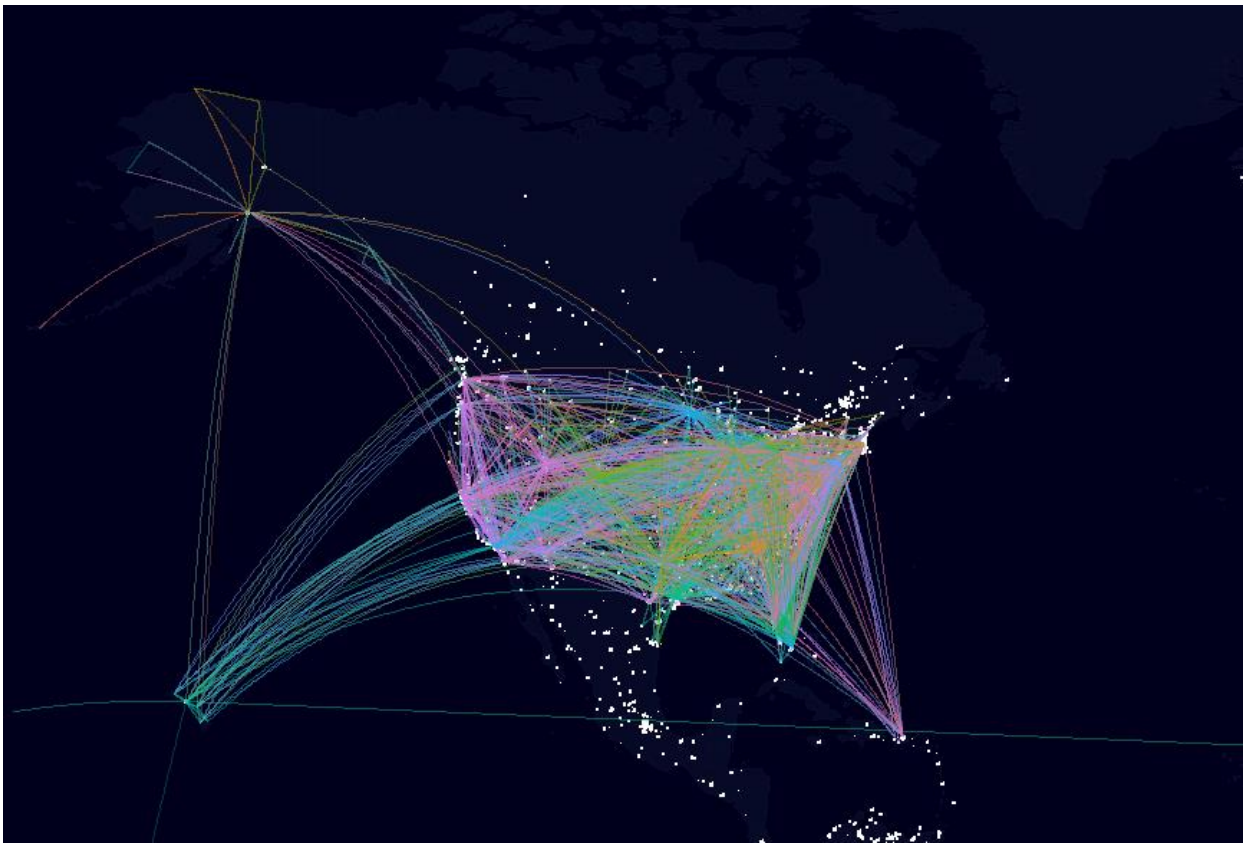
Syracuse University

School of Information Studies

Fall 2020

Flight Satisfaction Survey

IST687: Introduction to Data Science



(Plot of all flight patterns in the Flight Survey dataset. See appendix D)

Instructor: Gary Krudys

Presenter: Alec Schneider afschnei@syr.edu

Table of Contents

Introduction

Background

Airlines have been pioneers in the analytics community regarding pricing and route optimization, however they have long struggled with understanding customer satisfaction. It is possible they have struggled with this due to company bias, and lack of customer data from other airlines. To tackle this problem an industry funded customer survey gathered ~130,000 responses each with 25 attributes, to create 3.25 million data points that can be used to assist their understanding. I was tasked with analyzing this data thoroughly enough to provide actionable insights for the businesses to implement in their company strategies.

After cleaning and exploring the survey data, creating informative visualizations to assist in understanding the data, and applying machine learning algorithms to it, airlines are now closer to fully understanding what their customers value most, thus improving customer satisfaction. This will be a win for the airlines, as increased satisfaction is believed to increase demand of air travel, and the consumers who choose to travel on their aircrafts.

Business Questions

1. Which airlines and origin cities have the highest rates of departure delays and cancellations?
2. Do vacation destinations have a higher customer satisfaction score?
3. Which demographics are most price sensitive?
4. Which airlines have the highest satisfaction scores?
5. Which factors affect the customer satisfaction score most?

Data Cleaning & Feature Engineering

The analysis began with first understanding the tidiness of the dataset and understanding what logic should be used to clean any naming conventions and missing values, before engineering any additional features. All rows and variables were used for the cleaning and feature engineering stages to ensure that the analysis would be non-biased and be strictly derived from the data itself (See appendix I for full list of data and definitions). After all cleaning logic and feature engineering logic was decided upon, each step was coded as function that could be imported and reused, to save storage space by not having to create an additional file of the cleaned dataset. These functions were then combined in one function so the raw data could be transformed with one line of code inside an R script (see appendix B).

Data Cleaning

Any column names that contained a space were substituted with an underscore to prevent any future difficulties in column extraction. In the raw data there were four columns that had missing data, Satisfaction, Departure_Delay_in_Minutes, Arrival_Delay_in_Minutes, and Flight_time_in_minutes. A function, *fillNAs*, was created to apply the following logic to clean up the missing values (See appendix C).

1. Satisfaction column
    a. These observations with missing data were not associated with any other attributes
    b. It was decided to impute the missing values with the mean satisfaction score, excluding the missing values
2. 337 rows of data were removed from the data set as they were survey responses where the flight was not cancelled, but there was no arrival delay, flight time, or departure delay information. The decision was made to remove these rows since it as these data points could be reliably imputed with a mean value
3. Departure_Delay_in_Minutes column
    a. Since all remaining missing values in this column were associated with a cancelled flight, the assumption applied was there was no delay before cancellation. A value of zero was used to replace the missing value
4. Arrival_Delay_in_Minutes column
    a. Since all remaining missing values in this column were associated with a cancelled flight, the assumption applied was there was no delay before cancellation. A value of zero was used to replace the missing value
5. Flight_time_in_minutes column
    a. Since all remaining missing values in this column were associated with a cancelled flight, the assumption applied was there was no delay before cancellation. A value of zero was used to replace the missing value

Feature Engineering

In order to extract additional information from the existing variables, new variables were created to categorize discrete variables and one hot encode category variables so they can be used in a machine learning algorithm.

Categorization:

Four discrete variables were transformed into category variables for exploratory data analysis and possible one hot encoding for modeling purposes. The *category_processing* function was created to handle this transformation (See appendix E).

1. No._of_other_Loyalty_Cards
    a. Transformed into the cat_loyalty_cards variable with values of None (0), Low (0, 3], Medium (3, 6], and High (6, 12] to create an ordinal hierarchy of the number of loyalty cards a customer holds
2. %_of_Flight_with_other_Airlines
    a. Transformed into the cat_%_of_FLight_with_other variable with values of Low (0, 1/3], Medium (1/3, 2/3], and High (2/3, 1] to create an ordinal hierarchy for the percent of flights that customer takes with other airlines
3. No_of_Flights_p.a.
    a. Transformed into the cat_No_of_Flights variable with values of Low (0, 1/3], Medium (1/3, 2/3], and High (2/3, 1] to create an ordinal hierarchy of number of flights a customer has been on
4. Age
    a. Transformed into the cat_Age variable with values of (-Inf, 20], (20, 30], (30, 40], (40, 50], (50, 60], (60, 70], and (70, Inf] to break up the age buckets to be of 10 years, excluding the small population of customers 20 and younger

One Hot Encoding:

One hot encoding is a form of data transformation that encodes categorical features as numeric features that take binary (0 or 1) values. One hot encoding allows these categorical columns to be used in machine learning models since their values become numeric. Using this methodology, I created n-1 columns for each categorical column below, where n is the number of unique categories. N-1 column methodology was used to remove any potential multi-collinearity in the dataset. The *dummy_processing* function was created to one hot encode the following variables (See appendix F).

1. Airline_Status
2. Type_of_Travel
3. Class
4. cat_loyalty_cards
5. cat_No_of_Flights
6. cat_Age

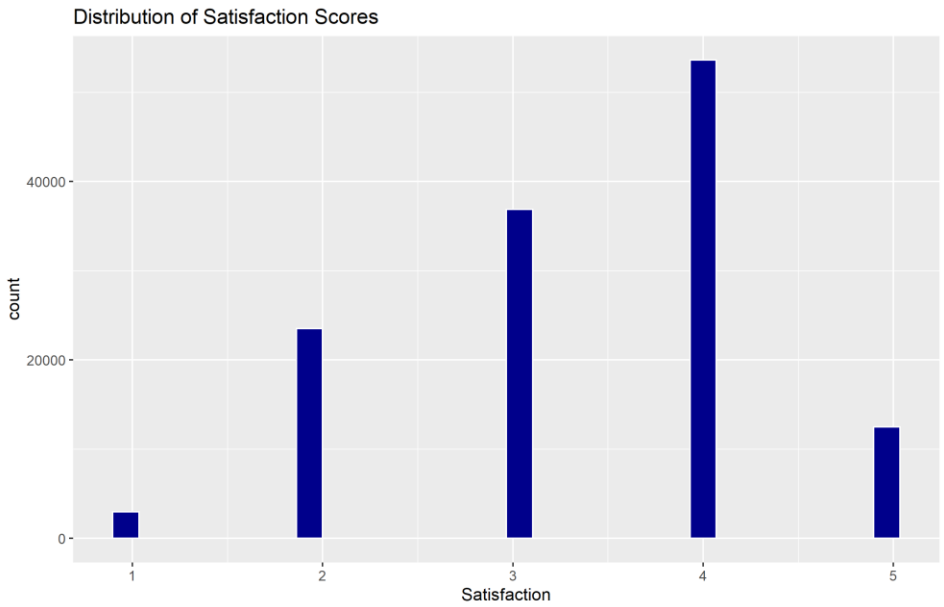
Additional Feature Engineering:

Since delay times that are of a higher proportion of the flight time may lead to decreased satisfaction, two additional variables were created to display these proportions.

1. DepDelayRatio
    a. Departure_Delay_in_Minutes / Flight_time_in_minutes
2. ArrDelayRatio
    a. Arrival_Delay_in_Minutes / Flight_time_in_minutes

Exploratory Data Analysis

First, I wanted to explore our target variable, Satisfaction. The distribution of the satisfaction scores is left skewed, with most customers giving an above average score of four, with a mean score of 3.38:

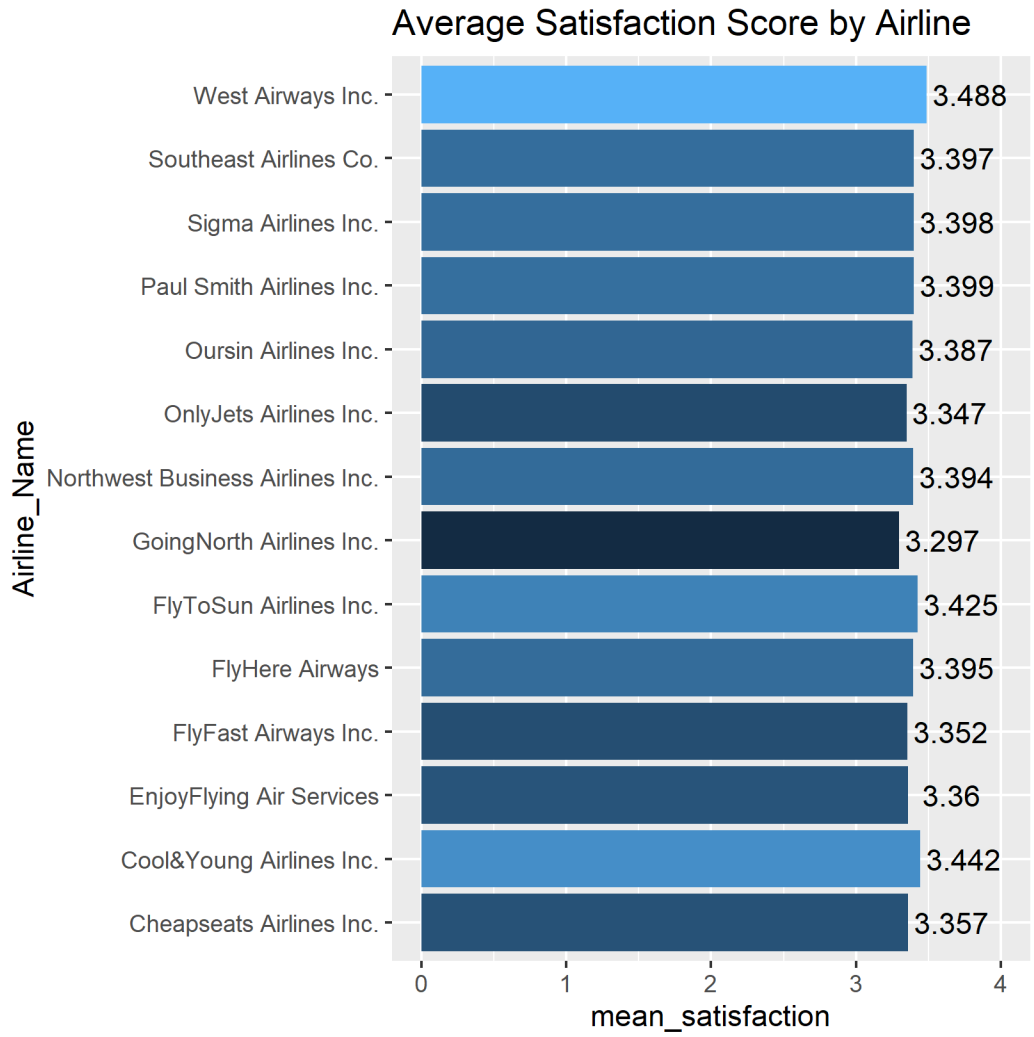


Second, it was important to analyze the individual airlines. When analyzing the mean satisfaction score across the whole population, large differences between the airlines would provoke further investigation to see which factors the higher performing airlines scored better in. The data was grouped by on the Airline_Name column, and the following summary statistics were calculated:

- Average satisfaction score
- Average flight delay
- Average flight time
- Number of flight cancellations
- Departure delay frequency
- Cancellation frequency

Airline_Name	count	avg_satisfaction	avg_delay	avg_flight_time	num_cancellations	delay_freq	cancellation_freq
<chr>	<int>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>
Cheapseats Airlines Inc.	25985	3.36	18.0	101.	316	0.607	0.0122
Cool&Young Airlines Inc.	1281	3.44	12.5	184.	1	0.364	0.000781
EnjoyFlying Air Services	8906	3.36	15.3	74.5	319	0.376	0.0358
FlyFast Airways Inc.	15356	3.35	21.2	74.8	661	0.432	0.0430
FlyHere Airways	2474	3.40	11.9	99.8	51	0.460	0.0206
FlyToSun Airlines Inc.	3392	3.42	6.02	160.	20	0.230	0.00590
GoingNorth Airlines Inc.	1568	3.30	16.3	116.	6	0.492	0.00383
Northwest Business Airlines Inc.	13790	3.39	12.7	73.6	248	0.335	0.0180
OnlyJets Airlines Inc.	5382	3.35	19.2	148.	123	0.421	0.0229
Oursin Airlines Inc.	10953	3.39	16.0	169.	151	0.524	0.0138
Paul Smith Airlines Inc.	12207	3.40	12.8	144.	156	0.388	0.0128
Sigma Airlines Inc.	17018	3.40	12.9	120.	217	0.387	0.0128
Southeast Airlines Co.	9555	3.40	8.69	124.	132	0.328	0.0138
West Airways Inc.	1685	3.49	2.60	76.5	0	0.207	0

All of the airlines had an average satisfaction score in a tight range, with no true outliers (See figure below). Looking further at the summary statistics, it is interesting to note that despite having the most amount of flights in the dataset, Cheapseats Airlines Inc. also had the highest frequency of delayed flights, with a margin of ~8% between them and the next closest airline Oursin Airlines Inc. The frequency of cancellations shows that FlyFast Airways Inc. scored the worst with a frequency of ~4%. It is very insightful to know that the airline with the best mean satisfaction score, West Airways Inc., had 0 cancellations on ~1700 flights, the lowest delay frequency at ~21%, and had the lowest average departure delay of only 2.6 minutes. Cool&Young Airlines Inc. had the second highest mean satisfaction score and scored very well in the three aforementioned categories relative to peers.



When doing a similar analysis on origin cities, there does not seem to be any patterns, but it is alarming that the ten highest departure delay rates are 57% and above:

```
# A tibble: 295 x 8
   Origin_City     count avg_satisfaction avg_delay avg_flight_time num_cancellations delay_freq cancellation_freq
   <chr>           <int>            <dbl>     <dbl>           <dbl>             <dbl>      <dbl>             <dbl>
 1 Adak Island, AK     3             2.67        21            154.                 0          1                 0
 2 Lincoln, NE        35             3           23.3           70.1                0          0.686             0
 3 Dallas, TX       1087             3.36        15.7           55.6               10          0.660             0.00920
 4 Dickinson, ND      22             3.36        41.1           72.7                0          0.636             0
 5 Denver, CO       4939             3.35        18.9          112.                61          0.595             0.0124
 6 Islip, NY         115             3.30        23.9          133.                 3          0.591             0.0261
 7 Chicago, IL      7615             3.33        22.0          107.               313          0.586             0.0411
 8 Baltimore, MD    1965             3.36        18.1          115.                44          0.582             0.0224
 9 Kodiak, AK          7             3.14        62.3           40                  0          0.571             0
10 Oakland, CA       910             3.37        16.0           85.5                7          0.569             0.00769
# ... with 285 more rows
```

It is also interesting to note that the cities with the ten highest cancellation rates all score pretty well, with each city have atleast a mean satisfaction score of three:

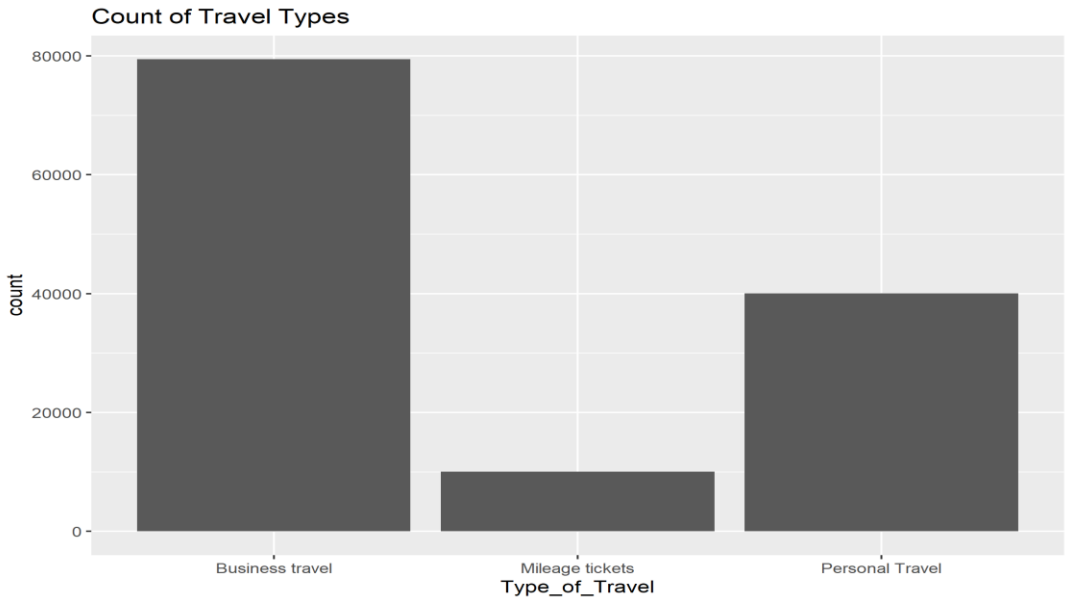
Origin_City	count	avg_satisfaction	avg_delay	avg_flight_time	num_cancellations	delay_freq	cancellation_freq
<chr>	<int>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>
1 Sun Valley/Hailey/Ketchum, ID	32	3.25	6.81	48.1	5	0.188	0.156
2 Muskegon, MI	14	3	6.42	31.2	2	0.214	0.143
3 Appleton, WI	53	3.45	16.7	59.9	7	0.321	0.132
4 Topeka, KS	23	3.13	32	69.6	3	0.217	0.130
5 Roanoke, VA	47	3.28	28.9	64.7	6	0.362	0.128
6 Branson, MO	24	3.12	11.5	67.1	3	0.292	0.125
7 Mammoth Lakes, CA	16	3.12	10.7	44.8	2	0.188	0.125
8 Rhinelander, WI	26	3.35	0.25	34.3	3	0.0769	0.115
9 Cody, WY	18	3.5	6.19	56.8	2	0.389	0.111
10 Marquette, MI	9	3.44	11.9	54.2	1	0.222	0.111

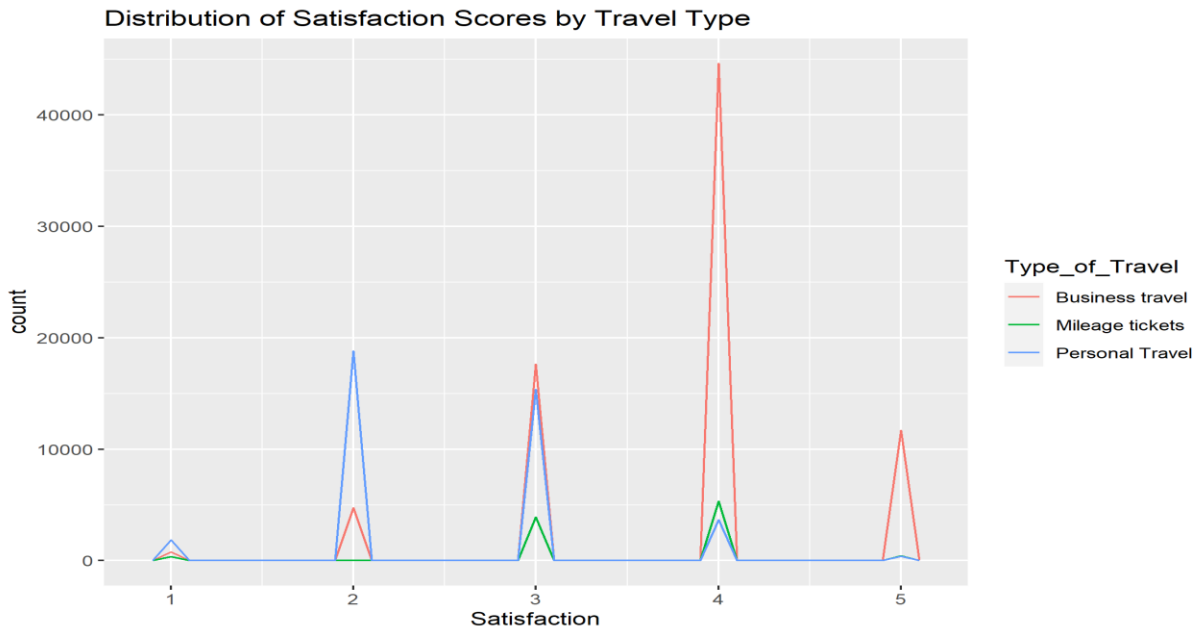
Third, to answer our question regarding vacation destinations, there needed to be a mapping of vacation destinations. To accomplish this, all unique destination cities were written to a csv file using R, manually tagged as a vacation destination based on the criteria of being a known vacation spot (West Palm Beach/Palm Beach, FL) or a high entertainment area (Nashville, TN). This file was then read back into R and merged with the cleaned dataset. Before calculating the mean scores, business travel types were filtered out so they did not skew the data. The data was then grouped on whether or not the destination was a vacation destination, and calculated the count of records, mean satisfaction score, and standard deviation score for each group. The mean satisfaction score and standard deviations were almost exact, which meant that customers are not more likely to be satisfied with their travel experience if they are going to a vacation destination. This goes against the initial hypothesis. The full list of vacation destinations can be found in appendix G.

```
> avg_score_vac_dest <- df_merged %>%
+   filter(Type_of_Travel != "Business travel") %>%
+   group_by(Vacation) %>%
+   summarize(avg_satisfaction = mean(Satisfaction),
+             count=n(),
+             std=sd(Satisfaction))
> avg_score_vac_dest
# A tibble: 2 x 4
  Vacation avg_satisfaction count   std
     <int>            <dbl> <int> <dbl>
1        0             2.75 22039 0.856
2        1             2.75 28088 0.859
```

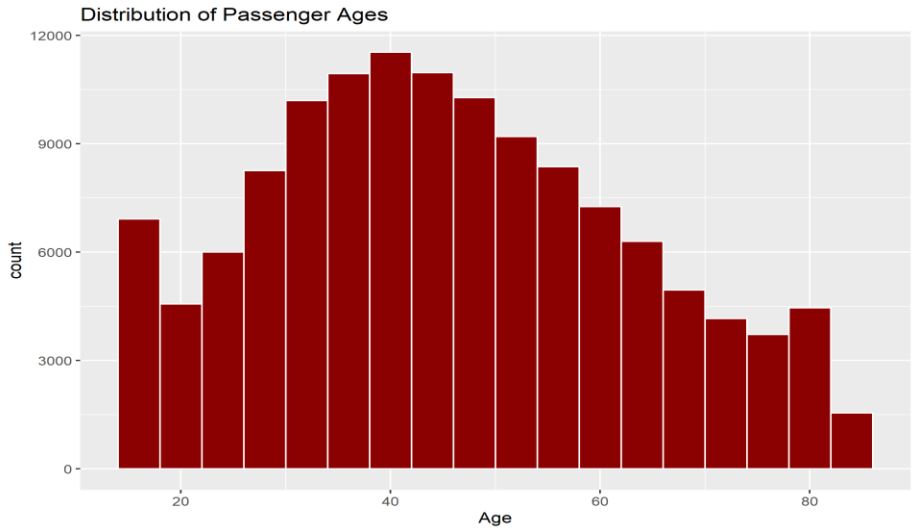
Business travel is a large part of the dataset (>50%), so while this type of travel should be removed from any vacation analysis, it should be included when understanding satisfaction scores. When looking at the distribution of scores by travel type, it is interesting to see that business travel has a heavy left skew towards higher satisfaction scores, whereas personal travel has a right skew towards lower satisfaction. Mileage tickets' distribution seems to mirror the overall distribution of scores, with a left skew as well.

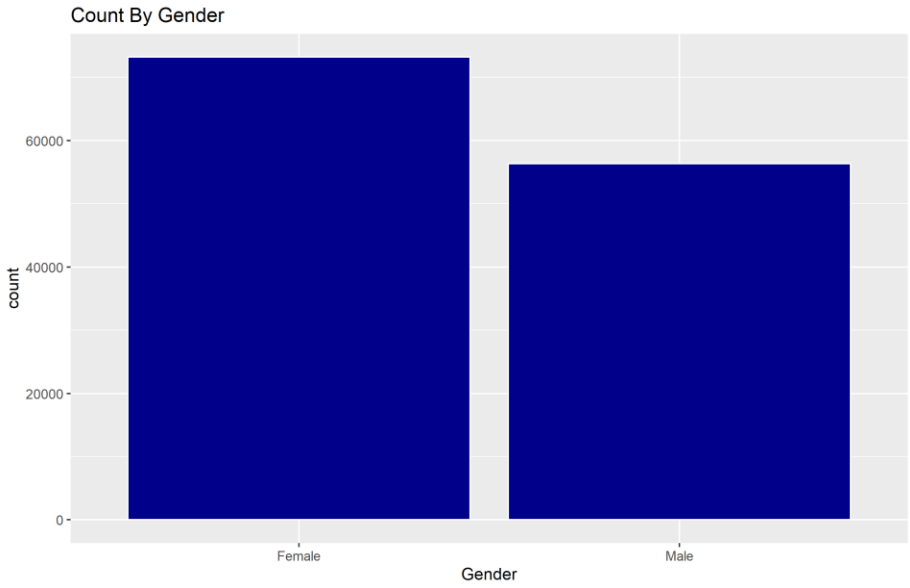
To improve satisfaction scores, it would be best to target business travelers, especially given how sparse they are due to the COVID-19 pandemic.



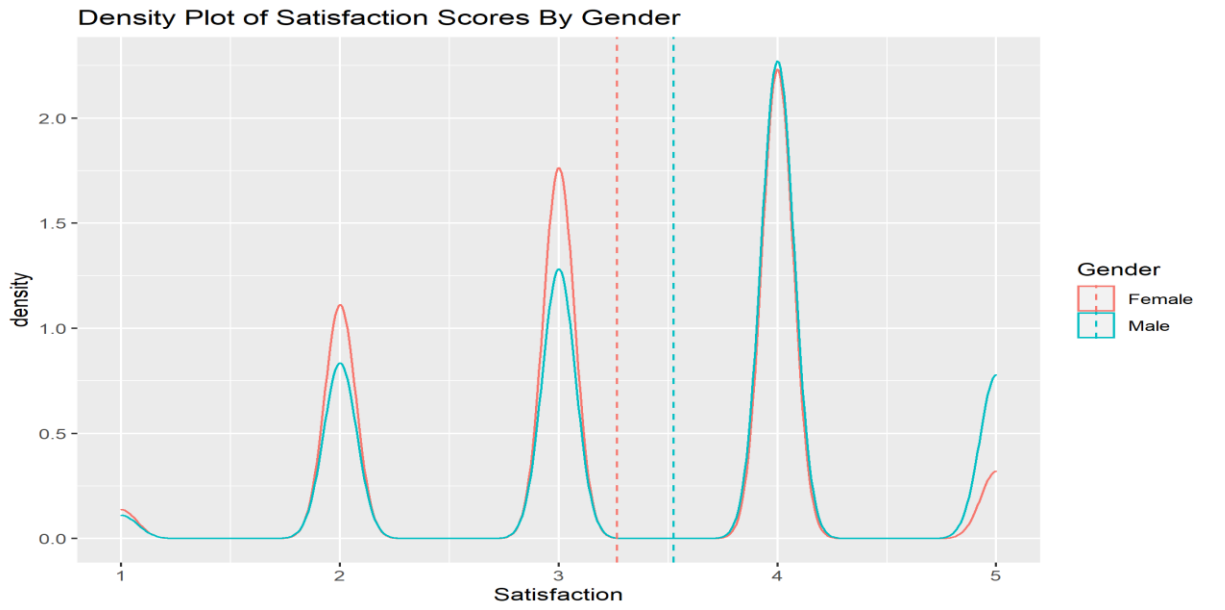


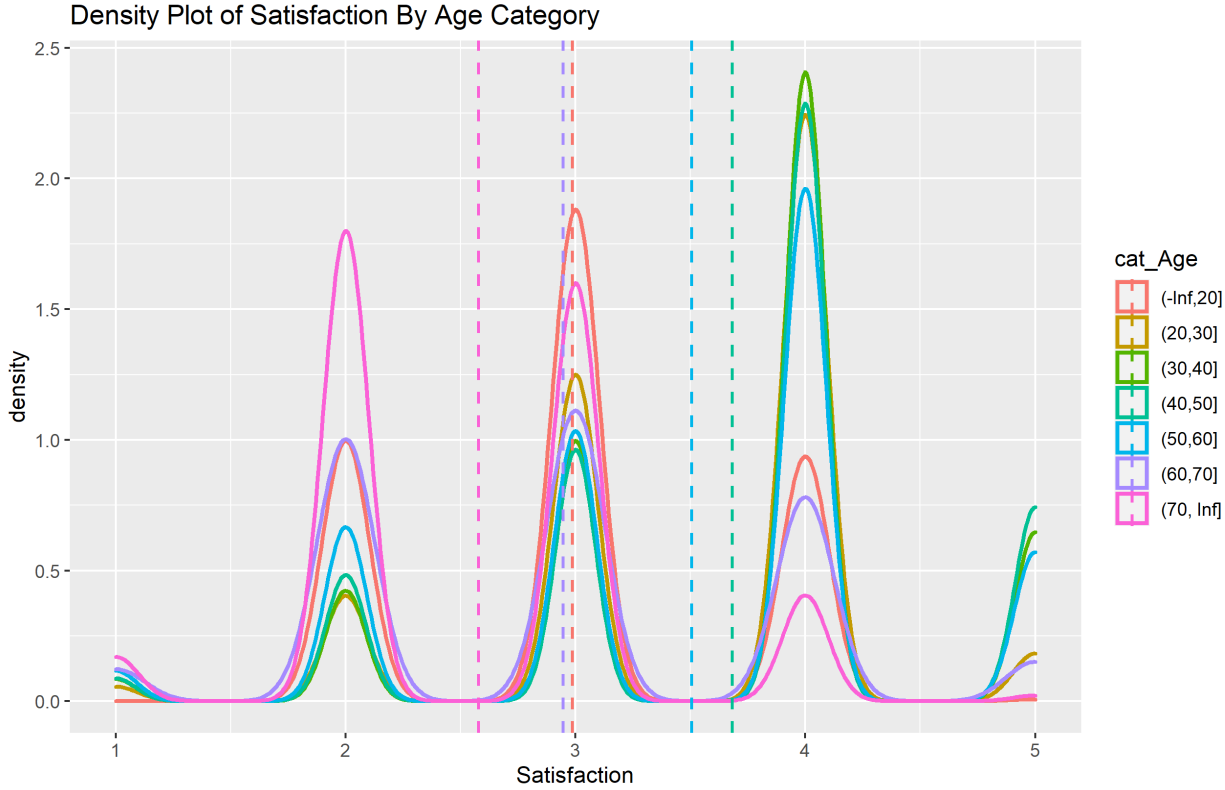
Fourth, to provide concrete recommendations to the industry, it was absolutely necessary to analyze the demographic data, which are represented in the Age and Gender columns. The distribution of ages is right skewed with spikes at both edges on the distribution but shows that most customers surveyed were around the age of forty years old. There are also ~20,000 more females surveyed in the data versus males.

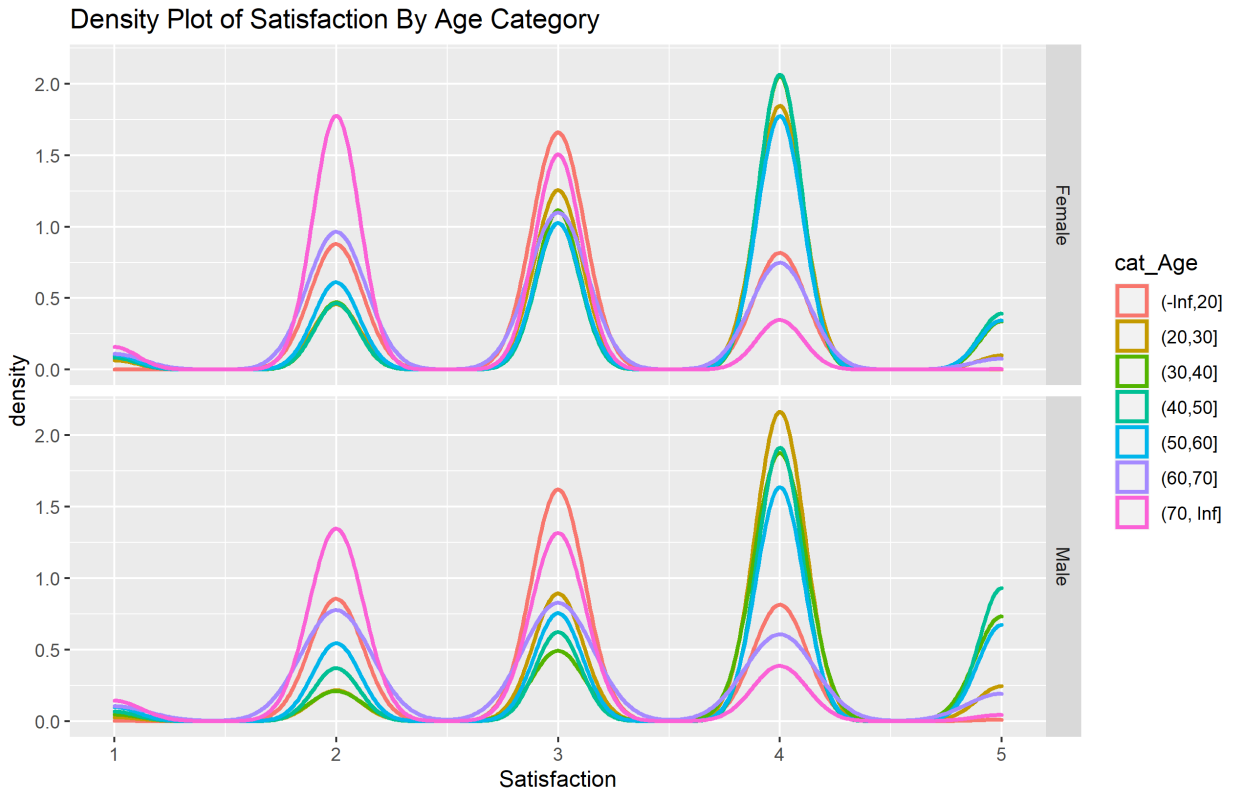




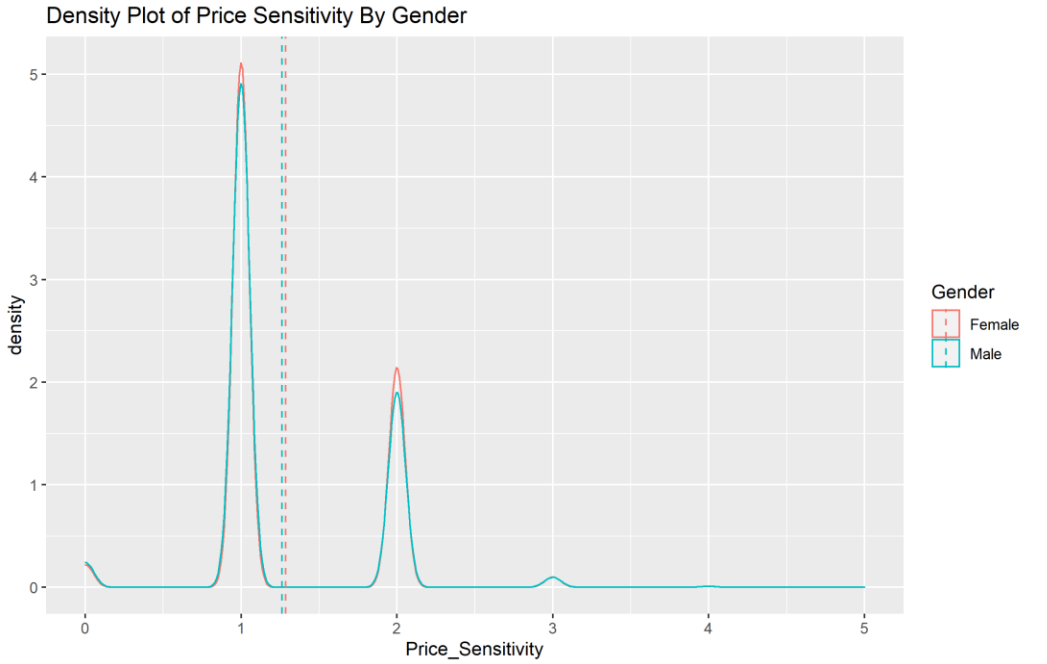
A density plot of the satisfactions scores by gender shows that males have a much higher likelihood to have a satisfaction score of 5 than females, which gives males a higher mean satisfaction score. The mean satisfaction scores for males and females are 3.53 and 3.27. The second plot below is also a density plot by age category and displays that the three of the most popular age buckets (30, 40], (40, 50], and (50, 60] also have high mean satisfaction scores, 3.68, 3.68, and 3.51, that are well above the average of 3.38. The oldest age bucket (70, Inf] has a way below average satisfaction score of 2.58, perhaps because they do not feel properly accommodated given the health situations this age group has compared to the other age groups.

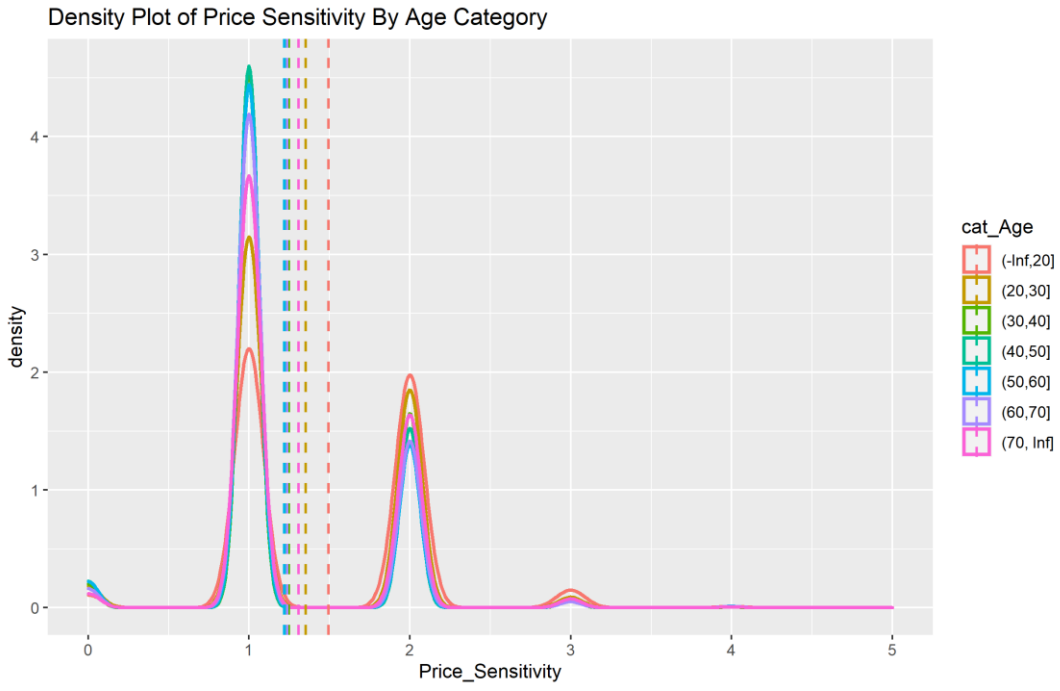

Density Plot of Satisfaction Scores By Gender
density
2.0
1.5
1.0
0.5
0.0
1
2
3
4
5
Satisfaction
Gender
Female
Male

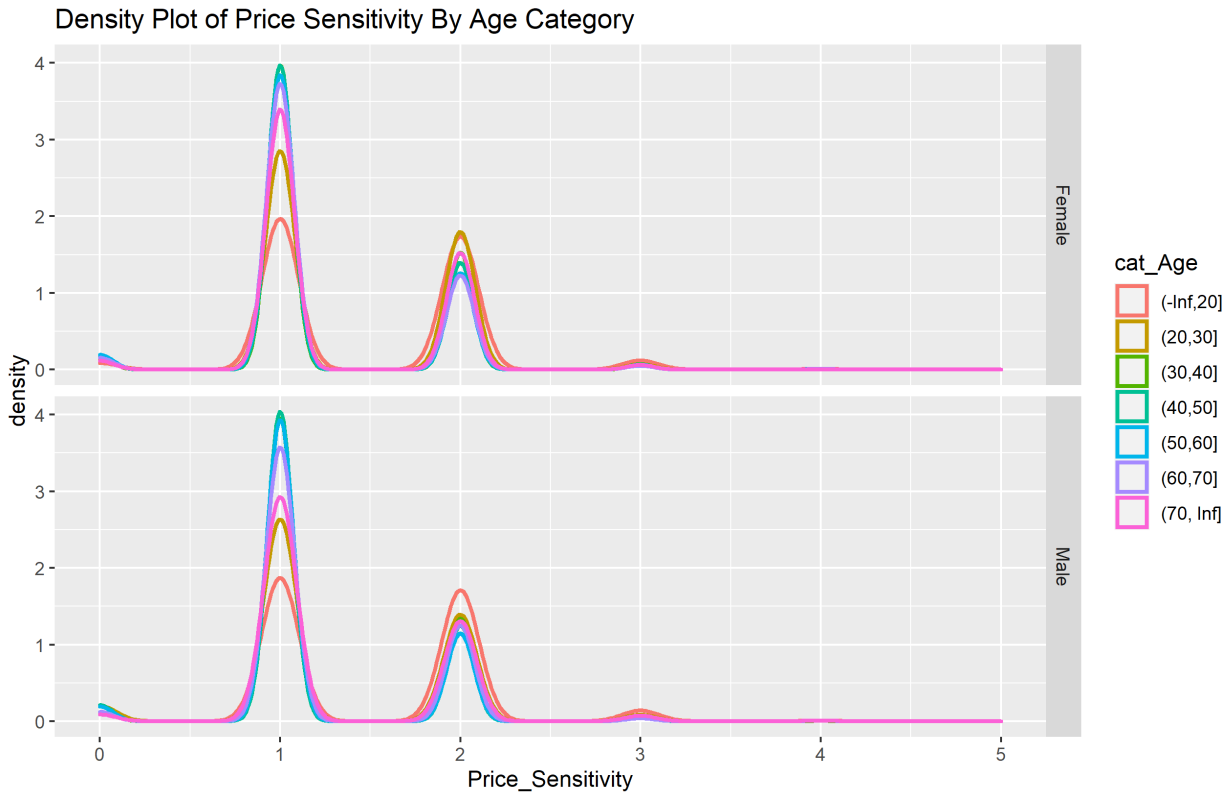

Density Plot of Satisfaction By Age Category
density
2.5
2.0
1.5
1.0
0.5
0.0
1
2
3
4
5
Satisfaction
cat_Age
(-Inf,20]
(20,30]
(30,40]
(40,50]
(50,60]
(60,70]
(70, Inf]



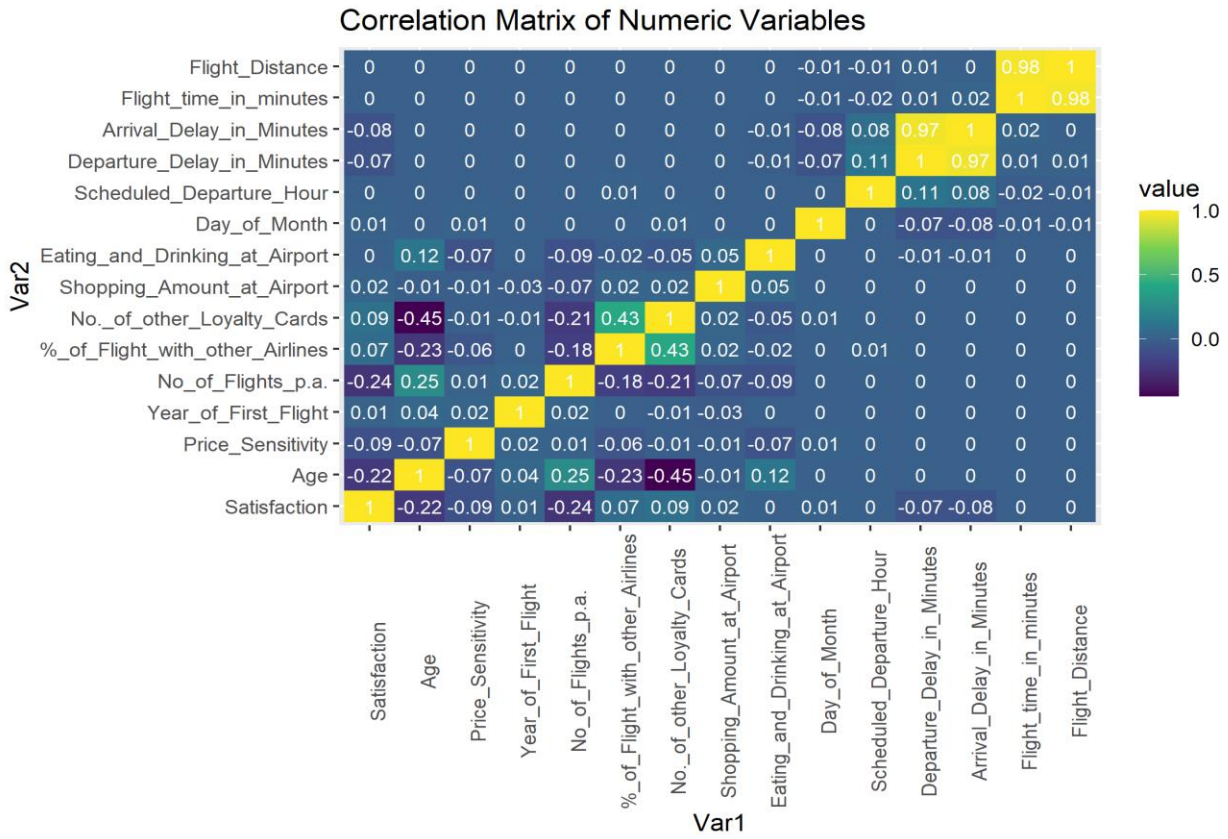
Another thing to look deeper at, since airlines are concerned about pricing correctly, is the price sensitivities of the demographics. Seeing how different demographics react to price increases can provide insights on how to price seat tickets. The first chart below displays a density plot of the price sensitivities by gender, and the mean price sensitives of 1.28 for females and 1.26 for males. This shows that overall, the price sensitivities are not statistically different between the genders. The second chart plots the density of price sensitivities and mean price sensitivities by the age groups that were described in the feature engineering section of this analysis. As one would assume, the two youngest age buckets are the most price sensitive, along with the oldest age bucket. The middle age buckets are the least price sensitive, which is a good sign for airlines as they are the most common flyers. Airlines can extract additional pricing value on this age demographic.


Density Plot of Price Sensitivity By Gender
density
Price_Sensitivity
Gender
Female
Male


Density Plot of Price Sensitivity By Age Category
density
Price_Sensitivity
cat_Age
(-Inf,20]
(20,30]
(30,40]
(40,50]
(50,60]
(60,70]
(70, Inf]


Density Plot of Price Sensitivity By Age Category
density
Price_Sensitivity
Female
Male
cat_Age
(-Inf,20]
(20,30]
(30,40]
(40,50]
(50,60]
(60,70]
(70, Inf]

Lastly, before beginning to build a statistical model to predict customer satisfaction, it is important to understand the correlations in the dataset. This ensures that multicollinearity can be avoided, and the highest correlated features with the satisfaction score can be used. The figure below is a correlation matrix of all the numeric variables in the data set. At first glance it is easy to see that Arrival_Delay_in_Mintues and Departure_Delay_in_Minutes are almost perfectly correlated, so one of these variables and the ratio variables created from them should always be excluded. There are no strong correlations with Satisfaction, but No_of_Flights_p.a. and Age have the strongest, with -0.24 and -0.22.



Modeling

Linear Regression

As a baseline, linear regression would be used to gauge if modeling results would be sufficient to continue to keep our target variable as a continuous variable, or if it should be transformed into a category variable to be used with classification algorithms. The distribution of the satisfaction scores displays the need for classification, but linear regression will put that to the test. Linear regression uses independent variables (the x variables), a coefficient variable for each independent variable, and an intercept to predict a continuous dependent variable (the y variable). These variables and coefficients are used to model a linear relationship in the data. Linear regression generally uses a least squares approach to attempt to fit the data.

Linear regression models in R can only take numeric variables as input, which is what makes our prior data transformation steps important. Before building the model, we will need to create a list of the numeric variables, which includes our one hot encoded variables:

- Gender
- Price_Sensitivity
- Shopping_Amount_at_Airport
- Day_of_Month
  Scheduled_Departure_Hour
- Flight_cancelled
- Flight_time_in_minutes
- Flight_Distance
- Airline_StatusGold
- Airline_StatusPlatinum
- Airline_StatusSilver
- Type_of_TravelMileage.tickets
  Type_of_TravelPersonal.Travel
- ClassEco
- ClassEco.Plus
- cat_loyalty_cards.Low
- cat_loyalty_cards.Medium
- cat_loyalty_cards.High
- cat_No_of_Flights.Medium
- cat_No_of_Flights.High
- cat_Age..20.30.
- cat_Age..30.40.
- cat_Age..40.50.
- cat_Age..50.60.
- cat_Age..60.70.
- cat_Age..70..Inf.
- X.cat_._of_FLight_with_other.
  Medium
- X.cat_._of_FLight_with_other.H
  igh Age

All of these variables will be passed to the *lm* function built into R to return a fitted model. To split the data between a training set used to fit the model, and a test set that the model can be evaluated on, the following code will be used to generate an 80/20 split:

```
# set the seed so we can compare our results on the same splits
set.seed(11)
# split the data into training and test data
split = sample.split(model_data$Satisfaction, SplitRatio=.8)
training = subset(model_data, split == TRUE)
test = subset(model_data, split == FALSE)
```

The training data was then passed to the regression model. Summary of the trained linear regression model:

```
Coefficients:
                                  Estimate Std. Error  t value Pr(>|t|)
(Intercept)                      3.512e+00  2.222e-02  158.013  < 2e-16 ***
GenderMale                       1.261e-01  4.739e-03   26.607  < 2e-16 ***
Price_Sensitivity               -2.442e-02  4.222e-03   -5.784 7.33e-09 ***
Shopping_Amount_at_Airport       1.488e-04  4.313e-05    3.451 0.000559 ***
Day_of_Month                     1.291e-03  2.620e-04    4.930 8.25e-07 ***
Scheduled_Departure_Hour        -9.374e-04  4.908e-04   -1.910 0.056133 .
Flight_cancelledYes             -1.684e-01  2.112e-02   -7.973 1.57e-15 ***
Flight_time_in_minutes          -1.492e-03  1.324e-04  -11.269  < 2e-16 ***
Flight_Distance                  1.771e-04  1.590e-05   11.142  < 2e-16 ***
Airline_StatusGold               4.179e-01  8.440e-03   49.514  < 2e-16 ***
Airline_StatusPlatinum           2.340e-01  1.314e-02   17.807  < 2e-16 ***
Airline_StatusSilver             5.963e-01  5.938e-03  100.433  < 2e-16 ***
Type_of_TravelMileage.tickets   -1.129e-01  8.801e-03  -12.825  < 2e-16 ***
Type_of_TravelPersonal.Travel   -9.967e-01  5.898e-03 -169.009  < 2e-16 ***
ClassEco                        -7.726e-02  8.384e-03   -9.215  < 2e-16 ***
ClassEco.Plus                   -5.339e-02  1.072e-02   -4.983 6.28e-07 ***
cat_loyalty_cards.Low            6.081e-03  6.160e-03    0.987 0.323550
cat_loyalty_cards.Medium        -5.460e-02  1.513e-02   -3.608 0.000308 ***
cat_loyalty_cards.High          -2.129e-01  8.454e-02   -2.518 0.011792 *
cat_No_of_Flights.Medium        -3.660e-02  5.618e-03   -6.515 7.29e-11 ***
cat_No_of_Flights.High          -8.031e-02  5.973e-03  -13.447  < 2e-16 ***
cat_Age..20.30.                  1.522e-01  1.302e-02   11.695  < 2e-16 ***
cat_Age..30.40.                  2.516e-01  1.768e-02   14.228  < 2e-16 ***
cat_Age..40.50.                  2.611e-01  2.406e-02   10.852  < 2e-16 ***
cat_Age..50.60.                  2.022e-01  3.124e-02    6.472 9.70e-11 ***
cat_Age..60.70.                  7.796e-03  3.855e-02    0.202 0.839747
cat_Age..70..Inf.               -3.578e-02  4.764e-02   -0.751 0.452633
X.cat_._of_FLight_with_other.Medium -8.104e-04  5.461e-03   -0.148 0.882024
X.cat_._of_FLight_with_other.High   -2.829e-02  6.387e-03   -4.430 9.43e-06 ***
Age                             -6.021e-04  7.762e-04   -0.776 0.437912
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 0.7301 on 103612 degrees of freedom
Multiple R-squared:  0.4275,    Adjusted R-squared:  0.4273
F-statistic:  2667 on 29 and 103612 DF,  p-value: < 2.2e-16
```

At first glance it is easy to see that there are a lot of statistically significant variables. These variables should be considered when building other models. The F-statistic of the equation is also statistically significant; however, the model seems to be a poor fit with an adjusted r-squared of only 42.7%. Now the model will be evaluated on test data, where root mean square error will be used to evaluate the prediction results. Root mean square error is defined as the standard deviation of the prediction errors and can be calculated as:

$$\mathrm{RMSD} = \sqrt{\frac{\sum_{i=1}^{N} (x_i - \hat{x}_i)^2}{N}}$$

RMSD = root-mean-square deviation

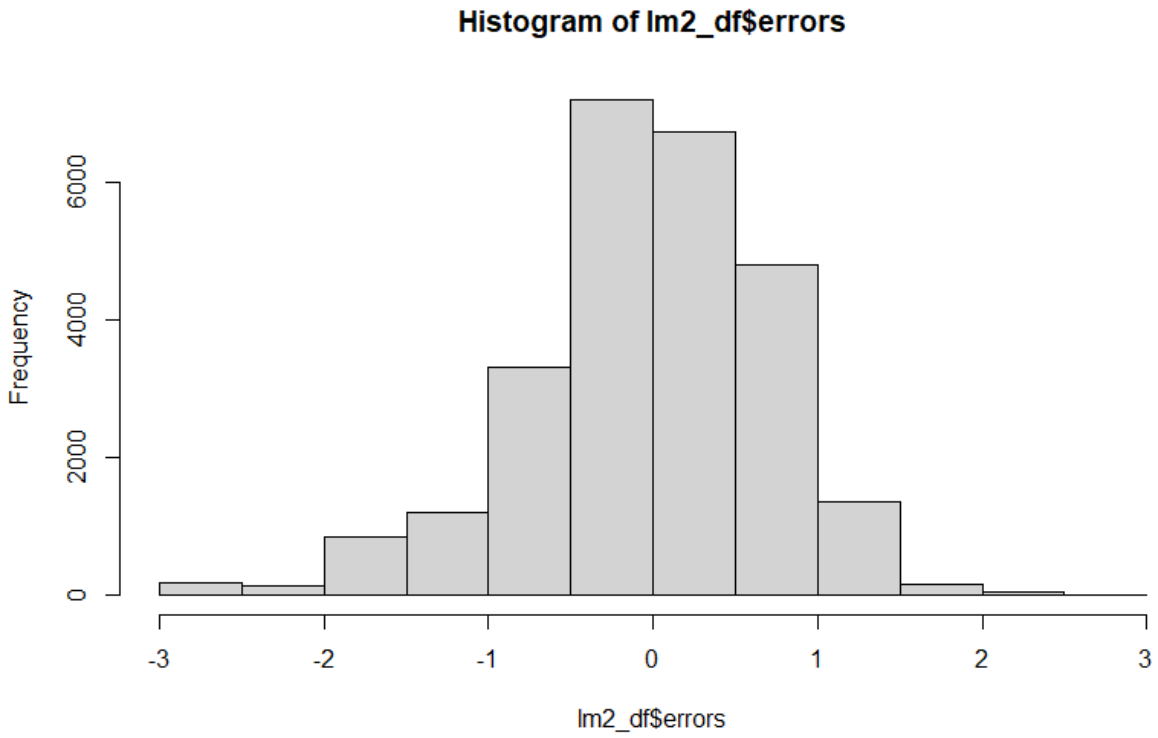
i = variable i

N = number of non-missing data points

x_i = actual observations time series

$\hat{x}_i$ = estimated time series

After predicting the Satisfaction score using the linear regression model and comparing to the actual Satisfaction score in the test set, the root mean square comes out to 0.731. This can be interpreted to mean that the linear regression model's errors have a standard deviation of 0.731 and given that our Satisfaction scores range from 1 to 5, this means about 68% of the errors are within a 15% difference of the actual result. This error rate is quite large, leading me to believe that a classification algorithm should be used to predict customer satisfaction scores.



Gradient Boosting Tree Classification

Gradient boosting is an algorithm that uses weak learner decision trees (trees that do not have many nodes, or decisions, in them) to improve learning sequentially, as trees are built one after another. This varies from random forests that build decision tress in parallel and can be much more complex. Due to its learning algorithm, gradient boosting trees tend to have high bias, and low variance, but can even reduce bias by aggregating the output from many different models. Gradient boosting trees can be very powerful and fast to run, which are a big advantage over machine learning algorithms like support vector machines. To implement the gradient boosting algorithm in R, the xgboost library will be used to create classification models and provide insights on which features of the dataset have a high importance, using the *xgb.plot.importance* function. At a high level, the importance of a feature is calculated based on how frequently the feature is used to make key decisions across all of the decision trees created.

Since the Satisfaction variable is still considered a numeric variable, and not a categorical variable that can be used in a classification algorithm, additional data processing will have to be done on top of the processing discussed in the Data Cleaning & Feature Engineering section of this analysis. First,

the Satisfaction values are mapped to the following integers (Note: it is a requirement for xgboost classification models to have n classes that are from 0 to n-1):

- 1 = 0
- 2 = 1
- 2.5 = 2
- 3 = 3
- 3.4 = 4
- 3.5 = 5
- 4 = 6
- 4.5 = 7
- 5 = 8

Second, since one hot encoding is not necessary for xgboost models and can take factor (or categorized) variables, the Type_of_Travel, Gender, and Flight_cancelled column will be factorized. Then, the data will be split using the same code shown above in the Linear Regression section. Before training the model on the training data, some of the parameters are going to have to be changed to properly predict on our data. Some other parameters can also be changed to assist in fine tuning the classification model. The following parameters will be used during each iteration of model tuning:

- Objective - function used to predict the final output
    - The multi:softmax function will be used for all models as it is used in multi-class classification to predict probabilities for each possible class value (in our case 0-8)
- Num_class – the number of possible unique outcomes in the target variable
    - There are 9 possible values of Satisfaction (as we mapped above) and this will be used in each model
- Eval_metric – metric which xgboost will use to optimize the model
    - Merror metric will be used as it is a multi-class classification metric calculated as num. of wrong predictions / num. of predictions
- Max_depth – maximum depth of the decision trees used in the model
    - This parameter will be changed to tune the model
- Eta – the learning rate used to help prevent overfitting
    - This parameter will be changed to tune the model
- Nrounds – the maximum number of iterations or trees to make
    - This parameter will be changed to tune the model

The *xgboost* function also requires that the data be passed in as a matrix, not as data.frame, so the *as.matrix* function will be used to transform the data in the *xgboost* function. The code to run the function will look like:

```
classifier4 <- xgboost(params=list(objective="multi:softmax", "num_class"=9, "eval_metric"="merror", "max_depth"=3, "eta"=0.1) ,

                       data=as.matrix(training[, usable_x_cols]),

                       label=training$Satisfaction, nrounds=1000,

                       verbose=1)
```

where usable_x_cols are as follows:

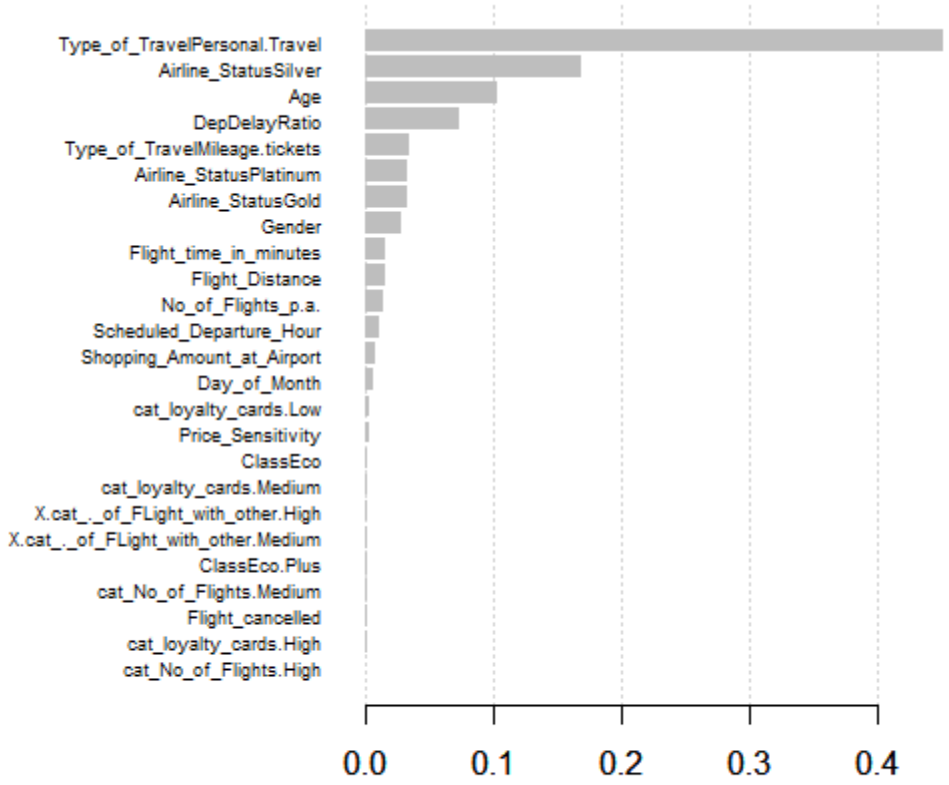
- Gender
- Price_Sensitivity
- Shopping_Amount_at_Airport
- Day_of_Month
- Scheduled_Departure_Hour
- Flight_cancelled

- Flight_time_in_minutes
- Flight_Distance
- Airline_StatusGold
- Airline_StatusPlatinum
- Airline_StatusSilver
- Type_of_TravelMileage.tickets
- Type_of_TravelPersonal.Travel
- ClassEco ClassEco.Plus
- cat_loyalty_cards.Low
- cat_loyalty_cards.Medium
- cat_loyalty_cards.High
- cat_No_of_Flights.Medium
- cat_No_of_Flights.High
- X.cat_._of_FLight_with_other.Medium
- X.cat_._of_FLight_with_other.High
- Age
- DepDelayRatio
- No_of_Flights_p.a.

A large number of columns were used since even with ~104k rows being used in the training set, xgboost models were still incredibly fast to run. Using all these columns would also provide a non-biased approach, letting the algorithm find the most important features. To attempt to get the best model and get an average of the most important features seven xgboost models were trained and evaluated on the test data using the following parameters:

1. Eta = 0.3; max_depth = 6; nrounds = 250
2. Eta = 0.3; max_depth = 3; nrounds = 500
3. Eta = 0.3; max_depth = 6; nrounds = 500
4. Eta = 0.1; max_depth = 3; nrounds = 1000
5. Eta = 0.3; max_depth = 6; nrounds = 1000
6. Eta = 0.3; max_depth = 3; nrounds = 2000
7. Eta = 0.1; max_depth = 3; nrounds = 5000

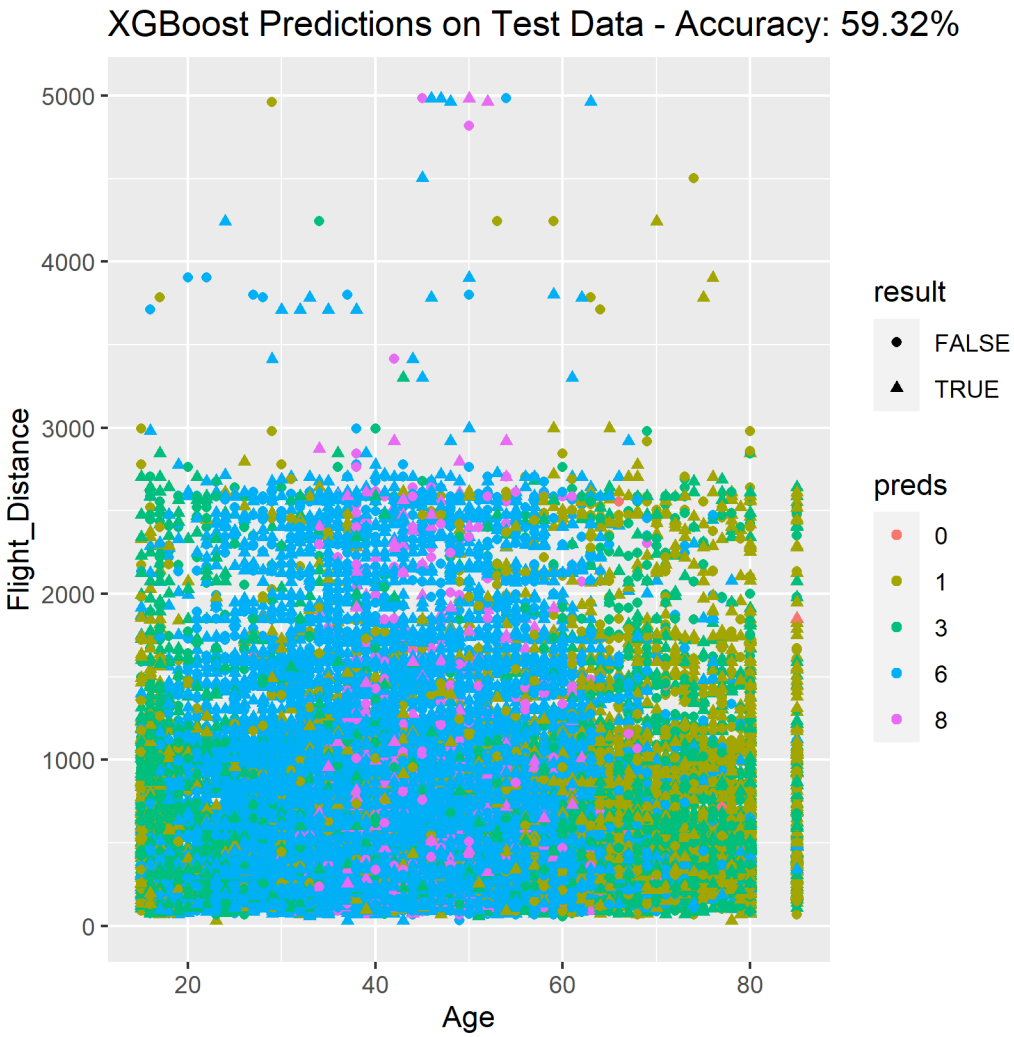
For each model the important features were plotted like below to get an understanding of which features the algorithm found to be most important for predicting satisfaction scores.



After viewing all of the feature importance plots for each of the seven classifiers, the following features were shown to be the most important to the model. When making recommendations to improve customer satisfaction, the recommendations should be focused on improving or targeting these top features. They should also be the subset of features used in any further modeling.

- Type_of_TravelPersonal.Travel – by far the most important feature across all models
- Airline_StatusSilver
- Age
- DepDelayRatio
- Flight_Distance
- Flight_time_in_minutes
- Gender
- Airline_StatusPlatinum
- Scheduled_Departure_Hour
- Shopping_Amount_at_Airport
- Airline_StatusGold
- Price Sensitivity
- Day_of_Month
- Type_of_TravelMileage.Tickets

Among all of the xgboost models trained, the classifier #4, with the parameters, Eta = 0.1; max_depth = 3; nrounds = 1000, had the lowest merror 40.68% (or highest accuracy of 59.32%). Using the Flight_Distance and Age variables, the below image plots the model's predication and a triangle if the prediction is correct, and a circle if incorrect. Just looking at Age, it appears that the model predicts the value 6 (a satisfaction score of 4) the most, and also in the Age range where customers are likely to give a higher satisfaction score, as shown in the exploratory data analysis section on demographics. The lower scores predicted are towards the edges of the Age spectrum, which also mirrors the pattern seen in the demographics section. With nine different classes to predict, a ~60% accuracy rate is quite satisfactory.



Support Vector Machine Classification

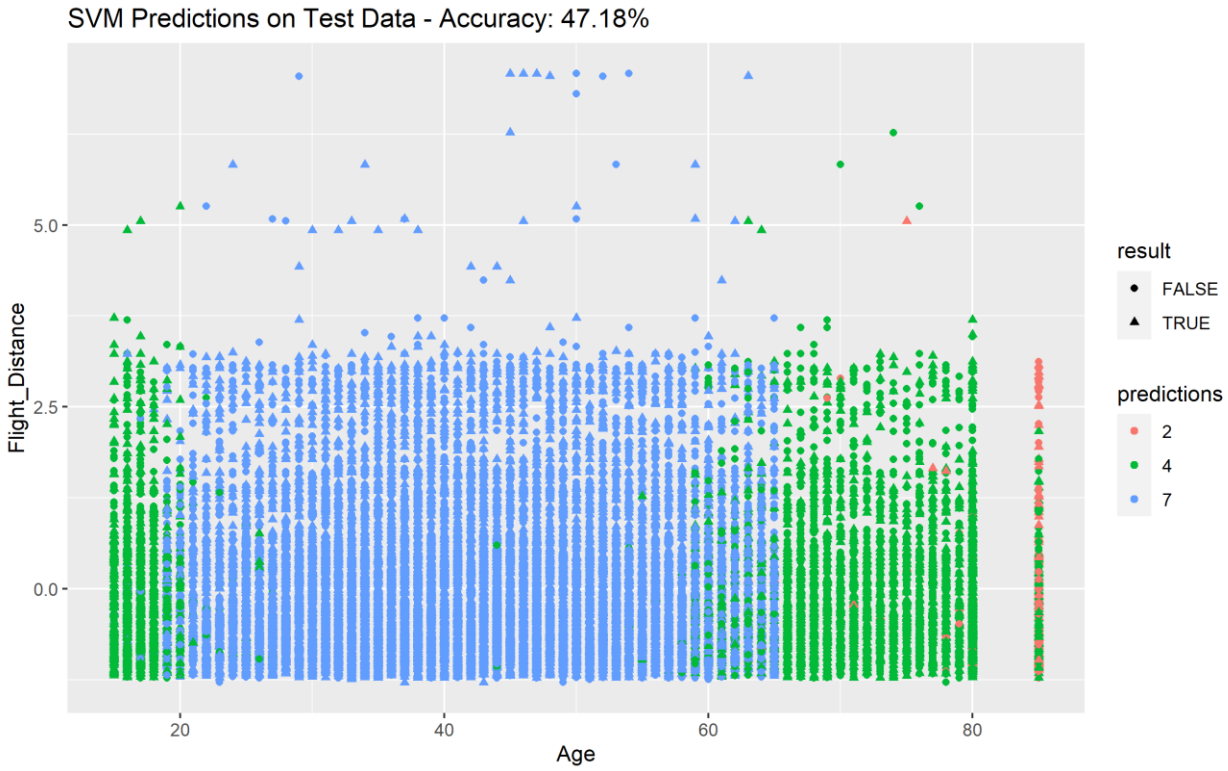
Support vector machines are powerful machine learning algorithms that attempt to find "support vectors" that linearly divide the data with the least amount of error for classification or regression. The support vectors try to find the least amount of error by dividing the data with a gap that is as wide as possible. When the data is not linearly separable, the "kernel trick" can be used to map the data into a higher dimensional space so the algorithm can attempt to separate it. Since the dataset we are using will not be linearly separable, we will have to apply a kernel function to the support vector machine. The radial basis function will be used in all the svm models created to apply the kernel trick. To implement a support vector machine, we will use the e1071 library to use the *svm* function for classification. Just as we did for the xgboost modeling, we will have to process the data further to prepare it for classification. The Gender and Flight_cancelled variables are transformed in the same manner as for xgboost, and the Satisfaction scores will be mapped similarly as well, except that the e1071 package does not require the class numbers to start at zero:

- 1 = 1
- 2 = 2
- 2.5 = 3
- 3 = 4
- 3.4 = 5
- 3.5 = 5
- 4 = 7
- 4.5 = 8
- 5 = 9

Additional processing was to scale any continuous variables to speed up the training time of the svm models. Scaling a variable is the process of subtracting a column's value by the mean of the column and dividing it by the standard deviation of the column to get the values centered around zero. The variable is then on a similar scale as other scaled variables, and truly shows how much larger that value is compared to the rest of the column. Scaled columns:

- Shopping_Amount_at_Airport
- Flight_time_in_minutes
- Flight_Distance
- No_of_Flights_p.a.
- Eating_and_Drinking_at_Airport
- Departure_Delay_in_Minutes
- Arrival_Delay_in_Minutes

The training and test data were created using the same logic used in linear regression and xgboost. Given how long a svm can take to train on ~104k rows of training (4-10+ hours), it was best to only use a small subset of the columns in the training data, and only two models were trained. The models were evaluated based on the accuracy score, which is the correct number of predictions made on the test data divided by the number of predictions made. The best svm mode achieved a poor accuracy of 47.18%, and took ~4 hours to train. The plot below mirrors the prediction plot shown in the xgboost section and shows similar patterns of predicting satisfaction scores. The most frequent customer ages and most satisfied customers seen in the exploratory data analysis section, are being predicted with a score of 7 (4) which is the most common score in the data. In this model the ends of the spectrum are again predicted to give a lower satisfaction score, however the svm model differs from xgboost in the fact that is only predicts very poor satisfaction scores for the oldest age group. Given the results of this model vs the xgboost model and the linear regression model, the svm model should not be used a predictor of customer satisfaction scores.


SVM Predictions on Test Data - Accuracy: 47.18%
Flight_Distance
5.0
2.5
0.0
20
40
60
80
Age
result
FALSE
TRUE
predictions
2
4
7

Conclusion

After multiple months of data transformation, analysis, and usage of both simple and advanced algorithms, I am confident that the observations and recommendations listed below will be insightful, accurate, and actionable. Given the analysis, the xgboost model proved to be the best model for predicting customer satisfaction, and should be leveraged by each company to predict whether their customer will be satisfied with their air travel experience. The hope is for all airlines in the industry to leverage the whole analysis summarized in this paper to improve their customer satisfaction scores, and thus their top and bottom lines, by creating a better customer experience.

Summary of Observations

- Airline with lower cancellation and departure delay rates score higher on customer satisfaction
- Males and females are equally price sensitive, but males tend to give a higher satisfaction score
- The youngest and oldest age groups are the most sensitive to price changes
- Personal travel is the most important determinant in customer satisfaction score
- Shorter flights lead to more satisfied customers
- Personal entertainment such as eating, drinking, an shopping are likely to increase satisfaction scores
- Airlines with a better reputation are likely to score higher

Final Recommendations

1. **Target customers from the ages of 35 to 45 for personal travel.** Customers in this age range have the highest satisfaction amongst any subset of the population, and a customer traveling for
2. **Reduce the proportion of cancellations and departure delays by collaborating with airports to improve runways**. Runways that are constantly backed should be assessed for expansion and usage optimization, and runways that cannot handle weather or are poorly maintained may need to be overhauled. The airport is likely a root cause of delays and cancellations that the airline company gets blamed for, at no fault of their own. Airlines should leverage their financial power to ensure that a proportion of fees paid to the airports is used to improve runway conditions and operations.
3. **Accommodate elderly passengers with early plane entry/departure and additional assistance around the airport and into/out of the plane**. The oldest age group scores the lowest in customer satisfaction by a wide margin, and this is most likely driven by the physical burden of travel, and the lack of comfort provided to them during the air travel process.
4. **Offer special price incentives and food/drink specials onboard to the age 21-30 age group**. Given that this group is price sensitive when compared to the population (due to being in the early stages of wealth creation compared to older customers) they will focus on finding the cheapest ticket but are hungry for travel experience. Offering drink and food specials onboard will be an additional incentive for this population since they are more prone to traveling to party destinations, and this will improve their experience on the way there. As an individual company, being able to breed customer loyalty early on will pay off in revenue and profitability in the long run.
5. **Attempt to corner the crushed business traveling customers**. Due to COVID-19 business travel has fallen to almost zero. However, in our data it was by the far the largest type of travel. Being

able to take care of these customers by making them feel safe and comfortable, can build brand loyalty that can be extracted in the future when the world begins to open again.

6. **Target younger customers to buy loyalty cards**. Younger people are more likely to use a loyalty card

Appendix

Appendix A

All code, images, and models can be found at the following location. Further descriptions on all the R scripts can be found here.

https://github.com/Alec-Schneider/FlightSurveyProject

Appendix B

Function to clean the entire dataset:

```
process_flight_survey <- function(data){


 data <- clean_column_names(data)

 data <- fillNAs(data)

 data$Satisfaction <- round(data$Satisfaction, 1)

 # Create columns for Departure Delay and Arrival Delay as a percent of Flight Time

 data <- data %>%

  mutate(DepDelayRatio =(Departure_Delay_in_Minutes / Flight_time_in_minutes),

       ArrDelayRatio =(Arrival_Delay_in_Minutes / Flight_time_in_minutes))

 # Fill in the NAs in ratio columns created from 0 / 0

 data$DepDelayRatio <- ifelse(is.na(data$DepDelayRatio), 0, data$DepDelayRatio)

 data$ArrDelayRatio <- ifelse(is.na(data$ArrDelayRatio), 0, data$ArrDelayRatio)


 data <- category_processing(data)

 data <- dummy_processing(data)


}
```

Appendix C

```
fillNAs <- function(data){
```

```
    # Impute the NAs in Satisfaction with the mean
    data$Satisfaction <- ifelse(is.na(data$Satisfaction),
                                mean(data$Satisfaction, na.rm = TRUE),
                                data$Satisfaction)

    # drop the 337 flights where the flight was not cancelled, and Arrival Delay in Minutes is NA.
    data <- data[!((data$Flight_cancelled == "No") & is.na(data$Arrival_Delay_in_Minutes)),]

    # impute the NAs in Arrival_Delay in
    data$Departure_Delay_in_Minutes <- ifelse(is.na(data$Departure_Delay_in_Minutes),
                                0,
                                data$Departure_Delay_in_Minutes)

    data$Arrival_Delay_in_Minutes <- ifelse(is.na(data$Arrival_Delay_in_Minutes),
                                    0,
                                    data$Arrival_Delay_in_Minutes)

    data$Flight_time_in_minutes <- ifelse(is.na(data$Flight_time_in_minutes),
                                    0,
                                    data$Flight_time_in_minutes)


    return(data)
}
```

Appendix D

Script used to map all flight patterns in the dataset:

https://github.com/Alec-Schneider/FlightSurveyProject/blob/main/FlightMapping.R

Appendix E

```
category_processing <- function(data){
 # Create categories for the Number of loyalty cards a customer has
 data <- data %>%
  mutate(cat_loyalty_cards=cut(No._of_other_Loyalty_Cards,
                       breaks=c(-Inf, 0, 3, 6, 12),
                       labels=c("None", "Low", "Medium", "High"))) %>%
  mutate(`cat_%_of_FLight_with_other`=cut(`%_of_Flight_with_other_Airlines`,
                                 breaks=c(-Inf, quantile(data$`%_of_Flight_with_other_Airlines`,
                                          c(1/3, 2/3, 1))),
                                 labels=c("Low", "Medium", "High"))) %>%
  mutate(cat_No_of_Flights=cut(No_of_Flights_p.a.,
                       breaks=c(-Inf, quantile(data$No_of_Flights_p.a.,
                              c(1/3, 2/3, 1))),
                       labels=c("Low", "Medium", "High"))) %>%
  mutate(cat_Age=cut(Age,
                breaks=c(-Inf, 20, 30, 40, 50, 60, 70, Inf)))



 return(data)
}
```

Appendix F

```
dummy_processing <- function(data) {
 # use a for loop to one hot encode all the
 # Will need to manuall do cat_%_of_FLight_with_other do to the % sign being an issue
 dummy_cols <- c("Airline_Status", "Type_of_Travel", "Class", "cat_loyalty_cards",
            "cat_No_of_Flights", "cat_Age")
```

```
 dummy_df <- data.frame()
 for (col in dummy_cols){
  dmy <- dummyVars(paste("~", col, sep = " ") , data=data)
  if (dim(dummy_df)[1] != 0){
   # need to return n-1 values of the categorical variable for analysis
   dummy_df <- cbind(dummy_df, data.frame(predict(dmy, newdata = data))[,-1])
  } else {
   dummy_df <- data.frame(predict(dmy, newdata = data))[,-1]
  }
 }
 dmy <- dummyVars(~ `cat_%_of_FLight_with_other`, data=data)
 dummy_df <- cbind(dummy_df, data.frame(predict(dmy, newdata = data))[,-1])

 data <- cbind(data, dummy_df)
 return(data)
}
```

Appendix G

Vacation Destinations:

- Dallas/Fort Worth, TX
- Chicago, IL Nashville, TN
- New York, NY Miami, FL Los Angeles, CA
- Houston, TX Charleston, SC Lexington, KY
- Aspen, CO Washington, DC Tampa, FL
- Salt Lake City, UT San Antonio, TX Memphis, TN
- Denver, CO Key West, FL San Diego, CA
- Tucson, AZ Boston, MA New Orleans, LA
- Orlando, FL Dallas, TX Phoenix, AZ
- Fort Myers, FL West Palm Beach/Palm Beach, FL Myrtle Beach, SC
- Fort Lauderdale, FL Las Vegas, NV San Francisco, CA
- Honolulu, HI Kahului, HI Kona, HI
- Lihue, HI Palm Springs, CA Portland, OR

- Long Beach, CA Ponce, PR Aguadilla, PR
- Reno, NV Guam, TT Daytona Beach, FL
- Panama City, FL

Appendix H

All modeling scripts:

1. Linear Regression
2. Xgboost
3. SVM
4. Naïve Bayes

Appendix I

Attributes Name:

1. **Satisfaction** – it is rated from 1 to 5, that how satisfied is the customer?
    a. 5 means higher satisfied, and 1 is lowest level of satisfaction.
2. **Airline Status** – each customer has a different type of airline status or package, which are platinum, gold, silver, and blue.
3. **Age** – the specific customer's age. That is starting from 15 to 85 years old.
4. **Gender** – male or female.
5. **Price Sensitivity** – the grade to which the price affects to customers purchasing. The price sensitivity has a range from 0 to 5.
6. **Year of First Flight** – this attributes shows the first flight of each single customer. The range of year of the first flight for each customer has been started in 2003 until 2012.
7. **No of Flights p. a.** – this could be the number of flights that each customer has taken. The range starting from 0 to 100.
8. **Percent of Flight with other Airlines** – if we were Southeast Airline, we would like to know how many time that customer fly with other Airlines.
9. **Type of Travel** – is provide three traveling purpose for each consumer, which are business travel, mileage tickets that based on loyalty card, and personal travel like to see the family or in vacation
10. **No. Of other Loyalty Cards** – it is kind of membership card of each customer, that for retail establishment to gain a benefits such as, discounts.
11. **Shopping Amount at Airport** – showing the costumer's result of how many products have been purchased. The range of shopping amount is from 0 to 875.
12. **Eating and Drinking at Airport** – it is the quantity eating and drinking per each consumer at the airport. The masseur of how often for eating and drinking, which is 0 to 895.
13. **Class** – it consisted of three different kinds of service level such as, business, and economy plus, economy. Moreover, customers have optional to choose their seat.
14. **Day of Month** – it means the traveling day of each costumer. In this attribute, shows total of 31 days of the month.
15. **Flight date** – all of these data are abbreviate the passenger's flight date travel, which were since 2014 and only in January, February, and March.

16. **Airline Code** – basically, it is unique two or three digits that mean what is the specific type of airline. There are several codes that consumers have been going with. For example, AA, AS, B6, and DL.
17. **Airline Name** – There are several airlines company names such as, West Airways, Southeast Airlines Co, and FlyToSun Airlines Inc. This attribute provide what airline name that passenger have been used.
18. **Origin City** – refers to actual city that customers have departed from. For example, Yuma AZ, Waco TX, and Toledo HO.
19. **Origin State** – same thing as origin city such as, what state that customers have departed from? A good example, Texas, Ohio, Alaska, and Utah.
20. **Destination City** – the place to which passenger travels to. For example, Akron HO, Alpena MI, Austin TX, and Boston MA.
21. **Destination State** – also, it is the same thing as origin city, such as, to what state passenger travel to? Some example of destination states, Alaska, Kentucky, Iowa, and Florida.
22. **Scheduled Departure Hour** – the specific time at which passengers are scheduled to depart. In this data in scheduled departure hour is starting at 1 am until 23 pm.
23. **Departure Delay in Minutes** – which are minutes of departure delayed for each passenger, when compared to schedule. In this data the rage are starting from 0 until 1128 minutes.
24. **Arrival Delay in Minutes** – how many minutes of arrival delayed of each passenger. Rang of delayed minutes in this data are starting from 0 until 1115 minutes.
25. **Flight Cancelled** – occurs when the airline dose not operates the flight at all, and that is for a certain reason.
26. **Flight time in minutes** – indicate to period time to the destination.
27. **Flight Distance** – the extent of space between two places. Also, that means how many minutes are passenger traveling between two different places. Rang in this data starting from 31 until 4983 minutes.

Arrival Delay greater 5 Minutes – It means the delay of arrival airline time, which is more than 5 minutes per each passenger in the data.