Analysis on M5 Forecasting – Accuracy

Huang Zhenyu 20744676

Luo Jiahao 20744418

Yang Yannan 20746131

Department of Mathematics, School of Science

The Hong Kong University of Science and Technology

Contents

1 Introduction

Department stores like Walmart have uncountable products and money transactions every day. Because of their rapid transaction rates, keeping a balance between inventory and customer is most important. Therefore, making an accurate sales prediction for different products becomes an essential need for stores to optimize profits.

The M5 Prediction Competition aims to improve the theory and practice of forecasting by identifying the method that provides the most accurate integral prediction for each of the time series of 42,840 commodity groups. In addition, in order to obtain information to estimate the uncertainty distribution of the realized values of these series as accurately as possible, participants in M5 were asked to provide 28-day advance predictions (PF) for all series races, along with the corresponding median and prediction intervals (PIs) of 50%, 67%, 95% and 99%.

2 Data description

The data provided by Walmart, covers stores in three US States (California, Texas, and Wisconsin) and includes item level, department, product categories, and store details. In addition, it has explanatory variables such as price, promotions, day of the week, and special events. Together, this robust data set can be used to improve forecasting accuracy.

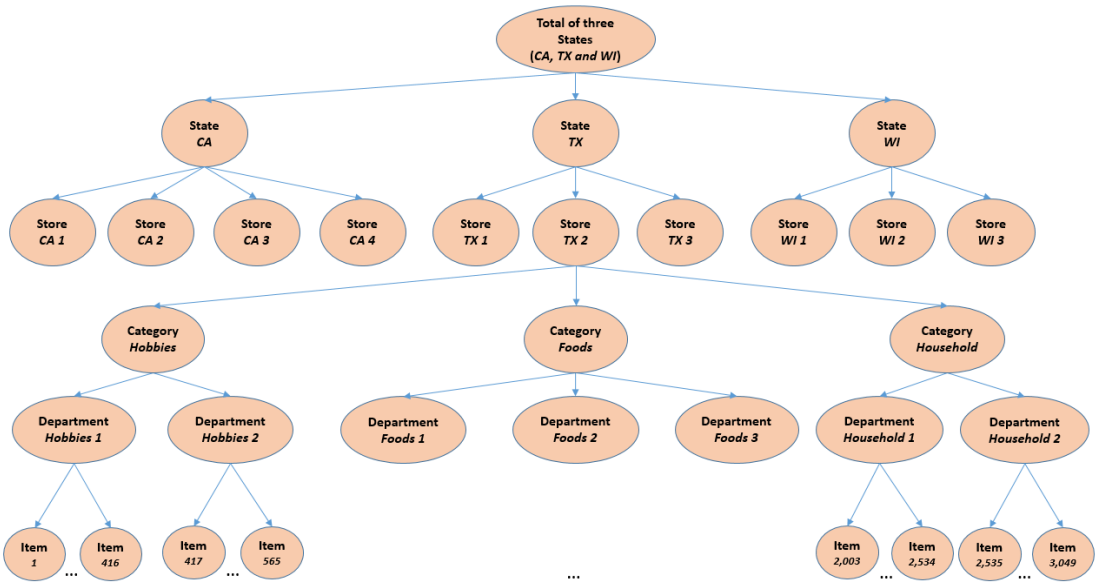

Figure 1: An overview of how the M5 series is organized

2.1 Item counts in each store and category

We counted the number of categories of items sold in each store and found that the number of categories of items in each store was the same. Among them, there are 1437 categories of food goods, 1047 categories of household goods and 565 categories of hobbies. The category of food goods is the largest, followed by household goods and hobby goods. This statistical result is also very consistent with the consumption habits of the public.

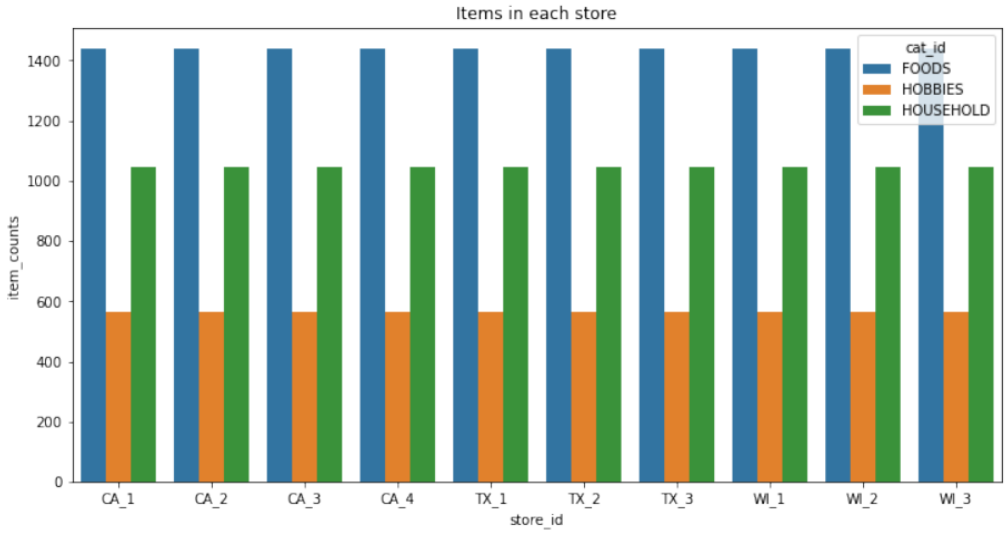

Figure 2: Item counts in each store and category

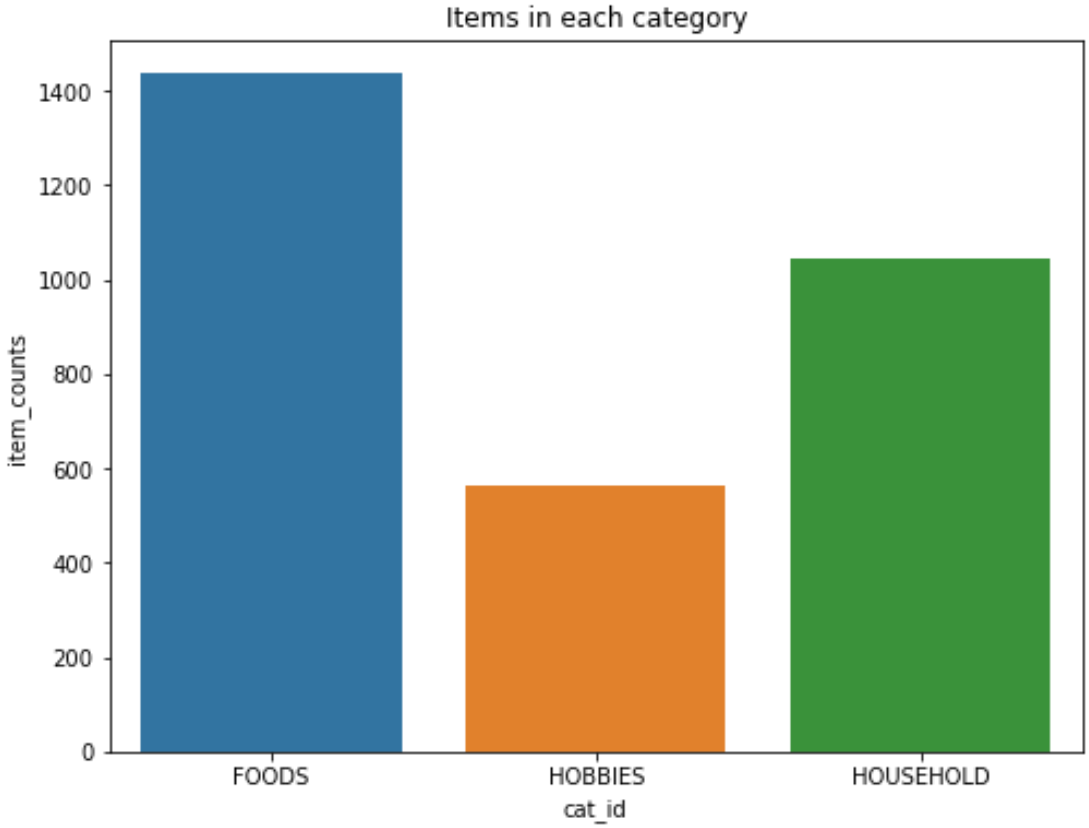

Figure 3: Item counts in each category

We fill in the missing values of items sold and sell price with -1, then we found that valid data, non-null values, accounted for 79.51% of the original sell price data, and 31.55% of items were sold at valid prices.

2.2 Sales and sell price

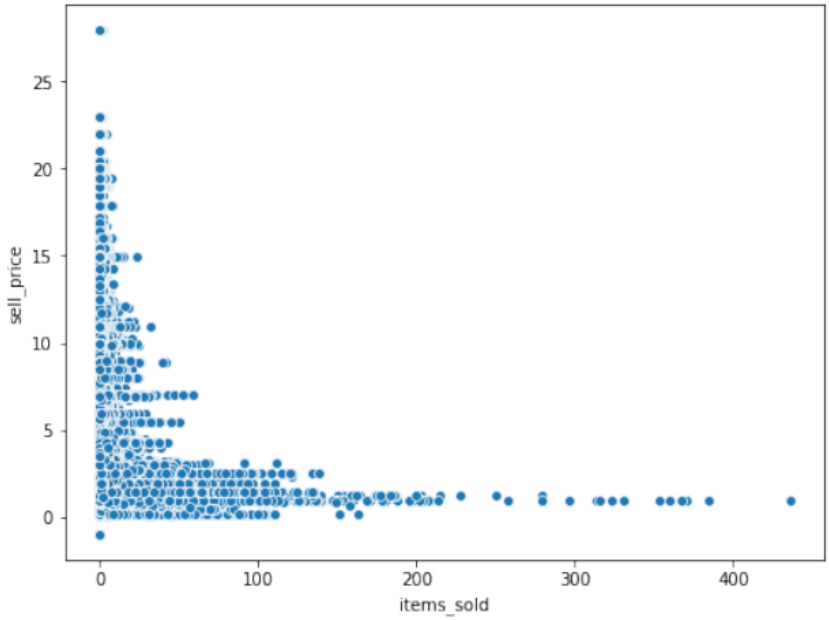

Figure 4: Distribution of items sold and sell price

In terms of FOODS, sell price is almost same for every store and lowest among all categories. In terms of HOBBIES, sell price is same for every store and higher than FOODS category. In terms of HOUSEHOLD, sell price is highest among all categories and sometimes sell price goes very high (more than 100).

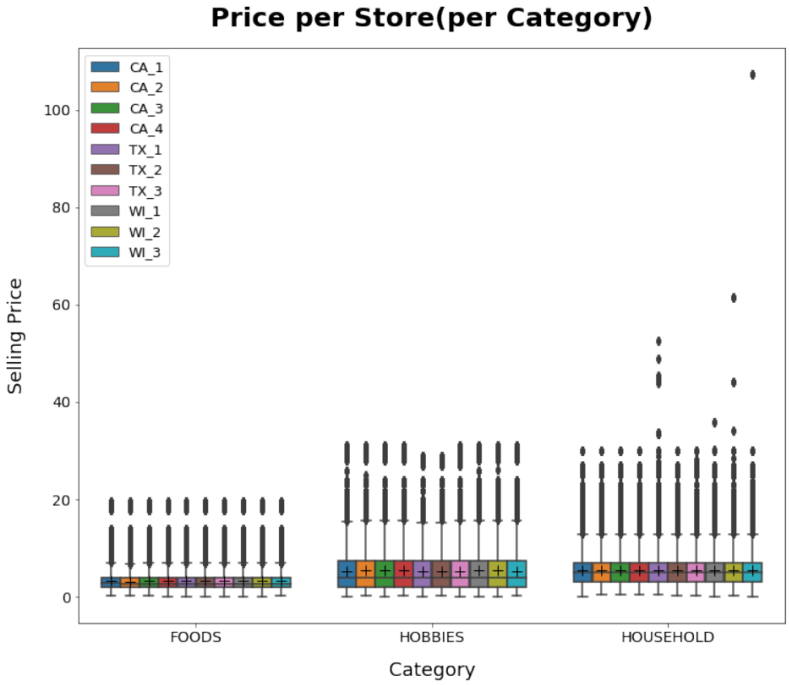

Figure 5: Sell prices in different stores and different categories

2.3 The sales aggregated by category

We studied the sales data of different categories of goods in different stores and in different months and found that the sales of food goods fluctuated relatively greatly in different stores and in different months

than that of hobbies category goods and household category goods. Sales of HOBBIES is almost constant while Sales of HOUSEHOLD raises gradually. This may indicate that people in different states have significantly different levels of food consumption, and there may also be seasonal effects in people's consumption behavior of food commodities.

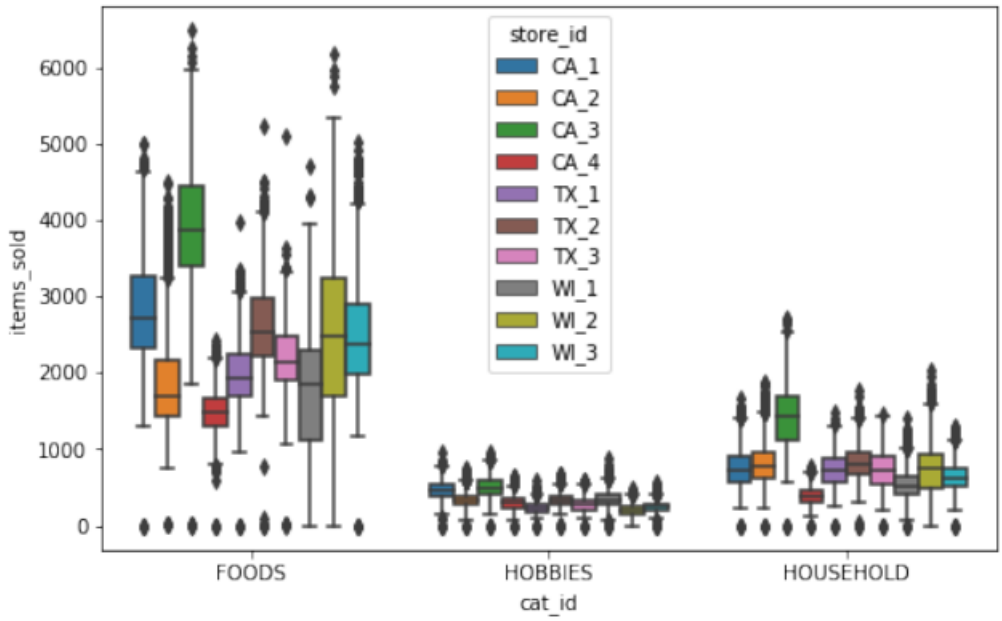

Figure 6: The sales in each category and store

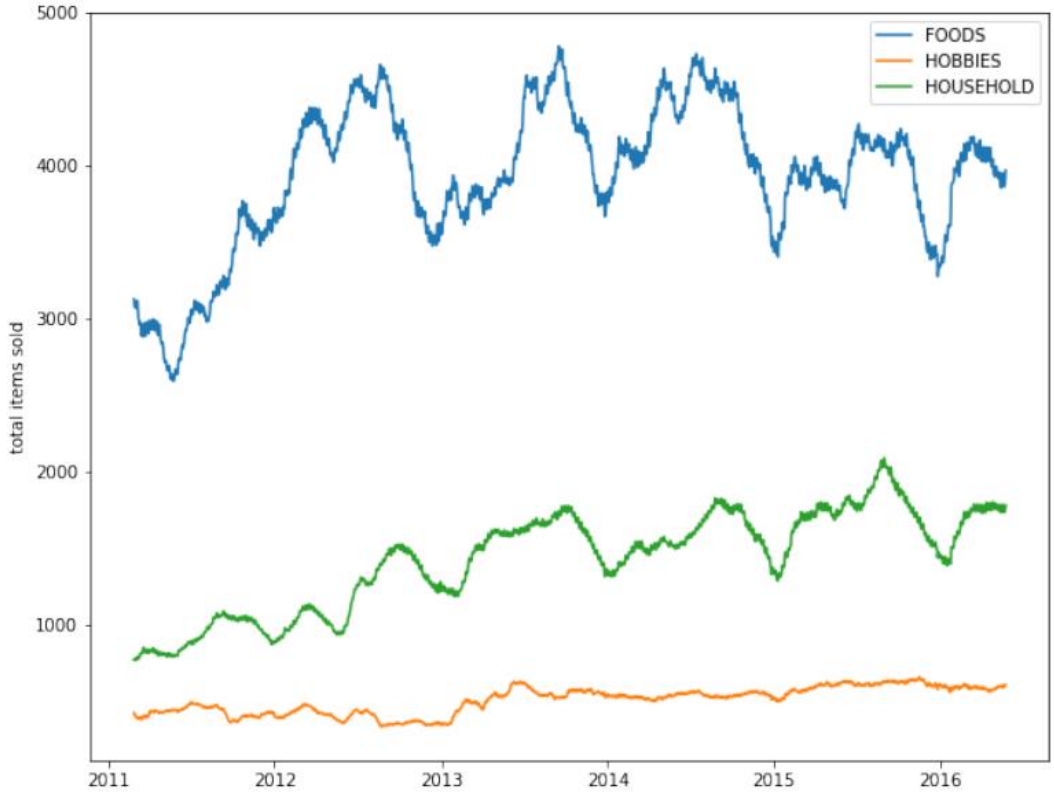

Figure 7: Total items sold of each category in different years

2.4 The sales aggregated by stores

We also counted the 30-day rolling mean number of items sold in each store. As can be seen from the figure below, the perennial sales volume of CA_3 ranks first, while the sales volume of CA_4 is relatively poor. In addition, we can roughly observe that the sales volume is increasing year by year, and the annual sales volume has a certain cyclical change. The sales volume often peak in around September every year.

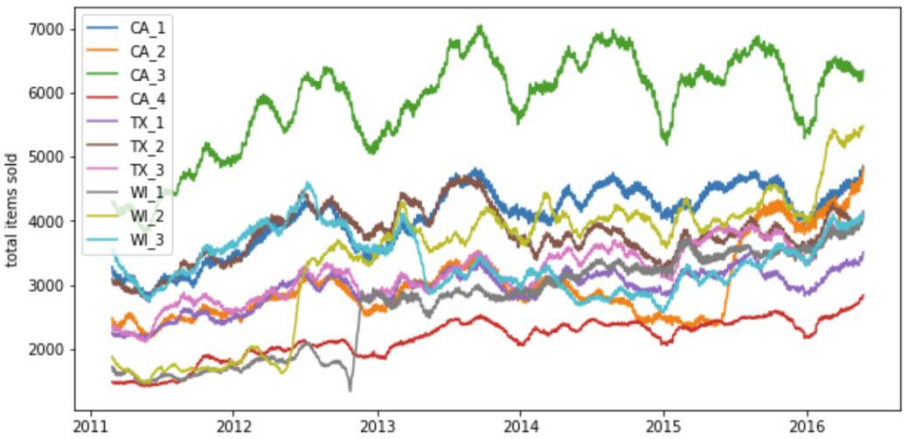

Figure 8: The total items sold of each store in different years

2.5 Holiday Season Investigation

To verify this cyclical rule, we selected the data of CA_3 for calendar heatmap drawing. From the heat map, we find that sales are higher in August and September every year, and we also find that sales are higher on weekends. To get a better perspective, we looked at the percentage change in the daily value of the 30-day rolling average relative to the overall average between 2015.01.01 and 2016.01.01, and it became clear that three commodities were particularly volatile in September. This could mean that people shop more frequently during holidays and weekends.

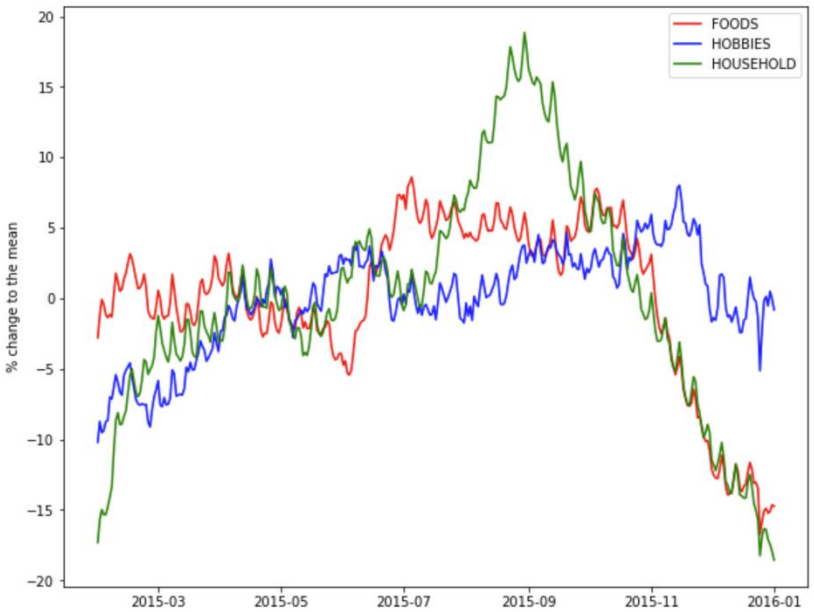

Figure 9: Percentage change of 30-day rolling mean between 2015-2016

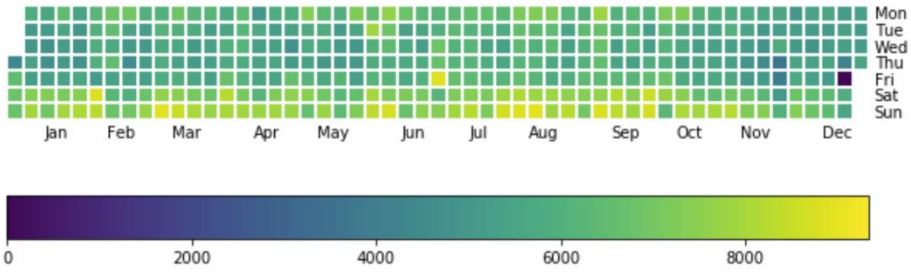

Figure 10: The heatmap of sales data of CA_3

3 Data processing

3.1 Memory reduction

To reduce memory consumption and improve performance, we have used many methods to improve the running speed of our code. Firstly, the reduce_mem_usage function helps us reduce the space occupied by the data in the memory by adjusting the data type. Before adjustment, the data type of the feature is fixed to int16, float32, object, datetime and other types, and some features do not require 16-bit or 32-bit to storage and perform. We made interval judgment on the characteristics of each column, allocated appropriate storage units, and effectively reduced the memory footprint.

Secondly, we made good use of the garbage collection function to delete unnecessary variables in time. Thirdly, we selected a smaller date rage to save memory.

3.2 Melt and combine the data

Currently, the dataframe contains daily sales data with days(d_1-d_1969) as columns.In this case what the melt function is doing is that is converting the sales dataframe which is in wide format to a long format. We have kept the id variables as id, item_id, dept_id, cat_id, store_id. They have in total 30490 unique values when compunded together. Now the total number of days for which we have the data is 1969 days. Therefore, the melted dataframe will be having 30490*1969 = 60034810 rows.

3.3 Feature engineering

From data analysis, we found that there may also be seasonal effects in people's consumption behavior of food commodities. Therefore, we added the is_weekend feature. We also want to see whether special events will affect people’s purchases, such as snap purchase activities, etc. We found that store, sells price and other factors will affect the purchase. Based on the time series, we created a sliding window function to create new features. We selected the maximum and average weekly sales as features to predict the data of the next day. In the case of a rolling window, the size of the window is constant while the window slides as we move forward in time. Hence, we consider only the most recent values and ignore the past values.

We encoded the characteristics of the string type, such as item_id, dept_id, cat_id, but there are too many types. One-hot will cause sparse data (and occupy a lot of memory), so we used the label encoder method.

4 Modelling

4.1 Training by aggregation

Level id	Aggregation Level	Number of series
1	Unit sales of all products, aggregated for all stores/states	1
2	Unit sales of all products, aggregated for each State	3
3	Unit sales of all products, aggregated for each store	10
4	Unit sales of all products, aggregated for each category	3
5	Unit sales of all products, aggregated for each department	7
6	Unit sales of all products, aggregated for each State and category	9
7	Unit sales of all products, aggregated for each State and department	21
8	Unit sales of all products, aggregated for each store and category	30
9	Unit sales of all products, aggregated for each store and department	70
10	Unit sales of product *x*, aggregated for all stores/states	3,049
11	Unit sales of product *x*, aggregated for each State	9,147
12	Unit sales of product *x*, aggregated for each store	30,490
Total		**42,840**

Figure 11: Number of M5 series per aggregation level

From the official document, we know that there are a total of 12 levels of sales information aggregated by 4 main characters: state, store category and department. We believe our prediction for the future sales should consider the effect of this aggregations and their combinations, since by making prediction within these levels, we are able to generate multiple time series sub datasets. The weighted final prediction can be more accurate as the variance is reduced during averaging work.

4.2 Reduce training scale

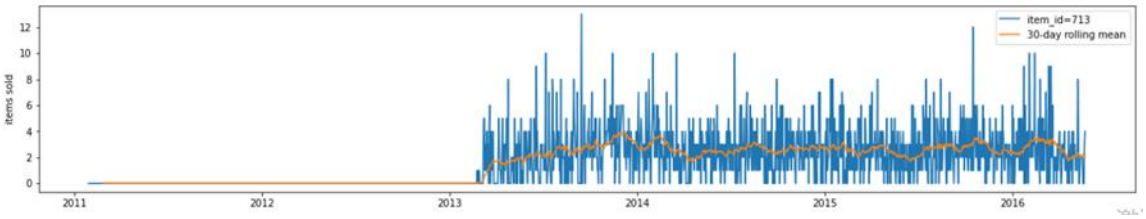

Figure 12: Sales trend by time of item 713

In our data description work, we found that 25% of the products are not sold from 2011 to 2013. The sales volume is 0, which means it makes very little sense in our prediction. So to exclude these noises and reduce the computational scale, we simply drop out the information in the first 2 years and use only the data from mid-2013 to train our models.

4.3 Target loss function

The evaluation metric of the competition is defined as

$$\boldsymbol{RMSSE} = \sqrt{\frac{1}{h}\frac{\sum_{t=n+1}^{n+h}\left(Y_t - \widehat{Y}_t\right)^2}{\frac{1}{n-1}\sum_{t=2}^{n}(Y_t - Y_{t-1})^2}}$$

where Y_t is the actual future value of the examined time series at point *t*, $\widehat{Y}_t$ the generated forecast, *n* the length of the training sample (number of historical observations), and *h* the forecasting horizon. The final ranking is judged by estimating the RMSSE for all the 42,840 time series data of the competition, the participating methods will be ranked using the Weighted RMSSE (WRMSSE). It should be way too complicated if we utilize this loss function in our model training.

However, according to a statistical assumption, the distribution of future sales can be approximately regarded as a passion process, since it is like a queuing problem, and we assume the sales in 2 different periods do not have mutual dependence. Therefore, we adopt the Poisson loss function in our model training. Another reason to use Poisson is that it is capable of handling data with large amount of 0s.

4.4 Handling missing values

From previous discussion, we know that not all products have sales information in all the recorded days, so there should be some missing values if we melt the whole data. However, sales equal to 0 does not mean that the sell price is also 0, and we have no information of how the historical price is distributed. In this case, the best method to deal with the NaNs is to just leave it as is. We will not perform data imputation to bring in extra information and avoid information leak.

4.5 Model selection

Based on the discussion in section 4.4, we need to select a model that can both cope with large amount of zero values and the missing data. So, we will not use simple time-series prediction and linear methods they are not suitable in our case. Instead, we choose tree-based method that helps achieve our goal. Recent research in tree-based method has generated models that can have faster training speed, recognize categorical data and missing data, and use less computational power. XGBoost, Catboost and LightGBM are 3 methods that have better prediction compared to traditional gradient boosting trees. In our analysis we choose LightGBM as our main model.

LightGBM is a gradient boosting framework that uses tree-based learning algorithms. It is designed to be distributed and efficient with the following advantages: faster training speed and higher efficiency, lower memory usage, better accuracy, support of parallel, distributed, and GPU learning and capable of handling large-scale data. One main characteristic of LightGBM is that it uses histogram algorithm in training. Histograms are generated by ‘bin’ method, a method feature compression. It maps continuous values into discrete ones, which generates new features, and uses the newly generated ones for prediction. It greatly saves computational time by using discrete characteristics and overcomes overfitting by restricting the maximum depth of tree.

5 Result

In our analysis, we create new features such as rolling windows with mean and median value of sales and sell price, time stamps like indicator of weekend, price momentum and statistical description of sell price, and the id of variables. We use LightGBM to train our model within category, state, department,

and their combinations. We directly predict the future sales in the whole testing horizon instead of using recursive one since we believe the prediction can go to a very different direction if we cannot deal with the prediction trend very well. Besides, we only use LightGBM for prediction without combining other methods.

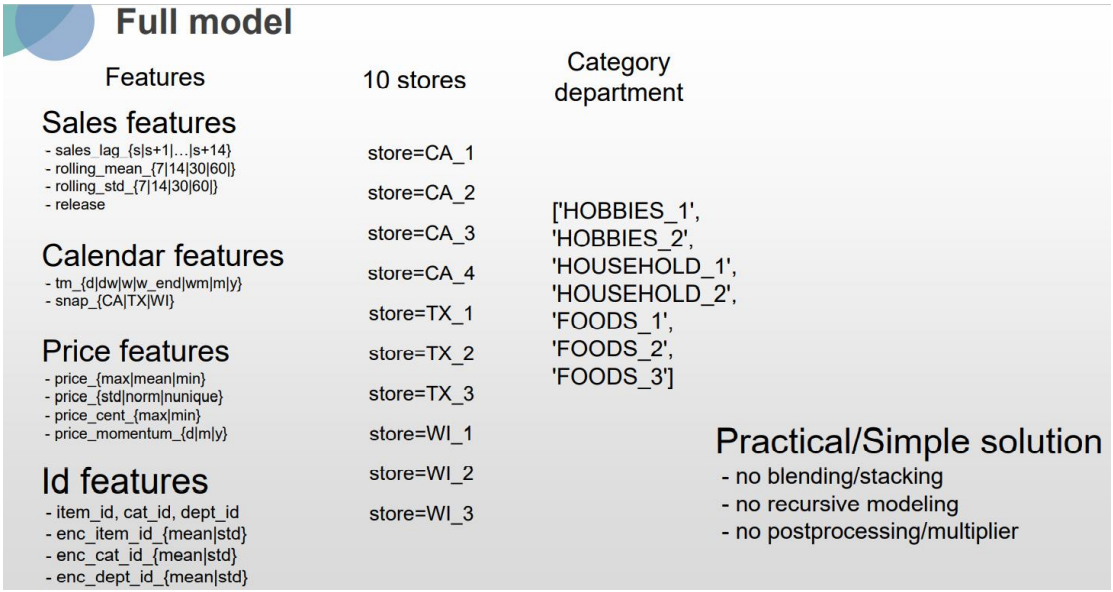

Figure 13: conclusion of the approaches

However, due to the limitation of our devices, we are not able to perform well in this project. The result on Kaggle is poor, and we believe it is due to our training method. We also plot the feature importance below, and it seems the rolling window feature of 30 and 60 days, as well as the ID of items contribute a lot to final prediction.

6 Conclusion, future work, and acknowledgement

We conduct this simple replication work following the dedicated work by those who have already shared their experience online. The result is not satisfactory, and we may consider doing future works.

First, instead of using a single model, we can combine other methods like Catboost and LSTM network, both are considered suitable in this time-series task. Introducing various methods may help overcome the shortcomings of overfitting in only one model. Second, we can try week-by-week methods instead of daily ones to get more robust features. Third, autoencoder and PCA may be useful in reducing the dimension of the huge dataset.

Thanks to those who shared their works online!

The presentation video is in the following link: https://hkust.zoom.us/rec/share/Wd-Wom6H1o0uVanTxlxu1P4XBQckWJyYgm5wHcjlZiMKFtP4Vfsa2Aitt_nuhQoy.2hR7soX5hm2mLG37?startTime=1639324464000

Reference

[1] M5-methods, https://github.com/Mcompetitions/M5-methods

[2] Anshul Sharma, Time Series Forecasting-EDA, FE & Modelling, https://www.kaggle.com/anshuls235/time-series-forecasting-eda-fe-modelling
[3] Dipanshu Rana, M5 Forecasting — Accuracy, https://dipanshurana.medium.com/m5-forecasting-accuracy-1b5a10218fcf
[4] Konstantin Yakovlev, Few thoughts about M5 competition, https://www.kaggle.com/c/m5-forecasting-accuracy/discussion/138881

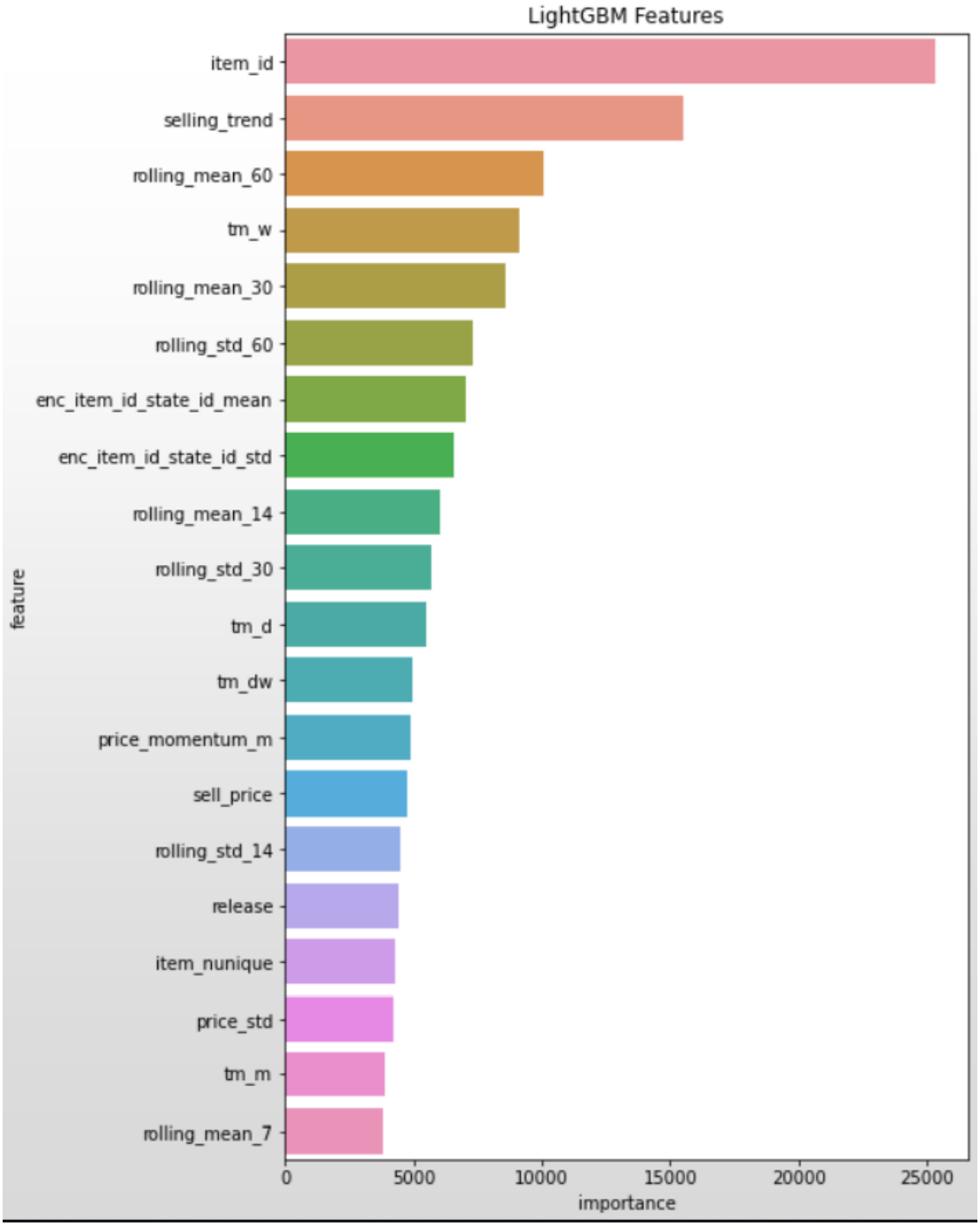

Figure 14: most importance features generated by LightGBM