DATA SCIENCE

• INTRODUCTION TO DATA SCIENCE

But first, let’s start by understanding what is data science?
How does Netflix suggest us shows? How does Amazon suggest you to buy certain stuff? How do they do it? These are the few questions we ponder from time to time. In reality, doing such tasks are impossible without the availability of data. Data science is all about using data to solve problems. The problem could be decision making such as identifying which email is spam and which is not. Or a product recommendation such as which movie to watch? Or predicting the outcome such as who will be the next President of the USA? So, the core job of a data scientist is to understand the data, extract useful information out of it and apply this in solving the problems.

• APPLICATIONS OF DATA SCIENCE

1. Banking

Banking is one of the biggest applications of Data Science. Big Data and Data Science have enabled banks to keep up with the competition. With Data Science, banks can manage their resources efficiently, furthermore, banks can make smarter decisions through fraud detection, management of customer data, risk modelling, real-time predictive analytics, customer segmentation, etc.

2. Manufacturing

In the 21st century, Data Scientists are the new factory workers. That means that data scientists have acquired a key position in the manufacturing industries. Data Science is being extensively used in manufacturing industries for optimizing production, reducing costs and boosting the profits.

3. Transport

Another important application of data science is transport. In the transportation sector, Data Science is actively making its mark in making safer driving environments for the drivers. It is also playing a key role in optimizing vehicle performance and adding greater autonomy to the drivers. Furthermore, in the transport sector, Data Science has actively increased its manifold with the **introduction of self-driving cars**

4. Healthcare

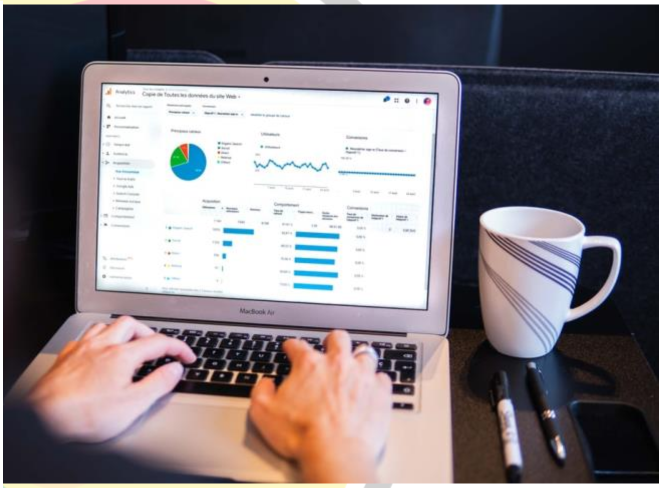
In the health-care industry, data science is making great leaps. The various industries in health-care making use of data science are –

a) Medical Image Analysis
b) Genetics and Genomics
c) Drug Discovery
d) Predictive Modelling for Diagnosis

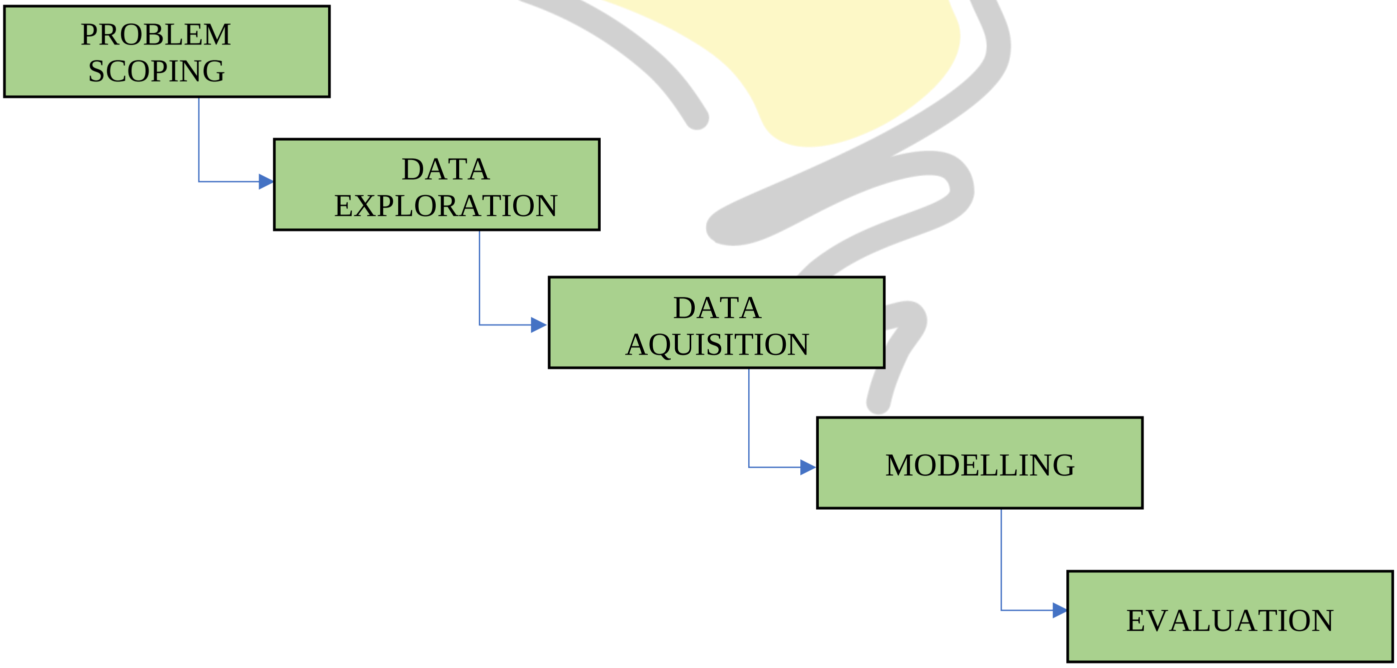
5. E-commerce

E-commerce and retail industries have been hugely benefitted by data science. Some of the ways in which data science has transformed the e-commerce industries are-

a) For identifying a potential customer base, data science is being heavily utilized.
b) Usage of predictive analytics for forecasting the goods and services.



• REVISITING AI PROJECT CYCLE


PROBLEM SCOPING

DATA EXPLORATION

DATA AQUISITION

MODELLING

EVALUATION

Problem Scoping	Understanding the problem
Data Acquisition	Collecting accurate and reliable data
Data Exploration	Arranging the data uniformly
Modelling	Creating Models from the data
Evaluation	Evaluating the project

*Example – **Creating a birthday card***

1. **Checking** the factors like budget, number of people, place, food etc Which will help us decide the next steps and understanding the Project.
2. **Acquiring** data from different sources like internet, from friends etc for Designs, ideas and suggestions.
3. Making a list of the gathered data and choosing the required specifications according to the budget.
4. Creating or **Modelling** a card on the basis of the data collected as what will be written in the card and what all pictures can be presented in it.
5. Showing it to Parents or cousins to Let them check it or **evaluate** it.

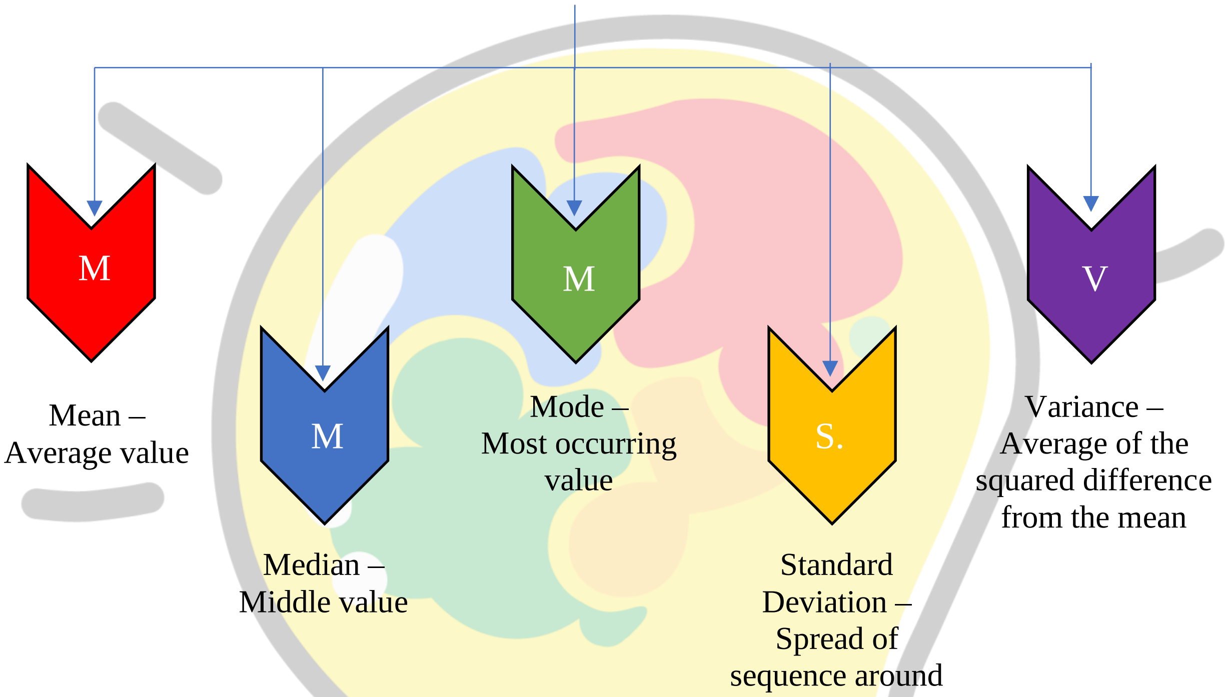
• PYTHON FOR DATA SCIENCE

Unlike some other programming languages, in Python, there is generally a best way of doing something. The three best and most important Python libraries for data science are NumPy, Pandas, and Matplotlib.

1. NumPy —A library that makes a variety of mathematical and statistical operations easier; it is also the basis for many features of the pandas library.
2. Pandas — A Python library created specifically to facilitate working with data, this is the bread and butter of a lot of Python data science work.
3. Matplotlib — A visualization library that makes it quick and easy to generate charts from your data.
4. Scikit-learn — The most popular library for machine learning work in Python.

• STATISTICS WITH PYTHON

We have already understood that Data Sciences works around analysing data and performing tasks around it. For analysing the numeric & alpha-numeric data used for this domain, mathematics comes to our rescue. Basic statistical methods used in mathematics come quite handy Python too for analysing and working around such datasets. Statistical tools widely used in Python are:



Mean – Average value

Median – Middle value

Mode – Most occurring value

Standard Deviation – Spread of sequence around

Variance – Average of the squared difference from the mean

• DATA VISUALISATION

While collecting data, it is possible that the data might come with some errors. Let us first take a look at the types of issues we can face with data:

1. Erroneous Data: There are two ways in which the data can be erroneous:

• Incorrect values: The values in the dataset (at random places) are incorrect. For example, in the column of phone number, there is a decimal value or in the marks column, there is a name mentioned, etc. These are incorrect values that do not resemble the kind of data expected in that position.

• Invalid or Null values: At some places, the values get corrupted and hence they become invalid. Many times you will find NaN values in the dataset. These are null values which do not hold any meaning and are not processible. That is why, these values (as and when encountered) are removed from the database.

2. Missing Data: In some datasets, some cells remain empty. The values of these cells are missing and hence the cells remain empty. Missing data cannot be interpreted as an error as the values here are not erroneous or might not be missing because of any error.

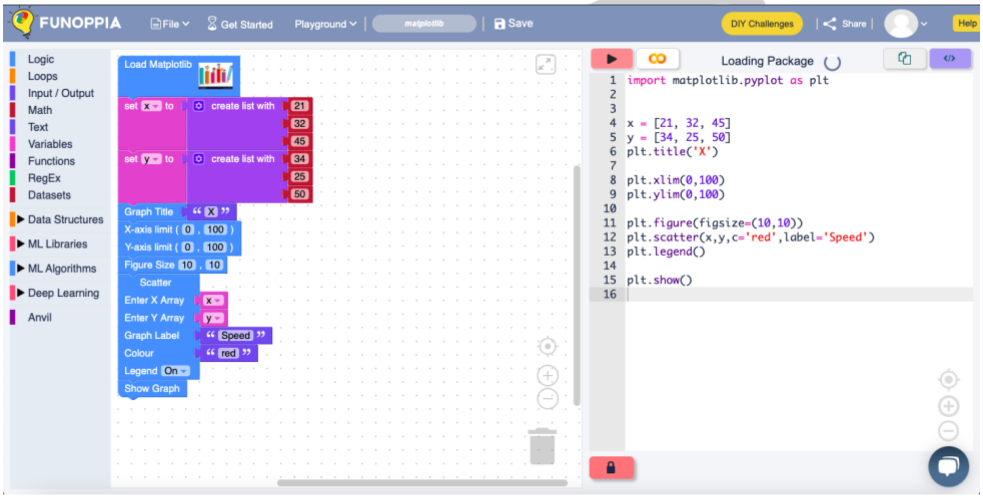
3. Outliers: Data which does not fall in the range of a certain element are referred to as outliers. To understand this better, let us take an example of marks of students in a class. Let us assume that a student was absent for exams and hence has got 0 marks in it. If his marks are taken into account, the whole class's average would go down. To prevent this, the average is taken for the range of marks from highest to lowest keeping this particular result separate. This makes sure that the average marks of the class are true according to the data. Analysing the data collected can be difficult as it is all about tables and numbers. While machines work efficiently on numbers, humans need visual aid to understand and comprehend the information passed. Hence, data visualisation is used to interpret the data collected and identify patterns and trends out of it.

In Python, Matplotlib package helps in visualising the data and making some sense out of it. As we have already discussed before, with the help of this package, we can plot various kinds of graphs. Let us discuss some of them here:

- Scatter Plot
- Bar Chart
- Histogram
- Box Plots

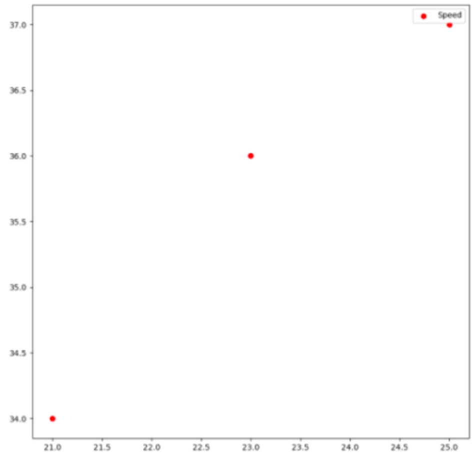
1. SCATTER PLOTS

Scatter plots are used to plot discontinuous data; that is, the data which does not have any continuity in flow is termed as discontinuous. There exist gaps in data which introduce discontinuity. A 2D scatter plot can display information maximum upto 4 parameters.



Load Matplotlib
set x to create list with 21 32 45
set y to create list with 34 25 50
Graph Title "X"
X-axis limit (0 , 100)
Y-axis limit (0 , 100)
Figure Size 10 , 10
Scatter
Enter X Array x
Enter Y Array y
Graph Label "Speed"
Colour "red"
Legend On
Show Graph

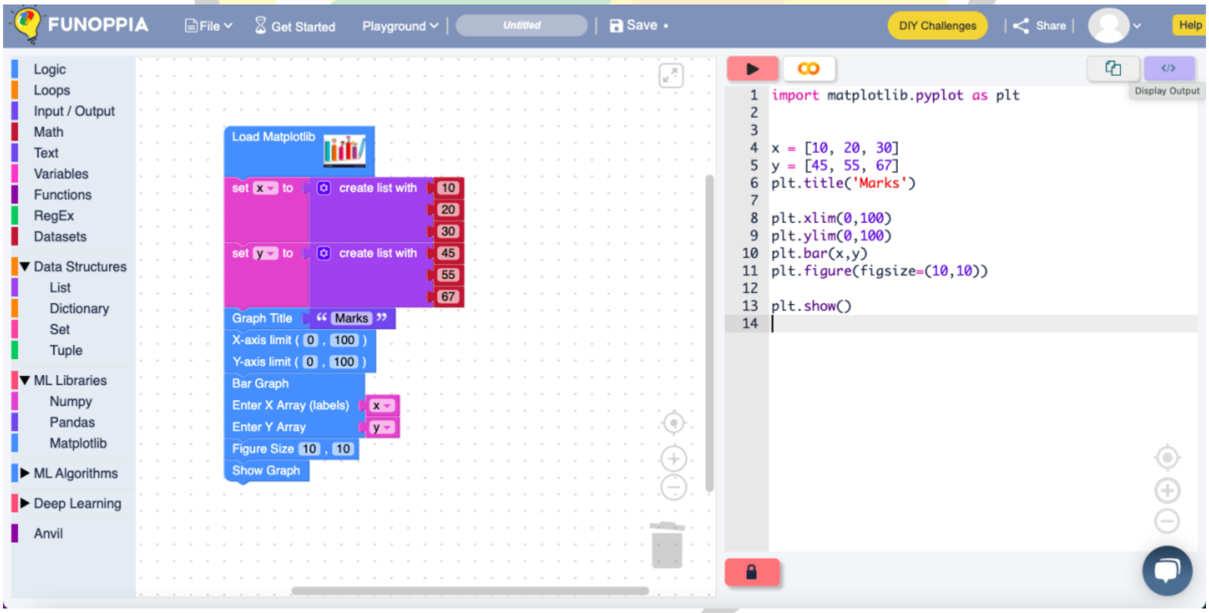
```
import matplotlib.pyplot as plt


x = [21, 32, 45]
y = [34, 25, 50]
plt.title('X')

plt.xlim(0,100)
plt.ylim(0,100)

plt.figure(figsize=(10,10))
plt.scatter(x,y,c='red',label='Speed')
plt.legend()

plt.show()

```



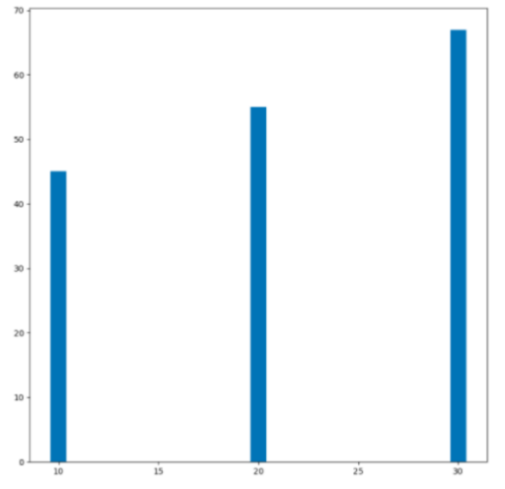
These random points that have been marked on a scatterplot for you to understand how a scatterplot looks. Hence you can try it yourself as well from the link and create a scatterplot on the same basis. Try making a scatterplot using Matplotlib on https://www.funoppia.com/funide/#

2. BAR CHART

It is one of the most commonly used graphical methods. From students to scientists, everyone uses bar charts in some way or the other. It is a very easy to draw yet informative graphical representation. Various versions of bar chart exist like single bar chart, double bar chart, etc



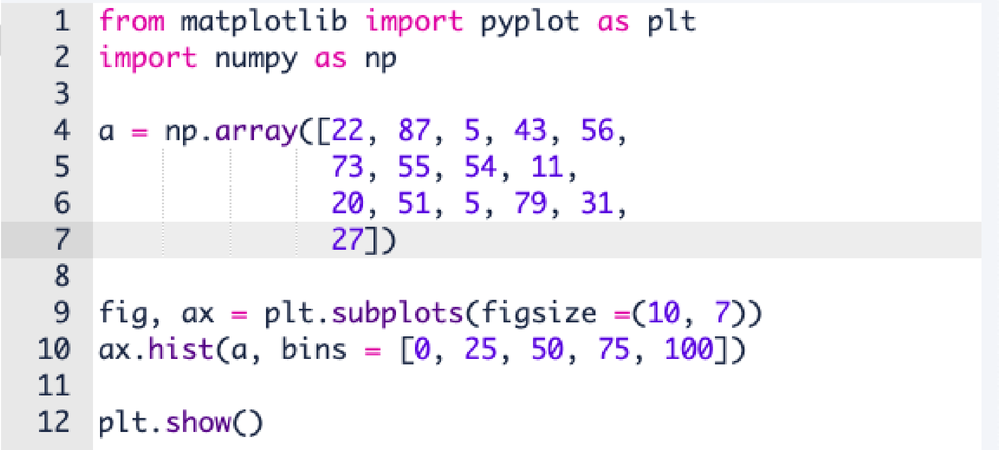
```python
import matplotlib.pyplot as plt


x = [10, 20, 30]
y = [45, 55, 67]
plt.title('Marks')

plt.xlim(0,100)
plt.ylim(0,100)
plt.bar(x,y)
plt.figure(figsize=(10,10))

plt.show()
```



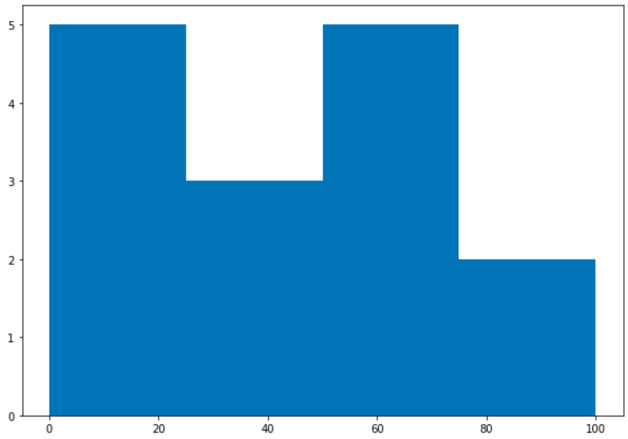
It is a bar plot which shoes us three different bars in different values with different values. Try making a bar graph using Matplotlib on https://www.funoppia.com/funide/#

3. HISTOGRAM

Histograms are the accurate representation of a continuous data. When it comes to plotting the variation in just one entity of a period of time, histograms come into the picture. It represents the frequency of the variable at different points of time with the help of the bins.



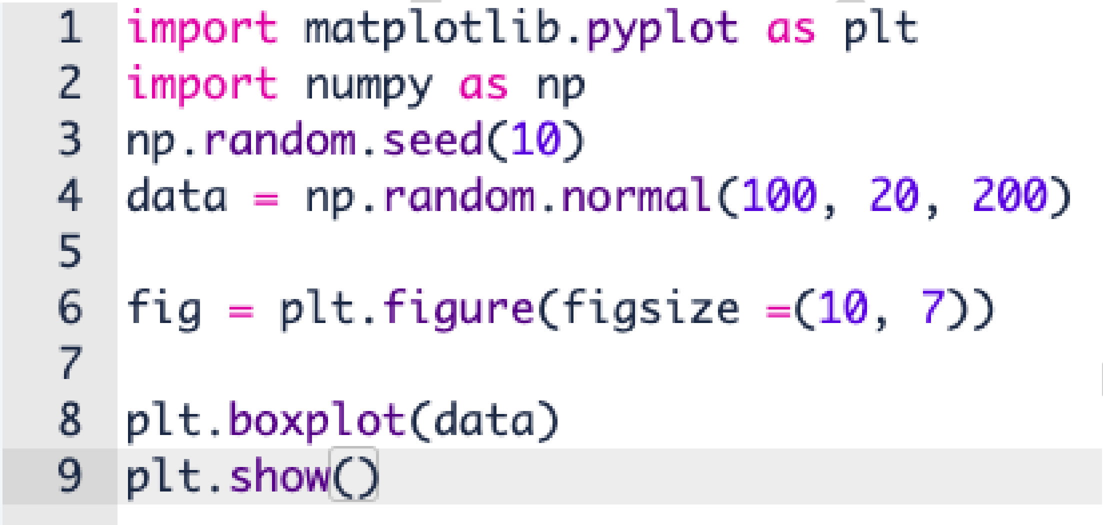
```python
from matplotlib import pyplot as plt
import numpy as np

a = np.array([22, 87, 5, 43, 56,
              73, 55, 54, 11,
              20, 51, 5, 79, 31,
              27])

fig, ax = plt.subplots(figsize =(10, 7))
ax.hist(a, bins = [0, 25, 50, 75, 100])

plt.show()
```



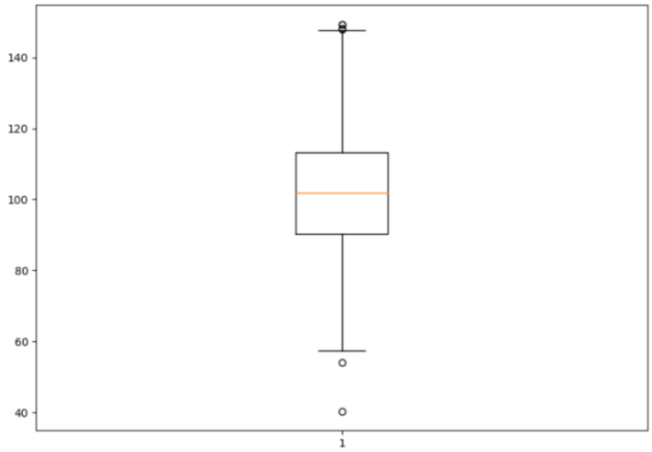
This is the output of a histogram with the array presented in the code above.

4. BOX PLOT

When the data is split according to its percentile throughout the range. Box plots also known as box and whiskers plot conveniently display the distribution of data throughout the range with the help of 4 quartiles



```python
import matplotlib.pyplot as plt
import numpy as np
np.random.seed(10)
data = np.random.normal(100, 20, 200)

fig = plt.figure(figsize =(10, 7))

plt.boxplot(data)
plt.show()
```



This is the output of a box plot with the random array presented in the code above.

• K-NEAREST NEIGHBOUR

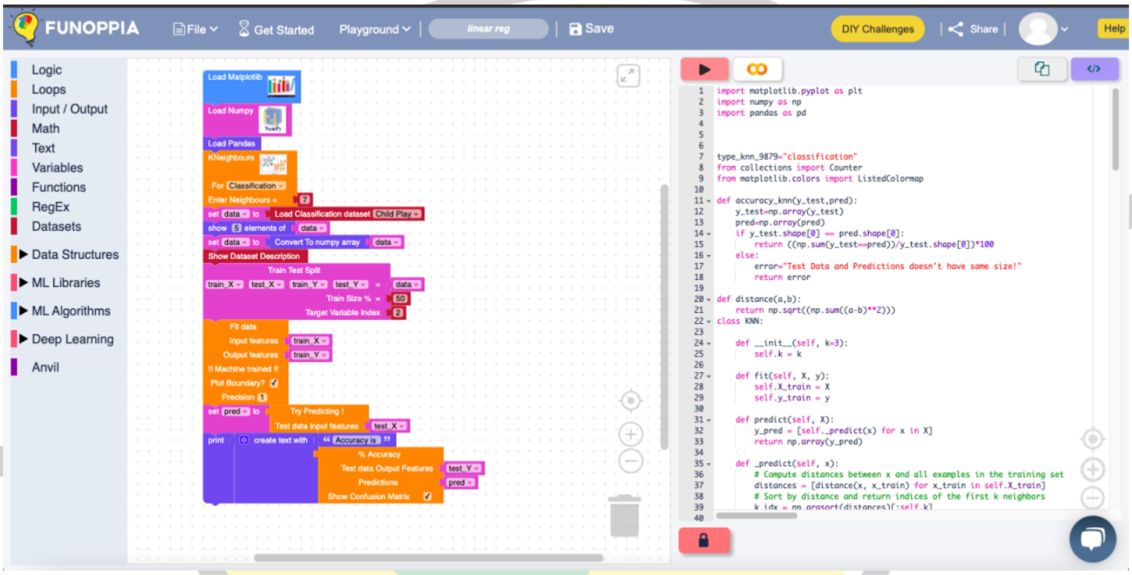
The k-nearest neighbours (KNN) algorithm is a simple, easy-to-implement supervised machine learning algorithm that can be used to solve both classification and regression problems. The KNN algorithm assumes that similar things exist in close proximity. In other words, similar things are near to each other as the saying goes “Birds of a feather flock together”. Some features of KNN are:

• The KNN prediction model relies on the surrounding points or neighbours to determine its class or group

- Utilises the properties of the majority of the nearest points to decide how to classify unknown points
- Based on the concept that similar data points should be close to each other

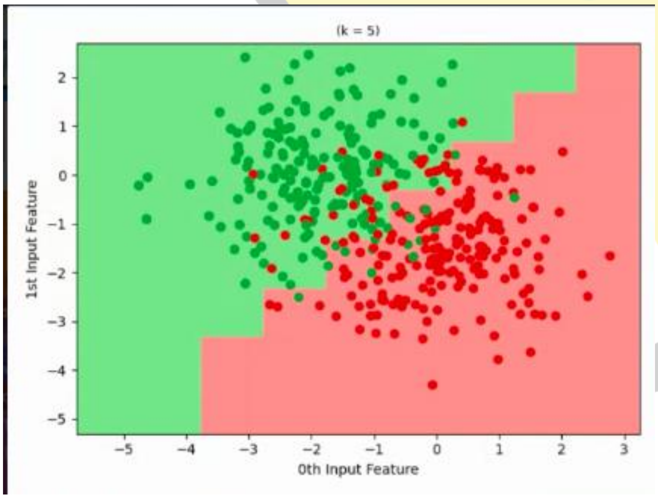
The personality prediction activity was a brief introduction to KNN

As you recall, in that activity, we tried to predict the animal for 4 students according to the animals which were the nearest to their points. This is how in a lay-man's language KNN works. Here, K is a variable which tells us about the number of neighbours which are taken into account during prediction. It can be any integer value starting from 1.



```
Load Matplotlib
Load Numpy
Load Pandas
KNeighbours
  For Classification
  Enter Neighbours = 7
set data to Load Classification dataset Child Play
show 5 elements of data
set data to Convert To numpy array data
Show Dataset Description
Train Test Split
train_X test_X train_Y test_Y = data
  Train Size % = 50
  Target Variable Index 2
Fit data
  Input features train_X
  Output features train_Y
  !! Machine trained !!
  Plot Boundary? ☑
  Precision 1
set pred to Try Predicting !
  Test data Input features test_X
print create text with " Accuracy is "
  % Accuracy
  Test data Output Features test_Y
  Predictions pred
  Show Confusion Matrix ☑
```

```python
import matplotlib.pyplot as plt
import numpy as np
import pandas as pd



type_knn_9879="classification"
from collections import Counter
from matplotlib.colors import ListedColormap

def accuracy_knn(y_test,pred):
    y_test=np.array(y_test)
    pred=np.array(pred)
    if y_test.shape[0] == pred.shape[0]:
        return ((np.sum(y_test==pred))/y_test.shape[0])*100
    else:
        error="Test Data and Predictions doesn't have same size!"
        return error

def distance(a,b):
    return np.sqrt((np.sum((a-b)**2)))
class KNN:

    def __init__(self, k=3):
        self.k = k

    def fit(self, X, y):
        self.X_train = X
        self.y_train = y

    def predict(self, X):
        y_pred = [self._predict(x) for x in X]
        return np.array(y_pred)

    def _predict(self, x):
        # Compute distances between x and all examples in the training set
        distances = [distance(x, x_train) for x_train in self.X_train]
        # Sort by distance and return indices of the first k neighbors
        k_idx = np.argsort(distances)[:self.k]
```



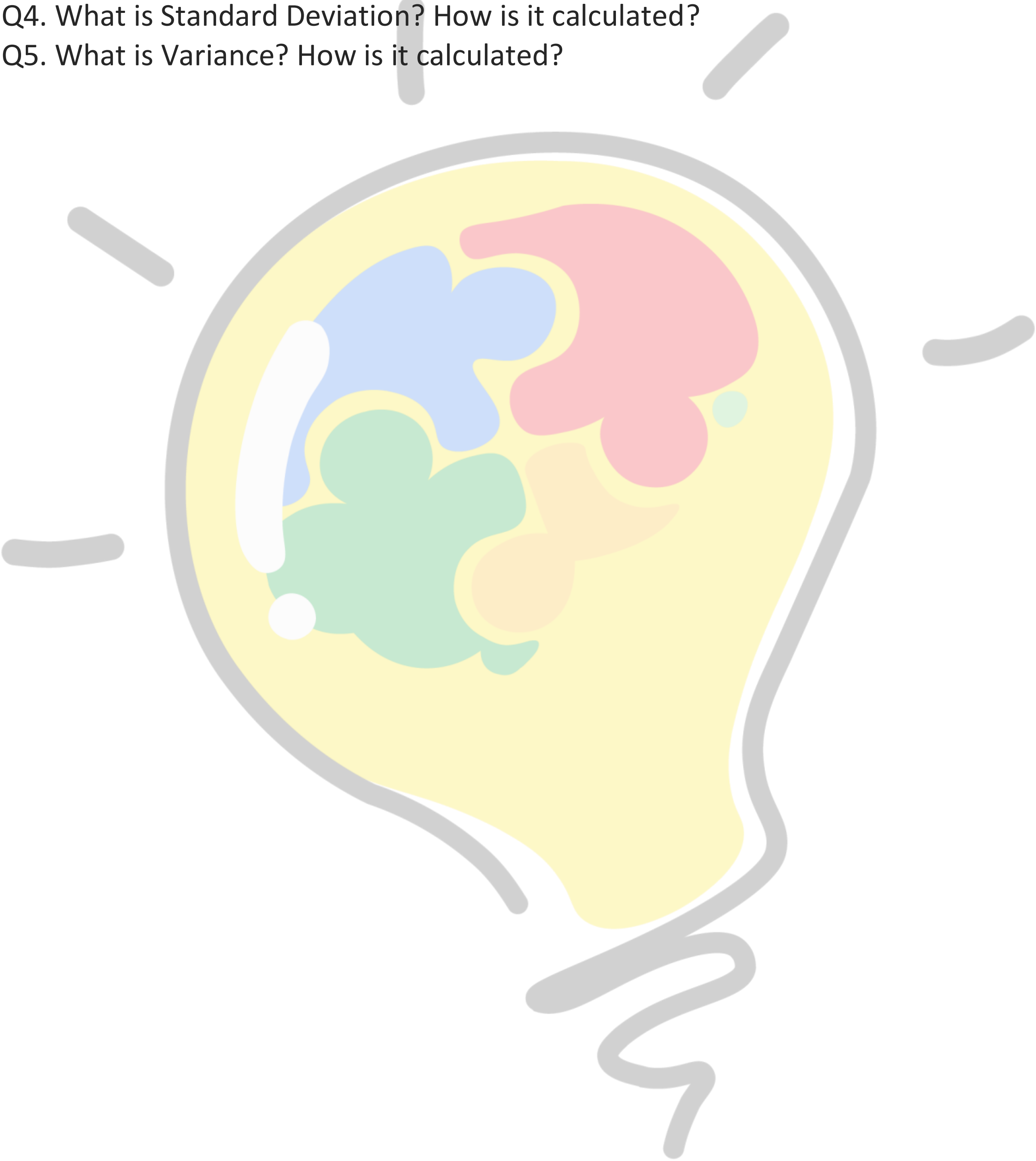
This is a key nearest neighbouring dataset which has the data whether a child can go to play due to a particular oxygen level and temperature outside. Try this K nearest Neighbours yourself and try changing the data.

Reference the code is given below-

https://www.funoppia.com/funide/index.php#c9f97xu

PRACTICE QUESTIONS

Q1. What is Mean? How is it calculated?
Q2. What is Median? How is it calculated?
Q3. What is Mode? How is it calculated?
Q4. What is Standard Deviation? How is it calculated?
Q5. What is Variance? How is it calculated?



PRACTICE HOTS QUESTIONS

Q1. Try making a scatterplot.
Q2. Try making a bar graph.