11]: array([[0.49123681, 0.92284889],
                [0.40263909, 0.19302602]])
```

Data Processing

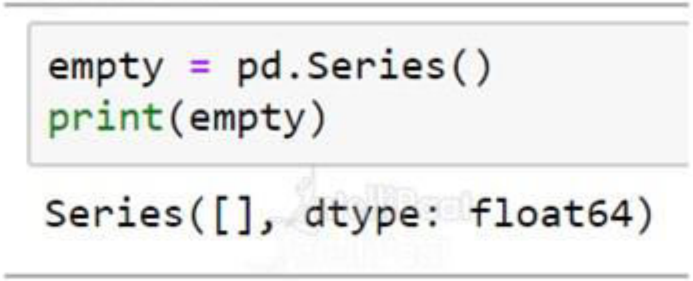
pandas

import pandas as pd

01 Series

- One-dimensional labeled array
- Can have any data type, but all elements of a single array should be of the same type

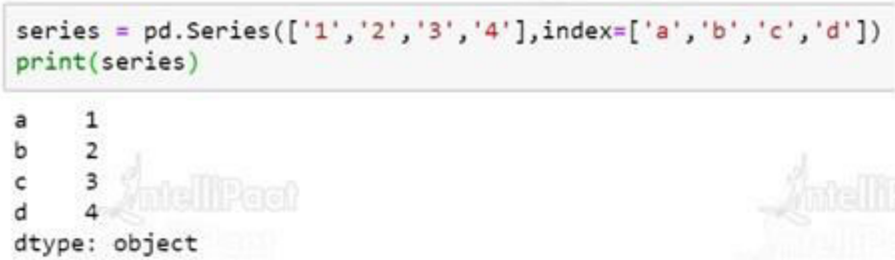
Creating an empty series:



```
empty = pd.Series()
print(empty)
```

```
Series([], dtype: float64)
```

Changing the index name:



```
series = pd.Series(['1','2','3','4'],index=['a','b','c','d'])
print(series)
```

```
a    1
b    2
c    3
d    4
dtype: object
```

Data Processing

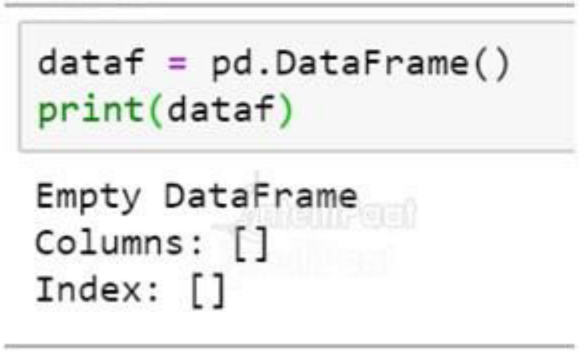
pandas

import pandas as pd

02 DataFrame

- Two-dimensional tabular structure
- Columns can hold different data types
- Size is mutable

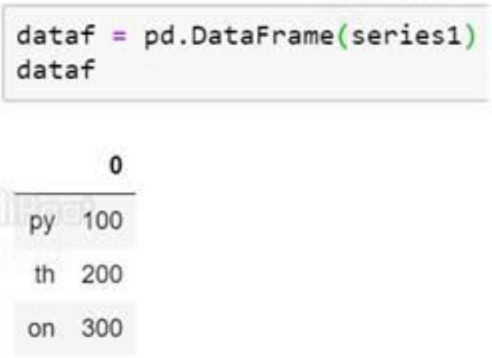
Creating an empty DataFrame:



```
dataf = pd.DataFrame()
print(dataf)
```

```
Empty DataFrame
Columns: []
Index: []
```

Converting a series to a DataFrame:



```
dataf = pd.DataFrame(series1)
dataf
```

	0
py	100
th	200
on	300

Hands-on: Data Processing and Manipulation

Merge, Join, and Concatenate

Merge, Join, and Concatenate

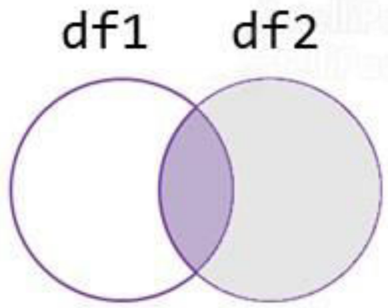
Merge and Join combines the given data to a new DataFrame based on a common column. **Concatenation** combines the data of multiple DataFrames without any gap

When we join/merge two DataFrames together, the df1 data is shown in one column beside the column with the df2 data in the same row

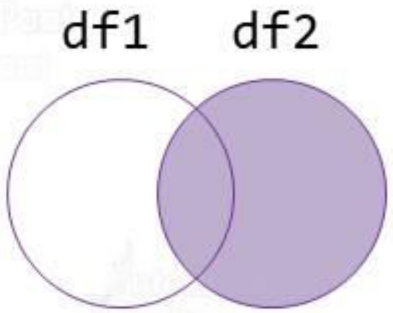
Function Names
merge()
join()
concat()

Merge, Join, and Concatenate

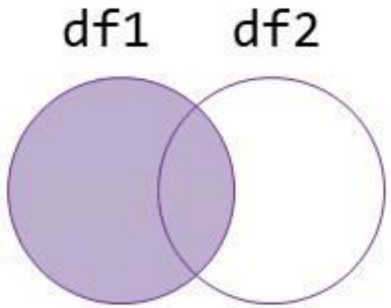
Types of Merges/Joins



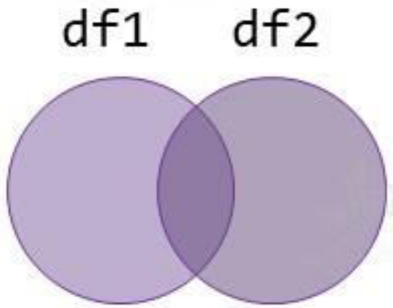
Inner Merge/
Inner Join



Right Merge/
Right Join



Left Merge/
Left Join



Outer Merge/
Outer Join

Hands-on: Merge, Join, and Concatenate

Data Analysis

Data Analysis

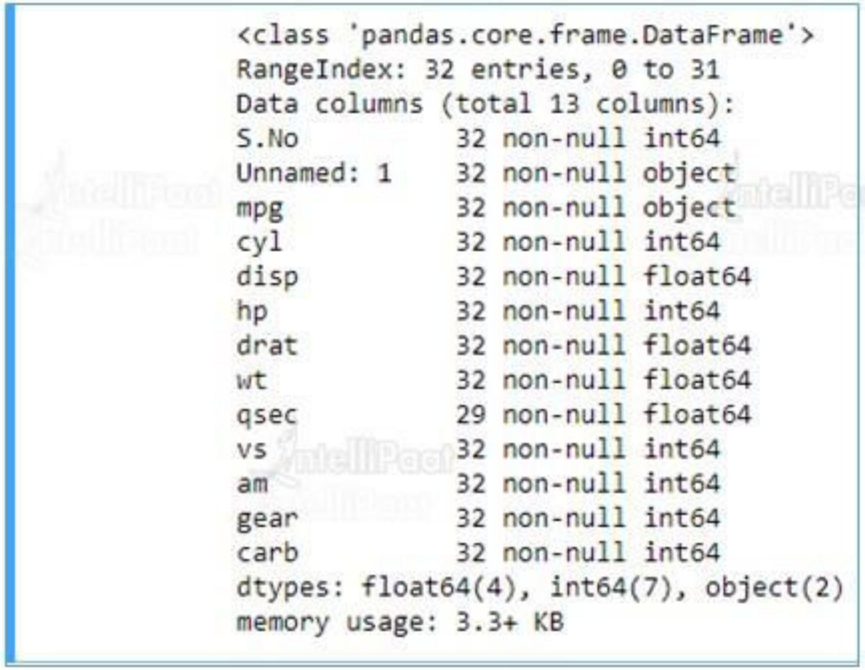
Importing a CSV File

```
variable = pd.read_csv("filename.csv")
```

Getting information about a DataFrame

```
df1.info(null_counts=True)
```

Using **null_counts=True** is to display all information about every column available



```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 32 entries, 0 to 31
Data columns (total 13 columns):
S.No           32 non-null int64
Unnamed: 1     32 non-null object
mpg            32 non-null object
cyl            32 non-null int64
disp           32 non-null float64
hp             32 non-null int64
drat           32 non-null float64
wt             32 non-null float64
qsec           29 non-null float64
vs             32 non-null int64
am             32 non-null int64
gear           32 non-null int64
carb           32 non-null int64
dtypes: float64(4), int64(7), object(2)
memory usage: 3.3+ KB
```

Data Analysis

Other Analysis Functions

Function	Description
.count()	Returns the non-null records in each column
.describe()	Gives the descriptive statistical summary of a DataFrame
.mean()	Returns the mean of a column
.median()	Returns the median of a column
.std()	Returns the standard deviation of a column
.min()	Returns the minimum of each attribute (column)
.max()	Returns the maximum of each attribute (column)

Hands-on: Data Analysis

Data Cleaning

Data Cleaning

Data cleaning/cleansing is the process of removing unwanted or inaccurate records from a table or a dataset. Analysis made on clean data is more accurate

Cleansing Functions

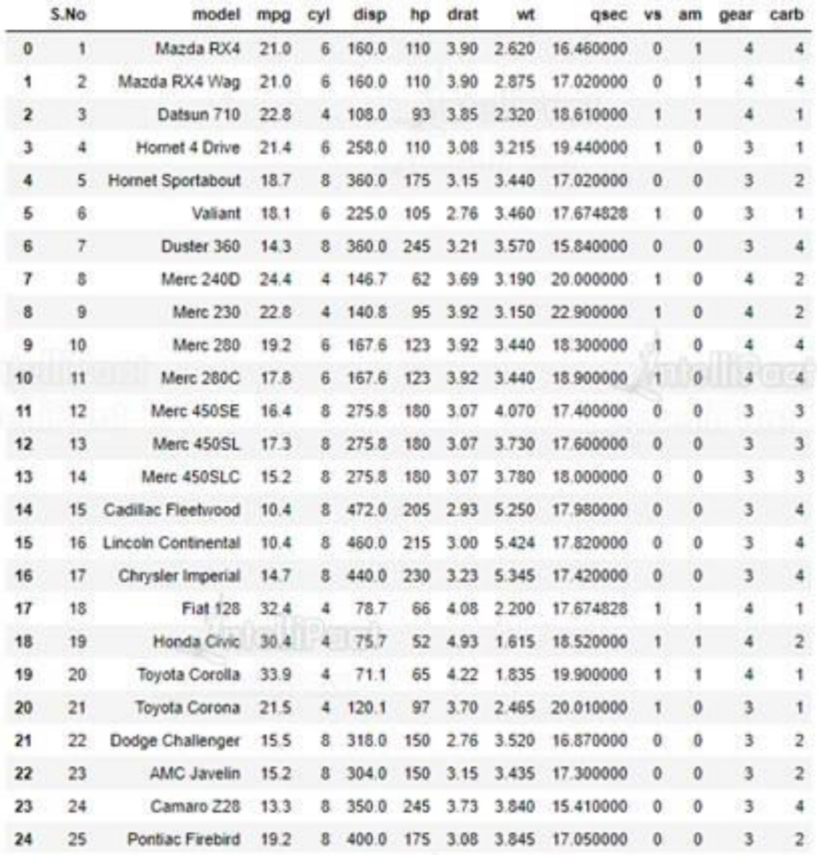
Function	Description
.rename()	Renames a column
.fillna()	Fills the null or empty cells with the mean value
.drop()	Drops the mentioned column
.corr()	Finds the correlation matrix
.astype()	Changes the data type of a column

Hands-on: Data Cleaning

Data Visualization

Data Visualization

Data visualization is the graphical/pictorial representation of information and data



	S.No	model	mpg	cyl	disp	hp	drat	wt	qsec	vs	am	gear	carb
0	1	Mazda RX4	21.0	6	160.0	110	3.90	2.620	16.460000	0	1	4	4
1	2	Mazda RX4 Wag	21.0	6	160.0	110	3.90	2.875	17.020000	0	1	4	4
2	3	Datsun 710	22.8	4	108.0	93	3.85	2.320	18.610000	1	1	4	1
3	4	Hornet 4 Drive	21.4	6	258.0	110	3.08	3.215	19.440000	1	0	3	1
4	5	Hornet Sportabout	18.7	8	360.0	175	3.15	3.440	17.020000	0	0	3	2
5	6	Valiant	18.1	6	225.0	105	2.76	3.460	17.674828	1	0	3	1
6	7	Duster 360	14.3	8	360.0	245	3.21	3.570	15.840000	0	0	3	4
7	8	Merc 240D	24.4	4	146.7	62	3.69	3.190	20.000000	1	0	4	2
8	9	Merc 230	22.8	4	140.8	95	3.92	3.150	22.900000	1	0	4	2
9	10	Merc 280	19.2	6	167.6	123	3.92	3.440	18.300000	1	0	4	4
10	11	Merc 280C	17.8	6	167.6	123	3.92	3.440	18.900000	1	0	4	4
11	12	Merc 450SE	16.4	8	275.8	180	3.07	4.070	17.400000	0	0	3	3
12	13	Merc 450SL	17.3	8	275.8	180	3.07	3.730	17.600000	0	0	3	3
13	14	Merc 450SLC	15.2	8	275.8	180	3.07	3.780	18.000000	0	0	3	3
14	15	Cadillac Fleetwood	10.4	8	472.0	205	2.93	5.250	17.980000	0	0	3	4
15	16	Lincoln Continental	10.4	8	460.0	215	3.00	5.424	17.820000	0	0	3	4
16	17	Chrysler Imperial	14.7	8	440.0	230	3.23	5.345	17.420000	0	0	3	4
17	18	Fiat 128	32.4	4	78.7	66	4.08	2.200	17.674828	1	1	4	1
18	19	Honda Civic	30.4	4	75.7	52	4.93	1.615	18.520000	1	1	4	2
19	20	Toyota Corolla	33.9	4	71.1	65	4.22	1.835	19.900000	1	1	4	1
20	21	Toyota Corona	21.5	4	120.1	97	3.70	2.465	20.010000	1	0	3	1
21	22	Dodge Challenger	15.5	8	318.0	150	2.76	3.520	16.870000	0	0	3	2
22	23	AMC Javelin	15.2	8	304.0	150	3.15	3.435	17.300000	0	0	3	2
23	24	Camaro Z28	13.3	8	350.0	245	3.73	3.840	15.410000	0	0	3	4
24	25	Pontiac Firebird	19.2	8	400.0	175	3.08	3.845	17.050000	0	0	3	2

25%
50%
65%
75%
20%
50%
100%
Name 1
Name 2
Name 3
Name 4
Name 5
Name 6
40%
70%
20%
80%
60%
0%
100%
Lorem ipsum dolor sit amet, consectetuer adipiscing eli.

Data Visualization

IntelliPaat

Why should we visualize data?

01 To view changes happening over time seamlessly using a visual aid rather than plain data

02 To discover correlations among two or more variables easily

03 To simplify complex information into user-friendly formats

04 To tell a better story with a bunch of pictures over time

Data Visualization

Popular Data Visualization Libraries

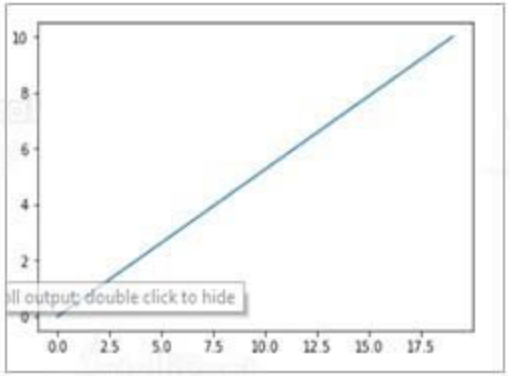
Data Visualization

Matplotlib	VS	Seaborn
Used for basic plotting and contains bars, lines, and pies		Has more interesting default themes and needs fewer syntax
A graphics package for data visualization and can mirror MATLAB		Better integration with Pandas and also extends Matplotlib for better graphics
Can open multiple figures at once, but needs to close them together		Automated creation of multiple figures is available
Works very well with DataFrames and arrays and has a set of helpful APIs		Works with the whole dataset instead of working with data structures
Has great customization features		Provides only commonly used templates as default but saves time

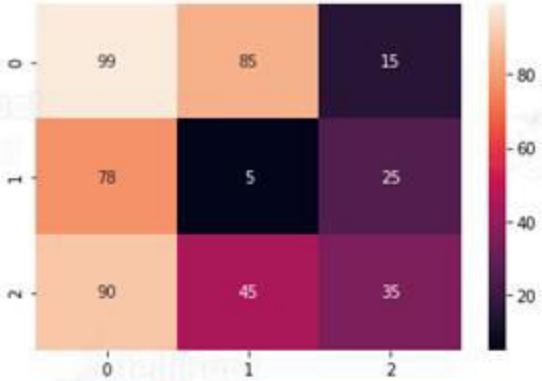
Data Visualization

Matplotlib Plots

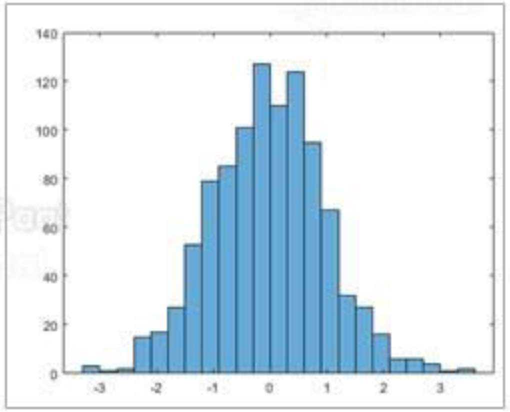
Seaborn Visuals



Line Plot

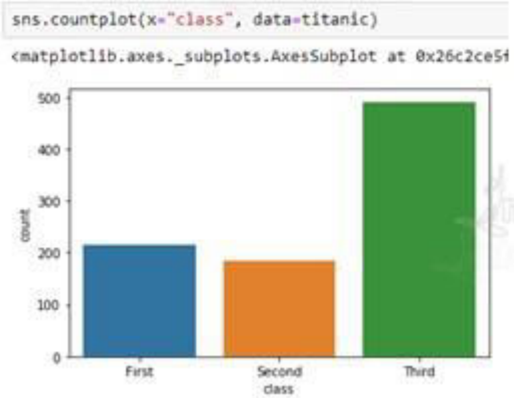


Heat Map



Histogram

Countplot



```
sns.countplot(x="class", data=titanic)
```

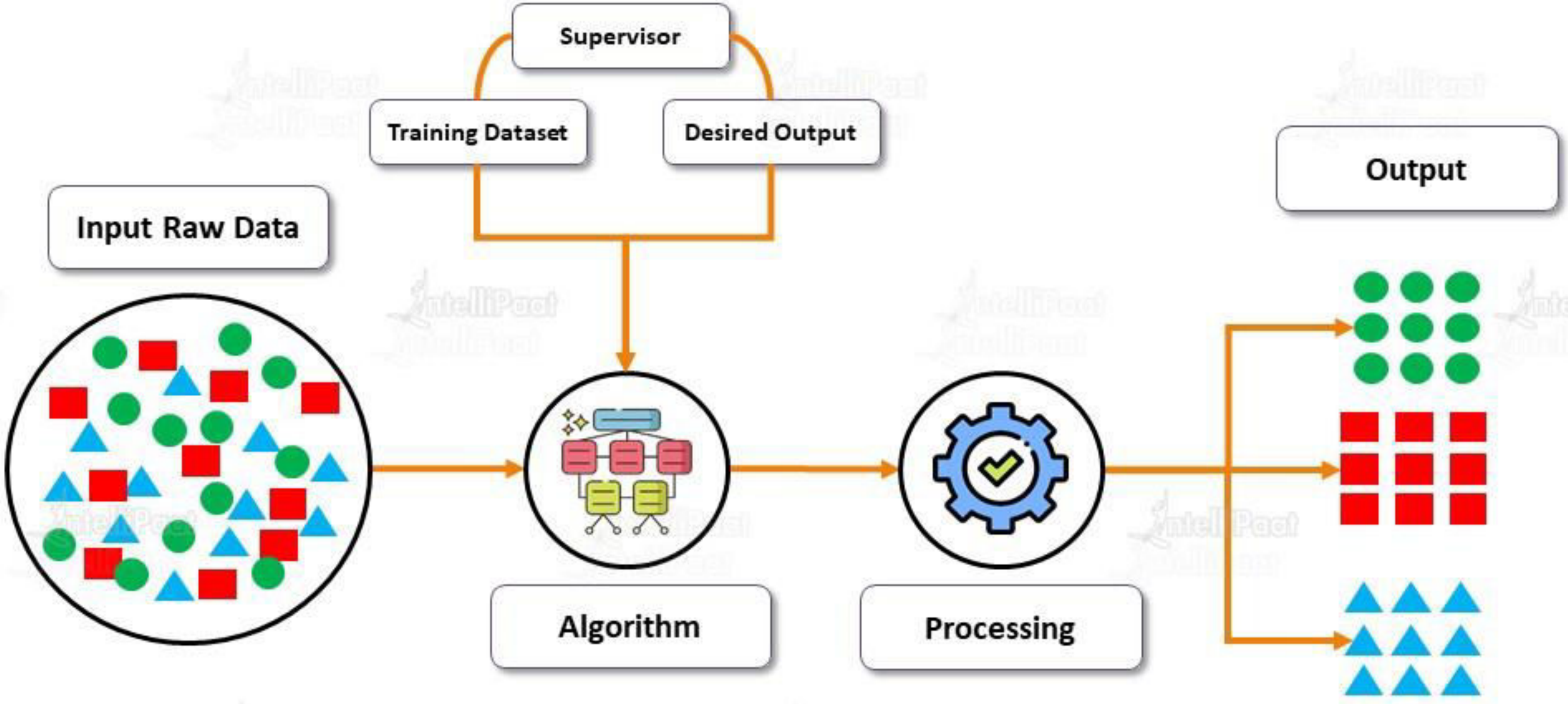
```
<matplotlib.axes._subplots.AxesSubplot at 0x26c2ce5f
```

Hands-on: Data Visualization

Creating a Model

Creating a Model

An ML model is a mathematical model that finds patterns when we feed raw data to it. Models are developed for specific use cases



Creating a Model

We have to choose an appropriate algorithm and start training it with a subset of the dataset that we have. Also, we have to segregate between training and testing data

Majority of the dataset is used for training

Testing the Model

Testing the Model

Testing a model refers to testing its performance and accuracy by providing the machine with new datasets/test datasets and comparing its accuracy with the existing model

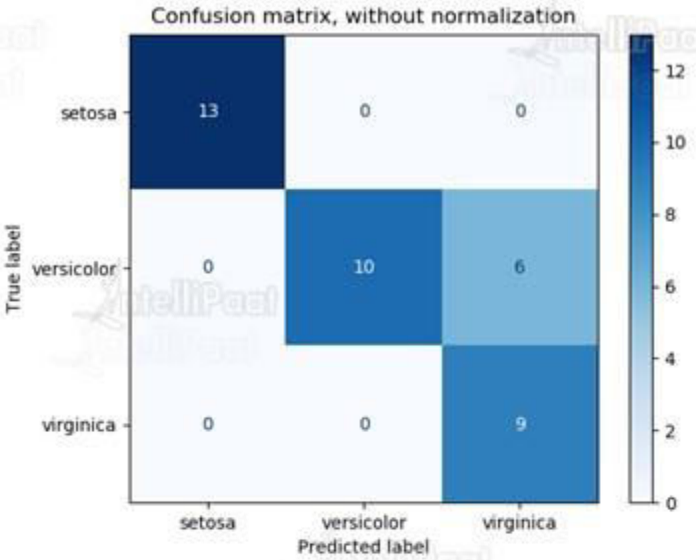
Testing the Model

Confusion Matrix

A confusion matrix is a table layout that allows us to visualize the performance of an algorithm

Accuracy Score

Accuracy score is equal to the percentage of rows in the testing data that are correctly classified



Predicted/Classified

		Negative	Positive
Actual	Negative	998	0
	Positive	1	1

Hands-on: Logistic Regression

IntelliPaat

India: +91-7847955955

US: 1-800-216-8930 (TOLL FREE)

support@intellipaat.com

24/7 Chat with Our Course Advisor