ITCS 6162

Knowledge Discovery in Databases

PROJECT REPORT

CRISP-DM PROCESS

SALES PREDICTION FOR ROSSMAN STORE DATA

Team Members:

Urvi Jayesh Gada

Sunidhi Kabra

Darshak Mehta

Nikita Nalawade

Ravil Bikmetov

TASK (Research understanding)

We are choosing the **Rossman Store Data** that can be found using the following link: https://www.kaggle.com/anshumanyp/rossman/data. This dataset provides us with the historical sales data for 1,115 Rossman stores. Based on the data that has been provided, the goal of our project and our major task is to predict the "Sales" column for the test set. This prediction will be performed using a set of relevant predictors. The project structure will follow a well-known CRISP-DM model, which is described further.

THE CRISP-DM PROCESS

CRISP-DM stands for **Cr**oss-**I**ndustry **S**tandard **P**rocess for **D**ata **M**ining which follows an iterative and adaptive live cycle consisting of 6 phases.

The 6 phases of the CRISP-DM process are as follows:

1. **Business/Research Understanding Phase**

In this phase, we define the project requirements and objectives and translate them into data mining problem definitions. We also prepare the preliminary strategy to meet objectives.

2. **Data Understanding Phase**

In this phase, we collect the data and perform EDA i.e. Exploratory Data Analysis, which helps in assessing the data quality. In this phase, we also decide which subset of data we wish to work with.

3. **Data Preparation Phase**

In this phase, we prepare for modelling and select the cases and variables that are appropriate for analysis. Also, we clean and prepare the data and perform transformations on certain variables so that the data is ready for the modelling tools.

4. **Modelling Phase**

In this phase, we select and apply one or more modelling techniques and calibrate the model settings to optimize the results. In some cases, additional data preparation may be required for supporting a technique.

5. **Evaluation Phase**

In this phase, we evaluate the models for effectiveness and determine whether the objectives that were defined have been achieved. We also check if some important facet of the problem has been sufficiently accounted for and make decisions regarding the data mining results before we deploy them.

6. **Deployment Phase**

In this phase, we make use of the models that have been created. A simple deployment would be to generate a report while a more complex example would be to implement parallel data mining algorithms in another department.

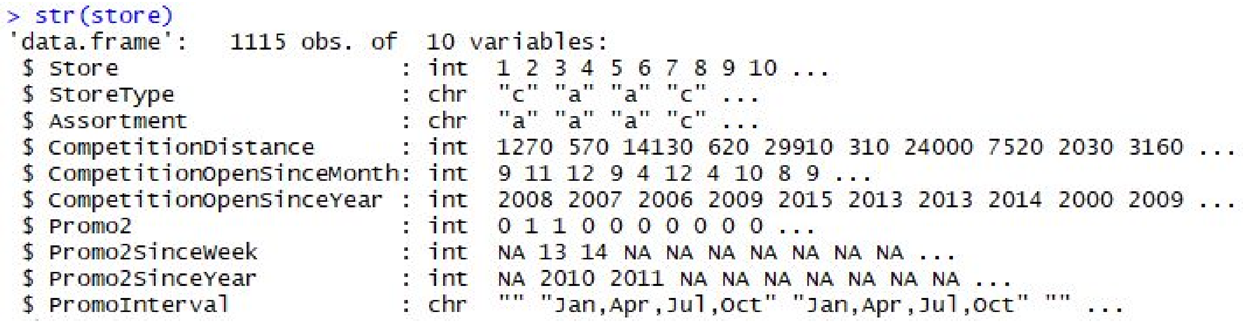
DATA UNDERSTANDING

Based on the metadata, the description of some of the fields of the dataset are provided below. The other fields in the dataset are self-explanatory.

- **Id** - an Id that represents a (Store, Date) duple within the test set
- **Store** - a unique Id for each store
- **Sales** - the turnover for any given day (our target variable)
- **Customers** - the number of customers on a given day
- **Open** - an indicator if the store was opened or not: 0 = closed, 1 = open
- **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
- **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]** - gives the 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** - Promo2 is a 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. E.g. "Feb,May,Aug,Nov" means each round starts in February, May, August, November of any given year for that store

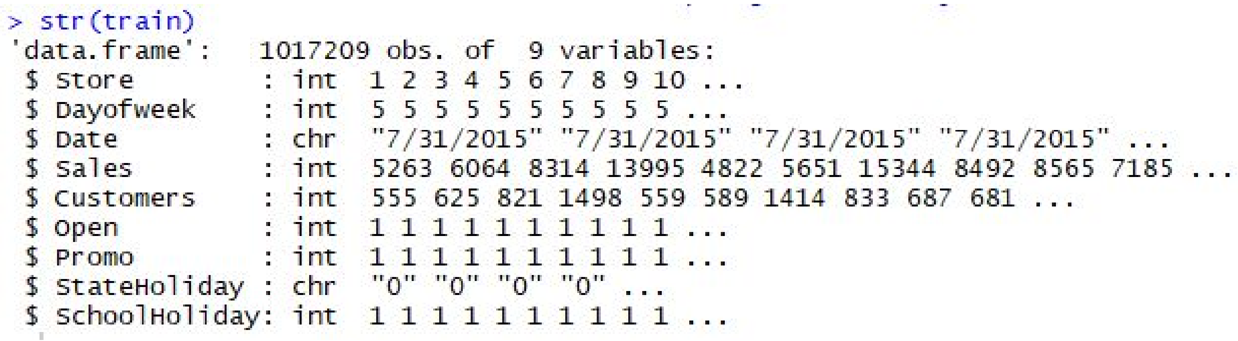
We have been provided with three files which are the test, train and the store datasets. There are a few columns which are common to all three datasets while most of the fields are different.

- Store Dataset



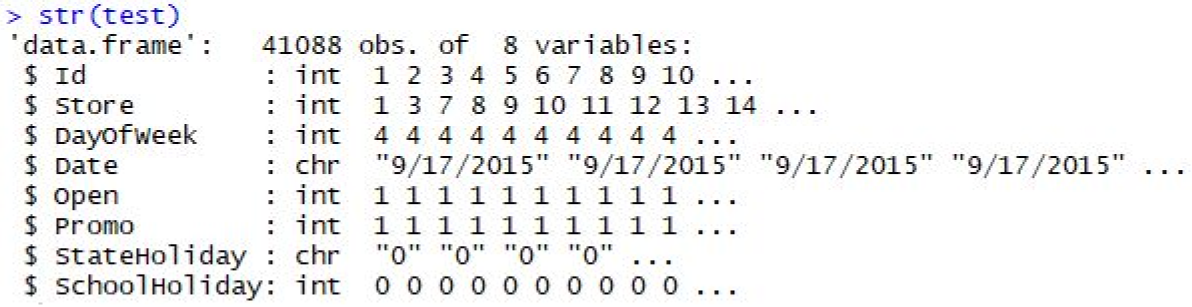
```
> str(store)
'data.frame':	1115 obs. of  10 variables:
 $ Store                    : int  1 2 3 4 5 6 7 8 9 10 ...
 $ StoreType                : chr  "c" "a" "a" "c" ...
 $ Assortment               : chr  "a" "a" "a" "c" ...
 $ CompetitionDistance      : int  1270 570 14130 620 29910 310 24000 7520 2030 3160 ...
 $ CompetitionOpenSinceMonth: int  9 11 12 9 4 12 4 10 8 9 ...
 $ CompetitionOpenSinceYear : int  2008 2007 2006 2009 2015 2013 2013 2014 2000 2009 ...
 $ Promo2                   : int  0 1 1 0 0 0 0 0 0 0 ...
 $ Promo2SinceWeek          : int  NA 13 14 NA NA NA NA NA NA NA ...
 $ Promo2SinceYear          : int  NA 2010 2011 NA NA NA NA NA NA NA ...
 $ PromoInterval            : chr  "" "Jan,Apr,Jul,Oct" "Jan,Apr,Jul,Oct" "" ...
```

- Train Dataset



```
> str(train)
'data.frame':	1017209 obs. of  9 variables:
 $ Store        : int  1 2 3 4 5 6 7 8 9 10 ...
 $ DayOfWeek    : int  5 5 5 5 5 5 5 5 5 5 ...
 $ Date         : chr  "7/31/2015" "7/31/2015" "7/31/2015" "7/31/2015" ...
 $ Sales        : int  5263 6064 8314 13995 4822 5651 15344 8492 8565 7185 ...
 $ Customers    : int  555 625 821 1498 559 589 1414 833 687 681 ...
 $ Open         : int  1 1 1 1 1 1 1 1 1 1 ...
 $ Promo        : int  1 1 1 1 1 1 1 1 1 1 ...
 $ StateHoliday : chr  "0" "0" "0" "0" ...
 $ SchoolHoliday: int  1 1 1 1 1 1 1 1 1 1 ...
```

- Test Dataset



```
> str(test)
'data.frame':	41088 obs. of  8 variables:
 $ Id           : int  1 2 3 4 5 6 7 8 9 10 ...
 $ Store        : int  1 3 7 8 9 10 11 12 13 14 ...
 $ DayOfWeek    : int  4 4 4 4 4 4 4 4 4 4 ...
 $ Date         : chr  "9/17/2015" "9/17/2015" "9/17/2015" "9/17/2015" ...
 $ Open         : int  1 1 1 1 1 1 1 1 1 1 ...
 $ Promo        : int  1 1 1 1 1 1 1 1 1 1 ...
 $ StateHoliday : chr  "0" "0" "0" "0" ...
 $ SchoolHoliday: int  0 0 0 0 0 0 0 0 0 0 ...
```

EDA

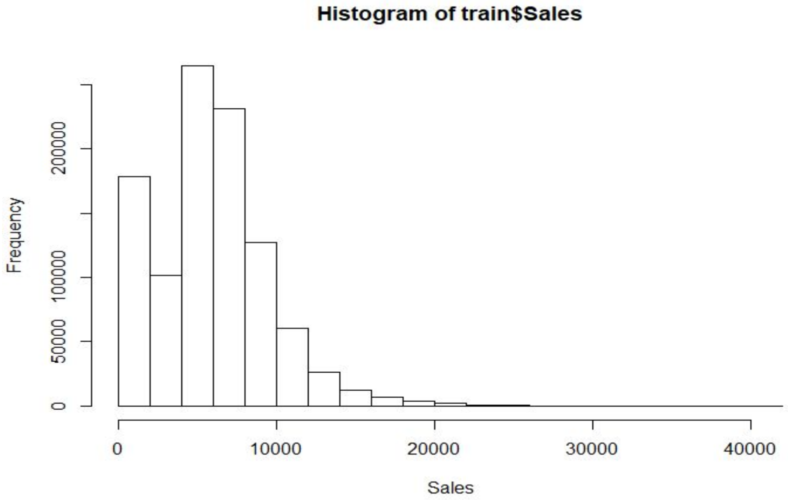
● *What is EDA?*

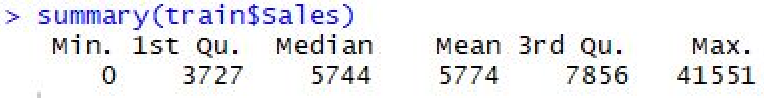
EDA stands for Exploratory Data Analysis and it is an approach for data analysis to make sense from the dataset and then figure out what questions we want to ask and the best way to manipulate the data to get the needed answers.

● *EDA on the Rossman Store Data.*

1. Descriptive Statistics on Sales

Since this is the variable to be predicted, we got the summary of the variable and checked the histogram. Outliers were already checked. Here, we also checked if it has NA values. The summary and the histogram are as follows:



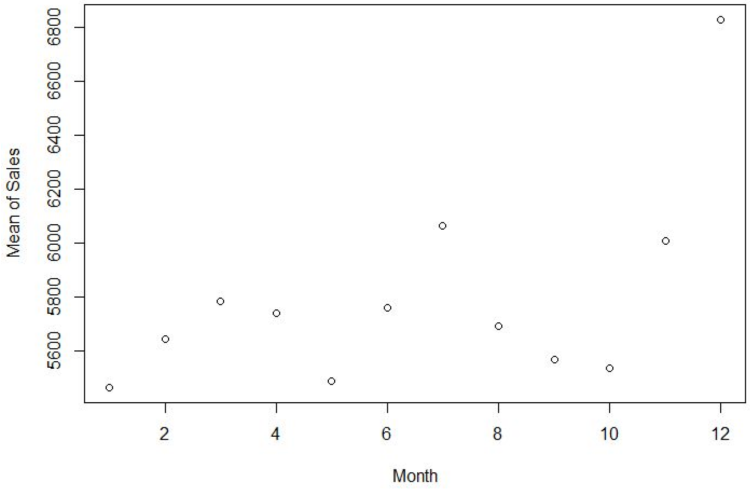


```
> summary(train$Sales)
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max. 
      0    3727    5744    5774    7856   41551 
```

From the summary we can see that the median is slightly less than that of mean which means that the data is right skewed (positive skewness). We already know that a lot of stores have high sales indicating that there are no outliers.

2. Variations in the Sales vs Months

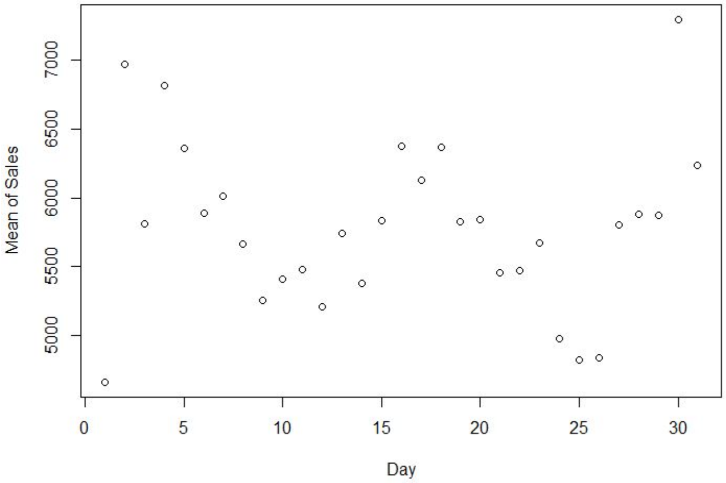
To check how the sales vales over the months of the year, we used the plot of the tapply function as got the following graph:



As we see, the sales are the maximum during the vacation period i.e. December and July. As it would be expected, the sales of the stores do tend to go higher during the holiday period as more customers visit stores due to the holidays. Hence, month will play an important role while predicting the sales of the stores.

3. Variation in the Sales vs Days

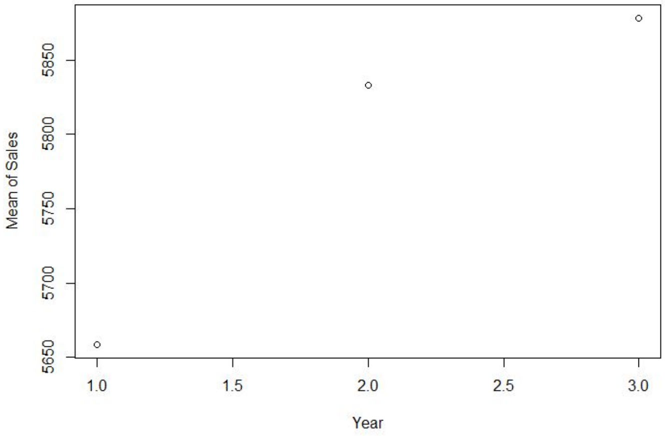
Just like for months, we also check if days of the month have any impact on the sales. The graph that we obtain is as follows:



As we know, most of the people get paid during the end of the month or during the first few days of the next month. As seen the sales are high during the start and end of the month which does match with the pay dates of the people. However, we also observe high sales during the middle of the month. Thus, further analysis may be needed to check if days does really affect the sales of the stores.

4. Variation in the Sales vs Years

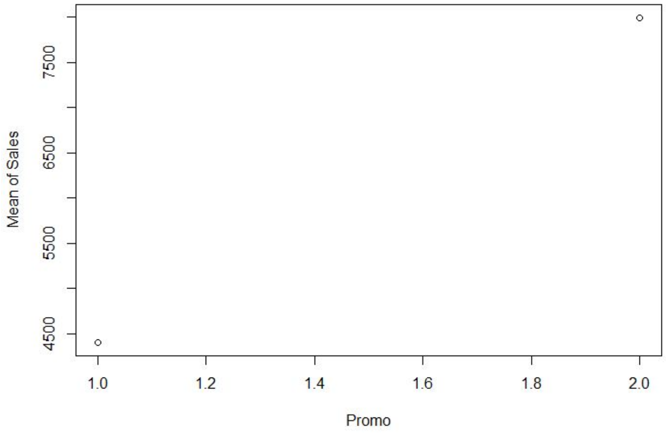
In the train dataset, we have data for the years 2013, 2014 and 2015. So, we will check the trend of the sales for these 3 months. The graph that we obtained is as follows:



As we can see that the sales goes on increasing every year. However, we are unsure as to what factor lead to the increase in the sales over the years. Thus, in order to use this for the prediction of the sales, we will have to first analyse the factors influencing the trend of increased saled over the years.

5. Variation in the Sales vs Promo

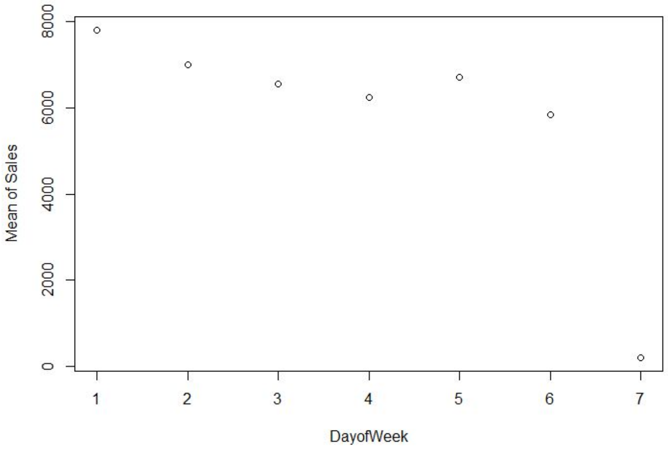
Also, promotional strategies always have a good influence over the sales. To determine that, we plot the following graph:



Thus, we see that the promotional events carried out do result in higher sales as compared to sales for stores that do not have the promotional event. However, as it has been noticed previously, “Promo…” columns have many missing values. The weight of these values contains about 50% of the total number of values for Promo2SinceWeek, Promo2SinceYear, and PromoInterval variables.

6. Variation in the Sales vs DayofWeek

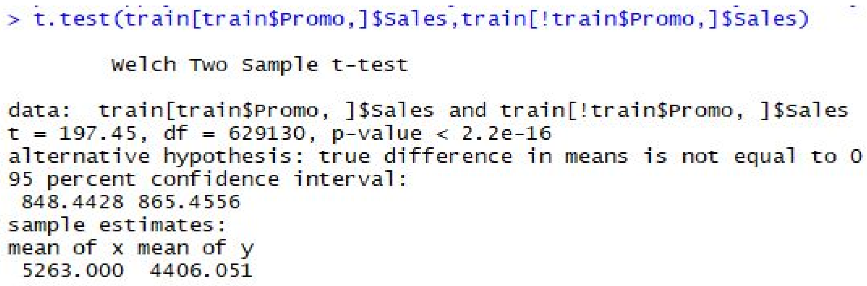
We know that the stores that are shut on Sunday have a 0 sales value while on open days the sales value is varying. Thus, to determine if the sales does get affected by the day of the week, we have the following graph:



Thus, we see that people tend to go to store during the beginning and end of the working days i.e. on Monday and Friday. And even from this graph the sales on Sunday when the stores are closed is 0.

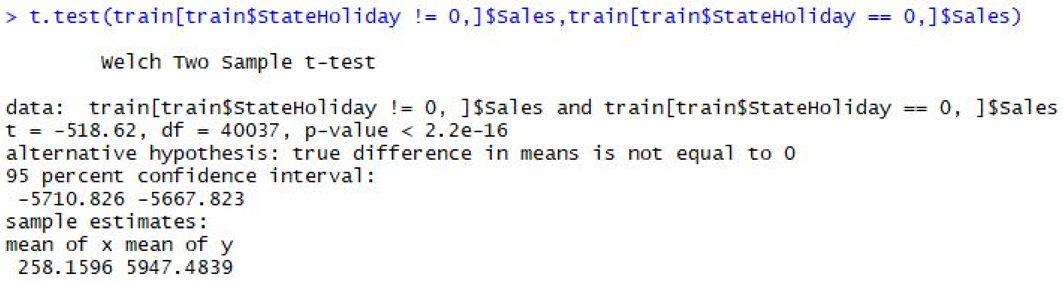
7. T-test for Promo

As we see, promotions have a very significant effect on sales, which are about 40 per cent higher on average during promotions.

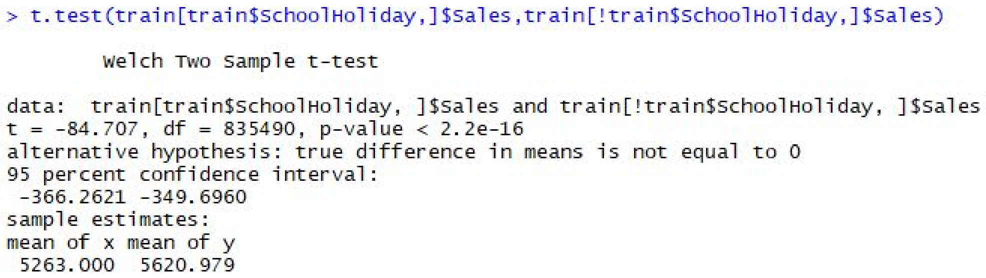


```
> t.test(train[train$Promo,]$Sales,train[!train$Promo,]$Sales)

        Welch Two Sample t-test

data:  train[train$Promo, ]$Sales and train[!train$Promo, ]$Sales
t = 197.45, df = 629130, p-value < 2.2e-16
alternative hypothesis: true difference in means is not equal to 0
95 percent confidence interval:
 848.4428 865.4556
sample estimates:
mean of x mean of y 
 5263.000  4406.051 
```

8. T-test for StateHoliday and SchoolHoliday



```
> t.test(train[train$StateHoliday != 0,]$Sales,train[train$StateHoliday == 0,]$Sales)

        Welch Two Sample t-test

data:  train[train$StateHoliday != 0, ]$Sales and train[train$StateHoliday == 0, ]$Sales
t = -518.62, df = 40037, p-value < 2.2e-16
alternative hypothesis: true difference in means is not equal to 0
95 percent confidence interval:
 -5710.826 -5667.823
sample estimates:
mean of x mean of y 
 258.1596 5947.4839 
```



```
> t.test(train[train$SchoolHoliday,]$Sales,train[!train$SchoolHoliday,]$Sales)

        Welch Two Sample t-test

data:  train[train$SchoolHoliday, ]$Sales and train[!train$SchoolHoliday, ]$Sales
t = -84.707, df = 835490, p-value < 2.2e-16
alternative hypothesis: true difference in means is not equal to 0
95 percent confidence interval:
 -366.2621 -349.6960
sample estimates:
mean of x mean of y
 5263.000  5620.979
```

Thus, we observe absurd variations in the variation of sales depending on StateHoliday and SchoolHoliday. Thus, we need further analysis to determine their impact of the sales of the stores.

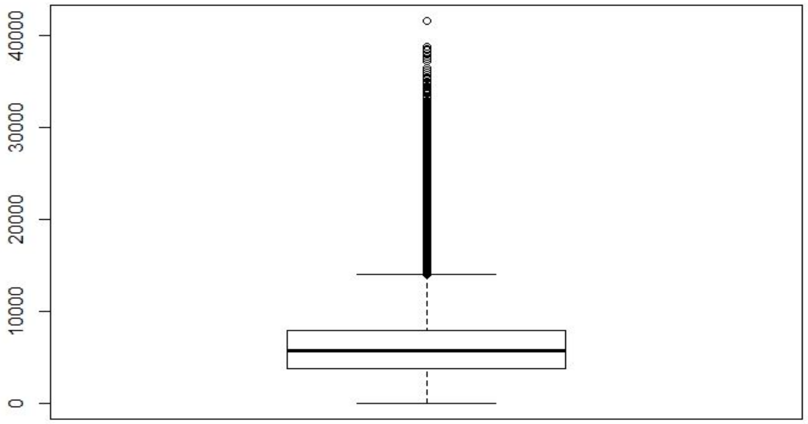
DATA PREPARATION AND PREPROCESSING

- ***What is Data Preprocessing?***

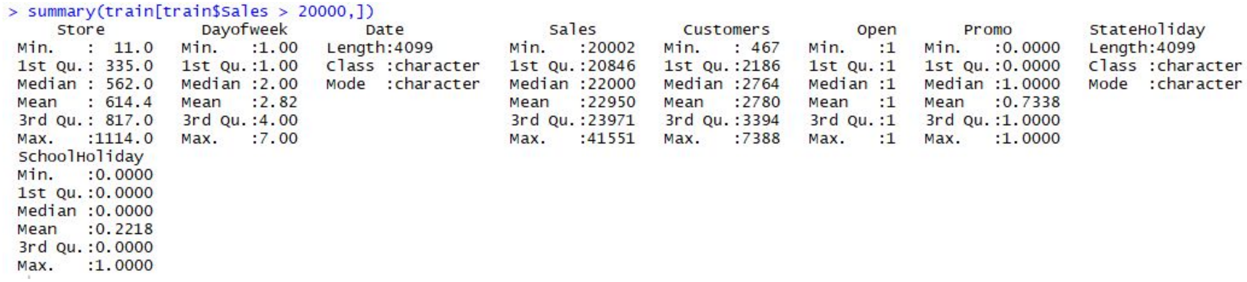
It is the technique in data mining that involves transforming the raw data into an understandable form. Most of the real-world data is incomplete, inconsistent and lacking certain behaviors or trends. To resolve these issues, data preprocessing must be carried out before going further with any data mining algorithms.

1. Determining the Outliers in Sales

The sales variable is presented in the train dataset is the target variable, e.g., that is to be predicted. Thus, the first step that we implemented was to check for the outliers in this variable. The box plot of the sales variable is as follows:



At the first instance, it may appear that the values which are greater than 20000 are outliers. However, before assuming that those may be outliers, we should check for the occurrences of those values in the dataset. This is done by taking the summary of sales variable for values greater than 20000. The results obtained are as follows:

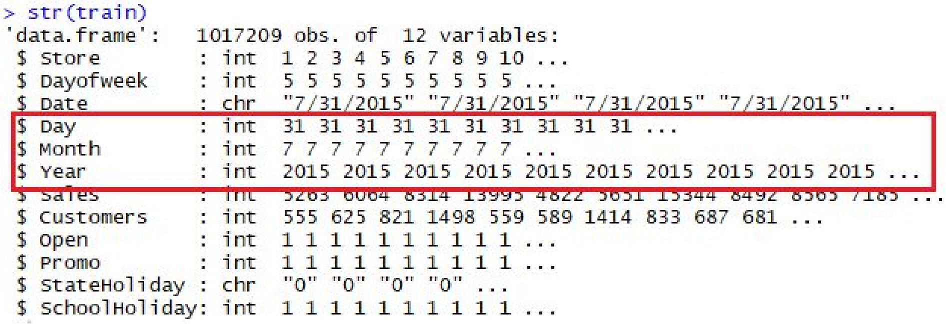


```
> summary(train[train$Sales > 20000,])
     Store          Dayofweek        Date               Sales         Customers         Open       Promo          StateHoliday      
 Min.   :  11.0   Min.   :1.00   Length:4099        Min.   :20002   Min.   : 467   Min.   :1   Min.   :0.0000   Length:4099       
 1st Qu.: 335.0   1st Qu.:1.00   Class :character   1st Qu.:20846   1st Qu.:2186   1st Qu.:1   1st Qu.:0.0000   Class :character  
 Median : 562.0   Median :2.00   Mode  :character   Median :22000   Median :2764   Median :1   Median :1.0000   Mode  :character  
 Mean   : 614.4   Mean   :2.82                      Mean   :22950   Mean   :2780   Mean   :1   Mean   :0.7338                     
 3rd Qu.: 817.0   3rd Qu.:4.00                      3rd Qu.:23971   3rd Qu.:3394   3rd Qu.:1   3rd Qu.:1.0000                     
 Max.   :1114.0   Max.   :7.00                      Max.   :41551   Max.   :7388   Max.   :1   Max.   :1.0000                     
 SchoolHoliday   
 Min.   :0.0000  
 1st Qu.:0.0000  
 Median :0.0000  
 Mean   :0.2218  
 3rd Qu.:0.0000  
 Max.   :1.0000  
```

As we see that there are a lot of stores that do have extremely high sales, we conclude that sales greater than 20000 are not outliers after all.

2. Working on the Date field in Train and Test Dataset

The date field is expressed as a character in the form of MM/DD/YYYY. Since we know that the sales will greatly be affected by the days and months we separate this field in three variables “Day”, “Month” and “Year” and convert it to integer data type.



```
> str(train)
'data.frame':	1017209 obs. of  12 variables:
 $ Store        : int  1 2 3 4 5 6 7 8 9 10 ...
 $ Dayofweek    : int  5 5 5 5 5 5 5 5 5 5 ...
 $ Date         : chr  "7/31/2015" "7/31/2015" "7/31/2015" "7/31/2015" ...
 $ Day          : int  31 31 31 31 31 31 31 31 31 31 ...
 $ Month        : int  7 7 7 7 7 7 7 7 7 7 ...
 $ Year         : int  2015 2015 2015 2015 2015 2015 2015 2015 2015 2015 ...
 $ Sales        : int  5263 6064 8314 13995 4822 5651 15344 8492 8565 7185 ...
 $ Customers    : int  555 625 821 1498 559 589 1414 833 687 681 ...
 $ Open         : int  1 1 1 1 1 1 1 1 1 1 ...
 $ Promo        : int  1 1 1 1 1 1 1 1 1 1 ...
 $ StateHoliday : chr  "0" "0" "0" "0" ...
 $ SchoolHoliday: int  1 1 1 1 1 1 1 1 1 1 ...
```

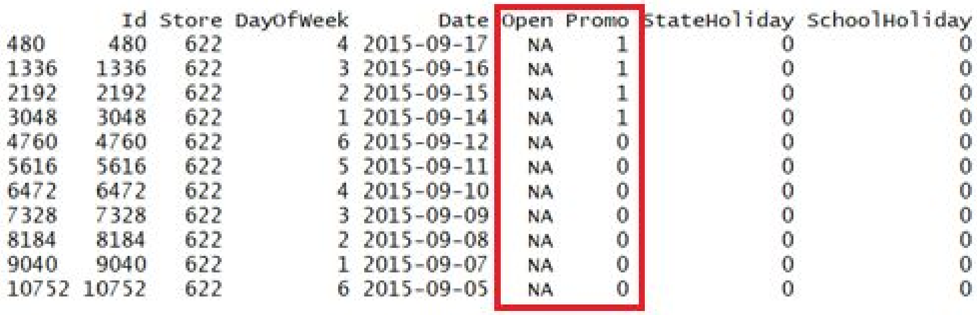
3. Consideration of Customers variable for prediction

Based on an intuition, Customer variable can affect the Sales variable, which is the target for our prediction. This initial thought can be confirmed with the corresponding correlation table, which shows a strong correlation between these variables. However, Customer variable is not presented in the test set and the initial task does not require to predict the number of customers that walk in the store, but the sales of the store. Hence, it could be useful to fit a model to predict the number of customers based on the given variables. After that, the Customers variable can be used for the further prediction of Sales.

4. Missing Values in the Open variable of the Test Dataset.

When we check for NA values in the Open variable, we get 11 values, but they are all for the same store ‘622’. However, we also see that for the first 4 entries we have a promotional event which is carried out. Also, these NA values are only for the working days i.e. Monday to Saturday. From the dataset, for a Sunday i.e. DayofWeek = 7, the stores are closed, and the

Promo field is also 0. Thus, we can assume that the store was open for the given days and thus we set this value to Open = 1 for all the records.



```
         Id Store DayOfWeek       Date Open Promo StateHoliday SchoolHoliday
480     480   622         4 2015-09-17   NA     1            0             0
1336   1336   622         3 2015-09-16   NA     1            0             0
2192   2192   622         2 2015-09-15   NA     1            0             0
3048   3048   622         1 2015-09-14   NA     1            0             0
4760   4760   622         6 2015-09-12   NA     0            0             0
5616   5616   622         5 2015-09-11   NA     0            0             0
6472   6472   622         4 2015-09-10   NA     0            0             0
7328   7328   622         3 2015-09-09   NA     0            0             0
8184   8184   622         2 2015-09-08   NA     0            0             0
9040   9040   622         1 2015-09-07   NA     0            0             0
10752 10752   622         6 2015-09-05   NA     0            0             0
```

5. Not considering the Promo2SinceWeek, Promo2SinceYear and PromoInterval

The Promo2SinceWeek, Promo2SinceYears, PromoInterval have many (about 550 records) values which are null. It contains about 50% of the overall number of records for this variable. Therefore, these three fields can't be used for the prediction of accurate results.

6. Sales of Closed Stores

It's assumed that the sales of a closed shop are 0. Therefore, there is no use of predicting the sales of a closed shop. So, all the data related to closed shops is removed which reduces the size of the data by almost one-fifth of the original data.

7. Merging fields from Store dataset to the Train dataset

The sales of a store will always be affected by the presence of other stores in the vicinity. The Store dataset has variables like CompetitionDistance, CompetitionOpenSinceMonth, and CompetitionOpenSinceYear. Whereas the variable Sales is present in the Train dataset. We would like to use CompetitionDistance, CompetitionOpenSinceMonth, and CompetitionOpenSinceYear to determine the sales of the store. Therefore, we add CompetitionDistance, CompetitionOpenSinceMonth, and CompetitionOpenSinceYear to the Train dataset.

8. Merging fields from Store dataset to the Test dataset

After merging the fields from the Store dataset to the Test dataset and building the model, this model should be tested. Therefore, the corresponding Test dataset should be developed. This dataset should have the same variables as the updated after merging Train dataset. Merging procedure of Store and Test datasets is similar to the merging procedure described in part 7. Besides, Customers variable should be predicted for the test data and added to it.

INTERPRETATION OF THE RESULTS

Based on the data preprocessing and EDA completion, we have the following findings:

1. Since the shops remain closed on Sundays and holidays, their sales will be 0. This is an important parameter to be considered for further analysis.
2. The sales are affected by the days of the month and specific months. Besides, the sales increase over the years. These factors must be a part of any further analysis which may be performed.
3. Promo, StateHoliday and SchoolHoliday have an impact o the sales. These trends must be considered for further analysis.
4. Since the Sales field does not have any outliers, there are no erroneous values which may be present.
5. There is not much difference in the mean and median of the sales. Thus, the data is not heavily skewed.
6. Merging the data from the train and store dataset was imperative, as the train dataset did not account for the competitor’s effect on the sales. Merging the data from store and test dataset should be completed for a similar reason.
7. Adding Customers variable will increase the accuracy of the results due to its strong correlation with the Sales variable.
8. Sales were affected by the day of the week. Thus, prediction model must consider that factor as well.
9. There were certain fields like Promo2SinceWeek, Promo2SinceYear and PromoInterval which had a lot of null values. Predicting these values would be difficult. Thus, these fields must be avoided as predictions made on their references would produce erroneous results.

MODELING PHASE

There are several well-known algorithms that can be used for predicting Sales as a numerical variable. We will use a one of the algorithms for the modelling that can predict Sales:

- **Multiple regression** can be used for estimating store sales using a set of numerical and categorical predictors. A regression fit line will be obtained using the corresponding independent predictors. Independence of variables can be confirmed through correlation matrix. Adjusted R^2 score can be used for prediction evaluation along with the lift-chart and ROC curve parameters.

- **Classification using Single Tree** can be used for a given limit (threshold), estimating store sales to be above or below the limit using the chosen set of predictors. During this classification, a training set is recursively divided until each following division consists of examples from one class. The whole classification process is based on recursive partition and divide and conquer algorithms. Specifically, a training set is recursively divided until each following division consists of examples from one class.

- **Classification using Random trees (forest)** can be used for estimating sales to be above or below a certain threshold. Such classification can be used as a way of averaging multiple deep decision trees trained on different parts of the same training set, with the goal of reducing the variance.

- **K-nearest neighbor (kNN) algorithm** can be utilized for an initial clustering of stores similar in a certain set of parameters, e.g., Customer, Promo, Assortment, CompetitionDistance, etc. In kNN algorithm, an object is classified by a majority vote of its neighbors and being assigned to the class most common among its k nearest neighbors. After clustering is completed, the prediction of sales can be performed for the obtained clusters using one of the described above techniques: Single Tree or Random Trees.

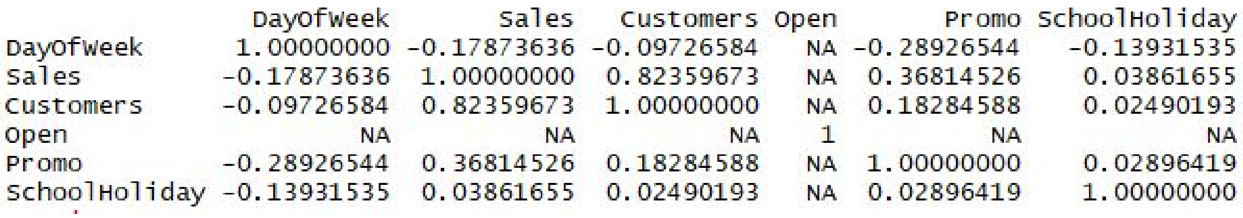
- **Chosen model**

Random forest algorithm has been chosen for modeling implementation. It contains multiple learning algorithms to obtain better predictive performance and allows for much more flexible structure than singular algorithms [1, 6]. Operating by constructing and averaging multiple deep decision trees at training time, random forest has been found an efficient algorithm for classification and regression tasks. Similar to kNN algorithm, it uses weighted neighborhood scheme. It is considered as the best classifier, since it has less variance, relatively high accuracy, and reduces overfitting [1, 6]. The linear regression has been also implemented in our project for comparison purposes. It can be noticed that Random Forest algorithm has shown better performance. Therefore, it has been chosen as the final model.

Predictors: Customers, DayOfWeek, DayOfMonth, Year, Open, SchoolHoliday, CompetitionDistance, CompetitionOpenSinceMonth, CompetitionOpenSinceYear, Promo, Promo2, Month, and Day.
Target variable: Sales

RESULTS AND DISCUSSION

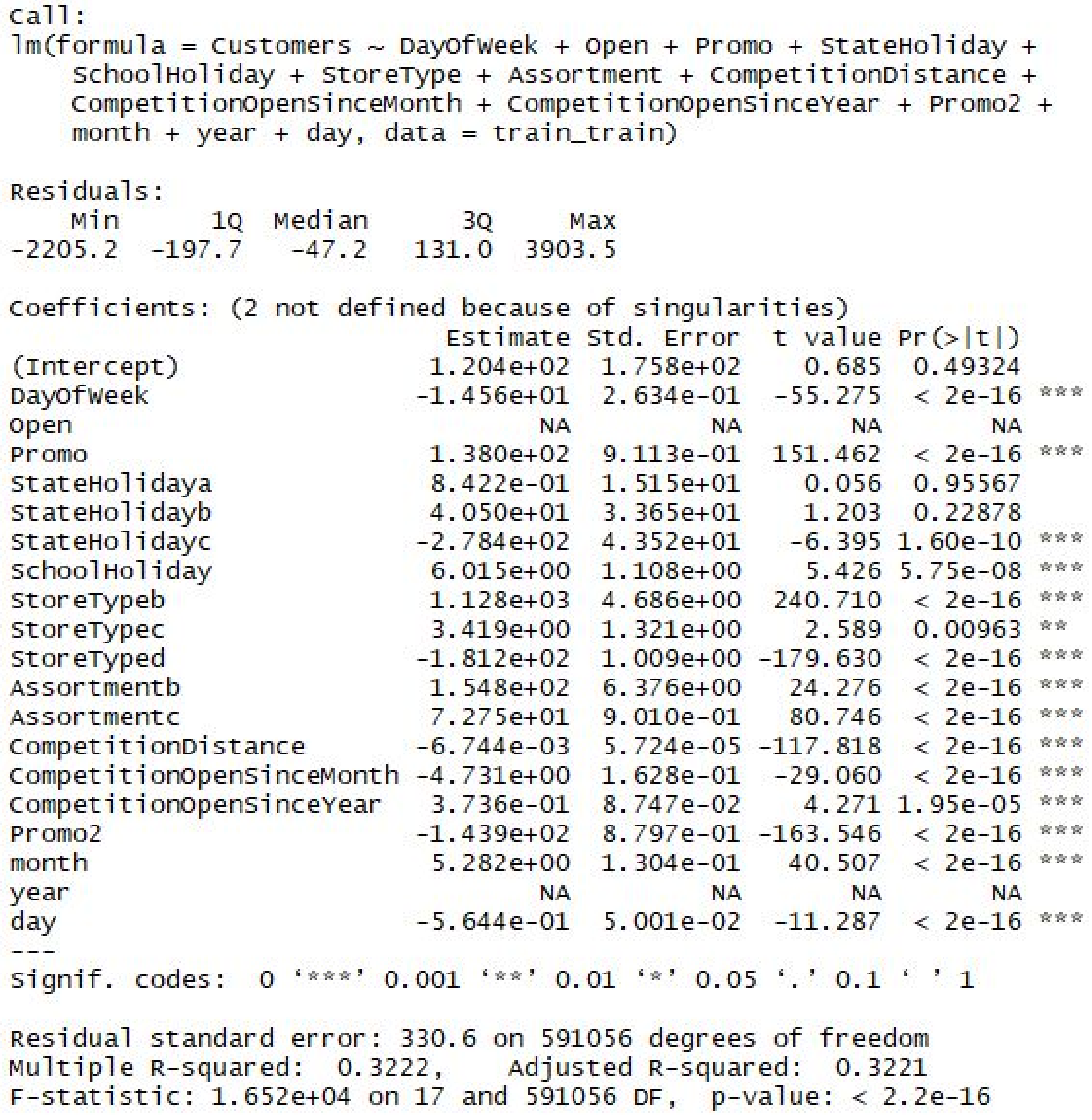
Correlation matrix of train variable: We can observe that customers is highly correlated (more than 82%) to the Sales. So we would require the customers predictor in the test dataset as well.



```
              DayOfWeek       Sales   Customers Open       Promo SchoolHoliday
DayOfWeek     1.00000000 -0.17873636 -0.09726584   NA -0.28926544   -0.13931535
Sales        -0.17873636  1.00000000  0.82359673   NA  0.36814526    0.03861655
Customers    -0.09726584  0.82359673  1.00000000   NA  0.18284588    0.02490193
Open                  NA          NA          NA    1          NA            NA
Promo        -0.28926544  0.36814526  0.18284588   NA  1.00000000    0.02896419
SchoolHoliday -0.13931535 0.03861655  0.02490193   NA  0.02896419    1.00000000
```

1. **Linear Regression including Open variable**

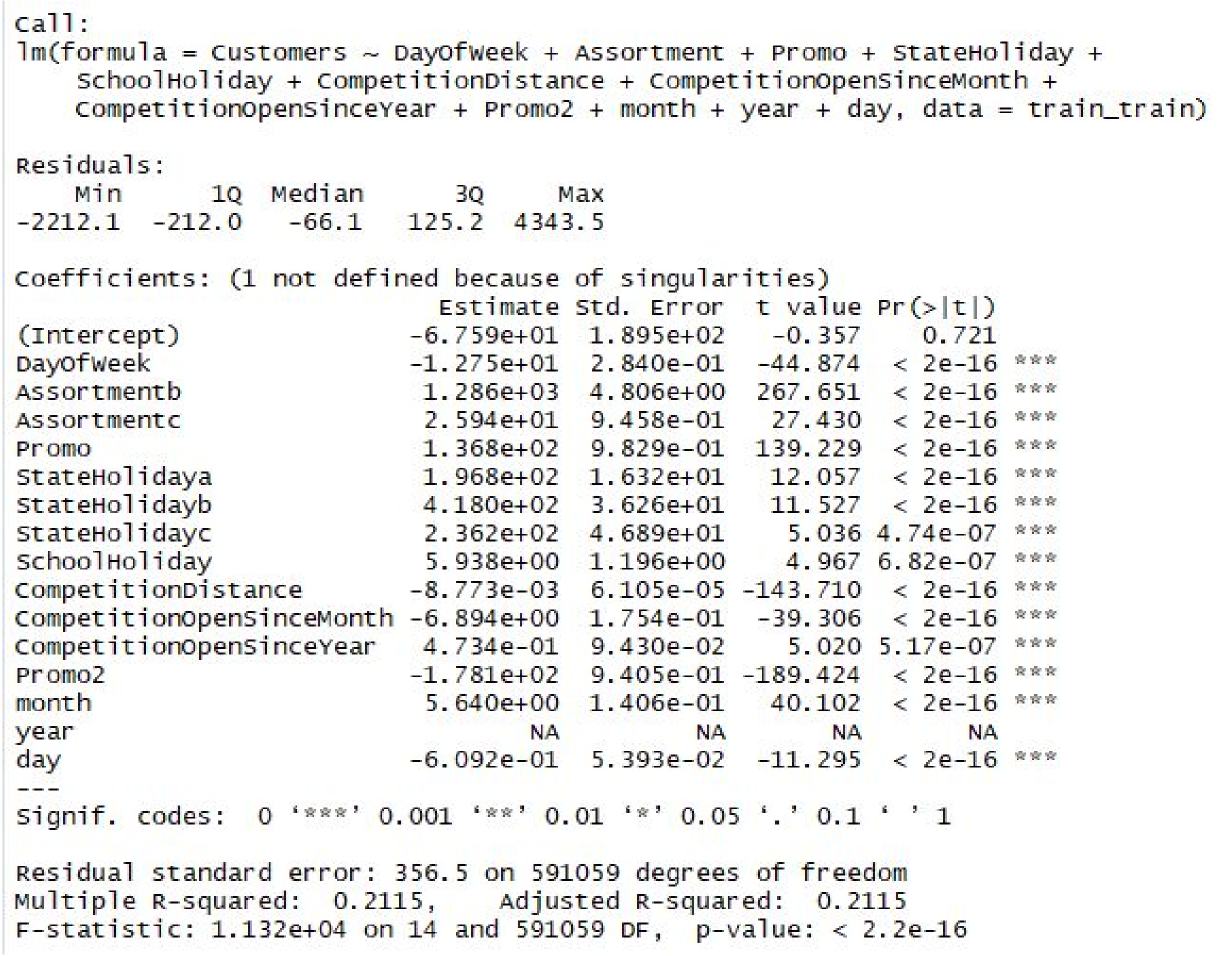
In this model, we considered only opened stores for Sales prediction.



```
Call:
lm(formula = Customers ~ DayOfWeek + Open + Promo + StateHoliday +
    SchoolHoliday + StoreType + Assortment + CompetitionDistance +
    CompetitionOpenSinceMonth + CompetitionOpenSinceYear + Promo2 +
    month + year + day, data = train_train)

Residuals:
    Min      1Q  Median      3Q     Max
-2205.2  -197.7   -47.2   131.0  3903.5

Coefficients: (2 not defined because of singularities)
                           Estimate Std. Error  t value Pr(>|t|)
(Intercept)               1.204e+02  1.758e+02    0.685  0.49324
DayOfWeek                -1.456e+01  2.634e-01  -55.275  < 2e-16 ***
Open                             NA         NA       NA       NA
Promo                     1.380e+02  9.113e-01  151.462  < 2e-16 ***
StateHolidaya             8.422e-01  1.515e+01    0.056  0.95567
StateHolidayb             4.050e+01  3.365e+01    1.203  0.22878
StateHolidayc            -2.784e+02  4.352e+01   -6.395 1.60e-10 ***
SchoolHoliday             6.015e+00  1.108e+00    5.426 5.75e-08 ***
StoreTypeb                1.128e+03  4.686e+00  240.710  < 2e-16 ***
StoreTypec                3.419e+00  1.321e+00    2.589  0.00963 **
StoreTyped               -1.812e+02  1.009e+00 -179.630  < 2e-16 ***
Assortmentb               1.548e+02  6.376e+00   24.276  < 2e-16 ***
Assortmentc               7.275e+01  9.010e-01   80.746  < 2e-16 ***
CompetitionDistance      -6.744e-03  5.724e-05 -117.818  < 2e-16 ***
CompetitionOpenSinceMonth -4.731e+00 1.628e-01  -29.060  < 2e-16 ***
CompetitionOpenSinceYear  3.736e-01  8.747e-02    4.271 1.95e-05 ***
Promo2                   -1.439e+02  8.797e-01 -163.546  < 2e-16 ***
month                     5.282e+00  1.304e-01   40.507  < 2e-16 ***
year                             NA         NA       NA       NA
day                      -5.644e-01  5.001e-02  -11.287  < 2e-16 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 330.6 on 591056 degrees of freedom
Multiple R-squared:  0.3222,    Adjusted R-squared:  0.3221
F-statistic: 1.652e+04 on 17 and 591056 DF,  p-value: < 2.2e-16
```

2. **Linear Regression without Open variable**

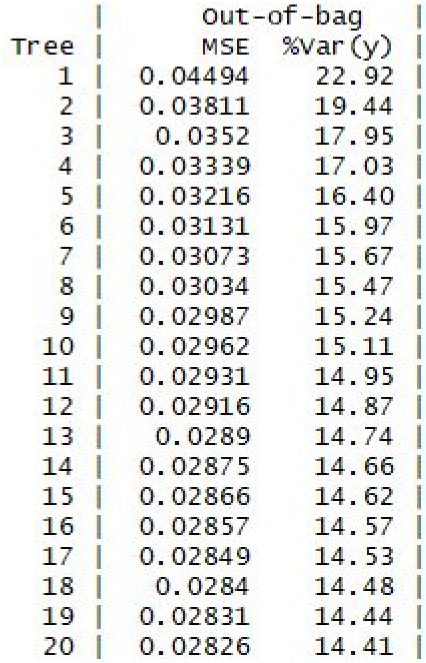
In this model, we haven't considered only opened stores for Sales prediction.



```
Call:
lm(formula = Customers ~ DayOfWeek + Assortment + Promo + StateHoliday +
    SchoolHoliday + CompetitionDistance + CompetitionOpenSinceMonth +
    CompetitionOpenSinceYear + Promo2 + month + year + day, data = train_train)

Residuals:
    Min      1Q  Median      3Q     Max
-2212.1  -212.0   -66.1   125.2  4343.5

Coefficients: (1 not defined because of singularities)
                            Estimate Std. Error  t value Pr(>|t|)
(Intercept)               -6.759e+01  1.895e+02   -0.357    0.721
DayOfWeek                 -1.275e+01  2.840e-01  -44.874  < 2e-16 ***
Assortmentb                1.286e+03  4.806e+00  267.651  < 2e-16 ***
Assortmentc                2.594e+01  9.458e-01   27.430  < 2e-16 ***
Promo                      1.368e+02  9.829e-01  139.229  < 2e-16 ***
StateHolidaya              1.968e+02  1.632e+01   12.057  < 2e-16 ***
StateHolidayb              4.180e+02  3.626e+01   11.527  < 2e-16 ***
StateHolidayc              2.362e+02  4.689e+01    5.036 4.74e-07 ***
SchoolHoliday              5.938e+00  1.196e+00    4.967 6.82e-07 ***
CompetitionDistance       -8.773e-03  6.105e-05 -143.710  < 2e-16 ***
CompetitionOpenSinceMonth -6.894e+00  1.754e-01  -39.306  < 2e-16 ***
CompetitionOpenSinceYear   4.734e-01  9.430e-02    5.020 5.17e-07 ***
Promo2                    -1.781e+02  9.405e-01 -189.424  < 2e-16 ***
month                      5.640e+00  1.406e-01   40.102  < 2e-16 ***
year                              NA         NA       NA       NA
day                       -6.092e-01  5.393e-02  -11.295  < 2e-16 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 356.5 on 591059 degrees of freedom
Multiple R-squared:  0.2115,	Adjusted R-squared:  0.2115
F-statistic: 1.132e+04 on 14 and 591059 DF,  p-value: < 2.2e-16
```

3. **Random Forest**

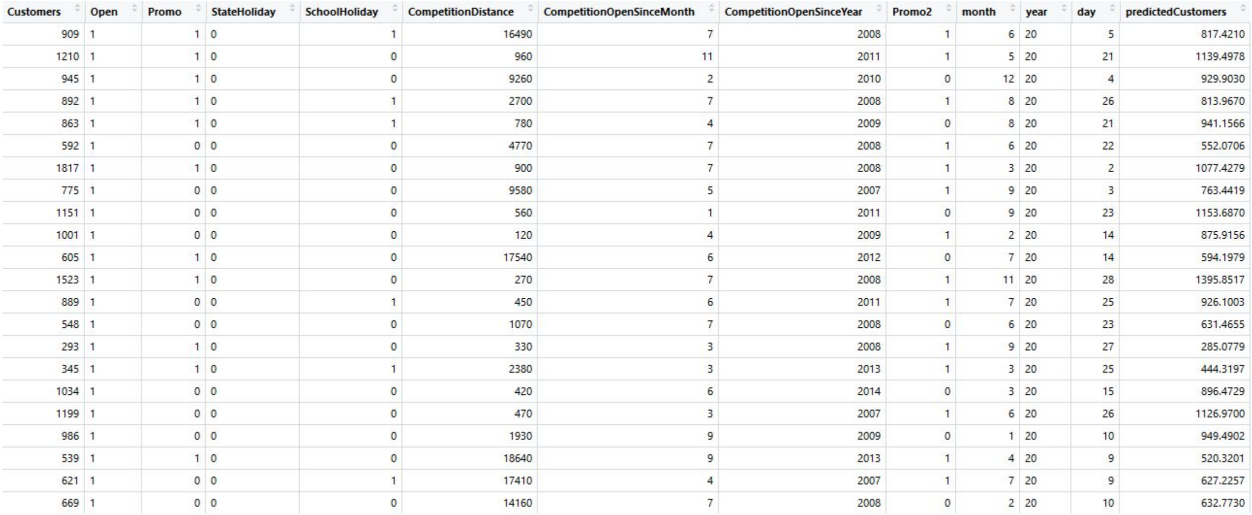
Training Model for prediction of Customer Values: Using randomForest library, allows train the model to predict the Customers variable values. We have trained model on the train data set initially.



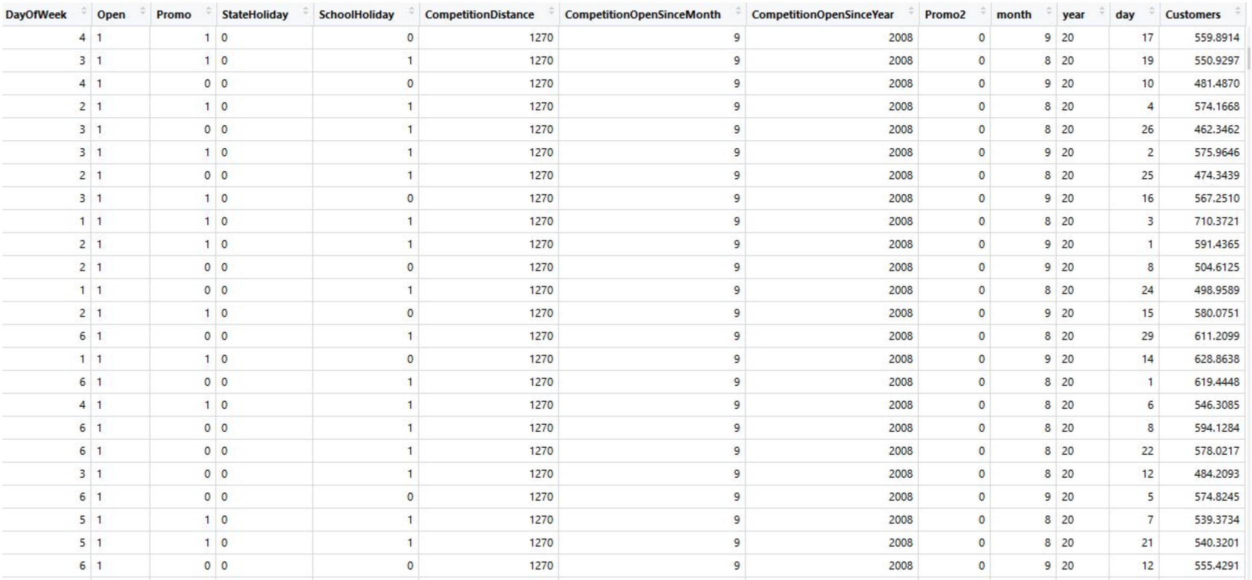
```
     |       Out-of-bag   |
Tree |       MSE  %Var(y) |
   1 |  0.04494    22.92 |
   2 |  0.03811    19.44 |
   3 |   0.0352    17.95 |
   4 |  0.03339    17.03 |
   5 |  0.03216    16.40 |
   6 |  0.03131    15.97 |
   7 |  0.03073    15.67 |
   8 |  0.03034    15.47 |
   9 |  0.02987    15.24 |
  10 |  0.02962    15.11 |
  11 |  0.02931    14.95 |
  12 |  0.02916    14.87 |
  13 |   0.0289    14.74 |
  14 |  0.02875    14.66 |
  15 |  0.02866    14.62 |
  16 |  0.02857    14.57 |
  17 |  0.02849    14.53 |
  18 |   0.0284    14.48 |
  19 |  0.02831    14.44 |
  20 |  0.02826    14.41 |
```

MSE of 0.02826 and variance of 14.41% show good performance of Random Forest algorithm. More detailed comments about Random Forest algorithm implementation are given in [7].

Testing Model for prediction of Customer Values: Using randomForest library lets test the trained model to predict the customers value for the test dataset.

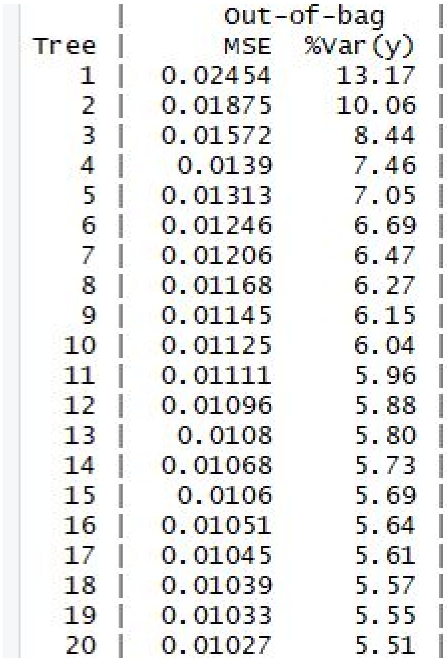


Customers	Open	Promo	StateHoliday	SchoolHoliday	CompetitionDistance	CompetitionOpenSinceMonth	CompetitionOpenSinceYear	Promo2	month	year	day	predictedCustomers
909	1	1	0	1	16490	7	2008	1	6	20	5	817.4210
1210	1	1	0	0	960	11	2011	1	5	20	21	1139.4978
945	1	1	0	0	9260	2	2010	0	12	20	4	929.9030
892	1	1	0	1	2700	7	2008	1	8	20	26	813.9670
863	1	1	0	1	780	4	2009	0	8	20	21	941.1566
592	1	0	0	0	4770	7	2008	1	6	20	22	552.0706
1817	1	1	0	0	900	7	2008	1	3	20	2	1077.4279
775	1	0	0	0	9580	5	2007	1	9	20	3	763.4419
1151	1	0	0	0	560	1	2011	0	9	20	23	1153.6870
1001	1	0	0	0	120	4	2009	1	2	20	14	875.9156
605	1	1	0	0	17540	6	2012	0	7	20	14	594.1979
1523	1	1	0	0	270	7	2008	1	11	20	28	1395.8517
889	1	0	0	1	450	6	2011	1	7	20	25	926.1003
548	1	0	0	0	1070	7	2008	0	6	20	23	631.4655
293	1	1	0	0	330	3	2008	1	9	20	27	285.0779
345	1	1	0	1	2380	3	2013	1	3	20	25	444.3197
1034	1	0	0	0	420	6	2014	0	3	20	15	896.4729
1199	1	0	0	0	470	3	2007	1	6	20	26	1126.9700
986	1	0	0	0	1930	9	2009	0	1	20	10	949.4902
539	1	1	0	0	18640	9	2013	1	4	20	9	520.3201
621	1	0	0	1	17410	4	2007	1	7	20	9	627.2257
669	1	0	0	0	14160	7	2008	0	2	20	10	632.7730



DayOfWeek	Open	Promo	StateHoliday	SchoolHoliday	CompetitionDistance	CompetitionOpenSinceMonth	CompetitionOpenSinceYear	Promo2	month	year	day	Customers
4	1	1	0	0	1270	9	2008	0	9	20	17	559.8914
3	1	1	0	1	1270	9	2008	0	8	20	19	550.9297
4	1	0	0	0	1270	9	2008	0	9	20	10	481.4870
2	1	1	0	1	1270	9	2008	0	8	20	4	574.1668
3	1	0	0	1	1270	9	2008	0	8	20	26	462.3462
3	1	1	0	1	1270	9	2008	0	9	20	2	575.9646
2	1	0	0	1	1270	9	2008	0	8	20	25	474.3439
3	1	1	0	0	1270	9	2008	0	9	20	16	567.2510
1	1	1	0	1	1270	9	2008	0	8	20	3	710.3721
2	1	1	0	1	1270	9	2008	0	9	20	1	591.4365
2	1	0	0	0	1270	9	2008	0	9	20	8	504.6125
1	1	0	0	1	1270	9	2008	0	8	20	24	498.9589
2	1	1	0	0	1270	9	2008	0	9	20	15	580.0751
6	1	0	0	1	1270	9	2008	0	8	20	29	611.2099
1	1	1	0	0	1270	9	2008	0	9	20	14	628.8638
6	1	0	0	1	1270	9	2008	0	8	20	1	619.4448
4	1	1	0	1	1270	9	2008	0	8	20	6	546.3085
6	1	0	0	1	1270	9	2008	0	8	20	8	594.1284
6	1	0	0	1	1270	9	2008	0	8	20	22	578.0217
3	1	0	0	1	1270	9	2008	0	8	20	12	484.2093
6	1	0	0	0	1270	9	2008	0	9	20	5	574.8245
5	1	1	0	1	1270	9	2008	0	8	20	7	539.3734
5	1	1	0	1	1270	9	2008	0	8	20	21	540.3201
6	1	0	0	0	1270	9	2008	0	9	20	12	555.4291

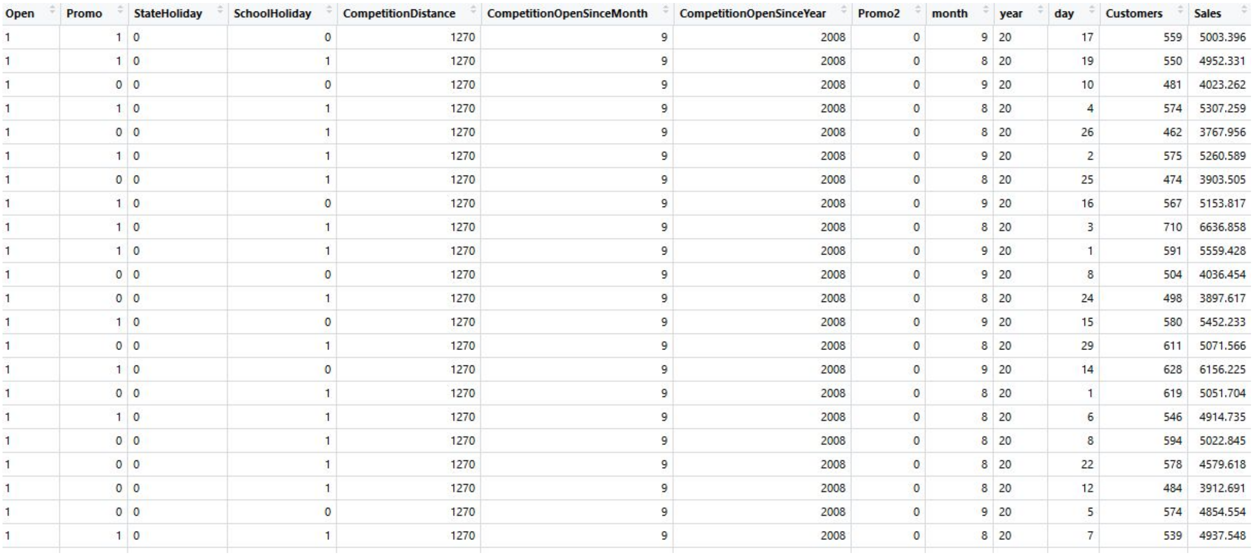
Training the model to predict sales on the training dataset: We will be training the model to predict the sales value using the randomForest library, and we can see that we have very small value of mean square error and variance is small, reducing overfitting and having most accurate results.



```
     |      Out-of-bag   |
Tree |        MSE  %Var(y) |
   1 |   0.02454    13.17 |
   2 |   0.01875    10.06 |
   3 |   0.01572     8.44 |
   4 |    0.0139     7.46 |
   5 |   0.01313     7.05 |
   6 |   0.01246     6.69 |
   7 |   0.01206     6.47 |
   8 |   0.01168     6.27 |
   9 |   0.01145     6.15 |
  10 |   0.01125     6.04 |
  11 |   0.01111     5.96 |
  12 |   0.01096     5.88 |
  13 |    0.0108     5.80 |
  14 |   0.01068     5.73 |
  15 |    0.0106     5.69 |
  16 |   0.01051     5.64 |
  17 |   0.01045     5.61 |
  18 |   0.01039     5.57 |
  19 |   0.01033     5.55 |
  20 |   0.01027     5.51 |
```

MSE: 0.01027 Variance: 5.51%

Predicted Values obtained of the Sales



Open	Promo	StateHoliday	SchoolHoliday	CompetitionDistance	CompetitionOpenSinceMonth	CompetitionOpenSinceYear	Promo2	month	year	day	Customers	Sales
1	1	0	0	1270	9	2008	0	9	20	17	559	5003.396
1	1	0	1	1270	9	2008	0	8	20	19	550	4952.331
1	0	0	0	1270	9	2008	0	9	20	10	481	4023.262
1	1	0	1	1270	9	2008	0	8	20	4	574	5307.259
1	0	0	1	1270	9	2008	0	8	20	26	462	3767.956
1	1	0	1	1270	9	2008	0	9	20	2	575	5260.589
1	0	0	1	1270	9	2008	0	8	20	25	474	3903.505
1	1	0	0	1270	9	2008	0	9	20	16	567	5153.817
1	1	0	1	1270	9	2008	0	8	20	3	710	6636.858
1	1	0	1	1270	9	2008	0	9	20	1	591	5559.428
1	0	0	0	1270	9	2008	0	9	20	8	504	4036.454
1	0	0	1	1270	9	2008	0	8	20	24	498	3897.617
1	1	0	0	1270	9	2008	0	9	20	15	580	5452.233
1	0	0	1	1270	9	2008	0	8	20	29	611	5071.566
1	1	0	0	1270	9	2008	0	9	20	14	628	6156.225
1	0	0	1	1270	9	2008	0	8	20	1	619	5051.704
1	1	0	1	1270	9	2008	0	8	20	6	546	4914.735
1	0	0	1	1270	9	2008	0	8	20	8	594	5022.845
1	0	0	1	1270	9	2008	0	8	20	22	578	4579.618
1	0	0	1	1270	9	2008	0	8	20	12	484	3912.691
1	0	0	0	1270	9	2008	0	9	20	5	574	4854.554
1	1	0	1	1270	9	2008	0	8	20	7	539	4937.548

After training the model, we will predict the Sales, and on observing above predicted sales table, we can see that we are getting better results, which shows Random Forest gives better results and it predicts most accurate sales value on the Open day of Stores. The prediction performance of the model can be assessed using MSE of 0.01027 and variance of 5.51%, which is better than the performance for linear regression.

On one side, it has been noticed that the sales are lowest for 10-15th and 25-27th days of 5th and 10th months for stores not running Promos and having nearby competitors, who is opened more than 4 years. Besides, the strong dependence of Sales on Customers has been observed. On the other side, it can also be seen that the sales are absolutely highest at the end of December for the stores running Promos and don’t have nearby competitors. It can be clearly observed that a high number of Customers (highest in December) gives high Sales.

Very often, the algorithm design can be improved. One possible direction is to try for different hyper parameter values, which directly affects the training of the model. For instance, we can increase the value of the hyper parameters like mtry value (variables randomly sampled) and ntree (deeper tree to explore every sample at least few times) and trying to reduce maxnodes to avoid tree growing to deep and in wrong direction. These different values of hyper parameters could be promising factors to increase the accuracy of the chosen algorithm for future modeling.

CONCLUSION

In this project, the analysis of Rossman Store Data has been performed following CRISP-DM framework. The major objective formulated at the research understanding phase of this project was to predict the Sales using all relevant predictors. This objective has been achieved by initial application of data preprocessing and EDA techniques. After that, the prediction of Customers variable has been performed due to its absence in Test dataset. Random Forest has been utilized as the main modeling technique, which has given the best results for the defined project objective. The performance of Random Forest algorithm has been evaluated and confirmed the validity of the developed model. The possible performance improvements has been discussed as well.

REFERENCES

1. Discovering Knowledge in Data: An Introduction to Data Mining by Larose and Larose. (2nd edition) ISBN: 978-0-470-90874-7
2. https://www.techopedia.com/definition/14650/data-preprocessing
3. https://www.kaggle.com/anshumanyp/rossman/data
4. https://rpubs.com/gpetho/142772
5. http://www.rpubs.com/mordoobadi/Rossmann-Store-Sales
6. Data Mining for Business Analytics. (3rd edition) ISBN: 978-1-118-72927-4
7. https://www.rdocumentation.org/packages/randomForest/versions/4.6-14/topics/randomForest