Recommendation System with Random Forest and Decision Tree using R

NYTEX Programming

Written in: 20 Feb 2021

Contents

1.0 Introduction

In this part of the report, a few algorithms techniques will be implement into the consumer dataset to predict and recommend the suitable product to the consumer. There are three data science techniques that will be performed in this report, which are Decision Tree with Classification and Random Forest.

The report will start with Data description, then several data preparation process will be carried out. In the end, three machine learning model which are Decision Tree with classification and Random Forest will be built and their performance will be evaluate.

2.0 Data Description

The dataset consists of 9 attributes which states the characteristics of each sales transaction. However, not all of the attribute will be useful in building model. The description of each attribute and whether they are selected to be used in model building is shown in the below table.

Attribute Name	Type	Use in Model Building
event_time	String	No
event_type	String	No
product_id	Double/Numeric	Yes
category_id	Double/Numeric	No
category_code	String	Yes
brand	String	Yes
price	Double/Numeric	Yes
user_id	Double/Numeric	Yes
user_session	String	No

Table 1 : The Selection and Characteristics of Attributes in Dataset

The attribute that are selected in the model building are ‘product_id’, ‘categpry_code’, ‘brand’, ‘price’ and ‘user_id’. The ‘event_time’ consists of the transaction time for each of the sales, however this attribute is too complex and will be difficult for model building. The ‘event_type’ only consists of low information which most of the value inside is ‘view’, since the information provided from this column is limited, the ‘event_type’ is also not been used in model building.

Besides, the ‘category_code’ is linked together with ‘category_id’, so basically they provide the same information about the transaction. Since ‘category_code’ and ‘category_id’ provide the same information, only one of them will be choose in the model building, in this case is the ‘category_code’.

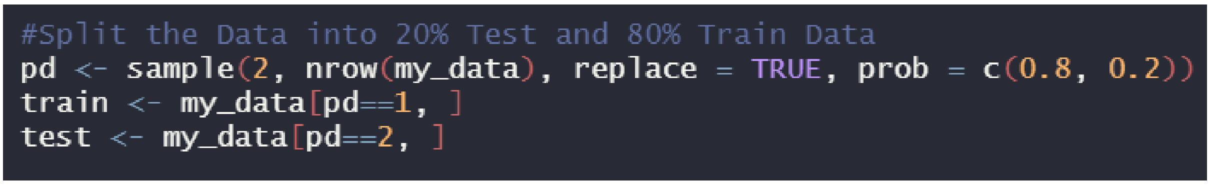
3.0 Data Preparation

To clean the data and ensure that the dataset is ready to be used, a few data cleaning process have been carried out as below:

- Factorization of Double attribute such as user id and product id
- Clean the NULL value of the dataset
- Modify the NA value from category code and brand into ‘others’
- Split data into test and train data

Since both product id and user id attributes are in numeric form, it have to be convert into factor in R so that it can be used in the classification process. To deal with this, the ‘factor’ function have been used on these two attributes to convert it into

Besides, to clean the dataset, the ‘na.omit’ function have been performed in R to clean the null value from the dataset. The ‘NA’ value in the attribute category code and brand have also been modified into ‘others’ for better prediction. Lastly, the dataset have been split into the train and test dataset by a ratio of 80% of train data and 20% of test data by using the below algorithms.



```
#Split the Data into 20% Test and 80% Train Data
pd <- sample(2, nrow(my_data), replace = TRUE, prob = c(0.8, 0.2))
train <- my_data[pd==1, ]
test <- my_data[pd==2, ]
```

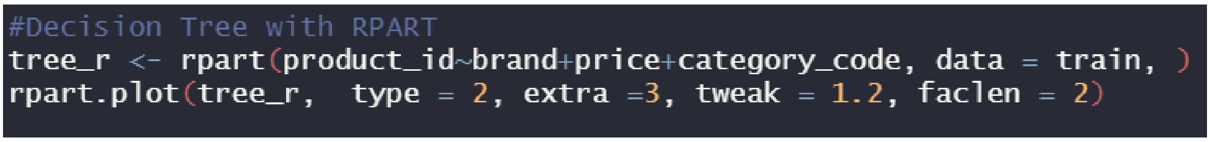
Figure 1: Splitting the Dataset into Test and Train in R

After the data preparation process, the dataset now is ready to be used in model building.

4.0 Machine Learning Algorithms

4.1 Decision Tree in Classification

Next, the decision tree method with classification will be used to build the model. The target attribute is the ‘product_id’ and the other attribute such as ‘brand’, ‘price’, ‘category_code’ will be used as the input attributes. The equation in R is stated as below. After training the tree model, the tree have been plot in R.



```
#Decision Tree with RPART
tree_r <- rpart(product_id~brand+price+category_code, data = train, )
rpart.plot(tree_r,  type = 2, extra =3, tweak = 1.2, faclen = 2)
```

Figure 2 : Equation of Decision Tree with R and Tree Plot

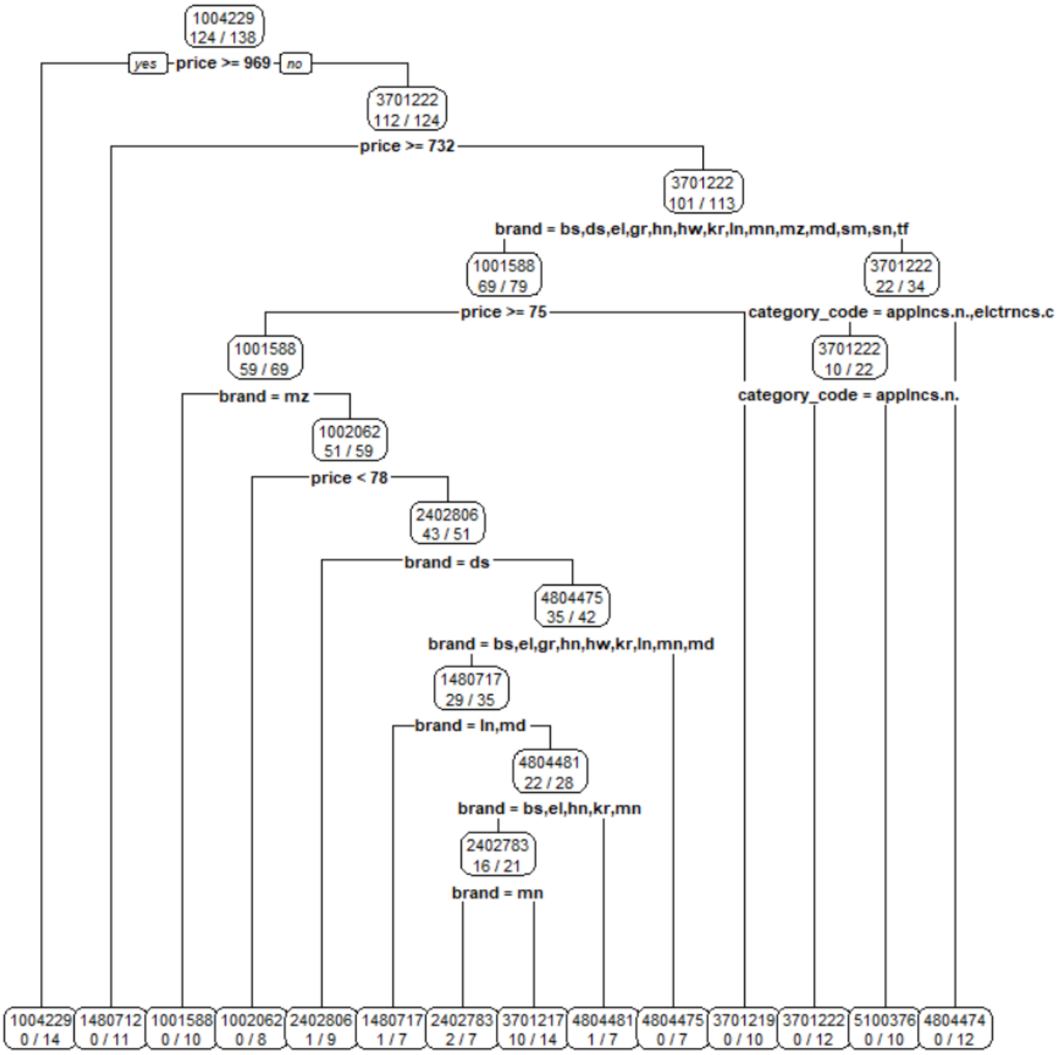
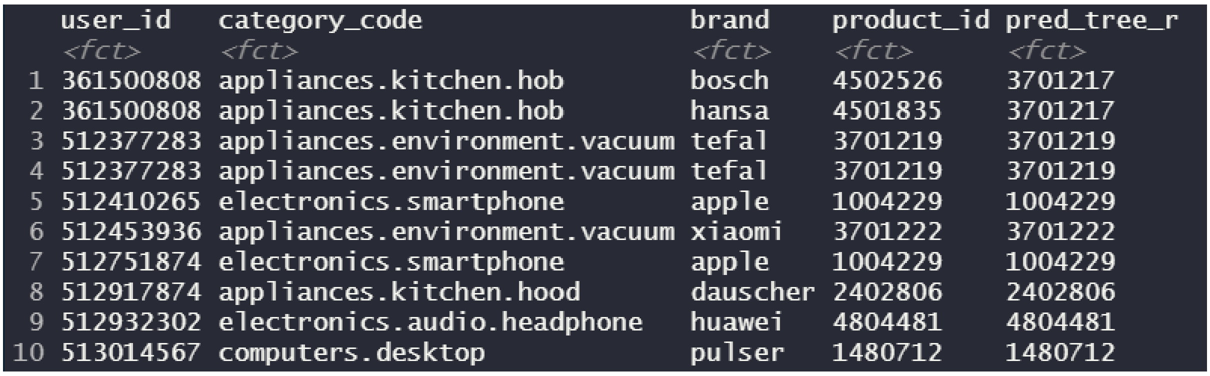


Figure 3 : The Decision Tree Built

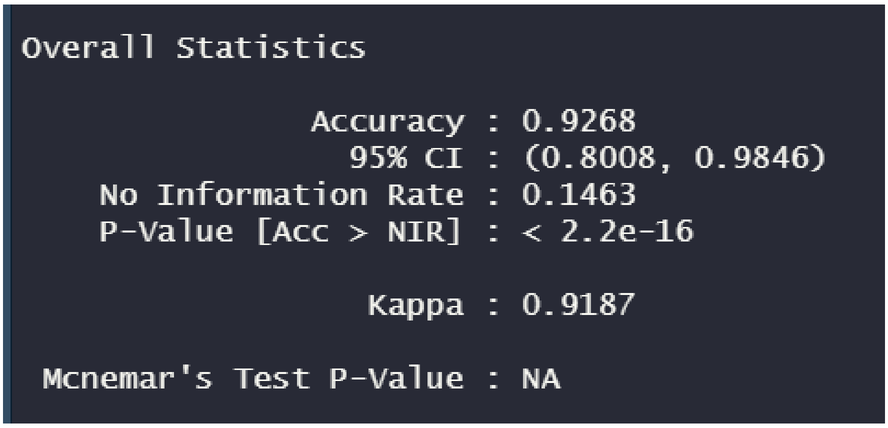
From the decision tree model, we can observed that the price with higher equal than or lower than 969 is the root of the tree. The price act as the highest two rank of decision rules, then the brand act as the third decision rules for the model. The recommended product to the test data and the true product bought have been plotted side by side in the below table.



```
   user_id   category_code                  brand    product_id pred_tree_r
   <fct>     <fct>                          <fct>    <fct>      <fct>      
 1 361500808 appliances.kitchen.hob         bosch    4502526    3701217    
 2 361500808 appliances.kitchen.hob         hansa    4501835    3701217    
 3 512377283 appliances.environment.vacuum tefal    3701219    3701219    
 4 512377283 appliances.environment.vacuum tefal    3701219    3701219    
 5 512410265 electronics.smartphone         apple    1004229    1004229    
 6 512453936 appliances.environment.vacuum xiaomi   3701222    3701222    
 7 512751874 electronics.smartphone         apple    1004229    1004229    
 8 512917874 appliances.kitchen.hood        dauscher 2402806    2402806    
 9 512932302 electronics.audio.headphone    huawei   4804481    4804481    
10 513014567 computers.desktop              pulser   1480712    1480712    
```

Figure 4: The Test Data Product with Recommended Product in First 10 Rows

The ‘product_id’ of the data is the true product that have been bought by the user, where the ‘pred_tree_r’ is the recommend product to the user based on the brands and category code on the first 10 rows. We can observed that except for the first two rows, all of the recommendation predict correctly to the user on the test data. After building the tree and showing the recommend table, the performance of the tree have been evaluated by using the confusion matrix. The performance of the trees is stated in the below image.



```
Overall Statistics
                                          
               Accuracy : 0.9268          
                 95% CI : (0.8008, 0.9846)
    No Information Rate : 0.1463          
    P-Value [Acc > NIR] : < 2.2e-16       
                                          
                  Kappa : 0.9187          
                                          
 Mcnemar's Test P-Value : NA              
```

Figure 5: Confusion Performance of the Decision Tree Built

The performance of the Decision Tree is satisfying with an accuracy score of 0.9268. Besides, the Kappa which is the random accuracy rate of the model is about 0.9187 that is also higher than 90%. In this case, we can conclude that the recommendation process by using Decision Tree model is satisfying and it is recommended to be use for recommendation system in this dataset.

4.2 Random Forest

In this part, Random Forest Model will be built for the dataset and their characteristics will be discussed. After model tunning, the accuracy of Random Forest Model stays high for about 0.95 < as long as the number of trees is more than 20. To ensure that the Random Forest model perform better and cover more information, the number of trees with 60 have been used. The graph of error against number of trees and equation of building the Random Forest Model have been shows in the below image.

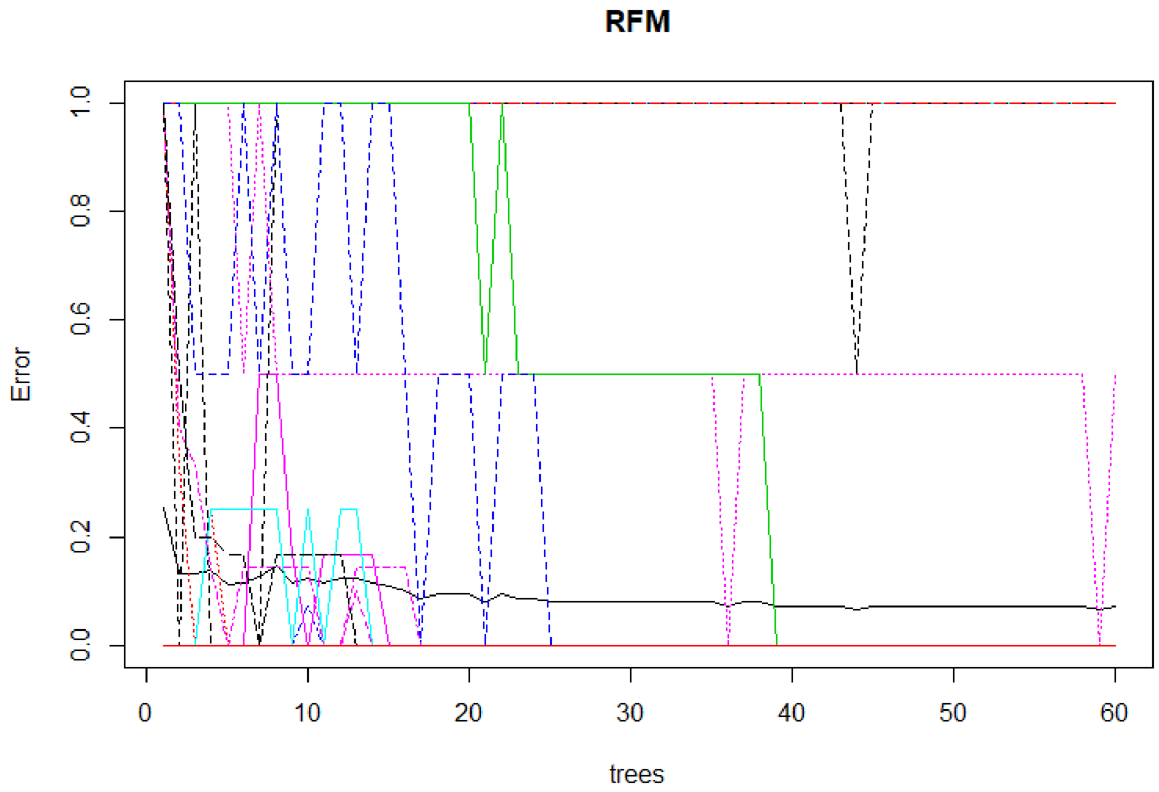
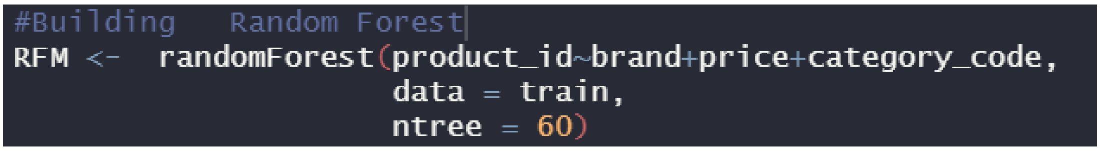


Figure 6: The Error Rate of Model to the Number of Trees



```
#Building   Random Forest
RFM <-   randomForest(product_id~brand+price+category_code,
                      data = train,
                      ntree = 60)
```

Figure 7: Equation to Build Random Forest Model in R

After building the Random Forest Model, the characteristics of the Random Forest model such as the number of nodes and the important nodes will be evaluate. The above graph describe the frequency of trees that contains the nodes' size In the Random Forest. The tree that contains 17 nodes have the highest frequency in the Random Forest Model, as frequency of 12 in 60.

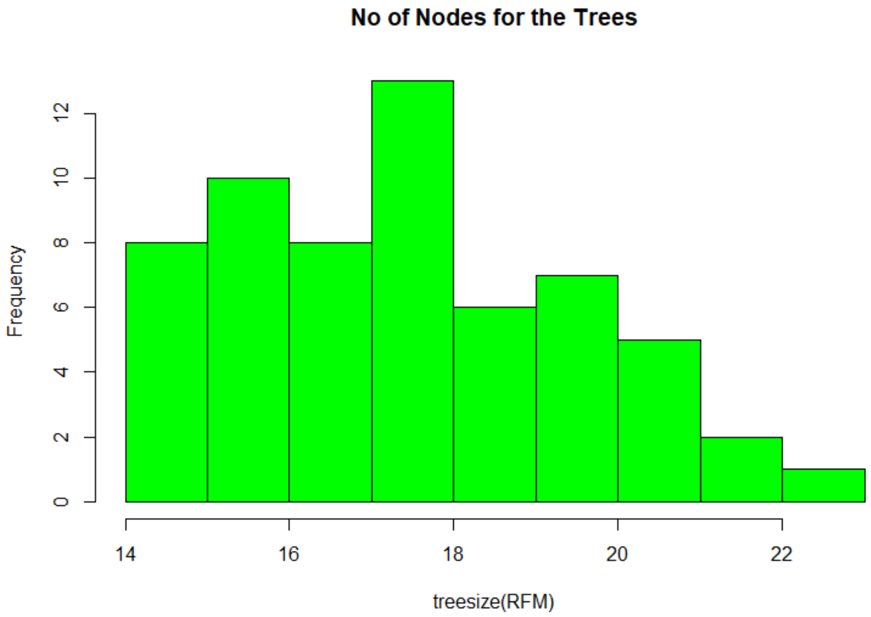


Figure 8: Frequency of Tree Nodes against the Tree Size

The Gini captures how pure the nodes are at the end of the tree. From the below graph we can observed that the Price attribute have the highest contribution on data entropy, as same with the Decision Tree, the attribute price have the highest Gini value and ends out become the root of the decision rules in the tree model. However both other two of the attribute also contribute a certain fair amount of impurity to the Random Forest Model.

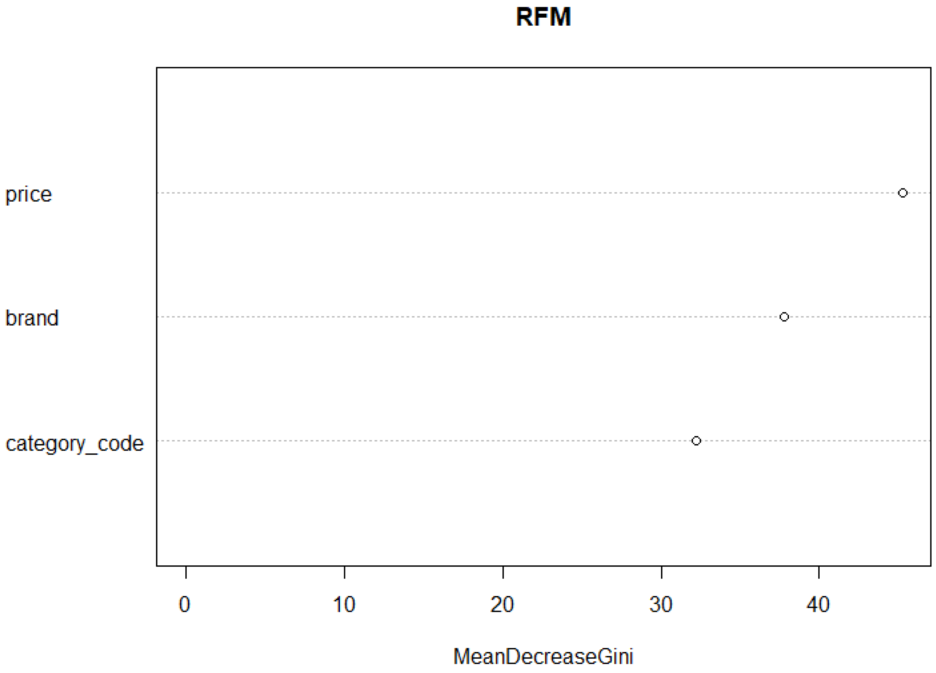
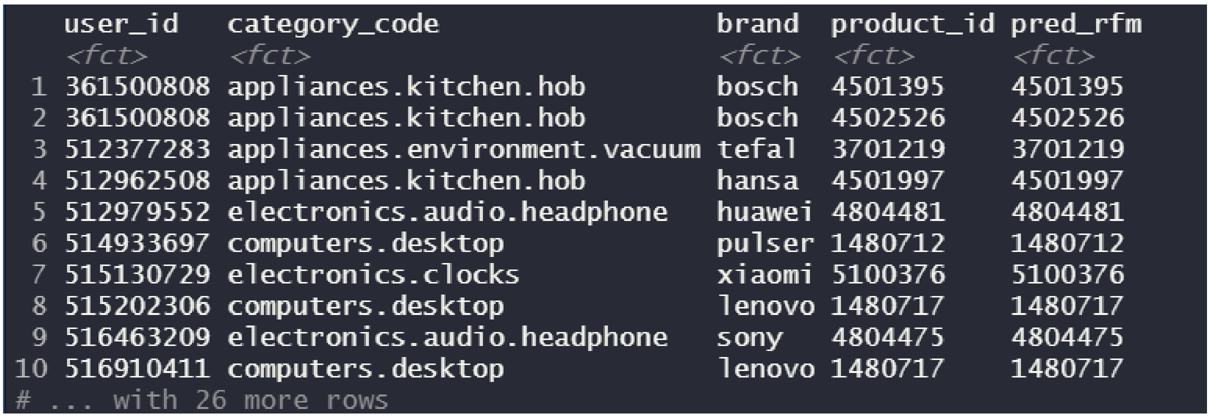


Figure 9: The Attribute against the Decrease of Gini value if Removed

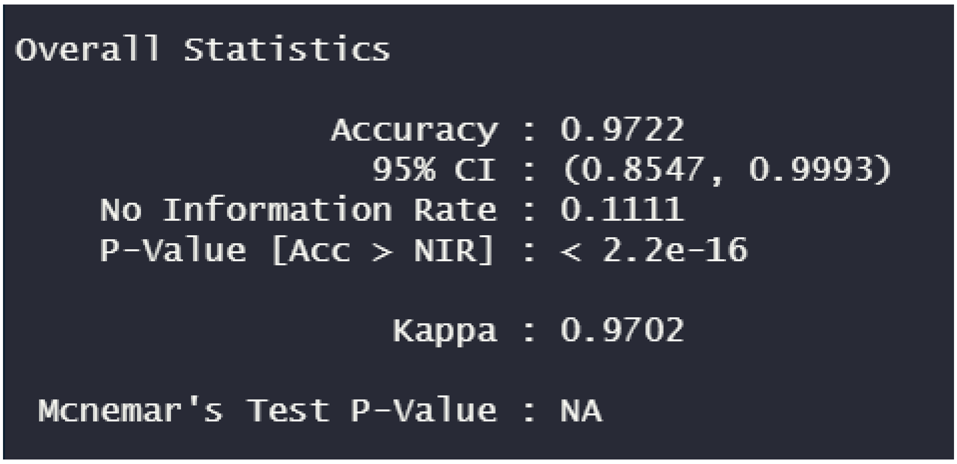
The below diagrams shows the recommended product to the user and their actual product bought in the test data for the first 10 rows. We can observed that the model successfully recommended the actual product the user interests for all the ten rows.



```
   user_id   category_code                  brand  product_id pred_rfm
   <fct>     <fct>                          <fct>  <fct>      <fct>
 1 361500808 appliances.kitchen.hob         bosch  4501395    4501395
 2 361500808 appliances.kitchen.hob         bosch  4502526    4502526
 3 512377283 appliances.environment.vacuum  tefal  3701219    3701219
 4 512962508 appliances.kitchen.hob         hansa  4501997    4501997
 5 512979552 electronics.audio.headphone    huawei 4804481    4804481
 6 514933697 computers.desktop              pulser 1480712    1480712
 7 515130729 electronics.clocks             xiaomi 5100376    5100376
 8 515202306 computers.desktop              lenovo 1480717    1480717
 9 516463209 electronics.audio.headphone    sony   4804475    4804475
10 516910411 computers.desktop              lenovo 1480717    1480717
# ... with 26 more rows
```

Figure 10: The Test Data Product with Recommended Product in First 10 Rows by RFM

After plotting the table of recommended products and looking at the characteristics of the Random Forest model, the performance of the Model is evaluate by using confusion matrix. The confusion matrix score of the model is shown as below.



```
Overall Statistics

               Accuracy : 0.9722
                 95% CI : (0.8547, 0.9993)
    No Information Rate : 0.1111
    P-Value [Acc > NIR] : < 2.2e-16

                  Kappa : 0.9702

 Mcnemar's Test P-Value : NA
```

Figure 11: Confusion Performance of the Random Forest Model

The performance of the Random Forest Model is satisfying with an accuracy score of 0.9722. Besides, the Kappa which is the random accuracy rate of the model is about 0.9702 that is also higher than 90%. In this case, we can conclude that the recommendation process by using Decision Tree model is satisfying and it is recommended to be use for recommendation system in this dataset.

5.0 Conclusion

In conclusion, we have performed two machine learning algorithm in the recommendation system on the dataset which are Random Forest and Decision Tree with Classification. Both of the model is satisfying with the accuracy of higher than 0.90.

Bibliography

Fernando, J. (18 Nov, 2020). *R-Squared Definition*. Retrieved from Investopedia: https://www.investopedia.com/terms/r/r-squared.asp