Honey, what was that?

An Audio Classifier for Household Sounds

Mateo Martinez

Data Science Capstone | December 2020

Springboard

Introduction

In the last ten years that has been huge leaps forward in the field of computer vision, with algorithms gaining high level understanding of the digital images.

For this project, we will apply these image classification algorithms to audio files by converting the raw audio into three feature sets: Mel-Frequency Spectrogram images,and Mel-Frequency Cepstral Coefficient images, and the mean Mel-Frequency Cepstral Coefficient values. The dataset we will investigate is comprised of a variety of typical household sounds.

Goal

Apply machine learning classification algorithms to our converted audio files to predict to the class label of typical household sounds.

Data

- The dataset for this project is titled Freesound Dataset 50k and was curated by The Music Technology Group of Universitat Pompeu Fabra.
- It contains 50,000 raw audio wav files. The curators have pre-split the data set into a training set of 40,966 files and a test set of 10,231 files.
- The dataset contains a title, description, and tags but no class labels, which need to be wrangled.
- The audio files contain a wide array of sounds, from music, speech, animal,nature, city, and household sounds.

title	description	tags	license	uploader	track_num
Spring Birds Forest 04 Amp.wav	An other birds singings recorded on the mornin...	[birdsong, bird, forest, environment, morning,...	http://creativecommons.org/publicdomain/zero/1.0/	ANARKYA	391277
Snap of fingers	a snap of one's fingers	[fingers, finger, 5maudio17, uam, fingersnap]	http://creativecommons.org/publicdomain/zero/1.0/	edton	392115
Pouring Water	A sound of Hot water pouring into a cup	[fill, can, beverage, glass, water, pour, drin...	http://creativecommons.org/publicdomain/zero/1.0/	edsward	411438
Tearing papers.wav	Tearing papers with reverb.\r\n\r\nI used a Sa...	[ripping, papier, paper, scheuren, rip, tearing]	http://creativecommons.org/licenses/by/3.0/	ellenmentor	395238

Business Application & Stakeholders

This audio classifier benefits stakeholders interested in home security systems. This algorithm could inform a resident of the category of a sound event, which could be used as an alert/warning system for unwarranted sounds.

This project benefits stakeholders interested in image based audio classification algorithms, as the comparative analysis of the Mel-Frequency, MFCC, and Mean-MFCC feature sets informs decisions for feature selection in future classification models.

Explanatory Features

To fully explore the best method of applying image classification algorithms, we will extract three different sets of features.

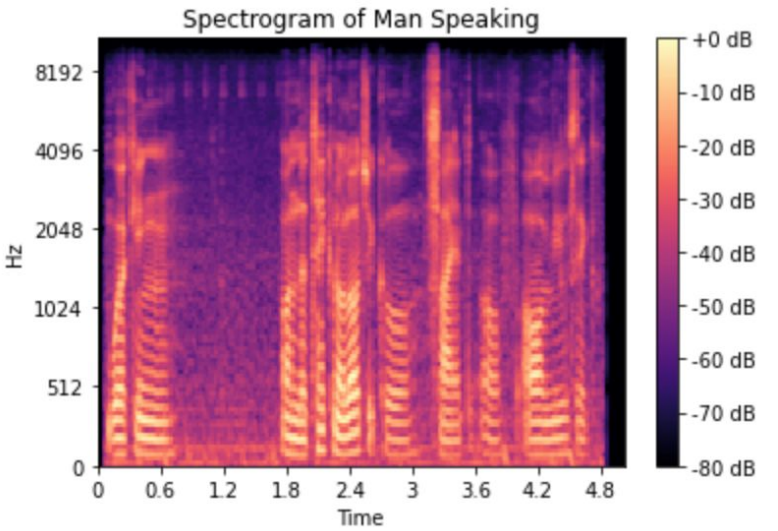
The First two sets of features, Mel-Frequency Spectrograms and Mel-Frequency Cepstral Coefficients Spectrograms, will be used to train Convolutional Neural Networks.

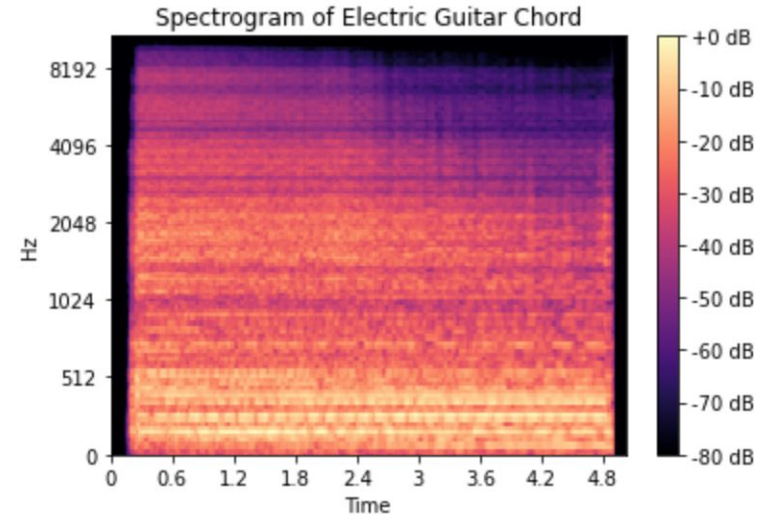
The Third set of features, the mean of the MFCC values over time elapsed, is now has a one dimensional feature space, which we will use to train non-image based classifiers: non-convolutional Neural Network, Logistic Regression, Random Forest, and Support Vector Classifier.

EDA: Mel-Frequency Spectrograms

Humans perceive sound logarithmically, thus to have a visualization that matches how we perceive equal increments of sound, we will convert the the audio to the Mel-Frequency scale.

For the Spectrogram, we will extract 128 Mel-Frequency bands, as the corresponds to the tonal range of human hearing. We will trim/pad the audio files to five seconds, extracting a total of 216 audio events with that period, resulting in a 128x216 pixel image.



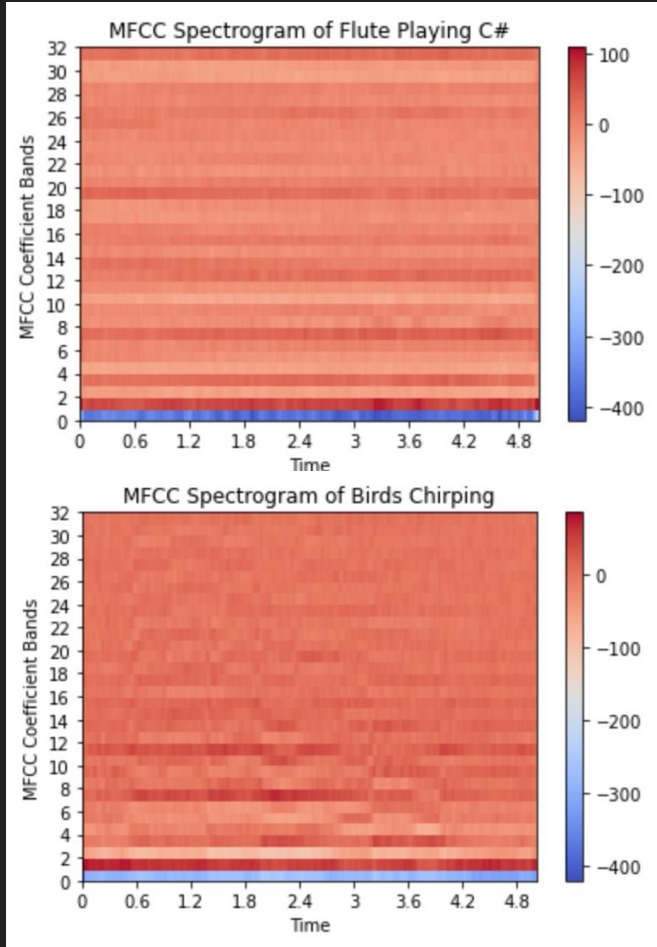


EDA: Mel-Frequency Cepstral Coefficient Spectrograms

The MFCC Spectrogram is a further compressed representation of a Mel-Frequency Spectrogram, taking the spectrum of the Mel-Frequency spectrum.

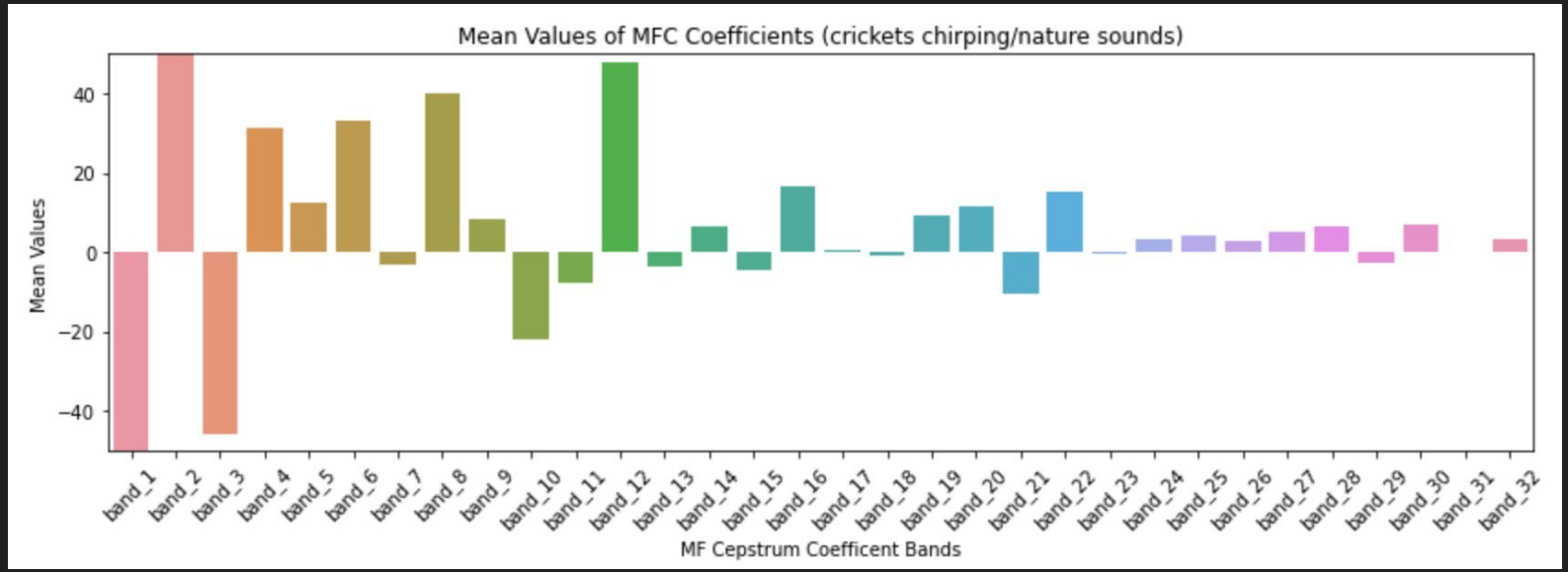
Typically for speech analysis, 12-13 Mel-Frequency bands are extracted for European languages and 20 for Asian languages.

However, given that this dataset set includes much more tonally nuanced audio events, we will extract 32 Mel-Frequency bands, with 216 audio events across the five second interval, resulting in 32x216 pixel images.



EDA: Mean MFCC Values

To use non-image based classifiers, we will take the mean of the MFFC values averaged over time. This will result in a one dimensional array of 32 features, each representing a Mel-Frequency band.



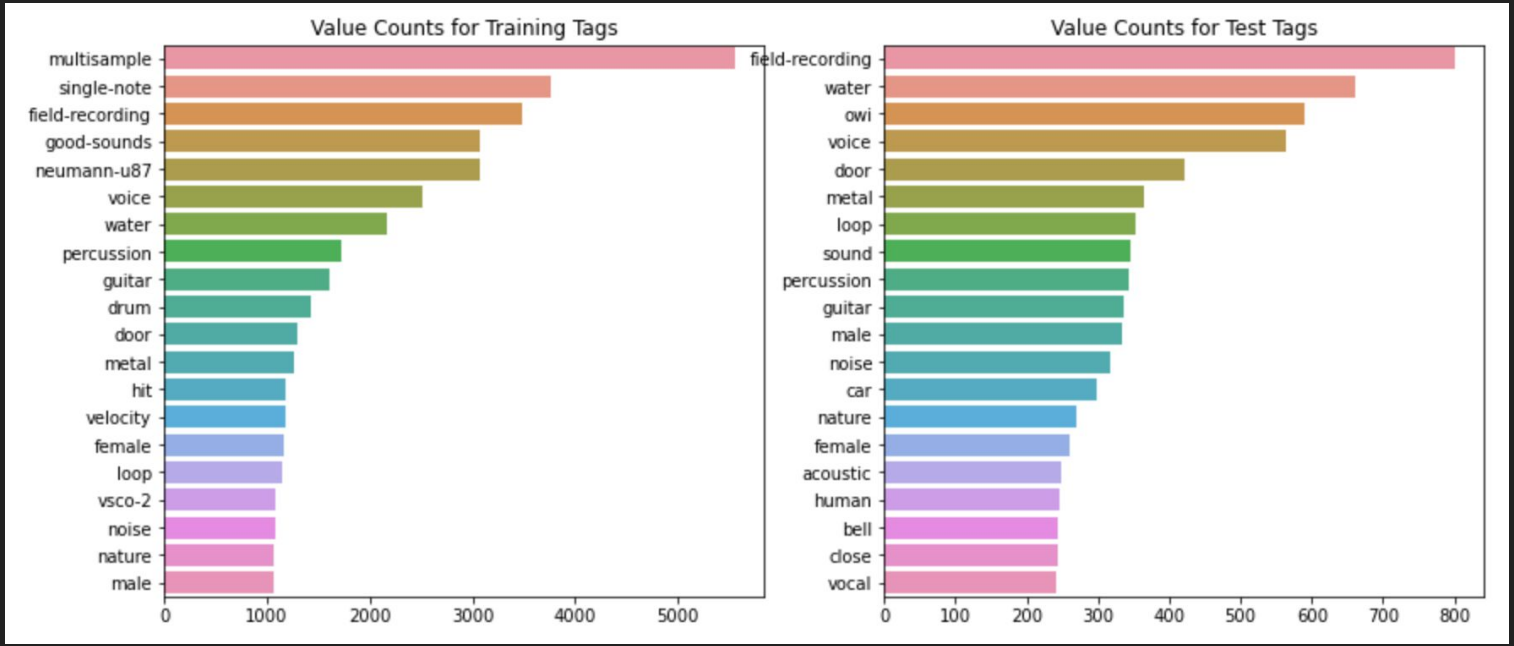
EDA-Audio Tags

This dataset was not curated with class labels. To wrangle class labels, we will use the audio tags, of which there are over 20,000 unique tags. Below is a sample of 20 rows of tags:

```
63                                                                [male, voice]
136                                                [keyboard, rhythmic, tap, type]
137    [computer, environmental-sounds-research, key, keyboard, tap, type, typing]
221                                                                       [bell]
236                                                [bus, depart, drive, station]
237                                                       [car, engine, passing]
247                                                        [dutch, male, speech]
263                                                           [children, crying]
281                                [environmental-sounds-research, field-recording, train]
283                                      [crescendo, environmental-sounds-research, train]
305                                                           [cat, purr, sneeze]
334                                                [anechoic, keys, metal, ring]
344                                                 [bark, dog, field-recording]
374                                          [door, environmental-sounds-research, lock]
410                                 [bass-drum, found-sound, household, kick, percussion]
420                                           [crickets, found-sound, frogs, nature]
421                                                  [drums, hi-hat, percussion]
422                                                  [drums, hi-hat, percussion]
423                                                  [drums, hi-hat, percussion]
424                                                  [drums, hi-hat, percussion]
Name: tags, dtype: object
```

EDA - Audio Tags

These graphs depict the most frequently used tags for the training and test data. It is notable that between the two datasets there is quite a discrepancy between the most common tags.



EDA - Audio Tags

Here is a look at the full tag metadata for highest tag counts:

Training:“multisample”

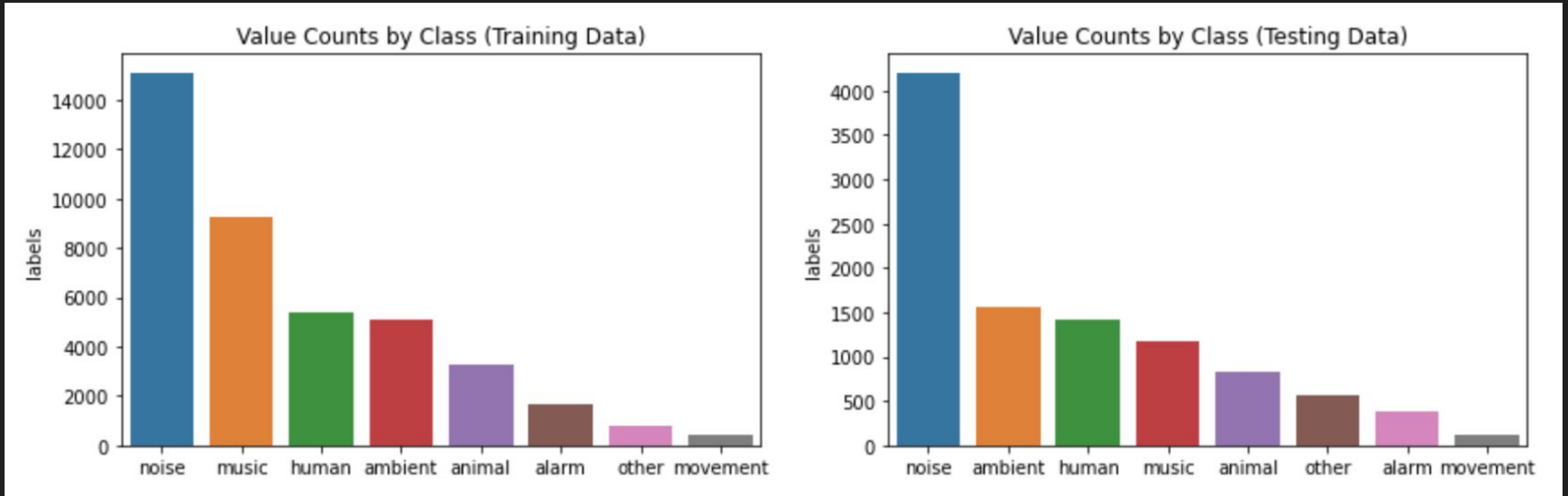
```
['acoustic', 'chord', 'guitar', 'minor', 'multisample', 'yamaha']
['acoustic', 'chord', 'diminished', 'guitar', 'multisample', 'yamaha']
['bells', 'gamelan', 'gong', 'indonesia', 'multisample', 'percussion',
['bells', 'gamelan', 'gong', 'indonesia', 'multisample', 'percussion',
['bells', 'gamelan', 'gong', 'indonesia', 'multisample', 'percussion',
['bells', 'gamelan', 'gong', 'indonesia', 'multisample', 'percussion',
['bells', 'gamelan', 'gong', 'indonesia', 'multisample', 'percussion',
['acoustic', 'cello', 'multisample', 'pizzicato', 'pluck', 'strings',
['acoustic', 'cello', 'multisample', 'pizzicato', 'pluck', 'strings',
['acoustic', 'cello', 'multisample', 'pizzicato', 'pluck', 'strings',
['acoustic', 'cello', 'multisample', 'pizzicato', 'pluck', 'strings',
['acoustic', 'cello', 'multisample', 'pizzicato', 'pluck', 'strings',
['acoustic', 'cello', 'multisample', 'pizzicato', 'pluck', 'strings',
['acoustic', 'cello', 'multisample', 'pizzicato', 'pluck', 'strings',
['acoustic', 'cello', 'multisample', 'pizzicato', 'pluck', 'strings',
['acoustic', 'cello', 'multisample', 'pizzicato', 'pluck', 'strings',
['acoustic', 'cello', 'multisample', 'pizzicato', 'pluck', 'strings',
['bass', 'fender', 'finger', 'multisample', 'p-bass', 'sidekick', 'str
['bass', 'fender', 'finger', 'multisample', 'p-bass', 'sidekick', 'str
['18', 'aax', 'crash', 'cymbal', 'cymbals', 'drums', 'multi-sample', '
```

Test: “field recording

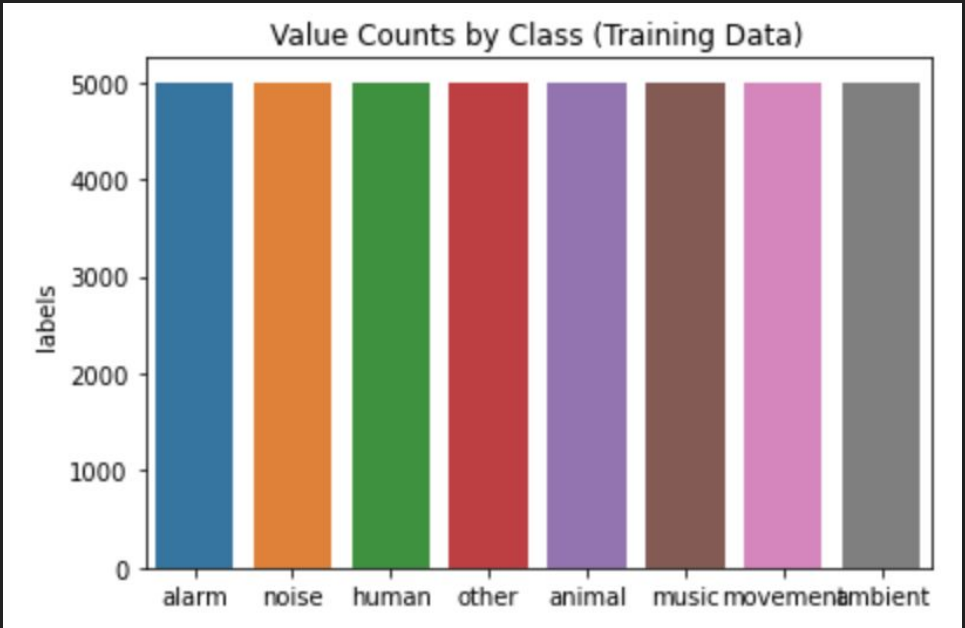
```
['environmental-sounds-research', 'field-recording', 'train']
['bark', 'dog', 'field-recording']
['cat', 'field-recording', 'purr']
['cat', 'field-recording', 'roar']
['cat', 'field-recording', 'purr']
['beat', 'field-recording', 'kitchen', 'metal']
['bass', 'drone', 'field-recording', 'machinery']
['field-recording', 'hiss', 'noise', 'rustle', 'stove-pipe', 'wind']
['field-recording', 'hiss', 'rustle', 'stove-pipe', 'wind']
['bridge', 'canada', 'field-recording', 'montreal', 'passing', 'train']
['fiddle', 'field-recording', 'melody']
['fiddle', 'field-recording', 'melody']
['field-recording', 'fireworks', 'sequence']
['bass', 'explosion', 'field-recording', 'fireworks', 'pan']
['explosion', 'field-recording', 'fireworks', 'snare']
['beep', 'field-recording', 'fireworks', 'light']
['close', 'field-recording', 'loud', 'purist', 'thunder', 'wheather']
['ball', 'bird', 'field-recording', 'street']
['car', 'field-recording', 'passing', 'street']
['car', 'datsun-180b', 'engine', 'field-recording', 'stop']
['car', 'datsun-180b', 'engine', 'field-recording', 'start']
['car', 'datsun-180b', 'field-recording', 'horn']
['car', 'datsun-180b', 'engine', 'field-recording', 'gas']
```

Imbalanced Classes

From these 20,000 unique, we will bin them into eight class labelled categories, by heuristically binning each tag into a class category. As is evident in the graphs below, there is significant imbalanced classes. We will resample the training data with random undersampling and random oversampling with data augmentation, consisting of adding gaussian noise and shifting the pitch.



Resampled Training Data



After resampling the training data, each class contains 5,000 audio files.

Pre-Processing

- Images scaled to a min-max range of zero to one
- Mean MFCC values scaled with a zero mean and unit variance
- Splitting Test data 50-50 into Validation and Test sets for neural network hyperparameter tuning
- Minor data cleaning of dataframe columns used in EDA

Modeling

We will train a total of seven classification models using the three feature sets:

1a. Convolutional Neural Network with Mel-Frequency Spectrograms

2a. CNN with MFCC Spectrograms
2b. CNN with ResNet50 Transfer Learning with MFCC Spectrograms

3a. Neural Network with Mean MFCC Values
3b. Logistic Regression with Mean MFCC Values
3c. Random Forest with Mean MFCC Values
3d. Support Vector Classifier with Mean MFCC Values

Modeling- CNN with Mel-Frequency Spectrograms

Given that the training and test datasets contained a significant proportion of distinct audio files, the biggest challenge was the model overfitting to the training set.

This was addressed with a combination of Max Pooling, Batch Normalization, Dropout layers, and relatively low number of filters.

Loss: Categorical Cross-Entropy
Optimizer: SGD
Epochs: 40

Layer (type)	Output Shape	Param #
conv2d (Conv2D)	(None, 128, 216, 16)	160
max_pooling2d (MaxPooling2D)	(None, 128, 108, 16)	0
conv2d_1 (Conv2D)	(None, 128, 108, 16)	2320
max_pooling2d_1 (MaxPooling2	(None, 64, 54, 16)	0
conv2d_2 (Conv2D)	(None, 64, 54, 32)	4640
max_pooling2d_2 (MaxPooling2	(None, 32, 27, 32)	0
conv2d_3 (Conv2D)	(None, 32, 27, 32)	9248
max_pooling2d_3 (MaxPooling2	(None, 16, 13, 32)	0
batch_normalization (BatchNo	(None, 16, 13, 32)	128
dropout (Dropout)	(None, 16, 13, 32)	0
flatten (Flatten)	(None, 6656)	0
dense (Dense)	(None, 64)	426048
dropout_1 (Dropout)	(None, 64)	0
dense_1 (Dense)	(None, 8)	520

Total params: 443,064
Trainable params: 443,000
Non-trainable params: 64

Modeling- CNN with Mel-Frequency Spectrograms

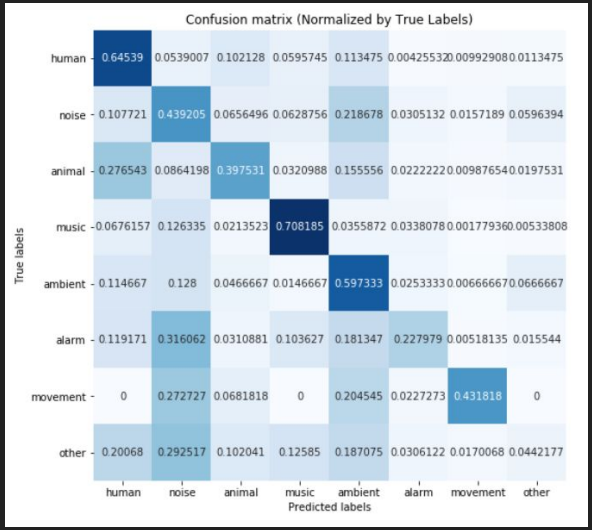
Results:

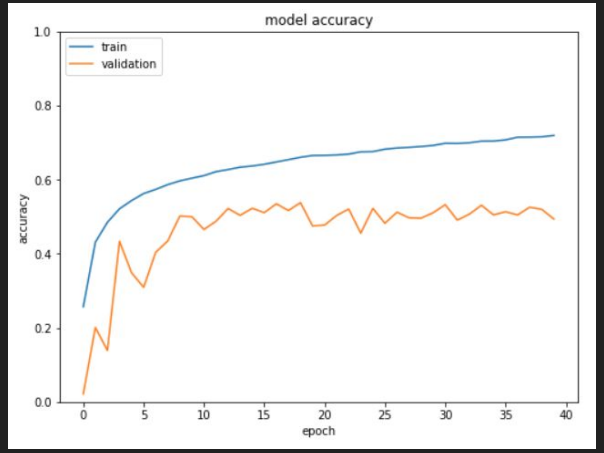
Train Accuracy: 0.7209

Validation Accuracy: 0.4936

Test Accuracy: 0.49

Test F1-Score: 0.40





classification Report:

	precision	recall	f1-score	support
0	0.45	0.65	0.53	705
1	0.70	0.44	0.54	2163
2	0.35	0.40	0.37	405
3	0.61	0.71	0.65	562
4	0.38	0.60	0.46	750
5	0.26	0.23	0.24	193
6	0.25	0.43	0.32	44
7	0.06	0.04	0.05	294
accuracy			0.49	5116
macro avg	0.38	0.44	0.40	5116
weighted avg	0.53	0.49	0.49	5116

Modeling- CNN with MFCC Spectrograms

This model struggled with the same problem of overfitting and required the similar layer architecture, with slightly different Max Pooling layers and number of filters.

Loss: Categorical Cross-Entropy
Optimizer: SGD
Epochs: 40

Layer (type)	Output Shape	Param #
conv2d (Conv2D)	(None, 32, 216, 8)	80
max_pooling2d (MaxPooling2D)	(None, 32, 108, 8)	0
conv2d_1 (Conv2D)	(None, 32, 108, 8)	584
max_pooling2d_1 (MaxPooling2	(None, 32, 54, 8)	0
conv2d_2 (Conv2D)	(None, 32, 54, 16)	1168
max_pooling2d_2 (MaxPooling2	(None, 32, 27, 16)	0
conv2d_3 (Conv2D)	(None, 32, 27, 16)	2320
max_pooling2d_3 (MaxPooling2	(None, 16, 13, 16)	0
conv2d_4 (Conv2D)	(None, 16, 13, 32)	4640
dropout (Dropout)	(None, 16, 13, 32)	0
conv2d_5 (Conv2D)	(None, 16, 13, 32)	9248
batch_normalization (BatchNo	(None, 16, 13, 32)	128
dropout_1 (Dropout)	(None, 16, 13, 32)	0
flatten (Flatten)	(None, 6656)	0
dense (Dense)	(None, 128)	852096
dropout_2 (Dropout)	(None, 128)	0
dense_1 (Dense)	(None, 8)	1032

Total params: 871,296
Trainable params: 871,232
Non-trainable params: 64

Modeling- CNN with MFCC Spectrograms

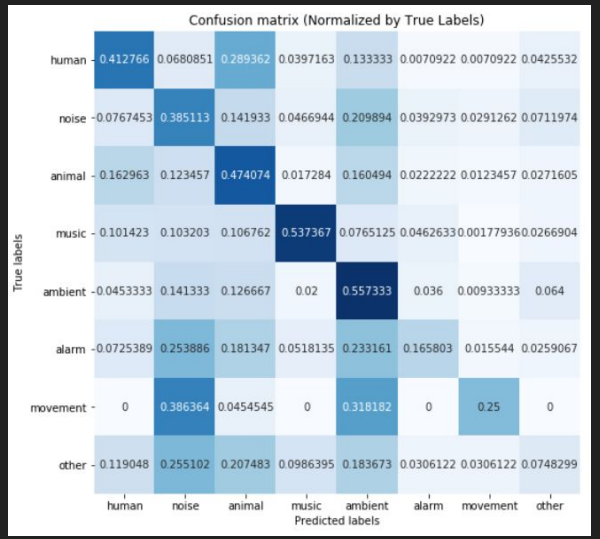
Results:

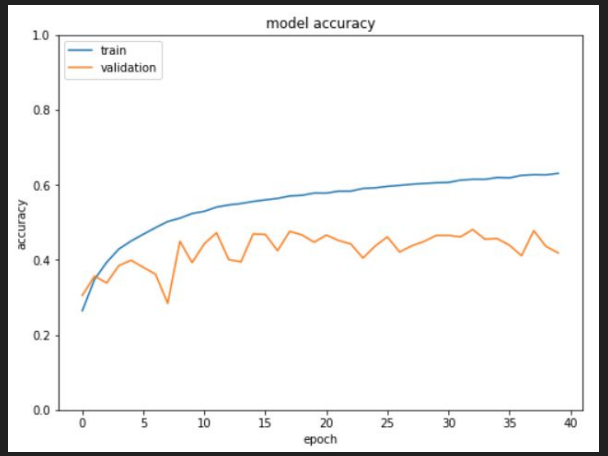
Train Accuracy: 0.6279

Validation Accuracy: 0.4186

Test Accuracy: 0.41

Test F1-Score: 0.32





classification Report:

	precision	recall	f1-score	support
0	0.44	0.41	0.43	705
1	0.67	0.39	0.49	2163
2	0.20	0.47	0.28	405
3	0.61	0.54	0.57	562
4	0.35	0.56	0.43	750
5	0.17	0.17	0.17	193
6	0.11	0.25	0.15	44
7	0.08	0.07	0.08	294
accuracy			0.41	5116
macro avg	0.33	0.36	0.32	5116
weighted avg	0.49	0.41	0.43	5116

Modeling- CNN with ResNet50 with MFCC Spectrograms

This model was trained using Keras to import the ResNet50 model with ImageNet weights. Trainable layers of the ResNet model were unfrozen.

This model did not perform as well as expected potentially due to the difference between input image dimensions used on the ImageNet dataset.

Loss: Categorical Cross-Entropy
Optimizer: SGD
Epochs: 40

conv5_block3_2_bn (BatchNormali	(None, 1, 7, 512)	2048
conv5_block3_2_relu (Activation	(None, 1, 7, 512)	0
conv5_block3_3_conv (Conv2D)	(None, 1, 7, 2048)	1050624
conv5_block3_3_bn (BatchNormali	(None, 1, 7, 2048)	8192
conv5_block3_add (Add)	(None, 1, 7, 2048)	0
conv5_block3_out (Activation)	(None, 1, 7, 2048)	0
flatten (Flatten)	(None, 14336)	0

Total params: 23,587,712
Trainable params: 53,120
Non-trainable params: 23,534,592

Layer (type)	Output Shape	Param #
model (Functional)	(None, 14336)	23587712
dense (Dense)	(None, 256)	3670272
dropout (Dropout)	(None, 256)	0
dense_1 (Dense)	(None, 128)	32896
dropout_1 (Dropout)	(None, 128)	0
dense_2 (Dense)	(None, 64)	8256
dropout_2 (Dropout)	(None, 64)	0
dense_3 (Dense)	(None, 8)	520

Total params: 27,299,656
Trainable params: 3,765,064
Non-trainable params: 23,534,592

Modeling- CNN with ResNet50 with MFCC Spectrograms

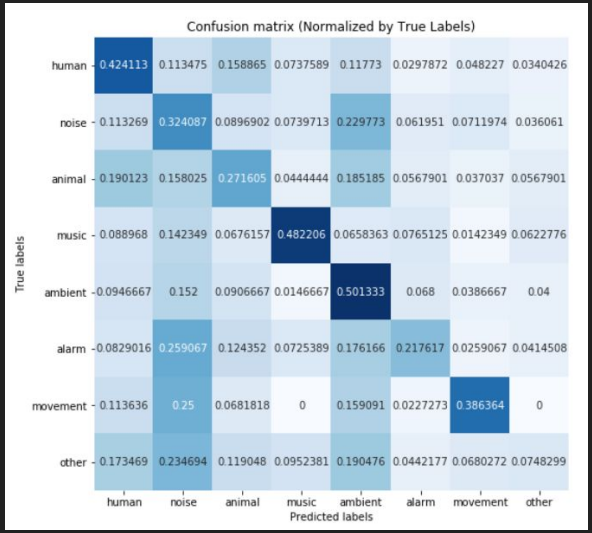
Results:

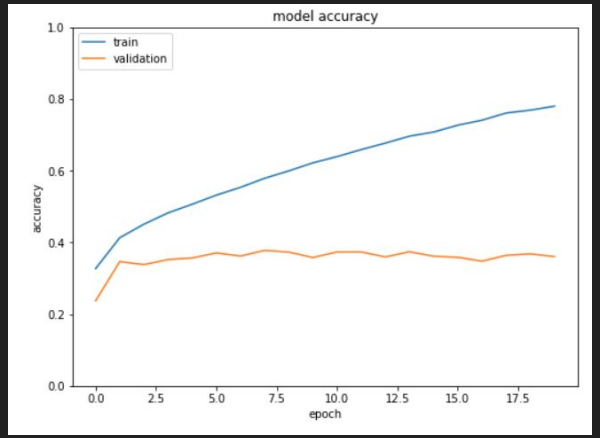
Train Accuracy: 0.7870

Validation Accuracy: 0.36

Test Accuracy: 0.36

Test F1-Score: 0.28





classification Report:

	precision	recall	f1-score	support
0	0.37	0.42	0.39	705
1	0.60	0.32	0.42	2163
2	0.19	0.27	0.22	405
3	0.49	0.48	0.49	562
4	0.32	0.50	0.39	750
5	0.13	0.22	0.16	193
6	0.06	0.39	0.10	44
7	0.10	0.07	0.09	294
accuracy			0.36	5116
macro avg	0.28	0.34	0.28	5116
weighted avg	0.43	0.36	0.37	5116

Modeling- Neural Network with Mean MFCC Values

This model used a non-convolutional neural network architecture, given the one dimensional feature space.

This model did not have the highest accuracy or F1-score, but its training speed was over twenty times faster than the fastest CNN model.

Loss: Categorical Cross-Entropy
Optimizer: SGD
Epochs: 40

Layer (type)	Output Shape	Param #
dense_30 (Dense)	(None, 32)	1056
dropout_21 (Dropout)	(None, 32)	0
dense_31 (Dense)	(None, 64)	2112
dropout_22 (Dropout)	(None, 64)	0
dense_32 (Dense)	(None, 128)	8320
dropout_23 (Dropout)	(None, 128)	0
dense_33 (Dense)	(None, 8)	1032

Total params: 12,520
Trainable params: 12,520
Non-trainable params: 0

Modeling- Neural Network with Mean MFCC Values

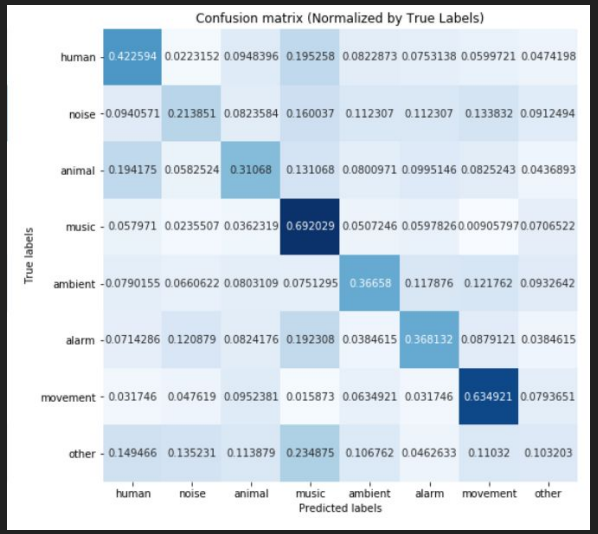
Results:

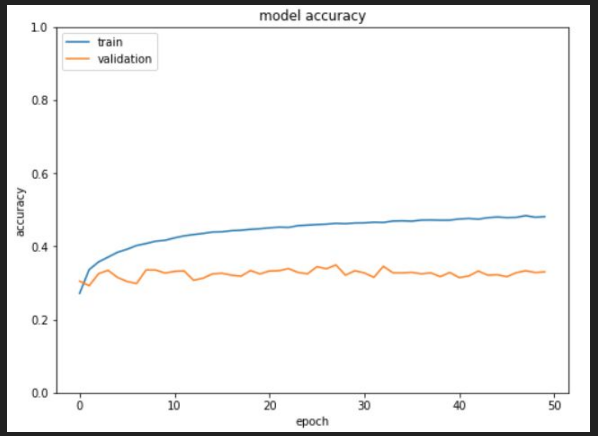
Train Accuracy: 0.4805

Validation Accuracy: 0.3301

Test Accuracy: 0.33

Test F1-Score: 0.29





classification Report:

	precision	recall	f1-score	support
0	0.41	0.42	0.42	717
1	0.73	0.21	0.33	2137
2	0.25	0.31	0.28	412
3	0.35	0.69	0.47	552
4	0.41	0.37	0.39	772
5	0.12	0.37	0.19	182
6	0.07	0.63	0.13	63
7	0.07	0.10	0.09	281
accuracy			0.33	5116
macro avg	0.30	0.39	0.29	5116
weighted avg	0.49	0.33	0.34	5116

Modeling- Logistic Regression with Mean MFCC Values

This model has been implemented to provide a baseline with which to compare the classifiers.

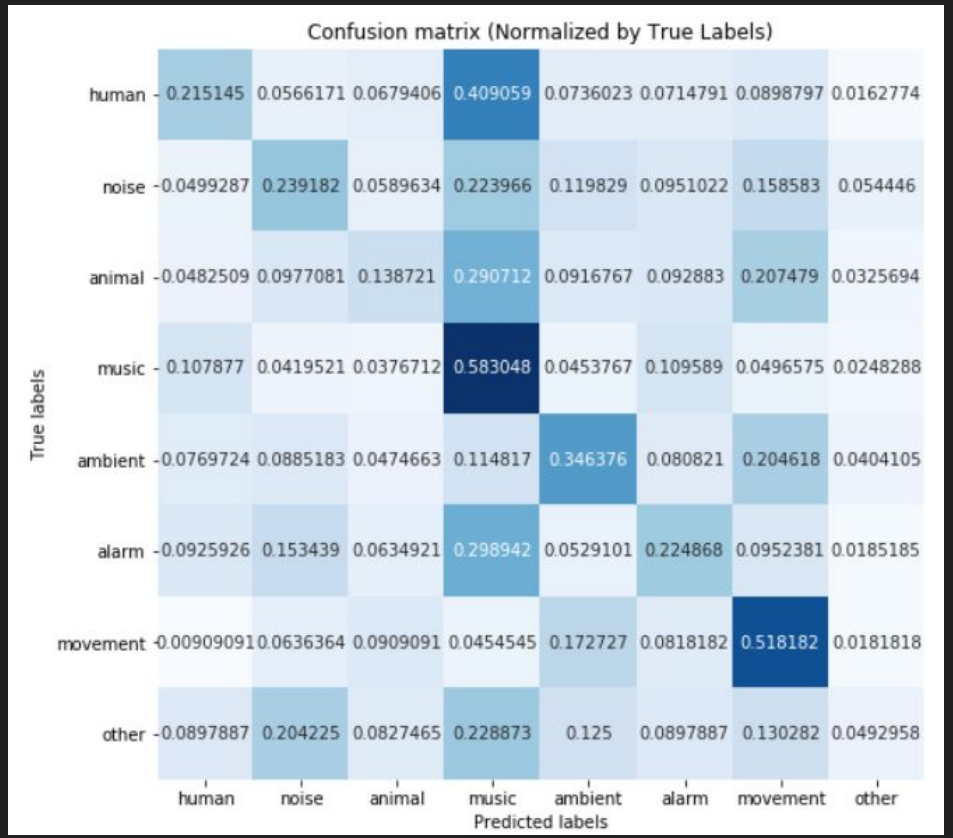
Additionally we will later use the coefficients of this model to derive an approximated feature importance.

Regularization: Elastic Net
L1/L2 Ratio: 0.5
Optimizer: Stochastic Average Gradient Descent
Max Iterations: 500

Modeling- Logistic Regression with Mean MFCC Values

Results:
Train Accuracy: 0.37
Test Accuracy: 0.28
Test F1-Score: 0.22

	precision	recall	f1-score	support
0	0.34	0.22	0.26	1413
1	0.66	0.24	0.35	4206
2	0.17	0.14	0.15	829
3	0.24	0.58	0.34	1168
4	0.39	0.35	0.37	1559
5	0.09	0.22	0.13	378
6	0.04	0.52	0.07	110
7	0.07	0.05	0.06	568
accuracy			0.28	10231
macro avg	0.25	0.29	0.22	10231
weighted avg	0.42	0.28	0.30	10231



Modeling- Random Forest with Mean MFCC Values

This Random Forest model utilized a random grid search for hyperparameter tuning. The random grid search was conducted with fifty iterations with three fold cross-validation.

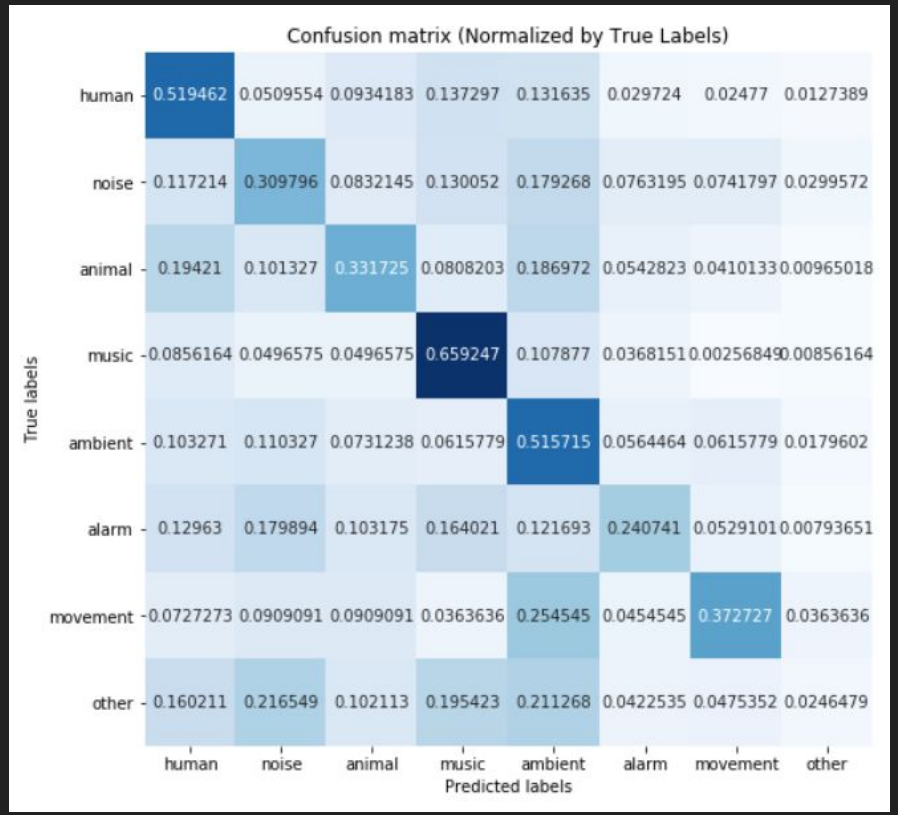
Grid Search: n_estimators: 50-500 trees,
max_features: binary logarithm, square root
max_depth: 5-50 levels

Best Parameters: n_estimators: 476,
max_features: 'sqrt',
max_depth: 41

Modeling- Random Forest with Mean MFCC Values

Results:
Train Accuracy: 0.47
Test Accuracy: 0.39
Test F1-Score: 0.31

	precision	recall	f1-score	support
0	0.41	0.52	0.46	1413
1	0.69	0.31	0.43	4206
2	0.27	0.33	0.29	829
3	0.42	0.66	0.51	1168
4	0.36	0.52	0.43	1559
5	0.14	0.24	0.18	378
6	0.07	0.37	0.12	110
7	0.07	0.02	0.04	568
accuracy			0.39	10231
macro avg	0.30	0.37	0.31	10231
weighted avg	0.47	0.39	0.40	10231



Modeling- Support Vector Classifier with Mean MFCC Values

This Support Vector Classifier utilized random grid search for hyperparameter tuning. The random grid search was performed with 20 iterations with three fold cross-validation. It is notable this was by far the slowest model to tune and train.

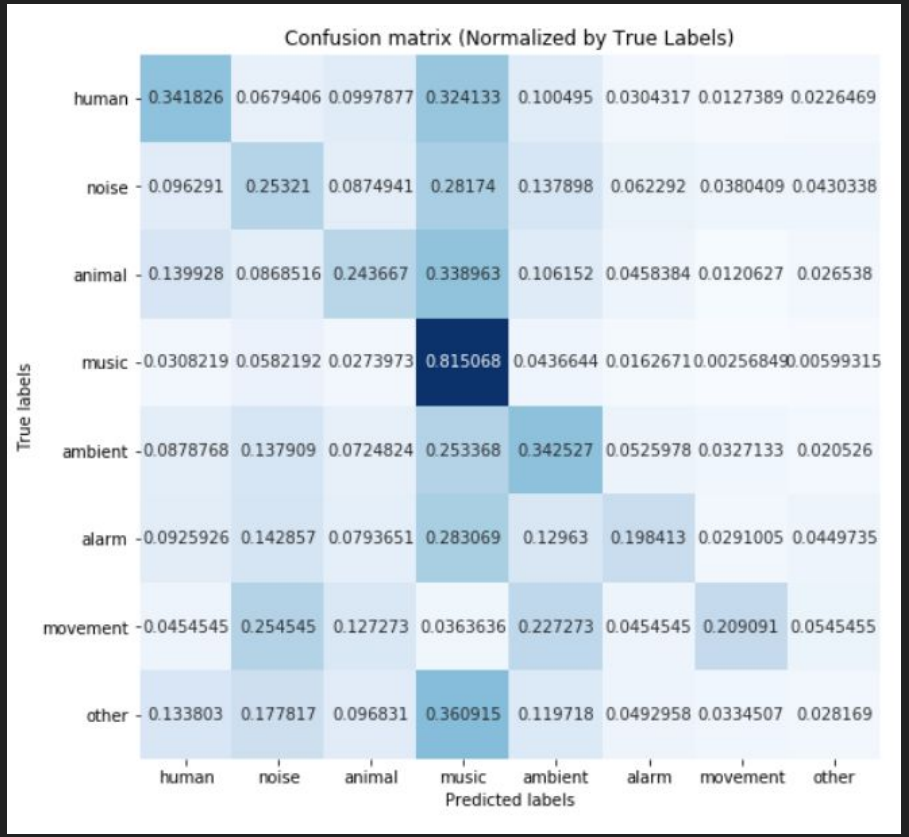
param_distributions = {"gamma": reciprocal(0.001, 1),
"C": uniform(0.1, 100),
"kernel": ['linear','rbf']}

Best Estimator = {"gamma": 0.20592,
"C": 75.0634,
"kernel": “radial bias function”}

Modeling- Support Vector Classifier with Mean MFCC Values

Results:
Train Accuracy: 0.41
Test Accuracy: 0.33
Test F1-Score: 0.32

	precision	recall	f1-score	support
0	0.37	0.34	0.36	1413
1	0.63	0.25	0.36	4206
2	0.21	0.24	0.23	829
3	0.27	0.82	0.40	1168
4	0.35	0.34	0.34	1559
5	0.14	0.20	0.16	378
6	0.08	0.21	0.11	110
7	0.05	0.03	0.04	568
accuracy			0.33	10231
macro avg	0.26	0.30	0.25	10231
weighted avg	0.42	0.33	0.32	10231

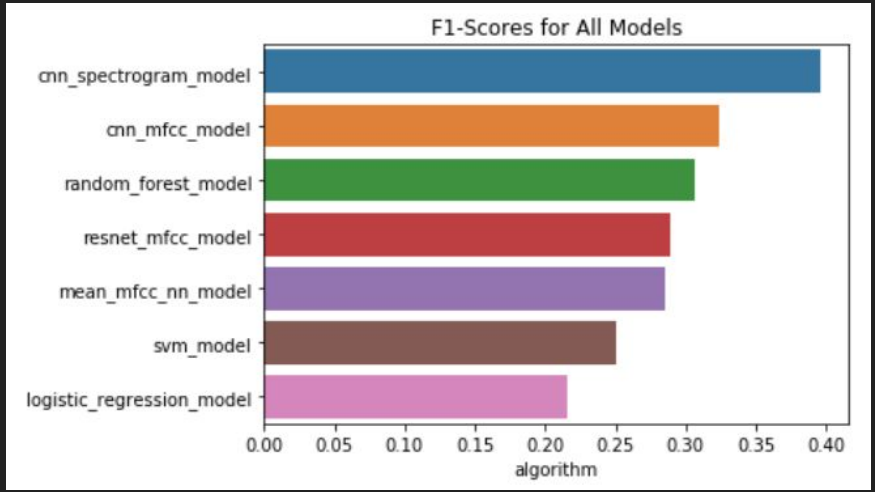


Model Analysis

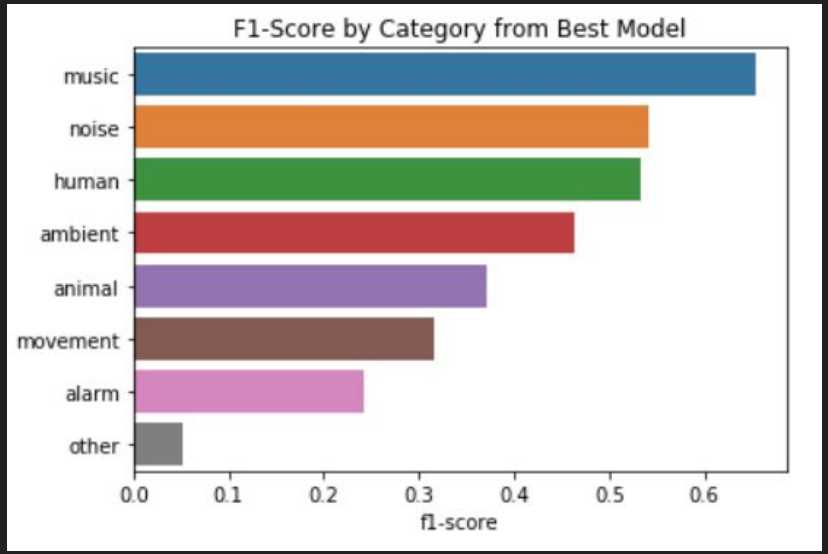
The evaluation metric for the best model we will use is F1-score, as it balances considerations for both false positives and false negatives.

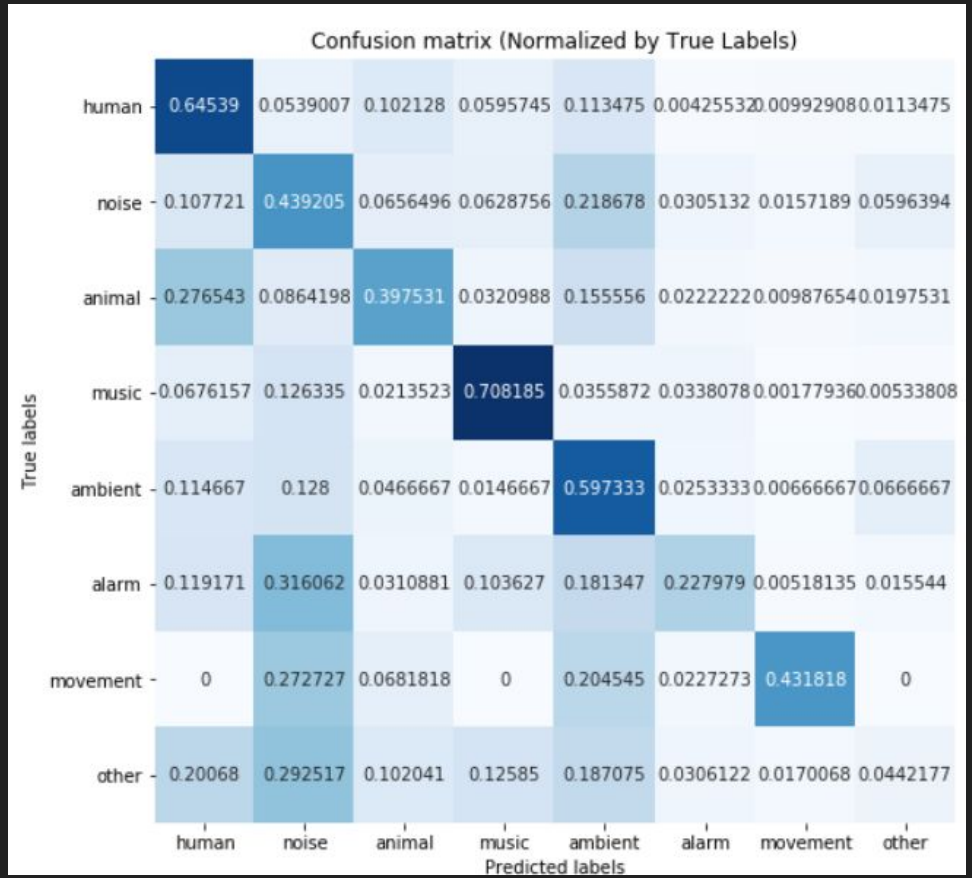
The Convolutional Neural Network trained with Mel-Frequency Spectrograms not only had the highest F1-score, but the highest precision and recall as well.

	precision	recall	f1-score
cnn_spectrogram_model	0.382476	0.436457	0.396434
cnn_mfcc_model	0.328554	0.357161	0.324103
random_forest_model	0.302282	0.371758	0.305980
resnet_mfcc_model	0.292270	0.333644	0.289393
mean_mfcc_nn_model	0.304392	0.388999	0.285756
svm_model	0.261212	0.303996	0.250091
logistic_regression_model	0.249118	0.289352	0.215834

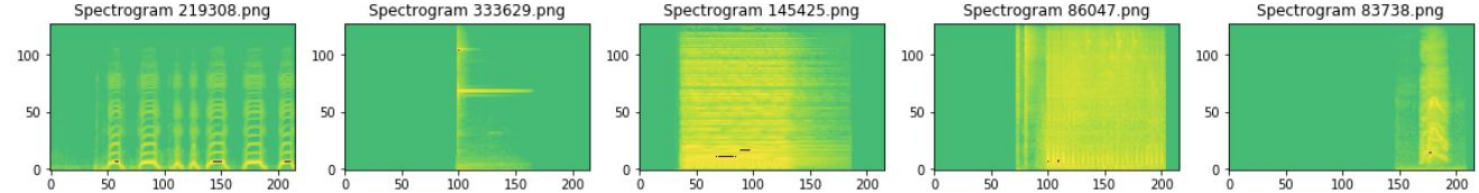


Model Analysis: Best Model





Modeling: What Went Wrong?

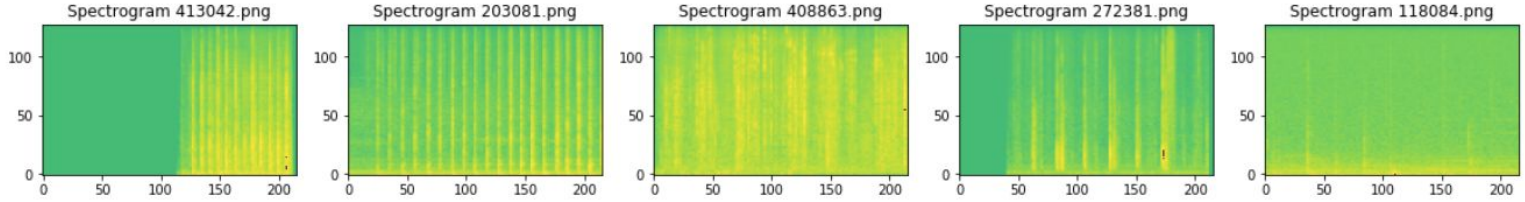


	description	tags	png_name	labels	pred_labels	correct
5508	48 KHz\r\n24 bit	[cell-phone, ring, iphone, vibrate, mobile-phone]	219308.png	other	human	False
7730	A bar chime	[note, chime, chimebar]	333629.png	other	noise	False
10123	A professional quality end/intro/fill effect, ...	[church-organ, film-production, game-developme...	145425.png	other	music	False
6131	machinegun	[machinegun]	86047.png	other	noise	False
5202	Multiple people gasping in horror. Recorded o...	[gasp, group, surpize, walla]	83738.png	other	human	False

Row 10123: the predicted label was music and in fact the track metadata describes it as a church organ. Row 5202: the predicted label is human and the metadata describes the audio as people gasping.

These examples highlight mislabelled tracks where the algorithm actually predicted the correct label but was scored as incorrect.

Modeling: What Went Wrong?

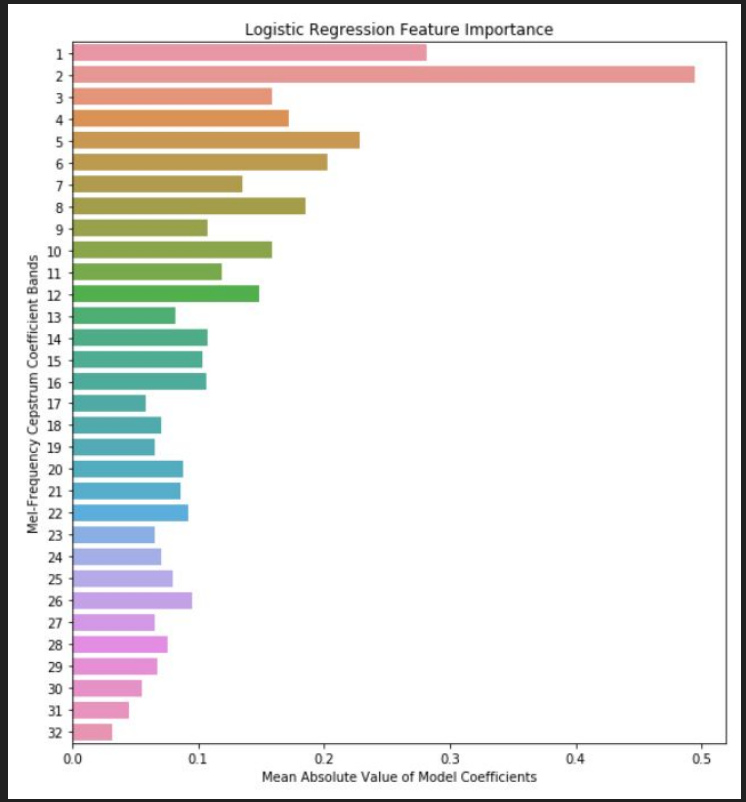


	description	tags	png_name	labels	pred_labels	correct
9822	This sound was created by me for the purpose o...	[fast, concret, running]	413042.png	movement	noise	False
9141	A person running up and down stairs.	[running, stairs, fnd112]	203081.png	movement	noise	False
6753	Sound of someone walking on rocks.	[rocks, owi, gravel, walking, pebbles]	408863.png	movement	noise	False
7228	Footsteps Coming To a Stop	[a, to, stop, footsteps, coming]	272381.png	movement	animal	False
6567	light footsteps	[foot, light, steps]	118084.png	movement	ambient	False

In four of these samples the model did not classify these tracks as movement, but rather as noise and ambient, which is a rather granular distinction. In future model development it was been advantageous to either use different ontological class labels or use data with clearer class distinctions.

Business Recommendation: Feature Importance

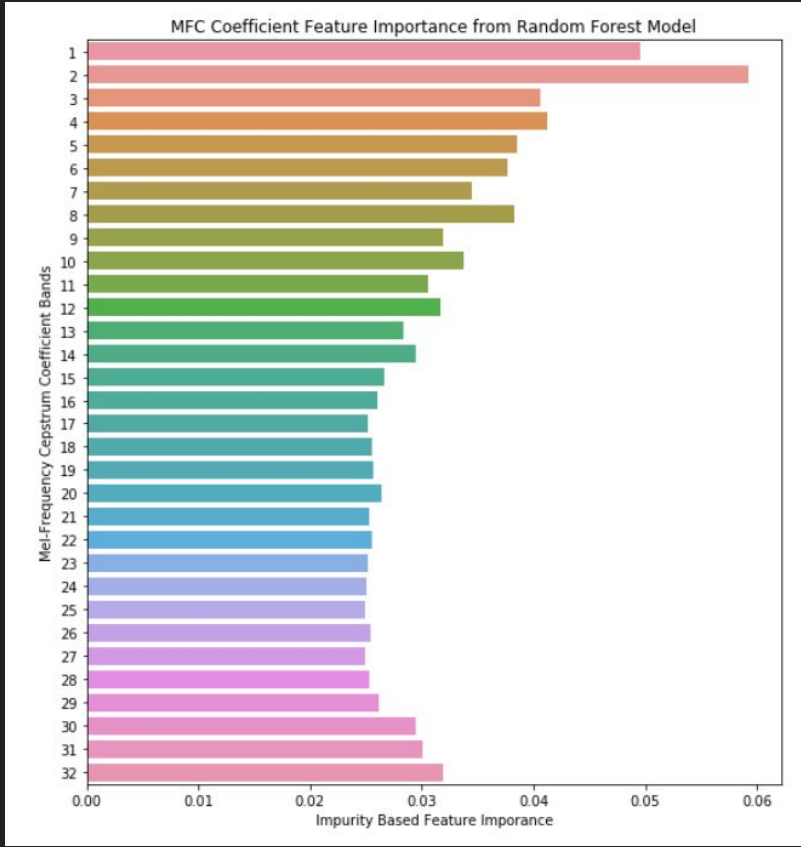
1. Utilize the Feature Importance derived from the Logistic Regression and Random Forest models. In the graph we can see that the first twelve Mel-Frequency Bands have the highest Feature Importance a determined by the mean absolute the model's coefficients. However there still is a significant proportion of Feature Importance in the last twenty features.



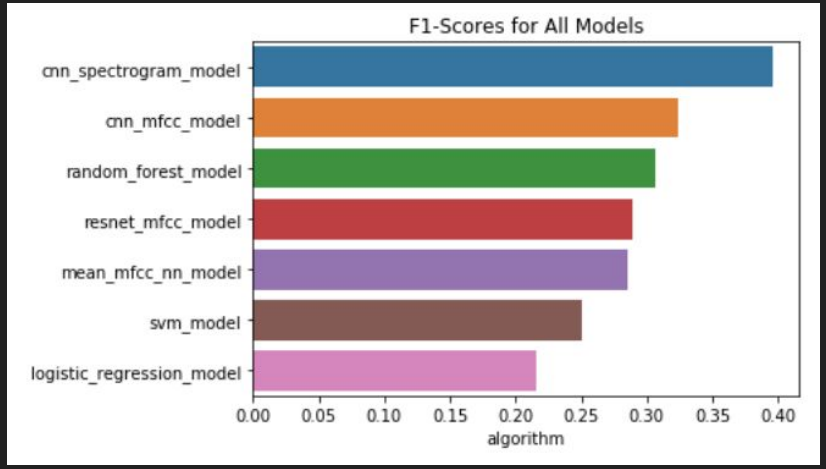
Business Recommendation: Feature Importance

This graph depicts the Feature Importance as determined by each feature's Gini Impurity. Here we see an even higher proportion of Feature Importance in the last twenty Mel-Frequency bands.

Recommendation: Stakeholders interested in training CNN models with MFCC images would benefit from extracting more than the standard 12-13 Mel-Frequency bands, with justification in extracting 32 bands.



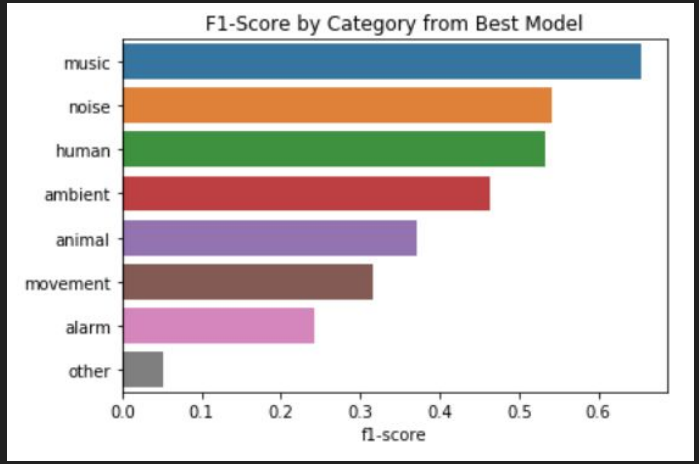
Business Recommendation: MF vs MFCC Spectrograms



2. Model results from this comparative analysis show much higher accuracy and F1-score with the MF Spectrogram features. However, training time was three times slower with the MF Spectrograms.

Recommendation: Stakeholders interested in classifying audio with CNNs would benefit from using Mel-Frequency Spectrograms if training time is not a primary consideration.

Business Recommendation: Retrain as Music Classifier



3. The best model's highest F1-score class was music. This correlates with the fact that the dataset's majority class was music as well. This model still struggled with the minority classes even with over sampling and data augmentation.

Recommendation: Stakeholders interested in higher accuracy models would benefit from re-training this model as a binary classification model to identify music or as a multi-classification model to classify specific musical instruments.

Business Recommendation: Model as Home Security System

3. While this model was benefit from increased training data, the CNN model has the existing architecture to be used as a framework for a home alert system.

Recommendation: Stakeholders interested in home security could build this model into an app that could alert homeowners of unwarranted audio events. For example, human sounds when owner is out of the house, or alarm sounds such as breaking glass at anytime.

Next Steps

Build this classification model into an app where a user could record and input raw audio in real time and receive a prediction for the class of the audio event.

Improve model accuracy by obtaining a larger training dataset or a dataset without multi-label audio events.

Explore different transfer learning models such as VGG-19 and InceptionV3.

The End.

Mateo Martinez
mateomartinez510@gmail.com
linkedin.com/in/mateomartinez510
github.com/mateomartinez510