A
LAB MANUAL REPORT
ON

Machine Learning

Prepared By

Dr. T.Bhaskar

Associate Professor

Google-Site: https://sites.google.com/view/bhaskart/ug-notes/machine-learning

Moodle-Site: https://proftbhaskar.gnomio.com/course/view.php?id=3 (Access as a Guest)

Machine Learning Virtual Lab: https://tinyurl.com/MLLabDrBhaskarT

(Savitribai Phule Pune University, Pune)

Department of Computer Engineering

Sanjivani College of Engineering (Autonomous)

Kopargaon - 423 603

SANJIVANI COLLEGE OF ENGINEERING

(An Autonomous Institute)

DEPARTMENT OF COMPUTER ENGINEERING

Machine Learning Lab Manual

Course Teacher: Dr.T.Bhaskar

Machine Learning Lab

Course Objectives:

1. To understand the need for machine learning for various problem solving
2. To understand nature of the problem and apply machine learning algorithm.
3. To study the various supervised, semi-supervised and unsupervised learning algorithms in machine learning
4. To understand the latest trends in machine learning
5. To design appropriate machine learning algorithms for problem solving.

Course Outcomes (COs):

On completion of the course, student will be able to–

CO No.	Statement of Course Outcome	Bloom's Taxonomy Level	
CO1	Apply the Regression Techniques to various problems	3	CO1
CO2	Apply pre-processing methods & Feature Engineering to prepare training data set for machine learning	3	CO2
CO3	Apply the Bayesian algorithm to various problems	3	CO3
CO4	Apply the classification Techniques to various problems	3	CO4
CO5	Apply the ensemble techniques for Data	3	CO5
CO6	Ability to apply Clustering techniques for data.	3	CO6

Tools for Machine Learning Laboratory Practice

Operating System recommended: - 64-bit Open source Linux or its derivative
Programming Languages: PYTHON/R
Programmingtools recommended:Anaconda or Miniconda Frameworks
Online Tools:Google Colab

Guidelines for Student Journal

The laboratory assignments are to be submitted by student in the form of journal. Journal may consists of prologue, Certificate, table of contents, and **handwritten write-up** of each assignment (Title, Objectives, Problem Statement, Outcomes, software and Hardware requirements, Date of Completion, Assessment grade/marks and assessor's sign, Theory- Concept in brief, Algorithm/Database design, test cases, conclusion/analysis). **Program codes with sample output of all performed assignments are to be submitted as softcopy.**

Guidelines for Assessment

Continuous assessment of Machine Learning laboratory work is to be done based on overall performance and lab assignments performance of student. Each lab assignment assessment will assign grade/marks based on parameters with appropriate weightage. Suggested parameters for overall assessment as well as each lab assignment assessment include- timely completion, performance, innovation, efficient codes,punctuality and neatness documentation.

Rubrics for Assessment of Machine Learning Lab

Evaluation of practical assignment is based on the following criteria's. Each Assignment is evaluated out of 10 Marks

Criteria	Excellent	Good	Average	Poor
Write Ups (2)	Timely submission within deadline in all respects. (2)	Timely submission but needs some improvement. (1)	Submission with maximum one-week delay. (1)	Delayed in submission or found copied. (0)
Understanding (4)	Understand all the concepts, algorithm, or logic. (4)	Understand the concepts, algorithm, or logic but need improvement. (3)	Limited understanding of the concepts or algorithm or logic but need more improvement (2)	Failed to understand the concepts, algorithm, or logic. (0)
Performance (4)	Implemented the concepts, algorithm, or logic with correct expected output considering test cases. (4)	Implemented the concepts, algorithm, or logic with expected results. (3)	Implemented the concepts, algorithm, or logic with partial results and needs improvement. (2-1)	Not implemented and no output. (0)

SANJIVANI COLLEGE OF ENGINEERING

(An Autonomous Institute)

DEPARTMENT OF COMPUTER ENGINEERING

Machine Learning Lab Manual

Course Teacher: Dr.T.Bhaskar

List of Machine Learning Laboratory Assignments

1. **Assignment on Linear Regression:**

The following table shows the results of a recently conducted study on the correlation of the number of hours spent driving with the risk of developing acute backache. Find the equation of the best fit line for this data.

Number of hours spent driving (x)	Risk score on a scale of 0-100 (y)
10	95
9	80
2	10
15	50
10	45
16	98
11	38
16	93

2. Apply the **Principal Component Analysis (PCA)** for Feature Reduction Techniques on any dataset. (For ex: IRIS Dataset.)

(Dataset Downloads Link): https://drive.google.com/file/d/12BY34aCbYLoLjy3gDUMrZEBUf7l5FZsd/view?usp=share_link

3. Assignment on **Decision Tree Classifier**:

A dataset collected in a Cloth shop showing details of customers and whether or not they responded to a special offer to buy a new Sarry is shown in table below. Use this dataset to build a decision tree, with Buys as the target variable, to help in buying lip-sticks in the future. Find the root node of the decision tree. According to the decision tree you have made from the previous training data set, what is the decision for the test data: [Age < 21, Income = Low, Gender = Female, Marital Status = Married]?

ID	Age	Income	Gender	Marital Status	Buys
1	< 21	High	Male	Single	No
2	< 21	High	Male	Married	No
3	21-35	High	Male	Single	Yes
4	>35	Medium	Male	Single	Yes
5	>35	Low	Female	Single	Yes
6	>35	Low	Female	Married	No
7	21-35	Low	Female	Married	Yes
8	< 21	Medium	Male	Single	No
9	<21	Low	Female	Married	Yes
10	> 35	Medium	Female	Single	Yes
11	< 21	Medium	Female	Married	Yes
12	21-35	Medium	Male	Married	Yes
13	21-35	High	Female	Single	Yes
14	> 35	Medium	Male	Married	No

4. Implement **Naive Bayes Classification Algorithm** on any suitable dataset.

(Dataset Downloads Link): https://drive.google.com/file/d/12BY34aCbYLoLjy3gDUMrZEBUf7l5FZsd/view?usp=share_link

5. Assignment on **SVM Classification**:

Implement SVM Classification Technique on any dataset.
(For ex: Breast Cancer Dataset.) (get Dataset from sklearn.datasets.load_breast_cancer())

6. **Assignment on K-Means Clustering**:

We have given a collection of 8 points. P1=[0.1,0.6] P2=[0.15,0.71] P3=[0.08,0.9] P4=[0.16, 0.85] P5=[0.2,0.3] P6=[0.25,0.5] P7=[0.24,0.1] P8=[0.3,0.2]. Perform the k-mean clustering with initial centroids as m1=P1 =Cluster#1=C1 and m2=P8=cluster#2=C2.
Answer thefollowing

1] Which cluster does P6 belongs to?
2] What is the population of cluster around m2?
3] What is updated value of m1 and m2?

7. Implement **Gradient Boost Classifier Model** on Income Evaluation Data set.
(Dataset Downloads Link) https://drive.google.com/file/d/1zI-X3zdiuM9u74zQyKIShvAUtPJQ7jUK/view?usp=sharing

Title: ML Lab: Assignment on Simple Linear Regression

Aim:

Implement Simple Linear Regression on given Problem

Problem Definition/Objective:

The following table shows the results of a recently conducted study on the correlation of the number of hours spent driving with the risk of developing acute backache. Find the equation of the best fit line for this data.

Number of hours spent driving (x)	Risk score on a scale of 0-100 (y)
10	95
9	80
2	10
15	50
10	45
16	98
11	38
16	93

Input:CSV Dataset

Answer-

y = 4.59x + 12.58

Hints: For each x calculate the value of y using the given equations. Then calculate error for each equation. Equation with lowest error is the desired answer. For error calculation

Given (x1,y1),(x2,y2),…,(xn,yn), best fitting data to y = f(x) by least squares requires minimization of

$\sum_{i=1}^{n}[y_i - f(x_i)]^2$

Outcomes:

After completion of this assignment students are able to understand the How to find the correlation between to Two variable, How to Calculate Accuracy of the Linear Model and how to plot graph using **matplotlib.**

Theory:

Linear Regression

Regression analysis is used in stats to find trends in data. For example, you might guess that there's a connection between how much you eat and how much you weight; regression analysis can help you quantify that.

In a cause and effect relationship, the **independent variable** is the cause, and the **dependent variable** is the effect. **Least squares linear regression** is a method for predicting the value of a dependent variable *Y*, based on the value of an independent variable *X*.

Prerequisites for Regression:

Simple linear regression is appropriate when the following conditions are satisfied.

- The dependent variable *Y* has a linear relationship to the independent variable *X*. To check this, make sure that the XY scatterplot is linear and that the residual plotshows a random pattern. For each value of X, the probability distribution of Y has the same standard deviation σ.
- When this condition is satisfied, the variability of the residuals will be relatively constant across all values of X, which is easily checked in a residual plot.
- For any given value of X,
    - The Y values are independent, as indicated by a random pattern on the residual plot.
    - The Y values are roughly normally distributed (i.e., symmetric and unimodal). A little skewness is ok if the sample size is large. A histogramor a dotplot will show the shape of the distribution.

The Least Squares Regression Line:

Linear regression finds the straight line, called the **least squares regression line** or LSRL, that best represents observations in a bivariate data set. Suppose *Y* is a dependent variable, and *X* is an independent variable. The population regression line is:

$$Y = B_0 + B_1X$$

Where B_0 is a constant, B_1 is the regression coefficient, X is the value of the independent variable, and Y is the value of the dependent variable.
Given a random sample of observations, the population regression line is estimated by:

$$\hat{y} = b_0 + b_1 x$$

Where b_0 is a constant, b_1 is the regression coefficient, x is the value of the independent variable, and $\hat{y}$ is the *predicted* value of the dependent variable.

How to Define a Regression Line:

Normally, you will use a computational tool - a software package (e.g., Excel) or a graphing calculator- to find b_0 and b_1. You enter the *X* and *Y* values into your program or calculator, and the tool solves for each parameter. In the unlikely event that you find yourself on a desert island without a computer or a graphing calculator, you can solve for b_0 and b_1 "by hand". Here are the equations.

$$B_1 = \Sigma [(x_i - x)(y_i - y)] / \Sigma [(x_i - x)^2]$$

$$b_1 = r * (s_y / s_x)$$

$$b_0 = y - b_1 * x$$

where b_0 is the constant in the regression equation, b_1 is the regression coefficient, r is the correlation between x and y, x_i is the *X* value of observation *i*, y_i is the *Y* value of observation *i*, x is the mean of *X*, y is the mean of *Y*, s_x is the standard deviation of *X*, and s_y is the standard deviation of *Y*.

Coefficient of determination. The coefficient of determination (R^2) for a linear regression model with one independent variable is:

$$R^2 = \{ (1 / N) * \Sigma [(x_i - x) * (y_i - y)] / (\sigma_x * \sigma_y) \}^2$$

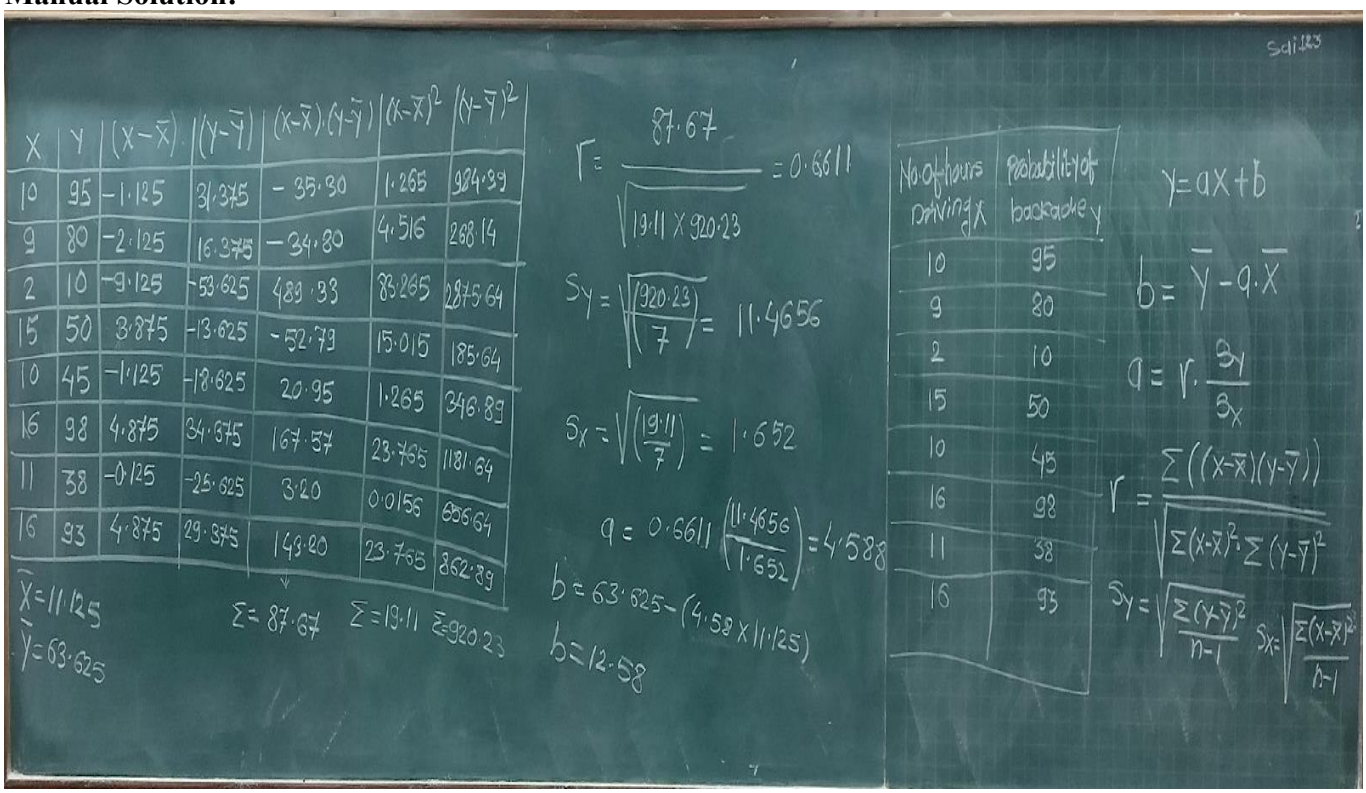
where N is the number of observations used to fit the model, Σ is the summation symbol, x_i is the x value for observation i, x is the mean x value, y_i is the y value for observation i, y is the mean y value, σ_x is the standard deviation of x, and σ_y is the standard deviation of y.

If you know the linear correlation (r) between two variables, then the coefficient of determination (R^2) is easily computed using the following formula: $R^2 = r^2$.

Standard Error

The **standard error** about the regression line (often denoted by SE) is a measure of the average amount that the regression equation over- or under-predicts. The higher the coefficient of determination, the lower the standard error; and the more accurate predictions are likely to be.

Manual Solution:



Example:

Last year, five randomly selected students took a math aptitude test before they began their statistics course. The Statistics Department has three questions.

- What linear regression equation best predicts statistics performance, based on math aptitude scores?
- If a student made an 80 on the aptitude test, what grade would we expect her to make in statistics?
- How well does the regression equation fit the data?

How to Find the Regression Equation?

In the table below ,the xi column shows scores on the aptitude test. Similarly, the y_i column shows statistics grades. The last two columns show deviations scores - the difference between the student's score and the average score on each test. The last two rows show sums and mean scores that we will use to conduct the regression analysis.

Student	x_i	y_i	(x_i-x)	(y_i-y)
1	95	85	17	8
2	85	95	7	18
3	80	70	2	-7
4	70	65	-8	-12
5	60	70	-18	-7
Sum	390	385		
Mean	78	77		

And for each student, we also need to compute the squares of the deviation scores (the last two columns in the table below).

Student	x_i	y_i	$(x_i-x)^2$	$(y_i-y)^2$
1	95	85	289	64
2	85	95	49	324

3	80	70	4	49
4	70	65	64	144
5	60	70	324	49
Sum	390	385	730	630
Mean	78	77		

And finally, for each student, we need to compute the product of the deviation scores.

Student	x_i	y_i	$(x_i-x)(y_i-y)$
1	95	85	136
2	85	95	126
3	80	70	-14
4	70	65	96
5	60	70	126
Sum	390	385	470
Mean	78	77	

The regression equation is a linear equation of the form: $\hat{y} = b_0 + b_1x$. To conduct a regression analysis, we need to solve for b_0 and b_1. Computations are shown below. Notice that all of our inputs for the regression analysis come from the above three tables.

First, we solve for the regression coefficient (b_1):

$$b_1 = \Sigma [(x_i - x)(y_i - y)] / \Sigma [(x_i - x)^2]$$
$$b_1 = 470/730$$
$$b_1 = 0.644$$

Once we know the value of the regression coefficient (b_1), we can solve for the regression slope (b_0):

$$b_0 = y - b_1 * x$$
$$b_0 = 77 - (0.644)(78)$$
$$b_0 = 26.768$$

Therefore, the regression equation is: $\hat{y} = 26.768 + 0.644x$

How to Use the Regression Equation:

Once you have the regression equation, using it is a snap. Choose a value for the independent

variable (x), perform the computation, and you have an estimated value ($\hat{y}$) for the dependent variable. In our example, the independent variable is the student's score on the aptitude test. The dependent variable is the student's statistics grade. If a student made an 80 on the aptitude test, the estimated statistics grade ($\hat{y}$) would be:

$$\hat{y} = b_0 + b_1 x$$
$$\hat{y} = 26.768 + 0.644x = 26.768 + 0.644 * 80$$
$$\hat{y} = 26.768 + 51.52 = 78.288$$

Algorithm:

1.**Import the Required Packages**

2.**Read Given Dataset**

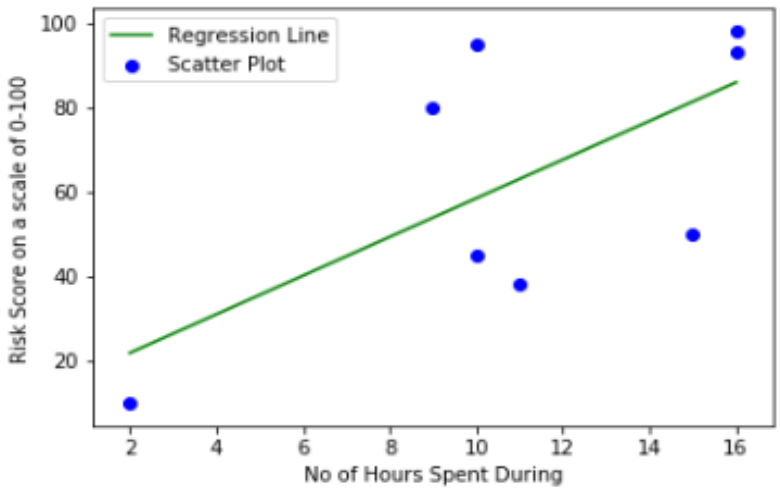
3.Import the Linear Regression and Create object of it

4.**Find the Accuracy of Model using Score Function**

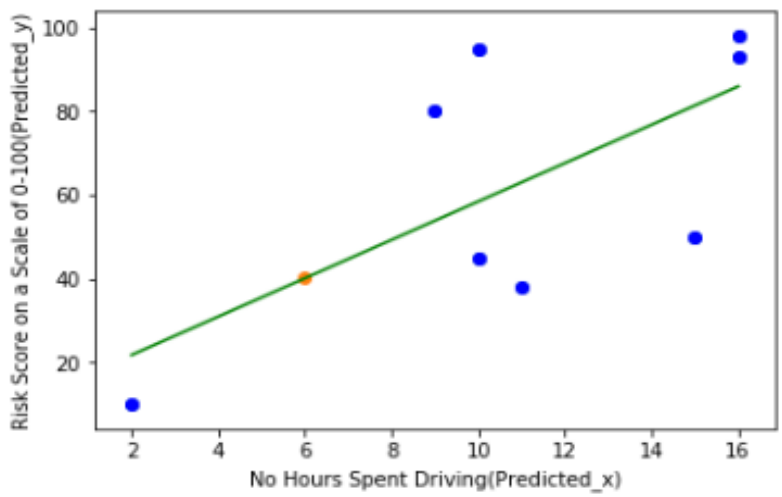
5.Predict the value using Regressor Object
6.Take input from User
7.Calculate the value of y
8.Draw Scatter PLot

Output:

```
(8, 2)
   No of Hours Spent During(X)  Risk Score on a scale of 0-100(Y)
0                           10                                 95
1                            9                                 80
2                            2                                 10
3                           15                                 50
4                           10                                 45
Slope,Intercept: 4.58789860997547 12.584627964022893
```




```
Root Mean Squares Error: 22.759716640449565
Accuracy: 43.709481451010035
Enter No Hours Spent in Driving:6
```



Linear Regression Applications:

1. **Trend lines:**A trend line represents the variation in some quantitative data with passage of time (like GDP, oil prices, etc.). These trends usually follow a linear relationship. Hence, linear regression can be applied to predict future values. However, this method suffers from a lack of scientific validity in cases where other potential changes can affect the data.
2. **Economics:**Linear regression is the predominant empirical tool in economics. For example, it is used to predict consumption spending, fixed investment spending, inventory investment, purchases of a country's exports, spending on imports, the demand to hold liquid assets, labor demand, and labor supply.
3. **Finance:**Capital price asset model uses linear regression to analyze and quantify the systematic risks of an investment.
4. **Biology**:Linear regression is used to model causal relationships between parameters in biological systems.

Conclusion:

Thus student can learn that to how to find the trend of data using X as Independent Variable and Y is and Dependent Variable by using Linear Regression.

Title: Assignment on PCA (Principal Component Analysis)

Aim:

Apply the Principal Component Analysis for feature reduction on IRIS Dataset

Prerequisite:

Basic of Python, Data Mining Algorithm, Concept of Principal component analysis (PCA)

Theory:

Principal component analysis (PCA):

Principal Component Analysis is an unsupervised learning algorithm that is used for dimensionality reduction in machine learning. It is a statistical process that converts the observations of correlated features into a set of linearly uncorrelated features with the help of orthogonal transformation. These new transformed features are called the Principal Components. It is one of the popular tools that is used for exploratory data analysis and predictive modeling. It is a technique to draw strong patterns from the given dataset by reducing the variances.

PCA generally tries to find the lower-dimensional surface to project the high-dimensional data.

PCA works by considering the variance of each attribute because the high attribute shows the good split between the classes, and hence it reduces the dimensionality. Some real-world applications of PCA are image processing, movie recommendation systems, and optimizing the power allocation in various communication channels. It is a feature extraction technique, so it contains the important variables and drops the least important variable.

The PCA algorithm is based on some mathematical concepts such as:

- Variance and Covariance
- Eigenvalues and Eigen factors

Some common terms used in PCA algorithm:

- **Dimensionality:** It is the number of features or variables present in the given dataset. More easily, it is the number of columns present in the dataset.
- **Correlation:** It signifies how strongly two variables are related to each other. Such as if one changes, the other variable also gets changed. The correlation value ranges from -1 to +1. Here, -1 occurs if variables are inversely proportional to each other, and +1 indicates that variables are directly proportional to each other.
- **Orthogonal:** It defines that variables are not correlated to each other, and hence the correlation between the pair of variables is zero.
- **Eigenvectors:** If there is a square matrix M, and a nonzero vector v is given. Then v will be an eigenvector if Av is the scalar multiple of v.
- **Covariance Matrix:** A matrix containing the covariance between the pair of variables is called the Covariance Matrix.

Principal Components in PCA:

As described above, the transformed new features or the output of PCA are the Principal Components. The number of these PCs are either equal to or less than the original features present in the dataset. Some properties of these principal components are given below:

- The principal component must be the linear combination of the original features.
- These components are orthogonal, i.e., the correlation between a pair of variables is zero.
- The importance of each component decreases when going from 1 to n, it means the 1 PC has the most importance, and n PC will have the least importance.

Steps for PCA algorithm

1. **Getting the dataset**
   Firstly, we need to take the input dataset and divide it into two subparts X and Y, where X is the training set, and Y is the validation set.

2. **Representing data into a structure**
   Now we will represent our dataset into a structure. Such as we will represent the two-dimensional matrix of independent variable X. Here each row corresponds to the data items, and the column corresponds to the Features. The number of columns is the dimensions of the dataset.
3. **Standardizing the data**
   In this step, we will standardize our dataset. Such as in a particular column, the features with high variance are more important compared to the features with lower variance.
   If the importance of features is independent of the variance of the feature, then we will divide each data item in a column with the standard deviation of the column. Here we will name the matrix as Z.
4. **Calculating the Covariance of Z**
   To calculate the covariance of Z, we will take the matrix Z, and will transpose it. After transposing, we will multiply it by Z. The output matrix will be the Covariance matrix of Z.
5. **Calculating the EigenValues and EigenVectors**
   Now we need to calculate the eigenvalues and eigenvectors for the resultant covariance matrix Z. Eigenvectors or the covariance matrix are the directions of the axes with high information. And the coefficients of these eigenvectors are defined as the eigenvalues.
6. **Sorting the EigenVectors**
   In this step, we will take all the eigenvalues and will sort them in decreasing order, which means from largest to smallest. And simultaneously sort the eigenvectors accordingly in matrix P of eigenvalues. The resultant matrix will be named as P*.
7. **Calculating the new features Or Principal Components**
   Here we will calculate the new features. To do this, we will multiply the P* matrix to the Z. In the resultant matrix Z*, each observation is the linear combination of original features. Each column of the Z* matrices are independent of each other.
8. **Remove less or unimportant features from the new dataset.**
   The new feature set has occurred, so we will decide here what to keep and what to remove. It means, we will only keep the relevant or important features in the new dataset, and unimportant features will be removed.

Applications of Principal Component Analysis:

- PCA is mainly used as the dimensionality reduction technique in various AI applications such **as computer vision, image compression, etc.**
- It can also be used for finding hidden patterns if data has high dimensions. Some fields where PCA is used are Finance, data mining, Psychology, etc.

Step by Step PCA with Iris dataset

How does PCA work -

- Calculate the covariance matrix X of data points.
- Calculate eigenvectors and corresponding eigenvalues.
- Sort the eigenvectors according to their eigenvalues in decreasing order.
- Choose first k eigenvectors and that will be the new k dimensions.
- Transform the original n dimensional data points into k dimensions.

Implementing PCA on a 2-D Dataset

Step 1: Normalize the data

First step is to normalize the data that we have so that PCA works properly. This is done by subtracting the respective means from the numbers in the respective column. So if we have two dimensions X and Y, all X become $\boldsymbol{x}$- and all Y become $\boldsymbol{y}$-. This produces a dataset whose mean is zero.

Step 2: Calculate the covariance matrix

Since the dataset we took is 2-dimensional, this will result in a 2x2 Covariance matrix.

$$Matrix(Covariance) = \begin{bmatrix} Var[X_1] & Cov[X_1, X_2] \\ Cov[X_2, X_1] & Var[X_2] \end{bmatrix}$$

Please note that Var[X1] = Cov[X1,X1] and Var[X2] = Cov[X2,X2].

Step 3: Calculate the eigenvalues and eigenvectors

Next step is to calculate the eigenvalues and eigenvectors for the covariance matrix. The same is possible because it is a square matrix. ƛ is an eigenvalue for a matrix A if it is a solution of the characteristic equation:

$$\det(\lambda I - A) = 0$$

Where, I is the identity matrix of the same dimension as A which is a required condition for the matrix subtraction as well in this case and 'det' is the determinant of the matrix. For each eigenvalue ƛ, a corresponding eigen-vector v, can be found by solving:

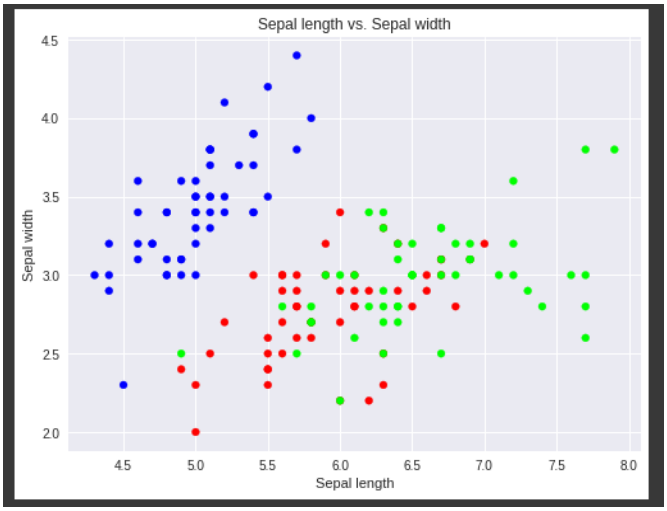
$$(\lambda I - A)v = 0$$

Step 4: Choosing components and forming a feature vector:

We order the eigenvalues from largest to smallest so that it gives us the components in order or significance. Here comes the dimensionality reduction part. If we have a dataset with n variables, then we have the corresponding n eigenvalues and eigenvectors. It turns out that the eigenvector corresponding to the highest eigenvalue is the principal component of the dataset and it is our call as to how many eigenvalues we choose to proceed our analysis with. To reduce the dimensions, we choose the first p eigenvalues and ignore the rest. We do lose out some information in the process, but if the eigenvalues are small, we do not lose much.

Output:

Plot the training points



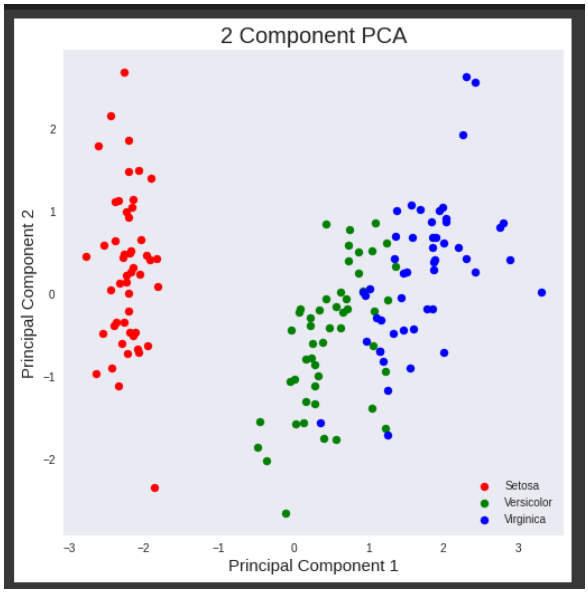
Standardize the Data

	sepal.length	sepal.width	petal.length	petal.width
0	-0.900681	1.019004	-1.340227	-1.315444
1	-1.143017	-0.131979	-1.340227	-1.315444
2	-1.385353	0.328414	-1.397064	-1.315444
3	-1.506521	0.098217	-1.283389	-1.315444
4	-1.021849	1.249201	-1.340227	-1.315444

PCA projection in 2D

	principal component 1	principal component 2
0	-2.264703	0.480027
1	-2.080961	-0.674134
2	-2.364229	-0.341908
3	-2.299384	-0.597395
4	-2.389842	0.646835

Visualize 2D Projection



Conclusion:

Thus, students can understand & implement principal component analysis for feature reduction on IRIS Dataset successfully.

Title: Assignment on Decision Tree Classifier

Aim:
Implement the Decision Tree Classifier on given Problem Statement.

Dataset collected in a cosmetics shop showing details of customers and whether or not they responded to a special offer to buy a new lip-stick is shown in table below. Use this dataset to build a decision tree, with Buys as the target variable, to help in buying lip-sticks in the future. Find the root node of decision tree. According to the decision tree you have made from previous training data set, what is the decision for the test data: [Age < 21, Income = Low, Gender = Female, Marital Status = Married]?

ID	Age	Income	Gender	Marital Status	Buys
1	< 21	High	Male	Single	No
2	< 21	High	Male	Married	No
3	21-35	High	Male	Single	Yes
4	>35	Medium	Male	Single	Yes
5	>35	Low	Female	Single	Yes
6	>35	Low	Female	Married	No
7	21-35	Low	Female	Married	Yes
8	< 21	Medium	Male	Single	No
9	<21	Low	Female	Married	Yes
10	> 35	Medium	Female	Single	Yes
11	< 21	Medium	Female	Married	Yes
12	21-35	Medium	Male	Married	Yes
13	21-35	High	Female	Single	Yes
14	> 35	Medium	Male	Married	No

Theory:

Decision Tree is a decision-making tool that uses a flowchart-like tree structure or is a model of decisions and all of their possible results, including outcomes, input costs and utility.

Decision-tree algorithm falls under the category of supervised learning algorithms. It works for both continuous as well as categorical output variables.

The branches/edges represent the result of the node and the nodes have either:

1. Conditions [Decision Nodes]
2. Result [End Nodes]

A decision tree can be visualized. A decision tree is one of the many Machine Learning algorithms.

It's used as classifier: given input data, it is class A or class B? In this lecture we will visualize a decision tree using the Python module **pydotplus and the module graphviz.**

If you want to do decision tree analysis, to understand the decision tree algorithm / model or if you just need a decision tree maker - you'll need to visualize the decision tree.

Decision Tree

Install

You need to install pydotplus and graphviz. These can be installed with your package manager and pip.

Graphviz is a tool for drawing graphics using dot files. Pydotplus is a module to Graphviz's Dot language.

Data Collection

We start by defining the code and data collection. Let's make the decision tree on Yes or No for Buys. We start with the training data:

ID	Age	Income	Gender	Marital Status	Buys
1	< 21	High	Male	Single	No
2	< 21	High	Male	Married	No
3	21-35	High	Male	Single	Yes
4	>35	Medium	Male	Single	Yes
5	>35	Low	Female	Single	Yes
6	>35	Low	Female	Married	No
7	21-35	Low	Female	Married	Yes
8	< 21	Medium	Male	Single	No
9	<21	Low	Female	Married	Yes
10	> 35	Medium	Female	Single	Yes
11	< 21	Medium	Female	Married	Yes
12	21-35	Medium	Male	Married	Yes
13	21-35	High	Female	Single	Yes
14	> 35	Medium	Male	Married	No

Step1-Compute Entropy for Data Set

Out of 14 instances we have 9 YES and 5 NO

So we have the formula,

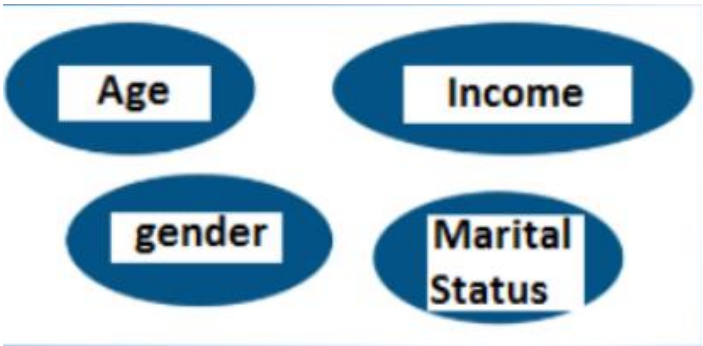
$E(S) = -P(Yes)\log_2 P(Yes) - P(No)\log_2 P(No)$

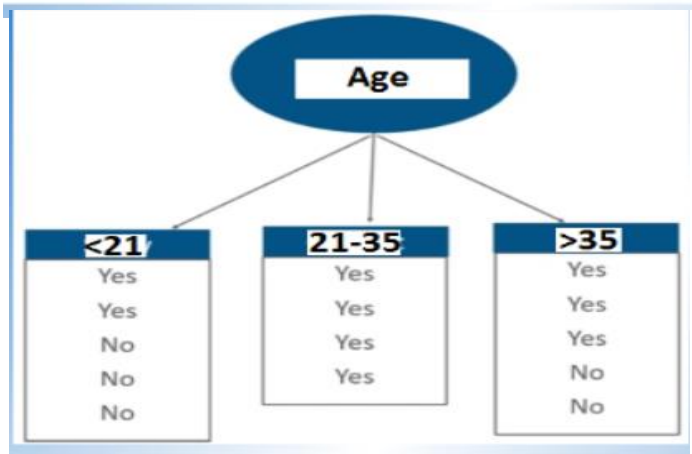
$E(S) = -(9/14)^* \log_2 9/14 - (5/14)^* \log_2 5/14$

$E(S) = 0.41+0.53 = 0.94$

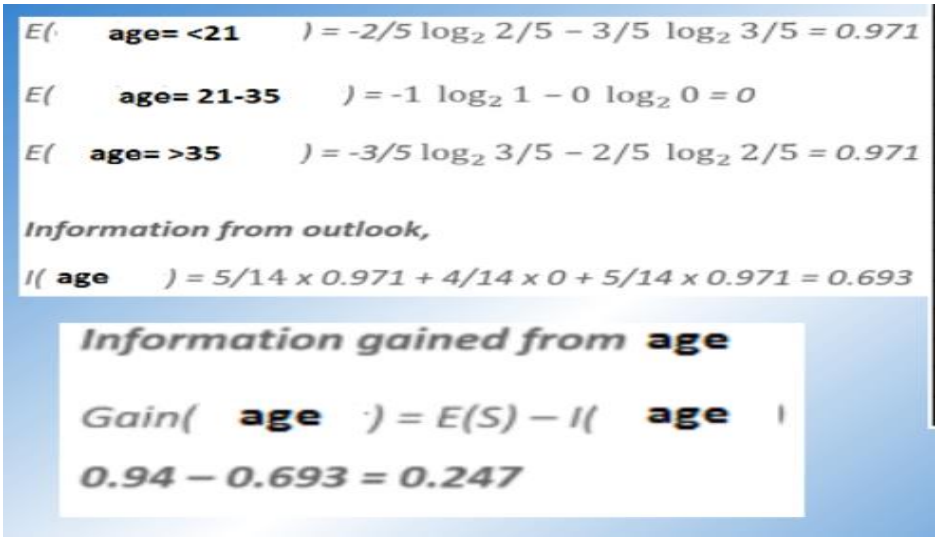
ID	Age	Income	Gender	Marital Status	Buys
1	< 21	High	Male	Single	No
2	< 21	High	Male	Married	No
3	21-35	High	Male	Single	Yes
4	>35	Medium	Male	Single	Yes
5	>35	Low	Female	Single	Yes
6	>35	Low	Female	Married	No
7	21-35	Low	Female	Married	Yes
8	< 21	Medium	Male	Single	No
9	<21	Low	Female	Married	Yes
10	> 35	Medium	Female	Single	Yes
11	< 21	Medium	Female	Married	Yes
12	21-35	Medium	Male	Married	Yes
13	21-35	High	Female	Single	Yes
14	> 35	Medium	Male	Married	No

Step2-Which Node to select as Root





Step3:Find Maximum Gain As Root

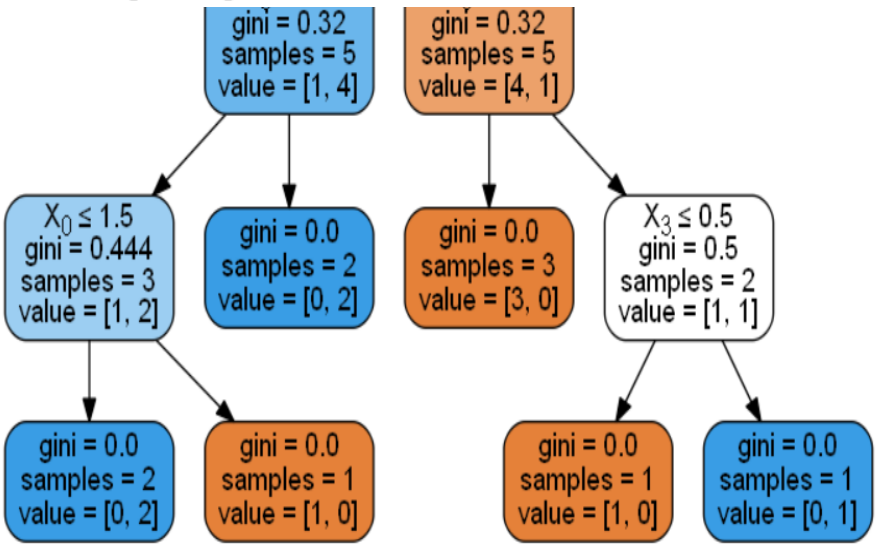


AGE IS Root Node Which has maximum Gain

Output:

	id	age	income	gender	marital_status
0	0	1	0	1	1
1	1	1	0	1	0
2	2	0	0	1	1
3	3	2	2	1	1
4	4	2	1	0	1
5	5	2	1	0	0
6	6	0	1	0	0
7	7	1	2	1	1
8	8	1	1	0	0
9	9	2	2	0	1
10	10	1	2	0	0
11	11	0	2	1	0
12	12	0	0	0	1
13	13	2	2	1	0

```
Prediction: ['Yes']
```



DECISION TREE ADVANTAGES

- Decision trees are powerful and popular tools for classification and prediction.
- Simpler and ease of use.
- They are able to handle both numerical and categorical attributes

- Easy to understand.
- State is recorded in memory.
- Provide a clear indication of which fields are most important for prediction or classification.
- Can be learned.

DECISION TREE DISADVANTAGES

- Each tree is "unique" sequence of tests, so little common structure.
- Perform poorly with many class and small data.
- Need as many examples as possible.
- Higher CPU cost - but not much higher.
- Learned decision trees may contain errors.
- Hugely impacted by data input.
- Duplicate in sub trees

DECISION TREE APPLICATIONS

- Medical diagnosis.
- Credit risk analysis.
- Library book use.

Conclusion: Thus, Students can learn how to create Decision Tree based on given decision, Find the Root Node of the tree using Decision tree Classifier successfully.

Title: Assignment on Naive Bayes Classification Algorithm

Aim: Implement Naive Bayes Classification Algorithm on a given dataset.

Objectives: 1. Study the concept of Naive Bayes Algorithm
2. Implement the Naive Bayes Algorithm on Irisis.csv dataset.

Input: Irises.csv

Theory:

Naive Bayes Classification Algorithm

Naive Bayes algorithm is a supervised learning algorithm, which is based on Bayes theorem and used for solving classification problems. It is mainly used in *text classification* that includes a high-dimensional training dataset.

Naive Bayes Classifier is one of the simple and most effective Classification algorithms which helps in building the fast machine learning models that can make quick predictions. It is a probabilistic classifier, which means it predicts on the basis of the probability of an object. Some popular examples of Naive Bayes Algorithm are spam filtration, Sentimental analysis, and classifying articles.

Bayes' Theorem:

Bayes' theorem is also known as **Bayes' Rule** or **Bayes' law**, which is used to determine the probability of a hypothesis with prior knowledge. It depends on the conditional probability.The formula for Bayes' theorem is given as:

$$P(A|B)= \frac{P(B|A)P(A)}{P(B)}$$

Where,

- P(A|B) is Posterior probability: Probability of hypothesis A on the observed event B.
- P(B|A) is Likelihood probability: Probability of the evidence given that the probability of a hypothesis is true.
- P(A) is Prior Probability: Probability of hypothesis before observing the evidence.
- P(B) is Marginal Probability: Probability of Evidence.

Algorithm:

- Data Preprocessing step
- Fitting Naive Bayes to the Training set
- Predicting the test result
- Test accuracy of the result(Creation of Confusion matrix)
- Visualizing the test set result.

Advantages of Naïve Bayes Classifier:

- Naïve Bayes is one of the fast and easy ML algorithms to predict a class of datasets.
- It can be used for Binary as well as Multi-class Classifications.
- It performs well in Multi-class predictions as compared to the other Algorithms.
- It is the most popular choice for text classification problems.

Disadvantages of Naïve Bayes Classifier:

- Naive Bayes assumes that all features are independent or unrelated, so it cannot learn the relationship between features.

Applications of Naïve Bayes Classifier:

- It is used for Credit Scoring.
- It is used in medical data classification.
- It can be used in real-time predictions because Naïve Bayes Classifier is an eager learner.
- It is used in Text classification such as Spam filtering and Sentiment analysis.

Output:

Correct predictions: 28

False Predictions: 2

Accuracy of the Naive Bayes Classification is: 0.9333333333333333

Conclusion:

Students can have studied about Naive Bayes Classification and implemented it on Irisis.csv dataset for classification

Title: Assignment on SVM (Support Vector Machine) Algorithm

Aim: Implement SVM Algorithm on a given dataset.

Objectives: Implement the SVM on any dataset.

Input: .csv File

Theory:

Support Vector Machine Algorithm:

Support Vector Machine 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 Machine Learning.

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 the correct category in the future. This best decision boundary is called a hyperplane.

SVM chooses the extreme points/vectors that help in creating the hyperplane. These extreme cases are called as support vectors, and hence algorithm is termed as Support Vector Machine. Consider the below diagram in which there are two different categories that are classified using a decision boundary or hyperplane

Types of SVM

SVM can be of two types:

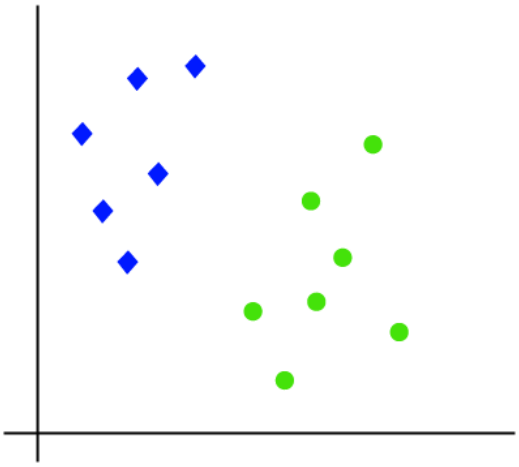
- **Linear SVM:** Linear SVM is used for linearly separable data, which means if a dataset can be classified into two classes by using a single straight line, then such data is termed as linearly separable data, and classifier is used called as Linear SVM classifier.

- **Non-linear SVM:** Non-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.

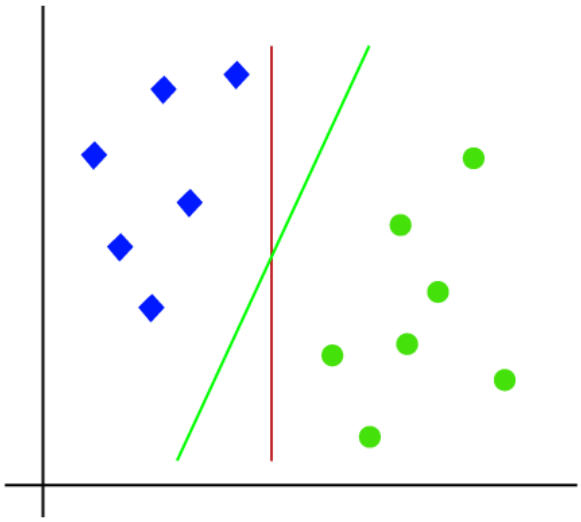
Hyperplane and Support Vectors in the SVM algorithm:
Hyperplane: There can be multiple lines/decision boundaries to segregate the classes in n-dimensional space, but we need to find out the best decision boundary that helps to classify the data points. This best boundary is known as the hyperplane of SVM.
The dimensions of the hyperplane depend on the features present in the dataset, which means if there are 2 features (as shown in image), then hyperplane will be a straight line. And if there are 3 features, then hyperplane will be a 2-dimension plane.
We always create a hyperplane that has a maximum margin, which means the maximum distance between the data points.

Support Vectors:
The data points or vectors that are the closest to the hyperplane and which affect the position of the hyperplane are termed as Support Vector. Since these vectors support the hyperplane, hence called a Support vector.

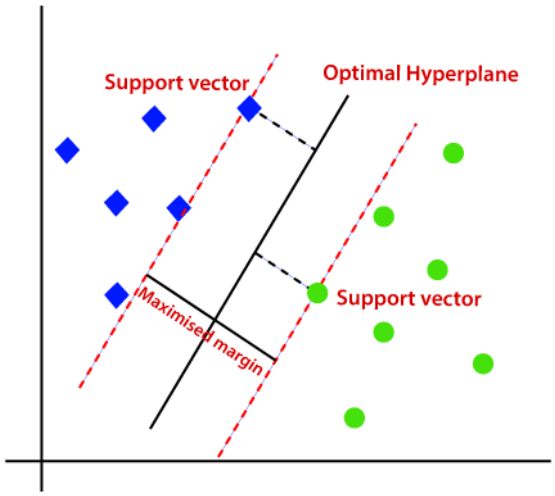
How does SVM works?
Linear SVM:
The working of the SVM algorithm can be understood by using an example. Suppose we have a dataset that has two tags (green and blue), and the dataset has two features x1 and x2. We want a classifier that can classify the pair(x1, x2) of coordinates in either green or blue. Consider the below image:



So as it is 2-d space so by just using a straight line, we can easily separate these two classes. But there can be multiple lines that can separate these classes. Consider the below image:



Hence, the SVM algorithm helps to find the best line or decision boundary; this best boundary or region is called as a **hyperplane**. SVM algorithm finds the closest point of the lines from both the classes. These points are called support vectors. The distance between the vectors and the hyperplane is called as **margin**. And the goal of SVM is to maximize this margin.
The **hyperplane** with maximum margin is called the **optimal hyperplane**.



Algorithm Steps:

- Load the breast cancer dataset from sklearn.datasets
- Separate input features and target variables.
- Buil and train the SVM classifiers using RBF kernel.
- Plot the scatter plot of the input features.
- Plot the decision boundary.
- Plot the decision boundary

Applications of SVM Classifier:

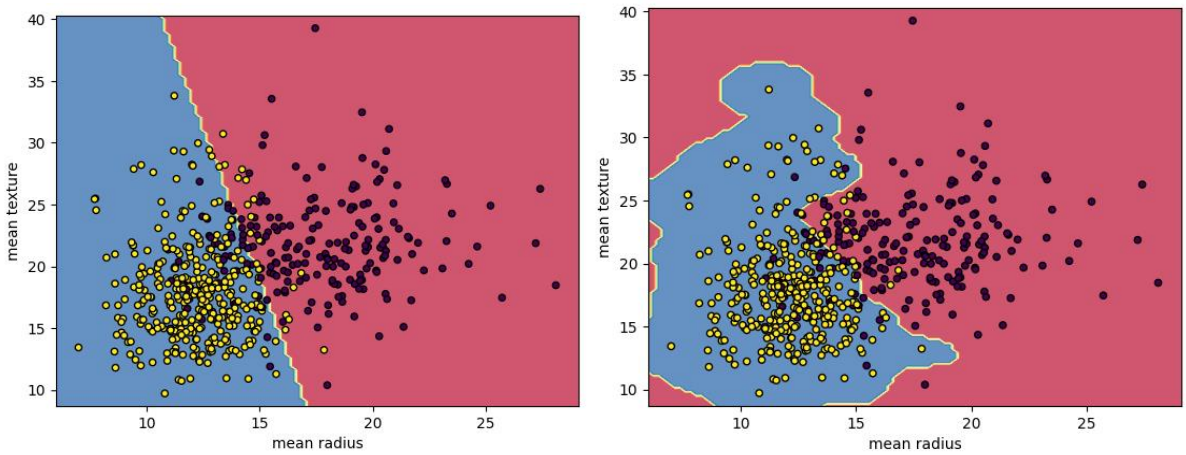
Some common applications of SVM are-

- **Face detection** – SVMc classify parts of the image as a face and non-face and create a square boundary around the face.
- **Text and hypertext categorization** – SVMs allow Text and hypertext categorization for both inductive and transductive models. They use training data to classify documents into different categories. It categorizes on the basis of the score generated and then compares with the threshold value.
- **Classification of images** – Use of SVMs provides better search accuracy for image classification. It provides better accuracy in comparison to the traditional query-based searching techniques.
- **Bioinformatics** – It includes protein classification and cancer classification. We use SVM for identifying the classification of genes, patients on the basis of genes and other biological problems.

- **Protein fold and remote homology detection** – Apply SVM algorithms for protein remote homology detection.
- **Handwriting recognition** – We use SVMs to recognize handwritten characters used widely.
- **Generalized predictive control(GPC)** – Use SVM based GPC to control chaotic dynamics with useful parameters.

Output:

Output: **Linear –Kernal** **RBF-Kernal**



Conclusion:

Studied about SVM Classification and implemented it on `breast_cancer` dataset for classification.

Title:Assignment on k-Mean Algorithm

Aim:

Implement k-Mean algorithm

Problem Statement:

We have given a collection of 8 points. P1=[0.1,0.6] P2=[0.15,0.71] P3=[0.08,0.9] P4=[0.16,0.85] P5=[0.2,0.3] P6=[0.25,0.5] P7=[0.24,0.1] P8=[0.3,0.2]. Perform the k-mean clustering with initial centroids as m1=P1 =Cluster#1=C1 and m2=P8=cluster#2=C2. Answer the following
1] Which cluster does P6 belongs to?
2] What is the population of cluster around m2?
3] What is updated value of m1 and m2?

Prerequisite:

Basic of Python, Data Mining Algorithm, Concept of K-mean Clustering

Theory:

A Hospital Care chain wants to open a series of Emergency-Care wards within a region. We assume that the hospital knows the location of all the maximum accident-prone areas in the region. They have to decide the number of the Emergency Units to be opened and the location of these Emergency Units, so that all the accident-prone areas is covered in the vicinity of these Emergency Units.

The challenge is to decide the location of these Emergency Units so that the whole region is covered. Here is when K-means Clustering comes to rescue!

A cluster refers to a small group of objects. Clustering is grouping those objects into clusters. In order to learn clustering, it is important to understand the scenarios that lead to cluster different objects. Let us identify a few of them.

What is Clustering?

Clustering is dividing data points into homogeneous classes or clusters:

Points in the same group are as similar as possible

Points in different group are as dissimilar as possible

When a collection of objects is given, we put objects into group based on similarity.

Application of Clustering:

Clustering is used in almost all the fields. You can infer some ideas from Example 1 to come up with lot of clustering applications that you would have come across.

Listed here are few more applications, which would add to what you have learnt.

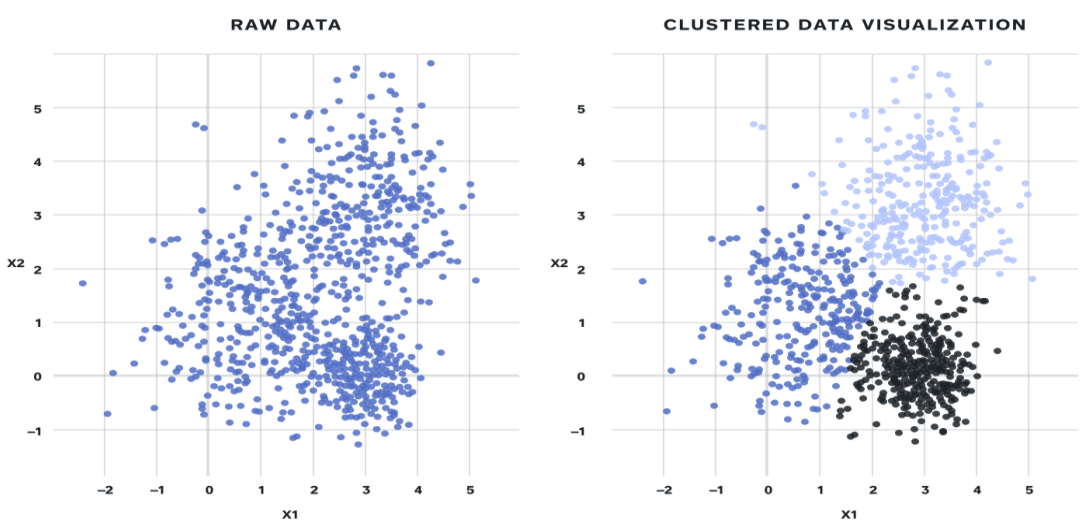
5. Clustering helps marketers improve their customer base and work on the target areas. It helps group people (according to different criteria's such as willingness, purchasing power etc.) based on their similarity in many ways related to the product under consideration.

- Clustering helps in identification of groups of houses on the basis of their value, type and geographical locations.
- Clustering is used to study earth-quake. Based on the areas hit by an earthquake in a region, clustering can help analyse the next probable location where earthquake can occur.

Clustering Algorithms:

A Clustering Algorithm tries to analyse natural groups of data on the basis of some similarity. It locates the centroids of the group of data points. To carry out effective clustering, the algorithm evaluates the distance between each point from the centroids of the cluster.

The goal of clustering is to determine the intrinsic grouping in a set of unlabelled data.



What is K-means Clustering?

K-means (Macqueen, 1967) is one of the simplest unsupervised learning algorithms that solve the well-known clustering problem. K-means clustering is a method of vector quantization, originally from signal processing, that is popular for cluster analysis in data mining.

K-means Clustering – Example 1:

A pizza chain wants to open its delivery centres across a city. What do you think would be the possible challenges?

- They need to analyse the areas from where the pizza is being ordered frequently.
- They need to understand as to how many pizza stores has to be opened to cover delivery in the area.
- They need to figure out the locations for the pizza stores within all these areas in order to keep the distance between the store and delivery points minimum.

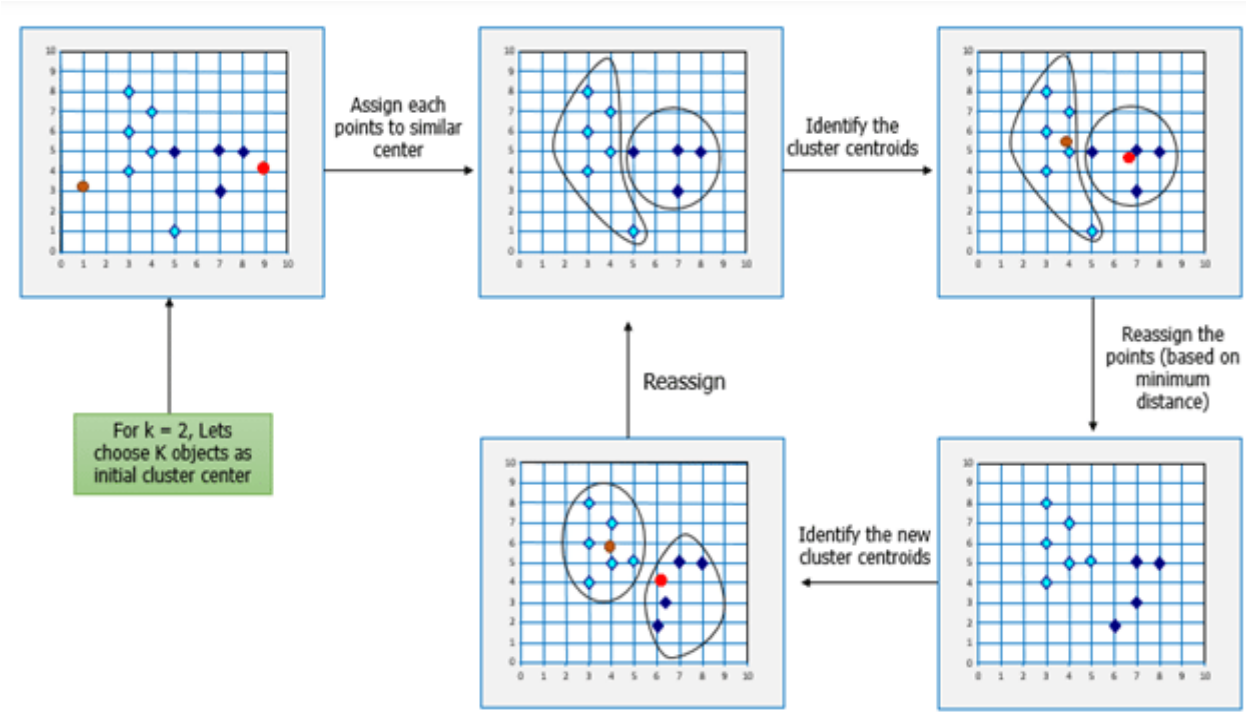
Resolving these challenges includes a lot of analysis and mathematics. We would now learn about how clustering can provide a meaningful and easy method of sorting out such real life challenges. Before that let's see what clustering is.

K-means Clustering Method:

If k is given, the K-means algorithm can be executed in the following steps:

- Partition of objects into k non-empty subsets
- Identifying the cluster centroids (mean point) of the current partition.
- Assigning each point to a specific cluster
- Compute the distances from each point and allot points to the cluster where the distance from the centroids is minimum.
- After re-allotting the points, find the centroids of the new cluster formed.

The step by step process of clustering:



Now, let's consider the problem in Example 1 and see how we can help the pizza chain to come up with centres based on K-means algorithm.

Similarly, for opening Hospital Care Wards:

K-means Clustering will group these locations of maximum prone areas into clusters and define a cluster center for each cluster, which will be the locations where the Emergency Units will open. These Clusters centers are the centroids of each cluster and are at a minimum distance from all the points of a particular cluster, henceforth, the Emergency Units will be at minimum distance from

all the accident prone areas within a cluster.

Here is another example for you, try and come up with the solution based on your understanding of K-means clustering.

K-means Clustering – Example 2:

Let's consider the data on drug-related crimes in Canada. The data consists of crimes due to various drugs that include, Heroin, Cocaine to prescription drugs, especially by underage people. The crimes resulted due to these substance abuse can be brought down by starting de-addiction centres in areas most afflicted by this kind of crime. With the available data, different objectives can be set. They are:

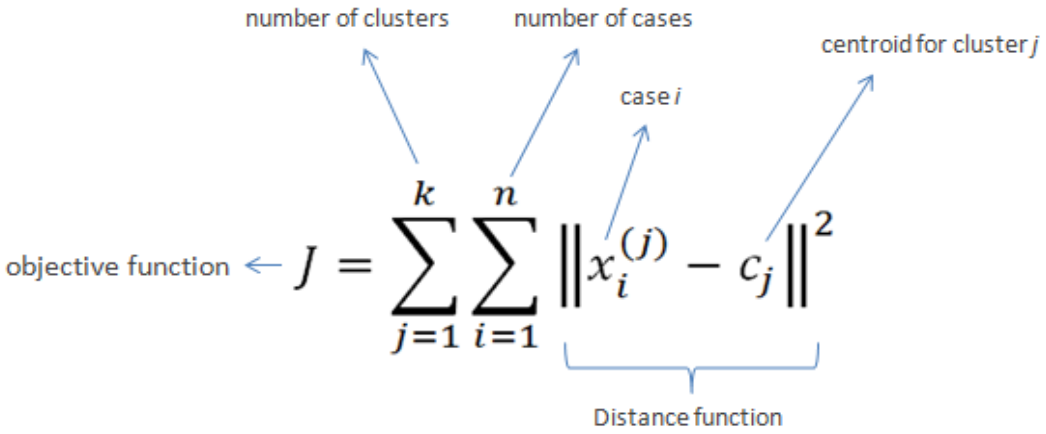
- Classify the crimes based on the abuse substance to detect prominent cause.
- Classify the crimes based on age groups.
- Analyze the data to determine what kinds of de-addiction centre are required.
- Find out how many de-addiction centres need to be setup to reduce drug related crime rate.

The K-means algorithm can be used to determine any of the above scenarios by analyzing the available data.

Following the K-means Clustering method used in the previous example, we can start off with a given k, following by the execution of the K-means algorithm.

Mathematical Formulation for K-means Algorithm:

K-Means clustering intends to partition n objects into k clusters in which each object belongs to the cluster with
nearest mean. This method produces exactly k different clusters of greatest possible distinction. The best numbe
of clusters k leading he greatest separation (distance) is not known as a priori and must be computed from the da
The objective of K-Means clustering is to minimize total intra-cluster variance, or, the squared error function

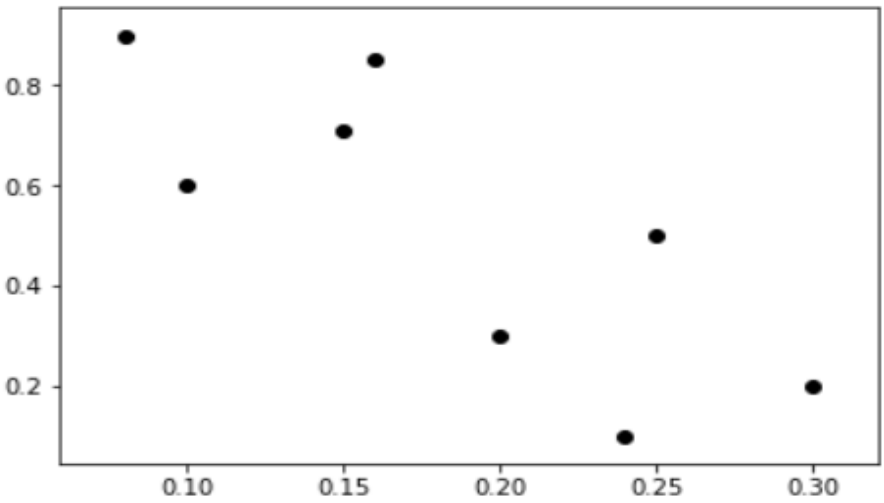


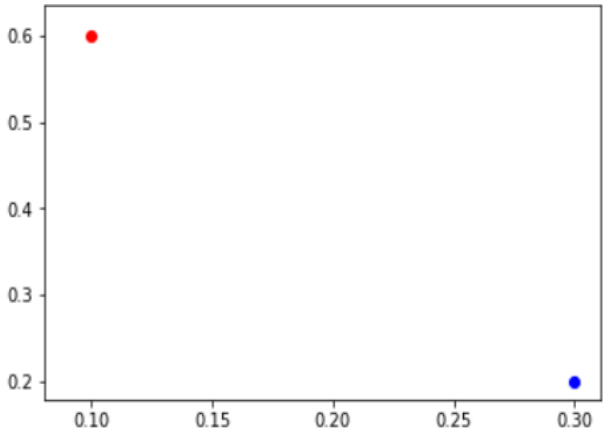
Algorithm:

1. **Import the Required Packages**
2. **Create dataset using DataFrame**
3. **Find centroids points**
4. **Plot the given points**
5. **For i in centroids ():**
6. **Plot given elements with centroids elements**
7. **Import KMeans class and create object of it**
8. **Using labels find population around centroids**
9. **Find new centroids**

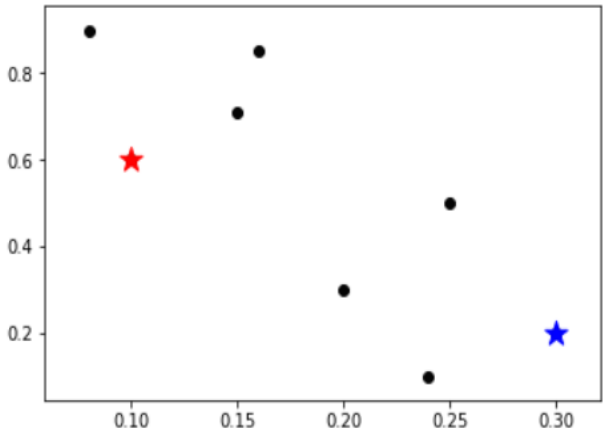
Output:

```
[[0.1  0.6 ]
 [0.15 0.71]
 [0.08 0.9 ]
 [0.16 0.85]
 [0.2  0.3 ]
 [0.25 0.5 ]
 [0.24 0.1 ]
 [0.3  0.2 ]]
```






```
[[0.1 0.3]
 [0.6 0.2]]
point No.6[0.25,0.5] is belongs to blue cluster(cluster no:2)
```




```
[1 1 1 1 0 0 0 0]
No of population around cluster 2: 3
Previous value of m1 and m2 is:
M1== [0.1 0.3]
M1== [0.6 0.2]
Updated value of m1 and m2 is:
M1== [0.2475 0.275 ]
M1== [0.1225 0.765 ]
```

Conclusion:

Students can have studied & implement Kmean Clustering Algorithm successfully.

Title: Assignment on Gradient Boost Classifier

Aim: Implement **Gradient Boost Classifier** Algorithm on a given dataset.

Objectives: Implement the GBC on any dataset.

Input: .csv File

Theory:

Gradient Boost Classifier Algorithm:

Gradient Boosting is a functional gradient algorithm that repeatedly selects a function that leads in the direction of a weak hypothesis or negative gradient so that it can minimize a loss function. Gradient boosting classifier combines several weak learning models to produce a powerful predicting model.

Gradient Boosting in Classification

Gradient Boosting consists of three essential parts:

Loss Function

The loss function's purpose is to calculate how well the model predicts, given the available data. Depending on the particular issue at hand, this may change.

Weak Learner

A weak learner classifies the data, but it makes a lot of mistakes in doing so. Usually, these are decision trees.

Additive Model

This is how the trees are added incrementally, iteratively, and sequentially. You should be getting closer to your final model with each iteration.

Steps to Gradient Boosting

Gradient boosting classifier requires these steps:

Fit the model

Adapt the model's Hyperparameters and Parameters.

Make forecasts

Interpret the findings

Visualizing Gradient Boosting

1. The method will obtain the log of the chances to make early predictions about the data. Typically, this is the ratio of the number of True values to the False values.

2. If you have a dataset of six cancer occurrences, with four people with cancer and three who are not suffering, then the log(odds) is equal to log(4/3) 1.3, and the person who is free of cancer will have a value of 0. The person who has cancer will have a value of 1.

3. To make predictions, you must first convert the log(odds) to a probability with the help of a logistic function. Here, it would be around 1.3, the same as the log(odds) value of 1.3

4. Since it is greater than 0.5, the algorithm will use 1.3 as its baseline estimate for each occurrence.

e * log(odds) / (1 + e * log(odds))

5. The above formula will determine the residuals for each occurrence in the training set.

6. After completing this, it constructs a Decision Tree to forecast the estimated residuals.

7. A maximum number of leaves can be used while creating a decision tree. This results in two potential outcomes:

Several instances are into the same leaf.
The leaf is not a single instance.
You must use a formula to modify these values here:

ΣResidual / Previous Prob (1 - Previous Prob)]

8. You must now complete two things:

Obtain the log forecast for each training set instance.
Transform the forecast into a probability.
9. The formula for producing predictions would be as follows:

base_log_odds + (learning_rate * predicted residual value)

Advantages and Disadvantages of Gradient Boost

Advantages:
Frequently has remarkable forecasting accuracy.
Numerous choices for hyperparameter adjustment and the ability to optimize various loss functions.
It frequently works well with numerical and categorical values without pre-processing the input.
Deals with missing data; imputation is not necessary.

Disadvantages:
Gradient Boosting classifier will keep getting better to reduce all inaccuracies. This may lead to overfitting and an overemphasis on outliers.
Costly to compute since it frequently requires a large number of trees (>1000), which can be memory and time-consuming.
Due to the high degree of flexibility, numerous variables interact and significantly affect how the technique behaves.
Less interpretative, even though this can be easily corrected with several tools.

Output:

{'Gradient-Boost Default Test Score': 0.8708736373407032,
'Gradient-Boost GridSearch Test Score': 0.8785505911254414}

Conclusion:

Students can have studied about Gradient Boost Classification and implemented it on income_evaluation.csv dataset for classification.