MACHINE LEARNING
PRACTICAL INDEX

SR.NO.	TITLE
1.	Classify the email using the binary classification method. Email Spam detection has two states: a) Normal State – Not Spam, b) Abnormal State – Spam. Use K-Nearest Neighbors and Support Vector Machine for classification. Analyze their performance.
2.	Given a bank customer, build a neural network-based classifier that can determine whetherthey will leave or not in the next 6 months.
3.	Implement Gradient Descent Algorithm to find the local minima of a function.
4.	Implement K-Nearest Neighbors algorithm on diabetes.csv dataset. Compute confusion matrix, accuracy, error rate, precision and recall on the given dataset.
5.	Implement K-Means clustering/ hierarchical clustering on sales_data_sample.csv dataset. Determine the number of clusters using the elbow method.
6.	**Mini Project** Use the following dataset to analyze ups and downs in the market and predict future stock price returns based on Indian Market data from 2000 to 2020. Dataset Link: https://www.kaggle.com/datasets/sagara9595/stock-data **OR** Build a machine learning model that predicts the type of people who survived the Titanic shipwreck using passenger data (i.e. name, age, gender, socio-economic class, etc.). Dataset Link: https://www.kaggle.com/competitions/titanic/data

EXPT NO.1

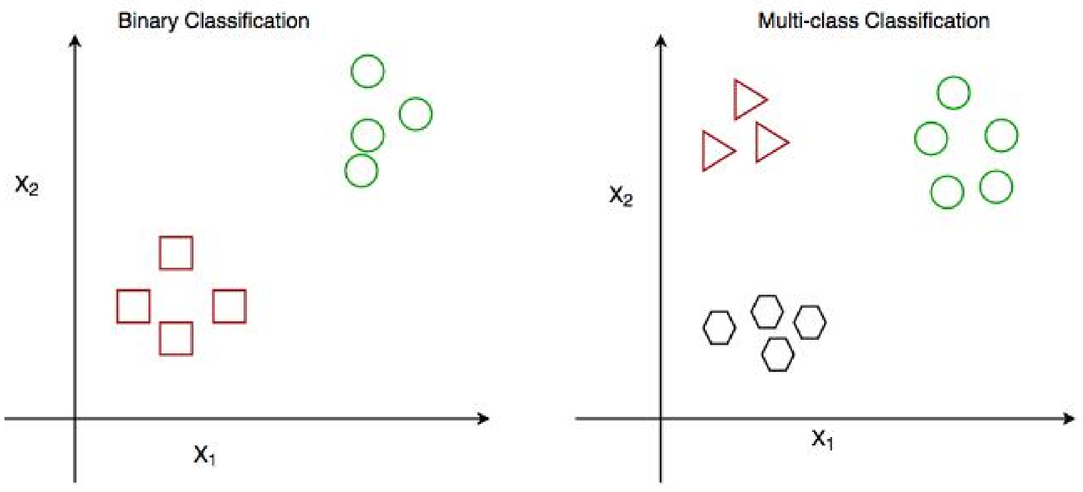
TITLE:Classify the email using the binary classification method. Email Spam detection has two states: a) Normal State – Not Spam, b) Abnormal State – Spam. Use K-Nearest Neighbors and Support Vector Machine for classification. Analyze their performance.

What is Classification?
Classification is a process of categorizing data or objects into predefined classes or categories based on their features or attributes. In machine learning, classification is a type of supervised learning technique where an algorithm is trained on a labeled dataset to predict the class or category of new, unseen data.

The main objective of classification is to build a model that can accurately assign a label or category to a new observation based on its features. For example, a classification model might be trained on a dataset of images labeled as either dogs or cats and then used to predict the class of new, unseen images of dogs or cats based on their features such as color, texture, and shape.

Types of Classification

1. **Binary Classification**: In binary classification, the goal is to classify the input into one of two classes or categories. Example – On the basis of the given health conditions of a person, we have to determine whether the person has a certain disease or not.
2. **Multiclass Classification**: In multi-class classification, the goal is to classify the input into one of several classes or categories. For Example – On the basis of data about different species of flowers, we have to determine which specie our observation belongs to.



Types of classification algorithms

- **1.Linear Classifiers:** Linear models create a linear decision boundary between classes. They are simple and computationally efficient. Some of the linear **classification** models are as follows:
- Logistic Regression
- Support Vector Machines having kernel = ‘linear’

- Single-layer Perceptron
- Stochastic Gradient Descent (SGD) Classifier

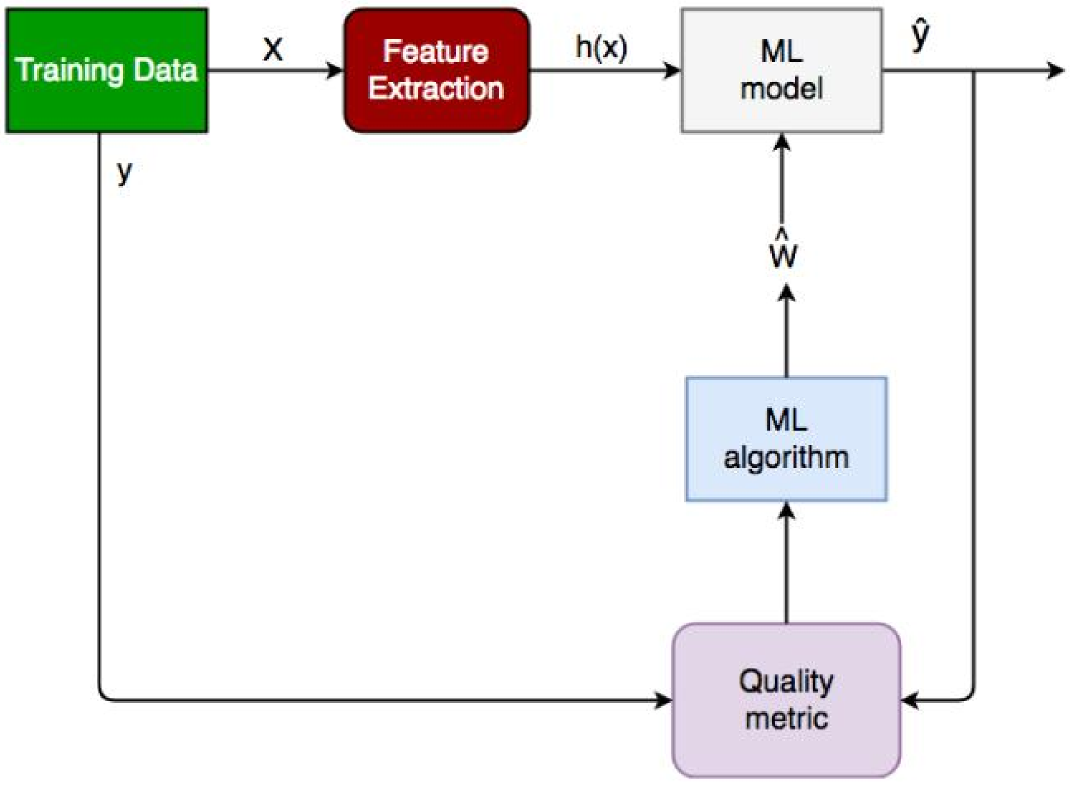
2.Non-linear Classifiers: Non-linear models create a non-linear decision boundary between classes. They can capture more complex relationships between the input features and the target variable. Some of the non-linear **classification** models are as follows:

- K-Nearest Neighbours
- Kernel SVM
- Naive Bayes
- Decision Tree Classification
- Ensemble learning classifiers:
  - Random Forests,
  - AdaBoost,
  - Bagging Classifier,
  - Voting Classifier,
  - ExtraTrees Classifier
- Multi-layer Artificial Neural Networks

Classification model Evaluations

1. Classification Accuracy: The proportion of correctly classified instances over the total number of instances in the test set. It is a simple and intuitive metric but can be misleading in imbalanced datasets where the majority class dominates the accuracy score.
2. Confusion matrix**:** A table that shows the number of true positives, true negatives, false positives, and false negatives for each class, which can be used to calculate various evaluation metrics.
3. Precision and Recall: Precision measures the proportion of true positives over the total number of predicted positives, while recall measures the proportion of true positives over the total number of actual positives. These metrics are useful in scenarios where one class is more important than the other, or when there is a trade-off between false positives and false negatives.
4. F1-Score: The harmonic mean of precision and recall, calculated as 2 x (precision x recall) / (precision + recall). It is a useful metric for imbalanced datasets where both precision and recall are important.
5. ROC curve and AUC: The Receiver Operating Characteristic (ROC) curve is a plot of the true positive rate (recall) against the false positive rate (1-specificity) for different threshold values of the classifier's decision function. The Area Under the Curve (AUC) measures the overall performance of the classifier, with values ranging from 0.5 (random guessing) to 1 (perfect classification).
6. Cross-validation**:** A technique that divides the data into multiple folds and trains the model on each fold while testing on the others, to obtain a more robust estimate of the model's performance.

It is important to choose the appropriate evaluation metric(s) based on the specific problem and requirements, and to avoid overfitting by evaluating the model on independent test data.



Classification Lifecycle

Applications of Classification Algorithm

Classification algorithms are widely used in many real-world applications across various domains, including:

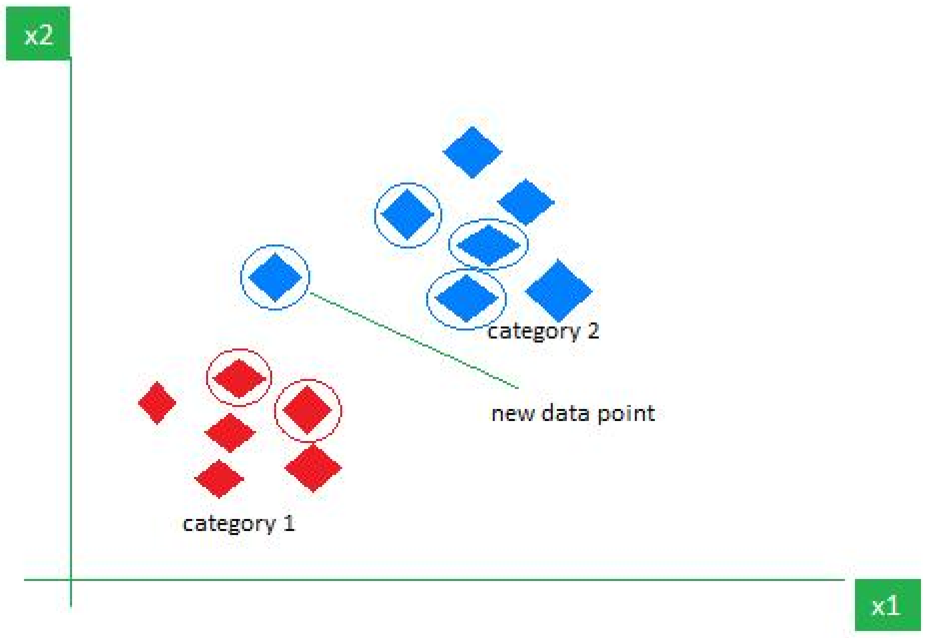
- Email spam filtering
- Credit risk assessment
- Medical diagnosis
- Image classification
- Sentiment analysis.
- Fraud detection
- Quality control
- Recommendation systems

A. K-Nearest Neighbor(KNN) Algorithm

K-Nearest Neighbours is one of the most basic yet essential classification algorithms in Machine Learning. It belongs to the supervised learning domain and finds intense application in pattern recognition, data mining, and intrusion detection.

It is widely disposable in real-life scenarios since it is non-parametric, meaning, it does not make any underlying assumptions about the distribution of data (as opposed to other algorithms such as GMM, which assume a Gaussian distribution of the given data). We are given some prior data (also called training data), which classifies coordinates into groups identified by an attribute.

As an example, consider the following table of data points containing two features:



KNN Algorithm working visualization

Now, given another set of data points (also called testing data), allocate these points to a group by analyzing the training set. Note that the unclassified points are marked as 'White'.
If we plot these points on a graph, we may be able to locate some clusters or groups. Now, given an unclassified point, we can assign it to a group by observing what group its nearest neighbors belong to. This means a point close to a cluster of points classified as 'Red' has a higher probability of getting classified as 'Red'.

Intuitively, we can see that the first point (2.5, 7) should be classified as 'Green' and the second point (5.5, 4.5) should be classified as 'Red'.

How to choose the value of k for KNN Algorithm?
The value of k is very crucial in the KNN algorithm to define the number of neighbors in the algorithm. The value of k in the k-nearest neighbors (k-NN) algorithm should be chosen based on the input data. If the input data has more outliers or noise, a higher value of k would be better. It is recommended to choose an odd value for k to avoid ties in classification. Cross-validation methods can help in selecting the best k value for the given dataset.

Applications of the KNN Algorithm
1. **Data Preprocessing** – While dealing with any Machine Learning problem we first perform the EDA part in which if we find that the data contains missing values then there are multiple imputation methods are available as well. One of such method is KNN Imputer which is quite effective ad generally used for sophisticated imputation methodologies.
2. **Pattern Recognition** – KNN algorithms work very well if you have trained a KNN algorithm using the MNIST dataset and then performed the evaluation process then you must have come across the fact that the accuracy is too high.

3. **Recommendation Engines** – The main task which is performed by a KNN algorithm is to assign a new query point to a pre-existed group that has been created using a huge corpus of datasets. This is exactly what is required in the recommender systems to assign each user to a particular group and then provide them recommendations based on that group's preferences.

Advantages of the KNN Algorithm

1. **Easy to implement** as the complexity of the algorithm is not that high.
2. **Adapts Easily** – As per the working of the KNN algorithm it stores all the data in memory storage and hence whenever a new example or data point is added then the algorithm adjusts itself as per that new example and has its contribution to the future predictions as well.
3. **Few Hyperparameters** – The only parameters which are required in the training of a KNN algorithm are the value of k and the choice of the distance metric which we would like to choose from our evaluation metric.

Disadvantages of the KNN Algorithm

1. **Does not scale** – As we have heard about this that the KNN algorithm is also considered a Lazy Algorithm. The main significance of this term is that this takes lots of computing power as well as data storage. This makes this algorithm both time-consuming and resource exhausting.
2. **Curse of Dimensionality** – There is a term known as the peaking phenomenon according to this the KNN algorithm is affected by the curse of dimensionality which implies the algorithm faces a hard time classifying the data points properly when the dimensionality is too high.
3. **Prone to Overfitting** – As the algorithm is affected due to the curse of dimensionality it is prone to the problem of overfitting as well. Hence generally feature selection as well as dimensionality reduction techniques are applied to deal with this problem.

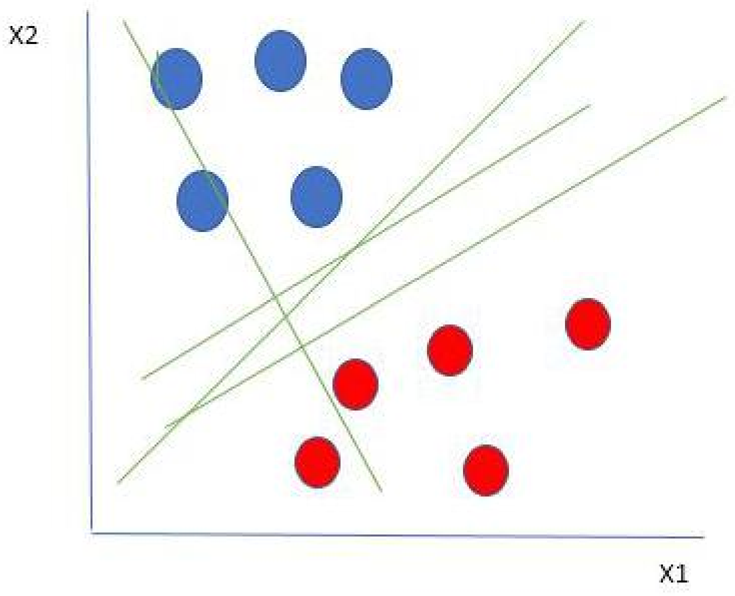
B. Support Vector Machine (SVM) Algorithm

Support Vector Machine (SVM) is a powerful machine learning algorithm used for linear or nonlinear classification, regression, and even outlier detection tasks. SVMs can be used for a variety of tasks, such as text classification, image classification, spam detection, handwriting identification, gene expression analysis, face detection, and anomaly detection. SVMs are adaptable and efficient in a variety of applications because they can manage high-dimensional data and nonlinear relationships.

SVM algorithms are very effective as we try to find the maximum separating hyperplane between the different classes available in the target feature.

Support Vector Machine (SVM) is a supervised machine learning algorithm used for both classification and regression. Though we say regression problems as well it's best suited for classification. The main objective of the SVM algorithm is to find the optimal hyperplane in an N-dimensional space that can separate the data points in different classes in the feature space. The hyperplane tries that the margin between the closest points of different classes should be as maximum as possible. The dimension of the hyperplane depends upon the number of features. If the number of input features is two, then the hyperplane is just a line. If the number of input features is three, then the hyperplane becomes a 2-D plane. It becomes difficult to imagine when the number of features exceeds three.

Let's consider two independent variables x1, x2, and one dependent variable which is either a blue circle or a red circle.

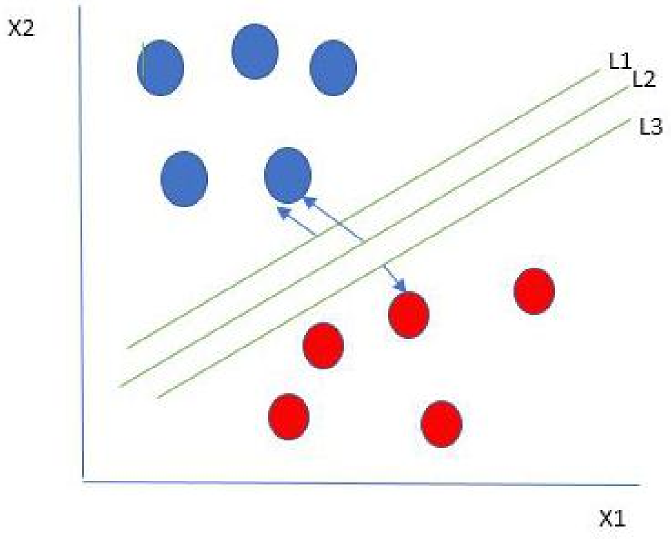


Linearly Separable Data points

From the figure above it's very clear that there are multiple lines (our hyperplane here is a line because we are considering only two input features x1, x2) that segregate our data points or do a classification between red and blue circles. So how do we choose the best line or in general the best hyperplane that segregates our data points?

How does SVM work?

One reasonable choice as the best hyperplane is the one that represents the largest separation or margin between the two classes.



Multiple hyperplanes separate the data from two classes

So we choose the hyperplane whose distance from it to the nearest data point on each side is maximized. If such a hyperplane exists it is known as the **maximum-margin hyperplane/hard margin**

Support Vector Machine Terminology

1. **Hyperplane:** Hyperplane is the decision boundary that is used to separate the data points of different classes in a feature space. In the case of linear classifications, it will be a linear equation i.e. $wx+b = 0$.
2. **Support Vectors:** Support vectors are the closest data points to the hyperplane, which makes a critical role in deciding the hyperplane and margin.
3. **Margin**: Margin is the distance between the support vector and hyperplane. The main objective of the support vector machine algorithm is to maximize the margin. The wider margin indicates better classification performance.
4. **Kernel**: Kernel is the mathematical function, which is used in SVM to map the original input data points into high-dimensional feature spaces, so, that the hyperplane can be easily found out even if the data points are not linearly separable in the original input space. Some of the common kernel functions are linear, polynomial, radial basis function(RBF), and sigmoid.
5. **Hard Margin:** The maximum-margin hyperplane or the hard margin hyperplane is a hyperplane that properly separates the data points of different categories without any misclassifications.
6. **Soft Margin:** When the data is not perfectly separable or contains outliers, SVM permits a soft margin technique. Each data point has a slack variable introduced by the soft-margin SVM formulation, which softens the strict margin requirement and permits certain misclassifications or violations. It discovers a compromise between increasing the margin and reducing violations.
7. **C:** Margin maximisation and misclassification fines are balanced by the regularisation parameter C in SVM. The penalty for going over the margin or misclassifying data items is decided by it. A stricter penalty is imposed with a greater value of C, which results in a smaller margin and perhaps fewer misclassifications.
8. **Hinge Loss:** A typical loss function in SVMs is hinge loss. It punishes incorrect classifications or margin violations. The objective function in SVM is frequently formed by combining it with the regularisation term.
9. **Dual Problem:** A dual Problem of the optimisation problem that requires locating the Lagrange multipliers related to the support vectors can be used to solve SVM. The dual formulation enables the use of kernel tricks and more effective computing.

CONCLUSION:

EXPT NO.2

II)Given a bank customer, build a neural network-based classifier that can determine whetherthey will 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.

Link to the Kaggle project: https://www.kaggle.com/barelydedicated/bank-customer-churn-modeling 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 of improvement and implement the same.

1. Print the accuracy score and confusion matrix (5 points).

THEORY

Neural Networks

Neural networks are artificial systems that were inspired by biological neural networks. These systems learn to perform tasks by being exposed to various datasets and examples without any task-specific rules. The idea is that the system generates identifying characteristics from the data they have been passed without being programmed with a pre-programmed understanding of these datasets. Neural networks are based on computational models for threshold logic. Threshold logic is a combination of algorithms and mathematics. Neural networks are based either on the study of the brain or on the application of neural networks to artificial intelligence. The work has led to improvements in finite automata theory. Components of a typical neural network involve neurons, connections which are known as synapses, weights, biases, propagation function, and a learning rule. Neurons will receive an input $p_j(t)$ from predecessor neurons that have an activation $a_j(t)$, threshold θ_j , an activation function f, and an output function f_{out}. Connections consist of connections, weights and biases which rules how neuron i transfers output to neuron j. Propagation computes the input and outputs the output and sums the predecessor neurons function with the weight. The learning of neural network basically refers to the adjustment in the free parameters i.e. weights and bias. There are basically three sequence of events of learning process.

These includes:

1. The neural network is simulated by an new environment.
2. Then the free parameters of the neural network is changed as a result of this simulation.
3. The neural network then responds in a new way to the environment because of the changes in its free parameters.

Supervised vs Unsupervised Learning: Neural networks learn via supervised learning; Supervised machine learning involves an input variable *x* and corresponding desired output variable *y*. Here we introduce the concept of teacher who has knowledge about the environment. Thus we can say that the teacher has both input-output set. The neural network is unaware of the environment. The input is exposed to both teacher and neural network, the neural network generates an output based on the input. This output is then compared with the desired output that teacher has and simultaneously an error signal is produced. The free parameters of the network is then step by step adjusted so that error is minimum. The learning stops when the algorithm reaches an acceptable level of performance. Unsupervised machine learning has input data X and no corresponding output variables. The goal is to model the underlying structure of the data to understand more about the data. The keywords for supervised machine learning are classification and regression. For unsupervised machine learning, the keywords are clustering and association.

Evolution of Neural Networks: Hebbian learning deals with neural plasticity. Hebbian learning is unsupervised and deals with long-term potentiation. Hebbian learning deals with pattern recognition and exclusive-or circuits deal with if-then rules. Backpropagation solved the exclusive-or issue that Hebbian learning could not handle. This also allowed for multi-layer networks to be feasible and efficient. If an error was found, the error was solved at each layer by modifying the weights at each node. This led to the development of support vector machines, linear classifiers, and max-pooling. The vanishing gradient problem affects feedforward networks that use back propagation and recurrent neural network. This is known as deep-learning. Hardware-based designs are used for biophysical simulation and neurotrophic computing. They have large scale component analysis and convolution creates new class of neural computing with analog. This also solved back-propagation for many-layered feedforward neural networks. Convolutional networks are used for alternating between convolutional layers and max-pooling layers with connected layers (fully or sparsely connected) with a final classification layer. The learning is done without unsupervised pre-training. Each filter is equivalent to a weights vector that has to be trained. The shift variance has to be guaranteed to dealing with small and large neural networks. This is being resolved in Development Networks. Some of the other learning techniques involve error-correction learning, memory-based learning and competitive learning.

Types of Neural Networks

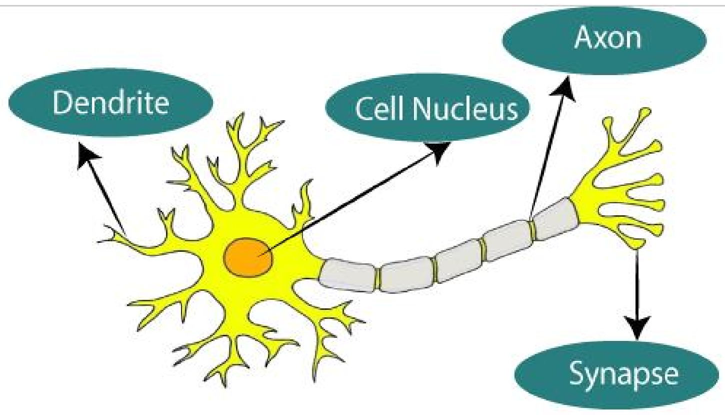
There are *seven* types of neural networks that can be used.

- Multilayer Perceptron (MLP): A type of feedforward neural network with three or more layers, including an input layer, one or more hidden layers, and an output layer. It uses nonlinear activation functions.
- Convolutional Neural Network (CNN): A neural network that is designed to process input data that has a grid-like structure, such as an image. It uses convolutional layers and pooling layers to extract features from the input data.
- Recursive Neural Network (RNN): A neural network that can operate on input sequences of variable length, such as text. It uses weights to make structured predictions.

- Recurrent Neural Network (RNN): A type of neural network that makes connections between the neurons in a directed cycle, allowing it to process sequential data.
- Long Short-Term Memory (LSTM): A type of RNN that is designed to overcome the vanishing gradient problem in training RNNs. It uses memory cells and gates to selectively read, write, and erase information.
- Sequence-to-Sequence (Seq2Seq): A type of neural network that uses two RNNs to map input sequences to output sequences, such as translating one language to another.
- Shallow Neural Network: A neural network with only one hidden layer, often used for simpler tasks or as a building block for larger networks.

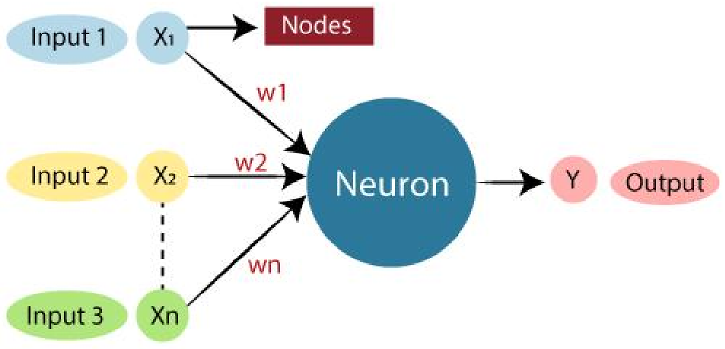
Applications of Artificial Neural Networks

1. **Social Media:** Artificial Neural Networks are used heavily in Social Media. For example, let's take the **'People you may know'** feature on Facebook that suggests people that you might know in real life so that you can send them friend requests. Well, this magical effect is achieved by using Artificial Neural Networks that analyze your profile, your interests, your current friends, and also their friends and various other factors to calculate the people you might potentially know. Another common application of Machine Learning in social media is **facial recognition**. This is done by finding around 100 reference points on the person's face and then matching them with those already available in the database using convolutional neural networks.
2. **Marketing and Sales:** When you log onto E-commerce sites like Amazon and Flipkart, they will recommend your products to buy based on your previous browsing history. Similarly, suppose you love Pasta, then Zomato, Swiggy, etc. will show you restaurant recommendations based on your tastes and previous order history. This is true across all new-age marketing segments like Book sites, Movie services, Hospitality sites, etc. and it is done by implementing **personalized marketing**. This uses Artificial Neural Networks to identify the customer likes, dislikes, previous shopping history, etc., and then tailor the marketing campaigns accordingly.
3. **Healthcare**: Artificial Neural Networks are used in Oncology to train algorithms that can identify cancerous tissue at the microscopic level at the same accuracy as trained physicians. Various rare diseases may manifest in physical characteristics and can be identified in their premature stages by using **Facial Analysis** on the patient photos. So the full-scale implementation of Artificial Neural Networks in the healthcare environment can only enhance the diagnostic abilities of medical experts and ultimately lead to the overall improvement in the quality of medical care all over the world.
4. **Personal Assistants:** I am sure you all have heard of Siri, Alexa, Cortana, etc., and also heard them based on the phones you have!!! These are personal assistants and an example of speech recognition that uses **Natural Language Processing** to interact with the users and formulate a response accordingly. Natural Language Processing uses artificial neural networks that are made to handle many tasks of these personal assistants such as managing the language syntax, semantics, correct speech, the conversation that is going on, etc.



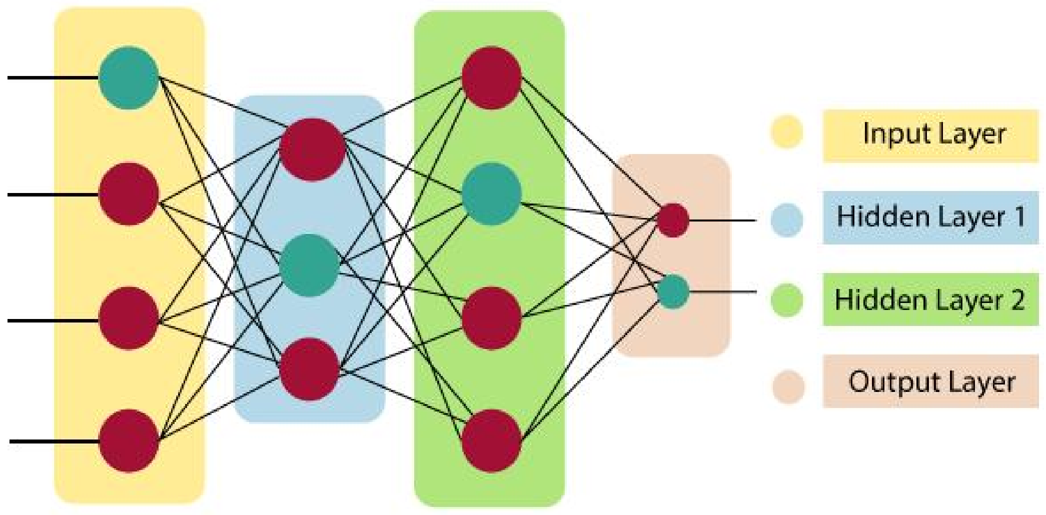
The given figure illustrates the typical diagram of Biological Neural Network.

The typical Artificial Neural Network looks something like the given figure.



Biological Neural Network	Artificial Neural Network
Dendrites	Inputs
Cell nucleus	Nodes
Synapse	Weights
Axon	Output

The architecture of an artificial neural network:



Input Layer:

As the name suggests, it accepts inputs in several different formats provided by the programmer.

Hidden Layer:

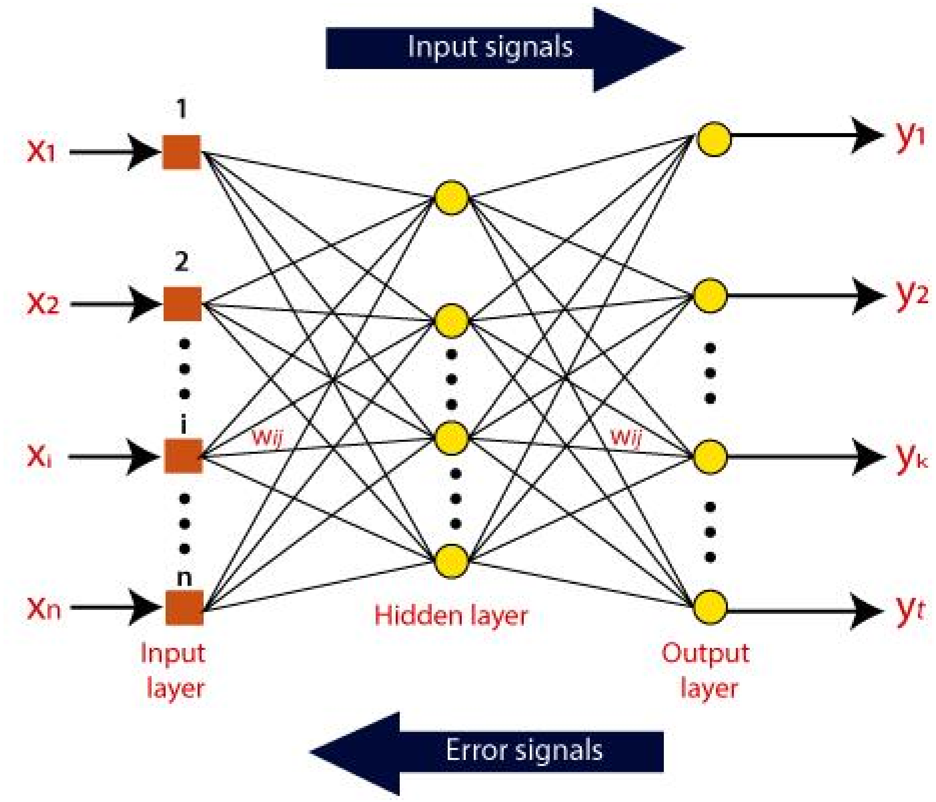
The hidden layer presents in-between input and output layers. It performs all the calculations to find hidden features and patterns.

Output Layer:

The input goes through a series of transformations using the hidden layer, which finally results in output that is conveyed using this layer.

How do artificial neural networks work?

Artificial Neural Network can be best represented as a weighted directed graph, where the artificial neurons form the nodes. The association between the neurons outputs and neuron inputs can be viewed as the directed edges with weights. The Artificial Neural Network receives the input signal from the external source in the form of a pattern and image in the form of a vector. These inputs are then mathematically assigned by the notations x(n) for every n number of inputs.



Afterward, each of the input is multiplied by its corresponding weights (these weights are the details utilized by the artificial neural networks to solve a specific problem). In general terms, these weights normally represent the strength of the interconnection between neurons inside the artificial neural network. All the weighted inputs are summarized inside the computing unit.

If the weighted sum is equal to zero, then bias is added to make the output non-zero or something else to scale up to the system's response. Bias has the same input, and weight equals to 1. Here the total of weighted inputs can be in the range of 0 to positive infinity. Here, to keep the response in the limits of the desired value, a certain maximum value is benchmarked, and the total of weighted inputs is passed through the activation function.

The activation function refers to the set of transfer functions used to achieve the desired output. There is a different kind of the activation function, but primarily either linear or non-linear sets of functions. Some of the commonly used sets of activation functions are the Binary, linear, and Tan hyperbolic sigmoidal activation functions.

CONCLUSION:

EXPT NO.3

TITLE:Implement Gradient Descent Algorithm to find the local minima of a function. For example, find the local minima of the function $y=(x+3)^2$ starting from the point x=2.

THEORY:

What is Gradient Descent

Gradient Descent is an iterative optimization algorithm that tries to find the optimum value (Minimum/Maximum) of an objective function. It is one of the most used optimization techniques in machine learning projects for updating the parameters of a model in order to minimize a cost function.

The main aim of gradient descent is to find the best parameters of a model which gives the highest accuracy on training as well as testing datasets. In gradient descent, The gradient is a vector that points in the direction of the steepest increase of the function at a specific point. Moving in the opposite direction of the gradient allows the algorithm to gradually descend towards lower values of the function, and eventually reaching to the minimum of the function.

Steps Required in Gradient Descent Algorithm

- **Step 1** we first initialize the parameters of the model randomly
- **Step 2** Compute the gradient of the cost function with respect to each parameter. It involves making partial differentiation of cost function with respect to the parameters.
- **Step 3** Update the parameters of the model by taking steps in the opposite direction of the model. Here we choose a hyperparameter learning rate which is denoted by alpha. It helps in deciding the step size of the gradient.
- **Step 4** Repeat steps 2 and 3 iteratively to get the best parameter for the defined model

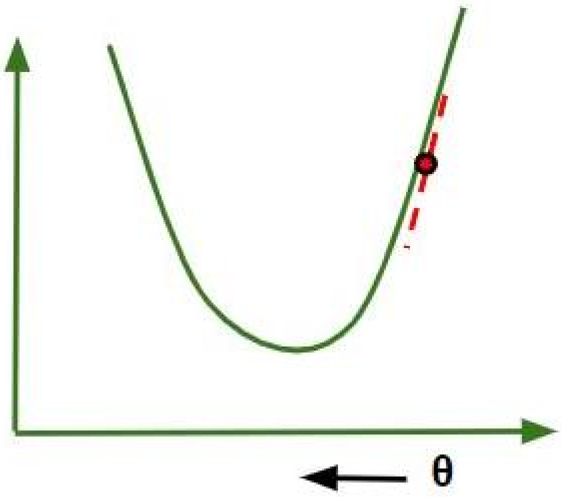
To apply this gradient descent on data using any programming language we have to make four new functions using which we can update our parameter and apply it to data to make a prediction. We will see each function one by one and understand it

1. **gradient_descent** – In the gradient descent function we will make the prediction on a dataset and compute the difference between the predicted and actual target value and accordingly we will update the parameter and hence it will return the updated parameter.
2. **compute_predictions** – In this function, we will compute the prediction using the parameters at each iteration.
3. **compute_gradient** – In this function we will compute the error which is the difference between the actual and predicted target value and then compute the gradient using this error and training data.
4. **update_parameters** – In this separate function we will update the parameter using learning rate and gradient that we got from the compute_gradient function.

How Does Gradient Descent Work

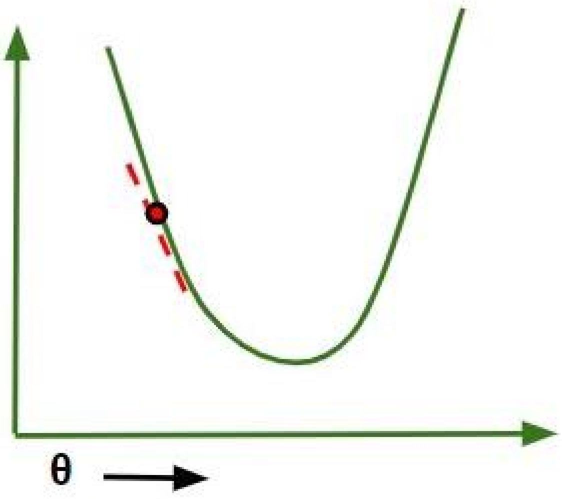
Gradient descent works by moving downward toward the pits or valleys in the graph to find the minimum value. This is achieved by taking the derivative of the cost function, as illustrated in the figure below. During each iteration, gradient descent step-downs the cost function in the direction of the steepest descent. By adjusting the parameters in this direction, it seeks to reach the minimum of the cost function and find the best-fit values for the parameters. The size of each step is determined by parameter **α** known as **Learning Rate**.
In the Gradient Descent algorithm, one can infer two points :

- **If slope is +ve** : θj = θj – (+ve value). Hence the value of θj decreases.



If slope is +ve in Gradient Descent

- **If slope is -ve** : θj = θj – (-ve value). Hence the value of θj increases.

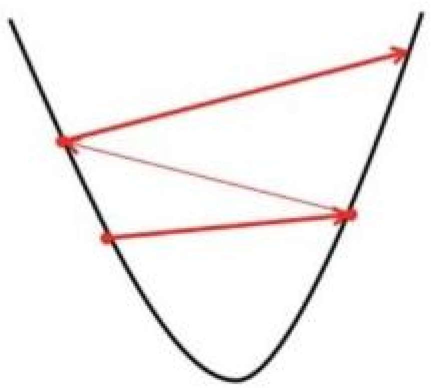


If slope is -ve in Gradient Descent

How To Choose Learning Rate

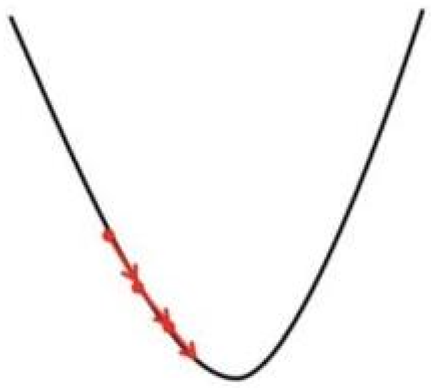
The choice of correct learning rate is very important as it ensures that Gradient Descent converges in a reasonable time. :

- If we choose **α to be very large**, Gradient Descent can overshoot the minimum. It may fail to converge or even diverge.



Effect of large alpha value on Gradient Descent

If we choose α to be very small, Gradient Descent will take small steps to reach local minima and will take a longer time to reach minima.



Effect of small alpha value on Gradient Descent

Advantages Of Gradient Descent

- **Flexibility:** Gradient Descent can be used with various cost functions and can handle non-linear regression problems.
- **Scalability:** Gradient Descent is scalable to large datasets since it updates the parameters for each training example one at a time.
- **Convergence:** Gradient Descent can converge to the global minimum of the cost function, provided that the learning rate is set appropriately.

Disadvantages Of Gradient Descent

- **Sensitivity to Learning Rate**: The choice of learning rate can be critical in Gradient Descent since using a high learning rate can cause the algorithm to overshoot the minimum, while a low learning rate can make the algorithm converge slowly.

- **Slow Convergence:** Gradient Descent may require more iterations to converge to the minimum since it updates the parameters for each training example one at a time.
- Local Minima**:** Gradient Descent can get stuck in local minima if the cost function has multiple local minima.
- **Noisy updates:** The updates in Gradient Descent are noisy and have a high variance, which can make the optimization process less stable and lead to oscillations around the minimum.

CONCLUSION:

EXPT NO.4

Implement K-Nearest Neighbors algorithm on diabetes.csv dataset. Compute confusionmatrix, accuracy, error rate, precision and recall on the given dataset.
THEORY:

Classification Metrics:
In a classification task, our main task is to predict the target variable which is in the form of discrete values. To evaluate the performance of such a model there are metrics as mentioned below:

- Classification Accuracy
- Logarithmic loss
- Area under Curve
- F1 score
    - Precision
    - Recall
- Confusion Matrix

Classification Accuracy

Classification accuracy is the accuracy we generally mean, whenever we use the term accuracy. We calculate this by calculating the ratio of correct predictions to the total number of input Samples.

$$\text{Accuracy} = \frac{\text{No. of correct predictions}}{\text{Total number of input samples}}$$

It works great if there are an equal number of samples for each class. For example, we have a 90% sample of *class A* and a 10% sample of *class B* in our training set. Then, our model will predict with an accuracy of 90% by predicting all the training samples belonging to *class A*. If we test the same model with a test set of 60% from class A and 40% from class B. Then the accuracy will fall, and we will get an accuracy of 60%.

Classification accuracy is good but it gives a False Positive sense of achieving high accuracy. The problem arises due to the possibility of misclassification of minor class samples being very high.

Logarithmic Loss
It is also known as Log loss. Its basic working propaganda is by penalizing the false (False Positive) classification. It usually works well with multi-class classification. Working on Log loss, the classifier should assign a probability for each and every class of all the samples. If there are N samples belonging to the *M* class, then we calculate the Log loss in this way:

$$LogarithmicLoss = \text{-}1 \frac{}{N \sum_{i=1}^{N} \sum_{j=1}^{M} y_{ij} * \log(p_{ij})}$$

Now the Terms,

- *yij* indicate whether sample *i* belongs to class j.
- *pij* – The probability of sample *i* belongs to class j.

- The range of log loss is [0,?). When the log loss is near 0 it indicates high accuracy and when away from zero then, it indicates lower accuracy.
- Let me give you a bonus point, minimizing log loss gives you higher accuracy for the classifier.

Area Under Curve(AUC)

It is one of the widely used metrics and basically used for binary classification. The AUC of a classifier is defined as the probability of a classifier will rank a randomly chosen positive example higher than a negative example. Before going into AUC more, let me make you comfortable with a few basic terms.

True positive rate:

Also called or termed sensitivity. True Positive Rate is considered as a portion of positive data points that are correctly considered as positive, with respect to all data points that are positive.

$$\mathrm{TPR} = \frac{\mathrm{TP}}{\mathrm{TP{+}FN}}$$

True Negative Rate

Also called or termed specificity. False Negative Rate is considered as a portion of negative data points that are correctly considered as negative, with respect to all data points that are negatives.

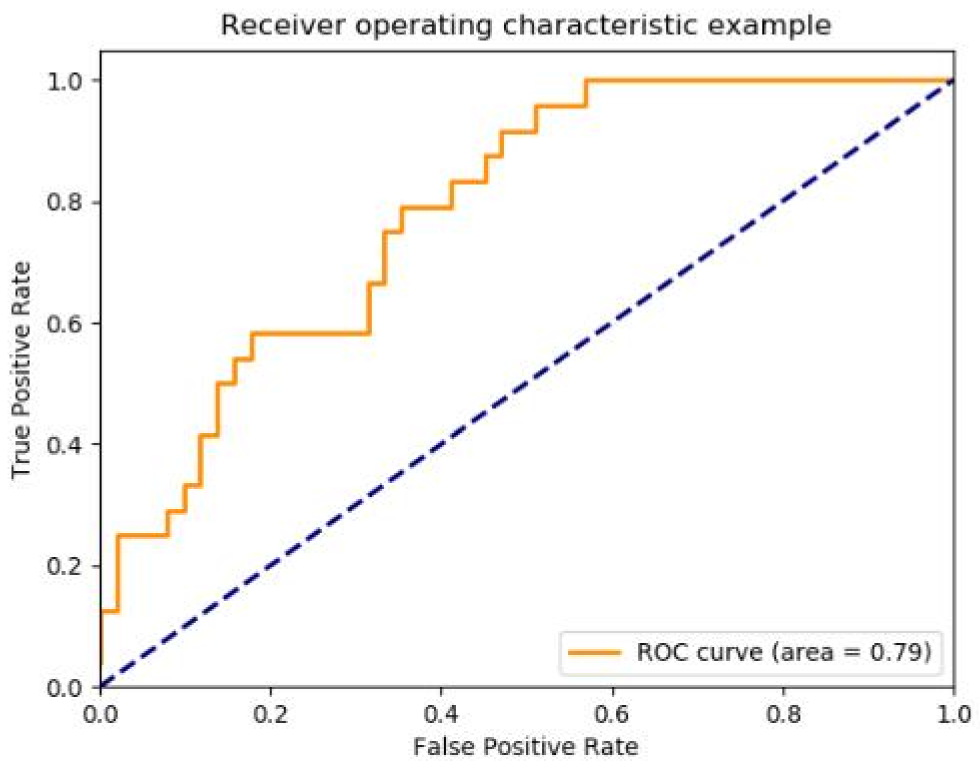
$$\mathrm{TNR} = \frac{\mathrm{TN}}{\mathrm{TN} + \mathrm{FP}}$$

False-positive Rate

False Negative Rate is considered as a portion of negative data points that are mistakenly considered as negative, with respect to all data points that are negative.

$$\mathrm{FPR} = \frac{\mathrm{FP}}{\mathrm{FP} + \mathrm{TN}}$$

False Positive Rate and True Positive Rate both have values in the range [0, 1]. Now the thing is what is A U C then? So, A U C is a curve plotted between False Positive Rate Vs True Positive Rate at all different data points with a range of [0, 1]. Greater the value of AUCC better the performance of the model.



ROC Curve for Evaluation of Classification Models

F1 Score

It is a harmonic mean between recall and precision. Its range is [0,1]. This metric usually tells us how precise (It correctly classifies how many instances) and robust (does not miss any significant number of instances) our classifier is.

Precision
There is another metric named Precision. Precision is a measure of a model's performance that tells you how many of the positive predictions made by the model are actually correct. It is calculated as the number of true positive predictions divided by the number of true positive and false positive predictions.

$$\text{Precision} = \frac{\text{TP}}{\text{TP} + \text{FP}}$$

Recall

$$\text{Recall} = \frac{\text{TP}}{\text{TP} + \text{FN}}$$

Lower recall and higher precision give you great accuracy but then it misses a large number of instances. The more the F1 score better will be performance. It can be expressed mathematically in this way:

$$F1 = 2 * \frac{1}{\frac{1}{\text{precision}} + \frac{1}{\text{recall}}}$$

Confusion Matrix

It creates a *N X N* matrix, where N is the number of classes or categories that are to be predicted. Here we have *N = 2*, so we get a *2 X 2* matrix. Suppose there is a problem with our practice which is a binary classification. Samples of that classification belong to either *Yes* or *No*. So, we build our classifier which will predict the class for the new input sample. After that, we tested our model with *165* samples, and we get the following result.

$\mathbf{n = 165}$	Predicted: NO	Predicted: YES
Actual: NO	50	10
Actual: YES	5	100

There are 4 terms you should keep in mind:

1. **True Positives:** It is the case where we predicted Yes and the real output was also yes.
2. **True Negatives:** It is the case where we predicted No and the real output was also No.
3. **False Positives:** It is the case where we predicted Yes but it was actually No.
4. **False Negatives:** It is the case where we predicted No but it was actually Yes.

The accuracy of the matrix is always calculated by taking average values present in the ***main diagonal i.e.***

$Accuracy = (TruePositive + TrueNegative)/TotalSampleAccuracy = (100 + 50)/165Accuracy = 0.91$

Regression Evaluation Metrics
In the regression task, we are supposed to predict the target variable which is in the form of continuous values. To evaluate the performance of such a model below mentioned evaluation metrics are used:

- Mean Absolute Error
- Mean Squared Error
- Root Mean Square Error
- Root Mean Square Logarithmic Error
- R2 – Score

Mean Absolute Error(MAE)

It is the average distance between Predicted and original values. Basically, it gives how we have predicted from the actual output. However, there is one limitation i.e.

it doesn't give any idea about the direction of the error which is whether we are under-predicting or over-predicting our data. It can be represented mathematically in this way:

$$\mathrm{MAE} = \frac{1}{\mathrm{N}} \sum_{\mathrm{j}=1}^{\mathrm{N}} \left|\mathrm{y_j} - \hat{\mathrm{y}}_\mathrm{j}\right|$$

Mean Squared Error(MSE)

It is similar to mean absolute error but the difference is it takes the square of the average of between predicted and original values. The main advantage to take this metric is here, it is easier to calculate the gradient whereas, in the case of mean absolute error, it takes complicated programming tools to calculate the gradient. By taking the square of errors it pronounces larger errors more than smaller errors, we can focus more on larger errors. It can be expressed mathematically in this way.

$$\mathrm{MSE} = \frac{1}{\mathrm{N}} \sum_{\mathrm{j}=1}^{\mathrm{N}} \left(\mathrm{y_j} - \hat{\mathrm{y}}_\mathrm{j}\right)^2$$

Root Mean Square Error(RMSE)

We can say that RMSE is a metric that can be obtained by just taking the square root of the MSE value. As we know that the MSE metrics are not robust to outliers and so are the RMSE values. This gives higher weightage to the large errors in predictions.

$$\mathrm{RMSE} = \sqrt{\frac{\sum_{\mathrm{j}=1}^{\mathrm{N}} \left(\mathrm{y_j} - \hat{\mathrm{y}}_\mathrm{j}\right)^2}{\mathrm{N}}}$$

Root Mean Squared Logarithmic Error(RMSLE)

There are times when the target variable varies in a wide range of values. And hence we do not want to penalize the overestimation of the target values but penalize the underestimation of the target values. For such cases, RMSLE is used as an evaluation metric which helps us to achieve the above objective.

Some changes in the original formula of the RMSE code will give us the RMSLE formula that is as shown below:

$$\mathrm{RMSLE} = \sqrt{\frac{\sum_{\mathrm{j}=1}^{\mathrm{N}} \left(\log(\hat{\mathrm{y}}_\mathrm{j}+1) - \log(\mathrm{y_j}+1)\right)^2}{\mathrm{N}}}$$

R2 – Score

The coefficient of determination also called the R2 score is used to evaluate the performance of a linear regression model. It is the amount of variation in the output-dependent attribute which is predictable from the input independent variable(s). It is used to check how well-observed results are reproduced by the model, depending on the ratio of total deviation of results described by the model.

CONCLUSION:

EXPT NO.5

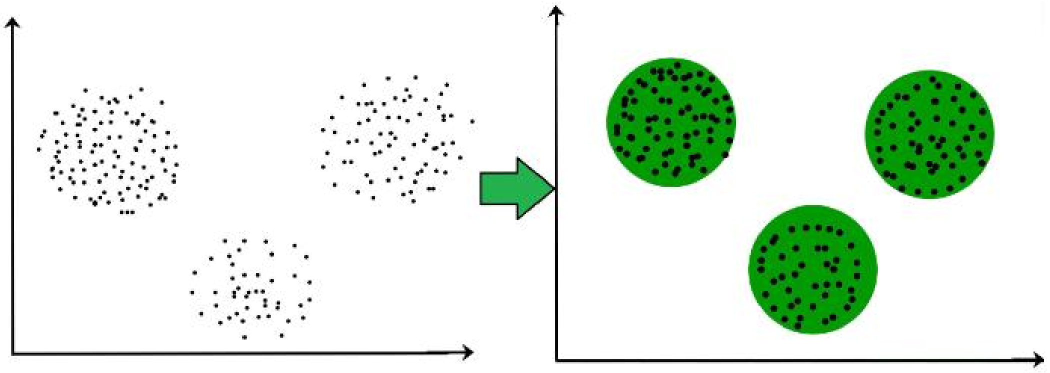
Implement K-Means clustering/ hierarchical clustering on sales_data_sample.csv dataset.Determine the number of clusters using the elbow method.

THEORY:

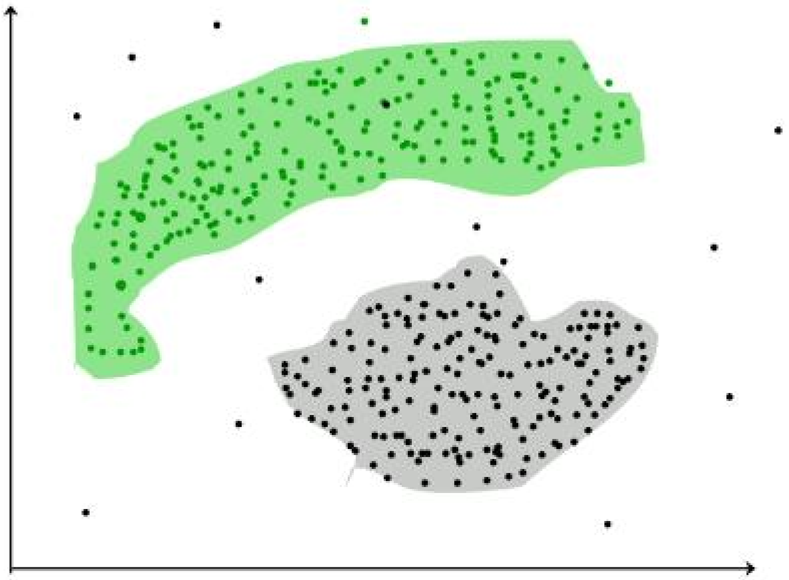
Introduction to Clustering: It is basically a type of *unsupervised learning method*. An unsupervised learning method is a method in which we draw references from datasets consisting of input data without labeled responses. Generally, it is used as a process to find meaningful structure, explanatory underlying processes, generative features, and groupings inherent in a set of examples.

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 and dissimilar to the data points in other groups. It is basically a collection of objects on the basis of similarity and dissimilarity between them.

For example The data points in the graph below clustered together can be classified into one single group. We can distinguish the clusters, and we can identify that there are 3 clusters in the below picture.



It is not necessary for clusters to be spherical as depicted below:



DBSCAN: Density-based Spatial Clustering of Applications with Noise

These data points are clustered by using the basic concept that the data point lies within the given constraint from the cluster center. Various distance methods and techniques are used for the calculation of the outliers.

Why Clustering?

Clustering is very much important as it determines the intrinsic grouping among the unlabelled data present. There are no criteria for good clustering. It depends on the user, and what criteria they may use which satisfy their need. For instance, we could be interested in finding representatives for homogeneous groups (data reduction), finding "natural clusters" and describing their unknown properties ("natural" data types), in finding useful and suitable groupings ("useful" data classes) or in finding unusual data objects (outlier detection). This algorithm must make some assumptions that constitute the similarity of points and each assumption make different and equally valid clusters.

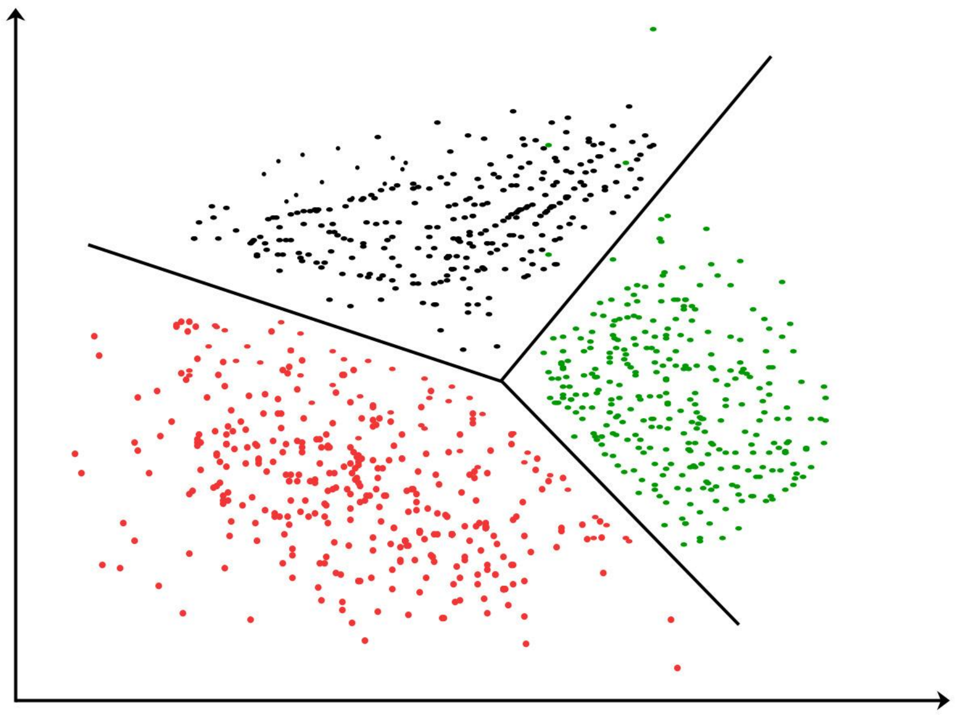
Clustering Methods:

- **Density-Based Methods:** These methods consider the clusters as the dense region having some similarities and differences from the lower dense region of the space. These methods have good accuracy and the ability to merge two clusters. Example *DBSCAN (Density-Based Spatial Clustering of Applications with Noise)*, *OPTICS (Ordering Points to Identify Clustering Structure)*, etc.
- **Hierarchical Based Methods:** The clusters formed in this method form a tree-type structure based on the hierarchy. New clusters are formed using the previously formed one. It is divided into two category
- **Agglomerative** (bottom-up *approach*)
- **Divisive** (top-down *approach*)

Examples *CURE (Clustering Using Representatives), BIRCH (Balanced Iterative Reducing Clustering and using Hierarchies)*, 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 (Clustering Large Applications based upon Randomized Search)*, etc.
- **Grid-based Methods:** In this method, the data space is formulated into 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 objects example *STING (Statistical Information Grid), wave cluster, CLIQUE (CLustering In Quest)*, etc.

Clustering Algorithms: K-means clustering algorithm – It is the simplest unsupervised learning algorithm that solves clustering problem.K-means algorithm partitions n observations into k clusters where each observation belongs to the cluster with the nearest mean serving as a prototype of the cluster.



Applications of Clustering in different fields:

1. **Marketing:** It can be used to characterize & discover customer segments for marketing purposes.
2. **Biology:** It can be used for classification among different species of plants and animals.
3. **Libraries:** It is used in clustering different books on the basis of topics and information.
4. **Insurance:** It is used to acknowledge the customers, their policies and identifying the frauds.
5. **City Planning:** It is used to make groups of houses and to study their values based on their geographical locations and other factors present.
6. **Earthquake studies:** By learning the earthquake-affected areas we can determine the dangerous zones.
7. **Image Processing**: Clustering can be used to group similar images together, classify images based on content, and identify patterns in image data.
8. **Genetics:** Clustering is used to group genes that have similar expression patterns and identify gene networks that work together in biological processes.
9. **Finance:** Clustering is used to identify market segments based on customer behavior, identify patterns in stock market data, and analyze risk in investment portfolios.
10. **Customer Service:** Clustering is used to group customer inquiries and complaints into categories, identify common issues, and develop targeted solutions.
11. **Manufacturing**: Clustering is used to group similar products together, optimize production processes, and identify defects in manufacturing processes.
12. **Medical diagnosis:** Clustering is used to group patients with similar symptoms or diseases, which helps in making accurate diagnoses and identifying effective treatments.

13. **Fraud detection:** Clustering is used to identify suspicious patterns or anomalies in financial transactions, which can help in detecting fraud or other financial crimes.
14. **Traffic analysis:** Clustering is used to group similar patterns of traffic data, such as peak hours, routes, and speeds, which can help in improving transportation planning and infrastructure.
15. **Social network analysis:** Clustering is used to identify communities or groups within social networks, which can help in understanding social behavior, influence, and trends.
16. **Cybersecurity:** Clustering is used to group similar patterns of network traffic or system behavior, which can help in detecting and preventing cyberattacks.
17. **Climate analysis:** Clustering is used to group similar patterns of climate data, such as temperature, precipitation, and wind, which can help in understanding climate change and its impact on the environment.
18. **Sports analysis:** Clustering is used to group similar patterns of player or team performance data, which can help in analyzing player or team strengths and weaknesses and making strategic decisions.
19. **Crime analysis:** Clustering is used to group similar patterns of crime data, such as location, time, and type, which can help in identifying crime hotspots, predicting future crime trends, and improving crime prevention strategies.

CONCLUSION:

410246: Laboratory Practice III
Group B: Machine Learning

I) Classify the email using the binary classification method. Email Spam detection has two states: a) Normal State – Not Spam, b) Abnormal State – Spam. Use K-Nearest Neighbors and Support Vector Machine for classification. Analyze their performance.
Datasetlink: The emails.csv dataset on the Kaggle
https://www.kaggle.com/datasets/balaka18/email-spam-classification-dataset-csv

LINK FOR PRACTICAL IMPLEMENTATION:
https://www.youtube.com/watch?v=vWblvsF-XF8

II)Given a bank customer, build a neural network-based classifier that can determine whetherthey will 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.
Link to the Kaggle project: https://www.kaggle.com/barelydedicated/bank-customer-churn-modeling 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 of improvement and implement the same.

1. Print the accuracy score and confusion matrix (5 points).

LINK FOR PRACTICAL IMPLEMENTATION:

https://www.youtube.com/watch?v=Wjt_WHG7Q70

III)Implement Gradient Descent Algorithm to find the local minima of a function.
For example, find the local minima of the function $y=(x+3)^2$ starting from the point x=2.

LINK FOR PRACTICAL IMPLEMENTATION:

https://www.youtube.com/watch?v=bo7qg7O5jCk

IV)Implement K-Nearest Neighbors algorithm on diabetes.csv dataset. Compute confusionmatrix, accuracy, error rate, precision and recall on the given dataset.

Dataset link : https://www.kaggle.com/datasets/abdallamahgoub/diabetes

LINK FOR PRACTICAL IMPLEMENTATION:

https://www.youtube.com/watch?v=8mCM2JHSKIs

V)Implement K-Means clustering/ hierarchical clustering on sales_data_sample.csv dataset.Determine the number of clusters using the elbow method.

Dataset link : https://www.kaggle.com/datasets/kyanyoga/sample-sales-data

LINK FOR PRACTICAL IMPLEMENTATION:

https://www.youtube.com/watch?v=8mCM2JHSKIs

https://dsrajnor.wordpress.com/lp-iii/