**BIGMART CASE STUDY REPORT**

**Group No.**: Group 25

**Student Names:** Mayur Bhat & Adit Mehta

**Background and Introduction**

Background:

Data scientists at a retail chain have collected sales data for 1559 products across 10 stores in different cities across 12 variables. The aim is to build a predictive model and find out the sales of each product at a store.

Goals and possible solutions:

This report analyzes data from a retail chain called Big Mart. It is an attempt to try and predict sales of products across the stores. It also gives overview of sales of products across the various stores. This report also shows the data in a visual format to better understand the various aspects of the stores and the products. Instead of using just one prediction model, we will be testing the performance of 7 models. Later, based on an ensemble of models (multiple models used by weights of each model’s prediction) a final model is chosen as the predictive model and used to predict the sales of each item at each outlet. This is a prediction-based exercise where we try to predict the sales of each product in multiple stores. The data contains attributes like item\_weight, MRP etc. which help us to train a model for good accuracy.

Objective:

The objective of this exercise is to build multiple predictive models and test the accuracy of predictions from these various models. Data cleaning and data quality are some of the important tasks along this path.

Data Origin: The data has been curated and downloaded from “Kaggle.com”, a popular open source data repository.

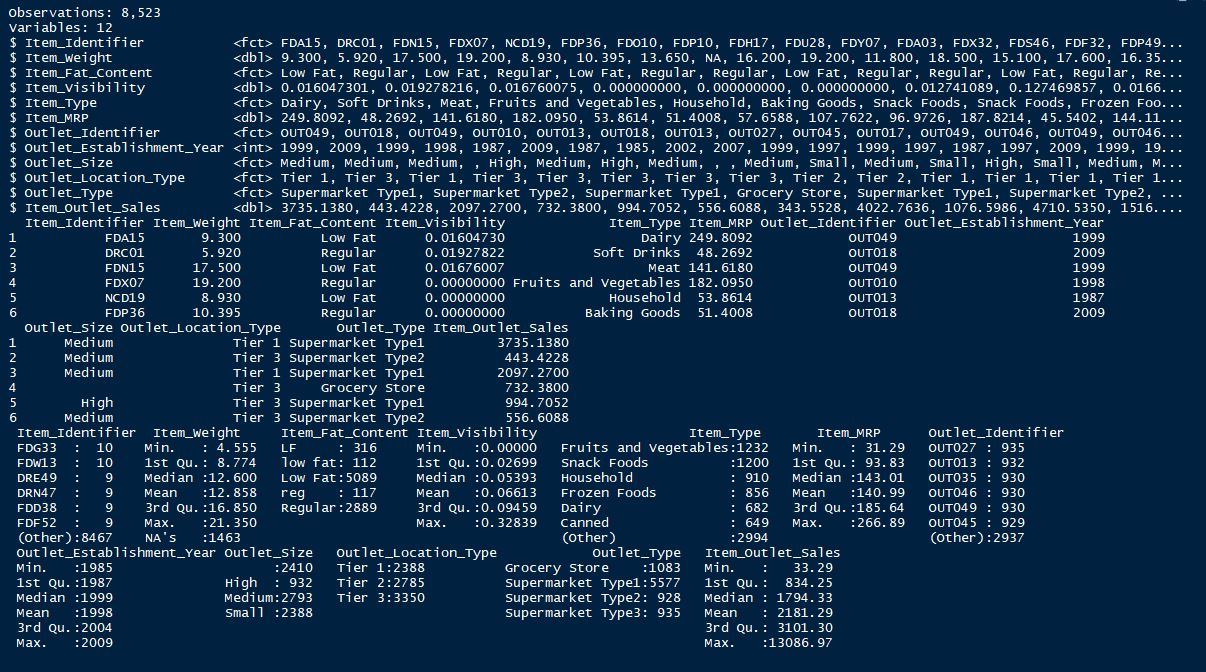


There is a total of 12 variables

* Item\_Identifier
* Item\_weight
* Item\_Fat\_Content
* Item\_Visibility
* Item\_Type
* Item\_MRP
* Outlet\_Identifier
* Outlet\_Establishment\_Year
* Outlet\_Size
* Outlet\_Location\_type
* Outlet\_Type
* Item\_Outlet\_Sales

**Data Exploration and Preprocessing**

We’ll be performing some basic data exploration here and come up with some inferences about the data. We’ll try to figure out some irregularities. Based on the outcome of the data exploration, we can make changes to attributes, transform them or perform some other tasks to ensure that while we build the model, there are no issues faced.

The first step in any analysis is to try and understand the type of data being dealt with. We chose to use *“glimpse ()”*, *“head ()”* and *“summary ()”* to get a feel of the data.

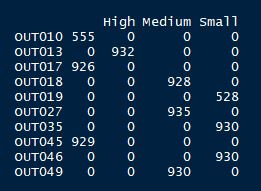
It can be seen from the data that the Item\_Fat\_Content column contains observations that need cleaning. The acceptable values in this column is "Low Fat" or "Regular". The different observations which do not conform to these values are stored as LF, low fat or reg. Therefore, these need to be cleaned.

Additionally, there are also 1463 missing values for the Item\_Weight column. These missing values will severly affect the formulation of Machine Learning models and hence have to be imputed. For this analysis, we are using Knn imputation. This method imputes a value based on other observations with similar values for the other variables in the dataset.

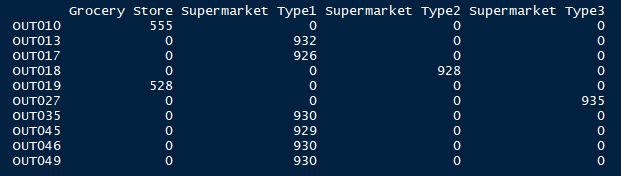
From our observation, we also noted that Outlets are divided based on Size(High, Medium, Small) and Type(Grocery Store, Supermarket type1, Supermarket type2, Supermarket type3)

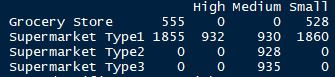
To better understand the distribution of stores, we create tables to see the dispersion as follows:

1. Outlet by Size



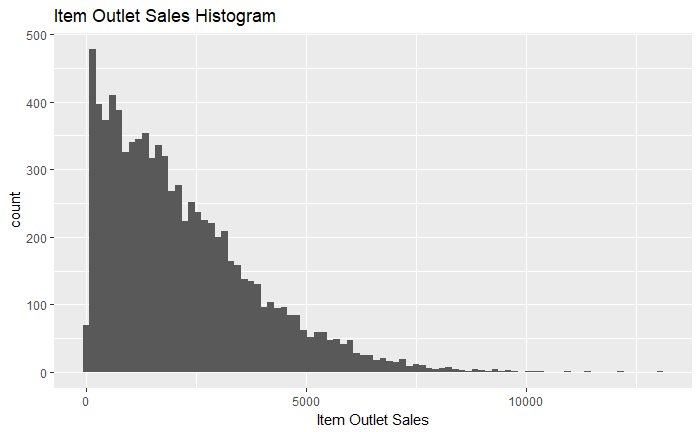
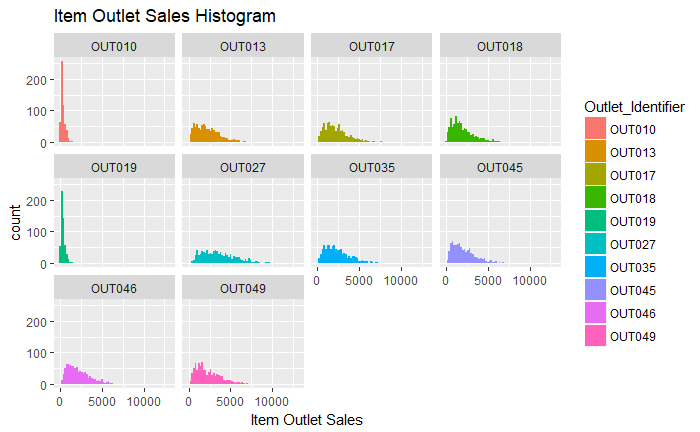
We can clearly see that there are 3 outlets which aren’t correclty labelled for Size. Upon deeper investigation, We see that OUT010 is a Grocery Store and OUT017 is a Supermarket Type2 and OUT045 is a Supermarket Type1. We will assign “Small” to OUT010 and OUT017. Also, we will assign “Medium” to OUT045.

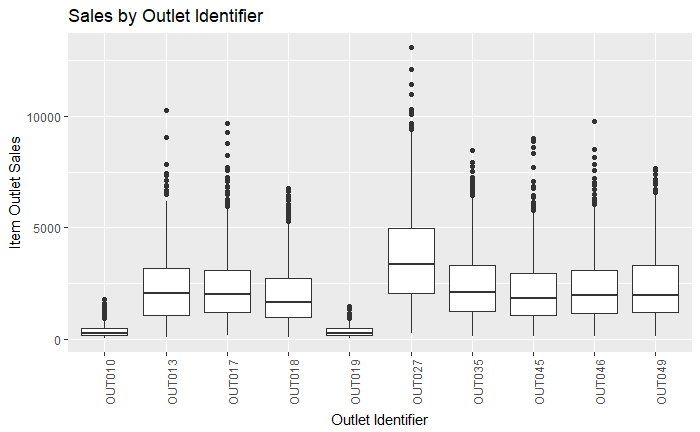
1. Outlet by Type
2. Type by Size



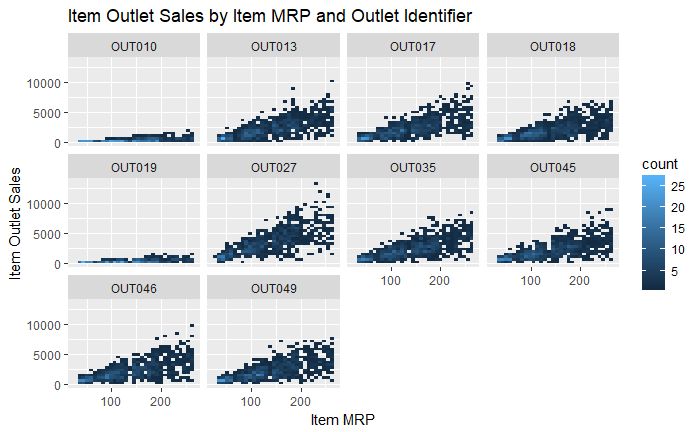
We impute values where it is necessary. This step typically involves replacing or neglecting the missing values, deriving and adding some missing values or utilizing the best estimate (mean) of values for missing items. We also need to ensure that outliers are handled properly. Though outlier removal is very important in regression techniques, advanced tree-based algorithms are impervious to outliers.

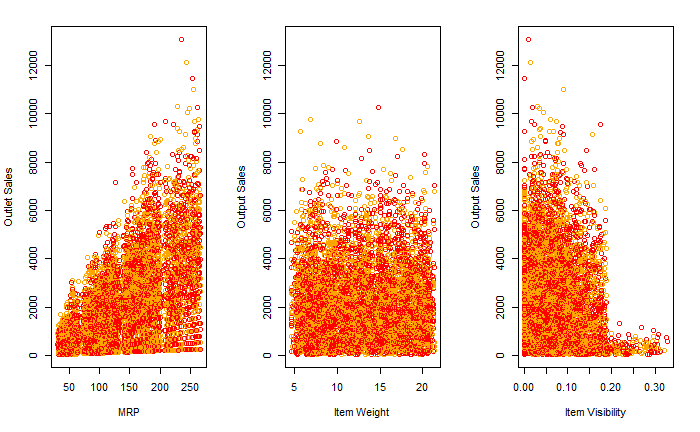
To visualize the data we used histograms, box plots and scatterplots which showed the items outlets sales segregated by amount, sales by outlet, sales by MRP and visibility. Some of the visualizations are as shown below:

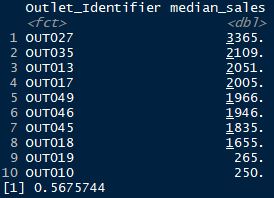
1. Histogram of Item Outlet Sales
2. Plots of Item Outlet sales grouped by outlet identifier.
3. Boxplot of Sales grouped by Outlet Identifier



1. Item Outlet Sales by Item MRP



1. Scatterplot of MRP, Item Weight and Item Visibility
2. Median Sales by Location and Correlation of Item Outlet Sales and Item MRP (Value = 0.5675744)



These charts and visualizations show that most Outlet Sales occur between the ranges of 0 and 5000. The histogram of item outlet sales grouped by Outlets shows that most of the low item sales were in outlets OUT010 and OUT019. Further examination showed that these are small grocery stores as opposed to larger Supermarkets. Therefore, these low sales numbers are justified. The boxplot confirms with the above fact. The outlet with highest sales was OUT027.

Highest sales would naturally follow that the outlet is a larger outlet. However, this is not the case. It was a “Medium” store but the only store which is identified as a Supermarket Type3.

There is a moderate positive correlation between Item outlet sales and Items MRP. This is confirmed when we run a correlation test which yields a coefficient of correlation of 0.5675744.

**Data Mining Techniques and Implementation**

Ours is a prediction problem and our strategy for building the best model is to use some form of an Ensemble learning (Bootstrap Aggregation). Ensemble Learning is a type of Supervised Learning Technique in which the basic idea is to generate multiple models on a training dataset and then simply combining(average) their Output Rules or their Hypothesis to generate a Strong Model which performs very well and does not overfit and which balances the Bias-Variance Tradeoff too.

In general, ensembling is a technique of combining two or more algorithms of similar or

dissimilar types called base learners. This is done to make a more robust system which

incorporates the predictions from all the base learners. It can be understood as conference

room meeting between multiple traders to decide on whether the price of a stock will

go up or not.

Let’s assume we have a sample dataset of 1000 instances (x) and we are using the CART

algorithm. Bootstrap Aggregation or Bagging of the CART algorithm would work as follows.

Create many (e.g. 100) random sub-samples of our dataset with replacement.

Train a CART model on each sample.

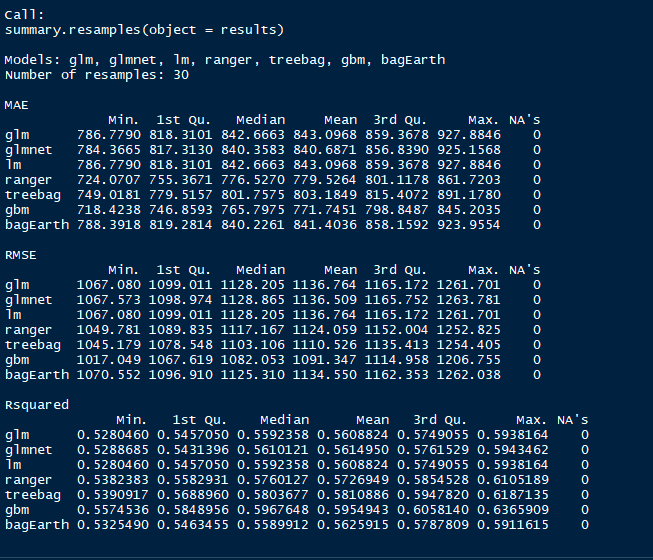
Given a new dataset, calculate the average prediction from each model.

The numerous models which we first test individually with 30 resamples are:

* Generalized Linear Model (glm)
* Generalized Linear Model with lasso and elastic-net model paths (glmnet)
* Linear Regression Model (lm)
* Random Forest Regression Model (ranger)
* Classification with a Bagging Model (treebag)
* Generalized boosted Regression Model (gbm)
* Bagging wrapper Model (bagEarth)

Before the model can be built, the columns Item\_Identifier and Outlet\_Identifier were removed. These columns had zero variance because they are unique to each item and each outlet. Next the data was split into a train set and a test set. The test set is used to test the accuracy of the model.

The summary of the models based on 3 performance metrices (MAE, RMSE and Rsquared) is as shown below

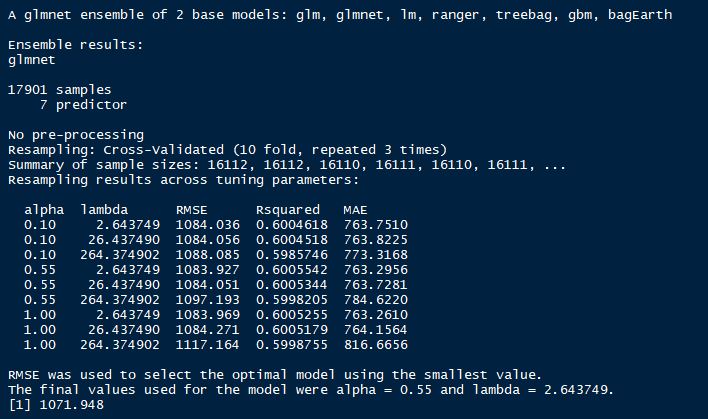


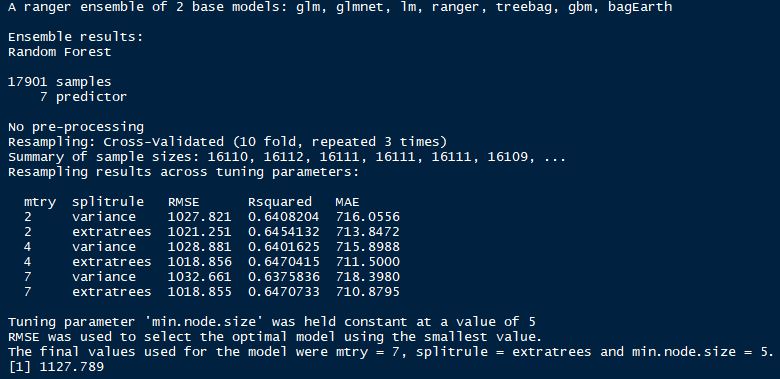
From the summary we can see that when comparing the RMSE, the best performing model is the gbm model. This model has an out of sample error of 1091.347.

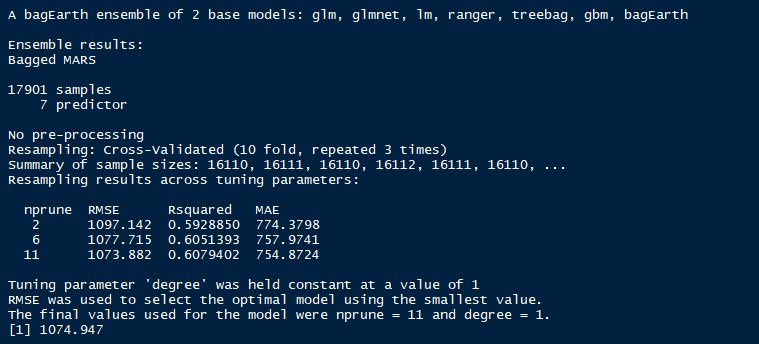
Using the results shown above, we build Ensembles to see if any of them perform better than the base models. If none of the Ensembles perform better than the base models, then gbm will be used to make predictions about the sales of items across the stores.

Next step is to build the Ensembles and test their performance. We built 3 Ensembles whose results are shown below along with their corresponding RMSE:

1. GLMNET Ensemble (RMSE: 1071.948)



1. Random Forest Ensemble (RMSE: 1127.789)
2. Bagging Ensemble (RMSE: 1074.947)

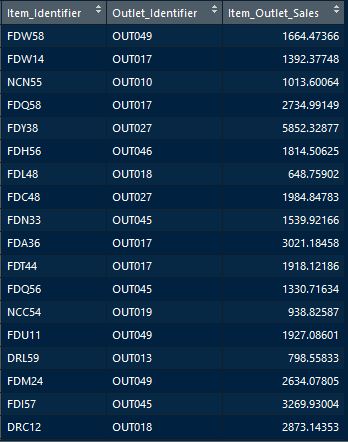


It can be clearly seen that the GLMNET Ensemble has the best performance of the lot and will be used for our prediction of sales at a product and store level.

**Predictions**

Based on the GLMNET Ensemble, we are getting accurate predictions for sales of products at individual stores and total sales for a store as seen below

1. Item\_Sales at Outlets



1. Total sales at outlets

