Popularity Analysis of Spotify Songs Using Machine Learning

CPSC6300 – Applied Data Science

Prof. Alexander Herzog

Clemson University, Fall 2020

Presented By:

Nirali Bandaru (C12845262)

Rohan Gangisetty (C12850729)

Introduction

This study investigates the impact of audio features on the popularity of a song on the music platform Spotify. The dataset was taken from Kaggle, an online repository for datasets. Because the response variable "popularity" is numerical, this problem was initially classified as a regression problem. However, the error rate was undesirably high. An alternative model was built using random forests. Due to the large size of the dataset, the random forest model did not produce results and crashed multiple times. To overcome this, a random sample of fewer data points was used to represent and train the data. The error rate from the random forest model was lower and produced better results.

Exploratory Data Analysis

Basic information about the dataset:

Unit of Analysis	Song/Track
Total Observations	169,909
Unique Observations	154417
Time Period Covered	1921-2020

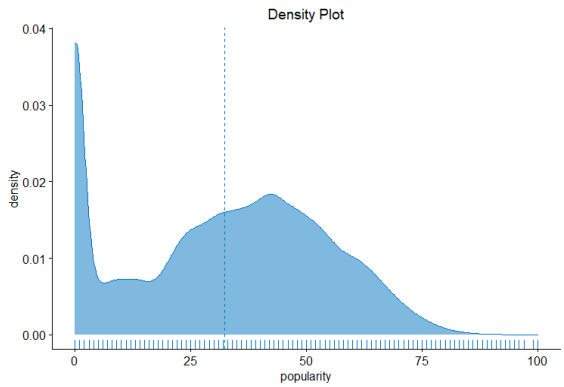
A summary of the variables:

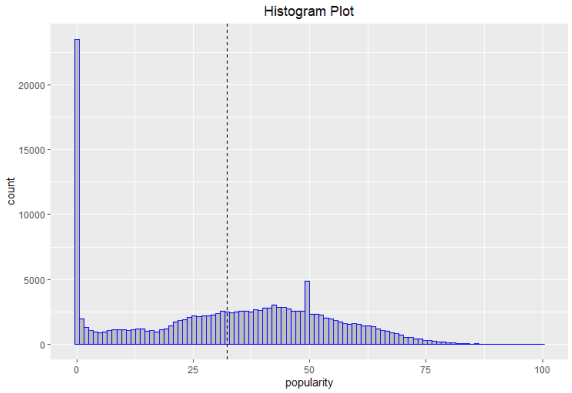
Predictor Name	**Data Type**	**Range before cleaning**	**Range after cleaning**
Acousticness	Numeric	0 - 1	0-0.996
Danceability	Numeric	0 - 1	0-0.988
Popularity	Numeric	0 - 100	0-100
Mode	Binary	0, 1	0-1
Key	Categorical	0 - 11	0-11
Loudness	Float	-60.00 - 3.855	-60-0
Liveness	Numeric	0 - 1	0-1
Instrumentalness	Numeric	0 - 1	0-1
Artist	Categorical	NA	removed
Year	Categorical	1921-2020	1921-2020
Name	Categorical	NA	removed
Duration (ms)	Integer	5k-500k	60000-360000
Explicit	Binary	0, 1	removed
Speechiness	Numeric	0 - 1	0-0.967
Tempo	Numeric	0-250	0-244.09
Release Date (yyyy-mm-dd)	Categorical	NA	removed

As part of the exploratory data analysis, unwanted columns (artists, release date, explicit, id, and name) were removed, data was checked for missing values, duplicate observations, non-unique values, discrepancies in data type, illegal values, typos, mislabeled classes and unwanted outliers. Some illegal values were fixed for the predictor "loudness."

The following are graphical representations of the response variable and the correlation between each predictor variables and the response variable:

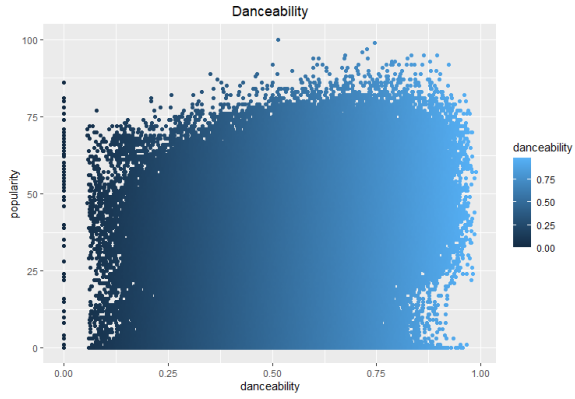
Response Variable - Popularity



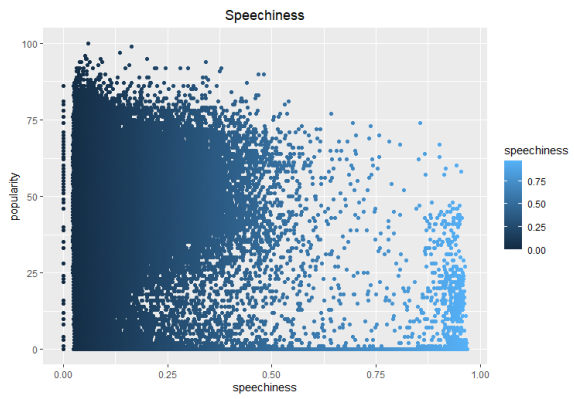


The response variable, popularity, is distributed nearly evenly across the popularity index, but there are a maximum number of songs that are unpopular, or with a popularity index of zero.

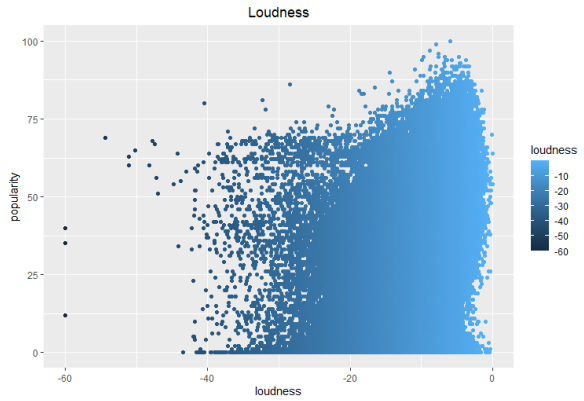
Danceability



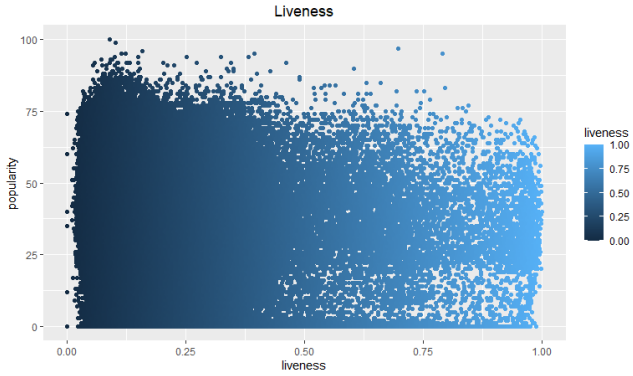
Speechiness



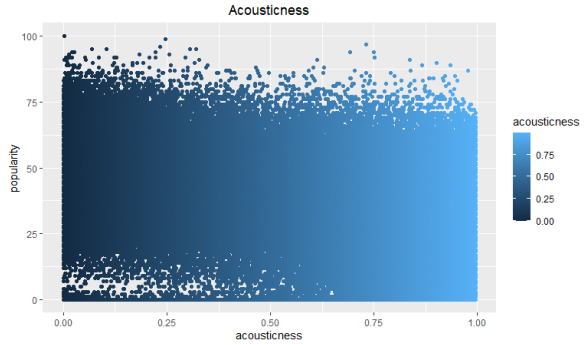
Loudness



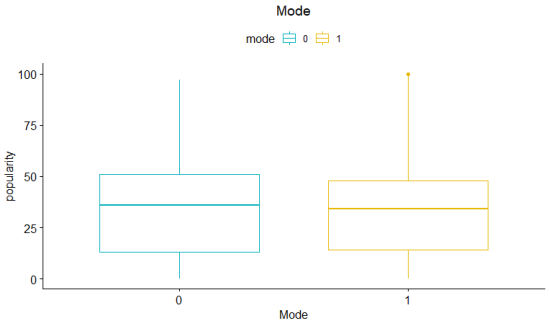
Liveness



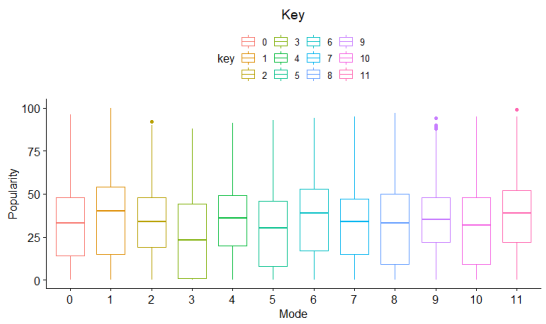
Acousticness



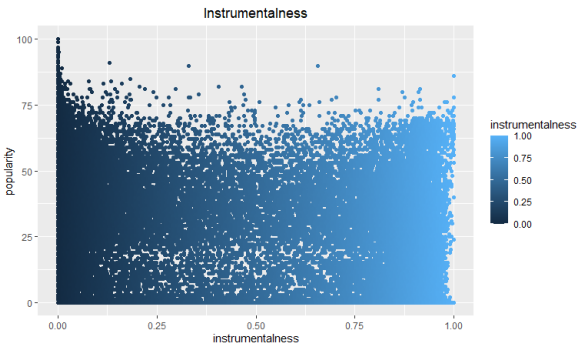
Mode



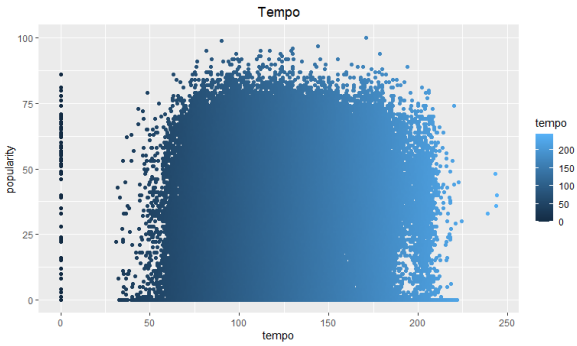
Key



Instrumentalness



Mode



The following table summarizes observations made from the EDA plots:

Predictor	**Observation**
Danceability	No apparent pattern
Loudness	Distribution skewed right, nearly all the songs are on the top half of the loudness index
Speechiness	Distribution skewed left, most points on the bottom half of the speechiness index
Liveness	Distribution suggests most songs are popular if the liveness index is fairly low. Top half of the distribution still has a significant number of data but is less dense when compared with the bottom half.
Acousticness	No apparent pattern
Key	Similarly distributed, no observed extremities
Instrumentalness	Distribution is dense towards the ends of the instrumentalness index, less dense in the middle. Shape could be distributed as a U.
Duration	Slightly peaks at the middle, where duration is approximately 3 min
Tempo	Is condensed slightly inward. Extreme tempos at 0-40 and 225-250 have barely any data points. Otherwise, no apparent pattern is observed between tempo and popularity.

Summary of Machine Learning Models and Discussion

Polynomial Regression Model

The response variable, popularity, has a value range of 1-100. If these numbers were divided into intervals, the response variable could be considered categorical, in which case the problem would be categorized as a classification problem. However, popularity was considered as a quantitative variable, so the problem statement was identified as a regression problem. Upon careful examination of each predictor vs. response variable plot, it was also observed that the relationships mostly appear to be non-linear. Moreover, many of the predictors in the dataset are quantitative. Taking these observations into consideration, the **polynomial regression model** was concluded to be the most suitable model for the given data set.

The resampling method used for this model was k-fold cross-validation at K=10. The following image shows the result from the training and testing of a polynomial regression model.

Model summary:

```
> fitmodel

Call:  glm(formula = I(popularity > 25) ~ poly(acousticness, 10) + poly(speechiness,
    10) + poly(liveness, 10) + poly(loudness, 10) + poly(instrumentalness,
    10) + poly(duration_ms, 10) + poly(tempo, 10) + poly(key,
    10) + poly(danceability, 10), subset = training)

Coefficients:
                  (Intercept)       poly(acousticness, 10)1       poly(acousticness, 10)2       poly(acousticness, 10)3       poly(acousticness, 10)4
                      0.63555                     -89.63069                     -31.95154                     -16.15362                      -3.68042
      poly(acousticness, 10)5       poly(acousticness, 10)6       poly(acousticness, 10)7       poly(acousticness, 10)8       poly(acousticness, 10)9
                     -6.01661                      -1.32848                      -3.03074                       0.67917                      -0.85578
     poly(acousticness, 10)10        poly(speechiness, 10)1        poly(speechiness, 10)2        poly(speechiness, 10)3        poly(speechiness, 10)4
                      1.67009                     -31.10199                     -12.80760                      -8.83111                       7.01766
       poly(speechiness, 10)5        poly(speechiness, 10)6        poly(speechiness, 10)7        poly(speechiness, 10)8        poly(speechiness, 10)9
                     -4.51993                       3.27652                      -3.50926                       2.95818                      -1.88177
      poly(speechiness, 10)10           poly(liveness, 10)1           poly(liveness, 10)2           poly(liveness, 10)3           poly(liveness, 10)4
                      2.35066                      -7.25765                       5.95292                      -0.59917                      -2.50494
          poly(liveness, 10)5           poly(liveness, 10)6           poly(liveness, 10)7           poly(liveness, 10)8           poly(liveness, 10)9
                      4.21800                      -3.98542                       2.46774                      -1.35710                      -1.63882
         poly(liveness, 10)10          poly(loudness, 10)1           poly(loudness, 10)2           poly(loudness, 10)3           poly(loudness, 10)4
                      2.15676                       6.52354                      19.51928                       4.29147                      -2.27751
          poly(loudness, 10)5           poly(loudness, 10)6           poly(loudness, 10)7           poly(loudness, 10)8           poly(loudness, 10)9
                     -0.47178                      -1.97819                      -2.56600                      -0.61307                       0.62395
        poly(loudness, 10)10    poly(instrumentalness, 10)1   poly(instrumentalness, 10)2   poly(instrumentalness, 10)3   poly(instrumentalness, 10)4
                      2.10638                      -6.32777                       8.36489                       0.74089                       4.45874
 poly(instrumentalness, 10)5   poly(instrumentalness, 10)6   poly(instrumentalness, 10)7   poly(instrumentalness, 10)8   poly(instrumentalness, 10)9
                     -0.02526                       2.59257                      -0.63712                       1.40673                      -0.52345
poly(instrumentalness, 10)10        poly(duration_ms, 10)1        poly(duration_ms, 10)2        poly(duration_ms, 10)3        poly(duration_ms, 10)4
                      1.06657                      24.23234                      -6.49776                      -9.60027                       4.09845
       poly(duration_ms, 10)5        poly(duration_ms, 10)6        poly(duration_ms, 10)7        poly(duration_ms, 10)8        poly(duration_ms, 10)9
                      1.45927                      -2.16642                      -1.02808                      -0.13480                       1.54010
      poly(duration_ms, 10)10             poly(tempo, 10)1              poly(tempo, 10)2              poly(tempo, 10)3              poly(tempo, 10)4
                     -0.31817                       1.80579                       6.37792                      -3.25964                       0.21256
            poly(tempo, 10)5              poly(tempo, 10)6              poly(tempo, 10)7              poly(tempo, 10)8              poly(tempo, 10)9
                     -1.74900                       2.20224                      -0.57289                      -0.03628                       0.32352
           poly(tempo, 10)10               poly(key, 10)1                poly(key, 10)2                poly(key, 10)3                poly(key, 10)4
                      1.31430                       0.42461                       1.01725                       1.96743                      -0.31160
             poly(key, 10)5                poly(key, 10)6                poly(key, 10)7                poly(key, 10)8                poly(key, 10)9
                      0.44880                       0.55014                       1.76762                       1.21425                       1.71975
            poly(key, 10)10       poly(danceability, 10)1       poly(danceability, 10)2       poly(danceability, 10)3       poly(danceability, 10)4
                      3.93030                      10.75599                       5.36691                       2.13096                      -3.36049
      poly(danceability, 10)5       poly(danceability, 10)6       poly(danceability, 10)7       poly(danceability, 10)8       poly(danceability, 10)9
                      2.95594                      -4.03415                       3.78477                      -1.70482                       1.28731
     poly(danceability, 10)10
                      0.17790

Degrees of Freedom: 108090 Total (i.e. Null);  108000 Residual
Null Deviance:      25070
Residual Deviance: 13590        AIC: 82790
```

R-squared:

```
> print(model)
Generalized Linear Model

154417 samples
     9 predictor

No pre-processing
Resampling: Cross-Validated (10 fold)
Summary of sample sizes: 138975, 138976, 138975, 138976, 138977, 138975, ...
Resampling results:

  RMSE      Rsquared   MAE
  16.47103  0.4202556  13.31138
```

As seen above, the r-squared value is low at 0.4203, and the RMSE is at 16.47. Although the RMSE is fair, the r-squared is *too low* for this model to be considered a good fit for the data.

Predictions for this model:

```
> predict(fitmodel,newdata )
         1
0.06514788
> predict(fitmodel,newdata1 )
         1
0.8815902
> predict(fitmodel,newdata2 )
         1
0.4873546
>
```

The numbers above are predictions of the response variable, *popularity*.

Random Forest Model

The **random forest model** was selected as a suitable alternative and was expected to produce better results and better predictions. Random forests are known for their robustness and ability to fit non-linear data better than some others. Additionally, the data set used in this study contains both numerical and categorical features, and random forests are known to be better in such cases. Lastly, the random forest was chosen for the ease in interpretability.

The following table summarizes the parameters used to build the random forest model:

Parameter	**Value**
Sample size	10,000
Predictors	10
mtry	3
ntree	500 (default)
Training set	7500
Testing set	2500

Since the dataset is very large, the random forest crashed after training for hours. To overcome this, random samples of 10,000 and 20,000 data points were taken multiple times and the average MSE was taken.

The model summary for 10,000 sample data points is given below:

```
> mean((yhat.rf6-df6.testing)^2)
[1] 203.883
> print(rf6)

Call:
 randomForest(formula = popularity ~ ., data = df6, mtry = 3,      importance = TRUE, subset = training)
               Type of random forest: regression
                     Number of trees: 500
No. of variables tried at each split: 3

          Mean of squared residuals: 200.4266
                    % Var explained: 56.49
>
```

The model summary for 20,000 sample data points is given below:

```
> mean((yhat.rf9-df9.testing)^2)
[1] 198.5886
> print(rf9)

Call:
 randomForest(formula = popularity ~ ., data = df9, mtry = 3,      importance = TRUE, subset = training)
               Type of random forest: regression
                     Number of trees: 500
No. of variables tried at each split: 3

          Mean of squared residuals: 203.9293
                    % Var explained: 55.99
>
```

The percent of variability explained by the models (repeated 10 times) is, on average 56% for both 10,000 points and 20,000 points. This is an improvement from the logistic regression model previously built.

For sample size = 10,000, the following values were obtained:

%Var	MSE Test	MSR Train
57.64	205.55	193.35
55.93	203.13	206.51
55.34	216.09	207.39
58.08	196.17	198.55
54.82	213.63	209.10
57.64	205.55	193.35
55.28	206.22	206.69
56.86	197.91	199.55
54.46	191.99	212.99
55.84	201.13	208.64

For sample size = 20,000, the following values were obtained:

%Var	MSE Test	MSR Train
57.57	203.94	197.45
57.6	202.18	199.33
56.49	203.62	205.10
57.13	190.22	201.03
56.27	202.86	204.67
57.57	203.94	197.45
55.85	199.17	204.22
57.09	182.63	200.17
55.99	198.58	203.92
56.62	199.00	202.14

The following predictions were made for this model:

```
> predict(fitmodel,newdata )
         1
0.06514788
> predict(fitmodel,newdata1 )
         1
0.8815902
> predict(fitmodel,newdata2 )
         1
0.4873546
>
```

A comparison between predictions made with the two machine learning models is shown below. Prediction cases were chosen in three levels: lows, mediums, and highs, with some predictors kept constant. These constant predictors were chosen based on observations made from the EDA.

Predictor	**Case 1**	**Case 2**	**Case 3**
Acousticness	0.99	0.1	0.5
Danceability	0.99	0.15	0.5
Instrumentalness	0.99	0.15	0.85
Key	7	7	7
Speechiness	0.99	0.15	0.3
Loudness	-1	-55	-10
Liveness	0.99	0.15	0.2
Duration (ms)	180000	180000	180000
Tempo	100	55	30
Polynomial Regression	0.292	0.019	0.439
Random Forest (10k)	13.596	33.407	28.621
Random Forest (20k)	11.857	41.104	30.374

The above table for prediction comparison shows that in the polynomial regression model, the popularity index predictions are too low compared to the predictions made by the random forest models. It also appears as though the random forest model with higher sample points made better predictions than the model with lower number of sample points.

The following summary ranks predictors by importance. Note that the plot was not able to be produced due to an error that said, "figure margins too large."

```
> importance(rf1)
                  %IncMSE IncNodePurity
acousticness     193.758650    1119599.04
danceability      49.713945     232242.93
duration_ms       64.302328     345052.97
instrumentalness  38.685774     234378.43
key                2.376188      87916.22
liveness          26.034626     191964.41
loudness          77.254269     634464.08
speechiness       60.077808     304952.88
tempo             11.828939     164386.44
```

It is observed that "acousticness" impacts the response variable much more than the other predictors in the data set.

Conclusion

The exploratory data analysis provided useful insights regarding the correlation between the predictors and the response variable. Many decisions regarding the model were made based on these observations, such as which predictors to take into consideration, which variables should be kept constant while making predictions, etc.

While polynomial regression was theoretically a suitable model for the type of problem being addressed, it was not suitable for this particular data set. The random forest model was much more successful in explaining the variance in the data. Since the data set is very large and caused problems while being trained, multiple random samples of smaller sizes were trained for results. However, it is inferred that training the full data set would yield better results.

Overall, the random forest model performed better than the polynomial regression model.