Mid Semester

Nzambuli Daniel 665721

2024-02-20

Question 1

*i) Differentiate the three Anomaly detection techniques **(5mks)***

1. **K Nearest Neighbour** (KNN) – technique associates a value to its k nearest neighbor. The points furthest from the neighbor are considered anomalies. `It needs the anomalies to be  isolated from the dense regions.`
2. **Density Based Spatial Clustering of Applications with Noise** (DBSCAN) – Groups together closely packed points into clusters. The points that lie alone are grouped in low density regions and then marked as anomalies. `it needs data to be grouped as maximum distance between points and minimum elements to form a cluster.` **DOES NOT** need prior clusters but **points without prior clusters** are automatically anomalies.
3. **Z - score** uses a normal distribution curve to compare to the highest and lowest value in a distribution. Values over a threshold standard deviation sd `4` are considered anomalies. `Assumes that data is normally distributed`

ii)**A financial institution analyzes customer transactions to identify potential fraud. How could anomaly detection be used in this situation? How would you balance the need for accurate fraud detection with the risk of wrongly flagging legitimate transactions? **(4mks)

In finance, there is a trend to how money gets into a persons account. This can be tracked to indicate either an `over withdrawal of money from an account` or `an excessive increase in the number of deposits to an account` .

Over withdrawal may indicate forced transaction while an excessive increase in the amount deposited can indicate miscellaneous sources of income.

The financial institution can grade people based on the jobs they say they do when opening a bank account. Based on this grouping an individual will tend to be within the normal distribution, or clusters of the already present account holders in the bank with the same job. The people also tend to have similar spending habits based on their income level and money flow needs.

If an individual begins to move towards the extremes of their groups and the move is too rapid a system can be developed to track this as a potential for fraud. To prevent `wrongful flagging` individuals can be required to re-declare their sources of money or a temporary lock can be placed on their account until a proper filling of financial improvement and financial need is filled respectively

Question 2

*i) Using the Groceries dataset, find the top 5 most frequent itemsets with a minimum support threshold of 0.05 using the Apriori algorithm. Provide the itemsets and their corresponding support values. **(3mks)***

Data

```
library(arules)
```

```
## Loading required package: Matrix
```

```
## 
## Attaching package: 'arules'
```

```
## The following objects are masked from 'package:base':
## 
##     abbreviate, write
```

```
data(Groceries)

head(Groceries)
```

```
## transactions in sparse format with
##  6 transactions (rows) and
##  169 items (columns)
```

Apriori Analysis

```
apri_rules = apriori(Groceries, parameter = list(supp = 0.09,target="frequent itemsets"))
```

```
## Apriori
## 
## Parameter specification:
##  confidence minval smax arem  aval originalSupport maxtime support minlen
##          NA    0.1    1 none FALSE            TRUE       5    0.09      1
##  maxlen            target   ext
##      10 frequent itemsets TRUE
## 
## Algorithmic control:
##  filter tree heap memopt load sort verbose
##     0.1 TRUE TRUE  FALSE TRUE    2    TRUE
## 
## Absolute minimum support count: 885 
## 
## set item appearances ...[0 item(s)] done [0.00s].
## set transactions ...[169 item(s), 9835 transaction(s)] done [0.03s].
## sorting and recoding items ... [10 item(s)] done [0.00s].
## creating transaction tree ... done [0.01s].
## checking subsets of size 1 2 done [0.00s].
## sorting transactions ... done [0.00s].
## writing ... [10 set(s)] done [0.00s].
## creating S4 object  ... done [0.00s].
```

```
inspect(apri_rules)
```

```
##      items              support    count
## [1]  {shopping bags}    0.09852567  969
## [2]  {sausage}          0.09395018  924
## [3]  {bottled water}    0.11052364 1087
## [4]  {tropical fruit}   0.10493137 1032
## [5]  {root vegetables}  0.10899847 1072
## [6]  {soda}             0.17437722 1715
## [7]  {yogurt}           0.13950178 1372
## [8]  {rolls/buns}       0.18393493 1809
## [9]  {other vegetables} 0.19349263 1903
## [10] {whole milk}       0.25551601 2513
```

Get the values

```
inspect(head(apri_rules, sort = "support", 10))
```

```
##      items              support    count
## [1]  {shopping bags}    0.09852567  969
## [2]  {sausage}          0.09395018  924
## [3]  {bottled water}    0.11052364 1087
## [4]  {tropical fruit}   0.10493137 1032
## [5]  {root vegetables}  0.10899847 1072
## [6]  {soda}             0.17437722 1715
## [7]  {yogurt}           0.13950178 1372
## [8]  {rolls/buns}       0.18393493 1809
## [9]  {other vegetables} 0.19349263 1903
## [10] {whole milk}       0.25551601 2513
```

ITEMS	SUPPORT
Shopping bags	0.0985
Sausages	0.09395
Bottled Water	0.1105
Tropical Fruit	0.1049
Root Vegetables	0.109
Soda	0.17438
Yogurt	0.1395
rolls/ buns	0.1839
other vegetables	0.1935
whole milk	0.2555

*ii) Using the same Groceries dataset, identify the top 5 most frequent itemsets with a minimum support threshold of 0.09 using the ECLAT algorithm. Compare the results with those from the Apriori algorithm. **(2mks)***

Eclat analysis

```
eclat_rules = eclat(Groceries, parameter = list(supp = 0.09, maxlen = 10, target = "frequent itemse
t"))
```

```
## Eclat
##
## parameter specification:
##  tidLists support minlen maxlen            target  ext
##     FALSE    0.09      1     10 frequent itemsets TRUE
##
## algorithmic control:
```

```
##  sparse sort verbose
##       7   -2    TRUE
##
## Absolute minimum support count: 885
##
## create itemset ...
## set transactions ...[169 item(s), 9835 transaction(s)] done [0.02s].
## sorting and recoding items ... [10 item(s)] done [0.00s].
## creating bit matrix ... [10 row(s), 9835 column(s)] done [0.00s].
## writing  ... [10 set(s)] done [0.00s].
## Creating S4 object  ... done [0.00s].
```

```
inspect(head(eclat_rules, 10))
```

```
##      items              support    count
## [1]  {whole milk}       0.25551601 2513
## [2]  {other vegetables} 0.19349263 1903
## [3]  {rolls/buns}       0.18393493 1809
## [4]  {yogurt}           0.13950178 1372
## [5]  {soda}             0.17437722 1715
## [6]  {root vegetables}  0.10899847 1072
## [7]  {tropical fruit}   0.10493137 1032
## [8]  {bottled water}    0.11052364 1087
## [9]  {sausage}          0.09395018  924
## [10] {shopping bags}    0.09852567  969
```

items	support	count
<chr>	**<dbl>**	**<int>**
{whole milk}	0.25551601	2513
{other vegetables}	0.19349263	1903
{rolls/buns}	0.18393493	1809
{yogurt}	0.13950178	1372
{soda}	0.17437722	1715
{root vegetables}	0.10899847	1072
{tropical fruit}	0.10493137	1032
{bottled water}	0.11052364	1087
{sausage}	0.09395018	924
{shopping bags}	0.09852567	969

Comparing ECLAT and APRIORI

Similarities

- Models indicate that whole milk is the most bought item in most grocery carts
- Models indicate that the least bought item is the shopping bag
- The models also offer a similar support value for all items

iii) *Using the Groceries dataset, find all association rules where* ***soda*** *is part of the LHS. Use support threshold of*

*0.05. & confidence of 0.05. What is the highest support value among these rules? **(3mks)***

```
soda.rules <- apriori(Groceries, parameter = list(supp = 0.05, conf = 0.05, target = "frequent itemset"), appearance = list(lhs = "soda"), control = list(verbose = FALSE))

inspect(sort(soda.rules, by = "support"))
```

```
##      items                   support    count
## [1]  {whole milk}            0.25551601 2513
## [2]  {other vegetables}      0.19349263 1903
## [3]  {rolls/buns}            0.18393493 1809
## [4]  {soda}                  0.17437722 1715
## [5]  {yogurt}                0.13950178 1372
## [6]  {bottled water}         0.11052364 1087
## [7]  {root vegetables}       0.10899847 1072
## [8]  {tropical fruit}        0.10493137 1032
## [9]  {shopping bags}         0.09852567  969
## [10] {sausage}               0.09395018  924
## [11] {pastry}                0.08896797  875
## [12] {citrus fruit}          0.08276563  814
## [13] {bottled beer}          0.08052872  792
## [14] {newspapers}            0.07981698  785
## [15] {canned beer}           0.07768175  764
## [16] {pip fruit}             0.07564820  744
## [17] {fruit/vegetable juice} 0.07229283  711
## [18] {whipped/sour cream}    0.07168277  705
## [19] {brown bread}           0.06487036  638
## [20] {domestic eggs}         0.06344687  624
## [21] {frankfurter}           0.05897306  580
## [22] {margarine}             0.05856634  576
## [23] {coffee}                0.05805796  571
## [24] {pork}                  0.05765125  567
## [25] {butter}                0.05541434  545
## [26] {curd}                  0.05327911  524
## [27] {beef}                  0.05246568  516
## [28] {napkins}               0.05236401  515
```

ANSWER

Whole Milk

iv)** Generate rules from the Groceries dataset where coffee is only in the RHS. List the top 5 rules by their support values. **(4mks)

```
soda.rules <- apriori(Groceries, parameter = list(supp = 0.05, conf = 0.05, target = "frequent itemset"), appearance = list(rhs = "coffee"), control = list(verbose = FALSE))

inspect(head(sort(soda.rules, by = "support"), 5))
```

```
##     items              support   count
## [1] {whole milk}       0.2555160 2513
## [2] {other vegetables} 0.1934926 1903
## [3] {rolls/buns}       0.1839349 1809
## [4] {soda}             0.1743772 1715
## [5] {yogurt}           0.1395018 1372
```

Question 3

*i) Using the airquality dataset, use the Z-score method to detect outliers in the Ozone column. Consider observations with a Z-score greater than 2 or less than -2 as outliers. How many outliers are there? **(3mks)***

```
data(airquality)
head(airquality)
```

```
##   Ozone Solar.R Wind Temp Month Day
## 1    41     190  7.4   67     5   1
## 2    36     118  8.0   72     5   2
## 3    12     149 12.6   74     5   3
## 4    18     313 11.5   62     5   4
## 5    NA      NA 14.3   56     5   5
## 6    28      NA 14.9   66     5   6
```

```
d = na.omit(airquality)

d$z_score = abs((d$Ozone - mean(d$Ozone))/ sd(d$Ozone))
d$outliers = d$z_score > 2
d$count = rep(1, nrow(d))

library(dplyr)
```

```
##
## Attaching package: 'dplyr'
```

```
## The following objects are masked from 'package:arules':
##
##     intersect, recode, setdiff, setequal, union
```

```
## The following objects are masked from 'package:stats':
##
##     filter, lag
```

```
## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
```

```
data = d %>% group_by(outliers) %>% summarise(count = sum(count))
data
```

```
## # A tibble: 2 × 2
##   outliers count
##   <lgl>    <dbl>
## 1 FALSE      105
## 2 TRUE         6
```

There are `6` outliers in OZONE

ii) *Discuss the role of a data warehouse in supporting the data mining process for a global retail chain. How can data warehousing facilitate the extraction of actionable business intelligence? Consider Data Integration, Historical Data Analysis, Data Quality Scalability and Performance in your answer* ***(6mks)***

Data warehouse stores a collection of `historical data` about the retail chain. This is done through the data enterprise that can track regional sales and orders. The virtual enterprise that can track the individual customers at each local shop and detect repeat acquisitions and the combinations of repeated acquisitions

Data warehouses have a MOLAP (Multi-dimensional OLAP) system that is able to track non relational data. This tends to be the high frequency data that is generated on a daily even hourly basis by the regional business. They also have a ROLAP(Relational OLAP) system that uses relational databases to increase the efficiency of data analysis by documenting the way different stores are related. This is also the permanent long-term storage that allows the whole organization to have a central information store. This coupled with a HOLAP(Hybrid OLAP) which is able to allow `data integration` between the two systems.

Following a uniform development system data extraction, loading and transformation protocol ensures that data is consistent in how it is stored across all the departments of the. Consistency of the data ensures that the `data quality` is always high. A uniform system allows for templetized development of data infrastructure making `data scalability easier`

All these benefits ensure that the decisions the organization makes are informed and actionable as there is enough data to guide decisions, the data is of high quality, the data collection systems are highly integrated and the systems can be scaled across the whole international organization.