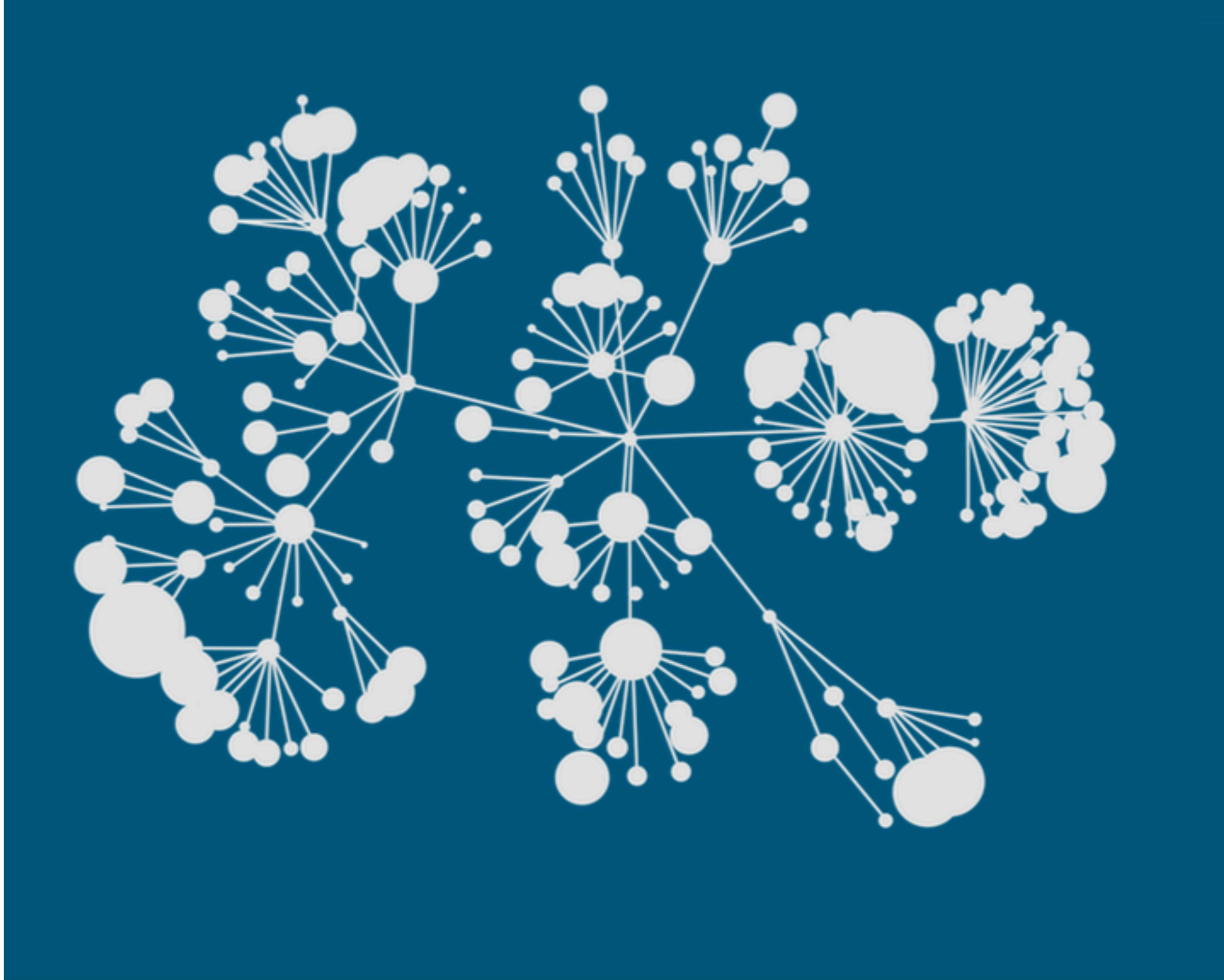
Kaggle

Winner Presentation

kaggle™



Agenda

1. Background
2. My Thought
3. Model Summary
4. EDA
5. Feature selection & engineering
6. Training methods

1. Background

Competition Name: M5 Forecasting - Uncertainty

Team Name: IHiroaki(solo)

Private Leaderboard Score: 0.16147

Private Leaderboard Place: 5th

Competition Name: M5 Forecasting - Accuracy

Team Name: IHiroaki(solo)

Private Leaderboard Score: 0.57644

Private Leaderboard Place: 52th

1. Background

- What is your academic/professional background?

Academic ：Bachelor’s degree

professional ： ～19/8 - Financial Accounting for Manufacturing Industry

20/1～20/4 - Dive into code(https://diveintocode.jp/)

=> I went to “Dive into code” and started learning machine learning.

- What made you decide to enter this competition?

I participated in this competition as an opportunity to put my knowledge of machine learning studied at school into practice.

- How much time did you spend on the competition?

Three and a half months.

2. My Thought

As a way of calculating uncertainty.

1. use machine learning to predict uncertainty itself
2. predict “Accuracy” with machine learning.
   Uncertainty is the error between the actual and prediction in the CV period in that model.

To address both the Accuracy and Uncertainty competitions, I opted for 2.

2. My Thought

In order to optimize both accuracy and uncertainty, we need to create a more general model in Accuracy, even if the CV score is somewhat poor.

(If Accuracy's model is generalized, the difference between actual and forecasted in CV periods is an uncertainty. This uncertainty is a generalization and will be true for any period of time.").

↓

(1)Create non-overfit model

(2)Remove as much risk as possible.

Pretty innocuous model design.(I've eliminated a lot of risk factors.)

(The fact that there was one final submission was also a major factor.)

3. Model Summary

Accuracy

Model ： LightGBM

Model Structure ： 28 days * 10 store_id (280models)

Important feature : Basic Lag, Average Encoding, ID

How long does it take to train your model? :

About 8-9days(very time consuming…)

Uncertainty

Model : Nothing

Method :

The difference between the actual and predicted values over the CV period in Accuracy is used as the uncertainty. point:0.500 uses Accuracy's FinalSubmission.

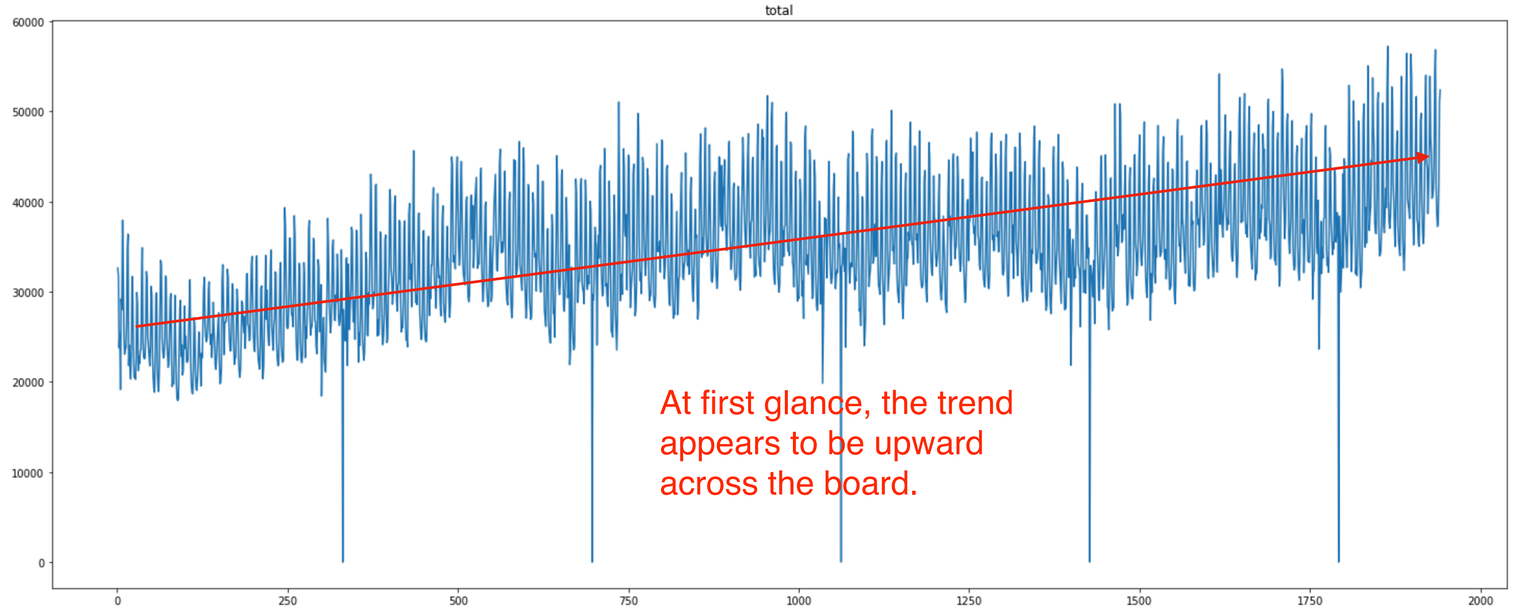
How long does it take to train your model? :

If you want to use the calculated Accuracy results, it only takes a few minutes.

4. EDA

(1) Trend

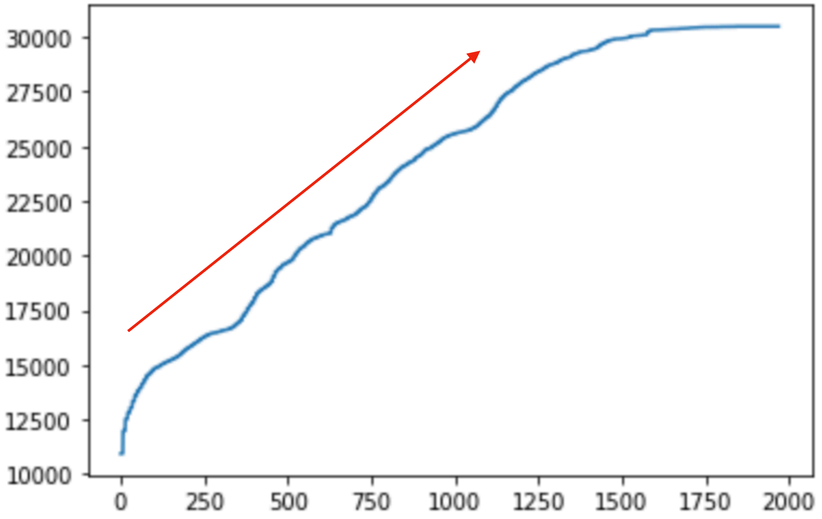
4.1 Total sales plot



Graph 4.1 makes it difficult to see the trend because new items are added as shown in Graph 4.2 .
Therefore, in the next slide, I will try to plot each ID for each year in which sales were started.

By doing so, I should be able to catch the trend.

4.2 Daily unique items



30490

Items are being added on a daily basis, and it is assumed that this is a factor in the upward trend in Total.

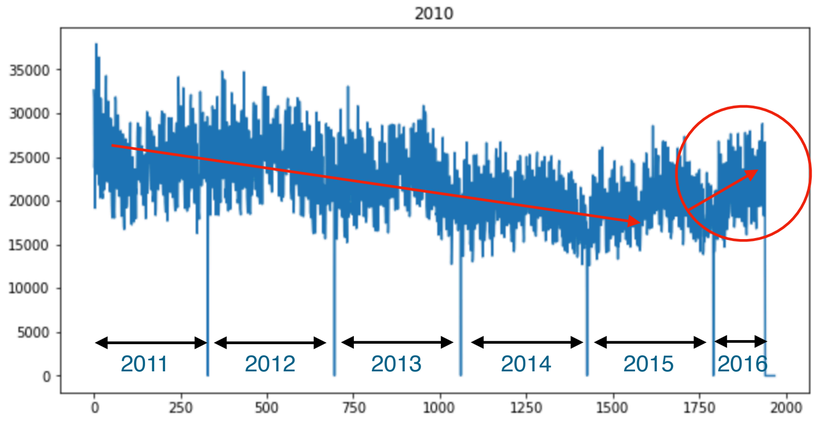
4. EDA

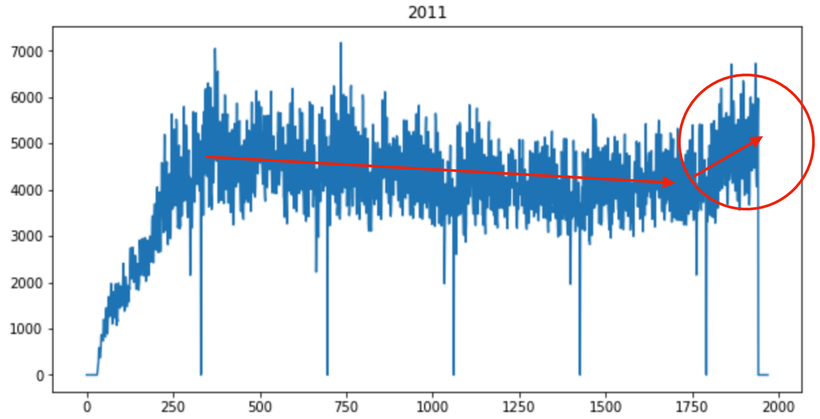
(1) Trend

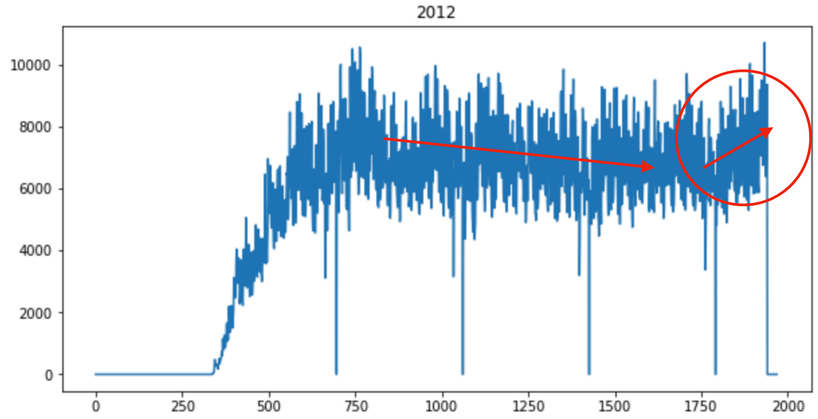
Graph 4.3 shows that all of these trends are on a downward trend until 2015, but from late 2015 Sales have recovered rapidly over 2016. (The same trend by state, and by store_id, especially in CA_2 and WI_2.)

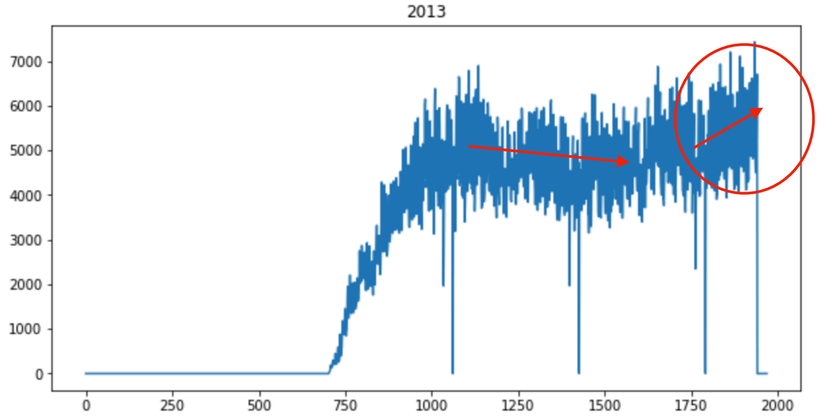
The fact of this increase is significant, and the sharp increase in fiscal year 2016 could be the result of some kind of campaign as a company, but the nature of the increase is unknown, so it is impossible to examine whether it is temporary or ongoing.

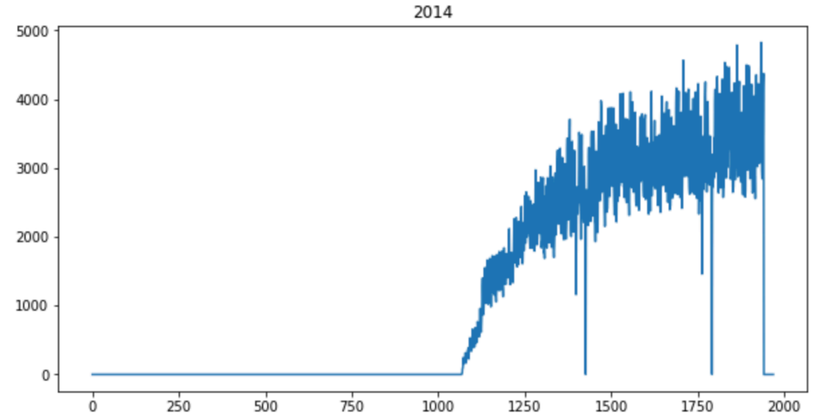
4.3 Total sales plot by year sales started

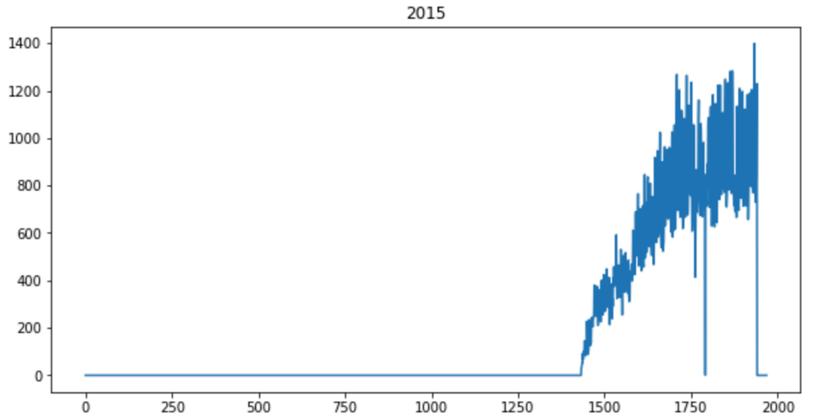












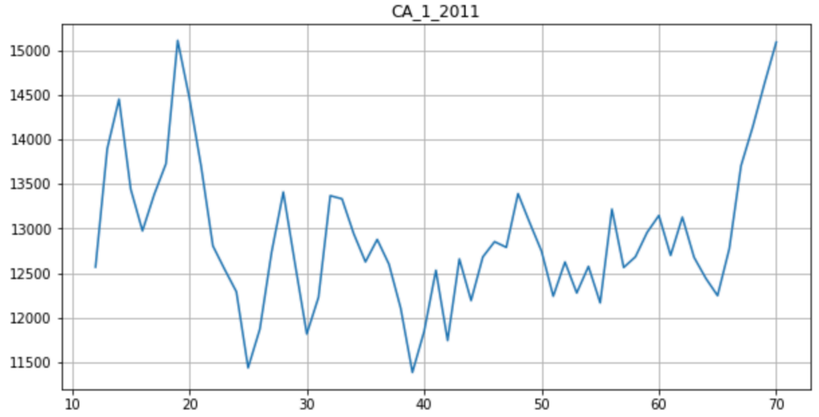
4. EDA

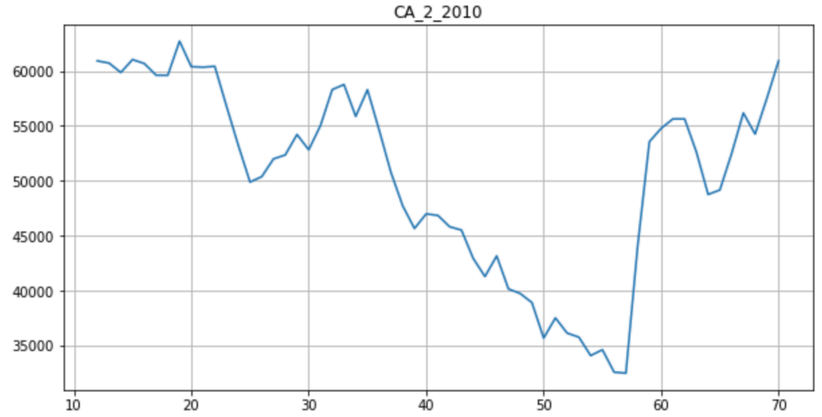
(1) Trend

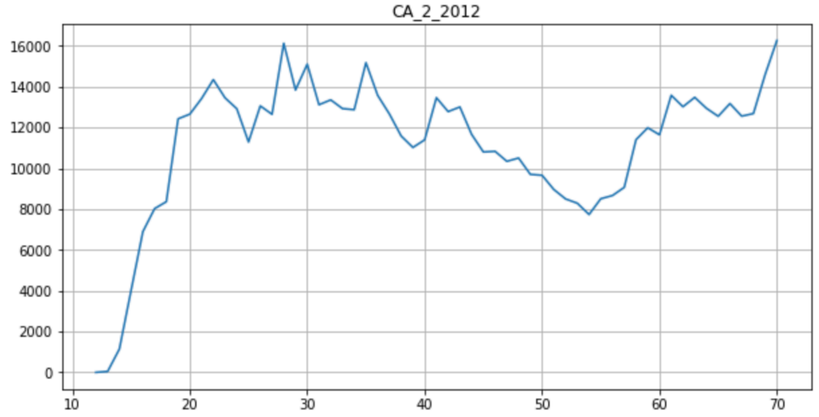
Graph 4.4 plots 28-day totals by sales start year and store_id.

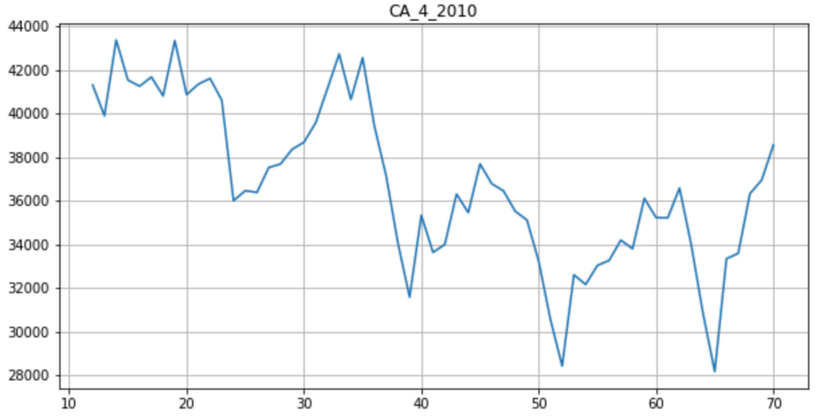
Looking at graph 4.4, it seems that block70 (public period) is at the extreme point in many situations appears to be.
Past trends show that sales often decline after the apex is reached.
Therefore, I can see that Private LB may be lower than Public LB sales.

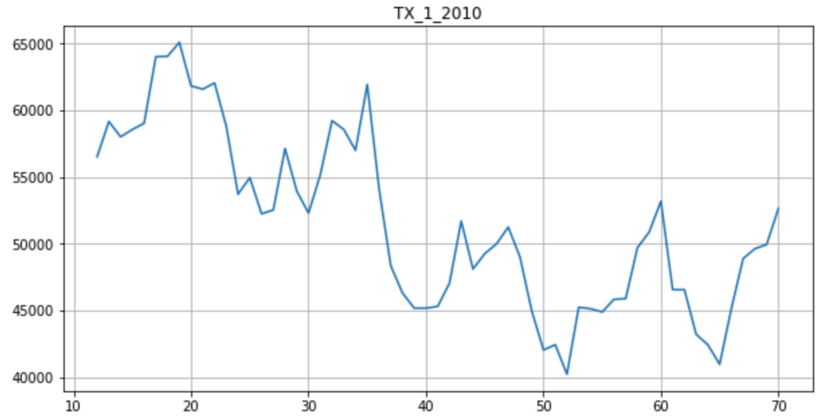
4.4-1 Plot of the total number of sales per 28 days by store_id(1)

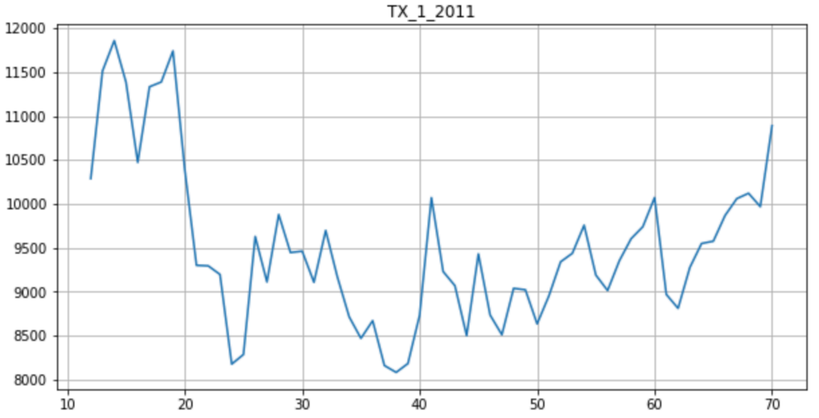








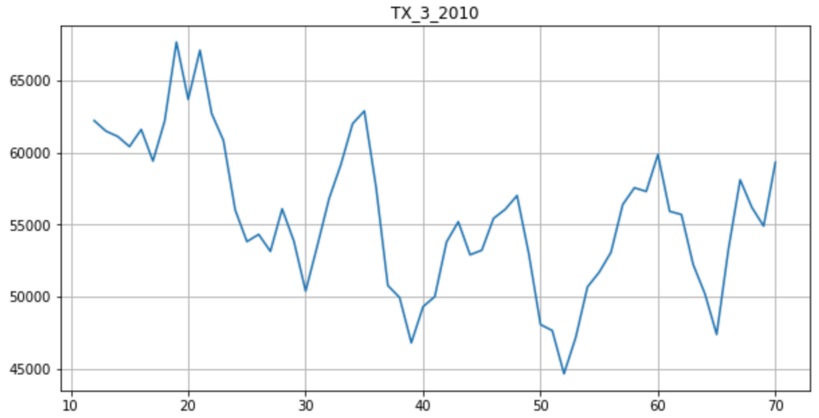


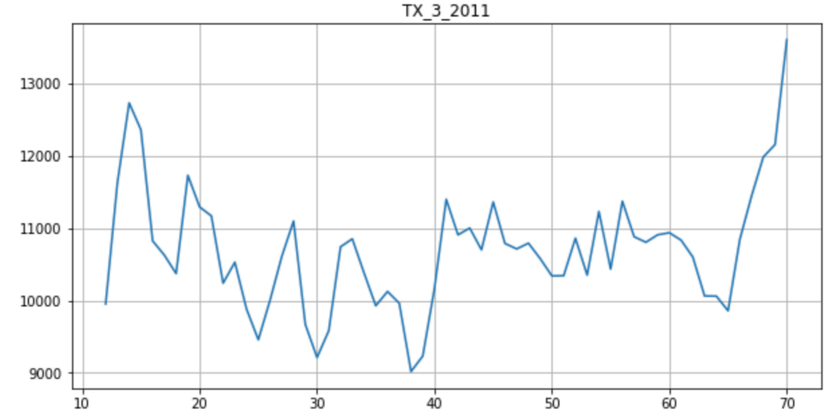


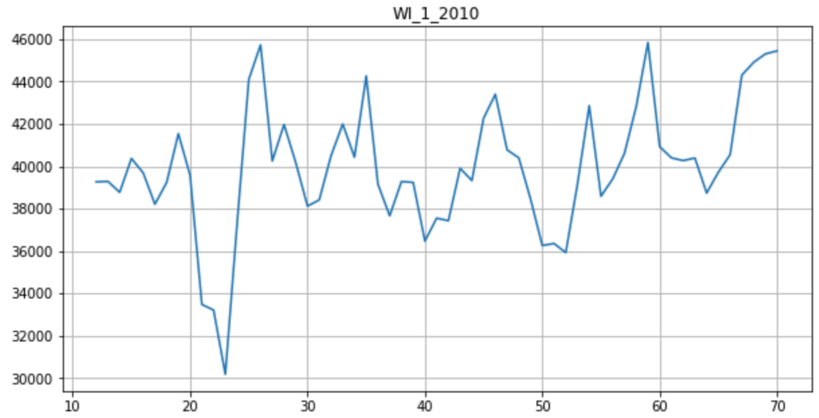
4. EDA

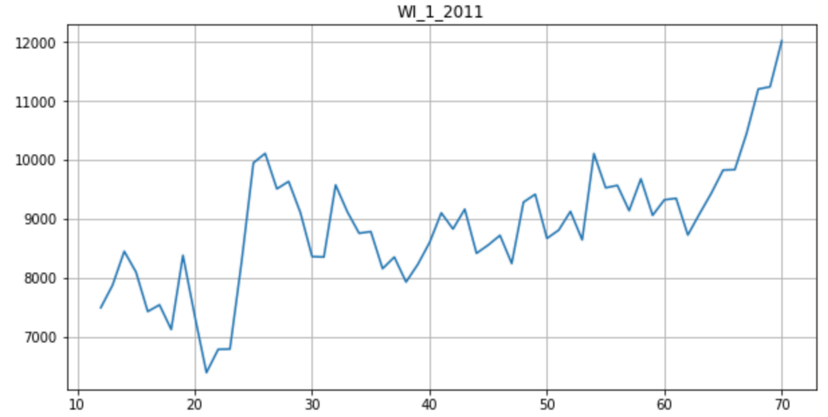
(1) Trend

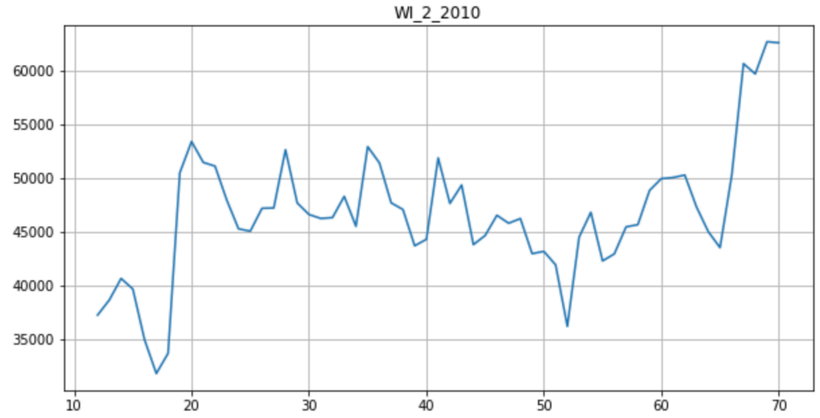
4.4-2 Plot of the total number of sales per 28 days by store_id(2)

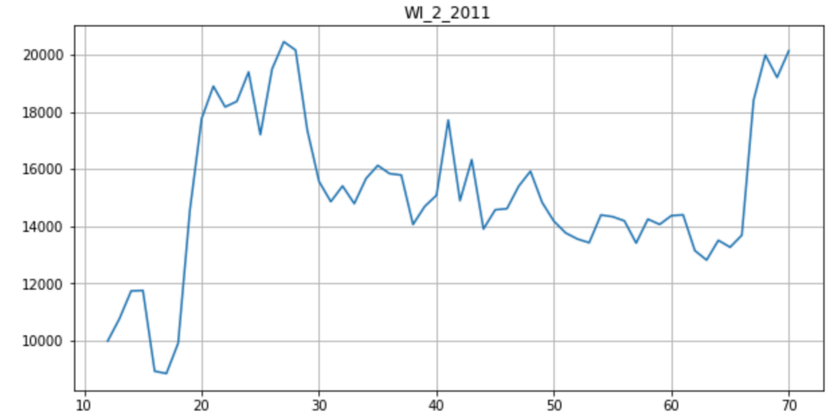










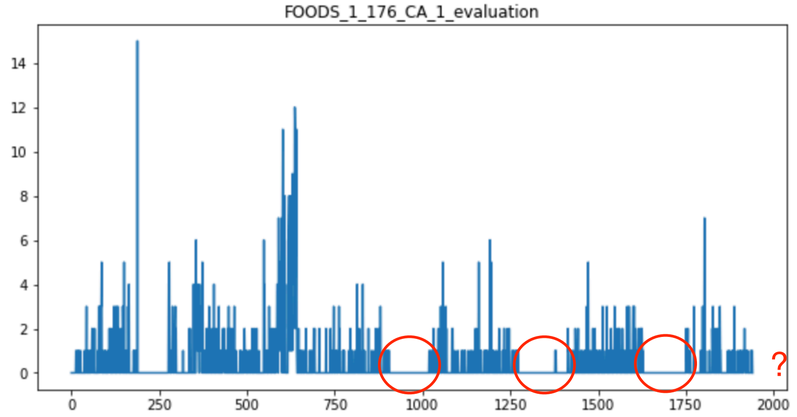


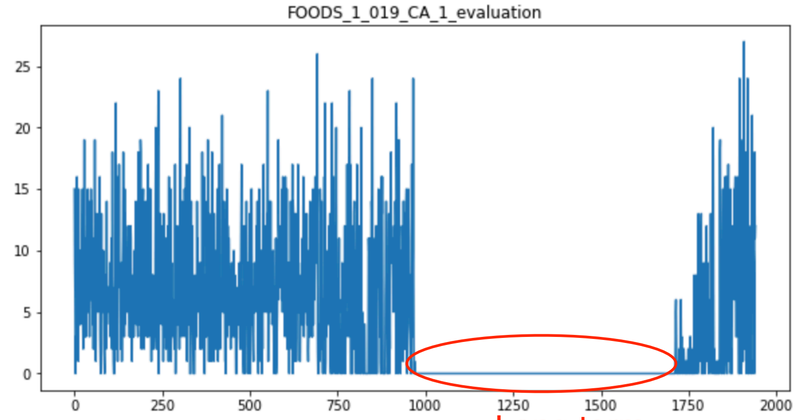
4. EDA

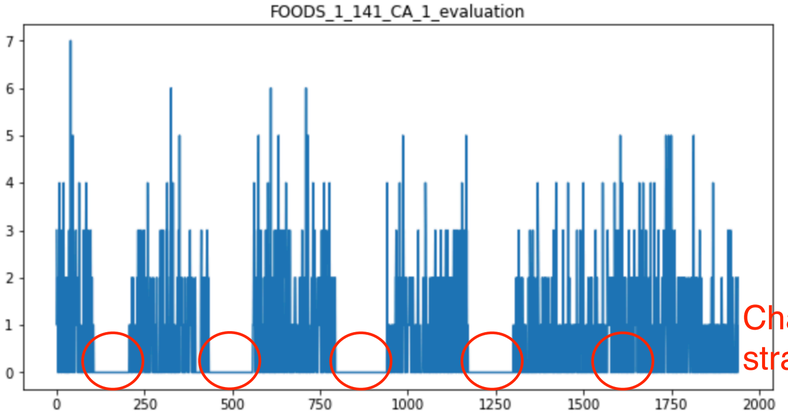
(2) Zero Pattern

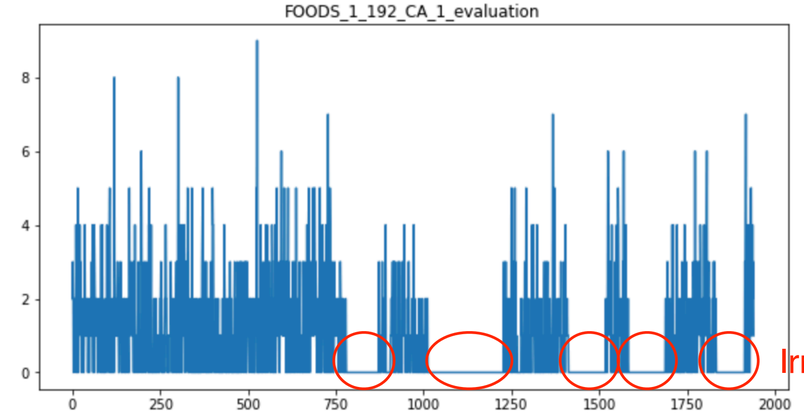
While there are many zeros in this time series, there are likely to be many strategic and inevitable zeros in this time series.
Companies have inventory and product strategies, and these strategies can change every year. Therefore, predicting zero patterns without knowing the inventory and product strategies is not an easy task. I feel it's a risk.

And even if I expect zero, such as out of stock or no sales by chance, the risk is still high because even a one-day deviation in the valuation index(WRMSSE) is not acceptable.









There is a risk in predicting zero patterns without knowing Walmart's product and inventory strategies.

5. Feature selection & engineering

To avoid risk, I eliminated all features that used predictions, so I only used basic features.

Due to the sparse distribution, we tried clustering by distributional approximation, but all of our attempts resulted in poor scores.

Features adopted

- Basic Lag(mean, std, max, min, median)
- Encoding by “level”, “level and day of week”, “level and day”(average, std)
- Basic calendar
- price fluctuation
- ID

**Feature importance plots on the next slide

Features that were not adopted

- Features using predictions (features that quantify zero sales patterns, etc…)
- New Categorization through Clustering
- External data

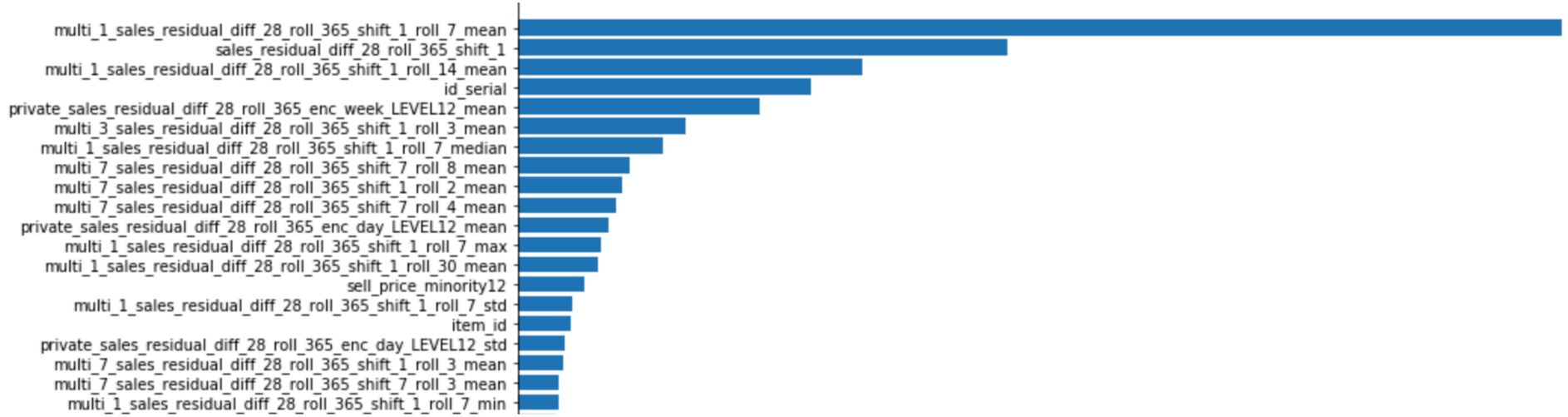
etc…..

5. Feature selection & engineering

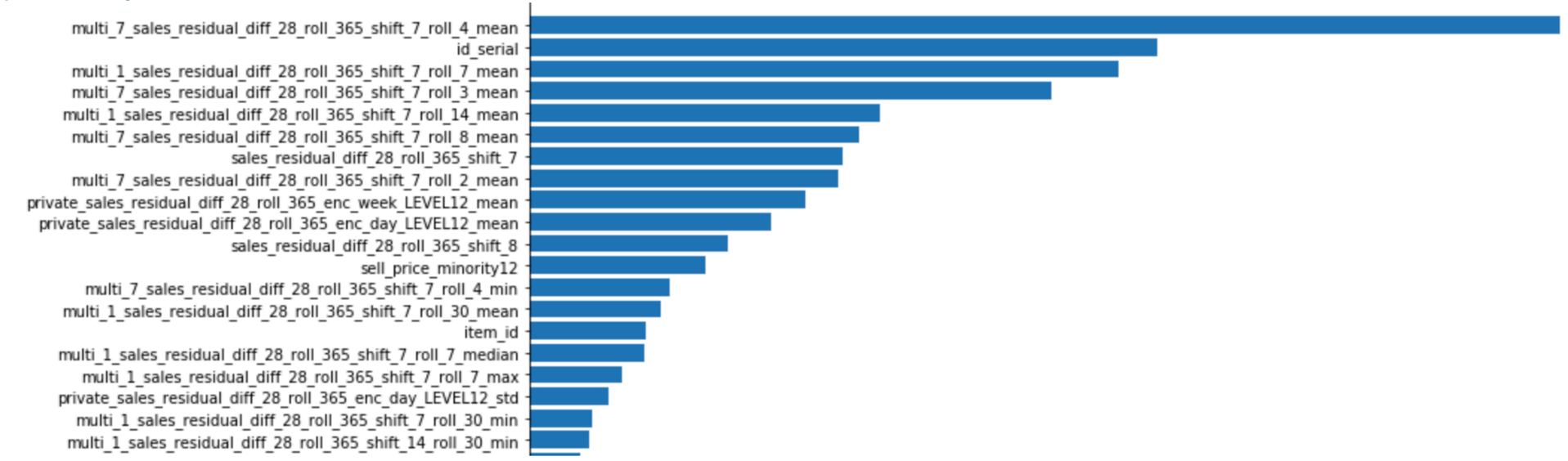
The features that are important for pred_day1 model and pred_day28 model are different.
The closer to day 1, the more important are the preceding numerical data (shift and rolling).
As I get closer to day28, generalized features such as categories and average encoding become more important.
Therefore, it is very meaningful and important to use 28 models.

Feature Importance Plot - Top 20

pred_day1



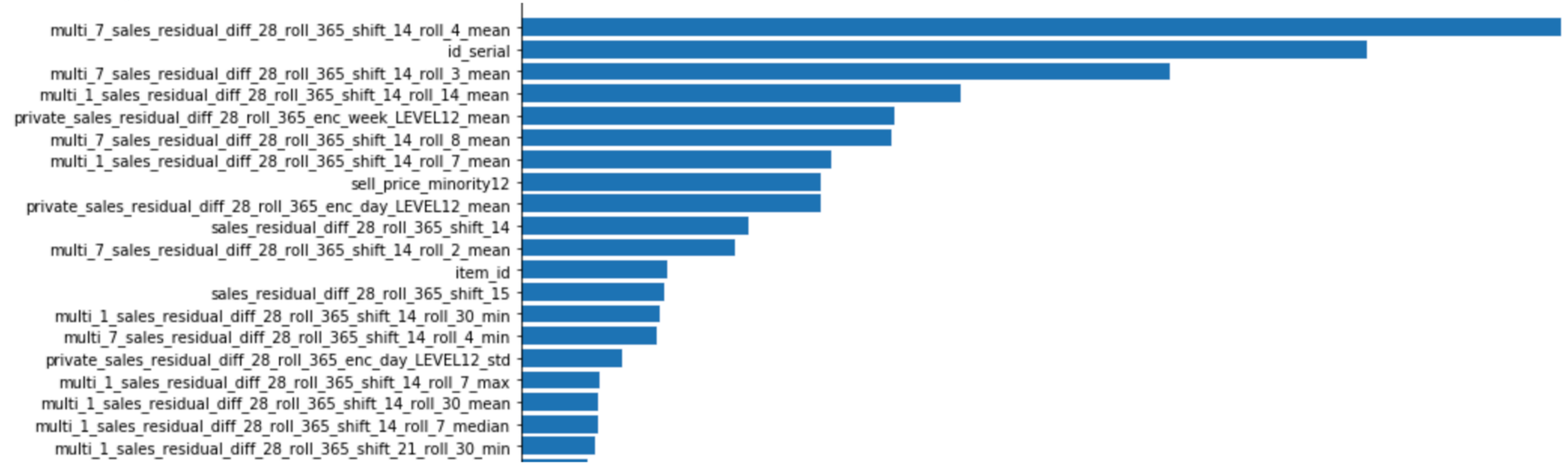
pred_day7



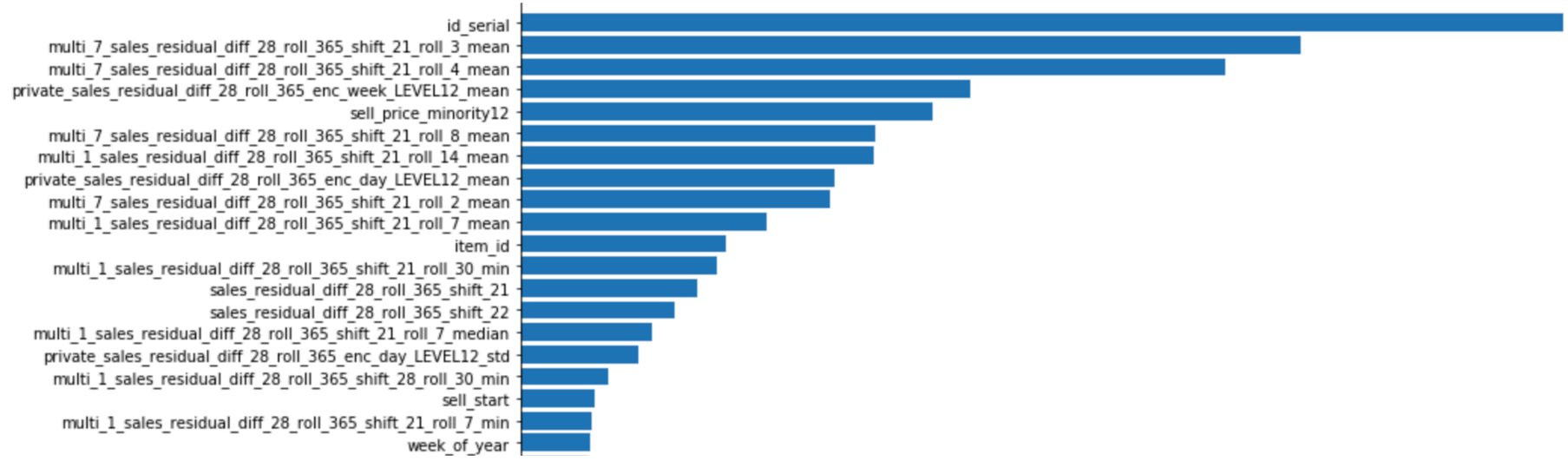
5. Feature selection & engineering

Feature Importance Plot - Top 20

pred_day14



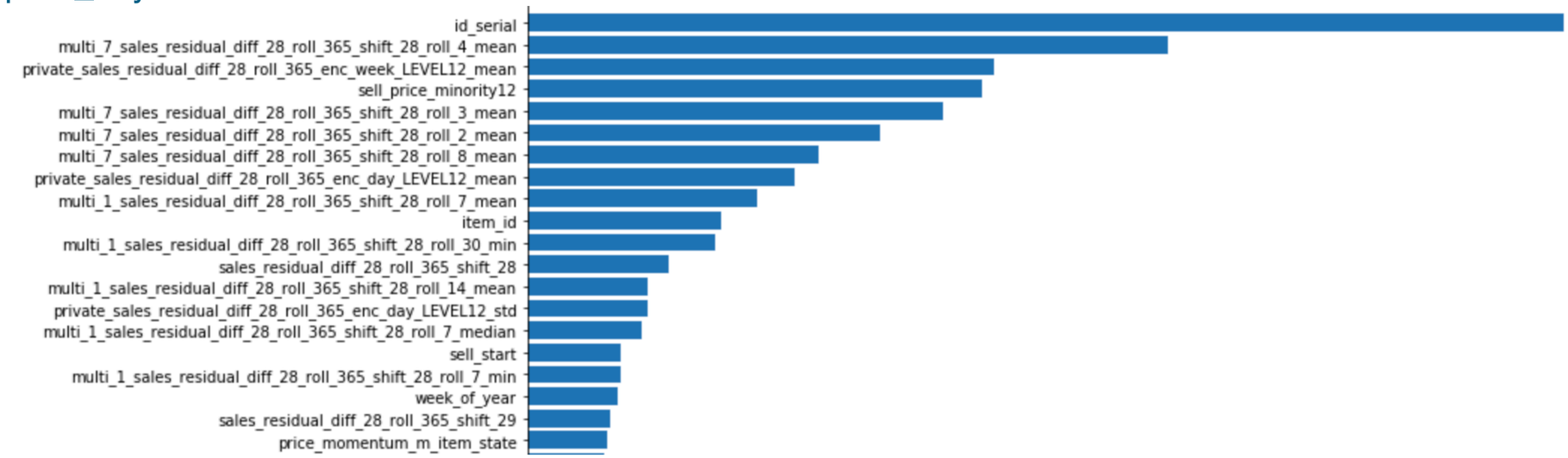
pred_day21



5. Feature selection & engineering

Feature Importance Plot - Top 20

pred_day28



5. Feature selection & engineering

Features name explanation

- sales_residual_diff_28_roll_365 : Target
- multi_5_sales_residual_diff_28_roll_365_shift_1_roll_4_mean :

Code: df[“Target_shift_1”] = df.groupby([“id”])[“Target”].transform(lambda x : x.shift(1)

df.groupby([“id”, “multi_5”])[“Target_shift_1”].transform(lambda x: x.rolling(4).mean())

~	1927	1928	1929	1930	1931	1932	1933	1934	1935	1936	1937	1938	1939	1940	1941	
~	7	1	2	3	4	5	6	7	1	2	3	4	5	6	7	=>multi_7
~	1	2	3	4	5	1	2	3	4	5	1	2	3	4	5	=>multi_5
~	1	2	3	1	2	3	1	2	3	1	2	3	1	2	3	=>multi_3
~	2	1	2	1	2	1	2	1	2	1	2	1	2	1	2	=>multi_2
~	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	=>multi_1

- private_sales_residual_diff_28_roll_365_enc_week(day)_LEVEL12_mean:
  Private => Use sales data until just before the private period
  enc_week_LEVEL12_mean => Average for each day of the week(day) in the level 12 category

- sell_price_minority12 :
  Two-digit number for sell_price combined with the first and second decimal places
  ex) 10.58345 => 58

- id_serial : = id(30490)

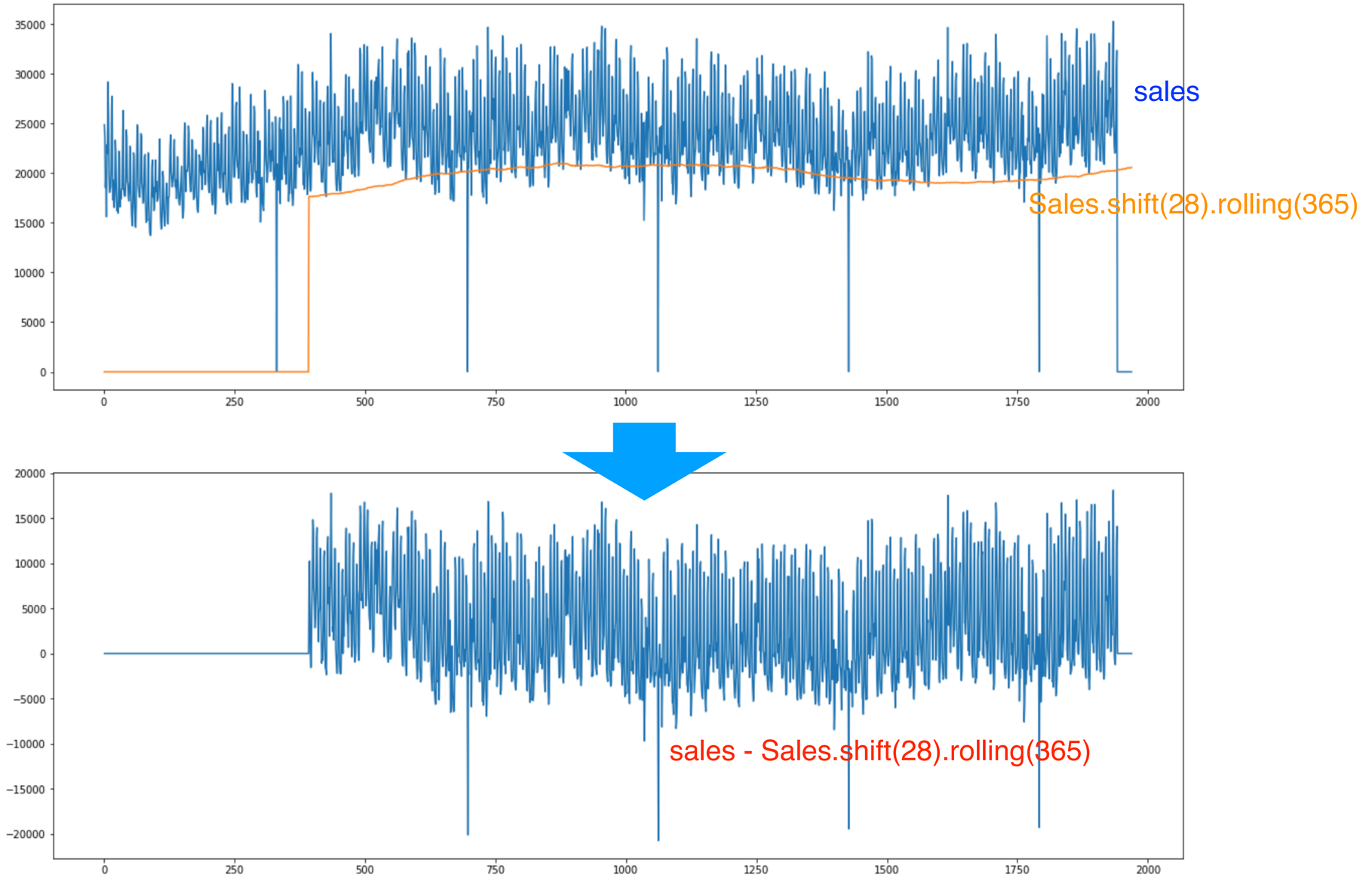
6. Training methods

(1) Removing the sales trend

I wanted to remove trends in the time series, but I didn't want to use predictions, such as machine learning, because of the risks involved.
Therefore, I used actual results.
The most stable of them all, "sales-sales.shift(28)rolling(365)" as a Target.
However, the effect is limited because it does not completely remove the trend.

Accuracy

Target = sales - sales.shift(28).rolling(365)



6. Training methods

(2) weight

Squared WRMSSE, the evaluation index, and Calculate its grad.
The coefficients of the results of that calculation were passed as weight in lightgbm.Datasets.

This efficiently compensates for errors.

Accuracy

lightgbm.Datasets(x_train, y_train, weight = myweight)

$$\frac{d}{dy}(WEIGHTS\sqrt{\frac{MSE}{SCALED}})^2 = 2 * \frac{WEIGHTS^2}{SCALED}(true - pred)$$

objective : regression

6. Training methods

(3) LearningRate & num_iterations

Depending on the Validation period and store_id, the graph shows that when the learning rate is 0.08 and 0.01, there is about 0.01 difference in the score.
When the learning rate is 0.03 and 0.01, the score is the same, but learning rate 0.01 was more stable.

A similar trend was observed for the others, and lr0.01 was chosen, although it takes considerably longer.

However, for some store_ids, the parameters are adjusted as a result of CV verification. It is blended with iter 1200 and iter 1500 for better stability.(Only CA_3 differs.)

Accuracy

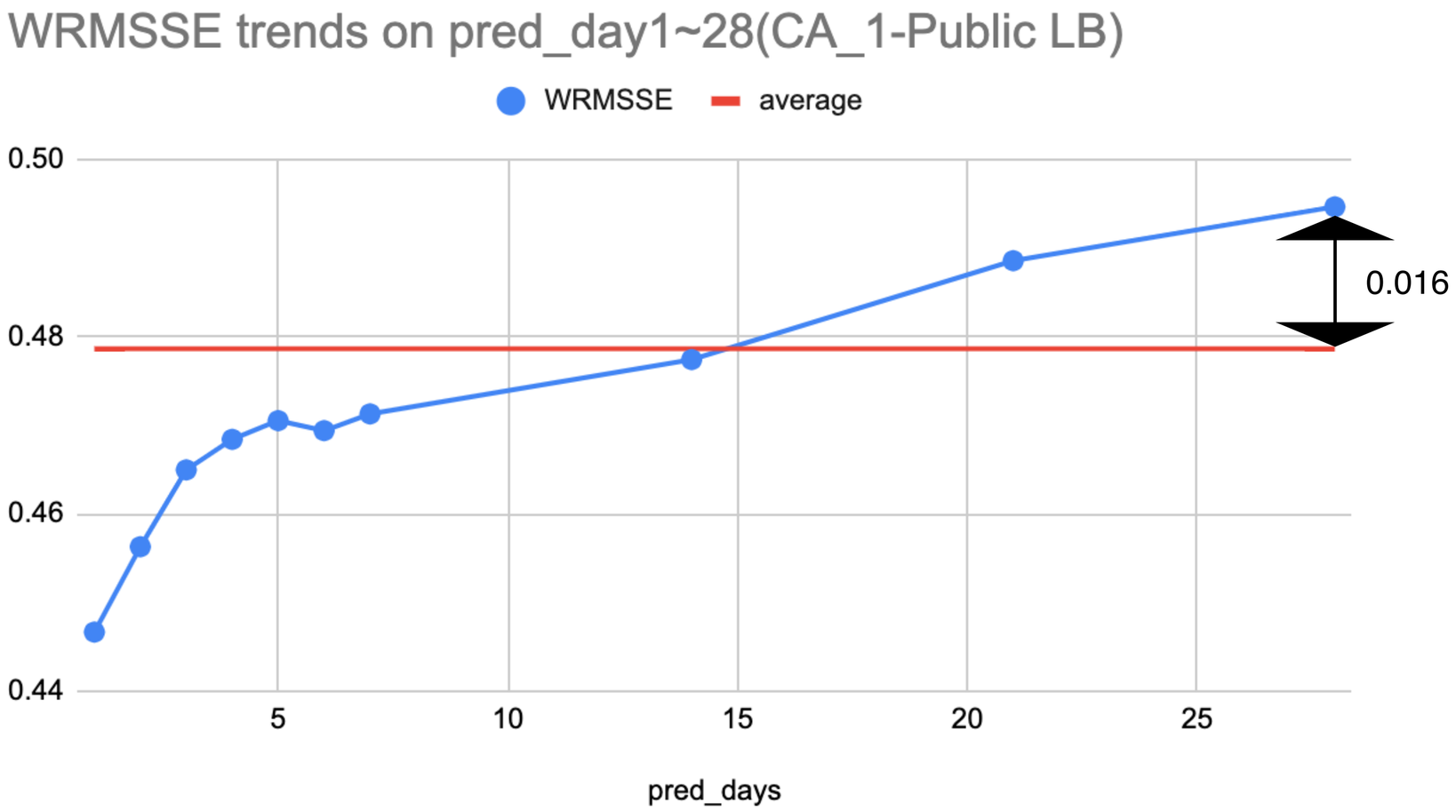
Learning Rate 0.01 was selected to achieve a more stable model.
The learning rate was blended with iter 1200 and iter 1500.

6. Training methods

(4) day-by-day modeling

The WRMSSE is better on pred_day 1 and worse on pred_day 28.
It is very important to use a day_by_day model, especially since the score variation from day 1 to day 3 is large.

Accuracy



Compared to 1 model(pred_day28), the day_by_day model has a better 0.016 score

6. Training methods

(5) others

Accuracy

- Validation
  2016-04-25 ~ 2016-05-22 : score around 0.53 (Public LB)
  2016-03-28 ~ 2016-04-24 : score around 0.51
  2016-02-29 ~ 2016-03-27 : score around 0.60
  => Use the most recent as items were being added from time to time.
  2016-05-23 ~ 2016-06-19 : score 0.576 (Private LB)

- Parameters
  Some changes according to store_id

- Metric
  Refer to (https://www.kaggle.com/girmdshinsei/for-japanese-beginner-with-wrmsse-in-lgbm)

- Non Recursive model
  The recursive approach cannot create a stable model because the errors accumulate exponentially.

- No Multiplier
  The Mulitiplier function must be built into the Model and should not be done in post-processing.

- No Post-processing

6. Training methods

In fact, I wanted to use the CV period for 6 months, but it became 3 months. Moreover, 28 models could not be made in the CV period, and five models for the 1st, 7th, 14th, 21st and 28th days were made. They came about because of a lack of time.

Uncertainty

2016-04-25 ~ 2016-05-22(28days)
2016-03-28 ~ 2016-04-24(28days)
2016-02-29 ~ 2016-03-27(28days)

3CV(84days)

Using the Accuracy model, 84 errors (true - pred) are calculated

Accuracy + 84th error => 0.995
Accuracy + 80th error => 0.975
Accuracy + 56th error => 0.835
Accuracy + 42nd error => 0.750

Accuracy Final Submission => 0.500

Accuracy - 42nd error => 0.250
Accuracy - 56th error => 0.165
Accuracy - 80th error => 0.025
Accuracy - 84th error => 0.005

The prediction model for day 1 to day 28 is used to calculate and expand the error for each.

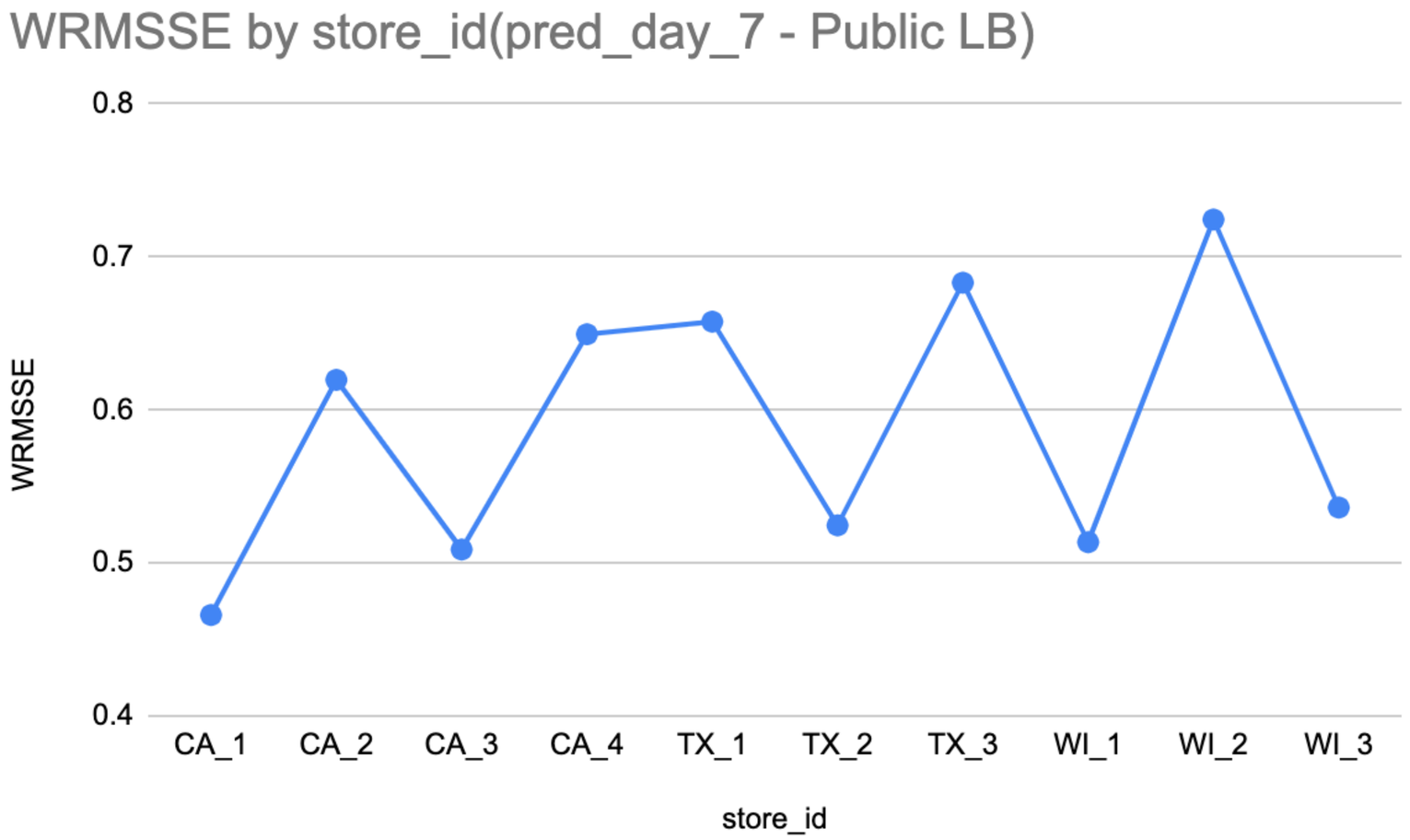
kaggle™

Supplementary Information

6. Training methods

(5) Model by store_id

Accuracy



Each store_id has a different score.