ensiie
école nationale supérieure d'informatique
pour l'industrie et l'entreprise

Mars - Avril

Rapport du MOST

CHEN ZEYU
CHEN GUANGYUE
LI ZIHENG

Sommaire

1 Abstract

In this project, we applied machine learning techniques to a real world problem of predicting stores sales.This kind of prediction enables store managers to create effective staff schedules that increase productivity and motivation. We used popular open source statistical programming language R. We used feature selection, model selection to improve our prediction result. In view of nature of our problem, Root Mean Square Error (RMSE) is used to measure the prediction accuracy

2 Introduction

Rossmann is a chain drug store that operates in 7 European countries. We obtained Rossmann 1115 Germany stores' sales data from Kaggle.com. The goal of this project is to have reliable sales prediction for each store for up to six weeks in advance. The topic is chosen, because the problem is intuitive to understand. We have a well understanding of the problem from our daily life, which makes us more focused on training methodology. The input to our algorithm includes many factors impacting sales, such as store type, date, promotion etc. The result is to predict 1115 stores' daily sale numbers.

3 Data Analysis

Training data is comprised of two parts. One part is historical daily sales data of each store from 01/01/2013 to 07/31/2015. This part of data has about 1 million entries. Data included multiple features that could impact sales. Table 1 describes all the fields in this training data.

Field Name	Description
Store	a unique Id for each store: integer number
DayofWeek	the date in a week: 1-7
Date	in format YYYY-MM-DD
Sales	the turnover for any given day: integer number (This is what to be predict)
Customers*	the number of customers on a given day: integer number (this is not a feature. Based on the test data from Kaggle, this feature is not included in test data)
Open	an indicator for whether the store was open: 0 = closed, 1 = open
Promo	indicates whether a store is running a promo on that day: 0 = no promo, 1 = promo
StateHoliday	indicates a state holiday. Normally all stores, with few exceptions, are closed on state holidays. Note that all schools are closed on public holidays and weekends. a = public holiday, b = Easter holiday, c = Christmas, 0 = None
SchoolHoliday	indicates if the (Store, Date) was affected by the closure of public schools: 1 = school holiday, 0 = not school holiday

Figure 1: Historical sales data table features

The second part of training data is supplement store information. It has 1115 store info entries, which listed the store type, competitor and a different kind promotion info. Table 2 below describes all the field in this file.

Field Name	**Description**
Store	a unique Id for each store: integer number
StoreType	differentiates between 4 different store models: a, b, c, d
Assortment	describes an assortment level: a = basic, b = extra, c = extended
CompetitionDistance	distance in meters to the nearest competitor store
CompetitionOpenSinceMonth	gives the approximate year and month of the time the nearest competitor was opened
CompetitionOpenSinceYear	
Promo2	Promo2 is a continuing and consecutive promotion for some stores: 0 = store is not participating, 1 = store is participating
Promo2SinceWeek	describes the year and calendar week when the store started participating in Promo2
Promo2SinceYear	
Promointerval	describes the consecutive intervals Promo2 is started, naming the months the promotion is started anew. E.g. "Feb,May,Aug,Nov" means each round starts in February, May, August, November of any given year for that store

Figure 2: Store Information data table features

Here the distribution of Sales

Figure 3: hist of sales

We have previously speculated that the store id has a relationship with sales. It is possible that the smaller the store id is, the it was built early, so it is more likely to be in the city center, so the sales are higher.But from the graph we can see that sales are not related to store id.

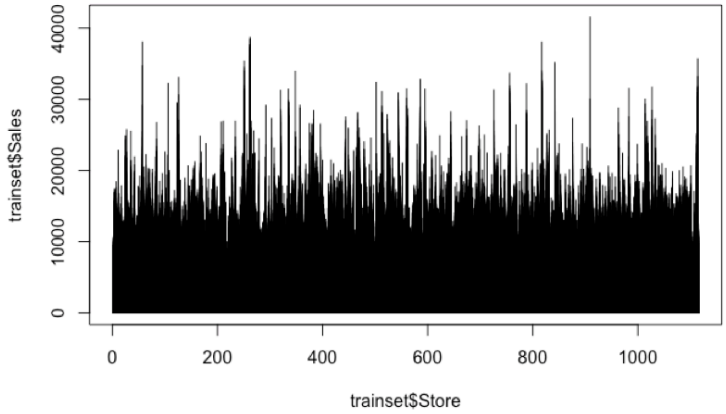


Figure 4: relationship between store id and sales

We can see that most stores are closed on Sundays, while sales are more higher on Monday.

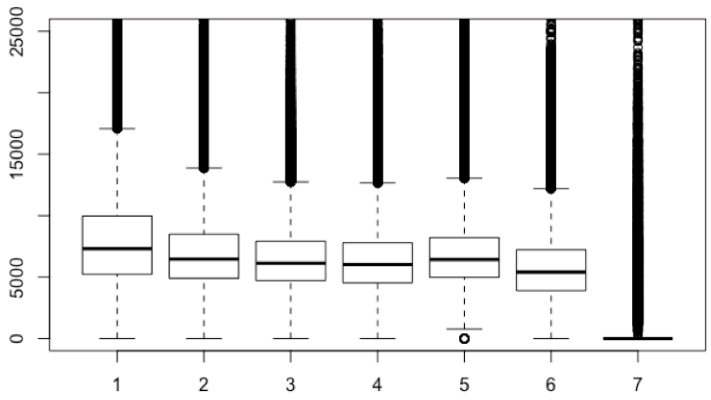


Figure 5: relationship between day of week and sales

It can be seen that each year from October to December is a period of rapid growth, December to the peak, and from January to the bottom. And the overall fluctuations are also obvious.

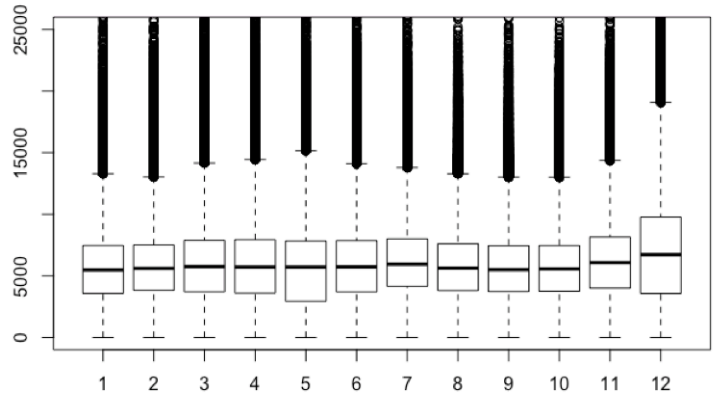


Figure 6: relationship between month and sales

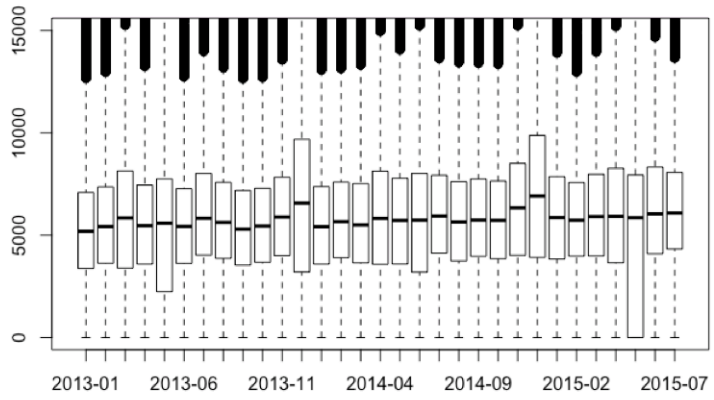


Figure 7: relationship between year and month and sales

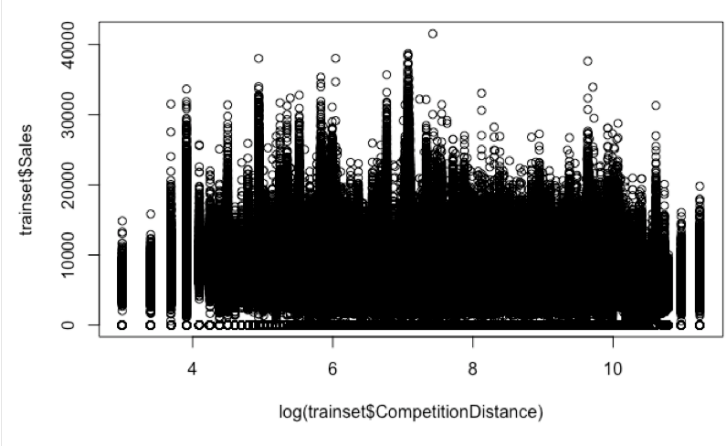


Figure 8: relationship between competition distance and sales

It can be seen that in a certain distance range, many stores will get higher sales than other stores.

The same day sales are obviously better than non-promotional sales, but the difference is not very big.

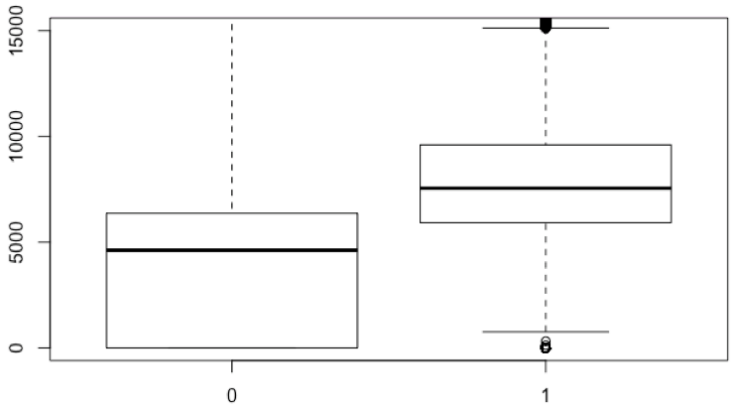


Figure 9: relationship of having promo or not

The first state holiday has more effect on sales than the others.

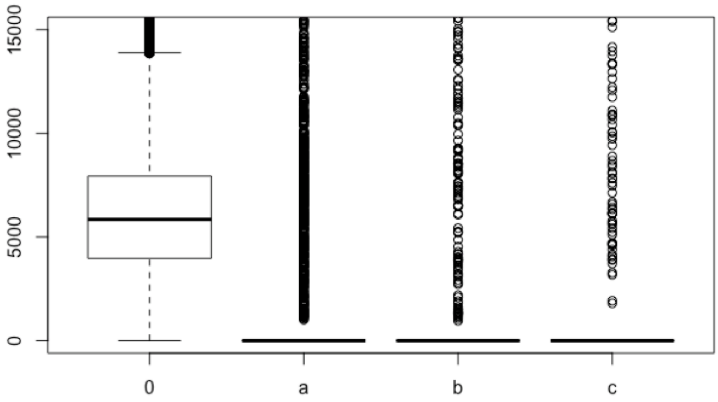


Figure 10: relationship of having stateholida or not

The school holiday has little effect on sales.

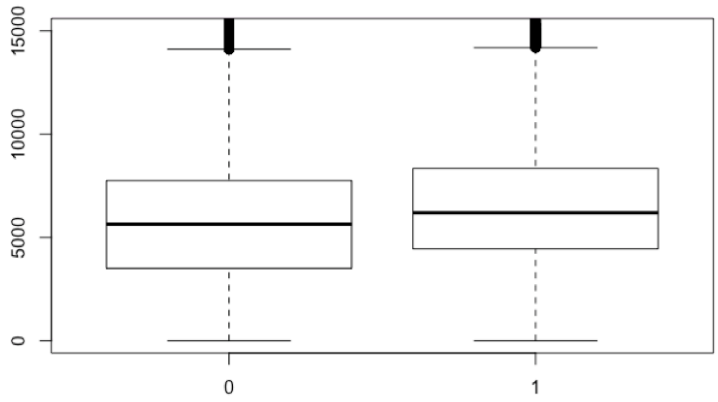


Figure 11: relationship of having schoolholida or not

For StoreType, it can be seen that b type sales are higher than others, and there is basically no difference between a, c, and d types.

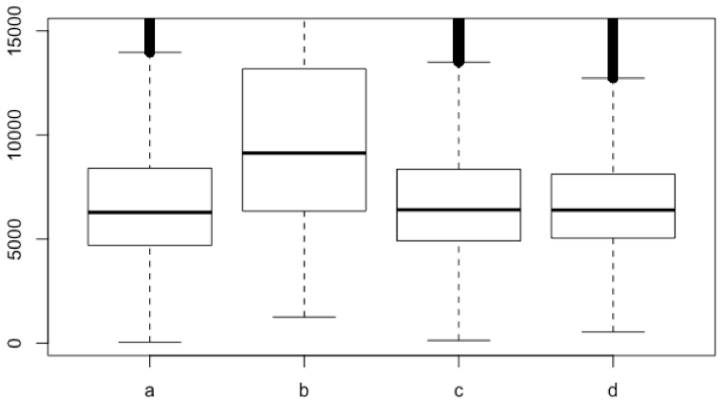


Figure 12: relationship of storetype

Sales without a promotion type 2 are overall better.So it can be seen that the promotion2 has little effect on the customer.

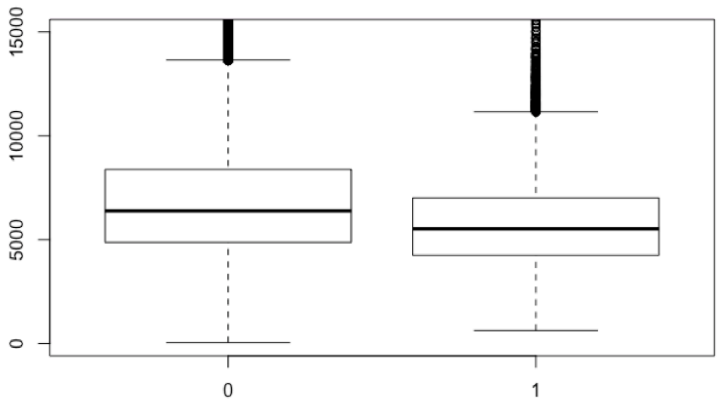


Figure 13: relationship of having promo2 or not

4 Data processing

4.1 CompetitionDistance

CompetitionDistance is a feature of store table , when we merge train store and store table , this feature will give us the wrong information . For exemple , the CompetitionDistance of store 1 is 1000, CompetitionOpenSinceYear and CompetitionOpenSinceMonth tell us the moment where his competator opened. So, we cannot say that CompetitionDistance is also 1000 in the day before this moment , beacause this store even dont have a competator before this moment.

So,we compared the date and competitor open date. For any date and any store which doesn't has competitor in this date, we assign CompetitionDistance as a large number 100000.

4.2 Promo2

Similarly, we treated Promo2 for the same reason, but it is more complicated .Because we also take PromoInterval into account. So, we combined Promo2, Promo2SinceWeek, Promo2SinceYear and Promointerval to a promotion 2 indicator in historical sales data. The indicator indicates on a certain day whether a certain store is on promotion 2.

4.3 Store and Sales

We treated Store as a factor and we did a log transform for sales in oder to not be so sensitive to high sales.

5 Methodology

5.1 Linear Model

For this problem, the linear models have bad performances, but there are also a way to build a linear model. Here we tried 'linear regression' , 'Ridge Regression' and ' Lasso Regression'. To make some features working such as 'DayOfWeek', 'StoreType' and 'Assortment', we use the ont hot function from package 'mltools'. As the results, all of these three regression have a test error feedbacked from kaggle.com which is arroud 0.4.

5.2 SVM

For the SVM method, we choose the package "e1071" to train our model, so our first step is choosing the regression type and the kernel for this regression.

5.2.1 Kernels and Type selection:

	linear	polynomial	radial	sigmoid
eps-regression	0.17834	0.15844	0.13579	0.35867
nu-regression	0.18955	0.16936	0.14461	0.46985

FigureX: Train Error with different kernels and types used in SVM regression

We can see that with the Kernel 'radial' and the type 'eps-regression', the model permance better.

5.2.2 Parameter Choosing:

Here we use the function 'tune.svm' from the package "e1071" to compare the different gamma and cost. This function use K-ford cross validation to choose the parameter. The gamma parameter defines how far the influence of a single training example reaches. The cost parameter rules the error of the cutting plane. With higher cost the train error will be lower but the test error may growth, and the parameter gamma is for the Kernel, it's also sensitive for our model.

```
Parameter tuning of 'svm':

- sampling method: 10-fold cross validation

- best parameters:
 gamma  cost
  0.05    3
```

FigureX: Best Parameter

5.2.3 Result:

Because of the running time of the svm model is long, we didn't pay too much attention into this model. Our result error is .

Submission and Description	Private Score	Public Score
SVM.csv just now by guangyue add submission details	0.13841	0.12382

FigureX: Result

5.3 Random Forest and H2o

5.3.1 Random Forest:

Random Forest is a flexible, 'easy to use' machine learning algorithm, it porduce a good result most of the time. It is also one of the most used algorithms, because it's simplicity and the fact that it can be used for both classification and regression tasks.

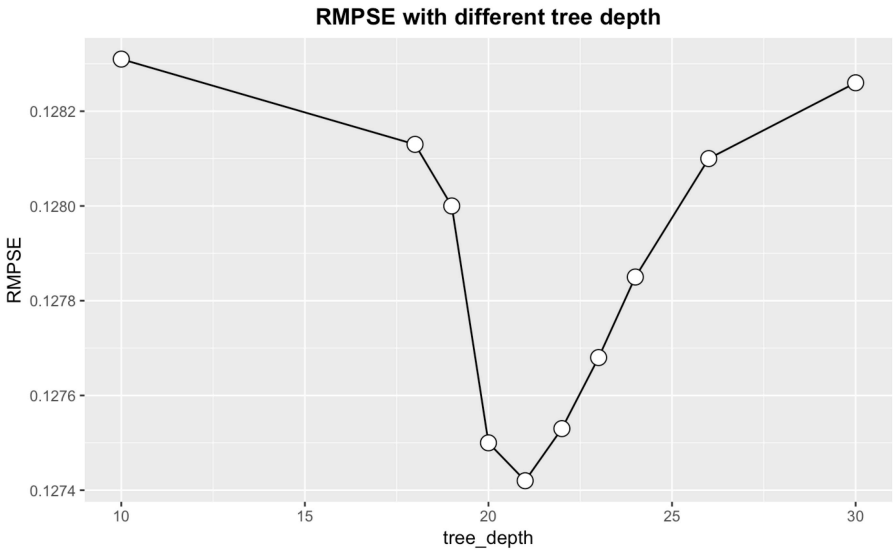
When given a set of data, Random Forest generates a forest of classification or regression trees, rather than a single classification or regression tree. Each of these trees is a weak learner built on a subset of rows and columns. It chose the features and the subset of the data(for training a tree) randomly. More trees will reduce the variance. So it could handle well the overfitting issues. For our problem, it has also a good performance.

5.3.2 H2o Random Forest:

'H2o' package use Distributed Random Forest, which is a powerful classification and regression tool. For this package, 'h2o.randomforest' run faster than the normal one in R, it can also limit the tree depth, (R's randomForest builds really deep trees), allowing for having a better predictions.

5.3.3 Parameter Choosing:

So here we should choose the parameter 'max_depth', here we use Cross-Validation to compare the test error.



FigureX: RMPSE with different tree depth

We can see that for the test error, the models with depths 20 and 21 have the best porfermences. So we decided to build two models with the depth which are 21 and 20. Then we build a forest with a big quantity of trees, Which is 100.

5.3.4 Feature Selection:

For h2o random forest, we should load the data into h2o cluster. After our several test, we find that some features make the models perform worse. With the summary of our model, we decide to remove some features which have a low importance to our model. After we remove two features 'SchoolHoliday' and 'StateHoliday', our random forest model perform better.

5.3.5 Result:

Submission and Description	Private Score	Public Score
h2o_rf.csv 27 minutes ago by zeyu Max_depth=20	0.12742	0.11466
h2o_rf.csv an hour ago by zeyu Max_depth=21	0.12750	0.11449

FigureX: The Result Of Our Models

We run our best models on kaggle.com provided test data. The test errors feed-backed from kaggle.com are 0.11449 and 0.11466. So we can say that the forest with depth maximum 21 is better. And this result is already in the top 150 on kaggle.com.

6 Conclusion

According our results, Random Forest has the lowest test error feedbacked from kaggle.com. But we believe that the SVM model could perform as well as RF although it cost so much time for learning once. So for our future work, we will do more anlyses and tries on SVM.

After this project, we realize that the feature treatment has a large impact on training model quality. A correct feature selection could helps us to develop simpler and faster models. Once features are chosen and formatted correctly, the prediction error improved dramatic. Data preprocessing is the same, because, the representation and quality of data is first and foremost before running an analysis.

What's more, after this project, we have an unforgetable experience of data analysis. We know clearly the steps to treat the data and train the machine learing models, and have a clearer understanding of models such as Random Forest and Support Vector Machine.

7 Reference

1. Data source:https://www.kaggle.com/c/rossmann-store-sales/data
2. Distributed Random Forest Introduction:http://docs.h2o.ai/h2o
3. Data preprocessing and feature selection:(en chinois)https://blog.csdn.net
4. Support Vector Machine: http://uc-r.github.io/svm