M5 Forecasting Report

Qiu Qingqing
20799190
qqiuae@connect.ust.hk

LI Muxiao
20787989
mlidm@connect.ust.hk

1. INTRODUCTION

1.1 Background

A sales forecast is an estimate of the quantity and value of products to be sold at a specified time in the future. Its essence is in the full consideration of the future on the basis of various factors combined with the actual sales performance of the enterprise, through a certain analysis method to put forward feasible sales goals. Sales forecasting based on historical data plays an important role in helping enterprises make better business strategies. It can be used to calculate the sales target, determine the total cost, so as to predict the profit and loss. In anticipation of the loss, measures can be taken to control and reduce costs, which ensure that the enterprise will not die quickly because of high costs and high inventory.

In this project, our goal is to predict retail giant Walmart's sales per item over the next 28 days, which will be carried out in the following three phases. Firstly, data integration and cleaning are carried out through data preprocessing. The second step is to visualize the integrated data and intuitively understand the relationship between variables. Finally, the regression model is built and the predictive analysis is carried out on the basis of the previous two steps.

1.2 Dataset

The data set used in this project is from Kaggle's competition M5 Forecasting which covers Walmart from three US states(California, Texas, and Wisconsin) , involving 10 stores, 3 categories of products, 3049 products and a total of 42840 time series. In addition, it has explanatory variables such as price, day of the week, and special events.

Time series data from 2011-01-29 to 2016-06-19 are day-granularity data with a time span of about five and a half years, which can be subdivided into 12 levels according to different object categories and levels. There are four data files we mainly used in our project, which are as follows.

train-evaluation.csv: Contains the historical daily unit sales data per product and store; [d1,d1941].
calendar.csv: Contains the dates on which the products are sold along with associated functions (such as day of the week, month, year) and 3 binary markers that indicate whether stores in each state are allowed to purchase SNAP food stamps on that date.
sell-prices.csv: Contains information about store, item ID, and average weekly prices of the products sold per store.
sample-submission.csv: The correct format for submissions.

Figure 1 shows the data style.

	id	item_id	dept_id	cat_id	store_id	state_id	d_1	d_2	d_3	d_4	...
0	HOBBIES_1_001_CA_1_validation	HOBBIES_1_001	HOBBIES_1	HOBBIES	CA_1	CA	0	0	0	0	...
1	HOBBIES_1_002_CA_1_validation	HOBBIES_1_002	HOBBIES_1	HOBBIES	CA_1	CA	0	0	0	0	...
2	HOBBIES_1_003_CA_1_validation	HOBBIES_1_003	HOBBIES_1	HOBBIES	CA_1	CA	0	0	0	0	...
3	HOBBIES_1_004_CA_1_validation	HOBBIES_1_004	HOBBIES_1	HOBBIES	CA_1	CA	0	0	0	0	...
4	HOBBIES_1_005_CA_1_validation	HOBBIES_1_005	HOBBIES_1	HOBBIES	CA_1	CA	0	0	0	0	...
5	HOBBIES_1_006_CA_1_validation	HOBBIES_1_006	HOBBIES_1	HOBBIES	CA_1	CA	0	0	0	0	...
6	HOBBIES_1_007_CA_1_validation	HOBBIES_1_007	HOBBIES_1	HOBBIES	CA_1	CA	0	0	0	0	...
7	HOBBIES_1_008_CA_1_validation	HOBBIES_1_008	HOBBIES_1	HOBBIES	CA_1	CA	12	15	0	0	...
8	HOBBIES_1_009_CA_1_validation	HOBBIES_1_009	HOBBIES_1	HOBBIES	CA_1	CA	2	0	7	3	...
9	HOBBIES_1_010_CA_1_validation	HOBBIES_1_010	HOBBIES_1	HOBBIES	CA_1	CA	0	0	1	0	...

Figure 1: train-evaluation.csv

2. THEORY

The theories about LightGBM(LGBM) models we used to research our topic are introduced in this section.

2.1 Gradient Boosting Decision Tree

Gradient Boosting Decision Tree (GBDT) is a Boosting Decision Tree model in machine learning, whose main idea is to use weak classifier (Decision Tree) iterative training to get the optimal model, which has the advantages of good training effect and not easy to over-fit. However, the algorithm needs to traverse the entire training data for many times in each iteration, and repeatedly reading and writing training data will consume a lot of time. Therefore, the general GBDT algorithm can not deal with industrial-level massive data. The proposed LightGBM is a good solution to this problem, which provides a framework to implement GBDT algorithm, support efficient parallel training, and has a faster training speed, lower memory consumption and better accuracy.

2.2 LightGBM

LGBM is an improved implementation of GBDT algorithm framework. It is a fast, distributed and high-performance GBDT framework based on decision tree algorithm. It is proposed to solve the problems encountered by GBDT in massive data, so that GBDT can be better and faster used in industrial practice.

Specifically, the "Light" in the LGBM model is mainly reflected in three aspects. Firstly, Gradient-based one-side

Sampling technique is used to generate fewer samples. At second, Exclusive features Bundling technology yields fewer features. Last but not least, Histogram yields less memory. At the same time, LGBM also make many optimizations in parallel computing, supporting feature parallelism and data parallelism, and optimizing their parallel modes to reduce traffic.

Before the introduction of LGBM, the best-known GBDT tool was XGBoost, which was a decision tree algorithm based on the pre-sorting method. Compared with XGBoost, LGBM is a decision tree algorithm based on Histogram, which proposes Gradient-based One-side Sampling(GOSS) and Exclusive Feature Bundling(EFB)to optimize the down-sampling and dimension reduction.

Definition 1. GOSS is a technique that can reduce a large number of data instances with only small gradients, so that only the remaining data with high gradients can be used in the calculation of information gain, which saves a lot of time and space overhead compared with XGBoost traversing all eigenvalues.

Definition 2. EFB is a technique that can bind many mutually exclusive features into one feature, thus achieving the purpose of dimensional reduction.

3. METHODOLOGY

3.1 Data Analysis

This section discusses the basic characteristics of data and some data exploration analysis we performed which are the preparation for the model building.

3.1.1 Data Aggregation

Since the dataset we need to work on is extremely huge, before feeding this dataset into the model, we need to aggregate several csv files to obtain a tabular file which has a long format instead of wide format(Figure 2). The compounded key variables contains id, item-id, dept-id, cat-id, store-id and state-id, which in total have 30490 unique values . And the total number of days is 1969, which lead to 30490x1969=60034810 rows of the aggregated dataframe.

	id	item_id	dept_id	cat_id	store_id	state_id	d	sales
0	HOBBIES_1_001_CA_1_evaluation	HOBBIES_1_001	HOBBIES_1	HOBBIES	CA_1	CA	d_1	0
1	HOBBIES_1_002_CA_1_evaluation	HOBBIES_1_002	HOBBIES_1	HOBBIES	CA_1	CA	d_1	0
2	HOBBIES_1_003_CA_1_evaluation	HOBBIES_1_003	HOBBIES_1	HOBBIES	CA_1	CA	d_1	0
3	HOBBIES_1_004_CA_1_evaluation	HOBBIES_1_004	HOBBIES_1	HOBBIES	CA_1	CA	d_1	0
4	HOBBIES_1_005_CA_1_evaluation	HOBBIES_1_005	HOBBIES_1	HOBBIES	CA_1	CA	d_1	0
5	HOBBIES_1_006_CA_1_evaluation	HOBBIES_1_006	HOBBIES_1	HOBBIES	CA_1	CA	d_1	0

Figure 2: Long Format Data

3.1.2 Exploratory Data Analysis

(1) Intermittent Data

By visualizing the distribution of zeros per time series(Figure 3), we found that the mean value is around 0.8, which means there is a lot of intermittent data. To observe this phenomenon more clearly, we randomly chose an item and visualized its sale(Figure 4). Obviously this particular item had

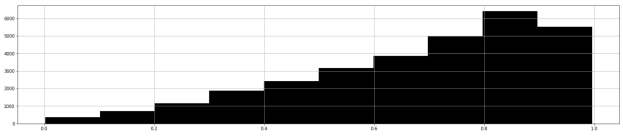

Figure 3: The Zero Distribution of Time Series

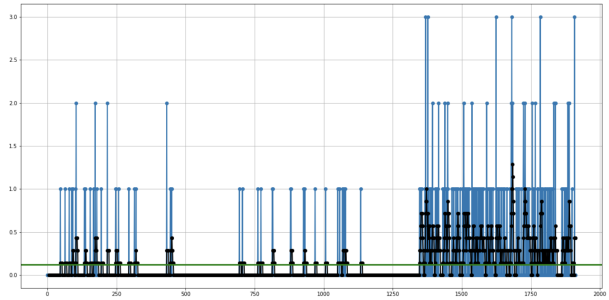

Figure 4: Time Series of Different Items

quite a lot of days where the sales was zero. This problem would be a huge challenge for this project.

(2) State Wise Analysis

In order to find out some potential patterns, we plotted the daily sales across three states(Figure 5). From daily sales graph, we can find that in the end of each year, there exists one day of which the sale is zero. The sharp decline before and after the Christmas season and the small decline in other points show the importance of inputting holiday details.

However, due to the high volatility of daily sales data, monthly sales data are used instead of daily sales data in order to better observe the variation trend of data. It can be seen that the trend line is much clearer(Figure 6). At the same time, we found that the sales time series of the three states showed basically the same patterns(Figure 7). For example, sales are always lower at the beginning of the year and higher in the middle, and overall sales growth year by year, macroscopically there is a strong cyclical and seasonal. Periodicity is reflected in the week and month dimension. And seasonality is reflected in the year, for example, summer and autumn sales are lower than spring and winter sales. And the latest sales figures for 2016 appear to be growing faster than in previous years.

Also, people in CA are better off and more goods are sold. WI sales gradually caught up with TX after 13 years and eventually surpassed TX in the final months of the data.

(3) Category Wise Analysis

When it comes to the category wise sales, we found that FOODS has the highest sales, followed by HOUSEHOLD, while HOBBIES has the lowest. And Foods has the most volatile sales curve, while HOBBIES has the smoothest. Meanwhile, for all three categories, they are all sold more on Saturday and Sunday. Food sales were most closely correlated with different days of the week, with HOUSEHOLD also slightly affected, while HOBBIES sales were almost the same on every day of the week(Figure 9).

Now that we learn the difference among different categories,we

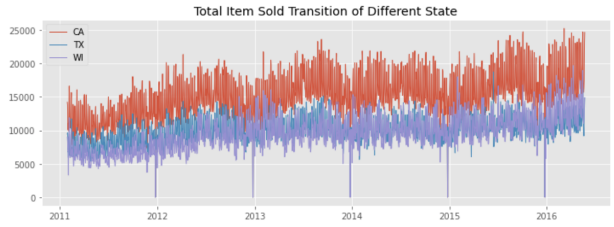

Figure 5: State Wise Time Series of Daily Sales

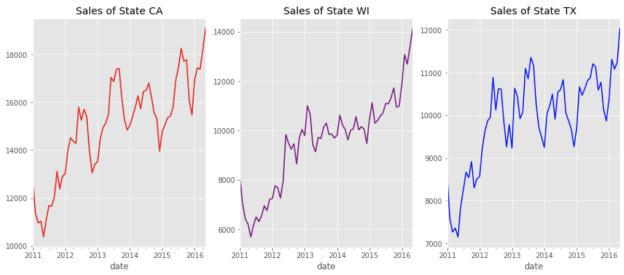

Figure 6: State Wise Time Series of Monthly Average Sales

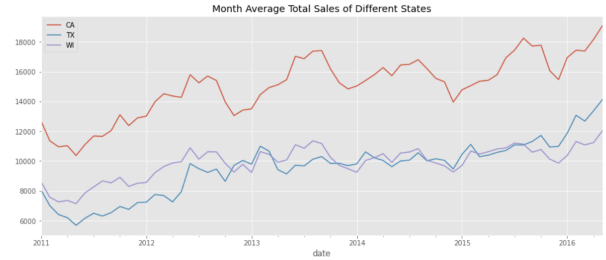

Figure 7: Comparison of Monthly Average Sales

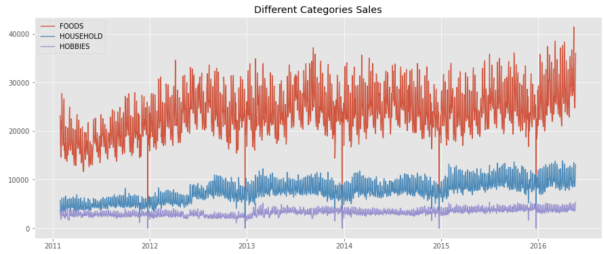

Figure 8: Category Sales

explored the insights of each department. From figure 10, we observed that the food 3 sales are more than other departments. Therefore, the food 3 sector may include food for daily consumption, such as rice.

(4) Store Wise Analysis
Table 3 shows that there are 12 stores in three state. The monthly average sales for all stores in each state are visualized in Figure 11. We observed that store CA 3's sales are

Table 1: Store Distibution

State	Store ID	Number
CA	$CA_1; CA_2; CA_3; CA_4$	4
WI	$WI_1; WI_2; WI_3$	3
TX	$TX_1; TX_2; TX_3$	3

more than other stores, it may be a bigger store. Obviously, the sales of stores in all the three states showed seasonality. Compared to the other two states, the sales of most stores located in CA fluctuates around a relatively stable value. However, the sales of stores in WI and TX states showed an obvious upward or downward trend year by year, with stronger fluctuation. People in different states have different income, which influence their purchasing behaviors.

(5) Price Analysis
Most commodities range in price from 0 to 5. Most FOODS prices are concentrated around 2.5, while HOUSEHOLD has a wider range of price concentration between 2.5 and 6.5. HOBBIES do not have a continuous range of concentration, so the dispersion is stronger.(Figure 12)

By observing the boxplot of sell price per category(Figure 13), We observed that more expensive goods were purchased much less frequently, and the chart was highly right leaning. We can see that, as expected, price plays a great role in sales. Similar trends were observed in all three categories. It was found that the average price level of those three categories were not much different, with HOBBIES having the widest range of selling prices and FOODS having the smallest range. Besides, the sell price of HOUSEHOLD has the most outliers which are far away from the normal range.

(6) Calendar Analysis
Firstly, in a year, August has the highest overall sales with around 36K units sold(Figure 14). Figure 15 shows the sales of weekends before different type events. We can see that the weekend before all type events had higher sales than average daily sales. Religious activities had the biggest impact on sales the previous weekend while national events had the least impact.
In Figure 16, we can see that events do affect sales, as almost all weekend sales are above average. And the highest sales occurred in the weekend before the Easter, followed closely by EidAlAdha, which are both religious events. Also, the weekend before New Year's is the slowest time for selling.

3.1.3 Summary

From above exploratory analysis, we assumed that these factors below will affect sales.
(1)Day Type: Sales on weekend are more than sales on weekdays.
(2)Events / Holidays: customers' purchase behavior may change according to events and holidays. For example, on Christmas, there is no sales. Although it is not caused by customers themselves, they can not purchase because shop is closed.
(3)Product Price: product price affects sales.
(4)Product Category: product type has a great impact on sales. Compared with food, which is necessary need, some hobbies or household purchased can be reduced.

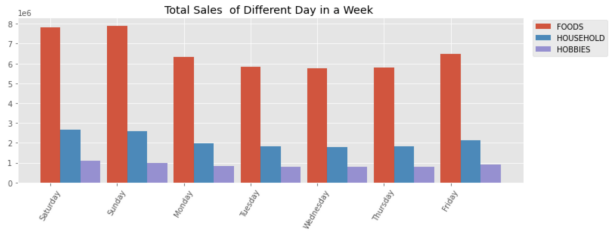

Figure 9: Category Sales

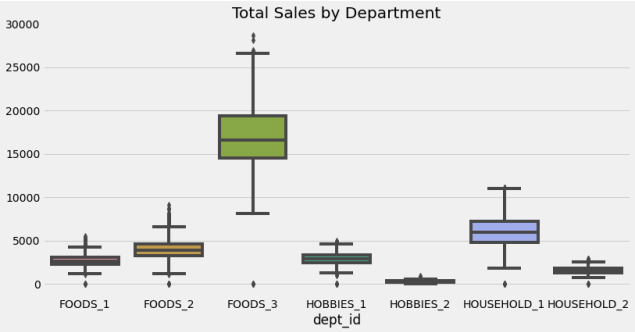

Figure 10: Total Sales of Different Department

(5)Location: location also plays an important role in sales. In states such as California, consumers may buy what they want regardless of price, while consumers in another region may be price sensitive.
After we build our model, we can check whether our assumptions are true.

3.2 Model building

3.2.1 Pre-Processing

(1)Memory usage reduction
Although contrast with XGBT, LightGBM costs less time and memory. To avoid memory spikes, we need to reduce the memory usage before we melt any datasets.
For each datatype, including integer,float,date and other string types, we judged each data type and memory usage, then defined its type. For integer data, it can be divided into int8, int16, int32 and int64. Float data is the same. For date data, we limited its format to 'year-month-day'. Lastly, all the other string data are define to 'category' type.

(2)Melt the train dataset
Through melting, a new column "d" was added, which included dates data. And another added column was "sales", which shows each particular day's sales.

(3)Merge the datasets
Apart from sales data, we also have calendar and price data, so we did a combination of them.

(4)Prepare for training
Convert the object data type to category data types and fill all sales' Null values to 0.

(5)Lag and rolling feature creating
As LightGBM is tree-based, it just splits according to features to the leaf node with its target values in training set. We intended to add lags and rolling mean features to enable it better deal with time series focasting, learn the trend and add the trend to the forecast value.

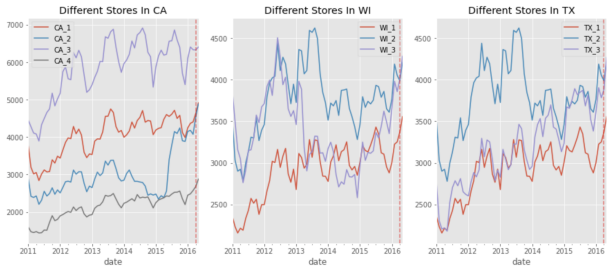

Figure 11: Monthly Average Sales in Different State

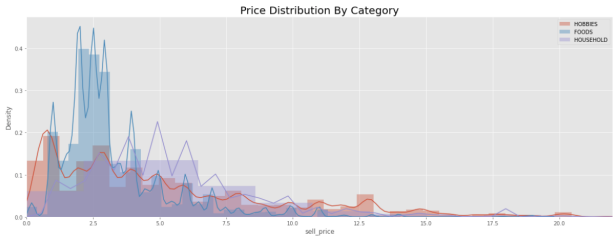

Figure 12: The Distribution of Sell Price

We first created a lag feature quantity for each item. Lag feature is a classical method to transform time series prediction problem into supervised learning problem.A lag is a fixed amount of passing time. If we make a 1 lag change and train a model with this new feature, the model can predict a forward step and observe the current state of the sequence. Increasing the lag, for example, to 28, will allow the model to predict 28 steps in advance. Based on this lag feature, we created a rolling mean feature. If we make rolling 7 mean, it computes the moving average in the previous 7 days.
In our first try, we set lag range in [1,2,3,7,14,28] and rolling window size in [7, 14, 28, 84] When we finished our model training. From feature importance, we found that features correlated with lag 3 and 28, features correlated with rolling 28 and 84 did not show great importance, so we remove them.
Finally, lag in range [1,2,7,14] and rolling window size in range [7,14] was decided. And we totally had 12 new features.

3.2.2 Parameters Tuning

We use LightGBM regressor model to train data and use hyperopt to get the best hyperparameters setting for the model.Table 2 shows the best parameter group we got.

And we chose RMSE as our evaluation function. When validation scores don't improve for 2 rounds, training will stop.

3.3 Model Result

3.3.1 Feature Importance

Figures 17, 18 and 19 show the feature importance of our trained model from three categories. We can find that all lag

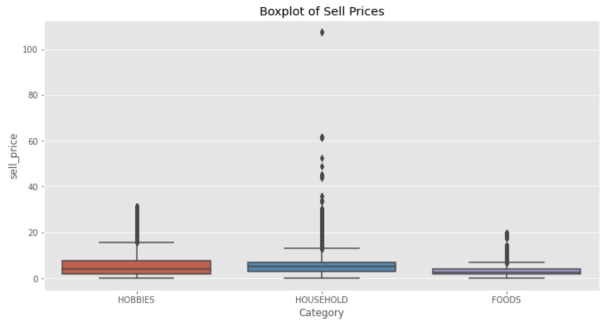

Figure 13: The Boxplot of Price for Different Categories

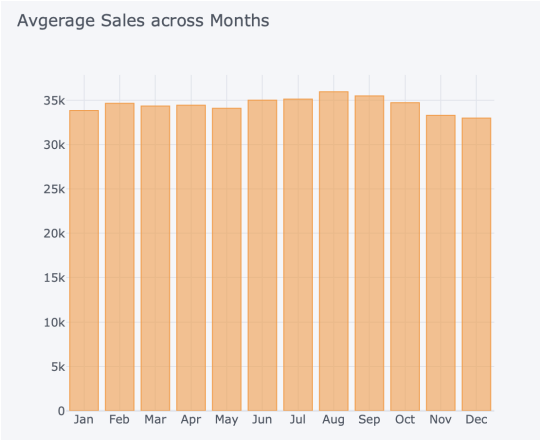

Figure 14: Total Sales Per Month

features have large importance in three models. And 'lag 7 rolling mean 7' has the largest importance among all rolling features.

There are some small difference among them. For hobbies sales prediction, sell price has much greater influence than another two models. The following important features are store and state. For household and food sales prediction, apart from lag variables, price , store and wday show importance in predicting. Contrast with state' great importance in hobbies category, state shows less importance in these two categories. The difference is that wday plays an more important role in food sales prediction.

3.3.2 Kaggle Score

We submitted our output file to kaggle, Table 3 shows our score. The score shows that our model has a good performance on public dataset. However, on private dataset, it does not have the same performance. Although we tried removing some lag and rolling features, it did not improve obviously.

4. CONCLUSIONS

4.1 Conclusions

As we have observed in exploratory data analysis, price plays an important role in sales. Also, the day type. The model feature importance helps us demonstrate them. Meanwhile, the 'wday' feature which represents the day type also shows

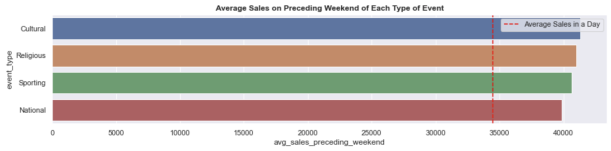

Figure 15: Impact of Different Event

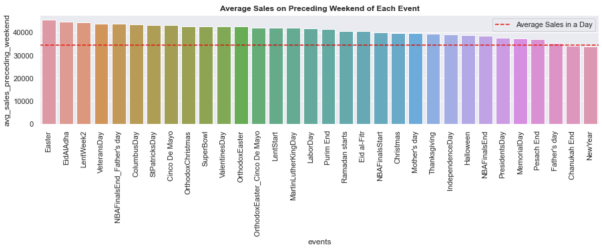

Figure 16: Impact of Different Event

importance. It accords with the analysis before: weekend sales are higher than weekday's. Before we assumed that location and product category affects sales, through model building, we go further that location's influence shows on hobbies category more, which is not necessities of life. As for our created features, lag and rolling, all lag features show great importance, the shorter the day gap, the greater importance. And rolling 7 mean sales feature is helpful to predict future sales.

The advantage of our model is that as long as there are enough weak classifiers, the greedy algorithm of LightGBM itself will find the correlation between input and output as much as possible, which is very suitable for grasping the data relationship between nonlinear input and output, and can well fit the fluctuation characteristics within the value. AS the data we use in this project is a time series, which requires full consideration of seasonality, periodicity and trend , our feature engineering helps detect the trend term of time series. Recursive prediction uses the predicted value as input for feature engineering to predict the next day, and repeat this 28 times. Because in the recursive process, the trend term of time series will be integrated into the input process, and the algorithm is constantly corrected.

4.2 Future Improvements

(1) Consider External Factors

Since the data set we use for feature engineering contains limited influencing factors, many external factors, such as the impact of natural disasters on sales, cannot be taken into account. If the time of natural disasters such as floods and tornadoes that occurred in the three states in these years can be obtained, we can determine the degree of their influence, then the final data can be multiplied by an insurance coefficient to realize model optimization.

At the same time, the core idea of the LightGBM model we used is to pre-sort all features according to the value of features, find an optimal value segmentation point on features when traversing segmentation points, and then split the data into left and right leaf nodes and extend them. Fi-

Table 2: Best parameters

Parameters	Value
$n_estimators$	1000
$learning_rate$	0.3
max_depth	8
num_leaves	50
subsample	0.8
$colsample_bytree$	0.8
min_child_weight	300

nally, a large number of weak classifiers search integration and provide results via voting. If we can take account of more external factors, the prediction result can be improved.

(2) Smaller Predict Level
In our model, we predict sales at category level. What we can do is going further into other levels. We can aggregate sales at store level or department level.

(3) Deal with lag and rolling variables
Although adding these variables helps our model perform better, the model does not perform well on kaggle's private dataset. We may need to look for a better way to deal with them.

Table 3: Kaggle Score

Private score	Public score
4.07945	0.32097

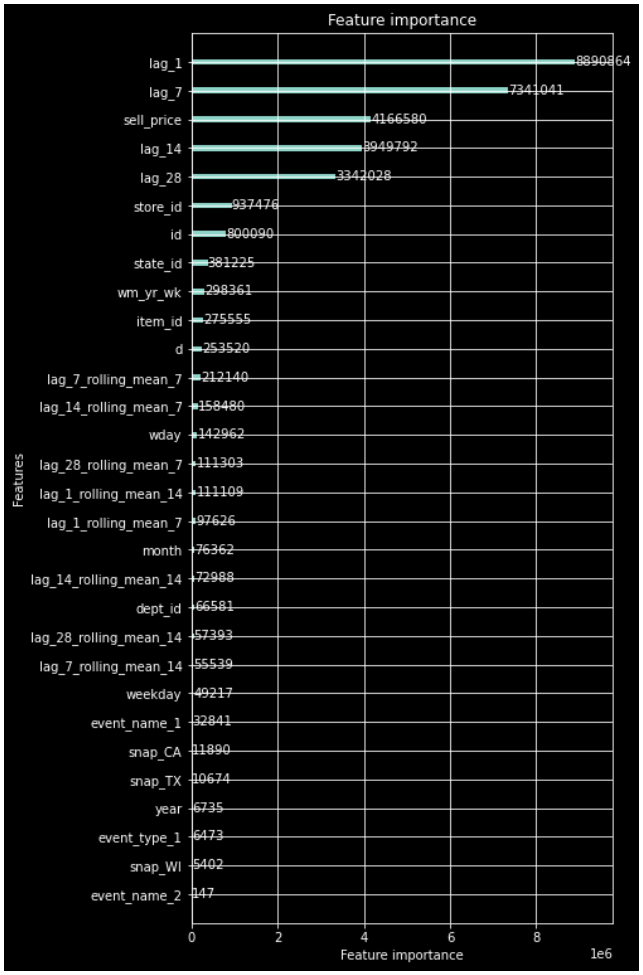

Figure 17: Hobbies Feature Importance

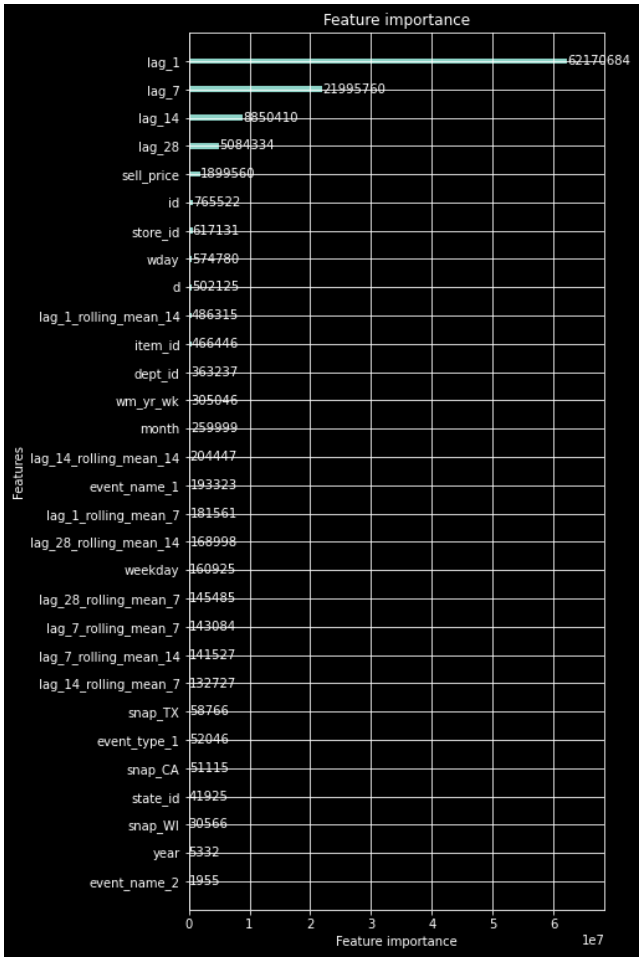

Figure 18: Household Feature Importance

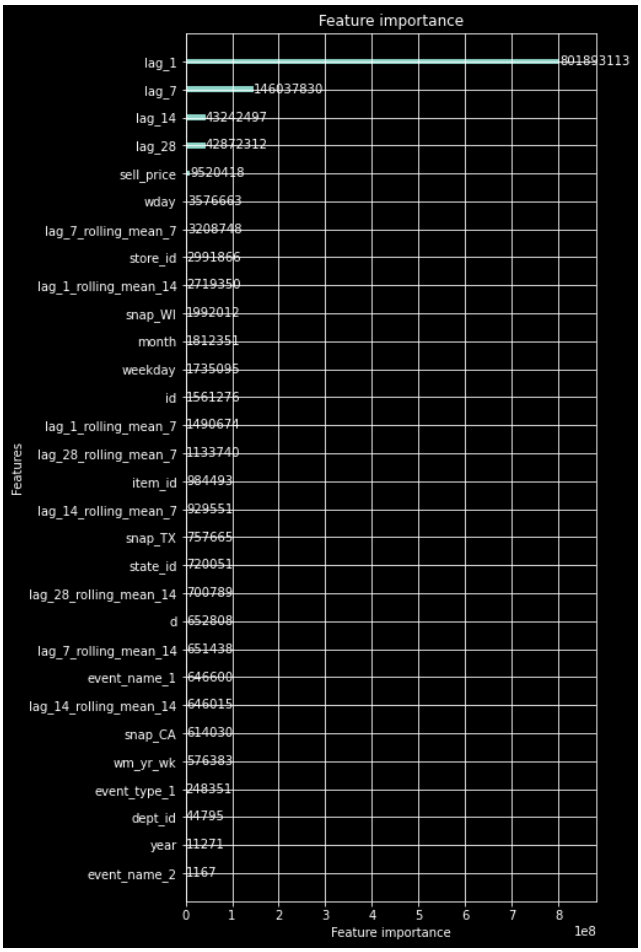

Figure 19: Food Feature Importance

5. REFERENCES

[1] https://www.kaggle.com/code/jagdmir/m5-forecasting-part-1-exploratory-data-analysis?scriptVersionId=55678276

[2] https://xw.qq.com/amphtml/20220303A01NGO00

[3] https://www.kaggle.com/code/jagdmir/m5-forecasting-part-two-lgbm-regressor#Model-Building---LGBM

Each member's contribution to this project is as follows.

Table 4: Division of Labour

Work	Li Muxiao	Qiu Qingqing
Preparation	Study theoretical knowledge, discuss and select models	
Programming	Data Exploration Analysis	LGBM Model
Report	INTRODUCTION;THEORY;Data Analysis	Model Building;Model Result;Conclusions