BJAM9224 COMP5310 ASSIGNMENT 2

Jamin Jamieson

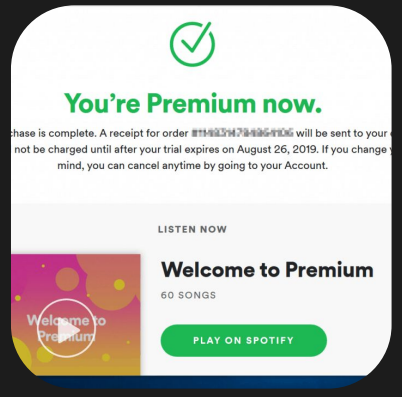
Predicting Popular Music with Machine Learning

Why?

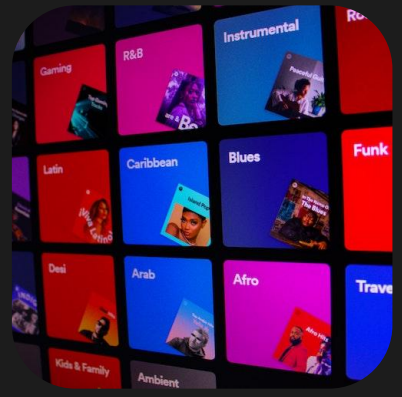
There are several reasons to find good models for predicting songs on Spotify:

Predict next top hit

Promote new artists



Retain customers

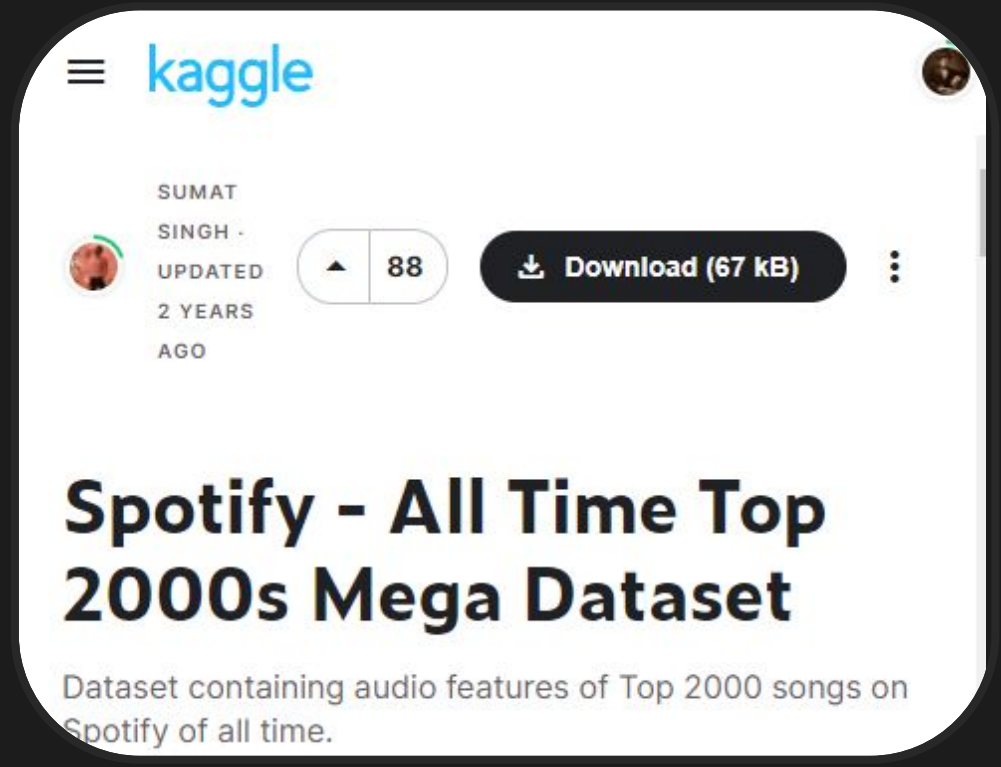


Curate playlists

> Machine Learning models are a suitable method for analyzing song data to predict future trends

The Dataset

Description and Exploration

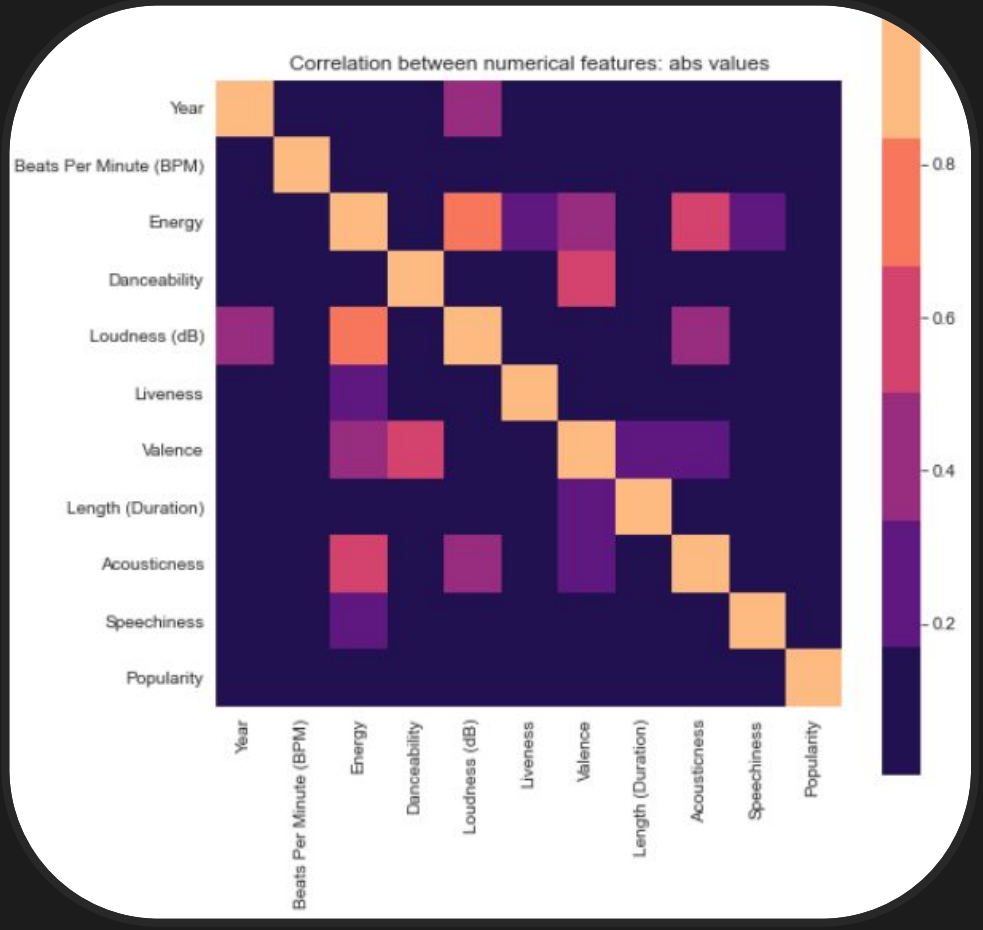


Spotify - All Time Top 2000s

- 1994 unique songs
- Contains 13 other unique features for each song, including indexes that rate danceability, energy and valence between 0-100
- Popularity is a proxy for sustained aggregate listens
- Mean popularity of songs sits around ~65/100

The Dataset

Description and Exploration



Suitability

- Previous studies have found small correlations between features and popularity
- These correlations drop off once popularity reaches higher levels

- Problem: music that is already popular has no obvious determinants from the features
- Solution: use a ML model to predict popularity using all relevant features

Which Model is best?

A benchmark **Logistic Regression** model was compared against three other classification models

H_0: Logistic regression model is *significantly* better at predicting popularity than other models

H_1: Logistic regression model cannot be said to be *significantly* better at predicting popularity than alternative models

Explanation:

- In general, Logistic regression performs best when the number of noise variables is less than or equal to the number of explanatory variables[1].
- As we have very few suspected noise variables, we should be able to generate an accurate model based on the many explanatory variables, although with low correlation coefficients

Pre-processing

Elimination, One-hot encoding, Splitting and Model Selection

1. Elimination

Incomplete features unlikely to add value to the model were eliminated such as genre, song title and year of release

2. One-hot encoding

Artist features were one-hot encoded, generating 793 rows, along with the target vector popularity split into *more popular* (0) and *less popular* (1) using a quantile cut

3. Train-test split

Data was split into train and test at the *train = 0.7* mark with consistent random state

4. Models selected

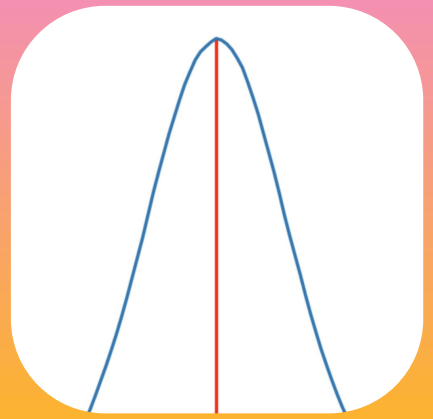
Three obvious classifiers were selected for comparison:

1. Decision Tree Classifier
2. KNN Classifier
3. Random Forest Classifier

Hyper-parameter tuning

Models were tuned to increase F1 accuracy scores

Model	Initial F1 Accuracy	Default Params	Best	Final F1 Accuracy	Error	Retain?
Logistic Regression	0.69	L2, C: 1	L2, C: 1000000	0.7	~0.4	Retain
KNN Classifier	0.63	K = 5	K = 30	0.63	≥0.5	Drop
Decision Tree Classifier	0.62	Depth = 2	Depth = 2	0.62	~0.4	Drop
Random Forest Classifier	0.6	N est = 100	N est = 512 Depth = 64	0.66	~0.4	Retain



Reliability

Chi-squared test of random guessing model

H_0: The model in question is not a random guesser at a 95% confidence interval

Model	*Chi-square statistic*	*P-value*	*C.I*	Conclusion
LogReg	844.5	1.13e-185	95%	Retain H_0
RandomForest	811.7	1.33e-178	95%	Retain H_0

Evaluation

K-fold cross validation on test data using optimized trained models with N = 10 folds

Initial Research Hypothesis:

H_0: Logistic Regression mean F1 score > Random Forest mean F1 score

> Since we are evaluating classification accuracy in both directions, F1 scores are appropriate measure
> Reject H_0 at all confidence intervals with paired t-test

Model	*Mean F1 score*	*P-value*	*C.I*	Conclusion
LogReg	0.678	0.00048	90%	Reject H_0
RandomForest	0.79			

Interpretation

Explanation, Limitations and Improvements

Explanation

- Both models predict the more popular songs quite well, with errors not increasing with complexity
- Random Forest experienced an unexpected increase in accuracy on the test data set

Limitations

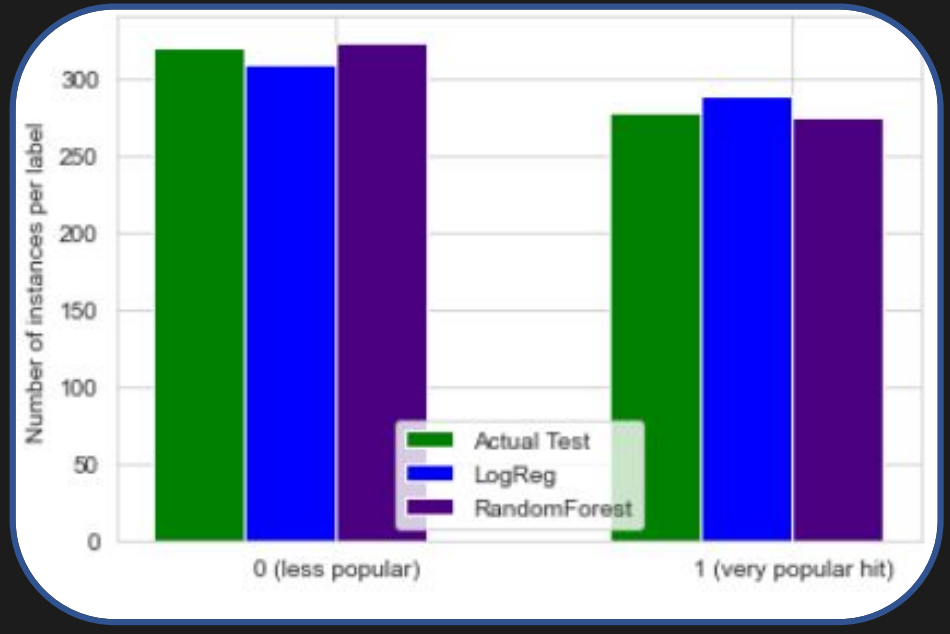
- Random Forest model jump in accuracy (+~0.1) could be due to random error or better fitting on smaller dataset
- Models mostly performed the same, culling models was a bit arbitrary and likely could have been tuned more (I did not understand them well enough to make better adjustments)

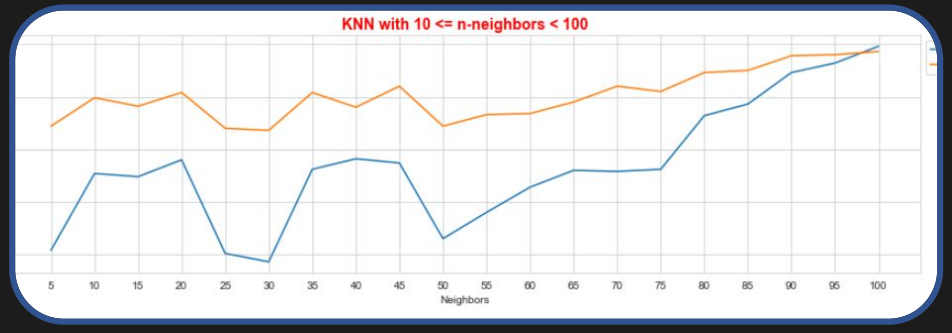
Improvements

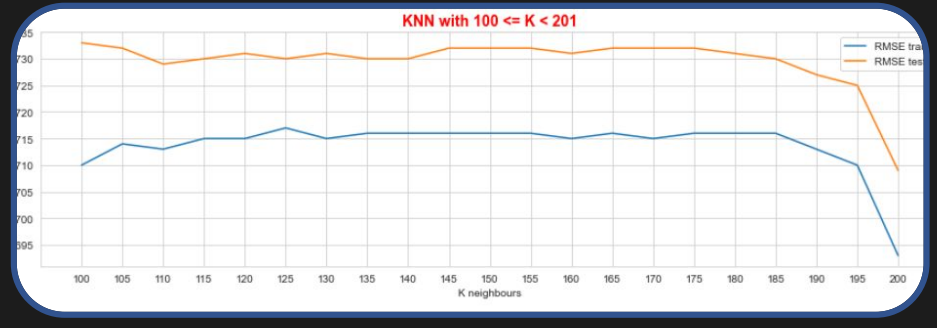
- Consider whether the other models work at different scales of datasets. Logistic Regression may work better at scale than Random Forest classifier. Take note the test dataset is only 599 values
- Test and validate the models on different datasets of similar composition
- Check for additional models that may explain better, such as a Neural Network or Support Vector Machine

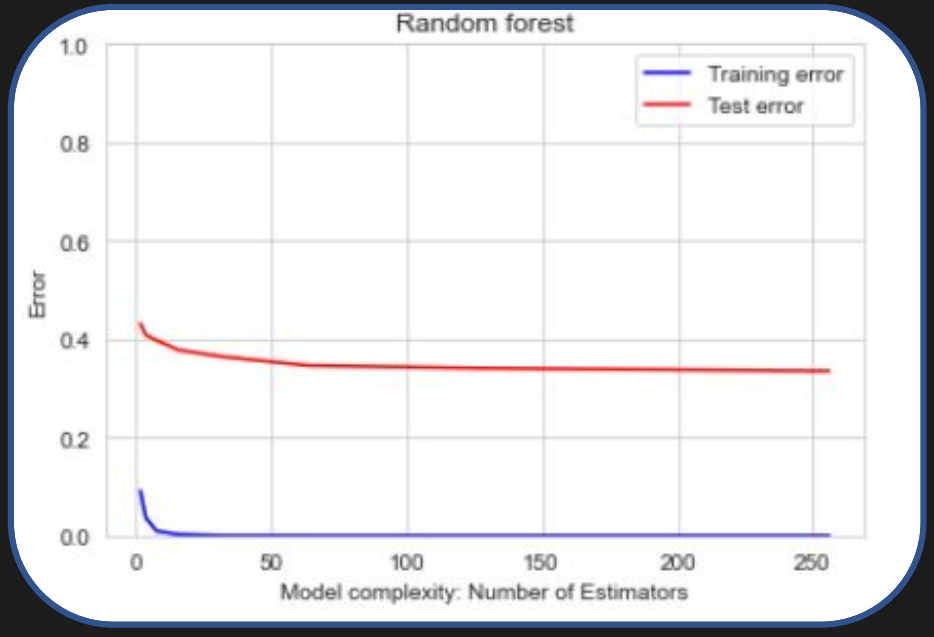
Q&A

Thank you!









References:

1. 'Random Forest vs Logistic Regression: Binary Classification for Heterogeneous Datasets', SMU Data Science Review 2018: https://scholar.smu.edu/cgi/viewcontent.cgi?article=1041&context=datasciencereview