Identification of Instruments Through Sound Characteristics

Avery Cameron, Raymond Knorr, Mason Lane, Kegan Lavoy

Department of Engineering, University of Regina, Saskatchewan, Canada

Project Summary

Using the NSynth dataset, a large, high-quality dataset of annotated musical notes, our aim is to train a model to classify a note played on a violin against a note played on a piano [1]. We extract audio features from WAV files using the LibROSA python package, and use these features to train supervised learning models. We also recorded and labelled a smaller set of our own WAV files for violin and piano to compare against the model produced from the NSynth dataset. This smaller recorded dataset is called the RACK dataset. Algorithms being used to train models include Random Forests, AdaBoost, K-NN, and Support Vector Machines [2].

Project Objectives

- Accurately predict the instrument being played based off of an input audio file
- Compare against metrics of success for our project: an accuracy of 97.5%, precision of 95% and a recall of 95% [3][4]
- Identify audio features that are good indicators of an instrument, specifically piano and violin
- Investigate whether a model trained on the NSynth dataset can generalize to our own recorded samples

Methodologies

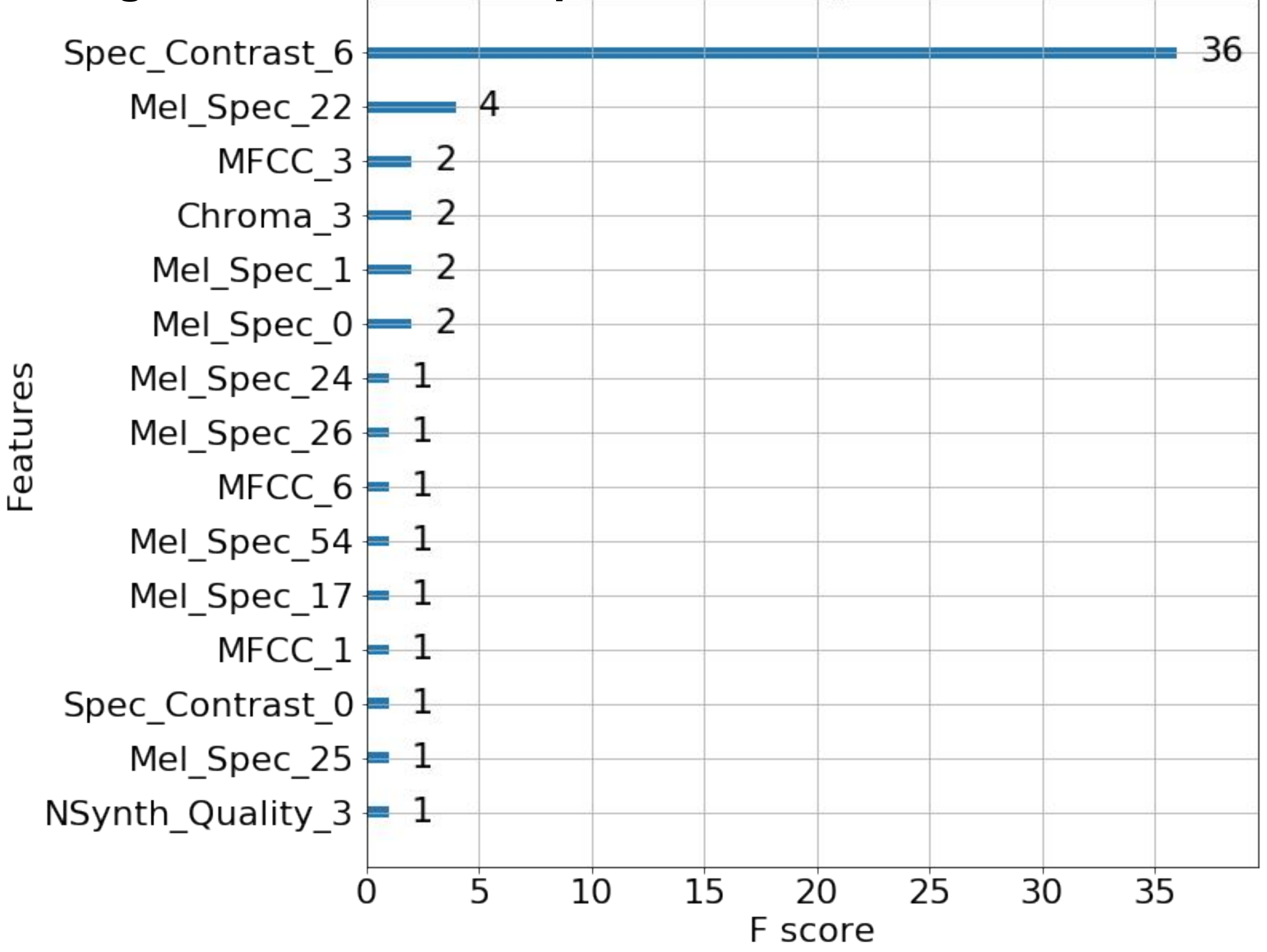
1. **Extract Features**
   Extracted features from WAV files using the LibROSA package. These features were: *Harmonic, Percussive, Chroma Energy Normalized, Chroma Constant Q, Mel Frequency Cepstrum Coefficients, Mel Spectrum and Spectral Contrast, and Spectral Centroid* [3][4][5]. Extracted target values from NSynth JSON files.

2. **Analyze Predictive Strength**
   Analyzed features to determine important predictors using XGBoost. The selected features were: *Harmonic, Percussive, Chroma Energy Normalized, Mel Frequency Cepstrum Coefficients, Mel Spectrum* and *Spectral Contrast*

3. **Save Features**
   Save the extracted features for each WAV file to a CSV file separated by dataset, train, validation and test. This saved processing time, removing the need to build our dataset frequently

4. **Train & Predict with Classifiers**
   Use the extracted datasets to train and predict with various classifiers.

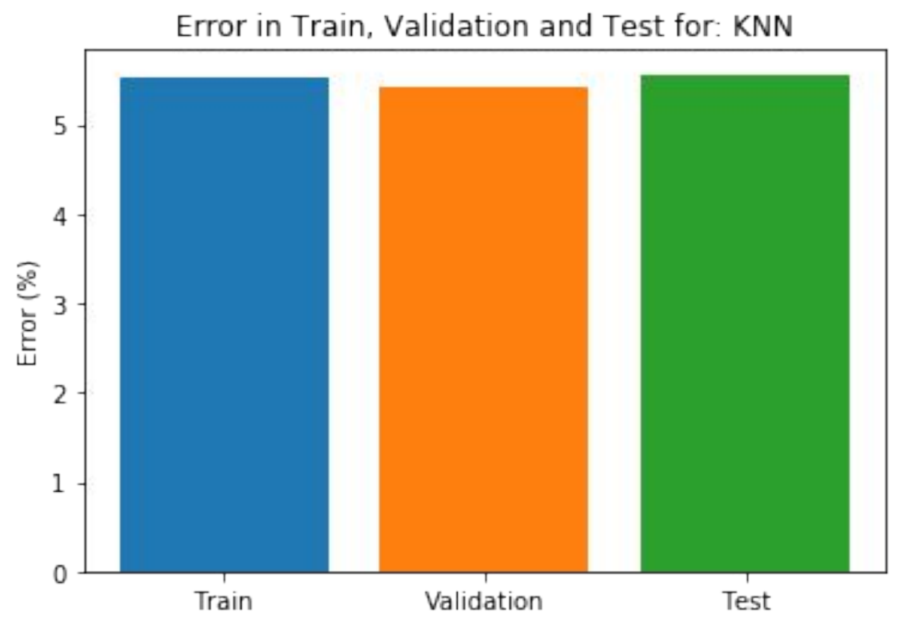
Figure 0: XGBoost Importance Results



Approaches & Results

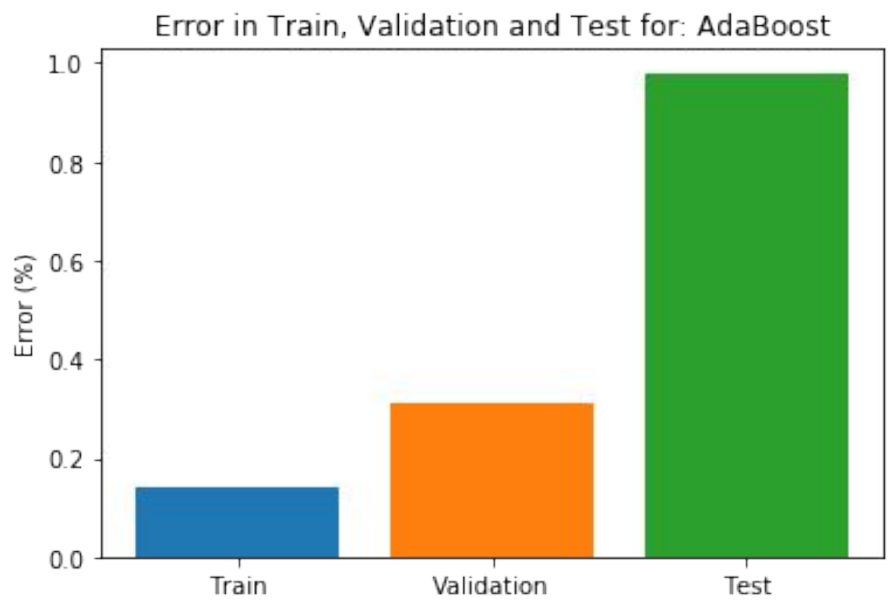
1. **KNN**
   Used a Scikit-learn KNN-Classifier. Hyperparameter K was optimized manually to 27 and 17 for Nsynth and RACK datasets respectively.

Figure 1: KNN Error %



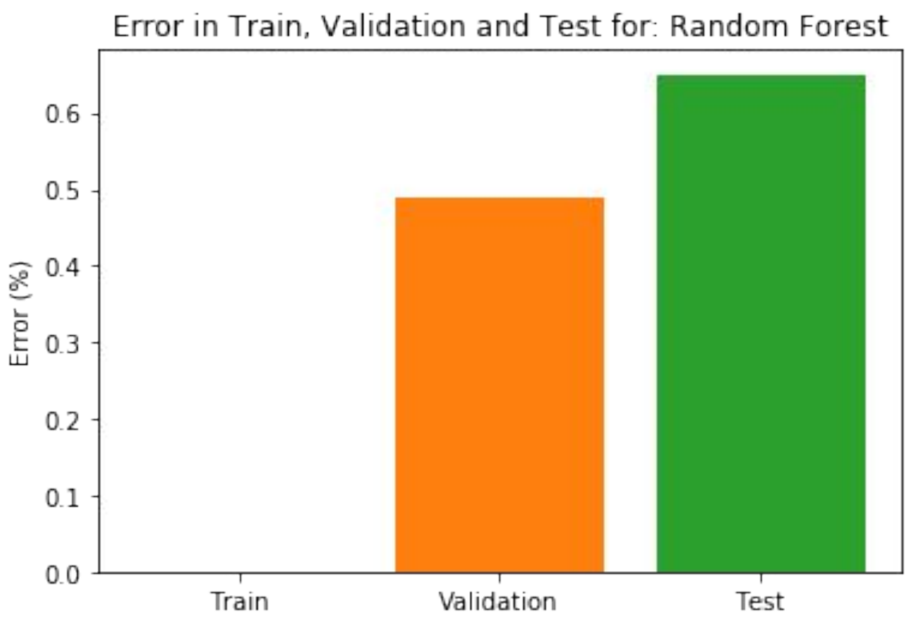
2. **AdaBoost**
   Used Scikit-learn AdaBoost classifier. Hyperparameter *estimator* value tuned to 98 using an automated search.

Figure 2: AdaBoost Error %



3. **Random Forests**
   Used Scikit-learn RandomForestClassifier. hyperparameters were optimized using a Scikit-learn RandomizedSearchCV.

Figure 3: Random Forest Error %



4. **SVM**
   Used Scikit-learn SVM implementation. Optimized parameters of an *rbf* kernel. *C* of 100, and a *gamma* of 0.00001 delivered the best results

Figure 4: SVM Error %

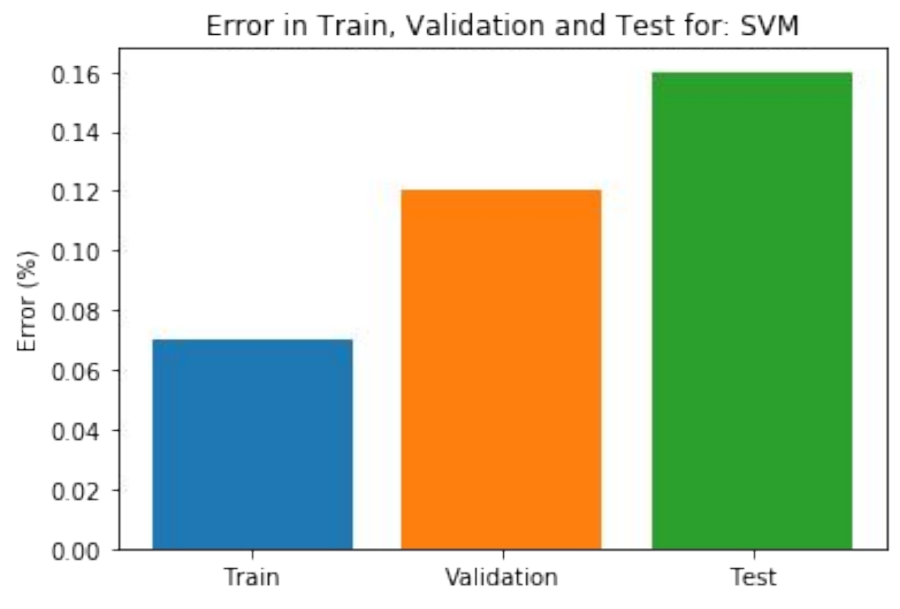


Figure 5: Overall NSynth Results

Classifier	Accuracy (%)	Precision (%)	Recall (%)	F1 (%)
KNN	94.44	93.46	95.33	94.39
AdaBoost	99.02	100	98.08	99.03
Random Forests	99.35	100	98.71	99.35
SVM	**99.84**	**99.67**	**100**	**99.83**

RACK Results

The classifiers were trained on the NSynth dataset and later applied to the RACK dataset for comparison. The accuracy for the predictions was 73.15% using SVM and AdaBoost classifiers with similar results for Random Forests and KNN. The RACK dataset contains both short and long violin note audio files. With the removal of the short violin notes, results improved to approximately 95% accuracy overall. We believe that the classifiers were misclassifying these short violin notes because their rapid decay and percussive quality more closely resembled the piano NSynth files over the violin.

Conclusion

Classification of piano versus violin was completed using the extracted audio features from WAV files in the NSynth dataset. These extracted features were used to train various classification techniques. The average accuracy of our techniques when training and testing on the NSynth dataset was 97.97%, and the best accuracy came from our SVM implementation at 99.84%. Testing using our RACK dataset resulted in an accuracy of ~95% for SVM, Random Forests, and AdaBoost.

References

1. Engel, J., Resnick, C., Roberts, A., Dieleman, S., Eck, D., Simonyan, K., & Norouzi, M. (2017). Neural Audio Synthesis of Musical Notes with WaveNet Autoencoders.
2. Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J., Passos, D., Brucher, M., Perrot, M., & Duchesnay, E. (2011). Scikit-learn: Machine Learning in Python
3. Seipel, F. (2018). Music Instrument Identification using Convolutional Neural Networks
4. Xu, M., Duan, L.-Y., Cai, J., Chia, L.-T., Xu, C., & Tian, Q. (2004). HMM-Based Audio Keyword Generation. *Advances in Multimedia Information Processing - PCM*
5. Kawwa, N. (2019, April 29). Can We Guess Musical Instruments With Machine Learning?