

Assignment No: C2

Title : To study different business and analytics tools.

Aim: Write Select an Industrial sector and write a BIA tool for maximizing the profit

Objectives:

- To study the Business intelligence.
- Select an Industrial sector and write a BIA tool for maximizing the profit.

Theory**BIA:**

Business intelligence (BI) can be described as "a set of techniques and tools for the acquisition and transformation of raw data into meaningful and useful information for business analysis purposes". The term "data surfacing" is also more often associated with BI functionality. BI technologies are capable of handling large amounts of unstructured data to help identify, develop and otherwise create new strategic business opportunities. The goal of BI is to allow for the easy interpretation of these large volumes of data. Identifying new opportunities and implementing an effective strategy based on insights can provide businesses with a competitive market advantage and long-term stability.

BI technologies provide historical, current and predictive views of business operations. Common functions of business intelligence technologies are reporting, online analytical processing, analytics, data mining, process mining, complex event processing, business performance management, benchmarking, text mining, predictive analytics and prescriptive analytics.

BI can be used to support a wide range of business decisions ranging from operational to strategic. Basic operating decisions include product positioning or pricing. Strategic business decisions include priorities, goals and directions at the broadest level. In all cases, BI is most effective when it combines data derived from the market in which a company operates (external data) with data from company sources internal to the business such as financial and operations data (internal data). When combined, external and internal data can provide a more complete picture which, in effect, creates an "intelligence" that cannot be derived by any singular set of data.

Business intelligence is made up of an increasing number of components including: Multidimensional aggregation and allocation Denormalization, tagging and standardization Real-time reporting with analytical alert A method of interfacing with unstructured data sources Group consolidation, budgeting and rolling forecasts Statistical inference and probabilistic simulation Key performance indicators optimization Version control and process management Open item management.

BIA tool

Business intelligence tools are a type of application software designed to retrieve, analyze, transform and report data for business intelligence. The tools generally read data that have been previously stored, often, though not necessarily, in a data warehouse or data mart.

Types of business intelligence tools:

The key general categories of business intelligence tools are:

- Spreadsheets
- Reporting and querying software: tools that extract, sort, summarize, and present selected data
- OLAP: Online analytical processing
- Digital dashboards
- Data mining
- Process Visualization
- Data warehousing
- Local information systems
- Except for spreadsheets, these tools are provided as standalone tools, suites of tools, components of ERP systems, or as
- Components of software targeted to a specific industry. The tools are sometimes packaged into data warehouse appliances.

Simple cumulative value calculator:

In this approach, the scores of each individual parameter is added, and a cumulative value is obtained. For the maximum salary, the corresponding parameters are displayed. When parameters are given as input, the cumulative value of those parameters is compared with the available dataset. If a nearby match is found, the corresponding company name and salary is displayed.

KNN:

The K Nearest Neighbor (k-NN) is a very intuitive method that classifies unlabeled examples based on their similarity with examples in the training set. k-NN is a type of instance-based learning, or lazy learning, where the function is only approximated locally and all computation is deferred until classification. The k-NN algorithm is among the simplest of all machine learning algorithms.

k-NN advantages

1. The cost of the learning process is zero
2. No assumptions about the characteristics of the concepts to learn have to be done
3. Complex concepts can be learned by local approximation using simple procedures
4. Robust to noisy training data

MATHEMATICAL MODEL

Let S be the solution

$S = \{s, e, i, o, f, DD, NDD, \text{success}, \text{failure}\}$

s = initial state that is constructor of the class.

e = be the end state or destructor of the class.

i = input of the system.

o = output of the system.

DD-deterministic data it helps identifying the load store functions or assignment functions.

NDD- Non deterministic data of the system S to be solved.

Success-desired outcome generated.

Failure-Desired outcome not generated or forced exit due to system error.

Success= {Desired outcome generated}

Failure= {Desired outcome not generated or forced exit due to system error}

Algorithm

1. Start
2. Determine parameter K = number of nearest neighbours
3. Calculate the distance between the query instance and all the training samples
4. Sort the distance and determine nearest neighbours based on the K-th minimum distance
5. Gather the category Y of the nearest neighbours
6. Use simple majority of the category of nearest neighbours as the prediction value of the query instance

Conclusion:

Thus we have implemented Knn in java.

Code:

```

import java.io.*;
import java.util.*;
class Knn
{
    ArrayList<String> transactions = new ArrayList<String>();
    ArrayList<Double> transCount = new ArrayList<Double>();
    String testData;
    int count,k;
    void print()
    {
        for(int i=0;i<k;i++)
        {
            int index=transCount.indexOf(Collections.min(transCount));
            System.out.println(transactions.get(i));
            transactions.remove(index);
            transCount.remove(index);
        }
    }
    void operate()
    {
        String tData[]=testData.split(" ");
        for(String input:transactions)
        {
            double cost=0;
            String train[]=input.split(" ");
            for(int i=0;i<tData.length;i++)
            {
                try
                {
                    cost+=Math.abs(Integer.parseInt(train[i])-Integer.parseInt(tData[i]));
                }
                catch(NumberFormatException e)
                {
                    if(train[i].equals(tData[i]))
                        cost+=1;
                }
            }
            cost/=tData.length;
            transCount.add(cost);
        }
    }
    void input()throws Exception
    {
        BufferedReader br = new BufferedReader(new InputStreamReader(System.in));
        //Input of training data can also be taken from file in case of large input
        /*
        BufferefReader fr = new Bufferedreader(new FileReader(filename));
        String input;
        while((input=fr.readLine())!=null){
            transactions.add(input);
            count++;
        }
        And replace lines below, directly take testing data.
        */
        System.out.println("Enter No of transactions");
        count=Integer.parseInt(br.readLine());
        System.out.println("Enter the training data in (FirstYearMarks-SecondYearMarks-ThirdYearMarks-CompanyAssigned)");
        for(int i=0;i<count;i++)
        {
            transactions.add(br.readLine());
        }
        System.out.println("Enter the testing data in (FirstYearMarks-SecondYearMarks-ThirdYearMarks-enter)");
    }
}

```

```
        testData=br.readLine();
        System.out.println("Enter the value of k");
        k=Integer.parseInt(br.readLine());
    }
    public static void main(String []args)throws Exception
    {
        Knn knn=new Knn();
        knn.input();
        knn.operate();
        knn.print();
    }
}

/*
Enter No of transactions
5
Enter the training data in (FirstYearMarks-SecondYearMarks-ThirdYearMarks-CompanyAssigned)
890 980 850 Accenture
1040 1030 990 Accenture
1033 1034 1017 TechMahindra
890 870 1015 Infosys
950 1000 990 Cognizant
Enter the testing data in (FirstYearMarks-SecondYearMarks-ThirdYearMarks-enter)
980 990 1080
Enter the value of k
3
890 980 850 Accenture
1040 1030 990 Accenture
890 870 1015 Infosys
*/
```

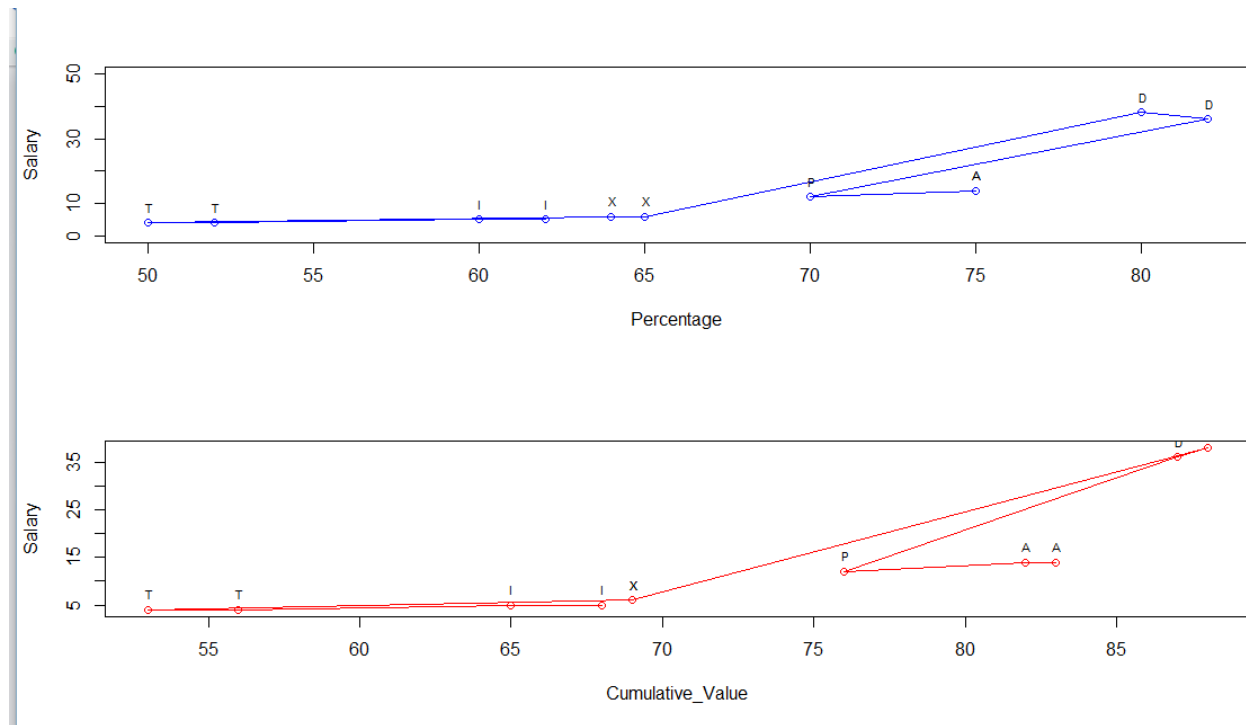
```

cat ("Reading placement data from records .csv file . Data is : " )
cat ("\n")
heads<-read.csv("record.csv")
heads
Percentage=heads$Percentage
Projects=heads$Projects
Internships=heads$Internships
Papers=heads$Papers
Company=heads$Company
Salary=heads$Salary
#curve ( x^2+x , from=0, to=100)
calculate <-function(Percentage , Projects , Internships , Papers ){
  return ( Percentage+Projects+Internships+Papers )
}
Cumulative_Value=calculate ( Percentage , Projects , Internships , Papers )
par (mfrow = c ( 2 , 1 ) )
plot ( Percentage , Salary , type="o" , col="blue " , ylim = c ( 0 , 50 ) )
text ( Percentage , Salary , labels=Company , cex= 0.7 , pos=3)
plot ( Cumulative_Value , Salary , type="o" , col="red ")
text ( Cumulative_Value , Salary , labels=Company , cex= 0.7 , pos=3)
max_value=max( Salary )
index=match(max_value , Salary)
cat ("The maximum profit ( highest salary ) is in the case of : ")
cat ("\n")
cat ( paste ( " Percentage : " , Percentage [ index ] ) )
cat ("\n")
cat ( paste ( " Projects : " , Projects [ index ] ) )
cat ("\n")
cat ( paste ( " Internships : " , Internships [ index ] ) )
cat ("\n")
cat ( paste ( " Papers : " , Papers [ index ] ) )
cat ("\n")
cat ( paste ( "With the company being : " ,Company [ index ] ) )
cat ("\n")
cat ( paste ( " Cumulative Value being : " , Cumulative_Value [ index ] ) )
cat ("\n")
cat ("*** Refer Rplots.pdf for overview ***")
cat ("\n")
cat (" Enter data to check possible placement company and salary : " )
cat ("\n")
cat (" Enter percentage : ")
percent1 <-80
cat (" Enter number of projects : ")
projects1 <-6
cat (" Enter number of internships : ")
internships1 <-1
cat (" Enter number of papers : ")
papers1 <-2

check_value=percent1+projects1+internships1+papers1
temp<-c( ( check_value-1) : (check_value+1))
storeindex=-1
for(i in 1 : length ( temp ) ) {
  for( j in 1 : length ( Cumulative_Value ) ) {
    if ( temp [ i ]==Cumulative_Value [ j ] ) {
      storeindex=j
      break
    }
  }
}

if(storeindex!=-1){
  print(paste ("Recommended company is : " ,Company [ storeindex] , " with salary : " , Salary))
}
else{
  print("Canot determine possible placement scenerio ")
}

```



Record.csv

	Percentage	Projects	Internships	Papers	Company	Salary
1	75	5	2	1	A	14
2	75	4	0	3	A	14
3	70	4	1	1	P	12
4	82	5	0	0	D	36
5	80	6	1	1	D	38
6	65	4	0	0	X	6
7	64	4	1	0	X	6
8	50	3	0	0	T	4
9	52	3	1	0	T	4
10	60	4	1	0	I	5
11	62	5	0	1	I	5