A Comparison of Logistic Regression and Random Forests Through the Classification of Popular Songs on Spotify

Description and Motivation

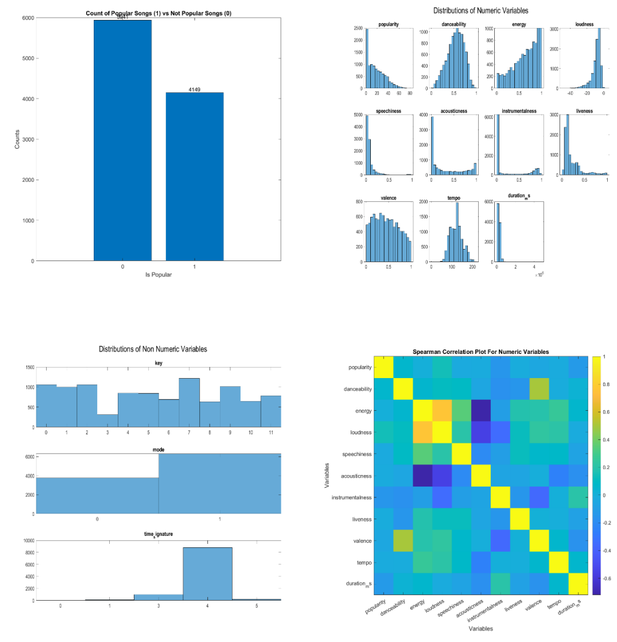
Is it possible to predict if a certain song will be "popular"? Given the metrics that are calculated and songs that are made available by Spotify, Logistic Regression and Random Forests will be used to classify whether a song will become popular. The results from these models will be compared to those achieved by Pareek et al (2022)[1]. While their final dataset consisted of a different sample, the features they used were the same as this project, making it a better comparison than Pham et al (2015)[2].

Initial Analysis of Dataset

- The data set that is being used is the "Spotify 1 Million Tracks" that is available on Kaggle [3].
- 10,090 observations were randomly sampled from the dataset through stratification to maintain the imbalance of the target class.
- The target class is defined from the continuous column "popularity" (0 to 100) where a threshold of 20 or more has been set. This creates a binary classification problem where 1 is the positive class and 0 is the negative class. This information is stored in a new column "is_popular". (see top left figure in matrix on the right). We have an imbalance ratio of approximately 2:3.
- There are 20 columns in total, 13 are being used as the predictors while 1 is the target. The rest have been dropped. There are no missing values.
- The numeric and non numeric distributions (see top right and bottom left in the matrix on the right) are showing greatly skewed data with possibly only 2 variables, tempo and dancebality, following the standard normal distribution. This will affect our choice of scaling for our logistic regression as we will attempt to use a Lasso technique [4].
- The correlation table (see bottom right in the matrix on the right) shows medium strength correlation between only a few variables such as energy and acousticness. This may present a small issue of multicollinearity but the magnitude and presence seems small enough to disregard this for now [5].
- The EDA summary table for numerical variables below shows various summary statistics for the sample such as the mean and standard deviation. What stands out is the prevalence of 0s either as the mode or the minimum value for many variables. Especially in the case for Logistic Regression where the interest lies in the co-efficients and their strength (the change in the log-odds for a one unit change), too many zeros might make the model sensitive to zero. This makes the Lasso technique more important to remove the impact of variables that may not add any value to our predictions.

EDA Summary Table For Numeric Variables

statistic	popularity	danceability	energy	loudness	speechiness	acousticness	instrumentalness	liveness	valence	tempo	duration_ms
mean	18.391	0.539	0.635	-9.051	0.092	0.327	0.255	0.218	0.455	120.940	248995.269
std	15.788	0.183	0.269	5.727	0.126	0.356	0.368	0.196	0.269	29.788	150528.901
mode	0.000	0.565	0.942	-7.362	0.037	0.995	0.000	0.111	0.961	120.010	180000.000
median	15.000	0.553	0.685	-7.540	0.050	0.156	0.002	0.133	0.439	121.126	224585.500
min	0.000	0.000	0.000	-49.457	0.000	0.000	0.000	0.000	0.000	0.000	6966.000
max	81.000	0.986	1.000	3.724	0.964	0.996	1.000	0.998	0.995	216.034	4739947.000



Logistic Regression

- The logistic regression model is part of the family of supervised machine learning algorithms. It has its strengths in binary classification problems.
- Mathematically, the logistic regression uses a sigmoid function, an s shaped curve. Despite the curved nature of the function, it is still a linear model.
- The sigmoid function and, consequently, the log-odds allow us to create linear relationships for the target class and the predictors to predict the chance of the target to fall into either binary class.
- The final class is selected by setting a threshold value, such as 0.5.
- This is different compared to relying on a flow chart structure which is the case for Decision Trees.
- This makes the logistic regression a probabilistic model [10].

Advantages

- It is interpretable due to the equation we create by estimating the coefficients [11]
- Can be extended to non binary problems
- Is fast compared to other algorithms [12]

Disadvantages

- Too many independent features can lead to large standard errors [5]
- If 2 variables are highly correlated and both are included in the model then their effect becomes less precise [5]
- Non linear problems will not work well

Random Forest

- The random forest model is a part of the family of supervised machine learning algorithms. It is capable of both regression and classification tasks.
- It is an "ensemble learning" technique, which is to say that it creates a large number of decision trees and aggregates their answers [6].
- It creates these trees by randomly sampling from the data set (otherwise known as bootstrapping) and at each split of each tree it takes only a subset of the possible feature space. This is what adds the randomness.
- Finally, in the case of classification, it takes the majority class that was predicted across the trees that were created.

Advantages

- It is not prone to overfitting [7]
- It can handle noise [7]
- It is tolerant of highly correlated predictors [8]

Disadvantages

- Mathematical interpretability is lost [8]
- Can bias towards categorical attributes that have a larger count [9]
- Given the large number of trees created, large compute power could be required

Hypothesis Statement

Due to the imbalance in the data, the imbalanced class ie the popular songs, are likely to be misclassified. Papers that used similar variables to the ones in this project did not use Logistic Regression but it is expected to preform worse than Random Forests given the accuracy of 89% Pareek et al (2022)[1] achieved. Although it should be noted that their sample was different and they have not stated a popularity score threshold unlike this project - the "is_popular" variable defined at a popularity score of 20 or more.

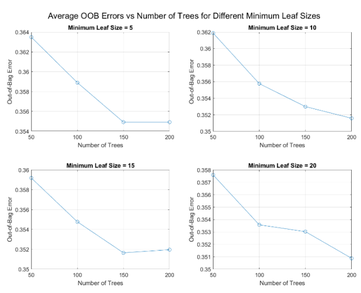
Methodology

1. Data is first randomly sampled from the source [3] through a stratified split to maintain imbalance.
2. An 80:20 split is created for train and test data. The test data is entirely unseen during the training and pre-processing step.
3. We process the data differently for each model to make it ready for prediction. For Logistic Regression we create dummy variables for the categories and robust scale the data due to outliers that were present in the time variable. For Random Forests, due to the nature of the algorithm, little to no processing is done.
4. 2 base models (1 for each algorithm) are trained on the training data with 10-fold stratified cross validation applied. These serve as a comparative base.
5. 2 tuned models (1 for each algorithm) are trained on the training data with 10-fold stratified cross validation and are tuned through grid search to select the optimal values for certain hyperparameters. For Logistic Regression, the lambda and iteration values are chosen for tuning on the basis of the best precision. For Random Forests, the number of trees and minimum leaf size was chosen for tuning on the basis of reducing the Out of Bag (OOB) error [13].
6. The 4 models are then tested on the test data and metrics are compared.

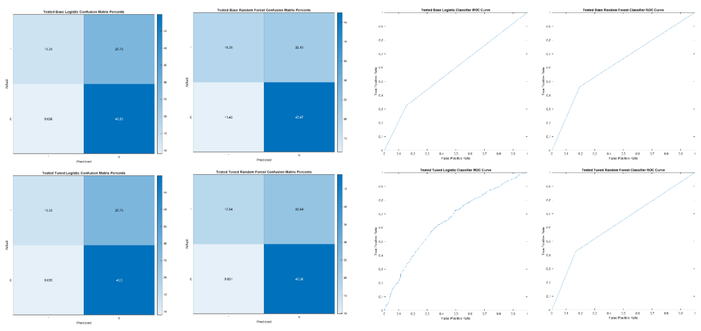
Results

Best Training Mean Metrics

Model	Number of Trees	Minimum Leaf Size	Lambda	Iterations	Precision	Accuracy	Recall	F1	AUC	OOB Error	Fit Time
Base Logistic Regression					0.5700	0.6202	0.3130	0.4038	0.5739		0.0515
Base Random Forest					0.5982	0.6479	0.4395	0.5063	0.6165		2.5560
Tuned Logistic Regression			0.0060	25	0.5866	0.6173	0.2340	0.3342	0.6336		0.0418
Tuned Random Forest	200	20			0.6099	0.6496	0.4117	0.4914	0.6138	0.3509	2.4572



- Logistic Regression (Lasso) was run to obtain a model that would give importance to the variables that truly affected the prediction. From 10 values of Lambda ranging from 0.0001 to 10000 (evenly spaced on the logscale), the best value is 0.006. From 4 values for the number of iterations (25, 50, 100, 150), the best number of iterations is 25.
- Random Forest was fitted by bagging. From 4 values for the number of trees (50, 100, 150, 200), the best number of trees is 200. From 4 values for the minimum leaf size (5,10,15, 20), the best leaf size is 20.



Testing Metrics

Model	Precision	Accuracy	Recall	F1	AUC
Base Logistic Regression	0.5951	0.6318	0.3245	0.4200	0.5853
Base Random Forest	0.6232	0.6640	0.4608	0.5298	0.6333
Tuned Logistic Regression	0.5965	0.6323	0.3245	0.4203	0.6517
Tuned Random Forest	0.6401	0.6660	0.4270	0.5123	0.6298

Lessons Learned

- Focus on imbalance and skewed data for to ensure the models do not become biased
- Hyperparameter tuning is an effective tool to improve model performance however it can become computationally expensive

Future Work

- Different scaling measures could be applied rather than a single scale to see if performance improves
- The threshold value of 0.5 for performance metrics could also be tuned
- Address class imbalance through methods such as SMOTE
- Include lyrical sentiment as new data
- Check for significance between sample and population for robustness

Analysis and Critical Evaluation

- The accuracy achieved in the 4 models was much lower compared to what was achieved by Pareek et al (2022)[1] on their Random Forest. This is likely because they used a different sample. Although it is difficult to pin the difference on just a difference in sample. They have not explicitly stated their classification threshold which could be a possible reason for the increase in accuracy. Observing their results, their K Nearest Neighbour and Linear Support Vector Classifier preformed worse than the models in this project. The performance that was noticed in their Random Forest might also then be due to some specific hyperparameter tuning which is again not stated in the paper.
- Tuning the models for the chosen set of hyperparameters has improved performance slightly on the training metrics (a threshold of 0.5 was set) which likely means that it is possible for better combinations of hyperparameters to exist outside of the defined space. Take for example, the decrease in the OOB error for the Random Forest. Regardless of leaf size, increasing the number of trees reduced the error, so it is likely that there is a higher tree number that improves the model.
- Regarding the OOB estimates, it may have been superfluous to use cross validation along with OOB estimates as the OOB estimates function as a natural cross validation. However, including it helps to compare the models.
- The Logistic Regression was tuned towards finding the best precision but it is interesting to see that even the base Random Forest has a better precision than the tuned regression.
- Regarding the tuning of the Logistic Regression, the optimal number of iterations is chosen to be 25, which is the smallest value defined in the space. This is interesting because previous test models defined a space of 3 values with 50 being the smallest. Increasing the space an adding a smaller value shows that the model is converging faster to find the optimal weights. This could be due to the sample being small or the Lasso technique.
- For the ROC curves, the expected pattern of a jagged curve was not followed for 3 out of 4 models. Initially it felt like the implementation was incorrect but it does not seem to be the case. Assuming the code is correct and given the literature around ROC curves [14] the curves are at a higher angle than a 45 degree line which means there is some predictive power (validated by the other metrics). However the straight lines and kink suggest that for most thresholds the models are predicting the same classes, but for one threshold the models begin to switch their preference. This could be due to the skewed data seen in the initial analysis and the imbalance in the target class. For the Random Forest this sounds plausible as there may not be enough observations for the splits created leading it to prefer the ones where more data is available. Looking at the metrics for the Logistic Regression it seems that after a certain value of Lambda no positive predictions were being made (see supplementary material) suggesting that the threshold of 20 popularity is blurring the lines between popularity. This further suggests that the models are not preforming optimally and there is room for improvement in selecting different hyperparameters and feature engineering.
- The confusion matrices are reflective of the other performance metrics. The higher precision and accuracy values are likely due to the correct prediction of the not popular songs given their abundance in the dataset (bias towards this class).
- The training (seen data) and test (unseen data) metrics are similar suggesting that the models have been able to generalize well (relative to their training) to unseen data.

REFERENCES

1. Prashant Pareek, P. Shankar, M. Pathak, and M. Sakariya, "Predicting Music Popularity Using Machine Learning Algorithm and Music Metrics Available in Spotify Predicting Music Popularity Using Machine Learning Algorithm and Music Metrics Available in Spotify," 2022. Available: https://www.cdes.org.in/wp-content/uploads/2022/01/Predicting-Music-Popularity.pdf
2. J. Pham and E. Kyauk, "Predicting Song Popularity." Available: https://cs229.stanford.edu/proj2015/140_report.pdf
3. "Spotify_1Million_Tracks," *www.kaggle.com*. https://www.kaggle.com/datasets/amitanshjoshi/spotify-1million-tracks/code
4. A. Mathur, S. Moka, and Z. Botev, "Feature Selection in Generalized Linear models via the Lasso: To Scale or Not to Scale?" Accessed: Dec. 16, 2023. [Online]. Available: https://arxiv.org/pdf/2311.11236.pdf
5. P. Ranganathan, C. S. Pramesh, and R. Aggarwal, "Common pitfalls in statistical analysis: Logistic regression," Perspectives in Clinical Research, vol. 8, no. 3, pp. 148–151, 2017, doi: https://doi.org/10.4103/picr.PICR_87_17.
6. R. Couronné, P. Probst, and A.-L. Boulesteix, "Random forest versus logistic regression: a large-scale benchmark experiment," BMC Bioinformatics, vol. 19, no. 1, Jul. 2018, doi: https://doi.org/10.1186/s12859-018-2264-5
7. K. Fawagreh, M. M. Gaber, and E. Elyan, "Random forests: from early developments to recent advancements," Systems Science & Control Engineering, vol. 2, no. 1, pp. 602–609, Oct. 2014, doi: https://doi.org/10.1080/21642583.2014.956265
8. L. Langsetmo et al., "Advantages and Disadvantages of Random Forest Models for Prediction of Hip Fracture Risk Versus Mortality Risk in the Oldest Old," JBMR plus, vol. 7, no. 8, Jul. 2023, doi: https://doi.org/10.1002/jbm4.10757
9. P. T. R, "A Comparative Study on Decision Tree and Random Forest Using R Tool," IJARCCE, vol. 4, no. 1, pp. 196–199, Jan. 2015, doi: https://doi.org/10.17148/ijarcce.2015.4142
10. J. K. Harris, "Primer on binary logistic regression," Family Medicine and Community Health, vol. 9, no. Suppl 1, p. e001290, Dec. 2021, doi: https://doi.org/10.1136/fmch-2021-001290
11. P. Schober and T. R. Vetter, "Logistic Regression in Medical Research," Anesthesia and Analgesia, vol. 132, no. 2, pp. 365–366, Feb. 2021, doi: https://doi.org/10.1213/ANE.0000000000005247
12. A. C. Chang, "Chapter 5 - Machine and Deep Learning," ScienceDirect, Jan. 01, 2020. https://www.sciencedirect.com/science/article/abs/pii/B9780128233375000056?via%3Dihub
13. T. Hastie, R. Tibshirani, and J. Friedman, "Springer Series in Statistics The Elements of Statistical Learning Data Mining, Inference, and Prediction Second Edition." Accessed: Dec. 16, 2023. [Online]. Available: https://hastie.su.domains/Papers/ESLII.pdf#page=611&zoom=auto
14. F. S. Nahm, "Receiver operating characteristic curve: overview and practical use for clinicians," Korean Journal of Anesthesiology, vol. 75, no. 1, pp. 25–36, Feb. 2022, doi: https://doi.org/10.4097/kja.21209