Online Shopper's intention

Filip Mordarski

10/1/2019

Contents

Introduction

This report is part of the capstone project of the EdX course 'HarvardX: PH125.9x Data Science: Capstone'. Participants had to create their own project. Firstly, they need to seek out a dataset. I chose Online Shopper's Intention database from Kaggle. URL: Link

Executive Summary

The main goal of this project is building algorithms to predict if given online shopper will be generated revenue or not. It can be really useful information for entrepreneur. Online Shopper's Intention dataset contains around 12 thousands sessions of online shoppers. The information from the analysis of this database is used to generate revenue predictions that are compared with actual zero one variable revenue to check the quality of the forecasting algorithm.

Overview

This report is split in four sections. First, **Introduction** describes content of data set, summarizes the goal of the project and key steps that were performed. **Analysis** section explains the process and techniques used, such as data cleaning, data exploration and visualization and my modeling approach. In a **Results** section I present the modeling results and discuss the model performance. **Conclusion** section gives us a brief summary of the report, its limitations and future work.

Data set description

The dataset consists of feature vectors belonging to 12,330 sessions. The dataset was formed so that each session would belong to a different user in a 1-year period to avoid any tendency to a specific campaign, special day, user profile, or period. The dataset consists of 10 numerical and 8 categorical attributes.

- Administrative, Administrative Duration, Informational, Informational Duration, Product Related and Product Related Duration

These variables represent the number of different types of pages visited by the visitor in that session and total time spent in each of these page categories.

- Bounce Rate

Feature for a web page refers to the percentage of visitors who enter the site from that page and then leave ("bounce") without triggering any other requests to the analytics server during that session.

- Exit Rate

Feature for a specific web page is calculated as for all pageviews to the page, the percentage that were the last in the session.

- Page Value

Feature represents the average value for a web page that a user visited before completing an e-commerce transaction.

- Special Day

Feature indicates the closeness of the site visiting time to a specific special day (e.g. Mother's Day, Valentine's Day) in which the sessions are more likely to be finalized with transaction. The value of this attribute is determined by considering the dynamics of e-commerce such as the duration between the order date and delivery date. For example, for Valentina's day, this value takes a nonzero value between February 2 and February 12, zero before and after this date unless it is close to another special day, and its maximum value of 1 on February 8.

- Month

Attribute represents which month was during the visit.

- Operating System, Browser, Region, Traffic type

Attributes represent what operating system (8 different operating systems) and browser (13 different browsers) online shoppers were using. Region (9 different regions) and traffic type (20 different types) indicate region of the user and traffic type during shopping. We can not say anything more about these attributes, because they are numeric values. We can only determine their relationships between them and the impact on the dependent variable.

- Visitor type

Feature indicates visitors as returning or new visitor.

- Weekend

Boolean value indicating whether the date of the visit is weekend, and month of the year.

- Revenue

Attribute indicates if visitor will be generated revenue or not.

Analysis

This section expains the process and techniques used. It shows us some data visualization or data exploration. This section explains also modelling approach.

Data cleaning

Firstly I would like to remove all rows with any NA values. The code below.

```
# removing any rows with NA value
data <- data[complete.cases(data), ]
```

Now I am prepared to create train and testset for my algorithm.

```
y <- data$Revenue

# Create trainset and test set
set.seed(1)
test_index <- createDataPartition(y, times = 1, p = 0.1, list = FALSE)
test_set <- data[test_index, ]
train_set <- data[-test_index, ]
```

Table 1: What percentage of visitors generate revenue?

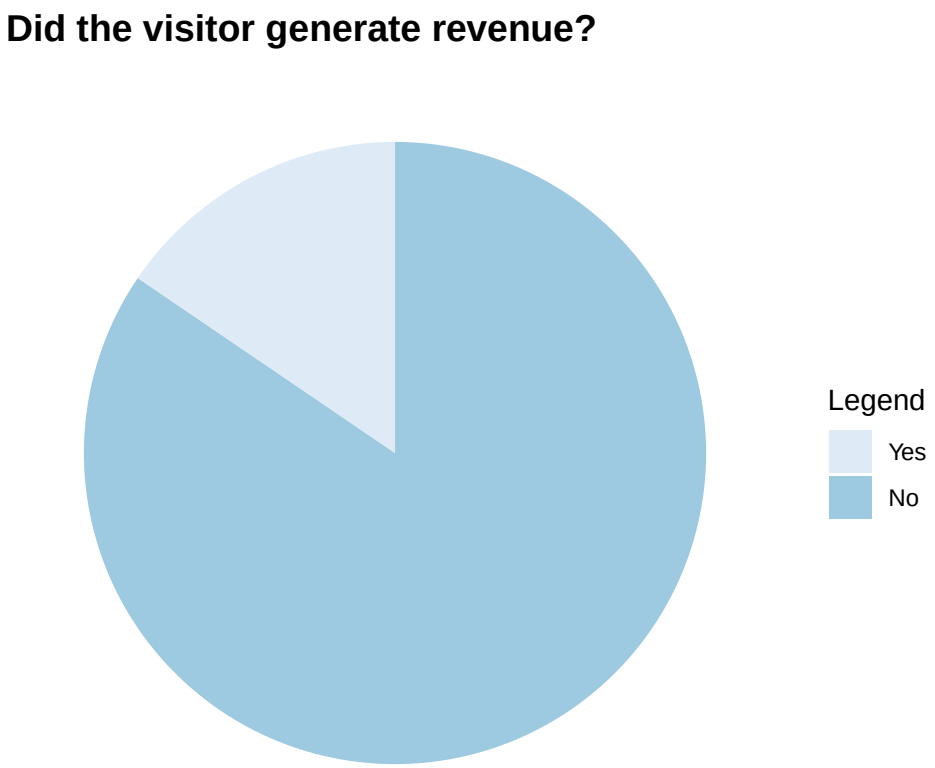
FreqPercTrue	FreqPercFalse
0.15	0.85

Data exploration and visualization

Now we can begin the eye-pleasing part of the report. The below section will be full of plots and tables.

Above table shows us what percentage of visitors generate revenue or not.

Below we can see pie chart which shows us the above data in a pleasant view.



From the previous table and plot indicates that about 85 % sessions were not generated any revenue.

The below part and table show us counts of sessions that generated revenue or not. It represents that the month in which the most recorded sessions was May. However, in November the most sessions generated revenue.

Table 2: Revenue counts by month

Month	True	False	Total
Dec	216	1511	1727
Nov	760	2238	2998
Oct	115	434	549
Sep	86	362	448
Aug	76	357	433
Jul	66	366	432
June	29	259	288
May	365	2998	3363
Mar	192	1702	1894
Feb	3	181	184

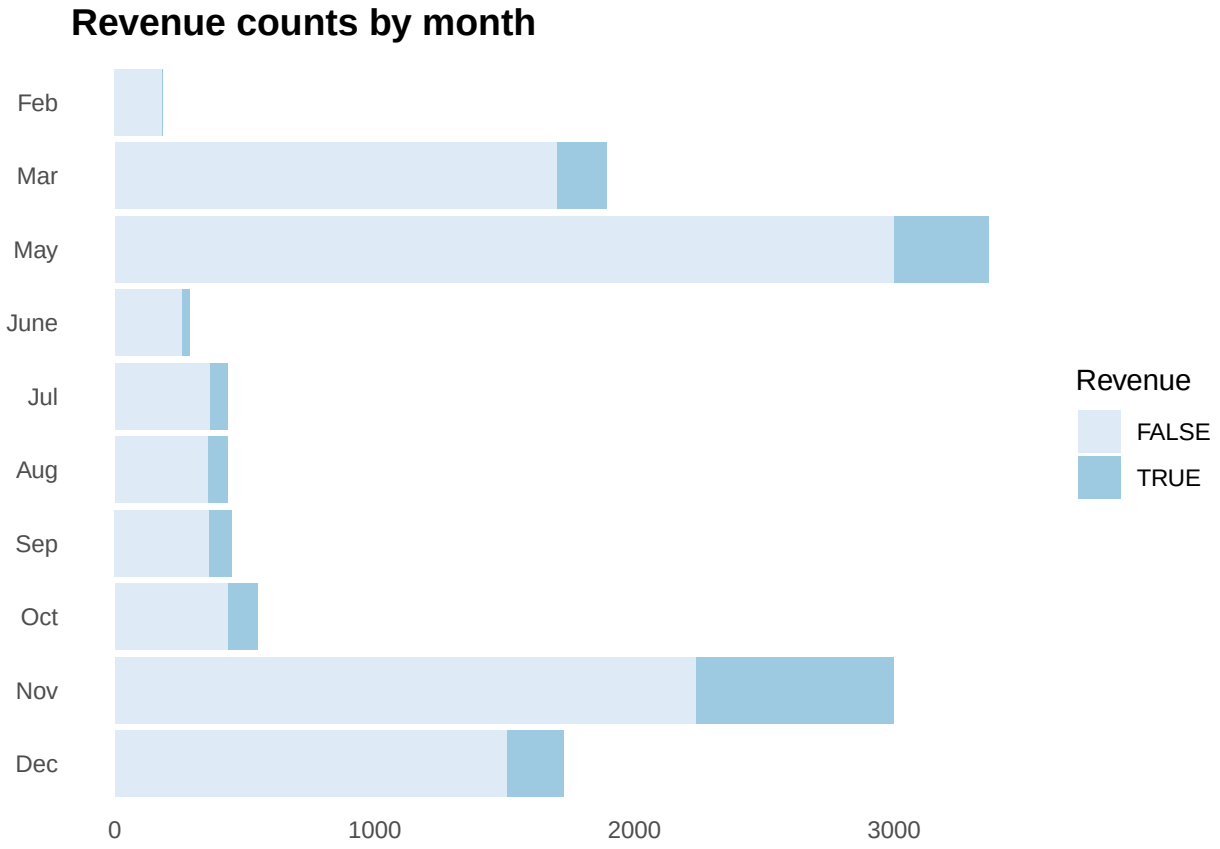
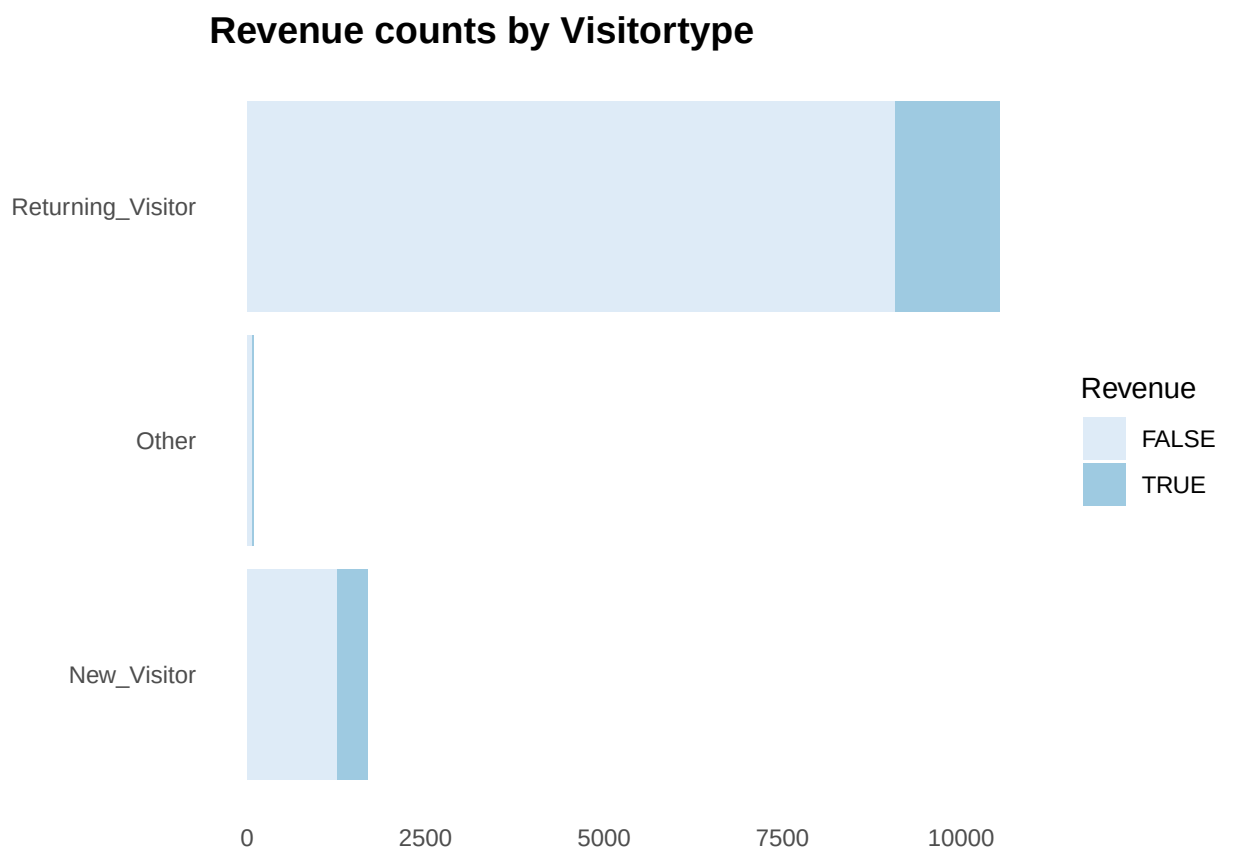


Table 3: Revenue counts by month

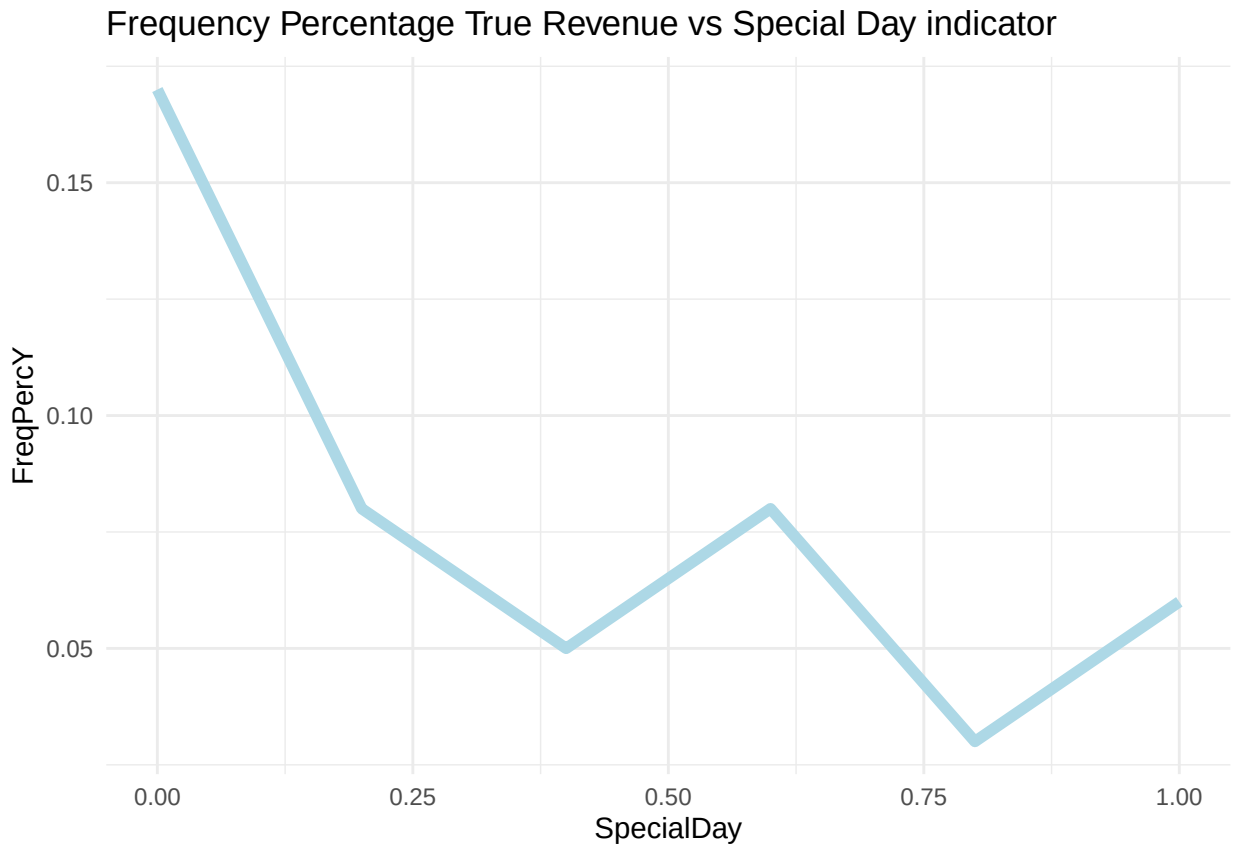
VisitorType	True	False	FreqPercTrue	Total
New_Visitor	422	1272	0.25	1694
Other	16	69	0.19	85
Returning_Visitor	1470	9067	0.14	10537



The above chart and table may seem surprising. Follows from them new visitor generated revenue more often than returning visitor. Type 'new visitor' generated revenue in 25 % sessions, 'other' 19 % and 'returning visitor' only 14 %. It can be really useful information for online shop.

Table 4: Frequency Percentage True Revenue by Special Day

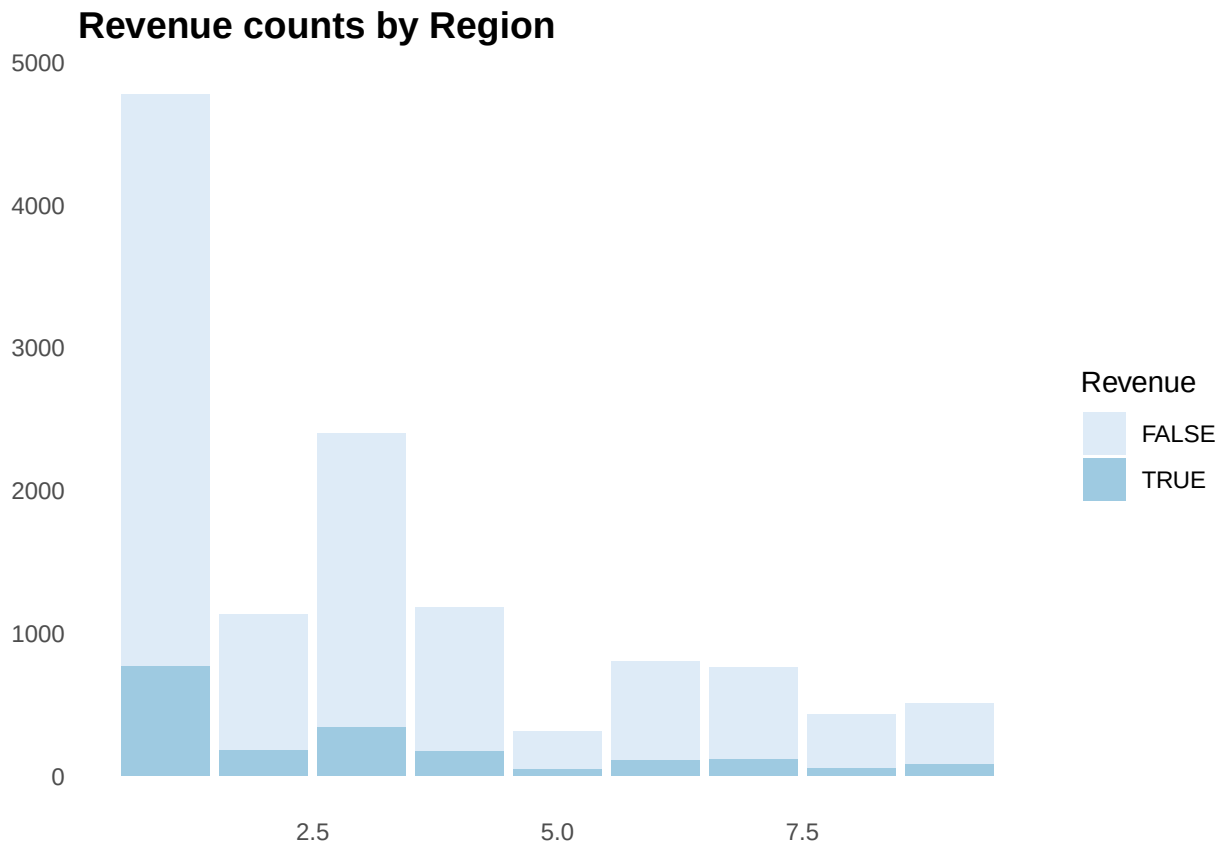
SpecialDay	CountsAllSessions	FreqPercY
0.0	11065	0.17
0.2	178	0.08
0.4	243	0.05
0.6	351	0.08
0.8	325	0.03
1.0	154	0.06



We can see from the above table and chart that the highest frequency percentage when session generate revenue is for Special Day equals 0. It also may seem surprising that the indicator is relatively low when the closeness of the site visiting time to a specific special day is relatively high.

Table 5: Revenue counts by Region

Region	CountsRevenueY	FreqPercY
1	771	0.16
2	188	0.17
3	349	0.15
4	175	0.15
5	52	0.16
6	112	0.14
7	119	0.16
8	56	0.13
9	86	0.17



The above table and plot represent statistics for Revenue grouped by region. we can infer from them that the most often region was Region 1. The most sessions that generated revenue were also in Region 1.

Modelling approach

I will use in my algorith classification method to predict if session generate revenue or not. The method 'Random Forest' from 'caret' package will be used. Below section was wrote based on 'Introduction to Data Science' written by Rafael A. Irizarry.

Random forests are a very popular machine learning approach that addresses the shortcomings of decision trees using a clever idea. The goal is to improve prediction performance and reduce instability by averaging multiple decision trees (a forest of trees constructed with randomness). It has two features that help accomplish this.

The first step is bootstrap aggregation or bagging. The general idea is to generate many predictors, each using regression or classification trees, and then forming a final prediction based on the average prediction of all these trees. To assure that the individual trees are not the same, we use the bootstrap to induce randomness. These two features combined explain the name: the bootstrap makes the individual trees randomly different, and the combination of trees is the forest.

So firstly what we need to do is convert type of columns 'Revenue' in trainset and testset to factors.

```
train_set$Revenue <- as.factor(train_set$Revenue)
test_set$Revenue <- as.factor(test_set$Revenue)
```

Now we can train our trainset using Random Forest method.

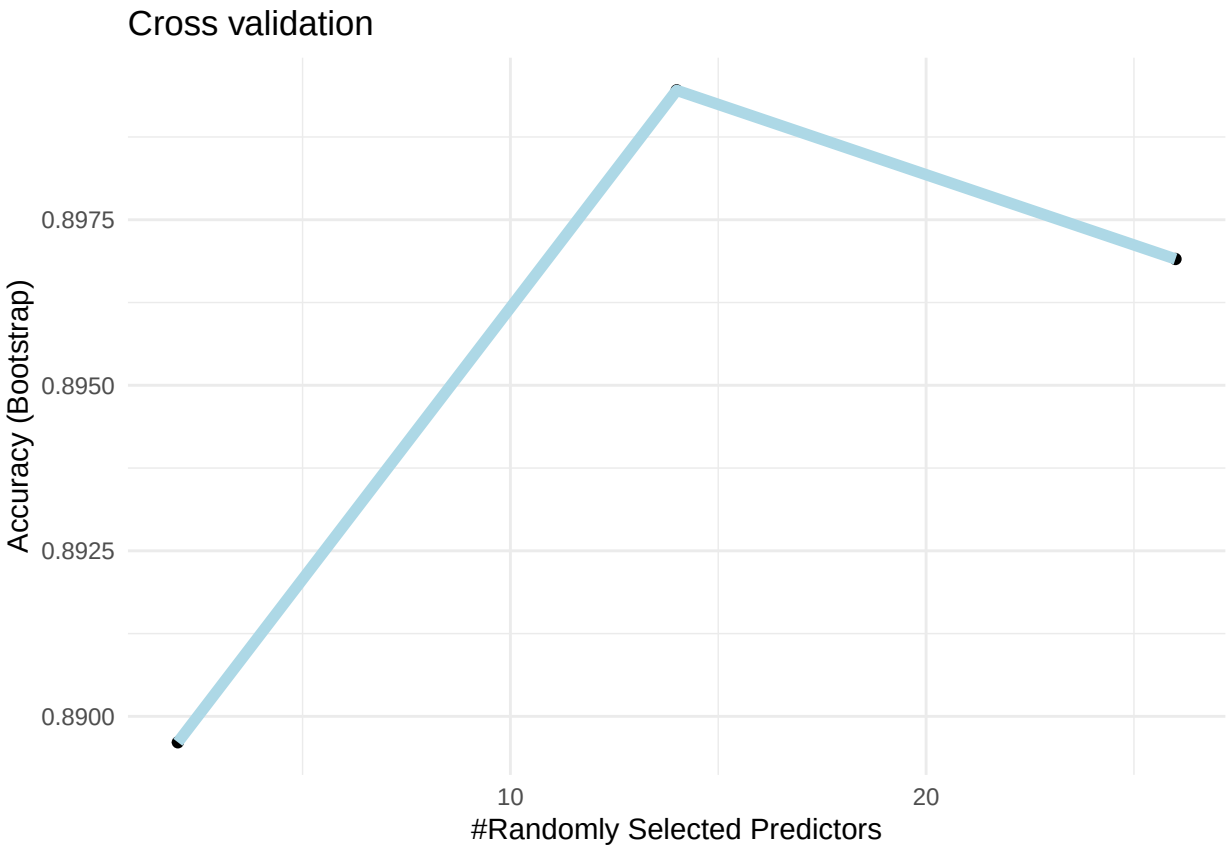
```
train_rf <- train(Revenue~., method = 'rf',data = train_set)
```

Results

In this section I will present the modeling results and discuss the model performance.

Cross validation

We can quickly see the results of the cross validation using the ggplot function.



From the above plot we can see that the smallest mtry value is in the range of 10 and 15. If we execute below code we can see the specific value of mtry (Randomly selected predictors).

```
train_rf$finalModel$mtry
```

```
## [1] 14
```

Algorithm efficiency

Now we are ready to predict our data in testset. Below code allows us to do this.

```
y_hat_knn <- predict(train_rf, test_set,type = "raw")
```

Below code informs us about our algorithm efficiency.

```
confusionMatrix(data=y_hat_knn, reference=test_set$Revenue)
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction FALSE TRUE
```

```
##      FALSE  1011   76
##      TRUE     30  115
##
##                Accuracy : 0.914
##                  95% CI : (0.8969, 0.929)
##     No Information Rate : 0.845
##     P-Value [Acc > NIR] : 4.358e-13
##
##                   Kappa : 0.6358
##
##  Mcnemar's Test P-Value : 1.238e-05
##
##             Sensitivity : 0.9712
##             Specificity : 0.6021
##          Pos Pred Value : 0.9301
##          Neg Pred Value : 0.7931
##              Prevalence : 0.8450
##          Detection Rate : 0.8206
##    Detection Prevalence : 0.8823
##       Balanced Accuracy : 0.7866
##
##        'Positive' Class : FALSE
##
```

If we would like to extract only Accuracy of our algorithm we can do this.

```
confusionMatrix(data=y_hat_knn, reference=test_set$Revenue)$overall["Accuracy"]
```

```
## Accuracy
## 0.913961
```

Conclusion

The aim of the project was to predict if given online shopper will be generated revenue or not. My algorithm achieved nice result of accuracy of **0.914**. This algorithm can be very useful for an entrepreneur running an online store to predict profits.