math 6010-Zhou-Sun-Huang-Tian Project3 Report
Forecast the Sales of Walmart

Zhou Xiaomin 20749212
Sun Ke 20747903
Tian Xinyu 20750015
Huang Yuning 20738524

December 12, 2021

Abstract

In this paper, we first do EDA for the data. Then we use LightGBM and SARIMA methods to predict the sales in the next 28 days of Walmart and use RMSE to evaluate the effectiveness of each method.

1 Data Description

The dataset involves the unit sales of 3049 products of Walmart, classified in 3 product categories (Hobbies, Foods, and Household) and 7 product departments, in which the above-mentioned categories are disaggregated. The products are sold in three States (CA, TX, and WI). Dataset contains 4 parts. 'calendar.csv' contains information about the dates on which the products are sold. 'sales-train-validation.csv' contains the historical daily unit sales data per product and store. 'sell-prices.csv' contains information about the price of the products sold per store and date. 'sales-train-evaluation.csv' includes sales data from day1 to day1941.

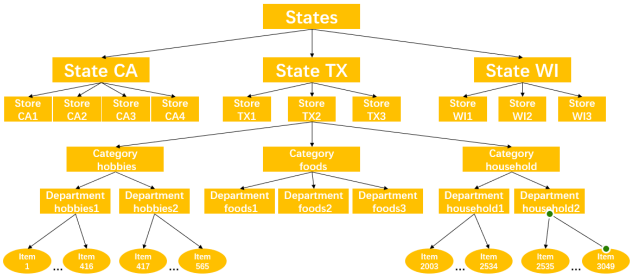


Figure 1: Data description

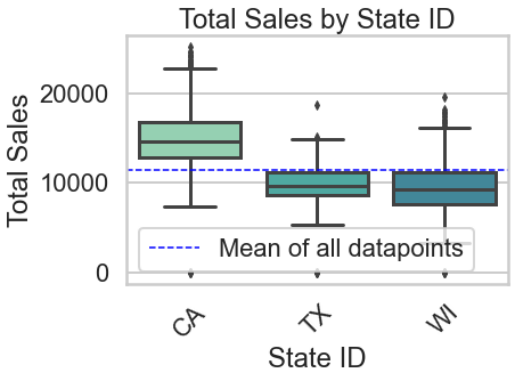
2 EDA

We do EDA to extract the information enfolded in the given data and summarize the main characteristics of it.

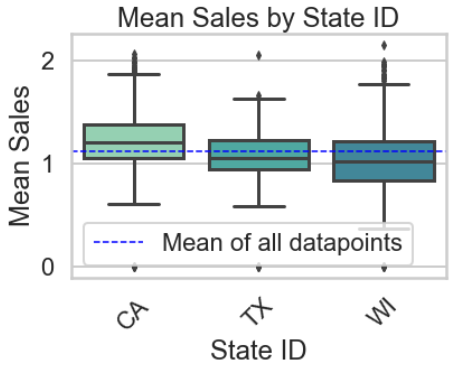
Firstly, we find that the lowest sales day is Dec 25, 2012 with only 11 sales. This may because that Walmarts are closed on Chirstmas day. The highest sales day is Mar 6th, 2016 with 57218 sales. This may because the seventh democratic presidential candidates debate was hold on that day.

2.1 Sales of Different States

We select the sales data of different states from the dataset and analyze them. As we can see in the Figure 2, the total sales in CA are significantly higher than that of TX and WI, while the mean sales of it are not. This may simply because that CA has more stores (4 stores) than TX (3 stores) and WI (3 stores).



(a) Total sales of each state

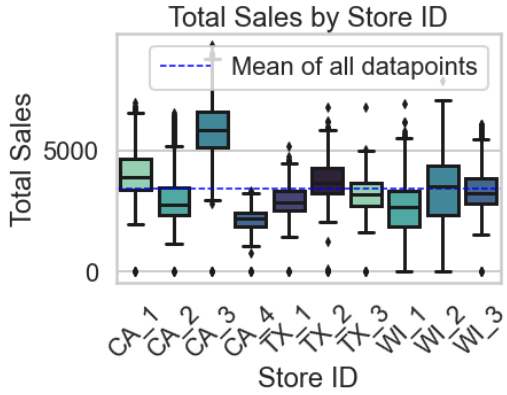


(b) Mean sales of each state

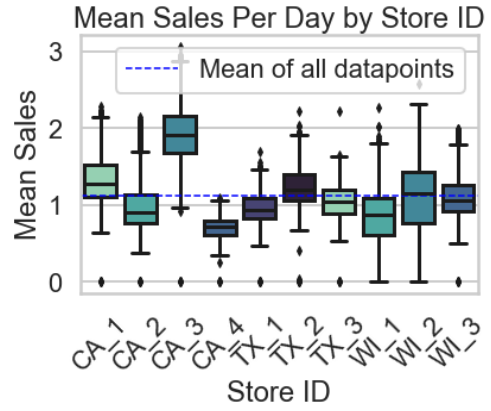
Figure 2: Sales of Different Sates

2.2 Sales of Different Store

According to Figure 2, we can find CA does have more overall mean sales. In order to find the reason, we select the sales data of each store owned by different states. According to figure3, CA_3 has the most sales, CA_2 is similar to other stores and CA_4 has few sales. The reason why CA has more overall mean sales may be because CA_3 is significantly higher than the others.



(a) Total sales of each store



(b) Mean sales of each store

Figure 3: Sales of Different Stores

2.3 90-day Average Total Sales of Stores

We calculate the rolling 90-day average of total sales for each store. As we can see in the Figure 4, some stores have wide fluctuations in average total sales. It is worth noting that sales of CA_2 seem to have fluctuated considerably in 2015 and WI_2 seem to have seen a big increase in 2012 and 2016.

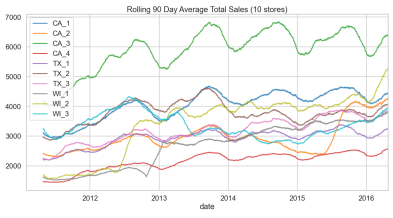


Figure 4: Rolling 90-day Average Total Sales of Stores

2.4 Sales of Different Categories

After analyzing the data according to state and store, we start to analyze the data according to categories. We combined sales over time by 'HOBBIES', 'HOUSEHOLD' and 'FOODS'. Figure 5 shows that FOODS has the most sales, followed by HOUSEHOLD and HOBBIES. According to Figure 6, FOOD sales are higher in the middle of the year and generally decline in the second half of the year; sales of HOUSEHOLD and HOBBIES hit a low in January, which may be because the holiday season is over. In addition to this, for all categories, weekends have more sales than weekdays.

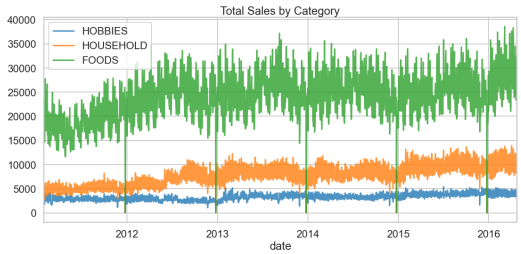
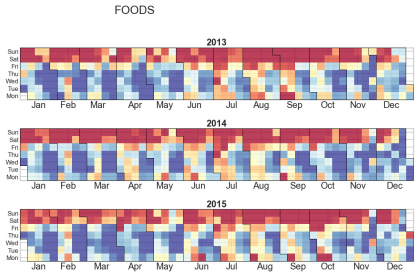
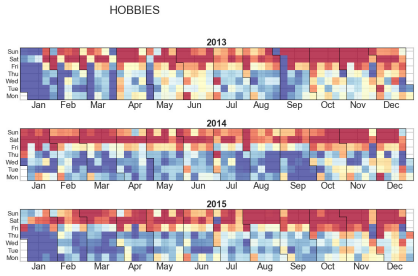


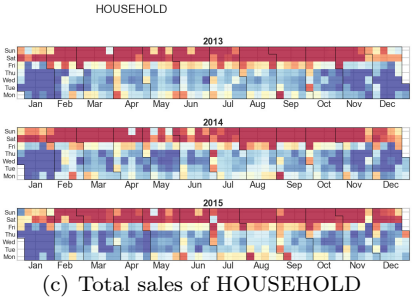
Figure 5: Total of Sales of Different Categories



(a) Total sales of FOODS



(b) Total sales of HOBBIES



(c) Total sales of HOUSEHOLD

Figure 6: Total Sales of Each Categories

2.5 Sales of Different Departments

Then we calculate total sales of different departments. According to Figure 7, FOODS_3 and HOUSEHOLD_1 have the most total sales. In order to check whether the items are evenly distributed among various categories and departments, we calculate the number of items in each category per store. According to Figure 8, all stores have the same kind of items, which means the items are evenly distributed among various categories and departments.

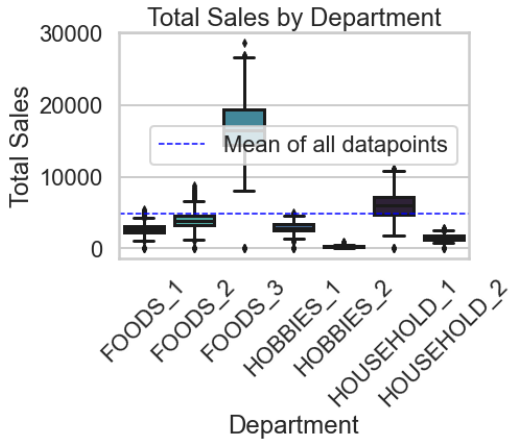


Figure 7: Total Sales of Each Department

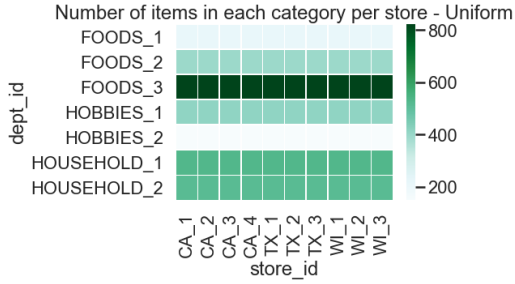


Figure 8: Number of Items in Each Category Per Store

2.6 Price of Sample Item

We choose a sample item and see its sale price over time in different stores. We choose FOODS_3_090 as our example item and analyze the historical sale prices of it. According to Figure 9, the price of this item is growing over time. And at the same time period, different stores have different selling prices.

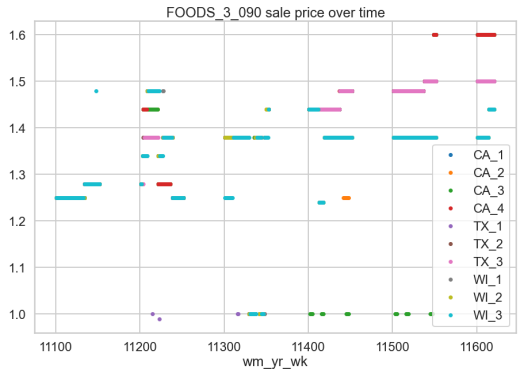
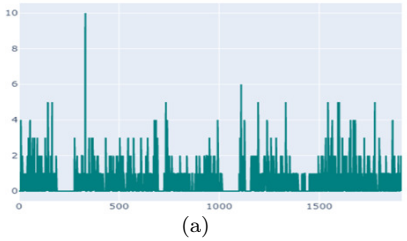


Figure 9: FOODS-3-090 Sale Prices over Time

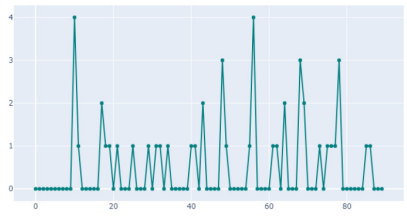
2.7 Instability

We select the sales data of FOODS_1_001_TX_3 and analyze them. According to the figure 10(a), we can see that there are large fluctuations, which may be because that there are too many things that could affect the sales on that day. On certain days, there are no sales, which means that a certain product may not be available that day or the stores are closed.

In order to see the data more clearly, we choose a snippet from that (Figure 10(b)), which shows that the data is very volatile. Thus, we need to de-noise the data to find the underlying trends in the sales data when doing time series model predictions.



(a)



(b)

Figure 10: Sales of FOODS_1_001_TX_3

2.8 Seasonality and Trend

We take total sales of store CA_1 on each day as an example to do seasonal decomposition and explore the seasonality and trends of the data. As shown in Figure 11, the data experiences regular and predictable changes that recur every calendar year.

Due to the instability, seasonality and trend of the data, we choose SARIMA as one of our models since it has parameters to solve these problems.

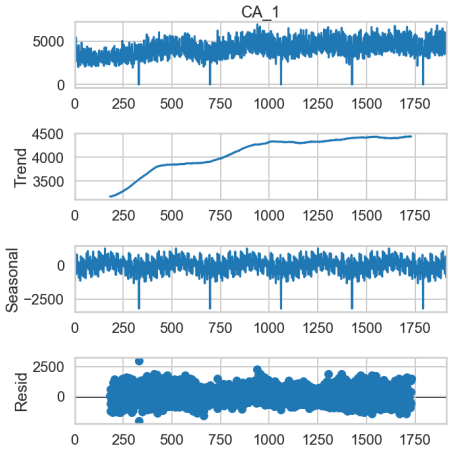


Figure 11: Seaonality and Trend of CA_1

3 Model Introduction

3.1 SARIMA

SARIMA or Seasonal ARIMA(Seasonal Autoregressive Integrated Moving Average) is an extension of ARIMA that explicitly supports univariate time series data with a seasonal component. It adds three new hyperparameters to specify the autoregression (AR), differencing (I) and moving average (MA) for the seasonal component of the series, as well as an additional parameter for the period of the seasonality.

3.2 LightGBM

LightGBM (Light Gradient Boosting Machine) is a framework that implements the GBDT algorithm. It is a decision tree algorithm based on Histogram. It uses Gradient-based One-Side Sampling (GOSS), which can reduce a large number of data with only small gradients, so that when calculating information gain, only the remaining data with high gradients can be used, compared to XGBoost traversing all the feature value saves a lot of time and space overhead; it uses Exclusive Feature Bundling (EFB) to bind many mutually exclusive features into one feature, which achieves the purpose of dimensionality reduction; different from most GBDT tools that use inefficient level-wise decision tree growth strategy, lightGBM uses the leaf-wise strategy with depth limitation, which solves many unnecessary splits caused by treating the same layer of leaves indiscriminately in the original method.

4 Modelling and Forecasting

4.1 Performance Evaluation: RMSSE

The competition use the RMSSE(Root Mean Squared Scaled Error) to evaluate the model accuracy. This measure is calculated as follows:

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

Here Y_t is the real value of the test data at t, $\widehat{Y}_t$ is the forecast value. n is the number of the training data, h is the forecasting horizon.

The training set includes data from d_1 to d_1913, the validation set includes data from d_1914 to d_1941. After constructing models for training set, we will calculate RMSE of validation set and use it to evaluate the performance of fitted model. At the end, we will predict the future sales for each item in 3 states in next 28 days, i.e. d_1942 to d_1969.

(Note: In this paper we don't scale the mean square error and directly use RMSE because the denominator is constant.)

4.2 SARIMA

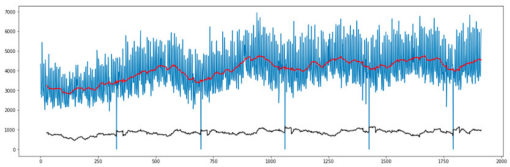
At first, we only consider the time shift effect. So, we choose to use SARIMA model to fit the data.

4.2.1 Model Construction

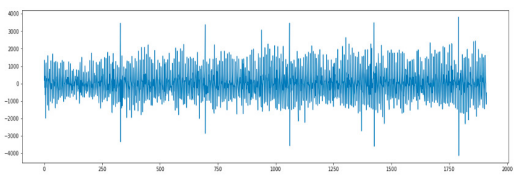
Since it is hard to tune the parameters of the SARIMA model for each item, we would like to use the sum of sales of each store to construct a unique model for each store and then apply the model to predict the sales data for each item sold in this store and obtain the corresponding RMSE.

First, we construct time series of total sales for each store. Taking total sales of CA_1 as an example. As shown in Figure 12(a), the series shows a pattern of seasonality. And according to the p-value of Dickey-Fuller Test (0.271267) , we can conclude that the series is not stationary.

After differencing the data, the p-value of Dickey-Fuller Test becomes 0, which means the series is now stationary.



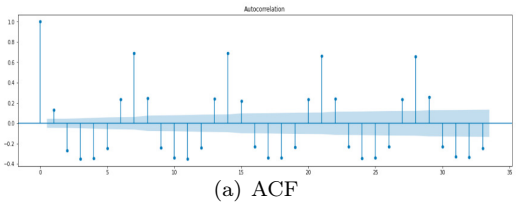
(a) Time series before differencing



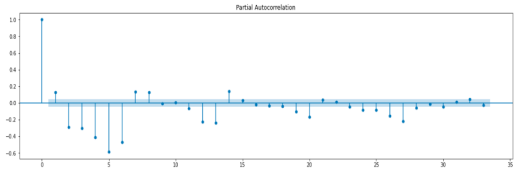
(b) Time series after differencing

Figure 12: Time Series: Total Sales of CA_1

Then, we calculate the ACFs and PACFs. From the results shown in Figure 13, we decide the range of p and q is from 0 to 5, and the the seasonal parameter equals to 7. The resultant model automatically selected by the auto_arima() function is SARIMA(5, 1, 0)x(1, 0, [1], 7). Finally, we apply this model to predict the sales data for each item sold in this store and obtain the corresponding RMSE which is 5.9030.



(a) ACF



(b) PACF

Figure 13: ACF PACF

4.2.2 Performance

We repeat the whole procedure demonstrated above for the remaining nine stores, and the RMSE results for all stores are shown in Table 1.

Store_id	RMSE
CA_1	5.9030
CA_2	2.8008
CA_3	3.2848
CA_4	1.6776
TX_1	2.3938
TX_2	3.2404
TX_3	3.2241
WI_1	2.1808
WI_2	4.3800
WI_3	3.0816

Table 1: SARIMA-RMSE at the store level

4.3 LightGBM

4.3.1 Features Construction

To capture more information of data, we construct several features:

- lag_7: sales shifted 7 steps downwards for each item
- lag_28: sales shifted 28 steps downwards for each item
- rmean_28: rolling mean sales of a window size of 7 over column lag_7
- rmean_7_28: rolling mean sales of a window size of 7 over lag_28
- rmean_28_7: rolling mean sales of a window size of 28 over lag_7
- rmean_28_28: rolling mean sales of a window size of 28 over lag_28
- item_sold_avg: mean sales for each item
- store_sold_avg: mean sales for each store

Lags capture the weekly similarity. People usually go shopping on a fixed day each month. Rolling mean of lags provides the information about a whole period rather than just a single day.

4.3.2 Parameters Tuning and Modelling

Since all stores have the same item, we make a simplification here. Similar to constructing SARIMA model, we also construct models at the store level. The difference is that we don't regard the sales data as time series and we use all the information of each item in the store to construct a certain model.

After tuning parameters many times, the parameters of the LightGBM model we finally use are as following.

- n_estimators=1000
- learning_rate=0.3
- subsample=0.8
- colsample_bytree=0.8
- max_depth=8
- num_leaves=50
- min_child_weight=300

We calculate RMSE at the store level. As shown in Table 2, the results are no more than 3, which are slightly smaller than the results of SARIMA model. It seems that lightGBM performs better.

Store_id	RMSE
CA_1	2.1068
CA_2	1.9536
CA_3	2.5180
CA_4	1.4128
TX_1	1.7023
TX_2	1.8498
TX_3	1.9519
WI_1	1.6655
WI_2	2.8888
WI_3	1.9855

Table 2: LightGBM-RMSE at the store level

Figure 14 reports the resultant importance of the top-20 features for lightGBM. "mean_7_28", "mean_7_-7" and "mean_28_28" are the top three important features, which means sales are strongly influenced by time. "sell_price" is also a very importance feature.

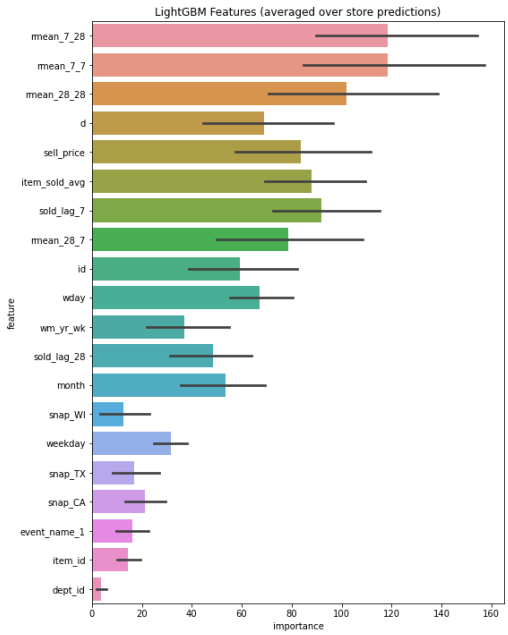


Figure 14: Feature importance of LightGBM

4.4 Forecasting

We choose to directly predict(we don't use "recursively prediction" method like some people) sales the predictions for the sale of each item(30490 in total) by lightGBM models we constructed before. Following are the results.

	id	F1	F2	F3	F4	F5	F6	F7	F8	F9	...	F19	F20
0	FOODS_1_001_CA_1_validation	1.142495	0.979554	0.979554	0.955764	1.124632	1.354340	1.358487	1.167306	1.014468	...	1.096283	1.250018	1.2
1	FOODS_1_001_CA_2_validation	1.162249	1.665484	1.271887	1.535840	1.550325	1.952053	1.701352	0.966333	1.185334	...	0.998394	1.356910	1.5
2	FOODS_1_001_CA_3_validation	1.332868	1.319077	1.319077	1.319077	1.180423	1.714863	0.921875	0.932192	0.856835	...	1.024279	1.077544	0.8
3	FOODS_1_001_CA_4_validation	0.476540	0.409494	0.425530	0.433012	0.463312	0.503810	0.562930	0.473710	0.446124	...	0.394920	0.454987	0.3
4	FOODS_1_001_TX_1_validation	0.224414	0.217026	0.217026	0.217026	0.183568	0.245822	0.231586	0.168520	0.161944	...	0.246031	0.542685	0.5
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
60975	HOUSEHOLD_2_516_TX_2_evaluation	0.173998	0.144320	0.144320	0.144320	0.153528	0.182590	0.245652	0.152823	0.144320	...	0.106851	0.135913	0.1
60976	HOUSEHOLD_2_516_TX_3_evaluation	0.188675	0.178994	0.178994	0.178994	0.222024	0.289938	0.432262	0.347155	0.296699	...	0.152898	0.180651	0.1
60977	HOUSEHOLD_2_516_WI_1_evaluation	0.134760	0.134760	0.138328	0.138328	0.176149	0.246158	0.220549	0.148555	0.156905	...	0.145201	0.105293	0.0
60978	HOUSEHOLD_2_516_WI_2_evaluation	0.146429	0.118822	0.118822	0.118822	0.141331	0.164129	0.147083	0.066896	0.071987	...	0.100621	0.092388	0.0
60979	HOUSEHOLD_2_516_WI_3_evaluation	0.150935	0.133265	0.133265	0.133265	0.172303	0.115858	0.115858	0.112838	0.112838	...	0.083169	0.093400	0.0

Figure 15: Predicted Sales

5 Reference

[1] https://www.kaggle.com/ar2017/m5-forecasting-lightgbm/notebook

[2] https://www.kaggle.com/gopidurgaprasad/m5-forecasting-eda-lstm-pytorch-modeling/notebook

6 Contribution

Huang Yuning: Latex, report, presentation, slides

Sun Ke: code, report

Zhou Xiaomin: code, report, slides

Tian Xinyu: code, report