1. How to run the code, how to use the system, functionalities of each program file:

- To run the GUI: run the .exe file
- For the ML model, we've included a jupyter notebook file (FINAL MODEL.ipynb) to easily visualize - the notebook should automatically install the necessary libraries involved, so you just need to make sure you can run it.
    - https://jupyter.org/install
- NOTE: Make sure there's an export.csv file after exporting the data using the GUI application
- Simply open the ipynb file after opening jupyter notebook and run through all of the cells, and you should see by the end a visualization of our process and our final result.
- The notebook however is mostly a formality, as we've documented the process and results within this report, so no need to actually run it unless you want to verify our code.

2. Brief introduction of libraries/tools/techniques you used in your development:

- C# Feature extraction GUI
    - NAudio - used to process and extract amplitude.
    - AForge.Math - used to process and extract frequencies and magnitude
    - Newtonsoft.Json - used in exporting features for ML code to read
    - Winforms - used as the GUI framework
- Python ML model
    - Pytorch - ML library used to train our model
    - Scikit-Learn - Used for a little bit of cross validation in the ML process
    - Pandas - Used to help manipulate dataset easily for eventual Pytorch manipulation
    - Matplotlib - data visualization library to help show results

3. List of features and their values for the audio files and show what files are used as training data, and what is tested:

The features being used are:

- Zero Crossing Rate
- Average Energy
- Bandwidth
- Spectral Centroid (over frequency)

	A	B	C	D	E	F
1	file_name	avg_energy	zero_cross_range	bandwidth	spectral_centroid	label
2	Audio_Files/Music\mu1.wav	0.0135809619249186	0.245164915786023	7995.1171875	3442.5370025738	True
3	Audio_Files/Music\mu10.wav	0.00612919615867232	0.124285133910435	7995.1171875	1599.2067093153	True
4	Audio_Files/Music\mu11.wav	0.00639878122730169	0.0616736649860294	7995.1171875	1003.0774980364	True
5	Audio_Files/Music\mu12.wav	0.0103937993395375	0.235691182674729	7995.1171875	2206.0999106274	True
6	Audio_Files/Music\mu13.wav	0.0168704782627043	0.0681740054376699	7995.1171875	1352.0875328899	True
7	Audio_Files/Music\mu14.wav	0.036661446373754	0.0399387480858777	7995.1171875	1527.2554496112	True
8	Audio_Files/Music\mu15.wav	0.00200785444512439	0.110986109157956	7995.1171875	1590.2979495598	True
9	Audio_Files/Music\mu16.wav	0.0172653324184503	0.138033406771981	7995.1171875	2152.9322624265	True
10	Audio_Files/Music\mu17.wav	0.076718759502648	0.0987202925045704	7995.1171875	2432.6430244709	True
11	Audio_Files/Music\mu18.wav	0.00394412702108093	0.135580243441304	7995.1171875	2347.4104063794	True
12	Audio_Files/Music\mu19.wav	0.00181142434339563	0.168987015422116	7995.1171875	2722.9324405605	True
13	Audio_Files/Music\mu2.wav	0.00968006876183337	0.0965498921841308	7995.1171875	1896.1121084931	True
14	Audio_Files/Music\mu20.wav	0.000672354311172967	0.121814403350052	7995.1171875	2589.6867566477	True
15	Audio_Files/Music\mu3.wav	0.00785261910757346	0.0928935279227476	7995.1171875	1766.3315453915	True
16	Audio_Files/Music\mu4.wav	0.00693985686341427	0.114206693959186	7995.1171875	1786.1386966061	True
17	Audio_Files/Music\mu5.wav	0.000804243735881755	0.100876576196503	7995.1171875	2223.8298038414	True
18	Audio_Files/Music\mu6.wav	0.0110080037050062	0.0568611519109972	7995.1171875	1716.1457058256	True
19	Audio_Files/Music\mu7.wav	0.000471108456474807	0.128832151004719	7995.1171875	1805.4062958511	True
20	Audio_Files/Music\mu8.wav	0.000866658837086387	0.206421803537143	7995.1171875	2738.7038745398	True
21	Audio_Files/Music\mu9.wav	0.00222460697433824	0.0922841577821832	7995.1171875	2007.7010581006	True
22	Audio_Files/speech\sp1.wav	0.000528156630811282	0.155377427772309	7998.046875	2680.061302532	False
23	Audio_Files/speech\sp10.wav	0.139020341253432	0.143142861607213	7998.046875	1028.3776329841	False
24	Audio_Files/speech\sp11.wav	0.000554115027785883	0.135119298739043	7998.046875	2580.830630866	False
25	Audio_Files/speech\sp12.wav	0.00710451955738245	0.17276832450507	7998.046875	1926.3817806433	False
26	Audio_Files/speech\sp13.wav	0.177054650301056	0.0956889951405491	7998.046875	1033.7763464033	False
27	Audio_Files/speech\sp14.wav	0.00660737182948328	0.129419477701637	7998.046875	1263.8480714743	False
28	Audio_Files/speech\sp15.wav	0.00381537588003266	0.144033500523446	7998.046875	1453.5499836008	False
29	Audio_Files/speech\sp16.wav	0.0356854844021291	0.203956311817372	7998.046875	3152.2090582275	False
30	Audio_Files/speech\sp17.wav	0.0618837064609834	0.204206315723683	7998.046875	2824.5904993058	False
31	Audio_Files/speech\sp18.wav	0.0035985503325521	0.0605341616845917	7998.046875	1409.8569825466	False
32	Audio_Files/speech\sp19.wav	0.00341002830628912	0.11585470552425	7998.046875	2947.6891585307	False
33	Audio_Files/speech\sp2.wav	0.000454288347245893	0.121892529570775	7998.046875	1686.2693724242	False
34	Audio_Files/speech\sp20.wav	0.0158364342521211	0.0796091347579726	7998.046875	880.4402806769	False
35	Audio_Files/speech\sp3.wav	0.000502355925113079	0.131423928498883	7998.046875	1630.5747207121	False
36	Audio_Files/speech\sp4.wav	0.106326516403526	0.141939717808091	7998.046875	733.8801451315	False
37	Audio_Files/speech\sp5.wav	0.00071942570572719	0.183346614790856	7998.046875	2688.6141608646	False
38	Audio_Files/speech\sp6.wav	0.000317843364192648	0.0578961334935922	7998.046875	902.0030332841	False
39	Audio_Files/speech\sp7.wav	0.00830729107945808	0.081188768574509	7998.046875	1396.4846046661	False
40	Audio_Files/speech\sp8.wav	0.0164563363175296	0.0738491797137193	7998.046875	713.5522240477	False
41	Audio_Files/speech\sp9.wav	0.150312658431212	0.103157861841591	7998.046875	788.3757411488	False

Example export data for all features

The audio files used to collect features for training data are in the audio.zip in the Assignment 4 folder of the CSS 484 Canvas course files.

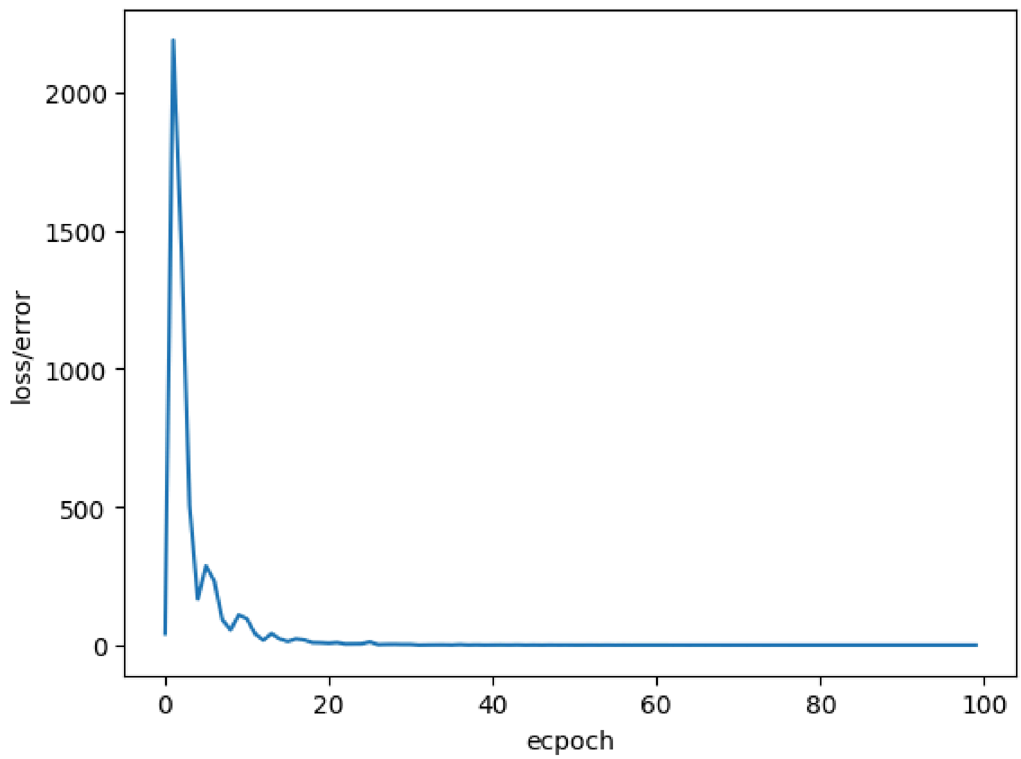
We are testing the features mentioned above with our ML model - our pytorch model processes the above 4 features using pandas into tensors.

The data is processed using linear regression under an adams optimizer, with the layers structured through an input layer > h1 (hidden layer) > h2 (hidden layer) > final output layer (binary output, either 0 or 1 aka yes this is a music file or no it is not)

About 2/3rds of the data is relegated for training, while the other is relegated for the testing phase.

Training phase graph portraying error rate over epoch:

```
]:  Text(0.5, 0, 'ecpoch')
```



Comparison between model output and ground truth label for TESTING phase (as discussed in 2.4)
The training dataset:
Precision: 1/(1 + 1)=0.5
Recall: 1/(1+3)=0.25

```
1.) Audio_Files/speech\sp14.wav tensor([43.6581, 39.9064])     0        0
2.) Audio_Files/speech\sp11.wav tensor([42.2627, 42.9876])     0        1
3.) Audio_Files/speech\sp13.wav tensor([45.3080, 38.8191])     0        0
4.) Audio_Files/speech\sp18.wav tensor([42.6873, 40.5636])     0        0
5.) Audio_Files/Music\mu11.wav tensor([45.5220, 38.6493])     1         0
6.) Audio_Files/Music\mu6.wav tensor([41.0262, 41.8431])     1          1
7.) Audio_Files/Music\mu13.wav tensor([43.0525, 40.2911])     1         0
8.) Audio_Files/Music\mu14.wav tensor([41.8880, 41.0795])     1         0
4
```

0 = no, not music 1 = yes, is music
Left number column is actual answer, right column is predicted
(filename), model output (tensor), ground truth label (as 0 or 1)

4. Display of the precision and recall value of our model on the testing data, as well as a final retest over the ENTIRE dataset

Precision: 14/(14+6) = 0.7

Recall: 14/(14+ 6) = 0.7

Ground Truth	**Model Output**
Yes	No
Yes	No
Yes	No
Yes	Yes
Yes	No
Yes	No
Yes	No
Yes	Yes
Yes	Yes
Yes	Yes
Yes	Yes
Yes	Yes
Yes	Yes
Yes	Yes
Yes	Yes
Yes	Yes
Yes	Yes
Yes	Yes
Yes	Yes
Yes	Yes
No	Yes
No	No

No	Yes
No	Yes
No	No
No	No
No	No
No	No
No	No
No	No
No	No
No	Yes
No	No
No	Yes
No	No
No	Yes
No	No
No	No
No	No
No	No

```
1.) Audio_Files/Music\mu1.wav tensor([35.2813, 34.6810])      1          0
2.) Audio_Files/Music\mu10.wav tensor([41.4416, 41.3970])      1         0
3.) Audio_Files/Music\mu11.wav tensor([45.5220, 38.6493])      1         0
4.) Audio_Files/Music\mu12.wav tensor([41.2506, 43.1198])      1         1
5.) Audio_Files/Music\mu13.wav tensor([43.0525, 40.2911])      1         0
6.) Audio_Files/Music\mu14.wav tensor([41.8880, 41.0795])      1         0
7.) Audio_Files/Music\mu15.wav tensor([41.4739, 41.3626])      1         0
8.) Audio_Files/Music\mu16.wav tensor([41.2058, 43.0628])      1         1
9.) Audio_Files/Music\mu17.wav tensor([41.8418, 43.0397])      1         1
10.) Audio_Files/Music\mu18.wav tensor([41.6060, 43.0776])      1        1
11.) Audio_Files/Music\mu19.wav tensor([42.6233, 42.8862])      1        1
12.) Audio_Files/Music\mu2.wav tensor([40.8693, 42.4267])      1         1
13.) Audio_Files/Music\mu20.wav tensor([42.2743, 42.9675])      1        1
14.) Audio_Files/Music\mu3.wav tensor([40.8477, 42.0345])      1         1
15.) Audio_Files/Music\mu4.wav tensor([40.7854, 42.1144])      1         1
16.) Audio_Files/Music\mu5.wav tensor([41.2718, 43.1220])      1         1
17.) Audio_Files/Music\mu6.wav tensor([41.0262, 41.8431])      1         1
18.) Audio_Files/Music\mu7.wav tensor([40.7500, 42.2008])      1         1
19.) Audio_Files/Music\mu8.wav tensor([42.5362, 42.7341])      1         1
20.) Audio_Files/Music\mu9.wav tensor([41.0162, 42.7044])      1         1
21.) Audio_Files/speech\sp1.wav tensor([42.5363, 42.9423])      0        1
22.) Audio_Files/speech\sp10.wav tensor([45.3490, 38.7913])      0       0
23.) Audio_Files/speech\sp11.wav tensor([42.2627, 42.9876])      0       1
24.) Audio_Files/speech\sp12.wav tensor([40.9231, 42.5157])      0       1
25.) Audio_Files/speech\sp13.wav tensor([45.3080, 38.8191])      0       0
26.) Audio_Files/speech\sp14.wav tensor([43.6581, 39.9064])      0       0
27.) Audio_Files/speech\sp15.wav tensor([42.3967, 40.7601])      0       0
28.) Audio_Files/speech\sp16.wav tensor([38.7558, 38.1770])      0       0
29.) Audio_Files/speech\sp17.wav tensor([42.0842, 41.9331])      0       0
30.) Audio_Files/speech\sp18.wav tensor([42.6873, 40.5636])      0       0
31.) Audio_Files/speech\sp19.wav tensor([41.2264, 40.6686])      0       0
32.) Audio_Files/speech\sp2.wav tensor([41.1493, 41.7419])      0        1
33.) Audio_Files/speech\sp20.wav tensor([46.7540, 39.5357])      0       0
34.) Audio_Files/speech\sp3.wav tensor([41.3473, 41.5295])      0        1
35.) Audio_Files/speech\sp4.wav tensor([48.2714, 40.8600])      0        0
36.) Audio_Files/speech\sp5.wav tensor([42.5598, 42.9382])      0        1
37.) Audio_Files/speech\sp6.wav tensor([46.5314, 39.3311])      0        0
38.) Audio_Files/speech\sp7.wav tensor([42.7762, 40.5034])      0        0
39.) Audio_Files/speech\sp8.wav tensor([48.4755, 41.0392])      0        0
40.) Audio_Files/speech\sp9.wav tensor([47.7078, 40.3716])      0        0
Correct: 28
```

(real proof of our data as run with our pytorch model)

As you can see based on a run with every file, we have about 70% accuracy/success rate in guessing correctly. This is about as fairly well run as we could get it with the simplistic conditions of our model, considering the relatively somewhat low amount of sample data to train on.