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

Variable	Description
Item_Identifier	Unique product ID
Item_Weight	Weight of product
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
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
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.

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.

```
Observations: 8,523
Variables: 12
$ Item_Identifier          <fct> FDA15, DRC01, FDN15, FDX07, NCD19, FDP36, FDO10, FDP10, FDH17, FDU28, FDY07, FDA03, FDX32, FDS46, FDF32, FDP49...
$ Item_Weight              <dbl> 9.300, 5.920, 17.500, 19.200, 8.930, 10.395, 13.650, NA, 16.200, 19.200, 11.800, 18.500, 15.100, 17.600, 16.35...
$ Item_Fat_Content         <fct> Low Fat, Regular, Low Fat, Regular, Low Fat, Regular, Regular, Low Fat, Regular, Regular, Low Fat, Regular, Re...
$ Item_Visibility          <dbl> 0.016047301, 0.019278216, 0.016760075, 0.000000000, 0.000000000, 0.000000000, 0.012741089, 0.127469857, 0.0166...
$ Item_Type                <fct> Dairy, Soft Drinks, Meat, Fruits and Vegetables, Household, Baking Goods, Snack Foods, Snack Foods, Frozen Foo...
$ Item_MRP                 <dbl> 249.8092, 48.2692, 141.6180, 182.0950, 53.8614, 51.4008, 57.6588, 107.7622, 96.9726, 187.8214, 45.5402, 144.11...
$ Outlet_Identifier        <fct> OUT049, OUT018, OUT049, OUT010, OUT013, OUT018, OUT013, OUT027, OUT045, OUT017, OUT049, OUT046, OUT049, OUT046...
$ Outlet_Establishment_Year <int> 1999, 2009, 1999, 1998, 1987, 2009, 1987, 1985, 2002, 2007, 1999, 1997, 1999, 1997, 1987, 1997, 2009, 1999, 19...
$ Outlet_Size              <fct> Medium, Medium, Medium, , High, Medium, High, Medium, , , Medium, Small, Medium, Small, High, Small, Medium, M...
$ Outlet_Location_Type     <fct> Tier 1, Tier 3, Tier 1, Tier 3, Tier 3, Tier 3, Tier 3, Tier 3, Tier 2, Tier 2, Tier 1, Tier 1, Tier 1, Tier 1...
$ Outlet_Type              <fct> Supermarket Type1, Supermarket Type2, Supermarket Type1, Grocery Store, Supermarket Type1, Supermarket Type1, Supermarket Type2, ...
$ Item_Outlet_Sales        <dbl> 3735.1380, 443.4228, 2097.2700, 732.3800, 994.7052, 556.6088, 343.5528, 4022.7636, 1076.5986, 4710.5350, 1516....
  Item_Identifier Item_Weight Item_Fat_Content Item_Visibility              Item_Type Item_MRP Outlet_Identifier Outlet_Establishment_Year
1           FDA15       9.300          Low Fat      0.01604730                  Dairy 249.8092           OUT049                      1999
2           DRC01       5.920          Regular      0.01927822            Soft Drinks  48.2692           OUT018                      2009
3           FDN15      17.500          Low Fat      0.01676007                   Meat 141.6180           OUT049                      1999
4           FDX07      19.200          Regular      0.00000000 Fruits and Vegetables 182.0950           OUT010                      1998
5           NCD19       8.930          Low Fat      0.00000000              Household  53.8614           OUT013                      1987
6           FDP36      10.395          Regular      0.00000000           Baking Goods  51.4008           OUT018                      2009
  Outlet_Size Outlet_Location_Type        Outlet_Type Item_Outlet_Sales
1      Medium               Tier 1 Supermarket Type1         3735.1380
2      Medium               Tier 3 Supermarket Type2          443.4228
3      Medium               Tier 1 Supermarket Type1         2097.2700
4                           Tier 3      Grocery Store          732.3800
5        High               Tier 3 Supermarket Type1          994.7052
6      Medium               Tier 3 Supermarket Type2          556.6088
 Item_Identifier  Item_Weight      Item_Fat_Content Item_Visibility                  Item_Type       Item_MRP      Outlet_Identifier
 FDG33  :  10    Min.   : 4.555   LF     : 316     Min.   :0.00000   Fruits and Vegetables:1232   Min.   : 31.29   OUT027 : 935
 FDW13  :  10    1st Qu.: 8.774   low fat: 112     1st Qu.:0.02699   Snack Foods           :1200   1st Qu.: 93.83   OUT013 : 932
 DRE49  :   9    Median :12.600   Low Fat:5089     Median :0.05393   Household             : 910   Median :143.01   OUT035 : 930
 DRN47  :   9    Mean   :12.858   reg    : 117     Mean   :0.06613   Frozen Foods          : 856   Mean   :140.99   OUT046 : 930
 FDD38  :   9    3rd Qu.:16.850   Regular:2889     3rd Qu.:0.09459   Dairy                 : 682   3rd Qu.:185.64   OUT049 : 930
 FDF52  :   9    Max.   :21.350                    Max.   :0.32839   Canned                : 649   Max.   :266.89   OUT045 : 929
 (Other):8467    NA's   :1463                                        (Other)               :2994                    (Other):2937
 Outlet_Establishment_Year Outlet_Size   Outlet_Location_Type          Outlet_Type   Item_Outlet_Sales
 Min.   :1985                    :2410   Tier 1:2388          Grocery Store    :1083   Min.   :   33.29
 1st Qu.:1987              High  : 932   Tier 2:2785          Supermarket Type1:5577   1st Qu.:  834.25
 Median :1999              Medium:2793   Tier 3:3350          Supermarket Type2: 928   Median : 1794.33
 Mean   :1998              Small :2388                        Supermarket Type3: 935   Mean   : 2181.29
 3rd Qu.:2004                                                                          3rd Qu.: 3101.30
 Max.   :2009                                                                          Max.   :13086.97
```

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:

a) Outlet by Size

	High	Medium	Small
OUT010	555	0	0
OUT013	0	932	0
OUT017	926	0	0
OUT018	0	0	928
OUT019	0	0	0
OUT027	0	0	935
OUT035	0	0	0
OUT045	929	0	0
OUT046	0	0	0
OUT049	0	0	930

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.

b) Outlet by Type

	Grocery Store	Supermarket Type1	Supermarket Type2	Supermarket Type3
OUT010	555	0	0	0
OUT013	0	932	0	0
OUT017	0	926	0	0
OUT018	0	0	928	0
OUT019	528	0	0	0
OUT027	0	0	0	935
OUT035	0	930	0	0
OUT045	0	929	0	0
OUT046	0	930	0	0
OUT049	0	930	0	0

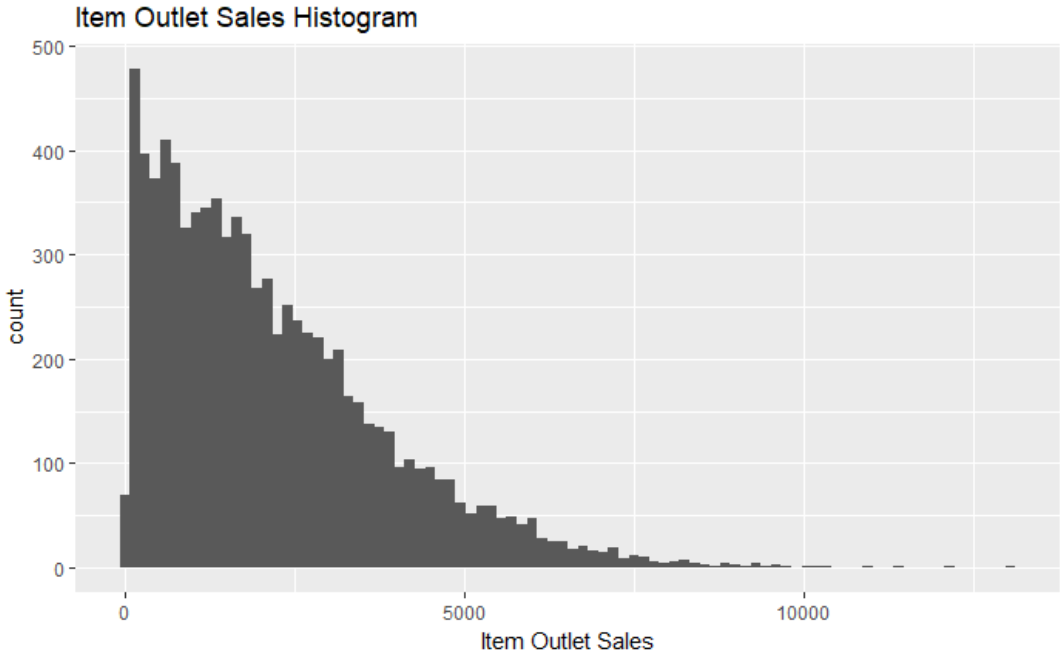
c) Type by Size

	High	Medium	Small
Grocery Store	555	0	0	528
Supermarket Type1	1855	932	930	1860
Supermarket Type2	0	0	928	0
Supermarket Type3	0	0	935	0

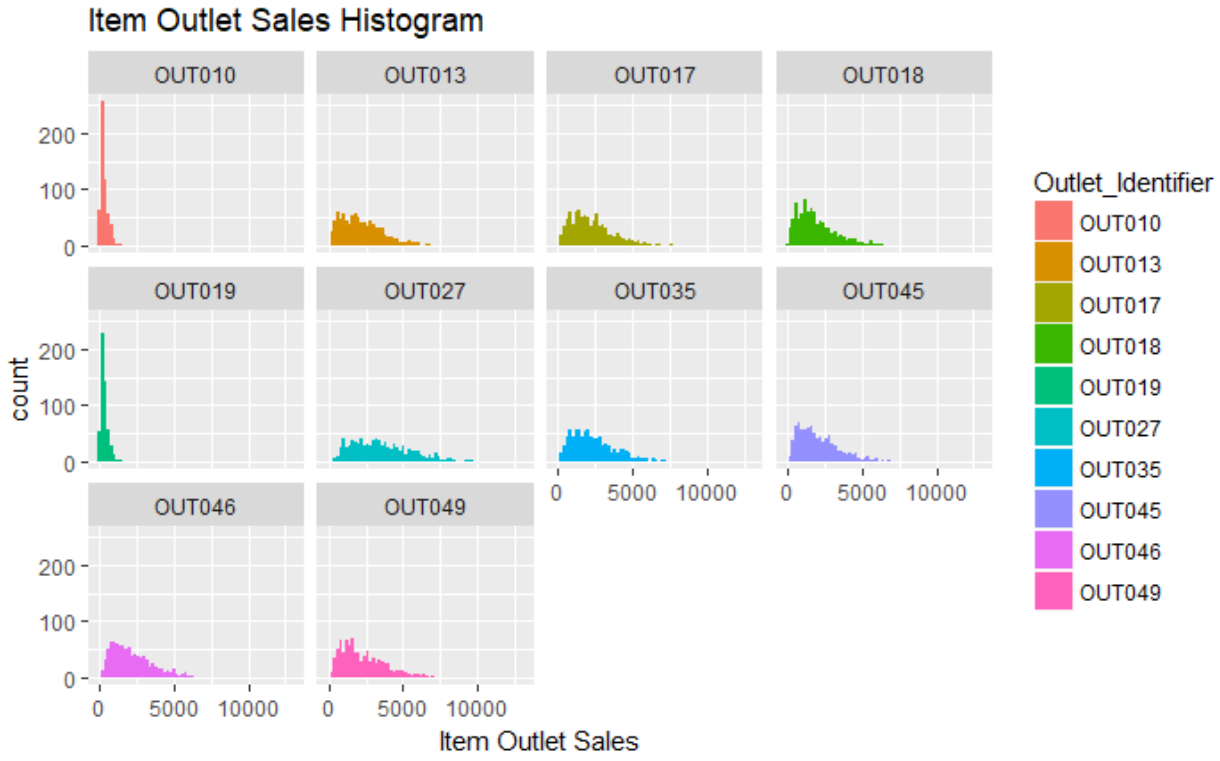
We impute values where it is necessary. This step typically involves replacing or neglecting the missing values, deriving and adding some missing values or utilizing the best estimate (mean) of values for missing items. We also need to ensure that outliers are handled properly. Though outlier removal 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:

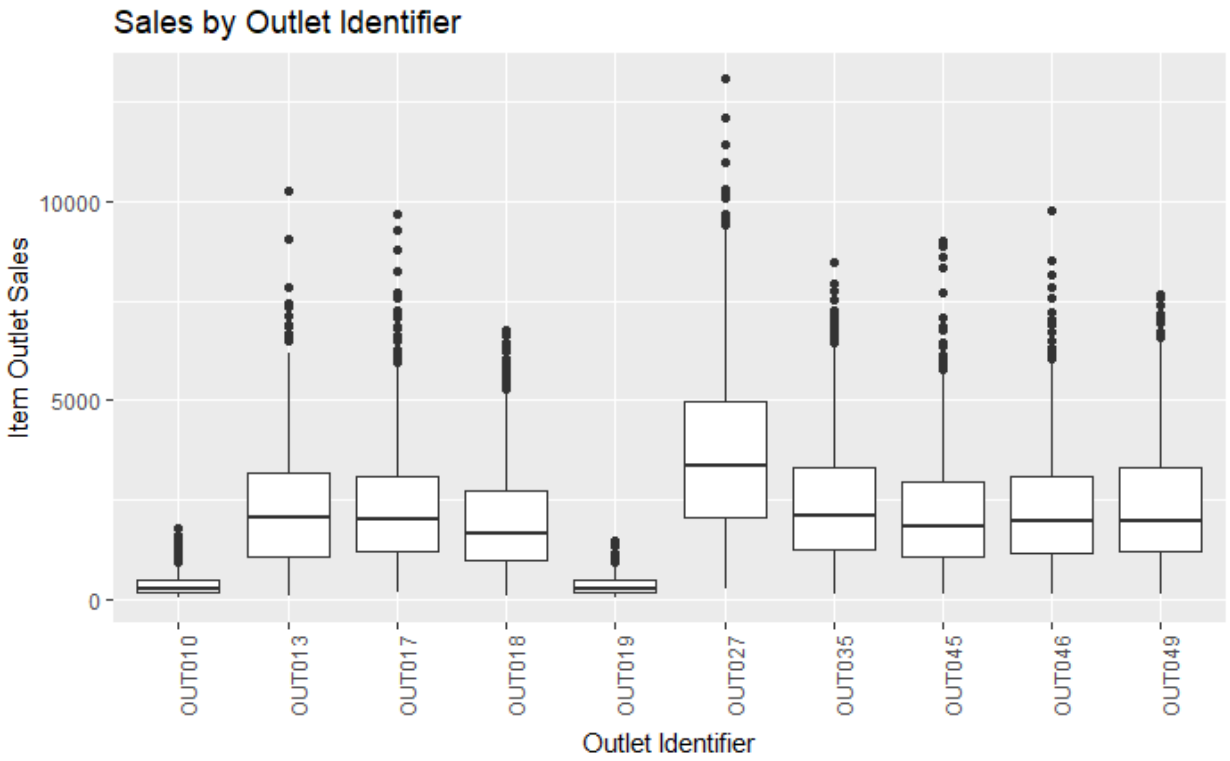
1) Histogram of Item Outlet Sales



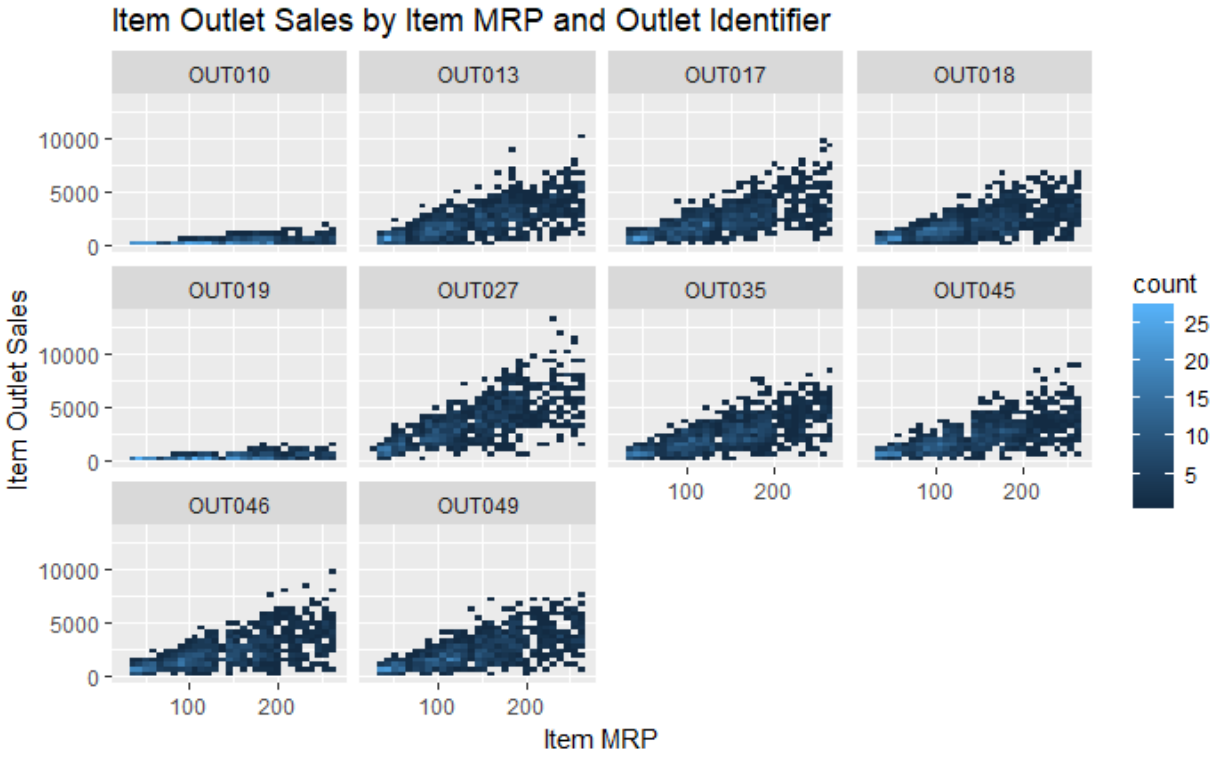
2) Plots of Item Outlet sales grouped by outlet identifier.



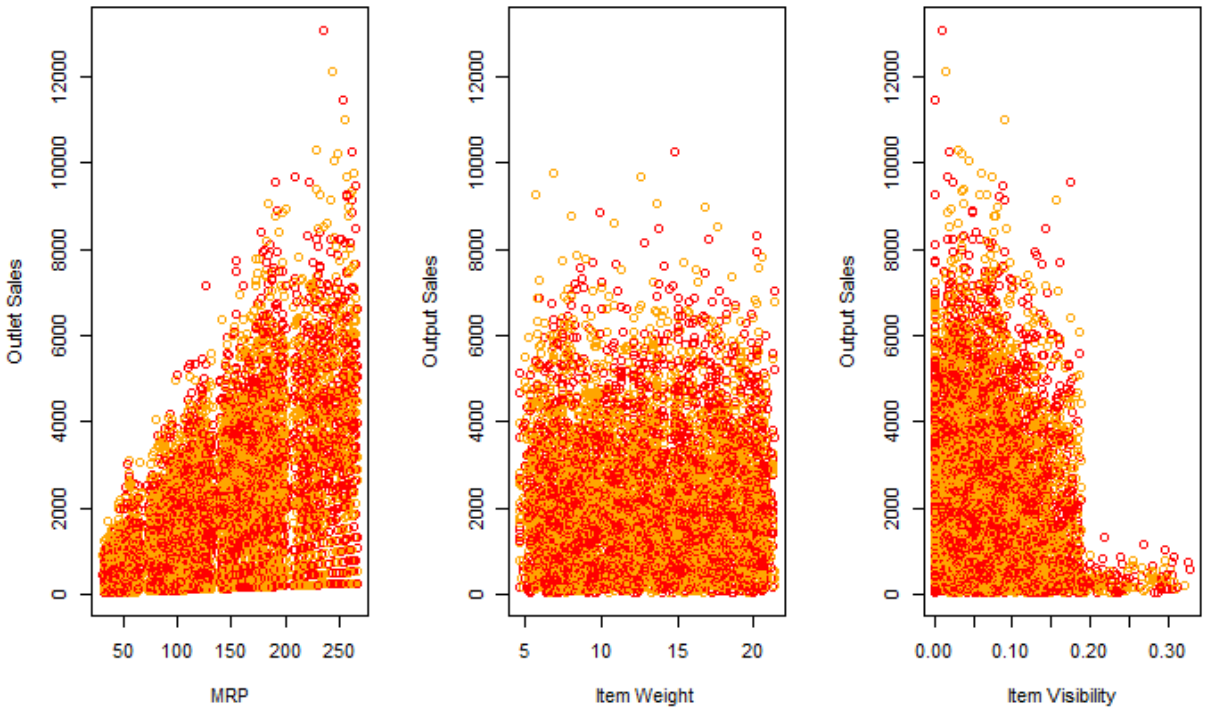
3) Boxplot of Sales grouped by Outlet Identifier



4) Item Outlet Sales by Item MRP



5) Scatterplot of MRP, Item Weight and Item Visibility



6) Median Sales by Location and Correlation of Item Outlet Sales and Item MRP (Value = 0.5675744)

	Outlet_Identifier	median_sales
	<fct>	<dbl>
1	OUT027	3365.
2	OUT035	2109.
3	OUT013	2051.
4	OUT017	2005.
5	OUT049	1966.
6	OUT046	1946.
7	OUT045	1835.
8	OUT018	1655.
9	OUT019	265.
10	OUT010	250.

[1] 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

```
Call:
summary.resamples(object = results)

Models: glm, glmnet, lm, ranger, treebag, gbm, bagEarth
Number of resamples: 30

MAE
            Min.  1st Qu.   Median     Mean  3rd Qu.     Max. NA's
glm      786.7790 818.3101 842.6663 843.0968 859.3678 927.8846    0
glmnet   784.3665 817.3130 840.3583 840.6871 856.8390 925.1568    0
lm       786.7790 818.3101 842.6663 843.0968 859.3678 927.8846    0
ranger   724.0707 755.3671 776.5270 779.5264 801.1178 861.7203    0
treebag  749.0181 779.5157 801.7575 803.1849 815.4072 891.1780    0
gbm      718.4238 746.8593 765.7975 771.7451 798.8487 845.2035    0
bagEarth 788.3918 819.2814 840.2261 841.4036 858.1592 923.9554    0

RMSE
            Min.  1st Qu.   Median     Mean  3rd Qu.     Max. NA's
glm      1067.080 1099.011 1128.205 1136.764 1165.172 1261.701    0
glmnet   1067.573 1098.974 1128.865 1136.509 1165.752 1263.781    0
lm       1067.080 1099.011 1128.205 1136.764 1165.172 1261.701    0
ranger   1049.781 1089.835 1117.167 1124.059 1152.004 1252.825    0
treebag  1045.179 1078.548 1103.106 1110.526 1135.413 1254.405    0
gbm      1017.049 1067.619 1082.053 1091.347 1114.958 1206.755    0
bagEarth 1070.552 1096.910 1125.310 1134.550 1162.353 1262.038    0

Rsquared
              Min.   1st Qu.    Median      Mean   3rd Qu.      Max. NA's
glm      0.5280460 0.5457050 0.5592358 0.5608824 0.5749055 0.5938164    0
glmnet   0.5288685 0.5431396 0.5610121 0.5614950 0.5761529 0.5943462    0
lm       0.5280460 0.5457050 0.5592358 0.5608824 0.5749055 0.5938164    0
ranger   0.5382383 0.5582931 0.5760127 0.5726949 0.5854528 0.6105189    0
treebag  0.5390917 0.5688960 0.5803677 0.5810886 0.5947820 0.6187135    0
gbm      0.5574536 0.5848956 0.5967648 0.5954943 0.6058140 0.6365909    0
bagEarth 0.5325490 0.5463455 0.5589912 0.5625915 0.5787809 0.5911615    0
```

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)

```
A glmnet ensemble of 2 base models: glm, glmnet, lm, ranger, treebag, gbm, bagEarth

Ensemble results:
glmnet

17901 samples
    7 predictor

No pre-processing
Resampling: Cross-Validated (10 fold, repeated 3 times)
Summary of sample sizes: 16112, 16112, 16110, 16111, 16110, 16111, ...
Resampling results across tuning parameters:

  alpha  lambda      RMSE      Rsquared   MAE
  0.10     2.643749  1084.036  0.6004618  763.7510
  0.10    26.437490  1084.056  0.6004518  763.8225
  0.10   264.374902  1088.085  0.5985746  773.3168
  0.55     2.643749  1083.927  0.6005542  763.2956
  0.55    26.437490  1084.051  0.6005344  763.7281
  0.55   264.374902  1097.193  0.5998205  784.6220
  1.00     2.643749  1083.969  0.6005255  763.2610
  1.00    26.437490  1084.271  0.6005179  764.1564
  1.00   264.374902  1117.164  0.5998755  816.6656

RMSE was used to select the optimal model using the smallest value.
The final values used for the model were alpha = 0.55 and lambda = 2.643749.
[1] 1071.948
```

2) Random Forest Ensemble (RMSE: 1127.789)

```
A ranger ensemble of 2 base models: glm, glmnet, lm, ranger, treebag, gbm, bagEarth

Ensemble results:
Random Forest

17901 samples
    7 predictor

No pre-processing
Resampling: Cross-Validated (10 fold, repeated 3 times)
Summary of sample sizes: 16110, 16112, 16111, 16111, 16111, 16109, ...
Resampling results across tuning parameters:

  mtry  splitrule   RMSE      Rsquared   MAE
  2     variance    1027.821  0.6408204  716.0556
  2     extratrees  1021.251  0.6454132  713.8472
  4     variance    1028.881  0.6401625  715.8988
  4     extratrees  1018.856  0.6470415  711.5000
  7     variance    1032.661  0.6375836  718.3980
  7     extratrees  1018.855  0.6470733  710.8795

Tuning parameter 'min.node.size' was held constant at a value of 5
RMSE was used to select the optimal model using the smallest value.
The final values used for the model were mtry = 7, splitrule = extratrees and min.node.size = 5.
[1] 1127.789
```

3) Bagging Ensemble (RMSE: 1074.947)

```
A bagEarth ensemble of 2 base models: glm, glmnet, lm, ranger, treebag, gbm, bagEarth

Ensemble results:
Bagged MARS

17901 samples
    7 predictor

No pre-processing
Resampling: Cross-Validated (10 fold, repeated 3 times)
Summary of sample sizes: 16110, 16111, 16110, 16112, 16111, 16110, ...
Resampling results across tuning parameters:

  nprune  RMSE      Rsquared   MAE
   2      1097.142  0.5928850  774.3798
   6      1077.715  0.6051393  757.9741
  11      1073.882  0.6079402  754.8724

Tuning parameter 'degree' was held constant at a value of 1
RMSE was used to select the optimal model using the smallest value.
The final values used for the model were nprune = 11 and degree = 1.
[1] 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

a) Item_Sales at Outlets

Item_Identifier	Outlet_Identifier	Item_Outlet_Sales
FDW58	OUT049	1664.47366
FDW14	OUT017	1392.37748
NCN55	OUT010	1013.60064
FDQ58	OUT017	2734.99149
FDY38	OUT027	5852.32877
FDH56	OUT046	1814.50625
FDL48	OUT018	648.75902
FDC48	OUT027	1984.84783
FDN33	OUT045	1539.92166
FDA36	OUT017	3021.18458
FDT44	OUT017	1918.12186
FDQ56	OUT045	1330.71634
NCC54	OUT019	938.82587
FDU11	OUT049	1927.08601
DRL59	OUT013	798.55833
FDM24	OUT049	2634.07805
FDI57	OUT045	3269.93004
DRC12	OUT018	2873.14353

b) Total sales at outlets

Outlet_Identifier	Store_Total
OUT049	1444387.0
OUT017	1482058.3
OUT010	169084.4
OUT027	2351685.0
OUT046	1368594.8
OUT018	1202668.8
OUT045	1415065.9
OUT019	152317.2
OUT013	1410325.5
OUT035	1412529.0