Predicting Airline Customer Satisfaction

Elif KARTAL

2024-05-31

Predicting airline customer satisfaction enhances customer loyalty by increasing the likelihood of repeat business and aids in acquiring new customers through word-of-mouth marketing. High customer satisfaction provides a competitive advantage, increasing market share and supporting differentiation. It improves financial performance, generates additional revenue, and reduces operational costs. Moreover, it identifies operational inefficiencies and enhances service quality. It personalizes the customer experience and allows for proactive improvements by predicting potential dissatisfaction. It contributes to data-driven decision-making processes, facilitates the setting of strategic goals, and ensures compliance with regulatory requirements. Ultimately, it strengthens brand reputation and promotes sustainable growth.

Features

- **id :** Unique id number to each passenger. (numerical)
- **Gender:** Gender of the passengers (Female, Male) (categorical)
- **Customer Type:** The customer type (Loyal customer, disloyal customer) (categorical)
- **Age:** The actual age of the passengers (numerical)
- **Type of Travel:** Purpose of the flight of the passengers (Personal Travel, Business Travel) (categorical)
- **Class:** Travel class in the plane of the passengers (Business, Eco, Eco Plus) (categorical)
- **Flight distance:** The flight distance of this journey (numerical)
- **Inflight wifi service:** Satisfaction level of the inflight * wifi service (0:Not Applicable;1-5) (numerical)
- **Departure/Arrival time convenient:** Satisfaction level of Departure/Arrival time convenient (numerical)
- **Ease of Online booking:** Satisfaction level of online booking (numerical)
- **Gate location:** Satisfaction level of Gate location (numerical)
- **Food and drink:** Satisfaction level of Food and drink (numerical)
- **Online boarding:** Satisfaction level of online boarding (numerical)
- **Seat comfort:** Satisfaction level of Seat comfort (numerical)
- **Inflight entertainment:** Satisfaction level of inflight entertainment (numerical)
- **On-board service:** Satisfaction level of On-board service (numerical)
- **Leg room service:** Satisfaction level of Leg room service (numerical)
- **Baggage handling:** Satisfaction level of baggage handling (numerical)
- **Check-in service:** Satisfaction level of Check-in service (numerical)
- **Inflight service:** Satisfaction level of inflight service (numerical)
- **Cleanliness:** Satisfaction level of Cleanliness (numerical)

- **Departure Delay in Minutes:** Minutes delayed when departure (numerical)
- **Arrival Delay in Minutes**: Minutes delayed when Arrival (numerical)
- **Satisfaction:** Airline satisfaction level(Satisfaction, neutral or dissatisfaction)(categorical)

Needed Packages:

```
install.packages("caret")
install.packages("randomForest")
install.packages("ranger")
install.packages("dplyr")
install.packages("rpart")
install.packages("rpart.plot")
library(rpart)
library(rpart.plot)
library(dplyr)
library(ranger)
library(caret)
library(randomForest)
library(readr)
```

```
airline_s <- read_csv("airline_s.csv")
```

Checking the missing values:

```
anyNA(airline_s)
```

```
## [1] TRUE
```

Omitting missing values:

```
airline_s <- na.omit(airline_s)
```

Reducing the size of data for ease of processing:

```
set.seed(333)
data <- airline_s[sample(nrow(airline_s),3000),]
```

Changing the name of the target variable and its levels:

```
colnames(data)[colnames(data) == "satisfaction"] <- "target"
data <- data|>
  mutate(target= recode(target, satisfied = "1",
                        "neutral or dissatisfied"="0"))
```

Check summary statistics of the data:

```
summary(data)
```

```
##        id            Gender          Customer Type           Age      
##  Min.   :    49   Length:3000        Length:3000        Min.   : 7.0  
##  1st Qu.: 34598   Class :character   Class :character   1st Qu.:28.0  
##  Median : 66620   Mode  :character   Mode  :character   Median :40.0  
##  Mean   : 66111                                         Mean   :39.8  
##  3rd Qu.: 98879                                         3rd Qu.:51.0  
##  Max.   :129868                                         Max.   :85.0  
##  TypeofTravel          Class           FlightDistance Inflightwifiservice
##  Length:3000        Length:3000        Min.   :  31   Min.   :0.000      
##  Class :character   Class :character   1st Qu.: 423   1st Qu.:2.000      
##  Mode  :character   Mode  :character   Median : 850   Median :3.000      
##                                         Mean   :1191   Mean   :2.688
##                                         3rd Qu.:1751   3rd Qu.:4.000
##                                         Max.   :4983   Max.   :5.000
##  Departure/Arrivaltimeconvenient EaseofOnlinebooking  Gatelocation
##  Min.   :0.000                   Min.   :0.000       Min.   :1.000
##  1st Qu.:2.000                   1st Qu.:2.000       1st Qu.:2.000
##  Median :3.000                   Median :3.000       Median :3.000
##  Mean   :3.046                   Mean   :2.708       Mean   :2.959
##  3rd Qu.:4.000                   3rd Qu.:4.000       3rd Qu.:4.000
##  Max.   :5.000                   Max.   :5.000       Max.   :5.000
##   Foodanddrink   Onlineboarding   Seatcomfort    Inflightentertainment
##  Min.   :0.000   Min.   :0.000   Min.   :1.000   Min.   :1.000
##  1st Qu.:2.000   1st Qu.:2.000   1st Qu.:3.000   1st Qu.:2.000
##  Median :3.000   Median :3.000   Median :4.000   Median :4.000
##  Mean   :3.226   Mean   :3.229   Mean   :3.466   Mean   :3.386
##  3rd Qu.:4.000   3rd Qu.:4.000   3rd Qu.:5.000   3rd Qu.:4.000
##  Max.   :5.000   Max.   :5.000   Max.   :5.000   Max.   :5.000
##  On-boardservice Legroomservice  Baggagehandling Checkinservice Inflightservice
##  Min.   :1.000   Min.   :0.000   Min.   :1.000   Min.   :1.00   Min.   :1.000
##  1st Qu.:2.000   1st Qu.:2.000   1st Qu.:3.000   1st Qu.:3.00   1st Qu.:3.000
##  Median :4.000   Median :4.000   Median :4.000   Median :3.00   Median :4.000
##  Mean   :3.402   Mean   :3.313   Mean   :3.637   Mean   :3.31   Mean   :3.659
##  3rd Qu.:4.000   3rd Qu.:4.000   3rd Qu.:4.000   3rd Qu.:4.00   3rd Qu.:5.000
##  Max.   :5.000   Max.   :5.000   Max.   :5.000   Max.   :5.00   Max.   :5.000
##   Cleanliness    DepartureDelayinMinutes ArrivalDelayinMinutes
##  Min.   :1.000   Min.   :  0.0           Min.   :  0.00
##  1st Qu.:2.000   1st Qu.:  0.0           1st Qu.:  0.00
##  Median :3.000   Median :  0.0           Median :  0.00
##  Mean   :3.317   Mean   : 13.6           Mean   : 14.08
##  3rd Qu.:4.000   3rd Qu.: 11.0           3rd Qu.: 12.00
##  Max.   :5.000   Max.   :565.0           Max.   :586.00
##     target
##  Length:3000
##  Class :character
##  Mode  :character
##
##
##
```

Check structure of the data:

```
str(data)
```

```
## tibble [3,000 x 24] (S3: tbl_df/tbl/data.frame)
##  $ id                             : num [1:3000] 14824 25543 24702 36609 22936 ...
##  $ Gender                         : chr [1:3000] "Male" "Male" "Male" "Female" ...
##  $ Customer Type                  : chr [1:3000] "Loyal Customer" "Loyal Customer" "Loyal Customer"
##  $ Age                            : num [1:3000] 50 28 35 41 40 33 18 30 51 71 ...
##  $ TypeofTravel                   : chr [1:3000] "Personal Travel" "Personal Travel" "Business travel
##  $ Class                          : chr [1:3000] "Business" "Eco" "Eco" "Eco" ...
##  $ FlightDistance                 : num [1:3000] 314 547 397 1197 817 ...
##  $ Inflightwifiservice            : num [1:3000] 1 3 2 1 4 3 2 4 5 2 ...
##  $ Departure/Arrivaltimeconvenient: num [1:3000] 4 4 1 1 3 3 5 4 5 3 ...
##  $ EaseofOnlinebooking            : num [1:3000] 1 3 1 1 3 3 2 4 5 3 ...
##  $ Gatelocation                   : num [1:3000] 2 4 1 4 3 4 3 4 5 3 ...
##  $ Foodanddrink                   : num [1:3000] 4 5 2 5 4 5 1 5 5 3 ...
##  $ Onlineboarding                 : num [1:3000] 4 3 2 1 4 3 2 4 5 2 ...
##  $ Seatcomfort                    : num [1:3000] 5 2 2 5 4 5 3 5 5 3 ...
##  $ Inflightentertainment          : num [1:3000] 3 5 2 5 4 5 1 5 5 2 ...
##  $ On-boardservice                : num [1:3000] 3 2 1 4 2 3 1 5 2 2 ...
##  $ Legroomservice                 : num [1:3000] 1 5 4 4 4 3 1 2 1 2 ...
##  $ Baggagehandling                : num [1:3000] 3 4 4 4 1 3 4 4 2 2 ...
##  $ Checkinservice                 : num [1:3000] 4 1 2 1 2 2 3 5 5 3 ...
##  $ Inflightservice                : num [1:3000] 3 3 3 4 4 4 2 5 1 2 ...
##  $ Cleanliness                    : num [1:3000] 4 5 2 5 4 5 1 5 5 3 ...
##  $ DepartureDelayinMinutes        : num [1:3000] 0 3 0 0 19 0 0 5 0 12 ...
##  $ ArrivalDelayinMinutes          : num [1:3000] 0 3 6 18 5 0 0 6 0 23 ...
##  $ target                         : chr [1:3000] "0" "0" "0" "0" ...
##  - attr(*, "na.action")= 'omit' Named int [1:83] 517 657 1072 1225 1590 1817 1833 2772 2912 3195 ...
##   ..- attr(*, "names")= chr [1:83] "517" "657" "1072" "1225" ...
```

Subsetting data into two parts as traindata and testdata:

```
set.seed(333)
index <- sample( nrow(data), round(nrow(data)*0.80))
traindata <- data[index,]
testdata <- data[-index,]
```

Checking balance of the traindata:

```
class_table <- table(traindata$target)
class_table
```

```
##
##    0    1
## 1361 1039
```

There are 1361 observations in the dissatisfied (0) class and 1039 observations in the satisfied (1) class.

Class Propotions:

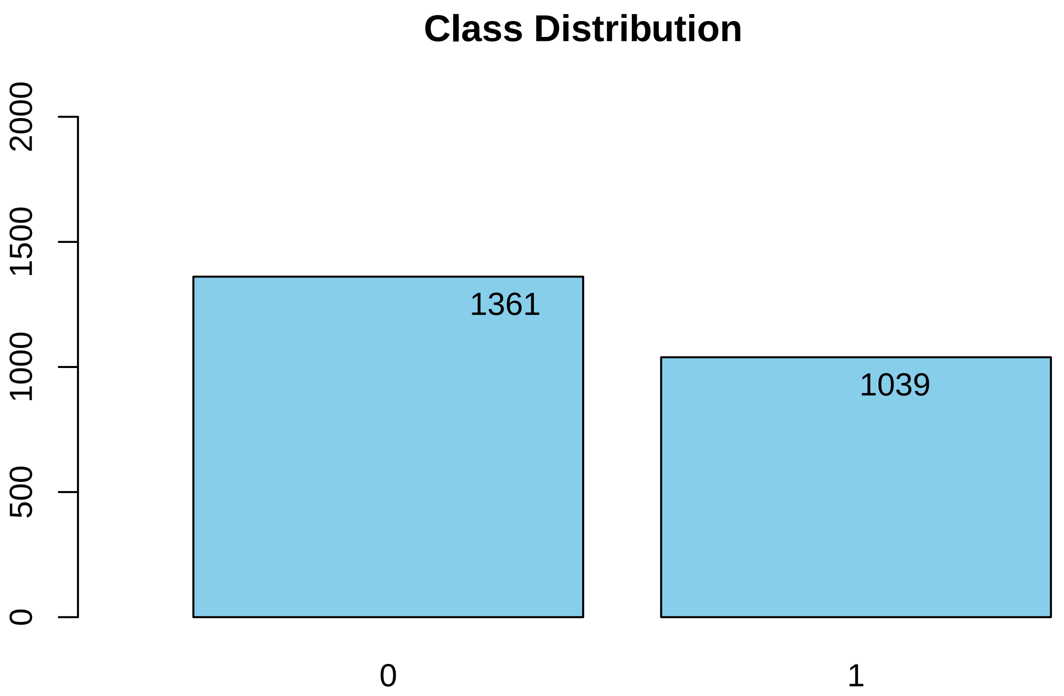
```
class_proportions <- prop.table(class_table )
class_proportions
```

```
##
##         0         1
## 0.5670833 0.4329167
```

While the dissatisfied (0) class constitutes 56.7% of all data, the satisfied (1) class constitutes 43.3% of all data.

Visualization of the imbalanced target variable:

```
barplot(class_table , main="Class Distribution",
        ylim = c(0, max(class_table ) * 1.5),
        xlim = c(0, length(class_table ) * 1.2),
        col = "skyblue")
text(x = 1:length(class_table ), y = class_table  + 0.5,
     labels = class_table , pos = 1)
```



By looking chart and proportions we can understand that the target variable in the traindata has imbalanced distribution.

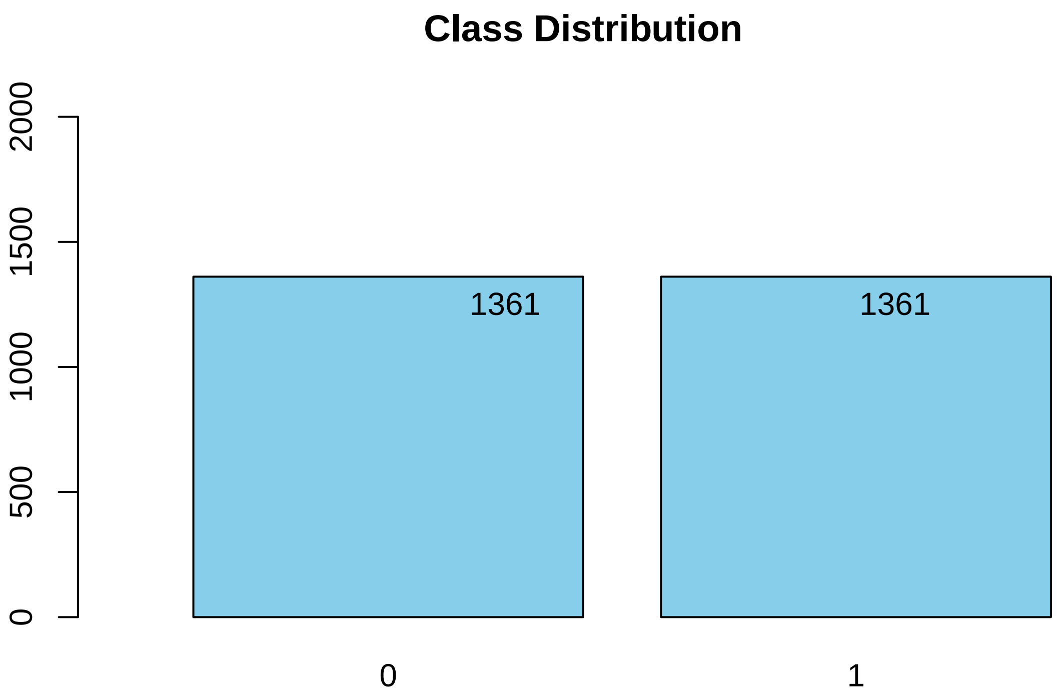
Fixing the imbalance:

In order to fix imbalance problem oversampling method is used.

```
majority <- traindata |> filter(target == "0")
minority <- traindata |> filter(target == "1")
oversampled_minority <- minority[sample(1:nrow(minority),
                                        nrow(majority),
                                        replace = TRUE), ]
oversampled_data_random <- rbind(majority, oversampled_minority)
balanced_table <- table(oversampled_data_random$target)
```

Visualization of the distribution of the balanced target variable:

```
barplot(balanced_table , main="Class Distribution",
        ylim = c(0, max(balanced_table ) * 1.5),
        xlim = c(0, length(balanced_table ) * 1.2),
        col = "skyblue")
text(x = 1:length(balanced_table ), y = balanced_table  + 0.5,
     labels = balanced_table , pos = 1)
```



Balanced proportions:

```
class_proportions2 <- prop.table(balanced_table )
class_proportions2
```

```
##
##   0   1
## 0.5 0.5
```

The number of observations in the classes is equalized and the imbalance problem is solved.

Changing the name of oversampled data:

```
balanced_train <- oversampled_data_random
```

Logistic regression:

```
set.seed(333)
glm_model <- glm(as.numeric(target) ~.,
                 data = balanced_train,
                 family = "binomial")
summary(glm_model)
```

```
##
## Call:
## glm(formula = as.numeric(target) ~ ., family = "binomial", data = balanced_train)
##
## Coefficients:
##                                       Estimate Std. Error z value Pr(>|z|)
## (Intercept)                         -6.647e+00  4.740e-01 -14.024  < 2e-16 ***
## id                                  -7.418e-06  1.647e-06  -4.504 6.66e-06 ***
## GenderMale                           4.238e-01  1.190e-01   3.561 0.000370 ***
## `Customer Type`Loyal Customer        1.989e+00  1.787e-01  11.132  < 2e-16 ***
## Age                                 -1.755e-02  4.459e-03  -3.935 8.33e-05 ***
## TypeofTravelPersonal Travel       -2.821e+00  1.928e-01 -14.633  < 2e-16 ***
## ClassEco                          -6.105e-01  1.576e-01  -3.873 0.000107 ***
## ClassEco Plus                     -8.399e-01  2.620e-01  -3.206 0.001348 **
## FlightDistance                    -3.530e-05  7.055e-05  -0.500 0.616893
## Inflightwifiservice                3.391e-01  7.053e-02   4.809 1.52e-06 ***
## `Departure/Arrivaltimeconvenient` -1.462e-01  5.009e-02  -2.918 0.003522 **
## EaseofOnlinebooking               -2.596e-01  6.729e-02  -3.858 0.000114 ***
## Gatelocation                       1.293e-01  5.824e-02   2.220 0.026439 *
## Foodanddrink                      -4.441e-02  6.856e-02  -0.648 0.517163
## Onlineboarding                     5.372e-01  6.206e-02   8.655  < 2e-16 ***
## Seatcomfort                        1.184e-01  6.532e-02   1.813 0.069806 .
## Inflightentertainment             -7.408e-03  8.885e-02  -0.083 0.933557
## `On-boardservice`                  3.589e-01  6.061e-02   5.923 3.17e-09 ***
## Legroomservice                     2.036e-01  5.019e-02   4.056 4.99e-05 ***
## Baggagehandling                    4.752e-03  6.647e-02   0.071 0.943003
## Checkinservice                     3.774e-01  5.224e-02   7.225 5.01e-13 ***
## Inflightservice                    2.486e-01  7.474e-02   3.327 0.000879 ***
## Cleanliness                        2.620e-01  7.572e-02   3.460 0.000541 ***
## DepartureDelayinMinutes            1.820e-02  6.122e-03   2.973 0.002950 **
## ArrivalDelayinMinutes             -2.284e-02  6.047e-03  -3.777 0.000159 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 3773.5  on 2721  degrees of freedom
## Residual deviance: 1916.3  on 2697  degrees of freedom
## AIC: 1966.3
##
## Number of Fisher Scoring iterations: 5
```

By looking at the coefficients of the variables, it was decided to remove these variables to increase the performance of the model:

- id
- TypeofTravel
- Inflightentertainment
- FlightDistance

Predicting logistic regression model:

```r
glm_predict <- predict(glm_model, testdata)
glm_class <- ifelse(glm_predict > 0.5, 1,0)
confusionMatrix(as.factor(glm_class), as.factor(testdata$target))
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction   0   1
##          0 332  47
##          1  28 193
##
##                Accuracy : 0.875
##                  95% CI : (0.8458, 0.9004)
##     No Information Rate : 0.6
##     P-Value [Acc > NIR] : < 2e-16
##
##                   Kappa : 0.7361
##
##  Mcnemar's Test P-Value : 0.03767
##
##             Sensitivity : 0.9222
##             Specificity : 0.8042
##          Pos Pred Value : 0.8760
##          Neg Pred Value : 0.8733
##              Prevalence : 0.6000
##          Detection Rate : 0.5533
##    Detection Prevalence : 0.6317
##       Balanced Accuracy : 0.8632
##
##        'Positive' Class : 0
##
```

Removing some variables from the data to increase the performance of the model:

```r
balanced_train <-    balanced_train[, -which(names(balanced_train) == "id")]

balanced_train <-    balanced_train[, -which(names(balanced_train) == "TypeofTravel")]

balanced_train <-    balanced_train[, -which(names(balanced_train) ==
                                               "Inflightentertainment")]

balanced_train <-    balanced_train[, -which(names(balanced_train) == "FlightDistance")]
```

Random Forests:

```r
#tuning hyperparameters
tunegrid <- expand.grid(.mtry=c(1:sqrt(ncol(balanced_train))))
control2 <- trainControl(method="cv", number=10)


set.seed(333)
rf_gridsearch <- train(target ~ .,
                       data = balanced_train,
                       method = "rf",
                       tuneGrid = tunegrid,
                       trControl = control2,
                       ntree = 400,
                       nodesize = 10)

print(rf_gridsearch)
```

```
## Random Forest
##
## 2722 samples
##   19 predictor
##    2 classes: '0', '1'
##
## No pre-processing
## Resampling: Cross-Validated (10 fold)
## Summary of sample sizes: 2450, 2450, 2450, 2450, 2449, 2449, ...
## Resampling results across tuning parameters:
##
##   mtry  Accuracy   Kappa
##   1     0.9129323  0.8258728
##   2     0.9364455  0.8728965
##   3     0.9397517  0.8795057
##   4     0.9404897  0.8809834
##
## Accuracy was used to select the optimal model using the largest value.
## The final value used for the model was mtry = 4.
```

Predicting RandomForests Model:

```
predictions2 <- predict(rf_gridsearch, newdata = testdata)
confusionMatrix(as.factor(predictions2), as.factor(testdata$target))
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction   0   1
##          0 342  32
##          1  18 208
##
##                Accuracy : 0.9167
##                  95% CI : (0.8916, 0.9375)
##     No Information Rate : 0.6
##     P-Value [Acc > NIR] : < 2e-16
##
##                   Kappa : 0.8247
##
##  Mcnemar's Test P-Value : 0.06599
##
##             Sensitivity : 0.9500
##             Specificity : 0.8667
##          Pos Pred Value : 0.9144
##          Neg Pred Value : 0.9204
##              Prevalence : 0.6000
##          Detection Rate : 0.5700
##    Detection Prevalence : 0.6233
##       Balanced Accuracy : 0.9083
##
##        'Positive' Class : 0
##
```

Although random forests and logistic regression models are high-performance, they are disadvantageous in terms of interpretability. For this reason, it was decided to use the decision trees model, which provides easy interpretability.

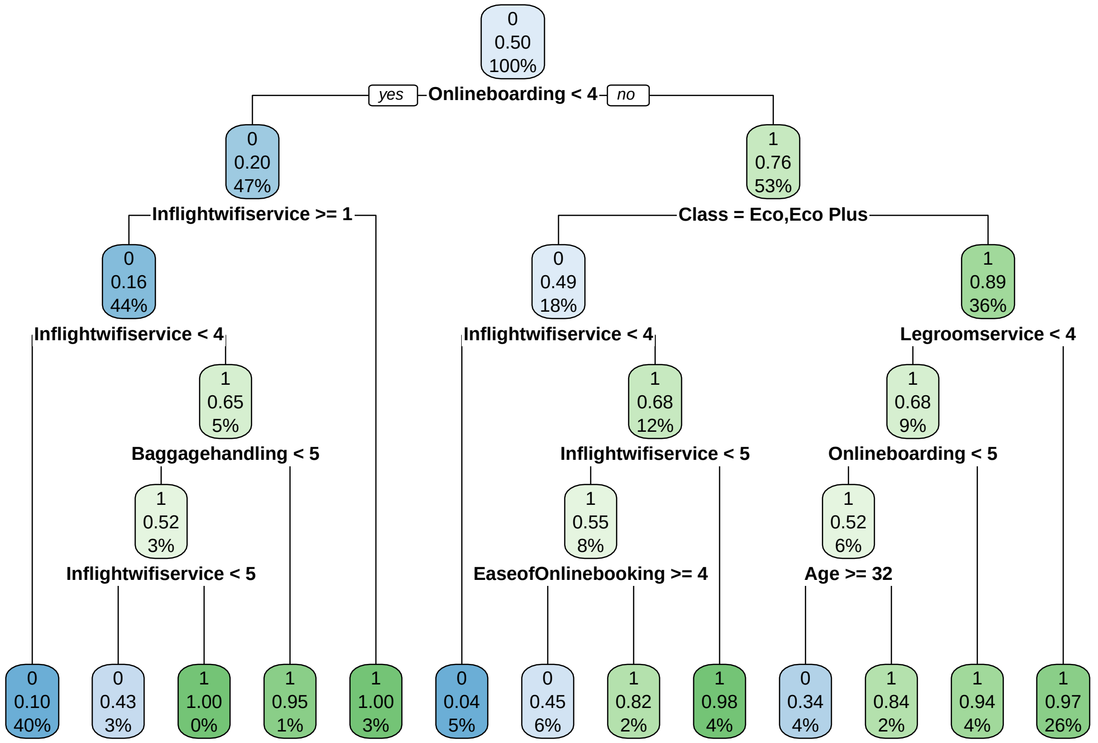
Decision Tree

```
set.seed(333)
dt_model2 <- rpart(target ~ .,
                  data = balanced_train,
                  method = "class",
                  control = rpart.control(
                    maxdepth = 5, # Maximum tree depth
                    minsplit = 30,# Minimum number of observations
                    minbucket = 5,# Minimum number of bucket
                    cp = 0.0020,  # complexity coefficient

                  ),
                  parms = list(split = "gini")
)
#Gini impurity refers to the probability of misclassification of a
#randomly selected sample at a node.
```

Visualizing the decision tree:

```
rpart.plot(dt_model2)
```



```
pred2 <- predict(dt_model2, testdata, type="class")

confusionMatrix(as.factor(pred2), as.factor(testdata$target))
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction   0   1
##          0 343  45
##          1  17 195
##
##                Accuracy : 0.8967
##                  95% CI : (0.8695, 0.9199)
##     No Information Rate : 0.6
##     P-Value [Acc > NIR] : < 2.2e-16
##
##                   Kappa : 0.7805
##
##  Mcnemar's Test P-Value : 0.0006058
##
##             Sensitivity : 0.9528
##             Specificity : 0.8125
##          Pos Pred Value : 0.8840
##          Neg Pred Value : 0.9198
##              Prevalence : 0.6000
##          Detection Rate : 0.5717
##    Detection Prevalence : 0.6467
##       Balanced Accuracy : 0.8826
##
##        'Positive' Class : 0
##
```

This decision tree represents a model that an airline company uses to predict customer satisfaction. The tree will attempt to predict whether the customer will be satisfied or not, based on a number of variables related to the customer's flight experience.

The tree consists of a series of nodes that branch off, each based on a question or decision. Each node represents a subset of customers. Branches show how customers are divided into subsets. The final nodes contain predictions about the satisfaction of customers in each subset.

Root Node

- **Onlineboarding < 4**: The root node of the decision tree starts with the `Onlineboarding` score. This score indicates customer satisfaction with the online boarding process.
    - If the score is less than 4, these customers go to the left branch.
    - If the score is 4 or greater, these customers go to the right branch.

Left Branch (Onlineboarding < 4)

1. **Inflightwifiservice >= 1**: The first split is based on the `Inflightwifiservice` score.
    - If the `Inflightwifiservice` score is 1 or greater, these customers go to the left branch.
    - If the `Inflightwifiservice` score is less than 1, these customers go to the right branch.
2. **Inflightwifiservice < 4**: At this point, customers with an `Inflightwifiservice` score less than 4 go to the left branch.
    - **Baggagehandling < 5**: Customers with a `Baggagehandling` score less than 5 go to the left branch.
        - Customers in this branch are generally dissatisfied (`0`).
    - **Inflightwifiservice < 5**: Customers with an `Inflightwifiservice` score less than 5 go to the right branch.
        - Customers in this branch are generally satisfied (`1`).

Right Branch (Onlineboarding >= 4)

1. **Class = Eco, Eco Plus**: In this branch, customers traveling in economy or economy plus class go to the left branch.
    - **Inflightwifiservice < 4**: Customers with an `Inflightwifiservice` score less than 4 go to the left branch.
        - **EaseofOnlinebooking >= 4**: Customers with an `EaseofOnlinebooking` score of 4 or greater go to the left branch.
            - Customers in this branch are generally satisfied (`1`).

    - **Age >= 32**: Customers aged 32 or older go to the right branch.
        - Customers in this branch are generally satisfied (`1`).
- **Legroomservice < 4**: Customers with a `Legroomservice` score less than 4 go to the right branch.
    - Customers in this branch are generally satisfied (`1`).

General Interpretation

1. **Onlineboarding**: Serving as the first split, the online boarding process significantly impacts customer satisfaction. Customers with lower online boarding scores are generally dissatisfied.
2. **Inflightwifiservice**: The second most important factor is the quality of in-flight Wi-Fi service, which greatly influences satisfaction.
3. **Class**: The flight class (economy or economy plus) also affects satisfaction, with economy class customers typically showing more dissatisfaction.
4. **Baggagehandling**: The quality of baggage handling is another determinant of customer satisfaction.
5. **EaseofOnlinebooking**: The ease of online booking is particularly important for customers with certain Wi-Fi service scores.
6. **Age**: The age of the customer can also impact satisfaction in some scenarios.

Recommendations to Improve Customer Satisfaction

1. **Improve Online Boarding Process**: Making the online boarding process more user-friendly can enhance customer satisfaction.
2. **Enhance In-Flight Wi-Fi Service**: Improving the quality of Wi-Fi service can significantly boost satisfaction levels.
3. **Streamline Baggage Handling**: Efficient and smooth baggage handling processes can enhance customer satisfaction.
4. **Simplify Online Booking System**: Making the booking process easier can improve satisfaction, especially for those who rely on the internet.
5. **Improve Legroom Service**: Enhancing legroom service quality can particularly boost satisfaction for specific customer segments.

These insights can help the airline company identify areas for improvement to enhance overall customer satisfaction.