Contents

Introduction 5

SECTION 1 Data Model 5

Business rules 5

Entity Relationship Diagram (ERD) 7

Entity Attribute Diagram (EAD)..7

Design..8

Table Relationship Diagram (TRD) 9

Assumptions 9

SECTION 2 Implementation 10

SQL Statements 10

SECTION 3 Queries... 12

a. What is our typical retirement age?.................................12

SQL Statement 12

Result...13

b. What is the youngest age for promotion?.........................13

SQL Statement 13

Result...14

c. Do we have at least one first aider for each of our locations? 14

Overview...14

SQL Statement 15

Result...15

d. Whose evaluations are on the decline?............................16

Overview...16

SQL Statement 16

Result...17

e. Who should be considered for promotion in 2018?............18

Overview...18

SQL Statement 18

Result...19

Conclusion...19

f. Is the training budget being shared fairly among the departments? 20
Overview 20
SQL Statement 20
Result 21
Conclusion 22
g. Is there any evidence of sexism in our organization that we should investigate further? 23
Overview 23
SQL Statements and Results 24
Conclusion 27
h. What is a 'typical' Plastic Solution career? 28
Overview 28
SQL Statement and Result 28
Conclusion 30
i. Does our training course on writing a CV help people get promoted? 30
Overview 30
SQL Statement and Result 31
j. Is our evaluation system working (that is, does it help to develop employees)? 33
Overview 33
SQL Statement and Result 33
Conclusion 34
SECTION 4 Why Do Our Employees Leave Us? 35
(I) Introduction -- Data Mining 35
(II) Apriori Algorithm 35
Category Criteria 36
Terminology 37
Step 1: Installing and calling packages “Matrix” and “arules” 38
Step 2: Applying “Apriori” and Inspect the “rules” 39

Step 3: First Experiment (“rules1” - support=0.04, confidence=0.8) --- 150 association rules. 40
Step 4: Second Experiment (“rules2”-support=0.1, confidence=0.8) ---16 association rules. 41
Step 5: Checking Redundant Rules 42
Step 6: Visualizing Association Rules 43
Step 7: Results 44
(III) K-NN Algorithm 45
Step 1: Import the data into R 46
Step 2: Pre-processing data 46
Step 3: Implementation of k-NN 49
Step 4: Results 51
Conclusion **53**
Appendices **54**
Appendix 1: Profiles 54
Appendix 2: Employee 55
Appendix 3: Retiree 56
Appendix 4: Promotion 57
Appendix 5: Evaluation 58
Appendix 6: Training 59
Appendix 7: Profiles_Training 60
References **61**

Introduction

This document is established to assist the human resources division of Plastic Solutions company to manage information relating to its employees. The information involved, which includes employees' personal, career, training, promotion and evaluation data, helps to conduct further analysis. A complete data model will be constructed, followed by an exhibition of SQL statements. Moreover, several queries raised will be addressed in a concise report with SQL technique and relevant explanations. Lastly, by adopting R language and harnessing Apriori as well as k-Nearest Neighbor (k-NN), which are believed as two appropriate approaches in this scenario, factors influencing employees' demission choices will be described and predicted.

SECTION 1 Data Model

Business rules

- An employee is a person who is currently (year 2018) serving the company
- A retiree is a person who once served the company and now left the firm.

- Promotion is the action of raising someone to a higher position.
- Evaluation is a yearly assessment about the employees' performance.
- Training is the action that employees take various courses.
- Profiles are the basic information shared by both employees and retirees.

- A piece of PROFILES can have one record of EVALUATIONS.

  Each record of EVALUATIONS belongs to a piece of PROFILES.

- A piece of PROFILES can contain one record of RETIREES.

  Each record of RETIREES is accommodated by a piece of PROFILES.

- A piece of PROFILES can consist one record of EMPLOYEES.

  Each record of EMPLOYEES is an item of a piece of PROFILES.

- A piece of PROFILES can write many records of PROMOTIONS.

  Each record of PROMOTIONS is written by a piece of PROFILES.

- A piece of PROFILES can contain many records of TRAININGS.

  Each record of TRAININGS belongs to several pieces of PROFILES.

Entity Relationship Diagram (ERD)

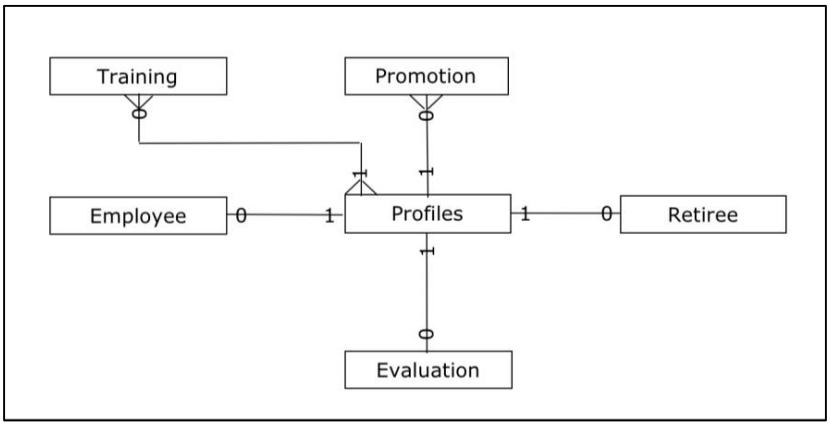


Figure 1-1. Entity Relationship Diagram

Entity Attribute Diagram (EAD)

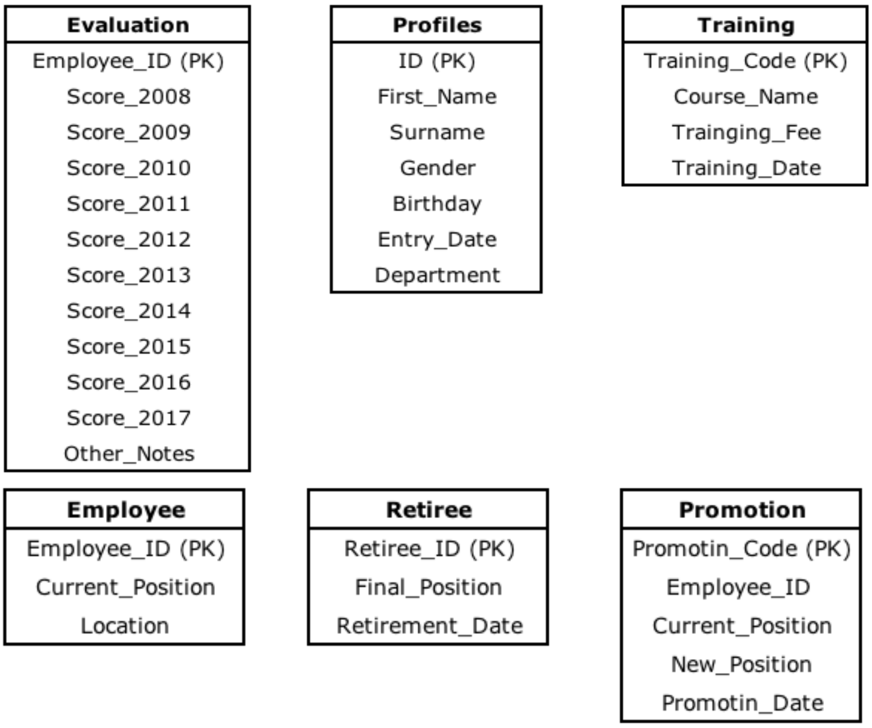


Figure 1-2. Entity Attribute Diagram

Design

- Profiles {ID (PK), First_Name, Surname, Gender, Birthday, Entry_Date, Department}

- Employee {Employee_ID (PK, FK), Current_Position, Location}

- Retiree {Retiree_ID (PK,FK), Final Position, Retirement_Date}

- Promotion {Promotion_Code(PK), Employee_ID(FK), Current_Position, New_Position, Promotion_Date}

- Evaluation {Employee_ID (PK, FK), Score_2008, Score_2009, Score_2010, Score_2011, Score_2012, Score_2013, Score_2014, Score_2015, Score_2016, Score_2017, Other_Notes}

- Training {Training_Code (PK), Course_Name, Training_Fee, Training_Date}

- Profiles_Training {Training_Code(PK, FK), Employee_ID(PK, FK)}

Table Relationship Diagram (TRD)

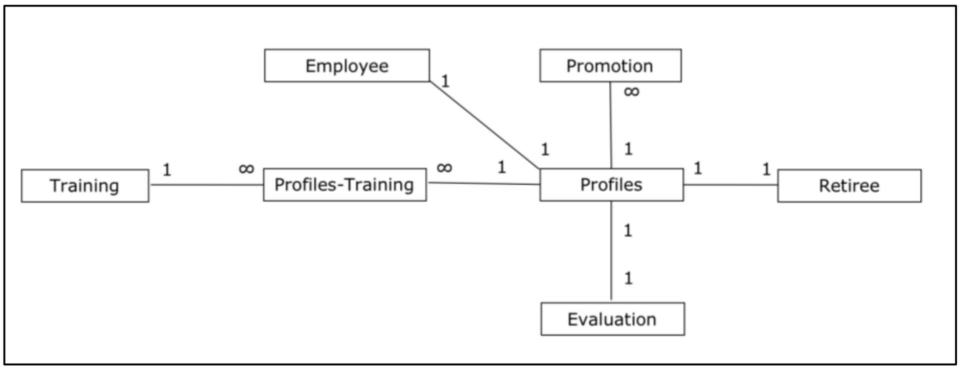


Figure 1-3. Table Relationship Diagram

Assumptions

- Promotion takes place if the evaluation score is higher than 80 unless the additional notes states otherwise.
- If one's evaluation score is under 60, he or she will be sacked.
- All employees and retirees were evaluated on a yearly basis.
- Employees enrolled before 1st July will not be evaluated in that year.

SECTION 2 Implementation

SQL Statements

```
CREATE TABLE Profiles(
ID NUMBER PRIMARY KEY,
First_Name TEXT,
Surname TEXT,
Gender TEXT,
Birthday DATE,
Entry_Date DATE
Department TEXT)
```

```
CREATE TABLE Employee(
Employee_ID NUMBER,
Current_Position TEXT,
Location TEXT,
PRIMARY KEY (Employee_ID),
FOREIGN KEY (Employee_ID) REFERENCES Profiles(ID))
```

```
CREATE TABLE Retiree(
Retiree_ID NUMBER,
Final_Position TEXT,
Retirement_Date DATE,
PRIMARY KEY (Retiree_ID),
FOREIGN KEY (Retiree_ID) REFERENCES Profiles(ID))
```

```
CREATE TABLE Promotion(
Promotion_Code NUMBER,
Employee_ID NUMBER,
Current_Position TEXT,
New_Position TEXT,
Promotion_Date DATE,
PRIMARY KEY (Promotion_Code),
FOREIGN KEY (Employee_ID) REFERENCES Profiles(ID))
```

```
CREATE TABLE Evaluation(
Employee_ID NUMBER,
Score_2008 NUMBER,
Score_2009 NUMBER,
Score_2010 NUMBER,
Score_2011 NUMBER,
Score_2012 NUMBER,
Score_2013 NUMBER,
Score_2014 NUMBER,
Score_2015 NUMBER,
Score_2016 NUMBER,
Score_2017 NUMBER,
Other_Notes TEXT,
PRIMARY KEY (Employee_ID),
FOREIGN KEY (Employee_ID) REFERENCES Profiles(ID))
```

```
CREATE TABLE Training(
Training_Code TEXT PRIMARY KEY,
Course_Name TEXT,
Training_Fee CURRENCY,
Training_Date DATE)
```

```
CREATE TABLE Profiles_Training(
Training_Code TEXT,
Employee_ID NUMBER,
PRIMARY KEY (Training_Code,Employee_ID),
FOREIGN KEY (Training_Code) REFERENCES Training(Training_Code),
FOREIGN KEY (Employee_ID) REFERENCES PROFILES(ID))
```

Figure 2-1. SQL Statements for Table Creation

- Seven tables (Profiles, Employee, Retiree, Promotion, Evaluation, Training and Profiles_Training) are created via SQL Statements above and exhibited in the Appendices.

SECTION 3 Queries

a. What is our typical retirement age?

SQL Statement

```
SELECT AVG (Retirement_Age)
AS Typical_Retirement_Age
FROM (SELECT Datediff ("yyyy", Birthday, Retirement_Date)
        AS Retirement_Age
        FROM (SELECT *
                FROM Profiles, Retiree
                WHERE Profiles.ID = Retiree.Retiree_ID))
```

Figure 3-1. Question a - SQL

1. Join the Profiles and Retiree table by matching the primary key (i.e. ID) in Profiles table and the foreign key (i.e. Retiree_ID) in Retiree table. It creates a table which contains the basic information of retirees.
2. Use the Datediff function to calculate the difference in year between a retiree's retirement date and birthday, i.e. the retirement age of each retiree.
3. Use the AVG function to calculate the average retirement age.

Result

Typical_Retirement_Age
32.4242424242424

Figure 1-2. Question a - Test Sample

The typical retirement age in the company is approximately 32.

b. What is the youngest age for promotion?

SQL Statement

```
SELECT MIN (Promotion_Age)
AS Youngest_Promotion_Age
FROM (SELECT Datediff ("yyyy", Birthday, Promotion_Date)
        AS Promotion_Age
        FROM (SELECT *
                FROM Profiles, Promotion
                WHERE Profiles.ID = Promotion.Employee_ID))
```

Figure 3-3. Question b - SQL

Similarly, two tables are joined to calculate the promotion age of each person and find the youngest figure.

Result

Youngest_Promotion_Age
23

Figure 3-2. Question b - Test Sample

The youngest promotion age in the company is 23.

c. Do we have at least one first aider for each of our locations?

Overview

A list of locations with first aiders is created to compare with total location options to justify whether at least one employee has taken the first aid course in each floor.

SQL Statement

```
SELECT *
FROM (SELECT Location FROM Employee GROUP BY Location)
WHERE Location NOT IN
                    (SELECT Location
                     FROM Profiles, Employee, Profiles_Training, Training
                     WHERE Course_Name = "First Aid Course"
                     AND Profiles.ID = Employee.Employee_ID
                     AND Profiles.ID = Profiles_Training.Employee_ID
                     AND Profiles_Training.Training_Code = Training.Training_Code
                     GROUP BY Location)
```

Figure 3-3. Question c - SQL

Apply the "NOT IN" function to make the comparison and find the location without any first aider.

Result

Location
A3

Figure 3-4. Question c - Test Sample

There are first aiders in each location except the third floor of building A.

d. Whose evaluations are on the decline?

Overview

- Only the recent three-year evaluation scores are focused in this research, i.e：2015, 2016, 2017.
- Continuously decreasing in the figures for these years refers to the decline of evaluation.

SQL Statement

```
SELECT *
FROM (SELECT Evaluation.Employee_ID, Profiles.First_Name, Profiles.Surname,
                Evaluation.Score_2015, Evaluation.Score_2016, Evaluation.Score_2017
      FROM Profiles, Evaluation, Employee
      WHERE Profiles.ID = Evaluation.Employee_ID
      AND Profiles.ID = Employee.Employee_ID)
WHERE Evaluation.Score_2017 < Evaluation.Score_2016
AND Evaluation.Score_2016 < Evaluation.Score_2015
```

Figure 3-5. Question d – SQL

1. Employee IDs, names and their corresponding evaluation scores from 2015 to 2017 can constitute a new table via joining three tables.

2. Select the data from the previous created table based on the criteria: Evaluation.Score_2015 < Evaluation.Score_2016 < Evaluation.Score_2017.

Result

Employee_ID	First_Name	Surname	Score_2015	Score_2016	Score_2017
20080203	Atwood	MACMILLAN	89	79	76
20080503	Hedy	JACOB	77	73	71
20080606	Valerie	WESLEY	74	71	68
20090208	Letitia	NOYES	84	72	68
20090211	Herman	HERTY	79	76	64
20090212	Truman	TWAIN	71	67	63
20090307	Olivia	JENKIN	77	64	61
20090407	Maria	STEINBECK	78	77	72
20100608	Irene	SPENCER	79	77	72
20100611	Marguerite	LLOYD	83	75	64
20110417	Eve	WILSON	71	69	62
20110422	Hiram	PAUL	79	74	73
20110424	Dennis	FIELD	74	71	70
20130432	Modesty	FOX	79	72	69
20130436	Saxon	EVE	73	67	60
20130438	Grover	BRUNO	76	75	70
20130442	Joseph	WILCOX	79	74	69
20130455	Kennedy	GRESHAM	77	72	70
20150510	Elsie	MORTON	79	77	74

Figure 3-6. Question d - Test Sample

The evaluations of employees in the figure 3-8 are on the decline.

e. Who should be considered for promotion in 2018?

Overview

- Since no evaluation has been made for year 2018 yet, promotion can only be judged by previous evaluation data.
- Promotion Criteria:

  Avg_Scores of employees from 2015 to 2017 >80--- Promotion.

  Other_Notes and Current_Position of employees will also be considered.

SQL Statement

```
SELECT Employee.Employee_ID, Entry_Date, First_Name, Surname, Current_Position,
        Evaluation.Score_2017, Score_2016, Score_2015,
        ROUND((Score_2017+Score_2016+Score_2015)/3,2) AS Avg_Score, Other_Notes
FROM Employee, Profiles, Evaluation
WHERE Employee.Employee_ID = Profiles.ID
AND Evaluation.Employee_ID = Profiles.ID
AND Entry_Date < #2015/07/01#
AND ROUND((Score_2017+Score_2016+Score_2015)/3,2) >= 80
ORDER BY ROUND((Score_2017+Score_2016+Score_2015)/3,2) DESC
```

Figure 3-7. Question e - SQL

Employees that entered the firm before 2015/7/1 are selected because employees would have no evaluation data for 2015 if they entered later than 2015/07/01.

Result

Employee_ID	Entry_Date	First_Name	Surname	Current_Position	Score_2017	Score_2016	Score_2015	Avg_Score	Other_Notes
20130446	2013/8/15	Matthew	BESSIE	04F	80	79	89	82.67	
20150509	2014/3/8	Hubery	TOYNBEE	05D	77	88	81	82	2015:Poor behavioral dis
20120701	2012/6/14	Adela	CRONIN	07F	79	79	87	81.67	
20080203	2008/4/5	Atwood	MACMILLAN	02E	76	79	89	81.33	2010:Made huge contribut
20140216	2014/3/8	Amy	WILCOX	02B	85	79	79	81	
20140214	2014/3/8	Angela	MAUD	02F	88	76	79	81	2015: Gained a major cli

Figure 3-8. Question e - Test Sample

Conclusion

- Employee “20140216” has the lowest current position of 02B and will be promoted to 02C or even higher.
- Employee “20150509” and “20080203” with medium position of 05D and 02E separately can also be promoted.
- Employee “20150446”, “20120701” and “20140214” have high current position level of “F”, so their promotion need to be further considered.

f. Is the training budget being shared fairly among the departments?

Overview

Judging Criteria:

- Training budget is distributed to each department according to employee number.
- BOD Department will not take any trainings.
- Data of 2018 is not complete. 2017 Training Budget Distribution will be considered.
- Assuming all training budget has been used to do trainings each year.

SQL Statement

1. The training budget of each department in 2017 is calculated:

```
SELECT Department, SUM(Training_Fee) AS 2017_Department_Training_Budget
FROM Training, Profiles_Training, Profiles
WHERE Profiles_Training.Training_Code = Training.Training_Code
AND Profiles_Training.Employee_ID = Profiles.ID
AND Training_Date BETWEEN #2017/01/01# AND #2017/12/31#
GROUP BY Department
```

Figure 3-9. Question f – SQL – 1

2. The number of employees in each department in year 2017:

```
SELECT Department, COUNT (*) AS 2017_Employee_Number
FROM Employee, Profiles
WHERE Employee.Employee_ID = Profiles.ID
AND LEFT(Employee_ID, 4) < 2018
GROUP BY Department
```

Figure 3-10. Question f - SQL - 2

3. "LEFT(Employee_ID,4)" shows the first 4 figures of Employee_ID, which are the entry years of employees.
4. "<2018" is used to deduct the employees recruited in 2018.

Result

Department	2017_Department_Training_Budget
Finance Department	€90.00
Production Department	€290.00
Purchasing Department	€170.00
Sales Department	€120.00

Department	2017_Employee_Number
BOD	7
Finance Department	13
HR Department	7
Production Department	50
Purchasing Department	13
Research & Development	7
Sales Department	12

Figure 3-11. Question f - Test Sample

- In a fair situation, average budget/person should equal to 6.57. The pie chart should be divided equally into six pieces (1/6=16.67% each department)

Department	2017 Department Training Budget (€)	2017 Employee Number	2017 Avg Budget/Person(€)
Finance Department	90	13	6.92
HR Department	0	7	0.00
Production Department	290	50	5.80
Purchasing Department	170	13	13.08
R&D Office	0	7	0.00
Sales Department	120	12	10.00
TOTAL	670	102	6.57

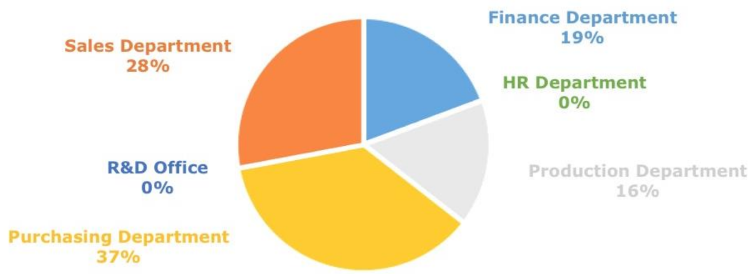


Figure 3-12. 2017 Average Training Budget Per Person (€)

Conclusion

- The average training budget per person of Purchasing Department & Sales Department are much higher than standard, accounting for 37% and 28% > 16.67%.
- HR Department & R&D Office has no training budget.
- Only training budget of Finance Department & Production Department are approximately distributed fairly.

- Overall, the training budget is not normally distributed in 2017 among departments.

g. Is there any evidence of sexism in our organization that we should investigate further?

Overview

- Whether sexism exist in this company will be tested from two dimensions.
- Dimension 1: Compare the male-female ratio in employees' enrollment and promotion. If males or females reveal an over proportional ratio in promotion, to some extent, it is possible for sexism to exist.
- Dimension 2: Compare average evaluation scores of females and males during the past 10 years. If any gender had an obvious advantage, the sexism may exist.

SQL Statements and Results

[Dimension 1]

1. Test how many males and females are enrolled in the company.

```
SELECT Gender, COUNT(*) AS Amount
FROM Profiles
GROUP BY Gender
```

Figure 3-13. Question g – SQL - 1

- **Result:** Among 148 employees, 80 females and 68 males are hired.

Gender	Amount
Female	80
Male	68

Figure 3-14. Question g - Test Sample - 1

2. Test how many males and females have ever been promoted.

```
SELECT Gender, COUNT(*) AS Promotion_Times
FROM (SELECT DISTINCT Promotion.Employee_ID, Profiles.Gender
      FROM Promotion, Profiles
      WHERE Profiles.ID = Promotion.Employee_ID)
GROUP BY Gender
```

Figure 3-15. Question g - SQL – 2

- **Result**: Among 49 employees, 34 females and 15 males have been promoted.

Gender	Promotion_Times
Female	34
Male	15

Figure 3-16. Question g -Test Sample – 2

3. Combine the results.

Gender	Amount	Proportion(Hire)	Promotion	Proportion(Promotion)
Female	80	0.540540541	34	0.693877551
Male	68	0.459459459	15	0.306122449

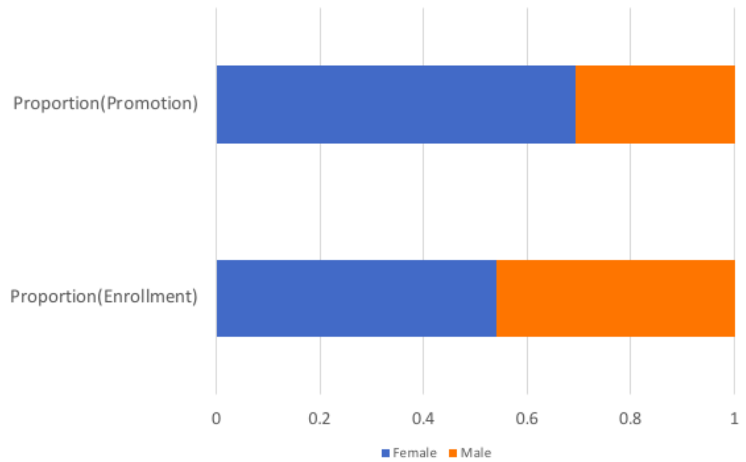


Figure 3-17. Proportion of F/M in Enrollment and Promotion

- **Description:** The percentage of female who have been promoted (69.4%) is greater than the percentage of female in the enrollment (54.1%).

[Dimension 2]

1. Select average evaluation scores of females and males in each year, for example, 2008.

```
SELECT Gender, AVG (Score_2008) AS 2008_AVG
FROM (SELECT Profiles.Gender, Evaluation.Score_2008
        FROM Profiles, Evaluation
        WHERE Profiles.ID = Evaluation.Employee_ID)
GROUP BY Gender
```

Figure 3-18. Question g - SQL - 3

- **Result**: The average evaluation score of female was 71.2 in 2008 and that for males was 74.5.

Gender	2008_AVG
Female	71.2
Male	74.5

Figure 3-19. Question g - Test Sample - 3

2. Similarly, select data in all years and observe their trends.

Gender	2008_AVG	2009_AVG	2010_AVG	2011_AVG	2012_AVG
Female	71.20	73.38	73.42	74.95	74.37
Male	74.50	72.62	72.93	74.16	73.77
Gender	**2013_AVG**	**2014_AVG**	**2015_AVG**	**2016_AVG**	**2017_AVG**
Female	74.11	73.35	73.57	73.92	73.32
Male	70.53	73.14	73.80	72.92	72.67

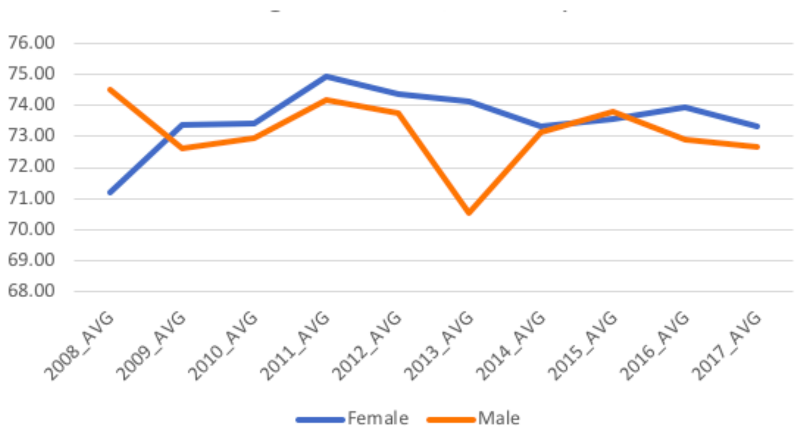


Figure 3-20. Average Scores pf F/M Each Year

- **Result:** Males' scores were generally lower than females' and had more volatilities.

Conclusion

According to two dimensions, females show advantages in both promotion proportions and average scores, meaning that sexism is likely to appear in this organization.

h. What is a 'typical' Plastic Solution career?

Overview

The typical career of employees can be inferred by their promotion frequencies. The overall trend can be tested by their promotion times. Also, how many years they took to have promotions will be analyzed.

SQL Statement and Result

1. Select how many times have employees been promoted.

```
SELECT Promotion_Times, COUNT(*) AS Number_of_Employee
FROM (SELECT COUNT(*) AS Promotion_Times
        FROM Promotion
        GROUP BY Employee_ID)
GROUP BY Promotion_Times
ORDER BY Promotion_Times ASC
```

Figure 3-21. Question h – SQL – 1

- **Results:** 32, 11, 4, 1 and 1 persons have been promoted for 1, 2, 3, 4 and 5 times, respectively. Apart from this, 99 people who had no promotion were not recorded in this table.

Promotion_Times	Number_of_Employee
1	32
2	11
3	4
4	1
5	1

Figure 3-22. Question h - Test Sample - 1

- **Explanation:** For each position level, approximately 1/3 persons have chances to be promoted to higher positions.

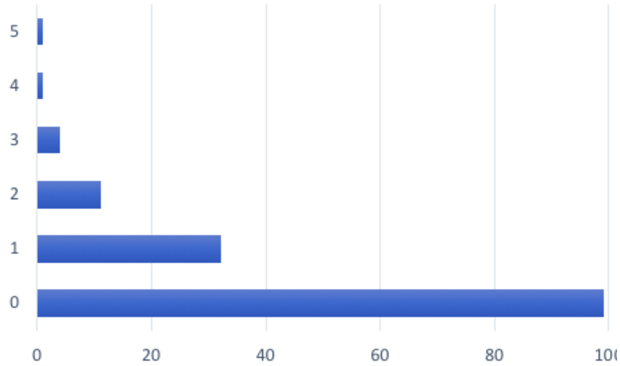


Figure 3-23. Promotion Frequencies

2. Select how many years they generally used to have a promotion

```
SELECT AVG (First_Promotion) AS AVG_First
FROM (SELECT MIN(Promotion_Year) AS First_Promotion
        FROM (SELECT Employee_ID,
                        Datediff ("yyyy", Entry_Date, Promotion_Date) AS Promotion_Year
                FROM (SELECT Promotion.Employee_ID, Promotion.Promotion_Date,
                                Profiles.Entry_Date
                        FROM Promotion, Profiles
                        WHERE Promotion.Employee_ID = Profiles.ID))
        GROUP BY Employee_ID)
```

Figure 3-24. Question h - SQL - 2

- **Result:** They usually take 2.14 years to have a promotion.

AVG_First
2.14285714285714

Figure 3-25. Question h - Test Sample - 2

Conclusion

In this company, the typical career is that 1/3 employees may have chances to promote to each higher level, and employees generally spend over 2 years to get one promotion.

i. Does our training course on writing a CV help people get promoted?

Overview

- Among employees who have attended CV training course, calculate the proportion of promoted employees.
- Among employees who have not attended CV training course, calculate the proportion of promoted employees.
- Compare two rates of two categories to show difference.

SQL Statement and Result

```
SELECT COUNT (*) AS Promotion_CV
FROM (SELECT *
        FROM Training, Profiles_Training, Profiles, Promotion
        WHERE Profiles_Training.Training_Code = Training.Training_Code
        AND Profiles.ID = Profiles_Training.Employee_ID
        AND Profiles.ID = Promotion.Employee_ID
        AND Course_Name = "CV Writing")


SELECT COUNT (*) AS CV
FROM (SELECT *
        FROM Training, Profiles_Training, Profiles
        WHERE Profiles_Training.Training_Code = Training.Training_Code
        AND Profiles.ID = Profiles_Training.Employee_ID
        AND Course_Name = "CV Writing")
```

Figure 3-26. Question i - SQL - 1

Promotion_CV
25

CV
44

Figure 27. Question i - Test Sample - 1

- Therefore, the rate of promoted employees without CV training is 25/44 = 56.8%

```
SELECT COUNT (*) AS Promotion_Without_CV

FROM (SELECT *

          FROM Training, Profiles_Training, Profiles, Promotion

          WHERE Profiles_Training.Training_Code = Training.Training_Code

          AND Profiles.ID = Profiles_Training.Employee_ID

          AND Profiles.ID = Promotion.Employee_ID

          AND Course_Name <> "CV Writing")



SELECT COUNT (*) As Without_CV

FROM (SELECT *

          FROM Training, Profiles_Training, Profiles

          WHERE Profiles_Training.Training_Code = Training.Training_Code

          AND Profiles.ID = Profiles_Training.Employee_ID

          AND Course_Name <> "CV Writing")
```

Figure 3-28. Question i - SQL - 2

Promotion_Without_CV
184

Without_CV
355

Figure 3-29. Question i - Test Sample - 2

- The rate of promoted employees without CV training is 184/355= 51.8%
- Therefore, those employees who attended CV training course have a 5% higher possibility of getting promoted than those who did not.

j. Is our evaluation system working (that is, does it help to develop employees)?

Overview

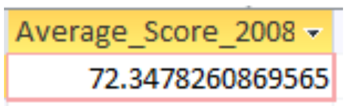
If the evaluation system has been working, then the evaluation scores from continuous years should be an upward trend.

SQL Statement and Result

```
SELECT AVG(Score_2008)
FROM Evaluation
```

Figure 3-30. Question j – SQL

- The result for year 2008 is:



Average_Score_2008
72.3478260869565

Figure 3-33. Question j - Test Sample

- Likewise, select the average scores for 2009 to 2017 and derive the result.

Year	2008	2009	2010	2011	2012
Average Evaluation Score	72.34783	73.12821	73.26667	72.34783	74.12987
Year	**2013**	**2014**	**2015**	**2016**	**2017**
Average Evaluation Score	72.64103	73.25962	73.66364	73.66364	73.04902

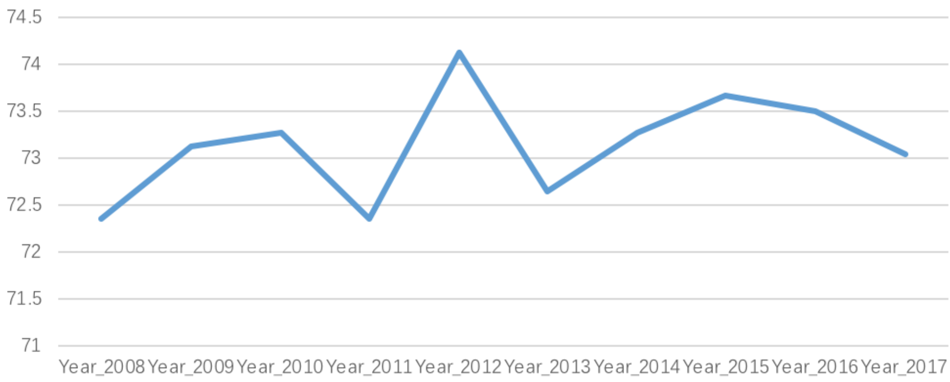


Figure 3-34. Average Evaluation Scores for 10 Years

Conclusion

A slightly upward trend is presented. Therefore, the evaluation system is working holding other variables fixed.

SECTION 4 Why Do Our Employees Leave Us?

(I) Introduction -- Data Mining

Data mining is a popular aspect of business intelligence, which is an iterative process to analyze large databases (Kuncheva, 2004). The aim of it is to extract information which is important to increase the accuracy of data analysis and decision making. More specifically, the two objectives of data mining are description and prediction. Description focuses on finding patterns and models of data and interpret them, while prediction is to discover unknown or future values using existing fields and variables. In this case, Apriori is utilized to analysis the specific route to acquire the reasons for their dismission. Alternatively, K Nearest Neighbor (k-NN) will be used to directly predict which employees are going to leave the company.

(II) Apriori Algorithm

Apriori is one of data mining algorithms, which aims to extract frequent items from large databases and obtain association rules for humans to analysis and utilize (Al-Maolegi and Arkok, 2014). A typical characteristic of Apriori is it can only handle categorical variables but not numeric variables, so firstly, all numeric ones should firstly be converted to be categorical.

Category Criteria

Firstly, according to the file HRSurveyData.csv, we classify data in each column into three types (low, medium and high) based on approximately 1/3 persons are in one group.

- Showing criteria in the following table:

	Satisfaction level	Last evaluation	Number project	Average monthly hours	Time spend in company
Low	[0,0.52]	[0,0.60]	2 or 3	[96,168]	2
Medium	(0.52,0.76]	(0.60,0.82]	4	(168,232]	3
High	(0.76,1]	(0.82,1]	5 to 7	(232,310]	4 to 10

	Work accident	Left	Promotion last 5 years
Yes	1	1	1
No	0	0	0

Figure 4-1. Apriopri – Category Criteria

- Based on the criteria, we convert the numeric into corresponding factors with R.

```
> HRSurveyData<-read.csv(file="c:/Users/micha/Desktop/HRSurveyData.csv")
> HRSurveyData$Work_accident <- factor(HRSurveyData$Work_accident,levels=c(0,1),labels=c("No","Yes"))
> HRSurveyData$left <- factor(HRSurveyData$left,levels=c(0,1),labels=c("No","Yes"))
> HRSurveyData$promotion_last_5years <- factor(HRSurveyData$promotion_last_5years,levels=c(0,1),labels=c("No","Yes
"))
> attach(HRSurveyData)
> satisfaction_level <- cut(satisfaction_level, breaks=c(0,0.52,0.76,1), labels=c("Low","Medium","High"))
> last_evaluation <- cut(last_evaluation, breaks=c(0,0.60,0.82,1), labels=c("Low","Medium","High"))
> number_of_project <- cut(number_project, breaks=c(0,3,4,7), labels=c("Low","Medium","High"))
> average_montly_hours <- cut(average_montly_hours, breaks=c(0,168,232,310), labels=c("Low","Medium","High"))
> time_spend_company <- cut(time_spend_company, breaks=c(0,2,3,10), labels=c("Low","Medium","High"))
> SURVEY <- data.frame(satisfaction_level,last_evaluation,number_of_project,average_montly_hours,time_spend_compan
y,HRSurveyData$Work_accident,HRSurveyData$left,HRSurveyData$promotion_last_5years,HRSurveyData$sales,HRSurveyData$
salary)
> str(SURVEY)
'data.frame':	14999 obs. of  10 variables:
 $ satisfaction_level                : Factor w/ 3 levels "Low","Medium",..: 1 3 1 2 1 1 1 3 3 1 ...
 $ last_evaluation                   : Factor w/ 3 levels "Low","Medium",..: 1 3 3 3 1 1 2 3 3 1 ...
 $ number_of_project                 : Factor w/ 3 levels "Low","Medium",..: 1 3 3 3 1 1 3 3 3 1 ...
 $ average_montly_hours              : Factor w/ 3 levels "Low","Medium",..: 1 3 3 2 1 1 3 3 2 1 ...
 $ time_spend_company                : Factor w/ 3 levels "Low","Medium",..: 2 3 3 3 2 2 3 3 3 2 ...
 $ HRSurveyData.Work_accident        : Factor w/ 2 levels "No","Yes": 1 1 1 1 1 1 1 1 1 1 ...
 $ HRSurveyData.left                 : Factor w/ 2 levels "No","Yes": 2 2 2 2 2 2 2 2 2 2 ...
 $ HRSurveyData.promotion_last_5years: Factor w/ 2 levels "No","Yes": 1 1 1 1 1 1 1 1 1 1 ...
 $ HRSurveyData.sales                : Factor w/ 10 levels "accounting","hr",..: 8 8 8 8 8 8 8 8 8 8 ...
 $ HRSurveyData.salary               : Factor w/ 3 levels "high","low","medium": 2 3 3 2 2 2 2 2 2 2 ...
```

Figure 4-2. Apriopri - R Statement - 1

Terminology

To obtain association rule, which is discovering relations between variables in large databases, constraints on varieties of measures of significance and interest are used. The constrains involves minimum support, minimum confidence and lift (Kotsiantis and Kanellopoulos, 2006).

- SUPPORT indicates the frequency of the itemset appears in the dataset.

$$\text{support }(A{=}{>}B) = \frac{\text{number of A and B}}{\text{total items}}$$

- CONFIDENCE reveals how often the rule has been found to be true.

$$\text{confidence }(A{=}{>}B) = \frac{\text{number of A and B}}{\text{number of A}}$$

- LIFT refers to the dependence between probability of antecedent and that of consequent. Lift =1 means they are independent of each other; lift > 1 implies their positive relevance, vice versa.

$$\text{lift }(A{=}{>}B) = \frac{\text{confidence }(A{=}{>}B)}{\text{support }(B)}$$

Step 1: Installing and calling packages "Matrix" and "arules"

```
> install.packages("Matrix")
trying URL 'https://cran.rstudio.com/bin/windows/contrib/3.5/Matrix_1.2-14.zip'
Content type 'application/zip' length 4461638 bytes (4.3 MB)
downloaded 4.3 MB

package 'Matrix' successfully unpacked and MD5 sums checked

The downloaded binary packages are in
        C:\Users\micha\AppData\Local\Temp\RtmpkTDkbc\downloaded_packages
> install.packages("arules")
trying URL 'https://cran.rstudio.com/bin/windows/contrib/3.5/arules_1.6-1.zip'
Content type 'application/zip' length 2675516 bytes (2.6 MB)
downloaded 2.6 MB

package 'arules' successfully unpacked and MD5 sums checked

The downloaded binary packages are in
        C:\Users\micha\AppData\Local\Temp\RtmpkTDkbc\downloaded_packages
> library(Matrix)
> library(arules)
```

Figure 4-3. Apriori - R Statement - 2

Step 2: Applying "Apriori" and Inspect the "rules"

```
> rules=apriori(SURVEY)
Apriori

Parameter specification:
 confidence minval smax arem  aval originalSupport maxtime support minlen maxlen target   ext
        0.8    0.1    1 none FALSE            TRUE       5     0.1      1     10  rules FALSE

Algorithmic control:
 filter tree heap memopt load sort verbose
    0.1 TRUE TRUE  FALSE TRUE    2    TRUE

Absolute minimum support count: 1499

set item appearances ...[0 item(s)] done [0.00s].
set transactions ...[34 item(s), 14999 transaction(s)] done [0.01s].
sorting and recoding items ... [25 item(s)] done [0.00s].
creating transaction tree ... done [0.01s].
checking subsets of size 1 2 3 4 5 6 7 done [0.01s].
writing ... [1083 rule(s)] done [0.00s].
creating S4 object  ... done [0.00s].
> inspect(rules)
```

```
[1081] {last_evaluation=Low,

        number_of_project=Low,

        average_montly_hours=Low,

        time_spend_company=Medium,

        HRSurveyData.Work_accident=No,

        HRSurveyData.promotion_last_5years=No} => {satisfaction_level=Low}              0.1008734   0.9305043 2.6
865513  1513
[1082] {satisfaction_level=Low,

        last_evaluation=Low,

        number_of_project=Low,

        time_spend_company=Medium,

        HRSurveyData.Work_accident=No,

        HRSurveyData.promotion_last_5years=No} => {average_montly_hours=Low}            0.1008734   0.9533711 2.8
536447  1513
[1083] {satisfaction_level=Low,

        number_of_project=Low,

        average_montly_hours=Low,

        time_spend_company=Medium,

        HRSurveyData.Work_accident=No,

        HRSurveyData.promotion_last_5years=No} => {last_evaluation=Low}                 0.1008734   0.9594166 2.8
995144  1513
```

Figure 4-4. Apriori - R Statement - 3

1083 rules are shown in total.

Step 3: First Experiment ("rules1" - support=0.04, confidence=0.8) --- 150 association rules.

```
> rules1 <- apriori(SURVEY, parameter = list(minlen=1, supp=0.04, conf=0.8), appearance = list(rhs="HRSurveyData.l
eft=Yes", default="lhs"), control = list(verbose=F))
> rules1.sorted <- sort(rules1, by="lift")
> inspect(rules1.sorted)
      lhs                                             rhs                        support confidence     lift count
[1]   {satisfaction_level=Low,
       last_evaluation=Low,
       number_of_project=Low,
       average_montly_hours=Low,
       time_spend_company=Medium,
       HRSurveyData.Work_accident=No,
       HRSurveyData.salary=low}                   => {HRSurveyData.left=Yes} 0.05907060  0.9568035 4.018789   886
[2]   {satisfaction_level=Low,
       last_evaluation=Low,
       number_of_project=Low,
       average_montly_hours=Low,
       time_spend_company=Medium,
       HRSurveyData.Work_accident=No,
       HRSurveyData.promotion_last_5years=No,
       HRSurveyData.salary=low}                   => {HRSurveyData.left=Yes} 0.05873725  0.9565689 4.017804   881
[3]   {satisfaction_level=Low,
       last_evaluation=Low,
       number_of_project=Low,
       average_montly_hours=Low,
       time_spend_company=Medium,
       HRSurveyData.salary=low}                   => {HRSurveyData.left=Yes} 0.06200413  0.9528689 4.002263   930
[4]   {satisfaction_level=Low,
       last_evaluation=Low,

                                    ......

       HRSurveyData.promotion_last_5years=No}     => {HRSurveyData.left=Yes} 0.08587239  0.8029925 3.372748  1288
[148] {last_evaluation=Low,
       number_of_project=Low,
       time_spend_company=Medium,
       HRSurveyData.Work_accident=No,
       HRSurveyData.salary=low}                   => {HRSurveyData.left=Yes} 0.05953730  0.8008969 3.363946   893
[149] {last_evaluation=Low,
       number_of_project=Low,
       time_spend_company=Medium,
       HRSurveyData.Work_accident=No,
       HRSurveyData.promotion_last_5years=No,
       HRSurveyData.salary=low}                   => {HRSurveyData.left=Yes} 0.05920395  0.8007214 3.363209   888
[150] {satisfaction_level=Low,
       number_of_project=Low,
       average_montly_hours=Low,
       HRSurveyData.Work_accident=No}             => {HRSurveyData.left=Yes} 0.09653977  0.8004422 3.362037  1448
```

Figure 4-5. Apriori - R Statement - 4

- Set "rhs" as "HRSurveyData.left=Yes" because only employees leaving the company are our focus.
- 150 pieces of records satisfy the restrictions, meaning the filtered conditions should be further narrowed.

Step 4: Second Experiment ("rules2"-support=0.1,

confidence=0.8) ---16 association rules.

```
> rules2 <- apriori(SURVEY, parameter = list(minlen=1, supp=0.1, conf=0.8), appearance = list(rhs="HRSurveyData.le
ft=Yes", default="lhs"), control = list(verbose=F))
> rules2.sorted <- sort(rules2, by="lift")
> inspect(rules2.sorted)
     lhs                                           rhs                         support   confidence lift     count
[1]  {satisfaction_level=Low,
      last_evaluation=Low,
      number_of_project=Low,
      average_montly_hours=Low,
      time_spend_company=Medium,
      HRSurveyData.promotion_last_5years=No}  => {HRSurveyData.left=Yes} 0.1001400 0.9387500 3.942960 1502
[2]  {satisfaction_level=Low,
      last_evaluation=Low,
      number_of_project=Low,
      average_montly_hours=Low,
      time_spend_company=Medium}              => {HRSurveyData.left=Yes} 0.1009401 0.9374613 3.937547 1514
[3]  {satisfaction_level=Low,
      last_evaluation=Low,
      average_montly_hours=Low,
      time_spend_company=Medium,
      HRSurveyData.promotion_last_5years=No}  => {HRSurveyData.left=Yes} 0.1002067 0.9087062 3.816769 1503
[4]  {satisfaction_level=Low,
      last_evaluation=Low,
      average_montly_hours=Low,
      time_spend_company=Medium}              => {HRSurveyData.left=Yes} 0.1010067 0.9077292 3.812666 1515
[5]  {satisfaction_level=Low,
      number_of_project=Low,
      average_montly_hours=Low,
      time_spend_company=Medium,
      HRSurveyData.promotion_last_5years=No}  => {HRSurveyData.left=Yes} 0.1001400 0.8945801 3.757437 1502
[6]  {satisfaction_level=Low,
      number_of_project=Low,
      average_montly_hours=Low,
      time_spend_company=Medium}              => {HRSurveyData.left=Yes} 0.1009401 0.8921626 3.747283 1514
[7]  {satisfaction_level=Low,
      last_evaluation=Low,
      number_of_project=Low,
      time_spend_company=Medium,
      HRSurveyData.promotion_last_5years=No}  => {HRSurveyData.left=Yes} 0.1004067 0.8900709 3.738497 1506

[8]  {satisfaction_level=Low,
      last_evaluation=Low,
      number_of_project=Low,
      time_spend_company=Medium}              => {HRSurveyData.left=Yes} 0.1012067 0.8892794 3.735173 1518
[9]  {satisfaction_level=Low,
      last_evaluation=Low,
      number_of_project=Low,
      average_montly_hours=Low,
      HRSurveyData.promotion_last_5years=No}  => {HRSurveyData.left=Yes} 0.1003400 0.8790888 3.692370 1505
[10] {satisfaction_level=Low,
      last_evaluation=Low,
      number_of_project=Low,
      average_montly_hours=Low}               => {HRSurveyData.left=Yes} 0.1011401 0.8748558 3.674590 1517
[11] {last_evaluation=Low,
      number_of_project=Low,
      average_montly_hours=Low,
      time_spend_company=Medium,
      HRSurveyData.promotion_last_5years=No}  => {HRSurveyData.left=Yes} 0.1001400 0.8632184 3.625711 1502
[12] {last_evaluation=Low,
      number_of_project=Low,
      average_montly_hours=Low,
      time_spend_company=Medium}              => {HRSurveyData.left=Yes} 0.1009401 0.8616961 3.619317 1514
```

```
[13] {satisfaction_level=Low,
      average_montly_hours=Low,
      time_spend_company=Medium,
      HRSurveyData.promotion_last_5years=No} => {HRSurveyData.left=Yes} 0.1006067  0.8286656 3.480581  1509
[14] {satisfaction_level=Low,
      average_montly_hours=Low,
      time_spend_company=Medium}            => {HRSurveyData.left=Yes} 0.1014068  0.8266304 3.472033  1521
[15] {satisfaction_level=Low,
      last_evaluation=Low,
      time_spend_company=Medium,
      HRSurveyData.promotion_last_5years=No} => {HRSurveyData.left=Yes} 0.1011401  0.8164693 3.429354  1517
[16] {satisfaction_level=Low,
      last_evaluation=Low,
      time_spend_company=Medium}            => {HRSurveyData.left=Yes} 0.1019401  0.8145978 3.421493  1529
```

Figure 4-6. Apriori - R Statement - 5

- The satisfied association rules are reduced into 16 records.

Step 5: Checking Redundant Rules

```
> subset.matrix=is.subset(rules2.sorted, rules2.sorted, sparse = FALSE)
> subset.matrix[lower.tri(subset.matrix, diag=T)] = NA
> redundant <- colSums(subset.matrix, na.rm=T) >= 1
> which(redundant)
named integer(0)
```

Figure 4-7. Apriori - R Statement - 6

- A specific rule may be prudent if it has a lower lift/confidence compared with another more general rule (same “rhs”, but less items in “lhs”).
- The above codes are used to check the redundant rules and “named integer (0)” is displayed, meaning that no redundant rules are found in the 16 records of “rules2”.

Step 6: Visualizing Association Rules

```
> install.packages(c("arules", "scatterplot3d", "vcd", "seriation", "igraph", "grid", "cluster", "TSP", "gclus", "
colorspace"))
```

```
> install.packages("arulesViz")
```

```
> library(arulesViz)
> plot(rules2.sorted)
```

Figure 4-8. Apriori - R Statement - 7

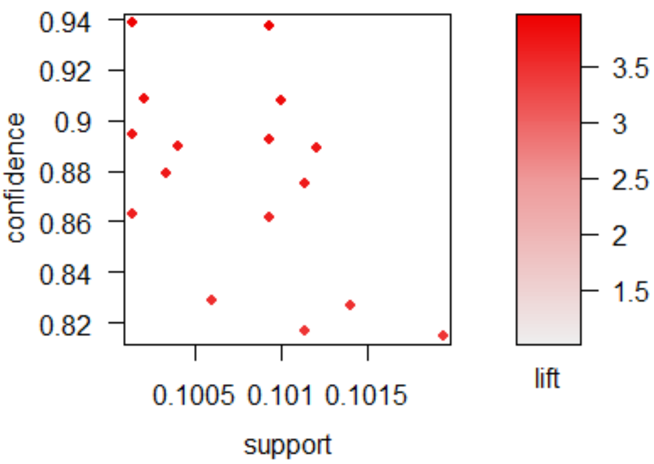


Figure 4-9. Apriori - Scatter Plot for 16 Rules

- After installing the ancillary packages, the association rules in the "rules2.sorted" can be visualized.
- "lift": A significant positive link between "lhs" and "rhs" with a color of deep red.
- "support": The 16 datasets have records number between 1,500(0.1*14,999)-1530(0.102*14,999), which is not significantly different.
- "confidence" will be the focus.

Step 7: Results

- The 16 datasets sorted above include 6 variables, which may be the main reasons why people leave:

1. satisfaction_level=Low= [0,0.52]
2. last_evaluation=Low= [0,0.60]
3. number_of_project=Low= 2 or 3
4. average_Monthly_hours=Low = [96,168]
5. Time_spend_company=Medium= 3
6. Promotion_last_5years= No

- The datasets above can also be used to predict the people who may leave the company in the future. E.g. {1,2,3,4,5,6}, {1,2,3,4,5}, {1,2,4,5,6}, {1,2,4,5}, which has the confidence over 0.90, support over 0.10 and lift over 3.80, can be used as prediction combinations.

(III) K-NN Algorithm

To predict whether an employee is going to leave, k-NN algorithm can be used to classify a data point by using the existing data in the database. The object is assigned to the class based on feature similarities (Sutton, 2012). The test sample in figure 4-10 is defined as red when k=3 but blue if k=10 because the classification depends on the majority votes of its k nearest neighbors.

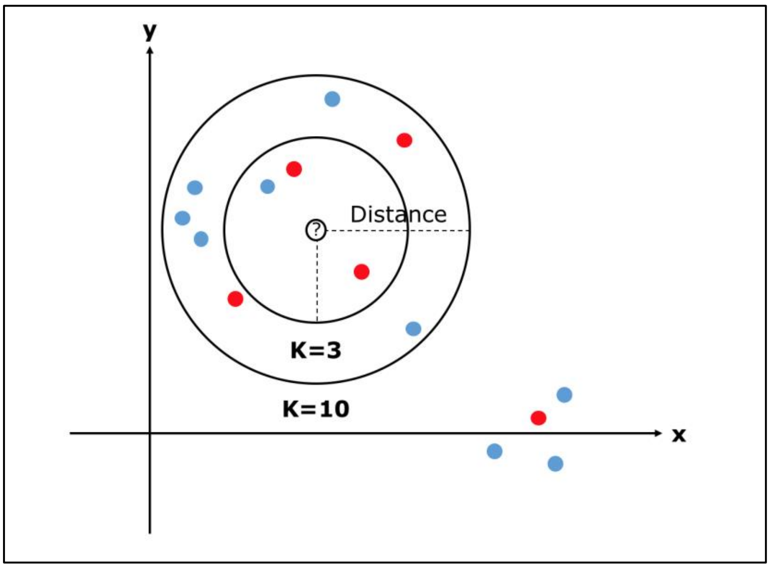


Figure 4-10. k-NN – Principle

Step 1: Import the data into R

```
> HRSurveyData<-read.csv(file="~/Desktop/HRSurveyData.csv")
> HRSurveyData
  satisfaction_level last_evaluation number_project average_monthly_hours time_spend_company work_accident promotion_last_5years
1               0.15            0.98              2                    96                  2             0                     0
2               0.34            0.67              5                    96                  2             1                     0
3               0.40            0.48              4                    97                  2             0                     0
4               0.31            0.61              4                    97                  2             0                     0
5               0.36            0.66              4                    97                  2             0                     0
6               0.36            0.69              3                    98                  2             0                     0
```

Figure 4-11. k-NN - R Statement - 1

Step 2: Pre-processing data

- Convert the factor-type variables (department and salary) to numeric ones for the standardization of calculation.

```
> str(HRSurveyData)
'data.frame':	14999 obs. of  10 variables:
 $ satisfaction_level   : num  0.15 0.34 0.4 0.31 0.36 0.36 0.77 0.61 0.3 0.61 ...
 $ last_evaluation      : num  0.98 0.67 0.48 0.61 0.66 0.69 0.42 0.99 0.54 0.39 ...
 $ number_project       : int  2 5 4 4 4 3 4 5 2 3 ...
 $ average_monthly_hours: int  96 96 97 97 97 98 98 98 99 99 ...
 $ time_spend_company   : int  2 2 2 2 2 2 2 2 2 2 ...
 $ work_accident        : int  0 1 0 0 0 0 0 0 0 0 ...
 $ promotion_last_5years: int  0 0 0 0 0 0 0 0 0 0 ...
 $ department           : Factor w/ 10 levels "accounting","hr",..: 8 3 3 9 8 8 8 9 7 9 ...
 $ salary               : Factor w/ 3 levels "high","low","medium": 1 2 2 2 1 2 2 3 3 2 ...
 $ left                 : int  0 0 0 0 0 0 0 0 0 0 ...
> HRSurveyData$salary = as.numeric(HRSurveyData$salary)
> HRSurveyData$department = as.numeric(HRSurveyData$department)
> str(HRSurveyData)
'data.frame':	14999 obs. of  10 variables:
 $ satisfaction_level   : num  0.15 0.34 0.4 0.31 0.36 0.36 0.77 0.61 0.3 0.61 ...
 $ last_evaluation      : num  0.98 0.67 0.48 0.61 0.66 0.69 0.42 0.99 0.54 0.39 ...
 $ number_project       : int  2 5 4 4 4 3 4 5 2 3 ...
 $ average_monthly_hours: int  96 96 97 97 97 98 98 98 99 99 ...
 $ time_spend_company   : int  2 2 2 2 2 2 2 2 2 2 ...
 $ work_accident        : int  0 1 0 0 0 0 0 0 0 0 ...
 $ promotion_last_5years: int  0 0 0 0 0 0 0 0 0 0 ...
 $ department           : num  8 3 3 9 8 8 8 9 7 9 ...
 $ salary               : num  1 2 2 2 1 2 2 3 3 2 ...
 $ left                 : int  0 0 0 0 0 0 0 0 0 0 ...
```

Figure 4-12. k-NN - R Statement - 2

- After checking the top 6 observations, it is found that all of them are left employees, which indicates that the frame of the data is too organized into order of "left" = 0 or 1. So firstly the order of rows need to be mixed to ensure the accuracy of random sampling.

```
> head(HRSurveyData)
  satisfaction_level last_evaluation number_project average_monthly_hours time_spend_company work_accident promotion_last_5years department
1               0.15            0.98              2                    96                  2             0                     0          8
2               0.34            0.67              5                    96                  2             1                     0          3
3               0.40            0.48              4                    97                  2             0                     0          3
4               0.31            0.61              4                    97                  2             0                     0          9
5               0.36            0.66              4                    97                  2             0                     0          8
6               0.36            0.69              3                    98                  2             0                     0          8
  salary left
1      1    0
2      2    0
3      2    0
4      2    0
5      1    0
6      2    0
```

Figure 4-13. k-NN - R Statement - 3

- Use a random number generator to produce 14,999 random numbers from uniform distribution.

```
> set.seed(9850)
> ID<-runif(nrow(HRSurveyData))
> ID
  [1] 7.495759e-01 9.970860e-01 6.520020e-01 4.329283e-01 3.323124e-01 8.654065e-01 1.793312e-01 4.784937e-01 2.957953e-01 6.644066e-01
 [11] 7.117703e-01 7.801193e-01 1.356792e-01 8.191885e-02 6.316598e-01 5.296842e-01 1.708337e-02 9.474428e-01 2.882614e-01 1.024376e-02
 [21] 8.897228e-01 9.983856e-02 9.765785e-01 6.263110e-01 8.189664e-01 8.142171e-01 2.289383e-01 3.933451e-01 9.630646e-01 1.585229e-01
 [31] 3.653627e-01 8.156729e-01 1.604756e-01 3.250755e-01 9.560772e-01 2.218299e-01 2.403589e-01 6.548423e-01 8.333722e-01 2.233174e-01
 [41] 5.673195e-01 6.456432e-01 4.659324e-02 1.668618e-01 9.560074e-02 2.807087e-01 2.743727e-01 3.060213e-01 4.662362e-01 7.148338e-01
 [51] 8.190107e-01 4.139657e-01 3.348089e-02 1.631715e-01 6.145222e-01 6.255911e-01 9.879692e-01 5.909557e-01 2.919432e-01 8.480136e-01
```

Figure 4-14. k-NN - R Statement - 4

- Rank all the rows in order of the 14,999 random numbers generated before. Check if the observations have been mixed up.

```
> HRSurveyData<-HRSurveyData[order(ID),]
> str(HRSurveyData)
'data.frame':	14999 obs. of  10 variables:
 $ satisfaction_level   : num  0.92 0.58 0.82 0.87 0.73 0.72 0.84 0.59 0.68 0.39 ...
 $ last_evaluation      : num  0.97 0.76 0.91 0.84 1 0.53 0.43 0.56 0.92 0.5 ...
 $ number_project       : int  4 4 5 5 4 3 6 4 3 4 ...
 $ average_monthly_hours: int  238 197 276 137 146 179 246 250 226 294 ...
 $ time_spend_company   : int  5 5 6 2 3 3 4 2 2 3 ...
 $ work_accident        : int  1 0 0 0 0 0 0 0 0 0 ...
 $ promotion_last_5years: int  0 0 0 0 0 0 0 0 0 0 ...
 $ department           : num  9 8 6 6 8 9 10 1 3 10 ...
 $ salary               : num  3 2 3 2 3 2 3 3 1 2 ...
 $ left                 : int  0 0 1 0 0 0 0 0 0 1 ...
```

Figure 4-15. k-NN - R Statement - 5

- Rescale and normalize the numerical variables using the function x-min(x)/(max(x)-min(x)) to standardize the effect on distance caused by different dimensions.

```
> normalize<-function(x){
+   +return((x-min(x))/(max(x)-min(x)))}
> survey<-as.data.frame(lapply(HRSurveyData[,c(1,2,3,4,5,6,7,8,9)],normalize))
> str(survey)
'data.frame':	14999 obs. of  9 variables:
 $ satisfaction_level   : num  0.912 0.538 0.802 0.857 0.703 ...
 $ last_evaluation      : num  0.953 0.625 0.859 0.75 1 ...
 $ number_project       : num  0.4 0.4 0.6 0.6 0.4 0.2 0.8 0.4 0.2 0.4 ...
 $ average_monthly_hours: num  0.664 0.472 0.841 0.192 0.234 ...
 $ time_spend_company   : num  0.375 0.375 0.5 0 0.125 0.125 0.25 0 0 0.125 ...
 $ work_accident        : num  1 0 0 0 0 0 0 0 0 0 ...
 $ promotion_last_5years: num  0 0 0 0 0 0 0 0 0 0 ...
 $ department           : num  0.889 0.778 0.556 0.556 0.778 ...
 $ salary               : num  1 0.5 1 0.5 1 0.5 1 1 0 0.5 ...
> summary(survey)
 satisfaction_level last_evaluation  number_project   average_monthly_hours time_spend_company work_accident    promotion_last_5years
 Min.   :0.0000     Min.   :0.0000   Min.   :0.0000   Min.   :0.0000        Min.   :0.0000     Min.   :0.0000   Min.   :0.00000
 1st Qu.:0.3846     1st Qu.:0.3125   1st Qu.:0.2000   1st Qu.:0.2804        1st Qu.:0.1250     1st Qu.:0.0000   1st Qu.:0.00000
 Median :0.6044     Median :0.5625   Median :0.4000   Median :0.4860        Median :0.1250     Median :0.0000   Median :0.00000
 Mean   :0.5745     Mean   :0.5564   Mean   :0.3606   Mean   :0.4909        Mean   :0.1873     Mean   :0.1446   Mean   :0.02127
 3rd Qu.:0.8022     3rd Qu.:0.7969   3rd Qu.:0.6000   3rd Qu.:0.6963        3rd Qu.:0.2500     3rd Qu.:0.0000   3rd Qu.:0.00000
 Max.   :1.0000     Max.   :1.0000   Max.   :1.0000   Max.   :1.0000        Max.   :1.0000     Max.   :1.0000   Max.   :1.00000
   department         salary
 Min.   :0.0000   Min.   :0.0000
 1st Qu.:0.4444   1st Qu.:0.5000
 Median :0.7778   Median :0.5000
 Mean   :0.6596   Mean   :0.6736
 3rd Qu.:0.8889   3rd Qu.:1.0000
 Max.   :1.0000   Max.   :1.0000
```

Figure 4-16. k-NN - R Statement - 6

Step 3: Implementation of k-NN

- Create a train data set including 90% of the observations and a test data set including the remaining 10% of the observations from the normalized data frame, which are used to test how well the model makes prediction.

```
> survey_train<-survey[1:13499,]
> survey_test<-survey[13500:14999,]
```

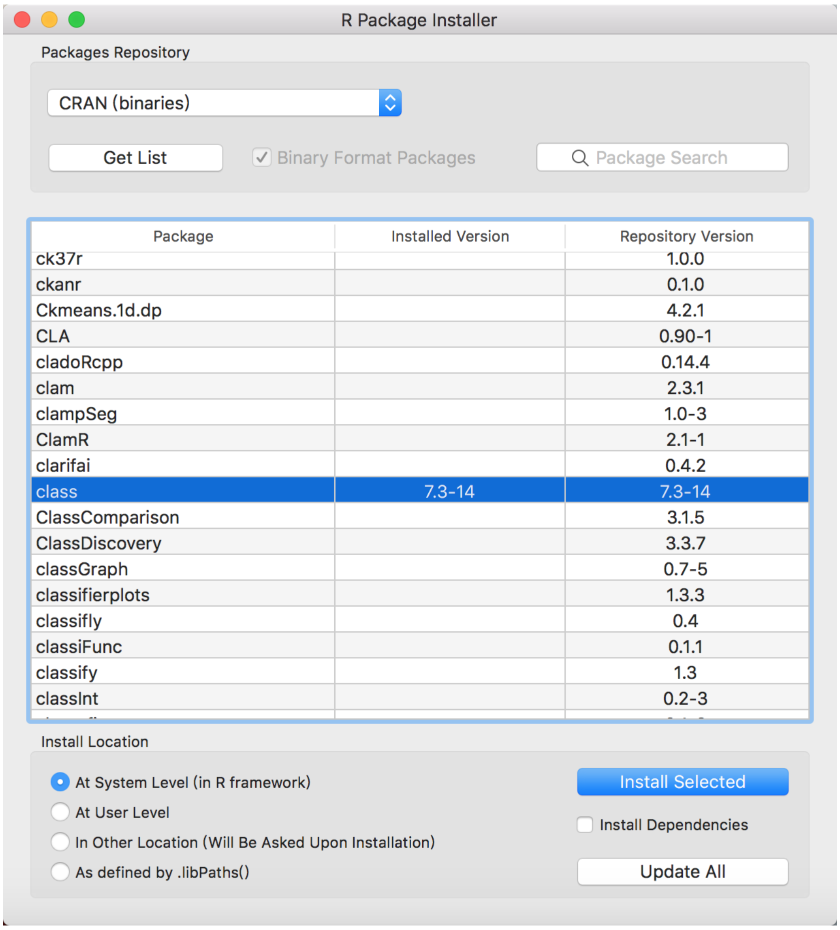
Figure 4-17. k-NN - R Statement – 7

- Set the 10th column (left) as the target for both the train and the test data set.

```
> survey_train_target<-HRSurveyData[1:13499,10]
> survey_test_target<-HRSurveyData[13500:14999,10]
```

Figure 4-18. k-NN - R Statement - 8

- Pre-install the class package where the k-NN algorithm locates, and recall it using the require function.




```
> require(class)
Loading required package: class
trying URL 'https://cran.ma.imperial.ac.uk/bin/macosx/el-capitan/contrib/3.4/class_7.3-14.tgz'
Content type 'application/x-gzip' length 95032 bytes (92 KB)
==================================================
downloaded 92 KB


The downloaded binary packages are in
	/var/folders/hq/j7rzxnmd6zg62n4hncxxxzfw0000gn/T//Rtmpo5ckir/downloaded_packages
```

Figure 4-19. k-NN - R Statement - 9

- Calculate the square root of the number of total observations. Recognize its odd integer as the value of k, a representation of the number of voters, to avoid the ties (Hassanat *et al.*, 2014). Implement k-NN algorithm and the prediction for classifying all values in the test data set will be stored in m1.

```
> sqrt(15000)
[1] 122.4745
> m1<-knn(train=survey_train,test=survey_test,cl=survey_train_target,k=123)
> m1
   [1] 0 1 0 1 1 0 1 0 0 0 0 1 0 1 0 0 0 0 0 0 0 0 0 1 0 0 1 0 0 0 0 0 1 0 0 1 0 0 0 0 0 0 0 0 1 1 0 0 0 0 0 0 1 0 0 0 1 0 0 0 1 0 0 0 0 0 0
  [68] 0 0 1 0 1 0 0 0 0 0 1 0 0 0 1 0 0 0 0 0 0 1 0 0 0 0 0 1 0 0 0 1 0 1 0 0 1 0 1 0 0 0 0 0 1 1 1 1 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 1 0 1 1
 [135] 0 0 0 1 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 1 1 0 0 1 0 0 0 1 0 1 1 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 1 0 0 1 0 1 0 0 0 0 0 1 0 1 0 0 0 0 1 0
 [202] 0 0 0 0 0 1 0 0 0 1 0 1 1 0 1 0 0 1 1 1 1 0 1 0 0 1 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 1 1 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 1 1 0 0 1 0 0
 [269] 0 0 0 0 0 0 0 0 1 1 1 1 0 0 0 0 0 1 0 0 0 0 0 0 1 1 0 0 0 0 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 1 0 0 0 0 0 0 0 1 0 0 0 0 0 0 1 0 0 0 0
 [336] 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 0 0 0 0 1 1 0 1 0 0 0 1 0 0 0 0 0 0 0 1 1 0 0 0 0 0 1 0 0 1 0 0 0 0 0 1 1 0 0 0 0 0 0 0 0 0 0 1
 [403] 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 1 0 1 1 1 0 0 0 0 0 1 0 0 0 0 0 0 0 0 1 1 1 0 0 0 1
 [470] 1 0 0 0 0 0 0 0 1 0 0 1 0 0 1 0 0 0 0 0 1 0 1 0 0 0 0 0 0 0 1 0 1 0 1 0 0 0 0 0 1 0 0 0 0 0 1 1 0 1 0 1 0 0 0 0 0 0 1 0 1 0 0 0 0 0 0 0
 [537] 0 0 0 0 0 1 0 0 0 0 1 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 1 0 0 1 1 0 0 0 1 1 1 0 0 0 0 0 0 1 0 0 1 0 0 0 1 0 0 0 0 0 0 0 1 1 0
 [604] 0 0 1 0 0 1 1 0 0 1 1 0 0 1 0 0 0 0 1 1 1 0 1 0 0 1 0 0 0 1 0 0 0 0 0 0 1 0 0 0 0 1 1 0 1 0 0 0 1 0 0 0 0 0 0 0 1 0 0 1 0 0 0 0 0 0 0 1
 [671] 0 0 0 0 0 0 0 0 0 0 1 0 1 0 0 1 0 1 1 1 0 0 1 1 0 0 1 0 0 1 0 1 0 0 0 0 0 0 0 0 0 0 0 0 1 1 0 0 0 1 0 0 1 0 0 0 0 1 0 0 0 1 0 0 0 1 0 0
 [738] 1 0 0 0 0 0 0 1 0 0 0 1 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 1 1 0 1 0 0 0 0 0 0 1 1 0 0 0 0 0 0 0 0 1 0
 [805] 1 0 0 0 1 1 1 0 0 0 1 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 1 0 0 1 0 0 1 0 0 0 0 0 0 0 0 1 0 0 0 0 0 1 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
 [872] 0 0 1 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 1 0 1 0 0 0 0 0 0 1 1 0 0 0 0 1 0 1 1 0 1 0 0 1 0 0 0 1 0 1
 [939] 1 0 0 0 0 0 0 0 1 0 0 0 1 0 0 0 0 0 0 0 0 0 0 1 1 0 0 0 0 1 0 1 0 1 1 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 1 0 0 0 0 0 0 0 0 1 1 0 0
[1006] 0 0 0 0 0 0 0 1 1 0 0 0 1 0 1 0 0 0 0 0 0 0 1 0 1 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 1 0 1 0 0 0 0 0 0 1 0 0 1 0 0 0 1 0 0 1 0 1 0
[1073] 0 1 0 0 1 0 1 0 0 1 1 0 1 0 1 0 0 0 0 0 0 1 0 0 0 0 1 0 0 1 1 1 1 1 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 1 1 1 1 0 0 0 0 0 1 0 0 0
[1140] 1 0 0 0 1 1 0 0 0 1 1 0 0 0 1 1 0 0 0 0 0 0 0 0 0 1 0 0 0 1 0 1 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 1 1 0 0 0 0 0 0 0 0 0 0 0 1 1 1 0 0 1
[1207] 0 1 0 0 0 0 1 0 1 0 0 1 0 1 0 0 0 0 0 0 1 0 1 1 0 0 0 0 0 0 0 0 1 0 0 0 1 0 0 0 0 0 0 0 0 0 0 1 0 0 0 1 1 1 1 1 0 0 0 0 0 0 1 1 0 0 0 0 0 1
[1274] 0 0 0 0 0 0 1 0 0 0 0 0 0 0 1 0 0 0 0 1 0 0 0 0 0 0 0 1 1 1 0 0 0 0 1 0 0 1 0 0 0 0 0 0 1 1 0 0 0 0 1 0 1 0 0 0 0 1 0 0 0 1 0 0 0 0 0
[1341] 0 0 0 1 0 0 0 0 0 0 1 1 1 0 0 1 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 1 1 0 0 1 0 0 0 0 0 0 0 0 1 0 0 0 1 0 1 0 0 0 0 0 0 0 0 0 1 0 1
[1408] 0 0 0 0 1 0 0 0 0 0 0 0 0 1 0 0 1 0 1 0 0 0 0 0 0 0 1 0 0 0 0 0 1 0 0 0 1 0 0 1 0 0 0 0 1 1 0 0 0 0 0 0 0 0 1 0 1 0 0 0 0 0 0 0 1 0 1 0 0 0
[1475] 1 0 0 1 0 0 0 1 0 0 0 0 0 1 0 0 0 0 0 0 0 0 1 0 0 0 0
Levels: 0 1
```

Figure 4-20. k-NN - R Statement - 10

Step 4: Results

- Compare how the prediction of all tested observations different from their original values.

```
> table(survey_test_target,m1)
                  m1
survey_test_target    0    1
                 0 1095   60
                 1   55  290
```

Figure 4-21. k-NN - R Statement - 11

- The results of 1095 current employees and 290 retirees in the sample are consistent with their actual values (i.e. (0,0), (1,1)). However, there are still 55 retired employees who have characteristics that are similar to those of current ones, and 60 employees are similar to retirees (i.e.

(1,0), (0,1)). This final group can be defined as staff who are predicted to leave soon since these employees stand in locations that are close to the cluster of retired workers. The point B in figure 4-22 can clearly illustrate this relationship.

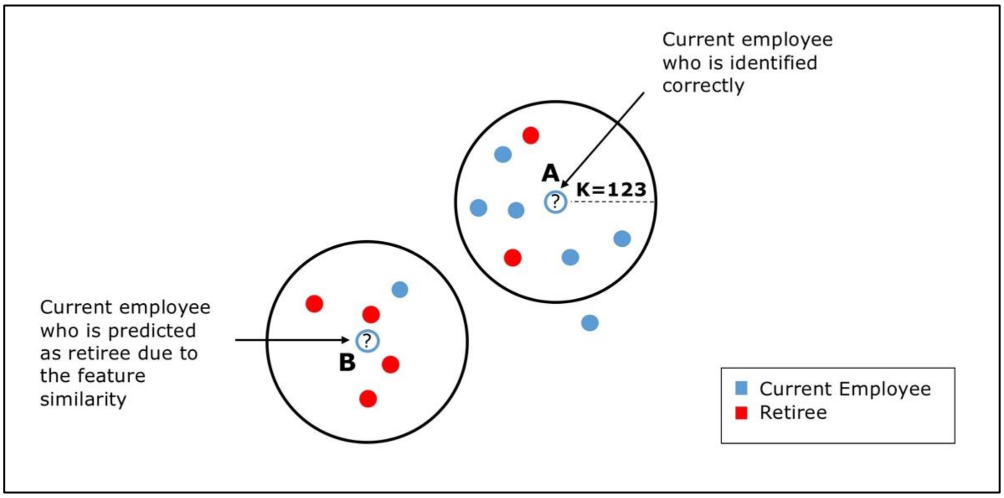


Figure 4-22. k-NN – Result of Prediction

Conclusion

This report utilizes both database construction and data analysis to fulfil Plastic Solution Human Resource department's demands. In the former sector, an aggregate profiles table is created to encompass foreign keys (also primary keys) of both employees and retirees. This tackles the difficulties of including two primary keys into one foreign key of one table. Comparing to incorporate all the data into one giant table, this approach allocates diverse spheres into appropriate tables, which shows more conciseness and leaves fewer null fields. In the latter sector, Aprioir is adopted to investigate the data sets. By testing their accuracy, the factors that influences employees' resigning decisions emerge. However, although Aprioir algorithm works in this scenario, it takes a large amount of calculation resources to scan the database repeatedly (Rao and Gupta, 2012). If the number of data sets dramatically increases, considerable amount of time will be spent to calculate the result. As for the k-NN, it functions well in this case with 15 thousand employees and retirees since it is more effective if the input data is huge. Its drawback is similar to that of Aprioir. Because the distance of each training sample to its corresponding query instance need to be calculated, considerable computational cost may incur.

Appendices

Appendix 1: Profiles

Profiles

ID	First_Name	Surname	Gender	Birthday	Entry_Date	Department
20080101	Archer	MICHESON	Male	1979/3/19	2008/3/19	BOD
20080102	Erica	CROMWELL	Female	1981/5/21	2008/3/19	BOD
20080103	Bruno	CARROLL	Male	1979/12/3	2008/3/19	BOD
20080104	Emmanuel	JONAH	Male	1980/4/24	2008/3/19	BOD
20080105	Jodie	BRYCE	Female	1978/2/24	2008/3/19	BOD
20080106	Ulysses	RAMAN	Male	1979/7/3	2008/3/19	BOD
20080107	Everley	COOK	Male	1983/3/15	2008/3/19	BOD
20080201	Aries	EVELINA	Male	1980/9/2	2008/3/26	Finance Department
20080202	Ama	BERTIE	Female	1982/1/13	2008/4/3	Finance Department
20080203	Atwood	MACMILLAN	Male	1986/7/13	2008/4/5	Finance Department
20080204	Barlow	PRIESTLEY	Male	1986/2/21	2008/4/6	Finance Department
20080301	Leonard	ADAMS	Male	1981/11/9	2008/4/3	Purchasing Department
20080302	Tiffany	MORSE	Male	1977/10/1	2008/4/3	Purchasing Department
20080303	Katherine	SHELLEY	Female	1983/10/3	2008/4/4	Purchasing Department
20080401	Nicole	WILDE	Female	1986/5/2	2008/2/1	Production Department
20080402	Pearl	MARSHALL	Female	1982/1/17	2008/3/26	Production Department
20080403	Arlene	TYLER	Female	1986/8/23	2008/3/26	Production Department
20080404	Linda	BAKER	Female	1981/9/13	2008/3/26	Production Department
20080405	Michelle	ROGER	Female	1976/5/13	2008/3/26	Production Department
20080406	Joan	WILLARD	Female	1978/6/4	2008/3/26	Production Department
20080501	Constance	CHAPLIN	Female	1984/7/29	2008/2/1	Sales Department
20080502	Florence	CROFTS	Female	1986/12/22	2008/2/1	Sales Department
20080503	Hedy	JACOB	Female	1979/5/4	2008/2/8	Sales Department
20080601	Jason	CLARE	Male	1978/9/23	2008/3/24	HR Department
20080602	Michael	HU	Male	1985/8/13	2008/3/24	HR Department
20080603	Astrid	KITTO	Female	1980/12/12	2008/3/24	HR Department
20080604	Candice	NOYES	Female	1978/11/7	2008/3/24	HR Department
20080605	Iggy	ZHAO	Female	1982/12/20	2008/3/24	HR Department
20080606	Valerie	WESLEY	Female	1985/3/6	2008/3/24	HR Department
20080607	Dana	HART	Male	1982/4/30	2008/3/24	HR Department
20090205	Sigird	MARJORY	Female	1987/4/6	2009/4/3	Finance Department
20090206	Veronica	MARTHA	Female	1987/6/6	2009/4/3	Finance Department
20090207	[illegible]	ALBERT	Female	1985/5/2	2009/4/3	Finance Department

Record: 149 of 149 No Filter Search

Appendix 2: Employee

Employee

Employee_ID	Current_Position	Location
20080101	01I	A3
20080102	02H	A3
20080103	03G	A3
20080104	04G	A3
20080105	05G	A3
20080106	06G	A3
20080107	07G	A3
20080203	02E	A1
20080204	02A	A1
20080303	03A	A1
20080401	04G	A1
20080403	04A	A1
20080501	05A	A2
20080502	05F	A2
20080503	05A	A2
20080602	06A	A2
20080605	06E	A2
20080606	06A	A2
20090205	02D	A1
20090206	02A	A1
20090207	02A	A1
20090208	02E	A1
20090209	02A	A1
20090211	02A	A1
20090212	02D	A1
20090304	03B	A1
20090305	03A	A1
20090307	03E	A1
20090407	04A	B1
20090408	04D	B1
20090410	04A	B1
20090412	04A	B1
20090414	04A	B1

Record: 116 of 116 No Filter Search

Appendix 3: Retiree

Retiree

Retiree_ID	Final_Position	Retirement_Date
20080604	06A	2011/2/4
20080601	06B	2013/9/2
20080607	06A	2013/12/23
20090411	04B	2014/5/5
20090413	04A	2014/11/7
20080402	04A	2015/9/7
20090306	03C	2015/9/28
20100504	05C	2015/10/2
20080301	02E	2015/10/6
20110310	03F	2015/10/7
20080406	04A	2016/8/15
20080405	04D	2016/9/4
20090213	02A	2016/9/7
20080302	03C	2016/10/1
20140215	02B	2016/10/2
20150511	05C	2016/10/5
20130447	04C	2016/10/7
20090210	02E	2016/11/1
20130433	04C	2016/11/2
20090409	04F	2017/2/15
20120704	07D	2017/3/19
20110419	04B	2017/6/5
20080404	04B	2017/9/3
20080603	06E	2017/9/4
20130439	04B	2017/9/16
20080201	02A	2017/10/1
20080202	02G	2017/12/1
20160315	03B	2017/12/2
20130452	04D	2017/12/2
20170456	04A	2017/12/8
20170460	04A	2018/4/2
20170464	04A	2018/4/2
20170461	04A	2018/4/2

Record: 34 of 34 No Filter Search

Appendix 4: Promotion

Promotion

Promotion_Code	Employee_ID	Current_Position	New_Position	Promotion_Date
1	20080202	02B	02C	2009/9/2
2	20080203	02B	02C	2010/9/2
3	20080401	04B	04C	2010/9/2
4	20080502	05B	05C	2010/9/2
5	20080605	06B	06C	2010/9/2
6	20090208	02B	02C	2010/9/2
7	20080202	02C	02D	2011/9/4
8	20080302	03B	03D	2011/9/4
9	20080401	04C	04D	2011/9/4
10	20080405	04C	04D	2011/9/4
11	20090205	02B	02D	2011/9/4
12	20090210	02C	02E	2011/9/4
13	20090212	02C	02D	2011/9/4
14	20090409	04B	04C	2011/9/4
15	20080202	02D	02E	2012/9/6
16	20080203	02C	02D	2012/9/6
17	20080603	06C	06E	2012/9/6
18	20090307	03C	03D	2012/9/6
19	20090408	04B	04D	2012/9/6
20	20090415	04C	04E	2012/9/6
21	20100504	05B	05C	2012/9/6
22	20100611	06C	06D	2012/9/6
23	20110310	03B	03D	2012/9/6
24	20090208	02C	02D	2013/9/4
25	20090409	04C	04D	2013/9/4
26	20100507	05B	05C	2013/9/4
27	20100609	06B	06D	2013/9/4
28	20110420	04C	04E	2013/9/4
29	20110423	04B	04D	2013/9/4
30	20110428	04C	04D	2013/9/4
31	20120701	07C	07E	2013/9/4
32	20080202	02E	02F	2014/9/5
33	20080401	04D	04E	2014/9/5

Record: 76 of 76 No Filter Search

Appendix 5: Evaluation

Evaluation

Employee_ID	Score_2008	Score_2009	Score_2010	Score_2011	Score_2012	Score_2013	Score_2014	Score_2015	Score_2016	Score_2017	Other_Notes
20080201	77	75	74	77	69	70	76	74	74	72	
20080202	75	82	70	85	87	69	85	71	83	77	
20080203	79	70	78	77	83	78	69	89	79	76	2010:Made huge contribution to a project in
20080204	68	66	73	77	79	76	76	77	77	78	
20080301	69	65	71	72	77	64	75	72			
20080302	72	79	77	75	79	72	77	68	73		
20080303	70	73	72	76	77	71	63	65	72	65	
20080401	71	67	80	86	74	71	82	78	88	71	
20080402	68	78	74	64	68	73	73	54			2015:Lower than 60, fired
20080403	62	72	79	76	79	74	68	66	78	77	
20080404	75	66	69	71	71	69	69	63	76	58	2017:Lower than 60, fired
20080405	67	78	76	87	62	72	77	77	79		
20080406	74	79	72	75	68	76	66	72	78		
20080501	77	77	74	76	70	79	72	69	73	70	
20080502	81	79	83	76	74	72	86	79	83	72	2008:Poor behavioral disclpines
20080503	69	72	77	78	72	69	67	77	73	71	
20080601	79	70	67	74	78	72					
20080602	78	76	74	79	70	63	68	70	70	63	
20080603	71	74	70	79	80	79	70	77	74	71	
20080604	72	68	53								2010:Lower than 60, fired
20080605	72	75	85	79	71	70	89	71	61	76	
20080606	64	65	62	75	69	79	70	74	71	68	
20080607	74	79	70	72	79	48					2013:Lower than 60, fired
20090205		78	75	76	77	77	79	69	73	73	2011:Made great contribution to a project
20090206		74	79	70	69	65	69	75	67	72	
20090207		78	74	70	74	68	71	77	75	77	
20090208		74	87	76	73	87	68	84	72	68	
20090209		73	75	71	70	71	75	66	72	73	
20090210			68	80	63	65	75	74	79		
20090211		69	79	71	78	73	69	79	76	64	
20090212		72	72	81	77	72	73	71	67	63	
20090213			69	74	63	77	72	73	79		
20090304		78	78	79	69	65	72	79	67	79	
20090305		70	69	79	75	70	67	75	69	75	
20090306			76	79	77	66	67	79			
20090307		74	76	70	81	62	69	77	64	61	2015:Gained a major client
20090407		77	72	65	70	69	72	78	77	72	
20090408		66	79	75	83	82	68	73	66	66	2013：Poor relationship with colleagues
20090409			76	80	79	85	74	70	80		
20090410		71	75	71	75	75	71	77	78	79	

Record: 125 of 125 No Filter Search

Appendix 6: Training

Training

Training_Code	Course_Name	Training_Fee	Training_Date
NE00801	New Employee Orientation	€5.00	2008/2/13
NE00802	New Employee Orientation	€5.00	2008/4/5
ME0801	Microsoft Excel Course	€10.00	2008/9/15
NE00901	New Employee Orientation	€5.00	2009/4/9
NE00902	New Employee Orientation	€5.00	2009/4/20
FA0901	First Aid Course	€10.00	2009/7/1
NE00903	New Employee Orientation	€5.00	2009/9/3
BP0901	How to Write A Business Proposal	€20.00	2009/10/31
ME1001	Microsoft Excel Course	€12.00	2010/3/31
NE01001	New Employee Orientation	€5.00	2010/9/20
CV1001	CV Writing	€20.00	2010/10/14
NE01101	New Employee Orientation	€7.00	2011/4/7
NE01102	New Employee Orientation	€7.00	2011/4/24
FA1101	First Aid Course	€15.00	2011/7/1
NE01103	New Employee Orientation	€7.00	2011/9/3
BP1101	How to Write A Business Proposal	€23.00	2011/10/27
NE01201	New Employee Orientation	€7.00	2012/6/20
CV1201	CV Writing	€25.00	2012/9/1
PS1201	Photoshop	€20.00	2012/12/10
ME1301	Microsoft Excel Course	€12.00	2013/4/1
NE01301	New Employee Orientation	€7.00	2013/8/9
NE01302	New Employee Orientation	€7.00	2013/8/23
NE01303	New Employee Orientation	€7.00	2013/9/4
BP1301	How to Write A Business Proposal	€25.00	2013/11/11
NE01401	New Employee Orientation	€10.00	2014/3/9
ME1401	Microsoft Excel Course (1)	€15.00	2014/10/31
ME1402	Microsoft Excel Course (2)	€15.00	2014/11/7
PS1401	Photoshop	€22.00	2014/12/20
NE01501	New Employee Orientation	€10.00	2015/3/22
FA1501	First Aid Course	€20.00	2015/7/2
FM1501	FinancIal Management	€30.00	2015/10/18
NE01601	New Employee Orientation	€10.00	2016/4/12
CV1601	CV Writing	€20.00	2016/10/22

Record: 40 of 40 No Filter Search

Appendix 7: Profiles_Training

Profiles_Training

Training_Code	Employee_ID
BP0901	20080103
BP0901	20080104
BP0901	20080105
BP0901	20080106
BP0901	20090206
BP0901	20090212
BP0901	20090304
BP0901	20090305
BP0901	20090307
BP0901	20090407
BP0901	20090408
BP0901	20090410
BP0901	20090412
BP1301	20100609
BP1301	20100610
BP1301	20100611
BP1301	20110308
BP1301	20110309
BP1301	20110416
BP1301	20110426
BP1301	20110427
BP1301	20110428
BP1301	20110429
BP1301	20110430
BP1301	20120701
BP1301	20120702
BP1301	20120708
BP1301	20130431
BP1301	20130438
BP1301	20130440
BP1301	20130441
BP1301	20130442
BP1301	20130443

Record: 400 of 400 No Filter

References

Al-Maolegi, M. and Arkok, B., 2014. An improved Apriori algorithm for association rules. *arXiv preprint arXiv:1403.3948*.

Hassanat, A.B., Abbadi, M.A., Altarawneh, G.A. and Alhasanat, A.A., 2014. *Solving the Problem of the K Parameter in the KNN Classifier Using an Ensemble Learning Approach.* International Journal of Computer Science and Information Security 12 (8), pp.33-39.

Kotsiantis, S. and Kanellopoulos, D., 2006. Association rules mining: A recent overview. *GESTS International Transactions on Computer Science and Engineering*, *32*(1), pp.71-82.

Kuncheva, L.I., 2004. *Combining pattern classifiers: methods and algorithms*. John Wiley & Sons.

Rao, S. and Gupta, P., 2012. *Implementing Improved Algorithm Over APRIORI Data Mining Association Rule Algorithm 1.*

Sutton O., 2012. *Introduction to k Nearest Neighbour Classification and Condensed Nearest Neighbour Data Reduction.*