R Project: Airline Satisfaction

4375 Machine Learning with Dr. Mazidi

David Allen

March 24, 2022

This is a classification dataset for determining whether or not a customer was satisfied with a particular airline given their flight arrangements. The name Invistico Airline is a pseudonym to protect the privacy of the actual airline.

*Number of Rows: 130k

Source: https://www.kaggle.com/datasets/sjleshrac/airlines-customer-satisfaction

The three algorithms we will use are Naive Bayes, kNN, and Logistic Regression.

Load the Data

```
df <- read.csv("data/Invistico_Airline.csv")
str(df)
```

```
## 'data.frame':    129880 obs. of  23 variables:
##  $ satisfaction                     : chr  "satisfied" "satisfied" "satisfied" "satisfied" ...
##  $ Gender                           : chr  "Female" "Male" "Female" "Female" ...
##  $ Customer.Type                    : chr  "Loyal Customer" "Loyal Customer" "Loyal Customer" "Loyal
##  $ Age                              : int  65 47 15 60 70 30 66 10 56 22 ...
##  $ Type.of.Travel                   : chr  "Personal Travel" "Personal Travel" "Personal Travel" "Pe
##  $ Class                            : chr  "Eco" "Business" "Eco" "Eco" ...
##  $ Flight.Distance                  : int  265 2464 2138 623 354 1894 227 1812 73 1556 ...
##  $ Seat.comfort                     : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ Departure.Arrival.time.convenient: int  0 0 0 0 0 0 0 0 0 0 ...
##  $ Food.and.drink                   : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ Gate.location                    : int  2 3 3 3 3 3 3 3 3 3 ...
##  $ Inflight.wifi.service            : int  2 0 2 3 4 2 2 2 5 2 ...
##  $ Inflight.entertainment           : int  4 2 0 4 3 0 5 0 3 0 ...
##  $ Online.support                   : int  2 2 2 3 4 2 5 2 5 2 ...
##  $ Ease.of.Online.booking           : int  3 3 2 1 2 2 5 2 4 2 ...
##  $ On.board.service                 : int  3 4 3 1 2 5 5 3 4 2 ...
##  $ Leg.room.service                 : int  0 4 3 0 0 4 0 3 0 4 ...
##  $ Baggage.handling                 : int  3 4 4 1 2 5 5 4 1 5 ...
##  $ Checkin.service                  : int  5 2 4 4 4 5 5 5 5 3 ...
##  $ Cleanliness                      : int  3 3 4 1 2 4 5 4 4 4 ...
##  $ Online.boarding                  : int  2 2 2 3 5 2 3 2 4 2 ...
##  $ Departure.Delay.in.Minutes       : int  0 310 0 0 0 0 17 0 0 30 ...
##  $ Arrival.Delay.in.Minutes         : int  0 305 0 0 0 0 15 0 0 26 ...
```

As evident in the dataset summary, this is a very large collection of data: 130,000 observations of 23 features.

Data Exploration and Cleaning

Cleaning Steps

- Convert character fields to factors (satisfaction, class, travel reason, etc.)
- Handle null values
- Handle redundant features

We start by converting all the applicable fields into factors (ex. Male/Female, and our target variable Satisfied/Dissatisfied)

```
c_factors <- c(1, 2, 3, 5, 6)

df[,c_factors] <- lapply(df[,c_factors], factor)
str(df)
```

```
## 'data.frame':    129880 obs. of  23 variables:
##  $ satisfaction                     : Factor w/ 2 levels "dissatisfied",..: 2 2 2 2 2 2 2 2 2 2 ...
##  $ Gender                           : Factor w/ 2 levels "Female","Male": 1 2 1 1 1 2 1 2 1 2 ...
##  $ Customer.Type                    : Factor w/ 2 levels "disloyal Customer",..: 2 2 2 2 2 2 2 2 2 2
##  $ Age                              : int  65 47 15 60 70 30 66 10 56 22 ...
##  $ Type.of.Travel                   : Factor w/ 2 levels "Business travel",..: 2 2 2 2 2 2 2 2 2 2 .
##  $ Class                            : Factor w/ 3 levels "Business","Eco",..: 2 1 2 2 2 2 2 2 1 2 ..
##  $ Flight.Distance                  : int  265 2464 2138 623 354 1894 227 1812 73 1556 ...
##  $ Seat.comfort                     : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ Departure.Arrival.time.convenient: int  0 0 0 0 0 0 0 0 0 0 ...
##  $ Food.and.drink                   : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ Gate.location                    : int  2 3 3 3 3 3 3 3 3 3 ...
##  $ Inflight.wifi.service            : int  2 0 2 3 4 2 2 2 5 2 ...
##  $ Inflight.entertainment           : int  4 2 0 4 3 0 5 0 3 0 ...
##  $ Online.support                   : int  2 2 2 3 4 2 5 2 5 2 ...
##  $ Ease.of.Online.booking           : int  3 3 2 1 2 2 5 2 4 2 ...
##  $ On.board.service                 : int  3 4 3 1 2 5 5 3 4 2 ...
##  $ Leg.room.service                 : int  0 4 3 0 0 4 0 3 0 4 ...
##  $ Baggage.handling                 : int  3 4 4 1 2 5 5 4 1 5 ...
##  $ Checkin.service                  : int  5 2 4 4 4 5 5 5 5 3 ...
##  $ Cleanliness                      : int  3 3 4 1 2 4 5 4 4 4 ...
##  $ Online.boarding                  : int  2 2 2 3 5 2 3 2 4 2 ...
##  $ Departure.Delay.in.Minutes       : int  0 310 0 0 0 0 17 0 0 30 ...
##  $ Arrival.Delay.in.Minutes         : int  0 305 0 0 0 0 15 0 0 26 ...
```

```
summary(df$satisfaction)
```

```
## dissatisfied    satisfied
##        58793        71087
```

```
summary(df$Gender)
```

```
## Female   Male
##  65899  63981
```

It seems there is a relatively balanced split between men and women as well as satisfied customers and dissatisfied.

```
sum(is.na(df))
```

```
## [1] 393
```

```
colSums(sapply(df, is.na))
```

```
##                      satisfaction                            Gender
##                                 0                                 0
##                     Customer.Type                               Age
##                                 0                                 0
##                    Type.of.Travel                             Class
##                                 0                                 0
##                   Flight.Distance                      Seat.comfort
##                                 0                                 0
## Departure.Arrival.time.convenient                    Food.and.drink
##                                 0                                 0
##                     Gate.location             Inflight.wifi.service
##                                 0                                 0
##            Inflight.entertainment                    Online.support
##                                 0                                 0
##            Ease.of.Online.booking                  On.board.service
##                                 0                                 0
##                  Leg.room.service                  Baggage.handling
##                                 0                                 0
##                  Checkin.service                       Cleanliness
##                                 0                                 0
##                   Online.boarding          Departure.Delay.in.Minutes
##                                 0                                 0
##          Arrival.Delay.in.Minutes
##                               393
```

For some reason there are null values in the Arrival Delay column. Let's just pretend this means there was no delay.

```
df$Arrival.Delay.in.Minutes[is.na(df$Arrival.Delay.in.Minutes)] <- df$Departure.Delay.in.Minutes[is.na(
sum(is.na(df))
```

```
## [1] 0
```

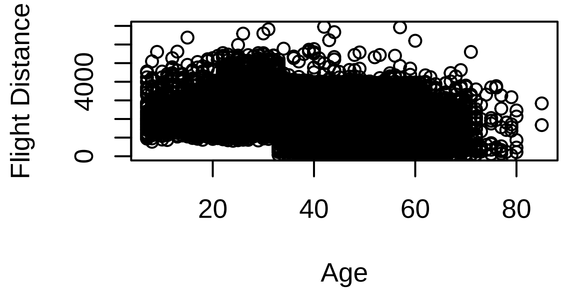
To visualize the data we'll take a random 10% sample of the 130k observations, so we can do it a little bit faster.

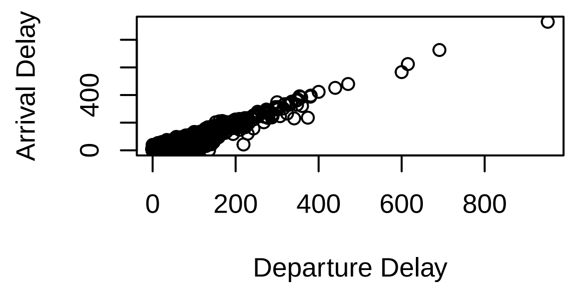
```
set.seed(1234)
i <- sample(1:nrow(df), nrow(df)*0.1, replace=FALSE)
df_sample <- df[i,]
```

```r
summary(df_sample[-c_factors])
```

```
##       Age        Flight.Distance  Seat.comfort
##  Min.   : 7.00   Min.   :  50    Min.   :0.000
##  1st Qu.:27.00   1st Qu.:1353    1st Qu.:2.000
##  Median :40.00   Median :1933    Median :3.000
##  Mean   :39.43   Mean   :1982    Mean   :2.849
##  3rd Qu.:51.00   3rd Qu.:2548    3rd Qu.:4.000
##  Max.   :85.00   Max.   :6951    Max.   :5.000
##  Departure.Arrival.time.convenient Food.and.drink  Gate.location
##  Min.   :0.000                     Min.   :0.000   Min.   :1.000
##  1st Qu.:2.000                     1st Qu.:2.000   1st Qu.:2.000
##  Median :3.000                     Median :3.000   Median :3.000
##  Mean   :2.983                     Mean   :2.855   Mean   :2.998
##  3rd Qu.:4.000                     3rd Qu.:4.000   3rd Qu.:4.000
##  Max.   :5.000                     Max.   :5.000   Max.   :5.000
##  Inflight.wifi.service Inflight.entertainment Online.support
##  Min.   :0.000         Min.   :0.000          Min.   :1.000
##  1st Qu.:2.000         1st Qu.:2.000          1st Qu.:3.000
##  Median :3.000         Median :4.000          Median :4.000
##  Mean   :3.236         Mean   :3.377          Mean   :3.508
##  3rd Qu.:4.000         3rd Qu.:4.000          3rd Qu.:5.000
##  Max.   :5.000         Max.   :5.000          Max.   :5.000
##  Ease.of.Online.booking On.board.service Leg.room.service Baggage.handling
##  Min.   :1.000          Min.   :1.000    Min.   :0.000    Min.   :1.000
##  1st Qu.:2.000          1st Qu.:3.000    1st Qu.:2.000    1st Qu.:3.000
##  Median :4.000          Median :4.000    Median :4.000    Median :4.000
##  Mean   :3.471          Mean   :3.465    Mean   :3.487    Mean   :3.705
##  3rd Qu.:5.000          3rd Qu.:4.000    3rd Qu.:5.000    3rd Qu.:5.000
##  Max.   :5.000          Max.   :5.000    Max.   :5.000    Max.   :5.000
##  Checkin.service  Cleanliness    Online.boarding Departure.Delay.in.Minutes
##  Min.   :1.000   Min.   :1.000   Min.   :1.00    Min.   :  0.00
##  1st Qu.:3.000   1st Qu.:3.000   1st Qu.:2.00    1st Qu.:  0.00
##  Median :3.000   Median :4.000   Median :3.00    Median :  0.00
##  Mean   :3.346   Mean   :3.707   Mean   :3.34    Mean   : 14.59
##  3rd Qu.:4.000   3rd Qu.:5.000   3rd Qu.:4.00    3rd Qu.: 12.00
##  Max.   :5.000   Max.   :5.000   Max.   :5.00    Max.   :930.00
##  Arrival.Delay.in.Minutes
##  Min.   :  0.00
##  1st Qu.:  0.00
##  Median :  0.00
##  Mean   : 14.97
##  3rd Qu.: 13.00
##  Max.   :952.00
```

```r
par(mfrow=c(2, 2))
plot(df_sample$Age~df_sample$satisfaction, ylab="Age", xlab="Satisfaction")
plot(df_sample$Gender~df_sample$satisfaction, ylab="Gender", xlab="Satisfaction")
plot(df_sample$Flight.Distance~df_sample$Age, ylab="Flight Distance", xlab="Age")
plot(df_sample$Departure.Delay.in.Minutes~df_sample$Arrival.Delay.in.Minutes, ylab="Arrival Delay", xla
```





No strong conclusions can be made about the relationship between age and satisfaction, but it seems women tend to be more satisfied with their travels then men are. Age and flight distance show no correlation, but there seems to be a weird vertical shift in the body of the graph at around 30 years old. The shift is very jarring, making it appear unnatural. Why does no one under 30 fly very short distances? And why do those over 30 tend to not fly above 4000 miles all of a sudden? We will not go in-depth for this anomaly in the data, but it is interesting to note. As expected, departure delay and arrival delay are very highly correlated. For that reason, we will make a new column specifically for whether or not there was a delay at all:

```
df$Delay <- FALSE
df$Delay[df$Departure.Delay.in.Minutes > 0] <- TRUE
df$Delay[df$Arrival.Delay.in.Minutes > 0] <- TRUE
df$Delay <- as.factor(df$Delay)
append(c_factors, 24)
```

```
## [1]  1  2  3  5  6 24
```

```
names(df)
```

```
##  [1] "satisfaction"                      "Gender"
##  [3] "Customer.Type"                     "Age"
##  [5] "Type.of.Travel"                    "Class"
##  [7] "Flight.Distance"                   "Seat.comfort"
##  [9] "Departure.Arrival.time.convenient" "Food.and.drink"
## [11] "Gate.location"                     "Inflight.wifi.service"
## [13] "Inflight.entertainment"            "Online.support"
## [15] "Ease.of.Online.booking"            "On.board.service"
```

```
## [17] "Leg.room.service"                  "Baggage.handling"
## [19] "Checkin.service"                   "Cleanliness"
## [21] "Online.boarding"                   "Departure.Delay.in.Minutes"
## [23] "Arrival.Delay.in.Minutes"          "Delay"
```

```
summary(df$Delay)
```

```
## FALSE  TRUE
## 59498 70382
```

Train and Test

Let's narrow down our features. Seat comfort and leg room shouldn't be dealbreakers for flight satisfaction. Either way, food and drink is pretty correlated with overall comfort and generally more impactful, so we'll keep that feature only. Accuracy can be a good metric for this data since the set is pretty balanced.

```
predictors <- c(2, 3, 4, 5, 6, 7, 10, 11, 12, 13, 15, 16, 18, 19, 20, 24)
i <- sample(1:nrow(df), nrow(df)*0.75, replace=FALSE)
train <- df[i,c(1, predictors)]
test <- df[-i,c(1, predictors)]
```

Naive Bayes

We'll try Naive Bayes first. It has the advantage of calculating likelihoods for all our predictors.

```
library(e1071)
nb1 <- naiveBayes(satisfaction~., data=train)
nb1
```

```
##
## Naive Bayes Classifier for Discrete Predictors
##
## Call:
## naiveBayes.default(x = X, y = Y, laplace = laplace)
##
## A-priori probabilities:
## Y
## dissatisfied    satisfied
##    0.4522944    0.5477056
##
## Conditional probabilities:
##                 Gender
## Y                   Female      Male
##   dissatisfied 0.3900767 0.6099233
##   satisfied    0.6028640 0.3971360
##
##                 Customer.Type
## Y                disloyal Customer Loyal Customer
##   dissatisfied        0.30489355     0.69510645
##   satisfied           0.08037187     0.91962813
##
```

```
##              Age
## Y                 [,1]     [,2]
##   dissatisfied 37.55329 15.94277
##   satisfied    40.99775 14.25291
##
##              Type.of.Travel
## Y              Business travel Personal Travel
##   dissatisfied       0.6329838       0.3670162
##   satisfied          0.7361673       0.2638327
##
##              Class
## Y               Business        Eco   Eco Plus
##   dissatisfied 0.30918335 0.60068546 0.09013119
##   satisfied    0.61950817 0.32366172 0.05683011
##
##              Flight.Distance
## Y                 [,1]      [,2]
##   dissatisfied 2024.652  889.1576
##   satisfied    1944.902 1127.9700
##
##              Food.and.drink
## Y                 [,1]     [,2]
##   dissatisfied 2.660221 1.242628
##   satisfied    3.009953 1.573487
##
##              Gate.location
## Y                 [,1]     [,2]
##   dissatisfied 3.007422 1.217506
##   satisfied    2.976121 1.375742
##
##              Inflight.wifi.service
## Y                 [,1]     [,2]
##   dissatisfied 2.920832 1.345318
##   satisfied    3.522792 1.231093
##
##              Inflight.entertainment
## Y                 [,1]     [,2]
##   dissatisfied 2.609242 1.096339
##   satisfied    4.024985 1.185861
##
##              Ease.of.Online.booking
## Y                 [,1]     [,2]
##   dissatisfied 2.854192 1.302207
##   satisfied    3.986017 1.060109
##
##              On.board.service
## Y                 [,1]     [,2]
##   dissatisfied 2.973217 1.266733
##   satisfied    3.870577 1.120601
##
##              Baggage.handling
## Y                 [,1]     [,2]
##   dissatisfied 3.366494 1.141221
##   satisfied    3.970816 1.094883
```

```
##
##               Checkin.service
## Y                   [,1]     [,2]
##   dissatisfied 2.972831 1.277937
##   satisfied    3.647398 1.160402
##
##               Cleanliness
## Y                   [,1]     [,2]
##   dissatisfied 3.374643 1.145023
##   satisfied    3.980563 1.081736
##
##               Delay
## Y                  FALSE      TRUE
##   dissatisfied 0.4214445 0.5785555
##   satisfied    0.4891288 0.5108712
```

Those who flew Business class were more likely to be satisfied with their trip, which makes sense. Apparently, loyal customers are significantly more likely to be satisfied with their trip. Perhaps there could be deeper reasons for this? It's a little funny how out of those who were satisfied with their trip, more often than not they experienced a delay (although it is much more frequent among those who were unsatisfied).

```
pred1 <- predict(nb1, newdata = test, type="class")
table(pred1, test$satisfaction)
```

```
##
## pred1          dissatisfied satisfied
##   dissatisfied        11828      3288
##   satisfied            2907     14447
```

```
acc1 <- mean(pred1==test$satisfaction)
acc1
```

```
## [1] 0.8092085
```

kNN

All the data needs to be scaled for this algorithm to run well. Although kNN suffers the curse of dimensionality, hopefully narrowing down the predictors will lead to decent results. We will also keep the value of k low so the algorithm doesn't take too long to run.

```
library(class)
scaled_train <- scale(sapply(df[i,predictors], as.numeric))
scaled_test <- scale(sapply(df[-i,predictors], as.numeric))

pred2 <- knn(train = scaled_train, test=scaled_test, cl=train$satisfaction, k=17)
```

Playing around with the value of k, 17 produced the best results.

```
table(pred2, test$satisfaction)
```

```
##
## pred2          dissatisfied satisfied
##   dissatisfied        13307      1842
##   satisfied            1428     15893
```

```r
acc2 <- length(which(pred2 == test$satisfaction)) / length(pred2)
acc2
```

```
## [1] 0.8992917
```

Logistic Regression

```r
glm1 <- glm(satisfaction~., data=train, family="binomial")
summary(glm1)
```

```
##
## Call:
## glm(formula = satisfaction ~ ., family = "binomial", data = train)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -3.0047  -0.6074   0.1995   0.5474   3.4743
##
## Coefficients:
##                                Estimate Std. Error z value Pr(>|z|)
## (Intercept)                  -6.021e+00  7.133e-02 -84.410  < 2e-16 ***
## GenderMale                   -9.876e-01  1.858e-02 -53.167  < 2e-16 ***
## Customer.TypeLoyal Customer   1.834e+00  2.781e-02  65.940  < 2e-16 ***
## Age                          -8.615e-03  6.451e-04 -13.354  < 2e-16 ***
## Type.of.TravelPersonal Travel -8.637e-01  2.617e-02 -33.009  < 2e-16 ***
## ClassEco                     -6.641e-01  2.363e-02 -28.106  < 2e-16 ***
## ClassEco Plus                -7.025e-01  3.648e-02 -19.254  < 2e-16 ***
## Flight.Distance              -1.408e-04  9.662e-06 -14.577  < 2e-16 ***
## Food.and.drink               -1.377e-01  8.311e-03 -16.574  < 2e-16 ***
## Gate.location                 4.447e-02  8.162e-03   5.448 5.10e-08 ***
## Inflight.wifi.service        -4.845e-02  9.411e-03  -5.148 2.63e-07 ***
## Inflight.entertainment        7.991e-01  8.744e-03  91.378  < 2e-16 ***
## Ease.of.Online.booking        4.720e-01  1.081e-02  43.642  < 2e-16 ***
## On.board.service              2.994e-01  9.228e-03  32.440  < 2e-16 ***
## Baggage.handling              1.241e-01  1.044e-02  11.897  < 2e-16 ***
## Checkin.service               3.037e-01  7.705e-03  39.417  < 2e-16 ***
## Cleanliness                   9.612e-02  1.070e-02   8.984  < 2e-16 ***
## DelayTRUE                    -3.570e-01  1.818e-02 -19.636  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 134151  on 97409  degrees of freedom
## Residual deviance:  77563  on 97392  degrees of freedom
## AIC: 77599
```

```
##
## Number of Fisher Scoring iterations: 5
```

All the predictors are excellent for this problem.

```r
probs <- predict(glm1, newdata=test, type="response")
pred3 <- ifelse(probs>0.5, 2, 1)
table(pred3, test$satisfaction)
```

```
##
## pred3 dissatisfied satisfied
##     1        11987      2849
##     2         2748     14886
```

```r
acc3 <- mean(pred3==as.numeric(test$satisfaction))
acc3
```

```
## [1] 0.8276255
```

Analysis

Each algorithm performed pretty well in classifiying satisfied customers against dissatisfied.

Here is the ranking of the algorithms:

- Third Place: Naive Bayes
- Second Place: Logistic Regression
- First Place: kNN

The results are not too surprising. Naive Bayes was not expected to perform well on such a big dataset, especially since the features are harldy independent of each other. Despite that, an 80% accuracy is not bad. The kNN algorithm was the most worrisome. Our dataset has around 20 features and over 100,000 observations. However, it ultimately provided the best results by a decent margin. With even more fine tuning, of hyperparameters and of the data itself, we could probably get significantly better results.

It appears the algorithms were able to see the correlation between average to high survey scores and customer satisfaction. I would imagine a human could probably determine customer satisfaction better than any of these models, but given that the model is only wrong 10% of the time, it might be worth utilizing as a quick assesor of survey results.

If I were to further work on this problem, I would consider weighing the results to favor a better True-Positive rate (i.e. prefer false negatives over false positives), since dissatisfied customers are the only ones worth devoting extra attention to.