Exploratory Analysis: Rossmann Store Sales

Women in Tech

Gana Ramesan

Manasi Prabhune

Sanyogeeta Lawande

Shijna Surendran

ITCS 6162 Knowledge Discovery in Databases

University of North Carolina Charlotte

5/1/2018

Table of Contents

Project Description

The aim is to apply prediction model to the sales for every Rossmann store for the future months. This prediction will help the stores to manage the investment, the workforce and organize supply chain management.

In our project, we chose Rossmann store data to predict their sales. The data collected is the sales data for 1,115 Rossmann stores, including holiday, promotion, competitors etc. This research is required to predict the 6 weeks of sales for these stores.

We choose different techniques to help find the precise forecast result. This will also give a large amount of information about the data and their correlation between the number of sales in each store per day.

Getting Started

Prerequisites

Latest version of working R Studio Software

Rossmann Store Sales dataset available in Kaggle - https://www.kaggle.com/anshumanyp/rossman/data

Data Sets

- STORE.CSV - information about each store
- TRAIN.CSV – training data for sales
- TEST.CSV - test data for sales

Data Fields

- **Id** - represents a (Store, Date) duple within the test set
- **Store** - Unique Id for each store
- **Sales** - The turnover for any given day (variable to be predicted)
- **Customers** - The number of customers on a given day

- **Open** - An indicator for whether the store was open: 0 = closed, 1 = open
- **StateHoliday** - Indicates a state holiday
    - Normally all stores, with few exceptions, are closed on state holidays.
    - 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
- **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
- **CompetitionOpenSince** [Month/Year] - Approximate year and month of the time the nearest competitor was opened
- **Promo** - Indicates whether a store is running a promo on that day
- **Promo2** - Continuing and consecutive promotion for some stores:
    - 0 = store is not participating
    - 1 = store is participating
- **Promo2Since** [Year/Week] - Describes the year and calendar week when the store started participating in Promo2
- **PromoInterval** - Describes the consecutive intervals Promo2 is started, naming the months the promotion is started anew.
    - “Feb,May,Aug,Nov” means each round starts in February, May, August, November of any given year for that store

Analysis

Data Cleaning

1. Remove customers variable, since it is not an input variable of the test set. Also predicting the number of customers is not the goal.
2. Remove observations of closed stores in sales, as no sales can occur when a store is closed.

3. Separate date column into year, month, week and day. Since we need to compare these attributes with each other, we need to convert them into same format(integer).

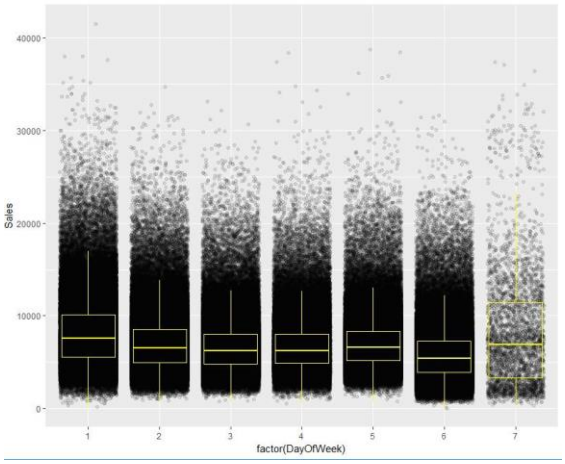
    - Year
    - Month
    - Week
    - Day

4. Save all store IDs in a separate variable ‘stores_to_test’ to fit into prediction models.
5. In the test dataset, the variable Open, which determines if the store was open or closed, has missing data. we see that only for store 622 there are missing values in Open variable. We interpret these missing values as TRUE i.e. the store was open. Also considering Sundays the stores must be closed, they are marked as FALSE.

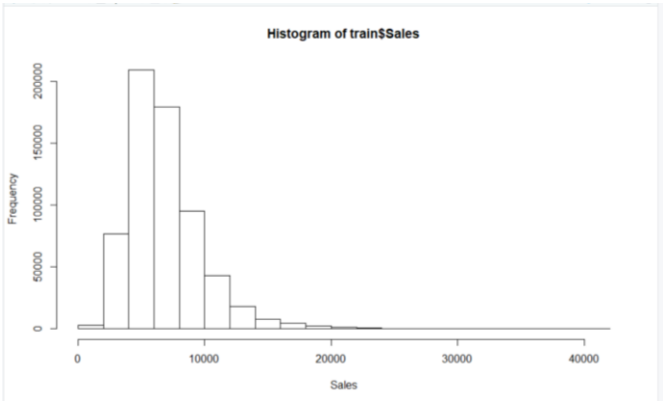
Exploratory Data Analysis

Exploratory data analysis was performed in R to find correlation between different variables in the dataset.

1. From the correlation plot, we observed that from Monday to Saturday, there are significant number of sales. The sales decreases significantly on Sundays.



2. By plotting the histogram on the sales variable of the train data set, we saw the sales information is not normally distributed. The data is right skewed.



10-Fold Cross Validation

To generalize our predictions to limit problem of overfitting on the training set, we decided to randomly partition the original training observations into 10 equal sized subsamples. There are about 800 days of data for every store, this gives each subsample 80 records. Therefore, the training set comprises of 9 equal sized folds or 720 records, and the rest the testing set.

Error Metric Used

We are using Root Mean Squared error (RMSE) to compute our validation error. **RMSE** or Root Mean Squared Error is the average deviation of the predictions from the observations. In terms of output variable units, it tells us how well (or not) an algorithm is doing.

In all the prediction models, we are trying to find out the average sale of each store. Since we are not considering the factors like was it a holiday, was there a promotion on a particular day, etc. to predict the daily sales, the RMSE value will be on a higher scale.

Prediction Models

Baseline Prediction

We can say that, the sales are zero when the store is closed. This implies that the mean of sales can be calculated by using only the sales data when the store was open/not closed. Therefore, we have removed the closed days from the dataset to perform the baseline prediction.
By the below formula, we can calculate the root mean square error. We first consider the sales data for store #1.

$$RMSE = \sqrt{\frac{\sum_{i=1}^{N} (Predicted_i - Actual_i)^2}{N}}$$

We also calculate the RMSE by calculating the median of the sales data for store #1. From this, we see that, the RMSE with mean is better than the RMSE with the median. Hence, we calculate the RMSE with the mean for the rest of the data.

```
##       [,1]     [,2]
##  [1,]  234 3504.596
##  [2,]  251 3545.364
##  [3,]  262 4666.399
##  [4,]  335 4759.825
##  [5,]  586 3537.216
##  [6,]  756 3910.047
##  [7,]  831 3615.355
##  [8,]  842 3785.939
##  [9,] 1014 4568.576
## [10,] 1027 3751.706
```

From the output received, we can say that the stores with higher sales had a high mean square error. This is true in an ideal case and hence, the data is correct.

Linear Regression

Linear regression is method for modeling the relationship between dependent variable Y (in our case, “Sales”) and independent variables X (X1, X2, X3...). To apply linear regression on the data, we use lm function.

lm is an implementation of multivariate linear regression. If we supply Y ~ X as the input, lm calculates and returns a model that can be used to predict Y using X.
The generic equation of linear regression is shown below.

$$Y = \beta_0 + \beta_1 X_1 + \beta_2 X_2 + \ldots + \beta_n X_n + \varepsilon$$

We first built linear model with all available features (“DayOfWeek”, “Open”, “Promo”, “StateHoliday”, “SchoolHoliday”, “StoreType”, “Assortment”, “CompetitionOpenSince”, “Promo2”, “Promo2Since”, “Sales”,”, “CompetitionDistance”, “day”, “month”, “year”) using the training set. Then, we applied the linear model on training dataset.

As discussed in the previous section, we use the 10-fold cross validation. The dataset is divided into train and test dataset using 10-fold method. Considering one store at a time, there are dataset for 800 days for each store. Which gives us fold of 80 data each. Therefore, the training dataset will contain 800-80-720 data records.

```
## [1] "RMSE:  601.900441744124"
```

The RMSE value we received for linear regression is 601.90 and the RMSE for the baseline prediction model was 1011.458. We see that there is a difference of approximately 500 errors. Hence, we can say that linear regression model is much better than the baseline prediction model.

Random Forest

Random forest is a collective method that uses training set to build a decision tree. To avoid overfitting, Random forest uses aggregation of many decision trees, since it uses an ensemble of regression trees over different data.

We have divided the training sets by store and models are created separately for each store, since the store variable is the strongest predictor for the sales. Like linear regression, we have used the 10-fold cross validation for Random Forest as well. The dataset is divided into train and test dataset using 10-fold method. Using the original values for the date components, we got results with 200 as the tree number. While there was only little improvement but large execution time on increasing the value to 500, We continued with the 200 tree number.

We have used the same cross-validation process used in linear regression, storelm with different models, used for both numeric and binary versions of the data set.

```
## [1] "RMSE:  568.4792"
```

Results using Random Forest for stores are clearly better than those that were attained using linear models.

Results

Below table shows the final results of the different algorithms applied on the dataset and their results:

Algorithms Applied	Results (Root Mean Square Error)
Baseline Prediction	1872.81
Linear Regression	1253.274
Random Forest	864.832

Conclusion

We used 3 different algorithms on the Rossmann Stores Sales dataset to predict the sale of the store.

Based on the results that we achieved after applying these algorithms we can conclude that Random Forest gave the best result with the least root mean squared error of 864.832. Also, the result is obtained in less time being few seconds.

Next model that works good in prediction is Linear regression for overall dataset but, when applied on individual store, e.g. store 262 the error rate was high, yet better that Baseline algorithm.