Using Data to Predict the Next Hit Album - Applying Machine Learning

There are two topics we will address using machine learning algorithms. The first topic is using musical data to predict numeric review scores, from which we hope to learn what aspects of music are the most important for critical acclaim. The second topic is attempting to classify albums into genres using their musical attributes. In both topics, we will start by using relatively more simple ML algorithms, and then see if we can improve the models by tuning hyperparameters or selecting other models.

Our regression problem is as follows: using all of the information available about an album, what score do we predict critics to review the album with?

To answer this question, we will start with using linear regression. To help improve our model, we will make some transformations to the data. We will square the review scores and attempt to predict this score squared value, which is done so that the variable we are trying to predict follows a more normal distribution. Additionally, we will scale all of our variables so that they have a mean of 0 and standard deviation of 1. These changes help the models run more smoothly, but also help us interpret the results more easily when all of the variables are set to the scale and thus are easier to compare. Finally, for the regression problem, we will separate the data by genre and create models for each genre independently, so that differences between genres are properly captured.

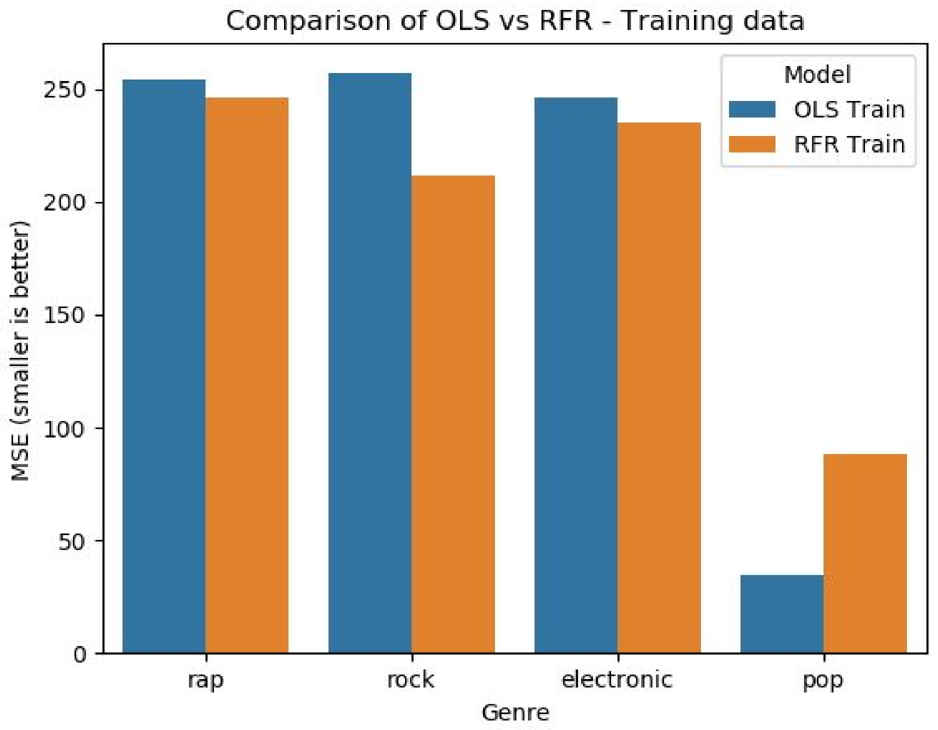
For the linear regression models, there are a few elements we will focus on. First, we can check the performance of each genre's model by looking at the F-score and adjusted R squared to see how well the model handles the data. Second, we can look at the coefficients and p-values associated with each variable in the model to see what lessons can be learned about the importance of individual musical attributes (ie, for each genre, which musical attributes are found to be significant and in what direction).

For the overall performance of our linear regression models (plural because there is one model for each genre), our results are mixed. Looking first at the overall F-score, which checks to see if at least one variable is statistically significant, nearly every model is significant at alpha = 0.05, and some have very high F scores. However, for all models, the adjusted R squared is very small - the highest, metal, has a value of 0.11, meaning only a tiny amount of the variance in review scores is successfully explained by our model. These adjusted R squared values are smaller than 0.1 for every other genre, showing how limited the predicted powers are for these models.

While the models overall do not have much predictive power, we can still learn from them. There are a few variables for some genres with sizeable coefficients. From these, lessons can be learned about what critics respond positively to across different genres of music.

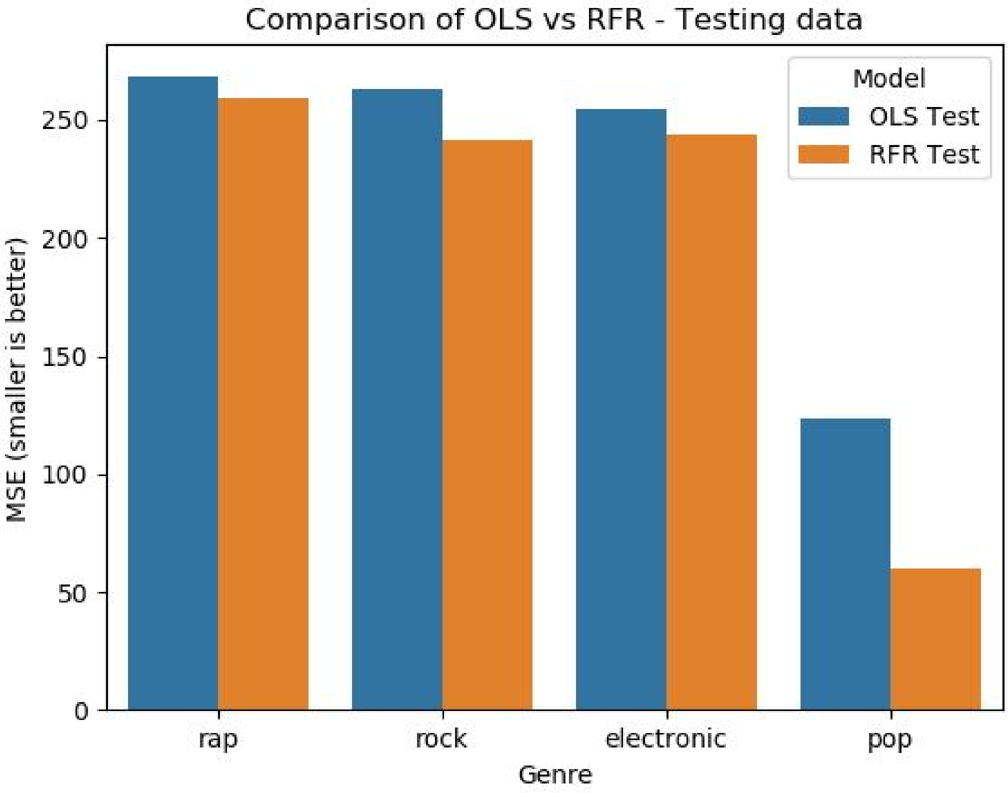
Genre	Positive Attributes	Negative Attributes
Rock	● Acousticness ● Instrumentalness ● Valence	● Danceability ● Minor Key
Metal	● Instrumentalness	● Danceability
Jazz	● Speechiness	● Composed in the key E
Folk/Country	● Acousticness ● Minor Key	
Experimental	● Longer duration ● Albums with variety in acousticness	
Electronic	● Instrumentalness ● Albums with variety in danceability	

Having fit OLS models to each genre, we'll now compare how OLS does with a more sophisticated machine learning algorithm - Random Forest Regression. First, for each genre, we'll use sklearn's GridSearchCV to test a wide variety of hyperparameters on our training data, selecting the set which does the best for each genre. Then, to compare the predictive power of the RFR models with the OLS models, we'll calculate a measure of performance in the Mean



Squared Error (MSE), and compare. Since MSE is a measurement of model error, smaller values indicate a better model.

On the training data, RFR outperforms OLS on three genres, but doesn't do as well on pop music. From this, we could either choose to use the RFR for all genres, or use OLS on pop music and RFR for the other genres. I would lean towards using RFR for all genres, as conceptually it seems more likely that OLS' very low MSE on pop music is a quirk of the training data, and may not necessarily do as well on data the model wasn't trained on. Looking at the test data, RFR ends up doing better on three genres, but does worse on electronic music. Again, I'm inclined to believe that, given a large volume data our model hasn't seen, RFR would outperform OLS in all genres.

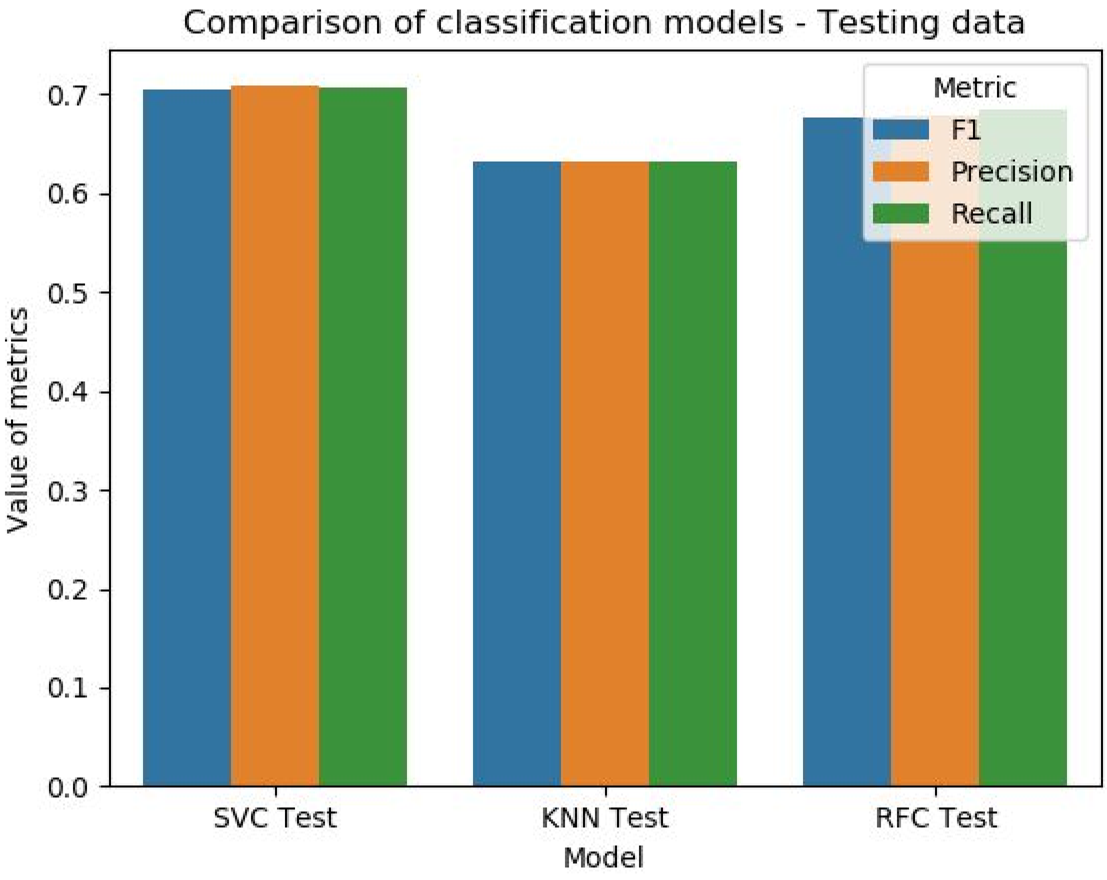


Next, we'll turn our attention to the classification problem. In this, we want to use all of our musical data to predict what genre an album falls into. We'll attempt a few different models and see what ends up handling this problem the best. The models that we'll use will be Support Vector Classification, K-Nearest Neighbors, and Random Forest Classification. For each of these, we'll use sklearn's GridSearchCV to test a wide variety of hyperparameters for each model, and establish what set of hyperparameters is the best for each model. Then, we'll compare the performance of our three model types to determine which is best suited for our task.

First, there is some data preprocessing necessary for this specific task. As was the case in the regression context, we'll scale all the numeric variables to be standardized to have mean 0 and standard deviation 1. The primary challenge created by the data is that while there are nine genres of music in the data, some genres have far more music than others. This creates a situation where, since a majority of the data is rock music, models can predict 'rock' far more than anything else, and end up correct not because it is finding insight in the data, but rather because most of the music in our data happens to be rock. In order to handle this, we'll keep only the top four genres, which each have at least 500 albums each, in for our analysis. Then, we'll sample 500 albums from each genre, so that we have the same amount of data for each genre. This is the data we'll use for the test/train data split, and for all of our classification.

To compare the three different models, we have a combination of metrics to use. Since we have four classes, we cannot create an ROC curve or use AUC. Instead, we'll look at accuracy, specificity, recall, and F score. After evaluating these metrics for each model, we'll make a selection about which has the overall best performance.

We'll use the testing data to compare models, as using random forest classification on the data it was trained on does not lead to accurate results.



On the test data (which the model was not fit on), support vector classification ends up doing the best of our three models, and so we will continue with SVC as our model of choice. To see how our selection ends up doing, we can look at a confusion matrix to see where the problems are arising. Ideally, we want to see very large numbers on the main diagonal, and zeroes everywhere else.

	Predicted Electronic	Predicted Pop/R&B	Predicted Rap	Predicted Rock
True Electronic	145	18	4	26
True Pop/R&B	33	91	9	61
True Rap	12	11	182	4
True Rock	17	37	2	148

From this, it is clear that the model handles some genres better than others. Pop/R&B gives the model the most trouble, which may be due to it being a broader label for a genre than the others. Outside of pop/R&B, the model has some trouble differentiating between electronic and rock albums, but has almost no trouble at all with correctly identifying rap music. But ultimately, it seems the model does a respectable job classifying albums into genres.