**Name of Students**: Abhishek Dandona, Ruchika Bhatnagar, Neha Dhamnani

**Title of Report**: Retail Stores’ Sales Prediction

**Area of Project**: Supervised Learning involving Regression Problem

**Abstract**: The dataset contains Sales data of items sold by the retail outlets spread across different cities. Various features of items and outlets have been recorded in data. Based on the attributes of item and the outlet, it is required to predict the sales of data given it’s item\_id and oulet\_id. To makes Sales prediction various Regression approaches have been used. Submissions were made on Analytics Vidhya leaderboard and it stood at position six.

The Evaluation Metric used to evaluate the performance of model was Root Mean Square Error.

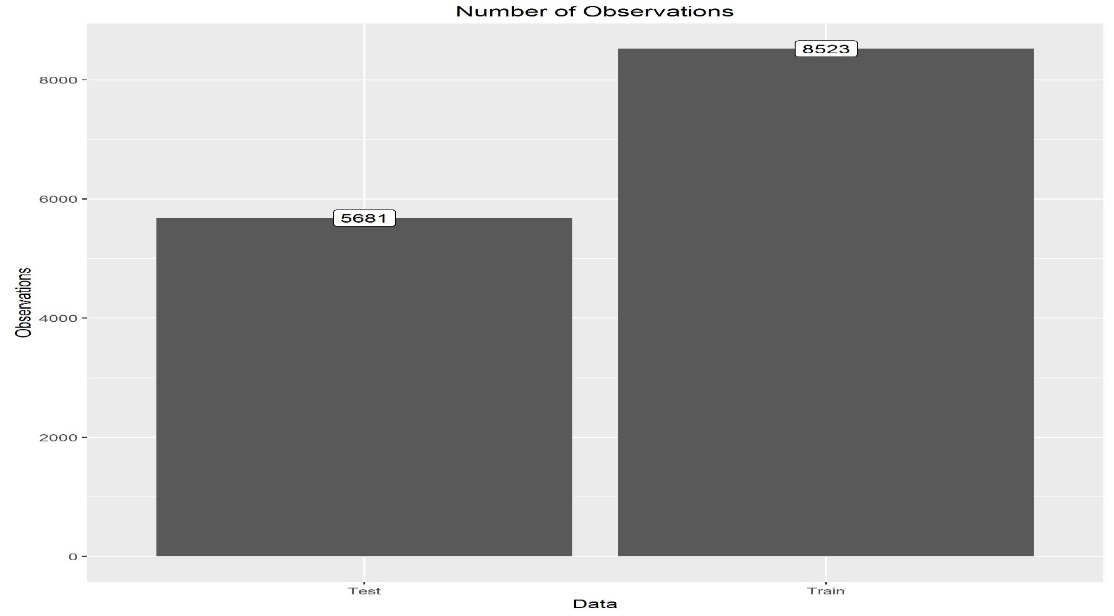
**Background of the:**

**work/prior-history**

**Description of the project work**

**Introduction:** The dataset for Project was taken from ‘Analytics Vidhya’. It has two parts, train set (850kB) to build models and test set(516kB) to check accuracy of model predictions. It has 11 independent variables and a response variable (continuous) to be predicted. The data comprises 2013 sales data for 1559 products across 10 stores in different cities. Also, certain attributes of each product and store have been defined. The aim is to build a predictive model and find out the sales of each product at a particular store.

Using this model, we will try to understand the properties of products and stores which play a key role in increasing sales.

****

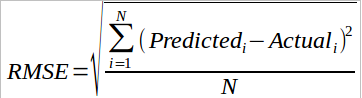
**Fig.1 Size of Data**

The data dictionary is tabulated below:

|  |  |
| --- | --- |
| Variable | Description |
| Item\_Identifier | Unique product ID |
| Item\_Weight | Weight of product. It had **2439** missing values. |
| Item\_Fat\_Content | Whether the product is low fat or not |
| Item\_Visibility | The % of total display area of all products in a store allocated to the particular product. It had **879** zero values. |
| Item\_Type | The category to which the product belongs |
| Item\_MRP | Maximum Retail Price (list price) of the product |
| Outlet\_Identifier | Unique store ID |
| Outlet\_Establishment\_Year | The year in which store was established |
| Outlet\_Size | The size of the store in terms of ground area covered. It had **4016** missing values. |
| Outlet\_Location\_Type | The type of city in which the store is located |
| Outlet\_Type | Whether the outlet is just a grocery store or some sort of supermarket |
| Item\_Outlet\_Sales | Sales of the product in the particular store. This is the outcome variable to be predicted.  **Table.1** |

Evaluation Metric:

Evaluation metric used to evaluate the performance for competition was Root Mean Square Error(RMSE).

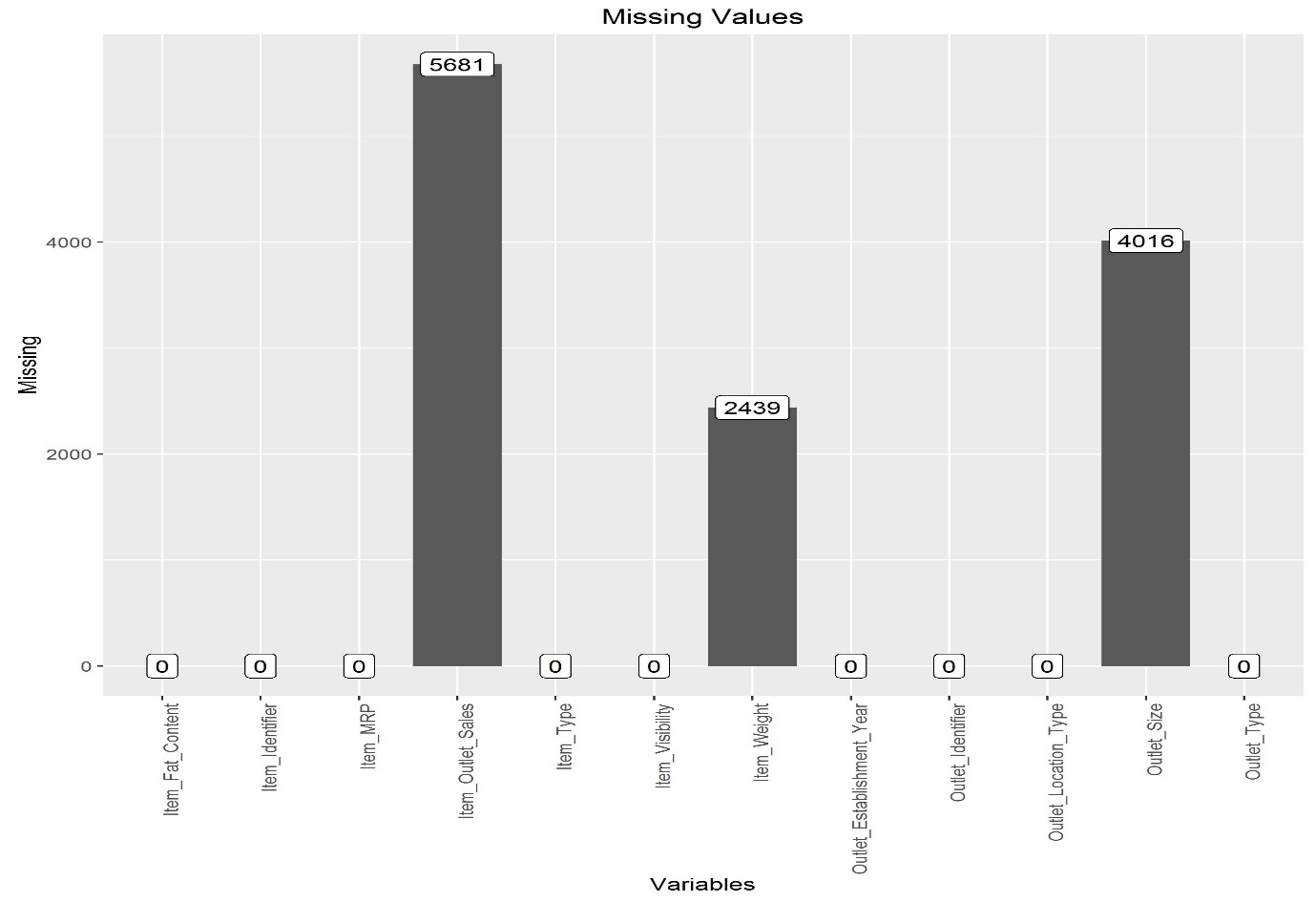


1. **Aim:** The aim of project is topredict sales generated (in rupees) by each item sold in a particular outlet. Sale is predicted based on various features of the item and the outlet where it is sold.
2. **Method used:** The project used Supervised Learning approaches to make predictions about sales, as historical data is available for the same. Regression Techniques have been used to predict and emphasis was laid on making accurate predictions rather than interpretability of model.

The basic flow of process included following steps:

1. Visualisation involved creating plots to visualise the data using ‘ggplot’ package in R.
2. Pre-Processing involved imputing missing values, replacing ‘0’ values in columns.
3. Feature Engineering involved creating new features based on judgement and decomposing categorical variables to dummy variables.
4. Feature Extraction involved extracting the relevant features from variable set based on their significance. This made use of Random Forest to extract features.
5. Modelling involved running various models on data and evaluating the relative performance of each model on the given data.
6. **Results and Analysis:**

Visualisation of data provided insights about data which were further used to pre-process the data. Visualisation and subsequent pre-processing steps have been detailed below:

****

**Fig.2 Count of Missing Values in combined data**

1. Item\_Weight imputation

There are 2439 missing weight data. Our approach to impute these missing values was to assume that the weight of a product is same across all outlets, locations, outlet types and outlet sizes.

From our observations, our assumption is true since the table shows the unique item identifiers having same weight across all outlets, locations and so on.



**Table.2**

Thus we imputed the missing weight data by the average weight grouped by item identifiers.

1. Outlet\_Size imputation

There are 4016 missing values in the Outlet size variable in the combined train and test data for Outlet 10, Outlet 17 and Outlet 45.

Outlet\_size has ordinal scale (small, medium, high) hence, we could choose the mode value to impute the missing values but, this would not be an appropriate approach as data is not missing at random (size is not missing across different outlets but for specific outlets only).

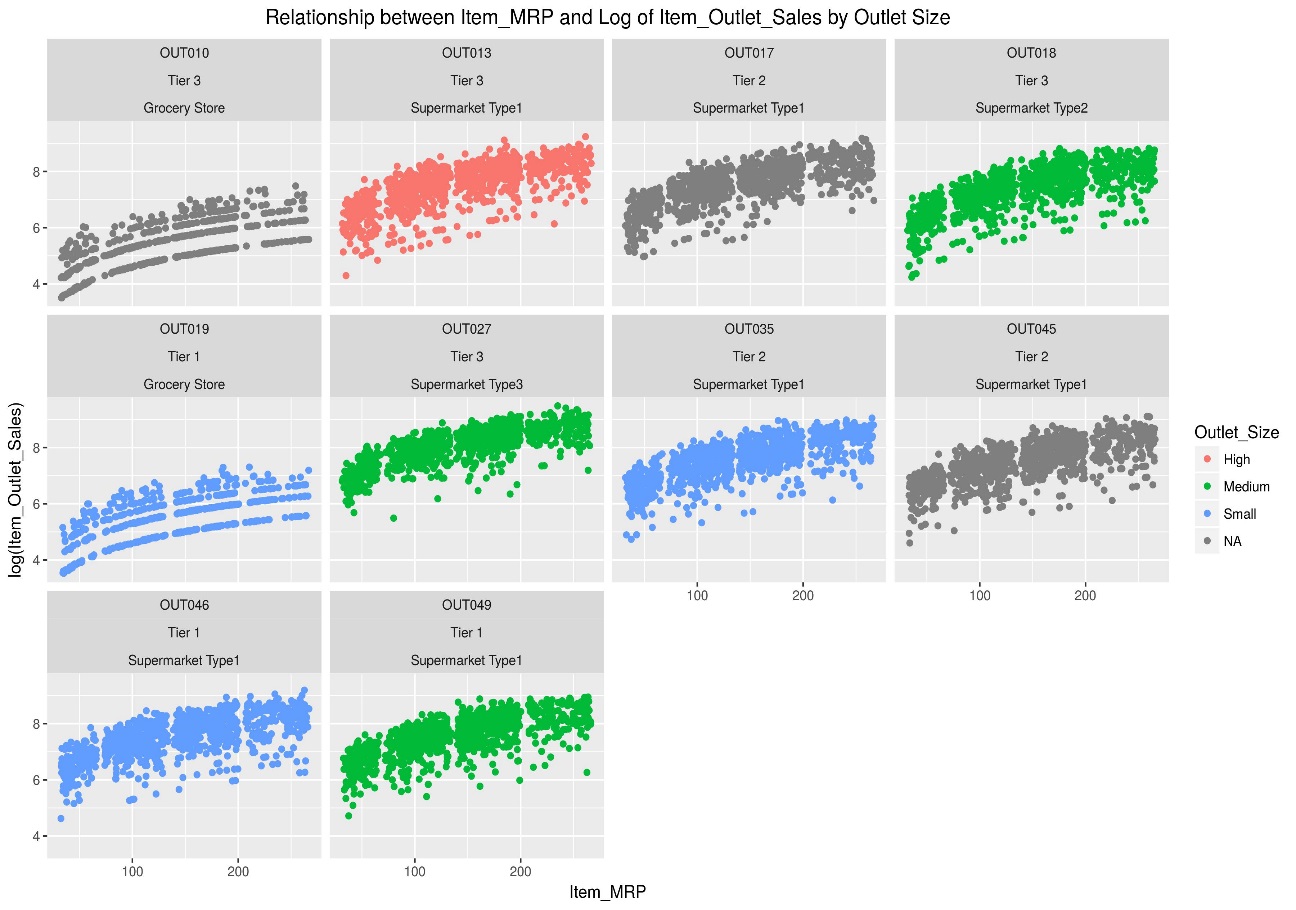
To impute these values, we observed the plot of Item MRP vs log of Item Outlet Sales by Outlet Identifier, Outlet Location Type and Outlet type.

Each outlet has a location type, outlet type and outlet size. We observed that the plot of outlet10 is very similar to the plot of outlet19. Outlet19 is the only grocery store among all the outlets and has a significantly different plot than other outlets, there is a significant association between Outlet10 and Outlet19. Out19 is small size outlet, we designated Out10 as having small size as well.

This assumption was proven correct when our score on the leader board improved marginally.

The dilemma was how to impute the missing values of Out17 and Out45 since their plots were very similar to each other as well as other outlets (Supermarket Type 1 and Type 2). Both these outlets are in tier 2 location and their type is supermarket type1, this meant that these outlets can be any of high, medium and small size.

We observed that outlets of high size are located only in tier 3 locations. Thus we were left with the choice of Small or Medium. We assigned Out45 as small and out17 as medium.



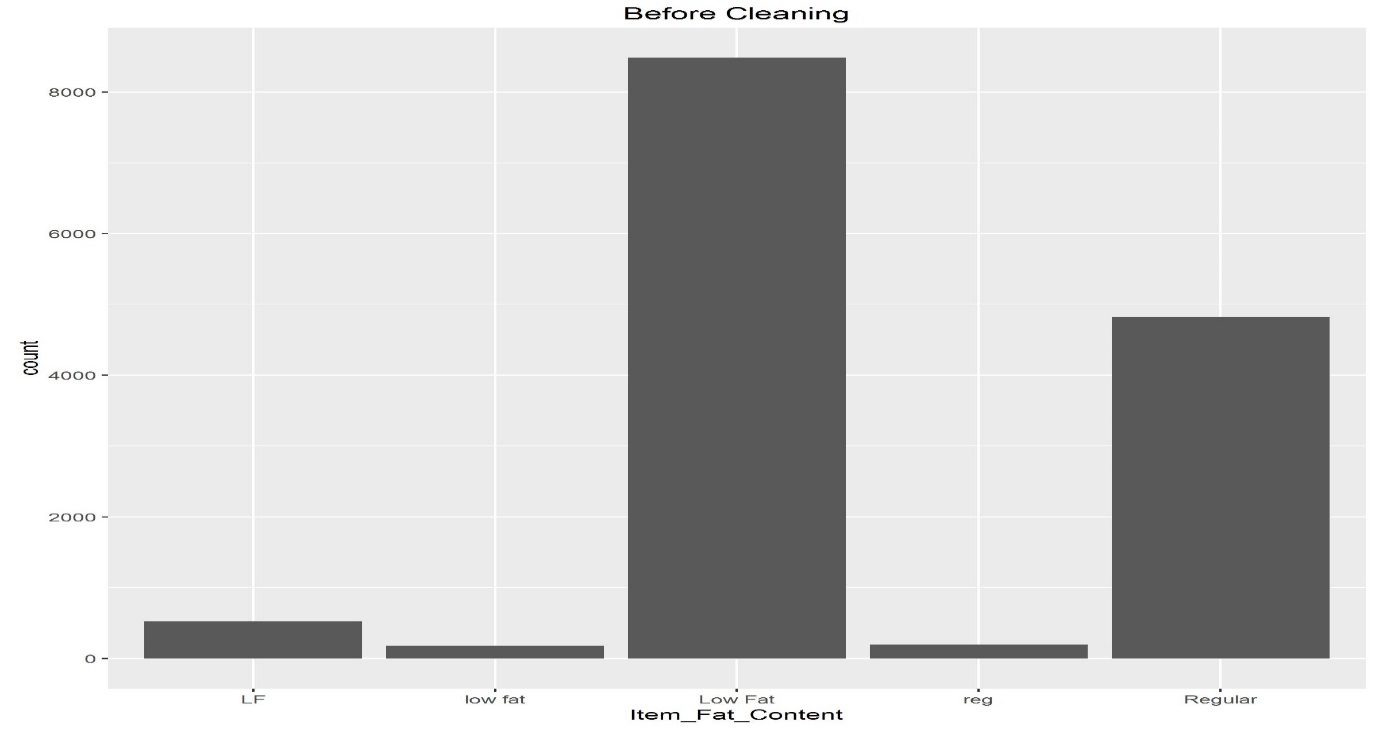
**Fig.3 Item\_Outlet\_Sales vs. Item\_MRP for outlets**

1. Item\_Visibility replacement

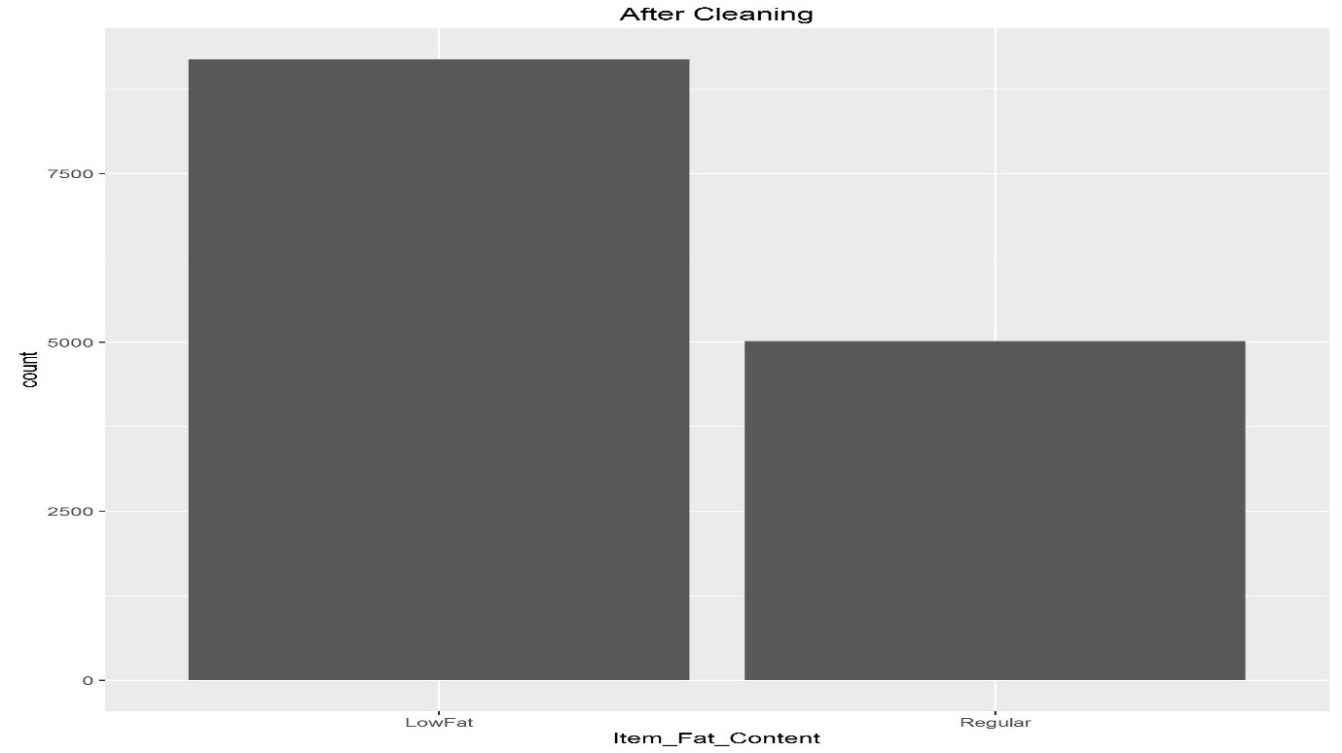
This column had ‘0’ values. These values were replaced with mean of Item\_Visibility grouped by Item\_Identifier.

1. Item\_Fat\_Content

It had five categories ‘LF’ ‘lowfat’ ‘LowFat’, ‘reg’ ‘regular’ which were reduced to two categories ‘LowFat’ and ‘Regular’ to make sense.



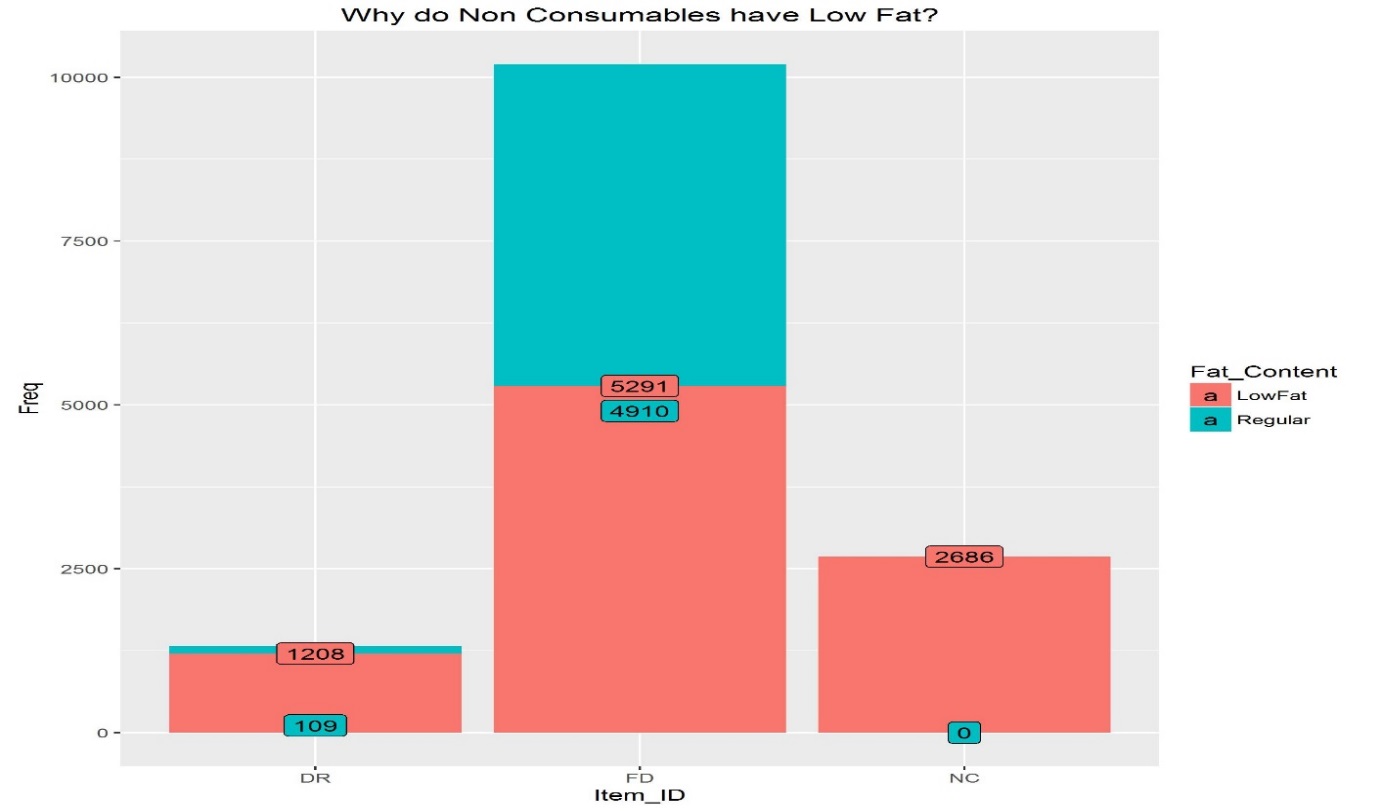
**Fig.4 Item\_Fat\_Content variable Before cleaning**

****

**Fig.5 Item\_Fat\_Content variable After cleaning**

1. Item\_Fat\_Content

It was observed that the items which had ‘NC’ in their Item\_Identifier were Non- Consumables and yet their Item\_Fat\_Content had ‘lowfat’, so, wherever Item\_Identifier had ‘NC’, Item\_Fat\_Content value was given as ‘nonedible’.

****

**Fig.6 NC (non consumables) having LowFat content??**

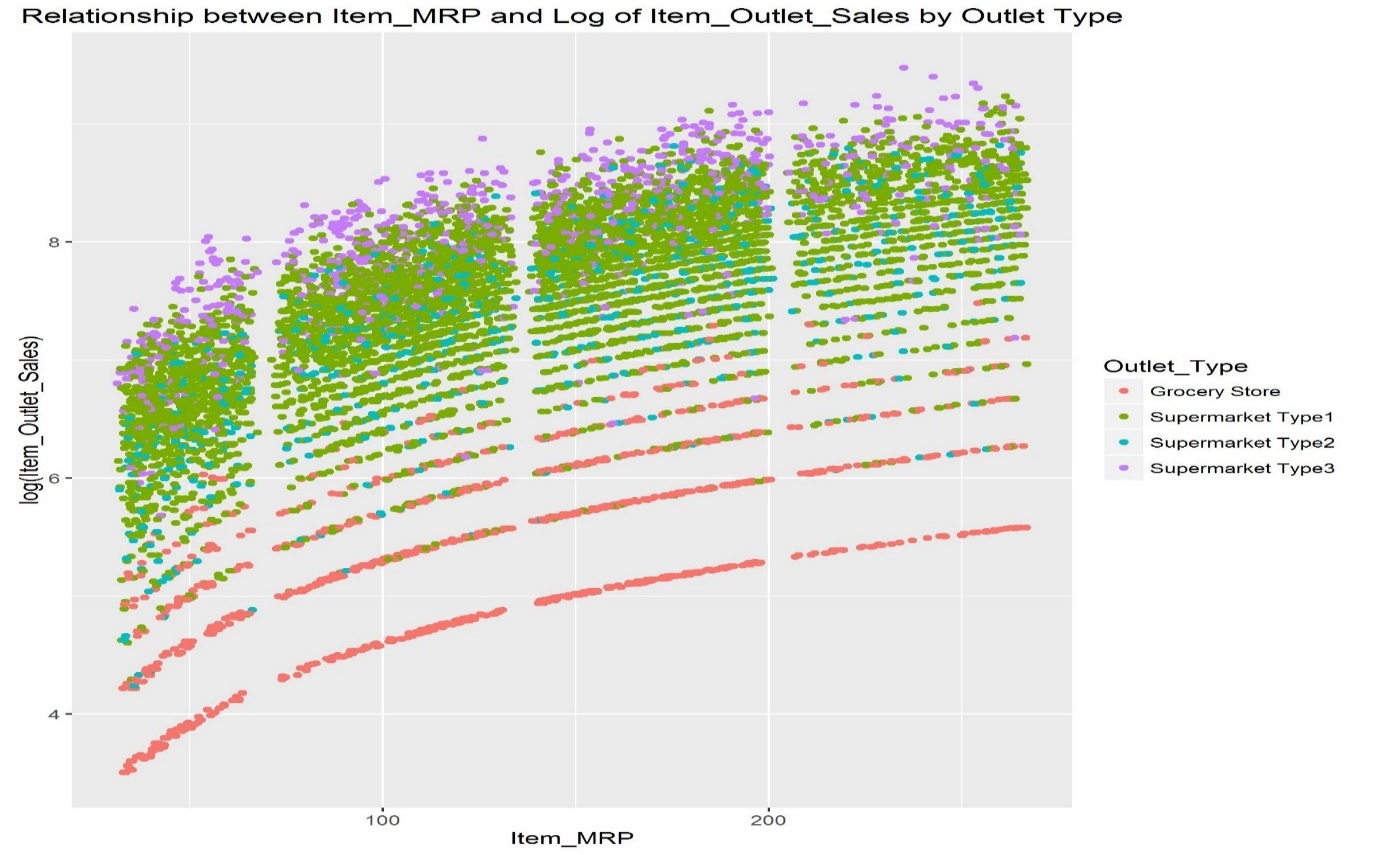
**Relationship of Response Variable with Independent Variables:**

1. Fig.7 illustrates that Medium sized outlets have higher sales than High and Small outlets for same priced products as shown by clustering of green dots on top.

****

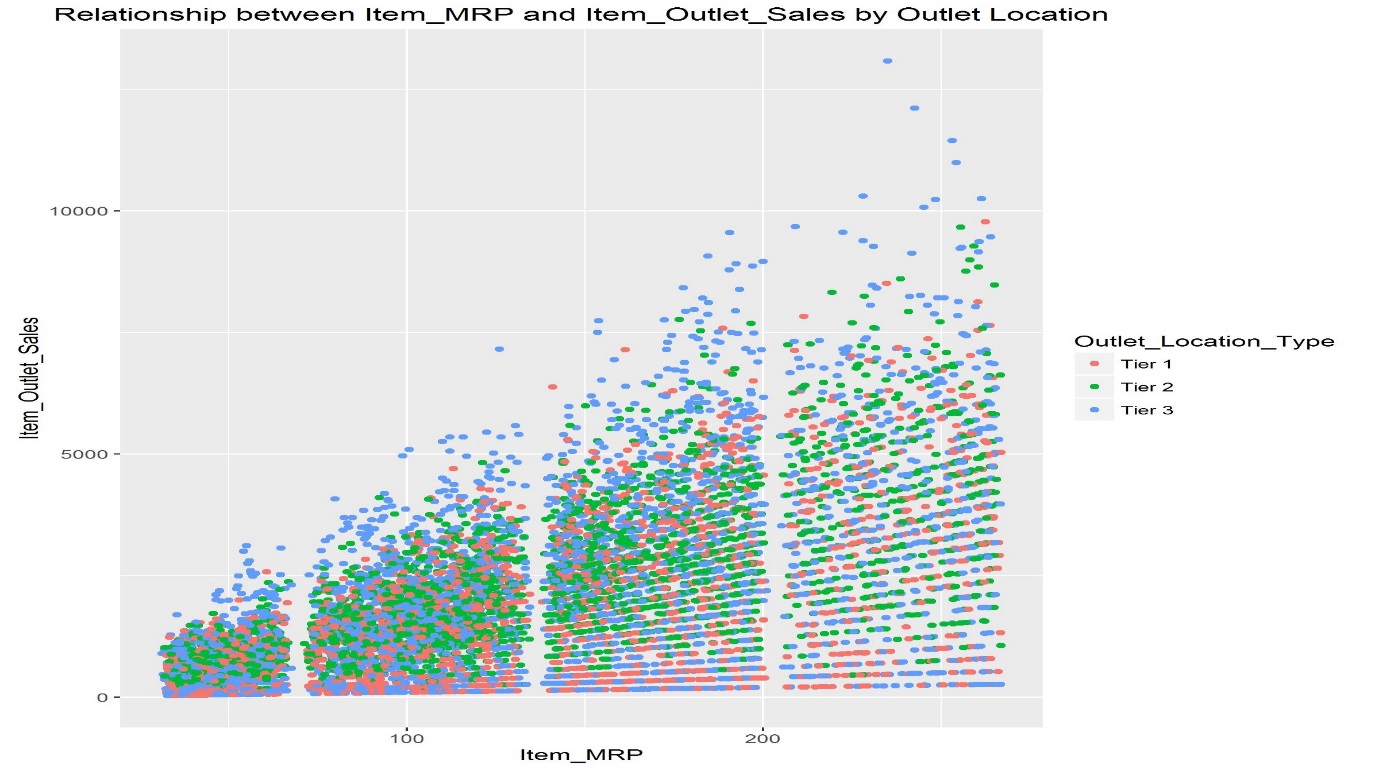
**Fig.7**

1. Fig.8 highlights the fact that Supermarket Type 3 has highest sales followed by Type 1 and Grocery stores have least. Also, there is heavy clustering of green points which shows that Supermarket Type 1 were more in our study than rest.

****

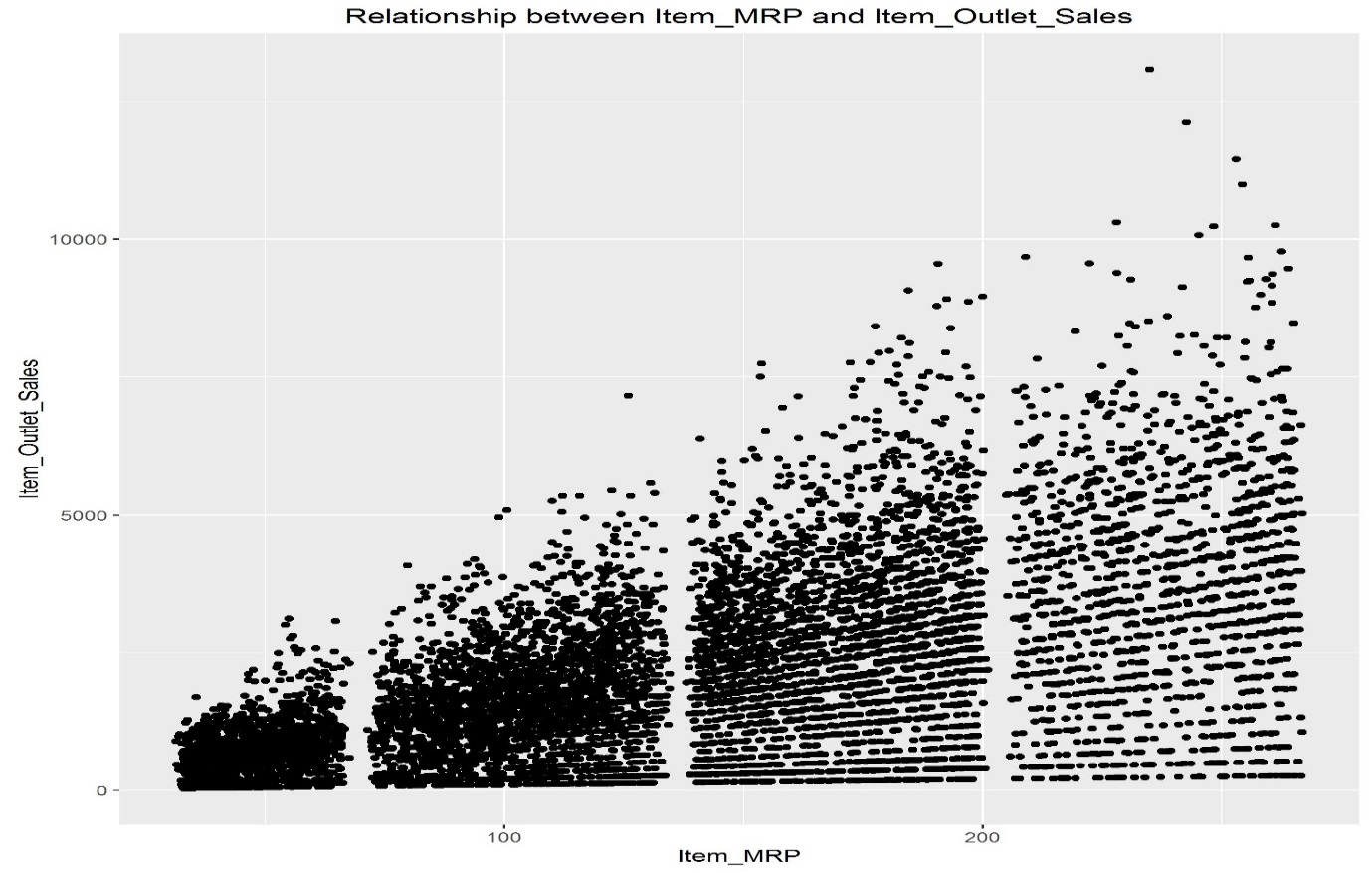
**Fig.8**

1. Fig.9 shows that Tier 3 Outlets occupy top position in terms of Sales, followed by outlets in Tier 2 locations.

****

**Fig.9**

1. Fig.10 shows the obvious fact that higher priced products contribute more towards sales and also that sales for lower MRP products has lower range whereas sales for high MRP products has wider range.

****

**Fig.10**

**Feature Engineering:**

Next step was Feature Engineering, that is new features were created. New Features created are as follows:

1. Item\_Identifier was broken down to categories that is a column was created for category FD(food), DR(drinks), NC (non consumables) under Item\_Identifier.
2. Item\_Identifier was broken down to sub-categories that is a column

(Item\_Id\_Subcat) was created for category (A to Z) under Item\_Identifier.

1. Item\_Identifier was further broken to sub-sub categories that is a column

(Item\_Id\_SubSubcat) was created for sub category (1 to 60) under Item\_Identifier.

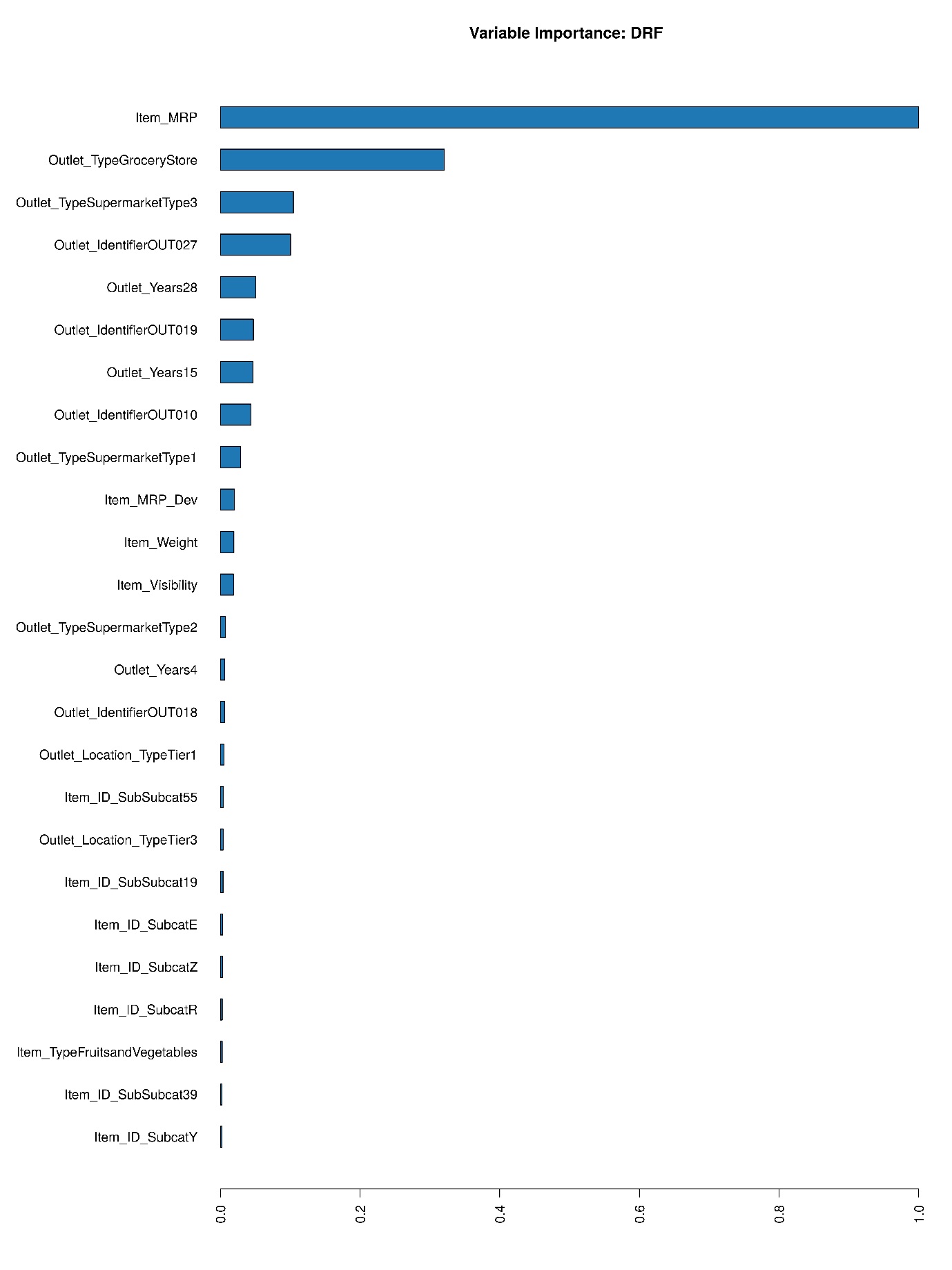
1. Dairy items were present both in FD(food) and DR(drinks), hence a new category of item was created as Dairy Food for dairy under food items.
2. Outlet\_Establishment\_Year was replaced with Outlet\_Years which showed age of outlet.
3. Each Item is priced different across different outlets. Therefore, a new feature was created that shows the difference between item MRP and its average MRP.
4. Last, all categorical variables were decomposed to dummy variables including the variables created above i.e. Item\_Type, Item\_Fat\_Content ,Outlet\_Identifier, Outlet\_Location\_Type, Outlet\_Type,Item\_Id, Item\_Id\_Subcat, Item\_Id\_SubSubcat, Outlet\_Years.

**Feature Extraction:**

After performing feature engineering we ended up with 152 variables.Due to high number of variables, we chose an algorithmic approach to select the significant features for our models.The method used was ‘Grid search for Random Forest model’.

The tuning parameters for the grid search were ntrees and max depth and we used h2o to perform the grid search. In total 16 Random Forest models were created and in very less time, thanks to h2o’s machine learning efficiency and parallel processing.

After performing grid search, we chose the model with the lowest RMSE metric and chose 25 most significant features. Fig.11 shows the importance of features.



**Fig.11 Variable Importance plot**

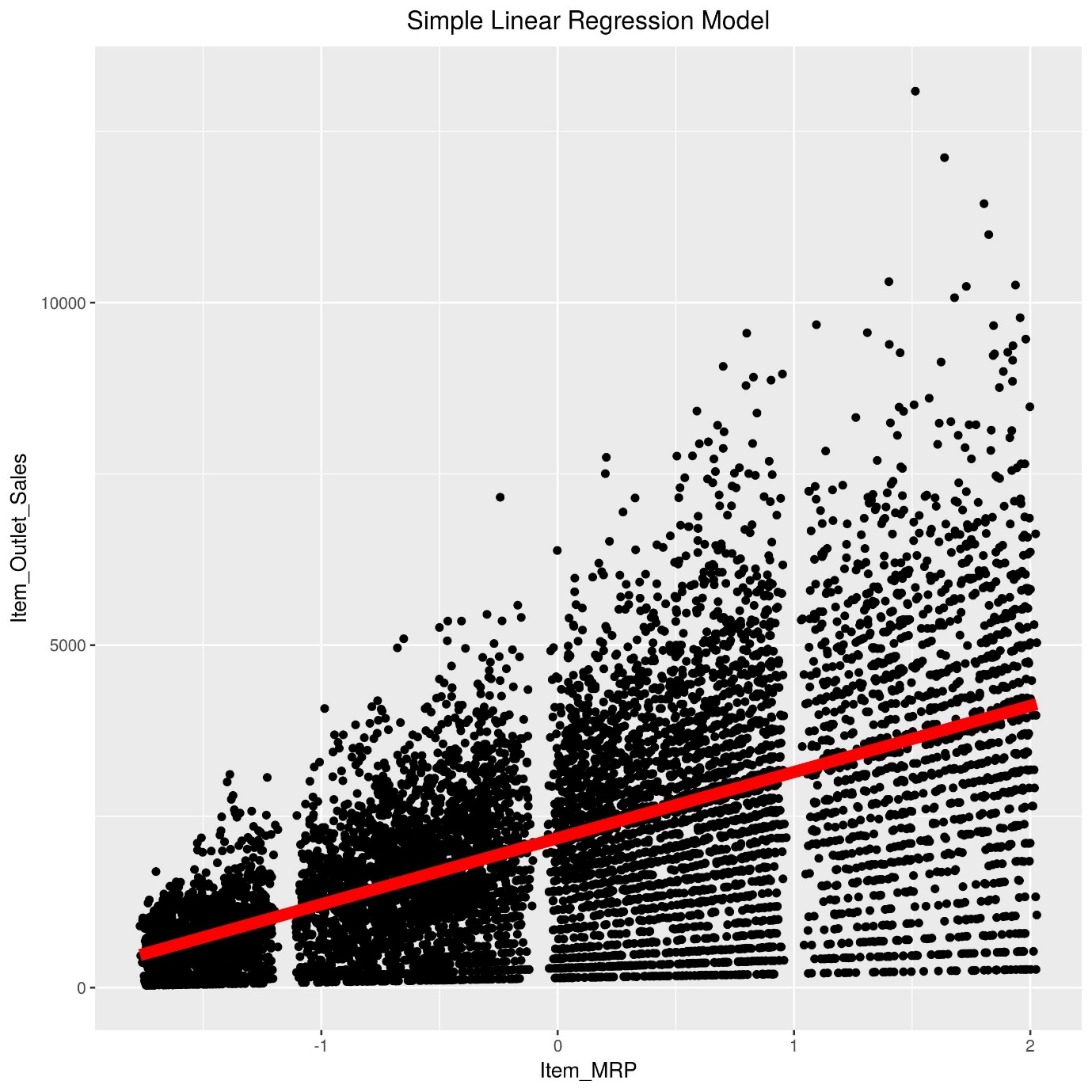
**Models Used:**

Baseline Model:

First, we created a baseline model by simply taking average Item outlet sales and reporting it as predicted sales. The **RMSE** on test data was **1773.825138**. The Models that we create ahead must have lower RMSE than the baseline model for the model to be significant.

Simple Linear Regression:

The simplest model made was one variable linear regression. The predictor was Item\_mrp since it has maximum correlation with the response variable. The performance of this model is significantly better than the baseline model. The reported **RMSE** of this model is **1483.214**



**Fig.12 Item\_Outlet\_Sales actual and predicted (red line)**

Multiple Linear Regression:

The next model built was a multiple linear regression model with 6 predictors. The predictors were Item Mrp, Outlet Type: Grocery Store, Outlet type: Supermarket Type3, Outlet type: Supermarket type 1, Item Id SubSub Cat 55, Item Id Subcat Y.

Multiple R2 :0.5629

Adjusted R2 : 0.5626

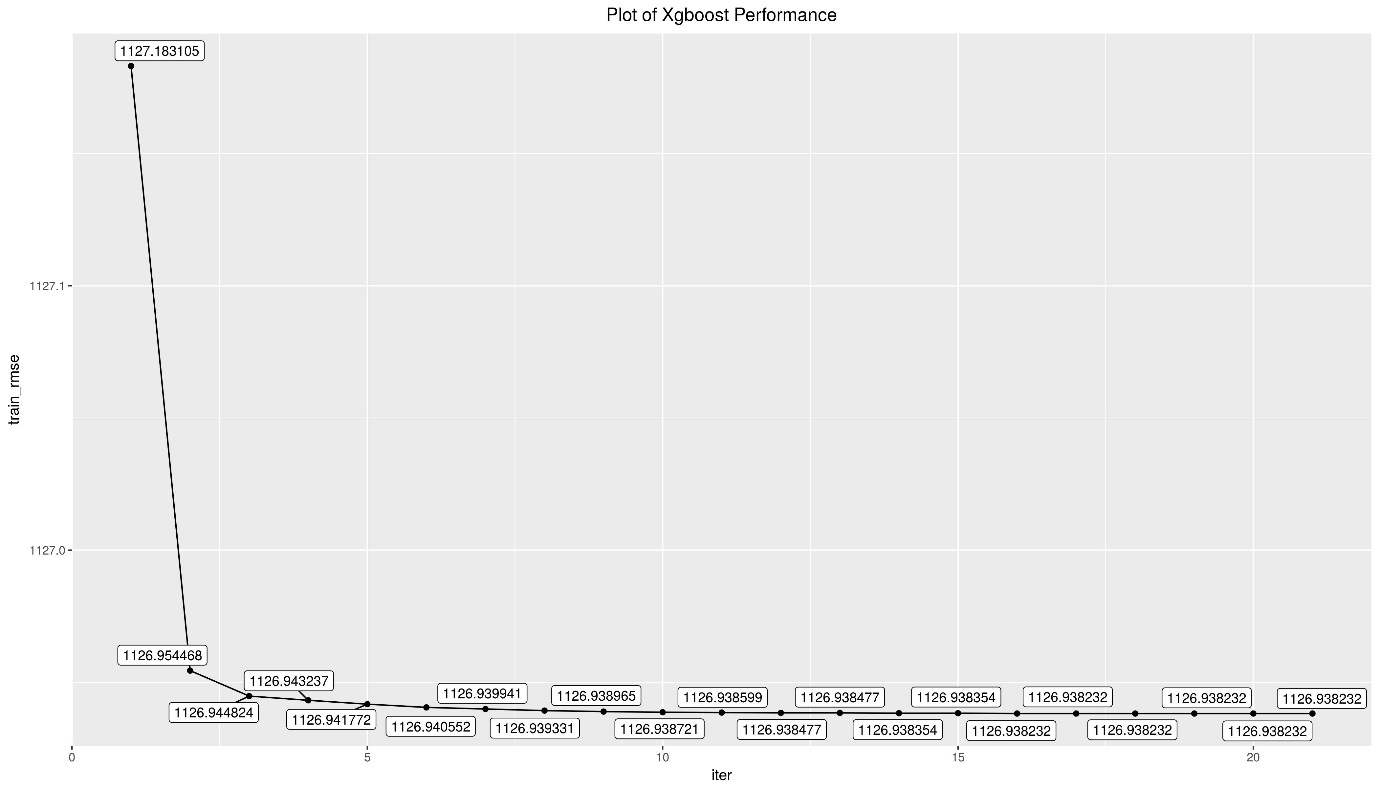
F statistic: 1828 on 6 and 8516 DF

The reported **RMSE** of this model on the test data is **1191.534**. The 10-fold cross validation of this model did not produce any improvement [Follow with explanation of the coefficients and interpretation]

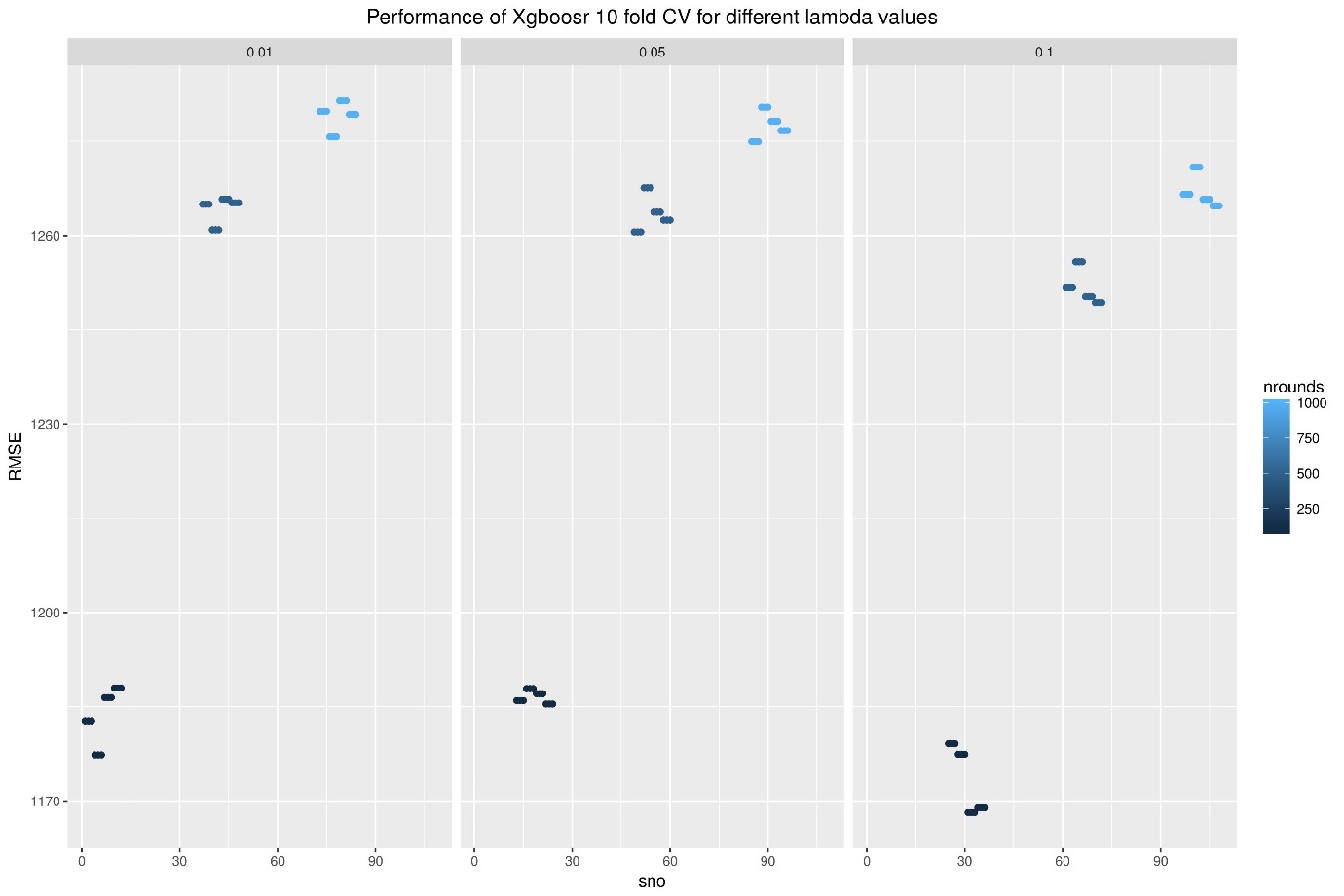
Xgboost:

Extreme Gradient Boosting is an efficient implementation of the gradient boosting frame. The package includes efficient linear model solver and tree learning algorithms. The package can automatically do parallel computation on a single machine which could be more than 10 times faster than existing gradient boosting packages. It supports various objective functions, including regression, classification and ranking. The package is made to be extensible, so that users are also allowed to define their own objectives easily.

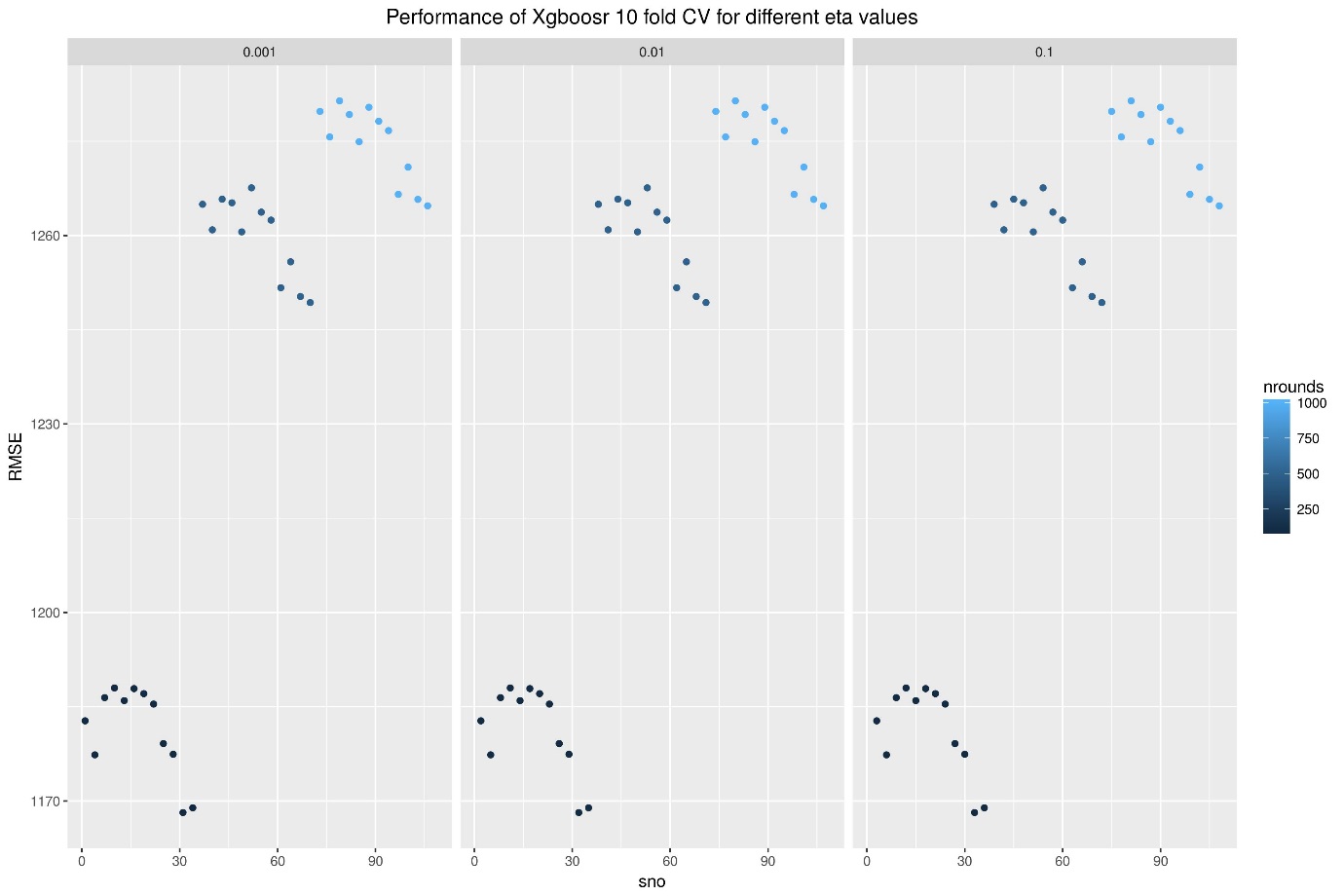
We shifted to black box models to better predict sales for a lower RMSE and sacrificing interpretability at the same time. Since xgboost has an internal feature selection mechanism, we fed in all the 25 predictors. Initially, we chose random parameters and the **RMSE** reported was **1188.582**. However, the cross validation score of xgboost was not satisfactory, it was 1218.671. Perhaps, an extensive grid search for optimal parameters of xgboost will improve the score significantly.



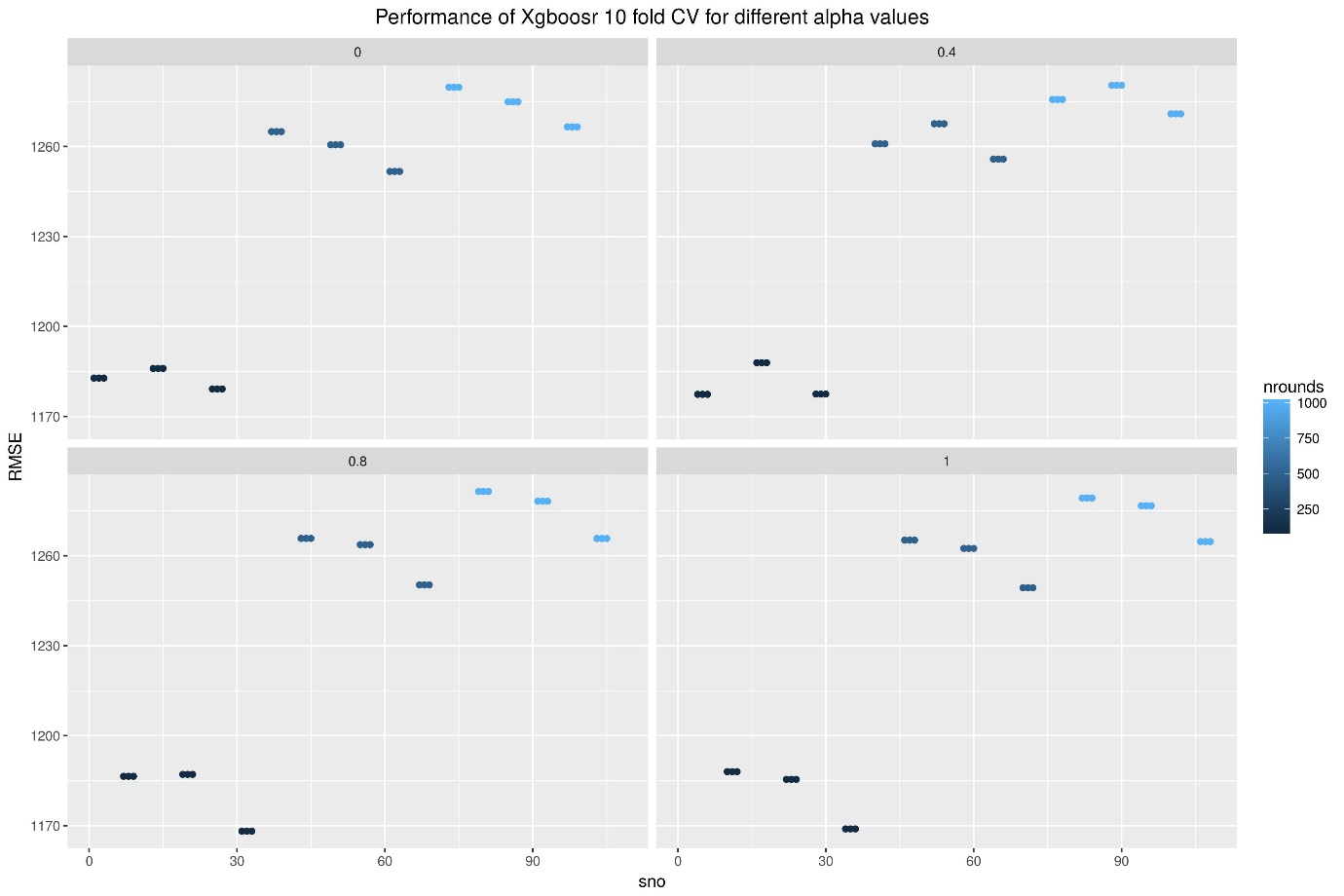
**Fig.13 Train\_rmse vs iterations for xgboost**

****

**Fig.14 10-fold RMSE for different lambda values**

****

**Fig.15 10-fold RMSE for different eta values**

****

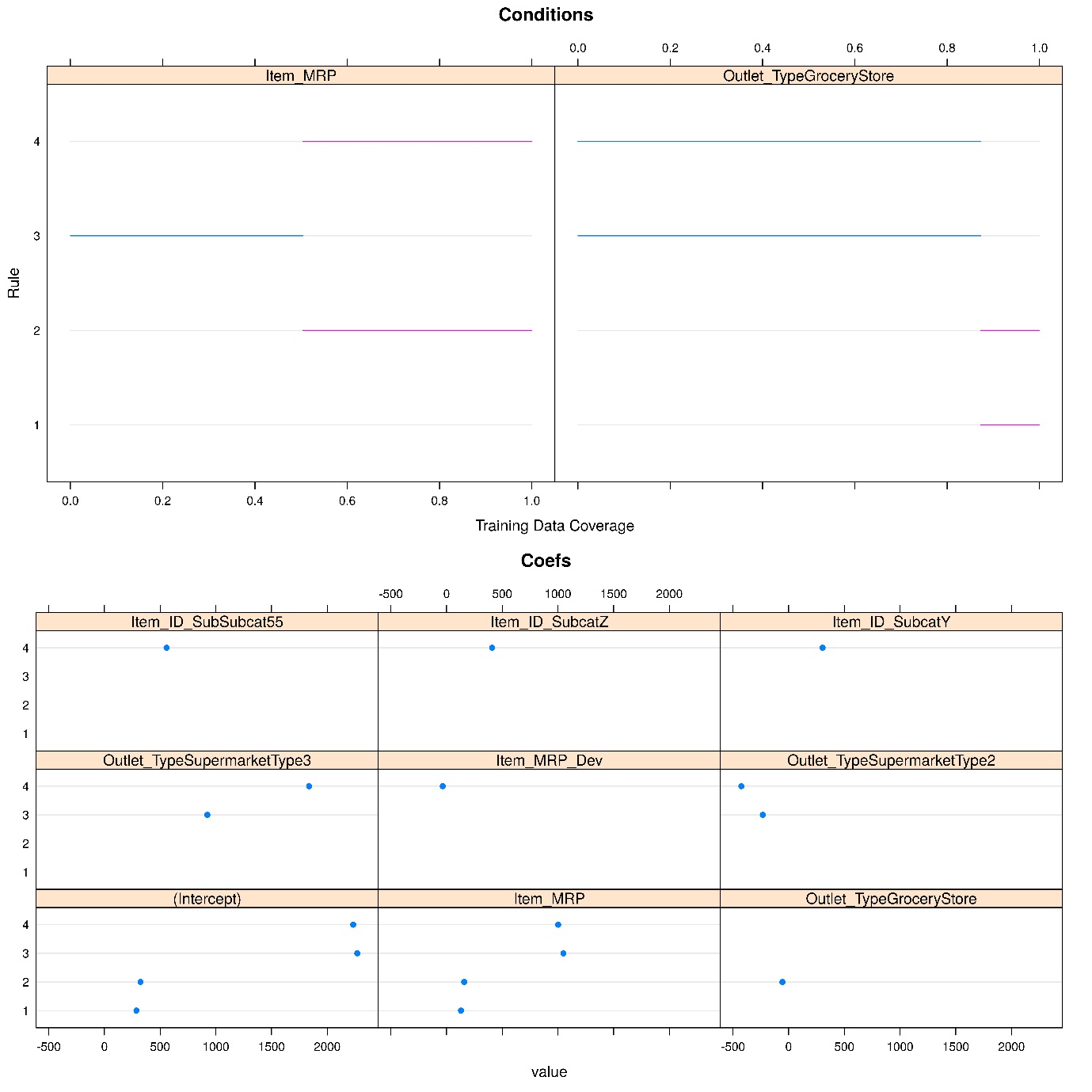
**Fig.16 10-fold RMSE for different alpha values**

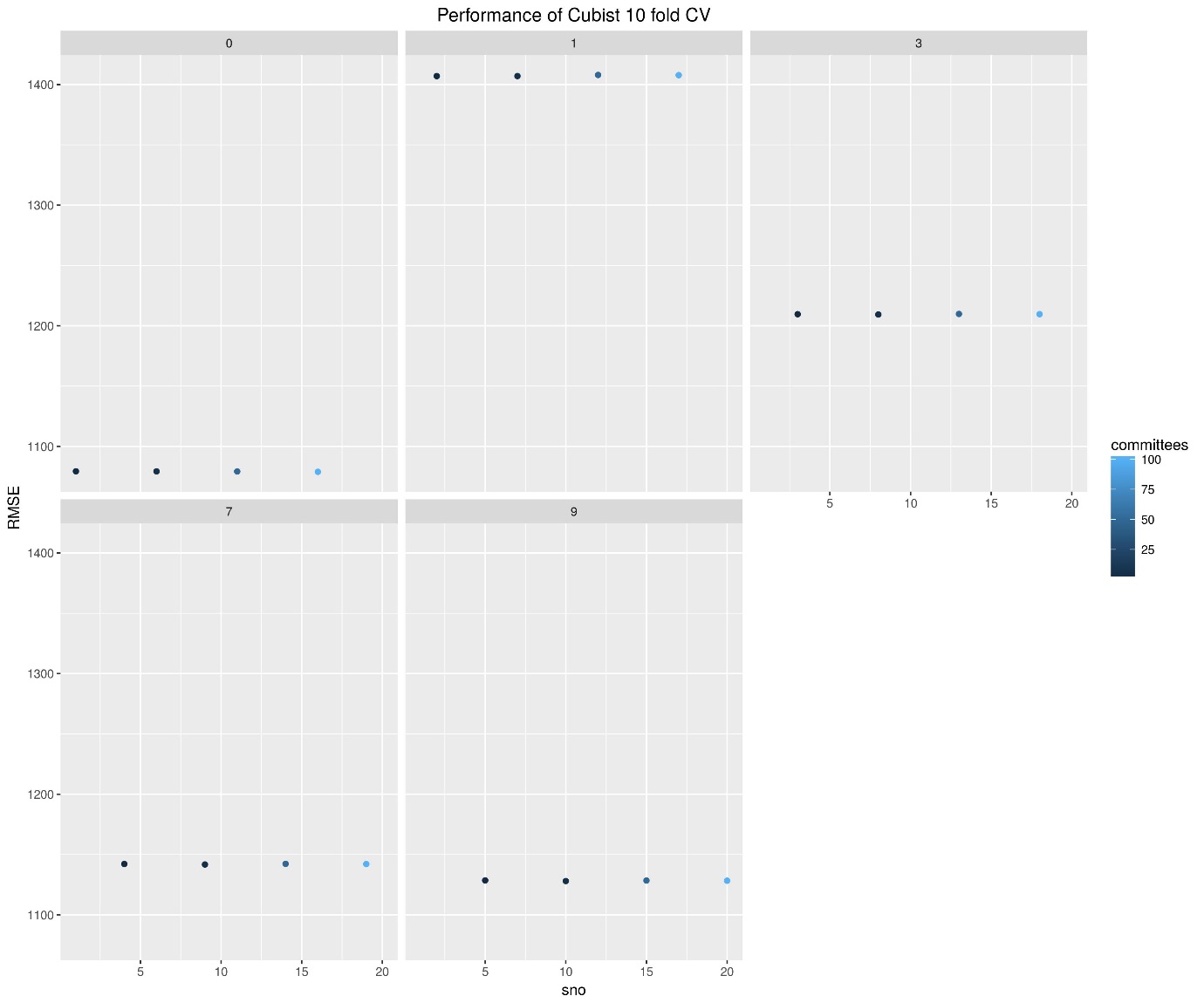
Cubist:

Cubist is a rule-based model that is an extension of Quinlan's M5 model tree. A tree is grown where the terminal leaves contain linear regression models. These models are based on the predictors used in previous splits. Also, there are intermediate linear models at each step of the tree. A prediction is made using the linear regression model at the terminal node of the tree, but is smoothed" by taking into account the prediction from the linear model in the previous node of the tree (which also occurs recursively up the tree). The tree is reduced to a set of rules, which initially are paths from the top of the tree to the bottom. Rules are eliminated via pruning and/or combined for

simplification.

This model was a surprise as its performance was significantly better while providing interpretability at the same time. Another useful feature of this model is that it trains fast. The **RMSE** reported by this model without cross validation was **1153.3581** and after cross validation the RMSE reported was **1152.750**

****

****

Bayesian Regularised Feed Forward Neural Network:

The brnn\_extended function fits a two-layer neural network as described in MacKay (1992) and Foresee and Hagan (1997). It uses the Nguyen and Widrow algorithm (1990) to assign initial weights and the Gauss-Newton algorithm to perform the optimization. The hidden layer contains two groups of neurons that allow us to assign different prior distributions for two groups of input variables.

Perhaps, this model is a complete black box but the performance provided by this model significantly improved our stand on the leaderboard. After using this model, our rank jumped from 30 to 6 on the leaderboard. The RMSE reported by using this model is **1137.294**, currently **this is the best result we got after all model executions**. However, the cross validation of this model is limited to the number of neurons only, thus the RMSE reported after cross validation was 1147.463.

Note: - Xgboost, Cubist and BRNN have one thing is common, it is that they have internal feature selection and except for cubist, the models have internal regularization as well.

1. **Conclusion:** This project tried five different models namely Simple Linear Regression, Multiple Linear Regression, BRNN (Bayesian Regularised Neural Networks), CUBIST and XgBoost, of which BRNN gave maximum accuracy and a score of 1137.9 on Analytics Vidhya leader board.
2. **Further Work:** Perhaps by manually changing the hyper parameters of BRNN, a lower RMSE can be achieved.

Similarly, with better parameter tuning of xgboost we may achieve a good rank.

More time could be spent on feature engineering to create new features such as volume of sales.

Since we made an assumption to impute outlet size, we could try other combinations to observe model performances.

1. **References:** Following references were used to complete the project.

* [**https://datahack.analyticsvidhya.com/contest/practice-problem-big-mart-sales-iii/**](https://datahack.analyticsvidhya.com/contest/practice-problem-big-mart-sales-iii/)
* [**https://www.r-bloggers.com/cross-validation-for-predictive-analytics-using-r/**](https://www.r-bloggers.com/cross-validation-for-predictive-analytics-using-r/)
* [**http://www.h2o.ai/h2o/machine-learning/**](http://www.h2o.ai/h2o/machine-learning/)
* [**http://tutorials.iq.harvard.edu/R/Rgraphics/Rgraphics.html**](http://tutorials.iq.harvard.edu/R/Rgraphics/Rgraphics.html)
* [**http://topepo.github.io/caret/train-models-by-tag.html#Linear\_Regression**](http://topepo.github.io/caret/train-models-by-tag.html#Linear_Regression)
* [**https://datahack.analyticsvidhya.com/contest/practice-problem-big-mart-sales-iii/lb**](https://datahack.analyticsvidhya.com/contest/practice-problem-big-mart-sales-iii/lb)

**Special Achievements :** The project achieved a rank 6 on Analytics Vidhya leader board, screenshot of which is shown here.

**Appendix:** R codefor the work is attached along.