Predict Future Sales

Vectorized Brains

Rohith Yogi Nomula
IMT2016072

Siddarth Reddy Desu
IMT2016037

Srujan Swaroop Gajula
IMT2016033

Abstract—The Main objective in this project is predicting the future sales of the products. We are provided with daily historical sales data. We need to forecast the total amount of products sold in every shop for the test set. The list of shops and products slightly changes every month.

1 Introduction

This Project helps in predicting the future sale of products offered by one of the largest Russian software firms - 1C Company. The data sets are provided on kaggle. The data set contains 6 files given below.

- train.csv
- test.csv
- sample-submission.csv
- items.csv
- item-categories.csv
- shops.csv

The training set has Daily historical data from January 2013 to October 2015. The test set we need to forecast the sales for these shops and products for November 2015. A sample submission file in the correct format is also provided. items file has a supplemental information about the items/products. Similarly shops file has supplemental information about the items categories.

The data fields are the contents or the various columns present in the above files. These are the various attributes through which we can obtain the desired result. After this section we talk about feature engineering where these columns come into picture.

- ID - an Id that represents a (Shop, Item) tuple within the test set
- shop-id - unique identifier of a shop
- item-id - unique identifier of a product
- item-category-id - unique identifier of item category
- item-cnt-day - number of products sold.
- item-price - current price of an item
- date - date in format dd/mm/yyyy
- date-block-num - a consecutive month number, used for convenience.
- item-name - name of item
- shop-name - name of shop
- item-category-name - name of item category

The evaluation is based on the root mean squared error (RMSE). True target values are clipped into [0,20] range.

2 EDA and Data Visualization

In this the data is visualized and analyzed based on various factors. Firstly we plotted the daily mean and daily sum behaviour over the span of a year so that the trend can be seen through daily basis.Basic features distribution count is plotted below see Figure 1.

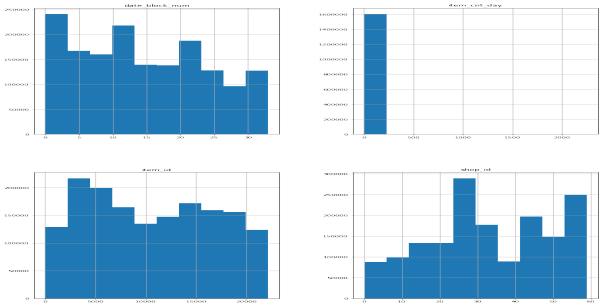

Figure 1. Distribution range and count

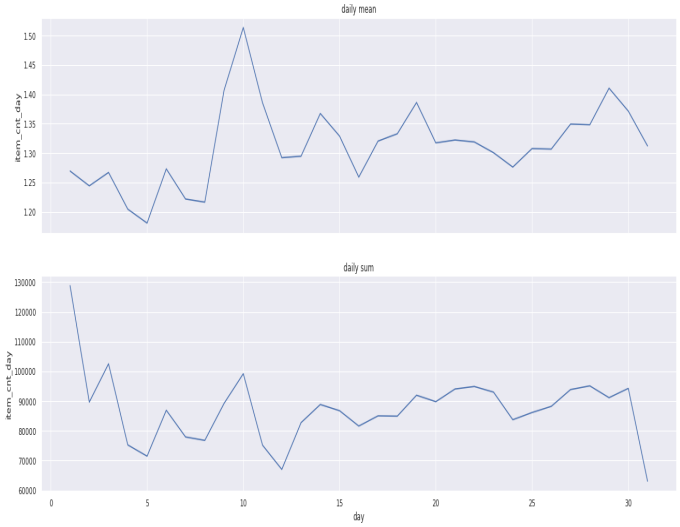

Figure 2. Sales Behaviour over the year

It is therefore clear by observing the Figure 2 how the sales behave over the span of a year .

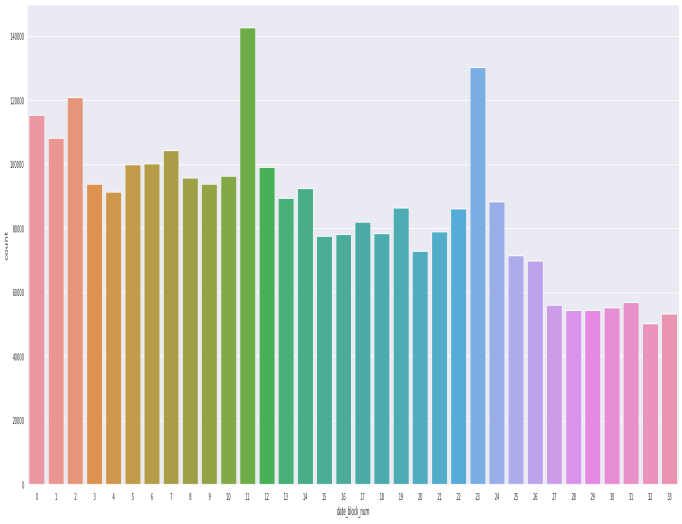

Figure 3. date-block-num counts over year

From the above visualization Figure 3 we can see that there are more entries for Decembers of 2013 and 2014.

From the above visualization Figure 4 we can see that Shop 31 has highest number of entires.

From the above visualization Figure 5 we can see that Item category 40 has highest number of entries.

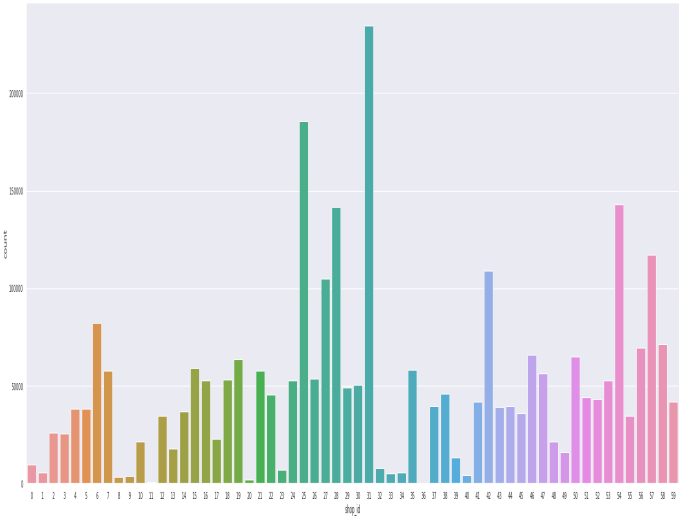

Figure 4. shop-id counts over year

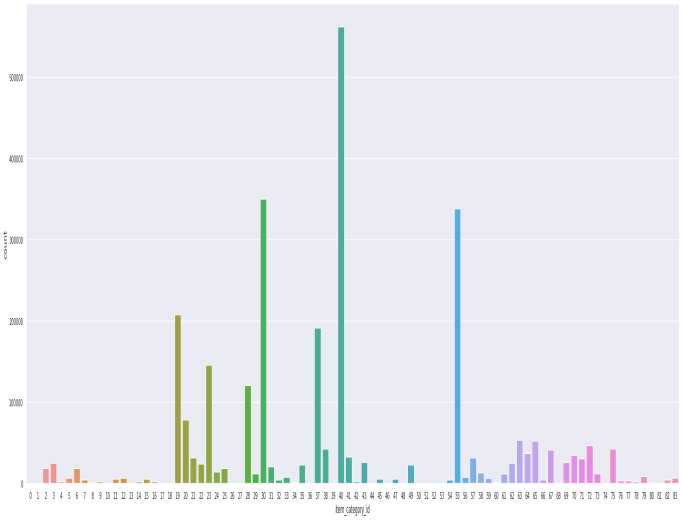

Figure 5. item-category counts over year

From the above visualization Figure 5 we can see that number of Items sold in December 2013 is greater December 2014 this is because of the holidays in this month.

From the above visualization Figure 6 we can see that shop 31 is selling highest number of items while Shop 36 and Shop 11 are selling very few items.

From the above visualization Figure 7 we can see that Item Category 40 is being sold in maximum The Item Category 19 makes maximum Revenue.see Figure 8

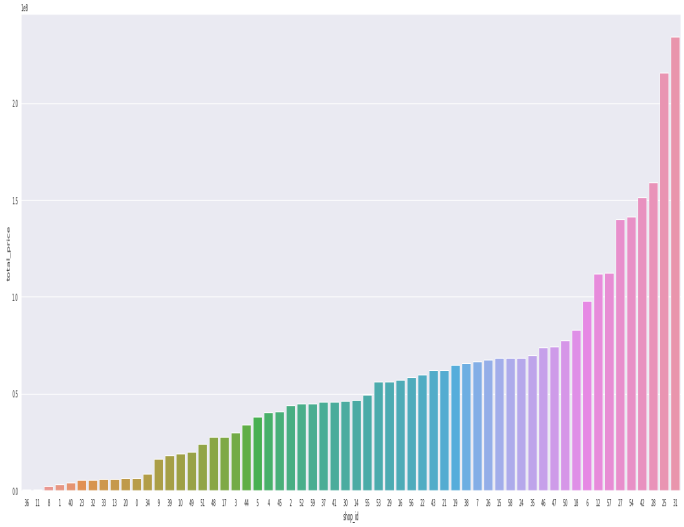

Figure 6. sales made by the shops

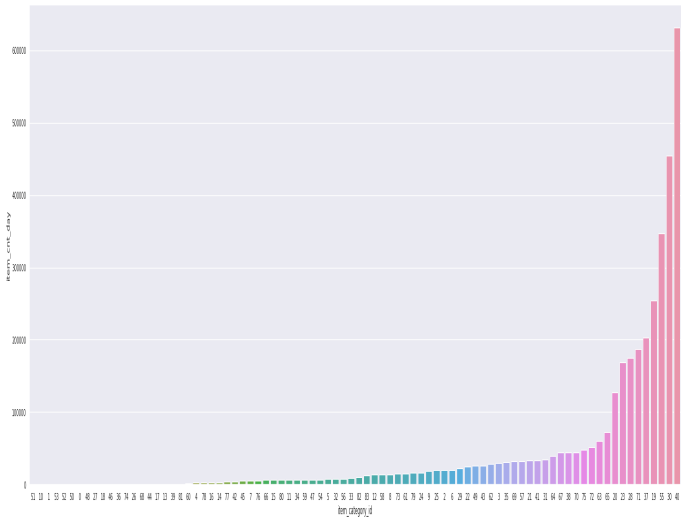

Figure 7. items sold over the year

3 Data Preprocessing

First we looked at the raw data that was provided to us in kaggle. Then we loaded the data into the code using pandas.

After which we removed the duplicate values and all the null values.

So we filled out the missing values by interpolating the data i.e looked at the trend in the values in the data fields and filled them in a similar pattern (using medians).

All the data types of the data fields were loaded accordingly.

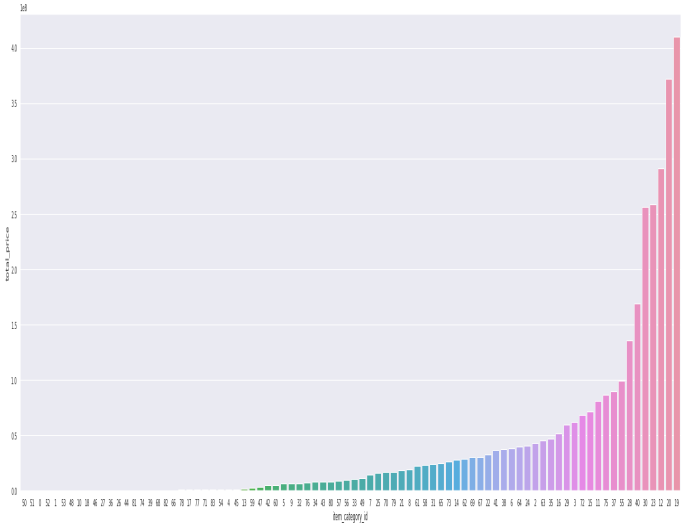

Figure 8. total revenue for each item

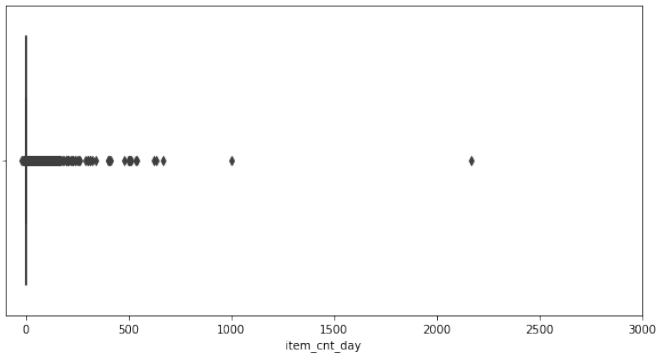

Figure 9. Outliers for item-cnt-day

As the dataset was large we split it into train and validation in the 70:30 ratio.We later thought of using the validation set for cross-referencing.

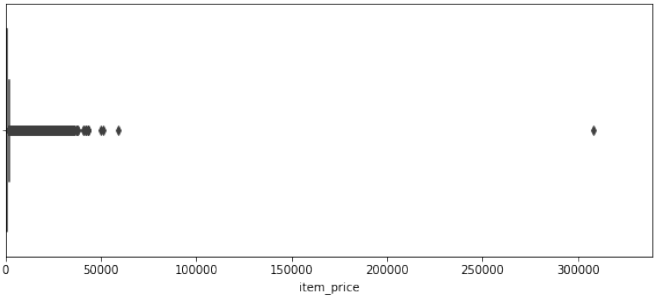

Figure 10. Outliers for item-price

All the negative values were removed.

Date format was changed from datatype of 'date' column from object to datetime

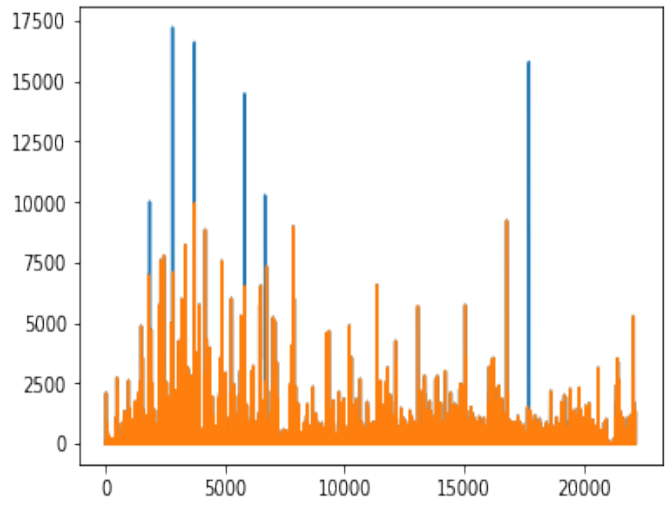

Figure 11. Outliers for item-id

We can clearly see the outliers in the item-id see Figure 22 we removed the outliers and clipped the values to 20.

4 Feature Engineering

In the context of feature engineering ,the very first attempt we started experimenting with variations of features just to capture the behaviour of features on shallow note and to get intuitive significance of attributes. Our initial experiments with features include:-

- Comparisons between item-cnt-day and various attributes(which include shop-id,item-id,item-category-id)
- frequency of sales in terms of :
  $-> $ Daily sales
  $-> $ Monthly sales
  $-> $ Yearly sales
- Dependence between individual features.
- Finding significance of relation between features.

Evidences of experiments using Data Analysis:

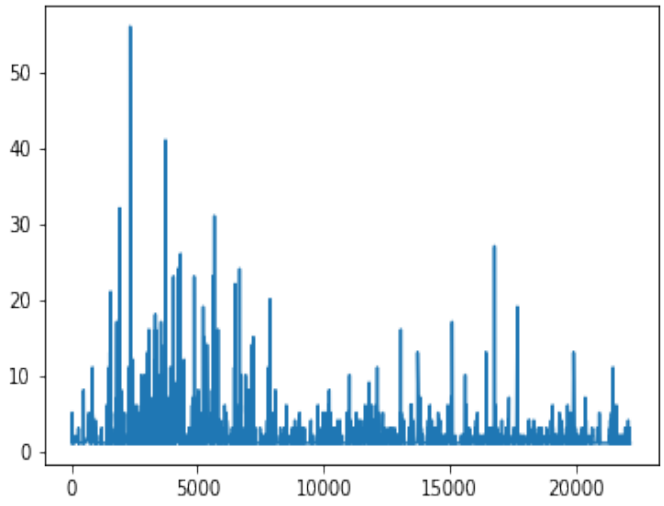

Figure 12. Item-cnt-day VS item-id

Inference:Exploration of number of sales with item-id.
Analysis:There is significant variation of item-cnt-day with item-id,which implies that item-id has significant impact on number of sales.
Inference:Exploration of number of sales with shop-id.
Analysis:Sales of items with variation of different shops,and rate of variance is significant enough which projects shop-id as assumable important attribute.

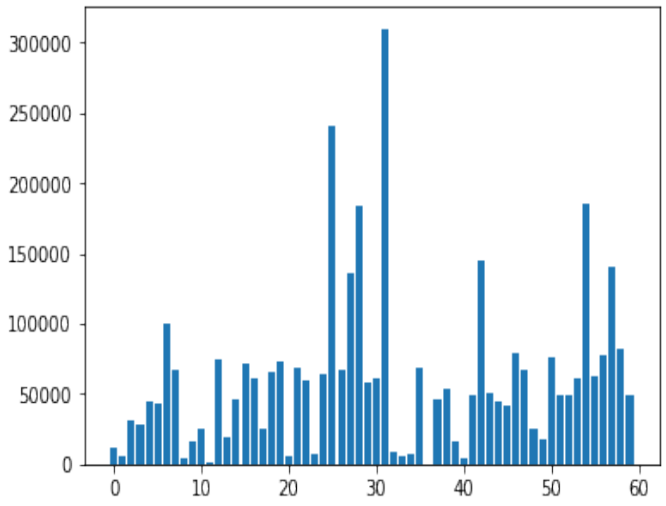

Figure 13. Item-cnt-day VS shop-id

Inference:Change in amount of sales of

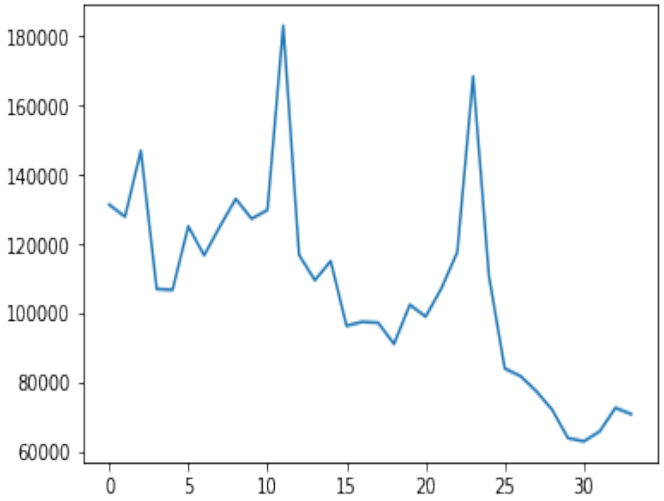

Figure 14. Item-cnt-day VS day-block-num

products with day-block-num(characteristic of month across years .
Analysis:There are several peaks in the corresponding plot which implies that there are certain months in which sales are at maximum,also there are several drops in between which has an implicit implication that sales have major dependency on months across years.
IMPLICATION:The above analysis gave rise to intuitive thought to seperate date to month and year,as months across different years have major influence on number of sales.This features turned out to be one of the major feature evolution.

Analysis:Fig 15,Fig 16 gives intuitive information about the maximum number of items of a particular category present in each shop. Inference:As per Fig 15,the item categories has less vital role in determination of sales.

Analysis:Fig 17 projects statistical information about frequency of items sold corresponding categories . Inference:As per Fig 17,you can see the steep fall with categories,which implies that there are less significance associated with Item-category-id. IMPLICATION:Decision made to not include item-categories-id.

Inference:As per Fig 17,you can see the steep fall with categories,which implies that there are less significance associated with Item-category-

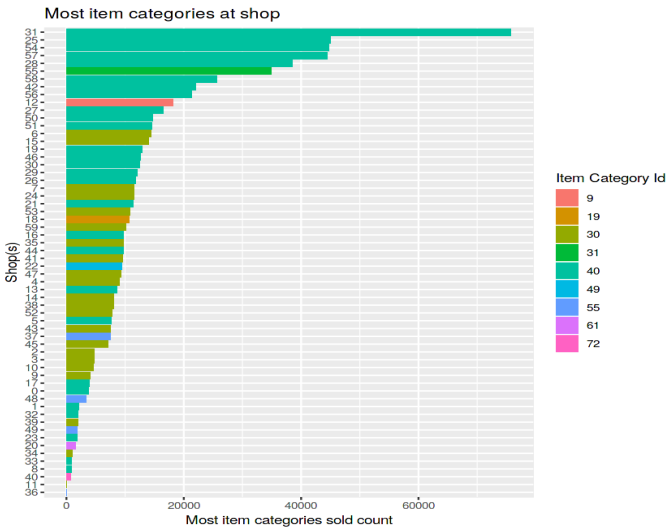

Figure 15. most-items-at-a-shop

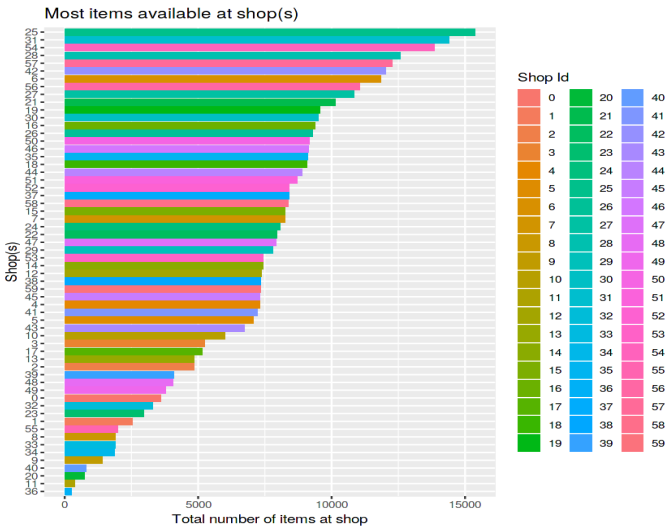

Figure 16. most-available-items

id. IMPLICATION:Decision made to not include item-categories-id.

Analysis:Fig 18,19,20 displays the information about density of sales in different period of times:

- Day
- Month
- Year

Inference:It is clearly extension to previous implication to include Day in features set as

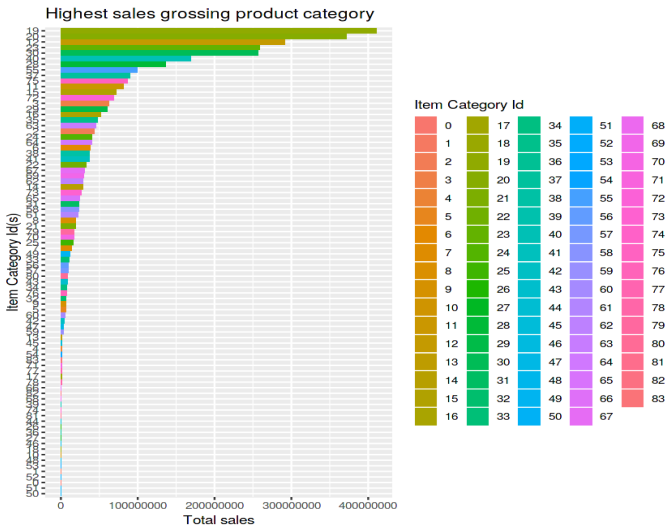

Figure 17. highest-sales-item-categories

it has a significant influence on distribution of sales which is clear evident from the figure:17. Major Implication:Inclusion of Day,Month,Year in Feature set

Major Conclusions from Co-relation matrix in figure 22.
.There is major Existence dependency and co-relation between label(item-cnt-day);

- Item-id
- Shop-Id
- Item-price

The Final conclusions of feature engineeering above Data Analysis and co-relation plot are:-

- Inclusion of Day
- Inclusion of Month
- Inclusion of Year
- Inclusion of Shop-id
- Inclusion of item-id
- Inclusion of item-price
- Expulsion of item-category attribute
- Expulsion of day block number instead replaced with feature month,year.
- Shop-Id
- Item-price

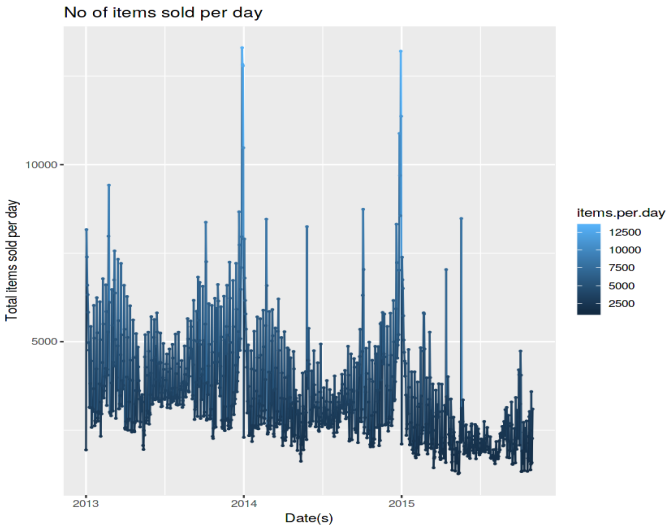

Figure 18. number-of-items-sold-perday

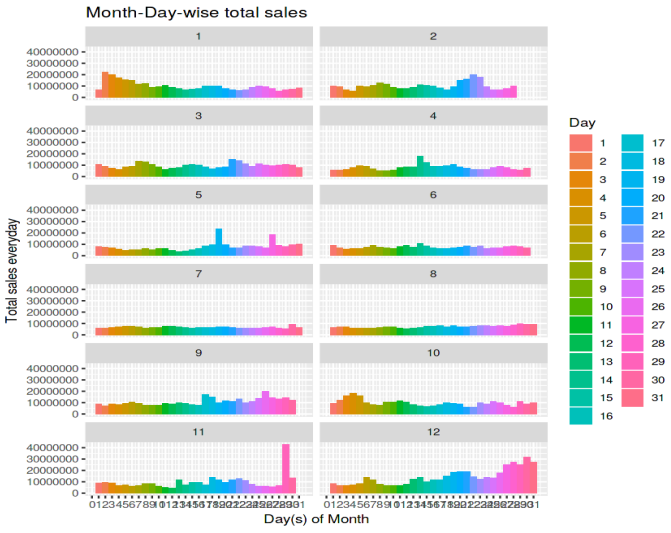

Figure 19. month-day-wise-totalsales

Analysis:Fig
With this construct the reference in text to Equation **??** is straightforward.

4.1 Figures

Placing figures is also very easy, as for example the following **??**:

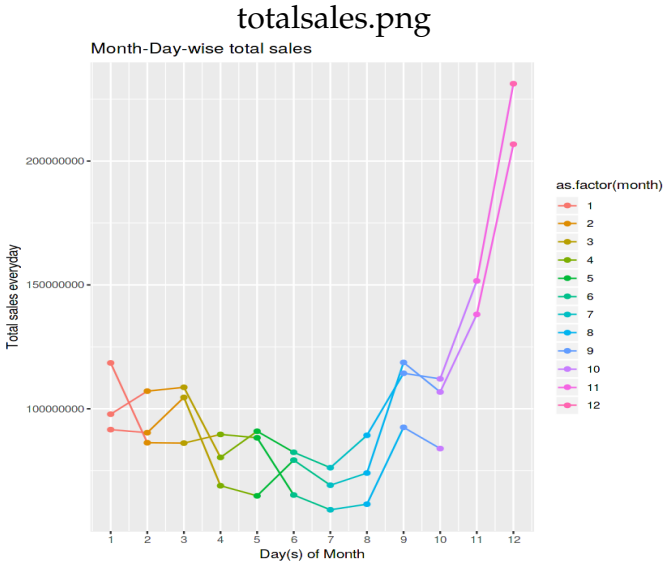

Figure 20. yearly totalsales

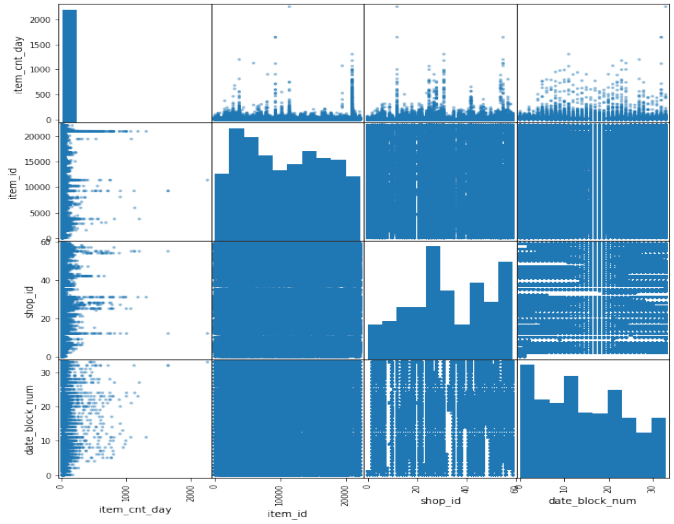

Figure 21. scatter-matrix-features

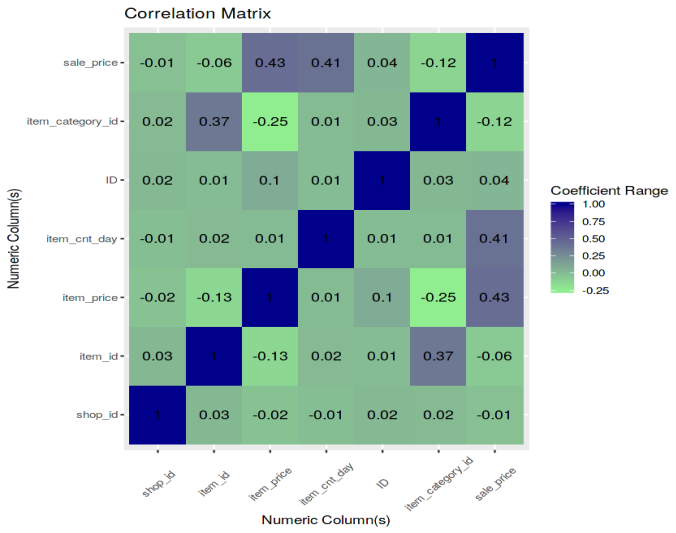

Figure 22. correlation-matrix

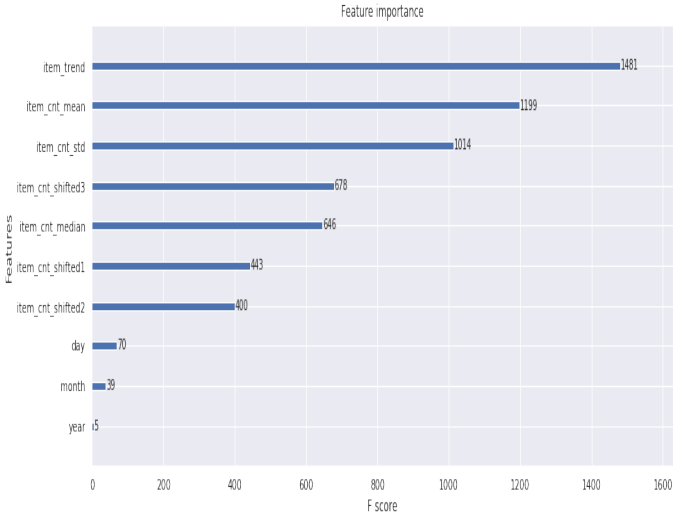

Figure 23. Importace plot in XGBoost after feature extraction

5 Model Selection

In this section we are going to cover the various models that we used while ariving at the optimal model in the context of our problem. First we took our chances with the liner regression model and got an rmse of this for the optimal features that we selected in the feature engineering part. We would call it the ideal features for our model.

Then we looked at the classical approach with XGBoost to plot the feature importance and experiment with various attributes for the models. This helped us in ordering the features based on their importance and arrive at an optimal feature set.

After which we tried Random forest and looked at various validation rmse.

The data set given to us was pretty big so it took about hours to run the random forest and obtain the result. We have split the data into test and validations but still the data was big enough for the model to take about 1-2 hrs of fitting. Finally we we ran it with the dataset it was seen to be overfitting. We tuned the hyper parameters in order to reduce the rmse but the result on the test data was pretty much the same.

Ensembling of the models would have helped us to get a good score. But we used more complex approach called the Extra-TreeRegressor and we got pretty good results after using this model.

The Extra-Tree method stands for extremely randomized trees.

The main objective is to further randomizing tree building in the context of numerical input features, where the choice of the optimal cut-point is responsible for a large proportion of the variance of the induced tree.

With respect to random forests, the method drops the idea of using bootstrap copies of the learning sample, and instead of trying to find an optimal cut-point for each one of the K randomly chosen features at each node, it selects a cut-point at random.

This idea is productive in the context of many problems characterized by a large number of numerical features varying more or less continuously: it leads often to increased accuracy thanks to its smoothing and at the same time significantly reduces computational burdens linked to the determination of optimal cut-points in standard trees and in random forests.

From a statistical point of view, dropping the bootstrapping idea leads to an advantage in terms of bias, whereas the cut-point randomization has often an excellent variance reduction effect. This method has yielded state-of-the-art results in several high-dimensional complex problems.

From a functional point of view, the Extra-Tree method produces piece-wise multilinear approximations, rather than the piece-wise constant ones of random forests.Hence we used the Extra-TreeRegressor.
Model selected is EXTRA TREE REGRESSOR.

6 Hyper Parameter Tuning

This section covers all the aspects how the hyperparameters were set to a particular value. Our model finally uses these parameters for the final estimation of the data. We look at only one case where we tuned the hyperparameters of the extra-tree regressor although we tuned the hyperparameters of the several other models that we used is because this was the Extra-Tree Regressor was the final model that got the best rmse value. Give below is the sorted order of the models that gave us the best rmse scores. A Extra-Tree Regressor has several parameters but we considered the important three they are:
1. n-estimators;
2. max-depth;
3. random-state;
each of them having their own significance.

n-estimators: Is the number of trees to be used in the forest. Since Random Forest is an ensemble method comprising of creating multiple decision trees, this parameter is used to control the number of trees to be used in the process.

max-features on the other hand, determines the maximum number of features to consider while looking for a split.

max-depth: The maximum depth of the tree. If None, then nodes are expanded until all leaves are pure or until all leaves contain less than min-samples-split samples.

random-state (int, RandomState instance or None, optional (default=None)): If int, random-state is the seed used by the random number generator; If RandomState instance, random-state is the random number generator; If None, the random number generator is the RandomState instance used by np.random.

These are all the validation test scores as you can see that the model keeps over-fitting for higher number of trees and increasing the max depth.

We have to run each model separately because of the overhead of each model consuming more that 13GB of RAM. So we could not have written them down in a for loop and let the model figure out the rest. As there would be a Memory Error and result in the termination of the entire program. So all the updates would have been lost and it would not be easy to keep track of the model results. So we used each values separately and looked at the estimates that it gave and figured the best one out.

link : https://drive.google.com/open?id=13Z8cHkRSyIE7RzGYsFAuKEVwfK4BzQOz

7 RESULTS

Used Extra Tree Regressor with 600 esimators
and depth 38 as our finale model
512 estimators max depth-23
RMSE cliped: 0.5232590552246942
RMSE private score : 1.02902
RMSE public score 0.92240

520 estimators max depth-25
RMSE cliped: 0.4117675588900876
RMSE private score :1.00078
RMSE public score 0.85306
550 estimators max depth-28
RMSE cliped: 0.254363174322717
RMSE private score : 0.98989
RMSE public score 0.80138
555 estimators max depth-32
RMSE cliped: 0.10506131290830834
RMSE private score : 0.98931
RMSE public score 0.79130
550 estimators max depth-35
RMSE cliped: 0.044216422980378145
RMSE private score : 0.98401
RMSE public score 0.79245

600 estimators max depth-38
RMSE cliped: 0.016129309930076777
RMSE private score : 0.98869
RMSE public score 0.79111
650 estimators max depth-38
RMSE cliped: 0.016169828893133543
RMSE private score : 0.98979
RMSE public score 0.79075

8 REFERENCES

kaggle kernels