GRP-B Experiment-1

Title:- Predict the price of Uber ride from given pickup point to agreed drop-off location. Perform following tasks:-

1. Preprocess the dataset.
2. Identify outliers.
3. Check the correlation.
4. Implement linear regression and random forest regression models.
5. Evaluate the models and compare their respective scores like R2, RMSE, etc.

Dataset Description:- The project is about on world's largest taxi company Uber Inc. In this project, we're looking to predict the fare for their future transactional cases. Uber ~~drivers~~ dilivers service to lakhs of customers daily. Now it becomes really imp to manage their data properly to come up with new business ideas to get results. Eventualy, it becomes really imp to estimate the fare prices accurately.

objective:- Students should be able to preprocess dataset and identify outliers, to check correlation and implement linear regression and random forest regression models. Evaluate them with respective scores like R2, RMSE ~~etc~~ etc.

Theory:-

Data preprocessing:- It is the process of preparing the raw data and making it suitable for machine learning model. It is first and crucial step while creating a ML model. When creating a ML project, it is not always a case that we come across the clean and formatted data. And while doing any operation with data, it is mandatory to clean it and put in formatted way.

Need of Data preprocessing:-

A real-world data generally contains noises, missing values, and maybe in an unusable format which cannot be directly used for ML models. Data preprocessing is required tasks for cleaning the data and making it suitable for a ML model which also increases the accuracy and efficiency of ML model.

It involves below steps:-

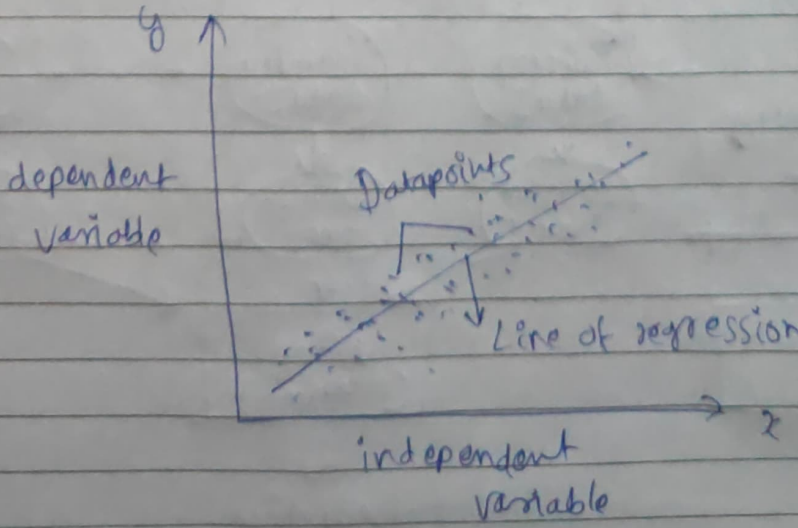
- Getting dataset.
- Importing libraries
- Importing datasets.
- Finding Missing Data.
- Encoding Categorical Data.
- Splitting dataset into training and test set.
- Feature scaling.

Linear regression :-

Linear regression is one of easiest and popular ML algo$^{m}$. It is a statistical method that is used for predictive analysis. It makes predictions for continuous or numeric variables such as sales, salary, age, product, price, etc.

Algo$^{m}$ shows a linear relationship bet$^{n}$ a dependent (y) and one or more independent variables, hence called linear regression. Since linear regression shows the linear relationship, which means it indicates how the value of the dependent variable is changing according to the value of the independent variable.

The linear regression model provides a sloped straight line representing the relationship bet$^{n}$ the variables.

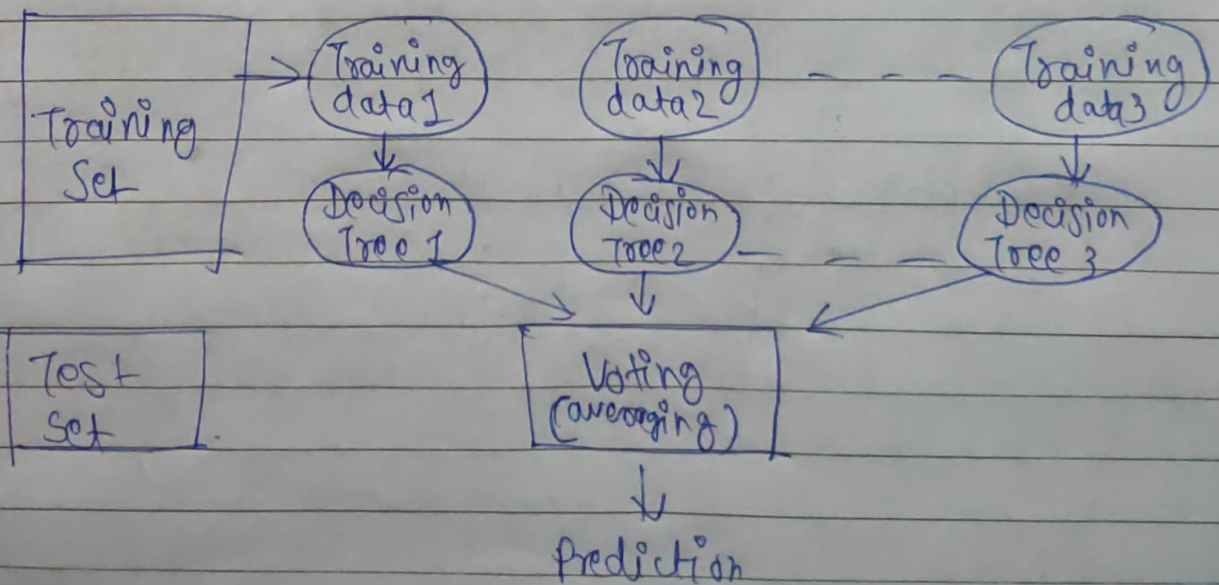


Random forest regression models:-

It is popular ML algo$^m$ that belongs to Supervised learning technique. It can be used for both classification and Regression problems in ML. It is based on the concept of ensemble learning, which is a process of combining multiple classifiers to solve a complex problem & and to improve the performance of model.

"Random forest is a classifier that contains a no. of decision trees on various subsets of given dataset and takes the average to improve the predictive accuracy of that dataset." Instead of relying on one decision tree, the random forest takes the prediction from each tree and based on the majority votes of predictions, and it predicts output.

The greater no. of trees in the forest leads to higher accuracy and prevents the problem of overfitting.



Boxplot :-

Boxplots are a measure of how well data is distributed across a dataset. This divides the data set into three quartiles. This graph represents the minm maxm, average, 1st quartile, and the third quartile in dataset. Boxplot is also useful in comparing the distribution of data in dataset by drawing a boxplot for each of them.

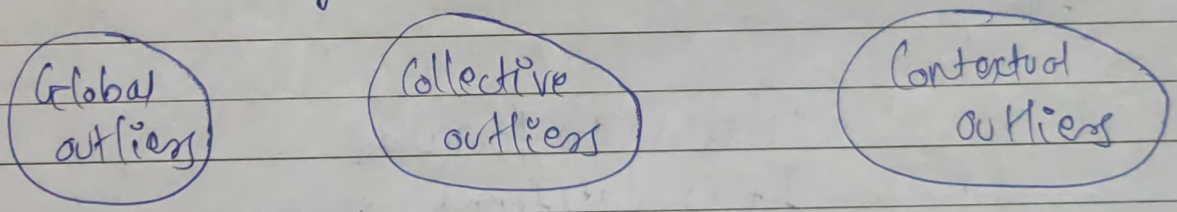
R provides a boxplot() fun to create boxplot. There is following syntax of boxplot() fun: boxplot(x, data, notch, varwidth, names, main).

Here,

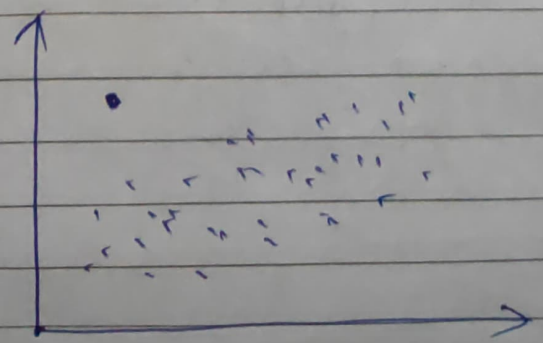
| | |
|---|---|
| 1. x | It is vector formula. |
| 2. data | It is dataframe. |
| 3. notch | It is a logical value set as true to draw notch |
| 4. varwidth | It is also a logical value set as true to draw the width of the box same as sample size. |
| 5. names | It is the group of labels that will be printed under each boxplot. |
| 6. main | It is used to give a title to graph. |

Outliers:- As name suggests, "outliers" refers to the data points that exist outside of what is to be expected. The major thing about the outliers is what you do with them. If you are going to analyze any task to analyze data sets, you will always have some assumptions based on how this data is generated. If you and some data points are likely to contain some form of error, then these are definitely outliers, and depending on the context, you want to overcome those errors.

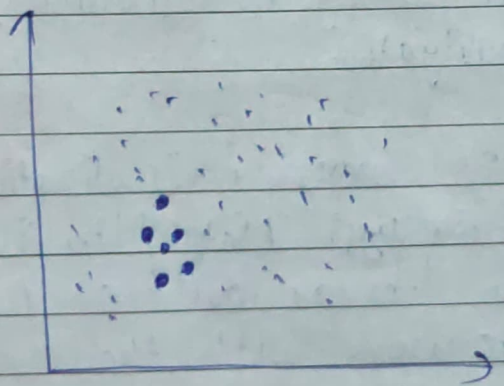
Types of outliers.



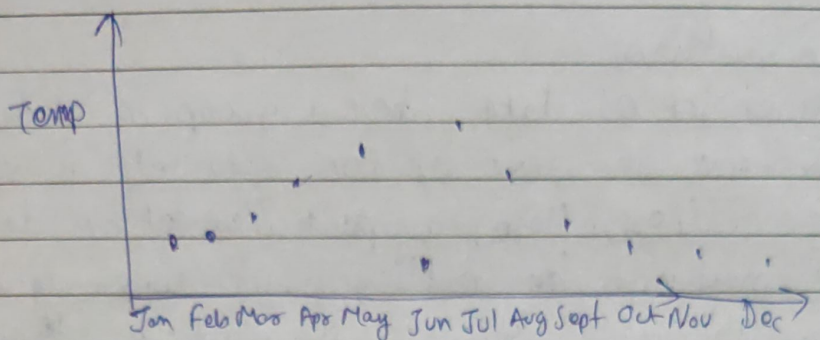

Global outliers:- These are also called point outliers. Global outliers are taken as the simplest form of outliers. When data points deviate from all the rest of the data points in a given data set, it is known as global outlier.



Collective outliers:-
In a given set of data, when a group of datapoints deviates from the rest of the data set is called collective outliers. Here, a particular set of data objects may not be outliers, but when you consider the data objects as a whole, they may behave as outliers. To identify the types of different outlier, you need to go through background info about relationship betⁿ the behavior of outlier shown by diffⁿ data objects.



Contextual outliers:- "Contextual" means this outliers introduced within context. Example:- in the speech recognition technique, the single background noise. Contextual outliers are also known as conditional outliers. Those types of outliers happen if a data object deviates from the other data points because of any specific condⁿ in a given dataset. As we know, these are 2 types of attributes of objects of data: contextual attributes and behavioral attributes.




Haversine :-

The Haversine formula calculates the shortest distance bet^n two points on a sphere using their latitudes measured along the surface. It is imp for use in navigation.

Matplotlib :- It is amazing visualization library in python 2D plots of arrays. Matplotlib is multi-platform data visualization library built on Numpy arrays and designed to work with the broader Scipy Stack. It was introduced by John Hunter in 2002.

One of the greatest benifits of visualization is that it allows us visual access to huge amounts of data in easily digestible visuals. Matplotlib consists of several plots like line, bar, scatter, histogram etc.

Mean Squared error:-

MSE of an estimator measures the average of error squares i.e. the average squared diff" betn the estimated values and true value. It is a risk fu" corresponding to the expected value of the squared error loss. It is always non negative and values close to zero are better. The MSE is second moment of the error and thus incorporates both the variance of the estimator and its bias.

Conclusion:-

In this way we have explored concept correlation and implement linear regression and random forest regression models.

Group B- Exp-2

Aim:- Classify the email using binary classification method. Email spam detection has 2 states:-
a) Normal state - spam
b) Abnormal state - not spam. Use K Nearest Neighbours and Support vector Machine for classification. Analyze their performance.

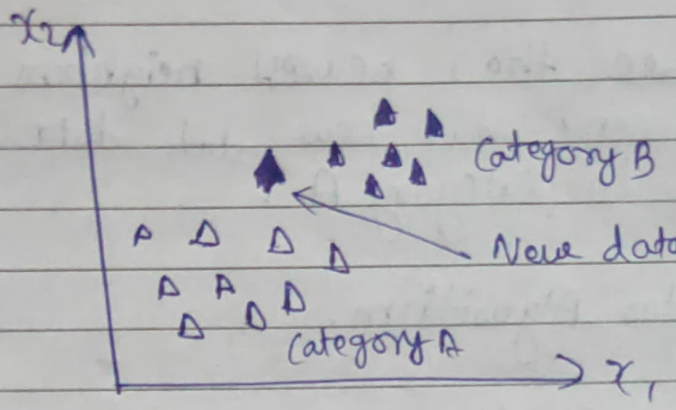
Theory:-
KNN Algorithm:-
- KNN is one of the simplest ML algo$^m$ based on Supervised learning Technique.
- KNN assumes similarities bet$^n$ new case/data and available cases and put new case into the category that is most similar to available categories.
- KNN algo stores all available data and classifies a new data point based on the similarity. This means when new data appears then it can be easily classified into well suite category by using K-NN algo.
- It can be used as regression as well as classification but mostly it is used for classification problems
- It is non-parametric algo, which means it does not make any assumption on underlying data.
- It is also called lazy learner algo because it does not learn from training set immediately instead it stores the dataset and at time of classification, it performs as 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 algo$^m$, as it works on a similar 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.

The KNN working can be explained based on below algo:-

1. Select K no. of neighbors.
2. Calculate Euclidean distance of K no. of neighbors.
3. Take the K nearest neighbors as per the calculated ~~eud~~ euclidean dist.
4. Among these K neighbors, count the no. of data points in each category.
5. Assign the new data points to that category for which the no. of neighbor is max$^m$.
6. Our model is ready.

Suppose we have new data point and we need to put it in the required category. Conside the diagram.

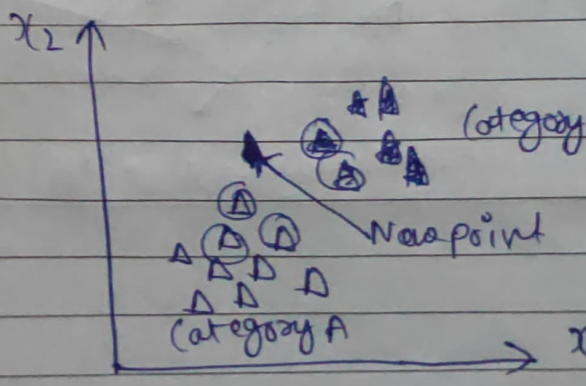



Firstly, we will choose the no. of neighbors, so we will choose the K = 5.

Next, we will calculate the Euclidean dist. bet$^n$ the data points. The Euclidean dist. is the dist. bet$^n$ 2 points, which have already studied in geometry. It can be calculated as:

$$\text{Euclidean distance} = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2}$$

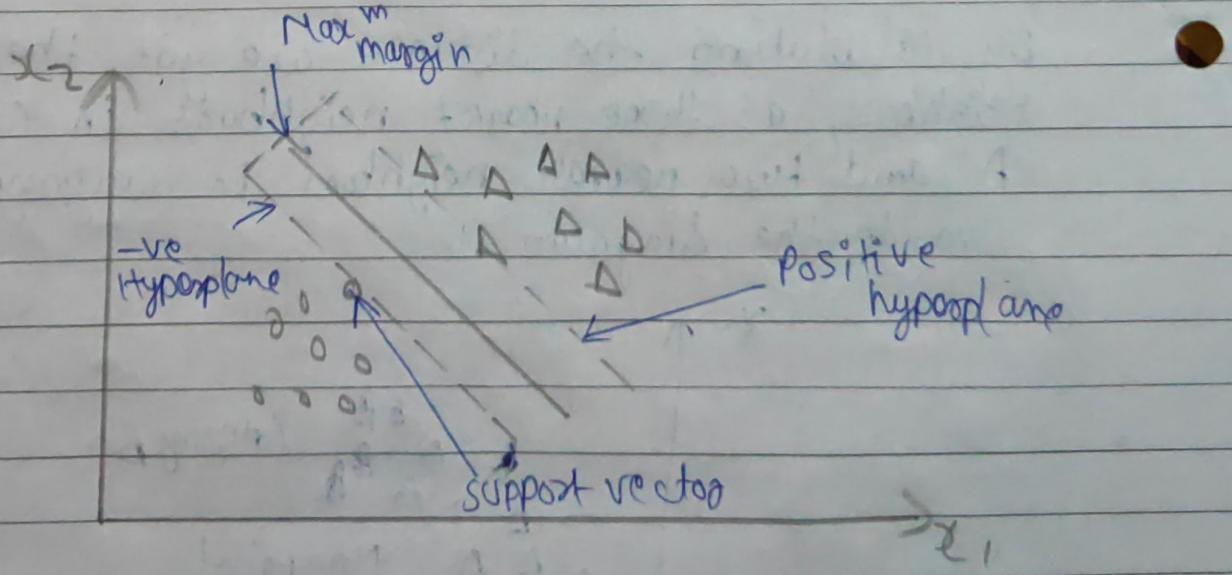
By calculating the distance we got the nearest neighbors, as three nearest neighbors in category A and two nearest neighbors in category B. Conside the diagram.



As we can see the 3 nearest neighbors are from category A, hence this new data point must belong to category A!

## Support Vector Algorithm:-

SVM is one of the most popular Supervised Learning algo$^m$ which is used for classification as well as regression problems. However, primarily, it is used for classification problems in ML. The goal of SVM is to create the best line or decision boundary that can segregate n-dimensional space into classes so that we can easily put new datapoint in correct category in future. This best decision boundary is called a hyperplane.



SVM chooses the extreme points/vectors that help in creating hyperplane. These extreme cases are called as support vectors, and hence algorithm is termed as Support Vector Machine.

Example:-
SVM can be understood with example that we used in KNN classifier. Suppose we see a strange cat that also has some features of dogs, so if we want a model that can accurately ~~modify~~ identify whether it is cat or dog, such model can be created using SVM algo^m. We will first train our model with lots of images of cats and dogs so it can learn about different features of cats and dogs, and then we test it with this strange creature. So as support vector creates a decision boundary bet^n these two data and chooses extreme cases (support vectors), it will see the extreme case of cat or dog. On the basis of support vectors it will classify.

Hyperplane:-
There can be multiple lines/decision boundaries to segregate the classes in n-dimensional space, but we need to find out the best decision boundary that helps to classify the data points. This best boundary is known as hyperplane of SVM.

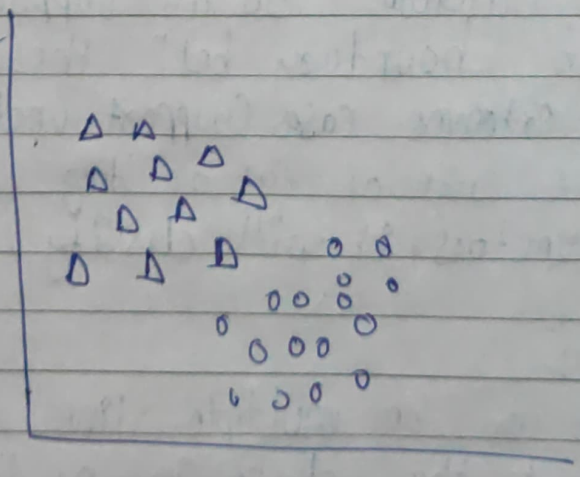
We always create a hyperplane that has max^m margin, which means the max^m dist. bet^n the data points.

Support vectors:-

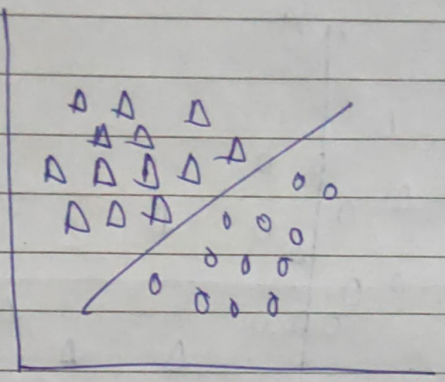
The datapoints or vectors that are closest to the hyperplane and which affect the position of hyperplane are termed as support vector. Since these vectors support the hyperplane, hence called a support vector.

Linear SVM:-

The working of SVM algo^m can be understood by using an example. Suppose we have a dataset that has 2 tags ~~(green~~, and dataset has 2 features x1 and x2. We want a classifier that can classify the pair(x1, x2) of coordinates in either of tag.

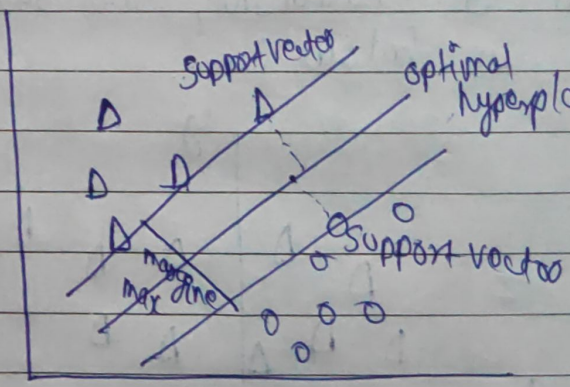


So as it is 2-D space so by just using straight line, we can easily separate these two classes. But there can be multiple lines that can separate these classes.

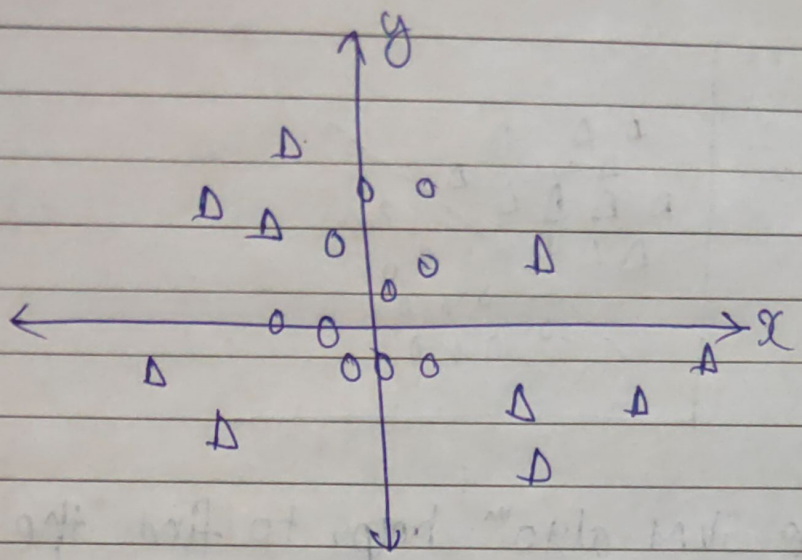


Hence, the SVM algo$^{m}$ helps to find the best line or decision boundary; it is called hyperplane.

SVM algo$^{m}$ finds the closest point of the lines from both classes. These points are called support vectors. The dist. bet$^{n}$ the vectors and the hyperplane is called as margin. And the goal of SVM is to maximize this margin. The hyperplane with max$^{m}$ margin is called the optimal hyperplane.



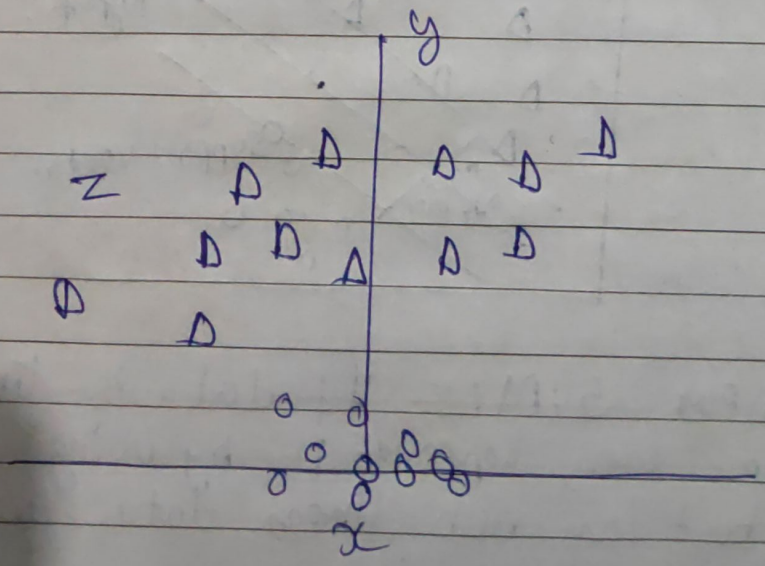

Non-linear SVM:- If data is linearly arranged, then we can separate it by using a straight line, but for non-linear data, we cannot draw a single straight line.



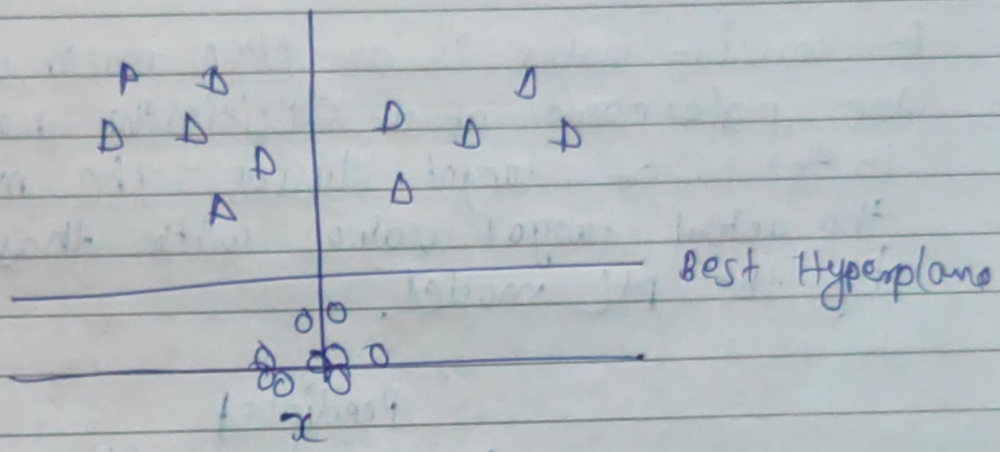

So to separate these points, we need to add one more dimension. For linear data, we have used two dimensions x and y, so for non-linear data, we will add 3rd dimension z. It can be calculated as:-

$$z = x^2 + y^2.$$

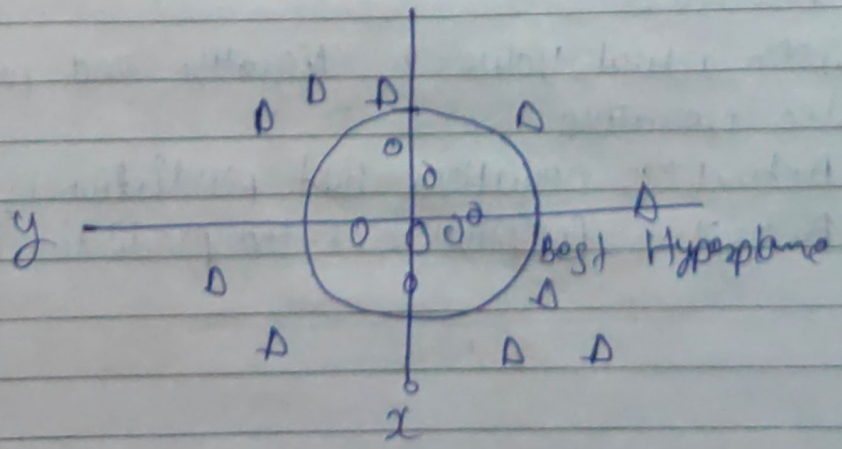
By adding 3rd dimension, the sample space will become as below:-



So now, SVM will divide the datasets into classes in following way.




Since we are in 3-D space, hence it is looking like a plane parallel to the x-axis. If we convert it in 2D space with z = 1, then it ~~we~~ will become as :-




Hence we get circumference of radius 1 in case of non-linear data.

Evaluation Metrics and Score:-

Confusion Matrix:-

A confusion matrix is an NXN matrix used for evaluating the performance of a classification model, where N is the no. of target classes. The matrix compares the actual target values with those predicted by the ML model.

| | | Predicted | |
|---|---|---|---|
| | | Negative(N) | Positive(P) |
| Actual | Negative | TN | FP |
| | Positive | FN | TP |

TP:- when actual value is Positive and predicted is also Positive.

TN:- when actual value is Negative and prediction is also Negative.

FP:- Actual is negative but prediction is Positive.

FN:- Actual is Positive but the prediction is Negative.

Classification Measure:-

Basically, it is extended version of Confusion Matrix. There are measures other than the CM which can help achieve better understanding and analysis of our model and its performance.

a) Accuracy :- Accuracy simply measures how often the classifier makes the correct prediction. It is the ratio bet$^n$ the no. of correct predictions and the no. of predictions. It is a measure of correctness that is achieved in true prediction. In simple words, it tells us how many predictions are actually positive out of total +ve predicted.

$$\text{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN} = \frac{\text{Correct Predictions}}{\text{Total predictions}}$$

b) Precision :- It is a measure of correctness that is achieved in True prediction. In simple words, it tells us how many predictions are actually positive out of all total +ve predicted.

Precision is defined as the ratio of no. of correctly classified +ve classes divided by total no. of predicted +ve classes. Or, out of all predictive +ve classes, how much we predicted correctly.

$$\text{Precision} = \frac{TP}{TP+FP} = \frac{\text{Predictions Actually +ve}}{\text{Total Predicted +ve}}$$

c) Recall:- It is a measure of actual observations which are predicted correctly, i.e. how many observations of +ve class are actually predicted as +ve. It is also known as sensitivity. Recall is a valid choice of evaluation metric when we want to capture as many positive as possible.

Recall is defined as ratio of no. of correctly classified +ve classes divide by the total no. of +ve classes. Or, out of all the positive classes, how much we have predicted correctly

$$\text{Recall} = \frac{TP}{TP+FN} = \frac{\text{Predictions Actually +ve}}{\text{Total Actual positive}}$$

Conclusion:- Thus we implemented KNN and SVM Algo$^m$ successfully

Group B- Exp 06

Title :- K-means clustering.

Aim:- Implement K-means clustering on sales_data_sample dataset. Determine no. of clusters using elbow method.

Theory:-

Clustering:-
It is a technique in which the datapoints are arranged in similar groups dynamically without any pre-assignment of groups.

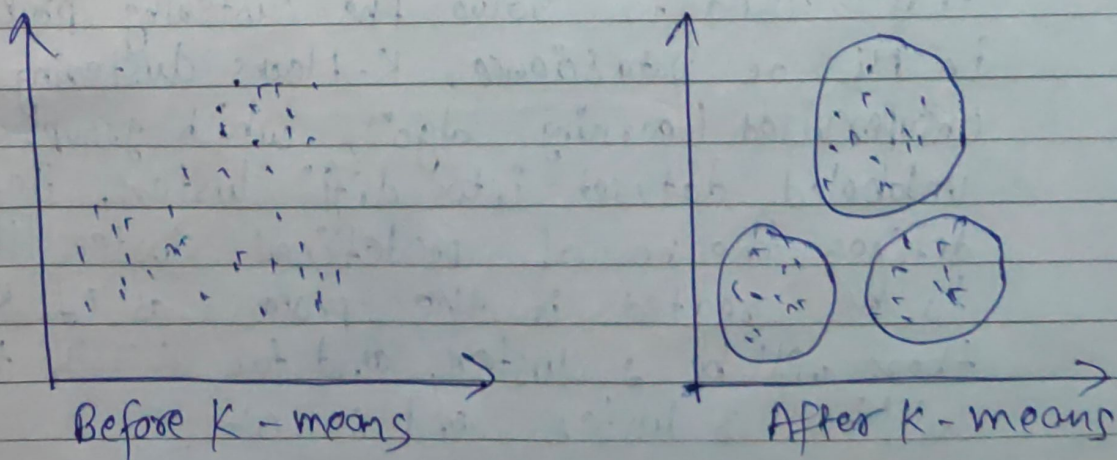
Example, here is simple plot of data points. As you can see, some set of points are closer to each other when compared with others. These could possibly form a group for further analytics:

K-means clustering:-
K-means clustering is an unsupervised learning algo$^m$ that is used to solve the clustering problems in ML or Data Science. K-Means clustering is unsupervised Learning algo$^m$, which groups the unlabeled dataset into diff$^n$ clusters. Here K defines the no. of predefined clusters that need to be created in the process, as if K=2, there will be 2 clusters, and for K=3, there will be 3 clusters, and so on.

It is an iterative algo$^{m}$ that divides the unlabelled dataset into K different clusters in such a way and for K=3, there will be 3 clusters and so on.

It allows us to cluster into diff$^{n}$ groups and convenient way to discover the categories of groups in the unlabeled dataset on its own without the need for any training. It is centroid-based algo$^{m}$, where each cluster is associated with centroid. The main algo$^{m}$ is to minimize the sum of distances bet$^{n}$ the data point and their corresponding clusters.

It mainly performs 2 tasks:-
- Determines the best value for K center points or centroids by an iterative process.
- Assigns each datapoint to its closest K-center. Those data points which are near to the particular K-center, create a cluster.



Working of K-means algo$^m$:-

1. Select no. of K to decide no. of clusters.
2. Select random K points or centeroids.
3. Assign each data point to their closest centroid, which will form the predefined K clusters.
4. Calculate the variance and place a new centroid of each cluster.
5. Repeat the third step, which means reassign each datapoint to the new closest centroid of each cluster.
6. If any reassignment occurs, then go to step 4 else to finish.
7. The model is ready.

Algorithm:-

At high level, the following steps are taken for clustering the data using K-means clustering algo$^m$.

1. Decide the no. of clusters, K, that you desire to group your data points into.
2. Select K random data points as centeroids.
3. Compute the distance from each data point to each centroid. Assign all the data points to the closest cluster centroid. The distance d bet$^n$ any two points, $(x_1, y_1)$ and $(x_2, y_2)$ is calculated as.

$$d = \sqrt{(x_1 - x_2)^2 + (y_1 - y_2)}$$

4. Recompute the centroids of newly formed clusters. The (centroid($x_c, y_c$) of the m data points in a cluster is calculated as.

$$(x_c, y_c) = \left( \frac{\sum_{i=1}^{m} x_i}{m}, \frac{\sum_{i=1}^{m} y_i}{m} \right).$$

It is a simple arithmetic mean of all x-coordinates and y-coordinates of the m data points in the cluster.

5. Repeat steps 3 and 4 until any of following Criteria is met.
   a. Centroids of newly formed clusters do not change.
   b. Points remain in the same clusters.
   c. Max^m number of iterations are reached as desired.

Elbow Method:-

The elbow method is one of the most popular ways to find the optimal no. of clusters. This method uses the concept of WCSS value. WCSS stands for within cluster sum of squares, which defines the total variations within a cluster. The formula to calculate the value of WCSS is:-

WCSS = $\sum P_i$ in cluster 1 distance $(P_i(1))^2 + \sum P_i$ in cluster 2 distance $(P_i(2))^2 + \sum P_i$ in cluster 3 distance $(P_i(3))^2$.

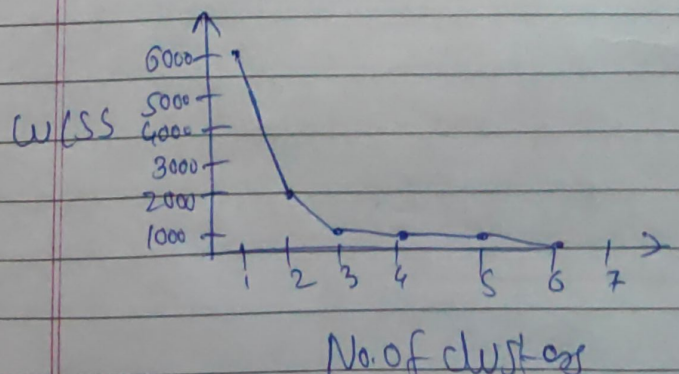
In above formula of WCSS,

$\sum P_i$ in cluster 1 distance $(P_i C_1)^2$: It is sum of the square of distances bet$^n$ each datapoint and its centroid within a cluster 1 and the same for the other 2 terms

To measure the dist. bet$^n$ data points and centroid, we can use any method such as Euclidean dist. or Manhattan dist.

To find the optimal value of clusters, the elbow method follows the below steps:-

- It executes the K-means clustering on given dataset for diff$^n$ K values (ranges from 1-10).
- For each value of K, calculates the WCSS value.
- Plots a curve bet$^n$ calculated WCSS values and no. of clusters K.
- The sharp point of bend or a point of the plot looks like an arm, then that point is considered as the best value of K.



Since graph shows sharp bend ~~at elbow~~, which looks like elbow, hence it is known as elbow method.

Conclusion:- Thus we have successfully implemented K-Means clustering Algorithm.