Classification

Kyle Chan, Ryan Banafshay

For classification, we used a dataset (https://www.kaggle.com/datasets/purumalgi/music-genre-classification (https://www.kaggle.com/datasets/purumalgi/music-genre-classification)) that contains information on 17,996 different songs.

The target variable for this dataset is Class, which represents genres of music. 0 = Folk, 1 = Alt, 2 = Blues, 3 = Bollywood, 4 = Country, 5 = Hip Hop, 6 = Indie, 7 = Instrumental, 8 = Metal, 9 = Pop.

Unlike linear regression models, the target variables in classification are qualitative. Logistic regression models predicts a dependent data variable by analyzing the relationship between one or more existing independent variables. For example, we will try to predict genre of music based on the independent variables of danceability, energy, and popularity. This is an important tool in the study of machine learning as logistic regression as it allows algorithms to predict a dependent data variable by analyzing the relationship between one or more existing independent variables. Naive Bayes models are based on conditional probability and assume strong, or naive, independence between attributes of data points.

Loading in the data

```
MusicGenre <- read.csv("~/Desktop/MusicGenre.csv", header=TRUE)
str(MusicGenre)
```

```
## 'data.frame':    17996 obs. of  17 variables:
##  $ Artist.Name       : chr  "Bruno Mars" "Boston" "The Raincoats" "Deno" ...
##  $ Track.Name        : chr  "That's What I Like (feat. Gucci Mane)" "Hitch a Ride" "N
o Side to Fall In" "Lingo (feat. J.I & Chunkz)" ...
##  $ Popularity        : num  60 54 35 66 53 53 48 55 29 14 ...
##  $ danceability      : num  0.854 0.382 0.434 0.853 0.167 0.235 0.674 0.657 0.431 0.7
16 ...
##  $ energy            : num  0.564 0.814 0.614 0.597 0.975 0.977 0.658 0.415 0.776 0.8
85 ...
##  $ key               : num  1 3 6 10 2 6 5 5 10 1 ...
##  $ loudness          : num  -4.96 -7.23 -8.33 -6.53 -4.28 ...
##  $ mode              : int  1 1 1 0 1 1 0 1 1 0 ...
##  $ speechiness       : num  0.0485 0.0406 0.0525 0.0555 0.216 0.107 0.104 0.025 0.052
7 0.0333 ...
##  $ acousticness      : num  1.71e-02 1.10e-03 4.86e-01 2.12e-02 1.69e-04 3.53e-03 4.0
4e-01 1.75e-01 2.21e-05 6.14e-02 ...
##  $ instrumentalness  : num  NA 4.01e-03 1.96e-04 NA 1.61e-02 6.04e-03 1.34e-06 5.65e-
06 1.30e-03 NA ...
##  $ liveness          : num  0.0849 0.101 0.394 0.122 0.172 0.172 0.0981 0.132 0.179
0.253 ...
##  $ valence           : num  0.899 0.569 0.787 0.569 0.0918 0.241 0.677 0.347 0.318 0.
833 ...
##  $ tempo             : num  134 116 148 107 199 ...
##  $ duration_in.min.ms: num  234596 251733 109667 173968 229960 ...
##  $ time_signature    : int  4 4 4 4 4 4 4 4 4 4 ...
##  $ Class             : int  5 10 6 5 10 6 2 4 8 9 ...
```

```
summary(MusicGenre)
```

```
##  Artist.Name         Track.Name          Popularity      danceability
##  Length:17996       Length:17996       Min.   :  1.00   Min.   :0.0596
##  Class :character   Class :character   1st Qu.: 33.00   1st Qu.:0.4320
##  Mode  :character   Mode  :character   Median : 44.00   Median :0.5450
##                                        Mean   : 44.51   Mean   :0.5434
##                                        3rd Qu.: 56.00   3rd Qu.:0.6590
##                                        Max.   :100.00   Max.   :0.9890
##                                        NA's   :428
##      energy               key            loudness            mode
##  Min.   :0.0000203   Min.   : 1.000   Min.   :-39.952   Min.   :0.0000
##  1st Qu.:0.5090000   1st Qu.: 3.000   1st Qu.: -9.538   1st Qu.:0.0000
##  Median :0.7000000   Median : 6.000   Median : -7.016   Median :1.0000
##  Mean   :0.6627767   Mean   : 5.952   Mean   : -7.911   Mean   :0.6368
##  3rd Qu.:0.8600000   3rd Qu.: 9.000   3rd Qu.: -5.189   3rd Qu.:1.0000
##  Max.   :1.0000000   Max.   :11.000   Max.   :  1.355   Max.   :1.0000
##                      NA's   :2014
##   speechiness       acousticness    instrumentalness    liveness
##  Min.   :0.02250   Min.   :0.0000   Min.   :0.000    Min.   :0.0119
##  1st Qu.:0.03480   1st Qu.:0.0043   1st Qu.:0.000    1st Qu.:0.0975
##  Median :0.04740   Median :0.0814   Median :0.004    Median :0.1290
##  Mean   :0.07971   Mean   :0.2471   Mean   :0.178    Mean   :0.1962
##  3rd Qu.:0.08300   3rd Qu.:0.4340   3rd Qu.:0.200    3rd Qu.:0.2580
##  Max.   :0.95500   Max.   :0.9960   Max.   :0.996    Max.   :1.0000
##                                     NA's   :4377
##     valence           tempo        duration_in.min.ms  time_signature
##  Min.   :0.0183   Min.   : 30.56   Min.   :      0.5   Min.   :1.000
##  1st Qu.:0.2970   1st Qu.: 99.62   1st Qu.: 166337.0   1st Qu.:4.000
##  Median :0.4810   Median :120.07   Median : 209160.0   Median :4.000
##  Mean   :0.4862   Mean   :122.62   Mean   : 200744.5   Mean   :3.924
##  3rd Qu.:0.6720   3rd Qu.:141.97   3rd Qu.: 252490.0   3rd Qu.:4.000
##  Max.   :0.9860   Max.   :217.42   Max.   :1477187.0   Max.   :5.000
##
##      Class
##  Min.   : 0.000
##  1st Qu.: 5.000
##  Median : 8.000
##  Mean   : 6.696
##  3rd Qu.:10.000
##  Max.   :10.000
##
```

Data Cleaning

Cleaning the data to only focus on the popularity, danceability, and energy columns. We will try to see how significant these elements are in determining the genre of music.

```
MusicGenre <- MusicGenre[,c(3,4,5,17)]
MusicGenre$Class <- factor(MusicGenre$Class)
head(MusicGenre)
```

```
##   Popularity danceability energy Class
## 1         60        0.854  0.564     5
## 2         54        0.382  0.814    10
## 3         35        0.434  0.614     6
## 4         66        0.853  0.597     5
## 5         53        0.167  0.975    10
## 6         53        0.235  0.977     6
```

Dividing our data into training and testing sets

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

Data exploration

Exploring the different variables we will use for our model

```
summary(train$danceability)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max. 
##  0.0599  0.4320  0.5460  0.5441  0.6600  0.9890
```

```
summary(train$energy)
```

```
##      Min.   1st Qu.    Median      Mean   3rd Qu.      Max. 
## 0.0000203 0.5090000 0.6990000 0.6623265 0.8580000 1.0000000
```

```
summary(train$Popularity)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.    NA's 
##    1.00   33.00   44.00   44.58   56.00  100.00     346
```

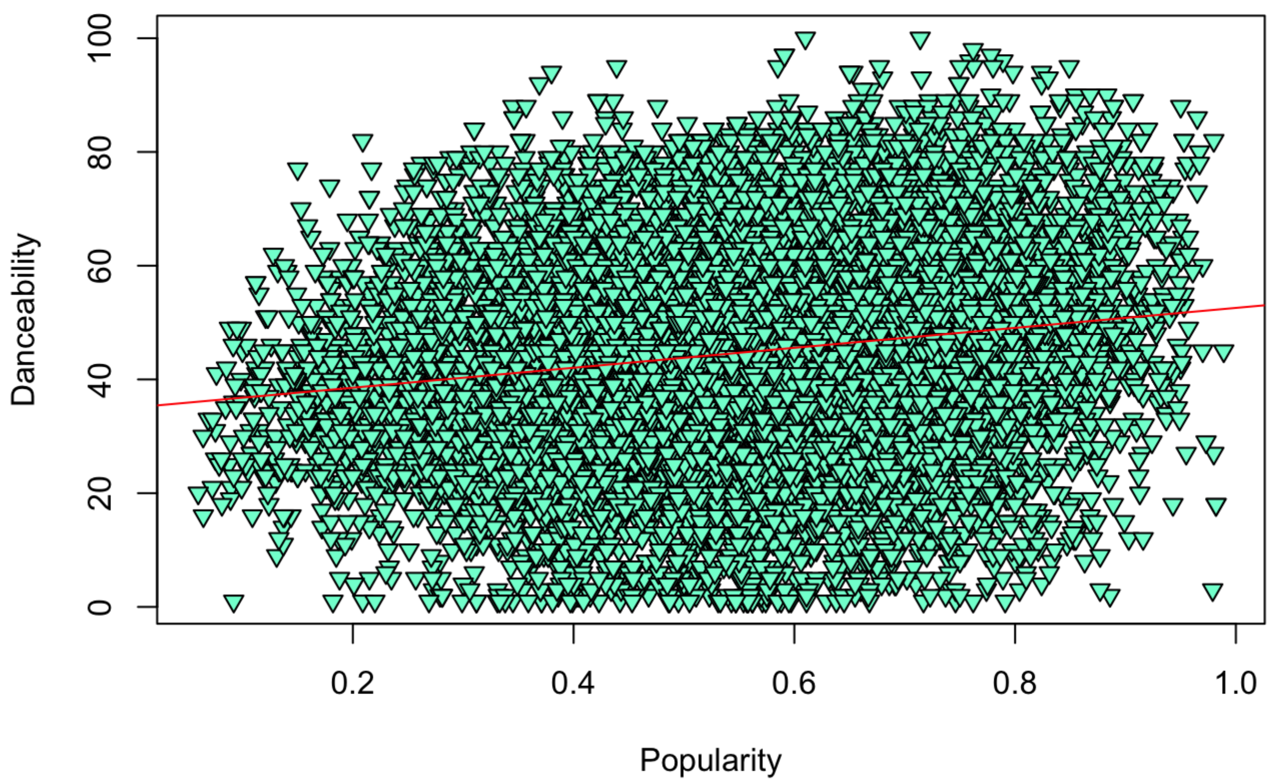
```
range(train$danceability)
```

```
## [1] 0.0599 0.9890
```

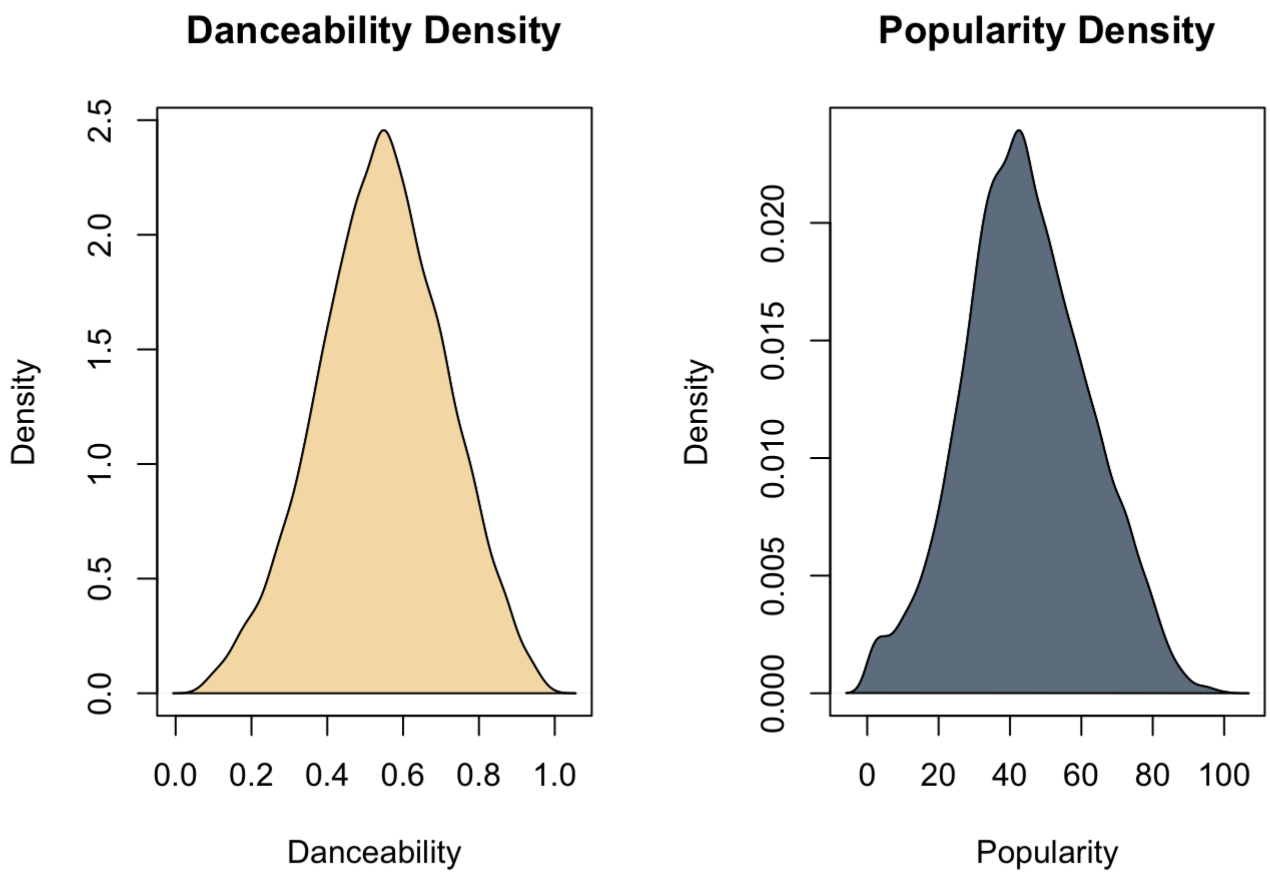
```
range(train$energy)
```

```
## [1] 2.03e-05 1.00e+00
```

```
par(mfrow=c(1,1))
plot(train$Popularity~train$danceability, xlab= "Popularity", ylab= "Danceability", pch=
25, bg=c("aquamarine1"))
abline(lm(train$Popularity~train$danceability), col = "red")
```




```
par(mfrow=c(1,2))
dance_den <- density(train$danceability, na.rm = TRUE)
plot(dance_den, main = "Danceability Density", xlab = "Danceability")
polygon(dance_den, col ="wheat")
Popularity_den <- density(train$Popularity, na.rm = TRUE)
plot(Popularity_den, main = "Popularity Density", xlab = "Popularity")
polygon(Popularity_den, col ="slategrey")
```



Logistic Regression Model

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

```
##
## Call:
## glm(formula = Class ~ ., family = "binomial", data = train)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -3.1441   0.1369   0.1817   0.2636   0.9422
##
## Coefficients:
##              Estimate Std. Error z value Pr(>|z|)
## (Intercept)  0.482192   0.191079   2.524   0.0116 *
## Popularity   0.019109   0.002917   6.551 5.72e-11 ***
## danceability 0.147124   0.289701   0.508   0.6116
## energy       3.682136   0.194862  18.896  < 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: 3996.1  on 14049  degrees of freedom
## Residual deviance: 3545.1  on 14046  degrees of freedom
##   (346 observations deleted due to missingness)
## AIC: 3553.1
##
## Number of Fisher Scoring iterations: 7
```

From this logistic regression model, we can determine that the popularity of the track has a significant impact the genre of music. This makes sense in some regards, if a music is classified under the genre of “pop” then we can probably guess it will be much more popular than a song under the genre of “folk”. Energy seems to also be contribute to the genre of music, while on the other hand danceability seems to have little to no impact.

Logistic regression model for just energy

```
glm2 <- glm(Class~energy, data=train, family="binomial")
summary(glm2)
```

```
##
## Call:
## glm(formula = Class ~ energy, family = "binomial", data = train)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -3.0708   0.1459   0.1971   0.2812   0.7351
##
## Coefficients:
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept)  1.17055    0.09273   12.62   <2e-16 ***
## energy       3.83456    0.18105   21.18   <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: 4435.2  on 14395  degrees of freedom
## Residual deviance: 3966.7  on 14394  degrees of freedom
## AIC: 3970.7
##
## Number of Fisher Scoring iterations: 7
```

Naïve Bayes

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

```
##
## Naive Bayes Classifier for Discrete Predictors
##
## Call:
## naiveBayes.default(x = X, y = Y, laplace = laplace)
##
## A-priori probabilities:
## Y
##          0          1          2          3          4          5          6
## 0.03570436 0.07696582 0.07106141 0.02222840 0.02076966 0.08092526 0.14288691
##          7          8          9         10
## 0.03105029 0.10148652 0.14184496 0.27507641
##
## Conditional probabilities:
##     Popularity
## Y        [,1]     [,2]
##   0  38.04425 16.74758
##   1  45.93161 14.16165
##   2  32.79802 12.94973
##   3  25.52273 17.32002
##   4  57.38869 12.89005
##   5  48.62141 18.74983
##   6  41.36368 15.89847
##   7  41.62736 10.40698
##   8  42.58467 13.08614
##   9  50.43537 20.97848
##   10 47.14242 16.64562
##
##     danceability
## Y         [,1]      [,2]
##   0  0.5232957 0.1222264
##   1  0.5375773 0.1616415
##   2  0.5637537 0.1391830
##   3  0.4927969 0.1277167
##   4  0.5972609 0.1088574
##   5  0.7266918 0.1377203
##   6  0.5516035 0.1657289
##   7  0.4298680 0.1691011
##   8  0.4110979 0.1321048
##   9  0.6344461 0.1427815
##   10 0.5015744 0.1396259
##
##     energy
## Y         [,1]      [,2]
##   0  0.4289459 0.1928056
##   1  0.6822909 0.2147621
##   2  0.5856543 0.2338232
##   3  0.5260478 0.1932745
##   4  0.5992234 0.1976177
##   5  0.6410686 0.1598373
##   6  0.6569323 0.2189865
##   7  0.1528568 0.1234892
```

```
##   8  0.8732270 0.1473804
##   9  0.6183667 0.1929602
##   10 0.7340396 0.1966761
```

Evaluate Naïve Bayes

```
p2 <- predict(nb1, newdata=test, type="class")
table(p2, test$Class)
```

```
##
## p2      0   1   2   3   4   5   6   7   8   9  10
##   0     3   0   5   0   0   0   9   2   0   2   4
##   1     0   0   0   0   0   0   0   0   0   0   0
##   2     9   3  27  14   1   4  23   8   3  23  33
##   3     3   1   7  11   0   0   7   0   0   3   2
##   4     0   0   0   0   0   0   0   0   0   0   0
##   5     5  13  10   2   3 116  47   2   0  74  23
##   6     2   4   5   4   1   6   7   1   3   7   6
##   7    20  10  21   1   5   2  32 108   2  23  18
##   8     1  31  20   1   0   2  34   1 183   4 146
##   9    21  30  50  11  28  68  79   5   3 142  86
##   10   47 173 104  38  50  84 292   2 199 204 671
```

```
mean(p2==test$Class)
```

```
## [1] 0.3522222
```

Based on this mean result, it is hard to rely too much on the naive bayes analysis as it shows to be not as accurate as the logistic model for this data.

Strengths and weaknesses of Logistic vs Naive Bayes

Both logistic regression and Naive Bayes have similarities, as they are both linear classifiers and are both used for classification. A strength of logistic regression is that it is typically low bias, meaning it incorporates fewer assumptions about the target function. Lower bias models tend to closely match the training data set. But on the flip side they tend to have a higher variance. This is the opposite for Naive Bayes models, as they tend to have higher bias but lower variance. So if the data set follows the bias then Naive Bayes will be a better classifier. Another benefit of Naive Bayes is that results are easier to predict with less variables and less data. Logisitic regression is better for multinomial classification problems, such as the one we did in this assignment.

Benefits and drawbacks

As we used a large dataset with multinomial classifications (more than two possible discrete outcomes rather than the binary 0 and 1), I felt that the results from logistic regression was far more beneficial for drawing conclusions on the data. The classification methods here are incredibely general though, and I felt some of the variables were a bit arbitrary. I don't understand how the model determined that energy is a determining factor of classification, but not dancability. This very well could be due to how the data was collected.