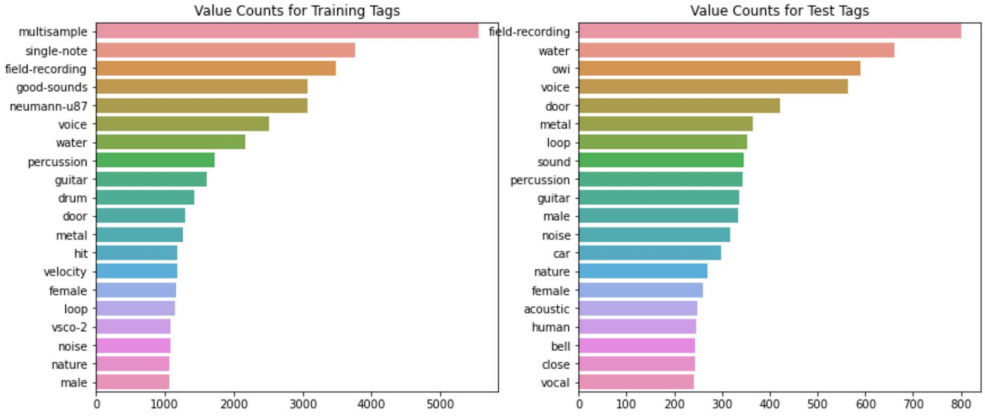
Classifying Household Sounds with Neural Networks

Problem Statement

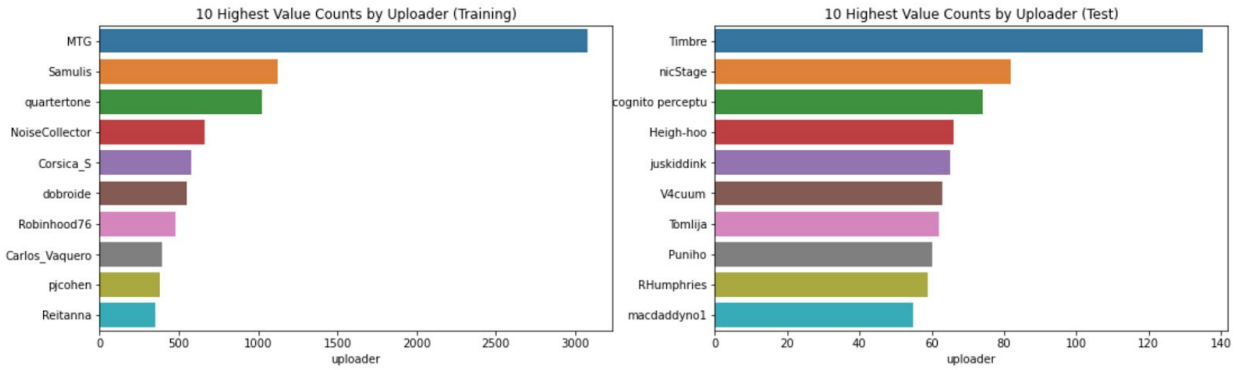
In the last decade there has been significant advancements in image classification machine learning algorithms. For this project, I was interested in applying these same algorithms to build an audio classifier. The dataset we will use for this project is the Free Sound Dataset 50k curated by the Music Technology Group of Universitat Pompeu Fabra. The data curators have already split the data into a training and test set for comparative model analysis. The dataset contains 50,000 raw audio files of a wide range of audio events, ranging from music, human speech, animal, nature, and ambient sounds. The dataset includes metadata with title, description, and tags describing the audio, but does not include class labels to use as a target variable, which will need to be wrangled. The theme of the class labels is categorization into classes of sounds that would be typical to hear in a household environment. The goal of this project was to build a model that can classify household sounds that could serve as a foundation for an app that could alert homeowners of unwarranted sounds, such as human sounds when the owner is at work.

Wrangling Class Labels

The dataset contains over 20,000 unique audio tags describing the 50,000 tracks. The training set has over 16,000 unique tags, the test set has over 10,000 unique tags, but the intersection between the two sets is only 5,609.



Additionally, it is worth noting that there were over 4,000 unique uploaders who contributed to the training dataset and over 2,000 unique uploaders to the test set with no intersection of uploaders between training and test sets. This difference in tags and uploaders between train and test data will lead to model overfitting in the training process and must certainly introduce exogenous factors.



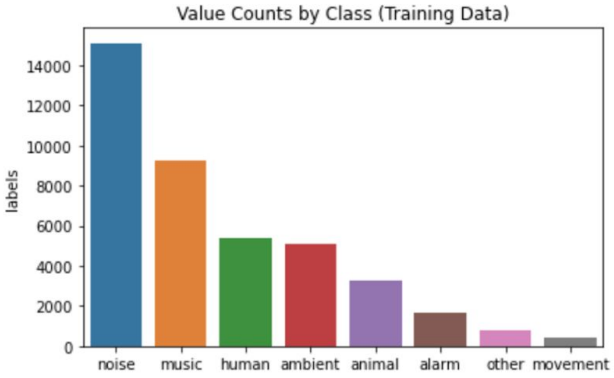
The class labels that will be used for are: 'human', 'noise', 'music', 'ambient', 'animal', 'other', 'alarm', and 'movement.' The tracks were binned into the class groups in a heuristic process of binning the most common audio tags into the class groups and iteratively investigating what tracks have yet to be binned. The ‘other’ category represents the tracks that did not have an apparent class to be binned into.

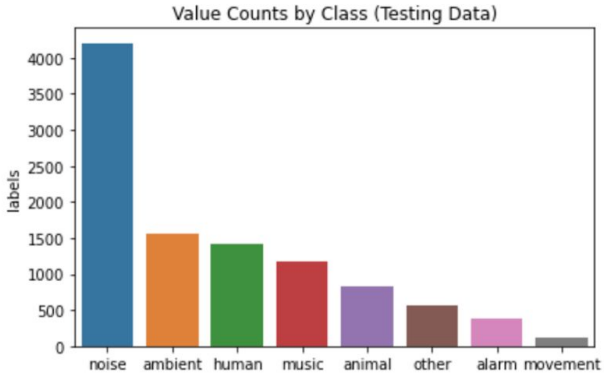
Wrangling Data and Feature Engineering

The main classification models that were used in this project were image recognition models, thus we had to first convert the raw audio files into image files. The first feature set that we extracted were Mel-Frequency Spectrograms, which provide a visual representation of sound by converting the audio into 128 Mel-Frequency ranges that correspond to the range of human auditory perception. All audio tracks are padded/trimmed to 5 seconds with a 23 millisecond sample rate, resulting in a 128x216 image.

The second feature set we extracted were Mel-Frequency Cepstral Coefficient Spectrograms, which 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.

The third feature set is the mean MFCC values, averaged across time, resulting in a one dimensional array of features. This feature set can be used by non-image based classifiers in the modeling process, and will be used for comparative model analysis.

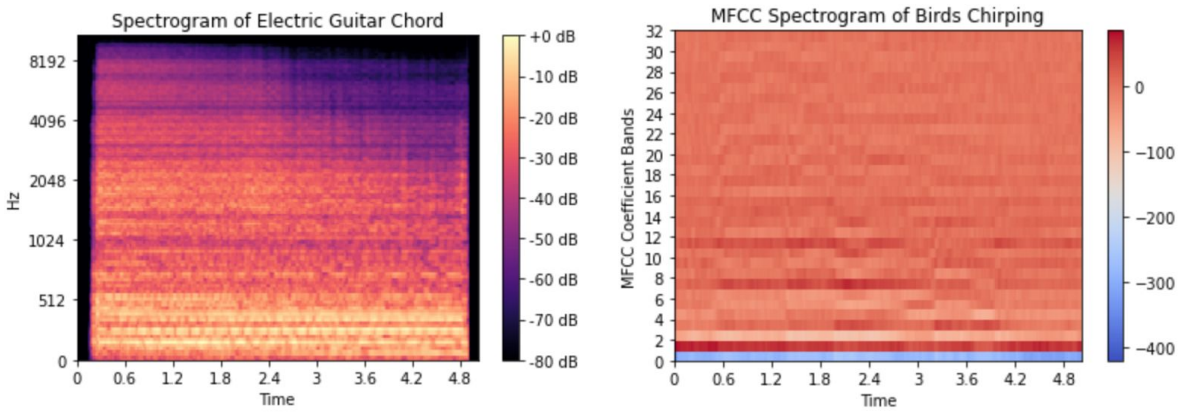


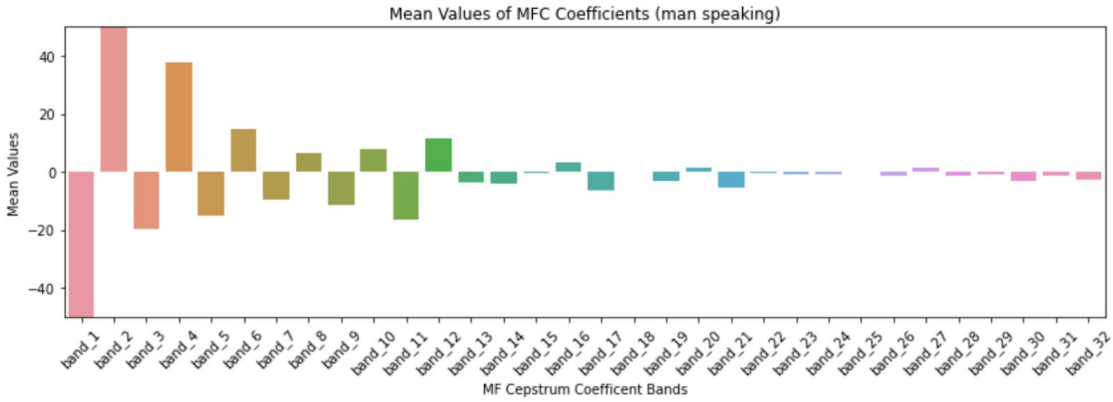


Additionally, after providing class labels to the audio files, it was clear that there were significant class imbalances in the training and test sets. To address this problem in the training set, random under sampling was conducted of the majority classes and random over sampling with data augmentation was conducted of the minority classes. Data augmentation consisted of adding gaussian noise and shifting the pitch of the audio. After resampling each class in the training set had a total of 5,000 samples.

Exploratory Data Analysis

In the EDA, we explored in more depth the audio tags, class labels, and class imbalance in the train and test sets. We will analyze each of the three sets of features as well as the raw audio files. On the left is a rendering of a Mel-Frequency Spectrogram and on the right is a MFCC Spectrogram.





By visual analysis, the left image has significantly more nuance and detail than the right image. The bottom image depicts the mean MFCC values, with much more intense MFCC values in the lower Mel-Frequency bands. This supports the reasoning of only extracting the first 12-13 MFCC bands, but it is also evident that there is still information in the last twenty MFCC bands, which we attempted to derive additional predictive insights.

Pre-Processing

There was only minimal Preprocessing required for these feature sets. The Convolutional Neural Networks required a validation dataset for hyperparameter tuning. To achieve this, we split the test data in half to use as validation data. As is evident in the graph of mean MFCC values, the higher Mel-Frequency ranges have a much lower intensity. To address this, the mean MFCC values were scaled with a zero mean and unit variance. The images files were processed by the Keras ImageDataGenerator to a min-max range of zero to one.

Modeling

Seven models were trained using the three feature sets. The first model trained was a Convolutional Neural Network with Mel-Frequency Spectrogram features. This model proved to be the best model, as determined by having the highest F1-score. The biggest challenge in training this model was overfitting, which was addressed by using a combination of Max Pooling, Batch Normalization, Dropout layers, and relatively low number of filters. The loss function used was categorical cross-entropy with a stochastic gradient descent optimizer trained with 40 epochs.

The second model trained was a Convolutional Neural Network with MFCC Spectrogram features. This model had a relatively similar neural network architecture, with slightly different number of filters and different Max Pooling layers, given the smaller input dimensions of 28x216 rather than the former model's dimensions of 128x216. This model had a notably lower F1-score, but model training ran three times faster than the first model, which could prove to be a significant factor in pushing one of these models into a production environment.

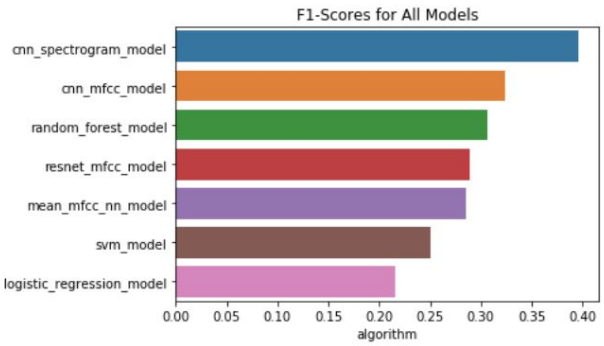
The third model trained was a Convolutional Neural Network with ResNet50 transfer learning with MFCC Spectrogram features. The ResNet50 model was loaded with pretrained ImageNet weights. In initial modeling, the ResNet50 model was trained with the Mel-Frequency Spectrograms, but the training time proved to be prohibitively slow. Even with the MFCC feature set, the ResNet training speed was over ten times slower than the same model without the ResNet layers. In setting up the ResNet model architecture, the trainable layers were unfrozen, and a few Dense and Dropout layers were added to prevent overfitting. This model proved to have worse model results than anticipated. This could be attributed to the fact that the ImageNet model weights were trained with images with different dimensionality than the images used in this model.

The fourth model was a non-convolutional Neural Network trained with the mean MFCC values. This model had a mid-range F1-score in comparison to the other models. However, it

did train significantly faster than any other model, even faster than the Logistic Regression model. The training speed alone should prompt its consideration in future audio classification models.

The final three models were trained with non-neural network models, for comparative analysis to determine the success of the neural networks. The Logistic Regression model had the lowest F1-score. The Support Vector Classifier performed slightly better than the Logistic Regression model, but its training time was the slowest out of all models, which I would consider prohibitive for future audio classification models. The Random Forest model had the third highest F1-score. Below is a graph and table of all the model metrics.

	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

The best model was the Convolutional Neural Network with Mel-Frequency Spectrogram features. In successive order the CNN with MFCC features and the NN with mean MFCC features traded accuracy for model training speed, which depending on the use case, could prove to be more beneficial.

	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

Here is a random subset of incorrectly predicted tracks from the “other” class. In row 10123, the predicted label was music and in fact the track metadata describes it as a church organ. In 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.

Business Recommendations

1. 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.
2. Stakeholders interested in classifying audio with CNNs would benefit from using Mel-Frequency Spectrograms if training time is not a primary consideration.
3. 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.
4. 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 the owner is out of the house, or alarm sounds such as breaking glass at any time.

Next Steps

For further model development, I would 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. I would also improve model accuracy by obtaining a larger training dataset or a dataset without multi-label audio events. Additionally, I would explore different transfer learning models such as VGG-19 and InceptionV3.