Experiment No : 1.

- Title :-

Predict the Price of the uber ride from a given Pickup Point to the agreed drop-off location. Perform following Step tasks :-

1. pre-process 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.

- Aim :-

Aim of this practical is to demonstrate data pre-processing techniques, identify and remove outliers, implement and evalute performance of different regression models.

- objectives :- Based on above main aim following are the objectives

1. To understand data preprocessing steps.
2. To identify and remove outliers.
3. To implement different regression models.

What is data Preprocessing :-

Data preprocessing is a data mining technique which is used to transform the raw data in a useful and efficient format.

Steps Involved in Data preprocessing.

1. Data cleaning:- The data can have many irrelevant and missing parts. To handle this part, data cleaning is done. It involves handling of missing data, noisy data etc.

(a) missing Data:-

1) Ignore the tuples:- This approach is suitable only when dataset we have is quite large and multiple values missing within a tuple.

(b) Noisy Data:-

Noisy data is a meaningless data that can't be interpreted by m/c. It can be generated due to faulty data collection, data entry errors, etc. It can be handled in following ways.

1) Binning method:-

1) This method works on sorted data in order to smooth it. The whole data is divided into segments of equal size and then various methods are performed to complete the task.

2) Regression:-

Here data can be made smooth by fitting regression function.

3) clustring :-
This approch groups the similar data in a cluster. The outliers may be undetected or it will fall outside the clusters.

2. Data Transformation :-

1) Normalization :- It is done in order to scale the data values in a specified range (-1.0 to 1.0 or 0.0 to 1.0)

2) Concept Hierarchy Generation :-
Here attributes are converted from lower level in hierarchy. for example. the attributes "city" can be converted to "country".

3. Data Reduction :-
Since data mining is a technique that is used to handle huge amount of data.

1) Data Cube Aggregation :-
Aggregation operation is applied to data for the construction of the data cube.

2) Attribute Subset Selection :- The highly relevent attributes should be used, rest all can be discarded.

3. Numerosity Reduction:- This enable to store the model of instead of whole data, for example: Regression models.

4. Dimensionality Reduction:- This reduce the size of data by encoding mechanisms. It can be lossy or lossless.

Outlier:- An outlier is an object that deviates significantly from the rest of the objects. They can be caused by measurement or execution error.

- Detecting outliers:- clustering based outliers detection, using the k-means clustering techniques, each cluster has a mean value. objects belong to the cluster whose mean value is closest to it.

Algorithm:-

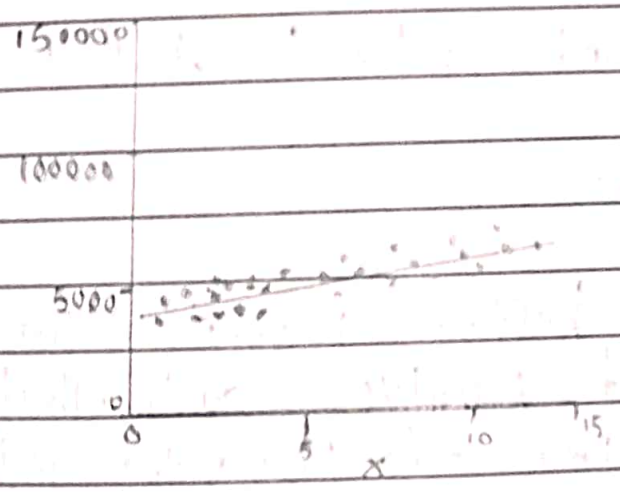
1. Calculate the mean of each cluster.
2. Initialize the threshold value.
3. Calculate the distance of the test data from each mean.
4. Find the nearest cluster to the test data.
5. If (Distance > Threshold) then, outlier.

Correlation matrix:

Correlation is an indication about the changes between two variables. In our previous chapters, we have discussed pearson's Correlation coefficient and the importance of Correlation too.

Linear Regression:

Linear Regression is a machine learning algorithm based on Supervised learning. It performs a regression task. Regression models a target prediction value based on independent variables.



Linear regression performs the task to predict a dependent variable (y) based on a given independent variable (x).

In the above fig, x (input) is the work experience and y (output) is salary of a person. The regression line is the best fit line for our model.

Hypothesis function for linear Regression :-

$$y = \theta_1 + \theta_2 x$$

x :- i/p training the data
y :- labels to data

θ_1 : intercept
θ_2 : coefficient of x.
once we find the best θ_1 and θ_2 values, we get the best fit line.

How to update θ_1 and θ_2 values to get the best fit line?

Cost function (J) :-

By achieving the best-fit regression line, the model aims to predict y value such that the error difference betn predicated your value and true value minimum.

$$\text{minimize } \frac{1}{n} \sum_{i=1}^{n} (pred_i - y_i)^2.$$

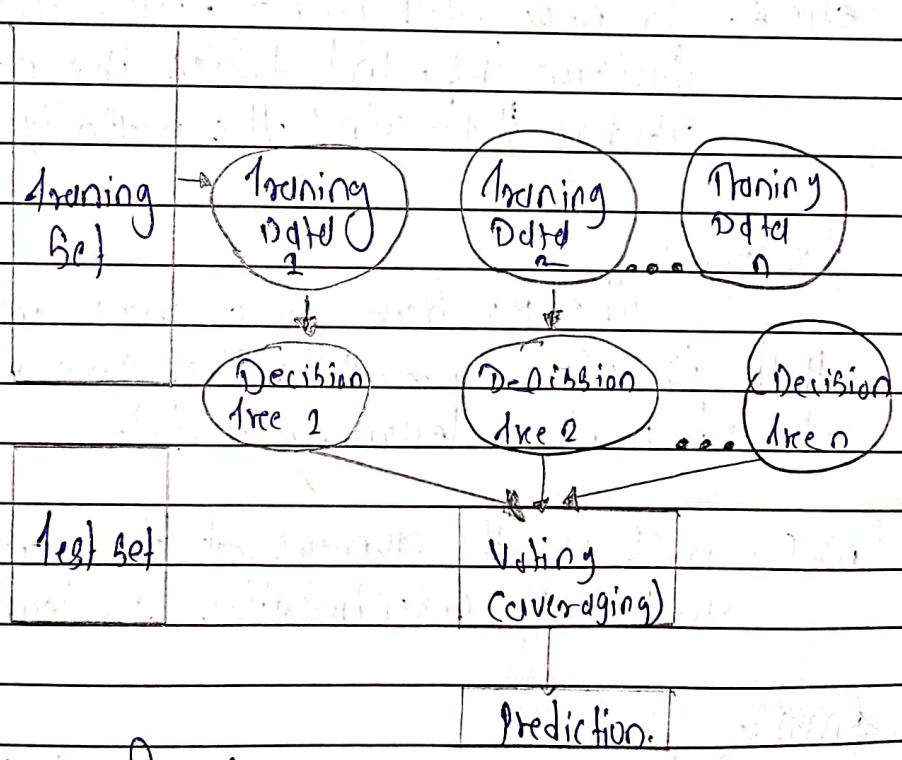
$$J = \frac{1}{n} \sum_{i=1}^{n} (pred_i - y_i)^2.$$

Cost funn (J) of linear Regression is the Root mean sq error (RMSE) between predicated y value (pred) and true y

Random forest algorithm:-

It can be used for both classification and Regression problem in ML. it is based on the Concept of ensemble learning which is process of combining multiple classifiers to be solve a complex problem and to improve the performance of the model.

The greater number of trees in the forest leads to higher accuracy and prevents the problem of overfitting.



How does Random forest algorithm work?

Random forest works in two-phase first is to create the random forest by combining n decision tree, and second

is to make predictions for each tree created in the first phase.

The working process can be explained in the below steps:

Step 1:- Select random k data points from the training set.
Step 2:- Build the decision trees associated with the selected data points (subsets).
Step 3:- Choose the number N for decision trees that you to build.
Step 4:- Repeat Step 1 and 2.
Step 5:- For new data points, find the predictions of decision tree, and assign the new data points to category that wins the majority votes.

R2 Score:-

The R2 Score is a very important metric that is used to evaluate the performance of a regression-based machine learning model.

MSE:- MSE is the average of the squared error that used to the loss function for least squares regression.

RMSE:-

Root mean square error or root mean square deviation is one of the most commonly used measures for evaluating the quality of predications.

Root mean Square error can be expressed as

$$RMSE = \sqrt{\frac{\sum_{i=1}^{n} \| y(i) - y(i) \|^2}{n}}$$

where n is the number of data points, y(i) is the i-th measurement and y(i) is its corresponding predication.

Note :- RMSE is not scale invariant and hence comparison of models using this measure is affected by scale of data.

Conclusion:-

Thus we have implemented linear Regression and Random forest model for given dataset which is pre-processed by us and we checked and checked and compared performance of each with respect to R^2 and RMSE.

Experiment No:2.

Title:- classify the email using the binary classification method. Email Spam detection has two states:

C. Normal state – not Spam
D. Abnormal state – Spam..

Use K-Nearest neighbors and Support Vector machine for classification. Analyze their performance.

Aim:- Aim of this practical is to demonstrate the use of classification methods, analyze the performance of the models for real time application such as Email-Spam Detection.

- Objectives:- Based on above main aim following are the objec-ves.
  1) To understand the classification method.
  2) To implement different classification models.

What is the classification algorithm?

The classification algorithm is a Supervised learning technique that is used to identify the category of new observation on the basis of training data. In classification a program

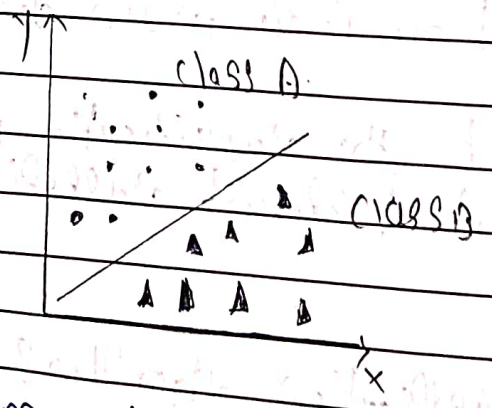
learns from the given dataset or observations and then classified new observations into number of classes or groups. Such as, yes or no, 0 or 1, Spam or not Spam, cat or dog, etc. Classes can be called as target labels or categories.

In the classification algorithm, a discrete o/p function is mapped to the i/p variable (x).

$y = f(x)$, where y = categorical o/ps.

The best example of an ML classification algorithm is Email Spam Detector.

The main goal of the classification algorithm is to identify the category of a given dataset, and these algorithm are mainly used to predict the o/p for the categorical data.



The algorithm which implements the classification on dataset is known as a classifier. There are two types of classifications.

Binary classifier:-
If the classification problem has only two possible outcomes, then it is called as binary classifier.

multi-class classifier:- If a classification problem has more than two outcomes, then it is called as multi-class classifier.

Type of ML classification Algorithms:-

classification Algorithm can be further divided into the mainly two category:

Linear models:-

1) logical regression.
2) Support vector mc.

Non-linear models:

1) K-nearest neighbours.
2) Kernel SVM.
3) Naive Bayes.
4) Decision tree classification.
5) Random forest classification.

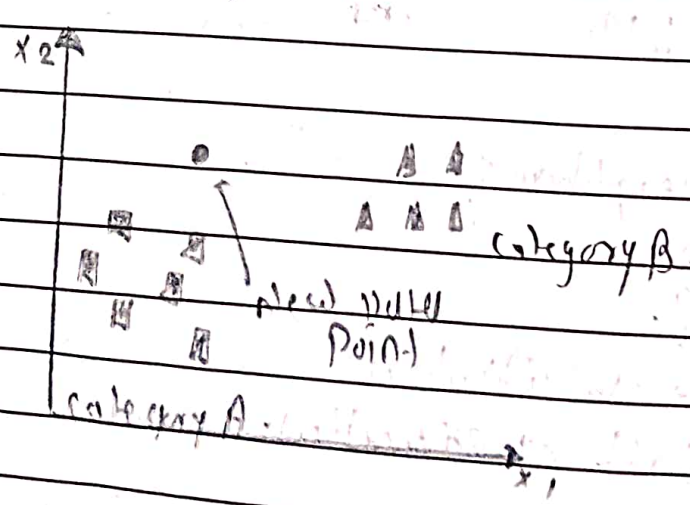
K-nearest neighbor (KNN) Algorithm for ML:-

K-Nearest Neighbor is one of the simplest ml algo based on supervised learning technique.

K-NN algorithm can be called for Regression as well as classification but mostly it is used for the classification problems.

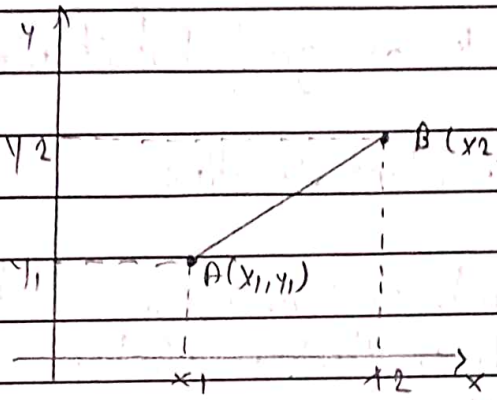
How does K-NN work?

1) Select the no. k of neighbors.
2) Calculate the Euclidean distance of k-no of ne
3) Take the k nearest neighbors as per the calculated distance.
4) Among these k neighbors, count the no. of the dat in each category.
5) Assign the new data points to that category which the number of the neighbor is maximum.
6) our model is ready.



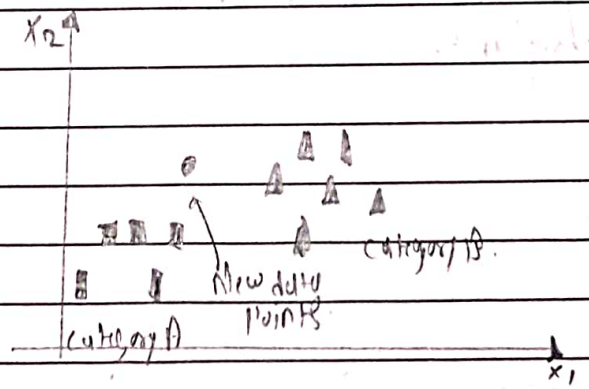
Next, we will calculate the Euclidean distance betw the data point. The Euclidean distance is the dist

between two points, which we have already studied in geometry. It can calculated as:



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

- By calculating the Euclidean distance we got the nearest neighbors in category A and two nearest neighbors in category B. Consider the below image:



How to Select the Values 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 of k is 5.

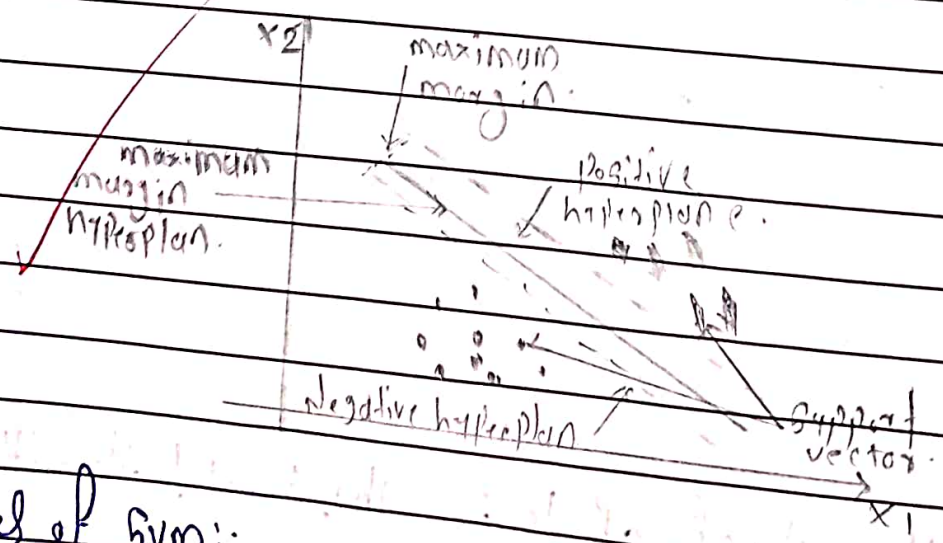
A Very low value for K such as K=1 or K=2, can be noisy and lead to the effects of outliers in the model.

Large values of K are good, but it may find some difficulties.

- Support Vector machine algorithm:-

Support Vector m/c or SVM is one of the most popular supervised learning algorithms, which is used for classification as well as Regression problems. However, primarily, it is used for classification problems in ML.

The goal of the SVM algorithm is to create the best line or decision boundary that can segregate n-dimensional space into classes so that we can easily put the new data point in correct category in the future. This best decision boundary is called a hyperplane.



Types of SVM:-

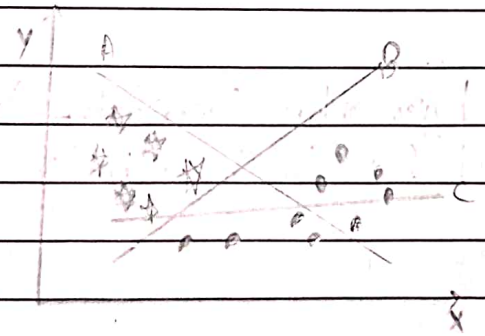
Linear SVM: Linear SVM is used for linearly separable data, which means if a dataset can be classified into two classes by using

a single straight line, then such as data is termed as linearly separable data, and classifier is used called as linear svm classifier.

Non linear Svm - linear Svm is used for non-linearly separated data, which means if a dataset cannot be classified by using a straight line then such data is termed as non-linear data and classifier used is called as non-linear Svm classifier.

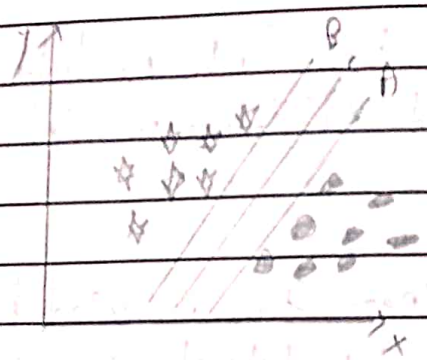
How does it work?

- Identify the right hyper-plane (Scenario-1). Here, we have three hyperplanes (A, B and C). Now, identify the right hyper-plane to classify stars and circles.

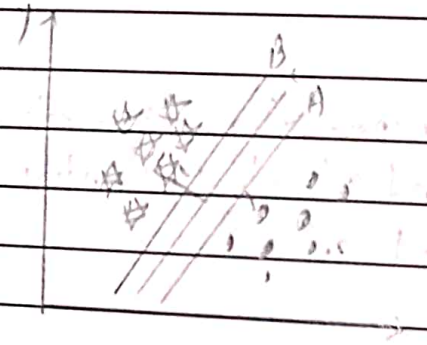


You need to remember a thumb rule to identify the right hyper-plane: "Select the hyper-plane which segregates the two classes better". In this scenario, hyper-plane "B" has excellently performed this job.

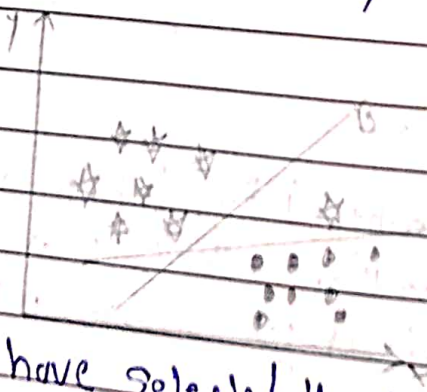
- Identify the right hyper-plane (Scenario-2): Here, we have three hyper-planes (A, B and C) and all are segregating the classes well. Now, How can we identify the right hyper-plane.



Here, maximizing the distance between nearest data point (both) and hyper-plane will help us to decide the right hy
This distance is called margin. Let's look at the below scenario



- identify the right hyper-plane (scenario-3): Hint: use the rules discussed in previous section to identify the right hyper-plane



Some of you may have selected the hyper-plane B as it higher margin compared to A.

Conclusion:- Thus we have implemented classification model, such as k-nearest neighbors and support ve

practical no: 3.

- TITLE: Given a bank customer, build a neural network-based classifier that can determine whether they will never leave or not in the next 6 months.

Dataset Description: The case study is from an open-source dataset from kaggle. The dataset contains 10,000 sample points with 14 distinct features such as customerId, creditScore, Geography, Gender, Age, Tenure, Balance etc.

Perform following steps:
1. Read the dataset.
2. Distinguish the feature and target set and divide the data set into training and test sets.
3. Normalize the train and test data.
4. Initialize and build the model. Identify the points improvement and implement the same.
5. print the accuracy score and confusion matrix.

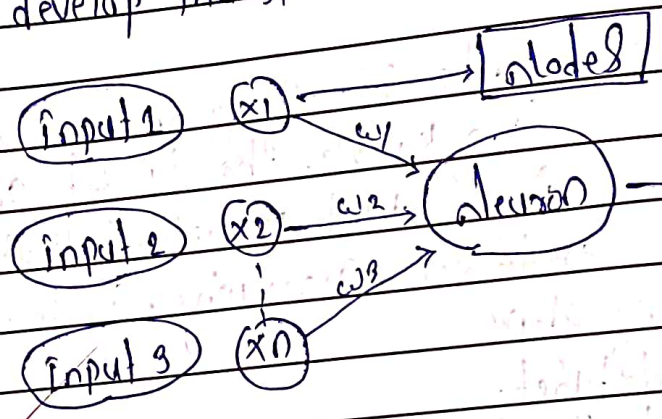
AIM:- Aim of this practical is to design the neural networks classifier, distinguish the dataset features, create training-testing data and normalize it, estimate the accuracy and confusion matrix.

OBJECTIVES: Based on above main aim following are the objectives.

1. To understand the concept of neural network.
2. To implement neural-network based classification models.

Artifical neural network.

The term "Artifical neural network" is derived from Biological neural networks that develop the structure of a human brain.



Dendrites from Biological neural network represent i/p in AI neural networks, cell nucleus represents nodes, synapse represents weights and Axon represents o/p.

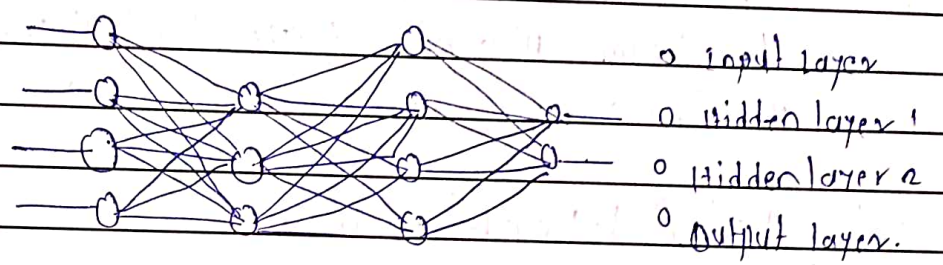
An Artifical neural network in the field of AI, where it tries to minic the network of neurons makes up a human brain so that computers will have an option to understand things and make decisions in a human-like manner.
There are around 1000 billion neurons in the human brain.

We can understand the artifical neural network with an example. Consider an example of a digital logic gates that takes an i/p and gives an o/p. "OR" gate, which takes two i/p. If one or both the i/p are "on", then we get "on" in o/p.

The architecture of an artifical neural network:

To understand the concept of the architecture of an artifical neural network, we have to understand what a neural network consist of. In ordered to define a neural network that consist of a large number of artifical neurons. which are termed units arranged in a sequenced of layers.

Artifical neural network primarily consist of three layers:



Input layer :-

As the name suggests, it accepts i/p in several different formats provided by the programmer.

The artifical neural network takes i/p and computes the weighted sum of the i/p and includes a bias. This computation is represented in the form of a transfer function.

$$\sum_{i=1}^{n} Wi * Xi + b.$$

It determines the weighted total is passed as an i/p to an activation function to produce the o/p. Activation functions choose whether a node should fire or not.

What is normalization in ml?

- mathematically, it is required only when features of ml models differ at ranges.
- mathematically, we can calculate normalization with the below

$$x_n = (x - x_{minimum}) / (x_{maximum} - x_{minimum}).$$

x_n = Value of normalization.
$x_{maximum}$ = Maximum Value of a feature.
$x_{minimum}$ = minimum value of a feature.

Normalization techniques in ml.

- min-max Scaling:- This technique is also referred to as Scaling. We have already discussed above, the min max method helps the dataset to shift and rescale the values of their attributes, so they end up ranging between 0 and 1.

- Standardization Scaling:-
Standardization Scaling is also known as Z-Score normalization, in which values are centered around the mean with a unit standard deviation, which means that the mean becomes zero and the resultant distribution has a unit standard deviation.

$$X' = \frac{X - \mu}{\sigma}$$

• However, unlike min-max Scaling technique, features values are not restricted to a specific range in the standardization technique.

What is the train test split procedure?

Train test split is a model validation procedure that allows you to simulate how a model would perform on new unseen data. Here is how the procedure works:

1. Arrange the Data:

make sure your data is arranged into a format acceptable for train test split. In Scikit-learn, this consist of separating your full data set into "features" and "target."

2) Split The Data:

Split the data set into two pieces - a training set and a testing set. This consist of random sampling without replacement about 75 percent of the rows and putting them into your training set.

3. Train the model:

Train the model on the training set. This is "X train" and "y train" in the image.

Confusion matrix:

The confusion matrix is used to determine the performance of the classification model for a given set of test data. It can

only be determined if the true values for test data are known.

- For the 2 prediction classes of classifiers, the matrix is of 2*2 table. For 3 classes, it is 3*3 table, and so on.
- The matrix is divided into two dimensions, that are predicted values and actual values along with the total number of predictions.
- predicated values are those values, which are predicated by the model and actual values are the true values for the given observations.
- It looks like the below table:-

n = total predictions	Actual: no	Actual: yes
predicated: no	True negative	False positive
predicated: yes	False negative	True positive

- True negative: model has given prediction No, and the real or actual value was also No.
- True positive:- The model has predicated yes, and the actual value also true.

Accuracy Score:- It is one of the important parameters to determine the accuracy of the classification problems. It defines how often the model predicates the correct o/p.

$$\text{Accuracy} = \frac{TP + TN}{TP + FP + FN + TN}.$$

Conclusion:- Thus we implemented a neural network-based classifier. Compute the accuracy score and confusion matrix.

Experiment No:4.

Title:- Gradient Descent Algorithm for logical locating local minima: Case Study on $y = (x+3)^2$ Function from $x = 2$.

Aim:- The aim of this practical is to illustrate the implementation of the gradient descent algorithm for locating local minima of a given function. This involves understanding the gradient descent method, performing iterations to converge towards a local minimum, and evaluting the effectiveness of the algorithm is the context of the function $y = (x+3)^2$ starting from the inital point $(x = 2)$.

OBJECTIVES:-

1. Understanding optimization Techniques:-

Develop a solid understanding of optimization methods, with a focus on the gradient descent algorithm. Grasp the concept of finding local minima in mathematical functions through iterative updates.

2. Algorithm Implementation:-

Implement the gradient descent algorithm particularly. Apply it to the function $y = (x+3)^2$, beginning with an initial value of $(x = 2)$, and observe how the algorithm converges towards a local minimum.

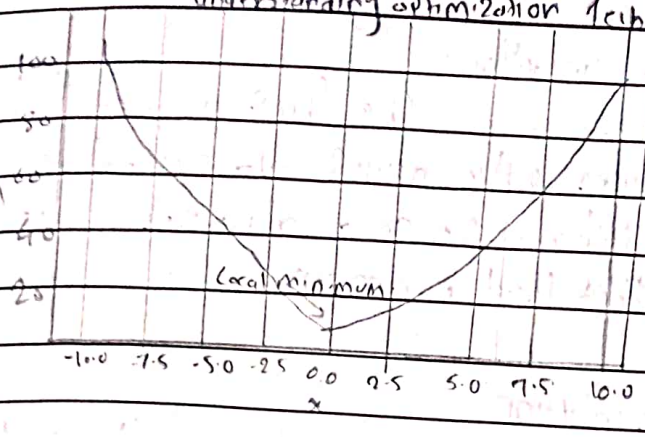
3. Analyzing Convergence:- Examine the iterative behaviour of the gradient decent algorithm gain insights into how the algorithm iteratively refines the value of (x) by utilizing the gradient of the function.

Understanding optimization techniques with a focus on Gradient Descent Algorithm:-

gradient descent is a prominent optimization approach in ml of minimizing a model's loss function. In layman's terms, it entails repeatedly changing the model's parameters until the ideal range of values is discovered that minimize the loss function. The method operates by making tiny steps in the direction of the loss function's negative gradient, or, more specifically, the path of steepest descent. The learning rate α, hyperparameter that regulates the algorithms trade-off between speed and accuracy, affects the size of the steps.

In this aspects, the goal is to establish a robust comprehension of optimization methodology, specifically emphasizing the gradient descent algorithm. optimization techniques are used to find the best possible solution among a set of feasible option.

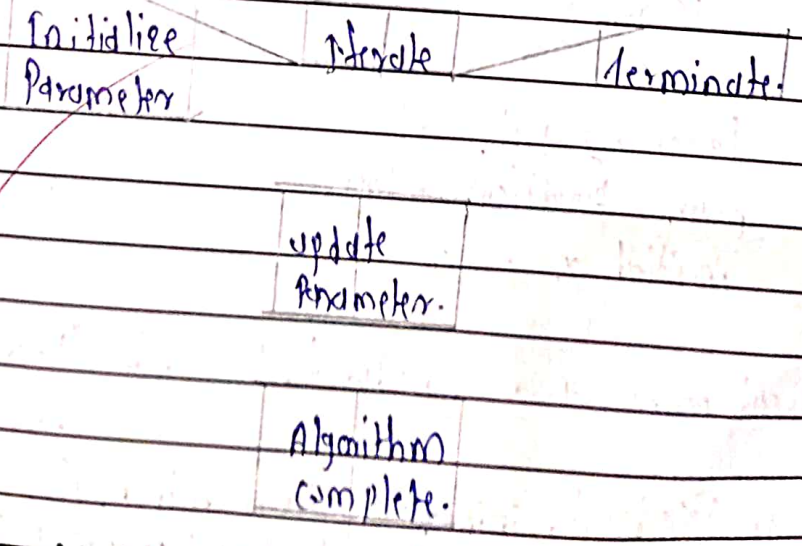
understanding optimization techniques



Grasp the Concept of finding local minima in mathematical functions through iterative updates: This part emphasizes the core concept behind Gradient Descent. Imagine a mathematical function as a landscape with hills and valleys. The goal is to find the lowest point in this landscape.

In each iterations, the algorithm calculates the gradient of the function at the current point, which indicates the direction of the steepest increase.

Algorithm Implementation:.



1. understanding the function:

The fun $y = (x+3)^2$ has a parabolic shape and its minimum value occurs at $(x = -3)$, where $(y = 0)$. The goal of the Gradient Descent algorithm is to iteratively converge towards this local minimum.

2. Algorithm implementation:

We'll implement the Gradient Descent algorithm with a learning rate, a specified number of iterations, and an initial value of $x = 2$.

3. Code Implementation:-

```
def gradient_descent(learning_rate, num_iterations, initial_x):
    x = initial_x
    for _ in range(num_iterations):
        gradient = 2*(x+3) # Gradient of the fun y=(x+3)^2
        x -= learning_rate * gradient # update x using the
        gradient descent rule.
    return x.

learning_rate = 0.1
num_iterations = 100
initial_x = 2.0

local_minima = gradient_descent(learning_rate, num_iterations,
                                initial_x).
print("local minima:", local_minima).
print("function value at local minima:", (local_minima+3)**2)
```

4. output Interpretation:-

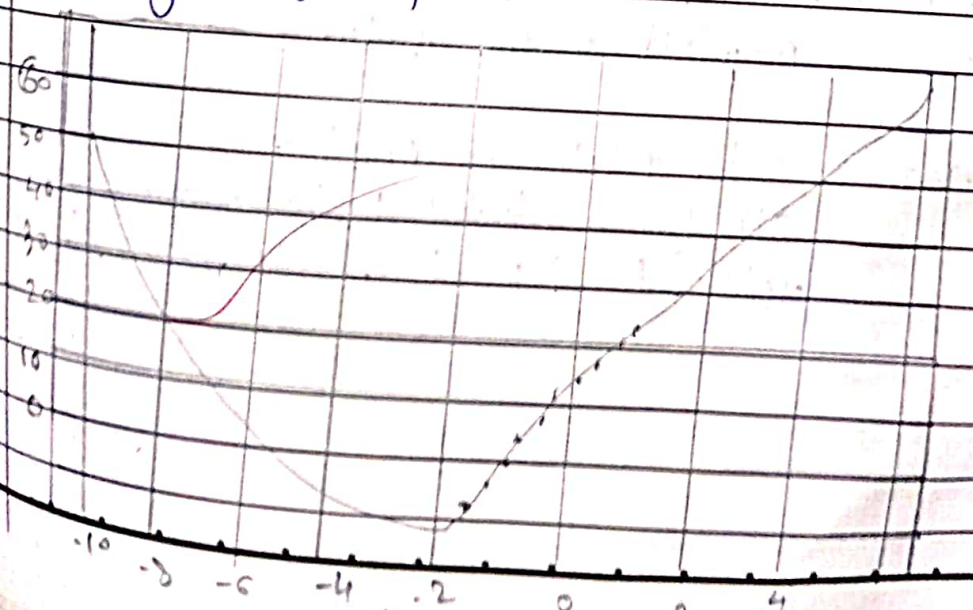
The expected output of the code will look something like this:

local minima :- 2.999999999999995.
Function value at local minima : 4.448920985006 3e-30.

local minima :- This is the value of (x) where the algorithm converged, which should be very close to the actual local minimum value of -3.

keep in mind that due to the limitations of floating-point arithmetic and the finite number of iterations, the values might not be exactly -3 and 0, but they should be extremely close. The purpose of the gradient descent algorithm is to get as close to the actual minimum as possible through iterative refinement.

Analyzing Convergence process:-



1. Initialization: The algorithm starts with an initial value of (x), in this case, (x=2).

2. Calculating Gradient: The gradient of the function $y = (x+3)^2$ at the current (x) value is calculated. The gradient gives us the direction of steepest increase of the fun^n at the point.

3. Updating (x): The algorithm updates the value of (x) by subtracting the gradient multiplied by a learning rate.

4. Iteration: The algorithm repeats steps 2 and 3 for a specific number of iterations.

5. Approching the minimum: As the iterations progress, (x) gradually approaches the local minimum of the fun^n.

6. Slowing Down: As (x) gets closer to the minimum, the magnitude of the gradient decreases.

7. Convergence: With a sufficient number of iterations, the algorithm eventually reaches a point where the updates to (x) result in only marginal changes

Algorithm Evaluation:

1. Expected Theoretical Solution:

The function $y = (x+3)^{12}$ has a local minimum at $(x = -3)$, where $(y = 0)$. This is a known mathematical fact, and it's the point we are aiming to reach using the Gradient Descent algorithm.

Algorithm Result:

After running the gradient descent algorithm should converge to a value of (x) that is very close to -3. This value represents the local minimum that the algorithm identified.

3. Comparison:

Compare the obtained value of (x) from the algorithm with the theoretical local minimum that the algorithm of -3. This value represents the local minimum that the algorithm identified.

Code Example:

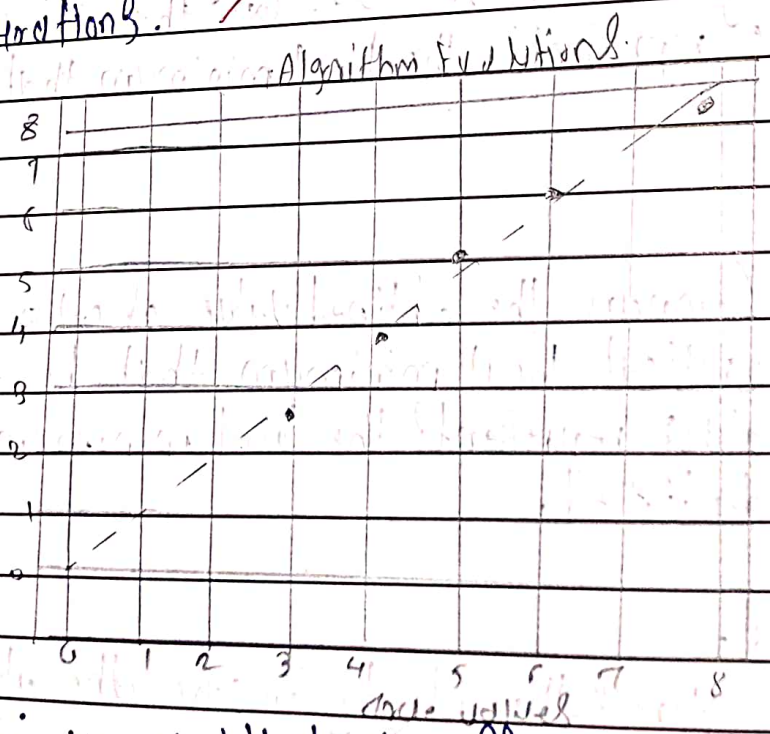
Here's the code to compare the algorithm's result with the theoritical solution:

```
expected_minima = 3.0
print("Expected Theoretical local minimum:",
expected_minima)
```

print ("obtained local minimum from algorithm:", local_minimum).

o/p Interpretation.

After running the code, if the output shows that the "obtained local minimum from algorithm:" is very close to -3 (e.g., -2.9999...), then the algorithm has effectively identified the local minimum. The small difference between the expected theoretical value and the obtained value is due to the limitation of numerical computation and finite iterations.



keep in mind that the effectiveness of the algorithm is also influenced by factors such as the learning rate and the number of iterations. With well-chosen parameters, the algorithm can accurately approximate the local minimum even for more complex functions.

Practical No: 5.

Title:-

Implement k-means clustering / hierarchical clustering on Sales_data_Sample.csv dataset Determine the number of clusters using the elbow method..

Aim:- Aim of this practical is to demonstrate the use of clustering algorithms analyze the performance of the model and implement the elbow curve method to determine the number of clusters.

objectives:- Based on above main aim following are the objectives:-

1. To understand the clustering algorithm.
2. To implement the elbow curve method.

clustering in ML:-

clustering or cluster analysis is a ML technique, which groups the unlabelled dataset. It can be defined as "A way of grouping the data points into different clusters, consisting of similar data points. The objects with the possible similarities remain in a group that has less or no similarities with another group". After applying this clustering technique, each cluster or group is provided to the algorithm, and it deals with the unlabled dataset.

clustering methods:-

- Density-Based methods:- These methods consider the cluster as the

dense region having some similarities and difference from the lower dense region of the space.

• Hierarchical Based methods:-
The cluster formed in this method form a tree type structure based on the hierarchy. New clusters are formed using the previously formed one.
- Agglomerative (bottom-up approach)
- Divisive (top-down approach).

Example: CURE, BIRCH, etc.

• partitioning methods:-
These methods partition the objects into k clusters and each partition forms one cluster. This method is used to optimize an objective criterion similarity function such as when the distance is a major parameter example k-means CLARANS etc.

• Grid based methods:-
In this method, the data space is formulated a finite number of cells that form a grid-like structure. All the clustering operations done on these grids are fast and independent of the number of data objective examples etc.

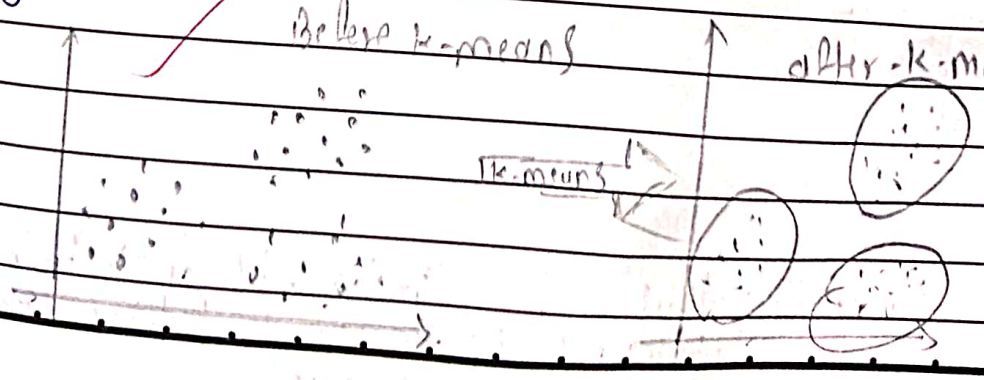
K-Means clustering algorithm:-

• K-means clustering is an unsupervised learning algorithm

is used to solve the clustering problems in machine learning or data science. In this topic.

What is k-means algorithm?

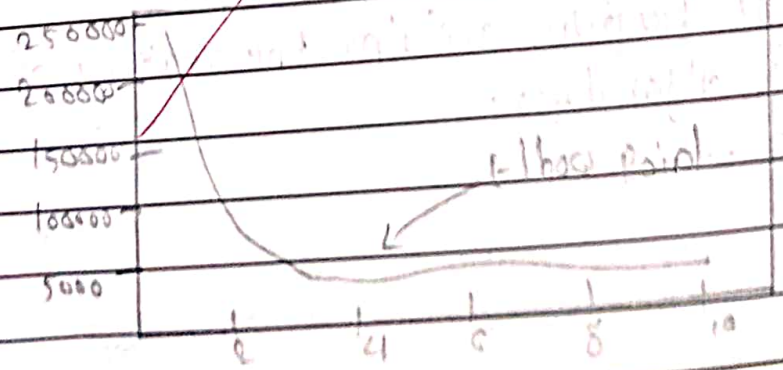
- k-means clustering is an unsupervised learning algorithm, which groups the unlabled dataset into different clusters.
- It is an iterative algorithm that divides the unlabeled dataset into K different clusters in such a way that each dataset belongs only one group that has similar properties.
- It is a centroid-based algorithm, where each cluster is associated with a centroid. The main aim of this algorithm is to minimize the sum of distances between the data points and their corresponding clusters.
- The k-means clustering algorithm mainly perform two tasks:
  - Determine 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.
- Hence each cluster has data points with some commonalities, and it is away from other clusters.
- The below diagram explains the working of the k-means clustering algorithm:



How does the k-means algorithm work?

Step 1:- Select the number k to decide the number of clusters.
Step 2 - Select random k points to centroids.
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 steps, 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.

Elbow method:-
In the Elbow method, we are actually varifying the number of clusters (k) from 1-10 for each value of K, we are calculating WCSS. WCSS is the sum of squared distance between each point and the centroid in a cluster. When we plot the WCSS with the K-value the plot looks like an Elbow.



Conclusion:- Thus we have implemented a clustering algorithm i.e. clustering and using elbow method determine the number of optimal value of K-means.