Assignment: 2

- **Aim :** assignment on classification technique.

Every year many student give the GRE exam to get admissionforeign universities the dataset content GRE score out off (340), TOEFL scores out off (120) ,university rating out off(5), statement of purpose strength out off(5),letter of recommendation straight out off (5) under graduate GPA out off(10), research experience (0=no,1=yes).admitted is the target variable. the counsellor of firm is supposed to check whether the student will get admission or not based on his/her GRE score and academic score.so to take a counsellor to take appropriate decision build the machine learning model classifier using the decision tree to predict whether a student will get admission or not. apply a data pre-processing(label encoding ,data transformation…) technique is necessary. Perform data-preparation (train-test split).

c. Apply machine learning algorithm.

d. Evaluate model.

Software used: pyCharm community edition 2022.2.1.

Dataset: Admission predict (Admission_predict .csv)

Link: https://www.Kaggle.com/mohansacharya/graduate-admission

- **Theory :**
  - Classification:

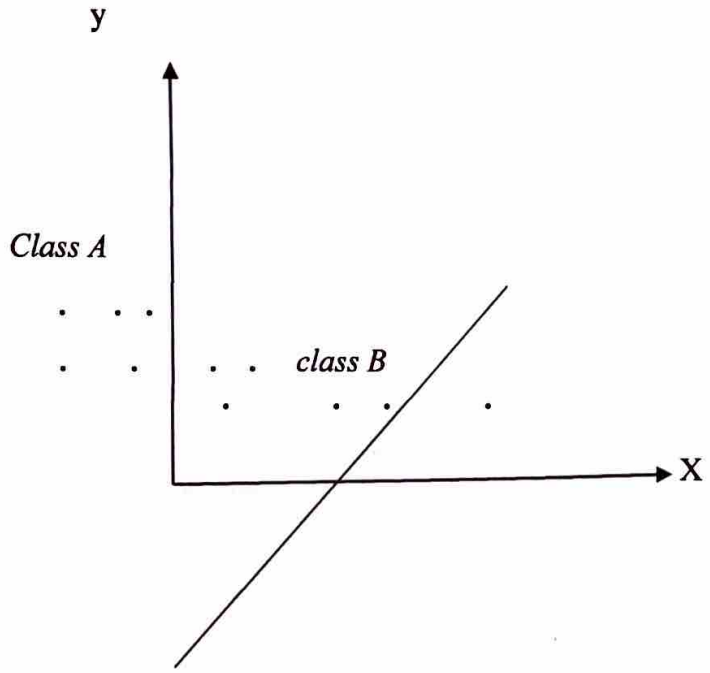
1. The classification is a supervised learning techniques
2. It is used to identify the category of new observation on the basis of training data.
3. In a classification a program a programs learns from the given dataset or observation and then classifies new observation into number of classes of groups. Such as yes or no,0 or 1,spam or not etc.classes can be called as targets/labvels or categories.

Unlike regression the output variable of classification is a category not a value , such as "green or blue", "fruit or animal" etc. since the classification algorithm is a supervised learning technique, hence it takes labelled input data, which means it contains input with the corresponding output

- In classification a discrete output function is(y)is mapped to input variable (x).

Y=f(x) where y= categorical output.

- The main goal of the classification algorithm is to identify category of a given dataset and these algorithm are used to predict the output for the categorical data.
- Classification can be classified into two classes class A and class B.
- These classes have features that they are similar to each other and dissimilar to other classes

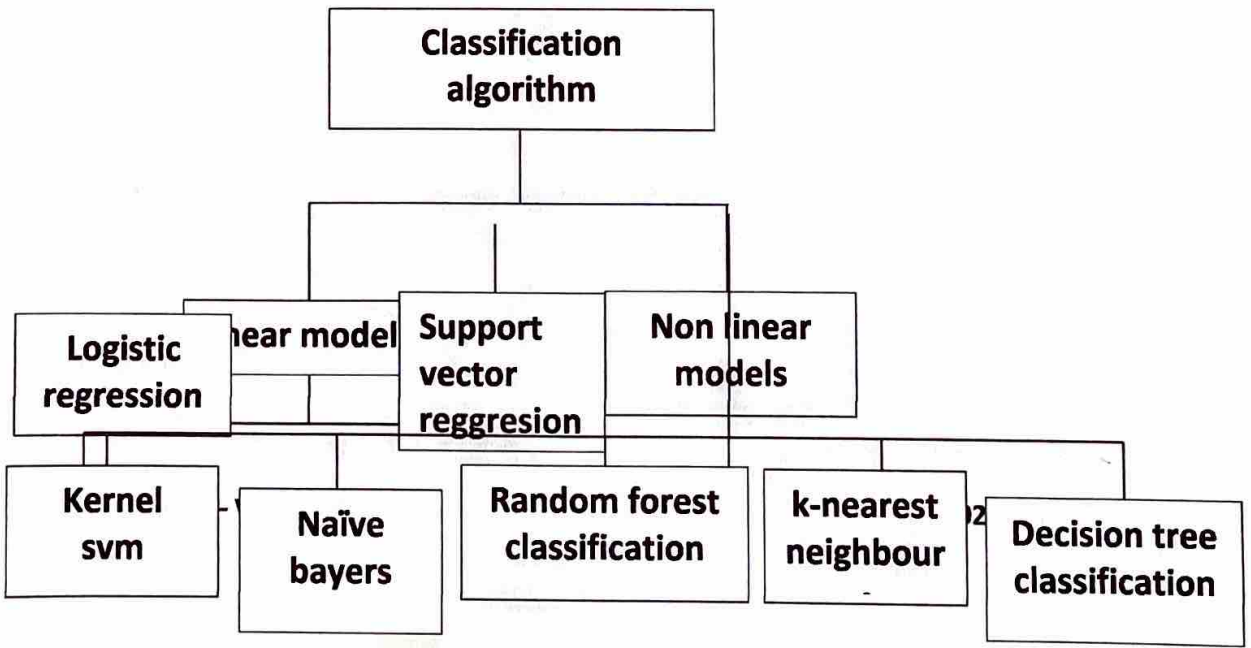


- The algorithm with implements the classification on a dataset is known as classifier.
- There are two types of classification :
    1. Binary classifier : if the classification problem has only two possible outcomes, then it is also called as binary classifier.

       Examples: a. yes or no,
       b. male or female
       c. spam or not spam.
    2. Multiclass classifier :if classification problem is more than two outcomes then it is called as binary classifier.

       Examples: a. classification of types of crops.
       b. classification of types of music.

- **Types of classification algorithms.**



1. Linear classification model :

   A classification algorithm (classifier) that makes its classification based on a linear predictor function combining a set of weight with the feature vector.

   Formula :

$$y = \beta 0 + \beta 1x1 + \beta 2x2.....\beta nxn + €$$

   Where,

   y = (dependent) outcome variables.

   X1 = the values of the independent variables.

   β0 = the value of y where each x is equal to 0.it is also called as y-intercept.

   β1 = the change in y based on unit change in x1 it is also called as regression coefficient or slope of the regression line.

   € = random error that represents the difference between The predicted value and actual value.

2. Simple linear regression =

$$Y=\beta 0 + \beta 1x1$$

3. To calculate β0 and β1 =

$$\beta 0 = \bar{y} - \beta 1\bar{x}$$

$$\beta 1 = \frac{\sum(x1-\bar{x})(y1-\bar{y})}{\sum(x1-\bar{x})^2}$$

Advantages :

- Logistic regression : probabilistic approach gives information on about statistical significance of features.
- K-nearest neighbour : simple to understand, past and efficient.
- Support vector machine(svm) : performance not based on by outliers, not sensitive to overfitting.
- Kernel svm : high performance or non-linear problems, not sensitive to overfitting.
- Naïve bayers : efficient not based by outliers, walks on non-linear problems, probabilistic approach.
- Decision tree classification : interpretability, no need for features scaling, works on both linear/non linear problems.
- Random forest classification : powerful and accurate, good performance on many problems, including non-linear.

Disadvantages :

- Logistic regression : the assumption of logistic regression
- K-nearest neighbours : need to manually choose the number of neighbour 'k'.
- Support vector machine(svm) : not appropriate for non linear problem, not the best choice for large no. of features.
- Kernel svm : not the best choice for large no. of features, more complex.
- Naïve bayers : based on the assumption that the features have some statistical relevance.
- Decision tree classification : poor result on very small datasets, overfitting can easily occur.
- Random forest classification : no interpretability, overfitting can easily occur. need to choose the no. of trees manually.

Application :

- Email spam detector.
- Documentation.
- Sentiment analysis.
- Image classification.

Examples :

- Optical character recognition.
- Face recognition.
- Speech recognition.
- Medical diagnosis.
- Knowledge extraction.
- Compression.

Libraries used for this classification practical are :

- Pandas as pd.
- NumPy as np.
- Matplotlib .pyplot as plt.
- Seaborn as seaborn Instance.
- Sklearn.

Conclusion :

- It conclude that, classification algorithms used in machine learning utilize input training data for the purpose of predicting the like hood or probability that the data that follows will fall into one of the predetermined categories.

Assignment: 3

Aim: This dataset gives the data of Income and money spent by the customers visiting a Shopping Mall. The data set contains Customer ID, Gender, Age, Annual Income, Spending Score. Therefore, as a mall owner you need to find the group of people who are the profitable customers for the mall owner. Apply at least two clustering algorithms (based on Spending Score) to find the group of customers.

A. Apply Data pre-processing (Label Encoding , Data Transformation….) techniques if necessary.

B. Perform data-preparation(Train-Test Split)

C. Apply Machine Learning Algorithm

D. Evaluate Model.

E. Apply Cross-Validation and Evaluate Mode

Theory:

Clustering in Machine Learning

Unsupervised learning uses machine learning algorithms to analyze and cluster unlabeled datasets. These algorithms discover hidden patterns or data groupings without the need for human intervention. Clustering is the task of dividing the population or data points into a number of groups such that data points in the same groups are more similar to other data points in the same group than those in other groups. In simple words, the aim is to segregate groups with similar traits and assign them into clusters. Clustering has a large no. of applications spread across various domains. Some of the most popular applications of clustering are:

- Recommendation engine
- Market segmentation
- Social network analysis
- Search result grouping
- Medical imaging
- Image segmentation
- Anomaly detection

Types of Clustering Algorithm:

Every methodology follows a different set of rules for defining the 'similarity' among data points.

- **Connectivity models:** As the name suggests, these models are based on the notion that the data points closer in data space exhibit more similarity to each other than the data points lying farther away. These models can follow two approaches. In the first approach, they start with classifying all data points into separate clusters and then aggregating them as the distance decreases. In the second approach, all data points are classified as a single cluster and then partitioned as the distance increases. Also, the choice of distance function is subjective. These models are very easy to interpret but lacks scalability for handling big datasets. Examples of these models are Hierarchical clustering algorithm and its variants.

- **Centroid models**: These are iterative clustering algorithms in which the notion of similarity is derived by the closeness of a data point to the centroid of the clusters. K-Means clustering algorithm is a popular algorithm that falls into this category. In these models, the no. of clusters required at the end have to be mentioned beforehand, which makes it important to have prior knowledge of the dataset. These models run iteratively to find the local optima.

K-Means Clustering Algorithm:

Step-1: Select the value of K, to decide the number of clusters to be formed.
Step-2: Select random K points which will act as centroids.
Step-3: Assign each data point, based on their distance from the randomly selected points (Centroid), to the nearest/closest centroid which will form the predefined clusters.
Step-4: place a new centroid of each cluster.
Step-5: Repeat step no.3, which 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 Step 7.
Step-7: FINISH

Python Functions:

1. KMeans(n clusters=8, *, init='kmeans++', n init=10, max iter=300, tol=0.0001, verbose=0, random state=None, copy x=True, algorithm='auto')
'k-means++' : selects initial cluster centers for k-mean clustering in a smart way to speed up convergence.
n init: default=10
Number of time the k-means algorithm will be run with different centroid seeds. The final results will be the best output of n init consecutive runs in terms of inertia.
max iter: int, default=300
Maximum number of iterations of the k-means algorithm for a single run random state: default=None
Determines random number generation for centroid initialization.
2. fit(X[, y, sample weight]): Compute k-means clustering.
3. fit predict(X[, y, sample weight]): Compute cluster centers and predict cluster index for each sample.
4. Find Optimal no. of clusters using Dendrogram: import scipy.cluster.hierarchy as shc
dendro = shc.dendrogram(shc.linkage(X, method="ward"))
plt.title("Dendrogram Plot")
plt.ylabel("Euclidean Distances")
plt.xlabel("Customers")
plt.show()
5. Train Model: from sklearn.cluster import AgglomerativeClustering
hc= AgglomerativeClustering(n clusters=5, affinity='euclidean', linkage='ward')
y pred= hc.fit predict(X)
print(y pred)
6. Visulaize the clusters: plt.scatter(X[y pred==0, 0], X[y pred==0, 1],s=50, c='red', label ='Cluster 1')
plt.scatter(X[y pred==1, 0], X[y pred==1, 1], s=50,c='blue', label ='Cluster 2')
plt.scatter(X[y pred==2, 0], X[y pred==2, 1], s=50, c='green', label ='Cluster 3')
plt.scatter(X[y pred==3, 0], X[y pred==3, 1], s=50, c='cyan', label ='Cluster 4')
plt.scatter(X[y pred==4, 0], X[y pred==4, 1], s=50, c='magenta', label ='Cluster 5')

Conclusion:

After completion of this assignment, student can able to apply k-means and hierarchical clustering techniques on any dataset.

Assignment: 4

Aim:

This dataset comprises the list of transactions of a retail company over the period of one week. It contains a total of 7501 transaction records where each record consists of the list of items sold in one transaction. Using this record of transactions and items in each transaction, find the association rules between items.

There is no header in the dataset and the first row contains the first transaction, so mentioned header = None here while loading dataset.

A. Follow following steps :

B. Data Preprocessing

C. Generate the list of transactions from the dataset

D. Train Apriori algorithm on the dataset

E. Visualize the list of rules

Theory:

Association Rule Learning

An association rule is a rule-based method for finding relationships between variables in a given dataset.These methods are frequently used for market basket analysis, allowing companies to better understand relationships between different products. Understanding consumption habits of customers enables businesses to develop better cross-selling strategies and recommendation engines.

Examples of this can be seen in Amazon's "Customers Who Bought This Item Also Bought" or Spotify's" Discover Weekly" playlist. While there are a few different algorithms used to generate association rules, such as Apriori, Eclat, and FP-Growth, the Apriori algorithm is most widely used.

Apriori Algorithm:

Step-1: Determine the support of itemsets in the transactional database, and select the minimum support and confidence.

Step-2: Take all supports in the transaction with higher support value than the minimum or selected
support value.

Step-3: Find all the rules of these subsets that have higher confidence value than the threshold or minimum confidence.

Step-4: Sort the rules as the decreasing order of lift.

In order to select interesting rules from the set of all possible rules, constraints on various measures of
significance and interest are used. The best-known constraints are minimum thresholds on support and
confidence.

• Support: Support is an indication of how frequently the itemset appears in the dataset.
supp(X)=Freq(X)/ T

• Confidence: Confidence indicates how often the rule has been found to be true. Or how often the
items X and Y occur together in the dataset when the occurrence of X is already given.
confidence=freq(X,Y)/Freq(X)

• Lift: It is strength of any rule.

Lift=supp(X,Y)/supp(X)*supp(Y)

Python Implementation:

1. Library Required:apyori (https://pypi.org/project/apyori/)

Apyori is a simple implementation of Apriori algorithm with Python 2.7 and 3.3 - 3.5, provided as APIs and as command line interfaces. Installation: Choose one from the following.

Put apyori.py into your project.

Run python setup.py install.

pip install apyori

2. #Getting the list of transactions from the dataset

transactions = []

for i in range(0, 7501):

transactions.append([str(dataset.values[i,j]) for j in range(0, 20)])

3. Training Apriori algorithm on the dataset

from apyori import apriori

rule list= apriori(transactions, min support = 0.003, min confidence = 0.3, min lift = 3, min length = 2)

4. Display Rules:

for item in results:

\# first index of the inner list

\# Contains base item and add item

pair = item[0]

items = [x for x in pair]

print("Rule: "+ items[0] + " -> " + items[1])

\#second index of the inner list

print("Support: " + str(item[1]))

\#third index of the list located at 0th

Lab Manual-Laborartory Practice-I (ML) 21 VPKBIET, Baramati

5.4. POST LAB Association Rule Learning

\#of the third index of the inner list

print("Confidence: " + str(item[2][0][2]))

print("Lift: " + str(item[2][0][3]))

print("==========")

Conclusion:

After completion of this assignment, student can able to apply apriori algorithm and construct rules for any dataset.