Final Report: Business Analysis

for the Grocery Store

Hui He
Zhimao Lin

Operational Management 420
December 8, 2018

Table of Content

Executive Summary

Data Preparation **4**

customersT Table 4

subdeptsT Table 4

specialsT Table 4

transactionsT Table 5

Goal 1: Two Interesting Findings **5**

The first interesting finding 5

Statement 5

Methodology 6

Business Implication 6

The second interesting finding 7

Statement 7

Methodology 7

Business Implication 8

Goal 2: Prediction Model **8**

Business Objective 8

Definition 8

Methodology 9

Decision Tree 11

Random Forest 12

LDA 12

Evaluation and Comparison 12

Business Implication 13

References **14**

Executive Summary

The purpose of this report is to show the analysis of data set from the grocery.accdb file for the grocery store. It includes two interesting findings and three prediction models:

- Interesting Finding 1: The top 10 most profitable departments in the grocery store
- Interesting Finding 2: People who buy products in ‘Meat-Pork’ and ‘Produce-Fresh-cut’ sub-departments are likely to buy products in ‘Frozen Food’ subdepartment
- Prediction Model Objective: Who are big shoppers?

This report also documents the data preparation process, corresponding result descriptions, methodologies, and business implications during analyses. In the prediction model section, this report documents the evaluation and comparison of three different models as well.

Data Preparation

We used Microsoft SQL Server Management Studio 17 to implement the data preparation on the grocery data. There are 5 tables in the grocery data: “deptsT”, “customersT”, “specialsT”, “subdeptsT”, and “transactionsT”. Among them, we found there is no issue in “deptsT” table.

customersT Table

For “customersT” table, we found there are 96 entries whose cu_age_range is NULL or 0. Also, there are 351 entries whose item cu_gender is NULL or “A”. In addition, there are 78 entries whose customer_type is NULL or “charge”. As a result, there are 363 invalid entries in total, which is 4.3% of the data in this table. Since the percentage is very low, we decided to delete those records from the table. Besides, we found some entries with “HO” or 0 customer_type. Since there is no clear evidence they are invalid, we decided not to delete them.

subdeptsT Table

For “subdeptsT” table, we found there is only 1 NULL entry. The percentage is only 0.94% and we decided to delete that row. Besides, we found there are 55,819 entries are from sub_department ID 31 in the “transactionsT” table, which is 7% in “transactionsT” table. However, the sub_department ID 31 is missing in “subdeptsT” table. Since the percentage is relatively high, we added a subdepart_id 31 in “subdeptsT” Table. As most of the product descriptions from sub-department 31 are about vegetables, we decided to call sub-department description of 31 as “Produce - Vegetable”, and department_id is set to 3(“Produce”).

specialsT Table

For “specialsT” table, there is no invalid data. However, some entries special_id of “regular” are NULL. So, we updated those entries’ special_id to 0.

transactionsT Table

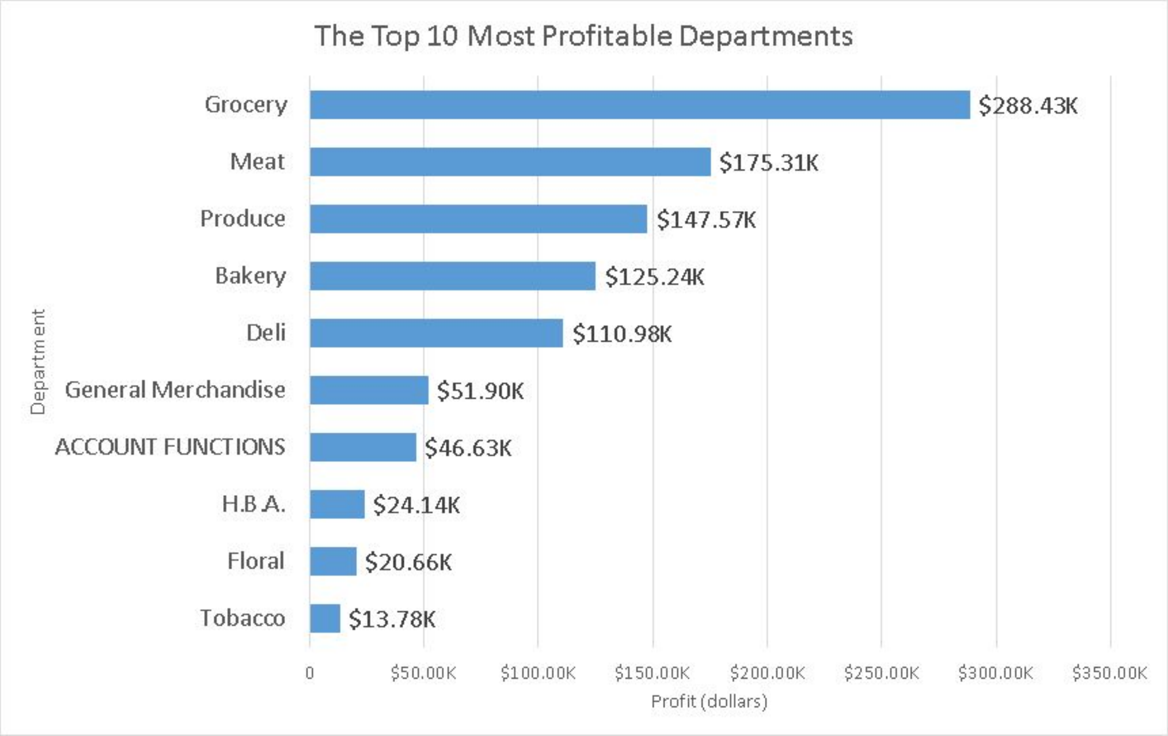
For “transactionsT” table, we found there are 48 entries whose sub-department number is null or 0; however, there is no sub-department 0 in the “subdeptsT” table. Also, there are 2,086 entries whose item description is “Item Not on File.” As a result, there are 2,134 invalid entries in total, which is 0.27% of the data in this table. Since the percentage is very low, we decided to delete those records from the table.

Goal 1: Two Interesting Findings

The first interesting finding

Statement

We found the top 10 most profitable departments in the grocery store.

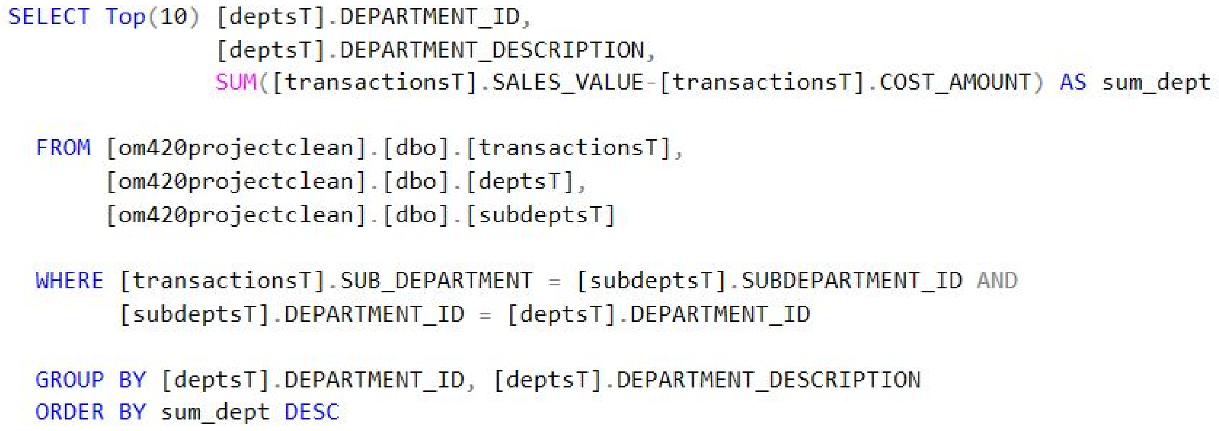


From the clustered bar graph above, it shows the total profit for each department. Based on the data, the grocery department is the most profitable department

($288.43k), whose profit is almost double higher than the meat department, which is the second most profitable department($175.31K). Also, an interesting observation is that the account function department is the 7th profitable department.

Methodology

In the Microsoft SQL Server Management Studio, we wrote a SQL query to get the top 10 profitable departments. The following is the SQL query:



```
SELECT Top(10) [deptsT].DEPARTMENT_ID,
               [deptsT].DEPARTMENT_DESCRIPTION,
               SUM([transactionsT].SALES_VALUE-[transactionsT].COST_AMOUNT) AS sum_dept

  FROM [om420projectclean].[dbo].[transactionsT],
       [om420projectclean].[dbo].[deptsT],
       [om420projectclean].[dbo].[subdeptsT]

  WHERE [transactionsT].SUB_DEPARTMENT = [subdeptsT].SUBDEPARTMENT_ID AND
        [subdeptsT].DEPARTMENT_ID = [deptsT].DEPARTMENT_ID

  GROUP BY [deptsT].DEPARTMENT_ID, [deptsT].DEPARTMENT_DESCRIPTION
  ORDER BY sum_dept DESC
```

Since each entry in “transactionsT” table only contains the sale values for each sub-department, we joined the “transactionsT” table and the “subdeptsT” table to get the total profit for each sub-department. Then, we also joined the “deptsT” table and grouped by department ID to get the total profit for each department. Lastly, we ordered the profit in descending order and selected the top 10 most profitable departments. Finally, we put the result to Microsoft Excel to generate clustered bar diagram for data visualization purpose.

Business Implication

The grocery store’s marketing team can focus on products in grocery, meat, produce, bakery, and deli departments since these departments provide the most profit. For example, they can increase the number of coupons in those departments to attract more customers. Besides, as the account function department is the 7th profitable department, the grocery store should pay more attention to the account function department. Since the account function department is a service-based department, they can easily increase the profit by introducing more and high-quality account services.

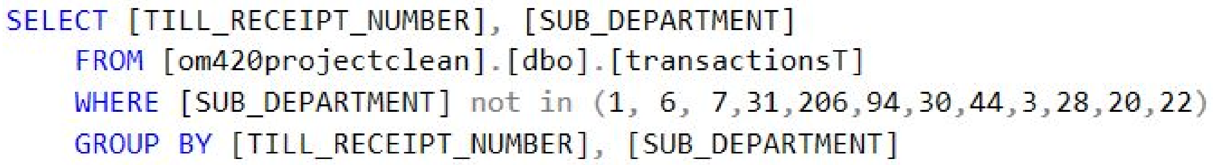
The second interesting finding

Statement

People who purchase products in “Meat-Pork”(ID: 21) and “Produce-Fresh-cut”(ID: 32) sub-departments are likely to purchase products in “Frozen Food” subdepartment(ID: 4).

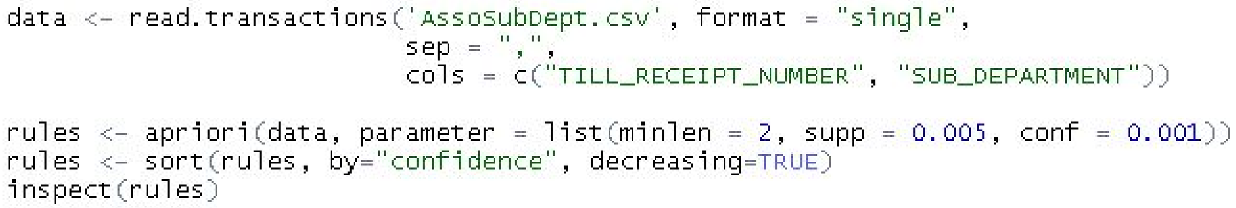
Methodology

In the Microsoft SQL Server Management Studio, we wrote a SQL query to get all the receipt numbers and the corresponding sub-department IDs. The following is the SQL query:



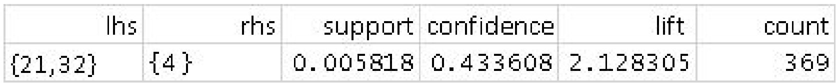
```
SELECT [TILL_RECEIPT_NUMBER], [SUB_DEPARTMENT]
    FROM [om420projectclean].[dbo].[transactionsT]
    WHERE [SUB_DEPARTMENT] not in (1, 6, 7,31,206,94,30,44,3,28,20,22)
    GROUP BY [TILL_RECEIPT_NUMBER], [SUB_DEPARTMENT]
```

Sub-departments 1, 6, 7, 31, 206, 94, 30, 44, 3, 28, 20, and 22 are the most frequently appearing sub-departments. It would be meaningless if we include them in finding association rules as most customers will buy products from those sub-departments. Then, we exported the data to a CSV file and to look for association rules in RStudio. The following is the codes in RStudio:



```
data <- read.transactions('AssoSubDept.csv', format = "single",
                          sep = ",",
                          cols = c("TILL_RECEIPT_NUMBER", "SUB_DEPARTMENT"))

rules <- apriori(data, parameter = list(minlen = 2, supp = 0.005, conf = 0.001))
rules <- sort(rules, by="confidence", decreasing=TRUE)
inspect(rules)
```

After several attempts, we set support = 0.005 and confidence = 0.001 to get the most meaningful results. From all the association rule results, some entries have a high lift because they are in the same department. After sorting confidence values descendingly, we consider the second one is most meaningful one because ‘Frozen Food’ subdepartment(ID: 4) is the 11th most profitable sub-department.



lhs	rhs	support	confidence	lift	count
{21,32}	{4}	0.005818	0.433608	2.128305	369

According to the table, there are 0.58% of the total buskets having products from sub-departments 21,32, and 4. The confidence means the probability of a customer also purchasing products from sub-department 4 is about 43.36% given the customer who has purchased products from sub-department 21 and 32. The lift is even greater than 2, which means the probability of a customer who buys products from sub-department 4 is increased by over 2 times after knowing the customer has bought products from sub-department 21 and 32.

Business Implication

Since people who buy products in “Meat-Pork” and “Produce-Fresh-cut” sub-departments have a greater possibility to buy products in “Frozen Food” sub-department, and the “Frozen Food” sub-department is very profitable. The store marketing team can increase the numbers of special sales in “Meat-Pork” and “Produce-Fresh-cut” sub-departments to boost the sales in the “Frozen Food” sub-department. Besides, the grocery store can also offer bundled products in “Meat-Pork” and “Produce-Fresh-cut” sub-departments to increase sales in “Frozen Food” sub-department.

Goal 2: Prediction Model

Business Objective

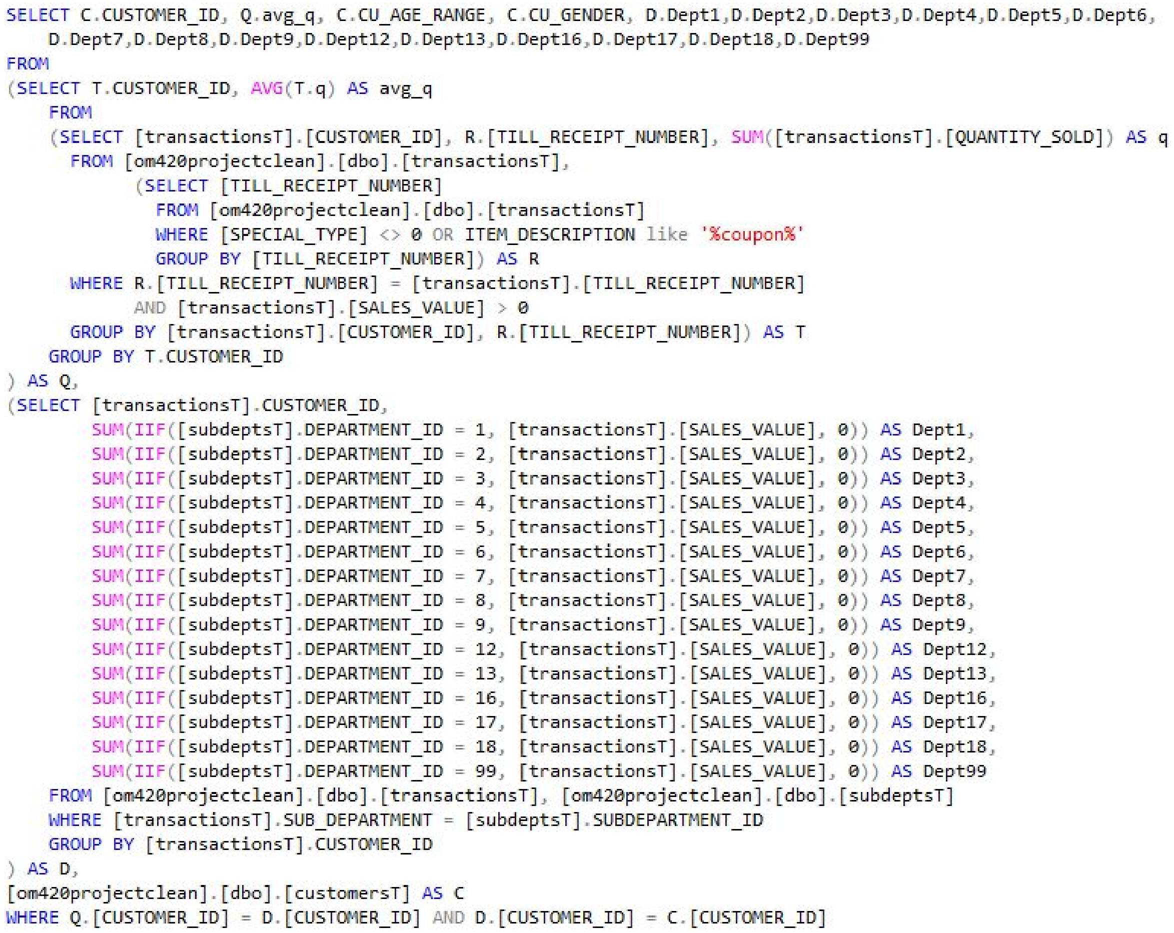
Our objective is to find who are big shoppers among all shoppers that used coupons. With this prediction model, we can send those big shoppers coupons and flyers and encourage them to buy more products in order to increase profit. Since our business strategy is to send coupons and flyers, we only focus on customers who would like to use coupons.

Definition

We consider customers who are big shoppers if their average quantity of product bought per receipt is greater than 13.5 among all receipts that include coupons and special sales. The reason why 13.5 is used as a threshold is that among all receipts that include coupons or special sales, the average quantity of products bought per receipt is 13.5.

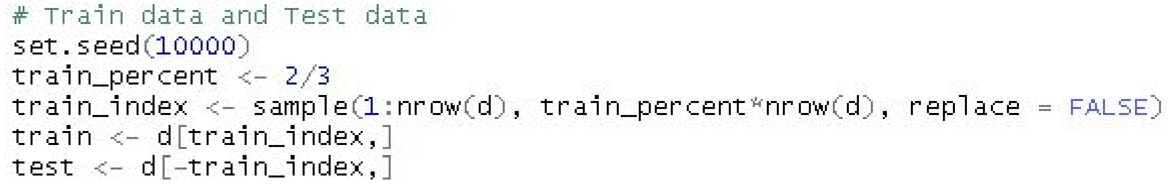
Methodology

Firstly, we set our prediction target as a boolean variable: whether a customer’s average quantity of product bought per receipt is greater than 13.5 among all receipts that include coupons and special sales. Secondly, we used customers’ age, gender, and spending habit as predictors. We wrote a SQL query to get the age, gender, and the spending of each customer from each department. The following is the SQL query:



```sql
SELECT C.CUSTOMER_ID, Q.avg_q, C.CU_AGE_RANGE, C.CU_GENDER, D.Dept1,D.Dept2,D.Dept3,D.Dept4,D.Dept5,D.Dept6,
    D.Dept7,D.Dept8,D.Dept9,D.Dept12,D.Dept13,D.Dept16,D.Dept17,D.Dept18,D.Dept99
FROM
(SELECT T.CUSTOMER_ID, AVG(T.q) AS avg_q
    FROM
    (SELECT [transactionsT].[CUSTOMER_ID], R.[TILL_RECEIPT_NUMBER], SUM([transactionsT].[QUANTITY_SOLD]) AS q
      FROM [om420projectclean].[dbo].[transactionsT],
            (SELECT [TILL_RECEIPT_NUMBER]
              FROM [om420projectclean].[dbo].[transactionsT]
              WHERE [SPECIAL_TYPE] <> 0 OR ITEM_DESCRIPTION like '%coupon%'
              GROUP BY [TILL_RECEIPT_NUMBER]) AS R
      WHERE R.[TILL_RECEIPT_NUMBER] = [transactionsT].[TILL_RECEIPT_NUMBER]
            AND [transactionsT].[SALES_VALUE] > 0
      GROUP BY [transactionsT].[CUSTOMER_ID], R.[TILL_RECEIPT_NUMBER]) AS T
    GROUP BY T.CUSTOMER_ID
) AS Q,
(SELECT [transactionsT].CUSTOMER_ID,
        SUM(IIF([subdeptsT].DEPARTMENT_ID = 1, [transactionsT].[SALES_VALUE], 0)) AS Dept1,
        SUM(IIF([subdeptsT].DEPARTMENT_ID = 2, [transactionsT].[SALES_VALUE], 0)) AS Dept2,
        SUM(IIF([subdeptsT].DEPARTMENT_ID = 3, [transactionsT].[SALES_VALUE], 0)) AS Dept3,
        SUM(IIF([subdeptsT].DEPARTMENT_ID = 4, [transactionsT].[SALES_VALUE], 0)) AS Dept4,
        SUM(IIF([subdeptsT].DEPARTMENT_ID = 5, [transactionsT].[SALES_VALUE], 0)) AS Dept5,
        SUM(IIF([subdeptsT].DEPARTMENT_ID = 6, [transactionsT].[SALES_VALUE], 0)) AS Dept6,
        SUM(IIF([subdeptsT].DEPARTMENT_ID = 7, [transactionsT].[SALES_VALUE], 0)) AS Dept7,
        SUM(IIF([subdeptsT].DEPARTMENT_ID = 8, [transactionsT].[SALES_VALUE], 0)) AS Dept8,
        SUM(IIF([subdeptsT].DEPARTMENT_ID = 9, [transactionsT].[SALES_VALUE], 0)) AS Dept9,
        SUM(IIF([subdeptsT].DEPARTMENT_ID = 12, [transactionsT].[SALES_VALUE], 0)) AS Dept12,
        SUM(IIF([subdeptsT].DEPARTMENT_ID = 13, [transactionsT].[SALES_VALUE], 0)) AS Dept13,
        SUM(IIF([subdeptsT].DEPARTMENT_ID = 16, [transactionsT].[SALES_VALUE], 0)) AS Dept16,
        SUM(IIF([subdeptsT].DEPARTMENT_ID = 17, [transactionsT].[SALES_VALUE], 0)) AS Dept17,
        SUM(IIF([subdeptsT].DEPARTMENT_ID = 18, [transactionsT].[SALES_VALUE], 0)) AS Dept18,
        SUM(IIF([subdeptsT].DEPARTMENT_ID = 99, [transactionsT].[SALES_VALUE], 0)) AS Dept99
    FROM [om420projectclean].[dbo].[transactionsT], [om420projectclean].[dbo].[subdeptsT]
    WHERE [transactionsT].SUB_DEPARTMENT = [subdeptsT].SUBDEPARTMENT_ID
    GROUP BY [transactionsT].CUSTOMER_ID
) AS D,
[om420projectclean].[dbo].[customersT] AS C
WHERE Q.[CUSTOMER_ID] = D.[CUSTOMER_ID] AND D.[CUSTOMER_ID] = C.[CUSTOMER_ID]
```

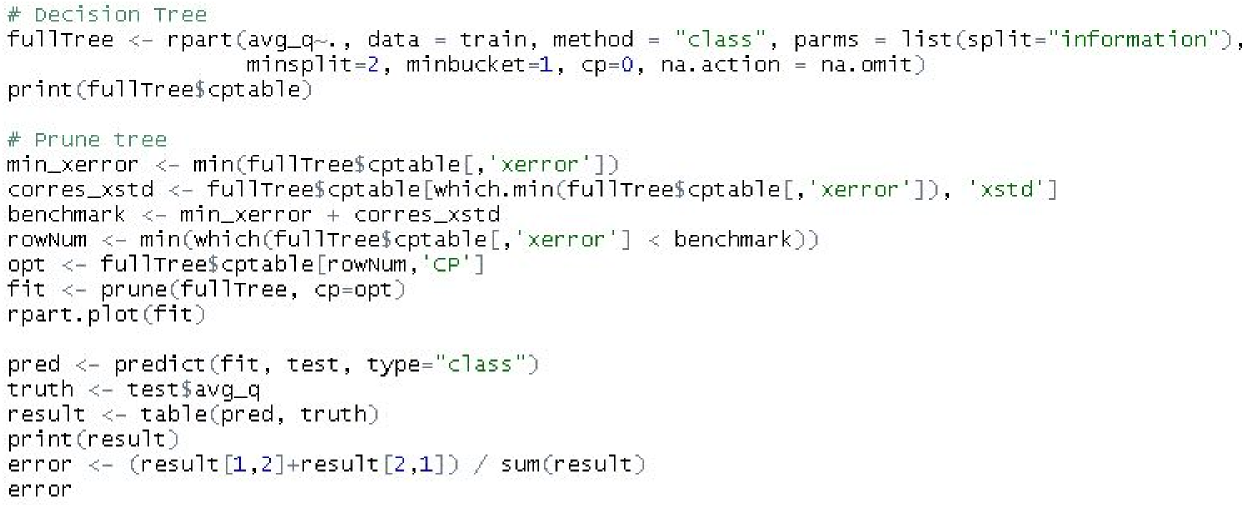
We did not include department 10, 11, 14, 15, or 19 because the total sale value of department of 10, 11, 14, and 15 are 0, and the total sale value of department 19 is $37.43, which is a very small number considering the size of the “transactionsT” table. Then, we exported the result into a CSV file and load the data into RStudio and derived the prediction target in R. After that, we randomly put ⅓ of our data aside as test dataset and used the rest of the data as training dataset.



```r
# Train data and Test data
set.seed(10000)
train_percent <- 2/3
train_index <- sample(1:nrow(d), train_percent*nrow(d), replace = FALSE)
train <- d[train_index,]
test <- d[-train_index,]
```

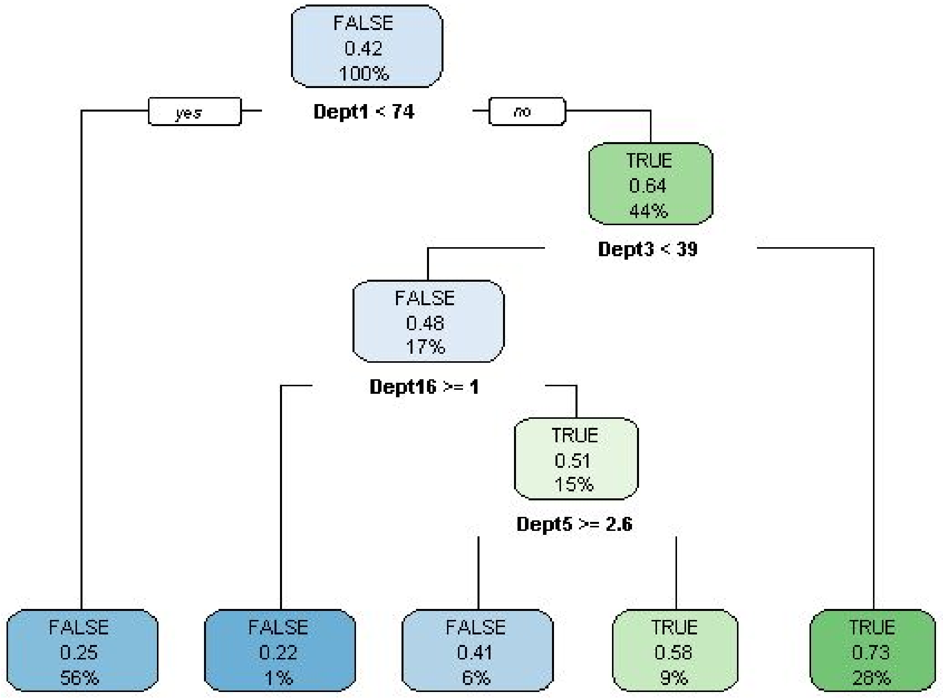
For the creation of the prediction model, we apply Decision Tree, Random Forest, and Linear Discriminant Analysis (LDA) methods to our training dataset to predict our target in R. In the end, we used our test dataset to test the performance of our data model and calculated the error rate of each model.

Decision Tree

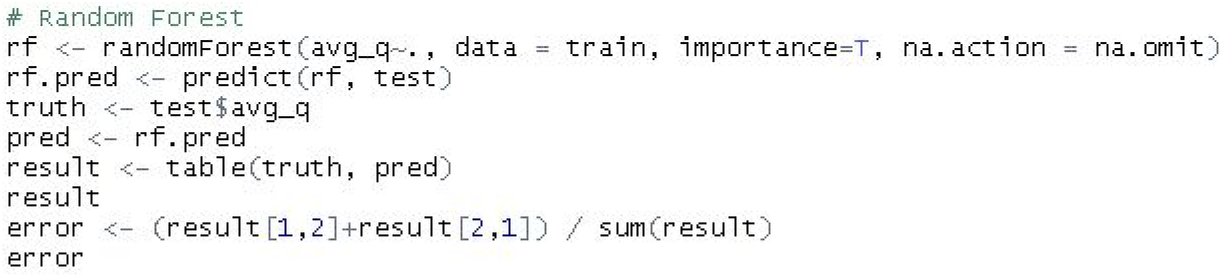


```
# Decision Tree
fullTree <- rpart(avg_q~., data = train, method = "class", parms = list(split="information"),
                  minsplit=2, minbucket=1, cp=0, na.action = na.omit)
print(fullTree$cptable)

# Prune tree
min_xerror <- min(fullTree$cptable[,'xerror'])
corres_xstd <- fullTree$cptable[which.min(fullTree$cptable[,'xerror']), 'xstd']
benchmark <- min_xerror + corres_xstd
rowNum <- min(which(fullTree$cptable[,'xerror'] < benchmark))
opt <- fullTree$cptable[rowNum,'CP']
fit <- prune(fullTree, cp=opt)
rpart.plot(fit)

pred <- predict(fit, test, type="class")
truth <- test$avg_q
result <- table(pred, truth)
print(result)
error <- (result[1,2]+result[2,1]) / sum(result)
error
```

We generated the full decision tree first and then prune it using the optimal CP value. The following is the final decision tree:



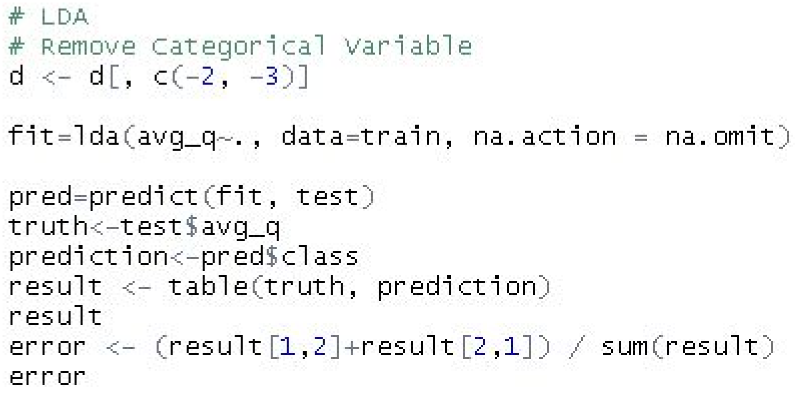
Random Forest



```
# Random Forest
rf <- randomForest(avg_q~., data = train, importance=T, na.action = na.omit)
rf.pred <- predict(rf, test)
truth <- test$avg_q
pred <- rf.pred
result <- table(truth, pred)
result
error <- (result[1,2]+result[2,1]) / sum(result)
error
```

We continued using the training dataset to train the Random Forest model and test it using the test dataset.

LDA

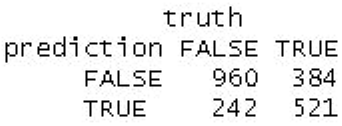


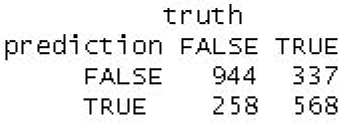
```
# LDA
# Remove Categorical Variable
d <- d[, c(-2, -3)]

fit=lda(avg_q~., data=train, na.action = na.omit)

pred=predict(fit, test)
truth<-test$avg_q
prediction<-pred$class
result <- table(truth, prediction)
result
error <- (result[1,2]+result[2,1]) / sum(result)
error
```

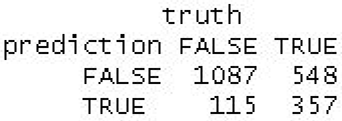
Since LDA cannot use categorical variables as predictors, we have to remove the customer age and gender from our training dataset and test dataset. After the removal, we trained the LDA model using the training data set and test it using the test dataset.

Evaluation and Comparison

The following is the prediction result and the error rate of each data model tested against our test dataset:

Prediction Model	Confusion Matrix	Overall Error Rate	Class 1 Error Rate
Decision Tree	 truth prediction FALSE TRUE FALSE 960 384 TRUE 242 521	29.71%	42.43%

Random Forest	
	`          truth`
	`prediction FALSE TRUE`
	`     FALSE   944  337`
	`     TRUE    258  568`
		28.24%	37.24%

Linear Discriminant Analysis (LDA)	
	`          truth`
	`prediction FALSE TRUE`
	`     FALSE  1087  548`
	`     TRUE    115  357`
		31.47%	60.55%

According to the table, in general, these prediction models perform well. The overall error rate of these three prediction model does not differ a lot. However, Random Forest has the lowest class 1 error rate(truth is “TRUE” but predict “FALSE”). LDA has the highest class 1 error rate since it cannot take categorical predictors. Thus, it has a higher confidence level than the LDA model if the Random Forest model predicts a customer as a big shopper.

Lessons Learned from Failure

In the beginning, we only included two predictors, customer age and gender. However, the decision tree prediction model had about 50% overall error rate. This was a very bad result because this is the same as flipping a coin. After a closer look at our transaction table, we realized that we can use customers’ spending habit as additional predictors. Then, we wrote a SQL query to find the spending of each department for each customer and included those in the prediction model. In the end, we got a much better result. Therefore, we think a lesson that we learned from here is that we should try to get more information from our data through deeper analysis. Also, we should consider adding more predictors when the prediction model does not predict well.

Business Implication

Since we can predict if customers are big shoppers, we can target these customers and send them coupons and flyers. As a result, we can expect them to buy more products with coupons and special sales. In addition, we can leverage our second finding in goal 1. Since customers who buy products from “Meat-Pork” and “Produce-Fresh-cut” sub-department tend to buy products from “Frozen Food” sub-department, we can send then coupons of “Meat-Pork” and “Produce-Fresh-cut” product and expect them to buy more from “Frozen Food” sub-department, which is the 11th profitable sub-department. In this way, the grocery store can have a huge increase in their profit.

References

- https://en.wikipedia.org/wiki/Produce
- https://www.rdocumentation.org/
- https://www.rdocumentation.org/packages/arules/versions/1.6-1/topics/read.transactions
- https://docs.microsoft.com/en-us/sql/ssms/sql-server-management-studio-ssms?view=sql-server-2017
- https://www.w3schools.com/sql/