Statistical Learning First Assignment

Camila San Jose

Introduction

Some say that music is a universal language, it is embedded in our daily lives since we are infants. However, music has changed drastically throughout generations, passing through various genres. What do people listen to now and, what makes a song a certain type of genre?

To get to know what people to listen to nowadays, I gathered two datasets containing the top songs of 2017[1] and 2018[2] from the application Spotify. Spotify is a music streaming service that is very well known globally as it has access to millions of songs, videos and playlist. The merge of both datasets result in 200 tracks and the following 14 variables:

- **Name** - the name of the song.
- **Artist** - the artist of the song
- **Danceability**- consists in combination of musical elements: tempo, beat strength, rhythm stability, and overall regularity. It goes from values 0.0, as the least danceable, to 1.0 which is the most danceable.
- **Energy**- Measure from 0 to 1 that represents a measure of intensity and activity.
- **Key**- Key the track is in. Pitches using the standard Pitch Class notation. (ex. C=0, C#/D flat = 1, D=2, etc.)
- **Loudness**- Overall loudness of a track in decibels (dB). It is the quality of a sound that is the primary psychological correlate of physical strength (amplitude). The values are between -60dB and 0dB.
- **Speechiness**- Presence of spoken words in a track. The more speech there is, the value is closer to 1. Values above 0.66 are tracks that are entirely of spoken words. Values between 0.33 and 0.66 attribute tracks that contain both speech and music, and values below 0.33 represent only music or other non-speech tracks.
- **Acousticness**- A confidence measure from 0 to 1 (highest confidence) whether the track is acoustic.
- **Instrumental**- Predicts whether a track contains no vocals. The closer the value to 1, the greater the likelihood the track does not contain any vocals.
- **Liveness**- Detects the presence of an audience in the recording. This is to detect if the track was performed live. A value above 0.8 provides strong likelihood that the track is live.
- **Valence**- Measured from 0 to 1 describing the musical positiveness conveyed by a track. Values of 0 is negative tracks (sad, depressed or angry) while 1 is high valence that are cheerful or happy tracks.
- **Tempo**- The overall estimated tempo of a track in beats per minute (BPM), also known as the speed or pace that derives from the average beat duration.
- **Duration**- Duration of track in milliseconds.
- **Genre**- The genre of the track which can be Hip-Hop, Pop or Reggaeton.

For this assignment, we are going to take the best explicative model of the variables when predicting the categorical variable: Genre. Similarly, we will obtain the best predictive model to predict if a song is `Pop`, `Hip-Hop` or `Reggaeton`. First of all, we are going to observe how each of the variables behave.

Data Preprocessing

Before we could use the data, we had to perform some data cleaning. Although there were no missing values, there were some repeated songs given the merge of the two datasets. These repeated tracks were due to

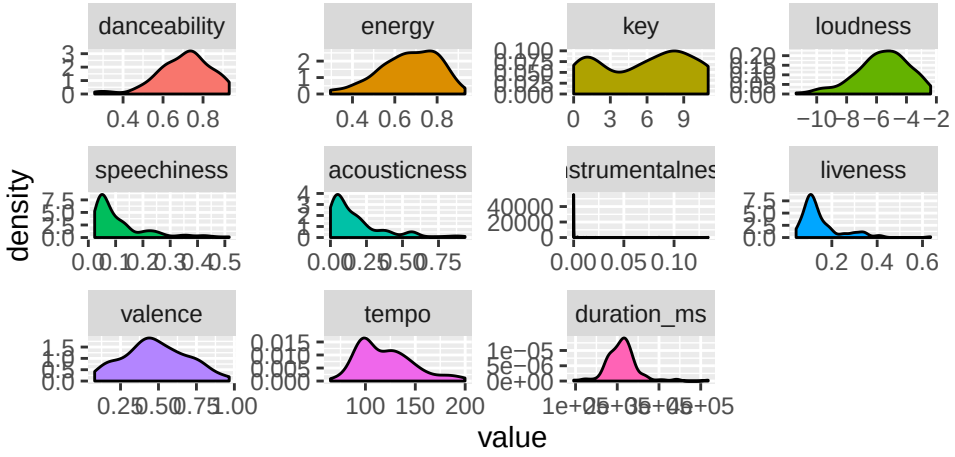
[1]Nadin Tamer. (2017, December). Top Spotify Tracks of 2017: Audio features of top Spotify songs [Version 1]. Retrieved November 25, 2019 from https://www.kaggle.com/nadintamer/top-tracks-of-2017/metadata

[2]Please note that the training was made with a `set.seed(71)` so as to have a control over results, However, results can vary every time the model is run without it.

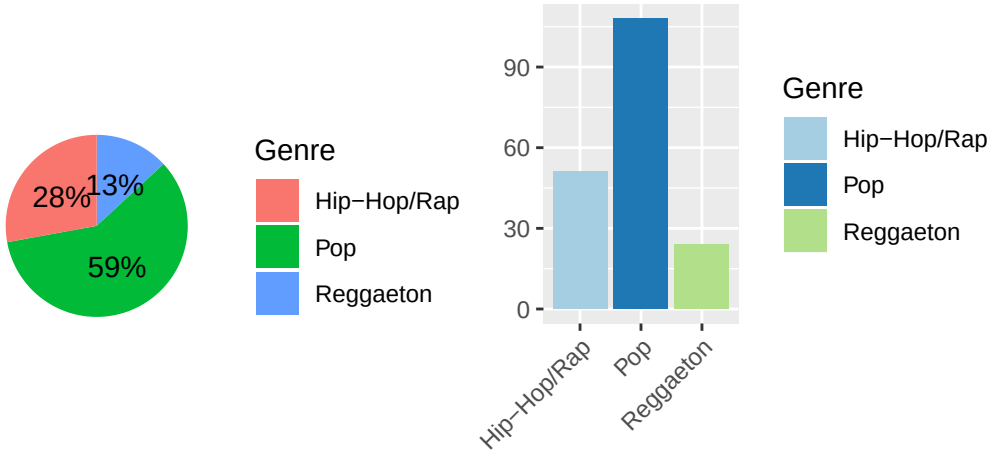
songs that were top tracks for two consecutive years (2017 and 2018), which lead to an elimination of one of the two tracks. There were 17 repeated songs and as a result, we were left with 183 tracks to perform the analysis. Likewise, we did not use the variables Name or Artist to perform our predictive or explicative analysis.

Descriptive Analysis

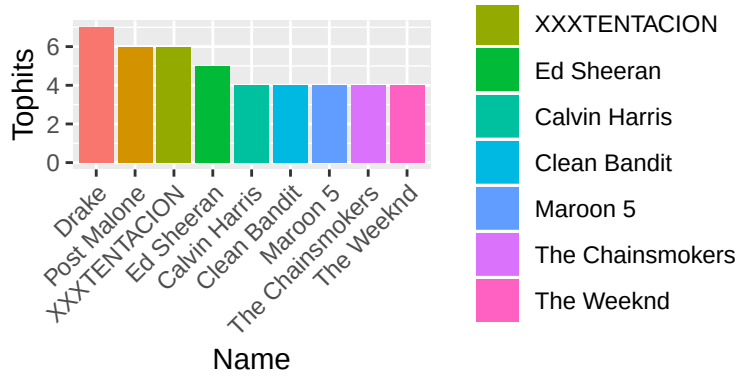
To analyze the variables more in-depth, we first made a graph of their densities to be able to see their behaviour (for the 11 continuous variables). In this graph we can observe that danceability and loudness are left skewed which means that most songs are very danceable and have a high amplitud, while speechiness, acousticness, and liveness are right skewed which means that they have high presence music, are not very acoustic, and are not usually recorded in live performances. At the same time, we can note that almost all tracks have vocals seen in the liveness variable, which shows that almost all songs have a value of 0 or close to 0, which represents the inclusion of vocals, as well as most tracks have a duration between 3 to 4 minutes. Nevertheless, to be able to observe the distribution more closely we applied a square root transformation to the variable instrumentalness. We did not apply any other transformation.



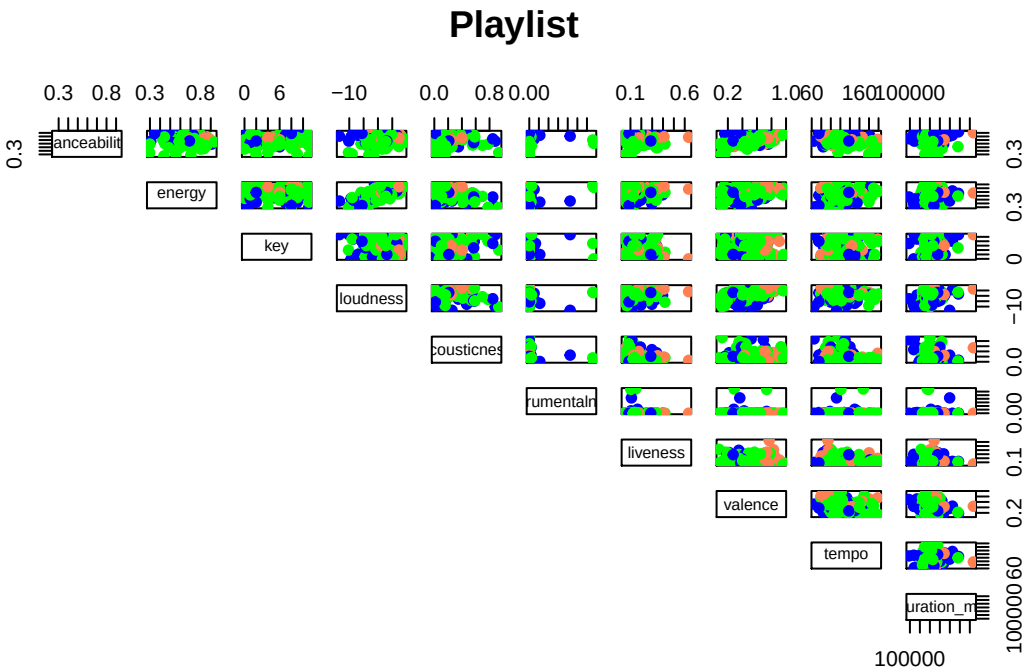
On the other hand, the dataset has one categorical variable, the genre of the songs. There are three categories: Pop, Hip-Hop, and Reggaeton. We can see in the graphs below that about 60% of the songs are Pop, followed by Hip-Hop with approximately 28%. This means that the observations are unbalanced. This will later be an issue when making predictive models as it can impact their accuracy.



Likewise, we can see that nine artists make up 24% of all the songs, all with 4 or more tracks. In the following graph we can see that the artist Drake has the most top tracks (7 songs), followed by Post Malone and XXXTentacion.



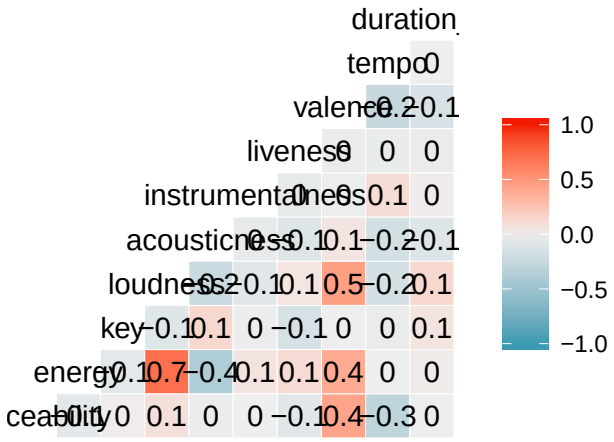
Furthermore, to observe the relation between the variables, we made scatterplots comparing each of the variables. Also, we divided these by the Genre to see if there is a clear distinction of each category. In the graph we can note that the Genre Pop (in green) is most dinstinguishable in variables such as speechiness, acousticness and loudness. In addition, there is a clear linear relation between energy and loudness, but we canot say the same for other variables.



Explicative Model

To explain what makes the genre of the song we will create a logistic regression model with the variable Genre as the response variable. Thus, to obtain the best explicative model, we first need to see if there is correlation between the variables. We made the following correlation plot to analyze this. In the graph we can observe that loudness and energy are the highest correlated variables, with a correlation of approximately

0.7. This will impact the explicative power of the variables if both variables are included. Similarly, Valence and loudness have a medium correlation of 0.5 which will also affect its explicative power, but will not be as impactfull. For the other variables, we can see there is little correlation, which is useful when obtaining the explicative model.



To model the logistic regression, in order to predict the variable `Genre`, we use the `vglm()` syntax which is the *generalized linear model* and include the argument `family= multinomial(refLevel=1` given that we have three categories (it would be binomial if we had only 2). We first include all the variables to see which is significant.

```
logit.exp <- vglm(Genre ~ danceability+energy+loudness+speechiness+acousticness+instrumentalness+
  liveness+valence+tempo+duration_ms, family=multinomial(refLevel=1), data=data2)
summary(logit.exp)
```

```
##
## Call:
## vglm(formula = Genre ~ danceability + energy + loudness + speechiness +
##     acousticness + instrumentalness + liveness + valence + tempo +
##     duration_ms, family = multinomial(refLevel = 1), data = data2)
##
## Pearson residuals:
##                         Min      1Q  Median       3Q    Max
## log(mu[,2]/mu[,1]) -3.991 -0.5613  0.3369  0.61103  1.813
## log(mu[,3]/mu[,1]) -3.673 -0.3285 -0.1006 -0.02111 13.798
##
## Coefficients:
##                      Estimate Std. Error z value Pr(>|z|)
## (Intercept):1       9.756e+00  3.707e+00      NA       NA
## (Intercept):2      -1.187e+01  6.642e+00      NA       NA
## danceability:1     -7.488e+00  2.098e+00  -3.569 0.000358 ***
## danceability:2      1.213e+00  3.727e+00   0.325 0.744895
## energy:1            1.101e+00  2.542e+00   0.433 0.664853
## energy:2            1.030e+01  4.170e+00   2.470 0.013496 *
## loudness:1          2.113e-01  1.953e-01   1.082 0.279426
## loudness:2         -5.874e-03  3.091e-01  -0.019 0.984838
## speechiness:1      -8.889e+00  2.444e+00  -3.638 0.000275 ***
## speechiness:2      -7.917e+00  4.548e+00  -1.741 0.081713 .
## acousticness:1     -6.753e-01  1.162e+00  -0.581 0.561155
```

```
## acousticness:2      1.044e+00  1.842e+00   0.567 0.570647
## instrumentalness:1 -3.322e+00  1.559e+01  -0.213 0.831258
## instrumentalness:2 -2.045e+02  3.706e+02  -0.552 0.581170
## liveness:1         -1.589e+00  2.181e+00  -0.729 0.466295
## liveness:2          1.985e+00  2.756e+00   0.720 0.471246
## valence:1           1.028e+00  1.249e+00   0.823 0.410624
## valence:2           4.492e+00  1.922e+00   2.338 0.019393 *
## tempo:1            -6.344e-03  7.937e-03  -0.799 0.424161
## tempo:2            -1.157e-03  1.205e-02  -0.096 0.923542
## duration_ms:1      -6.585e-06  5.613e-06  -1.173 0.240688
## duration_ms:2       5.217e-06  8.290e-06   0.629 0.529151
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Names of linear predictors: log(mu[,2]/mu[,1]), log(mu[,3]/mu[,1])
##
## Residual deviance: 243.1417 on 344 degrees of freedom
##
## Log-likelihood: -121.5709 on 344 degrees of freedom
##
## Number of Fisher scoring iterations: 9
##
## Warning: Hauck-Donner effect detected in the following estimate(s):
## '(Intercept):1', '(Intercept):2'
##
##
## Reference group is level  1  of the response
```

```
AICvlm(logit.exp)
```

```
## [1] 287.1417
```

In the summary, the significant variables (the ones that have a p-value lower than 0.05) are: `danceability, energy, speechiness,` and `valence`. Therefore, we only include these variables in the model, and also, we check to see if any of the interactions are significant. To check which model is the best, we will use the `AIC`, which is a method for assessing the quality of your model, penalizing when the model is more complicated. In other words, the model with the lowest `AIC` is your best explicative model. With this, we obtained the following model (note that we included the interaction of danceability and valeance):

```
logit.exp3 <- vglm(Genre ~ danceability+energy+speechiness+valence+
  danceability:valence, family=multinomial(refLevel=1), data=data2)
AICvlm(logit.exp3)
```

```
## [1] 269.126
```

Predictive Model

To get to know what kind of genre people listen to nowadays, we are interested in predicting whether a song falls in any of the three categories: `Pop, Hip-Hop`, or `Reggeaton`. We will try different models and see which one is the best for the data. First of all, we divide the data into two parts. 80% will be dedicated to the training, in other words, our method to estimate our response. Let x_{ij} represent the value of the j-th predictor, where $j = 1, ..., p$, $i = 1, ..., n$ ($p = 148$ and $n = 83$). The data consists of

$\{(x_1, y_1), (x_2, y_2), \ldots, (x_n, y_n)\}$ where $x_i = (x_{i1}, x_{i2}, \ldots, x_{ip})^T$, while the rest (20%) will be for testing. In addition, we will use the *kappa* to measure performance, given that it adjusts accuracy by the possibility of a correct prediction obtained by chance and likewise, we will use all the variables in our predictive models.[3]

```
set.seed(71)
in_train <- createDataPartition(data2$Genre, p = 0.8, list = FALSE)
training <- data2[ in_train,]
testing <- data2[-in_train,]
```

In the absence of a large designated test, the test error rate is hard to estimate. Therefore, we will use the method of *k-fold cross-validation* to reduce the uncertainty of the model prediction. This involves randomly dividing the set of training into k groups (in this case, we chose $k = 10$ given that we tried with many numbers and this one got the best results). The first fold is treated as a validation set, and the method is fit on the remaining $k - 1$ folds, then the mean squared error is computed on the observations in the held-out fold. This is repeated k times, and a different group of observations is treated as a validation set. The k-fold CV estimate is computed by averaging the test error: $\text{CV}_{(k)} = \frac{1}{k}\sum_{i=1}^{k} \text{MSE}_i$. We will use this method for all models, excluding the logist regression.

On the other hand, we will not adjust the threshold given the nature of the response variable - the genre. This means that it is not cost-sensitive, wehre if there is an error in classification it will have the same impact for `Pop`, `Hip-Hop` and `Reggaeton`.

Logistic Regression

As with the explicative model, we will first use the logistic regression to predict the genre, this time, taking into consideration all of the variables. The training part of our set and the `multinomial` family were used for this model. We obtain a kappa of 48%.

```
logit.model <- vglm(Genre ~ danceability+energy+loudness+speechiness+acousticness+
instrumentalness+liveness+valence+tempo+duration_ms, family=multinomial(refLevel=1), data=training)
prob.test=predict(logit.model, newdata=testing,type="response")
pred.test<-as.factor(levels(as.factor(data2$Genre))[max.col(prob.test)])
confusionMatrix(pred.test,testing$Genre)$table
```

```
##              Reference
## Prediction    Hip-Hop/Rap Pop Reggaeton
##   Hip-Hop/Rap           6   2         1
##   Pop                   3  17         1
##   Reggaeton             1   2         2
```

```
confusionMatrix(pred.test,testing$Genre)$overall[1:2]
```

```
##  Accuracy     Kappa
## 0.7142857 0.4807122
```

Penalized Logistic Regression

We are using the penalized logistic regression `glment()` for the next predictive model where the model imposes a penalty for having too many variables, which will shrink the coefficients of the less significant towards zero (also known as regularization). We then a lower kappa of 41%:

[3]Please note that the training was made with a `set.seed(71)` so as to have a control over results, However, results can vary every time the model is run without it.

```
lrFit<- train(
  Genre~ danceability+energy+loudness+speechiness+acousticness+
instrumentalness+liveness+valence+tempo+duration_ms,
  method ="glmnet",
  family="multinomial",
  data= training,
  metric ="Accuracy",
  trControl = trainControl(method= "cv", number =10)
)
lrpred<- predict(lrFit, testing)
confusionMatrix(lrpred,testing$Genre)$table
```

```
##              Reference
## Prediction    Hip-Hop/Rap Pop Reggaeton
##   Hip-Hop/Rap           5   2         1
##   Pop                   5  19         2
##   Reggaeton             0   0         1
```

```
confusionMatrix(lrpred,testing$Genre)$overall[1:2]
```

```
##  Accuracy     Kappa
## 0.7142857 0.4117647
```

Penalized Logistic Regression with Adjusted Hyper-Parameters

We can also adjust this model's hyperparameters to obtain a better kappa (a better fit of the model). For this, we need to know its alpha and lambda. When running the model we obtain an $\alpha = 0.5$ and a $\lambda = 0.09$. If we would have obtained an alpha closer to 0, then we would use the ridge regression and if alpha would be closer to 1, then we would have a model using Lasso. However, our alpha is between two methods. Nevertheless, the lambda obtained is very small, which represents a low multicollinearity. As a result, when we include these parameters in the predictive model we obtain a better predictive model with kappa of 49%. We obtain the following:

```
lrFit2<- train(Genre~ danceability+energy+loudness+speechiness+acousticness+
instrumentalness+liveness+valence+tempo+duration_ms,
  method ="glmnet",family="multinomial",
  data= training, preProcess = c("center", "scale"),
tuneGrid =expand.grid(alpha=seq(0,1,0.1), lambda = seq(0,0.1,0.01)),
    metric ="Accuracy", trControl = trainControl(method= "cv", number =10))
lrpred2<- predict(lrFit2, testing)
confusionMatrix(lrpred2,testing$Genre)$table
```

```
##              Reference
## Prediction    Hip-Hop/Rap Pop Reggaeton
##   Hip-Hop/Rap           6   2         1
##   Pop                   3  19         2
##   Reggaeton             1   0         1
```

```
confusionMatrix(lrpred2,testing$Genre)$overall[1:2]
```

```
##  Accuracy     Kappa
## 0.7428571 0.4943820
```

Linear Discrminant Analysis (LDA)

For the LDA model, we need to assume certain characteristics for our predictive model. First of all, we assume that the model is a multivariate gaussian and therefore, each class comes from a normal distribution $N(\mu_k, \mathbf{\Sigma})$ with a common covariance $\mathbf{\Sigma}$ but an individual mean (μ_k). Therefore, we are scaling the data to be able to assume normality. We get a Kappa of 41%.

```
ldaFit <- train(Genre ~ danceability+energy+loudness+speechiness+acousticness+
  instrumentalness+liveness+valence+tempo+duration_ms,
              method = "lda", data = training,
              preProcess = c("center", "scale"), metric = "Accuracy",
  trControl = trainControl(method= "cv", number =10))
ldaPred = predict(ldaFit, testing)
confusionMatrix(ldaPred,testing$Genre)$table
```

```
##              Reference
## Prediction    Hip-Hop/Rap Pop Reggaeton
##   Hip-Hop/Rap           6   2         1
##   Pop                   3  17         2
##   Reggaeton             1   2         1
```

```
confusionMatrix(ldaPred,testing$Genre)$overall[1:2]
```

```
##  Accuracy     Kappa
## 0.6857143 0.4140030
```

Quadratic Discriminant Analysis (QDA)

Similar to LDA, QDA assumes each class is drawn from a multivariate Gaussian distribution with class-specific mean vector. However, QDA results from plugging estimates for the parameters into Bayes' theorem in order to predict, and also it assumes that each class has its own covariance $\mathbf{\Sigma}$, and therefore each observation for the kth class is of the form $X \sim N(\mu_k, \mathbf{\Sigma}_k)$. We get a kappa of 27%.

```
qdaFit <- train(Genre ~ danceability+energy+loudness+speechiness+acousticness+
instrumentalness+liveness+valence+tempo+duration_ms,
            method = "qda", data = training,
            preProcess = c("center", "scale"),metric = "Accuracy",
            trControl = trainControl(method= "cv", number =10))
qdaPred = predict(qdaFit, testing)
```

```
confusionMatrix(qdaPred,testing$Genre)$table
```

```
##              Reference
## Prediction    Hip-Hop/Rap Pop Reggaeton
##   Hip-Hop/Rap           4   2         0
##   Pop                   5  15         2
##   Reggaeton             1   4         2
```

```
confusionMatrix(qdaPred,testing$Genre)$overall[1:2]
```

```
##  Accuracy     Kappa
## 0.6000000 0.2740741
```

Naives Bayes

We will now model with the Naives Bayes method. Therefore, we will assume independence of the variables with a high bias but low variance. We use the k-fold cross validation method, as well as center and scale the variables. We get a kappa of 23%.

```
nbFit <- train(Genre ~ danceability+energy+loudness+speechiness+acousticness+
instrumentalness+liveness+valence+tempo+duration_ms,
            method = "nb", data = training,
            preProcess = c("center", "scale"),metric = "Accuracy",
            trControl = trainControl(method= "cv", number =10))
nbPred = predict(nbFit, testing)
confusionMatrix(nbPred,testing$Genre)$table
```

```
##              Reference
## Prediction    Hip-Hop/Rap Pop Reggaeton
##   Hip-Hop/Rap           4   1         0
##   Pop                   6  14         2
##   Reggaeton             0   6         2
```

```
confusionMatrix(nbPred,testing$Genre)$overall[1:2]
```

```
##  Accuracy     Kappa
## 0.5714286 0.2290749
```

Economic Analysis

To analyze the economic effects that Spotify could have when predicting the genre of the songs, we are considering Spotify's one month free trial. In order to obtain new clients, Spotify offers its premium service one month for free. When this month is up, if spotify predicted the genre of the songs that people liked to hear correctly, the clients like the service and buy the premium service (80% of the times). Otherwise, if Spotify does not predict correctly their preferred genre based on what they listen to, then fewer clients buy the service.

```
##           Hip Hop Pop Reggaeton
## Hip Hop       0.8 0.4       0.6
## Pop           0.0 0.8       0.4
## Reggaeton     0.8 0.7       0.8
```

Using the LDA classifier to analyze the economic impact, we multiply the previous matrix with the confusion matrix of the predictive model to obtain the profit per applicant and obtain:

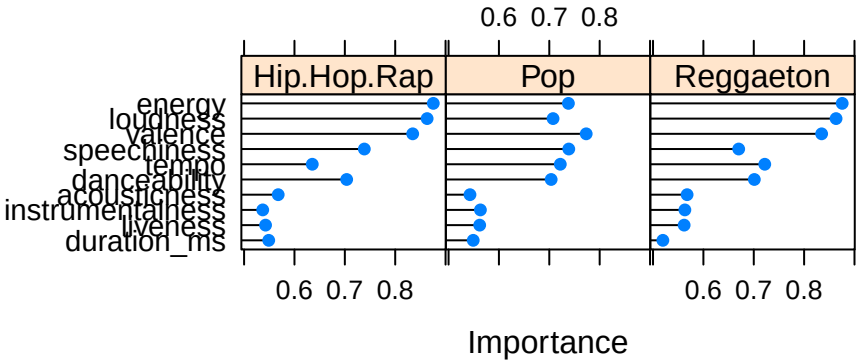
```
profit.applicant
```

```
## [1] 0.6742857
```

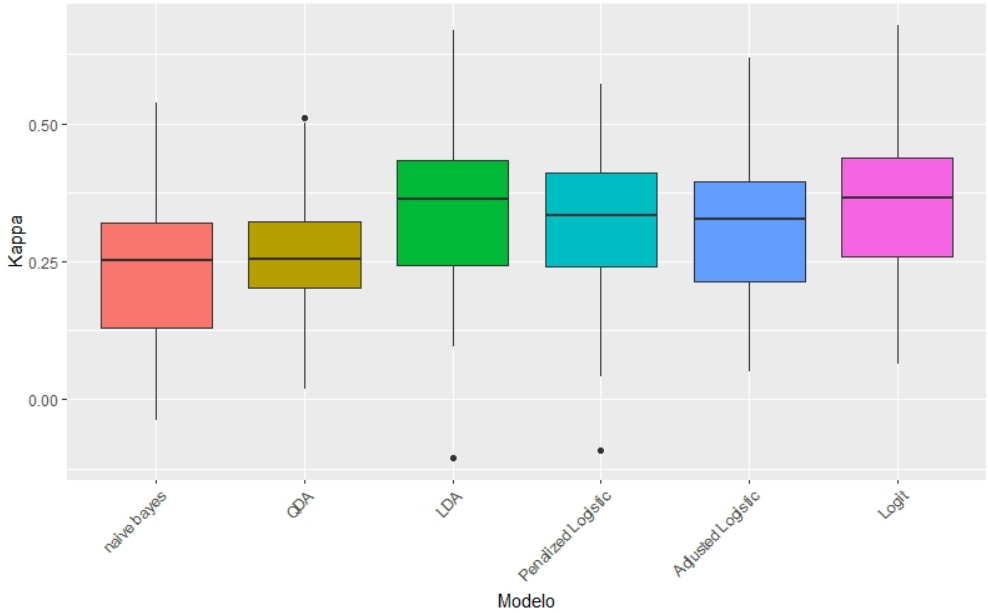
Consequently, if the service for Spotify's premium service is 9.99€, and there are over 10,000 applicants all over the world in one month then we the profit is profitapplicant$\times 9.99 \times 10000 = 63,651$£ per month.

Conclusions

In the following graph we can see how the variables of each genre behave in each of the Genres (this is true for all predictive models). Here we note that `Hip Hop` and `Reggaeton` are very similar, and `Pop` is the most distinguishable, specially the loudness variable. As well we can observe that `Hip Hop` has lower tempo, and danceability but higher speechiness when compared to the other categories.



To compare models and obtain the best one, as previously stated, we will use the kappa as the identifier. First, we make 100 iterations of the six models using different training sets (randomize the training section). Later, we save the kappas in each iterations, and graph these with a boxplot in order to observe their behavior and compare. We can see this in the graph below:



As stated by the graph, the best predictive model for this dataset according to the kappa is both the LDA and the logistic regression. Both have the highest median among all of the models. Nevertheless, although they have very similar kappa's, LDA is a better predictor for when the response variable is not binary, and given that the variables have very low multicollinearity which is shown in the low lambda obtained in the logistic regression. We can also get the ROC curve of the LDA to display the overall performance of the classifiiers. We get an ROC curve of 0.7817 for the model, which is pretty good. In conclusion, we will use the LDA model as the best predictive model where its median kappa for all of the 100 iterations is 36%, which is still a low percentage.