MSBD5013 Project: M5-Forecasting

Yi,Liu Zhixuan,Peng
yliujt@connect.ust.hk
zpengan@connect.ust.hk
School of Engineering,HKUST

Abstract

In this project, we try to predict, the point forecasts and the uncertainty distribution of the realized values of the same series of the unit sales of various products sold in the USA by Walmart, as precisely as possible.For these two different goals, we propose two different model construction schemes, and improve them in a targeted manner.We tried both traditional statistical approach and machine learning methods[1].

1 Introduction

How much camping gear will one store sell each month in a year? To the uninitiated, calculating sales at this level may seem as difficult as predicting the weather. Both types of forecasting rely on science and historical data. In this project, in addition to traditional forecasting methods we are also challenged to use machine learning to improve forecast accuracy.

The Makridakis Open Forecasting Center (MOFC) at the University of Nicosia conducts cutting-edge forecasting research and provides business forecast training. It helps companies achieve accurate predictions, estimate the levels of uncertainty, avoiding costly mistakes, and apply best forecasting practices. The MOFC is well known for its Makridakis Competitions, the first of which ran in the 1980s.

Our work will continue to advance the theory and practice of forecasting. The methods used can be applied in various business areas, such as setting up appropriate inventory or service levels. Through its business support and training, the MOFC will help distribute the tools and knowledge so others can achieve more accurate and better calibrated forecasts, reduce waste and be able to appreciate uncertainty and its risk implications.

2 Dataset

2.1 Overview

we use hierarchical sales data from Walmart, the world's largest company by revenue, to forecast daily sales for the next 28 days and to make uncertainty estimates for these forecasts. The data, covers stores in three US States (California, Texas, and Wisconsin) 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.The original data has four tables, and their specific data are shown in Table 1.

33rd Conference on Neural Information Processing Systems (NeurIPS 2019), Vancouver, Canada.

Table 1: Original Dataset Overview

table name	column number	row number
sell price	4	6,841,121
sells train validation	1,919	30,490
sells train evaluation	1,947	30,490
calendar	14	1,969

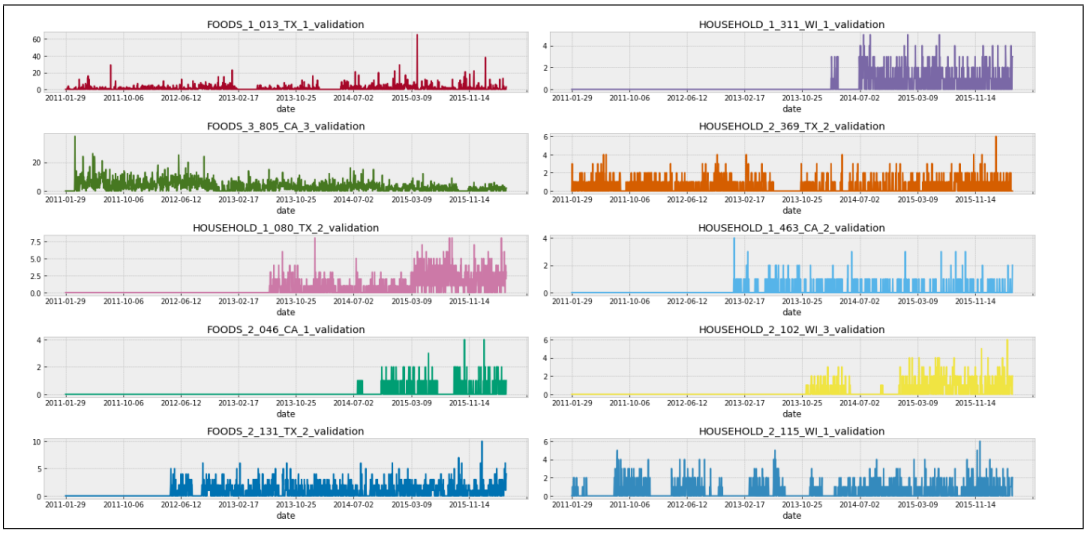

Figure 1: ten items as samples

2.2 Exploratory Data Analysis

In order to better understand and use the data we obtained, we performed EDA to visualize the data.Firstly, we randomly obtained 10 items as samples to see if there is any trend in each data as the figure 1 shows.

Then, we found that many items that were not sold before and are now on sale, and many items that were sold before but are no longer available, we need to deal with these products uniformly. Secondly. because our data comes from many different stores, and each store's sales are different,it shows that we may need to average these sales.These visual images provide ideas for our subsequent data preprocessing.

In order to further understand the sales trend of products, we used the wavelet denoising method to observe the sales trend of several sample products and figure 2 gives an example.

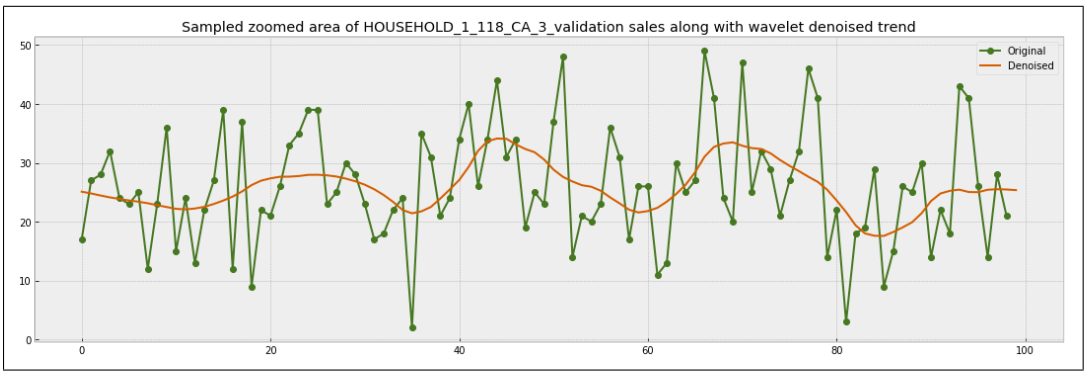

Figure 2: an example of wavelet denoised trend

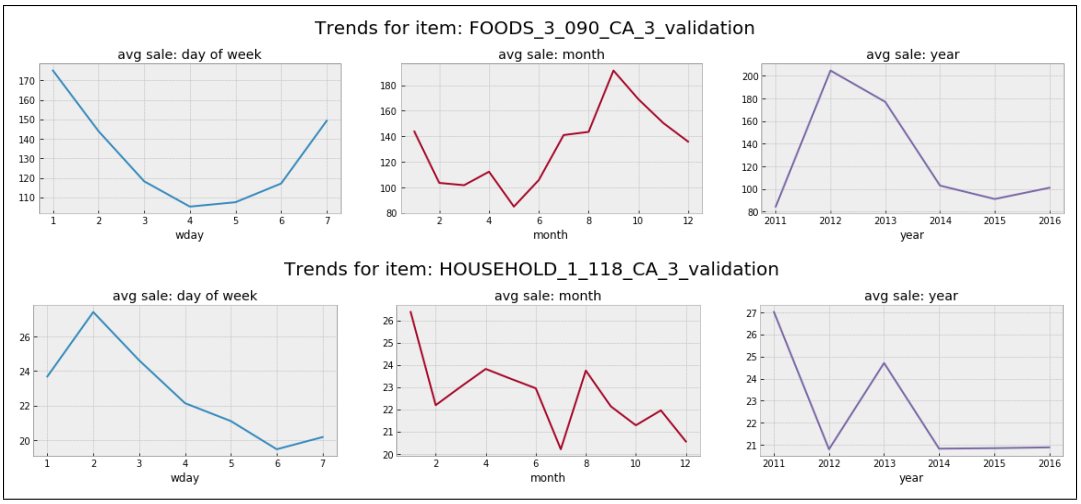

Figure 3: Sales trend in different period of time

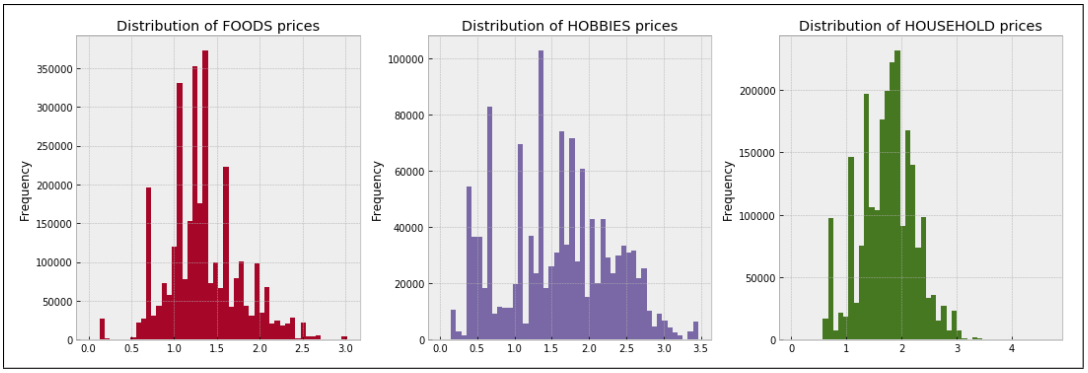

Figure 4: Distribution of different types

We also observed some commodities according to different time periods. We intercepted the average sales of each week, month and year to observe the changes in the sales of the commodity in a fixed period of time, and figure 3 shows some examples.These give us a general and comprehensive understanding of the changing trend of commodity sales, which provides ideas for our follow-up forecasts.

Since there is category information for items in the original dataset, to see the relationship between category and sales we looked at the sales frequency distribution for each category. Figure 4 shows an example.

We also found some outliers in sales due to holidays or weekends, which should be considered when necessary in subsequent data processing and forecasting.Figure 5 shows an example.

2.3 Data Preprocessing

Due to the nature of time series, we use R language for data preprocessing. Because the data set is too large, memory compression is required to avoid running out of memory. The original dtype will be changed to the dtype with the smallest memory. Results show that this method greatly saves the memories. After doing some basic preprocessing on the data such as drop duplications and padding with zeros, then we focus on feature engineering:

- One-hot encode events.We associate the events in the table with items with one-hot encoding, adding feature columns for items.

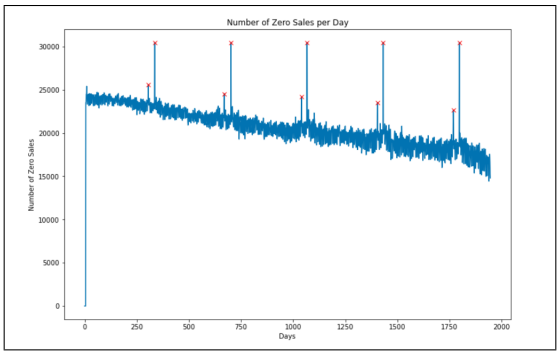

Figure 5: example of outliers

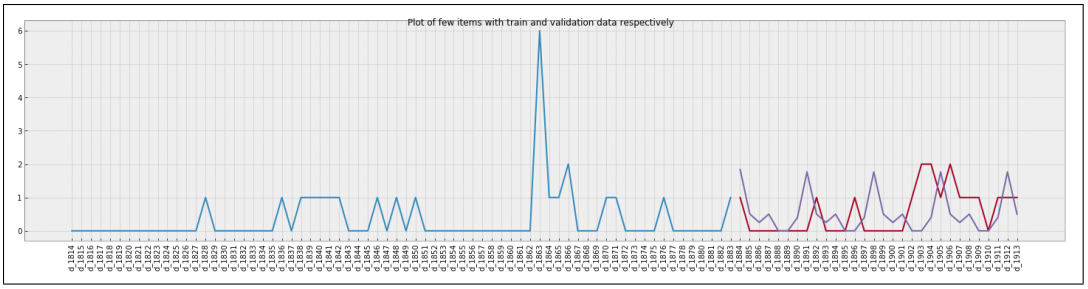

Figure 6: An example of ARIMA model's prediction

- Create features from date. We add year, month, day of month, day of week and whether it is a weekend (day of week is either 7 or 1). More importantly, holiday is added because there are outliers which are all holidays like thanksgiving[2].
- Price Momentum.We have calculated weekly price momentum, which is helpful in analyzing sales trends.
- Generate lags. We generated lags using time intervals of 7 days, 14 days, 30 days, 60 days, and 180 days, and calculated their mean and standard deviation, and made them into data as items' features.

Finally, we got the data table that will be put into the model, with a total of 43 columns, which means that each item has 43 features.

3 Model construction

3.1 Statistical Model: ARIMA

Since the data we are going to predict is a time series, we first thought of trying the traditional statistical model ARIMA to see the prediction effect.We differenciated the data to get a stable series, and observed ACF and PACF, and also tested p-values, and then we explore if the time series had seasonality, which will be added to the ARIMA model. The final prediction example result is shown in Figure 6. It can be seen that although ARIMA's prediction is not very accurate at the point level, it can get roughly accurate predictions for data having a typical trend, and this might be useful to predict the uncertainty distribution.

3.2 Ensemble Tree Model: LightGBM

Gradient Boosting Decision Tree (GBDT) is a popular machine learning algorithm with following reasons. First, tree model naturally supports feature combination and selection, and has strong robustness to outliers. In addition, based on the idea of ensemble, GBDT can achieve low variance and low bias at the same time.

In terms of LightGBM, it is a distributed framework to implement GBDT algorithm, which supports efficient parallel computation to achieve faster training speed. More importantly, LightGBM requires less memory, which is very crucial in this competition thanks to the huge data set. Besides, it can use the categorical feature directly with high accuracy.

3.3 Time Series Model: Prophet

Prophet is a modular regression model proposed by Facebook [3] to intuitively adjust analysts with domain knowledge about the time series. Prophet contains three main components: trend, seasonality and holidays, which are combined in the following equation:

$$y(t) = g(t) + s(t) + h(t) + \epsilon_t \tag{1}$$

Here $g(t)$ is the trend function to capture the non-periodic trend information, while the periodic changes, such as weekly and yearly seasonality are represented by $s(t)$. Besides, $h(t)$ will take holidays into account because holidays always have unusual results. The error term ϵ_t represents any idiosyncratic changes which are not accommodated by the model

3.4 Deep Learning Model: LSTM

Recurrent neural network (RNN) have been widely used in sequence tasks because it maintains a memory based on history information. However, it suffers from gradient vanishing and cannot achieve long distance dependencies. The same as RNN, Long Short-term Memory (LSTM) [4] is widely used in sequence tasks, while its hidden layer updates are replaced by purpose-built memory cells. The implementation of LSTM memory cell is showed as follow.

$$i_t = \sigma(W_{xi}x_t + W_{hi}h_{t-1} + W_{ci}c_{t-1} + b_i) \tag{2a}$$
$$f_t = \sigma(W_{xf}x_t + W_{hf}h_{t-1} + W_{cf}c_{t-1} + b_f) \tag{2b}$$
$$c_t = f_t c_{t-1} + i_t tanh(W_{xc}x_t + W_{hc}h_{t-1} + b_c) \tag{2c}$$
$$o_t = \sigma(W_{xo})x_t + W_{ho}h_{t-1} + W_{co}c_t + b_o \tag{2d}$$
$$h_t = o_t tanh(c_t) \tag{2e}$$

where σ is the sigmoid function, and i, f, o and c are the input gate, forget gate, output gate and cell vectors, all of which are the same size as the hidden vector h. Compared to RNN, LSTM introduced residuals to solve gradient vanishing and achieve long distance dependencies. Several gates ensure great performance and LSTM is used in both two competitions.

4 Evaluation

4.1 Evaluation Metrics

M5 Forecasting-Accuracy competition requires the point forecasting of the unit sales of various products sold in the USA by Walmart and the metric is Weighted Root Mean Squared Scaled Error (RMSSE), a variant of the well-known Mean Absolute Scaled Error (MASE).

In terms of M5 Forecasting-Uncertainty competition, Weighted Scaled Pinball Loss is used with the following formulas.

$$SPL(u) = \frac{1}{h} \cdot \frac{1}{\frac{1}{n-1} \cdot \sum_{t=2}^{n} |Y_t - Y_{t-1}|} \cdot \sum_{t=n+1}^{n+h} \begin{cases} (Y_t - Q_t(u)) \cdot u, & Y_t \geq Q_t(u) \\ (Q_t u - Y_t) \cdot (1-u), & Y_t < Q_t(u) \end{cases} \tag{3a}$$

$$WSPL = \sum_{i=1}^{N} \cdot w_i \cdot \frac{1}{m} \sum_{j=1}^{m} SPL(u_j) \tag{3b}$$

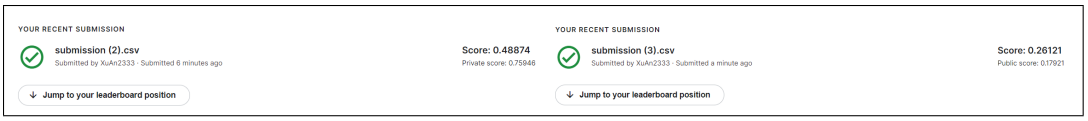

Figure 7: Kaggle results

whereas: Y_t is the actual true future value of the time series at point t; u is the considered quantile; Q_t is the generated forecast for quantile u; h is the forecasting horizon (28 days); n is the length of the training sample (number of historical N is the number of time series m is the number of quantiles observations).

4.2 Stacking Method

Given several models, stacking method can be used to integrate all the results. Stacking is an ensemble learning technique that uses predictions for multiple models to build a new model with lower bias and variance. However, as regression problems, we simply average the prediction of all models. There are some other stacking ways such as weighted average and fit the results into a linear regression model.

The final Kaggle contest scores are 0.75946(0.48874 for public) and 0.261121(0.17921 for public) for accuracy and uncertainty respectively.

5 Summary

In this project, we forecast both accuracy and uncertainty of M5 data. We analyzed the data thoroughly with a lot of data and figures. After that we apply some data preprocess and feature engineering such as memory compression and time-related feature creation. We propose several models, namely ARIMA, LightGBM, Prophet and LSTM and integrate the results of them to achieve better results.

6 Contribution

Peng Zhixuan:implementation of the LightGBM, LSTM and Prophet.
Liu Yi:Data Preprocessing, Feature engineering and ARIMA.

References

[1] Kaggle. M5 forecasting - accuracy. [EB/OL]. `https://www.kaggle.com/c/m5-forecasting-accuracy`.

[2] Kaggle. kaggle data preprocessing. [EB/OL]. `https://www.kaggle.com/code/qiwei13/data-preprocessing#Feature-Engineering`.

[3] Sean J. Taylor and Benjamin Letham. Forecasting at scale. *The American Statistician*, 72(1):37–45, 2018.

[4] Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural Computation*, 9(8):1735–1780, 1997.