# Predicting song popularity using Spotify Dataset

Group 8

Vira Hayuningrum ANR: 763173
Lisa Koutsoviti Koumeri ANR: 579370
Trang Le ANR: 156066
Sue Yoon ANR: 215976
Jennifer-Zhu Lezu ANR: 352740

Tilburg University

Master in Data Science and Society

Research Skills: Programming with R

Group project

June 22, 2020

Word count: 1338

## Introduction

People listen to music for several reasons but, primarily, to regulate their moods. Additionally, music's remarkable ability to serve different needs is the reason that music has been so important to people ever since antiquity (Lonsdale & North, 2011). Spotify Technology S.A. (Spotify) is a music streaming provider supporting 286 million users as of March 21, 2020, and offering more than 50 million tracks (Spotify Company Info, 2020). The authors' deep appreciation of music and Spotify's repute as a widespread music platform led to the creation of the present paper.

The authors chose to predict song popularity based on Kaggle's Spotify dataset (Singh, 2020) using the R software environment (R Core Team, 2017). The dataset describes the top 1,994 songs on Spotify using the following attributes: title, artist, year of release, genre, beats per minute (BPM), energy, danceability, loudness, valence, length, acoustic, speechiness and popularity. Popularity was chosen to be the dependent variable. Three different prediction models were created to explore which can best predict the popularity of a song.

## Preprocessing

All the variables in the dataset were deemed relevant for the predictive analysis and were subjected to preprocessing. The type and range of each variable were inspected, and no missing data were identified. To explore the distribution of each variable, histograms were built, as shown in Graph 1. Among all the variables, only danceability has a normal, bell-shaped distribution, whereas Speechiness, Liveness, and Length are highly skewed. In an effort to identify variables that are linearly related, a correlation matrix was created. A high positive correlation was observed between the following pairs: Energy and Loudness (0.74), Energy and Accousticness (- 0.67), Danceability and Valence (0.51), Loudness, and Accousticness (- 0.45), and Valence and Energy (0.41). The independent variable that has the highest correlation coefficient regarding the target variable is Loudness (0.17). Plots that depict these relationships can be found in Appendix A.

Several changes were made in the dataset. For clarity, some variables were renamed. A new variable was created indicating the number of characters in the song titles. Another new categorical variable was created based on the variable *year,* which ranges from 1956 to 2019. The new attribute indicates each decade from the fifties until the 2000s. As the ranges of the numeric variables varied greatly and their distribution was not Gaussian, they were normalized. Normalization is a necessary precondition for applying the k-nearest neighbor algorithm in the predictive modeling section as the algorithm does not make assumptions about the distribution of the data.

The target variable i.e. popularity was initially a numeric variable. It was transformed in two ways. Firstly, a categorical binary variable was created where songs were categorized as *popular* or *unpopular* using the median as the threshold. Secondly, a multiclass categorical variable was created, including the classes *popular, normal, unpopular,* and using the 33rd and 66th quantiles as thresholds.

Finally, an outlier analysis was performed on the normalized variables starting with a boxplot visualization (Graph 3). The plots proved the existence of outliers in five variables. To identify the most influential cases, a simple linear regression model was carried out between the numeric predictors and the target feature ($F(8, 1985) = 15.48$, $p < .001$, R-squared = 0.06) followed by calculating Cook's distance with a cut-off of four divided by the number of observations (Graph 4). All observations above the cut-off were removed from the dataset resulting in a final sample size of 1907 songs. A histogram was created using the normalized, outlier-free variables (Graph 5).

## Predictive modeling

For predictive modeling purposes, feature selection was made as follows. Normalized predictors that were significantly correlated ($p =< .05$) with the normalized popularity feature were chosen, i.e. Speechiness ($r = .11$), Loudness ($r = .17$), Year ($r = -.16$), Acousticness ($r = -.09$), Length ($r = -.07$), Valence ($r = .1$), Danceability ($r = .14$) and Energy ($r = .1$). Scatterplots of the above-mentioned relationships were generated and are exhibited in Graph 6. Two different combinations of these features were chosen as input into the different models: a full set of features as mentioned above and a reduced set comprising Speechiness, Energy, Loudness, and Danceability which were the features with the highest correlation with popularity. The existence of interaction effects between popularity and the predictors by the categorical variable year were also investigated (Graph 7). Therefore, the variable year was added to the linear regression models as moderator.

The sample was divided into the train (70%) and test (30%) datasets for all models. Plots of models' performance for different numbers of k-neighbors can be found in Appendix A, and the relevant confusion matrices, metric, ROC curves, and regression coefficient tables were placed in Appendix B.

Three classifiers were used to predict song popularity. The first classifier that was employed was the KNN algorithm. The first two models were trained using 10-fold cross-validation (CV) to predict the multiclass version of the popularity feature. Model 1.1.1 was trained using all features as hyperparameters and Model 1.1.2 was trained on a reduced set of predictors. Model 1.1.1's accuracy was higher (40%). Two additional models were created using the binarized version of popularity as the target feature. 5-fold CV was employed to train both models, firstly, on the full set of predictors (Model. 1.2.1) and, secondly, on the reduced set (Model 1.2.2). Model 1.2.1 was more accurate (58%). The numbers of k-neighbors, i.e. 2,3,5,8,10 and 20, were chosen empirically.

The second classifier was the logistic regression algorithm. The binarized popularity feature served as the target feature. Model 2.1 was trained using 10-fold CV on the full set of features and Model 2.2 using 5-fold CV on the reduced set. Model 2.1 demonstrated higher accuracy (63%). The higher level of accuracy of all four binary classification models, can be explained by the lower number of classes. As the number of classes decreases, the probability of correctly predicting a class increases slightly. A summary of the metrics of all classification models is presented in Table 1.

Table 1:

| | Model.No. | Specifications | Sensitivity | Specificity | Precision | Recall | F1 | Accuracy | Kappa |
|---|---|---|---|---|---|---|---|---|---|
| 1 | Model 1.1.1 | KNN Multiclass full set | 0.408 | 0.704 | 0.409 | 0.703 | 0.409 | 0.408 | 0.111 |
| 2 | Model 1.1.2 | KNN Multiclass reduced set | 0.359 | 0.680 | 0.360 | 0.680 | 0.360 | 0.359 | 0.040 |
| 3 | Model 1.2.1 | KNN Binary full set | 0.571 | 0.578 | 0.569 | 0.571 | 0.570 | 0.574 | 0.149 |
| 4 | Model 1.2.2 | KNN Multiclass reduced set | 0.436 | 0.592 | 0.510 | 0.436 | 0.470 | 0.515 | 0.028 |
| 5 | Model 2.1 | Log. reg. full set | 0.525 | 0.689 | 0.622 | 0.525 | 0.569 | 0.608 | 0.214 |
| 6 | Model 2.2 | Log. reg. reduced set | 0.628 | 0.637 | 0.628 | 0.628 | 0.628 | 0.632 | 0.264 |

Finally, the third method that was employed to predict song popularity was multiple linear regression. Target feature was the normalized numeric popularity. In Model 3.1, popularity was regressed on the full set of predictors using 10-fold CV and in Model 3.2 on the reduced set using 5-fold CV. Root mean squared error (RMSE) was 0.872 in Model 3.1 and 0.856 in Model 3.2. RMSE is expressed in the same units as the response variable, which means that, since popularity has a range of almost 100 points, a deviation of 0.856 points in predictions is extremely low.

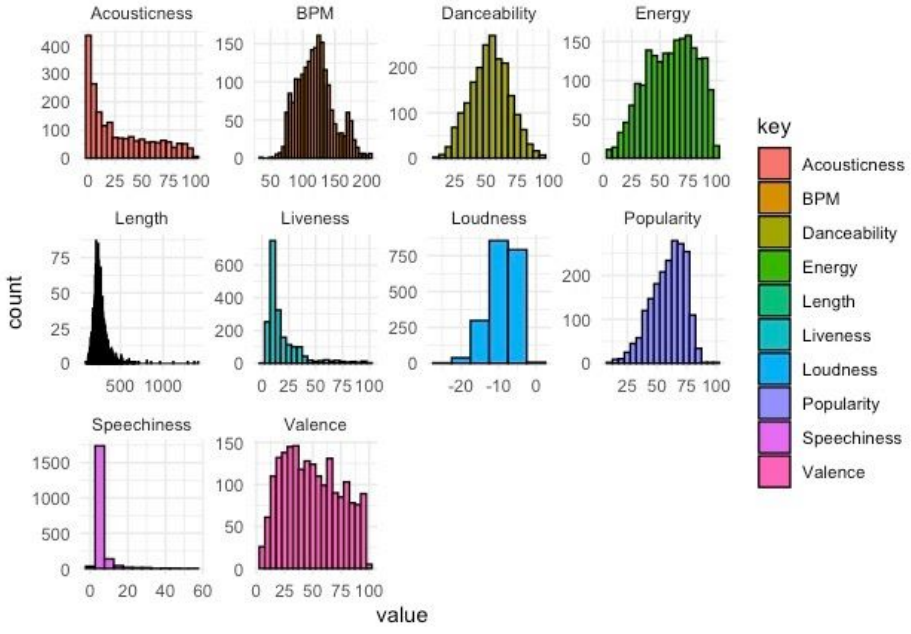
## Conclusion

This final section will present which models performed best at predicting song popularity. Generally, it is hard to compare classification and regression models, as they generate different outcomes. Therefore, all classification models were compared to one another in Graph 12. Among them, Model 2.1 was the winner with the highest level of accuracy (64%). Next, the binary classification models were compared to each other in Graph 13 featuring the same winning model as before. Among multiclass classification models, Model 1.1.1 had the highest percentage of accuracy. Finally, the two multiple linear regression models were compared with each other using their RMSEs as shown in Graph 14. Again the model with the full set of predictors was the winner.

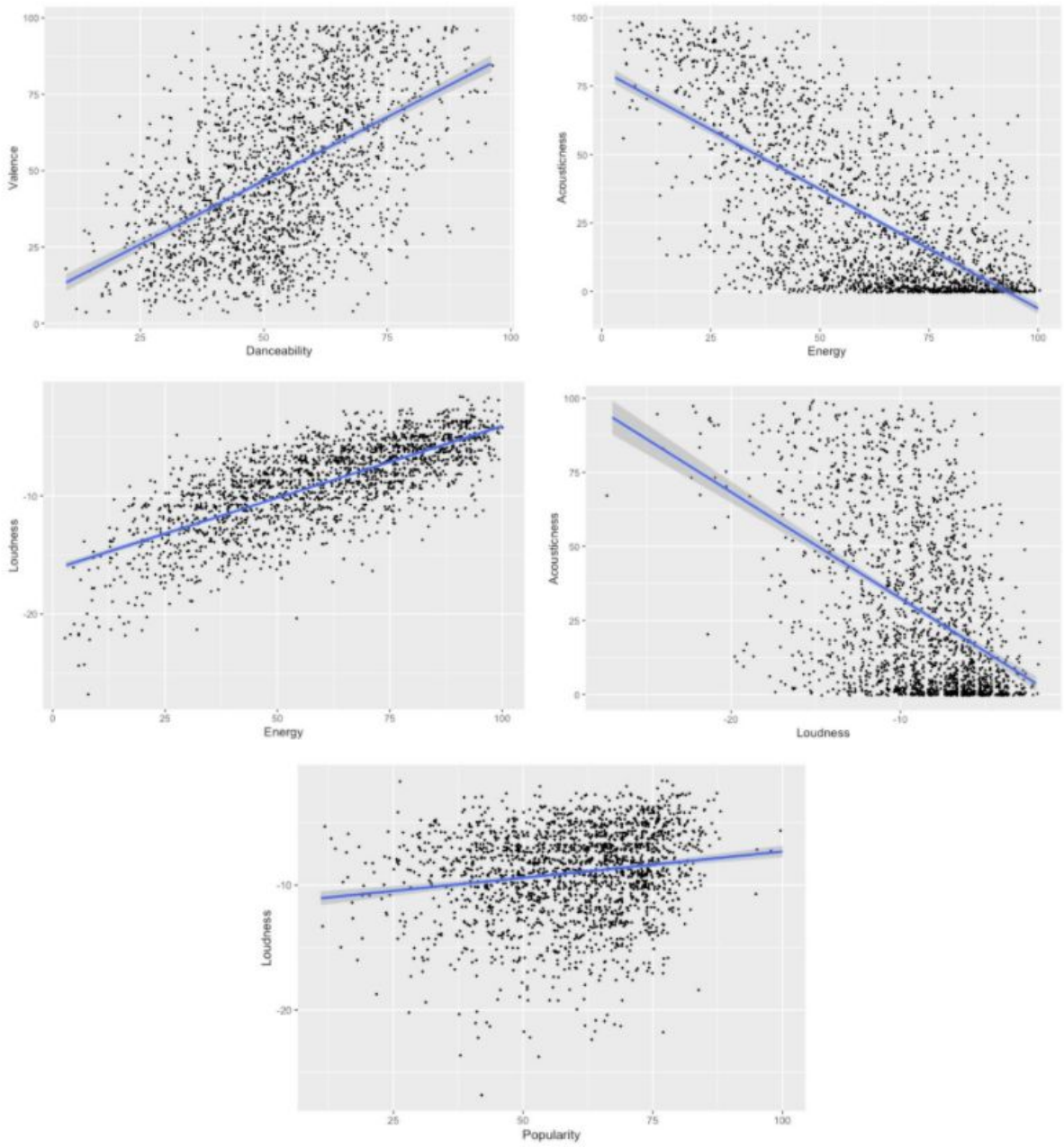
The authors conclude that the full set of features is slightly better at predicting the target in each predictive model combined with a higher number of cross-validation folds during training. Nevertheless, the winning model could only achieve a 64% classification rate, which means that roughly one out of three predictions might be wrong. Therefore, the authors suggest that other classifiers be explored in future research. The silver lining is that the RMSE of the winning linear regression model was extremely low which allows us to conclude that it is possible to predict song popularity given the predictors at hand.

# Appendix A

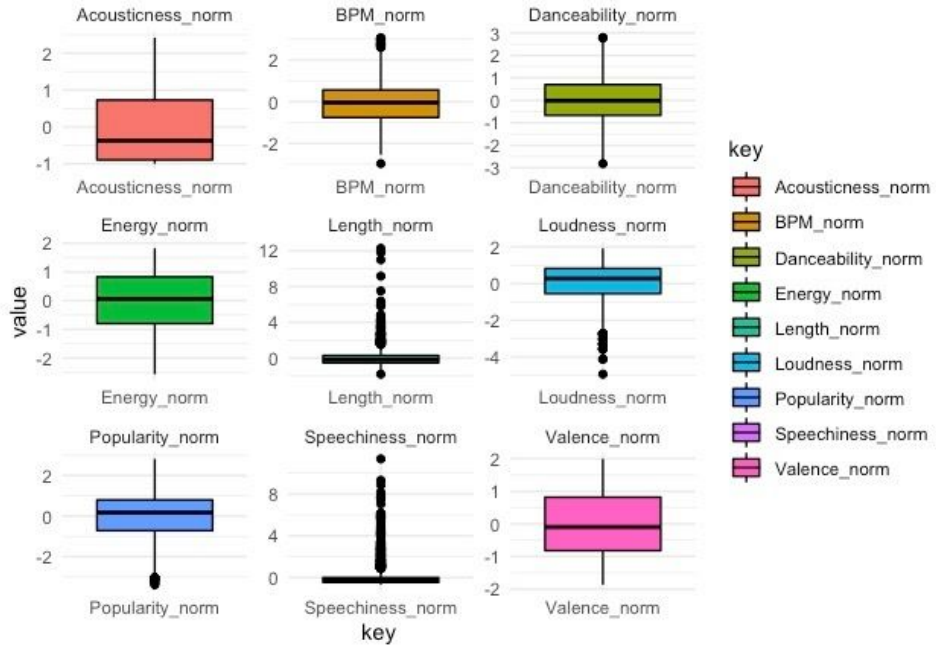
Graph 1: Frequency distribution of the the raw variables



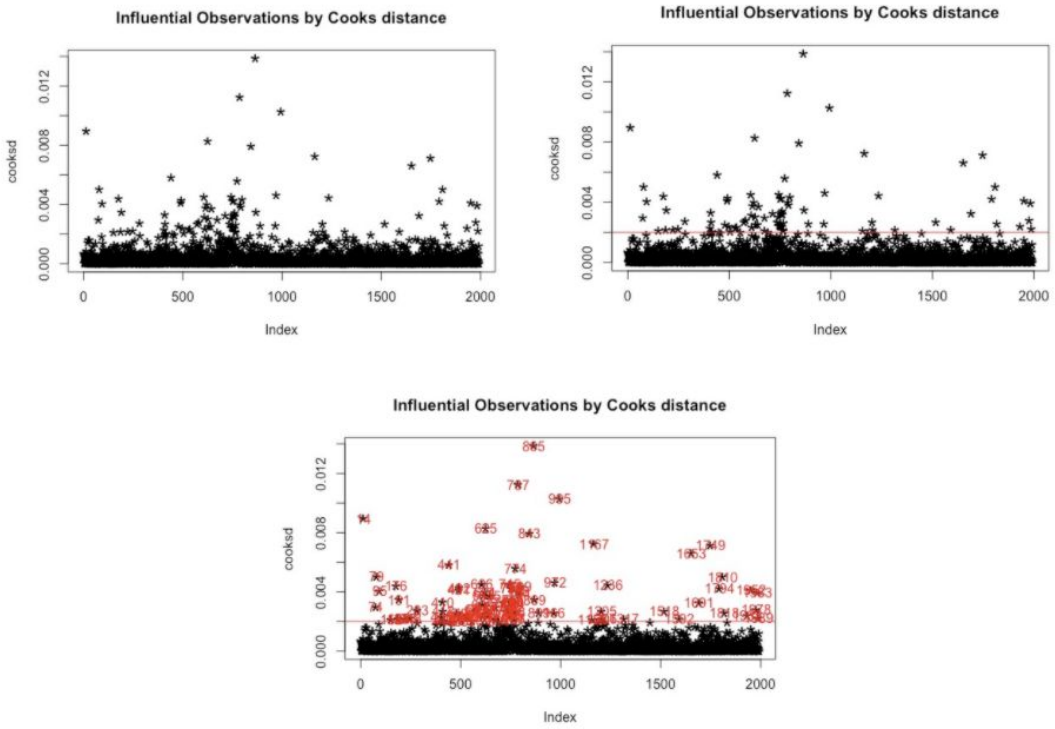
Graph 2: Correlation plots of the raw variables



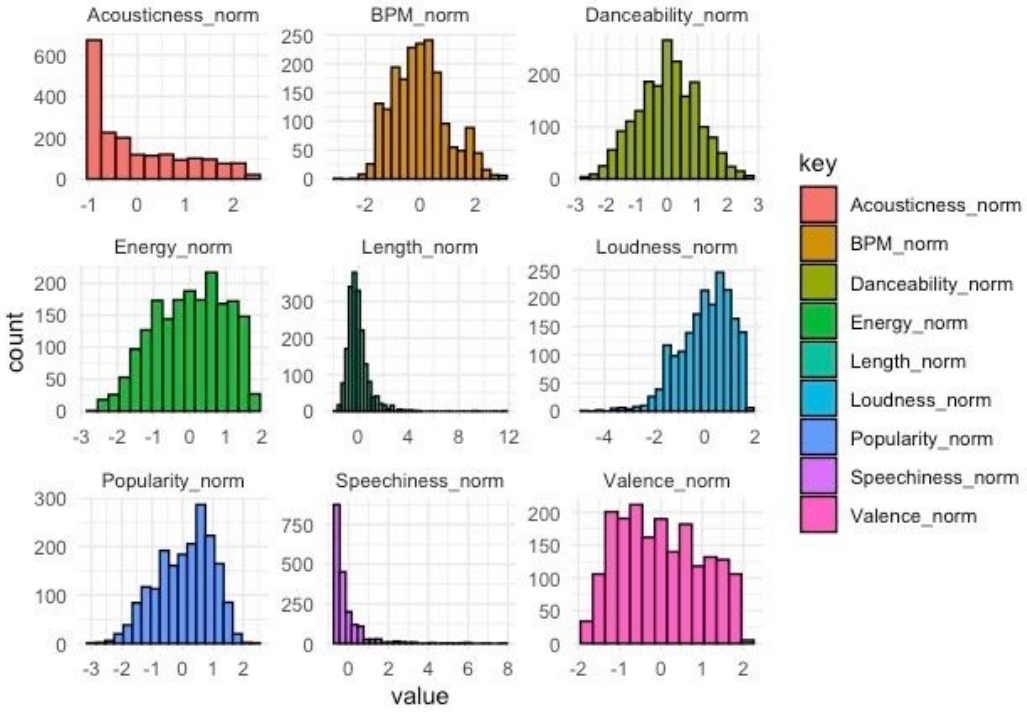
Graph 3: Outlier analysis using Boxplot Visualization



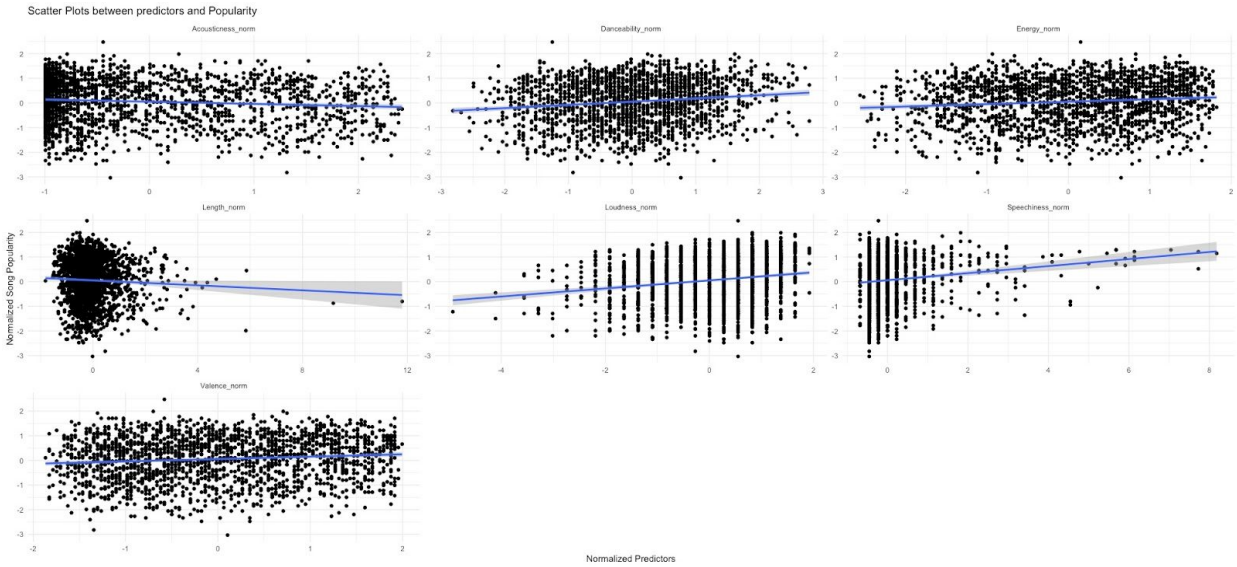

Graph 4: Identifying influential outliers using Cook's distance



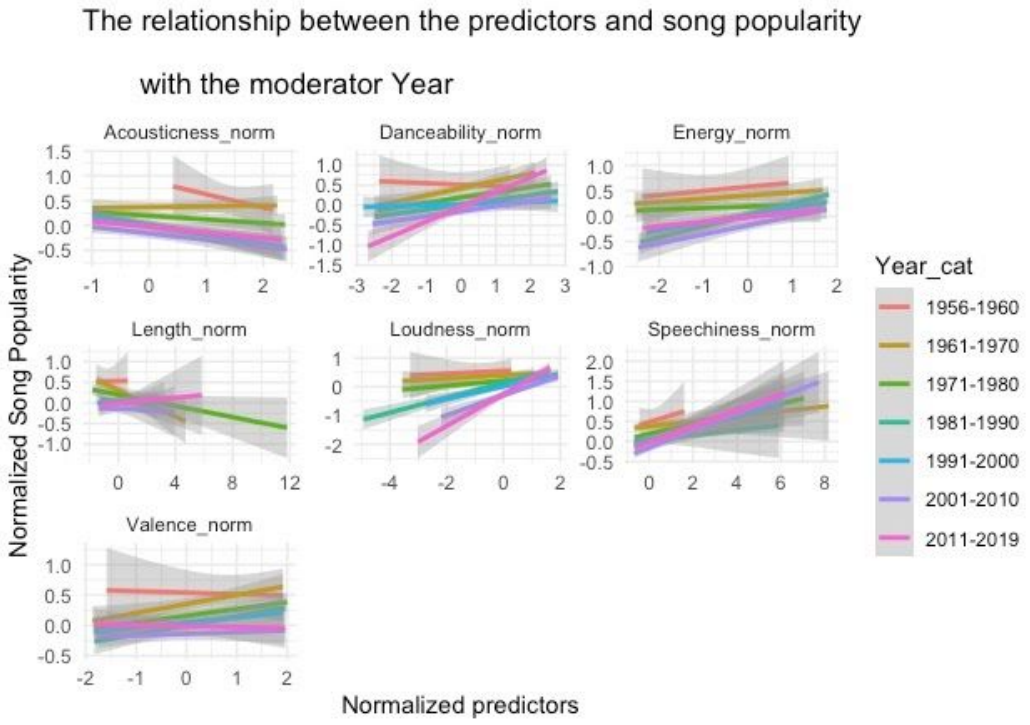
Graph 5: Histogram of normalized, outlier-free predictors



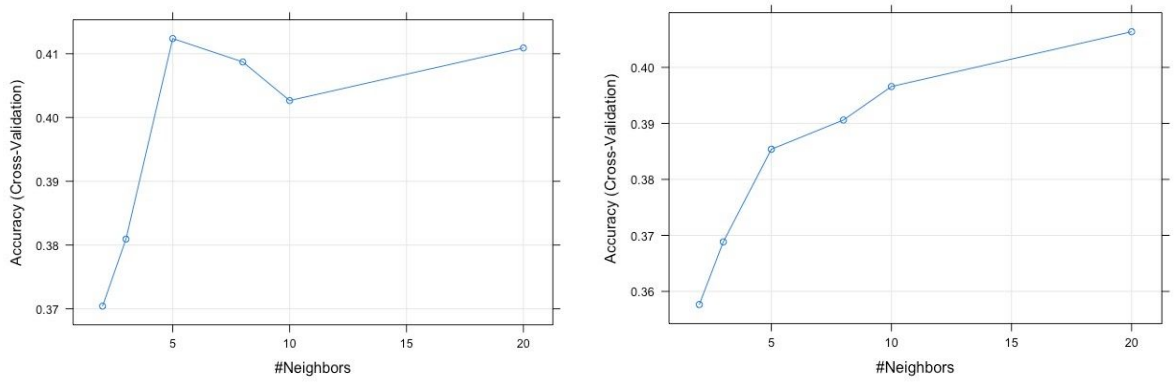

Graph 6: Correlation plots of the normalized, outlier free predictors on the normalized popularity feature



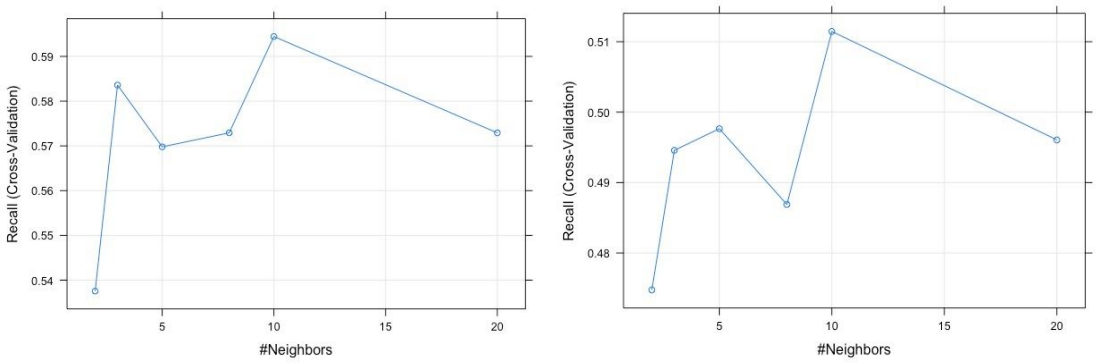
Graph 7: The relationship between the predictors and song popularity moderated by year



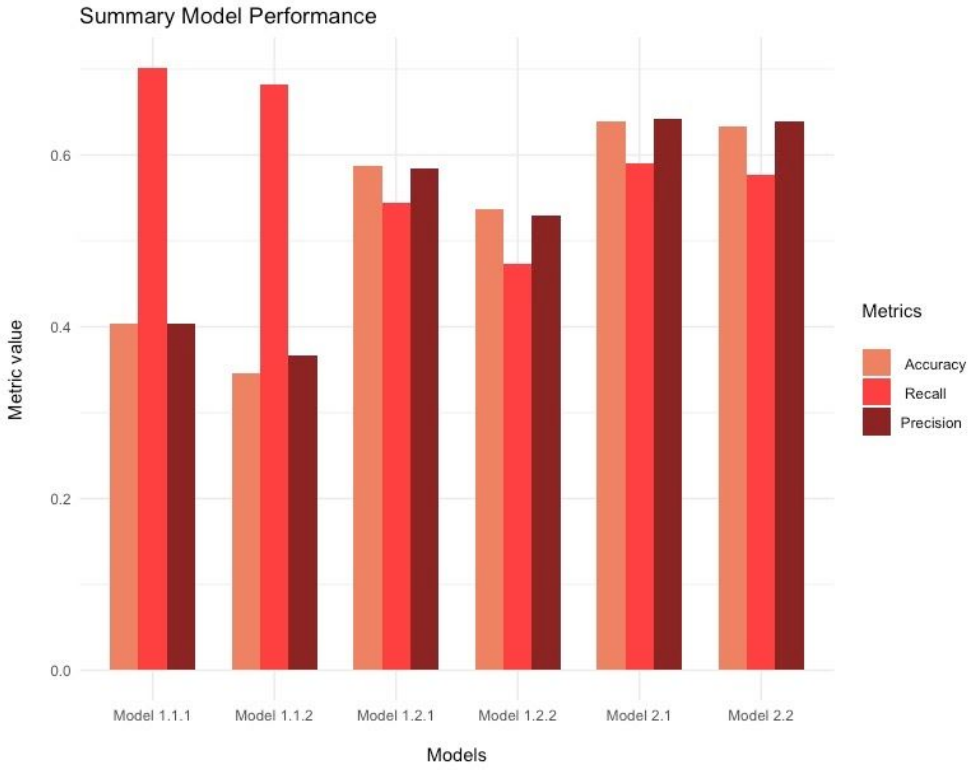

Graph 8: Accuracy results for the different numbers of k-neighbors of Model 1.1 with the full set of features (right) and reduced set of features (left).



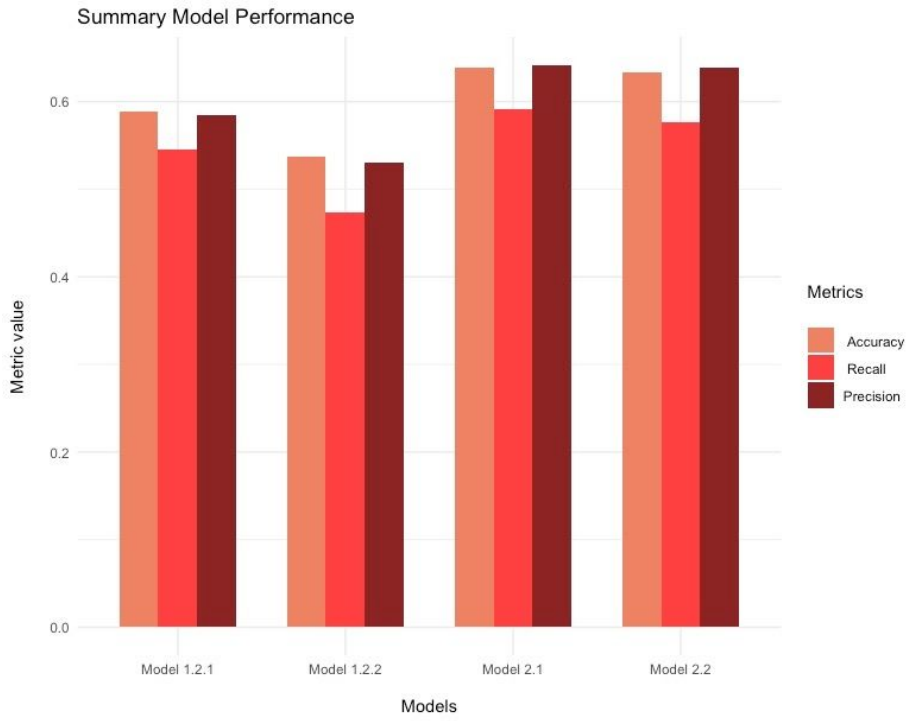

Graph 9: Accuracy results for the different numbers of k-neighbors of Model 1.2 with the full set of features (right) and reduced set of features (left).



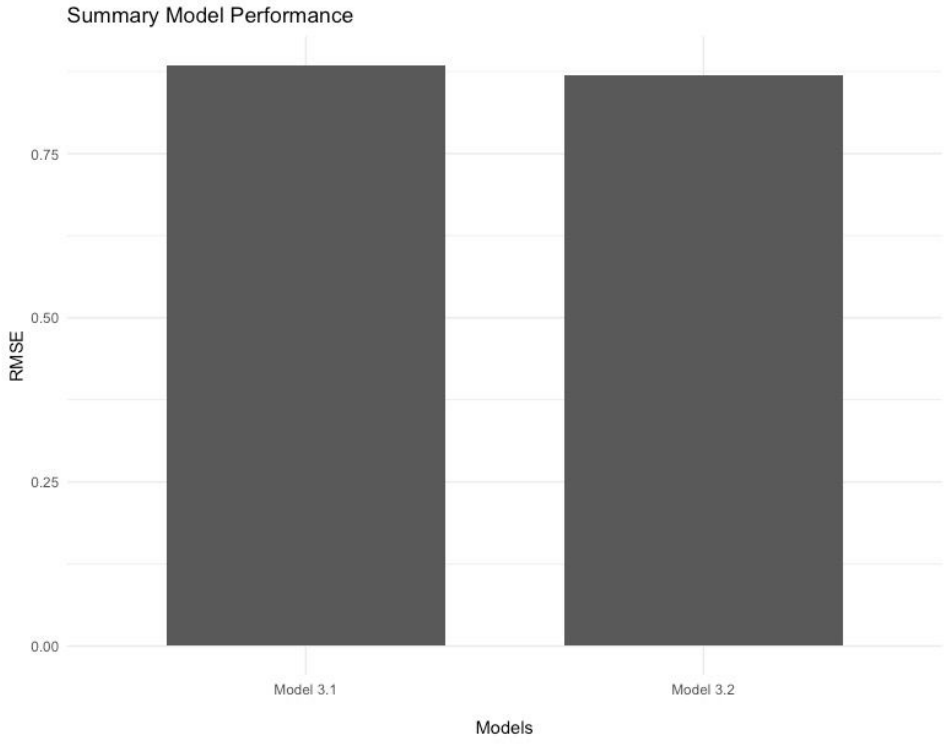

Graph 10: Comparison of classification model performance



Graph 11: Comparing performance of two classification models




Graph 12: Comparison of Model 3.1 and 3.2

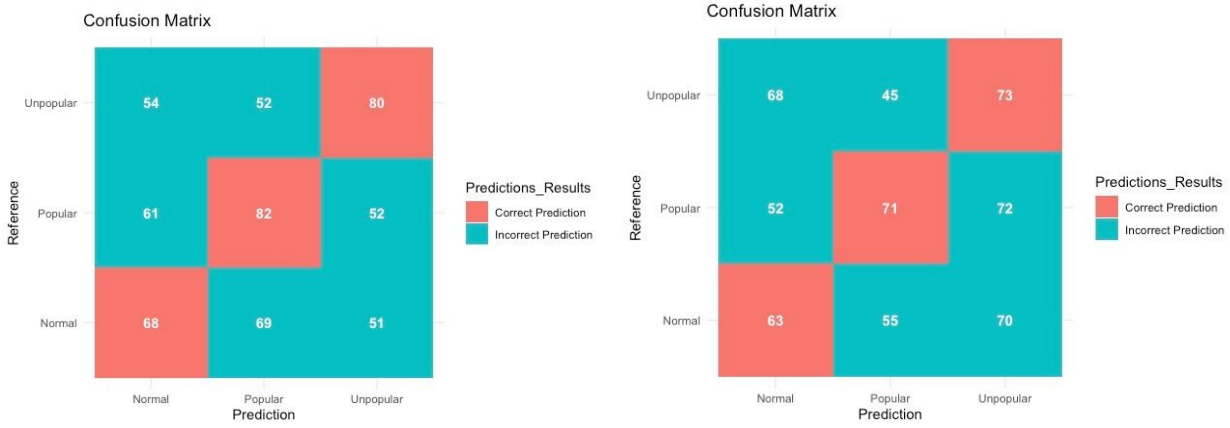


# Appendix B

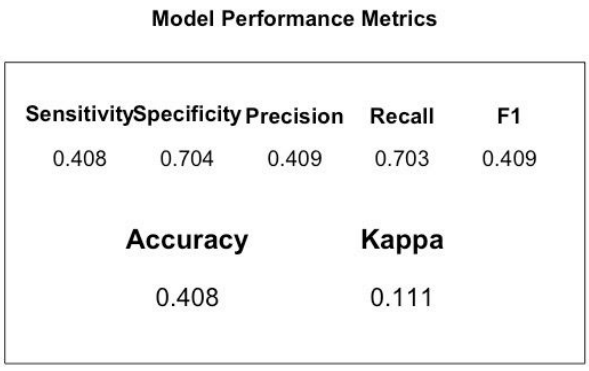
Figure 1: Multiple linear regression between popularity and the numeric predictors

```
Call:
lm(formula = Popularity_norm ~ ., data = just_norm)

Residuals:
    Min      1Q  Median      3Q     Max
-3.4806 -0.6333  0.1502  0.7516  2.5238

Coefficients:
                   Estimate Std. Error t value Pr(>|t|)
(Intercept)       -6.169e-17  2.177e-02   0.000 1.000000
Speechiness_norm   8.945e-02  2.251e-02   3.973 7.34e-05 ***
Acousticness_norm -5.301e-02  2.968e-02  -1.786 0.074227 .
Length_norm       -2.984e-02  2.287e-02  -1.305 0.192156
Valence_norm       3.455e-02  2.939e-02   1.176 0.239823
Energy_norm       -1.454e-01  4.359e-02  -3.334 0.000871 ***
Loudness_norm      2.257e-01  3.384e-02   6.669 3.33e-11 ***
Danceability_norm  1.142e-01  2.633e-02   4.339 1.50e-05 ***
BPM_norm          -1.341e-03  2.256e-02  -0.059 0.952587
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 0.9721 on 1985 degrees of freedom
Multiple R-squared:  0.05872,   Adjusted R-squared:  0.05492
F-statistic: 15.48 on 8 and 1985 DF,  p-value: < 2.2e-16
```

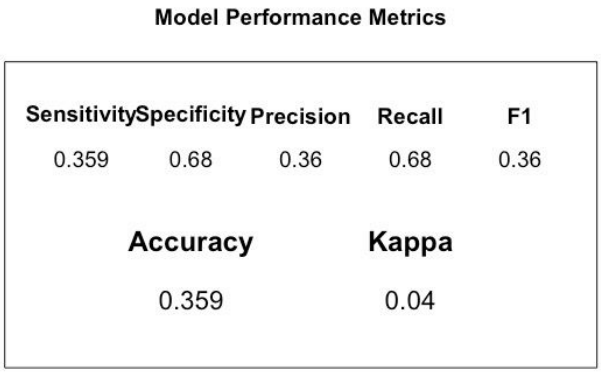
Figure 2: Confusion matrix (KNN multiclass) and metrics of Models 1.1.1 and 1.1.2 with the full set of features (right) and the reduced set of features (left)




Model 1.1.1 Model 1.1.2

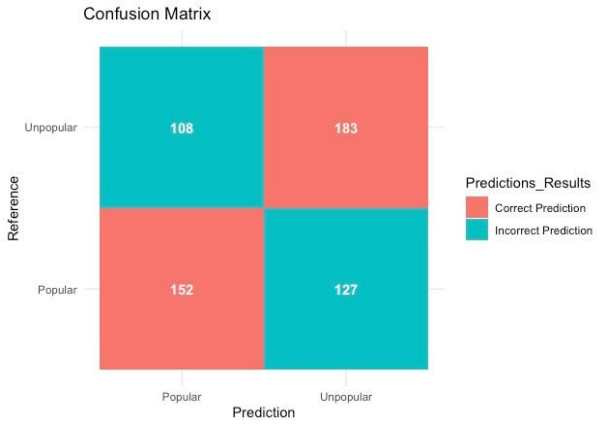



Model 1.1.1

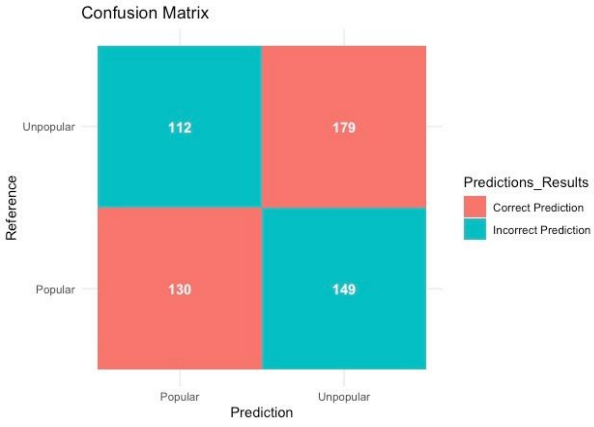



Model 1.1.2

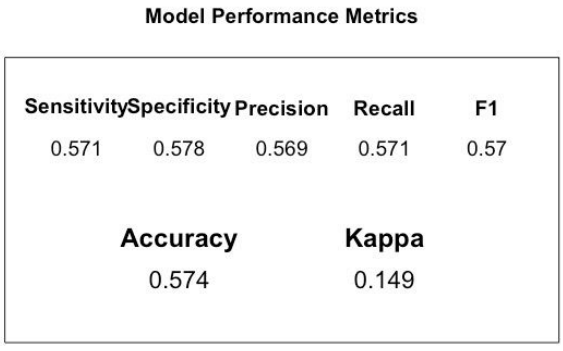
Figure 4: Confusion matrix (KNN binary) and metrics of Models 1.2.1 and 1.2.2 with the full set of features (right) and the reduced set of features (left)



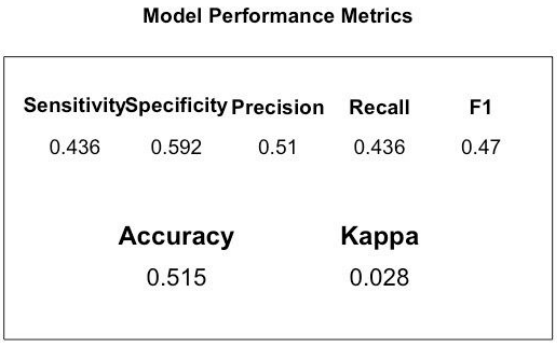

Model 1.2.1




Model 1.2.2

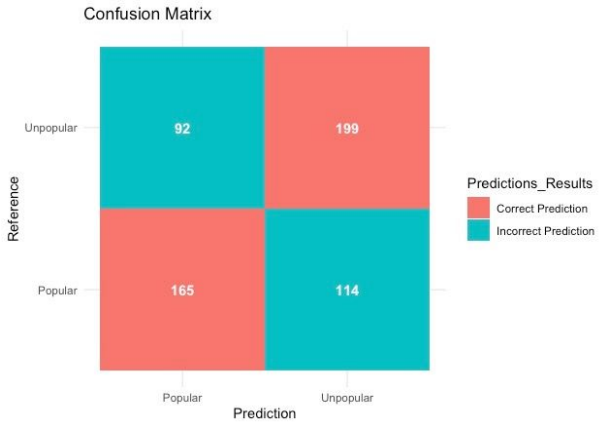



Model 1.2.1

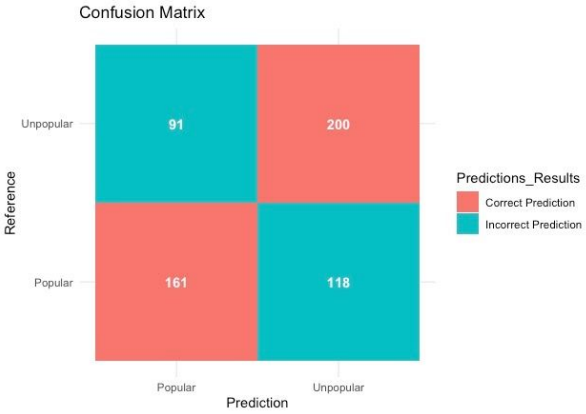



Model 1.2.2

Figure 5: Confusion matrix (Logistic regression) and metrics of Models 2.1 and 2.2 with the full set of features (right) and the reduced set of features (left)




Model 2.1




Model 2.2

**Model Performance Metrics**

| Sensitivity | Specificity | Precision | Recall | F1 |
|---|---|---|---|---|
| 0.525 | 0.689 | 0.622 | 0.525 | 0.569 |

| Accuracy | Kappa |
|---|---|
| 0.608 | 0.214 |

Model 2.1

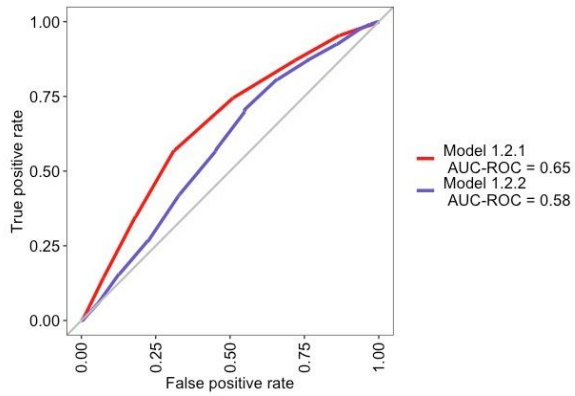
**Model Performance Metrics**

| Sensitivity | Specificity | Precision | Recall | F1 |
|---|---|---|---|---|
| 0.628 | 0.637 | 0.628 | 0.628 | 0.628 |

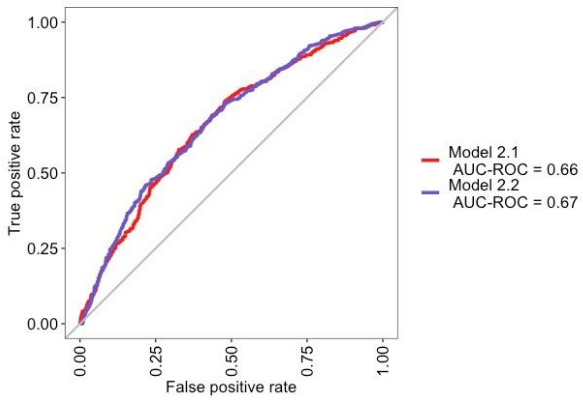
| Accuracy | Kappa |
|---|---|
| 0.632 | 0.264 |

Model 2.2

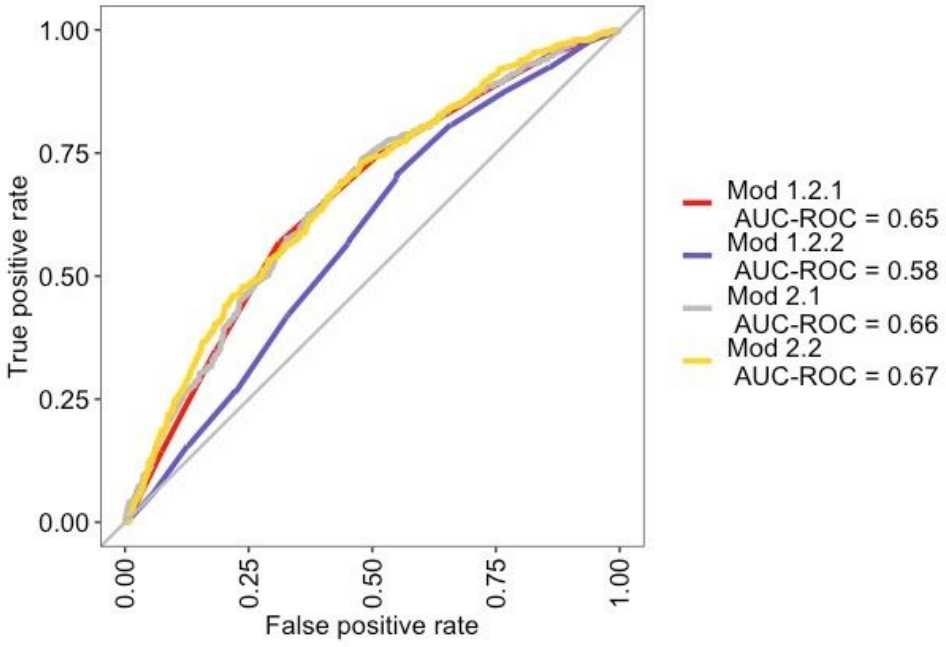
Graph 13: ROC curve for Models 1.2.1 and 1.2.2




Graph 14: ROC curve for Models 2.1 and 2.2



Graph 15: ROC curves for every binary classification model



## References

Lonsdale, Adam & North, A.. (2011). Why do we listen to music? A uses and gratifications analysis. British journal of psychology (London, England : 1953). 102. 108-34. 10.1348/000712610X506831.

Company Info. Retrieved from: https://newsroom.spotify.com/company-info/

Sumat Singh. (2020; February). Spotify - All Time Top 2000s Mega Dataset, Version 5. Retrieved in June, 2020, from https://www.kaggle.com/iamsumat/spotify-top-2000s-mega-dataset.

R Core Team (2013). R: A language and environment for statistical computing. R Foundation for Statistical Computing, Vienna, Austria. URL http://www.R-project.org/

Wickham, H., Romain, F., Henry, L., Müller, M. (2018). dplyr: A Grammar of Data Manipulation. R package version 0.7.6. https://CRAN.R-project.org/package=dplyr

Kuhn, M. (2008). Building Predictive Models in R Using the caret Package. Journal of Statistical Software, 28(5), 1 - 26. doi:http://dx.doi.org/10.18637/jss.v028.i05

Wickham, H. and Henry, L. (2018). tidyr: Easily Tidy Data with 'spread()' and 'gather()' Functions. R package version 0.8.1. https://CRAN.R-project.org/package=tidyr

Wickham H (2016). ggplot2: Elegant Graphics for Data Analysis. Springer-Verlag New York. ISBN 978-3-319-24277-4, https://ggplot2.tidyverse.org.

Yan. Y, (2016). MLmetrics: Machine Learning Evaluation Metrics. R package version 1.1.1. https://CRAN.R-project.org/package=MLmetrics