# Overview of Fall 2019 LE303 Independent Study

Mel Mark

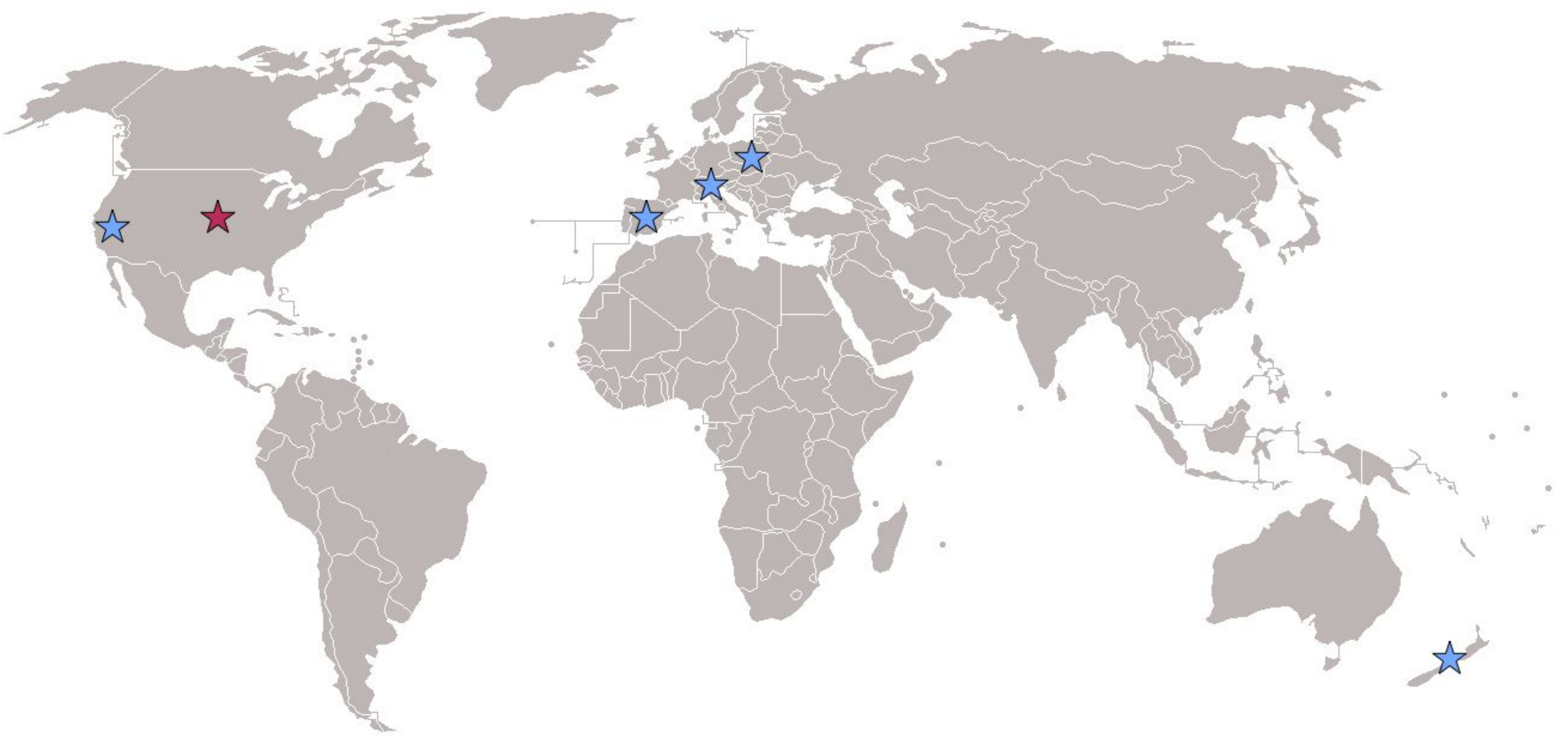
Park University

Fall 2019

**Table of Contents**

**INTRODUCTION**

In the fall semester of 2019 I started the implementation phase of my honors research project at Park University with advisor Professor Joe Wang. As a computer science major who plays the bass in their free time, I was very interested in the field of music information retrieval. Music information retrieval simply involves extracting data from music and manipulating or analyzing it in some fashion. Specifically, I was interested in assigning human emotions to music from all over the world with the help of machine learning. This was not a small task, and required lots of research and collaboration with researchers all over the world, including Switzerland, California, Poland, Spain, and New Zealand. This semester could be mainly divided into five sections respectively: gathering of information and literature, processing and selecting appropriate methodologies, metadata extraction, conversion, and analysis.



*Blue stars denote locations of some researchers I collaborated with, the maroon star is where Park University is located in Parkville, Missouri.*

## GATHERING OF INFORMATION AND LITERATURE

As any researcher knows, the first step to any problem is to gather relevant information and scholarly literature to aid in one's journey. This started with me browsing music information retrieval literature very broadly, looking carefully at datasets, intentions, and paths different researchers had taken. Then, as I began to narrow my search, I started to focus only on machine learning papers that specifically had to do with music and emotions. This proved to be difficult as it is a relatively new field that does not have a lot of substance in its libraries.

As I dove further and further into the literature, it became apparent that I was missing an understanding vital to this field: machine learning. As such, I began familiarizing myself with concepts and terms concerning machine learning by taking a course on Coursera and watching YouTube tutorials. This did help me understand some of the fundamental theories and practices behind machine learning, but it was difficult for me to wrap my head around a lot of it without tinkering with it myself. As a result, I began sifting through all the different methods and approaches various researchers before me had taken, searching for the one that would best work for my problem and that would allow me to put my new knowledge to the test. Eventually, I settled on the New Zealand tool Weka. Weka reads ARFF files, a file type only used for Weka, and is able to preprocess data, classify data, cluster data, associate data, and visualize data. I specifically liked this tool because it had a GUI as opposed to just a command line tool. It is also widely used across the field of machine learning and has reputable and accredited documentation and research.

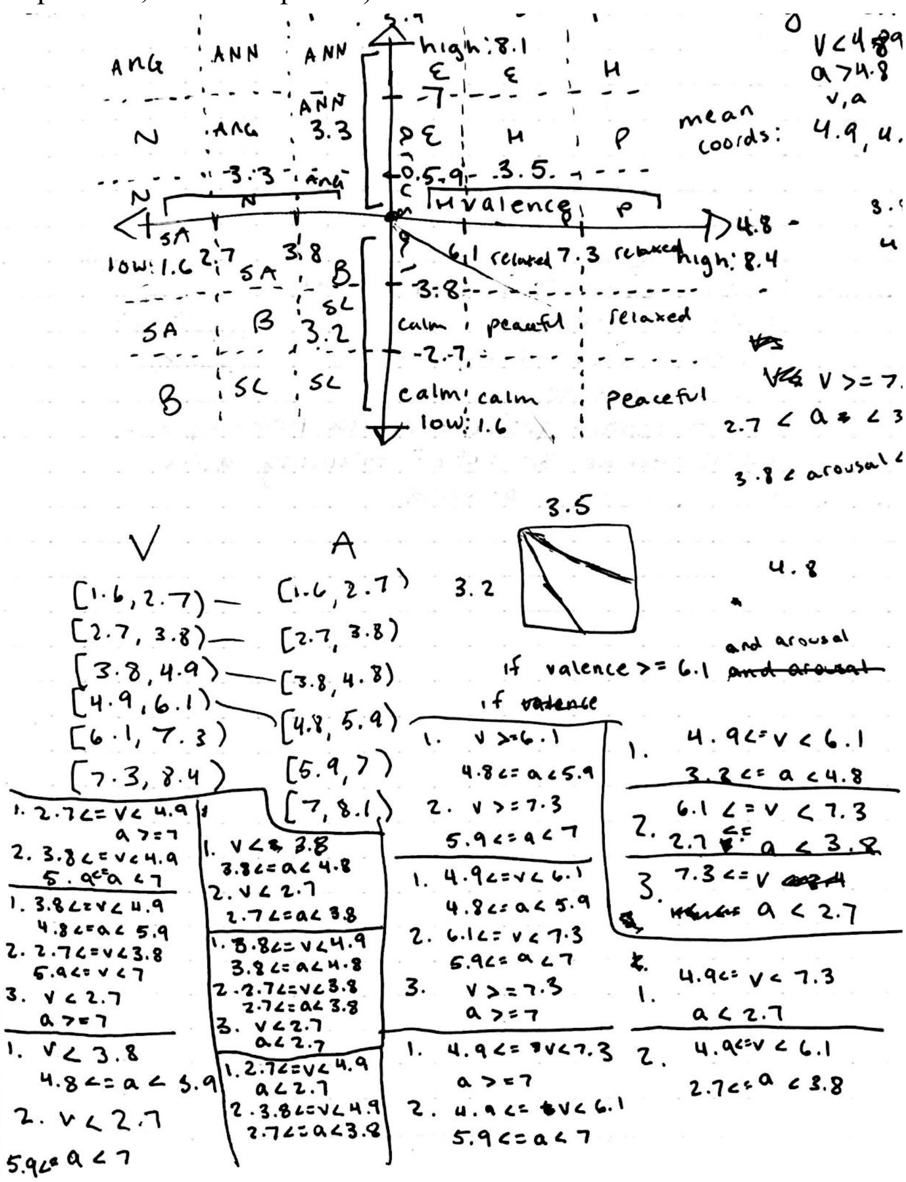
## PROCESSING AND SELECTING APPROPRIATE METHODOLOGIES

While the literature regarding emotive machine learning with music information retrieval is not abundant, there is a great variety in the approaches that are reported. Because of this, it took a lot of reading and searching to find which methods supposedly had the lowest margin of error and best datasets. Between the different regression algorithms implemented to the myriad of differences among datasets, it was overwhelming to choose one way or another when there is more than one way to bake a cake. Two papers that heavily assisted me in picking methods are "Detecting Emotion in Music" by Li and Ogihara and "Music Emotion Recognition: A State of the Art Review" from Ithaca College and Drexel University. Both papers detailed methods researchers experimented with before and included the results and accuracy of each.

Another field I was underprepared in was psychology. I have never taken a class in psychology and was not aware of the various models used to plot mood. As such, I had to read up on the many methods of enumerating mood and emotions on both two dimensional and three dimensional graphs. Eventually, I settled on a model known as Thayer's Model to two dimensionally plot emotions into four categories (Thayer, 2000). The two axes represent valence and arousal, two measurements that ultimately determine which quadrant and which emotion. I then further divided the four quadrants into nine subsections each, with three emotions representing three subsections in each quadrant. While there are 36 subsections in 4 quadrants, there are only 12 moods represented (pleased, happy, excited, annoyed, angry, nervous, sad, bored, sleepy, relaxed, peaceful, calm) because 36 divided by the 3 subsections occupied by each mood equals 12.

Now that I had a model to plot emotions, I then had to find an appropriate dataset that had a good number of songs of the same length and annotated with valence and arousal values. This was vital to build my training set of data for the machine learning algorithms. This was a tall order to fill as this field is already so niche, there are not many datasets created, let alone public. After lots of digging, I eventually came across the DEAM: MediaEval Database for Emotional Analysis in Music (Soleymani, 2016). It was created by researchers in Switzerland and contained lots of information that was not relevant to my research. However, it did contain valence and arousal values for 1,745 songs and their 45 second audio files. It was not clear originally how they extracted the arousal and valence, so I attempted to contact them. The issue was that there was only one method of contact left on the paper, an email. I emailed the address listed but was sent back an error from the address's postmaster stating I was not whitelisted on their servers and could not send it emails. This was annoying because this was the only means of finding out this very important detail. I went to the Swiss university's website in which they conducted the research, and tried to find a technology office email so I could be either whitelisted or put in contact with this researcher. The entire website was in French, which I do not speak, so I had to translate it while navigating and eventually found an email to some generic office. I emailed them in both English and French but never got a response. In the meantime, I searched the Internet for the researcher's name and managed to find his personal website where he stated he was now a professor in California. I emailed him and he immediately got back to me stating that they had not extracted arousal and valence directly like I had thought, instead, they used turkers to manually extract the values. Of course, this is not ideal as it is subjective and not replicable. However, it was better than nothing.

The next step was plotting all the songs in the dataset on the two dimensional model by Thayer. To do this, I had to first find the origin of the graph, which was not (0, 0) as one might expect. Instead, I found the mean of each and made that the origin (4.9, 4.8). Then I took the minimum and maximum values of the arousal mean column and the valence mean column and subtracted it to find the differences. Once I had the differences, I divided them each by 6 (three subsections high and wide in each quadrant, times 2 equals 6) to find the increment. I then drew the boundaries for each subsection and wrote a program in Python3.7 to automatically take in the valence and arousal mean columns and calculate and write each corresponding mood in a new CSV. Now that I knew they manually extracted the valence and arousal and I had annotated my dataset with moods according to Thayer's model, my next objective was to automatically extract the arousal and valence, which is much easier said than done.




*Hand-mapping each quadrant, section, and subsection for DEAM.*

## METADATA EXTRACTION

At the beginning of trying to extract music data, I naively thought that arousal and valence could be directly extracted from songs. Instead, what I learned is that arousal and valence are psychological terms that mean nothing to music. In order to calculate arousal and valence, calculations would have to be made from other aspects of music data. I, of course, had no idea how to do this so I began reviewing scholarly literature once again. Eventually I came across a gem, a Polish researcher named Jacek Grekow published a paper detailing differences in music metadata extractors and the pros and cons of each (Grekow, 2018). The two extractors he focused on were Marsyas and Essentia. Marsyas is a Greek command line tool written in C++ and Essentia is a Spanish command line tool written also in C++. Both extracted metadata from music, but Grekow noted that Essentia is more detailed and provides more features, which would ultimately lead to greater accuracy in detection of emotions. Because of this, I chose to use Essentia over Marsyas (in addition to Marsyas not wanting to compile correctly on my computer).

Essentia has three high-level categories it puts its features in: low-level, rhythm, and tonal. It takes the audio file and quickly and efficiently outputs the data in a JSON or YAML file. In addition, Essentia calculates thorough statistics for each feature extracted, including mean, geometric mean, power mean, median of an array, all its moments up to the fifth order, energy, and the root mean square (RMS). The only issue with Essentia is that it outputted the data in only JSON or YAML, with no option for ARFF files, which are the only kind Weka can read. As a result, I had to figure out how to convert the files, which would prove a challenge.

```
▲ ~/projects/essentia essentia_streaming_extractor_music test/audio/Ellipses.mp3 ellipses.json
[   INFO   ] MusicExtractor: Read metadata
[   INFO   ] MusicExtractor: Compute md5 audio hash, codec, length, and EBU 128 loudness
[   INFO   ] MusicExtractor: Replay gain
[   INFO   ] MusicExtractor: Compute audio features
[   INFO   ] MusicExtractor: Compute aggregation
[   INFO   ] All done
Writing results to file ellipses.json
```

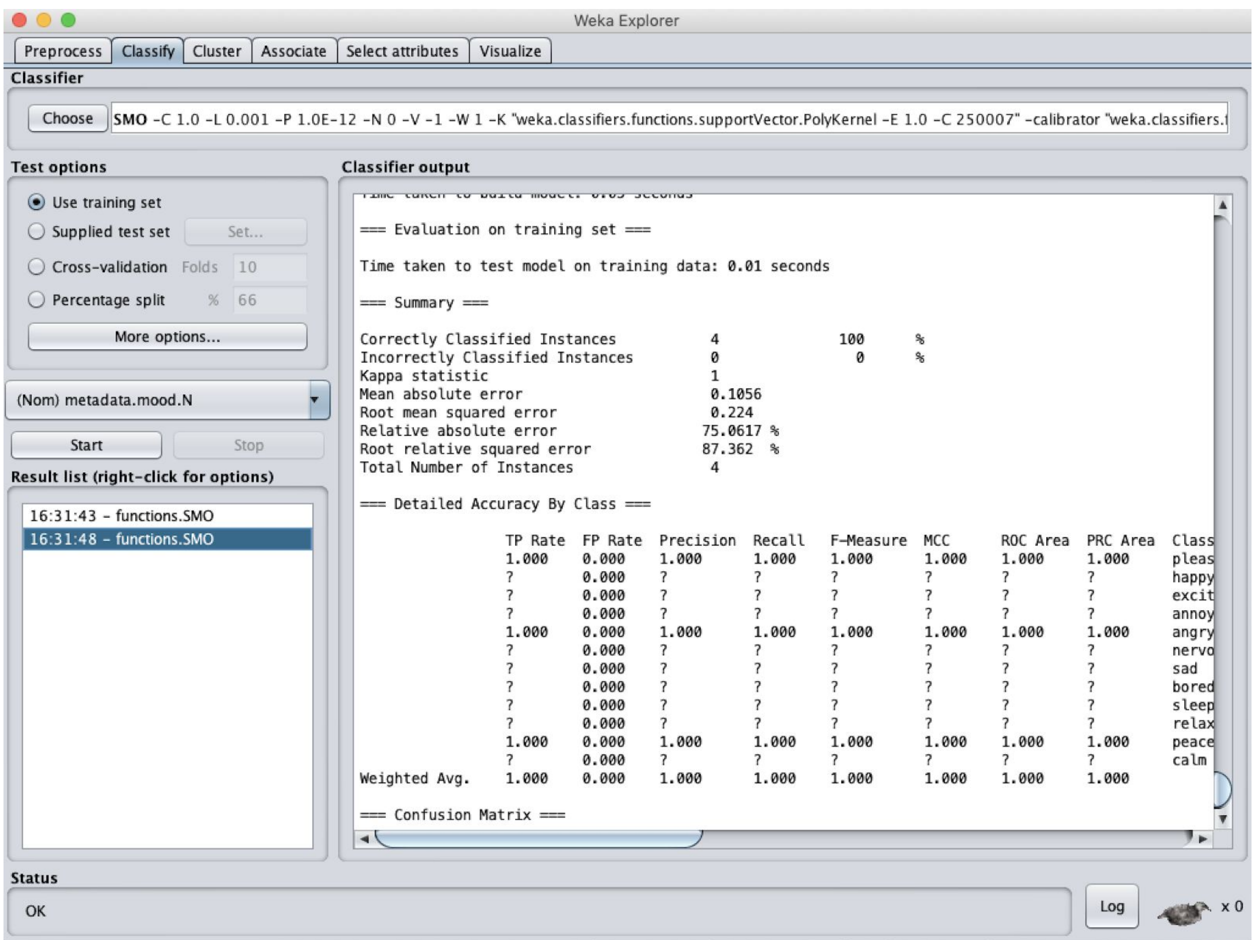
*Essentia reading and computing data within seconds.*

## CONVERSION

Because ARFF files are only for Weka, there are not any converters publicly available. Weka does have a built in 'converter' tool but it only works with very specific file types with very specific data. After conversing with some researchers who developed Weka, it was clear that I would need to build my own tool. I had never built a conversion tool before, but luckily both the input and the output types were not too complicated. ARFF files basically have a set of labels or 'attributes' that is paired to a data value, dependant on position within a comma separated list. I had a wide variety of languages to choose from to write this program but I ultimately chose Python3.7 for its simplicity and robust documentation. In addition, I had some experience with Python and was comfortable writing in it.

The first challenge was extracting just the numbers of the JSON file Essentia outputted. I was not interested in the labels, just the numerical values and two non-numerical values (key signature and major/minor key). This was made even more difficult by the fact JSON files from Essentia contain multiple levels, and it is not possible to grab just the values for this to work. There was another challenge within this: some labels contained numbers, such as 'dmean2' or 'melbands128', so I had to somehow figure out how to differentiate numbers in labels and actual numerical values. Also, a lot of numbers contained e+ or e- to denote scientific notation, so in addition to some labels containing numbers, some numbers contained letters. This was circumvented by a few conditional statements checking for those exceptions listed above. I also used some temporary lists and strings as placeholders to manipulate data before assigning the finalized data to the output variables.

Now that I was able to just extract the numbers, I had to write the actual converter that took the JSON data and outputted it to an ARFF file. In Essentia, all labels are the same for each song except one. There is a label called 'beats_position' that can vary with how many times it appears. Sometimes there are over 150 beats_position numbers, but sometimes there are under 75. I was able to write a loop that detected how many beats_position labels to generate in the ARFF files that solved that problem. Finally, I was able to open up successfully converted JSON → ARFF files in Weka and analyze them. Now I just needed to clean up the dataset acquired previously.



*I was finally able to successfully open my converted JSON → ARFF files and analyze the results.*

## ANALYSIS

Before I began analyzing and classifying data, I had to first ensure that my dataset was of the highest quality, so that my predictions would be as accurate as possible. Some of the moods the program assigned to songs were a little questionable. To avoid outliers or inaccurate data, I am in the process of going through and listening to each 1,745 45 second audio clip. I am marking each one I find accurate and am taking those songs and creating an improved dataset. Once this is done, I will be able to more accurately predict emotions in songs in Weka.

| song_id | valence_mean | arousal_mean | mood | accurate? y/n |
|---|---|---|---|---|
| 2 | 3.1 | 3 | bored | y |
| 3 | 3.5 | 3.3 | bored | y |
| 4 | 5.7 | 5.5 | happy | y |
| 5 | 4.4 | 5.3 | angry | n |
| 7 | 5.8 | 6.4 | excited | y |
| 8 | 3.2 | 4.8 | nervous | n |
| 10 | 4 | 4.7 | bored | y |
| 12 | 5.5 | 5.8 | happy | n |
| 13 | 3.2 | 4 | sad | n |
| 17 | 4.4 | 6 | annoyed | n |
| 18 | 4.8 | 3.9 | bored | y |
| 19 | 5.9 | 4.3 | peaceful | n |
| 20 | 5.4 | 6.5 | excited | n |
| 21 | 6.6 | 6.3 | happy | n |
| 22 | 4 | 4.8 | angry | n |
| 24 | 5.3 | 3.9 | peaceful | n |
| 25 | 7.9 | 4.7 | relaxed | n |
| 31 | 4.7 | 3.8 | bored | n |
| 32 | 4.2 | 3.3 | sleepy | y |
| 35 | 4.8 | 6.5 | annoyed | y |
| 37 | 6.6 | 5.3 | pleased | n |
| 39 | 7.3 | 5.4 | pleased | n |
| 40 | 4.6 | 4.8 | angry | n |

| | number | precentage |
|---|---|---|
| favor | 15 | 35.71% |
| disagreement | 27 | 64.29% |
| **total** | 42 | 100.00% |

| ranked: | | |
|---|---|---|
| **mood** | **# of instances** | **contribution** |
| happy | 404 | 23.15% |
| bored | 243 | 13.93% |
| sleepy | 222 | 12.72% |
| excited | 189 | 10.83% |
| angry | 152 | 8.71% |
| peaceful | 125 | 7.16% |
| sad | 121 | 6.93% |
| pleased | 94 | 5.39% |
| calm | 57 | 3.27% |
| annoyed | 57 | 3.27% |
| nervous | 44 | 2.52% |
| relaxed | 25 | 1.43% |

*I am currently going through each song individually and analyzing accuracy; this will take some time.*

## LOOKING FORWARD

Ideally, once my dataset is completely up to my standards of quality, I would then be able to analyze and predict the mood of any song. I also want to play with different classifiers and regression algorithms to ensure the most accurate predictions. There are many possible paths for the future of this project, and I am not sure where I will end up. It will most likely depend on the amount of time I have left at Park University, as I want to submit my paper to journals for publication. I am also giving a presentation at the Honors Symposium at Park University in the spring of 2020. Hopefully, all of my research will one day help others and contribute positively to the world.

References

Grekow, J. (2018). Audio features dedicated to the detection and tracking of arousal and valence in musical compositions. Journal of Information and Telecommunication, 2(3), 322-333.

Kim, Y. E., Schmidt, E. M., Migneco, R., Morton, B. G., Richardson, P., Scott, J., ... & Turnbull, D. (2010, August). Music emotion recognition: A state of the art review. In Proc. ISMIR (Vol. 86, pp. 937-952).

Li, T., & Ogihara, M. (2003). Detecting emotion in music.

Soleymani, M., Aljanaki, A., & Yang, Y. H. (2016). DEAM: MediaEval Database for Emotional Analysis in Music.

Thayer, J. F., & Lane, R. D. (2000). A model of neurovisceral integration in emotion regulation and dysregulation. Journal of affective disorders, 61(3), 201-216.