USE CASE STUDY REPORT

Group No.: Group 7

Student Names: Sai Srilakshmi Kanumuri and Divya Rajagopalan

Project Topic: WALMART STORE SALES PREDICTION

I. Background and Introduction

Many household products are sold by various subsidiaries of the retail store network which are geographically located at various locations. Supply chain inefficiencies will occur at different locations when the market potential is evaluated by the retailers. Many a times, it is not easy for the retailers to understand the market condition at various geographical locations. The organization of retail store network has to understand the market conditions to intensify its goods that are to be bought and sold, so that, many customers get attracted in that direction. Business prediction helps retailers in visualizing the big picture. By predicting the sales, we get a general idea of coming years if any changes are needed. Then, those changes are done in the retail store's objective, so that success is achieved more profitably. It also helps the customers to be happy by providing the products desired by them in desired time, when the customers are happy then they prefer the store that provides all the resources they need to their satisfaction. By this, the sales in the particular store in which the customers purchase more items increases, causing more profit. The prediction of sales helps to know the retailers the demand of the product. [1]

The purpose of this case study is to show how simple machine learning can predict the weekly sales of a company. Many models are powerful and flexible enough to be implemented in any industry, but in this study, we are going to be predicting sales for a retail company, Walmart, to be specific. As part of a recent recruiting effort, Walmart shared anonymized weekly sales data for 45 of its stores and asked candidates to predict the future sales.

Problem Statement:

There are three research problems that we aim to solve through this case study. They are:

Problem 1: Predict the *Type* of each Walmart Store based on the different features present in our dataset.
Problem 2: Predict the *Rank* that each Walmart store lies in with respect to their sales.
Problem 3: Predict the *Weekly Sales* value of each Walmart store.

As you can see, we are trying to predict the *Type* and *Rank* of each store, which means that research problems 1 and 2 are classification problems. And, we are trying to predict the *Weekly Sales* value through research problem 3, so it is a regression problem. All these three research problems follow supervised machine learning approach.

Goal:

In order to solve our above formulated research problems, the following goals are set:

- **Goal 1:** Predict the *Type* of each Walmart store using features from our dataset. So, the output variable (y) here is *Type* whereas the input variables (x's) are *Weekly Sales* and *Size*. We will apply *Decision tree* model on our data to accomplish this goal.
- **Goal 2:** Predict the *Rank* of each Walmart store using features from our dataset. So, the output variable (y) here is *Rank* whereas input variables (x's) are *Weekly Sales* and *Type*. We will apply *Decision tree* model on our data to accomplish this goal.
- **Goal 3:** Predict the *Weekly Sales* of each Walmart store using features from training dataset. So, the output variable (y) here is *Weekly Sales* whereas input variables (x's) are features from training dataset. We will apply *Linear Regression* model on our data to accomplish this goal.

Possible Solution:

- The first solution that will be obtained through this case study is the predictions for which *Type* of stores Walmart should invest in. This result can help the company categorize new stores and therefore predict how much they should sell based on the Type of store they are grouped into.
- The second solution that can be obtained is the predictions for *Rank* of the stores. This will help Walmart to determine within which range a certain store should sell in a given week of the year.
- The third solution that can be obtained is the predictions for *Weekly Sales* for each Walmart Store. This result will help Walmart to better stock their products in each department and enjoy profits.

II. Data Preparation and Preprocessing

Historic sales data provided by Walmart is used to perform our case study. The data contains information about 45 Walmart stores located in different regions. In addition, Walmart runs several promotional markdown events throughout the year. These markdowns precede prominent holidays, the four largest of which are the Super Bowl, Labor Day, Thanksgiving and Christmas. The weeks including these holidays are weighted five times higher in the evaluation than non-holiday weeks.

We had access to three different data sets from *Kaggle.com* about Walmart. These data sets contained information about the stores, departments, temperature, unemployment etc. We will explain each one of the data sets in more detail with each one of its features.

Stores.csv

- Store: The store number. Range from 1-45.
- Type: Three types of stores 'A', 'B' or 'C'.

- Size: Sets the size of a Store would be calculated by the no. of products available in the particular store ranging from 34,000 to 210,000.

Train.csv

- Date: The date of the week where this observation was taken.
- Weekly_Sales: The sales recorded during that Week.
- Store: The store which observation in recorded 1-45.
- Dept: One of 1-99 that shows the department.
- IsHoliday: Boolean value representing a holiday week or not.

Features.csv

- Temperature: Temperature of the region during that week.
- Fuel_Price: Fuel Price in that region during that week.
- MarkDown1-5: Represents the Type of markdown and what quantity was available during that week.
- CPI: Consumer Price Index during that week.
- Unemployment: The unemployment rate during that week in the region of the store.

Our first step is to join our train.csv and stores.csv by Store which is the common column.

Next we conducted some feature engineering on our data. We used the features that our data currently has but we tweaked them in a way that made our analysis easier. The most important objective in this step was to generate new features that will help us produce a better model. For this, we included *Week Number of the Year* to our dataset.

We have also noticed that some Weekly Sales contain negative values, after analyzing the data we have concluded that those refer to Returned Products from previous weeks. So, we added a *Returns* column to our data.

Initially, our data frame consisted of about 421570 observations. Since the objective of this model is to predict the *Weekly Sales* of a particular store given previous years, external information and tendency we will add the sales per department and put it together into one observation. In other words, we will not subdivide sales by department. Thus, we can make our Weekly Sales to be our Net Sales since we now can do *Weekly_Sales - Returns* to avoid negative values. For this we defined an *aggregate* function to aggregate Weekly Sales to Net Sales.

After performing this procedure, we ended up with 6435 observations, which made our data more manageable for further analysis.

This is how the aggregated data looks like:

- *head(train)*

	Store	*Dept*	*Date*	*Weekly_Sales*	*IsHoliday*	*Type*	*Size*
1	*1*	*1*	*2010-02-05*	*24924.50*	*FALSE*	*A*	*151315*
2	*1*	*1*	*2010-02-12*	*46039.49*	*TRUE*	*A*	*151315*
3	*1*	*1*	*2010-02-19*	*41595.55*	*FALSE*	*A*	*151315*

4 1 1 2010-02-26 19403.54 FALSE A 151315

5 1 1 2010-03-05 21827.90 FALSE A 151315

6 1 1 2010-03-12 21043.39 FALSE A 151315

Our next step is to merge the above aggregated data with features.csv by using *left_join* function.

After merging the aggregated data with features, we marked all the *NAs* in our data with *0*.

We will also add a feature called *Rank*, which is getting the range of values of Weekly Sales. We will make five Range Buckets namely A, B, C, D and E. We will also try to predict in which of this buckets a given store would lie in a given week.

III. Data Exploration and Visualization

Data Review:

Here is how our data looks like:

- *head(train)*

```
  Store       Date Weekly_Sales IsHoliday Type WeekNum   Size Temperature Fuel_Price MarkDown1 MarkDown2
1     1 2010-02-05    1643691     FALSE    A       6 151315       42.31      2.572         0         0
2     1 2010-02-12    1641957      TRUE    A       7 151315       38.51      2.548         0         0
3     1 2010-02-19    1613694     FALSE    A       8 151315       39.93      2.514         0         0
4     1 2010-02-26    1409728     FALSE    A       9 151315       46.63      2.561         0         0
5     1 2010-03-05    1554807     FALSE    A      10 151315       46.50      2.625         0         0
6     1 2010-03-12    1440938     FALSE    A      11 151315       57.79      2.667         0         0
  MarkDown3 MarkDown4 MarkDown5      CPI Unemployment Rank
1         0         0         0 211.0964        8.106    B
2         0         0         0 211.2422        8.106    B
3         0         0         0 211.2891        8.106    B
4         0         0         0 211.3196        8.106    B
5         0         0         0 211.3501        8.106    B
6         0         0         0 211.3806        8.106    B
```

For our data exploration, we started with the *aggregate ()* function because we wanted to know which Store and Type of store was having the most sales, on average.

- *head(aggregate(train[,"Weekly_Sales"], by=train[,c("Store"), drop=FALSE], mean))*

```
  Store       x
1     1 1555319.8
2     2 1925817.3
3     3  402721.1
```

4 4 2094731.4
5 5 318017.8
6 6 1564762.6

- *head(aggregate(train[,"Weekly_Sales"], by=train[,c("Type"), drop=FALSE], mean))*

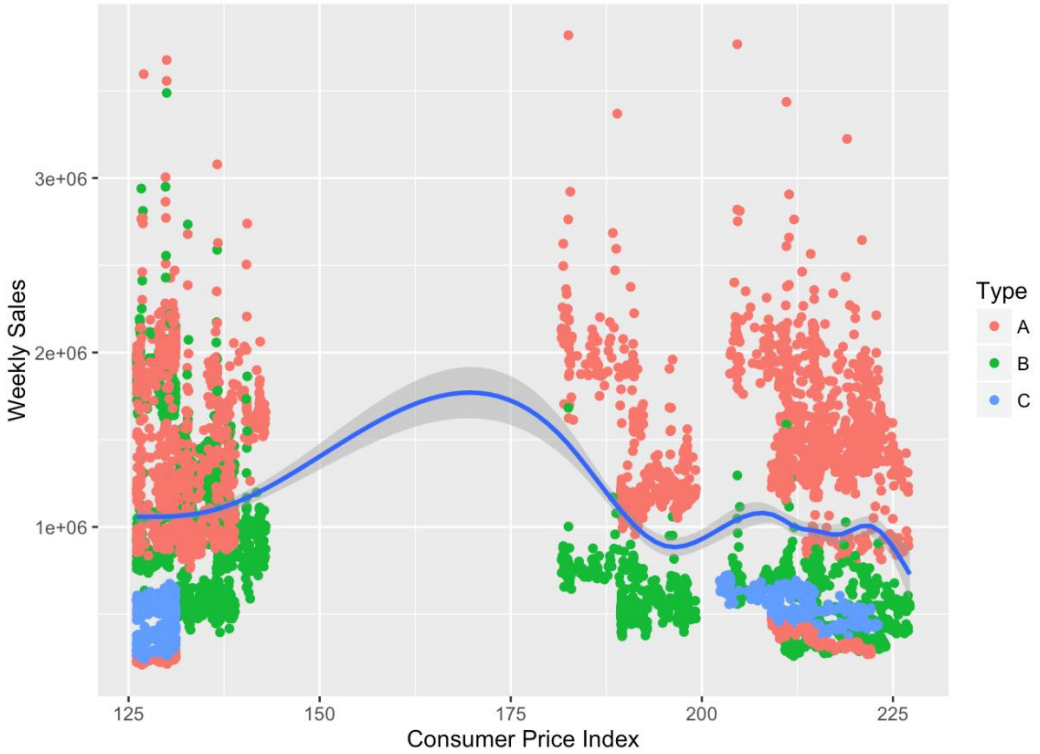
Type x
1 A 1376700.6
2 B 823028.6
3 C 472625.3

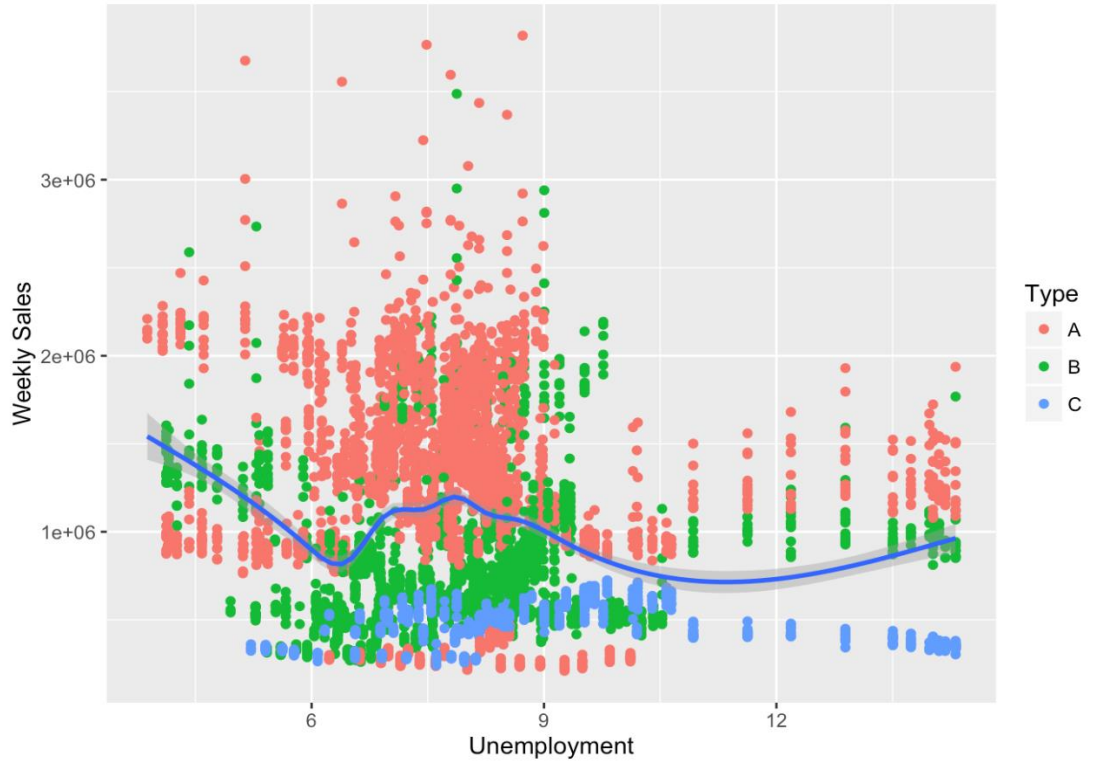
- *head(aggregate(train[,"Weekly_Sales"], by=train[,c("Type"), drop=FALSE], max))*

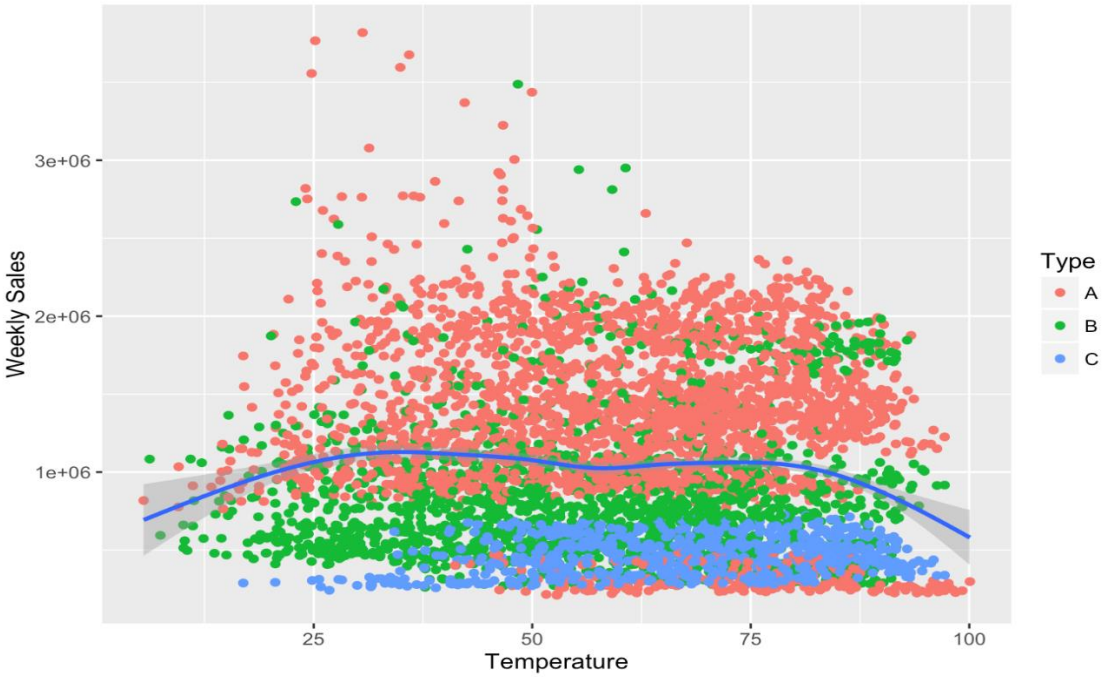
Type x
1 A 3818686
2 B 3749058
3 C 725043

With this initial information, we wanted to dig a little deeper and that is why we decided that graphic models will help us to find the interaction between each of the variables with Weekly Sales. Our goal with this exploration was to find correlation, patterns or any other insight that revealed more information between diving into our predictive model.

Before proceeding with our data exploration, we partitioned the data set into two different data frames (training and test) in order to keep our analysis consistent and avoid testing on our training data.

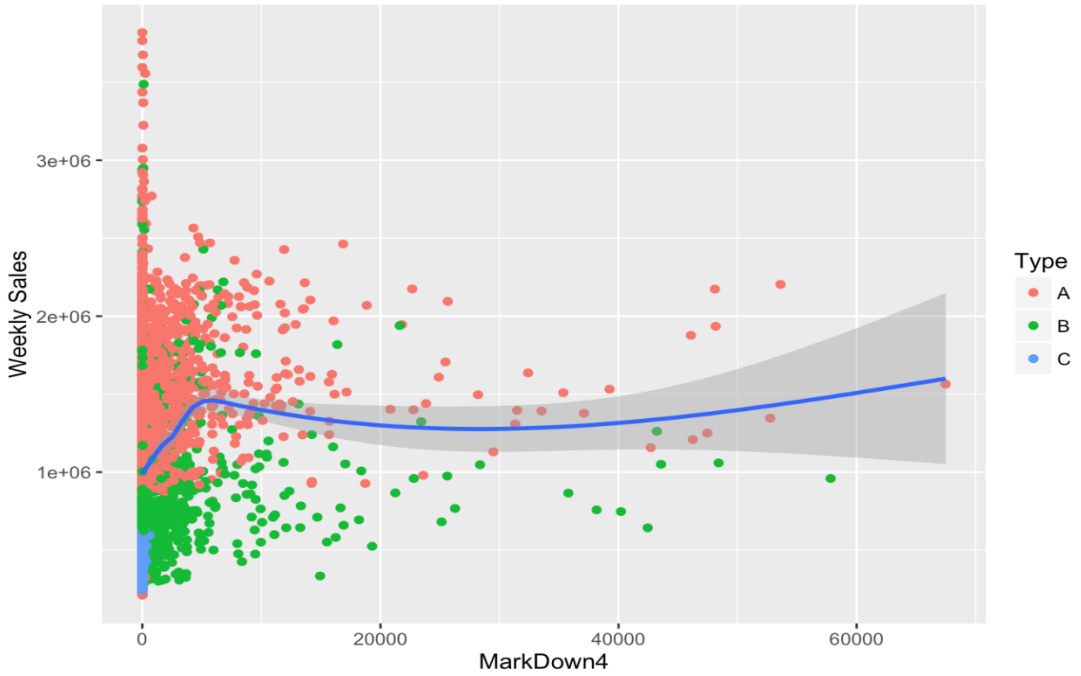




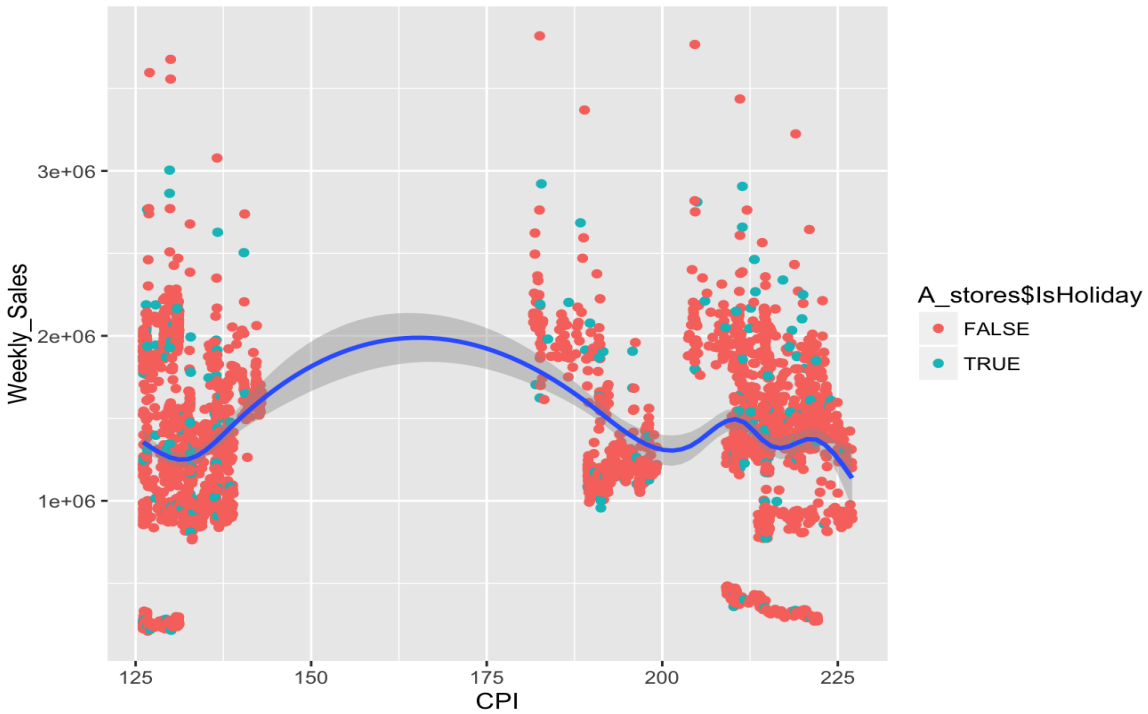


In the previous graphs one can see that there is no clear correlation between *Weekly Sales* and *Unemployment* or *Temperature*. A clearer correlation is visible between *CPI* and *Weekly Sales.* However, what is clear from this analysis is that *Type A* stores have more sales than any other type.

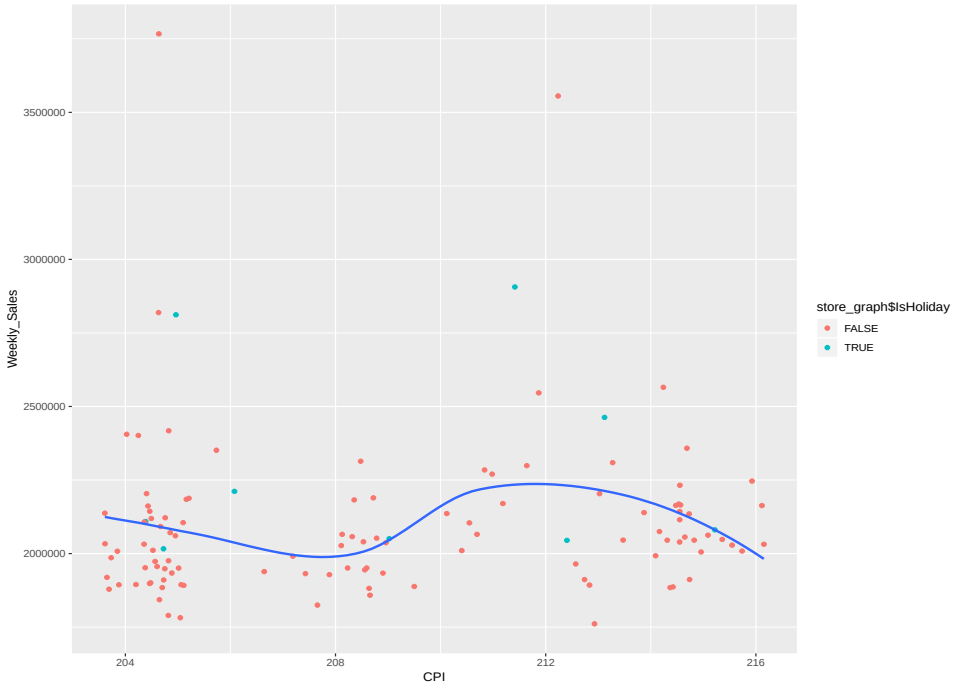
We also want to analyze what is the effect of the *MarkDowns* on the weekly sales of the company after analyzing the graphs we decided to show the one that had more impact. However, as one can see *the MarkDowns* don't show an immense correlation with Sales.



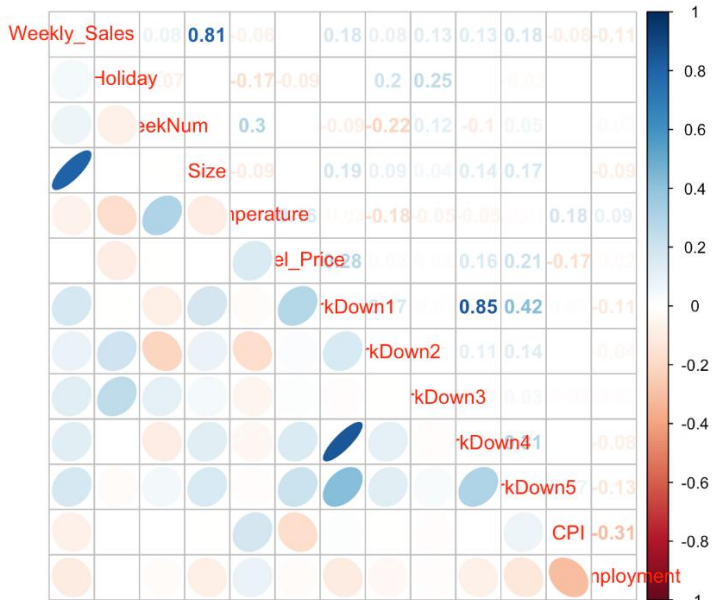
Plot Store sales divided by Type of Store, in the following plot we selected *Type A* stores.



Now, we want to partition the *Weekly Sales* based on a store. From our analysis we saw that Store 20 is the one with the most sales. We will analyze this stores' results.



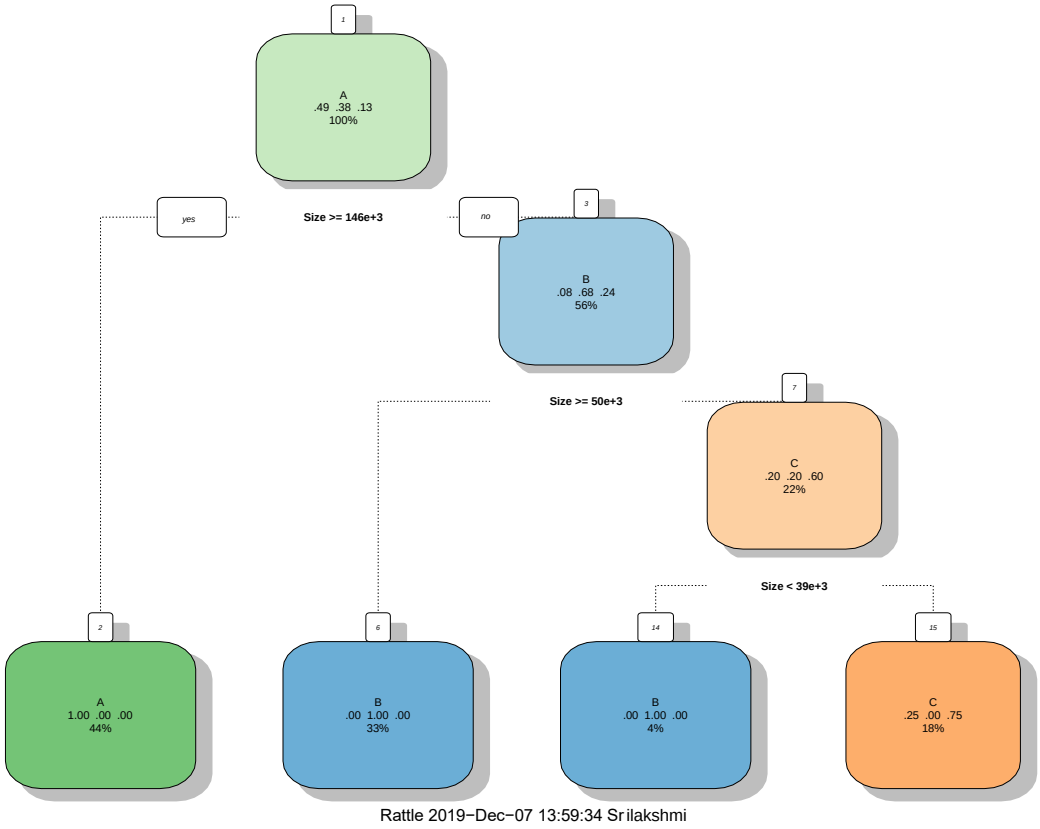
Finally, look for a correlation matrix between all of our numerical features.



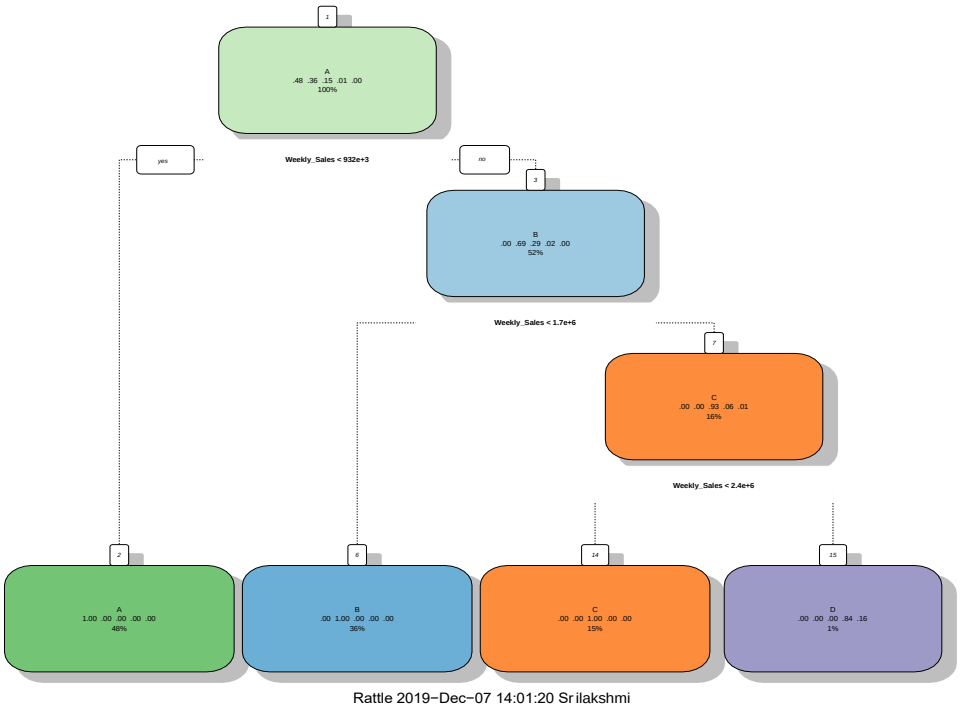
Store *Type* is very important to predict the *Weekly Sales* of a given store. We will run a Decision Tree model to predict what *Type* a Store should be based on the different features that we have on our model.

IV. Data Mining Techniques and Implementation

Since we saw that the Store type is very important to predict the Weekly Sales of a given store, we will run a Decision Tree model to predict what Type a Store should be based on the different features that we have on our model. Using a *Decision Tree* we would like to predict the *Type* of a store based on all the other parameters.



Now we will also like to form a *Decision Tree* for predicting the *Rank* that each store lies with respect to their sales. We want to look at the other features in order to predict what range of sales a store will have in the future taking into account anything but previous sales.



Next, we would also like to create a linear model to find a specific value for *Weekly Sales* that we want to predict. We fit the model with confidence interval of 95%. This line of best fit is intended to approximate further data points based on the line that we find in our training data. We predict the linear regression model for *Weekly Sales* using lm() function and find the corresponding equation of *Weekly Sales* as follows:

Weekly_sales= -1.322e+06 + (Store2 * -8.116e+03) + (Store3 * -8.934e+05) + (Store4 * 1.190e+05) + (Store5 * -9.892e+05) + (Store6 * -2.299e+04) + (Store7 * -7.390e+05) + (Store8 * -4.537e+05) + (Store9 * -7.660e+05) + (Store10 * -1.407e+04) + (Store11 * -1.536e+05) + (Store12 * -3.912e+05) + (Store13 * 5.048e+04) + (Store14 * 1.075+05) + (Store15 * -6.784e+05) + (Store16 * -8.039e+05) + (Store17 * -4.757e+05) + (Store18 * -3.903e+05) + (Store19 * -8.081e+04) + (Store20 * 1.273e+05) + (Store21 * -5.520e+05) +(Store22 * -4.341e+05) + (Store23 * -1.666e+05) + (Store24 * -1.547e+05) + (Store25 * -6.037e+05) + (Store26 * -4.491e+05) + (Store27 * 2.724e+03) + (Store28 * -1.408e+05) +(Store29 * -7.446e+05) + (Store30 * -8.547e+05) + (Store31 * -1.145e+05) + (Store32 * -3.301e+05) + (Store33 * -1.015e+06) + (Store34 * -4.482e+05) + (Store35 * -4.313e+05) + (Store36 * -9.190e+05) + (Store37 * -7.735e+05) + (Store38 * -8.556e+05) + (Store39 * -8.724e+04) + (Store40 * -4.789e+05) + (Store41 * -2.490e+05) + (Store42 * -7.261e+05) + (Store43 * -6.385e+05) + (Store44 * -1.002e+06) + (Store45 * -5.154e+05) + (Date * -3.784e+00) + (IsHolidayTRUE * 1.838e+04) + (WeekNum * 1.294e+03) + (Temperature * -3.888e+02) + (Fuel_Price * -7.942e+03) + (MarkDown1 * 8.325e-01) + (MarkDown2 * -8.128e-01) + (MarkDown3 * 2.903e+00) + (MarkDown4 * -4.926e-01) + (MarkDown5 * 1.776e+00) + (Unemployment * -9.079e+03) + (RankB * 1.973e+05) + (RankC * 5.972e+05) + (RankD * 1.231e+06) + (RankE * 2.177e+06

V. Performance Evaluation

We have done performance evaluation for all the data mining techniques implemented for the three problems in our case study.

Problem 1: Predict the *Type* of each Walmart Store based on the different features present in our dataset.

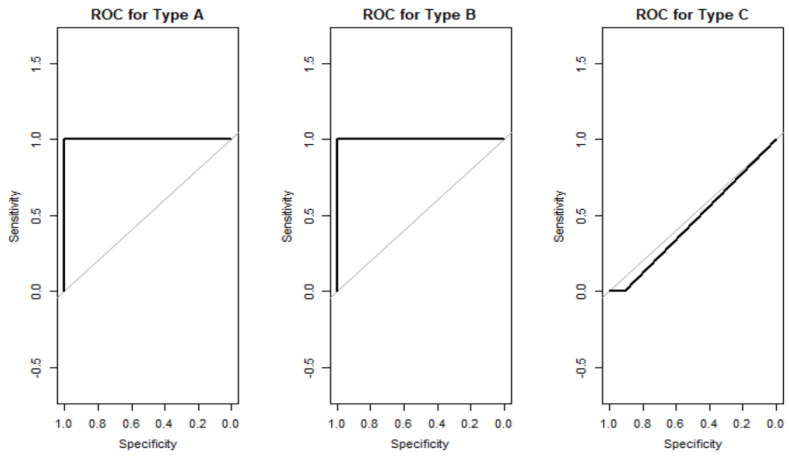
Classification Accuracy Measures for Decision Tree model of *Type*:

a) Confusion Matrix for Type

P value (2.2e-16) for the decision tree model is less than the significance level. So we have a statistically significant model. The Confusion matrix for *Type* shows a high accuracy of 95.17%, indicating that *Decision Tree* is a good model for predicting the *Type*.

```
Confusion Matrix and Statistics

          predictedclass
actualclass   A   B   C
          A 574   0   0
          B   0 479   0
          C  62   0 169

Overall Statistics

               Accuracy : 0.9517
                 95% CI : (0.9385, 0.9628)
    No Information Rate : 0.4953
    P-Value [Acc > NIR] : < 2.2e-16

                  Kappa : 0.9216

 Mcnemar's Test P-Value : NA

Statistics by Class:

                     Class: A Class: B Class: C
Sensitivity            0.9025   1.0000   1.0000
Specificity            1.0000   1.0000   0.9444
Pos Pred Value         1.0000   1.0000   0.7316
Neg Pred Value         0.9127   1.0000   1.0000
Prevalence             0.4953   0.3731   0.1316
Detection Rate         0.4470   0.3731   0.1316
Detection Prevalence   0.4470   0.3731   0.1799
Balanced Accuracy      0.9513   1.0000   0.9722
```

b) ROC for *Type*



ROC for the three Types of store indicates that *Decision Tree* is a good model for predicting the *Type*.

Problem 2: Predict the *Rank* that each Walmart store lies in with respect to their sales.

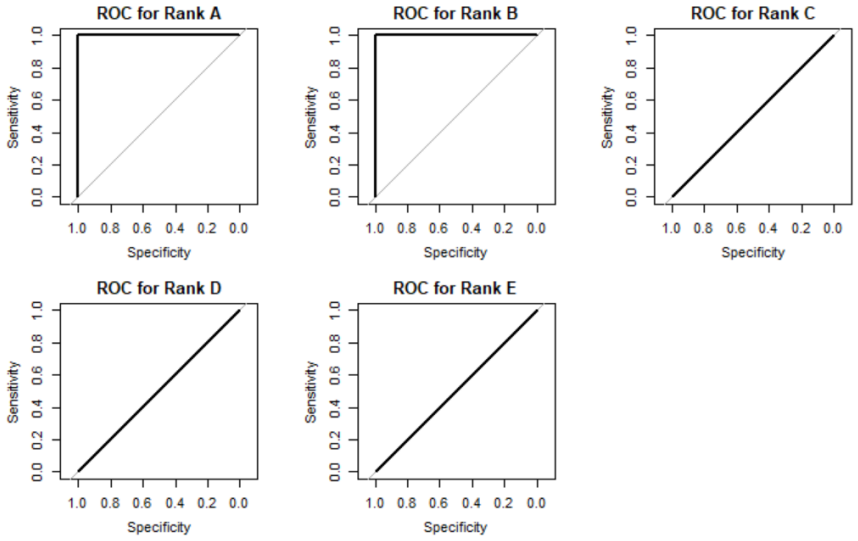
Classification Accuracy Measures for Decision Tree model of *Rank*:

a) Confusion Matrix for *Rank*

```
            predictedclass
actualclass   A   B   C   D   E
          A 608   0   0   0   0
          B   1 452   0   0   0
          C   0   0 203   1   0
          D   0   0   0  15   4
          E   0   0   0   0   0

Overall Statistics

               Accuracy : 0.9953
                 95% CI : (0.9899, 0.9983)
    No Information Rate : 0.4743
    P-Value [Acc > NIR] : < 2.2e-16

                  Kappa : 0.9925

 Mcnemar's Test P-Value : NA

Statistics by Class:

                     Class: A Class: B Class: C Class: D Class: E
Sensitivity            0.9984   1.0000   1.0000  0.93750 0.000000
Specificity            1.0000   0.9988   0.9991  0.99685 1.000000
Pos Pred Value         1.0000   0.9978   0.9951  0.78947      NaN
Neg Pred Value         0.9985   1.0000   1.0000  0.99921 0.996885
Prevalence             0.4743   0.3520   0.1581  0.01246 0.003115
Detection Rate         0.4735   0.3520   0.1581  0.01168 0.000000
Detection Prevalence   0.4735   0.3528   0.1589  0.01480 0.000000
Balanced Accuracy      0.9992   0.9994   0.9995  0.96717 0.500000
```

P value (2.2e-16) for the decision tree model is less than the significance level. So we have a statistically significant model. The Confusion matrix for *Rank* shows a high accuracy of 99.53%, indicating that decision tree is a good model for predicting the *Rank*.

b) ROC for *Rank*



ROC for the five Ranks indicates that *Decision Tree* is a good model for predicting the *Rank*.

Problem 3: Predict the *Weekly Sales* value of each Walmart store.

Prediction Accuracy Measures for Linear Regression model of *Weekly Sales*:

a) Correlation Accuracy Calculation

	actuals <dbl>	predicteds <dbl>
1	1404430	1462900
2	1508238	1495564
3	1513080	1487542
4	1507461	1527391
5	1430379	1499165
6	1459409	1511241

Here we can see that correlation between the actuals and predicteds values is high, indicating a good linear prediction model.

b) Min-Max Accuracy Calculation

The Min-Max Accuracy is high (93.23%), indicating a good linear prediction model.

c) MAE (Mean Absolute Error) Calculation

The value of MAE is 69846.7.

d) RMSE (Root Mean Square Error) Calculation

The value of RMSE is 97003.42.

e) MAPE (Mean Absolute Percentage Error) Calculation

The MAPE is low(approximately 7%), indicating a good linear prediction model.

f) R-squared Calculation

R-Squared value is high(approximately 97%), indicating a good linear prediction model.

VI. Discussion and Recommendation

In our case study, to approach the three problems or goals, we have implemented data mining techniques such as *Decision Trees* for prediction of *Type* of store and *Rank* of store, and *Linear Regression Model* for prediction of *Weekly Sales*.

Based on the performance evaluation of our three models, we have come to a conclusion that:

a) *Decision Trees* was a good model for predicting the *Type* of store. The Confusion matrix for *Type* shows a high accuracy of 95.17%.
b) *Decision Trees* was a good model for predicting the *Rank* of store. The Confusion matrix for *Rank* shows a high accuracy of 99.53%.
c) *Linear Regression Model* was a good model for prediction of *Weekly Sales*. All the performance accuracy measures calculated above for Weekly Sales show results that are favourable for a good Linear Regression model.

Thus, we would recommend using Decision Trees for classification problems and Linear Regression models for regression problems. However, Decision Trees and Linear Regression models have their own advantages and shortcomings. Thus, we need to take all the factors into consideration when choosing a model for prediction or classification.

The advantage of using Decision Trees is that they require less effort for data preparation during pre-processing. They also do not require normalization or scaling of data as well. Missing values in the data also does not affect the process of building decision tree. However, the shortcomings of using Decision Trees is that a small change in the data can cause a large change in the structure of the decision tree causing instability. Decision trees often involve higher time to train the model. Decision tree training is relatively expensive as complexity and time taken is more.

The advantage of using Linear Regression models is that they are easy to implement, interpret and very efficient to train. Linear Regression performs well when the dataset is linearly separable. However, Linear Regression models are prone to overfitting, outliers and multicollinearity, and thus we must get rid of these when implementing Linear Regression.

VII. Summary

The purpose of this case study 'Walmart Sales Forecasting' is to show how simple machine learning can be used to predict the weekly sales of a company. Walmart shared weekly sales data for 45 of its stores and asked candidates to predict the future sales.

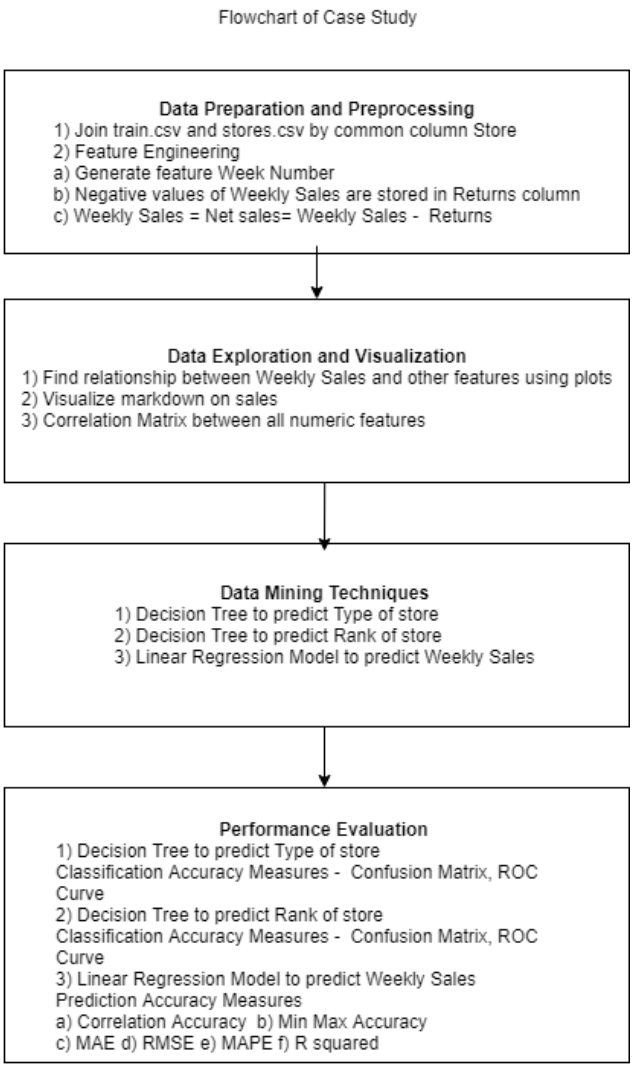
We decided to divide our case study into three research problems that we aim to solve through this case study. They are:

Problem 1: Predict the *Type* of each Walmart Store based on the different features present in our dataset.
Problem 2: Predict the *Rank* that each Walmart store lies in with respect to their sales.
Problem 3: Predict the *Weekly Sales* value of each Walmart store.

We tried to predict the *Type* and *Rank* of each store, which means that research problems 1 and 2 are classification problems. Thus, for problems 1 and 2, we implemented *Decision Trees*. We tried to predict the *Weekly Sales* value through research problem 3, so it is a regression problem. Thus, for problem 3, we implemented a *Linear Regression* model. All these three research problems follow supervised machine learning approach.

After implementing the data mining techniques for these models, we have done performance evaluation for all the data mining techniques implemented for the three problems in our case study. We performed classification accuracy measures for the Decision Trees models for Problems 1 and 2 to check the accuracy of the predicted *Type* and *Rank* of each store. Then, we performed prediction accuracy measures for the *Linear Regression* model for Problem 3 to check the accuracy of the predicted *Weekly Sales* value. All the accuracy measures calculated for the 3 problems showed results that are favorable for a good Decision Tree or Linear Regression model (as shown above).

Thus, we have accomplished the goal or purpose of conducting this case study.



REFERENCES

1) https://pdfs.semanticscholar.org/ed03/e6a1f4fb6fe0b185938e45ef69dc64b4bb69.pdf

Appendix: R Code for use case study

```
#Walmart Predictive Analysis
#Load Required Libraries
library(dplyr)
library(ggplot2)
library(reshape2)
library(readr)
library(lubridate)
library(rpart)
library(rattle)
library(car)
library(caret)
library(corrplot)
library(rpart.plot)

# Ensure that you have train.csv, stores.csv and features.csv in your current working directory.

# Loading files to work with
data <- read.csv("/Users/Srilakshmi/Downloads/walmart-recruiting-store-sales-
forecasting/train.csv")
stores <- read.csv("/Users/Srilakshmi/Downloads/walmart-recruiting-store-sales-
forecasting/stores.csv")
features <- read.csv("/Users/Srilakshmi/Downloads/walmart-recruiting-store-sales-
forecasting/features.csv")

# Our first step will be to join our two tables by Store which is the common column.
stores$Store <- factor(stores$Store)
data$Store <- factor(data$Store)
data <- full_join(data,stores,by=c("Store"))

# Preparation
# In this step of the process we will conduct some feature engineering, we will use the
# features that our data currently has but will tweak them in a way that makes our analysis easier.
# The most important objective in this step is to generate new features that will help us produce a
better model
data$WeekNum <- lubridate::week(data$Date)

# We have also noticed that some Weekly Sales contain negative values,
# after analyzing the data we have concluded that those refer to Returned Products from previous
weeks.

# Add a Returns Column
data$Returns <- lapply(data$Weekly_Sales,function(sales){
```

```
  ifelse(sales < 0,sales,0)
})
data$Weekly_Sales <- lapply(data$Weekly_Sales,function(sales){
  ifelse(sales > 0,sales,0)
})

# Now, our data frame contains 421570 observations, since the objective of this model is to
# predict the Weekly Sales of a particular store given previous years,
# external information and tendency we will add the sales per department and
# put it together into one observation. In other words we will not subdivide sales by department.
# Thus we can make our Weekly Sales to be our Net Sales since we now can do Weekly_Sales -
Returns to
# avoid negative values.

# Aggregating Weekly Sales to Net Sales
final_data <- data.frame(Store=factor(),Size = numeric(),
Date=as.Date(character()),Weekly_Sales=numeric(),IsHoliday=logical(),Type=factor(),WeekNu
m=factor())
aggregate_sales <- function(){
  for(i in 1:45){
    store_data <- data %>% filter(Store == i)
    dates <- unique(data$Date)
    for(next_date in seq_along(dates)){
      current_date <- unique(data$Date)[[next_date]]
      date_data <- store_data %>% filter(Date==current_date)
      #Add all the weekly sales
      net_sales <- sum(unlist(date_data$Weekly_Sales)) - sum(unlist(date_data$Returns))
      #Construct the data frame and append it
      next_row <-
data.frame(Store=i,Date=current_date,Weekly_Sales=net_sales,IsHoliday=date_data$IsHoliday[
[1]],Type=date_data$Type[[1]],WeekNum=date_data$WeekNum[[1]],Size=date_data$Size[[1]])
      next_row$Store <- factor(next_row$Store)
      final_data <- rbind(final_data,next_row)
    }
  }
  return(final_data)
}
##Sum the sales by store without taking into account each department
final_data <- aggregate_sales()

# Load the aggregated data
data_new <- read_csv("/Users/Srilakshmi/Desktop/merged3.csv")
data_new$Weekly_Sales <- as.numeric(data_new$Weekly_Sales)
data_new$Store <- factor(data_new$Store)
data_new$Type <- factor(data_new$Type)
head(data_new)
```

```
# After performing this procedure we now have 6435 observations which makes our
# data more manageable for further analysis.

features$Store <- factor(features$Store)
features$Date <- as.Date(features$Date)
data_new$Date <- as.Date(data_new$Date)
#Merge our final_data with our features
data_new <- left_join(data_new,features,by=c("Store","Date","IsHoliday"))

# Make the NA markdown as 0
data_new$MarkDown1 <- sapply(data_new$MarkDown1, function(value){
  ifelse(is.na(value),0,value)
})
data_new$MarkDown2 <- sapply(data_new$MarkDown2, function(value){
  ifelse(is.na(value),0,value)
})
data_new$MarkDown3 <- sapply(data_new$MarkDown3, function(value){
  ifelse(is.na(value),0,value)
})
data_new$MarkDown4 <- sapply(data_new$MarkDown4, function(value){
  ifelse(is.na(value),0,value)
})
data_new$MarkDown5 <- sapply(data_new$MarkDown5, function(value){
  ifelse(is.na(value),0,value)
})
data_new$CPI <- sapply(data_new$CPI, function(value){
  ifelse(is.na(value),0,value)
})
data_new$Unemployment <- sapply(data_new$Unemployment, function(value){
  ifelse(is.na(value),0,value)
})

# Rank
# We will also add a feature called rank,
# which is getting the range of values of Weekly Sales.
# We will make five Range Buckets namely A,B,C,D and E.
# We will also try to predict in which of this buckets a given store would lie in a given week.

# Range Weekly Sales: Divide our sales into five different groups
range_sales <- range(data_new$Weekly_Sales)
range <- (range_sales[[2]] - range_sales[[1]]) / 5
first <- c(range_sales[[1]], range_sales[[1]] + range)
second <- c(range_sales[[1]] + range, range_sales[[1]] + 2*range)
third <-c(range_sales[[1]] + 2*range, range_sales[[1]] + 3*range)
fourth <- c(range_sales[[1]] + 3*range, range_sales[[1]] + 4*range)
fifth <- c(range_sales[[1]] + 4*range, range_sales[[2]])
```

```
data_new$Rank <- sapply(data_new$Weekly_Sales, function(sales){
  if(sales >= first[[1]] &  sales <= first[[2]]){
    return('A')
  }
  else if(sales >= second[[1]] & sales <= second[[2]]){
    return('B')
  }
  else if(sales >= third[[1]] & sales <= third[[2]]){
    return('C')
  }
  else if(sales >= fourth[[1]] & sales <= fourth[[2]]){
    return('D')
  }
  else{
    return('E')
  }
})
data_new$Rank <- factor(data_new$Rank)

# Explortory Analysis
# Data Review
# This is how our data looks like.
head(data_new)

# For our exploration analysis we started with the aggregate() function because we wanted to
# know which Store and Type of store was having the most sales, on average.
aggregate(data_new[,"Weekly_Sales"], by=data_new[,c("Store"), drop=FALSE], mean)

aggregate(data_new[,"Weekly_Sales"], by=data_new[,c("Type"), drop=FALSE], mean)

aggregate(data_new[,"Weekly_Sales"], by=data_new[,c("Type"), drop=FALSE], max)


# With this initial information, we wanted to dig a little deeper and that is
# why we decided that graphic models will help us to find the interaction between
# each of the variables with Weekly Sales. Our goal with this exploration was to
# find correlation, patterns or any other insight that revealed more information between
# diving into our predictive model.

#Subset our data into train and test

index <- createDataPartition(data_new$Weekly_Sales,list = FALSE,p=0.8)
train <-data_new[index,]
valid <- data_new[-index,]
```

```
ggplot(train,aes(x=CPI,y=Weekly_Sales)) + geom_point(aes(color=train$Type)) +
geom_smooth() + scale_x_continuous(name="Consumer Price Index") +
scale_y_continuous(name="Weekly Sales") + scale_color_discrete(name="Type")

ggplot(train,aes(x=Unemployment,y=Weekly_Sales)) + geom_point(aes(color=train$Type)) +
geom_smooth()  + scale_x_continuous(name="Unemployment") +
scale_y_continuous(name="Weekly Sales") + scale_color_discrete(name="Type")

ggplot(train,aes(x=Temperature,y=Weekly_Sales)) + geom_point(aes(color=train$Type)) +
geom_smooth()  + scale_x_continuous(name="Temperature") +
scale_y_continuous(name="Weekly Sales") + scale_color_discrete(name="Type")

# In the previous graphs one can see that there is no clear correlation between Weekly Sales and
# Unemployment or Temperature. A clearer correlation is visible between CPI and Weekly
Sales.
# However what is clear from this analysis is that Type A stores have more sales than any other
type.

# We also want to analyze what is the effect of the MarkDowns on the weekly sales of the
company
# after analyzing the graphs we decided to show the one that had more impact. However, as one
can see the MarkDowns don't show an immense correlation with Sales.

ggplot(train,aes(x=MarkDown4,y=Weekly_Sales)) + geom_point(aes(color=train$Type)) +
geom_smooth()  + scale_x_continuous(name="MarkDown4") +
scale_y_continuous(name="Weekly Sales") + scale_color_discrete(name="Type")

# Plot Store sales divided by Type of Store, in the following plot we selected Type A stores
A_stores <- train %>% filter(Type=='A')
ggplot(A_stores,aes(x=CPI,y=Weekly_Sales)) + geom_point(aes(color=A_stores$IsHoliday)) +
geom_smooth()#Sales vary depending on the weeknum we are in

# Now we want to partition the Weekly Sales based on a store,
# from our analysis we saw that Store 20 is the one with the most sales.
# We will analyze this stores' results
store_graph <- train %>% filter(Store == 20)
ggplot(store_graph,aes(x=CPI,y=Weekly_Sales)) +
geom_point(aes(color=store_graph$IsHoliday)) + geom_smooth()

# Finally, look for a correlation matrix between all of our numerical features.
corrplot.mixed(cor(train[,c(-1,-2,-4,-5,-17)]), lower = "ellipse",upper =
"number",use="pairwise.complete.obs")

# Modeling
# Since we saw that the Store type is very important to predict the Weekly Sales
```

```
# of a given store, we will run a Decision Tree model to predict what Type a Store should be based on the
# different features that we have on our model.

# Decision Tree
#Using a decision tree we will like to predict the Type of a store based on all the other parameters
train.dt <-rpart(Type ~ Weekly_Sales+Size, data=train,
control=rpart.control(minsplit=1,cp=0.05))
summary(train.dt)
fancyRpartPlot(train.dt)
dim(train)

# Now we will also like to form a Decision Tree for predicting the Rank that each store lies
# with respect to their sales. We want to look at the other features in order to predict what
# range of sales a store will have in the future taking into account anything but previous sales.
rank.dt <- rpart(Rank~ Weekly_Sales+Type,data=train)
fancyRpartPlot(rank.dt)
summary(rank.dt)

# Linear Regression
# We would also like to create a linear model to find a specific value for Weekly Sales
# that we want to predict. This line of best fit is intended to approximate further data points
# based on the line that we find in our training data.

# Fitting the model with confidence interval of 95%
fit <- lm(Weekly_Sales ~.-Type-CPI, data=train)
predict_fit_confidence <- predict(fit, newdata=test, interval="confidence", level=0.95)
summary(fit)


# Predicting the linear regression model for Weekly_Sales
Model <- lm(Weekly_Sales ~.-Type-CPI, data=train)
predict_weeklysales <- predict(Model, newdata=test)
summary(Model)

# From the above linear model, we can see that the linear regression model for weekly sales is
# Weekly_sales= -1.322e+06 + (Store2 * -8.116e+03) + (Store3 * -8.934e+05) + (Store4 * 1.190e+05) + (Store5 * -9.892e+05) + (Store6 * -2.299e+04) + (Store7 * -7.390e+05) + (Store8 * -4.537e+05) + (Store9 * -7.660e+05) + (Store10 * -1.407e+04) +  (Store11 * -1.536e+05) + (Store12 * -3.912e+05) + (Store13 * 5.048e+04) + (Store14 * 1.075+05) + (Store15 * -6.784e+05) +  (Store16 * -8.039e+05) +  (Store17 * -4.757e+05) +  (Store18 * -3.903e+05) + (Store19 * -8.081e+04) +  (Store20 * 1.273e+05) + (Store21 * -5.520e+05) +(Store22 * -4.341e+05) +  (Store23 * -1.666e+05) +  (Store24 * -1.547e+05) +  (Store25 * -6.037e+05) + (Store26 * -4.491e+05) +  (Store27 * 2.724e+03) +  (Store28 * -1.408e+05) +(Store29 * -7.446e+05) +  (Store30 * -8.547e+05) + (Store31 * -1.145e+05) + (Store32 * -3.301e+05) +
```

(Store33 * -1.015e+06) + (Store34 * -4.482e+05) + (Store35 * -4.313e+05) + (Store36 * -9.190e+05) + (Store37 * -7.735e+05) + (Store38 * -8.556e+05) + (Store39 * -8.724e+04) + (Store40 * -4.789e+05) + (Store41 * -2.490e+05) + (Store42 * -7.261e+05) + (Store43 * -6.385e+05) + (Store44 * -1.002e+06) + (Store45 * -5.154e+05) + (Date * -3.784e+00) + (IsHolidayTRUE * 1.838e+04) + (WeekNum * 1.294e+03) + (Temperature * -3.888e+02) + (Fuel_Price * -7.942e+03) + (MarkDown1 * 8.325e-01) + (MarkDown2 * -8.128e-01) + (MarkDown3 * 2.903e+00) + (MarkDown4 * -4.926e-01) + (MarkDown5 * 1.776e+00) + (Unemployment * -9.079e+03) + (RankB * 1.973e+05) + (RankC * 5.972e+05) + (RankD * 1.231e+06) + (RankE * 2.177e+06)

```
# Linear Regression Model Evaluation and Results
# From the model summary, the model p value and predictor's p value are less than the significance level.
# So we have a statistically significant model.
# Also, the R-Squared and Adj R-Squared are comparative to the original model
# built on full data. R-Squared and Adj R-Squared values are approximately 97% indicating a good prediction model.

# Prediction Accuracy Measures for Linear Model
# 1. Correlation accuracy
actuals_preds <- data.frame(cbind(actuals=test$Weekly_Sales, predicteds=predict_weeklysales))
# make actuals_predicteds dataframe.
correlation_accuracy <- cor(actuals_preds)
correlation_accuracy
head(actuals_preds)
# Here we can see that correlation between the actuals and predicteds values is high,
# indicating a good prediction model.

# 2. Min-Max Accuracy Calculation
min_max_accuracy <- mean(apply(actuals_preds, 1, min) / apply(actuals_preds, 1, max))
min_max_accuracy
# The Min-Max Accuracy is high (93.23%), indicating a good linear prediction model.

# 3. MAE (Mean Absolute Error) Calculation
mae <- mean(abs(actuals_preds$predicteds - actuals_preds$actuals))
mae

# The value of MAE is 69846.7.

# sqrt(mean(error^2))
# 4. RMSE (Root Mean Square Error) Calculation
rmse <- sqrt(mean((actuals_preds$predicteds - actuals_preds$actuals)^2))
rmse
# The value of RMSE is 97003.42.

# 5. MAPE (Mean Absolute Percentage Error) Calculation
mape <- mean(abs((actuals_preds$predicteds - actuals_preds$actuals))/actuals_preds$actuals)
```

```
mape
# Here we can see that MAPE is low(approximately 7%), indicating a good prediction model.

# 6. R-squared Calculation
library(caret)
predictions <- predict( Model, test)
R2(predictions, test$Weekly_Sales)
# R-Squared value is high(approximately 97%), indicating a good prediction model.

# Model Evaluation and Results
# First we will evaluate the result of our Type prediction Model in order to have a clearer
# picture of its accuracy with unseen data.

prediction <- predict(train.rpart,test, type="class")
test$Prediction <- prediction

#Find the percentage accuracy of our model
accur_table <- test %>% select(Type,Prediction)
bool_vector <- accur_table$Type == accur_table$Prediction
length(which(bool_vector)) / length(bool_vector)

# We can see that the accuracy is a high 95%, so we can conclude that a Decision Tree is
# a very powerful technique for this data set. Since the Type of the store is really significant
# as we saw in our exploration. This result can help the company categorize new stores and
therefore predict how much they should sell based on the Type of store they are grouped into.

# Second, we evaluate the accuracy of our Rank prediction.
# This will help us to know within which range a certain store should sell in a given week of the
year.

# Evaluate Results of the Model , type="class"
prediction_rank <- predict(rank.dt,test, type="class")
test$RankPred <- prediction_rank
accuracy_test <- test %>% select(Rank,RankPred)
values <- accuracy_test$Rank == accuracy_test$RankPred
length(which(values)) / length(values)

# t<- table(predictions=prediction_rank,actual=test$Rank)
# t
# sum(diag(t))/sum(t)
# We can see that the accuracy is a high 99%.

# rank.rpart$cptable
# ROC curve for Rank
library(pROC)
predprobs <- predict(rank.rpart,train.test, type="prob")
```

```
par(mfrow=c(2,3))
# ROC for Rank A
plot(roc(train.test$RankPred,predprobs[,1]),main="ROC for Rank A")
# ROC for Rank B
plot(roc(train.test$RankPred,predprobs[,2]),main="ROC for Rank B")
# ROC for Rank C
plot(roc(train.test$RankPred,predprobs[,3]),main="ROC for Rank C")
# ROC for Rank D
plot(roc(train.test$RankPred,predprobs[,4]),main="ROC for Rank D")
# ROC for Rank E
plot(roc(train.test$RankPred,predprobs[,5]),main="ROC for Rank E")

# The plots above represent the ROC curve for the 5 Ranks.

# Decision Tree Confusion Matrix for Rank
conf <- table(actualclass=test$RankPred,predictedclass=test$Rank)
confusionMatrix(conf)

# The Confusion matrix for Rank shows a high accuracy of 99.69%, indicating that decision tree
is a good model.
# P value(2.2e-16) is less than the significance level. So we have a statistically significant model.

# train.rpart$cptable
# ROC curve for Rank Type
library(pROC)
predprobs2 <- predict(train.rpart,train.train, type="prob")
par(mfrow=c(1,3))
# ROC for Type A
plot(roc(train.train$Type,predprobs2[,1]),main="ROC for Type A")
# ROC for Type B
plot(roc(train.train$Type,predprobs2[,2]),main="ROC for Type B")
# ROC for Type C
plot(roc(train.train$Type,predprobs2[,3]),main="ROC for Type C")

# The plots above represent the ROC curve for the 3 Types.

# Decision Tree Confusion Matrix for Rank Type
conf2 <- table(actualclass=test$Prediction,predictedclass=test$Type)
confusionMatrix(conf2)

# The Confusion matrix for Type shows a high accuracy of 95.17%, indicating that decision tree
is a good model.
# P value(2.2e-16) is less than the significance level. So we have a statistically significant model.
```