Project 1: Interpretable and Explainable Classification for Medical Data

Project 1 will consist of two parts where you work with different modalities, tabular and imaging data. You will explore techniques that enable interpretable classification using simple and deep machine-learning methods. Each part is split up into smaller tasks. To get full points for a task, be aware of the following:

- For each question, we expect you to write a small paragraph where you document your implementation, elaborate on your findings, and provide proper visualizations or plots of your results. You are free to use whichever writing tool you like (word, latex, etc.), but make sure to hand in Project 1 as a pdf file on Moodle.
- Properly document and structure your code to make it reproducible. We recommend using Jupyter Notebooks for your code submission. In addition, please provide a requirements.txt or similar to install dependencies and a README.md explaining how to run your code.
- Make sure to use train/validation splits for training and tuning only. Report results on the test set. Further, clearly state if you report a metric on the train or validation set. Note that the performance of the different methods can vary a lot.
- If you encounter computational difficulties due to hardware constraints, feel free to subsample the dataset and preprocess/resize the data to reduce dimensionality. Such preprocessing will not impact the number of points you achieve. However, please clearly state if you apply some sort of preprocessing.
- Using publicly available code is okay, but properly reference repositories when you use them. Of course, you are not allowed to use the code of other teams from the course.
- You have to solve **at least one** of the two challenges to achieve the maximum grade.

Part 1: Heart Disease Prediction Dataset (30 Pts)

For Part 1, we will provide you with train and test splits from the Kaggle Heart Failure Prediction Dataset aggregated from UCI Machine Learning Repository over Moodle.

Q1: Exploratory Data Analysis (5 Pts)

Get familiar with the dataset by exploring the different features, their distribution, and the labels. Check for common pitfalls like missing or nonsensical data, unusual feature distribution, outliers, or class imbalance, and describe how to handle them. After having familiarized yourself with the data, explain how you preprocess the dataset for the remaining tasks of part 1. Interpretability and explainability aim at gaining more insights about the data than just optimizing predictive performance. A first simple step is to have a look at the (linear) dependencies of the variables. Visualize the pairwise correlation matrix and describe which features seem to be important for predicting the label.

Q2: Logistic Lasso Regression (5 Pts)

By design, linear models are interpretable due to the weights that intuitively provide feature importance values. Further, we can perform l_1 regularization to sparsify weights, allowing us to understand which features do not contribute to the outcome. For this question, fit a Lasso regression model with l_1 regularization on the dataset. What preprocessing step is crucial to ensure comparability of feature coefficients? Provide performance metrics such as f1-score or balanced accuracy to quantify the performance of this model.   Visualize the importance of the different features and present how they contribute to the model's output. Finally, argue for or against fitting a logistic regression using only the important variables, as determined by the Lasso model, to arrive at the final coefficients instead of keeping the coefficients of the Lasso model.

Q3: Decision Trees (3 Pts)

Like linear models, decision trees are intrinsically interpretable models by nature. For a given output, we can retrace every decision that led to the final prediction by following the path of a sample along the edges of the tree. Further, the models' impurity measure allows us to quantify feature importance within the dataset. As in the previous question, train a decision tree on the dataset and report classification performance on the test set. Visualize the influence of the different features according to the Gini importance.

Q4: Multi-Layer Perceptrons(7 Pts)

While often reaching superior performance, MLPs are generally hard to interpret, and it is not straightforward to see what is happening within these models. We thus opt for post-hoc explainability methods such as SHAP[1]. Post-hoc explainability methods typically use some procedure during inference to find the feature importance per sample. Similar to Q2 and Q3, implement a simple MLP, train it on the dataset, and report test set performance. Make sure to report the architecture and optimization procedure you used for training.
Further, visualize SHAP explanations of the outputs of four positive and negative samples and feature importances of the overall model. Are feature importances consistent across different predictions and compared to overall importance values? Elaborate on your findings!
Hint: There is an excellent SHAP library for python that provides many SHAP algorithms and visualizations out of the box.

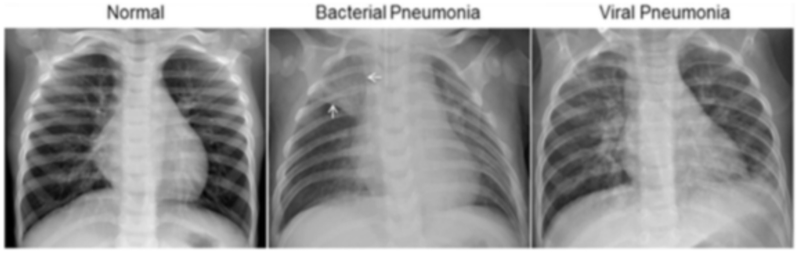
Challenge 1: Neural Additive Models[2] (10 Pts)

Another way to make deep models more interpretable is by careful design of the architecture. One example of such a model is the Neural Additive Model (NAM), which is an instance of the

[1] Lundberg and Lee, “A Unified Approach to Interpreting Model Predictions.”
[2] Agarwal et al., “Neural Additive Models.”

class of Generalized Additive Models[3] (GAM). Read the paper about NAMs, implement the model, and train it on the dataset. Like Q2-4, provide performance metrics on the test set. Similar to Q4, visualize the feature importances of four positive and four negative samples. Are the feature importances similar for different samples? Do the feature importances found with NAMs differ from the ones in Q2-4, or are they consistently the same? Conceptually, how does the model compare to Logistic Regression and MLPs? Why are NAMs more interpretable than MLPs despite being based on non-linear neural networks?

Part 2: Pneumonia Prediction Dataset (30 Pts)



For Part 2, download the Kaggle Dataset Chest X-Ray Images (Pneumonia)[4].

Q1: Exploratory Data Analysis (4 Pts)

Download and explore the data. Explore label distribution and qualitatively describe the data by plotting healthy and pneumonia samples. Do you see visual differences between healthy and disease samples? Do you find sources of bias that could influence model performance? How do you preprocess the data for your further analysis?

Q2: CNN Classifier (4 Pts)

In Q3 and Q4, we aim to use post-hoc explainability methods for visualizing the parts of the image that are important for the prediction of a model. Thus, design a small CNN classifier for the dataset and report its performance on a test set. Make sure to elaborate on your architecture and training details.

Q3: Integrated Gradients[5] (4 Pts)

Like MLPs, CNNs perform very well in tasks like classification but lack interpretability due to their black-box nature. Again, post-hoc explainability methods are thus suitable alternatives. One class of post-hoc procedures specific to image data are methods generating attribution maps, which try to highlight the most important regions on which the CNN bases its predictions. For this part of the assignment, implement the integrated gradient method. Visualize attribution

[3] Hastie, “Generalized Additive Models.”
[4] Kermany et al., “Identifying Medical Diagnoses and Treatable Diseases by Image-Based Deep Learning.”
[5] Sundararajan, Taly, and Yan, “Axiomatic Attribution for Deep Networks.”

maps of five healthy and five disease test samples. Do the maps highlight sensible regions? Are attributions consistent across samples? Do they match potential observations from Q1?

Q4: Grad-CAM[6] (5 Pts)

Grad-CAM is another post-hoc method that generates attribution maps. Like in Q3, implement the method and visualize attribution maps of five healthy and five disease test samples. Do the maps highlight sensible regions? Are attributions consistent across samples? Compare your findings with Q3.

Q5: Data Randomization Test[7] (3 Pts)

Recently, the paper “Sanity Checks for Saliency Maps.” introduced the data randomization test to check how trustworthiness of the saliency maps of specific methods. They propose to retrain the classifier on the train set when randomly permuting labels of all samples. Then, they compare the saliency maps on test samples for the perturbed and unperturbed classifiers. We expect the map to change if an attribution map accurately captures the relationship between instances and their labels. Conversely, if the attribution map captures another concept, e.g., acts like an edge detector independent of the label, we expect the maps to stay the same. Retrain your CNN on random training labels and perform the Data randomization Test for both Integrated Gradients and Grad-CAM. Do they pass or fail? Elaborate and visualize your findings!

Challenge 2: Prototype Learning (10 Pts)

A radically different idea for interpretable classification is finding prototypical samples for each class. Then, in addition to the prediction, we can return the prototype most similar to the input. This idea was explored in the paper “Examples are not Enough, Learn to Criticize! Criticism for Interpretability”[8]. They provide a method that allows you to extract representative prototypes from a given dataset by using the maximum mean discrepancy (MMD) measure as a distance function and introducing a kNN-like classifier. After reading the paper, we ask you to implement their method in two steps:

1. Implement the “Nearest Prototype Classifier” described in Section 5 of the paper. For now, set the set of prototypes S to be random points of the training set. Report the classification performance of this model on the test set.
2. Implement the function $J_b(S)$ (Section 3) and select prototypes S through their greedy algorithm (Algorithm 1). Refit the classifier, this time with the selected prototypes, and compare classification performance to random prototypes. Does the result behave as you expected? Why or why not? Visualize five healthy and five disease prototypes. Do you find representative class patterns among them? Would you say they look prototypical?

[6] Selvaraju et al., “Grad-Cam.”
[7] Adebayo et al., “Sanity Checks for Saliency Maps.”
[8] Kim, Khanna, and Koyejo, “Examples Are Not Enough, Learn to Criticize! Criticism for Interpretability.”

How could you improve performance beyond simply applying kNN to the raw images? Compare this type of interpretable method to the previously seen saliency maps. Which method do you think is more useful? Can you think of scenarios where one is more valuable than the other?

Remark: The proposed method can be computationally inefficient. Keep that in mind and implement the method as efficiently as possible. Also, feel free to subsample the training set and resize images to, e.g., 128x128. Additionally, you may find it helpful to transform the RGB images to grayscale by setting pixel values to 0.299*red+0.587*green+0.114*blue.

Optional Task: The authors further propose to return a set of so-called criticisms: samples that belong to a class but have unique characteristics, making them less representative of the dataset. If you are interested, you can additionally implement and explore criticism and see whether you find them a valuable interpretable component. Note that this is not required to get full points for this task.

Part 3: General Questions (10 Pts)

To conclude, we ask you to answer the following questions to recap and reason about the project. Please answer each question for Part 1 and Part 2 separately.

Q1: How consistent were the different interpretable/explainable methods? Did they find similar patterns? (2 Pts)

Q2: Given the “interpretable” or “explainable” results of one of the models, how would you explain and present them to a broad audience? Pick one example per part of the project. (2 Pts)

Q3: Did you encounter a tradeoff between accuracy and interpretability/explainability? (2 Pts)

Q4: Do your findings from the interpretability/explainability methods align with the current medical knowledge about these diseases? You may take inspiration from the references of the project presentation. (2 Pts)

Q5: If you had to deploy one of the methods in practice, which one would you choose and why? (2 Pts)