Air France Analysis

Sky Insights - Team 11

14/12/2020

Introduction

Air France is pursuing an international growth strategy and wants to enhance performance in regards of return on advertising (ROA) and dollars spent for (SEM) campaigns.We propose an optimization of the online marketing campaigns with the objective of driving more traffic on the website and converting it into customers while keeping the click costs minimized by running an in-depth analysis of the past and current campaigns. We are Sky Insigths from Media Contacts and in this report we are going to discuss how Air France can increase market share in the US air travel market.

Massaging the data

```
# renaming the columns for easier use in analysis
af_variables <- c("publisher_ID", "publisher_name", "keyword_ID", "keyword",
                  "match_type", "campaign", "keyword_group", "category",
                  "bid_strategy", "keyword_type", "status", "search_engine_bid",
                  "clicks", "click_charges", "avg_CpC", "impressions",
                  "engine_click_thru", "avg_pos", "trans_conv",
                  "total_cost.trans", "amount", "total_cost", "total_vol_bookings")
colnames(af_data) <- af_variables

# Printing the first five observations for the categorical and numerical data
print (af_data[1:5, 1:11])
```

```
## # A tibble: 5 x 11
##   publisher_ID publisher_name keyword_ID keyword match_type campaign
##   <chr>        <chr>          <chr>      <chr>   <chr>      <chr>
## 1 K2615        Yahoo - US     430000000~ fly to~ Advanced   Western~
## 2 K2615        Yahoo - US     430000000~ low in~ Advanced   Geo Tar~
## 3 K2003        MSN - Global   430000000~ air di~ Broad      Air Fra~
## 4 K1175        Google - Glob~ 430000000~ [airfr~ Exact      Air Fra~
## 5 K1123        Overture - Gl~ 430000000~ air fr~ Standard   Unassig~
## # ... with 5 more variables: keyword_group <chr>, category <chr>,
## #   bid_strategy <chr>, keyword_type <chr>, status <chr>
```

```
print (af_data[1:5, 12:23])
```

```
## # A tibble: 5 x 12
##   search_engine_b~ clicks click_charges avg_CpC impressions engine_click_th~
##              <dbl>  <dbl>         <dbl>   <dbl>       <dbl>            <dbl>
## 1             6.25      1         2.31   2.31            11             9.09
## 2             6.25      1         0.625  0.625            6            16.7
## 3             0         1         0.388  0.388            9            11.1
## 4             7.5      59         2.31   0.0392         401            14.7
## 5             0.25      8         2.2    0.275          318             2.52
## # ... with 6 more variables: avg_pos <dbl>, trans_conv <dbl>,
## #   total_cost.trans <dbl>, amount <dbl>, total_cost <dbl>,
## #   total_vol_bookings <dbl>
```

The data is made up of eleven categorical variables, with keyword ID identifying the lowest level of granularity where we can see all the information relating to a specific keyword within a particular campaign and publisher. The remaining twelve variables are numerical. It is important to consider that some of them are products of others (eg. engine click thru), while others are duplicated (click charge and total cost).

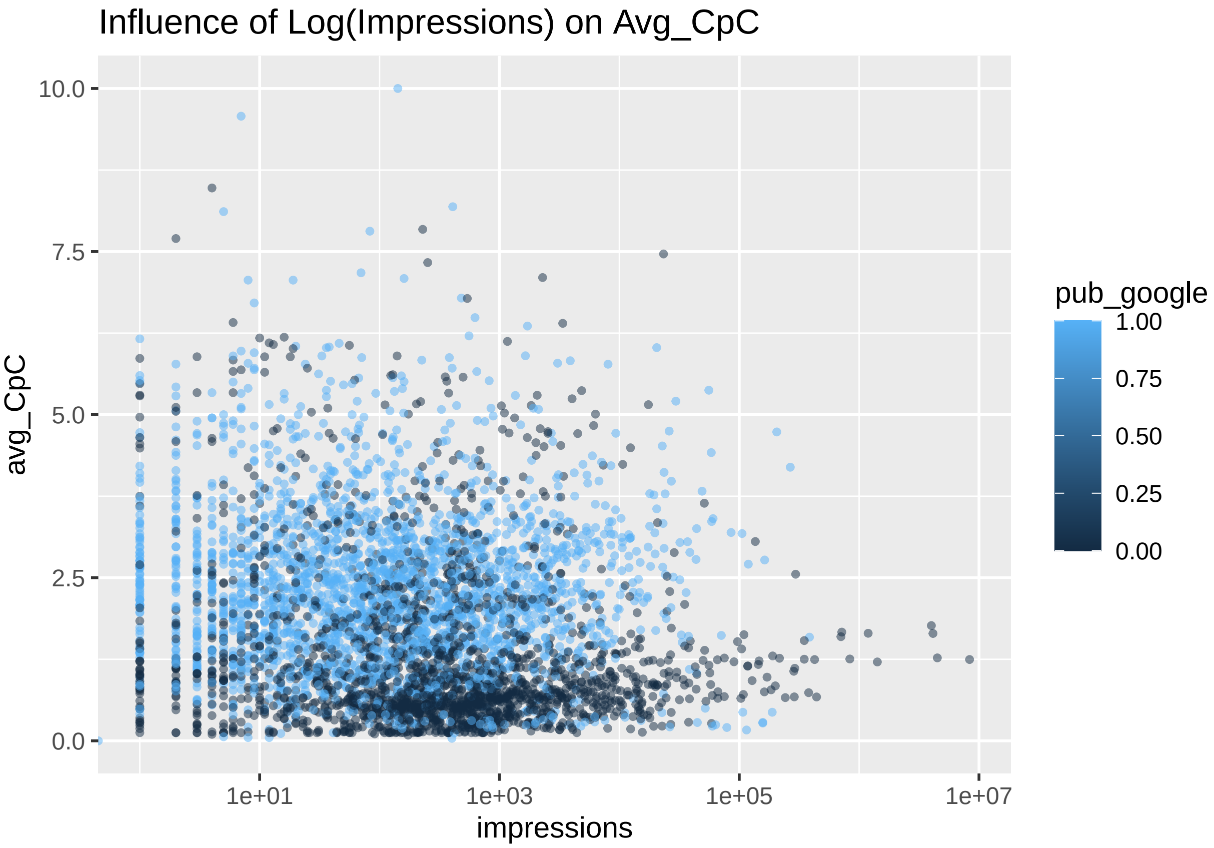
Exploratory analysis

```r
# For different publishers
af_data$pub_google <- as.numeric(grepl(pattern = ".*Google.*", x = af_data$publisher_name))
af_data$pub_msn <- as.numeric(grepl(pattern = ".*MSN.*", x = af_data$publisher_name))
af_data$pub_overture <- as.numeric(grepl(pattern = ".*Overture.*", x = af_data$publisher_name))
af_data$pub_yahoo <- as.numeric(grepl(pattern = ".*Yahoo.*", x = af_data$publisher_name))

# For different keywords
af_data$key_af <- as.numeric(grepl(pattern = ".*air.*france.*", x = af_data$keyword))
af_data$key_cheap <- as.numeric(grepl(pattern = ".*cheap.*", x = af_data$keyword))

#creating dummy variable for campaigns that were geo targeted
af_data$camp_geo <- as.numeric(grepl(pattern = ".*Geo.*Targeted.*", x = af_data$campaign))
af_data$camp_fr <- as.numeric(grepl(pattern = ".*Air.*France.*", x = af_data$campaign))

#creating dummy variable for keyword group
af_data$keygp_toP <- as.numeric(grepl(pattern = ".*to.*Paris.*", x=af_data$keyword_group))
af_data$keypg_fr <- as.numeric(grepl(pattern = ".*France.*", x = af_data$keyword_group))

# creating dummy variables for match_type
af_data$match_broad <- as.numeric(grepl(pattern = ".*Broad.*", x = af_data$match_type))
```

To deepen the insights of our exploratory analysis, certain categorical variables need to be converted to dummy variables so they can be used to explore their impact other variables and later potential business success.

```r
# , message=FALSE, warning=FALSE, include=FALSE
#Scatter of Impressions and avg_CPC
#my_scatter <-
ggplot (af_data, aes (impressions,avg_CpC,
                                    color = pub_google )) +
                 geom_jitter (alpha = 0.5, shape=20) +
                 scale_x_log10 () +
                 labs(title="Influence of Log(Impressions) on Avg_CpC") +
                 theme_grey ()
```

```
## Warning: Transformation introduced infinite values in continuous x-axis
```



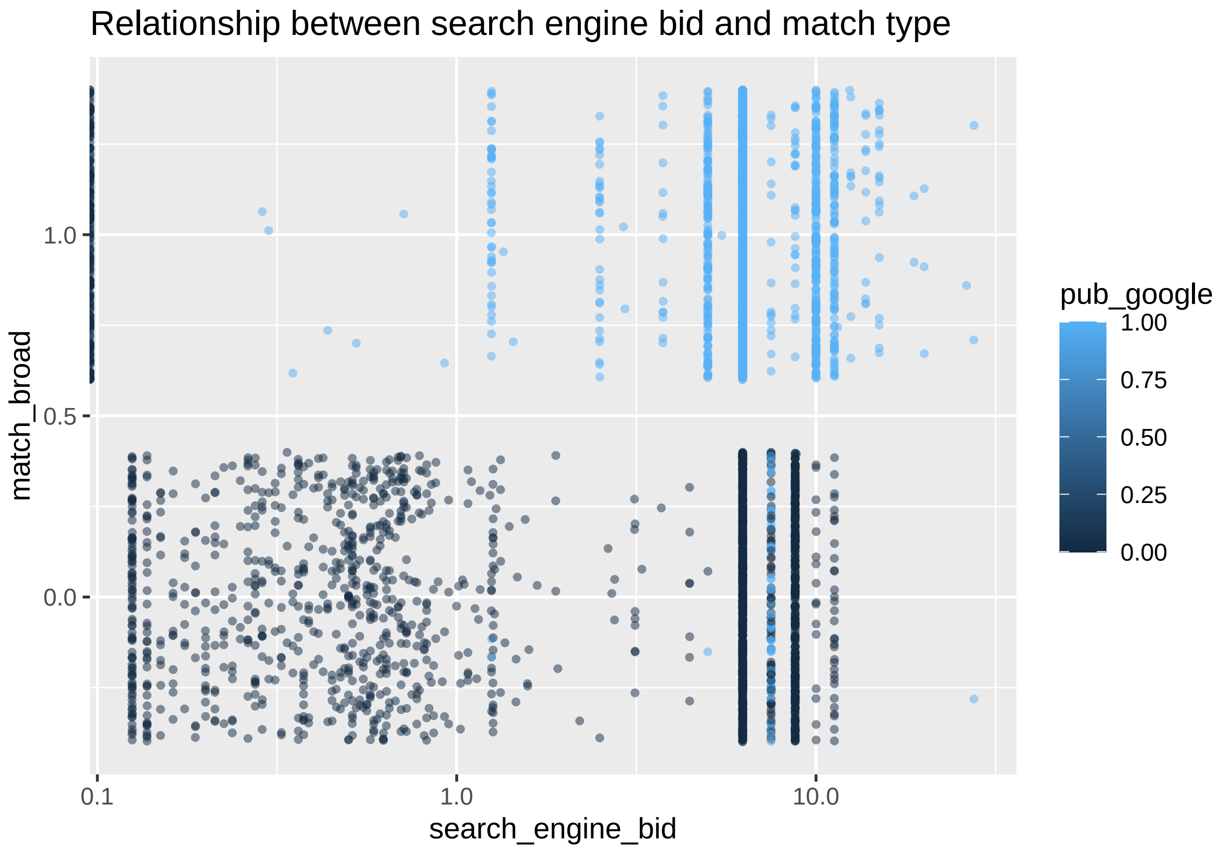
```
#ggplotly(my_scatter)
```

Scatter analysis: Influence of Log(Impressions) on Avg__CpC

There is a trend for the amount of impressions in regard to the average cost per click. For non-Google search engines, we found that the more impressions made led to an increase in the average cost per click. This means that it is more expensive for non-Google search engines to have more impressions. Meanwhile, Google does not seem to be affected regardless of the amount of impressions. When taking into consideration the ROI, what does this mean?

```
#Scatter of search engine bid and match broad
#my_vis_scatter <-
  ggplot (af_data, aes (search_engine_bid,match_broad,
                                              color = pub_google )) +
   geom_jitter (alpha = 0.5, shape=20) +
   scale_x_log10 () +
   labs(title="Relationship between search engine bid and match type") +
 theme_grey ()
```

```
## Warning: Transformation introduced infinite values in continuous x-axis
```



```
#ggplotly(my_vis_scatter)
```

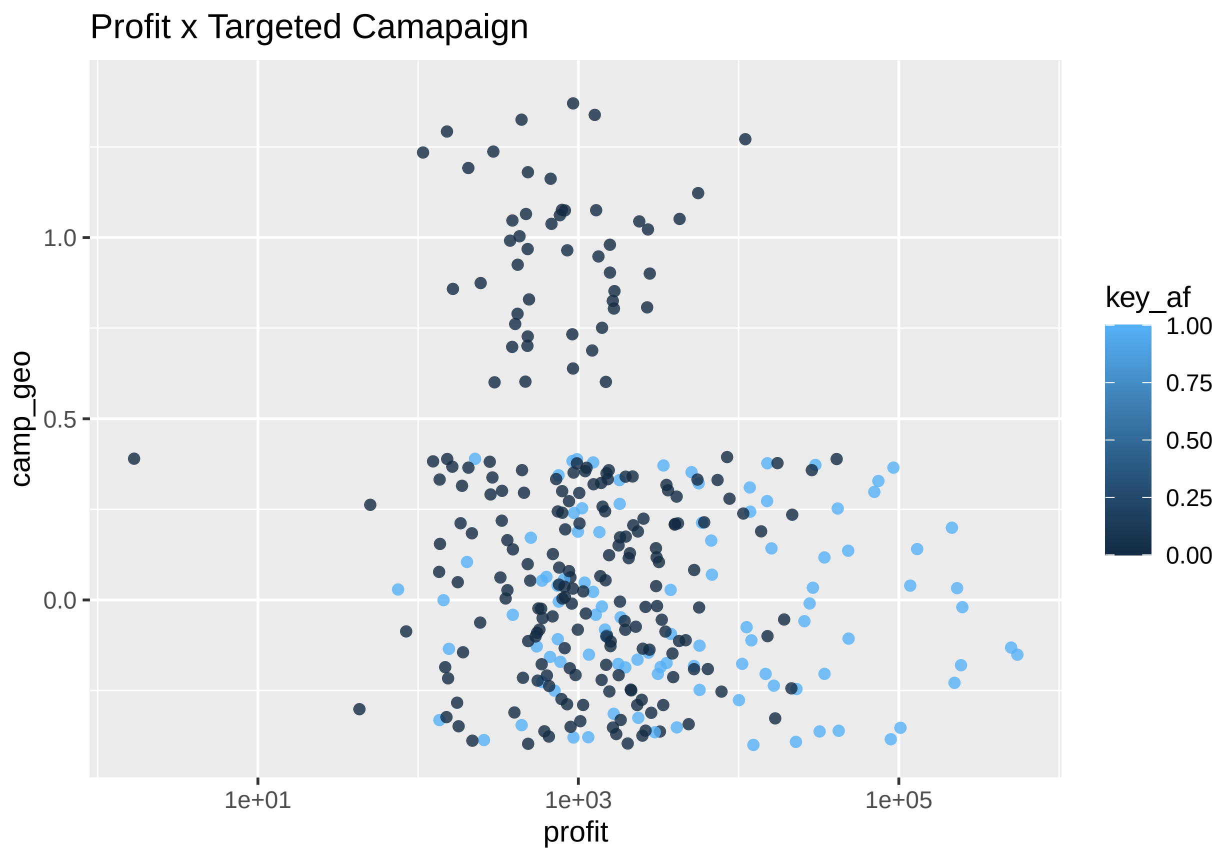
Scatter analysis: Relationship between search engine bid and match type

From this analysis, it appears that match broad only run by Google and tend to be more expensive on average. Since our goal is to keep cost low, the question to keep in mind is "Is this worth it?" This question will be examined using predictive analysis methodologies.

```
#Creating a metric to plot profit
af_data$profit <- af_data$amount-af_data$total_cost
#Scatter of Profit and Camp_Geo of Key_af
#Fig3<-
ggplot(af_data, aes (profit, camp_geo,
   color= key_af)) +
 geom_jitter (alpha = 0.8, shape=19) +
 scale_x_log10 () +
 labs(title="Profit x Targeted Camapaign") +
 theme_grey ()
```

```
## Warning in self$trans$transform(x): NaNs produced
```

```
## Warning: Transformation introduced infinite values in continuous x-axis
## Warning: Removed 4186 rows containing missing values (geom_point).
```



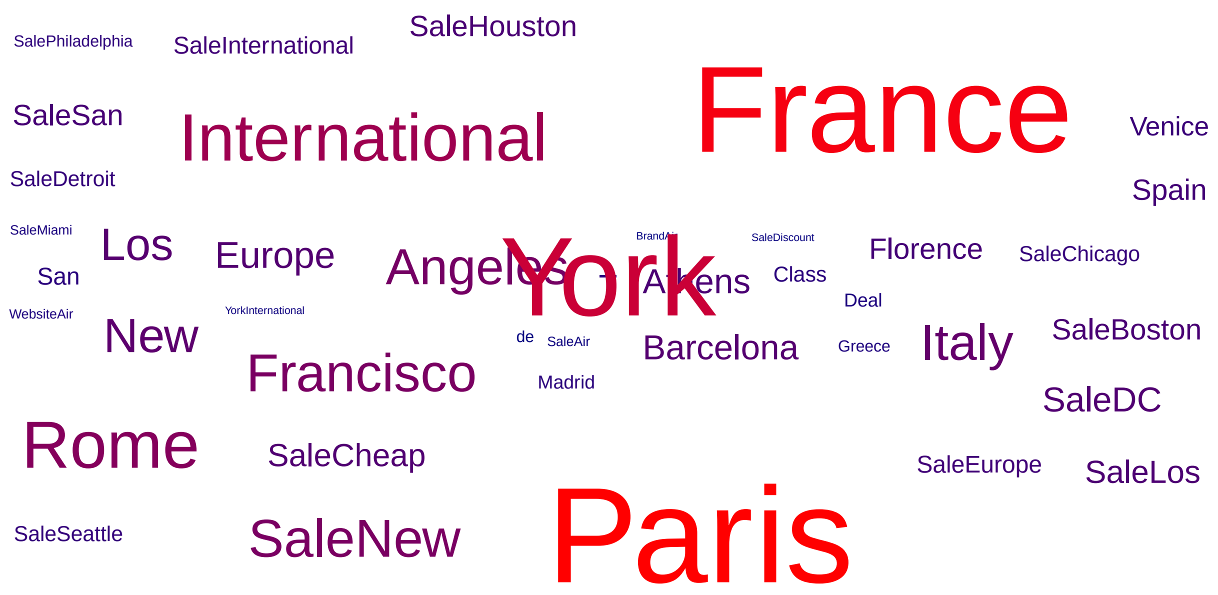
```
#ggplotly(Fig3)
```

Scatter analysis: Profit x Targeted Camapaign

Since Air France is pursuing an international growth strategy and wants to enter the highly competitive US market we wanted to look at the geo-targeted campaigns. From this analysis we can see that the geo-target campaigns do not appear to have much influence. The campaigns that have Air France in the keyword have an overall positive impact on the profit. However, from our analysis we found that Air France is used only in non geo targeted campaigns. Wether a combination of the keyword "Air France" and geo targeting could result in higher profits will be examined in this report.

```
temp <- paste(af_data$keyword_group, collapse = "")
temp <- strsplit(temp, " ")[[1]]
new_df <- as.data.frame(table(temp))
new_df <- new_df %>%
 filter(Freq > 10 & Freq < 500)

#plotting the wordcloud
set.seed(657)
ggplot(new_df, aes(label=temp, size= Freq, color= Freq)) +
geom_text_wordcloud_area(eccentricity = 0.35) +
scale_size_area(max_size = 18) +
theme_minimal()+
scale_color_gradient(low= "navy", high = "red")
```

```
## Warning in wordcloud_boxes(data_points = points_valid_first, boxes = boxes, :
## One word could not fit on page. It has been placed at its original position.
```



Word Cloud analysis: Keyword group

We developed a word cloud as a visual representation to show the words that are most frequently searched. As seen in the word cloud there are quite a few words that are larger than others. In terms of our analysis, we can easily visualize what words are being searched at a higher frequency and test impacts of those words on the business goals.

```
# creating a function to normalize data
my_normal <- function(x){
  my_min <- min(x, na.rm=T)
  my_max <- max(x, na.rm=T)
  normalized <- (x - my_min)/(my_max - my_min)
  return(normalized)
} # end of my_normal function

# using the normalize function on af_data
for(i in 1:23){
  af_data <- as.data.frame(af_data)
  if(is.numeric(af_data[ , i]) == TRUE){
    af_data$normalized <- my_normal(af_data[ , i])
    cname <- af_variables[i]
    column_name <- paste(cname,"_norm",sep="")
    af_data <- plyr::rename(af_data, c("normalized" = column_name))
  }
}#closing i-loop
```

All the numerical variables need to be normalized as they are on very different scales (with average position ranging from 0 to 15 and click charges from 0 to 46,188.44).

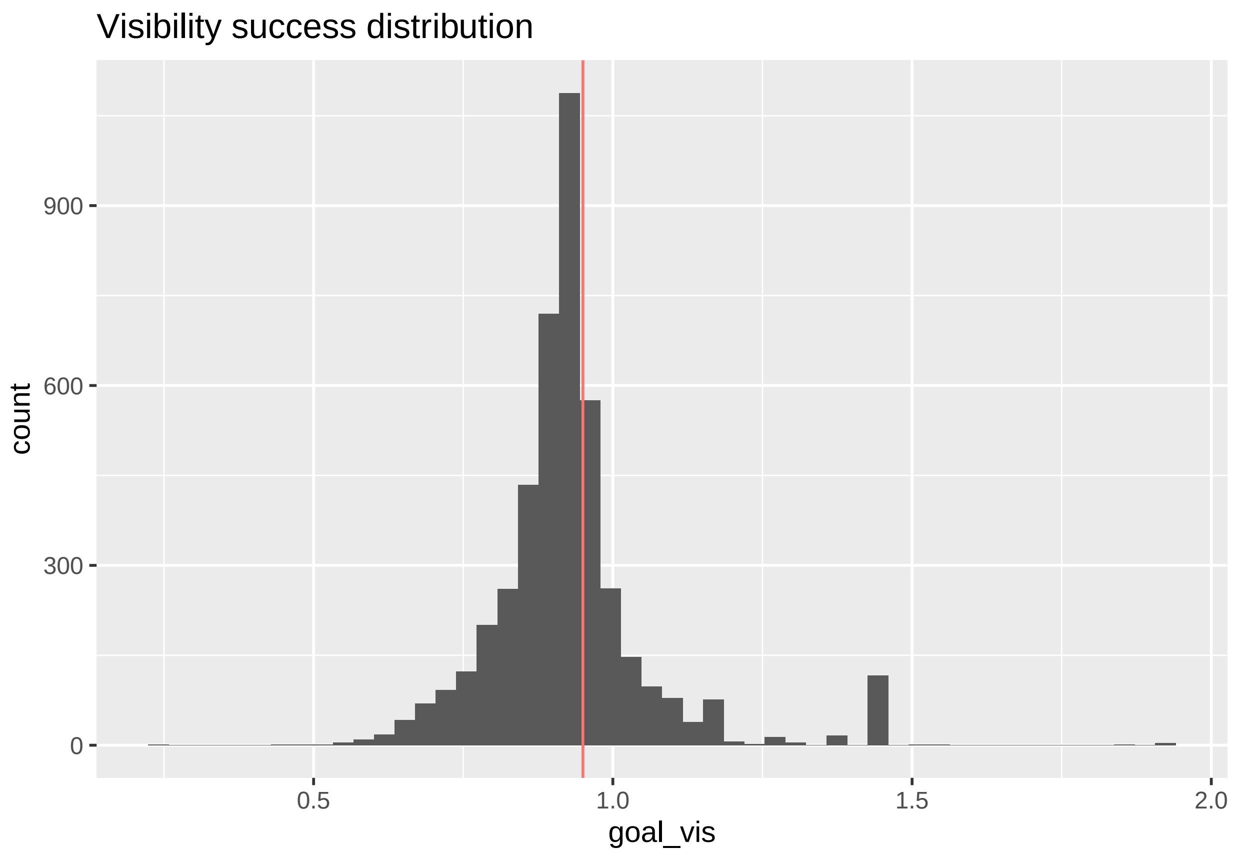
Predictive analysis

In order to understand what campaigns are successful and which aren't, we need to define what business success means. For Media Contacts, with regards to their account with Air France, this means attracting more people to Air France's website by improving their visibility, as well as transforming those visits into quality sales. As such, two success metrics have been developed as follows:

- Improving visibility: observations are considered successful when the click through rate is high, and the ad has a high position. Position is particularly key as it takes into account the quality of the ads and business landing page, as well as other metrics.
- Capturing return on invest: campaigns need to do more than just create visibility. The return on the investment also plays an important role in the determination of business success of a campaign. Hence, the features total cost per campaign and amount allow for assurance of a successful campaign. Since campaigns with no bookings cannot generate good return on invest, observations with 0 completed bookings are taken out for further analysis of this goal.

Increasing visibility

```
# Creating a new dataframe to work off
af_vis <- af_data

# Setting the parameters to determine success based on increasing visibility
af_vis$avg_pos_flip <- -1*af_vis$avg_pos  # flipping values so highest number is best outcome
af_vis$avg_pos_flip_norm <- my_normal(af_vis$avg_pos_flip)
af_vis$goal_vis <- af_vis$engine_click_thru_norm + af_vis$avg_pos_flip_norm

# Visualizing distribution to find a cutoff point beyond which observations will be successful
ggplot (af_vis, aes(goal_vis)) +
  geom_histogram(bins = 50) +
  geom_vline(aes(xintercept = 0.95, color = "red"), show.legend = FALSE) +
  labs (title = "Visibility success distribution")
```



```
# Creating goal_vis_binary to categorize each observation as success or failure
for(i in 1:nrow(af_vis)){
  if(af_vis$goal_vis[i] >= 0.95){
    af_vis$goal_vis_binary[i] <- 1
  }else{
    af_vis$goal_vis_binary[i] <- 0
  }
} # end of for i_loop
```

Observations are considered successful with regards to increasing the visibility of Air France on the market when: normalized engine click through rate + normalized average position >= 0.95 This limit was determined based on the distribution as seen in the above figure.

```
# using stratified sampling

# Creating a list with both datasets
set.seed (1212)
training_testing_vis <- stratified(as.data.frame(af_vis), group= 51, size=0.8, bothSets = T)

# Extracting the datasets from the list
train_vis <- training_testing_vis$SAMP1
test_vis <- training_testing_vis$SAMP2

table (train_vis$goal_vis_binary)
```

```
##
```

```
##    0    1
## 2566 1042
```

The data was then sampled with a stratified method in order to ensure the proportion of business successes to failures in the sample was representative of the population

```r
# creating a logistic regression
vis_logit <- glm(formula = goal_vis_binary ~ search_engine_bid +
                    key_af + key_cheap +
                   pub_google + match_broad + camp_geo,
                 data = train_vis, family = "binomial")
summ_vis <- summary (vis_logit)
summ_vis
```

```
##
## Call:
## glm(formula = goal_vis_binary ~ search_engine_bid + key_af +
##     key_cheap + pub_google + match_broad + camp_geo, family = "binomial",
##     data = train_vis)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -2.1759  -0.7120  -0.4550   0.9297   2.9423
##
## Coefficients:
##                   Estimate Std. Error z value Pr(>|z|)
## (Intercept)       -4.08734    0.17582 -23.247  < 2e-16 ***
## search_engine_bid  0.29942    0.02067  14.488  < 2e-16 ***
## key_af             2.23670    0.16366  13.667  < 2e-16 ***
## key_cheap          0.58310    0.10657   5.471 4.46e-08 ***
## pub_google        -2.09945    0.24532  -8.558  < 2e-16 ***
## match_broad        2.53760    0.24345  10.424  < 2e-16 ***
## camp_geo           1.80987    0.09290  19.482  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 4337.4  on 3607  degrees of freedom
## Residual deviance: 3403.4  on 3601  degrees of freedom
## AIC: 3417.4
##
## Number of Fisher Scoring iterations: 5
```

```r
# creating a logistic regression with normalized data
vis_logit_norm <- glm(formula = goal_vis_binary ~ search_engine_bid_norm +
                    key_af + key_cheap +
                   pub_google + match_broad + camp_geo,
                 data = train_vis, family = "binomial")
summary (vis_logit_norm)
```

```
##
## Call:
```

```
## glm(formula = goal_vis_binary ~ search_engine_bid_norm + key_af +
##     key_cheap + pub_google + match_broad + camp_geo, family = "binomial",
##     data = train_vis)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -2.1759  -0.7120  -0.4550   0.9297   2.9423
##
## Coefficients:
##                        Estimate Std. Error z value Pr(>|z|)
## (Intercept)             -4.0873     0.1758 -23.247  < 2e-16 ***
## search_engine_bid_norm   8.2340     0.5683  14.488  < 2e-16 ***
## key_af                   2.2367     0.1637  13.667  < 2e-16 ***
## key_cheap                0.5831     0.1066   5.471 4.46e-08 ***
## pub_google              -2.0995     0.2453  -8.558  < 2e-16 ***
## match_broad              2.5376     0.2434  10.424  < 2e-16 ***
## camp_geo                 1.8099     0.0929  19.482  < 2e-16 ***
## ---
## Signif. codes:  0 ’***’ 0.001 ’**’ 0.01 ’*’ 0.05 ’.’ 0.1 ’ ’ 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 4337.4  on 3607  degrees of freedom
## Residual deviance: 3403.4  on 3601  degrees of freedom
## AIC: 3417.4
##
## Number of Fisher Scoring iterations: 5
```

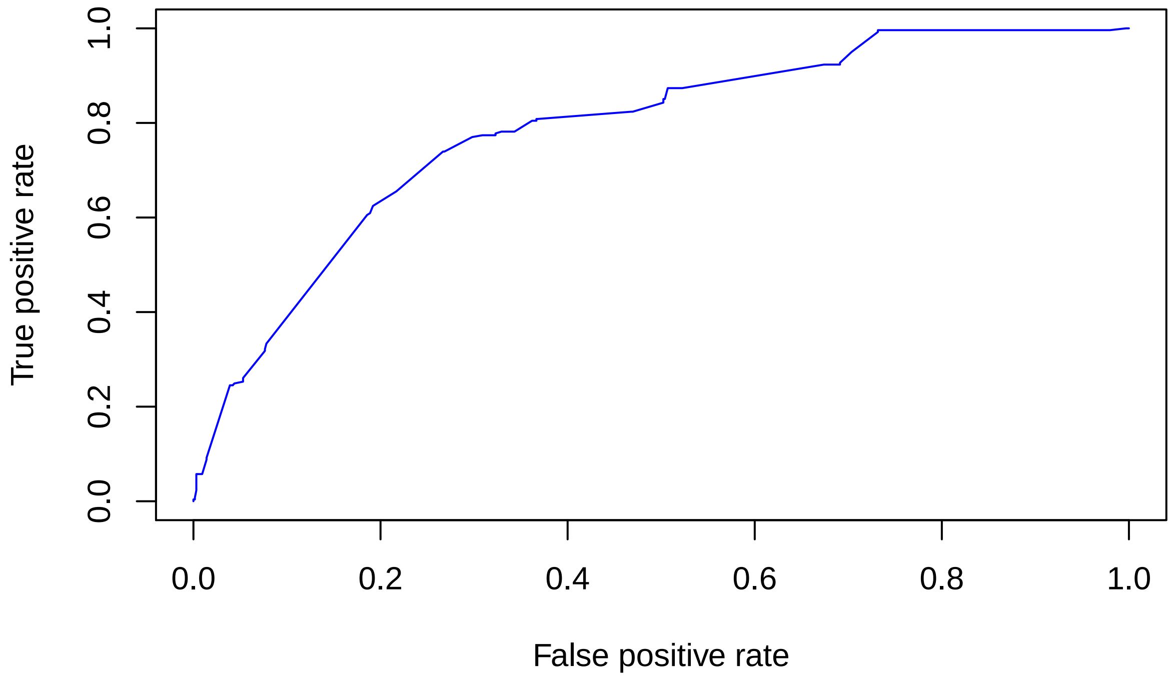
Understanding the model

All the variables in our model are statistically significant as their p-values are <0.05. As such, if testing on our model determines it to be reliable, we will be able to come to conclusions at a 95% confidence level. Furthermore, by looking at the coefficients, we can see that observations coming from Google are -87.7475815% less likely to have delivered business success. This could be explained by the fact that Google is the more popular search engine relative to the others used, and so Air France is facing more competition for the attention of it’s customers there than elsewhere. The largest impact however comes from the search engine bid Media Contacts set, with a 1 USD increase leading to a 34.9071096% increase in the odds of business success.

```r
vis_predict <- predict(vis_logit, test_vis, type="response")

confusionMatrix(data = as.factor(as.numeric(vis_predict>0.5)),
                reference=as.factor(as.numeric(test_vis$goal_vis_binary)))
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction   0   1
##          0 520 102
##          1 121 159
##
##                Accuracy : 0.7528
##                  95% CI : (0.7233, 0.7806)
##      No Information Rate : 0.7106
##      P-Value [Acc > NIR] : 0.00263
##
##                    Kappa : 0.4115
##
##   Mcnemar’s Test P-Value : 0.22806
##
##              Sensitivity : 0.8112
##              Specificity : 0.6092
##           Pos Pred Value : 0.8360
##           Neg Pred Value : 0.5679
##               Prevalence : 0.7106
##           Detection Rate : 0.5765
##     Detection Prevalence : 0.6896
##        Balanced Accuracy : 0.7102
##
##         ’Positive’ Class : 0
##
```

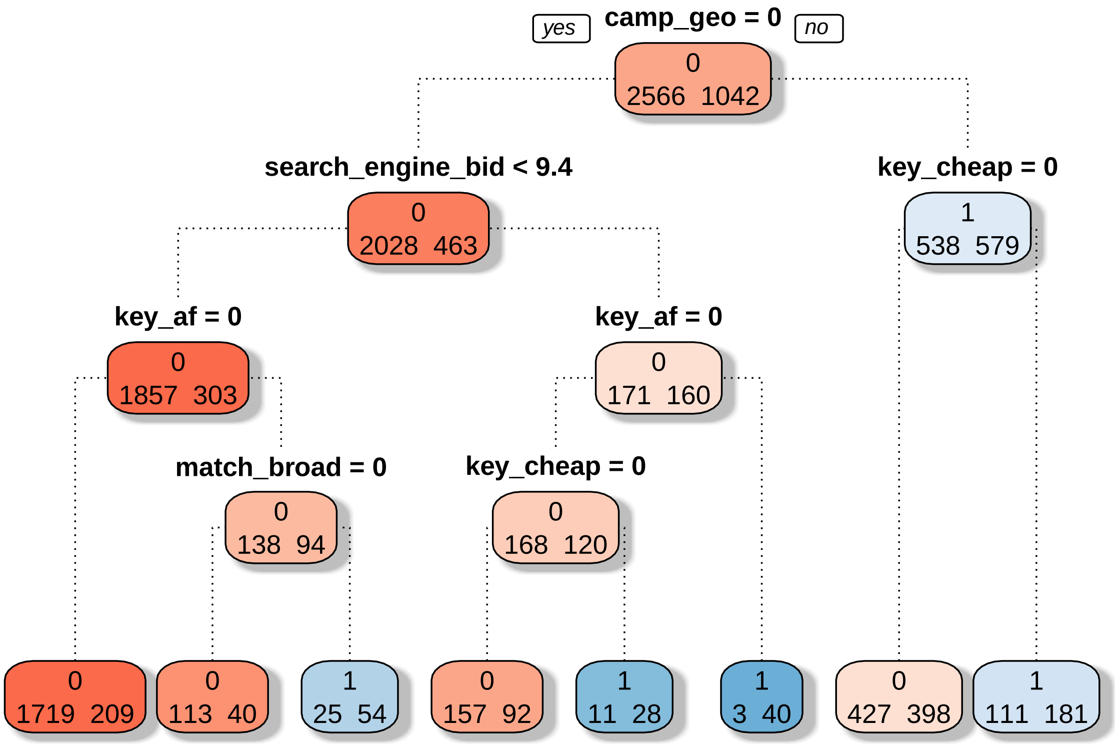
```r
#ROCS does not understand predict function
pred_vis_logit <- prediction(vis_predict, test_vis$goal_vis_binary)

#running 20-30- confusion matrices for different levels of p (threshold)
perf_logit_vis <- performance(pred_vis_logit, "tpr", "fpr")

plot(perf_logit_vis, col="blue")
```



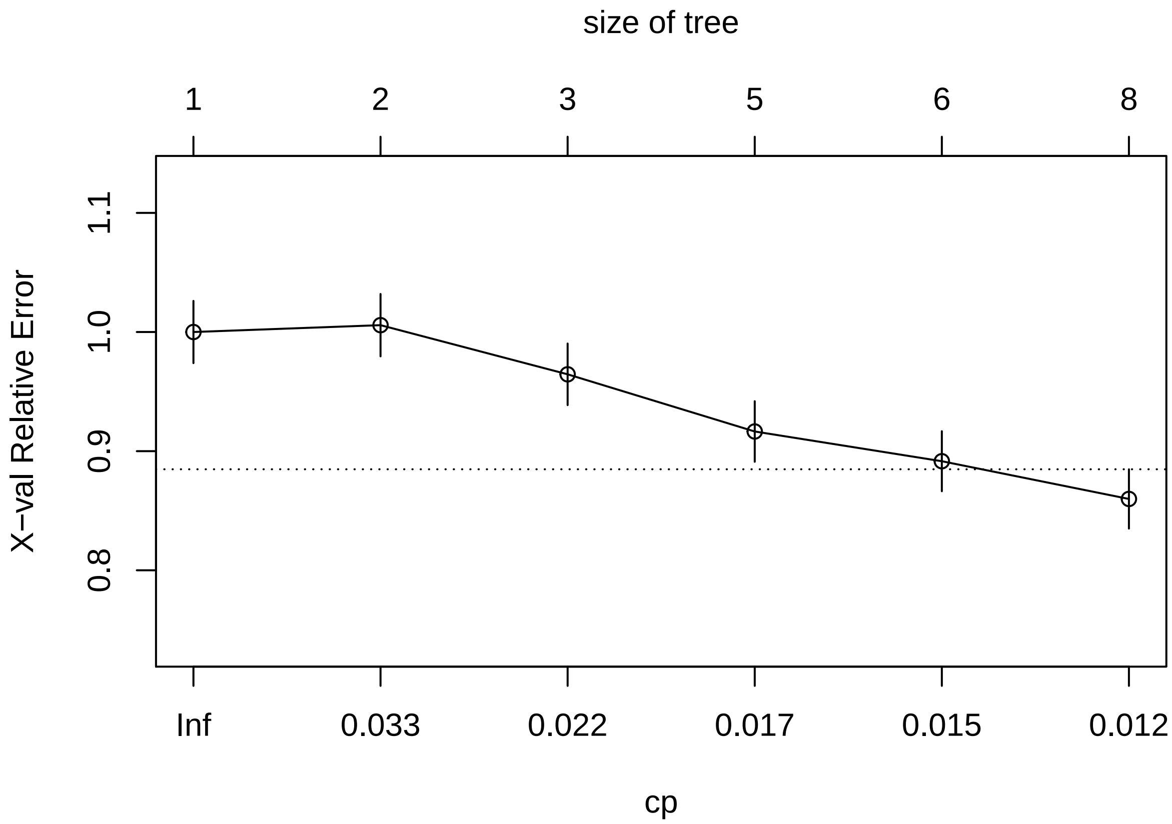
Evaluating the model

This model has an accuracy rate of 69.05%, with a sensitivity of 75% and a specificity of 59%. Furthermore, the ROC curve, as shown in the above figure, is above the pig without getting too closer to the top left corner of the chart (which would speak to overfitting). This leads us conclude that this model is robust and can be used for predictive analysis on new data. But one last question remains: is this the best model?

```
# using a GINI decision tree to analyze the data
vis_prediction_tree <- rpart(formula = goal_vis_binary ~ search_engine_bid +
                                 key_af + key_cheap + pub_google + match_broad + camp_geo,
                                  data = train_vis, method = "class", cp = 0.011)

rpart.plot(vis_prediction_tree, type=1, extra = 1,
           box.palette = "RdBu",
           branch.lty = 3, shadow.col = "gray")
```



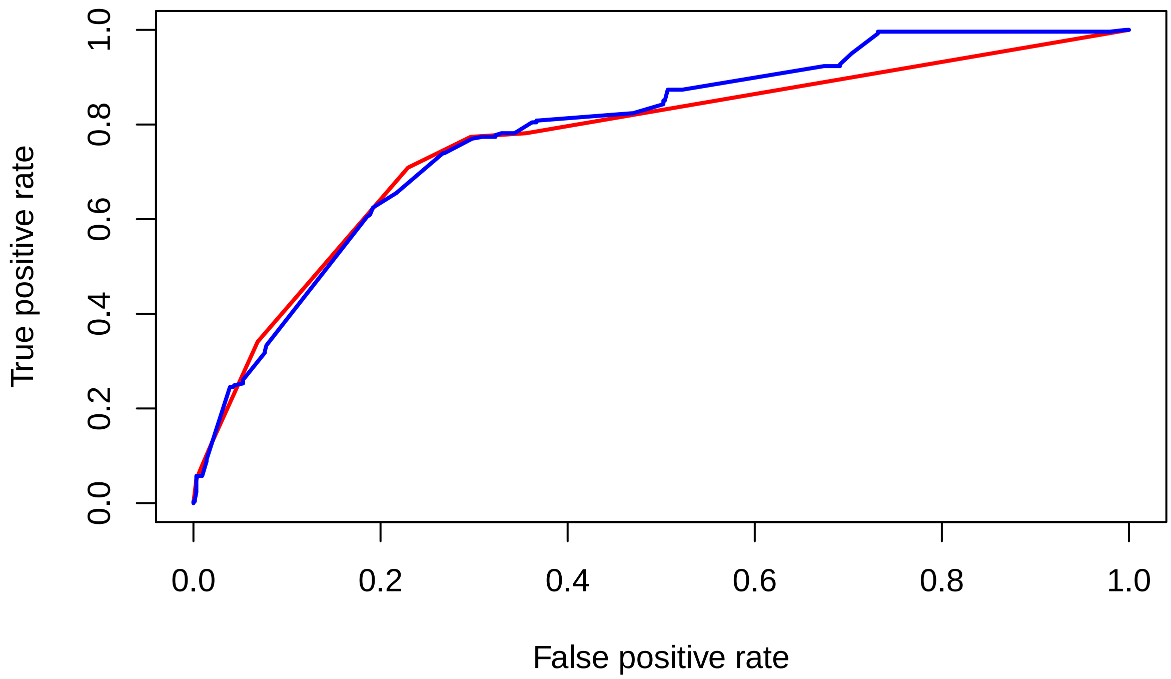
```
plotcp(vis_prediction_tree)
```



Gini Decision Tree

Above we have our GINI decision tree, using the same formula as our logistic regression. The main splitting variable is whether or not the observation was part of a geo targeted campaign. So to answer our question, is model preferrable to the logistic regression?

```r
#prediction
vis_prediction_tree_predict <- predict(vis_prediction_tree, test_vis, type = "prob")

#prepare for AUCROC
vis_prediction_tree_prediction <- prediction(vis_prediction_tree_predict[ , 2], test_vis$goal_vis_binar

#performance
vis_prediction_tree_performance <- performance(vis_prediction_tree_prediction, "tpr", "fpr")

plot(vis_prediction_tree_performance, col = "red", lwd = 2)
plot(perf_logit_vis, col = "blue", lwd = 2, add = TRUE)
```

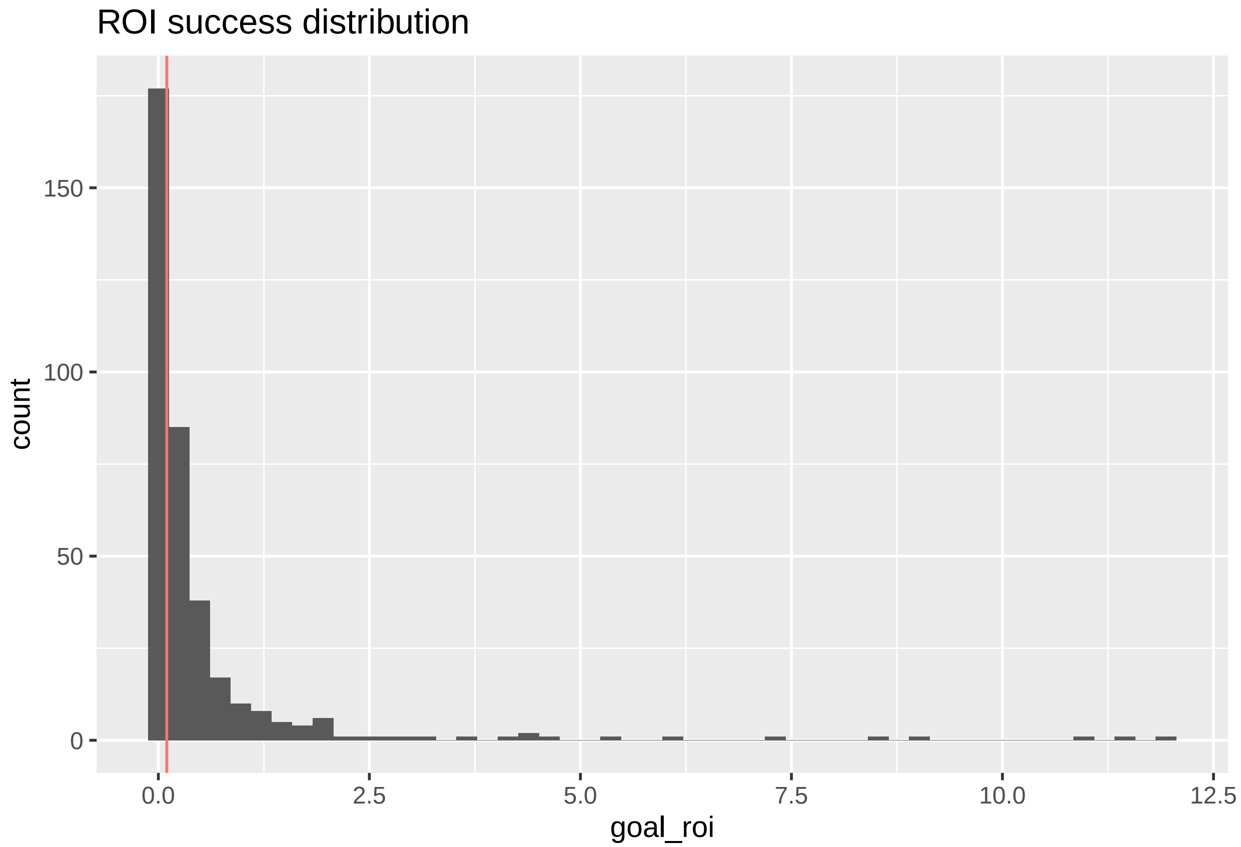


Comparing models

When overlapping the two ROC curves, we can visually see that while they are overlapping a lot of the time, the blue curve, representing the logistic regression model, has a higher AUC. As such, this is the model we would select to use for predictive analysis going forward.

Increasing ROI

```
af_roi <- af_data %>%
  filter (total_vol_bookings > 0)


# Setting the parameters to determine success based on increasing visibility
af_roi$goal_roi <- af_roi$total_cost/(af_roi$amount)
af_roi$goal_roi <- gsub ("NaN", 0, af_roi$goal_roi)
af_roi$goal_roi <- as.numeric(af_roi$goal_roi)

# Visualizing distribution to find a cutoff point beyond which observations will be successful
ggplot (af_roi, aes(goal_roi)) +
  geom_histogram(bins = 50) +
  geom_vline(aes(xintercept = 0.1, color = "red"), show.legend = FALSE) +
  labs (title = "ROI success distribution")
```



```r
# Creating goal_vis_binary to categorize each observation as success or failure
for(i in 1:nrow(af_roi)){
  if(af_roi$goal_roi[i] <= 0.1){
    af_roi$goal_roi_binary[i] <- 1
  }else{
    af_roi$goal_roi_binary[i] <- 0
  }
} # end of for i_loop

table (af_roi[ ,49])
```

```
##
##   0   1
## 201 167
```

The plot shows the distribution of the ROI success variable in a histogram. The lower the score, the better the campaign. To convert the continuous variable into a binary one for logistic regression and a clear cut of which ones can be considered successful and which ones not, the cut-off point needs to be decided on. A score of 0.1 or less (marked with the red line) seems like a good fit.

```r
# Creating a list with both datasets
set.seed (1212)
training_testing_roi <- stratified(as.data.frame(af_roi), group= 49, size=0.8, bothSets = T)

# Extracting the datasets from the list
```

```
train_roi <- training_testing_roi$SAMP1
test_roi <- training_testing_roi$SAMP2

table (test_roi$goal_roi_binary)
```

```
##
##  0  1
## 40 33
```

To keep the proportions of the successful and unsuccessful campaigns unchanged, stratified sampling is conducted. As only campaigns with positive bookings are regarded, the sample size is limited to 368 observations. A 80-20 sampling is conducted, separating 80% of the observations into a training dataset and the remaining 20% into a testing dataset.

```
# creating a logistic regression
roi_logit <- glm(formula = goal_roi_binary ~ search_engine_bid + clicks + avg_CpC + impressions+ +
                    engine_click_thru + avg_pos + amount + total_cost + total_vol_bookings +
                    pub_google + pub_msn + pub_overture + key_af + key_cheap + camp_geo +
                    keygp_toP + match_broad,
                 data = train_roi, family = "binomial")
```

```
## Warning: glm.fit: algorithm did not converge
```

```
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred
```

```
# creating a logistic regression  # key_cheap + engine_click_thru + match_broad
roi_logit <- glm(formula = goal_roi_binary ~ avg_CpC + impressions +
                       key_af + camp_geo,
                    data = train_roi, family = "binomial")
```

```
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred
```

```
summ_roi <- summary (roi_logit)
summary (roi_logit)
```

```
##
## Call:
## glm(formula = goal_roi_binary ~ avg_CpC + impressions + key_af +
##     camp_geo, family = "binomial", data = train_roi)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -2.4503  -0.6024  -0.1958   0.4278   2.4457
##
## Coefficients:
##               Estimate Std. Error z value Pr(>|z|)
## (Intercept)  1.076e+00  4.083e-01   2.637  0.00837 **
## avg_CpC     -1.335e+00  2.250e-01  -5.933 2.97e-09 ***
## impressions -7.824e-06  3.029e-06  -2.583  0.00980 **
## key_af       2.512e+00  4.319e-01   5.815 6.07e-09 ***
## camp_geo     2.449e+00  4.566e-01   5.365 8.12e-08 ***
## ---
## Signif. codes:  0 ’***’ 0.001 ’**’ 0.01 ’*’ 0.05 ’.’ 0.1 ’ ’ 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 406.48  on 294  degrees of freedom
## Residual deviance: 230.44  on 290  degrees of freedom
## AIC: 240.44
##
## Number of Fisher Scoring iterations: 7
```

```
exp(-1.335)-1 # avg_CpC
```

```
## [1] -0.7368418
```

```
exp(-0.000007824)-1 # impressions
```

```
## [1] -7.823969e-06
```

```
exp(2.512)-1 # key_af
```

```
## [1] 11.32956
```

```
exp(2.449)-1 # geo_targeted
```

```
## [1] 10.57676
```

```
#creating logistic regression on normalized data
roi_logit <- glm(formula = goal_roi_binary ~ avg_CpC_norm + impressions_norm +
                       key_af + camp_geo,
                    data = train_roi, family = "binomial")
```

```
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred
```

```
summary (roi_logit)
```

```
##
## Call:
## glm(formula = goal_roi_binary ~ avg_CpC_norm + impressions_norm +
##     key_af + camp_geo, family = "binomial", data = train_roi)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -2.4503  -0.6024  -0.1958   0.4278   2.4457
##
## Coefficients:
##                    Estimate Std. Error z value Pr(>|z|)
## (Intercept)         1.0764     0.4083   2.637  0.00837 **
## avg_CpC_norm     -13.3502     2.2501  -5.933 2.97e-09 ***
## impressions_norm -65.2728    25.2717  -2.583  0.00980 **
## key_af             2.5116     0.4319   5.815 6.07e-09 ***
## camp_geo           2.4494     0.4566   5.365 8.12e-08 ***
## ---
## Signif. codes:  0 ’***’ 0.001 ’**’ 0.01 ’*’ 0.05 ’.’ 0.1 ’ ’ 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 406.48  on 294  degrees of freedom
## Residual deviance: 230.44  on 290  degrees of freedom
## AIC: 240.44
##
## Number of Fisher Scoring iterations: 7
```

Understanding the model

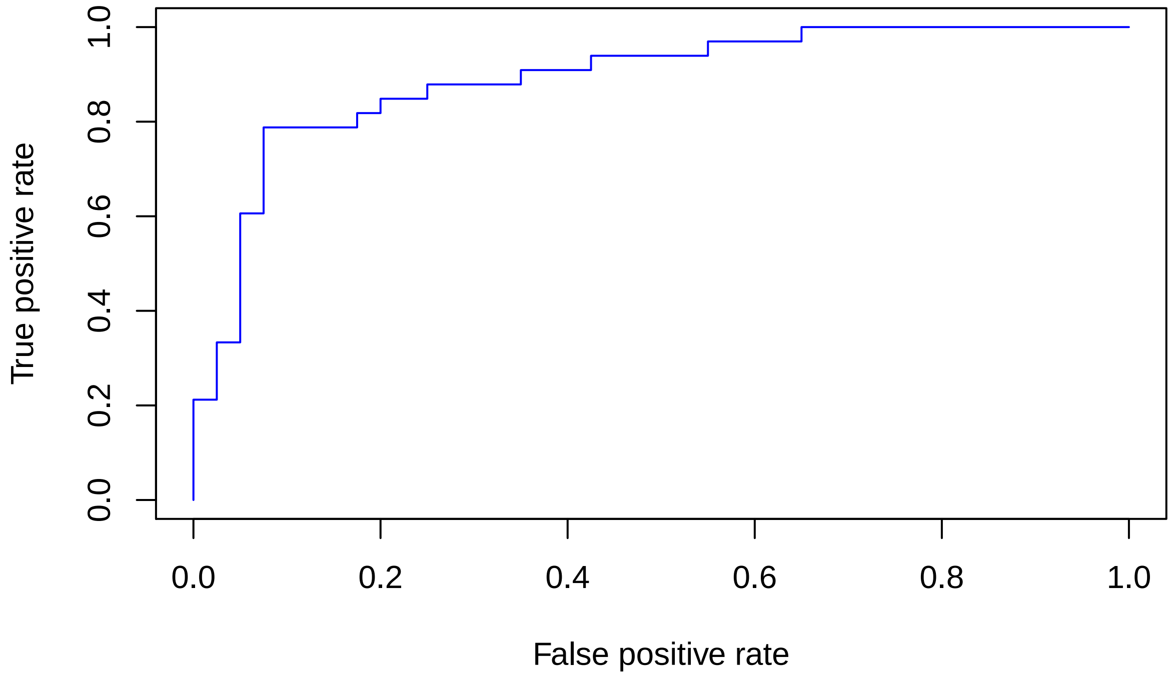
A logistic regression model is created to analyze the most important determinants of the ROI business goal. With a confidence of 95%, the features Average Cost per click, Impressions, Air France as a keyword and Geo Targeted campaigns, are statistically significantly influencing business success. With a one unit increase in the feature average cost per click, the odds of business success will decrease by 74%. Moreover, one more impression decrease the odds of business success by far less than 1%.

After a normalization of the independent variables, the sizes of impact between the variables can be compared. It appears that Impressions have the highest negative impact while the keyword Air France has the highest positive impact on the odds of business success.

```r
## testing the logistic regression
roi_predict <- predict(roi_logit, test_roi, type="response")

confusionMatrix(data = as.factor(as.numeric(roi_predict>0.5)),
                reference=as.factor(as.numeric(test_roi$goal_roi_binary)))
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction  0  1
##          0 38 13
##          1  2 20
##
##                Accuracy : 0.7945
##                  95% CI : (0.6838, 0.8802)
##     No Information Rate : 0.5479
##     P-Value [Acc > NIR] : 9.793e-06
##
##                   Kappa : 0.5728
##
##  Mcnemar’s Test P-Value : 0.009823
##
##             Sensitivity : 0.9500
##             Specificity : 0.6061
##          Pos Pred Value : 0.7451
##          Neg Pred Value : 0.9091
##              Prevalence : 0.5479
##          Detection Rate : 0.5205
##    Detection Prevalence : 0.6986
##       Balanced Accuracy : 0.7780
##
##        'Positive' Class : 0
##
```

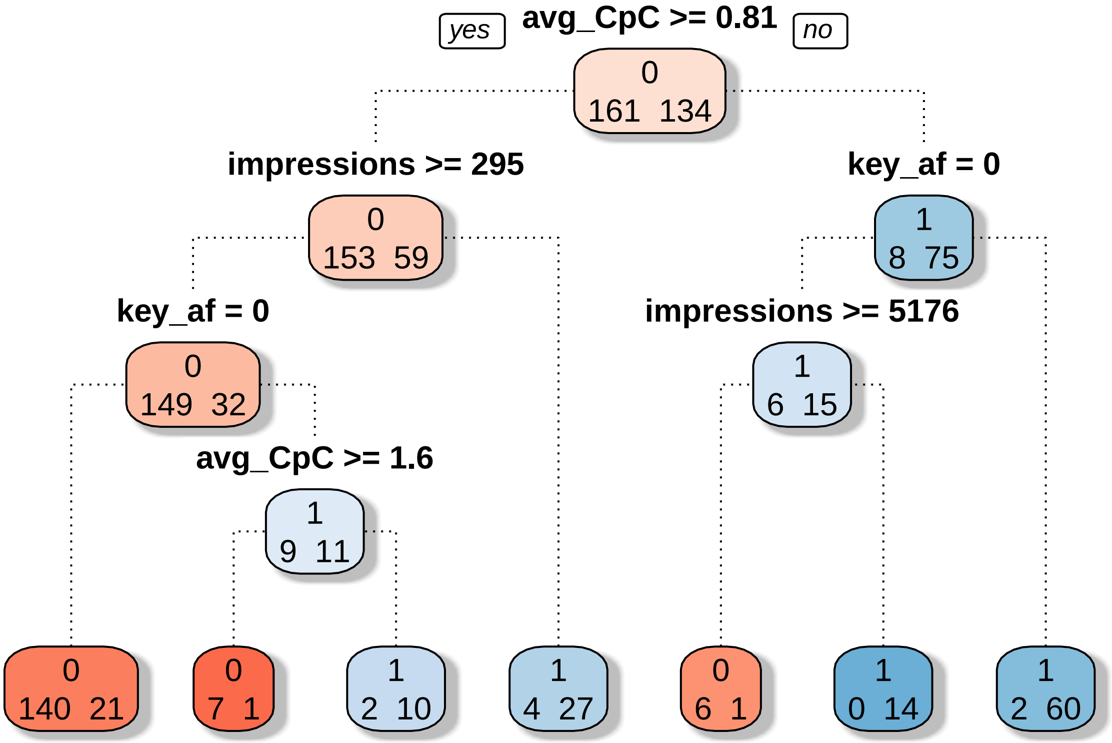
```
#ROCS does not understand predict function
pred_roi_logit <- prediction(roi_predict, test_roi$goal_roi_binary)

#running 20-30- confusion matrices for different levels of p (threshold)
perf_logit_roi <- performance(pred_roi_logit, "tpr", "fpr")

plot(perf_logit_roi, col="blue", ldw = 2)
```



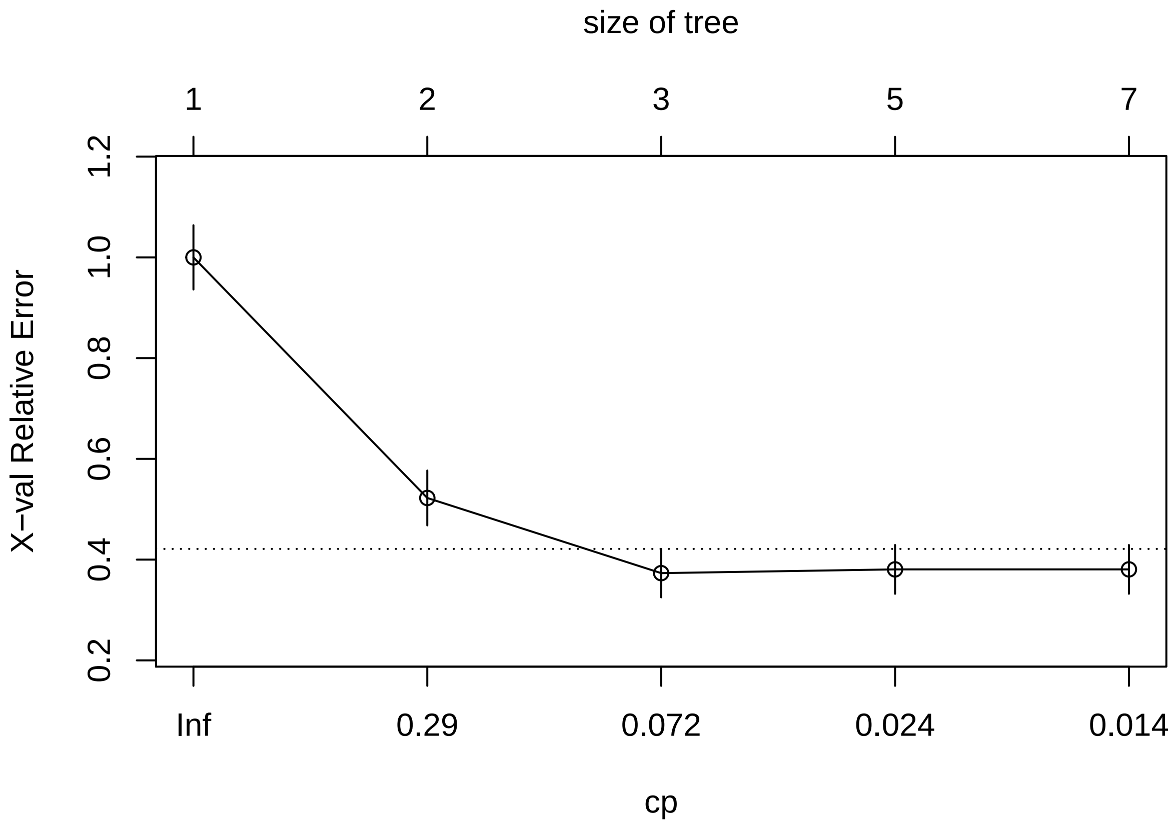
Evaluating the model

The performance of the model is tested with the generation of a confusion matrix and a visual analysis of the AUC ROC model. - The confusion matrix presents a 79.5% accuracy of the model and that paired with a 95% Sensitivity and 61% Specificity. Hence it can be seen as a very good model in predicting business success based on the four above mentioned features. - The AUC ROC model reveals the quality of the model and shows a big area under the curve, potentially a little over fitted, but still not closely reaching the top left corner of the plot area.

```r
# using a GINI decision tree to analyze the data
roi_prediction_tree <- rpart(formula =   goal_roi_binary ~ avg_CpC + impressions +
                                  key_af + camp_geo,
                             data = train_roi, method = "class", cp=0.01)

rpart.plot(roi_prediction_tree, type=1, extra = 1,
           box.palette = "RdBu",
           branch.lty = 3, shadow.col = "gray")
```



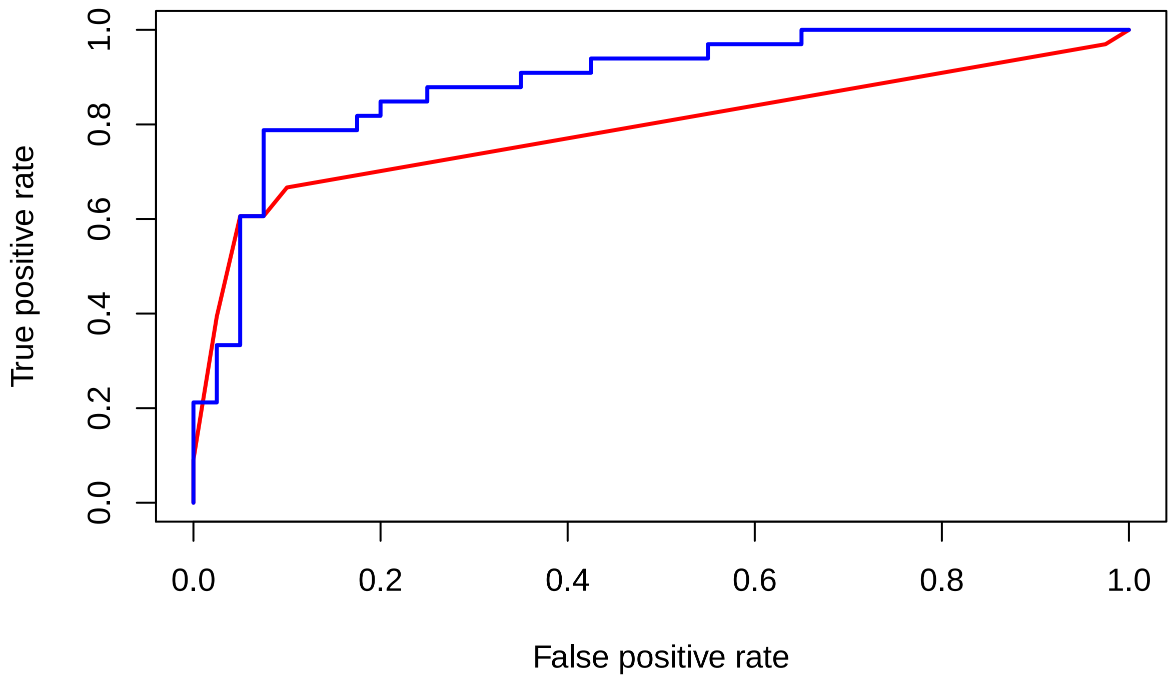
```r
plotcp(roi_prediction_tree)
```



Gini Decision Tree

The tree consists of 4 levels that lead to o7 terminal leafs. The gini tree model choses the average cost per click to be the most influencial determinant of business success followed by either the Impressions or the Air France keyword in the campaign. The feature, Geo targeted is not regarded in the tree.

```
#prediction
roi_prediction_tree_predict <- predict(roi_prediction_tree, test_roi, type = "prob")

#prepare for AUCROC
roi_prediction_tree_prediction <- prediction(roi_prediction_tree_predict[ , 2], test_roi$goal_roi_binar

#performance
roi_prediction_tree_performance <- performance(roi_prediction_tree_prediction, "tpr", "fpr")

plot(roi_prediction_tree_performance, col = "red", lwd = 2)
plot(perf_logit_roi, col = "blue", lwd = 2, add = TRUE)
```



Comparing models

Comparing the performance of the logistic regression model and the gini tree, it can be said that the area under the curve is bigger for the logistic regression model. This seems to more reliably predict the ROI goal. However, both models are far away from the “pig” line and therefore function rather well for the prediction of business success.

Recommendations on ROI goal

1. Analysis shows that the keyword Air France in the campaigns significantly impact the Return on Invest. In fact, this keyword has the biggest impact of all on the ROI. Therefore, we recommend to include this keyword in future campaigns with the focus on a high return on investment.
2. Whether the campaign was geo targeted or not is the second most influential factor in determining a successful campaign when success is defined by the return on invest. Hence, campaigns tend to perform better when geo targeted. Therefore, future campaigns of Air France shall be geo targeted as they appear to attract consumers more and thus generate higher returns.
3. The visibility of a campaign has very little impact on the return on invest. Therefore, it does not matter how long the campaign is active or how many people actually saw it. The common KPI to monitor campaign performance, impressions, can be misleading as the effect is very low, in fact negative.

Conclusion

Both of the approaches previously presented unveiled numerous insights that can be useful as main drivers for general success. At first, with the visibility model, the main point of focus should be the search engine bid media contacts set. Secondly, taking into account the ROI approach, we found that the main drivers for a predicted increase in the ROI are:

1. The use of the Air France key word
2. The use of geographically targeted campaigns

Considering that all of our models are statistically relevant and our insights are based on a dense sentiment analysis, these solutions can serve as a starting points towards the new growth strategy and market share development that the firm seeks.

Finally, solely the logistic regression model and the gini tree model have been facilitated in the search of the best machine learning model to predict both business goals. Further analysis could include other models such as Random Forest that could potentially outperform the two models of this analysis and deliver even more precise insights.

A video presentation of key insights of this report is avaible at: https://www.youtube.com/watch?v=bGiN74heH-A&feature=youtu.be