Rossmann Store Sales

Aravind Reddy Keesara (800976233)
Harshavardhan Ganti (800989176)
Chandrakanth Yela (800987193)

ITCS 6162 Knowledge Discovery in Databases

University of North Carolina Charlotte

ABSTRACT

In this project, we are given a real-world problem of predicting the sales of a particular store market named Rossmann based on the historical data of that store. We tried to apply the machine learning techniques that we have learned during the course work to get the job done.
This Predictive analysis will help the store manager to create effective staff schedules that increase productivity and motivation.

We have used feature selection and model selection to improve prediction result. We have considered two models, one being logistic regression and other being RandomForest model. After performing Root Mean Square Error (RSME), we found out that for effective prediction RandomForest model appears to be more accurate.

INTRODUCTION

Rossmann operates over 3,000 drug stores in 7 European countries. we are given the task to predict their daily sales for up to six weeks in advance for 1,115 stores located across Germany.

Approach:

1. In order to check the accuracy of a model we need to know the percent error between the known value and predicted value. But the test data has no sales attribute values, we have to predict them. Hence, we have divided the train.csv file into two separate datasets of train_pt1 (contains 70% of the data) and test_pt1 (contains 30% of the data).
2. Before proceeding to build the model, we have applied various imputation methods to process the data.
3. We have used the train_pt1 dataset to train the logistic regression and RandomForest models.
4. As we had derived the test data from train data itself as explained in point (1), we have removed the values of sales attributes from test_pt1 dataset. Then the prediction models are applied to the test_pt1 dataset to fill the sales attributes.
5. The predicted sales attribute values are then compared with the original sales attribute values and then checked for accuracy using Root Mean Square Error (RSME) approach.
6. We have found that RandomForest model that we have built is more accurate for the given dataset. Hence, we have used that model to predict the sales attribute values in the original test.csv dataset file.

DATASET & DESCRIPTION

The dataset has three files, namely train, test and store. The train data has sales data of each store from January 1st, 2013 to July 31st, 2015 with about one million rows. The store data file has the supplement information about each of the stores. The table below summarizes the type and purpose of each filed present in train and store data files.

Data Source: https://www.kaggle.com/c/rossmann-store-sales

Attribute	Description
Store	It is a unique Id for each store: integer number
DayofWeek	it indicates the date in a week: 1-7
Date	Provided in the format (YYYY-MM-DD)
Sales	turnover for any given day: integer number (This is what to be predict)
Customers*	Number of customers on a given day: integer number.

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
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.

DATA PREPROCESSING

Below are the steps followed for preprocessing the data:

1. The store data file cannot be used directly, hence we merge the store information data with training data.
2. We have checked the obtained train & test data from point (2) for the null values and applied missing value imputation technique by replacing those with zeros. The columns that we have found having null values are:
    a. Train-data: CompetitionDistance (2642), CompetitionOpenSinceMonth (323348), CompetitionOpenSinceYear (323348), Promo2SinceWeek (508031),
    b. Test-data: CompetitionDistance (96), CompetitionOpenSinceMonth (15216), CompetitionOpenSinceYear (15216), Promo2SinceWeek (17232), open (11).
3. Date attribute is of type factor, we have converted that to date format, then derived month, year, day and created separate columns for those. We are deriving month, year, day from the date attribute as each of it has its own contribution to the model.
4. The percent of stores that are open is about 83.01. Hence, we will consider the stores that are open to build the final data.
5. From the finalized data, there are three attributes StoreType state holiday and assortment, which are in the form of categorical value, so for any model to understand the data the values have to be in numerical values, so these categorical values are needed to be transformed as numerical values.
6. As mentioned in the "Approach" section, we have divided training data into train_part1 and test_part1 in the ratio (70:30) percent. For simplicity, in the further points train_part1 will be called train data and test_part1 will be considered as test data.

FEATURE SELECTION

We have explored the following details from the dataset provided.

1. The percentage of stores that are open is 83.01, closed is 16.98
2. The stores with promos available are about 38.15 percent and the ones with no promos are about 61.8 percent
3. State holidays are of three types, public holiday (1.9%), Easter holiday (0.6%), Christmas (0.4%). Which implies 96.9 percent of the days are contributing toward working days
4. School holidays are about (17.8%), which implies that 82.2 % are working days for schools.
5. StoreType feature has four models, a (54.2%), b (1.5%), c (13.4%), d (30.7%). Since these four models have considerable weightage among the total stores, this feature may be useful.
6. Assortment feature is of three types: basic (52.8%), extra (0.8%), extended (46.3%).

7. DayofTheWeek Feature :
   From the plot, we can see that the sales distributions are very much alike from Monday to Friday. But the distribution varies on Saturday and Sunday. The sales on weekends are very less compared to other days.

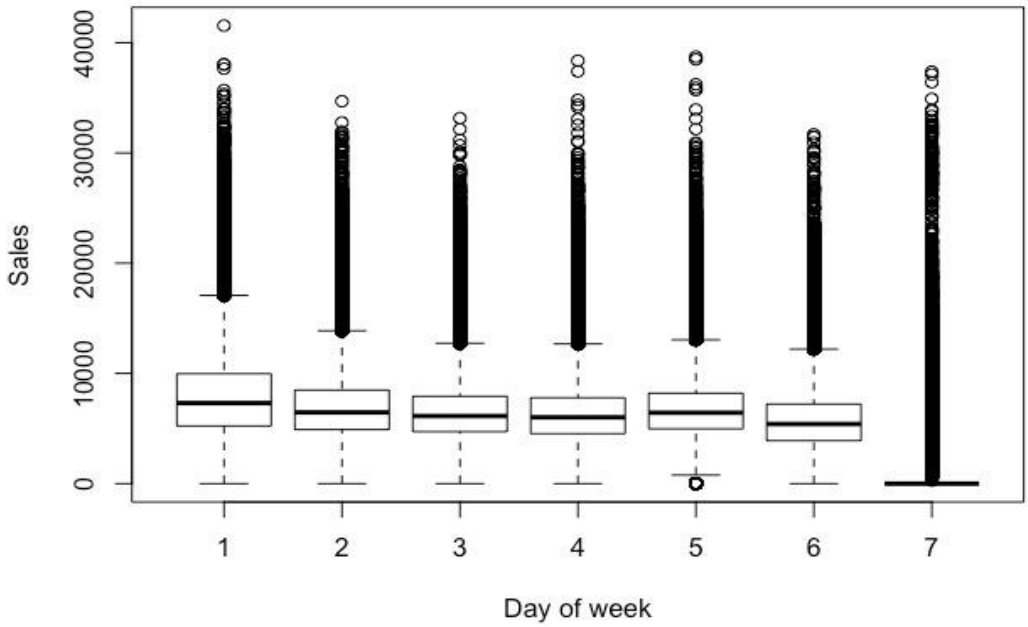

Figure 1. Boxplot for sales vs day of week

8. Holiday Feature:
   Plot shows that sales are very low where there is a StateHoliday compared with no StateHoliday. This clearly indicates that holidays affect sales.

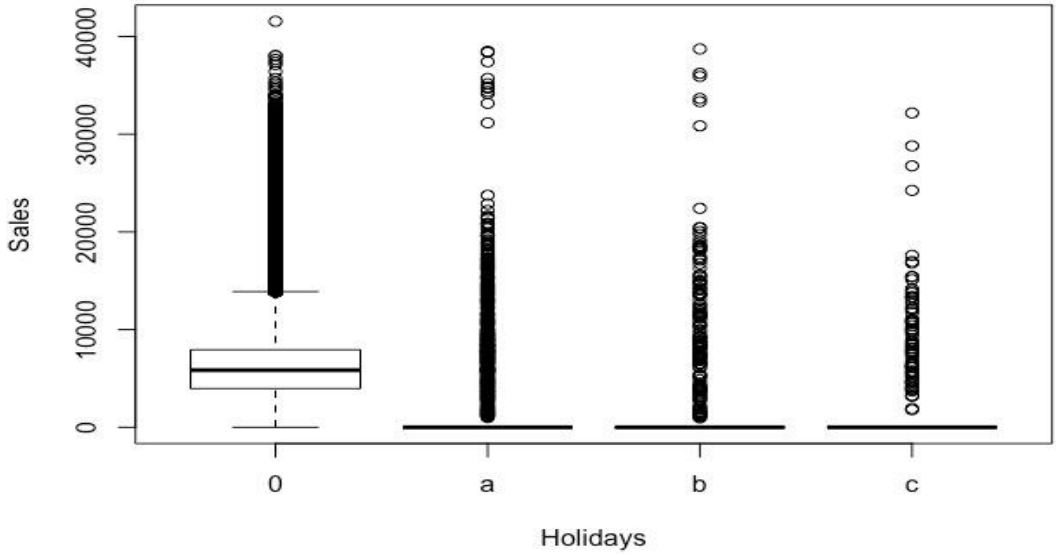

Figure 2. Boxplot for sales vs State Holidays

Features used for building the model are:

- Sales
- SchoolHoliday
- CompetitionDistance
- Promo2SinceWeek
- Year
- Promo
- StoreType
- CompetitionOpenSinceMonth
- Promo2SinceYear
- Day
- StateHoliday
- Assortment
- Promo2
- Month

MODEL SELECTION

1. **Linear Regression Model**:

When we have the train and test data ready, we can use them to create Linear regression model. Train data (Input and target variables) is used to build the model using the lm() function and model summary can be observed.

Model summary tells us different parameters like R-squared, Adjusted R-squared and P-values. R-squared value turned out to be 0.2066 and adjusted R-squared value is also 02066. P-value is also very less for almost all the variables, which tells us that all of them have good significance in building the model.

After observing the model summary, we test the model on the test data. And Root Mean Squared Error (RMSE) is calculated between the target and the actual values.

2. **Random Forest**:

After linear regression model, we proceed with the random forest model. Train data (Input and target variables) is used to build the model using the randomForest () function.

The following parameters are passed to the Random Forest model:

- Number of trees: 25
- Variables at each split: 5

Later the model is tested using Test data and Root Mean Square Error is calculated. RMSE value for Random Forest is calculated as **1438.334** whereas RMSE for Linear regression is obtained as **2791.178**

Observation:
We have applied both the models to the test data and obtained the predicted sales values for it. Then we have plotted the graph for original sales values and the predicted sales values obtained from both the models.

Linear Regression Model:

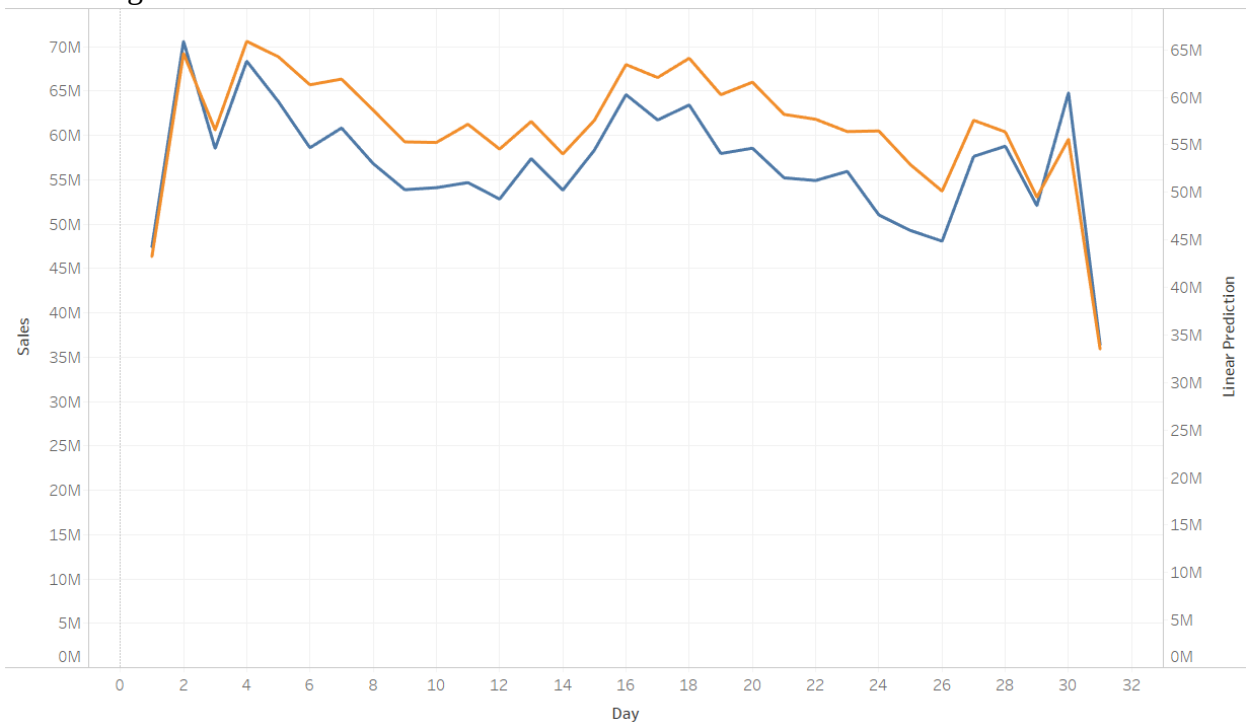

Figure-3. Graph for Original Sales vs Predicted Sales

Random Forest Model:

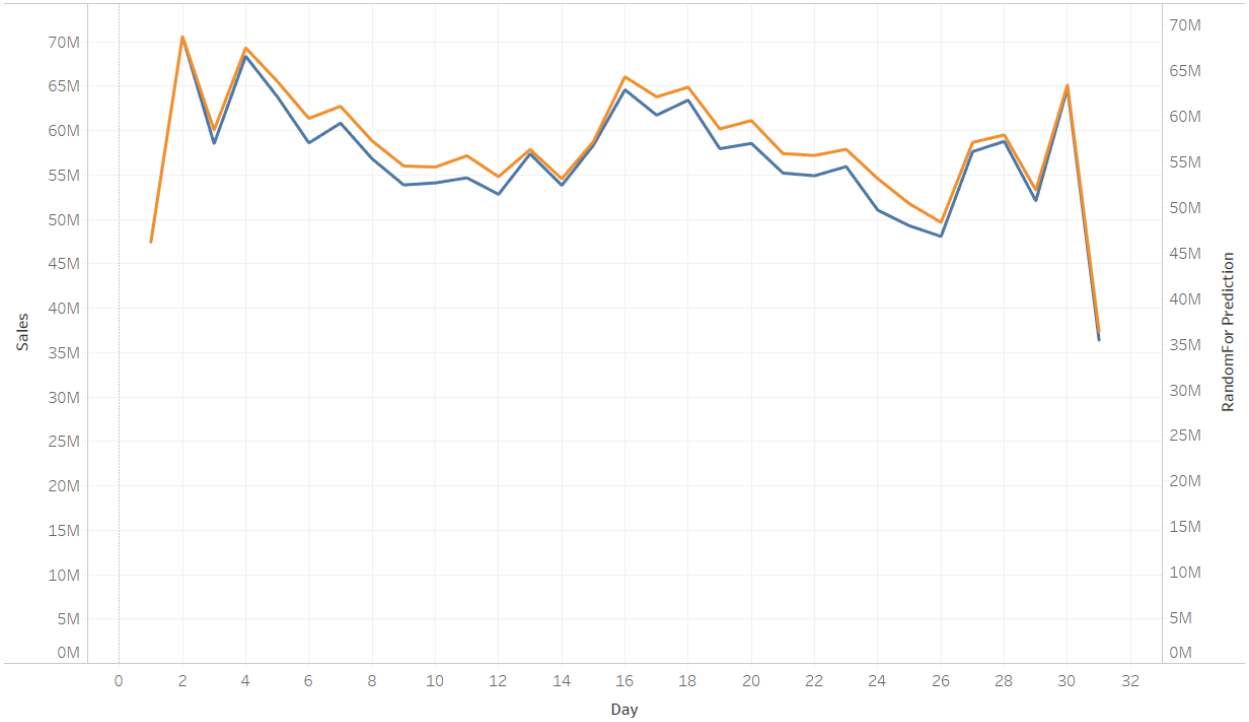

Figure-4. Graph for Original Sales vs Predicted Sales

It can be seen from the figure 3 and figure 4 that, the predicted sales data obtained from the Random Forest model aligns very closely with the original sales data.

Basing on the results obtained from RMSE values and the graph plotted for the data obtained from the both models, the RandomForest model is considered as better prediction model for this problem set.

CONCLUSION

The test error feedback we obtained from kaggle.com is **0.22698**. From what we have observed solving this problem set is, having a proper feature selection and model selection has a large impact on prediction data quality. Once the features are chosen and preprocessed correctly before training the model, the prediction error value will be reduced significantly. The prediction error we obtained during our first submission to kaggle was around 0.5. Through proper feature selection and appropriately selecting the parameters for the Random Forest model, we reduced the prediction error almost by half.