Team 4 – Tangina Parvez, Chaitanya Sai Battula, Roger Lopez Benet, Xiaoxuan Zhai, Deena Jiva

Visualization: Tools and Methods in R

Assignment: A1: Business Case Presentation (video submission assignment)

CODE – R

```
##Importing packages and data sets
install.packages("ggvis")
install.packages("caret")
install.packages("rlang")
install.packages("tidyverse")
install.packages("dplyr")
install.packages("tidyverse")

library(readxl)      # package for loading data
library(data.table)  # package for creating a table
library(dplyr)       # package for data manipulation in tables
library(plotly)      # package for plotting graphs
library(ggplot2)     # package for plotting graphs
library(caret)       # package conclusion tree
library(rpart)       # package for decision tree
library(rpart.plot)  # package to plot decision tree
library(readxl)      # package for loading data
library(tidyverse)

## loading the Air France data set
my_airfrance <- read_excel("C:/Users/deean/Desktop/HULT/Visualization_Methods and Tools R/Air France Case Spreadsheet Supplement.xls",sheet = "DoubleClick")
View(my_airfrance)
ncol(my_airfrance)
```

```
# View the structure of the dataset
str(my_airfrance)

table("Bid Strategy")

## viewing the numebr to rows(my_cust), columns(my_var)
my_cust <- nrow(my_airfrance)
my_var <- ncol(my_airfrance)

## indicating the 2 vectors are related will show the observations first then the measurements
my_dim <- c(my_cust, my_var)

## identify the number of NA in Bid Strategy
which(my_airfrance$`Bid Strategy` == "NA")
my_airfrance[my_airfrance==""] <- NA

# Check for missing values using the sapply() function
sapply(my_airfrance, function(x) sum(is.na(x)))

# print channels
publisher_name <- unique(my_airfrance$`Publisher Name`)

#sales by channel
sales <- c()
for (i in 1:length(publisher_name)) {
  sales <- c(sales,sum(my_airfrance$Amount[which(my_airfrance[,2]
                                    == publisher_name[i])]))
  i <- i + 1
}
```

```
print(sales)

cbind(publisher_name,as.numeric(sales))

# discovering the most profitable channel
max_sales<-max(sales)
publisher_name[which(sales == max_sales)]

#volumes of bookings by channel
clicks <- c()
for (i in 1:length(publisher_name)) {
  clicks <- c(clicks,sum(my_airfrance$Clicks[which(my_airfrance[,2]
                                    == publisher_name[i])]))
 i <- i + 1
}
print(clicks)

# discovering the most profitable channel
max_clicks<-max(clicks)
publisher_name[which(clicks == max_clicks)]

cbind(publisher_name,as.numeric(clicks),as.numeric(sales))

# viewing datasets
sum(my_airfrance$Clicks)
sum(my_airfrance$`Total Volume of Bookings`)

AF <- my_airfrance
```

```
Statistics <- c("Mean", "Median", "SD", "Min", "Max")

Amount <- round(c(mean(AF$Amount),median(AF$Amount), sd(AF$Amount),min(AF$Amount), max(AF$Amount)), 2)

Total_Cost <- round(c(mean(AF$`Total Cost`),median(AF$`Total Cost`),sd(AF$`Total Cost`),min(AF$`Total Cost`),max(AF$`Total Cost`)), 2)

Impressions <- round(c(mean(AF$Impressions),median(AF$Impressions),sd(AF$Impressions),min(AF$Impressions),max(AF$Impressions)), 2)

Clicks <- round(c(mean(AF$Clicks),median(AF$Clicks),sd(AF$Clicks),min(AF$Clicks),max(AF$Clicks)), 2)

CTR <- round(c(mean(AF$`Engine Click Thru %`),median(AF$`Engine Click Thru %`),sd(AF$`Engine Click Thru %`),min(AF$`Engine Click Thru %`),max(AF$`Engine Click Thru %`)), 2)


Summary <- as.data.frame(cbind(Statistics, Amount, Total_Cost, Impressions, Clicks,CTR))

View(AF)

View(Summary)


plot(Total_Cost)

plot(CTR)

plot(Clicks)


############################

## Finding of Live Status ##

############################

# Filter the data to only include rows with a "live" status

active_campaign <- subset(my_airfrance, Status == "Live")


# Calculate the click-through rate (CTR) for each campaign

active_campaign_summary <- aggregate(Impressions ~ Campaign, data = active_campaign, FUN = sum)

active_campaign_summary$Clicks <- aggregate(Clicks ~ Campaign, data = active_campaign, FUN = sum)$Clicks
```

```
active_campaign_summary$CTR <- active_campaign_summary$Clicks /
active_campaign_summary$Impressions


################################

## Finding of non Live Status ##

################################

# Filter the data to only include rows with a "live" status

inactive_campaign <- subset(my_airfrance, Status != "Live")


# Calculate the click-through rate (CTR) for each campaign

inactive_campaign_summary <- aggregate(Impressions ~ Campaign, data = inactive_campaign, FUN =
sum)

inactive_campaign_summary$Clicks <- aggregate(Clicks ~ Campaign, data = inactive_campaign, FUN =
sum)$Clicks

inactive_campaign_summary$CTR <- active_campaign_summary$Clicks /
inactive_campaign_summary$Impressions


# Substituting missing values with NA

my_airfrance[my_airfrance == ""] <- NA


# Check for missing values using the sapply() function

sapply(my_airfrance, function(x) sum(is.na(x)))


#typeof(my_airfrance$Amount)

#typeof(my_airfrance$`Total Cost`)


Total_Cost <- unique(my_airfrance$`Total Cost`)


# Printing all unique values for the Publisher Name variable:

publisher_names <- unique(my_airfrance$`Publisher Name`)
```

```
# Sales by channel
sales <- c()
for (i in 1:length(publisher_name)) {
  sales <- c(sales,sum(my_airfrance$Amount[which(my_airfrance[,2]
                                    == publisher_name[i])]))
 i <- i + 1
} # Closing for loop

print(sales)

# Using cbind to combine the columns:
cbind(publisher_name,as.numeric(sales))

### The most profitable channel
max_sales<-max(sales)
publisher_name[which(sales == max_sales)] # The most profitable channel is: "Google - US"

### Getting value_counts of each value in the keyword variable:
unique(my_airfrance$Keyword)
table(my_airfrance$Keyword) # Most keywords are only used once:

### Getting Stats:

AF <- my_airfrance
TVB <- AF$`Total Volume of Bookings`

Statistics <- c("Mean", "Median", "SD", "Min", "Max")
```

```
Amount <- round(c(mean(AF$Amount),median(AF$Amount), sd(AF$Amount),min(AF$Amount), max(AF$Amount)), 2)

Total_Cost <- round(c(mean(AF$`Total Cost`),median(AF$`Total Cost`),sd(AF$`Total Cost`),min(AF$`Total Cost`),max(AF$`Total Cost`)), 2)

Impressions <- round(c(mean(AF$Impressions),median(AF$Impressions),sd(AF$Impressions),min(AF$Impressions),max(AF$Impressions)), 2)

Clicks <- round(c(mean(AF$Clicks),median(AF$Clicks),sd(AF$Clicks),min(AF$Clicks),max(AF$Clicks)), 2)

CTR <- round(c(mean(AF$`Engine Click Thru %`),median(AF$`Engine Click Thru %`),sd(AF$`Engine Click Thru %`),min(AF$`Engine Click Thru %`),max(AF$`Engine Click Thru %`)), 2)


Summary <- as.data.frame(cbind(Statistics, Amount, Total_Cost, Impressions, Clicks,CTR))

View(AF)

View(Summary)


#########################

## Split The Publisher ##

#########################

# Create a new column to store the publisher type

my_airfrance$Publisher_Type <- ""


# Use grepl to check if the Publisher Name column contains "US"

my_airfrance$Publisher_Type[grepl("US", my_airfrance$`Publisher Name`)] <- "US"

my_airfrance$Publisher_Type[!grepl("US", my_airfrance$`Publisher Name`)] <- "Global"


# Check the results

head(my_airfrance)

View(my_airfrance)


#dataframe with just the US-publisher
```

```
US_publishers <- c("Google - US", "MSN - US", "Overture - US", "Yahoo - US")
my_airfrance_US <- my_airfrance[my_airfrance$`Publisher Name`%in% US_publishers,]
my_airfrance_US$count <- 1

# Define the values for the new row
search_engine <- "Kayak"
clicks <- 2839
media_cost <- 3567.13
total_bookings <- 208
avg_ticket <- 1123.53
total_revenue <- 233694.00
net_revenue <- 230126.87

# Create a new row with NA values for the remaining variables
kayak_row <-  data.frame(
  `Publisher ID` = NA,
  `Publisher Name` = search_engine,
  `Keyword ID` = NA,
  `Keyword` = NA,
  `Match Type` = NA,
  `Campaign` = NA,
  `Keyword Group` = NA,
  `Category` = NA,
  `Bid Strategy` = NA,
  `Keyword Type` = NA,
  `Status` = NA,
  `Search Engine Bid` = NA,
  `Clicks` = clicks,
  `Click Charges` = NA,
```

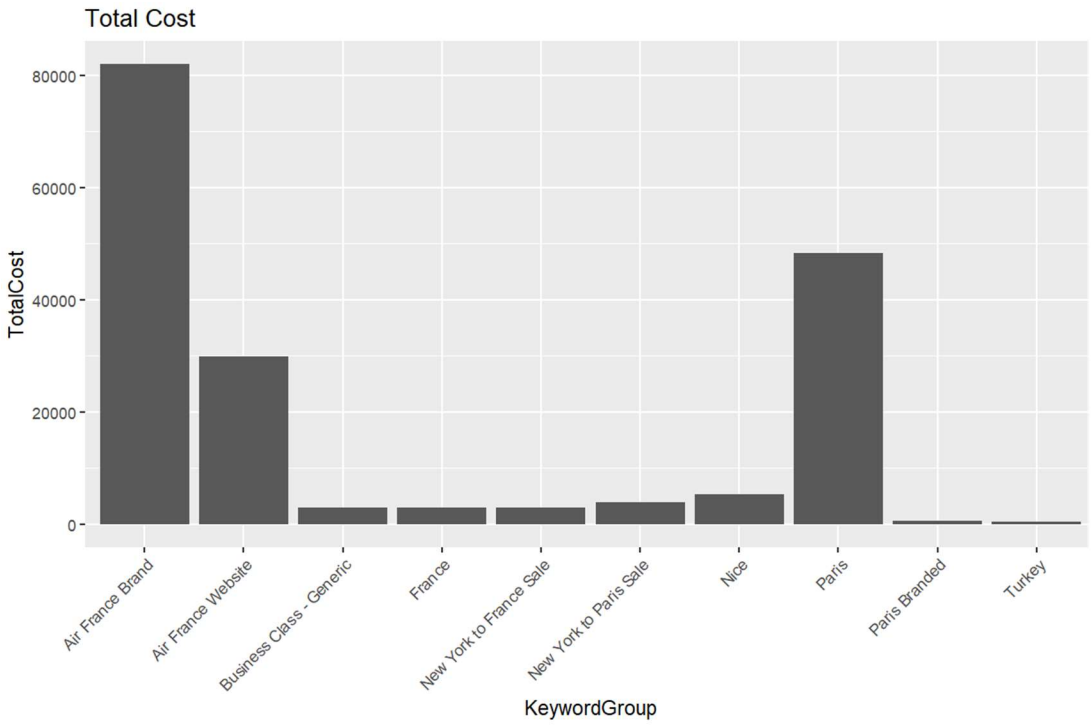
```
  `Avg. Cost per Click` = NA,
  `Impressions` = NA,
  `Engine Click Thru %` = NA,
  `Avg. Pos.` = NA,
  `Trans. Conv. %` = NA,
  `Total Cost/ Trans.` = NA,
  Amount = total_revenue,
  `Total Cost` = media_cost,
  `Total Volume of Bookings` = `total_bookings`,
  `Publisher_Type` = 'US'
)
colnames(kayak_row) <- gsub("[\\. \\/]", "", names(kayak_row))
my_airfrance2 <- my_airfrance
colnames(my_airfrance2) <- gsub("[\\. \\/%]", "", names(my_airfrance2))

# Append the new row to the existing data frame
my_airfrance2 <- rbind(my_airfrance2, kayak_row)

chk <- c()
for (org in names(my_airfrance2)){
  chk2 <- TRUE
  for (col in names(kayak_row)){
    if (org == col){
      chk2 <- FALSE
    }
  }
  if (chk2){
    chk <- c(chk, org)
  }
```

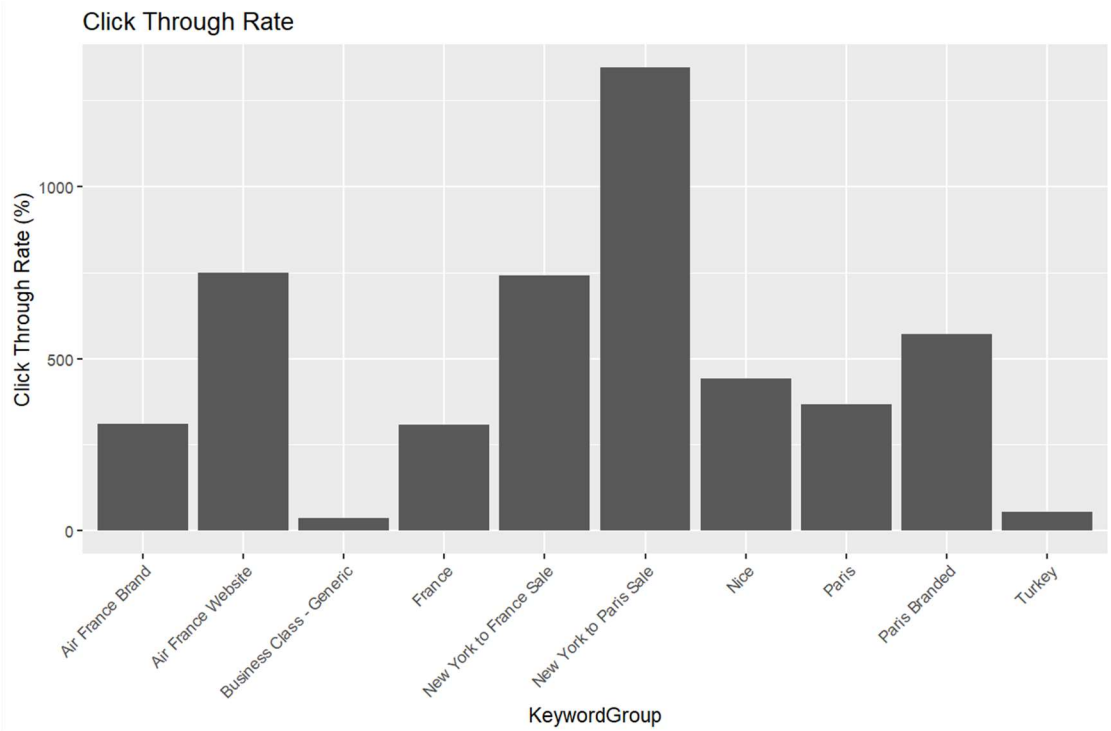
```
}

chk

my_airfrance2[4500:4511, ]
my_airfrance2
# Append the new row to the existing data frame
my_airfrance2 <- rbind(my_airfrance2, kayak_row)

# Rename the variables in the new row to match those in the existing data frame
names(my_airfrance2)[names(my_airfrance2) == "Search_Engine"] <- "Publisher Name"
names(my_airfrance2)[names(my_airfrance2) == "Media_Cost"] <- "Total Cost"
names(my_airfrance2)[names(my_airfrance2) == "Total_Revenue"] <- "Amount"
names(my_airfrance2)[names(my_airfrance2) == "Total_Bookings"] <- "Total Volume of Bookings"

# Add the values to the new row
my_airfrance2[nrow(my_airfrance2), "Publisher.Name"] <- search_engine
my_airfrance2[nrow(my_airfrance2), "Clicks"] <- clicks
my_airfrance2[nrow(my_airfrance2), "Total.Cost"] <- media_cost
my_airfrance2[nrow(my_airfrance2), "Total.Volume.of.Bookings"] <- total_bookings
my_airfrance2[nrow(my_airfrance2), "Amount"] <- total_revenue
my_airfrance2[nrow(my_airfrance2), "Net_Revenue"] <- net_revenue

# View the updated data frame
my_airfrance2

#Delete row 4512
my_airfrance2 <- my_airfrance2[-4512, ]
```

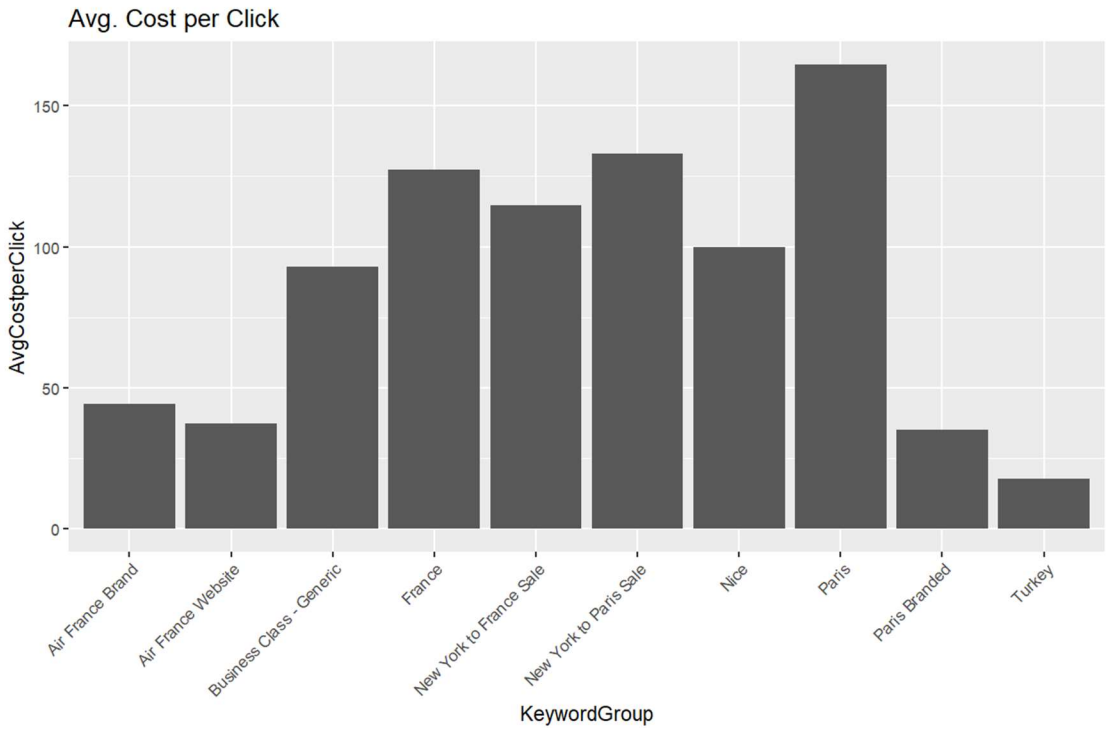
```
###################################
## Delete the extra column      ##
###################################

# Remove columns by name
my_airfrance2 <- my_airfrance2[, !(names(my_airfrance2) %in% c("Publisher.Name", "Total.Cost",
"Total.Volume.of.Bookings", "Net_Revenue"))]

# View the updated data frame
View(my_airfrance2)

# Create a new column to store the Campaign Status
my_airfrance2$Campaign_Status <- ""

# Use grepl to check if the Publisher Name column contains "US"
my_airfrance2$Campaign_Status[grepl("Live", my_airfrance2$`Status`)] <- "1"
my_airfrance2$Campaign_Status[!grepl("Live", my_airfrance2$`Status`)] <- "0"
my_airfrance2$Campaign_Status <- as.numeric(my_airfrance2$Campaign_Status)

View(my_airfrance2)
###################################
##          Graphs           ##
###################################
library(dplyr)
library(ggplot2)

Inactive_campaign <- my_airfrance2 %>% filter(Campaign_Status == 0)
Active_campaign <- my_airfrance2 %>% filter(Campaign_Status == 1)
```

```
Inactive_campaign
Active_campaign

# Determine the top N KeywordGroup values by TotalCost
N <- 10
top_keyword_groups <- Active_campaign %>%
  group_by(KeywordGroup) %>%
  summarise(TotalCost = sum(TotalCost, na.rm = TRUE)) %>%
  top_n(n = N, wt = TotalCost) %>%
  pull(KeywordGroup)

# Filter the dataset to only include the top N KeywordGroup values
filtered_data <- Active_campaign %>%
  filter(KeywordGroup %in% top_keyword_groups)

# Total Cost
ggplot(filtered_data, aes(x = KeywordGroup, y = TotalCost)) +
  geom_bar(stat = "identity") +
  theme(axis.text.x = element_text(angle = 45, hjust = 1)) +
  labs(title = "Total Cost", x = "KeywordGroup", y = "TotalCost")

# Click Through Rate
ggplot(filtered_data, aes(x = KeywordGroup, y = EngineClickThru)) +
  geom_bar(stat = "identity") +
  theme(axis.text.x = element_text(angle = 45, hjust = 1)) +
  labs(title = "Click Through Rate", x = "KeywordGroup", y = "Click Through Rate (%)")

# Avg. Cost per Click
ggplot(filtered_data, aes(x = KeywordGroup, y = AvgCostperClick)) +
```

```
geom_bar(stat = "identity") +
theme(axis.text.x = element_text(angle = 45, hjust = 1)) +
labs(title = "Avg. Cost per Click", x = "KeywordGroup", y = "AvgCostperClick")
```








```
###################################
##    Prepare for the model     ##
###################################

# Create a new column to store the binary variable
my_airfrance2$Bookings_Status <- 0

# Use ifelse to set Bookings_Status to 1 for rows where TotalVolumeofBookings > 0
my_airfrance2$Bookings_Status <- ifelse(my_airfrance2$TotalVolumeofBookings > 0, 1, 0)

# Check the distribution of Bookings_Status
table(my_airfrance2$Bookings_Status)

###################################
## Building the logistic model  ##
###################################
```

```
# Setting the train index
train_index <- sample(1:nrow(my_airfrance2),
                      size=0.8*nrow(my_airfrance2))

# Making train and test data
train_data <- my_airfrance2[train_index, ]
test_data  <- my_airfrance2[-train_index,]

# Fit a logistic regression model
my_logit <- glm(Bookings_Status ~ AvgCostperClick + TotalCost + EngineClickThru+ Campaign_Status,
                data = train_data, family = binomial)

# Print model summary
summary(my_logit)

#predict in the test data
my_prediction <- predict(my_logit, test_data, type="response")

###################################
##  Testing the performance     ##
###################################

#testing performance of your model
library(caret)
#doing the Confusion Matrix
confusionMatrix(data=as.factor(as.numeric(my_prediction>0.5)),
                reference=as.factor(as.numeric(test_data$Bookings_Status)))
library(tidyverse)
library(dplyr)
```

```
# Create the linear regression model
linear_model <- lm(TotalCost ~ EngineClickThru +
                   AvgCostperClick +
                   TotalVolumeofBookings +
                   Clicks +
                   Amount, data = train_data)

# Print the model summary
summary(linear_model)

# Make predictions using the test data
predictions <- predict(linear_model, newdata = test_data)

# Calculate residuals
residuals <- test_data$TotalCost - predictions

# Compare the predicted values to the actual values
comparison <- data.frame(
  Actual = test_data$TotalCost,
  Predicted = predictions,
  Residual = residuals, # use the calculated residuals here
  Difference = abs(test_data$TotalCost - predictions)
)

comparison

# Calculate the performance metrics
MAE <- mean(comparison$Difference)
MSE <- mean(comparison$Difference^2)
```

```
R2 <- 1 - (sum(comparison$Difference^2) / sum((test_data$TotalCost - mean(test_data$TotalCost))^2))

list(MAE = MAE, MSE = MSE, R2 = R2)

# Create a new data frame with the actual, predicted values, residuals, and Keyword variable from test_data
comparison_with_keyword <- data.frame(Actual = test_data$TotalCost,
                                      Predicted = predictions,
                                      Residual = residuals, # use the calculated residuals here
                                      Keyword = test_data$Keyword)

# ... (rest of the code, excluding the repeated blocks)

# ... (rest of the code, excluding the repeated blocks)

# Create a new data frame with the KeywordGroup variable from test_data
comparison_with_keyword_and_group <- cbind(comparison_with_keyword, KeywordGroup = test_data$KeywordGroup)

# Get the top 5 keyword groups by total cost
top_5_keyword_groups <- test_data %>%
  group_by(KeywordGroup) %>%
  summarize(TotalCost = sum(TotalCost)) %>%
  top_n(5, TotalCost) %>%
  pull(KeywordGroup)

# Filter the comparison_with_keyword_and_group data frame to include only the top 5 keyword groups
comparison_with_keyword_top_5_groups <- comparison_with_keyword_and_group %>%
  filter(KeywordGroup %in% top_5_keyword_groups)
```

```
# Scatterplot of Actual vs. Predicted TotalCost with top 5 Keyword Groups:
ggplot(comparison_with_keyword_top_5_groups, aes(x = Actual, y = Predicted, color = KeywordGroup))
+
  geom_point() +
  geom_abline(intercept = 0, slope = 1, color = "red", linetype = "dashed") +
  labs(title = "Scatterplot of Actual vs. Predicted TotalCost",
       x = "Actual TotalCost",
       y = "Predicted TotalCost") +
  scale_color_discrete(name = "KeywordGroup")

#testing performance of your model
library(caret)

#doing the Confusion Matrix
confusionMatrix(data=as.factor(as.numeric(my_prediction>0.5)),
                reference=as.factor(as.numeric(test_data$Bookings_Status)))

comparison_with_keyword <- data.frame(
  Actual = test_data$TotalCost,
  Predicted = my_prediction,
  residuals = test_data$TotalCost - my_prediction, # calculate residuals
  Keyword = test_data$Keyword
)
# Create a new data frame with the KeywordGroup variable from test_data
comparison_with_keyword_and_group <- cbind(comparison_with_keyword, KeywordGroup =
test_data$KeywordGroup)

# Get the top 5 keyword groups by total cost
```

```
top_5_keyword_groups <- test_data %>%
  group_by(KeywordGroup) %>%
  summarize(TotalCost = sum(TotalCost)) %>%
  top_n(5, TotalCost) %>%
  pull(KeywordGroup)

# Filter the comparison_with_keyword_and_group data frame to include only the top 5 keyword groups
comparison_with_keyword_top_5_groups <- comparison_with_keyword_and_group %>%
  filter(KeywordGroup %in% top_5_keyword_groups)

# Scatterplot of Actual vs. Predicted TotalCost with top 5 Keyword Groups:
ggplot(comparison_with_keyword_top_5_groups, aes(x = Actual, y = Predicted, color = KeywordGroup))
+
  geom_point() +
  geom_abline(intercept = 0, slope = 1, color = "red", linetype = "dashed") +
  labs(title = "Scatterplot of Actual vs. Predicted TotalCost",
       x = "Actual TotalCost",
       y = "Predicted TotalCost") +
  scale_color_discrete(name = "KeywordGroup")
[21:42, 4/7/2023] +1 (415) 404-2825: library(readxl)       # package for loading data
library(data.table)   # package for creating a table
library(dplyr)        # package for data manipulation in tables
library(plotly)       # package for plotting graphs
library(ggplot2)      # package for plotting graphs
library(caret)        # package conclusion tree
library(rpart)        # package for decision tree
library(rpart.plot)   # package to plot decision tree
library(readxl)       # package for loading data
```

```
## loading the Air France data set

my_airfrance <- read_excel("C:/Users/14154/Downloads/Air France Case Spreadsheet Supplement (1).xls", sheet = 1)

View(my_airfrance)

ncol(my_airfrance)


# View the structure of the dataset

str(my_airfrance)


table("Bid Strategy")


## viewing the numebr to rows(my_cust), columns(my_var)

my_cust <- nrow(my_airfrance)

my_var <- ncol(my_airfrance)


## indicating the 2 vectors are related will show the observations first then the measurements

my_dim <- c(my_cust, my_var)


## identify the number of NA in Bid Strategy

which(my_airfrance$`Bid Strategy` == "NA")

my_airfrance[my_airfrance==""] <- NA


# Check for missing values using the sapply() function

sapply(my_airfrance, function(x) sum(is.na(x)))


# print channels

publisher_name <- unique(my_airfrance$`Publisher Name`)


#sales by channel
```

```
sales <- c()
for (i in 1:length(publisher_name)) {
  sales <- c(sales,sum(my_airfrance$Amount[which(my_airfrance[,2]
                                    == publisher_name[i])]))
  i <- i + 1
}
print(sales)

cbind(publisher_name,as.numeric(sales))

# discovering the most profitable channel
max_sales<-max(sales)
publisher_name[which(sales == max_sales)]

#volumes of bookings by channel
clicks <- c()
for (i in 1:length(publisher_name)) {
  clicks <- c(clicks,sum(my_airfrance$Clicks[which(my_airfrance[,2]
                                     == publisher_name[i])]))
  i <- i + 1
}
print(clicks)

# discovering the most profitable channel
max_clicks<-max(clicks)
publisher_name[which(clicks == max_clicks)]

cbind(publisher_name,as.numeric(clicks),as.numeric(sales))
```

```
# viewing datasets

sum(my_airfrance$Clicks)

sum(my_airfrance$`Total Volume of Bookings`)


AF <- my_airfrance


Statistics <- c("Mean", "Median", "SD", "Min", "Max")

Amount <- round(c(mean(AF$Amount),median(AF$Amount), sd(AF$Amount),min(AF$Amount),
max(AF$Amount)), 2)

Total_Cost <- round(c(mean(AF$`Total Cost`),median(AF$`Total Cost`),sd(AF$`Total Cost`),min(AF$`Total
Cost`),max(AF$`Total Cost`)), 2)

Impressions <-
round(c(mean(AF$Impressions),median(AF$Impressions),sd(AF$Impressions),min(AF$Impressions),max(
AF$Impressions)), 2)

Clicks <- round(c(mean(AF$Clicks),median(AF$Clicks),sd(AF$Clicks),min(AF$Clicks),max(AF$Clicks)), 2)

CTR <- round(c(mean(AF$`Engine Click Thru %`),median(AF$`Engine Click Thru %`),sd(AF$`Engine Click
Thru %`),min(AF$`Engine Click Thru %`),max(AF$`Engine Click Thru %`)), 2)


Summary <- as.data.frame(cbind(Statistics, Amount, Total_Cost, Impressions, Clicks,CTR))

View(AF)

View(Summary)


plot(Total_Cost)

plot(CTR)

plot(Clicks)


############################

## Finding of Live Status ##

############################

# Filter the data to only include rows with a "live" status
```

```
active_campaign <- subset(my_airfrance, Status == "Live")


# Calculate the click-through rate (CTR) for each campaign

active_campaign_summary <- aggregate(Impressions ~ Campaign, data = active_campaign, FUN = sum)

active_campaign_summary$Clicks <- aggregate(Clicks ~ Campaign, data = active_campaign, FUN =
sum)$Clicks

active_campaign_summary$CTR <- active_campaign_summary$Clicks /
active_campaign_summary$Impressions


################################

## Finding of non Live Status ##

################################

# Filter the data to only include rows with a "live" status

inactive_campaign <- subset(my_airfrance, Status != "Live")


# Calculate the click-through rate (CTR) for each campaign

inactive_campaign_summary <- aggregate(Impressions ~ Campaign, data = inactive_campaign, FUN =
sum)

inactive_campaign_summary$Clicks <- aggregate(Clicks ~ Campaign, data = inactive_campaign, FUN =
sum)$Clicks

inactive_campaign_summary$CTR <- active_campaign_summary$Clicks /
inactive_campaign_summary$Impressions


# Substituting missing values with NA

my_airfrance[my_airfrance == ""] <- NA


# Check for missing values using the sapply() function

sapply(my_airfrance, function(x) sum(is.na(x)))


#typeof(my_airfrance$Amount)
```

```
#typeof(my_airfrance$`Total Cost`)

Total_Cost <- unique(my_airfrance$`Total Cost`)

# Printing all unique values for the Publisher Name variable:
publisher_names <- unique(my_airfrance$`Publisher Name`)

# Sales by channel
sales <- c()
for (i in 1:length(publisher_name)) {
  sales <- c(sales,sum(my_airfrance$Amount[which(my_airfrance[,2]
                                    == publisher_name[i])]))
  i <- i + 1
} # Closing for loop

print(sales)

# Using cbind to combine the columns:
cbind(publisher_name,as.numeric(sales))

### The most profitable channel
max_sales<-max(sales)
publisher_name[which(sales == max_sales)] # The most profitable channel is: "Google - US"

### Getting value_counts of each value in the keyword variable:
unique(my_airfrance$Keyword)
table(my_airfrance$Keyword) # Most keywords are only used once:

### Getting Stats:
```

```
AF <- my_airfrance

TVB <- AF$`Total Volume of Bookings`


Statistics <- c("Mean", "Median", "SD", "Min", "Max")

Amount <- round(c(mean(AF$Amount),median(AF$Amount), sd(AF$Amount),min(AF$Amount), max(AF$Amount)), 2)

Total_Cost <- round(c(mean(AF$`Total Cost`),median(AF$`Total Cost`),sd(AF$`Total Cost`),min(AF$`Total Cost`),max(AF$`Total Cost`)), 2)

Impressions <- round(c(mean(AF$Impressions),median(AF$Impressions),sd(AF$Impressions),min(AF$Impressions),max(AF$Impressions)), 2)

Clicks <- round(c(mean(AF$Clicks),median(AF$Clicks),sd(AF$Clicks),min(AF$Clicks),max(AF$Clicks)), 2)

CTR <- round(c(mean(AF$`Engine Click Thru %`),median(AF$`Engine Click Thru %`),sd(AF$`Engine Click Thru %`),min(AF$`Engine Click Thru %`),max(AF$`Engine Click Thru %`)), 2)


Summary <- as.data.frame(cbind(Statistics, Amount, Total_Cost, Impressions, Clicks,CTR))

View(AF)

View(Summary)


#########################

## Split The Publisher ##

#########################

# Create a new column to store the publisher type

my_airfrance$Publisher_Type <- ""


# Use grepl to check if the Publisher Name column contains "US"

my_airfrance$Publisher_Type[grepl("US", my_airfrance$`Publisher Name`)] <- "US"

my_airfrance$Publisher_Type[!grepl("US", my_airfrance$`Publisher Name`)] <- "Global"


# Check the results
```

```
head(my_airfrance)
View(my_airfrance)

#dataframe with just the US-publisher
US_publishers <- c("Google - US", "MSN - US", "Overture - US", "Yahoo - US")
my_airfrance_US <- my_airfrance[my_airfrance$`Publisher Name`%in% US_publishers,]
my_airfrance_US$count <- 1

# Define the values for the new row
search_engine <- "Kayak"
clicks <- 2839
media_cost <- 3567.13
total_bookings <- 208
avg_ticket <- 1123.53
total_revenue <- 233694.00
net_revenue <- 230126.87

# Create a new row with NA values for the remaining variables
kayak_row <-  data.frame(
  `Publisher ID` = NA,
  `Publisher Name` = search_engine,
  `Keyword ID` = NA,
  `Keyword` = NA,
  `Match Type` = NA,
  `Campaign` = NA,
  `Keyword Group` = NA,
  `Category` = NA,
  `Bid Strategy` = NA,
  `Keyword Type` = NA,
```

```
  `Status` = NA,
  `Search Engine Bid` = NA,
  `Clicks` = clicks,
  `Click Charges` = NA,
  `Avg. Cost per Click` = NA,
  `Impressions` = NA,
  `Engine Click Thru %` = NA,
  `Avg. Pos.` = NA,
  `Trans. Conv. %` = NA,
  `Total Cost/ Trans.` = NA,
  Amount = total_revenue,
  `Total Cost` = media_cost,
  `Total Volume of Bookings` = `total_bookings`,
  `Publisher_Type` = 'US'
)
colnames(kayak_row) <- gsub("[\\. \\/]", "", names(kayak_row))
my_airfrance2 <- my_airfrance
colnames(my_airfrance2) <- gsub("[\\. \\/%]", "", names(my_airfrance2))

# Append the new row to the existing data frame
my_airfrance2 <- rbind(my_airfrance2, kayak_row)

chk <- c()
for (org in names(my_airfrance2)){
  chk2 <- TRUE
  for (col in names(kayak_row)){
    if (org == col){
      chk2 <- FALSE
    }
```

```
 }
 if (chk2){
   chk <- c(chk, org)
 }
}

chk

my_airfrance2[4500:4511, ]
my_airfrance2
# Append the new row to the existing data frame
my_airfrance2 <- rbind(my_airfrance2, kayak_row)

# Rename the variables in the new row to match those in the existing data frame
names(my_airfrance2)[names(my_airfrance2) == "Search_Engine"] <- "Publisher Name"
names(my_airfrance2)[names(my_airfrance2) == "Media_Cost"] <- "Total Cost"
names(my_airfrance2)[names(my_airfrance2) == "Total_Revenue"] <- "Amount"
names(my_airfrance2)[names(my_airfrance2) == "Total_Bookings"] <- "Total Volume of Bookings"

# Add the values to the new row
my_airfrance2[nrow(my_airfrance2), "Publisher.Name"] <- search_engine
my_airfrance2[nrow(my_airfrance2), "Clicks"] <- clicks
my_airfrance2[nrow(my_airfrance2), "Total.Cost"] <- media_cost
my_airfrance2[nrow(my_airfrance2), "Total.Volume.of.Bookings"] <- total_bookings
my_airfrance2[nrow(my_airfrance2), "Amount"] <- total_revenue
my_airfrance2[nrow(my_airfrance2), "Net_Revenue"] <- net_revenue

# View the updated data frame
my_airfrance2
```

```
#Delete row 4512

my_airfrance2 <- my_airfrance2[-4512, ]


###################################

## Delete the extra column      ##

###################################

install.packages("tidyverse")

install.packages("dplyr")

library(tidyverse)


library(dplyr)


# Remove columns by name

my_airfrance2 <- my_airfrance2[, !(names(my_airfrance2) %in% c("Publisher.Name", "Total.Cost",
"Total.Volume.of.Bookings", "Net_Revenue"))]


# View the updated data frame

View(my_airfrance2)


# Create a new column to store the Campaign Status

my_airfrance2$Campaign_Status <- ""


# Use grepl to check if the Publisher Name column contains "US"

my_airfrance2$Campaign_Status[grepl("Live", my_airfrance2$`Status`)] <- "1"

my_airfrance2$Campaign_Status[!grepl("Live", my_airfrance2$`Status`)] <- "0"

my_airfrance2$Campaign_Status <- as.numeric(my_airfrance2$Campaign_Status)


View(my_airfrance2)
```

```
###################################
##         Graphs          ##
###################################

library(dplyr)
library(ggplot2)
Inactive_campaign <- my_airfrance2 %>% filter(Campaign_Status == 0)
Active_campaign <- my_airfrance2 %>% filter(Campaign_Status == 1)

Inactive_campaign
Active_campaign



# Determine the top N KeywordGroup values by TotalCost
N <- 10
top_keyword_groups <- Active_campaign %>%
  group_by(KeywordGroup) %>%
  summarise(TotalCost = sum(TotalCost, na.rm = TRUE)) %>%
  top_n(n = N, wt = TotalCost) %>%
  pull(KeywordGroup)

# Filter the dataset to only include the top N KeywordGroup values
filtered_data <- Active_campaign %>%
  filter(KeywordGroup %in% top_keyword_groups)

# Total Cost
ggplot(filtered_data, aes(x = KeywordGroup, y = TotalCost)) +
  geom_bar(stat = "identity") +
  theme(axis.text.x = element_text(angle = 45, hjust = 1)) +
```

```
  labs(title = "Total Cost", x = "KeywordGroup", y = "TotalCost")

# Click Through Rate
ggplot(filtered_data, aes(x = KeywordGroup, y = EngineClickThru)) +
  geom_bar(stat = "identity") +
  theme(axis.text.x = element_text(angle = 45, hjust = 1)) +
  labs(title = "Click Through Rate", x = "KeywordGroup", y = "Click Through Rate (%)")

# Avg. Cost per Click
ggplot(filtered_data, aes(x = KeywordGroup, y = AvgCostperClick)) +
  geom_bar(stat = "identity") +
  theme(axis.text.x = element_text(angle = 45, hjust = 1)) +
  labs(title = "Avg. Cost per Click", x = "KeywordGroup", y = "AvgCostperClick")

###################################
##    Prepare for the model     ##
###################################

# Create a new column to store the binary variable
my_airfrance2$Bookings_Status <- 0

# Use ifelse to set Bookings_Status to 1 for rows where TotalVolumeofBookings > 0
my_airfrance2$Bookings_Status <- ifelse(my_airfrance2$TotalVolumeofBookings > 0, 1, 0)

# Check the distribution of Bookings_Status
table(my_airfrance2$Bookings_Status)

###################################
## Building the logistic model  ##
```

```
###################################
# Split the data into training (80%) and testing (20%) sets

install.packages("caTools")
library(caTools)

set.seed(123)
split <- sample.split(filtered_data$TotalCost, SplitRatio = 0.8)
train_data <- filtered_data[split, ]
test_data <- filtered_data[!split, ]

###################################
##  Testing the performance     ##
###################################

#testing performance of your model
library(caret)
###################################
##  Linear Regression Model     ##
###################################
# Create the linear regression model
linear_model <- lm(TotalCost ~ EngineClickThru +
                   AvgCostperClick +
                   TotalVolumeofBookings +
                   Clicks +
                   Amount, data = train_data)

# Print the model summary
summary(linear_model)
```

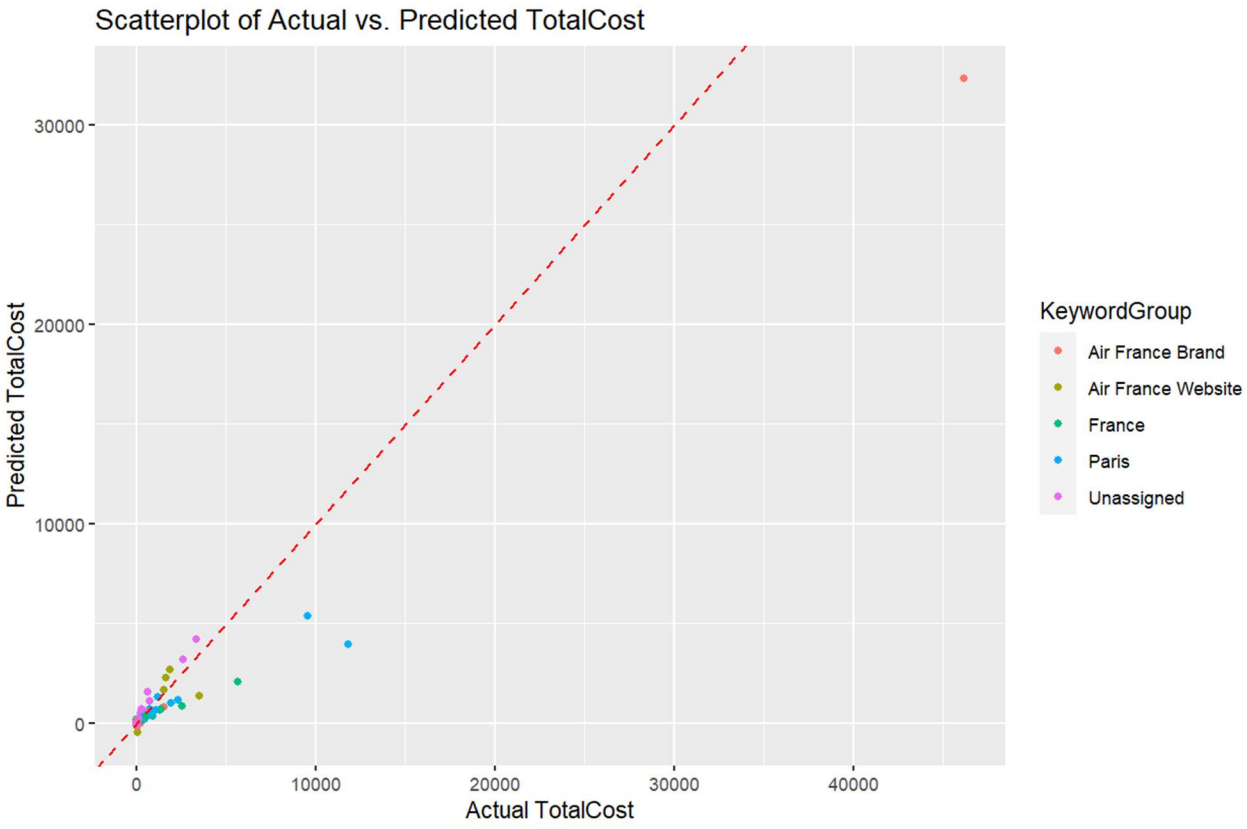
```
# Make predictions using the test data
predictions <- predict(linear_model, newdata = test_data)

# Calculate residuals
residuals <- test_data$TotalCost - predictions

# Compare the predicted values to the actual values
comparison <- data.frame(
  Actual = test_data$TotalCost,
  Predicted = predictions,
  Residual = residuals, # use the calculated residuals here
  Difference = abs(test_data$TotalCost - predictions)
)

comparison

# Calculate the performance metrics
MAE <- mean(comparison$Difference)
MSE <- mean(comparison$Difference^2)
R2 <- 1 - (sum(comparison$Difference^2) / sum((test_data$TotalCost - mean(test_data$TotalCost))^2))

list(MAE = MAE, MSE = MSE, R2 = R2)

# Create a new data frame with the actual, predicted values, residuals, and Keyword variable from test_data
comparison_with_keyword <- data.frame(Actual = test_data$TotalCost,
                                      Predicted = predictions,
                                      Residual = residuals, # use the calculated residuals here
```

```
                          Keyword = test_data$Keyword)

# ... (rest of the code, excluding the repeated blocks)

# ... (rest of the code, excluding the repeated blocks)

# Create a new data frame with the KeywordGroup variable from test_data
comparison_with_keyword_and_group <- cbind(comparison_with_keyword, KeywordGroup =
test_data$KeywordGroup)

# Get the top 5 keyword groups by total cost
top_5_keyword_groups <- test_data %>%
  group_by(KeywordGroup) %>%
  summarize(TotalCost = sum(TotalCost)) %>%
  top_n(5, TotalCost) %>%
  pull(KeywordGroup)

# Filter the comparison_with_keyword_and_group data frame to include only the top 5 keyword groups
comparison_with_keyword_top_5_groups <- comparison_with_keyword_and_group %>%
  filter(KeywordGroup %in% top_5_keyword_groups)

# Scatterplot of Actual vs. Predicted TotalCost with top 5 Keyword Groups:
ggplot(comparison_with_keyword_top_5_groups, aes(x = Actual, y = Predicted, color = KeywordGroup))
+
  geom_point() +
  geom_abline(intercept = 0, slope = 1, color = "red", linetype = "dashed") +
  labs(title = "Scatterplot of Actual vs. Predicted TotalCost",
       x = "Actual TotalCost",
       y = "Predicted TotalCost") +
```

```
  scale_color_discrete(name = "KeywordGroup")

#testing performance of your model
library(caret)

#doing the Confusion Matrix
confusionMatrix(data=as.factor(as.numeric(my_prediction>0.5)),
                reference=as.factor(as.numeric(test_data$Bookings_Status)))

comparison_with_keyword <- data.frame(
  Actual = test_data$TotalCost,
  Predicted = my_prediction,
  residuals = test_data$TotalCost - my_prediction, # calculate residuals
  Keyword = test_data$Keyword
)
# Create a new data frame with the KeywordGroup variable from test_data
comparison_with_keyword_and_group <- cbind(comparison_with_keyword, KeywordGroup =
test_data$KeywordGroup)

# Get the top 5 keyword groups by total cost
top_5_keyword_groups <- test_data %>%
  group_by(KeywordGroup) %>%
  summarize(TotalCost = sum(TotalCost)) %>%
  top_n(5, TotalCost) %>%
  pull(KeywordGroup)

# Filter the comparison_with_keyword_and_group data frame to include only the top 5 keyword groups
comparison_with_keyword_top_5_groups <- comparison_with_keyword_and_group %>%
  filter(KeywordGroup %in% top_5_keyword_groups)
```

Scatterplot of Actual vs. Predicted TotalCost with top 5 Keyword Groups:

```
ggplot(comparison_with_keyword_top_5_groups, aes(x = Actual, y = Predicted, color = KeywordGroup))
+
 geom_point() +
 geom_abline(intercept = 0, slope = 1, color = "red", linetype = "dashed") +
 labs(title = "Scatterplot of Actual vs. Predicted TotalCost",
     x = "Actual TotalCost",
     y = "Predicted TotalCost") +
 scale_color_discrete(name = "KeywordGroup")
```



Analysis Details

Intro:
The case focuses on Air France and Media Contacts' search for the optimum SEM campaign to provide the most ticket sales per dollar spent on advertising. The foundation of this ideal search marketing strategy is the selection of efficient distribution of advertising budgets across the numerous search engines, as well as the choice of pertinent keywords and bid techniques for positioning online users on the search result page. It highlights how to drive analysis and enable comparison of multiple SEM campaigns, and one must first identify the key performance metrics for the project.

Insights:

1.Different publishers have different cost-per-click and conversion rates. We found that the logistic regression model provides a statistical relationship between the predictor variables and the outcome variable Bookings Status. It identifies potential customers who are likely to make a booking and targets marketing efforts toward them. This can optimize marketing budgets and increase the overall conversion rate of bookings. The coefficient estimations are detailed in the model summary. A significant coefficient with a positive value denotes a positive influence of the related predictor on the likelihood of Bookings Status. In contrast, a significant coefficient with a negative value denotes a negative influence. The independent variable, campaign status contains all active SEO activities, which indicates a higher success rate as of selling.

The active campaigns had a greater click-through rate (CTR) overall compared to the non-active efforts, indicating that they are often more successful in generating engagement. With a standard deviation of 1.37%, the mean click-through rate (CTR) was 3.13 percent overall.

2. According to the tree model, Amount is the most significant predictor, followed by AvgCostperClick and Clicks. With the use of this data, business choices may be made, such as allocating marketing resources to campaigns with higher projected Amount values or adjusting campaign bids to decrease the average cost per click and increase the number of clicks. This offers valuable information for strategic planning and decision-making in the corporate world, such as determining which campaigns are more likely to succeed and which require development.

3.We calculated the value counts of each value in the keyword variable, which provided insights into which keywords are most effective. We notice Google.com is the most lucrative channel, with a total of 157,688 reservations, With total sales of 157,947.1 USD, "Google.com" is followed by "Yahoo.com" 56794 bookings and 59,955.08 USD sales. In Scatterplot compares the actual and predicted total costs for the top 5 keyword groups: Air France Website, New York to France Sale, New York to Paris Sale, Nice, and Paris. We determine these keyword groups where the model performed well and those where it did not by examining the scatterplot. The model's predictions are reasonably accurate if the points are around the red dashed line. This analysis will help businesses determine which keyword groups are more cost-effective to invest in and which ones to avoid in their SEM campaigns. They can use this information to optimize

their ad campaigns and allocate resources more efficiently, resulting in a better return on investment (ROI).

To improve the performance of the live campaigns, we suggest business following recommendation:

Target a less competitive audience: For companies trying to grow their clientele and improve ROI, targeting a less competitive demographic might be a wise move. Businesses may lessen competition, boost conversion rates, and boost profitability by finding and focusing on a specialized audience. I'll talk about how to attract a less cutthroat audience in this answer, along with some advice on how to do it successfully.

Optimize the keywords and targeting: Each effective internet marketing strategy must optimize its keywords and targeting. It entails choosing the audience and keywords most likely to promote your company's brand and increase sales. I will go into keyword and audience targeting optimization for your internet marketing campaign in this response, along with some advice on how to advertise successfully to your target market.

Optimize the ad copy and landing pages: Each effective internet advertising strategy must optimize its landing pages and ad wording. A strong landing page and ad language may boost conversion rates, raise quality scores, and enhance click-through rates. I'll go through the essentials of optimizing ad content and landing pages in this reply, as well as offer advice on how to write ads that work and build landing pages that convert well.

Track the results of the campaigns and make adjustments as needed: In order to make sure that your marketing initiatives are successful and yielding the expected outcomes, tracking the results of your campaigns is a crucial step. By monitoring, you may determine what aspects of your campaigns are doing effectively and what aspects require improvement. This allows you to tweak and improve your campaigns to be as effective as possible. I'll review how to evaluate your campaign’s success in this reply and suggest any necessary changes.

Conclusion: The case study on Air France's search engine marketing campaign highlights the importance of data analysis and statistical models to make informed business decisions. Businesses may improve their marketing efforts to achieve their intended results while saving expenses by examining data from numerous sources and determining critical performance measures. Making wise decisions was possible using the logistic regression and decision tree models, which offered helpful insights into the importance of various predictor factors in driving bookings. Our study helped businesses enhance conversion rates and optimize marketing spending by identifying the most efficient publications and keywords. According to the results, spending money on Google.com, which has a high click-through rate, and devoting resources to campaigns with more significant anticipated amounts, might increase the likelihood of successfully generating bookings. In general, data-driven decision-making may result in a more prosperous and efficient SEM campaign, giving businesses a larger return on their investment.