CMR TECHNICAL CAMPUSUGC AUTONOMOUS

Kandlakoya(V), Medchal Road, Hyderabad – 501 401

Accredited by NBA and NAAC with A Grade

Approved by AICTE, New Delhi and Affiliated to JNTU, Hyderabad

DEPARTMENT OF CSE (AI & ML)

MACHINE LEARNING LAB

COURSE CODE: 20AI703PC

R20

Prepared By:

Ms.Mamatha.B
Assistant Professor

Course Objective: The objective of this lab is to get an overview of the various machinelearning techniques and can able to demonstrate them using python.

Course Outcomes: After the completion of the course the student can able to:

1. understand complexity of Machine Learning algorithms and their limitations;
2. understand modern notions in data analysis-oriented computing;
3. be capable of confidently applying common Machine Learning algorithms in practice and implementing their own;
4. Be capable of performing experiments in Machine Learning using real-world data.

Sl.No	List of Experiments:
1.	The probability that it is Friday and that a student is absent is 3 %. Since there are 5 school days in a week, the probability that it is Friday is 20 %. What is theprobability that a student Is absent given that today is Friday? Apply Baye‟s rule in python to get the result. (Ans: 15%)
2.	Extract the data from database using python
3.	Implement k-nearest neighbours classification using python
4.	Given the following data, which specify classifications for nine combinations of VAR1 andVAR2 predict a classification for a case where VAR1=0.906 and VAR2=0.606, using the result of kmeans clustering with 3 means (i.e., 3 centroids) VAR1 VAR2 CLASS 1.713 1.586 0 0.180 1.786 1 0.353 1.240 1 0.940 1.566 0 1.486 0.759 1 1.266 1.106 0 1.540 0.419 1 0.459 1.799 1 0.186 1
5.	The following training examples map descriptions of individuals onto high,Medium and low credit-worthiness. medium skiing design single twenties no ->highRiskhigh golf trading married forties yes ->lowRisk low speedway transport married thirties yes ->medRisk medium football banking single thirties yes ->lowRisk high flying media married fifties yes ->highRisk low football security single twenties no ->medRiskmedium golf media single thirties yes ->medRisk medium golf transport married forties yes ->lowRiskhigh skiing banking single thirties yes ->highRisk low golf unemployed married forties yes ->highRisk Input attributes are (from left to right) income, recreation, job, status, age-group, home-owner. Find the unconditional probability of `golf' and the conditional probability of `single' given `medRisk'in the dataset
6.	Implement linear regression using python.
7.	Implement Naïve Bayes theorem to classify the English text
8.	Implement an algorithm to demonstrate the significance of genetic algorithm
9.	Implement the finite words classification system using Back-propagation algorithm
10.	Implementing FIND-S algorithm using python
11.	Implementing Candidate Elimination algorithm using python

EXPERIMENT NO:1

1. The probability that it is Friday and that a student is absent is 3 %. Since there are 5 school days in a week, the probability that it is Friday is 20 %. What is theprobability that a student is absent given that today is Friday? Apply Baye's rule in python to get the result. (Ans: 15%)

Aim:

To Find the the probability that a student is absent given that today is Friday from givendata with Baye‟s rule in python.

Theory:

P (Today is Friday)=0.2
P(B)=0.2

(It is Friday∩student is absent)= P(A∩B)=0.03
P(A | B) = P(A and B) / P(B)

it is required to find
P(student is absent |today is Friday) P(A|B)
The formula for obtaining the conditional probability of event A, given the event B has occurred is as follows:

P(A|B)=P(A∩B)/P(B)
Thus the required probability is as follows:

P(student is absent| today is Friday)=P(It is Friday∩student is absent)/P(Today is Friday)
=0.03/0.2=0.15

The answer is 0.15

Source Code

```
# calculate P(A|B) given P(A and B) and P(B)
def bayes_theorem(p_a_b, p_b):
        # calculate P(A|B) = P(A and B) / P(B)
        p_a_given_b = (p_a_b) / p_b
        return p_a_given_b#
 P(A and B)
p_a_b = 0.03

p_b = 0.20

# calculate P(A|B)

result = bayes_theorem(p_a_b, p_b) #
summarize
print('P(A|B) = %.f%%' % (result * 100))
```

Output:

```
# EXPERIMENT NO:1
def bayes_theorem(p_a_b, p_b):
    p_a_given_b = (p_a_b) / p_b
    return p_a_given_b
# P(A and B)
p_a_b = 0.03
# P(B)
p_b = 0.20
# calculate P(A|B)
result = bayes_theorem(p_a_b, p_b)
# summarize
print('P(A|B) = %.f%%' % (result * 100))
```

```
P(A|B) = 15%
```

Sample Viva Questions:

1. **What is Bayes rule?**

   We can simply define Bayes rule like this. Let A1, A2, …, An be a set of mutually exclusive events thattogether form the sample space S. Let B be any event from the same sample space, such that P(B) > 0.
   Then, P(Ak | B) = P(Ak ∩ B) / P(A1 ∩ B) + P(A2 ∩ B) + . . . + P(An ∩ B)

2. **What is Bayes classifier?**

   Naive Bayes classifiers are a family of simple probabilistic classifiers based on applying Bayes‟ theorem with strong (naive) independence assumptions between the features inmachine learning. Basically we can use above theories and equations for classification problem.

3. **What are Bayesian Networks (BN) ?**

   Bayesian Network is used to represent the graphical model for probability relationship among a set of variables. Bayes‟ theorem is a way to figure out conditional probability.

4. **Can you give any real time example using Bayes' Theorem (liver disease)**

   A could mean the event "Patient has liver disease." Past data tells you that 10% of patientsentering your clinic have liver disease. P(A) = 0.10.
   B could mean the litmus test that "Patient is an alcoholic." Five percent of the clinic‟spatients arealcoholics. P(B) = 0.05.
   You might also know that among those patients diagnosed with liver disease, 7% are alcoholics.This is your **B|A:** the probability that a patient is alcoholic, given that they haveliver disease, is 7%.

   Bayes‟ theorem tells you:
   P(A|B)=(0.07*0.1)/0.05=0.14

5. **Given the following statistics, what is the probability that a woman hascancer if she has apositive mammogram result?**

   One percent of women over 50 have breast cancer.
   Ninety percent of women who have breast cancer test positive on mammograms. Eight percent of women will have false positives.
   Step 1: Assign events to A or X. You want to know what a woman‟s probability of having cancer is, given a positive mammogram. For this problem, actually having cancer is A and a positive test result isX.
   Step 2: List out the parts of the equation (this makes it easier to work theactual equation):P(A)=0.01
   P(~A)=0.99

P(X|A)=0.9
P(X|~A)=0.08
Step 3: Insert the parts into the equation and solve. Note that as this is a medical test,we‟re using theform of the equationfrom example #2:
(0.9 * 0.01) / ((0.9 * 0.01) + (0.08 * 0.99) = 0.10.
The probability of a woman having cancer, given a positive test result, is 10%.

EXPERIMENT NO: 2

2. Extract the data from database using python

Aim:

To extract the data from database using python

Theory:

1. **Connect to MySQL from Python**

   Refer to Python MySQL database connection to connect to MySQL database from Python using MySQL Connector module

2. **Define a SQL SELECT Query**

   Next, prepare a SQL SELECT query to fetch rows from a table. You can select all or limited rows based on your requirement. If the where condition is used, then it decides the number of rows to fetch.
   For example, SELECT col1, col2,…colnN FROM MySQL_table WHERE id = 10;. This will return row number 10.

3. **Get Cursor Object from Connection**

   Next, use a connection.cursor() method to create a cursor object. This method creates anew MySQLCursor object.

4. **Execute the SELECT query using execute() method**

   Execute the select query using the cursor.execute() method.

5. **Extract all rows from a result**

   After successfully executing a Select operation, Use the fetchall() method of a cursor object to get all rows from a query result. it returns a list of rows.

6. **Iterate each row**

   Iterate a row list using a for loop and access each row individually (Access each row‟s column data using a column name or index number.)

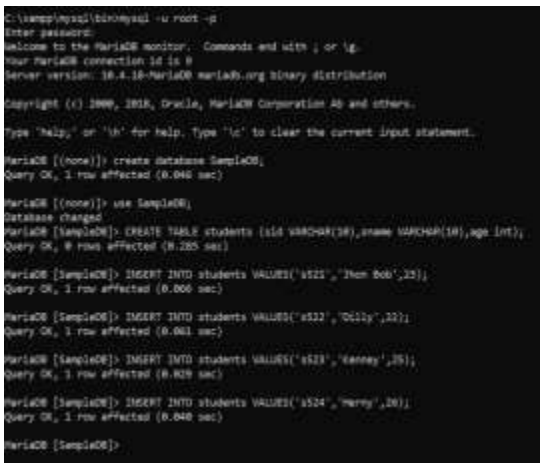
7. **Close the cursor object and database connection object**

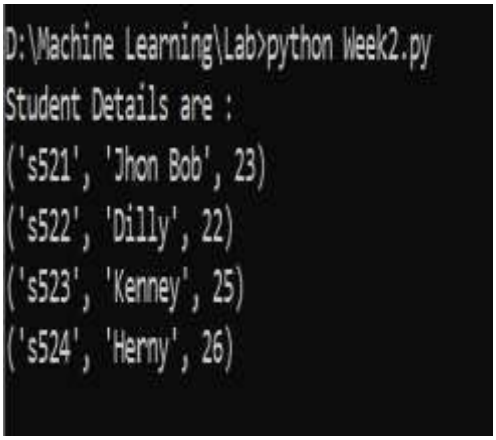
   use cursor.clsoe() and connection.clsoe() method to close open connections after your work completes.

Source Code:

```
import mysql.connector

# Step 1: Establish a connection to the MySQL database
try:
    conn = mysql.connector.connect(
        host='localhost',      # Replace with your MySQL host
        user='root',        # Replace with your MySQL username
        password='root', # Replace with your MySQL password
        database='abc'       # Replace with your MySQL database name
        )

    # Step 2: Create a cursor
    cursor = conn.cursor()

    # Step 3: Execute SQL query
    query = 'SELECT * from students;' # Replace with your actual table namecursor.execute(query)

    # Step 4: Fetch data
    data = cursor.fetchall()

    # Step 5: Close the cursor and connection
    cursor.close()
    conn.close()

    # Process the data as needed
    for row in data:
        print(row)  # Replace this with your desired data processing logic


    except mysql.connector.Error as
    e:print("Error:", e)
```

Output:





Sample Viva Questions:

1. **What are the methods to fetch data returned by a cursor.execute( )**

   - cursor.fetchall() to fetch all rows
   - cursor.fetchone() to fetch a single row
   - cursor.fetchmany(SIZE) to fetch limited rows

2. **How to extract data from database using Python?**

You can fetch data from MYSQL using the fetch() method provided by the mysql-connector-python. The cursor. MySQLCursor class provides three methods namely fetchall(), fetchmany() and, fetchone() where, The fetchall() method retrieves all the rows in the result set of a query and returns them as list of tuples.

3. **What data do you need to connect to the database in Python?**

To create a connection between the MySQL database and Python, the connect() method of mysql. connector module is used. We pass the database details like HostName, username,and the password in the method call, and then the method returns the connection object.

4. **How do we extract data from database?**

To extract data from RDBMS, you need to write SQL statements which include _clauses, table names, field names, logical expressions,_ etc. As shown in the below table, there is abasic order for clauses you need to follow while writing SQL statements for data extraction.

5. **Write a script to connect to MySql database using Python.**

```
#!/usr/bin/python
 importMySQLdb
 # Open database connection
 db =MySQLdb.connect("localhost","username","password","databasename" )Prepare a
 cursor object using cursor () method cursor =db.cursor()
 execute SQL query using execute() method. cursor.execute("SELECT VERSION()")
 Fetch a single row using fetchone() method. data =cursor.fetchone() print"Database
 version : %s "%data
 disconnect from server db.close()
```

If a connection is established with the datasource, then a Connection Object is returnedsaved into db for further use, otherwise db is set to None. Next, db object is used to create a cursor object, which in turn is used to execute SQL queries. Finally, before coming out, it ensures that database connection is closed and resources are released

EXPERIMENT NO: 3

3. Implement k-nearest neighbours classification using python

Aim:

To implement k-nearest neighbours classification using python

Theory:

- K-Nearest Neighbors is one of the most basic yet essential classification algorithms in Machine Learning. It belongs to the supervised learning domain and finds intense application in pattern recognition, data mining and intrusion detection.
- It is widely disposable in real-life scenarios since it is non-parametric, meaning, it does notmake any underlying assumptions about the distribution of data.

Algorithm

Input: Let m be the number of training data samples. Let p be an unknown point.Method:

1. Store the training samples in an array of data points arr[]. This means each elementof this array represents a tuple (x, y).
2. for i=0 to m

Calculate Euclidean distance d(arr[i], p).

3. Make set S of K smallest distances obtained. Each of these distances correspond toan already classified data point.
4. Return the majority label among S.

Source Code:

```
# import the required packages
from sklearn.model_selection import train_test_split
from sklearn.neighbors import KNeighborsClassifier
from sklearn import datasets
# Load dataset
iris=datasets.load_iris()
print("Iris Data set
loaded...")
# Split the data into train and test samples
x_train, x_test, y_train, y_test = train_test_split(iris.data,iris.target,test_size=0.1)
print("Dataset is split into training and testing...")
print("Size of trainng data and its label",x_train.shape,y_train.shape)
print("Size of trainng data and its label",x_test.shape, y_test.shape) #
Prints Label no. and their names
for i in range(len(iris.target_names)):
print("Label", i , "-",str(iris.target_names[i]))
# Create object of KNN classifier
classifier =
KNeighborsClassifier(n_neighbors=1)# Perform
Training
classifier.fit(x_train, y_train)
# Perform testing
y_pred=classifier.predict(x_test)
# Display the results
print("Results of Classification using K-nn with K=1 ")for r in range(0,len(x_test)):
print(" Sample:", str(x_test[r]), " Actual-label:", str(y_test[r]), " Predicted-label:",str(y_pred[r]))
```

```
print("Classification Accuracy :" , classifier.score(x_test,y_test));
#from sklearn.metrics import classification_report, confusion_matrix
#print('Confusion Matrix')
#print(confusion_matrix(y_test,y_pred))
#print('Accuracy Metrics')
#print(classification_report(y_test,y_pred))
```

Output:

Result-1

```
Iris Data set loaded...
Dataset is split into training and testing samples...
Size of trainng data and its label (135, 4) (135,)
Size of trainng data and its label (15, 4) (15,) Label
0 - setosa
Label 1 - versicolor
Label 2 - virginica

Results of Classification using K-nn with K=1
Sample: [4.4 3. 1.3 0.2] Actual-label: 0 Predicted-label: 0
Sample: [5.1 2.5 3. 1.1] Actual-label: 1 Predicted-label: 1
Sample: [6.1 2.8 4. 1.3] Actual-label: 1 Predicted-label: 1
Sample: [6. 2.7 5.1 1.6] Actual-label: 1 Predicted-label: 2
Sample: [6.7 2.5 5.8 1.8] Actual-label: 2 Predicted-label: 2
Sample: [5.1 3.8 1.5 0.3] Actual-label: 0 Predicted-label: 0
Sample: [6.7 3.1 4.4 1.4] Actual-label: 1 Predicted-label: 1
Sample: [4.8 3.4 1.6 0.2] Actual-label: 0 Predicted-label: 0
Sample: [5.1 3.5 1.4 0.3] Actual-label: 0 Predicted-label: 0
Sample: [5.4 3.7 1.5 0.2] Actual-label: 0 Predicted-label: 0
Sample: [5.7 2.8 4.1 1.3] Actual-label: 1 Predicted-label: 1
Sample: [4.5 2.3 1.3 0.3] Actual-label: 0 Predicted-label: 0
Sample: [4.4 2.9 1.4 0.2] Actual-label: 0 Predicted-label: 0
Sample: [5.1 3.5 1.4 0.2] Actual-label: 0 Predicted-label: 0
Sample: [6.2 3.4 5.4 2.3] Actual-label: 2 Predicted-label: 2
    Classification Accuracy : 0.93
```

Sample Viva Questions:

1. What is the KNN Algorithm?

KNN(K-nearest neighbors) is a **supervised** learning and **non-parametric** algorithmthat can be used tosolve both classification and regression problem statements.

It uses data in which there is a target column present **i.e, labeled data** to model a function to produce an output for the unseen data. It uses the Euclidean distance formulato compute the distance between the datapoints for classification or prediction.

The main objective of this algorithm is that similar data points must be close to each other so it uses thedistance to calculate the similar points that are close to each other.

2. Why is KNN a non-parametric Algorithm?

The term "**non-parametric"** refers to not making any assumptions on the underlying data distribution. These methods do not have any fixed numbers of parameters in the model.

Similarly in KNN, the model parameters grow with the training data by considering eachtraining case as aparameter of the model. So, KNN is a non-parametric algorithm.

3. What is "K" in the KNN Algorithm?

K represents the number of nearest neighbors you want to select to predict the class ofa given item, whichis coming as an unseen dataset for the model.

4. Why is the odd value of "K" preferred over even values in the KNN Algorithm?

The odd value of K should be preferred over even values in order to ensure that there are no ties in thevoting. If the square root of a number of data points is even, then add orsubtract 1 to it to make it odd.

5. How does the KNN algorithm make the predictions on the unseen dataset?

The following operations have happened during each iteration of the algorithm. For each of the unseen ortest data point, the kNN classifier must:

Step-1: Calculate the distances of test point to all points in the training set and store them

Step-2: Sort the calculated distances in increasing order
Step-3: Store the K nearest points from our training dataset
Step-4: Calculate the proportions of each class
Step-5: Assign the class with the highest proportion

EXPERIMENT NO:4

4. Given the following data, which specify classifications for nine combinations of VAR1 and VAR2 predict a classification for a case where VAR1=0.906 and VAR2=0.606, using the result of kmeans clustering with 3 means (i.e., 3 centroids)

VAR1 VAR2 CLASS
1.713 1.586 0
0.180 1.786 1
0.353 1.240 1
0.940 1.566 0
1.486 0.759 1
1.266 1.106 0
1.540 0.419 1
0.459 1.799 1
0.773 0.186 1

Aim:

To predict a classification for a case where VAR1=0.906 and VAR2=0.606, usingthe result ofkmeans clustering with 3 means and given data.

Theory:

Step 1: Python 3 code snippet demonstrates the implementation of a simple K- Means clustering to automatically divide input data into groups based on given features.

Step 2: " , " separated CSV file is loaded first, which contains three correspondinginput columns.

Step 3: K-Means clustering model is created from this input data. Afterwards,new data can be classified using the predict() method based on the learned model.

Step 4: The Scikit-learn and the Pandas library to be installed (pip install sklearn, pipinstall pandas).

Step 5

input_data.txt
VAR1,VAR2,c
ls1.713,1.586,0
0.180,1.786,1

0.353,1.240,1

0.940,1.566,0

1.486,0.759,1

1.266,1.106,0

1.540,0.419,1

0.459,1.799,1

0.773,0.186,1

Source Code:

```
from sklearn.cluster import KMeans
import numpy as np
X = np.array([[1.713,1.586], [0.180,1.786], [0.353,1.240],[0.940,1.566],
[1.486,0.759],[1.266,1.106],[1.540,0.419],[0.459,1.799],[0.773,0.186]])
y=np.array([0,1,1,0,1,0,1,1,1])
kmeans = KMeans(n_clusters=3, random_state=0).fit(X,y)
print("The input data is ")
print("VAR1 \t VAR2 \t
CLASS")i=0
for  val in X:

    print(val[0],"\t",val[1],"\t",y[i])
        i+=1
print("="*20)

# To get test data from the user
print("The Test data to predict ")
test_data = []
VAR1 = float(input("Enter Value for VAR1 :"))
VAR2 = float(input("Enter Value for VAR2 :"))
test_data.append(VAR1)
test_data.append(VAR2)
print("="*20)

print("The predicted Class is : ",kmeans.predict([test_data]))
```

Output:

```
The input data is
VAR1    VAR2    CLAS
                S
1.713   1.586   0
0.18    1.786   1
0.353   1.24    1
0.94    1.566   0
1.486   0.759   1
1.266   1.106   0
1.54    0.419   1
0.459   1.799   1
0.773   0.186   1
====================
The Test data to predict
Enter Value for VAR1 :1.7
Enter Value for VAR2 :1.5
====================
The predicted Class is :  [2]
```

Sample Viva Questions:

1. What is K means Clustering Algorithm?

K Means algorithm is a centroid-based clustering (unsupervised) technique. Thistechnique groups the dataset into k different clusters having an almost equal number of points. Each of the clusters has a centroid point which represents the mean of the data points lying in that cluster.

The idea of the K-Means algorithm is to find k-centroid points and every point in the dataset will belong to either of the k-sets having minimum Euclidean distance.

2. Is Feature Scaling required for the K means Algorithm?

Yes, K-Means typically needs to have some form of normalization done on the datasetsto work properly since it is sensitive to both the mean and variance of the datasets.

For performing feature scaling, generally, **StandardScaler** is recommended, butdepending on the specificuse cases, other techniques might be more suitable as well.

3. Which metrics can you use to find the accuracy of the K means Algorithm?

There does not exist a correct answer to this question as k means being an unsupervised learning technique does not discuss anything about the output column. As a result, one can not get the accuracy number or values from the algorithm directly.

4. What are the advantages of the K means Algorithm?

Advantages:

- Easy to understand and implement.
- Computationally efficient for both training and prediction.
- Guaranteed convergence.

5. What are the disadvantages of the K means Algorithm?

Disadvantages:

- We need to provide the number of clusters as an input variable to thealgorithm.
- It is very sensitive to the initialization process.
- Good at clustering when we are dealing with spherical cluster shapes,but it will perform poorly when dealing with more complicated shapes.
- Due to the leveraging of the Euclidean distance function, it is sensitive tooutliers.

EXPERIMENT NO: 5

6. The following training examples map descriptions of individuals onto high, medium and lowcredit-worthiness.

Aim:

To unconditional probability of `golf' and the conditional probability of `single' given `medRisk' in the dataset

medium skiing design single twenties no ->highRisk
high golf trading married forties yes ->lowRisk
low speedway transport married thirties yes ->medRisk
medium football banking single thirties yes ->lowRisk
high flying media married fifties yes ->highRisk
low football security single twenties no ->medRisk
medium golf media single thirties yes ->medRisk
medium golf transport married forties yes ->lowRisk
high skiing banking single thirties yes ->highRisk low
golf unemployed married forties yes ->highRisk

Input attributes are (from left to right) income, recreation, job, status, age-group, home- owner. Find the unconditional probability of `golf' and the conditional probability of `single' given `medRisk' in the dataset?

Theory:

In the given data set,
The total number of records are 10.
The number of records which contains 'golf' are 4.
Then, the Unconditional probability of golf :

= The number of records which contains 'golf' / total number of records
= 4 / 10
= 0.4

To find the Conditional probability of single given medRisk,

S : single
MR : medRisk

By the definition of Baye's rule(conditional probability), we have
$P(S \mid MR) = P(S \cap MR) / P(MR)$
Based on the given problem statement,

$P(S \cap MR)$ = The number of MedRisk with Single records / total number of Records
= 2 / 10 = 0.2

P(MR) = The number of records with MedRisk /total number of Records
= 3 / 10 = 0.3

Then, the Conditional probability of single given medRisk
$P(S \mid MR) = 0.2 / 0.3$
= 0.66666

Source Code:

```
total_Records=10
numGolfRecords=4
unConditionalprobGolf=numGolfRecords / total_Records
print("Unconditional probability of golf: ={}".format(unConditionalprobGolf))#conditional
probability of 'single' given 'medRisk'
numMedRiskSingle=2
numMedRisk=3
probMedRiskSingle=numMedRiskSingle/total_Records probMedRisk=numMedRisk/total_Records
conditionalProb=(probMedRiskSingle/probMedRisk)
print("Conditional probability of single given medRisk: = {}".format(conditionalProb))
```

Output:

```
Unconditional probability of golf: =0.4
Conditional probability of single given medRisk: = 0.6666666666666667
```

Sample Viva Questions:

1. What are the practical Considerations In K-Means:

- A choosing number of Clusters in Advance (K).
- Standardization of Data (scaling).
- Categorical Data (can be solved with K-Mode).
- Impact of initial Centroids and Outliers.

2. Why do you prefer Euclidean distance over Manhattan distance in the K means Algorithm?

Euclidean distance is preferred over Manhattan distance since Manhattan distance calculates distance only vertically or horizontally due to which it has dimension restrictions.

On the contrary, Euclidean distance can be used in any space to calculate the distances between the data points. Since in K means algorithm the data points can be present in any dimension, so Euclidean distance is a more suitable option.

3. Is it possible that the assignment of data points to clusters does notchange between successiveiterations in the K means Algorithm?

When the K-Means algorithm has reached the local or global minima, it will not changethe assignment ofdata points to clusters for two successive iterations during the algorithm run.

4. How to Choose K Value in K-Means:

Elbow method

step1: compute clustering algorithm for different values of k. for example k=[1,2,3,4,5,6,7,8,9,10]

step2: for each k calculate the within-cluster sum of squares(WCSS). step3: plot curve of WCSS according to the number of clusters.

step4: The location of bend in the plot is generally considered an indicator of the approximate number ofclusters.

5. What is the training and testing complexity of the K means Algorithm?

Training complexity in terms of Big-O notation:

If we use Lloyd‟s algorithm, the complexity for training is: **"K*I*N*M"**

where,

K: It represents the number of clusters **I:** It represents the number of iterations **N:** It represents the sample size
M: It represents the number of variables

Conclusion: There is a significant Impact on capping the number of iterations.

Predicting complexity in terms of Big-O notation:

"K*N*M"

Prediction needs to be computed for each record, the distance to each cluster and assigned to the nearestones.

EXPERIMENT NO: 6

6. Implement linear regression using python.

Aim:

To implement linear regression using python

Theory:

Linear Regression (Python Implementation)

Linear regression is a statistical method for modelling relationship between a dependentvariable with a given set of independent variables.

In order to provide a basic understanding of linear regression, we start with the most basicversion of linear regression, i.e. **Simple linear regression**.

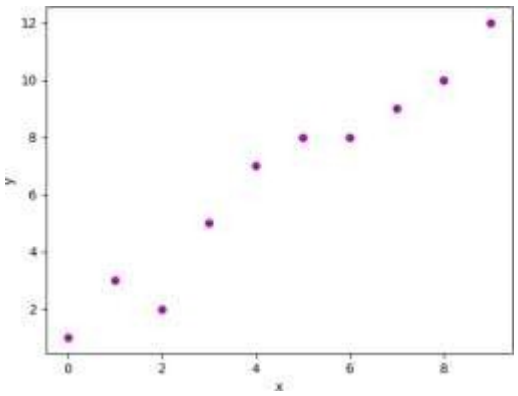
Simple Linear Regression

Simple linear regression is an approach for predicting a **response** using a **single feature**.It is assumed that the two variables are linearly related. Hence, we try to find a linear function that predicts the response value(y) as accurately as possible as a function of the feature or independent variable(x).

x	0	1	2	3	4	5	6	7	8	9
y	1	3	2	5	7	8	8	9	10	12

Let us consider a dataset where we have a value of response y for every feature x:

For generality, we define:
x as **feature vector**, i.e x = [x_1, x_2, …., x_n], y as **response vector**, i.e y = [y_1, y_2, …., y_n]for **n** observations (in above example, n=10).
A scatter plot of above dataset looks like:-



Now, the task is to find a **line which fits best** in above scatter plot so that we can predictthe response for any new feature values. (i.e a value of x not present in dataset)
This line is called **regression line**.
The equation of regression line is represented as:

Here,

- h(x_i) represents the **predicted response value** for i^{th} observation.
- b_0 and b_1 are regression coefficients and represent **y-intercept** and **slope** of regression line respectively.

To create our model, we must "learn" or estimate the values of regression coefficients b_0and b_1. And once we"ve estimated these coefficients, we can use the model to predict responses!
In this article, we are going to use the principle of **Least Squares** .
Now consider:

Here, e_i is **residual error** in ith observation.
So, our aim is to minimize the total residual error.
We define the squared error or cost function, J as:

and our task is to find the value of b_0 and b_1 for which J(b_0,b_1) is minimum!
Without going into the mathematical details, we present the result here:

where SS_xy is the sum of cross-deviations of y and x:
and SS_xx is the sum of squared deviations of x:

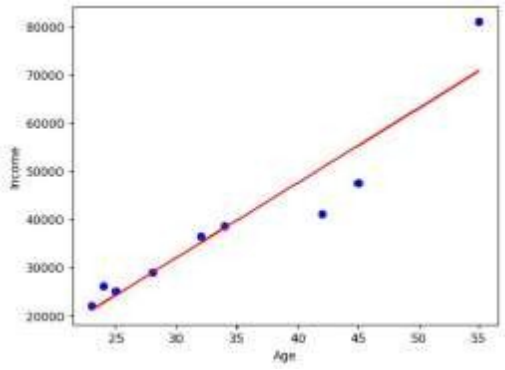
Source Code:

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
# To read data from Age_Income.csv file
dataFrame = pd.read_csv('Age_Income.csv')#
To place data in to age and income vectorsage
= dataFrame['Age']
income = dataFrame['Income']

# number of points
num = np.size(age)
# To find the mean of age and income vector
mean_age = np.mean(age)
mean_income = np.mean(income)

# calculating cross-deviation and deviation about age
CD_ageincome = np.sum(income*age) - num*mean_income*mean_ageCD_ageage =
np.sum(age*age) - num*mean_age*mean_age

# calculating regression coefficients
b1 = CD_ageincome / CD_ageage
b0 = mean_income - b1*mean_age#
to display coefficients
```

```
print("Estimated Coefficients :")
print("b0 = ",b0,"\nb1 = ",b1)
# To plot the actual points as scatter plot
plt.scatter(age, income, color = "b",marker = "o")
# TO predict response vector
response_Vec = b0 + b1*age#
To plot the regression line
plt.plot(age, response_Vec, color = "r")#
Placing labels
plt.xlabel('Age')
plt.ylabel('Income') #
To display plot
plt.show()
```

Output:

```
Estimated Coefficients :
b0 = -14560.45016077166
b1 = 1550.7923748277433
```



Sample Viva Questions:

1. **What do you mean by the Logistic Regression?**

It‟s a classification algorithm that is used where the target variable is of categorical nature. The main objective behind Logistic Regression is to determine the relationshipbetween features and the probability of a particular outcome.

For Example, when we need to predict whether a student passes or fails in an exam given the number of hours spent studying as a feature, the target variable comprises two values i.e. pass and fail.

Therefore, we can solve classification problem statements which is a supervise machine learning technique using Logistic Regression.

2. What are the different types of Logistic Regression?

Three different types of Logistic Regression are as follows:

Binary Logistic Regression: In this, the target variable has only two 2 possibleoutcomes. For Example**,** 0 and 1, or pass and fail or true and false.

Multinomial Logistic Regression: In this, the target variable can havethree or more possible valueswithout any order.

3. How do we handle categorical variables in Logistic Regression?

The inputs given to a Logistic Regression model need to be numeric. The algorithm cannot handle categorical variables directly. So, we need to convert the categorical data into a numerical format that is suitable for the algorithm to process.
Each level of the categorical variable will be assigned a unique numeric value also known as a **dummy variable.** These dummy variables are handled by the Logistic Regression model in the same manner as any other numeric value.

4. What are the assumptions made in Logistic Regression?

Some of the assumptions of Logistic Regression are as follows:

- It assumes that there is minimal or **no multi collinearity** among theindependent variables i.e,predictors are not correlated.
- There should be a linear relationship between the logit of the outcome andeach predictor variable. The logit function is described as **logit(p) = log(p/(1-p))**, where p is the probability of the target outcome.
- Sometimes to predict properly, it usually requires a **large sample size**.
- The Logistic Regression which has **binary classification**i.e, two classes assume that the target variable is binary, and ordered Logistic Regression requires the target variable to be ordered.

For example**,** Too Little, About Right, Too Much.

- It assumes there is **no dependency** between the observations.

5. Can we solve the multiclass classification problems using Logistic Regression? If Yes then How?

Yes, in order to deal with multiclass classification using Logistic Regression, the most famous method is known as the one-vs-all approach. In this approach, a number of models are trained, which is equal to the number of classes. These models work in a specific way.

For Example, the first model classifies the data point depending on whether it belongsto class 1 or some other class(not class 1); the second model classifies the data point into class 2 or some other class(not class2) and so-on for all other classes. So, in this manner, each data point can be checked over all the classes.

EXPERIMENT NO: 7

7. Implement Naïve Bayes theorem to classify the English text

Aim:

To implement Naïve Bayes theorem to classify the English text

Theory:

LEARN_NAIVE_BAYES_TEXT (Examples, V)

Examples is a set of text documents along with their target values. V is the set of all possibletarget values. This function learns the probability terms P(w_k |v_j,), describingthe probabilitythat a randomly drawn word from a document in class v_j will be the English word w_k. It also learns the class prior probabilities P(v_j).

1. *collect all words, punctuation, and other tokens that occur in Examples*
   - *Vocabulary* ← ***c*** the set of all distinct words and other tokens occurring in any textdocument from *Examples*

2. *calculate the required P(v_j) and P(w_k|v_j) probability terms*

   - For each target value v_j in V do
     - $docs_j$ ← the subset of documents from *Examples* for which the target valueis v_j
     - $P(v_j)$ ← | $docs_j$ | / |Examples|
     - $Text_j$ ← a single document created by concatenating all members of $docs_j$
     - n ← total number of distinct word positions in $Text_j$
     - for each word w_k in *Vocabulary*
       - n_k ← number of times word $\boldsymbol{w_k}$ occurs in $Text_j$
       - $P(w_k|v_j)$ ← (n_k + 1) / (n + | *Vocabulary*|)

CLASSIFY_NAIVE_BAYES_TEXT (Doc)

Return the estimated target value for the document Doc. a_i denotes the word found inthe i^{th}position within Doc.

- *positions* ← all word positions in *Doc* that contain tokens found in *Vocabulary*
- Return V_{NB}, where

Data set:

	Text Documents	Label
1	I love this sandwich	pos
2	This is an amazing place	pos
3	I feel very good about these beers	pos
4	This is my best work	pos
5	What an awesome view	pos
6	I do not like this restaurant	neg
7	I am tired of this stuff	neg
8	I can't deal with this	neg
9	He is my sworn enemy	neg
10	My boss is horrible	neg
11	This is an awesome place	pos
12	I do not like the taste of this juice	neg
13	I love to dance	pos
14	I am sick and tired of this place	neg
15	What a great holiday	pos
16	That is a bad locality to stay	neg
17	We will have good fun tomorrow	pos
18	I went to my enemy's house today	neg

Source Code:

```
import pandas as pd
msg=pd.read_csv('naivetext.csv',names=['message','label']) print('The
dimensions of the dataset',msg.shape)
msg['labelnum']=msg.label.map({'pos':1,'neg':0})
X=msg.message
y=msg.labelnum
print(X)
print(y)
```

```
from sklearn.model_selection import train_test_splitxtrain,xtest,ytrain,ytest=train_test_split(X,y)
print ('\n The total number of Training Data :',ytrain.shape)
print ('\n The total number of Test Data :',ytest.shape)
#output of count vectoriser is a sparse matrix
from sklearn.feature_extraction.text import CountVectorizer
count_vect = CountVectorizer()
xtrain_dtm = count_vect.fit_transform(xtrain)
xtest_dtm=count_vect.transform(xtest)
print('\n The words or Tokens in the text documents \n')
print(count_vect.get_feature_names_out())
df=pd.DataFrame(xtrain_dtm.toarray(),columns=count_vect.get_feature_names()) #
Training Naive Bayes (NB) classifier on training data.
from sklearn.naive_bayes import MultinomialNB
clf = MultinomialNB().fit(xtrain_dtm,ytrain)
predicted = clf.predict(xtest_dtm)
#printing accuracy, Confusion matrix, Precision and Recall
from sklearn import metrics
print('\n Accuracy of the classifer is',metrics.accuracy_score(ytest,predicted))print('\n
Confusion matrix')
print(metrics.confusion_matrix(ytest,predicted))

print('\n The value of Precision' , metrics.precision_score(ytest,predicted))print('\n
The value of Recall' , metrics.recall_score(ytest,predicted))
```

Output:

```
The dimensions of the dataset (18, 2)
0                    I love this sandwich
1              This is an amazing place
2      I feel very good about these beers
3                    This is my best work
4                    What an awesome view
5           I do not like this restaurant
6              I am tired of this stuff
7                I can't deal with this
8                    He is my sworn enemy
9                     My boss is horrible
10             This is an awesome place
11   I do not like the taste of this juice
12                       I love to dance
```

13 I am sick and tired of this place

14 What a great holiday

15 That is a bad locality to stay

16 We will have good fun tomorrow

17 I went to my enemy's house today
Name: message, dtype: object
0 1

1 1

2 1

3 1

4 1

5 0

6 0

7 0

8 0

9 0

10 1

11 0

12 1

13 0

14 1

15 0

16 1

17 0

Name: labelnum, dtype: int64

The total number of Training Data : (13,)
The total number of Test Data : (5,)

The words or Tokens in the text documents

['am', 'amazing', 'an', 'awesome', 'best', 'boss', 'can', 'dance', 'deal', 'do', 'enemy', 'great', 'he', 'holiday', 'horrible', 'is', 'juice', 'like', 'love', 'my', 'not', 'of', 'place', 'restaurant', 'sandwich', 'stuff', 'sworn', 'taste', 'the', 'this', 'tired', 'to', 'view', 'what', 'with', 'work']Accuracy of the classifer is 0.8

Confusion matrix
[[2 1]
[0 2]]

The value of Precision 0.6666666666666666
The value of Recall 1.0

Basic knowledge

Confusion Matrix

		Actual	
		Positive	Negative
Predicted	Positive	True Positive	False Positive
	Negative	False Negative	True Negative

True positives: data points labelled as positive that are actually positive **False positives:** data points labelled as positive that are actually negative **True negatives:** data points labelled as negative that are actually negative **False negatives:** data points labelled as negative that are actually positive

$$\text{Recall} = \frac{True\ Positive}{True\ Positive + False\ Negative}$$

$$= \frac{True\ Positive}{Total\ Actual\ Positive}$$

		Actual	
		Positive	Negative
Predicted	Positive	True Positive	False Positive
	Negative	False Negative	True Negative

$$\text{Precision} = \frac{True\ Positive}{True\ Positive + False\ Positive}$$

$$= \frac{True\ Positive}{Total\ Predicted\ Positive}$$

		Actual	
		Positive	Negative
Predicted	Positive	True Positive	False Positive
	Negative	False Negative	True Negative

Example:

		Actual	
		Positive	Negative
Predicted	Positive	1 TP	3 FP
	Negative	0 FN	1 TN

Sample Viva Questions:

1. What is the Naive Bayes algorithm to classify English text?

The Naive Bayes text classification algorithm is a type of probabilistic model used inmachine learning. Harry R. Felson and Robert M. Maxwell designed the first text classification method to classify text documents with zero or more words from the document being classified as authorship or genre.

2. How Bayes Theorem is used in naive Bayes classification?

Naive Bayes Classifier

The Bayes Theorem assumes that each input variable is dependent upon all other variables. This is a cause of complexity in the calculation. We can remove this assumption and consider each input variable as being independent from each other.

3. Which Naive Bayes is best for text classification?

Multinomial Naive Bayes
The Multinomial Naive Bayes can be accepted as the probabilistic approach to classifying documents in the case of acknowledging the frequency of a specified wordin a text document.

4. What is a real life example of Naive Bayes?

Naive Bayes is mostly used in real-world applications like sentiment analysis, spamfiltering, recommendation systems, etc. They are extremely fast and easy to implement as compared to other machine learning models.

5. What is the Naive Bayes algorithm for text data?

The Naive Bayes algorithm is a supervised machine learning algorithm based on theBayes' theorem. It is a probabilistic classifier that is often used in NLP tasks like sentiment analysis

EXPERIMENT NO: 8

8. Implement an algorithm to demonstrate the significance of geneticalgorithm

Aim:

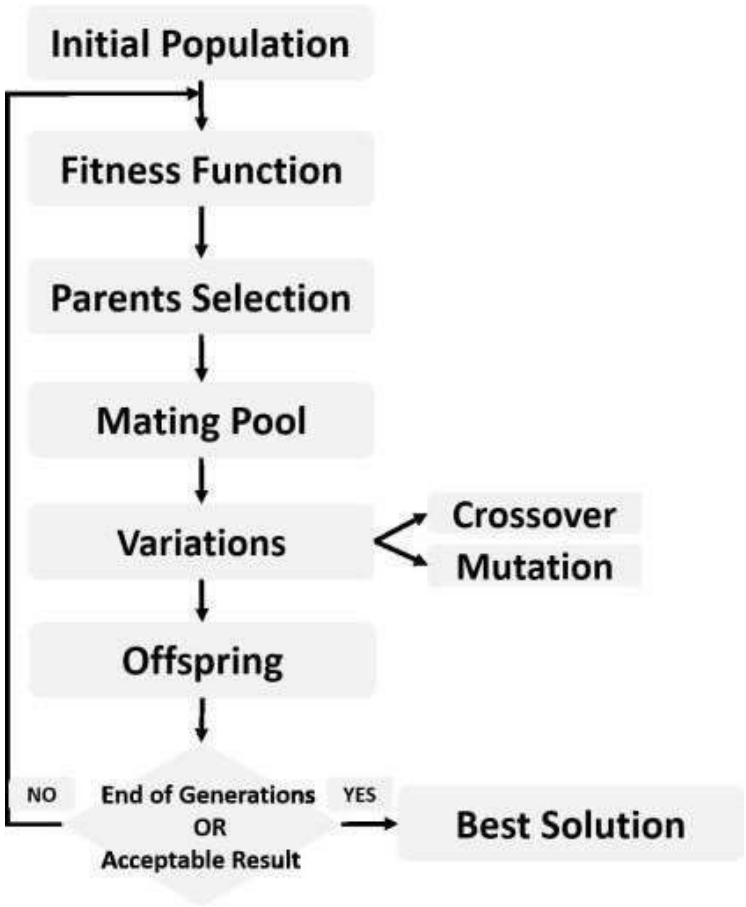
To implement an algorithm to demonstrate the significance of genetic algorithm

Theory:

Genetic algorithm

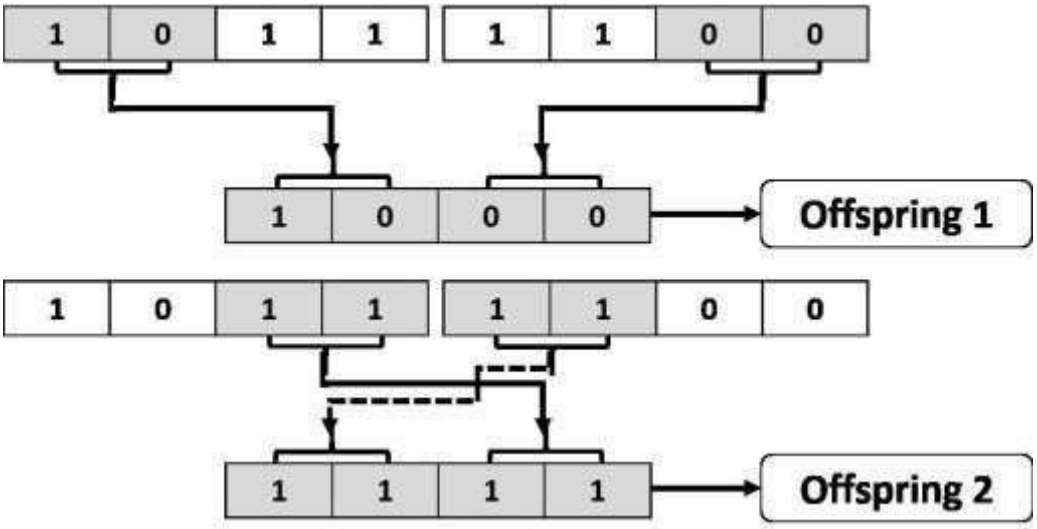
The genetic algorithm is a population-based evolutionary algorithm, where a group of solutions works together to find the optimal parameters for a problem. The belowfigure, from this book, summarizes all the steps in the genetic algorithm.

The population of solutions is initialized randomly, where each solution consists of anumber of genes. The quality of solutions is assessed using a fitness function, whichreturns a numeric value representing how fit the solution is.



The high-quality (high-fitness) solutions survive longer than the ones with low fitness. The higher the fitness, the higher probability of selecting the solution as a parent to produce new offspring. To produce the offspring, pairs of parents mate using the crossover operation, where a new solution is generated that carries genesfrom its parents.

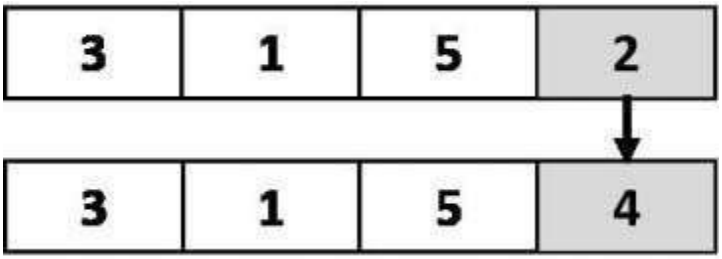
After crossover, mutation is applied to add some random changes over the solution.The evolution continues through a number of generations to reach the highest- quality solution.



If there are some bad genes within the parents, they will definitely be transferred totheir children after crossover. The mutation operation plays a crucial role in fixing this issue.

During mutation, some genes are randomly selected from each child where some random changes are applied. Genes are selected based on a random probability for each gene. If the probability of mutating a gene is smaller than or equal to a predefined threshold, then this gene will be selected for mutation. Otherwise, it will be skipped. We‟ll discuss mutation probability later on.

Let‟s assume there are 4 genes in the solution, as in the next figure, where only thelast gene is selected for mutation. A random change is applied to change its old value **2** and the new value is **4**



Source Code:

```
# Python3 program to create target string, starting from#
random string using Genetic Algorithm

import random

# Number of individuals in each generation
POPULATION_SIZE = 100

# Valid genes
GENES =
'''abcdefghijklmnopqrstuvwxyzABCDEFGHIJKLMNOP
QRSTUVWXYZ 1234567890, .-;:_!"#%&/()=?@${[]}'''

# Target string to be generated
TARGET = "CMR College"

  class
    Individual(object):'''
    Class representing individual in population'''
    def___init_(self, chromosome):
            self.chromosome = chromosome
            self.fitness = self.cal_fitness()

    @classmethod
    def mutated_genes(self):'''
            create random genes for mutation'''
            global GENES
            gene = random.choice(GENES)
            return gene

    @classmethod
    def create_gnome(self):'''
            create chromosome or string of genes'''
            global TARGET
            gnome_len = len(TARGET)
            return [self.mutated_genes() for _ in range(gnome_len)]

    def mate(self, par2):
            '''
            Perform mating and produce new offspring'''

            # chromosome for offspring
            child_chromosome = []
            for gp1, gp2 in zip(self.chromosome, par2.chromosome):
```

```
                # random probability prob =
                random.random()

                # if prob is less than 0.45, insert gene# from
                parent 1
                if prob < 0.45:
                        child_chromosome.append(gp1)

                # if prob is between 0.45 and 0.90, insert# gene
                from parent 2
                elif prob < 0.90:
                        child_chromosome.append(gp2)

                # otherwise insert random gene(mutate),# for
                maintaining diversity
                else:
                        child_chromosome.append(self.mutated_genes())

        # create new Individual(offspring) using#
        generated chromosome for offspring return
        Individual(child_chromosome)

  def cal_fitness(self):
        '''
        Calculate fitness score, it is the number of
        characters in string which differ from targetstring.
        '''
        global TARGET
        fitness = 0
        for gs, gt in zip(self.chromosome, TARGET):if gs
                != gt: fitness+= 1
        return fitness

  # Driver code
def main():
  global POPULATION_SIZE

  #current generation
  generation = 1

  found = False
  population = []

  # create initial population
  for _ in range(POPULATION_SIZE):
                        gnome = Individual.create_gnome() population.append(Individual(gnome))
  while not found:

        # sort the population in increasing order of fitness score population
        = sorted(population, key = lambda x:x.fitness)
```

```
        # if the individual having lowest fitness score ie.
        # 0 then we know that we have reached to the target# and
        break the loop
        if population[0].fitness <= 0:
                found = True
                break

        # Otherwise generate new offsprings for new generation
        new_generation = []

        # Perform Elitism, that mean 10% of fittest population# goes to
        the next generation
        s = int((10*POPULATION_SIZE)/100)
        new_generation.extend(population[:s])

        # From 50% of fittest population, Individuals# will
        mate to produce offspring
        s = int((90*POPULATION_SIZE)/100)
        for _ in range(s):
                parent1 = random.choice(population[:50])parent2
                = random.choice(population[:50])child =
                parent1.mate(parent2)
                new_generation.append(child)

        population = new_generation

        print("Generation: {}\tString: {}\tFitness: {}".\
                format(generation,
                "".join(population[0].chromosome),
                population[0].fitness))

        generation += 1


print("Generation: {}\tString: {}\tFitness: {}".\
        format(generation,
        "".join(population[0].chromosome),
        population[0].fitness))

if__name__== '__main__':
        main()
```

Output:

```
Generation: 1    String: CG H    Fitness: 3
Generation: 2    String: CG H    Fitness: 3
Generation: 3    String: CG H    Fitness: 3
Generation: 4    String: APR     Fitness: 2
Generation: 5    String: APR     Fitness: 2
Generation: 6    String: CMRS    Fitness: 1
Generation: 7    String: CMRS    Fitness: 1
Generation: 8    String: CMR     Fitness: 0
```

Sample Viva Questions:

1. **Explain some basic concepts and terms related to *Genetic Algorithm***

- **Population:** This is a subset of all the *probable* solutions that can solve the given problem.
- **Chromosomes:** A chromosome is one of the solutions in the population.
- **Gene:** This is an element in a chromosome.
- **Allele:** This is the value given to a gene in a specific chromosome.
- **Fitness function:** This is a function that uses a specific *input* to produce an improved *output*. The solution is used as the input while the output is in the form of solution suitability.

2. **What is a *Mutation* and why is it programmed into the algorithm?**

- ***Mutation*** introduces new patterns in the chromosomes, and it helps to find solutions inuncharted areas.
- ***Mutations*** are implemented as *random* changes in the chromosomes. It may be programmed, for example, as a *bit flip* where a single bit of the chromosome changes.
- The purpose of *mutation* is to periodically *refresh* the population.

3. **Compare the *Single-Point* and *Two-Point* crossover**

- In **Single-point Crossover**, a location is selected *randomly* on both the parents. The location is the same in both the parents *chromosome*. The genes to the right of the *crossover point* is swapped between the two parents.

- In **Two-point Crossover**, *two* locations are selected *randomly* in the parents' chromosome. Then the genes in between the two points are swapped between the twoparents.

4. Name some *types* of *Mutation* in GA

- **Flip bit mutation:** This mutation can be applied to binary chromosomes. In this type of mutation, a single bit is selected randomly and it is *flipped*. It can be programmed so that multiple bits are flipped instead of one.
- **Swap mutation:** When this mutation is applied, two *genes* are randomly selected andtheir values are swapped within the same chromosome.
- **Inversion mutation:** In this mutation, *random* sequences of genes are selected and theorder of those genes is *reversed*.
- **Scramble mutation:** In this mutation, a *random* sequence of genes is selected and thegenes in that sequence is *shuffled*.

5. What are some *Stopping Conditions* that a genetic algorithm may implement?Some stopping conditions that can be implemented are:

- Stopping when the **maximum (absolute) number of generations** has been reached.This limits the resources and time required by the algorithm.
- Stopping if there is **no noticeable improvement in the fitness score**. This can be implemented by storing the best fitness score achieved in every generation and comparingthe current best value to the one achieved at every generation. If the difference is small compared to the threshold, then the algorithm can stop.
- Stopping after a **predetermined time** after starting the computation..

EXPERIMENT NO:9

9. Implement the finite words classification system using Back-propagationalgorithm

Aim:

To implement the finite words classification system using Back-propagation algorithm

Theory:

Artificial neural networks (ANNs) provide a general, practical method for learningrealvalued, discrete-valued, and vector-valued functions from examples.

Algorithms such as BACKPROPAGATION gradient descent to tune network parameters to best fit a training set of input-output pairs.

ANN learning is robust to errors in the training data and has been successfully appliedto problems such as interpreting visual scenes, speech recognition, and learning robotcontrol.

Source Code:

```
import numpy as np # numpy is

X = np.array(([2, 9], [1, 5], [3, 6]),dtype=float)

y = np.array(([92], [86], [89]),dtype=float)

X = X/np.amax(X,axis=0) # Normalizey
    = y/100
def sigmoid(x):

return 1/(1+np.exp(-x))def
    sigmoid_grad(x):
return x * (1 - x)

# Variable initialization

epoch=1000 #Setting training iterationseta
    =0.2 #Setting learning rate (eta)
input_neurons = 2 #number of features in data set

hidden_neurons = 3 #number of hidden layers neurons
```

```
output_neurons = 1 #number of

# Weight and bias - Random initialization
    wh=np.random.uniform(size=(input_neurons,hidden_neurons)) # 2x3
    bh=np.random.uniform(size=(1,hidden_neurons)) # 1x3
    wout=np.random.uniform(size=(hidden_neurons,output_neurons))
    bout=np.random.uniform(size=(1,output_neurons))
for i in range(epoch):

h_ip=np.dot(X,wh) + bh # Dot product + biash_act =
    sigmoid(h_ip) # Activation function
    o_ip=np.dot(h_act,wout) + bout
output = sigmoid(o_ip)
    #Backpropagation
# Error at Output layer

Eo = y-output # Error at o/p
    outgrad =
    sigmoid_grad(output)
d_output = Eo* outgrad # Errj=Oj(1-Oj)(Tj-Oj)

Eh = d_output.dot(wout.T) # .T means transpose

hiddengrad = sigmoid_grad(h_act) # How much hidden layer wts contributed to errord_hidden = Eh *
    hiddengrad
wout += h_act.T.dot(d_output) *eta # Dotproduct of nextlayererror and currentlayeropwh +=
    X.T.dot(d_hidden) *eta
print("Normalized Input: \n" + str(X))
    print("Actual Output: \n" + str(y))
    print("Predicted Output: \n" ,output
```

Output:

```
Normalized Input:
[[0.66666667 1.        ]
 [0.33333333 0.55555556]
 [1.         0.66666667]]
Actual Output:
[[0.92]
 [0.86]
 [0.89]]
Predicted Output:
 [[0.77772836]
 [0.76211175]
 [0.77933662]]
```

Sample Viva Questions:

1. **How do you implement the back propagation algorithm?**

Backpropagation Algorithm:
Step 1: Inputs X, arrive through the preconnected path.
Step 2: The input is modeled using true weights W. Weights are usually chosenrandomly.
Step 3: Calculate the output of each neuron from the input layer to the hidden layer tothe output layer.

2. **How backpropagation algorithm can be used in classification?**

Static backpropagation networks can solve static classification problems, such as opticalcharacter recognition (OCR). Recurrent backpropagation. The recurrent backpropagationnetwork is used for fixed-point learning. Recurrent backpropagation activation feeds forward until it reaches a fixed value.

3. **What is the advantage of back propagation algorithm?**

Most prominent advantages of Backpropagation are: Backpropagation is fast, simple andeasy to program. It has no parameters to tune apart from the numbers of input. It is a flexible method as it does not require prior knowledge about the network.

4. **Why is it called backpropagation?**

The calculation of the error δ j k \delta_j^{k} δjk will be shown to be dependent on the values of error terms in the next layer. Thus, computation of the error terms will proceedbackwards from the output layer down to the input layer. This is where backpropagation,or backwards propagation of errors, gets its name.

5. **What is the difference between back propagation and guided back propagation**?

"Guided Backpropagation" uses vanilla backpropagation, except at the ReLUs. Guided Backpropagation combines vanilla backpropagation at ReLUs (leveraging which elementsare positive in the preceding feature map) with DeconvNets (keeping only positive error signals.

Additional Programs

EXPERIMENT NO:10

10. Implementing FIND-S algorithm using python

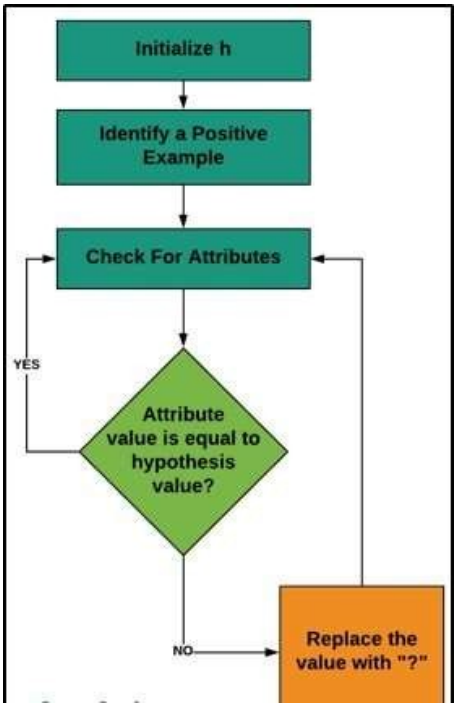
Aim:

Implementing FIND-S algorithm using python

Theory:

1. The process starts with initializing „h‟ with the most specific hypothesis, generally,it is the first positive example in the data set.
2. We check for each positive example. If the example is negative, we will move onto the next example but if it is a positive example we will consider it for the next step.
3. We will check if each attribute in the example is equal to the hypothesis value.
4. If the value matches, then no changes are made.
5. If the value does not match, the value is changed to „?‟.
6. We do this until we reach the last positive example in the data set.

How Does It Work?



Source Code:

```
import pandas as pdimport numpy as np
data = pd.read_csv('data.csv')

print(data)
```

```
concepts = np.array(data)[:,:-1]

print("\n The attributes are :\n",concepts)
target = np.array(data)[:,-1]
print("\n The Target is:\n",target)
def train(con, tar):
    for i, val in enumerate(tar):if val == 'yes':
    specific_h = con[i].copy()
    break


for i, val in enumerate(con):if
 tar[i] == 'yes':
    for x in range(len(specific_h)):if
        val[x] != specific_h[x]:
            specific_h[x] = '?'
        else:
            pass return specific_h
print("\nFinal hypothesis is :",train(concepts, target))
```

Output:

	Sky	air	temp	humidity	wind	water	forecast	Enjoy Sport
0	sunny	warm	normal	strong	warm	same	yes
1	sunny	warm	high	strong	warm	same	yes
2	rainy	cold	high	strong	warm	change	no
3	sunny	warm	high	strong	cool	change	yes

```
The attributes are :
[['sunny' 'warm' 'normal' 'strong' 'warm' 'same']
['sunny' 'warm' 'high' 'strong' 'warm' 'same']
['rainy' 'cold' 'high' 'strong' 'warm' 'change']
['sunny' 'warm' 'high' 'strong' 'cool' 'change']]


The Target is:
['yes' 'yes' 'no' 'yes']

Final hypothesis is : ['sunny' 'warm' '?' 'strong' '?' '?']
```

Sample Viva Questions:

1. What are the 3 functions of algorithm?

Algorithms and Functions

- Input: The algorithm receives input.
- Output: The algorithm produces output.
- Precision: The steps are precisely stated.
- Feasibility: It must be feasible to execute each instruction.
- Flexibility: It should also be possible to make changes in the algorithm withoutputting so much effort on it.

2. What are the 5 characteristics of an algorithm?

The main characteristics of an algorithm are

- well-defined inputs,
- well-defined outputs,
- feasibility,
- finiteness,
- unambiguity,
- Definitiveness and language independence.

3. What is the use of Find S algorithm?

Introduction: The find-S algorithm is a basic concept learning algorithm in machine learning. The find-algorithm finds the most specific hypothesis that fits all the positive examples. We have to note here that the algorithm considers only those positive trainingexample.

4. What are the limitations of the Find S algorithm?

Limitations of Find-S Algorithm

There are a few limitations of the Find-S algorithm listed down below: There is no way to determine if the hypothesis is consistent throughout the data. Inconsistent training sets can actually mislead the Find-S algorithm, since it ignores the negative examples.

5. What is the difference between find s and candidate elimination algorithm?

FIND-S outputs a hypothesis from H that is consistent with the training examples; this isjust one of many hypotheses from H that might fit the training data equally well. The keyidea in the Candidate-Elimination algorithm is to output a description of the set of all hypotheses consistent with the training examples.

EXPERIMENT NO:11

11. Implementing Candidate Elimination algorithm using python

Aim:

Implementing Candidate Elimination algorithm using python

Theory:

Candidate-Elimination Learning Algorithm The Candidate-Elimination algorithm computes the version space containing all hypothesis from H that are consistent with an observed sequence of training examples. It begins by initializing the version space to the set of all hypotheses in H; that is by initializing the G boundary set to contain most general hypothesis in H

G0 = {(?, ?, ?, ?)}

Then initialize the S boundary set to contain most specific hypothesis in H

S0 = {(Θ, Θ, Θ, Θ)}

For each training example, these S and G boundary sets are generalized and specialized, respectively, to eliminate from the version space any hypothesis found inconsistent with the new training examples. After execution of all the training examples, the computed version space contains all the hypotheses consistent with these training examples.

Candidate-Elimination Algorithm

Initialize G to the set of maximally general hypotheses in H
Initialize S to the set of maximally specific hypotheses in H
For every learning example d, do
- If d is a positive example
  - Remove from G any hypothesis inconsistent with d
  - For each hypothesis s in S that is not consistent with d
    - Remove s from S
    - Add to s all minimal generalization h of s such that
      - H is consistent with d, and some member of G is more general than h
    - Remove from s any hypothesis that is more general than another hypothesis is S
- If d is negative example
  - Remove from S any hypothesis inconsistent with d
  - For each hypothesis g in G that is not consistent with d
    - Remove g from G
    - Add to G all minimal specializations h of g such that
      - H is consistent with d, and some member of S is more specific than h
    - Remove from G any hypothesis that is less general than another hypothesis in G

Source Code:

```
import numpy as np
import pandas as pd

data = pd.read_csv('enjoysport.csv')
concepts = np.array(data.iloc[:,0:-1])
print("\nInstances are:\n",concepts) target =
np.array(data.iloc[:,-1]) print("\nTarget
Values are: ",target)

def learn(concepts, target): specific_h =
    concepts[0].copy()
    print("\nInitialization of specific_h and genearal_h")
    print("\nSpecific Boundary: ", specific_h)
    general_h = [["?" for i in range(len(specific_h))] for i in range(len(specific_h))]print("\nGeneric Boundary:
    ",general_h)

    for i, h in enumerate(concepts):
        print("\nInstance", i+1 , "is ", h)if
        target[i] == "yes":
            print("Instance is Positive ") for x
            in range(len(specific_h)):
                if h[x]!= specific_h[x]:
                    specific_h[x] ='?'
                    general_h[x][x] ='?'

        if target[i] == "no": print("Instance is
            Negative ") for x in
            range(len(specific_h)):
                if h[x]!= specific_h[x]: general_h[x][x]
                    = specific_h[x]
                else:
                    general_h[x][x] = '?'

        print("Specific Bundary after ", i+1, "Instance is ", specific_h)
        print("Generic Boundary after ", i+1, "Instance is ", general_h)
        print("\n")

    indices = [i for i, val in enumerate(general_h) if val == ['?', '?', '?', '?', '?', '?']]for i in indices:
        general_h.remove(['?', '?', '?', '?', '?', '?'])return
    specific_h, general_h

s_final, g_final = learn(concepts, target)

print("Final Specific_h: ", s_final, sep="\n")
print("Final General_h: ", g_final, sep="\n")
```

Output:

```
Instances are:
  [['[„sunny“' „warm“' „normal“' „strong“' „warm“' „same“" „yes“]']
  ['[„sunny“' „warm“' „high“' „strong“' „warm“' „same“" „yes“]']
  ['[„rainy“' „cold“' „high“' „strong“' „warm“' „change“" „no“]']
  ['[„sunny“' „warm“' „high“' „strong“' „cool“' „change“" „yes“]]']]

Target Values are: [' ' ' ' ' ' nan] Initialization of
specific_h and genearal_h
Specific Boundary:   ['[„sunny“' „warm“' „normal“' „strong“' „warm“' „same“"
  ' „yes“]']

Generic Boundary:  [['?', '?', '?', '?', '?', '?', '?'], ['?', '?', '?', '?', '?',
'?', '?'], ['?', '?', '?', '?', '?', '?', '?', '?'], ['?', '?', '?', '?', '?', '?', '?',
'?'], ['?', '?', '?', '?', '?', '?', '?'], ['?', '?', '?', '?', '?', '?', '?'],['?', '?', '?', '?', '?', '?', '?']]

Instance 1 is  ['[„sunny“' „warm“' „normal“' „strong“' „warm“' „same“" „yes“]']
Specific Bundary after   1 Instance is   ['[„sunny“' „warm“' „normal“' „strong“' „warm“' „same“"
  ' „yes“]']
Generic Boundary after 1 Instance is [['?', '?', '?', '?', '?', '?', '?'], ['?','?', '?', '?', '?', '?', '?'], ['?', '?', '?', '?', '?',
'?', '?'], ['?', '?', '?',
'?', '?', '?', '?'], ['?', '?', '?', '?', '?', '?', '?'], ['?', '?', '?', '?', '?','?', '?'], ['?', '?', '?', '?', '?', '?', '?']]


Instance 2 is  ['[„sunny“' „warm“' „high“' „strong“' „warm“' „same“' „yes“]']
Specific Bundary after   2 Instance is   ['[„sunny“' „warm“' „normal“' „strong“' „warm“' „same“"
  ' „yes“]']
Generic Boundary after 2 Instance is [['?', '?', '?', '?', '?', '?', '?'], ['?','?', '?', '?', '?', '?', '?'], ['?', '?', '?', '?', '?',
'?', '?'], ['?', '?', '?',
'?', '?', '?', '?'], ['?', '?', '?', '?', '?', '?', '?'], ['?', '?', '?', '?', '?','?', '?'], ['?', '?', '?', '?', '?', '?', '?']]

Instance 3 is  ['[„rainy“' „cold“' „high“' „strong“' „warm“' „change“' „no“]']
Specific Bundary after   3 Instance is   ['[„sunny“' „warm“' „normal“' „strong“' „warm“' „same“"
  ' „yes“]']
Generic Boundary after 3 Instance is [['?', '?', '?', '?', '?', '?', '?'], ['?','?', '?', '?', '?', '?', '?'], ['?', '?', '?', '?', '?',
'?', '?'], ['?', '?', '?',
'?', '?', '?', '?'], ['?', '?', '?', '?', '?', '?', '?'], ['?', '?', '?', '?', '?','?', '?'], ['?', '?', '?', '?', '?', '?', '?']]

Instance 4 is  ['[„sunny“' „warm“' „high“' „strong“' „cool“' „change“" „yes“]]']
Specific Bundary after   4 Instance is   ['[„sunny“' „warm“' „normal“' „strong“' „warm“' „same“"
  ' „yes“]']
```

Generic Boundary after 4 Instance is [['?', '?', '?', '?', '?', '?', '?'], ['?','?', '?', '?', '?', '?', '?'], ['?', '?', '?', '?', '?', '?', '?'], ['?', '?', '?',
'?', '?', '?', '?'], ['?', '?', '?', '?', '?', '?', '?'], ['?', '?', '?', '?', '?','?', '?'], ['?', '?', '?', '?', '?', '?', '?']]

Final Specific_h:
['[„sunny“' ' „warm“' ' „normal“' ' „strong“' ' „warm“' ' „same“" „yes“]']

Final General_h:
[['?', '?', '?', '?', '?', '?', '?'], ['?', '?', '?', '?', '?', '?', '?'], ['?',
'?', '?', '?', '?', '?', '?'], ['?', '?', '?', '?', '?', '?', '?'], ['?', '?', '?',
'?', '?', '?', '?'], ['?', '?', '?', '?', '?', '?', '?'], ['?', '?', '?', '?', '?','?', '?']]

Sample Viva Questions:

1. What is the candidate elimination algorithm used for?

The candidate elimination algorithm incrementally builds the version space given a hypothesis space H and a set E of examples. The examples are added one by one; eachexample possibly shrinks the version space by removing the hypotheses that are inconsistent with the example.

2. What is inductive bias in candidate elimination algorithm?

The inductive bias (also known as learning bias) of a learning algorithm is the set ofassumptions that the learner uses to predict outputs given inputs that it has not encountered.

3. What is inductive bias in ML?

Inductive bias is anything which makes the algorithm learn one pattern instead of another pattern (e.g. step-functions in decision trees instead of continous function in a linear regression model). In machine learning, one aims to construct algorithms that areable to learn to predict a certain target output.

4. Which is better supervised or unsupervised?

Supervised learning model produces an accurate result. Unsupervised learning model may give less accurate result as compared to supervised learning. Supervised learning isnot close to true Artificial intelligence as in this, we first train the model for each data, and then only it can predict the correct output.

5. What is the difference between version space and candidate eliminationalgorithm?

All the functions consistent with positive and negative objects (which means they classifythem correctly) constitute the version space of the target concept. The Candidate Elimination Algorithm (CEA) relies on a partial ordering of hypotheses to find the versionspace.