**PROJECT REPORT**

**PREDICTING SALES AND IMPROVING PROFITABILITY OF RETAIL CHAIN**

*Submitted towards partial fulfillment of the criteria*

*for award of PGPBA by GLIM*

**Submitted By:**

**Hari Prasad A** [BACJAN17017]

**Kathir Rajalingam** [BACJAN17024]

**Pavan Kumar G** [BACJAN17044]

**Ravindranadh Yelugula** [BACJAN17056]

Course & Batch: **PGPBA JAN 2017**

Research Supervisor: **Dr. Bharadhwaj**

**Great Lakes Institute of Management**



Dr. Bala V. Balachandar Campus   
East Coast Road, Manamai Village

Kancheepuram District   
Tamil Nadu - 603102

**Acknowledgements**

The success and final outcome of this project required a lot of guidance and assistance from many people and we are extremely privileged to have got this all along the completion of our project.We owe our deep gratitude to our project guide Dr.Bhardwaj, who took keen interest on our project work and guided us all along, till the completion of our project work by providing all the necessary information.

We are thankful to and fortunate enough to get constant encouragement, support and guidance from all Teaching and non-teaching staffs of Greatlakes institute of management that helped us in successfully completing our project work. Nonetheless, constant support and sacrifices from family enabled us to complete the project on time.

Last but not the least we wish to thank Prof. P.K.Vishwanathan, our course Director, for constant supervision, guidance and for being a source of inspiration in helping us to work on this project.

We certify that the work done by us for conceptualizing and completing this project is original and authentic.

Date: November 15, 2017 HariPrasad.A

Place: Chennai Kathir Rajalingam

Pavan Kumar.G

Ravindranadh.Y

**Certificate of Completion**

I hereby certify that the project titled “Predicting Sales and improving profitability of retail chain” was undertaken and completed under my supervision by Hari Prasad,Pavan Kumar,Kathir Rajalingam and Ravindranath students of 2017 batch of Postgraduate Program in Business Analytics (PGPBAJAN2017).

Date: November, 2017 (Dr. Bharadwaj)

Place: Chennai Mentor

Table of Contents

[Synopsis 6](#_Toc498549613)

[KEYWORDS 7](#_Toc498549614)

[Introduction 8](#_Toc498549615)

[Dataset Description 9](#_Toc498549616)

[Features 10](#_Toc498549617)

[Missing Value Treatment 11](#_Toc498549618)

[Methodology 13](#_Toc498549619)

[Exploratory Descriptive Analysis 13](#_Toc498549620)

[Sales – Time Series Analysis 16](#_Toc498549621)

[Check for stationarity: 17](#_Toc498549622)

[Forecasting the sales 17](#_Toc498549623)

[Exponential Time Smoothing 18](#_Toc498549624)

[Holt winters Model 18](#_Toc498549625)

[ARIMA Model 19](#_Toc498549626)

[TBATS Model 19](#_Toc498549627)

[NNETAR Model 20](#_Toc498549628)

[MAPE as Accuracy Measure 20](#_Toc498549629)

[Hypothesis 21](#_Toc498549630)

[Promotional Activity 22](#_Toc498549631)

[Clustering of Stores 23](#_Toc498549632)

[Forecasting Sales for Next Quarter 26](#_Toc498549633)

[Predicting Sales product Wise 27](#_Toc498549634)

[Product wise Forecast 28](#_Toc498549635)

[Neural Net 29](#_Toc498549636)

[Conclusion 31](#_Toc498549637)

[Appendix A 32](#_Toc498549638)

[R Code 34](#_Toc498549639)

[#Read Data From Work File 34](#_Toc498549640)

[#Missing Value Treatment 35](#_Toc498549641)

[#Time Series of Sales (Spend) of Retail chain 36](#_Toc498549642)

[# Differenced Series & Autocorrelation 36](#_Toc498549643)

[# Anova for Display and Sales 37](#_Toc498549644)

[#Choosing Forecasting model~ Predict next 8 future values 37](#_Toc498549645)

[#Exponential model 37](#_Toc498549646)

[#Holt-Winters Model - Tuned with R code 37](#_Toc498549647)

[#Tuning model 37](#_Toc498549648)

[#Holt Winter's Forecast 38](#_Toc498549649)

[#Arima Model 38](#_Toc498549650)

[#TbATS Model 38](#_Toc498549651)

[#NNETS Model 38](#_Toc498549652)

[# Building the Rawdata frame - complete data 39](#_Toc498549653)

[# Forecast of the next 12 Weeks overall Spends for the company 39](#_Toc498549654)

[# reading spend data as TimeSeries 39](#_Toc498549655)

[# model implementation 39](#_Toc498549656)

[# Checking for missing values - 3 fields have missing data 40](#_Toc498549657)

[# Fixing 1.1 BasePrice - replace with the weekly product average spend 40](#_Toc498549658)

[# Fixing 1.2 Price: replace by the weekly product average spend 40](#_Toc498549659)

[# Creating Discount based features & Modifying Volume units 41](#_Toc498549660)

[# re-defining features as factors 41](#_Toc498549661)

[# re-calculating the product size to milli scale 41](#_Toc498549662)

[# Store Wise Clustering 41](#_Toc498549663)

[# Optimal centers for Clustering 42](#_Toc498549664)

[# Clustering 42](#_Toc498549665)

[# Cluster visualization 43](#_Toc498549666)

[# Promotion Significance Product wise 43](#_Toc498549667)

[# Productwise Spend forecast for next 12 Weeks 44](#_Toc498549668)

[#TbATS Model 45](#_Toc498549669)

[#NNETS Model 45](#_Toc498549670)

[# Top Products Promotion Effectiveness 46](#_Toc498549671)

[# Top Products Sales Forecast 47](#_Toc498549672)

[References 49](#_Toc498549673)

# Synopsis

The capstone project deals with analysis of retail transaction data of a retail chain located at four states of the United States of America, that includes the sales, promotion information for multiple products and brands with different categories for the past 156 weeks. The key challenge is to derive decision that would enable the retail chain to identify the customer preferences out from the high volume of data.

The objective of the study to predict the sales for the future, address the relation between price and sales. Also, estimate the impact on sales by changing the price gaps between items, promotions, displays and feature. Also, address price cushion on increase in margin to retailer that would increase the sales. At the end, considerable effort should be spent to identify the products that would enhance the sales and margin to the retailer.

The capstone would address the problem with

* Data clean-up (identifying outliers and ways to treat them)
* Exploratory data analysis (identifying key factors)
* Forecasting the sales by time series analysis model (week wise)
* Develop models that enable to take the decision based on key retailer, by product and by category wise
* Measurable insights to the retail chain by identifying the profitable promotional mechanism
* Classify the store based on relevant factors
* Collective decision based on study that would enable retail chain to improve their profitability.

# KEYWORDS

**Techniques**: Time series prediction, Promotion effectiveness, Store clustering

**Tools**: R

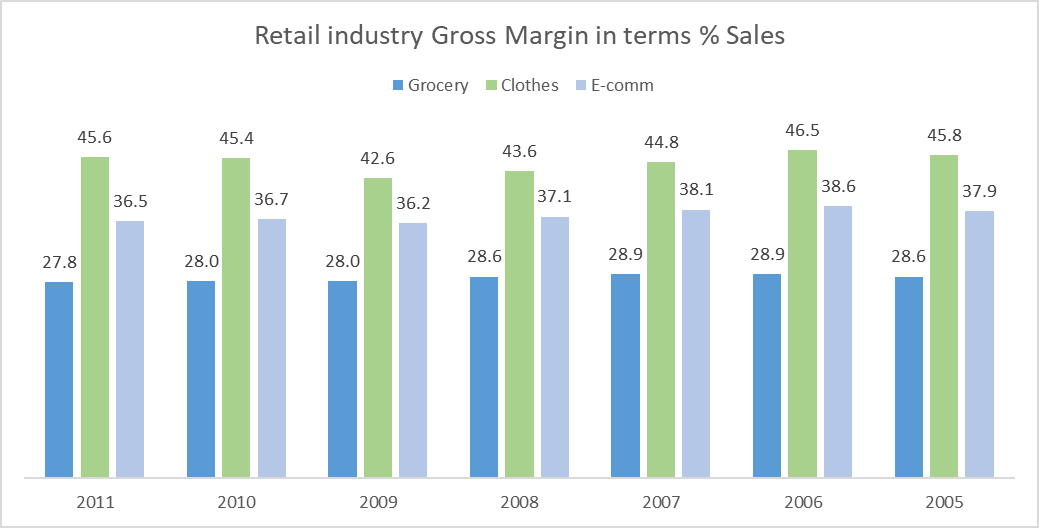
**Domain**: Retail analytics, Product Analysis

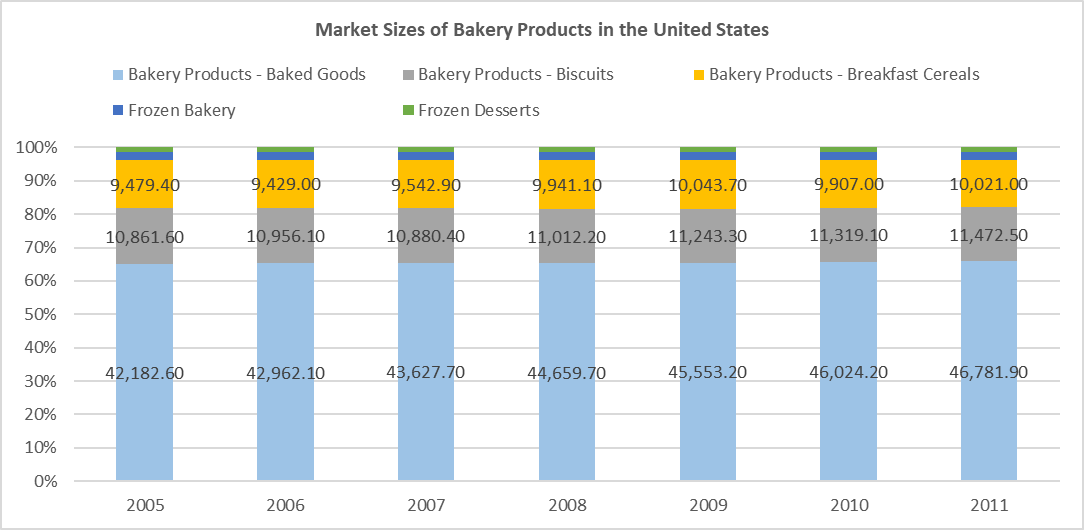
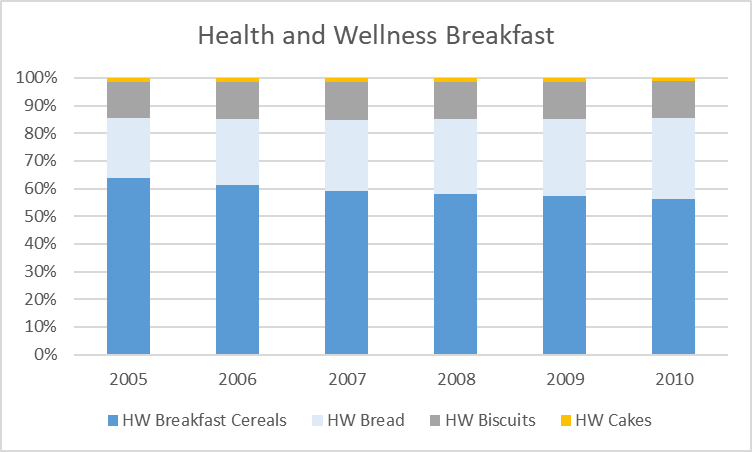
**ML Methods**: KNN , Linear regression, ANOVA, ARIMA, TBATS, NNETAR, Holt Winters

## Introduction

This project deals with analysis of retail transaction data of a retail chain located at four states of the United States of America, that includes the sales, promotion information for multiple products and brands with different categories for the past 156 weeks.

US Retail industry and its growth during 2005-2011

 The retail industry in united states is highly competitive esp. in food and beverage (grocery) stores command comparatively low gross margin with reference to clothes and e-commerce industry, with reference to Annual Retail Trade Survey1(Fig 1). To survive in such competitive environment, the brick and mortar retail must be innovative and competitive to stay in the market. Hence the management of this retail chain wish to classify the stores, value of each products they are delivering and thus enhance the margin to the retailer.

The dataset generated out of transaction/sales (spends by the customer) done over the year that highly depends on the customer preference of retail stores over the competition. The entire dataset filtered to products that used for prior and during breakfast. The figure 2, market size of bakery product in the United States of the America shows that bakery products (65%)2 dominate the consumption during breakfast. Also, it reinforces that consumption pattern of breakfast is not changed for the past during 2005-2011.But the drive of health and wellness among the consumers added another segment i.e. Health and wellness breakfast made an internal change in the subcategory. The health and wellness breakfast comprises cereals, bread, biscuits and cakes, the study shows that cereals losing their placing with breads. The breakfast shares for cereals decreased from 62 % to 57%, evident from fig 3.

In the prevailing condition, the retailer wish to understand the consumption/spending pattern of their consumers in five different states with past three-year data and preferably store the consumers choice of the product in the shelves.

The objective of this study to predict the sales for the future for different products, test hypothesis to address the relation between price and discount, price thresholds and price elasticity of the products. The study should estimate the impact on sales by changing the price gaps between items, promotions, displays and feature. Also, address price cushion on increase in margin to retailer that would increase the sales. At the end, considerable effort should be spent to identify the products that would enhance the sales and margin to the retailer.

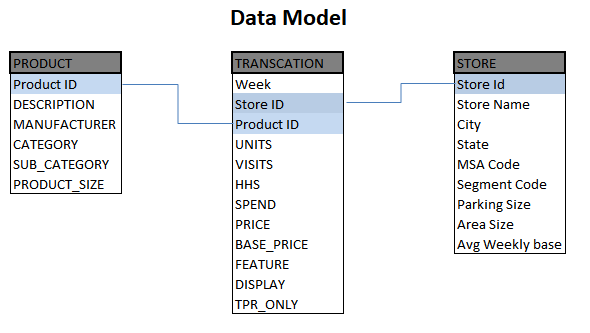
# Dataset Description

The dataset comprises Weekly aggregated product level sales transaction data at different stores located at Four different states of the united states of America. It contains the 55 products (unique product code wise), price, baseline and customer spend pattern on weekly basis for past 156 weeks.

The dataset aggregated in terms of product wise transaction with reference to UPC (unique product code) in date time line, details of the store in which the transaction made, details of the product with details on packaging, brand and manufacturer. There were few categorical variables such as product, category, sub category and Store id. There were many continuous variables such as Spend (Sale), sales areas and price.

## Features

|  |  |  |  |
| --- | --- | --- | --- |
| VARIABLE NAME | DESCRIPTION | DATA TYPE | CATEGORIES |
| ADDRESS\_CITY\_NAME | City | Categorical | 51 |
| ADDRESS\_STATE\_PROV\_CODE | State | Categorical | 4 |
| AVG\_WEEKLY\_BASKETS | average weekly baskets sold in the store | Continuous |  |
| BASE\_PRICE | base price of item | Continuous |  |
| MANUFACTURER | manufacturer | Categorical | 16 |
| CATEGORY | category of product | Categorical | 4 |
| DESCRIPTION | product description | Categorical | 52 |
| DISPLAY | product was a part of in-store promotional display | Categorical | 2 |
| FEATURE | product was in in-store circular | Categorical | 2 |
| HHS | # of purchasing households | Continuous |  |
| MSA\_CODE | (Metropolitan Statistical Area) geographic region with a high core population density and close economic ties throughout the surrounding areas | Categorical | 9 |
| PARKING\_SPACE\_QTY | number of parking spaces in the store parking lot | Continuous |  |
| PRICE | actual amount charged for the product at shelf | Continuous |  |
| WEEK\_END\_DATE | week ending date | Continuous |  |
| SALES\_AREA\_SIZE\_NUM | square footage of store | Continuous |  |
| STORE\_APPEAL | Retailer's designated store appeal | Categorical | 73 |
| SPEND | total spend (i.e., $ sales) | Continuous |  |
| STORE\_NUM | store number | Categorical | 77 |
| SUB\_CATEGORY | sub-category of product | Categorical | 7 |
| TPR\_ONLY | temporary price reduction only (i.e., shelf tag only, product was reduced in price but not on display or in an advertisment) | Categorical | 2 |
| UNITS | units sold | Continuous |  |
| UPC | (Universal Product Code) | Categorical | 55 |
| VISITS | number of unique purchases (baskets) that included the product | Continuous |  |
| PRODUCT\_SIZE | package size or quantity of product | Categorical | 28 |



# Missing Value Treatment

The data has 23 observations with price value, 185 observations with base price value and 52 observations with parking values missing. We have aggregated the data to product wise on weekly basis and took the average value to replace the missing price and base price values. For missing parking values we have consider them with a 0 replacement as most of the store may not have dedicated parking slots, they might be in a spared area.

# Methodology

## Exploratory Descriptive Analysis

A screenshot of a social media post

Description generated with very high confidenceThe sales of the retail chain were as shown in the figure. The retail chain commands average sale of 9600K /year. There is no predominant seasonality as seen in the chart. But this need to be ascertained with suitable hypothesis.

In line with US population choice, cereals are most preferred food for breakfast in these five states too. This is evident from the product preference chart and sub category.

A screenshot of a social media post

Description generated with very high confidenceA screenshot of a social media post

Description generated with very high confidenceA screenshot of a social media post

Description generated with very high confidenceThe brand penetration is highly evident from the chart shown below, the leaving the private lables and general misc brands, the sales of the retail chain is more than 80% contributed by major brands such as Kellogs,Quarker and Tombstone. This shows that population in the five states were highly concerned about the brands they consume. Apart A picture containing screenshot

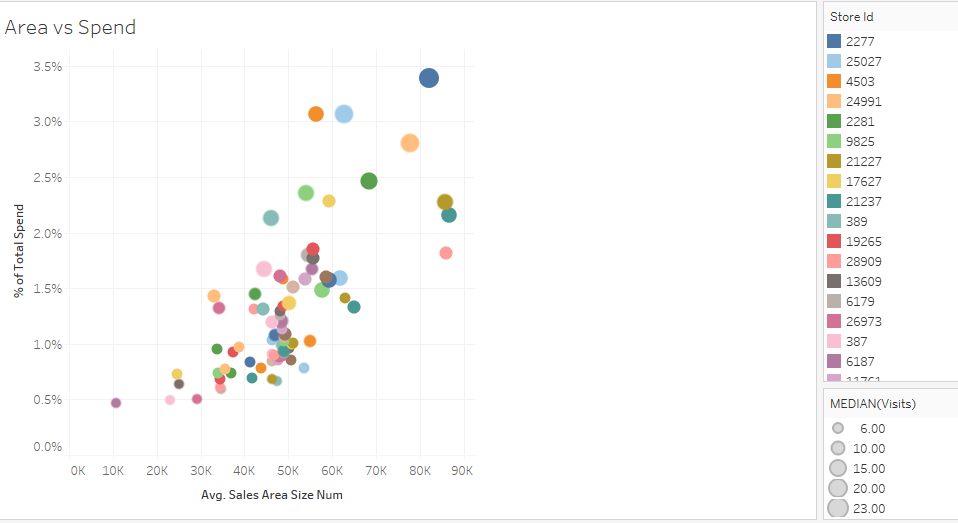
Description generated with very high confidencefrom major brands, while chosing the cold cereals, snacks and frozen pizza, consumers doesnot seem to worry about the brand and select their choice in the private labels. This augments by the picture shown below.

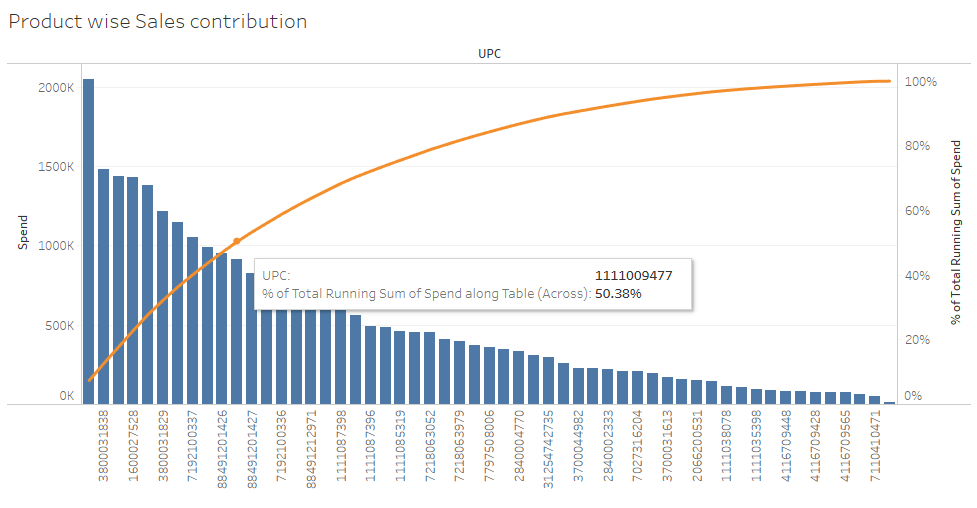
A close up of a map

Description generated with high confidenceA screenshot of a cell phone

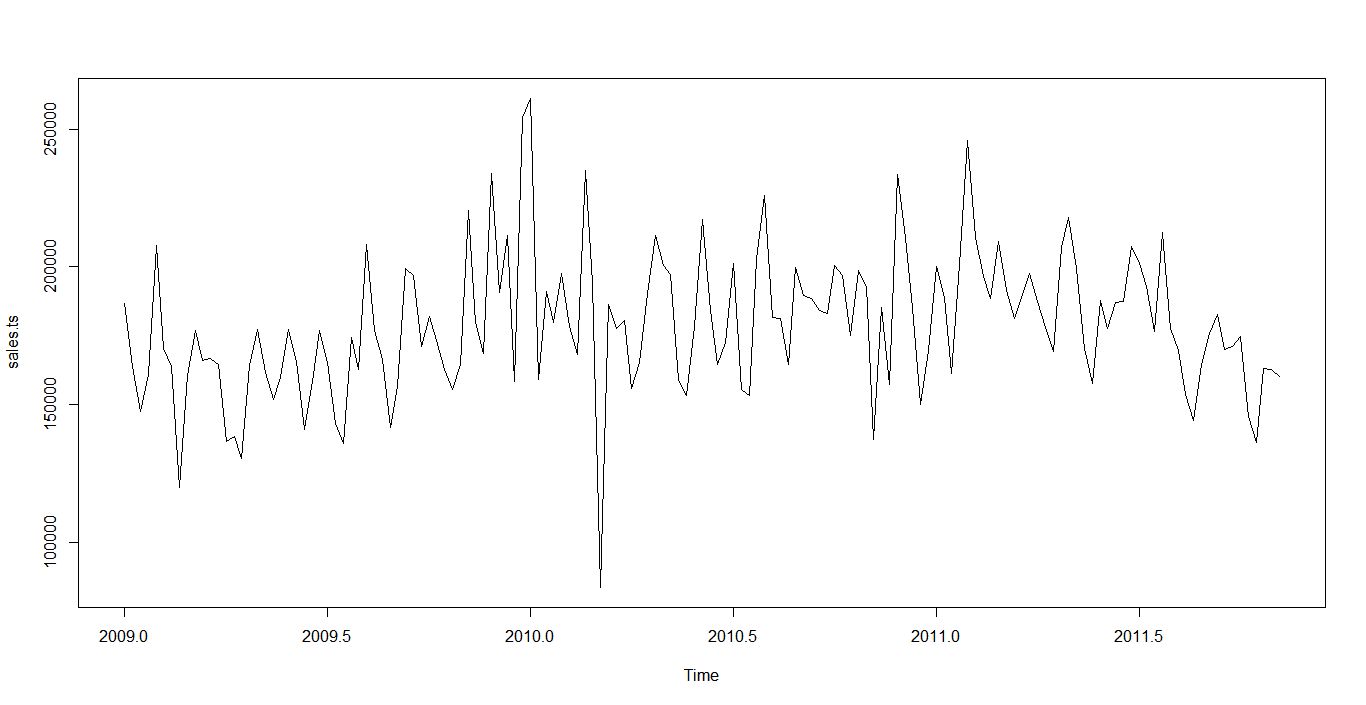
Description generated with very high confidenceBased on retailer choice, the top 10 retailers command better share in the sales of the retail chain. The sale of the retail chain primarily dependant on visits made by each customer . if the retailer has median visits more than 9 times , then the sales contributed would be 50% more than median sales. The customer visit also inherent function of sales area of the shop and average weekly basket size. There exist a linear relation between the number of visits and sales generated by the customer and this needs to be studied with suitable hypothesis. Those are evident from the above figures.

Another inference, if the store size or sale area greater than 70k Sq ft, it attracts the customers in to spend more and sales in those stores in the magnitude of 2X to 3X greater than average sale. There possible reasons could be lot of space for promotional mechanism such as instore display . This needs to be studied in detail with suitable hypothesis.





# Sales – Time Series Analysis

The transaction data for the sales happened during the period 14-01-2009 to 14-01-2012 has been collected. The transaction, the spend (sales) is aggregated sum on a weekly basis .the weekwise sales data for the retail chain collated and shown as in the figure attached. Following is the time series representaion of sales. (The blue para has been written twice by now)

## Check for stationarity:

A stationary time series is one whose statistical properties such as mean, variance, autocorrelation, etc. are all constant over time. The ADF (Augmented Dickey-Fuller Test) study on the sales data shows that reject the null hypothesis at 95% confidence level) (p<0.05 – Significant at 5%), hence it supports the conclusion that the data is stationary.

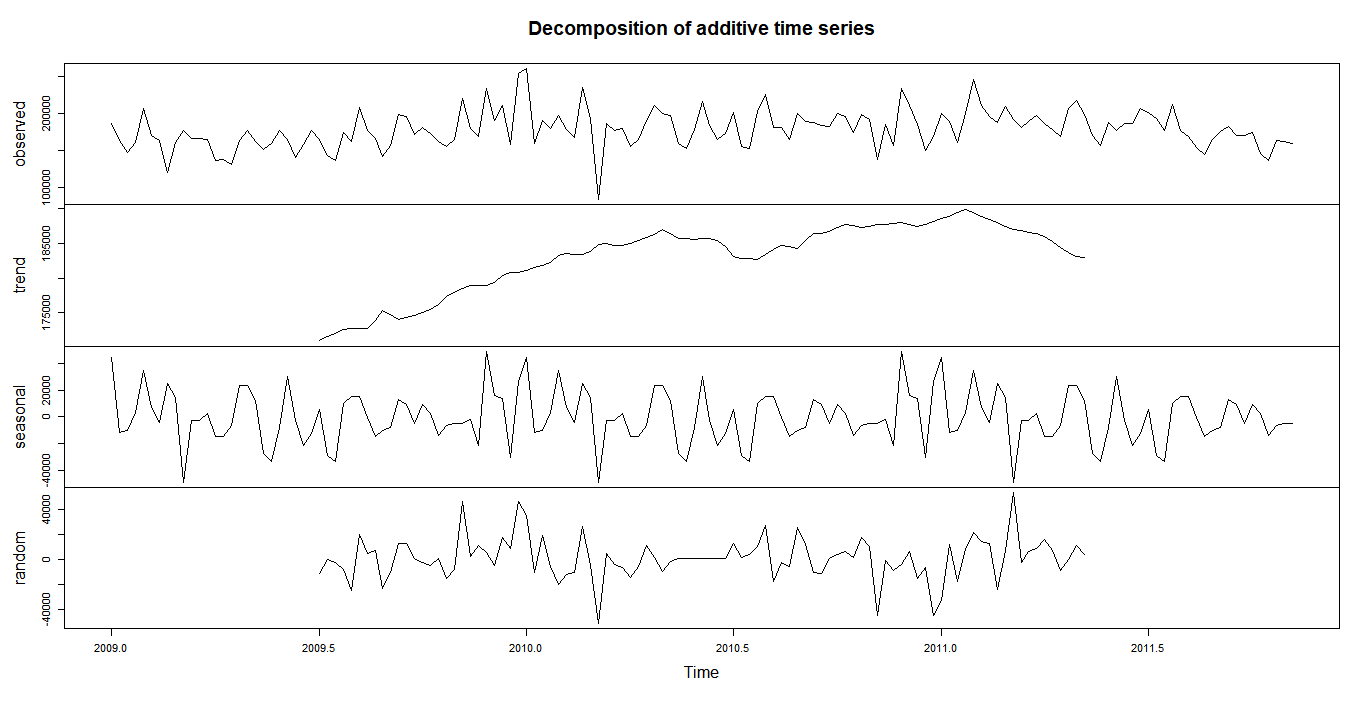
daftest(sales.ts/scale(sales.ts-sales.comp$seasonal), alternative = "stationary")

Augmented Dickey-Fuller Test

data: sales.ts/scale(sales.ts - sales.comp$seasonal)

Dickey-Fuller = -4.7267, Lag order = 5, p-value = 0.01

alternative hypothesis: stationary



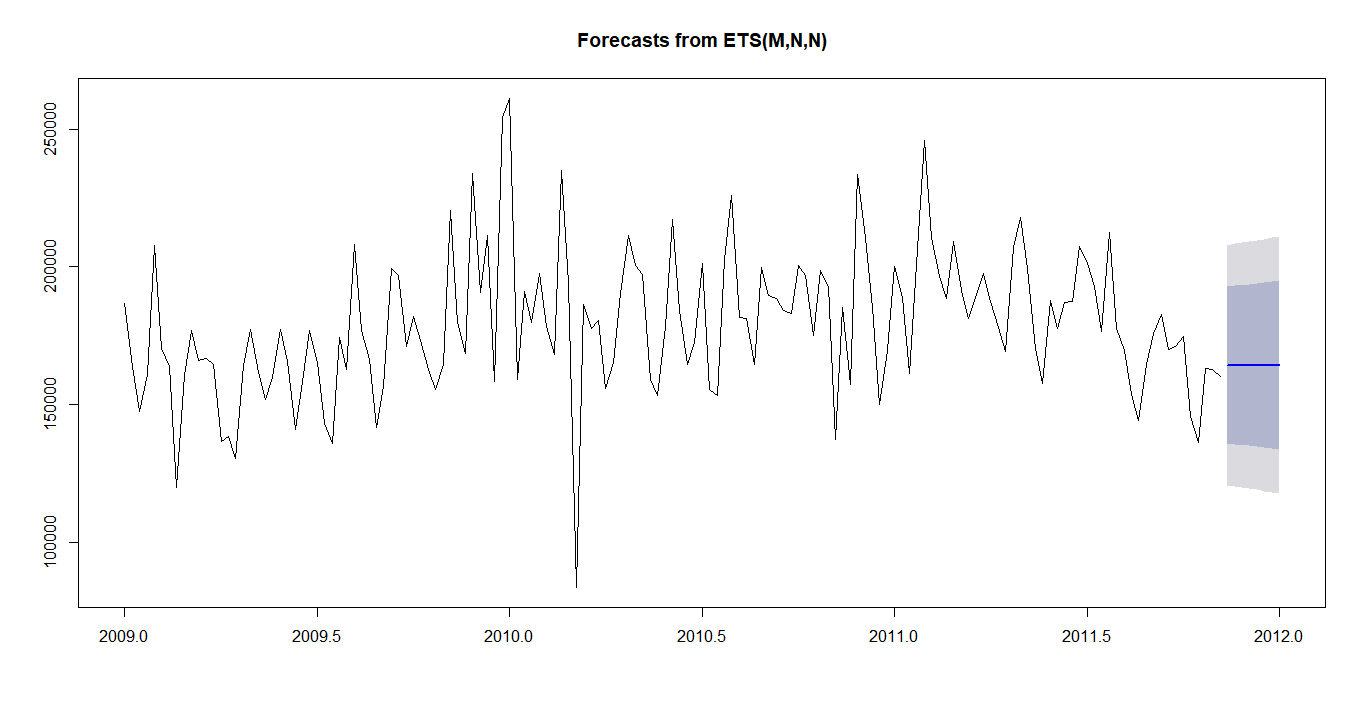
## Forecasting the sales

Forecasting the retail chain sales, is a crucial part in getting the things for the successful operations management. There are several forecasting model available and best models need to be selected based on minimal MAPE (Mean absolute percentage error). To name few critical

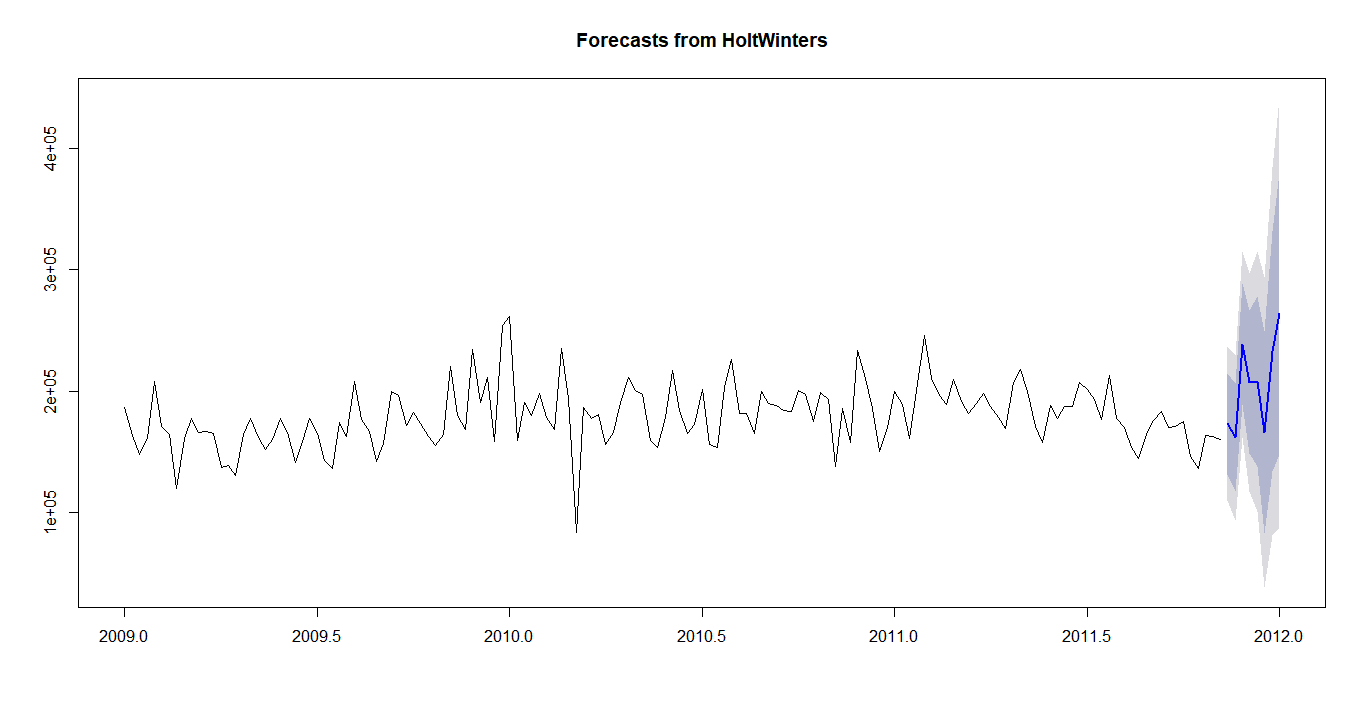
1. Exponential Time Smoothing
   1. Holt Winters
2. ARIMA - Autoregressive integrated moving average
3. TBATS -Exponential smoothing state space model with Box-Cox transformation
4. NNETAR – Neural Network Time Series

Two subsets were formed out of the time series data of sales, first 145 weeks were considered for the training set and last 8 weeks were considered to be test set. Based on the MAPE of each model and stability of the model in predicting the next weeks, the best model would be selected.

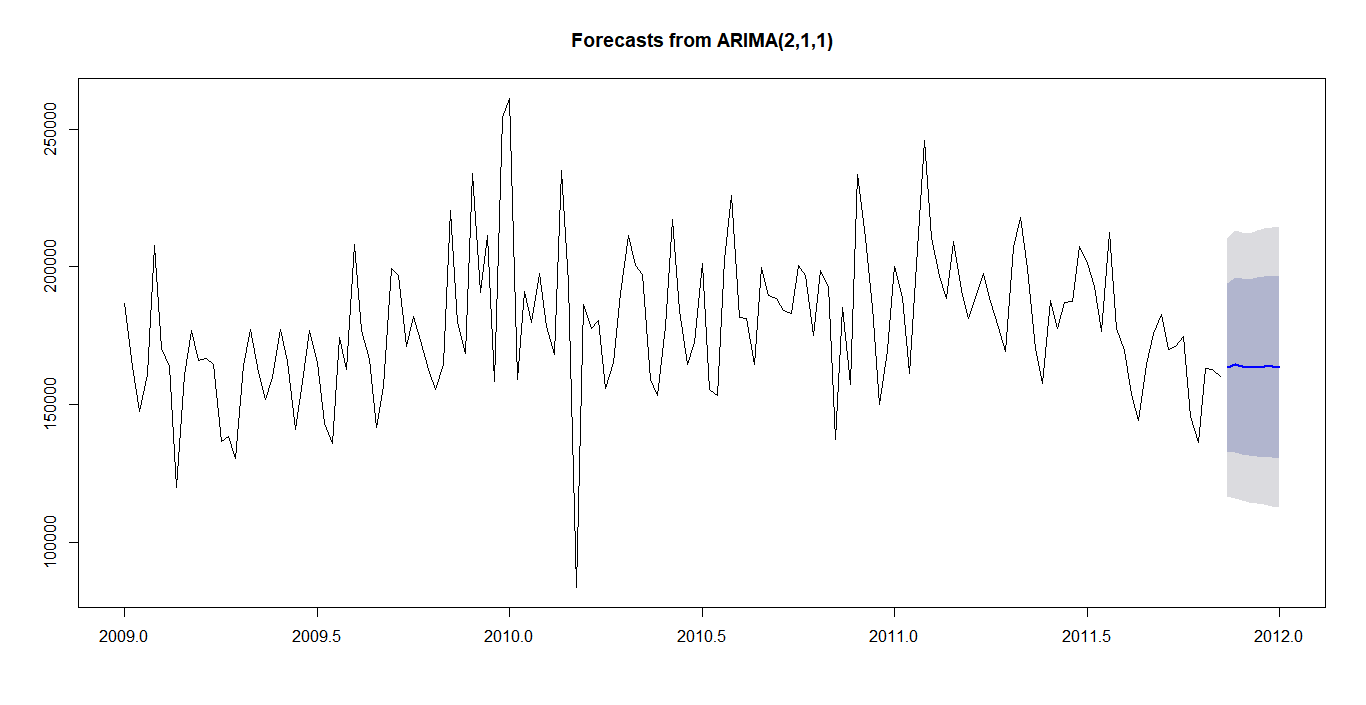
### Exponential Time Smoothing



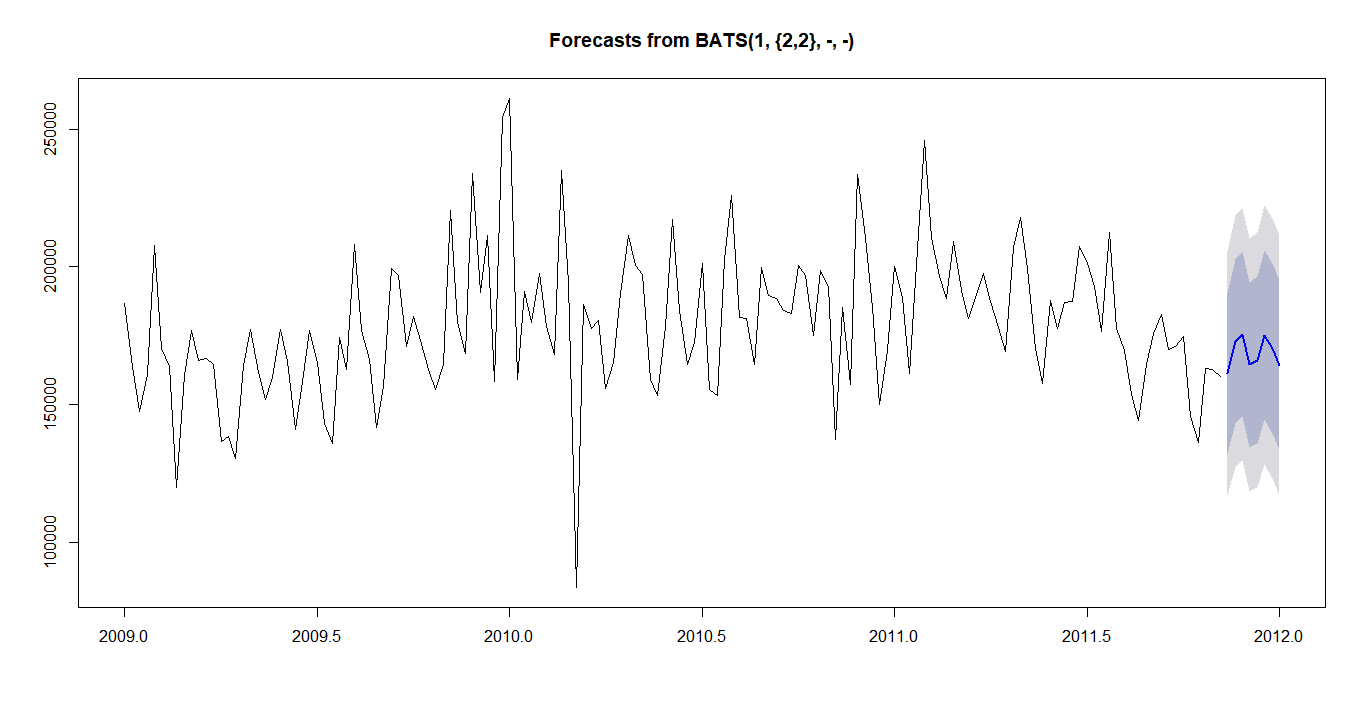
### Holt winters Model



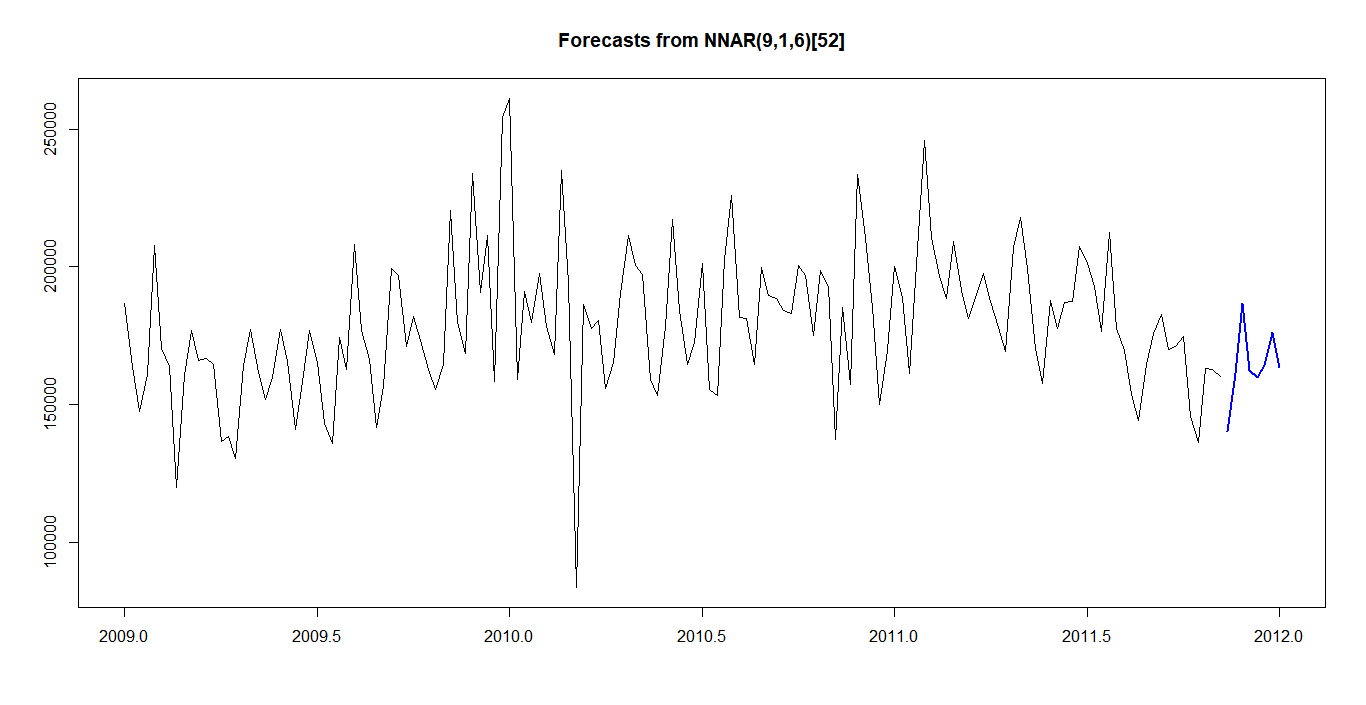
### ARIMA Model



### TBATS Model



### NNETAR Model



### MAPE as Accuracy Measure

Different models are verified with test set with 8-week sales data), the MAPE are listed for different model

|  |  |  |
| --- | --- | --- |
|  |  | MAPE |
| ETS | Training | 10.26999 |
| Test | 10.92934 |
| HW | Training | 11.46126 |
| Test | 32.92964 |
| ARIMA | Training | **9.952777** |
| Test | **10.87441** |
| TBATS | Training | 9.386718 |
| Test | 12.99714 |
| NN | Training | 1.109472 |
| Test | 15.54247 |

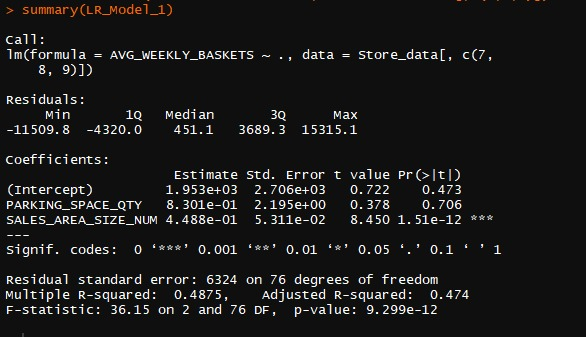
The ARIMA model is so stable with Training data as well as Test data, though Neural Net model is good in accuracy in terms of MAPE with training data set and have very high MAPE with Test data. Hence the NNETAR model could not be considered.

# Hypothesis

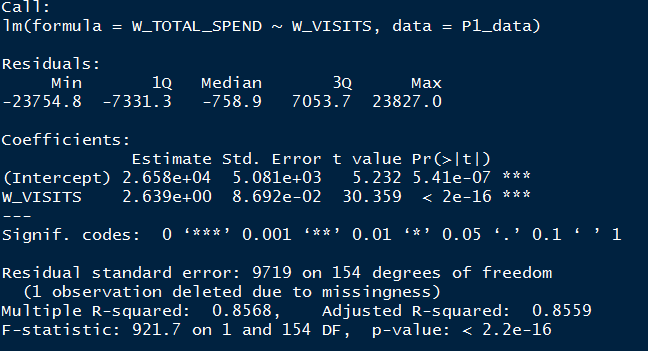
Hypothesis 1: The parking space doesn’t influence the total sales of stores

Hypothesis 2: The sale area size of retail shop doesn’t improve the total sales

The hypothesis was tested to identify whether parking space and area of the store would influence the total sales. Linear regression formed with average weekly basket as function of parking space available and sale area size. The hypothesis test results brought out the surprising outcome that

* Sales area size is a significant contributor to the average weekly basket. Larger the sale area would contribute to larger weekly basket. It means the super stores involve more sales to the retail chain
* Parking space is not significant factor to the weekly basket size, means that individual stores located independently would command a separate parking, the retail chain in the mall or outlet would not have specified parking. This pretty much aligned with living style of US people. 

Hypothesis 3: Number of Visits doesn’t increase the sales, the linear regression between the number of visits that enabled the sales transaction for all products have been conducted. The study brought the outcome that the stores that command higher footfalls/visits have higher sales.



# Promotional Activity

The promotional activity is the key driver for incremental sales in addition to the base sales. The sales (spend) is a function of base sales and incremental sales over the week. This would primarily test whether the promotional activity can aid the sales to improve.

Hypothesis 4: The promotional activity doesn’t improve the sales.

There were three key promotional activities were followed to increase the sales over the time.

1. Feature
2. Display
3. Temporary price reduction

*Feature* is a promotional activity, coupons and details about the offers were shown/printed in the magazine, and this would enable customers to pick their choice*.*

*Display* is another promotional activity, in which the products are showcased using in-store display

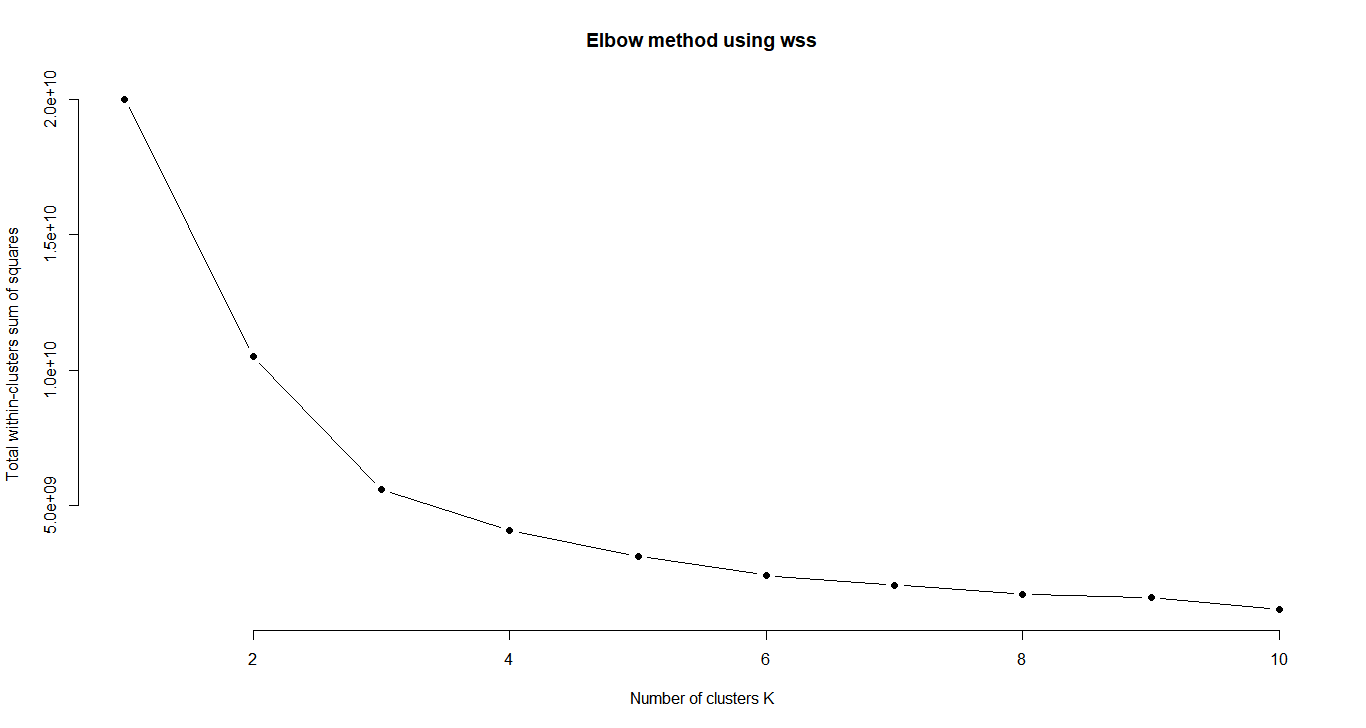
*Temporary price reduction only (TPR Only),* the products are given discounts in the price for a short period, it was shown only on the items. This promotion is highly specific to the stores may not be implemented in all the stores.

This project dealt with activities that analyze the critical promotional mechanism for the chosen product across all the stores. The analytical technique that employed is regression with 5% significance level. The product wise classification is detailed in the table; this would enable the retailer to identify the most significant promotional mechanism for any chosen product. Detailed product wise promotional mechanism listed in Appendix -A. The promotional mechanism for key products as follows

The promotion effectiveness of the product, defined as the ratio to the **incremental sales to promotional sales**. Higher the effectiveness, attributes higher the sensitivity to the promotion drive. The product that command higher sales and higher promotion effectiveness would yield better prospect for the sale.

# Clustering of Stores

Clustering of stores enables us to identify the like stores based on parameters with values in close neighborhood. Campaigns and store layout strategy and the kind of products with which the shelves could be stacked can be suggested based on the cluster. We have used KNN clustering method. This gave us three clusters which is evident from the below elbow graph.

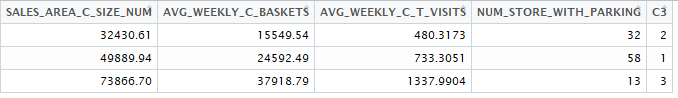


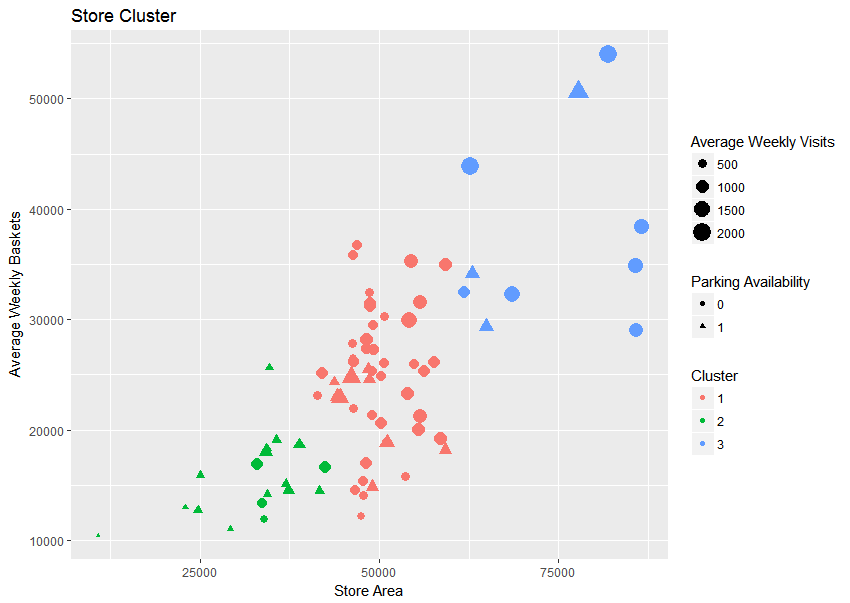
This helped us to zero down on four important parameters namely,

* Store size
* Average baskets in store per weekend
* Weekly visits
* Parking area

Based on the above key parameters the stores were divided as three clusters of which store size being the predominant variable in segregating the clusters.

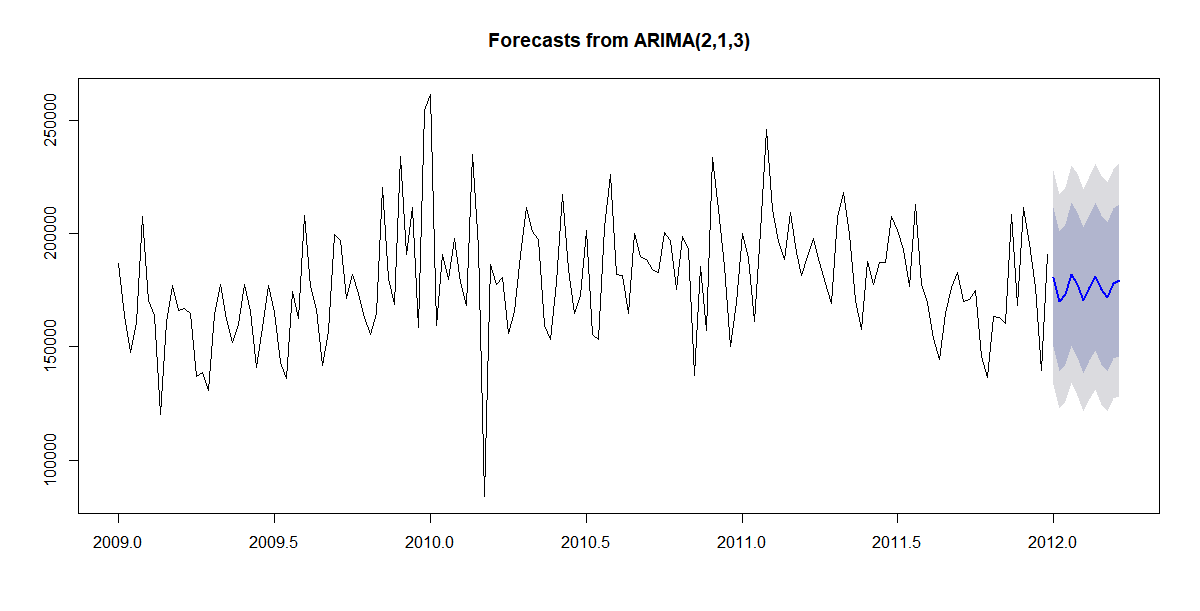
1. Stores with sales area less than 40,000 sq. ft. had parking availability in minority and the average number of baskets ranged from 10-20 thousand/week as “Value stores”
2. Sales area between 40,000-60,000 sq. ft. has contributed majorly to the sales where parking availability has improved which contributed to 12-35 thousand weekly baskets as “Super Stores”
3. Rest were large stores with humungous sales area and ample parking space.Theses can be termed as “Premium Stores”





# Forecasting Sales for Next Quarter

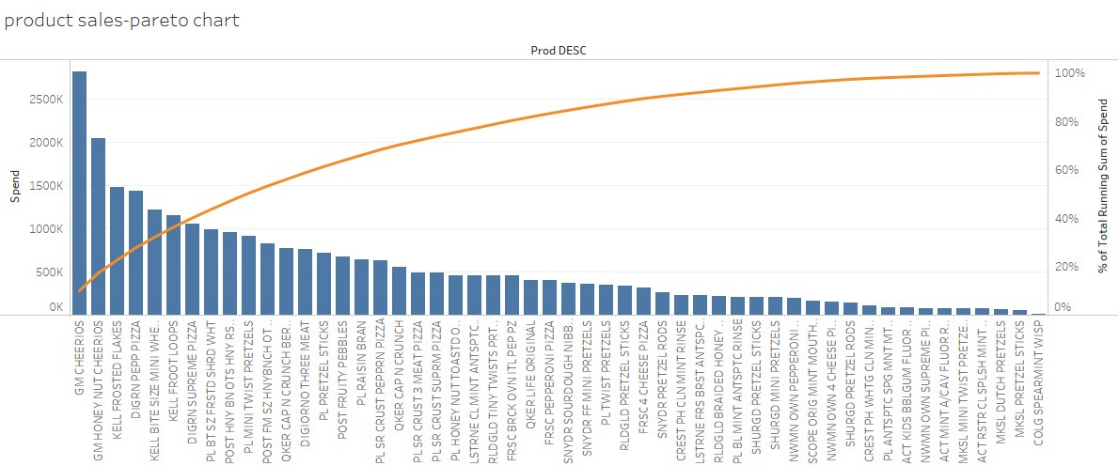
The total sales for the next 12 weeks have been forecasted using ARIMA and Neural Net AR.

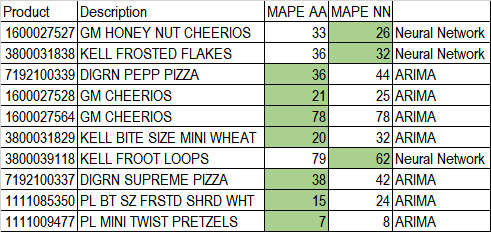


|  |  |  |  |
| --- | --- | --- | --- |
| Week No | Forecast | % Change | Focus |
| 1 | 180794.2 |  |  |
| 2 | 169866.1 | -6% | Week of focus |
| 3 | 173122.2 | 2% |  |
| 4 | 182080.3 | 5% |  |
| 5 | 177392.1 | -3% |  |
| 6 | 170459.4 | -4% | Week of focus |
| 7 | 175996.4 | 3% |  |
| 8 | 180973.4 | 3% |  |
| 9 | 175086.9 | -3% | Week of focus |
| 10 | 171902.7 | -2% | Week of focus |
| 11 | 177732.3 | 3% |  |
| 12 | 179349.9 | 1% |  |

It is evident from the above, week 3rd, 7th, 10th, 11th needs an attention to stop the dip in the sales, appropriate candidate for promotions. In other weeks, with the increasing footfalls the promotion should be aimed to increase the quantity.

# Predicting Sales product Wise

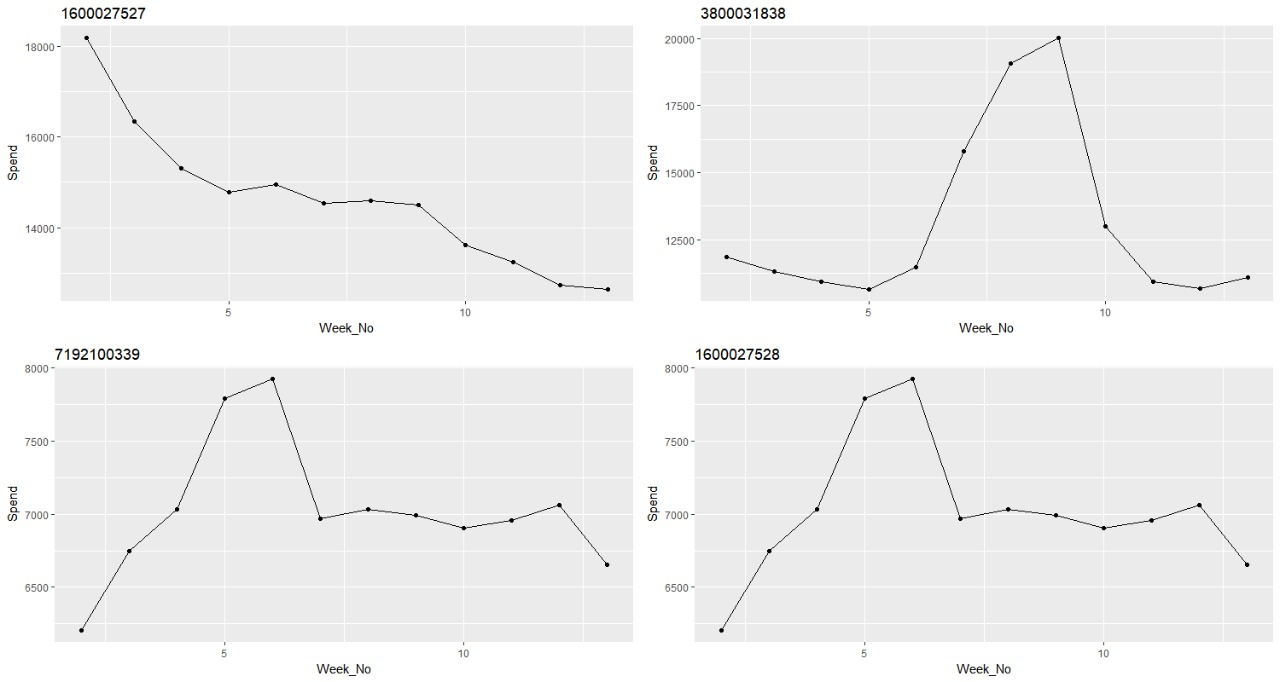
 Prediction or forecasting is the vital part of any time series analysis. There are 55 products listed as part of breakfast portfolio. The overall forecasting trend of the total sales would throw us many insights on how the sales performance, how to improve/support the sales. To get the detail to improve the sales, key 11 products that contribute to ~50% sales were identified and predicted the sales for next quarter.

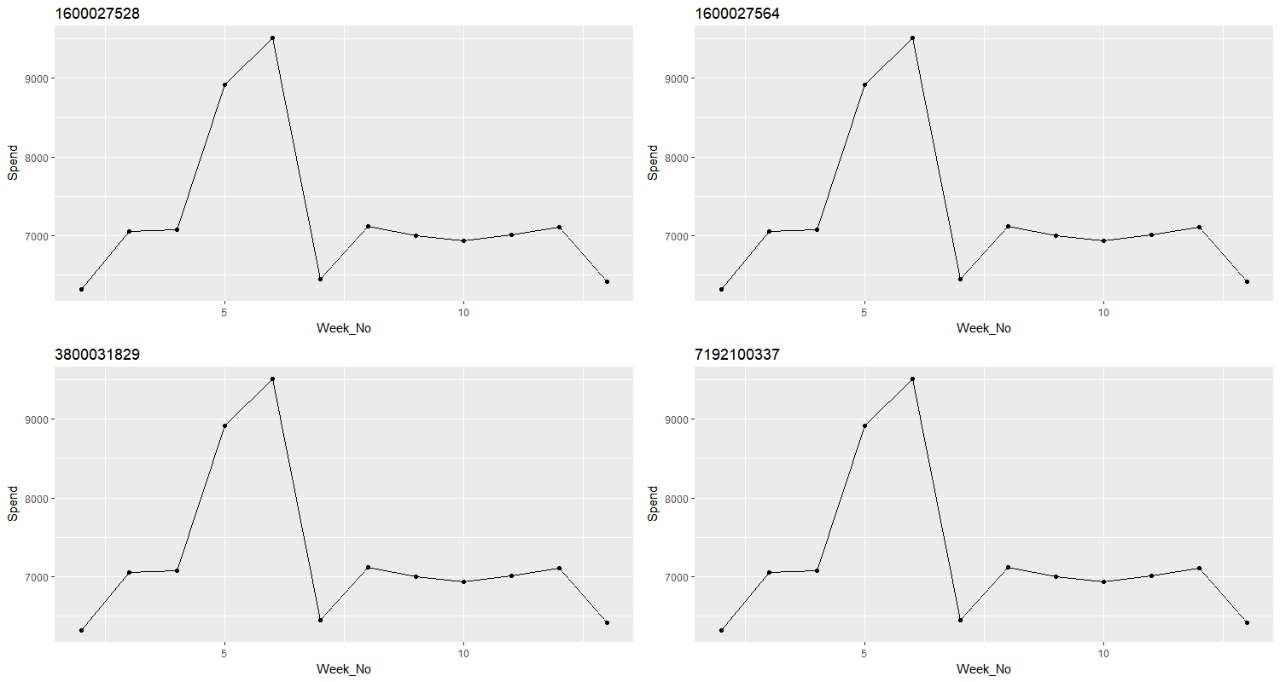


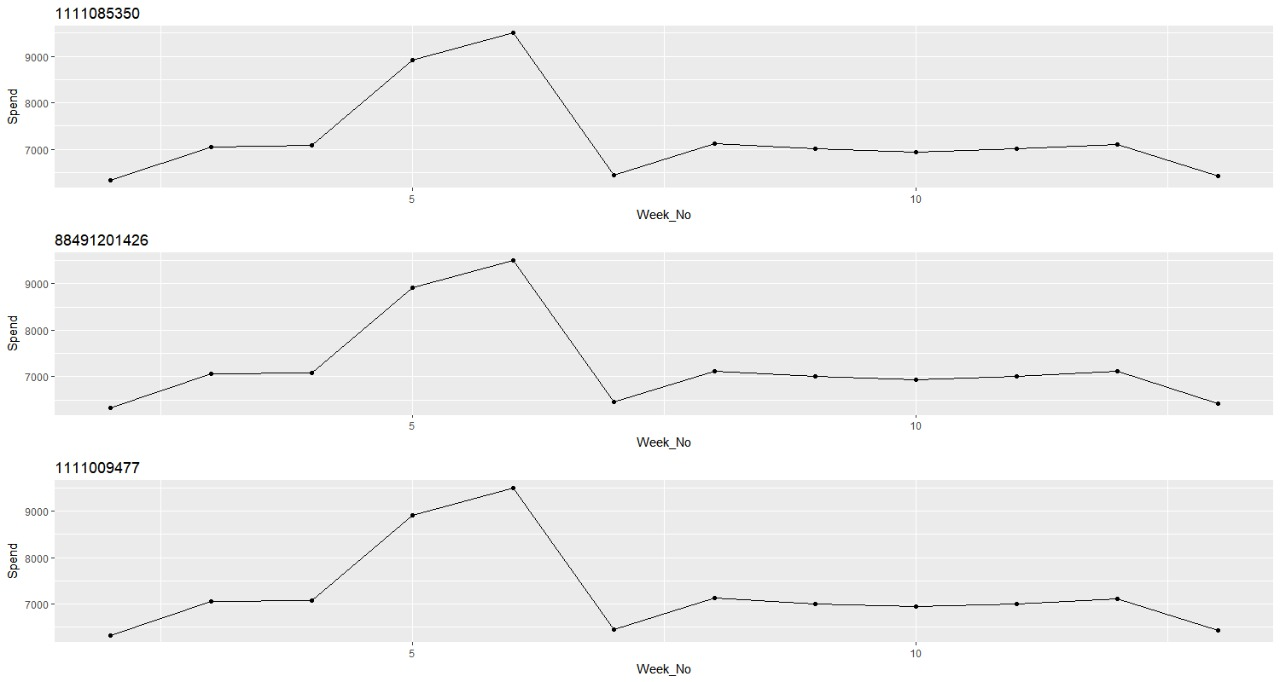
The products such as KELL FROOT LOOPS and GM CHEERIOS have higher amount of MAPE Error and forecast tend to have high variation. Hence it is advisable to have forecasted value based on Store request.

The forecasted charts are listed for key 11 products below and based on the sales, appropriate promotion mechanism could be implemented.

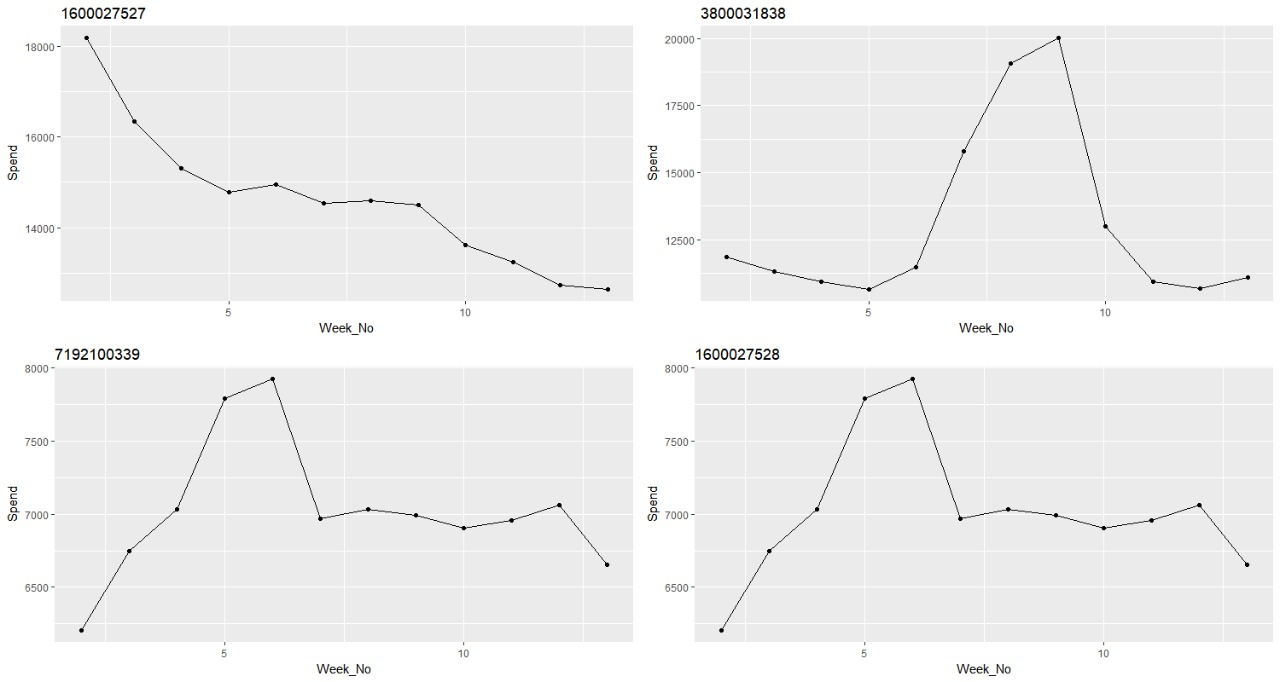
## Product wise Forecast

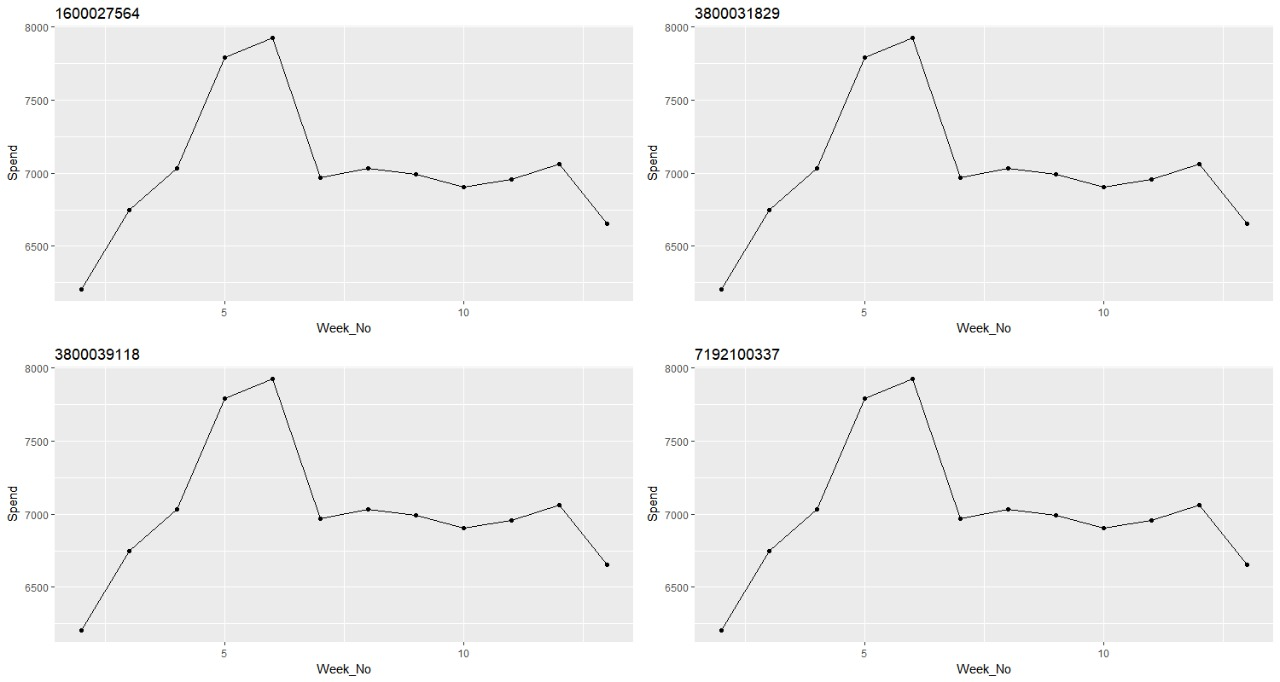


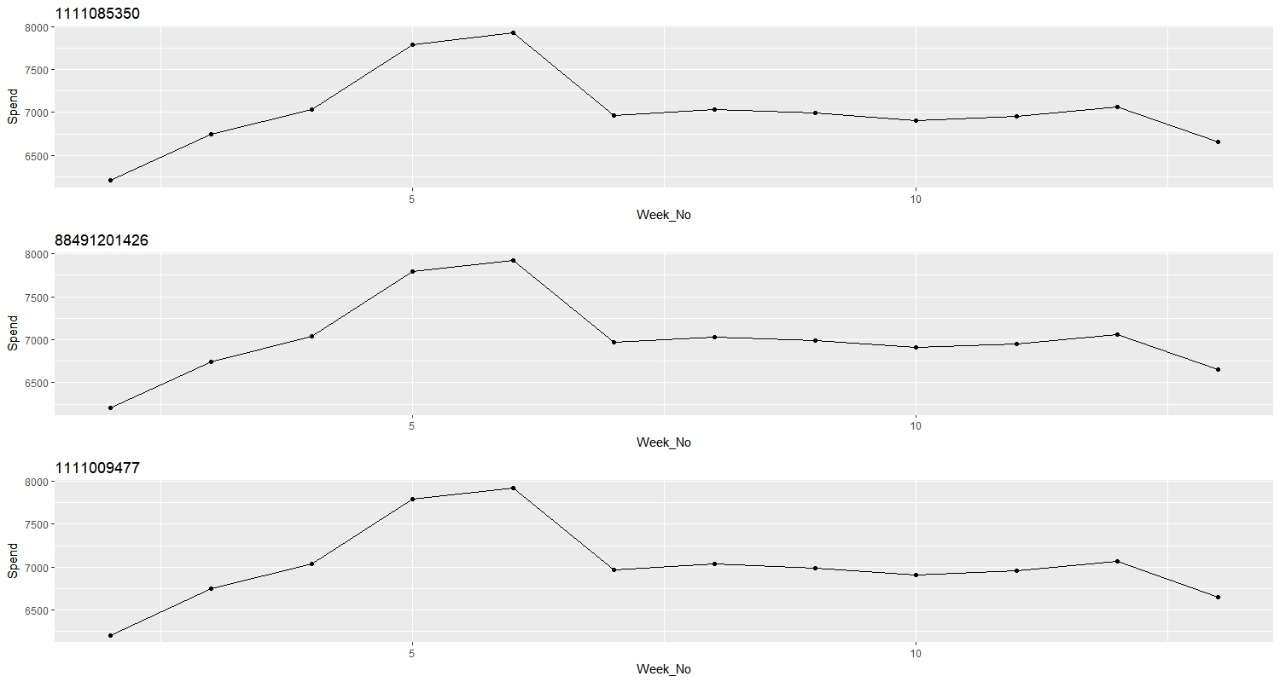




## Neural Net







# Conclusion

* Using Time series techniques, the sales of the complete retail chain on weekly basis completed with several models, better performing model based on MAPE was selected.
* The better model was ARIMA and Neural Net TS model with comparable forecast error, product wise models were compared and chosen with reference to the table
* With effective promotion planning week wise, the sales could be improved for the identified key 11 product. Special focus to be driven to make sure these products are part of customer’s basket.
* Stores classified using KNN and listed based on key parameters, such as parking area, store size, average baskets per week, average weekly visits while premium stores category contributing for more than 40% of sales.
* To improve the sales in the value stores, Since Visits are directly linked to sales we propose to improve the number of visits/foot falls by implementing attractive promotions or campaigns such as loyalty
* Slicing down to the product level and forecasting the same for the next quarter is also done. Based on this in forecasted weeks 3rd and 4th week on wards they can start pushing the top 3 products –mention their PE’s and also that they have forecasted dip during this – also that the top 2 are sig for 3 modes while the 3rd is only for TPR promo.
* Promotional effectiveness and method of promotion for all products have been listed, hence the retail chain can appropriately choose the effective promotional mechanism for any product in the portfolio
* For top selling products, Stores could instead transfer the same marketing expenditure on other insignificant methods such as feature and display to price discount which is visible at shelf level to further increase the sale.

# Appendix A

Products wise Promotional mechanism

The different ways of promotional methods for all the products in the breakfast portfolio. This would enable the retail chain to effectively promote their products in cost effective manner. The name F, D, T refers to the Feature, Display and Temporary price reduction.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Product Code** | **Product** | **F\_Sig** | **D\_Sig** | **T\_Sig** |
| 1111009477 | PL PRETZEL STICKS | F | D | T |
| 1111009497 | PL TWIST PRETZELS |  | D | T |
| 1111009507 | PL BL MINT ANTSPTC RINSE |  | D | T |
| 1111035398 | PL BL MINT ANTSPTC RINSE |  |  | T |
| 1111038078 | PL ANTSPTC SPG MNT MTHWS | F | D | T |
| 1111038080 | PL HONEY NUT TOASTD OATS | F | D |  |
| 1111085319 | PL RAISIN BRAN | F | D | T |
| 1111085345 | PL BT SZ FRSTD SHRD WHT |  | D | T |
| 1111085350 | PL SR CRUST SUPRM PIZZA | F | D | T |
| 1111087395 | PL SR CRUST 3 MEAT PIZZA | F | D | T |
| 1111087396 | PL SR CRUST PEPPRN PIZZA | F | D | T |
| 1111087398 | GM HONEY NUT CHEERIOS | F | D | T |
| 1600027527 | GM CHEERIOS | F | D | T |
| 1600027528 | GM CHEERIOS | F | D |  |
| 1600027564 | NWMN OWN PEPPERONI PIZZA | F | D | T |
| 2840004768 | NWMN OWN 4 CHEESE PIZZA | F | D | T |
| 3000006340 | NWMN OWN SUPREME PIZZA | F | D | T |
| 3000006560 | RLDGLD BRAIDED HONEY WHT | F | D | T |
| 3000006610 | RLDGLD TINY TWISTS PRTZL | F | D | T |
| 3700019521 | RLDGLD PRETZEL STICKS | F | D | T |
| 3700044982 | QKER LIFE ORIGINAL | F | D |  |
| 3800031829 | QKER CAP N CRUNCH BERRIES | F | D | T |
| 3800031838 | QKER CAP N CRUNCH | F | D | T |
| 3800039118 | HMRN CLSC SAUSAGE PIZZA | F | D | T |
| 4116709428 | HMRN CLSC SSG PEPP PIZZA |  |  | T |
| 7027316204 | HMRN CLSC CHS PIZZA |  |  | T |
| 7027316404 | COLG SPEARMINT WISP |  | D | T |
| 7192100336 | CREST PH WHTG CLN MINT TP | F | D | T |
| 7192100337 | SCOPE ORIG MINT MOUTHWASH | F | D | T |
| 7192100339 | CREST PH CLN MINT RINSE | F | D | T |
| 7218063052 | KELL BITE SIZE MINI WHEAT | F | D | T |
| 7218063979 | KELL FROSTED FLAKES | F | D | T |
| 7218063983 | KELL FROOT LOOPS | F | D | T |
| 7797508004 | ACT MINT A/CAV FLUOR RNS | F | D |  |
| 7797508006 | ACT KIDS BBLGUM FLUOR RNS | F | D |  |
| 31254742835 | ACT RSTR CL SPLSH MINT MW | F | D | T |
| 2840002333 | SHURGD PRETZEL RODS | F | D | T |
| 2840004770 | SHURGD MINI PRETZELS | F | D |  |
| 3700031613 | SHURGD PRETZEL STICKS | F | D | T |
| 4116709448 | MKSL MINI TWIST PRETZELS |  |  | T |
| 4116709565 | MKSL DUTCH PRETZELS |  |  |  |
| 7027312504 | MKSL PRETZEL STICKS |  | D | T |
| 7110410455 | DIGIORNO THREE MEAT |  | D | T |
| 7110410470 | DIGRN SUPREME PIZZA |  | D | T |
| 7110410471 | DIGRN PEPP PIZZA |  | D | T |
| 7797502248 | FRSC BRCK OVN ITL PEP PZ |  | D |  |
| 31254742725 | FRSC PEPPERONI PIZZA | F | D | T |
| 31254742735 | FRSC 4 CHEESE PIZZA | F | D | T |
| 3500068914 | SNYDR PRETZEL RODS |  |  | T |
| 88491201426 | SNYDR SOURDOUGH NIBBLERS | F | D |  |
| 88491201427 | SNYDR FF MINI PRETZELS | F | D |  |
| 88491212971 | LSTRNE CL MINT ANTSPTC MW | F | D | T |
| 2066200530 | LSTRNE CL MINT ANTSPTC MW |  | D | T |
| 2066200531 | LSTRNE FRS BRST ANTSPC MW | F | D | T |
| 2066200532 | POST HNY BN OTS HNY RSTD | F | D | T |

|  |
| --- |
| R Code#Read Data From Work File require(readxl) # Excel-File reader  require(dplyr) # Data Manupulation  require(forecast) # Time Series  require(tidyr) # splits a col into two based on the regular expression  require(car) # to set un-ordered factors ranking method, helmert - baseline is one method rest referring to it  require(dummies) ## to reacte dummy variables for factor data  require(ggplot2)  require(usdm)  # Building the Rawdata frame - complete data ------  excel\_sheets('dunnhumby - Breakfast at the Frat.xlsx')  rawdata = read\_excel(path = 'dunnhumby - Breakfast at the Frat.xlsx', sheet = 'dh Transaction Data')  rawStoreData = read\_excel(path = 'dunnhumby - Breakfast at the Frat.xlsx', sheet = 'dh Store Lookup')  rawdata = rawdata %>%  left\_join(rawStoreData[which(!duplicated(rawStoreData$STORE\_ID)),],  by = c('STORE\_NUM'='STORE\_ID')) %>%  left\_join(read\_excel(path = 'dunnhumby - Breakfast at the Frat.xlsx', sheet = 'dh Products Lookup'),  by = 'UPC')  rawdata$WEEK\_END\_DATE = as.Date(rawdata$WEEK\_END\_DATE)  # Forecast of the next 12 Weeks overall Spends for the company -----  # building data as per weekly total spend happened  company\_data = rawdata %>%  dplyr::select(WEEK\_END\_DATE,SPEND,FEATURE,DISPLAY,TPR\_ONLY,  BASE\_PRICE,PRICE) %>%  group\_by(WEEK\_END\_DATE) %>%  mutate(W\_TOTAL\_SPEND = sum(SPEND)) %>%  dplyr::select(WEEK\_END\_DATE,W\_TOTAL\_SPEND) %>%  unique() %>%  arrange(WEEK\_END\_DATE)  path=getwd()  data.work=read.csv("work1.csv")  data.work$timestamp=as.Date(data.work$WEEKEND\_DATE,format="%d-%b-%y")  data.work=data.work[-1]  require(dplyr)  new.df = data.work %>% group\_by(timestamp) %>% mutate(SPEND\_SUM = sum(SPEND))  subs.df=new.df[,25:26]  subs.new=unique(subs.df[,1:2]) #Missing Value Treatment # Checking for missing values - 3 fields have missing data ----  Missing\_data\_Check <- function(data\_set){  NA\_Count = sapply(data\_set,function(y) sum(length(which(is.na(y)))))  Null\_Count = sapply(data\_set,function(y) sum(length(which(is.null(y)))))  Length0\_Count = sapply(data\_set,function(y) sum(length(which(length(y)==0))))  Empty\_Count = sapply(data\_set,function(y) {if(class(y) != 'Date')  return(sum(length(which(y==''))))  else return(0)})  Total\_NonData = NA\_Count+Null\_Count+Length0\_Count+Empty\_Count  return( Total\_NonData )  }  Missing\_data\_Check(rawdata)  # Missing data handling ----  # Fixing 1.1 BasePrice - replace with the weekly product average spend  non\_baseprice\_missing\_rawdata = rawdata[,1:25][!apply(rawdata[,9], 1, function(x) any(is.na(x))),]  product\_WeeklyAvg\_basePrice = non\_baseprice\_missing\_rawdata %>%  dplyr::select(WEEK\_END\_DATE,UPC,BASE\_PRICE)%>%  group\_by(UPC,WEEK\_END\_DATE) %>%  mutate(AVG\_BASE\_PRICE= round(mean(BASE\_PRICE),2)) %>%  dplyr::select(WEEK\_END\_DATE,UPC,AVG\_BASE\_PRICE) %>%  unique()  baseprice\_missing\_rawdata = rawdata[,1:25][apply(rawdata[,9], 1, function(x) any(is.na(x))),]  baseprice\_missing\_rawdata = baseprice\_missing\_rawdata %>%  left\_join(product\_WeeklyAvg\_basePrice,  by=c("WEEK\_END\_DATE","UPC"))  baseprice\_missing\_rawdata$BASE\_PRICE = baseprice\_missing\_rawdata$AVG\_BASE\_PRICE  rawdata\_BasePrice = rbind(non\_baseprice\_missing\_rawdata,baseprice\_missing\_rawdata[-26])  rm(list = c('non\_baseprice\_missing\_rawdata','product\_WeeklyAvg\_basePrice','baseprice\_missing\_rawdata'))  # Fixing 1.2 Price: replace by the weekly product average spend  non\_price\_missing\_rawdata = rawdata\_BasePrice[,1:25][!apply(rawdata\_BasePrice[,8], 1, function(x) any(is.na(x))),]  product\_WeeklyAvg\_Price = non\_price\_missing\_rawdata %>%  dplyr::select(WEEK\_END\_DATE,UPC,PRICE)%>%  group\_by(UPC,WEEK\_END\_DATE) %>%  mutate(AVG\_PRICE= round(mean(PRICE),2)) %>%  dplyr::select(WEEK\_END\_DATE,UPC,AVG\_PRICE) %>%  unique()  price\_missing\_rawdata = rawdata\_BasePrice[,1:25][apply(rawdata\_BasePrice[,8], 1, function(x) any(is.na(x))),]  price\_missing\_rawdata = price\_missing\_rawdata %>%  left\_join(product\_WeeklyAvg\_Price,  by=c("WEEK\_END\_DATE","UPC"))  price\_missing\_rawdata$PRICE = price\_missing\_rawdata$AVG\_PRICE  rawdata\_BasePrice\_Price = rbind(non\_price\_missing\_rawdata,price\_missing\_rawdata[-26])  rm(list = c('non\_price\_missing\_rawdata','product\_WeeklyAvg\_Price','price\_missing\_rawdata','rawdata\_BasePrice'))  # Creating Discount based features & Modifying Volume units ----  rawdata\_BasePrice\_Price$DISCOUNT\_PRICE = rawdata\_BasePrice\_Price$BASE\_PRICE - rawdata\_BasePrice\_Price$PRICE  rawdata\_BasePrice\_Price$DISCOUNT\_PERCENT = (rawdata\_BasePrice\_Price$DISCOUNT\_PRICE/rawdata\_BasePrice\_Price$BASE\_PRICE)\*100  rawdata\_BasePrice\_Price$DISCOUNT = ifelse(rawdata\_BasePrice\_Price$DISCOUNT\_PRICE!=0,1,0)  # re-defining features as factors  rawdata\_factors = as.data.frame(unclass(rawdata\_BasePrice\_Price))  rawdata\_factors$FEATURE = as.factor(rawdata\_factors$FEATURE)  rawdata\_factors$DISPLAY = as.factor(rawdata\_factors$DISPLAY)  rawdata\_factors$TPR\_ONLY = as.factor(rawdata\_factors$TPR\_ONLY)  rawdata\_factors$DISCOUNT = as.factor(rawdata\_factors$DISCOUNT)  rawdata\_factors$MSA\_CODE = as.factor(rawdata\_factors$MSA\_CODE)  rm(rawdata\_BasePrice\_Price)  # re-calculating the product size to milli scale  unique(rawdata\_factors$PRODUCT\_SIZE)  # 1 oz = 29.5735 ml # 1 LT = 1000 ml # 1 ML = 1 ml # 1 CT = 200 mg  rawdata\_factors = extract(rawdata\_factors, PRODUCT\_SIZE, c("VOLUME", "V\_UNITS"), "([^ ]+) (.\*)")  rawdata\_factors$VOLUME=as.numeric(rawdata\_factors$VOLUME)  rawdata\_factors$PRODUct\_SIZE\_Milli = ifelse(rawdata\_factors$V\_UNITS=='ML', rawdata\_factors$VOLUME,  ifelse(rawdata\_factors$V\_UNITS=='CT',rawdata\_factors$VOLUME\*200,  ifelse(rawdata\_factors$V\_UNITS=='LT',rawdata\_factors$VOLUME\*1000,  ifelse(rawdata\_factors$V\_UNITS=='OZ',rawdata\_factors$VOLUME\*29.5735,-1))))  rawdata\_factors = rawdata\_factors[,-c(25,26)]  write.csv(rawdata\_factors,'rawdata\_factors.csv',row.names = F)   #Time Series of Sales (Spend) of Retail chain sales.ts=ts(subs.new[,-1],start =c(2009,1),end =c(2011,45),frequency = 52)  sales.ts1=ts(subs.new[,-1],start =c(2011,46),end =c(2012,1),frequency = 52)  plot(sales.ts)  sales.comp=decompose(sales.ts)  plot(sales.comp,title(main="Decomposition of Sales")) # Differenced Series & Autocorrelation d1=diff(sales.ts,lag = 1)  d2=diff(sales.ts, lag=2)  plot(d1)  auto.correl=acf(sales.ts, lag.max=50) # Autocorrelation  pa.correl=pacf(sales.ts, lag.max=10) # Partial autocorrelation  require(tseries)  adf.test(sales.ts/scale(sales.ts-sales.comp$seasonal), alternative = "stationary") # Anova for Display and Sales data.work$DISPLAY = as.factor(data.work$DISPLAY)  require(ez)  anova( aov(SPEND ~ DISPLAY, data=data.work) )  Mean\_data.work = data.work %>%  select(DISPLAY,SPEND) %>%  group\_by(DISPLAY) %>%  mutate (M = mean(SPEND, na.rm = T))  unique(Mean\_data.work[,c(1,3)]) #Choosing Forecasting model~ Predict next 8 future values#Exponential model sales.ets=ets(sales.ts)  sales.fs\_ets=forecast(sales.ets,h=8)  acc.ets=accuracy(sales.fs\_ets,sales.ts1)  plot(sales.fs\_ets)  acc.ets #Holt-Winters Model - Tuned with R code#Tuning model j=1  MAPE = numeric()  a1 = numeric()  b1 = numeric()  c1 = numeric()  sales.ts.dev = head(sales.ts,145)  sales.ts.hold = head(sales.ts,8)  for(a in seq(from=0.1,to= 0.9,by=0.1)){  for(b in seq(from=0.1,to= 0.9,by=0.1)){  for(c in seq(from=0.1,to= 0.9,by=0.1)){  HW\_Model = HoltWinters(sales.ts.dev, alpha=a, beta=b, gamma=c)  hold\_predict = forecast(HW\_Model,5)  MAPE[j] = sum(abs(hold\_predict$mean[1:5]-sales.ts.hold[1:5])/sales.ts.hold[1:5])/length(sales.ts.hold)  a1[j] = a  b1[j] = b  c1[j] =c  j = j+1  }  }  print(a)  }  temp = data.frame(MAPE = MAPE, alpha = a1, beta = b1, gamma = c1)  subset(temp, MAPE==min(MAPE)) #Holt Winter's Forecast sales.HW= HoltWinters(sales.ts, alpha=0.2, beta=0.9, gamma=0.5)  sales.fs.HW= forecast(sales.HW, 8)  plot(sales.fs.HW)  acc.HW=accuracy(sales.fs.HW,sales.ts1)  acc.HW #Arima Model sales.aa=auto.arima(sales.ts)  sales.fs.aa = forecast(sales.aa, h=8,model=Arima)  acc.aa=accuracy(sales.fs.aa,sales.ts1)  plot(sales.fs.aa)  acc.aa #TbATS Model sales.tbats = tbats(sales.ts)  sales.fs.tbats = forecast(sales.tbats, h=8)  acc.tb=accuracy(sales.fs.tbats,sales.ts1)  plot(sales.fs.tbats)  acc.tb #NNETS Model sales.nn = nnetar(sales.ts)  sales.fs.nn = forecast(sales.nn, h=8)  acc.nn=accuracy(sales.fs.nn,sales.ts1)  plot(sales.fs.nn)  acc.nn  2nd Part  require(readxl) # Excel-File reader  require(dplyr) # Data Manupulation  require(forecast) # Time Series  require(tidyr) # splits a col into two based on the regular expression  require(car) # to set un-ordered factors ranking method, helmert - baseline is one method rest referring to it  require(dummies) ## to reacte dummy variables for factor data  require(ggplot2)  require(usdm) # Building the Rawdata frame - complete data excel\_sheets('dunnhumby - Breakfast at the Frat.xlsx')  rawdata = read\_excel(path = 'dunnhumby - Breakfast at the Frat.xlsx', sheet = 'dh Transaction Data')  rawStoreData = read\_excel(path = 'dunnhumby - Breakfast at the Frat.xlsx', sheet = 'dh Store Lookup')  rawdata = rawdata %>%  left\_join(rawStoreData[which(!duplicated(rawStoreData$STORE\_ID)),],  by = c('STORE\_NUM'='STORE\_ID')) %>%  left\_join(read\_excel(path = 'dunnhumby - Breakfast at the Frat.xlsx', sheet = 'dh Products Lookup'),  by = 'UPC')  rawdata$WEEK\_END\_DATE = as.Date(rawdata$WEEK\_END\_DATE) # Forecast of the next 12 Weeks overall Spends for the company # building data as per weekly total spend happened  company\_data = rawdata %>%  dplyr::select(WEEK\_END\_DATE,SPEND,FEATURE,DISPLAY,TPR\_ONLY,  BASE\_PRICE,PRICE) %>%  group\_by(WEEK\_END\_DATE) %>%  mutate(W\_TOTAL\_SPEND = sum(SPEND)) %>%  dplyr::select(WEEK\_END\_DATE,W\_TOTAL\_SPEND) %>%  unique() %>%  arrange(WEEK\_END\_DATE) # reading spend data as TimeSeries TS\_data = ts(company\_data$W\_TOTAL\_SPEND, start = c(2009,1), end = c(2011,52), frequency = 52)  TS\_data\_train = ts(company\_data$W\_TOTAL\_SPEND[1:144], start = c(2009,1), end = c(2011,40), frequency = 52)  TS\_data\_test = ts(company\_data$W\_TOTAL\_SPEND[145:156], start = c(2011,41), end = c(2011,52), frequency = 52)  plot(TS\_data\_train)  plot(TS\_data\_test,type='b')  # arima mmodel builing and tuning  ARIMA\_Train=auto.arima(TS\_data\_train)  Predicted\_SPEND = forecast::forecast(ARIMA\_Train, h=12 )  Acc=accuracy(TS\_data\_test,Predicted\_SPEND$mean)  Acc  plot(Predicted\_SPEND) # model implementation ARIMA\_Full = arima(TS\_data, order=c(2,1,1))  Forecast\_12W = forecast::forecast(ARIMA\_Full, h=12 )  Forecast\_12W$mean  plot(Forecast\_12W$mean,type='b', xlab='TIME',ylab='SPEND',main='Forecast for 12 Weeks overall Spend')  rm(list = c('company\_data','TS\_data','TS\_data\_train','TS\_data\_test','ARIMA\_Train','Predicted\_SPEND','ARIMA\_Full')) # Checking for missing values - 3 fields have missing data Missing\_data\_Check <- function(data\_set){  NA\_Count = sapply(data\_set,function(y) sum(length(which(is.na(y)))))  Null\_Count = sapply(data\_set,function(y) sum(length(which(is.null(y)))))  Length0\_Count = sapply(data\_set,function(y) sum(length(which(length(y)==0))))  Empty\_Count = sapply(data\_set,function(y) {if(class(y) != 'Date')  return(sum(length(which(y==''))))  else return(0)})  Total\_NonData = NA\_Count+Null\_Count+Length0\_Count+Empty\_Count  return( Total\_NonData )  }  Missing\_data\_Check(rawdata)  # Missing data handling ---- # Fixing 1.1 BasePrice - replace with the weekly product average spend non\_baseprice\_missing\_rawdata = rawdata[,1:25][!apply(rawdata[,9], 1, function(x) any(is.na(x))),]  product\_WeeklyAvg\_basePrice = non\_baseprice\_missing\_rawdata %>%  dplyr::select(WEEK\_END\_DATE,UPC,BASE\_PRICE)%>%  group\_by(UPC,WEEK\_END\_DATE) %>%  mutate(AVG\_BASE\_PRICE= round(mean(BASE\_PRICE),2)) %>%  dplyr::select(WEEK\_END\_DATE,UPC,AVG\_BASE\_PRICE) %>%  unique()  baseprice\_missing\_rawdata = rawdata[,1:25][apply(rawdata[,9], 1, function(x) any(is.na(x))),]  baseprice\_missing\_rawdata = baseprice\_missing\_rawdata %>%  left\_join(product\_WeeklyAvg\_basePrice,  by=c("WEEK\_END\_DATE","UPC"))  baseprice\_missing\_rawdata$BASE\_PRICE = baseprice\_missing\_rawdata$AVG\_BASE\_PRICE  rawdata\_BasePrice = rbind(non\_baseprice\_missing\_rawdata,baseprice\_missing\_rawdata[-26])  rm(list = c('non\_baseprice\_missing\_rawdata','product\_WeeklyAvg\_basePrice','baseprice\_missing\_rawdata')) # Fixing 1.2 Price: replace by the weekly product average spend non\_price\_missing\_rawdata = rawdata\_BasePrice[,1:25][!apply(rawdata\_BasePrice[,8], 1, function(x) any(is.na(x))),]  product\_WeeklyAvg\_Price = non\_price\_missing\_rawdata %>%  dplyr::select(WEEK\_END\_DATE,UPC,PRICE)%>%  group\_by(UPC,WEEK\_END\_DATE) %>%  mutate(AVG\_PRICE= round(mean(PRICE),2)) %>%  dplyr::select(WEEK\_END\_DATE,UPC,AVG\_PRICE) %>%  unique()  price\_missing\_rawdata = rawdata\_BasePrice[,1:25][apply(rawdata\_BasePrice[,8], 1, function(x) any(is.na(x))),]  price\_missing\_rawdata = price\_missing\_rawdata %>%  left\_join(product\_WeeklyAvg\_Price,  by=c("WEEK\_END\_DATE","UPC"))  price\_missing\_rawdata$PRICE = price\_missing\_rawdata$AVG\_PRICE  rawdata\_BasePrice\_Price = rbind(non\_price\_missing\_rawdata,price\_missing\_rawdata[-26])  rm(list = c('non\_price\_missing\_rawdata','product\_WeeklyAvg\_Price','price\_missing\_rawdata','rawdata\_BasePrice')) # Creating Discount based features & Modifying Volume units rawdata\_BasePrice\_Price$DISCOUNT\_PRICE = rawdata\_BasePrice\_Price$BASE\_PRICE - rawdata\_BasePrice\_Price$PRICE  rawdata\_BasePrice\_Price$DISCOUNT\_PERCENT = (rawdata\_BasePrice\_Price$DISCOUNT\_PRICE/rawdata\_BasePrice\_Price$BASE\_PRICE)\*100  rawdata\_BasePrice\_Price$DISCOUNT = ifelse(rawdata\_BasePrice\_Price$DISCOUNT\_PRICE!=0,1,0) # re-defining features as factors rawdata\_factors = as.data.frame(unclass(rawdata\_BasePrice\_Price))  rawdata\_factors$FEATURE = as.factor(rawdata\_factors$FEATURE)  rawdata\_factors$DISPLAY = as.factor(rawdata\_factors$DISPLAY)  rawdata\_factors$TPR\_ONLY = as.factor(rawdata\_factors$TPR\_ONLY)  rawdata\_factors$DISCOUNT = as.factor(rawdata\_factors$DISCOUNT)  rawdata\_factors$MSA\_CODE = as.factor(rawdata\_factors$MSA\_CODE)  rm(rawdata\_BasePrice\_Price) # re-calculating the product size to milli scale unique(rawdata\_factors$PRODUCT\_SIZE)  # 1 oz = 29.5735 ml # 1 LT = 1000 ml # 1 ML = 1 ml # 1 CT = 200 mg  rawdata\_factors = extract(rawdata\_factors, PRODUCT\_SIZE, c("VOLUME", "V\_UNITS"), "([^ ]+) (.\*)")  rawdata\_factors$VOLUME=as.numeric(rawdata\_factors$VOLUME)  rawdata\_factors$PRODUct\_SIZE\_Milli = ifelse(rawdata\_factors$V\_UNITS=='ML', rawdata\_factors$VOLUME,  ifelse(rawdata\_factors$V\_UNITS=='CT',rawdata\_factors$VOLUME\*200,  ifelse(rawdata\_factors$V\_UNITS=='LT',rawdata\_factors$VOLUME\*1000,  ifelse(rawdata\_factors$V\_UNITS=='OZ',rawdata\_factors$VOLUME\*29.5735,-1))))  rawdata\_factors = rawdata\_factors[,-c(25,26)]  write.csv(rawdata\_factors,'rawdata\_factors.csv',row.names = F) # Store Wise Clustering # Building a dataframe =, with store details and avg sales details  storedata2 = rawdata\_factors %>%  group\_by(STORE\_NUM) %>%  dplyr::select(STORE\_NUM,STORE\_NAME,ADDRESS\_CITY\_NAME,  ADDRESS\_STATE\_PROV\_CODE,MSA\_CODE,SEG\_VALUE\_NAME,  PARKING\_SPACE\_QTY,SALES\_AREA\_SIZE\_NUM,AVG\_WEEKLY\_BASKETS) %>%  unique()  storedata1 = rawdata\_factors %>%  group\_by(STORE\_NUM,WEEK\_END\_DATE) %>%  mutate(T\_WEEKLY\_SPEND1 = sum(SPEND),  T\_WEEKLY\_UNITS\_SOLD1 = sum(UNITS),  T\_WEEKLY\_VISITS1 = sum(VISITS),  T\_WEEKLY\_HHS1 = sum(HHS)) %>%  dplyr::select(STORE\_NUM,T\_WEEKLY\_SPEND1,T\_WEEKLY\_UNITS\_SOLD1,  T\_WEEKLY\_VISITS1,T\_WEEKLY\_HHS1,WEEK\_END\_DATE) %>%  unique() %>%  group\_by(STORE\_NUM) %>%  mutate(AVG\_WEEKLY\_T\_SPEND = mean(T\_WEEKLY\_SPEND1),  AVG\_WEEKLY\_T\_UNITS\_SOLD = mean(T\_WEEKLY\_UNITS\_SOLD1),  AVG\_WEEKLY\_T\_VISITS = mean(T\_WEEKLY\_VISITS1),  AVG\_WEEKLY\_T\_HHS = mean(T\_WEEKLY\_HHS1)) %>%  dplyr::select(STORE\_NUM,AVG\_WEEKLY\_T\_SPEND,AVG\_WEEKLY\_T\_UNITS\_SOLD,  AVG\_WEEKLY\_T\_VISITS,AVG\_WEEKLY\_T\_HHS) %>%  unique()  storedata = left\_join(storedata1,storedata2,by='STORE\_NUM')  rm(list = c('storedata1','storedata2'))  storedata$PARKING\_FLAG = as.factor(ifelse(is.na(storedata$PARKING\_SPACE\_QTY),0,1))  storedata$PARKING\_NEW = ifelse(is.na(storedata$PARKING\_SPACE\_QTY),0,storedata$PARKING\_SPACE\_QTY) # Optimal centers for Clustering k\_max = 10  C\_data = storedata[,c(15,12,13,4)] #PARKING\_NEW,SALES\_AREA\_SIZE\_NUM,AVG\_WEEKLY\_BASKETS,AVG\_WEEKLY\_T\_VISITS  set.seed(1234)  wss = sapply(1:k\_max,  function(k){kmeans(C\_data, k, nstart=10,iter.max = 10 )$tot.withinss})  wss  plot(1:k\_max, wss,  type="b", pch = 19, frame = FALSE,  xlab="Number of clusters K",  ylab="Total within-clusters sum of squares",  main='Elbow method using wss') # Clustering set.seed(1234)  storedata$C3=kmeans(storedata[,c(15,12,13,4)],centers = 3)$cluster  rm(C\_data)  # Cluster's dataframe creating  Clusterdata = storedata %>%  group\_by(C3) %>%  mutate(AVG\_WEEKLY\_C\_T\_SPEND = mean(AVG\_WEEKLY\_T\_SPEND),  AVG\_WEEKLY\_C\_T\_UNITS\_SOLD = mean(AVG\_WEEKLY\_T\_UNITS\_SOLD),  AVG\_WEEKLY\_C\_T\_VISITS = mean(AVG\_WEEKLY\_T\_VISITS),  AVG\_WEEKLY\_C\_T\_HHS = mean(AVG\_WEEKLY\_T\_HHS),  AVG\_WEEKLY\_C\_BASKETS = mean(AVG\_WEEKLY\_BASKETS),  SALES\_AREA\_C\_SIZE\_NUM = mean(SALES\_AREA\_SIZE\_NUM),  NUM\_STORE\_WITH\_PARKING = sum(as.numeric(PARKING\_FLAG)),  NUM\_STORES = n()) %>%  dplyr::select(C3,AVG\_WEEKLY\_C\_T\_SPEND,NUM\_STORE\_WITH\_PARKING,  NUM\_STORES,AVG\_WEEKLY\_C\_T\_HHS,AVG\_WEEKLY\_C\_BASKETS,  AVG\_WEEKLY\_C\_T\_UNITS\_SOLD,AVG\_WEEKLY\_C\_T\_VISITS,  SALES\_AREA\_C\_SIZE\_NUM) %>%  unique() # Cluster visualization ggplot(storedata)+  geom\_point(aes(x= SALES\_AREA\_SIZE\_NUM,  y=AVG\_WEEKLY\_BASKETS,  color=as.factor(C3),  shape = as.factor(PARKING\_FLAG),  size = AVG\_WEEKLY\_T\_VISITS)) +  labs(title = "Store Cluster", x = "Store Area", y = "Average Weekly Baskets",  color = "Cluster", shape = "Parking Availability", size= "Average Weekly Visits") # Promotion Significance Product wise require(sjPlot)  require(broom)  Promo\_Sig = data.frame(  productCode = numeric(0),  Intercept\_PValue = numeric(0),  FEATURE\_PValue = numeric(0),  DISPLAY\_PValue = numeric(0),  TPR\_ONLY\_PValue = numeric(0),  F\_Length = numeric(0),  D\_Length = numeric(0),  T\_Length = numeric(0)  )  Promo\_Sig\_Names =names(Promo\_Sig)  Product\_list=unique(rawdata\_factors$UPC)  for(i in Product\_list){  print(i)  data1=subset(rawdata\_factors,UPC==i)  data1$FEATURE=as.numeric(data1$FEATURE)  data1$DISPLAY=as.numeric(data1$DISPLAY)  data1$TPR\_ONLY=as.numeric(data1$TPR\_ONLY)  model=lm(SPEND~FEATURE+DISPLAY+TPR\_ONLY,data = data1)  print(summary(model))  out=tidy(model)  temp = coef(summary(model))[, "Pr(>|t|)"]  Promo\_Sig = rbind(Promo\_Sig,data.frame(i,temp["(Intercept)"],  temp["FEATURE"],temp["DISPLAY"],  temp["TPR\_ONLY"],length(unique(data1$FEATURE)),  length(unique(data1$DISPLAY)),length(unique(data1$TPR\_ONLY))))  }  colnames(Promo\_Sig) = Promo\_Sig\_Names  Promo\_Sig$F\_Sig = ifelse(Promo\_Sig$FEATURE\_PValue<=0.05,'F','')  Promo\_Sig$D\_Sig = ifelse(Promo\_Sig$DISPLAY\_PValue<=0.05,'D','')  Promo\_Sig$T\_Sig = ifelse(Promo\_Sig$TPR\_ONLY\_PValue<=0.05,'T','')  Promo\_Sig$Sig\_modes = (Promo\_Sig$F\_Sig=='F')+(Promo\_Sig$D\_Sig=='D')+(Promo\_Sig$T\_Sig=='T') # Productwise Spend forecast for next 12 Weeks Product\_forecast = data.frame(numeric())  Product\_accr\_AA = data.frame(numeric())  Product\_accr\_TB = data.frame(numeric())  Product\_accr\_NN = data.frame(numeric())  P\_TOP\_LIST = c('1600027527','3800031838','7192100339','1600027528','1600027564','3800031829','3800039118','7192100337','1111085350','1111009477')  for(Product in P\_TOP\_LIST){  print(Product)  P1= subset(rawdata\_factors, UPC== Product)  P1$FEATURE=as.numeric(P1$FEATURE)  P1$DISPLAY=as.numeric(P1$DISPLAY)  P1$TPR\_ONLY=as.numeric(P1$TPR\_ONLY)  P1\_data = P1 %>%  dplyr::select(WEEK\_END\_DATE,SPEND,FEATURE,DISPLAY,TPR\_ONLY,  BASE\_PRICE,PRICE) %>%  group\_by(WEEK\_END\_DATE) %>%  mutate(W\_TOTAL\_SPEND = sum(SPEND),  W\_FEATURE = ifelse(sum(FEATURE)==0,0,1),  W\_DISPLAY = ifelse(sum(DISPLAY)==0,0,1),  W\_TPR\_ONLY = ifelse(sum(FEATURE)==0,0,1),  W\_BASE\_PRICE = mean(BASE\_PRICE),  W\_PRICE = mean(PRICE)) %>%  dplyr::select(WEEK\_END\_DATE,W\_TOTAL\_SPEND,W\_FEATURE,  W\_DISPLAY,W\_TPR\_ONLY,W\_BASE\_PRICE,W\_PRICE) %>%  unique()%>%  mutate(W\_PROMO = ifelse(W\_FEATURE+W\_DISPLAY+W\_TPR\_ONLY == 0,0,1),  W\_DISCOUNT = W\_BASE\_PRICE-W\_PRICE )%>%  arrange(WEEK\_END\_DATE)  print(length(P1\_data$W\_TOTAL\_SPEND))  TS\_data = ts(P1\_data$W\_TOTAL\_SPEND, start = c(2009,1), end = c(2011,52), frequency = 52)  TS\_data\_train = ts(P1\_data$W\_TOTAL\_SPEND[1:117], start = c(2009,1), end = c(2011,17), frequency = 52)  TS\_data\_test = ts(P1\_data$W\_TOTAL\_SPEND[118:156], start = c(2011,18), end = c(2011,52), frequency = 52)  ARIMA\_Train=auto.arima(TS\_data\_train)  Predicted\_SPEND = forecast::forecast(ARIMA\_Train, h=38)  Acc=accuracy(TS\_data\_test,Predicted\_SPEND$mean)  Product\_accr\_AA = rbind(Acc,Product\_accr\_AA) #TbATS Model sales.tbats = tbats(TS\_data\_train)  sales.fs.tbats = forecast(sales.tbats, h=38)  acc.tb=accuracy(sales.fs.tbats,TS\_data\_test)  Product\_accr\_TB = rbind(Product\_accr\_TB,acc.tb) #NNETS Model sales.nn = nnetar(TS\_data\_train)  sales.fs.nn = forecast(sales.nn, h=38)  acc.nn=accuracy(sales.fs.nn,TS\_data\_test)  Product\_accr\_NN= rbind(Product\_accr\_NN,acc.nn)  ARIMA\_Full = auto.arima(TS\_data)  Forecast\_12W = forecast::forecast(ARIMA\_Full, h=12 )  Forecast\_12W$mean  #print(Forecast\_12W$mean)  #Product\_forecast\_AA = Forecast\_12W$mean  Product\_forecast\_AA = rbind(Product,Forecast\_12W$mean,Product\_forecast\_AA)  name = paste('AA',Product,'\_Forecast\_SPEND\_12W.png', sep = '')  png(filename =paste0('plots/',name), width = 800, height = 600, units = 'px')  plot(Forecast\_12W$mean,type='b', xlab='TIME',ylab='SPEND',main= Product)  dev.off()  nn\_Full = nnetar(TS\_data)  Forecast\_12W\_nn = forecast::forecast(nn\_Full, h=12 )  Forecast\_12W\_nn$mean  #print(Forecast\_12W$mean)  #Product\_forecast\_NN = Forecast\_12W\_nn$mean  Product\_forecast\_NN= rbind(Product,Product\_forecast\_NN,Forecast\_12W\_nn$mean)  name = paste('NN',Product,'\_Forecast\_SPEND\_12W.png', sep = '')  png(filename =paste0('plots/',name), width = 800, height = 600, units = 'px')  plot(Forecast\_12W$mean,type='b', xlab='TIME',ylab='SPEND',main= Product)  dev.off()  }  #remove(Product\_forecast\_NN)  #remove(Product\_forecast\_AA)  #rownames(Product\_forecast) = c('1600027527','3800031838','7192100339','1600027528','1600027564','3800031829','3800039118','7192100337','1111085350','88491201426','1111009477')  #colnames(Product\_forecast) = c('2012W2','2012W3','2012W4','2012W5','2012W6','2012W7','2012W8','2012W9','2012W10','2012W11','2012W12','2012W13') # Top Products Promotion Effectiveness P\_TOP\_LIST = c('1600027527',  '3800031838',  '7192100339',  '1600027528',  '1600027564',  '3800031829',  '3800039118',  '7192100337',  '1111085350',  #'88491201426',  '1111009477'  )  data = read.csv("work1.csv")  data$WEEK\_END\_DATE=as.Date(data$WEEK\_END\_DATE,format="%d-%b-%y")  for(i in P\_TOP\_LIST){  P1= subset(data, UPC== i)  P1$FEATURE=as.numeric(P1$FEATURE)  P1$DISPLAY=as.numeric(P1$DISPLAY)  P1$TPR\_ONLY=as.numeric(P1$TPR\_ONLY)  P1\_data = P1 %>%  dplyr::select(WEEK\_END\_DATE,SPEND,FEATURE,DISPLAY,TPR\_ONLY,  BASE\_PRICE,PRICE) %>%  group\_by(WEEK\_END\_DATE) %>%  mutate(W\_TOTAL\_SPEND = sum(SPEND),  W\_FEATURE = ifelse(sum(FEATURE)==0,0,1),  W\_DISPLAY = ifelse(sum(DISPLAY)==0,0,1),  W\_TPR\_ONLY = ifelse(sum(FEATURE)==0,0,1),  W\_BASE\_PRICE = mean(BASE\_PRICE),  W\_PRICE = mean(PRICE)) %>%  dplyr::select(WEEK\_END\_DATE,W\_TOTAL\_SPEND,W\_FEATURE,  W\_DISPLAY,W\_TPR\_ONLY,W\_BASE\_PRICE,W\_PRICE) %>%  unique()%>%  mutate(W\_PROMO = ifelse(W\_FEATURE+W\_DISPLAY+W\_TPR\_ONLY == 0,0,1),  W\_DISCOUNT = W\_BASE\_PRICE-W\_PRICE )%>%  arrange(WEEK\_END\_DATE)    TS\_data = ts(P1\_data$W\_TOTAL\_SPEND, start = c(2009,1), end = c(2011,52), frequency = 52)  Decomposition = decompose(TS\_data)  P1\_data$W\_Seasonal = Decomposition$seasonal  print(i)  #print(vif(data.frame(P1\_data[,c(3,4,5,6,10)])))  #print(vif(data.frame(P1\_data[,c(3,4,6,10)]))) # removing TPR\_ONLY due to collinearity  LR\_Model = lm(W\_TOTAL\_SPEND ~ W\_BASE\_PRICE+W\_FEATURE+W\_DISPLAY+W\_TPR\_ONLY+W\_Seasonal,  data = P1\_data)  summary(LR\_Model)  Coef = LR\_Model$coefficients    Base\_Line = Coef[1]+Coef[2]\*(P1\_data$W\_BASE\_PRICE)+(Coef[5]\*P1\_data$W\_Seasonal)  INC = Coef[3]\*(P1\_data$W\_FEATURE)+(Coef[4]\*P1\_data$W\_DISPLAY)    P1\_data$BASE\_LINE = Base\_Line  P1\_data$INCREMENTAL = INC    name = paste(i,'\_Promo\_Incr\_12W.png', sep = '')  png(filename =paste0('plots/',name), width = 800, height = 600, units = 'px')  print(ggplot(P1\_data, aes(x=WEEK\_END\_DATE ))+  geom\_line(aes(y=W\_TOTAL\_SPEND),color='black',show.legend = F)+  geom\_line(aes(y=BASE\_LINE),color='blue',show.legend = F)+  geom\_line(aes(y=INCREMENTAL),color='red',show.legend = F)+  ylab('SALES')  )  dev.off()  } # Top Products Sales Forecast P\_TOP\_LIST = c('1600027527','3800031838','3800039118')  i= 1  plot\_list = list()  for(Product in P\_TOP\_LIST){  print(Product)  P1= subset(rawdata\_factors, UPC== Product )  P1$FEATURE=as.numeric(P1$FEATURE)  P1$DISPLAY=as.numeric(P1$DISPLAY)  P1$TPR\_ONLY=as.numeric(P1$TPR\_ONLY)  P1\_data = P1 %>%  dplyr::select(WEEK\_END\_DATE,SPEND,FEATURE,DISPLAY,TPR\_ONLY,  BASE\_PRICE,PRICE) %>%  group\_by(WEEK\_END\_DATE) %>%  mutate(W\_TOTAL\_SPEND = sum(SPEND),  W\_FEATURE = ifelse(sum(FEATURE)==0,0,1),  W\_DISPLAY = ifelse(sum(DISPLAY)==0,0,1),  W\_TPR\_ONLY = ifelse(sum(FEATURE)==0,0,1),  W\_BASE\_PRICE = mean(BASE\_PRICE),  W\_PRICE = mean(PRICE)) %>%  dplyr::select(WEEK\_END\_DATE,W\_TOTAL\_SPEND,W\_FEATURE,  W\_DISPLAY,W\_TPR\_ONLY,W\_BASE\_PRICE,W\_PRICE) %>%  unique()%>%  mutate(W\_PROMO = ifelse(W\_FEATURE+W\_DISPLAY+W\_TPR\_ONLY == 0,0,1),  W\_DISCOUNT = W\_BASE\_PRICE-W\_PRICE )%>%  arrange(WEEK\_END\_DATE)  #print(length(P1\_data$W\_TOTAL\_SPEND))  TS\_data = ts(P1\_data$W\_TOTAL\_SPEND, start = c(2009,1), end = c(2011,52), frequency = 52)  TS\_data\_train = ts(P1\_data$W\_TOTAL\_SPEND[1:117], start = c(2009,1), end = c(2011,17), frequency = 52)  TS\_data\_test = ts(P1\_data$W\_TOTAL\_SPEND[118:156], start = c(2011,18), end = c(2011,52), frequency = 52)  #NNETS Model  NN\_model = nnetar(TS\_data)  forecast\_12w = forecast(NN\_model, h=12)  print(forecast\_12w$mean)  y = c(2,3,4,5,6,7,8,9,10,11,12,13)  temp = data.frame(Spend = as.numeric(forecast\_12w$mean), Week\_No= y )  p=ggplot(temp,aes(Week\_No,Spend))+  geom\_line()+  geom\_point()+  ggtitle(Product)  plot\_list[[i]] = p  i= i+1  }  P\_TOP\_LIST = c(  '7192100339',  '1600027528','1600027564','3800031829',  '7192100337','1111085350',  '88491201426','1111009477')  i= 4  for(Product in P\_TOP\_LIST){  print(Product)  P1= subset(rawdata\_factors, UPC== Product )  P1$FEATURE=as.numeric(P1$FEATURE)  P1$DISPLAY=as.numeric(P1$DISPLAY)  P1$TPR\_ONLY=as.numeric(P1$TPR\_ONLY)  P1\_data = P1 %>%  dplyr::select(WEEK\_END\_DATE,SPEND,FEATURE,DISPLAY,TPR\_ONLY,  BASE\_PRICE,PRICE) %>%  group\_by(WEEK\_END\_DATE) %>%  mutate(W\_TOTAL\_SPEND = sum(SPEND),  W\_FEATURE = ifelse(sum(FEATURE)==0,0,1),  W\_DISPLAY = ifelse(sum(DISPLAY)==0,0,1),  W\_TPR\_ONLY = ifelse(sum(FEATURE)==0,0,1),  W\_BASE\_PRICE = mean(BASE\_PRICE),  W\_PRICE = mean(PRICE)) %>%  dplyr::select(WEEK\_END\_DATE,W\_TOTAL\_SPEND,W\_FEATURE,  W\_DISPLAY,W\_TPR\_ONLY,W\_BASE\_PRICE,W\_PRICE) %>%  unique()%>%  mutate(W\_PROMO = ifelse(W\_FEATURE+W\_DISPLAY+W\_TPR\_ONLY == 0,0,1),  W\_DISCOUNT = W\_BASE\_PRICE-W\_PRICE )%>%  arrange(WEEK\_END\_DATE)  #print(length(P1\_data$W\_TOTAL\_SPEND))  TS\_data = ts(P1\_data$W\_TOTAL\_SPEND, start = c(2009,1), end = c(2011,52), frequency = 52)  TS\_data\_train = ts(P1\_data$W\_TOTAL\_SPEND[1:117], start = c(2009,1), end = c(2011,17), frequency = 52)  TS\_data\_test = ts(P1\_data$W\_TOTAL\_SPEND[118:156], start = c(2011,18), end = c(2011,52), frequency = 52)  ARIMA\_Full = auto.arima(TS\_data)  Forecast\_12W = forecast::forecast(ARIMA\_Full, h=12 )  print(Forecast\_12W$mean)  y = c(2,3,4,5,6,7,8,9,10,11,12,13)  temp = data.frame(Spend = as.numeric(forecast\_12w$mean),Week\_No = y )  p=ggplot(temp,aes(Week\_No,Spend))+  geom\_line()+  geom\_point()+  ggtitle(Product)  plot\_list[[i]] = p  i= i+1    }  grid.arrange(plot\_list[[1]], plot\_list[[2]],plot\_list[[4]],plot\_list[[5]],  ncol = 2, nrow = 2)  grid.arrange( plot\_list[[6]], plot\_list[[7]], plot\_list[[3]],plot\_list[[8]],  ncol = 2, nrow = 2)  grid.arrange(plot\_list[[9]], plot\_list[[10]], plot\_list[[11]],  ncol = 1, nrow = 3) |

# References

1. Time Series and its applications by Robert Shumway
2. <http://www.cpgdatainsights.com/pricing-and-promotion/bww-part-1/>
3. <http://a-little-book-of-r-for-time-series.readthedocs.io/en/latest/src/timeseries.html>