M5 Forecasting - Accuracy

He Weiwei
wheal@connect.ust.hk

Li Xintong
xlifw@connect.ust.hk

Ma Jingkun
jmabg@connect.ust.hk

Xu Tongcan
txuav@connect.ust.hk

Abstract

Department stores like Walmart have uncountable products and money transactions every day. To forecast the unit sales of various products sold in the USA by Walmart, we first analyze the data and establish some data visualization. After selecting 16 most important features, we tried 6 different machine learning models(Linear regression, Lasso, KNN, Random Forest, Gradient Boosting and Neural Networks) and chose gradient boosting as our final model, which reaches top 10% on the private leaderboard.

1. Data Processing and Data Description

If we want to solve the problem through machine learning, we must first analyze the data and establish some data visualization, which is helpful to feature engineering.

1.1 Basic information about data

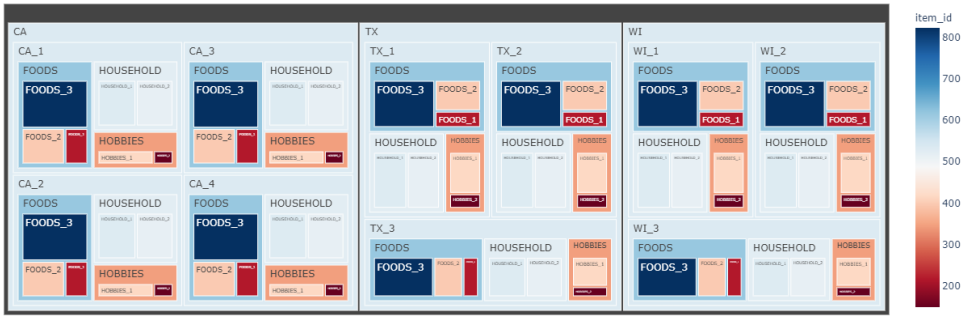


figure1 Basic information about data

As can be seen from the figure, the data has 3 states, 10 stores, 3 categories, 7 departments and 3,049 products.

1.2 The relationship between average sales and some characteristics

1.2.1 Calendar

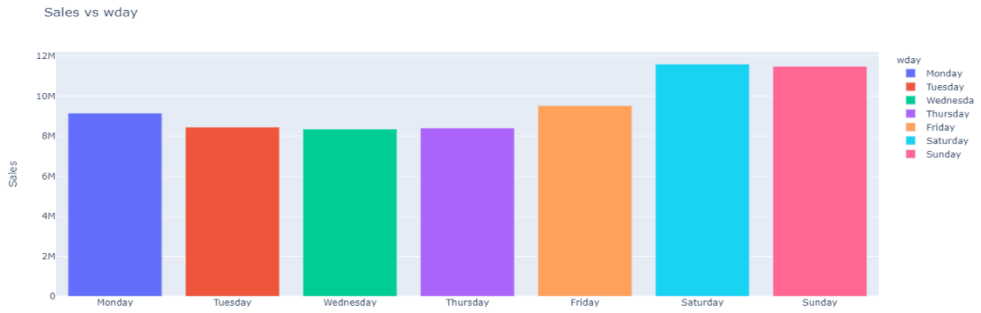


figure2 The average sales vs wday

We can see that the average sales volume gradually declined from Monday to Wednesday, then slowly recovered, and fell back on Sunday. At the same time, weekend sales were significantly higher than midweek. Therefore, in feature engineering, we choose to convert the 1-7 variables contained in ‘wday’ into the binary variable in ‘is_Weekend.’

figure3 The average sales vs events

We classify the number of events each day and calculate the average sales. We find that the date with multiple events has the highest sales volume, and the date with one event has the lowest sales volume. At the same time, we found that the degree of impact on sales is also different due to different events.

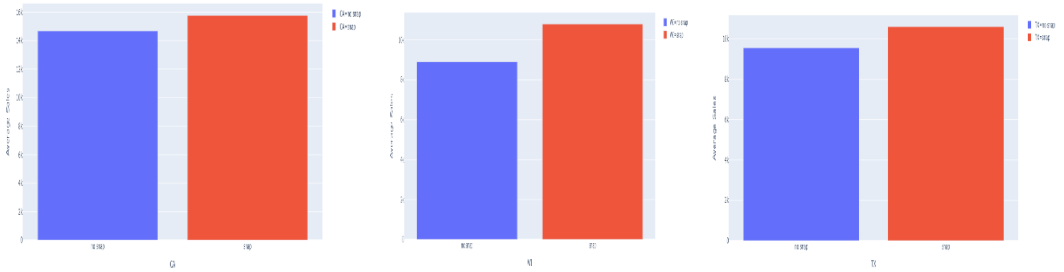


figure4 “SNAP” day vs “No_SNAP” day in three states

We found that the average sales of the "SNAP" day in these three states were significantly higher than the average sales of the "No_SNAP" day.

1.2.2 Stores

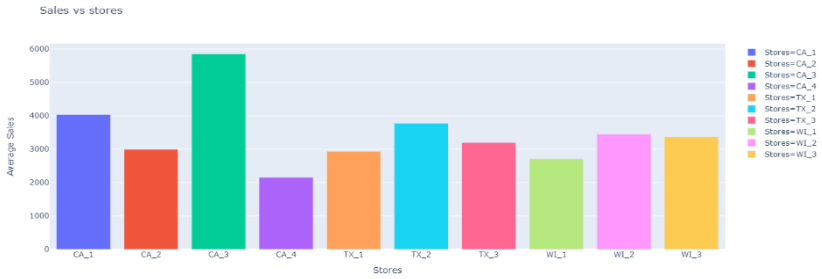


figure5 The average sales of different stories

We can see that the sales volume of CA_3 stores is the highest, and the sales volume of each store is also significantly different.

1.2.3 Categories and Departments

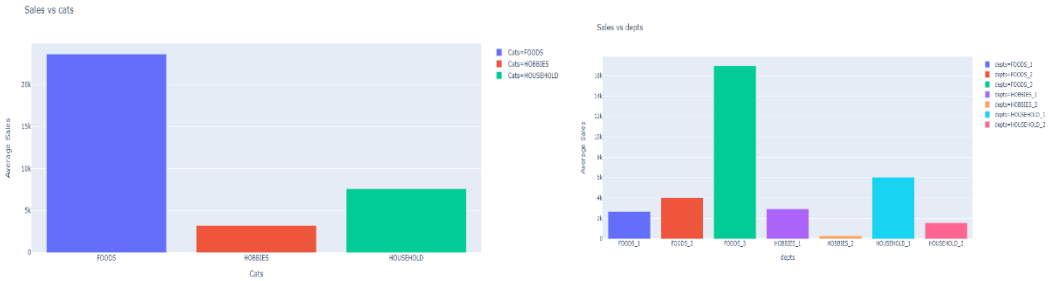


figure6 The average sales of different categories and departments

For different categories, FOODS has the highest sales volume and HOBBIES has the lowest sales volume. At the same time, FOOD_3 has the highest sales volume, and HOBBIES_2 has the lowest sales volume. There are significant differences in each department.

Therefore, for the existence of these differences, we use the 'LabelEncoder ()' function to process 'cat_id','dept_id', 'store_id' and 'state_id' respectively in the actual Feature engineering.

1.2.4 Rolling Averages sales

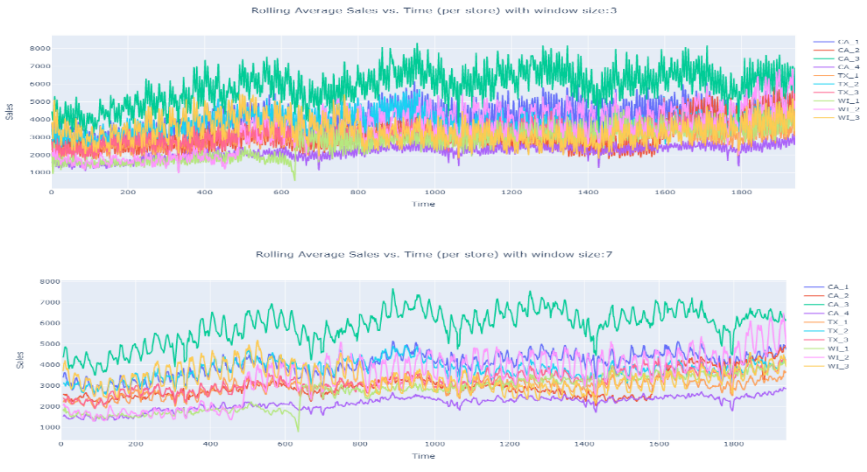


figure7 Rolling average sale with window=3,7,14,21,28

For time prediction models, Rolling Averages sales are also an important feature. Here we choose window=3, 7, 14, 21, 28 and visualize the data. We found that each curve has an upward linear trend. The difference between CA stores is more significant, and the difference between WI and TX stores is small, which may mean that there are differences in the development speed of various regions within the same state.

Finally, we draw the correlation heatmap of the features and in the actual modeling process, select some important features.

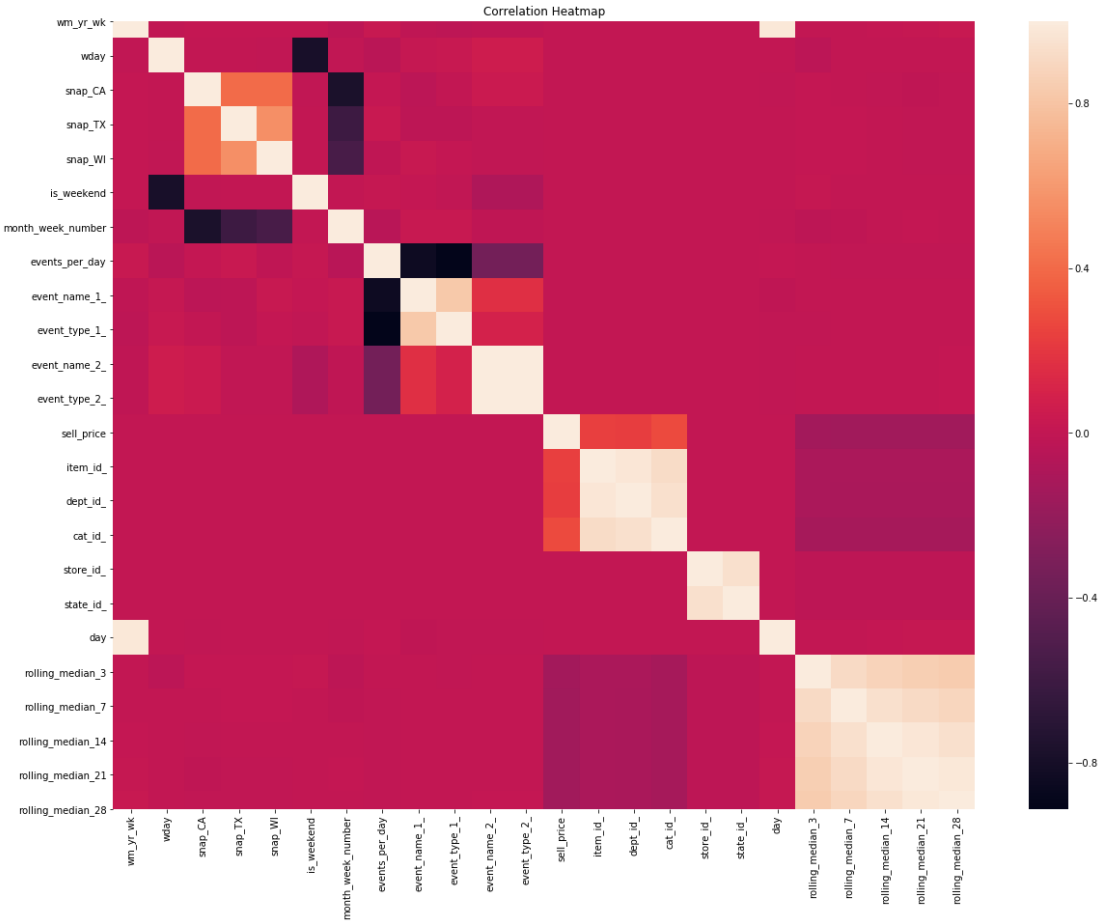


figure8 The correlation heatmap of the features

2. Model Selection

2.1 Regression

Linear regression is one of the most basic machine learning models. It is used for finding linear relationship between target and one or more predictors. We choose data from d_1001 to d_1913 as training data. The r2 score we got from the model is 1.0, while the private score and public score in Kaggle after we submit the data are 5.39065 and 0.00070.

2.2 Lasso

We use Lasso to train the data and do it in two ways. The first one is that we use whole validation data to be the in-sample data and do the lasso for every item. The train score is almost 1 which indicates that it may be overfitting. Then the Kaggle score is 0.00984 (public score) and 5.35232 (private score). The second way is that I combine some lasso with lags to do the training, for example, we choose the last 100 days to do the training first and the last 200 days to the last 100 days to be the second training sample and so on. After validation we choose 50 days to be the training window and train the data for 3 times. Then weight the three predictions by regression. The Kaggle score slightly improves which is 0.0809 for public score and 5.09942 for private score.

2.3 Knn

In this patt, we use the second way we used in the Lasso. By validation, we choose the training windows to be 100 and combine three Knn model predictions using regression to get the final result. The Kaggle score is 1.03715 for public score and 1.03073 for private score. We can see that the non-linear model is better than the linear model. Then we decided to combine Lasso and Knn to see if it could be better. However, the score is just 0.89409 for the public score and 1.79560 for the private score.

2.4 Random Forest

Random forest regressor is an ensemble technique which uses multiple decision trees and a technique called bootstrap and aggregation commonly known as bagging. The basic idea behind this is to combine multiple decision trees in determining the final output rather than relying on individual decision trees. Best hyper-parameters are max_depth=5, min_samples_leaf=20, n_estimators=80 after hyper-parameter tuning. But the private score and public score in Kaggle after we submit the data are only 1.31560 and 1.40682.

2.5 Gradient Boosting

Boosting creates weak learners by sequentially coordinate descent. And gradient boosting is one of the most commonly used boosting methods, especially with large and complex data. So we also tried gradient boosting on the M5 dataset using Lightgbm library. We trained a gradient boosting method for each id by iteration because we think this will get higher accuracy. Below is the figure of the distribution of **root mean squared error (RMSE)** of 610 ids by uniform sampling of the test set. While the average RMSE of these ids is 1.55, we can see that nearly 400 ids' RMSE is less than 1.5. Considering the pitfall of overfitting, we submitted the result to Kaggle and got 0.69530 for private score and 0.68616 for public score.

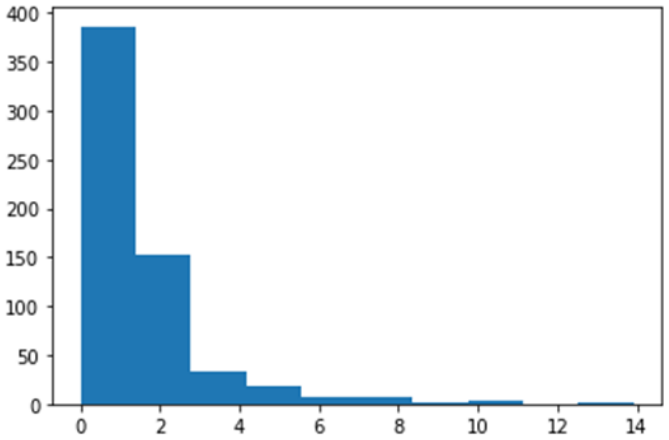


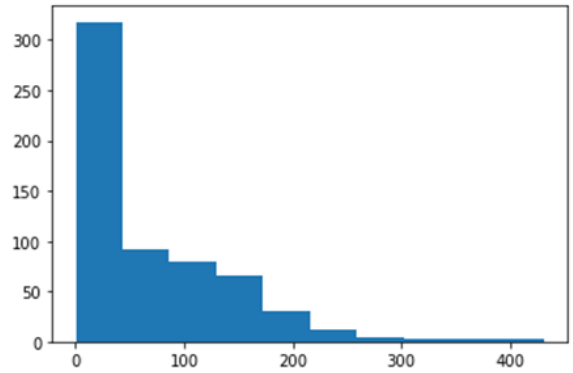
figure 9 distribution of RMSE of gradient boosting

2.6 Nueral Networks

Neural networks (NNs) are comprised of layers and nodes in each layer and can recognize hidden patterns and correlations in raw data. However, they are harder to be

tuned. We constructed two kinds of neural network models and train each id two neural network models. One with 2 layers(nn2) and the other with 4 layers(nn4).

We first tried nn2 with node information (32,16) and nn4 with node information (32,16,8,8), unfortunately the result was pretty bad (private score of nn2 is approximately 15 and that of nn4 is near 11). Then we decreased the number of nodes and tried nn2 with node information (16,8) and nn4 with node information (20,16,8,6) and the result is better but still not good. (The private score of nn2 is approximately 6 and that of nn4 is near 9) Moreover, when we further decrease the node to (8,4) for nn2 and (10,8,8,6) for nn4, the result becomes worse. Because train neural network models for each id (total 30490 ids) is time consuming. We chose the second try as our final result. Below is the figure of the distribution of **root mean squared error (RMSE)** of 610 ids by uniform sampling of the test set. While most of the RMSE is in lowest quantile, there are ids with extremely large RMSE (RMSE over 100). And we can also see that nn4 did a greater job than nn2. So for future improvement, further increasing the number of layers may be one of the choices.



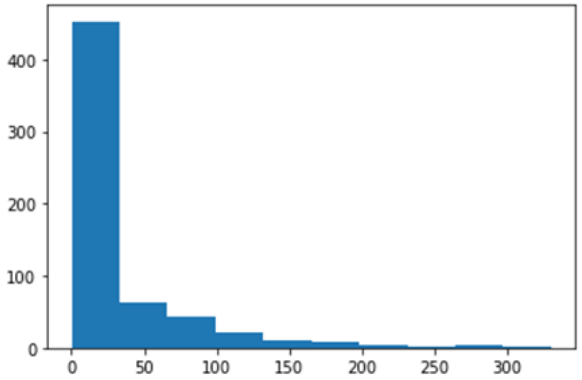


figure 10 distribution of RMSE of nn2

figure 11 distribution of RMSE of nn4

3. Conclusion

Model Name	Private Score	Public Score
Linear regression	5.39065	0.00070
Lasso	5.09942	0.08090
Knn	1.03073	1.03715
Random Forest	1.31560	1.40682
Gradient Boosting	0.69530	0.68616
Nueral Networks-2	9.95840	9.95426
Nueral Networks-4	6.43311	6.53994

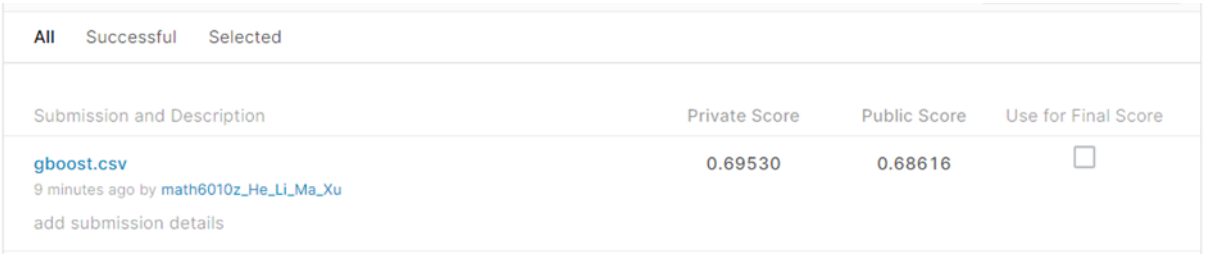


figure 12 scores on Kaggle of gradient boosting prediction result

Overall, Gradient Boosting performs better than other models with 0.69530 in private score. Because the competition is finished, we can not get our result to show up on the final leaderboard. But we can estimate that our final ranking is about 539, the top 10% on the private leaderboard. We can also see that the lower the public score, the higher the private score which means that there is overfitting when we train those models with low degree of freedom. Another reason is that the testing data may deviate a lot from the training data, so the non-linear models perform better than linear models.

4. Contribution

The contribution of each person is as follows:

Name	Contribution
Li Xintong	Gradient Boosting & Neural Network: model and report
Ma Jingkun	Modeling and report
He Weiwei	Linear Regression and Random Forest: model and report
XU Tongcan	Data Processing and Data Description: Feature engineering and report