The ARDS Prediction via Machine Learning

Brihat Sharma, Neha Goel, Paul Montegrande, Saharsh Patel

Loyola University Chicago
Dr. Dmitry Dligach

December 16, 2017

Abstract

Acute Respiratory Distress Syndrome dataset was provided from Loyola Medical Center. The data set consisted of patients with and without diagonised by the medical condition. Machine learning algorithms such as Stochastic Gradient Descent and Decision Tree was used to flag the patient with Acute Respiratory Distress Syndrome. The accuracy score was 0.87 using Decision Tree Classifier and the average F-1 score for both positive and negative class was also 0.87, whearas for the positive class it was 0.68.

I. Introduction

Acute Respiratory Distress Syndrome (ARDS) is a severe medical condition, which can be life-threatening if proper attention is not provided on time. The lungs transmit oxygen that is carried by the blood towards the heart where the blood can be oxygenated and circulated throughout the body. A person with ARDS prevents the necessary reaction with the blood and oxygen in the lungs. Lungs are inflamed, and the bronchioles get constricted which prevents the oxygen from reacting with the blood [2]. Eventually, kidneys and brain stops functioning due to oxygen deprivation. A person with ARDS has a hard time breathing, but also if it is undiagnosed and unnoticeable they have a hard time clotting wounds and injuries [1]. This is because oxygen in the blood is used to produce the white blood cells that help mend the open wounds.

There is a 20 to 50% chance of mortality when having this ailment and any age can obtain this malady [1]. Even children can be infected by such. They are more likely to die from an intense onset of respiratory inflation. The causes of ARDS are normally stem from large amounts of fluid entering the lungs due to a type of trauma, but infections in the lungs and other lung related injuries can cause them as well [1]. Doctors are still unsure on exactly what triggers this medical condition, and by the time they could figure out if the person is diagnosed by ARDS or not, it may already be too late to save the person. Therefore, there is such a high mortality rate.

II. Dataset Description

ARDS dataset was provided by Loyola Medical Center, which was collected from patients with and without the medical condition. These data files were pulled from doctors′ forms and converted into symbolic string iterations using cTAKES, and each string represented a symptom. The dataset was given in two different folders, where one folder represented patients diagnosed by ARDS and other without. Each file inside the given folder represented a patient. The dataset was already separated into train and test set. Train set consisted total of 421 files, with 106 ARDS patients, whearas test set consisted of total 106 files with only 27 files as a person diagnosed by ARDS. Using UMLS Methathesaurus Browser, each encoded string

inside the file can be decoded into the respective symptoms to get a better understanding of the dataset [4].

III. Baseline Approach Description

The given dataset consisted of files which represented an individual patient. Hence, each file needed to be classified into either positive class: diagnosed with ARDS or negative class: not diagnosed with ARDS. This was a binary document classification problem, where each document had encoded string representing symptoms of the patient. Initial approach was to read each document, use tokenization to break the document into individual encoded string, and count the number of occurrence of each encoded string. The frequency of each string can be normalized and different machine learning algorithm can be implemented to train the documents, and use the trained modeled on test set to flag the ARDS patients.

IV. Method Description

A pipeline and grid search methods were implemented to efficiently train the model. Pipeline and grid search were made to allow for the flexibility of trying different algorithms for testing the test set and the development set. Pipeline was used to assemble countvectorizer to count the encoded word frequency, TfidfTransformer to normalize the data and classifier as machine learning algorithm so that they can be cross validated together.

Grid search was used to select the best of a family of models, parametrized by a grid of parameters. Grid search was also used to split the data into k-folds, where one of the fold was kept as a development set and k-1 fold as train dataset. Different machine learning algorithms were used to train the train dataset and tested on development set by tweaking the range of parameters provided for the algorithm. This training process was done k-1 times, each time changing the development set into different fold, and average accuracy for each parameter was checked by grid search to find the optimum parameter.

V. Evaluation

The optimum parameter found by grid search was used to test on the test dataset to find the prediction accuracy. For this process, we tried with many machine learning algorithms such as Multinomial Naive Bayes, LinearSVC, SGDclassifier, Decision Tree and many more. For each of the method, we calculated accuracy and F-1 score for both positive and negative class. We choose F-1 score as primary evaluation metric since our test data is skewed with only 27 positive class and 79 negative class.

Some of the algorithms performed poorly such as Multinomial Naive Bayes with zero F-1 score for positive class. LinearSVC calculated F-1 score for the positive class as 0.41, which means out of all the positive class from the test dataset more than half of the prediction was incorrect. SGD classifier and Decision Tree preformed relatively better on the test dataset. We used wide range of alpha value for 0.001 to 1000 by factor of 10 for SGDclassifier to train the model. Our best prediction using the SGD classifier was with loss as squared hinge, alpha value of 0.01 along with l2 regularization. A confusion matrix for SGD classifier is given in the table below:

SGDClassifier Prediction

Actual Value	**p**	**n**
p′	TP 18	FN 9
n′	FP 13	TN 66

Lastly, decision tree was used to train the train dataset, which gave us our best result out of all the algorithms used. The best optimum parameter for the train dataset was criterion as

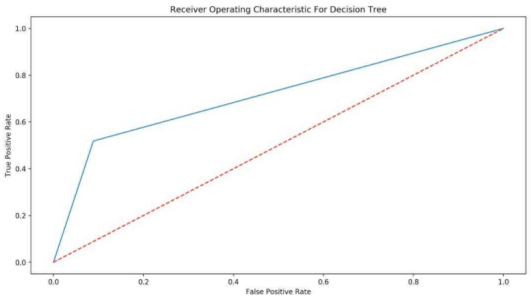

Figure 1: *Decision Tree: ROC_Curve*

entropy, maximum depth as 5, max leaf node as 10, min sample leaf as 10, and minimum sample split as 2. The accuracy on test dataset was 87% and F1 score for positive dataset as 0.68. A confusion matrix for decision tree is also show below along with roc_curve above. The red line in the roc_curve is F1 score of 0.5, and since the blue line never reached towards the lower right side of the red line we can infer that our algorithm is working positively, and there is a possibility of getting a much better result.

Decision Tree Prediction

Actual Value	p	n
p′	TP 15	FN 12
n′	FP 2	TN 77

VI. Discussion

Although decision tree gave us the better result out of all the other algorithms, the result is still not the best one. F1 score of 0.68 for the positive class tells us that out of 100 ARDS patient our model was only able to predict 68 of the correctly. We also have to keep in mind that this is a binary classification. For binary classification, one can randomly choose one of the class and the chances of getting the prediction correct is already 50%. Therefore, 68% is not a good result for binary classification.

The main issue that may arise is from overfitting. There were only 421 files for training dataset, which is not a lot of number to train such a complex model. Hence, our model didn't behave well with unknown dataset. Another issue was the data itself, as the information inside the each file was encoded using cTAKES, we werenâĂŹt able to make a better judgment on if there are any symptoms that could be ignore before we model the data.

VII. Conclusion

The overall goal of our project was to predict the ARDS patient. Although our F1 score for the positive dataset was 0.68, the average score was 0.86, giving F1 score for negative class as 0.92. It's equally important to flag patient who are not diagnosed with ARDS too. A larger dataset could have trained the model more efficiently and may be we could have increased the F1 score. In conclusion, this project helped us understand many machine learning algorithm in more depth, and gave us first hand experience working with real clinical dataset.

VIII. Contributions

Paul Montegrande
Report, Coded Methods*

Neha Goel
Report (Layout), Coded Methods*

Saharsh Patel
Report (Edited), Coded Methods*

Brihat Sharma
Report(Edited), Coded Methods*

All Team members had access to the code. Each team member ran the code to verify the results and then reimplemented it, using other parameters. The contributions mentioned above are the main contributions listed in this paper.

References

[1] Webmd: Acute Respiratory Distress Syndrome, https://www.webmd.com/lung/ards-acute-respiratory-distress-syndrome

[2] Webmd: How the Lungs and Respiratory System Work, https://www.webmd.com/lung/how-we-breathe