Anber Bhuiyan

Professor Saavedra

BUSQOM 1760: Team 5, Section 11 am

22 April 2022

Data Mining Final Report: Invistico Airlines

Managerial Problem

This dataset we intend to use comes from a real but unnamed company under the pseudonym, Invistico Airlines. On behalf of Invistico Airlines, we would like to delve into the dataset consisting of information about Invistico's passengers and their flying experiences. Our managerial concern was how we could predict if a customer would be satisfied or not. We wanted to explore this question further since it could help Invistico Airlines preserve and even increase their customer retention. Customer loyalty is especially important to an airlines company because of the potential to bring in billions in revenue. Delta Airlines recorded \$5.6 billion in loyalty and related revenues in 2019, Delta Airlines recorded \$9.1 billion and United Airlines \$5.3 billion, these numbers lead us to believe that understanding this satisfaction variable is very important. Of the factors the customers were surveyed on, a managerial concern was which specific variables happened to be of most importance for predicting if a customer was satisfied or not. We wanted to focus on the factors since it could give Invistico insight to identify problem areas or other valuable areas where they could allocate their money in order to improve. Some limitations we may encounter are the lack of variation in the levels of overall satisfaction.

Discussion of the Dataset

The data contains demographics about passengers who have either flown with them in the past or are first time customers. It also includes some variables that indicate they surveyed the customers after their flight and asked for their opinions and satisfaction levels. The following variables are included: Satisfaction, Gender, Customer Type, Age, Type of Travel, Class, Flight Distance, Seat Comfort, Departure/Arrival Time, Convenience, Food and Drink, Gate Location, Inflight Wifi Service, Inflight Entertainment, Online Support, Ease of Online Booking, On-Board Service, Leg Room Service, Baggage Handling, Checkin Service, Cleanliness, Online Boarding, Departure Delay in Minutes, and Arrival Delay in Minutes. These variables were a mix of categorical and numerical with 129,880 rows in all. This set was found on the Website Kaggle.com.

Initial Data Setup

The first thing we did was create a new data set called datOK that omitted any rows that had a missing value. Only 393 rows were taken out which is small compared to the 129,880 rows we started with. The next thing we had to do with the data was change the column "Satisfaction" to be a binary variable instead of a categorical one. This was easy to do because the data already had only two levels: "unsatisfied" and "satisfied", you can see the exact code we used to do this in appendix exhibit 1. We decided to check that there was a good split between 1s ("satisfied") and 0s ("unsatisfied") in the set and saw that 55% had 1 and 45% had 0 which we deemed good enough to continue. Then we split the data into training and testing sets based on a split ratio of 70%. Lastly, we did a quick test to make sure we had split the 0s and 1s proportionally between the sets; the numbers matched very well so we continued on to predictive modeling.

Logistic Regression

One of the methods we decided to use was logistic regression to answer the question, "How can we predict if a customer will be satisfied or not?" For this approach we started by looking at all the variable's correlations to identify any multicollinearity. To do this, the noncontinuous variables were first removed and a new dataset was created to test for the correlation amongst the continuous independent variables. The variable Food and Drink was excluded because of its correlation of 0.7160 with Seat Comfort, as the rule of thumb is to remove any over (+/-) 0.7. Additionally, the variables Class, Type of

Travel, Gender, and Customer Type were converted from character variables to factor variables in the training set.

For this method we really wanted to find what the best combination of variables would be, but the usual regsubsets function that we learned in class is for regression models. We attempted to use the package bestglm to accomplish this but ultimately the computer did not have enough power and we had to try a more manual technique instead. See appendix exhibit 2 for more information on this attempt.

The next approach to determine the logistic regression model we wanted to use was trial and error of various variable combinations in an attempt to find the model with the lowest Akaike Information Criterion (AIC) value. A model with the following variables resulted in the lowest AIC value we were able to find, at a value of 78,150: Inflight Entertainment, Seat Comfort, Ease of Online Booking, Online Support, Customer Type, On-Board Service, and Class.

To evaluate the model, we determined a threshold value of 0.25 to be sufficient. Then, using our logistic regression model, we calculated the predicted probabilities for the observations in the test set. Using the predicted probabilities, we added a column to the test set with a value of 1 for the observations where the predicted probability is greater than our threshold probability, and a value of 0 for the observations where it is less. A confusion matrix was then created displaying the actual values of the observations and the predicted values of the observations. See appendix exhibit 3 for a figure illustrating the confusion matrix. Using this confusion matrix, the results of this method included an accuracy of 74.67%, a sensitivity (true positive rate) of 93.07%, and a specificity (true negative rate) of 54.24%. The accuracy of this method is not too bad, the sensitivity value is great indicating the model does well at predicting satisfaction, and the specificity value is a little lower than we would prefer indicating that the model does not do as well at predicting unsatisfaction. In addition, a Receiving Operator Characteristic (ROC) curve was built standardizing the predicted probabilities and actual values of the observations of the test set, then using the performance function in R with the standardized values, true positive rate, and false positive rate needed for plotting the ROC curve. A diagonal line was added to represent the benchmark as a result of random classification. See appendix exhibit 4 for a figure of the ROC curve.

Using the ROC curve, the area under the curve (AUC) was computed to be 0.8888. In evaluating the AUC value, it is preferred to be greater than 0.5, representing the benchmark using random classification, and close to 1, representing the perfect model. With a value of 0.8888, we would conclude that as a good result. In addition, when viewing the ROC curve, it is ideal for the curve to be as close to the top left point of the graph, a sensitivity value of 1 and a specificity value of 1. In our graph, the curve can be seen as relatively far from the diagonal, and thus can be concluded as a good result.

Classification Trees

We decided to run a classification tree because this would be able to answer the question, "How do we predict which customers will be satisfied?" It would also give the airline employees a visual representation of which variables are important and how the end result is actually determined. We experimented with different values for both the number of observations we want in each leaf (minbucket) and the complexity parameter which adds a penalty for more splits. We started with 30 for minbucket and a cp of 0 then created a cp plot. Usually, as a rule of thumb, you want to choose the first cp that is under the dotted line which represents one standard deviation above the minimum cross validation error. When we did this we saw that none of the cp options went below the dotted line even after we tried different minbuckets so instead we used our judgment and chose .0019. This cp has an error of only around .02 while still keeping a good number of splits in the tree. Once we created the code with our optimized minbucket and cp, we got the tree shown in appendix exhibit 5. Next we used this model to predict values for the test set and created a confusion matrix, shown in appendix exhibit 6. With this matrix we found an accuracy of 91.2%, a sensitivity of 92.6%, and a specificity of 89.5%. These are great results and we are very happy with them; they show that this model not only has high predictive power, but it predicts both 1s (sensitivity) and 0s (specificity) well. Looking at the tree's output we can see that Inflight Entertainment is the first determining variable, Seat Comfort and Class (economy, first, etc) also pop up fairly often in the tree. Intuitively it makes sense that these 3 variables are important. When one thinks of the things that are most important to them on a plane, these are some of the first to come to mind. It also makes intuitive sense that we can see when a variable is higher on the scale of 1 to 5, it is more likely to end in a 1 or “satisfied” result. Overall this method could be very useful to help the airline understand the factors involved in determining if someone is satisfied or not.

Random Forest

To answer the managerial question, “Which variables are the most important for predicting if a customer is satisfied?”, we decided to run a random forest. To start, we first determined the number of explanatory variables in our dataset, 22, then using the rule of thumb for classification, took the approximate value of the square root of 22 to determine the initial mtry value of 5. An initial random forest was run using the training set, 100 total trees, a minimum of 400 observations in each leaf, and 5 random independent variables to be used in determining each split.To ensure that the best value of mtry was used, we used the tuneRF function to find the mtry that results in the lowest out-of-bag (OOB) error. The tuneRF was run using the following characteristics: the dependent variable was separated from the independent variables, the initial value of mtry was used as the starting point for the mtry value, for each run of the function the mtry will be multiplied and divided by a value of 2, 100 total trees, a minimum of 400 observations in each leaf, and the OOB error must be improved by at least 0.01 to be considered. As a result, we found that an mtry value of 10 results in the lowest OOB error, see appendix exhibit 7 for the resulting graph. A final random forest was then run using the same characteristics as the initial model, with the exception that the mtry value is now 10.

In evaluating our final model, we generated a variable importance plot, as shown in appendix exhibit 8. In viewing this plot, it can be seen that Inflight Entertainment is the most important variable to consider when predicting customer satisfaction, with Seat Comfort, Ease of Online Booking, and Online Support following, respectively. In addition, a confusion matrix was created using the predicted probabilities for the observations in the test set and the actual values of the observations. See appendix exhibit 9 for a figure illustrating the confusion matrix. Using this confusion matrix, the results of this method include accuracy of 93.53%, a sensitivity of 94.23%, and a specificity of 92.67%. In evaluating these results, the accuracy, sensitivity, and specificity values are all high percentages, which is a great result because we are accurately predicting a majority of the observations in the test set.

Conclusion - Main Findings, Implications, & Recommendations

If Invistico wanted to prioritize interpretability, we recommend using the classification tree. The classification tree's accuracy is 91.19%, and its specificity is 89.52%. Classification trees are easier to interpret overall, which will help the airline officials implement changes regardless if they know the methods behind building a classification tree..

If Invistico wanted to prioritize predictability, we would recommend using the random forest. Our random forest model had the highest accuracy (93.53%) and highest specificity (92.67%). We chose to examine specificity because we concluded that it would be worse for the airline to predict false positives, meaning that they are predicting customers are satisfied when in reality they are not. If the model predicts more customers to be satisfied when they are not, the airline will not have the data to support making changes for their unsatisfied customers.

Our variable importance plot indicates that Inflight Entertainment is the most important variable to consider when predicting customer satisfaction. Its average decrease in Gini Index is over 12,000, which is over double of the next leading variable. We recommend that Invistico invest in personal televisions (PTVs) for every seat on its aircraft. For those planes that already have PTVs installed, we further recommend that they invest in purchasing the rights to a good mix of classic and current movies, shows, and music.

The next two most important variables, also found from the output of our variable importance plot, are Seat Comfort and Ease of Online Booking, respectively. Seat Comfort has an average decrease in Gini Index of just under 6,000, and Ease of Online Booking has an average decrease in Gini Index of just under 4,000. To address the importance of seat comfort, we recommend that Invistico increase the frequency of their seat inspections to check for wear and tear from high passenger traffic. This can include torn or frayed seat cushions, malfunctioning or broken armrests, and a fractional range of the reclining function. Additionally, since e-commerce continues to grow in popularity, we recommend that Invistico conduct an additional survey about online booking to identify areas of improvement. Airline rivalry is high, switching costs may be low for some customer segments, and the variable importance plot highlights the value of simplifying the online booking process.

Implementing these recommendations will allow Invistico to increase customer retention because more of their customers may be satisfied, and these customers will continue flying with Invistico due to their positive experiences. Investing in inflight entertainment, seat inspections for seat comfort, and a simplified user experience on their website may aid Invistico in increasing their customer satisfaction rate. By increasing customer satisfaction, which in return may increase customer retention, will boost Invistico’s revenues and provide them with a sustainable consumer base.

Appendix

Exhibit 1: Initial Data Setup Code - Binary

```
dat = read.csv("Invistico_Airline.csv")
datOK = na.omit(dat)


datOK$satisfaction<-ifelse(datOK$satisfaction=="satisfied",1,0)
# this changes the categorical variable satisfaction to a binary
# 1 if satisfied, 0 if not
str(datOK)


datOK$satisfaction = as.factor(datOK$satisfaction)
str(datOK)
```

Exhibit 2: Logistic Regression, Bestglm Attempt

The first thing we did for this technique was reorder the columns using the dplyr package so satisfaction was in the last position which is needed for this to work. We also needed all of the categorical variables to be factors so we wrote some code to do that. First we ran it and got the error “p = 22. must be <= 15 for GLM." so we removed 7 variables that had very high but not quite 0.7 correlation with another variable. We ran this new, smaller, data set through the bestglm function again and let it run for over 12 hours with no result. This is when we decided to move to the technique we used in the report.

Exhibit 3: Logistic Regression, Confusion Matrix

	Predicted Values

Observed Values		**0**	**1**
	0	9,216	8,365
	1	1,474	19,791

Exhibit 4: Logistic Regression, ROC Curve

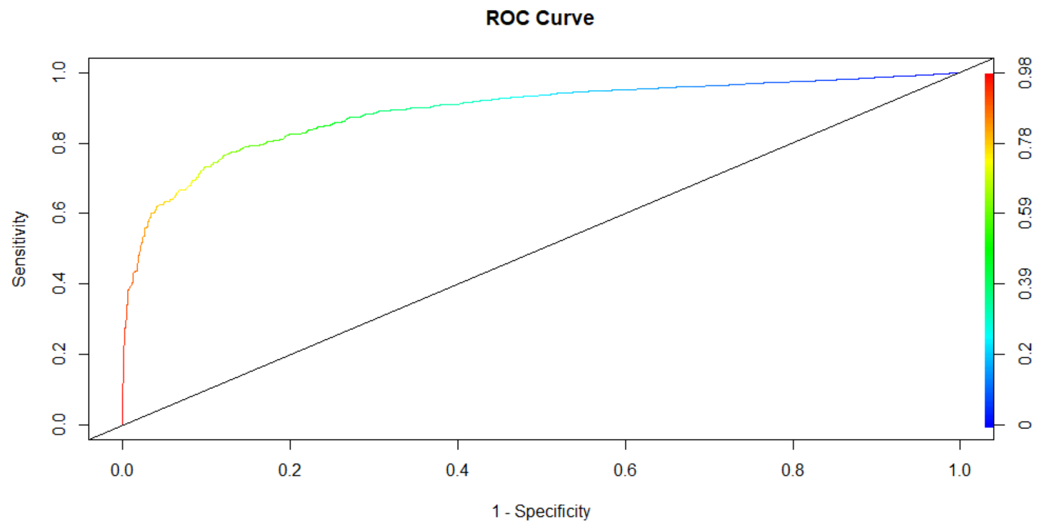

Exhibit 5: Classification Tree

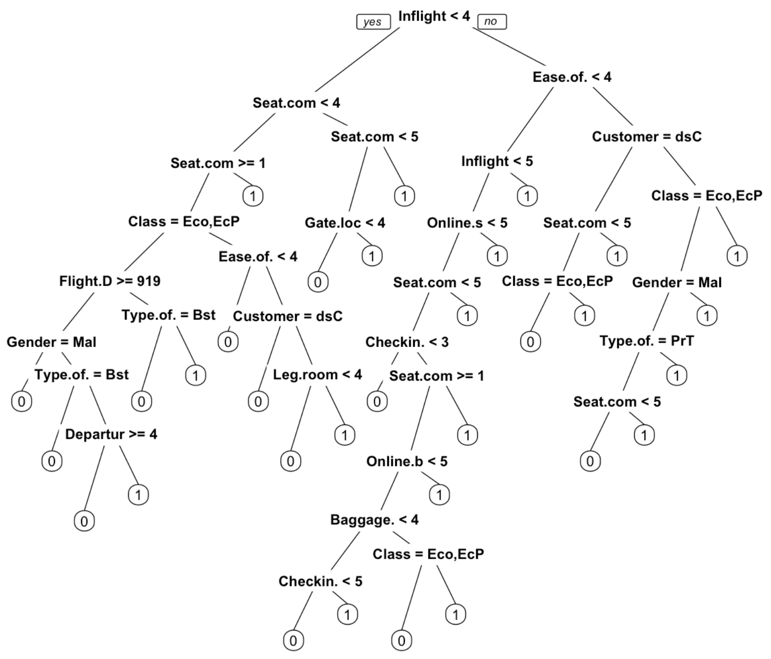

Exhibit 6: Classification Tree, Confusion Matrix

	Predicted Values		
Observed Values		**0**	**1**
	0	15,739	1,842
	1	1,579	19,686

Exhibit 7: Random Forest, OOB Error

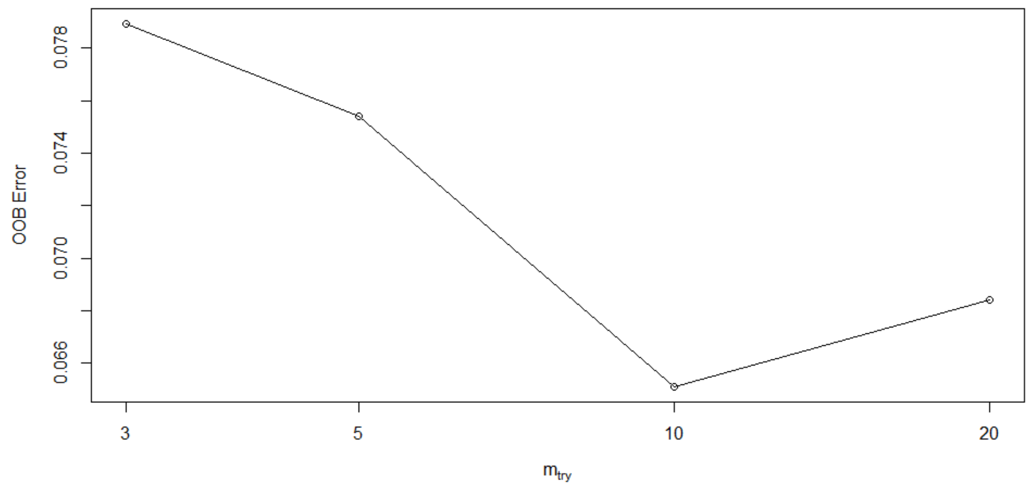

Exhibit 8: Random Forest, Variable Importance Plot

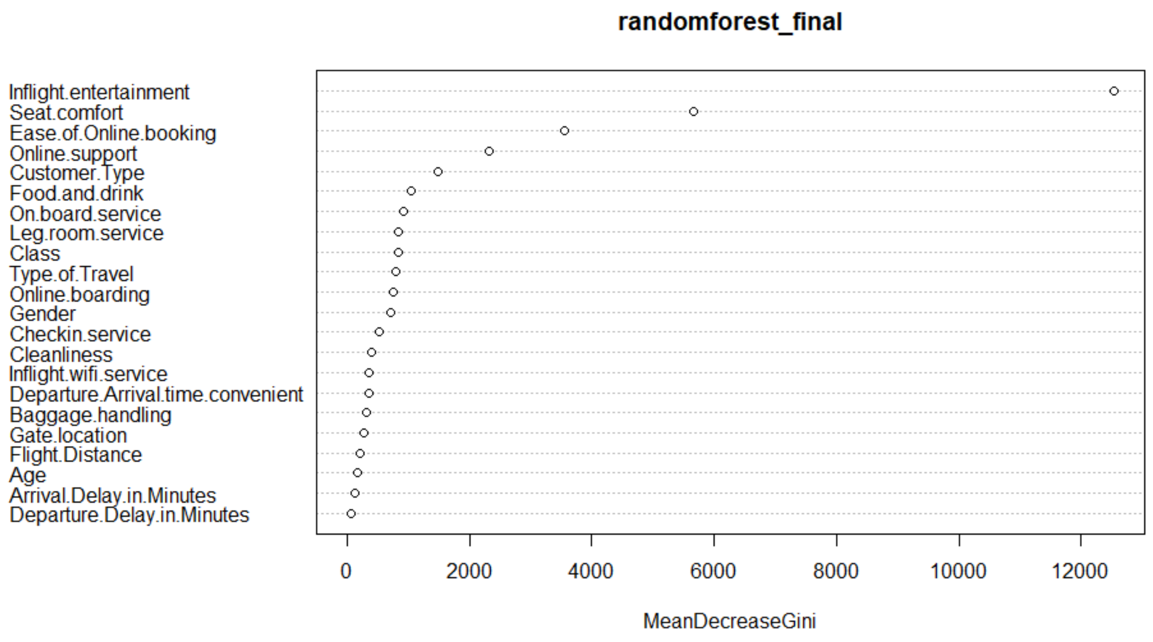

Exhibit 9: Random Forest, Confusion Matrix

	Predicted Values		
Observed Values		**0**	**1**
	0	16,293	1,288
	1	1,227	20,038

Exhibit 10: R Code

Initial Data Setup

```
dat = read.csv("Invistico_Airline.csv")


library(caTools)

library(rpart)

library(rpart.plot)

library(dplyr)

library(randomForest)

library(ROCR)


datOK = na.omit(dat)


datOK$satisfaction<-ifelse(datOK$satisfaction=="satisfied",1,0)

# this changes the categorical variable satisfaction to a binary

# 1 if satisfied, 0 if not
```

```
str(datOK)


datOK$satisfaction = as.factor(datOK$satisfaction)
str(datOK)


summary(datOK)
table(datOK$satisfaction)/nrow(datOK)
# tells us what % of the data is a 1 or a 0
# here we can see that about 45% is 0 and 55% is 1


set.seed(12, sample.kind = "Rejection")
spl = sample.split(datOK$satisfaction, SplitRatio = 0.7)
train.dat = datOK[spl==TRUE,]
test.dat = datOK[spl==FALSE,]


nrow(train.dat)/nrow(datOK)
table(train.dat$satisfaction)/nrow(train.dat)
table(test.dat$satisfaction)/nrow(test.dat)
# this checks that the data is proportional between the training set and the test set
```

Classification Tree

```
mb = 30
```

```
set.seed(12, sample.kind = "Rejection")
tree0 = rpart(satisfaction ~., data = train.dat, method = "class", minbucket = mb, cp = 0)
  # start with cp as 0 so can do cross validation to find best value
plotcp(tree0)
  # this is not showing very promising results, just go with .0019, its around 0.2 error
  # tried it with mb = 10, 20 and 30 and all were about the same, went with 30


set.seed(12, sample.kind = “Rejection”)
tree1 = rpart(satisfaction ~., data = train.dat, method = "class", minbucket = mb, cp = .0019)
prp(tree1)


pred = predict(tree1, newdata = test.dat, type = "class")
ConfMat = table(test.dat$satisfaction, pred)
ConfMat


#       0     1
#  0 15739  1842
#  1  1579 19686


# Accuracy:
(15739+19686)/nrow(test.dat)
# 0.9119343
# Sensitivity:
19686/(1579+19686)
# 0.9257465

# Specificity:
15739/(15739+1842)
# 0.8952278
```

Random Forest

```
ncol(datOK) - 1
sqrt(22)
# the square root of 22 is approximately a value of 5, which will be used as the initial mtry value

set.seed(12, sample.kind = "Rejection")
randomforest = randomForest(satisfaction~., data = train.dat, ntree = 100, nodesize = 400, mtry = 5)
varImpPlot(randomforest)

excl = 1

x = train.dat[,-excl]
y = train.dat$satisfaction
```

```
set.seed(12, sample.kind = "Rejection")
tuneRF(x, y, mtryStart = 5, stepFactor = 2, ntreeTry = 100, nodesize = 400, improve = 0.01)
#An mtry value of 10 results in the lowest OOB error

set.seed(12, sample.kind = "Rejection")
randomforest_final = randomForest(satisfaction~., data = train.dat, ntree = 100, nodesize = 400,
mtry = 10)

varImpPlot(randomforest_final)

pred.rf = predict(randomforest_final, newdata=test.dat)
ConfMat.rf = table(test.dat$satisfaction, pred.rf)
ConfMat.rf
 #      0     1
 # 0 16293  1288
 # 1  1227 20038

# Accuracy:
(16293+20038)/nrow(test.dat)
# 0.9352572

# Sensitivity:
20038/(1227+20038)
```

```
# 0.9422996


# Specificity:
16293/(16293+1288)
# 0.9267391
```

Logistic Regression

```
str(datOK)
excl2 = c(1,2,3,5,6)
datnum = datOK[,-excl2]
cor(datnum)
#Food.and.drink removed because correlation is greater than 0.7.


train.dat$Class = as.factor(train.dat$Class)
train.dat$Type.of.Travel = as.factor(train.dat$Type.of.Travel)
train.dat$Gender = as.factor(train.dat$Gender)
train.dat$Customer.Type = as.factor(train.dat$Customer.Type)
# changes Class,Type.of.Travel, Gender, and Customer.Type to factors
logreg = glm(satisfaction ~ Inflight.entertainment + Seat.comfort +
        Ease.of.Online.booking + Online.support + Customer.Type +
        On.board.service + Class,
        data = train.dat, family="binomial")
summary(logreg)
```

```
#AIC value of 78150


threshold = 0.25

test.dat$predProbability = predict(logreg, newdata = test.dat, type = "response")

test.dat$predSatisfaction = ifelse(test.dat$predProbability >= threshold, 1, 0)


table(test.dat$satisfaction, test.dat$predSatisfaction)

#      0     1

#0  9216  8365

#1  1474 19791


#Accuracy:

(19791 + 9216)/nrow(test.dat)

# 0.7467178


#Sensitivity:

19791 / (19791 + 1474)

# 0.9306842


#Specificity:

9216 / (9216 + 8365)

# 0.5242023
```

```
roc.curve = prediction(test.dat$predProbability, test.dat$satisfaction)

performance = performance(roc.curve, "tpr", "fpr")


plot(performance, main = "ROC Curve", xlab = "1 - Specificity", ylab = "Sensitivity", colorize =
TRUE)

abline(0,1)

auc = performance(roc.curve, "auc")

as.numeric(auc@y.values)

# AUC value of 0.8888263
```

Logistic Regression - Bestglm Attempt

```
excl2 = c(1,2,3,5,6)

        # removes categorical variables and dependent variable

datnum = datOK[,-excl2]

cor(datnum)


logreg = glm(satisfaction ~.-Food.and.drink, data = train.dat, family="binomial")


dat_bglm = datnum %>% relocate(satisfaction, .after = Arrival.Delay.in.Minutes)

  # puts satisfaction column at the end

dat_bglm$Class = as.factor(dat_bglm$Class)

dat_bglm$Type.of.Travel = as.factor(dat_bglm$Type.of.Travel)

  # changes Class and Type.of.Travel to factors
```

```
install.packages("bestglm")

library(bestglm)


best.log <- bestglm(dat_bglm, IC = "BIC", family = binomial, method = "forward")

# "ERROR  in bestglm(dat_bglm, IC = "BIC", family = binomial, method = "forward") : p = 22.

must be <= 15 for GLM."

# so we have to get rid of 7 variables

  # Arrival.Delay.in.Minutes, Baggage.handling, Online.boarding, Online.support, Gate.location,

On.board.service, Age


excl3 = c(22, 17, 20, 13, 10, 15, 3)

dat_bglm2 = dat_bglm[, -excl3]

best.log <- bestglm(dat_bglm2, IC = "BIC", family = binomial, method = "forward")

        # this ran for over 12 hours and did not come up with a result so we had to stop

plot(best.log)
```