CPC351 – PRINCIPLES OF DATA ANALYTICS

Project – Data Analytics

Lecturer:

Assoc. Prof. Dr. Wong Li Pei

Submission Date:

31 January 2022

Name	Matric No
Sharvin A/L Kogilavanan	148056
Thanish A/L Natarajan	149156
Katheeravan A/L Balasubramaniam	147744

Road Traffic Accidents Dataset

Problem Statement

Road accidents are a major concern that not many people are aware of. We picked this dataset in order to get a better understanding about the casualties involved in an accident based on thier age, sex and type of people. Besides that, this dataset also has data like road class, weather and more which will also help us to get a better understanding of the number of casualties as well. Understanding this will have us to classify an accident into slight, serious and fatal more accurately as this would help in saving people's lives.

Objective

1) Determine casualty severity of an accident.

Hypothesis

1) Casualty class will have a direct impact on casualty severity.
2) Road class and type of vehicle have a direct impact on severity of casualty.

Data Preprocessing

We have performed several operations to preprocess the data, starting with data cleaning to data transformation. For data cleaning, we have removed unnecessary columns such as reference number, coordinates of location, accident date, first road class number, local authority and vehicle number. This is because none of them may not have a direct impact on the severity of casualties. Then, we have also renamed the variable names to more appropriate one for better clarity. For data transformation, we have transformed the categorical variables that are found as characters in the dataset into factors. Finally, we created two new variables which are age_group and time_fix because both of them were numerical types. So, we wanted to change these values into categorical to get a better understanding when analysing the data.

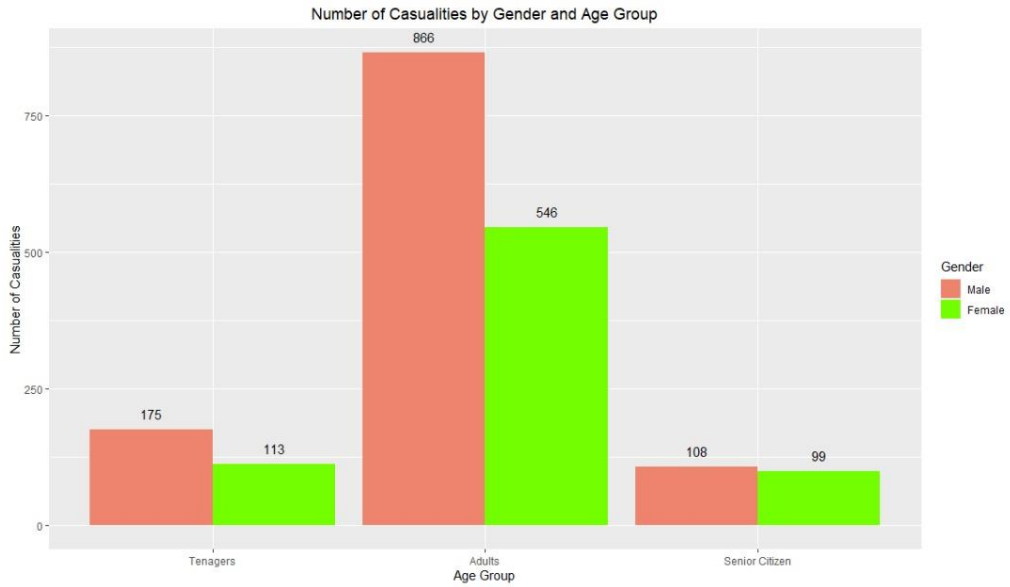


Figure 1.1 shows bar plot of Number of Casualties by Gender and Age Group

Based on figure 1.1, we can see that adult males are the most number of people who are involved in accidents. Senior citizens are the least involved in accidents compared to other group ages as well as genders. This is due to the fact that most drivers will be adults.

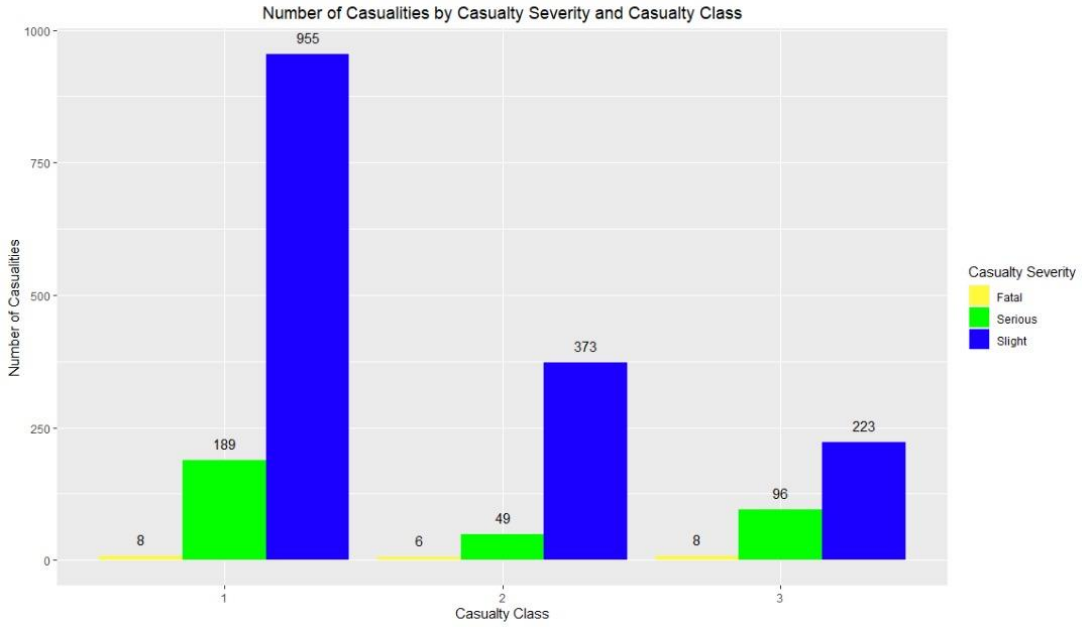


Figure 1.4 shows bar plot of Number of Casualties by Gender and Casualty Class

Based on figure 1.4, drivers or riders are the most number of people who are involved in slight accidents. Same number of drivers and pedestrians are involved in fatal accidents, which is 8 people. Passengers are the least number of people who are involved in serious accidents.

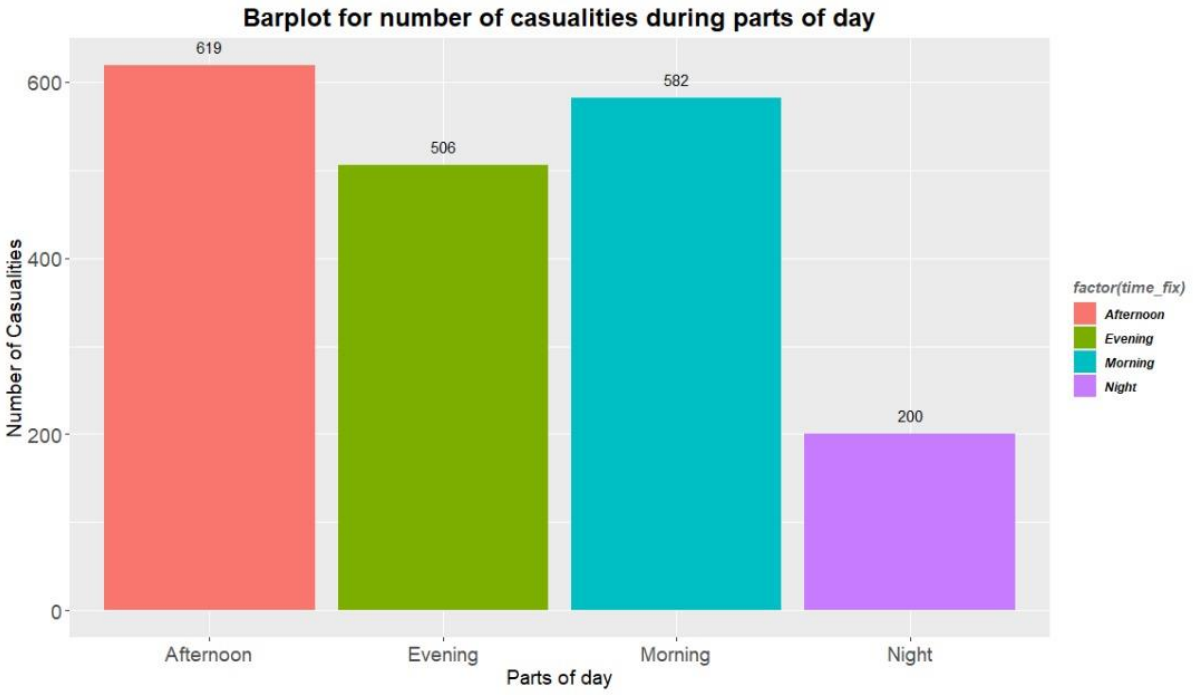


Figure 1.5 shows bar plot of number of casualties during parts of the day

Based on figure 1.5, the most casualties are during Afternoon. This may be due to the fact that some roads will be congested or some people will be driving fast to their office after lunch. During night time very least number of accidents occur so the number of casualties is less.

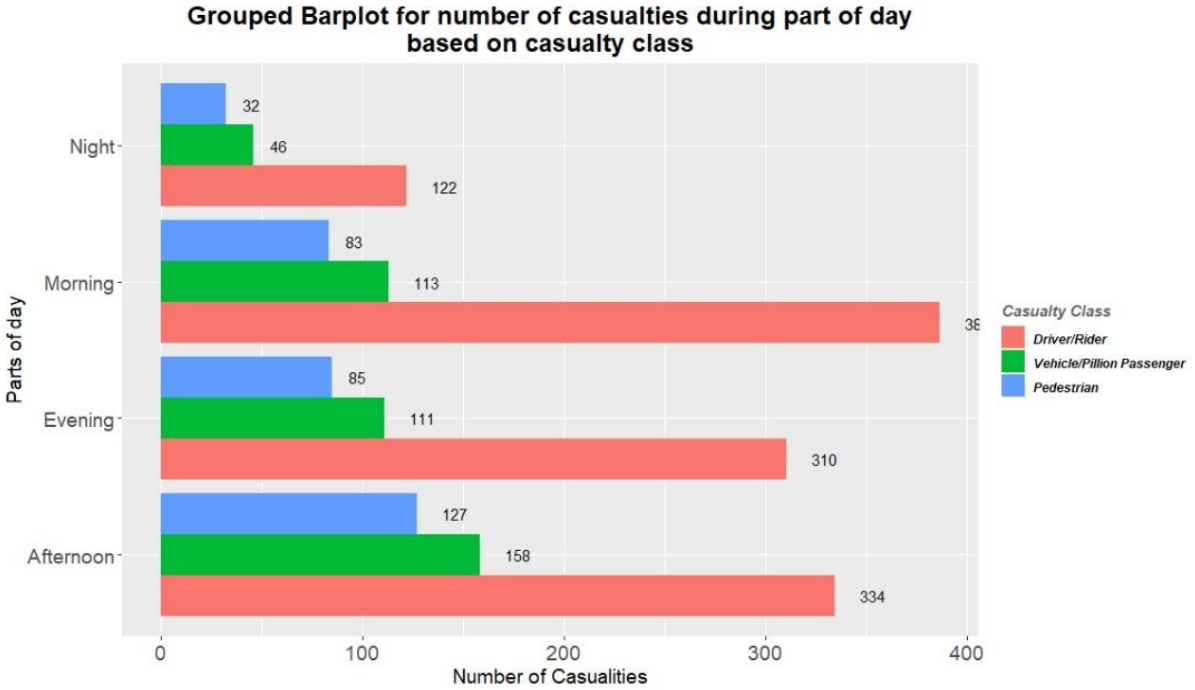


Figure 1.6 shows grouped bar plot for number of casualties during part of the day based on causality class.

Based on figure 1.6, we can say that the number of casualties who are drivers are at the highest during morning. This is because most drivers would be rushing to the office for their work. The number of casualties who are passengers and pedestrians are at the lowest during night.

Model Explanation

In order to predict the casualty severity, we planned to build a classification model using a naive bayes algorithm. The reason why we picked naive bayes algorithm is it can perform better with categorical data. Most of our features are categorical data. We found that all the features we had after cleaning the data is enough to build the naive bayes model. Thus we did not perform any feature selection. Then, we have portioned our data into train and test data by allocating 80% to training and 20% for testing. After that we build our model in naive bayes using the train data and the prediction result is as below.

```
Confusion Matrix and Statistics

          Reference
Prediction   1   2   3
         1   0   1   7
         2   2   5  80
         3   1  11 413

Overall Statistics

               Accuracy : 0.8038
                 95% CI : (0.7671, 0.8371)
    No Information Rate : 0.9615
    P-Value [Acc > NIR] : 1

                  Kappa : 0.0595

 Mcnemar's Test P-Value : 2.385e-12

Statistics by Class:

                     Class: 1 Class: 2 Class: 3
Sensitivity          0.000000 0.294118  0.82600
Specificity          0.984526 0.836978  0.40000
Pos Pred Value       0.000000 0.057471  0.97176
Neg Pred Value       0.994141 0.972286  0.08421
Prevalence           0.005769 0.032692  0.96154
Detection Rate       0.000000 0.009615  0.79423
Detection Prevalence 0.015385 0.167308  0.81731
Balanced Accuracy    0.492263 0.565548  0.61300
```

Figure 1.7 shows the naive bayes model results

Based on figure 1.7, we can see that the model is not able to classify the class severity fatal correctly. All the data of class fatal was miss classified. Secondly, the modal is only able to classify 5 sample dataset with class serious correctly. However, it classified 82 samples incorrectly. Finally, the modal classified most of the severity class slight correctly. Only 12 of them were miss classified. The modal was not able to generalize well for class fatal and class serious.

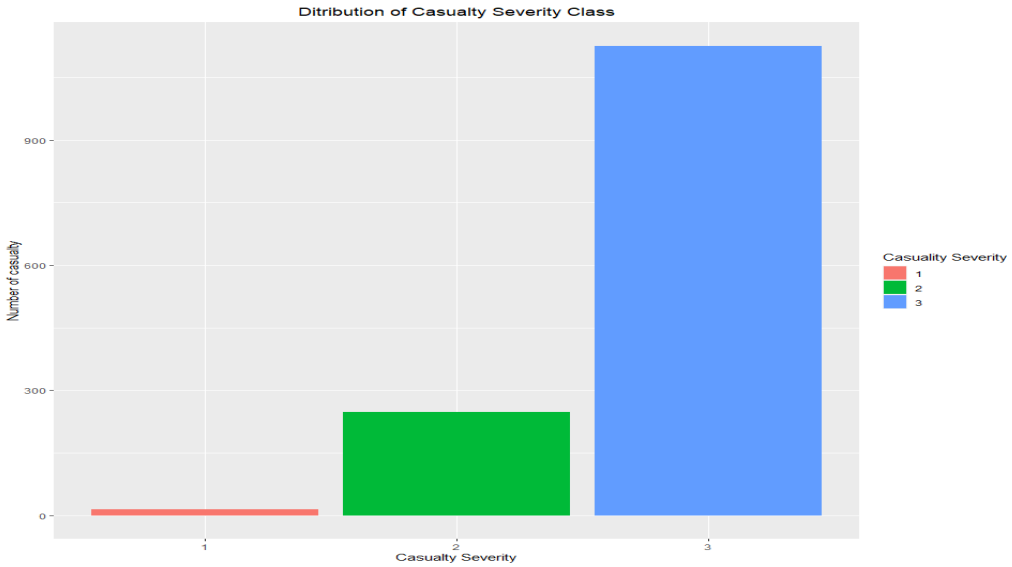


Figure 1.8

Based on the figure 1.8, we can see that the data is imbalanced. The imbalance distribution of the data has severely affected the model to generalize well with the casualty severity type of class slightly because it has more data samples in the dataset compared to the other two classes. There are some

ways to overcome this issue and one of them is by performing subsampling. We have implemented an oversampling method to allow the data for class serious and fatal to be aligned with class slight. However the model performed even worse after subsampling.

H&M Sales Dataset

Problem Statement

When a customer buys a product, they might buy another product as well. If that particular product is not available customers would be disappointed and sales will be affected. So, there must be a way to determine all possible products a customer would buy when they buy a product that must be known by the company in order to increase sales.

Objective

1) Determine possible products will be purchased by a customer based on their current purchase.
2) Finding most frequently bought products and cross selling it with the least bought products to increase the company sales.

Hypothesis

1) Discount and sales will have a direct impact on profit.

Data Preprocessing

We have performed several operations to preprocess the data, starting with data cleaning to data transformation. For data cleaning, we have removed unnecessary columns such as Ship Mode, Country, City, State and Product ID. This is because none of them may not have a direct impact on the Sub Category. Then, we have also renamed the variable names to more appropriate one for better clarity. For data transformation, we have transformed the categorical variables that are found as characters in the dataset into factors to get a better understanding when analysing the data.

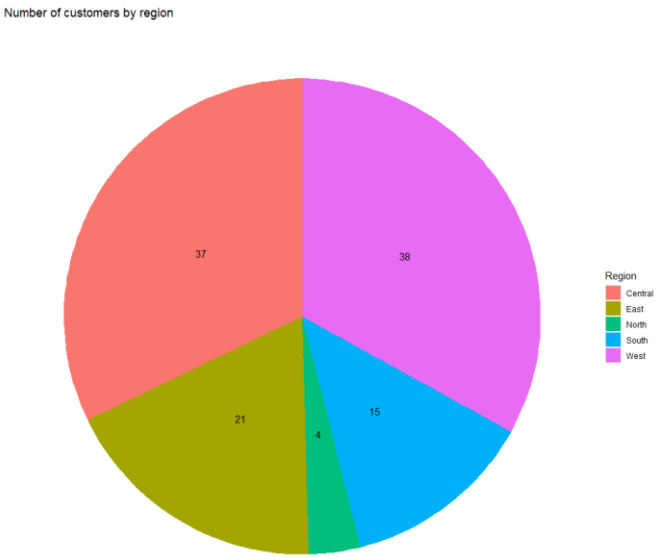


Figure 2.0 shows the pie chart for number of customers in different regions

Based on Figure 2.0, we can see that H&M has a larger customer base in the Central region followed by the West Region. This shows that more people from the Central region are interested to buy products in H&M and H&M may open more outlets in Central region of United States to attract more customers.

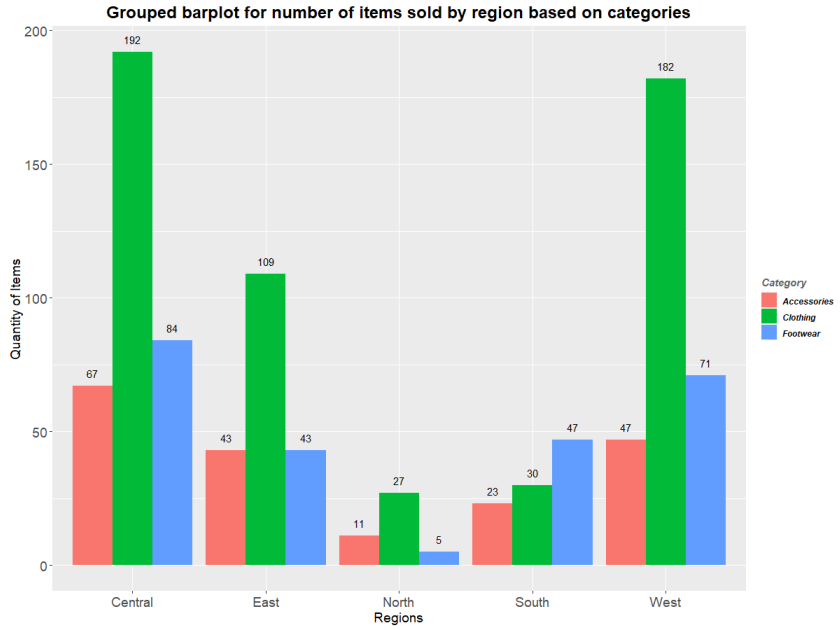


Figure 2.1 shows the grouped barplot for number of items sold by region based on categories

Based on the figure 2.1 it can be seen that apparels from clothing categories were sold more than other categories in the Central, East, North and West region. This shows that many people from these regions are attracted by the clothing items sold by H&M. Hence, H&M need to make sure they have enough stock for clothing items in these regions. It can be also seen that there are more items sold in all categories in the central region which is around 343 items.

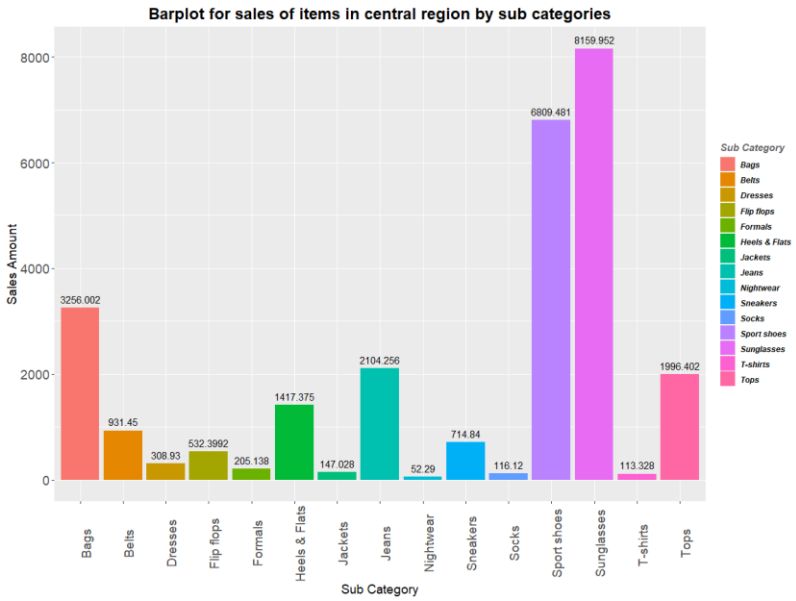


Figure 2.2 shows the barplot for sales of items in central region by sub categories

Previously in figure 2.0 and figure 2.1 it can be seen that the central region has a bigger customer base and more items sold compared to the other regions. From figure 2.2 it can be seen that customers from central regions are not only attracted to clothing items from H&M but also fancy items such as sport shoes and sunglasses. These two apparel sub categories have generated more sales compared to the other items.

For the road traffic dataset, we decided to implement classification. The algorithm that we chose was a naive bayes algorithm. First we divided the data into train and test data where 80% of data will be set for training and the remaining 20% will be set for testing. Next we wanted to check whether the graph is balanced or not.

HnM Dataset visual representation

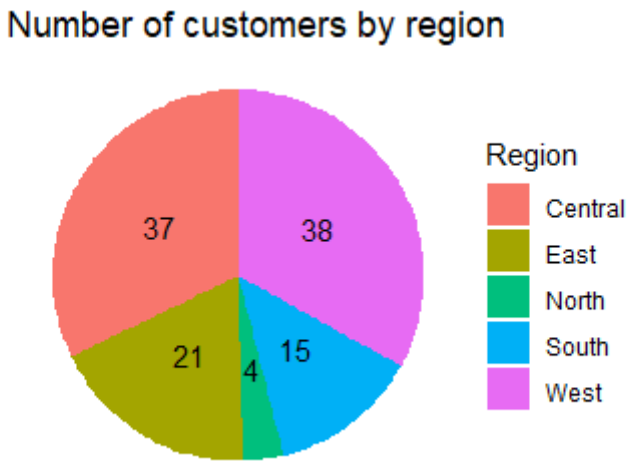


Figure 3.1 shows the number of customers by region

The figure 3.1 shows the number of customers of H&M by region. We can see that a big portion of customers are from the West and Central region. The difference between both regions is only 1. We also can see that there are only 4 customers in the North region.

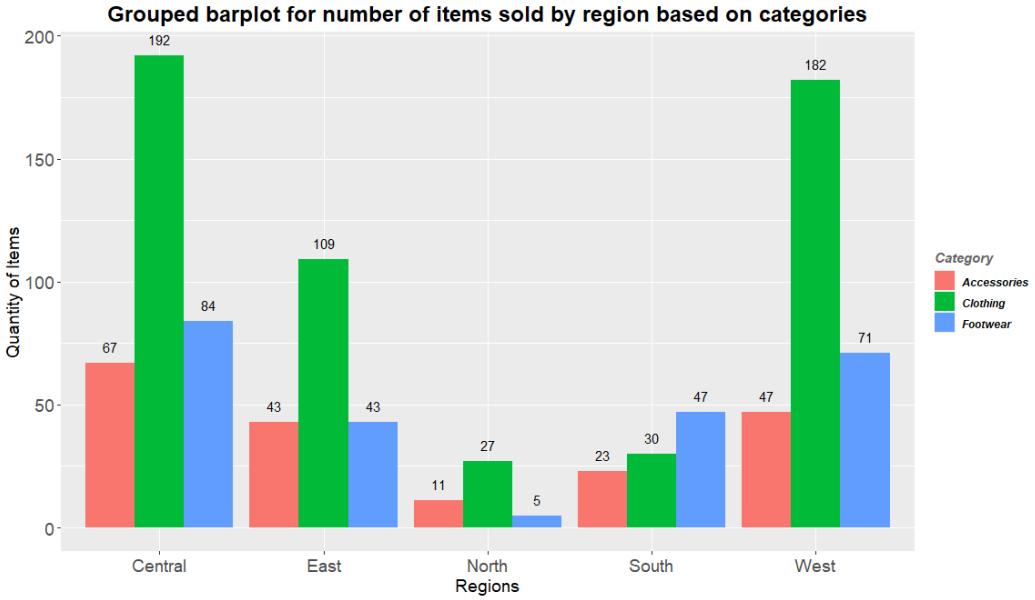


Figure 3.2 shows the number of items sold by region based on categories.

Figure 3.2 shows that the clothing category is the highest sold item in all regions except South. We also can see that the Central region has the highest sold items of all categories compared to other respective categories in other regions. The North region has the lowest sold items of all categories compared to other respective categories in other regions.

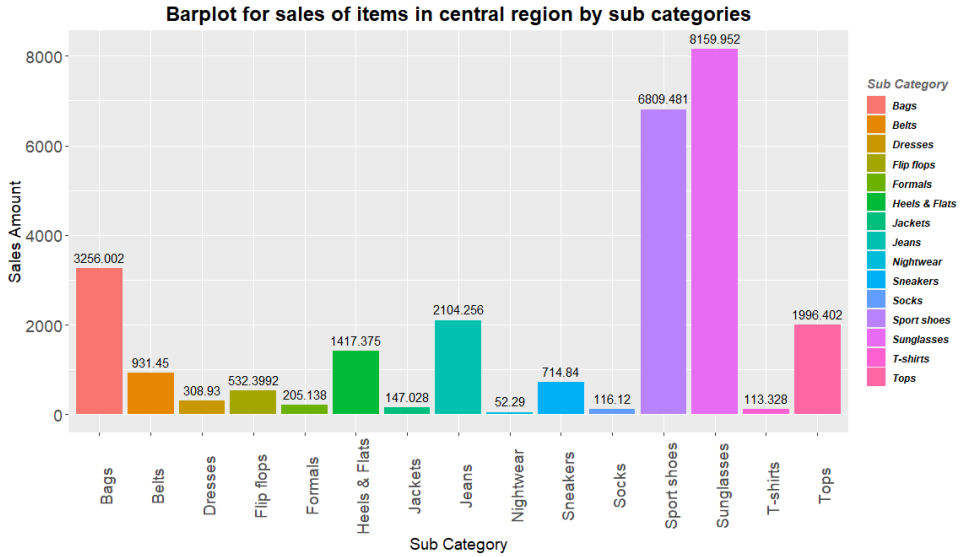


Figure 3.3 shows the sales items in the central region by sub category.

Figure 3.3 shows that the sub category item which is sunglasses is the highest saled amount collected in the Central region which is 8159.95 dollars.. The sports shoes have the second highest amount collected. The lowest sales amount of sub category is nightwear with the value of 52.29 dollars.

Model for H&M using Association rule.

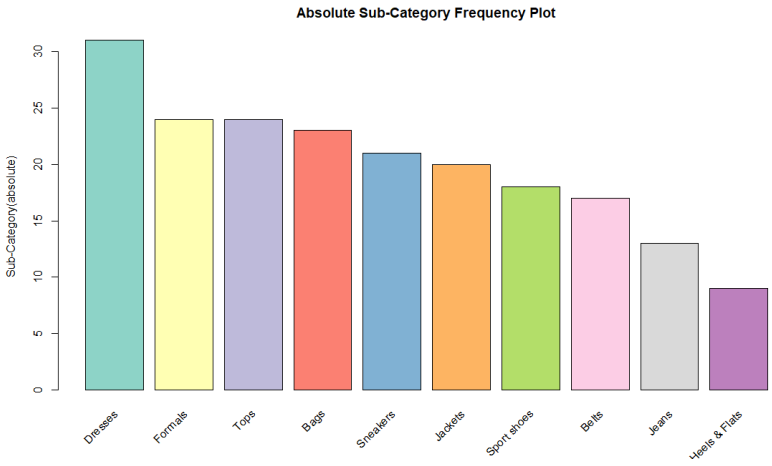


Figure 3.4 shows that the frequency of top 10 sub categories

```
> summary(tr)
transactions as itemMatrix in sparse format with
 116 rows (elements/itemsets/transactions) and
 17 columns (items) and a density of 0.11714

most frequent items:
 Dresses  Formals     Tops     Bags Sneakers  (Other)
      31       24       24       23       21      108

element (itemset/transaction) length distribution:
sizes
 1  2  3  4  5  6
59 29 12  7  4  5

   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
  1.000   1.000   1.000   1.991   2.000   6.000

includes extended item information - examples:
   labels
1    Bags
2   Belts
3 Dresses
>
```

Figure 3.5 shows summary of transactions

Figure 3.4 shows that the dresses sub category has the highest frequency than other sub categories. This frequency is achieved by having grouping order date, order ID and customer ID. This will merge the sub categories which have the same transactions. After that, the columns used to merge the sub categories will be dropped since they will not be used again. The sub categories datasets will be written in new csv and get read again. Figure 3.5 shows the summary of new dataset that has been read. The element length distribution shows that the dresses sub category had 59 transactions. Then we use apriori function algorithm to build frequent sub categories set.

<u>Apriori rule 1 output:</u>

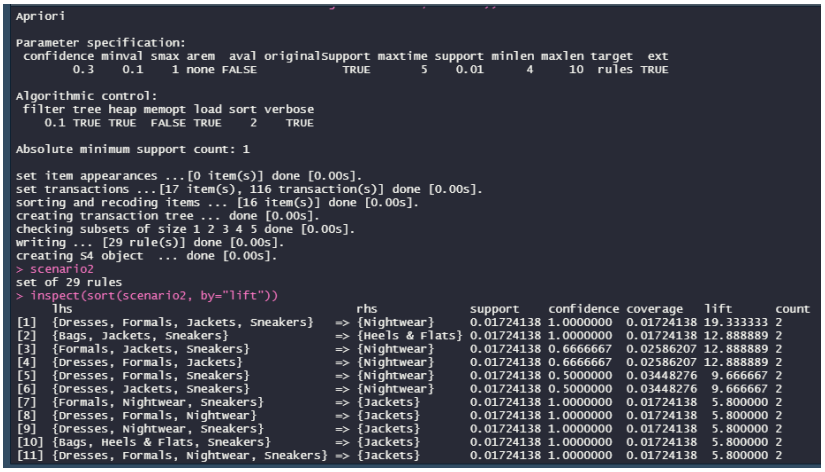
```
Apriori

Parameter specification:
 confidence minval smax arem  aval originalSupport maxtime support minlen maxlen target  ext
        0.5    0.1    1 none FALSE            TRUE       5    0.05      2     10  rules TRUE

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

Absolute minimum support count: 5

set item appearances ...[0 item(s)] done [0.00s].
set transactions ...[17 item(s), 116 transaction(s)] done [0.00s].
sorting and recoding items ... [13 item(s)] done [0.00s].
creating transaction tree ... done [0.00s].
checking subsets of size 1 2 3 done [0.00s].
writing ... [1 rule(s)] done [0.00s].
creating S4 object  ... done [0.00s].
> inspect(scenario1)
    lhs           rhs       support    confidence coverage  lift     count
[1] {Sneakers} => {Dresses} 0.09482759 0.5238095  0.1810345 1.960061 11
> scenario1
set of 1 rules
>
```

Figure 3.6 shows rule 1 output.

We use support of 0.05, confidence of 0.5 with minimum length of 2. The set of rules outputted is only 1 which is dresses bought together with sneakers. It is found in 9% of transactions and have 50% of

confidence that customers buy these together and have a lift more than 1 which means its a meaningful rule.

Apriori rule 2 output:

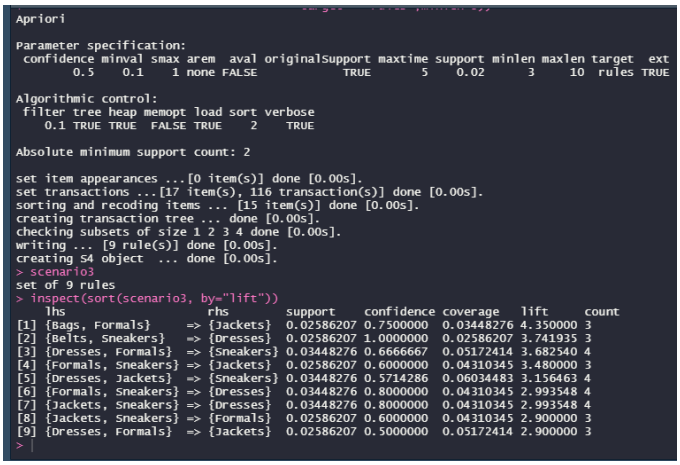


```
Apriori

Parameter specification:
 confidence minval smax arem  aval originalSupport maxtime support minlen maxlen target  ext
        0.3    0.1    1 none FALSE            TRUE       5    0.01      4     10  rules TRUE

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

Absolute minimum support count: 1

set item appearances ...[0 item(s)] done [0.00s].
set transactions ...[17 item(s), 116 transaction(s)] done [0.00s].
sorting and recoding items ... [16 item(s)] done [0.00s].
creating transaction tree ... done [0.00s].
checking subsets of size 1 2 3 4 5 done [0.00s].
writing ... [29 rule(s)] done [0.00s].
creating S4 object  ... done [0.00s].
> scenario2
set of 29 rules
> inspect(sort(scenario2, by="lift"))
     lhs                                  rhs             support    confidence coverage   lift      count
[1]  {Dresses, Formals, Jackets, Sneakers} => {Nightwear}     0.01724138 1.0000000  0.01724138 19.333333 2
[2]  {Bags, Jackets, Sneakers}             => {Heels & Flats} 0.01724138 1.0000000  0.01724138 12.888889 2
[3]  {Formals, Jackets, Sneakers}          => {Nightwear}     0.01724138 0.6666667  0.02586207 12.888889 2
[4]  {Dresses, Formals, Jackets}           => {Nightwear}     0.01724138 0.6666667  0.02586207 12.888889 2
[5]  {Dresses, Formals, Sneakers}          => {Nightwear}     0.01724138 0.5000000  0.03448276  9.666667 2
[6]  {Dresses, Jackets, Sneakers}          => {Nightwear}     0.01724138 0.5000000  0.03448276  9.666667 2
[7]  {Formals, Nightwear, Sneakers}        => {Jackets}       0.01724138 1.0000000  0.01724138  5.800000 2
[8]  {Dresses, Formals, Nightwear}         => {Jackets}       0.01724138 1.0000000  0.01724138  5.800000 2
[9]  {Dresses, Nightwear, Sneakers}        => {Jackets}       0.01724138 1.0000000  0.01724138  5.800000 2
[10] {Bags, Heels & Flats, Sneakers}       => {Jackets}       0.01724138 1.0000000  0.01724138  5.800000 2
[11] {Dresses, Formals, Nightwear, Sneakers} => {Jackets}     0.01724138 1.0000000  0.01724138  5.800000 2
```

Figure 3.7 shows rule 2 output

We use support of 0.01, confidence of 0.3 with minimum length of 4. The set of rules outputted is 29 and ordered according to descending order of lift. The top 4 outputs have the lift more than 10 which means the items bought together occur more often. It is known that a customer who bought dresses, formals, jackets and sneakers also buys sneakers. This kind of transaction occurs 1.7% only and has 100% confidence.

Apriori rule 3 output:



```
Apriori

Parameter specification:
 confidence minval smax arem  aval originalSupport maxtime support minlen maxlen target  ext
        0.5    0.1    1 none FALSE            TRUE       5    0.02      3     10  rules TRUE

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

Absolute minimum support count: 2

set item appearances ...[0 item(s)] done [0.00s].
set transactions ...[17 item(s), 116 transaction(s)] done [0.00s].
sorting and recoding items ... [15 item(s)] done [0.00s].
creating transaction tree ... done [0.00s].
checking subsets of size 1 2 3 4 done [0.00s].
writing ... [9 rule(s)] done [0.00s].
creating S4 object  ... done [0.00s].
> scenario3
set of 9 rules
> inspect(sort(scenario3, by="lift"))
    lhs                   rhs          support    confidence coverage   lift     count
[1] {Bags, Formals}    => {Jackets}   0.02586207 0.7500000  0.03448276 4.350000 3
[2] {Belts, Sneakers}  => {Dresses}   0.02586207 1.0000000  0.02586207 3.741935 3
[3] {Dresses, Formals} => {Sneakers}  0.03448276 0.6666667  0.05172414 3.682540 4
[4] {Formals, Sneakers} => {Jackets}  0.02586207 0.6000000  0.04310345 3.480000 3
[5] {Dresses, Jackets} => {Sneakers}  0.03448276 0.5714286  0.06034483 3.156463 4
[6] {Formals, Sneakers} => {Dresses}  0.03448276 0.8000000  0.04310345 2.993548 4
[7] {Jackets, Sneakers} => {Dresses}  0.03448276 0.8000000  0.04310345 2.993548 4
[8] {Jackets, Sneakers} => {Formals}  0.02586207 0.6000000  0.04310345 2.900000 3
[9] {Dresses, Formals} => {Jackets}   0.02586207 0.5000000  0.05172414 2.900000 3
>
```

Figure 3.8 shows rule 3 of output

We use support of 0.02, confidence of 0.5 with minimum length of3. The set of rules outputted is 9 and ordered according to descending order of lift. This shows that the customers buying bag and formals also buying jackets having the highest lift. Other than that, we can take note that customers who buying dresses and formals, are also buying sneakers. This has 3.4% support, 66% confidence and lift of 3.68 which is still a good frequency of occurrence. This is also one of the highest count of purchases among the 9 set of rules.

From these 3 rules we can predict that most customers who are buying dresses, are also buying sneakers. This sneaker sub category can be cross selled with customers to increase its sales.

Apriori prediction.

```
Apriori

Parameter specification:
 confidence minval smax arem  aval originalSupport maxtime support minlen maxlen target  ext
        0.2    0.1    1 none FALSE            TRUE       5    0.05      2     10  rules TRUE

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

Absolute minimum support count: 5

set item appearances ...[1 item(s)] done [0.00s].
set transactions ...[17 item(s), 116 transaction(s)] done [0.00s].
sorting and recoding items ... [13 item(s)] done [0.00s].
creating transaction tree ... done [0.00s].
checking subsets of size 1 2 done [0.00s].
writing ... [3 rule(s)] done [0.00s].
creating S4 object  ... done [0.00s].
> inspect(sort(scenario4, by="lift"))
    lhs          rhs        support    confidence coverage  lift     count
[1] {Dresses} => {Sneakers} 0.09482759 0.3548387  0.2672414 1.960061 11
[2] {Dresses} => {Belts}    0.06034483 0.2258065  0.2672414 1.540797  7
[3] {Dresses} => {Jackets}  0.06034483 0.2258065  0.2672414 1.309677  7
>
```

Figure 3.9 shows prediction output

We set the left hand side with the default sub category of dresses and predict the item set sold on the right hand side. We can clearly see that the sneakers have a higher lift which makes good meaning, and have a 35% confidence with 9% transaction happening from overall transaction. It is also the most selling sub category that has been sold together with dresses.

Eclat algorithm

We want to suggest another alternative algorithm that can replace the apriori algorithm in doing association rules for this dataset. This is because it is suitable for medium datasets like H&M dataset. It is also faster than the Apriori algorithm. Below are the outputs of 3 rules created with the same support and confidence and minimum length of Apriori algorithm above. Both have the same outputs given.

```
> inspect(sort(rules_1,by="lift"))
    lhs          rhs        support    confidence lift     itemset
[1] {Sneakers} => {Dresses} 0.09482759 0.5238095  1.960061 4
> inspect(sort(rules_2,by="lift"))
    lhs                                rhs                 support    confidence lift      itemset
[1]  {Dresses, Formals, Jackets, Sneakers}   => {Nightwear}     0.01724138 1.0000000  19.333333 1
[2]  {Formals, Jackets, Sneakers}            => {Nightwear}     0.01724138 0.6666667  12.888889 3
[3]  {Dresses, Formals, Jackets}             => {Nightwear}     0.01724138 0.6666667  12.888889 4
[4]  {Bags, Jackets, Sneakers}               => {Heels & Flats} 0.01724138 1.0000000  12.888889 6
[5]  {Dresses, Formals, Sneakers}            => {Nightwear}     0.01724138 0.5000000   9.666667 2
[6]  {Dresses, Jackets, Sneakers}            => {Nightwear}     0.01724138 0.5000000   9.666667 5
[7]  {Dresses, Formals, Nightwear, Sneakers} => {Jackets}       0.01724138 1.0000000   5.800000 1
[8]  {Formals, Nightwear, Sneakers}          => {Jackets}       0.01724138 1.0000000   5.800000 3
[9]  {Dresses, Formals, Nightwear}           => {Jackets}       0.01724138 1.0000000   5.800000 4
[10] {Dresses, Nightwear, Sneakers}          => {Jackets}       0.01724138 1.0000000   5.800000 5
[11] {Bags, Heels & Flats, Sneakers}         => {Jackets}       0.01724138 1.0000000   5.800000 6
[12] {Dresses, Formals, Jackets, Nightwear}  => {Sneakers}      0.01724138 1.0000000   5.523810 1
[13] {Dresses, Formals, Nightwear}           => {Sneakers}      0.01724138 1.0000000   5.523810 2
[14] {Formals, Jackets, Nightwear}           => {Sneakers}      0.01724138 1.0000000   5.523810 3
```

```
> inspect(sort(rules_3,by="lift"))
    lhs                rhs          support    confidence lift     itemset
[1] {Bags, Formals}    => {Jackets}  0.02586207 0.7500000  4.350000 5
[2] {Belts, Sneakers}  => {Dresses}  0.02586207 1.0000000  3.741935 1
[3] {Dresses, Formals} => {Sneakers} 0.03448276 0.6666667  3.682540 2
[4] {Formals, Sneakers} => {Jackets} 0.02586207 0.6000000  3.480000 3
[5] {Dresses, Jackets} => {Sneakers} 0.03448276 0.5714286  3.156463 6
[6] {Formals, Sneakers} => {Dresses} 0.03448276 0.8000000  2.993548 2
[7] {Jackets, Sneakers} => {Dresses} 0.03448276 0.8000000  2.993548 6
[8] {Dresses, Formals} => {Jackets}  0.02586207 0.5000000  2.900000 4
[9] {Jackets, Sneakers} => {Formals} 0.02586207 0.6000000  2.900000 3
>
```

Figure 4.0 show outputs of 3 rules of Eclat algorithm

Problems and Pitfalls

We faced some problems and pitfalls during the project. First problem we faced was when choosing the best model for classification for road accident dataset. We used a decision tree algorithm at first and it takes a very long time to fit the model with training data. Since it requires more time, we cannot focus on other projects. So we decided to change the algorithm and choose the Naive Bayes algorithm.

Another problem we faced is when finding the correct association rule to get the best result. We tried many tuning for the rules and we do not know if the output is good or not. We had to do some research on it and it dragged on for some time. In the end, we believe we are able to come up with the optimized rules for association rules analysis.

References

1. Michael Hahsler [aut, cre. (2022, January 10). *Eclat: Mining Associations with Eclat in Arules: Mining Association rules and frequent itemsets*. eclat: Mining Associations with Eclat in arules: Mining Association Rules and Frequent Itemsets. Retrieved January 31, 2022, from https://rdrr.io/cran/arules/man/eclat.html
2. *Visualize market basket analysis in R*. DataScience+. (n.d.). Retrieved January 31, 2022, from https://datascienceplus.com/visualize-market-basket-analysis-in-r/