

IST 557 Data Mining: Techniques and Applications

Project A. Future sales

Team: IST557_A4_2019

Name: Ting-Yao Hsu, Szu-Chi Kuan

1. Introduction

The task we need to do in this project is to forecast the total amount of every product sold in every shop in the next month. And the datasets which be provided are included daily historical sales data. The detailed description of the dataset as shown below.

1.1 File Description

1. sales_train.csv - the training set. Daily historical data from January 2013 to October 2015.
2. test.csv - the test set. We need to forecast the sales for these shops and products for November 2015.
3. sample_submission.csv - This is a sample submission file in the correct format.
4. items.csv - This is supplemental information about the items/products.
5. item_categories.csv - This is supplemental information about the items categories.
6. shops.csv- This is supplemental information about the shops.

1.2 Data fields

1. ID - an Id that represents a (Shop, Item) tuple within the test set
2. shop_id - unique identifier of a shop
3. item_id - unique identifier of a product
4. item_category_id - unique identifier of item category
5. item_cnt_day - number of products sold. You are predicting a monthly amount of this measure
6. item_price - current price of an item
7. date - date in format dd/mm/yyyy
8. date_block_num - a consecutive month number, used for convenience. January 2013 is 0, February 2013 is 1,..., October 2015 is 33

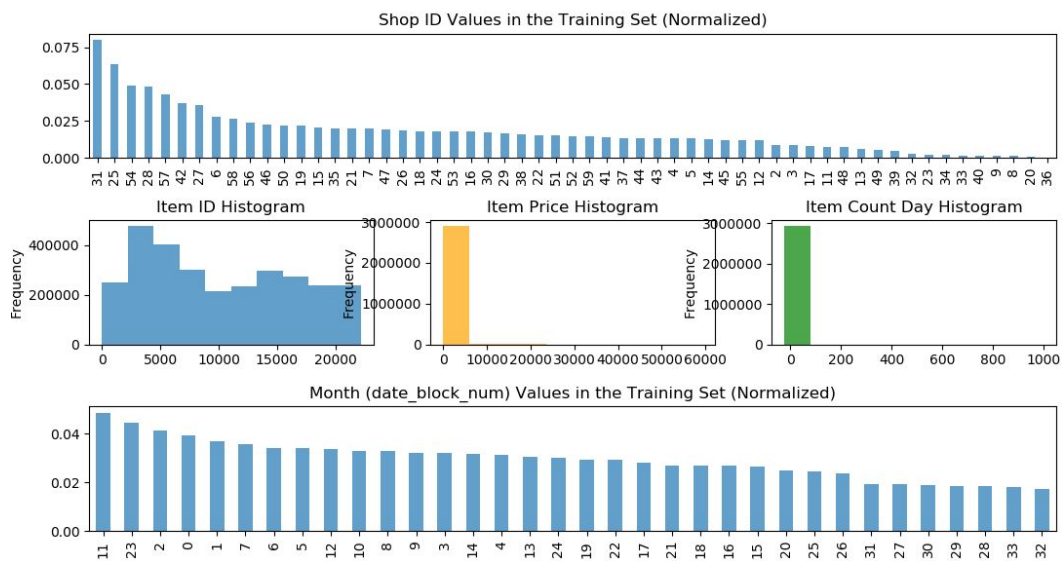
9. item_name - name of item
10. shop_name - name of shop
11. item_category_name - name of item category

2. Method

In this section, we will describe the detailed method that we produced the best performance.

2.1 Data visualization

Before doing data preprocessing, We have to check what kind of dataset we have. So we did the data visualization to understand our dataset better. The data visualization results are shown below:

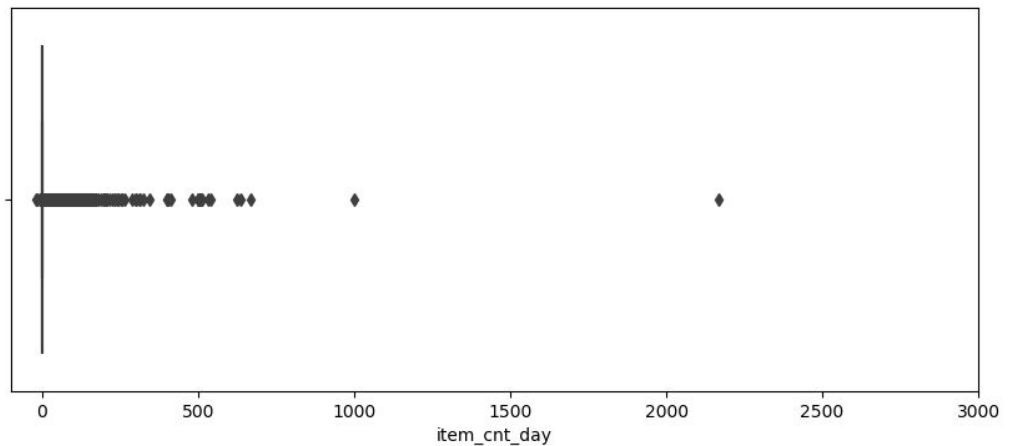
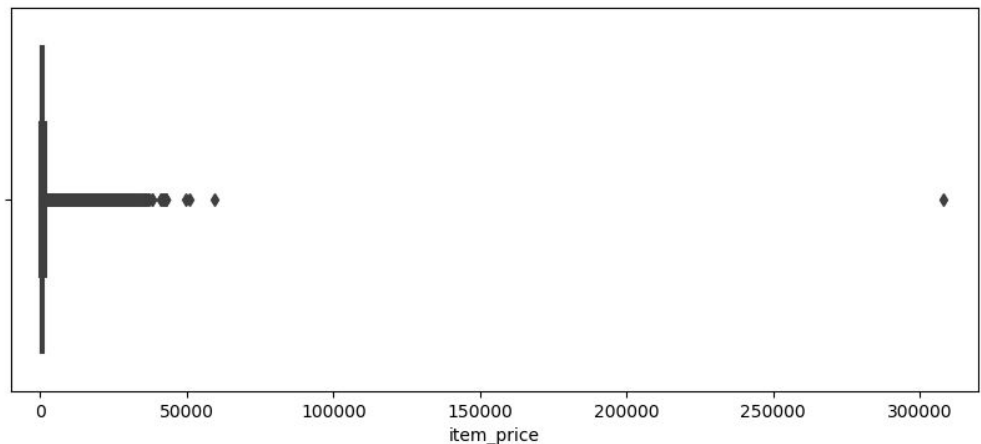


2.2 Data pre-processing

Then we first do data pre-processing.

- Deal with Outliers

The following step is checking whether there are any strange data or missing value. And we found there are items with strange prices and sales.



After detailed exploration, we decided to remove items with price > 100000 and sales > 1001

In addition, we also found that there is one item with a price below zero. We replace its price as the median price that we get from the same shop_id, item_id, and date_block_num. And we found there are several shops (according to its name) that are replicated. We also merged them.

- Shops/Category preprocessing

Each shop comes with the city name, so we retrieve city name from shop_name (ex. of original shop_name: "Philadelphia Wegmans" >> "Philadelphia" "Wegmans") and encode them.

We also retrieve the type and subtype of an item from category value and retrieve revenue for each item by multiplying item_price by item_cnt_day as well.

We compare the train data to test data, and we found that there are some new products in test data. Also, the existing shop_id and item_id pairs in the training data were past information, we might have different pairs in the future. Thus, we increase the train data to make sure we have all different pairs. We then merge the train and test data. One more thing is that we need to predict the month sales, we sum the item sales in each month.

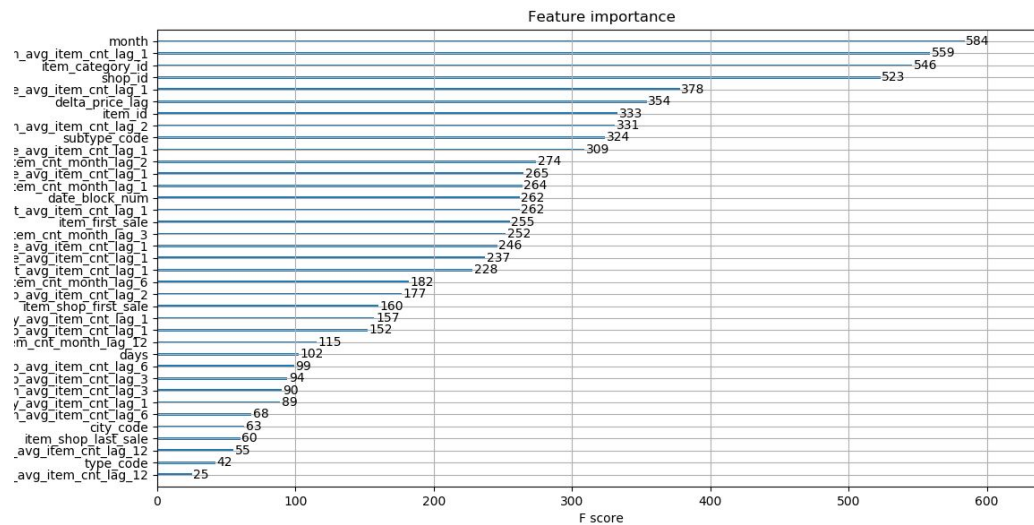
- Features engineering

Because this is a time-series problem, lag and encoded means had to be used to get more insight. First, we use lage features which is a common technique for transforming time series datasets into supervised learning datasets. For example, matrix = lag_feature (matrix, [1,2,3,6,9,12], 'item_cnt_month') refers to adding the same product to each sample last month, 2 months ago, 3 months ago and so on. The monthly sales field half a year ago and one year ago. In this way, each sample is connected to the data from its previous time, and the data has time series features. The machine learning algorithm can predict the future based on these features. Mean encoded variables can be created by using the target variable and another categorical variable present in the data set to further increase its suitability. It can decrease the possibility of random data. We also record the sales time period as new features that can connect to previous months' sales. We also use price trend features, since sales amounts and selling price are inverse, we add the price changing features for each sample. We know that new products not found in the train set will appear in test set. With the help of the revenue trend of the store, the model can predict the sales of new products.

2. 3 Model building

After feature engineering, we applied several models to get the prediction. Compared with the results, we find that the XGBoost got the best performance which the **Root Mean Squared Error = 0.90680** .

We also did the feature importance analysis. According to the plot below, the most important features in this dataset to predict if the treatment will work are 'month', 'date_item_avg_item_cnt_lag_1', 'item_category_id' and 'shop_id'.



In order to get higher accuracy, we did the parameter tuning: (max_depth: 6~10, min_child_weight: 100~1000, n_estimators: 100~1000), and use Gridsearch to find the best parameters. Following are our best parameters for XGB model: max_depth=10, n_estimators=1000, min_child_weight=1000, colsample_bytree=0.8, subsample=0.6, eta=0.3, seed=42. And we also set the model will train until validation_1-rmse hasn't improved in 10 rounds.

3. Results

3.1 The best method

We applied several models to get the prediction. Compared with the results, we find that the XGBoost got the best performance which the Root Mean Squared Error = 0.90680 .

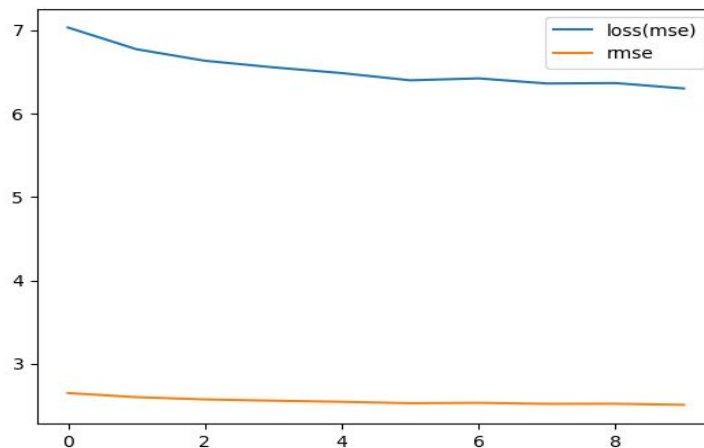
In order to get higher accuracy, we did the parameter tuning: (max_depth: 6~10, min_child_weight: 100~1000, n_estimators: 100~1000), and use Gridsearch to find the best parameters. Following are our best parameters for XGB model: max_depth=10, n_estimators=1000, min_child_weight=1000,

colsample_bytree=0.8, subsample=0.6, eta=0.3, seed=42. And we also set the model will train until validation_1-rmse hasn't improved in 10 rounds.

3.2 Other methods

We also tried three different models and compared their performance. The three models we choose are as follows:

- LSTM: we used LSTM (Long Short-Term Memory) model to do the prediction. In the LSTM model, we simply use two hidden layers and apply dropout (0.5) for each layer, then concatenating one fully connected layer for output. When training this LSTM model, we used batch_size = 4096 and epochs = 10. To sum up, we get the performance result, which is Root Mean Squared Error = 1.02138. And the figure is shown below.



- Random Forest: We did the parameter tuning to get high performance: (n_estimators: 100~1000, criterion: gini, entropy, max_depth: 3~8), and use Gridsearch to find the best parameters. Following are our best parameters for Random Forest Classifier: n_estimators=1000, criterion=gini, max_depth=6. Other parameters are default. The Root Mean Squared Error = 0.91260.
- CNN: We add three convolutional layers (output dimension = 64, kernel size = 3, "relu" activation function), one maxpooling layer and two fully connected layers. The Root Mean Squared Error = 1.10015.

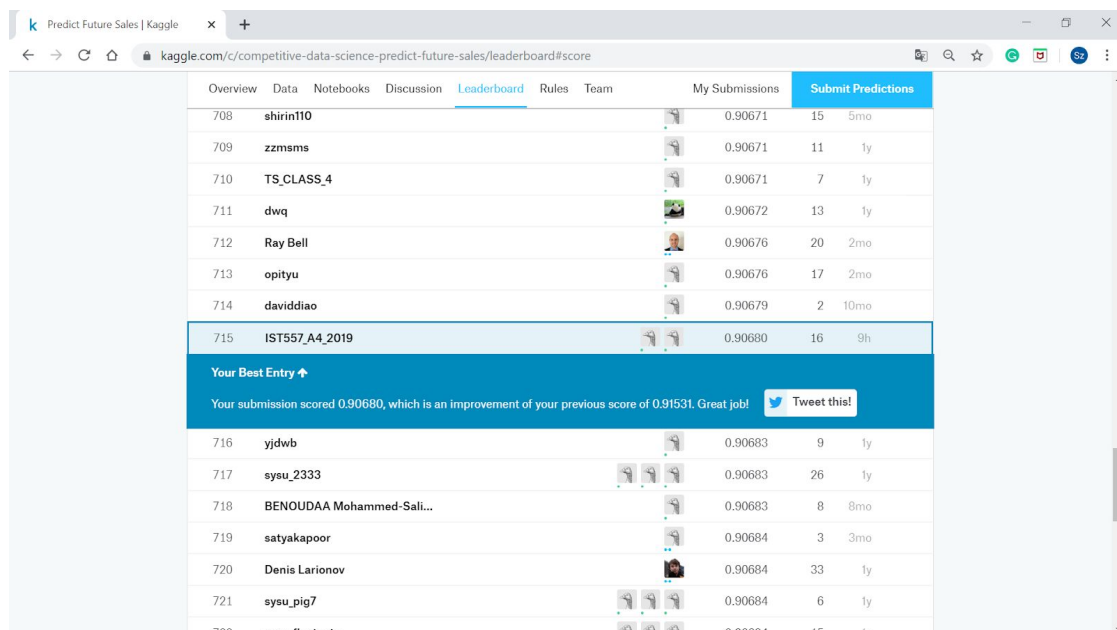
3.3 A table summarizing the results of all the methods.

Methods	RMSE (Root Mean Squared Error)
CNN	1.10015
LSTM	1.02138
Randomforest	0.91260
XGboost	0.90680

3.4 Performance discussion

The reason why XGboost and Randomforest are better than CNN, RNN models is because our datasets are tiny and neural networks can easily overfit, so you would have to turn hyperparameters with regularization to get comparable results to xgboost. If you have a huge dataset, said with billions (even trillions) of rows, a simple network and a few hidden layers can work very well without much tuning process.

3.5 A screenshot of the best method performance



Overview	Data	Notebooks	Discussion	Leaderboard	Rules	Team	My Submissions	Submit Predictions
708	shirin110						0.90671	15 5mo
709	zzmsms						0.90671	11 1y
710	TS_CLASS_4						0.90671	7 1y
711	dwq						0.90672	13 1y
712	Ray Bell						0.90676	20 2mo
713	opityu						0.90676	17 2mo
714	daviddiao						0.90679	2 10mo
715	IST557_A4_2019						0.90680	16 9h
Your Best Entry ↑								
Your submission scored 0.90680, which is an improvement of your previous score of 0.91531. Great job! Tweet this!								
716	yjdwb						0.90683	9 1y
717	sysu_2333						0.90683	26 1y
718	BENOUAA Mohammed-Sali...						0.90683	8 8mo
719	satyakapoor						0.90684	3 3mo
720	Denis Larionov						0.90684	33 1y
721	sysu_pig7						0.90684	6 1y
722	sysu_flinninn						0.90684	15 1y

4. Summary

4.1 Summarize what we learnt from technical side.

- Data preprocessing is important and it includes cleaning, instance selection, normalization, transformation, feature extraction and selection, etc. Data pre-processing may affect the way in which outcomes of the final data processing can be interpreted. So in this project, we did a lot of work in this part in order to get the better performance.
- For time series problem, feature engineering is important. We need to create the lag features to know the sale trend and those features can give us more information about the time series selling.
- XGBoost is the best model compared with LSTM, random forest and CNN. This is because our datasets are tiny and neural networks can easily overfit.
- Compare to CNN and RNN model, XGBoost and Randomforest take more time for training since we do lots of features engineering, so we should consider the priority of the features and try to make a balance between training time and performance.

4.2 Summarize what we learnt throughout the project

From this project, we learnt know how to separate the work for each team member and we also learnt from each other since both of us have different backgrounds. In addition, through the discussion, we can illustrate and change our ideas to double check the solution is correct and toward the right direction. And the most important thing is that we apply all the technical knowledge we learn from the class on this Kaggle competition. It is a best way to examine whether we really understand all these machine learning methods or not. After doing this project, we get more experience and builds a bridge to real world applications.

5. Team contribution

Ting-Yao Hsu: data preprocessing, model building and report writing

Szu-Chi Kuan: data preprocessing, model building and report writing