Application of ML to the medical field, classification of patients with lung diseases

C.D. Nguyen[a], D Fremming[a]

[a]*Department of Informatics, UiO, Oslo*

Summary

This project investigates the application of machine learning (ML) techniques to classify patients with lung diseases. ML plays a crucial role in healthcare by processing medical data and providing insights for medical professionals. Respiratory sounds are vital indicators of lung health and disorders. This study explores various methods for detecting pulmonary diseases through the analysis of a synthetic dataset.

Sound files (.wav) are preprocessed and classified using logistic regression, k-nearest neighbor, and support vector machines to determine the specific type of disease. The study employs a voting classifier to evaluate and compare the effectiveness of these methods. Findings from this project contribute to the advancement of pulmonary disease classification through the analysis of respiratory sounds.

1. Introduction

In medical settings artificial intelligence it is used for machine learning models - models which process medical data and provide medical experts with insights from readable patient data. Its role range from clinical decision support to image analysis, where AI assist medical experts in the decision process about treatments, medicaments and other patient needs [1].

In our context, the case for respiratory sounds are interesting since they play a crucial role in healthcare as these are important indicators in diagnostics of both lung health and respiratory disorders. Typical examination methods for diagnosis of pulmonary diseases can range from simple tests such as spirometry, to more advanced medical examinations such as CT scans and bronchoscopy. The latter are invasive to patients, so doctors need to balance diagnostic accuracy with patient comfort and safety. An examination is usually performed by a general practitioner with the use of analogue stethoscope. In modern medicine it has been replaced by digital and image analysis - procedures which can record and generate data.

The data can be represented as sounds emitted when a person breathes. Wheezing sounds and crackles is a common sign of obtrusive airways disease such as asthma or chronic obstructive pulmonary diseases(COPD) [2].

This project explores different methods in the detection of pulmonary diseases through analysis of a synthetic dataset available at Kaggle[2].

We pivot towards sound recording analysis for multiclass classification. The sound files vary in length and quality due to differences in equipment used to record patient respiration cycles. Patient age and health conditions introduce additional inconsistencies in the recordings. To address these imbalances, we apply preprocessing techniques to the training data before implementing our multiclass classification models. After preprocessing the data we apply different machine learning methods involving logistic regression, k-nearest neighbour(kNN) and support vector machines(SVM) to classify which specific types of pulmonary diseases a patient has. To compare the effectiveness of our three classification methods, we implement an ensemble voting classifier that evaluates each model's performance based on accuracy and error metrics. Finally, we discuss our findings and their implications for pulmonary disease classification.

2. Theory

This project begins with an examination of the demographic data set, focusing on determining pulmonary disease. The following sections provide an overview of the theory behind the central algorithms used for multiclass classification.

2.1. Mutli-class Classification

For classifying different types of lung disease, this study uses decision trees, random forest, logistic regression, support vector machines, and algorithms of k-nearest neighbors. The theory of logistic regression is covered in [3].

2.1.1. Decision Trees

In general, decision trees are about breaking down the data and extracting the most informative features. This process is executed sequentially by asking a series of questions and making decisions based on those - the original dataset undergoes splitting, thus creating a tree-like data structure containing the most informative features - features that best describe the target feature[4]. The anatomy of decision trees can be segmented into root node(initial point), interior nodes, and the leaf nodes which contains the final prediction. This system of nodes makes up branches[5].

Decision trees seek to minimize the impurity function, which is a criterion of the probability of misclassification when using Gini impurity:

$$I_G(t) = \sum_{i=1}^{c} p(i|t)(1 - p(i|t)) = 1 - \sum_{i=1}^{c} p(i|t)^2 \qquad (1)$$

Minimizing the impurity function allows the decision tree to maximize the information gain which is central for the splitting criterion or decision:

$$I_G(D_p, f) = I_G(D_p) - \sum_{j=1}^{m} \frac{N_j}{N_p} I(D_j) \qquad (2)$$

Where f is the feature to do the split; D_p and D_j are the parent dataset and jth child node respectively; I is the *Gini* impurity measure; N_p total number objects in the parent nodes; N_j is the number of objects in the jth child node[4]. When using decision trees we must be deliberate when working with features and the data as small changes can produce entirely different trees. We must also pay attention to the possibility of overfitting since decision trees are prone to this problem.

2.1.2. Random Forests

Introducing an ensemble of decision trees we get random forests - a robust classification model with better generalization performance, which is less prone to overfitting. As mentioned, typical decision trees are susceptible to overfitting, sensitivity to features and data, including being susceptible to higher variance. The latter is bypassed by taking the average of multiple deep decision trees that individually suffers from high variance, thus creating a random forest [4]. The random forest algorithm is given by Algorithm 1.

Algorithm 1 Random Forest Algorithm

```
1. Draw a random bootstrap sample of size $n$ from the training data $\boldsymbol{X}$

2. Grow a decision tree from the bootstrap sample.
At each node:
    a. Randomly select $d$ features without replacement.
    b. Split the node using the feature that provides the best
       split according to the objective function, for instance,
       maximizing the information gain

3. Repeat steps 1-2 k times.

4. Aggregate the prediction by each tree to assign the class label by majority vote[4].
```

2.1.3. SVM

A suitable classifier for supervised learning is support vector machines (hereafter SVM), where it is suitable for small- or medium-sized datsets. The underlying mathematics of the SVM relies on hyperplanes and a well-defined margin for separating the classes of variables, making it an easy choice for linear separable datasets [5]. We will not delve into the mathematics of the hyperplanes since many examples only take on 2nd dimensions, where for our use case SVM's role is solely for the multiclass classification. Onwards, we give a brief explanation about the margin. The concept of margin relies on the distance of the margin separating the hyperplane to the closest data points from each class, i.e. the support vectors. We want to maximize the margin r while data points belonging to the classes lies on the correct side of the margins:

$$\max_{\mathbf{w},b,r} \underbrace{r}_{\text{margin}}$$
$$\text{subject to} \quad \underbrace{y_n(\langle \mathbf{w}, \mathbf{x}_n \rangle + b)}_{\text{data fitting}} \geq r, \quad \underbrace{\|\mathbf{w}\| = 1}_{\text{normalization}}, \quad r > 0, \qquad (3)$$

[6]

where $<w, x> + b$ is defined as the hyperplane and x_n and y_n are points in the dataset. For our implementation of SVM we combine it with a linear kernel to introduce non-linearity as we are working with multiclass classification. Rather than having a hard margin constraint, this creates a soft margin that is more flexible, allowing occasional misclassification of data points belonging to the training set[7].

2.1.4. kNN

k-nearest neighbour(kNN) is another prediction method we use which is commonly used for binary classification problems, but due to its flexibility it can be extended to classification problems belonging to mutliclasses. It fit by the following formula:

$$\hat{Y}(x) = \frac{1}{k} \sum_{x_i \in N_k(x)} y_i \qquad (4)$$

where $N_k(x)$ neighbourhood of x defined by the k closest points to x_i in the training dataset[8]. The final prediction is decided by k nearest points, where it is set to the majority of class. As for metrics, it can be either Euclidean or Manhattan distance, and even cosine similarity. Them most commonly used is still Euclidean distance, which also is for our case. While kNN can be used on dataset with features of higher dimensions, it comes with certain drawbacks: For large datasets during the fit, kNN calculates the distance between the selected point to its k nearest neighbours - a task than can turn out to be computationally expensive. The algorithm is also sensitive to assigned values for k: A low k might produce a model which overfits the data, whereas a higher k tend to rely more heavily on averaging values across the codomain

2.2. Dataset

The provided dataset on Kaggle was created by two research teams - one in Portugal and the other in Greece. It is a collection of well-documented sound recordings taken from 126 patients, of which have varying lengths - spanning from 10s to 90s. In total, these recordings comprise 5.5 hours of data, which includes both clear and noise respiratory sounds. "The data contains 6,898 respiratory cycles, including 1,864 crackles, 886 wheezes, and 506 instances with both crackles and wheezes. The patients in the dataset represent all children, adults and elderly - all age groups are represented [2].

2.2.1. Features

The recordings are provided as WAV files of different lengths and respiratory cycles. These variations likely reflect differences in the health conditions and fitness levels of the patients. Due to this inherent imbalance in the dataset, there are concerns about the potential overfitting of the model to the training data. To address this, the sound recordings undergo preprocessing, where longer recordings are trimmed and shorter ones are padded with zeros. Patients with asthma and LRTI were excluded from the dataset due to their significantly lower representation compared to other classes. To address the class imbalance, where chronic obstructive pulmonary disease (COPD) cases dominate the other classes, the sample size was reduced to 600. The multiclass classification focused on six categories: bronchiectasis, bronchiolitis, COPD, healthy, pneumonia, and upper respiratory tract infection (URTI).

Feature extraction of audio signals are plenty. *Trends in audio signal feature extraction methods*[[9]] explains all the state-of-the-art feature extraction methods for their respective area of application. Most commonly used are Mel Spectrocrams, MFCC and Chroma CSTFT.

3. Methods

Our approach to multi-class classification begins with preprocessing of the sound files. Since respiration recordings vary in length between patients, we first analyzed the distribution of recording durations by calculating the difference between start and end times for each patient. We visualized these results using histograms to determine an appropriate standardized length. Based on this analysis, we selected T seconds as our standard duration, corresponding to the upper 90th percentile of the histogram. We selected this duration to capture multiple instances of wheezes and crackles to reduce the risk of missed diagnoses.

After identifying a standard length for our sound recordings we iterated through the recordings, measured the length and trimmed or padded the recordings accordingly where the libraries librosa [10] and soundfile [11] were used for this task.

Previously we mentioned the existence of various methods for feature extraction from audio signals; and we have decided to experiment with MFCC. The features are technically equivalent to images, which means it is also possible to apply computer vision techniques to them. Further, we will flatten the MFCC features such that we can consider every pixel and their value as a feature.

In section 2.2.1 we addressed the issue with class imbalance, hence we utilize Synthetic Minority Over Sampling Technique(SMOTE) to the training data [12].

In order to correctly classify the different diseases, we experiment with five different models, which are Random Forest, Decision Tree, KNN, SVM, and Logistic Regression. For each model, we will perform a grid search to find the optimal hyper-parameters based on its F1-score on the validation set. After we have found the optimal hyper-parameters, we then perform classification on the test set for the final evaluation. This evaluation will be used as a benchmark and comparison to our voting classifier method consisting of the three best performing models alongside their optimal hyper-parameters.

4. Experiments/Results/Discussion

Table 1 shows the different diseases and how many recordings we have of each. As you can see, there are significantly fewer samples of Asthma and LRTI than the others. We have thus decided to prune off those classes for our classification task, such that we only have six different classes to predict from.

Figure 4.2 shows the histogram of the respiratory cycle lengths. Based on the histogram and the fact that we would prefer not to unnecessarily trim or pad audio samples, we decided to go with a fixed size of 6 seconds for each respiratory cycle. This gives us 6898 audio samples of respiratory cycles, where one recording has been split into multiple individual cycles.

Disease	**Count**
Asthma	6
LRTI	32
COPD	662
Bronchiectasis	46
Pneumonia	52
URTI	148
Bronchiolitis	79
Healthy	249

Table 1: The number of data samples for each class.

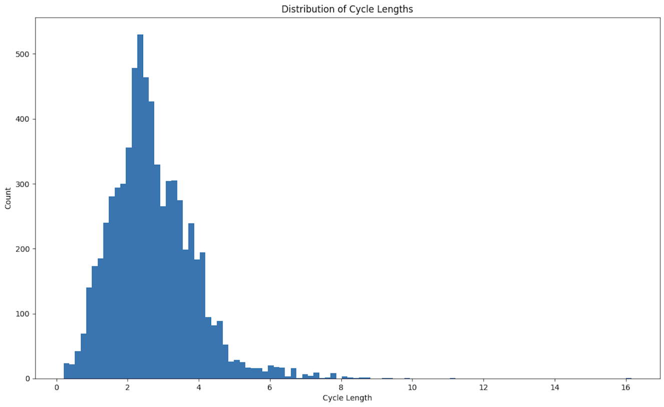

Figure 4.1: Histogram of the lengths of each respiratory cycle.

The plot of a MFCC feature is seen in Figure **??**

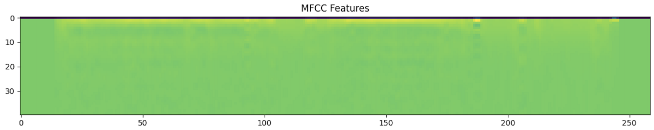

Figure 4.2: MFCC feature plot of one respiratory cycle.

We reduced the number of COPD samples to 600, and performed a SMOTE on our training dataset to increase the sample size of all classes to the same amount.

Following our methodology, we found the best hyperparameters for each model by running a grid search based on the F1-score.

Table 2 shows the complete overview of the models performance on our validation set after fitting to our training set.

Table 2: Highest F1-Score gained from tuning of parameters for different models

Model	**Parameters**	**F1-Score**
Random Forest (RF)	max depth = 17	78%
Decision Tree	max depth = 100	54%
SVM	kernel = linear	74%
Logistic Regression	–	73%
K-NN	N=1	67%

Finally, each classifier have their respective confusion matrix plotted below.

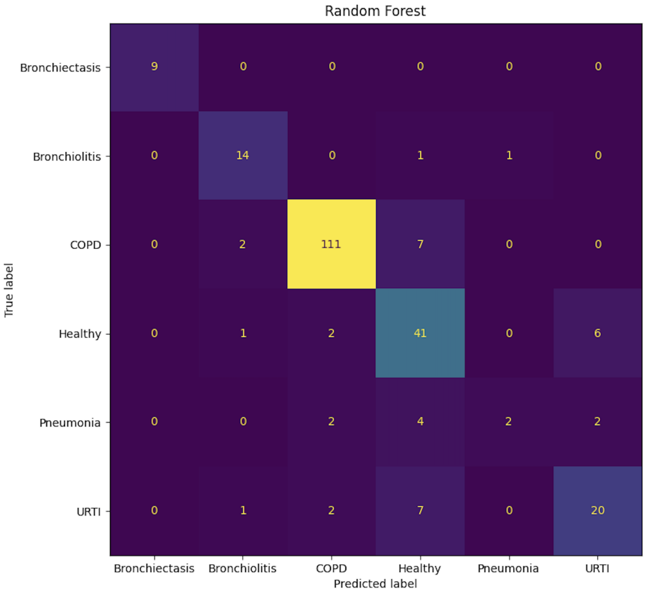

Figure 4.3: Confusion matrix of RF-based classification on the test set.

The Random Forest classifier demonstrated varying performance across different pulmonary conditions. The model exhibited the strongest classification accuracy for COPD, correctly identifying 111 cases. This was followed by successful identification of healthy individuals, with 41 correct classifications. URTI showed moderate classification success with 20 correct identifications, while Bronchiolitis and Bronchiectasis presented 14 and 9 correct classifications, respectively.

The model showed confusion between healthy cases and other conditions, with 7 COPD cases and 7 URTI cases being incorrectly classified as healthy. Pneumonia proved particularly challenging for the classifier, achieving only 2 correct predictions while showing scattered misclassifications across other categories. This suggests that the acoustic signatures of pneumonia may share characteristics with multiple respiratory conditions in our dataset.

The classifier's performance in identifying healthy individuals, while generally robust with 41 correct classifications, showed some limitations with 6 cases being misclassified as URTI. This indicates a potential overlap in the acoustic features between normal breathing patterns and mild upper respiratory conditions. Bronchiolitis classification, while achieving 14 correct identifications, showed some confusion with other respiratory conditions, particularly with one case each being misclassified as healthy and COPD.

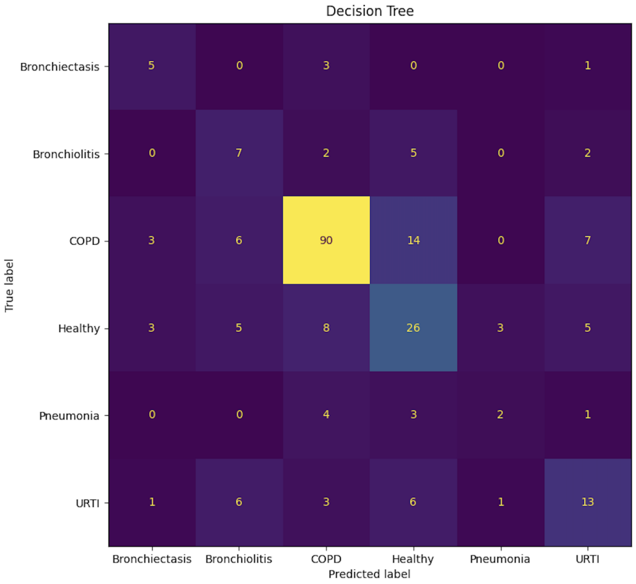

Figure 4.4: Confusion matrix of DT-based classification on the test set.

The Decision Tree classifier showed distinct performance patterns across the pulmonary conditions studied. COPD detection demonstrated the highest accuracy with 90 correct classifications, though this performance was lower than the Random Forest classifier (111 correct classifications). The model successfully identified 13 URTI cases and 26 healthy cases, while showing moderate performance for Bronchiolitis (7 correct classifications) and lower accuracy for Bronchiectasis (5 correct classifications).

COPD cases showed considerable scatter, with 14 cases misclassified as healthy and 7 as URTI. The healthy category demonstrated substantial misclassification spread, with 8 cases incorrectly identified as COPD and 5 cases each misclassified as Bronchiolitis and URTI. The classifier showed particular difficulty with Pneumonia, achieving only 2 correct predictions and displaying scattered misclassifications across other respiratory conditions.

The model's performance in distinguishing Bronchiolitis cases was modest, with 5 cases being misclassified as healthy and 2 cases as COPD. Bronchiectasis classification showed some confusion with COPD (3 cases) and URTI (1 case). URTI classification, while achieving 13 correct identifications, showed notable confusion with other conditions, particularly with 6 cases each being misclassified as Bronchiolitis and healthy.

Compared to the Random Forest classifier, the Decision Tree showed generally lower classification accuracy across most conditions, particularly for COPD (90 vs 111 correct classifications) and healthy cases (26 vs 41 correct classifications). This suggests that the ensemble approach of Random Forest may be more suitable for capturing the complex acoustic patterns associated with respiratory conditions.

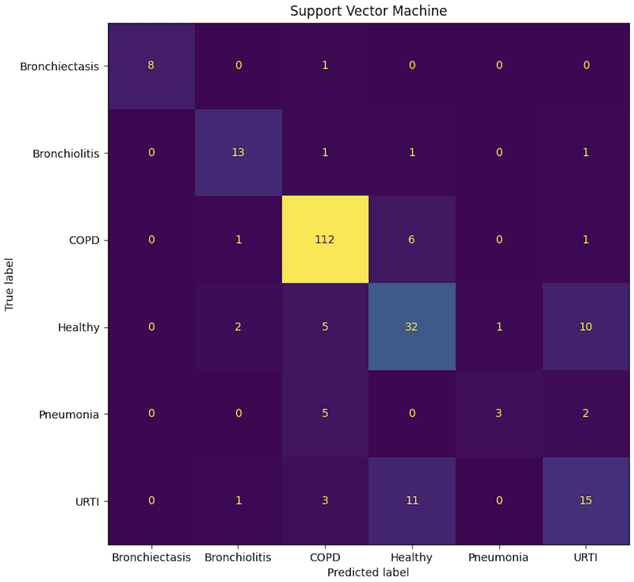

Figure 4.5: Confusion matrix of SVM-based classification on the test set.

The SVM model demonstrated strong performance in identifying COPD, correctly classifying 112 out of 120 cases, which has been the highest so far of the three models mentioned.

In contrast, the model exhibited inconsistencies between the "Healthy" and "URTI" classes, with ten "Healthy" cases being misclassified as "URTI." This misclassification highlights potential overlap in the spectral or temporal features extracted from these conditions, which the current SVM configuration struggles to differentiate. Similarly, the classification of "Pneumonia" proved challenging, with a number of cases being misclassified as "COPD" or "URTI." This indicates a possible similarity in the acoustic properties of these conditions, which may require more discriminative feature extraction techniques or refined modeling approaches to address.

Despite these challenges, the model showed high specificity for "Bronchiectasis" and "Bronchiolitis," with very few misclassifications in these categories. This performance suggests that the MFCC features capture distinct patterns for these diseases, enabling accurate classification. However, the imbalanced performance across classes, particularly the difficulties in distinguishing "Healthy," "Pneumonia," and "URTI", indicates areas for improvement. The use of additional features or domain-specific transformations, such as incorporating time-frequency representations or non-linear feature selection methods, may enhance the separability of these classes.

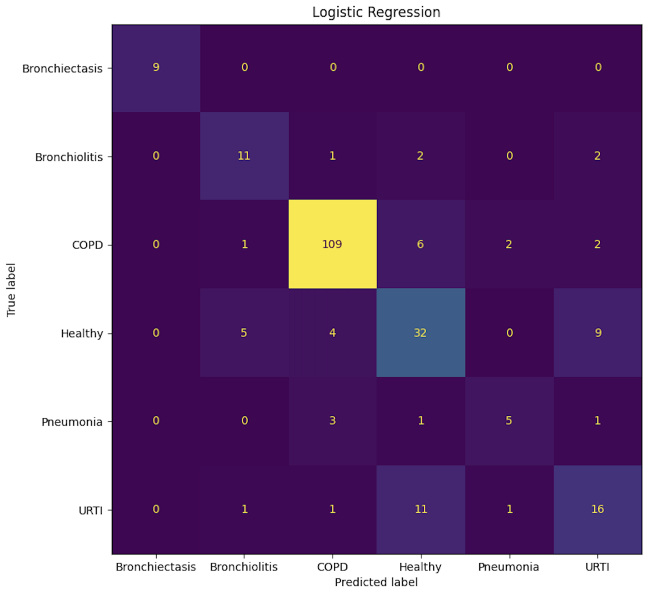

Figure 4.6: Confusion matrix of LR-based classification on the test set.

The confusion matrix for the Logistic Regression model reveals a similar pattern of performance to the SVM model, with some differences in the classification outcomes. The model shows high accuracy for "COPD," correctly classifying 109 out of 120 instances, indicating that the MFCC features effectively capture the distinctive characteristics of this condition. However, it struggles with certain other categories, particularly "Healthy" and "URTI," where a significant overlap in features results in frequent misclassifications. For instance, nine "Healthy" cases were misclassified as "URTI," reflecting difficulty in differentiating these two classes. This has also been the case for all of the previous models, and seems to be one of the bigger challenges for this problem.

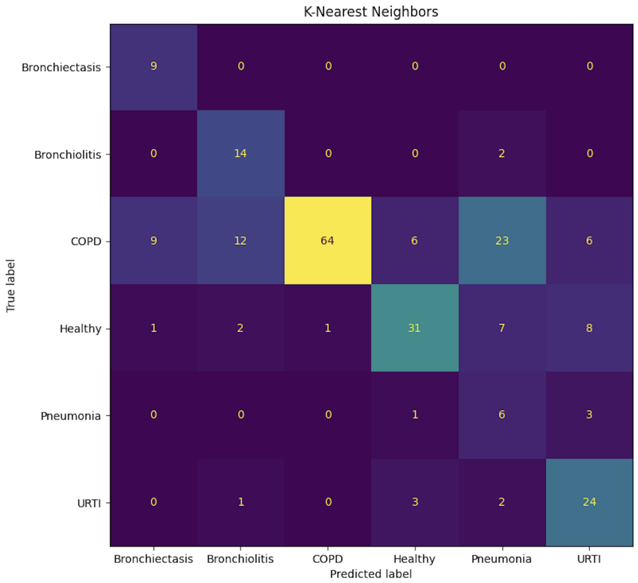

Figure 4.7: Confusion matrix of KNN-based classification on the test set.

Finally, we have the last model, which is the KNN. One significant thing we can see from the confusion matrix is that it outperforms our other models in terms of correctly classifying the URTI cases with 24 correctly classified cases, whereas the closest one is 20, coming from the RF classifier.

The cumulation of the results of our different models, seeing that some models have strengths over others in terms of classifying particular diseases, we believe creating a voting classifier model would provide an improvement in our model performance!

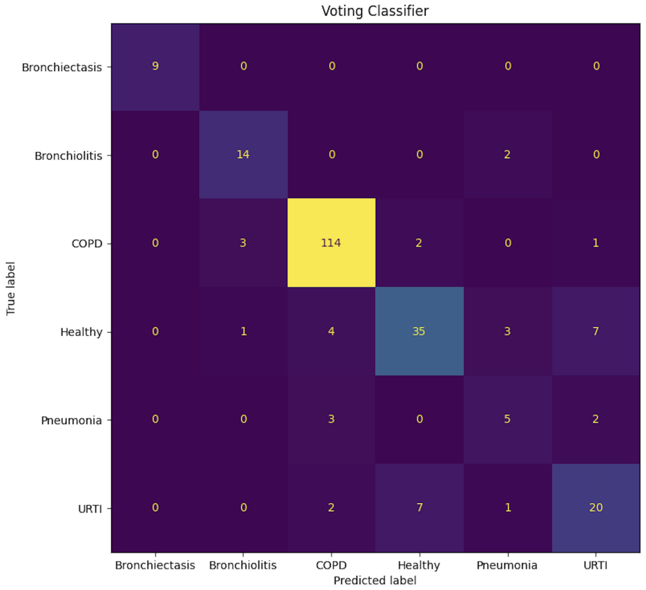

Figure 4.8: Voting Classifier model with SVM, RF and KNN as estimators.

Finally we have our voting classifier consisting of RF, SVM and KNN. The voting classifier seems to outperform the other models in terms of classification for "Brochiectasis", "Bronchiolitis", and COPD. However, it seems to fall short where the other models also struggled, indicating that an ensemble model is not sufficient enough, and that further feature processing or different approaches should be explored. The results are quite promising considering that we are using our other models as a benchmark to discuss the performance of our final ensemble model. Comparing the performance of our Voting Classifier, we have managed to achieve an F1-score of 84%, which is significantly higher than our previous models.

The result also suggests that our preprocessing techniques and techniques to counter the imbalanced dataset issue have worked, considering that we have an acceptable average weighted F1-Score on the unseen samples.

5. Conclusion

This project aimed to classify lung diseases using machine learning models. It was found that high accuracy in models trained on imbalanced datasets did not necessarily indicate reliable results. The researchers used a voting classifier combining SVM, RF, and K-NN for multi-class classification. These models performed the best individually. The resulting model yielded promising results with better performance compared to individual models. The model achieved an average weighted F1-score of 84%. This success suggests that the preprocessing techniques and methods used to address the imbalanced dataset were effective. Future research in this area could involve exploring other feature extraction methods and incorporating additional features into the model, such as combining demographic data with audio features.

References

[1] IBM: *How is artificial intelligence used in medicine?* https://www.ibm.com/topics/artificial-intelligence-medicine.

[2] Wijayasingha, Lahiru Nuwan: *Respiratory sound database*, 2018. https://www.kaggle.com/vbookshelf/respiratory-sound-database.

[3] Daniel Fremming, Andreas Isene, Christian Dahn Nguyen: *Linear and neural approaches to regression and classification: A comparative study*. 2024.

[4] Raschka, Sebastian, Yuxi(Hayden) Liu, and Vahid Mirjalili: *Machine Learning with PyTorch and Scikit-Learn*. Packt Publishing, 2022.

[5] Hjorth-Jensen, Morten: *Computational physics, lecture notes fall 2024*. Department of Physics, University of Oslo, August 2024. https://compphysics.github.io/MachineLearning/doc/LectureNotes/_build/html/intro.html.

[6] Deisenroth, Marc Peter, A. Aldo Faisal, and Cheng Soon Ong: *Mathematics for Machine Learning*. Cambridge University Press, 2019.

[7] Bishop, C.M.: *Pattern Recognition and Machine Learning*. Springer, 2006.

[8] Hastie, T., R. Tibshirani, and J. Friedman: *The Elements of Statistical Learning*. Springer, 2009.

[9] *Trends in audio signal feature extraction methods*, January 2020. https://www.sciencedirect.com/science/article/abs/pii/S0003682X19308795?casa_token=8hRkavweyycAAAAA:v0Sm5lDJ1Reb40IOIs9v1tb6WRXxbZkPu8kw31vk-OvN_4-quWj4LUbzse74LHgMRpEgcrIGTlX3.

[10] $\mathit{librosa.util.pad}_c\mathit{enter.}$, visited on 2024-27-11.

python-soundfilepython-soundfile. https://python-soundfile.readthe
visited on 2024-27-11.

SmoteSMOTE. https://imbalanced-learn.org/stable/references/gene