



DATE: _____

PRACTICAL-1

AIM: Explore following python libraries:

- i. numpy
- ii. pandas
- iii. sklearn
- iv. matplotlib

Description:

1. NumPy :

```
import numpy as np
arr = np.array([1, 2, 3, 4, 5])
print("Numpy Array:", arr)
print("Sum:", np.sum(arr))
print("Mean:", np.mean(arr))
print("Standard Deviation:", np.std(arr))
```

Output:

Numpy Array:[1, 2 3 4 5]
Sum: 15
Mean: 3.0
Standard Deviation: 1.4142135623730951

2. Pandas :

```
import pandas as pd
data = {'Name': ['Alice', 'Bob', 'Charlie'],
        'Age': [25, 30, 35]}
df = pd.DataFrame(data)
print("Pandas Dataframe:\n", df)
```

Output: Pandas

0
1
2

Dataframe:

Name	Age
Alice	25
Bob	30
Charlie	35



**SILVER OAK
UNIVERSITY**
EDUCATION TO INNOVATION

3. sklearn

```
import numpy as np
from sklearn.linear_model import LinearRegression.

x = np.array([1, 2, 3, 4, 5]).reshape(-1, 1)
y = np.array([2, 4, 6, 8, 10])
model = LinearRegression()
model.fit(x, y)
print("Prediction for x=6:", model.predict([6])[0])
```

Output:

Prediction for x=6: 12.0

4. matplotlib:

```
import matplotlib.pyplot as plt
x = [1, 2, 3, 4, 5]
y = [i**2 for i in x]
plt.plot(x, y, marker='o', linestyle='-', color='b',
          label='y = x^2')
plt.xlabel("x-axis")
plt.ylabel("y-axis")
plt.title("Simple Plot")
plt.legend()
plt.show()
```



**SILVER OAK
UNIVERSITY**

EDUCATION TO INNOVATION

Post Practical Questions:

1. Who developed Python Programming Language?
a) Wick van Rossum
b) Rasmus Lerdorf
 c) Guido van Rossum
d) Niene Stom

2. Which type of Programming does Python support?
a) object-oriented programming
b) structured programming
c) functional programming
 d) all of the mentioned

3. Which of the following is the correct extension of the Python file?
a) .python
 b) .pl
 c) .py
d) .p

4. Which of the following is used to define a block of code in Python language?
 a) Indentation
b) Key
c) Brackets
d) All of the mentioned

5. Which of the following is not a core data type in Python programming?
a) Tuples
b) Lists
 c) Class
d) Dictionary

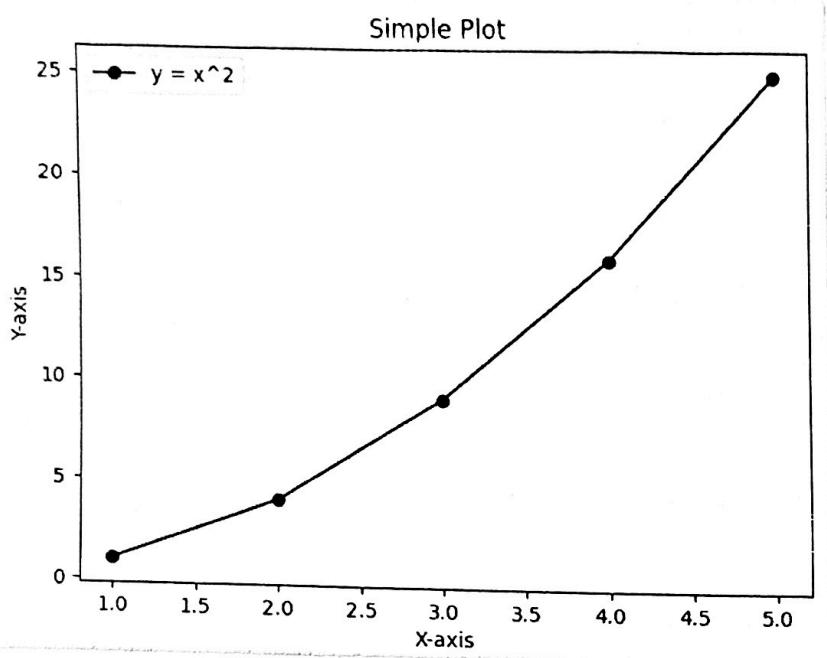
Conclusion:

In this practical, we study about different Python libraries.

Marks (out of 10)	10
Signature with Date of Completion	✓, 5/3/25

Output:

matplotlib library.





DATE: _____

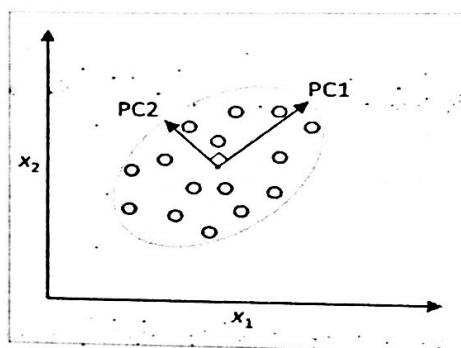
PRACTICAL-2

AIM: Using PCA reduces the dimensions of linearly inseparable data.

Description:

Principal Component Analysis (PCA) is a linear transformation technique that is widely used across different fields, most prominently for feature extraction and dimensionality reduction. Other popular applications of PCA include exploratory data analyses and de-noising of signals in stock market trading, and the analysis of genome data and gene expression levels in the field of bioinformatics.

PCA helps us to identify patterns in data based on the correlation between features. In a nutshell, PCA aims to find the directions of maximum variance in high-dimensional data and projects it onto a new subspace with equal or fewer dimensions than the original one.



The orthogonal axes (**principal components**) of the new subspace can be interpreted as the directions of maximum variance given the constraint that the new feature axes are orthogonal to each other, as illustrated in the following figure: In the preceding figure, x_1 and x_2 are the original feature axes, and **PC1** and **PC2** are the principal components.

If we use PCA for dimensionality reduction, we construct a $d \times k$ -dimensional transformation matrix W that allows us to map a sample vector x onto a new k -dimensional feature subspace



**SILVER OAK
UNIVERSITY**
EDUCATION TO INNOVATION

Program:

```
import numpy as np.  
import matplotlib.pyplot as plt.  
from sklearn.decomposition import PCA, KernelPCA.
```

```
X = np.random.rand(20, 2) * 10  
Y = np.random.randint(0, 2, 20)
```

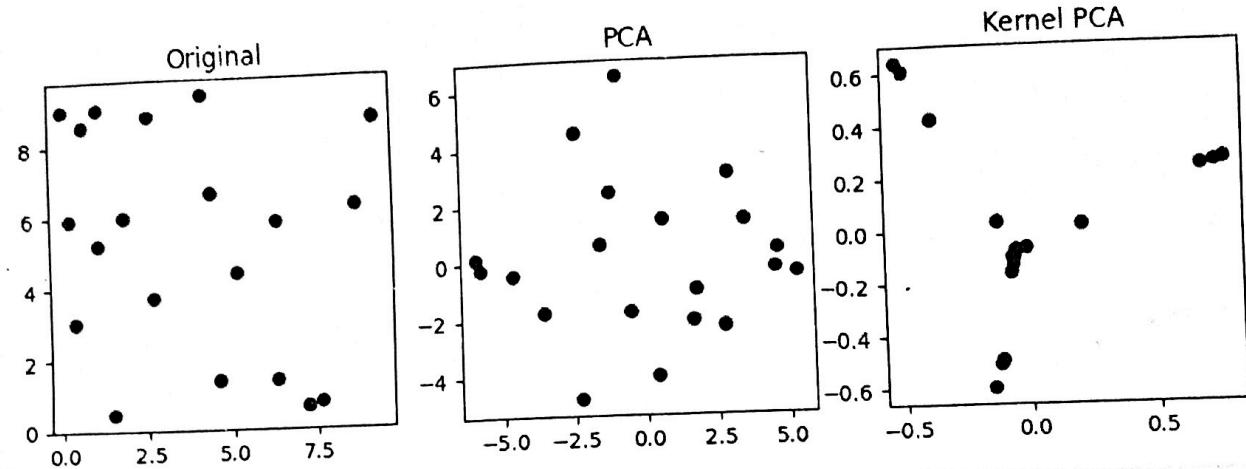
```
X_pca = PCA(2).fit_transform(X)  
X_kpca = KernelPCA(2, kernel='rbf').fit_transform(X)
```

```
fig, axes = plt.subplots(1, 3, figsize=(10, 3))  
titles = ["Original", "PCA", "Kernel PCA"]
```

```
for ax, data, title in zip(axes, [X, X_pca, X_kpca], titles):  
    ax.scatter(data[:, 0], data[:, 1], c=Y,  
               cmap='coolwarm')  
    ax.set_title(title)  
  
plt.show()
```



**SILVER OAK
UNIVERSITY**
EDUCATION TO INNOVATION



Output:-



SILVER OAK UNIVERSITY

EDUCATION TO INNOVATION

Post Practical Questions:

1. _____ is a tool which is used to reduce the dimension of the data.

- a) Principal components analysis
- b) Product Components analysis
- c) Principle Components analysis
- d) Pre Complex analysis

2. PCA reduces the dimension by finding a few _____:

- a) Hexagonal linear combination
- b) Orthogonal linear combinations
- c) Octagonal linear combination
- d) Pentagonal Linear Combination

3. PCA is a _____.

- a) Non linear method
- b) Linear method
- c) Continuous method
- d) Repeated method

4. PCA is used to find _____.

- a) Relationship between components
- b) Linear regression
- c) Linear relation
- d) Inter relation

5. There are _____ types of Supervised Learning algorithms used for classification in Machine Learning. a) 2

- b) 3
- c) 4
- d) 5

6. ICA stands for _____.

- A. Independent component analysis
- B. Inactive component analysis
- C. Intractive component analysis
- D. Inactive component Automation

Conclusion:

study about PCA & linear inseparable data

Marks (out of 10)	10
Signature with Date of Completion	MF. 5/3/25



**SILVER OAK
UNIVERSITY**
EDUCATION TO INNOVATION

DATE: _____

PRACTICAL-3

AIM: Write a program for reducing the dimensionality of sparse feature matrices.

Description:

A matrix is a two-dimensional data object made of m rows and n columns, therefore having total $m \times n$ values. If most of the elements of the matrix have 0 value, then it is called a sparse matrix.

Why to use Sparse Matrix instead of simple matrix ?

- **Storage:** There are lesser non-zero elements than zeros and thus lesser memory can be used to store only those elements.
- **Computing time:** Computing time can be saved by logically designing a data structure traversing only non-zero elements.

Example:

0 0 3 0 4
0 0 5 7 0
0 0 0 0 0
0 2 6 0 0

Representing a sparse matrix by a 2D array leads to wastage of lots of memory as zeroes in the matrix are of no use in most of the cases. So, instead of storing zeroes with non-zero elements, we only store non-zero elements. This means storing non-zero elements with **triples- (Row, Column, value)**.



**SILVER OAK
UNIVERSITY**
EDUCATION TO INNOVATION

Program:

```
import numpy as np
import scipy.sparse as sp
import matplotlib.pyplot as plt
from sklearn.decomposition import TruncatedSVD

np.random.seed(42)
X_Sparse = sp.random(100, 10, density=0.2, format='csr')

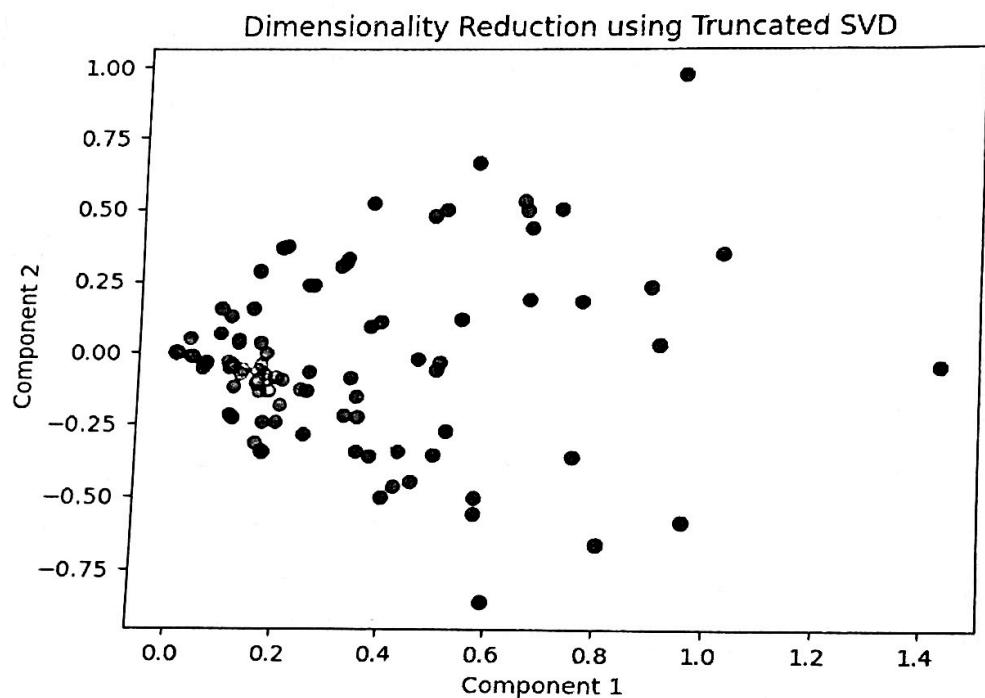
svd = TruncatedSVD(n_components=2)
X_reduced = svd.fit_transform(X_Sparse)

plt.scatter(X_reduced[:, 0], X_reduced[:, 1],
            color='blue', edgecolors='k', alpha=0.7)
plt.xlabel("Component 1")
plt.ylabel("Component 2")
plt.title("Dimensionality Reduction using Truncated SVD")
plt.show()
```



**SILVER OAK
UNIVERSITY**
EDUCATION TO INNOVATION

Output:-





**SILVER OAK
UNIVERSITY**
EDUCATION TO INNOVATION

Post Practical Questions:

1. Who coined the term Sparse Matrix?

- a) Harry Markowitz
- b) James Sylvester
- c) Chris Messina
- d) Arthur Cayley

2. Is $O(n)$ the Worst case Time Complexity for addition of two Sparse Matrix?

- a) True
- b) False

3. Which of the following is not the method to represent Sparse Matrix?

- a) Dictionary of Keys
- b) Linked List
- c) Array
- d) Heap

4. Is Sparse Matrix also known as Dense Matrix?

- a) True
- b) False

5. Which one of the following is a Special Sparse Matrix?

- a) Band Matrix
- b) Skew Matrix
- c) Null matrix
- d) Unit matrix

6. In what way the Symmetry Sparse Matrix can be stored efficiently?

- a) Heap
- b) Binary tree
- c) Hash table
- d) Adjacency List

Conclusion:

Study about ~~Sparse~~ matrix

Marks (out of 10)	09
Signature with Date of Completion	 5/3/25



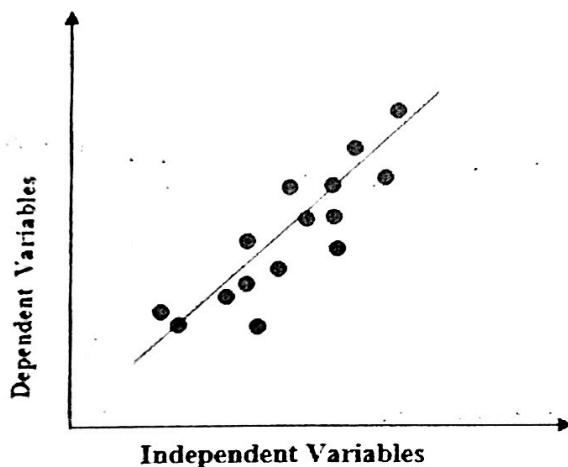
DATE: _____

PRACTICAL-4

AIM: Perform Linear Regression.

Description:

Linear regression is a quiet and simple statistical regression method used for predictive analysis and shows the relationship between the continuous variables. Linear regression shows the linear relationship between the independent variable (X-axis) and the dependent variable (Y-axis), consequently called linear regression. If there is a single input variable (x), such linear regression is called simple linear regression. And if there is more than one input variable, such linear regression is called multiple linear regression. The linear regression model gives a sloped straight line describing the relationship within the variables.



The above graph presents the linear relationship between the dependent variable and independent variables. When the value of x (independent variable) increases, the value of y (dependent variable) is likewise increasing. The red line is referred to as the best fit straight line. Based on the given data points, we try to plot a line that models the points the best.

To calculate best-fit line linear regression uses a traditional slope-intercept form.



**SILVER OAK
UNIVERSITY**
EDUCATION TO INNOVATION

```
import numpy as np
import matplotlib.pyplot as plt
from sklearn.linear_model import LinearRegression.
```

```
X = np.array([1, 2, 3, 4, 5]).reshape(-1, 1)
```

```
y = np.array([2, 4, 5, 4, 5])
```

```
model = LinearRegression()
```

```
model.fit(X, y)
```

```
y_pred = model.predict(X)
```

```
plt.scatter(X, y, color='blue', label='Actual Data')
```

```
plt.plot(X, y_pred, color='black', label='Regression Line')
```

```
plt.xlabel("X")
```

```
plt.ylabel("y")
```

```
plt.title("Linear Regression Example")
```

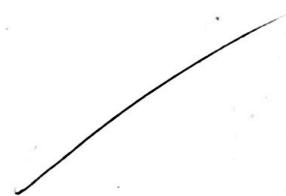
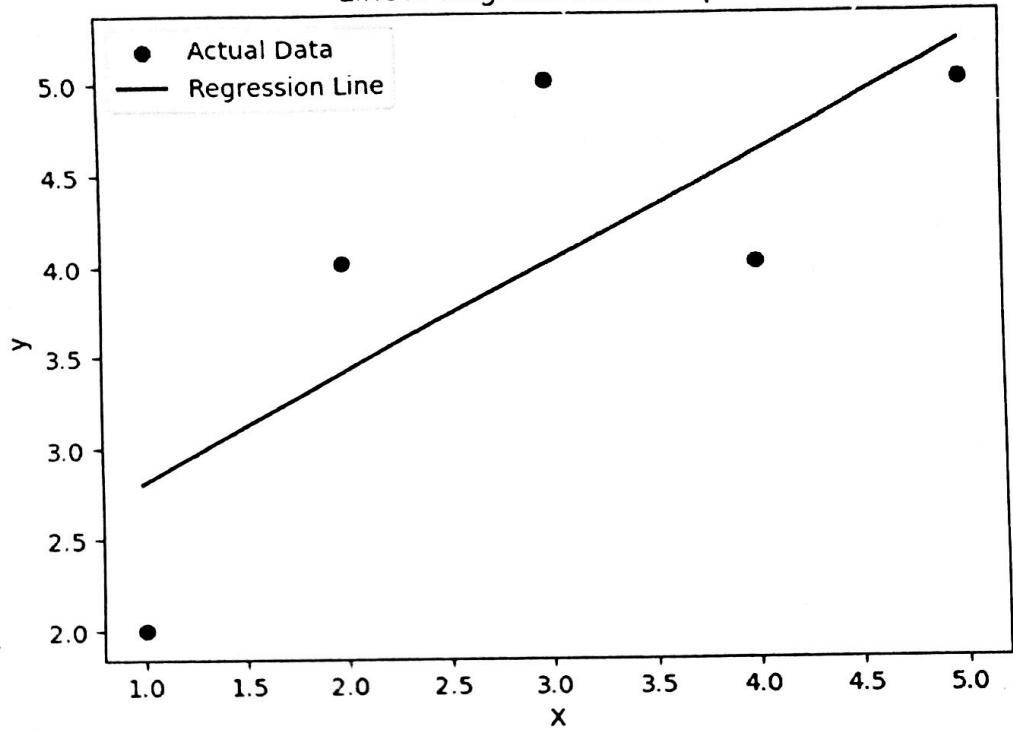
```
plt.legend()
```

```
plt.show()
```



**SILVER OAK
UNIVERSITY**
EDUCATION TO INNOVATION

Linear Regression Example





SILVER OAK UNIVERSITY

EDUCATION TO INNOVATION

DATE: _____

PRACTICAL-5

AIM: Perform Logistic Regression.

Description:

Logistic regression is a supervised learning classification algorithm used to predict the probability of a target variable. The nature of target or dependent variable is dichotomous, which means there would be only two possible classes.

In simple words, the dependent variable is binary in nature having data coded as either 1 (stands for success/yes) or 0 (stands for failure/no).

Mathematically, a logistic regression model predicts $P(Y=1)$ as a function of X . It is one of the simplest ML algorithms that can be used for various classification problems such as spam detection, Diabetes prediction, cancer detection etc.

Types of Logistic Regression

Generally, logistic regression means binary logistic regression having binary target variables, but there can be two more categories of target variables that can be predicted by it.

- **Binary or Binomial**

In such a kind of classification, a dependent variable will have only two possible types either 1 and 0. For example, these variables may represent success or failure, yes or no, win or loss etc.

- **Multinomial**

In such a kind of classification, dependent variable can have 3 or more possible unordered types or the types having no quantitative significance. For example, these variables may represent "Type A" or "Type B" or "Type C".

- **Ordinal**

In such a kind of classification, dependent variable can have 3 or more possible ordered types or the types having a quantitative significance. For example, these variables may represent "poor"



**SILVER OAK
UNIVERSITY**
EDUCATION TO INNOVATION

or "good", "very good", "Excellent" and each category can have the scores like 0,1,2,3.

Logistic Regression Assumptions

Before diving into the implementation of logistic regression, we must be aware of the following assumptions about the same –

- In case of binary logistic regression, the target variables must be binary always and the desired outcome is represented by the factor level 1.
- There should not be any multi-collinearity in the model, which means the independent variables must be independent of each other.
- We must include meaningful variables in our model.
- We should choose a large sample size for logistic regression.

Program:

```
import numpy as np
import matplotlib.pyplot as plt
from sklearn.linear_model import LogisticRegression

X = np.array([[1], [2], [3], [4], [5], [6]])
y = np.array([0, 0, 0, 1, 1, 1])

model = LogisticRegression()
model.fit(X, y)
```



**SILVER OAK
UNIVERSITY**
EDUCATION TO INNOVATION

X-test = np.linspace(0, 7, 100). reshape(-1, 1)
X-prob = model.predict_proba(X-test)[:, 1]

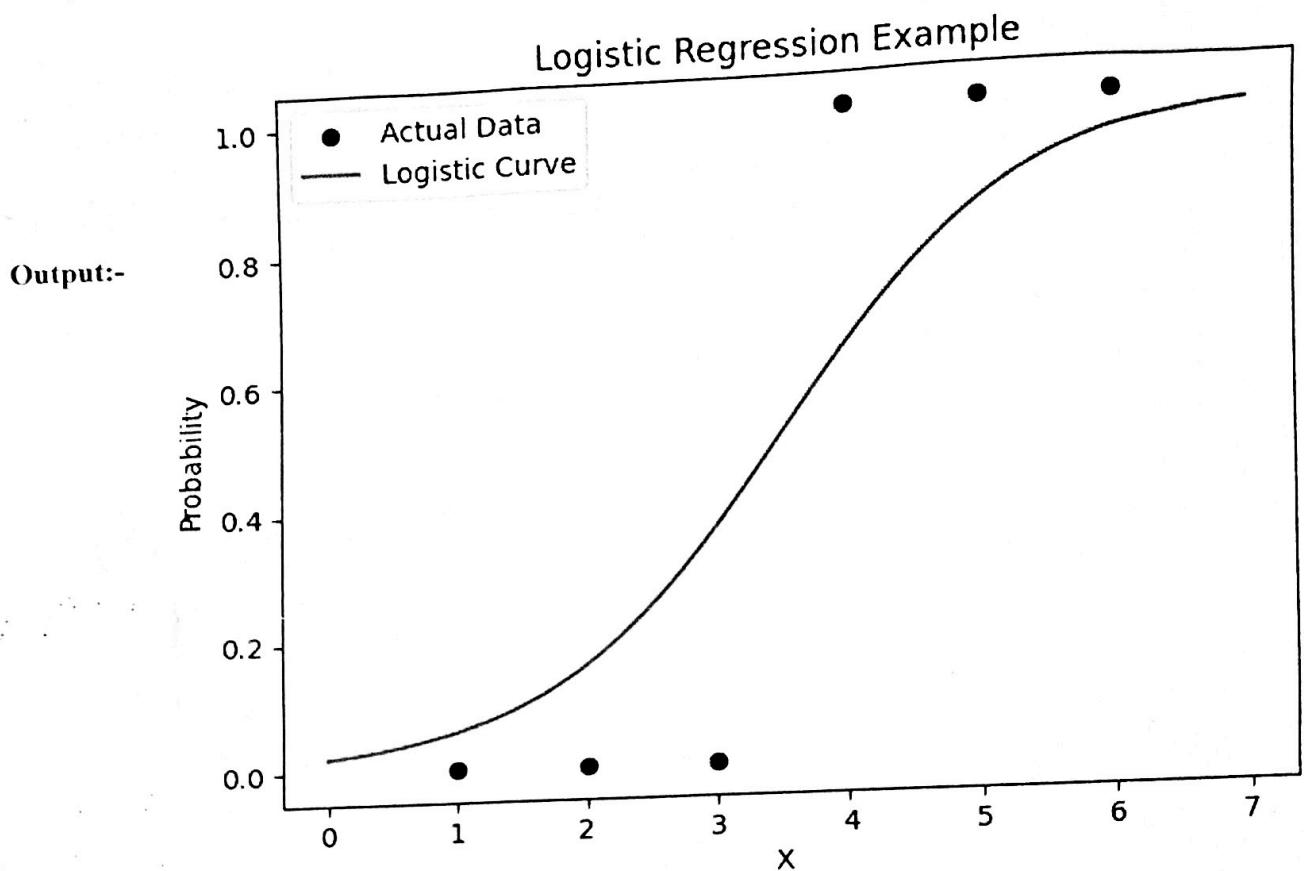
plt.scatter(X, y, color='blue', label='Actual Data')
plt.plot(X-test, y-prob, color='red', label='Logistic Curve')
plt.xlabel('X')

plt.ylabel("Probability")

plt.title("Logistic Regression Example")

plt.legend()

plt.show()



Post Practical Questions:

1. What are Regression methods?

Ans: Regression methods predict continuous values using models like Linear and Polynomial Regression. They analyze relationships between variables to make numerical predictions.



SILVER OAK UNIVERSITY

EDUCATION TO INNOVATION

2. Why do we use logistic regression?

Ans: Logistic regression is used for binary classification. Problems like spam detection or medical diagnoses. It estimates probabilities and applies a threshold to classify data.

3. Logistic Regression methods are used in Supervised or Unsupervised?

Ans: Logistic Regression is a Supervised Learning method because it learns from labeled data. It maps input features to a probability output using a sigmoid function.

4. Which type of data set is being used in logistic regression?

Ans: Logistic regression uses categorical or binary data, such as Yes/No or 0/1 outcomes. It helps classify data points into distinct categories based on input features.

Conclusion:

In this practical, we learnt to perform logistic Regression.

Marks (out of 10)	
Signature with Date of Completion	



DATE: _____

PRACTICAL-6

AIM: Write a program to implement the naïve Bayesian classifier for a sample training data set stored as a .CSV file. Compute the accuracy of the classifier, considering few test data sets.

Description:

A Bayesian classifier is based on the idea that the role of a (natural) class is to predict the values of features for members of that class.

A Bayesian classifier is a probabilistic model where the classification is a latent variable that is probabilistically related to the observed variables. Classification then becomes inference in the probabilistic model.

The simplest case is the naive Bayesian classifier, which makes the independence assumption that the

input features are conditionally independent of each other given the classification. The independence of the naive Bayesian classifier is embodied in a particular belief network where the features are the nodes, the target variable (the classification) has no parents, and the classification is the only parent of each input feature. This belief network requires the probability distributions $P(Y)$ for the target

feature Y and $P(X_i|Y)$ for each input feature X_i . For each example, the prediction can be computed by conditioning on observed values for the input features and by querying the classification.

Given an example with inputs $X_1=v_1, \dots, X_k=v_k$, Bayes' rule is used to compute the posterior probability distribution of the example's classification, Y :

$$P(Y | X_1=v_1, \dots, X_k=v_k)$$

$$= (P(X_1=v_1, \dots, X_k=v_k | Y) \times P(Y)) / (P(X_1=v_1, \dots, X_k=v_k))$$

$$= (P(X_1=v_1 | Y) \times \dots \times P(X_k=v_k | Y) \times P(Y)) / (\sum_Y P(X_1=v_1 | Y) \times \dots \times P(X_k=v_k | Y) \times P(Y))$$

where the denominator is a normalizing constant to ensure the probabilities sum to 1. The



**SILVER OAK
UNIVERSITY**
EDUCATION TO INNOVATION

denominator does not depend on the class and, therefore, it is not needed to determine the most likely class.

Program:

```
import pandas as pd  
import numpy as np  
np.random.seed(12)
```

```
num_samples = 10
```

```
enrollment_numbers = [f"ENR{1000+i}" for i in range(num_samples)]
```

```
names = np.random.choice(["Alice", "Bob", "Charlie",  
"David", "Emma", "Frank", "Grace", "Henry", "Ivy",  
"Jack"], num_samples, replace=False)
```

```
degrees = np.random.choice(["B-TECH", "B.Sc",  
"BBA", "BCA", "BA"], num_samples)
```

```
df = pd.DataFrame({  
    "Enrollment Number": enrollment_numbers,  
    "Name": names,  
    "Degree": degrees  
})
```



**SILVER OAK
UNIVERSITY**
EDUCATION TO INNOVATION

```
df.to_csv("student.csv", index=False)
```

Point("Random students.csv file created successfully
with 10 rows!")

OUTPUT:

Random students.csv file created successfully with
10 rows!

students.csv

Enrollment Number, Name, Degree.

ENR1000, Frank, BA

ENR1001, Ivy, B.Sc

ENR1002, Henry, BBA

ENR1003, Alice, BCA

ENR1004, Emma, BBA

ENR1005, Jack, B.TECH

ENR1006, David, B.TECH

ENR1007, Charlie, BA

ENR1008, Bob, BBA

ENR1009, Grace, B.Sc.



**SILVER OAK
UNIVERSITY**
EDUCATION TO INNOVATION

Post Practical Questions:

1.What is Bayes' Theorem?

Ans: Bayes' Theorem → It is a mathematical formula used to update probabilities based on new evidence.

It is given by:

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

where $P(A|B)$ is the posterior probability of event A given B.

2.Application of Bayesian Classifier?

Ans:

Applications of Bayesian classifier.

It is used for:

- ① Spam filtering.
- ② Medical diagnosis
- ③ Sentiment analysis
- ④ Recommendation systems.
- ⑤ Fraud detection.



SILVER OAK UNIVERSITY

EDUCATION TO INNOVATION

3.What is the conditional probability?

Ans: Conditional Probability: It is the probability of an event occurring given that another event has already occurred, expressed as $P(A|B) \Rightarrow \frac{P(A \cap B)}{P(B)}$

4.What are the common conditions for Bayesian classifiers to be required before applying on data set?

common conditions for Bayesian classifiers:

- ① Data should be conditionally independent
- ② Proper prior probabilities should be assigned.
- ③ The dataset should be sufficiently large for accurate probability estimation.
- ④ Features should be relevant and contribute to classification.

Conclusion:

In this practical, we learnt about program to implement the naïve Bayesian for sample training data set.

Marks (out of 10)	
Signature with Date of Completion	



DATE: _____

PRACTICAL-7

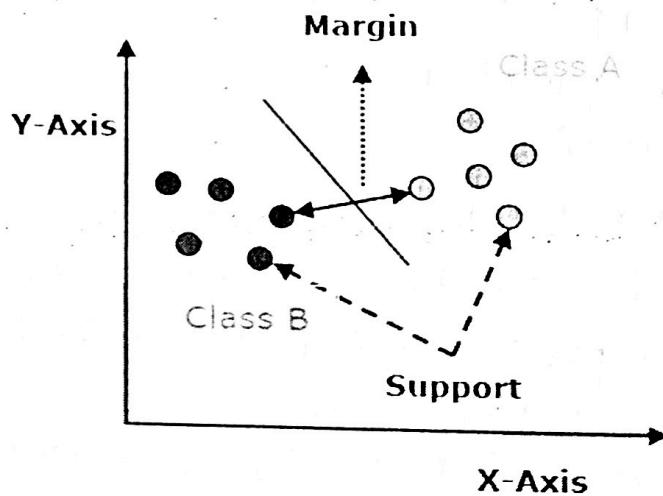
AIM: Write a program to implement SVM algorithm to classify the iris dataset.

Description:

Support vector machines (SVMs) are powerful yet flexible supervised machine learning algorithms which are used both for classification and regression. But generally, they are used in classification problems. In the 1960s, SVMs were first introduced but later they got refined in 1990. SVMs have their unique way of implementation as compared to other machine learning algorithms. Lately, they are extremely popular because of their ability to handle multiple continuous and categorical variables.

Working of SVM

An SVM model is basically a representation of different classes in a hyperplane in multidimensional space. The hyperplane will be generated in an iterative manner by SVM so that the error can be minimized. The goal of SVM is to divide the datasets into classes to find a maximum marginal hyperplane (MMH).



The followings are important concepts in SVM –

- **Support Vectors** – Datapoints that are closest to the hyperplane is called support



**SILVER OAK
UNIVERSITY**

EDUCATION TO INNOVATION

vectors. Separating line will be defined with the help of these data points.

- **Hyperplane** – As we can see in the above diagram, it is a decision plane or space which is divided between a set of objects having different classes.
- **Margin** – It may be defined as the gap between two lines on the closest data points of different classes. It can be calculated as the perpendicular distance from the line to the support vectors. Large margin is considered as a good margin and small margin is considered as a bad margin.

The main goal of SVM is to divide the datasets into classes to find a maximum marginal hyperplane (MMH) and it can be done in the following two steps –

- First, SVM will generate hyperplanes iteratively that segregates the classes in the best way.
- Then, it will choose the hyperplane that separates the classes correctly.

Program:

```
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
from sklearn import datasets
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import StandardScaler
from sklearn.svm import SVC
from sklearn.metrics import accuracy_score,
classification_report, confusion_matrix

iris = datasets.load_iris()
X = iris.data
y = iris.target

X_train, X_test, y_train, y_test = train_test_split(X, y,
test_size=0.2, random_state=42)
```

scaler = StandardScaler()

X_train = scaler.fit_transform(X_train)

X_test = scaler.transform(X_test)

SVM-classifier =



**SILVER OAK
UNIVERSITY**
EDUCATION TO INNOVATION

SVC(kernel='linear', C=1.0)

SVM-classifier.fit(X_train, y_train)

y_pred = SVM-classifier.predict(X_test)

accuracy = accuracy_score(y_test, y_pred)

print(f'Accuracy: {accuracy:.2f}%')

print('\nClassification Report:\n', classification_report(y_test, y_pred))

conf_matrix = confusion_matrix(y_test, y_pred)

conf_matrix = confusion_matrix(y_test, y_pred)

sns.heatmap(conf_matrix, annot=True, cmap='Blues', fmt='d', xticklabels=iiris.target_names, yticklabels=iiris.target_names)

plt.xlabel("Predicted")

plt.ylabel("Actual")

plt.title("Confusion Matrix")

plt.show()

Output:

Accuracy: 0.97

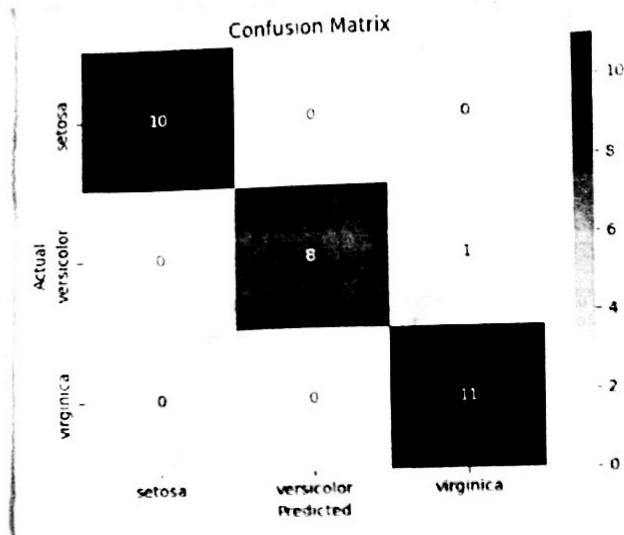
Classification Report:

	Precision	recall	f1-score	support
0	1.00	1.00	1.00	10
1	1.00	0.89	0.94	9
2	0.92	1.00	0.96	11
accuracy	0.97	-	0.97	30
macro avg	0.97	0.96	0.97	30
weighted avg	0.97	0.97	0.97	30



**SILVER OAK
UNIVERSITY**
EDUCATION TO INNOVATION

Output:-



Post Practical Questions:

1. What does SVM stand for?

Ans: SVM stands for:

Support Vector machine.

2. What is the application of SVM?

Ans: Application of SVM:

- ① Text and image classification
- ② Handwriting recognition.
- ③ Bioinformatics
- ④ Face detection
- ⑤ Spam filtering ..



SILVER OAK UNIVERSITY

EDUCATION TO INNOVATION

3. Characteristics of SVM?

Ans: Characteristics of SVM.

- Works well with high-dimensional data.
- Effective for both linear and non-linear classification
- Uses Kernel trick to handle complex data.
- Maximizes margin for better generalization.
- Robust against overfitting, especially in high-dimensional spaces.

4. SVM is used for _____ machine learning method.

Ans:

SVM is used for Supervised machine learning method.

Conclusion:

In this practical, we learnt about program to implement SVM algorithm to classify the iris dataset.

Marks (out of 10)	
Signature with Date of Completion	



DATE: _____

PRACTICAL-8

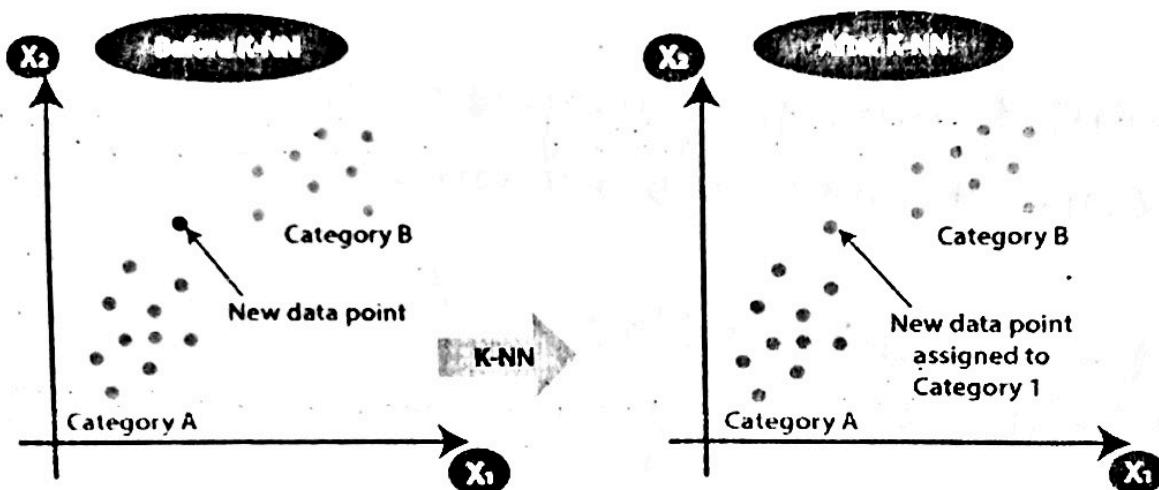
AIM: Write a program to implement Random Forest to classify the iris data set. Print both correct and wrong predictions.

Description:

K-NN algorithm stores all the available data and classifies a new data point based on the similarity.

This means when new data appears then it can be easily classified into a well suited category by using K-NN algorithm.

Suppose there are two categories, i.e., Category A and Category B, and we have a new data point x_1 , so this data point will lie in which of these categories. To solve this type of problem, we need a K-NN algorithm. With the help of K-NN, we can easily identify the category or class of a particular dataset. Consider the below diagram:



KNN Algorithm works as follow: -

Step-1: Select the number K of the neighbors

Step-2: Calculate the Euclidean distance of K number of neighbors

Step-3: Take the K nearest neighbors as per the calculated Euclidean distance.

Step-4: Among these k neighbors, count the number of the data points in each category.

Step-5: Assign the new data points to that category for which the number of the neighbor is maxi.



SILVER OAK
UNIVERSITY
EDUCATION TO INNOVATION

Program:

```
from sklearn.datasets import load_iris
from sklearn.ensemble import RandomForestClassifier
from sklearn.model_selection import train_test_split

iris = load_iris()
X = iris.data
y = iris.target
labels = iris.target_names

X_train, X_test, y_train, y_test = train_test_split(
    (X, y, test_size=0.3, random_state=0))

model = RandomForestClassifier()
model.fit(X_train, y_train)

Prediction = model.predict(X_test[:5])
actual = y_test[:5]

print("Correct Predictions:")
for i in range(5):
    if predictions[i] == actual[i]:
        print(f"Sample {i}: Predicted = {labels}
```



**SILVER OAK
UNIVERSITY**

EDUCATION TO INNOVATION

[predictions[i]]}, Actual = {labels[actual[i]]}"))

Point(" \n Wrong Predictions: ")

for i in range(5):

if predictions[i] != actual[i]:

Print(f "Sample {i}: Predicted = {labels
[predictions[i]]}, Actual = {labels
[actual[i]]}").



**SILVER OAK
UNIVERSITY**
EDUCATION TO INNOVATION

Output:-

Correct Predictions:

Sample 0: Predicted = virginica, Actual = virginica

Sample 1: Predicted = versicolor, Actual = versicolor

Sample 2: Predicted = setosa, Actual = setosa

Sample 3: Predicted = virginica, Actual = virginica

Sample 24: Predicted = setosa, Actual = setosa.

Wrong Predictions:

Post Practical Questions:

1. Why is the K Nearest Neighbor algorithm a lazy learning algorithm?

Ans:

K Nearest Neighbors algorithm is a lazy learning algorithm because it doesn't learn during training. It just stores data and delays computation until prediction.

2. Differentiate Clustering and Classification.

Ans:

Clustering

Classification

① Type: Unsupervised

① Type: Supervised.

② Labels are not used

② Labels are used.

③ Goal: Group similar data.

③ Goal: Assign to know categories.



3. Mention classification algorithms.

Ans: Classification Algorithm:

- ① K-Nearest Neighbors (K-NN)
- ② Decision Tree
- ③ Random Forest
- ④ Logistic Regression
- ⑤ Support Vector Machine (SVM)
- ⑥ Naïve Bayes

4. Define Pruning :

Ans:

Pruning is the process of removing unnecessary branches from a decision tree to reduce overfitting and improve accuracy.

Conclusion:

In this practical, we learnt about program to implement Random Forest to classify the iris data set.

Marks (out of 10)	
Signature with Date of Completion	



DATE: _____

PRACTICAL-9

AIM: Implement K means Clustering algorithm on dataset. Visualise the results for the same.

Description:

K-Means Clustering is an Unsupervised Learning algorithm, which groups the unlabeled dataset into different clusters. Here K defines the number of predefined clusters that need to be created in the process, as if $K=2$, there will be two clusters, and for $K=3$, there will be three clusters, and so on.

It is an iterative algorithm that divides the unlabeled dataset into k different clusters in such a way that each dataset belongs to only one group that has similar properties.

It allows us to cluster the data into different groups and a convenient way to discover the categories of groups in the unlabeled dataset on its own without the need for any training.

It is a centroid-based algorithm, where each cluster is associated with a centroid. The main aim of this algorithm is to minimize the sum of distances between the data point and their corresponding clusters.

The algorithm takes the unlabeled dataset as input, divides the dataset into k-number of clusters, and repeats the process until it does not find the best clusters. The value of k should be predetermined in this algorithm.

The k-means clustering algorithm mainly performs two tasks:

- a. Determines the best value for K center points or centroids by an iterative process.
- b. Assigns each data point to its closest k-center. Those data points which are near to the particular k-center, create a cluster.

The below diagram explains the working of the K-means Clustering Algorithm:



```
from sklearn.datasets import load_iris  
from sklearn.cluster import KMeans  
import matplotlib.pyplot as plt  
from sklearn.decomposition import PCA
```

```
iris = load_iris()
```

```
X = iris.data
```

```
Kmeans = KMeans(n_clusters=3, random_state=0)  
Kmeans.fit(X)
```

```
labels = Kmeans.labels_
```

```
pca = PCA(n_components=2)
```

```
X_reduced = pca.fit_transform(X)
```

```
plt.scatter(X_reduced[:, 0], X_reduced[:, 1],  
            c=labels, cmap='viridis')
```

```
plt.title("K-Means Clustering (Iris Data Set)")
```

```
plt.xlabel("PCA Feature 1")
```

```
plt.ylabel("PCA Feature 2")
```

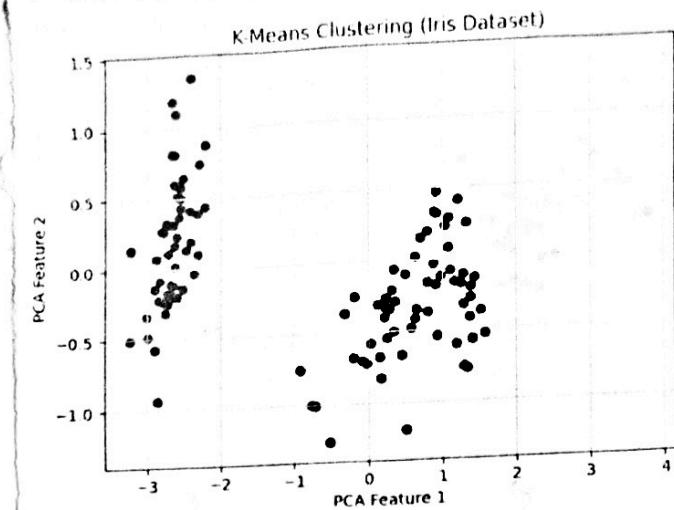
```
plt.grid(True)
```

```
plt.show()
```



**SILVER OAK
UNIVERSITY**
EDUCATION TO INNOVATION

OUTPUT:



Post Practical Questions:

1. What is Cluster?

Ans:

A cluster is a collection of data points that are grouped together based on similarity. Points in the same cluster are more similar to each other than to those in other clusters.

2. Why is K-Mean actually used?

Ans:

- K-Means is used to automatically group unlabeled data into k distinct clusters.
- It helps in identifying hidden patterns & structures in large datasets by grouping similar data point together.



SILVER OAK UNIVERSITY

EDUCATION TO INNOVATION

3. What will be the output if we apply K-Mean clustering Technique on any database?

Ans:

- The output is a set of k-clusters with each data point assigned to one of them.
- It also gives the centroid of each cluster and the cluster label for each data point.

4. Mention Clustering Methods:

Ans:

- ① K-Means clustering - centroid-based & fast
- ② Hierarchical clustering - builds tree of clusters
- ③ DBSCAN - density-based, good for irregular shapes
- ④ Mean shift - centroid-based, does not need pre-defined K.

Conclusion:

In this practical, we learnt about implementing K-means clustering algorithm on dataset.

Marks (out of 10)	
Signature with Date of Completion	



DATE: _____

PRACTICAL-10

AIM: Build an Artificial Neural Network by implementing the Backpropagation algorithm and test the same using appropriate data sets.

Description:

What is Artificial Neural Networks?

A neural network is a group of connected I/O units where each connection has a weight associated with its computer programs. It helps you to build predictive models from large databases. This model builds upon the human nervous system. It helps you to conduct image understanding, human learning, computer speech, etc.

What is Backpropagation?

- Backpropagation is the essence of neural network training. It is the method of fine-tuning the weights of a neural network based on the error rate obtained in the previous epoch (i.e., iteration). Proper tuning of the weights allows you to reduce error rates and make the model reliable by increasing its generalization.
- Backpropagation in a neural network is a short form for “backward propagation of errors.” It is a standard method of training artificial neural networks. This method helps calculate the gradient of a loss function with respect to all the weights in the network.

Backpropagation Algorithm :

1. Inputs X, arrive through the preconnected path
2. Input is modeled using real weights W. The weights are usually randomly selected.
3. Calculate the output for every neuron from the input layer, to the hidden layers, to the output layer.
4. Calculate the error in the outputs

$$\text{Error}_B = \text{Actual Output} - \text{Desired Output}$$

5. Travel back from the output layer to the hidden layer to adjust the weights such that the error is decreased.
6. Keep repeating the process until the desired output is achieved



Program:

```
from sklearn.datasets import load_iris  
from sklearn.model_selection import train_test_  
split.  
from sklearn.neural_network import MLPClassifier  
from sklearn.metrics import accuracy_score.
```

```
iris = load_iris()
```

```
X = iris.data
```

```
y = iris.target -
```

```
X-train, X-test, y-train, y-test = train-test-  
split (x, y, test_size = 0.2, random_state=1)
```

```
model = MLPClassifier(hidden_layer_sizes=(10,),  
max_iter=1000, random_state=1)
```

```
model.fit(X-train, y-train)
```

```
y-pred = model.predict(X-test)
```

```
Print("Test Accuracy:", accuracy_score(y-test,  
y-pred))
```



**SILVER OAK
UNIVERSITY**
EDUCATION TO INNOVATION

Point ("In Sample Predictions:")

for i in range (5):

 Print (f"Sample {i+1}: Predicted = {iris.target_names[y-pred[i]]}, Actual = {iris.target_names[y-test[i]]}")



Output:

Test Accuracy: 0.9333333333

Sample Predictions:

Sample 1: Predicted = setosa, Actual = setosa
Sample 2: Predicted = versicolor, Actual = versicolor
Sample 3: Predicted = versicolor, Actual = versicolor
Sample 4: Predicted = setosa, Actual = setosa
Sample 5: Predicted = virginica, Actual = virginica..

Post Practical Questions:

1. What do you mean by Back propagation?

Ans:

- Backpropagation is a training algorithm used in neural networks to minimize error by adjusting the weights.
- It calculates the error at the output layer & distributes it back through the network layers to update the weights using gradient descent.

2. Define: Deep Learning.

Ans:

- Deep learning is a branch of machine learning that uses artificial neural networks with multiple hidden layers.
- It is capable of learning complex patterns & representations from large amounts of data.

3. Advantages of NN?

- ① Can model non-linear relationships.
- ② Adaptable to wide range of data types.



Ans:

- ③ Can learn directly from raw data.
- ④ Performs well with large datasets.
- ⑤ Robust to noise and incomplete data.

4. Factors which influence Back propagation training?

Ans:

- ① Learning Rate.
- ② Number of Hidden layers / Neurons.
- ③ Data Quality.
- ④ Activation Function
- ⑤ Weight Initialization
- ⑥ Epochs & Batch Size.

Conclusion:

In this practical, we learnt about building ^{on} Artificial Neural Network by implementing Backpropagation algorithm.

Marks (out of 10)	
Signature with Date of Completion	