Machine Learning Technical Assignment:

Predicting mobile operator sales

Uma Sankar Sahoo Jan, 2018

I. Definition Project Overview

The project is about predicting the sales for different mobile operator operating in different cities of Indonesia.The objective of this project is to predict the final sales of a mobile operator selling different products to customer given 21 explanatory variables.

The data in this project can be downloaded from the below link :

https://drive.google.com/uc?id=1bQFMYJcgFB4IUf-xixkD7oPy5KfoUn52&export=download

The dataset contains 7 xlsx file namely :

code.xlsx	: Contains all categorical to numerical mapping used in below datasets.
RO RANDOM W8-1.xlsx	: Dataset for sales in W8
RO RANDOM W8-2.xlsx	: Dataset for sales in W8
RO RANDOM W9-1.xlsx	: Dataset for sales in W9
RO RANDOM W9-2.xlsx	: Dataset for sales in W9
RO RANDOM W10-1.xlsx	: Dataset for sales in W10
RO RANDOM W10-2.xlsx	: Dataset for sales in W10

Note :

1. code.xlsx is not exactly any dataset but rather a mapping sheet that maps all categorical features to their corresponding numerical valued features used in the "RO RANDOM" datasets.

2. For the ease of convenience i have mapped the columns corresponding to all the categorical features used in code.xlsx(Originally absent in the downloaded raw file) . A refined version of code.xlsx is attached to the supporting docs which is required for analysis .

Problem Statement

The Ultimate goal of this project is to train a model based on the given data to predict the sales as accurate as possible. The related tasks to this problem include:

- Exploratory data analysis – Understanding the type of features in the dataset, how much missing values of different features, how many outliers are existed and whether specific features are skewed, are crucial in order to create a good model in the end.

- Feature preprocessing – Preprocess the data using the insights obtained from the exploratory data analysis, potentially including feature transformation, data type transformation, outlier detection and imputing missing values.

- Benchmark modeling – Creating a model using a standard technique for the problem in order to set up a benchmark for the future modeling improvement.

- Model improvement – Tuning model parameters to improve the performance of individual model and stacking individual models to provide final submission using ensemble learning technique.

Metrics

The Root Mean Squared Logarithmic Error (RMSLE) will be used to as the evaluation metrics for this project. The RMSLE can be expressed a

$$\epsilon = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(\log(p_i + 1) - \log(a_i + 1))^2}$$

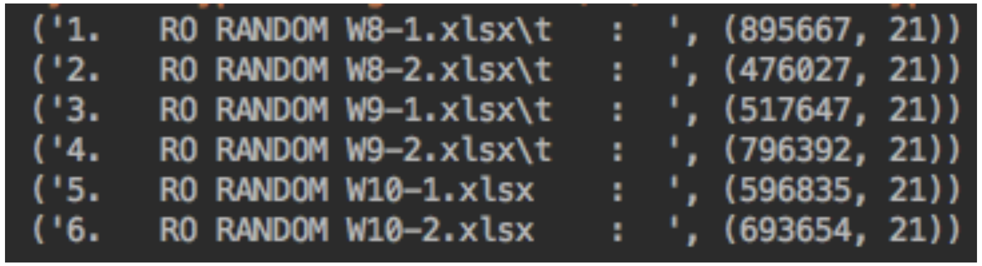
where ϵ is the RMSLE score, n is the total number of observations, p_+ is the predicted number of the product price,a_+ is the associated actual product price and $\log(x)$ is the natural logarithm of x. The value of RMSLE is higher when the differences between the predicted and actual product prices are larger. Compared to the most common Root Mean Squared Error (RMSE), RMSLE does not heavily penalize the huge difference between the predicted and actual values.

II. Analysis

Taking logs means that errors in predicting expensive products/plans and cheap products/plans will affect the result equally

Data Exploration

The training dataset is mixed with categorical and numerical features as well as some missing values. While most of them have been already mapped to their corresponding numerical equivalent we still need to remap few of them.Specifically, there are around six files with around 400k-900k observations, each with 21 features including the Total selling price of the operators. The training dataset has first 3 file observations, each with 20 features. File 4 and file 5 will serve as cross validation sets which will help us validate our trained model in a random fashion.Our task is to find the total sales price of the test dataset which is file no 6 using the model trained from the training dataset. The file size and file numbering is shown in fig 1 which will be used throughout the analysis.The histogram of the sales is shown in Figure 2, which shows that it is skewed. Thus, the log transformation needs to be performed in order to normalize it and this is the reason why we need to use the RMSLE as our evaluation metric.



```
('1.   RO RANDOM W8-1.xlsx\t   :  ', (895667, 21))
('2.   RO RANDOM W8-2.xlsx\t   :  ', (476027, 21))
('3.   RO RANDOM W9-1.xlsx\t   :  ', (517647, 21))
('4.   RO RANDOM W9-2.xlsx\t   :  ', (796392, 21))
('5.   RO RANDOM W10-1.xlsx    :  ', (596835, 21))
('6.   RO RANDOM W10-2.xlsx    :  ', (693654, 21))
```

Figure 1:Input files and their size

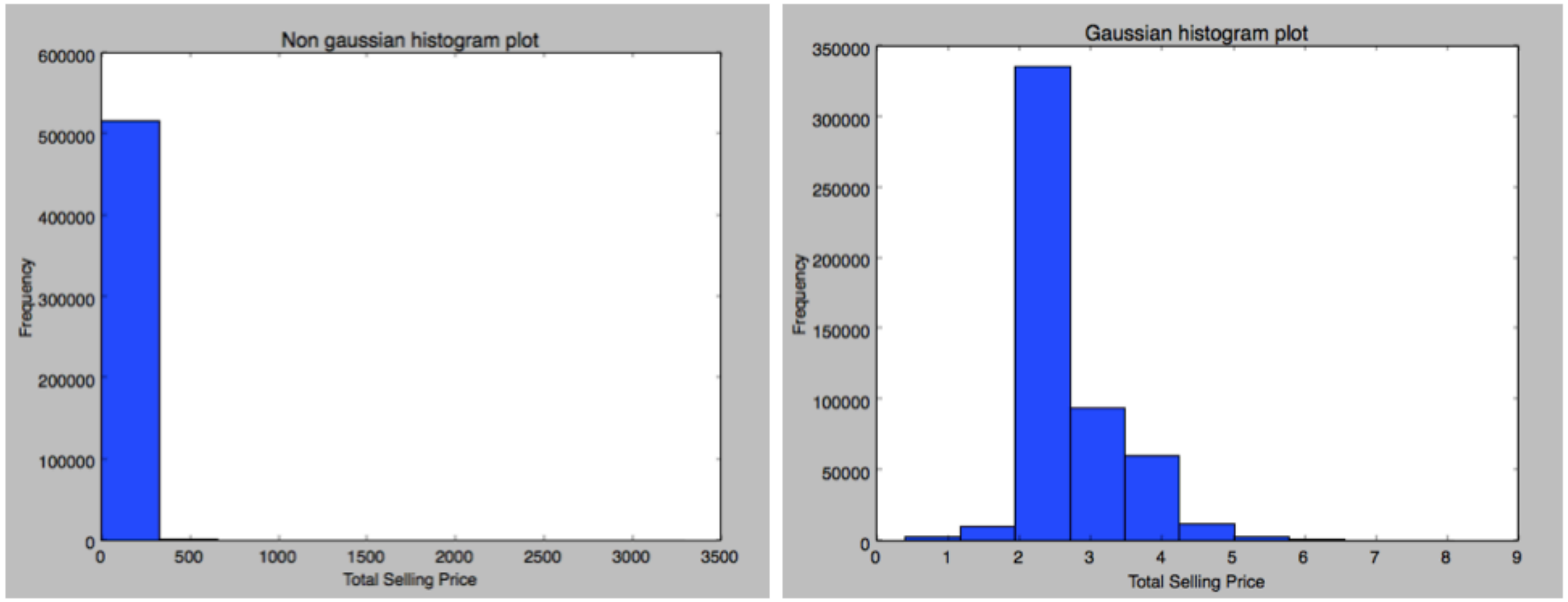
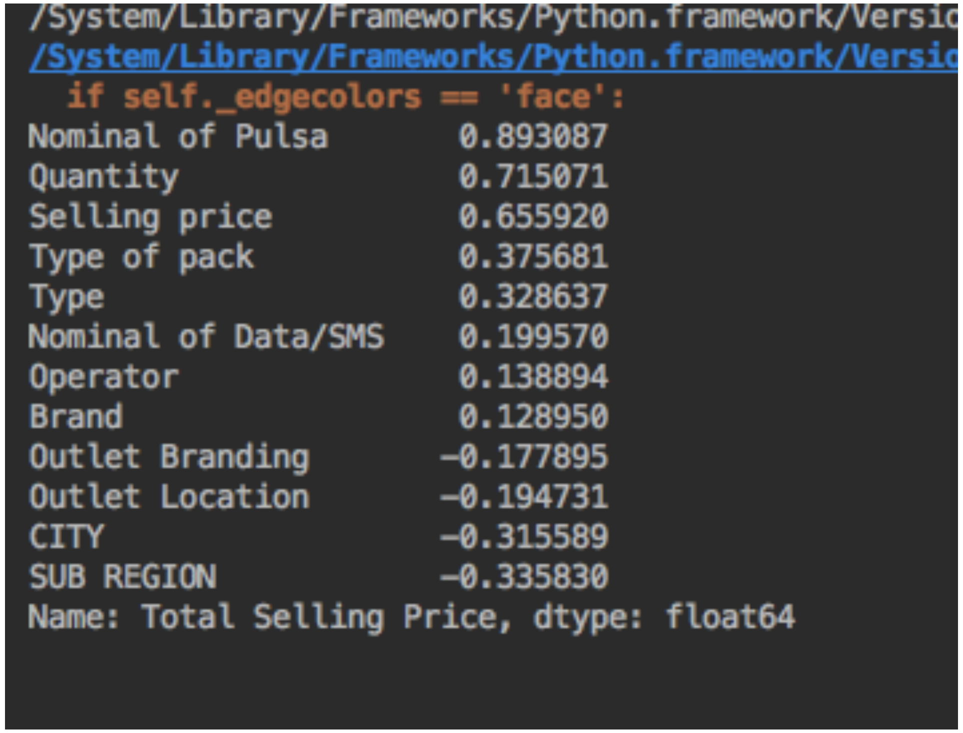


Figure 2: The histograms of the operators sale price. (left) Original data; (right) Log transformed data.

Exploratory Visualization

Figure 3 shows the correlation matrix between the numerical variables and the operators selling price. There are several variables that are strongly correlated with the sales shown in Figure 3. I also explore the top variables that are highly correlated with the sale price in Figure 4.



```
/System/Library/Frameworks/Python.framework/Versio
/System/Library/Frameworks/Python.framework/Versio
  if self._edgecolors == 'face':
Nominal of Pulsa        0.893087
Quantity                0.715071
Selling price           0.655920
Type of pack            0.375681
Type                    0.328637
Nominal of Data/SMS     0.199570
Operator                0.138894
Brand                   0.128950
Outlet Branding        -0.177895
Outlet Location        -0.194731
CITY                   -0.315589
SUB REGION             -0.335830
Name: Total Selling Price, dtype: float64
```

Figure 3: The correlation coefficients between some variables and the sale price

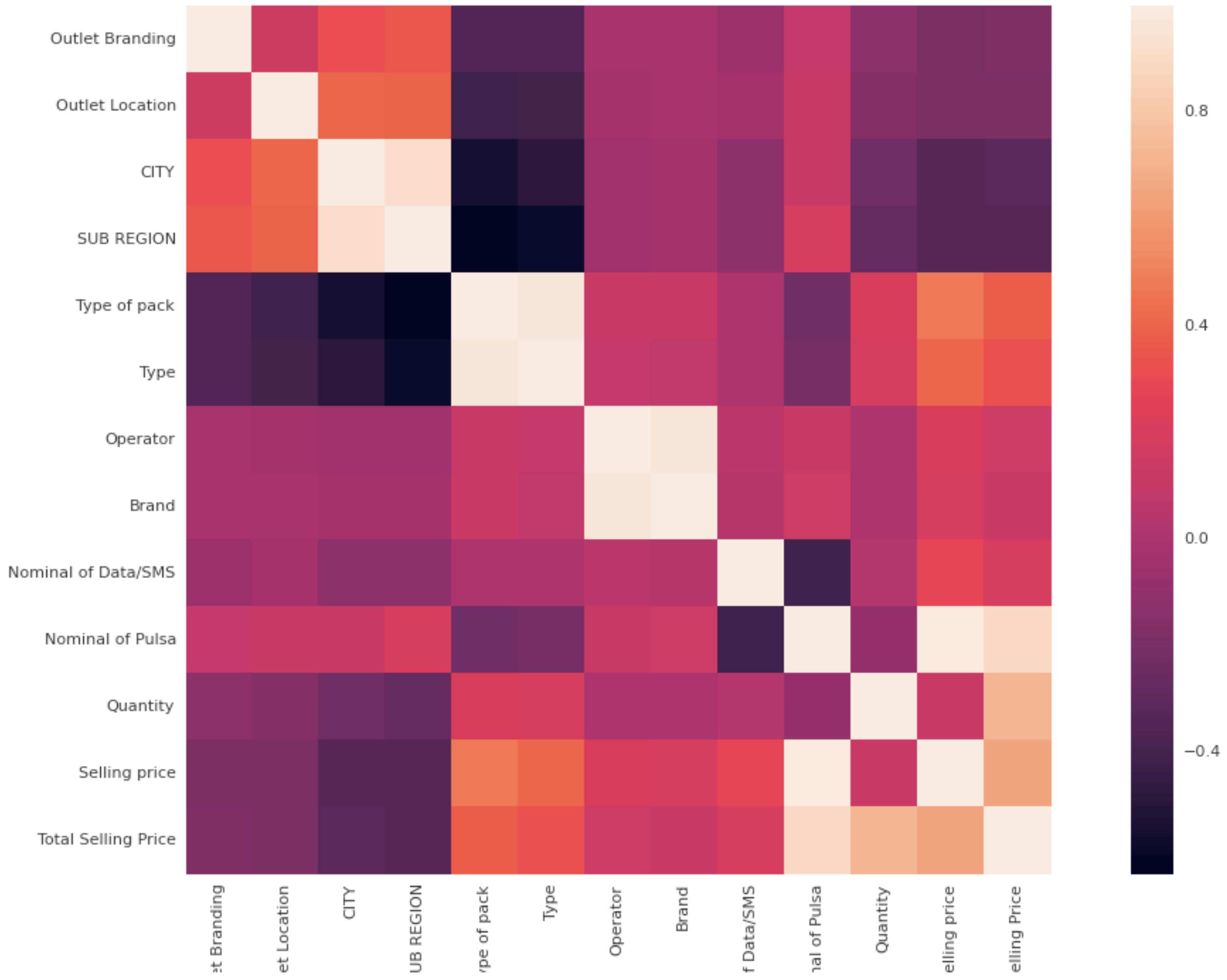


Figure 4: The correlation matrix between different numerical variables.

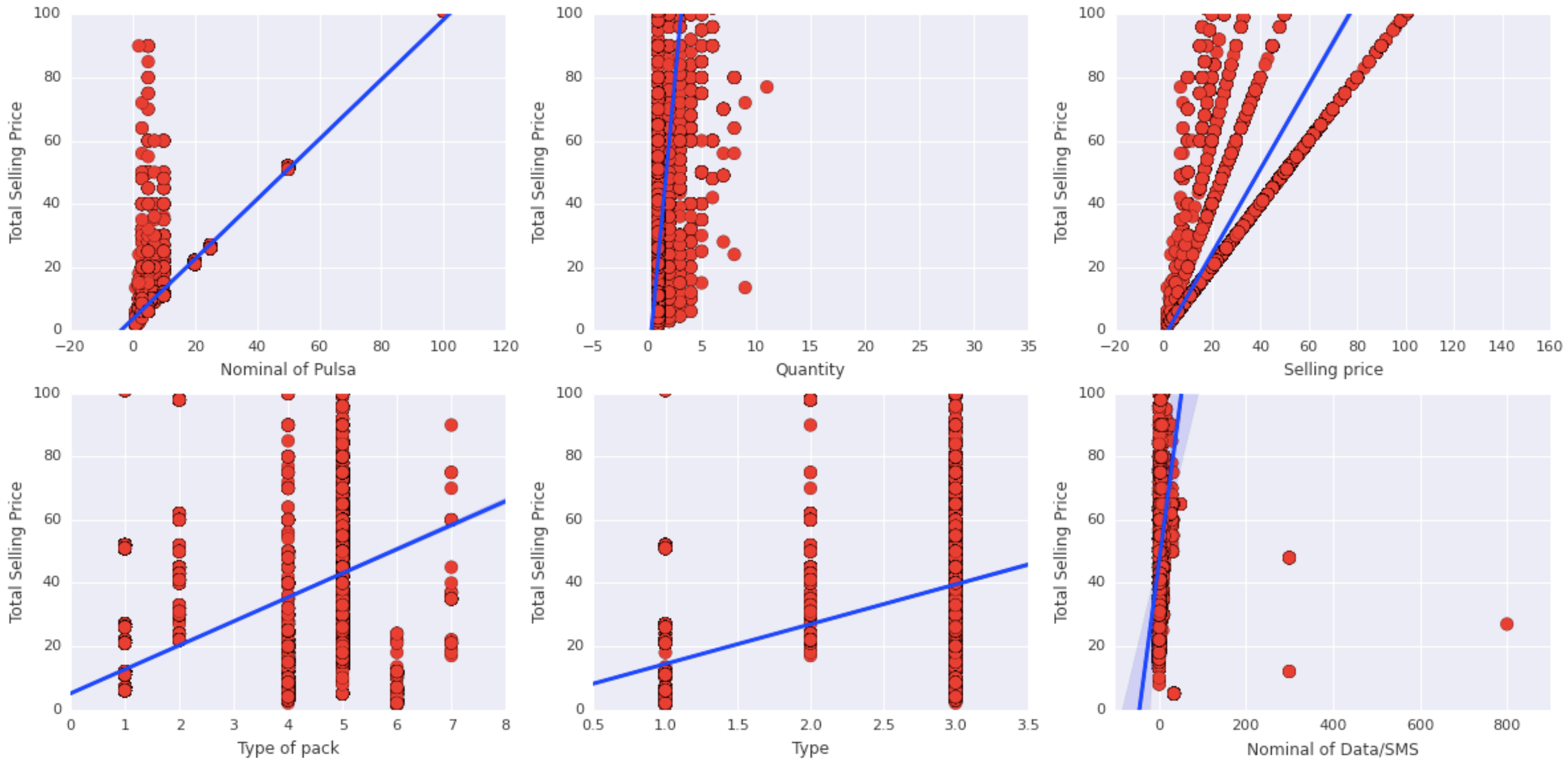


Figure 5: The scatter plot of the top 6 strongly correlated variables with the sales. The regression curve is shown in blue. Gives inference of linearity for few features.

To give an example of the distribution of feature variable, the boxplot between the all the correlated variables is shown in Figure 6.

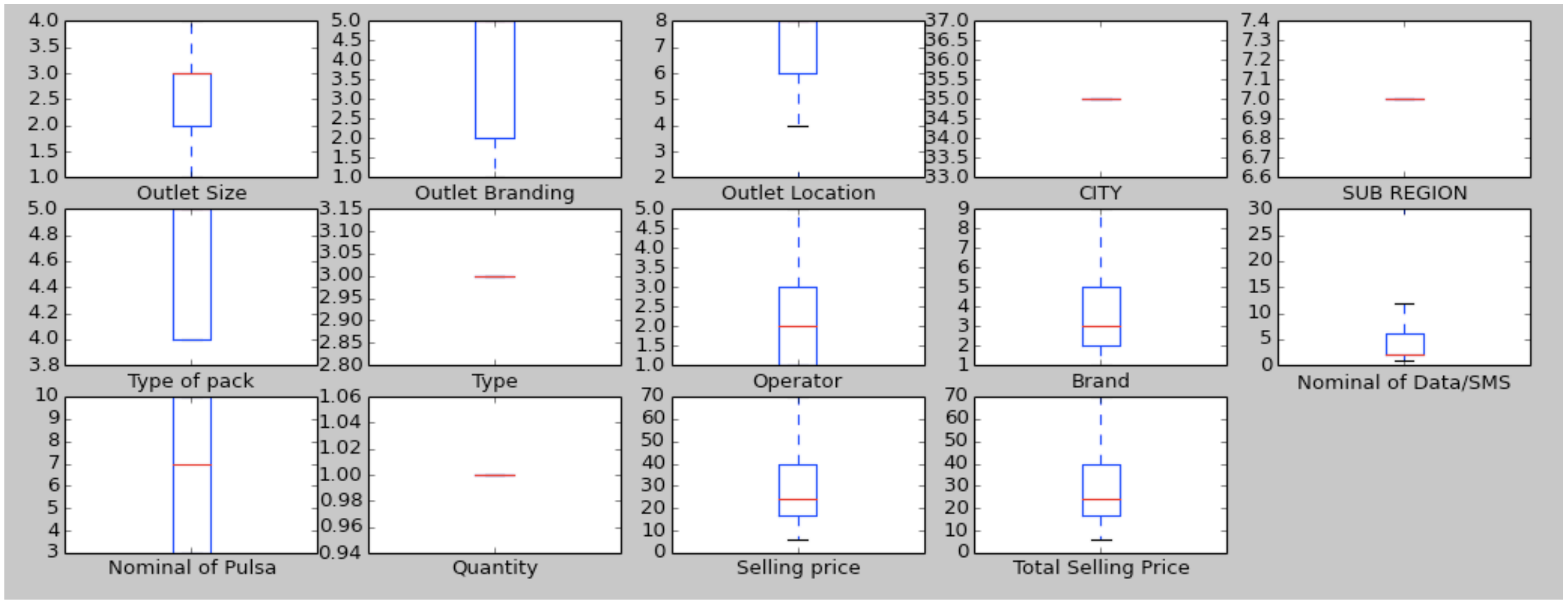


Figure 6: Boxplot of features. (Omitted the outliers like the ID columns and date columns)

Algorithms and Techniques

Given one of the purposes of this project is to explore the best model, I am going to use the ensemble learning technique to achieve better RMSLE score. The ensemble learning techniques aim to learn a weighted combination of the base models called a committee method. The technique I implemented in this project is called stacking. Stacking, which stands for “stacked generalization”.

The basic idea of stacking is to use another model or “stacker” to combine all previous model predictions in order to reduce the generalization error. An illustration of a 2 level 5 folds stacking approach is shown in Figure 7. First, the training dataset need to be split into 5 folds. Second, we iterate over this 5 folds training dataset. In each iteration, each base model will be trained using 4 folds and predict on the hold out fold. At the same time, each base model also need to provide a prediction on the entire test dataset. After the iteration over all folds, we will have the prediction of the entire training dataset for each model and 5 copies of the prediction of the entire test dataset for each model. Finally, we train second level model, or stacker, using the prediction in the training dataset as new features and use the average of the 5 copies of the test dataset predictions as the test input for the trained model to provide the final prediction.

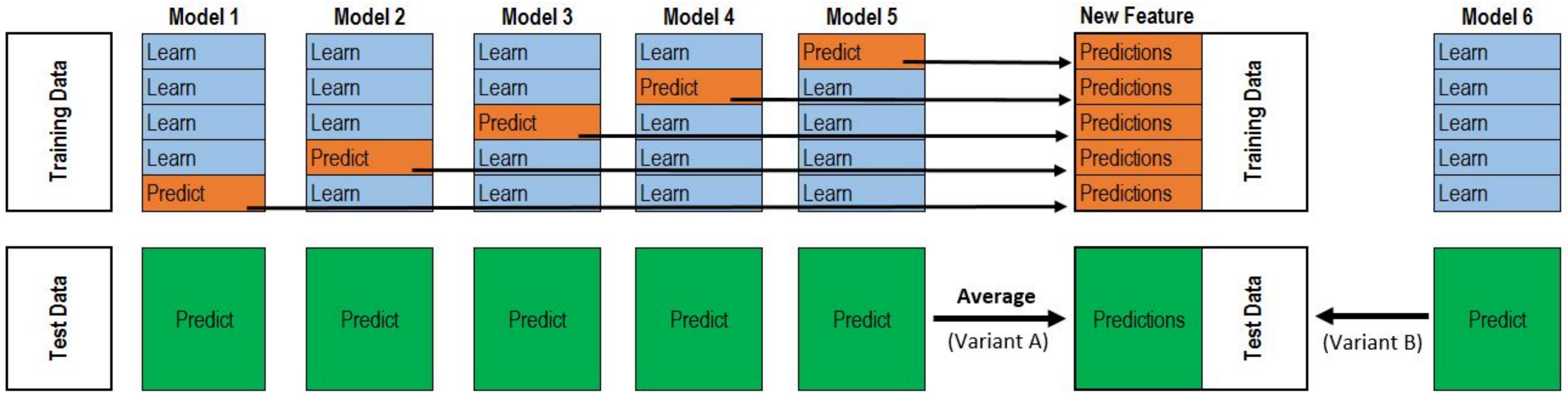


Figure 7: An illustration of stacking approach. (Credit: Google Images)

Benchmark

The Random Forest regression is selected as our benchmark model for this project. The default parameters in scikit-learn package will be used as the benchmark, which earns a cross validation score of 0.0136097398396.

III. Methodology

The overall methodology and modeling approach can be summarized using the flowchart shown in Figure 8. The details of the approach will be discussed in the following subsections.

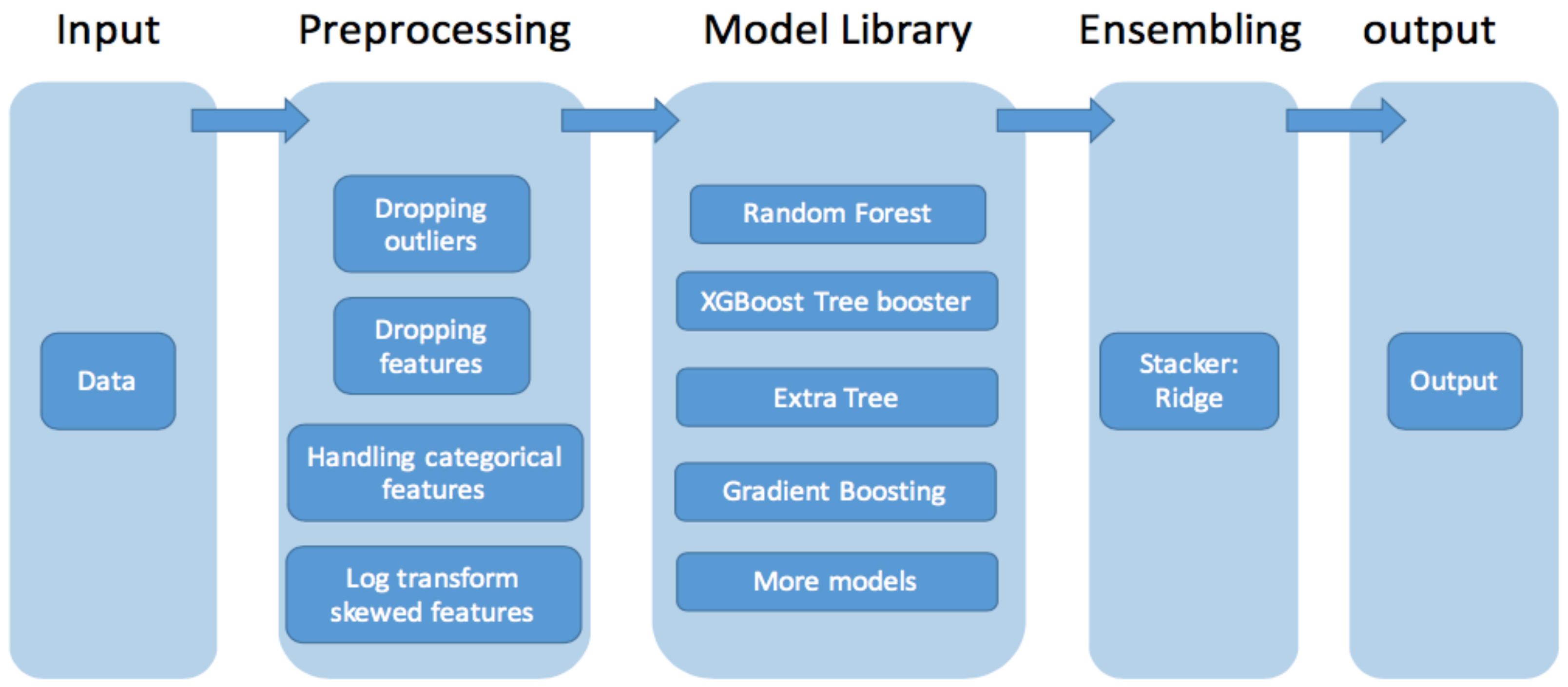


Figure 8: Flowchart of the modeling approach.

Data Preprocessing

- Outliers deletion

Any predictions from the benchmark that differ more than 5000 (e.g considering the max value to be 3293 for file 3 RO RANDOM W9-1.xlsx) than the actual sales are labeled as outliers. The threshold selected here is somewhat arbitrary but this approach should impact the final result set slightly. All outliers are deleted from the training dataset for the final submission if any.

• Feature selection

The feature selection approach I adopted here is a very simple and naïve one: delete features with high number of missing values. The variables with the largest number of missing values are Nominal of Data/SMS and Type of Data/SMS.However the final CV score is not greatly affected so i have considered them for final model replacing the missing values with 0.(CV score of 0.000154 with Nominal of Data/SMS Vs CV score of 0.000157 without Nominal of Data/SMS) In the data exploration analysis, there are some features that are highly correlated with the total sale price. We may treat some of these features as redundant variables and delete them from the training dataset. However, previous investigation shows that this will leads to a small decrease in scores.

- Log transform skewed features
  All features that have a skewness larger than 0.75 are log transformed.

- Handle categorical features

Almost all categorical features are already mapped to their numerical equivalent prior to any handling exceptions. Still i have handled by the pandas get_dummies function that converts categorical variables into dummy or indicator variables considering the scaled model.

Implementation

Before applying the ensemble learning method, I trained and tuned 4 base model including Random Forest regressor, Extra Trees regressor, Gradient Boosting regressor, Extreme Gradient Boosting regressor.

At the beginning of the ensembling implementation, I use only 4 base models to form my model library, including Random Forest regressor, Extra Trees regressor, Gradient Boosting regressor, Extreme Gradient Boosting regressor. And I use the Ridge Regression as my second level model. The idea of the stacking approach is to use the predictions from the first level model as the new features for the second level model and make prediction based on that.

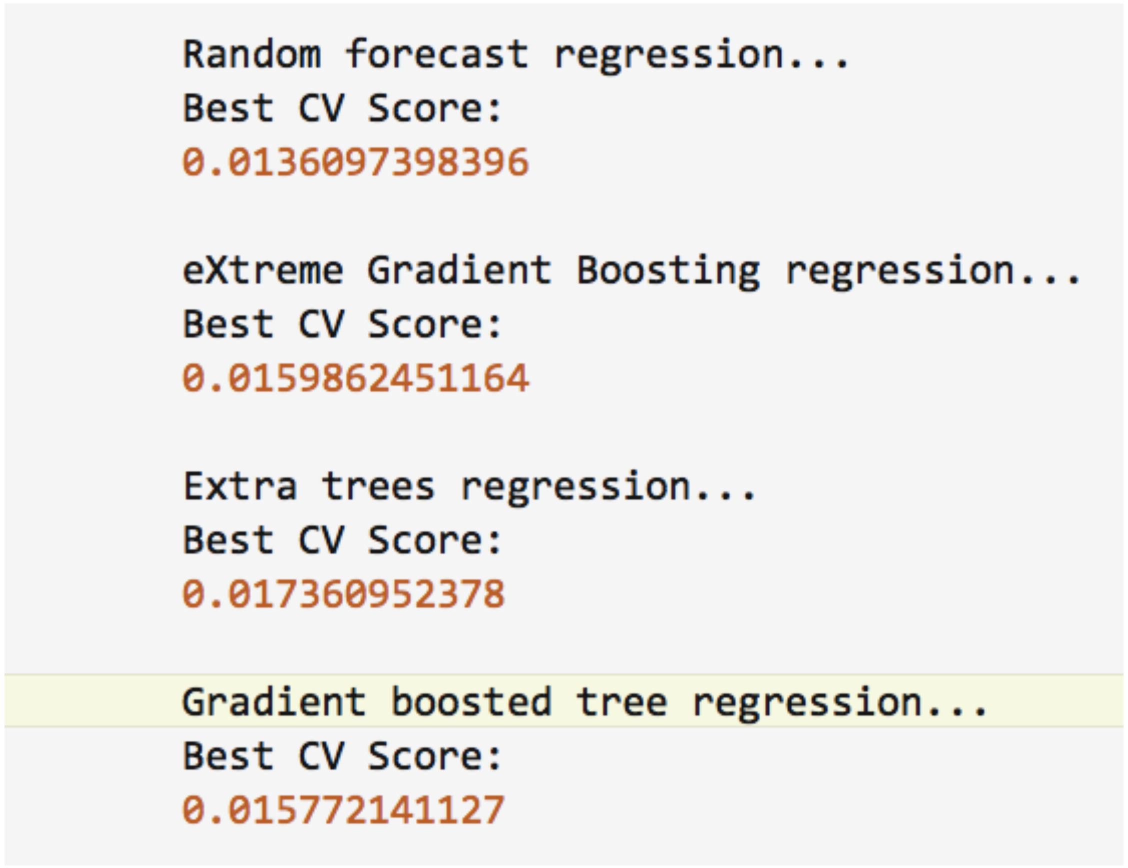
The base models should be as unrelated as possible and this is why we tend to include more models in the ensemble even though they may not perform well. For my final, I use a much larger much library including Random Forest regressor, Extra Trees regressor, Gradient Boosting regressor, Extreme Gradient Boosting regressor, LassoCV regressor, K neighbors regressor, LassoLarsCV regressor, Elastic Net regressor, Support Vector Machine regressor. The Extreme Gradient Boosting regressor is from the XGBoost package, all other techniques are from scikit- learn package. The basic idea of the final model is to use a larger base model library in order to achieve better results. Some of the base models are used only with their scikit-learn default parameter values without fine tuning.

There are two challenging parts during the coding process. The first one is that we need to make sure the feature for the final second level prediction is based on the prediction of the first level base models. Thus, we also need to use the model prediction as the features for the test dataset. The second one is how exactly should we choose the base models. This is a tricky and not fully addressed question in this project since there are so many models to select. But the rule of thumb in my investigation is that I choose the one that improve the cross validation score.

Refinement

The refinement is conducted using the grid search technique. For the learning purpose of this project, I only use a few tunable parameters shown in the following (the detailed grid range can be found in the submitted code). Figure 8 and Figure 9 show the cross validation score before and after the grid search.

Before the parameter tuning, I just use the scikit-learn default parameter values for each model. After the refinement of the base models, we using stacking technique to ensembling them together to provide the final submission and the final model best parameter was used to produce the final result.



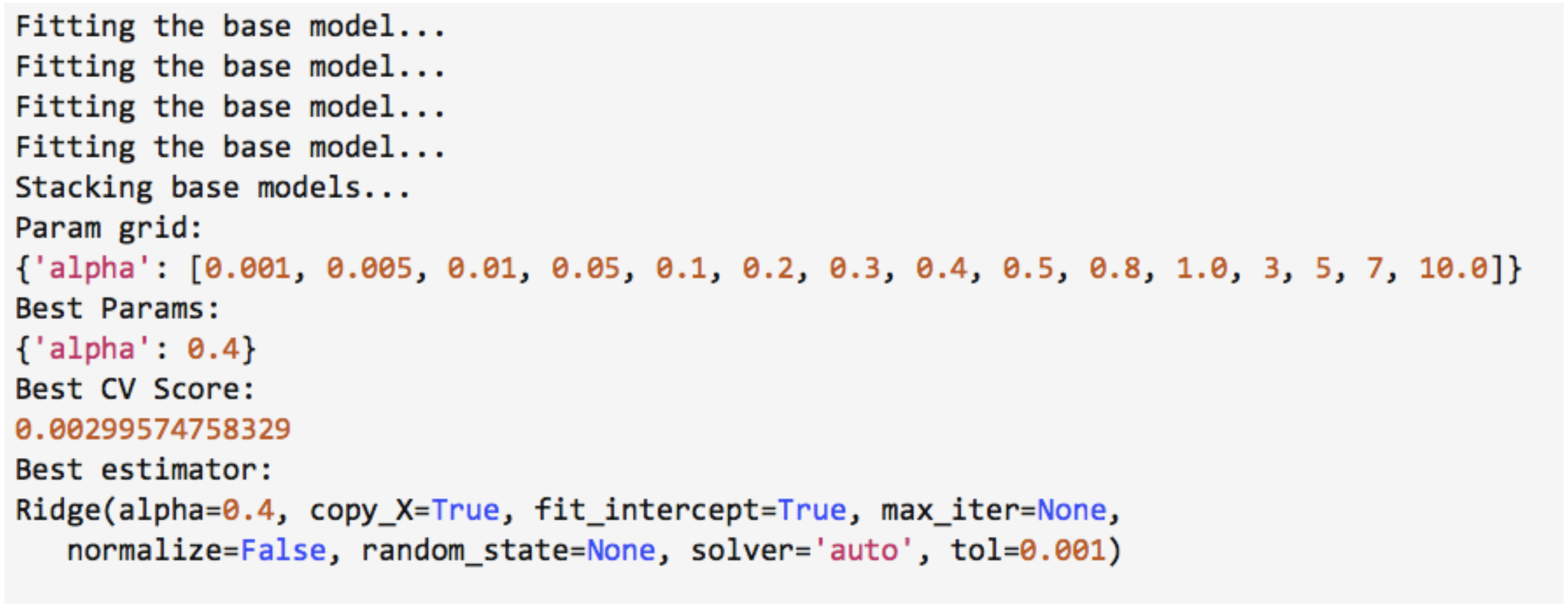
```
Random forecast regression...
Best CV Score:
0.0136097398396

eXtreme Gradient Boosting regression...
Best CV Score:
0.0159862451164

Extra trees regression...
Best CV Score:
0.017360952378

Gradient boosted tree regression...
Best CV Score:
0.015772141127
```

Figure 9: The cross validation score with default scikit-learn parameter values.



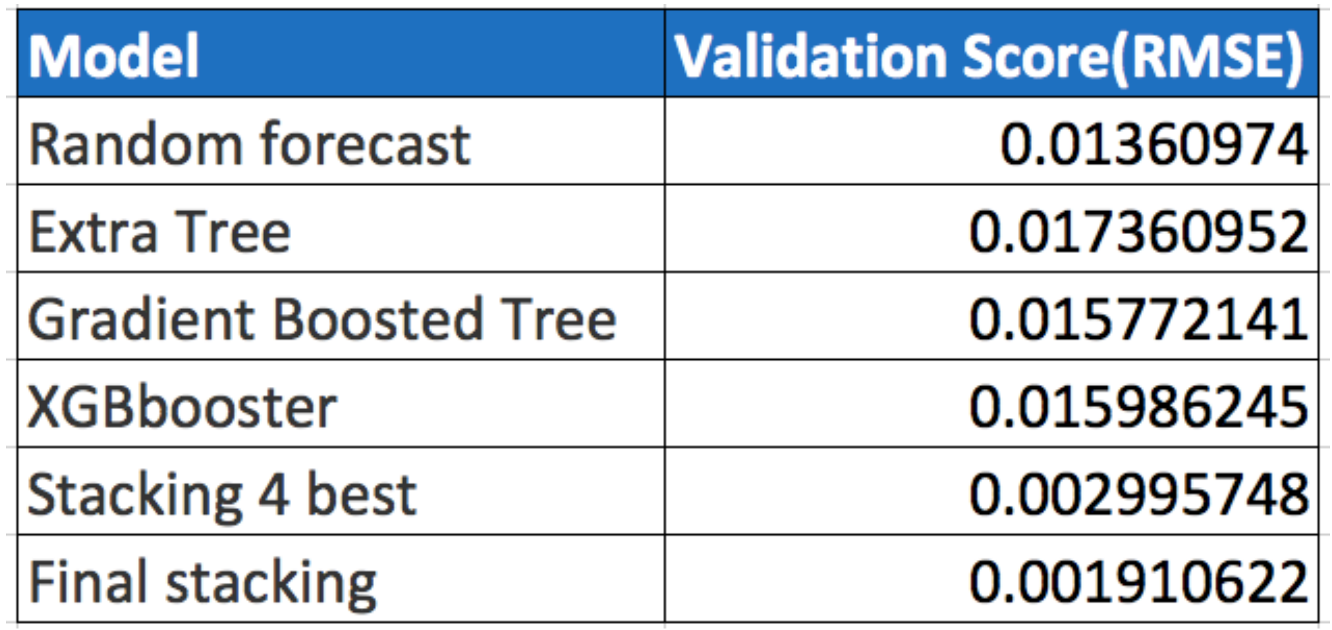
```
Fitting the base model...
Fitting the base model...
Fitting the base model...
Fitting the base model...
Stacking base models...
Param grid:
{'alpha': [0.001, 0.005, 0.01, 0.05, 0.1, 0.2, 0.3, 0.4, 0.5, 0.8, 1.0, 3, 5, 7, 10.0]}
Best Params:
{'alpha': 0.4}
Best CV Score:
0.00299574758329
Best estimator:
Ridge(alpha=0.4, copy_X=True, fit_intercept=True, max_iter=None,
   normalize=False, random_state=None, solver='auto', tol=0.001)
```

Figure 10 : The cross validation score for ensemble model after the grid search refinement.

IV. Results

Model Evaluation and Validation

One interesting observation during the model ensembling process is that increasing the size of the model library can potentially increase the cross validation score . In the final submission, a variety of base models are included in the model library with their scikit-learn default parameter values. Using a much larger library enables me achieve better results . The approach i adopted is somewhat brute force and a more elegant way to do this is to find the base models to be as uncorrelated as possible.



Model	Validation Score(RMSE)
Random forecast	0.01360974
Extra Tree	0.017360952
Gradient Boosted Tree	0.015772141
XGBbooster	0.015986245
Stacking 4 best	0.002995748
Final stacking	0.001910622

Table 1 : Models and their CV score consolidated

V. Conclusion

In summary, the ensemble approach in this project greatly improve the performance of the model.There were many challenges which i faced are listed below :

1. The input dataset size was huge covering my system configuration

2. Many a time system responds too slow while i was training the model.In many case subset of training sequence from selected input files are used for training purpose.

3. Hyper parameter tuning is extensively used which was sometimes hard to find the optimum value.Tried my best to fit the model with different combination and their certainly are chances that the system can still be improved.

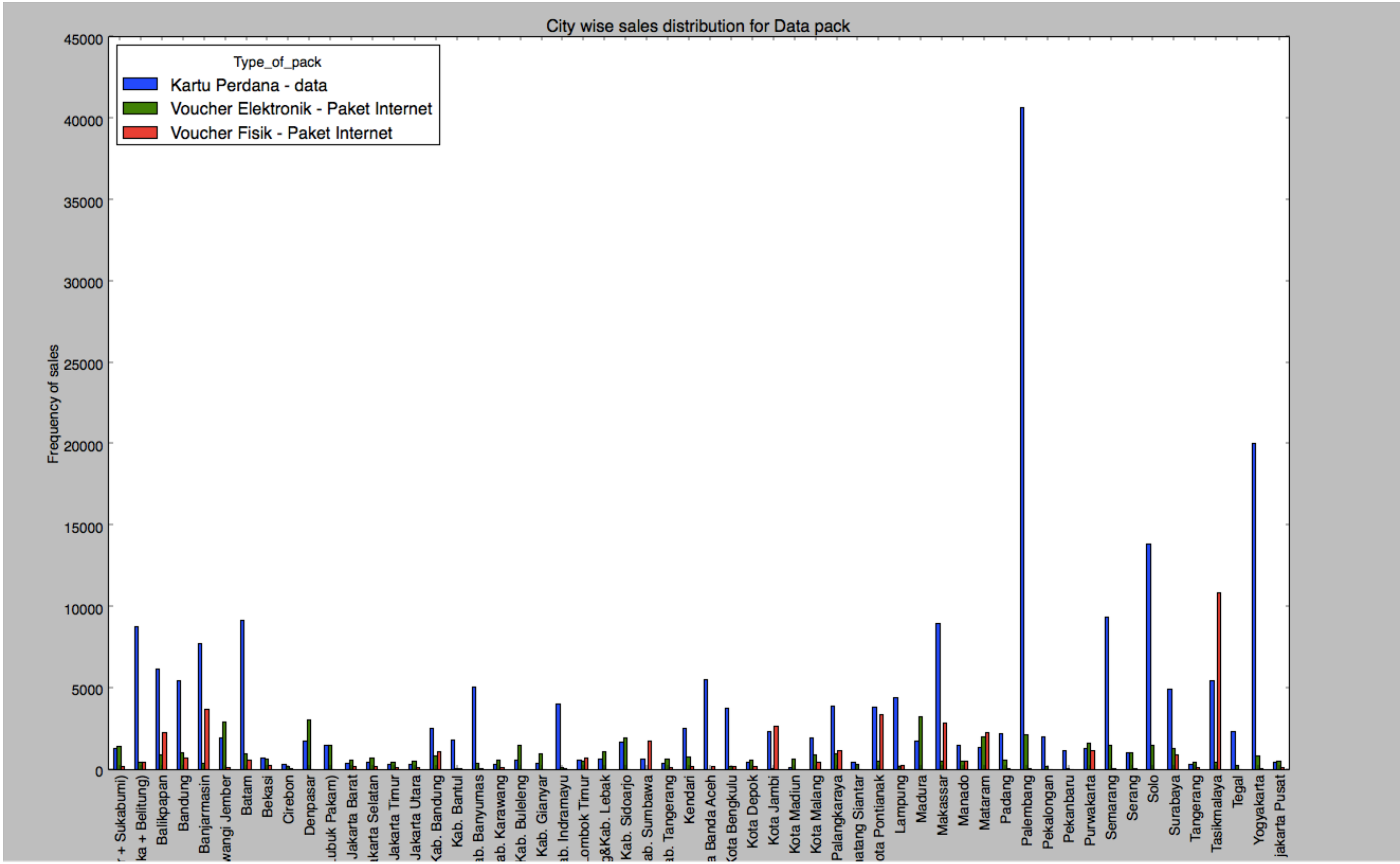
Reflection

For this project, I started with an exploratory data analysis, in which I found that there are some features being skewed, some features are categorical, and some missing values. Several data preprocessing steps were conducted in order to make the data ready for the machine learning modeling part. For the modeling part, I firstly trained several base models and tuned their parameters. Then, I applied the ensemble learning method to stack all the base models together to provide the final submission. One of difficult parts is data preprocessing since it is hard to explain in a rigorous way of why I did a specific choice to improve the model. There are a lot of exploratory tests and investigations to justify the choice only based on the cross validation scores. The other challenge is the stacking approach itself since it is more complicated than a single model approach.

City Wise sales prediction of data pack :

Steps :

1. Combining all the data sets
2. Dropping outliers and rows with either city or the of pack having NaN values
3. Select plans specific to data
4. Numerical to categorical mapping value mapping for city and The of packs
5. Plotting . Fig shown below


City wise sales distribution for Data pack
Type_of_pack
Kartu Perdana - data
Voucher Elektronik - Paket Internet
Voucher Fisik - Paket Internet
Frequency of sales
45000
40000
35000
30000
25000
20000
15000
10000
5000
0
r + Sukabumi)
ka + Belitung)
Balikpapan
Bandung
Banjarmasin
wangi Jember
Batam
Bekasi
Cirebon
Denpasar
ubuk Pakam)
Jakarta Barat
akarta Selatan
Jakarta Timur
Jakarta Utara
Kab. Bandung
Kab. Bantul
ab. Banyumas
ab. Karawang
Kab. Buleleng
Kab. Gianyar
ab. Indramayu
ombok Timur
g&Kab. Lebak
Kab. Sidoarjo
ab. Sumbawa
ab. Tangerang
Kendari
a Banda Aceh
Kota Bengkulu
Kota Depok
Kota Jambi
Kota Madiun
Kota Malang
Palangkaraya
atang Siantar
ota Pontianak
Lampung
Madura
Makassar
Manado
Mataram
Padang
Palembang
Pekalongan
Pekanbaru
Purwakarta
Semarang
Serang
Solo
Surabaya
Tangerang
Tasikmalaya
Tegal
Yogyakarta
jakarta Pusat

Conclusion :

1. City Palembang is most likely to excel at selling “Kartu Perdana - data” plan in future.
2. Yogyakarta has potential to sell “Kartu Perdana - data” plan.
3. Overall cities like Makassar,Semarang,Solo,Batam ,Babel (Bangka + Belitung) and few others have potentiality to sell data packs as compared to the rest.