ORIE 4741 Final Project
Prof. M. Udell

An Insight into Sounds from Spotify

Vidita Gawade, Shruti Sanghavi, Feiyu Tao

Abstract

While technology has drastically changed the way in which people listen to and share music, one thing has remained the same: everyone loves to listen to music. Spotify is a music, podcast, and video-streaming service that was founded in 2008. While Spotify does currently recommend songs to its users based on their listening history, the algorithm it uses are focused on grouping similar music listeners together, also known as collaborative filtering. By the way of this project, we aim to build a recommendation tool that uses the rich features of songs to predict its likeability for a Spotify user solely using a user's own listening history.

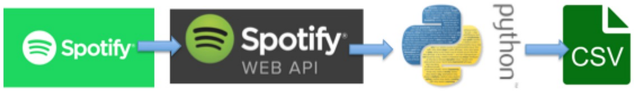
Keywords

Spotify, Music Recommendation, Supervised Learning

Contents

1. Data Collection, Creation, and Connection

Before we performed the analysis, we needed to get the data into a format readable by the programming language we used. First, we recognized that musical characteristics can greatly differ between genres, so we decided to collect songs from a variety of genres that were determined by Spotify's 'Insights' website[1]. The datasets were collected by a team member and tailored for the purpose of this project. On Spotify, our initial playlist comprised of over 700 songs, with equal representation across four genre classes (Pop, Hip Hop, Mellow, Rock). The songs in each of the four genres were separated into two playlists ("Like" and "Not Like").



Using Spotify's Developers' toolkit, we created a user authentication token to approve access to the application. The API was connected via Python using the library: **spotipy**. Within Python, the playlists were combined, the features "Likeability" and "Genre" were added since the two were not part of the original Spotify features, and csv files were generated.

After preliminary analysis, we decided to focus on making predictions for each individual genre. The reasoning behind this decision is further discussed in the Preliminary Analysis section. So, as a first step, we extended the Hip Hop playlist so that there were approximately 1,400 songs of "Not Like" and 400 songs of "Like".

For the purposes of analysis, all datasets were randomly shuffled and split into one-tenth cross validation set, and of the remaining, two-third training set and one-third test set.

2. Data Features

Our dataset is clean to start with. Indeed, the features are rich in that they were simplified by reporting confidence measures through a combination of various musical elements. The list of the features is as follows:

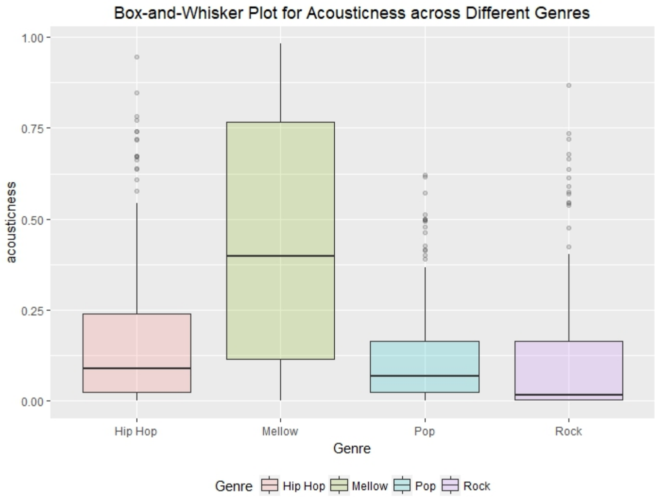


Figure 1. Acousticness across Different Genres

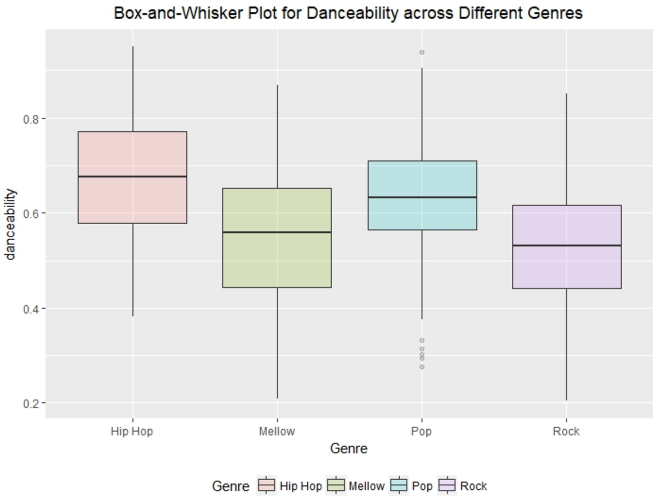


Figure 2. Danceability across Different Genres

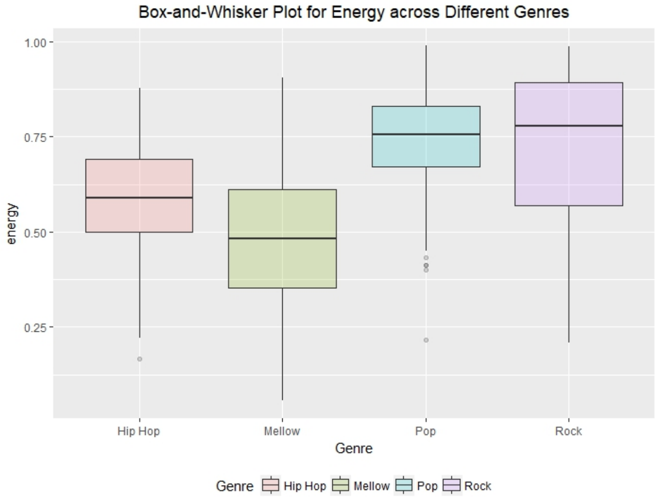


Figure 3. Energy across Different Genres

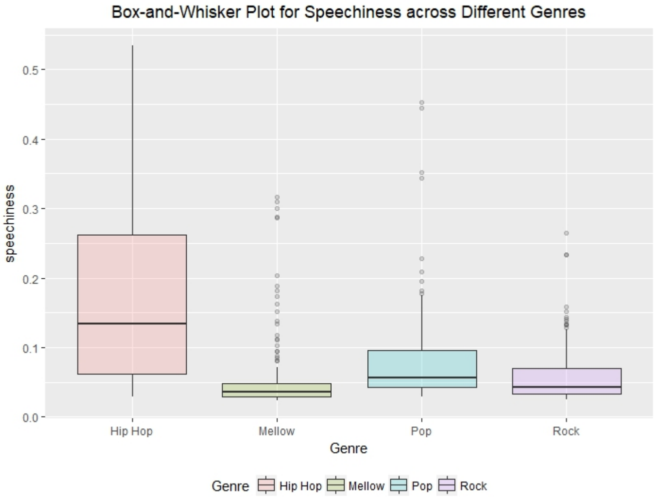


Figure 4. Speechiness across Different Genres

Acousticness, Danceability, Energy, Duration, Instrumentalness, Key, Liveness, Loudness, Mode, Speechiness, Tempo, Time Signature, Valence, Genre, Like or ot Like. *Where*, **Acousticness**: reports a confidence measure between 0 and 1 of how naturally sounding the sounds are, where a 0 implies the sounds are more electrically amplified as opposed to being naturally produced; **Danceability**: reports a confidence measure between 0 and 1 of how danceable the music is using a combination of elements such as tempo, beat and strength; **Energy**: refers to the speed, loudness, and noisiness of a song in which elements such as timbre and general entropy impact the energy of a song; **Instrumentalness**: refers to the vocalness of a song; in other words, sounds such as “ooh” and “aah” would make the Instrumentalness value closer to 1 whereas spoken words would make the Instrumentalness value closer to 0; **Liveness**: detects whether the songs were performed in the presence of an audience as opposed to a studio; **Speechiness**: recognizes the performance of words as speech over music, **Valence**: measures the degree of positivity of the song; Happy, cheerful, euphoric songs have higher valence; **Key**, **Loudness**, **Mode**, **Tempo**, **Time Signature** take on their respective conventional definitions. For a further description of the features, refer to Spotify’s audio feature description: `https://developer.spotify.com/web-api/get-audio-features/`.

3. Data Visualization

Multiple box and whisker plots were generated for observing the spread of Acousticness, Danceability, Speechiness, Energy, and Valence across genres. These plots

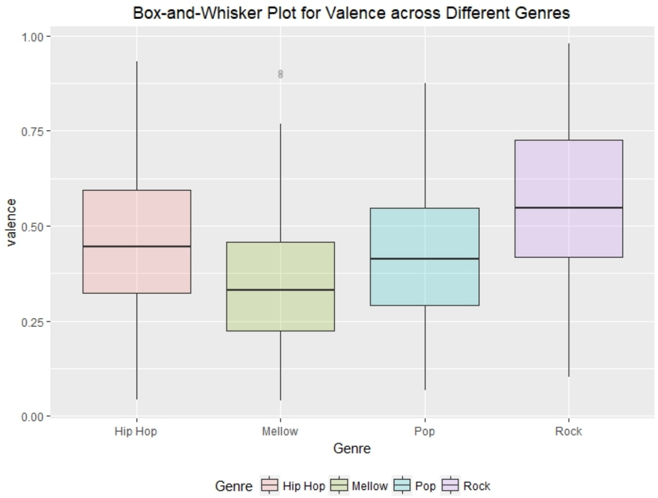


Figure 5. Valence across Different Genres

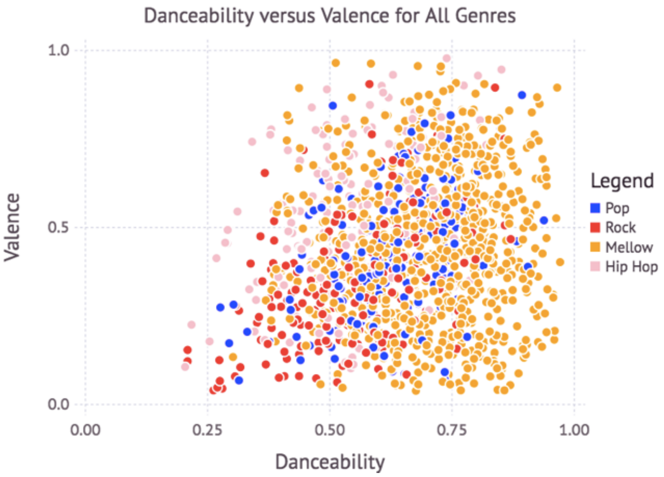


Figure 6. Danceability vs Valence

provide a visual insight into characteristic differences amongst genres. While model analysis will show which features are significant, these plots throw light on the spread of the features.

For the Acousticness feature, we see that the Mellow genre has more diversified values, while 75% or more songs from Pop, Hip Hop, and Rock tend to be less acoustic (see **figure 1**). As expected, for the Danceability feature, Pop, Hip Hop songs have more songs that are danceable (values closer to 1), while Rock and Mellow songs tend to have values across the spectrum (see **figure 2**). For the Energy feature, Pop and Rock songs tend to be more energetic, with Hip Hop as a runner-up, while surprisingly Mellow songs tend to be spread out in energy (see **figure 3**).For the Speechiness feature, Pop, Mellow, and Rock songs tend to favor musical sounds over spoken words, and while Hip Hop tends to favor musical sounds as well, it has more elements of Speechiness compared to the other genres (see **figure 4**). This is because Hip Hop has elements of Rap music, songs in which more words are spoken rather than sung. For the Valence feature, while all genres seem to be spread in Valence, or Happiness, the middle 50% of Rock songs seem to be higher in Valence values, while the middle 50% of Mellow songs seem to be lower in Valence values (see **figure 5**).

Figure 6 shows a comparison of values for Danceability against values for Valence for all genres. In particular, while Pop and Hip Hop songs tend to have values closer to 1 for Danceability, both Pop and Hip Hop songs tend to have values spread out from 0 to 1 for Valence.

On the other hand, even though Rock and Mellow songs tend to have spread out values from 0 to 1 for Danceability, Rock songs tend to have values greater than 0.5, while Mellow songs have values less than 0.5 for Valence.

4. Performance Measure

It is important for Spotify to recommend songs that a user would like. However, it is also important for Spotify to not incorrectly classify a song as a "Not Like" as a user would probably never be recommended that particular song in the future. In other words, it is important for the model to minimize the false negative rate and maximize the true positive rate as much as it can.

As a result, the success of our model was dependent on a trade-off between these two considerations. We measured the performance of our model based on the results of weighted average. Weighted average is calculated as follows:

$$\text{Weighted Average} = 0.5 \times \frac{a}{b} + 0.5 \times \frac{a}{c} \tag{1}$$

- a = number of songs that I actually like that Spotify also recommends
- b = total number of songs Spotify recommends
- c = total number of songs I like in my evaluation dataset; where evaluation dataset refers to either the test or cross validation set

Realistically, on average, a person would not listen to more than 10 new songs in one sitting. We averaged

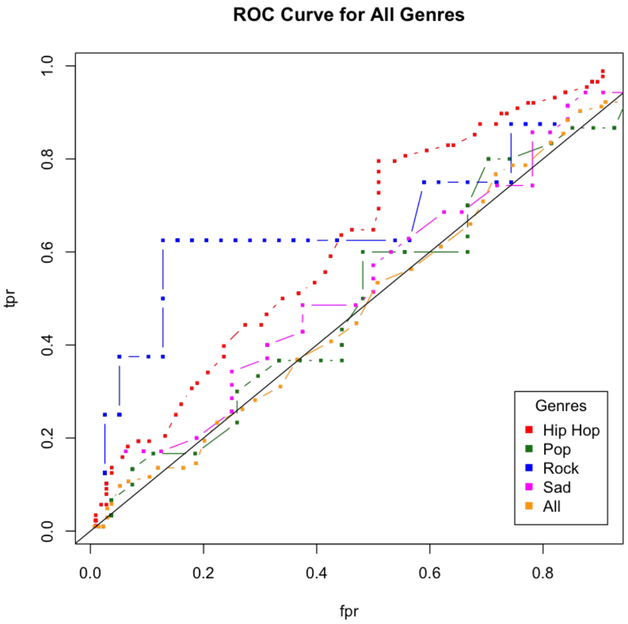


Figure 7. ROC Curve

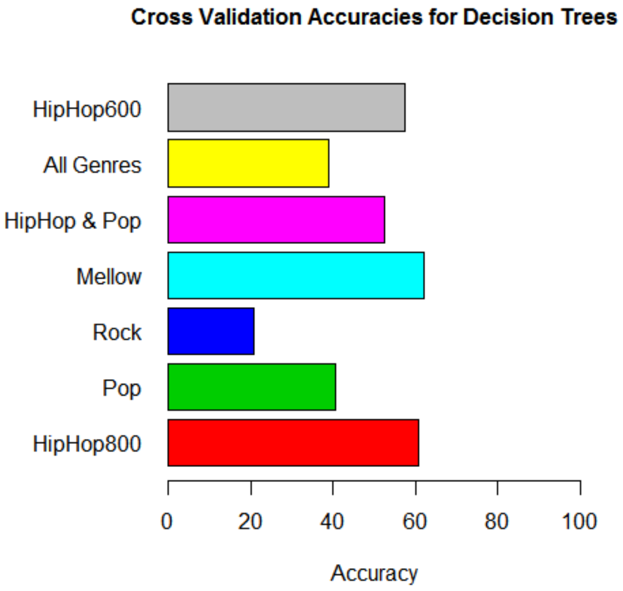


Figure 8. Accuracy of Decision Trees

our model performance by running 500 replications and reported the mean value of the weighted average. We selected the model which performed the best on the cross validation dataset: Logistic Regression. Specifically, Logistic Regression yielded a weighted average of 67% for the test set and a weighted average of 70% for the cross validation set.

5. Preliminary Analysis

The Genre feature was initially coded as “1, 2, 3, 4” to represent the four different genres in the playlists. However, that implied one genre was more important than another. In fact, Genre is a nominal feature, meaning that it exists in the form of a name only, not having any specific priority or characteristic to it. To resolve this, the Genre feature was one-hot encoded. Yet, our classification techniques did a poor job at fitting the models to the new dataset which included the feature-engineered Genre feature. The weighted average results were low (45%).

Preliminary analysis showed that predicting songs within one genre, rather than all together, gave better results. In particular, the Hip Hop genre seemed to give good results. Therefore, our adapted dataset comprised of about 2000 songs, spanning across four major genres, and we decided to fit different models for different genres.

When building individual models, we were interested in the performance of the models amongst different genres. The first model that we tested is Logistic Regression. In Logistic Regression, there is a parameter referred to as the threshold usually set to 0. In this case, if the model predicts a value greater than 0.5, the data point gets classified as a 1, otherwise a 0. But, one can change the threshold to increase true positive rate at the expense of true negative rate. We used a Receiver Operating Characteristic curve to plot the true positive rates (also known as sensitivity) against false positive rates for various cutoff points (or thresholds) since we were interested in increasing our true positive rate.

A ROC curve above the diagonal line y=x and skewing towards the upper left corner indicates good accuracy, while a ROC curve lying close to the diagonal line $y = x$ indicates a model that is no better than a random guess, while a ROC curve underneath the diagonal line $y = x$ indicates poor fit.

For varying thresholds in the Logistic Regression, the ROC curve shows that the model trained on only the Hip Hop genre is associated with a greater accuracy than the models for Rock, Mellow, Pop, and All Genres combined (see **figure 7**). Furthermore, the ROC Curve for Hip Hop indicated that a threshold at 0.4 would yield

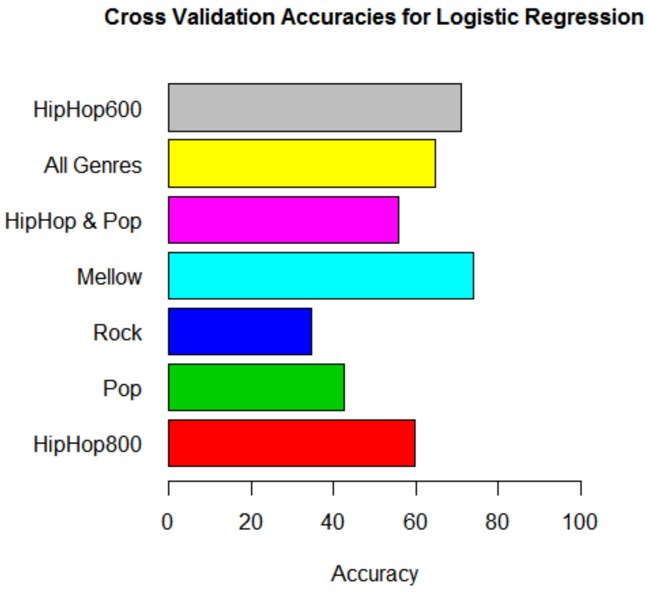


Figure 9. Accuracy of Logistic Regression

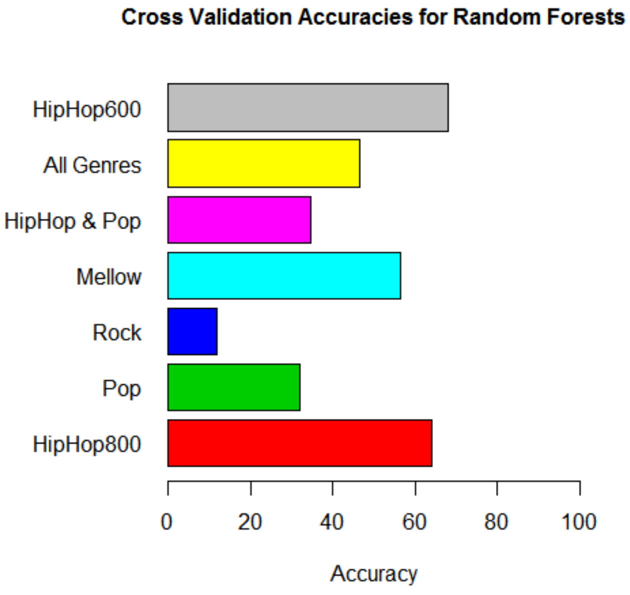


Figure 10. Accuracy of Random Forests

an optimal true positive rate for at the expense of a true negative rate. As a result, our focus of analysis centered on the Hip Hop genre since it tended to produce models with better predictions.

Within the Hip Hop Genre, we ran the model on three types of datasets:

Dataset 1: 400 songs liked and 1400 songs as not liked

Dataset 2: 400 songs liked and 400 songs as not liked

Dataset 3: 300 songs liked and 300 songs as not liked.

As **figure 8, 9, 10, 11** show, dataset 3 performed the most accurate so our focus of analysis specifically centered on the dataset 3 where models were trained, tested, cross validated.

6. Model Analysis

We decided to use the following classification techniques since our problem involves mapping real valued input data to binary outcomes. The classification techniques used are Logistic Regression, Decision Trees, Random Forests, and Support Vector Machines. The following results are based on dataset 3 of Hip Hop songs.

6.1 Logistic Regression

Logistic Regression maps the relation between the binary outcomes of “Like” and “Not Like” to the statistical probabilities. The model determined that Acousticness, Danceability, Speechiness, and Valence were important musical features in predicting this specific user’s likability of songs. This seems logically accurate as we can see from **figure 1, 2, 4, 5** that Hip Hop shows a preference for the aforementioned features. Furthermore, as **figure 9** shows, the test set weighted average was reported as 70% and cross validation set was reported as 67%.

6.2 Decision Trees

Decision Trees is an appropriate model to use since our aim is to classify our categorical response variable. Moreover, trees are simple and easy to interpret. A Decision Tree uses a recursive greedy approach to determine the best split at each feature. In other words, the tree only looks at the present step and attempts to grab the most information that it can from that particular feature. In the end, we also prune the true to obtain a subtree to prevent over-fitting. **figure 12** shows an example of a decision tree’s split points for different features in Hip Hop songs. As **figure 8** shows, the test set weighted average was reported as 54%, whereas the cross validation set weighted average was reported as 58%, indicating not as great of a fit as Logistic Regression.

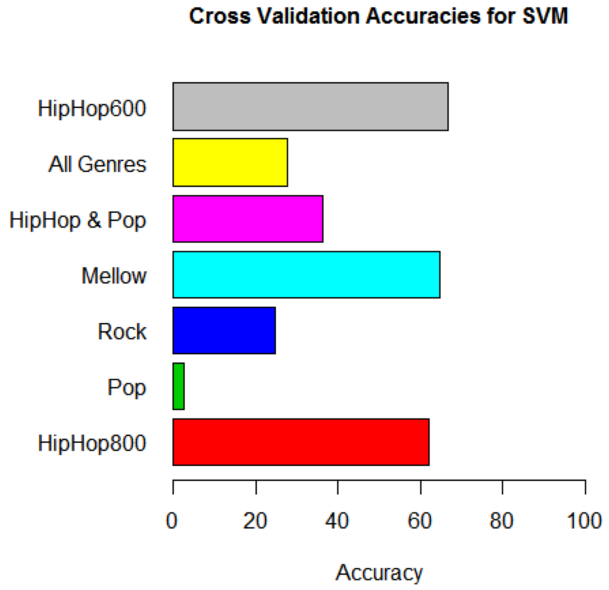


Figure 11. Accuracy of Support Vector Machines

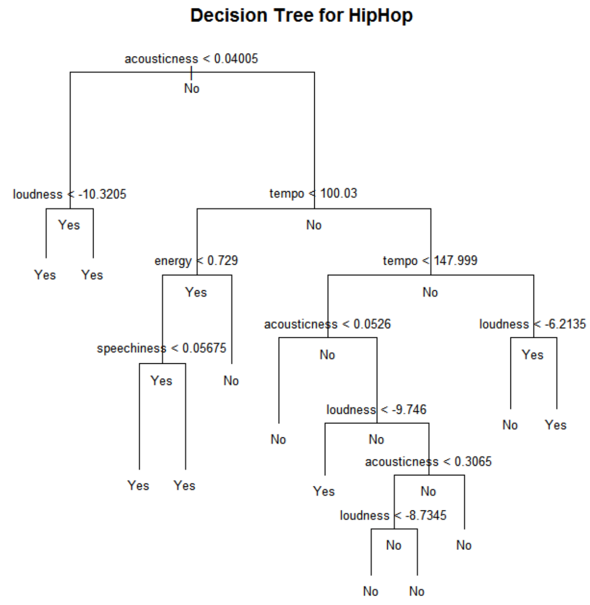


Figure 12. A Plot of Decision Trees

6.3 Random Forests

Bagging is a special type of Decision Tree which combines multiple Decision Trees and uses the average votes from all trees for classifying each data point, reducing the prediction variance. Random Forests is a special type of Bagging technique that decorrelates the trees by finding the best split among only 'm' randomly selected predictors. Leo Breiman, suggested to make 'm' be $\sqrt{p}$ for classification, where p is the total number of features in our dataset. Since the dataset has nine principal features, Random Forests selected splits among three features. As **figure 10** shows, the test set weighted average was reported as 59%, whereas the cross validation set weighted average was reported as 68%, indicating a better fit than Decision Trees, but not as great as for Logistic Regression.

6.4 Support Vector Machines

Support Vector Machines is the last model we decided to use for our project since it classifies the data by finding a hyperplane that differentiates the classes. It integrates the hinge loss to allow margin of errors when classifying response variables. As **figure 11** shows, the test set weighted average was reported as 55%, whereas the cross validation set weighted average was reported as 66%, indicating not as great a fit as Logistic Regression.

7. Results and Conclusion

Among the models, Logistic Regression yielded the highest weighted average in both the test set (70%) and cross validation set (67%). This was because an optimal threshold for true positive rate at the expense of true negative rate was chosen. The weighted average accuracies for Decision Trees, Random Forests, and Support Vector Machines were lower than for Logistic Regression. As an improvement, parameter tuning should be performed as future work to improve the results of the aforementioned classification techniques.

8. Future Work

For the purposes of this project, we manually classified the songs as "Like" and "Not Like" depending on the preference of the particular user.

However, we suggest that an automated tool be setup to monitor the time that the user listens to a song before it was stopped. If the average listening time of a particular song is greater than a threshold, then, it is automatically sent to the "Like" playlist. In this automated procedure, there can be a flaw such that a user may listen to a song he or she does not like for fun but it gets classified as a song he or she likes. In this case, these songs would have to be treated and learned as noise in the data, or outliers.

We also believe that although our study was done only on four genres, this methodology can in fact be applied to other genres. In particular, feature engineering should be done such as adding the feature of whether a song has made the top playlist, such as the top Billboard playlist, or the top Spotify playlist, or been played on the radio. For example, it is often the case that the song a user likes is often related to whether he or she has heard it on the radio or not.

This project focused on musical features amongst songs to build a model that would predict a user's likability. Unsupervised learning, or in this case, Spotify's existing tool based on collaborative filtering, clusters users according to their similarities such as demographics. This project, on the other hand, can be used by a user to build his or her own model using the rich musical elements in songs, based solely on their song selections and independently of other users, adding uniqueness.

References

[1] Genres in the Key of Life: Different Music Uses Different Scales, Kenny Ning, Eric Humphrey, Eliot Van Buskirk,
https://insights.spotify.com/us/2017/10/03/genres-and-key-signatures/