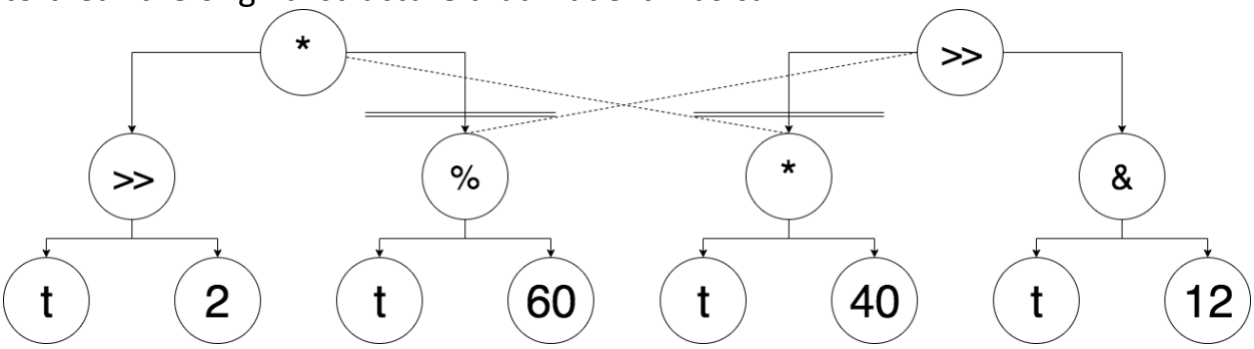
Progress Report 5: Crossover Strategy, Tree Feature Extraction & Support Vector Regression

This week I focused on finding a crossover strategy for our tree chromosomes, this allows us to combine trees from different clusters and mimic nature by “mating” them. I also spent some time extracting tree-based features in order to observe the relationship between or ByteBeat tracks and their underlying structure. Lastly, I attempted to used a Support Vector Regression to get a better prediction of ratings.

Starting with the crossover strategy, my approach is to select two trees from different sound-based clusters, and have them swap sub-trees. Sub-trees of lower heights have a higher probability of swapping while higher ones swap quite rarely. Additionally, the sub-trees are swapped at equal heights as shown in the figure below. The probability of crossover in out Genetic algorithm should be low, considering this is a significant mutation to the original tree and has the potential to break the original structure that made it musical.

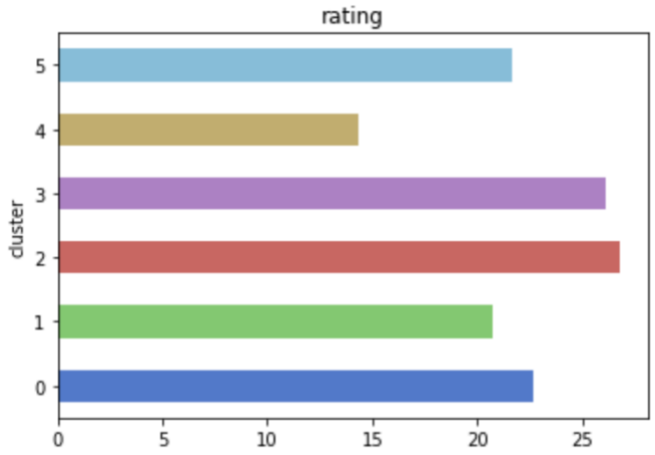


Once the trees could freely mutate and mate, I wanted to analyze these them and how they relate to the music they generate. For this, I extracted the following tree-based features and ran a k-mean clustering algorithm based on them.

- Height: height of the tree.
- Leaves: number of leaves.
- t-count: number of ”t” leaves.
- Operators: number of operators.
- Count of each operator: 9 columns counting each operator.
- Powers of 2: number of leaf nodes that are powers of 2.
- Average Operand: Average value of operands

The clustering revealed some interesting patterns. The highest rated group contained many bit-shift and bit-wise “&” operators, which is in-line with Viznut’s

research on the Sierpinski Harmonies. The lowest rated group had a surprisingly high-amount of powers of two, which runs contrary to Viznut's hypothesis about these numbers.



	>>	*	&	^	%	|	-	+	/	2_powers
0	1.394366	0.926056	1.183099	0.647887	0.306338	0.461268	0.264085	0.281690	0.059859	0.669014
1	1.742331	1.153374	3.509202	0.828221	0.257669	0.644172	0.325153	0.282209	0.018405	0.822086
2	3.867470	1.271084	2.030120	0.668675	0.240964	0.759036	0.240964	0.259036	0.030120	1.162651
3	2.035088	1.175439	1.298246	0.842105	0.482456	2.701754	0.307018	0.333333	0.052632	1.236842
4	2.223214	1.401786	2.107143	1.625000	0.473214	1.000000	0.571429	0.491071	0.017857	3.312500
5	1.813665	3.316770	1.732919	0.919255	0.242236	0.708075	0.285714	0.273292	0.068323	1.298137

The last bit of progress I made this week was trying out different models to try to predict a song's rating based on its tree and spectral features. So far I have had little success in generating accurate predictions, but the best model so far has been the support-vector regression. I will meet with my Machine Learning TA to try to solve this issue.