Project RedSiren

Classification of Police Scanner Audio for Emergency Responders

Blake Wallace, Lance Carroll, Neal Mannahan,
Rodolfo Flores Méndez, and Sardorkhon Tursunov

Contents

- Define Data Science Problem
- Gather and Process Data
- Exploratory Data Analysis
- Modelling
- Interpret Results
- Future Iterations

Prompt from New Light Tech

Problem 10: Using live police radio reports for real time identification of people needing assistance.

> Currently, FEMA identifies areas that require immediate attention (for search and rescue efforts) either by responding to reports and requests put directly by the public or, recently, using social media posts. This tool will utilize live police radio reports to identify hot spots representing locations of people who need immediate attention. The tool will flag neighborhoods or specific streets where the police and first-respondents were called to provide assistance related to the event.

Convert to Data Science Problem

Problem 10: Using live police radio reports for real time identification of people needing assistance.

> Currently, FEMA identifies areas that require immediate attention (for search and rescue efforts) either by responding to reports and requests put directly by the public or, recently, using social media posts. This tool will utilize live police radio reports to identify hot spots representing locations of people who need immediate attention. The tool will flag neighborhoods or specific streets where the police and first-respondents were called to provide assistance related to the event.

Roadblocks in Prompt

- Working with live streaming audio is currently prohibited by the website that streams police scanner audio, so would require authorization from them.

- Speech-to-text, which would be necessary for identifying specific streets, is currently a very expensive service.

- A fully fledged tool would require development skills and resources that are beyond the scope of this course.

Convert to Data Science Problem

Create a model that can accurately classify audio into two categories: emergency and non-emergency.

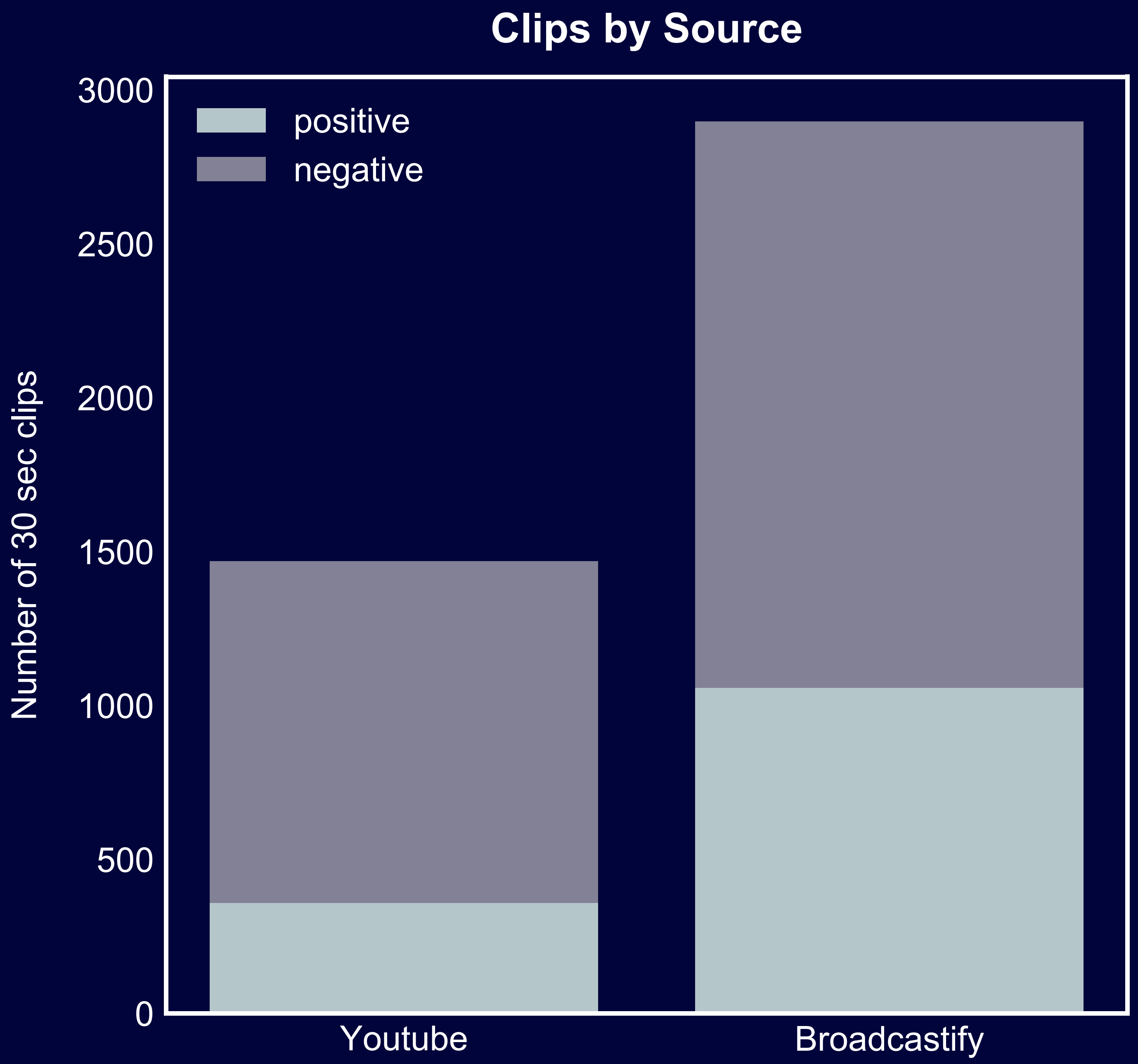
Gathering Data

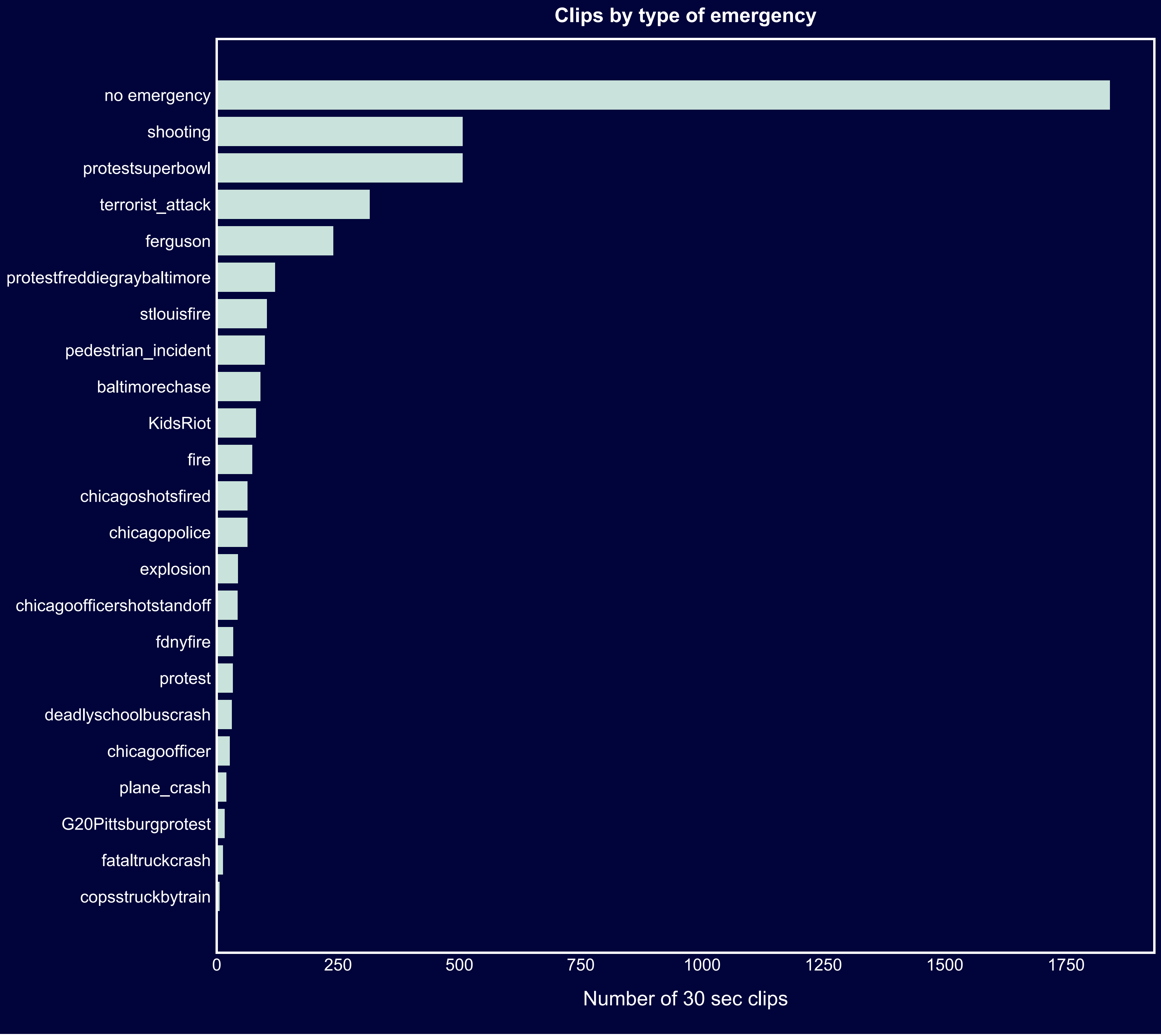
We gathered data primarily from Broadcastify.com, which houses an archive of police scanner data from all over the world.

We used Broadcastify’s notable broadcasts section to build our positive class - this includes scanner audio from mass shootings, terrorist attacks, and natural disasters.

We listened to random police scanners from the same regions that we gathered our positive class one and if no emergencies happened during the stream we marked it as negative class.

To balance our classes, we incorporated further scanner audio from FEMA declared emergencies.


Clips by Source
positive
negative
Number of 30 sec clips
3000
2500
2000
1500
1000
500
0
Youtube
Broadcastify



Clips by type of emergency

Preprocessing

Segmentation

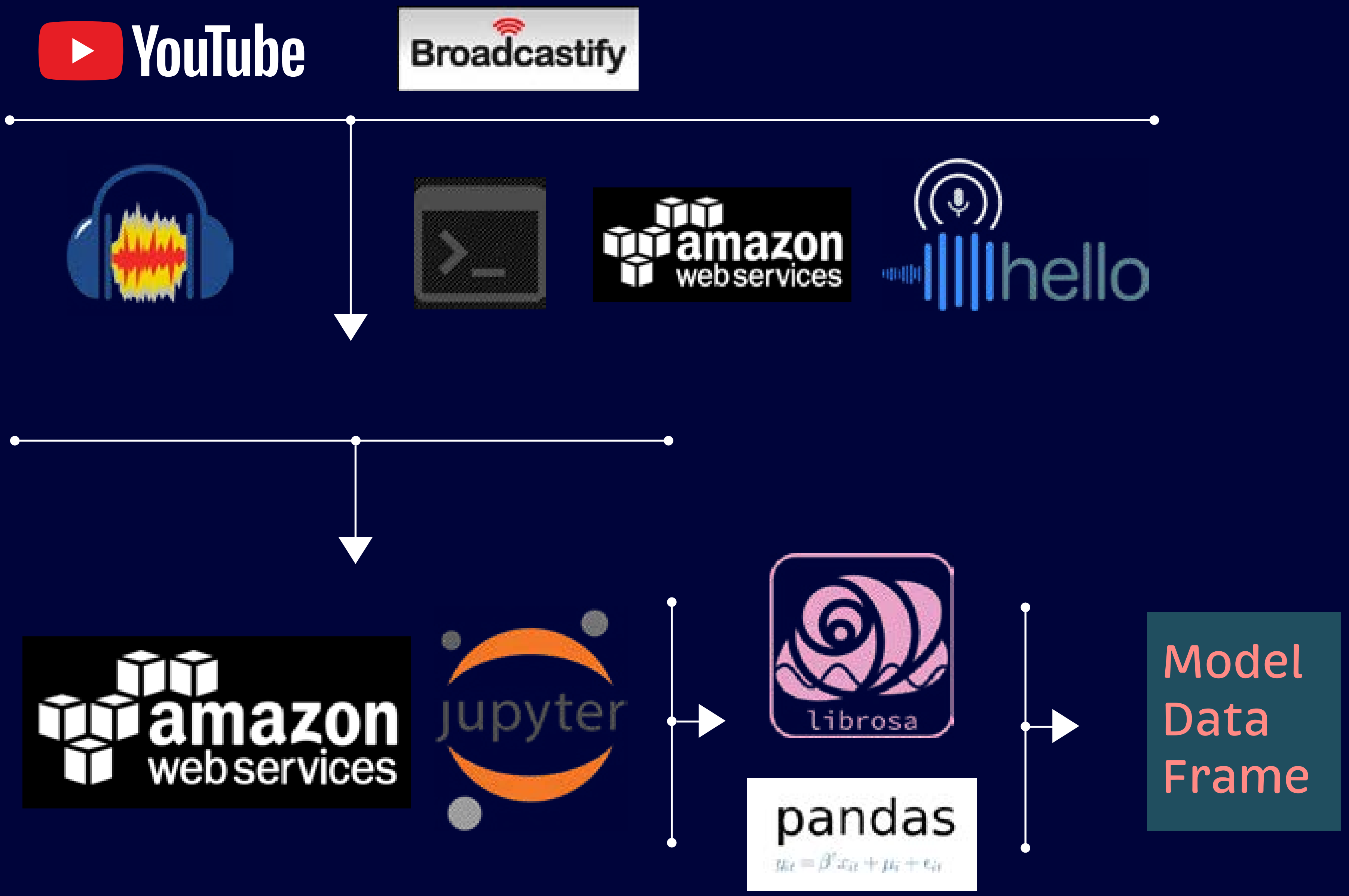
Each audio file was cut into thirty second clips to create a larger dataset and to more accurately simulate the process of listening to live audio.

Librosa Audio Features

Librosa is a powerful python package for audio processing and includes the ability to transform an audio track into its component features.

Using an AWS pipeline, we exported 612 features for each of the 4,368 clips, outputting the results into a CSV file.

Data Pipeline



Model Data Frame

Exploratory Data Analysis

Negative Class

Non-emergency Audio Example

We ended up with 2,950 negative samples.

Positive Class

Emergency Audio Example

We ended up with 1,418 positive samples.

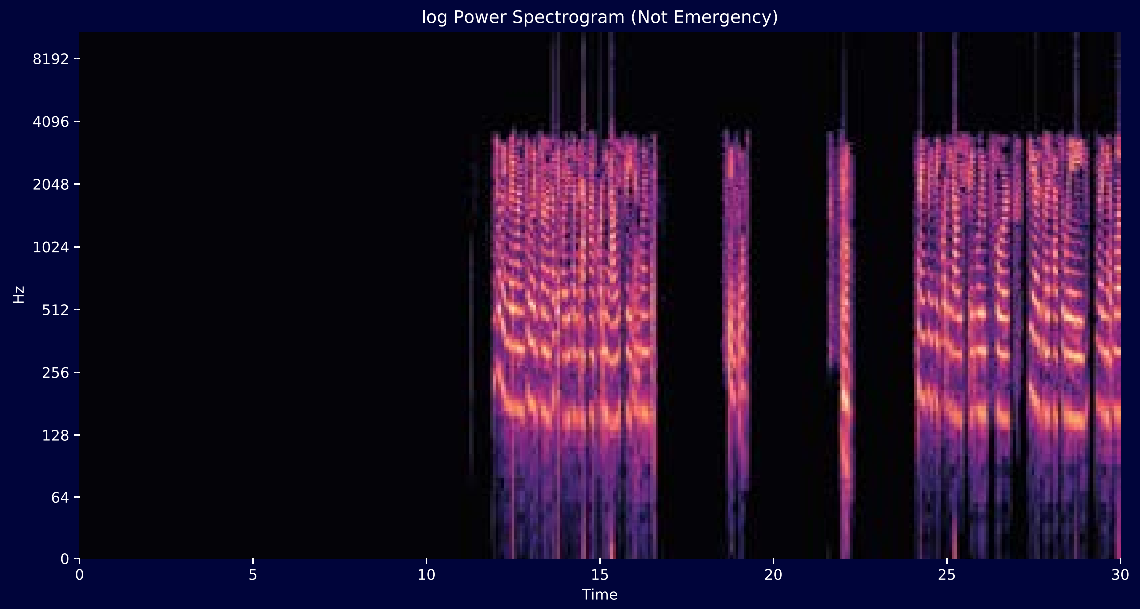
Exploratory Data Analysis

A brief explanation of some of the audio features:

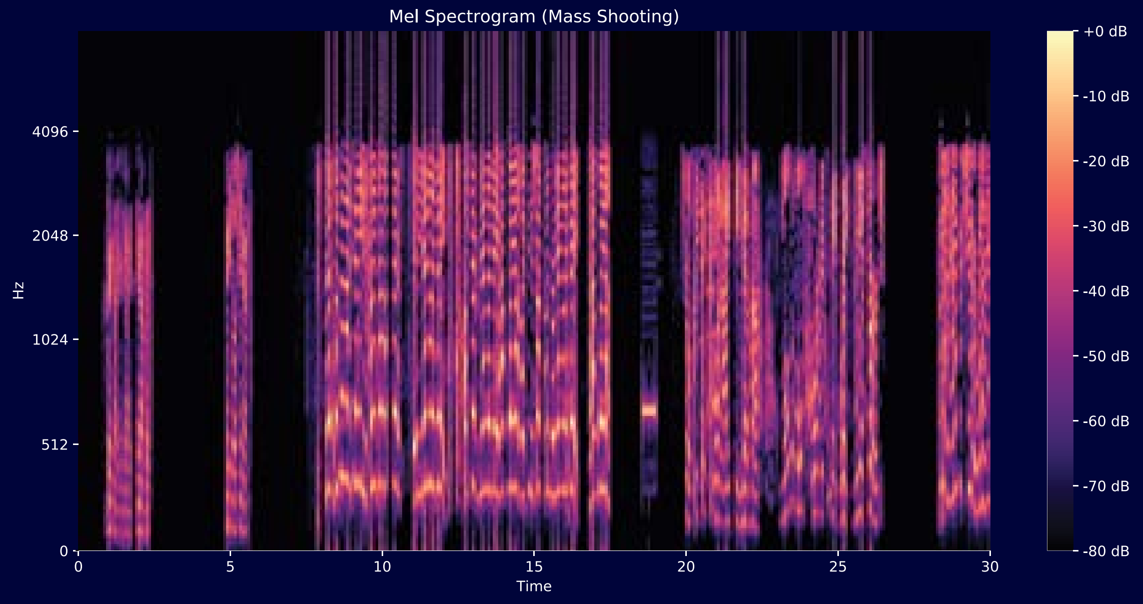
- Tempogram
- Chroma Features
- Spectral Contrast, Centroid, Band, and Flatness
- Tonnetz
- Contrast
- Mel
- MFCC

Exploratory Data Analysis - Mel Spectrogram

Negative Class

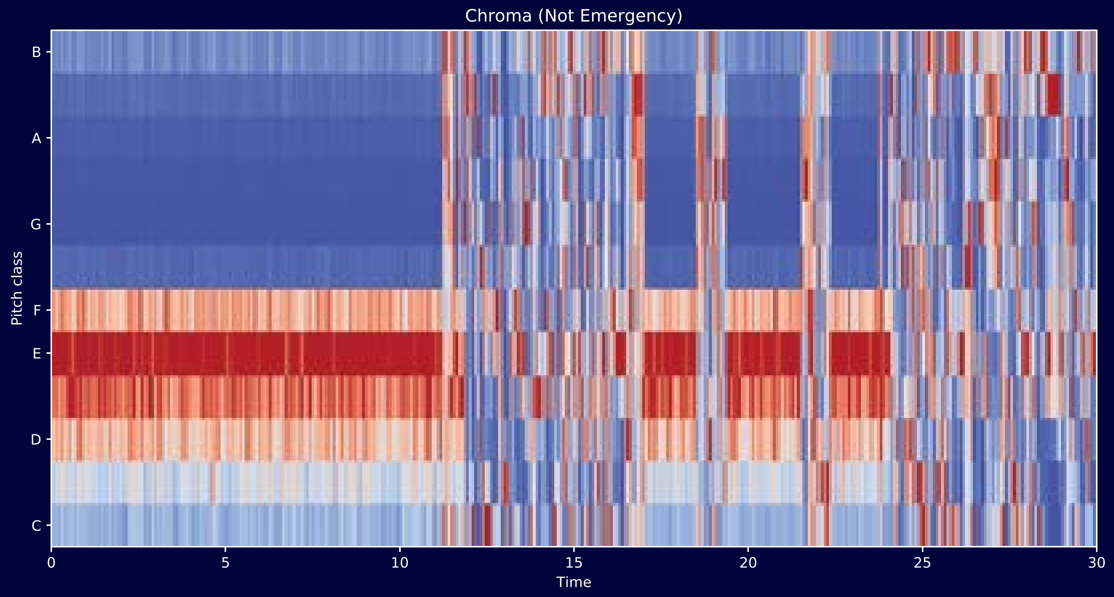


Positive Class

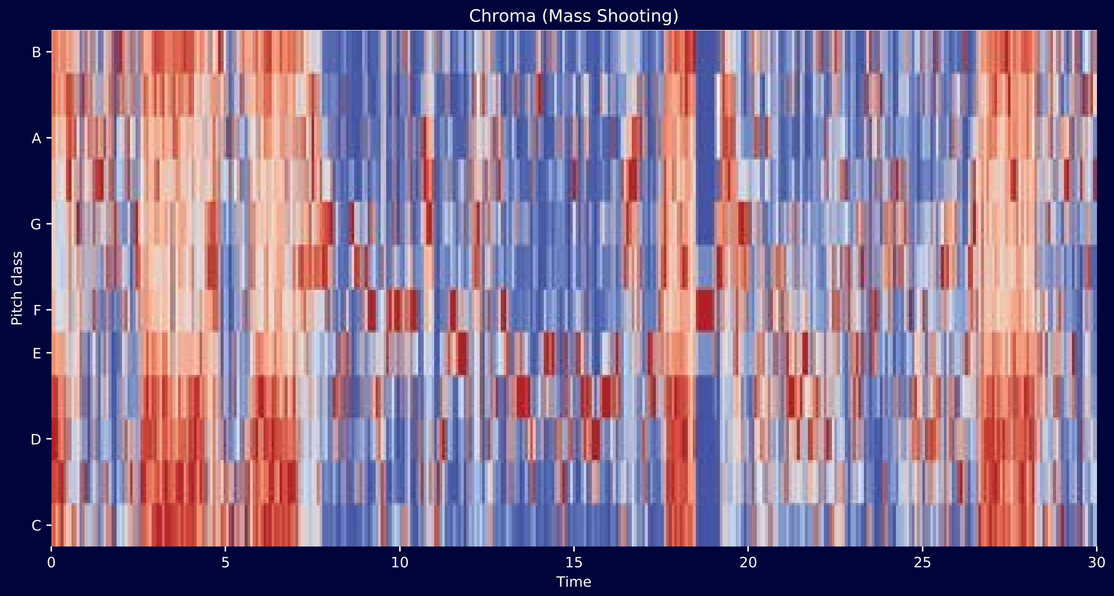


Exploratory Data Analysis - Chromagram

Negative Class

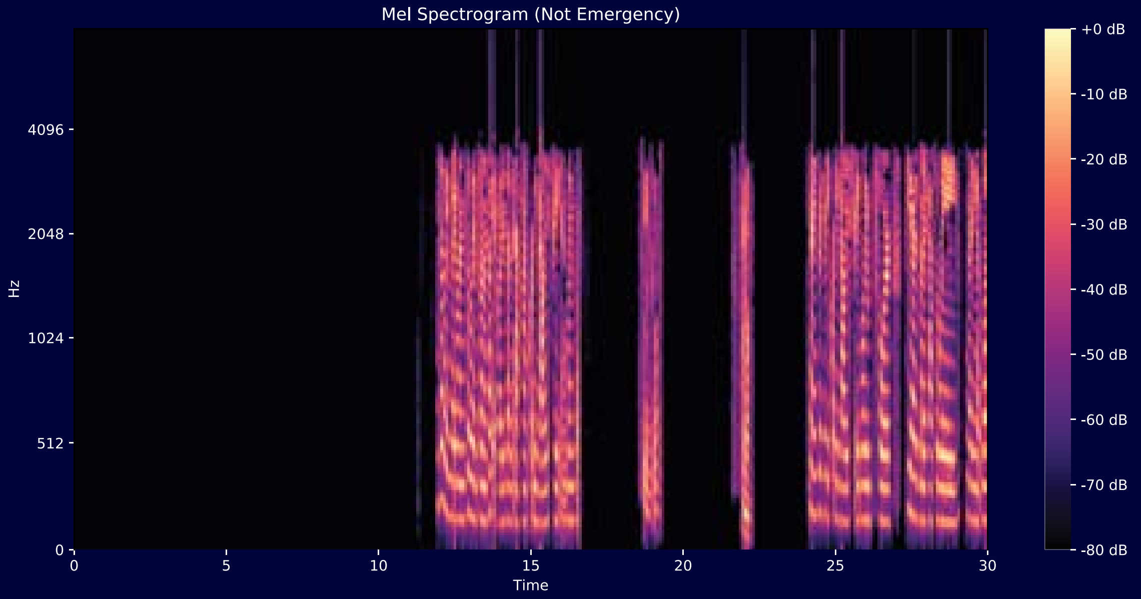


Positive Class

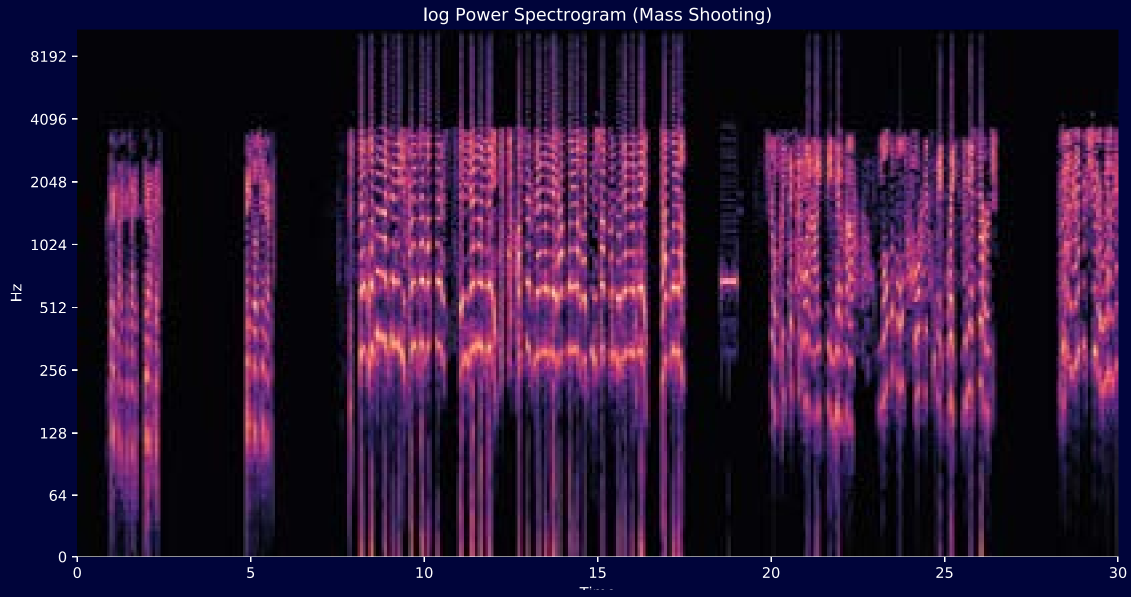


Exploratory Data Analysis - Spectrogram

Negative Class



Positive Class



Modelling

Initial Performance

- We fit logistic regression, gradient boost, decision tree, and random forest models to our data.
- All of the models performed exceptionally well, consistently providing accuracy of 95% and above.
- This caused immense suspicion on our part and we worked hard to ‘break’ the models and figure out why they were performing so well.
- The random forest model gave us the greatest accuracy and best reduced the variance between our training and testing sets.

Modelling

Optimization

> We went forward with the random forest model, given its performance and the ability to look at feature importance for additional validation of results.

> Due to the nature of emergency audio, we did not decide to optimize for sensitivity because some type 2 errors are to be expected.

Modelling

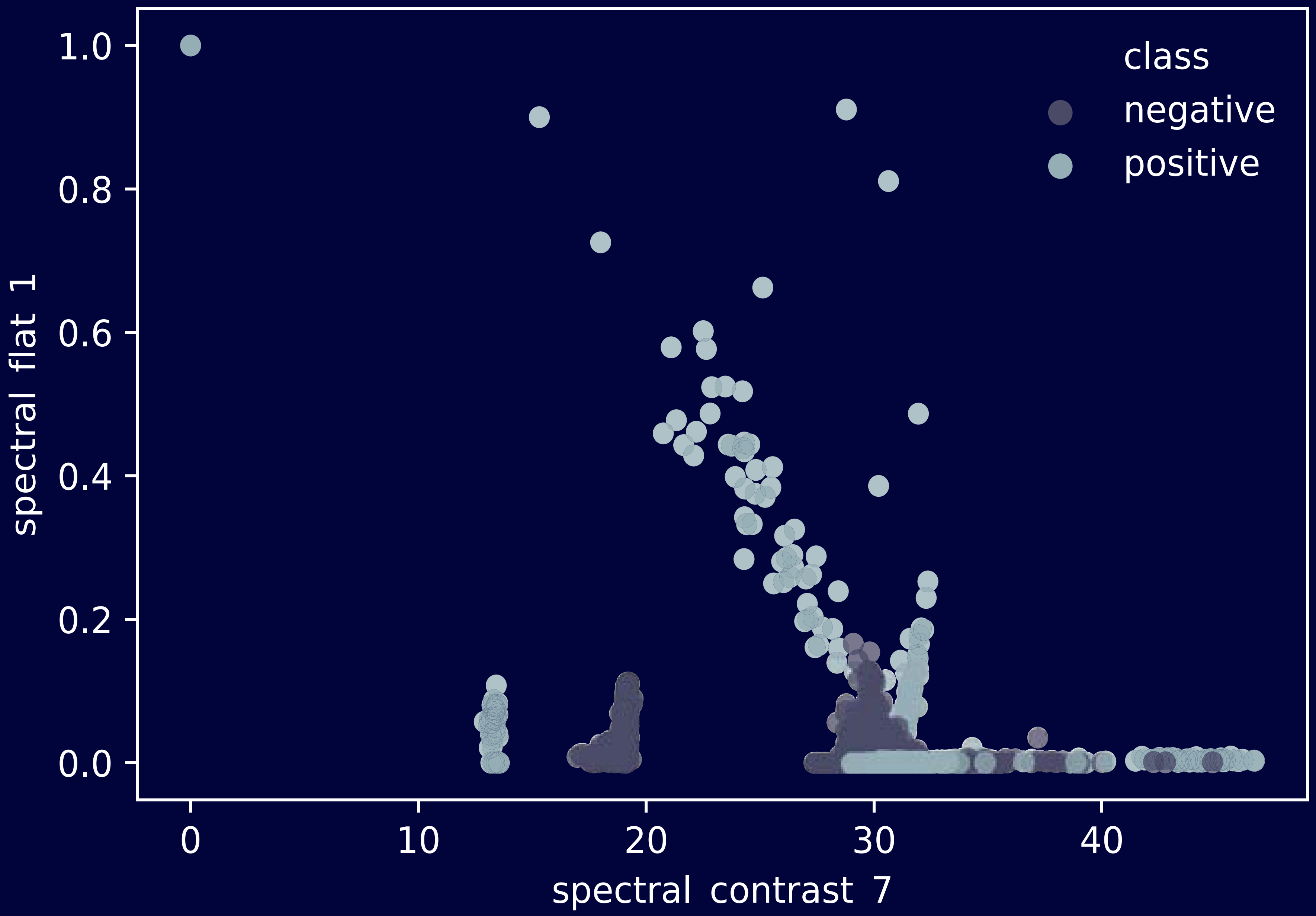
Final Performance

98.72%
Accuracy

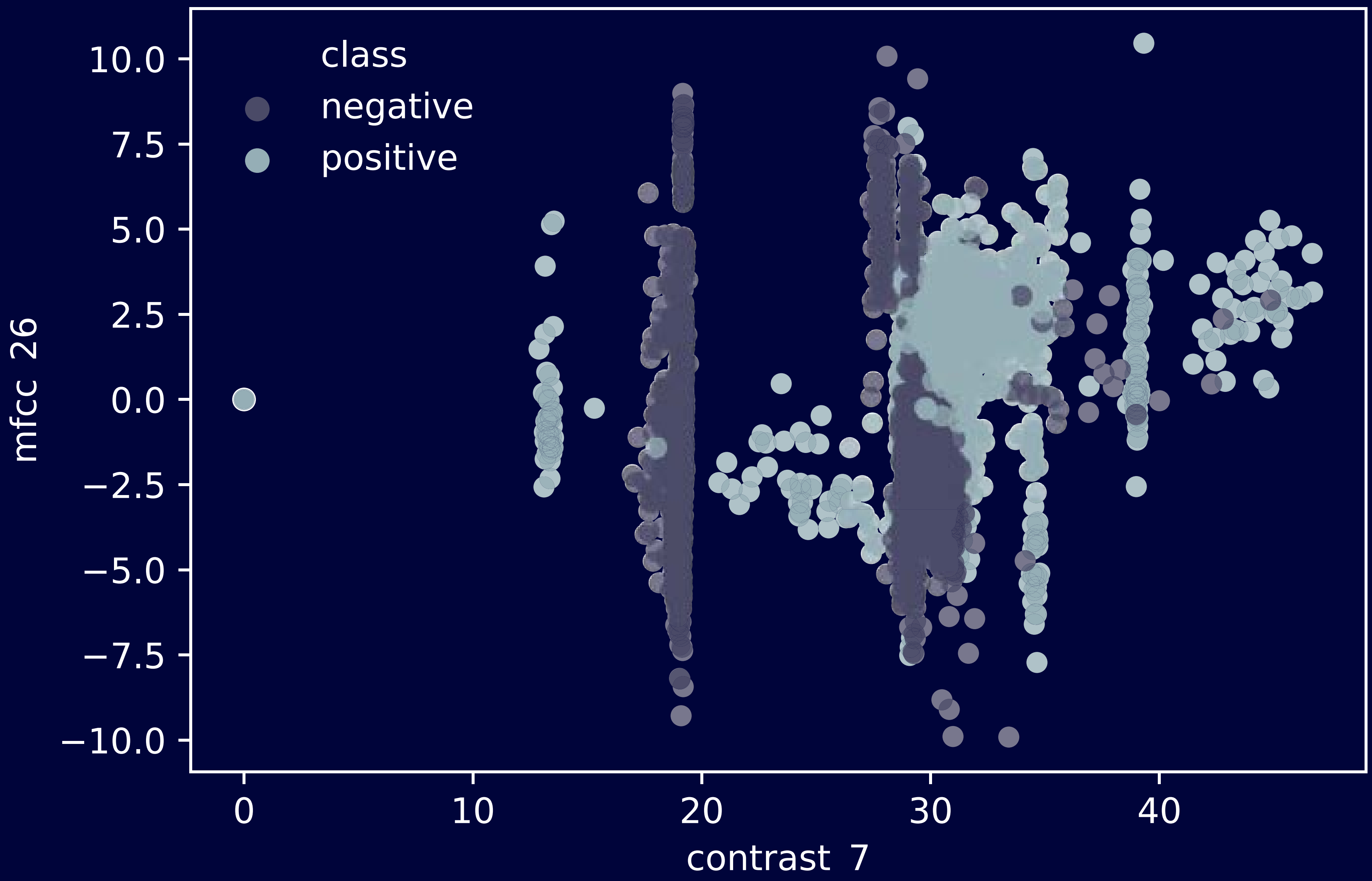
Confusion Matrix

	Not Emergency (P)	Emergency (P)
Not Emergency (A)	753	4
Emergency (A)	10	325

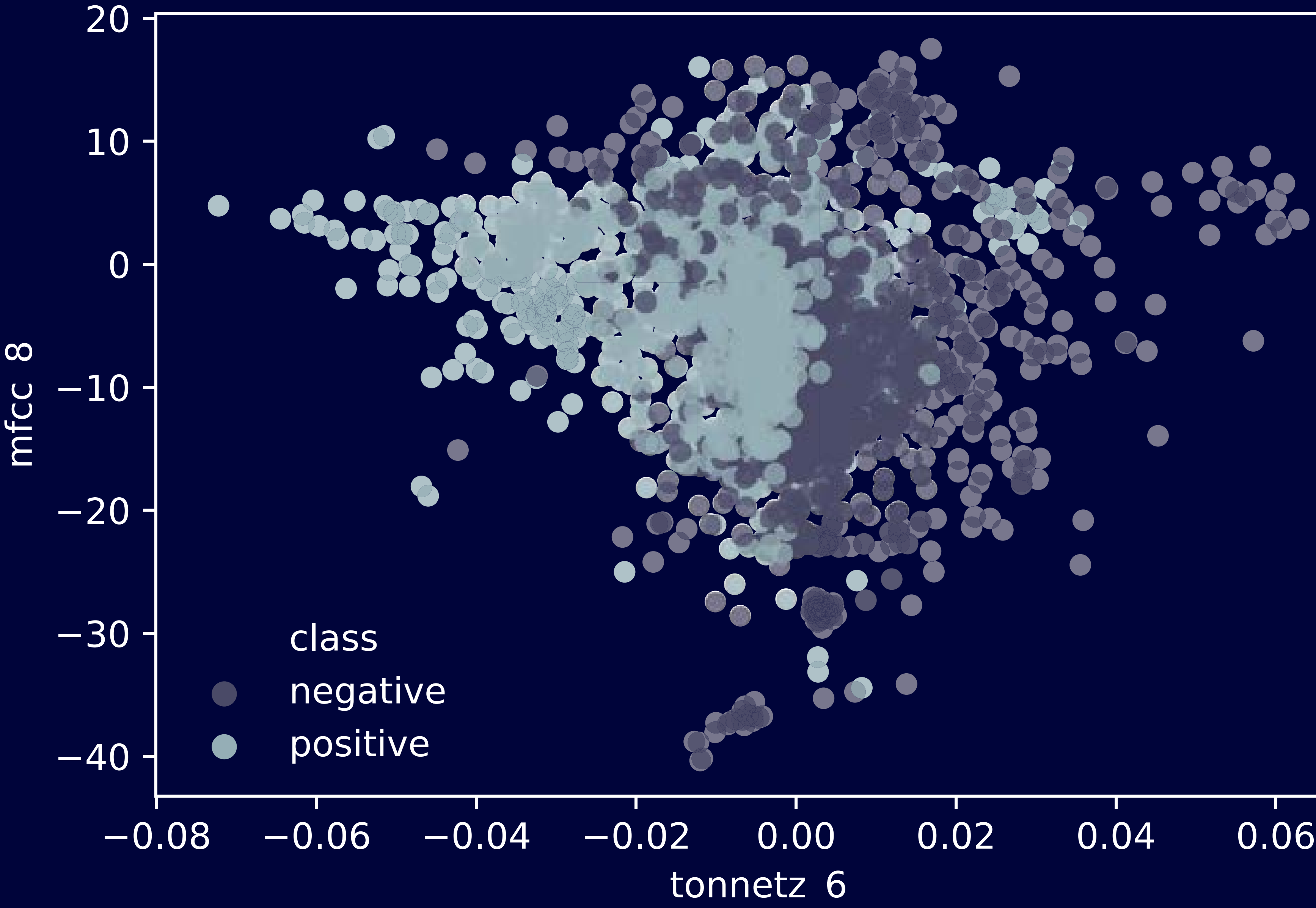
Spectral Flatness(bin 1) vs Spectral Contrast(bin 7)



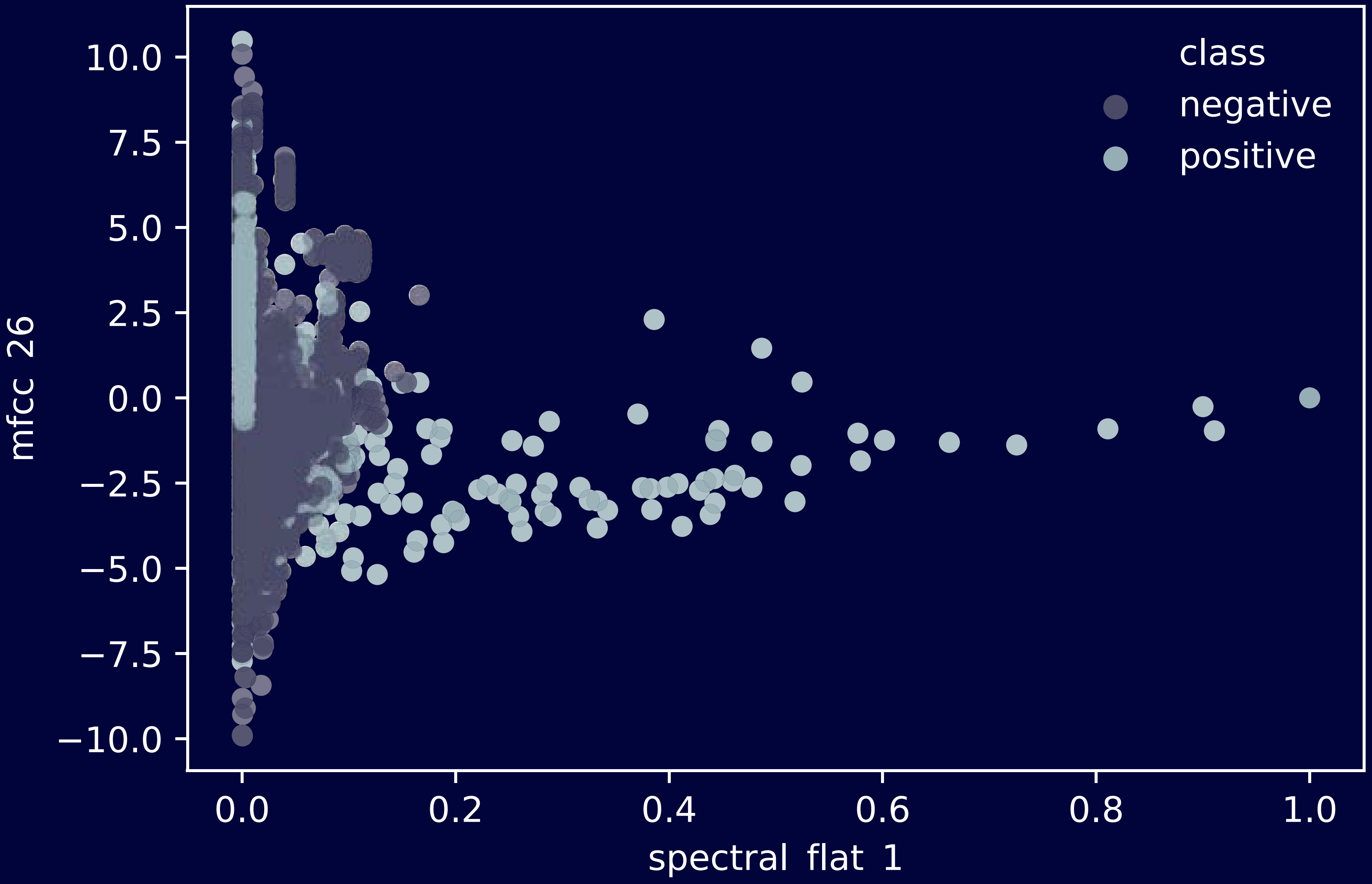
MFCC(bin 26) vs Contrast(bin 7)



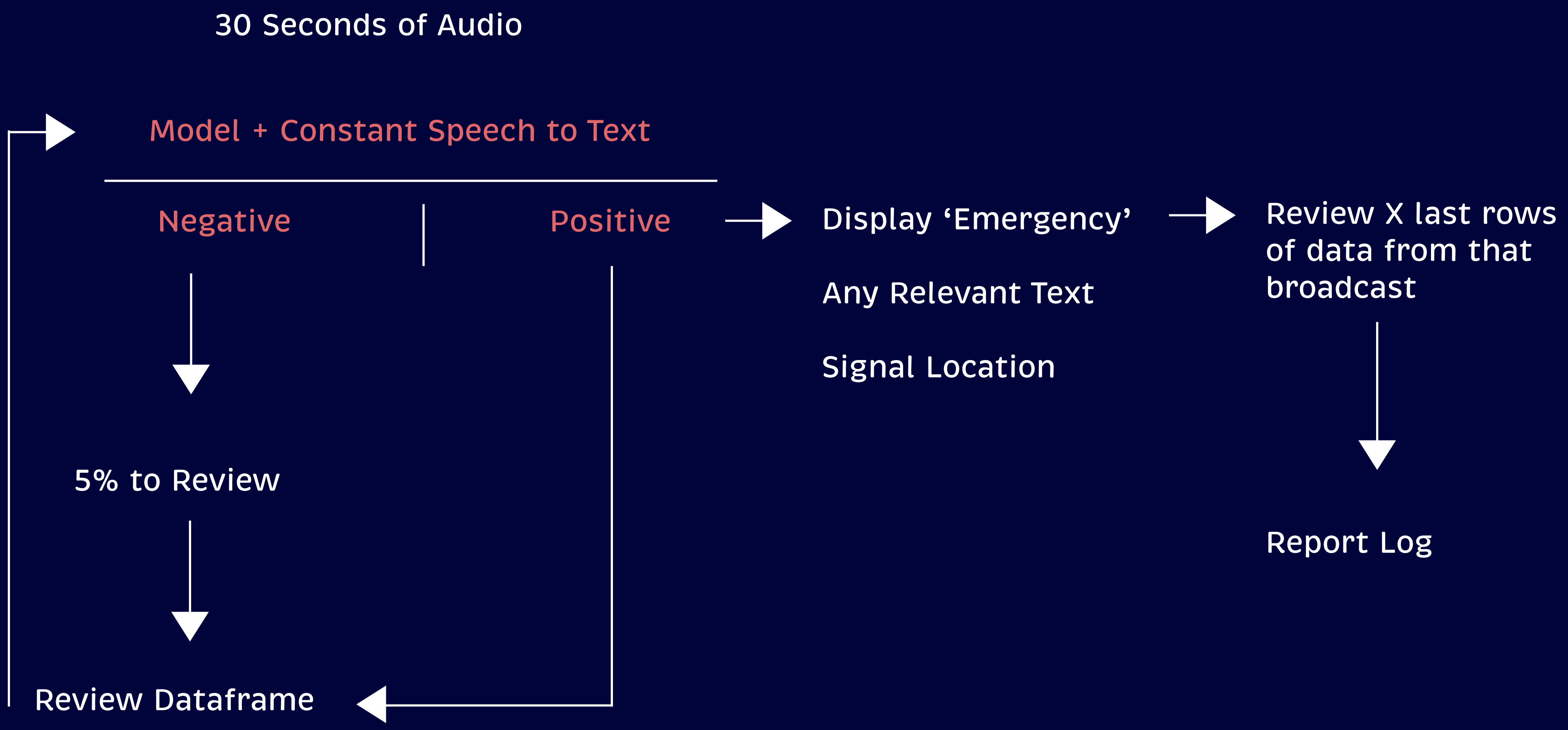
MFCC(bin 8) vs Tonnetz(bin 6)



MFCC(bin 26) vs Tonnetz(bin 1)



Future Iterations



30 Seconds of Audio

Model + Constant Speech to Text

Negative

Positive

Display ‘Emergency’

Any Relevant Text

Signal Location

Review X last rows of data from that broadcast

5% to Review

Review Dataframe

Report Log

Key Takeaways

Some recommendations based on our work:

- Audio proved to be a powerful predictor for an emergency situation. However, ideally this would be coupled with an equally powerful speech recognition model.

- Implementing a model such as this would allow disaster relief services to assume a more active role.

- The final model should incorporate self-evolving methodologies so that its predictive power grows with use.

Questions?