Medical Sound Classification

Sukriti Kushwaha (skushwaha@wpi.edu) Thea Caplan (tcaplan@wpi.edu)
Ellys Gorodisch (emgorodisch@wpi.edu) Era Kalaja (ekalaja@wpi.edu)

1 Introduction

This project aims to develop a disease classification model that predicts respiratory disease using audio files and other features. The dataset includes sound recordings, embeddings, and patient demographic information. Each individual has a directory containing their recordings and corresponding embeddings, which train the model for disease classification. Given a set of features, the model outputs 0, 1, or 2, representing the most probable disease type. For privacy reasons, the specific respiratory diseases are censored. While there are three potential classes, it is unclear which value corresponds to which class. Misidentifying a healthy patient as sick, or vice versa, is a more critical error than misclassifying between diseases. This problem is particularly interesting due to the limited data, mirroring real-world scenarios, and its intersection of ML and healthcare. The project could contribute to more accessible, cost-effective, and non-invasive screening methods.

2 Methods

Baseline Model For our baseline model, we used a technique that took the mode of the labels of the training set and used that value as the prediction for each item in the testing set. This ended up giving us an f_{PC} accuracy of about 50%, but an F1 macro score from Kaggle of about 0.09.

Simple Model - Softmax Softmax regression was used as a simple model to create predictions on our dataset. We initially tested our data with a custom implementation of softmax and stochastic gradient descent using the numpy library. At this point, the dataset consisted of 523 features and 546 training samples. The highest F1 macro score achieved at this stage with no data augmentation was 0.47. We also created a Softmax Regression class in Pytorch that applies a linear layer followed by the softmax activation function. We applied k-fold validation to our increased dataset and trained using the Softmax Regression class. Using 5 folds, we achieved an average test accuracy of 78%. This increase was likely due to the fact that our model learned more patterns in the data when we increased the number of samples. K-fold may have also helped our model perform well on unseen data, as the validation accuracy increased each fold.

Data Augmentation / Feature Engineering There were many challenges when it came to modifying the data to improve our results. One challenge is that one of the columns coldPresent in the dataset did not have any value for the first 148 entries, while the rest were either 0 or 1. To solve this issue, we used one hot encoding. Another challenge was that our datasets were quite small, with only 546 entries in the training set and 338 in the testing set. We solved this by doubling the size of the training data, adding one copy of the original training data with noise added to the cough data, age, and packYears. The next challenge was that some of the candidates did not have any form of vowel encodings. To solve this, we removed the candidates from the training data that did not have any vowel data, and we averaged the remaining training data's vowel encodings for the testing data that did not have vowel data. To improve the vowel data, we implemented an extra neural network model to generate more vowel encodings based on the data that we had. We then incorporated more of the cough data into our training. All of the candidates had a cough .npy file, but many of them also had a cough .json file with up to 4 times the amount of data in the .npy file. We added all of the cough encodings for the training dataset by duplicating the rest of the data for each entry that had more than one cough encoding. For the testing set we averaged the .json encodings if they existed, or just used the .npy data.

Data Visualization To visualize PCA, data was separated based on the type of data: ordinal (Figure 1a), continuous (Appendix 1b), continuous without audio, including packs smoked and age (Appendix 1c), and continuous spectrograph data (Appendix 1d). The data was normalized or scaled depending on if the data was ordinal or continuous respectively. As seen in these figures, only the ordinal data and the continuous data without audio were linearly separable via PCA. The other continuous data visualized in PCA was clustered together and not linearly separable.

Deep Neural Networks We started with a deep neural network architecture of four layers. ReLU was used as the activation function, and a dropout rate of 0.3 was applied for regularization. The model overfit as the training F1 score was much higher than the testing F1 score. We then designed a new architecture of five layers, including a larger first hidden layer with 512 units, and applied a higher dropout rate. This model achieved a better F1 score of 0.75. Since, hyperparameter tuning did not improve the performance, we experimented with a Multilayer Perceptron model in PyTorch, which was similar to the previous model except that it incorporated more layers and a higher number of neurons in each layer. Despite modifications this model also did not achieve a higher accuracy. As we continued tuning and trying other architectures, we realized this may be a limitation from our data, since we were only using the sound embeddings and demographic features in our training set. To mitigate the issues of our data we converted our sound files into mel spectrograms which visualize the frequencies of sound as a graph. These were passed to a custom neural network of three connected layers using LeakyReLU activation. This model produced a 0.86 Macro F1 score which is much better than our other deep neural network models. Thus, we learned the mel spectrograms of the sound files were much more powerful indicators.

Table of Results: Model/Technique	**F1 Macro**	f_{PC}
Baseline	0.091	0.518
Softmax (custom) with with cough data	0.469	0.363
Softmax with noise and one hot encoding for coldPresent	0.507	0.6
Softmax with vowel data	0.630	0.627
Softmax with vowel generation	0.673	0.6545
Softmax with more cough data	0.713	0.691
Softmax (pytorch) with k-fold with all data augmentation	0.706	0.773
Multilayer Perceptron with all data augmentation	0.754	0.782
Deep Neural Network all data augmentation	0.768	0.636
Custom Neural Network with sound wave data only	0.862	0.218

3 Conclusions

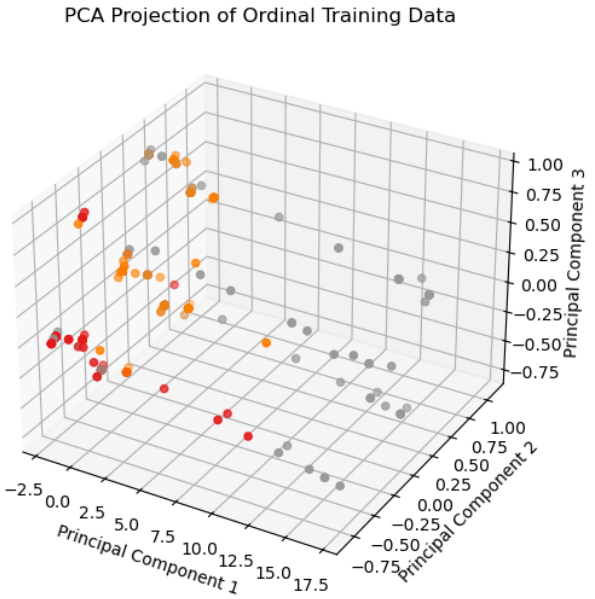
The deep model custom sound classifier achieved the highest Macro F1 score, making it a strong model for this problem. We prioritize the F1 score over f_{PC} because, in medical contexts, balancing precision and recall is crucial to ensuring both high true positive and true negative rates. This helps minimize the risk of misdiagnosis while maintaining reliable predictions. Additionally, we observed that increasing the amount of training data significantly improves prediction accuracy. In cases where data is limited, data augmentation techniques can be leveraged to generate additional samples and enhance model performance.

4 References

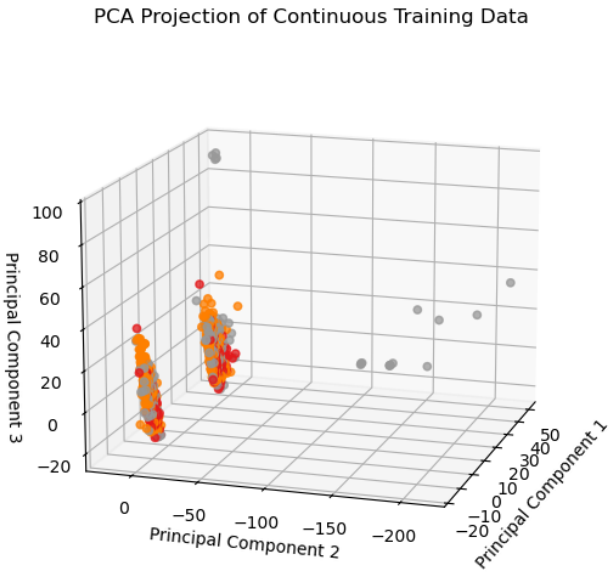
Lewis, C. (2021, April 7). How to Create & Understand Mel-Spectrograms. Medium. https://importchris.medium.com/how-to-create-understand-mel-spectrograms-ff7634991056

Medical Sound Classification Challenge. (2025). @Kaggle. https://www.kaggle.com/competitions/airs-ai-in-respiratory-sounds/overview

5 Figures & Appendices

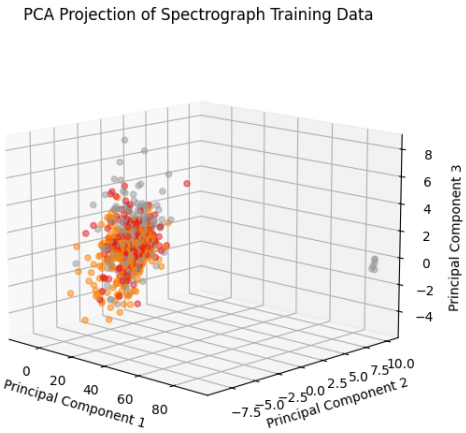


(a) PCA of Ordinal Data



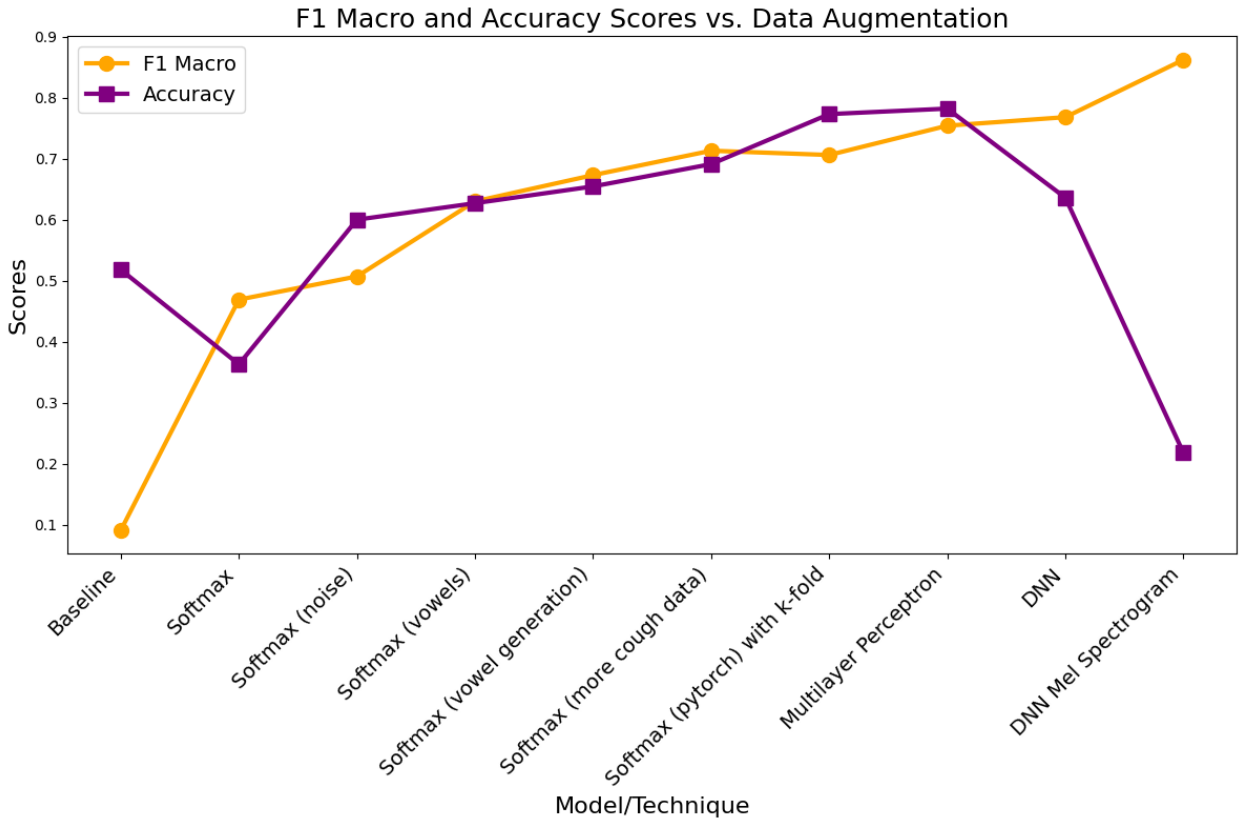
(b) PCA of Continuous Data

(c) PCA of Age and Packs Smoked

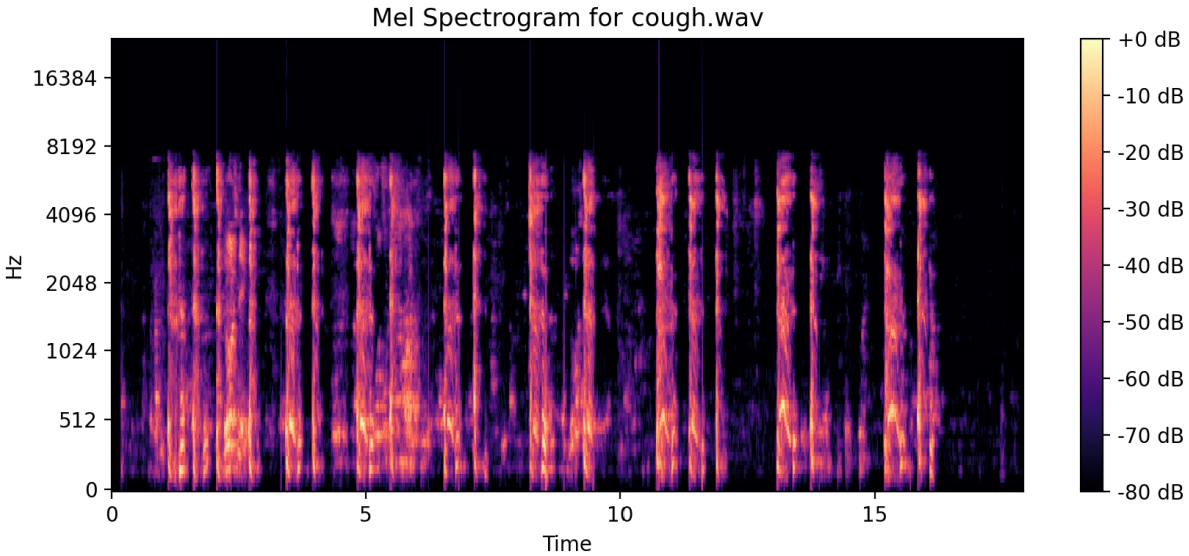


(d) PCA visualization of mel spectrogram dataset

Figure 1: Various PCA visualizations for different datasets



(a) Visualization of change in F1 macro and FPC accuracy metrics through model iterations



(b) Example visualization of mel spectrogram

Figure 2: Other interesting figures