BUAN 6356.005 | Business Analytics with R

Online Shoppers' Intention Analysis

Presented by Group 9:

Hongbo Li

Jielin Zhang

Yixiao Ji

Nayan Nandakishore Joshi

Paritosh Mishra

Presentation video link:

https://cometmail-my.sharepoint.com/:v:/g/personal/yxj210010_utdallas_edu/EeBPVzcxCs5Oih0KVaxc41gBMB0iqiQIvNgXRbzgjKtbNg?e=XtkON0

Table of Contents

Executive Summary

This report provides an analysis on how to identify key metrics that contribute the most to predict an online shopper's behavior and build a predictive model to suggest prioritized critical recommendations and performance improvements.

We used several different methods to get the most accurate model. First, to understand the shopper segments, we used clustering analysis. Second, we implemented 3 different classification analysis methods, which include decision tree, neural network, and logistic regression. Third, to select the best model, we evaluated the 3 models by comparing their accuracy rate and Area Under the Curve (AUC).

Our report suggests that there is a positive correlation between conversion rate and the average time a user spent browsing the webpage before completing a transaction (page value). Higher page values will increase the conversion rate where users are converted to customers by placing orders online. In addition, shopping in November will also increase the probabilities of purchasing. In contrast, higher Product Related Duration and Exit Rates will decrease the conversion rate. Product related duration represents the number of different types of pages visited by a user in that session and total time spent in each of these page categories. For all pageviews to the page, exit rate is the percentage that were the last in the session. It is also observed that online shoppers are less likely to place an order during the months of December, February, June, March, and May.

To make the best use of the marketing resources, it is recommended to allocate more marketing resources to the webpages with higher page values and to the month of November. Limited marketing resources should be allocated to webpages with high product related duration and exit rates, and to the months of December, February, June, March, and May.

Project Background and Motivation

In the past 20 years, online shopping has become the new trend. More and more people have come to e-retailers online shopping. In 2021 alone, the number of online shoppers has risen to 2.14 billion, this figure is approximately 28% of the global population. This indicates a 4.4% increase per year, adding 900 million additional buyers in 2021.

For businesses to keep up with changing demands and behaviors, it’s better to arm ourselves with enough knowledge on emerging trends. Therefore, it’s critical to build models to predict if a visitor will make a purchase on the website.

Dataset Description

Data Origin

Our dataset “**online_shoppers_intention**” is a second-hand data. The dataset is a csv file and we retrieved it from UCI Machine Learning Repository. (Source:https://archive.ics.uci.edu/ml/datasets/Online+Shoppers+Purchasing+Intention+Dataset#)

Data Set Information

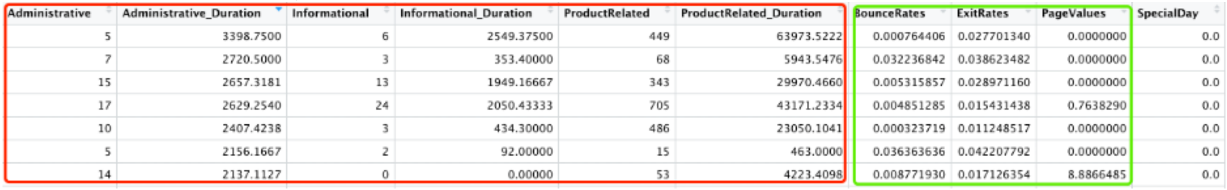
The dataset consists of feature vectors belonging to 12,330 sessions, 84.5% (10,422) were negative class samples where visitors did not end with shopping, and the rest (1,908) were positive class samples.

Number of Instances:	12,330	**Data Mining Objective:**	Classification, Clustering
Number of Attributes:	18	**Missing Values?**	N/A

Attribute Information

The dataset consists of 10 numerical and 8 categorical attributes.

Numerical attribute samples:



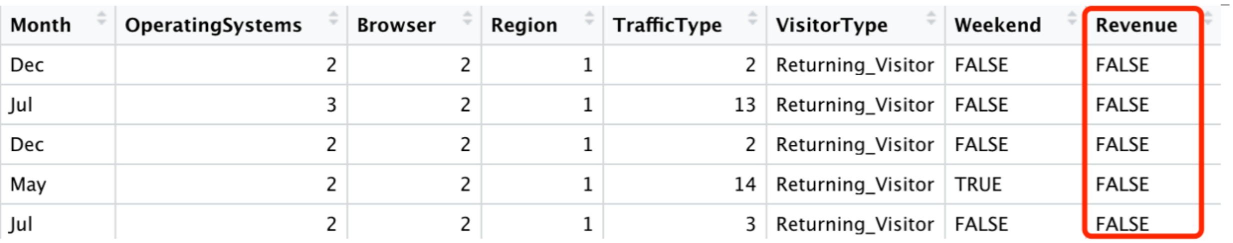
Administrative	Administrative_Duration	Informational	Informational_Duration	ProductRelated	ProductRelated_Duration	BounceRates	ExitRates	PageValues	SpecialDay
5	3398.7500	6	2549.37500	449	63973.5222	0.000764406	0.027701340	0.0000000	0.0
7	2720.5000	3	353.40000	68	5943.5476	0.032236842	0.038623482	0.0000000	0.0
15	2657.3181	13	1949.16667	343	29970.4660	0.005315857	0.028971160	0.0000000	0.0
17	2629.2540	24	2050.43333	705	43171.2334	0.004851285	0.015431438	0.7638290	0.0
10	2407.4238	3	434.30000	486	23050.1041	0.000323719	0.011248517	0.0000000	0.0
5	2156.1667	2	92.00000	15	463.0000	0.036363636	0.042207792	0.0000000	0.0
14	2137.1127	0	0.00000	53	4223.4098	0.008771930	0.017126354	8.8866485	0.0

"Administrative", "Administrative Duration", "Informational", "Informational Duration", "Product Related" and "Product Related Duration" represent the number of different types of pages visited by users in that session and total time spent in each of these page categories. This helps us to identify the page types that users spend the most of their time on and the most frequently visited pages.

"Bounce Rate", "Exit Rate" and "Page Values" represent the metrics measured by Google Analytics for each page in the e-commerce site. The value of "Bounce Rate" refers to the percentage of visitors who enter the site from that page and then leave without triggering any other requests to the analytics server during that session. "Page Values" represents the average value for a web page that a user visited before completing an e-commerce transaction.

The "Special Day" indicates the closeness of the site visiting time on a 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.

Categorical attribute samples:



Month	OperatingSystems	Browser	Region	TrafficType	VisitorType	Weekend	Revenue
Dec	2	2	1	2	Returning_Visitor	FALSE	FALSE
Jul	3	2	1	13	Returning_Visitor	FALSE	FALSE
Dec	2	2	1	2	Returning_Visitor	FALSE	FALSE
May	2	2	1	14	Returning_Visitor	TRUE	FALSE
Jul	2	2	1	3	Returning_Visitor	FALSE	FALSE

The "Revenue" attribute (last column) is used as the class label.

The dataset also includes information of the Operating System, Browser, Region, Traffic Type, Visitor Type (either returning visitor or new visitor). A Boolean value indicates whether the date of the visit is weekend, and month of the year.

Business Objective

The main objective revolved around the identification of key metrics which contributes the most to predicting a shopper's behavior. On this base, we'll build a predicting model as accurate as possible to suggest critical recommendations and performance improvements. Revenue is the attribute of interest which identifies if a purchase was made or not.

Exploratory Data Analysis

The frequency of the revenue class in the dataset shows that our dataset is unbalanced. We will walk through sampling process afterwards.

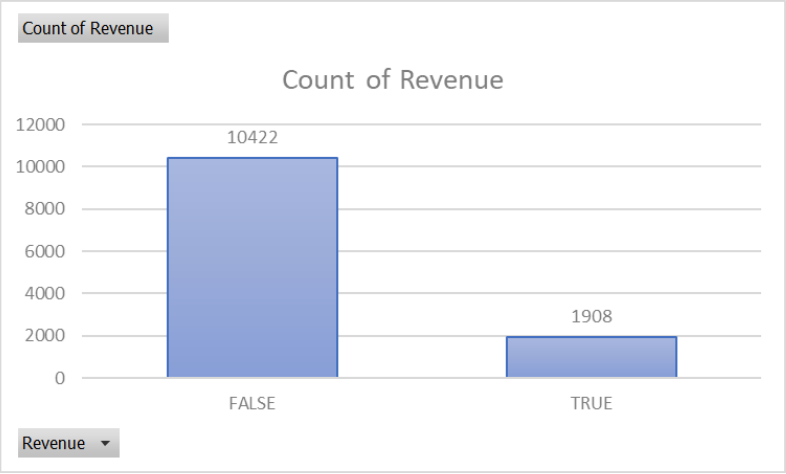


Figure 1 - Distribution of purchase conversion(revenue)

There are many influential factors for users' purchase behaviors. For example, the shopping behavior could be affected by the month of the year. The chart below gives an example where online purchase is significantly high in the months of May and November.

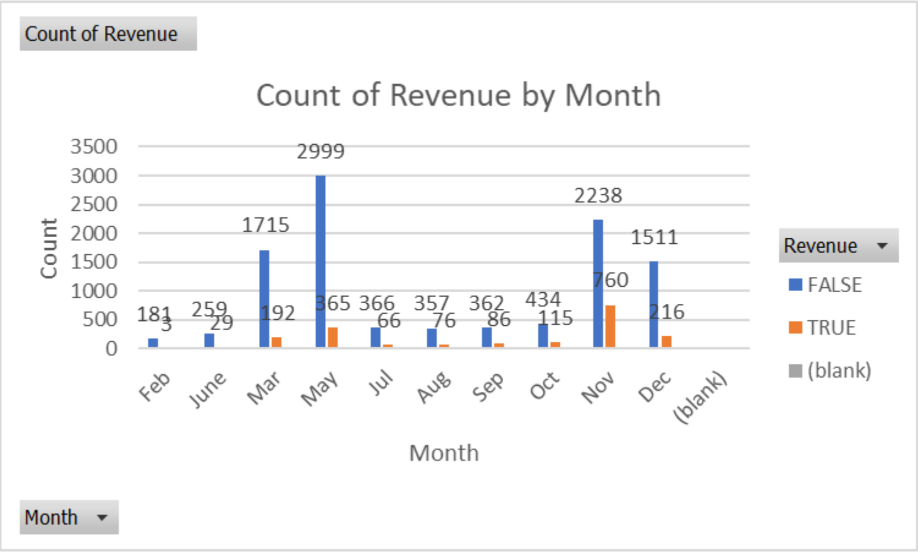


Figure 2 - Distribution of purchase conversion by Month

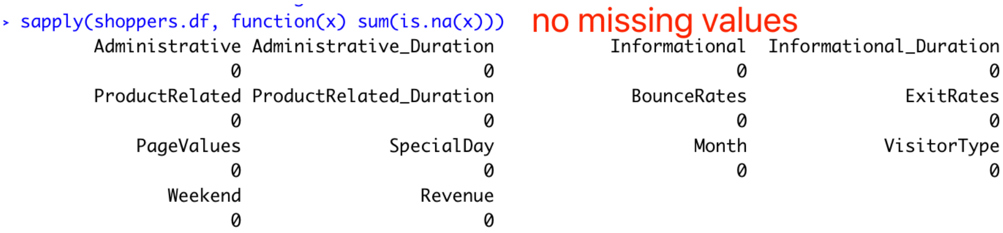
The objective of this project is to develop a model to predict users' future purchases by identifying independent variables with the most impacts on the revenue.

Data Preprocessing and Sampling

Data Preprocessing

The first step is to examine if there is missing value in the dataset. The second step is to turn the logic variables containing True/False labels into binary variables. We executed the following codes in RStudio and examined that there is no missing value in the dataset.

```
# check if there is missing value
sapply(shoppers.df, function(x) sum(is.na(x)))

# turn logic variables into binary
str(shoppers.df)
shoppers.df$Revenue <- ifelse(shoppers.df$Revenue==TRUE, 1, 0)
shoppers.df$Weekend <- ifelse(shoppers.df$Weekend==TRUE, 1, 0)
str(shoppers.df)
```



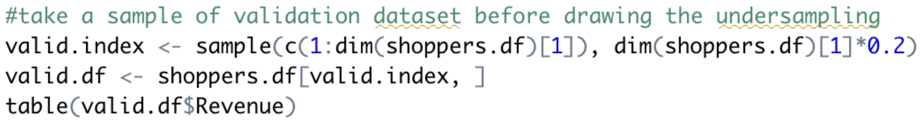
```
> sapply(shoppers.df, function(x) sum(is.na(x)))
         Administrative Administrative_Duration           Informational  Informational_Duration
                      0                       0                       0                       0
         ProductRelated ProductRelated_Duration             BounceRates               ExitRates
                      0                       0                       0                       0
             PageValues              SpecialDay                   Month             VisitorType
                      0                       0                       0                       0
                Weekend                 Revenue
                      0                       0
```

no missing values

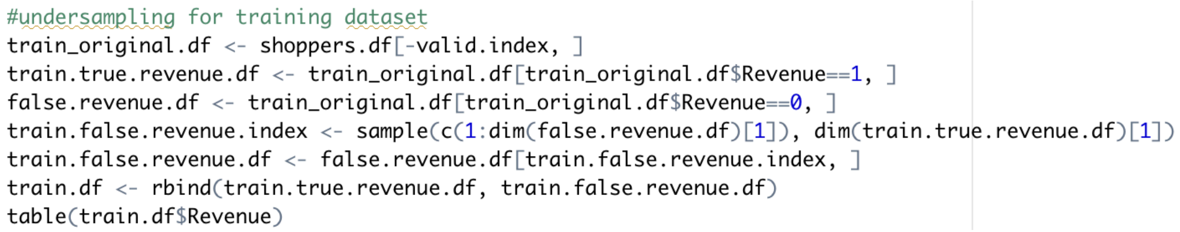
Data Sampling

As shown in the *Figure 1,* the dataset isn't balanced. With imbalanced datasets, an algorithm cannot get the necessary information of the minority class for accurate prediction. It will result in biased predictions and misleading accuracies.

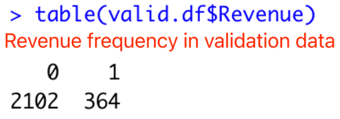
When sampling the training data, we used under-sampling method to randomly reduce the number of observations from majority class to match total observations from minority class to balance dataset.



```
#take a sample of validation dataset before drawing the undersampling
valid.index <- sample(c(1:dim(shoppers.df)[1]), dim(shoppers.df)[1]*0.2)
valid.df <- shoppers.df[valid.index, ]
table(valid.df$Revenue)
```



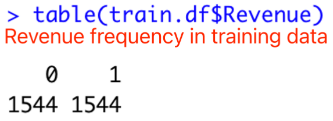
```
#undersampling for training dataset
train_original.df <- shoppers.df[-valid.index, ]
train.true.revenue.df <- train_original.df[train_original.df$Revenue==1, ]
false.revenue.df <- train_original.df[train_original.df$Revenue==0, ]
train.false.revenue.index <- sample(c(1:dim(false.revenue.df)[1]), dim(train.true.revenue.df)[1])
train.false.revenue.df <- false.revenue.df[train.false.revenue.index, ]
train.df <- rbind(train.true.revenue.df, train.false.revenue.df)
table(train.df$Revenue)
```



```
> table(valid.df$Revenue)
```

Revenue frequency in validation data

```
   0    1 
2102  364 
```



```
> table(train.df$Revenue)
```

Revenue frequency in training data

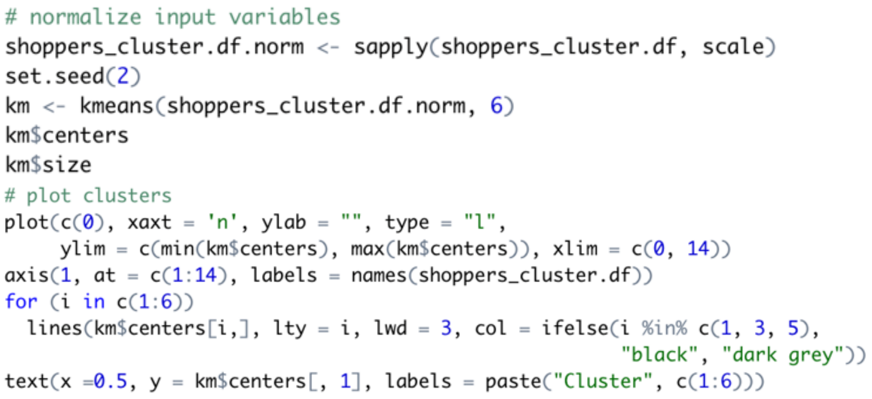
```
   0    1 
1544 1544 
```

Data Mining - Model Building, Evaluation and Selection

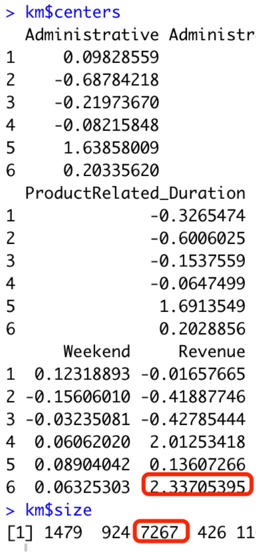
In data mining, we first implemented clustering analysis to identify shopper segments. To build the most accurate model, we also executed 3 different classification analysis methods which include decision tree, neural network, and logistic regression. Lastly, we evaluated these 3 models by comparing their accuracy rate and AUC value.

Cluster Analysis

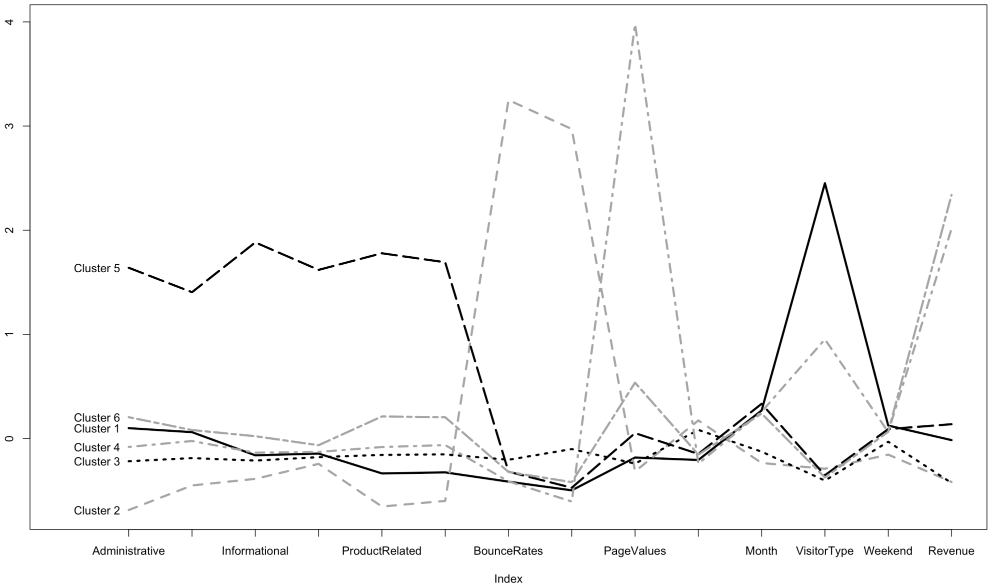
Before running k-means clustering method, we turned all the categorical variables into numerical values and then normalized the dataset to get more reasonable cluster partitions with the following code.



```
# normalize input variables
shoppers_cluster.df.norm <- sapply(shoppers_cluster.df, scale)
set.seed(2)
km <- kmeans(shoppers_cluster.df.norm, 6)
km$centers
km$size
# plot clusters
plot(c(0), xaxt = 'n', ylab = "", type = "l",
     ylim = c(min(km$centers), max(km$centers)), xlim = c(0, 14))
axis(1, at = c(1:14), labels = names(shoppers_cluster.df))
for (i in c(1:6))
  lines(km$centers[i,], lty = i, lwd = 3, col = ifelse(i %in% c(1, 3, 5),
                                                       "black", "dark grey"))
text(x =0.5, y = km$centers[, 1], labels = paste("Cluster", c(1:6)))
```



```
> km$centers
  Administrative Administrative_Duration Informational Informational_Duration ProductRelated
1     0.09828559              0.05965959   -0.16277236            -0.14605279    -0.33657779
2    -0.68784218             -0.45229338   -0.38879327            -0.24492057    -0.65459636
3    -0.21973670             -0.18945219   -0.21336773            -0.18058277    -0.15870985
4    -0.08215848             -0.02536264   -0.13772322            -0.13012158    -0.08321326
5     1.63858009              1.40398428    1.88211423             1.61835534     1.77776864
6     0.20335620              0.08004324    0.02069211            -0.06574944     0.21051473
  ProductRelated_Duration BounceRates  ExitRates  PageValues  SpecialDay      Month VisitorType
1              -0.3265474  -0.4155342 -0.4990924 -0.18411570 -0.20751674  0.2683573   2.4504006
2              -0.6006025   3.2499905  2.9711129 -0.31716498  0.17214987 -0.2363502  -0.2918299
3              -0.1537559  -0.2057491 -0.1027822 -0.24244011  0.08149733 -0.1278426  -0.4053793
4              -0.0647499  -0.4130220 -0.6067052  3.97830254 -0.24272338  0.2555404   0.9515568
5               1.6913549  -0.3197853 -0.4741935  0.05051777 -0.14959926  0.3324535  -0.3541981
6               0.2028856  -0.3224810 -0.4204879  0.53518777 -0.15592897  0.2381981  -0.3753591
      Weekend     Revenue
1  0.12318893 -0.01657665
2 -0.15606010 -0.41887746
3 -0.03235081 -0.42785444
4  0.06062020  2.01253418
5  0.08904042  0.13607266
6  0.06325303  2.33705395
> km$size
[1] 1479   924 7267  426 1162 1072
```

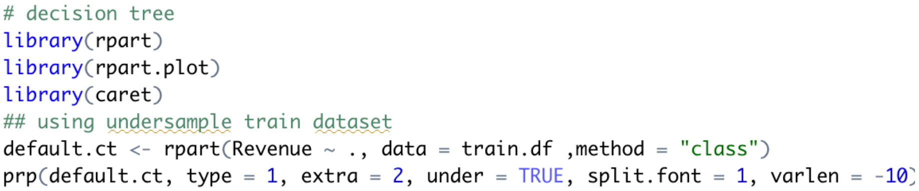


Cluster 6 ends up as the group with the highest Revenue. In addition, the members in Cluster 6 have higher values of the attributes of ProductRelated, PageValues and VisitorType than all other attributes.

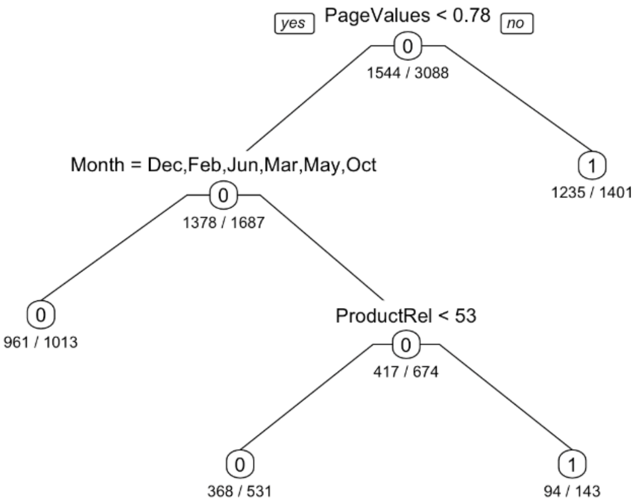
Decision Tree

Since we already have two existing classes of "True" or "False" of revenue, we are going to use the technique of classification to build models. To acquire the most accurate result, we carried out the analysis in 3 different ways.

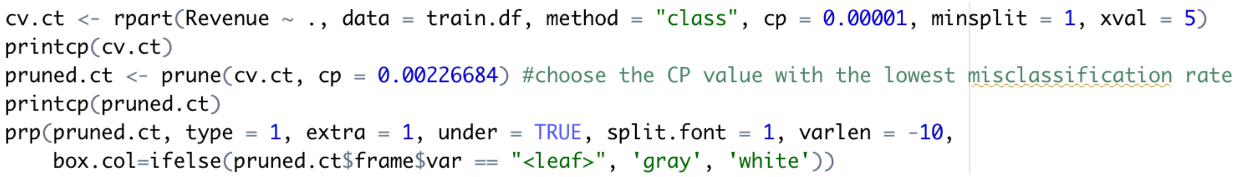
i. Model Building



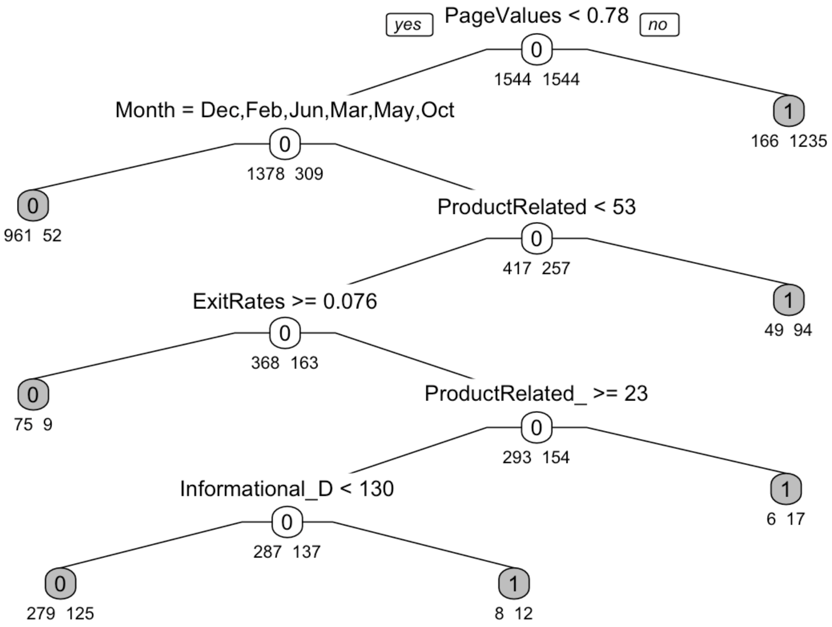
```
# decision tree
library(rpart)
library(rpart.plot)
library(caret)
## using undersample train dataset
default.ct <- rpart(Revenue ~ ., data = train.df ,method = "class")
prp(default.ct, type = 1, extra = 2, under = TRUE, split.font = 1, varlen = -10
```



The returned default decision tree includes 3 nodes of page values, month, and product related pages, along with 4 leaves. In order to find the optimal number of leaves, we chose the complexity parameter with the lowest misclassification rate and input it manually into the prune function.



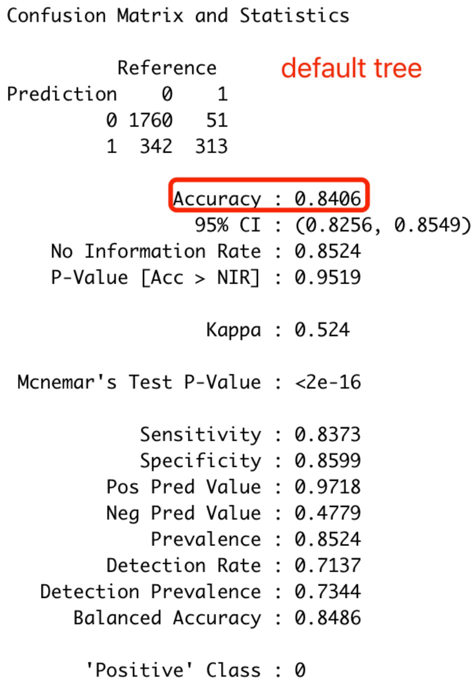
```
cv.ct <- rpart(Revenue ~ ., data = train.df, method = "class", cp = 0.00001, minsplit = 1, xval = 5)
printcp(cv.ct)
pruned.ct <- prune(cv.ct, cp = 0.00226684) #choose the CP value with the lowest misclassification rate
printcp(pruned.ct)
prp(pruned.ct, type = 1, extra = 1, under = TRUE, split.font = 1, varlen = -10,
    box.col=ifelse(pruned.ct$frame$var == "<leaf>", 'gray', 'white'))
```



The variables that were used in pruned tree construction are PageValues, Month, ProductRelated, ExitRates, ProductRelated_Duration and Informational_Duration.

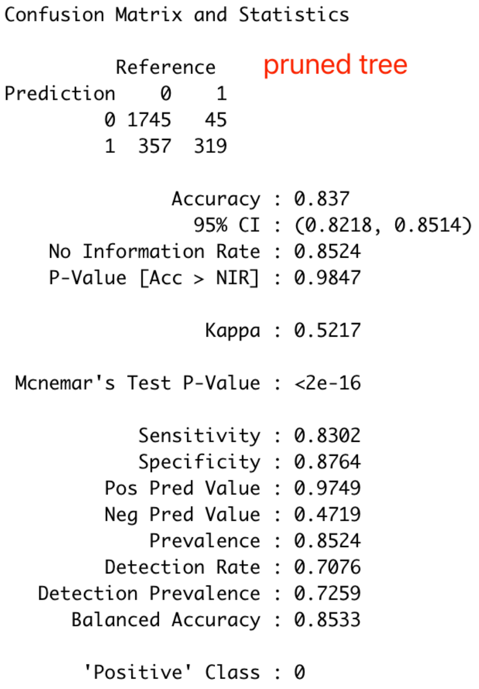
ii. Accuracy Rate of the Decision Trees

```
#default model accuracy rate: 0.8406
default.ct.pred.valid <- predict(default.ct, valid.df, type = "class")
confusionMatrix(default.ct.pred.valid, as.factor(valid.df$Revenue))
# pruned tree accuracy rate: 0.837
pruned.ct.pred.valid <- predict(pruned.ct, valid.df, type = "class")
confusionMatrix(pruned.ct.pred.valid, as.factor(valid.df$Revenue))
```



default tree

```
Confusion Matrix and Statistics

          Reference
Prediction    0    1
         0 1760   51
         1  342  313

               Accuracy : 0.8406
                 95% CI : (0.8256, 0.8549)
    No Information Rate : 0.8524
    P-Value [Acc > NIR] : 0.9519

                  Kappa : 0.524

 Mcnemar's Test P-Value : <2e-16

            Sensitivity : 0.8373
            Specificity : 0.8599
         Pos Pred Value : 0.9718
         Neg Pred Value : 0.4779
             Prevalence : 0.8524
         Detection Rate : 0.7137
   Detection Prevalence : 0.7344
      Balanced Accuracy : 0.8486

       'Positive' Class : 0
```



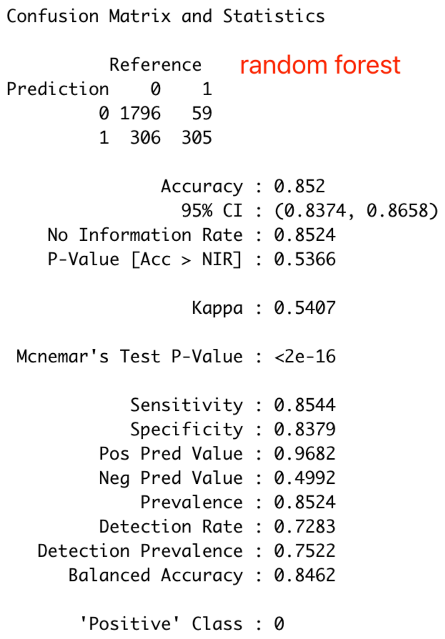
pruned tree

```
Confusion Matrix and Statistics

          Reference
Prediction    0    1
         0 1745   45
         1  357  319

               Accuracy : 0.837
                 95% CI : (0.8218, 0.8514)
    No Information Rate : 0.8524
    P-Value [Acc > NIR] : 0.9847

                  Kappa : 0.5217

 Mcnemar's Test P-Value : <2e-16

            Sensitivity : 0.8302
            Specificity : 0.8764
         Pos Pred Value : 0.9749
         Neg Pred Value : 0.4719
             Prevalence : 0.8524
         Detection Rate : 0.7076
   Detection Prevalence : 0.7259
      Balanced Accuracy : 0.8533

       'Positive' Class : 0
```

Compared to the default tree, the pruned tree with the optimal number of splits returned a lower accuracy rate when applied to validation dataset. We decided to implement random forest method get a model with better performance.

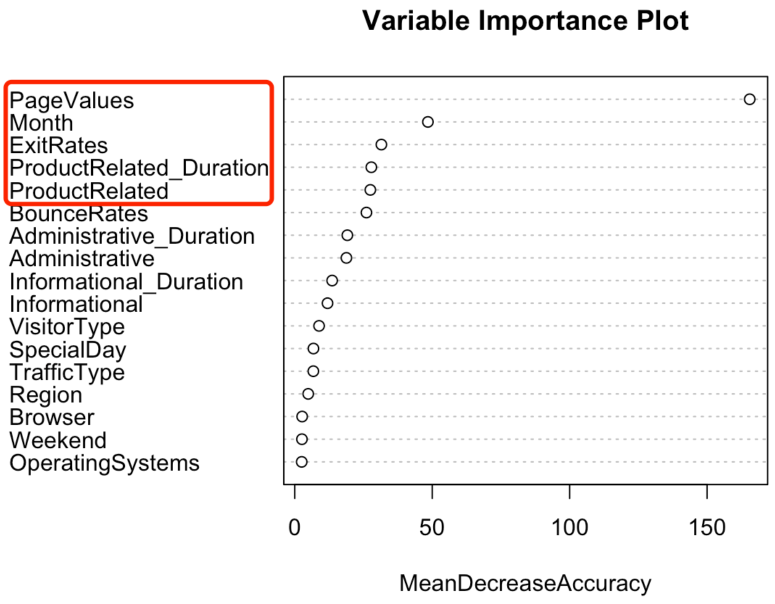
iii. Accuracy Rate of the Random Forest

```
library(randomForest)
rf <- randomForest(as.factor(Revenue) ~ ., data = train.df, ntree = 500,
                   mtry = 4, nodesize = 5, importance = TRUE)
summary(rf)
# randomForest accuracy rate: 0.852
rf.pred <- predict(rf, valid.df)
confusionMatrix(rf.pred, as.factor(valid.df$Revenue))
# variable importance plot
varImpPlot(rf, type = 1)
```



```
Confusion Matrix and Statistics

          Reference
Prediction    0    1
         0 1796   59
         1  306  305

               Accuracy : 0.852
                 95% CI : (0.8374, 0.8658)
    No Information Rate : 0.8524
    P-Value [Acc > NIR] : 0.5366

                  Kappa : 0.5407

 Mcnemar's Test P-Value : <2e-16

            Sensitivity : 0.8544
            Specificity : 0.8379
         Pos Pred Value : 0.9682
         Neg Pred Value : 0.4992
             Prevalence : 0.8524
         Detection Rate : 0.7283
   Detection Prevalence : 0.7522
      Balanced Accuracy : 0.8462

       'Positive' Class : 0
```

random forest

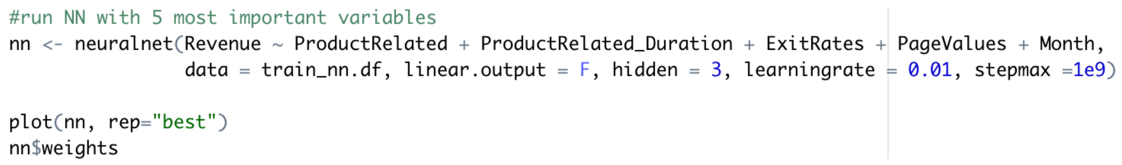


The accuracy rate is 0.852 based on the confusion matrix of the random forest model, it is the highest among the above 3 decision tree models. According to the variable importance plot on the right side, the top 5 variables maked in red align with the previous 2 models.

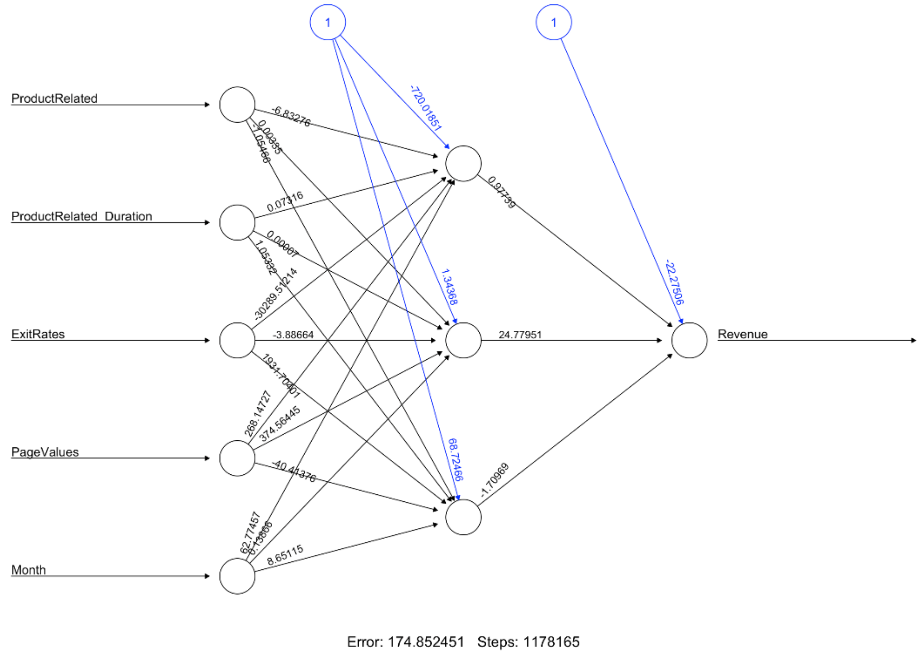
Neural Network

i. Model building

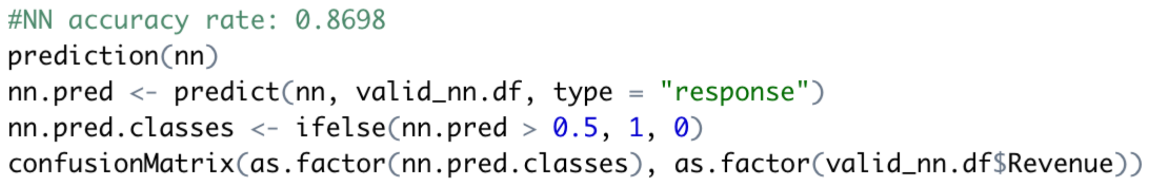
Given that revenue is binomial variable, we decide to use neural network classification for a better classifier. By running decision tree models, we acknowledge relatively important attributes of our dataset. And therefore, we pass the top 5 important variables into the neural network model to avoid system crush in R studio.



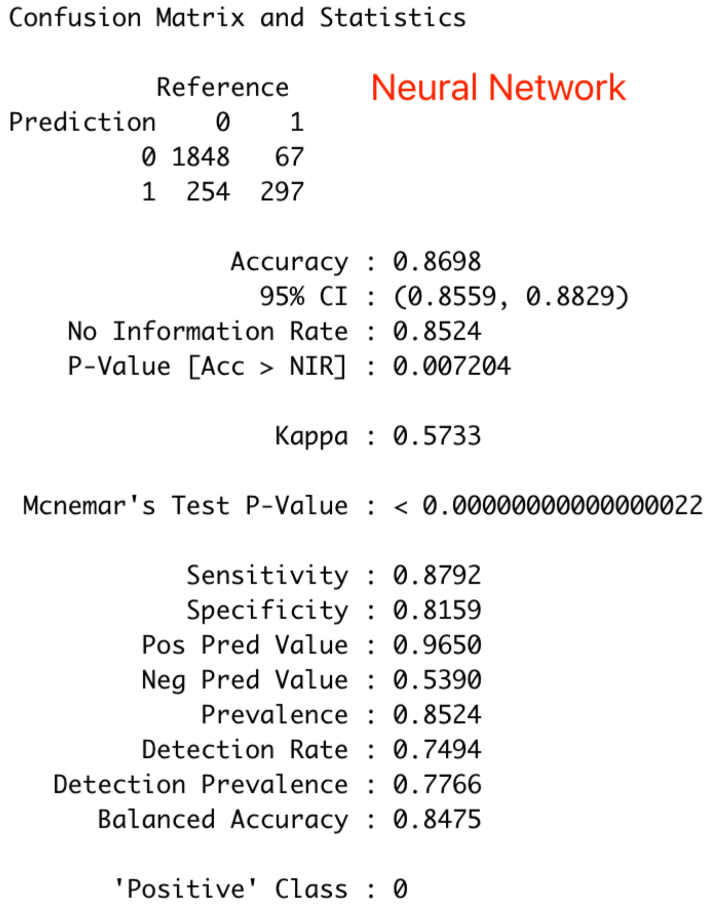
```
#run NN with 5 most important variables
nn <- neuralnet(Revenue ~ ProductRelated + ProductRelated_Duration + ExitRates + PageValues + Month,
                data = train_nn.df, linear.output = F, hidden = 3, learningrate = 0.01, stepmax =1e9)

plot(nn, rep="best")
nn$weights
```



ii. Accuracy Rate of the Neural Network



```
#NN accuracy rate: 0.8698
prediction(nn)
nn.pred <- predict(nn, valid_nn.df, type = "response")
nn.pred.classes <- ifelse(nn.pred > 0.5, 1, 0)
confusionMatrix(as.factor(nn.pred.classes), as.factor(valid_nn.df$Revenue))
```



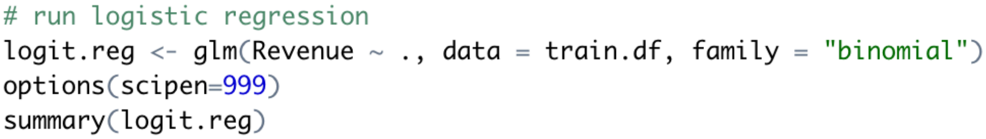
```
Confusion Matrix and Statistics

          Reference
Prediction    0    1
         0 1848   67
         1  254  297
                                          
               Accuracy : 0.8698          
                 95% CI : (0.8559, 0.8829)
    No Information Rate : 0.8524          
    P-Value [Acc > NIR] : 0.007204        
                                          
                  Kappa : 0.5733          
                                          
 Mcnemar's Test P-Value : < 0.00000000000000022
                                          
            Sensitivity : 0.8792          
            Specificity : 0.8159          
         Pos Pred Value : 0.9650          
         Neg Pred Value : 0.5390          
             Prevalence : 0.8524          
         Detection Rate : 0.7494          
   Detection Prevalence : 0.7766          
      Balanced Accuracy : 0.8475          
                                          
       'Positive' Class : 0               
```

The neural network gives the output in decimal values ranging from 0 to 1. We keep the threshold value as 0.5, values below the threshold are converted to 0 and values above the threshold are converted to 1. We concluded an accuracy rate of 0.8698 of the neural network model, which is higher than that of the decision tree.

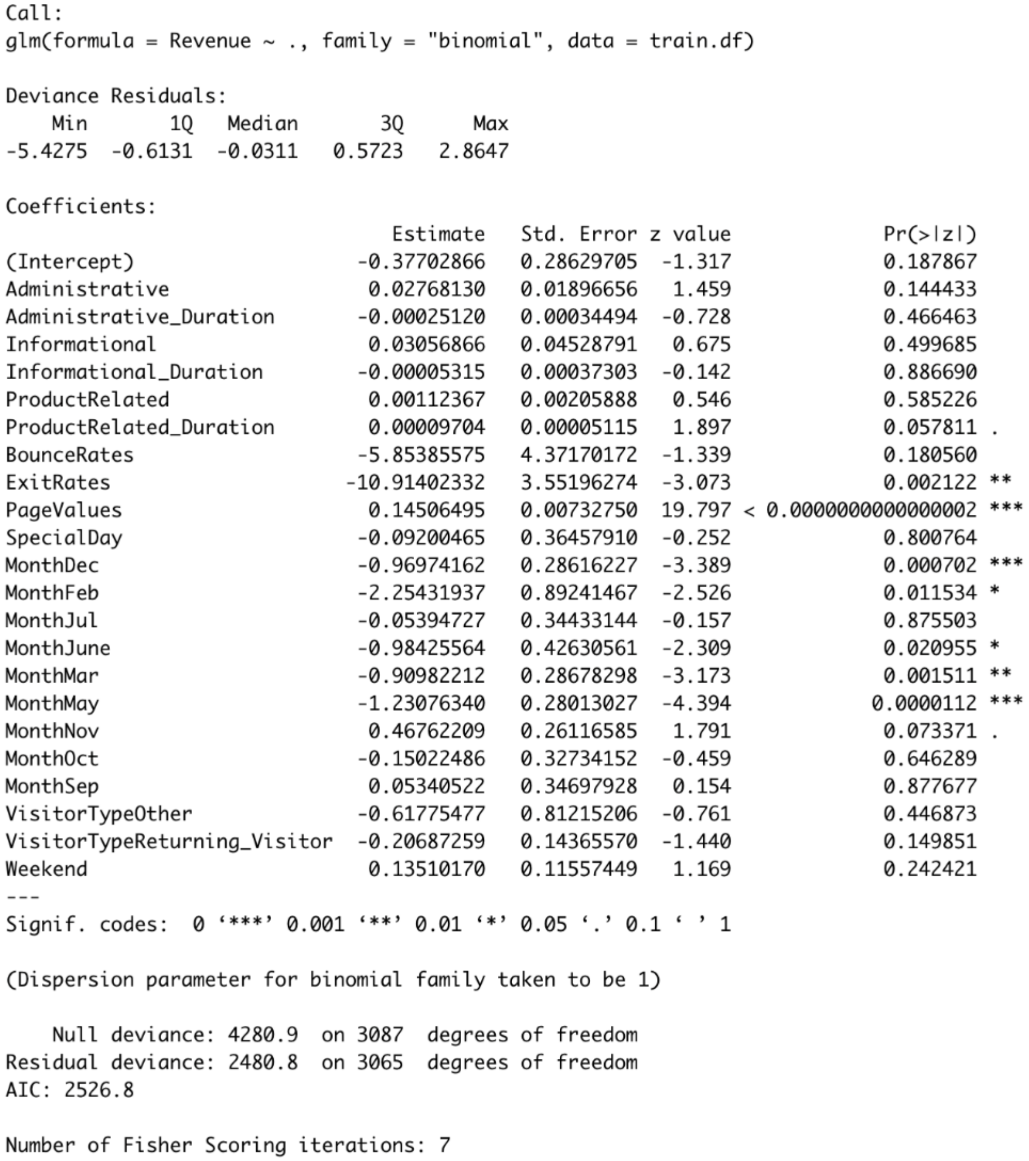
Logistic Regression

The attribute “Revenue” in the dataset is a binominal variable, thus logistic regression can also be applied for the analysis.

i. Model Building



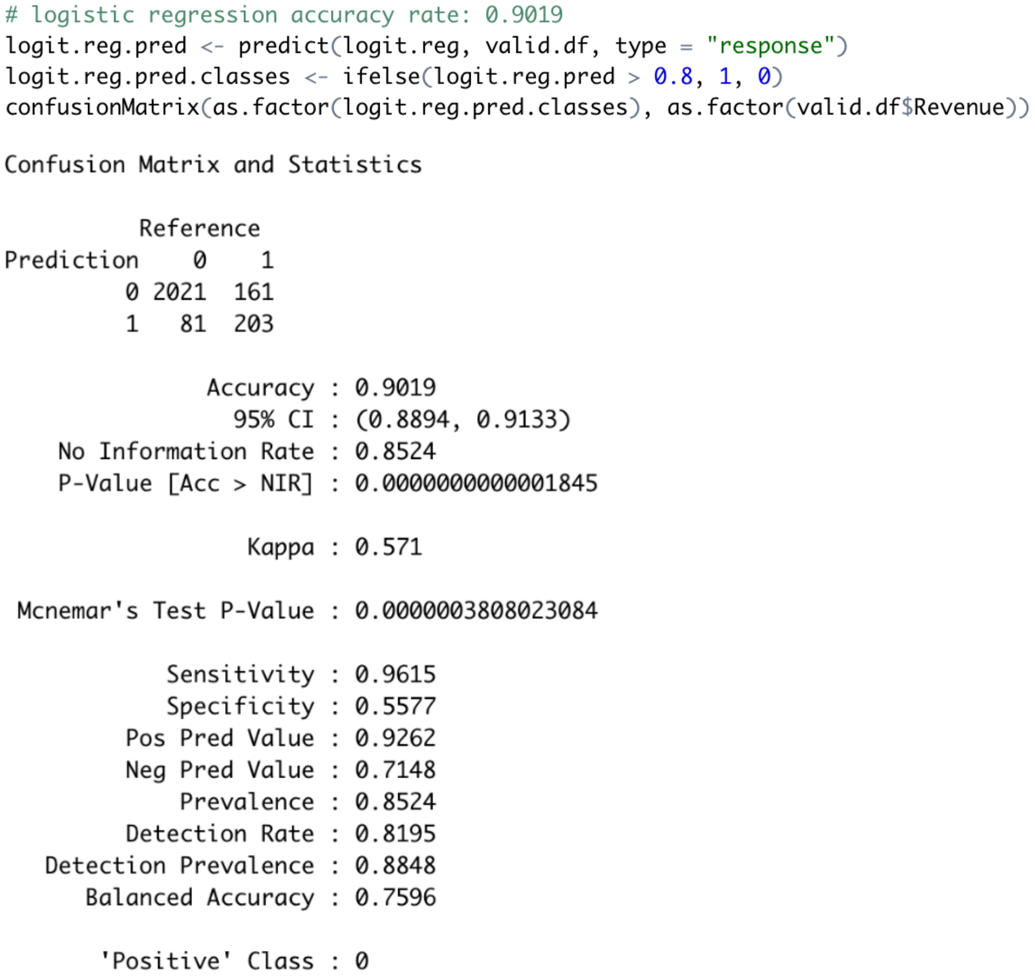
```
# run logistic regression
logit.reg <- glm(Revenue ~ ., data = train.df, family = "binomial")
options(scipen=999)
summary(logit.reg)
```



```
Call:
glm(formula = Revenue ~ ., family = "binomial", data = train.df)

Deviance Residuals:
    Min       1Q   Median       3Q      Max  
-5.4275  -0.6131  -0.0311   0.5723   2.8647  

Coefficients:
                               Estimate  Std. Error z value             Pr(>|z|)    
(Intercept)                 -0.37702866  0.28629705  -1.317             0.187867    
Administrative               0.02768130  0.01896656   1.459             0.144433    
Administrative_Duration     -0.00025120  0.00034494  -0.728             0.466463    
Informational                0.03056866  0.04528791   0.675             0.499685    
Informational_Duration      -0.00005315  0.00037303  -0.142             0.886690    
ProductRelated               0.00112367  0.00205888   0.546             0.585226    
ProductRelated_Duration      0.00009704  0.00005115   1.897             0.057811 .  
BounceRates                 -5.85385575  4.37170172  -1.339             0.180560    
ExitRates                  -10.91402332  3.55196274  -3.073             0.002122 ** 
PageValues                   0.14506495  0.00732750  19.797 < 0.0000000000000002 ***
SpecialDay                  -0.09200465  0.36457910  -0.252             0.800764    
MonthDec                    -0.96974162  0.28616227  -3.389             0.000702 ***
MonthFeb                    -2.25431937  0.89241467  -2.526             0.011534 *  
MonthJul                    -0.05394727  0.34433144  -0.157             0.875503    
MonthJune                   -0.98425564  0.42630561  -2.309             0.020955 *  
MonthMar                    -0.90982212  0.28678298  -3.173             0.001511 ** 
MonthMay                    -1.23076340  0.28013027  -4.394            0.0000112 ***
MonthNov                     0.46762209  0.26116585   1.791             0.073371 .  
MonthOct                    -0.15022486  0.32734152  -0.459             0.646289    
MonthSep                     0.05340522  0.34697928   0.154             0.877677    
VisitorTypeOther            -0.61775477  0.81215206  -0.761             0.446873    
VisitorTypeReturning_Visitor -0.20687259  0.14365570  -1.440             0.149851    
Weekend                      0.13510170  0.11557449   1.169             0.242421    
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 4280.9  on 3087  degrees of freedom
Residual deviance: 2480.8  on 3065  degrees of freedom
AIC: 2526.8

Number of Fisher Scoring iterations: 7
```

From the logistic regression model, we had similar results as the decision tree. Page values and some of the months are still significant. However, we had two more significant independent variables of Exit Rates and Bounce Rates with both p values < 0.01.

ii. Accuracy Rate of the Logistic Regression Full Model

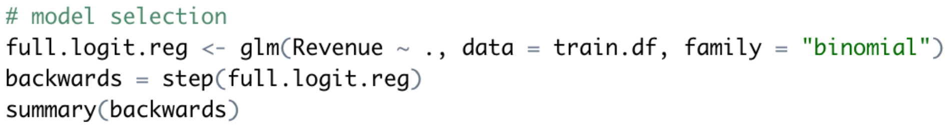


```
# logistic regression accuracy rate: 0.9019
logit.reg.pred <- predict(logit.reg, valid.df, type = "response")
logit.reg.pred.classes <- ifelse(logit.reg.pred > 0.8, 1, 0)
confusionMatrix(as.factor(logit.reg.pred.classes), as.factor(valid.df$Revenue))

Confusion Matrix and Statistics

          Reference
Prediction    0    1
         0 2021  161
         1   81  203

               Accuracy : 0.9019
                 95% CI : (0.8894, 0.9133)
    No Information Rate : 0.8524
    P-Value [Acc > NIR] : 0.0000000000001845

                  Kappa : 0.571

 Mcnemar's Test P-Value : 0.0000003808023084

            Sensitivity : 0.9615
            Specificity : 0.5577
         Pos Pred Value : 0.9262
         Neg Pred Value : 0.7148
             Prevalence : 0.8524
         Detection Rate : 0.8195
   Detection Prevalence : 0.8848
      Balanced Accuracy : 0.7596

       'Positive' Class : 0
```

iii. Accuracy Rate of the Partial Model

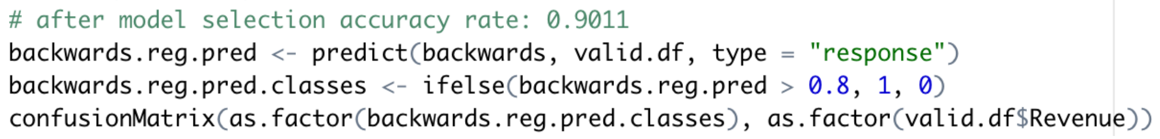
The above logistic regression model takes all variables as input. We used stepwise function to come up with the most significant variables and include these variables in the second model to compare their performance differences. It ended up in the following formula:

$$\text{Revenue} \sim \text{Administrative} + \text{ProductRelated\_Duration} + \text{ExitRates} + \text{PageValues} + \text{Month}$$



```
# model selection
full.logit.reg <- glm(Revenue ~ ., data = train.df, family = "binomial")
backwards = step(full.logit.reg)
summary(backwards)
```

```
Call:
glm(formula = Revenue ~ Administrative + ProductRelated_Duration + 
    ExitRates + PageValues + Month, family = "binomial", data = train.df)

Deviance Residuals: 
    Min       1Q   Median       3Q      Max  
-5.4209  -0.6239  -0.0376   0.5755   2.7296  

Coefficients:
                           Estimate   Std. Error z value             Pr(>|z|)    
(Intercept)             -0.39632515   0.26344109  -1.504             0.132474    
Administrative           0.02374646   0.01485188   1.599             0.109846    
ProductRelated_Duration  0.00011134   0.00002692   4.136     0.00003533051279 ***
ExitRates              -15.69528839   2.23135463  -7.034     0.00000000000201 ***
PageValues               0.14414833   0.00724857  19.886 < 0.0000000000000002 ***
MonthDec                -0.98886194   0.28581184  -3.460             0.000541 ***
MonthFeb                -2.33216858   0.87986553  -2.651             0.008035 ** 
MonthJul                -0.07625775   0.34367289  -0.222             0.824399    
MonthJune               -0.99402450   0.42385653  -2.345             0.019017 *  
MonthMar                -0.92819271   0.28589614  -3.247             0.001168 ** 
MonthMay                -1.27629204   0.27222336  -4.688     0.00000275349305 ***
MonthNov                 0.46677876   0.26129728   1.786             0.074036 .  
MonthOct                -0.16552815   0.32745380  -0.506             0.613207    
MonthSep                 0.06984381   0.34718353   0.201             0.840564    
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 4280.9  on 3087  degrees of freedom
Residual deviance: 2487.7  on 3074  degrees of freedom
AIC: 2515.7

Number of Fisher Scoring iterations: 7
```



```
# after model selection accuracy rate: 0.9011
backwards.reg.pred <- predict(backwards, valid.df, type = "response")
backwards.reg.pred.classes <- ifelse(backwards.reg.pred > 0.8, 1, 0)
confusionMatrix(as.factor(backwards.reg.pred.classes), as.factor(valid.df$Revenue))
```

```
Confusion Matrix and Statistics

          Reference
Prediction    0    1
         0 2022  164
         1   80  200
                                          
               Accuracy : 0.9011          
                 95% CI : (0.8886, 0.9126)
    No Information Rate : 0.8524          
    P-Value [Acc > NIR] : 0.0000000000004684
                                          
                  Kappa : 0.5653          
                                          
 Mcnemar's Test P-Value : 0.0000001075214194
                                          
            Sensitivity : 0.9619          
            Specificity : 0.5495          
         Pos Pred Value : 0.9250          
         Neg Pred Value : 0.7143          
             Prevalence : 0.8524          
         Detection Rate : 0.8200          
   Detection Prevalence : 0.8865          
      Balanced Accuracy : 0.7557          
                                          
       'Positive' Class : 0               
```

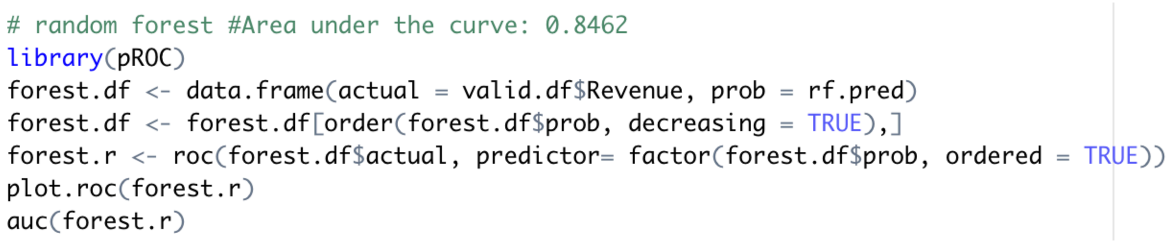
The best performance formula almost has the same accuracy rate as the full model, but with less variables which is more practical in application.

In conclusion, from the comparison of accuracy rate, the best model with the highest accuracy rate is from logistic regression: Revenue ~ Administrative + ProductRelated_Duration + ExitRates + PageValues + Month.

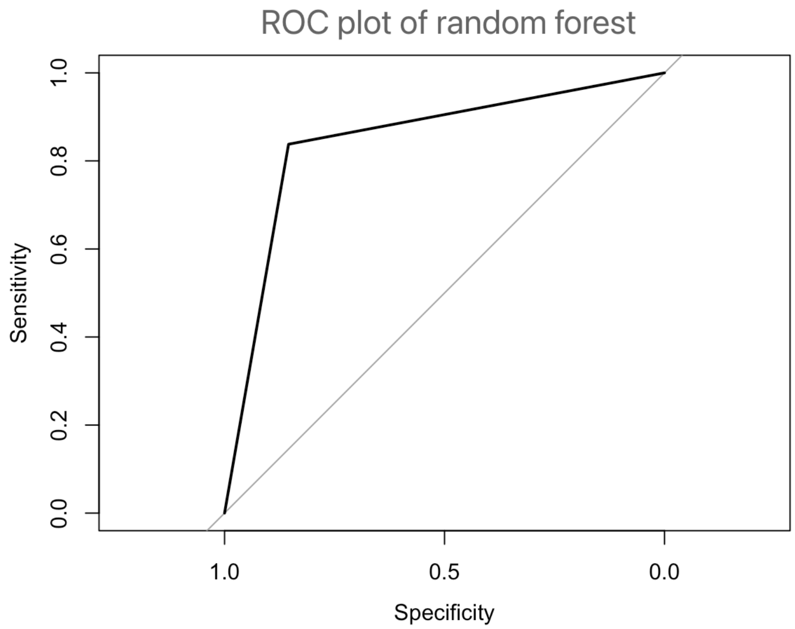
Classifiers Evaluation Using Receiver Operating Characteristic (ROC) Curve

Since accuracy rate can be misleading sometimes, it is not enough to decide on the model just by accuracy rate. We also used ROC curve to compare the 3 different models using validation data.

ROC Curve of the Random Forest

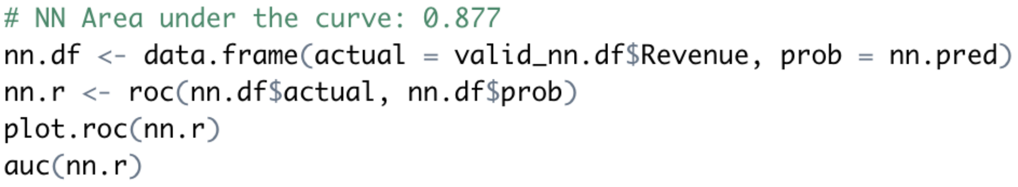


```
# random forest #Area under the curve: 0.8462
library(pROC)
forest.df <- data.frame(actual = valid.df$Revenue, prob = rf.pred)
forest.df <- forest.df[order(forest.df$prob, decreasing = TRUE),]
forest.r <- roc(forest.df$actual, predictor= factor(forest.df$prob, ordered = TRUE))
plot.roc(forest.r)
auc(forest.r)
```

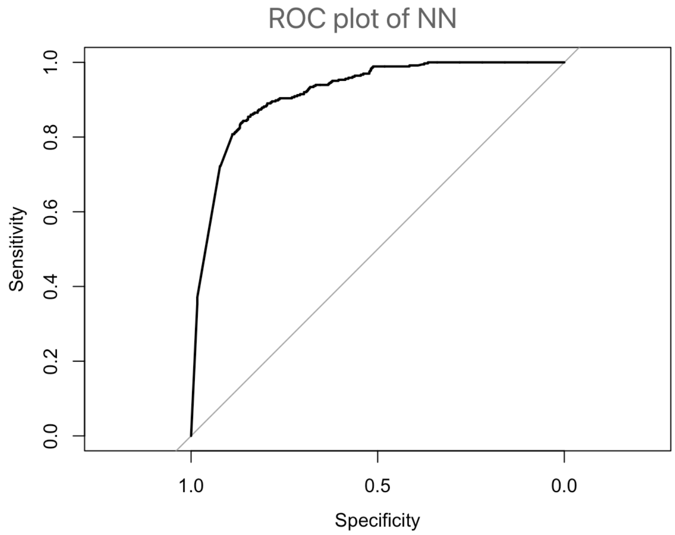


The above codes were executed in R and the area under the curve is 0.8462.

ROC Curve of the Neural Network

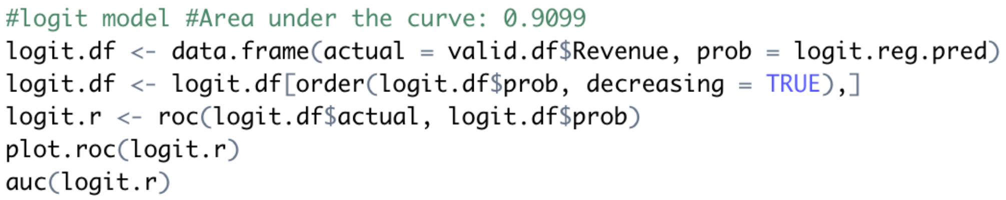


```
# NN Area under the curve: 0.877
nn.df <- data.frame(actual = valid_nn.df$Revenue, prob = nn.pred)
nn.r <- roc(nn.df$actual, nn.df$prob)
plot.roc(nn.r)
auc(nn.r)
```

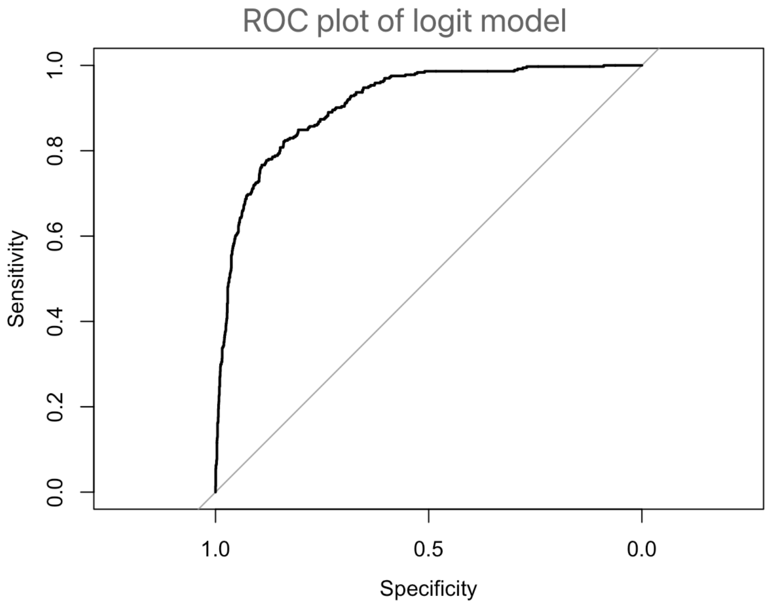


The above codes were executed in R and the area under the curve is 0.877.

ROC Curve of the Logistic Regression



```
#logit model #Area under the curve: 0.9099
logit.df <- data.frame(actual = valid.df$Revenue, prob = logit.reg.pred)
logit.df <- logit.df[order(logit.df$prob, decreasing = TRUE),]
logit.r <- roc(logit.df$actual, logit.df$prob)
plot.roc(logit.r)
auc(logit.r)
```



The above codes were executed in R and the area under the curve is 0.9099.

The logistic regression model outperforms the other two models according to the area under the ROC curve. This result is in accordance with the accuracy rate comparison result.

Model Selection

Model Building Method	Accuracy Rate	AUC
Decision tree/random forest	0.8520	0.8462
Neural Network	0.8698	0.8770
Logistic Regression	0.9011	0.9099

The above table is a summary of the accuracy rate and AUC of the 3 models. We can see that logistic model has both the highest accuracy rate and AUC value. We can conclude that the logistic regression model is the best one. The ultimate model we selected is:

$$\mathrm{Revenue} \sim 0.0001 * \mathrm{ProductRelated\_Duration} - 15.6953 * \mathrm{ExitRates} + 0.1441 * \mathrm{PageValues} - 0.9889 * \mathrm{MonthDec} - 2.3321 * \mathrm{MonthFeb} - 0.994 * \mathrm{MonthJune} - 0.9282 * \mathrm{MonthMar} - 1.2763 * \mathrm{MonthMay} + 0.4668 * \mathrm{MonthNov}$$

Findings and Managerial Conclusions

From the built models, we can conclude that there are positive correlations between revenue and PageValues, and MonthNov. Higher page values will lead to more online shopping conversions. Similarly, in November, online visitors are more likely to make purchases compared to other months. From a management perspective, marketing traffics should be allocated to webpages with high page values and to the month of November to boost sales.

The models also showed us negative relations between revenue and ProductRelated_Duration, and ExitRates. Web visitors are less likely to be converted to customers when browsing these webpages. In the months of December, February, June, March, and May, sales are decreasing compared to other months of the year. The management team should consider allocating less marketing traffics and resources on these webpages.

References

Sakar, C.O., Polat, S.O., Katircioglu, M. et al. Neural Comput & Applic (2018).

Dua, D. and Graff, C. (2019). UCI Machine Learning Repository [http://archive.ics.uci.edu/ml]. Irvine, CA: University of California, School of Information and Computer Science.