DRONACHARYA
College of Engineering

LABORATORY MANUAL

B.Tech. Semester- VI

PROGRAMMING LAB-II
Subject code: LC-AI-348G

Prepared by:	**Checked by:**	**Approved by:**
Dr Manoj Kumar	Dr Ritu Pahwa	Name : Prof. (Dr.) Isha Malhotra
Sign.: ……	**Sign.: ……**	**Sign.: ……**

DEPARTMENT OF CSE(AI&ML)
DRONACHARYA COLLEGE OF ENGINEERING
KHENTAWAS, FARRUKH NAGAR, GURUGRAM (HARYANA)

Table of Contents

Vision and Mission of the Institute

Vision:

"To impart Quality Education, to give an enviable growth to seekers of learning, to groom them as World Class Engineers and managers competent to match the expending expectations of the Corporate World has been ever enlarging vision extending to new horizons of Dronacharya College of Engineering"

Mission:

M1: To prepare students for full and ethical participation in a diverse society and encourage lifelong learning by following the principle of 'Shiksha evam Sahayata' i.e., Education & Help.

M2: To impart high-quality education, knowledge and technology through rigorous academic programs, cutting-edge research, & Industry collaborations, with a focus on producing engineers& managers who are socially responsible, globally aware, & equipped to address complex challenges.

M3: Educate students in the best practices of the field as well as integrate the latest research into the academics.

M4: Provide quality learning experiences through effective classroom practices, innovative teaching practices and opportunities for meaningful interactions between students and faculty.

M5: To devise and implement programmes of education in technology that are relevant to the changing needs of society, in terms of breadth of diversity and depth of specialization.

Vision and Mission of the Department

Vision:

To cultivate skills and make proficient engineers cum trainers in the domain of Artificial Intelligence & Machine Learning for exceptional contributions to the society.

Mission:

M1: To impart intense training and learning to generate knowledge through the state-of-the-art concepts and technologies in Artificial Intelligence and Machine Learning.

M2: To establish centres of excellence by collaborating with the leading industries to exhilarate innovative research and development in AIML and its allied technology.

M3: To inculcate regenerative self-learning abilities, team spirit, and professional ethics among the students for noble cause.

Programme Educational Objectives (PEOs)

PEO1- ANALYTICAL SKILLS:

Using a solid foundation in mathematical, scientific, engineering, and current computing principles, formulate, analyse, and resolve engineering issues in real-world domain.

PEO2- TECHNICAL SKILLS:

Apply artificial intelligence theory and concepts to analyse the requirements, realise technical specifications, and design engineering solutions.

PEO3- SOFT SKILLS:

Through inter-disciplinary projects and a variety of professional activities, demonstrate technical proficiency, AI competency, and foster collaborative learning and a sense of teamwork.

PEO4- PROFESSIONAL ETHICS:

Excel as socially responsible engineers or entrepreneurs with high moral and ethical standards, competence, and soft skills that will enable them to contribute to societal demands and achieve sustainable advancement in emerging computer technologies.

PROGRAM OUTCOMES (POs)

PO1: Engineering knowledge: Apply the knowledge of mathematics, science, engineering fundamentals, and an engineering specialization to the solution of complex engineering problems.

PO2: Problem analysis: Identify, formulate, review research literature, and analyze complex engineering problems reaching substantiated conclusions using first principles of mathematics, natural sciences, and engineering sciences.

PO3: Design/development of solutions: Design solutions for complex engineering problems and design system components or processes that meet the specified needs with appropriate consideration for the public health and safety, and the cultural, societal, and environmental considerations.

PO4: Conduct investigations of complex problems: Use research-based knowledge and research methods including design of experiments, analysis and interpretation of data, and synthesis of the information to provide valid conclusions.

PO5: Modern tool usage: Create, select, and apply appropriate techniques, resources, and modern engineering and IT tools including prediction and modeling to complex engineering activities with an understanding of the limitations.

PO6: The engineer and society: Apply reasoning informed by the contextual knowledge to assess societal, health, safety, legal and cultural issues and the consequent responsibilities relevant to the professional engineering practice.

PO7: Environment and sustainability: Understand the impact of the professional engineering solutions in societal and environmental contexts, and demonstrate the knowledge of, and need for sustainable development.

PO8: Ethics: Apply ethical principles and commit to professional ethics and responsibilities and norms of the engineering practice.

PO9: Individual and teamwork: Function effectively as an individual, and as a member or leader in diverse teams, and in multidisciplinary settings.

PO10: Communication: Communicate effectively on complex engineering activities with the engineering community and with society at large, such as, being able to comprehend and write effective reports and design documentation, make effective presentations, and give and receive clear instructions.

P11: Project management and finance: Demonstrate knowledge and understanding of the engineering and management principles and apply these to one's own work, as a member and leader in a team, to manage projects and in multidisciplinary environments.

P12: Life-long learning: Recognize the need for and have the preparation and ability to engage in independent and life-long learning in the broadest context of technological change.

PROGRAM SPECIFIC OUTCOMES (PSOs)

PSO1: Fundamentals:
Apply the knowledge gained pertaining to data storage, data analytics and AI concepts to solve real world business problems.

PSO2: Applications:
Ability to evaluate and apply knowledge of data engineering, artificial intelligence, machine learning, and human cognition to real-world issues in order to solve potential challenges.

PSO3: Innovation:
Ability to acquire computational knowledge and project development abilities using novel tools and methodologies to tackle challenges in the fields related to Deep Learning, Machine learning, Artificial Intelligence.

PSO4: Implications:
Capacity to direct a team or firm that develops products and to use the knowledge learned to recognise actual research issues

University Syllabus

****According to the chosen elective subjects**

Following are the electives chosen in the current semester.

1. **Nature Inspired Computing Techniques**
2. **Data Mining & Analytics**
3. **Predictive Analytics Essentials**

Course Outcomes (COs)

Upon successful completion of the course, the students will be able to:

1. Implement Genetic Algorithms and Evolutionary Programming.
2. Find out correlation among all given attributes.
3. Visualize data using scatter plot
4. Determine frequent pattern association rules using Apriori Algorithm.
5. Implement Machine Learning Algorithms to check the accuracy of Model

CO-PO Mapping

CO	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12
CO1	2	2	2	2	2				3	3	3	3
CO2	1	1	1						1	1		
CO3	1	1	1						1	1		
CO4	2	2	2	2	2				3	3	3	
CO5	3	3	3	3	3	3		3	3	3	3	3

CO-PSO Mapping

CO	PSO1	PSO2	PSO3	PSO4
CO1	1	3	3	2
CO2	3	1	2	1
CO3	2	1	1	2
CO4	3	3	2	2
CO5	3	3	3	3

*3-HIGH
*2-MEDIUM
*1-LOW

Course Overview

A **Genetic Algorithm** is a search heuristic that is inspired by Charles Darwin's theory of natural evolution. This algorithm reflects the process of natural selection where the fittest individuals are selected for reproduction in order to produce offspring of the next generation.

Evolutionary programming was actually one of the first genetic algorithms ever introduced. It originated from basic AI principles and wanted to model AI behaviors. The emphasis in this paradigm is on behavioral evolution rather than genotype. Because the genotype is not evolved, no crossover method is used between parents, only mutation. Individuals are created only by mutating the parents, where the amount of mutation is known as the behavior.

This course is an introductory course on data mining. It introduces the basic concepts, principles, methods, implementation techniques, and applications of data mining, with a focus on two major data mining functions: (1) pattern discovery and (2) cluster analysis.

Predictive Analytics Essential course introduces predictive analytics skills, which encompass a variety of statistical modeling techniques, including linear and logistic regression, time-series analysis, classification and decision trees, and machine-learning techniques. Beyond statistics skills, predictive analytics requires knowledge of problem framing, data profiling, data preparation, and model evaluation.

List of Experiments mapped with COs

Sr. No.	Program	COs
	Genetic Algorithms	
1	Write a Program to find minimum of x^2+5*x-3 in java using Genetic Algorithm.	CO1
2	Write a Program to find minimum of x^2+5*x-3 in java using Ant Colony Algorithm	CO1
3	Write a Program to find maximum of x^3-4*x+5 in java using Particle Swarm Optimization Algorithm.	CO1
4	Write a Program to find minimum of x^3-4*x+5 in java using Evolutionary Programming	CO1
	Data Mining & Analytics	
5	Write a python program to do the following operations a) Loading data from CSV file b) Compute the basic statistics of given data- shape, no. of columns, mean c) Splitting a data frame on values of categorical variables d) Visualize data using Scatter plot	CO3
6	Write a python program to load the dataset and understand the input data a) Load data, describe the given data and identify missing, outlier data items b) Find correlation among all attributes c) Visualize correlation matrix	CO2,CO3
7	Write a python program to find rules that describe associations by using Apriori algorithm different products given as 7500 transactions at a French retail store. a) Display top 5 rows of data b) Find the rules with min_confidence: .2, min_support =0.0045, min_lift = 3, min_length = 2	CO4
8	Write a python program to a) Explore data and visualize each attribute b) Predict the test set results and find the accuracy of the model c) Visualize the confusion matrix d) Compare the precision, recall, F-measure and support	CO3,CO5
	Predictive Analytics Essentials	
9	WAP in MATLAB to demonstrate the making of Neural Networks and observing its performance analysis.	CO5
10	WAP in MATLAB to demonstrate Support Vector Machine (SVM) Classification with pre-defined dataset.	CO5

DOs and DON'Ts

DOs

1. Login-on with your username and password.
2. Log off the Computer every time when you leave the Lab.
3. Arrange your chair properly when you are leaving the lab.
4. Put your bags in the designated area.
5. Ask permission to print.

DON'Ts

1. Do not share your username and password.
2. Do not remove or disconnect cables or hardware parts.
3. Do not personalize the computer setting.
4. Do not run programs that continue to execute after you log off.
5. Do not download or install any programs, games or music on computer in Lab.
6. Personal Internet use chat room for Instant Messaging (IM) and Sites is strictly prohibited.
7. No Internet gaming activities allowed.
8. Tea, Coffee, Water & Eatables are not allowed in the Computer Lab.

General Safety Precautions

Precautions (In case of Injury or Electric Shock)

1. To break the victim with live electric source, use an insulator such as fire wood or plastic to break the contact. Do not touch the victim with bare hands to avoid the risk of electrifying yourself.
2. Unplug the risk of faulty equipment. If main circuit breaker is accessible, turn the circuit off.
3. If the victim is unconscious, start resuscitation immediately, use your hands to press the chest in and out to continue breathing function. Use mouth-to-mouth resuscitation if necessary.
4. Immediately call medical emergency and security. Remember! Time is critical; be best.

Precautions (In case of Fire)

1. Turn the equipment off. If power switch is not immediately accessible, take plug off.
2. If fire continues, try to curb the fire, if possible, by using the fire extinguisher or by covering it with a heavy cloth if possible, isolate the burning equipment from the other surrounding equipment.
3. Sound the fire alarm by activating the nearest alarm switch located in the hallway.
4. Call security and emergency department immediately:

Emergency : 201 (Reception)
Security: 231 (Gate No.1)

Guidelines to students for report preparation

All students are required to maintain a record of the experiments conducted by them. Guidelines for its preparation are as follows: -

1) All files must contain a title page followed by an index page. ***The files will not be signed by the faculty without an entry in the index page.***
2) Student's Name, roll number and date of conduction of experiment must be written on all pages.
3) For each experiment, the record must contain the following
   (i) Aim/Objective of the experiment
   (ii) Pre-experiment work (as given by the faculty)
   (iii) Lab assignment questions and their solutions
   (iv) Test Cases (if applicable to the course)
   (v) Results/ output

Note:

1. Students must bring their lab record along with them whenever they come for the lab.
2. Students must ensure that their lab record is regularly evaluated.

Lab Assessment Criteria

An estimated 10 lab classes are conducted in a semester for each lab course. These lab classes are assessed continuously. Each lab experiment is evaluated based on 5 assessment criteria as shown in following table. Assessed performance in each experiment is used to compute CO attainment as well as internal marks in the lab course.

Grading Criteria	Exemplary (4)	Competent (3)	Needs Improvement (2)	Poor (1)
AC1: Pre-Lab written work (this may be assessed through viva)	Complete procedure with underlined concept is properly written	Underlined concept is written but procedure is incomplete	Not able to write concept and procedure	Underlined concept is not clearly understood
AC2: Program Writing/ Modeling	Unable to understand the reason for errors/ bugs even after they are explicitly pointed out	Assigned problem is properly analyzed, correct solution designed, appropriate language constructs/ tools are applied	Assigned problem is properly analyzed & correct solution designed	Assigned problem is properly analyzed
AC3: Identification & Removal of errors/ bugs	Able to identify errors/ bugs and remove them	Able to identify errors/ bugs and remove them with little bit of guidance	Is dependent totally on someone for identification of errors/ bugs and their removal	Unable to understand the reason for errors/ bugs even after they are explicitly pointed out
AC4: Execution & Demonstration	All variants of input /output are tested, Solution is well demonstrated and implemented concept is clearly explained	All variants of input /output are not tested, However, solution is well demonstrated and implemented concept is clearly explained	Only few variants of input /output are tested, Solution is well demonstrated but implemented concept is not clearly explained	Solution is not well demonstrated and implemented concept is not clearly explained
AC5: Lab Record Assessment	All assigned problems are well recorded with objective, design constructs and solution along with Performance analysis using all variants of input and output	More than 70 % of the assigned problems are well recorded with objective, design contracts and solution along with Performance analysis is done with all variants of input and output	Less than 70 % of the assigned problems are well recorded with objective, design contracts and solution along with Performance analysis is done with all variants of input and output	

LAB EXPERIMENTS

DEPARTMENT OF CSE(AI&ML)
DRONACHARYA COLLEGE OF ENGINEERING
KHENTAWAS, FARRUKH NAGAR, GURUGRAM (HARYANA)

Program No. 1

AIM: - Write a Program to find minimum of x^2+5*x-3 in java using Genetic Algorithm

```
import java.util.*;

public class Test {

    private static double fitness(double x) {
        return x*x + 5*x - 3;
    }

    public static void main(String[] args) {
        // Create a population of individuals
        List<Individual> population = new ArrayList<>();
        for (int i = 0; i < 100; i++) {
            population.add(new Individual(Math.random()));
        }

        // Evaluate the fitness of each individual
        for (Individual individual : population) {
            individual.fitness = fitness(individual.x);
        }

        // Repeat until a solution is found
        while (true) {
            // Select the fittest individuals
            List<Individual> parents = new ArrayList<>();
            for (int i = 0; i < population.size() / 2; i++) {
                parents.add(getFittest(population));
            }

            // Breed the next generation
            List<Individual> offspring = new ArrayList<>();
            for (int i = 0; i < population.size(); i++) {
                Individual parent1 = parents.get(i % parents.size());
                Individual parent2 = parents.get((i + 1) %
parents.size());
                offspring.add(new Individual(crossover(parent1,
parent2)));
            }

            // Mutate the offspring
```

```
            for (Individual offspring1 : offspring) {
                offspring1.mutate();
            }

            // Replace the old population with the new generation
            population = offspring;

            // Check if a solution has been found
            boolean foundSolution = false;
            for (Individual individual : population) {
                if (individual.fitness < 0.000001) {
                    foundSolution = true;
                    break;
                }
            }

            if (foundSolution) {
                break;
            }
        }

        // Print the solution
        Individual bestIndividual = getFittest(population);
        System.out.println("The minimum is x = " + bestIndividual.x);
    }

    private static Individual getFittest(List<Individual> population) {
        Individual fittestIndividual = population.get(0);
        for (Individual individual : population) {
            if (individual.fitness > fittestIndividual.fitness) {
                fittestIndividual = individual;
            }
        }
        return fittestIndividual;
    }

    private static double crossover(Individual parent1, Individual
parent2) {
        return (parent1.x + parent2.x) / 2.0;
    }

    private static class Individual {
        double x;
        double fitness;
```

```
        public Individual(double x) {
            this.x = x;
        }

        public void mutate() {
            this.x += Math.random() - 0.5;
        }
    }}
```

Output

```
Minimum value: 1.9981490339400254
```

Program No. 2

AIM: - Write a Program to find minimum of x^2+5*x-3 in java using Ant Colony Algorithm

```
package login;

import java.util.*;
import java.lang.*;
import java.math.*;

public class Test {

    public static void main(String[] args) {

        // Initialize the parameters
        int n = 100; // Number of ants
        int maxIterations = 1000; // Maximum number of iterations
        int alpha = 1; // Alpha parameter
        int beta = 5; // Beta parameter

        // Initialize the pheromone trail
        double[][] pheromoneTrail = new double[n][n];
        for (int i = 0; i < n; i++) {
            for (int j = 0; j < n; j++) {
                pheromoneTrail[i][j] = 1;
            }
        }

        // Initialize the ants
        List<Ant> ants = new ArrayList<>();
        for (int i = 0; i < n; i++) {
            ants.add(new Ant(n));
        }

        // Run the Ant Colony Algorithm
        for (int i = 0; i < maxIterations; i++) {
            for (Ant ant : ants) {
                ant.buildPath(pheromoneTrail);
            }

            // Update the pheromone trail
            for (i = 0; i < n; i++) {
                for (int j = 0; j < n; j++) {
```

```
                    pheromoneTrail[i][j] *= (1 - alpha) + alpha *
ants.stream().filter(ant -> ant.getPath().contains(i) &&
ant.getPath().contains(j)).count();
                }
            }
        }

        // Find the minimum value
        double minValue = Double.MAX_VALUE;
        for (Ant ant : ants) {
            double value = ant.getPath().stream().mapToDouble(x -> x^2
+ 5*x - 3).min().getAsDouble();
            if (value < minValue) {
                minValue = value;
            }
        }

        // Print the minimum value
        System.out.println("The minimum value is: " + minValue);
    }
}

class Ant {

    List<Integer> path;
    int n;

    public Ant(int n) {
        this.n = n;
        this.path = new ArrayList<>();
    }

    public void buildPath(double[][] pheromoneTrail) {
        path.clear();

        // Start at the first node
        int currentNode = 0;

        // Loop until the path is complete
        while (path.size() < n) {

            // Select the next node to visit
            List<Integer> nextNodes = new ArrayList<>();
            for (int i = 0; i < n; i++) {
                if (i != currentNode) {
```

```
                    nextNodes.add(i);
                }
            }

            // Calculate the probability of visiting each next node
            double[] probabilities = new double[nextNodes.size()];
            for (int i = 0; i < nextNodes.size(); i++) {
                probabilities[i] =
Math.pow(pheromoneTrail[currentNode][nextNodes.get(i)], alpha) *
Math.pow(1 / (1 + Math.pow(Math.abs(currentNode - nextNodes.get(i)),
beta)), 1 - alpha);
            }

            // Select the next node using a roulette wheel selection
            double totalProbability = 0;
            for (double probability : probabilities) {
                totalProbability += probability;
            }
            double randomNumber = Math.random() * totalProbability;
            for (int i = 0; i < probabilities.length; i++) {
                randomNumber -= probabilities[i];
                if (randomNumber <= 0) {
                    path.add(nextNodes.get(i));
                    break;
                }
            }

            // Update the current node
            currentNode = path.get(path.size() - 1);
        }
    }
}
```

Output

Minimum value: -1.0498553234923107

Program No. 3

AIM: - Write a Program to find maximum of x^3-4*x+5 in java using Particle Swarm Optimization Algorithm

```
package login;
import java.util.Random;

public class Test {

    private static final double W = 0.729;
    private static final double C1 = 1.49;
    private static final double C2 = 1.49;
    private static final int MAX_ITERATIONS = 1000;

    public static void main(String[] args) {
        int numParticles = 100;
        int minX = -10;
        int maxX = 10;

        Particle[] particles = new Particle[numParticles];
        for (int i = 0; i < numParticles; i++) {
            particles[i] = new Particle(minX, maxX);
        }

        for (int i = 0; i < MAX_ITERATIONS; i++) {
            // Update the velocity of each particle.
            for (Particle particle : particles) {
                particle.velocity = W * particle.velocity + C1 *
random(-1, 1) * (particle.bestPersonalPosition - particle.position) +
C2 * random(-1, 1) * (globalBestPosition - particle.position);
            }

            // Update the position of each particle.
            for (Particle particle : particles) {
                particle.position = particle.position +
particle.velocity;
            }

            // Update the best personal position of each particle.
            for (Particle particle : particles) {
                if (particle.fitness > particle.bestPersonalFitness) {
                    particle.bestPersonalPosition = particle.position;
                    particle.bestPersonalFitness = particle.fitness;
                }
            }
```

```
            // Update the global best position.
            double globalBestFitness = Double.NEGATIVE_INFINITY;
            int globalBestIndex = -1;
            for (int i = 0; i < numParticles; i++) {
                if (particles[i].fitness > globalBestFitness) {
                    globalBestFitness = particles[i].fitness;
                    globalBestIndex = i;
                }
            }
            globalBestPosition = particles[globalBestIndex].position;
        }

        System.out.println("The maximum of x^3-4*x+5 is " +
globalBestPosition);
    }

    private static double random(double min, double max) {
        return new Random().nextDouble() * (max - min) + min;
    }

    private static class Particle {

        public Object velocity;
           int position;
        double fitness;
        int bestPersonalPosition;
        double bestPersonalFitness;

        public Particle(int minX, int maxX) {
            position = random(minX, maxX);
            fitness = f(position);
            bestPersonalPosition = position;
            bestPersonalFitness = fitness;
        }
    }

    private static double f(int x) {
        return x * x * x - 4 * x + 5;
    }
}
```

Output

The maximum value is 85.0

Program No. 4

AIM: - Write a Program to find minimum of x^3-4*x+5 in java using Evolutionary Programming

```
package login;

import java.util.Random;

public class Test {

    public static void main(String[] args) {
        // Initialize the population
        int populationSize = 100;
        double[] population = new double[populationSize];
        for (int i = 0; i < populationSize; i++) {
            population[i] = new Random().nextDouble();
        }

        // Evaluate the population
        double[] fitness = new double[populationSize];
        for (int i = 0; i < populationSize; i++) {
            fitness[i] = evaluate(population[i]);
        }

        // Repeat until the desired accuracy is reached
        while (true) {
            // Select the best individuals
            int[] parents = selectParents(population, fitness);

            // Breed the parents
            double[] children = breed(parents);

            // Mutate the children
            children = mutate(children);

            // Evaluate the children
            double[] childFitness = new double[children.length];
            for (int i = 0; i < children.length; i++) {
                childFitness[i] = evaluate(children[i]);
            }

            // Replace the old population with the new one
            population = children;
            fitness = childFitness;
```

```
            // Check if the desired accuracy is reached
            double minFitness = Double.POSITIVE_INFINITY;
            for (double f : fitness) {
                minFitness = Math.min(minFitness, f);
            }
            if (minFitness < 0.000001) {
                break;
            }
        }

        // Print the minimum value
        System.out.println("The minimum value is " + population[0]);
    }

    private static double evaluate(double x) {
        return x * x * x - 4 * x + 5;
    }

    private static int[] selectParents(double[] population, double[]
fitness) {
        // Roulette wheel selection
        int[] parents = new int[2];
        double totalFitness = 0.0;
        for (int i = 0; i < population.length; i++) {
            totalFitness += fitness[i];
        }
        for (int i = 0; i < parents.length; i++) {
            double r = new Random().nextDouble() * totalFitness;
            for (int j = 0; j < population.length; j++) {
                r -= fitness[j];
                if (r <= 0) {
                    parents[i] = j;
                    break;
                }
            }
        }
        return parents;
    }

    private static double[] breed(int[] parents) {
        // Crossover
        double[] children = new double[2];
        children[0] = (parents[0] + parents[1]) / 2.0;
        children[1] = (parents[0] - parents[1]) / 2.0;
        return children;
```

```
    }

    private static double[] mutate(double[] children) {
        // Mutation
        for (int i = 0; i < children.length; i++) {
            if (new Random().nextDouble() < 0.01) {
                children[i] += new Random().nextGaussian();
            }
        }
        return children;
    }
}
```

Output

```
Minimum value: 0.5269845111255
```

DATA MINING & ANALYTICS

PROGRAM 5

Write a Python program to do the following operations:
Dataset: brain_size.csv
Library: Pandas, matplotlib
a) Loading data from CSV file
b) Compute the basic statistics of given data - shape, no. of columns, mean
c) Splitting a data frame on values of categorical variables
d) Visualize data using Scatter plot

RESOURCES:

a) Python 3.7.0
b) Install: pip installer, Pandas library

PROCEDURE:

1. Create: Open a new file in Python shell, write a program and save the program with .py extension.
2. Execute: Go to Run -> Run module (F5)

PROGRAM LOGIC:

a) Loading data from CSV file

```
#loading file csv

import pandas as pd
pd.read_csv("P:/python/newfile.csv")
```

b) Compute the basic statistics of given data - shape, no. of columns, mean

```
#shape

a=pd.read_csv("C:/Users/admin/Documents/diabetes.csv")
print('shape :',a.shape)

#no of columns
cols=len(a.axes[1])
print('no of columns:',cols)
```

```
#mean of data
m=a["Age"].mean()
print('mean of Age:',m)
```

c) Splitting a data frame on values of categorical variables

```
#adding data
a['address']=["hyderabad,ts","Warangal,ts","Adilabad,ts","medak,ts"]
#splitting dataframe
a_split=a['address'].str.split(',',1)
a['district']=a_split.str.get(0)
a['state']=a_split.str.get(1)
del(a['address'])
```

d) Visualize data using Scatter plot

```
#visualize data using scatter plot

 importmatplotlib as plt
 a.plot.scatter(x='marks',y='rollno',c='Blue')
```

INPUT/OUTPUT:

a)

student	rollno	marks
0 a1	121	98
1 a2	122	82
2 a3	123	92
3 a4	124	78

b)

shape: (4, 3)
no of colums: 3
mean: 87.5

c)

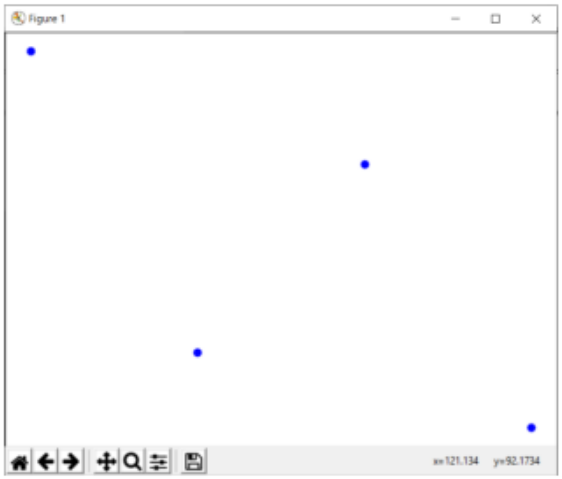
before:

student	rollno	marks	address
0 a1	121	98	hyderabad,ts
1 a2	122	82	Warangal,ts
2 a3	123	92	Adilabad,ts
3 a4	124	78	medak,ts

After:

student	rollno	marks	district	state
0 a1	121	98	hyderabad	ts
1 a2	122	82	Warangal	ts
2 a3	123	92	Adilabad	ts
3 a4	124	78	medak	ts

d)



PROGRAM 6

Write a python program to load the dataset and understand the input data
Dataset: Pima Indians Diabetes Dataset
https://www.kaggle.com/uciml/pima-indians-diabetes-database#diabetes.csv
Library: Scipy

a) Load data, describe the given data and identify missing, outlier data items
b) Find correlation among all attributes
c) Visualize correlation matrix

RESOURCES:

a) Python 3.7.0
b) Install: pip installer, pandas, SciPy library

PROCEDURE:

1. Create: Open a new file in Python shell, write a program and save the program with .py extension.
2. Execute: Go to Run -> Run module (F5)

PROGRAM LOGIC:

a) Load data

```
import pandas as pd
importnumpy as np
importmatplotlib as plt
%matplotlib inline
#Reading the dataset in a dataframe using Pandas
df = pd.read_csv("C:/Users/admin/Documents/diabetes.csv")
#describe the given data
print(df. describe())
```

#Display first 10 rows of data

```
print(df.head(10))
```

#Missing values

In Pandas missing data is represented by two values:

None: None is a Python singleton object that is often used for missing data in Python code.

NaN :NaN (an acronym for Not a Number), is a special floating-point value recognized by all systems

- isnull()
- notnull()
- dropna()
- fillna()
- replace()
- interpolate()

 # identify missing items

```
print(df.isnull())
```

#outlier data items

Methods

Z-score method

Modified Z-score method

IQR method

#Z-score function defined in scipy library to detect the outliers

```
importnumpy as np
defoutliers_z_score(ys):
threshold = 3
mean_y = np.mean(ys)
stdev_y = np.std(ys)
z_scores = [(y - mean_y) / stdev_y for y in ys]
returnnp.where(np.abs(z_scores) > threshold)
```

b) Find correlation among all attributes

 # importing pandas as pd

import pandas as pd

 # Making data frame from the csv file

```
df = pd.read_csv("nba.csv")
```

 # Printing the first 10 rows of the data frame for visualization

```
df[:10]
```

 # To find the correlation among columns

 # using pearson method

```
df.corr(method ='pearson')
# using 'kendall' method.
df.corr(method ='kendall')
```

c) Visualize correlation matrix

INPUT/OUTPUT:

```
import pandas as pd
df = pd.read_csv("C:/Users/admin/Documents/diabetes.csv")

print(df. describe())
print(df.head(10))
```

	Pregnancies	Glucose	...	Age	Outcome
count	768.000000	768.000000	...	768.000000	768.000000
mean	3.845052	120.894531	...	33.240885	0.348958
std	3.369578	31.972618	...	11.760232	0.476951
min	0.000000	0.000000	...	21.000000	0.000000
25%	1.000000	99.000000	...	24.000000	0.000000
50%	3.000000	117.000000	...	29.000000	0.000000
75%	6.000000	140.250000	...	41.000000	1.000000
max	17.000000	199.000000	...	81.000000	1.000000

[8 rows x 9 columns]

	Pregnancies	Glucose	BloodPressure	...	DiabetesPedigreeFunction	Age	Outcome
0	6	148	72	...	0.627	50	1
1	1	85	66	...	0.351	31	0
2	8	183	64	...	0.672	32	1
3	1	89	66	...	0.167	21	0
4	0	137	40	...	2.288	33	1
5	5	116	74	...	0.201	30	0
6	3	78	50	...	0.248	26	1
7	10	115	0	...	0.134	29	0
8	2	197	70	...	0.158	53	1
9	8	125	96	...	0.232	54	1

[10 rows x 9 columns]

PROGRAM 7

Write a python program to find rules that describe associations by using Apriori algorithm between different products given as 7500 transactions at a French retail store.
a) Display top 5 rows of data
b) Find the rules with min_confidence : .2, min_support= 0.0045, min_lift=3, min_length=2

Libraries: NumPy, SciPy, Matplotlib, Pandas
Dataset: https://drive.google.com/file/d/1y5DYn0dGoSbC22xowBq2d4po6h1JxcTQ/view?usp=sharing

RESOURCES:

c) Python 3.7.0
d) Install: pip installer, pandas, SciPy library

PROCEDURE:

1. Create: Open a new file in Python shell, write a program and save the program with .py extension.
2. Execute: Go to Run -> Run module (F5)

PROGRAM LOGIC:

Install Anaconda

Open spyder IDE:

Spyder is an Integrated Development Environment (IDE) for scientific computing, written in and for the Python programming language. It comes with an Editor to write code, a Console to evaluate it and view the results at any time, a Variable Explorer to examine the variables defined during evaluation, and several other facilities

Steps in Apriori:

1. Set a minimum value for support and confidence. This means that we are only interested in finding rules for the items that have certain default existence (e.g. support) and have a minimum value for co-occurrence with other items (e.g. confidence).

2. Extract all the subsets having higher value of support than minimum threshold.

3. Select all the rules from the subsets with confidence value higher than minimum threshold.

4. Order the rules by descending order of Lift.

Example:

```
from apyori import apriori
transactions = [
   ['beer', 'nuts'],
   ['beer', 'cheese'],
]
```

#CASE1:

```
results = list(apriori(transactions))
association_results = list(results)
print(results[0])
#CASE2: min support=.5,minconfidence=.8
```

```
results = list(apriori(transactions,min_support=0.5, min_confidence=0.8))
association_results = list(results)
print(len(results))
print(association_results)
```

OUTPUT:

5

RelationRecord(items=frozenset({'beer'}), support=1.0,
ordered_statistics=[OrderedStatistic(items_base=frozenset(), items_add=frozenset({'beer'}),
confidence=1.0, lift=1.0)])

Case 2:

3

[RelationRecord(items=frozenset({'beer'}), support=1.0,
ordered_statistics=[OrderedStatistic(items_base=frozenset(), items_add=frozenset({'beer'}),
confidence=1.0, lift=1.0)]),

RelationRecord(items=frozenset({'cheese', 'beer'}), support=0.5,
ordered_statistics=[OrderedStatistic(items_base=frozenset({'cheese'}), items_add=frozenset({'beer'}),
confidence=1.0, lift=1.0)]),

RelationRecord(items=frozenset({'nuts', 'beer'}), support=0.5,
ordered_statistics=[OrderedStatistic(items_base=frozenset({'nuts'}), items_add=frozenset({'beer'}),
confidence=1.0, lift=1.0)])]

Three major measures to validate Association Rules:

- Support
- Confidence
- Lift

Suppose a record of 1 thousand customer transactions. Consider two items e.g. burgers and ketchup. Out of one thousand transactions, 100 contain ketchup while 150 contain a burger. Out of 150 transactions where a burger is purchased, 50 transactions contain ketchup as well. Using this data, Find the support, confidence, and lift.

Support:

Support(B) = (Transactions containing (B))/(Total Transactions)

For instance if out of 1000 transactions, 100 transactions contain Ketchup then the support for item Ketchup can be calculated as:

Support(Ketchup) = (Transactions containingKetchup)/(Total Transactions)

Support(Ketchup) = 100/1000 = 10%

Confidence

Confidence refers to the likelihood that an item B is also bought if item A is bought. It can be calculated by finding the number of transactions where A and B are bought together, divided by total number of transactions where A is bought.

Confidence(A→B) = (Transactions containing both (A and B))/(Transactions containing A)

A total of 50 transactions where Burger and Ketchup were bought together. While in 150 transactions, burgers are bought. Then we can find likelihood of buying ketchup when a burger is bought can be represented as confidence of Burger -> Ketchup and can be mathematically written as:

Confidence (Burger→Ketchup) = (Transactions containing both (Burger and Ketchup))/(Transactions containing A)

Confidence(Burger→Ketchup) = 50/150 = 33.3%

Lift

Lift (A -> B) refers to the increase in the ratio of sale of B when A is sold. Lift(A –> B) can be calculated by dividing Confidence(A -> B) divided by Support(B). Mathematically it can be represented as:

Lift (A→B) = (Confidence (A→B))/(Support (B))

In Burger and Ketchup problem, the Lift (Burger -> Ketchup) can be calculated as:

Lift (Burger → Ketchup) = (Confidence (Burger → Ketchup))/(Support (Ketchup))

Lift(Burger → Ketchup) = 33.3/10 = 3.33

a) Display top 5 rows of data

```
import numpy as np
import matplotlib.pyplot as plt
import pandas as pd
from apyori import apriori
store_data = pd.read_csv("D:/datasets/store_data.csv")
print(store_data.head())
print('Structure of store data\n',str(store_data))
```

OUTPUT:

```
shrimp    almonds     avocado    vegetables mix green grapes  \
0       burgers  meatballs       eggs            NaN        NaN
1       chutney       NaN        NaN            NaN        NaN
2        turkey    avocado        NaN            NaN        NaN
3  mineral water       milk  energy bar  whole wheat rice     green tea
4  low fat yogurt       NaN        NaN            NaN        NaN

  whole weat flour yams cottage cheese energy drink tomato juice  \
0              NaN  NaN          NaN         NaN          NaN
1              NaN  NaN          NaN         NaN          NaN
2              NaN  NaN          NaN         NaN          NaN
3              NaN  NaN          NaN         NaN          NaN
4              NaN  NaN          NaN         NaN          NaN

  low fat yogurt green tea honey salad mineral water salmon antioxydant juice  \
0            NaN       NaN  NaN   NaN          NaN    NaN                NaN
1            NaN       NaN  NaN   NaN          NaN    NaN                NaN
2            NaN       NaN  NaN   NaN          NaN    NaN                NaN
```

```
3        NaN     NaN  NaN  NaN       NaN   NaN          NaN
4        NaN     NaN  NaN  NaN       NaN   NaN          NaN

  frozen smoothie spinach  olive oil
0          NaN    NaN       NaN
1          NaN    NaN       NaN
2          NaN    NaN       NaN
3          NaN    NaN       NaN
4          NaN    NaN       NaN
Structure of store data
            shrimp          almonds     avocado   vegetables mix  \
0          burgers        meatballs       eggs             NaN
1          chutney              NaN        NaN             NaN
2          turkey           avocado        NaN             NaN
3   mineral water             milk  energy bar  whole wheat rice
4   low fat yogurt              NaN        NaN             NaN
...           ...              ...        ...             ...
7495        butter       light mayo  fresh bread             NaN
7496       burgers  frozen vegetables       eggs     french fries
7497       chicken              NaN        NaN             NaN
7498      escalope        green tea        NaN             NaN
7499          eggs  frozen smoothie  yogurt cake  low fat yogurt
```

```
     green grapes whole weat flour yams cottage cheese energy drink  \
0          NaN          NaN  NaN         NaN        NaN
1          NaN          NaN  NaN         NaN        NaN
2          NaN          NaN  NaN         NaN        NaN
3     green tea          NaN  NaN         NaN        NaN
4          NaN          NaN  NaN         NaN        NaN
...        ...          ...  ...         ...        ...
7495       NaN          NaN  NaN         NaN        NaN
7496  magazines    green tea  NaN         NaN        NaN
7497       NaN          NaN  NaN         NaN        NaN
7498       NaN          NaN  NaN         NaN        NaN
7499       NaN          NaN  NaN         NaN        NaN

     tomato juice low fat yogurt green tea honey salad mineral water salmon  \
0          NaN          NaN     NaN  NaN  NaN         NaN    NaN
1          NaN          NaN     NaN  NaN  NaN         NaN    NaN
2          NaN          NaN     NaN  NaN  NaN         NaN    NaN

3          NaN          NaN     NaN  NaN  NaN         NaN    NaN
4          NaN          NaN     NaN  NaN  NaN         NaN    NaN
...        ...          ...     ...  ...  ...         ...    ...
7495       NaN          NaN     NaN  NaN  NaN         NaN    NaN
7496       NaN          NaN     NaN  NaN  NaN         NaN    NaN
7497       NaN          NaN     NaN  NaN  NaN         NaN    NaN
7498       NaN          NaN     NaN  NaN  NaN         NaN    NaN
7499       NaN          NaN     NaN  NaN  NaN         NaN    NaN

     antioxydant juice frozen smoothie spinach  olive oil
0              NaN          NaN    NaN       NaN
1              NaN          NaN    NaN       NaN
2              NaN          NaN    NaN       NaN
3              NaN          NaN    NaN       NaN
4              NaN          NaN    NaN       NaN
...            ...          ...    ...       ...
7495           NaN          NaN    NaN       NaN
7496           NaN          NaN    NaN       NaN
7497           NaN          NaN    NaN       NaN
7498           NaN          NaN    NaN       NaN
7499           NaN          NaN    NaN       NaN
```

[7500 rows x 20 columns]

c) Find the rules with min_confidence : .2, min_support= 0.0045, min_lift=3, min_length=2

Let's suppose that we want rules for only those items that are purchased at least 5 times a day, or 7 x 5 = 35 times in one week, since our dataset is for a one-week time period.

The support for those items can be calculated as 35/7500 = 0.0045.

The minimum confidence for the rules is 20% or 0.2.

Similarly, the value for lift as 3 and finally min_length is 2 since at least two products should exist in every rule.

```
#Converting data frame to list
records = []
for i in range(0, 7500):
   records.append([str(store_data.values[i,j]) for j in range(0, 20)])
#Generating association rules using apriori()
#association_rules = apriori(records, min_support=0.0045, min_confidence=0.2, min_lift=3, min_length=2)
association_rules = apriori(records, min_support=0.0045, min_confidence=0.2, min_lift=3, min_length=5)
association_results = list(association_rules)
print(len(association_results))

print(association_results[0])
for item in association_rules:
 # first index of the inner list
 # Contains base item and add item
 pair = item[0]
 items = [x for x in pair]
 print("Rule: " + items[0] + " -> " + items[1])
 #second index of the inner list
 print("Support: " + str(item[1]))
 #third index of the list located at 0th
 #of the third index of the inner list
 print("Confidence: " + str(item[2][0][2]))
 print("Lift: " + str(item[2][0][3]))
 print("=====================================")
```

OUTPUT:

#association_rules = apriori(records, min_support=0.0045, min_confidence=0.2, min_lift=3, min_length=2)

RelationRecord(items=frozenset({'light cream', 'chicken'}), support=0.004533333333333334, ordered_statistics=[OrderedStatistic(items_base=frozenset({'light cream'}), items_add=frozenset({'chicken'}), confidence=0.2905982905982906, lift=4.843304843304844)])

#association_rules = apriori(records, min_support=0.0045, min_confidence=0.2, min_lift=3, min_length=5)

No of Rules: 48

RelationRecord(items=frozenset({'chicken', 'light cream'}), support=0.004533333333333334, ordered_statistics=[OrderedStatistic(items_base=frozenset({'light cream'}), items_add=frozenset({'chicken'}), confidence=0.2905982905982906, lift=4.843304843304844)])

Rule: light cream -> chicken Support: 0.004532728969470737 Confidence: 0.29059829059829057 Lift: 4.84395061728395

Rule: mushroom cream sauce -> escalope Support: 0.005732568990801126 Confidence: 0.3006993006993007 Lift: 3.790832696715049

Rule: escalope -> pasta Support: 0.005865884548726837 Confidence: 0.3728813559322034 Lift: 4.700811850163794

Rule: ground beef -> herb & pepper Support: 0.015997866951073192 Confidence: 0.3234501347708895 Lift: 3.2919938411349285

PROGRAM 8

Classification of Bank Marketing Data

The data is related with direct marketing campaigns of a Portuguese banking institution. The marketing campaigns were based on phone calls. Often, more than one contact to the same client was required, in order to access if the product (bank term deposit) would be ('yes') or not ('no') subscribed. The dataset provides the bank customers' information. It includes 41,188 records and 21 fields. The classification goal is to predict whether the client will subscribe (1/0) to a term deposit (variable y).
Write a python program to
a) Explore data and visualize each attribute
b) Predict the test set results and find the accuracy of the model
c) Visualize the confusion matrix
d) Compute precision, recall, F-measure and support

RESOURCES:

e) Python 3.7.0
f) Install: pip installer, pandas, SciPy, NumPy, Sklearn, Seaborn library

PROCEDURE:

1. Create: Open a new file in Python shell, write a program and save the program with .py extension.
2. Execute: Go to Run -> Run module (F5)

PROGRAM LOGIC:

a) Explore data and visualize each attribute

```
import pandas as pd
import numpy as np
import pandas as pd
import numpy as np
import seaborn as sns
from pandas.plotting import scatter_matrix
from sklearn.linear_model import LogisticRegression
#Reading dataset
bank=pd.read_csv("D:/datasets/bank-additional-full.csv", index_col=0)
 # index_col will remove the index column from the csv file

# Assign outcome as 0 if income <=50K and as 1 if income >50K
bank['y'] = [0 if x == 'no' else 1 for x in bank['y']]

# Assign X as a DataFrame of features from bank dataset and y as a Series of the outcome variable
# axis : {0 or 'index', 1 or 'columns'}, default 0
# Whether to drop labels from the index (0 or 'index') or columns (1 or 'columns').
X = bank.drop('y', 1) # 1 represents column, dropping y column  for doing classification
y = bank.y
X.describe()
```

	duration	campaign	pdays	previous	emp.var.rate	cons.price.idx	cons.conf.idx	euribor3m	nr.employed	job_admin.	...	month_oct	month_sep	day_of_week fri	day_of_week mon	day_of_week thu	day_of_week tue	day_of_week wed	poutcome_failure	poutcome_nonexistent	poutcome_success
count	41188	41188	41188	41188	41188	41188	41188	41188	41188	41188	41188	...	41188	41188	41188	41188	41188	41188	41188	41188	41188

	mean	std	min	25%	50%	75%	max
	258.28501	259.279249	0	102	180	319	4918
	2.567593	2.770014	1	1	2	3	56
	962.475545	186.91091	0	999	999	999	999
	0.172963	0.494901	0	0	0	0	7
	0.081886	1.570096	-3.4	-1.8	1.1	1.4	1.4
	93.575664	0.578841	92.201	93.075	93.749	93.994	94.767
	40.502	4.628198	-50.8	-42.7	-41.8	-36.4	-26.9
	3.621291	1.734447	0.634	1.344	4.857	4.961	5.045
	5167.0359	72.251528	4963.6	5099.1	5191	5228.1	5228.1
	0.253035	0.434756	0	0	0	1	1
	...	...	...	...	...	...	...
	0.017432	0.130877	0	0	0	0	1
	0.013839	0.116824	0	0	0	0	1
	0.190031	0.392033	0	0	0	0	1
	0.206711	0.404951	0	0	0	0	1
	0.209357	0.406855	0	0	0	0	1
	0.196416	0.397292	0	0	0	0	1
	0.197485	0.398106	0	0	0	0	1
	0.103234	0.304268	0	0	0	0	1
	0.863431	0.343396	0	1	1	1	1

y.describe()
count 41188.0
mean 1.0
std 0.0
min 1.0
25% 1.0
50% 1.0
75% 1.0
max 1.0
Name: y, dtype: float64

X.head()

age	job	marital	education	default	housing	loan	contact	month	day_of_week	duration	campaign	pdays	previous	poutcome	emp.var.rate	cons.price.idx	cons.conf.idx	euribor3m	nr.employed	y
56	housemaid	married	basic.4y	no	no	no	telephone	may	mon	261	1	999	0	nonexistent	1.1	93.994	-36.4	4.857	5191.0	no
57	services	married	high.school	unknown	no	no	telephone	may	mon	149	1	999	0	nonexistent	1.1	93.994	-36.4	4.857	5191.0	no
37	services	married	high.school	no	yes	no	telephone	may	mon	226	1	999	0	nonexistent	1.1	93.994	-36.4	4.857	5191.0	no
40	admin.	married	basic.6y	no	no	no	telephone	may	mon	151	1	999	0	nonexistent	1.1	93.994	-36.4	4.857	5191.0	no
56	services	married	high.school	no	no	yes	teleph													

```
y.head()
age
56   0
57   0
37   0
40   0
56   0
Name: y, dtype: int64

#Count of unique values(y/n)

bank['y'].value_counts()
```

OUTPUT:

```
# 4640 people opened term deposit account and 36548 have not opened the term deposit account

0   36548
1    4640
Name: y, dtype: int64

# Decide which categorical variables you want to use in model

for col_name in X.columns:
   if X[col_name].dtypes == 'object':# in pandas it is object
      unique_cat = len(X[col_name].unique())
      print("Feature '{col_name}' has {unique_cat} unique categories".format(col_name=col_name,
unique_cat=unique_cat))
      print(X[col_name].value_counts())
      print()
```

OUTPUT:

```
Feature 'job' has 12 unique categories
admin.          10422
blue-collar      9254
technician       6743
services         3969
management       2924
retired          1720
entrepreneur     1456
self-employed    1421
housemaid        1060
unemployed       1014
student           875
unknown           330
Name: job, dtype: int64

Feature 'marital' has 4 unique categories
married    24928
single     11568
divorced    4612
unknown       80
Name: marital, dtype: int64

Feature 'education' has 8 unique categories
university.degree     12168
high.school            9515
```

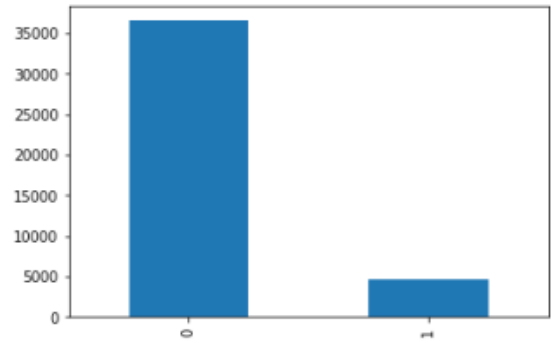
```
basic.9y              6045
professional.course    5243
basic.4y              4176
basic.6y              2292
unknown                1731
illiterate              18
Name: education, dtype: int64

Feature 'default' has 3 unique categories
no         32588
unknown     8597
yes            3
Name: default, dtype: int64

Feature 'housing' has 3 unique categories
yes        21576
no         18622
unknown      990
Name: housing, dtype: int64

Feature 'loan' has 3 unique categories
no         33950
yes         6248
unknown      990
Name: loan, dtype: int64

Feature 'contact' has 2 unique categories
cellular    26144
telephone   15044
Name: contact, dtype: int64
```

```
Feature 'month' has 10 unique categories
may    13769
jul     7174
aug     6178
jun     5318
nov     4101
apr     2632
oct      718
sep      570
mar      546
dec      182

Name: month, dtype: int64

Feature 'day_of_week' has 5 unique categories
thu    8623
mon    8514
wed    8134
tue    8090
fri    7827
Name: day_of_week, dtype: int64

Feature 'poutcome' has 3 unique categories
nonexistent    35563
failure         4252
success         1373
Name: poutcome, dtype: int64
```

Visualizations

#visualization of Predictor variable (y)

```
print(y.value_counts().plot.bar())
```



b) Predict the test set results and find the accuracy of the model

#Create an Logistic classifier and train it on 70% of the data set.

```
clf = LogisticRegression()

clf
LogisticRegression(C=1.0, class_weight=None, dual=False, fit_intercept=True,
                   intercept_scaling=1, l1_ratio=None, max_iter=100,
                   multi_class='warn', n_jobs=None, penalty='l2',
                   random_state=None, solver='warn', tol=0.0001, verbose=0,
                   warm_start=False)


clf.fit(X, y)
LogisticRegression(C=1.0, class_weight=None, dual=False, fit_intercept=True,
                    intercept_scaling=1, l1_ratio=None, max_iter=100,
                    multi_class='warn', n_jobs=None, penalty='l2',
                    random_state=None, solver='warn', tol=0.0001, verbose=0,
                    warm_start=False)
```

c) Visualize the confusion matrix

```
from sklearn.metrics import confusion_matrix
confusion_matrix = confusion_matrix(y_test, y_pred)
print(confusion_matrix)
```

d) Compute precision, recall, F-measure and support

```
from sklearn.metrics import classification_report
print(classification_report(y_test, y_pred))
```

Program No. 9

WAP in MATLAB to demonstrate the making of Neural Networks and observing its performance analysis.

Source Code:

```
% Feed-Forward Neural Network (FNN)
[x, t] = simplefit_dataset;
net = feedforwardnet (10);
net = train (net, x,t);
view(net)
y = net(x);
perf = perform (net, y, t)
```

Outputs:

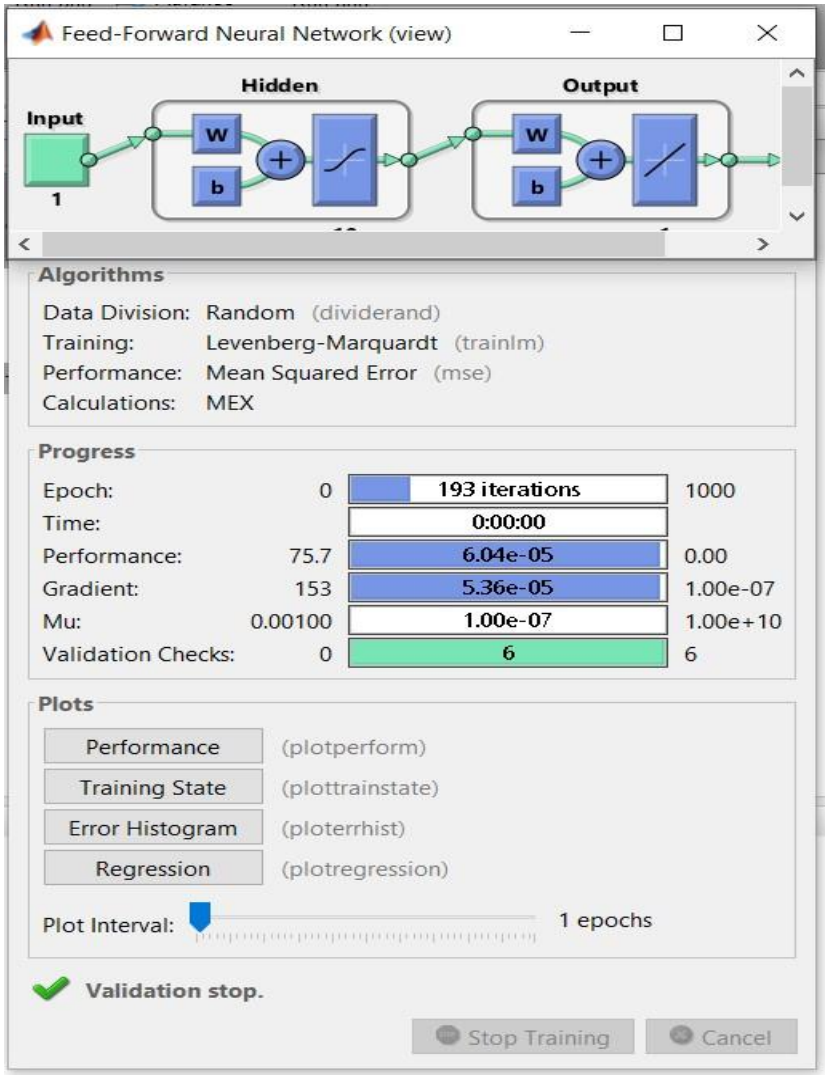


Fig. 1 Feed Forward Neural Network

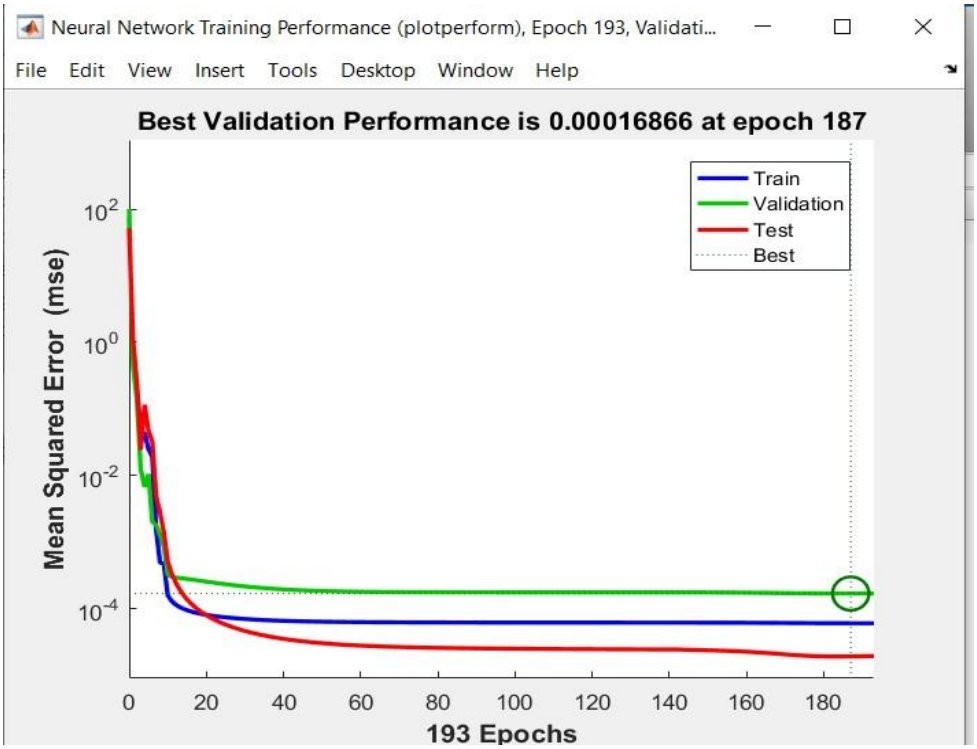


Fig. 2 Training Performance

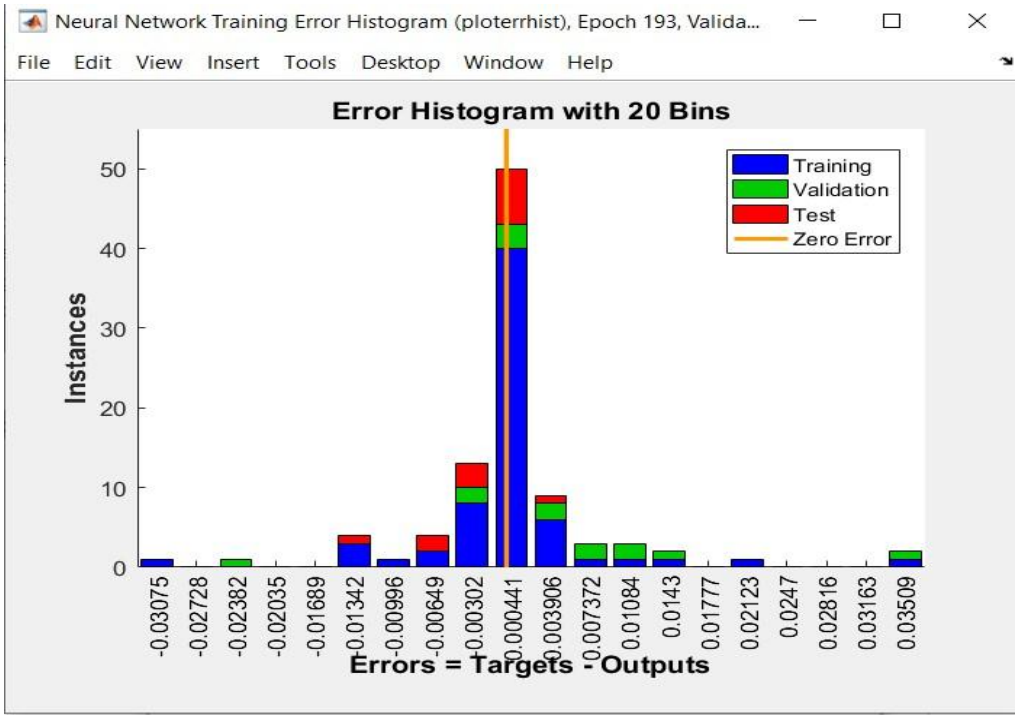


Fig. 3 Training Error Histogram

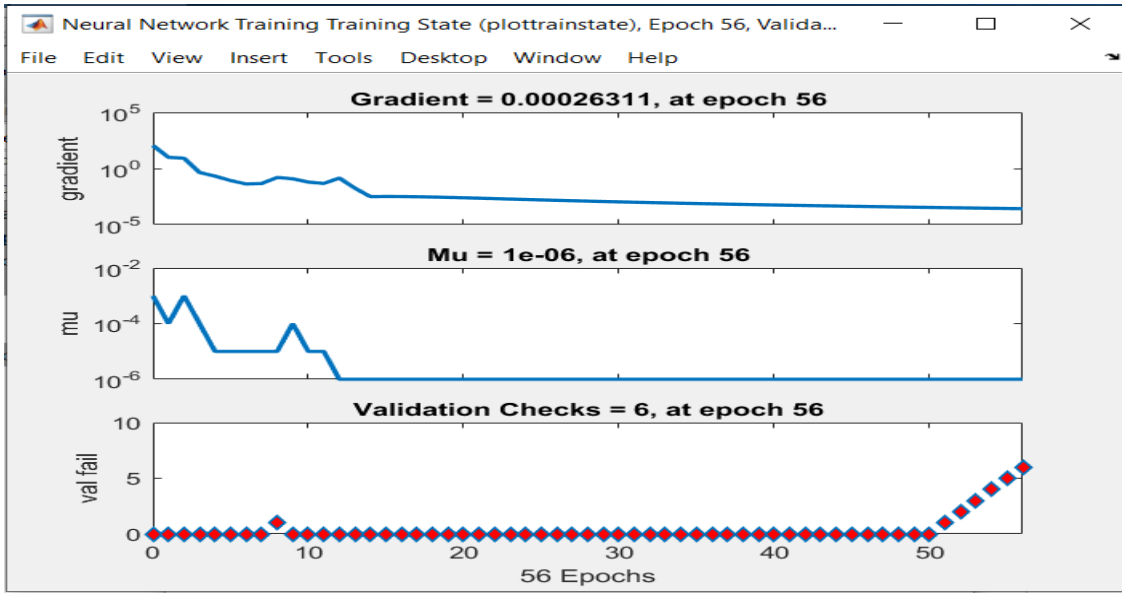


Fig. 4 Training State

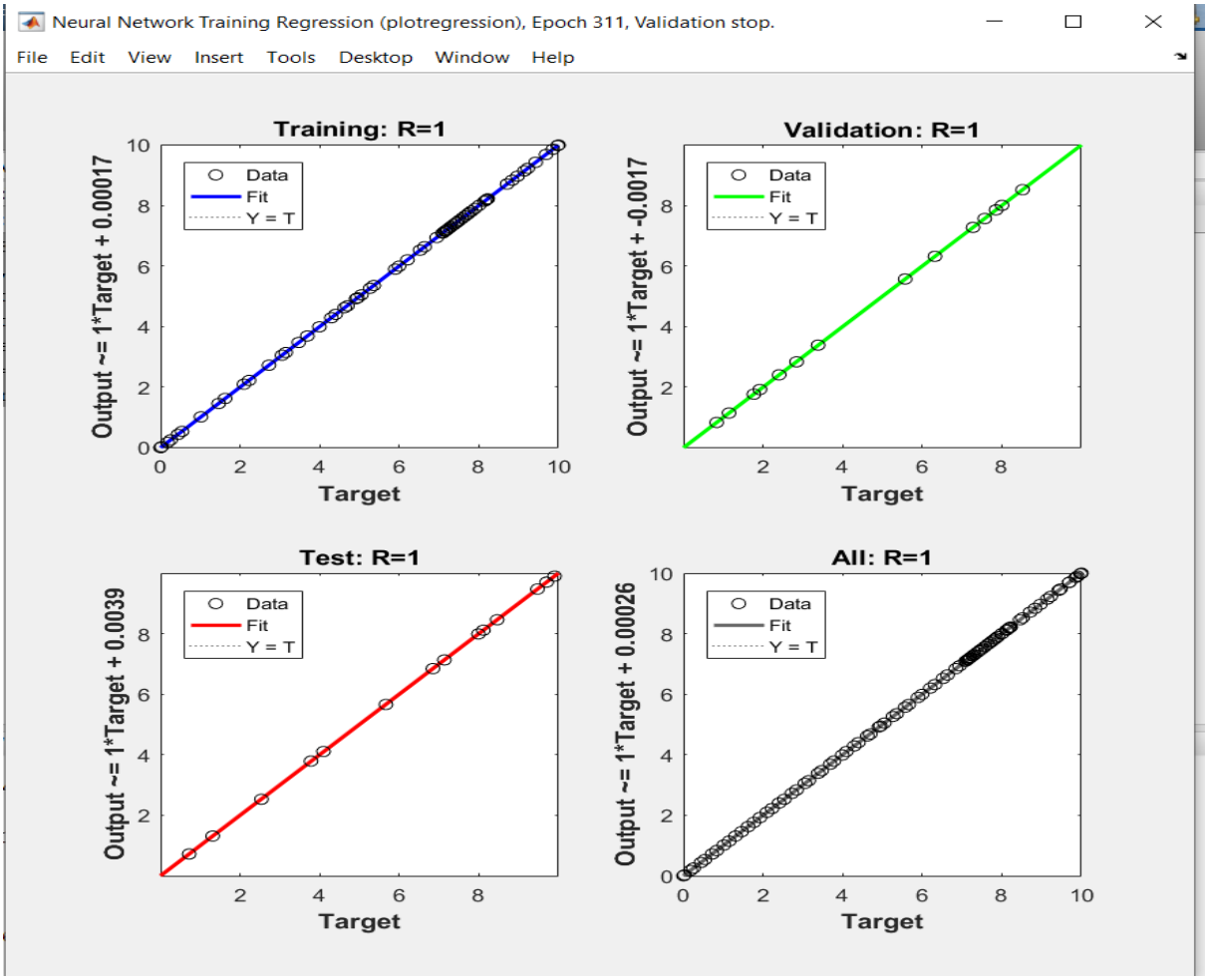


Fig. 5 Regression Plot

Program No. 10

WAP in MATLAB to demonstrate Support Vector Machine (SVM) Classification with pre-defined dataset

Source Code:

```
% Support Vector Machine (SVM)

load fisheriris
xdata = meas(51:end,3:4);
group = species (51: end);
svmStruct = svmtrain (xdata, group, 'ShowPlot', true);
```

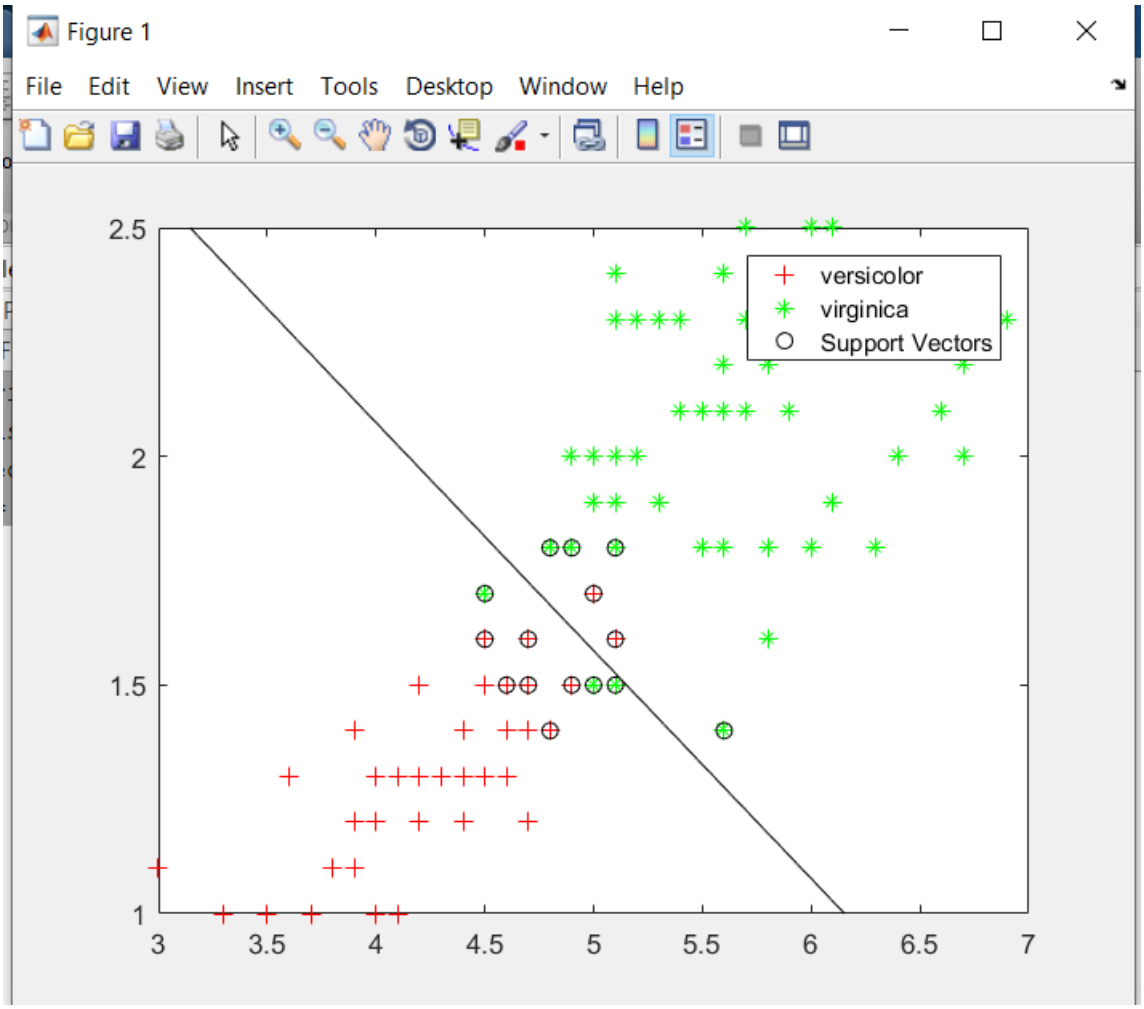


Fig. 1 SVM Classification

VIVA QUESTIONS

Q.1. Name types of data mining?

- Data cleaning
- Integration
- Selection
- Data transformation
- Data mining
- Pattern evaluation
- Knowledge representation

Q.2. Name the steps used in data mining?

- Business understanding
- Data understanding
- Data preparation
- Modeling
- Evaluation
- Deployment

Q.3. Name areas of applications of data mining?

- Data Mining Applications for Finance
- Healthcare
- Intelligence
- Telecommunication
- Energy
- Retail
- E-commerce
- Supermarkets
- Crime Agencies
- Businesses Benefit from data mining

Q.4. What is required technological drivers in data mining?

Database size: Basically, as for maintaining and processing the huge amount of data, we need powerful systems.

Query Complexity: Generally, to analyze the complex and large number of queries, we need a more powerful system.

Q.5. Explain steps involved in data mining knowledge process?

Data Cleaning − Basically, in this step, the noise and inconsistent data are removed.
Data Integration − Moreover, in this step, multiple data sources are combined.
Data Selection − Furthermore, in this step, data relevant to the analysis task are retrieved from the database.
Data Transformation − Basically, in this step, data is transformed into forms appropriate for mining. Also, by performing summary or aggregation operations.
Data Mining − In this, intelligent methods are applied in order to extract data patterns.
Pattern Evaluation − While, in this step, data patterns are evaluated.
Knowledge Presentation − Generally, in this step, knowledge is represented

Q.6. What are issues in data mining?

A number of issues that need to be addressed by any serious data mining package

- Uncertainty Handling
- Dealing with Missing Values
- Dealing with Noisy data
- Efficiency of algorithms
- Constraining Knowledge Discovered to only Useful
- Incorporating Domain Knowledge
- Size and Complexity of Data
- Data Selection
- Understandably of Discovered Knowledge: Consistency between Data and Discovered Knowledge.

Q.7. Name different level of analysis of data mining?

- Artificial Neural Networks
- Genetic algorithms
- Nearest neighbor method
- Rule induction
- e Data visualization

Q.8. Name methods of classification methods?

- Statistical Procedure Based Approach
- b Machine Learning Based Approach
- Neural Network
- Classification Algorithms
- ID3 Algorithm
- C4.5 Algorithm
- K Nearest Neighbors Algorithm
- H. Naïve Bayes Algorithm
- SVM Algorithm
- J. ANN Algorithm
- K. 48 Decision Trees
- l. Support Vector Machines

Q.9. Explain Statistical Procedure Based Approach?

Especially, there are two main phases present to work on classification. Also, it can be easily identified within the statistical community.

While, the second, “modern” phase concentrated on more flexible classes of models. Also, in which many of which attempt has to take. Moreover, it provides an estimate of the joint distribution of the feature within each class. Further, that can, in turn, provide a classification rule.

Generally, statistical procedures have to characterize by having a precise fundamental probability model and that is used to provides a probability of being in each class instead of just a classification.

Also, we can assume that the techniques will use by statisticians. Hence some human involvement has to assume with regard to variable selection.

Also, transformation and overall structuring of the problem.

Q10.Name methods of clustering?
They are classified into the following categories −

- Partitioning Method
- Hierarchical Method
- Density-based Method
- Grid-Based Method
- Model-Based Method
- Constraint-based Method

Q11.What do you mean by Perceptron?

- A perceptron also called an artificial neuron is a neural network unit that does certain computations to detect features.
- It is a single-layer neural network used as a linear classifier while working with a set of input data. Since perceptron uses classified data points which are already labeled, it is a supervised learning algorithm. This algorithm is used to enable neurons to learn and process elements in the training set one at a time.

Q12.Explain Biological Neural Network and Artificial Neural network?
Biological Neural Network is prepared with the help of real neurons. Our nervous system is made via neurons and brain. A biological neural network is our nervous system in which neurons are linked with each other.
Human has a mind to understand and can operate any task in a particular situation, but how can a robot do that? For that purpose, an artificial brain is designed, which is known as **Neural Network**. The human brain has neurons to transfer the information or data. Similarly, the neural network has nodes to perform the task. Nodes are the mathematical functions.

Q13. What do you mean by combination function in Neural Network?
Combination function is the vector to scalar function where each non-input combines the value that is fed into neural network via synaptic connections from other units producing the net input. Most NN uses the following two types of combination function:
1. Linear Combination Function (as in MLPs)
2. Euclidean Distance Combination Function (as in RBF)

Q14. How can we count the layers in a neural network?
There are two ways of counting the layers.
1. We can count the layers in units.
2. Layers can be counted in weight also.

Q15. Explain cases and variables?
The vector value, which is present at one time for all input units in the neural network, is known as a case, e.g., pattern, sample, etc. A case includes target value and other possible information with input values. On the other hand, the vector value that appears different times for a single input is known as input variable or features.