# IST 687: APPLIED DATA SCIENCE

## Final Project Report

Instructors: Jeffrey Saltz; Gary Krudys

Team Members:
Sharat Sripada

# TABLE OF CONTENTS



## Introduction

The final report was prepared solely by Sharat Sripada, a student of course IST-687 taught by Professor Gary Krudys during the timeframe Jan-Mar 2020.

The initial dataset that was provided was 16MB in size and comprised data pertinent to Airline Passengers, their preferences in Airlines bookings, Routes, Satisfaction and factors that were likely to determine a choice. The goal was to be able to interpret the data-set using advanced concepts of:

- Statistics - descriptive and inferential
- Programming skills - R programming language, a proficiency that was gained during the course of study.

# Business Questions

The intent when analyzing the airline booking data-set was to represent an airline and play the role of a Business Analyst or Consultant with specific function of Data-Scientist. In this context, it was determined from initial analysis that *Southeast Airlines Co.* was not the most preferred airline among passengers and that therefore would offer sufficient challenge and scope to analyze and make or show improvements.

The analysis will broadly be based on the following:

1. Determine commonly top/worst performing airlines in the segment
2. Key factors or variables likely to impact passenger sentiment, airline bookings etc.
3. Using advanced prediction techniques provide consulting suggestions

# Methodology

## Importing the data, cleansing, munging and preparation

- Import the data using read-xls function of R
- Omit NAs for descriptive statistics
- Identifying variables that would aid the analysis

After spending hours analyzing the data decided to review the data-set with respect to the following specific variables:

| SL | Variable Name | Meaning |
|---|---|---|
| 1 | Satisfaction | It is rated from 1 to 5, that how satisfied is the customer?<br>1. 5 means higher satisfied, and 1 is lowest level of satisfaction. |
| 3 | 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. |
| 4 | No. of other Loyalty Cards | It is kind of membership card of each customer, that for retail establishment to gain a benefit such as, discounts. |
| 5 | Airline Name | There are several airlines company names such as, West Airways, Southeast Airlines Co, and FlyToSun Airlines Inc. This attribute provides what airline name that passenger have been used. |
| 6 | 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 |
| 7 | Airline Status | Each customer has a different type of airline status or package, which are platinum, gold, silver, and blue. |
| 8 | Price Sensitivity | The grade to which the price affects to customers purchasing. The price sensitivity has a range from 0 to |

| | | |
|---|---|---|
| 9 | Airline Status | Each customer has a different type of airline status or package, which are platinum, gold, silver, and blue. |

- Recoding some of the variables to be able to use for predictive analysis

```
df_recode <- df_recode %>% mutate(`Airline Name`=recode(`Airline Name`,
                                                        "EnjoyFlying Air Services" = 1,
                                                        "FlyFast Airways Inc." = 2,
                                                        "FlyHere Airways" = 3,
                                                        "FlyToSun Airlines Inc." = 4,
                                                        "GoingNorth Airlines Inc." = 5,
                                                        "West Airways Inc." = 6,
                                                        "OnlyJets Airlines Inc." = 7,
                                                        "Northwest Business Airlines Inc." = 8,
                                                        "Oursin Airlines Inc." = 9,
                                                        "Paul Smith Airlines Inc." = 10,
                                                        "Cheapseats Airlines Inc." = 11,
                                                        "Sigma Airlines Inc." = 12,
                                                        "Southeast Airlines Co." = 13,
                                                        "Cool&Young Airlines Inc." = 14))
df_recode <- df_recode %>% mutate(`Type of Travel`=recode(`Type of Travel`,
                                                          "Business travel" = 1,
                                                          "Personal Travel" = 2,
                                                          "Mileage tickets" = 3))
df_recode <- df_recode %>% mutate(`Airline Status`=recode(`Airline Status`,
                                                                "Platinum" = 1,
                                                                "Gold" = 2,
                                                                "Silver" = 3,
                                                                "Blue" = 4))
```

## Initial analysis

Initial analysis was based on determining key factors effecting passenger sentiment, highest/lowest booking etc.

### Descriptive statistics

Predominantly, used the mean() function in R skipping NA values where applicable when determining parameters related to delays, satisfaction and other factors

### First cut of analysis

Below are outputs of functions implemented to provide conclusive data regarding variables that seem obvious:

- Classify airlines in terms of delay (arrival/departure)

```
> get_best_worst_delays(df, consulting_airline)
FlyFast Airways Inc. arrives on average 42.83042 minutes late
West Airways Inc. arrives on average 6.691395 minutes late
Southeast Airlines Co. arrives on average 19.20768 minutes late
```

- Based on the booking volumes determine most/least popular airlines

```
> pop_score <- get_popular_carrier(df, consulting_airline)
Cheapseats Airlines Inc. is the most used airline ( 20.06175 % of all bookings)
Cool&Young Airlines Inc. is the least used airline ( 0.9916159 % of all bookings)
                                tapply.my_data..Airline.Name...my_data..Airline.Name...length.                  Airline Names
Cheapseats Airlines Inc.                                                                 26058         Cheapseats Airlines Inc.
Sigma Airlines Inc.                                                                      17037              Sigma Airlines Inc.
FlyFast Airways Inc.                                                                     15407             FlyFast Airways Inc.
Northwest Business Airlines Inc.                                                         13840 Northwest Business Airlines Inc.
Paul Smith Airlines Inc.                                                                 12248         Paul Smith Airlines Inc.
Oursin Airlines Inc.                                                                     10968             Oursin Airlines Inc.
Southeast Airlines Co.                                                                    9577           Southeast Airlines Co.
EnjoyFlying Air Services                                                                  8927         EnjoyFlying Air Services
OnlyJets Airlines Inc.                                                                    5395           OnlyJets Airlines Inc.
FlyToSun Airlines Inc.                                                                    3407           FlyToSun Airlines Inc.
FlyHere Airways                                                                           2481                  FlyHere Airways
West Airways Inc.                                                                         1688                West Airways Inc.
GoingNorth Airlines Inc.                                                                  1568         GoingNorth Airlines Inc.
Cool&Young Airlines Inc.                                                                  1288         Cool&Young Airlines Inc.
Southeast Airlines Co. is at # 7 ( 7.373219 % of all bookings)
```

- Determine the mean satisfaction score per airline

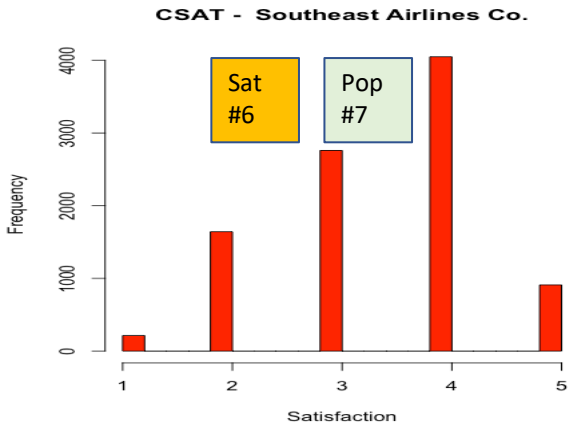
```
> sat_score <- get_satisfaction_data(df, consulting_airline)
                     airline_names as.numeric.sat_score.
6                West Airways Inc.                3.486967
14         Cool&Young Airlines Inc.                3.442547
4            FlyToSun Airlines Inc.                3.425301
10         Paul Smith Airlines Inc.                3.399167
11              Sigma Airlines Inc.                3.397547
13           Southeast Airlines Co.                3.396888
3                   FlyHere Airways                3.395002
8  Northwest Business Airlines Inc.                3.394666
9               Oursin Airlines Inc.                3.386534
1          EnjoyFlying Air Services                3.360199
12         Cheapseats Airlines Inc.                3.357318
2              FlyFast Airways Inc.                3.352567
7            OnlyJets Airlines Inc.                3.346803
5          GoingNorth Airlines Inc.                3.297194
West Airways Inc. has highest average CSAT( 3.486967  )
GoingNorth Airlines Inc. has lowest average CSAT( 3.297194  )
Southeast Airlines Co. is at # 6 (average CSAT 3.396888  )
```

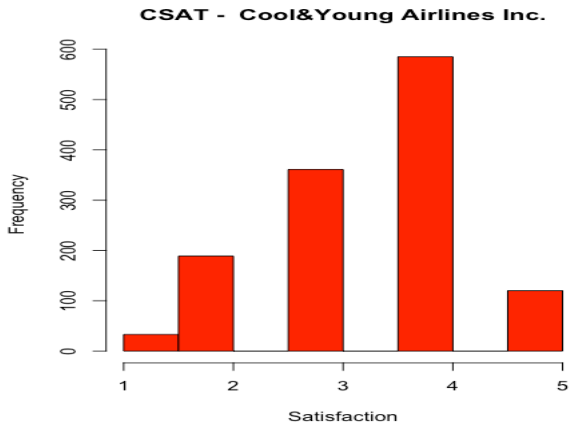
- Total number of loyalty cards offered per airline

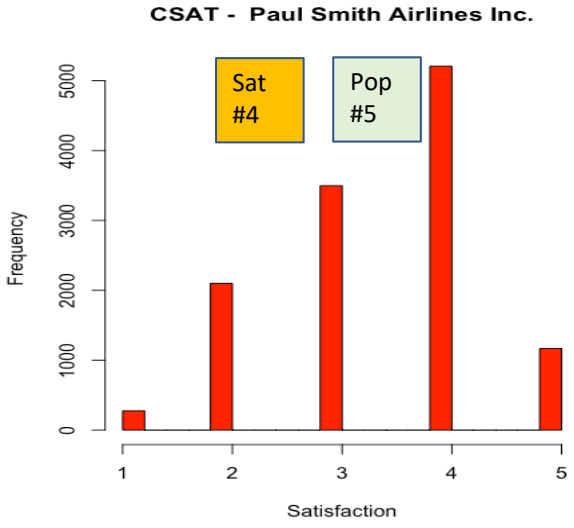
```
> # Get loyalty data per Airline
> loy_score <- get_loyalty_membership_data(df, consulting_airline)
                    airline_names as.numeric.loyalty_cards.
12         Cheapseats Airlines Inc.                   23209
11              Sigma Airlines Inc.                   14859
2              FlyFast Airways Inc.                   13763
8  Northwest Business Airlines Inc.                   12115
10         Paul Smith Airlines Inc.                   10627
9              Oursin Airlines Inc.                    9679
13           Southeast Airlines Co.                    8439
1          EnjoyFlying Air Services                    7949
7            OnlyJets Airlines Inc.                    4728
4            FlyToSun Airlines Inc.                    3082
3                  FlyHere Airways                     2226
6                West Airways Inc.                     1568
5          GoingNorth Airlines Inc.                    1422
14         Cool&Young Airlines Inc.                    1128
Cheapseats Airlines Inc. has highest loyalty-cards offered/used ( 23209  )
Cool&Young Airlines Inc. has lowest loyalty-cards offered/used ( 1128  )
Southeast Airlines Co. is at # 7 (loyalty-cards offered/used 8439  )
```

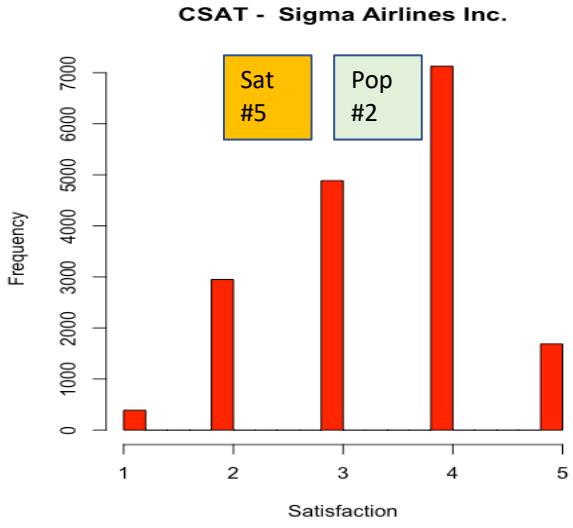
## Visualizations

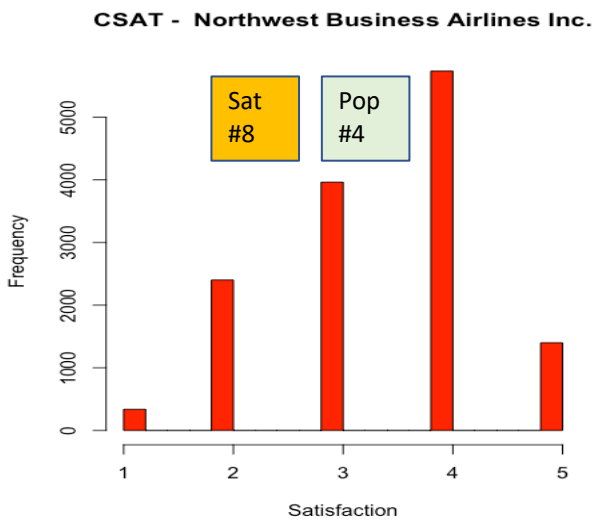
Visual techniques helped to ascertain correlation between variables. Below for instance is a histogram plot of variable 'Satisfaction' for passengers by variable 'Airline Name' type

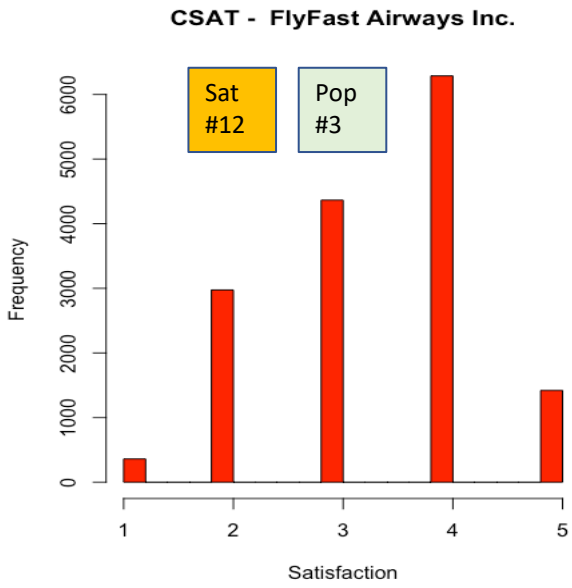


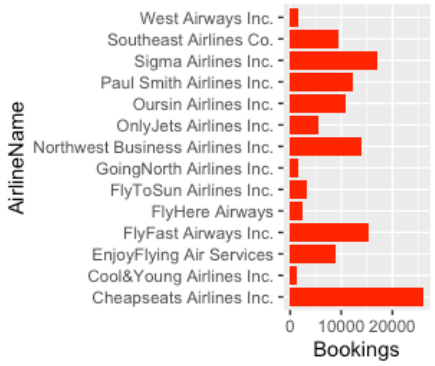



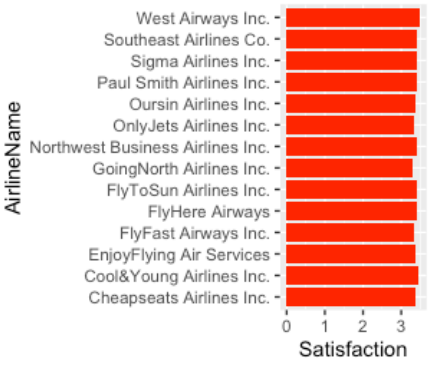


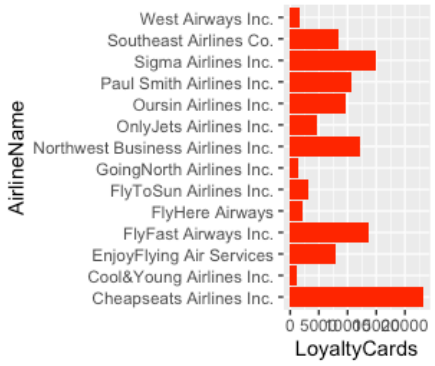




The visualization helped ascertain that the Satisfaction variable largely had little bearing on how popular an airline with respect to passenger booking.

## Summary of the initial analysis



Hence, derive the following intermediate or visual conclusion:

- Delay & Popularity of airline - **WEAK**
- Satisfaction & Popularity of airline - **WEAK**
- Loyalty cards & Popularity of airline - **STRONG**
- Delay & Satisfaction – **MEDIUM**

# Data Modeling Techniques and Predictive Analysis

## Linear Regression

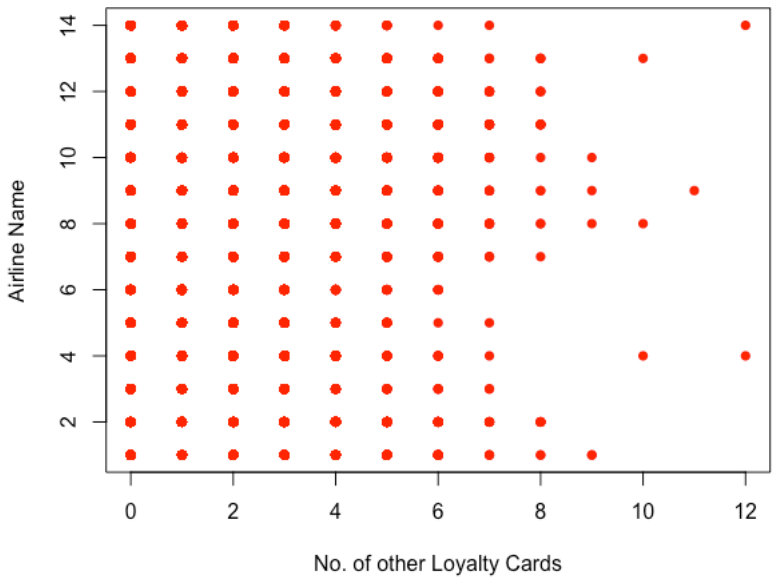
Based on the initial analysis we now move to corroborate intermediate conclusions with known techniques or methodologies related to Linear Regression modeling. Since the model is highly efficient and automated there is scope to expand the number of variables.

Following are some of the variables from the data-set that have been identified:

- Arrival Delay in minutes
- Departure Delay in minutes
- Satisfaction
- No. of other Loyalty cards
- Airline Names
- Airline Status
- Type of Travel
- Price Sensitivity

## Detailed Analysis

### *Correlation between Airline Name & No. of other Loyalty cards*




Here's the corresponding statistical analysis/R-square value:

```
> lm_model <- lm(formula=`Airline Name`~`No. of other Loyalty Cards`, data=df_models)
> summary(lm_model)

Call:
lm(formula = `Airline Name` ~ `No. of other Loyalty Cards`, data = df_models)

Residuals:
   Min     1Q Median     3Q    Max
-7.381 -2.366  1.619  2.650  5.801

Coefficients:
                            Estimate Std. Error t value Pr(>|t|)
(Intercept)                 8.380740   0.013760 609.059   <2e-16 ***
`No. of other Loyalty Cards` -0.015104   0.009532  -1.585    0.113
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 3.88 on 127146 degrees of freedom
Multiple R-squared:  1.975e-05, Adjusted R-squared:  1.188e-05
F-statistic: 2.511 on 1 and 127146 DF,  p-value: 0.1131

> summary(lm_model)$r.square
[1] 1.974913e-05
```

## Conclusion-1

Low R-square values indicates low correlation & we are therefore unable to reject the NULL hypotheses – ‘No correlation exists between Airline Names and No. of other Loyalty Cards.

### *Correlation between some of the other variables*

```
Call:
lm(formula = Satisfaction ~ `Airline Status`, data = df_models)

Residuals:
    Min      1Q  Median      3Q     Max
-3.2193 -0.5605  0.1101  0.7689  1.7689

Coefficients:
                Estimate Std. Error t value Pr(>|t|)
(Intercept)     4.548698   0.012114  375.50   <2e-16 ***
`Airline Status` -0.329411   0.003346  -98.44   <2e-16 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 0.9317 on 127146 degrees of freedom
Multiple R-squared:  0.07082,   Adjusted R-squared:  0.07081
F-statistic:  9690 on 1 and 127146 DF,  p-value: < 2.2e-16
```

```
Call:
lm(formula = Satisfaction ~ `Airline Status`, data = df_models)

Residuals:
    Min      1Q  Median      3Q     Max
-3.2193 -0.5605  0.1101  0.7689  1.7689

Coefficients:
                Estimate Std. Error t value Pr(>|t|)
(Intercept)     4.548698   0.012114  375.50   <2e-16 ***
`Airline Status` -0.329411   0.003346  -98.44   <2e-16 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 0.9317 on 127146 degrees of freedom
Multiple R-squared:  0.07082,   Adjusted R-squared:  0.07081
F-statistic:  9690 on 1 and 127146 DF,  p-value: < 2.2e-16
```

## Conclusion-2

Low R-square values/correlation co-efficient between several combination of variables

### *Parsimonious Model/step-function*

Created the following data-frame considering all possible variables and ran a step() function so we could the Parsimonius Model to help determine the least possible number of variables that can predict Airline Name

Here's the code excerpt & the corresponding result of the step() function:

```
colnames(df_models) <- c("Airline Name",
                         "Satisfaction",
                         "No. of other Loyalty Cards",
                         "Departure Delay in Minutes",
                         "Arrival Delay in Minutes",
                         "Price Sensitivity",
                         "Type of Travel",
                         "Airline Status")
.
.
lm_model <- lm(formula=`Airline Name`~.,
data=df_models)
step(lm_model, data=df_models,
direction="backward")
```

```
                                s` +
                             Df Sum of Sq     RSS    AIC
<none>                                     1904430 344145
- `No. of other Loyalty Cards`  1      38.8 1904469 344146
- `Departure Delay in Minutes`  1    4602.2 1909032 344450
- `Arrival Delay in Minutes`    1    6935.6 1911366 344605

Call:
lm(formula = `Airline Name` ~ `No. of other Loyalty Cards` +
    `Departure Delay in Minutes` + `Arrival Delay in Minutes`,
    data = df_models)

Coefficients:
                (Intercept)  `No. of other Loyalty Cards`  `Departure Delay in Minutes`    `Arrival Delay in Minutes`
                    8.45120                      -0.01531                       0.01903                      -0.02304
```

## *Predictive Analysis*

Ran the predictive analysis using Linear regression, factoring in results from Parsimonius model

- All variables shortlisted

```
> lmoutput <- lm(formula=`Airline Name`~., data=trainData)
> test <- data.frame(testData$`Airline Name`,
+                    testData$`Satisfaction`,
+                    testData$`No. of other Loyalty Cards`,
+                    testData$`Departure Delay in Minutes`,
+                    testData$`Arrival Delay in Minutes`,
+                    testData$`Price Sensitivity`,
+                    testData$`Type of Travel`,
+                    testData$`Airline Status`
+                    )
> colnames(test) <- c("Airline Name", "No. of other Loyalty Cards",
+                     "Satisfaction",
+                     "Departure Delay in Minutes",
+                     "Arrival Delay in Minutes",
+                     "Price Sensitivity",
+                     "Type of Travel",
+                     "Airline Status")
> lmpredict <- round(predict(lmoutput, test, type="response"))
> lm_compTable <- data.frame(testData[,1], lmpredict)
> colnames(lm_compTable) <- c('Test', 'Pred')
> percentage_lm <- length(which(lm_compTable$Test == lm_compTable$Pred))/dim(lm_compTable)[1]
> percentage_lm
[1] 0.1144327
```

- Variables as a result of Parsimonius Model/Step function did not yield very useful prediction as well at 11.48%

NOTE

Since the variable 'Airline Name' was recoded from text/chr to a numeric it was important to determine the absolute prediction than a Root Mean Square Error (RMSE). RMSE itself was low at ~3, but in this context is not relevant.

## Support Vector Machines (SVM)

The linear regression model proved to be not successful with predicting variable 'Airline Names' from a set of variables likely to have causation. We now move to exploring the data-set using techniques or algorithms outlined in support vector machines.

### KSVM model

Below is code excerpt using KSVM model

```
library(kernlab)
cutpoint2_3 <- floor((2 * length(randindex) /3))
trainData <- df_models[randindex[1:cutpoint2_3],]
testData <- df_models[randindex[(cutpoint2_3 + 1):length(randindex)],]

ksvmoutput <- ksvm(`Airline Name`~., data=trainData,
                   kernel="rbfdot", #kernel function that projects the low dimensional problem into higher dimensional space
                   kpar="automatic", #params used to control radial function kernel(rbfdot)
                   C=10, #C -> cost of constraints
                   cross=10, #use 10 fold cross-validation in this model
                   prob.model=TRUE)
```

Using this prediction accuracy was at 11.74%:

```
> ksvmpredict <- round(predict(ksvmoutput, test, type="response"))
> ksvm_compTable <- data.frame(testData[,1], ksvmpredict)
> colnames(ksvm_compTable) <- c('Test', 'Pred')
> percentage_ksvm <- length(which(ksvm_compTable$Test == ksvm_compTable$Pred))/dim(ksvm_compTable)[1]
> percentage_ksvm
[1] 0.1174528
```

### SVM model

Finally, ran the SVM linear model that yielded similar results of accuracy as well. Below is the code:

```
library(e1071)
svmoutput <- svm(`Airline Name`~., data=trainData,
          kernel="linear", #kernel function that projects the low dimensional
problem into higher dimensional space
          cross=10, #use 10 fold cross-validation in this model
          scale=FALSE)

svmpredict <- round(predict(svmoutput, test, type="response"))
svm_compTable <- data.frame(testData[,1], svmpredict)
colnames(svm_compTable) <- c('Test', 'Pred')

percentage_svm <- length(which(svm_compTable$Test ==
svm_compTable$Pred))/dim(svm_compTable)[1]
```

NOTE

Both the SVM and KSVM models take several hours to complete on the large dataset owing to complex underlying computations. Some of the outputs may therefore not be available here.

### Conclusion

Neither the linear nor KSVM/SVM models were accurate in predicting the Airlines that passengers were likely to book based on the filtered data that seemed likely to influence passengers.
Ideally, in use-cases related to human behavior a prediction accuracy of 30% is recommended however, we see a far lower number. Given this, I would use KSVM model with a slightly better prediction rate for all analysis further.

## Actionable Insights

From analysis earlier passengers offered with Loyalty cards or rewards (may include discounts etc.) preferred a certain Airline/carrier. Below is the summary of that analysis where Cheapseats Airline Inc, had the highest passenger bookings and had also offered the highest number of loyalty cards:

- Cheapseats Airlines Inc. has highest loyalty-cards offered/used (23209)
- Cool&Young Airlines Inc. has lowest loyalty-cards offered/used (1128)

*Southeast Airlines Co*. is at # 7 (loyalty-cards offered/used 8439 )

Based on the above & as an experiment, 'No. of loyalty cards' variable was adjusted in the following manner for SouthEast Airlines Co.:

- If no cards were offered, offer one card at the least
- If cards were already offered then increase the number of loyalty cards by 10%, 50% and 100%

```
df_models_order <- df_models[order(df_models$`Airline Name`),]
rownames(df_models_order) <- NULL
df_models_my_airline <- df_models_order[df_models_order$`Airline
Name` == 13,]
start_row <- as.numeric(rownames(df_models_my_airline)[1])
end_row <- start_row + nrow(df_models_my_airline) - 1
for (i in c(start_row:end_row)){
  if (df_models_order[i,3] == 0){
    df_models_order[i,3] <- 1
  } else {
    df_models_order[i,3] <- ceiling(df_models_order[i,3] * 2.0)
  }
}
```

Now, using the KSVM model following are the results of bookings of Southeast Airlines Co. as a direction prediction of increase in loyalty cards:


| Percentage increase in loyalty cards offered | Bookings per data set (in numbers) | Bookings per data set (in %) |
|---|---|---|
| 10% | 294 | 3.9% |
| 50% | 889 | 11.88% |
| 100% | 414 | 5.53% |

Clearly in my opinion variable 'No. of Loyalty cards' seems a mere distractor and cannot be used in any meaningful analysis. Also, that statement possibly holds some ground to the dataset in entirety. As a Data-Scientist or Business Analyst for Southeast Airlines Co. I would recommend getting more insightful data:

- Adding passenger first and last names can possibly help in determining how a passenger truly rated his/her experience when flying an airline provider and how that impacted flying again with the same airline in the future
- Specific satisfaction indices – related to delay and service

## Appendix

Following is all the code that was written in the R programming language for the purpose of analyzing the data-set

```
# Final-project

# Role - Business consultant for Southeast Airlines Co.
# Analyse current data-set to determine:
#  - standing of Southeast Airlines wrt to some KPIs - delay arrival,dept.
times, cust. satisfaction etc.
#  - improvement suggestions
#  - prediction of where the Airline would stand after implementing
improvements
install.packages("readxl")
library("readxl")
library(gridExtra)
library(ggplot2)

readfromxl <- function(xlpath) {
  return(read_excel(xlpath))
}

get_best_worst_delays <- function(my_data, my_airline) {
  # Summary should show us NAs which we will munge!
  summary(my_data)
  # Get data dim() prior munging
  dim(my_data)
  # Let's get data regarding the total delay in departure
  # and arrival times across each airline-type
  my_data$total_delay_time <- my_data$`Departure Delay in Minutes` +
my_data$`Arrival Delay in Minutes`
  airline_names <- unique(my_data$`Airline Name`)
  # > unique(my_data$`Airline Name`)
  # [1] "EnjoyFlying Air Services"         "FlyFast Airways Inc."
"FlyHere Airways"
  # [4] "FlyToSun Airlines Inc."           "GoingNorth Airlines Inc."
"West Airways Inc."
```

```
  # [7] "OnlyJets Airlines Inc."           "Northwest Business Airlines
Inc." "Oursin Airlines Inc."
  # [10] "Paul Smith Airlines Inc."         "Sigma Airlines Inc."
"Cheapseats Airlines Inc."
  # [13] "Southeast Airlines Co."           "Cool&Young Airlines Inc."
  # 14x Airlines in all!
  mean_delay_time <- character()
  for (name in airline_names) {
    mean_delay_time <- c(mean_delay_time,
mean(my_data$total_delay_time[my_data$`Airline Name` == name],
                                                 na.rm = TRUE))
  }
  # Make a data-frame with Airline-name & Mean-delay-time
  dfmeandelaytimes <- data.frame(airline_names,
as.numeric(mean_delay_time),
                                 stringsAsFactors = FALSE)
  # Worst airline in terms of delay
  worst_delay_index <-
which.max(dfmeandelaytimes$as.numeric.mean_delay_time.)
  cat(dfmeandelaytimes[worst_delay_index, 1],
      "arrives on average", dfmeandelaytimes[worst_delay_index, 2],
      "minutes late\n")
  # Best airline in terms of delay
  best_delay_index <-
which.min(dfmeandelaytimes$as.numeric.mean_delay_time.)
  cat(dfmeandelaytimes[best_delay_index, 1],
      "arrives on average", dfmeandelaytimes[best_delay_index, 2],
      "minutes late\n")
  # How Southeast compares to the best & worst
  my_airline_index <- which(dfmeandelaytimes$airline_names == my_airline)
  cat(my_airline, "arrives on average", dfmeandelaytimes[my_airline_index,
2],
      "minutes late\n")
}

get_popular_carrier <- function(my_data, my_airline){
  # Get a grouping of Airline-names by count (to get popular Airlines)
  my_df <- data.frame(tapply(my_data$`Airline Name`, my_data$`Airline
Name`, length))
  my_df$`Airline Names` <- row.names(my_df)
  total <- sum(my_df[,1])
  max_index <- which.max(my_df[,1])
  min_index <- which.min(my_df[,1])
  cat(row.names(my_df[max_index,]), "is the most used airline", "(",
      as.numeric(my_df[max_index, 1])/total * 100, "% of all bookings)\n")
  cat(row.names(my_df[min_index,]), "is the least used airline", "(",
      as.numeric(my_df[min_index, 1])/total * 100, "% of all bookings)\n")
  # Sort the data to get position references
  # Determine where my_airline stands in terms of popularity
  sorted_df <- my_df[order(my_df[, 1], decreasing = TRUE),]
  print(sorted_df)
  my_airline_index <- which(sorted_df$`Airline Names` == my_airline)
  cat(row.names(sorted_df[my_airline_index, ]), "is at #",
my_airline_index, "(",
```

```
                as.numeric(sorted_df[my_airline_index, 1])/total * 100, "%
of all bookings)\n")
  return(sorted_df)
}

get_satisfaction_data <- function(my_data, my_airline){
  airline_names <- unique(my_data$`Airline Name`)
  sat_score <- character()
  for (name in airline_names) {
    sat_score <- c(sat_score, mean(my_data$`Satisfaction`[my_data$`Airline
Name` == name],
                                                  na.rm = TRUE))
  }
  # Make a data-frame with Airline-name & mean of satisfaction scores
  dfmeansatscores <- data.frame(airline_names, as.numeric(sat_score),
                                stringsAsFactors = FALSE)
  sorted_df <- dfmeansatscores[order(dfmeansatscores[, 2],
                                                    decreasing = TRUE), ]
  print(sorted_df)
  max_index = which.max(sorted_df[,2])
  min_index = which.min(sorted_df[,2])
  cat(sorted_df[max_index, 1], "has highest average CSAT(",
      sorted_df[max_index, 2], " )\n")
  cat(sorted_df[min_index, 1], "has lowest average CSAT(",
      sorted_df[min_index, 2], " )\n")
  my_airline_index <- which(sorted_df$`airline_names` == my_airline)
  cat(my_airline, "is at #", my_airline_index, "(average CSAT",
      sorted_df[my_airline_index, 2], " )\n")
  # Conclusion at this point is:
  #  - There is no correlation between SAT-score and most-flown Airline
  #  - SouthEast Airlines Co. is at 6th position (off 14) in terms of CSAT
  #  - Figure why passengers are flocking Cheapseats Airlines Inc. (12 off
14)

  # Let's plot histograms for each Airline wrt to Satisfaction survey
  for (name in airline_names) {
    sat_score <- my_data$`Satisfaction`[my_data$`Airline Name` == name]
    hist(sat_score, main=paste("CSAT - ", name), xlab="Satisfaction",
col="red")
  }
  return(sorted_df)
}

get_loyalty_membership_data <- function(my_data, my_airline){
  airline_names <- unique(my_data$`Airline Name`)
  loyalty_cards <- character()
  for (name in airline_names) {
    loyalty_cards <- c(loyalty_cards, sum(my_data$`No. of other Loyalty
Cards`[my_data$`Airline Name` == name],
                                    na.rm = TRUE))
  }
  # Make a data-frame with Airline-name & loyalty-cards passengers used
  dfsumloyaltycards <- data.frame(airline_names,
as.numeric(loyalty_cards),
```

```
                                    stringsAsFactors = FALSE)
  sorted_df <- dfsumloyaltycards[order(dfsumloyaltycards[,2], decreasing =
TRUE), ]
  print(sorted_df)
  max_index = which.max(sorted_df[,2])
  min_index = which.min(sorted_df[,2])
  cat(sorted_df[max_index, 1], "has highest loyalty-cards offered/used (",
      sorted_df[max_index, 2], " )\n")
  cat(sorted_df[min_index, 1], "has lowest loyalty-cards offered/used (",
      sorted_df[min_index, 2], " )\n")
  my_airline_index <- which(sorted_df$`airline_names` == my_airline)
  cat(my_airline, "is at #", my_airline_index, "(loyalty-cards
offered/used",
      sorted_df[my_airline_index, 2], " )\n")
  return(sorted_df)
}

# Main
localxlpath <- '/Users/ssharat/Downloads/FinalProjectMaterial/Satisfaction
Survey(2).xlsx'
df <- readfromxl(localxlpath)
consulting_airline <- "Southeast Airlines Co."
get_best_worst_delays(df, consulting_airline)
# Get most popular airline
pop_score <- get_popular_carrier(df, consulting_airline)
colnames(pop_score) <- c('Bookings', 'AirlineName')
gg_book <- ggplot(data=pop_score, aes(x=AirlineName, y=Bookings))
gg_book <- gg_book + geom_bar(stat="identity", fill='red')
gg_book <- gg_book + coord_flip()
gg_book

# Get data for mean 'Satisfaction' per Airline (sorted)
sat_score <- get_satisfaction_data(df, consulting_airline)
colnames(sat_score) <- c('AirlineName', 'Satisfaction')
gg_sat <- ggplot(data=sat_score, aes(x=AirlineName, y=Satisfaction))
gg_sat <- gg_sat + geom_bar(stat="identity", fill='red')
gg_sat <- gg_sat + coord_flip()
gg_sat

# Get loyalty data per Airline
loy_score <- get_loyalty_membership_data(df, consulting_airline)
colnames(loy_score) <- c('AirlineName', 'LoyaltyCards')
gg_loy <- ggplot(data=loy_score, aes(x=AirlineName, y=LoyaltyCards))
gg_loy <- gg_loy + geom_bar(stat="identity", fill='red')
gg_loy <- gg_loy + coord_flip()
gg_loy

# Get a combined plot of the above
grid.arrange(gg_book, gg_sat, gg_loy, nrow=2)


# Digging deeper: Using linear regression/correlation analysis to
determine
library("gdata")
```

```
library(dplyr)
df_recode <- na.omit(df)
df_recode <- df_recode %>% mutate(`Airline Name`=recode(`Airline Name`,
                                                        "EnjoyFlying Air
Services" = 1,
                                                        "FlyFast Airways Inc." =
2,
                                                        "FlyHere Airways" = 3,
                                                        "FlyToSun Airlines Inc."
= 4,
                                                        "GoingNorth Airlines
Inc." = 5,
                                                        "West Airways Inc." = 6,
                                                        "OnlyJets Airlines Inc."
= 7,
                                                        "Northwest Business
Airlines Inc." = 8,
                                                        "Oursin Airlines Inc." =
9,
                                                        "Paul Smith Airlines
Inc." = 10,
                                                        "Cheapseats Airlines
Inc." = 11,
                                                        "Sigma Airlines Inc." =
12,
                                                        "Southeast Airlines Co."
= 13,
                                                        "Cool&Young Airlines
Inc." = 14))
df_recode <- df_recode %>% mutate(`Type of Travel`=recode(`Type of
Travel`,
                                                          "Business travel" = 1,
                                                          "Personal Travel" = 2,
                                                          "Mileage tickets" = 3))
df_recode <- df_recode %>% mutate(`Airline Status`=recode(`Airline
Status`,
                                                                 "Platinum" = 1,
                                                                 "Gold" = 2,
                                                                 "Silver" = 3,
                                                                 "Blue" = 4))

# Create a new data-frame with some important columns
df_models <- data.frame(df_recode$`Airline Name`,
                        df_recode$`Satisfaction`,
                        df_recode$`No. of other Loyalty Cards`,
                        df_recode$`Departure Delay in Minutes`,
                        df_recode$`Arrival Delay in Minutes`,
                        df_recode$`Price Sensitivity`,
                        df_recode$`Type of Travel`,
                        df_recode$`Airline Status`)

colnames(df_models) <- c("Airline Name",
                         "Satisfaction",
                         "No. of other Loyalty Cards",
```

```
                          "Departure Delay in Minutes",
                          "Arrival Delay in Minutes",
                          "Price Sensitivity",
                          "Type of Travel",
                          "Airline Status")
str(df_models)
lm_model <- lm(formula=`Satisfaction`~`Airline Status`, data=df_models)
summary(lm_model)

lm_model <- lm(formula=`Satisfaction`~`Age`, data=df_models)
summary(lm_model)

lm_model <- lm(formula=`Airline Name`~`No. of other Loyalty Cards`,
data=df_models)
summary(lm_model)

plot(df_models$`No. of other Loyalty Cards`, df_models$`Airline Name`,
pch=16, col='red',
     ylab="Airline Name",
     xlab="No. of other Loyalty Cards")
# abline(h = 0, v = 0, col = "gray60")
lm_model <- lm(formula=`Airline Name`~., data=df_models)
step(lm_model, data=df_models, direction="backward")
# By running the step function or Parsimonious Model we have:
# Step:  AIC=344145
#`Airline Name` ~ `No. of other Loyalty Cards` + `Departure Delay in
Minutes` +
#  `Arrival Delay in Minutes`
summary(lm_model)

# Predictions using linear model
randindex <- sample(1:dim(df_models)[1])
cutpoint2_3 <- floor(2 * length(randindex) /3)
trainData <- df_models[randindex[1:cutpoint2_3],]
testData <- df_models[randindex[(cutpoint2_3 + 1):length(randindex)],]

lmoutput <- lm(formula=`Airline Name`~., data=trainData)
test <- data.frame(testData$`Airline Name`,
                   testData$`Satisfaction`,
                   testData$`No. of other Loyalty Cards`,
                   testData$`Departure Delay in Minutes`,
                   testData$`Arrival Delay in Minutes`,
                   testData$`Price Sensitivity`,
                   testData$`Type of Travel`,
                   testData$`Airline Status`
                   )
colnames(test) <- c("Airline Name", "No. of other Loyalty Cards",
                    "Satisfaction",
                    "Departure Delay in Minutes",
                    "Arrival Delay in Minutes",
                    "Price Sensitivity",
                    "Type of Travel",
                    "Airline Status")
```

```
lmpredict <- round(predict(lmoutput, test, type="response"))
lm_compTable <- data.frame(testData[,1], lmpredict)
colnames(lm_compTable) <- c('Test', 'Pred')

percentage_lm <- length(which(lm_compTable$Test ==
lm_compTable$Pred))/dim(lm_compTable)[1]
#11.44%

# Based on AIC:
lmoutput <- lm(formula=`Airline Name`~ `No. of other Loyalty Cards` +
                 `Departure Delay in Minutes` +
                 `Arrival Delay in Minutes`, data=trainData)
test <- data.frame(testData$`Airline Name`,
                   testData$`No. of other Loyalty Cards`,
                   testData$`Departure Delay in Minutes`,
                   testData$`Arrival Delay in Minutes`)
colnames(test) <- c("Airline Name", "No. of other Loyalty Cards",
                    "Departure Delay in Minutes",
                    "Arrival Delay in Minutes")

lmpredict <- round(predict(lmoutput, test, type="response"))
lm_compTable <- data.frame(testData[,1], lmpredict)
colnames(lm_compTable) <- c('Test', 'Pred')

percentage_lm <- length(which(lm_compTable$Test ==
lm_compTable$Pred))/dim(lm_compTable)[1]

# Calculate the root mean square error(RMSE)
sqrt(mean((lm_compTable$Test - lm_compTable$Pred) ^ 2))
# [1] 3.879554

# Predictions using ksvm
library(kernlab)
cutpoint2_3 <- floor((2 * length(randindex) /3))
trainData <- df_models[randindex[1:cutpoint2_3],]
testData <- df_models[randindex[(cutpoint2_3 + 1):length(randindex)],]

ksvmoutput <- ksvm(`Airline Name`~., data=trainData,
                   kernel="rbfdot", #kernel function that projects the low
dimensional problem into higher dimensional space
                   kpar="automatic", #params used to control radial
function kernel(rbfdot)
                   C=10, #C -> cost of constraints
                   cross=10, #use 10 fold cross-validation in this model
                   prob.model=TRUE)
#test <- data.frame(testData$`Airline Name`,
#                    testData$`No. of other Loyalty Cards`
#)
#colnames(test) <- c("Airline Name", "No. of other Loyalty Cards")
test <- data.frame(testData$`Airline Name`,
                   testData$`Satisfaction`,
                   testData$`No. of other Loyalty Cards`,
                   testData$`Departure Delay in Minutes`,
                   testData$`Arrival Delay in Minutes`,
```

```
                    testData$`Price Sensitivity`,
                    testData$`Type of Travel`,
                    testData$`Airline Status`
)
colnames(test) <- c("Airline Name", "No. of other Loyalty Cards",
                    "Satisfaction",
                    "Departure Delay in Minutes",
                    "Arrival Delay in Minutes",
                    "Price Sensitivity",
                    "Type of Travel",
                    "Airline Status")

ksvmpredict <- round(predict(ksvmoutput, test, type="response"))
ksvm_compTable <- data.frame(testData[,1], ksvmpredict)
colnames(ksvm_compTable) <- c('Test', 'Pred')

percentage_ksvm <- length(which(ksvm_compTable$Test ==
ksvm_compTable$Pred))/dim(ksvm_compTable)[1]
percentage_ksvm
# Calculate the root mean square error(RMSE)
sqrt(mean((ksvm_compTable$Test - ksvm_compTable$Pred) ^ 2))


# Predictions using svm
library(e1071)
svmoutput <- svm(`Airline Name`~., data=trainData,
                 kernel="linear", #kernel function that projects the low
dimensional problem into higher dimensional space
                 cross=10, #use 10 fold cross-validation in this model
                 scale=FALSE)

svmpredict <- round(predict(svmoutput, test, type="response"))
svm_compTable <- data.frame(testData[,1], svmpredict)
colnames(svm_compTable) <- c('Test', 'Pred')

percentage_svm <- length(which(svm_compTable$Test ==
svm_compTable$Pred))/dim(svm_compTable)[1]


# Making predictions after bumping up the bookings by 10%, 50% and 100%
# Can potentially make a function of this - ran it manually for 10%, 50%
and 100%
df_models_order <- df_models[order(df_models$`Airline Name`),]
rownames(df_models_order) <- NULL
df_models_my_airline <- df_models_order[df_models_order$`Airline Name` ==
13,]
start_row <- as.numeric(rownames(df_models_my_airline)[1])
end_row <- start_row + nrow(df_models_my_airline) - 1
df_models_order[125867,]
df_models_order[125868,]
for (i in c(start_row:end_row)){
  if (df_models_order[i,3] == 0){
    df_models_order[i,3] <- 1
  } else {
```

```
    df_models_order[i,3] <- ceiling(df_models_order[i,3] * 2.0)
  }
}

# Running the KSVM prediction for a small subset since
# KVSM is compute intensive and taking several hours to give a result
randindex <- sample(1:dim(df_models_order)[1])
cutpoint2_3 <- floor((2 * length(randindex) /3) * 0.1)
trainData <- df_models_order[randindex[1:cutpoint2_3],]
testData <- df_models_order[randindex[(cutpoint2_3 + 1):1000],]

ksvmoutput <- ksvm(`Airline Name`~., data=trainData,
                   kernel="rbfdot", #kernel function that projects the low
dimensional problem into higher dimensional space
                   kpar="automatic", #params used to control radial
function kernel(rbfdot)
                   C=10, #C -> cost of constraints
                   cross=10, #use 10 fold cross-validation in this model
                   prob.model=TRUE)

test <- data.frame(testData$`Airline Name`,
                   testData$`Satisfaction`,
                   testData$`No. of other Loyalty Cards`,
                   testData$`Departure Delay in Minutes`,
                   testData$`Arrival Delay in Minutes`,
                   testData$`Price Sensitivity`,
                   testData$`Type of Travel`,
                   testData$`Airline Status`
)
colnames(test) <- c("Airline Name", "No. of other Loyalty Cards",
                    "Satisfaction",
                    "Departure Delay in Minutes",
                    "Arrival Delay in Minutes",
                    "Price Sensitivity",
                    "Type of Travel",
                    "Airline Status")

ksvmpredict <- round(predict(ksvmoutput, test, type="response"))
ksvm_compTable <- data.frame(testData[,1], ksvmpredict)
colnames(ksvm_compTable) <- c('Test', 'Pred')
total_my_airline <- count(ksvm_compTable[ksvm_compTable$Pred == 13,])
total_my_airline/nrow(ksvm_compTable) * 100

# Results from above test (also in Project document)
stage_results <- cbind(294, 889, 414)
barplot(stage_results, names.arg = c("10%","50%","100%"), main = "Bookings
with increased loyalty cards")
```