**OM 620 - Predictive Business Analytics**

**Project Report - Grocery Data**

**Analysis and Prediction of Customer Profile**

Dec 9th, 2023

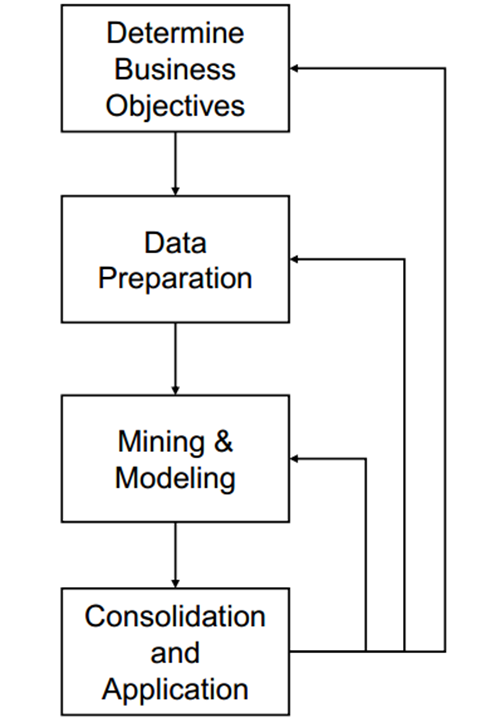
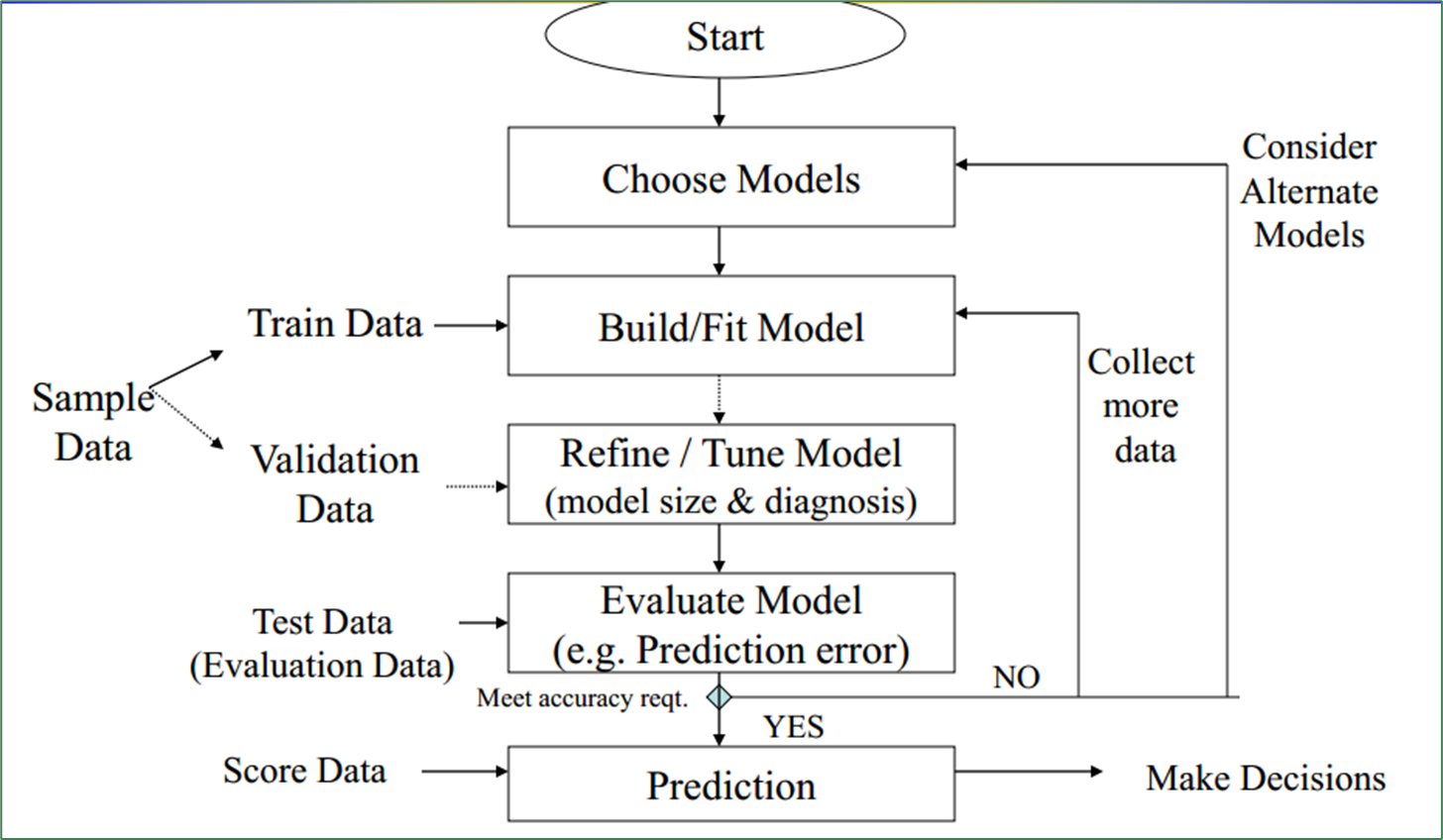
Prepared by: Benjamin Palanthikal, Harry Singh

| **Table of contents** | **Page** |
| --- | --- |
| Introduction & Objective Setting | 1 - 2 |
| Goal #1 - Insights and Correlations | 2 - 4 |
| Goal #2 - Prediction Models and Evaluation | 5 - 7 |

**Introduction**

With the increased usage of data at every operational level in making strategic decisions, developing stronger and more reliable prediction models has become highly significant for businesses. Today, if as a customer, you call Amazon’s customer support, there is no human representative that speaks to you. Yes - they have been almost entirely replaced by voice bots. There is an A.I. bot that takes inputs from your speech and converts them into information to be processed. Who builds, controls and manages these voice bots? How do they do it? It's humans - teams of data scientists and analysts that work on large chunks of historic and ongoing data, and train these bots through machine learning models. Our goal throughout the Predictive Business Analytics course, as future business managers and leaders, has been to carefully understand different stages and processes involved in building accurate, reliable, and responsible prediction models by mining, preparing and processing large and noisy datasets.

The following report focuses on our analysis of the Grocery dataset which spans various dimensions of the retail experience, encompassing customer demographics, transaction details, sales, costs, purchase quantities and departmental structures. Our primary objectives are twofold: to explore the presence of (non-obvious) trends or patterns within the data and highlight key insights and, subsequently, to construct predictive models that can enhance profitability and overall operational efficiency.

Broad Overview Detailed Process Flowchart

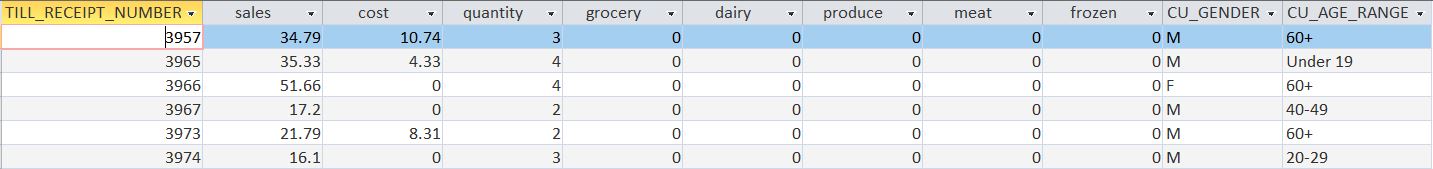
Our first task was to set up the objective. Eyeballing the raw data gave us some ideas to look at two separate data frames within the grocery dataset - “customersT” and “transactionsT”. This led us to our objective of doing a customer profiling for the grocery store. As analysts for this grocery store, we want to find correlations and trends between customer demographics and their purchasing patterns, and then we want to predict customer demographics for the store, so they can customize their promotional campaigns, special sales, and advertisements to target specific customer groups. This is a marketing analytics technique that some of the large global grocery chains are deploying to capitalize on the market share most effectively.

**Goal #1**

Our initial analysis engages in Exploratory Data Analysis. Starting with writing SQL queries (Appendix S) to join cross-referenced customers and transactions tables, we used “INNER JOIN” to join the tables on “Customer ID”. We grouped our data into a single data frame which is grouped by till receipt numbers. The removal of noise from the data included:

* Removing NA values or blank cells from the transactions table
* Excluding all other customer types, like “Staff”, “HO”, “Charge” and null values
* Excluding Customer IDs for which demographic data is not available

Interestingly, we observed that certain product types such as “Lottery” and “Tobacco” were billed at their separate specific till numbers. There were no other products that were billed at these tills besides Tobacco and Lottery. No demographic data was available for these purchases. Based on the subdepartment IDs in the transactions table against each transaction, in our SQL query, we linked the subdepartment names to record the sales value for each of these considered subdepartments, using the “IIf” function. We ended up with around 36,000 entries/data points. Below is a snippet of the table that we created in SQL (Please note that this wasn’t the final table that we used as we added more sub-departments into this, this is just for representation purposes):



Subsequent filtering and cleaning, wherever required was done on RStudio, but largely this was the dataset that we used as our input “.csv” file on RStudio. Next, to look at the customer characteristics and purchasing patterns, we used the “ggplot2” library on R and tried to look at the following visualizations:

* Histogram of “*Total Sales*” vs “*Customer Age Range*” (+ colour - for “Gender”)
* Histograms of “*Sales of xyz sub-department*” vs “*Age Range*” (+ colour - for “Gender”)
* Histograms of “*Quantity Purchased of xyz sub-department*” vs “*Age Range*” (+ colour - for “Gender”)
* Scatter Plot with Trendline of “*Total Sales*” vs “*Quantity Purchased*” (+ colour - for “Gender”)
* Boxplot of “*Total Sales*” vs “*Customer Age Range*”
* Heatmap of “*Sales of xyz sub-department*” vs “*Age Range*” (+ Gender)
* 2d Bin Plots for cross-departmental sales - “*Sales of x*” vs “*Sales of y*” (+ Gender)

These plots (Appendix G) result in correlations between demographics and purchasing habits, positive in some cases and negative in some. In the first 2 histograms, we observed almost a similar pattern for overall sales and sub-department-wise sales. The sales amount is largely normally distributed around the 30-39 year age group except for a few anomalies. The 40-60-year age group for instance, overall spends much lower than the 30-39 group, but for meat and produce, they’re among the highest spenders. This could be because this age group is expected to live in families with kids or even grandchildren in some cases, and hence there is a lot of amount spent on the staples - produce and meat.

This is also supported by the fact that the quantities purchased by this 40-49 and 50-59-year age group for produce are much higher than the average. Ideally, as per our data, and from the graph between sales amount and quantity (overall), there should be a mild positive relationship between the two, but this age group is exceptionally negative in its sales-quantity relationship. This could also indicate one of two things - either the store might be offering a lower price once the customer buys more than a certain quantity of produce, or the per-item price of certain produce items is much lower than the average per-item price of all products in the store.

The second interesting finding for us was that for both male and female customers, there are strong positive and strong negative correlations between purchases from certain sub-departments. For instance, the female customers who spend a lot on frozen food, almost spend nothing on hot food. Female customers in the 20-39 age range combined, spend less than half on hot food than what they spend on frozen food. On the other hand, there is a strong positive correlation between fruit and vegetable spending. This can help the store understand its target customers for certain low-profit or loss-making products, and try to come up with strategies to improve their average selling price. So, what is our recommendation for the store?

* *Targeted Marketing & Family-Oriented Promotions* - Specialized Loyalty Program, Bundle Promotions for better customer retention in 30-39 and 40-49 age groups
* *Discounts* - Offering discounts on bulk purchases
* *Produce Pricing* - Consider adjusting & optimizing pricing strategies
* *Personalized Advertisements* - Utilize customer transaction data to send personalized promotions, discounts, or product recommendations

**Goal #2**

The objective of our Goal #2 was to create prediction models which could accurately predict customer demographics using transaction data. The specific categorical target variables were gender and whether a person was over/under 40 years old.

Sub-department transaction totals were used as input attributes into the model since the total transaction amounts in each department vary with each demographic *(see Figure 2)*. Using sub-department transactions was favoured over specific product transactions due to the expediency of filtering the data and because of the large amount of transaction data that would be readily available. Initially, the model incorporated 10 different sub-departments; however, in an attempt to improve the accuracy of the models, the number of sub-departments was increased to 18. As further discussed in the KNN section, transaction sales were also eventually inputted into the model.

**KNN**

We initially used a KNN model to predict our gender and age demographics with the input sub-department transactions; however, our model ultimately led to having “too many ties”. To potentially improve our model, besides increasing the number of sub-departments, we also incorporated the total transaction's (i.e. till receipt) sales amount as an input attribute recognizing that these values typically are different for each demographic. For example, women typically spend more than males and those in the 40-49 age demographic also spend more than other demographics. Initial results on the validation set using a k-value of 1 are shown in *Figures 3 and 4* below. For gender, the overall error rate is 38%, the class 1 error rate is 25% and the class 2 error rate is 72%; therefore the model does a poor job of accurately predicting males. For age, the overall error rate is 45%, the class 1 error rate is 36% and the class 2 error rate is 62%; therefore the model does a poor job of accurately predicting those under 40 years old.

To further validate whether our input attributes had any significance in predicting our target variables, we decided to compare them to random numbers generated in R. As shown in *Figures 5 and 6*, a random number inputted into the prediction model gives similar error rates as those with our initial input attributes. Therefore, it can be concluded that the input attributes are too noisy and were not accurate predictors of customer demographics.

The accuracy of the KNN model was also evaluated based on the k-value chosen. As seen in *Figures 7 and 8*, the k-value was changed to 5 and 10 to determine the effect of this parameter on the error rates. Increasing the k-value for both the gender and age prediction models decreased the overall error rate; however, this came at the expense of a higher class 0 error rate or a poorer prediction of males or those under 40.

**LDA**

LDA models were also constructed to predict gender and those over 40 years old. As seen in Figure 9, the LDA models strongly over-predicted female customers and those over 40 to minimize the overall error rate.

**Random Forest**

Random Forest models were also generated to predict the target gender and age variables and the results for each model can be found in Figure 10. Like LDA, the Random Forest models strongly over-predicted female customers and those over 40 to minimize the overall error rate.

**Decision Trees**

In applying Decision Trees to predict customer demographics based on input attributes - subdepartment sales and quantity, a notable challenge encountered was the model's susceptibility to overfitting. Decision trees captured noise in the data, resulting in an excessively complex model that performed well on training data but generalized poorly to new instances.

**Model Discussion**

Both the LDA and Random Forest models have a lower overall error rate than the KNN models; however, the LDA and Random Forest models have class 0 error rates greater than 80%. Therefore these LDA and Random Forest models are simply predicting female customers or customers over 40 to minimize overall error. KNN also shows the same over-prediction tendency but with a k-value of 1, which provides greater model flexibility, and is somewhat able to minimize the Class 0 error rates. While it has the highest overall error rate, KNN models with a K-value of 1 are chosen to apply to the test data set and the results are shown in *Figure 11*. The test error rates shown in Figure 11 are similar to the validation error rates that were calculated and shown in *Figures 5 and 6.*

The overall poor predictions of each model are caused by noisy input attributes. If a similar future analysis were to be performed then it is critical that each demographic’s input attributes are highly distinctive. While on average some demographics might spend more or less compared to one another, the noise or variance within subdepartment transactions makes it difficult to delineate any meaningful differences. Therefore, increased statistical analysis of the input attributes would be required in the future prior to creating a prediction model.

**Conclusion**

KNN struggled with excessive ties and yielded poor predictions for gender and age groups, with noisy input attributes contributing to its limitations. LDA and Random Forest models exhibited lower overall error rates but displayed a propensity to over-predict females and customers over 40. Decision Trees encountered challenges with overfitting, underscoring the importance of careful parameter tuning. Despite model shortcomings, KNN with a k-value of 1 demonstrated better flexibility and alignment with test data. In summary, the models' inefficacy was rooted in the noise within subdepartment transactions, emphasizing the need for distinct and statistically relevant input attributes in future analyses.

**Appendix for the Report:**

**Appendix S:**

SELECT transactionsT.TILL\_RECEIPT\_NUMBER,

Sum(transactionsT.SALES\_VALUE) AS sales,

Sum(transactionsT.COST\_AMOUNT) AS cost,

Count(transactionsT.QUANTITY\_SOLD) AS quantity,

Sum(IIf(sub\_department In (1),sales\_value,0)) AS grocery,

Sum(IIf(sub\_department In (3),sales\_value,0)) AS dairy,

Sum(IIf(sub\_department In (30,31,32),sales\_value,0)) AS produce,

Sum(IIf(sub\_department In (20,21,22,28),sales\_value,0)) AS meat,

Sum(IIf(sub\_department In (4),sales\_value,0)) AS frozen,

Sum(IIf(sub\_department In (94,97),sales\_value,0)) AS deli,

Sum(IIf(sub\_department In (50,51),sales\_value,0)) AS hba,

Sum(IIf(sub\_department In (70),sales\_value,0)) AS horticulture,

Sum(IIf(sub\_department In (41,40,44),sales\_value,0)) AS bread,

customersT.CU\_GENDER,

customersT.CU\_AGE\_RANGE

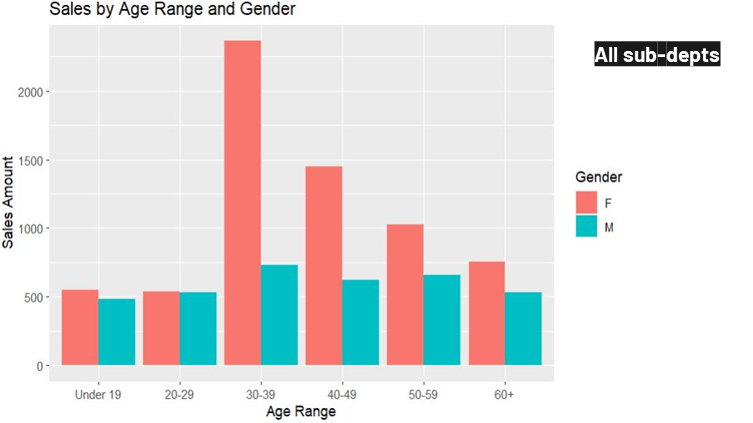
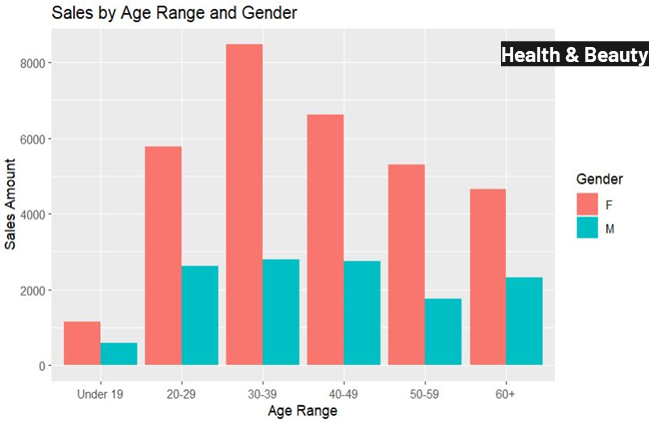
FROM transactionsT INNER JOIN customersT ON transactionsT.CUSTOMER\_ID = customersT.CUSTOMER\_ID

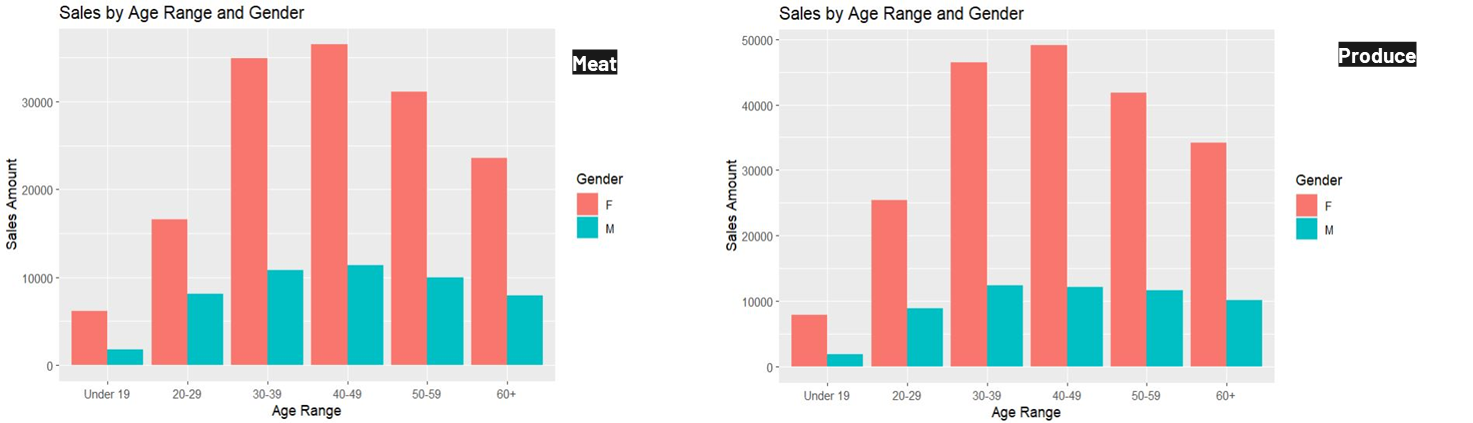
GROUP BY transactionsT.TILL\_RECEIPT\_NUMBER, customersT.CU\_GENDER, customersT.CU\_AGE\_RANGE, customersT.CUSTOMER\_TYPE, customersT.CUSTOMER\_ID

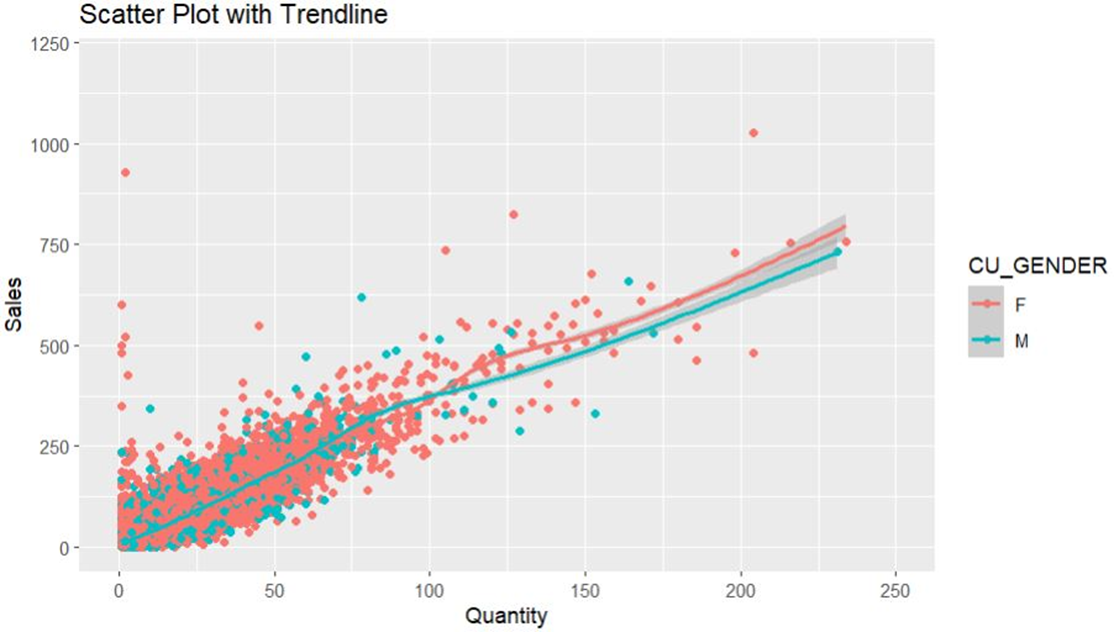
HAVING (((customersT.CUSTOMER\_TYPE)="CUSTOMER") And ((Asc(customersT.CU\_GENDER))<>32));

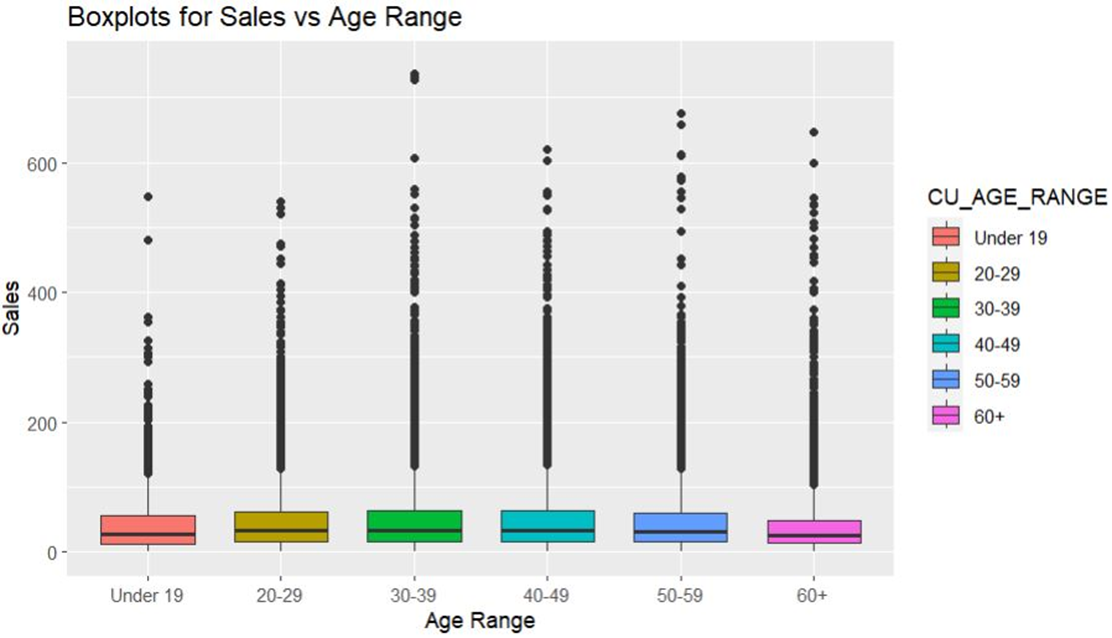
**Appendix G:**

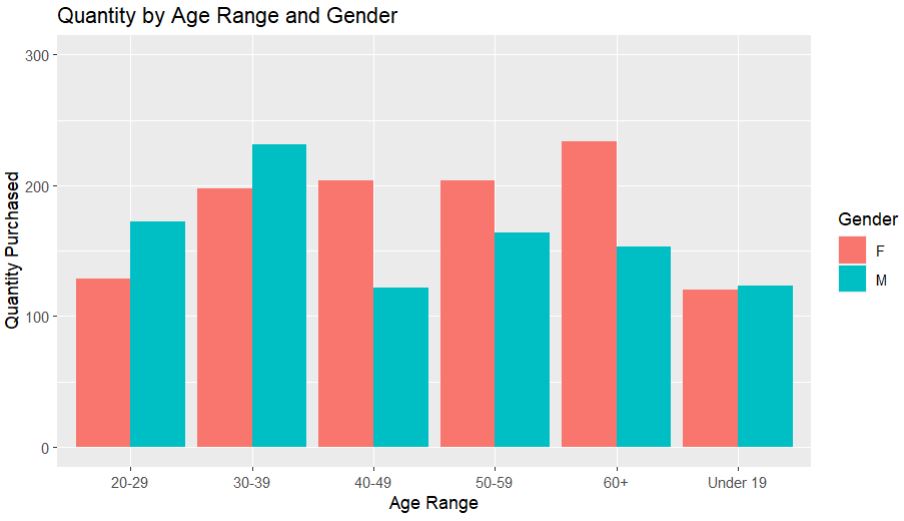
Graphs

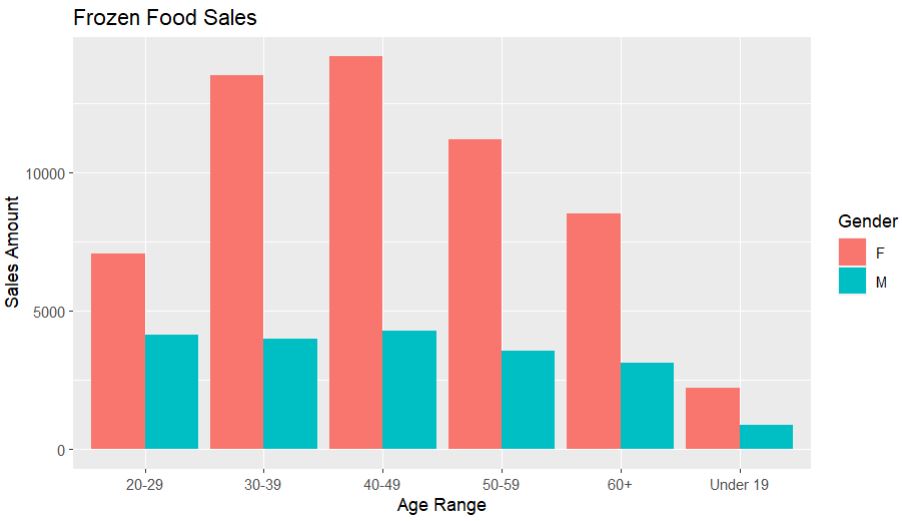
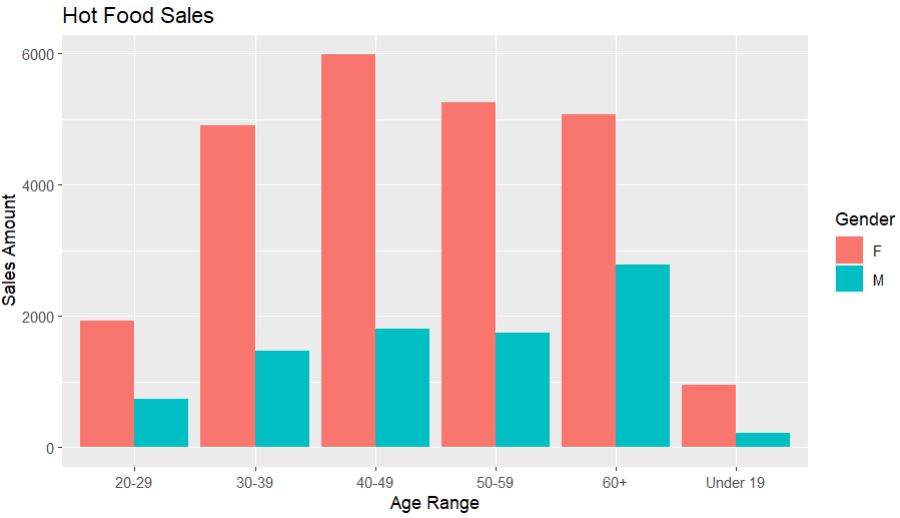












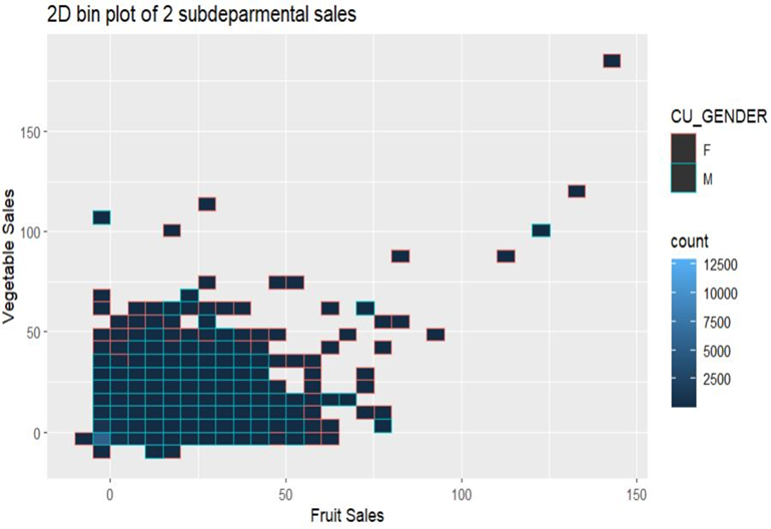
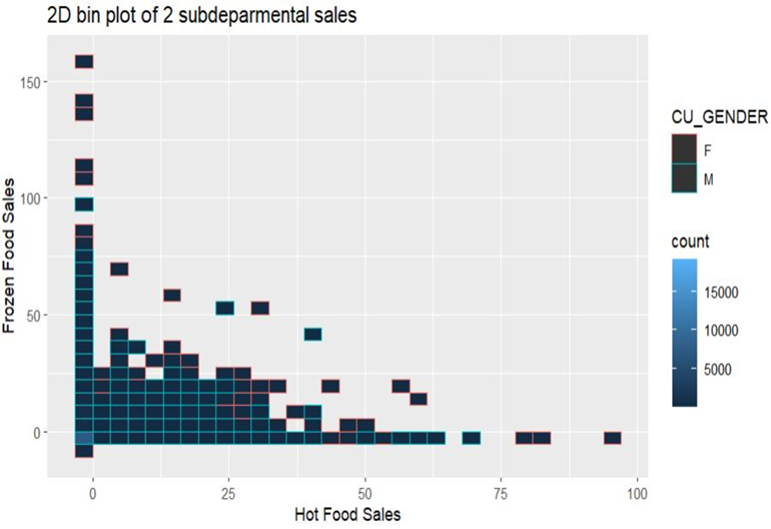
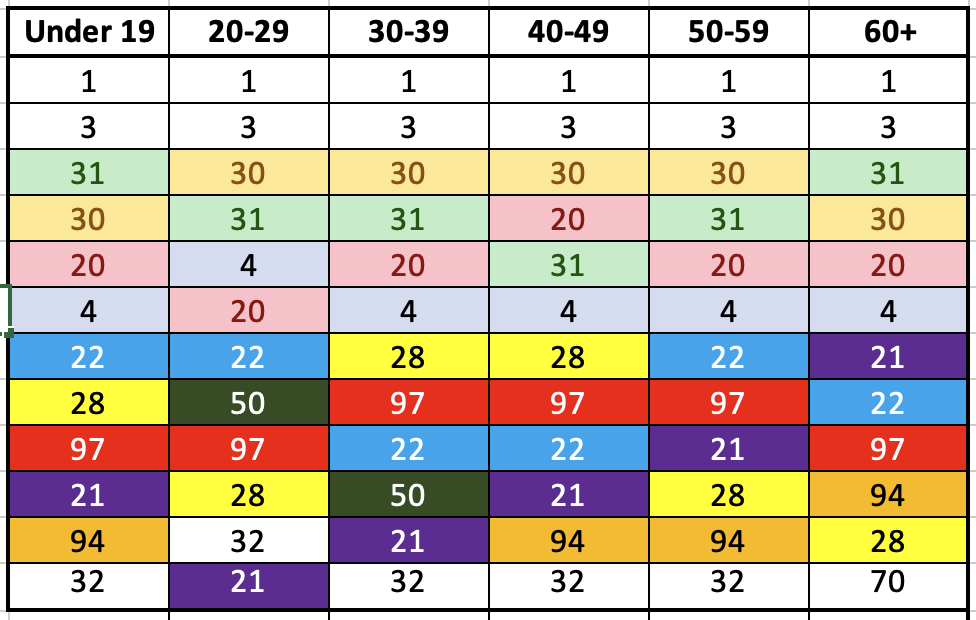


Figure 2. A Ranking of Total Sub-Department Transactions by Demographic

Figure 3. Predicting Gender using Subdepartment and Transaction Sales with the validation set. (KNN, k=1)

|  |  | **Predicted** | |
| --- | --- | --- | --- |
|  |  | **F** | **M** |
| **Actual** | **F** | 3235 | 1107 |
| **M** | 1190 | 453 |

Figure 4. Predicting Over 40 using Subdepartment and Transaction Sales with the validation set. (KNN, k=1)

|  |  | **Predicted** | |
| --- | --- | --- | --- |
|  |  | **Over 40** | **Under 40** |
| **Actual** | **Over 40** | 2414 | 1366 |
| **Under 40** | 1357 | 848 |

Figure 5. For Gender, comparison of prediction models using random numbers as input attributes with validation set. (KNN, K=1)

|  | **Sales+ SubDept** | **Random #** | **SubDept + Random #** | **Sales+**  **Random #** |
| --- | --- | --- | --- | --- |
| **Overall Error** | 38% | 40% | 38% | 39% |
| **Class 1** | 25% | 27% | 26% | 27% |
| **Class 0** | 72% | 74% | 71% | 71% |

Figure 6. For Over 40, comparison of prediction models using random numbers as input attributes with validation set. (KNN, K=1)

|  | **Sales+ SubDept** | **Random #** | **SubDept + Random #** | **Sales+**  **Random #** |
| --- | --- | --- | --- | --- |
| **Overall Error** | 45% | 46% | 46% | 46% |
| **Class 1** | 36% | 36% | 37% | 37% |
| **Class 0** | 62% | 62% | 60% | 62% |

Figure 7. For gender, comparing prediction models with various levels of k with validation set.

|  | **K = 1** | **K = 5** | **K = 10** |
| --- | --- | --- | --- |
| **Overall Error** | 38% | 33% | 31% |
| **Class 1** | 25% | 13% | 8% |
| **Class 2** | 72% | 86% | 91% |

Figure 8. For Over 40, comparing prediction models with various levels of k with validation set.

|  | **K =1** | **K =5** | **K =10** |
| --- | --- | --- | --- |
| **Overall Error** | 45% | 43% | 41% |
| **Class 1** | 36% | 30% | 24% |
| **Class 0** | 62% | 67% | 70% |

Figure 9. LDA model results for predicting gender and Over 40 with validation set.

|  | **Gender** | | **Over 40** | |
| --- | --- | --- | --- | --- |
| **Input Attribute** | **Sales +Sub Departments** | **Sales Only** | **Sub Departments** | **Sales Only** |
| **Overall Error** | 27% | 27% | 37% | 37% |
| **Class 1** | 0% | 0% | 2% | 1% |
| **Class 0** | 100% | 100% | 98% | 99% |

Figure 10. Random forest model results for predicting gender and Over 40 with validation set.

|  | **Gender** | | **Over 40** | |
| --- | --- | --- | --- | --- |
| **Input Attribute** | **Sales +Sub Departments** | **Sales Only** | **Sub Departments** | **Sales Only** |
| **Overall Error** | 28% | 35% | 39% | 44% |
| **Class 1** | 2% | 15% | 16% | 27% |
| **Class 2** | 97% | 86% | 80% | 73% |

Figure 11. KNN (k=1) model results for predicting gender and Over 40 with the test set.

|  | **Gender** | **Over 40** |
| --- | --- | --- |
| **Overall Error** | 39% | 47% |
| **Class 1** | 27% | 38% |
| **Class 2** | 71% | 61% |

**Appendix R: Rscript**

library(readr)

library(dplyr)

library(ggplot2)

library(MASS)

library(class)

library(rpart)

library(randomForest)

d<-read\_csv("salesbreakdown5.csv")

d\_clean<-d

any(is.na(d\_clean$CU\_GENDER))

any(is.na(d\_clean$CU\_AGE\_RANGE))

d\_clean$CU\_GENDER<-factor(d\_clean$CU\_GENDER)

d\_clean$CU\_AGE\_RANGE<-factor(d\_clean$CU\_AGE\_RANGE)

d\_clean<-mutate(d\_clean,over40=ifelse(d\_clean$CU\_AGE\_RANGE=="40-49" | d\_clean$CU\_AGE\_RANGE=="50-59" |d\_clean$CU\_AGE\_RANGE=="60+", "Over40", "Under40"))

d\_clean$over40<-factor(d\_clean$over40)

d\_clean <- mutate(d\_clean, RandomColumn = runif(n = nrow(d\_clean), min = 1, max = 10))

rowSample<-sample(nrow(d\_clean),nrow(d\_clean)\*2/3)

d\_clean.train<-d\_clean[rowSample,]

d\_clean.rest<-d\_clean[-rowSample,]

rowSample2<-sample(nrow(d\_clean.rest),0.5\*nrow(d\_clean.rest))

d\_clean.val<-d\_clean.rest[rowSample2,]

d\_clean.test<-d\_clean.rest[-rowSample2,]

#Gender and KNN

pred.train\_1<-knn(d\_clean.train[,c(2,3,4,17)],d\_clean.val[,c(2,3,4,17)],d\_clean.train$CU\_GENDER,k=1)

table(d\_clean.val$CU\_GENDER,pred.train\_1)

pred.train\_1<-knn(d\_clean.train[,c(2:13)],d\_clean.val[,c(2:13)],d\_clean.train$CU\_GENDER,k=10)

table(d\_clean.val$CU\_GENDER,pred.train\_1)

pred.train\_1<-knn(d\_clean.train[,c(5:13,17)],d\_clean.val[,c(5:13,17)],d\_clean.train$CU\_GENDER,k=1)

table(d\_clean.val$CU\_GENDER,pred.train\_1)

pred.train\_1<-knn(d\_clean.train[,c(17)],d\_clean.val[,c(17)],d\_clean.train$CU\_GENDER,k=1)

table(d\_clean.val$CU\_GENDER,pred.train\_1)

#Over 40 and KNN

pred.train\_1<-knn(d\_clean.train[,c(2,3,4,17)],d\_clean.val[,c(2,3,4,17)],d\_clean.train$over40,k=1)

table(d\_clean.val$over40,pred.train\_1)

pred.train\_1<-knn(d\_clean.train[,c(2:13)],d\_clean.val[,c(2:13)],d\_clean.train$over40,k=10)

table(d\_clean.val$over40,pred.train\_1)

pred.train\_1<-knn(d\_clean.train[,c(17)],d\_clean.val[,c(17)],d\_clean.train$over40,k=1)

table(d\_clean.val$over40,pred.train\_1)

#LDA and Gender

attach(d\_clean.train)

#LDA, Gender, Sales

lda.fit<-lda(CU\_GENDER~sales,data=d\_clean.train)

lda.pred<-predict(lda.fit,d\_clean.val)

lda.class<-lda.pred$class

table(d\_clean.val$CU\_GENDER,lda.class)

#LDA, Gender, Sales and Subdepartments

lda.fit<-lda(CU\_GENDER~sales+grocery+dairy+produce+meat+frozen+deli+hba+horticulture+bread,data=d\_clean.train)

lda.pred<-predict(lda.fit,d\_clean.val)

lda.class<-lda.pred$class

table(d\_clean.val$CU\_GENDER,lda.class)

#LDA with Over 40, sales

lda.fit<-lda(over40~sales,data=d\_clean.train)

lda.pred<-predict(lda.fit,d\_clean.val)

lda.class<-lda.pred$class

table(d\_clean.val$over40,lda.class)

#LDA with subdepartments and sales

lda.fit<-lda(over40~sales+grocery+dairy+produce+meat+frozen+deli+hba+horticulture+bread,data=d\_clean.train)

lda.pred<-predict(lda.fit,d\_clean.val)

lda.class<-lda.pred$class

table(d\_clean.val$over40,lda.class)

#randomForest and Gender with subdepartments

ranForest<-randomForest(CU\_GENDER~sales+grocery+dairy+produce+meat+frozen+deli+hba+horticulture+bread,data=d\_clean.train,importance=T,na.action=na.omit)

ranForest.pred <- predict(ranForest,d\_clean.val)

ranForest.pred.results <- table(d\_clean.val$CU\_Gender, ranForest.pred)

ranForest.val.error.rate <- 1-sum(diag(ranForest.pred.results))/sum(ranForest.pred.results)

ranForest.val.error.rate

#randomForest and Gender with sales only

ranForest<-randomForest(CU\_GENDER~sales,data=d\_clean.train,importance=T,na.action=na.omit)

ranForest.pred <- predict(ranForest,d\_clean.val)

ranForest.pred.results <- table(d\_clean.val$CU\_GENDER, ranForest.pred)

ranForest.val.error.rate <- 1-sum(diag(ranForest.pred.results))/sum(ranForest.pred.results)

ranForest.val.error.rate

#randomForest Age and subdepartments

ranForest<-randomForest(over40~sales+grocery+dairy+produce+meat+frozen+deli+hba+horticulture+bread,data=d\_clean.train,importance=T,na.action=na.omit)

ranForest.pred <- predict(ranForest,d\_clean.val)

ranForest.pred.results <- table(d\_clean.val$over40, ranForest.pred)

ranForest.val.error.rate <- 1-sum(diag(ranForest.pred.results))/sum(ranForest.pred.results)

ranForest.val.error.rate

#randomForest Age and sales only

ranForest<-randomForest(over40~sales,data=d\_clean.train,importance=T,na.action=na.omit)

ranForest.pred <- predict(ranForest,d\_clean.val)

ranForest.pred.results <- table(d\_clean.val$over40, ranForest.pred)

ranForest.val.error.rate <- 1-sum(diag(ranForest.pred.results))/sum(ranForest.pred.results)

ranForest.val.error.rate

#knn with test data for gender

pred.train\_1<-knn(d\_clean.train[,c(5:13,17)],d\_clean.test[,c(5:13,17)],d\_clean.train$CU\_GENDER,k=1)

table(d\_clean.test$CU\_GENDER,pred.train\_1)

#knn with test data for age

pred.train\_1<-knn(d\_clean.train[,c(5:13,17)],d\_clean.test[,c(5:13,17)],d\_clean.train$over40,k=1)

table(d\_clean.test$over40,pred.train\_1)