Project 1 Machine Learning for Healthcare

Lorenzo Tarricone, Sophia Houhamdi, Eva Sarlin

April 16, 2024

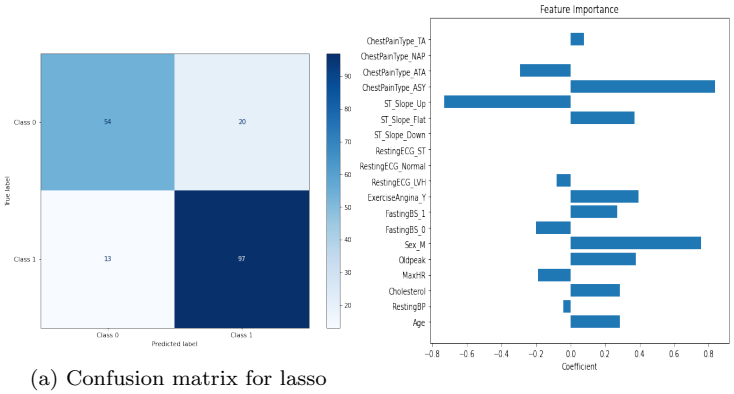
1 Heart Disease Prediction Dataset

1.1 Exploratory Data Analysis

The training and test sets are composed of 734 and 184 samples with binary targets: 'zero' for healthy and 'one' for heart disease. There are five numerical features ('Age', 'RestingBP','MaxHR', 'Cholesterol','Oldpeak') and 6 categorical features ('Sex', 'ChestPainType', 'ExerciseAngina', 'RestingECG', 'STSlope'). The classes are only slightly imbalanced: 398 with the disease and 336 without. Numerical features are only poorly correlated and no missing values were observed. When plotting feature distributions we can detect some nonsensical values and outliers: some patients have zero cholesterol, zero blood pressure or values outside the normal range.
We thus preprocess our data by :

- First, removing the outliers in Resting Blood Pressure and Oldpeak.
- Then we replaced categorical values by converting them into numerical values using a one-hot encoder.
- Impute the zero cholesterol values using a linear regression on all the other processed features. We chose to do so because discarding them would have reduced the dataset in a consequent way.
- After this imputation, outliers in the cholesterol values were also excluded from the training set.
- As plotted on histograms, numerical features take different value ranges, so a last pre-processing step was involved in order to normalize the values using a standard scaler

1.2 Logistic Lasso Regression



(a) Confusion matrix for lasso logistic regression evaluated on test set

(b) Lasso logistic regression coefficients

Figure 1: Lasso logistic regression

A Logistic Regression with L1 regularization was then performed using the preprocessed data. The normalization pre-processing step is crucial to ensure that the magnitude of the coefficients obtained

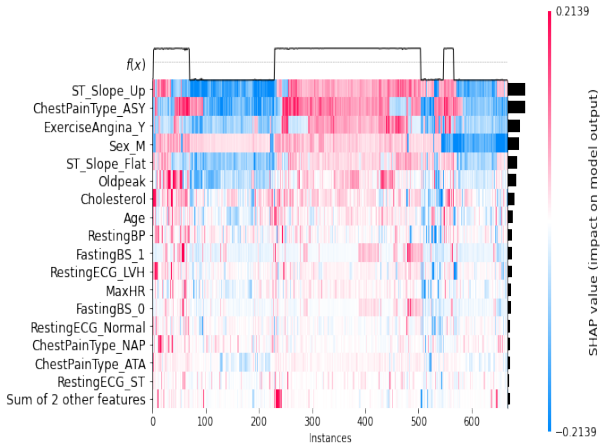
with the regression is comparable. The Lasso logistic Regressor achieved a balanced accuracy of 80,58% and a f1-score of 85,46%.

Features with the greatest importance according to the magnitude of the coefficient are the chest pain type, the ST slope type, the sex, exercise angina, oldpeak, cholesterol, age and fasting blood sugar. This suggests that a patient is more likely to have heart disease if he has asymptonic chest pain, if he's a man, has exercise angina, a flat ST slope, high Oldpeak, fasting blood sugar, cholesterol and is old. On the other hand, patients are more likely to be healthy if they have an ST slope going down, an atypical angina, low blood sugar and a high maximum heart rate.

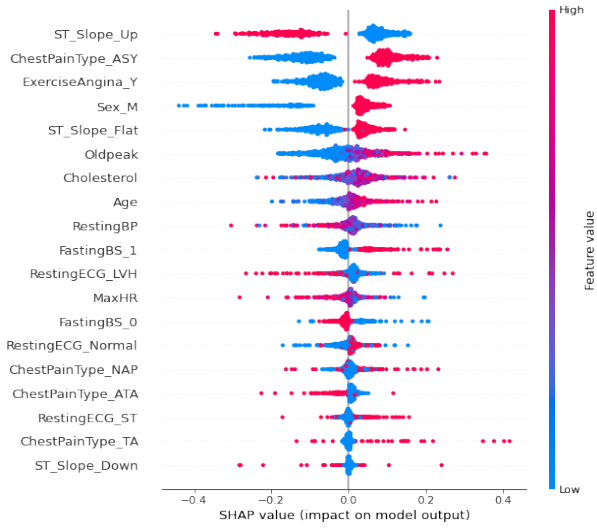
The researcher's idea of fitting the Logistic Lasso Regression to determine the important variables and then, training a Logistic Regression solely on these variables has been tested by discarding the RestingECG, Resting Bood Pressure, low importance chest pain type and ST slope (downslope). This approach achieved an accuracy of 81% and f1-score of 86%, thus it can be useful to reduce overfitting by reducing the number of features. Moreover, reducing the number of features can facilitate the model application in the clinical setting as, in practice, healthcare staff can only spend a little time entering the medical variables.

1.3 Multi-Layer Perceptron

A simple multi-layer perceptron with a flexible number of layers and neurons per layer was implemented in Pytorch. The chosen architecture is composed of 2 layers of 40 neurons with a tanh activation function and a final sigmoid function. It achieved an accuracy of 76,5% and a f1-score of 80,73%. In a second part, the SHAP library explainer was used to generate the Shapley values and plot them in a heatmap, barplot and bee swarm plot showing how the top features affect the multilayer perceptron outputs. The most important features are again the ST Slope, Chest pain type, exercise angina, sex, oldpeak, cholesterol and age, which is similar to what we found in lasso logistic regression. Features influencing the output towards a positive prediction (diseased) are asymptonic chest pain, exercise angina, sex being male, a flat ST slope, great oldpeak, cholesterol, age, and fasting blood sugar.



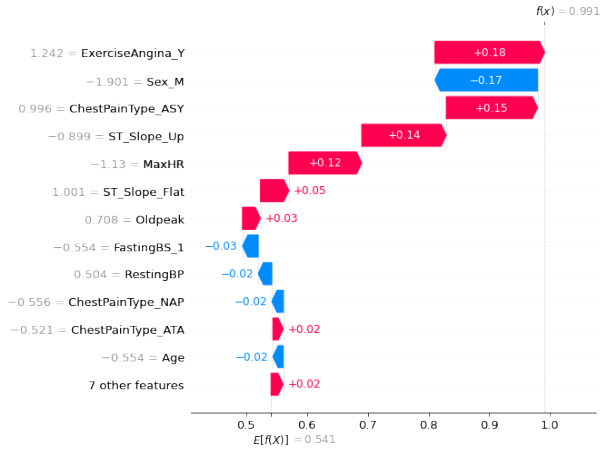
(a) Heatmap plot showing global important features



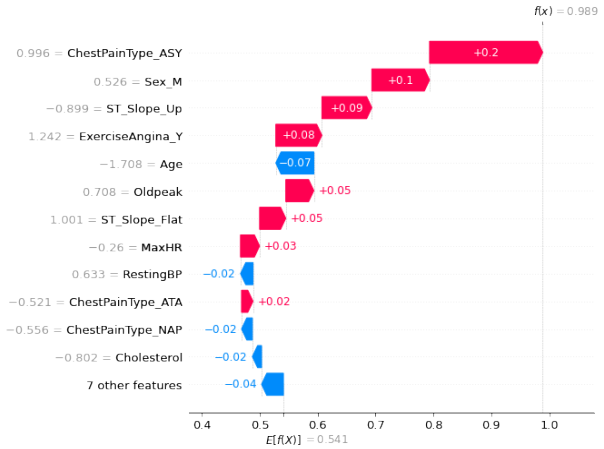
(b) Beeswarm plot

For the four examples we vizualized, feature importances are consistent across different predictions and compared to overall importance values. Both positive predictions have positive features contributions by: Exercise angina, asymptonic chest pain, ST Slope and max heart rate. Also, the two individuals are from opposite sex: the first one is a women, which contributes negatively to the positive prediction, whereas the second individual is a man, which contributes positively to the output. Thus, these two explanations are consistent with each other but also with the heatmap and beeswarm plotted earlier. In the same manner, the two negative predictions show feature values influencing the output negatively : both have a downslopping ST, no asymptonic chest pain, no exercise angina, no flat ST

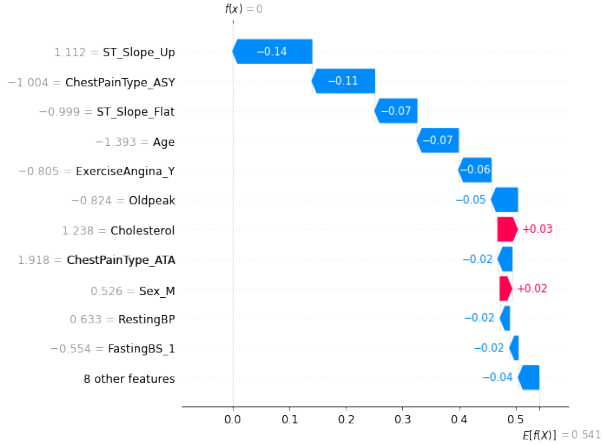
slope, are young, with low oldpeak and low fasting blood sugar. Both predictions are coherent and well explained by the shapley values. This observations also make sense from the medical perspective, where we would associate these features values to a healthy person.



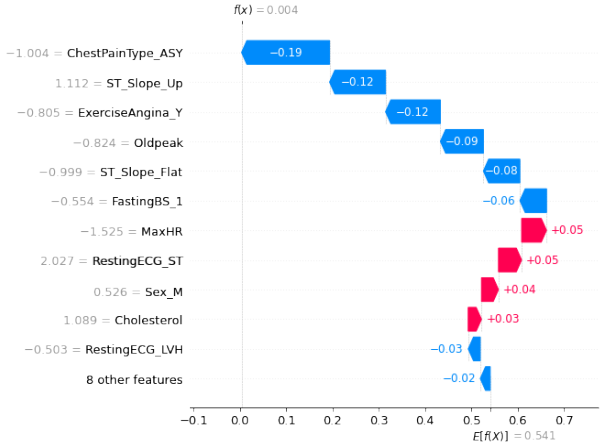
(a) SHAP model's output explanations for the first positive prediction



(b) SHAP model's output explanations for the second positive prediction



(a) SHAP model's output explanations for the first negative prediction



(b) SHAP model's output explanations for the second negative prediction

1.4 Neural Additive Models

We implemented a custom NAM in pytorch using one neural network of 4 layers for each feature, each neural network being a single input-output fully connected network. Their outputs are then summed, a bias is added and a sigmoid function is applied. An accuracy of 85% was reached. Then the individual feature-specific network outputs were plotted as a function of the feature values. A global feature importance plot was also created by plotting the mean outputs of these individual networks over the whole training set. The same important features as before were observed: the sex, whether the chest pain is asymptonic or not, whether there is exercise angina or not, the oldpeak and fasting blood sugar.

Compared to MLPs and Lasso Regression, NAMs have the capability to capture non-linear relationships like an MLP but maintain a level of interpretability similar to logistic regression because the contribution of each feature to the prediction is isolated and can be individually examined. It is thus a better model to apply for such a medical classification task, as it will present the advantages of both techniques.

The structure of NAM is inherently more interpretable than MLPs because it allows us to see exactly how much each feature contributes to the final prediction by constructing the output from a

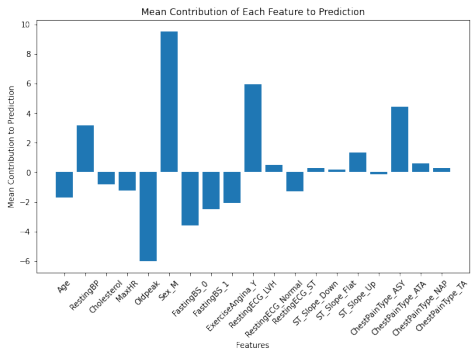


Figure 5: Mean individual feature-specific neural network outputs for each feature

sum of individual contributions from each input feature: it is an additive model. We can isolate every contribution before the summation and plot it has done in figure 4 and 5 for positive and negative predictions. Since each feature in a NAM is processed independently through its own sub-network, it becomes straightforward to trace the relationship between any given feature and the output.

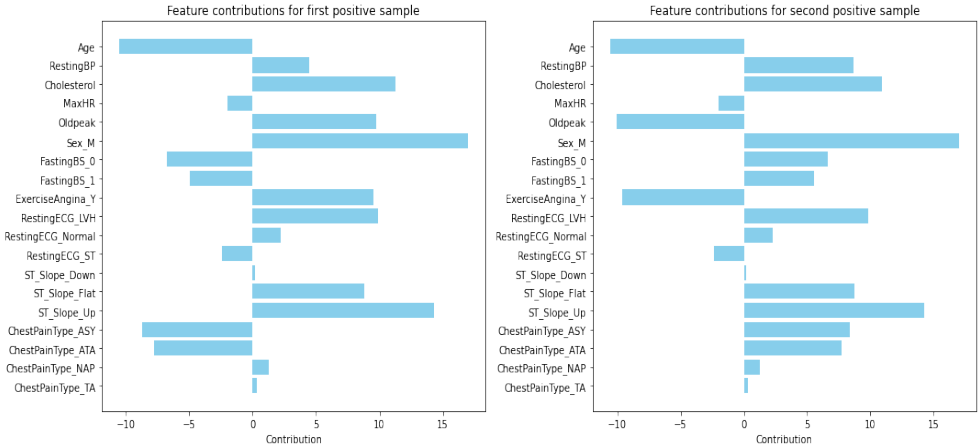


Figure 6: Individual feature-specific neural network outputs of the NAM for two positive predictions

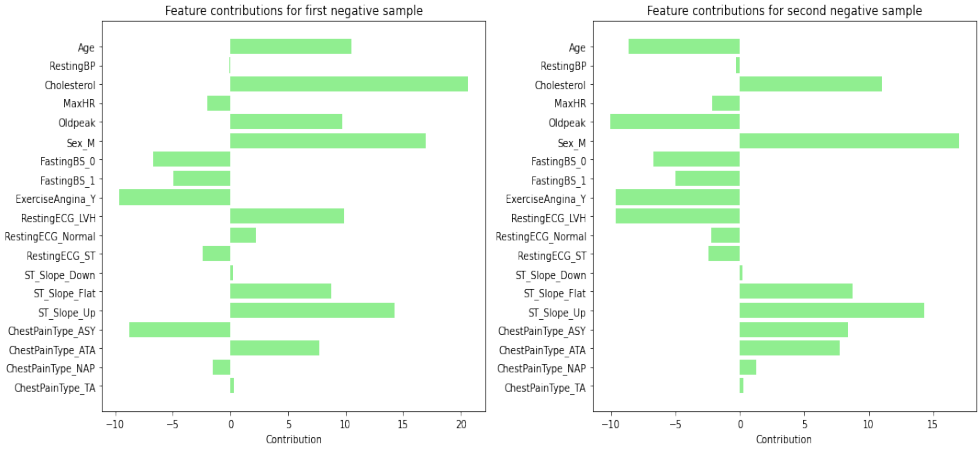
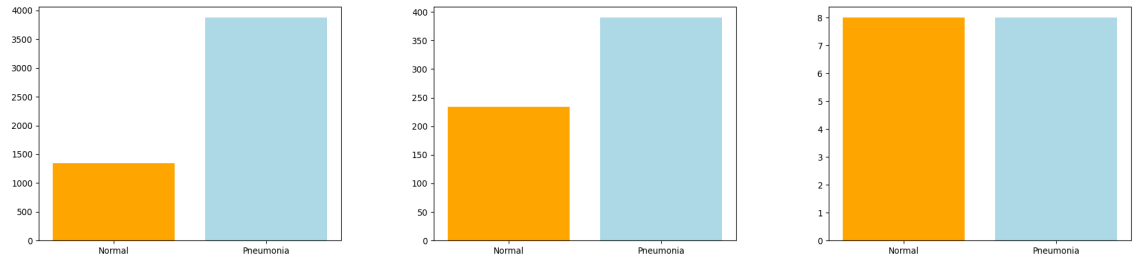


Figure 7: Individual feature-specific neural network outputs of the NAM for two negative predictions

2 Pneumonia Prediction Dataset

2.1 Exploratory Data Analysis

We first plotted the distribution of the labels in each data set (Figure 8)



(a) Distribution of the training labels (b) Distribution of the test labels (c) Distribution of the validation labels

Figure 8

We can see that the classes are:

- very unbalanced for the training set (only around 26% of the elements belong to the 'NORMAL' class)
- quite unbalanced for the test set (around 38% of the elements belong to the 'NORMAL' class)
- balanced for the validation set (exactly 50% of the element belong to the 'NORMAL' class)

This means that we will probably need to do some data augmentation in order to have more balanced classes and therefore perform better training of our model.

By comparing some scans (Figure 9), we can see that it is very difficult to differentiate between a normal and a pneumonia case with the naked eye. However, we can notice that the pneumonia scans have a higher opacity and an enlarged heart (the white area in the centre of the thorax).

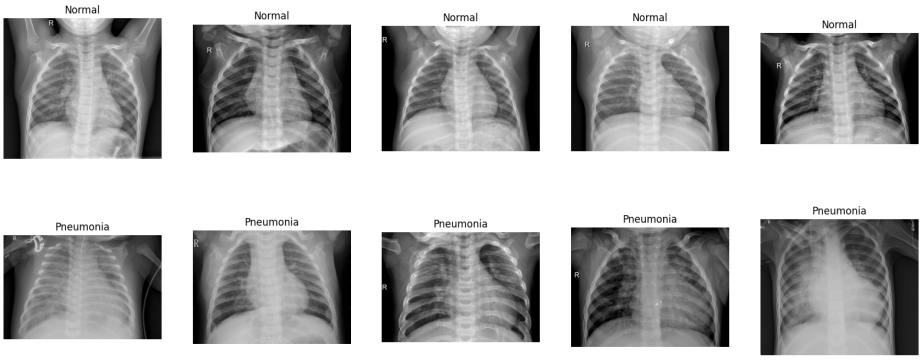


Figure 9: Samples from Pneumonia Prediction Dataset

We also noticed some other sources of bias:

- The presence of some external elements (probably medical devices) appears almost always in the X-rays of pneumonia patients (Figure 10)
- The rotation of the chest differs from one scan to another.
  -The sizes of the scans have different ratios (Figure 11)

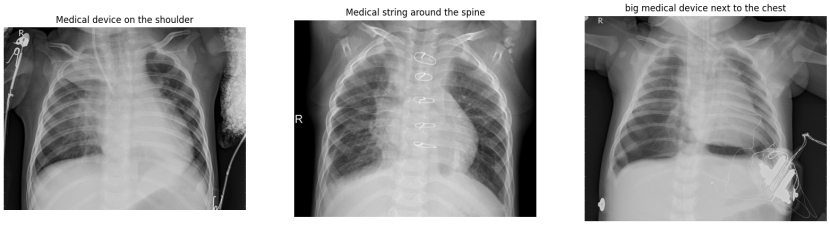


Figure 10: Examples of scans with medical devices

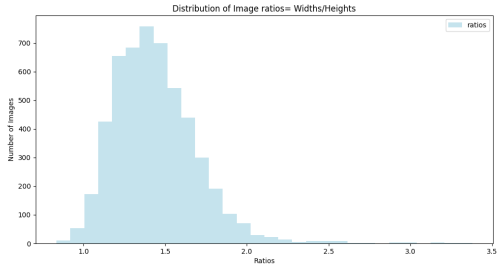


Figure 11: Distribution of ratio of the images within the train set

Many things can be corrected in order to try to mitigate the biases introduced in the previous point. Some ideas are:

1. Manually remove (maybe also with the help of a physician) the images with too many external elements that might interfere with the learning process of the Neural Network.
2. Correct for rotation of the images in order for the spine to be always vertical or do some data augmentation (that we have seen before is needed) in order to train the CNN to be invariant up to small rotations.
3. Filter the training data by discarding those with too high ratio width/height (which would be too distorted when resized later)
4. Apply data augmentation to cope with the imbalanced class issue and help with generalization.
5. Standardized image sizes (by resizing given the distribution of sizes in the dataset) so they can be all of the same dimension because the CNN receives inputs of fixed size.
6. Normalize pixel intensities across channels to uniform the input images (This has been proven to increase accuracy) and make sure they are all grayscaled but stored in an RGB format.

We also trained a different model based on the publicly available ResNet50 model, doing transfer learning by freezing all the convolutional layers and only fine-tuning the last fully connected layer's weights to our specific task. This model required fewer prepossessing methods (resizing, cropping, and adding some random horizontal flips and rotations).

2.2 CNN Classifier

We designed two different models: a CNN implemented by hand from scratch, and used transfer learning to fine-tune the ResNet50 model (adapted from here) to our task (Figure 12). We used both

the Adam Optimizer with a learning rate of 0.001 and Binary Cross Entropy loss. We trained the first one for 10 epochs and reached an accuracy of 78% and the second for 17 epochs and reached an accuracy of 86% (Figure 13).

```python
class CNN(nn.Module):
    def __init__(self):
        super(CNN, self).__init__()
        self.conv1 = nn.Conv2d(1, 8, 3, padding=1)
        self.conv2 = nn.Conv2d(8, 16, 3, padding=1)
        self.conv3 = nn.Conv2d(16, 32, 3, padding=1)
        self.conv4 = nn.Conv2d(32, 128, 3, padding=1)


        self.batchnorm1 = nn.BatchNorm2d(8)
        self.batchnorm2 = nn.BatchNorm2d(16)
        self.batchnorm3 = nn.BatchNorm2d(32)
        self.batchnorm4 = nn.BatchNorm2d(128)

        self.dropout = nn.Dropout(0.1)

        self.pool = nn.MaxPool2d(2, 2)

        # Adjusted the fully connected layer input size to match the flattened feature size
        self.fc1 = nn.Linear(128 * (300 // 2**4) * (200 // 2**4), 128)
        self.fc2 = nn.Linear(128, 1)

    def forward(self, x):
        x = self.pool(F.relu(self.batchnorm1(self.conv1(x))))
        x = self.dropout(x)

        x = self.pool(F.relu(self.batchnorm2(self.conv2(x))))
        x = self.dropout(x)

        x = self.pool(F.relu(self.batchnorm3(self.conv3(x))))
        x = self.dropout(x)

        x = self.pool(F.relu(self.batchnorm4(self.conv4(x))))
        x = self.dropout(x)

        x = x.view(x.size(0), -1)  # Flatten the feature maps
        x = F.relu(self.fc1(x))
        x = self.fc2(x)
        return x
```

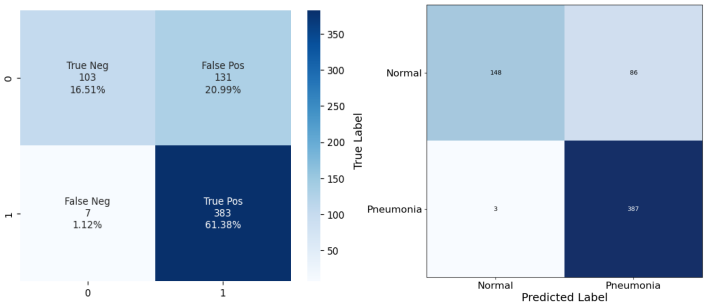
(a) CNN by hand

```python
class PneumoniaResnet(PneumoniaModelBase):
    def __init__(self):
        super().__init__()
        # Use a pretrained model
        self.network = models.resnet50(weights=True)
        # Freeze training for all layers before classifier
        for param in self.network.fc.parameters():
            param.require_grad = False
        num_features = self.network.fc.in_features # get number of in features of last layer
        self.network.fc = nn.Linear(num_features, 2) # replace model classifier

    def forward(self, xb):
        return self.network(xb)
```

(b) CNN from ResNet50

Figure 12: CNNs architectures



(a) CNN by hand (b) CNN from ResNet50

Figure 13: CNNs confusion matrices

2.3 Integrated Gradients

We implemented the Integrated Gradients using the Captum Library on top of PyTorch as a post-hoc explainability method for our image classification task. The attribution maps generated (Figures 14 and 15) do highlight sensible regions as they mostly focus on the heart and areas with high opacity and are mostly consistent across each category, which are the features usually used to diagnose pneumonia from chest x-ray scans.

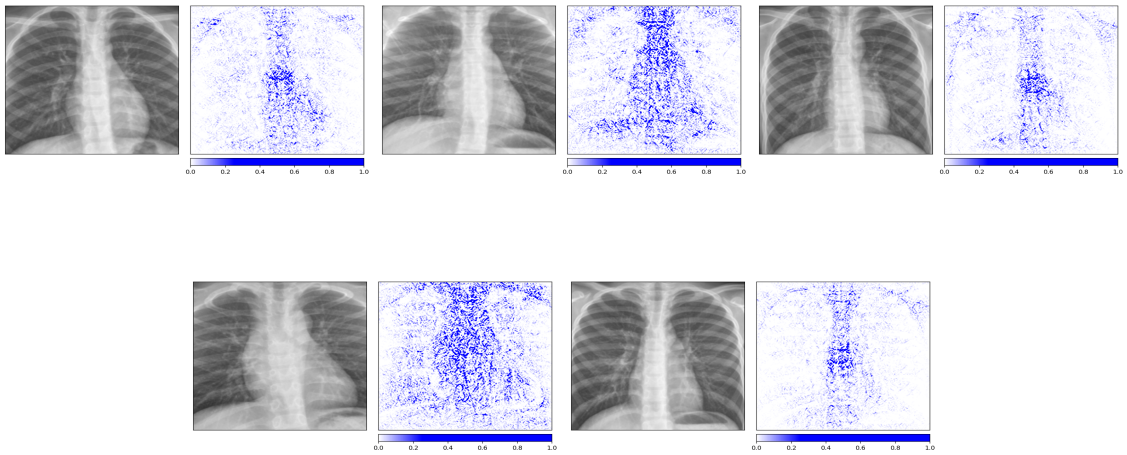


Figure 14: Integrated Gradients attribution maps for 5 healthy X-ray scans (using the mean baseline)

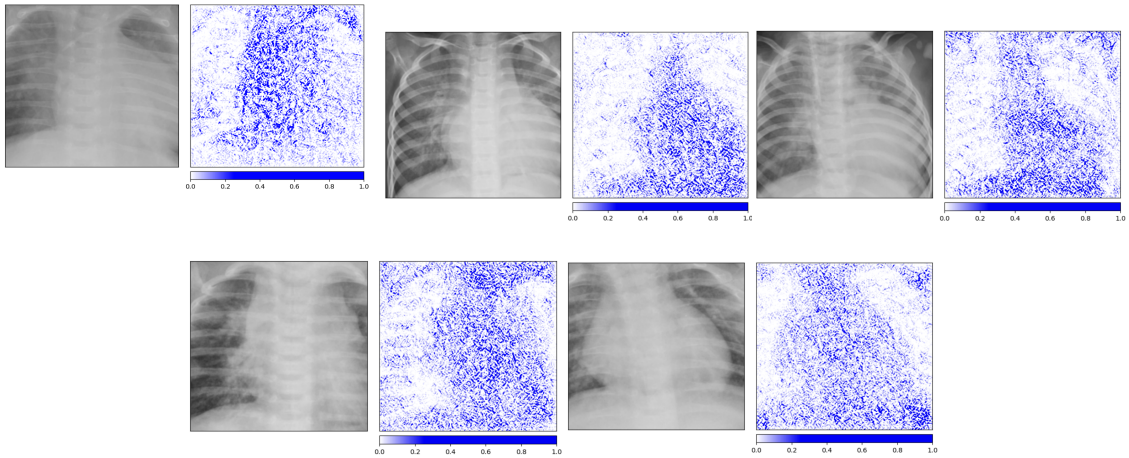
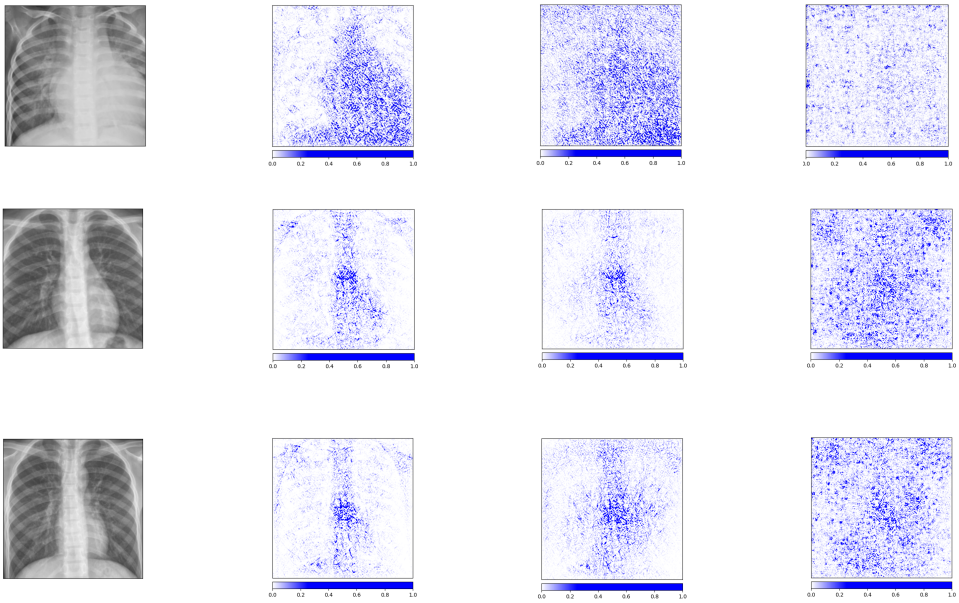


Figure 15: Integrated Gradients attribution maps for 5 X-ray scans with pneumonia (using the mean baseline)

The choice of the baseline has a big effect on the attribution map as we can see that when it's set to an all-black image (Figure 16 (a)), or a mean value of pixel intensities (Figure 16 (b)), we can still identify the sensible regions, whereas for the randomized baseline, the attribution maps are all very noisy (Figure 16 (c)).



(a) original scan (b) 'mean' baseline (c) 'zero' baseline (d) 'random' baseline

Figure 16: Integrated Gradients attribution maps using different baselines

2.4 Grad-CAM

For this section, we show the results on the first five images of the test set for each of the two classes. (17 shows the normal ones and 18 shows the pneumonia patients).

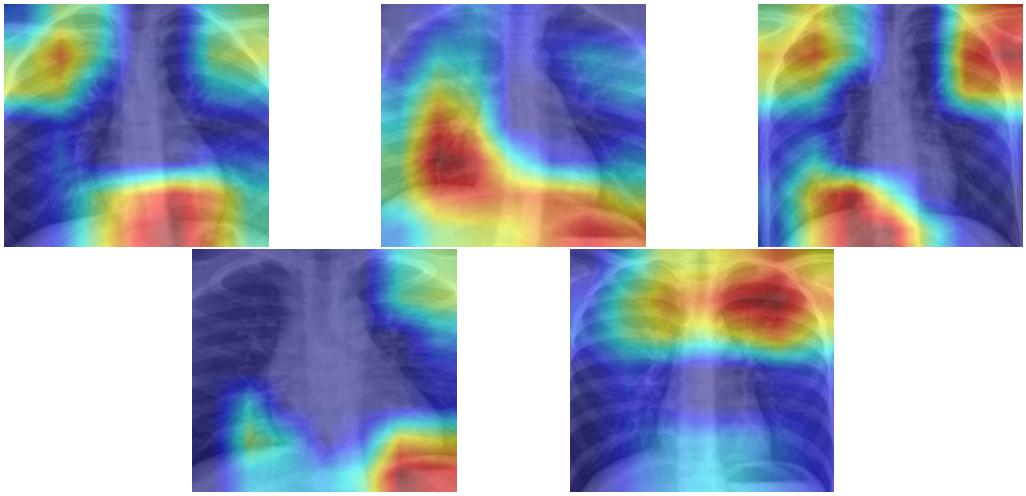


Figure 17: Grad-CAM attribution maps for 5 healthy X-ray scans

In the case of Grad-CAM, we can notice how the model focuses, when looking at a pneumonia patient x-ray, on the top right part of the thorax. When predicting the X-ray image from a normal patient, it instead usually focuses on random parts (usually three) outside the thorax area. We think that this is due to the fact that this is the region where the model can better assess the contrast between the white-looking hearth and the black-looking lungs. If there is not a net contrast between these two parts, this is probably due to the presence of pleural effusions.
We cannot really explain the heat map for normal patients, but given the very good recall on the test set, we think there might be some important characteristics outside the thorax that the model is picking up. Our results are confirmed by [KR21], where they show that sometimes the ResNet branch

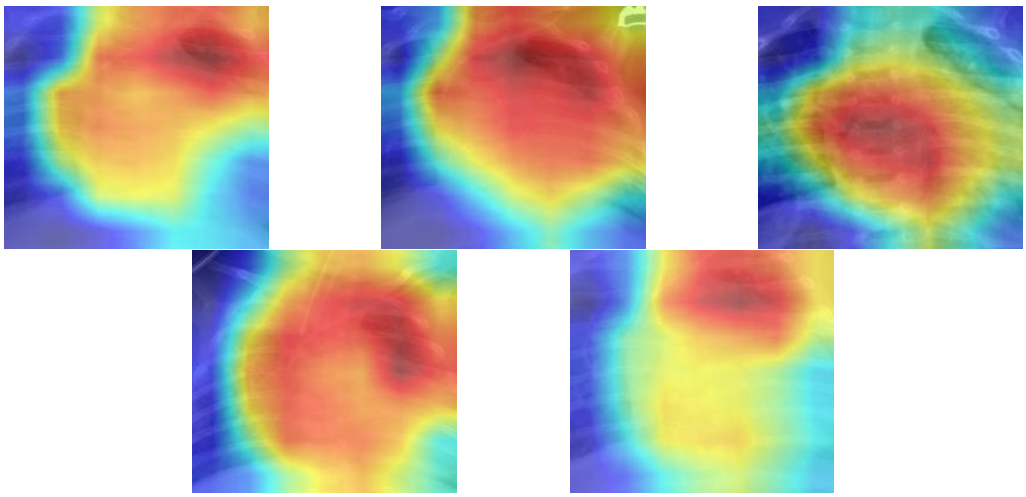


Figure 18: Grad-CAM attribution maps for 5 pneumonia X-ray scans

of their ensemble learning architectures highlights (with Grad-CAM) regions outside the thorax.

Within this distinction made by the class label, the model is fairly consistent in underlining the characteristics defined above, probably more than when using Integrated Gradients, where the highlighted parts have a less well-defined nature.

We want to stress

2.5 Data Randomization Test

We performed a data randomization test by training our model (the CNN based on ResNet50) on randomly permuted labels and applied the two post-hoc explainability methods (Integrated Gradients and Grad-Cam) two produce new attribution maps. We can observe from the plots below (Figure 19) that no sensible regions are highlighted by either of the two methods, showing that our methods do depend on the labeling of the data and therefore pass the Data Randomization test.

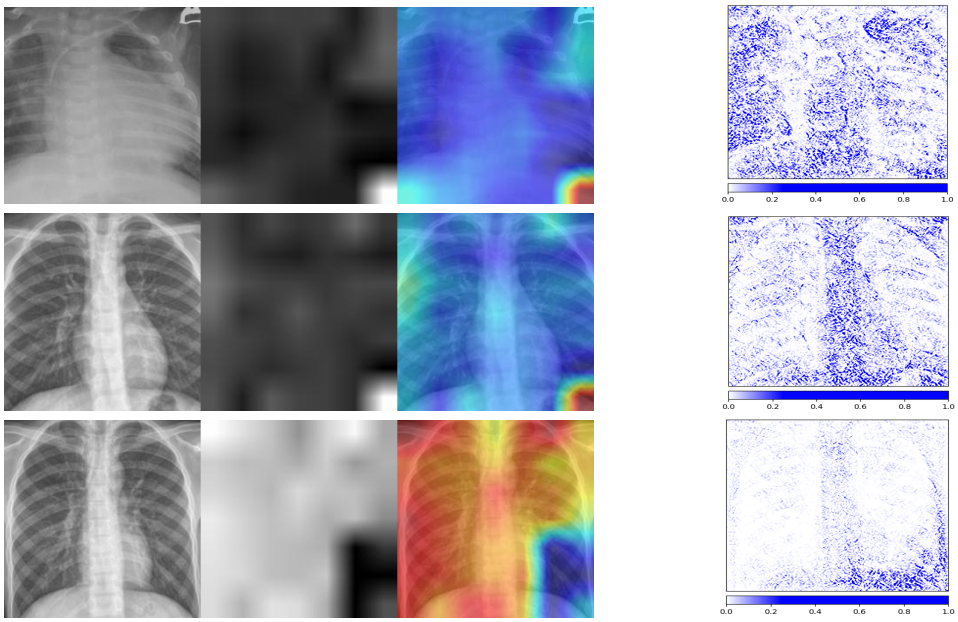


Figure 19: Grad-CAM (on the left) and Integrated Gradients (on the right) performed on the randomized test dataset

3 General Questions

3.1 How consistent were the different interpretable and explainable methods? Did they find similar patterns?

3.1.1 Heart disease prediction

We noticed that the different interpretability methods gave results that were consistent with each other. In particular, we can see how the shap values confirm the results of the lasso regression, underlining how the most important features are again the ST Slope, Chest pain type, exercise angina, sex, oldpeak, cholesterol and age.

NAM visualization where we plot the output of each feature-specific network is very useful to confirm that the explanations are based on the same reasons as with the previous methods 5. Features contributing to a positive output are also in this case: asymptonic chest pain(second positive and negative samples), flat ST slope and male sex (all 4 samples), exercise angina (first positive sample), high fasting blood sugar (second positive sample), great age(first positive sample) oldpeak (first samples) and cholesterol.

3.1.2 Pneumonia prediction

To assess the consistency of the methods and be able to compare them with one another, we applied both the methods Grad-CAM and Integrated Gradients on the first five images of the test for each class (for a total of ten images).

The methods were not very consistent with each other, because, as underlined in the previous section Grad-CAM, tends to focus on the part of the thorax just above the heart to make a prediction, while in Integrated gradients there is more of a focus on the heath itself.
It is remarkable to underline the fact that these Integrated gradients gave the same kind of explanation both for pneumonia and normal patients, while Grad-CAM focuses on different parts (above the heart for pneumonia patients and on parts outside the thorax for normal patients) 20

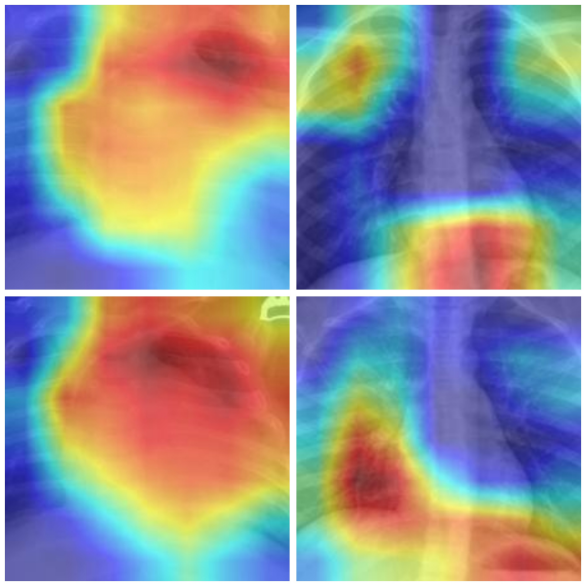


Figure 20: On the left: Pneumonia patients. On the right: Normal patient

3.2 Given the "interpretable" or "explainable" results of one of the models, how would you convince a doctor to trust them?

3.2.1 Heart disease prediction

We would first underline how important features which were already taken into consideration when doing a prognosis are also found by the interpretability techniques. Then we would underline how novel important features (such as "Asymptonic chest pain" in our case) impact the prediction accuracy (on the test set) of the model. Last but not least we would bring other examples from the literature demonstrating the benefit of data-driven approaches in healthcare.

3.2.2 Pneumonia prediction

We would propose Grad-CAM to a physician as its heat-maps are probably more interpretable than the scatter points of Integrated gradients and they give label-dependent explanations. Also in this case we would underline how high throughput these methods are and how integrating some computer vision in the clinical examination routine can be used for example to discard very rapidly and with high precision normal patients (our model has 99% of recall, making it perfect for this scope)

3.3 Elaborate whether the feature importance from the interpretability and explainability methods intuitively make sense to find the respective disease

3.3.1 Heart disease prediction

As described in the section above, all the methods seem to give results with the same set of important variables, namely: chest pain type, sex, ST slope, exercise angina, oldpeak, cholesterol and age.

We expected types such as sex and age to indirectly correlate with the presence of a hearth disease because they probably influence the diet and lifestyle of the patients, which will be the direct cause of the hearth disease.
There are then indicated a few indicators coming from an ECG that we expect to be a good measure of the normal/abnormal activity of the heart and therefore the presence of a disease.

- Oldpeak (or ST-segment depression) occurs when the heart muscle doesn't receive enough oxygen-rich blood, typically due to narrowed or blocked coronary arteries
- ST slope refers to the direction of the ST segment on an electrocardiogram (ECG or EKG). A horizontal or downward-sloping ST segment can indicate myocardial ischemia, suggesting reduced blood flow to the heart muscle, often due to coronary artery disease. This information helps in diagnosing heart diseases and assessing the risk of future cardiac events
- Exercise angina refers to chest pain or discomfort that occurs during physical exertion or exercise, typically due to reduced blood flow to the heart muscle. It's a common symptom of coronary artery disease (CAD), where the coronary arteries become narrowed or blocked, restricting blood flow to the heart. The occurrence of angina during exercise can be a warning sign of underlying heart disease and helps in diagnosing CAD

Cholesterol (especially the one named low-density lipoprotein (LDL) cholesterol) is probably known to be one most famous causes of heart disease. When LDL cholesterol levels are elevated, it can lead to the accumulation of cholesterol in the walls of arteries, causing them to narrow and stiffen. This narrowing restricts blood flow to the heart muscle, increasing the risk of heart attack or angina (chest pain). Additionally, if a plaque ruptures, it can trigger blood clot formation, which can completely block blood flow to the heart, resulting in a heart attack

3.3.2 Pneumonia prediction

We think that the fact that Grad-CAM classifies pneumonia patients by looking at the area just above their heart is due to the possible change in transparency that happens in this region. The heart is always seen as a white mass by the x-ray and the lungs are black in a normal patient, and whiter in a

pneumonia patient (due to the presence of fluid caused by the inflammation, as explained here).

Integrated gradients focus more on the hearth region. This also makes sense according to the literature as cardiomegaly (enlarged hearth) is sometimes also associated with pneumonia, as stated here.

3.4 If you had to deploy one of the methods in practice, which one would you choose and why?

3.4.1 Heart disease prediction

We would select the Shapely values for this task for many reasons:

- It is a model agnostic method, so we can have a model with great complexity (also allowing for ensemble learning techniques) and optimized for performance and still have an explanation of it
- it allows for different kinds of plots (both feature wise-explanation and datapoint-wise explanation)
- It is able to handle missing values
- It comes with a well-documented and intuitive library implementation that makes the deployment easy

.

Moreover, we have noticed that the features underlined by this method are "robust", meaning that using just these features when performing the classification task leads to results comparable (if not, even higher) than when using the set of all regressors. A small set of regressors is very valuable for the deployment of the model because it makes debugging and the possible data curation pipeline much faster and easier.

3.4.2 Pneumonia prediction

As mentioned above, we would probably deploy Grad-CAM because of its easier to interpret outputs and the fact that it gives result that are label dependent.

References

[KR21] Geem ZW Han GT Sarkar R Kundu R, Das R. Pneumonia detection in chest x-ray images using an ensemble of deep learning models. *PLoS One*, 2021.