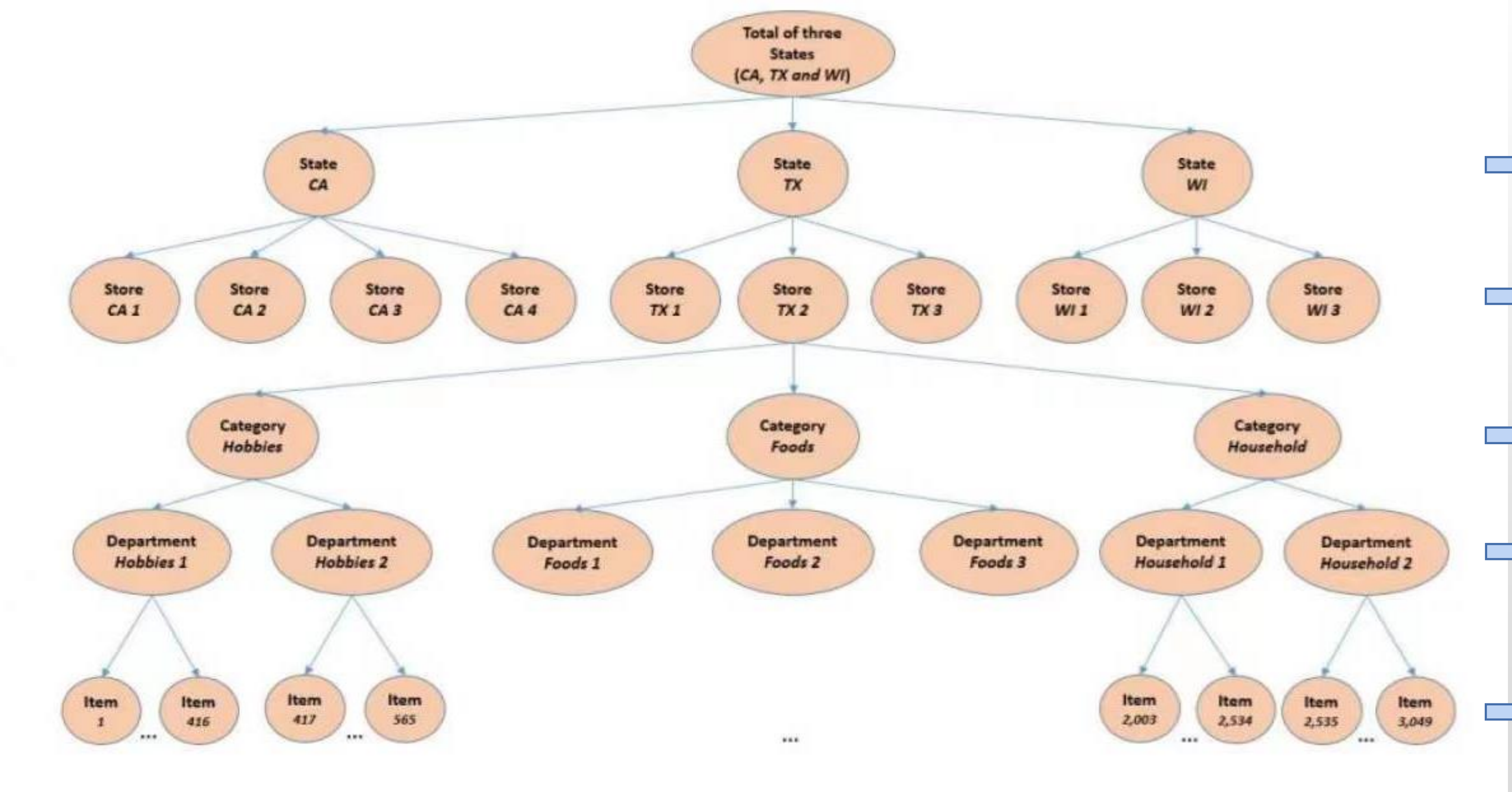
Analysis on M5 Forecasting - Accuracy

Huang Zhenyu 20744676
Luo Jiahao 20744418
Yang Yannan 20746131
Kaggle team: math6010zluoylinie

Data description



States

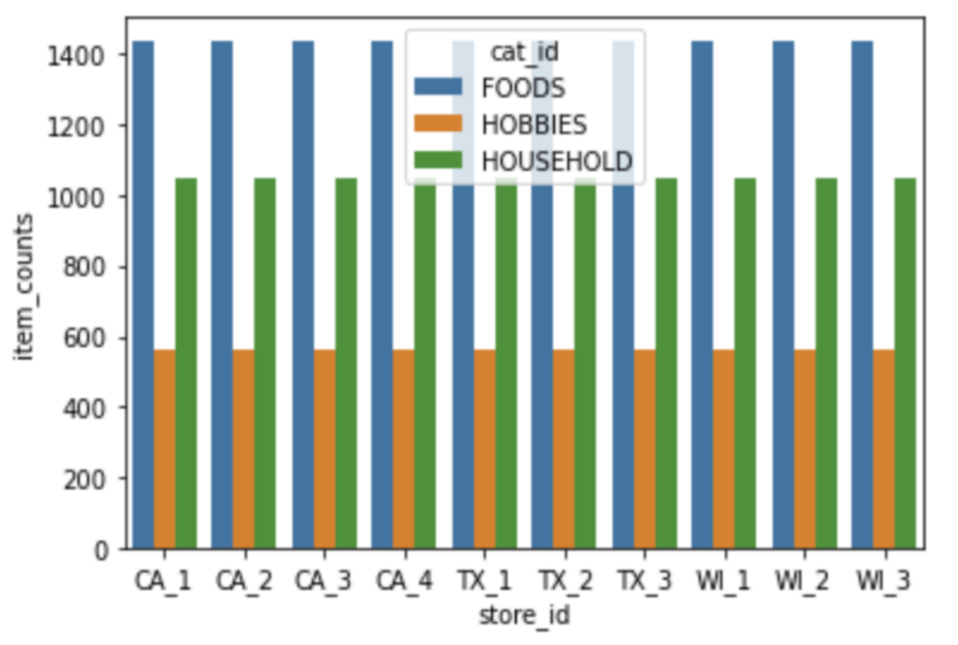
Stores

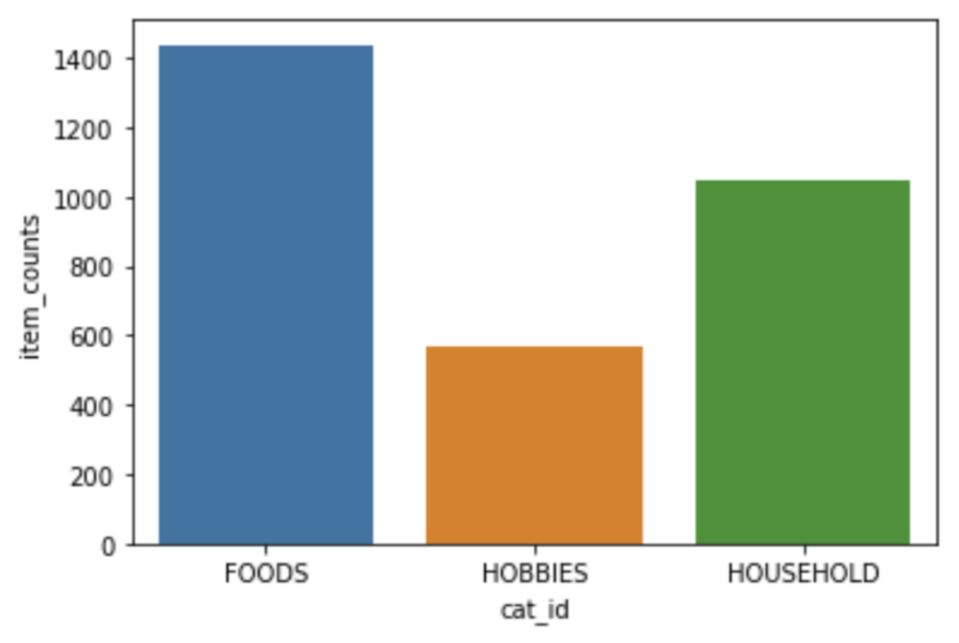
Product Categories

Departments

Items

Item counts in each store and category

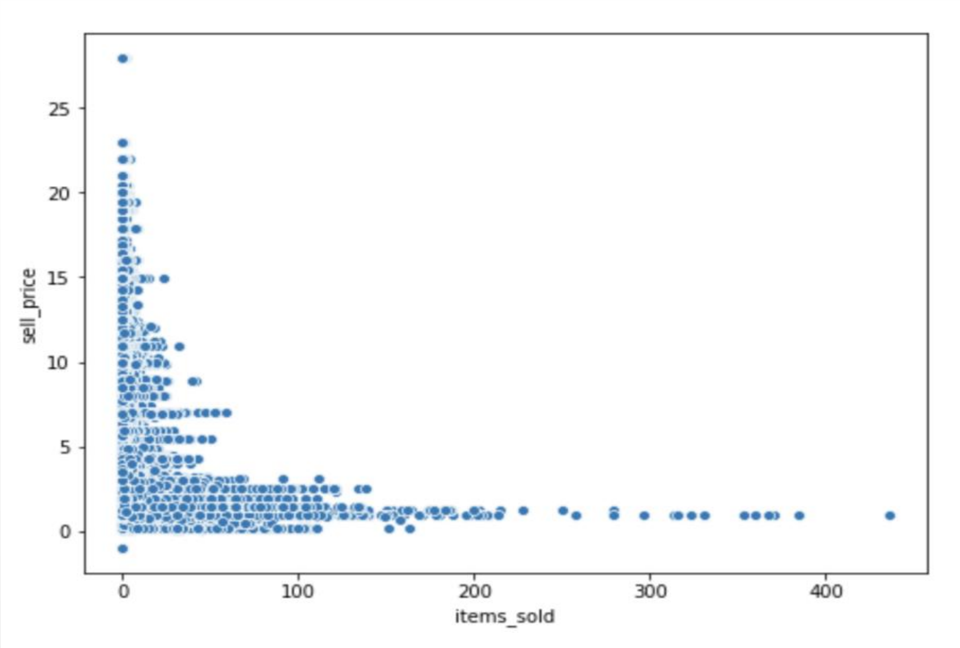




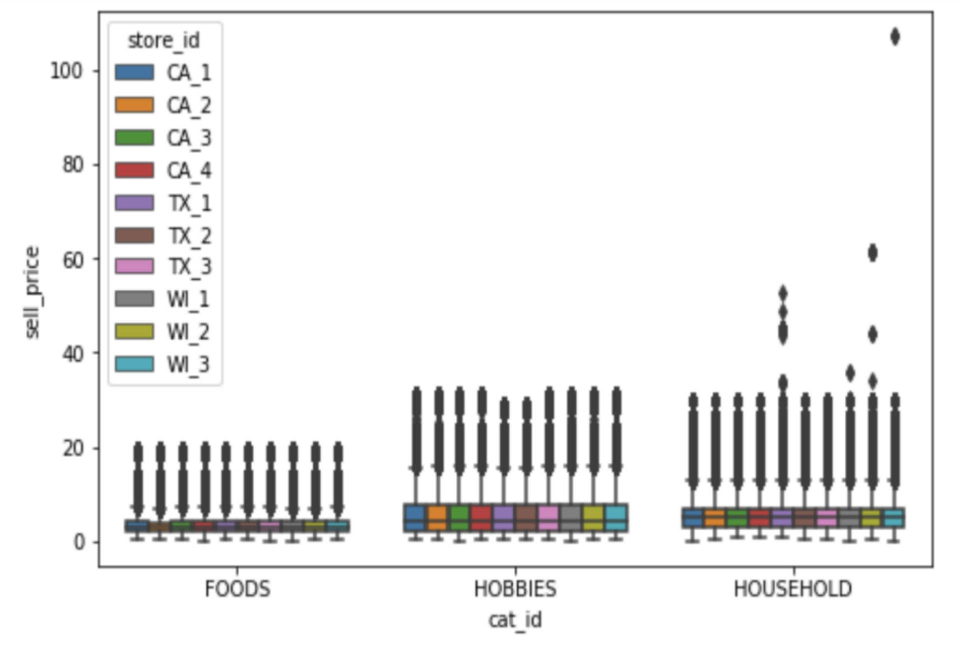
Item counts in each store and category

1437 categories of food goods, 1047 categories of household goods and 565 categories of hobbies. The category of food goods is the largest, followed by household goods and hobby goods.

Sales and sell price



Distribution of items sold and sell price

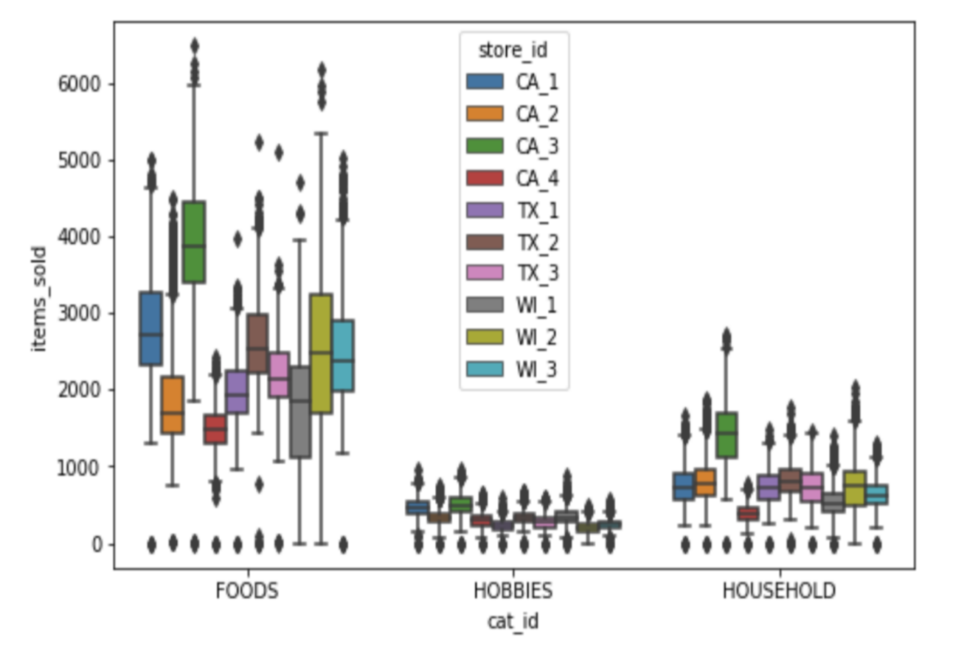


Sell prices in different stores and different categories

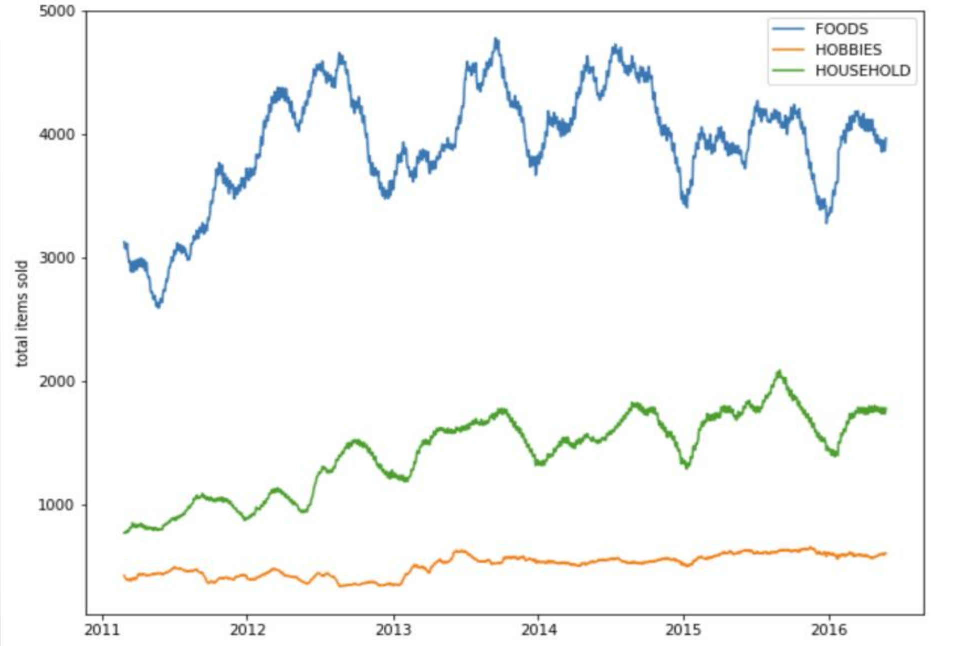
Valid Data: **79.51%**
Items sold at valid prices: **31.55%**

Sell Prices:
HOUSEHOLD > HOBBIES > FOODS

The sales aggregated by category



The sales in each category and store

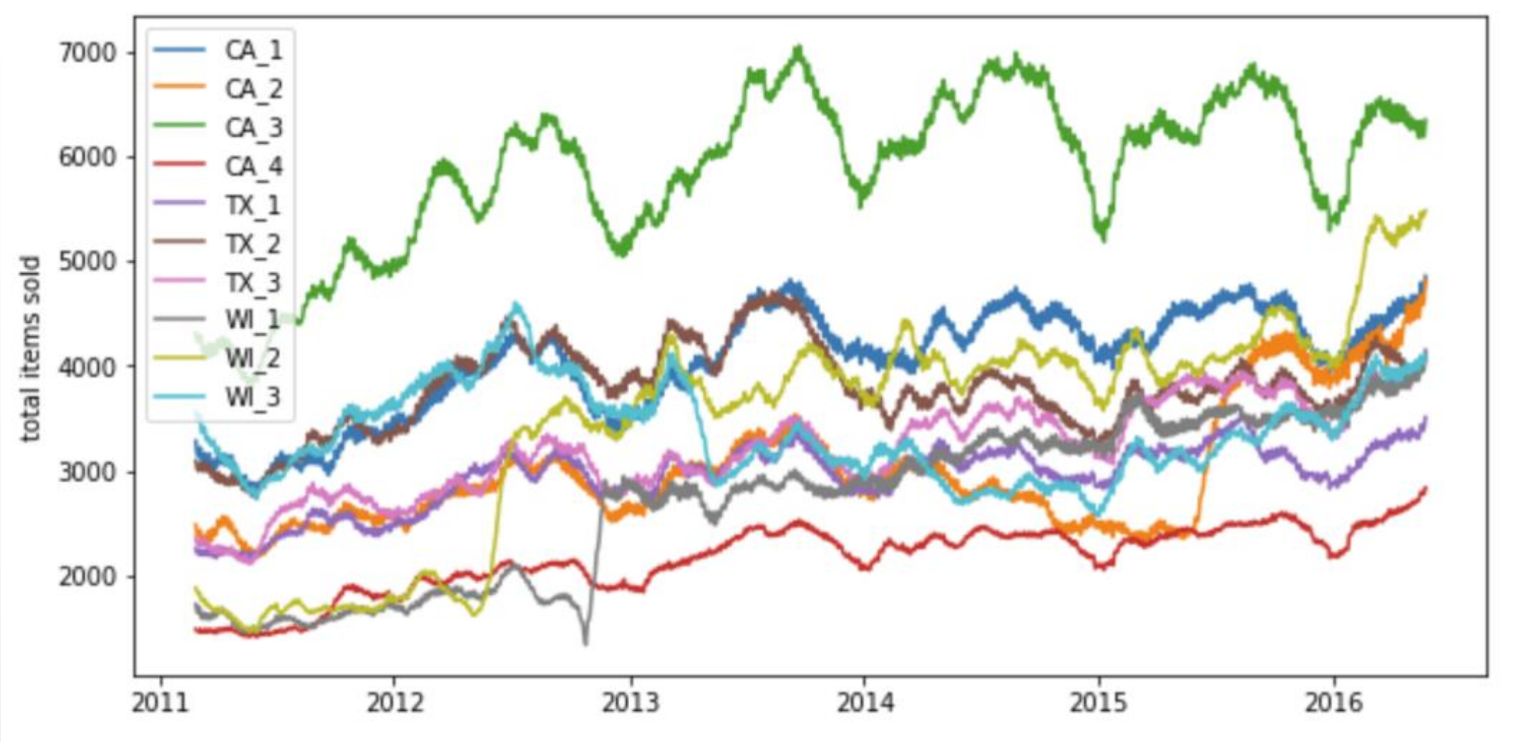


Total items sold of each category in different years

People in different states have significantly different levels of food consumption

There may also be seasonal effects in people's consumption behavior of food commodities

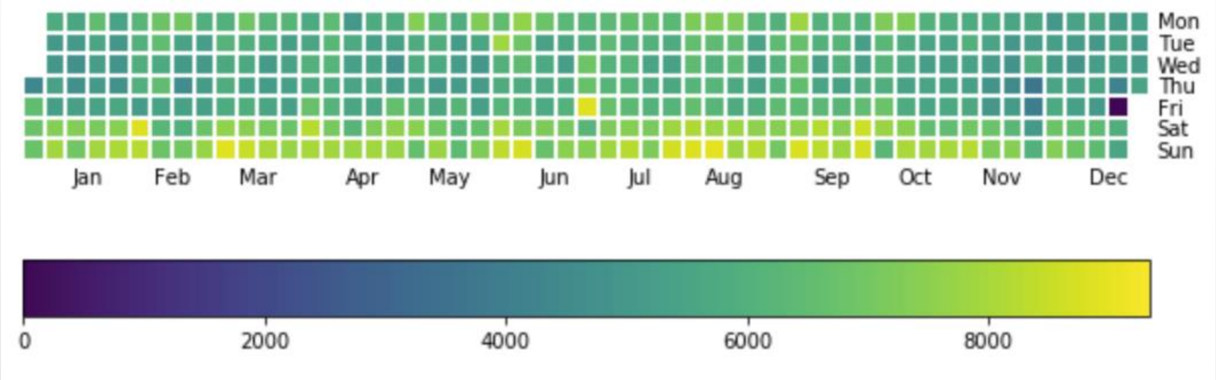
The sales aggregated by stores



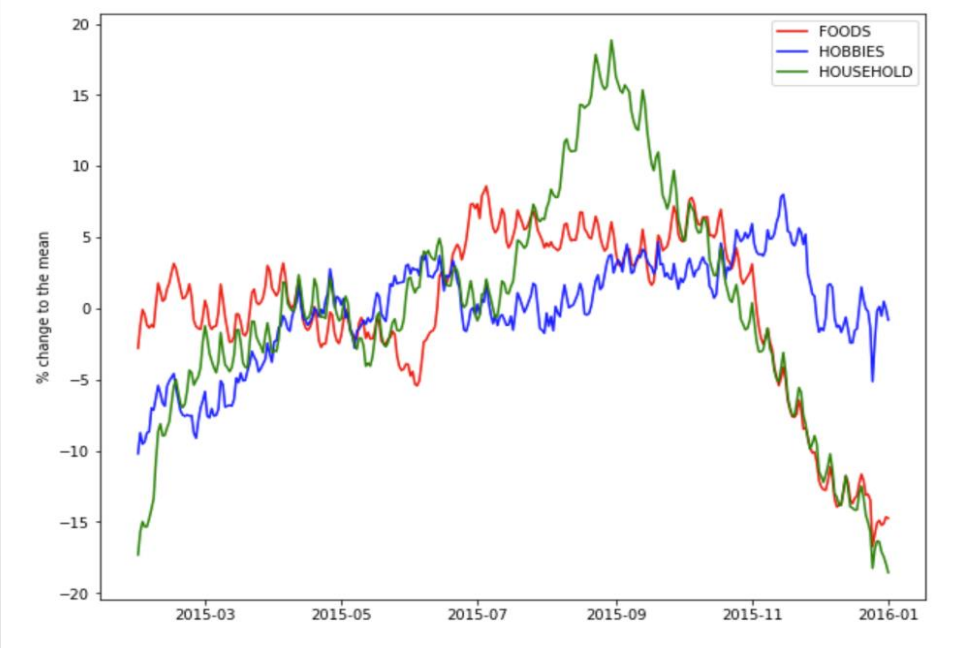
The total items sold of each store in different years

The annual sales volume has a certain cyclical change and the sales volume often peak in around September every year

Holiday Season Investigation



The heatmap of sales data of CA_3



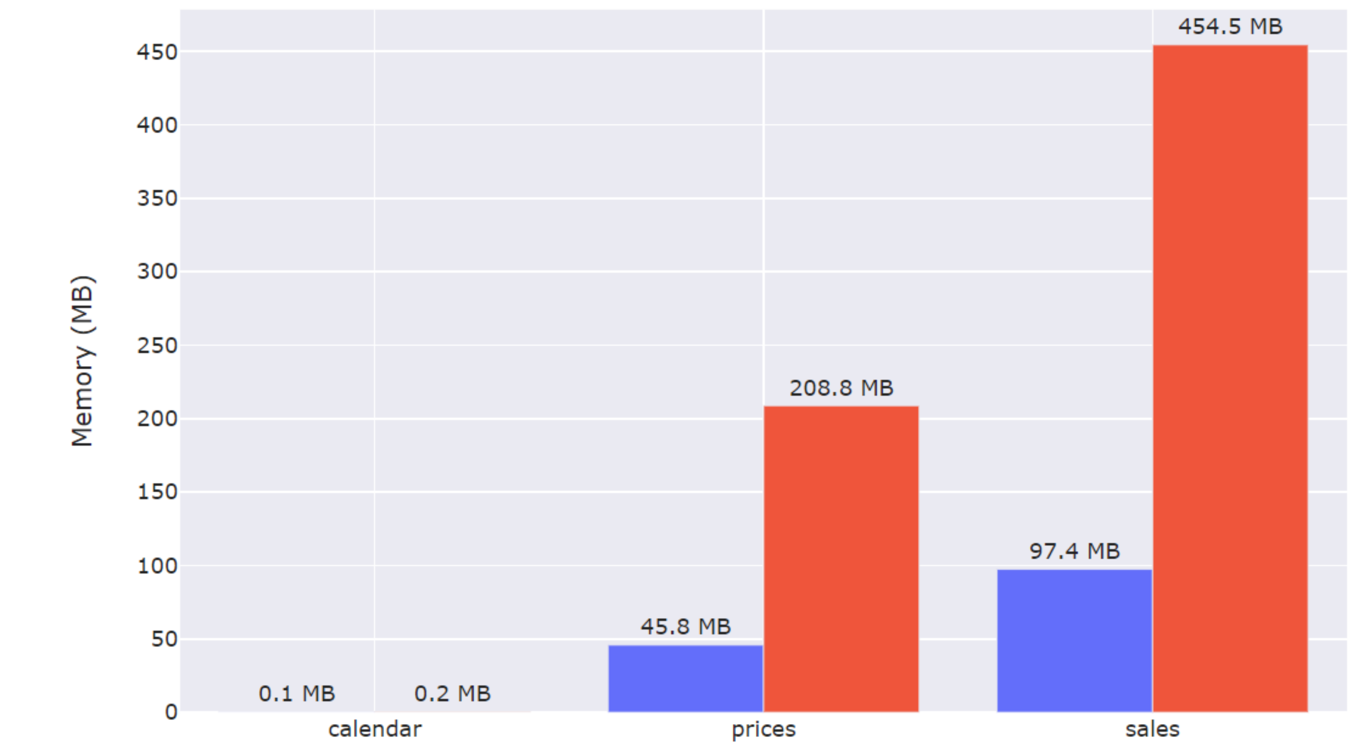
Percentage change of 30-day rolling mean between 2015-2016

This could mean that people shop more frequently during **holidays and weekends**

Data Processing

Reduce memory consumption

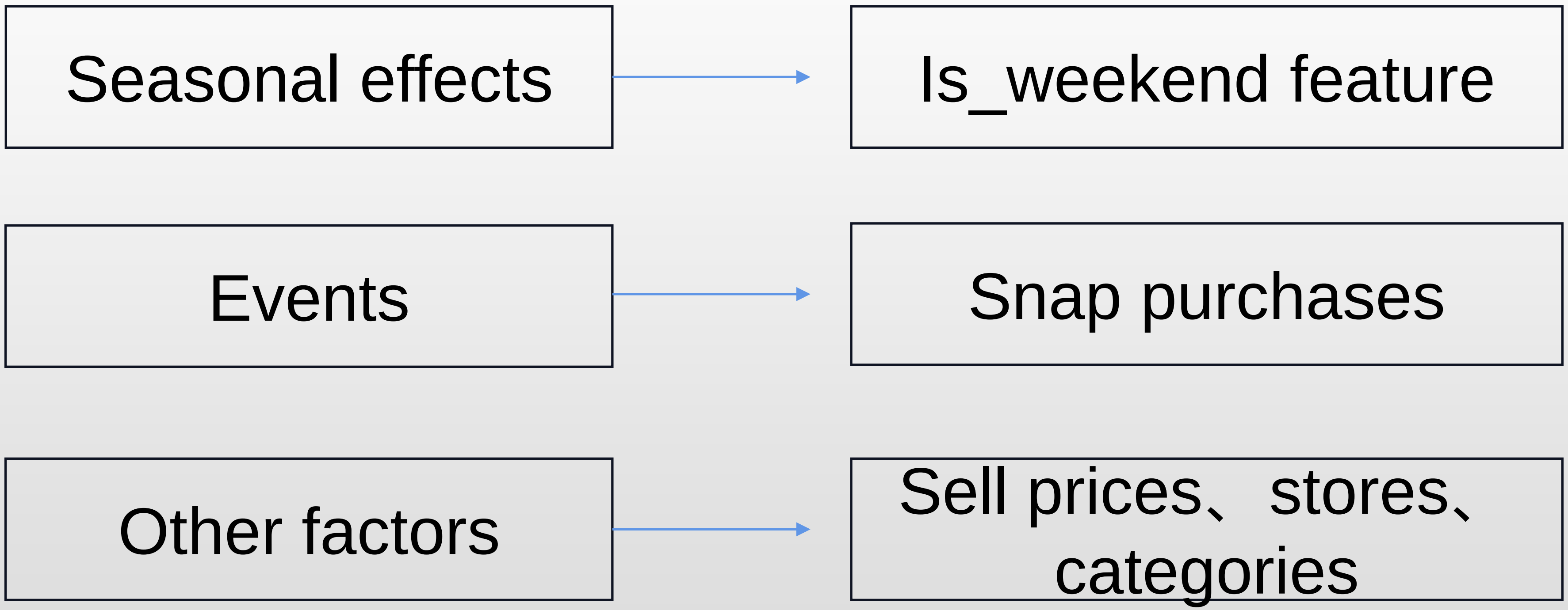
- some features do not require 16-bit or 32-bit to storage and perform



Melt and combine the data

- the dataframe contains daily sales data with days(d_1-d_1969) as columns

Feature selection



Rolling window method

The size of the window is constant

the window slides move forward constantly

We consider only the most recent values and ignore the past values

Training

M5

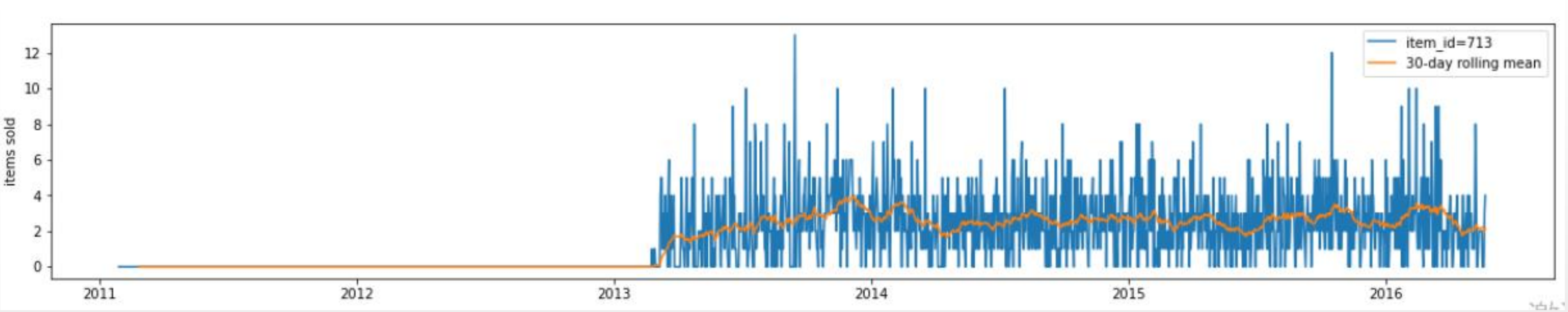
Table 1: Number of M5 series per aggregation level.

Level id	Aggregation Level	Number of series
1	Unit sales of all products, aggregated for all stores/states	1
2	Unit sales of all products, aggregated for each State	3
3	Unit sales of all products, aggregated for each store	10
4	Unit sales of all products, aggregated for each category	3
5	Unit sales of all products, aggregated for each department	7
6	Unit sales of all products, aggregated for each State and category	9
7	Unit sales of all products, aggregated for each State and department	21
8	Unit sales of all products, aggregated for each store and category	30
9	Unit sales of all products, aggregated for each store and department	70
10	Unit sales of product *x*, aggregated for all stores/states	3,049
11	Unit sales of product *x*, aggregated for each State	9,147
12	Unit sales of product *x*, aggregated for each store	30,490
	Total	**42,840**

Should be more accurate if we predict the future sales by store, state, category and department and combine all these together

A total of 12 levels of sales information

Reduce training scale



Some products are not sold in early periods

	0
count	30490.000000
mean	406.194490
std	477.176658
min	0.000000
25%	1.000000
50%	159.000000
75%	766.000000
max	1845.000000

Around 25% of the products only sold after 2013

Only use data after 2013 to reduce memory usage and save time

Adjust the loss function

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

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

Use Poisson loss since we have a large amount of 0s
Assume the number of sales follows Poisson distribution

Handling missing values

No sales data ➡ No sell price data

So there must be missing values in the fully connected dataset

Remember no sales does not mean sell price = 0

Currently a better solution: Leave it as is

Model selection

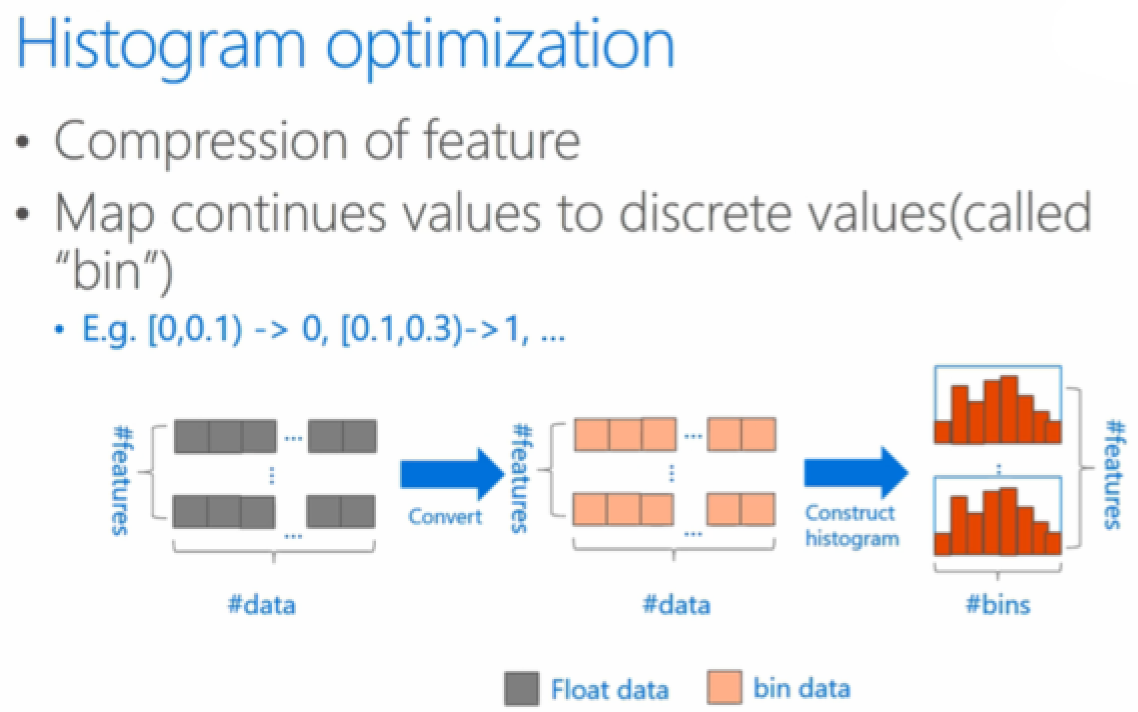
Use a model that can deal with missing values and large amount of 0s: Tree-Based method

Simple methods like time-series AR or linear regression are not considered in out work

We mainly consider the methods combining faster computation and lower error: LightGBM, Catboost and Xgboost

LightGBM

Histogram algorithm



Supports categorial features and missing values

Full model

Features

Sales features

- sales_lag_{s|s+1|…|s+14}
- rolling_mean_{7|14|30|60|}
- rolling_std_{7|14|30|60|}
- release

Calendar features

- tm_{d|dw|w|w_end|wm|m|y}
- snap_{CA|TX|WI}

Price features

- price_{max|mean|min}
- price_{std|norm|nunique}
- price_cent_{max|min}
- price_momentum_{d|m|y}

Id features

- item_id, cat_id, dept_id
- enc_item_id_{mean|std}
- enc_cat_id_{mean|std}
- enc_dept_id_{mean|std}

10 stores

store=CA_1

store=CA_2

store=CA_3

store=CA_4

store=TX_1

store=TX_2

store=TX_3

store=WI_1

store=WI_2

store=WI_3

Category department

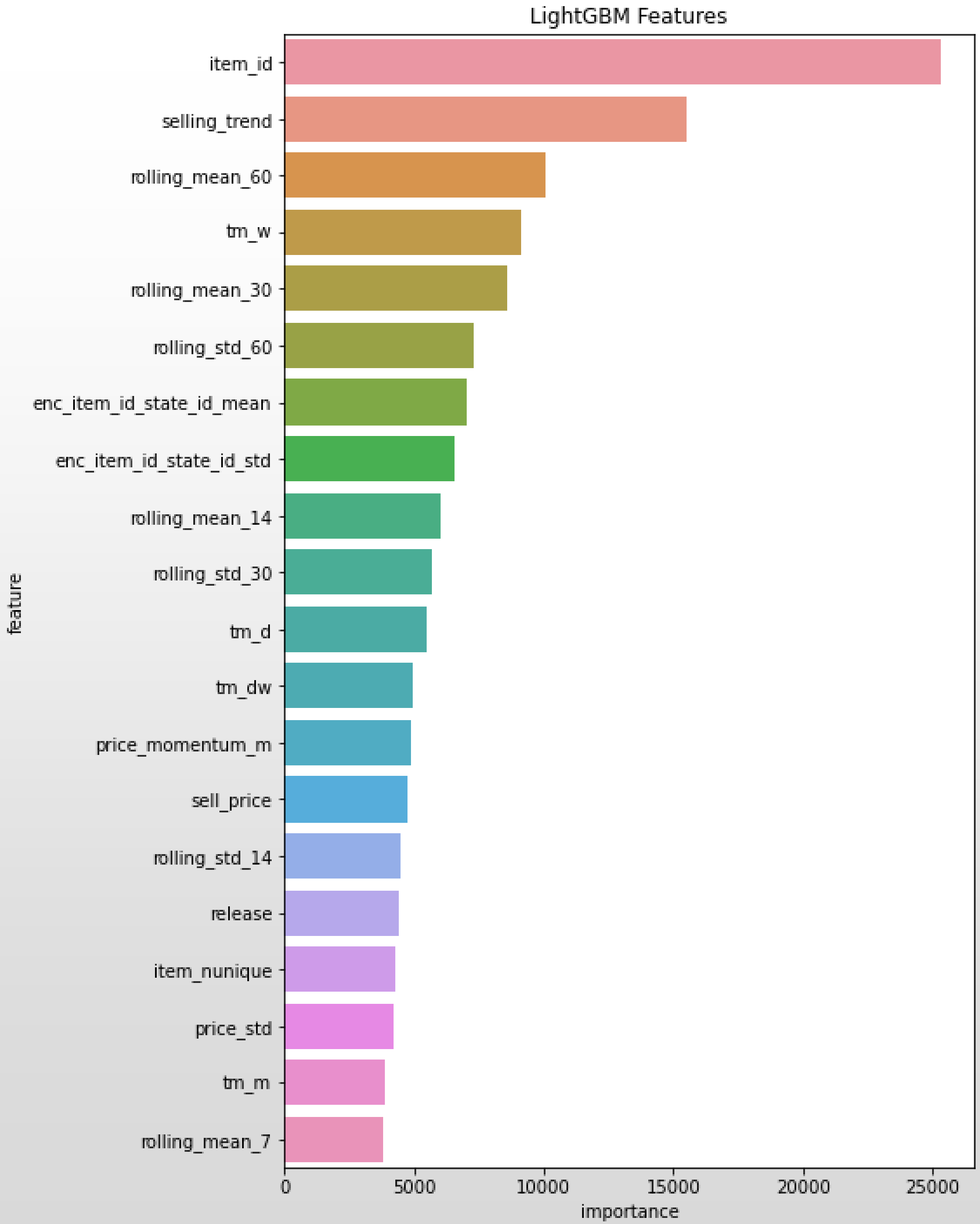
['HOBBIES_1',
'HOBBIES_2',
'HOUSEHOLD_1',
'HOUSEHOLD_2',
'FOODS_1',
'FOODS_2',
'FOODS_3']

Practical/Simple solution

- no blending/stacking
- no recursive modeling
- no postprocessing/multiplier

Results

Item_ID and rolling statistics for 30 and 60 days seem to be the most important features among all



Further Imporvement

Try different methods and average all the results together, like LSTM

Try week-by-week methods to overcome overfitting problem

Combine PCA or other methods to reduce collinearity

THANKS!