Machine Learning Model Overview

**Which model did you chose and why?**

* I chose to use a random forest model because they:
  + Allow for feature ranking, which we can use in order to eliminate features that have little impact on the output of our model
  + Run efficiently on large datasets
  + Are robust to outliers

**How are you training your model?**

* Split the data into testing and training sets
* Scale the input data
* Fit the classifier with the training data

**What is the model’s accuracy?**

* The model’s accuracy is 15%

**How does the model work?**

* Samples the data and builds several decision trees from a random subset of features
* Makes predictions using the testing data

**What is the input and output data of your model?**

* The input data of our model are song lyrics that were pulled from the LyricsGenius python client for the Genius.com API and have been put through a natural language processing pipeline.
* The output data of our model is the category/genre of the song.

**What are other possible models that could be used?**

* Neural network model
* K-means algorithm

Notes

**Supervised ML**

Classification

* Random Forest
  + Allows for feature ranking, which we can use in order to eliminate features that have little impact on the output of our model
  + Run efficiently on large datasets
  + Are robust to outliers

Boosting

* Combining a set of weak learners into a strong one

Class Imbalance

* We can implement oversampling, under sampling, or combination sampling techniques to ensure that we have an even amount of instances from each category/genre

**Unsupervised ML**

Clustering Algorithms

* K-means
  + We will have a predetermined number of categories that we can use as the K value in order to create clusters that predict the genre of a song
  + Runs relatively quickly, and can scale to large datasets
  + Works best for spherical-looking data with similar density points closely grouped together

Dimensionality Reduction

* Principal Component Analysis (PCA)
  + A statistical technique to speed up ML algorithms when the number of input features/dimensions is too high
  + *Feature extraction*: reduces the number of dimensions by transforming a large set of variables into a smaller one that contains most of the information from the original set
  + Use principal component data with the K-means algorithm

**DL Neural Networks**

* Can combine the performance of multiple statistical and ML models
* Effective at detecting complex, nonlinear relationships
* Prone to overfitting
* Used to classify consumers
* *Perceptron model*: capable of classifying datasets with many dimensions; however, it is most commonly used to separate data into two groups