INDEX

Abstract …...………………………………………………………………………… **2**

Introduction ………………………………………………………………………. **2**

Methodology…………………………………………………………………….... **3**

Data Exploration and Preparation ……………………………………………… **3**

k-Nearest Neighbors (KNN)…………………………………..……………………… **4**

Decision Trees ……………………………..…………………………………………… **6**

Random Forest …………………………………………………………………. **8**

Support Vector Machine (Radial Kernel)……………………………………………. **10**

Other Methods: Linear Discriminant Analysis ………………………………………… **11**

Other Methods: Quadratic Discriminant Analysis …………………………………….. **11**

Conclusion ……………………………………………………………………….. **11**

Discussion and Future Research Plans ………………………………………………… **12**

Appendix: Tables **13**

Appendix: Confusion Matrices **14**

References………………………………………………………………………… **17**

Appendix (R-Code)………………………………………………………………… **19**

Abstract

In this final project a dataset containing 50k songs from Spotify separated 60% training and 40% testing was used to train k-Nearest Neighbors, Decision Trees, Random Forests, Support Vector Machine, Linear and Quadratic Discriminant Analysis models. For each model, a discussion on Accuracy, Sensitivity and Specificity is provided. After comparing all models, the best model (SVM) was selected within the context of Accuracy. Future developments pertaining this data set are also provided in the Discussion section.

Introduction

Music streaming platforms have been widely popularized during the last decade. Among its competitors, Spotify has become the platform of choice for millions of users. Spotify's appeal relays not only on its accessibility and access to millions of songs, but that it also personalizes user's experience by using machine learning algorithms that create playlists curated to the user's musical preferences. As one the greatest streaming services, classification models are widely used by to aid on the creation of genre specific musical playlists, and to categorize songs uploaded by artists.

To conduct this project, a dataset containing 50k songs from Spotify was obtained from Kaggle. In the dataset, we have a total of 50k observations and 18 possible variables. The variables included in the raw data set are given in *Table 1*.

Table 1: Variables given in music data set.

Variable	Class	Description
instance_id	num	song ID number
artist_name	chr	name of the artist
track_name	chr	name of the song
popularity	num	popularity ranking of the song
acousticness	num	how acoustic a song is
danceability	num	how danceable a song is
duration_ms	num	duration of the song in milliseconds
energy	num	level of energy of the song
instrumentalness	num	how instrumental a song is
key	chr	song's musical key (A A# B C C# D D# E F F# G G#)
liveness	num	how lively a song is
loudness	num	how loud a song is

mode	chr	mode of the song (Major or Minor)
speechiness	num	measure of talking in a song
tempo	chr	tempo of the song
obtained_date	num	date the song was obtained
valence	num	valence of the song
music_genre	chr	musical genre (Alternative, Anime, Blues, Classical, Country, Electronic, Hip-Hop, Jazz, Rap, and Rock)

The purpose of this project is to compare the accuracy of different models of classification to help categorize songs according to music_genre. This project deals with multiclass classification since the response variable contains 10 levels, each corresponding to musical genre.

For this project, confusion matrices for the models, tables, and codes can by consulted in the Appendix section.

Methodology

For this project R Studio (Version 4.1.2) was used. Additionally, the following packages and libraries were used to conduct the statistical analysis. The library “dplyr” was used to aid on data cleaning, “nortest” to conduct the Anderson-Darling Normality Test, “caret” to aid on data partitioning and to read the confusionMatrix which was pivotal on the development of this project, “class” to conduct K-Nearest Neighbors classification, “tree” for classification through Decision Trees, “randomForest” for classification using Random Forest model, and “e1071” for classification using Support Vector Machines (SVM).

Additionally, it is worth mentioning that - as this is a multiclass classification project - the confusionMatrix given by the caret package calculates Sensitivity and Specificity using a “One vs All” approach. Meaning that True Positives(TP) are all C1 instances that are classified as C1, True Negatives(TN) all non-C1 instances not classified as C-1, etc…

Data Exploration and Preparation

The data set contains 50k songs that could be classified in 10 different musical genres. From the pie-chart in *Fig. 1*, it is observed that the data set is evenly distributed among musical genres. Each musical genre contains a total of 5k songs.

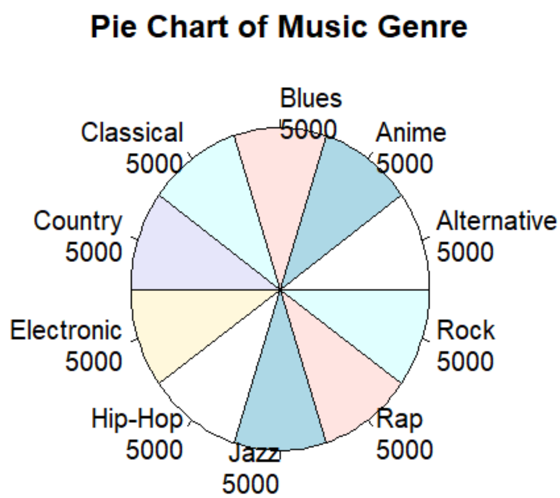

Figure 1 Pie Chart of Music Genre

As part of data preparation, some of the variables in the raw data set were recoded into classes appropriate to conduct statistical analysis. The predictor "tempo" was recoded from character to numeric and the missing values replaced by the mean, "key" recoded from character to factor, and "music_genre" recoded from character to factor to allow predictions. Additionally, "mode" was recoded from character to numeric and dummy variables were used. For dummies in "mode", songs containing a mode of Major were recoded to 1, and for Minor recoded to 0.

For the final data set the variables instance_id, artist_name, track_name, popularity, duration_ms, and obtained_date were removed using the library dplyr. As some models perform better under normalized data, numerical predictors were tested for normality using the Anderson-Darling normality test. As numerical predictors did not meet AD test, they were normalized using min-max function.

The final data set used for this project was designed to use music_genre as predictor, and acousticness, danceability, energy, instrumentalness, liveness, loudness, mode, speechiness, tempo, valence, and key as predictors.

As the final step of data preparation for this project, data was partitioned using the library caret. 60% of the data was used for training the models, and 40% of the data used for testing them. That is, 30k observations for training, and 20k for testing.

k-Nearest Neighbors (KNN)

For KNN classification, the following libraries were required: class for KNN classification, and caret to run the confusionMatrix. To prepare the dataset for KNN classification, the predictor "key" was removed from the data set as it contained 12 different levels of classification that could not be recoded into dummy variables.

For KNN classification, k-values ranging from 1 through 50 were tested and the model with the highest level of Accuracy selected. In order to test and compare each k-value, an array to store each accuracy value was created and named "acc.knn". Then, a for loop was created to run k-values 1-50. The commands coded within the for loop are: set.seed value of 1; KNN algorithm using training data, testing for validation, and k: 1-50; confusionMatrix from the caret library; and the Accuracy value from the confusionMatrix extracted and stored in acc.knn using the code *cm$overall[1]*.

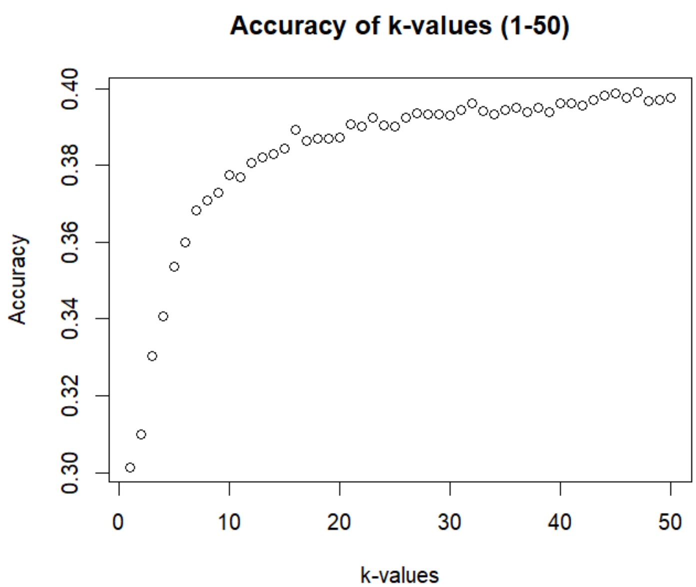

Figure 2 Accuracy plot of k-values for KNN.

For k (1-50) the model with the highest Accuracy was k=47. A scatter plot of the different Accuracy values for k 1-50 is provided in *Fig. 2*. From Figure 2, it can be observed that the Accuracy increases as the number of neighbors (k) increases. For k: 1-50, the models with the highest Accuracy are the ones with k= 30-50. However, having a higher Accuracy does not mean that you will have a more accurate model as you increase k. For this scatter plot, the model with the highest Accuracy was the one that contained k=47 with an Accuracy of 39.89%, and then Accuracy starts fluctuating in a decreasing manner. A table containing the Accuracy calculated for k values 1 through 50 can be found in the Appendix of this study.

After finding the best *k* for k-values 1-50, a KNN model was fitted using the same set.seed value and k=47. Within its confusion matrix, the sensitivity and specificity was calculated for each class. Considering Sensitivity as the measure in which the predicted genre matched the correct genre, and Specificity as a measure of the model to differentiate the genre (E.g Specificity for Alternative are those songs that were not predicted to be Alternative and are not Alternative).

```
> #getting sensitivity and specificity by class
> cm.best$byClass[1:10,1:2]
                    Sensitivity Specificity
Class: Alternative       0.2260   0.9286667
Class: Anime             0.2860   0.9457222
Class: Blues             0.2880   0.9416667
Class: Classical         0.8515   0.9648333
Class: Country           0.5835   0.8588333
Class: Electronic        0.4910   0.9524444
Class: Hip-Hop           0.4485   0.9220000
Class: Jazz              0.3965   0.9405000
Class: Rap               0.2870   0.9311667
Class: Rock              0.1310   0.9462778
```

Figure 3 Sensitivity and Specificity by Genre for KNN

The Specificity and Sensitivity by class is given in *Figure 3*. Looking at the calculations for Sensitivity in Figure 3, we can observe that the KNN model (k=47) performed the best at predicting the genre "Classical" as KNN predicted Classical songs correctly 85% of the time. This can be further confirmed as KNN correctly predicted 1703 out of 2000 classical songs. However, for the remaining genres, KNN did not perform well as the sensitivity measures for Rock, Rap, Jazz, Hip-Hop, Blues, Anime, and Alternative range from 13%-45%, and 50% to 60% at best for Electronic and Country.

For Specificity across genres, the measurements ranged from 85% to 95%. The genre with the lowest Specificity was Country. Country's low specificity can be further confirmed by the confusion matrix as our KNN model confused country the most with Rock (670 rock songs misclassified as Country) and Blues (442 Blues songs that were misclassified as Country).

Decision Trees

A decision tree model was also fitted using the library tree. First, a decision tree was fitted using music_genre was used as a response, and all the variables in the training data set were used as predictors. Upon calling the summary function it is learned that the tree contained 8 terminal nodes, and that the variables that were actually used in the the tree construction were acousticness, instrumentalness, speechiness, danceability, energy, and loudness. Further exploration of the decision tree can be observed in *Fig. 4* .

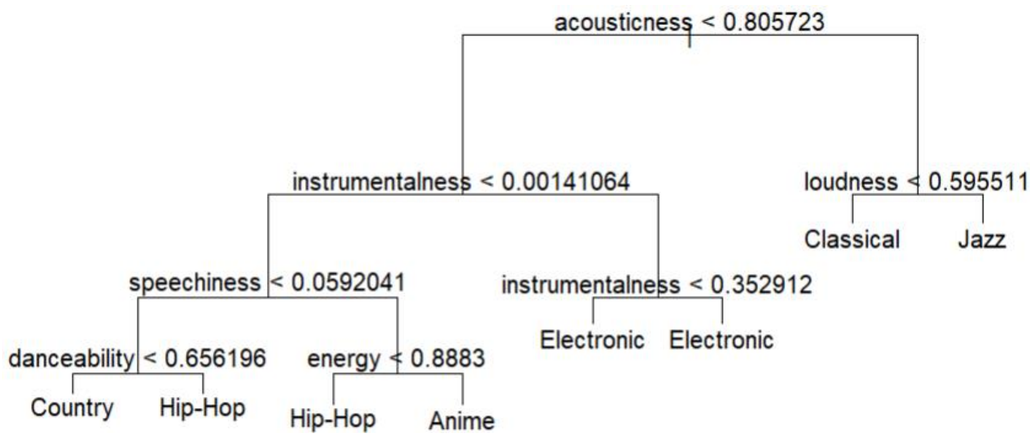

Figure 4 Plot of Decision Tree

In the plot of the decision tree, it can be observed how each genre was classified. Starting from acousticness, if a song has an acousticness higher than 0.81, the decision will be based on loudness, songs with a loudness higher than 0.60 will be classified as Jazz, if loudness is smaller than 0.60 then they will be classified as Classical. For songs with low acousticness, the decision will be based on instrumentalness. Songs with high instrumentalness will be classified as Electronic, and for songs with low instrumentalness the decision will be narrowed based on Speechiness. For songs high in speechiness, the decision will be narrowed to energy. For songs with an energy level higher than 0.88 will be classified as anime, and songs lower than said value will be classified as Hip-Hop. Songs with low speechiness, the decision will be narrowed down in terms of danceability. Songs high in danceability will be classified as Hip Hop, and songs low in danceability will be classified as Country.

In an attempt to find a better fit of the tree, K-fold validation was used. However, the tree with the greatest Accuracy resulted to be the original model, so it was kept to make predictions and be compared to the different models tested in this project.

After predictions were made, the Decision Tree had only a 32.52% of Accuracy according to the confusionMatrix.

Analyzing the Specificity and Sensitivity values given by the confusion matrix in *Fig 5*, it is learned that the decision tree performed the best at predicting Hip-Hop

```
> #getting sensitivity and specificity by class
> tree.cm$byClass[1:10,1:2]
                   Sensitivity Specificity
Class: Alternative      0.0000   1.0000000
Class: Anime            0.1285   0.9668889
Class: Blues            0.0000   1.0000000
Class: Classical        0.7300   0.9753889
Class: Country          0.6370   0.7973333
Class: Electronic       0.7660   0.7540556
Class: Hip-Hop          0.8155   0.8071111
Class: Jazz             0.1760   0.9495556
Class: Rap              0.0000   1.0000000
Class: Rock             0.0000   1.0000000
```

Figure 5 Sensitivity and Specificity by Genre for Decision Tree

songs as it predicted 81.55% of them correctly. Similarly, the decision tree performed well at predicting Electronic and Classical songs with (0.7660 and 0.7300) of Sensitivity. However, the decision tree was not able to predict any of the Alternative, Blues, Rap, and Rock songs. The highest Specificity across genres was once again Classical songs (0.9753).

Overall, the decision tree lacked skill to distinguish musical genres. By the confusion matrix it can be confirmed that the decision tree did not make predictions for Alternative, Blues, Rap, and Rock genres.

Random Forest

To fit the random forest, the library randomForest was used. Similar to KNN, in order to pick the best Random Forest model, a for loop that allowed to test different mtry values and store their Out-of-Bag (OOB) error in an array was coded and the Random Forest with the lowest OOB selected. Within the array, the random forest model used music_genre as response, and all predictors within the training data set, the mtry values used were 1 through 11, and contained 1000 trees.

Mtry	Out-of-Bag Error (OOB)
1	0.5933333
2	0.5800000
3	0.5779667
4	0.5800667
5	0.5810333
6	0.5804333
7	0.5812333
8	0.5828000
9	0.5813667
10	0.5853333
11	0.5856333

Table 2 OOB errors for Random Forest

From mtry values 1-11 the Random Forest with the lowest OOB error was the one with mtry = 3. The OOB errors for each mtry value can be visualized in *Table 2.*

After selecting the model with the lowest OOB error, a random forest was fit specifying mtry=3. After fitting for the best Random Forest model, predictions were made using the model and testing data. Additionally, a variable importance plot was obtained in *Fig. 6*

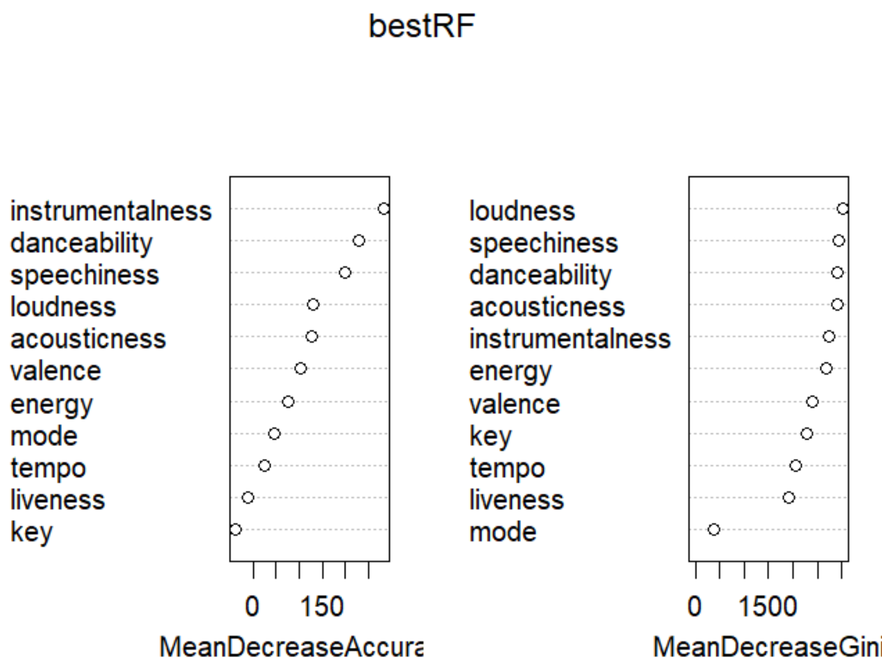

Figure 6 Variable Importance Plot for Random Forest

From the variable importance plot for the Random Forest, it can be observed that instrumentalness, danceability, speechiness, loudness, and acousticness were ranked similarly according to MeanDecreaseAccuracy and MeanDecreaseGini.

After making predictions on the Random Forest Model, the confusionMatrix was used to estimate the model's Accuracy, and the Sensitivity and Specificity according to each class. Overall, the model had an accuracy of 41.57%.

```
> pRF$byClass[1:10,1:2]
                    Sensitivity Specificity
Class: Alternative       0.2000   0.9302778
Class: Anime             0.3640   0.9565000
Class: Blues             0.4215   0.9276667
Class: Classical         0.8385   0.9756667
Class: Country           0.5745   0.9117778
Class: Electronic        0.5985   0.9411111
Class: Hip-Hop           0.3865   0.9042778
Class: Jazz              0.4665   0.9345556
Class: Rap               0.2100   0.9240556
Class: Rock              0.0970   0.9448889
```

Figure 7 Sensitivity and Specificity by Genre for Random Forest

Looking at the Sensitivity and Specificity by class in *Fig 7*, the Random Forest model performed significantly better on classifying Classical songs compared to the rest of the genres. Classical songs had 83.85% of Sensitivity, while the rest of the genres Sensitivity ranged from 9.70% as the lowest (Rock) to 59.85% (Electronic) as the second best. All Specificity values were among 90%.

Examining the confusion matrix for the random forest, all genres had the highest number of classifications in the correct genre except for Hip-Hop and Rap as the Random Forest model struggled differentiating among those two genres.

Support Vector Machine (Radial Kernel)

A Support Vector Machine(SVM) model was fitted thanks to the library e1071. For the SVM, music_genre was selected as the response, a radial kernel used, and the data obtained from the training data set. Initially, a cross validation approach using different values for cost and gamma was attempted. However, after hours of run-time the computation was interrupted and could not be completed. Because of said issue, for the SVM model a cost of 10, and gamma of 0.1 was chosen to researcher's random preference. The SVM model had a total of 26422 support vectors. After the SVM model was trained, predictions were made using the testing data set. Looking at the confusionMatrix, the SVM model had a 41.75% of Accuracy.

```
> svm.cm$byClass[1:10,1:2]
                    Sensitivity Specificity
Class: Alternative       0.2335   0.9259444
Class: Anime             0.3415   0.9456111
Class: Blues             0.3475   0.9369444
Class: Classical         0.8300   0.9753889
Class: Country           0.5660   0.8998333
Class: Electronic        0.5385   0.9477222
Class: Hip-Hop           0.4135   0.9182222
Class: Jazz              0.4425   0.9381667
Class: Rap               0.3150   0.9260556
Class: Rock              0.1475   0.9389444
```

Figure 8 Sensitivity and Specificity by Genre for SVM

Looking at the Sensitivity and Specificity by class in *Fig 8*, the SVM model had the highest Sensitivity (83%) at predicting Classical music. Similarly, the model had the second highest specificity when predicting Country music. All Specificity values are among 90%.

Looking at the confusionMatrix the SVM model predicted most of the songs to be Country music. Looking at the predictions, it classified almost 30% of the data as Country.

Other classification methods

Although not the focus but also worth mentioning for this project, Linear Discriminant Analysis (LDA), and Quadratic Discriminant Analysis (QDA) models were also fitted using the MASS library. For both models, music_genre was used as a response, and the remaining variables in the training data sets as predictors. Details surrounding their Accuracy, Sensitivity, and Specificity are given below.

Other methods: Linear Discriminant Analysis (LDA)

```
> lda.cm$byClass[1:10,1:2]
                    Sensitivity Specificity
Class: Alternative       0.2880   0.8943333
Class: Anime             0.1360   0.9553889
Class: Blues             0.3340   0.9170000
Class: Classical         0.8165   0.9695556
Class: Country           0.4515   0.8950556
Class: Electronic        0.4700   0.9504444
Class: Hip-Hop           0.4685   0.9205000
Class: Jazz              0.3205   0.9483889
Class: Rap               0.2455   0.9460000
Class: Rock              0.2420   0.9113889
```

Figure 9 Sensitivity and Specificity by Genre for LDA

For the LDA model, the overall Accuracy was 37.72%. The highest Sensitivity and Specificity tells us that the model was best at predicting Classical music compared to the rest of the genres as it had measures of 81.65% and 97% respectively. This can be further confirmed by examining the confusion matrix. The LDA model made perfect distinctions between Classical music and Hip-Hop/Rap as none of the testing data corresponding to classical music were classified as Hip-Hop or Rap.

Other methods: Quadratic Discriminant Analysis (QDA)

```
> qda.cm$byClass[1:10,1:2]
                    Sensitivity Specificity
Class: Alternative       0.1020   0.9604444
Class: Anime             0.1190   0.9667778
Class: Blues             0.2365   0.9463333
Class: Classical         0.8115   0.9582222
Class: Country           0.7055   0.7848889
Class: Electronic        0.3785   0.9665556
Class: Hip-Hop           0.5435   0.8794444
Class: Jazz              0.2830   0.9595000
Class: Rap               0.2190   0.9199444
Class: Rock              0.1195   0.9376667
```

Figure 10 Sensitivity and Specificity by Genre for QDA

For the LDA model the Accuracy was 35.18%. Looking at the Sensitivity values, the model performed best at predicting Classical (81.15%) and Country (70.55%) music. All Specificity values also range among 90%.

Conclusions

Within the context of Accuracy, Support Vector Machines (SVM) performed the best with an Accuracy of 41.75%. A close second is the Random Forest model that had an Accuracy of 41.57%, and the third best performing model corresponds to KNN using a k=47, having 39.89% of Accuracy. The lowest performing models were LDA, QDA, and Decision Tree. A ranking of the best to worst performing model is given in *Table 3*

Model	Accuracy
SVM (radial)	41.75%
Random Forest	41.57%
KNN (k=47)	39.89%
LDA	37.72%
QDA	35.18%
Decision Tree	32.52%

Table 3 Model Performance

Almost all of the models (SVM, Random Forest, KNN, LDA, QDA) performed best at classifying Classical music correctly. However, the Decision Tree performed better at classifying Hip-Hop music. Some theories on the agreement of classical music being the best classified among models are that Classical music is inherently different to the rest of the music as it uses different instruments not used in mainstream music that could have an effect on the measurements given by the predictors (loudness, speechiness, etc). Also, the decision tree shows a better breakdown at how each song is classified.

Discussion and Future Research Plans

Looking at all the classification models used in this project, SVM was the highest model as it had an Accuracy of 41.75%. Although it is progress, it is not the best as it doesn't even guarantee 50% of Accuracy. This does not come as a surprise as Spotify has confirmed that for their classification models they use Convolutional Neural Networks (CNN). In said algorithm, the problem is 'simplified' by analyzing songs out of their Mel Spectogram instead of analyzing songs' speechiness, loudness, key, instrumentalness, etc… CNN models permit to have accurate classifications out of images.

Further developments for the improvement of this classification models are comparing model's performance for non- normalized data. Also, evaluating predictors to make sure only the subset of most relevant predictors is used.

Additionally, one of the most important developments for this study are pertaining model performance. As complexity increases with classes, ROC curves and AUC values suited for multiclass classification can be incorporated in the future.

APPENDIX

Tables

Table 4: Accuracy Values for k 1 through 50.

k	Accuracy	k	Accuracy
1	0.30145	**26**	0.39250
2	0.30990	**27**	0.39370
3	0.33035	**28**	0.39330
4	0.34065	**29**	0.39330
5	0.35360	**30**	0.39310
6	0.36005	**31**	0.39440
7	0.36830	**32**	0.39605
8	0.37085	**33**	0.39420
9	0.37300	**34**	0.39320
10	0.37740	**35**	0.39440
11	0.37700	**36**	0.39490
12	0.38055	**37**	0.39385
13	0.38195	**38**	0.39505
14	0.38300	**39**	0.39385
15	0.38450	**40**	0.39605
16	0.38920	**41**	0.39615
17	0.38640	**42**	0.39550
18	0.38695	**43**	0.39695
19	0.38710	**44**	0.39810
20	0.38730	**45**	0.39880
21	0.39080	**46**	0.39750
22	0.39010	**47**	0.39890
23	0.39245	**48**	0.39670
24	0.39050	**49**	0.39690
25	0.39020	**50**	0.39760

Confusion Matrices (CM)

KNN CM

```
> cm.best
Confusion Matrix and Statistics

             Reference
Prediction    Alternative Anime Blues Classical Country Electronic Hip-Hop Jazz  Rap Rock
  Alternative         452   224   173        60     109        158      64   59  129  308
  Anime               228   572   137        52      73        149      13   68   61  196
  Blues               120    73   576        29     199         38      52  242   49  248
  Classical             9   306    37      1703       9         10       0  240    1   21
  Country             396   371   442        21    1167        118     136  197  190  670
  Electronic          139   165    80        27      11        982      69  194   75   96
  Hip-Hop             190    16    44         1      73        118     897   94  832   36
  Jazz                122   149   264        85      76        198      40  793   32  105
  Rap                 158    21    40         1      68        151     677   65  574   58
  Rock                186   103   207        21     215         78      52   48   57  262

Overall Statistics

               Accuracy : 0.3989
```

Figure 11 Confusion Matrix of KNN

Decision Tree CM

```
> tree.cm
Confusion Matrix and Statistics

             Reference
Prediction    Alternative Anime Blues Classical Country Electronic Hip-Hop Jazz  Rap Rock
  Alternative           0     0     0         0       0          0       0    0    0    0
  Anime               166   257    57        22      34         73      72   10   76   86
  Blues                 0     0     0         0       0          0       0    0    0    0
  Classical             3   173    38      1460      12          6       1  199    1   10
  Country             695   649   568        69    1274        181      98  235  203  950
  Electronic          631   610   861       188     179       1532     183  986  199  590
  Hip-Hop             442   107   303         6     409        178    1631  218 1510  299
  Jazz                 63   204   173       255      92         30      15  352   11   65
  Rap                   0     0     0         0       0          0       0    0    0    0
  Rock                  0     0     0         0       0          0       0    0    0    0

Overall Statistics

               Accuracy : 0.3253
```

Figure 12 Confusion Matrix of Decision Tree

Random Forest CM

```
> pRF
Confusion Matrix and Statistics

             Reference
Prediction    Alternative Anime Blues Classical Country Electronic Hip-Hop Jazz  Rap Rock
  Alternative         400   177   140        61      89        104      86   57  139  402
  Anime               164   728    95        67      89        121       7   61   32  147
  Blues               166    92   843        28     208         89      28  302   27  362
  Classical             3   230    19      1677       5         11       0  158    1   11
  Country             248   312   245         9    1149         45      43   81   75  530
  Electronic          192   167    96        33      15       1197      96  246   96  119
  Hip-Hop             224    18    52         2      93         83     773   88 1118   45
  Jazz                129   157   288       109      69        246      41  933   32  107
  Rap                 162    25    38         0      75         51     896   37  420   83
  Rock                312    94   184        14     208         53      30   37   60  194

Overall Statistics

               Accuracy : 0.4157
```

Figure 13 Confusion Matrix of Random Forest

SVM CM

```
> svm.cm
Confusion Matrix and Statistics

             Reference
Prediction    Alternative Anime Blues Classical Country Electronic Hip-Hop Jazz  Rap Rock
  Alternative         467   217   186        60     115        128      78   86  134  329
  Anime               217   683   101        88     101        137      10   93   48  184
  Blues               135   103   695        34     185         67      38  239   33  301
  Classical             3   220    23      1660       8         11       0  162    2   14
  Country             310   292   316        16    1132         65      51  114   78  561
  Electronic          147   157    83        23      17       1077     101  214   95  104
  Hip-Hop             197    14    47         1      73        114     827  103  887   36
  Jazz                118   150   286        96      65        195      53  885   39  111
  Rap                 164    20    39         1      75        115     805   47  630   65
  Rock                242   144   224        21     229         91      37   57   54  295

Overall Statistics

               Accuracy : 0.4175
```

Figure 14 Confusion Matrix of SVM

LDA CM

```
> lda.cm
Confusion Matrix and Statistics

             Reference
Prediction    Alternative Anime Blues Classical Country Electronic Hip-Hop Jazz  Rap Rock
  Alternative         576   381   164        46     212        249     162   64  241  383
  Anime               166   272   192        58      35         98       8  104   14  128
  Blues               134   130   668        64     302         78      52  329   58  347
  Classical             5   262    29      1633       6         21       0  213    0   12
  Country             277   275   325        25     903         75     131  198  154  429
  Electronic          123   166    43        33      18        940      95  228  106   80
  Hip-Hop             172    17    36         9      55        170     937  120  826   26
  Jazz                114   186   196       108      36        132      40  641   37   80
  Rap                 114    22    28         1      42        137     549   48  491   31
  Rock                319   289   319        23     391        100      26   55   73  484

Overall Statistics

               Accuracy : 0.3772
```

Figure 15 Confusion Matrix of LDA

QDA CM

```
> qda.cm
Confusion Matrix and Statistics

             Reference
Prediction    Alternative Anime Blues Classical Country Electronic Hip-Hop Jazz  Rap Rock
  Alternative         204   174   105        30      44         77      45   41   68  128
  Anime                73   238    95        99       5        136       3  117   10   60
  Blues               128    81   473        46     116         77      53  211   48  206
  Classical            31   300    55      1623      27         24       3  265    3   44
  Country             626   592   693        47    1411        183     193  303  278  957
  Electronic           90   132    44        15       6        757      32  201   26   56
  Hip-Hop             267    75   101         5     160        278    1087  137 1026  121
  Jazz                 72   125   131        97      32        158      38  566   31   45
  Rap                 245   136    69         7     117        151     510   62  438  144
  Rock                264   147   234        31      82        159      36   97   72  239

Overall Statistics

               Accuracy : 0.3518
```

Figure 16 Confusion Matrix of QDA

References

Ambika, Anjana. "Performance Evaluation Measures of Classification Model." *Analytics Vidhya*, Analytics Vidhya, 18 Dec. 2020, www.analyticsvidhya.com/blog/2020/12/decluttering-the-performance-measures-of-classification-models.

"Anderson-Darling Normality Test." *Variation.Com*, Taylor Enterprises, variation.com/wp-content/distribution_analyzer_help/hs140.htm. Accessed 8 Dec. 2021.

Brownlee, Jason. "How to Evaluate Machine Learning Algorithms with R." *Machine Learning Mastery*, 12 Dec. 2019, machinelearningmastery.com/evaluate-machine-learning-algorithms-with-r.

Hoare, Jake. "How Is Variable Importance Calculated for a Random Forest?" *Displayr*, 9 June 2021, www.displayr.com/how-is-variable-importance-calculated-for-a-random-forest.

"How Do You Measure Specificity and Sensitivity in a Multiple Class Classification Problem?" *Research Gate*, 22 Apr. 2015, www.researchgate.net/post/How-do-you-measure-specificity-and-sensitivity-in-a-multiple-class-classification-problem.

Josh, Starmer. "ROC and AUC in R." *YouTube*, uploaded by StatQuest with Josh Starmer, 18 Dec. 2018, www.youtube.com/watch?v=qcvAqAH60Yw.

Narkhede, Sarang. "Understanding AUC - ROC Curve - Towards Data Science." *Medium*, 15 June 2021, towardsdatascience.com/understanding-auc-roc-curve-68b2303cc9c5.

"Prediction of Music Genre." *Kaggle*, 2 Nov. 2021, www.kaggle.com/vicsuperman/prediction-of-music-genre.

"R - Arrays." *Tutorials Point*, www.tutorialspoint.com/r/r_arrays.htm. Accessed 8 Dec. 2021.

Rodrigo, Hansapani. "Lecture Ch 8 Tree-Based Methods." *OneDrive*, uploaded by Dr. Hansapani Rodrigo, 1 Nov. 2021

---. “Lecture Ch 9 Support Vector Machines.” *OneDrive*, uploaded by Hansapani Rodrigo, 8 Nov. 2021,

---. “Lecture Decision Trees, Random Forests, Bagging, and Boosting.” *OneDrive*, uploaded by Hansapani Rodrigo, 3 Nov. 2021,

Science, Odsc-Open Data. “Build a Multi-Class Support Vector Machine in R - ODSC - Open Data Science.” *Medium*, 15 Nov. 2018, medium.com/@ODSC/build-a-multi-class-support-vector-machine-in-r-abcdd4b7dab6.

Zach. “How to Normalize Data in R.” *Statology*, 2 June 2021, www.statology.org/how-to-normalize-data-in-r.