M5 Forecasting – Accuracy

MATH6010Z Project3
LOU Ruoyu (20743763)
CHEN Yuying (20744353)
SHANG Zhiheng (20738938)
Department of Financial Mathematics
Hong Kong University of Science and Technology
Dec. 12, 2021

Workload

LOU Ruoyu: Data processing and model building
CHEN Yuying: Refinement and report drafting
SHANG Zhiheng: Data analysis and report drafting

Presentation link: *https://www.bilibili.com/video/BV1sS4y1Q7av/*

Abstract

In this report, we aim to estimate the point forecasts of the unit sales of various products sold in the USA by Walmart. Given datasets involving unit sales of 3049 products sold in three states of the USA, explanatory data analysis and feature engineering is implemented. In model fitting process, CatBoost model is finally chosen to conduct the estimation. Explanation of result and following discussion also contained in this report.

1 Introduction

In this Kaggle competition we aim to estimate the point forecasts of the unit sales of various products sold in the USA by Walmart. We are given four datasets involving the unit sales of 3049 products sold in three states of the USA (California (CA), Texas (TX), and Wisconsin (WI)), which are organized in the form of 42,840 hierarchical time series, the products are divided into 3 product categories (hobby, food and family) and 7 product sectors. The dataset structure can be observes from the overview picture of original dataset organization[1].

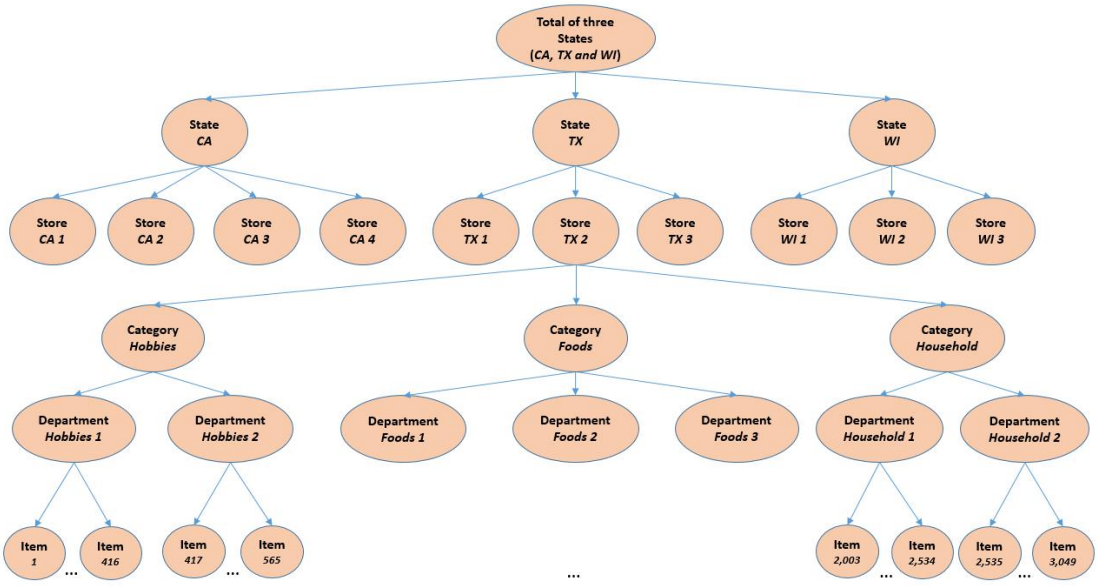

Figure 1: Overview picture of original dataset organization

In order to find possible patterns and corresponding relationships in different level, several explanatory data analysis (EDA) strategies are applied: Boxplot, Time Series and Calendar Heat Maps[2].

In data processing part, we first deal with the calendar file and then merge it with the two sales files and do the future processing as well as the feature engineering.

The Catboost algorithm is employed for training and forecasting. The point forecast submission are being evaluated using the Root Mean Squared Scaled Error (RMSSE), which is derived from the Mean Absolute Scaled Error (MASE) that was designed to be scale invariant and symmetric.

2 Explanatory Data Analysis

From the overview picture of original dataset organization[1], we can find that the data is highly hierarchical and can be classified in several level: State, Store, Category, Department and Item. In order to find possible patterns in different level and corresponding relationships, several explanatory data analysis (EDA) strategies are applied: Boxplot, Time Series and Calendar Heat Maps[2].

2.1 Boxplot

First of all, through the comparison of Total Sales in highest level: State, we find that althrough California has the higher total sales, while there is no similar pattern in mean sales. So the reason that CA has higher total sales may possible because there are more stores in this state.

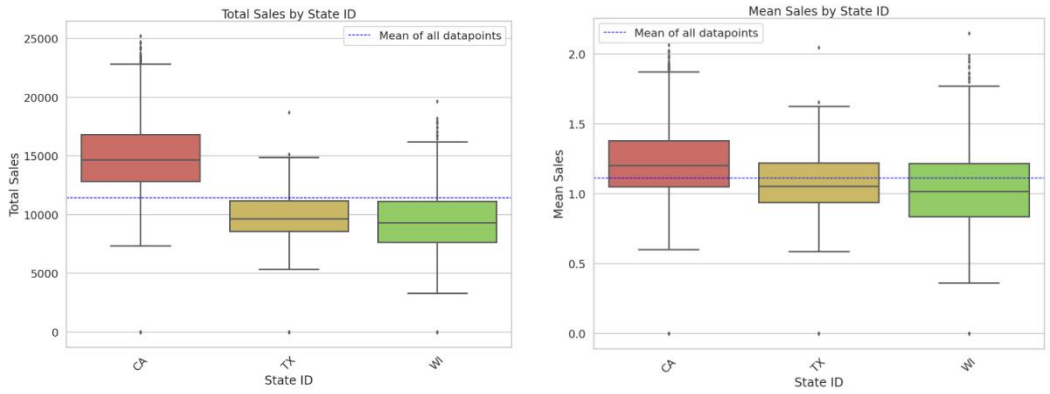

Figure 2: Total and mean sales by state

When it comes to next level, says store, we find CA_3 store has the highest total sales and mean sales, while CA_4 store is relatively low. So the high performance of CA_3 store may also contributes to the total sales value in CA states.

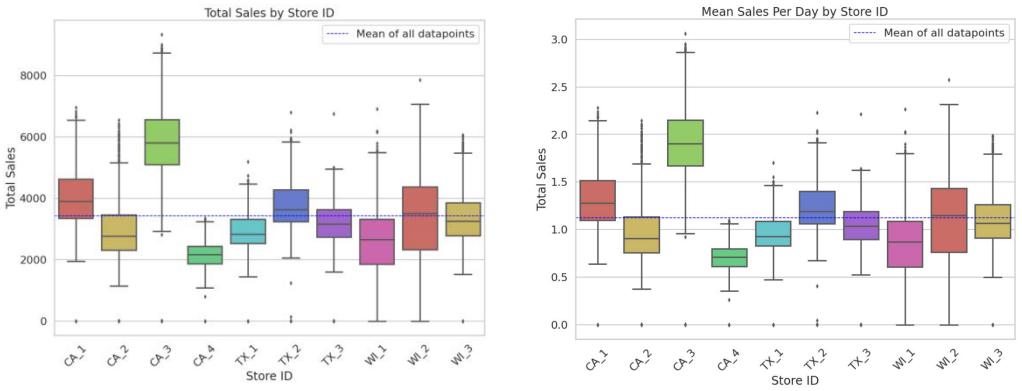

Figure 3: Total and mean sales by store

Then we focus on department and category level. Here we find that among all categories, FOODS has relatively high total sale score, and particularly, the FOODS_3 is on a high total sale position. So the changes of FOODS category with time may have high influence on the total sale performance, which will be discussed later.

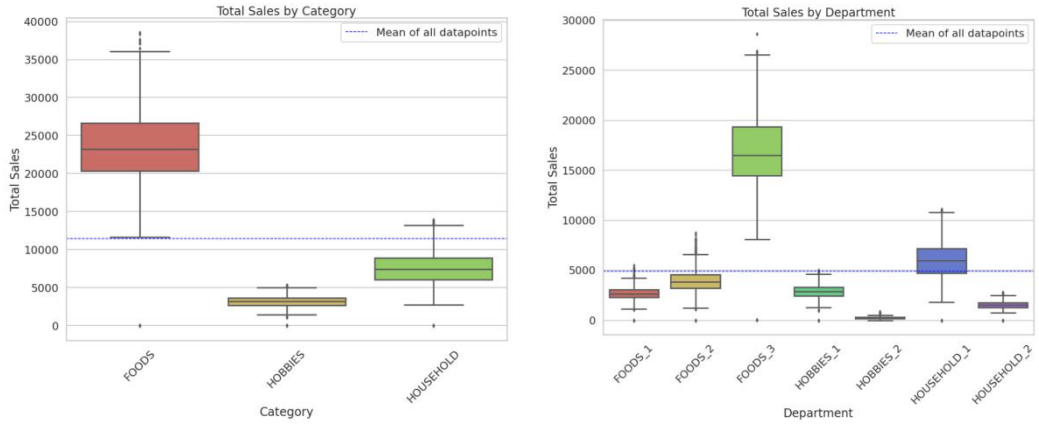

Figure 4: Total and mean sales in category and department

2.2 Time Series

In order to find the relationship between total sales and time, a total sales by item type is plotted. As shown in the figure below, food has long been highest position of total sales, then followed by household and then hobbies. The unique sales peaks also found in this picture, which means people tend to buy more food in particular time of the year.

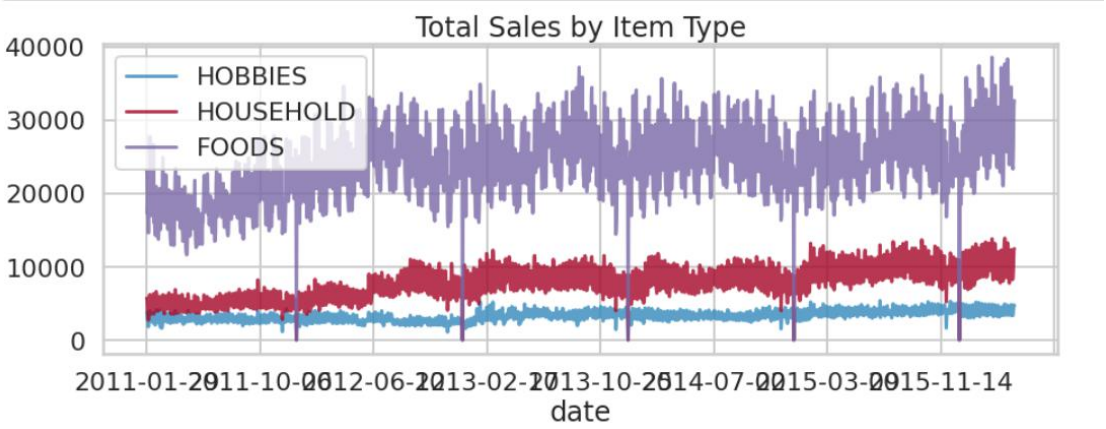

Figure 5: Total sales by item type

2.3 Calendar Heat Maps

From the heatmaps shown below, weekends are the most popular shopping days for customers among all item categories. While FOODS category tended to take the highest position of sales amount and then experience a long decrease in each month. HOUSEHOLD and HOBBY items sold much less in January, which indicates that people tend not to buy those things at the beginning of each year

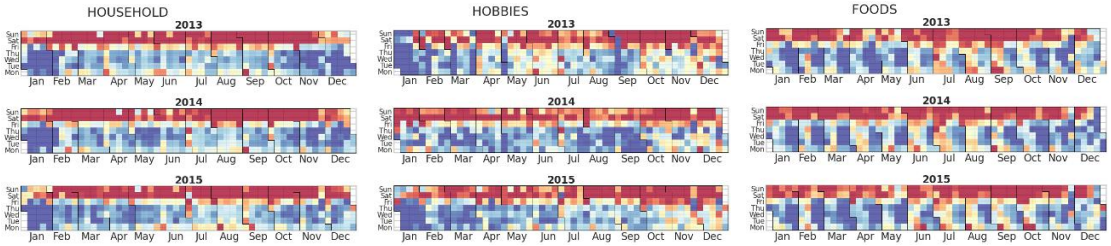

Figure 6: Calendar heat maps in household, hobbies and department

3 Data Processing and Feature Engineering

There are three files including “calendar.csv”, “sales_evaluation.csv”, and “sell_prices.csv”. We first deal with the calendar file and then merge it with the two sales files for the future processing.

3.1 Deal with the calendar file

We need to reduce the memory usage first. Since the final datasets would be large, this operation will be carried out through the whole process. Changing the types of the string variables and restrict the precision of the numeric variables would be helpful. Specifically, for the category variables like “d” and “wm_yr_wk”, we

transform their types from string to category. For the numeric variables like “day of week” and “day of month” which we get from the original variable “date”, their precision are restricted to be int8. For the binary variable including “snap_CA”, ”snap_TX”, and ”snap_WI” which indicates whether the stores of CA, TX or WI allow SNAP purchases on the examined date, their types are transformed into bool.

Some other new variables are created as well. We combined the variables “event name 1” and “event name 2” to a comprehensive variable named “event name”. The variables “event_type 1” and “event type 2” are combined to another variable named “event_type”. These two variables contains the information about the event situations regarding a certain day. Then the missing values of these two variables are filled with “None”. Since they are categorical variables, we map them into a series of integer according to the ranking of their occurrences. Apart from these two variables, we also create a variable named “days_to_event” indicating the number of days from the examined date to the next event.

Finally, we drop some variables which seem to be repetitive, including “year”, “weekday”, “month”, “wday”, “event_name_1”, “event_name_2”, “event_type_1” and “event_type_2”, to reduce the memory usage. The processed table are shown below:

	date	wm_yr_wk	d	snap_CA	snap_TX	snap_WI	events_names	events_types	day_of_month	day_of_week	days_to_event
0	2011-01-29	11101	d_1	False	False	False	35	8	29	5	1
1	2011-01-30	11101	d_2	False	False	False	35	8	30	6	1
2	2011-01-31	11101	d_3	False	False	False	35	8	31	0	1
3	2011-02-01	11101	d_4	True	True	False	35	8	1	1	1
4	2011-02-02	11101	d_5	True	False	True	35	8	2	2	1

Table 1: Processed calendar table

3.2Merge the files and Process the whole data

Before future processing, we need to merge calendar file and the two files concerning the sales information. The calendar file and the sell price file are merged according to “wm_yr_wk” then the “sales_evaluation.csv” is merged in according to ”item_id”, “store_id” and “d”.

Reducing the memory usage is considered as well. Here the data type processing is employed again. The type of the category variables including “store_id”, “item_id” and “id” are set to be categorical while that of the numeric variables like “demand”

and “d”(set to be numerical first) are set to be float16 and int16. In addition, reducing the number of variables would be helpful, “snap_TX”, “snap_CA” and “snap_WI” are combined to a new variable named “is_snap_avaliable” indicating whether the SNAP purchases are allowed for the examined sample, then we can drop the original variables.

Some new features can be generated based on the time series to capture the information shown by the dependent variable(demand) itself. First we group the data of dependent variable by “store_id” and “item_id” and set the window to be 29 days, then we create four new features listed below:

variables	meaning
cumulative_mean_demand_29d_ago	the cumulative mean value of the demand after a lag of 29 days for each examined date.
cumulative_md_low_demand_29d_ago	the cumulative median value of the demand after a lag of 29 days for each examined date.
ewm_mean_1w_demand_29d_ago	Exponential moving weighted mean value of the demand after a lag of 29 days at a span equals 7*1. $(\alpha = 2/(span + 1))$
ewm_mean_4w_demand_29d_ago	Exponential moving weighted mean value of the demand after a lag of 29 days at a span equals 7*4.
ewm_mean_8w_demand_29d_ago	Exponential moving weighted mean value of the demand after a lag of 29 days at a span equals 7*8.
rolling_mean_1w_demand_29d_ago	the mean value of the demand after a lag of 29 days in a 7*1 days rolling window.
rolling_mean_4w_demand_29d_ago	the mean value of the demand after a lag of 29 days in a 7*4 days rolling window.
rolling_mean_8w_demand_29d_ago	the mean value of the demand after a lag of 29 days in a 7*8 days rolling window.

Table 2: Newly defined variables and corresponding meanings

The categorical variables still need to be encoded for the future training. Since we will employ Catboost algorithm in the following steps, we use the function “CatBoostEncoder” to encode “store_id”, “item_id” and “dept_id”. As for the “state_id”, three binary variables, “is_in_CA”, “is_in_TX” and “is_in_WI” are employed to replace it. The variable “cat_id” is also dealt with in the same way that we transform it into three new binary variables “is_foods”, “is_household” and “is_hobbies”.

4 Model Fitting

According to the competition guide[1], the accuracy of the point forecasts will be evaluated using the Weighted Root Mean Squared Scaled Error (WRMSSE). The measure is calculated as follows:

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

$$\boldsymbol{WRMSSE} = \sum_{i=1}^{42{,}840} w_i * RMSSE\,,$$

where t is a point in the generated forecast, n is the length of the training sample (number of historical observations), h is the forecasting horizon, and w_i is the weight of the i_{th} series of the competition. A lower WRMSSE score is better.

Based on the result of EDA and feature engineering, the raw data of the competition is highly hierarchical, which means the data can be aggregated on different levels: item level, department level, product category level, and state level. Due to the large number of data features and calculating power limitations, CatBoost is finally chosen as the primary model in this project, since CatBoost converges to a good solution in the shortest time comparing with XGBoost and LightGBM within gradient boosted decision trees (GBDT)[3].

4.1 CatBoost

CatBoost, like XGBoost and LightGBM, is a machine learning algorithm that uses gradient boost on decision trees. CatBoost has advantages such as dealing with categorical data more efficiently than other boosting methods and not requiring conversion of datasets into specific formats.

4.2 Fitting Strategy

In the data splitting process, three categories are set as train part, valid part and test part. The train part contains combined data before date '2015-05-01', valid part contains combined data laying before sales date ’1942’ and after date '2015-05-01', while test part contains those laying after sales date ’1914’. In order to turn numerical features of raw data into Features Data class, the Pool constructor in CatBoost Lib is deployed to achieve the goal.

Some key parameter tuning process is shown below.

Parameter	Description	Value
Learning Rate	This setting is used for reducing the gradient step. It affects the overall time of training: the smaller the value, the more iterations are required for training.	0.1
Depth	In most cases, the optimal depth ranges from 4 to 10. Values in the range from 6 to 10 are recommended.	10
Loss Function	The metric to use in training. The specified value also determines the machine learning problem to solve.	RMSE
Leaf estimation iterations	This parameter defines the rules for calculating leaf values after selecting the tree structures. The default value depends on the training objective and can slow down the training for datasets with a small number of features	1
Evaluation Metric	Standard of evaluation	RMSE

Table 3: Parameter Description of GBDT

In the first fitting round, the 'best Test' score converge to 2.2777 with the increase of iteration steps. When it comes to 900 iteration steps, the Overfitting Detector of CatBoost is triggered due to overfitting tendency. As the result, we get best test score at 2.2777 and corresponding best iteration 625th step. The whole model then shrink to first 626 iterations.

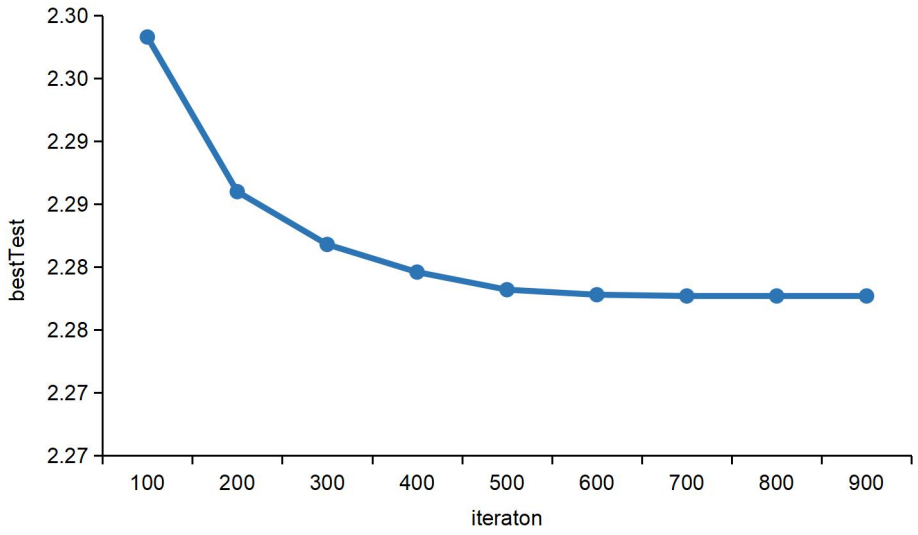

Figure 7: Convergence of model iterations

In the final round fitting, the iteration is set as the product of former best iteration step (625) and train-valid-ratio (ratio of valid data size and train data size). In this

round, the model shrinkage in combination with learning continuation is not implemented, which means the fitting process will finish all iterations set before.

5 Results & Discussion

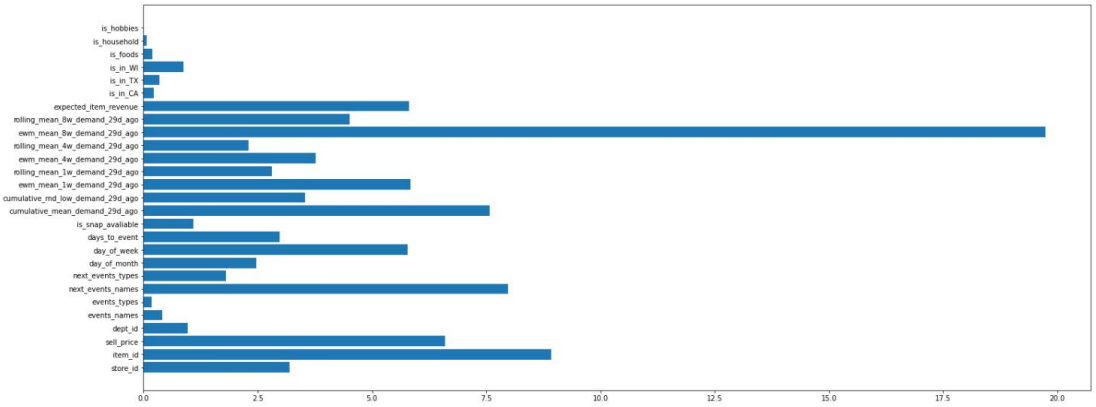

Figure 8: Important features in dataset

The picture above shows the important features in combined dataset. We can find that the feature 'ewm_mean_8w_demand_29d_ago', which means 'exponential moving weighted mean value of the demand after a lag of 29 days at a span equals 7*8' has the highest importance. Among all exponential moving weighted mean value of the demand at a span of 7*X, it can be found that the corresponding feature tends to be more important as X increases. Some other features like item id, next event names, expected item revenue also has influence on the final results.

M5	Competition Notebook	Run	Private Score	Public Score
	M5 Forecasting - Accuracy	19448.2s	0.60280	0.53116

Figure 9: Final score on Kaggle competition

According to the Kaggle competition platform, finally we get 0.603 as private score and 0.5312 as public score.

There are several ways to improve the result. First of all, more GBDT models should be tested based on this competition[4]. Actually, during this project our team have tried other models like LightGBM and XGBoost. However, due to the limitation of calculation resources and time, only CatBoost succeed in finishing the result. Second, although some strategies has been taken in data process part in order to cut down calculation pressure, progress will be made if more effective data conversion methods could be found.

Those aspects above may be the next key factors for us to improve this interesting project.

References

[1]. THE M5 COMPETITION Competitors' Guide

[2]. https://github.com/QingweiMeng1234/Kaggle_M5_Accuracy_Report

[3]. Anghel A , Papandreou N , Parnell T , et al. Benchmarking and Optimization of Gradient Boosting Decision Tree Algorithms[J]. 2018.

[4]. Makridakis S , Spiliotis E , Assimakopoulos V . The M5 Accuracy competition: Results, findings and conclusions[J]. International Journal of Forecasting, 2020, 36(1):224-227.