BIG MARKET SALES PREDICTION

REVIEW REPORT

Submitted by

Naman Kapoor (18BCE0370)

Adithya Yalagoud (18BCE0377)

Prithiraj Bhuyan (18BCE2512)

Prepared For

Data Visualization (CSE3020) – PROJECT COMPONENT

Submitted To

Prof. Nalini N

School of Computer Science and Engineering

VIT®

Vellore Institute of Technology

(Deemed to be University under section 3 of UGC Act, 1956)

INDEX

1. ABSTRACT

Nowadays shopping malls and Big Marts keep track of their sales data of each and every individual item for predicting future demand of the customer and update the inventory management as well.

These data stores basically contain a large number of customer data and individual item attributes in a data warehouse. Further, anomalies and frequent patterns are detected by mining the data store from the data warehouse. The resultant data can be used for predicting future sales volume with the help of different machine learning techniques for the retailers like Big Mart. In this paper, we propose a predictive model using Xgboost technique for predicting the sales of a company like Big Mart and found that the model produces better performance as compared to existing models. A comparative analysis of the model with others in terms performance metrics is also explained in detail.

KEYWORDS: Predictive Model, Visualization, XgBoost, Machine Learning

2. INTRODUCTION

2.1 Background

- Day by day competition among different shopping malls as well as big marts is getting more serious and aggressive only due to the rapid growth of the global malls and on-line shopping. Every mall or mart is trying to provide personalized and short-time offers for attracting more customers depending upon the day, such that the volume of sales for each item can be predicted for inventory management of the organization, logistics and transport service, etc.
- Present machine learning algorithms are very sophisticated and provide techniques to predict or forecast the future demand of sales for an organization, which also helps in overcoming the cheap availability of computing and storage systems.
- In this project, we are addressing the problem of big mart sales prediction or forecasting of an item on customer's future demand in different big mart stores

across various locations and products based on the previous record.

- Different machine learning algorithms like linear regression analysis, random forest, etc are used for prediction or forecasting of sales volume.
- As good sales are the life of every organization so the forecasting of sales plays an important role in any shopping complex. Always a better prediction is helpful, to develop as well as to enhance the strategies of business about the marketplace which is also helpful to improve the knowledge of the marketplace.
- A standard sales prediction study can help in deeply analyzing the situations or the conditions previously occurred and then, the inference can be applied about customer acquisition, funds inadequacy and strengths before setting a budget and marketing plans for the upcoming year.

2.2 Objective

- The aim is to build a predictive model and find out the sales of each product at a particular store and hence predict the sales of supermarkets according to the sample supermarket dataset. The idea is to find out the properties of a product, and store which impacts the sales of a product. Using this model, we will try to understand the properties of products and stores which play a key role in increasing sales.
- We will also use various visual plots between the data to better understand it.
- Algorithm we will use to build our models are:
    - Linear Regression
    - Ridge Regression
    - Decision Tree Regression

2.3 Motivation

Seeing the tremendous growth of sales in big markets, it has become very important for the owner of the big markets to visualize the data in order to gain various insights of the data. Understanding the data will help the owner determine the best product in his/her market and discard those products that are less preferred to be sold. As the products are ordered in bulk in big markets, visualization of data will allow the owner to order those products that are sold the most so that all the products ordered by the owner are sold to the customers. Analysing the data is also important as it will help the owner to predict the sales of their products. This has motivated us to take up the project so that we can help the big market owners to analyse their data and grow their business.

2.4 Contributions of the Project

Each of us contributed equally towards analysing the visualizations of different attributes using scatter plots, bar plots, box plots, cow plot, correlation plots, etc.
Then we studied about the three regression models - linear, ridge and decision tree and using these models we worked on predicting the overall sales for the supermarket.

2.5 Organization of the Project

- Business User: Super markets like Big Mart etc.
- Project Manager: Ensures that key milestones and objectives are met on time and at the expected quality.
- Business Intelligence Analyst: Provides business domain expertise based on a deep understanding of the data, key performance indicators (KPIs), key metrics, and business intelligence from a reporting perspective. Business Intelligence Analysts generally create dashboards and reports and know the data feeds and sources.
- Database Administrator (DBA): Provisions and configures the database environment to support the analytics needs of the working team. These responsibilities may include providing access to key databases or tables and ensuring the appropriate security levels are in place related to the data repositories.
- Data Scientist: Provides subject matter expertise for analytical techniques, data modeling, and applying valid analytical techniques to given business problems.

Ensures overall analytics objectives are met. Designs and executes analytical methods and approaches with the data available to the project.

2.6 Software Requirements

- R x64 3.6.1 - It provides the R library functions which can be used to work, analyse and visualize various datasets.
- RStudio - RStudio is an integrated development environment for R, a programming language for statistical computing and graphics.
- Google Collab - Collab is a free Jupyter notebook environment that runs entirely in the cloud. Most importantly, it does not require a setup and the notebooks that you create can be simultaneously edited by your team members.

2.7 Dataset Used

- The dataset used in our project is - BIG_MART_SALES, from Kaggle.com
- LINK TO THE DATASET: https://www.kaggle.com/aakash2016/big-mart-sales-dataset
- It contains two files- Test dataset and train dataset
- The training set contains 8253 rows (instances) of 12 variables (attributes).
- The testing set contains 5681 rows (instances) of 11 variables (attributes).
- It contains different data attributes such as: Item_Identifier, Item_Weight, Item_Type, Item_MRP, etc.

2.8 Packages/Libraries Used

Cowplot
Corrplot
Ggplot2
Magrittr
Xgboost
Caret

3. LITERATURE SURVEY

3.1 Background

We started with making some hypotheses about the data without looking at it. Then we moved on to data exploration where we found out some nuances in the data which required remediation. Next, we performed data cleaning and feature engineering, where we imputed missing values and solved other irregularities, made new features and also made the data model-friendly by one-hot-coding. Finally we made a linear regression, decision tree and ridge regression model and got a glimpse of how to tune them for better results.

3.2 Literature Review

Research Paper	Authors	Published Year	Description
Business data mining machine learning perspective. Information & management 39(3), 211–225	Bose, I., Mahapatra, R.K.	2001	This paper mainly talks about the various statistical and computational methods using machine learning techniques. It also elaborates upon the automated process of knowledge acquisition in the field of Machine Learning. Various machine learning (ML) techniques with

			their applications in different sectors are also presented in the paper
A few useful things to know about machine learning. Commun. acm 55(10), 78–87	Domingos, P.M.	2012	Machine learning is the process where a machine will learn from data in the form of statistically or computationally method and process knowledge acquisition from experiences. This paper focuses on the above fact and beautifully explains the concept of machine learning.
Applications of machine learning and rule induction. Communications of the ACM 38(11), 54–64	Langley, P., Simon, H.A.	1995	This paper mainly points out that the most widely used data mining technique in the field of business is the Rule Induction (RI) technique as compared to other data mining

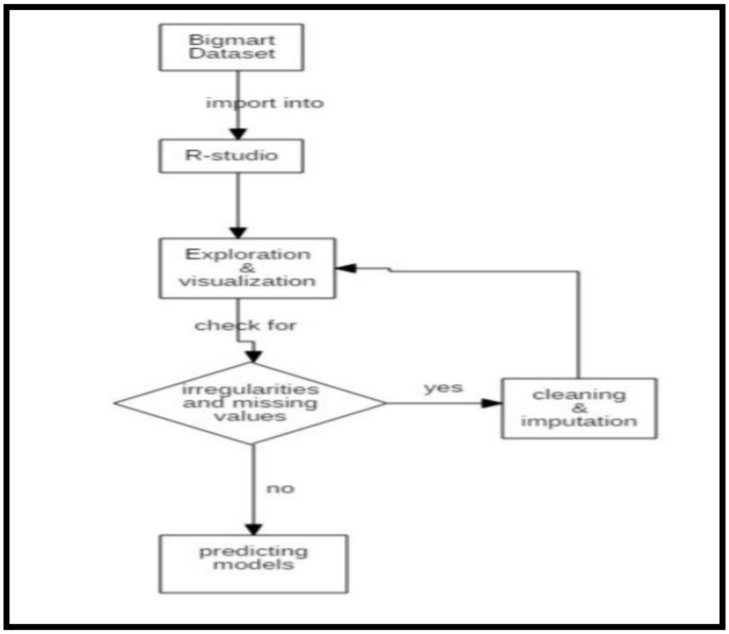
			techniques.
A two-level statistical model for big mart sales prediction. IEEE	Punam, K., Pamula, R., Jain, P.K	2018	This paper presents forth a two-level statistical model for analysing the big market sales which provides us a guide to start off with our project.
A comparative study of linear and nonlinear models for aggregate retail sales forecasting. International Journal of production economics 86(3), 217–231	Chu, C.W., Zhang, G.P.	2003	Linear and non-linear comparative analysis models for sales forecasting is proposed for the retailing sector. This paper provides forth a comparison between the linear and non-linear models helping us to determine when to use each of them.
A seasonal discrete grey forecasting model for fashion retailing. Knowledge-Based Systems 57, 119–126	Xia, M., Wong, W.K	2014	This paper proposes the differences between classical methods (based on mathematical and statistical models) and modern heuristic

			methods and also named exponential smoothing, regression, auto regressive integrated moving average (ARIMA), generalized auto regressive conditionally heteroscedastic (GARCH) methods.
Forecasting methods and applications.	Makridakis, S., Wheelwright, S.C., Hyndman, R.J.	2008	This paper mainly focuses on the challenges that are faced by linear models to deal with the asymmetric behavior in most real-world sales data. Some of the challenging factors include lack of historical data, consumer-oriented markets face uncertain demands, and short life cycles of prediction methods.

Ridge Regression in Practice	Donald W. Marquardt, Ronald D. Snee	2012	A review of the theory of ridge regression and its relation to generalized inverse regression is presented along with the results of a simulation experiment and three examples of the use of ridge regression in practice.
Prediction of retail sales of footwear using feedforward and recurrent neural networks. Neural Computing and Applications 16(4-5), 491–502	Das, P., Chaudhury, S.	2007	This paper focuses on using neural networks for predicting weekly retail sales, which decrease the uncertainty present in the short term planning of sales. This helps in understanding how we can also use neural networks in the field of prediction and even in big mart sales analysis.

4. PROPOSED WORK

4.1 Proposed Architecture



4.2 Tables and Constraints

Attribute	Datatype	Constraint
Item_Identifier	chr	pk
Item_Weight	num	
Item_Fat_Content	chr	Item_Fat_Content in (Low Fat, Regular, LF)
Item_Visibility	num	Not null
Item_Type	chr	Not null
Item_MRP	num	Not null
Outlet_Identifier	chr	Foreign Key (Outlet)

Outlet_Establishment_Year	int	Not null
Outlet_Size	chr	
Outlet_Location_Type	chr	Outlet_Location_Type in (Tier 1, Tier 2, Tier 3)
Outlet_Type	chr	Outlet_Type in (Supercart, Grocery Store)
Item_Outlet_Sales	num	Not null

4.3 Implementation Details

In order to carry out the big markets sales prediction, which helped us to understand our dataset in various ways and use appropriate machine learning models to predict the data. This has also helped us in carrying out proper analysis of the data and exploring various techniques to predict the sales in big markets. The stages followed are:

Hypothesis Generation: This is a very pivotal step in the process of analyzing data. This involves understanding the problem and making some hypothesis about what could potentially have a good impact on the outcome. This is done BEFORE looking at the data, and we end up creating a laundry list of the different analysis which we can potentially perform if data is available. Making a proper hypothesis from the problem allows us to perform our analysis in a much efficient way as we now know what we should be looking for in the data provided.

Data Exploration: In the step of Data Exploration, we have explored the dataset available with us. We have performed some basic data exploration steps including finding the number of columns, dimensions of each column, and figuring out some irregularities in the dataset. In this particular step, we have also visualized the data using various data visualization techniques which has helped us to infer a lot of information from the dataset including the growth of sales, regularity of the dataset and so on.

Data Cleaning: This step typically involves imputing missing values and treating outliers. Though outlier removal is very important in regression techniques, advanced tree based algorithms are impervious to outliers. Here we modify some of the data values so that our predictive model would be able to learn in a better way from the training dataset provided to it.

Feature Engineering: This step is mainly to overcome the nuances found in the data exploration phase. This is the final step of making our data ready for analysis. Here some new variables are also created based on the existing ones in order to have a better analysis of the dataset. This phase includes combining the outlet_type, modifying the item_visibility, creating a broad category of type of item, determining the years of operation of the store, modifying categories of item_fat_content, numerical and one hot encoding of categorical variables, and exporting the final dataset to be analysed.

Model Building: Now the dataset is ready for it to be analysed by the predictive models. In this particular step, we have implemented three predictive models in order to determine the Item_Outlet_Sales. The three models are linear regression model, ridge regression model and decision tree regression model.

The first step will be declaring variables that will do the calculations of data. The variables should be declared for Item visibility, Item type, Outlet size, Outlet location type, Outlet type, and Item outlet sales. The data is categorized, and the first step will be to the correction of irregularities through data pre-processing. The variation of data is a real tough task as there are around 1562 unique items in a single store.

The second step is to combine the outlet type through various parameters such as item visibility, years of operation, etc. Then create a broad category for item type using many item identifiers. Then the algorithm of ML will study the variations. A generic function that makes the model and performs cross-validation should be made.

The next step will be the model making of the application, which will comprise the linear regression model, ridge regression model, decision tree model to decide the results, etc. The data fed to the application will go through sorting and arrangements which will be efficiently performed by Machine Learning.

Missing Value Treatment:

There are different methods to treat missing values based on the problem and the data. Some of the common techniques are as follows:

1. Deletion of rows: In the train dataset, observations having missing values in any variable are deleted.
2. The downside of this method is the loss of information and drop in prediction power of the model.
3. Mean/Median/Mode Imputation: In case of continuous variable, missing values can be replaced with
4. Mean or median of all known values of that variable. For categorical variables, we can use mode of the given values to replace the missing values.

Treatment 1: We have missing values in Item_Weight and Item_Outlet_Sales. Missing data in Item_Outlet_Sales can be ignored since they belong to the test dataset. We'll now impute Item_Weight with mean weight based on the Item_Identifier variable.

```
sm = sum(is.na(combi$Item_Weight))
nonNullVal = nrow(combi) - sm
missing_index = which(is.na(combi$Item_Weight))
for(i in missing_index){
  item = combi$Item_Identifier[i]
  combi$Item_Weight[i] = sum(combi$Item_Weight, na.rm = T)/nonNullVal
}
```

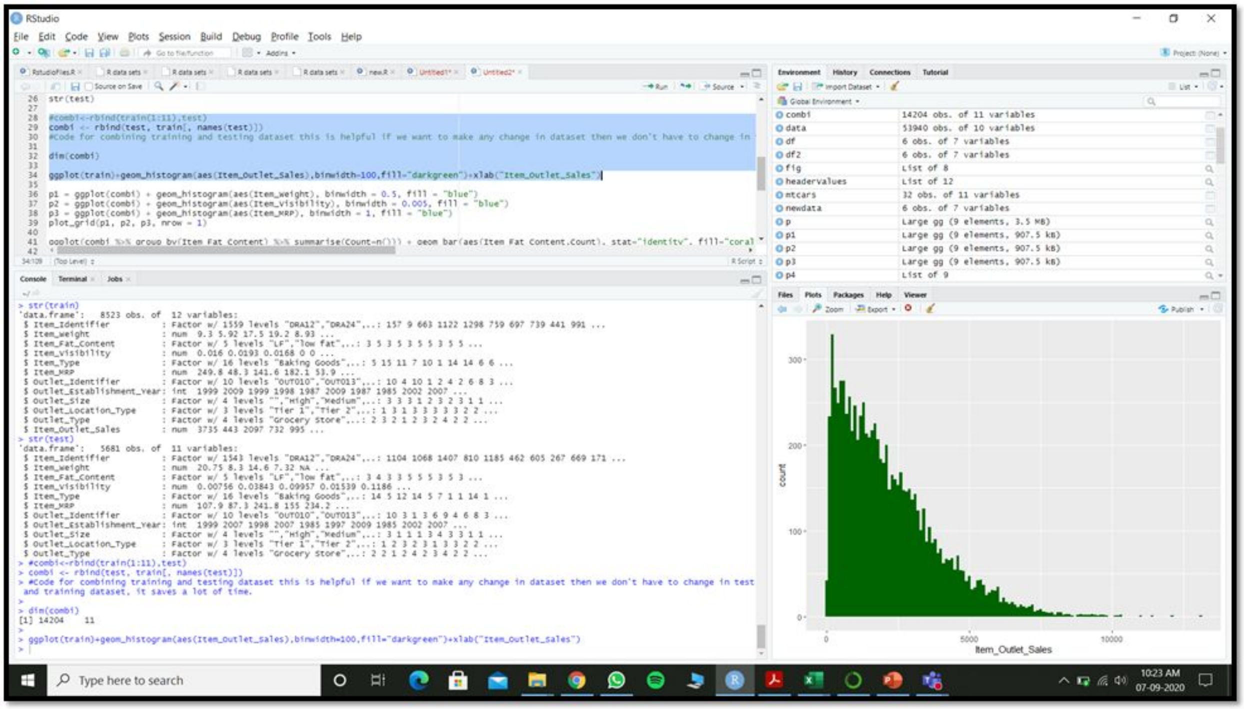
Treatment2: Replacing 0's in Item_Visibility variable Similarly, zeroes in Item_Visibility variable can be replaced with Item_Identifier wise mean values of Item_Visibility. It can be visualized in the plot below.

Before: Since item visibility can not be zero so we replace it with the mean of the Item_Identifier.

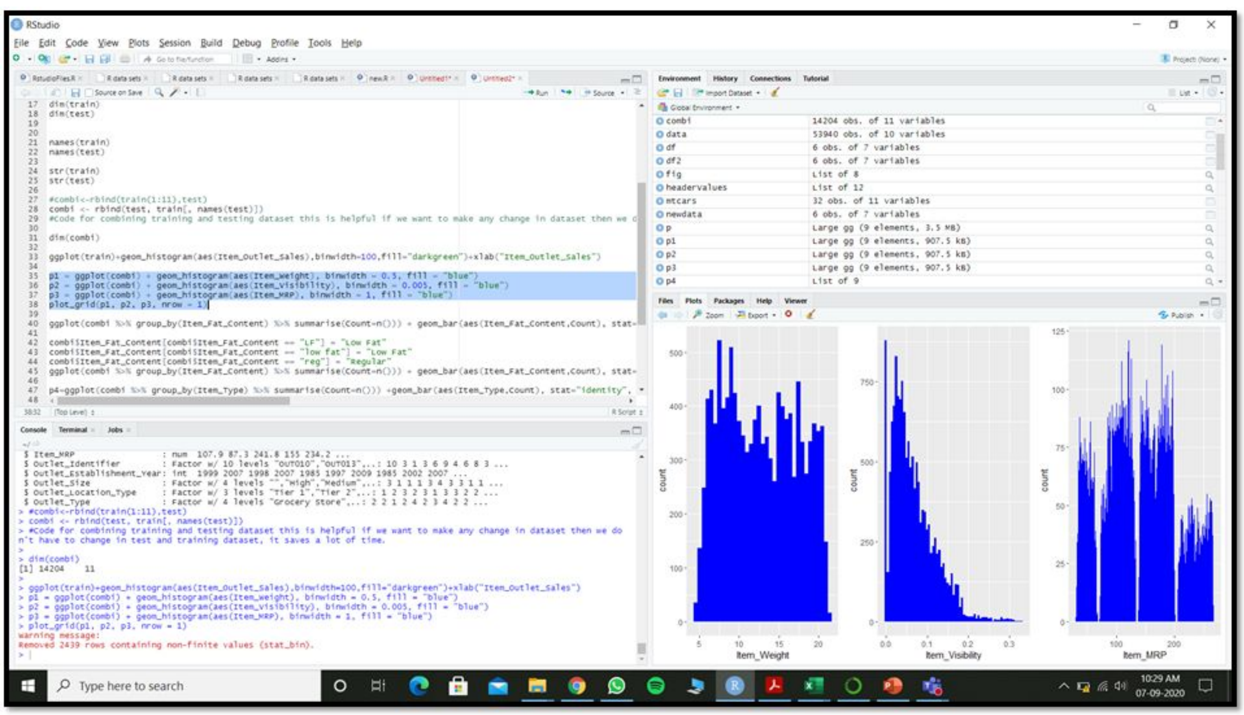
```
ggplot(combi) + geom_histogram(aes(Item_Visibility), bins=100)
```

5. RESULTS

5.1 VISUALIZATIONS

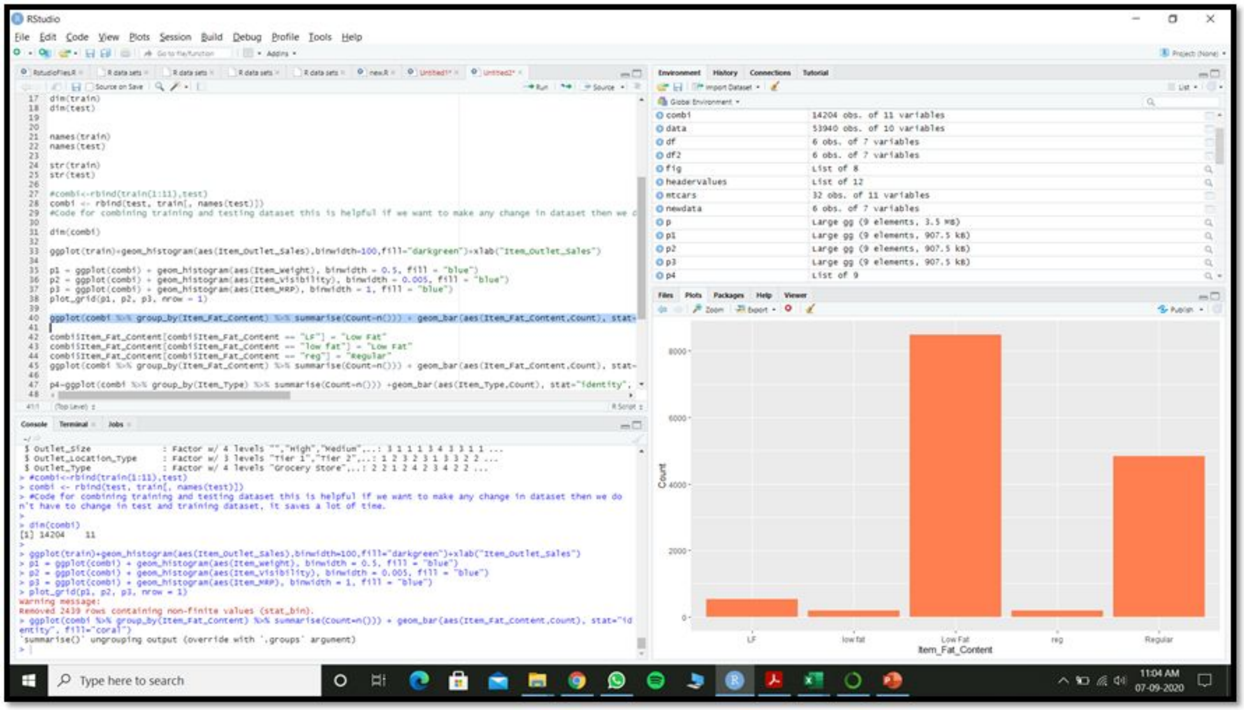


HISTOGRAM-ITEM_OUTLET_SALES

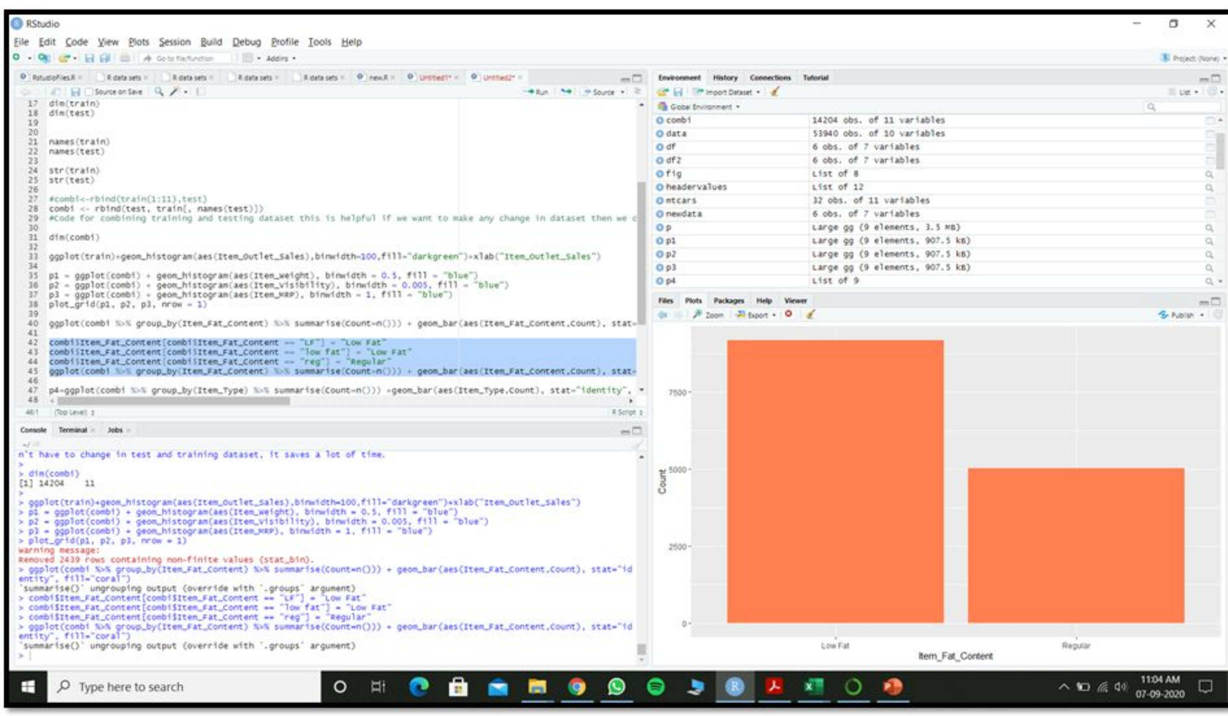


HISTOGRAM-ITEM_WEIGHT, ITEM_VISIBILITY, ITEM_MRP

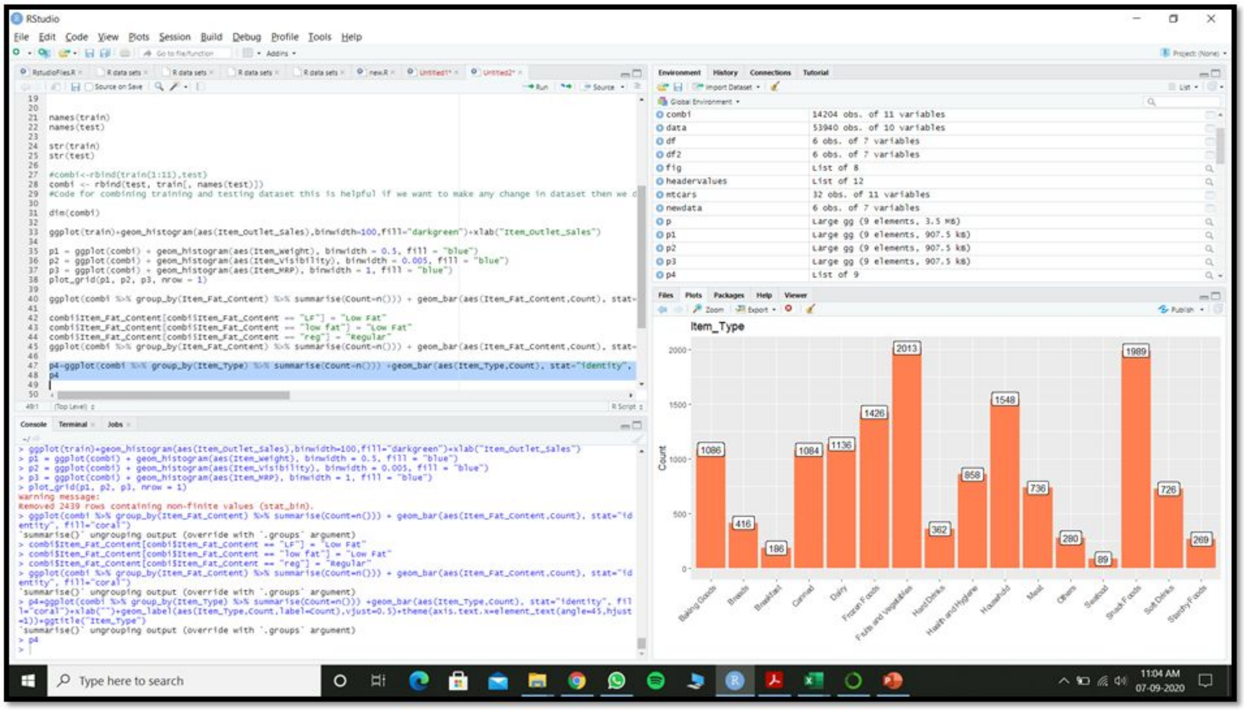
OBSERVATIONS: NO SPECIFIC PATTERN IN ITEM_WEIGHT, ITEM_VISIBILITY IS RIGHT SKEWED, ITEM_MRP IS CLASSIFIED INTO 4 CLASS RANGES



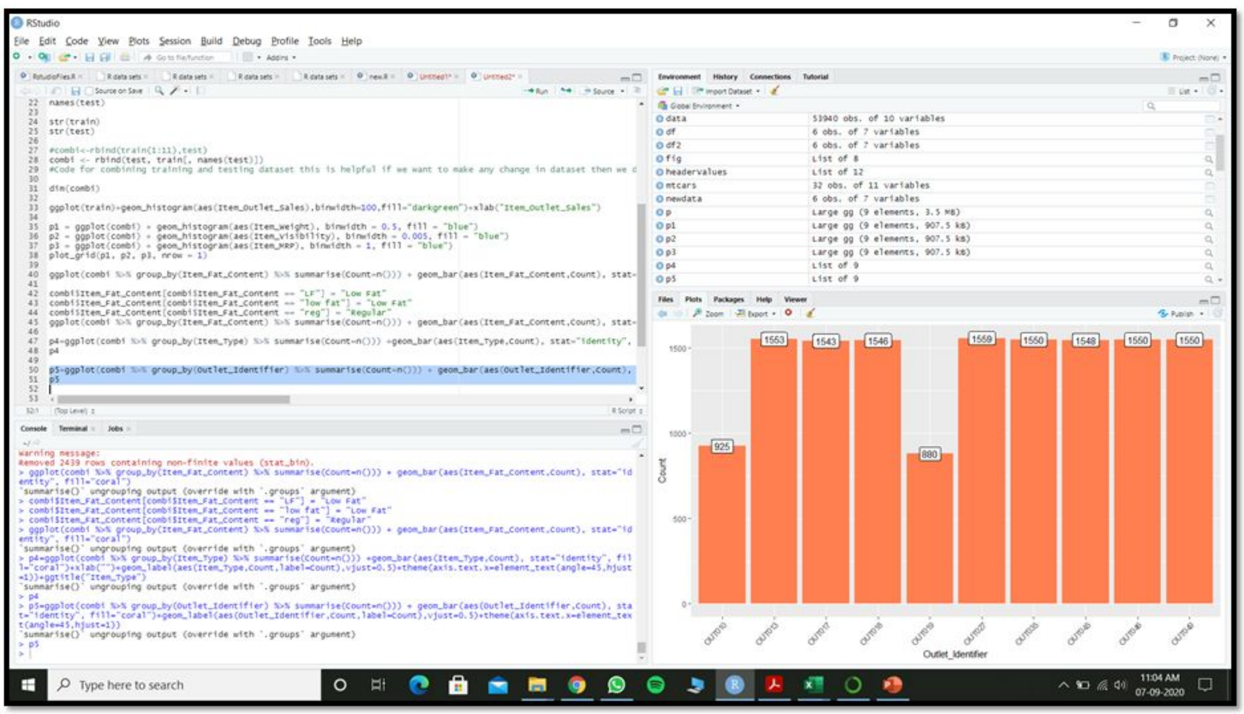
BAR GRAPH-ITEM_FAT_CONTENT



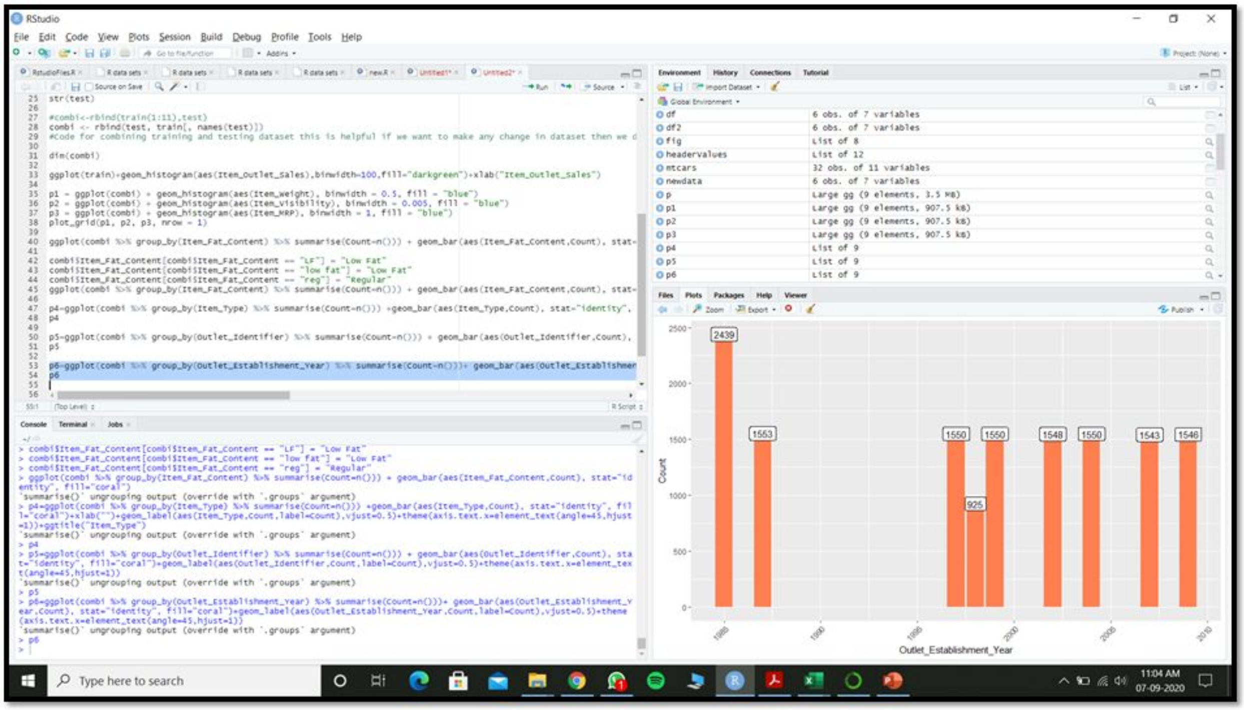
BAR GRAPH-ITEM_FAT_CONTENT



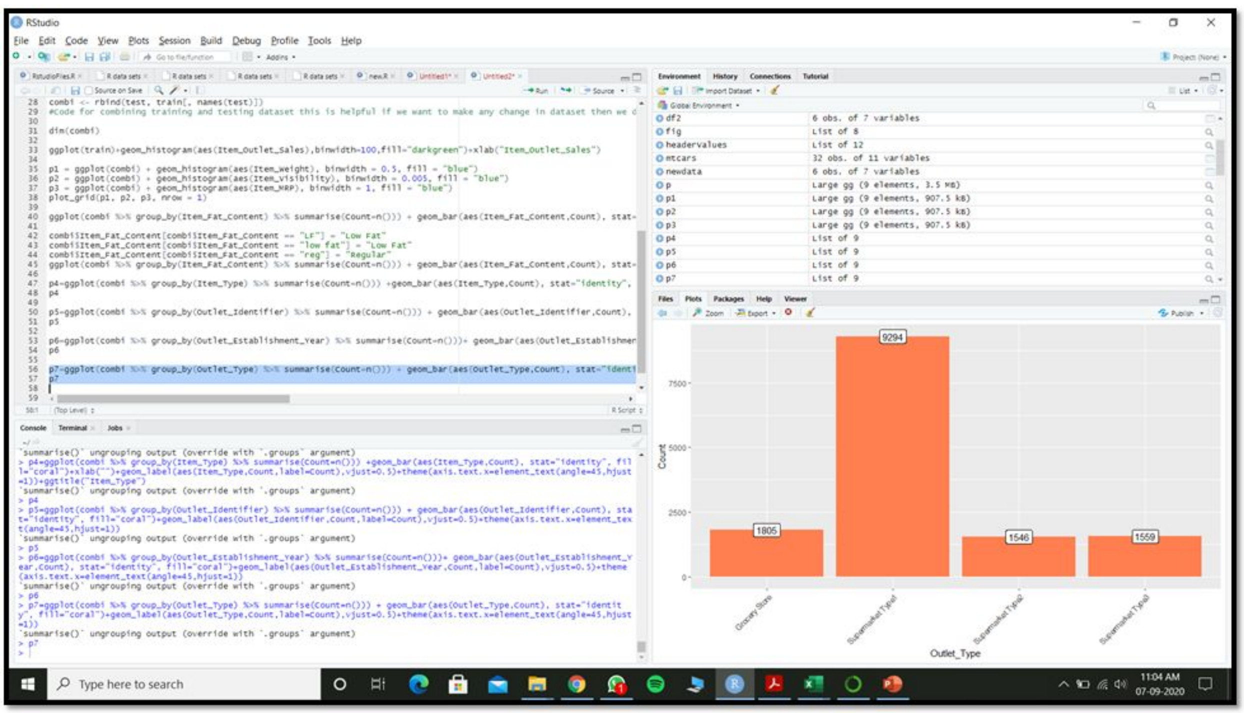
BAR GRAPH-ITEM_TYPE



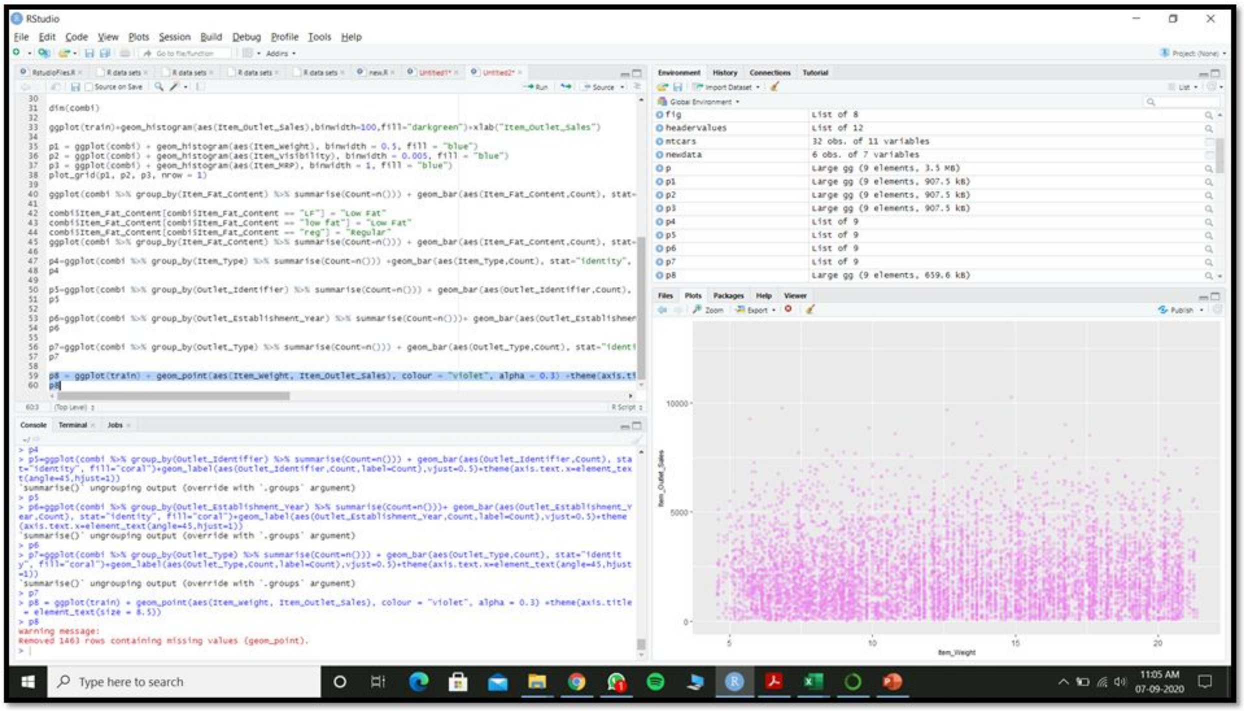
BAR GRAPH-OUTLET IDENTIFIER



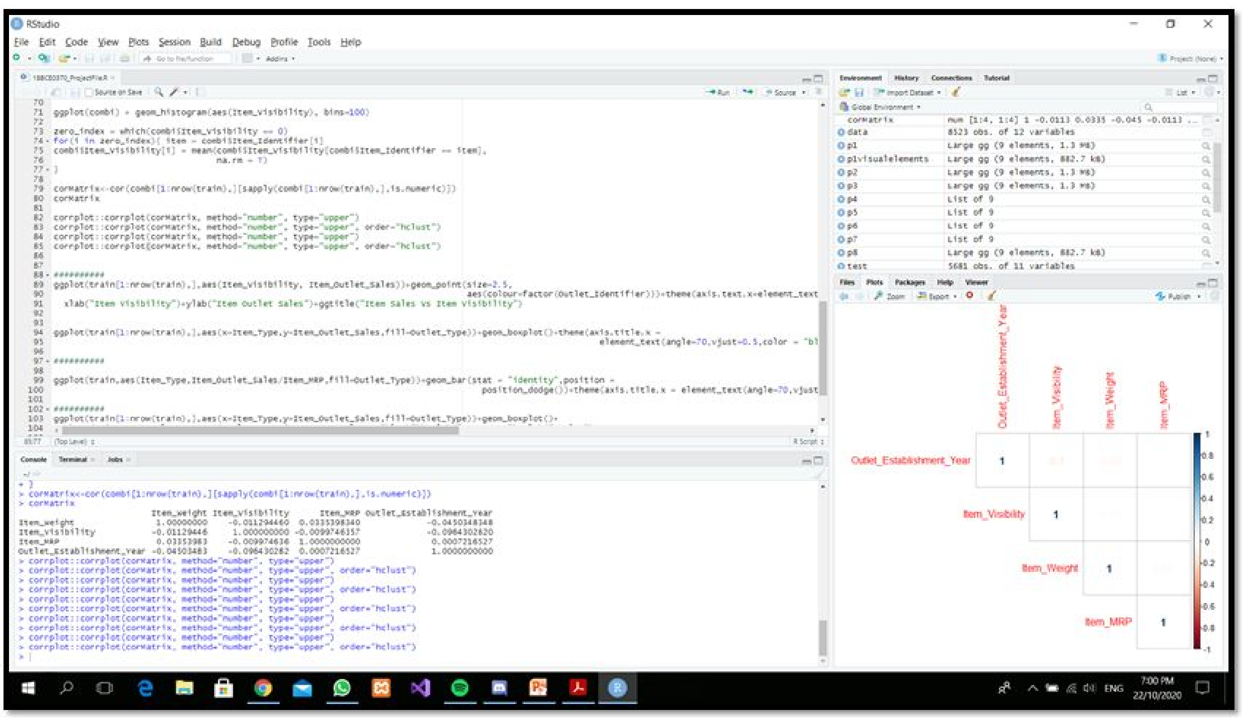
BAR GRAPH-OUTLET_ESTABLISHMENT_YEAR



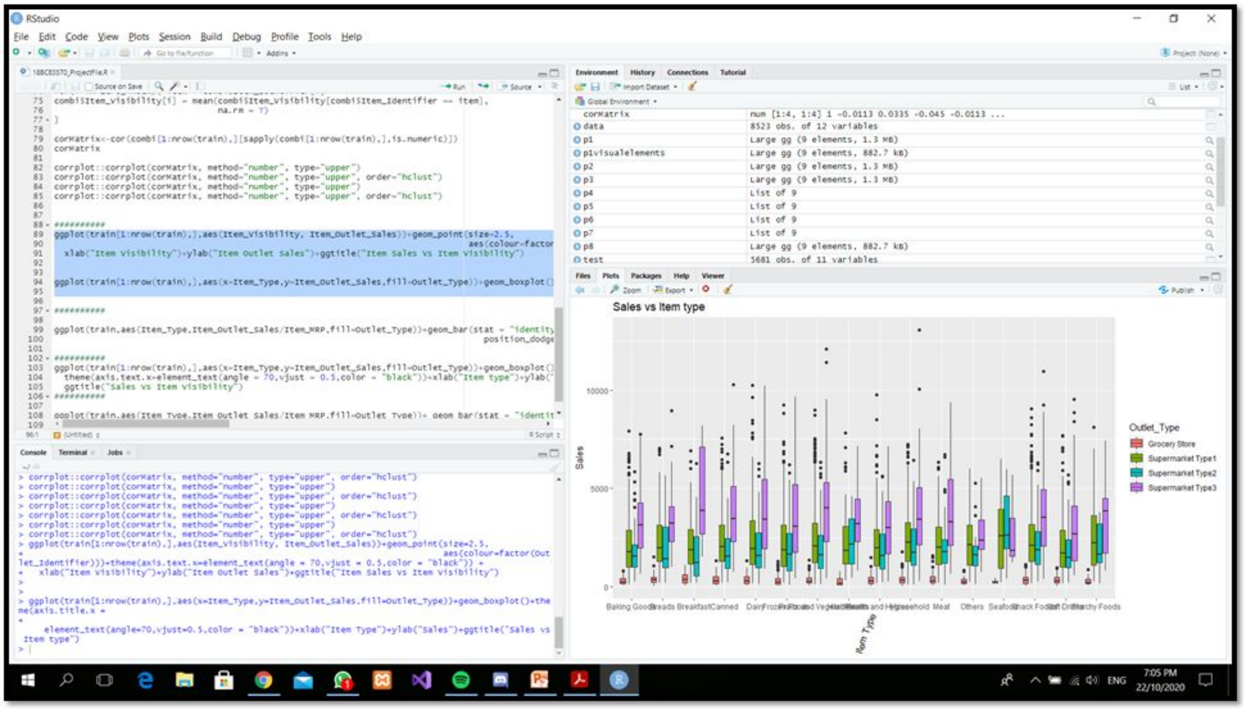
BAR GRAPH-OUTLET_TYPE



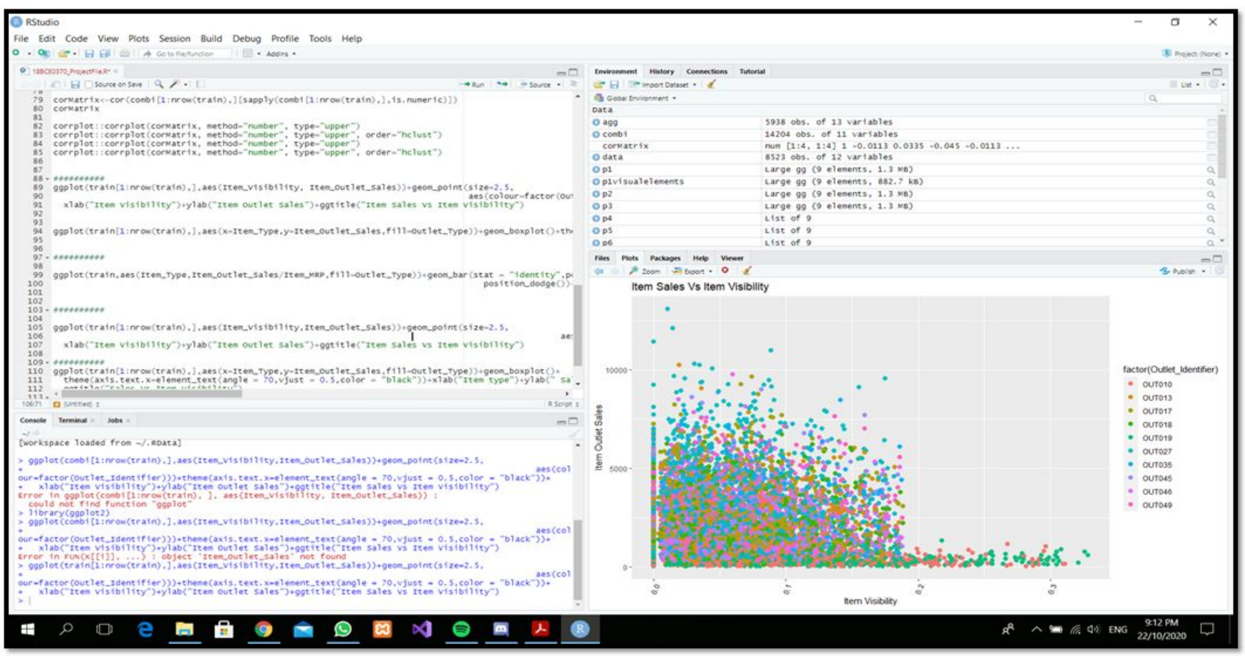
SCATTER PLOT-ITEM_WEIGHT



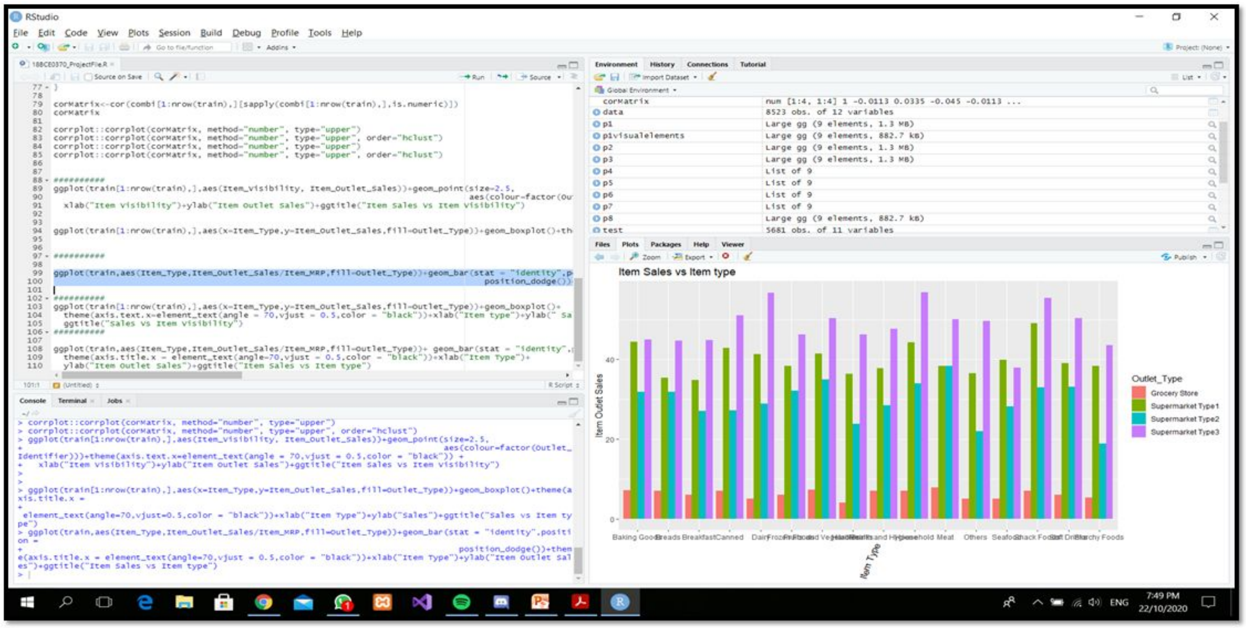
CORRELATION MATRIX



BOXPLOT-SALES vs ITEM TYPE



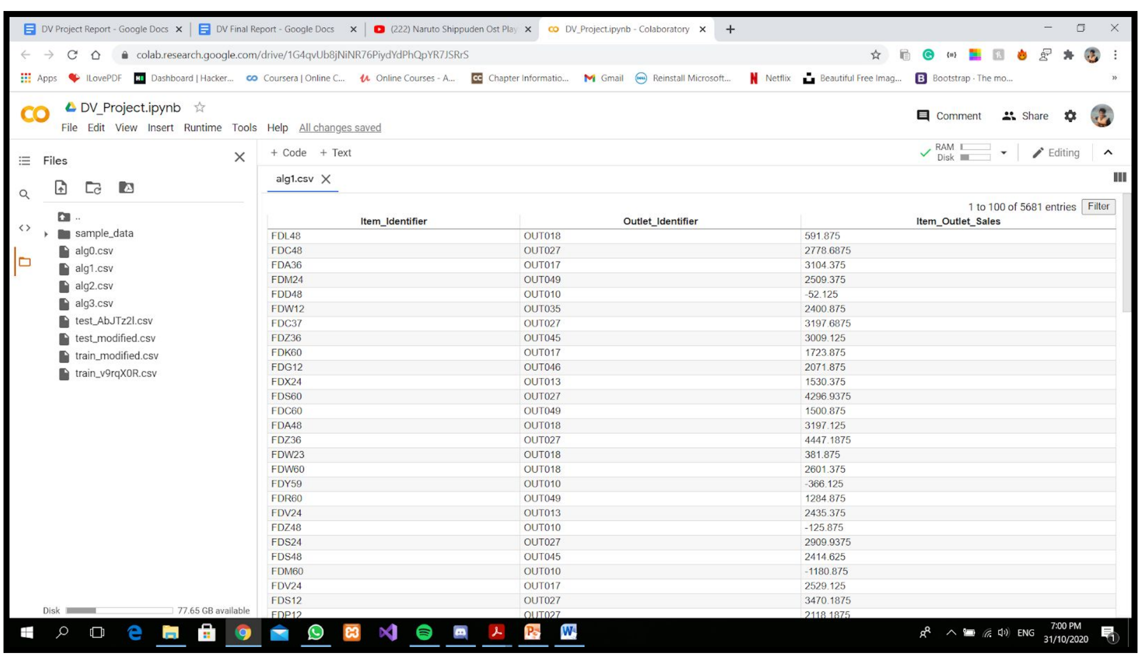
BUBBLE PLOT



BAR GRAPH-ITEM_OUTLET_SALES vs ITEM_TYPE

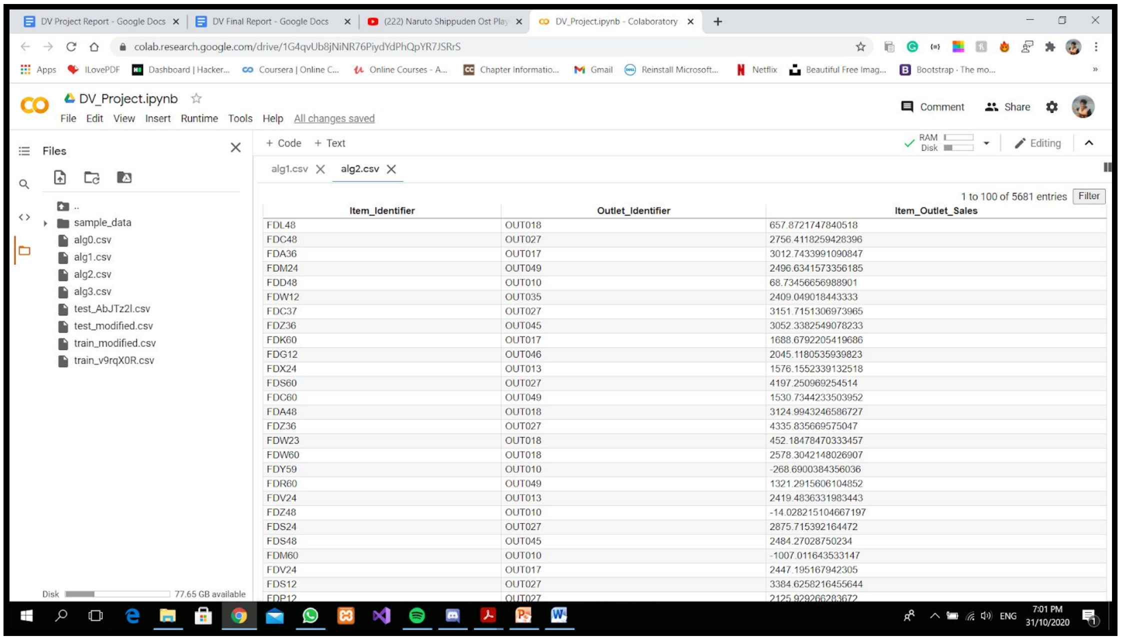
5.2 MACHINE LEARNING PREDICTION MODULE:

LINEAR REGRESSION:



OUTPUT OF LINEAR REGRESSION MODEL

RIDGE REGRESSION:

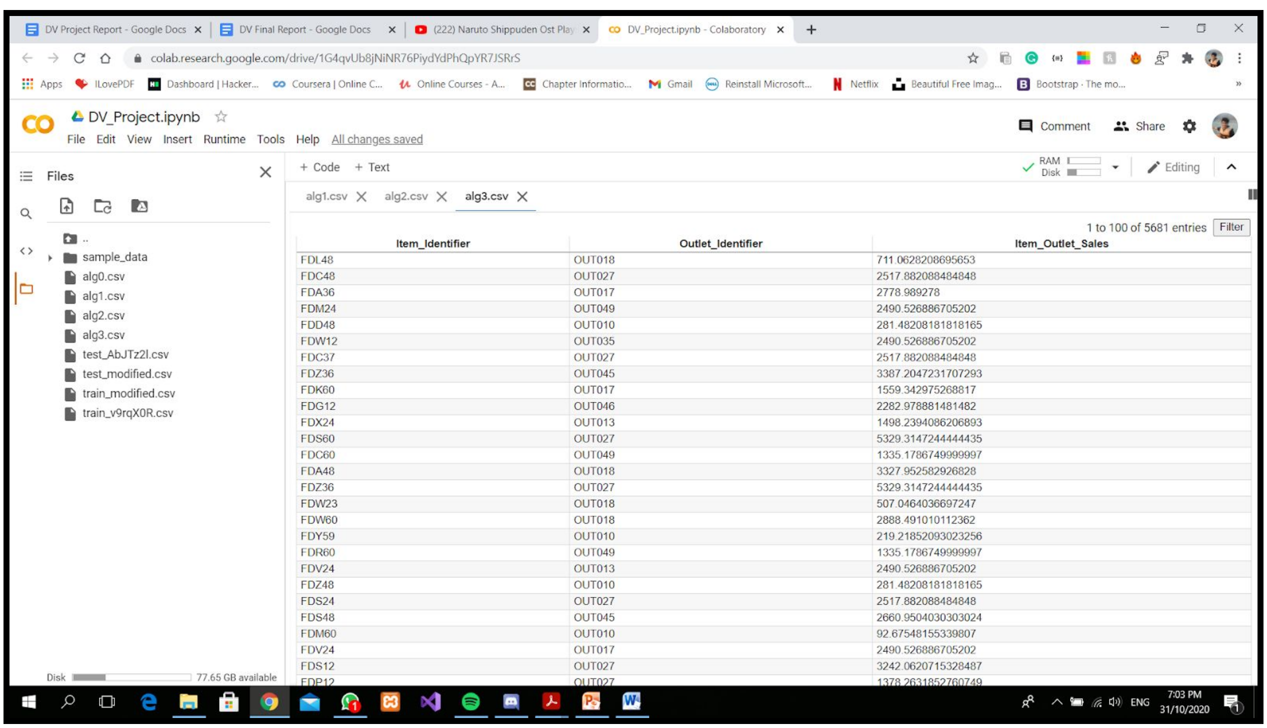


1 to 100 of 5681 entries

Item_Identifier	Outlet_Identifier	Item_Outlet_Sales
FDL48	OUT018	657.8721747840518
FDC48	OUT027	2756.4118259428396
FDA36	OUT017	3012.7433991090847
FDM24	OUT049	2496.6341573356185
FDD48	OUT010	68.73456656988901
FDW12	OUT035	2409.049018443333
FDC37	OUT027	3151.7151306973965
FDZ36	OUT045	3052.3382549078233
FDK60	OUT017	1688.6792205419686
FDG12	OUT046	2045.1180535939823
FDX24	OUT013	1576.1552339132518
FDS60	OUT027	4197.250969254514
FDC60	OUT049	1530.7344233503952
FDA48	OUT018	3124.9943246586727
FDZ36	OUT027	4335.835669575047
FDW23	OUT018	452.18478470333457
FDW60	OUT018	2578.3042148026907
FDY59	OUT010	-268.6900384356036
FDR60	OUT049	1321.2915606104852
FDV24	OUT013	2419.4836331983443
FDZ48	OUT010	-14.028215104667197
FDS24	OUT027	2875.715392164472
FDS48	OUT045	2484.27028750234
FDM60	OUT010	-1007.011643533147
FDV24	OUT017	2447.195167942305
FDS12	OUT027	3384.6258216455644
FDP12	OUT027	2125.92926628367[illegible]

<u>OUTPUT OF RIDGE REGRESSION MODEL</u>

DECISION TREE REGRESSION:



1 to 100 of 5681 entries

Item_Identifier	Outlet_Identifier	Item_Outlet_Sales
FDL48	OUT018	711.0628208695653
FDC48	OUT027	2517.882088484848
FDA36	OUT017	2778.989278
FDM24	OUT049	2490.526886705202
FDD48	OUT010	281.48208181818165
FDW12	OUT035	2490.526886705202
FDC37	OUT027	2517.882088484848
FDZ36	OUT045	3387.2047231707293
FDK60	OUT017	1559.342975268817
FDG12	OUT046	2282.978881481482
FDX24	OUT013	1498.2394086206893
FDS60	OUT027	5329.3147244444435
FDC60	OUT049	1335.1786749999997
FDA48	OUT018	3327.952582926828
FDZ36	OUT027	5329.3147244444435
FDW23	OUT018	507.0464036697247
FDW60	OUT018	2888.491010112362
FDY59	OUT010	219.21852093023256
FDR60	OUT049	1335.1786749999997
FDV24	OUT013	2490.526886705202
FDZ48	OUT010	281.48208181818165
FDS24	OUT027	2517.882088484848
FDS48	OUT045	2660.9504030303024
FDM60	OUT010	92.67548155339807
FDV24	OUT017	2490.526886705202
FDS12	OUT027	3242.0620715328487
FDP12	OUT027	1378.2631852760749

<u>OUTPUT OF DECISION TREE REGRESSION MODEL</u>

6. CONCLUSION AND FUTURE WORK

6.1 Conclusion

Nowadays shopping malls and Big Marts keep track of their sales data of each and every individual item for predicting future demand of the customer and update the inventory management as well. These data stores basically contain a large number of customer data and

individual item attributes in a data warehouse. Further, anomalies and frequent patterns are detected by mining the data store from the data warehouse. The resultant data can be used for predicting future sales volume with the help of different machine learning techniques for the

retailers like Big Mart. Some of the inferences that can be drawn from our research are -

- Item_MRP is the most important variable in predicting the target variable. New features created by us, like price_per_unit_wt, Outlet_Years, Item_MRP_Clusters, are also among the top most important variables.
- Item_outlet_Sales has a strong positive correlation with Item_MRP and a somewhat weaker negative with item_Visibility.
- In most of the cases the supermarket type3 has the most amount of sales irrespective of item_type.
- There seems to be no clear-cut pattern in Item_Weight
- We can clearly see 4 different distributions for Item_MRP. It is an interesting insight.

6.2 Future Work

In present era of digitally connected world every shopping mall desires to know the customer demands beforehand to avoid the shortfall of sale items in all sea-sons. Day to day the companies or the malls are predicting more accurately the demand of product sales or user demands. Extensive research in this area at enterprise level is happening for accurate sales prediction. As the profit made by a company is directly proportional to the accurate predictions of sales, the Big marts are desiring more accurate prediction algorithms so that the company will not suffer any losses. The attributes that we considered till now in our prediction models can be increased further to make the results more effective and legit. Also we can build a recommendation system customized to the respective supermarket which will help them to give them suggestions on how they can increase the overall sale.

7. REFERENCES

[1] Beheshti-Kashi, S., Karimi, H.R., Thoben, K.D., Lutjen, M., Teucke, M.: A survey on retail sales forecasting and prediction in fashion markets. Systems Science & Control Engineering 3(1), 154–161 (2015)

[2] Bose, I., Mahapatra, R.K.: Business data mining machine learning perspective. Information & management 39(3), 211–225 (2001)

[3] Chu, C.W., Zhang, G.P.: A comparative study of linear and nonlinear models for aggregate retail sales forecasting. International Journal of production economics 86(3), 217–231 (2003)

[4] Claypool, M., Gokhale, A., Miranda, T., Murnikov, P., Netes, D., Sartin, M.: Combing content-based and collaborative filters in an online newspaper (1999)

[5] Das, P., Chaudhury, S.: Prediction of retail sales of footwear using feedforward and recurrent neural networks. Neural Computing and Applications 16(4-5), 491–502 (2007)

[6] Domingos, P.M.: A few useful things to know about machine learning. Commun. acm 55(10), 78–87 (2012)

[7] Langley, P., Simon, H.A.: Applications of machine learning and rule induction. Communications of the ACM 38(11), 54–64 (1995)

[8] Loh, W.Y.: Classification and regression trees. Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery 1(1), 14–23 (2011)

[9] Makridakis, S., Wheelwright, S.C., Hyndman, R.J.: Forecasting methods and applications. John wiley & sons (2008)

[10] Ni, Y., Fan, F.: A two-stage dynamic sales forecasting model for fashion retail. Expert Systems with Applications 38(3), 1529–1536 (2011)

[11] Punam, K., Pamula, R., Jain, P.K.: A two-level statistical model for big mart sales prediction. In: 2018 International Conference on Computing, Power and Communication Technologies (GUCON). pp. 617–620. IEEE (2018)

APPENDIX A - CODING

RStudio Code:

```
install.packages("caret")
install.packages("corrplot")
#install.packages("xgboost")
#install.packages("cowplot")
install.packages("magritrr")
install.packages("ggplot2")
library(data.table) # used for reading and manipulation of data
library(dplyr) # used for data manipulation and joining
library(ggplot2) # used for plotting
library(caret) # used for modeling
library(corrplot) # used for making correlation plot
library(xgboost) # used for building XGBoost model
library(cowplot) # used for combining multiple plots

train<-read.csv("C:/Users/Naman/Desktop/DV_Dataset/train_v9rqX0R.csv")
test<-read.csv("C:/Users/Naman/Desktop/DV_Dataset/test_AbJTz2l.csv")

dim(train)
dim(test)

names(train)
names(test)
```

```
str(train)
str(test)

combi <- rbind(train[1, 11], test)
combi <- rbind(test, train[, names(test)])
combi
```

Code for combining training and testing dataset this is helpful if we want to make any change in dataset then we don't have to change in test and training dataset, it saves a lot of time.

```
dim(combi)

ggplot(train)+geom_histogram(aes(Item_Outlet_Sales),binwidth=100,fill="darkgreen")+xlab("Item_Outlet_Sales")

p1 = ggplot(combi) + geom_histogram(aes(Item_Weight), binwidth = 0.5, fill = "blue")
p2 = ggplot(combi) + geom_histogram(aes(Item_Visibility), binwidth = 0.005, fill = "blue")
p3 = ggplot(combi) + geom_histogram(aes(Item_MRP), binwidth = 1, fill = "blue")
plot_grid(p1, p2, p3, nrow = 1)

ggplot(combi %>% group_by(Item_Fat_Content) %>% summarise(Count=n())) +
geom_bar(aes(Item_Fat_Content,Count), stat="identity", fill="coral")

combi$Item_Fat_Content[combi$Item_Fat_Content == "LF"] = "Low Fat"
combi$Item_Fat_Content[combi$Item_Fat_Content == "low fat"] = "Low Fat"
combi$Item_Fat_Content[combi$Item_Fat_Content == "reg"] = "Regular"
ggplot(combi %>% group_by(Item_Fat_Content) %>% summarise(Count=n())) +
geom_bar(aes(Item_Fat_Content,Count), stat="identity", fill="coral")

p4=ggplot(combi %>% group_by(Item_Type) %>% summarise(Count=n()))
+geom_bar(aes(Item_Type,Count), stat="identity",
fill="coral")+xlab("")+geom_label(aes(Item_Type,Count,label=Count),vjust=0.5)+theme(axis.text.x=element_text(angle=45,hjust=1))+ggtitle("Item_Type")
p4

p5=ggplot(combi %>% group_by(Outlet_Identifier) %>% summarise(Count=n())) +
geom_bar(aes(Outlet_Identifier,Count), stat="identity",
fill="coral")+geom_label(aes(Outlet_Identifier,Count,label=Count),vjust=0.5)+theme(axis.text.x=element_text(angle=45,hjust=1))
p5
```

```
p6=ggplot(combi %>% group_by(Outlet_Establishment_Year) %>% summarise(Count=n()))+
geom_bar(aes(Outlet_Establishment_Year,Count), stat="identity",
fill="coral")+geom_label(aes(Outlet_Establishment_Year,Count,label=Count),vjust=0.5)+theme
(axis.text.x=element_text(angle=45,hjust=1))
p6

p7=ggplot(combi %>% group_by(Outlet_Type) %>% summarise(Count=n())) +
geom_bar(aes(Outlet_Type,Count), stat="identity",
fill="coral")+geom_label(aes(Outlet_Type,Count,label=Count),vjust=0.5)+theme(axis.text.x=ele
ment_text(angle=45,hjust=1))
p7

p8 = ggplot(train) + geom_point(aes(Item_Weight, Item_Outlet_Sales), colour = "violet", alpha
= 0.3) +theme(axis.title = element_text(size = 8.5))
p8

sm = sum(is.na(combi$Item_Weight))
nonNullVal = nrow(combi) - sm
missing_index = which(is.na(combi$Item_Weight))
for(i in missing_index){
  item = combi$Item_Identifier[i]
  combi$Item_Weight[i] = sum(combi$Item_Weight, na.rm = T)/nonNullVal
}

ggplot(combi) + geom_histogram(aes(Item_Visibility), bins=100)

zero_index = which(combi$Item_Visibility == 0)
for(i in zero_index){ item = combi$Item_Identifier[i]
combi$Item_Visibility[i] = mean(combi$Item_Visibility[combi$Item_Identifier == item],
                                na.rm = T)
}

corMatrix<-cor(combi[1:nrow(train),][sapply(combi[1:nrow(train),],is.numeric)])
corMatrix

corrplot::corrplot(corMatrix, method="number", type="upper")
corrplot::corrplot(corMatrix, method="number", type="upper", order="hclust")
corrplot::corrplot(corMatrix, method="number", type="upper")
corrplot::corrplot(corMatrix, method="number", type="upper", order="hclust")
```

```
##########
ggplot(train[1:nrow(train),],aes(Item_Visibility, Item_Outlet_Sales))+geom_point(size=2.5,

aes(colour=factor(Outlet_Identifier)))+theme(axis.text.x=element_text(angle = 70,vjust =
0.5,color = "black")) +
  xlab("Item Visibility")+ylab("Item Outlet Sales")+ggtitle("Item Sales Vs Item Visibility")


ggplot(train[1:nrow(train),],aes(x=Item_Type,y=Item_Outlet_Sales,fill=Outlet_Type))+geom_b
oxplot()+theme(axis.title.x =

element_text(angle=70,vjust=0.5,color = "black"))+xlab("Item
Type")+ylab("Sales")+ggtitle("Sales vs Item type")

##########

ggplot(train,aes(Item_Type,Item_Outlet_Sales/Item_MRP,fill=Outlet_Type))+geom_bar(stat =
"identity",position =
                                                                   position_dodge())+theme(axis.title.x =
element_text(angle=70,vjust = 0.5,color = "black"))+xlab("Item Type")+ylab("Item Outlet
Sales")+ggtitle("Item Sales vs Item type")


##########

ggplot(train[1:nrow(train),],aes(Item_Visibility,Item_Outlet_Sales))+geom_point(size=2.5,

aes(colour=factor(Outlet_Identifier)))+theme(axis.text.x=element_text(angle = 70,vjust =
0.5,color = "black"))+
  xlab("Item Visibility")+ylab("Item Outlet Sales")+ggtitle("Item Sales Vs Item Visibility")

##########
ggplot(train[1:nrow(train),],aes(x=Item_Type,y=Item_Outlet_Sales,fill=Outlet_Type))+geom_b
oxplot()+
  theme(axis.text.x=element_text(angle = 70,vjust = 0.5,color = "black"))+xlab("Item
type")+ylab(" Sales")+
  ggtitle("Sales Vs Item Visibility")
##########
```

```
ggplot(train,aes(Item_Type,Item_Outlet_Sales/Item_MRP,fill=Outlet_Type))+ geom_bar(stat =
"identity",position = position_dodge())+
  theme(axis.title.x = element_text(angle=70,vjust = 0.5,color = "black"))+xlab("Item Type")+
  ylab("Item Outlet Sales")+ggtitle("Item Sales vs Item type")
```

PYTHON CODE:

```
import pandas as pd #import numpy as np

train=pd.read_csv("train_v9rqX0R.csv", na_values={"Item_Visibility":[0]})
test=pd.read_csv("test_AbJTz2l.csv", na_values={"Item_Visibility":[0]})

train['source']='train'
test['source']='test'
data=pd.concat([train,test],ignore_index=True)

#the one thing we have to focus is item_outlet_Sales
discpt=data.describe()

#Lets find out how many zero'es values are
nan_descript=data.apply(lambda x: sum(x.isnull()))

#Now lets find out the unique values in each of the catogorical columns
uniq=data.apply(lambda x: len(x.unique()))

#let do grouping in each catogorical columns
col=["Item_Fat_Content","Item_Type","Outlet_Location_Type","Outlet_Size"]
for i in col:
    print("The frequency distribution of each catogorical columns is--" + i+"\n")
    print(data[i].value_counts())
```

```
#Replacing the minimum nan values in the Item_Weight with its mean value
data.fillna({"Item_Weight":data["Item_Weight"].mean()},inplace=True)

#checking the current status of nan values in the dataframe
nan_descript=data.apply(lambda x: sum(x.isnull()))

#Now we have 0 nan values in Item_Weight
data["Outlet_Size"].fillna(method="ffill",inplace=True)
nan_descript=data.apply(lambda x: sum(x.isnull()))

#Now working on the item_visibility
visibilty_avg=data.pivot_table(values="Item_Visibility", index="Item_Identifier")
itm_visi=data.groupby('Item_Type')
data_frames=[]
for item,item_df in itm_visi:
   data_frames.append(itm_visi.get_group(item))
for i in data_frames:
   i["Item_Visibility"].fillna(value=i["Item_Visibility"].mean(),inplace=True)
   i["Item_Outlet_Sales"].fillna(value=i["Item_Outlet_Sales"].mean(),inplace=True)
new_data=pd.concat(data_frames)
nan_descript=new_data.apply(lambda x: sum(x.isnull()))
new_data

#Now we have successfully cleaned our complete dataset.
new_data["Item_Fat_Content"].replace({'LF':'Low Fat','reg':'Regular','low fat':'Low
Fat'},inplace=True)
new_data["Item_Fat_Content"].value_counts()

#Implementing one-hot-Coding method for getting the categorical variables
from sklearn.preprocessing import LabelEncoder
le = LabelEncoder()
```

```
data=new_data
data['Outlet'] = le.fit_transform(data['Outlet_Identifier'])
var_mod = ['Item_Fat_Content','Outlet_Location_Type','Outlet_Size','Item_Type','Outlet_Type']
le = LabelEncoder()
for i in var_mod:
    data[i] = le.fit_transform(data[i])

data = pd.get_dummies(data,
columns=['Item_Fat_Content','Outlet_Location_Type','Outlet_Size','Outlet_Type',
'Item_Type'])

#Exporting the datas
train = data.loc[data['source']=="train"]
test = data.loc[data['source']=="test"]

#Drop unnecessary columns:
test.drop(['Item_Outlet_Sales','source'],axis=1,inplace=True)

#Here we are dropping the "Item_Outlet_Sales because this only we want to be predicted from
the model that we are going to built
train.drop(['source'],axis=1,inplace=True)

#Export files as modified versions:
train.to_csv("train_modified.csv",index=False)
test.to_csv("test_modified.csv",index=False)

#Let's start building the baseline model as it is non -predicting model and also commonly known
as informed guess
#Mean based:
mean_sales = train['Item_Outlet_Sales'].mean()
```

```
#Define a dataframe with IDs for submission:
base1 = test[['Item_Identifier','Outlet_Identifier']]
base1['Item_Outlet_Sales'] = mean_sales

#Export submission file
base1.to_csv("alg0.csv",index=False)

#Define target and ID columns:
target = 'Item_Outlet_Sales'
IDcol = ['Item_Identifier','Outlet_Identifier']

import numpy as np
import sklearn
from sklearn.model_selection import cross_validate
from sklearn import metrics
from sklearn.model_selection import cross_val_score
from sklearn.model_selection import train_test_split
def modelfit(alg, dtrain, dtest, predictors, target, IDcol, filename):
    #Fit the algorithm on the data
    alg.fit(dtrain[predictors], dtrain[target])

    #Predict training set:
    dtrain_predictions = alg.predict(dtrain[predictors])

    #Perform cross-validation:
    cv_score = sklearn.model_selection.cross_val_score(alg, dtrain[predictors], dtrain[target],
cv=20, scoring='neg_mean_squared_error')
    cv_score = np.sqrt(np.abs(cv_score))

    #Print model report:
    print ("\nModel Report")
```

```
    print ("RMSE : %.4g" % np.sqrt(metrics.mean_squared_error(dtrain[target].values,
dtrain_predictions)))
    print ("CV Score : Mean - %.4g | Std - %.4g | Min - %.4g | Max - %.4g" %
(np.mean(cv_score),np.std(cv_score),np.min(cv_score),np.max(cv_score)))

    #Predict on testing data:
    dtest[target] = alg.predict(dtest[predictors])

    #Export submission file:
    IDcol.append(target)
    submission = pd.DataFrame({ x: dtest[x] for x in IDcol})
    submission.to_csv(filename, index=False)

#Linear Regression model
print("Creating the models and processing")
from sklearn.linear_model import LinearRegression, Ridge
predictors = [x for x in train.columns if x not in [target]+IDcol]
# print predictors
alg1 = LinearRegression(normalize=True)
modelfit(alg1, train, test, predictors, target, IDcol, 'alg1.csv')
coef1 = pd.Series(alg1.coef_, predictors).sort_values()
coef1.plot(kind='bar', title='Model Coefficients')

#Ridge Regression Model
predictors = [x for x in train.columns if x not in [target]+IDcol]
alg2 = Ridge(alpha=0.05,normalize=True)
modelfit(alg2, train, test, predictors, target, IDcol, 'alg2.csv')
coef2 = pd.Series(alg2.coef_, predictors).sort_values()
coef2.plot(kind='bar', title='Model Coefficients')
print("Model has been successfully created and trained. The predicted result is in alg2.csv")
```

```
#Decision Tree Model
from sklearn.tree import DecisionTreeRegressor
predictors = [x for x in train.columns if x not in [target]+IDcol]
alg3 = DecisionTreeRegressor(max_depth=15, min_samples_leaf=100)
modelfit(alg3, train, test, predictors, target, IDcol, 'alg3.csv')
coef3 = pd.Series(alg3.feature_importances_, predictors).sort_values(ascending=False)
coef3.plot(kind='bar', title='Feature Importances')
print("Model has been successfully created and trained. The predicted result is in alg3.csv")
```