Explainable AI

Covid Chest X-Ray Hackathon

Giovanni Camarda, Eleonora Cappuccio, Andrea Rafanelli

October 13, 2022

Abstract

This project was developed as part of the Covid CRX Hackaton. During the first wave of covid-19, hospitals suffered from the huge number of inpatients, which often exceeded available ICU spaces. A predictive model can forecast the need of intensive care for a patients, in order to better allocate the available resources. To make this tool usable for domain expert (physician, hospital staff...) different explainability techniques have been to explain the predicitons of the model.

Keywords: Covid-19, predictive model, explainability

Contents

1. Introduction

During the first Covid-19 outbreak, between March and June 2020, 6 Hospitals in Northern Italy collected data from Covid-19 hospitalized patients. The data were later stored by the Italian Diagnostic Centre in the AIforCovid archive. For this hackathon both tabular and image data have been made available: the first, referred to as "clinical data" were taken during the patients' triage, Section 2 further describes the dataset. The image data are early chest X-ray images of the people hospitalized. Sub-Section 4.1 describes the pre-processing of the CRX images. We developed 3 different models:(i)a random forest using only clinical data, see Section 3.3, (ii)EfficientNet B3 using only image data (see Section 4) and eventually (iii)a mixed model using both the scores extracted with Brixia from the images and the clinical data (See Section 5). Different explainability techniques have been used for each model: Section 3.3 describes the explanations created for the clinical data, in Section 4.3 various techniques for explanation are used for the images. Eventually Section 5 shows the explanations for the mixed model.

2. Exploratory Descriptive Analysis

The data used in this research came from the Italian Diagnostic Centre[1] (CDI) and were collected in six different hospitals in northern Italy between March and June 2020. The records present refer to COVID-19 positive patients. Several clinical data of these patients were recorded and chest X-rays were reported. There are 486 test data and 1103 training data available. The dataset contains 39 variables that include both general information about the patient (such as gender and age) and clinical information about his condition at the time of admission to the hospital. Both the existence and absence of previous diseases, as well as the findings of analyses performed at the time of admission, are included in the clinical information. There are several missing data points in the dataset, as shown in Figure 1. It is possible to notice that the number of missing data for some columns exceeds the number of data present; in this regard,the methods used to deal with the presence of various missing data will be described in the following section. To better understand the dataset structure, Figure 26 and Figure 27 depict the distribution of

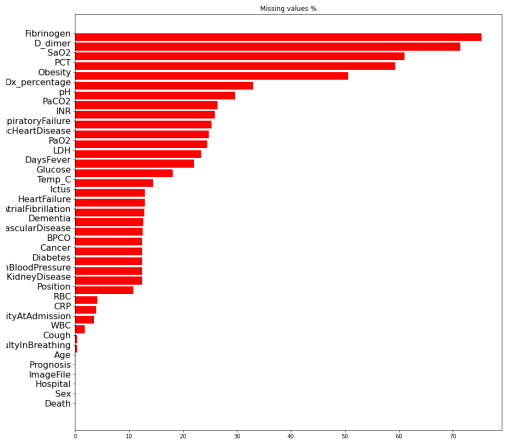

Figure 1. Missing values percentage

the continuous and the categorical variables present. The two figures can be found in the **??** From Figure 27, one can deduce certain information. The majority of patients, for example, are between the ages of 50 and 80, according to the distribution of the variable *age*. When it comes to the temperature variable, it is clear that a significant number of patients had the fever at the time of admission, with temperatures exceeding 37 degrees Celsius. The remaining elements in the Figure are clinical variables (for a better understanding of these please refer to the Appendix **??**). The distribution of categorical variables is visualized in Figure 27. Within the dataset, there is a prevalence of male patients, with the preponderance of patients coming from hospital F. Many patients complain of coughing, difficulty breathing, and elevated blood pressure. In addition, some patients were found to be deceased. It is worth noting that the variable position refers to the patient's position during the X-ray, which can be upright or supine. It is also clear that the target variable, *prognosis*, is well-balanced. Figure 2 illustrate the age distribution in connection to the variables *death, prognosis, obesity, and cardiovascular issues*. The purpose of this study was to see if there was a link between specific characteristics and the age of the patients. Following that, the target variable was examined in relation to a number of factors, including diabetes, breathing difficulties, atrial fibrillation, cough, and death. There was no discernible link between the presence of severe prognosis and the presence of other illnesses among the

[1] https://aiforcovid.radiomica.it/

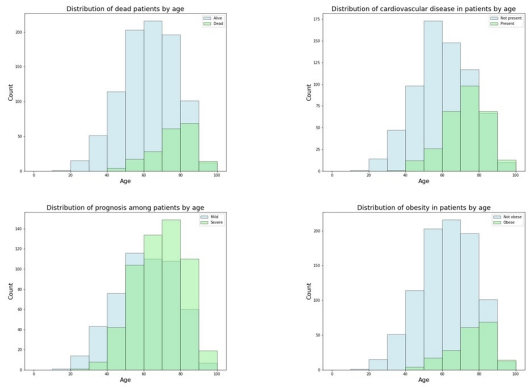

Figure 2. Variables distribution by age

variables considered. Respiratory difficulties is the only variable with a positive correlation, as one can see in Figure 3.

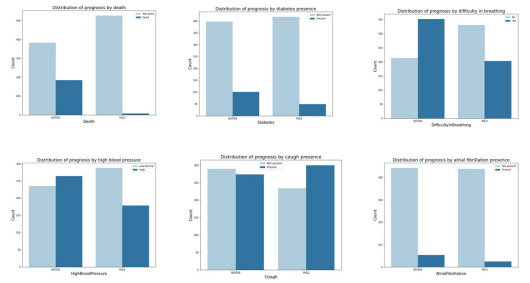

Figure 3. Prognosis distribution among variables

3. Clinical Features

3.1 Data preprocessing

The model was built using the dataset described in Section 1. The target column to predict was *prognosis* which described the course of the disease that could be *mild* or *severe*. Out of the 39 columns, the variable *death* was dropped, as not considered useful for the prediction. This was dropped in particular due to the nature of the task, which is to predict whether the patient will develop a severe or milder disease course. The numerical data were standardized. As stated in the preceding Section, the clinical data contains a high percentage of missing values. To address this issue, a KNN algorithm was used to fill in the missing data. With KNN imputation, the missing values are imputed using the mean value from n neighbours nearest neighbours found in the training set. The number of neighbours was set to 5 and it has to be considered as an hyperparameter.

3.2 Random Forest Classification

A randomized search was performed to find the best parameters for the Random Forest classifier. The tuning of the hyperparameter is performed using Random Search. Cross-validated search across parameter settings is used to optimize the estimator's parameters. Instead of trying all parameter values, a predetermined number of parameter settings are sampled from the defined distributions. The number of iteration specifies the number of parameter settings that are tried. In our experiment the number of iterations was set equal to 100 and a 3-fold cross-validation was implemented. Eventually, the parameters used were:

- 120 for the number of trees (number of estimators).
- The maximum depth of the tree equals 10.
- 2 for the minimum number of samples to split a node.
- 2 the minimum number of samples required to be at a leaf node.
- Bootstrap equals False.

The model was therefore fitted using the best parameters. The Table 2 shows the model results on the test set and in Figure 4 the features that contributed most to the model are shown.

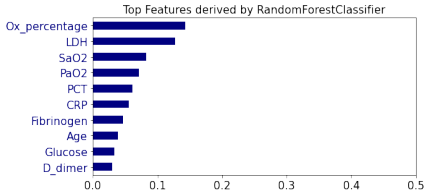

Figure 4. Features importance

3.3 Explanation

3.3.1 Lime

The predictions of the Random forest classifier are explained first with Lime. This algorithm learns an

interpretable model locally around the prediction. It provides local explanations with features importance. The Lime function is used, which samples numerical features from a Normal(0,1) and executes the inverse mean-centering and scaling process based on the training data's means and standard deviation. For categorical features, sample precisely according to the training distribution, and create a binary feature that is 1 when the value is the same as the instance being described. In Lime, every generated point has a weight that is assigned using a Gaussian (RBF) Kernel. The kernel width decides how large is the circle of the meaningful weights around a reference point. In this case, it was set equal to 5. As an example, in

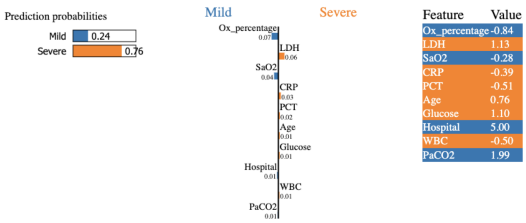

Figure 5. Lime: local explanation

Figure 5 an instance classified as *severe* is explained. The image shows the features that have contributed most to the prediction. While the oxygen percentage seems to influence the prediction towards *mild*, the LDH and CPR impact positively for the *severe* class. According to [1], an increased LDH level indicates lung damage, just as an elevated CPR level indicates lung inflammation.

3.3.2 Shap

Shap is then used to provide more information about our model. Below are the local and global explanations. In Figure 6, the local explanation is illustrated. The blood oxygen percentage has a nega-

Figure 6. Shap: local explanation

tive impact on the severity of prognosis. A lower than average oxygen percentage (= -0.83 < 0.008) pushes the prediction to the right. This is also true for Sa02, a lower than average percentage of oxygen saturation (= -0.28<0.05) pushes the prediction to the right. In contrast, PCT is positively related to the severity prediction. A lower than the average PCT (= -0.50 < -0.15) pushes the prediction to the left. Also, LDH has a positive impact on the severity of prognosis. A lower than the average LDH (y= 1.13 > -0.05) pushes the prediction to the right. The same is true for PaO2, a higher than average score (= 0.09 > -0.19) pushes thevprediction to the right, and for PaCO2, a higher than average score (= 1.9 > 0.03). Figure 7instead shows the overall explanation provided by shap on our test dataset.If one looks at Figure 4, which shows the model's ten most important variables, one can find that these variables are also taken as top factors in a dataset influencing the model's output by Shap. To assess the goodness of

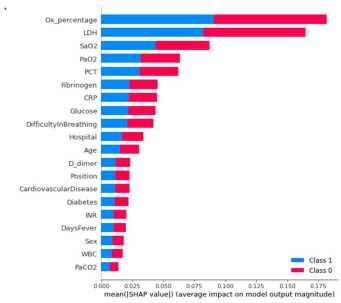

Figure 7. Shap: global explanation

our explainers, a comparison is made using the metric of faithfulness. The metric validates if the relevance scores correspond to an actual importance of the features for the prediction. The method progressively removes each of the attributes rated as important by the explanation. The effect on the performance is evaluated at the end of every removal. These values are then employed to compute the overall correlation between feature importance and model performance. This metrics corresponds to a value between -1 and 1: the higher the value, the better the faithfulness of the explanation. The histogram of faithfulness (Figure 8) below illustrates the metric values for all the cases. Lime's histogram is on the right, while Shap's is on the left. Table 1 shows that Lime is more faithful than Shap. In fact, Shap explanations are not faithful via this metric. The standard deviation shows that the distribution has probably a number of cases with high correlation. Observing the Figure 8, it can be noted that the values distribution of Shap is uneven, so it can not be considered as a faithful explainer in our

analysis. Differently, Lime is quite more accurate, in

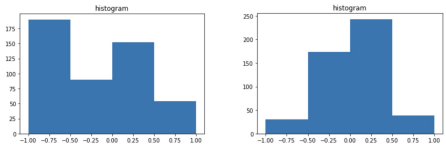

Figure 8. Faithfulness histograms of Lime and Shap

point of fact Table 1 presents an higher mean value and a lower standard deviation. Also the histogram, in Figure 8, shows a more even distribution of the values.

	Faithfulness mean	Faithfulness std
SHAP	-0.13	0.48
LIME	0.06	0.32

Table 1. Faithfulness metrics on test set

4. Images

4.1 Image preprocessing

Since the original images for this challenge were acquired with different precisions, we decided to process them starting from the uint16 resolution, and normalize them. Next, we manually check the images with white lungs and black bones and proceed to invert them reconstructing the DICOM format for medical images. We further applied CLAHE and Medianblur equalization, centre cropped them according to the 98th percent. Moreover, to augment a limited real dataset, we randomly triggered various transformations such as RandomRotation, HorizzontalFlip and ColorJitter among others. Moreover, we preceded to convert them into three channels images since the used Neural Network (NN) architecture required them. Finally, the images are resized to 300x300, normalized with Imagenet mean and standard deviation for each channel [0.485, 0.456, 0.406] and std for each channel equal to [0.229, 0.224, 0.225] [2]. We applied the same set of transformations also to obtain the lung segmented images, as we will see later.

4.2 Model

As previously cited, among the wide range of available NNs also considering the explicitly designed ones for CXR images, we finally tested and exploited EfficientNet B3 [8] since consisting of a reasonable number of parameters but providing demonstrated performances. After a standard hyper-parameters tuning phase, we obtained the results depicted in . We reasoned about various perspectives to explain these poor results: differently from common Covid-19 classification tasks committed to identifying healthy or infected lungs, discerning between a severe and a mild course of the disease represents a more problematic assignment due to the possible narrow difference between the two class prototypes. Moreover, up to our knowledge, no clinical studies highlighted a clear correlation between what can be considered a severe disease course and undoubtedly more damaged lungs with respect to mild cases: it is often possible the intersections of these depicted variables, making harder the location of a clear pattern. Finally, the provided dataset consisted of a thousand samples, at least an order of magnitude short to suitably tackle computer vision tasks.
Among the tested attempts such as enlarging the dataset by exploiting images from different Covid-related tasks with synthetic labels, we further add a preliminary step to improve the classification model: using the segmentation and the aligning models from the Brixia Project[3], we masked the original images to obtain lung segmentations. As depicted by Table 2, this variation did not boost the considered metrics.
We decided to carry out both of the previously-cited pipelines, with and without lung segmentation.

4.3 Explanation

The following section describes three different image explainability methods and the correspondent fidelity scores exploiting the XAI library.

[2]The resizing of the images as well as the use of [0.485, 0.456, 0.406] values for the mean and [0.229, 0.224, 0.225] ones for standard deviation were chosen given the use of the EfficientNet B3. This network is structured for images with a resolution of 300 and trained on Imagenet images.

[3]`https://brixia.github.io/`

4.3.1 Brixia Score

We first employed the model-specific image explanation method provided by the BrixIA project: the pipeline from Signorini et al. [6] provides a pipeline composed of three different models suitable for CXR image segmentation, aligning and classification. The final step from the BS-Net model delivers the so-called BrixIA scores to describe lung abnormalities since it divides each lung into three regions according to two horizontal lines, assigning to each one a severity grade on a scale from 0, corresponding to no lung abnormalities, to 3 denoting interstitial, and dominant alveolar infiltrates.
The heatmap 9 highlights the areas considered more or less severe according to the colours used [4] Moreover, the Figure 9 shows multiple image superpixels classified with the BS-Net that depict a narrower scope on specific lung sections: for each superpixel, the image reports the most significant severity grade blurred according to the corresponding confidence the BS-Net attached to it.The upper right superpixel, for example, is green and white since it is designated as grade 0-1. The top left super pixel is yellow-red, since that area is classified as grade 1-2. The central part of both lungs is green-yellow, indicating that the model ranks that part as grade 0-1, whereas the lower parts have colors ranging from yellow to red, indicating a of grade 1-2, and colors ranging from red to grey, indicating grade 2-3. The model's final classification is shown in the table on the right, with the lower part being the most dangerous for both lungs. A point to note is that because the model is classified using the maximum value among the scores rather than an average value, this implementation may cause some conflicts with the related heatmap. We intend to improve the resultant explanation in the future by creating an image of the same superpixels for each grade and comparing their blurring to capture the most likely severity score, as described in the original paper. Despite the inaccuracies, this map should be viewed as a tool that alerts the user to potential risk zones within the lungs. The heatmap in this scenario, for example, reveals that the patient has problematic spots in the lower lungs. Finally, unlike the following further explainability method for images, we can not evaluate BrixIA-score explanations using the XAI library since they rely on the specific BS-Net architecture and would require a mapping from its outputs and the binary Covid assessing task.

[4] Green intuitively corresponds to grade 0 while yellow, red and black correspond to 1, 2, and 3 respectively.

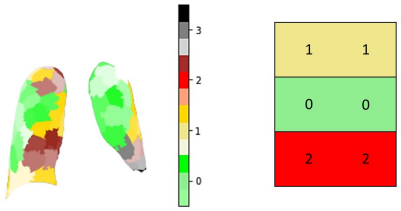

Figure 9. Brixia heatmap

4.3.2 Lime

LIME is another explanatory approach employed to explain our Image model. LIME starts by turning on and off some of the super-pixels in the image to generate many samples that are similar to our input image. LIME will next use our trained model to estimate the class of each of the generated samples. Following the prediction of each sample's class, the weight of each generated data is determined to assess its relevance. To accomplish so, the distance between the created samples points and the original image is calculated using a distance metric. The distance will then be transferred using a kernel function to a value between zero and one. The smaller the distance between two points, the closer the value is mapped to one, and hence the greater the weight. The bigger the weight, the more significant some artificial data point is. The weighted generated data points are then used to train a linear regression model. The features with higher coefficients are those that are significant in defining our black-box machine learning model's prediction. The graphs generated by the LIME algorithm for both models, the one with segmented images and the one with unsegmented images, are shown below, along with a comparison of the two cases. Two patients, one with *mild* and one with *severe* prognoses, were used as examples. In the first case, the example of the patient with mild prognosis is given. As shown in Figure 10 the green regions, as well as equivalent segments of the original images that support the decision in favor of the prediction in the non-segmented model, are scattered and almost random, implying that the model predicted class 0 (Mild) while taking into account parts of the image that should not have anything to do with the diagnosis of Covid severity. There is a slightly improvement in prediction with segmentation. As shown in Figure 11, the green parts, i.e., those most determinant in the model's prediction, move somewhat within the

Figure 10. Example 1:
non segmented images

area that should be considered to make a prognosis of covid severity. However, as can be seen, the segmentation is not perfect, and the green parts also cover a portion of the black image with no visual content. Another example is given next. In this case, the

Figure 11. Example 1:
segmented images

patient has been diagnosed with severe covid. The non-segmented model fails to predict in this situation, whereas the segmented model succeeds. It is worth looking at the following Figures to see how the two models compare in terms of explaining their choice. The result of the non-segmented model is shown in Figure 12. Again, we see that the motivation for its prediction is based on scattered points within the image, and ambiguous parts, such as the shoulders, are considered to make a diagnosis of severity to the Covid. In the case of the segmented model, Figure 13, there is always some imperfection, but it appears that this helps the model focus on parts that are more suitable for diagnosis. In fact, aside from a segmentation based on a black dot in the image, there is a very large green portion on a lung.

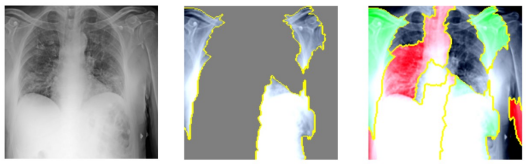

Figure 12. Example 2:
non segmented images

Figure 13. Example 2:
segmented images

4.3.3 Abele

According to [2], ABELE gives an explanation for the proposed classification for a picture classified using a specific black-box model. The explanation is divided into two parts: i) exemplars and counter-examples and (ii) a saliency map. Exemplars and counter-exemplars depict examples with the same outcome as x and examples with a different outcome than x, respectively. They can be visually analyzed to learn why a decision was made. The saliency map shows which parts of x contribute to its classification and which parts push it into a different category. The steps in the explanation process are as follows. First, ABELE uses an Adversarial Autoencoder (AAE) to construct a neighborhood in the latent feature space. Then, on that latent neighborhood, it learns a decision tree that provides local decision and counterfactual rules. Ultimately, ABELE picks and decodes exemplars and counter-exemplars that satisfy these rules, extracting a saliency map from the results and highlighting the areas of the image to explain that contribute to its classification and areas of the image that push it towards another label. Using the patient with a mild prognosis as an example, where

both models correctly predict, the results of using Abele on the segmented and unsegmented models are shown below. Figure 14 illustrates saliency maps for both models, unsegmented on top and segmented on the bottom. The saliencies highlight the areas that contributed the most to the prediction (green-blue), also known as exemplars. The yellow-brown areas are those that, if activated, would change the prediction class, whereas the white areas are useless for the model's decision, i.e. the model is not influenced in its decision by those parts of the images. When the

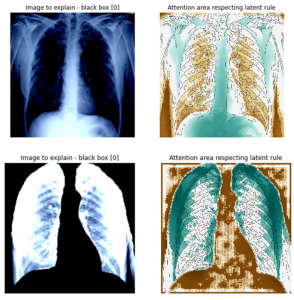

Figure 14. Abele saliency maps

two images are compared, it is possible to see that in the case of the non-segmented model, the parts highlighted in blue are primarily concentrated on the skeleton rather than the lungs, which are actually colored in yellow/brown. This means that if the model had taken the lungs into account, the prediction class would have changed. In the case of the segmented model, however, it can be seen that the parts that contribute the most to the prediction move inside the lungs, despite the presence of some white parts inside. Eventually, based on these findings, it is possible to conclude that Abele improves, to some degrees, with segmentation. As we had previously observed in the case of Lime.

4.4 Saliency

In order to obtain a more complete explanation result, several saliency maps were used in this analysis. These are the following: Guided IG, Smoothgrad, and XRAI. Google PAIR[5] created the maps that were used. According to [3], the Guided IG method is an Integrated Gradients method that introduces the concept of Adaptive Path Methods. This computes attribution in a way similar to Integrated Gradients, but it can use model knowledge to dynamically construct a path that meets the desired goal. By moving in the direction of the lowest associated partial derivatives, noise is reduced. This saliency was applied to the same image as Section 4.3.3 and was performed on both segmented and nonsegmented models. Figure 15 depicts the result for the nonsegmented model on the left and the result for the segmented one on the right. The salient parts of the image are highlighted

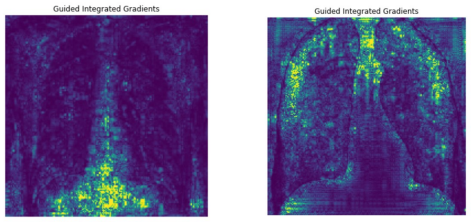

Figure 15. Guided IG saliency maps

in the first case at the bottom center, while in the second case, although there are highlighted parts in which the original image has black areas (such as between the two lungs), there are also highlighted parts within the lungs.
Another method used in this analysis is Smoothgrad, [7]. The main reason for using this method was to determine the important features in model prediction within pixels. In fact, using this method, it is possible to compute an importance mask of the features that contributed the most to the classification error. The feature importance mask can be visualized as a grayscale image, with the brightness corresponding to the importance of the pixels. Because this technique adds Gaussian noise to many copies of the image and averages the resulting gradients, this mask has a noisy appearance. In fact, looking at

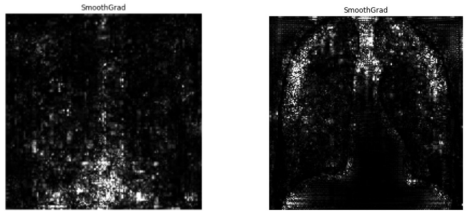

Figure 16. Smoothgrad maps

the Figure 16, one can see that the two images are

[5] https://pair-code.github.io/saliency/

not very clear and rather noisy. The result of the non-segmented model is shown on the right, while the result of the segmented model is shown on the left. In both cases, the images show the same pixels highlighted by the Guided IG saliency as the most important. Again, it appears that the most discriminating parts are concentrated at the bottom center in the non-segmented model, whereas in the segmented model, parts of the lung, and always the part at the center between the two lungs, are highlighted. Finally, the XRAI method [4] was used to identify the areas that contribute the most to image prediction. This is useful to also confirm the results of the previous saliencies used. Pixel-based saliency methods, such as XRAI, can be difficult to read and interpret at times. Positive and negative attributions can be mixed in with salient pixels scattered throughout the image. Instead of pixels, XRAI identifies salient regions. The algorithm then overlays the image using a pixel attribution method such as Integrated Gradients. It sums the attributes for each segment and then sorts the segments from most to least positive. The results of the two non-segmented and segmented models are visualized in Figure 17 Again, the result of the non-segmented model is on the left, and the result of the segmented model is on the right. The most important area in the left image, as seen, is represented by the central part of the image, confirming the previous saliencies. In making a decision, the algorithm prioritizes the sternum and other lower parts of the skeleton. The right image highlights the outline of the lungs in the center (which is why the two previous saliencies showed important pixels in that point) and then highlights some parts of the lungs in darker yellow and orange.

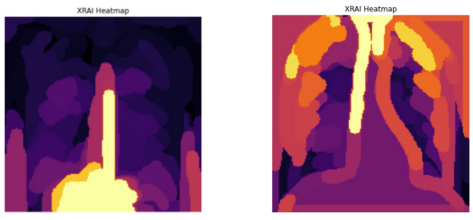

Figure 17. XRAI heatmaps

4.5 Metrics evaluation

We used deletion and insertion metrics to evaluate our image models. According to [5], the deletion metric is a metric that computes the model's ability to make accurate predictions as input values are gradually converted to something neutral, in a sense being deleted. The cancellation process is guided by salience maps, which ensure that values in the most important dimensions are deleted first, followed by values in less important dimensions. The metric's intuition is that an explanation is better if the performance drop is rapid rather than slow, and vice versa. The insertion metric is the opposite to the deletion metric. Beginning with the initial input (represented by a black image), the accuracy increases as more dimensions of the input are inserted. The intuition here is that as more information is introduced into the input, the model's prediction accuracy should improve. That is, when the increase is rapid, the explanation is more effective than when it is gradual. To demonstrate the use of deletion and insertion, the saliencies presented above, as well as Lime for both segmented and non-segmented image models, were subjected to these procedures. The results of the deletion and insertion procedures are shown in the figures below. The area under the curve (AUC) was used to compare the techniques quantitatively. Smaller AUC values are preferable to larger AUC values for deletion. Similarly, larger AUC values are preferable to smaller ones for insertion. The Figure 18 depicts the Lime algorithm's elimination and insertion metrics, with the non-segmented model at the top and the segmented model at the bottom. Although there is not a significant difference between the two models, it appears that Lime performs slightly better in the case of the non-segmented model, as evidenced by a slightly higher AUC in the case of the deletion method but also by a faster descent from 50% of the removed pixels. In the non-segmented model, the insertion method also performs slightly better. However, no impressive results emerge in general. The Figure 19 captures the Guided IG assessment metrics. Again, the non-segmented model has slightly better metrics. Once more, it can be seen in the top figures that the accuracy decreases faster in the deletion method case than in the bottom image and grows faster in the insertion method case than in the bottom image. Following that, the evaluation metrics for the smoothgrad method, Figure 20 and the XRAI, Figure 21, are presented. In these two cases, there is also a marginal improvement in the non-segmented model. The metrics show faster growth with pixel

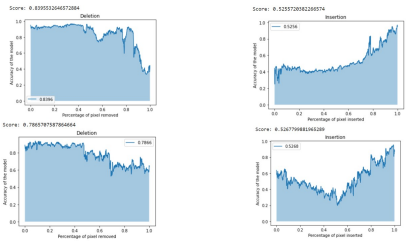

Figure 18. Deletion/Insertion metric Lime

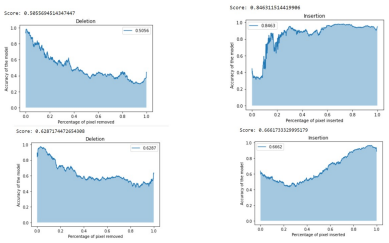

Figure 19. Deletion/Insertion metric Guided IG

insertion and a faster decrease with pixel deletion.

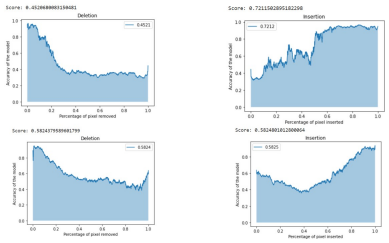

Figure 20. Deletion/Insertion metric Smoothgrad

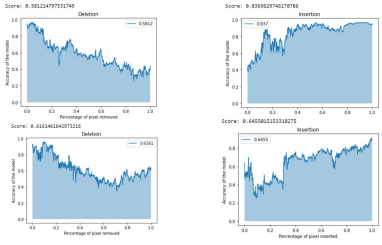

Figure 21. Deletion/Insertion metric XRAI

5. Mixed Model

Furthermore, an attempt was made to integrate image and clinical features in order to provide a more comprehensive degree of explanation. To do so, model in Section 3.3 (which is utilized for clinical data) is employed, and Brixia scores generated via the BSNet (discussed in Section 4.3.1) are added. This study aims to determine whether features from both sides, images and metadata, can be incorporated in the classification of *severe* or mild prognosis and whether image features dominate metadata or vice versa in prediction. As can be observed in Table 2, the model does not vary significantly in terms of performance metrics. What is far more significant is Figure 22 which shows the 10 most significant features of the model. Variables *middle right, down right,* and *upper left* are among the ten most important, as can be seen. As a result, images features become relevant within the model. Entering more in details, if we

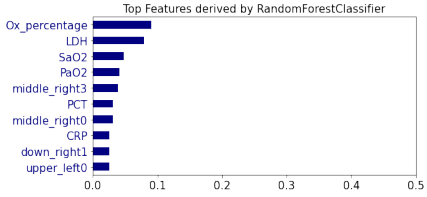

Figure 22. Top 10 features

observe the Figure 23 we can notice the importance plot lists of SHAP. In particular, the 20 variables that contribute most to the model are represented, and here too it can be observed that the characteristics of the images take over from the variables with greater predictive power. We can deduce the reasons

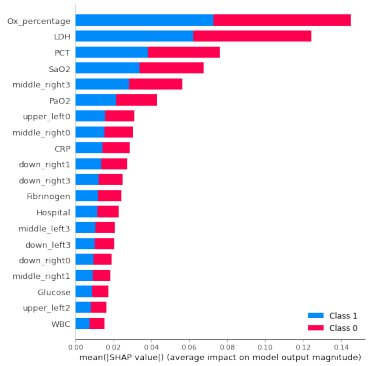

Figure 23. SHAP: Variable Importance Plot

for the model's classification of the example record if

we ignore the model's global explanation and focus on a local explanation. The motivations of the two algorithms SHAP and LIME about the model's prediction are illustrated in the Figures 23 and 24. The identical record used as an example in Section 3.3 is kept for consistency. Figure 24 can be interpreted

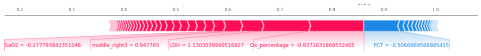

Figure 24. SHAP: local explanation

as follows: blood oxygen percentage has a negative impact on the severity of prognosis. A lower than average oxygen percentage (= -0.84 < 0.008) pushes the prediction to the right. This is also true for Sa02, a lower than average percentage of oxygen saturation (= -0.28< 0.05) pushes the prediction to the right. In contrast, PCT is positively related to the severity prediction. A lower than the average PCT (= -0.50 < -0.15) pushes the prediction to the left. Also, LDH has a positive impact on the severity of prognosis. A lower than the average PCT (= 1.13 > -0.05) pushes the prediction to the right. The same is true for middle right3, a higher than average score (= 0.95 >0.21) pushes the prediction to the right. If we look at Figure 25, these aspects are confirmed. Also according to LIME's explanation, severity in the patient is established by higher than average values for LDH and middle right and lower than average values for oxygen and blood saturation.

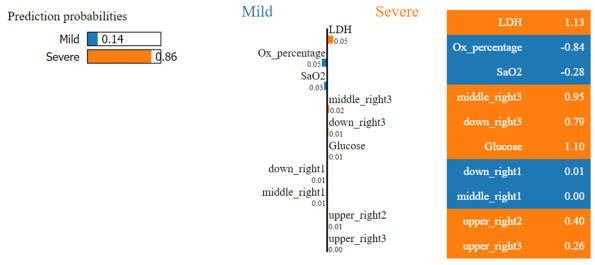

Figure 25. Lime: local explanation

6. Conclusion

To summarize, the image analysis did not produce the expected results, most likely because the images were provided in a format with varying degrees of accuracy; additionally, making a diagnosis on radiography images is difficult, especially if the sample, as in our case, is inconsistent. Furthermore, the segmentation model did not result in the desired improvements, as the segmentation did not show high accuracy, i.e. certain images are not well segmented. The saliencies and various methods used to explain our neural network showed a very slight improvement in terms of evaluation metrics with the non-segmented images, but if we look at the Figures shown in Section 4.4 we can see that the segmented images are sharper visually. In terms of prediction and explanation, the two models do not differ significantly. We would like to improve the segmentations in the future to see if we can actually see improvements. At the explanation level, the tools we provided in this analysis could be useful if applied to a model with better predictive power or segmentation. This necessitates improved image processing and model training, which results in improved accuracy and sensitivity. For what we've done so far, the best model in terms of performance is that of clinical data in Section 3.3 and in terms of explanation, interesting observations emerge, with the variables most involved in the prediction being plausible variables for classification. This is even if the faithfulness of Lime and Shap is not sufficient to conclude that the two explainers are efficient. In conclusion, there are numerous perspectives and improvements to be made in our analysis. The main goal is to increase the goodness of the models used in order to improve the metrics of evaluation of XAI techniques used, but also to have more accurate and explainable predictions.

References

[1] Peter Gemmar. "An interpretable mortality prediction model for COVID-19 patients – alternative approach". In: *medRxiv* (2020). DOI: 10.1101/2020.06.14.20130732. eprint: https://www.medrxiv.org/content/early/2020/06/22/2020.06.14.20130732.full.pdf. URL: https://www.medrxiv.org/content/early/2020/06/22/2020.06.14.20130732.

[2] Riccardo Guidotti et al. "Black Box Explanation by Learning Image Exemplars in the Latent Fea-

ture Space". In: (2020). DOI: 10.48550/ARXIV.2002.03746. URL: https://arxiv.org/abs/2002.03746.

[3] Andrei Kapishnikov et al. "Guided Integrated Gradients: An Adaptive Path Method for Removing Noise". In: (2021). DOI: 10.48550/ARXIV.2106.09788. URL: https://arxiv.org/abs/2106.09788.

[4] Andrei Kapishnikov et al. "XRAI: Better Attributions Through Regions". In: (2019). DOI: 10.48550/ARXIV.1906.02825. URL: https://arxiv.org/abs/1906.02825.

[5] Vitali Petsiuk, Abir Das, and Kate Saenko. "RISE: Randomized Input Sampling for Explanation of Black-box Models". In: (2018). DOI: 10.48550/ARXIV.1806.07421.

[6] Alberto Signoroni et al. "BS-Net: Learning COVID-19 pneumonia severity on a large chest X-ray dataset". In: *Medical Image Analysis* 71 (July 2021). DOI: 10.1016/j.media.2021.102046.

[7] Daniel Smilkov et al. "SmoothGrad: removing noise by adding noise". In: (2017). DOI: 10.48550/ARXIV.1706.03825. URL: https://arxiv.org/abs/1706.03825.

[8] Mingxing Tan and Quoc V. Le. "EfficientNet: Rethinking Model Scaling for Convolutional Neural Networks". In: (2019). DOI: 10.48550/ARXIV.1905.11946. URL: https://arxiv.org/abs/1905.11946.

7. Appendix

[t]

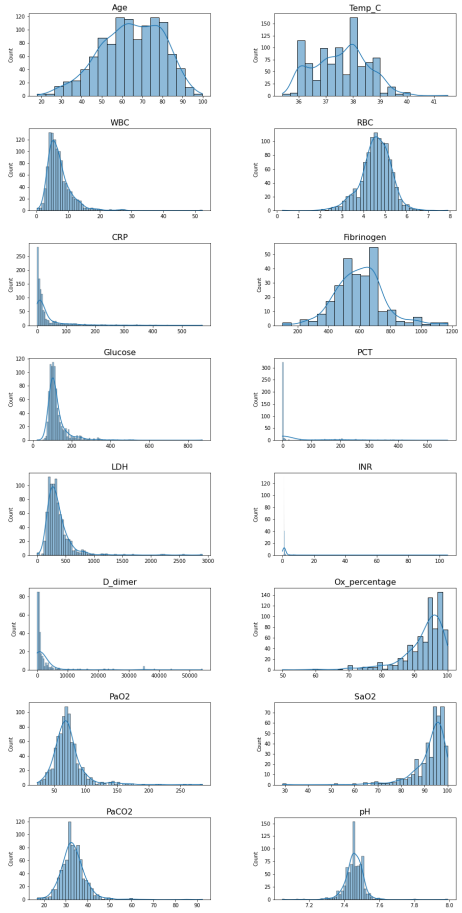

Figure 26. Continuous variables distribution

- CXR: Chest X-Ray
- WBC: White Blood Cells count;
- RBC: Red Blood Cells count;
- CRP: C-Reactive Protein concentration;
- LDH: Lactate DeHydrogenase concentration;
- INR: International Normalized Ratio;
- PCT: ProCalciTonin;
- BCPO: Chronic obstructive pulmonary disease;

Metrics	Not Segmented	Segmented	Clinical	Mixed
Specificity	84%	88%	78%	76%
Sensitivity	45%	40%	87%	83%
Accuracy	70%	70%	81%	78%

Table 2. Performance metrics

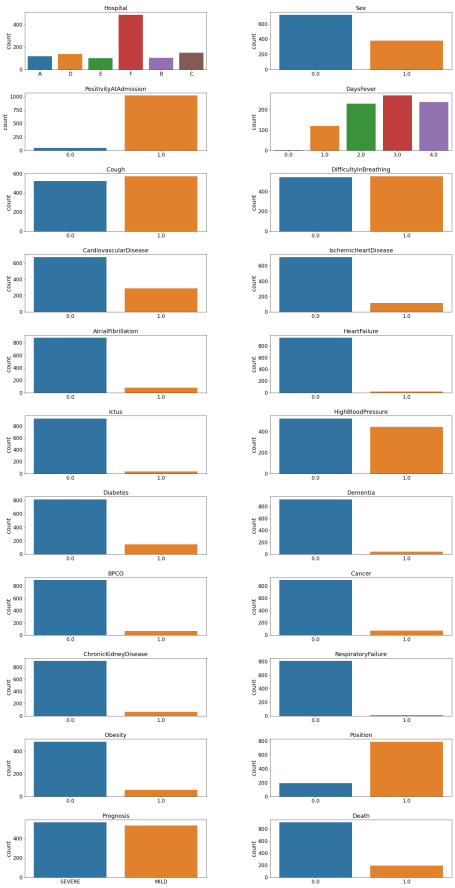

Figure 27. Categorical variables distribution