MAFS 6010U Final Report

Estimate the Unit Sales of Walmart Retail Goods in the USA

ARTIFICIAL INTELLIGENCE IN FINANCE

LAO Shengbo 20660636
SHI Kai 20643901
Tang Haokai 20662098
ZHOU Zhixuan 20659742

May 2020

1. Introduction

This paper tries different time series and machine learning methods to predict daily unit sales at stores in various locations for two 28-day time periods in the M5 Kaggle Competition.

1.1 Background

The M5 Competition, the latest of the M Competitions in Kaggle, will run from 2 March to 30 June 2020.

The aim of the M5 Competition is similar to the previous four: that is to identify the most appropriate method(s) for different types of situations requiring predictions and making uncertainty estimates. Its ultimate purpose is to advance the theory of forecasting and improve its utilization by business and non-profit organizations. Its other goal is to compare the accuracy/uncertainty of ML and DL methods versus those of standard statistical ones, and to assess possible improvements versus the extra complexity and higher costs of using the various methods.

This paper consists of four main part. The first part describes the background, aim and data structure of the M5 Competition. The second part introduces methods that will be used in modeling, including Simple Exponential Smoothing Model (SES), Light Gradient Boosting Machine and Cat Boost. The third part presents the results of the models, including comprehensive tables and graphs of the overall findings, various subcategories and the forecasting performances achieved by the participating methods for each forecasting horizon. The last part then summarizes the paper and highlights the conclusions of the model.

1.2 Data Description

The M5 Competition differs from the previous four ones in five important ways, some of them suggested by the discussants of the M4 Competition.

First, it uses hierarchical sales data, generously made available by Walmart, starting at the item level and aggregating to that of departments, product categories, stores in three geographical areas of the US: California, Texas, and Wisconsin.

Second, besides the time series data, it also includes explanatory variables such as price, promotions, day of the week, and special events (e.g. Super Bowl, Valentine’s Day, and Orthodox Easter) that affect sales which are used to improve forecasting accuracy. The distribution of uncertainty is being assessed by asking participants to provide information on four indicative prediction intervals and the median.

Third, the majority of the more than 42,840 time series display intermittency (sporadic sales including zeros).

Fourth, instead of having a single competition to estimate both the point forecasts and the uncertainty distribution, there will be two parallel tracks using the same dataset, the first requiring 28 days ahead point forecasts and the second 28 days ahead probabilistic forecasts for the median and four prediction intervals (50%, 67%, 95%, and 99%).

Fifth, for the first time, it focuses on series that display intermittency, i.e., sporadic demand including zeros.

- Information about the dates on which the products are sold.
- The historical daily unit sales data per product and store [day 1 to day 1913].
- Information about the price of the products sold per store and date.

1.3 Exploratory Data Analysis

The main tasks of Exploratory Data Analysis (EDA) are cleaning the data, describing the data (descriptive statistics, charts), viewing the distribution of the data, comparing the relationships between the data, cultivating the intuition of the data, and summarizing the data, etc.

Compared with traditional statistical analysis methods, exploratory data analysis methods pay attention to the true distribution of data and emphasize the visualization of data, so that the analyst can see the hidden rules in the data at a glance, so as to be inspired to help the analyst find a suitable model of the data. "Exploratory" means that the analyst's understanding of the solution to the problem will continue to change with the in-depth study.

We need to predict the sales of goods, so let us visually view the total sales. First, we will plot the time series of daily unit sales. The blue line is a smooth fitted line using the loess method.

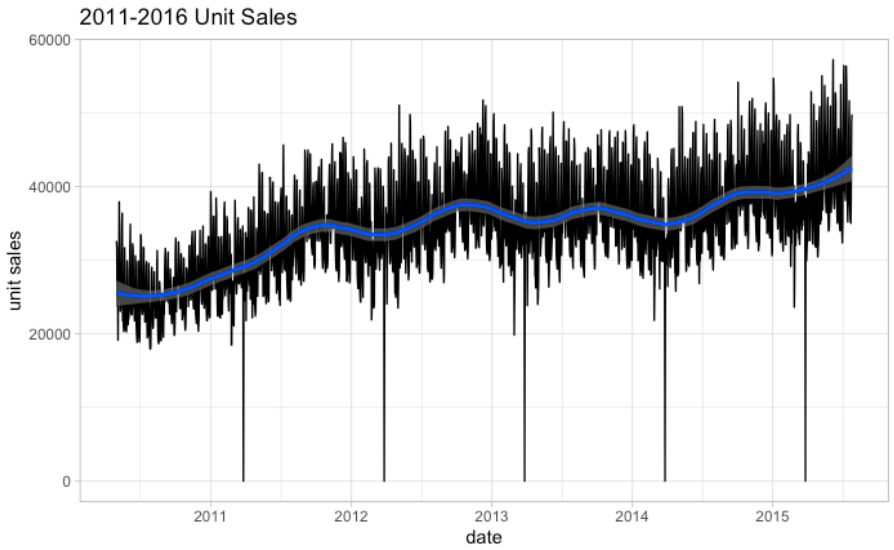

Figure 1: Daily Unit Sales from 2011 to 2016

We can observe from the graph that the overall trend indicates that unit sales are gradually increasing, but there seems to be a significant decline in December and January each year. Then, we hope to better understand whether sales follow a seasonal pattern. The next graph will look at "Average Monthly Sales".

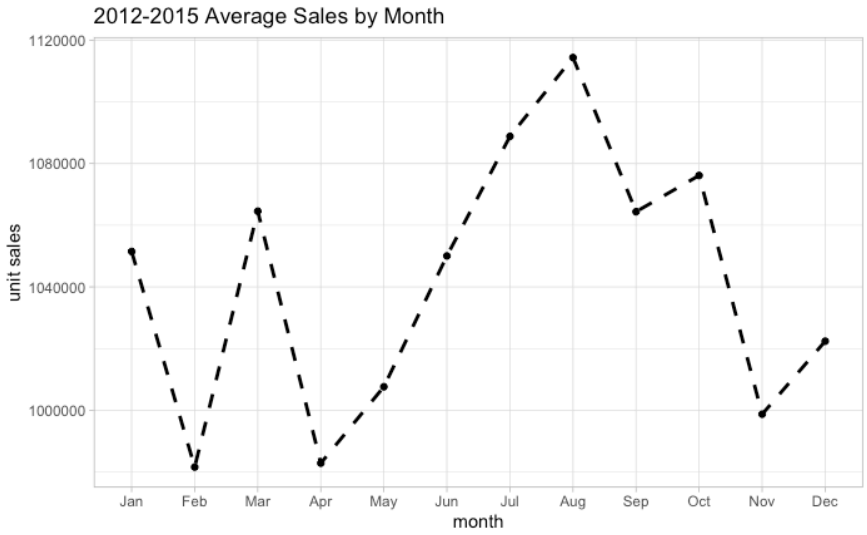

Figure 2: Average Sales by Month from 2012 to 2015

We observe that:

1. February and April are the months with the lowest median unit sales.

2. Regardless of the year, sales will soar in the three months of summer (June / July / August), and August is the peak month. Sales fall after autumn and winter.

Now, we want to know if there is a trend for 7 days of the week. For example, we suspect that as more people go shopping, their weekend sales will be higher.

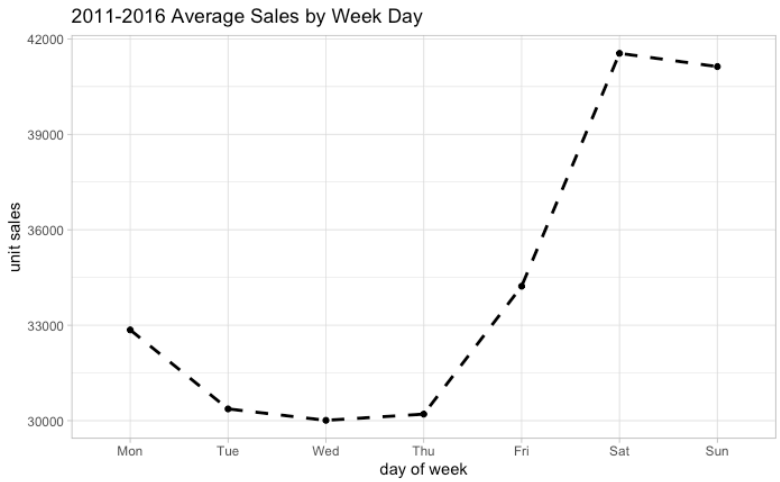

Figure 3: Average Sales by Weekday from 2011 to 2016

The chart above shows that sales during the week (Tuesday / Wednesday / Thursday) are almost flat. Everyone is working and less time shopping, so the demand is reduced. In contrast, sales on the weekend have skyrocketed, the average unit sales on Friday, Saturday and Sunday have increased significantly.

2. Methodology

2.1 Simple Exponential Smoothing Model

For the model building and forecasting part, we first consider using the time series model. At the same time, we noticed that in the M4 competition forecast, the official provides many benchmark models, such as naïve 1, Seasonal Naive, Naive 2, Simple Exponential Smoothing, Holt's Exponential Smoothing, etc. Among them, we prefer to start with Simple Exponential Smoothing Model.

The exponential smoothing method is proposed by Brown (Robert G. Brown), who believes that the trend of time series is stable or regular, so the time series can be reasonably postponed. He believes that the recent past the situation will continue to some extent to the nearest future, so put a larger weight on the most recent data.

Exponential smoothing is a method commonly used in production forecasting. It is also used for forecasting economic development trends in the short to medium term. Among all forecasting methods, exponential smoothing is the most widely used. The simple whole-period averaging method uses all the past data in the time series equally without any leakage; the moving average method does not consider the data in the longer term and gives greater weight to the recent data in the weighted moving average method; and the index The smoothing rule is compatible with the strength of full-time average and moving average. It does not discard the past data, but only gives a gradually weakening effect, that is, as the data moves away, it is given a weight that gradually converges to zero.

Commonly used exponential smoothing methods include primary exponential smoothing, secondary exponential smoothing and cubic exponential smoothing. One-time exponential smoothing, also called simple exponential smoothing (SES), is suitable for predicting time series without obvious trends and seasonality.

The forecast equation is that:

$$\hat{y}_{t+h|t} = l_t$$

The smoothing equation is that:

$$l_t = \alpha y_t + (1-\alpha) l_{t-1}$$

Where y_t is the true value, $\hat{y}_{t+h|t}$ is prediction value, l_t is the smoothed value with

$$0 < \alpha < 1$$

To remove the seasonality of the data, we use classical multiplication decomposition. Specific steps are as follows:

- Step 1: If m is even, use 2×m-MA to calculate the trend period $\hat{T}_t$.

  If m is odd, use m-MA to calculate the trend period $\hat{T}_t$.
- Step 2: Calculate the detrending sequence: $\frac{y_t}{\hat{T}_t}$
- Step 3: In order to estimate the seasonal items for each quarter, simply average the detrending values for that quarter. For example, for monthly data, the seasonal term in March is the average of all March values after trend removal. Then adjust these seasonal items so that their sum is 0. Season items are obtained by combining the data arrangement of these years, namely $\hat{S}_t$.
- Step 4: The residual term is obtained by dividing the time series by the estimated seasonal term and trend-period term: $\hat{R}_t = \frac{y_t}{\hat{T}_t \hat{S}_t}$.

m is seasonal cycle (eg: m = 4 seasonal frequency, m = 12 monthly frequency).

After removing the seasonality of the data, we only use simple exponential smoothing to make predictions and convert the data into the format required by the competition.

2.2 Decision Tree Models: Light GBM and CatBoost

The models we used are decision tree-based model, which are the two most popular and efficient ones in most Kaggle competition: Light GBM and CatBoost. Both models are derived from Gradient Boosting trees. The basic idea of decision tree model and Gradient Boosting are demonstrated as follow.

The decision tree is made of nodes and branches. It would split the data based on the input features, and then generate two or more branches as output. This splitting process would continue until a node is generated where all features were used in or any further splits are no longer possible. A simple visualized example is shown below:

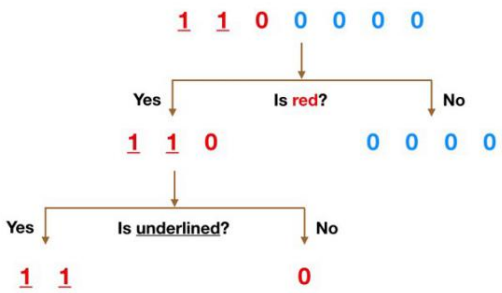

Figure 4: Simple Example of Decision Tree Model

The main idea of Gradient Boosting is to create a strong learner from an ensemble of weak learners. It calculates the loss function and add another weak learner to reduce the loss of the loss function. The loss is calculated as error residuals. During each iteration, the loss function is reducing the error residuals by adding more weak learners. The new weak leaners are concentrating on the part where the current learners perform poorly. When the desired iteration or the error residuals become super small, it will get the final prediction from the ensemble weak learners.

2.2.1 Light Gradient Boosting Machine

The Light GBM was developed as a more efficient GBM technique, it has a faster, more distributed and higher-performance gradient boosting framework. Although it is also based on decision tree algorithms, compared with other algorithms, it splits the tree leaf wise with the best fit instead of splitting the tree depth wise or level wise. So, when growing on the same leaf in Light GBM, the leaf-wise algorithm can reduce more loss than the level-wise algorithm and hence results in much better accuracy which can rarely be achieved by any of the existing boosting algorithms. Besides the accuracy, it has other advantages: Taking less memory, Faster training speed and higher efficiency. Thus, these features make it more compatible with large datasets.

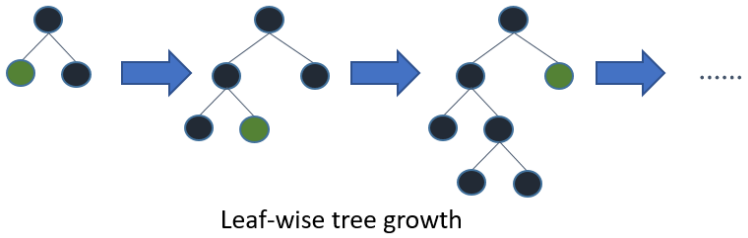

Figure 5: Light GBM Tree Architecture

The main reason for Light GBM has better performance is that it uses One-Side Sampling (GOSS) to filter out the data instances for finding a split value. Since GOSS can apply random sampling on the small part of data with small gradients and at the same time to keep all the data with large gradients. It assumes in training sets, the samples with small gradients will have smaller training error and can be considered as well-trained. It can keep the same data distribution while to use a constant multiplier for the data with small gradients. Thus, these make it can have a good balance between reducing the data size and keeping the accuracy for trained decision trees.

2.2.1 CatBoost

CatBoost is another type of gradient boosting on decision tree-based models. It is an open-source gradient boosting algorithm developed by Yandex team. It can allow users to handle categorical features for datasets, and this is normally better than Light GBM. Compared with Light GBM, it has several advantages:

- Very less requirement for tuning parameters
- Efficient categorical features encoding method (one-hot encoding)
- Built-in GPU version for training process
- Faster prediction speed

Table 1: Important Hyperparameter Comparison of Light GBM and CatBoost

Function	Light GBM	CatBoost
Overfitting Control	1. **learning_rate** 2. **max_depth, num_leaves:** to control the number of tree level and leaves in each tree 3. **min_data_in_leaf**	1. **learning_rate** 2. **Depth:** max is 16 3. **L2-leaf-reg:** L2 regularization coefficient
Speed Control	1. **feature_fraction:** fraction of features to be taken for each iteration 2. **bagging_fraction:** data used in each iteration to speed up the training process and avoid overfitting 3. **num_iterations:** number of boosting iterations performed	1. **rsm:** random subspace method. The percentage of features to use at each split selection 2. **iterations:** max number of trees that can be built

3. Results

Applying SES Model, Light GBM and CatBoost to our dataset, we conduct day by day sales prediction in the following 28 days. This Kaggle competition uses a Weighted Root Mean Squared Scaled Error (RMSSE) to measure the accuracy of models. Based on the score of

prediction submission in Kaggle competition, below we obtained the accuracy of three models.

Table 2: Accuracy of Models

SES Model	Light GBM	CatBoost
0.76302	0.58271	0.52694

From the two feature importance plots of Light GBM and CatBoost, we can see the feature selection of Catboost is more averaged than Light GBM, and compared with their accuracy scores, the Light GBM has higher score than CatBoost. It implies Light GBM has the better ability to abstract the important features than CatBoost.

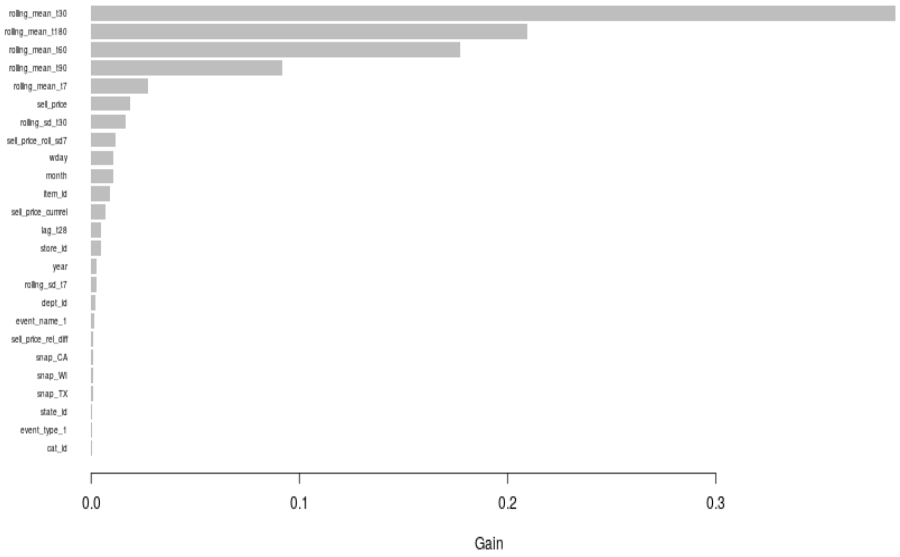

Figure 6: Variable Importance in Light GBM

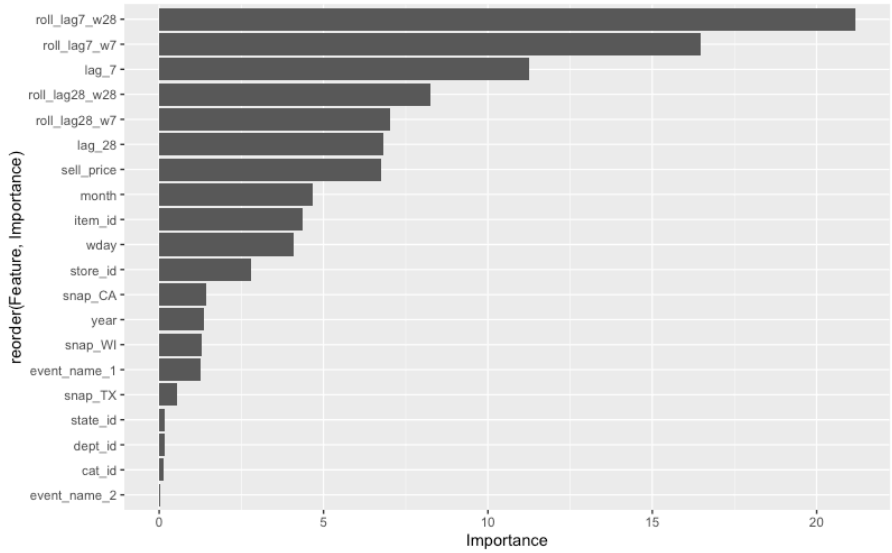

Figure 7: Variable Importance in CatBoost

4. Discussion and Conclusion

In this paper, we employ three models, namely SES model, Light GBM and CatBoost, to forecast the unit sales. Simply comparing the accuracy among three models in this paper, we can draw such a conclusion that SES model shows the best performance in prediction, while

CatBoost behaves the worst.

Surprisingly, SES model as time series model performs better than the two machine learning models, and the time of model fitting is greatly shortened. However, SES model is just a simple method to create some baselines using univariate forecasting techniques.

Even though the accuracy of the latter two models is not that outstanding in this scenario, Light GBM and CatBoost, as derivatives of gradient boosted machines, are excellent in terms of flexibility, optimizing on different loss functions and providing several hyperparameter tuning options that make the function fit very flexible.

Reference

[1] Molnar, Christoph. “Interpretable Machine Learning.” Christoph Molnar, 17 Dec. 2019, https://christophm.github.io/interpretable-ml-book/.

[2] Gandhi, Rohith. “Gradient Boosting and XGBoost.” Medium, HackerNoon.com, 24 May 2019, https://medium.com/hackernoon/gradient-boosting-and-xgboost-90862daa6c77.

[3] SauceCat. “Boosting Algorithm: XGBoost.” Medium, Towards Data Science, 16 May 2017, https://towardsdatascience.com/boosting-algorithm-xgboost-4d9ec0207d.

[4] SauceCat. “Boosting Algorithm: GBM.” Medium, Towards Data Science, 11 May 2017, https://towardsdatascience.com/boosting-algorithm-gbm-97737c63daa3.

[5] Silipo, Rosaria. “From a Single Decision Tree to a Random Forest.” Medium, Towards Data Science, 8 Oct. 2019, https://towardsdatascience.com/from-a-single-decision-tree-to-a-random-forest-b9523be65147.

[6] Singh, Harshdeep. “Understanding Gradient Boosting Machines.” Medium, Towards Data Science, 4 Nov. 2018, https://towardsdatascience.com/understanding-gradient-boosting-machines-9be756fe76ab.

[7] Algorithm - Finding Nearest Neighbors. (n.d.). Retrieved from https://www.tutorialspoint.com/machine_learning_with_python/machine_learning_with_python_knn_algorithm_finding_nearest_neighbors.htm

[8] 1.17. Neural network models (supervised). (n.d.). Retrieved from https://scikit-learn.org/stable/modules/neural_networks_supervised.html

[9] State-of-the-art open-source gradient boosting library with categorical features support. (n.d.). Retrieved May 28, 2020, from https://catboost.ai/

[10] Swalin, A. (2019, June 11). CatBoost vs. Light GBM vs. XGBoost. Retrieved May 28, 2020, from https://towardsdatascience.com/catboost-vs-light-gbm-vs-xgboost-5f93620723db

Link of Source:

https://drive.google.com/drive/folders/1NDX32jE505XAQbX8EhaAhZYQnnfnQtk9

Link of Presentation:

https://drive.google.com/drive/folders/1UbL7TvoBMBqvwdX7U4dUdClvI0euz6p9

Contribution:

Average contribution of each member.