Classification of Cough Acoustics for COVID-19 Detection

Cloe Lu, Nongfeng Wang, Sheng-Lien Lee, Nikitha Reddy Malkannagari
Team 12, Data Science Capstone

December 16, 2023

1 Introduction

The COVID-19 pandemic emerged as a global health catastrophe, with the World Health Organization [1] reporting over 772 million confirmed cases and nearly 7 million deaths worldwide. Spanning from the end of 2019 into the following years, it not only strained healthcare systems but also wreaked havoc on the global economy, with the International Monetary Fund[2] estimating a contraction of 3.5% in global GDP in 2020 alone. In the United States, the economic impact was stark, with millions losing jobs and the economy shrinking by 3.4%. Low-income countries faced even harsher realities, with limited resources to combat the virus and its consequences.

In the wake of such devastating pandemic, the importance of low-cost early pandemic detection became evident. This is where our sponsor comes in, Virufy. Virufy's initiative aims to address this critical need by developing a smartphone application that utilizes cough sounds to detect viral infections. It aims towards making early detection a reality for people everywhere, irrespective of their socioeconomic status.

Our team's collaboration with Virufy centered on the pivotal task of analyzing and classifying cough data. This project delved into diverse classification methods and harnessed the capabilities of machine learning to build a model to classify with a high degree of accuracy, precision and recall. This report unfolds our investigative approach and the promising outcomes we've achieved, in hopes that it will serve as a valuable addition to the collective global effort to preempt and mitigate the impacts of pandemics. With Virufy's initiative, we're part of a movement towards a future where technology plays a key role in maintaining public health, offering a new layer of defense against potential health threats.

2 Dataset

2.1 Dataset Description

Our project's dataset was generously provided by our sponsor, Virufy, and it is composed of two distinct parts. The first part consists of audio recordings—specifically, cough sounds or audio files. The second part of the dataset includes metadata containing personal information about the individuals who provided the cough recordings.

The dataset is derived from two publicly crowdsourced collections. The first source is the Coswara dataset[3], curated by the Indian Institute of Science (IISc) in Bangalore, India. It includes 33 features along with the audio files, totaling 12,094 records. The second source is the Coughvid dataset[4], assembled by the Embedded Systems Laboratory (ESL) at EPFL Lausanne, Switzerland. It has 19 features along with audio recordings, with 10,906 records in total. The attributes provided in the datasets capture the necessary information that could potentially be linked to COVID-19 symptoms.

The metadata included personal details like gender, country, and recording date, alongside COVID-19-related information such as symptom presence and COVID status.Additionally, Virufy enhanced the dataset with derived audio features, crucial for our analysis. These features included the length of the audio, volume levels, clipping ratio (indicating the quality of the recording), discernible cough segments, and background noise levels.

2.2 Data Preprocessing

Within this section, we would convey significant insights regarding to our attempts in cleaning the Coswara and Coughvid datasets, as well as the process by which we generate and select features for the implementation of our models.

2.2.1 Data Cleaning

It was initially challenging to resolve complications related to the data format. The "Submission_date" and "Smoker" indicators were formatted differently in our two datasets. The formats vary since the 'Submission timestamp' for Coughvid was extracted from audio files, whereas the 'Submission_date' for Coswara was initially gathered. Additionally, the 'Smoker' indicator in Coughvid was represented as 'True/False' while in Coswara it was stored as 'Y/N'. These have been standardized to ensure consistency, which facilitates the execution of analyses.

Next, we address the issue of missing columns. Specific symptoms, including 'Fever' and 'Muscle pain,' were classified into distinct columns in Coswara. Coughvid, in contrast, contained simply two columns labeled 'Covid_symptoms' and 'fever_or_muscle_pain.' Therefore, we applied if conditions to the Coswara dataset to filter it and add columns that were identical to those in the Coughvid dataset. To provide a more comprehensive breakdown, we have chosen the columns for 'Covid_symptoms' that correspond most closely with the gathered logics, including 'breathing_difficulties' and 'cough'.

In the case of missing values, we appended the appropriate labels to NaN values. Certain columns, for instance, ought to have true and false labels, but instead display true or false and NaN for the remainder. Therefore, we completed the labeling process by converting the alternative label to the opposite of the current one. We also eliminated rows containing NaN values in columns (e.g., country) that lacked information that could be used. The entire procedure yielded a total of 7426 data samples.

2.2.2 Feature Engineering

Feature engineering, which entails selecting, creating, or modifying features (input variables for model) to enhance the ability of the model to generate precise predictions

or classifications, was undertaken following to the data cleaning process.

At first, we assumed that the "Respiratory condition" column indicated whether or not the sample was afflicted with a respiratory illness unrelated to COVID-19. Upon closer examination of the data, we discovered a significant correlation (0.99%) between the variables "Respiratory condition" and the outcomes of the PCR tests. In particular, the "Respiratory_condition" column can be utilized to identify COVID tests results. It was a clear indication that we must exercise caution in our interpretation of this feature.

Consequently, we introduced an additional column named "Respiratory_illness" to Coswara in order to validate our assumption and optimize the functionality of the attributes. Columns are filtered and identified using related columns, such as 'Pneunomia' and 'Chronic_lung_disease.' We significantly lowered the correlation to 0.08. This value of the new correlation approached the 0.15 reported in the Coughvid dataset.

For the purpose of improving our comprehension of cough patterns and generate cough-related features that are meaningful, we leverage the 'cough_segments' column in our dataset, given that the main focus of our project is about cough audio. We generate the 'number_of_coughs' feature for frequency calculation by counting the number of coughs per sample. We then create the 'cough_length' column, which indicates the duration of each cough, extract the coughing time from a recording to determine the 'total_cough_length' per sample, and compute the'mean_cough_length' by averaging the cough lengths of each sample.

3 Exploratory Analysis

During the exploratory analysis, a thorough comprehension is obtained by examining various components of the dataset through the application of perceptive charts. This section presents a selection of findings that we considered to be of greater significance to our research. They are organized primarily in four categories: Country, Cough Related Features, Audio Related Features, and the PCR test result, which serves as our target variable.

- Country:

  To begin, We began our examination by analyzing the distribution of each dataset individually. The disparity between the country patterns of the two datasets is noticeable as shown in Figure1. Coughvid data originates primarily from European countries, whereas Coswara data is predominantly collected in India. Given that the Coughvid dataset is compiled by a Swiss organization and the Coswara dataset was initiated by an Indian institution, these results are not unexpected.

  A world map is also generated to visually represent positive cases, wherein the sample prevalence is denoted by a color gradient in Figure**??**. A greater quantity of samples is indicated by darker hues, while fewer are represented by lighter hues. It is easy to recognize that the color disparities on the map clearly indicate that the difference between two datasets.

  We further explores cases confirmed by the World Health Organization, revealing that India and France are two of the six countries worldwide with the highest

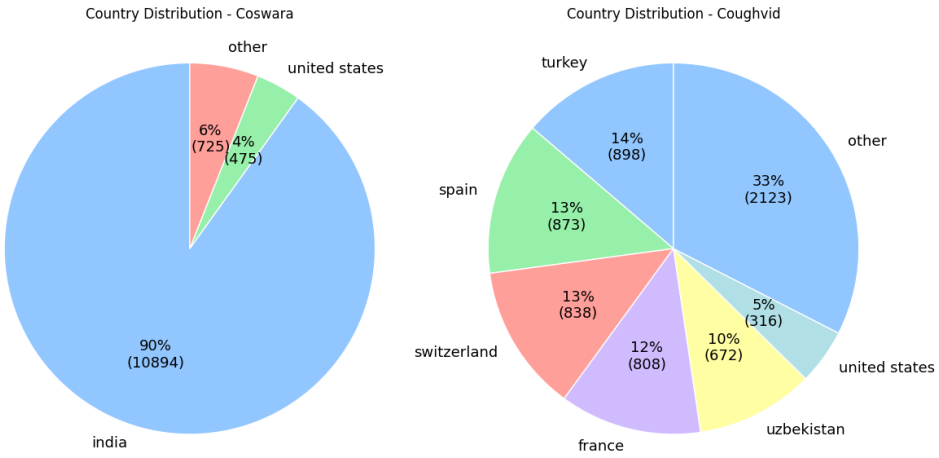

Figure 1: Country Distribution for CoughVid and Coswara Individually

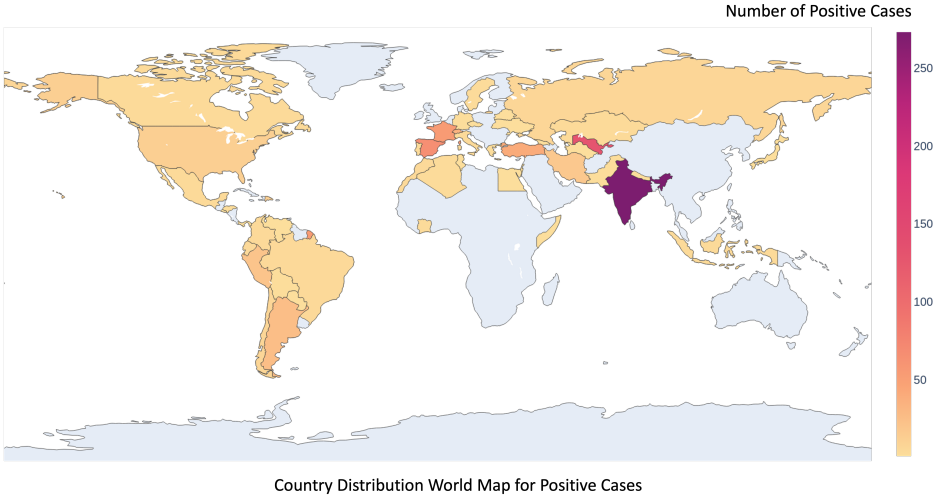

Figure 2: Country Distribution World Map for All Positive Samples

number of positive cases. Although not entirely representative, our dataset does include regions that have a substantial number of confirmed cases.

- Audio Related Features:

  Additional exploration entails applying boxplots to depict audio-related features, including the number of coughs, audio length, volume detected, clipping ratio, and background noise. Due to the existence of numerous outliers in the figures, cautiousness should be taken when taking into account these features for our models.

- Cough Related Features:

  By focusing on the analysis of cough audio, which is the key component of our project, we examine variations in cough counts between positive and negative samples. Nevertheless, the graph illustrates just slight variations in the mean and median values.

- PCR Test Results:

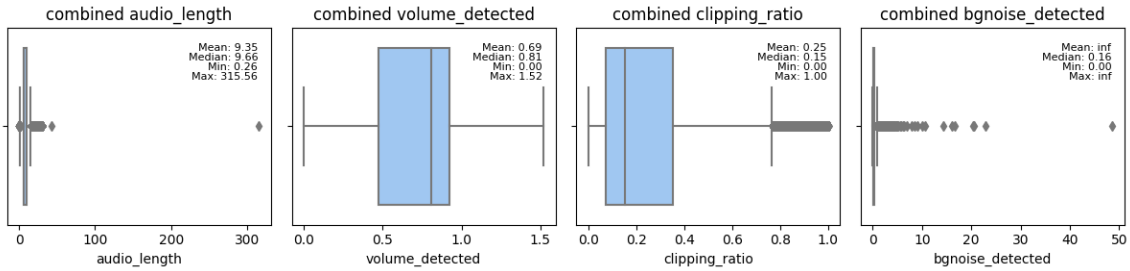

Figure 3: Boxplots of Audio Related Features

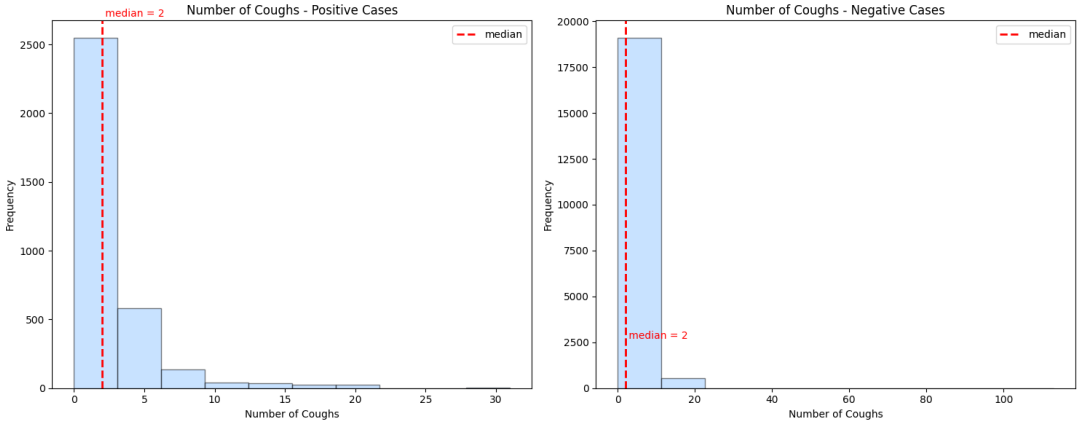

Figure 4: Cough Related Features by Class

Figure5 are pie charts point out the significant imbalance present in our datasets, as positive samples are considerably outnumbered by negative cases. This highlights the crucial significance of mitigating the challenges associated with imbalanced data collection before deploying any model.

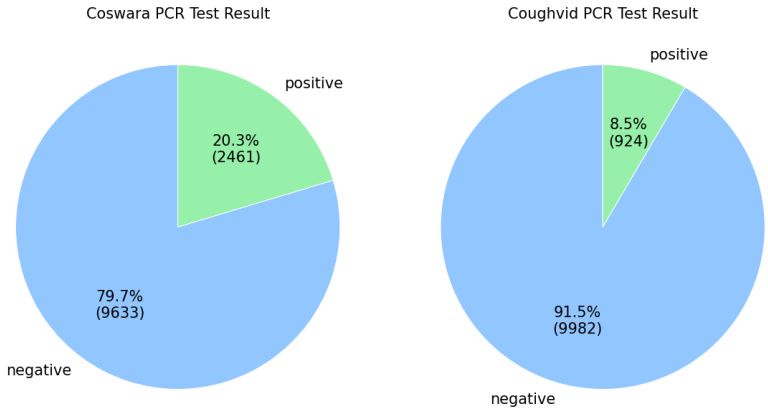

Figure 5: PCR Test Results

4 Model development

Our dataset encompasses a range of categorical features, including country, biological sex, age, and symptoms like fever or muscle pain, as well as a set of audio features. These

audio features consist of original cough sound recordings, along with derived attributes such as cough segments and the count of coughs extracted from these recordings. This section will delve into the methodologies employed for balancing data, preprocessing the audio data and outline the diverse array of deep learning and machine learning models that we have employed, leveraging the various features our dataset offers.

4.1 Handle Imbalanced Data

In addressing the issue of skewed data across features such as country, biological sex, and PCR test results, as mentioned in our data analysis section, we employed various techniques, including Resampling and the Negative Binomial (NB) model to improve the overall usability of our dataset.

4.1.1 Negative Binomial Model

As an extension of the Poisson distribution, the Negative Binomial Model proves useful when the Poisson assumption of equal mean and variance is violated. It enables flexibility in managing fluctuating dispersion and captures the additional variability present in our data effectively.

The simplest method for implementing this model is through R. We made use of the 'glm.nb' function that is included in the 'MASS' package. This function facilitates the fitting of our data to negative binomial regression models.

Figure6 illustrates the probability distribution after executing of the Negative Binomial Model. When contrasting logistic regression with the Negative Binomial Model, our results indicate that the latter more effectively captures the complexity inherent in our unbalanced data.

Figure 6: Probability Distribution for Different Models

4.1.2 Resampling Techniques

- Oversampling:

  Oversampling is a resampling method, addresses class imbalance by augmenting minority class representation. In our dataset, positive instances were multiplied

eightfold, resulting in a final count of 13,284 rows, while 6,682 negative rows remained after data cleaning. This technique enhances the model's understanding of existing positive samples. While promoting a more balanced class distribution, oversampling may introduce redundancy from duplicate minority class instances, potentially reducing informative significance. Figure 7 visually illustrates the mechanics of oversampling and its impact on achieving a balanced class distribution [5].

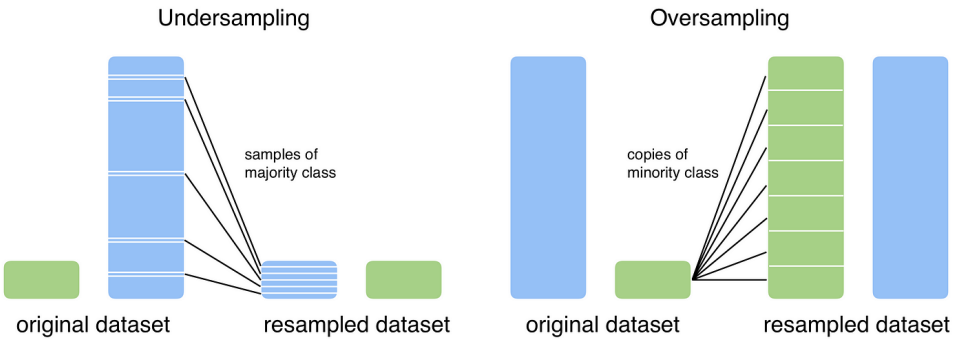

Figure 7: Diagram of Oversampling and Undersampling Comparison

- Undersampling:

  In contrast, undersampling is another resampling method for solving the class imbalance by the reduction of majority class instances (see Figure7). A total of 1568 rows were obtained through our implementation, wherein the number of positive and negative instances was balanced. By preventing the majority class from dominating the model, undersampling involves discarding instances from the majority class (in this case, negative test results), which results in the loss of crucial information where caution is required.

- Balanced by Weight:

  Weight-based balancing is achieved through the assignment of weights to distinct classes so as to influence the model's learning process. We mapped class indices to their corresponding weights in a dictionary and incorporated it as the class weight in our models. By deploying this method instead of the conventional 0:1, 1:1 weight setting (0 represents negative values and 1 represents positive values) that is applied to classes that are more evenly distributed, the model becomes more sensitive to the minority class.

The models that were developed were subsequently updated with the datasets produced by these three resampling techniques. As an instance, undersampling provides the ideal outcomes when applied to our baseline model; therefore, we employ it by default to address imbalanced data prior to model execution. Further elaboration on the model and the intended application of the output data will be provided in the following sections.

4.2 Using Audio related features as input

4.2.1 Data preprocessing for models

- **Mel-spectrogram**

  The Mel spectrogram is a frequency spectrum converted to the Mel scale, designed to mimic human ear perception, as humans detect frequencies non-linearly. To handle audio inputs effectively, the data is transformed into a Mel spectrogram, treating audio as an image for integration into models. Figure 8 shows an example of cough sounds transformed into a Mel Spectrogram [6].

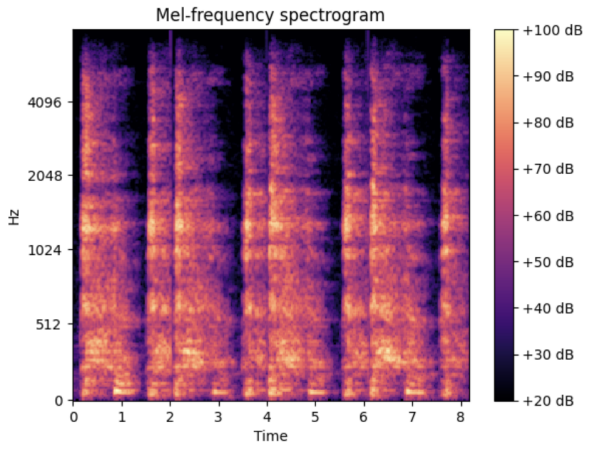

Figure 8: Mel-Spectrogram example

- **Mel-Frequency Cepstral Coefficients (MFCCs)**

  MFCCs is a widely used feature extraction method for sound signals, are pivotal in tasks like speech recognition and music analysis. In our cough sound analysis project, Figure 9 outlines the MFCC technique. In simpler terms, MFCCs are coefficients capturing the power spectrum shape of a sound signal. They're derived by transforming the audio signal to a frequency domain using techniques like DFT, applying the mel-scale for human-like auditory perception, and computing cepstral coefficients from the mel-scaled spectrum [7]. MFCCs are crucial for three reasons: providing a human-like understanding of sound, efficiently reducing signal complexity while preserving essential acoustic characteristics, and their widespread adoption and standardization in sound analysis.

4.2.2 Deep learning models - CNN

Convolutional Neural Networks (CNNs) represent a class of deep learning algorithms predominantly utilized in image analysis and classification tasks. Our project has incorporated a CNN model, originally developed by Virufy, as a primary component of our analysis.

4.2.2.1 Input: Mel-Spectrogram (Baseline Model)

Figure 10 shows the architecture of our baseline CNN, which incorporates the Mel-Spectrogram as input. This model is structured around six blocks, each comprising

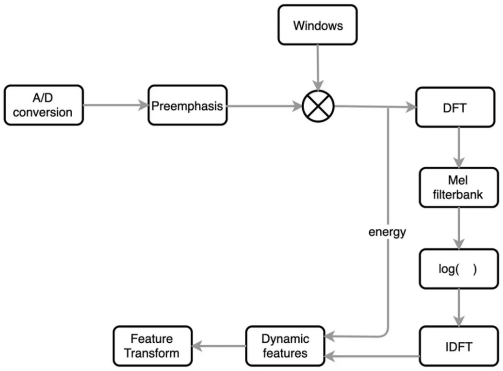

Figure 9: Flow of extracting the MFCC features

multiple convolutional layers, a normalization layer, a dropout layer, and pooling layers. The optimizer employed is Adam, with a learning rate set at 0.002. A sigmoid activation function is utilized for binary classification. The baseline model has recorded a test accuracy of 50% and an Area Under the Curve (AUC) of 0.5.

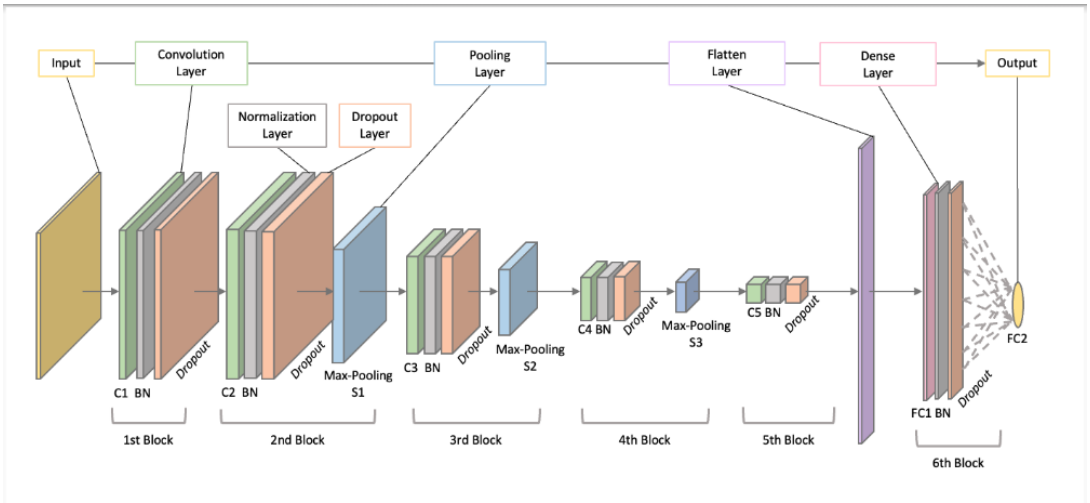

Figure 10: Baseline CNN model structure

4.2.2.2 Input: Mel-Frequency Cepstral Coefficients (MFCCs)

For the second approach, we experimented with Mel-Frequency Cepstral Coefficients (MFCCs) as the input for the CNN model. This structure includes one input layer and three dense layers, maintaining the same parameters as the baseline model. The adoption of this approach yielded a test accuracy of 55.10%.

4.2.2.3 Input: MFCCs and Mel Spectrogram CNN

The third CNN model combines the features used of the previous two models, using both the Mel-Spectrogram and MFCCs as inputs. The architecture of this model can be segmented into three sections: initially, a CNN processes the Mel-Spectrogram images; subsequently, dense layers process the numeric features. Then, we concatenate the outputs of both the CNN and the numeric feature processing layers, passing them

through additional Dense layers and culminating in a final output layer with a sigmoid activation function for binary classification. This model, using the same parameters as its predecessors, achieved a test accuracy of 57.89%, surpassing the baseline model.

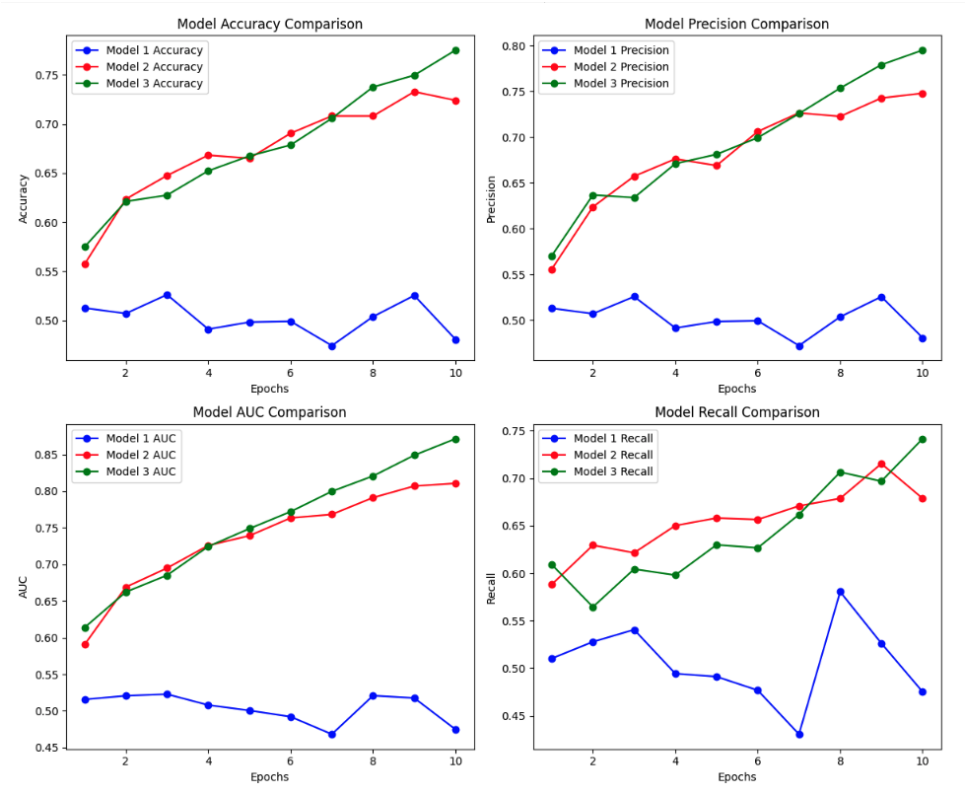

Figure 11: Compare three CNN models performance

Figure 11 includes a series of plots comparing the performance of the three CNN models. We focus on metrics include accuracy, precision, AUC, and recall. These plots clearly show that Models 2 and 3 outperform the baseline Model 1, with Model 3 exhibiting the best performance across all four metrics. Notably, Model 3 demonstrates a consistent increase in performance with more epochs, suggesting effective learning over time. Therefore, among the three Convolutional Neural Network models, the CNN utilizing both Mel-Frequency Cepstral Coefficients and Mel-Spectrogram as inputs (Model 3) exhibits the best performance.

4.2.3 Machine learning models

In this section, we implemented 15 machine learning models, including Logistic Regression, Random Forest, SVM, and K-Nearest Neighbors. We assessed their performance using various inputs, extracting features like 40 MFCCs ('mfcc1' to 'mfcc40') and audio-related features such as 'cough_segments,' 'chroma_stft,' 'rmse,' 'spectral_centroid,' 'spectral_bandwidth,' 'rolloff,' and 'zero_crossing_rate.' Results were compared for all 40 MFCCs, 20 MFCCs, 40 MFCCs with audio-related features, and 20 MFCCs with audio-related features. Findings show that combining 20 MFCCs with audio-related features provided the best machine learning performance. Figures 12 and 13 illustrate the effective results, particularly with tree-based models like XGBoost, Gradient Boosting Classifier, Random Forest, and Gaussian Process classifier using only audio features.

	Model	Accuracy	Precision	Recall	F1 Score
0	GradientBoostingClassifier	0.627389	0.647059	0.560510	0.600683
1	XGBoost	0.624204	0.630872	0.598726	0.614379
2	Gaussian Process	0.614650	0.650000	0.496815	0.563177
3	K-Nearest Neighbors	0.605096	0.617021	0.554140	0.583893
4	ExtraTreesClassifier	0.605096	0.617021	0.554140	0.583893
5	LightGBM	0.601911	0.609589	0.566879	0.587459
6	AdaBoost	0.595541	0.602740	0.560510	0.580858
7	CatBoost	0.595541	0.610294	0.528662	0.566553
8	Random Forest	0.589172	0.601449	0.528662	0.562712
9	SVM (Polynomial Kernel)	0.589172	0.700000	0.312102	0.431718
10	SVM	0.579618	0.598425	0.484076	0.535211
11	SVM (RBF Kernel)	0.579618	0.598425	0.484076	0.535211
12	LogisticRegression	0.573248	0.587786	0.490446	0.534722
13	Decision Tree	0.573248	0.571429	0.585987	0.578616
14	BaggingClassifier	0.560510	0.571429	0.484076	0.524138

Figure 12: Machine learning models performance

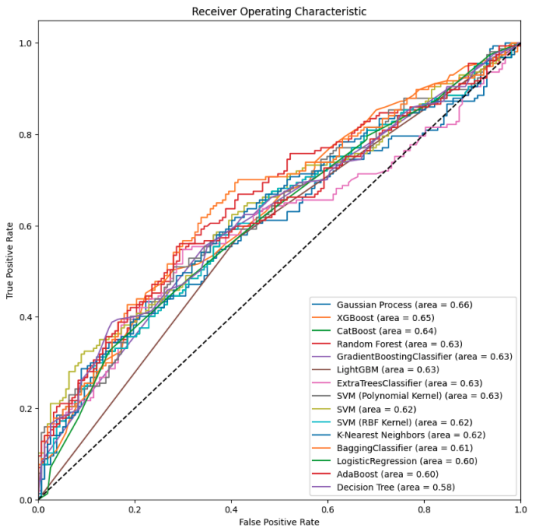

Figure 13: ROC curves of machine learning models

4.3 Using Categorical features as input

The dataset employed in our project encompasses not only audio recordings of cough sounds, but also categorical features such as country, biological sex, age, and symptoms like fever or muscle pain. Even though using categorical features for COVID-19 prediction may not be our ultimate goal, it is still worth considering how these features perform in the model and using them for comparison with results obtained from other input features.

4.3.1 Machine learning models

We applied the same models and parameters as before (using MFCCs), but with different inputs. Categorical features from Figure 14 were used, including 'biological_sex', 'country,' 'covid_symptoms,' 'fever_or_muscle_pain,' and 'respiratory_illness.' The model was trained with 'pcr_test_result_inferred' as the y_label. Figure 14 shows the correlation between these features and the y_label. Notably, 'covid_symptoms' and 'fever_or_muscle_pain' exhibit high correlations of 0.93 and 0.24, respectively, with the prediction label.

Owing to these high correlations, the models we trained demonstrated high performance. The performances results (Figure 15) from the machine learning models show that the top 5 models have a similar accuracy of 0.968 and a precision around 0.98, which is significant. According to the ROC curve (Figure 16), most models exhibit a high AUC. Similar to the models trained with audio features, tree-based models such as XGBoost, AdaBoost, and CatBoost performed well in our project.

4.4 Using the Audio samples and Categorical features as input

4.4.1 VGGish model

In addition to the previously audio feature extraction methods like MFCCs, the state of the art deep learning methods to extract features[8] from our audio files was also perfomed. For that, VGGish model[9] show in 17 was utilized. The VGGish pre-trained model, a variant of the VGG model[10] adapted for audio analysis, employs a deep

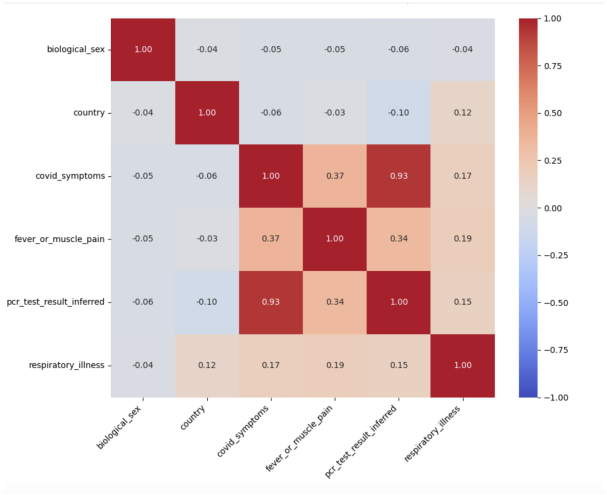

Figure 14: Correlation matrix of categorical features

	Model	Accuracy	Precision	Recall	F1 Score
0	LogisticRegression	0.968153	0.986755	0.949045	0.967532
1	XGBoost	0.968153	0.980392	0.955414	0.967742
2	AdaBoost	0.968153	0.986755	0.949045	0.967532
3	CatBoost	0.968153	0.980392	0.955414	0.967742
4	GradientBoostingClassifier	0.968153	0.980392	0.955414	0.967742
5	LightGBM	0.964968	0.980263	0.949045	0.964401
6	ExtraTreesClassifier	0.964968	0.980263	0.949045	0.964401
7	Random Forest	0.961783	0.967742	0.955414	0.961538
8	BaggingClassifier	0.961783	0.967742	0.955414	0.961538
9	Decision Tree	0.958599	0.973684	0.942675	0.957929
10	Gaussian Process	0.799363	0.801282	0.796178	0.798722
11	SVM (Polynomial Kernel)	0.742038	0.662393	0.987261	0.792839
12	K-Nearest Neighbors	0.735669	0.737179	0.732484	0.734824
13	SVM	0.601911	0.580808	0.732484	0.647887
14	SVM (RBF Kernel)	0.601911	0.580808	0.732484	0.647887

Figure 15: Machine learning models performance

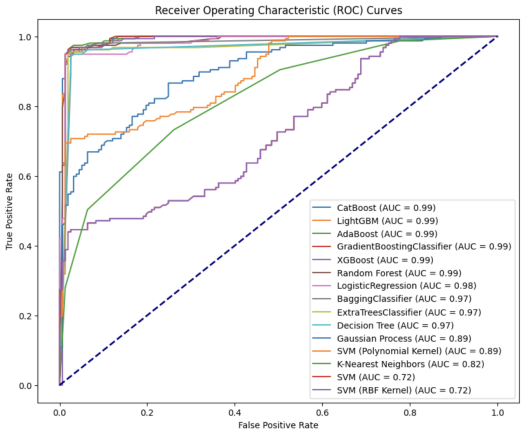

Figure 16: ROC curves of machine learning models

convolutional neural network architecture to convert audio input into 128-dimensional embeddings. Originally derived from the VGG-16 model, which was designed for image recognition tasks, VGGish has been fine-tuned for the acoustic domain to capture a wide range of audio features. It operates on spectrograms of audio signals, utilizing multiple convolutional layers followed by fully connected layers to learn hierarchies of features. From this pretrianed model, we extracted 128 embedding and combined it with categorical features [['covid symptoms', 'fever or muscle pain', 'respiratory illness'] for our classification model. We experimented with different machine learning models as discussed before and achieved high performances. The results (18) from this show that Top 4 models have a accuracy of 0.9681 and Light Gradient Boosting Model has highest AUC of 0.9792. Similar to the pervious method with MFCCs, we see that tree based model works the best here, but it is worth noting that the high correlation of categorical features resulted in better performance of the model.

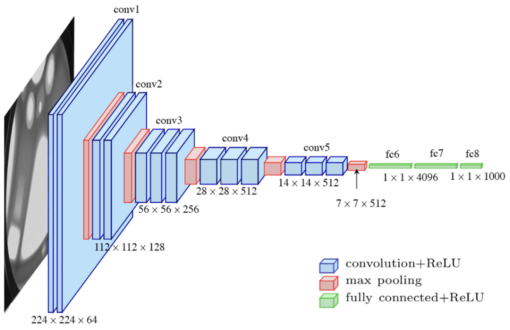

Figure 17: VGGish model structure

	Model	Accuracy	AUC	Recall	Prec.	F1
lr	Logistic Regression	0.9681	0.9723	0.9432	0.9927	0.9673
ridge	Ridge Classifier	0.9681	0.0000	0.9432	0.9927	0.9673
lda	Linear Discriminant Analysis	0.9681	0.9775	0.9432	0.9927	0.9673
lightgbm	Light Gradient Boosting Machine	0.9681	0.9792	0.9432	0.9927	0.9673
knn	K Neighbors Classifier	0.9647	0.9727	0.9432	0.9858	0.9639
rf	Random Forest Classifier	0.9635	0.9730	0.9386	0.9881	0.9627
gbc	Gradient Boosting Classifier	0.9612	0.9700	0.9432	0.9790	0.9606
et	Extra Trees Classifier	0.9612	0.9670	0.9295	0.9927	0.9601
ada	Ada Boost Classifier	0.9601	0.9745	0.9432	0.9768	0.9596
svm	SVM - Linear Kernel	0.9589	0.0000	0.9386	0.9794	0.9584

Figure 18: Performance of VGGish model

5 Conclusion

Based on two public datasets and the basic model that Virufy provided to us. We have conducted various ways to test different combinations and feature engineering methods to have insight to improve the classification.

5.1 Data imbalanced

The dataset consisted of approximately 10%. positive samples and a predominant 90%. negative samples after data cleaning. To address this, we employed over-sampling, under-sampling, weighted by balance, and data augmentation. Overall, under-sampling has a better performance.

For further analysis the imbalanced dataset, our team evaluated the negative binomial model and logistic regression to characterize the distribution of our dataset. The analysis revealed that the negative binomial model has a better performance than logistic regression in capturing the distribution of the dataset. The performance is attributed to the model's inherent capability to manage data imbalance

5.2 Model improvement

Our team improves the efficacy of our foundational Convolutional Neural Network (CNN) by leveraging Mel-Frequency Cepstral Coefficients (MFCCs).We conducted a series of experiments, feeding the network with original Mel-spectrograms, standalone MFCCs, and a combination of both. The result shows that integrating the MFCCs with the Mel-spectrograms as a combined input to the CNN improved the model's ability to classify audio samples by 7.89%. on test accuracy. Furthermore, our research encompassed an evaluation of 15 diverse machine learning models, integrating them with various feature engineering approaches.The result of our analysis highlighted the better performance of tree-based models, such as XGBoost, Gradient Boosting Classifier, and Random Forest since their capability in managing non-linear data relationships. In terms of feature selection, our analysis indicated a preference for using a subset of 20 MFCCs (mfcc1 to mfcc20) over the complete set of 40 MFCCs. Furthermore, including the categorical features related to audio samples was found to significantly augment the models' predictive power.

6 Next Step

There is a data imbalanced issue in our dataset. To tackle the issue, we implement under-sampling. However, the result of our model is not good enough. This approach led to an overall dataset size of 748 samples for each class post-processing which is insufficient to train our model. This inadequacy in data volume has impacted the accuracy of our model, highlighting the need for a more robust solution. Therefore, for our next step, we will keep working on dealing with the data imbalanced problem, Our ongoing research has identified that Generative adversarial networks(GAN) have the potential to generate high-quality audio samples to augment our positive dataset to address the data imbalanced issue.

6.1 Generative adversarial networks(GAN)

The basic GAN structure includes two main parts, a generator and a discriminator showing in figure 19. These two parts are trained at the same time against each other. The generator is trained to generate a sample that mimics the real-world distribution of the related dataset. The discriminator learned to differentiate between the sample that is collected from the real-world dataset and produced by the generator. [11]

The goal of the whole GAN model is to train a model that can generate samples that are barely discriminated from the real-world samples.

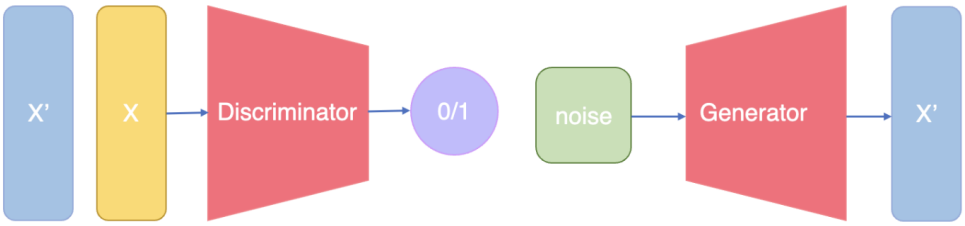

Figure 19: Basic GAN Structure

6.1.1 specGAN

One approach is sepcGAN. It generates audio first creating the spectrogram. A spectrogram is a visual representation showing how the spectral density of a signal varies with time. Therefore, with two dimensions to describe the audio, the spectrogram can capture more detail of the sound.

6.1.2 waveGAN

Another approach is waveGAN. It modifies the structure of the basic GAN to work with one-dimensional audio data. Unlike SpecGAN or other generated methods generate the audio from the feature or abstract representation of the sound. It generates the audio from the audio waveform directly. This allow it to generate more natural sound for human hearing.

For the next step, we think it will be a good choice to implement the specGAN and compare the result with waveGAN which has been implemented by the previous capstone team to reach a better method to deal with data imbalanced.

References

[1] Who coronavirus (covid-19). https://covid19.who.int/. Accessed: 2023-12-12.

[2] The impact of the imf's covid-19 support to developing and emerging economies). https://www.imf.org/en/Publications/WP/Issues/2022/12/17/The-Impact-of-the-IMFs-COVID-19-Support-to-Developing-and-Emerging-Economies-527 Accessed: 2023-12-12.

[3] Coswara dataset. https://github.com/iiscleap/Coswara-Data. Accessed: 2023-12-12.

[4] Coughvid dataset. https://zenodo.org/records/4048312. Accessed: 2023-12-12.

[5] Unbalanced datasets what to do about them. https://medium.com/strands-tech-corner/unbalanced-datasets-what-to-do-144e0552d9cd. Accessed: 2023-12-12.

[6] How to detect covid19 cough from mel spectrogram using convolutional neural network. https://www.analyticsvidhya.com/blog/2021/06/how-to-detect-covid19-cough-from-mel-spectrogram-using-convolutional-neural-netwo Accessed: 2023-12-12.

[7] Mfcc technique for speech recognition. https://www.analyticsvidhya.com/blog/2021/06/mfcc-technique-for-speech-recognition/#:~:text=MFCCs%20are%20used%20to%20represent,spectrum%20of%20a%20sound%20signal./. Accessed: 2023-12-12.

[8] Maria G Campana, Francesca Delmastro, and Elisa Pagani. Transfer learning for the efficient detection of covid-19 from smartphone audio data. *Pervasive and Mobile Computing*, 89:101754, 2023. Epub 2023 Jan 30. PMID: 36741300; PMCID: PMC9884612.

[9] Vggish model. https://github.com/tensorflow/models/blob/master/research/audioset/vggish/README.md. Accessed: 2023-12-12.

[10] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition, 2015.

[11] Adversarial audio synthesis. https://arxiv.org/abs/1802.04208. Accessed: 2023-12-13.