# Classification

Code ▾

## Introduction

Linear classification is acheived via logistic regression, which isn’t actually regression at all. Here the target is qualitative so we want to figure out what class out target falls into. This is great for creating decsion boundaries so that we can divide by our binary classes. A big assumption we make with logistic regression is that there is some linearity between our target and predictor, meaning that it can be somewhat limited in what it can do, seeing as we need an established linear relationship between target and predictor.

Today we’ll look at airline customer satsifaction with our predictor being flight distance. The dataset was found on Kaggle. Link (https://www.kaggle.com/datasets/sjleshrac/airlines-customer-satisfaction) For pdf viewers: https://www.kaggle.com/datasets/sjleshrac/airlines-customer-satisfaction (https://www.kaggle.com/datasets/sjleshrac/airlines-customer-satisfaction)

## Data Exploration

First we’ll need to divide the data into train and test.

Hide

```
df <- read.csv("airline.csv", header = TRUE)
str(df)
nrow(df)
sum(is.na(df$satisfaction))
sum(is.na(df$Flight.Distance))
```

For the sake of being concise, let’s factorize our satisfaction column.

Hide

```
df$satisfaction <- as.factor(df$satisfaction)
```

And now we’ll split the data and conclude our data exploration.

Hide

```
set.seed(1234)
i <- sample(1:nrow(df), .80*nrow(df), replace = FALSE)
train <- df[i, ]
test <- df[-i, ]
head(train)
str(train)
tail(train)
```

## Plots

We'll make two plots, a box-plot and CD plot.

Hide

```
plot(df$satisfaction, df$Flight.Distance, data = train, main = "Satisfaction based on Flight
Distance", varwidth = TRUE)
cdplot(df$satisfaction~df$Flight.Distance)
```

# Building the Models

Below we'll build the logistic regression model.

Hide

```
glm1 <- glm(df$satisfaction~df$Flight.Distance, data=train, family=binomial)
summary(glm1)
```

Our deviance stats are concerning, as we want to see a residual much lower than the null deviance and they are approximately 200 units away. Worrying stats. Especially considering the AIC, which should be as low as possible. Seeing a value closing in on 200,000 does not inspire me with too much confidence. We have a really small change in log odds in our coefficient, log odds being the logarithm of odds, where odds is the ratio of successes to failures or in our case, satisfaction over dissatisfaction.

Next we'll construct a Naive Bayes model.

Hide

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

We see that the percent of dissatisfied is around 45% and those who are satisfied are around 55% of all passengers whose data we have in the csv file. Our flight distance variable is continuous, so we see that the mean flight distance for dissatisfaction would be upwards of 2055 units and satisfactory flight distance would be anywhere from our minimum flight distance to 1944 units.

# Testing the Models

Hide

```
probs <- predict(glm1, data = test, type = 'response')
pred <- ifelse(probs>0.5, 1, 0)
table(pred, df$Flight.Distance)
acc <- mean(pred==df$Flight.Distance)

raw <- predict(nb1, newdata = test, type = "raw")
pred <- predict(nb1, newdata = test, type = "class")
```

# Conclusions

- Logistic regression is a great classification method for figuring out binary decision cutoffs. Additionally, it is easily extendable for multiple classes and is pretty easy to use in R. Logistic regression also makes an inherent assumption that there is a linear relationship between target and predictor(s) when this isn’t always the case. This can be difficult in real-world scenarios as a strictly linear data set may be hard to come by. Naive Bayes is another great and efficient classification method. Bayes is fast and easy to interpret. It has its strengths in having a lot of predictors, because Naive Bayes will always assume that its predictors are independent. This is far less computationally intense, but it’s a drastic assumption. However, it is a good compromise for computing power and accuracy.