# Machine Learning

## *Summary*

*After reviewing the means and distributions of the features, the decision to merge similar subgenres was done in order to improve overall accuracy of the model.*

*The logic behind the amalgamation of features was largely attributed to indistinguishableness of subgenres with only using the 14 features available for the model.*

*For example, the distinction between 'rock' and 'modern rock' is marginal when looking at features such as the tempo, valence, key etc.*

*However, when a person listens to two subgenres, they become distinguishable based on other subtleties. For example, the two songs below display the difference between 'rock' and 'modern rock'.*

*Rock - https://open.spotify.com/artist/0qEcf3SFlpRcb3lK3f2GZI*

*Modern Rock - https://open.spotify.com/artist/4OTFxPi5CtWyj1NThDe6z5*

*Using the 14 features, the model is able to correctly distinguish between genres 59% of the time (a random selection would be 25%)*

*Genres that contrast quite a bit (ex. rock and rap music) were more accurate whereas similar genres such as EDM and pop were more difficult for the model to select correctly.*

## *Next steps – improvements to the model*

*To improve the model further, additional data is required. Introducing sound data into the model could possibly improve the accuracy.*

*Another idea would be introducing lyric data into the dataset. Lyrics vary widely depending on genre. Training the data on similar words or sentences might provide improvements in model accuracy.*

---

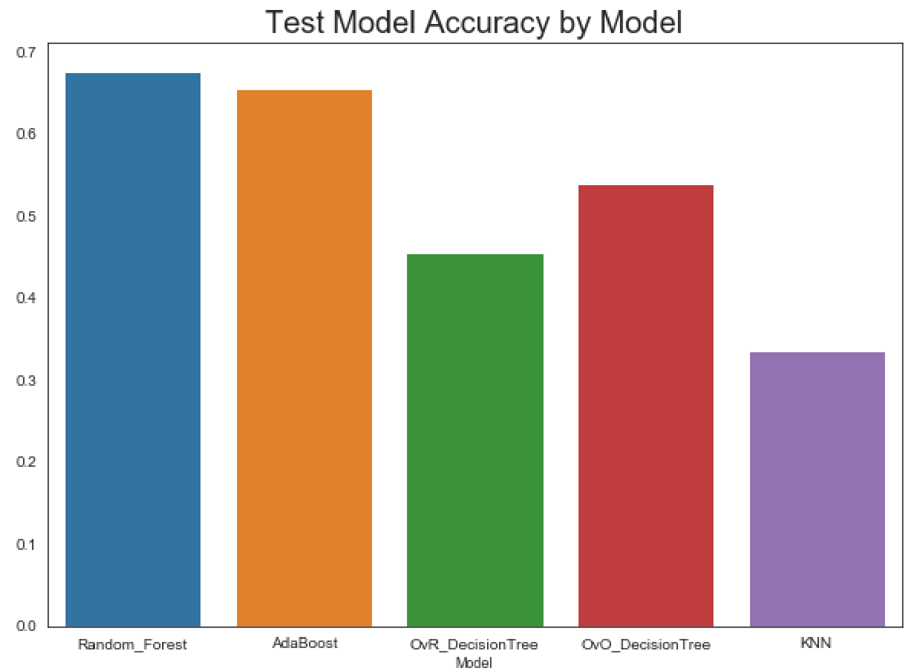
## Model predictions

10 models were used to predict the song genre:

- Random Forest
- AdaBoost
- One Vs Rest (Naïve Bayes, Logistic Regression, Decision Tree)
- One Vs One (Naïve Bayes, Logistic Regression, Decision Tree)
- Support Vector Machine
- KNN

Each model was evaluated using out of the box parameters. 25% of the data was used as holdout to evaluate model performance.

Of the above models, a Random Forest performed the best in terms of test accuracy (67.6%). Based on the results of the out of the box models, it appears that this problem is better suited towards models that can predict non-linear variations in the data. As such, for model optimization, a random forest model was selected.




## Improving Selected Model Performance

To improve the overall performance of the Random Forest model, the following steps were taken:

A. Scaling the data
B. Dropping unnecessary features
C. Grid Search – Hyperparameter tuning and cross validation

**A) Scaling the data –** Using Sklearn standard scaler, the dataset was scaled to a mean of 0 with a standard deviation of 1. Model performance increased marginally (~0.1%) after scaling the data. This marginal increase is expected due to most of the data already being scaled between 0 and 100 by Spotify.

**B) Dropping unnecessary features –** The feature "Mode" was dropped due to the it's feature importance being <~1%

**C) Grid Search –** Hyperparameter tuning was performed by using Sklearn grid search along with 4 fold cross validation and a holdout set of 25%.

The decision to perform both cross validation and use 25% of the training data as a hold out set was done due to the diminishing returns on increasing the training sample. The model converged using roughly ~50% of the training data. As such, it was unnecessary to train on such a large sample.

The following parameters were used in the parameter tuning.

bootstrap: True, False
max depth: 5,10,15,20,25,30
max features: 2,5,6,7,8,9,10,11
n estimators: 10,20,30,40,50,60,70,80,90,100

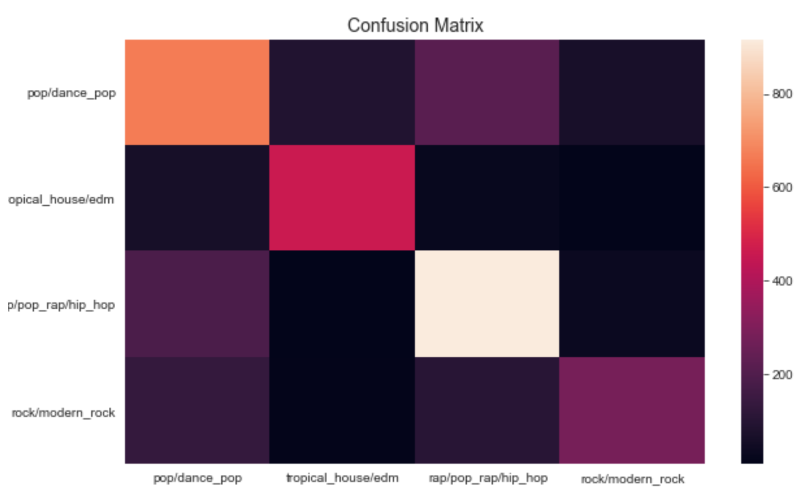
A total of 2,880 fits were performed over the 4 parameters. The optimal model is as follows:

bootstrap: True
max_depth: 25
max_features: 2
n_estimators: 90

The optimal model performed at 70% accuracy vs the accuracy of the out of the box performance of 67%

## Analyzing Model Performance – Further Improvements

Hyperparameter tuning improved the model by 3%. To determine how to improve the model performance further, the confusion matrix for the 'best model' was plotted.




The genre the most difficult to predict is Pop & Hip-Hop. To improve model performance further, adding additional features could help differentiate between pop and rock.

One idea is to improve the model by adding lyric data. Pop & Rock music have very different lyrics. Adding lyrical data in the form of a sparse matrix could potentially improve the accuracy.