Project 1: Interpretable and Explainable Classification for Medical Data

Julian Minder, Leon Noirclerc and Johanna Ribas

1 Heart Disease Prediction Dataset

1. Exploratory Data Analysis

We start the exploratory data analysis by looking at the size of the dataset and the number of positives (defined as those observations with "HeartDisease = 1"). A summary of these values can be found in Table 1. We observe that there is a slight class imbalance and that the positive rate is higher in the test than in the train set. We can expect that the model might underestimate points in the test set. Next we look at potential outliers of the numerical variables. We consider outliers those points that are 1.5 IQR under Q1 or 1.5 IQR over Q3. We find that "RestingBP" and "Oldpeak" have a low percentage of outliers (3.5% and 1.8%), and thus we can delete these data points without a major impact on the model.

For the case of *Cholesterol*, we find that over 20% of the points are outliers. If we have a closer look at this feature (Figure 4), we find that in fact an 18.7% of the observations has Cholesterol=0 . We suspect that these might be error measures because it is not possible to have 0 cholesterol. Therefore, during preprocess if a data point has null *Cholesterol*, the value is substituted by the mean *Cholesterol* of the training set. To assess this decision we trained the Logistic Lasso Regression with different feature sets including the original *Cholesterol* feature and the preprocessed version. We find that the model performs better after this preprocess step.

As part of the Exploratory Data Analysis we look at the frequency distribution of all features and the distribution of the output variable.

For categorical features (Figure 1) for every category we plot frequency and the positive ratio of the target variable. We can observe remarkable traits of the data. For instance, *FastingBS, ExerciseAngina* or *ST Slope* look like potential good predictors for *HeartDisease* because they have big difference in positive rate among their categories.

For numerical features , we look at the histogram of every feature and the positive rate per bin (Figure 2). We see, for example, that between the age of 40 and 60, the older the higher the propensity to suffer from heart disease. For the analysis of this variation of the positive rate it is important that we also look at the frequency of the respective group (bin or category). Note that outliers are not excluded for these plots.

Last, to further infer the predictive potential of the features, we look at the pairwise correlation matrix of the numerical predictors.(Figure 3). We can observe that the features with the strongest correlation to the target are *Oldpeak* and *MaxHR*.

Before training the models we preprocess the data in different steps. First we clean the *Cholesterol* feature as introduced above: if 0, we substitute it by the mean value of the training set. Second we remove outliers. Third we convert categorical variables into dummies. Last we standardize the data by subtracting the mean and scaling to unit variance.

	Observations	**Positives**	**Positive rate**
Train + Validation	734	398	0.542
Test	184	110	0.598
Total	918	508	0.570

Table 1: Distribution of observations and positives.

	Outliers	**%**	**Lower bound**	**Upper bound**
Age	0	0.0	27.5	79.5
RestingBP	26	3.5	90.0	170.0
Cholesterol	148	20.2	30.1	409.1
Oldpeak	13	1.8	-2.2	3.8

Table 2: Summary of outliers.

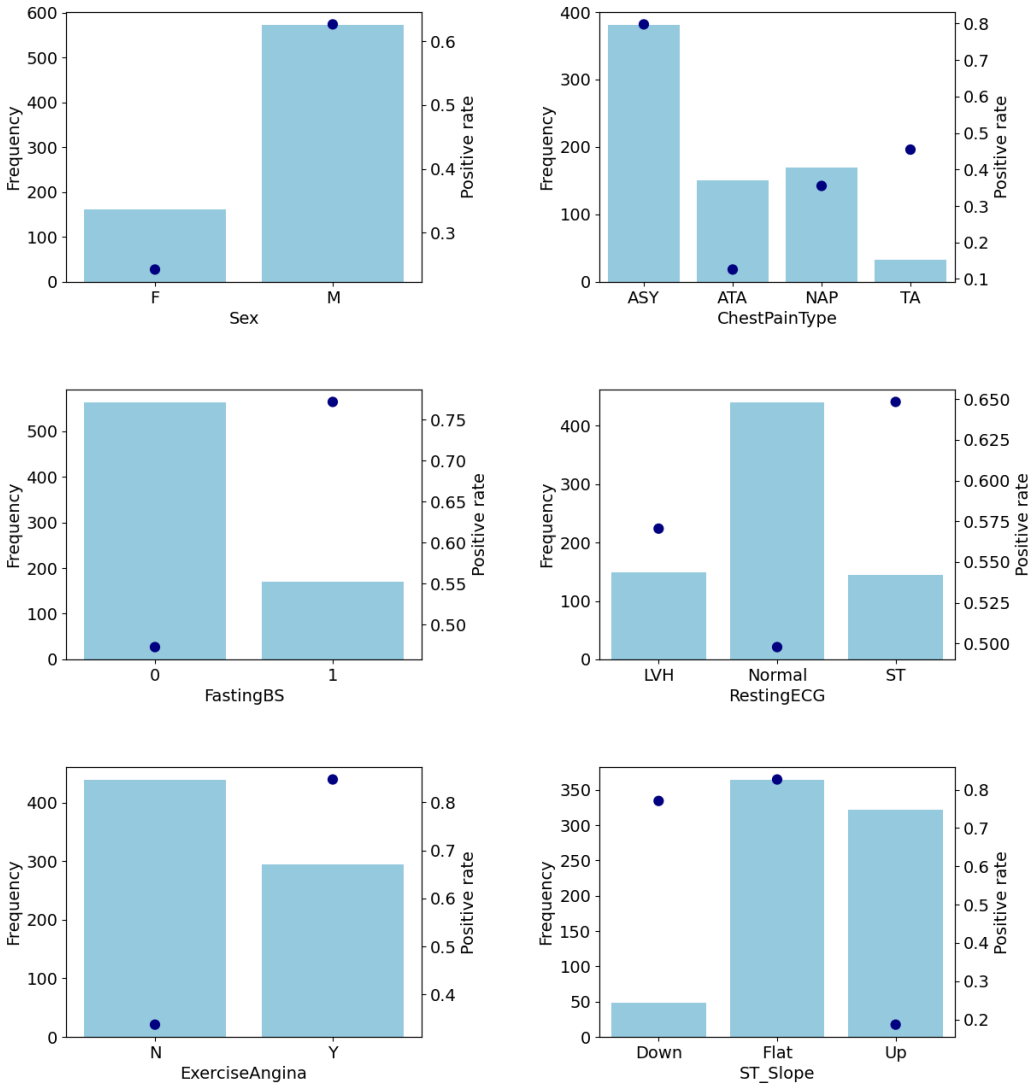


Figure 1: Frequency distribution of categorical variables and rate of positive output per category.

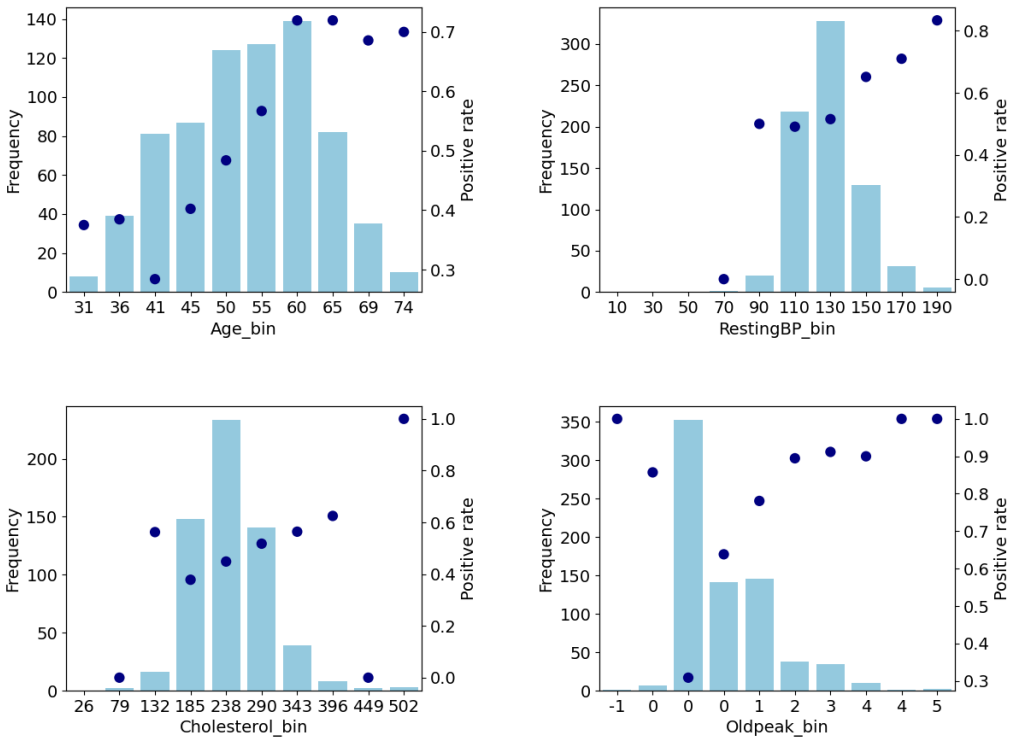


Figure 2: Frequency distribution of numerical variables and rate of positive output per bin.

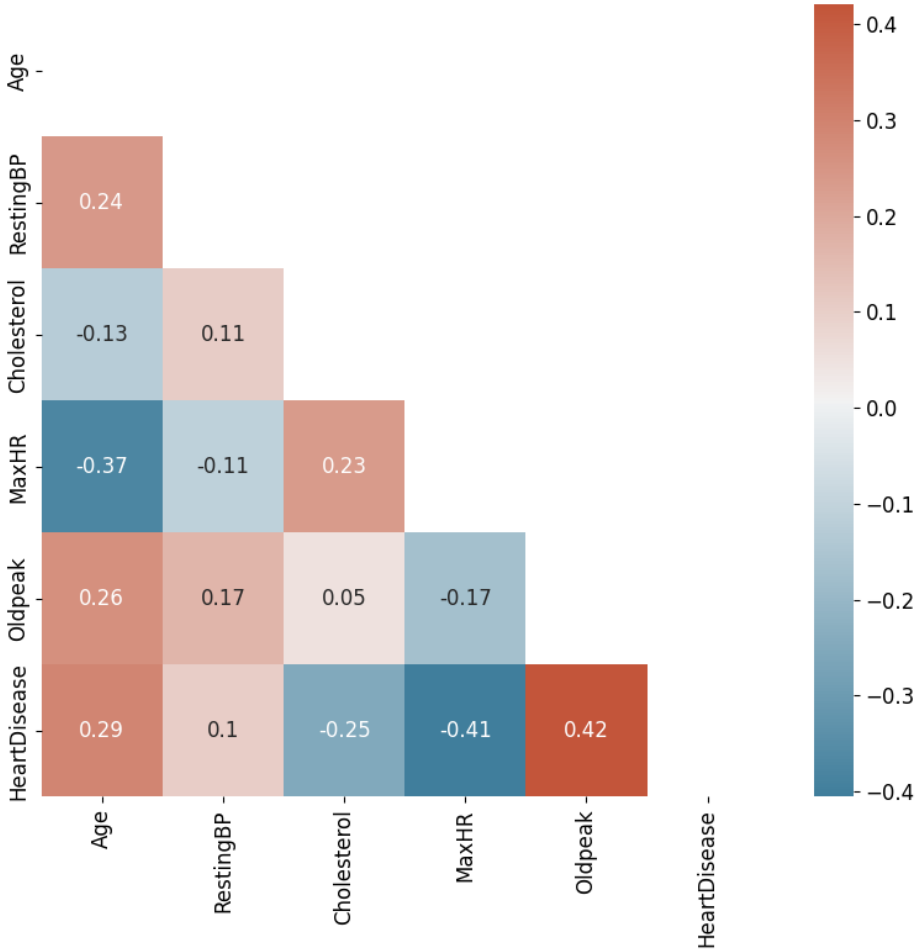


Figure 3: Pairwise correlation matrix.

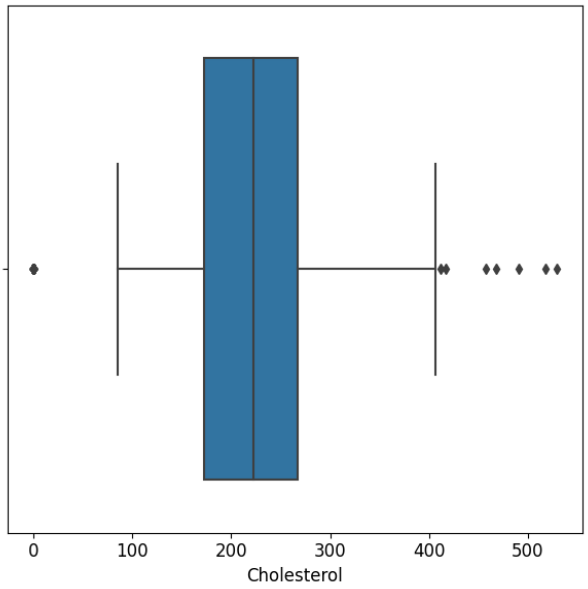


Figure 4: Boxplot distribution of "Cholesterol" feature.

2. Logistic Lasso Regression

The first used algorithm for this regression task is Logistic Lasso Regression. As already mentioned in the EDA section, we try out different feature sets including different information of *Cholesterol*. We decide to keep the preprocessed version of the feature *Cholesterol_clean*. The feature importance as given by the coefficients of the regression is visualized in Figure 5.

We select the five most important features (*ChestPainType_ASY, Sex, ST_Slope_Flat, ST_Slope_Up, FastingBS*) to run the final version of the model. The same feature selection will be used for the Decision Trees and MLP models.

We decide to keep this reduced set of features because it improves the interpretability of the model without negatively affecting its performance. We assessed this by comparing cross-validation with the complete set of features and with the reduced set. Moreover, if physicians were to use this model, the predictor features will have to be reported for each patient and it is therefore preferable to have fewer features.

Finally, we fit this model on the entire train set and predict the outcome of the test set.

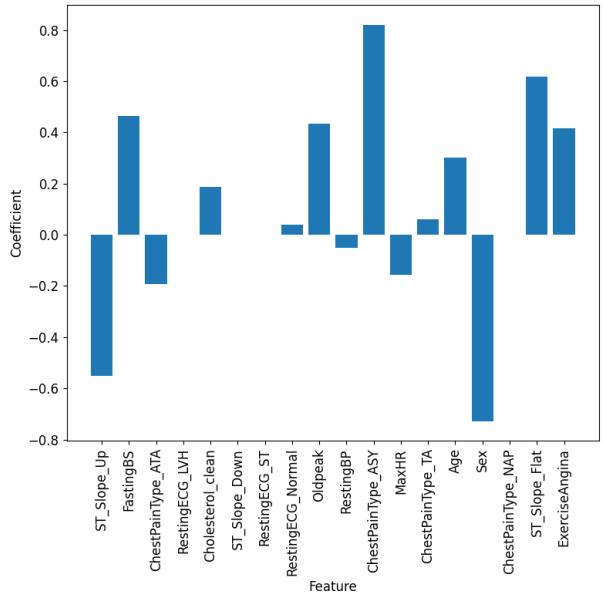


Figure 5: Feature importance

3. Decision Trees

We train a Decision Tree model with the selected five most important features according to the Lasso model. After tuning the hyperparameters we find that best results are obtained with *min_samples_leaf = 1*, *min_samples_split = 2*. Because we already use a reduced feature set, there is no need to tune other hyperparameters as *max_depth* or *max_features*. The feature importance given by the Gini coefficient is shown in Figure 6. It shows that *ST_Slope_Up* is the feature that achieves the highest impurity reduction in the decision tree. As before, the performance metrics on the validation and set can be found in Table 3.

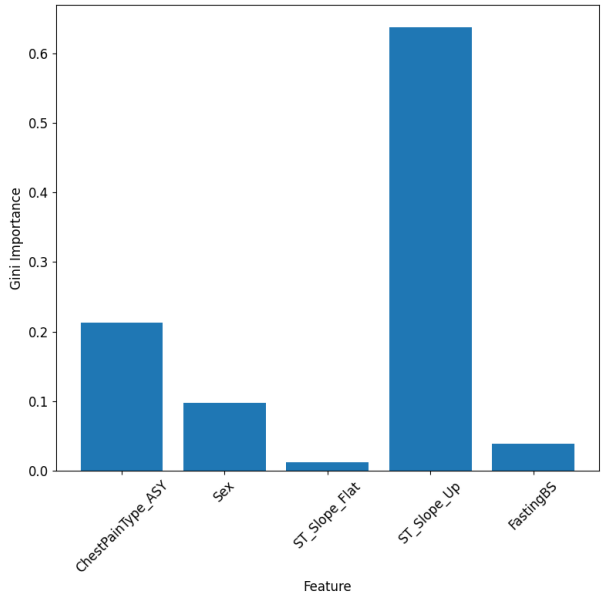
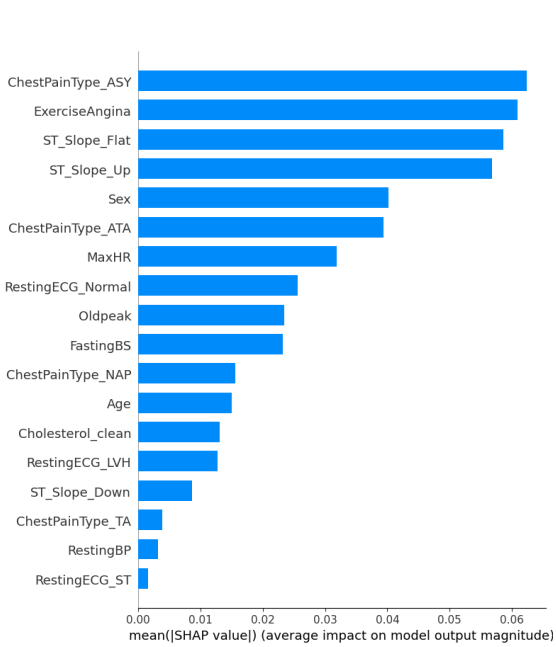


Figure 6: Feature importance

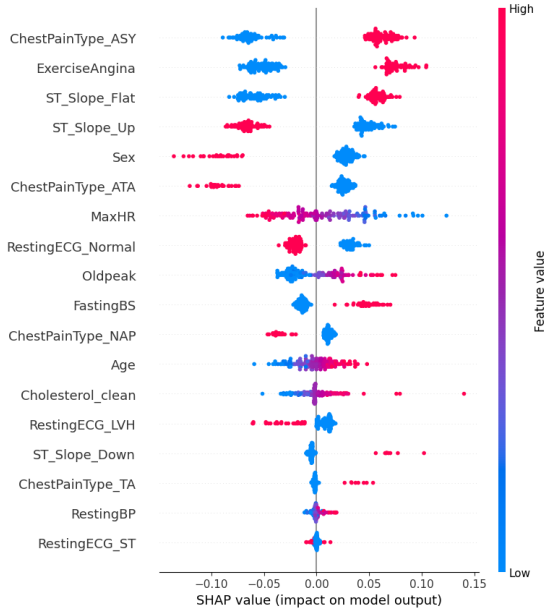
4. Multi Layer Perceptron

We train a Multi Layer Perceptron (MLP) on the same dataset and compare performances. The network is trained with the AdamW optimiser and a binary cross entropy loss. The MLP uses the ReLu activation function. We evaluate different hyperparameters and use grid search with a fixed train-validation split to search for optimal hyperparameters. The hyperparameters include the learning rate ($1e-3, 1e-4$ or $1e-5$), the batch size (8 or 16) and the architecture. For architecture we evaluate single layer perceptrons with hidden layer sizes 8, 16 and 32. Further we explore the use of two layer perceptrons with layer sizes (16, 8) and (32, 16). We repeat this experiment both for the full set of preprocessed features as well as just the features selection made by the Logistic Lasso Regression. For the reduced feature set we have determined the best hyperparameters of a single hidden layer with 8 nodes, a learning rate of 0.001. This model is called *MLP (FS)*. For the full set of features the optimal hyperparameters are two layers of sizes 16 and 8 with a learning rate of 0.0001. The model trained on the full feature set is called *MLP (Full)*. The optimal batch size is 16 for both models.

We analyzed the SHAP values of the final MLP (Full) model. Figure 7 depicts the global feature importance. We would like to highlight that there is a significant overlap of 4 out of 5 features between the top 5 most important features obtained from the SHAP values and those selected by Lasso Regression. Therefore, we conclude that the feature importance is consistent across different methods. When observing the point cloud, it is evident that the top 5 features, all binary, have a considerable impact on the model's output. For instance, if *ChestPainType_ASY* is positive, this has a definite positive effect on the output for all training data points. Furthermore, we observed a stable impact of the scalar feature *MaxHR* on the output value. The higher the *MaxHR* gets, the lower the chances of the model outputting a positive value. Lastly, upon inspecting the *Sex* feature, which takes the value of 1 if the patient is a woman, we notice that men seem to be more susceptible to heart diseases, based on what the model has learned.



(a) Absolute feature importance



(b) Point cloud of feature values and their impact.

Figure 7: Global feature importance based on SHAP values.

Additionally we plot the local feature importance for 4 positive and 4 negative samples in Figure 8 and 9 respectively. When examining the positive samples, it is noticeable that the top 4 features are consistent. Interestingly for the first 3 samples *FastingsBS* has a positive effect on the model output, while for the last one it slightly pulls the output to 0. For all samples *ExerciseAngina* is the most important driver. Also all but the last sample have only features with positive effects on the output among the top features. This reflects in the final model output that is slightly lower for the last sample. One might conclude that the model is less certain due to the values of *RestingECG_Normal* and *FastingsBS*. Looking at the negative samples the picture looks slightly different. It appears that for our sampled set of healthy patients the effect of features is much more balanced, as especially the top right and bottom left patient have features with both positive and negative influence among their top features. This also reflects in the final prediction of the model. For both the *MaxHR* seems atypicall and pull the model more towards a positive prediction. The effect of the *ExerciseAngina* feature on negative features seems to be much less pronounced.

1.1 Results

The performance metrics are reported in Table 3, together with the metrics for the validation set. Interestingly the MLP does not bring an improvement compared to the more simple models. We hypothesise that the MLP overfitted on the validation set as it is the best performing model on the validation set but falls behind on the test set. Lasso outperforms all other methods on the test set, both on F1 score and Balanced Accuracy. Further it is interesting to see that the MLP (Full) performs slightly worse on the validation set than MLP (FS). On the test set MLP (Full) is better but still lies behind Lasso. We conclude that training on the full set of features does not improve performance significantly. Based

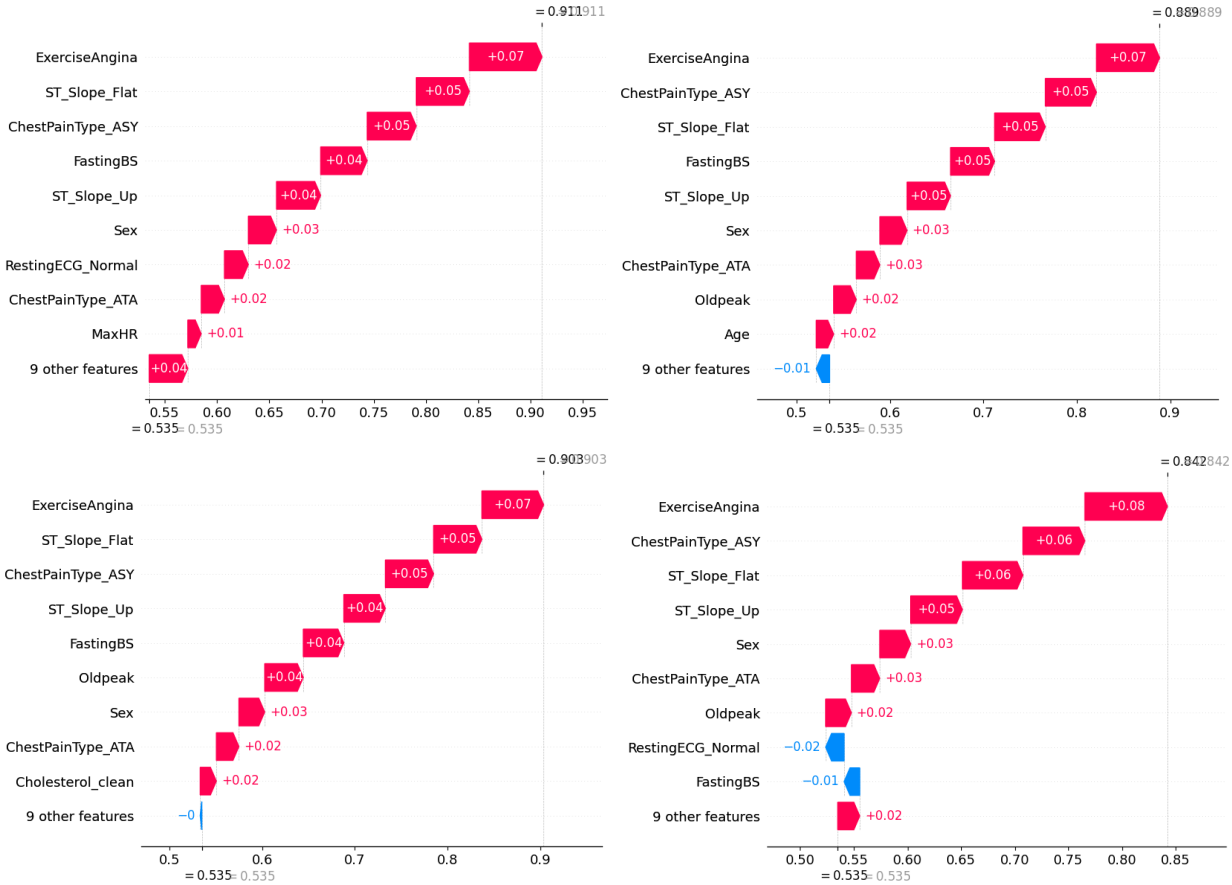


Figure 8: Local feature importance for 4 positive samples of the test set.

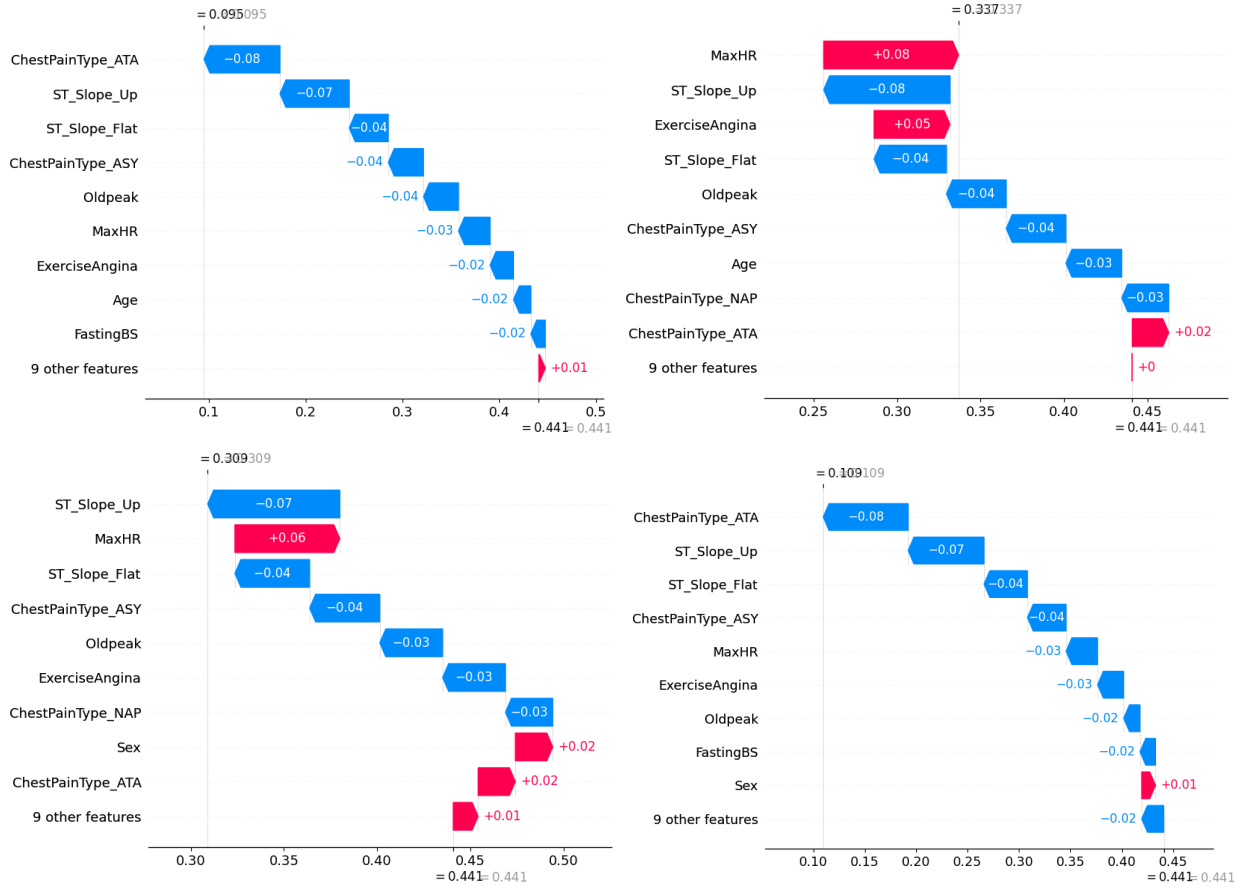


Figure 9: Local feature importance for 4 negative samples of the test set.

on our analysis, it appears that the optimization of the function space is not particularly complex given the available data. Thus, adding complexity through the use of MLPs does not appear to lead to improved performance. We therefore suspect that a larger amount of data would likely be necessary to improve the predictive performance.

	Validation set		Test set	
	B. Accuracy	**F1 score**	**B. Accuracy**	**F1 score**
Lasso	0.862	0.873	**0.821**	**0.861**
Decision Trees	0.825	0.855	0.803	0.848
MLP (FS)	**0.877**	**0.876**	0.810	0.835
MLP (Full)	0.869	0.869	0.812	0.853

Table 3: Performance metrics (Balanced Accuracy and F1 score) of the models when predicting the validation and test sets of the reduced features.

2 Pneumonia Prediction Dataset

2.1 Exploratory Data Analysis

The data set contains a training set of 5216 chest X-ray images, a validation set of 16 images and a testing set of 624 images. There are almost three times as many images of

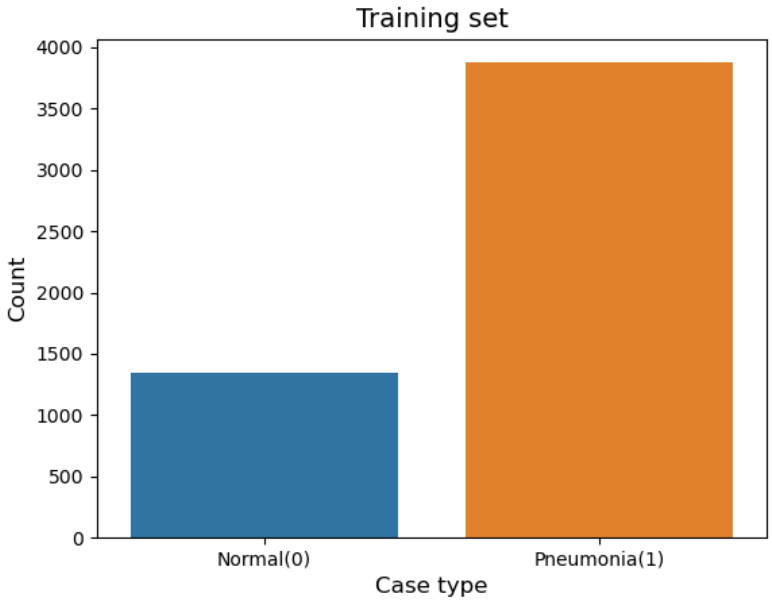


Figure 10: Original training set classes

patients with pneumonia as there are images of normal patients. The data is therefore strongly unbalanced towards category 1 (pneumonia) and the validation set is too small to be representative. We then chose to merge the training set and the validation set to create a new validation set containing more images.

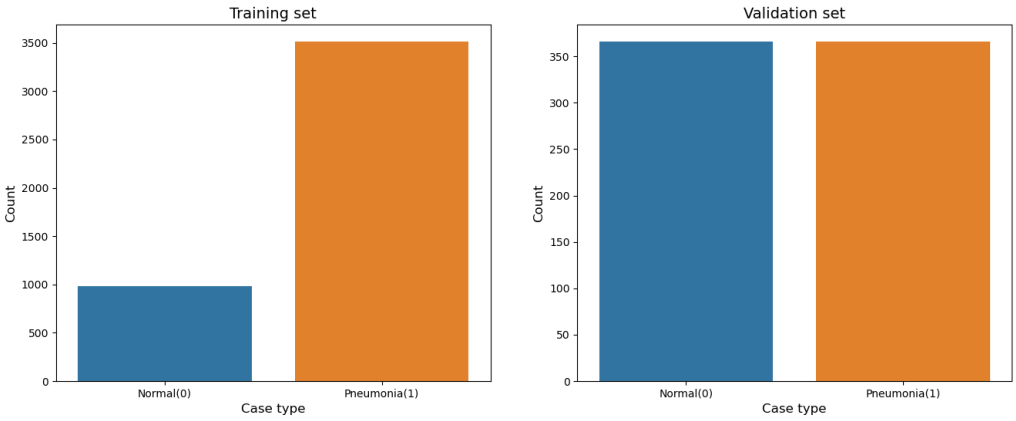


Figure 11: The new training and validation set

The original count of labels in the training set is given in Figure 10 and new count of labels is given in Figure 11. In order to balance the deficit of images in category 0 (normal) we decided to increase the number of images by applying random rotations and symmetries to all the images. Then we sample more regularly in the images of the normal category than in the pneumonia category when training the neural network.

The difference between the images of normal and pneumonia patients is often quite difficult to make, however, it is generally noticed that patients with pneumonia have images that appear to be more obstructed within the thoracic cavity. There is then the presence of a white veil on the image.

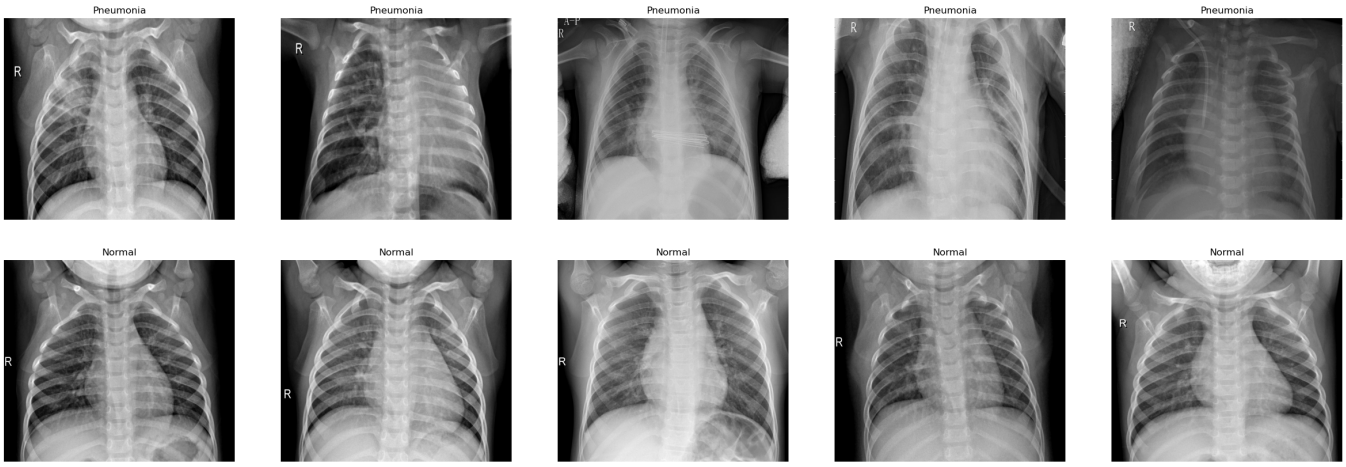


Figure 12: Examples of Pneumonia and Normal samples

2.2 CNN Classifier

For the CNN, we have chosen a small and simple architecture (Figure 13).We have augmented the images as explained above and resized all images to 3x224x224 pixels. To combat the class imbalance we used weighted sampling during training. This ensures that the model sees the same amount of positive and negative samples. We use pretrained VGG16 weights and fine tune them on our training set. As the criterion we use the CrossEntropy and Stochastic Gradient Descent for the optimization algorithm. The CNN is based on the VGG16 architecture, starting with three blocks of Convolution layer and MaxPooling layer with ReLu activation functions. For the classifier part we use two Fully Connected layers with Dropout to reduce overfitting.
The analysis of our model (Table 4) still shows some overfitting on the validation set and

Table 4: Performance Metrics of the CNN Model

	Precision	Recall	F1 Score	Accuracy
Validation set	0.99	0.96	0.97	0.97
Testing set	0.75	0.99	0.85	0.78

hence lower performance on the test set. The results remain good overall because the metric to look at here is the F1 score due to the strong asymmetry of the classes. However, there is a bias towards the over-represented class in the testing set.

```
==========================================================================================

Layer (type)                            Output Shape           Param #
==========================================================================================

SimpleCNN                               [1, 2]                 --
+ Conv2d                                [64, 224, 224]         1,792
+ ReLU                                  [64, 224, 224]         --
+ MaxPool2d                             [64, 112, 112]         --
+ Conv2d                                [64, 112, 112]         36,928
+ ReLU                                  [64, 112, 112]         --
+ MaxPool2d                             [64, 56, 56]           --
+ Conv2d                                [128, 56, 56]          73,856
+ ReLU                                  [128, 56, 56]          --
+ MaxPool2d                             [128, 28, 28]          --
+ Linear                                [1, 512]               51,380,736
+ ReLU                                  [1, 512]               --
+ Dropout                               [1, 512]               --
+ Linear                                [1, 2]                 1,026
+ Dropout                               [1, 2]                 --
==========================================================================================

Total params: 51,494,338
Trainable params: 51,494,338
Non-trainable params: 0
Total mult-adds (M): 871.17
==========================================================================================

Input size (MB): 0.60
Forward/backward pass size (MB): 35.33
Params size (MB): 205.98
Estimated Total Size (MB): 241.91
==========================================================================================
```

Figure 13: Model summary

2.3 Integrated Gradients

The map highlights important areas that appear to be inside the rib cage. This map is consistent across the different examples. However, the difference in the map between normal and pneumonia patients is not clear. Both seem to give importance to the ribs and the interior of the rib cage but it would require the opinion of a specialist in the field to validate the relevance of the selected areas.

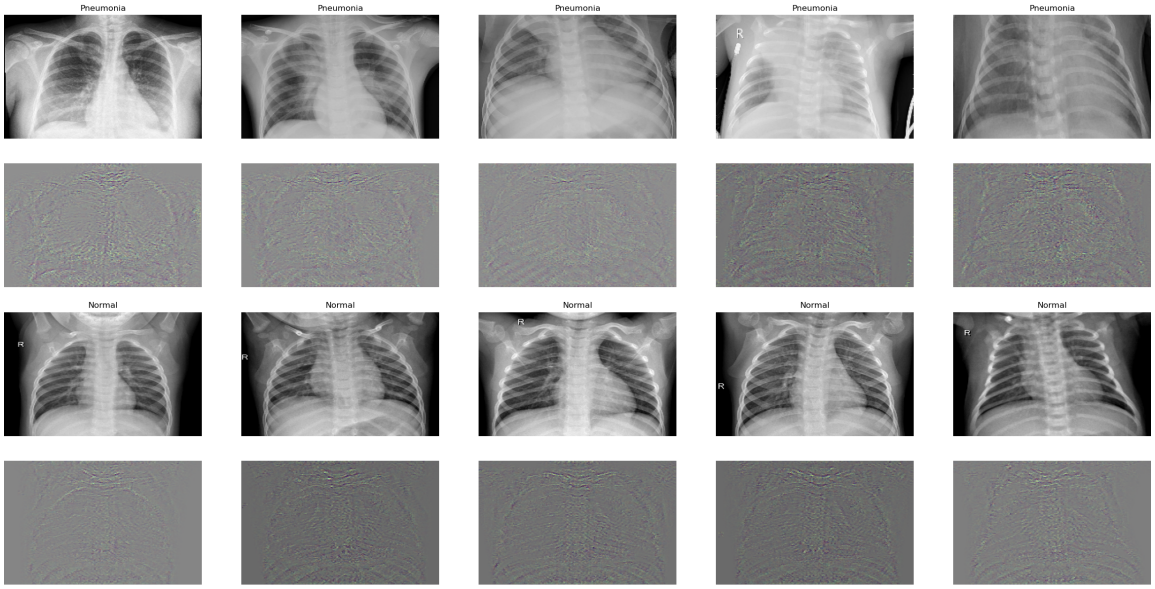


Figure 14: Integrated gradient of five examples of both categories

2.4 Grad-CAM

The results of the grad Cam (Figure 15) are quite different from the results of the gradient integration, as difference between the two categories is much more pronounced. For the pneumonia category the important areas seem to be outside the rib cage while for the normal category the ribs seem to be more important. This confirms our hypothesis in question three where we suggested that pneumonia should obscure the vision of the ribs by adding a white veil on the image inside the rib cage.

2.5 Data Randomization Test

For this part we then re-trained our CNN on a randomized version of our tarining set. We can see a clear difference between the randomized version and the original. At first, the gradient integrator (Figure 16) does not seem to know which part to focus on and seems to be interested in the whole body structure of the patient. The Grad-CAM (Figure 17) method has a null gradient on several images and seems to act as an edge detector on others (highlighting among others the "R" logo present on some images). The reason why the gradient is sometimes 0 during the evaluation comes from the fact that the neural network trained on mixed labels is not able to classify the images well and tends to classify all images as class 1. This results in a lack of gradient when using Grad-CAM.
These maps are very different from the original ones and let us think that the original model does not just act as an edge detector but as a classifier that understands the important areas to look at.

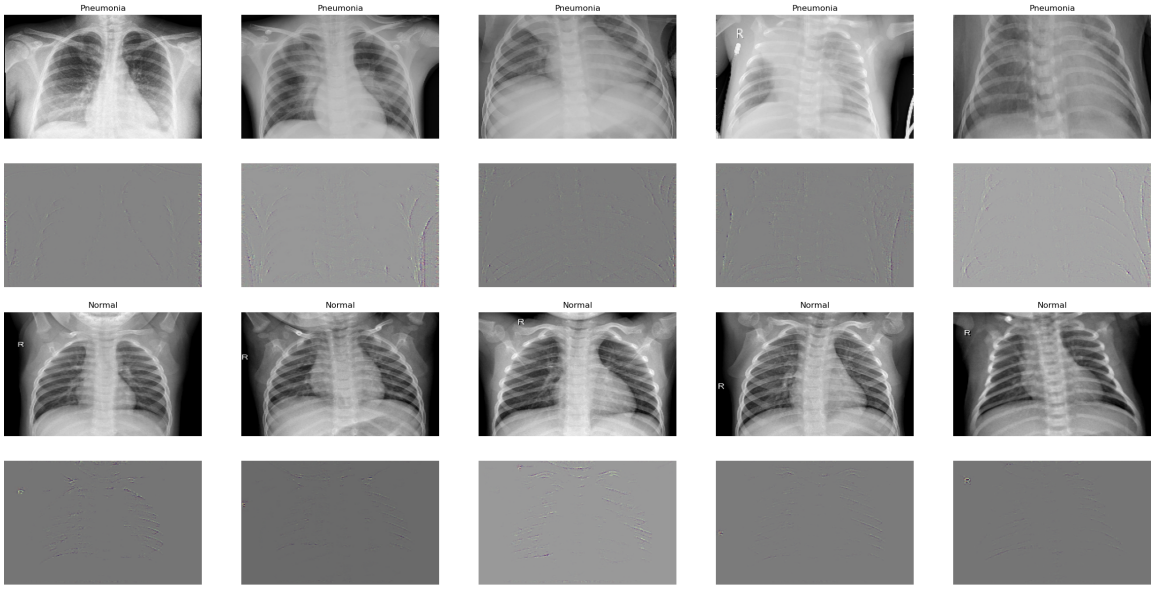


Figure 15: Grad-CAM of five examples of both categories

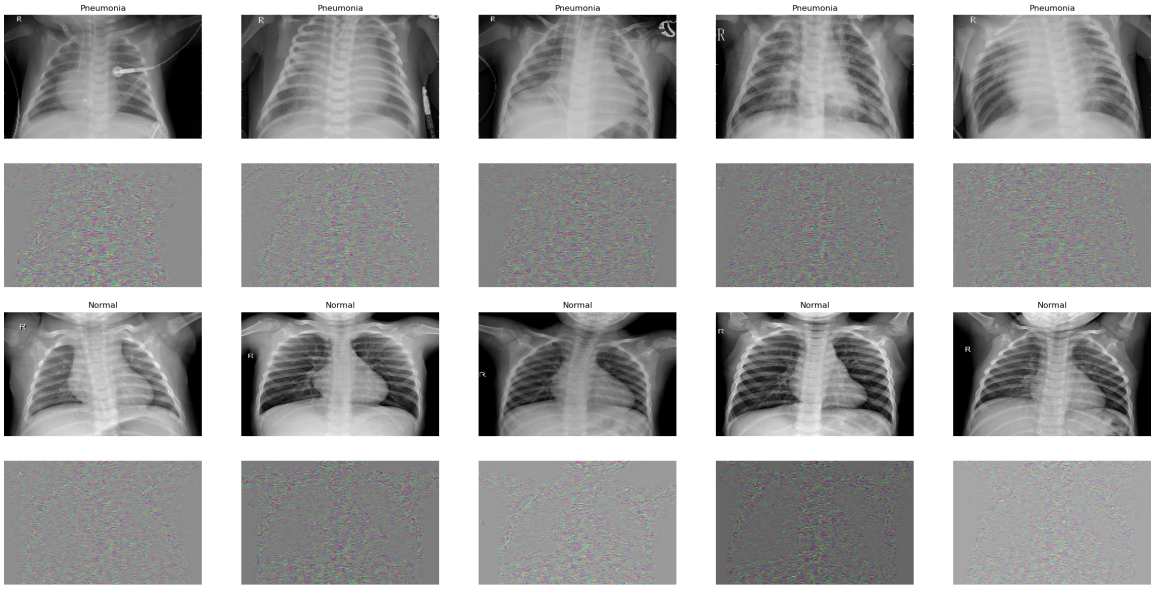


Figure 16: Integrated gradient on randomized data of five examples of both categories

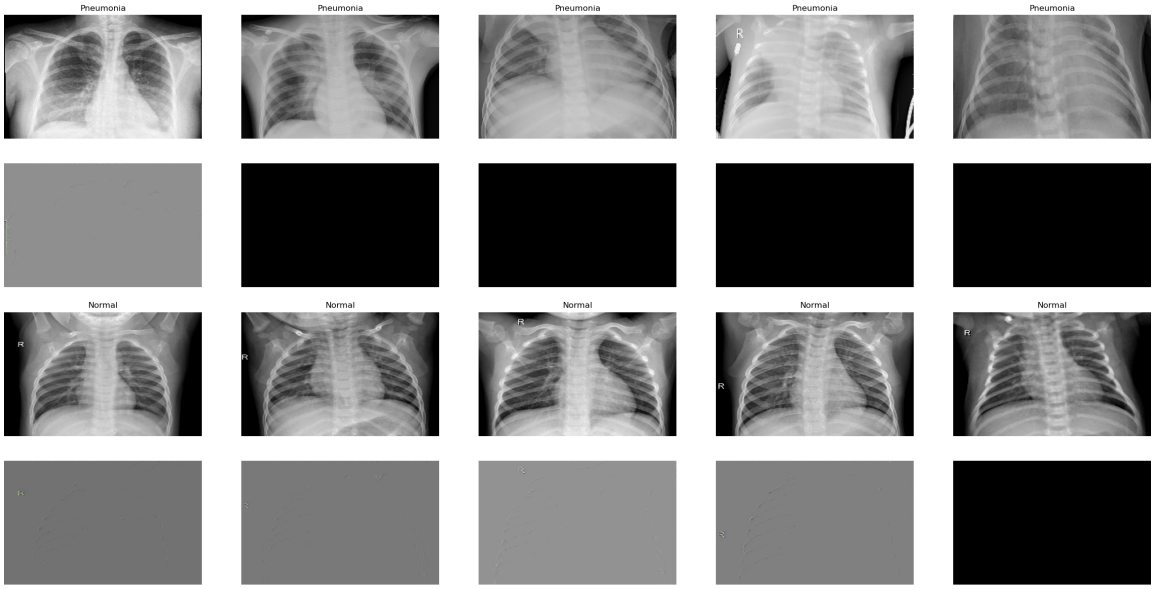


Figure 17: Grad-CAM on randomized data of five examples of both categories

2.6 Challenge 2: Prototype Selection

We examine the use of prototypes for improved interpretability on the Pneumonia Prediction Dataset. For this we have implemented an efficient version of the methods proposed by [1]. We compare the performance of random prototype selection to their proposed maximum mean discrepancy (MMD) method for greedily finding good prototypes. Further we explore extensions to their methods that enable better performance on our dataset.

2.6.1 Methodology

Our preprocessing pipeline involves converting the input images to grayscale and resizing them to 128×128 as per the project sheet's guidelines. Due to the dataset's imbalance, we propose a new method for prototype computation called *separate* prototype computation, which ensures that half of the prototypes come from each class. The computation of prototypes is performed separately for each class and then merged to obtain the final set of prototypes.

Since the prototype computation method is based on the distances of raw pixel values, we suspect that slight variations in the input images may have a substantial impact on the method's performance. To address this issue, we investigate the effect of *histogram equalization* on input images, which we hope will reduce input noise.

The MMD-based prototype computation requires large RBF kernel matrices to be calculated. To overcome this computational hurdle, we developed an implementation that can utilize precomputed kernel matrices and compute scores for prototype candidates in batches.

This implementation, combined with parallelization, allows for more extensive hyperparameter search. Our search was conducted with cross-validation on five folds and a fixed m^* of 2^5 prototypes. For random prototypes, we sampled five sets for each hyperparameter configuration.

2.6.2 Results

For both random and MMD prototype selection we set $gamma = 1e-6$ based on the validation results from the grid search. Refer to Table 5 for details on the MMD hyperparameters. Similar to the findings of [1] we found no significant benefit of using localized kernel matrices. Further it is to highlight that our proposed *separate* prototype computation significantly improves the validation performance.

	separate	not separate
local	0.92	0.88
not local	0.92	0.88

Table 5: Cross validation f1-score for MMD hyperparameters with fixed $m^* = 1e-6$

Figure 18 illustrates the test set performance of the proposed methods for different values of m^*. The baseline performance of always predicting pneumonia is also shown for comparison. The random methods were evaluated by sampling 10 times a random set of prototypes and averaging the results. As expected, across all methods, increasing the number of prototypes leads to better performance. The MMD method generally outperforms the random methods, except around $m^* = 2^5$, which we attribute to the noise in the random methods. This is to expect because, simplified, the greedy MMD algorithm chooses prototypes such that their position in the pixel space is most representative of all the training images. It should therefore perform better than random prototypes. Interestingly the standart MMD performs poorly for very low prototype counts. In contrast by equalising the images MMD performs very well and reaches an $f1 - score$ of 0.829 for only 4 prototypes. We hypothesis that due to noise in the input images equalising plays an important role. The more prototypes we allow the less important the influence of equalisation gets and at after 2^5 the unequalised MMD clearly outperforms the equalised MMD. At this point the higher number of prototypes is probably able to capture the input noise and can even benefit from it. Refer to Table 6 for detailed f1 scores for $m^* = 4$ and $m^* = 128$. It is important to note that the performance of the proposed methods is still far behind that of CNNs. This is to be expected since CNNs are more complex and can capture local features better than the prototype-based methods.

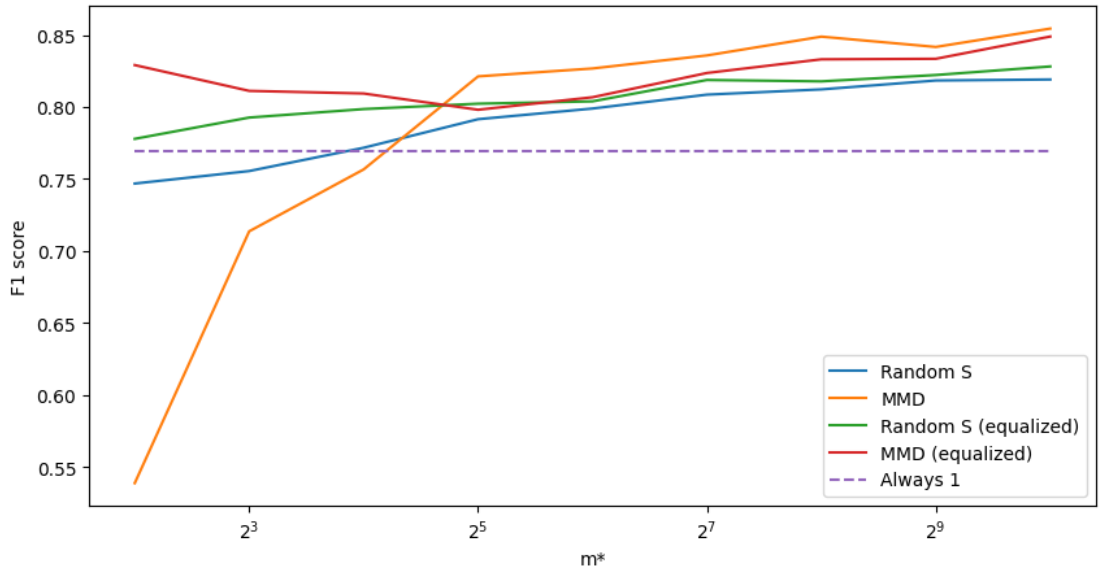


Figure 18: F1-scores on the test set. X-axis is the number of prototypes m^*.

	MMD	MMD (equalised)	Random S	Random S (equalised)	Always 1
$m^* = 4$	0.538	**0.829**	0.746	0.778	0.769
$m^* = 128$	**0.835**	0.827	0.809	0.818	0.769

Table 6: F1-scores on the test set.

2.6.3 Interpretability Analysis

Figure 19 presents the first 5 positive and 5 negative prototypes selected by the MMD (equalised) method. It is important to note that this approach does not exhibit shift-invariance, meaning that it only exploits absolute global pixel values and not local properties. This characteristic is reflected in the set of prototypes, which exhibit varying positions. For example, Negative 1 is slightly tilted to the right, while Negative 3 leans to the left. Furthermore, the Pneumonia prototypes display different levels of zoom and chest forms. For instance, Pneumonia 3 appears more zoomed out, while Pneumonia 1 is more zoomed in. Notably, Pneumonia 3 and Pneumonia 4 exhibit hands-down positions, whereas the other prototypes display hands-up positions. It is also interesting to note that there is no Normal prototype with hands-down position. Additionally, all Pneumonia prototypes display shadows over the rib cage, particularly when compared to the Negative prototypes. Overall, we argue that these prototypes are meaningful.

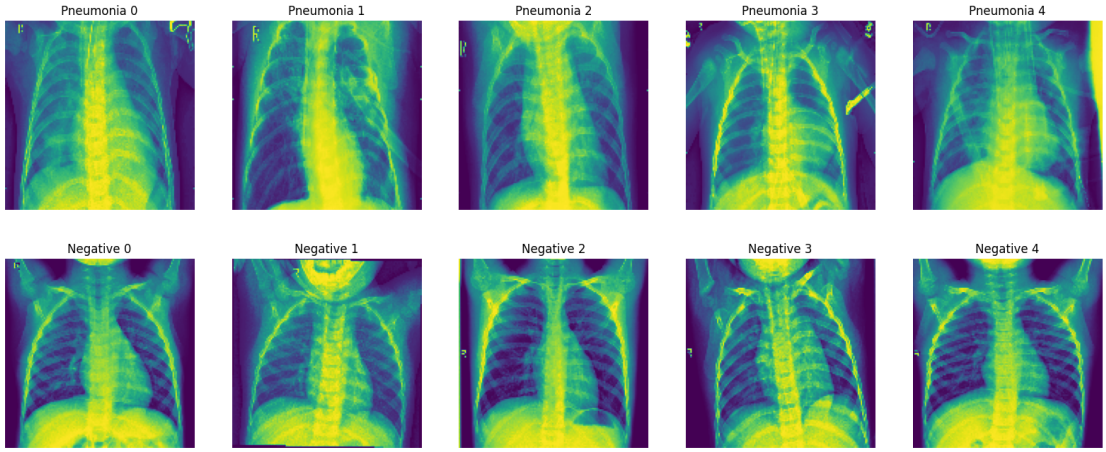


Figure 19: Top 5 prototypes selected by MMD (equalised) for both positive and negative samples.

2.6.4 Comparison to other methods and extensions

The prototype approach's ability to provide users with a similar image from the training set can be particularly useful in practice, especially for doctors who may not have extensive technical expertise. The simplicity of the approach's prediction mechanism makes it easy to explain to non-technical staff, as it simply returns the closest image from a set of prototypes. However, despite these advantages, the prototype approach exhibits poorer performance compared to deep learning approaches. Thus, CNNs with saliency maps may still be the preferred choice in practice, as they further have the added advantage of being able to highlight specific regions of an image. By applying post-processing techniques to saliency maps, important areas of an image can be highlighted, allowing doctors to visually verify potential suspicious features. In contrast, the prototype approach only provides a full image, making it more challenging for doctors to identify potentially dangerous regions that require closer examination.

The prototype approach exhibits a genuine drawback in that it lacks the capacity to capture local features. In contrast, CNN layers possess the ability to detect local features, irrespective of their location within an image. This shift-invariant property enables CNN layers to learn a feature at one position and recognize it at any other position in an image. Unfortunately, the prototype approach lacks this shift-invariance and requires one prototype for each global feature position. To overcome this limitation, images could be patchified into smaller regions such as 32×32 patches with overlapping regions. Consequently, prototype patches can be computed instead of prototype images. To classify a new image, a majority vote could be performed on all patches. This extension could also allow users to identify healthy and sick areas/patches of an image instead of a global prediction, similarly to what saliency maps are able to provice. However, it incurs additional computational overheads. Therefore, further exploration of this approach is deemed intriguing.

3 General Questions

3.1

Part 1: The performance of the different models is very similar. As per the feature importance, results are consistent for the interpretable and explainable methods. According to the Lasso weights, the features that are better at classifying are *ChestPainType, Sex, ST_Slope and FastingBS*. Out of the top 5 features for both the MLP as well as the Lasse Regression, 4 overlap.

Part 2: For the second part there were only post-hoc methods (explainability) but both methods don't really seem to identify the same parts of the images. Further it is very hard to compare the saliency maps to prototypes as this is a completely different approach.

3.2

Part 1: Among the models trained, the functioning of Lasso Regression is probably the easiest to understand for a broad audience. We would thus opt for explaining how Lasso does the classification using the five selected features. Additionally we would show the feature importance and explain that it corresponds to the weight given to each feature by the model.

Part 2: We would explain Grad-CAM puts forward the important parts for the classification decision of the neural network. It seems to focus on the ribs when these are more easily visible and the patient is in good health. And it focuses more on the rest of the image when the center is obstructed.

3.3

Part 1: We opted for including only the five best features because this improves interpretability and explainability of the model. Even if the accuracy would be slightly reduced, we also need to care about how the model will be best used. If doctors only need to report four features[1] of the patient, it is more likely that they will use and benefit from this tool. Further we have shown that training on the full feature set does not even bring big advantages. We suspect that this is due to more noise that comes with more features.

Part 2: There is no real tradeoff when using explanability methods because they are ad-hoc. However the size of the neural network influences the time needed to evaluate these methods and there is a tradeoff between very large and preformant models but slower to verify and smaller but faster models. When looking at the prototype methods, there is a clear trade-off. The plots have shown that the more prototypes we use the better the prediction gets. But more prototypes will make it harder to really grasp what is special about a prototype.

[1]Note that 2 out of the 5 features are dummy variables belonging to the same input feature. A doctor would therefore need to fill in 4 values.

3.4

Part 1: We have seen before that *ST_slope* is an important feature. Current medical research is aligned with this and some studies propose the ST/heart rate slope (ST/HR slope) as a ECG criterion for diagnosing significant coronary artery disease. (`https://pubmed.ncbi.nlm.nih.gov/3739881/`). Sex is also known to be an important factor of risk for CVDs. (`https://gh.bmj.com/content/2/2/e000298`).

Part 2: As explained in the presentation : *To diagnose pneumonia, the infected tissue will show denser areas and therefore appear as white spots in the darker background of the lungs.* This is indeed what the Grad-CAM method seems to reveal by highlighting the importance of ribs in the classification of cases in the normal category. Further also all positive prototypes show these denser areas.

3.5

Part 1: As mentioned, the performance of the models is similar. But the models differ in complexity, interpretability and explainability. Looking at this we would choose Lasso Ression because it performs best, it is less complex than MLP and easier to interpret and explain than Decision Trees.

Part 2: We would choose the Grad-CAM method because it seems more likely to reveal differences between the two choices of neural network categories. And that these choices seem to be in phase with the medical knowledge of pneumonia. As the deep neural approaches perform much better than the prototypes, we would not recommend the prototype approaches as they are implemented currently.

References

[1] B. Kim, R. Khanna, and O. Koyejo, "Examples are not enough, learn to criticize! criticism for interpretability," in *Proceedings of the 30th International Conference on Neural Information Processing Systems*, ser. NIPS'16. Red Hook, NY, USA: Curran Associates Inc., 2016, p. 2288â2296.