Classical Music Predictor

Repository: https://github.com/carpetxie/math50-final-project

Eddie Bae, Jeffrey Xie, Manraaj Singh, Warren Huang

The hypothesis of this project is that the linear combination of musical parameters significantly predicts the target variable of the composer.

I. Literature Review

Automatically identifying a composer from a musical piece is a central challenge in Music Information Retrieval (MIR). Our project engages with this problem by adopting a "global feature" approach, as supported by Herremans, Martens, and Sörensen (2016). In their study, they were able to classify pieces by Bach, Haydn, and Beethoven through 12 statistical features from symbolic music data, which validates our method of creating a musical "fingerprint" from MIDI segments and modeling the target variable as a linear combination of these features. While other studies have shown success with sequential "local feature" models like n-grams, the global approach is specifically chosen for its interpretability. As the MIR literature emphasizes, "strategies based on hand-crafted mid-level features are still of relevance" precisely because they "allow interpretable and controllable systems" that reveal why a classification was made, a goal often obscured in complex "black box" models (Chowdhury et al., 2022).

Building on this foundation, our feature set includes sophisticated metrics, such as *ioi_entropy* (inter-onset interval entropy), to capture rhythmic complexity by quantifying the uncertainty in the discrete probability distribution of note intervals. This concept is well-supported by studies like Febres & Jaffe (2017), who propose viewing music through its "entropy content" and "symbolic diversity" as a powerful method for "music style recognition," and by research such as Gündüz (2023), which explores how entropy is inherently linked to musical order, complexity, and even perceived instability in melodies. For our classification model, we adopted a one-vs-one (OVO) strategy, decomposing the multi-composer problem into a series of binary classifiers to adapt the standard binary regression framework for a multiclass response. This approach is a standard and highly effective technique for multiclass classification, a conclusion supported by foundational comparative studies in the field (Hsu & Lin, 2002). By grounding our work in established methods (global features, entropy, and OVO classification), our project provides a robust analysis that quantifies the stylistic "fingerprints" of different composers.

II. Dataset

For our dataset, we used drengskauper's Hugging Face MIDI files on various classical music pieces (Drengskapur 2022). For analysis, 10 parameters were used: mean pitch, pitch standard deviation, pitch range, note density, mean inter-onset interval, inter-onset standard deviation, inter-onset entropy, mean note duration, mean velocity, and velocity standard deviation. These parameters are standards in evaluating music but for inter-onset entropy (IOI entropy) ideas of Shannon's entropy were used, which dictates rhythmic complexity in a piece: higher entropy is reflected in more diverse IOI distribution (Gündüz et. al 2023). Then using these parameters, they were used to distinguish 3 composers: Albeniz, Bach, and Alkan. These composers were chosen because they had the most amount of data/pieces available for analysis.

To increase our dataset size, we split each piece into segments of 30 seconds. This roughly quadrupled our dataset to 385 datapoints. With the 10 features mentioned above, our input data has a shape of (385,10). Our labels consisted of tuples with the format of (composer, piece name).

III. Methodology

Our analysis includes four modeling components: (1) a least-squares linear regression used as a binary classifier in a one-vs-one setup, where continuous outputs are converted to class labels using an optimized threshold; (2) a feature ablation study using the same linear model in a one-vs-rest setup to evaluate the contribution of each musical parameter; (3) a logistic regression classifier, included to compare the linear-probability model with a more appropriate method for binary outcomes. This sequence allows us both to meet the course's emphasis on linear modeling and to evaluate composer prediction with a more suitable probabilistic classifier; and finally, (4) classical linear regression analyses required by the course, including multiple-predictor regression, residual diagnostics, ridge regression, and bias–variance evaluation.

1. Least-Squares Regression

Our first experiment was a simple OVO classification between each pair of composers from the set of Albeniz, Bach, and Alkan. This was built via a least squares linear regression model with binary labels for either composer. Features were standardized to zero mean and unit variance before training. The model finds the optimal parameter weightings that minimize the squared error between the linear combination of features and the binary labels. However, since least squares produce continuous output values rather than binary predictions, we need to find a threshold such that if the output value exceeds that threshold, the classification is the second composer, and if it falls below, the classification is the first composer. We select the threshold that maximizes balanced accuracy on the training set, and we evaluate performance using balanced accuracy to account for class imbalance. The results are depicted in Figure 2. Interestingly, OVO classifications including Bach all exceed 0.9 accuracy yet Alkan vs. Albeniz only yields a 0.682 accuracy.

2. Feature Ablation

The second experiment tests ablations on each of the ten features, still utilizing the least squares linear regression model from above. However, instead of an OVO, we run an One versus Rest (OVR) classification for a balanced accuracy assessment on the top three composers (Albeniz, Bach, and Alkan). For each composer, we first establish a baseline accuracy using all features, then remove each feature individually and compare the resulting accuracy against this baseline. Features were standardized to zero mean and unit variance before training, consistent with the first experiment. In Figure 4, we see the results for each of the composers. Interestingly, some features, when removed, would increase the accuracy for some composers. On Figure 3, these are the variables with the most negative values in the ablation test such as inter-onset entropy and mean velocity.

3. Logistic Regression

For our third experiment, we repeated the OVO classification between each pair of composers (Albeniz, Bach, and Alkan) using logistic regression instead of least squares. Features were standardized to zero mean and unit variance before training, consistent with the previous experiments. Unlike least squares, logistic regression models the probability of class membership directly through the logistic function, which naturally

constrains outputs to the range [0,1]. This eliminates the need for threshold optimization, as the decision boundary is fixed at 0.5 probability. The model finds optimal parameter weightings that maximize the likelihood of the observed binary labels. We evaluate performance using balanced accuracy to account for class imbalance. The results are depicted in Figure 10. The accuracy values are nearly identical to those from the least squares approach, with Albeniz vs Bach at 0.943, Bach vs Alkan at 0.977, and Albeniz vs Alkan at 0.688. This similarity is expected, as both methods are linear classifiers that differ primarily in their optimization objectives and output interpretation.

4. Classical Linear Regression with Multiple Predictors and Residuals

In addition to our classifier-based experiments, we also apply the classical linear regression tools required by the course. Unlike the previous sections, the goal here is not to build a composer classifier but to use our dataset to illustrate multiple-predictor regression, residual diagnostics, ridge regression, and the bias–variance tradeoff.

Through the regression_analysis.py script, a multiple-predictor linear regression was fitted using all 10 features at once, where $Y = B0 + B1X1 + \ldots + B10X10 + e$, with Y as a binary composer label and the Xi as the musical features. Compared to single-predictor regressions, the multiple-predictor model controls for correlations among features. For example, pitch standard deviation and pitch range are correlated, and a simple regression can show that part of the change in pitch standard deviation can be explained by the change in pitch range. The multiple-predictor model separates these contributions so each Bi reflects the effect of its feature with all these other parameters controlled for.

Figure 7 shows the residual plot comparing y-hat to (y – y-hat). Because Y is binary, the residuals fall on two diagonal lines: when Y = 1, the residual is 1 – y-hat; when Y = 0, it is –y-hat. These lines come from algebra, not model behavior, and they make the residual plot uninformative for diagnosing issues like curvature or heteroskedasticity. This limitation is one reason logistic regression is usually preferred for binary outcomes.

5. Ridge Regression and Bias-Variance

Beyond the multiple-predictor fit, we also study model complexity using ridge regression and the bias–variance tradeoff to understand how regularization affects performance. Figure 8 shows the ridge regression analysis. The left plot shows train and test R^2 as a function of the regularization parameter λ. The optimal λ is around 100, where test R^2 is highest. As expected, train R^2 is slightly higher than test R^2; reversing this pattern would indicate over-regularization or noise. The right plot shows coefficient shrinkage: as λ increases, the coefficients move toward zero, producing a simpler model.

Figure 9 illustrates the bias–variance tradeoff. As model complexity increases, training error decreases (lower bias), while test error eventually increases for a number of parameters exceeding our current 10 (higher variance). Since MSE = Variance + $Bias^2$, the test MSE begins forming the expected U-shaped curve. Even though Y is binary-coded, these diagnostics remain valid because they evaluate the continuous fitted predictions y-hat, which are later thresholded for classification.

IV. Closing Thoughts

While linear regression was used in several forms to connect with class concepts (multiple predictors, residuals, ridge regression, and bias–variance), we also implemented logistic regression as the more appropriate model for binary composer prediction. This gives us a clean comparison: the linear probability model is helpful pedagogically, but logistic regression is better aligned with the underlying statistics of a 0/1 outcome. The similar performance between the two approaches suggests that the main structure in our feature space is largely linearly separable, and that simple linear classifiers capture most of the signal in distinguishing these composers. This project proves our hypothesis that it is possible to use regression to classify classical music composers from different components of their music. However, many open questions remain for discovery. For example, this study used global, hand-crafted musical features found in literature like entropy or variance but a question that could be explored is: What efficacy does different types of musical analyses have on identifying composers? Also, this study used 30 second intervals but we could explore further how segment length affects the results. Finally, expanding from the mathematical models of the class, how does polynomial regression (or other types of non linear regression) increase or decrease the accuracy of the prediction model?

V. Figures

Figure 1:

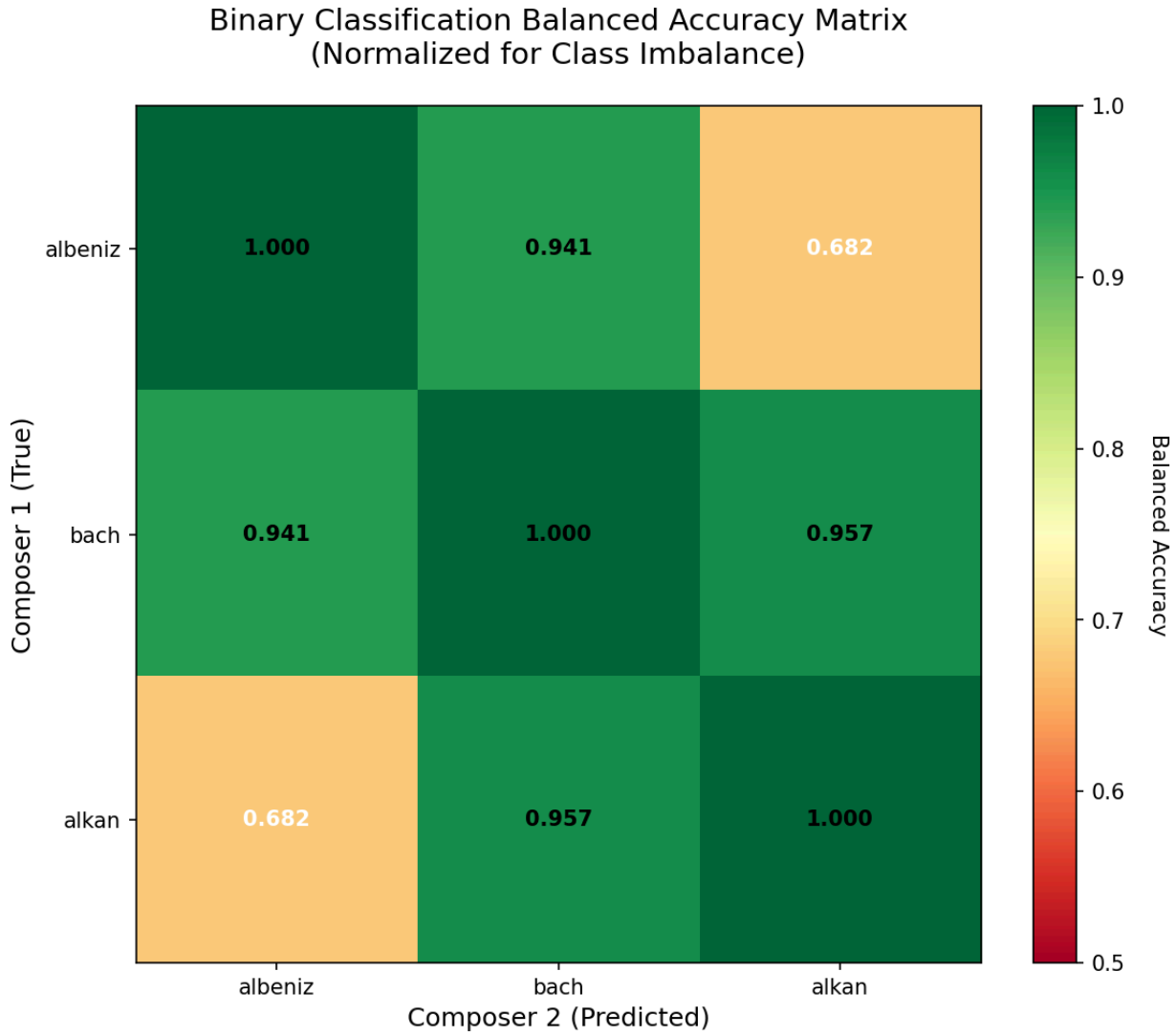

Figure 2:

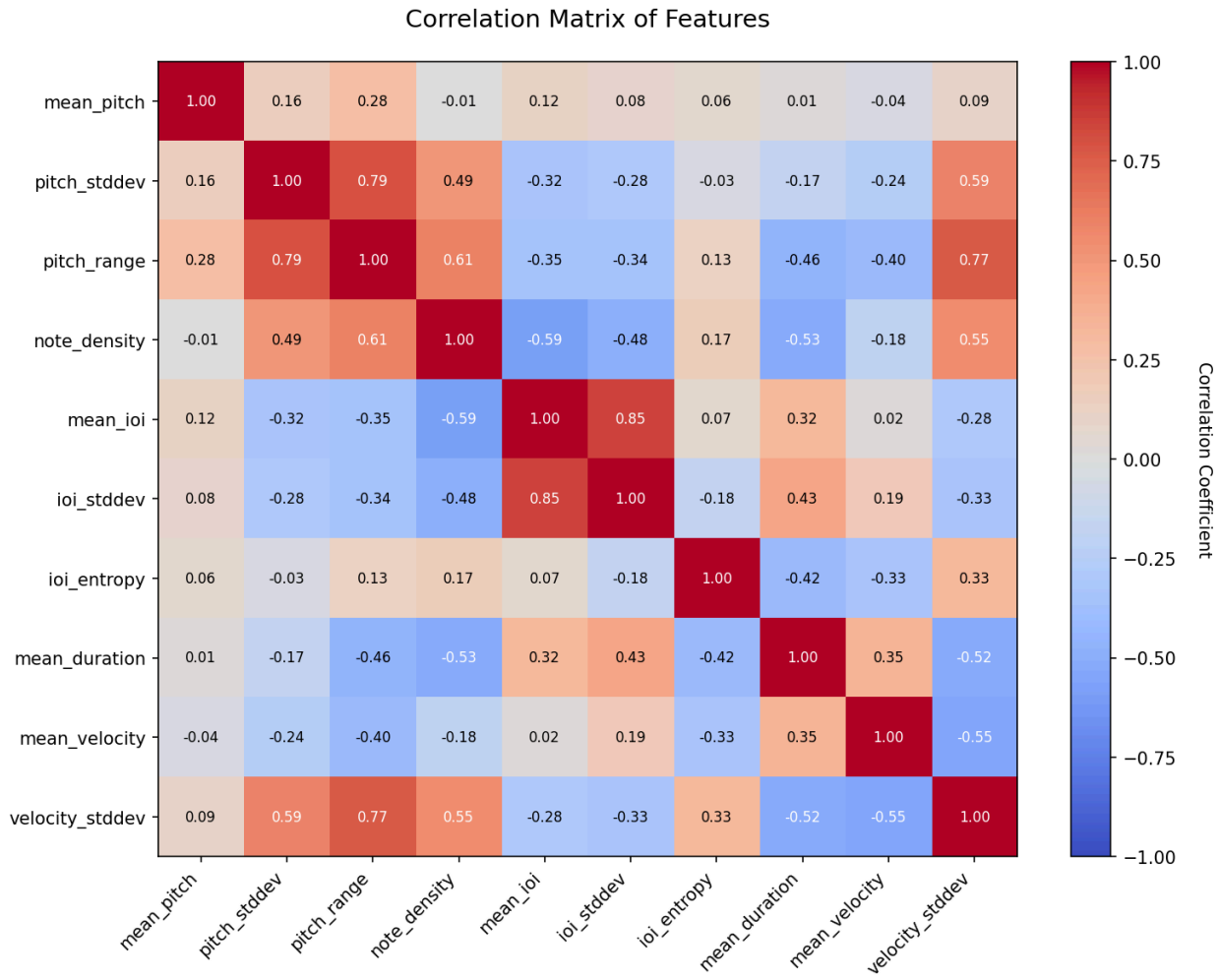
Correlation Matrix of Features
mean_pitch 1.00 0.16 0.28 -0.01 0.12 0.08 0.06 0.01 -0.04 0.09
pitch_stddev 0.16 1.00 0.79 0.49 -0.32 -0.28 -0.03 -0.17 -0.24 0.59
pitch_range 0.28 0.79 1.00 0.61 -0.35 -0.34 0.13 -0.46 -0.40 0.77
note_density -0.01 0.49 0.61 1.00 -0.59 -0.48 0.17 -0.53 -0.18 0.55
mean_ioi 0.12 -0.32 -0.35 -0.59 1.00 0.85 0.07 0.32 0.02 -0.28
ioi_stddev 0.08 -0.28 -0.34 -0.48 0.85 1.00 -0.18 0.43 0.19 -0.33
ioi_entropy 0.06 -0.03 0.13 0.17 0.07 -0.18 1.00 -0.42 -0.33 0.33
mean_duration 0.01 -0.17 -0.46 -0.53 0.32 0.43 -0.42 1.00 0.35 -0.52
mean_velocity -0.04 -0.24 -0.40 -0.18 0.02 0.19 -0.33 0.35 1.00 -0.55
velocity_stddev 0.09 0.59 0.77 0.55 -0.28 -0.33 0.33 -0.52 -0.55 1.00
mean_pitch
pitch_stddev
pitch_range
note_density
mean_ioi
ioi_stddev
ioi_entropy
mean_duration
mean_velocity
velocity_stddev
1.00
0.75
0.50
0.25
0.00
−0.25
−0.50
−0.75
−1.00
Correlation Coefficient

Figure 3:

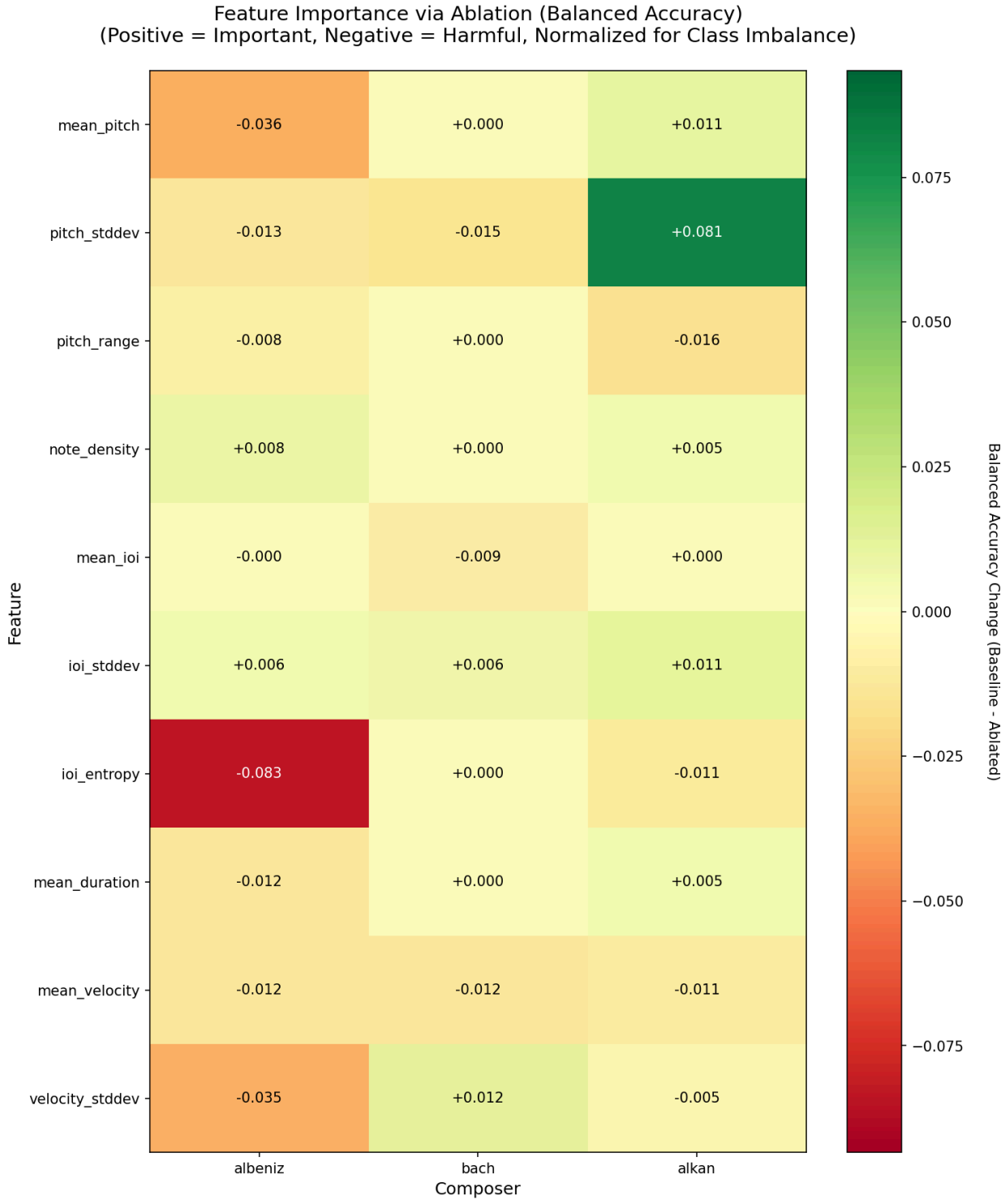
Feature Importance via Ablation (Balanced Accuracy)
(Positive = Important, Negative = Harmful, Normalized for Class Imbalance)
mean_pitch
pitch_stddev
pitch_range
note_density
mean_ioi
ioi_stddev
ioi_entropy
mean_duration
mean_velocity
velocity_stddev
-0.036
+0.000
+0.011
-0.013
-0.015
+0.081
-0.008
+0.000
-0.016
+0.008
+0.000
+0.005
-0.000
-0.009
+0.000
+0.006
+0.006
+0.011
-0.083
+0.000
-0.011
-0.012
+0.000
+0.005
-0.012
-0.012
-0.011
-0.035
+0.012
-0.005
albeniz
bach
alkan
Composer
Feature
0.075
0.050
0.025
0.000
−0.025
−0.050
−0.075
Balanced Accuracy Change (Baseline - Ablated)

Figure 4:

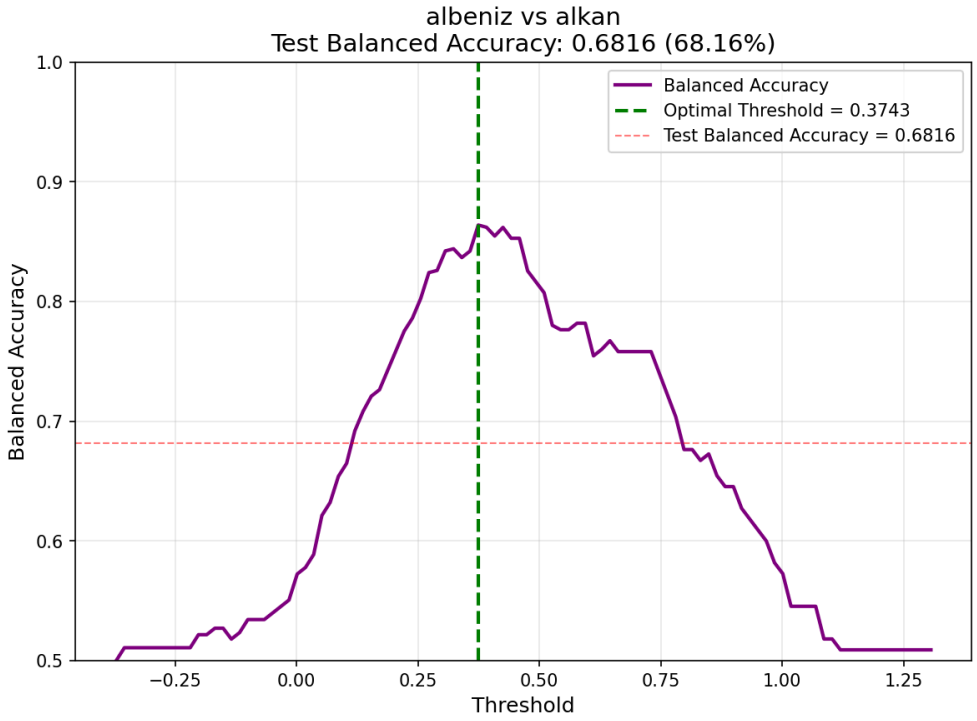

Figure 5:

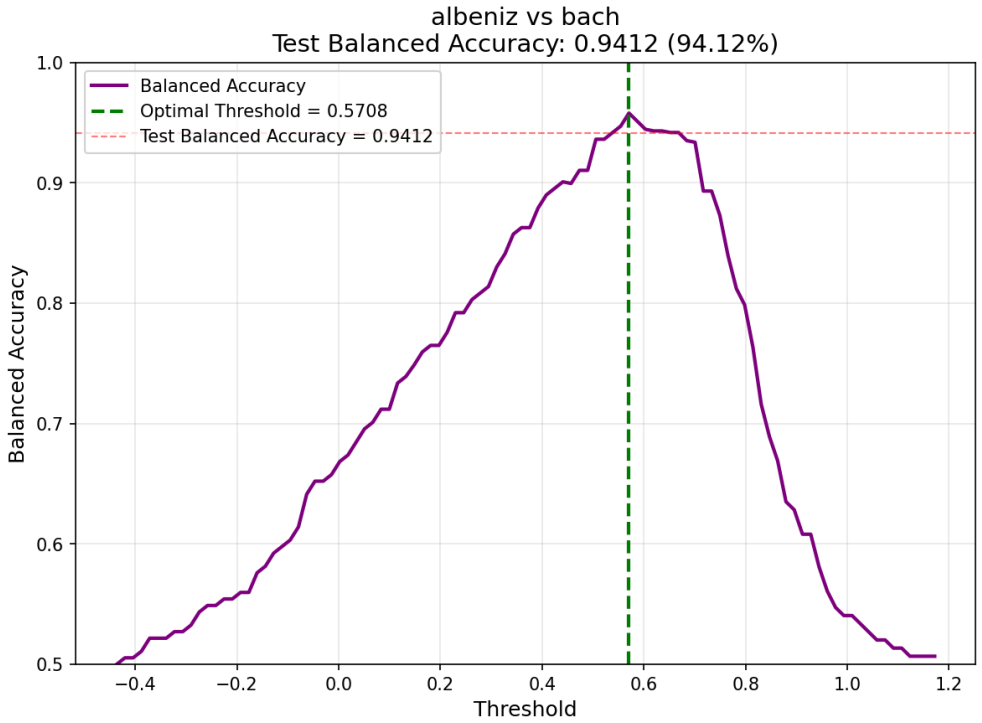

Figure 6:

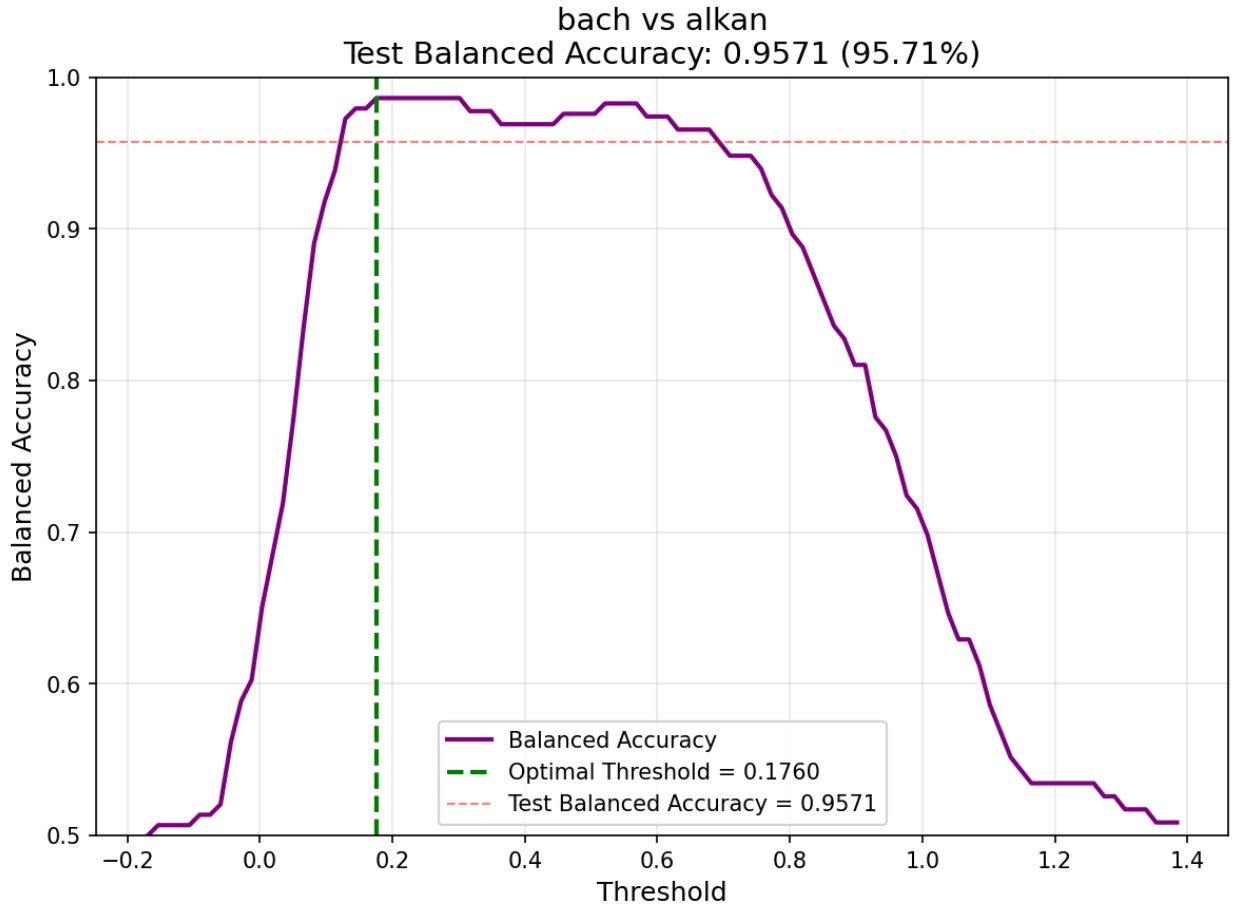

Figure 7:

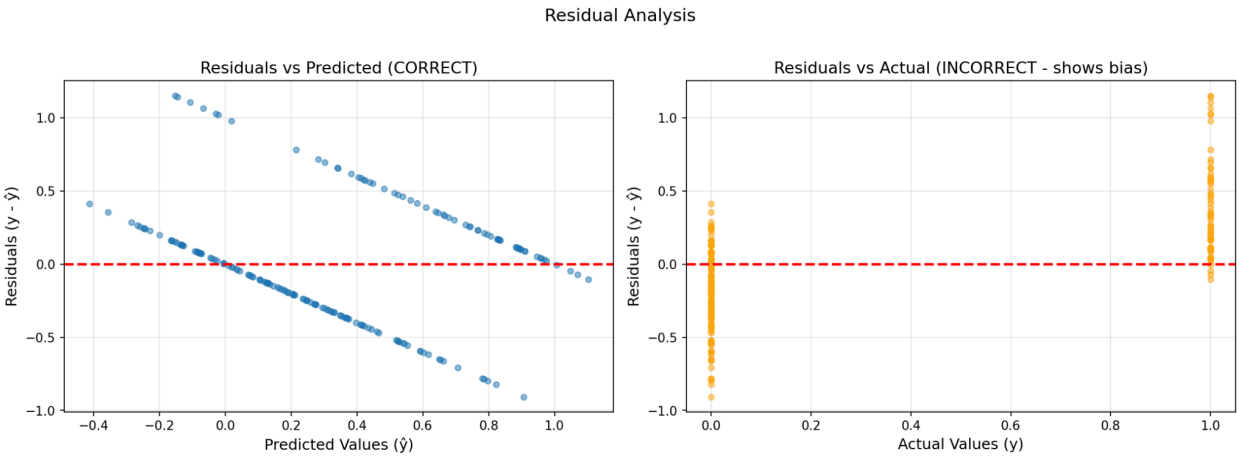

Figure 8:

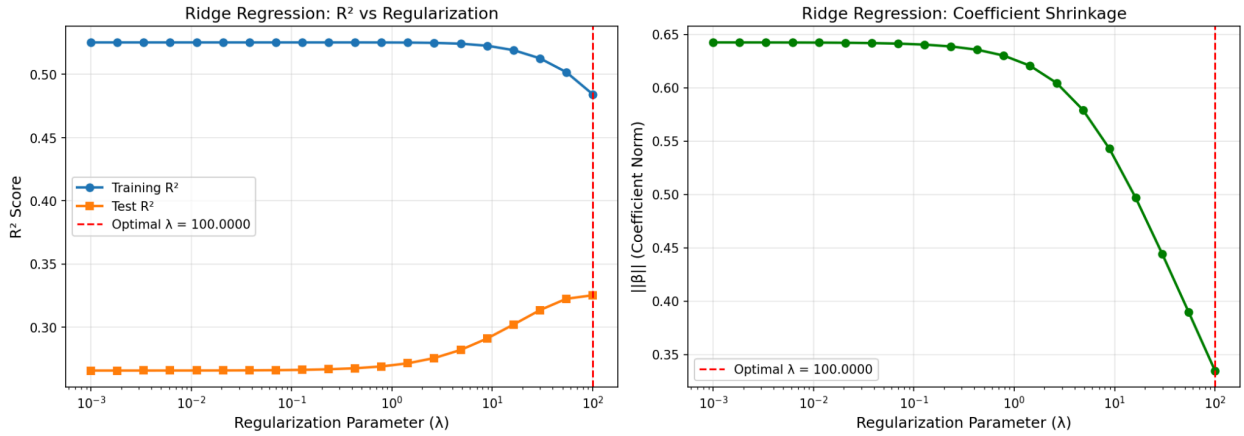

Figure 9:

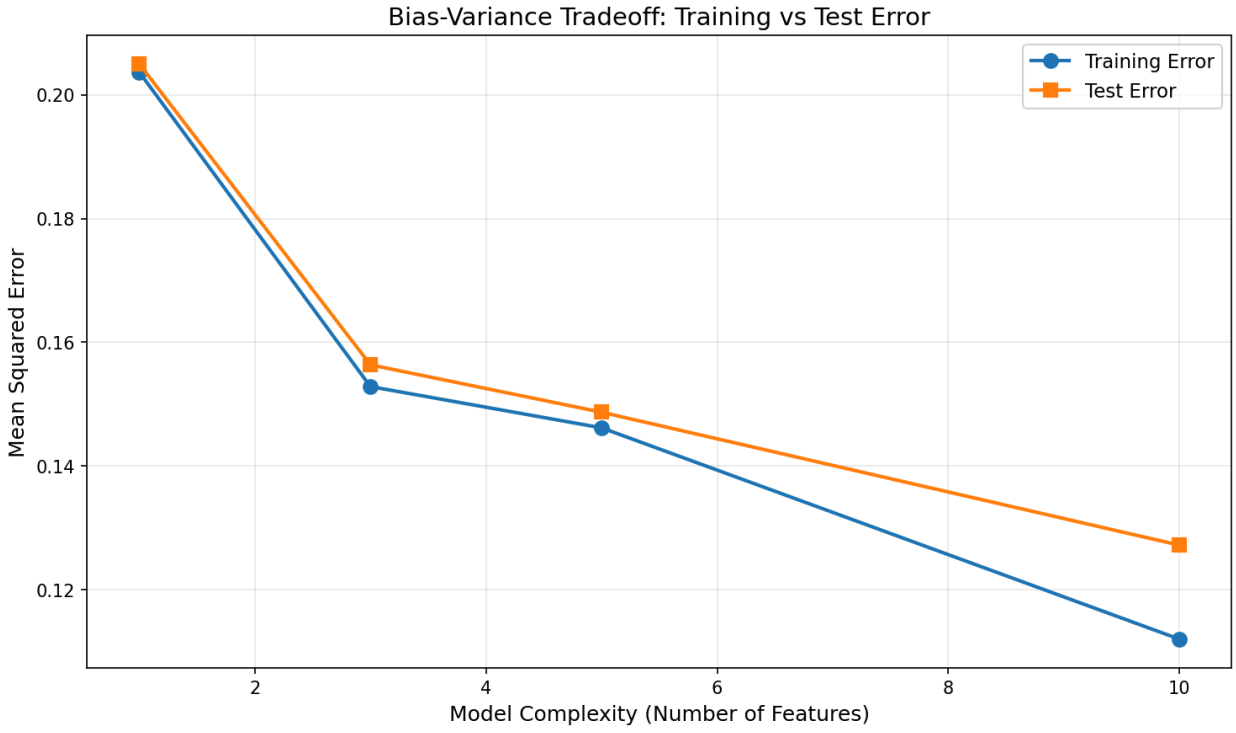

Figure 10:


Logistic Regression: Binary Classification Balanced Accuracy
(Normalized for Class Imbalance)
albeniz
bach
alkan
Composer 1 (True)
1.000
0.943
0.688
0.943
1.000
0.977
0.688
0.977
1.000
albeniz
bach
alkan
Composer 2 (Predicted)
1.0
0.9
0.8
0.7
0.6
0.5
Balanced Accuracy

VI. Sources

1. Chowdhury, A., et al. (2022). How Do You See Me? A Framework for "Musicologist-Friendly" Explanations. *Proceedings of the 23rd International Society for Music Information Retrieval Conference (ISMIR).*
2. Drengskapur. (2022). *MIDI Classical Music Dataset*. Hugging Face. Retrieved from https://huggingface.co/datasets/drengskapur/midi-classical-music
3. Febres, G., & Jaffe, K. (2017). Music viewed by its entropy content: A novel window for comparative analysis. *PLoS ONE 12*(10): e0185757. https://doi.org/10.1371/journal.pone.0185757
4. Gündüz, Güngör. (2023). "Entropy, energy, and instability in music". *Physica A: Statistical Mechanics and its Applications*, vol. 609, 128365. https://doi.org/10.1016/j.physa.2022.128365.
5. Herremans, D., Martens, D., & Sörensen, K. (2016). "Composer Classification Models for Music-Theory Building." In D. Meredith (Ed.), *Computational Music Analysis* (pp. 369-392). Springer.
6. Hsu, C. W., & Lin, C. J. (2002). A comparison of methods for multiclass support vector machines. *IEEE Transactions on Neural Networks*, 13(2), 415-425.