Unsupervised Learning.

- ## What is unsupervised learning?

  Unsupervised learning is a type of machine learning in which the algorithm is not provided with any pre-assigned labels or scores for the training data. As a result, unsupervised learning algorithms must first self-discover any naturally occurring patterns in that training data set. it solves the problem by learning the data and classifying it without any labels.

- ## What is Clustering.

  Clustering of datasets means grouping the data points into different clusters, consisting of similar data points. Clustering Technique is mostly done to unsupervised datasets. It helps to find usefulness and meaningfulness in data.

  There are 3 types of clustering technique

  1. Partitional clustering.
  2. Hierarchical clustering.
  3. Density-based clustering.

- ## What is pre-processing

  Data pre-processing can refer to manipulation or dropping of data before it is used in order to ensure or enhance performance, and is an important step in the data mining process.

- ## Importing libraires and cleaning data using python.

  Importing libraires:- numpy , pandas and Matplotlib.

Reading data:- using pandas we can read csv/excel/text with code pandas.read_csv(“filename.csv”).

Exploring data:- using pandas library we can explore data like

to explore the columns names we should code:- dataframe.target

to explore top 5 data of dataset we should code:- dataframe.head

exploring datatypes of data by using:- dataframe.dtypes

exploring shape of data by using:- dataframe.shape

checking if there is null values :- dataframe.isnull().sum()

from the above code we got all the information above the null values, which has to be removed

we can remove null values by dropping them or replacing them by mean or median or mode

by dropping columns we use:- dataframe.dropna(axis = 0)

it is very important to define axis or it will remove so much important data. 0 in axis stand for row and 1 stand for columns.

Replacing the values by calculating the mean or median for integer type data using :-dataframe.mean()/dataframe.meadian()

For categorical type data we calculate mode using:- dataframe.mode()

Filling these empty value by using:- dataframe.fillna(x,inplace=True)

After the understanding and removal of all the null values the data is read to cluster there are various method of clustering of data

- **Which algorithm to used, why to use :-**

  For unsupervised learning the best algorithm to use is K-means

  It is part of Partitional clustering. It is which groups the unlabelled dataset into different clusters. Here K defines the number of pre-defined clusters that need to be created in the process, as if K=2, there will be two clusters, and for K=3, there will be three clusters, and so on. It allows us to cluster the data into different groups and a convenient way to discover the categories of groups in the unlabelled dataset on its own without the need for any training.

  It is a centroid-based algorithm, where each cluster is associated with a centroid. A centroid is the imaginary or real location representing the center of the cluster. The main aim of this algorithm is to minimize the sum of distances between the data point and their corresponding clusters.

  The k-means clustering algorithm mainly performs two tasks:

- Determines the best value for K center points or centroids by an iterative process.
- Assigns each data point to its closest k-center. Those data points which are near to the particular k-center, create a cluster.

  Use of K-means Algorithm in given datasets:-

  After exploring the data, the housing loan and loan data can be clustered using K-means as rest of data age, job, married status, and and others too, cannot be easily iterated. housing and loan are in category which consist of 2 values yes and no we need to convert it into integer it can be done using.

  dataframe = dataframe.map ({‘yes’:1, ‘no’ : 0 })

from the above code the data is converted into integer form

now the k –means algorithm convert data into small numbers like if user is trying to cluster 2 rows k means will convert it into 1

it reduces the data and increases the meaning of data that is unsupervised K –means add value to the data.

The algorithm of K – means is :-

Step 1:- Select the number K to decide the number of clusters.

Step 2:- Select the number K to decide the number of clusters.

Step 3:- Assign each data point to their closest centroid, which will form the predefined K clusters.

Step 4:- Calculate the variance and place a new centroid of each cluster.

Step 5:- Repeat the third step, which means reassign each datapoint to the new closest centroid of each cluster.

Step 6:- If any reassignment occurs, then go to step-4 else go to FINISH.

Step-7: The model is ready.

Since Housing and loan is of same type then there is no need of 2 rows so it can be cluster using K –means and can be formed in 1 row which can make data easy to interpret to user

Importing housing and loan to a new variable

x = dataset.iloc[:, [6, 7]].values

K-means is a part of python sklearn library

Importing K-means from python

from sklearn.cluster import KMeans

- **Determining of clusters.**

There are two methods to find the number of clusters

1. The elbow method.
2. The silhouette coefficient.

We will use The elbow method to determine number of cluster.

Elbow method run several k-means, increment k with each iteration, and record the SSE. When we plot SSE as a function of the number of clusters, notice that SSE continues to decrease as you increase k. As more centroids are added, the distance from each point to its closest centroid will decrease.

There is a spot where the SSE curve starts to bend known as the elbow point. The x-value of this point is thought to be a number of clusters.

Elbow method is most preferred because it iterates through each number of cluster for example 1 to 10 it will calculate each SSE value and store it in list here SSE stands for sum of square error

Then after the call the graph is plot we can easily determine the elbow point and exact number of cluster from the graph

Implementation of Elbow point Using Python :-

Create an empty list to store values :- n_list = []

Calculating graph points value using for loop :-

```
for i in range(1, 11):
kmeans = KMeans(n_clusters=i, init='k-means++', random_state= 42)
kmeans.fit(x)
n_list.append(kmeans.inertia_)
```

here the k_means parameter used are

n_clusters = sets k for the clustering step.

init = sets the number of initializations to perform.

random state = ensures that the splits that you generate are reproducible.

We can draw graph using matplotlib function to determine the accurate cluster number and can calculate the k-means of the dataset we want by using that cluster using the formula of K-means used in elbow point.

Thank You.

A case study by:-

Yash Anil Joshi.