598: Machine Learning Project

Application of Classification Methods on Spotify Data

Antonio Campbell, Sinta Sulistyo, Atta Ullah, Penny Wu

4/30/2021

1 Introduction

In this project we have considered a dataset of 2017 songs from Spotify. This data was provided by Spotify and posted by user GeorgeMcIntire on Kaggle ("https://www.kaggle.com/geomack/spotifyclassification"). Within the data, each song has 16 features. The feature of interest for classification is called `target` which indicates whether the creator of the dataset liked or disliked a song. A song is labeled "1" if it is liked and "0" when it is disliked. The other features include `acousticness`, `danceability`, `durationMs (duration in milliseconds)`, `energy`, `instrumentalness`, `key`, `liveness`, `loudness`, `mode`, `speechiness`, `tempo`, `time-signature`, `valence`, `songname`, `artist`.

The goal of the project is to build several classifiers for prediction that is based on the rest of the features to determine whether the individual would like a song. We begin by preparing the data to fit possible models appropriately.

2 Data exploration and feature selection:

Data Sample

X	acousticness	...	target	song_title	artist
0	0.0102	...	1	Mask Off	Future
1	0.199	...	1	Redbone	Childish Gambino
2	0.0344	...	1	Xanny Family	Future
3	0.604	...	1	Master Of None	Beach House
4	0.18	...	1	Parallel Lines	Junior Boys
5	0.00479	...	1	Sneakin'	Drake
⋮	⋮	⋮	⋮	⋮	⋮
2011	0.000586	...	0	Brightside - Borgeous Remix	Icona Pop
2012	0.00106	...	0	Like A Bitch - Kill The Noise Remix	Kill The Noise
2013	0.0877	...	0	Candy	Dillon Francis
2014	0.00857	...	0	Habit - Dack Janiels & Wenzday Remix	Rain Man
2015	0.00164	...	0	First Contact	Twin Moons
2016	0.00281	...	0	I Wanna Get Better	Bleachers

Data Preparation

We ensure the data is adequate for fitting before we begin modeling. In particular we are looking for features which have missing values, the songs that are duplicated in the dataset, the types of features (if a feature is numerical or categorical), and as well as the variables of importance for the fit.

Data Dimensions

	rows	columns
count	2017	17

We drop the first column, which is just for indexing and has no use to us here. The data has the remaining features

```
##  [1] "acousticness"     "danceability"     "duration_ms"      "energy"
##  [5] "instrumentalness" "key"              "liveness"         "loudness"
##  [9] "mode"             "speechiness"      "tempo"            "time_signature"
## [13] "valence"          "target"           "song_title"       "artist"
```

Next we check if there are any missing values within the data and we find that there are 0 records with missing data in them; however, there are 5 duplicate records:

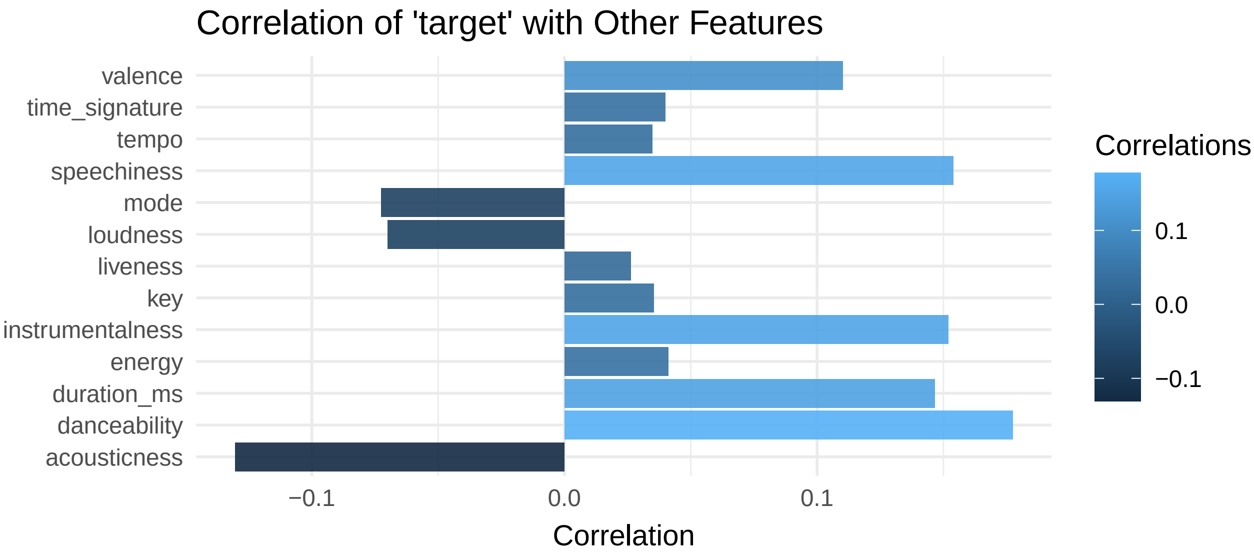
Duplicated Records

	acousticness	danceability	...	song_title	artist
268	0.096200	0.654	...	River	Ibeyi
509	0.024600	0.586	...	Her Fantasy	Matthew Dear
895	0.000334	0.907	...	Jack	Breach
928	0.934000	0.440	...	Episode I - Duel of The Fates	John Williams
982	0.036900	0.448	...	Myth	Beach House

We remove the 5 duplicate data points and the data that 2012 data points remain.

Correlation among all features with Target

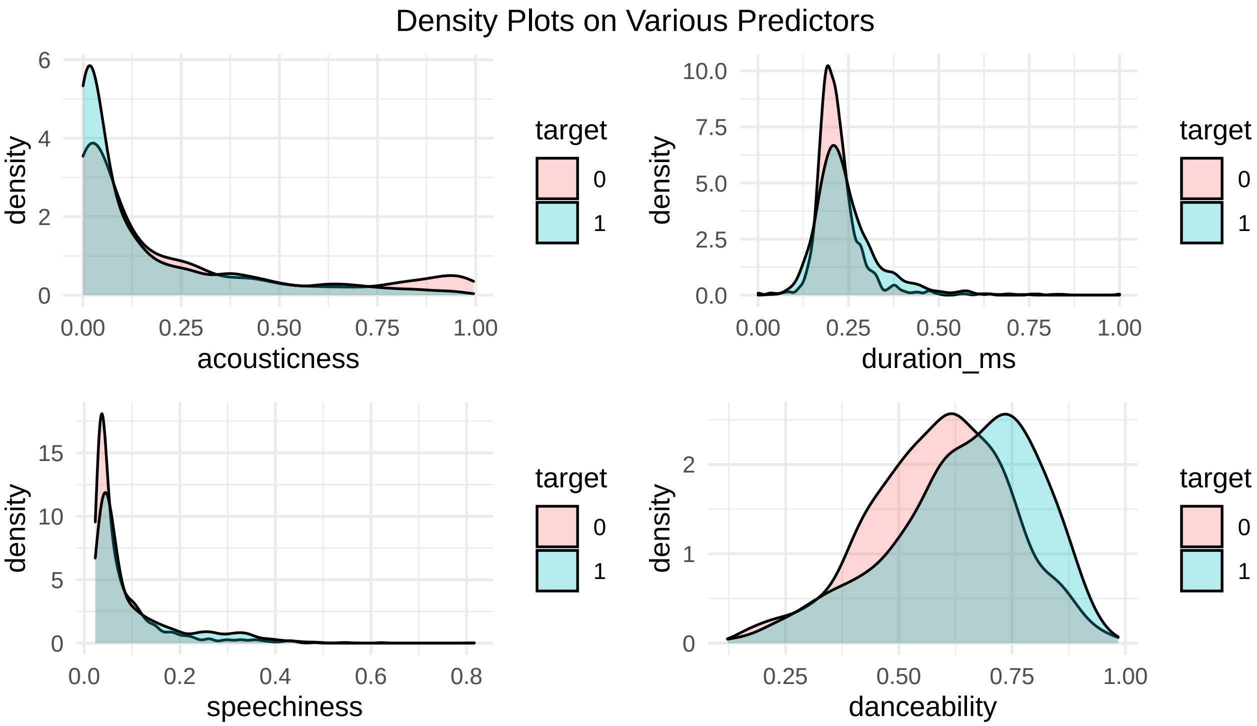
We include a correlation heat plot. For this analysis we have dropped the last two features 'song_title' and 'artist_name', which we do not use in our final analysis as well as those previously dropped.



This plot tells us that `instrumentalness`, `danceability`, `speechiness`, and `acousticness` are among the most correlated with the the target variable. Additionally, `duration_ms`, `instrumentalness` and `valence` relatively higher correlations as compared to the remaining features.

Covariation between individual variables and target

We also want to have a look at how individual variables may vary with `target` by plotting the density of some features grouped by the two levels of `target`. We plot the density of `acousticness`, `duration_ms`, `speechiness`, and `dancebility`. We see these features do have some dependency or covariation on our response variable `target`.The plots also justify the results from our correlation heat plot. We will then choose features with stronger correlations to the response variable for our predictive modeling.



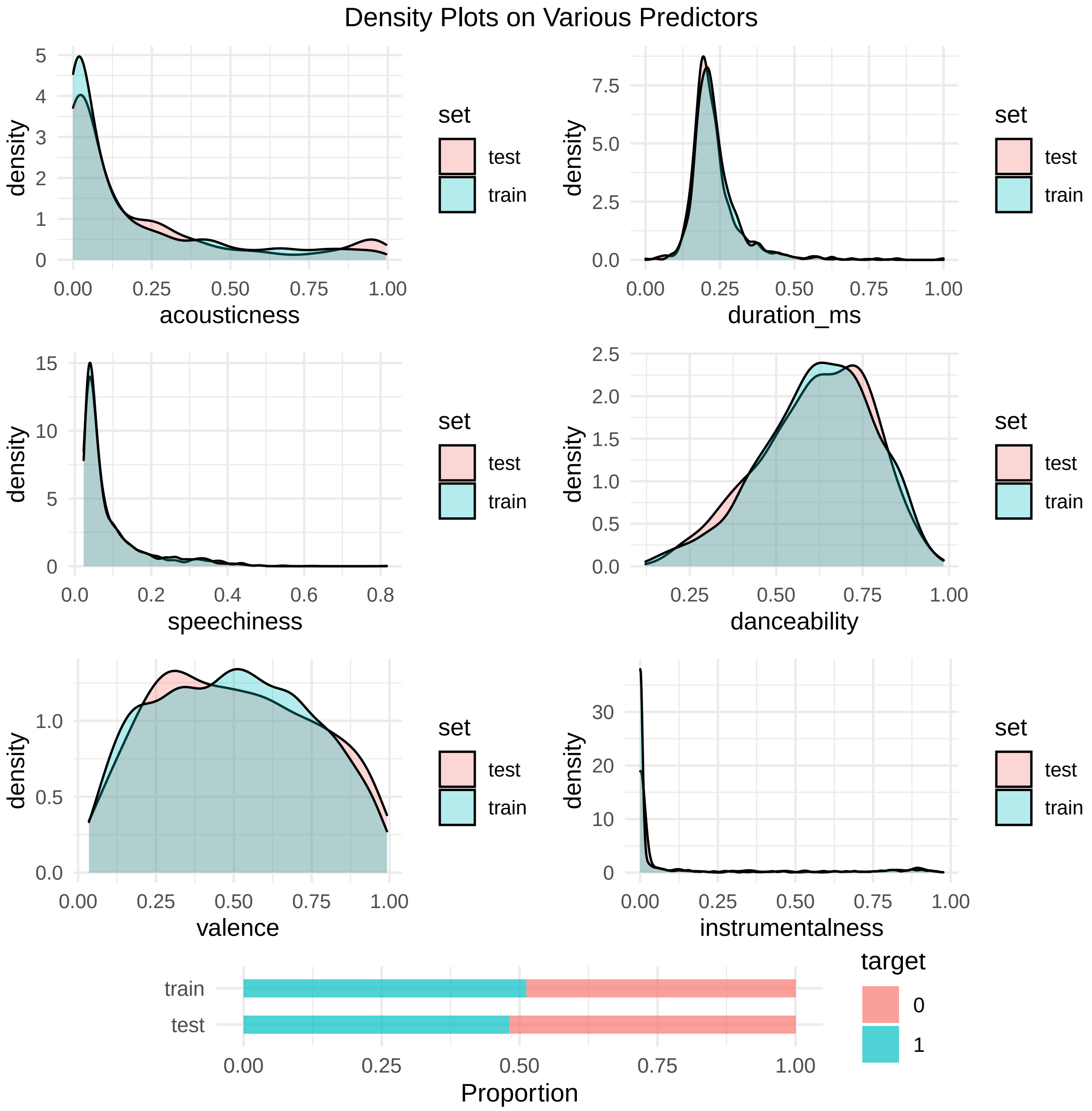
View the covariation plots amid the correlations provided, we determine that the predictors we will keep for modeling are `acousticness`, `duration_ms`, `speechiness`, `dancebility`, `instrumentalness`, and `valence`. Particularly we use the predictors `acousticness`, `speechiness`, `danceability` as a baseline for our simplest fitting model and then add predictors at our discretion.

3 Test/Train split

For training and testing purposes we split the data into 2 different groups. The training data contains approximately 75% of the data and the remaining data is in the test set. We keep these sets consistent throughout all of the modelling approaches we use. Individual model fitting often uses additional hold out sets for validation purposes that are sampled from the training data itself. The validation sets are not held constant from model to model. The dimension follow and we see the proportions predictions is constant.

Data Split Dimensions

	rows	columns
train	1509	7
test	503	7



The distributions and proportions of the test data are representative of those of the training data so we proceed with these splits. One quality of interest in the data is that there are nearly equal ammount of target observations that are 'like' and 'dislike'.

4 Modeling

We consider various models for this classificiation problem.

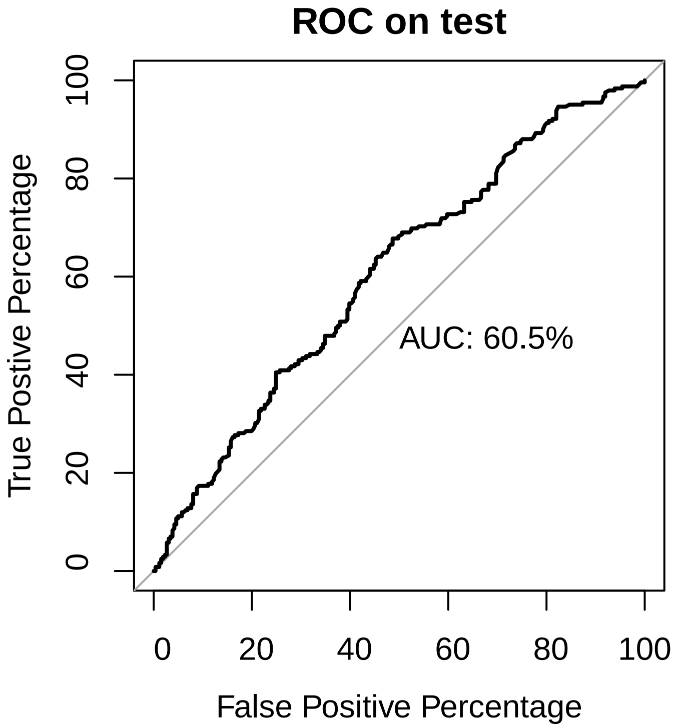
Logistic Regression

One predictor

We use the train and test splits to build a logistic regression model for our problem. We want to start out simple, so we use only one predictor to fit the training sample and predict on the testing sampe. Then we evaluate its performance using based on accuracy and ROC. Note: we will keep the probability threshold as 0.5 to separate the two levels of `target`.

One-predictor logistic regression on test sample

Accuracy	0.5745527



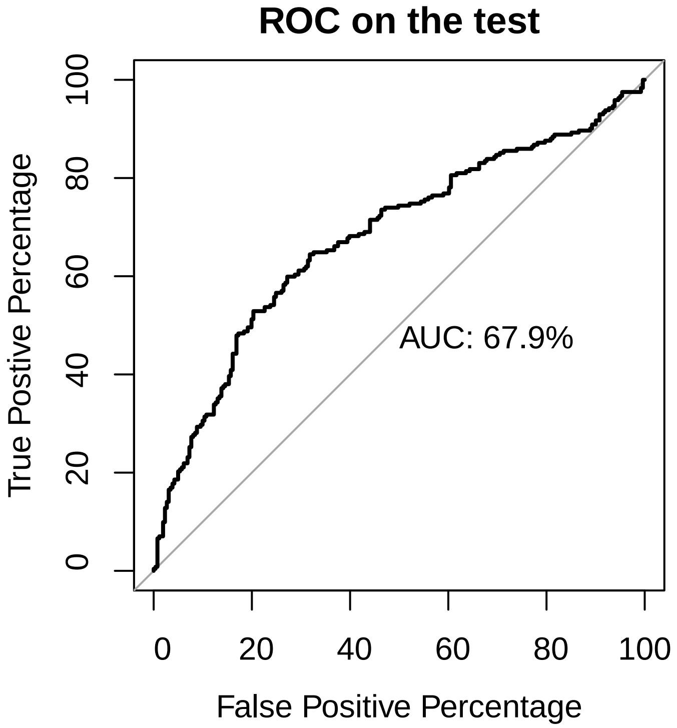
The performance of one-predictor logistic regression is not very ideal, with 57.5% accuracy on and 60.5% of AUC the test. Even though it's not super ideal with such performance, it is still acceptable as we have such a simple model.

Logistic regression with multiple predictors

Next we will try to use more than one predictors to fit logistic regression to compare with only one predictor. Again, we fit on the training sample and predict on the test sample.

Muti-predictor logistic regression on test

Accuracy	0.6540755



It does do a better job than the one-predictor model above, both accuracy and AUC are increased to 65% and 68% respectively on the test set. By experimenting with a simple model like this we can use it as a baseline to compare with more complicated models that follow.

Naive Bayes

We consider 6 predictors based on the correlation value, `danceability`, `speechiness`, `acousticness`, `duration_ms`, `instrumentalness`, `valence`.We develop 7 Naive Bayes classification model using different combination of predictors. We also use three-set approach by splitting the data set into training, validation, and testing data set. We develop the seven classification models using the training data set and then predict using the validation data set. We then compare the accuracy of each model, and select the Naive Bayes model with the highest accuracy.

Naive Bayes Validation Accuracy

accuracyNB1	accuracyNB2	accuracyNB3	accuracyNB4	accuracyNB5	accuracyNB6	accuracyNB7
0.596817	0.6127321	0.6366048	0.5994695	0.6127321	0.6312997	0.6366048

The two highest ranking models on the validation set have an accuracy of 63.67%. The first model is the model that uses all 6 predictors and the second model is the model that uses 3 predictors, `danceability`, `speechiness`, `acousticness`. We refit the model using the combination of training and validation data set on both models and predict using the testing data set. Then, we compare the accuracy of both models.

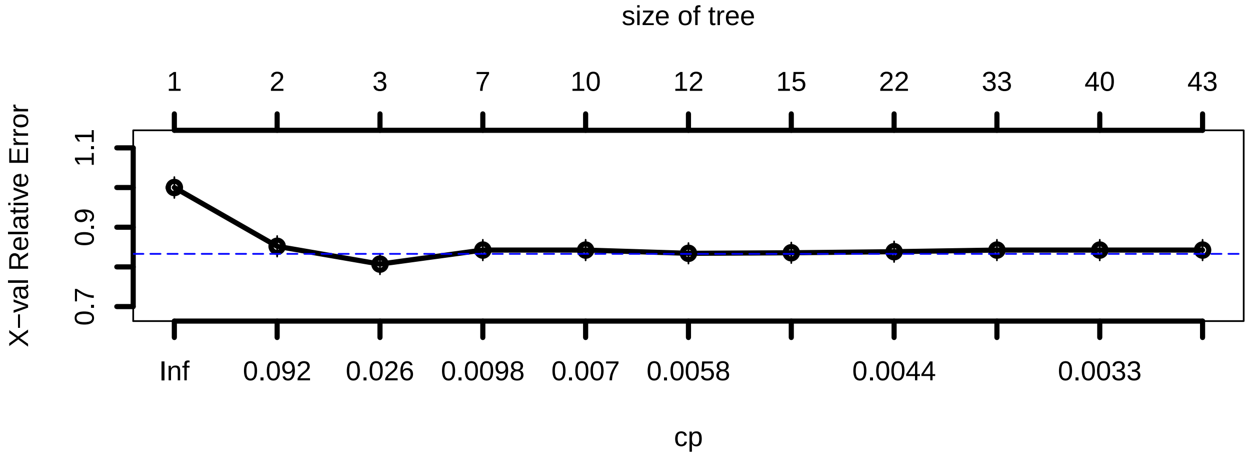
Naive Bayes Test Accuracy

NBaccuracy1	NBaccuracy2
0.6520875	0.5805169

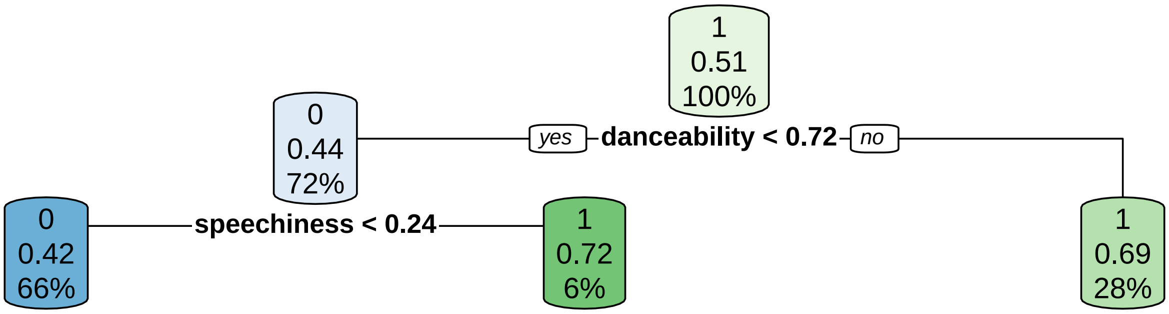
It shows that the highest accuracy is obtained from the Naive Bayes model that uses all the 6 predictors with accuracy of 0.6520875.

Decision Trees

We fit a big tree of size 43 on the training data using using only three features `danceability`, `speachiness`, and `acousticnes`. As suggested by the minimum cross validation relative error, the best size for our fits is 3.



We prune the big tree to a tree of size 3 as shown in the following figure. Although `acounstiness` was included among the predictive variables, but from the plot we observe that it does not play any rule decision making.

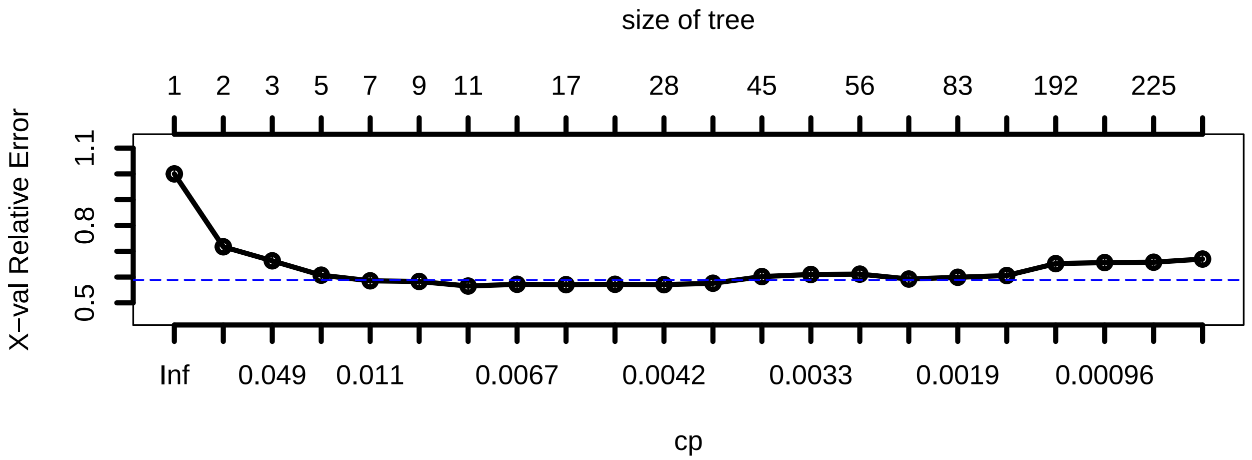


We obtain the accuracy of the fit on the testing data:

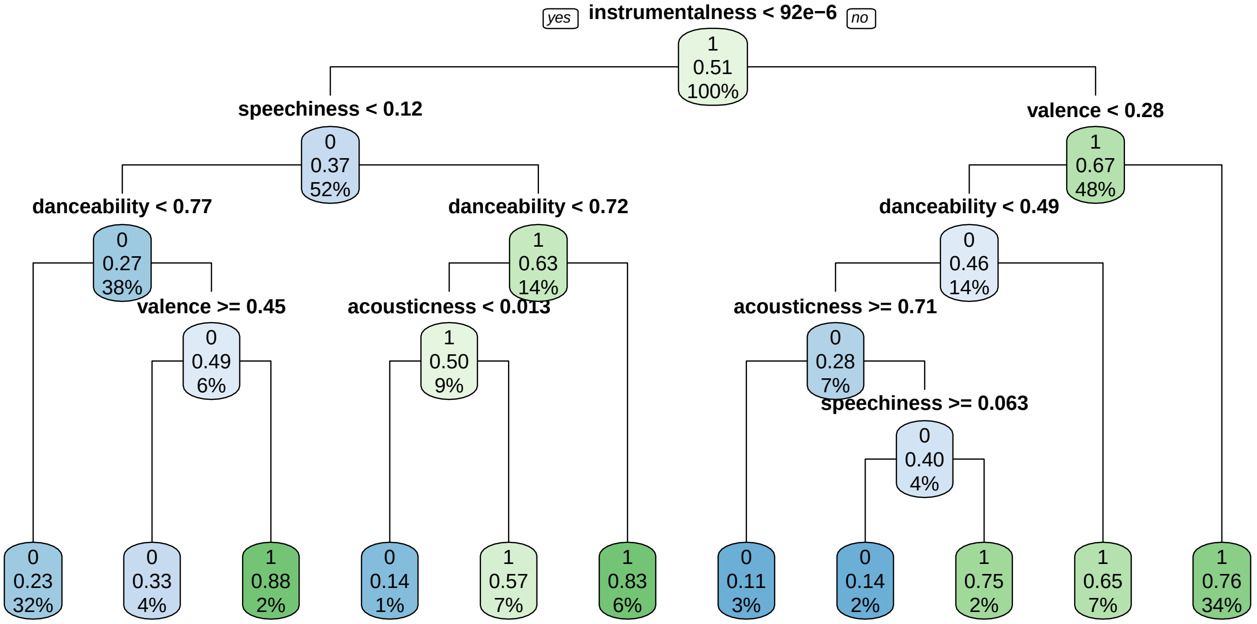
Three Predictor Tree Accuracy

Accuracy	0.5924453

We have now included three more features to the set of predictive variables. These additional features are `duration_ms`, `instrumentalness`, and `valence`. This time we initially fit a big tree of size 259 and then prune it back to the size of best tree which is 11.



The accuracy of this fit on the test data is 0.693837 which is significantly better than our previous fit. Following figure shows the tree of optimal size.



Boosting

Since there are three tuning parameters of the boosting model, we will use 8 different combinations of those parameters to fit the models on the train and chose the model that gives best performance on the validation data to predict for the test. Here are the choices for the three hyper-parameters:

- maximum depth = 4 or 10
- number of trees = 1000 or 5000
- shrinkage = 0.001 or 0.2

Train, Validation and Test Split We split the training data in this method again for a validation set. As before we keep the test set from the beginning but split the train into two subsets: a new train set and a validation set. We fit on the new train subset and predict on the validation set. We choose the probability threshold to be 0.5 to classify probability greater than 0.5 as the class '1', '0' otherwise.

Evaluation of 8 different models We then calculate the accuracy of all models on the validation set. According to the accuracy plot, the best result belongs to the fourth model with maximum depth of 4, 5000 trees, and shrinkage of 0.001.

Boosting Models Accuracy

	ac1	ac2	ac3	ac4	ac5	ac6	ac7	ac8
Accuracy	0.7374005	0.7374005	0.7533156	0.7692308	0.7427056	0.7559682	0.7586207	0.7612732

Now let us use the fourth model to fit the combined train and validation sets then predict on the test. As a result, we get accuracy of 73.56% which is a decent fit on the test set.

Neural Nets

Simple Fit with three predictors

We begin with the simplest model we can fit based on the three main predictors that we have used for the previous models, `danceability`, `speachiness`, `acousticness`. We will use a single-layer neural network to fit on the training data. For optimization we will use a grid search accross value of layer size. We also want to introduce regularization through weight decay so we will include those in our grid as well. The values traversed through are:

Size Grid

5	10	15	20	25	30	35	40	45	50	55	60	65	70	75	80	85	90	95	100

Decay Grid

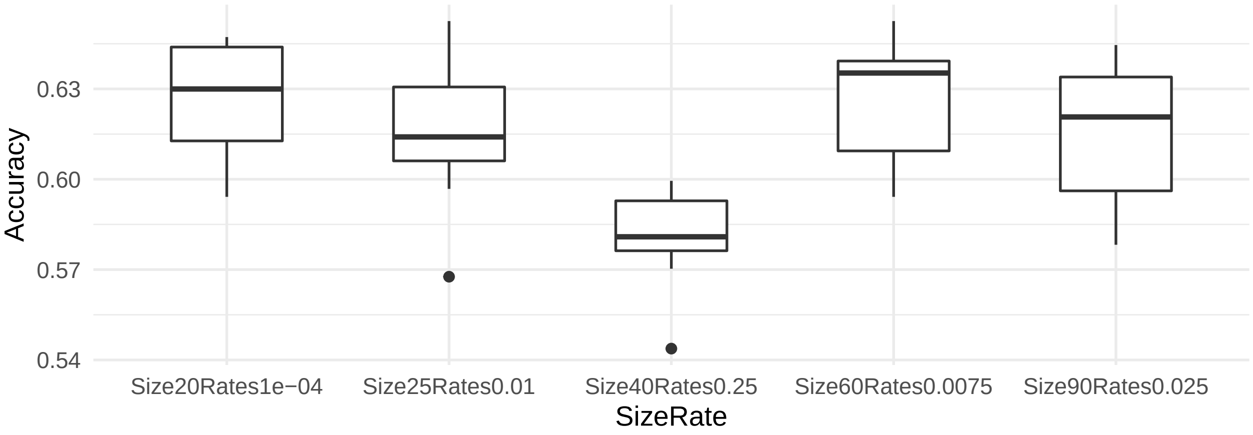
1e-04	0.001	0.0025	0.005	0.0075	0.01	0.025	0.05	0.075	0.1	0.25	0.5

For an initial grid optimization based on the expansion of the grid over sizes/decay rates, we hold out 25% of the data training data for a validation set and compare the methods based on accuracy after fitting on what remains in the training set. This gives us a basis of what parameters will work well, but we will see that these can be unrelaible and very biased to the data. We sort the fits based on their accuracy, and then use the following candidates as parameters from the following fits for comparison: the best fit, the 25th/50th/75th percentiles, and the worst fit. These are:

Various Parameters Considered

size	decay	acc
60	0.0075	0.6074271
25	0.0100	0.5888594
90	0.0250	0.5809019
40	0.2500	0.5729443
20	0.0001	0.5517241

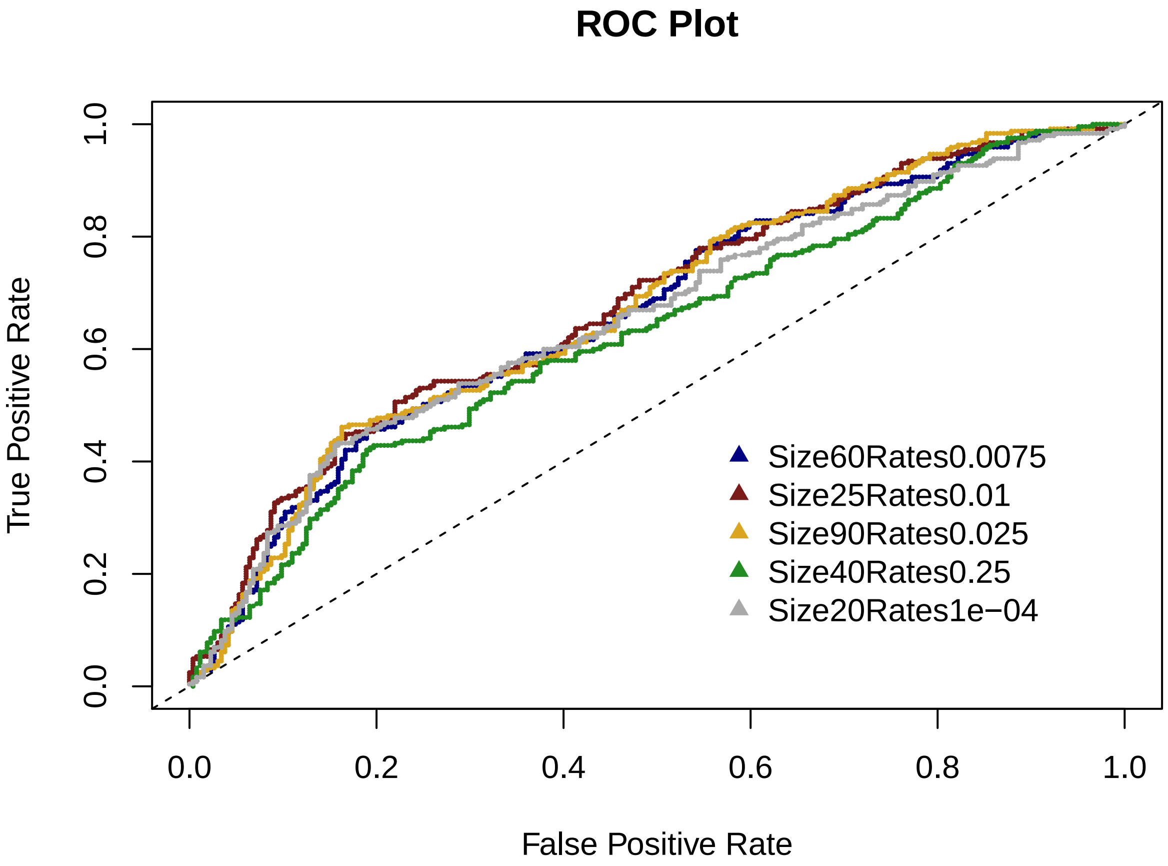
The following box plots show how these parameters fair against eachother over various random splits of the train/validation splits based on accuracy, specificity, and sensitiviy. We iterate serveral times over several splits to obtain these distributions.



Over 10 iterations we have a comparison of these methods, and what we thought would provide parameters for the best fit do not always do well in the prescense of different train and test splits. We compare these methods based on their fit on the test data and their ROC curves.

Test Accuracy

Size60Rates0.0075	Size25Rates0.01	Size90Rates0.025	Size40Rates0.25	Size20Rates1e-04
0.5944334	0.6063618	0.5944334	0.5646123	0.5984095



The best fit on the test data is provided by the suggested parameters that predicted moderately on the validation set rathar than the 'best' parameter's validation fit, but the ROC curves gives us the hint that making the distinction between the two may not be too important since their predictions will be coomparable anyway.

Fitting with all predictors

Now we consider how neural networks compare when we use all of the predictors rather than just the three we originally look at. We fit on various parameters again and compare different candidates based on their accuracy on a train/validation split within the allocated training data. We immediately notice from the validation fits that accuracy has improved in the prescnece of more predictors, which has been the case of all of our examples so far.

Various Parameters Considered on Validation Set

size	decay	acc
55	0.0050	0.7267905
20	0.0075	0.6896552
90	0.0025	0.6737401
60	0.0750	0.6419098
45	0.5000	0.6206897

The following are the results we ended up with for candidates.

Test Accuracy

Size55Rates0.005	Size20Rates0.0075	Size90Rates0.0025	Size60Rates0.075	Size45Rates0.5
0.6858847	0.7176938	0.7037773	0.7236581	0.6719682

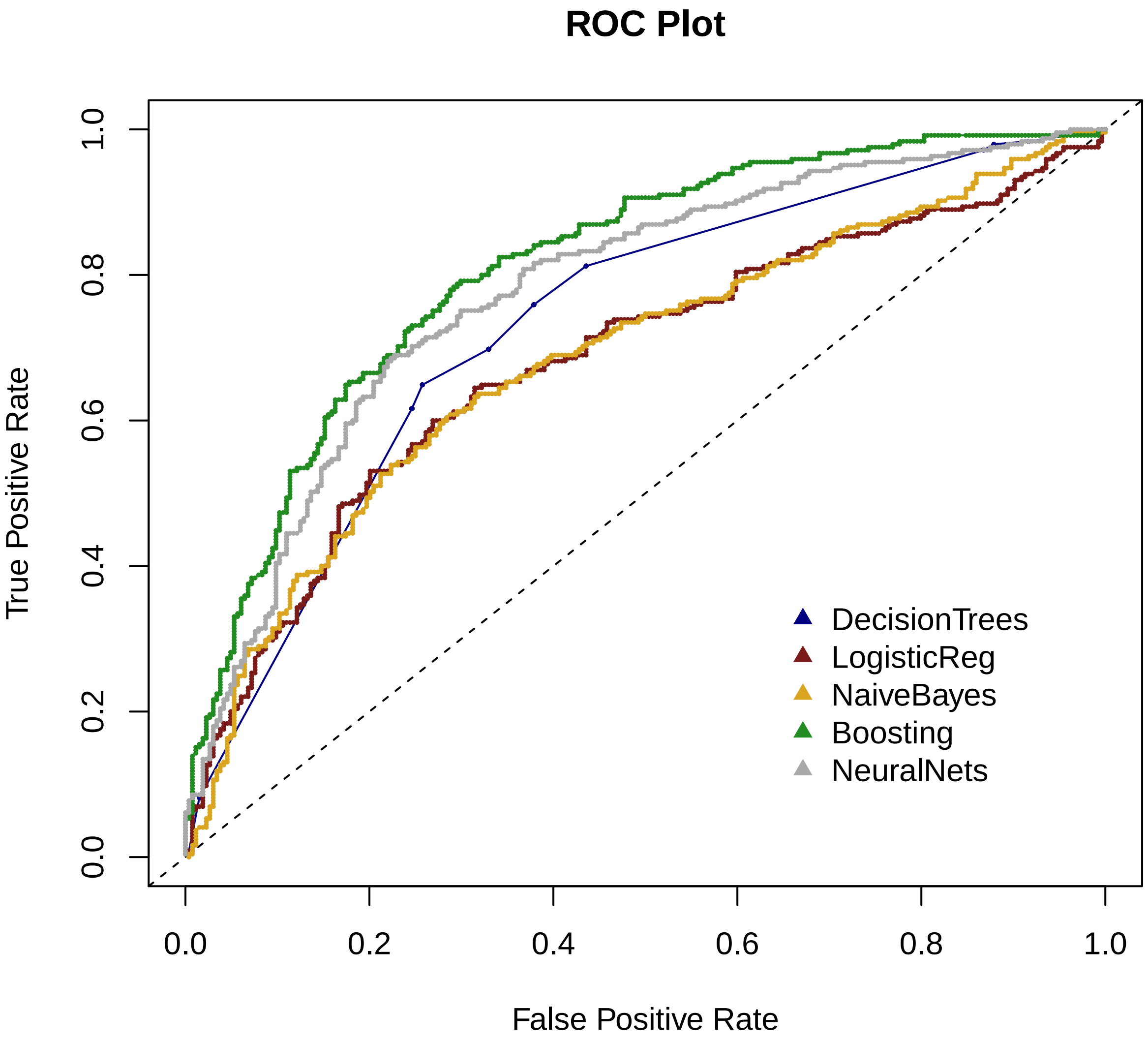
We use the model with the maximum acurracy to compare to rest of the methods in the next section which has parameters:

Best Neural Net Accuracy

size	decay	accuracy
60	0.075	0.7236581

6 Model Comparison

We have used various modeling techniques in several ways, and now we compare the best of those. We have held our testing data constant from model to model so that we use the predicted classifications on that data as comparison. We first plot thier ROC curves.



7 Conclusion