NYU SHANGHAI
上海纽约大学

MACHINE LEARNING PROJECT

FINAL REPORT - FALL 2022

Store Sales Forecasting for Favorita

Xingji Jax Li

supervised by
Mathieu Laurière

1 Abstract

In this project, I mainly tried to apply several kinds of machine learning methods (regressions, decision trees) for the task of predicting sales for Favorita, a big company base in Ecuador that sells all kinds of daily commodities.

2 Introduction

Time series forecasting can be one of the most long-lasting topic in data science and machine learning fields which has been addressed even long before computer is invented. Although sales prediction seems not as fancy and exciting as heated topics like NLP or CV, it is still an application with high demand in the real world. And the abundant methods accumulated through out the history enable me to practice different areas of machine learning, which I believe can be more helpful in the sense of applying skills I just learnt from this course to the project than plugging in complicated models online that I do not understand at all and do some fancy applications.

3 Problem Description and Exploratory Data Analysis(EDA)

The project is based on a Kaggle competition (Click This) in which the ultimate goal is to forecast well the sales for date range from 2017-08-16 to 2017-08-31, 15 days after the provided training data with date range from 2013-01-01 to 2017-08-15. However, in this project I would more address building and comparing the specialties of different kinds of models rather than just focusing on the performance and spend all my energy in turning hyperparameters.

Following parts are the data descriptions and EDA[1] result interpretations.

3.1 The Data

The data and some background information are provided in the competition link above, I mainly used 4 data files in this project. First of all, the sales records data (train.csv), which contains the sales history in granularity of date * store * product faimly * sales. Secondly, store information data (stores.csv), which provides more information for the stores including the location, type, and cluster(a grouping of similar stores). Moreover, a data(holidays events.csv) that records the local holidays' date and name as intuitively people would be more free and willing to go shopping on holidays. Lastly, the oil price data(oil.csv) that stores the daily oil price, as Ecuador is an oil-dependent country and it's economical health is highly vulnerable to shocks in oil prices, oil price may have visible effects on sales.

3.2 EDA : sales and oil price correlation

As Ecuador is an oil-dependent country and it's economical health is highly vulnerable to shocks in oil prices, it is possible that sales for daily products could be correlated to the oil price. In this EDA, I roughly checked the pattern in oil price and aggregated sales. The main result is shown in the graph blow.

[1] refer to the EDA notebooks in DA folder

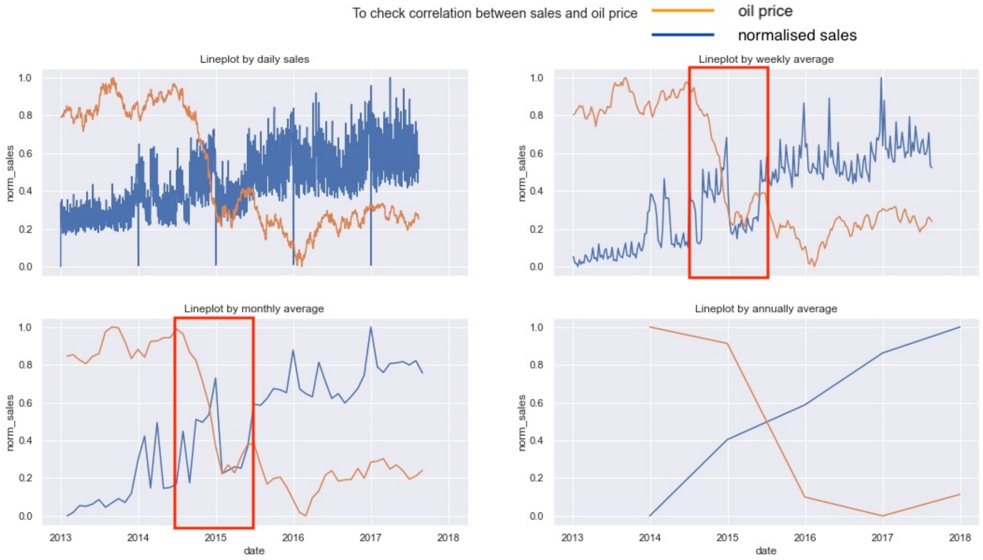


Figure 1: Normalized Oil Price and Sales Aggregated by Different Time Periods

From the lineplots, we can tell that there might be a negative correlation between the oil price and total sales in a long (yearly) scope, while there is no clear pattern within a relatively short period scopes. Interesting, these two lines align with a big drop in around 2015, which might be reflection to economic crisis in the country. Thus, it could be worth it to put the oil price as a feature in the next modeling steps.

3.3 EDA : sales clustered by store features

Despite the sales records data, the store information data seem to be juicy as from my experience different kinds of store are possible to have clearly different sales patterns. In this EDA, I tried to find out possible clusters of sales by the provided store features (city, state, store type, store 'cluster'). The following graph shows the visible pattern of sales clustered by store type.

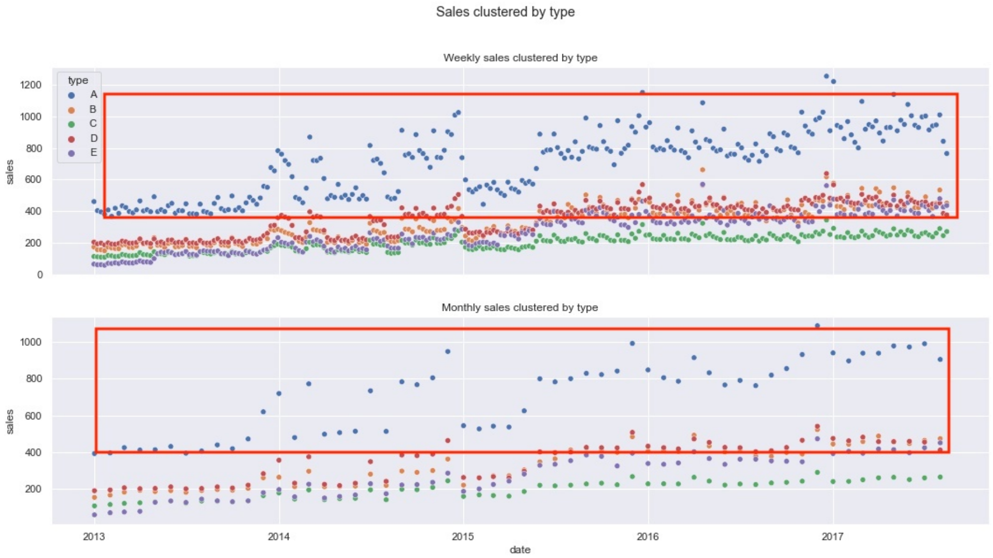


Figure 2: Average Aggregated Sales Scatter Plot Hued by Store Type

From the graph, it is easy to tell that among the 5 (A - E) types of stores the sales for A type is clearly clustered form the others while the other types seem to be more blended. Thus, it could be practical to build individual models for just A type although sales trends among different kind stores seem to be similar.

Besides the store type clusters, I did tried with a few other possible categorical features[2] though those results are not significant enough for building possible sub-models.

3.4 EDA : sales in holidays

Another additional data that can be useful is the local holidays and events data. Intuitively people would purchase more on holidays especially for those to celebrate festivals or events. The idea to this EDA is to check if the sales are lifted by holidays overall, or just by a few particular ones, or even show no significant effects.

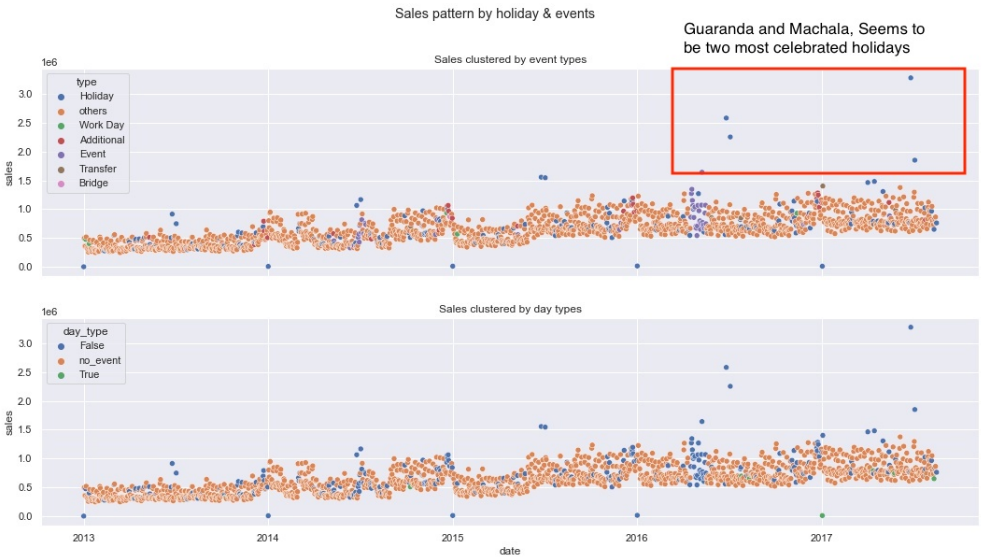


Figure 3: Daily Aggregated Sales Scatter Plot Hued by Holidays

From the graph,firstly we can tell that we need to deal with the 0-sale bug caused probably by yearly refreshing records in the later feature engineering part. More interestingly, we can find significant lifted outliers in around June and July every year. After precisely checking, these are two holidays lie on June 23rd and June 25th respectively. While other holidays do not have visible effect with respect to normal days, it can be a more reasonable and complexity-reducing idea to put just the two outstanding ones as holidays in a dummy variable than inserting the whole set of holiday data into the model later.

4 Methodology

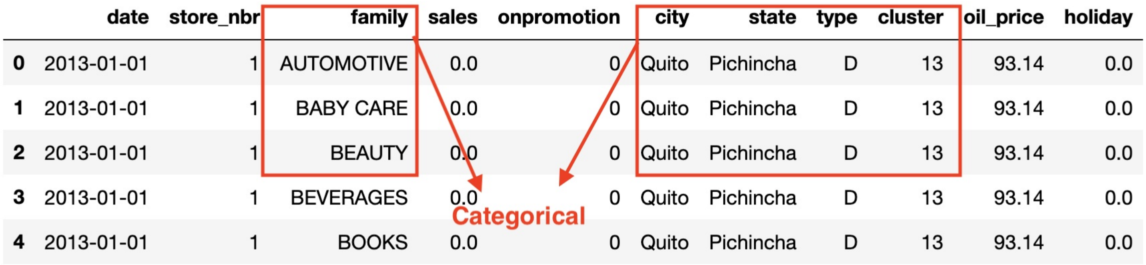
4.1 Feature Engineering

Based on the conclusions from above EDA, I conducted feature engineering by combining raw data files to produce files for training and validation[3].

[2] refer to the graphs in DA folder for more cluster plots

[3] refer to the feature engineering.ipynb in modeling folder for more code

After cleaning combining the raw data files and leave the columns that I would use in following steps. I realized that the features are consisted with continuous and categorical ones which would fit models like decision tree well but could be problematic for regressions and neuron networks.

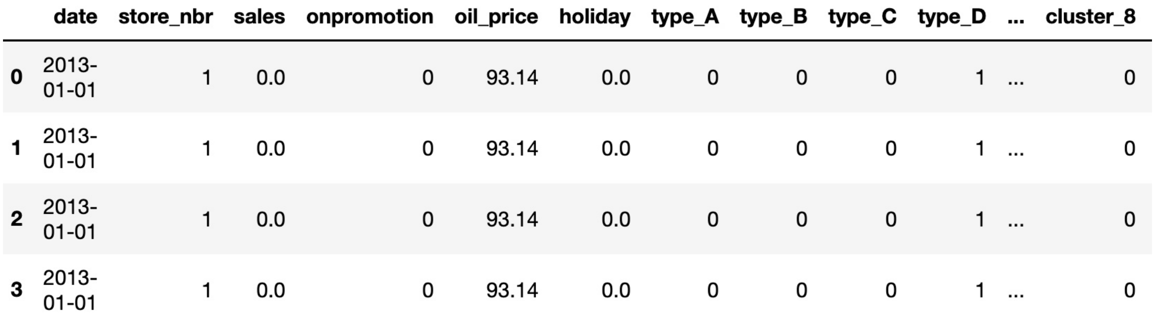


	date	store_nbr	family	sales	onpromotion	city	state	type	cluster	oil_price	holiday
0	2013-01-01	1	AUTOMOTIVE	0.0	0	Quito	Pichincha	D	13	93.14	0.0
1	2013-01-01	1	BABY CARE	0.0	0	Quito	Pichincha	D	13	93.14	0.0
2	2013-01-01	1	BEAUTY	0.0	0	Quito	Pichincha	D	13	93.14	0.0
3	2013-01-01	1	BEVERAGES	0.0	0	Quito	Pichincha	D	13	93.14	0.0
4	2013-01-01	1	BOOKS	0.0	0	Quito	Pichincha	D	13	93.14	0.0

Categorical

Figure 4: Head of Combined Data

The problem in these categorical feature, for example, would occur when we represent cities by numbers like 1,2,3... in the model it may be interpreted as city NO.2 is mathematically twice of city NO.1, which is apparently wrong. To deal with this problem, I choose to use the one hot encoding method, which would transform a categorical feature to a series of dummy indicator variables.



	date	store_nbr	sales	onpromotion	oil_price	holiday	type_A	type_B	type_C	type_D	...	cluster_8
0	2013-01-01	1	0.0	0	93.14	0.0	0	0	0	1	...	0
1	2013-01-01	1	0.0	0	93.14	0.0	0	0	0	1	...	0
2	2013-01-01	1	0.0	0	93.14	0.0	0	0	0	1	...	0
3	2013-01-01	1	0.0	0	93.14	0.0	0	0	0	1	...	0

Figure 5: Head of Combined Data After One Hot Encoding

By this encoding, we can make the model treat each category evenly. The trade-off however, is the growing amount of features (in this case from 9 to 90+) which could increase the model complexity.

4.2 Cutoffs

The idea of making cutoffs is from the cross-validation[4]. As intuitively the model is not likely to have robust and stable performance if we simply train the model with data from 2013-01-01 to 2017-07-31 and do validation test for 2017-08-01 to 2017-08-15. Inspired by the idea of cross-validation, "cutoff validation" is shown below.

[4] refer to the cutoffs.ipynb in modeling folder for more code

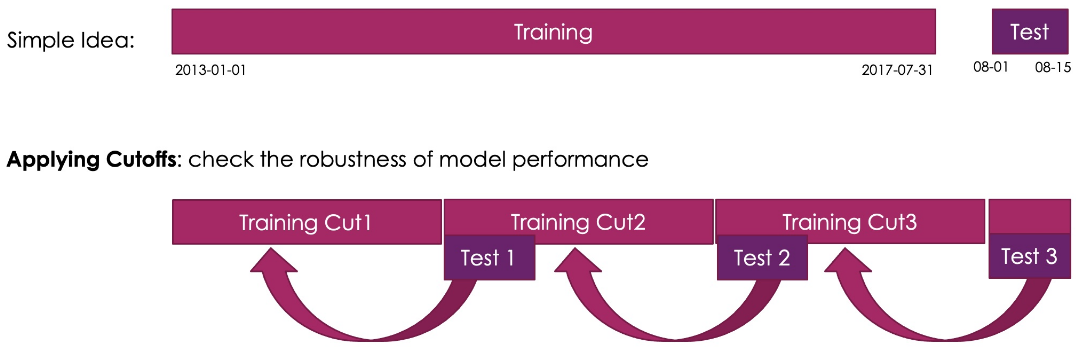


Figure 6: Illustration of the Cutoff Idea

With cutoffs, we can see how the performances of models in a more consistent way. Moreover, it can be possible to have a better performance by tuning the hyperparameter of training time range (learning from past 3 months can be better than learning from past 3 years for the target of predicting next 15 days).

For simplicity and alignment among models, in this project I just use the data with each cutoff training size of 90 days(roughly 3 months) and testing size of 15 days, and cutoff moving stride of 30 days (roughly 1 month)

4.3 Regressions

For the first section of models, my initial idea was to try linear and polynomial regressions. However, I have normalized the data by one hot encoding which making the number of features increased a lot. In such case, polynomial regression may not be a good decision as polymonially feature engineering would increase the feature amount event further.

So instead, I tried linear, ridge, and LASSO regression. Here is the MSE plot with cutoffs for this three model[5].

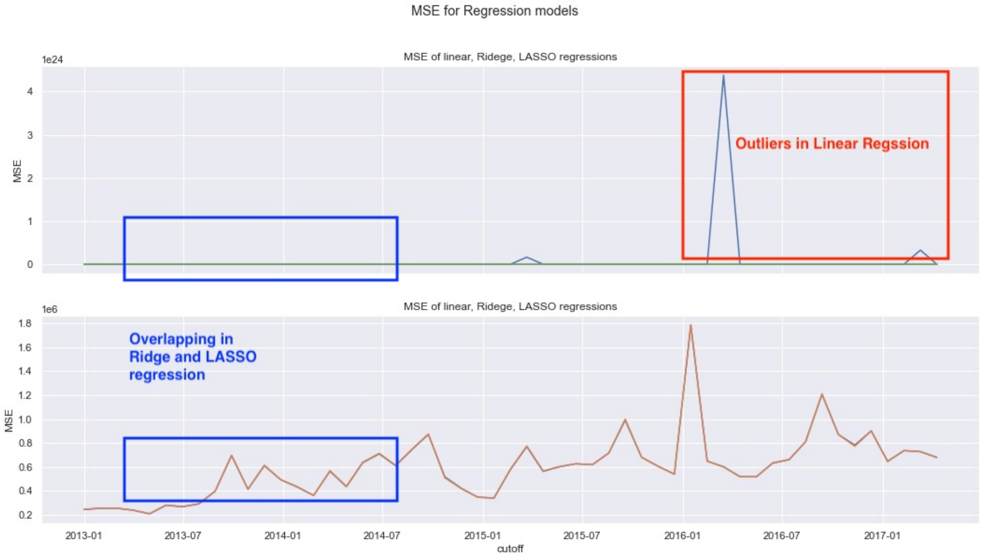


Figure 7: MSE for cutoffs tainded with regressions

Interestingly, even that the MSEs for all three models fluctuate through the cutoffs, the linear

[5] refer to the regressions.ipynb in modeling folder for more code

regression model returns a few massive outliers that in the plot even forced the other two to almost a strait line. Moreover, from the lineplots for just ridge and LASSO regressions are almost overlapping with each other, which from my understanding is reasonable as the idea behind these two regressions are similar and they just take different terms as the penalty.

4.4 Decision Trees

Different from the regression models, theoretically decision trees' structure should fit the categorical features better. Moreover, tree should be able to take the integer encoded categorical features directly. So I first tried to feed the tree model with integer encoded and one hot encoded data respectively and compare the results.

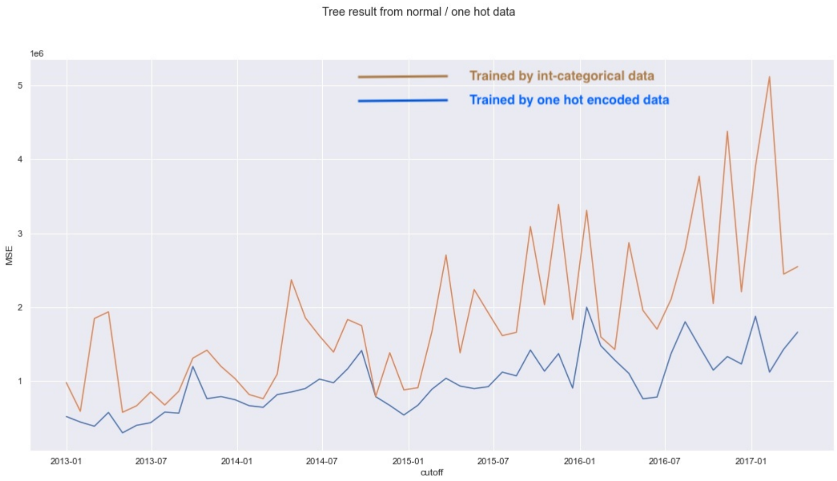


Figure 8: Comparing the performance by int / one hot encoded data

From the plot above, it is clear that even under the same tree model with same hyperparameters, training by the one hot encoding data would return better (less MSE and more consistent over cutoffs). Next I tried to improve the performance by some ensemble methods like adaboost. Below is the performance of tree model improved by adaboost.

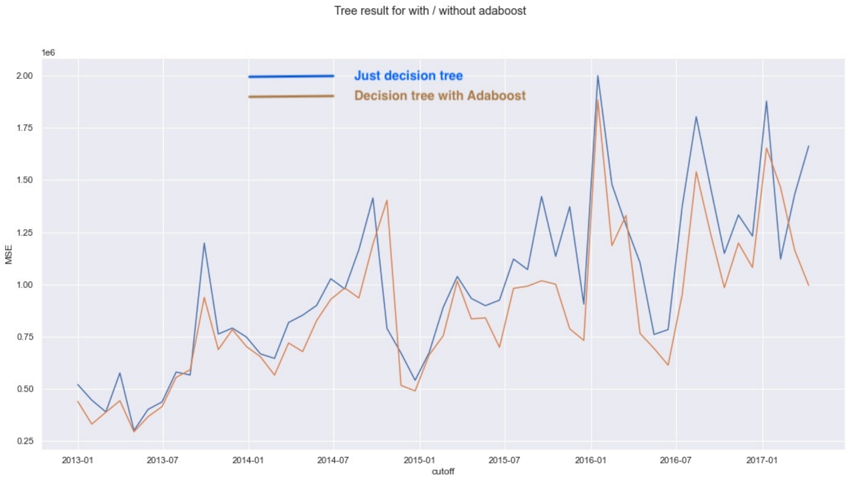


Figure 9: Comparing the performance by normal / adaboosted tree

From the plot, we can see that the performance is slightly improve by the adaboost method. However, adaboosted decision tree takes much longer to train compared with normal decision tree[6].

5 Results: Cross-model Evaluation

With the several models above, In this section I will compare the performance of these methods. As we already found that simple linear regression model would return result with high inconsistency (massive outliers), so in this section the cross model comparison does not include the linear regression model. Here are the plot for ridge and LASSO regression and decision trees. [7].

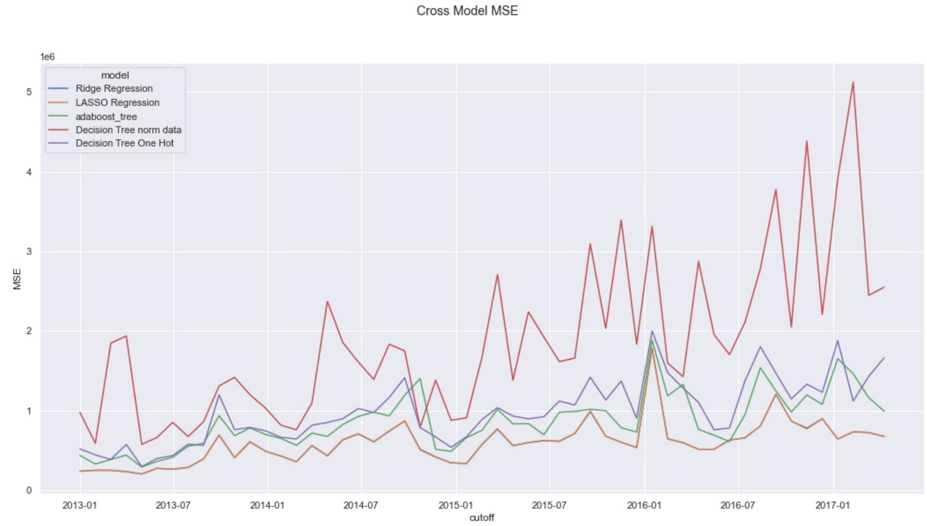


Figure 10: MSE for Different Models - 1

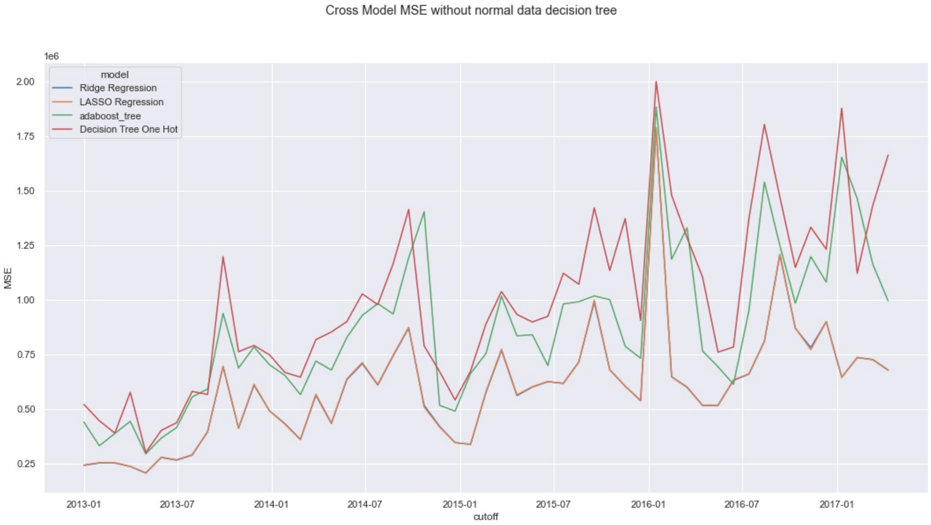


Figure 11: MSE for Different Models - 2

[6] refer to the decision tree.ipynb in modeling folder for more code

[7] refer to the cross model evaluation.ipynb in modeling folder for more code

From the plot, we can see that by one hot encoding the categorical features, the overall MSE would be smaller and more consistent. And by checking the results by the one hot encoding data more closely, an interesting finding is that despite the decision tree models seem to be more complex and take more time to train, the result form ridge / LASSO regression would return slightly better result. So in application, it could be a goog idea to simply use a ridge / LASSO regression.

6 Next Steps

6.1 RNN - on going

Recurrent neuron network can be a very good idea for the task of time series data predicting. But because of the time limit, I have not finished building the RNN for this project. The on going note book is in the modeling folder named RNN.ipynb.

6.2 Possible Improvement

More possible improvements can be done by tuning the hyperparameters, such as the cutoff time range, and more in the model like the max depth in the decision trees. Moreover, I am also planning to add another model to combine the results from different models.

7 Appendix : Code Structure

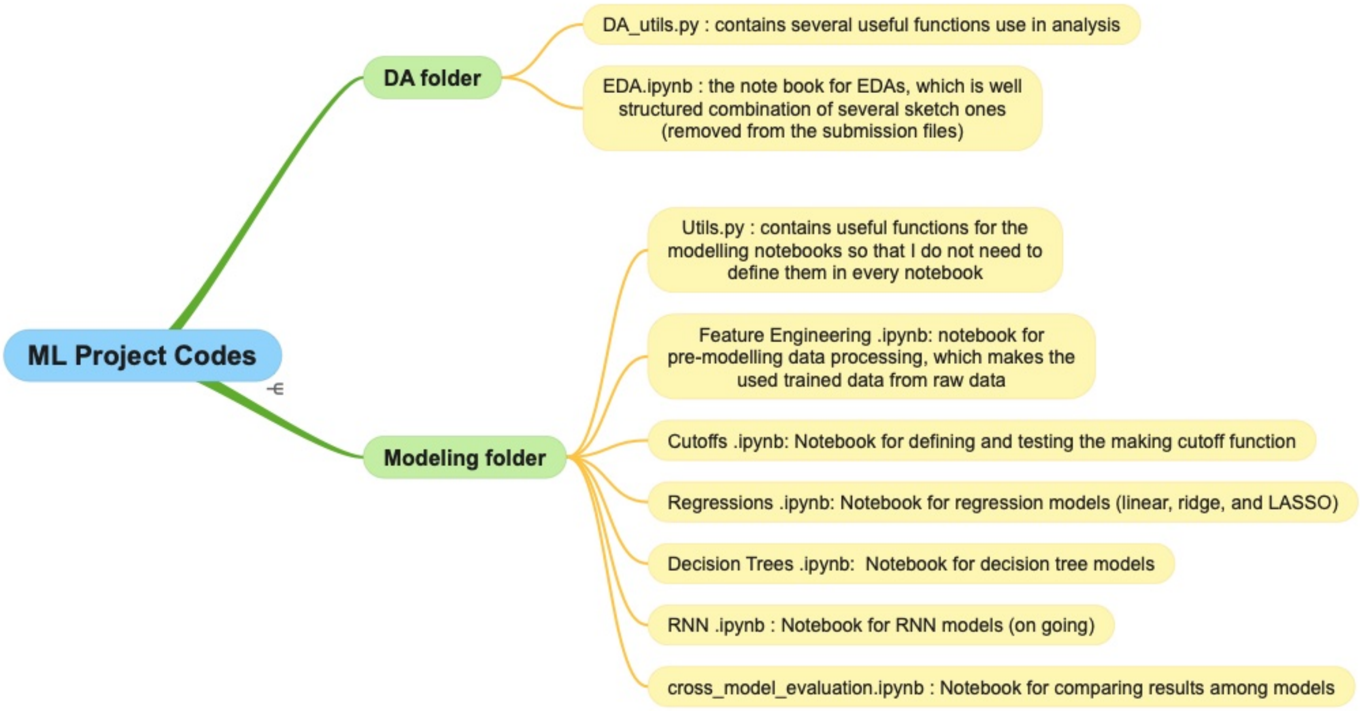

ML Project Codes
DA folder
DA_utils.py : contains several useful functions use in analysis
EDA.ipynb : the note book for EDAs, which is well structured combination of several sketch ones (removed from the submission files)
Modeling folder
Utils.py : contains useful functions for the modelling notebooks so that I do not need to define them in every notebook
Feature Engineering .ipynb: notebook for pre-modelling data processing, which makes the used trained data from raw data
Cutoffs .ipynb: Notebook for defining and testing the making cutoff function
Regressions .ipynb: Notebook for regression models (linear, ridge, and LASSO)
Decision Trees .ipynb: Notebook for decision tree models
RNN .ipynb : Notebook for RNN models (on going)
cross_model_evaluation.ipynb : Notebook for comparing results among models

Figure 12: Code and Notebook Structure for this Project