Final Project Proposal

Zhijie Ren, Weinan Fu

I. INTRODUCTION

This project is from website: https://www.kaggle.com/c/rossmann-store-sales. Rossmann operates over 3,000 drug stores in 7 European countries. Currently, Rossmann store managers are tasked with predicting their daily sales for up to six weeks in advance. Store sales are influenced by many factors, including promotions, competition, school and state holidays, seasonality, and locality. With thousands of individual managers predicting sales based on their unique circumstances, the accuracy of results can be quite varied.

Rossmann is challenging us to predict 6 weeks of daily sales for 1,115 stores located across Germany. Reliable sales forecasts enable store managers to create effective staff schedules that increase productivity and motivation.

II. PROBLEMS TATEMENT

The training dataset and testing dataset contains the following items: Store, Day of week, Data, Sales, Customers, Open, Promo, State Holiday, School Holiday. And the store information contains the following items: Store type, assortment, competition distance, competition Open Since month, competition Open Since year, Promotion2, Promotion2 Since Week, Promotion2 Since Year, promotion interval.

We will separate the dataset into two partitions, one as training data, the other as test data. Then we use all the dataset to build modules to predict the next six weeks of daily sales. Finally, we evaluate all the models ‘s result on the Root Mean Square Percentage Error (RMSPE).

III. RELATED WORK

I found this problem has already been solved, but the evaluation scores are not good enough. Some solutions just train some models and compare the predictions. We focused on training the model and then correcting the model in different method to get a high evaluation score. The details will be expressed as follows.

IV. DATASET

A. Pre-processing

We analyzed the dataset and found it has some missing data. The missing dataset of store information are all from the store which id is 622, and it already has no holiday for six weeks. So we suppose that this store is still open as usual. Then we check why the Promo2 data miss. For some reason, maybe this store didn’t join the promotion activity. Through the experience, we found that if we replace ‘0’ to full the missing data, it will have a better performance.

B. Data analysis

The first step is to analyze the data graph of the store sales from June to September. From this graph, we can find the sales is periodic, November and December have the peak of sales during one year. Moreover, the sales in 2014 June and July is similar with Aug and Sep. So we can add the 2015 June and Sep as hold-out dataset to verify and optimize the module.

C. Feature create

Transform the String type to Int. Merge the year, month, day as one feature named Week of Year. Add two new feature named Competiton Open and PromoOpen to calculate the time of the competitors’ open time and promotion time.

V. APPROACH

We train a XGBoost model as an initial model and left three stores’ information as validation data. We find that it already has a good performance to predict the hold-out dataset. But it is still a little higher than the true value. Then we use the weight correction and find a good value for weight to get a high score. Next step is to correct the details. We split the different stores into groups to check the details that is to provide every stores have a best RMSPE score. Now, we get the model after correction and train the two models to merge as the final model.

VI. EXPERIMENTS

A. Missing data processing

In the experiments, we found that it doesn’t have a good performance if we replace the missing store information as the medium value of the other competitors’ information. So we just replace them as 0.

B. Merge feature

Due to the dataset about the store information has some data redundancy, we need to merge these data.

VII. DISCUSSION

When we merge the two models before and after correction, we have a couple of methods to merge. The first solution is to use the average value, but maybe it is not convinced. So we can analyze the feature and set weights for the two models. We also need to pick some data which haven’t a good performance in the initial models and compare with the final models’ prediction. Now the problem to be solved is how to build the algorithm’s initial model without using packets. It is hard to train the model with large training data in a reasonable limited time.