Predicting Song Popularity using Machine Learning.

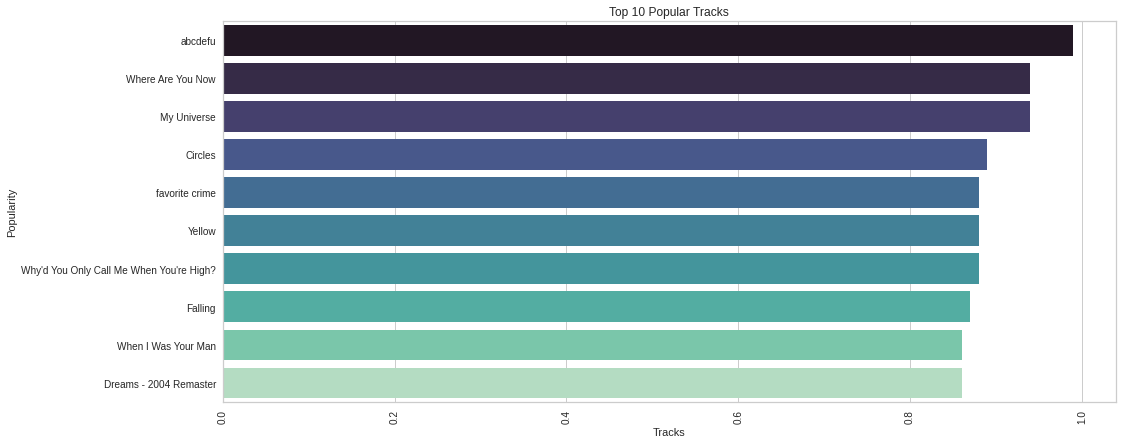
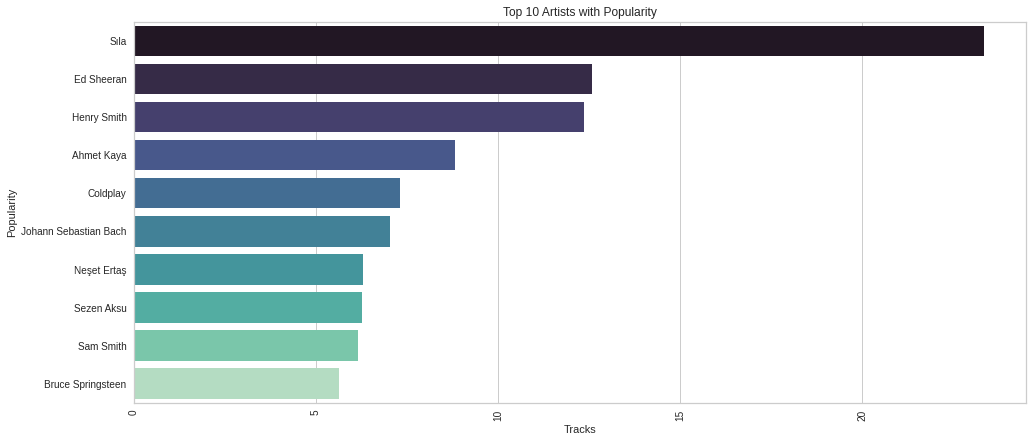
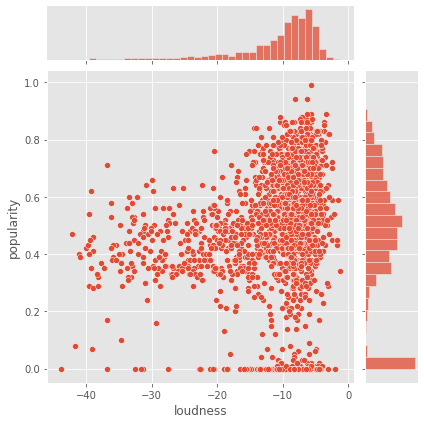
Data mining has gone a long way to prove an essential technique even in the Music Industry. Data regarding songs and playlists from various streaming platforms have been passed through a set of algorithm to give personalized recommendation to their users. Spotify, has not been left behind. Most of the content creators want to understand a listeners preference so that they can be able to satisfy market requirements. In this particular presentation, the question to answer is how song popularity can be predicted based on machine learning techniques. In an objective based manner, data is acquired from Spotify through scrapping, exploratory data analysis done, preprocessing procedures carried out and finally the model building pipeline implemented.

# The Data

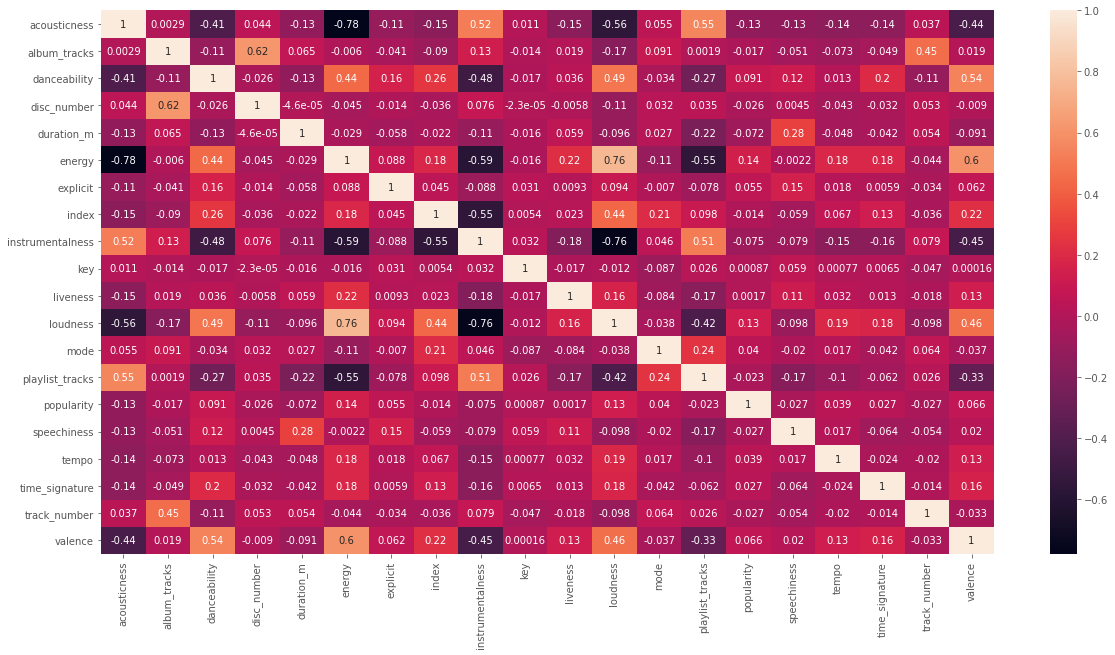
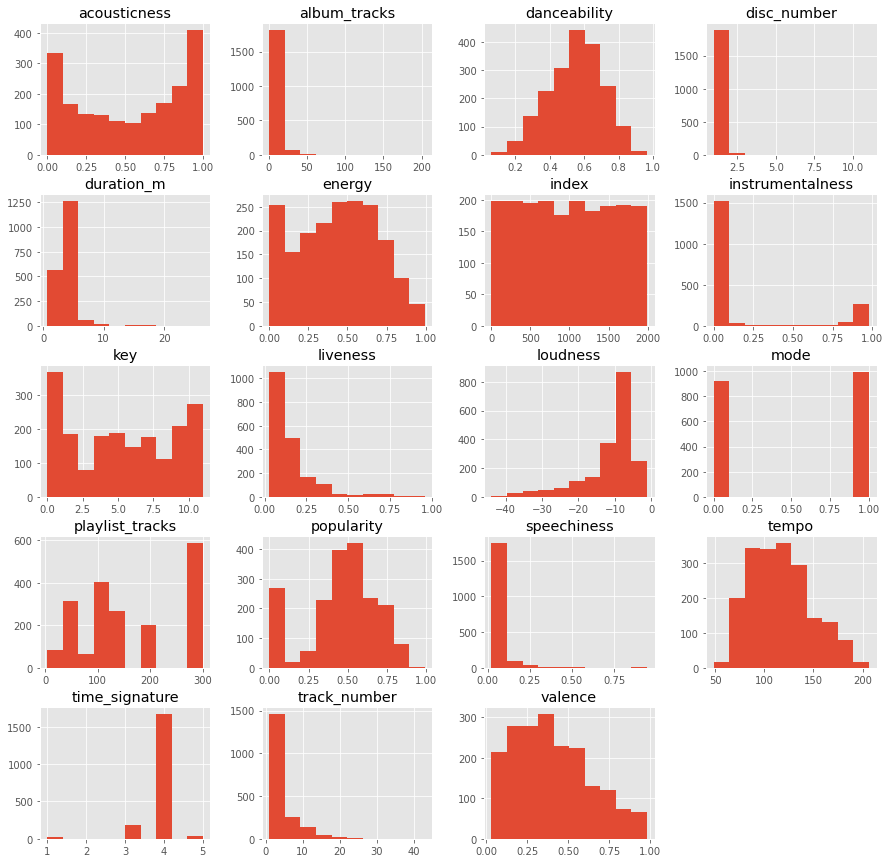
Data was scrapped from my Spotify playlist using a Spotify API from the Spotify’s developer account. The data had 40 variables and 1921 examples. The target variable was song popularity for regression tasks and ratings, which was engineered from popularity for the classification tasks. There was a lot of feature extraction and feature engineering to gather audio features from the data. The data was divided into 3 samples, one containing 500 examples, another 1000 examples and the final one containing 1500 examples. The 500 and 1000 samples was mostly used in the modelling stage due to use of crossvalidation techniques to avoid overfitting of models.

# Exploratory Data Analysis

Bivariate Analysis



Multivariate analysis

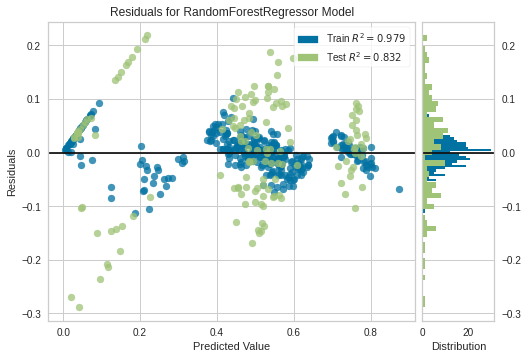


# The Models

The universe of models selected included Random Forests, Support Vector Machines, Decision Tree models and Extreme Gradient Boosting. The first set of Models were used for regressing the feature set on popularity. Clearly, Regression Models did not perform well on the data as compared to classification models. The results for regression models depict that a random forest regression would predict Song popularity perfectly using cross-validation techniques and automatic hyper-parameter tuning.

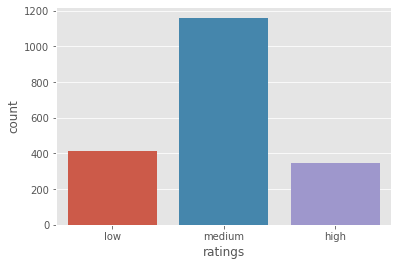


The random forest algorithm had a performance shown below.



| **CV** | **MAE** | **MSE** | **RMSE** | R2 | RMSLE | **MAPE** |
| --- | --- | --- | --- | --- | --- | --- |
| 0 | 0.0599 | 0.0057 | 0.0758 | 0.8263 | 0.0564 | 0.1372 |
| 1 | 0.0696 | 0.0085 | 0.0922 | 0.8194 | 0.0662 | 0.1519 |
| 2 | 0.0859 | 0.0102 | 0.1008 | 0.8385 | 0.0759 | 0.1789 |
| 3 | 0.0720 | 0.0082 | 0.0903 | 0.7466 | 0.0681 | 0.2764 |
| 4 | 0.0701 | 0.0080 | 0.0893 | 0.8709 | 0.0685 | 0.2359 |
| 5 | 0.0721 | 0.0083 | 0.0911 | 0.8291 | 0.0704 | 0.2091 |
| 6 | 0.0516 | 0.0046 | 0.0682 | 0.9125 | 0.0523 | 0.0980 |
| 7 | 0.0699 | 0.0073 | 0.0854 | 0.8710 | 0.0659 | 0.4683 |
| 8 | 0.0572 | 0.0052 | 0.0720 | 0.8675 | 0.0493 | 0.1187 |
| 9 | 0.0762 | 0.0113 | 0.1062 | 0.7357 | 0.0854 | 0.1966 |
| Mean | 0.0685 | 0.0077 | 0.0871 | 0.8317 | 0.0658 | 0.2071 |
| SD | 0.0094 | 0.0020 | 0.0115 | 0.0525 | 0.0103 | 0.1012 |

The task was converted to a classification task by creating three bins for the popularity column. These bins were high, medium and low popularity. There was a problem of class imbalance which was taken care of using SMOTE sampling technique. The target variable ‘ratings’ is visualized as shown.



The algorithm that performed the best was a decision tree classifier with a recall score of 100% as well as the Xgboost Algorithm with 100% Recall score. There could have been a chance of overfitting but this was taken care of by using 10-fold cross-validation.

The results are as shown below.

