Machine Learning for Health Care
Interpretable and Explainable Classification for Medical Data
GitHub

Marvin Lob, Moritz Miller, Jakob Ketterer

April 16, 2024

1 Heart Disease Prediction

1.1 Exploratory Data Analysis

1.1.1 Explore the different features, their distribution, and the labels

We were given a heart disease dataset with the following 11 features (description taken from Kaggle):

- Age: age of the patient [years]
- Sex: sex of the patient [M: Male, F: Female]
- ChestPainType: chest pain type [TA: Typical Angina, ATA: Atypical Angina, NAP: Non-Anginal Pain, ASY: Asymptomatic]
- RestingBP: resting blood pressure [mm Hg]
- Cholesterol: serum cholesterol [mm/dl]
- FastingBS: fasting blood sugar [1: if FastingBS $>$ 120 mg/dl, 0: otherwise]
- RestingECG: resting electrocardiogram results [Normal: Normal, ST: having ST-T wave abnormality (T wave inversions and/or ST elevation or depression of $>$ 0.05 mV), LVH: showing probable or definite left ventricular hypertrophy by Estes' criteria]
- MaxHR: maximum heart rate achieved [Numeric value between 60 and 202]
- ExerciseAngina: exercise-induced angina [Y: Yes, N: No]
- Oldpeak: oldpeak = ST [Numeric value measured in depression]
- ST_Slope: the slope of the peak exercise ST segment [Up: upsloping, Flat: flat, Down: downsloping]

and the class label

- HeartDisease [1: Yes, 0: No].

We split the features into numerical and categorical scale and visualized the empirical distributions of numerical features via histograms in 1. Bar charts of the categorical features can be found in 2 and the bar chart of the label in 3.

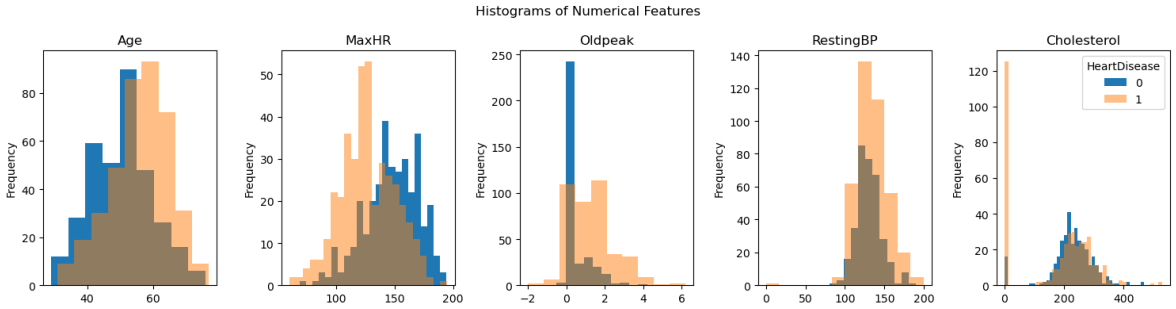


Figure 1: Histograms of numerical features where the color distinguishes the corresponding label

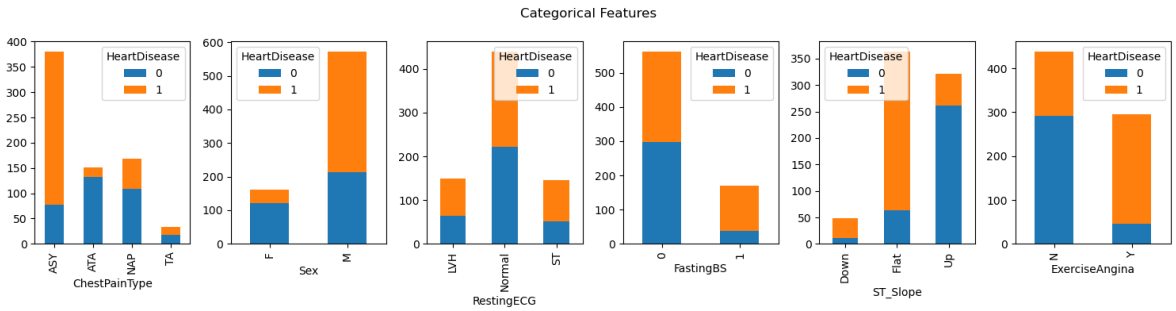


Figure 2: Stacked bar charts for categorical features

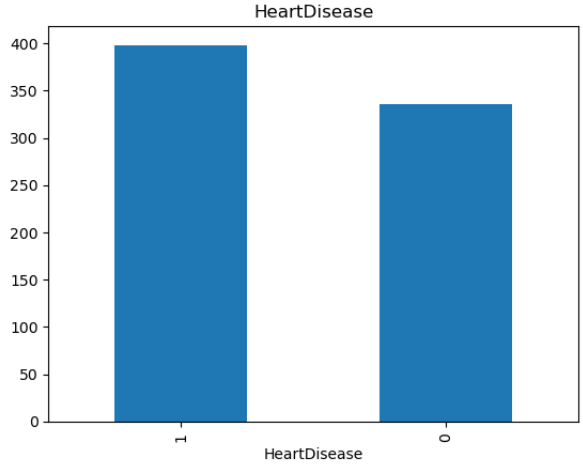


Figure 3: Bar chart for the label of the dataset

1.1.2 Check for common pitfalls like missing or nonsensical data, unusual feature distribution, outliers, or class imbalance, and describe how to handle them

The dataset provided contains no apparent missings. However, the distribution of Cholesterol shows lots of mass on zero which, from a medical point of view, is very unlikely. Interestingly, most of the cases with zero Cholesterol are such with heart disease. The literature (Grundy, 1986) suggests a positive relationship between Cholesterol levels and heart disease. Thus, we assume that Cholesterol zeros are missing data and flag them with a new column "Cholesterol_missing" to include the "missingness" information in the dataset. Sensible imputation approaches would e.g. replace the zero by the median Cholesterol level of the respective disease class on the training data after a train-test-split. We choose the median over the mean as it is more robust. Another feature with nonsensical zero is the resting blood pressure "RestingBP" which we impute analogously.
All categorical features suffer from imbalanced category frequencies. For instance, <5% of cases have the "ChestPainType" = "TA". This is problematic if such cases only appear in the test set, but not in the train set. We thus checked that they occur in both. Alternatively one could form a new category of the rare groups based on a cut off like <5% occurrence.
Our label "HeartDisease" is rather balanced such that we forego increasing the weight of the minority class.

1.1.3 Explain how you preprocess the dataset for the remaining tasks of part 1

As already mentioned, we perform a train-test-split stratified on the label to assess the generalization error on an unseen dataset. Furthermore, we use "One-Hot-Encoding" to encode the categorical features. As we use regularization, we don't drop the first column.

1.2 Q2: Logistic Lasso Regression

1.2.1 Describe which preprocessing steps are crucial to ensure comparability of feature coefficients

Lasso performs L1-regularization via penalizing the coefficients. It thus requires standardization of the features (even the one-hot-encoded ones), so that the penalization scheme is fair to all features.

1.2.2 Fit a Lasso regression model with l1 regularization (1 Pt) on the dataset

We used the class "LogisticRegression" with L1-penalty argument from `scikit-learn` (Pedregosa et al., 2011).

1.2.3 Provide performance metrics such as f1-score or balanced accuracy

Metrics for Logistic Lasso Regression	Score
F1 Score	0.8622
Balanced Accuracy Score	0.8193

1.2.4 Visualize the importance of the different features and argue how they contribute to the model's output.

Lasso Logistic Regression is an interpretable model and we can associate the size of the coefficients with their importance 4. According to the size of the coefficients, "ChestPainType_ASY" and "ST_Slope_Flat" are the two strongest contributors to heart disease. This result is backed by the stacked bar plots in 2 as for both of the feature groups the cases with heart disease were the clear majority. The coefficients for logistic regression can be interpreted as follows: for age we have $\beta_{age} = 0.3034$ meaning that the log-odds for heart disease increase by β_{age} ceteris paribus. Some features were excluded by the Lasso, e.g. MaxHR. For comparison, we also ran Perumtation-Feature-Importance and found that the two most important features are identical 5.

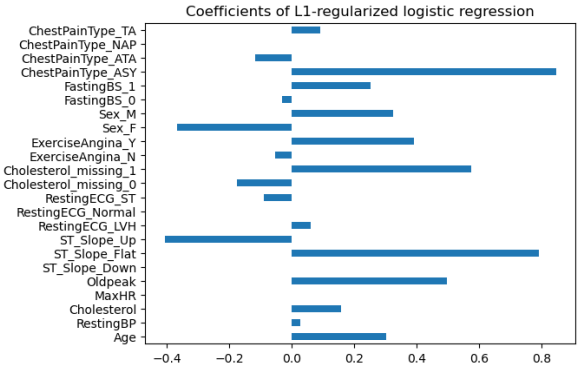


Figure 4: Coefficients of L1-regularized logistic regression

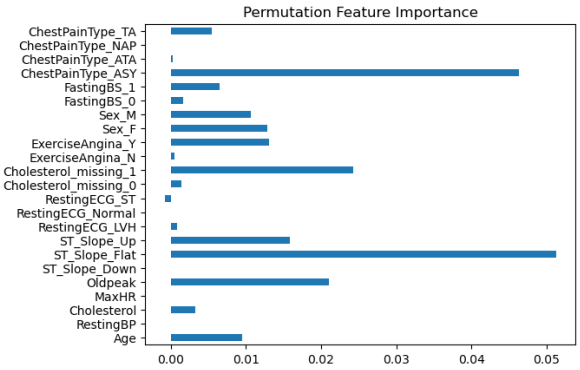


Figure 5: Perumtation Feature Importance

1.2.5 Consider the following setting: A researcher is interested in the important variables and their influence on the label. They have fitted the Logistic Lasso Regression to determine the important variables. Then, they train a Logistic Regression solely on these variables and use this model to make conclusions. Elaborate why this would be a good or bad idea

Our answer is based on Bühlmann and Van De Geer (2011). Lasso yields biased coefficient estimates because of the regularization. If Lasso Logistic Regression uncovered all the true coefficients and all of them are included in the model, the coefficient estimates of the logistic regression are asymptotically unbiased under regularity conditions as they are Maximum Likelihood estimates. However, this is not advisable as the resulting p-values would not be valid as they would not consider the effect of variable selection based on the same data. Ideally, one would split the data, train the Lasso logistic regression on the first half and benefit from statistical advantages of logistic regression after training on the second half.

1.3 Q3: Multi-Layer Perceptrons

1.3.1 Implement a simple MLP, train it on the dataset, and report test set performance

We used the class "MLPClassifier" from scikit-learn with two hidden layers with 128 and 64 nodes but otherwise default parameters and achieved the following performance on the test set:

Metrics for MLPClassifier	**Score**
F1 Score	0.8125
Balanced Accuracy Score	0.7582

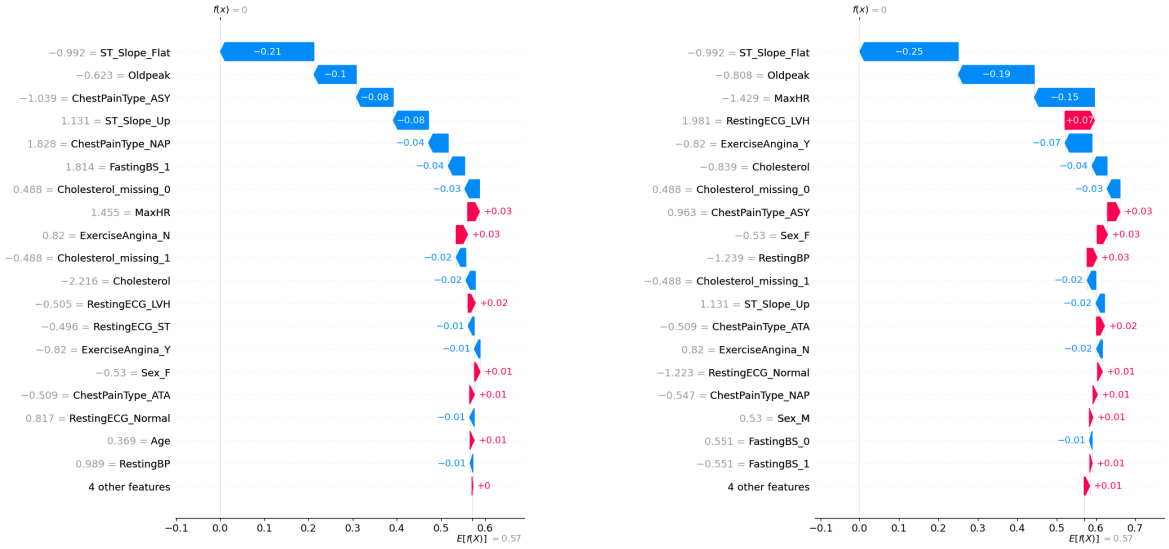


Figure 6: **SHAP explanations of the outputs of two negative samples**

Sample	**Negative Sample 1**	**Negative Sample 2**	**Positive Sample 1**	**Positive Sample 2**	**Overall Importance**
Top 3 Features	ChestPainType_ASY ST_Slope_Flat, Oldpeak	ST_Slope_Flat, Oldpeak, MaHR	Oldpeak, ST_Slope_Flat, RestingECG_LVM	ST_Slope_Flat, ST_Slope_Up, Age	ST_Slope_Flat, Oldpeak, ST_Slope_Up

Table 1: Three most important features for samples and overall

1.3.2 Then, visualize SHAP explanations of the outputs of two positive and negative samples and feature importances of the overall model

Negative samples are visualized in 6, positive samples in 7 and global feature importances in 8. We also visualized a summary plot with all Shapley values for all features in 9.

1.3.3 Are feature importances consistent across different predictions and compared to overall importance values

We look at three most important features for individual explanations and overall feature importance 1: the order and selection of the three most important features is inconsistent across the 4 samples and also not consistent with the overall feature importance:

- e.g. Age, ChestPainType_ASY, RestingECG_LVM appear only once
- e.g. ST_Slope_Up is second most important feature, but only appears once across all 4 samples

However, there is significant overlap among the top features and the most important feature ST_Slope_Flat appears among top three for all samples.

1.4 Q4: Neural Additive Models

1.4.1 Read the paper about NAMs, implement the model, and train it on the dataset

The paper proposing Neural Additive Models Agarwal et al. (2021) was partly written by Google Research employees who provide a PyTorch implementation. We use a fork of this official repository that wraps the code in scikit-learn fashion and provides the class "NAMClassifier". We set num_learners to 1 in order to only fit one network with two hidden dimensions of sizes 64 and 32 per feature. After training for 20 epochs we achieve the following performance:

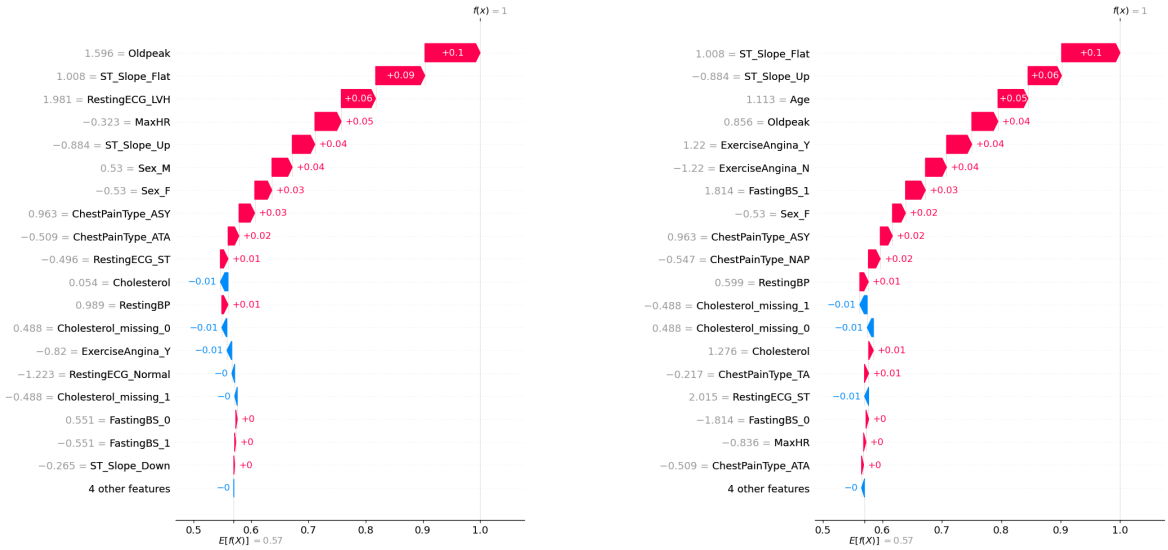


Figure 7: **SHAP explanations of the outputs of two positive samples**

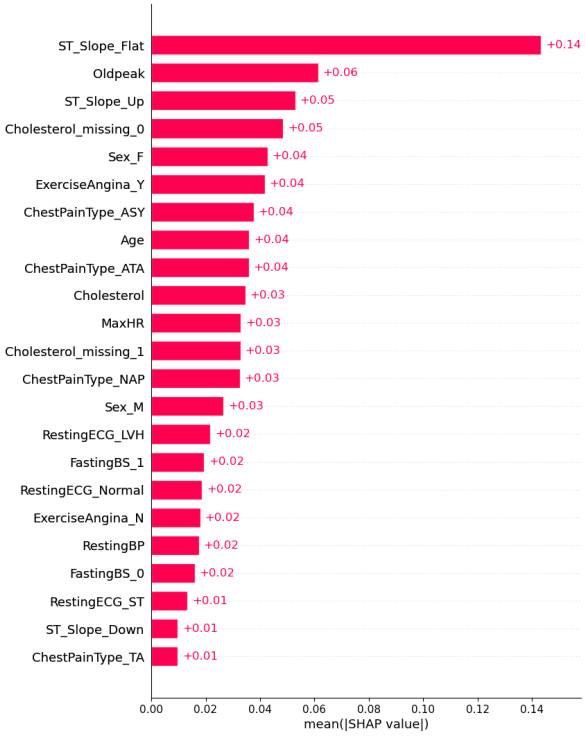


Figure 8: Global feature importance as measured by the mean absolute Shapley value per feature over all samples

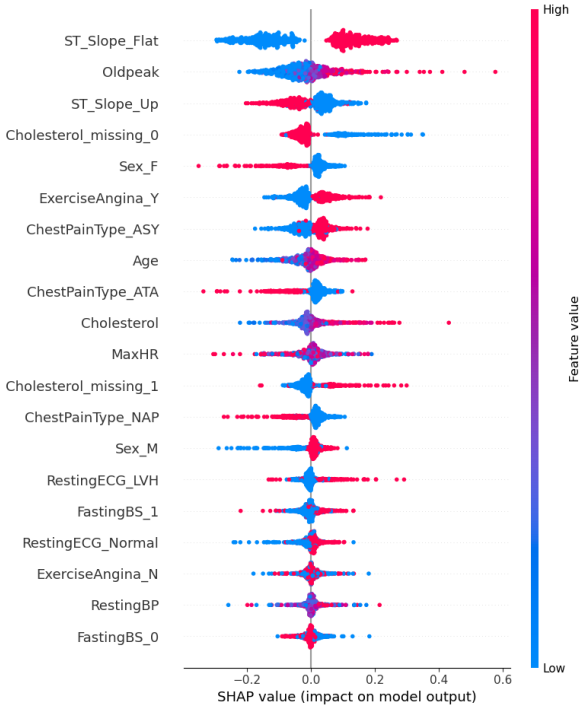


Figure 9: SHAP summary plot to visualize Shapley values and feature values for all instances

Metrics for Neural Additive Model	Score
F1 Score	0.8821
Balanced Accuracy Score	0.8375

1.4.2 Utilize the interpretability of NAMs to visualize the feature importances

For every feature, an individual neural network is trained. In the following, we visualize the learned relationship similarly to the paper and plot the numerical 11 and categorical 12 feature values against the predictions of the respective network. If the networks prediction does not vary over the inputs, the feature is considered unimportant. This is the case for "ChestPainType_TA" and is consistent with the finding that it's the least important feature based on Shapley values 8.
Furthermore we provide a measure of global feature importance and a corresponding plot 10 inspired by the Google Research Repository. The feature importance is measured as the mean absolute deviation (MAD) from the mean FeatureNN prediction over all the data points. The higher the deviation from the mean contribution, the more important the feature is considered in discriminating between the target classes.

1.4.3 Conceptually, how does the model compare to Logistic Regression and MLPs?

NAM vs. Logistic Regression A generalized additive model (GAM) is of the form $g(E[y]) = \beta + f_1(x_1) + f_2(x_2) + \cdots + f_k(x_k)$. Neural additive models (NAMs) are linear combinations of neural networks f_i that attend to single input features. Logistic regression is a generalized linear model with the logit function $g(p) = \log\left(\frac{p}{1-p}\right)$ as link. Compared to the NAM, it is a GAM with f_i restricted to be linear. NAMs and logistic regression are thus both GAMs and have separate weights for separate features. However, by parametrizing f_i with neural networks, NAMs can fit arbitrarily complex relationships between the feature and the target.

NAM vs. MLP NAMs use one MLP per feature and use separate weights for every of them. MLPs in contrast process all input features at once and share the weights among all input features if they are fully connected. In general, MLPs do not belong to the GAM class.

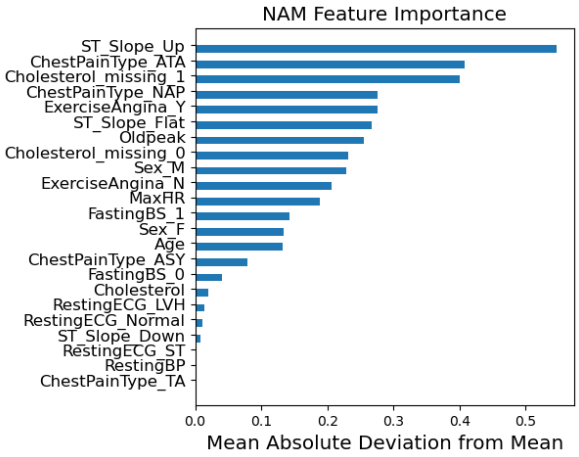


Figure 10: Global feature importance for the NAM model based on the mean absolute deviation from the mean of individual FeatureNN outputs

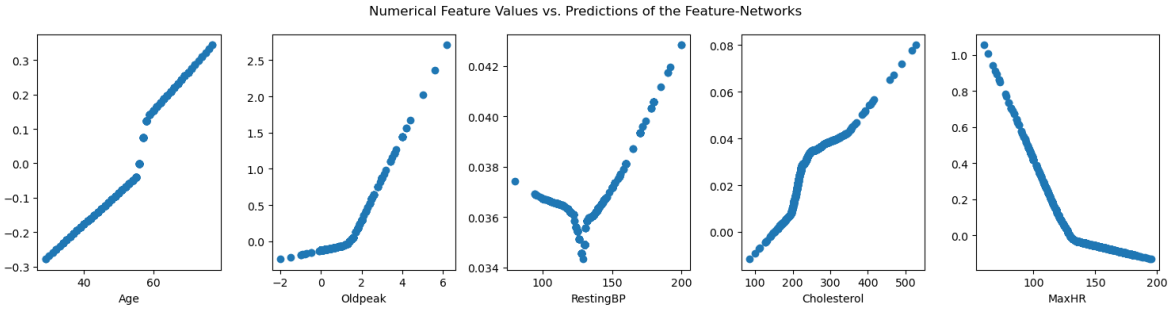


Figure 11: Numerical Feature Values vs. Predictions of the Feature-Networks

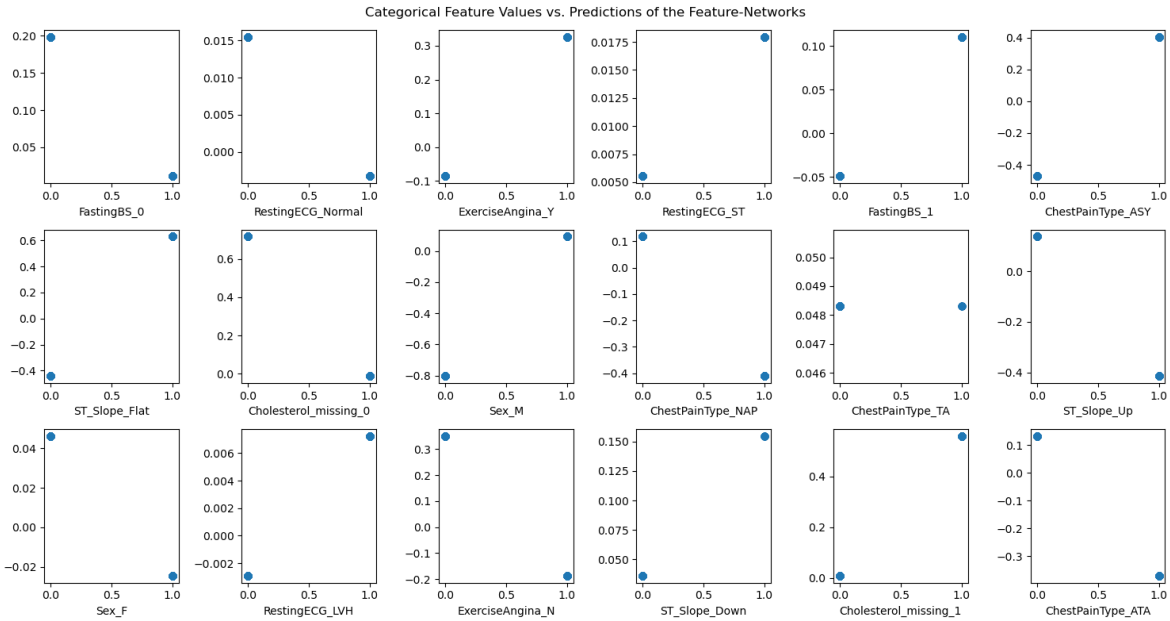
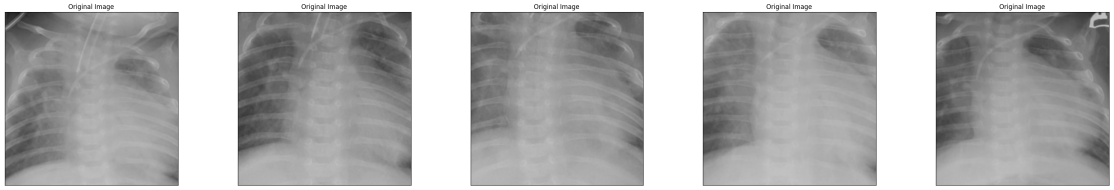


Figure 12: Categorical Feature Values vs. Predictions of the Feature-Networks

(a) Five healthy patients



(b) Five infected patients

Figure 13: X-ray scans of five healthy and infected patients

1.4.4 Why are NAMs more interpretable than MLPs despite being based on non-linear neural networks?

NAMs have different networks for each feature and thus one can plot $f_i(x_i)$ vs. x_i and inspect the relationship between input and output. This makes NAMs interpretable. MLPs process all input features at once and weights are shared among these. As features depend on each other, it is harder to explain the impact of one feature on the prediction.

2 Pneumonia Prediction Dataset

2.1 Exploratory Data Analysis

2.1.1 Explore the label distribution and qualitatively describe the data by plotting some examples for both labels.

In total, the dataset contains 5856 images of which 5232 belong to the training set and 624 to the test set. A binary classification task at hand, the labels are NORMAL and PNEUMONIA. Overall, we observe more PNEUMONIA samples as 74.2% of the training set and 62.5% of the test set make up infected patients.

2.1.2 Do you see visual differences between healthy and disease samples?

Figure 13 represents five healthy and five sick patients. At first sight, images labelled NORMAL are considerably better visible. It is straightforward to make out the person's heart at the center of the image. In addition, lungs are well inflated forming a contrast to the heart. Finally, the boundary between lungs and diaphragm is crisp and clear. Compared to that, the scan for infected patients is more obscure. Not only is the image very blurry, but also is the patient's left side of the heart covering most of the left lung wing. Furthermore, it is more difficult to make out the boundary between the lungs and the diaphragm.

2.1.3 Describe one potential source of bias that could influence model performance.

Two of the sick patients are under assisted breathing. Inspecting closely, one can spot the pipe of a ventilator running down the spine to the person's heart. That is, these people are already under medical treatment. One potential bias, therefore, may be that the model recognizes that and misidentifies this very fact as an indicator for pneumonia. A more immediate potential source of bias marks the position of the letter R on the image. All ten samples possess a label indicating the patient's perspective.

However, the font and location of the R in both groups varies marginally. More importantly, it appears to be consistent within the groups. Hence, it is of utmost importance to crop the images and focus only on the chest.

2.1.4 How do you preprocess the data for your further analysis?

To overcome the above biases, we have resized and cropped dataset to obtain 224x224 images. Doing so, we can later apply transfer learning using VGG-16. Moreover, we standardize the images by dividing the mean and subtracting the standard deviation. Last, we apply a Gaussian blur of size 3x3 to the training set.

2.2 CNN Classifier

2.2.1 Design a CNN classifier for the dataset.

To train our classifier, we relied on transfer learning. Importing the VGG-16 model trained on ImageNet, we retrained the weights for the medical x-Ray images. A binary classification task at hand, we also improved the classifier in that way. In particular, we have added two linear layers and reduced the output features of the final layer to 2. Figure 14 represents our layer structure.

2.2.2 Report its performance on a test set.

Using the test set provided on the cluster, we find that the model does very well when classifying PNEUMONIA samples. With respect to healthy patients, we observe a lower accuracy. Overall, we achieve a test accuracy of 88.62%.

2.3 Integrated Gradients

2.3.1 Implement the integrated gradients method and visualize attribution maps of five healthy and five disease test samples.

Implementing Integrated Gradients proposed by Sundararajan et al. (2017), we have utilized the `captum` (Kokhlikyan et al., 2020) package. For consistency, we have included the same ten images as above.

2.3.2 Do the maps highlight sensible regions?

Indeed, we observe sensible regions down the patients' spines. Confirming our intuition, Integrated Gradients identifies the crisp boundaries between lungs and diaphragm as an important region. This is especially well visible in the final attribution of Figure 15a. In a similar vein, the technique displays the outline of the heart indicating a stark contrast to the well-inflated lungs. Too, the abdomen marks an important region. With respect to color coding, we note that those regions are generally relevant for both positive and negative attributions. Inspecting the images in Figure 13a, we note that the lower body is characterized by clear white visibility of the spine. In comparison to that, the lower spine of pneumonia patients is substantially darker. Apart from that, Integrated Gradients on PNEUMONIA samples grants high attribution to the lungs. As previously mentioned, the lung of an infected patients is partly covered by the heart and does not mark as strong a contrast as it does for healthy patients. In plain terms, the lungs are not as dark. Integrated Gradients picks up on this and assigns attribution to that very fact.

2.3.3 Are attributions consistent across samples?

In general, these characteristics are consistent across samples.

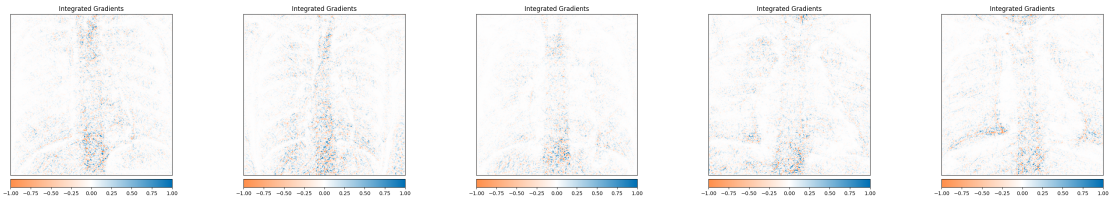
2.3.4 Does the choice of baseline input image have a big effect on the attribution maps?

To conclude, we will discuss the choice of baseline image. For our main analysis, we have implemented Integrated Gradients using a black image as the baseline. Alternatively, the authors also advocate using a noisy image. Out of curiosity, we also performed the task using a pink image as our baseline.

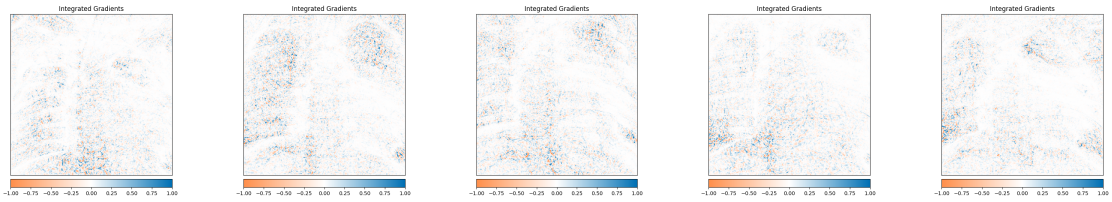
```
==========================================================================
Layer (type:depth-idx)                   Param #
==========================================================================
VGG                                      --
├─Sequential: 1-1                        --
│    └─Conv2d: 2-1                       1,792
│    └─ReLU: 2-2                         --
│    └─Conv2d: 2-3                       36,928
│    └─ReLU: 2-4                         --
│    └─MaxPool2d: 2-5                    --
│    └─Conv2d: 2-6                       73,856
│    └─ReLU: 2-7                         --
│    └─Conv2d: 2-8                       147,584
│    └─ReLU: 2-9                         --
│    └─MaxPool2d: 2-10                   --
│    └─Conv2d: 2-11                      295,168
│    └─ReLU: 2-12                        --
│    └─Conv2d: 2-13                      590,080
│    └─ReLU: 2-14                        --
│    └─Conv2d: 2-15                      590,080
│    └─ReLU: 2-16                        --
│    └─MaxPool2d: 2-17                   --
│    └─Conv2d: 2-18                      1,180,160
│    └─ReLU: 2-19                        --
│    └─Conv2d: 2-20                      2,359,808
│    └─ReLU: 2-21                        --
│    └─Conv2d: 2-22                      2,359,808
│    └─ReLU: 2-23                        --
│    └─MaxPool2d: 2-24                   --
│    └─Conv2d: 2-25                      2,359,808
│    └─ReLU: 2-26                        --
│    └─Conv2d: 2-27                      2,359,808
│    └─ReLU: 2-28                        --
│    └─Conv2d: 2-29                      2,359,808
│    └─ReLU: 2-30                        --
│    └─MaxPool2d: 2-31                   --
├─AdaptiveAvgPool2d: 1-2                 --
├─Sequential: 1-3                        --
│    └─Linear: 2-32                      102,764,544
│    └─ReLU: 2-33                        --
│    └─Dropout: 2-34                     --
│    └─Linear: 2-35                      16,781,312
│    └─ReLU: 2-36                        --
│    └─Dropout: 2-37                     --
│    └─Linear: 2-38                      16,781,312
│    └─ReLU: 2-39                        --
│    └─Dropout: 2-40                     --
│    └─Linear: 2-41                      16,781,312
│    └─ReLU: 2-42                        --
│    └─Dropout: 2-43                     --
│    └─Linear: 2-44                      8,194
==========================================================================
Total params: 167,831,362
Trainable params: 167,831,362
Non-trainable params: 0
==========================================================================
```

Figure 14: Conventional VGG-16 with appended classifier.

However, neither of the two approaches yield results comparable to the one for the black baseline. That is, the choice of baseline image did have a big effect with the black image performing best. The interested reader shall be referred to Figure 16.

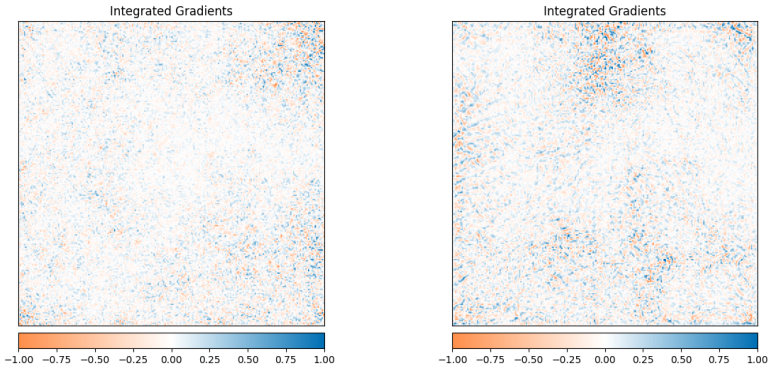


(a) Integrated gradients for healthy people

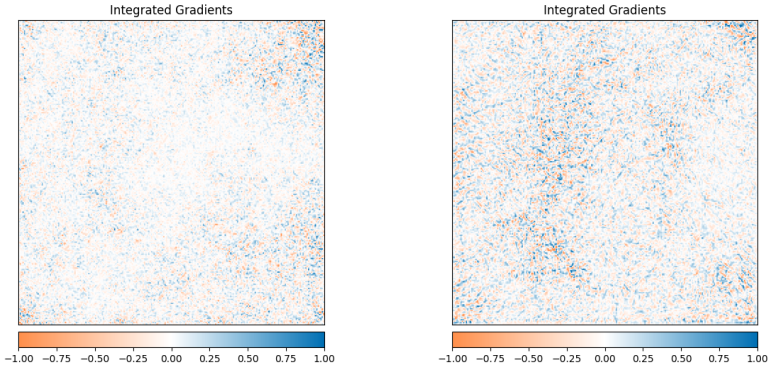


(b) Integrated gradients for infected patients

Figure 15: Integrated gradients heatmaps for five healthy and five sick people.



(a) Integrated gradients for a healthy person using a noisy and pink image baseline



(b) Integrated gradients for an infected person using a noisy and pink image baseline

Figure 16: Integrated gradients heatmaps for one healthy and one sick person using noisy and pink image baselines.

2.4 Q4: Grad-CAM

2.4.1 Implement the method and visualize attribution maps of five healthy and five disease test samples?

Grad-CAM (Selvaraju et al., 2019) offers visual explanations of the decision made by a model based on the gradients of the last convolutional layer. Our implementation follows Ulyanin (2019) and Gildenblat and contributors (2021) as we based our project on Pytorch, whereas the original paper used Tensorflow. We visualised the saliency maps using Grad-CAM for five healthy and five sick patients. We used images from the test set to do the visualisation in order to compare the results in this section with those of Q5, where the data randomisation test is based on test samples. Figure 17 depicts our results. Our model correctly predicted all of the ten cases. In order to focus on the areas with the highest activations seen in the heatmap, we threshold the heat map produced by Grad-CAM for all images equally. Outputs without this threshold can be found in the appendix. It can clearly be seen that the activation area is a bit stronger if the model predicts that the patient has pneumonia.

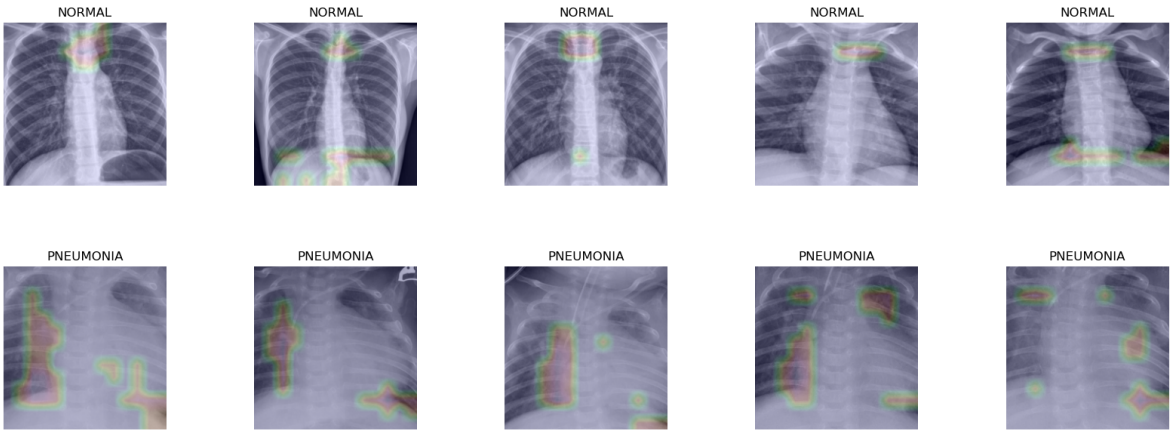


Figure 17: Grad-CAM using our CNN

2.4.2 Do the maps highlight sensible regions?

Activations for positive predictions are mainly seen in the lung wings, where one might expect signs of pneumonia. In contrast, the activations are seen further up the windpipe in the images where the model predicted no pneumonia. This is also present in the pictures without threshold (Figure 22).

2.4.3 Are attributions consistent across samples?

We also see a consistent trend across the same predicted class. For normal cases, the activations are more concentrated on the upper parts of the windpipe or lower part of the chest area. For pneumonia cases, the activations are in the lung wings.

2.4.4 Compare your findings with Q3

We can compare the heat maps with the results of the integrated gradient method from Q3. Let us visualize the 10 cases using integrated gradients just using the heat map (figure 18). Similar to Grad-CAM, we see consistent patterns across the same predicted class. Also, integrated gradients highlight the spline/windpipe for normal cases. Contrary to Grad-CAM, it focuses on the whole wind-pipe. For pneumonia cases, the similarity between the two methods continues as highlighted regions to some extent coincide. For example, looking at the second and fourth pneumonia cases, we see astonishing similarities between the two methods of the left part of the lung and four the fourth picture in the upper right part.

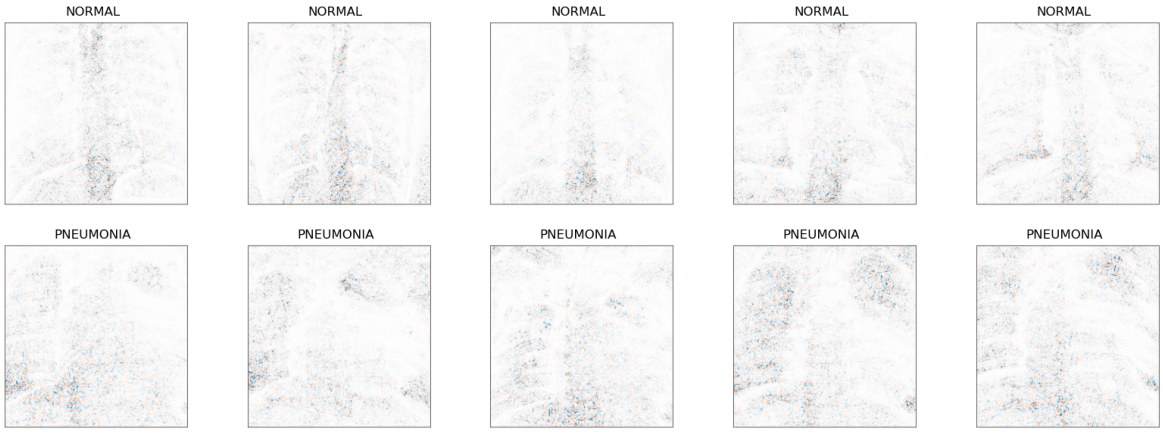


Figure 18: Heat map of integrated gradients

2.5 Q5: Data Randomization Test

Adebayo et al. (2020) introduce a data permutation test to evaluate if a method produces saliency maps that explain a relationship between the input and the label. This is achieved by permuting the labels of the training images. If saliency maps do not change between the model trained on the original data and the same model trained on the permuted data, one can not expect it to explain the underlying relationship. We retrained the whole CNN on the permuted data to a training accuracy of 87.8%, which is close to the test accuracy of our original model. Note that on the test set, the permuted model can not be expected to perform better than random guessing as it had to memorize each training sample with the permuted labels.

2.5.1 Grad-CAM

Figure 19 shows the Grad-CAM saliency maps for our CNN trained on the permuted labels. A figure without thresholding is again found in the appendix. We see a profound change in the results compared to the original Grad-CAM output in section 2.4. Across the photos, the highlighted regions vary considerably or are not strong enough to escape the threshold, although we used a smaller threshold here. There is no apparent difference between normal and pneumonia cases. This is also apparent when looking at the predictions made by the model based on permuted; 6 of the 10 cases were correctly classified. For the normal cases, only the first image was correctly classified. For pneumonia cases, all

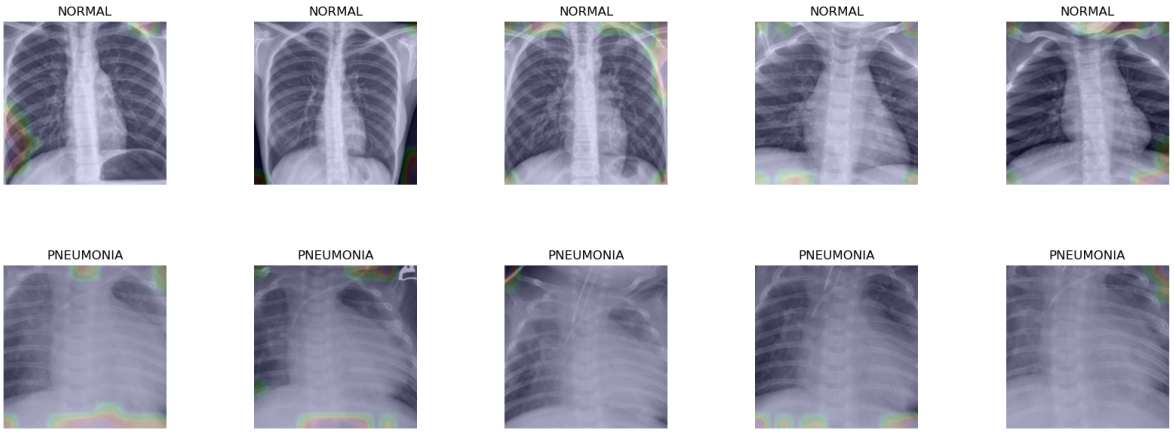


Figure 19: Grad-CAM fit on permuted dataset

images were classified as pneumonia. Additionally, we see the model focusing on unintuitive regions for cases classified as positive. For example, the third normal case was predicted as pneumonia, but the highlighted areas are at the edges, far away from the lung wings. Because the activations are weaker

in this output, this trend is more profound in the image without threshold (see figure 23), Grad-CAM seems to pass the data randomisation test based on the saliency maps, as its saliency maps considerably change when permuting the data set. This gives ground to argue that Grad-CAM has captured part of the data generation process and the relationship between the images and the labels. This coincides with the findings of Adebayo et al. (2020) where the Grad-CAM method also passed.

2.5.2 Integrated Gradients

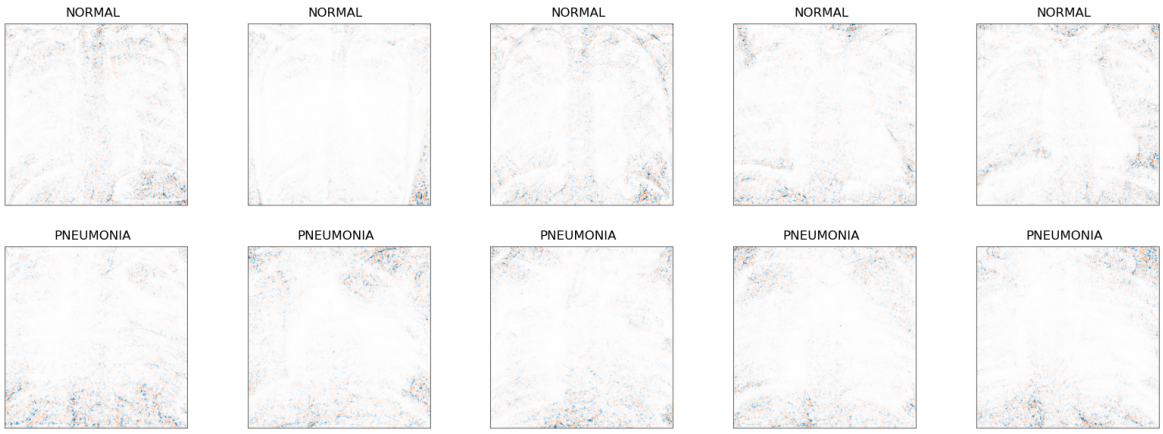


Figure 20: Heat map of Integrated Gradients based on permuted dataset

Similar to the previous subsection, we can see how well Integrated Gradients perform on our permuted model. Figure 20 shows the heatmaps produced by Integrated Gradients for the permuted model in the same fashion as figure 18. Again, we see profound changes in the heat maps produced. For normal cases, we lose the intense focus on the windpipe that was previously seen. For pneumonia cases, Integrated Gradients lose their focus on the lung wings and allocate more focus on the edges of the images. Interestingly, we see a similar pattern between Grad-CAM and Integrated Gradients again since the former method also shifted its attention to the borders of the image after permuting the data. It is to note that the fifth NORMAL image has the same characteristics as the scans analyzed in section 1.3. Again, we observe a crisp boundary between the heart and the lung wings. Thus, although Integrated Gradients passes the test, we find instances for which Integrated Gradients attributes the known features to the patient's scan. Nonetheless, the method overall produces severely different images after permuting the data. Thus, it also passes the data permutation test. This is opposed to the findings of Adebayo et al. (2020), where Integrated Gradients did not pass.

3 Part 3: General Questions

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

Part 1 The three studied importance measures, namely coefficients of logistic regression, mean absolute Shapley values and NAM feature importance have some overlap in discovering the most and least important features. For instance, "ST_Slope_Flat" is the second largest coefficient, the most important SHAP feature and among the most important NAM features. In addition, the least important SHAP feature, "ChestPainType_TA", also has the least important NAM importance and one of the smallest coefficients. However, it is not always that clear and the consistence is violated for several features, e.g. "ChestPainType_ASY", which has the largest coefficient, but only a medium Shapley importance and a low NAM importance. So overall there might be a tendency, but no consistency.

Part 2 Grad-Cam (Section 2.4) showed consistent patterns within the same predicted class. For pneumonia predictions, it highlighted features and patterns in the lung wings. Contrary to the pre-

dictions for NORMAL samples, the highlighted features were not present in the lung wings but in the windpipe and abdominal region. These patterns were seen across samples from the same predicted class. For Integrated Gradients, we observe comparable results as the technique attributes high importance to similar regions for the respective groups. In addition, Integrated Gradients has found common characteristics within groups. Comparable to Grad-Cam, Integrated Gradients focuses on the lung wings for infected patients and on the difference between diaphragm and lungs for healthy people.

3.2 Q2: Given the "interpretable" or "explainable" results of one of the models, how would you convince a doctor to trust them? Pick one example per part of the project.

Part 1: Lasso Logistic Regression For winning the trust of the doctor, we choose the interpretable logistic regression. First, we explain what was learned by the model: the relationship between features and the target class is expressed by the coefficients that can be intuitively interpreted on the odds-scale. The odds and relative importance of features is then discussed with the doctor to check plausibility. Second, we explain that new predictions on the odds-scale are made by the *exp* of a simple linear combination. Third, we ask the doctor for intuition on additional feature engineering like interactions between features or higher order polynomial transformations. In contrast to MLPs, every feature is explicitely modeled and transparent to the doctor in this way. The accuracy for logistic regression is furthermore better than for the MLP and it is the less complex model - according to Occam's razor the logistic regression is the model to pick.

Part 2: Grad-Cam In order to convince a doctor to trust the explanations of the heatmap Grad-Cam provided, we could look at a specific example. Looking at the second pneumonia case in Figure 17, we see which part of the image was relevant for the correct PNEUMONIA classification. In particular, two prominent regions highlighted could easily be communicated to a doctor. In practice, we could provide a doctor with the original image and ask her to diagnose the case by highlighting regions she deems important for her classification. Ideally, the heatmap constructed by humans coincides with the one produced by Grad-Cam. If not, we could still show the doctor the heatmap and ask if she agrees with the importance of the presented features. Nonetheless, the chance remains that neither strategy works to convince the doctor. In this case, it appears difficult to provide an explanation for the model's highlighted features. Equipped with expertise, the doctor only deems the model valuable if she can follow its "argumentation".

3.3 Q3: Elaborate whether the feature importances from the interpretability/explainability methods intuitively make sense to find the respective disease.

Part 1 Intuitively, we would consider "Age" one of the risk factors for heart diseases. Also, men tend to eat less healthy and thus could be more at risk. These hypotheses are confirmed by positive coefficients, positive Shapley values and the NAM plots. The less intuitive medical feature "ST_Slope" corresponds to the slope of ST-segment depression recorded by ECG during physical exercise. According to this website, flat ST-segment depressions can be a sign of disease. This is confirmed by positive coefficient, positive Shapley values and the NAM plot.

Part 2 Given our discussion in section 2.1, Integrated Gradients is intuitive. By simply observing the images in Figure 13, one notes differences in the clarity and crispness of the images. As mentioned, the infected patients' scans are considerably blurrier than those of healthy people. Overlaying Integrated Gradients highlights exactly this. Figure 21 displays one of the healthy people from above. Immediately, we see very clear boundaries between different organs. On top of that, Integrated Gradients notices the same and attributes its decision to this very matter. That is, we can easily spot how the conclusion is drawn here.
Grad-Cam highlights learned features that strongly impact the made prediction. Thus, the heatmap can be seen as an indicator of feature importance. The features depicted by Grad-Cam make sense for the pneumonia features. In fact, the method highlights regions in the lung wings which might

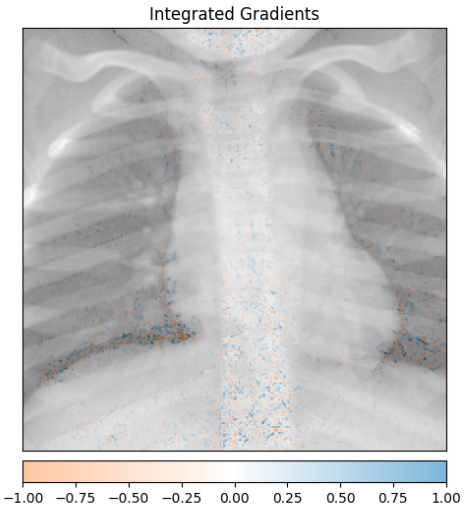


Figure 21: Integrated Gradients for a healthy patient

be impacted by pneumonia. Ideally, the model learns that the signs of pneumonia can be seen in the highlighted areas, which is why they substantially impact the class prediction. Whether or not the important features highlighted for the normal cases make sense is less clear. First of all, the discriminating features learned for normal cases should only correspond to the absence of abnormal signs. Thus, the features highlighted are harder to interpret. For example, the absence of inflammation in some regions might be a good predictor for a patient not having pneumonia, but highlighting those regions does not correspond with seeing something special. Generally, from our unfounded biological perspective, we find that Grad-Cam intuitively highlights regions that make sense and help explain how the model made a decision.

3.4 Q4: If you had to deploy one of the methods in practice, which one would you choose and why?

Part 1 If we have a small dataset and a focus on understanding the learned relationships, we would use simple, interpretable models like Logistic Regression and hand-craft the features. It also provides inference to test hypotheses. If more data is available and the focus is on prediction, we would deploy a more complex model like an MLP and using Shapley values for post-hoc explanation. However, we would still fit Logistic Regression as a benchmark.

Part 2 Considering the two saliency methods, we would deploy Grad-Cam in practice. First, it passes the data randomization test of section 2.5. That is, when deploying Grad-Cam for a variety of tasks, we can perform this test to check if we can trust our saliency maps. Second, the heatmaps produced by Grad-Cam are straight-up intuitive. Even the incognizable reader can interpret the heatmaps in Figure 17 without any prior machine learning knowledge. Third, Integrated Gradients shows many points of different colors that can be confusing, hard to interpret, and difficult to convey.

A Additional graphs and data

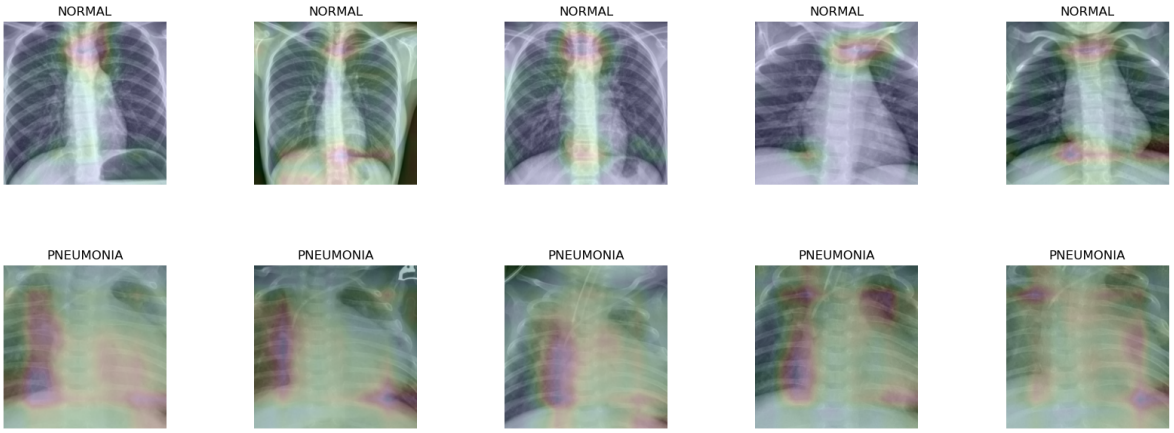


Figure 22: Grad-Cam using our CNN without threshold

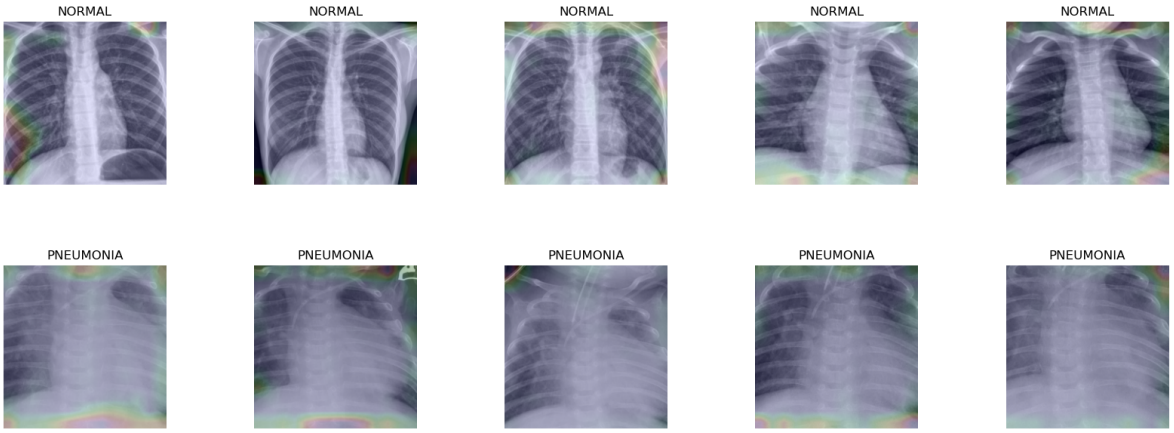


Figure 23: Grad-Cam fit on permuted dataset without threshold

References

Adebayo, J., Gilmer, J., Muelly, M., Goodfellow, I., Hardt, M., & Kim, B. (2020). Sanity checks for saliency maps.

Agarwal, R., Melnick, L., Frosst, N., Zhang, X., Lengerich, B., Caruana, R., & Hinton, G. (2021). Neural additive models: Interpretable machine learning with neural nets.

Bühlmann, P., & Van De Geer, S. (2011). *Statistics for high-dimensional data: Methods, theory and applications*. Springer Science & Business Media.

Gildenblat, J., & contributors. (2021). Pytorch library for cam methods.

Grundy, S. M. (1986). Cholesterol and Coronary Heart Disease: A New Era. *JAMA*, *256*(20), 2849–2858. https://doi.org/10.1001/jama.1986.03380200087027

Kokhlikyan, N., Miglani, V., Martin, M., Wang, E., Alsallakh, B., Reynolds, J., Melnikov, A., Kliushkina, N., Araya, C., Yan, S., & Reblitz-Richardson, O. (2020). Captum: A unified and generic model interpretability library for pytorch.

Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J., Passos, A., Cournapeau, D., Brucher, M., Perrot, M., & Duchesnay, E. (2011). Scikit-learn. *Journal of Machine Learning Research*, *12*, 2825–2830.

Selvaraju, R. R., Cogswell, M., Das, A., Vedantam, R., Parikh, D., & Batra, D. (2019). Grad-cam: Visual explanations from deep networks via gradient-based localization. *International Journal of Computer Vision*, *128*(2), 336–359. https://doi.org/10.1007/s11263-019-01228-7

Sundararajan, M., Taly, A., & Yan, Q. (2017). Axiomatic attribution for deep networks. *CoRR*, *abs/1703.01365*. http://arxiv.org/abs/1703.01365

Ulyanin, S. (2019, February). Implementing grad-cam in pytorch. https://medium.com/@stepanulyanin/implementing-grad-cam-in-pytorch-ea0937c31e82