Assignment No. 1

Page No.
Date

Title: Linear Regression Analysis :- Driving hours vs Backache Risk.

Problem Statement:
We are given small dataset containing the number of hours spent driving (x) and a corresponding risk score (y) for developing acute backache.
Our goal is to:
- Analyze the data using simple linear regression.
- Derive the equation of the best-fit-line.
- Interpret the results and their real world implications.

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

Independent Variable : x is the no. of hours spent driving
dependent variable : y is the risk score for backache.

Analysis and Consumption:

1. Assumptions:

Before proceeding, we assume:

- A linear relationship exists between driving hours and risk score.
- No significant outliers affect the results.
- The dataset is representative of the population.

2. Linear Regression formula:

$$Y = mx + c$$

m is slope of line

c is the y-intercept

Calculation of m and c

$$m = \frac{n(\Sigma xy) - (\Sigma x)(\Sigma y)}{n(\Sigma x^2) - (\Sigma x)^2}$$

Calculations:

x	y	x^2	y^2	xy
10	95	100	9025	950
9	80	81	6400	720
2	10	4	100	20
15	50	225	2500	750
10	45	100	2025	450
16	98	256	9604	1568
11	38	121	1444	418
16	93	256	8649	1488

Now sum each column

$\Sigma x = 89$

$\Sigma y = 509$

$\Sigma x^2 = 1143$

$\Sigma xy = 6364$

$n = 8$

$$m = \frac{8(6364) - (89)(509)}{8(1143) - (89)^2} = \frac{50912 - 45201}{9144 - 7921} = \frac{5701}{1223} = 4.66$$

$$c = \frac{509 - (4.66)(89)}{8} = \frac{509 - 414.74}{8} = \frac{94.26}{8} = 11.78$$

Final Regression formula :

$$y = 4.66x + 11.78$$

This is best fit line describing the relationship between driving hours and backache risk.

Interpretation :

slope (m) = 4.66 :- For every additional hour spent driving, the risk score increases by approximately 4.66 units.

Intercept (c = 11.78) : Even at 0 hours of driving, a baseline risk score of around 11.78 exists, possibly due to other factors like posture or lifestyle.

Page No.
Date

Prediction:

If a person drives 12 hours a day:

$$y = 4.66(12) + 11.78 = 66.7 + 11.78 = 78.48$$

So, the predicted risk score is approximately 78.5

Conclusion:

From our analysis, there is a positive linear correlation between the number of driving hours and the risk of developing acute backache. This implies that the no. of hours spent driving increases, the backache risk also increases significantly.

Assignment No. 2

Page No.
Date

Title:

Aim: Apply the principal Component Analysis for feature reduction on IRIS Dataset

Objective: The objective is to reduce the number of features in a dataset while retaining the most important information. PCA helps simplify data visualization and speeds up processing without losing significant patterns.

Theory Overview:

- PCA is an unsupervised machine learning algorithm used for dimensionality reduction.
- It transforms correlated variables into a new set of uncorrelated variables called principle components.

Why PCA ?

- High dimensional data can be noisy and computationally expensive.
- PCA helps by compressing data into fewer dimensions while preserving trends and patterns.

Key Concepts:

Covariance matrix : Measures how variables move together
Eigen values / Eigen vectors : Identify principle direction of variance.
Orthogonality : Principal components are peripendicular (uncorrelated)
Dimensionality : Number of variables of features in the dataset.

Page No.
Date

PCA Algorithm Steps:

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. Standardize the data
   - Standardize our dataset with the particular column, the features with high variance are more important compared to the features with lower variance.
   - Make features have mean = 0 and unit variance.

3. Calculate the covariance Matrix.
   - Represents relationships between features.

4. Compute Eigenvalues and Eigenvectors
   - Identify directions with the most variance

5. Sort Eigenvalues and select top-k
   - keep the most significant components

6. Transform the Data
   - Multiply standardized data with top-k eigenvectors to get the reduced dataset.

7. Drop unimportant features.
   keep only significant principal components.

Page No.
Date

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.

Implementing PCA on a 2-D Dataset

step 1: Normalize the data
This is done by subtracting the respective means from the numbers in the respective column.

step 2: Calculate the covariance matrix
Since the dataset we took is 2-dimensional, this will result in a 2×2 Covariance matrix

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

step 3: Calculate the eigenvalues & eigenvectors

step 4: Choosing components and forming a feature vector.

Page No.
Date

Output:

- Standardized data
- PCA projection (2D scatter plot)
- Visualized reduced-dimension dataset

Conclusion: PCA was successfully implemented on the IRIS dataset for feature reduction. The dimensionality of the data was reduced while preserving its important characteristics, enabling easier visualization and analysis.

Assignment No. 3

Page No.
Date

Title: Assignment on decision tree Classifier

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	M	Single	No
2	<21	High	M	Married	No
3	21-35	High	M	Single	Yes
4	>35	Medium	M	Single	Yes
5	>35	Low	F	Single	Yes
6	>35	Low	F	Married	No
7	21-35	Low	F	Married	Yes
8	<21	Medium	M	Single	No
9	<21	Medium Low	F	Married	Yes
10	>35	Medium	F	Single	Yes
11	<21	Medium	F	Married	Yes
12	21-35	Medium	M	Married	Yes
13	21-35	High	F	Single	Yes
14	>35	Medium	M	Married	No

Page No.
Date

Relevant 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.

- Supervised learning algorithms. Work for both continous as well as categorical output variables.

- branches (edges represent the result of the node and the nodes have either.
  1. Conditions [Decision Nodes]
  2. Result [End Nodes]

Steps:

① Compute Entropy For Data set
② Which Node to select as Root
  - Find maximum Gain As Root

Age has maximum Gain therefore Age become Root node.

Source Code:

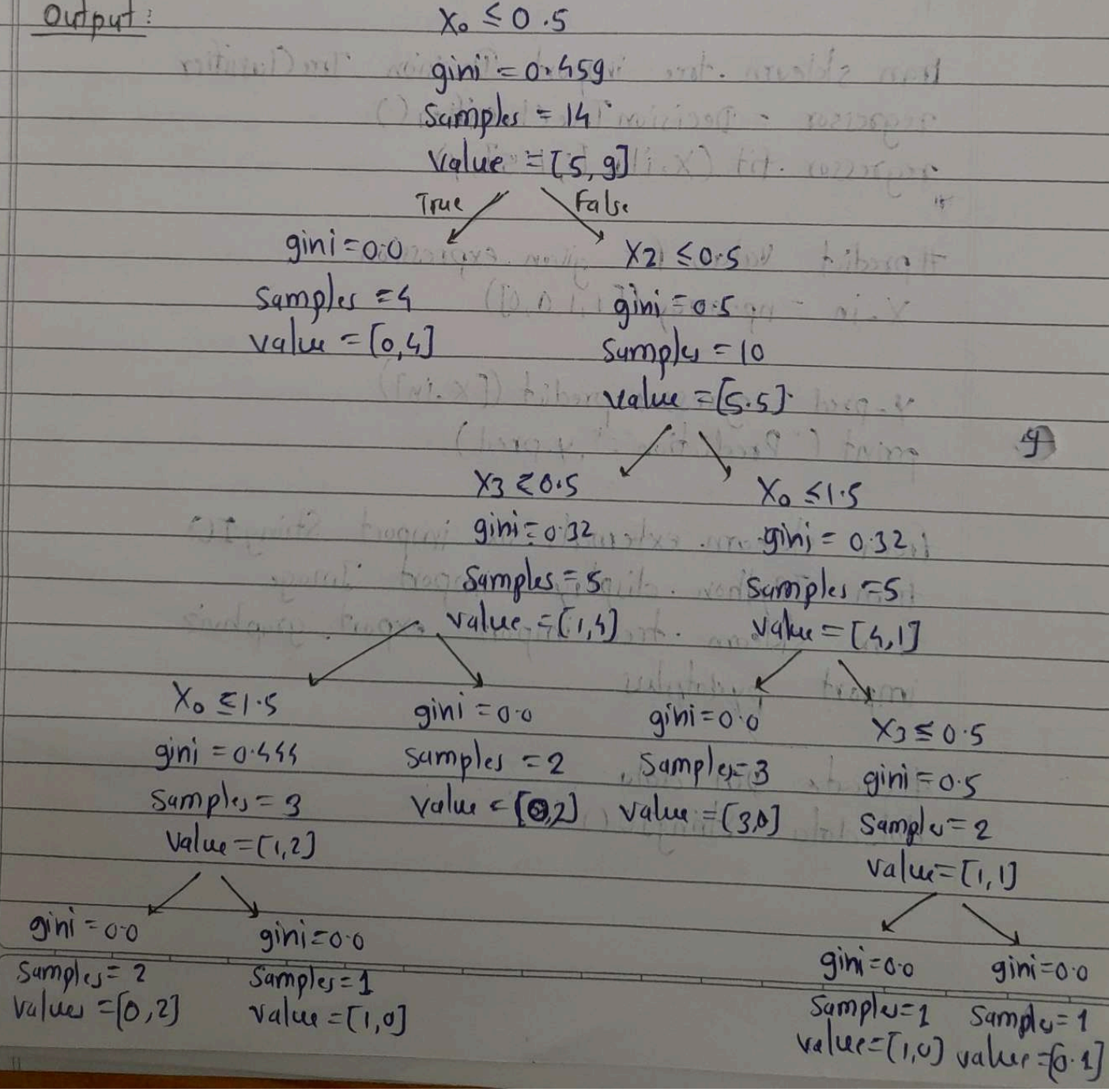
```
#import packages
import numpy as np
import matplotlib.pyplot as plt
import pandas as pd
```

Date

```
#reading dataset
dataset = pd.read_csv('data.csv')
X = dataset.iloc[:, :-1]
y = dataset.iloc[:, 5].values

#perform label encoding
from sklearn.preprocessing import LabelEncoder
labelencoder_X = LabelEncoder()

X = X.apply(LabelEncoder().fit_transform)
print(X)

from sklearn.tree import DecisionTreeClassifier
regressor = DecisionTreeClassifier()
regressor.fit(X.iloc[:,1:5], y)

#predict value for given expression
X_in = np.array([1,1,0,0])

y_pred = regressor.predict([X_in])
print("Prediction:", y_pred)

from sklearn.externals.six import StringIO
from IPython.display import Image
from sklearn.tree import export_graphviz
import pydotplus

#create DOT data
dot_data = StringIO()
```

```
export_graphviz (regressor, out_file = dot_data, filled = True,
special_characters = True)

# Draw graph
graph = pydotplus.graph_from_dot_data(dot_data.getvalue())
graph.write_png('Decision_Tree.png')

# Show graph
Image(graph.create_png())
```

Output:



Assignment No. 4.

Page No.
Date

Title: Assignment on Naive Bayes Classifier Notes

Problem Statement: Implement Naive Bayes Classification Algorithm on a given dataset

Objective: 1. Study the concept of Naive Bayes Algorithm
2. Implement the Naive Bayes Algorithm on Iris.csv dataset

Relevant Theory:

- Supervised machine learning algorithm
- Based on bayes theorem and used for solving classification problems.
- Mainly used in text classification that includes a high dimensional training dataset.
- Probabilistic model/classifier, it predicts on the basis of the probability of an object.

ex. spam filtration, Sentimental analysis and classifying articles

Formula for Bayes theorem: $P(A|B) = \frac{P(B|A)P(A)}{P(B)}$

Where, $P(A|B)$ → posterior probability: Probability of hypothesis A on observed event B.

$P(B|A)$ → Likelihood probability: probability of the evidence given that the probability of a hypothesis is true.

$P(A)$ → Prior Probability: probability of hypothesis before observing the evidence.

$P(B)$ → 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:

- One of the fast and easy ML algorithms to predict a class of datasets.
- Can be used for Binary as well as Multiclass Classifications
- Performs well in Multi-class predictions as compared to the other algorithms.
- Popular choice for text classification problems.

Disadvantages:

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

Applications of:

- Used for credit Scoring
- Used in medical data classification
- Used in real-time predictions because Naive Bayes Classifier is an eager learner.
- Used in Text classification such as Spam filtering and Sentiment analysis.

Page No.
Date

Output:

Correct Predictions: 28
False Predictions : 2
Accuracy of the Naive Bayes Classification is 0.9333333333333333

Conclusion:

Studied about Naive Bayes Classification and implemented it on Iris.csv dataset for classification.

Assignment No. 5

Page No.
Date

Title: Assignment on svm on any dataset

Objective: Implement the Svm on any dataset

Input: csv file

Relevant Theory:

- Supervised learning algorithms
- Used for classification as well as regression problem
- Goal: To create the best line or decision boundry 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.
- The best decision line is called hyperplane.

Types of SVM :-

① Linear SVM :-
- Used for linearly separable data
- 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.

② Non-linear SVM :-
- Used for non-linearly separated data
- 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.

Page No.
Date

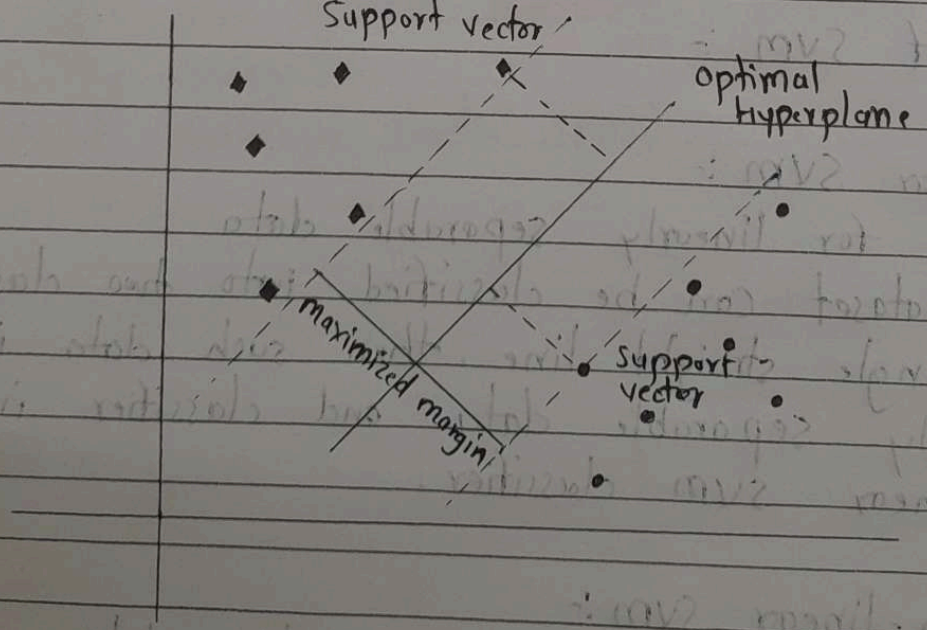
Hyperplane and Support Vectors in SVM algorithm :-

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 hyperplane of SVM.

Support Vectors

The data points or vectors that are closest to the hyperplane and which affect the position of the hyperplane are termed as the Support Vector.



Page No.
Date

Algorithm Steps

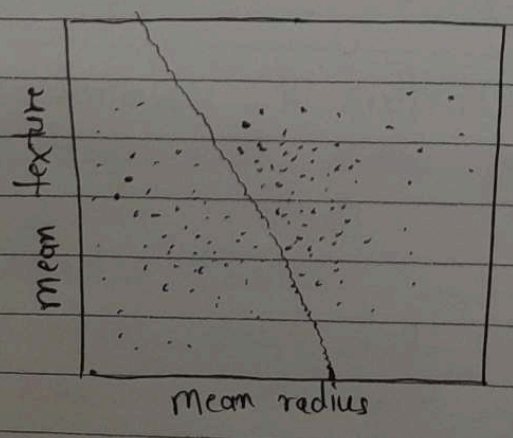
- Load the brest cancer dataset from sklearn datasets
- Separate input features and target variables
- Buil and train the SVM classifiers using RBF kernel.
- Plot the scatter plot of the input features
- plot the decision boundary
- ~~plot the~~

Applications of svm classifier:-

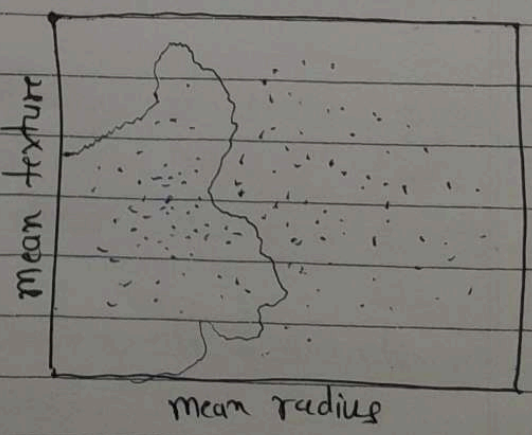
- face detection
- Text and hypertext categorization
- Classification of images
- Bioinformatics
- Protein fold and remote homology detection
- Handwriting recognition
- Generalized predictive control (GPC)

output:

Linear kernel



RBF- Kernel



Page No.
Date

Conclusion:

Studied about SVM classification and implemented it on breast-cancer dataset for classification.

Assignment No. 6

Page No.
Date

Title: Assignment on k-means clustering

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-means 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 the updated value of m1 and m2?

Relevant Theory:

- k-Means Clustering is an unsupervised machine learning algorithm
- Used to partition a dataset into k distinct clusters based on distance (usually Euclidean).

The algorithm works in iterative steps:-

1. Initialize k centroids randomly or with given points.
2. Assign each data points to the nearest centroid.
3. Update centroids as the mean of all points in the respective cluster.
4. Repeat steps 2-3 until convergence (no change in centroids or cluster assignments).

Euclidean Distance formula :-

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

Algorithm steps :-

① Initialize the centroids as m1 = P1 and m2 = P8
② for each point Pi, compute the distance to both m1 and m2.
③ Assign the point to the cluster with the nearest centroid.
④ After assigning all points, compute the new Centroid of each cluster.
⑤ Repeat until centroids do not change (only 1 iteration is asked here)

Implementation Steps :

Initial Centroids :
- m1 = [0.1, 0.6]
- m2 = [0.3, 0.2]

step① : Calculate Distance of P6 from m1 and m2

: P6 = [0.25, 0.5]

$$d(P_6, m_1) = \sqrt{(0.25 - 0.1)^2 + (0.5 - 0.6)^2}$$
$$= \sqrt{0.0225 + 0.01}$$
$$= \sqrt{0.0325}$$
$$\approx 0.18$$

$$d(P_6, m2) = \sqrt{(0.25-0.3)^2 + (0.5-0.2)^2}$$
$$= \sqrt{0.0925} \approx 0.304$$

P6 is closer to m1, so it belongs to cluster 1 (C1)

step ② :- Assign All points to clusters (First iteration)

Point	Distance to m1	Distance to m2	cluster
P1	0	0.5	C1
P2	0.1118	0.5831	C1
P3	0.3041	0.7071	C1
P4	0.254	0.6503	C1
P5	0.3162	0.1414	C2
P6	0.1803	0.3041	C1
P7	0.5099	0.1414	C2
P8	0.5	0	

step ③ : Answering the questions

① Which cluster does P6 belong to ?
→ cluster 1 (C1)

② What is the population of the cluster around m2 (cluster 2) ?
→ P5, P7, P8 → 3 points

③ Updated centroids for C1 (P1, P2, P3, P4, P6) :

$$m1_{new} = \frac{1}{5} \Sigma P = \frac{[0.1+0.15+0.08+0.16+0.25, 0.6+0.71+0.9+0.85+0.5]}{5}$$

$$= [0.148, 0.712]$$

- For C2 (P5, P7, P8):

$$m2_{new} = \frac{[0.2+0.24+0.3,\ 0.3+0.1+0.2]}{3}$$

$$= [0.2467, 0.2]$$

Conclusion:

The k-means clustering algorithm was successfully applied on the given 8 data points. After the first iteration.

- P6 belongs to cluster 1
- Cluster 2 has population of 3 points
- Updated centroids are:
  - $m1 = [0.148, 0.712]$
  - $m2 = [0.2467, 0.2]$

This demonstrates the fundamental working of k-means clustering in grouping points based on proximity to centroids.

Assignment No. 7

Page No.
Date

Title: Assignment on Gradient Boost Classifier

Aim: Implement Gradient Boost Classifier algorithm on given dataset.

Relevant Theory:

- 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 consists of three essential parts:

① Loss function
- loss function's purpose is to calculate how well the model predicts, given the available data.

② Weak learner
- It classifies the data, but it makes a lot of mistakes in doing so. Usually these are decision trees.

③ 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.

Page No.
Date

Steps in Gradient Boosting :-

1. Fit the initial model
2. Adjust hyperparameters
3. Make predictions
4. Analyze results.

Working:

- Uses log(odds) to make predictions
- Converts log(odds) to probability using logistic function

$$\text{Probability} = \frac{e^{\log(\text{odds})}}{1+e^{\log(\text{odds})}}$$

- Updates predictions using:

New log odds = Base log odds + (learning rate × residual)

Formula for modifying leaf values:

$$\Sigma \text{Residual} / (\text{Previous Prob} \times (1 - \text{Previous Prob}))$$

Pros and Cons

Advantages :-
- High accuracy
- Handles both numerical & categorical data
- Robust to missing data
- Flexible with hyperparameters

Page No.
Date

Disadvantages :

- Can overfit
- Requires many trees → high computation cost
- Difficult to interpret without tools.

Output :

{'Gradient-Boost Default Test Score': 0.8708736373507032
'Gradient-Boost GridSearch Test Score': 0.8785505911254414}

Conclusion:

~~Student~~ a studied about Gradient Boost Classification and implemented it on income_evaluation.csv dataset for classification