Comparison of Random Forest and Naïve bayes using the Million-song dataset

Yousef Gharib

Description and motivation of the problem

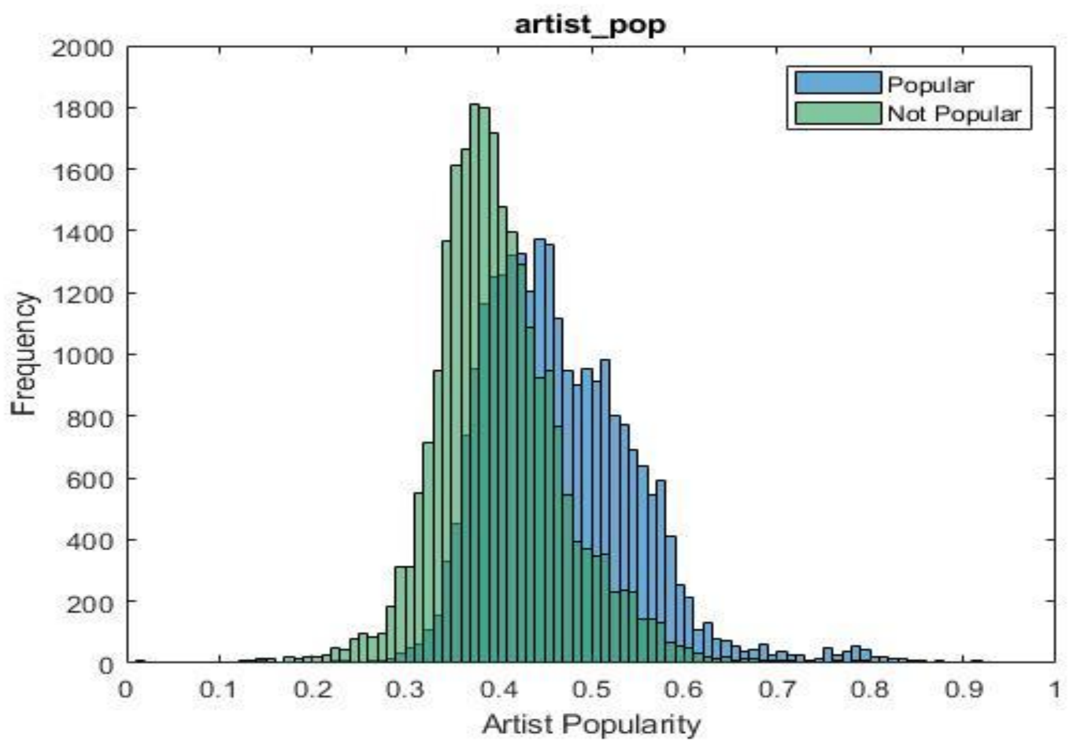
- Analyse and compare the performance of Random forest and Naïve bayes in a binary classification task
- Using the million-song dataset, predict if a song is popular or not

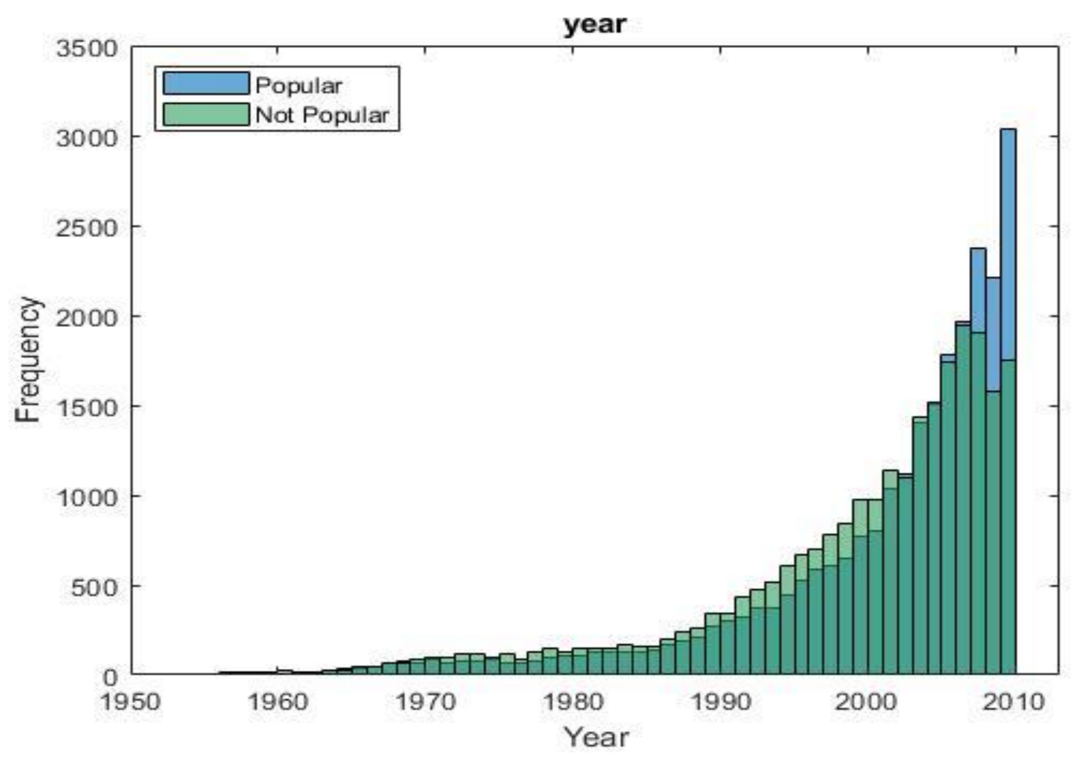
Initial Analysis of dataset including basic statistics

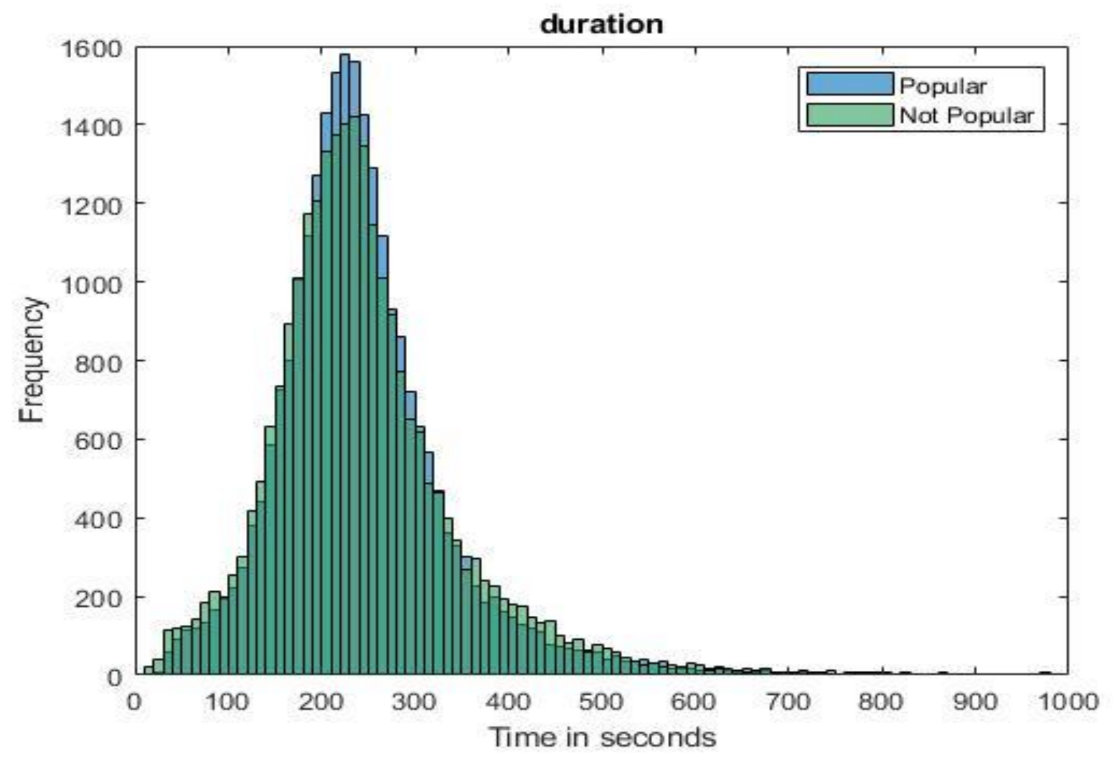
- Dataset: Million song dataset
- Unwanted audio data variables deleted as well as rows with missing information
- Randomly sampled 50,000 songs from remaining dataset
- Left with 24,994 popular songs and 25006 unpopular songs, classes relatively balanced, no bias towards specific class
- Dataset contains 11 variables including response variable
- Calculation of mean and standard deviation of popular and unpopular songs shown in table 1
- Histograms of features indicate most are good representations of a normal distribution
- Insight on feature importance may be determined by comparing histograms based on popularity and audio feature, generally popular songs have higher artist_pop compared to unpopular songs

Table 1: Mean and standard deviation of features based on song popularity

Audio Feature	Mean pop	Mean unpop	Std Pop	Std unpop
artist_pop	0.46775	0.40203	0.085068	0.07124
artist_fam	0.66478	0.57457	0.10411	0.10375
year	2000.6	1998.9	9.4888	9.903
loudness	-8.7117	-9.8717	4.5372	4.9908
tempo	126.38	124.73	35.052	34.572
duration	244.4	246.79	102.82	115.78







Summary of the two ML models with their pros and cons

Random Forest:

- An ensemble method of that constructs independent decision trees which are trained on a bagged subset of the training set
- The randomness of selecting predictor variables results in lower correlation among trees as well as lower error rate[3]
- Each trained model makes a prediction, the average is taken for regression and a majority vote for classification

Pros

- Unlikely to overfit usually producing higher accuracy rates
- Not required to normalize data points
- Robust against outliers and noise[4]
- Works well with missing data

Cons

- Computation time can be very high
- Complex, making it difficult to visualize or understand

Naïve Bayes:

- Assumes predictors are sampled from a distribution
- Uses Bayes theorem to obtain class probabilities
- New data is classified as the class with highest probability

Pros

- Easy to understand and implement
- Scalable
- Surprisingly good results for text classification problems [1]

Cons

- Assumes independence between features which is almost never the case in real life applications.
- Simple approach generally leads lower accuracy rates compared to other models.
- Very prone to overfitting
- Estimations which have zero conditional probability have problems being classified, known as Zero-frequency problem [2]

Hypothesis statement

- Expect Random forest to generate better results compared to Naïve bayes
- Expect accuracy and loss for random forest to converge as number of trees increases
- Since Naïve bayes is simpler to understand and implement we expect the runtime for Naïve bayes to be significantly lower than Random forest

Description of choice of training an evaluation methodology

- Split 75% of dataset for training and 25% for testing
- Carried out 10 runs of grid search, random search, Bayesian optimization on both random forest and naïve bayes
- For random forest, tested Bagging and Adaptive boosting
- Set the maximum objective evaluation to be 10 epochs for all iterations of hyperparameter optimization since difference in estimated objective value and observed objective value does not change significantly and objective runtime is reduced
- Used hyperparameters that resulted in highest accuracy rates

Choice of parameters and experimental results

Random Forest:

Parameters

- Used Bayesian optimization and Bagging as it was the optimization that achieved highest accuracy rates
- Optimizing number of trees, minimum leaf size and number of variables to sample

Experimental results

- Model that had highest accuracy used 480 trees, minimum leaf size of 1 and sampled from all 8 variables
- Accuracy achieved by best model is 0.71936
- Hyperparameter optimization duration was 1743 seconds

Further results

- Bayesian optimization resulted in the highest accuracy as well as highest average accuracy compared to other optimization methods but had the second highest average computation time of 1326.5

Naïve Bayes:

Parameters

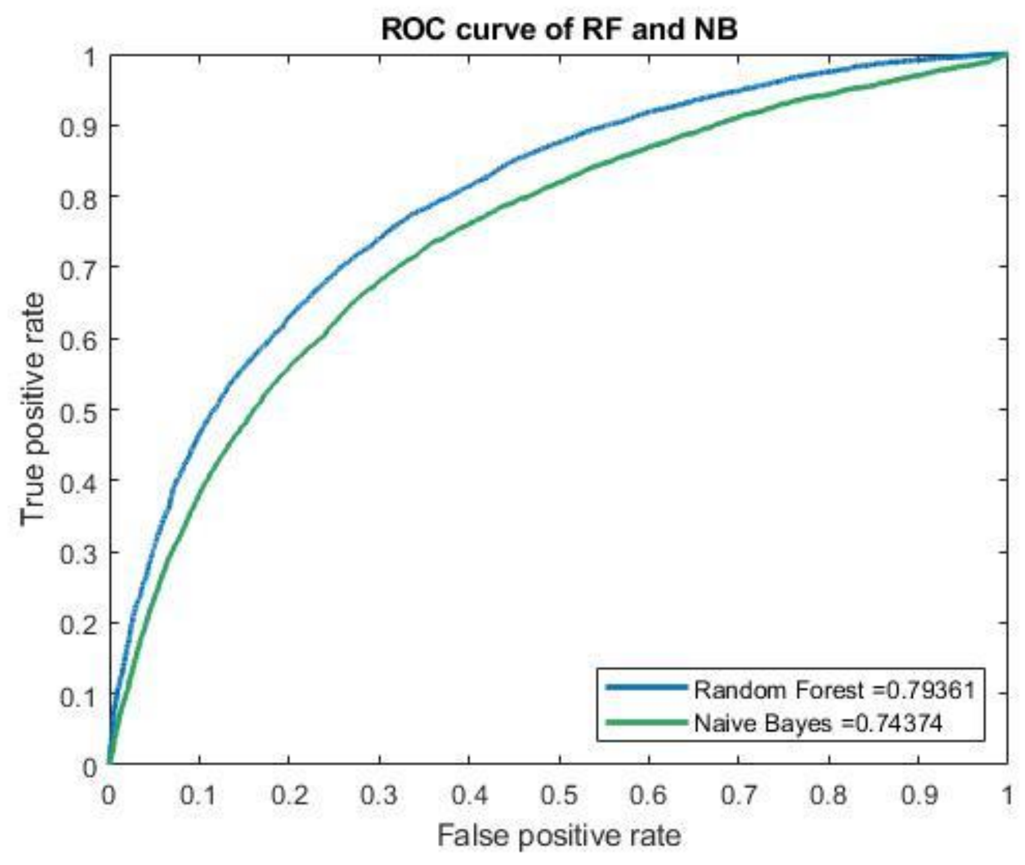
- All optimization methods gave same accuracy rates, for fair comparison Bayesian optimization is used.
- Optimizing distribution, smoothing width and kernel smoother type

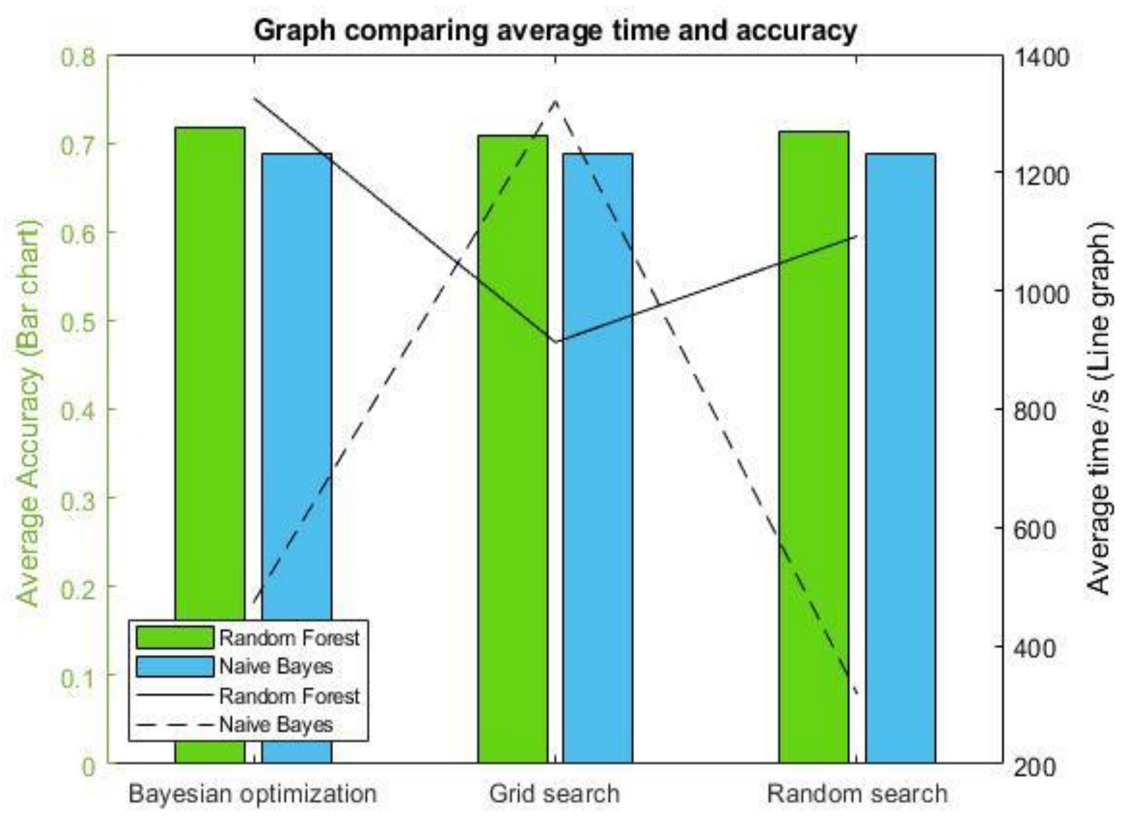
Experimental results

- Optimal model assumes all variables resemble a normal distribution, width and kernel do not apply to normal distributions
- Accuracy achieved by best model is 0.68768
- Hyperparameter optimization duration was 372 seconds

Further results

- All hyperparameter optimization methods resulted in the same accuracy rates
- Bayesian optimization took 150 seconds longer on average than random search to produce the same results while grid search took over 1000 seconds





Model	Accuracy	Precision	Recall	Specificity	F-measure
Random Forest	0.71936	0.70965	0.72582	0.71312	0.71764
Naïve Bayes	0.68768	0.74276	0.67098	0.70862	0.70505

Analysis and critical evaluation of results

- Adaptive boosting took significantly longer than Bagging and resulted in much lower accuracy rates. Adaptive boosting is known to perform better in regression tasks therefore this is expected.
- For Naïve bayes, all optimization methods assumed variables are sampled from a normal distribution resulting in all methods giving the same accuracy rate, the predictor independence assumption results in Naïve bayes having high bias and low variance which is demonstrated by producing lower accuracy rates as well as all optimization methods resulting in the same accuracy.
- Random forest accuracy rate seems to converge to approximately 0.72. Generally, accuracy rate is proportional to the number of trees with 480 trees producing the best rates.
- The area under the ROC curve is above 0.5 for both Naïve bayes and random forest suggesting it is more reliable than a random prediction, random forest having a higher area compared to naïve bayes.
- Naïve bayes surprisingly has a higher precision compared to random forest, suggesting it might perform better than random forest in tasks where the goal is to maximise precision instead of accuracy such as spam detection
- Random forest is more computationally expensive compared to Naïve bayes, all optimization methods took longer to run for random forest except grid search
- The F-measure for Naïve bayes is approximately 0.02 higher than its accuracy rate while for Random forest the F-measure is not significantly lower than the accuracy rate, while Random forest has a higher F1-measure compared to Naïve bayes, Naïve Bayes performs better compared to itself when false positives and false negatives are emphasised whereas Random forests performs worse.

Lessons learned and future work

- Choice of optimization method can result in significant differences in accuracy rates and other measures, it is worth running optimization methods multiple times and comparing them.
- Investigate and compare performance of both Random forest and Naïve bayes on other tasks such as regression and text classification.
- Explore testing with extra hyperparameters, such as depth of tree for Random forest.

[1] EDUCBA. (2019). *Naive Bayes Algorithm | Discover the Naive Bayes Algorithm*
[2] J. Wu, Z. Cai and X. Zhu, "Self-adaptive probability estimation for Naive Bayes classification," The 2013 International Joint Conference on Neural Networks (IJCNN), Dallas, TX, 2013, pp. 1-8, doi: 10.1109/IJCNN.2013.6707028.
[3]Singh, A., N., M. and Lakshmiganthan, R. (2017). Impact of Different Data Types on Classifier Performance of Random Forest, Naïve Bayes, and K-Nearest Neighbors Algorithms. [online] 8(12)
[4]Breiman, L. (1999). Breiman random forests.
[5]Segal, M.R. (2017). Machine Learning Benchmarks and Random Forest Regression.
[6]Breiman, L. (2001)
[7]Strobl, C., Boulesteix, A.-L., Zeileis, A. and Hothorn, T. (2007). Bias in random forest variable importance measures: Illustrations, sources and a solution
[8]Song, Y., Kołcz, A. and Giles, C.L. (2009). Better Naive Bayes classification for high-precision spam detection. 39(11), pp.1003–1024.
[9]Wu, Y., Ke, Y., Chen, Z., Liang, S., Zhao, H. and Hong, H. (2020). Application of alternating decision tree with AdaBoost and bagging ensembles for landslide susceptibility mapping. CATENA, 187, p.104396.