James “Collin” Guidry

Data Science Consultant

Composer Classification

Data Science Challenge

Overview

MIDI Composer Classification

Background & Objectives

- Given a collection of musical pieces from four composers, develop a method to predict whether a composition is not written by any of the four composers.
- System must predict given only 30 seconds of a composition.

Solution

- Ingest compositions as MIDI files and process them as 30-second chunks.
- Extract metadata from MIDI files to perform composer classification via a machine learning model.

Solution Methodology

Data Availability Influences Modeling Techniques

Primary Challenge

- The desired class to be predicted is not present in the training data.
  - Model is unable to learn which musical components are unlike those of the four composers.

Approach

- Train a multi-class classification model to distinguish between the four composers.
- Compositions with low probability of belonging to all four composers are classified as such.
- Cross validation methods are prioritized to increase performance on unseen data.

Solution Alternatives Considered

Balancing time, feasibility, and performance in approach selection

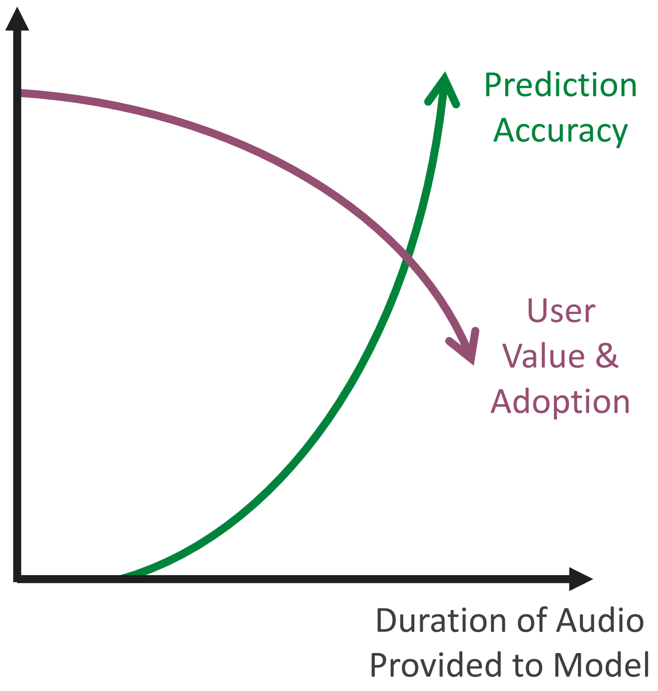
Approach	Classes	Final Prediction Based on	Pros	Cons
Single, multi-class classifier	0, 1, 2, 3	• Maximum threshold of predicted class probabilities. • "Unknown" if the maximum predicted probability is below the threshold.	• Simplified model architecture. • Greater flexibility for inclusion of other composers. • Better for holistic learning of all classes via a single model.	• Post-processing of predictions and threshold setting is separate from model training. • Choice of threshold can significantly affect performance.
Separate, binary classifiers for each composer	0,1; 0,1; 0,1; 0,1	• Comparison of independent model results. • "Unknown" if all models predict below probability of 0.5.	• Allows examination and tuning models by composer individually. • Resulting predictions more closely reflect the probability of belonging to a composer.	• More complex and time-consuming model architecture. • Accumulation of errors when combining predictions and Bernoulli trial type effects. • Class probabilities of each composer will not be normalized or sum to 1. Less explainable approach.
Single, unsupervised model or outlier detection	None	• Samples with outlier scores above a certain threshold are classified as “unknown."	• Does not require training on specific classes. • Can detect anomalies that do not fit any known class.	• Unproven method. • Threshold selection may be difficult. • Requires a well-defined notion of what constitutes an outlier and which features have out-of-sample variance.

Key Requirement: Audio Sample Duration

30-second chunk requirement highlights tradeoff between prediction accuracy and practicality

Tradeoffs Encountered

- **Shorter sample durations** make the solution more convenient and useful for users but may result in a lower accuracy rate.
- **Longer sample durations** (e.g., recording the entire song) results in more accurate classification but are less convenient for users.
- **Striking a balance** between sample duration and accuracy/value is crucial to ensure that the solution is both accurate and convenient for users.*



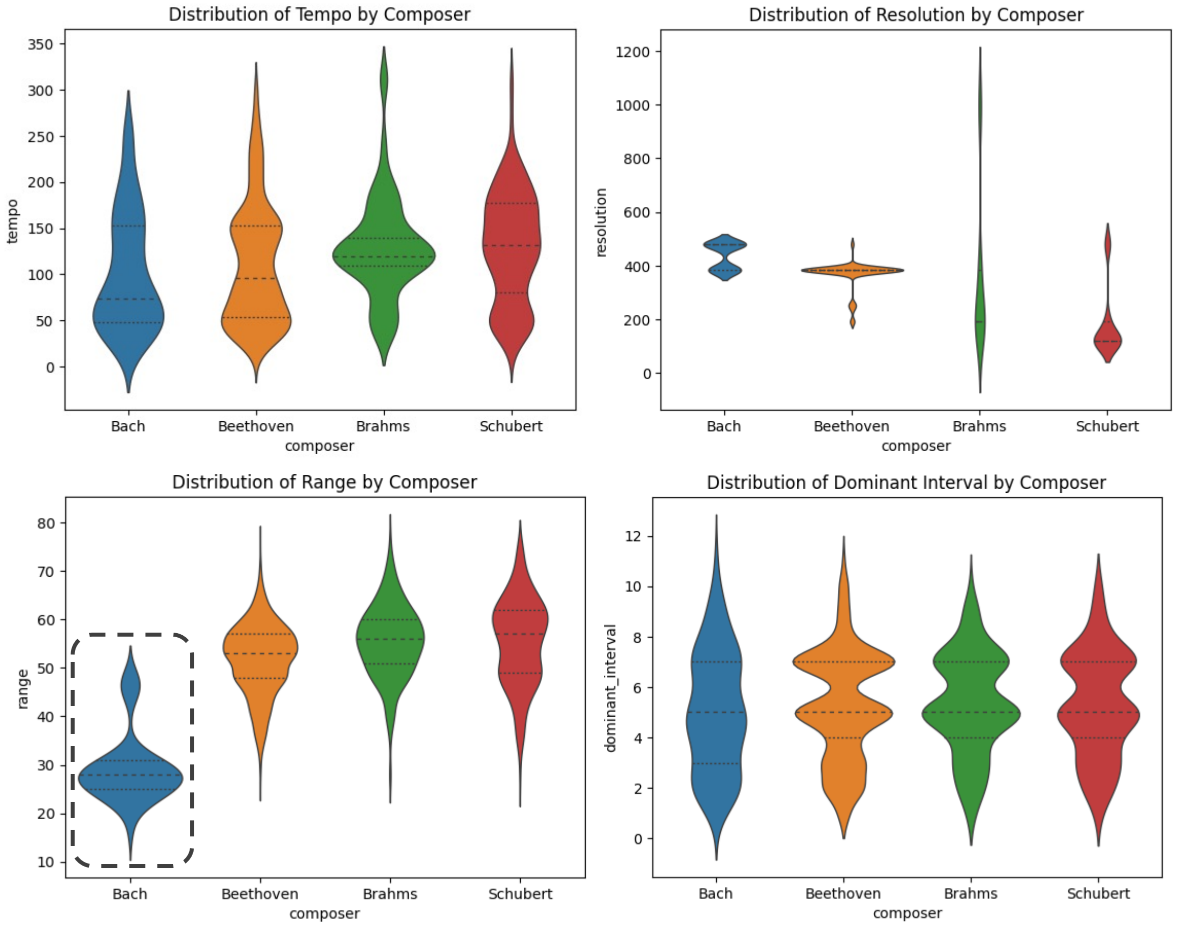
* Easily parameterized pipeline enables experimentation

Exploratory Data Analysis

Musical Characteristics with Visible Variance Among Composers

Features Analyzed

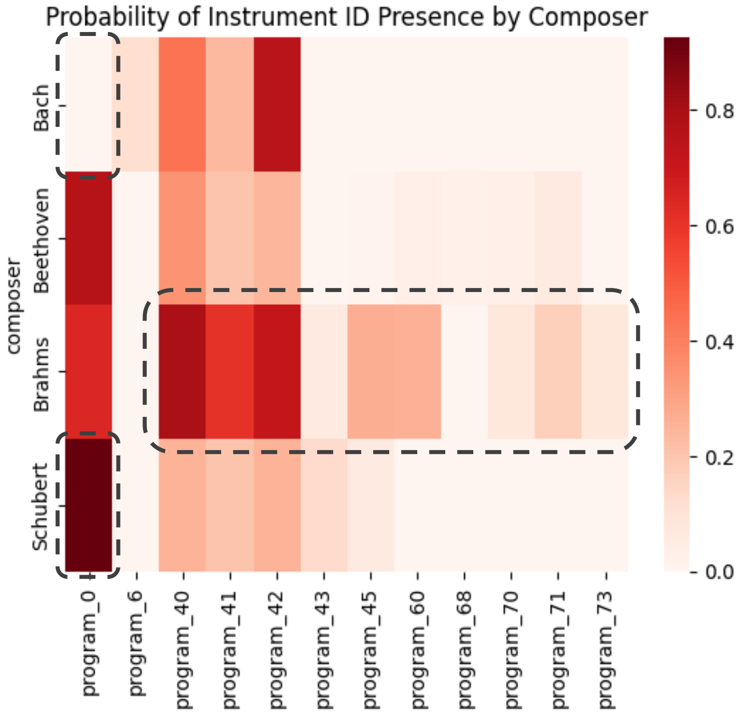
- **Tempo**
  - Beats per minute.
- **Resolution**
  - Number of ticks per quarter note.
  - Indicative of the smallest duration between notes.
- **Range**
  - Span of notes, from lowest to highest.
- **Dominant Interval**
  - Distance between the fifth note and the first note of a major or minor scale.
  - Used to measure tension or instability between chord progressions.

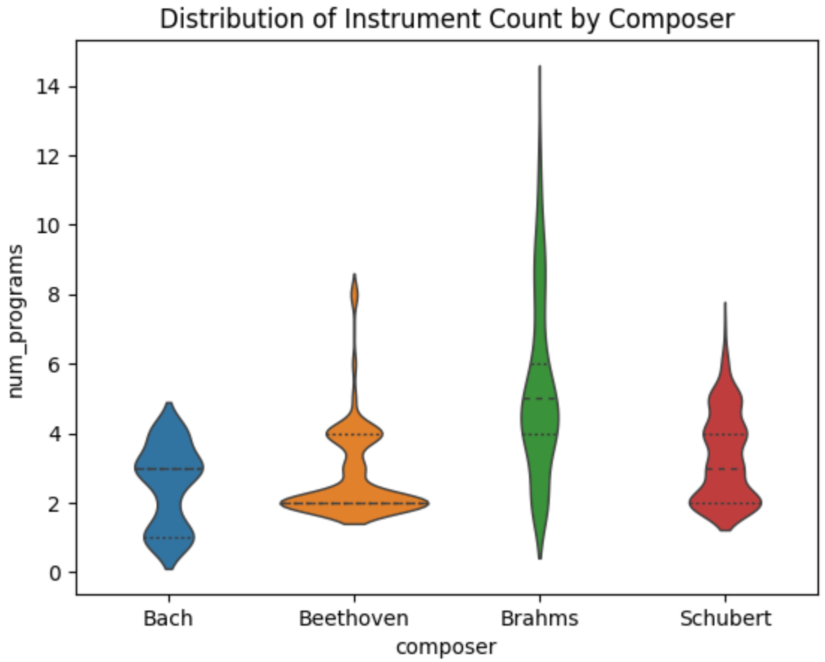


Exploratory Data Analysis

Instrument Usage Varies Among Composers

- Some **composers use specific instruments** which other composers do not.
- Certain composers consistently use **fewer total instruments** than others.



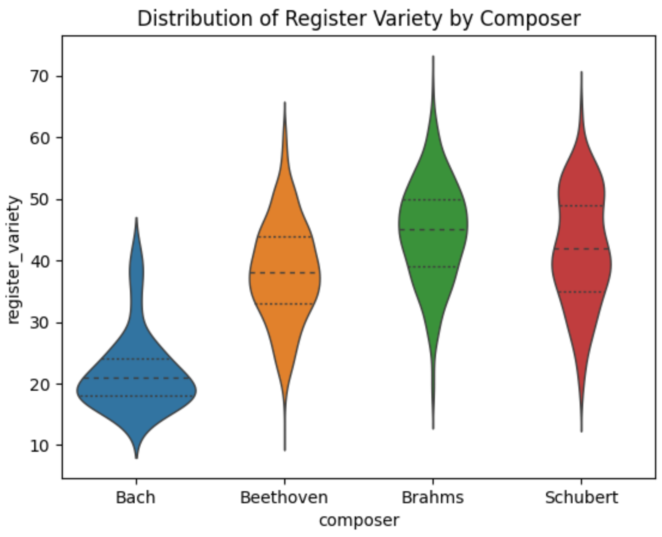


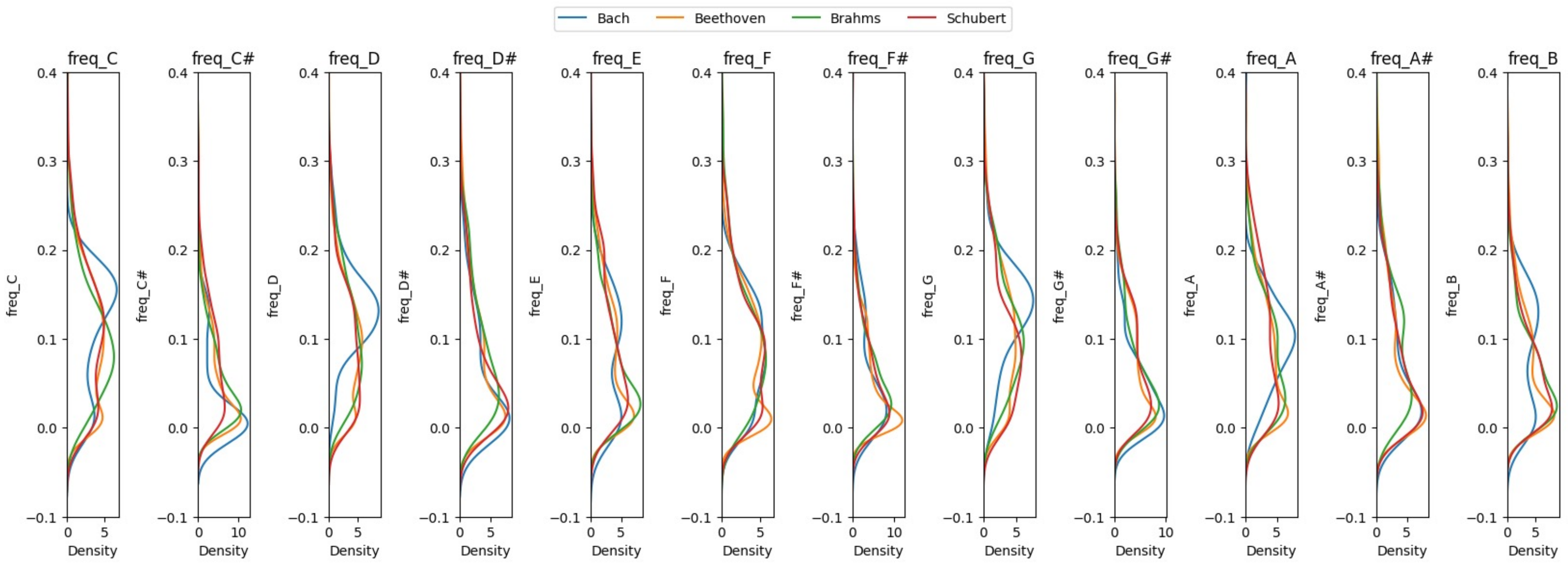
Exploratory Data Analysis

Pitch variety and frequency vary between composers, especially for Bach

Features Analyzed

- **Register Variety**
  - The number of pitches played at least once in a piece.
- **Pitch Frequency**
  - The percentage of notes in a sample belonging to a specific pitch.



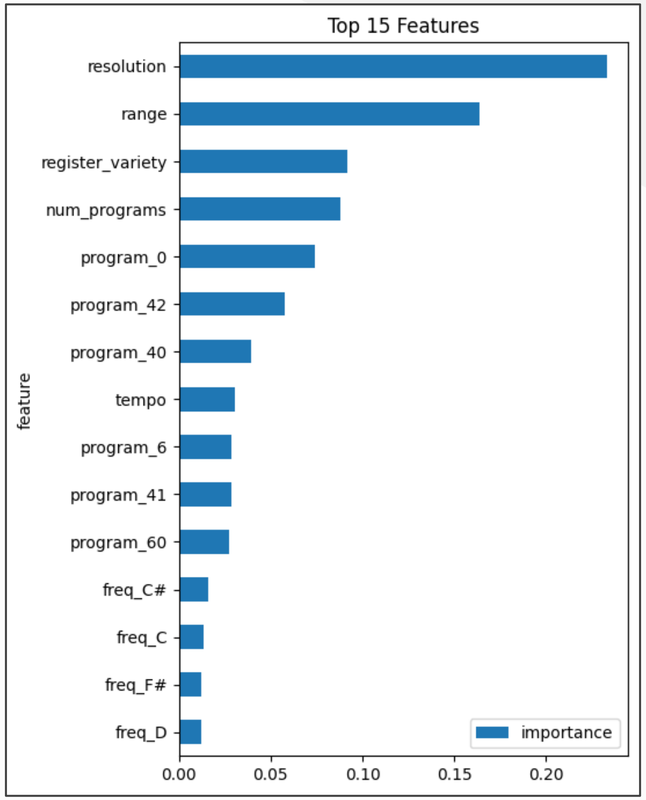


Feature Selection

Feature set informed by EDA and Predictive Power Scores

Features Chosen

- Tempo
- Resolution
- Range
- Presence of 11 distinct instruments
- Number of instruments
- Register (pitch) variety
- Pitch frequencies
  - C, C#, D, D#, E, F, F#, G, G#, A, A#, B
- Dominant interval



Selection Based on:

- Features with highest predictive power scores
- Exploratory Data Analysis
- Model cross validation accuracy via experimentation
- Multicollinearity checking

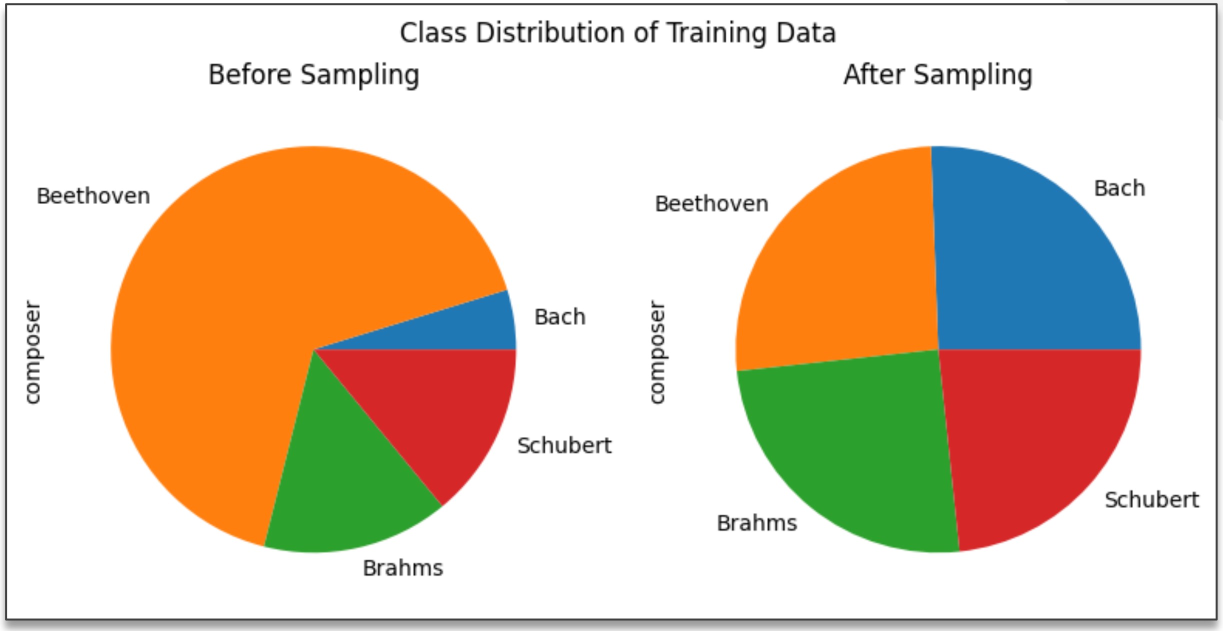
Pre-modeling Considerations

Training model with 30-second samples exacerbates class imbalance and risk of data leakage

Actions taken

- Stratified re-sampling implemented to correct for class imbalance.
  - Normalizes the training data to equal proportions of each class, to improve performance. Test data unaffected.
- Cross validation strategy prohibits samples of the same composition to be split across train, eval, and test sets.

Class Imbalance Observed and Corrected



Model Selection and Tuning

Broad search of models and parameters tested

- 6 types of machine learning models were fit, tuned, and evaluated on training data, using grid search cross validation.
  - Five random 80-20 splits used in each search, averaging results to select best hyperparameters.
- Random forest classifier yields highest accuracy.
- Final best parameters used to re-fit on entire training data.

Model	Accuracy (Training)	Hyperparameters Tested
RandomForest	67%*	n_estimators: [100, 200, 500], max_depth: [None, 5, 10, 15, 20]
GradientBoosting	66%	n_estimators: [100, 200, 500], max_depth: [3, 5, 7, 9]
LightGBM	65%	n_estimators: [100, 200, 500], 'max_depth: [3, 5, 7], 'learning_rate: [0.01, 0.1, 1]
AdaBoost	55%	n_estimators': [50, 100, 200], 'learning_rate': [0.01, 0.1, 1]
SVM	54%	C': [0.1, 1, 10], 'gamma': [0.1, 1, 10], 'kernel': ['rbf', 'linear']
KNN	41%	n_neighbors': [3, 5, 7], 'weights': ['uniform', 'distance'], 'algorithm': ['auto', 'ball_tree', 'kd_tree', 'brute']

* Training data is stratified – not representative of true accuracy

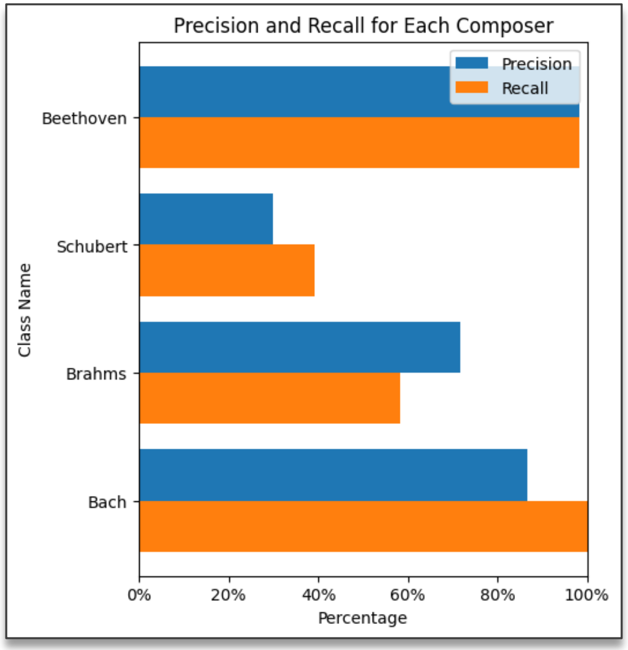
Model Validation & Results

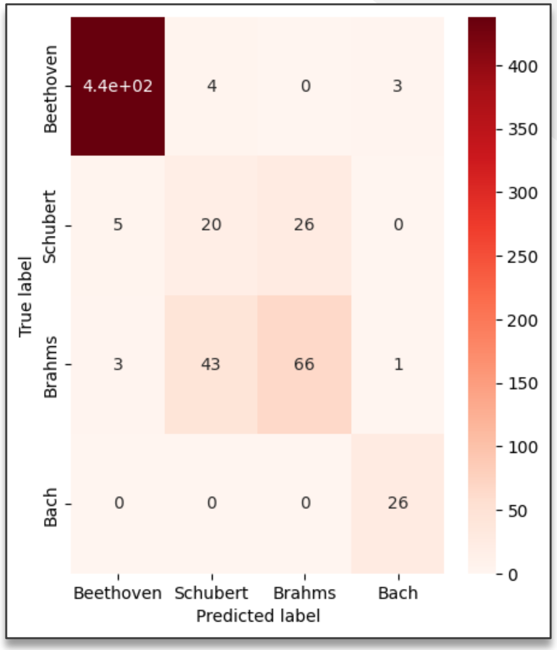
Performance on test set exceeds baseline

Test/Holdout Set Performance

- Classifier predicts composers with **85% accuracy** on unseen test data.
    - Baseline accuracy of 70%, based on most common class

- The model has high recall for **Beethoven** and **Bach** (**98%** and **100%,** respectively), but lower recall for **Schubert** and **Brahms** (**39%** and **58%** respectively).
    - Meaning, samples belonging to Beethoven and Bach classes were correctly identified more-so than that of Schubert and Brahms.

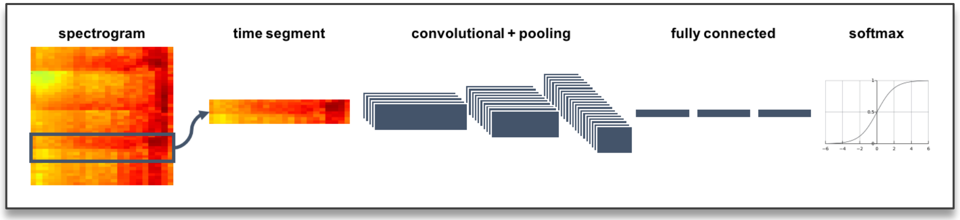




Recommendations / Areas of Maturity

MIDI Composer Classification

1. Immediate Enhancements
   - More advanced feature creation
   - Additional EDA
   - ML Improvements
     - PCA implementation
     - Categorical embeddings
2. Near-term Enhancements
   - Add subsequent training layer for probability calibration
     - Secondary logistic regression meta-model can estimate empirical probabilities more accurately.
     - Reduces need for use of arbitrary threshold.
   - Creation of separate binary classifiers
3. Long term "ground-up" solution
   - Convolutional Neural Network (CNN) and spectrogram-based features are state-of-the art approach for audio classification



   - More advanced technology stack and preprocessing efforts required, leading to more performant results.

Source

Conclusion

MIDI Composer Classification

- Developed an end-to-end solution for composer classification of 30-second audio samples, with 85% prediction accuracy on unseen compositions.
- Improvement opportunities identified to enhance real-world capabilities.