Rossmann Store Sales Prediction

Fazal Hyder, Rishu Garg(summer interns at REGex software services)

OVERVIEW

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. Using library of python RandomForestRegressor

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. Generalized linear model (GLM) and Supporting vector machine (SVM) regression, RandomForestRegressor ,were used to train model and predict sales.

DATASET AND FEATURES

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

Table 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

Table 2: Store Information data table features

We did several things to combine features and create features directly related to sales number. The work we did is:

1. The supplement store information can't be used directly. We merged store information and historical sales data. Story type and Assortment is merged into each entry of historical sales data
2. Combine 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.
3. Similarly, we combined CompetitionDistance, CompetitionOpenSinceMonth,

CompetitionOpenSinceYear to a competitor indicator. The indicator indicates on a certain day whether a certain store has competitor.

4. Since CompetitionDistance is provided, we used CompetitionDistance to train model, instead of competitior indicator. For any date and any store which doesn't has competitor (competitor==1), we assign CompetitionDistance as a large number 100000. This method enable us to use only one

   CompetitionDistance feature. It also models the no competitor case by weakening CompetitionDistance impact.

5. Historical sales dataset has Date feature. We created Month and Year feature based on Date feature. Month and Year is used as feature, since they correlated with sales data.

The final training/test dataset used includes the following features.

- StoreID Open Promo2 indicator
- DayOfWeek StateHoliday StoreType
- Month Year SchoolHoliday Assortment
- Promo CompetitionDistance

METHODOLOGY

A random forest is a meta estimator that fits a number of classifying decision trees on various subsamples of the dataset and uses averaging to improve the predictive accuracy and control over-fitting. The sub-sample size is controlled with the `max_samples` parameter if `bootstrap=True` (default), otherwise the whole dataset is used to build each tree. are relatively

Feature Selection

The input features have many category features. Category feature should be treated as factor in training.

Too many factors increase runtime a lot. We tried to combine similar category features as much as possible. We looked at feature distribution vs sales and combine features with similar distribution. Two examples are listed below.

DayofWeek feature:

Removing store close data and plotting Mon through Sun's sales data distribution. From the plot, we can tell that Tue through Fri's sales distributions are very close. Mon, Sat and Sun's sales distributions are unique. In database, DayofWeek is represented as numeric number 1-7. From intuitive, we know that there is no linear relationship from 1-7 number to sales data. We treat DayofWeek as four factors, Mon, Weekday(Tue-Fri), Sat, Sun.

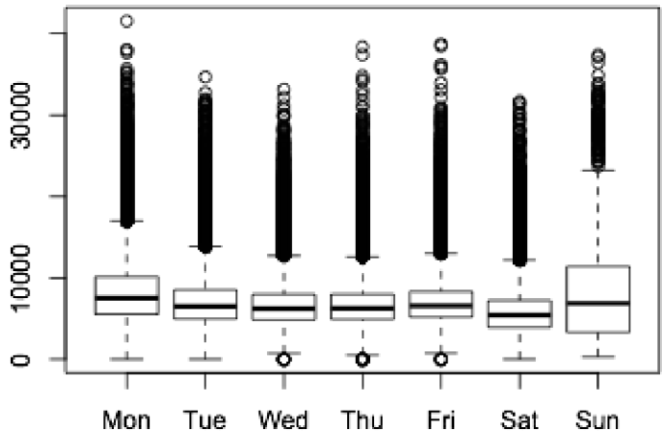

Figure 1: Sales distribution of each weekday

Holiday Features:

StateHoliday, SchoolHoliday and Open are highly correlated features. They don't meet IID assumption. Removing store close data and plotting state holiday sales distribution. State holiday a, b, c's sales distribution is not similar. However, state holiday == b only has 145 data points. state holiday == c only has 71 data points. Since the training data points are not large, we combined state holiday b and c as one category.

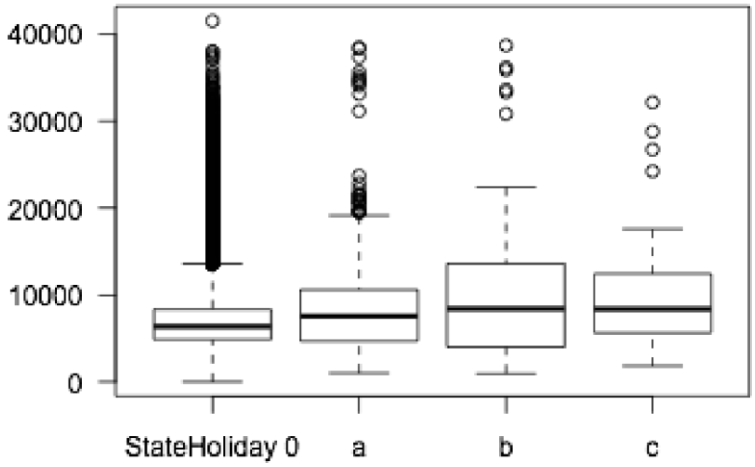

Figure 2: Sales distribution of each state holiday

Result evaluation metrics:

RSME (root mean square error) method is used to evaluate the prediction quality. We took the percentage error of predicted sale data to real sale data and then calculated the standard deviation. Equation is shown below.

$$\epsilon = \sqrt{\frac{1}{m}\sum_{i=1}^{m}\left(\frac{|PredSales^{(i)} - Sales^{(i)}|}{Sales^{(i)}}\right)^2}$$

RESULT AND ANALYSIS RANDOMFOREST REGRESSOR

Feature exploration:

We started with taking the features as is. Using “DayOfWeek”, “Promo”, “SchoolHoliday”, “StateHoliday” as feature vector feed into GLM regression. Realizing that in Christmas season, the sales are much higher than what the model predicted, we added “WeekOfYear” feature to the model. By treating these as factors instead of numeric numbers, we can achieve better training and testing error. We also noticed that although most of the stores close on state holidays, the sales near state holidays are higher than normal. Therefore, we added flag on whether a date is before or after a holiday. With that, we lower the testing error further by 0.009.

	Actual_Values	Predictions
157731	5144	5472.74
883585	8415	9870.62
482186	3358	3982.17
631484	5740	4858.36
972304	0	0.00
424017	6170	6401.51
366997	5399	4787.54
644217	0	0.00
278369	5556	4731.19
835741	4940	4876.67
909731	0	0.00
267628	15833	15060.89
492858	12729	11053.75
404831	9349	8766.93
298455	0	0.00
796096	2796	3021.60
320860	7194	8194.16
605453	0	0.00
754134	2276	1975.56
275489	5727	5485.02

FUTURE WORK

We believe the sales number of a particular day is also related to the sales number before that day. Adding time series to the model can improve accuracy [4]. We will try adding time series to the feature vector to see what we can achieve. Different machine learning algorithm such as GLM or SVM REGRESSIONSN can also be interesting to explore. If any viewer of this report has a data on their store and wants to predict their sales

Contact at mail : hshaik172@gmail.com

CONCLUSION

We've used various algorithms to test,train and predict the sales only **`RandomForestRegressor gave whopping 96% accuracy for predictions`**

REFERENCE

1. Data source: https://www.kaggle.com/c/rossmann-store-sales/data

2. python-Programming site: https://www.python.org/

3. REGex software services : https://www.regexsoftware.com/

4. http://scikit-learn.org/stable/auto_examples/svm/plot_rbf_parameters.html

5.RandomForestRegressor https://scikit-learn.org/stable/modules/generated/sklearn.ensemble.RandomForestRegressor.html