Final Project:
Airline Satisfaction

Dr. Xiyue Liao

STAT 473

Cindy Acuna, Luis Osorio,

Maria Sanchez-Beltran & Tammy Huynh

Introduction

Businesses conduct surveys on their customers to get a better understanding of satisfaction levels which can lead to improvements in a product or service. In the airline industry it is fundamental to keep track of customers expectations and focus on service quality levels to keep user engagement and retention. Given a dataset that contains observations from surveys conducted in 2015 provides passenger satisfaction measurements. We aim to identify key components that heavily influence satisfaction levels using different machine learning algorithms. For interoperability we aim to use algorithms such as Logistic Regression, Random Forest Classifier, and Boosting to predict a binary classification problem. The outcome is to return whether a passenger was satisfied with their flight or neutral/dissatisfied with their travel. Also given a slightly imbalanced target, how does each model perform despite learning different class sizes? We also wish to determine the best performing model in terms of accuracy for this particular data set.

Question of Interest

In this paper, we hope to answer the following questions:

1. What are the statistically significant predictors for each of the machine learning models?

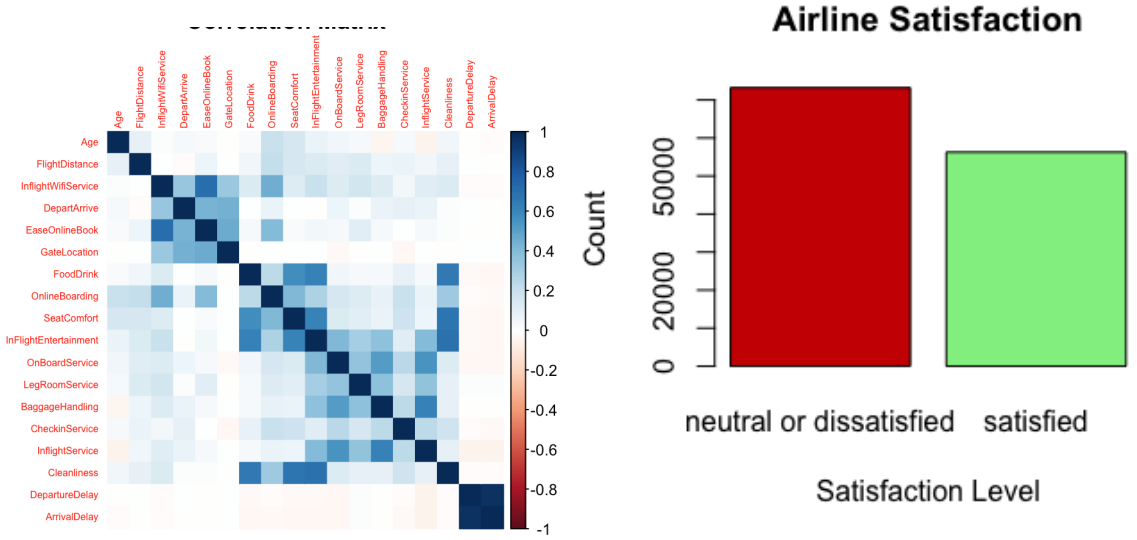
2. How does each model handle imbalance classes in the response variable?

3. What machine learning algorithm produced the highest accuracy in determining airline satisfaction level?

Analysis

Observations

First thing we observed during the exploratory data analysis stage is that we are given a data set that contains a total of 129,880 observations and 24 variables. Some of the categorical variables include satisfaction levels, inflight wifi service, customer type, travel type, gender and class. Some of the numeric columns included flight distance, age, departure delay time, and arrival delay time. We investigated for any missing values which we counted to be a column with 393 null values where we ended up dropping the rows since we had a relatively large sample size. We checked that our columns had the adequate data types and in the process converted our target variable into a factor, which represents a categorical variable and stores it on multiple levels. We also ended up changing the column names for simplicity and for variable name conventions. After the data analysis our next step was to use visualization to gain further data insights.



Once we plotted a bar chart with the different levels of satisfaction we immediately saw an imbalanced target variable where we had 73,225 neutral/dissatisfied passengers and 56,262 satisfied passengers. We also observed that a majority of passengers bought business and eco fares far more compared to the eco plus fare. Where the genders attribute provided an even distribution of males and females passengers and where flight distance was usually no more than 1,000 kilometers. We plotted a correlation matrix where we saw Seat Comfort had a positive

relationship between food/drinks, in flight entertainment, and cleanliness. We can also see On-board service also had a positive relationship with leg room service, baggage handling, in flight service, and in flight entertainment. We then split the data for modeling training where we performed an 80/20 split for our train and testing sets.

Model 1 - Logistic Regression

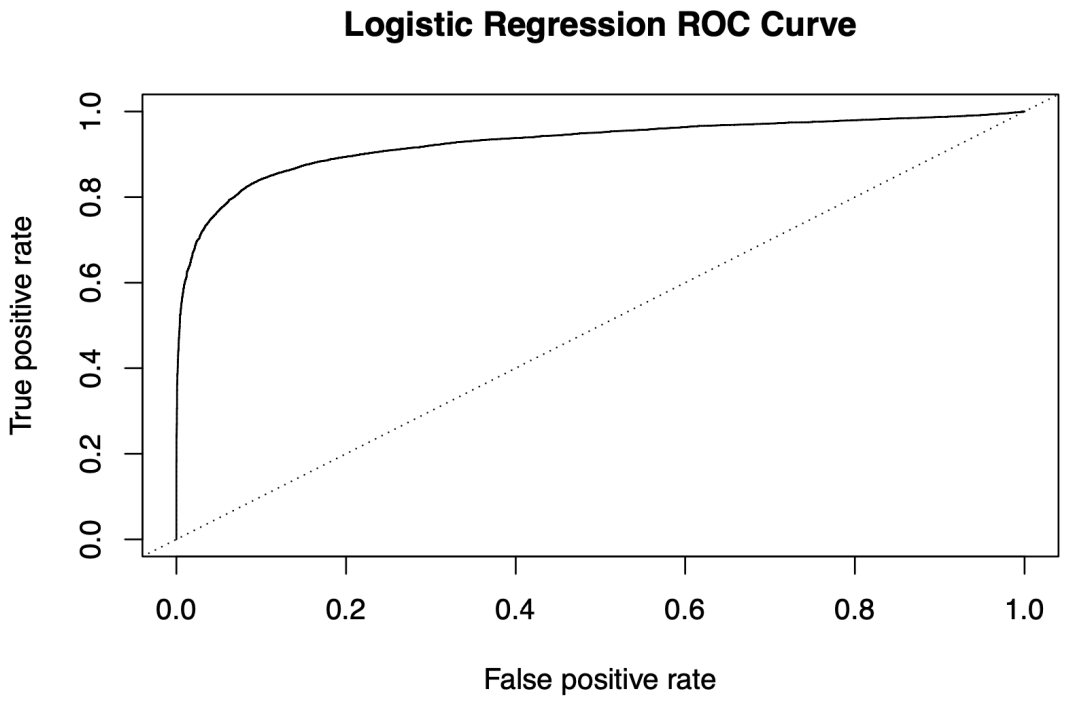
Next was the modeling part where we began with logistic regression using all the predictors to predict the target, satisfaction. To answer the first research question in identifying the significant predictors we performed a t-test on the slope parameters to check if there was a relationship. We conduct a hypothesis test where our null hypothesis assumes that the coefficient of the predictor is equal to zero, hence no relationship. Our alternative hypothesis states that the slope parameter for the predictor is not equal to zero and assumes the predictor is significant for our model. Setting a significance level of five percent we checked for the predictors that returned a p-value greater than our threshold.

```
Coefficients:
                            Estimate Std. Error z value Pr(>|z|)
(Intercept)               -7.835e+00  7.864e-02 -99.633  < 2e-16 ***
GenderMale                 6.947e-02  1.945e-02   3.573 0.000353 ***
CustomerTypeLoyal Customer 2.041e+00  2.978e-02  68.541  < 2e-16 ***
Age                       -8.856e-03  7.114e-04 -12.450  < 2e-16 ***
TypeTravelPersonal Travel -2.737e+00  3.150e-02 -86.905  < 2e-16 ***
ClassEco                  -7.135e-01  2.556e-02 -27.920  < 2e-16 ***
ClassEco Plus             -8.388e-01  4.162e-02 -20.155  < 2e-16 ***
FlightDistance            -1.920e-05  1.129e-05  -1.702 0.088821 .
InflightWifiService        3.971e-01  1.144e-02  34.697  < 2e-16 ***
DepartArrive              -1.362e-01  8.191e-03 -16.623  < 2e-16 ***
EaseOnlineBook            -1.557e-01  1.130e-02 -13.774  < 2e-16 ***
GateLocation               3.703e-02  9.148e-03   4.048 5.17e-05 ***
FoodDrink                 -3.137e-02  1.070e-02  -2.933 0.003361 **
OnlineBoarding             6.112e-01  1.025e-02  59.617  < 2e-16 ***
SeatComfort                6.034e-02  1.119e-02   5.392 6.96e-08 ***
InFlightEntertainment      5.603e-02  1.426e-02   3.928 8.56e-05 ***
OnBoardService             3.041e-01  1.018e-02  29.875  < 2e-16 ***
LegRoomService             2.504e-01  8.531e-03  29.350  < 2e-16 ***
BaggageHandling            1.360e-01  1.144e-02  11.890  < 2e-16 ***
CheckinService             3.298e-01  8.563e-03  38.520  < 2e-16 ***
InflightService            1.271e-01  1.203e-02  10.572  < 2e-16 ***
Cleanliness                2.319e-01  1.209e-02  19.178  < 2e-16 ***
DepartureDelay             4.334e-03  9.924e-04   4.367 1.26e-05 ***
ArrivalDelay              -9.195e-03  9.812e-04  -9.371  < 2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Flight distance was the only predictor that failed to reject the null hypothesis thus it was a statistically insignificant predictor for our target. Now to determine the most significant predictors we check the p-values for each one, where the smaller the value the more significant the predictor. Next, we computed a confusion matrix using the testing set which helped check the performance of the logistic model using only the significant predictors. The logistic regression model returned scores of accuracy 87.5%, sensitivity 90%, and specificity 83.6%.

```
##                         true_status
## predict_status          neutral or dissatisfied satisfied
##   neutral/dissatisfied                   13250      1834
##   satisfied                               1393      9420
```

Sensitivity takes into account the true positives which was the prediction that the passengers were neutral/dissatisfied, predicted accurately. Recall, we had an imbalanced data set where we had more neutral/dissatisfied passengers compared to satisfied passengers which explains why our sensitivity rate was greater than our specificity rate. Therefore our model predicts neutral/dissatisfied passengers better than predicting satisfied customers which answers our second research question. Down below is the ROC curve for Logistic Regression with an area under curve of 92.7%.



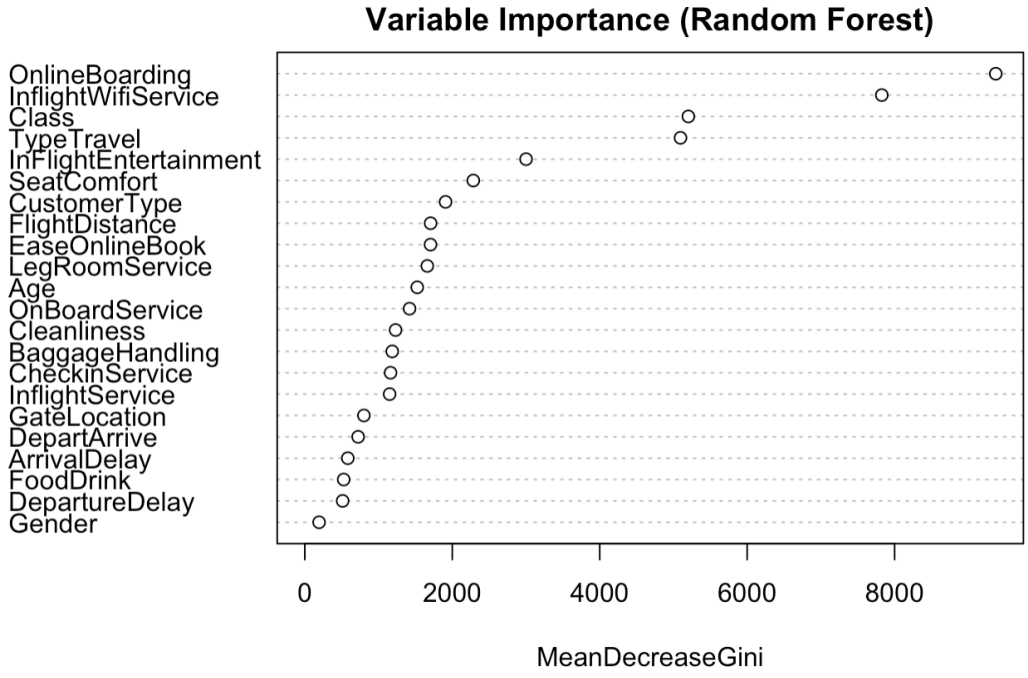
Model 2 - Random Forest Classifier

Our second algorithm that we decided to use was Random Forest Classifier. Setting the random seed to be 123 for our algorithm, our training data was split into 500 trees. After fitting our data, we found its corresponding confusion matrix on the testing set. The following was the output:

```
                         true_status
predict_status           neutral or dissatisfied satisfied
  neutral or dissatisfied                   14392       649
  satisfied                                   251     10605
```

If we count both categories, we can see that we have 14,392 neutral or dissatisfied cases and 11,254 satisfied cases that were predicted accurately. Since we have more neutral or dissatisfied cases, our sensitivity rate will be higher than our specificity. From the confusion matrix we calculate the accuracy of 96.5%, sensitivity of 98.2%, and specificity of 94.2%. Observe that the discrepancy between the sensitivity and specificity for random forest wasn't as great compared to the logistic regression model. Hence, we can say our random forest classifier did a better job in learning both classes of our target, which may be due to splitting of the decision trees. This answers our second research question and found that our random forest does fairly well on imbalanced classes.

To answer the first question, we created a Variable Importance Plot to see which predictors are most significant when using Random Forest.



The mean decrease in Gini coefficient is a measure of how each variable contributes to the homogeneity of the nodes and leaves in the resulting random forest. The higher the value of mean decrease accuracy or mean decrease Gini score, the higher the importance of the variable in the model. From the plot above, we can see that the most significant predictors in order are OnlineBoarding, InflightWifiService, Class and TypeTravel. These findings make sense because they all have a strong correlation to the overall satisfaction of an airline passenger, answering our first question.

Model 3 - Boosting Algorithm

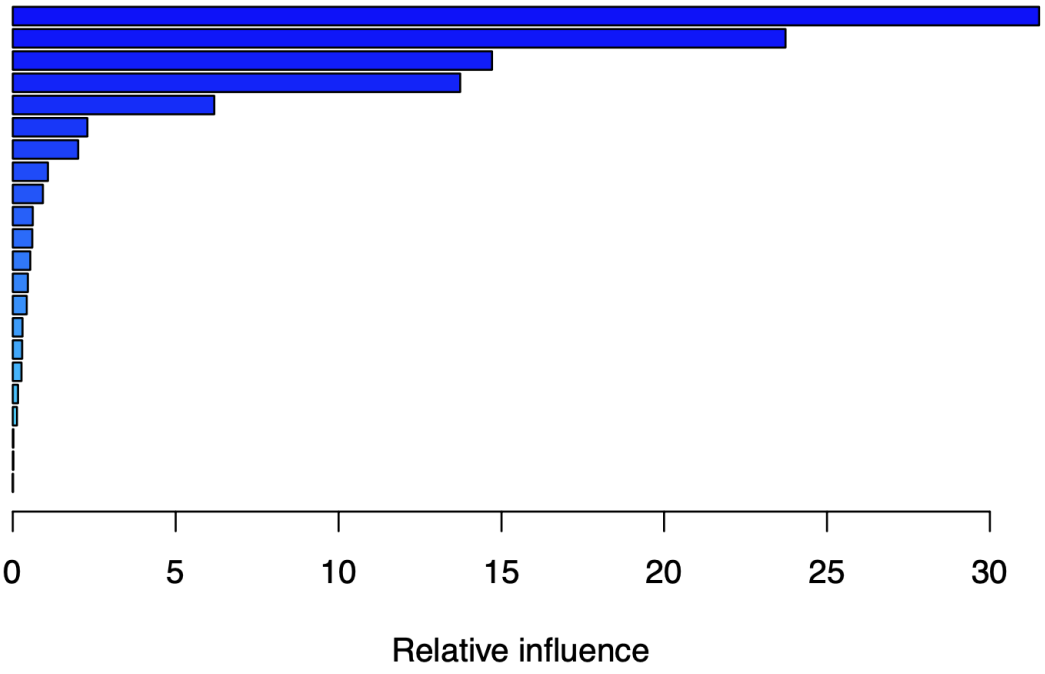
The final model that we implemented was the boosting algorithm, where we had to label encode our target variable into a numeric attribute. First, we ran a grid search to look for the most optimal parameters for our algorithm in terms of number of trees estimators, depth of the tree, and its learning rate. After finding the best parameters for our boosting model, the resulting confusion matrix on the testing set gave

```
Using 200 trees...

    true
pred     0     1
   0 14315   772
   1   419 10391

The Accuracy is: 0.9540101
The Sensitivity is: 0.9715624
The Specificity is: 0.930843
```

Again we can clearly see that our sensitivity rate was greater than our specificity metric. However, again compared to the logistic regression model the difference between the two metrics wasn't as great. Hence, we can also conclude that the boosting model handled the imbalanced classes well. This may be due to again the splitting of the trees which can handle imbalances pretty good. Next we aim to answer the first question for our boosting model and check for variable importance.



	var <chr>	rel.inf <dbl>
OnlineBoarding	OnlineBoarding	31.510913934
InflightWifiService	InflightWifiService	23.727254297
TypeTravel	TypeTravel	14.709324024
Class	Class	13.732213317
InFlightEntertainment	InFlightEntertainment	6.183822680
LegRoomService	LegRoomService	2.290264611
CustomerType	CustomerType	2.007891072
CheckinService	CheckinService	1.079360599
OnBoardService	OnBoardService	0.925417062
BaggageHandling	BaggageHandling	0.614632499

From the table above, we can see the influence of each predictor where the higher the score the more important the feature is to the model. The variable importance here was quite similar to importance feature plot for the random forest with the difference that class was more important than type of travel in the random forest and was vice versa for boosting. The table above provides a list of the top 10 important features for the boosting algorithm and answers the second research question.

Now, random forest was the top performer in predicting satisfaction levels in terms of accuracy with a 96.5% and our second best model was boosting with an accuracy of 95.4%. The lowest accuracy score was our logistic regression model with 87.5% accuracy, however it's important to note that out of the three models logistic regression is the most flexible model. If the business only cares about accurately predicting a satisfied passenger we can change the logistic threshold for the probabilities and increase specificity rate. Sometimes sacrificing accuracy for interoperability is preferred when it comes down to business sense. Finally, answering our final research question.

Conclusion

We observed the data and used logistic regression, Random Forest, and the boosting algorithm to make models and answer our three research questions. We determined that the statistically significant predictors for the logistic regression were every predictor except for FlightDistance. The most significant predictors would have the lowest p-value. For the Random Forest, we found that the most significant predictors were OnlineBoarding, InflightWifiService, Class, and TypeTravel by comparing the Gini coefficient. For the boosting algorithm, the most significant predictors were the same as the Random Forest. Next, we considered how each model handles imbalanced classes in the response variable by analyzing the accuracy, specificity, and

sensitivity of each model. For the logistic regression model, the slight unbalanceness of our data set is reflected with the sensitivity being 6.4% larger than the specificity. For our Random Forest model, the discrepancy between specificity and sensitivity is 4% (both being higher than the logistic regression outputs), so this model was better at learning the attributes for the different classes and handling the imbalanced dataset. The boosting model had similar findings to the Random Forest model with a 4.07% difference between the sensitivity and specificity. Although the boosting set has a very lower percentage for both sensitivity and specificity when compared to the Random Forest, they both do well with dealing with the unbalanced set and identifying the differences between dissatisfied/neutral passengers and satisfied passengers. In terms of accuracy, Random forest had the highest percentage at 96.5%. Airlines who want to determine predicted satisfactory outcomes for passengers accurately would be best suited to use the Random Forest model.

Business Applications

The dataset and our models have practical uses especially when applied in the aviation industry. From the analysis, we found that OnlineBoarding, Inflight Wifi Service, Check In Service, On-Board Service and Leg Room Service are important in increasing passenger satisfaction. If Airlines companies will invest on improving those specific variables rather than improving all variables. It can lead to reduced cost, increases in efficiency, and an improvement in passenger satisfaction. The attributes that we can control to increase satisfaction level for airlines are in this specific order descending from the level of influence determined from the Logistic Regression model are:

1. Online Boarding - make it easier to purchase fares online and see flight information.
2. In Flight WiFi Service - provide WiFi services to all passengers, we can include a premium service bundle that provides faster internet speeds and food/drinks for said customer.
3. Check In Service & On Board Service - provide excellent service to make them feel more welcome/comfortable.
4. Leg Room Service - improve leg room in our seating.
5. Cleanliness - make sure the environment stays clean at all times.
6. Baggage Handling - provide better service when it comes to baggage handling.
7. In Flight Service - improve our in flight service.

An improvement on a combination of features above can lead to greater satisfaction levels from our passengers. By improving passenger satisfaction, passengers are less likely to complain, and the airline's reputation will increase. More notoriety increases the chances of introducing potential new customers to the business and retaining current customers.

Future Work

In the future, we would like to extend our research and consider other factors. We could consider if having specific airlines have an effect on the overall satisfaction of the airline. Different airlines tend to have different ticket prices despite having the same flight destinations. As a result, the price differences could be as a result of the overall "experience" each airline provides.

Another factor we could consider is to split the passengers into specific groups: solos or 2+. We would like to discover if traveling alone or with others has an effect on the overall satisfaction of an airline. In addition to categorizing the passengers, we could also further examine more specific reasons for traveling: vacation, business or emergency. For example, a passenger who catches a flight due to an unforeseen circumstance elsewhere could cause them to enter the flight feeling uneasy. However, if that airline offers convenient features such as online boarding and inflight wifi, they could possibly help distract and soothe the passenger's distress.

A final question we could answer is whether or not one's income level has an effect on how much one is satisfied with their flight experience. We could possibly split the income level into lower class, middle class and high class groups. Higher income allows passengers to gain access to premium flight features, which could positively affect their satisfaction.

There are many factors to discover whether or not a passenger is satisfied with their airline, so future work in this topic could help us determine which factors are most significant.

Appendix

Data Visualization Code:

```
# correlation matrix plot

# extract numeric columns only
numeric_cols <- sapply(df, is.numeric)
df_numeric <- df[, numeric_cols]

corr_matrix <- cor(df_numeric)
col <- colorRampPalette(c("#BB4444", "#EE9988", "#FFFFFF", "#77AADD", "#4477AA"))
corrplot(corr_matrix, method = 'color', tl.cex = 0.5, title = "Correlation Matrix",
         mar=c(0,0,1,0))
```

```
# bar chart on satisfaction
plot(df['Satisfaction'], main = 'Airline Satisfaction',
     ylab = 'Count', xlab = 'Satisfaction Level', col = rainbow(2))
```

```
# bar chart on Gender
plot(df['Gender'], main = 'Airline Genders',
     ylab = 'Count', xlab = 'Gender', col = c('pink','lightblue'))
```

```{r echo=TRUE}
# bar chart on Airline Class
plot(df['Class'], main = 'Airline Class',
     ylab = 'Count', xlab = 'Airline Class',  col = c("#E69F00", "#56B4E9", "#009E73"))
```

```
# bar chart on Airline Type of Travel
plot(df['TypeTravel'], main = 'Airline Travel Type',
     ylab = 'Count', xlab = 'Travel Type', col = c("yellow", "#009E73"))
```

```{r echo=TRUE}
# bar chart on Airline Customer Type
plot(df['CustomerType'], main = 'Airline Customer Type',
     ylab = 'Count', xlab = 'Customer Type', col = c("#56B4E9", "#009E73"))
```

```
# Age histogram separated by Gender
ggplot(df, aes(x = Age, color = Gender, fill = Gender)) +
  geom_histogram(bins = 30) +
  labs(title = "Age Histogram Separated by Gender", x = "Age", y = "Count")
```

```
# Flight Distance Density Plot
ggplot(df, aes(x = FlightDistance)) +
  geom_histogram(aes(y = after_stat(density)), bins = 40, fill = "lightblue") +
  geom_density(alpha = 0.1, fill = "lightgreen") +
  labs(title="Flight Distance Density Plot",x="Flight Distance")
```

Metric Functions:

```
specificity = function(cm){
  list = confusion(cm)
  return(list[[4]] / (list[[4]]+list[[2]]))
}
```

```
confusion =  function(cm){
  TP = cm[1,1]
  FP = cm[1,2]
  FN = cm[2,1]
  TN = cm[2,2]
  list = list(TP,FP,FN,TN)
  return(list)
}

accuracy = function(cm){
  list = confusion(cm)
  return(((list[[1]]+list[[4]])/(list[[1]]+list[[2]]+list[[3]]+list[[4]])))
}

sensitivity = function(cm){
  list = confusion(cm)
  sense = list[[1]]/ (list[[1]]+list[[3]])
  return(sense)
}
```

Data Splitting:

```r
# split train and test sets to a 80/20 split
n = nrow(df)
prop = .80
set.seed(1)
train_id = sample(1:n, size = round(n*prop), replace = FALSE)
test_id = (1:n)[-which(1:n %in% train_id)]
train_set = df[train_id, ]
test_set = df[test_id, ]
```

Model 1 - Logistic Regression:

```r
# Fitting a Logistic Regression Model with all predictors
log.fit = glm(Satisfaction ~., data = train_set, family = 'binomial')
summary(log.fit)
```

```r
# Fitting a Logistic Regression Model with all significant predictors

log.fit2 = glm(Satisfaction ~ Gender + CustomerType + Age + TypeTravel +
                 Class + InflightWifiService + DepartArrive +
                 EaseOnlineBook + GateLocation + FoodDrink +
                 OnlineBoarding + SeatComfort + InFlightEntertainment +
                 OnBoardService + LegRoomService + BaggageHandling +
                 CheckinService + InflightService + Cleanliness +
                 DepartureDelay + ArrivalDelay,
                  data = train_set, family = 'binomial')

summary(log.fit2)
```
```{r}
# log confusion matrix with significant predictors
y_pred_log = predict(log.fit2, newdata = test_set, type = 'response')
y_pred_log = ifelse(y_pred_log > 0.5, 'satisfied', 'neutral/dissatisfied')
log_cm = table(predict_status = y_pred_log, true_status = test_set$Satisfaction)
print(log_cm)
```

```r
cat('\nThe Accuracy is:', accuracy(log_cm))
cat('\nThe Sensitivity is:', sensitivity(log_cm))
cat('\nThe Specificity is:', specificity(log_cm))
```

```
# Logistic Regression ROC Curve
y_pred_log = predict(log.fit2, newdata = test_set, type = 'response')
pred_log = prediction(y_pred_log, test_set$Satisfaction)
perf = performance(pred_log, "tpr", "fpr")
plot(perf, main = "Logistic Regression ROC Curve")
abline(0, 1, lty=3)
```
```{r}
# Logistic Regression AUC Value
log_auc = as.numeric(performance(pred_log, "auc")@y.values)
log_auc
```

Model 2 - Random Forest:

```
# Fitting a Random Forest with all predictors
p = ncol(train_set) - 1

set.seed(123)
forest.fit = randomForest(Satisfaction ~., data = train_set, mtry = round(sqrt(p)), importance = TRUE)
forest.fit
```

```
# Random Forest Confusion Matrix
yhat.forest = predict(forest.fit, test_set, type = "class")
forest_cm = table(predict_status = yhat.forest, true_status = test_set$Satisfaction)
forest_cm
cat('\nThe Accuracy is:', accuracy(forest_cm))
cat('\nThe Sensitivity is:', sensitivity(forest_cm))
cat('\nThe Specificity is:', specificity(forest_cm))
```

```
# Random Forest Feature Importance
varImpPlot(forest.fit, main = "Variable Importance (Random Forest)", type = 2)
```

Model 3 - Boosting Algorithm:

```
# Encoding Our Satisfaction column
df = df |> mutate(satisfaction_numeric = ifelse(Satisfaction == "satisfied",1,0)) |>
dplyr::select(-Satisfaction)
```

```
# splitting our data into train/test with 80/20
n = nrow(df)
prop = .8
set.seed(123)
train_id = sample(1:n, size = round(n*prop), replace = FALSE)
test_id = (1:n)[-which(1:n %in% train_id)]

train_set = df[train_id, ]
test_set = df[test_id, ]
```

```r
# parameters we check
grid = expand.grid(
    n.trees_vec = c(200),
    shrinkage_vec = c(0.25, 0.30, 0.32),
    interaction.depth_vec = c(3),
    miss_classification_rate = NA,
    time = NA
)

head(grid, 10)
```

```r
# grid search for best parameters for our Boosting model
set.seed(1)
for(i in 1:nrow(grid)){
  time = system.time({
    boost_fit = gbm(satisfaction_numeric ~ ., train_set,
                    n.trees = grid$n.trees_vec[i],
                    shrinkage = grid$shrinkage_vec[i],
                    interaction.depth = grid$interaction.depth_vec[i],
                    distribution = "bernoulli", cv.folds = 5)
}
)
  grid$miss_classification_rate[i] =
    boost_fit$cv.error[which.min(boost_fit$cv.error)]
  grid$time[i] = time[["elapsed"]]
}
```

```r
# arranging the miss_classification_rate in ascending order
grid |> arrange(miss_classification_rate)
```

```r
# Our best Boosting model with lowest miss classification rate
boost.fit.best = gbm(satisfaction_numeric ~ ., train_set, n.trees = 200,
                    shrinkage = 0.32, interaction.depth = 3,
                    distribution = "bernoulli")
boost.fit.best
```

```r
# Feature Importance
summary.gbm(boost.fit.best)
```

```
# Boosting Confusion Matrix
phat.test.boost.best = predict(boost.fit.best, test_set, type = "response")
yhat.test.boost.best = ifelse(phat.test.boost.best > 0.5, 1, 0)
boost_cm = table(pred = yhat.test.boost.best, true = test_set$satisfaction_numeric)
boost_cm
cat('\nThe Accuracy is:', accuracy(boost_cm))
cat('\nThe Sensitivity is:', sensitivity(boost_cm))
cat('\nThe Specificity is:', specificity(boost_cm))
```

```
Models = c('Logistic Regression', 'Random Forest', 'Boosting')
Accuracy = c(accuracy(log_cm), accuracy(forest_cm), accuracy(boost_cm))
Sensitivity = c(sensitivity(log_cm), sensitivity(forest_cm), sensitivity(boost_cm))
Specificity = c(specificity(log_cm), specificity(forest_cm), specificity(boost_cm))

Results = data.frame(Models,Accuracy,Sensitivity, Specificity)
Results
```