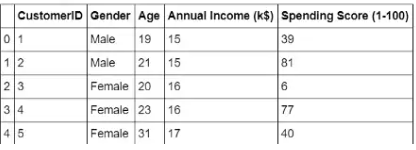
1.Consider that you are owning a supermarket mall and through membership cards, you have some basic data about your customers like Customer ID, age, gender, annual income and spending score. For the above scenario, the Problem Statement was You want to understand the customers who can easily converge [Target Customers] so that the data can be given to the marketing team and plan the strategy accordingly. For the above scenario prepare a dataset and perform **Clustering Analysis** to segment the customers in the Mall. There are clearly Five segments of Customers based on their Annual Income and Spending Score namely *Usual Customers, Priority Customers, Senior Citizen Target Customers, and Young Target Customers.*Sample data



2.Create the following dataset using CSV file format. To perform cluster analysis using K- Means in WEKA. To change the cluster size and plot the graph and illustrate the visualization of cluster.

| EmployeID | Gender | Age | Salary | Credit |
| --- | --- | --- | --- | --- |
| 111 | Male | 28 | 150000 | 39 |
| 222 | Male | 25 | 150000 | 27 |
| 333 | Female | 26 | 160000 | 42 |
| 444 | Female | 25 | 160000 | 40 |
| 555 | Female | 30 | 170000 | 64 |
| 666 | Male | 29 | 200000 | 72 |

3.Prediction of categorical data using Naïve Bayes classification through WEKA using any datasets. Compare the Naïve Bayes algorithm with SVM using the summary of results given by the classifiers and plot the graph.

**4.T**he following list of persons with vegetarian or not details given in the table. How will you find out how many of them are vegetarian and how many of them are non-vegetarian? Which type of the person total count is greater value?

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Person** | Gopu | Babu | Baby | Gopal | Krishna | Jai | Dev | Malini | Hema | Anu |
| **Vegetarian** | yes | yes | yes | no | yes | no | no | yes | yes | yes |

**num\_vegetarian <- sum(vegetarian\_status == "yes")**

**num\_non\_vegetarian <- sum(vegetarian\_status == "no")**

**print(paste("Number of Vegetarian Individuals:", num\_vegetarian))**

**print(paste("Number of Non-Vegetarian Individuals:", num\_non\_vegetarian))**

**if (num\_vegetarian > num\_non\_vegetarian) {**

**print("Vegetarian individuals have a greater count.")**

**} else if (num\_non\_vegetarian > num\_vegetarian) {**

**print("Non-vegetarian individuals have a greater count.")**

**} else {**

**print("The counts of vegetarian and non-vegetarian individuals are equal.")**

**}**

5.The following table would be plotted as (x,y) points, with the first column being the x values as number of mobile phones sold and the second column being the y values as money. To use the scatter plot for how many mobile phones sold.

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| x | 4 | 1 | 5 | 7 | 10 | 2 | 50 | 25 | 90 | 36 |
| y | 12 | 5 | 13 | 19 | 31 | 7 | 153 | 72 | 275 | 110 |

**x\_values <- c(4, 1, 5, 7, 10, 2, 50, 25, 90, 36)**

**y\_values <- c(12, 5, 13, 19, 31, 7, 153, 72, 275, 110)**

**install.packages("ggplot2") # Install ggplot2 package**

**library(ggplot2) # Load ggplot2 package**

**# Create a data frame**

**data <- data.frame(x = x\_values, y = y\_values)**

**# Create a scatter plot**

**scatter\_plot <- ggplot(data, aes(x = x, y = y)) +**

**geom\_point() + # Add points to the plot**

**labs(title = "Scatter Plot of Mobile Phones Sold", x = "Number of Mobile Phones Sold", y = "Money") # Set title and axis labels**

**# Display the scatter plot**

**print(scatter\_plot)**

6.Generate rules using FP growth algorithm using the given dataset which has the following transactions with items purchased: Consider the values as support=50% and confidence=75%.



**7.P**rediction of Diabetes Data using Decision tree classifier in WEKA. Compare it with Support Vector Machine classifier. Show the result accuracy and F1 measure calculation .Plot the graph and explain the summary of results.

8.Implement of the R script using marks scored by a student in his model exam has been sorted as follows: 55, 60, 71, 63, 55, 65, 50, 55,58,59,61,63,65,67,71,72,75. Partition them into three bins by each of the following methods. Plot the data points using histogram.

(a) equal-frequency (equi-depth) partitioning

(b) equal-width partitioning

(c) clustering

**# Sample data**

**marks <- c(55, 60, 71, 63, 55, 65, 50, 55, 58, 59, 61, 63, 65, 67, 71, 72, 75)**

**# Equal-frequency (equi-depth) partitioning**

**bins\_eq\_freq <- cut(marks, breaks = 3, labels = c("Low", "Medium", "High"),**

**include.lowest = TRUE)**

**# Equal-width partitioning**

**bins\_eq\_width <- cut(marks, breaks = c(50, 60, 70, 80), labels = c("Low", "Medium",**

**"High"), include.lowest = TRUE)**

**# Clustering**

**library(cluster)**

**kmeans\_result <- kmeans(matrix(marks), centers = 3)**

**bins\_clustering <- cutree(kmeans\_result, k = 3)**

**# Plot histograms**

**par(mfrow = c(1, 3))**

**hist(marks, main = "Equal-Frequency Partitioning", col = "blue", breaks = 3)**

**hist(marks, main = "Equal-Width Partitioning", col = "green", breaks = c(50, 60, 70,**

**80))**

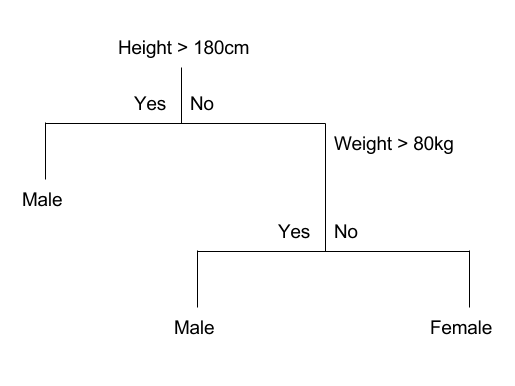
**hist(marks, main = "Clustering", col = "purple", breaks = 3)**

9.Consider this Decision tree :

a)create the data set for the below tree using ARFF format and calculate accuracy and decision for the same

b) Using this decision tree generate the rules based on rule based induction.

c) Compare both the algorithms and plot the confusion matrix.



10.Create an ARFF file for the table below and implement for the Apriori Algorithm and FP growth algorithm and compare the rules generated by both the algorithms. Identify the unique rules generated by the above algorithms.

NOTE: Assume Min\_sup=2 and confidence= 50%

|  |  |
| --- | --- |
| T.ID | ITEMS |
| T1 | SONY, BPL, LG |
| T2 | BPL, SAMSUNG |
| T3 | BPL, ONIDA |
| T4 | SONY, BPL, SAMSUNG |
| T5 | SONY, ONIDA |
| T6 | BPL, ONIDA |
| T7 | SONY, ONIDA |
| T8 | SONY, BPL, ONIDA, LG |
| T9 | SONY, BPL, ONIDA |

**11,.**The given are the strike-rates scored by a batsman in season 1 in different tournaments. 100, 70, 60, 90, 90

1. min-max normalization by setting min = 0 and max = 1
2. z-score normalization
3. z-score normalization using the mean absolute deviation instead of standard deviation
4. normalization by decimal scaling

**install.packages("dplyr") # For data manipulation**

**library(dplyr)**

**strike\_rates <- c(100, 70, 60, 90, 90)**

**min\_max\_normalized <- (strike\_rates - min(strike\_rates)) / (max(strike\_rates) - min(strike\_rates))**

**min\_max\_normalized**

**z\_score\_normalized <- (strike\_rates - mean(strike\_rates)) / sd(strike\_rates)**

**z\_score\_normalized**

**mad <- mean(abs(strike\_rates - mean(strike\_rates)))**

**mad\_normalized <- (strike\_rates - mean(strike\_rates)) / mad**

**mad\_normalized**

**k <- 2**

**decimal normalized <- strike\_rates / 10^k**

**decimal normalized**

**print("Min-Max Normalization:")**

**print(min\_max\_normalized)**

**print("Z-Score Normalization:")**

**print(z\_score\_normalized)**

**print("Z-Score Normalization with MAD:")**

**print(mad normalized)**

**print("Normalization by Decimal Scaling:")**

**print(decimal normalized)**

12.Suppose some car is tested for the AvgSpeed and TotalTime data for 9 randomly selected car with the following result

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| AvgSpeed  (in kph) | **78** | **81** | **82** | **74** | **83** | **82** | **77** | **80** | **70** |
| TotalTime  (in mins) | **39** | **37** | **36** | **42** | **35** | **36** | **40** | **38** | **46** |

1. Calculate the standard deviation of AvgSpeed and TotalTime.
2. Calculate the Variance of AvgSpeed and TotalTime for the above dataset.

13.Consider this table

1. **TID items bought**
2. **T100 {M, O, N, K, E, Y}**
3. **T200 {D, O, N, K, E, Y }**
4. **T300 {M, A, K, E}**
5. **T400 {M, U, C, K, Y}**
6. **T500 {C, O, O, K, I ,E**}
7. (a) Find all frequent item set using Apriori and FP-growth, respectively. Compare the efficiency of the two mining processes.
8. (b) List all of the strong association rules (with support s and confidence c) matching the following metarule, where X is a variable representing customers, and itemi denotes variables representing items (e.g., “A”, “B”, etc.):
9. ∀x ∈ transaction, buys(X, item1) ∧ buys(X, item2) ⇒ buys(X, item3)

**install.packages("arules") # Install the arules package**

**library(arules) # Load the arules package**

**data <- data.frame(**

**TID = c("T100", "T200", "T300", "T400", "T500"),**

**items = list(**

**c("M", "O", "N", "K", "E", "Y"),**

**c("D", "O", "N", "K", "E", "Y"),**

**c("M", "A", "K", "E"),**

**c("M", "U", "C", "K", "Y"),**

**c("C", "O", "O", "K", "I", "E")**

**))**

**# Convert the items column to a transaction format**

**transactions <- as(data$items, "transactions")**

**# Perform Apriori**

**apriori\_rules <- apriori(transactions, parameter = list(support = 0.2, confidence = 0.6))**

**# Perform FP-Growth**

**fp\_growth\_rules <- fpgrowth(transactions, parameter = list(support = 0.2, confidence = 0.6))**

**# Frequent itemsets from Apriori**

**frequent\_itemsets\_apriori <- inspect(apriori\_rules)**

**# Frequent itemsets from FP-Growth**

**frequent\_itemsets\_fpgrowth <- inspect(fp\_growth\_rules)**

**# Print the results**

**print("Frequent Itemsets from Apriori:")**

**print(frequent\_itemsets\_apriori)**

**print("Frequent Itemsets from FP-Growth:")**

**print(frequent\_itemsets\_fpgrowth)**

**# Specify the metarule**

**metarule <- "buys(X, item1) & buys(X, item2) => buys(X, item3)"**

**# Generate rules using the metarule**

**association\_rules <- apriori(transactions, parameter = list(support = 0.2, confidence = 0.6), appearance = list(lhs = metarule))**

**# Print the generated rules**

**print("Association Rules Matching the Metarule:")**

**print(association\_rules)**