|  |  |  |
| --- | --- | --- |
| Picture1 | ***DEPARTMENT OF COMPUTER ENGINEERING EXPERIMENT NO. 2*** | |
| Semester | | TE Semester VI – Computer Engineering | |
| Subject | | System Programming And Compiler Construction | |
| Professor In-charge | | Prof. Avinash Shrivas | |
| Laboratory In-charge | | Prof. Suja Jayachandran | |
| Laboratory | | M 310 | |

|  |  |  |
| --- | --- | --- |
| Student Name | Chinmay Terse | |
| Roll Number | 13106B0004 | |
| Grade and Professor’s Signature |  |  |

|  |  |  |
| --- | --- | --- |
| Experiment No. | 2 | |
| Experiment Title | To implement multiple regression. | |
| Resources/Apparatus Required | Hardware:  IBM PC compatible computer system | Software:  Eclipse, Java |
| Objectives | * To work with a dataset containing two predictor variables and one response variable. * To learn about multiple regression and to use it to define a line of regression and solve the given query. * To use Java and its various libraries as tools to implement multiple regression. | |
| Theory of Operation | The general purpose of multiple regression (the term was first used by Pearson, 1908) is to learn more about the relationship between several independent or predictor variables and a dependent or criterion variable. For example, a real estate agent might record for each listing the size of the house (in square feet), the number of bedrooms, the average income in the respective neighbourhood according to census data, and a subjective rating of appeal of the house. Once this information has been compiled for various houses it would be interesting to see whether and how these measures relate to the price for which a house is sold. Personnel professionals customarily use multiple regression procedures to determine equitable compensation. The general computational problem that needs to be solved in multiple regression analysis is to fit a straight line to a number of points. At the center of the multiple linear regression analysis is the task of fitting a single line through a scatter plot.  More specifically the multiple linear regression fits a line through a multi-dimensional cloud of data points.  The simplest form has one dependent and two independent variables, the general form of the multiple linear regression is defined as  http://www.statisticssolutions.com/wp-content/uploads/2010/01/mlr38.jpg  for i = 1…n .  There are 3 major uses for Multiple Linear Regression Analysis – (1) causal analysis, (2) forecasting an effect, (3) trend forecasting.  Other than correlation analysis, which focuses on the strength of the relationship between two or more variables, regression analysis assumes a dependence or causal relationship between one or more independent and one dependent variable. | |
| Implementation | import java.text.DecimalFormat;  import java.util.ArrayList;  import java.util.List;  import java.util.Scanner;  import Jama.Matrix;  public class MultipleRegression {  public static void main(String[] args) {  // TODO Auto-generated method stub  Scanner t = new Scanner(System.in);  System.out.print("Enter size of dataset: ");  int size = t.nextInt();  System.out.println("Enter the data in the [IA1\_marks IA2\_marks ESE\_marks] format:");  List<Integer> predictor1 = new ArrayList<Integer>();  List<Integer> predictor2 = new ArrayList<Integer>();  List<Integer> response = new ArrayList<Integer>();  for(int i=0; i<size; i++) {  predictor1.add(t.nextInt());  predictor2.add(t.nextInt());  response.add(t.nextInt());  }  int sumPred1 = 0, sumPred2 = 0, sumResponse = 0, sumPred1Square = 0, sumPred2Square = 0;  int sumPred1ResponseProd = 0, sumPred2ResponseProd = 0, sumPred1Pred2Prod = 0;  sumPred1 = sumList(predictor1);  sumPred2 = sumList(predictor2);  sumResponse = sumList(response);  sumPred1Square = sumListSquare(predictor1);  sumPred2Square = sumListSquare(predictor2);  sumPred1ResponseProd = sumProductOfLists(predictor1, response);  sumPred2ResponseProd = sumProductOfLists(predictor2, response);  sumPred1Pred2Prod = sumProductOfLists(predictor1, predictor2);  System.out.println("\nThe three equations are: ");  System.out.println(sumResponse+" = "+size+"a + "+sumPred1+"b + "+sumPred2+"c");  System.out.println(sumPred1ResponseProd+" = "+sumPred1+"a + "+sumPred1Square+"b + "+sumPred1Pred2Prod+"c");  System.out.println(sumPred2ResponseProd+" = "+sumPred2+"a + "+sumPred1Pred2Prod+"b + "+sumPred2Square+"c");  double[][] rhsArray = {{size,sumPred1,sumPred2},  {sumPred1, sumPred1Square, sumPred1Pred2Prod},  {sumPred2, sumPred1Pred2Prod, sumPred2Square} };  double[] lhsArray = {sumResponse, sumPred1ResponseProd, sumPred2ResponseProd};  Matrix rhs = new Matrix(rhsArray);  Matrix lhs = new Matrix(lhsArray, 3);  Matrix answer = rhs.solve(lhs);  System.out.println("\na = "+answer.get(0, 0));  System.out.println("b = "+answer.get(1, 0));  System.out.println("c = "+answer.get(2, 0));  System.out.print("\nThe equation of the line of regression becomes: ");  System.out.println("y = "+new DecimalFormat(".###").format(answer.get(0, 0))+" + "+new DecimalFormat(".###").format(answer.get(1, 0))+" \* x1 + "+new DecimalFormat(".###").format(answer.get(2, 0))+" \* x2");  System.out.print("\nEnter the marks in IA1 and IA2: ");  int marks1 = t.nextInt();  int marks2 = t.nextInt();  double eseMarks = answer.get(0, 0) + answer.get(1, 0)\*marks1 + answer.get(2, 0)\*marks2;  System.out.print("The marks predicted in ESE are: ");  System.out.println(new DecimalFormat(".###").format(eseMarks));  }  private static int sumProductOfLists(List<Integer> list1, List<Integer> list2) {  // TODO Auto-generated method stub  int sum = 0;  for(int i=0; i<list1.size(); i++) {  sum += list1.get(i) \* list2.get(i);  }  return sum;  }  private static int sumListSquare(List<Integer> list) {  // TODO Auto-generated method stub  int sum = 0;  for(int i=0; i<list.size(); i++) {  sum += list.get(i) \* list.get(i);  }  return sum;  }  private static int sumList(List<Integer> list) {  // TODO Auto-generated method stub  int sum = 0;  for(int i=0; i<list.size(); i++) {  sum += list.get(i);  }  return sum;  }  } | |
| Output |  | |
| Conclusion | Hence, we learnt that multiple regression is used to obtain a line of regression when more than one predictor variables affect the value of the response variable. We have successfully written a Java program to implement multiple regression and solved a example which predicts the marks obtained by a student in the end semester examination based upon his marks in the internal assessment tests. The equation of the line of regression is also given and it can be seen that the obtained result to the provided query is correct. | |