- **K-Nearest Neighbor(KNN) Algorithm**

Machine Learning

Dr. Jagendra Singh

K-NEAREST NEIGHBOR (KNN) CLASSIFICATION ALGORITHM

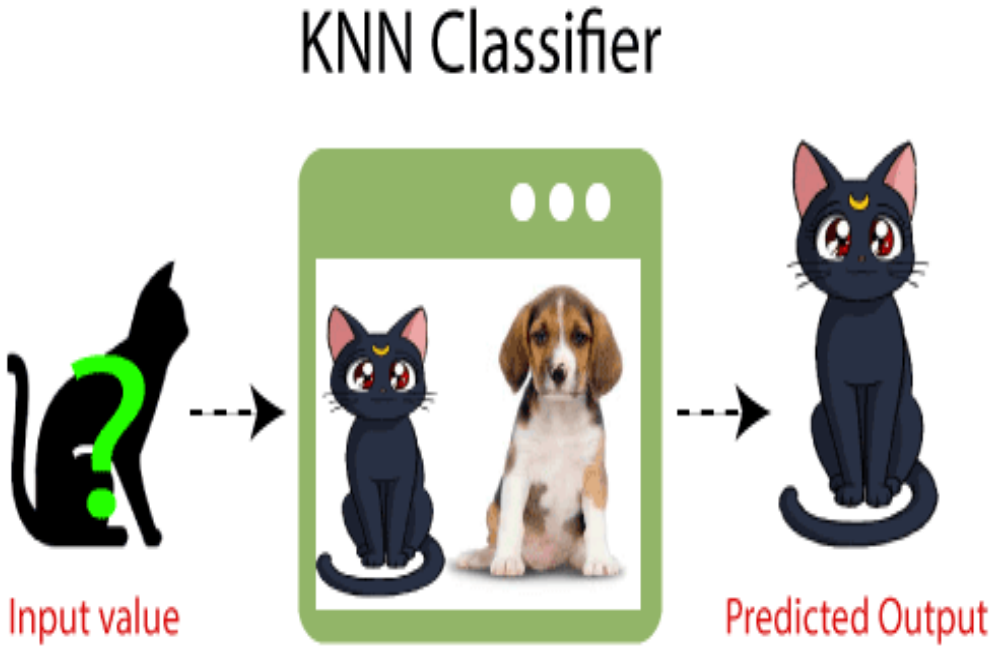
- K-Nearest Neighbour is one of the simplest Machine Learning algorithms based on Supervised Learning technique.
- K-NN algorithm assumes the similarity between the new case/data and available cases and put the new case into the category that is most similar to the available categories.
- K-NN algorithm stores all the available data and classifies a new data point based on the similarity.
- This means when new data appears then it can be easily classified into a well suite category by using K-NN algorithm.

K-NEAREST NEIGHBOR (KNN)

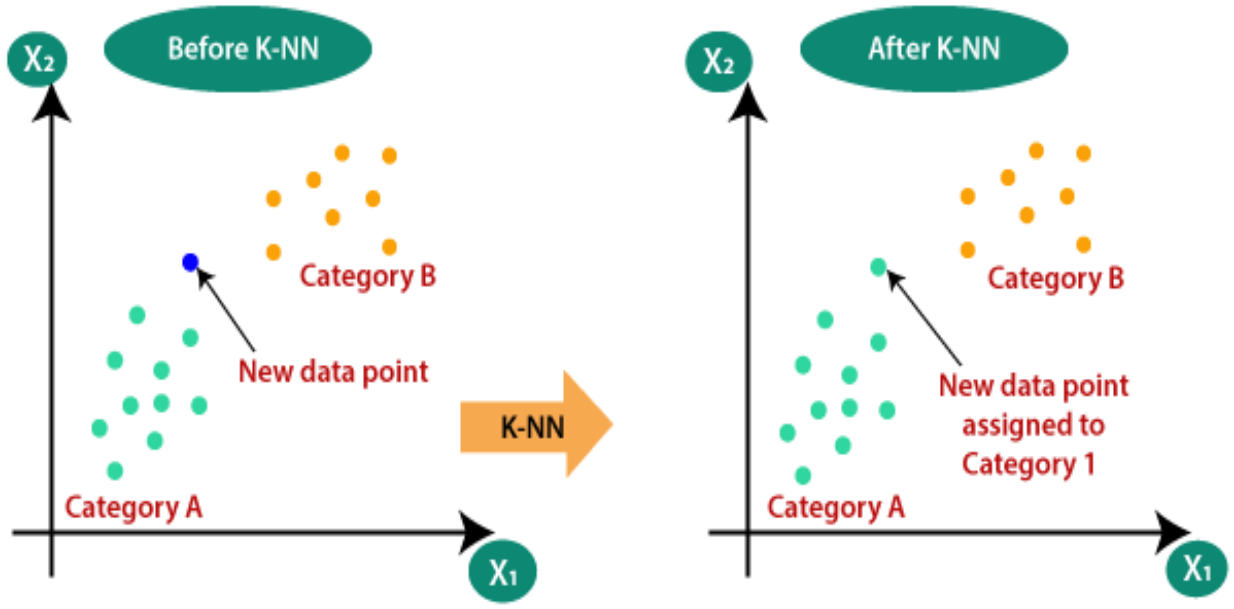
- K-NN algorithm can be used for Regression as well as for Classification but mostly it is used for the Classification problems.
- K-NN is a **non-parametric algorithm**, which means it does not make any assumption on underlying data.
- It is also called a **lazy learner algorithm** because it does not learn from the training set immediately instead it stores the dataset and at the time of classification, it performs an action on the dataset.

EXAMPLE:

- Suppose, we have an image of a creature that looks similar to cat and dog, but we want to know either it is a cat or dog.
- So for this identification, we can use the KNN algorithm, as it works on a similarity measure.
- Our KNN model will find the similar features of the new data set to the cats and dogs images and based on the most similar features it will put it in either cat or dog category.



WHY DO WE NEED A K-NN ALGORITHM?



- Suppose there are two categories, i.e., Category A and Category B, and we have a new data point x1, so this data point will lie in which of these categories.
- To solve this type of problem, we need a K-NN algorithm. With the help of K-NN, we can easily identify the category or class of a particular dataset.
- Consider this diagram:

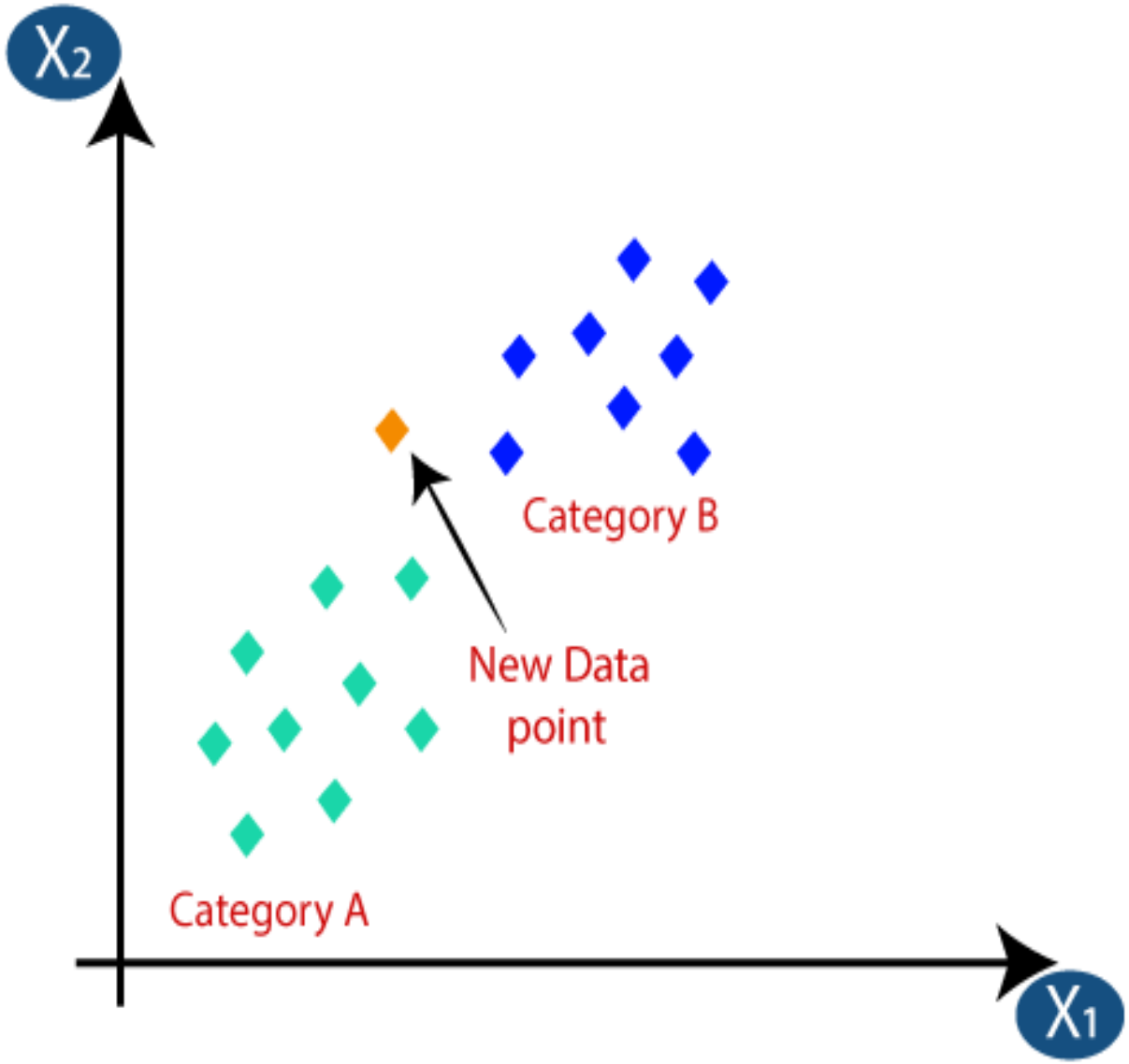
HOW DOES K-NN WORK?

The K-NN working can be explained on the basis of the below algorithm:

- **Step-1:** Select the number K of the neighbors
- **Step-2:** Calculate the Euclidean distance of **K number of neighbors**
- **Step-3:** Take the K nearest neighbors as per the calculated Euclidean distance.
- **Step-4:** Among these k neighbors, count the number of the data points in each category.
- **Step-5:** Assign the new data points to that category for which the number of the neighbor is maximum.
- **Step-6:** Our model is ready.

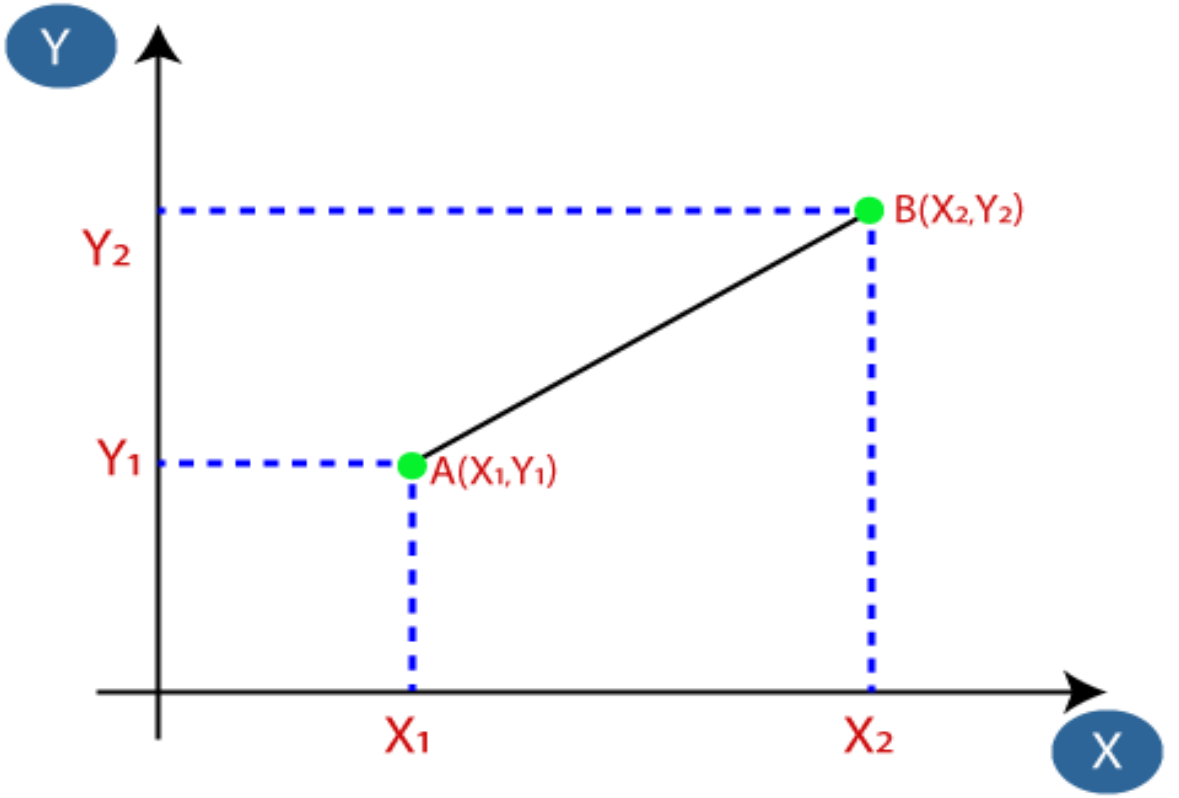
HOW DOES K-NN WORK?

- Suppose we have a new data point and we need to put it in the required category. Consider the below image:



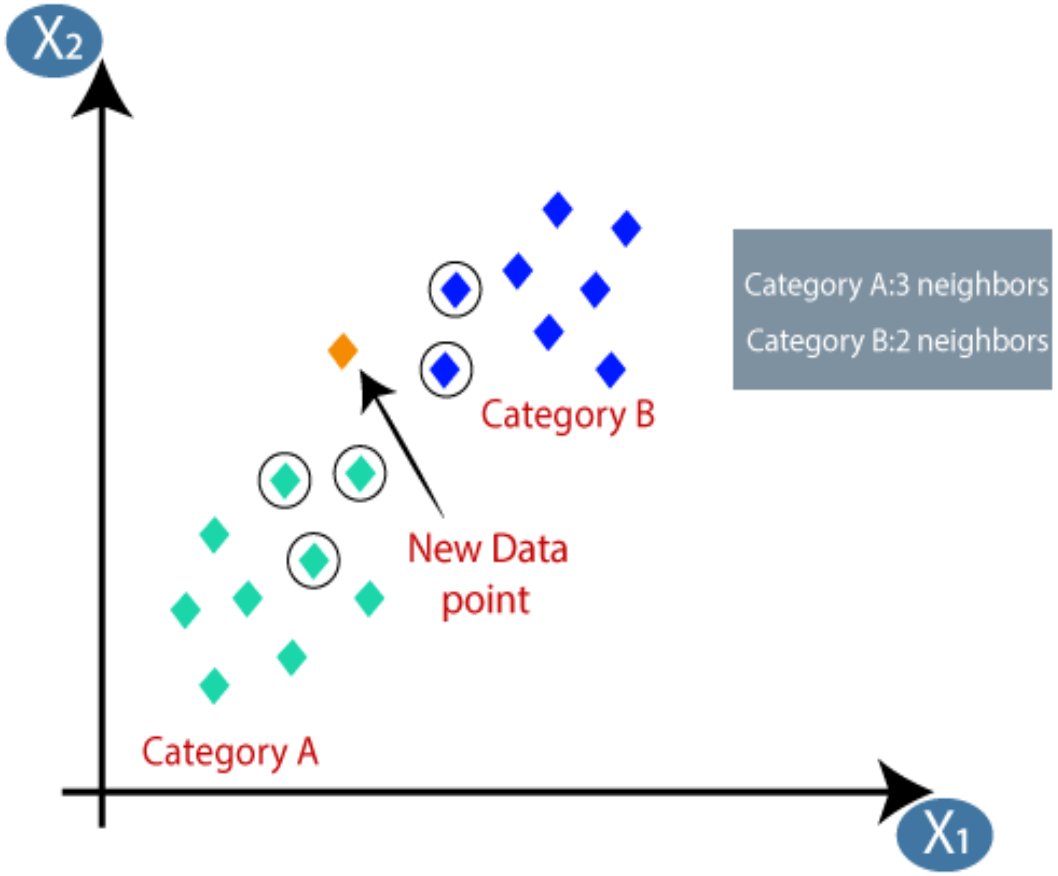
K-NN WORKING

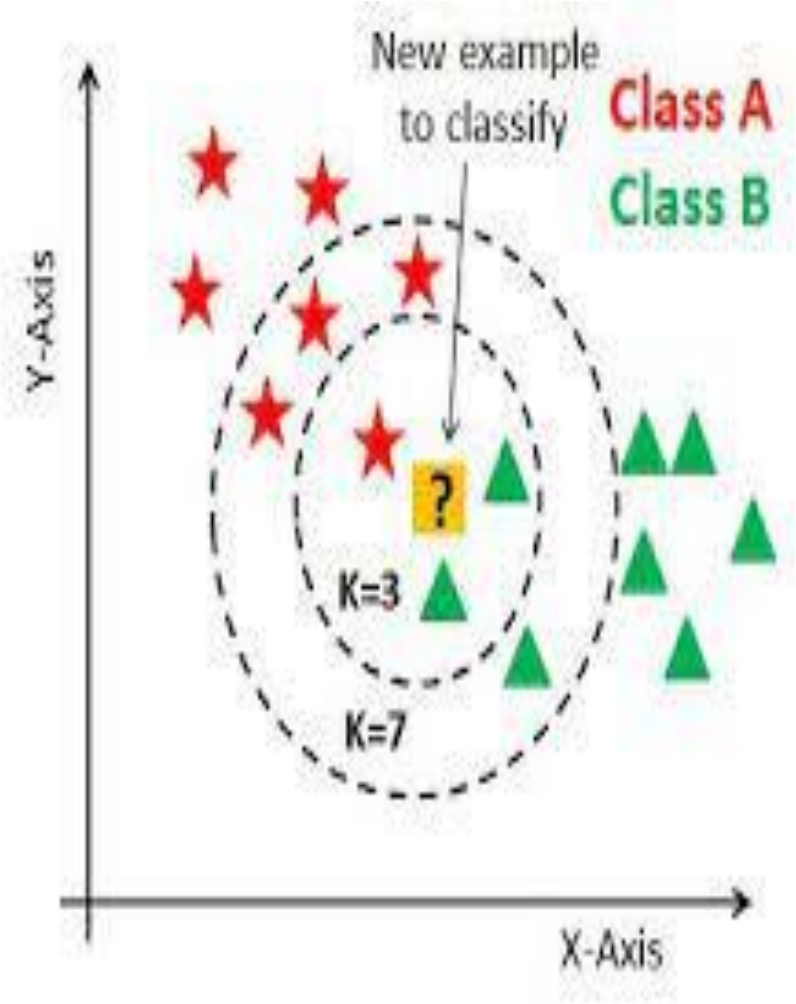
- Firstly, we will choose the number of neighbors, so we will choose the k=5.
- Next, we will calculate the **Euclidean distance** between the data points. The Euclidean distance is the distance between two points, which we have already studied in geometry. It can be calculated as:



KNN WORKING

- By calculating the Euclidean distance we got the nearest neighbors, as three nearest neighbors in category A and two nearest neighbors in category B.
- Consider this image:
- As we can see the 3 nearest neighbors are from category A, hence this new data point must belong to category A.





HOW TO SELECT THE VALUE OF K IN THE K-NN ALGORITHM?

- Below are some points to remember while selecting the value of K in the K-NN algorithm:
  - There is no particular way to determine the best value for "K", so we need to try some values to find the best out of them. The most preferred value for K is 5.
  - A very low value for K such as K=1 or K=2, can be noisy and lead to the effects of outliers in the model.
  - Large values for K are good, but it may find some difficulties.

ADVANTAGES OF KNN ALGORITHM:

It is simple to implement.

It is robust to the noisy training data

It can be more effective if the training data is large.

DISADVANTAGES OF KNN ALGORITHM:

Always needs to determine the value of K which may be complex some time.

The computation cost is high because of calculating the distance between the data points for all the training samples.

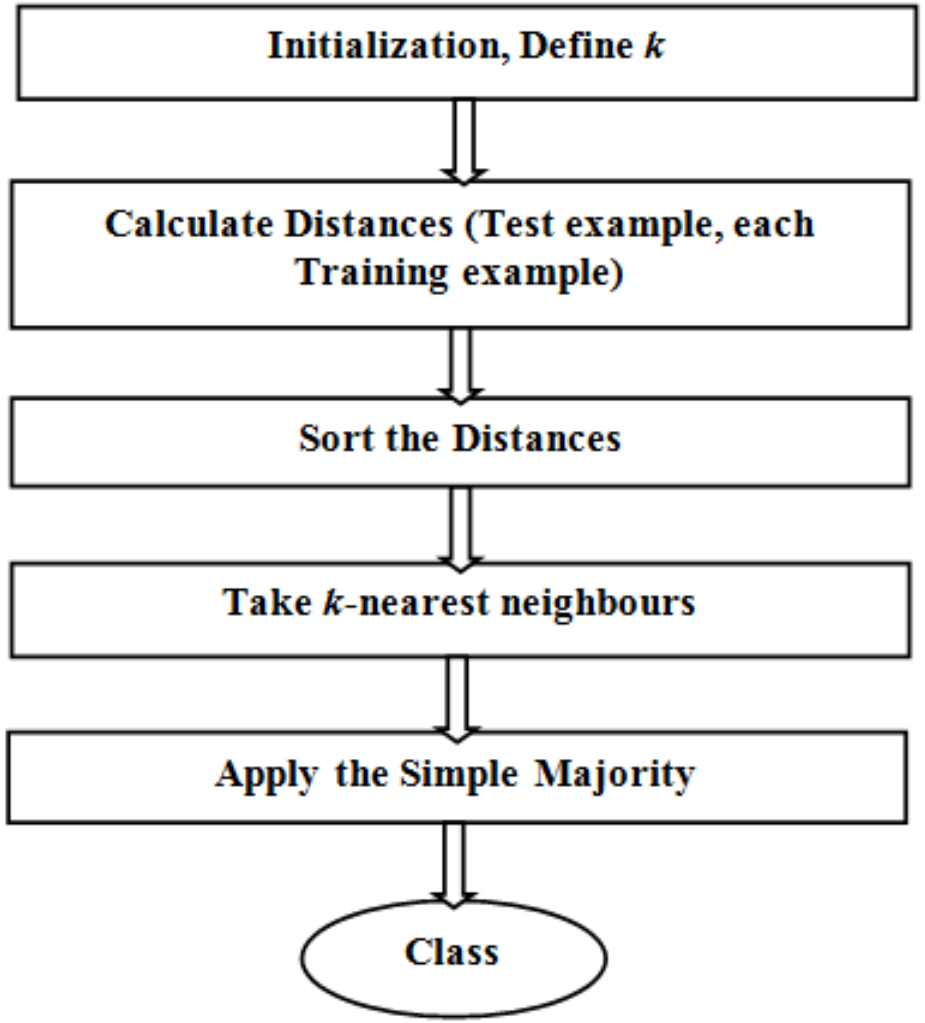
PYTHON IMPLEMENTATION OF THE KNN ALGORITHM

- To do the Python implementation of the K-NN algorithm, we will use the same problem and dataset which we have used in Logistic Regression.
- But here we will improve the performance of the model.
- Next is the problem description:

User ID	Gender	Age	EstimatedSalary	Purchased
15624510	Male	19	19000	0
15810944	Male	35	20000	0
15668575	Female	26	43000	0
15603246	Female	27	57000	0
15804002	Male	19	76000	0
15728773	Male	27	58000	0
15598044	Female	27	84000	0
15694829	Female	32	150000	1
15600575	Male	25	33000	0
15727311	Female	35	65000	0
15570769	Female	26	80000	0
15606274	Female	26	52000	0
15746139	Male	20	86000	0
15704987	Male	32	18000	0
15628972	Male	18	82000	0
15697686	Male	29	80000	0
15733883	Male	47	25000	1
15617482	Male	45	26000	1
15704583	Male	46	28000	1
15621083	Female	48	29000	1
15649487	Male	45	22000	1
15736760	Female	47	49000	1

PROBLEM FOR K-NN ALGORITHM:

- There is a Car manufacturer company that has manufactured a new SUV car. The company wants to give the ads to the users who are interested in buying that SUV.
- So for this problem, we have a dataset that contains multiple user's information through the social network.
- The dataset contains lots of information but the **Estimated Salary** and **Age** we will consider for the independent variable and the **Purchased variable** is for the dependent variable. Below is the dataset:



STEPS TO IMPLEMENT THE K-NN ALGORITHM:

- Data Pre-processing step
- Fitting the K-NN algorithm to the Training set
- Predicting the test result
- Test accuracy of the result(Creation of Confusion matrix)
- Visualizing the test set result.

THANK YOU