Similarity Classification

Joshua Durana

2022-10-09

Source : https://www.kaggle.com/datasets/teejmahal20/airline-passenger-satisfaction (https://www.kaggle.com/datasets/teejmahal20/airline-passenger-satisfaction)

This data set contains data from an airline satisfaction survey

Load and Clean Data

```
airplaneData <- read.csv("Data/airplaneData.csv", header = TRUE)

#Convert survey items to factors
cols <- c("Gender", "Customer.Type", "Class", "Inflight.wifi.service", "Departure.Arrival.tim
e.convenient", "Ease.of.Online.booking", "Food.and.drink", "Online.boarding", "Seat.comfort",
"Inflight.entertainment", "On.board.service", "Leg.room.service", "Baggage.handling", "Checki
n.service", "Inflight.service", "Cleanliness", "satisfaction")
airplaneData[cols] <- lapply(airplaneData[cols], as.factor)

#Drop X, ID, and Gate Location
airplaneData <- subset(airplaneData, select = -c(X, id, Gate.location))

#Obtain only numeric columns
numCol <- unlist(lapply(airplaneData, is.numeric))

#Convert arrival delay to int
airplaneData$Arrival.Delay.in.Minutes <- as.integer(airplaneData$Arrival.Delay.in.Minutes)
```

Split Data set

```
set.seed(10622)

i <- sample(1:nrow(airplaneData), .80*nrow(airplaneData), replace = FALSE)
planeTrain <- airplaneData[i,]
planeTest <- airplaneData[-i,]
```

Data Exploration Logistic Regression

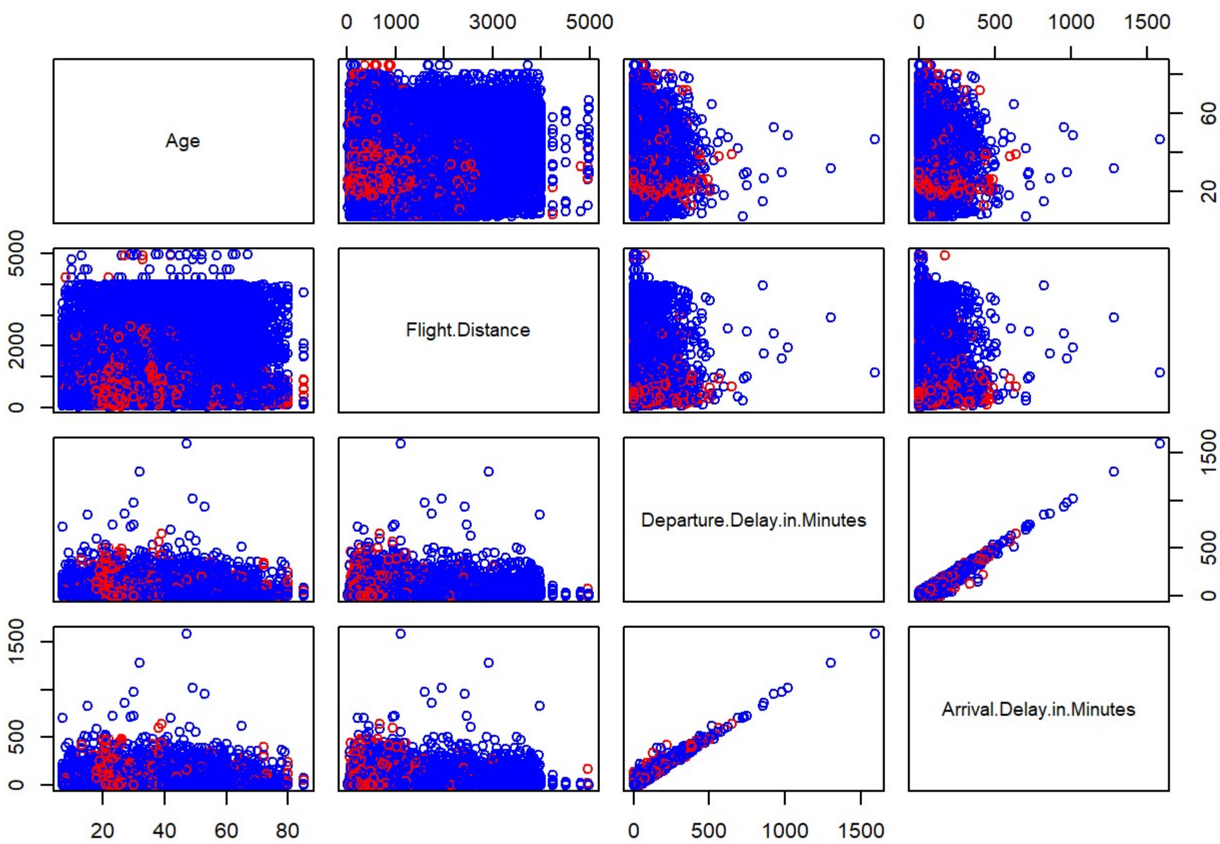
For this data set we’re trying to predict whether a customer is loyal or disloyal

```
summary(planeTrain)
```

```
##    Gender                Customer.Type        Age        Type.of.Travel
##  Female:42094   disloyal Customer:15202   Min.   : 7.00   Length:83123
##  Male  :41029   Loyal Customer   :67921   1st Qu.:27.00   Class :character
##                                           Median :40.00   Mode  :character
##                                           Mean   :39.37
##                                           3rd Qu.:51.00
##                                           Max.   :85.00
##
##       Class       Flight.Distance Inflight.wifi.service
##  Business:39699   Min.   :  31    0: 2493
##  Eco     :37436   1st Qu.: 413    1:14310
##  Eco Plus: 5988   Median : 842    2:20669
##                   Mean   :1189    3:20733
##                   3rd Qu.:1744    4:15788
##                   Max.   :4983    5: 9130
##
##  Departure.Arrival.time.convenient Ease.of.Online.booking Food.and.drink
##  0: 4252                           0: 3617                0:   91
##  1:12376                           1:14088                1:10289
##  2:13748                           2:19164                2:17642
##  3:14362                           3:19578                3:17781
##  4:20423                           4:15669                4:19485
##  5:17962                           5:11007                5:17835
##
##  Online.boarding Seat.comfort Inflight.entertainment On.board.service
##  0: 1940         0:    1      0:   12                0:    2
##  1: 8520         1: 9652      1: 9970                1: 9436
##  2:14012         2:11968      2:14178                2:11853
##  3:17438         3:14903      3:15280                3:18220
##  4:24572         4:25393      4:23503                4:24608
##  5:16641         5:21206      5:20180                5:19004
##
##  Leg.room.service Baggage.handling Checkin.service Inflight.service Cleanliness
##  0:  382          1: 5831          0:    1         0:    2          0:   11
##  1: 8308          2: 9247          1:10337         1: 5630          1:10630
##  2:15614          3:16502          2:10236         2: 9167          2:12922
##  3:16135          4:29843          3:22746         3:16279          3:19590
##  4:22976          5:21700          4:23297         4:30282          4:21807
##  5:19708                           5:16506         5:21763          5:18163
##
##  Departure.Delay.in.Minutes Arrival.Delay.in.Minutes
##  Min.   :   0.00            Min.   :   0.00
##  1st Qu.:   0.00            1st Qu.:   0.00
##  Median :   0.00            Median :   0.00
##  Mean   :  14.87            Mean   :  15.21
##  3rd Qu.:  12.00            3rd Qu.:  13.00
##  Max.   :1592.00            Max.   :1584.00
##                             NA's   :250
##                   satisfaction
##  neutral or dissatisfied:47111
##  satisfied              :36012
```

```
##
##
##
##
##
```

```
pairs(planeTrain[,numCol], col = c("red", "blue")[unclass(planeTrain$Customer.Type)])
```



While Departure Delay and Arrival Delay is linear, both factors seem to be well mixed. Flight Distance and Arrival Delay seems better since it's somewhat more linear and each factor seems to be better separated.

Logistic Regression

```
#Create Logistic Regression Model
custLr <- glm(Customer.Type~Flight.Distance + Departure.Delay.in.Minutes, data=planeTrain,fam
ily="binomial")

#Metrics
summary(custLr)
```

```
##
## Call:
## glm(formula = Customer.Type ~ Flight.Distance + Departure.Delay.in.Minutes,
##     family = "binomial", data = planeTrain)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -3.1467   0.2567   0.5539   0.7466   0.9215
##
## Coefficients:
##                              Estimate Std. Error z value Pr(>|z|)
## (Intercept)                 0.6814201  0.0146456  46.527   <2e-16 ***
## Flight.Distance             0.0008588  0.0000141  60.917   <2e-16 ***
## Departure.Delay.in.Minutes -0.0002772  0.0002399  -1.156    0.248
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 79090  on 83122  degrees of freedom
## Residual deviance: 73795  on 83120  degrees of freedom
## AIC: 73801
##
## Number of Fisher Scoring iterations: 5
```

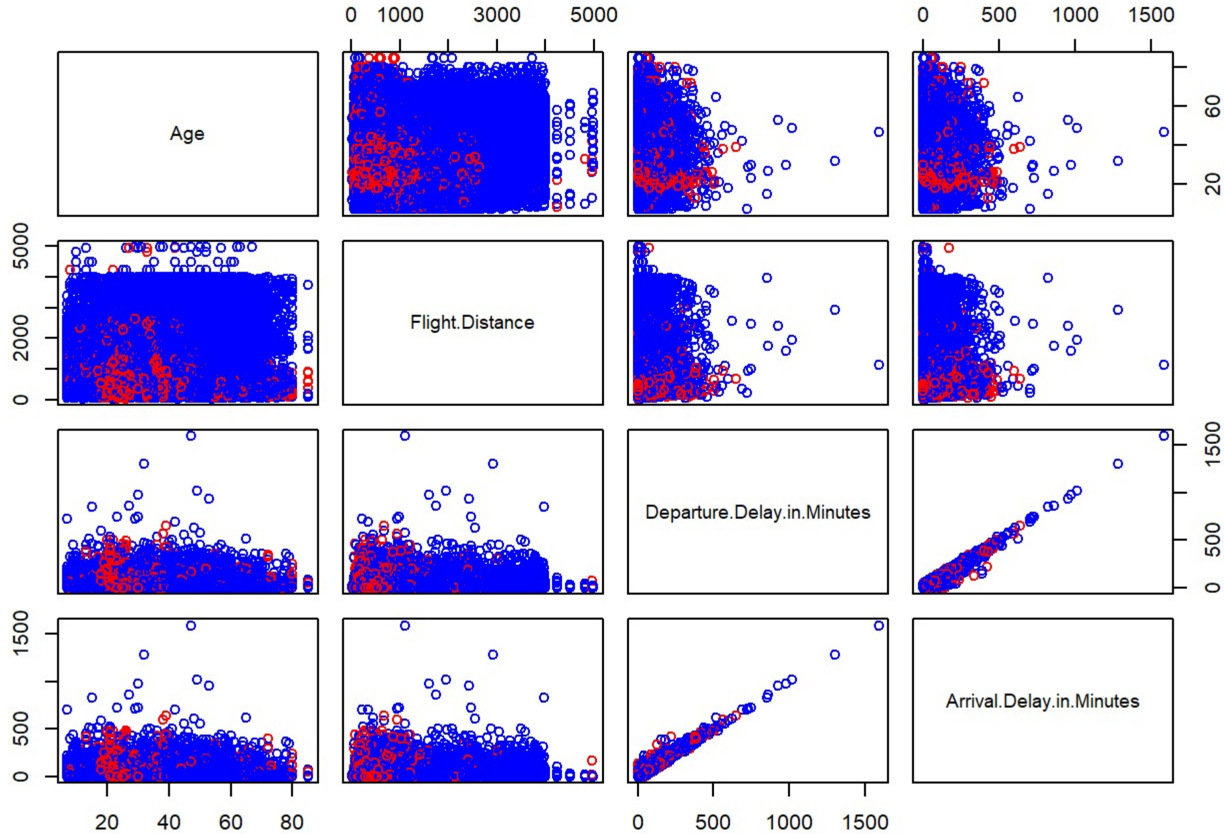
```
prob <- predict(custLr, newdata=planeTest, type="response")
pred <- ifelse(prob>.5,1,0)
acc <- mean(pred==as.integer(planeTest$Customer.Type))
print(paste("Accuracy = ", acc))
```

```
## [1] "Accuracy =  0.181848804196141"
```

Flight distance and departure delay are good factors, but the model isn't accurate. The while residual deviance is lower than null deviance, they're pretty close together in value.

Data Exploration KNN Classification

```
pairs(planeTrain[,numCol], col = c("red", "blue")[unclass(planeTrain$Customer.Type)])
```



The most likely numeric pairs to use for knn seems to be flight distance with arrival delay or departure delay because the 2 customer types seems to be more clustered together

Data KNN Classification

```
library(class)

#Get unlabeled data and labels
unlabeled <- sample(2, nrow(airplaneData), replace=TRUE, prob=c(.8,.2))
uTrain <- airplaneData[unlabeled==1, c(6,20)]
uTest <- airplaneData[unlabeled==2, c(6,20)]
uTrainLabel <- airplaneData[unlabeled==1, 2]
uTestLabel <- airplaneData[unlabeled==2, 2]

#Scale data
uTrainScale <- scale(uTrain)
uTestScale <- scale(uTest)

#KNN
knnPlane <- knn(train=uTrain, test=uTest, cl=uTrainLabel, k=3)

#Obtain Accuracy
knnResults <- knnPlane == uTestLabel
acc <- length(which(knnResults == TRUE)) / length(knnResults)
acc
```

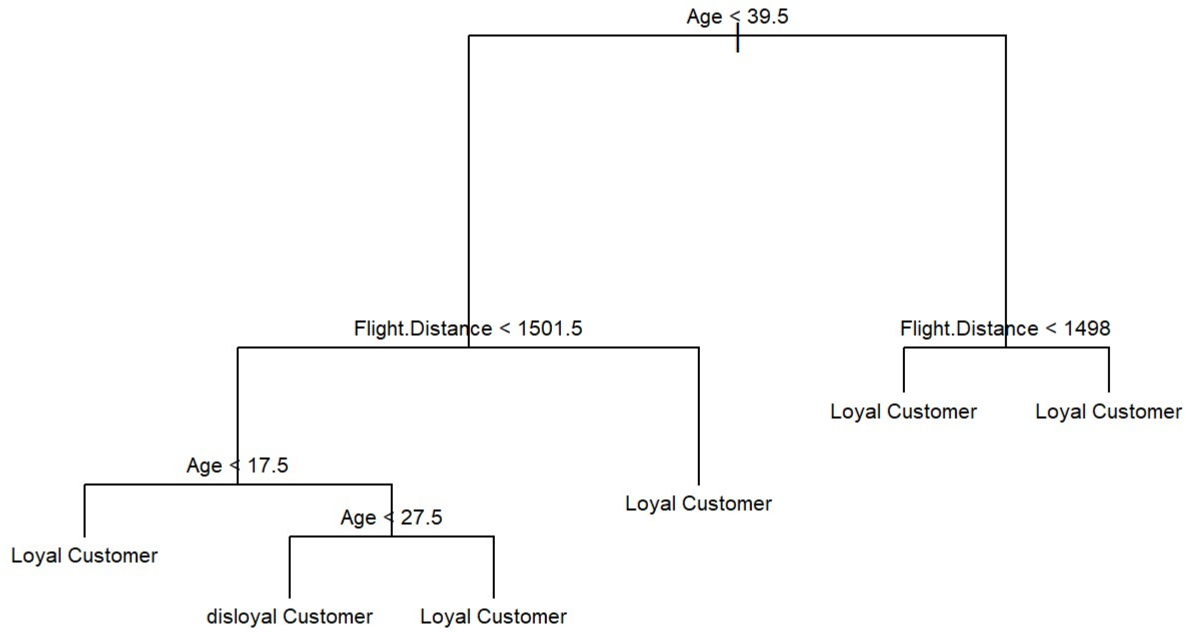
```
## [1] 0.7927937
```

This model is pretty accurate, mainly because the different factors seem to be well separated as shown in the pairs graph

Decision Tree

```
library(tree)

#Making Decision Tree
planeTree <- tree(Customer.Type~Age + Flight.Distance + Departure.Delay.in.Minutes + Arrival.
Delay.in.Minutes, data=planeTrain)
plot(planeTree)
text(planeTree, cex = .65, pretty = 1)
```

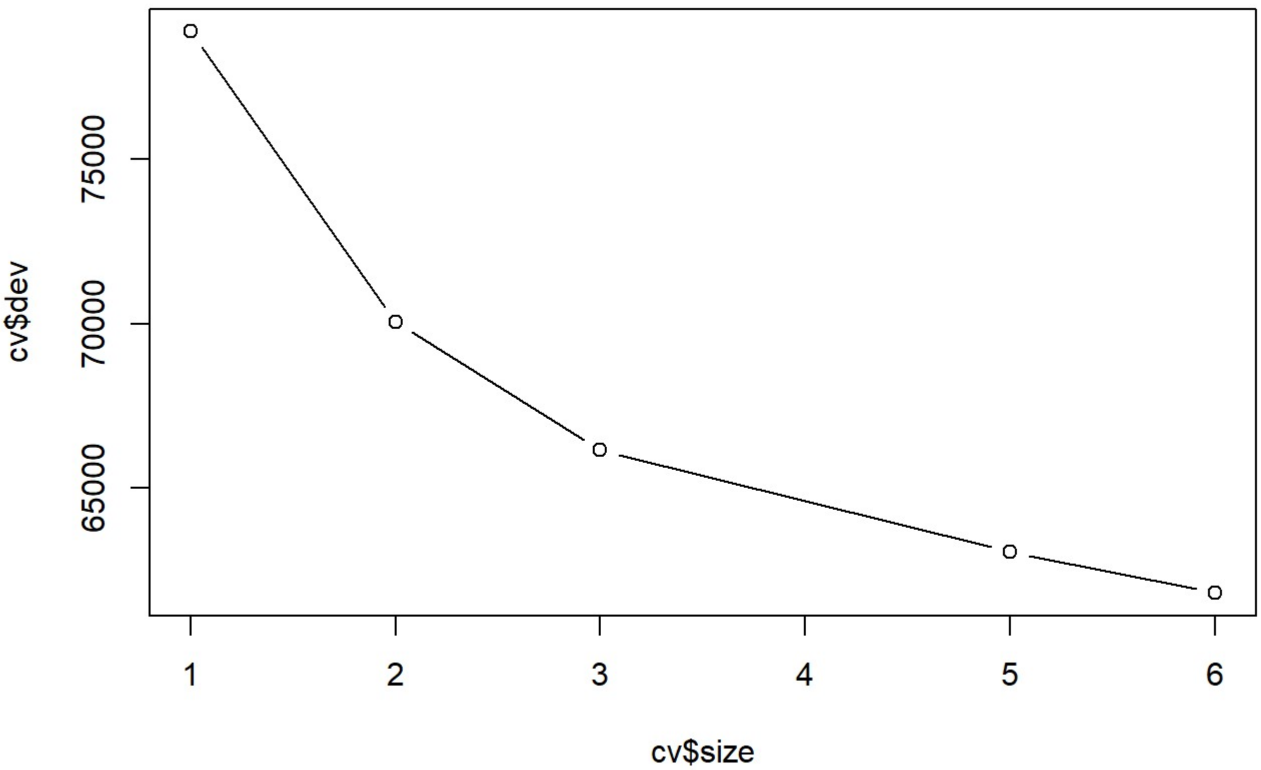



```
#Getting Accuracy
predTree <- predict(planeTree, newdata=planeTest, type="class")
mean(predTree == planeTest$Customer.Type)
```

```
## [1] 0.8322025
```

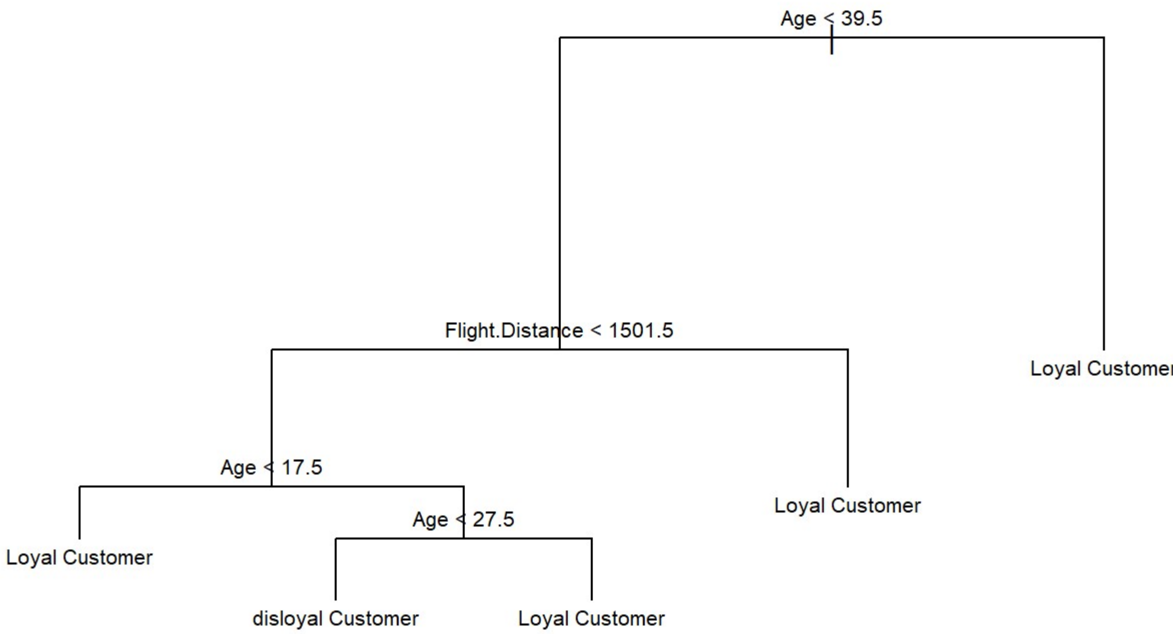
Let's see if pruning the tree would improve performance.

```
#Find the size to prune
cv <- cv.tree(planeTree)
plot(cv$size, cv$dev, type='b')
```




```
#Prune tree
prunedPlaneTree <- prune.tree(planeTree, best=5)

plot(prunedPlaneTree)
text(prunedPlaneTree, cex = .65, pretty = 1)
```




```
#Getting Accuracy
predPruneTree <- predict(prunedPlaneTree, newdata=planeTest, type="class")
mean(predPruneTree == planeTest$Customer.Type)
```

```
## [1] 0.8322025
```

This is a little more accurate than KNN, I think this is due to it's similarity of KNN. Both algorithms predict making different regions that contain each factor. While KNN uses an observation's nearest neighbor, decision trees split the training data into different regions. I think the small increase of accuracy is due to decision trees can have multiple regions of each factor, so it can get a pocket of a factor unlike KNN.