MKT 591 - Report

Team C-02: Siddharth Srivastava, Bhargava Sukkala, Vikram Vegesna, Tejas Mehta

Contents

1 Executive Summary **2**

2 Problem Description **3**

2.1 Objective 3

3 Data **4**

3.1 Description 4

3.2 Preprocesing 4

3.3 Summary Statistics 5

4 Models **7**

4.1 RandomForest 7

4.2 Linear Regression 9

5 Results **13**

5.1 RandomForest 13

5.2 Linear Regression 13

5.3 RandomForest vs Linear Regression 13

6 Managerial Implications **14**

7 Appendix I - R code **15**

8 Appendix II - R Output **18**

1 Executive Summary

The marketing problem chosen was a sales prediciton problem for a pharmacy. The objective of the problem was restated and a lateral approach was adopted to observe *Sales* and *Customer* behavior as a function of *Competitor's parameters* and *Holidays*, individually.

The data was dealt with at an aggregate level, eliminating the inclusion of time dimension so as to study the general impact of *Competition* and *Holidays* on *Store performance*.

The *Predictive Analytics* method used here was *Regression*. Several models were built using two very powerful *Machine Learning* algorithms: *Random Forests* and *Linear Regression*.

The models were evaluated based on accuracy of prediction given by *Root Mean Squared Error*. Apart from the *RMSE* value, models were evaulated based on the level of insights provided by them. *Linear Regression* models were evaluated based on the *R-squared* value while *RandomForest* models were evaulated based on *Predictor Importance* scores.

Insights from both type of models were considered for final analysis. Analysis results and Recommendations were provided based on findings from studying both the models back and forth and combining their results.

2 Problem Description

The problem chosen was Rossmann Store Sales, a sales prediction problem posted on **kaggle**. Rossmann Pharmacy wanted to forecast sales for it's 1115 stores using historical data about *Sales, Customers, Holidays, Stores, Promotion and Competition.*

After observing the data, we realized that using *Competition, Sales* and *Holiday* data to determine consumers would be a better marketing problem rather than an ordinary sales forecast. So we disregarded time information to study the general impact of *Competition, Promotion and Holidays* on *Sales and Customers.* We also performed validation on the train set and ignored the test set as a consequence of the new objective.

2.1 Objective

The objective was to use Competition, Promotion and Holiday information to forecast footfalls and sales for Rossmann Pharmacy at an aggrate year level and suggest any areas for improvement based on our models.

3 Data

3.1 Description

We were provided with 2 datasets:

1. Train: Comprised of daily Sales

- `Store` - Unique id for each store
- `DayofWeek` - Day of Week such as Monday, Tuesday, etc.
- `Date` - Date of Sales.
- `Sales` - the turnover for any given day
- `Customers` - Number of Customers
- `Open` - 0: Store closed on that day, 1: Store open on that day
- `Promo` - Indicates if a store was running a promotion on that day. 0 or 1
- `StateHoliday` - 0: No Holiday, a: Public Holiday, b: Easter Holiday, c: Christmas Holiday
- `SchoolHoliday` - Indicates if a store was impacted by closure of public schools. 0 or 1.

2. Store: Attribute for each store

- `Store` - Unique id for each store
- `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
- `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

3.2 Preprocesing

3.2.1 For Competition Analysis

We removed `Date`, `Promo`, `StateHoliday`, `SchoolHoliday` and `DayofWeek` attributes in the **Train** dataset, and *aggregated* `Sales` and `Customers` for each 1115 individual `Store`

The **Store** dataset consists of attributes for each 1115 stores. Since we decided to disregard time related information for our analysis, we removed `Promo2Since[Year/Week]` and `PromoIntervalfrom` **Store** dataset. We also converted the `CompetitionOpenSince[Month/Year]` variable to a continuous `Comp_Since` age (*in number of days*) attribute by subtracting the open date of the Competitor from the *Current Date*.

After making the specified changes on the **Train** and **Store** datasets, we merged the two datasets to get `Customer` and `Sales` for each store with `Comp_Since` and `CompetitionDistance` along with other `Store` related attribtues.

Lastly, we replaced missing values for `CompetitionDistance` and `Comp_since` with 0, implying no competition.

Split the **Merged** dataset into train and validation sets with a 75:25 split.

3.2.2 For Holiday and Promotion Analyis

We aggregated `Sales` and `Customer` for all combinations of `Store`, `StateHoliday`, `SchoolHoliday`, and `Promo` to study impact of holiday on sales and customers.

Split the aggregated **Train** dataset into train and validation sets with a 75:25 split.

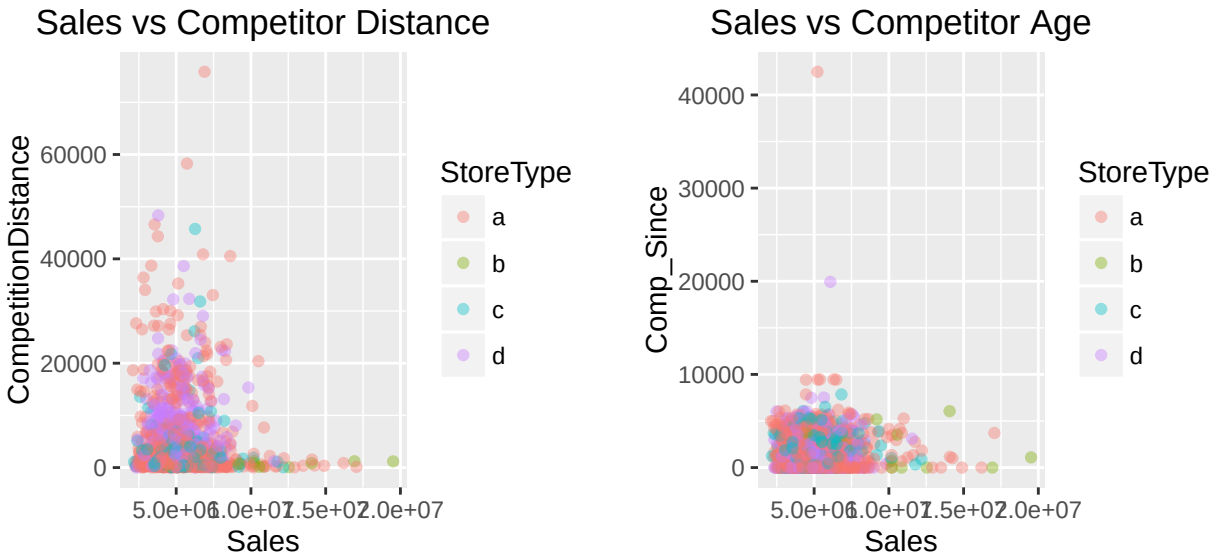
3.3 Summary Statistics

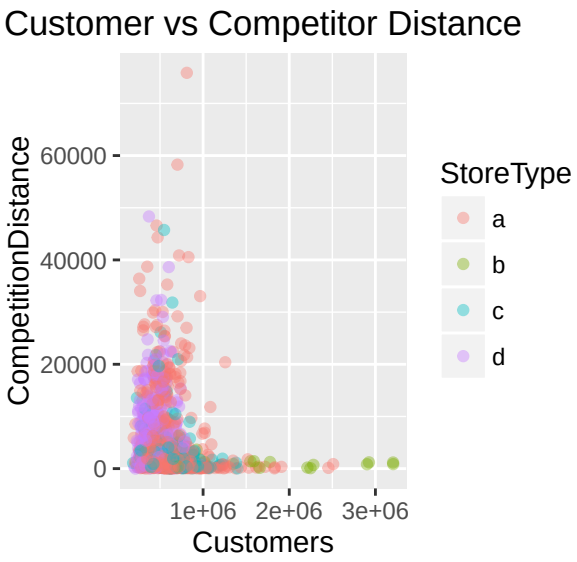
3.3.1 Merged dataset

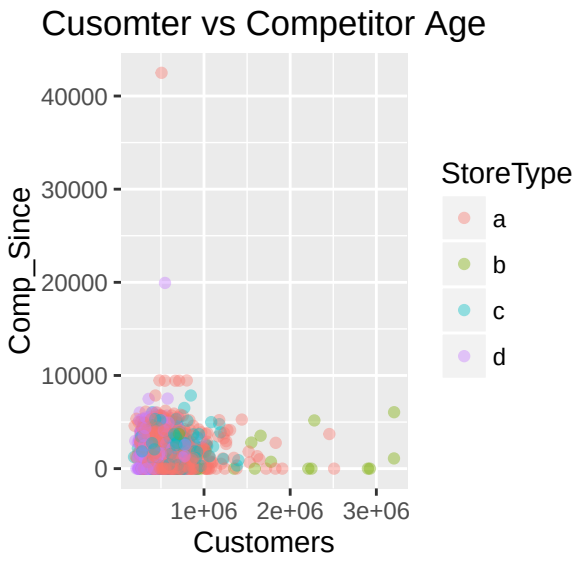
```
summary(store_comp)
```

```
##      Store          Customers          Sales          StoreType
##  Min.   :   1.0   Min.   : 187583   Min.   : 2114322   a:602
##  1st Qu.: 279.5   1st Qu.: 405391   1st Qu.: 3949377   b: 17
##  Median : 558.0   Median : 509233   Median : 4990259   c:148
##  Mean   : 558.0   Mean   : 577616   Mean   : 5267427   d:348
##  3rd Qu.: 836.5   3rd Qu.: 671544   3rd Qu.: 6084148
##  Max.   :1115.0   Max.   :3206058   Max.   :19516842
##  Assortment CompetitionDistance   Comp_Since    Promo2
##  a:593      Min.   :    0       Min.   :    0   0:544
##  b:  9      1st Qu.:  710       1st Qu.:    0   1:571
##  c:513      Median : 2320       Median : 1278
##             Mean   : 5390       Mean   : 1782
##             3rd Qu.: 6875       3rd Qu.: 2984
##             Max.   :75860       Max.   :42490
```

Plots



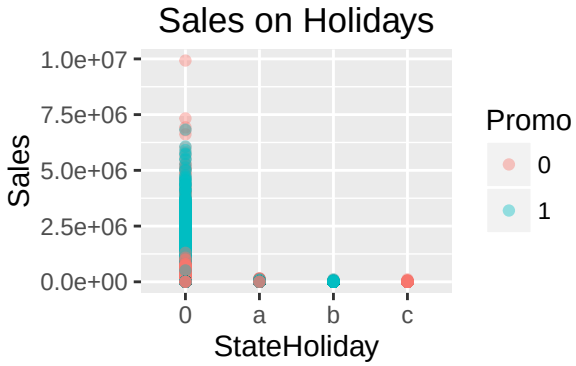


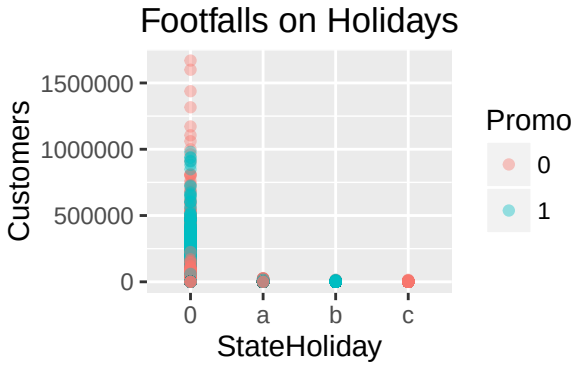


3.3.2 Train dataset

```
summary(train_hol)
```

```
##      Store          StateHoliday SchoolHoliday Promo    Open
##  Min.   :   1.0   0:5929       0:6008        0:8036   0:8553
##  1st Qu.: 279.0   a:3930       1:7293        1:5265   1:4748
##  Median : 557.0   b:2324
##  Mean   : 557.5   c:1118
##  3rd Qu.: 837.0
##  Max.   :1115.0
##      Sales            Customers
##  Min.   :      0   Min.   :      0
##  1st Qu.:      0   1st Qu.:      0
##  Median :      0   Median :      0
##  Mean   : 441559   Mean   :  48421
##  3rd Qu.: 497817   3rd Qu.:  51395
##  Max.   :9925575   Max.   :1669048
```





Plots

4 Models

4.1 RandomForest

The motivation behind using RandomForest was to use decision trees to perform regression as we have majority *Categorical* variables.

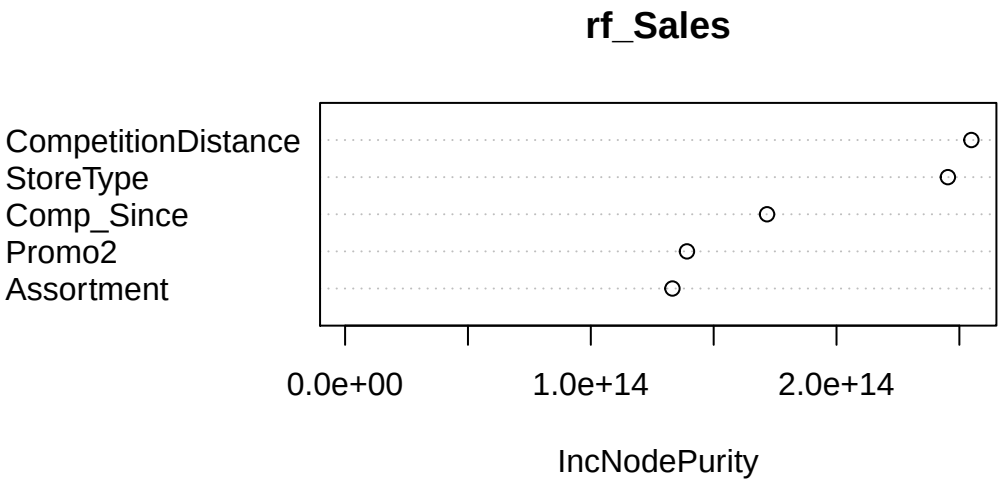
However, using *Classification and Regression Tree* can be misleading and lead to overfitting owing to the small size of the dataset. Hence, we decided to use *RandomForest* for regression.

Measure of accuracy was *Root Mean Squared Error* between the predicted result and the validation set.

4.1.1 Competition Analysis

We created a *RandomForest* model for `Sales` by using the predictors `CompetitionDistance`, `Comp_Since`, `Assortment`, `StoreType` and `Promo2` with 500 trees.

```
rf_Sales <- randomForest(Sales ~ CompetitionDistance + Comp_Since + StoreType +
    Assortment + Promo2, data = st_comp, ntree = 500)
```

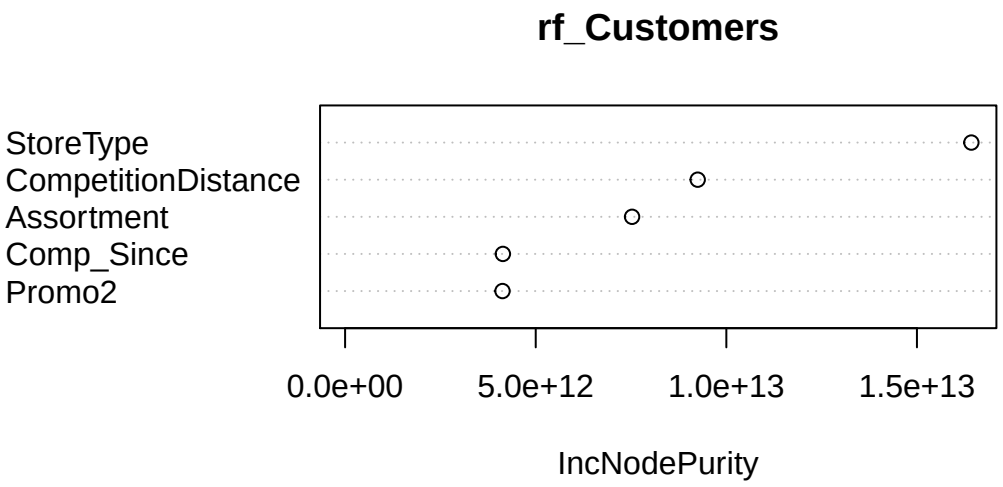


The rmse of the model is:

```
## [1] 1756441
```

We created another *RandomForest* model for `Customers` by using the predictors `CompetitionDistance`, `Comp_Since`, `Assortment`, `StoreType` and `Promo2` with 500 trees.

```
rf_Customers <- randomForest(Customers ~ CompetitionDistance + Comp_Since +
    StoreType + Assortment + Promo2, data = st_comp, ntree = 500)
```



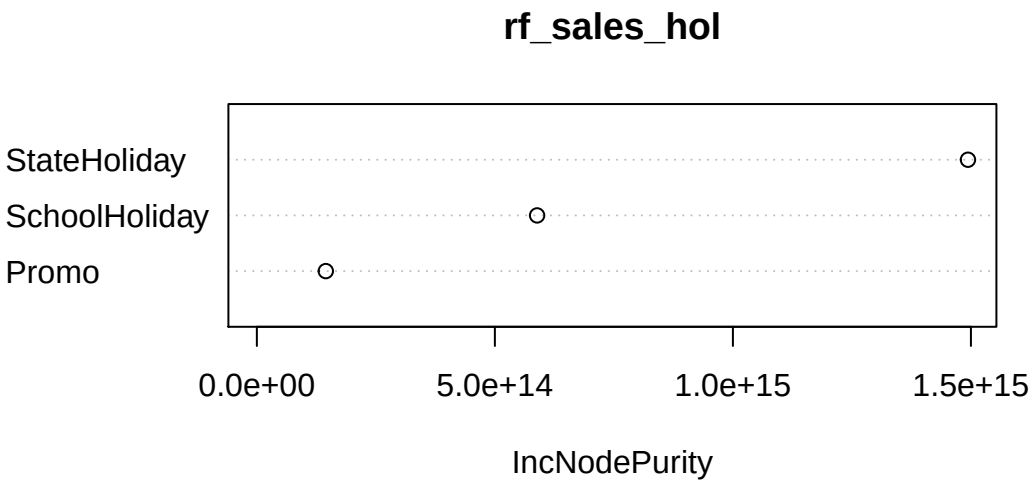
The rmse of the model is:

```
## [1] 227620.9
```

4.1.2 Holiday and Promotions Analysis

We created a *RandomForest* model for `Sales` by using the predictors `StaeHoliday`, `SchoolHoliday`, and `Promo` with 500 trees.

```
rf_sales_hol <- randomForest(Sales ~ StateHoliday + SchoolHoliday + Promo, data = tr_hol,
    ntree = 500)
```

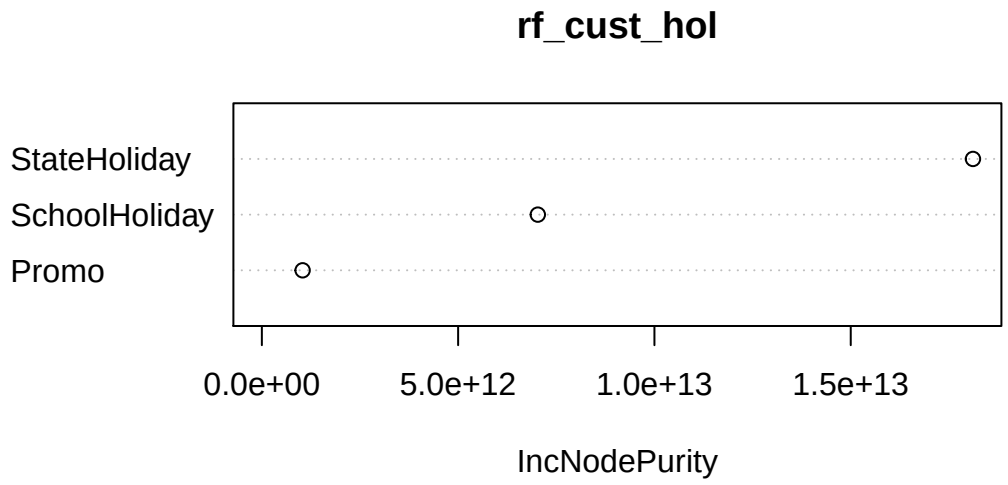


The rmse of the model is:

```
## [1] 611744.1
```

We created a *RandomForest* model for `Customers` by using the predictors `StaeHoliday`, `SchoolHoliday`, and `Promo` with 500 trees.

```
rf_cust_hol <- randomForest(Customers ~ StateHoliday + SchoolHoliday + Promo,
    data = tr_hol, ntree = 500)
```



The rmse of the model is:

```
## [1] 78840.76
```

4.2 Linear Regression

The motivation behind using *Linear Regression* model was the concept of parsimony. Owing to less but meaningful attributes and normalized data, linear regression was a very powerful model that could explain the relationship between the dependent variables, `Sales` and `Customers`, with the independent variables in both the data sets.

Measure of accuracy was *Root Mean Squared Error* between the predicted result and the validation set.

4.2.1 Competition Analysis

We created a *Linear Regression* model for `Sales` by using the predictors `CompetitionDistance`, `Comp_Since`, `Assortment`, `StoreType` and `Promo2`.

```
lm_Sales <- lm(Sales ~ CompetitionDistance + Comp_Since + StoreType + Assortment +
    Promo2, data = st_comp)
```

```
summary(lm_Sales)
```

```
##
## Call:
## lm(formula = Sales ~ CompetitionDistance + Comp_Since + StoreType +
##     Assortment + Promo2, data = st_comp)
```

```
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -8210103 -1161865  -242150   882889 11765395
##
## Coefficients:
##                       Estimate Std. Error t value Pr(>|t|)
## (Intercept)          5.352e+06  1.419e+05  37.719  < 2e-16 ***
## CompetitionDistance -1.360e+01  8.754e+00  -1.554 0.120585
## Comp_Since          -1.561e+01  3.626e+01  -0.431 0.666853
## StoreTypeb           7.406e+06  8.110e+05   9.132  < 2e-16 ***
## StoreTypec           1.135e+05  1.909e+05   0.594 0.552341
## StoreTyped          -2.505e+05  1.467e+05  -1.708 0.088084 .
## Assortmentb         -3.697e+06  1.054e+06  -3.508 0.000477 ***
## Assortmentc          8.280e+05  1.322e+05   6.263 6.08e-10 ***
## Promo21             -8.071e+05  1.272e+05  -6.347 3.61e-10 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1798000 on 827 degrees of freedom
## Multiple R-squared:  0.1927, Adjusted R-squared:  0.1849
## F-statistic: 24.67 on 8 and 827 DF,  p-value: < 2.2e-16
```

The rmse of the model is:

```
## [1] 1833351
```

We created another *Linear Regression* model for `Customers` by using the predictors `CompetitionDistance`, `Comp_Since`, `Assortment`, `StoreType` and `Promo2`.

```
lm_Cust <- lm(Customers ~ CompetitionDistance + Comp_Since + StoreType + Assortment +
    Promo2, data = st_comp)
```

```
summary(lm_Cust)
```

```
##
## Call:
## lm(formula = Customers ~ CompetitionDistance + Comp_Since + StoreType +
##     Assortment + Promo2, data = st_comp)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -1470554  -134145   -31998    98569  1804770
##
## Coefficients:
##                       Estimate Std. Error t value Pr(>|t|)
## (Intercept)          6.600e+05  1.854e+04  35.598  < 2e-16 ***
## CompetitionDistance -5.192e+00  1.144e+00  -4.539 6.49e-06 ***
## Comp_Since          -2.610e+00  4.738e+00  -0.551    0.582
## StoreTypeb           1.532e+06  1.060e+05  14.454  < 2e-16 ***
## StoreTypec           2.068e+04  2.494e+04   0.829    0.407
## StoreTyped          -1.390e+05  1.917e+04  -7.252 9.47e-13 ***
```

```
## Assortmentb         -5.987e+04  1.377e+05  -0.435    0.664
## Assortmentc          7.319e+04  1.727e+04   4.237 2.52e-05 ***
## Promo21             -1.333e+05  1.662e+04  -8.020 3.60e-15 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 235000 on 827 degrees of freedom
## Multiple R-squared:  0.4523, Adjusted R-squared:  0.447
## F-statistic: 85.38 on 8 and 827 DF,  p-value: < 2.2e-16
```

The rmse of the model is:

```
## [1] 258738.1
```

4.2.2 Holiday and Promotions Analysis

We created a *Linear Regression* model for `Sales` by using the predictors `StaeHoliday`, `SchoolHoliday`, and `Promo`.

```
lm_sales_hol <- lm(Sales ~ StateHoliday + SchoolHoliday + Promo, data = tr_hol)
```

```
summary(lm_sales_hol)
```

```
##
## Call:
## lm(formula = Sales ~ StateHoliday + SchoolHoliday + Promo, data = tr_hol)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -1340768  -336097   -73891   174372  8833069
##
## Coefficients:
##                Estimate Std. Error t value Pr(>|t|)
## (Intercept)     1092506      12796   85.38   <2e-16 ***
## StateHolidaya  -1004672      15311  -65.62   <2e-16 ***
## StateHolidayb   -781315      19809  -39.44   <2e-16 ***
## StateHolidayc   -606463      26439  -22.94   <2e-16 ***
## SchoolHoliday1  -485563      14680  -33.08   <2e-16 ***
## Promo1           248263      13682   18.14   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 646100 on 9969 degrees of freedom
## Multiple R-squared:  0.4211, Adjusted R-squared:  0.4208
## F-statistic:  1450 on 5 and 9969 DF,  p-value: < 2.2e-16
```

The rmse of the model is:

```
## [1] 614354
```

We created a *Linear Regression* model for `Customers` by using the predictors `StaeHoliday`, `SchoolHoliday`, and `Promo`.

```
lm_cust_hol <- lm(Customers ~ StateHoliday + SchoolHoliday + Promo, data = tr_hol)
```

```
##
## Call:
## lm(formula = Customers ~ StateHoliday + SchoolHoliday + Promo,
##     data = tr_hol)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -142617  -32881   -4608   20782 1546280
##
## Coefficients:
##                Estimate Std. Error t value Pr(>|t|)
## (Intercept)      122768       1583   77.54   <2e-16 ***
## StateHolidaya   -109736       1894  -57.92   <2e-16 ***
## StateHolidayb    -84347       2451  -34.41   <2e-16 ***
## StateHolidayc    -69032       3272  -21.10   <2e-16 ***
## SchoolHoliday1   -53663       1816  -29.54   <2e-16 ***
## Promo1            19849       1693   11.72   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 79950 on 9969 degrees of freedom
## Multiple R-squared:  0.358,  Adjusted R-squared:  0.3577
## F-statistic:  1112 on 5 and 9969 DF,  p-value: < 2.2e-16
```

The rmse of the model is:

```
## [1] 78783.14
```

5 Results

5.1 RandomForest

5.1.1 Competition Analysis

For `Sales`, *RandomForest* predicts `CompetitionDistance` to be of the highest importance followed by `StoreType` and `Comp_Since`. This implies that `Sales` are influenced by the distance of the competitor's store and it's age.

For `Cusomers`, *RandomForest* predicts `StoreType` to be of the highest importance followed by `CompetitionDistance` and `Assortment`. This implies that Footfalls are influenced by the type of store, it's distance from a competitor store and it's assortment type.

However, based on the rmse values for both the models, the prediction is not significant for `Sales` as much as it is for `Customers`, implying `Competition` better explains footfalls than revenue.

5.1.2 Holiday and Promotion Analysis

RandomForest predicts `StateHoliday` to be of the highest importance followed by `SchoolHoliday` and `Promo` for both `Sales` and `Customers`.

This implies that the nature of the holiday largely impacts sales, but *RandomForest* doesn't provide insights into the holiday types even though the *RMSE* values are low and significant.

5.2 Linear Regression

5.2.1 Competition Analysis

For `Sales`, *Linear Regression* model doesn't consider *Competition* to be of great importance. Rather, it suggests that `Assortment` levels b and c, `StoreType` b and `Promo2`, promotions largely influence sales. The *R-squared* value as well as *RMSE* is low implying the model is insignificant

For `Customers`, *Linear Regression* model considers `CompetitionDistance` to be of great importance along with `Assortment` levels c, `StoreType` b and d, and `Promo2`, promotions largely influence sales.The *R-squared* value as well as *RMSE* is decent enough to be consideredd significant.

Linear Regression model too suggests that *Competition* better explains *Footfalls* than *Sales*

5.2.2 Holiday and Promotion Analysis

Linear Regression model considers `StateHoliday`, `SchoolHoliday` and `Promo` as highly significant for both `Sales` and `Customers`.

This implies that the nature of the holiday largely impacts *Sales* and *Footfalls*. The *R-squared* values and *RMSE* values suggest the model to be significant.

5.3 RandomForest vs Linear Regression

Both *RandomForest* and *Linear Regression* models for all objectives have similar *RMSE* values of prediction. However, both present different insights to the problem and it is wiser to consider results of both models to develop insights.

6 Managerial Implications

The analysis of both *RandomForest* and *Linear Regression* models suggests that *Competition* affects the *Number of Customers*. The *Regression co-efficient* for `CompetitionDistance` is negative and significant implying that as the distance of the competitor's store increases, the number of customers entering a *Rossmann* store decreases. Also `Customers` are more likely to visit a larger `Assortment` which is significant for `Sales` too.

Also, Holidays largely impact the *Turnover* and *Footfalls*. The co-efficients for all holiday types are negative and significant, implying that *Rossmann's* Sales drop down during holidays.

Our suggestions to the manager would be:

1. To open more store at an optimal distance from each other to increase market accessibility and presence.
2. Customers are likely to go for larger assortments and the store type b. Hence, Rossmann should consider this information if and when it plans to expand.
3. Holidays cause loss to Rossmann stores. It could be because the stores are closed during holidays. Rossmann should open 24/7 express stores as it is a pharmacy. Opening such stores would increase it's sales through smaller assortments and account for the losses incurred during the holidays.

7 Appendix I - R code

```
library(data.table)
library(dplyr)
library(randomForest)
library(stringr)
library(tidyr)
library(ggplot2)
library(ggthemes)
library(psych)
library(corrplot)
set.seed(1501)

########################################## Data Loading ##########################################

train <- fread("train.csv", stringsAsFactors = T)
store <- fread("store.csv", stringsAsFactors = T)

str(train)
str(store)
summary(train)
summary(store)

########################################## Data Cleaning ##########################################

train$Open <- as.factor(train$Open)
train$DayOfWeek <- as.factor(train$DayOfWeek)
train$Promo <- as.factor(train$Promo)
train$Date <- as.Date(train$Date)
train$DayOfWeek <- NULL
train$Date <- NULL

store$CompetitionOpenSinceMonth <- str_pad(store$CompetitionOpenSinceMonth,
    width = 2, side = "left", pad = "0")
store <- unite(store, Comp_Since, CompetitionOpenSinceMonth, CompetitionOpenSinceYear,
    sep = "-")
store$Comp_Since <- as.yearmon(store$Comp_Since, format = "%m-%Y")
store$Comp_Since <- as.Date.yearmon(store$Comp_Since)
store$Comp_Since <- today() - store$Comp_Since
store$Comp_Since <- as.integer(store$Comp_Since)
store$Promo2SinceWeek <- NULL
store$PromoInterval <- NULL
store$Promo2SinceYear <- NULL
store$Promo2 <- as.factor(store$Promo2)

# Create different train sets.

# 1st Train set for Sales ~ Competition

train_comp <- train %>% select(Store, Customers, Sales)
train_comp <- train_comp %>% group_by(Store) %>% summarise_each(funs(sum))

store_comp <- train_comp %>% inner_join(store, by = "Store")
```

```
summary(store_comp)
store_comp[is.na(store_comp), ] <- 0

index <- sample(nrow(store_comp), nrow(store_comp) * 0.75)
st_comp <- store_comp[index, ]
st_compv <- store_comp[-index, ]

# Sales and Competition Models
lm_Sales <- lm(Sales ~ CompetitionDistance + Comp_Since + StoreType + Assortment +
    factor(Promo2), data = st_comp)
summary(lm_Sales)
rf_Sales <- randomForest(Sales ~ CompetitionDistance + Comp_Since + StoreType +
    Assortment + Promo2, data = st_comp, ntree = 500)
rf_Sales$importance
varImpPlot(rf_Sales)

pred1 <- predict(lm_Sales, st_compv)
pred2 <- predict(rf_Sales, st_compv)
rmse(pred2, st_compv$Sales)
rmse(pred1, st_compv$Sales)

lm_Cust <- lm(Customers ~ CompetitionDistance + Comp_Since + StoreType + Assortment +
    factor(Promo2), data = st_comp)
summary(lm_Cust)
rf_Customers <- randomForest(Customers ~ CompetitionDistance + Comp_Since +
    StoreType + Assortment + Promo2, data = st_comp)
rf_Customers$importance

pred3 <- predict(lm_Cust, st_compv)
pred4 <- predict(rf_Customers, st_compv)
rmse(pred3, st_compv$Customers)
rmse(pred4, st_compv$Customers)


# 2nd Train set for Holiday Analysis

train_hol <- train %>% group_by(Store, StateHoliday, SchoolHoliday, Promo, Open) %>%
    summarise_each(funs(sum))

index <- sample(nrow(train_hol), nrow(train_hol) * 0.75)
tr_hol <- train_hol[index, ]
tr_holv <- train_hol[-index, ]

lm_sales_hol <- lm(Sales ~ StateHoliday + SchoolHoliday + Promo + Open, data = tr_hol)
summary(lm_sales_hol)
pred5 <- predict(lm_sales_hol, tr_holv)
rmse(pred5, tr_holv$Sales)

lm_cust_hol <- lm(Customers ~ StateHoliday + SchoolHoliday + Promo + Open, data = tr_hol)
summary(lm_cust_hol)
pred6 <- predict(lm_cust_hol, tr_holv)
rmse(pred6, tr_holv$Customers)
```

```
rf_sales_hol <- randomForest(Sales ~ StateHoliday + SchoolHoliday + Promo, data = tr_hol,
    ntree = 500)
pred7 <- predict(rf_sales_hol, tr_holv)
rmse(pred7, tr_holv$Sales)

rf_cust_hol <- randomForest(Sales ~ StateHoliday + SchoolHoliday + Promo, data = tr_hol)
varImpPlot(rf_cust_hol)
pred8 <- predict(rf_cust_hol, tr_holv)
rmse(pred8, tr_holv$Customers)
```

8 Appendix II - R Output

```
## Classes 'data.table' and '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         : Factor w/ 942 levels "2013-01-01","2013-01-02",..: 942 942 942 942 942 942 942 942
##  $ 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 : Factor w/ 4 levels "0","a","b","c": 1 1 1 1 1 1 1 1 1 1 ...
##  $ SchoolHoliday: Factor w/ 2 levels "0","1": 2 2 2 2 2 2 2 2 2 2 ...
##  - attr(*, ".internal.selfref")=<externalptr>
```

```
## Classes 'data.table' and 'data.frame':   1115 obs. of  10 variables:
##  $ Store                    : int  1 2 3 4 5 6 7 8 9 10 ...
##  $ StoreType                : Factor w/ 4 levels "a","b","c","d": 3 1 1 3 1 1 1 1 1 1 ...
##  $ Assortment               : Factor w/ 3 levels "a","b","c": 1 1 1 3 1 1 3 1 3 1 ...
##  $ 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            : Factor w/ 4 levels "","Feb,May,Aug,Nov",..: 1 3 3 1 1 1 1 1 1 1 ...
##  - attr(*, ".internal.selfref")=<externalptr>
```

```
##      Store          DayOfWeek            Date             Sales
##  Min.   :   1.0   Min.   :1.000   2013-01-02:   1115   Min.   :    0
##  1st Qu.: 280.0   1st Qu.:2.000   2013-01-03:   1115   1st Qu.: 3727
##  Median : 558.0   Median :4.000   2013-01-04:   1115   Median : 5744
##  Mean   : 558.4   Mean   :3.998   2013-01-05:   1115   Mean   : 5774
##  3rd Qu.: 838.0   3rd Qu.:6.000   2013-01-06:   1115   3rd Qu.: 7856
##  Max.   :1115.0   Max.   :7.000   2013-01-07:   1115   Max.   :41551
##                                   (Other)   :1010519
##    Customers           Open            Promo        StateHoliday
##  Min.   :   0.0   Min.   :0.0000   Min.   :0.0000   0:986159
##  1st Qu.: 405.0   1st Qu.:1.0000   1st Qu.:0.0000   a: 20260
##  Median : 609.0   Median :1.0000   Median :0.0000   b:  6690
##  Mean   : 633.1   Mean   :0.8301   Mean   :0.3815   c:  4100
##  3rd Qu.: 837.0   3rd Qu.:1.0000   3rd Qu.:1.0000
##  Max.   :7388.0   Max.   :1.0000   Max.   :1.0000
##
##  SchoolHoliday
##  0:835488
##  1:181721
##
##
##
##
##
```

```
##      Store        StoreType Assortment CompetitionDistance
```

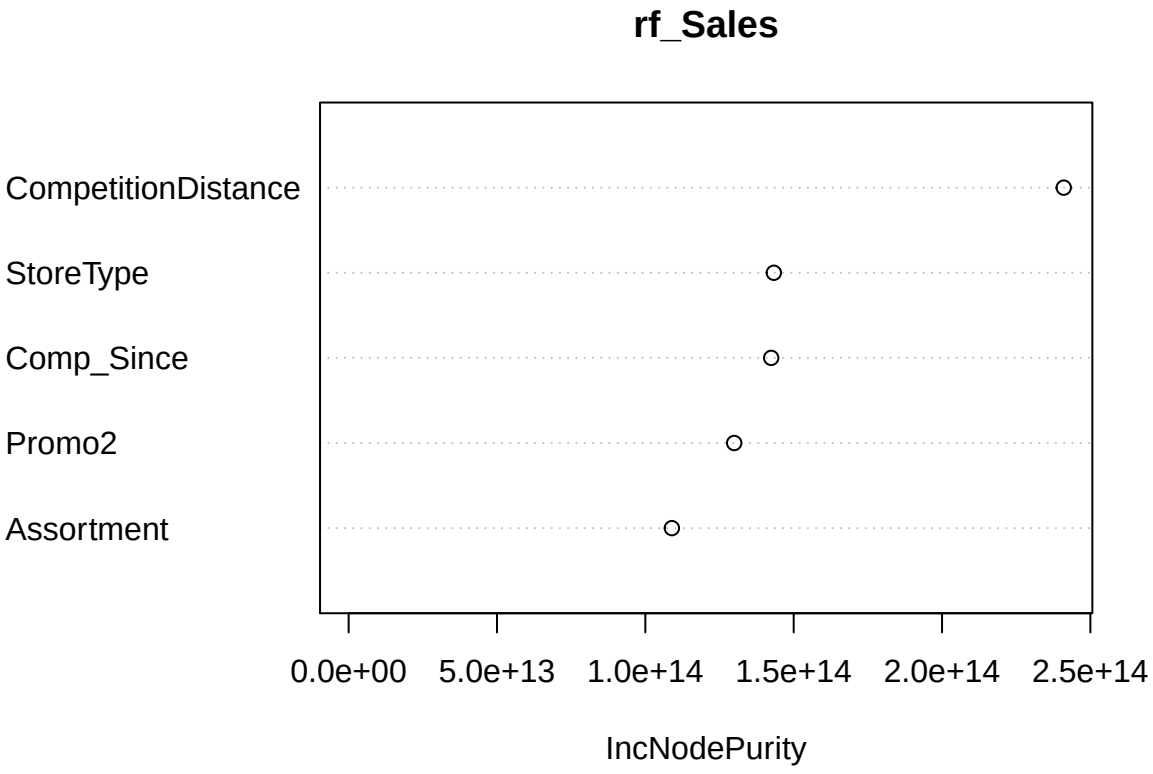
```
##  Min.   :   1.0   a:602     a:593      Min.   :   20.0
##  1st Qu.: 279.5   b: 17     b:  9      1st Qu.:  717.5
##  Median : 558.0   c:148     c:513      Median : 2325.0
##  Mean   : 558.0   d:348                Mean   : 5404.9
##  3rd Qu.: 836.5                        3rd Qu.: 6882.5
##  Max.   :1115.0                        Max.   :75860.0
##                                        NA's   :3
##  CompetitionOpenSinceMonth CompetitionOpenSinceYear     Promo2
##  Min.   : 1.000            Min.   :1900             Min.   :0.0000
##  1st Qu.: 4.000            1st Qu.:2006             1st Qu.:0.0000
##  Median : 8.000            Median :2010             Median :1.0000
##  Mean   : 7.225            Mean   :2009             Mean   :0.5121
##  3rd Qu.:10.000            3rd Qu.:2013             3rd Qu.:1.0000
##  Max.   :12.000            Max.   :2015             Max.   :1.0000
##  NA's   :354               NA's   :354
##  Promo2SinceWeek Promo2SinceYear          PromoInterval
##  Min.   : 1.0    Min.   :2009                    :544
##  1st Qu.:13.0    1st Qu.:2011    Feb,May,Aug,Nov :130
##  Median :22.0    Median :2012    Jan,Apr,Jul,Oct :335
##  Mean   :23.6    Mean   :2012    Mar,Jun,Sept,Dec:106
##  3rd Qu.:37.0    3rd Qu.:2013
##  Max.   :50.0    Max.   :2015
##  NA's   :544     NA's   :544

##      Store          Customers           Sales          StoreType
##  Min.   :   1.0   Min.   : 187583   Min.   : 2114322   a:602
##  1st Qu.: 279.5   1st Qu.: 405391   1st Qu.: 3949377   b: 17
##  Median : 558.0   Median : 509233   Median : 4990259   c:148
##  Mean   : 558.0   Mean   : 577616   Mean   : 5267427   d:348
##  3rd Qu.: 836.5   3rd Qu.: 671544   3rd Qu.: 6084148
##  Max.   :1115.0   Max.   :3206058   Max.   :19516842
##
##  Assortment CompetitionDistance   Comp_Since    Promo2
##  a:593      Min.   :   20.0     Min.   :  275   0:544
##  b:  9      1st Qu.:  717.5     1st Qu.: 1186   1:571
##  c:513      Median : 2325.0     Median : 2282
##             Mean   : 5404.9     Mean   : 2611
##             3rd Qu.: 6882.5     3rd Qu.: 3684
##             Max.   :75860.0     Max.   :42490
##             NA's   :3           NA's   :354

##
## Call:
## lm(formula = Sales ~ CompetitionDistance + Comp_Since + StoreType +
##     Assortment + factor(Promo2), data = st_comp)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -3892776 -1200363  -260809   907051 11585983
##
## Coefficients:
##                          Estimate Std. Error t value Pr(>|t|)
## (Intercept)          5.463e+06  1.363e+05  40.086  < 2e-16 ***
## CompetitionDistance -1.754e+01  8.666e+00  -2.024  0.04331 *
```

```
## Comp_Since           3.148e+00  2.665e+01   0.118  0.90601
## StoreTypeb           5.308e+06  8.121e+05   6.536 1.10e-10 ***
## StoreTypec          -1.106e+05  1.989e+05  -0.556  0.57848
## StoreTyped          -2.808e+05  1.453e+05  -1.932  0.05370 .
## Assortmentb         -2.823e+06  1.028e+06  -2.746  0.00616 **
## Assortmentc          7.963e+05  1.318e+05   6.039 2.34e-09 ***
## factor(Promo2)1     -8.386e+05  1.273e+05  -6.588 7.94e-11 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1800000 on 827 degrees of freedom
## Multiple R-squared:  0.1448, Adjusted R-squared:  0.1365
## F-statistic:  17.5 on 8 and 827 DF,  p-value: < 2.2e-16
```

```
##                     IncNodePurity
## CompetitionDistance  2.410204e+14
## Comp_Since           1.424088e+14
## StoreType            1.433063e+14
## Assortment           1.089405e+14
## Promo2               1.299513e+14
```




```
## [1] 1821377
```

```
## [1] 1792177
```

```
##
## Call:
## lm(formula = Customers ~ CompetitionDistance + Comp_Since + StoreType +
##     Assortment + factor(Promo2), data = st_comp)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -840420 -141809  -35231   94999 1785215
##
## Coefficients:
##                       Estimate Std. Error t value Pr(>|t|)
## (Intercept)          6.740e+05  1.808e+04  37.291  < 2e-16 ***
## CompetitionDistance -5.340e+00  1.149e+00  -4.645 3.95e-06 ***
## Comp_Since          -1.116e+00  3.535e+00  -0.316    0.752
## StoreTypeb           1.130e+06  1.077e+05  10.493  < 2e-16 ***
## StoreTypec          -2.381e+02  2.638e+04  -0.009    0.993
## StoreTyped          -1.514e+05  1.928e+04  -7.853 1.26e-14 ***
## Assortmentb          1.187e+05  1.363e+05   0.871    0.384
## Assortmentc          7.348e+04  1.749e+04   4.202 2.94e-05 ***
## factor(Promo2)1     -1.380e+05  1.688e+04  -8.173 1.13e-15 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 238800 on 827 degrees of freedom
## Multiple R-squared:  0.3887, Adjusted R-squared:  0.3828
## F-statistic: 65.74 on 8 and 827 DF,  p-value: < 2.2e-16
```

```
##                     IncNodePurity
## CompetitionDistance  9.102555e+12
## Comp_Since           3.381724e+12
## StoreType            1.245054e+13
## Assortment           6.917395e+12
## Promo2               3.934281e+12
```

```
## [1] 237596.7
```

```
## [1] 237885
```

```
##
## Call:
## lm(formula = Sales ~ StateHoliday + SchoolHoliday + Promo + Open,
##     data = tr_hol)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -1697382  -338432    23960   150940  8369964
##
## Coefficients:
##                Estimate Std. Error t value Pr(>|t|)
## (Intercept)      338433      15076  22.448  < 2e-16 ***
## StateHolidaya   -145904      17394  -8.388  < 2e-16 ***
## StateHolidayb    219444      21447  10.232  < 2e-16 ***
## StateHolidayc    294756      24959  11.810  < 2e-16 ***
```

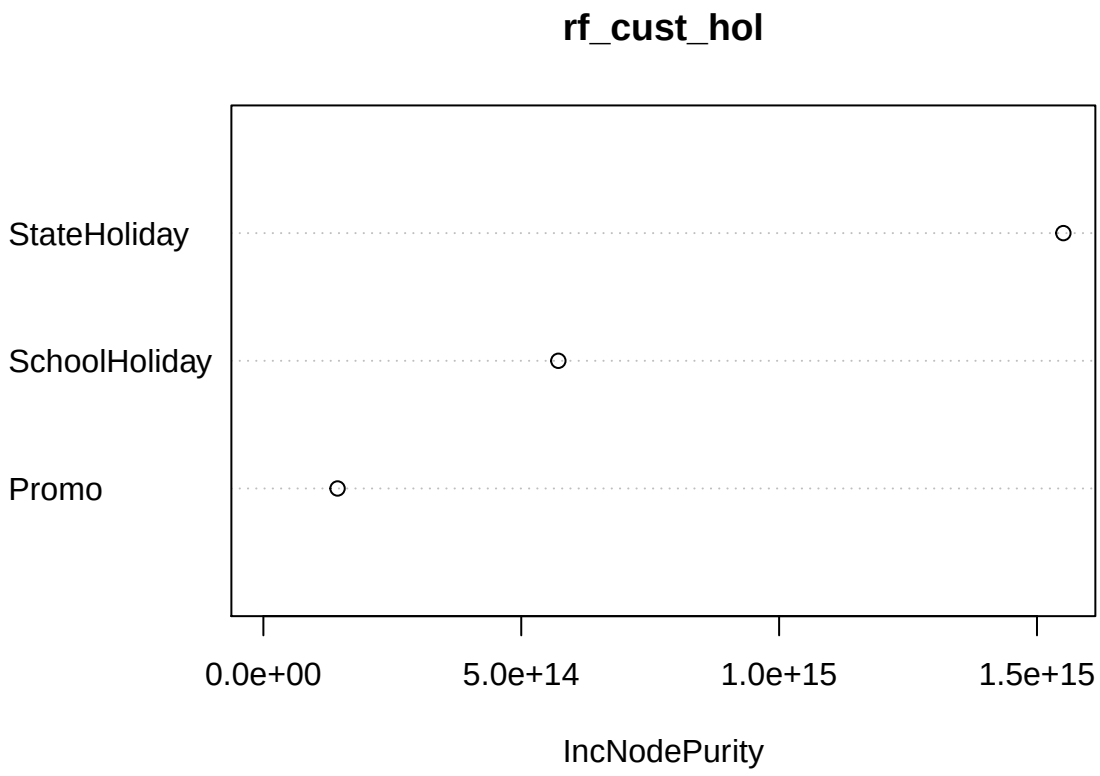
```
## SchoolHoliday1  -657148      12205 -53.843  < 2e-16 ***
## Promo1            33185      11538   2.876  0.00403 **
## Open1           1217179      17260  70.519  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 526600 on 9968 degrees of freedom
## Multiple R-squared:  0.6142, Adjusted R-squared:  0.614
## F-statistic:  2645 on 6 and 9968 DF,  p-value: < 2.2e-16
```

```
## [1] 502711.9
```

```
##
## Call:
## lm(formula = Customers ~ StateHoliday + SchoolHoliday + Promo +
##     Open, data = tr_hol)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -184034  -38262   -2560   11381 1493352
##
## Coefficients:
##                Estimate Std. Error t value Pr(>|t|)
## (Intercept)       38468       1997  19.260  < 2e-16 ***
## StateHolidaya    -13448       2304  -5.836 5.52e-09 ***
## StateHolidayb     28195       2841   9.923  < 2e-16 ***
## StateHolidayc     32312       3306   9.772  < 2e-16 ***
## SchoolHoliday1   -73452       1617 -45.429  < 2e-16 ***
## Promo1            -4592       1528  -3.005  0.00267 **
## Open1            137227       2287  60.013  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 69770 on 9968 degrees of freedom
## Multiple R-squared:  0.5238, Adjusted R-squared:  0.5235
## F-statistic:  1827 on 6 and 9968 DF,  p-value: < 2.2e-16
```

```
## [1] 62457.78
```

```
## [1] 606656
```


rf_cust_hol
StateHoliday
SchoolHoliday
Promo
0.0e+00
5.0e+14
1.0e+15
1.5e+15
IncNodePurity

```
## [1] 506013.2
```