Rene Sanchez

# SPOTIFY USER ANALYSIS AND PREDICTION

# The Data

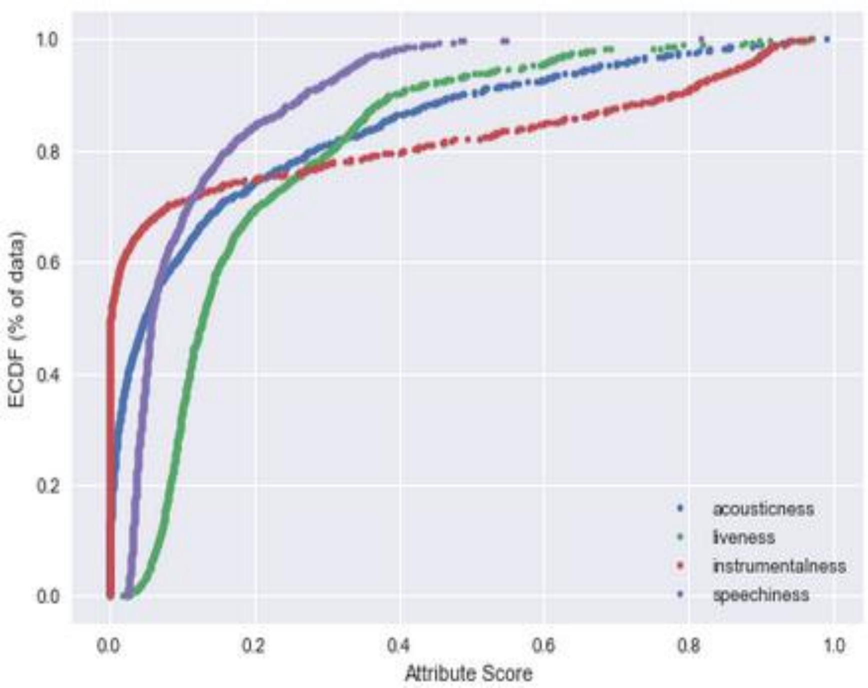
- The dataset used was obtained from Kaggle and contains 2017 songs with their attribute scores and whether or not they were liked by the user
- This data was initially gathered from the Spotify API
- The data started off very clean with the columns properly formatted as ‘int’ and ‘float’ types

# The Goals

- Create a clear profile of attributes that define the user’s preferences
- Create a recommendation engine that can accurately predict what songs the user will like

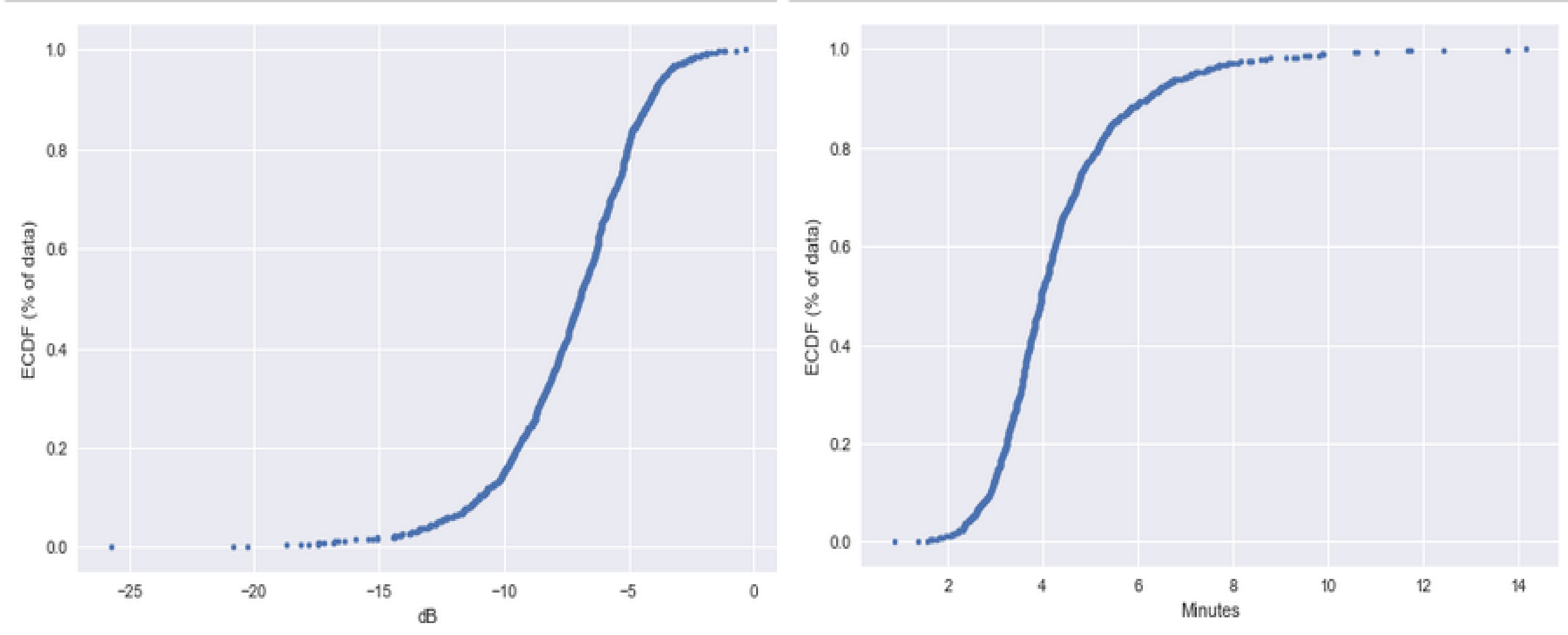
# Exploratory Analysis

- The skewed attributes provide evidence for preferred values for the given attributes because of the concentration of songs around that value.
- Attributes that are normally distributed are not concentrated at certain values but are instead relatively evenly spread out over the range



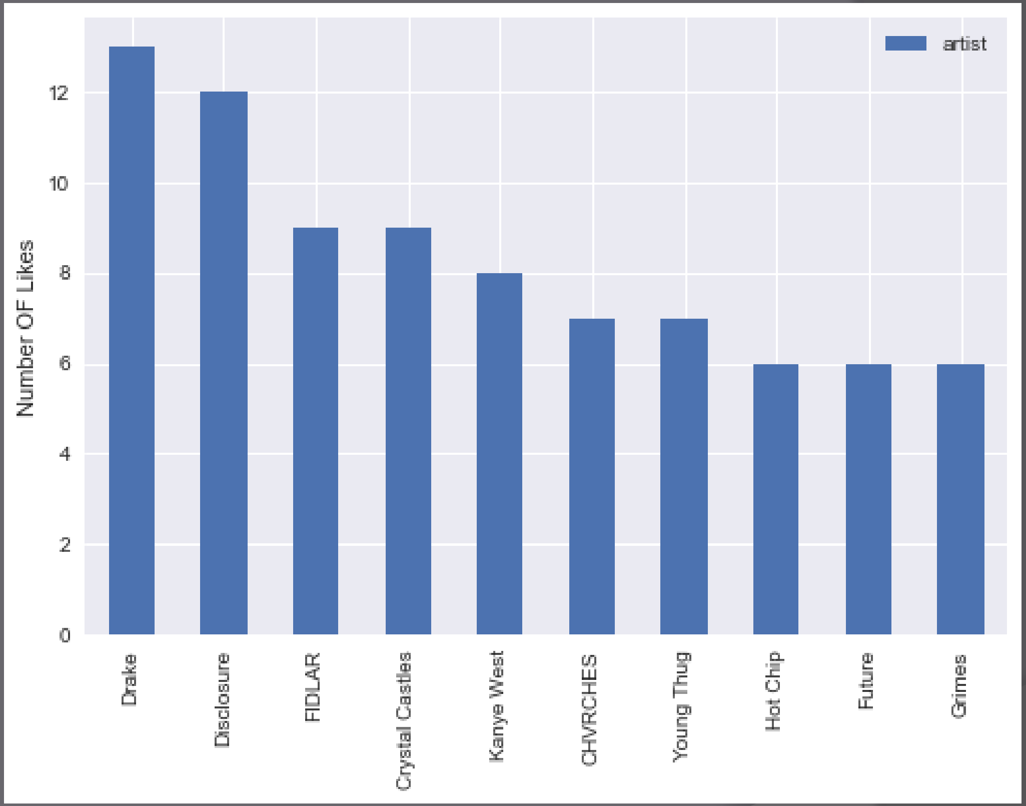
# Exploratory Analysis

- Loudness and Duration are plotted below on their given scales



# Exploratory Analysis

- The user’s favorite artist is likely Drake or Disclosure. With this information we can confidently present the user new releases from these and other artists to keep them up to date with their favorite music



# Exploratory Analysis

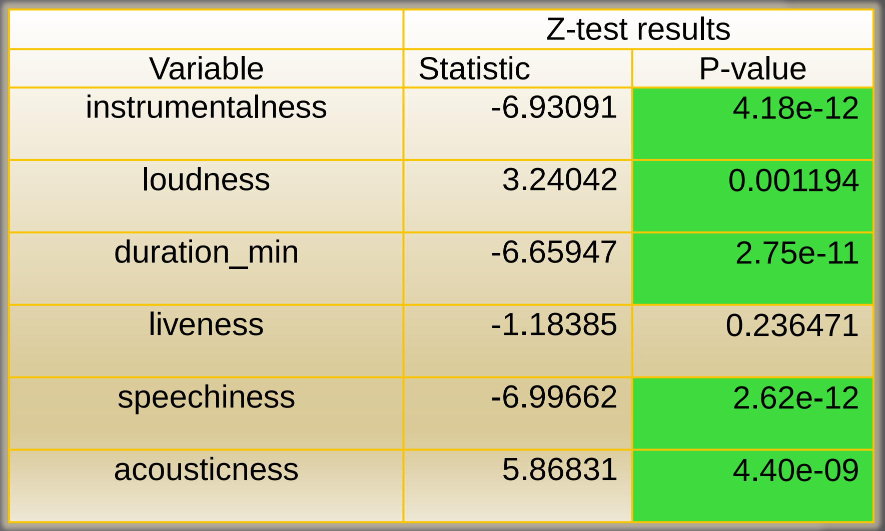
- This cursory visual analysis gives us some indication as to what the user’s focus is in determining what song they like and gives us criteria that can be used to selected songs from Spotify’s database to run the recommendation engine on

# Statistical Analysis

- To test whether these variables are significant indicators of the user’s preferences a statistical examination of the numerical attributes that exhibited skewness as well those that appeared to have considerable mean differences between liked and disliked songs was conducted.

# Statistical Analysis

- Z-tests were used to determine if the differences between groups was significant and indeed the highlighted variables all had significantly different means suggesting they are prominent features that the user uses to determine whether or not they like a given song.



| | Z-test results | |
|---|---|---|
| Variable | Statistic | P-value |
| instrumentalness | -6.93091 | 4.18e-12 |
| loudness | 3.24042 | 0.001194 |
| duration_min | -6.65947 | 2.75e-11 |
| liveness | -1.18385 | 0.236471 |
| speechiness | -6.99662 | 2.62e-12 |
| acousticness | 5.86831 | 4.40e-09 |

# Predictive Model, But First…

- Before creating a predictive model for the data I will explore the possibility of some features being more important than others.
- This will hopefully reduce any noise in the predictive model and increase accuracy by removing unhelpful variables.
- I will use Lasso regularization to determine if any variables should be dropped from the model due to their coefficient being reduced to 0 with an L1 constant of .01 (chosen based on max scale values).

# Predictive Model, But First…

- Lasso reduced energy, key, and time signature coefficients to 0 indicating they are not well correlated with the target variable. Therefore I will run one model with the recommended variables removed and one with all the variables included to compare the initial results before improving the model.

# Predictive Model - ANN

- Two ANNs were designed using either the total number of variables or the recommended size given dropped variables.
- 2 hidden layers with 15 and 11 nodes was chosen; more layers significantly reduced accuracy while reducing variance and more nodes improved accuracy less than the consequential increase in variance.

# Predictive Model - ANN

| ANN Model | Variance | Mean Accuracy |
|---|---|---|
| Full Model result: | 0.0257 | 70.652% |
| Lasso result: | 0.0250 | 49.477% |

- The initial results were determined using a batch size of 10 over 50 epochs since the training dataset is not particularly large.
- 5-fold cross validation revealed that following the Lasso recommendation significantly reduces accuracy of the ANN.

# Predictive Model - ANN

- Numerous values were tested through GridSearchCV for batch size and epochs based on the size of the dataset with batches size 1-15 and epochs 50-200 being explored. There appeared to be convergence at batch size = 1 and epoch=99.
- The final ANN model yielded a 74.917% accuracy rate on the test set, a better score than training but one to challenged by other methods.

# Predictive Model - SVM

| SVM Model | Variance | Mean Accuracy |
|---|---|---|
| Full Model result: | 0.0287 | 73.686% |
| Lasso result: | 0.0024 | 51.342% |

- A Support Vector Machine (SVM) was used next to create a predictive model through the same methodology.
- Using initial values of C = 1.5 for the tolerance value and gamma=.14 for the vector distance parameter, the model was tested with both the Lasso recommendations and the full model.
- Again we see here that the Lasso recommendation model underperforms and the full model performs slightly better than the ANN did in its training

# Predictive Model - SVM

- The tolerance parameter ‘C’ was explored from 1-1000 and ‘gamma’ was explored from .01-100 with GridSearchCV converging at C = 1 and gamma=.136 with an accuracy rate of 74.562%.
- The final SVM model we find that it does indeed outperform the ANN with an accuracy rate of 77.227%.

# Predictive Model - SVM

| SVM Model | Variance | Mean Accuracy |
|---|---|---|
| Full Model result: | 0.0287 | 73.686% |
| Lasso result: | 0.0024 | 51.342% |

- A Support Vector Machine (SVM) was used next to create a predictive model through the same methodology.
- Using initial values of C = 1.5 for the tolerance value and gamma=.14 for the vector distance parameter, the model was tested with both the Lasso recommendations and the full model.
- Again we see here that the Lasso recommendation model underperforms and the full model performs slightly better than the ANN did in its training

# Predictive Model - RFC

- Given that the Lasso recommendation has thus far consistently underperformed, the following models won’t be cross validated using the Lasso recommendation.
- A Random Forest classifier (RFC) will use only a subset of the total amount of features to create a predefined number of decision trees and then average those trees out.
- The number of trees was explored exponentially from 10 – 10,000 and converged at 90 trees. This RFC configuration yielded a 77.887% accuracy rate on the training data and yielded the highest accuracy thus far on the test data, 79.208%.

# Predictive Model - GBT

- The final model will be Gradient Boosted Trees (GBT) which will operate in a similar fashion as the RFC but instead of treating each decision tree as independent it trains the next tree based on the previous tree’s residual.
- There are several parameters to be specified for GBT:
  - Number of Estimators - Number of trees to utilize
  - Learning Rate - Defines the rate the trees adjust based on the previous tree
  - Max Depth - Defines how large each tree can be

# Predictive Model - GBT

- The GBT had 77.479% accuracy on the training set and converged at a Learning Rate of .2, Max Depth of 5, and Number of Estimators of 365.
- Values were explored for these parameters exponentially just as was done in the previous models to ensure that an earnest attempt was made to find a global minimum rather than a local minimum.
- This model specification yielded 80.198% accuracy on the test set, taking the top spot for performance on this dataset.

# Closing Thoughts

- To implement these findings, we would query songs whose attributes of interest (defined in the statistical analysis) fall within the concentrated range identified in the visual inspection of the data.
- Using these queried songs, we would then run the final GBT model and recommend the songs that are classified as ‘1’ in the model results.

# Closing Thoughts

- We would expect this model to be able to fill a playlist with approximately 80% of songs that the user will like and add to their own library.
- Additionally, as the user goes through our predictions and likes or dislikes them, we would add those to our training data to further improve the models performance over time.