Using Song Attributes to Classify Popular Songs

CSC 422, Group 9

Jack Kurtz
Department of Computer Science
North Carolina State University
Raleigh, NC USA
jbkurtz@ncsu.edu

Chang Lee
Department of Computer Science
North Carolina State University
Raleigh, NC USA
cnlee2@ncsu.edu

Shengdong Chen
Department of Computer Science
North Carolina State University
Raleigh, NC USA
schen42@ncsu.edu

BACKGROUND

Humanity has created many different forms of self-expression but very few have stood the test of time as well as music. As such, many have strived to create the perfect piece of musical composition. Some have chosen to pursue a formulaic approach - attempting to create some type of reusable pattern capable of generating popular songs repeatedly. This option always seems to fall short to those who opt to create music strictly from "feel" or some type of internal inspiration. The most notable example of this method being used to gain meteoric popularity is the Beatles, when they captured the #1-#5 spots on the USA top singles chart during April of 1964: a feat that has still gone unmatched.[1] And yet, check any current top 100 list of singles in the US and the Beatles are nowhere to be found. This drives up many questions regarding music, popularity, and what it takes to create a popular piece of music. As such, with the music dataset provided from the CORGIS dataset project, our group aims to tackle some of these questions. More specifically, our goal is to test if certain song attributes can be used to predict the popularity of a song.

Music is defined as "the science or art of ordering tones or sounds in succession, in combination, and in temporal relationships to produce a composition having unity and continuity." Many people often simplify this by saying "You know it when you hear it." While these definitions do not exactly help us specify what music is, they do help reinforce the idea that music is composed of multitudes of different attributes. As such, it can be hard to drill down which attributes should be analyzed the most. A few groups have tried this analysis before, and we will be drawing inspiration from each of these groups. R. Mitchell Parry explored the connection between musical complexity and popularity in his paper by looking specifically at instrumental metrics such as rhythm and tone variation, and how these factors play an impact on a song's placement in the Top 40 charts.[2] Dhanaraj and Logan take this concept a step further in their study, where they examine how lyric and acoustic information can impact the popularity of songs.[3] Pham, Kyauk, and Edwin expanded on this idea once again when they studied how song information and artist metadata can influence a song's popularity.[4] All of these studies provided relevant points of interest for us to pursue, which led us to our dataset, the CORGIS Dataset Project music dataset. This dataset contains many important song attributes, artist metadata, and a popularity score - all of which should be sufficient for us to build our classifier and analyze.

Through the analysis of this data, we hope to strengthen some of the concepts learned throughout the duration of this course, such as logistic regression, multilayer perceptrons, and support vector machines. We will also explore some more novel ways to perform feature selection aside from Principle Component Analysis, such as forward selection and backward elimination stepwise regression. Furthermore, the use of a publicly available dataset such as the one from CORGIS Dataset Project means that our results can be compared to many other similar analyses who choose to use the same dataset. Additionally, our personal interest in the dataset provides a more profound motivation for us to build appropriate models. Finally, we believe that there are multiple different parties that would be interested in our results. Obviously, our group and other music lovers can gain a deeper understanding of what makes our favorite songs so popular. Other artists, creators, or producers will also be able to use this analysis to learn how to create effective music pieces that will become popular. Media providers such as Spotify, Pandora, or radio stations will be able to identify patterns in popular songs to determine what their listeners want to hear or even be able to identify potentially popular songs before they are in the spotlight.

METHODS

For this project, the following machine learning techniques will be used to help us determine which attributes are important for classifying song popularity and how we can classify songs by popularity based off of these attributes

Stepwise Regression

Stepwise Regression is a statistical technique to fit a regression model that identifies important attributes. The main reason to

[1] Ruth Dhanaraj, Beth Logan. 2005. Automatic Prediction of Hit Songs.
[2] R. Mitchell Parry. 2004. Musical Complexity and Top 40 Chart Performance.
[3] Ruth Dhanaraj, Beth Logan. 2005. Automatic Prediction of Hit Songs.
[4] James Pham, Edric Kyauk, Edwin Park. 2016. Predicting Song Popularity.

choose this algorithm is to reduce the set of attributes from the dataset. There is a total of 35 attributes in the dataset. By adding or removing the variables based on p values or likelihood, the stepwise regression technique can help extract the attribute data and determine some of the most important attributes that affect the song's popularity effectively so that our team can start the experiment with a relatively small set that contains most of the important information from the large set. Our team decided to use two different types of stepwise regression to use for this project: Forward Selection and Backward Elimination.

1. Forward Selection

Forward Selection determines the most important attributes by adding each attribute from the model that does not contain any variable initially. When the attributes are added, the forward selection model can determine the statistical improvement that the certain added attribute gives. For each iteration of forward selection, the model compares each p-value of each attribute and picks the attribute that has the lowest p-value. If that value is higher than the significance level chosen as a boundary, the model adds the attribute to the model and moves to the next iteration until the addition of all the significant variables.

2. Backward Elimination

Backward Selection determines the most important attributes by removing each attribute from the model that contains all the variables initially. Unlike forward selection, backward elimination can determine the loss of statistical fit when a certain variable is removed. For each iteration of backward elimination, the model picks the attribute that has the largest p-value. If that p-value is bigger than the significance level chosen as a boundary, the model removes the attribute and moves to the next iteration until the removal of all the insignificant variables.

Classification Models

To make the most accurate predictions of the song's hotness, our team decided to construct the three learning models using three different approaches: logistic regression for classification, multilayer perceptrons, and support vector machines. By comparing three different models, our team can choose the model that can give the best performance.

1. Logistic Regression with 10-Fold Cross Validation

Logistic regression classification is one of the most accurate and popular classification algorithms of all other classification algorithms. It can find the relationship between input variables and target variables even in the case of binary classification. To avoid overfitting and reduce model complexity, logistic regression uses a regularization technique, Lasso (1) or Ridge regression (2).

$$\sum_{i=1}^{n}(y_i-\beta_0-\sum_{j=1}^{p}\beta_j x_{ij})^2+\lambda\sum_{j=1}^{p}\beta_j^2 = RSS+\lambda\sum_{j=1}^{p}\beta_j^2 \quad (1)$$

$$\sum_{i=1}^{n}(y_i-\beta_0-\sum_{j=1}^{p}\beta_j x_{ij})^2+\lambda\sum_{j=1}^{p}|\beta_j| = RSS+\lambda\sum_{j=1}^{p}|\beta_j| \quad (2)$$

By those regularization techniques, we can find the best point between bias and variance that minimize the test mean squared error.

For the validation, we use k-fold cross validation with a k value of 10, which is the value commonly used for applied machine learning. It evaluates the created predictive model by partitioning the data into a training set and test set to evaluate. Our team chose k-fold cross validation because it is great to reduce bias and relatively faster computation.

2. Multilayer Perceptron

Multilayer Perceptron is a feedforward artificial neural network class that contains multiple layers of perceptrons, a neural network unit responsible for binary classifier calculations. To create the second learning model, our team uses a multilayer perceptron with one hidden layer, a layer between input and output layers for additional nodes. Multilayer perceptrons are very useful to extract the pattern and detect the trend of input data so that the model successfully can predict the result.

3. Support Vector Machine with SVM Tuning

For the last machine learning model, the support vector machine is used to analyze and classify the data. It calculates and defines the best decision boundaries for classification. For this, it uses support vectors, the data points near the decision boundary to see the margin, the distance between support vectors and boundary because the best decision boundary maximizes the margin. In the case of a data set that cannot be separated by linear decision boundaries, we tried four different kernels, linear, radial, polynomial and sigmoid for support vector machines to obtain the best results.

4. Confusion Matrix

A confusion matrix is a table describing the performance of the classification model. Our team uses a confusion matrix to evaluate the performance of three models constructed by Logistic Regression, Multilayer Perceptron, and SVM. We chose the confusion matrix because it is effective especially for representing the accuracy performance of the model.

PLAN & EXPERIMENT

Dataset

Our sole dataset comes from the CORGIS Dataset project and is titled ‘music.csv’. This dataset contains 35 different attributes about songs, ranging from metadata about the artist, such as artist hotness or familiarity, to acoustic data, such as the loudness or tempo of the song. The key attribute is song hotness, which is the measure of song popularity at the time of data collection (December 2010). These scores can range from 0-1. We used this attribute to determine whether or not a song could be considered popular. The remaining attributes will be used to form our models.

Hypothesis

Ultimately, we hypothesize that the multilayer perceptron will have the highest accuracy, precision, recall, and F1 scores, as was found by Pham et al.[5] This model is the best at identifying obscure patterns in data and using the dataset to adapt the model, which will be necessary in an obscure problem such as this one.

To predict if a song is hot, we followed 4 steps listed below:

1. Data Preprocessing

Our original dataset has 35 attributes for 10,000 unique objects. The goal of this project is to classify whether or not a song is popular based on the song hotness attribute. Therefore, we started by removing all objects that did not have a valid song hotness score. A song with a hotness score of 0 or below was considered invalid. This left us with 4214 songs. We took the top 10th percentile of songs to be our popular class, with the remaining being considered not popular. As such, songs with a song hotness score of above 0.685 were considered popular.

2. Feature Selection

For our feature selection, we first started with principle component analysis. However, as discussed in our results section, this method was not sufficient for selecting vital features, so we moved forward with forward selection stepwise regression and backward elimination stepwise regression to reduce the number of features in our dataset.

This was accomplished using the *step* function in R. Forward selection begins with an empty list of attributes, whereas backward elimination begins with a full list of all 34 attributes. The *step* function produces a list of attributes that it has determined as important for determining the final class attribute.

3. Building Models

We split our list of 4214 songs into a training and testing dataset, with our training dataset consisting of 3214 songs and our testing dataset consisting of the remaining 1000 songs. Our training data was used to build our models and tune our parameters. The testing data was used to produce our final results.

The first type of model we built were logistic regression models. This was accomplished using the *glm* function. For our classification, we needed the output to be represented as a binomial – either popular or not popular. When training each model, we use 10-fold cross validation to help reduce overfitting. We tested each model using different alpha values – 0, 0.5, and 1. These alpha values represent the types of regularization to use, with 0 meaning ridge regression and 1 meaning lasso regression.

Once a model was produced, we tested it against our test dataset and compared the results by looking at the confusion matrices.

The second kind of model we built is a multilayer perceptron. This was accomplished with the *monmlp* library in R. We chose a multilayer perceptron with a single hidden layer consisting of 2 nodes. The ensemble size for training was 500 iterations. It is important to note that the *monmlp* library uses its own activation function and cannot be modified.

There were no hyperparameters that we could experiment with when creating the multilayer perceptron. However, this specific library is not capable of producing binomial classifications. Instead, it produces a weight vector that represents the networks numerical output for each test object. Thus, we had to experiment with using different output values as our threshold for determining whether or not a song was popular. For this output threshold, we used the following values: 0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, and 0.9, as our model only produce weights that ranged from 0 – 1.

The third type of model we built were support vector machines models for classification and regression analysis. Basically, we build 4 models based on SVM's kernel tricks: "linear", "radial", "polynomial" and "sigmoid". When creating the models, we used the *svm* function in R.

Each kernel type had its own set of parameters that could be optimized. These variables were cost, gamma, and degree. Cost is used by all kernels to control the cost of a misclassification on the training data. Gamma is a parameter for non-linear hyperplanes that defines how far the influence of a single training example reaches. Degree is only used for polynomial kernels and it represents the degree of the polynomial used to split the hyperplane. We used the *tune* function of R to rapidly test for optimal parameters in all of our kernels. We tested against the cost values of 0.01, 0.1, 1, and 10, the gamma values of 0.05, 0.5, 1, and 2, and the degree values of 1, 2, 3, 4, and 5.

4. Performance Observation

Finally, we observed the performance of each model we build based on their confusion matrices, looking specifically at accuracy, precision, recall, and F1 scores. This was accomplished using the *confusionMatrix* function from the *caret* library.

RESULTS

Feature Selection

Our group originally anticipated using just Principle Component Analysis for our feature selection. During our analysis, PCA only

[5] James Pham, Edric Kyauk, Edwin Park. 2016. Predicting Song Popularity.

produced one component with a significant eigenvalue, with the rest being less than zero. Furthermore, the weights for all attributes in this component were very similar and all were less than zero. This suggested that every attribute was equally important, although their overall level of importance was low. In an effort to find alternative approaches to feature selection, our group also decided to perform forward step selection and backward step selection – two novel methods for feature selection as discussed in "Predicing Song Popularity."[6] Of our original 35 attributes, forward step selection picked out 17 attributes as influential, see in table ____. Backward selection only picked out 12, shown in table ____. Our plan was to use the common attributes between both backward and forward selection as the attributes to build our models with, but they only selected 3 of the same attributes – reinforcing the results of PCA – that each attribute had roughly the same impact on the popularity of the song. As such, we decided to create different models from each separate list of attributes and using the full set of attributes for comparison. From here, we formed an additional hypothesis. We predicted that the models built using the features generated from backward elimination would perform the best, as this attribute list seemed to contain more meta data as opposed to acoustical information. Meta data has been shown to be more important to predicting song popularity than acoustic information.[7]

Table 1. Features selected with Stepwise Regression.

Forward Selection	Backward Selection
Song mode	Song tatums confidence
Song key	Artist latitude
Release name	Song tempo
Song end of fade in	Song artist mbtags count
Song time signature confidence	Artist terms freq
Song bars start	Song duration
Song time signature	Song start of fade out
Song key confidence	Song release id
Song title	Song loudness
Song start of fade out	Artist hotness
Song beats start	Artist familiarity
Song mode confidence	Song year
Artist longitude	
Song bars confidence	
Song duration	
Song tatums start	
Song beats confidence	

Classification Models

As pointed out by Pham et al., accuracy was not the only metric by which we could judge our models.[8] We chose to classify all songs whose popularity score fell within the top 10 percentile as our "popular" songs. As such, any model could achieve 90% accuracy by classifying all songs as not popular – and some did produce these results, as we will discuss later. Due to this, we had to use other metrics to compare the performance of our models. Aside from accuracy, we also used precision, recall and F1 score to determine how good our models were.

In the context of this project, precision represents the amount of songs that were correctly classified as popular out of all of the songs that were labeled popular. Recall represents the amount of songs that were correctly classified as popular out of all the songs that are popular from the test data set. F1 score gives us an average between these two values. Using all of these metrics give us a better method for determining which model performed the best.

Logistic Regression

For logistic regression, we tested each of the models with varying alpha values. We found that an alpha value of 1 produced the best results for each model. In context of R code, setting our alpha value to 1 means we performed lasso regularization when creating the model. Overall, the model with attributes from our backward stepwise selection performed the best, with higher results for each metric, as seen in Table 2. This is also where we ran into our first example of a model classifying all items as not popular because that still produced a high accuracy, as the model created with attributes from our forward stepwise selection did exactly that.

Table 2. Metrics for Logistic Regression Models.

Attributes	Accuracy	Precision	Recall	F1-Score
All	0.911	0.484	0.153	0.246
Forward	0.902	0.0	0.0	0.0

[6] James Pham, Edric Kyauk, Edwin Park. 2016. Predicting Song Popularity.

[7] James Pham, Edric Kyauk, Edwin Park. 2016. Predicting Song Popularity.

[8] James Pham, Edric Kyauk, Edwin Park. 2016. Predicting Song Popularity.

Backward	0.912	0.679	0.194	0.302

Multilayer Perceptron

For our multilayer perceptron, we tested with different probability thresholds for the output at our final node. We found that the best threshold was at 0.4, classifying items that received greater than 0.4 as popular, and songs that received less than 0.4 as not popular. The models created with the full list of attributes and the attributes selected from backward stepwise selection were competitive, with the full list model having a higher accuracy and precision, but the backward stepwise model having a higher recall and F1 score, as shown in Table 3. Once again, the forward stepwise model classified every song as only not popular.

Table 3. Metrics for Multilayer Perceptron Models.

Attributes	Accuracy	Precision	Recall	F1-Score
All	0.905	0.579	0.112	0.188
Forward	0.902	0.0	0.0	0.0
Backward	0.893	0.424	0.255	0.318

Support Vector Machine

For our support vector machines, the full list of attributes could not be used due to the constancy of some of the attributes. We decided to focus on only the models created with the forward and backward selected attributes and tested against the four basic kernel types with varying levels of parameters. In the end, the svm created using the backward selected attributes and a radial kernel with a cost of 1 and a gamma value of 0.5 performed the best, scoring highest in all metrics, as shown in Table 4. However, this is also where we encountered the greatest level of failure with our models, as every other kernel for the backward selected attributes and every svm for the forward selected attributes created a model where every song was classified as not popular.

Table 4. Metrics for Support Vector Machine Models.

Attributes	Accuracy	Precision	Recall	F1-Score
Forward	0.902	0.0	0.0	0.0
Backward	0.905	0.636	0.071	0.128

Overall

Overall, our highest performing model was the model created with linear regression using the backward selected attributes. This model had the highest value for each metric overall, meaning it correctly identified the most popular and not popular songs without classifying every song as not popular. Our results do not support our original hypothesis. It appears that the logistic regression model performed better in this situation, although it was just slightly.

It also appears that forward selection did not choose many appropriate attributes for classifying songs, as every model that used these attributes just defaulted to classifying all songs as not popular. When you look at the actual attributes selected by each feature selection, this makes some logical sense. The attributes selected by backward selection are ones that we expect to have an impact on song popularity – such as the artist popularity, song tempo, duration, and loudness. With this data in mind, it appears that our secondary hypothesis can be supported.

DISCUSSION

The results of these models show that it is possible to classifying songs by their popularity based on certain attributes. We can achieve a high accuracy without simply classifying all songs as not popular. This is consistent with previous studies, such as Pham et al.[9] This study also found that multilayer perceptrons and logistic regression classified these songs well, although their support vector machines performed much better than ours. In a broad sense though, we can say that artists, media companies, and music lovers can use song attributes to classify songs by popularity and potentially guess the future popularity of songs based on these attributes.

We can also use these results to evaluate which song attributes help contribute to song popularity. As expected, the artists popularity and familiarity played a role in determining a songs popularity, as well as basic song attributes such as song tempo, duration and loudness. However, it appears that some lesser known attributes such as the songs fade in or tatums can also play a heavy role. Future producers or song creators should keep these features in mind when attempting to create their own popular music.

This study could be improved by increasing the pre-processing and feature selection. Thirty-one attributes is not a lot when compared to other studies. Additionally, the format of some of the attributes could have contributed to their overall impact (or lack thereof) on the song popularity. By binning or bagging some of the attributes, we may have been able to improve our model further. However,

[9] James Pham, Edric Kyauk, Edwin Park. 2016. Predicting Song Popularity.

our research shows that classification of these songs is possible and provides a baseline for what these models are capable of achieving.

CODE

All of our code for this project can be found at: https://github.ncsu.edu/jbkurtz/csc-422-final-project.

MEETING ATTENDANCE

Date	Time	Attendee
4/6/20	3-4PM	Jack Kurtz, Chang Lee
4/10/20	3-4PM	Jack Kurtz, Chang Lee, Schengdong Chen
4/13/20	3-4PM	Jack Kurtz, Chang Lee, Schengdong Chen
4/17/20	3-4PM	Jack Kurtz, Chang Lee, Schengdong Chen
4/20/20	3-4PM	Jack Kurtz, Schengdong Chen
4/22/20	3-4PM	Jack Kurtz, Chang Lee, Schengdong Chen

REFERENCES

[1] James Pham, Edric Kyauk, Edwin Park. 2016. Predicting Song Popularity. *5 Pages*. http://cs229.stanford.edu/proj2015/140_report.pdf

[2] R. Mitchell Parry. 2004. Musical Complexity and Top 40 Chart Performance. In *GVU Technical Reports, 26 pages.* http://hdl.handle.net/1853/50

[3] Ruth Dhanaraj, Beth Logan. 2005. Automatic Prediction of Hit Songs. In *ISMIR, 4 pages.* https://ismir2005.ismir.net/proceedings/2024.pdf