Project Description

IMPORTANT LINKS:

- **App Functionality Screencast:** https://drive.google.com/file/d/1CEZrLO-owY44TG012B5l7L0jh8VlbTEI/view?usp=sharing
- **Deployed Streamlit App:** https://heart-disease-prediction-final.streamlit.app

Goals

We want to create an interactive application which allows our target audience achieve the following goals:

1. Explore the data on surface level (i.e. see the dataset and feature distributions)
2. Assess performance of different classification models
3. Generate and understand explanations for white-box and black-box models
4. Compare different models in performance and juxtapose their explanation techniques
5. Evaluate biases in data and in trained classifiers

Target Audience

Our primary target audience is fellow data scientists working in the field of medical research. We therefore expect a basic knowledge of statistics (e.g. knowing what a histogram is) and a surface-level understanding of machine learning. Knowledge of medical concepts is welcome, but not required; some more complex terms can be explored via the provided links.

Dataset Description

The Heart Disease dataset was created by a collaboration of medical professionals from multiple institutions: Andras Janosi, M.D., from the Hungarian Institute of Cardiology in Budapest, William Steinbrunn, M.D., from the University Hospital in Zurich, Switzerland, Matthias Pfisterer, M.D., from the University Hospital in Basel, Switzerland, and Robert Detrano, M.D., Ph.D., from the V.A. Medical Center in Long Beach and the Cleveland Clinic Foundation. The dataset was created on July 22, 1988, to primarily predict the presence or absence of heart disease in patients.

There are 14 attributes, including age, sex, chest pain type, resting blood pressure, serum cholesterol, fasting blood sugar, resting electrocardiographic results, maximum heart rate achieved, exercise-induced angina, ST depression induced by exercise relative to rest, the slope of the peak exercise ST segment, number of major vessels

colored by fluoroscopy, and Thallium defect, with the last attribute being the diagnosis of heart disease.

In total, 920 patients' records are sampled, comprising 303 from Cleveland, 294 from Hungary, 123 from Switzerland, and 200 from Long Beach VA. The sampling makes sense as it helps to mitigate any geography-based bias. It is a good idea to take into consideration data from various locations.

Some data points, such as names and social security numbers, were removed for privacy and replaced with dummy data. The original dataset retains 76 attributes, with published experiments typically using a subset of 14, which the preprocessed files contain, instead of all the 76 attributes. Some columns have missing values from specific sub-records, and the meaning behind the "thal" attribute requires a reference to the original paper for clarity. The dataset includes metadata in a .names file and has protected attributes like sex and age, as they are significant factors in heart disease but also require protection to prevent discrimination. Despite these considerations, the unprotected attributes can be used for analysis related to heart disease prediction.

Design

The app consists of a header, a multi-purpose sidebar, and 4 tabs: *Exploring Data*, *Building Models*, *Explaining Algorithms*, and *Evaluating Fairness*. Such a partition is conducive to a more user-friendly interface and eases the app navigation overall. We will now discuss each of the app's elements in more detail.

- ***Header***
  The header holds an introduction to the app. It also has a link to the original dataset page for user convenience.

- ***Sidebar***
  The sidebar serves several purposes. First of all, it has several filters that the user can customize to affect certain elements of the app (these will be discussed later in the text). There are filtering options for *age*, *sex,* and *diagnosis*. The sidebar also displays the total amount of dataset entries that correspond to the filters specified, as well as their fraction w.r.t. the original dataset size.

  Secondly, the sidebar has a feature explorer, which can promptly give information about a specific feature to the user. Each feature is given a short description and a list of possible values.

Finally, the user can also find the dataset description in the sidebar. This section provides some general information about the dataset creation and, additionally, a reference to the paper that originally introduced the dataset.

- ***"Exploring Data" tab***
  The first tab of the app provides an outlook on the data, as well as some basic statistical information.

  First of all, the tab allows the user to explore the entire dataset in tabular form. One table displays the original dataset, while the other displays the filtered dataset. If the filters are arranged in a way such that no entries are fitting, a special message is displayed.

  The second part of the tab is the plots of individual and pairwise distributions of data features. Individual distributions display bar plots for categorical features and histograms for continuous features. Pairwise distributions have 4 display options depending on the selected pair of feature names. If the feature names coincide, an individual distribution is shown. If both features are continuous, a scatterplot of the data points is displayed. If one feature is continuous and the other categorical, boxplots of the continuous variables w.r.t. their categories are presented. Finally, if both features are categorical, a heatmap of intersected data entries is displayed.

- ***"Building Models" tab***
  In this section, we explore the performance of different classification models in predicting whether a patient is likely to have heart disease. Our dataset originally contains 5 classes (0 to 4), where 0 represents a healthy class and 1-4 represents varying intensities of heart disease. For simplicity, we have combined classes 1-4 into a single class, making this a binary classification task: class 0 for healthy individuals and class 1 for diseased individuals.

  We trained our dataset using two models: Random Forest and Logistic Regression. Based on the user's selection from the dropdown menu, the results for the chosen model are displayed as a table, an AUROC curve, and a confusion matrix heatmap.

  The Logistic Regression model outperformed the Random Forest model in terms of overall accuracy and AUROC score. Specifically, the Logistic Regression model achieved an accuracy of 85% and an AUROC score of 0.91, compared to the Random Forest model's accuracy of 83% and AUROC score of 0.89.

Logistic Regression demonstrated higher precision (0.88 vs. 0.84) and recall (0.87 vs. 0.78) for the diseased class compared to the Random Forest model. For the healthy class, Logistic Regression also showed superior recall (0.90 vs. 0.87) but slightly lower precision (0.82 vs. 0.82) compared to the Random Forest model. The models show good precision and recall for both healthy (class 0) and diseased (class 1) individuals, indicating that they can distinguish well between patients with and without heart disease.

The F1 scores for both classes in both models are high, reflecting a balanced performance in terms of precision and recall. This balance is crucial for medical predictions to avoid both false positives (misclassifying healthy individuals as diseased) and false negatives (failing to identify diseased individuals).

Overall, while both models performed well, Logistic Regression showed a slight edge in accuracy, AUROC score, and balanced performance across precision, recall, and F1 score, making it a more reliable choice for predicting heart disease in this dataset.

- ***"Explaining Algorithms" tab***

  The "Explaining Algorithms" tab aims to provide an understanding of how different models make predictions. This involves analyzing the impact of each feature on the model's decision-making process, using techniques like SHAP (SHapley Additive exPlanations) values. In this tab, users can select between the Random Forest and Logistic Regression models to explore their inner workings. The explanations are crucial for gaining insights into model behavior, ensuring transparency, and identifying potential biases.

  Random Forest Model Explanation

  For the Random Forest model, users can see the importance of each feature based on SHAP values. The SHAP values represent the contribution of each feature to the prediction. A higher SHAP value indicates a more significant impact on the model's output. The visualization includes a bar plot showing the mean absolute SHAP values for each feature, sorted in descending order of importance. This helps in understanding which features are most influential in predicting heart disease. Based on the plot we notice that the thal (thallium defect) variable and the ca (number of major vessels) variable have the highest SHAP values, so they have the most effect on the predictions, while fbs (fasting blood sugar) has the least effect.

Logistic Regression Model Explanation

For the Logistic Regression model, the tab displays the coefficients of each feature, indicating their weight in the decision-making process. Positive coefficients suggest that higher values of the feature increase the likelihood of predicting the positive class (heart disease), while negative coefficients indicate the opposite. The features are sorted by the absolute value of their coefficients, emphasizing the most impactful features. Additionally, users can view the formula and mathematical representation of the Logistic Regression model. Based on the plot the sex has the highest importance, while serum cholesterol level has the lowest importance.

- ***"Evaluating Fairness" tab***

  The prediction results from the Logistic Regression and Random Forest models were tested for group fairness, group statistical fairness, predictive parity and false positive rate error balance metrics for both balanced and unbalanced data with respect to sex feature. Balancing of the dataset was performed by oversampling the minority class which in our case was female and then retraining was performed for both Logistic Regression and Random Forest models.

  Logistic Regression Model Fairness

  For the Logistic Regression model, group fairness did not exist before and after the training on balanced and unbalanced data. Conditional statistical fairness also did not exist and it worsened after balancing of the dataset and retraining of the model. Predictive parity and false positive error rate balance was very good for females compared to males before balancing the dataset and retraining and it worsened a bit for both sexes later on the retrained model on the balanced dataset.

  Random Forest Model Fairness

  For the Random Forest model, group fairness did not exist before and after the training on balanced and unbalanced data. Conditional statistical fairness existed almost perfectly before balancing and retraining of the model but deteriorated slightly after retraining. Predictive parity and false positive error rate balance was quite good for both sexes and it improved slightly after balancing and retraining of the model.

Issues

1. Scalability concerns: The current implementation might face challenges with larger datasets, particularly in terms of computation time for SHAP values and interactive visualizations.

2. Limited feature engineering: The project uses raw features without extensive feature engineering, potentially missing out on valuable derived features.

3. Data availability constraints: In real-world scenarios, obtaining comprehensive, unbiased datasets for healthcare can be challenging due to privacy concerns and data collection limitations

Reflections

1. Emphasis on responsible AI: By incorporating fairness metrics and discussions, the project promotes awareness of ethical considerations in AI development.

2. Educational value: The application serves as an excellent educational tool, allowing users to explore data, understand model performance, and grasp concepts of explainable AI and fairness.

3. Transparency: The project emphasizes the importance of model explainability, using techniques like SHAP values and coefficient analysis to provide insights into model decisions.

4. Proactive bias mitigation: The project doesn't just identify bias but takes active steps to try to mitigate it through data balancing techniques, showcasing a commitment to improving model fairness