M5 Forecast - Accuracy:
Estimate the Unit Sales of Walmart Retail Goods

Group members:
MA Rongyue 20826086
NI Xiaohan 20825846
Peng Junkai 20756772
YE Mengxiang 20799762

December 12th, 2021

Abstract

We try to predict item sales of the largest retail company in the world, Walmart, at stores in three states in US during two periods, and each period lasts 28 days. We used four datasets: calendar.csv, sales_train_validation.csv, sample_submission.csv and sell_prices.csv. We took several steps for Data Preprocessing including handling null values, reducing memory usage, spliting data and encoding.

In the Exploratory Data Analysis part, we plot the total demand over time for each type, and then deduce that the highest sales among all three categories is from FOODS, followed by sales of HOUSEHOLD which are still quite a bit above those of HOBBIES. Besides, from the distributions of sale prices of these three types of items, we could know that the distributions of FOOD and HOUSEHOLD obeyed are similar to the right-skewed distribution, while the distribution of HOBBIES is disordered.

In the section of Feature Engineering, we create new features to the existing data set. The new features can be classified into three categories – Rolling demand feature, Price feature, and time feature. Besides, we are using lag-days and roll means to improve the model.

In the model part, we used Light GBM and K-fold Cross-Validation to train the model. We choose Light GBM, because it, as a gradient boosting method based on decision tree, has the advantage of fast training speed and high efficiency. Besides, we follow some steps during the parameter tuning process for Light GBM. Meanwhile, we also set early stopping parameters to avoid overfitting. Then from the feature importance, we could know that the top five features are Item ID, Week, Sell price, Rolling Mean t7, and Rolling skew t30.

On the Kaggle platform, we achieve a public score of 0.77184 and a private score of 5.39065. The exaggerated private score results from the unused sales_train_evaluation data.

1 Introduction

1.1 Background

A lot of department stores like Walmart have a lot of products and also they need to do money transactions on a daily basis. Therefore, it is vital for these stores to get the highest profits by making more accurate prediction for different items, and also keeping a balance between inventory and customer.

We want to seek a more simplified model to make sales prediction of different items according to the historical sales record. But it is necessary to get a lot of extra information including product and customer behavior analysis for sales prediction of previous studies. So it is not appropriate for us.

The aim of the M5 Accuracy competition was to forecast daily sales for the next 28 days and make uncertainty estimates for these forecasts by the hierarchical unit sales of Walmart.

The competition offered data from stores in California, Texas, and Wisconsin in the United States, 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 dataset can be used to improve forecasting accuracy.

Different from the previous M4 competition, the M5 was designed and conducted with the aim of addressing the majority of concerns raised while extending its achievements into several directions. The competition was hosted by Kaggle, which offers large online community for data scientists to compete and provide solutions for various tasks, including forecasting.

1.2 Overview of Dataset

In this competition, we try to make a prediction on sales of different items at stores in different locations during two periods, and each period lasts 28 days. We used four datasets here: The first one is called calendar.csv, and this dataset tells some information about the exact dates the products are sold. The second one is called sales_train_validation.csv, which contains the historical daily unit sales data per product and store. The third dataset is called sample_submission.csv, and it is the correct format for submissions. The forth dataset is called sell_prices.csv, and this dataset contains information about the price of the products sold per store and date.

The data downloaded have been primarily cleaned. Following, we can process them to make them more suitable and comprehensive for building and training Machine Learning models.

There are several steps for Data Preprocessing:

(1) Handling null values

(2) Reducing memory usage

(3) Splitting data

(4) Encoding

2 Data Preprocessing

Before examining different kinds of analysis, the crucial preliminary steps are to conduct data preprocessing. Admittedly, this is also the most time-consuming part for us during the project.

2.1 Memory Usage Reduction

Memory capacity is the biggest bottleneck that we encountered when participating in the featured competition. Because the time series data needs to be reconstructed, the original data capacity is increased. Therefore, many algorithms and tricks can't show their strength on this data. From the perspective of memory, Light GBM is almost the best algorithm, and its memory consumption and running time have been greatly optimized. Light GBM also supports batch training. Only part of the data is needed for each training, which saves memory and running time while ensuring accuracy. In addition, garbage collection and changing the number of bytes in Python can also greatly reduce memory. For example, convert 64 bit floating-point numbers to 32 bits, and so on.

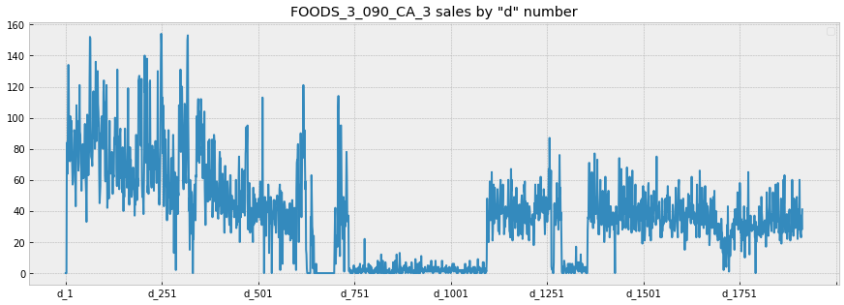
When the program becomes larger, larger memory usage can significantly slow down your program. In order to avoid system hardware calculation problem, one easy way is to reduce memory usage and speed up programs.

The way to reduce the memory usage is using certain function whose main principle is to use smaller int(float) 32/16/8 for int 64/float 64 type values, especially for data with a large number of digital types.

2.2 Spliting Data

We don't use the data offered, and we try to separate the validation and evaluation in submission as the test data we used later. And we change column names to the next 28 days in test 1 and test 2. d_1914 - d_1941 represents the validation rows which we will predict in stage 1, and d_1942 - d_1969 represents the evaluation rows which we will predict for the final competition standings. We try to forecast sales for 28 forecast days. The columns represent 28 forecast days, and we fill these forecast days with our predictions. Each row represents a specific item, and the id tells us the item type, state, and store. However, we don't know what these items are exactly.

We could visualize the data for a single item as the figure.



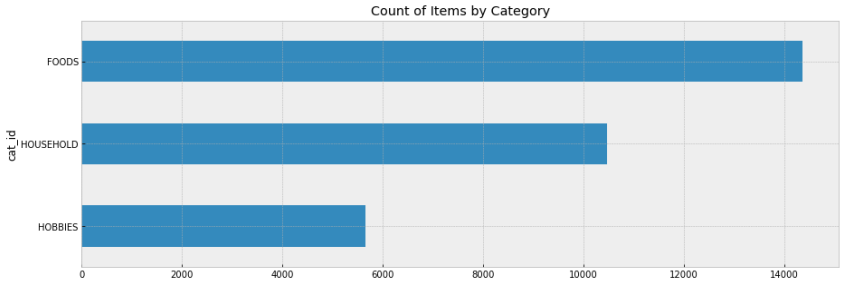
2.3 Encoding

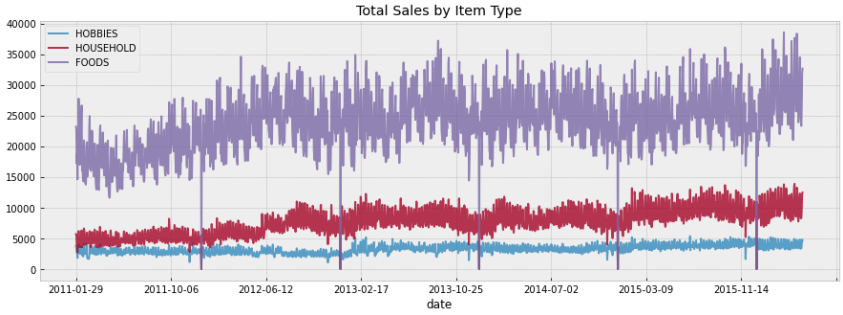
First, we fill the NaN data of category features. We list out the NaN columns, and we replace it with "unknown". Then, before we go any further, we need to transform the category features into encoding features. Label encoding is an effective way and it is the process of assigning each unique category in a categorical variable with an integer, and no new columns are created.

3 Exploratory Data Analysis

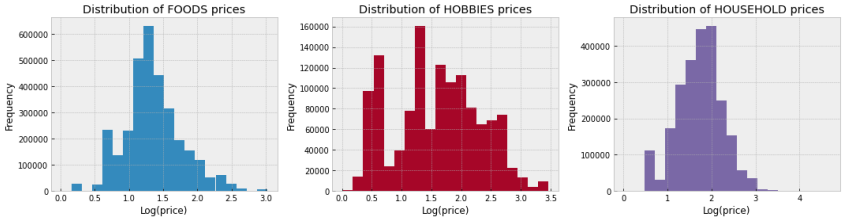
The objective of this part: Exploratory Data Analysis, which is also called EDA, is an open-ended process, we use this method to calculate statistics and make use of these results to get figures in order to find some certain relationships, possible trends, anomalies and staff like that. We use EDA to get some results and similarity from the data. Usually, we start with make an overview in a high level, and then try to view it in more specific and narrow areas. The findings may be interesting in their own right, or they can be used to inform our modeling choices, such as by helping us decide which features to use.

Then we could combine sales over time by type. We have several item types including FOODS, HOUSEHOLD and HOBBIES. We could plot the total demand over time for each type, and could also see from the figure below that sales of FOODS are highest among all three categories, followed by HOUSEHOLD which are still quite a bit above HOBBIES, and sales of HOBBIES are the lowest among all three categories.





Besides, we could explore the distributions of sale prices of different types of items, whose historical sale prices are given. From the distributions of sale prices of these three types of items, we could know that the distributions of FOOD and HOUSEHOLD obeyed are similar to the right skewed distribution, and the distribution of HOBBIES is disordered.



4 Data Engineering

In the section of Feature Engineering, we create new features to the existing data set.

In order to enhance the accuracy of the data, the specific data point is expanded to an interval, which is the so-called window for further judgment.

We are using lag-days and rolling means to improve the model. To be more detailed, new rolling mean, standard deviation, skewness, and kurtosis are introduced for one week, one month, two months, and three months lag variables.

The new features can be classified into three categories – Rolling demand feature, Price feature, and Time feature. And details are as follows:

Table 1: Summary of New Features

Category	**New features**
Rolling demand feature	lag_t28, rolling_mean_t7/30/60/90, rolling_std_t7/30, rolling_skew_t30, rolling_kurt_t30
Price feature	price_change_t1, price_change_t365, rolling_price_std_t7, rolling_price_std_t30
Time feature	date, year, month, week, day, day of week

5 Model Training

5.1 Light GBM

The method we choose is Light GBM, because it, as a gradient boosting method based on leaf-wise algorithm, has the advantage of fast training speed and high efficiency. Meanwhile, Light GBM also supports efficient parallel training.

5.2 K-fold Cross-Validation

Compared with depth-wise growth, Light GBM can converge much efficiently. However, we may easily encounter the problem of over-fitting. In order to derive a more accurate estimate, we conduct 3-fold Cross-Validation, predict with each fold and take the average. The advantage of K-fold cross-validation is to flag over-fitting and ensure all data are used for training and validation.

3-fold cross-validation is a re-sampling procedure that the initial data set is divided randomly into three sub-samples which is normally equal-sized. Of the three sub-samples, a single sub-sample is treated as the validation data for testing the model, and the remaining two sub-samples are used as training data. The cross-validation process is repeated three times, which means each fold has the opportunity to be used exactly once for validation. At last, the three results can then be averaged to produce a single estimation.

5.3 Model Parameter

During the parameter tuning process for training model, choosing the suitable values of num_iterations and learning_rate is the critical step, and the choice of values varies widely according to different data set and objective.

First of all, a higher learning rate was used to make the convergence faster, but the accuracy is definitely not as good as in the long run. At last, we use a lower learning rate and more decision trees to train the data to optimize the score further.

We follow the below steps during the parameter tuning process for Light GBM.

Step 1: Choose a higher learning rate to speed up the speed of convergence.
Step 2: Adjust the basic parameters of the decision tree
Step 3: Regularization parameter tuning
Step 4: Reduce the learning rate to enhance the accuracy

Meanwhile, we also set early stopping parameters to avoid over-fitting. Early stopping means the training will stop when the validation performance is not improving after the last early stopping round.

After tuning, we get the following parameters.

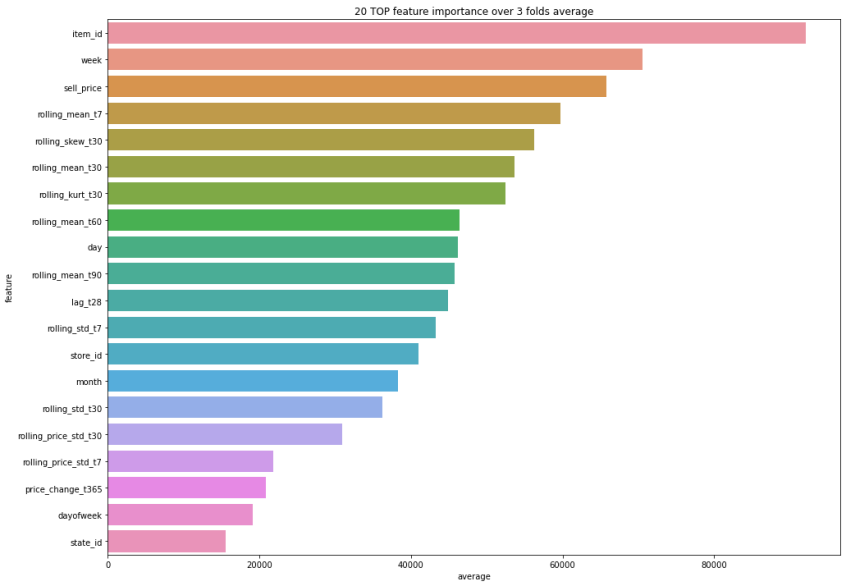
Table 2: Summary of Model Parameters

Parameter name	**Desciption**	**Value**
num_leaves	Main parameter to manage the complexity of the tree model	555
min_child_weight	To deal with overfitting	0.034
feature_fraction	Decrease feature_fraction to reduce training time	0.379
bagging_fraction	To control the size of the sample. Decrease bagging_fraction to reduce training time	0.418
min_data_in_leaf	To prevent over-fitting in a leaf-wise tree	106
objective	Regression by default and we are predicting numerical results	Regression
max_depth	To limit the tree depth explicitly	-1
learning_rate	To impact the training accuracy	0.007
boosting_type	GBDT by default	gbdt
metric	We choose Root square loss as indicated in the competition	rmse
reg_alpha	Parameter for regression application	0.3899
reg_lambda	Parameter for regression application	0.648
num_boost_round	Number of boosting iterations	2500
early_stopping_rounds	Stop training if the metric of data does not improve in last rounds	50

5.4 Feature Importance

We then check the feature importance returned from Light GBM by plotting the feature importance graph. The top five features are Item ID, Week, Sell price, Rolling Mean t7, and Rolling skew t30.

This figure below shows the resultant importance of the top-20 characteristics in the Light GBM.



6 Performance Measure

6.1 Root Mean Square Error (RMSE)

In order to simplify the training process, we use RMSE as our measure matrix. Root Mean Square Error (RMSE) calculates the standard deviation of the residuals, which is also known as prediction errors.

$$|q_t| = \frac{Y_t - F_t}{\frac{1}{n-1}\sum_{i=2}^{n}|Y_i - Y_{i-1}|}$$

$$MASE = \frac{1}{h}\sum_{t=n+1}^{n+h}(|q_t|)$$

6.2 Weighted Root Mean Squared Scaled Error (WRMSSE)

As indicated, this competition applies Weighted Root Mean Squared Scaled Error (RMSSE) as the performance measure matrix.

The RMSSE metric is a variant of the original MASE metric aiming to get a scale-free error to compare forecasts across series with different scales effectively.

The measure is calculated for each series as follows:

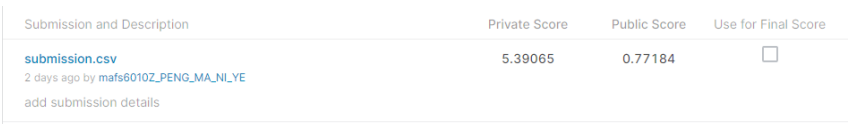
$$|sq_t| = \frac{(Y_t - F_t)^2}{\frac{1}{n-1}\sum_{i=2}^{n}|(Y_i - Y_{i-1})^2|}$$

$$RMSSE = \sqrt{\frac{1}{h}\sum_{t=n+1}^{n+h}(|sq_t|)}$$

6.3 Kaggle score

On the Kaggle platform, we achieve a public score of 0.77184 and a private score of 5.39065. The exaggerated private score results from the unused sales_train_evaluation data. The reason why is that we want to keep consistent with the data source in Kaggle's competition. When the competition was in progress, the evaluation part, known as the public score label part, did not post publicly. After the competition ended, some teams considered using it as another baseline to train a new model. Since the training result is perfect, the public score (WRMSE) would be zero if you upload it into Kaggle.

Looking back at our final score, we would better ignore our private score since we decided to use K-fold cross-validation to evaluate the model rather than the evaluation data we were supposed to use. Our goal is to optimize the public score on the Kaggle website.



7 Conclusion

The project helps us become more skilled in parameter tuning for the Light GBM model with K-fold cross-validation, and also verifies the effectiveness of Light GBM under various circumstances.

Concerning take-aways, the part that left us the most profound impression is the part of memory usage reduction, which is relatively new for us. Meanwhile, we gain a rough picture of people's habit of shopping in the supermarket through data visualization, which encourages us to think of machine learning methods to solve real-time business problems.

At last, we believe that we can have a better prediction and a lower error when using more supplementary datasets for further training. And we may try to apply Neural networks and LSTM in further exploration.

References

[1] Rana, D. (2021, May 31). M5 forecasting-accuracy. Medium. Retrieved December 11, 2021, from https://dipanshurana.medium.com/m5-forecasting-accuracy-1b5a10218fcf.

[2] Robikscube. (2020, March 27). M5 forecasting - starter data exploration. Kaggle. Retrieved December 11, 2021, from https://www.kaggle.com/robikscube/m5-forecasting-starter-data-exploration.

[3] Hewamalage, Hansika Montero-Manso, Pablo Bergmeir, Christoph Hyndman, Rob. (2021). A Look at the Evaluation Setup of the M5 Forecasting Competition.

Group Work Contribution

We all actively participated in the initial discussion for the overall structure and ideas.

After the outline is settled, PENG Junkai and NI Xiaohan are mainly responsible for coding, and MA Rongyue and YE Mengxiang are mainly responsible for report writing, slide making and video recording. Moreover, we also support and conduct cross-checking for others' parts and give suggestions.