Automatic Polyphonic Music Transcription with Multilabel Classification

# Abstract

Automatic music transcription is an active research field in Music Information retrieval. The technologies of it have also been used in many mobile and web applications. The goal of music transcription is to determine the score-like representation of the input audio music. However the accuracy of the current transcription approach is still below that by human expert. Most of the current research focus on how to separate the pitches from the audio and estimate each of them. In this project instead of separating the feature of each note we are exploring the multilabel classification based automatic music transcription for monophonic and polyphonic piano music. We evaluate the performance of four different classifiers on various feature sets from the music.

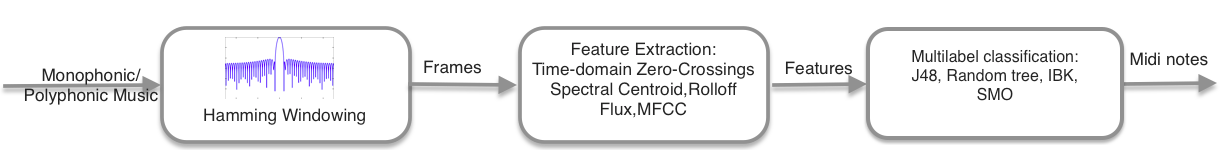
# 1 Introduction

Automatic music transcription is a process of translating audio music signal into music score sheet[[1]](https://paperpile.com/c/yFWBg3/wM3x). There are many applications of this technology, such as helping the instrument learning to get the music score sheet from a piece of music they want to play or allowing the musician or composer to record their improvised performance[[2]](https://paperpile.com/c/yFWBg3/oDWU). There various approach to achieve this goal[[3–7]](https://paperpile.com/c/yFWBg3/bOxo+TQEf+Ihbh+fcD9+mE8Q).

In the real world, music form can be dramatically complicated composed with different instrument and human voice. To address such kind of problem is a big challenge. In this project we are focusing on the problem of monophonic(includes a single melodic line with no accompaniment) and polyphonic(multiple melodic voices which are to a considerable extent independent from or in imitation with one another) piano music. We experiment on monophonic and polyphonic music compose with 2 to 4 notes at a time.

In this work, we build our own model. First apply windows on the input music files, segment it into frames consist with 512 samples. Then we extract the 63 attributes from each frame. The acoustic features are then analysis with Weka[[8]](https://paperpile.com/c/yFWBg3/DFXs) and Meka[[9]](https://paperpile.com/c/yFWBg3/JVmH). We select several groups of features for comparison of the classification in accuracy. We apply multilabel classification[[10]](https://paperpile.com/c/yFWBg3/s1kY) with 4 classifiers: IBK[[11]](https://paperpile.com/c/yFWBg3/hdLU),J48[[12]](https://paperpile.com/c/yFWBg3/uYxo),SMO[[13]](https://paperpile.com/c/yFWBg3/Dyka) and RandomTree[[14]](https://paperpile.com/c/yFWBg3/rUkm). The result of each classifier and attribute groups are visualized in the evaluation section.

The overview of our model can be described in the below figure. We train and test our classifiers based on this model.



This report is organized in the following sections: Dataset section introduce the dataset we used for this project. Feature Extraction section describe the tool we use for the extraction as well as the meaning of each set of features. Feature selection section indicate how the extracted feature are analysed and grouped. Classification section describe the classifiers and the parameters of each one for each dataset and how we perform the training and testing. Evaluation section give the result of the our experiments with each dataset and visualize the result in a more perceptive way. Finally we conclude the result and indicate the feature work.

# 2 Dataset

We evaluate our proposed model with the MAPS dataset[[15]](https://paperpile.com/c/yFWBg3/FQ8J). The dataset consists of audio and corresponding annotations for isolated sounds, chords and complete pieces of piano music. For monophonic music, we use 2 second long notes(NO) in ISOL set and categorized this dataset as P1 later in this report. For the polyphonic music, we use polyphonic level 2 to level 4 in RAND set, they are corresponding to P2, P3, P4 as mentioned later in this report.

# 3 Feature Extraction

In signal processing domain, feature extraction is one of the most widely used method to analysis the signal. There are many features of audio signal. In our work we explored some of them and see how much they contribute in automatic music transcription.

## 3.1 Acoustic Feature Sets

In our work, the Acoustic Features we are using are described in this section, each acoustic feature set consist of several sub features which will be extracted based on different parameters.

**Time-domain Zero-Crossings**[[16]](https://paperpile.com/c/yFWBg3/xhps) is a point where the sign of a mathematical function changes (e.g. from positive to negative), represented by a crossing of the axis (zero value) in the graph of the function. It is a commonly used term in electronics, mathematics, sound, and image processing.

**Spectral Centroid**[[17]](https://paperpile.com/c/yFWBg3/T2wW) is a measure used in digital signal processing to characterise a spectrum. It indicates where the "center of mass" of the spectrum is. Perceptually, it has a robust connection with the impression of "brightness" of a sound.

**Rolloff**[[18]](https://paperpile.com/c/yFWBg3/PtsU)is the steepness of a [transmission function](https://en.wikipedia.org/wiki/Transfer_function) with [frequency](https://en.wikipedia.org/wiki/Frequency).

**Flux**[[19]](https://paperpile.com/c/yFWBg3/7VpY) is a measure of how quickly the [power spectrum](https://en.wikipedia.org/wiki/Power_spectrum) of a [signal](https://en.wikipedia.org/wiki/Signal_(electrical_engineering)) is changing, calculated by comparing the power spectrum for one frame against the power spectrum from the previous frame. More precisely, it is usually calculated as the 2-norm (also known as the [Euclidean distance](https://en.wikipedia.org/wiki/Euclidean_distance)) between the two [normalised](https://en.wikipedia.org/wiki/Audio_normalization) spectra.

**Mel-Frequency Cepstral Coefficients**[[20]](https://paperpile.com/c/yFWBg3/9q36) are coefficients that collectively make up an MFC. In [sound processing](https://en.wikipedia.org/wiki/Sound_processing), the mel-frequency cepstrum (MFC) is a representation of the short-term [power spectrum](https://en.wikipedia.org/wiki/Power_spectrum) of a sound, based on a [linear cosine transform](https://en.wikipedia.org/wiki/Cosine_transform) of a [log power spectrum](https://en.wikipedia.org/wiki/Power_spectrum) on a [nonlinear](https://en.wikipedia.org/wiki/Nonlinear_system) [mel scale](https://en.wikipedia.org/wiki/Mel_scale) of frequency.

## 3.2 Feature extraction tool

Marsyas (Music Analysis, Retrieval and Synthesis for Audio Signals) is an open source software framework for audio processing with specific emphasis on Music Information Retrieval applications[[21]](https://paperpile.com/c/yFWBg3/NdRb). Bextract is one of the most powerful executables provided by Marsyas. It can be used for complete feature extraction and classification experiments with multiple files. However its classification feature is limited and did not fulfill the requirement of our project, so we only utilize its feature extraction functionality.

## 3.3 Feature extraction strategy

We use bextract provided by Marsyas to extract the Acoustic Feature Sets mentioned before. We extract the features base on each frame. By default the frame size of bextract is 512 samples. Totally it extracts 63 features from each frame. All the extracted features are stored in .arff format file. Bextract does labeling for each item is the item is single label. However it does not support multi-labelling. We post-process the output .arff file from bextract with our mulit-labeling script to have the multi-label .arff files which can run directly on Meka. We label the output of each frame with the note numbers from the corresponding .txt file of each audio file.

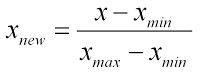
# 4 Feature Selection

Feature selection was done in order to reduce the original feature size of 63 attributes into a more reasonable size. This section outlines the approach and result of feature set selection. The development team was not able to invest the appropriate time required to learn sufficient domain knowledge to naturally prune the attribute set themselves. In addition, the team wanted to avoid tailoring a feature subset to a specific classifier for best results and performances, instead wanting the best general subset to give the most useful result for any classifier. As such, systematic filter-based[[22]](https://paperpile.com/c/yFWBg3/ZqrX) approaches were used to create 6 feature sets per feature extraction data set (i.e., for audio data sets P1 - P4). The results of the 6 feature sets were then used in 5 different classifiers for 5-fold cross validation.

The WEKA ‘select attributes’ toolset[[23]](https://paperpile.com/c/yFWBg3/SviS) was used to generate 5 feature subsets, including two entropy sets with threshold values of 0.03 and 0.05, a learner subset, and two pearson correlation coefficient subsets with threshold values of 0.02 and 0.03. These threshold values were chosen by observation as they provided either approximately half the original feature set, or slightly less. 50% of the original 63 features was considered a reasonable subset size.

# 5 Normalization

Normalization, which scales all numeric variables in the range [0,1]. One possible formula is given below:



In data mining, normalization is a technique used to preprocess the dataset. We apply the normalization in our P1 original dataset to get the idea that if this technique affect the accuracy.

# 6 Multilabel classification

In [machine learning](https://en.wikipedia.org/wiki/Machine_learning), multi-label classification and the strongly related problem of multi-output classification are variants of the [classification](https://en.wikipedia.org/wiki/Statistical_classification) problem where multiple target labels must be assigned to each instance. In music information retrieval field, multi-label classification are usually used in genres or emotion recognition but not in transcription.

Many current research of automatic transcription either only focus on the main melody or approach by separating the multiple melodic and deal with one of them at a time.

In our project, we explore multilabel classification for polyphonic transcription.

We select 4 classifiers in our experiment for all datasets from P1 to P4. The four classifiers are IBK,J48,SMO and RandomTree. We run training and testing in 5 fold cross-validation for each dataset. The problem of monophonic become single label classification.

# 7 Evaluation

We run the training and testing for P1 dataset in Weka while P2 to P4 dataset in Meka. We collect the result of each experiment and analysis them in this section.

## 7.1 P1 dataset

This section summarizes the percentage output of the correctly classified instances for each feature set, as well as provides the visualization of each feature set’s result J48 tree.

## Filter-based results

Table X summarizes the result of each filter-based feature subset as used in 4 different classifiers. SMO took the most time to run and provided the worst accuracy while IBK consistently provided the best. However, randomtree executed the fastest while still providing highly accurate results.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Subset (count) | **IBK** | **J48** | **SMO** | **randomtree** |
| **P1 Original (63)** | 97.9833 | 88.9794 | 54.7562 | 87.6555 |
| **P1 Entropy 03 (31)** | 97.3751 | 88.5446 | 46.1134 | 87.6285 |
| **P1 Entropy 05 (14)** | 91.4875 | 87.0854 | 27.8789 | 86.8748 |
| **P1 Pearson 02 (36)** | 98.6161 | 88.5894 | 51.8581 | 87.8306 |
| **P1 Pearson 03 (27)** | 99.4104 | 87.2656 | 48.9633 | 87.2597 |
| **P1 Learner (28)** | 99.1372 | 88.1047 | 48.5835 | 87.8594 |

Table X: % Accuracy results of different feature subsets on different classifiers

J48 Analysis

The results of each J48 tree classifier were analyzed further and visualized by looking at the confusion matrix and result tree.

The root node and first layer were recreated using Google diagrams since the extremely large trees created in Weka were too difficult to read. To conserve space, note the substitutions for Figure X in Table X.

|  |  |
| --- | --- |
| **Shorthand** | **Value** |
| PP512Ch0 | Power\_powerFFT\_WinHamming\_HopSize512\_WinSize512\_SumAudioCh0 |
| M20\_PR\_MCHRA | Mem20\_PeakRatio\_Minimum\_Chroma\_A |

Table X: Notation for J48 tree in Figure X

p1_original.png

*Figure X: Root Node and First layer for all P1 instances*

It was found that the initialization of each tree was the same. That is, each feature subset derived the same root node and first layer of their J48 classifier. However, to show the differences between the trees, Figures X through X are the general structures of each.

Figure X is the general tree shape of the full 68 attribute feature set. There is nothing particularly special about this tree. This will be the baseline of how accurate the subset trees are.

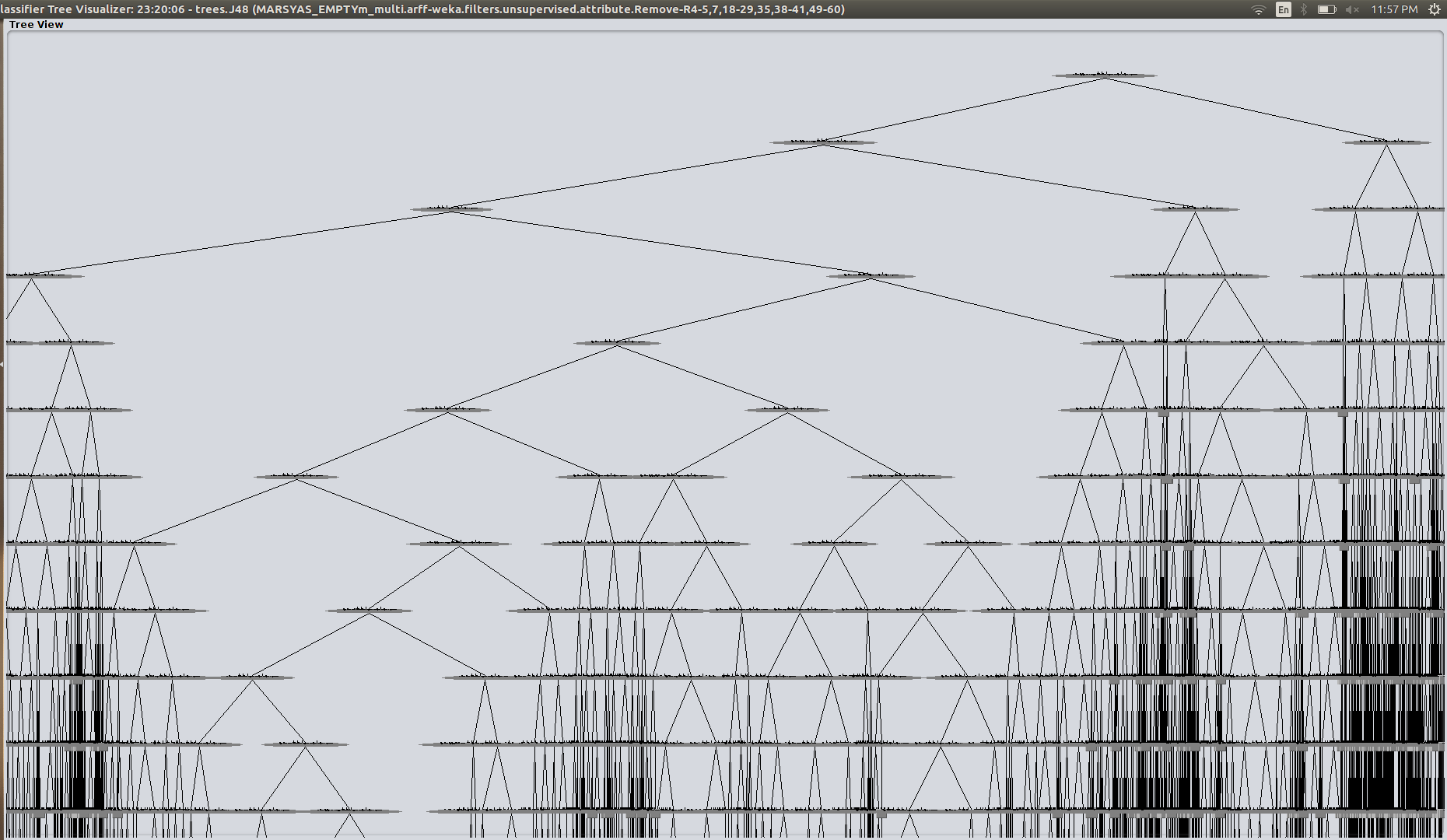


Figure X: General J48 Tree Shape for original feature set

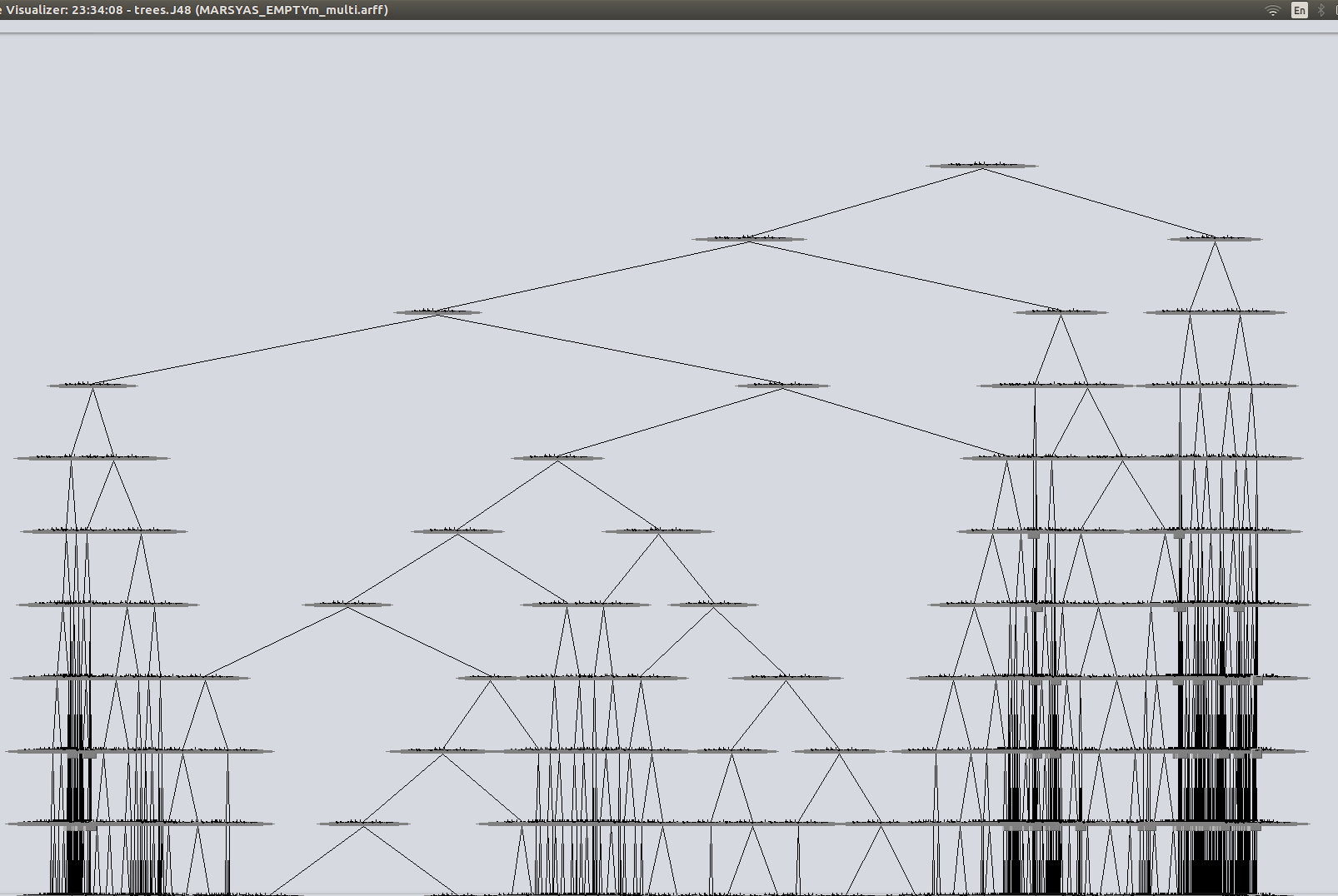


Figure X: General J48 Tree shape for entropy feature set (threshold of 0.03)

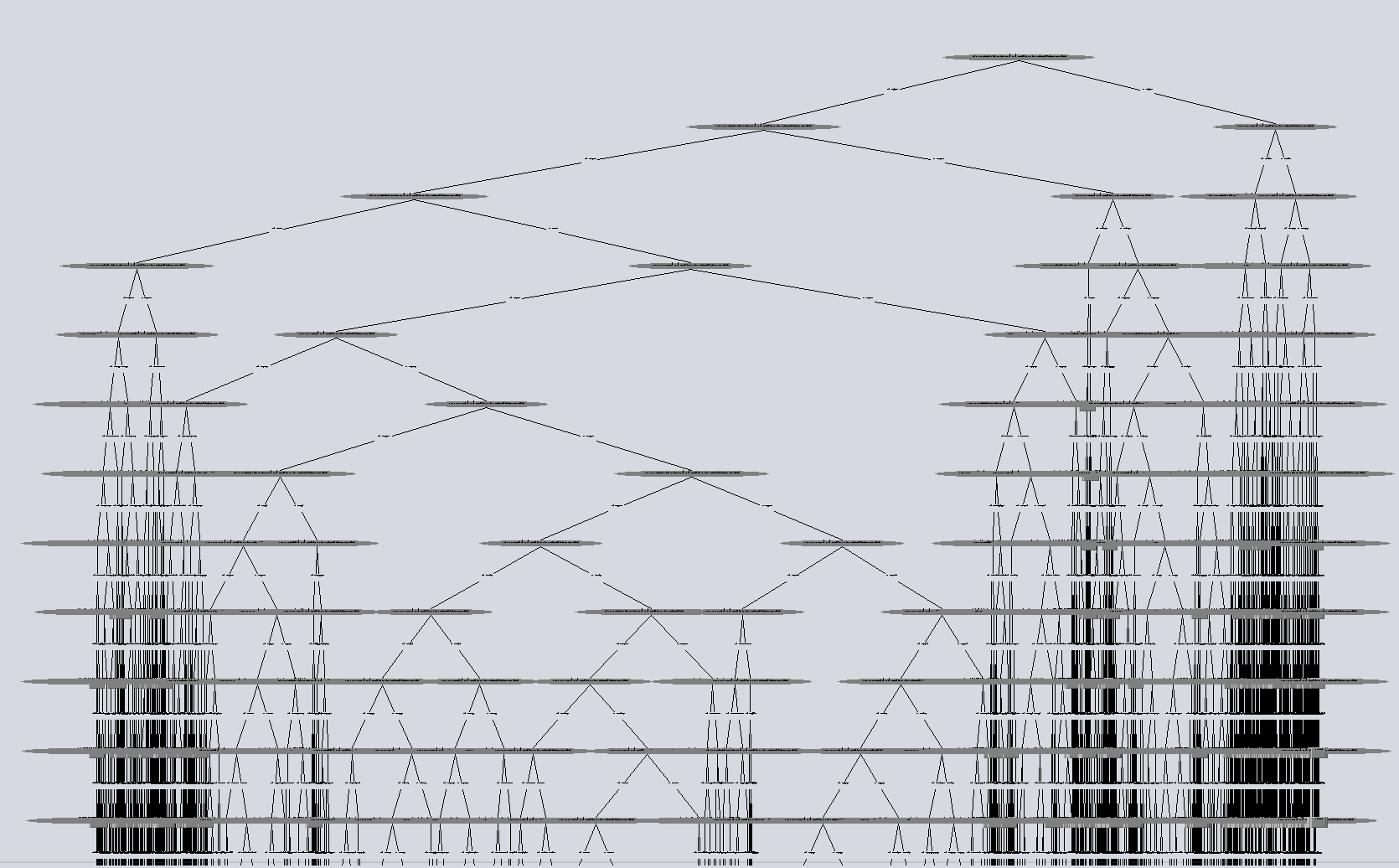


Figure X: General J48 Tree Shape for the P1 entropy feature set (threshold of 0.05)

Figure X and X represent the two information gain trees. Compared to the original, they have a greater depth (are longer). However, the tree with the more inclusive entropy threshold of 0.03 has more similarities than the one with threshold 0.05 (14 attributes) which appears to have significantly change the LHS of the tree.

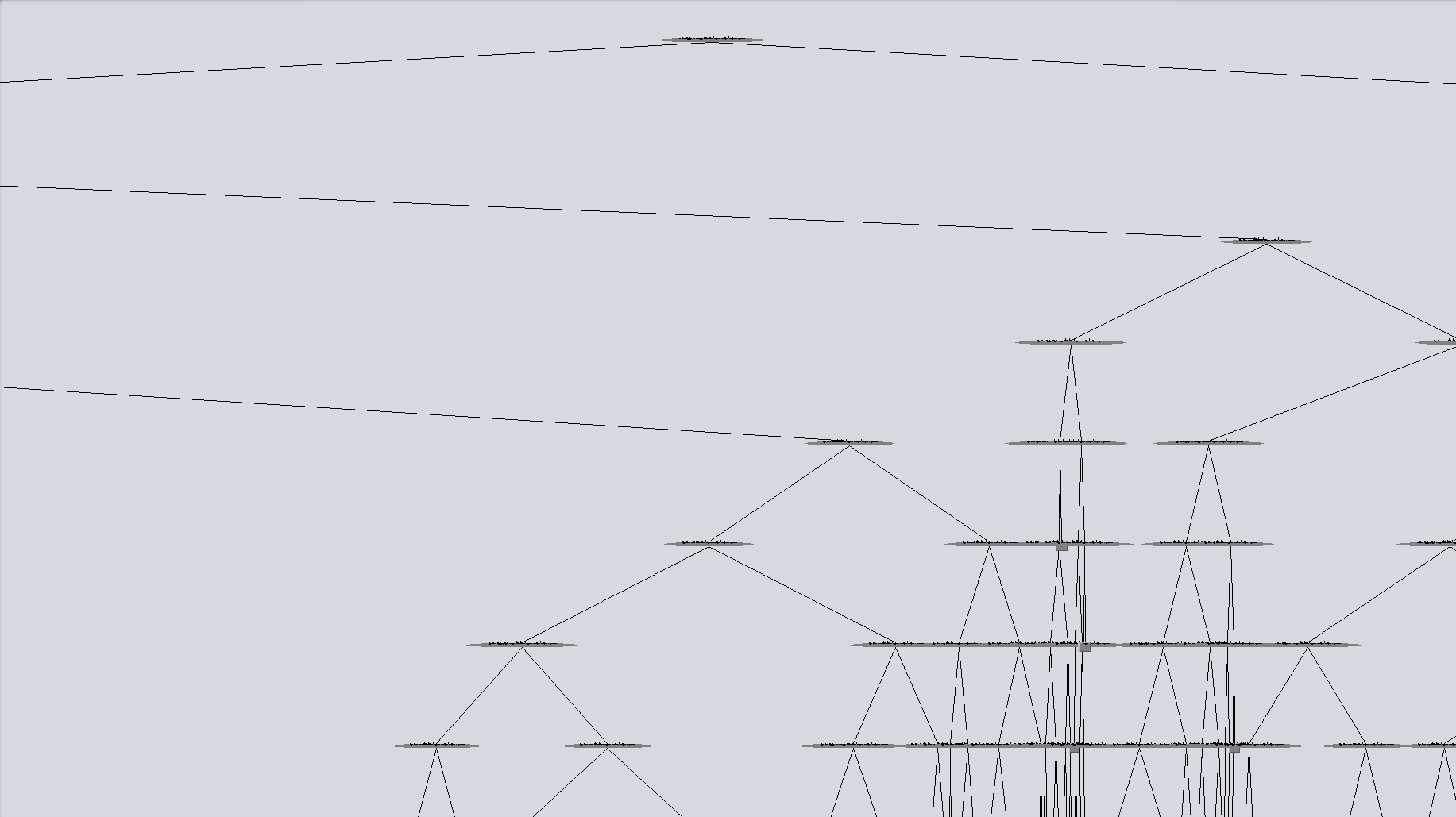


Figure X: General Tree shape for the P1 learner feature set

Figure X represents the tree for the learner-based feature selection. As clearly shown in the image, this tree is significantly wider than the original, and less deep.



Figure X: General tree shape for P1 Pearson 02 feature set

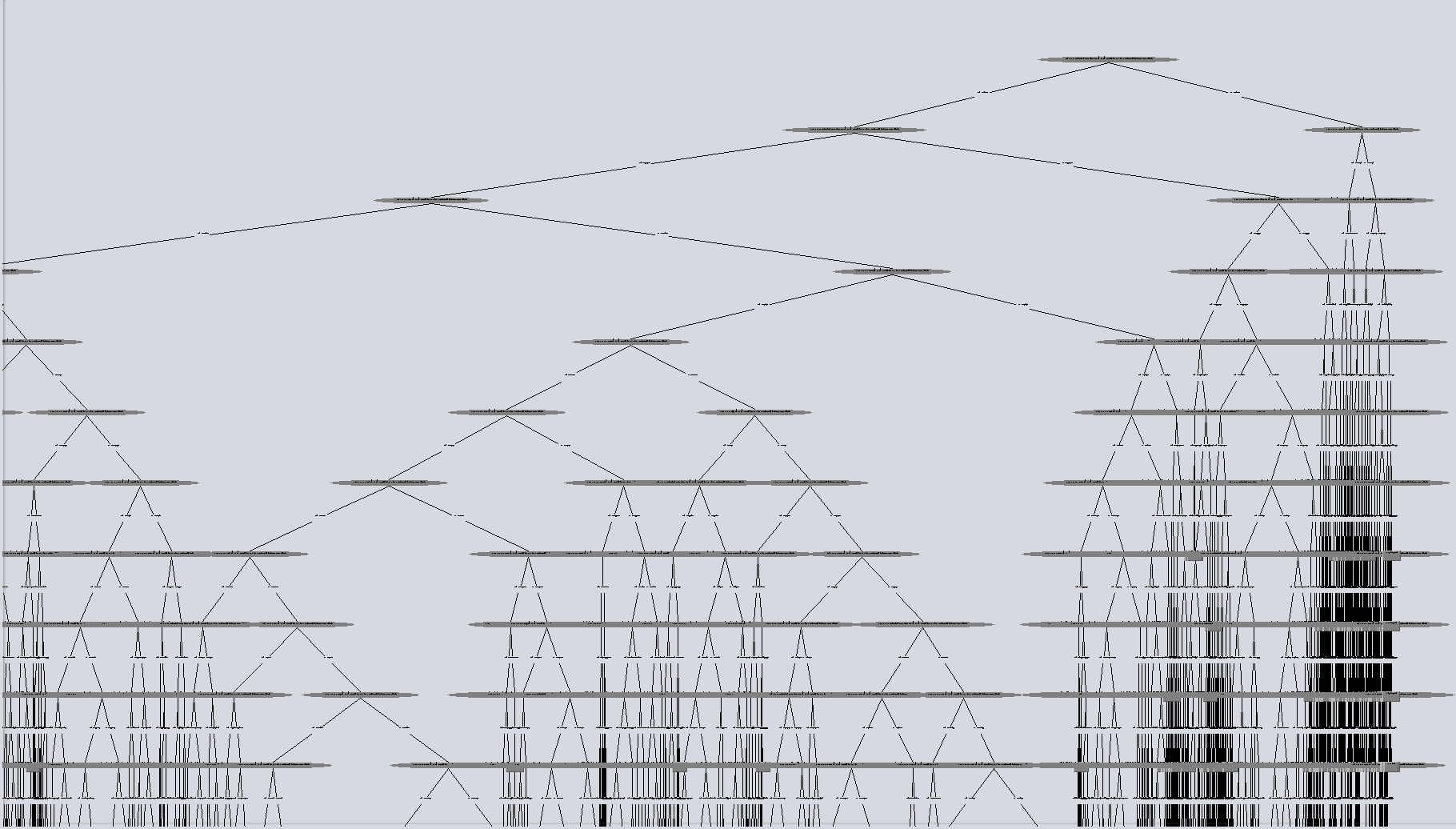


Figure X: General tree shape for P1 Pearson 02 feature set

Finally, figure X and X represent the two trees for the pearson correlation coefficient filters. Similar to entropy, these trees are quite similar to each other and to the original.

Similarly, the confusion matrix of each feature set was reviewed using the matlab script available in Appendix A. The script found the intersection of notes with the highest misclassified percentage value. Table X summarizes the results.

|  |  |  |
| --- | --- | --- |
| **Feature set (threshold)** | **Intersection** | **Value / ??** |
| Original | (m28, m27) | 55 |
| Learner | (m28, m27) | 66 |
| Entropy (05) | (m23, m24) | 73 |
| Entropy (03) | (m28, m27) | 75 |
| Pearson (02) | (m28, m27) | 61 |
| Pearson (03) | (m22, m21) | 72 |

Table X: P1 Feature subset highest confusion results.

It is not surprising that the feature sets have similar misclassified note intersections. It is reasonable to expect that similar sounding notes would be consistently misclassified no matter which subset and classifier were used.

## Normalization V.S. Original data in accuracy

We only compare the result of **P1 Original (63)** dataset. The reason we didn’t apply it on other dataset is that we didn’t see any obvious improvement with normalization preprocess.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | **IBK** | **J48** | **SMO** | **Random tree** |
| **Without Normalization** | 97.9833 | 88.9794 | 54.7562 | 87.6555 |
| **With Normalization** | 97.9833 | 88.9801 |  | 87.6765 |

## 7.2 P2 dataset

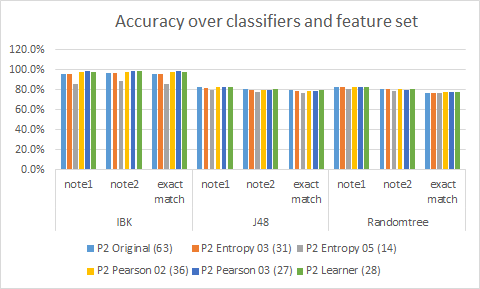
This section summarizes the percentage output of the accuracy for each feature set. The accuracy is considered as note1 being correctly classified, note2 being correctly classified, all notes are exactly being classified. The train and test time are also summarized in this section. For P2-P4 dataset SMO is almost impossible, I try to run it on the original dataset and it took overnight and didn’t finish. So we skip SMO classifier for P2-P4 due to its bad performance. As we can see in the result IBK has highest accuracy but takes much longer time to run compare to the other two classifier. For the dataset P2 Entropy 05 dataset consistently has the worst performance while the other ones has similar accuracy result.

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Subset (count)** | **IBK** | | | **J48** | | | **Random Tree** | | |
| note1 | note2 | exact match | note1 | note2 | exact match | note1 | note2 | exact match |
| **P2 Original (63)** | 96.3 | 96.9 | 96.1 | 83.1 | 80.9 | 79.7 | 82.7 | 80.9 | 76.9 |
| **P2 Entropy 03 (31)** | 96.3 | 96.9 | 96.1 | 82.3 | 80.1 | 78.8 | 82.7 | 80.9 | 76.9 |
| **P2 Entropy 05 (14)** | 86.2 | 88.8 | 85.5 | 80.2 | 78.0 | 76.4 | 80.9 | 78.4 | 76.9 |
| **P2 Pearson 02 (36)** | 97.6 | 98.0 | 97.5 | 82.5 | 80.2 | 79.0 | 82.9 | 80.6 | 78.2 |
| **P2 Pearson 03 (27)** | 99.0 | 99.1 | 99.0 | 82.5 | 80.1 | 78.8 | 82.6 | 79.7 | 78.1 |
| **P2 Learner (28)** | 98.4 | 98.6 | 98.3 | 82.9 | 80.7 | 79.5 | 82.7 | 81.0 | 78.3 |

Table X: % Accuracy results of different feature subsets on different classifiers

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Subset (count)** | **IBK** | | | **J48** | | | **Random Tree** | | |
|
| **P2 Original (63)** | 248.6 | | | 64.956 | | | 8.3 | | |
| **P2 Entropy 03 (31)** | 205.2 | | | 46.8 | | | 5.5 | | |
| **P2 Entropy 05 (14)** | 334.4 | | | 33.7 | | | 6.0 | | |
| **P2 Pearson 02 (36)** | 226.5 | | | 51.9 | | | 12.3 | | |
| **P2 Pearson 03 (27)** | 139.3 | | | 43.4 | | | 5.2 | | |
| **P2 Learner (28)** | 197.0 | | | 43.5 | | | 8.4 | | |

Table X: Train/Test time in seconds

Figure X: Accuracy for each dataset and classifiers

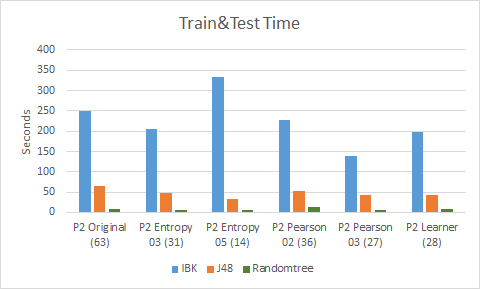


Figure X: Train and Test Time for each dataset and classifiers

## 7.3 P3 dataset

## 7.4 P4 dataset

## 7.5 Note base accuracy comparison for PN dataset

# 8 Conclusion

***Save this section for last! Erika will write on Wednesday after reviewing/editing the report. Really appreciated, from Nora...***

# 9 Future work

The result of our experiment show a satisfied result. However, there is an important problem need to be solved before it can be used in the real world. In the real music, we never know how many notes a piece of music consist with. The future work should focus on undetermined number of note polyphonic music transcription using multilabel classification.

# Reference

1. [Duan Z, Benetos E. Automatic music transcription. ISMIR; 2015.](http://paperpile.com/b/yFWBg3/wM3x)

2. [Benetos E, Dixon S, Giannoulis D, Kirchhoff H, Klapuri A. Automatic music transcription: challenges and future directions. J Intell Inf Syst. Springer US; 2013;41: 407–434.](http://paperpile.com/b/yFWBg3/oDWU)

3. [Ming H, Huang D, Xie L, Li H. Learning optimal features for music transcription. 2014 IEEE China Summit International Conference on Signal and Information Processing (ChinaSIP). 2014. pp. 105–109.](http://paperpile.com/b/yFWBg3/bOxo)

4. [Lee CT, Yang YH, Chen HH. Multipitch Estimation of Piano Music by Exemplar-Based Sparse Representation. IEEE Trans Multimedia. 2012;14: 608–618.](http://paperpile.com/b/yFWBg3/TQEf)

5. [Anan Y, Hatano K, Bannai H, Takeda M, Satoh K. Polyphonic Music Classification on Symbolic Data Using Dissimilarity Functions. ISMIR. 2012. pp. 229–234.](http://paperpile.com/b/yFWBg3/Ihbh)

6. [Sigtia S, Benetos E, Dixon S. An End-to-end Neural Network for Polyphonic Piano Music Transcription. IEEE/ACM Trans Audio, Speech and Lang Proc. Piscataway, NJ, USA: IEEE Press; 2016;24: 927–939.](http://paperpile.com/b/yFWBg3/fcD9)

7. [Poliner GE, Ellis DPW. A Classification Approach to Melody Transcription. ISMIR. 2005. p. 6th.](http://paperpile.com/b/yFWBg3/mE8Q)

8. [Holmes G, Donkin A, Witten IH. WEKA: a machine learning workbench. Intelligent Information Systems,1994 Proceedings of the 1994 Second Australian and New Zealand Conference on. 1994. pp. 357–361.](http://paperpile.com/b/yFWBg3/DFXs)

9. [Read J, Reutemann P. MEKA: a multi-label extension to WEKA. URL http://meka sourceforge net. 2012; Available:](http://paperpile.com/b/yFWBg3/JVmH) <http://meka.sourceforge.net/>

10. [Herrera F, Charte F, Rivera AJ, del Jesus MJ. Multilabel Classification. Multilabel Classification. Springer International Publishing; 2016. pp. 17–31.](http://paperpile.com/b/yFWBg3/s1kY)

11. [Aha DW, Kibler D, Albert MK. Instance-based learning algorithms. Mach Learn. Kluwer Academic Publishers; 1991;6: 37–66.](http://paperpile.com/b/yFWBg3/hdLU)

12. [Ross Quinlan J. C4.5: Programs for Machine Learning. Elsevier; 2014.](http://paperpile.com/b/yFWBg3/uYxo)

13. [Platt JC. 12 fast training of support vector machines using sequential minimal optimization. Advances in kernel methods. 1999; 185–208.](http://paperpile.com/b/yFWBg3/Dyka)

14. [Dugan TM, Mukhopadhyay S, Carroll A, Downs S. Machine Learning Techniques for Prediction of Early Childhood Obesity. Appl Clin Inform. 2015;6: 506–520.](http://paperpile.com/b/yFWBg3/rUkm)

15. [Emiya V, Badeau R, David B. Multipitch Estimation of Piano Sounds Using a New Probabilistic Spectral Smoothness Principle. IEEE Trans Audio Speech Lang Processing. 2010;18: 1643–1654.](http://paperpile.com/b/yFWBg3/FQ8J)

16. [Bundy A, Wallen L. Zero-Crossings. In: Bundy A, Wallen L, editors. Catalogue of Artificial Intelligence Tools. Springer Berlin Heidelberg; 1984. pp. 135–136.](http://paperpile.com/b/yFWBg3/xhps)

17. [Spectral Centroid. Van Nostrand’s Scientific Encyclopedia. John Wiley & Sons, Inc.; 2005.](http://paperpile.com/b/yFWBg3/T2wW)

18. [Wikipedia contributors. Roll-off. In: Wikipedia, The Free Encyclopedia [Internet]. 18 Dec 2016 [cited 20 Mar 2017]. Available:](http://paperpile.com/b/yFWBg3/PtsU) <https://en.wikipedia.org/w/index.php?title=Roll-off&oldid=755518739>

19. [Wikipedia contributors. Flux. In: Wikipedia, The Free Encyclopedia [Internet]. 27 Feb 2017 [cited 20 Mar 2017]. Available:](http://paperpile.com/b/yFWBg3/7VpY) <https://en.wikipedia.org/w/index.php?title=Flux&oldid=767714264>

20. [Wikipedia contributors. Mel-frequency cepstrum. In: Wikipedia, The Free Encyclopedia [Internet]. 31 Oct 2016 [cited 20 Mar 2017]. Available:](http://paperpile.com/b/yFWBg3/9q36) <https://en.wikipedia.org/w/index.php?title=Mel-frequency_cepstrum&oldid=747054068>

21. [Tzanetakis G, Cook P. Marsyas: A framework for audio analysis. Organised sound. 2000;4: 169–175.](http://paperpile.com/b/yFWBg3/NdRb)

22. [Filter-Based Feature Selection [Internet]. [cited 20 Mar 2017]. Available:](http://paperpile.com/b/yFWBg3/ZqrX) <https://msdn.microsoft.com/en-us/library/azure/dn905854.aspx>

23. [Brownlee J. How to Perform Feature Selection With Machine Learning Data in Weka - Machine Learning Mastery. In: Machine Learning Mastery [Internet]. 13 Jul 2016 [cited 20 Mar 2017]. Available:](http://paperpile.com/b/yFWBg3/SviS) <http://machinelearningmastery.com/perform-feature-selection-machine-learning-data-weka/>