STATS 415 Project Report

Music Genre Classification Based On Lyrics

Mei Fu, Sicun Chen

Introduction

Unstructured data such as document and text is an important form of data widely available on the internet. Lyrics often contain political, social, and economic themes—as well as aesthetic elements—and so can communicate culturally significant messages, which makes it an ideal source for doing natural language studying. Our project is interested in predicting music genres, specifically pop and christian music, based on the song's lyrics and extracting important words that can discriminate these two genres well.

Data

The raw dataset was scraped by Kaggle user Sergey Kuznetsov from LyricsFreak. It contains total 57670 songs and 4 columns: Artist, Song Name, Link, and Lyrics. Since the original dataset didn't contain the genre information, we used Spotify API to fetch each song's genre and merged the genre column into the raw dataset. The genres returned from Spotify API include: pop, rock, alternative rock, christian music, country, metal, r&b, folk, hip-hop, jazz, comedy, and others.

Text Preprocessing

Due to the nature of lyrics data, there were many phrases that indicate the compositional part of a song or lyric repetition, such as [intro], [verse], [coda], [guitar solo], and [repeat x2], etc. We removed these patterns along with non-English lyrics, punctuations, numbers, whitespaces, stopwords, duplicates, and songs without genres. After we got rid of unwanted intances, we lowercased and stemmed all the words.

String to Document Term Matrix

Using the clean lyric corpus, we created two Document Term Matrixs with 55571 rows and 59703 columns. The cells of the first matrix represent the number of times of each unique word appeared in each document(song), and those of the second use tf-idf score to measure the relative importance of a word to a document.

Exploratory Data Analysis

Most Frequent Words & Most Important Words

By sorting the Document Term Matrixs with different weightings from the preprocessing step, we found the most frequently used words and most important words in the lyrics corpus.

It seems like "love" is the most common and prominent word used in all kinds of music. There are some minor differences between the two tables: for example, babi ranks high in the tf-idf table although it doesn't make it into the top frequency list, which means that babi is representative of the lyric corpus although it doesn't show up in certain genre of lyrics very often. To get a sense of different lyric sets, let's look at the lyric wordclouds of each genre:

word	freq
love	77022
just	65326
dont	57310
know	57082
like	52513
see	41252
time	40617
can	40451
got	39184
now	39069

word	tf_idf
love	1164.9375
dont	833.0196
babi	776.6131
just	753.4357
know	752.8415
want	742.2499
like	724.6648
will	712.8482
come	704.9696
time	701.7837

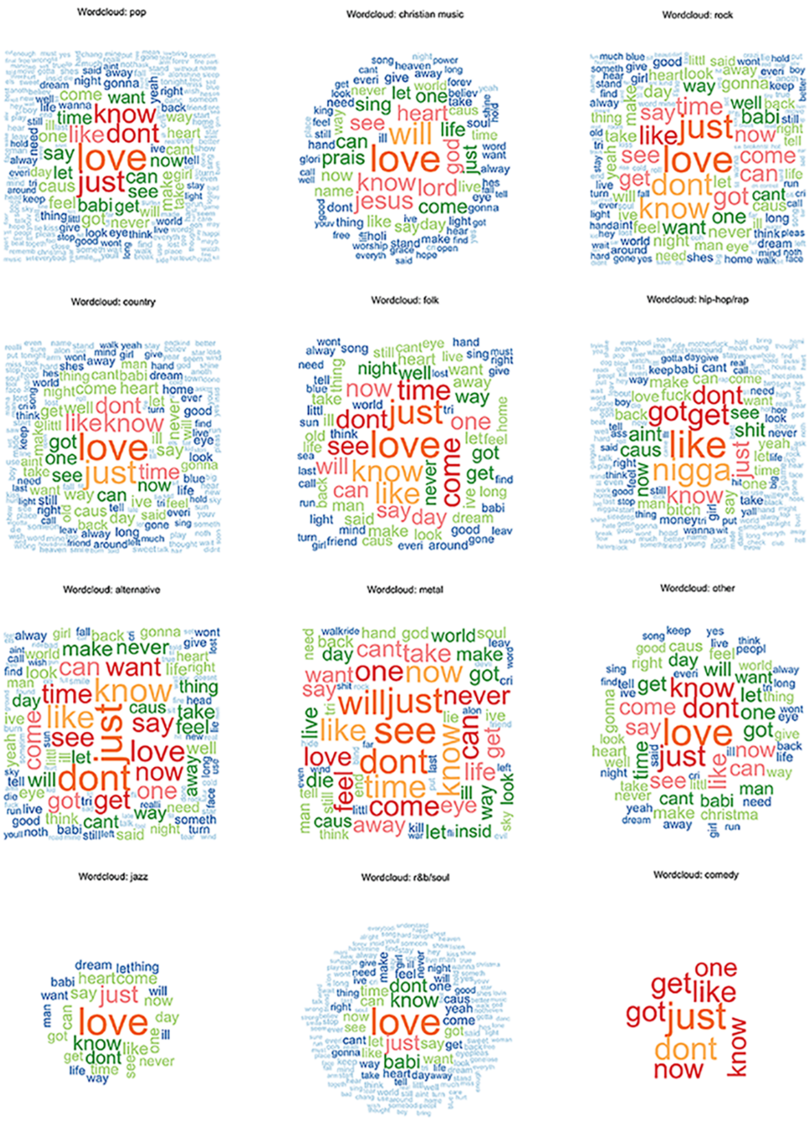
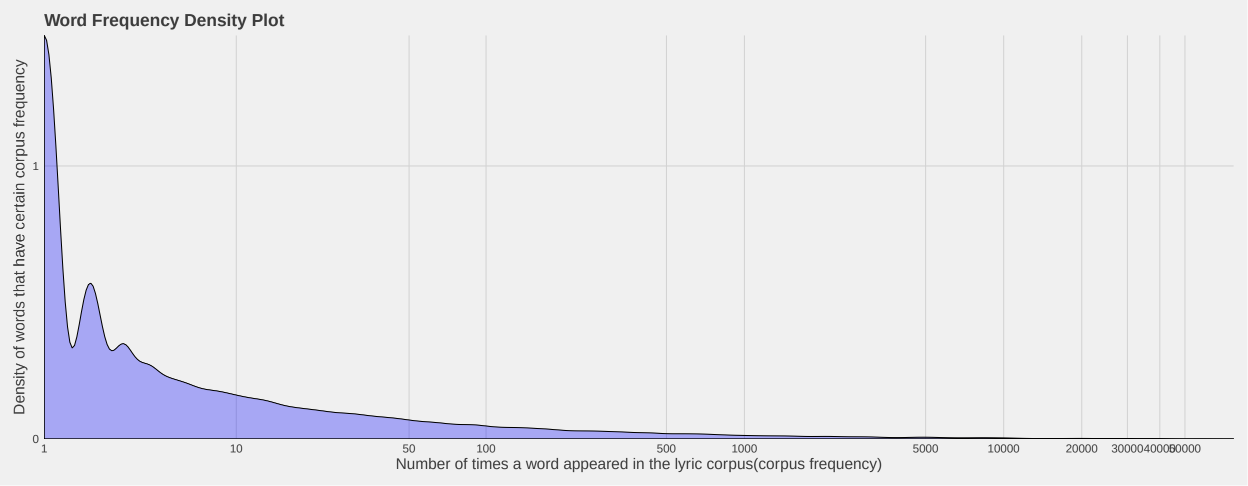


Figure 1:

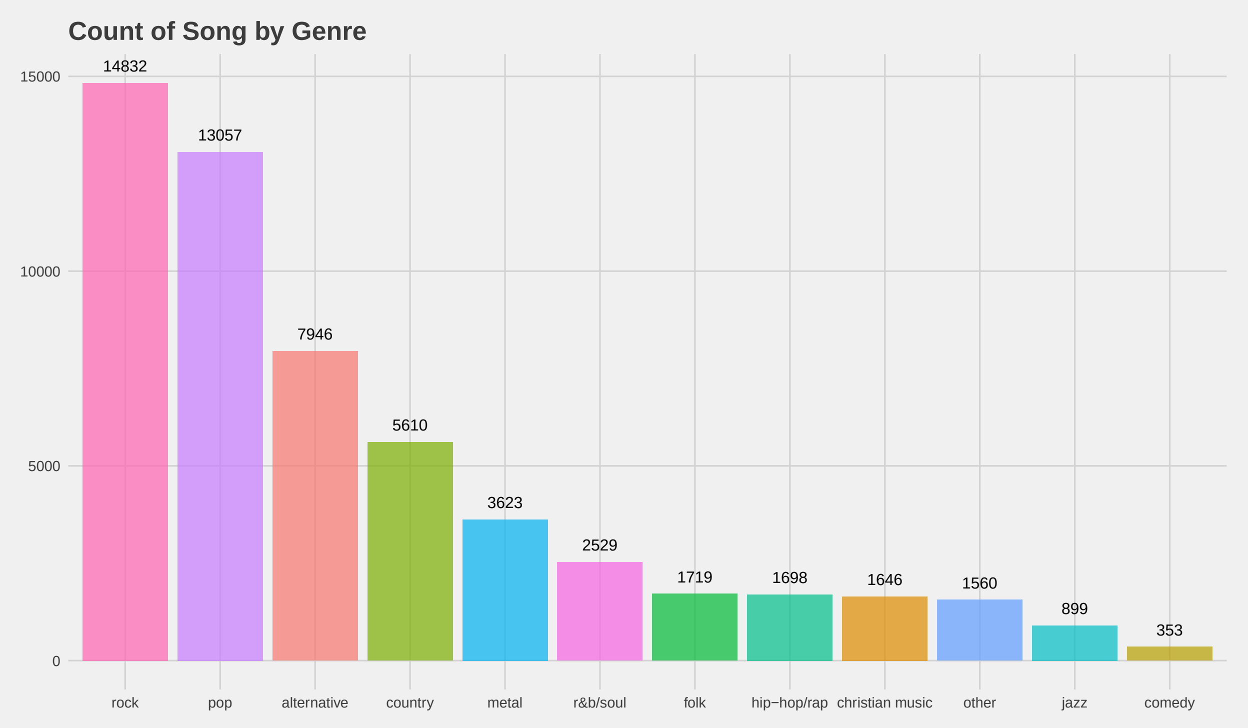
We can see from the above wordclouds that the few genres that are not centered around “love” are hip-hop/rap, metal and comedy.

Word Distribution



The long-tailed distribution shown here is very typical in natural language. The corpus frequency is mostly centered in a small range (1-10) but there is a significant number of words which are in (10-50000).

Genre Count



The dataset is very heavily class-imbalanced, due to the very nature of the means of collecting the data. The original website contains significantly more rock and pop songs than it does other genres. We will discuss the implications of imbalanced data in later parts.

Dimension Reduction

Before we run any classification algorithms, we need to consider the scale of dimensions. Currently there are overwhelmingly 59703 predictors(every unique word in the lyric corpus), which makes it difficult to obtain any accurate and meaningful models. Since we have an extremely long-tailed word frequency distribution as shown in the density plot, meaning that there are lots of zeros in the matrix, we set a sparse threshold above which the term will be removed. After we dropped enough terms that are very infrequent, the resulting matrix has only 294 terms, which is suitable for further analysis.

Classification

Since the multinomial classification task is basically to transform the multiple classes into a one versus all problem, we decided to trim down our problem into a binomial classification problem directly with two genres: pop and chrisitan music. The final matrix is 21039 x 290. The reasons why we chose these two genres are:

- The number of songs in pop genre is about eight times of that of christian music. We want to see if there are any improvements of prediction accuracy if we resampled the data to make it balanced before we feed them into a classifier.
- Pop music and christian music share a lot of common themes in our opinion: love, admiration, lament,etc. We think it would be interesting to see what words contribute significantly to discriminate the two genres.

Classifiers

In terms of classifiers, we used five common classifiers: logistic regression with regulation, classification trees, Random Forest, SVM, and KNN.

Performance Metrics

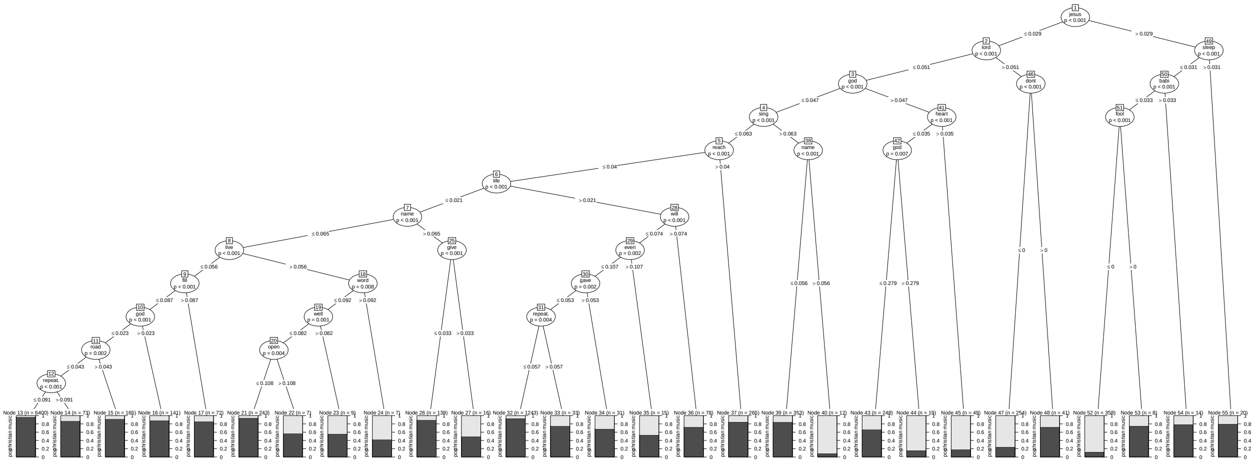
As performance evaluation we will use following metrics:

- Kappa Statistics: this statistic compares observed accuracy with expected accuracy. A kappa value close to 1 indicates that observed accuracy is very high.
- Accuracy: 1-test error rate
- Sensitivity: We set christian music as true positive value, so sensitivity here means the percentage of true christian music songs that are correctly detected
- Specificity: the percentage of true pop music that are identified

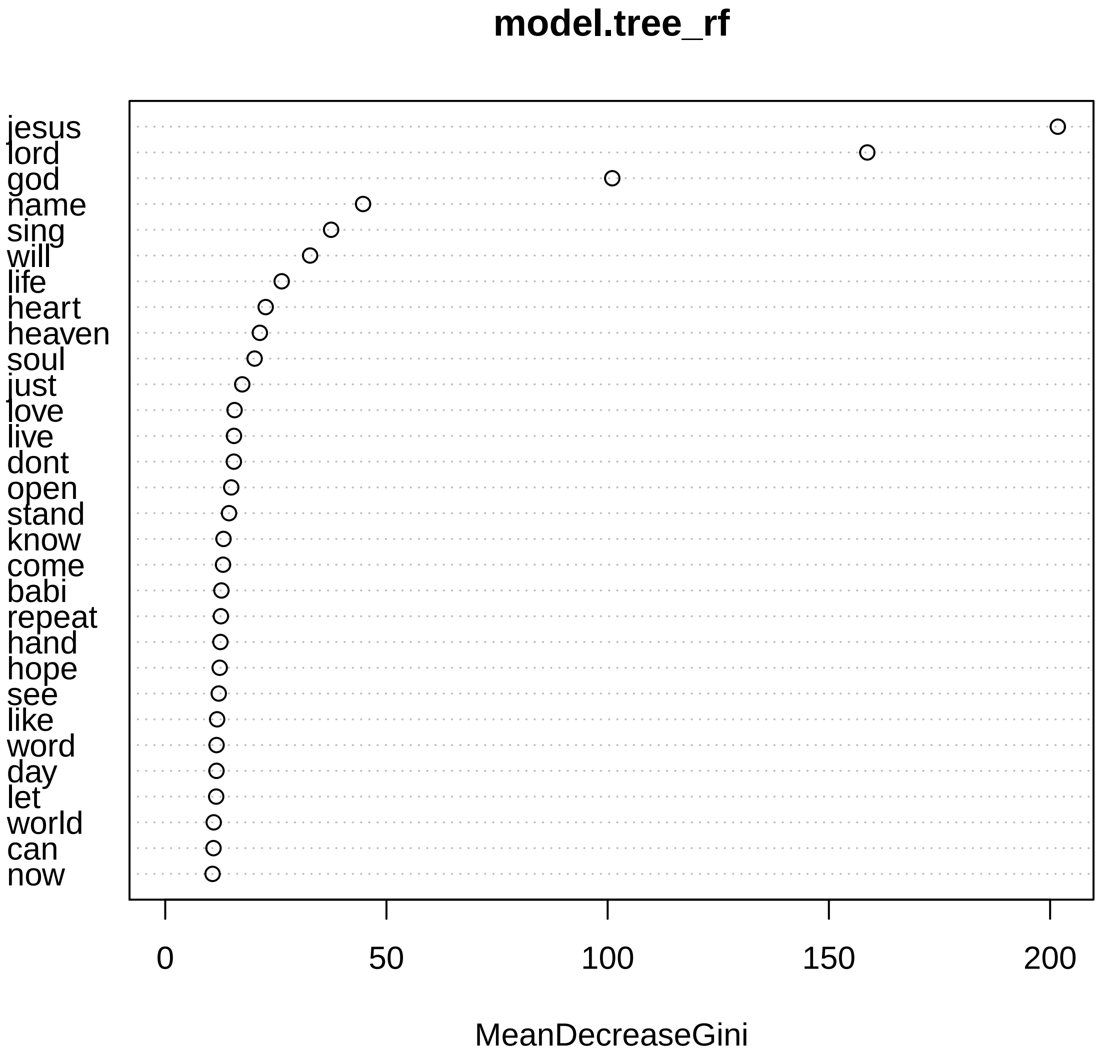
Results

Important Features

From the exploratory data analysis, we have already known some frequent words and important words using wordcloud and tf-idf score. Here we are using tree method to further extract the important words of the pop and christian music that can distinguish pop and christian music better.



Tree is one of the most interpretable and intuitive models. By plotting the nodes of a basic tree, we can see that the root is ‘jesus’, a typical word that frequently appeared in christian music but less frequent in pop music. Each split in the tree is based on a maximum information gain decision, which means that a split on a feature closer to the root has more information gain than a split with a different feature lower in the tree, thus can better split the tree. By following the nodes, we can find the most likely class of the lyric at the end of the tree.

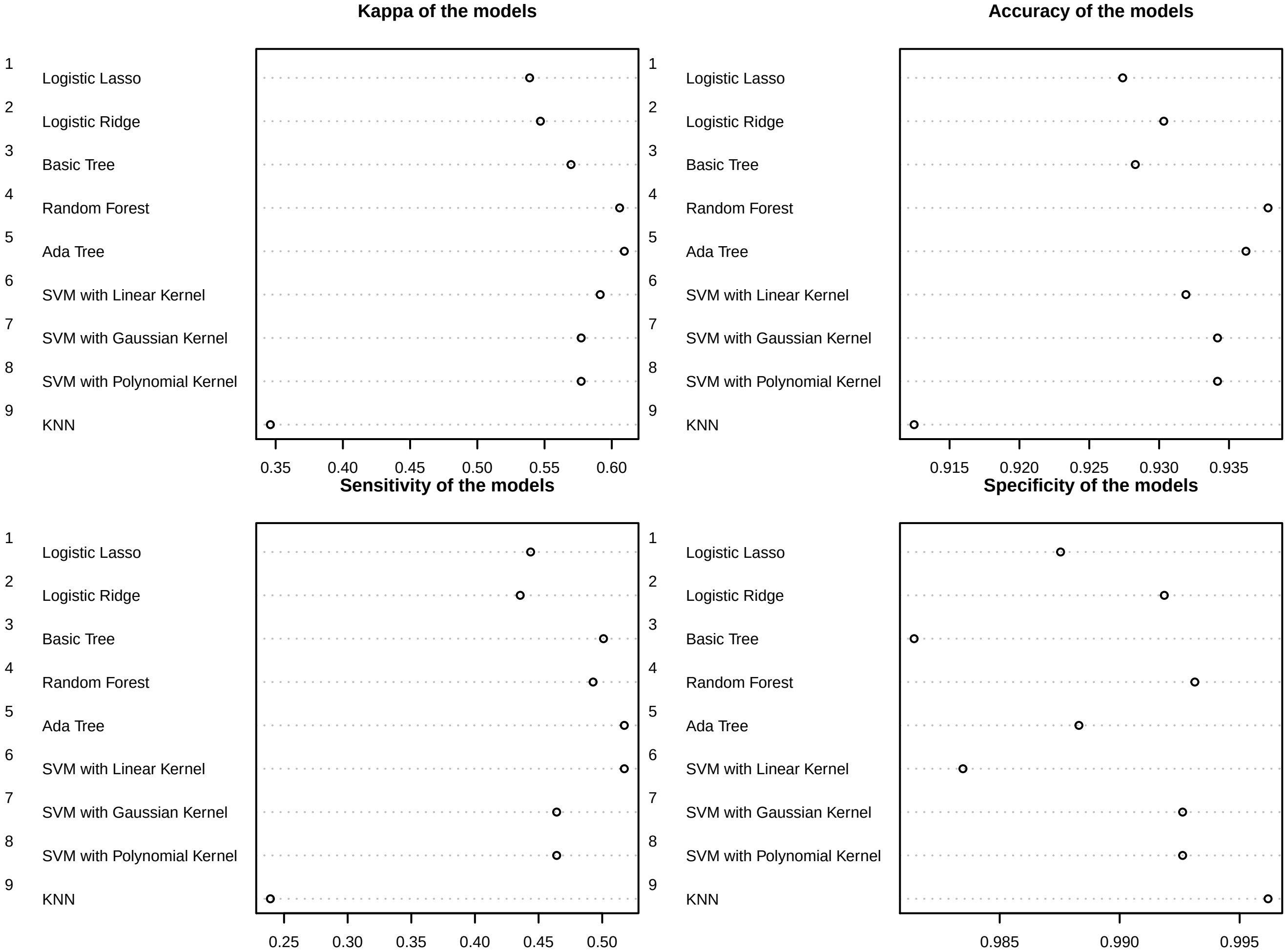


From the important variable graph from our random forest model, we can see a pretty similar result. The importance of a variable illustrates its split quality, which is usually measured by Gini. The 'jesus' here has the highest mean decrease Gini, which means that it has the highest quality of split, and thus is the most important variable. It is followed by 'lord' and 'god'. The mean decrease gets less steep starting from 'name'. Although the threshold of importance is decided by prior information (usually domain experts), from our knowledge, the threshold here could be set at 'heaven', since the later words tend to appear in both pop and christian music.

Model Evaluation

Ignoring the Imbalanced Problem

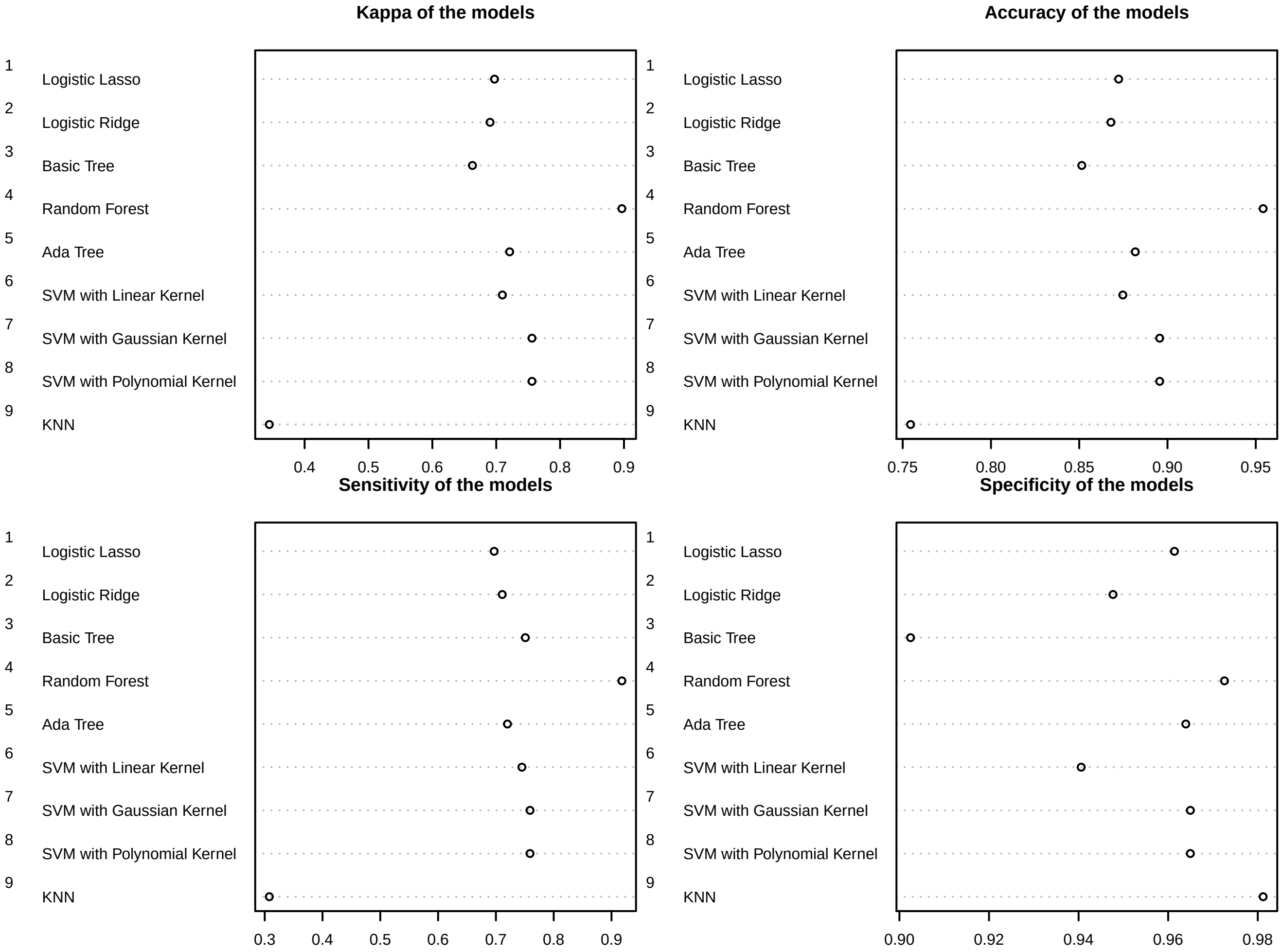
We first ran various classification methods on the imbalanced dataset. For all methods, a percentage split was used to separate training data to build models on and testing data to evaluate them. 70% (10314 instances) of the data was used to train the model, and the remaining 30% (4420 instances) to test it. The values K(10) for KNN, lambda for lasso(0.002) and ridge(0.017) logistic regression were tuned by 10-fold cross validation process. The parameters that generated the lowest test error were used on test dataset.


Kappa of the models
Accuracy of the models
Sensitivity of the models
Specificity of the models

We can see from the above model evaluations that Random Forest performed the best with the highest Kappa statistic and test accuracy rate, followed by ada tree and svm methods. However, the sensitivity performance of these methods are not so satisfactory, while most of the specificity is very close to one. Since we set the positive reference as christian music during evaluation process, low sensitivity values mean that almost all songs are detected as pop genre, and christian music seldom got correctly identified. We'll deal with this problem by undersampling the majority class and oversampling the minority class.

Balance the dataset

Here we duplicated the christian music genre and downsized the pop genre so that the ratio is now 1:2(3298:6596). We again applied the same classification techniques to the more balanced dataset:



We can see that after balancing two genres, the predict accuracy and sensitivity level of each method indeed improved. Still, KNN performed well badly even after we balanced the data, perhaps in a high dimensional space the distance to all neighbors becomes more or less the same, and the notion of nearest and far neighbors becomes blurred.

Conclusion Remarks and Future Work

This project is focused on genre classification between pop and christian music. We discovered important variables that can best distinguish those two genres by using word frequency, tf-idf weighting score, and classification trees. It turns out that the importance measure of classification tree gave us most intuitive features like 'jesus', 'lord', and 'god' that meet our prior knowledge. The trees also provided quantitative scores measuring importance that is more persuasive than just using wordcloud.

We also tried and tuned different classification models and compared their performance. Most of the models have high accuracy above 92.5% except KNN, which meets our expectation because KNN suffers from high dimension curse. It requires much more data in the high dimension to find a neighbor. Among these models, we found that tree-based methods like random forest and ada tree performed best with the highest accuracy and kappa scores. We can also see that SVM and ada tree have good sensitivity on our unbalanced data (pop genre is about 8 times more than christian music). After balanced the dataset so that the ratio of christian song to pop is 1:2, random forest still had the highest performance with accuracy rate, kappa, sensitivity, and specificity.

We'd like to thank our instructor Gun Woong Park and GSI Jack Goetz for offering the chance to work on this project and advising throughout the process.