Heart Disease Prediction with Tabular Data

Xinrui Wang | https://github.com/xinrui-wang1/heart-diseases

Introduction

Heart disease is the number one cause of death in the U.S. According to the CDC, 1 in 5 deaths in the U.S. was caused by heart disease [1]. Therefore, there is an urgent need to develop early diagnosis tools so that physicians can target high-risk patients with timely treatments. In this project we aim to explore the correlation between coronary disease and health indicators. We will train machine learning classification models to predict whether a patient has heart disease.

We use the 2020 annual CDC survey data collected by the CDC's Behavioral Risk Factor Surveillance System (BRFSS). The data includes more than 290 health status features of more than 400,000 U.S. adults. The data is preprocessed by Kaggle user, Kamil Pytlak, who dropped entries with null values and conducted feature selection that reduced the feature counts to 18 [2]. The result dataset consists of 319,795 entries, 17 input features, and one target variable.

Our target variable is “HeartDisease", which takes binary values “Yes" and “No.” For the input features, we examine the effects of patient’s health factors (smoking and drinking habits, sleep and exercise patterns, etc.), demographic features (sex, race, age groups), as well as the underlying comorbidities (asthma, strokes, etc.). We also consider patients’ self-evaluation of their physical and mental health.

Exploratory Data Analysis

	BMI	PhysicalHealth	MentalHealth	SleepTime
count	319795	319795	319795	319795
mean	28.33	3.37	3.90	7.10
std	6.36	7.95	7.96	1.44
min	12.02	0	0	1
max	94.85	30	30	24

Table 1. Key statistics of numeric features

BMI ranges from 12.02 to 94.85 with a mean around 28.33. Both PhysicalHealth and MentalHealth features are self-reported and range from 0 to 30. SleepTime ranges from 1 to 24 with a mean around 7 hours.

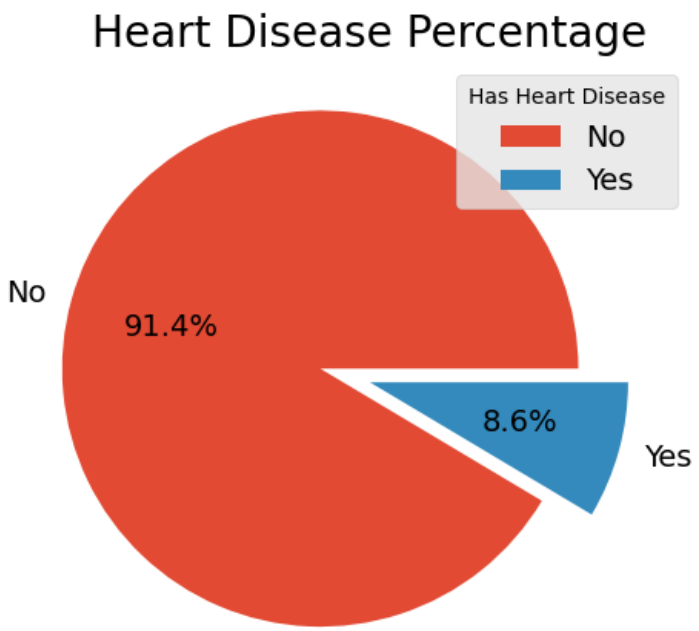

Figure 1. Percentage of samples with heart disease

Figure 1 shows the percentage of population samples that have heart disease. Around 8.6% of the population (27,502) have heart disease while 91.4% (292,292) do not have heart disease.

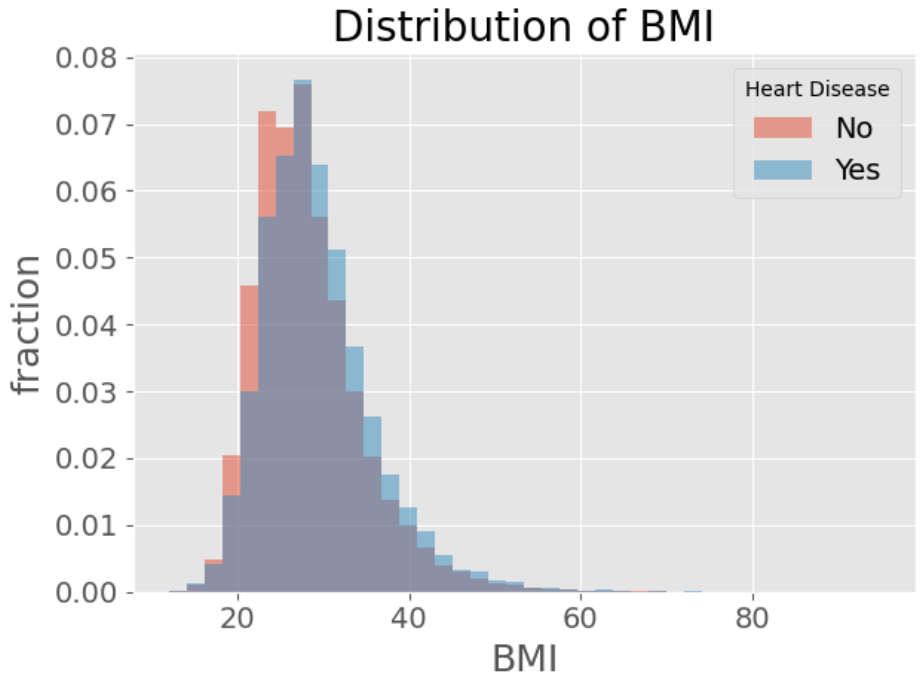

Figure 2. Distribution of BMI by heart disease

Figure 2 shows the distribution of BMI by target variable. The blue histogram shows the distribution of BMI for people with heart disease, and the red histogram shows the distribution of BMI of those without heart disease. The two histograms follow similar distribution, but the blue histogram has slightly higher mean and median.

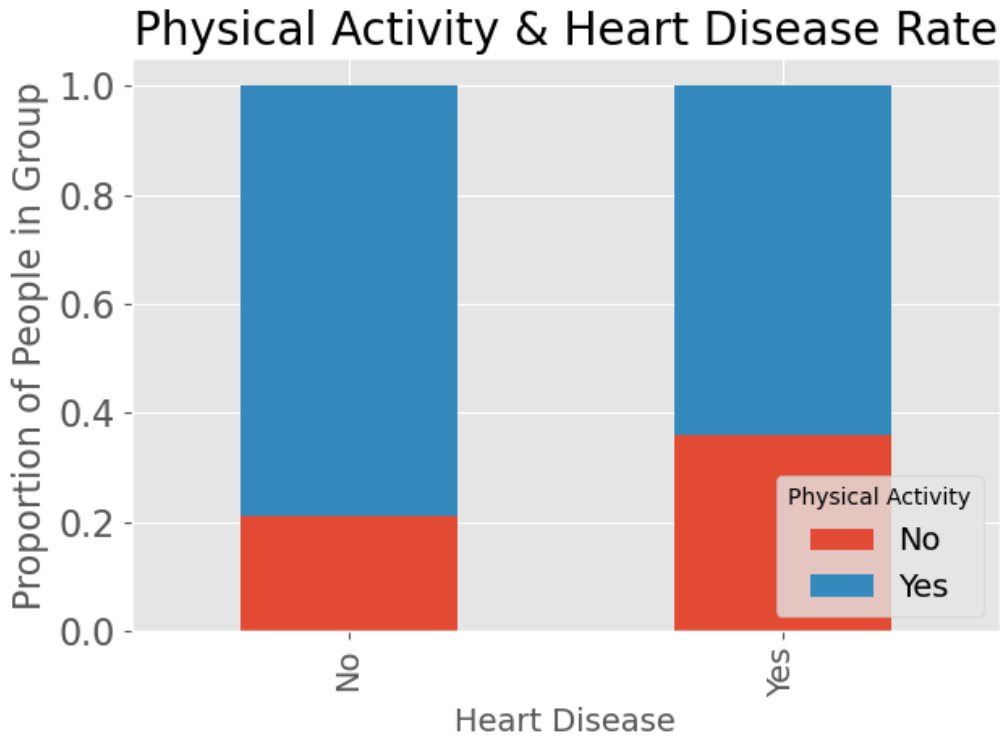

Figure 3. Physical activity and heart disease rate

Figure 3 divides the sample population into two cohorts - those with heart disease and those without. Within each cohort, it shows the fraction of people who identify themselves as someone who exercises regularly. Around 21% of the patients who have heart disease indicated that they exercise regularly, compared to 36% for those without heart disease.

Methods

Split

Each row represents a unique survey participant who is independent from other participants; furthermore, there is no group structure nor time series elements in the dataset. Therefore, we treat our data as Independent and Identically Distributed (I.I.D.) and use a train, validation, test split with ratio 0.6, 0.2, and 0.2.

Data Preprocessing

For preprocessing, we group our features into four categories. We have one ordinal feature - "GenHealth", which is the survey participants' self-reported health status. We use the ordinal encoder to encode this feature (0 for “Poor”, 1 for “Fair”, etc.). For categorical features, such as "Sex", "AgeCategory", and "Race", we use the one hot encoder. For numerical features, such as "BMI", we apply a standard scaler to normalize the values. The dataset has 17 input features before transformation and 30 features after transformation.

Model Pipeline

We utilize four classification models - logistic regression, random forest, k-nearest neighbors, and XGBoost. For each model, we search through hyperparameters using the training and the validation sets to find the optimal combinations. We then iterate through ten random states and compute the mean and standard deviation of the test scores using the training and the testing sets. We evaluate our models based on our chosen metrics and the confusion matrices and lastly conduct feature importance analysis using the best-performing model.

Hyperparameter Tuning

For each model, we tune a series of different hyperparameters. Since our dataset is imbalanced with 91.4% of the data belonging to class 0, we balance the class weights for each of the models except for the KNN classifier.

For logistic regression, we examine different inverse regularization constant C, different penalties (no penalty, l1, l2, and elastic net). For the elastic net penalty, we choose among different l1 ratios, which specifies the amount of weights we want to set for l1 and l2 penalties.

For the random forest classifier, we search through values of the trees’ maximum depth and the number of estimators in the random forest.

For the k-nearest neighbors classifier, we examine different values of k - the number of neighbors, and weights (uniform weight and distance weight). The uniform weight function weighs all neighboring data equally while the distance weight function weighs the neighboring data points by their inverse distance from the query point.

For the XGBoost classifier, we consider its learning rate - eta, the maximum depth, and the number of estimators. To balance class weights, we tune the scale for the weights between the positive and the negative classes.

Evaluation Metrics

91.4% of the participants in our data do not have heart disease, leading to data imbalance. Our objective is to build a predicative system that can identify potential heart disease patients. In addition, our system will be used alongside physician oversight, which will likely capture the false positives in our prediction. In this case, the harm of missing a heart disease patient is significantly greater than the harm of mislabeling a healthy patient.

Therefore, our objective becomes minimizing the number of false negatives. At the same time, the nature of our prediction task forbids us from predicting every patient as class 1, as this destroys any significance of our predictions. As a result, we want to set an evaluation metric that

takes into account both the false positives and the false negatives but puts more emphasis on the false negatives. We choose the f-beta score to be the evaluation metrics, with β greater than 1. The f-beta score is defined as the following:

$$f_\beta = (1 + \beta^2) \cdot \left(\frac{precision \cdot recall}{\beta^2 \cdot precision + recall}\right) \quad (1), \text{ where}$$

$$precision = \frac{TP}{TP+FP} \quad (2)$$

$$recall = \frac{TP}{TP + FN} \quad (3)$$

Precision measures the fraction of true positives among predicted positives. Equivalently, optimizing the precision minimizes the false positives in the prediction. Recall measures the fraction of true positives among conditioned positives. Optimizing recall minimizes the false negatives in our prediction. The parameter β weighs between precision and recall. When β is greater than 1, we put more weights on recall than precision. Since we our model should prioritize minimizing false negatives, we choose β equals to 2.

Results

Baseline Score

After choosing the evaluation metrics and our β, we calculate the baseline score. Normally, we would set all predictions to the majority class (class 0); however, this is infeasible in this case since precision and recall both equal to 0, and the f-beta score would be in turn undefined. Alternatively, we set all of our predictions to class 1 to get the following results:

$precision = \frac{TP}{TP+FP} = frac_{n_1} = 0.086$, where $frac_{n_1}$ is the fraction of class 1

$recall = \frac{TP}{TP + FN} = 1$, and

$$f_\beta = (1 + \beta^2) \cdot \left(\frac{precision \cdot recall}{\beta^2 \cdot precision + recall}\right) = (1 + 2^2) \cdot \left(\frac{0.086 \cdot 1}{2^2 \cdot 0.086 + 1}\right) = 0.32$$

Therefore, our baseline f_2 equals 0.32.

Model Comparison

models	Best parameters	F_2 mean	F_2 Std.	Accuracy mean
Logistic regression	{'C': 1, 'l1_ratio': 0.2, 'penalty': 'elasticnet"}	0.5089	0.0044	0.7319
Random forest	{'max_depth': 10, 'n_estimators': 100}	0.506	0.0048	0.7321
KNN	{'n_neighbors': 3, 'weights': 'distance'}	0.1766	0.0037	**0.8864**
XGBoost	{'eta': 0.1, 'max_depth': 5, 'n_estimators': 100, 'scale_pos_weight': 10}	**0.511**	0.0043	0.7323

Table 2. Model parameters and test scores

The above table shows the best hyperparameters for each model as well as the test scores' mean and standard deviation. We also examine the test accuracy. The model with the best average test score is the XGBoost classifier with learning rate equals 0.1, max depth equals 5, number of estimators equals 100, and the weight scale for positive class equals 10. The resulting average test score is 0.511 and the standard deviation equals 0.0043.

We notice that the 3-nearest neighbors model with distance weights has the highest average test accuracy. However, the model produces poor f_2 scores. This is because we cannot balance the class weights in KNN classifiers. As a result, the model predicts almost all of the labels as class 0.

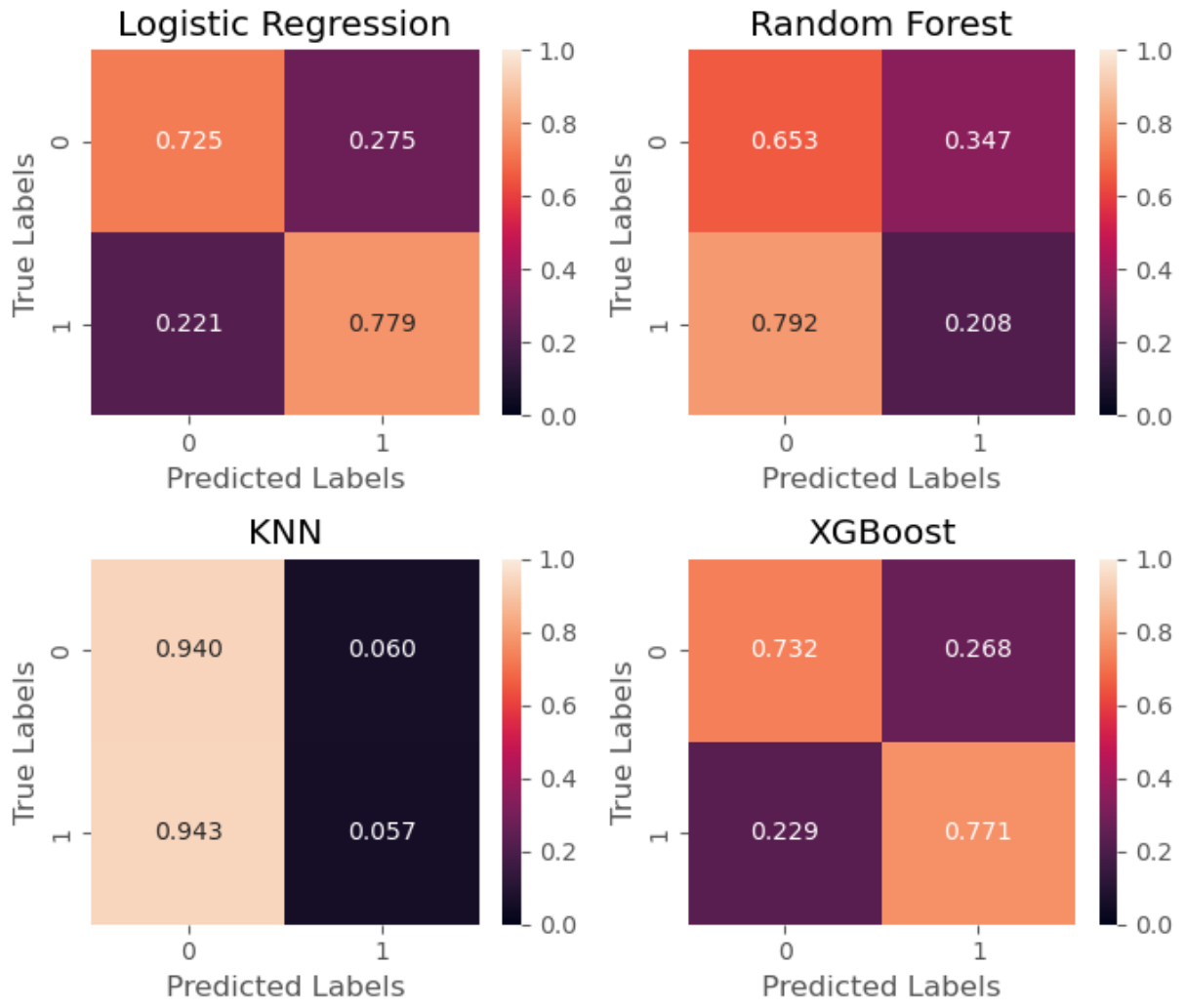

Figure 4. Confusion matrices

Figure 4 shows each model's confusion matrix. As previously stated, the KNN classifier produces high false negatives due to predicting most of the labels as class 0. We see similar results in the confusion matrix, as the KNN classifier shows a high false negative rate. Logistic regression and XGBoost share similar performances, and both models outperforms the random forest classifier.

Overall, the XGBoost predictor produces the lowest average test f_2. We will utilize this particular model for feature importance analysis.

Feature Importance Analysis

We examine three global feature importances, including the permutation importance, the XGBoost gain importance, and the absolute SHAP values. We also examine the local feature importance using SHAP values.

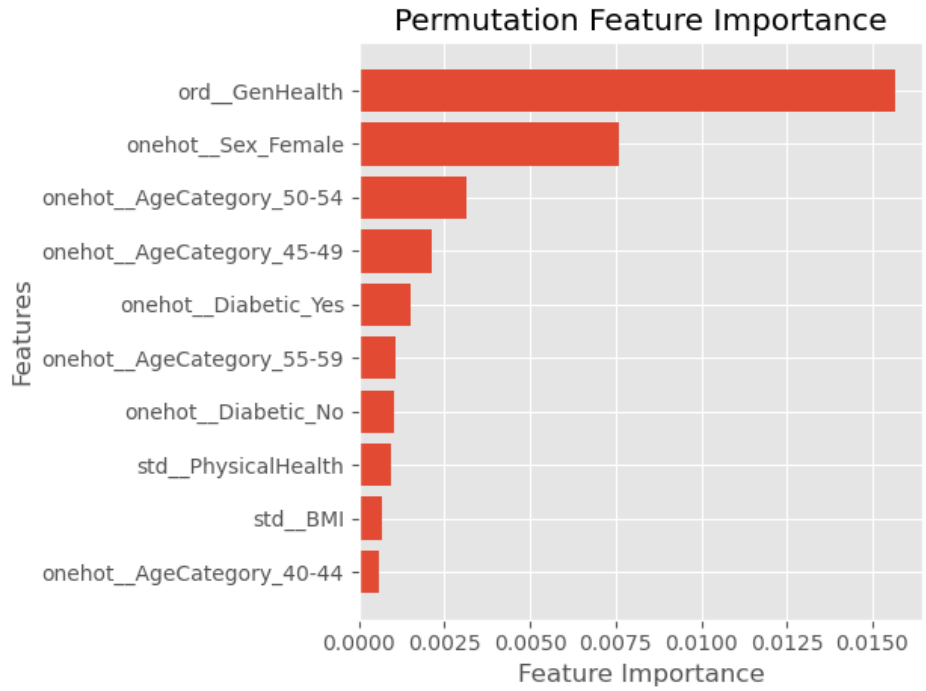

Figure 5. Permutation Feature Importance

Figure 5 displays the permutation feature importance using the XGBoost classifier. We calculate the importance by randomly shuffling individual features and recording the decrease in model score. The ordinal-encoded "GenHealth" feature is the most important feature, followed by "Sex_Female" and "AgeCategory_50-54."

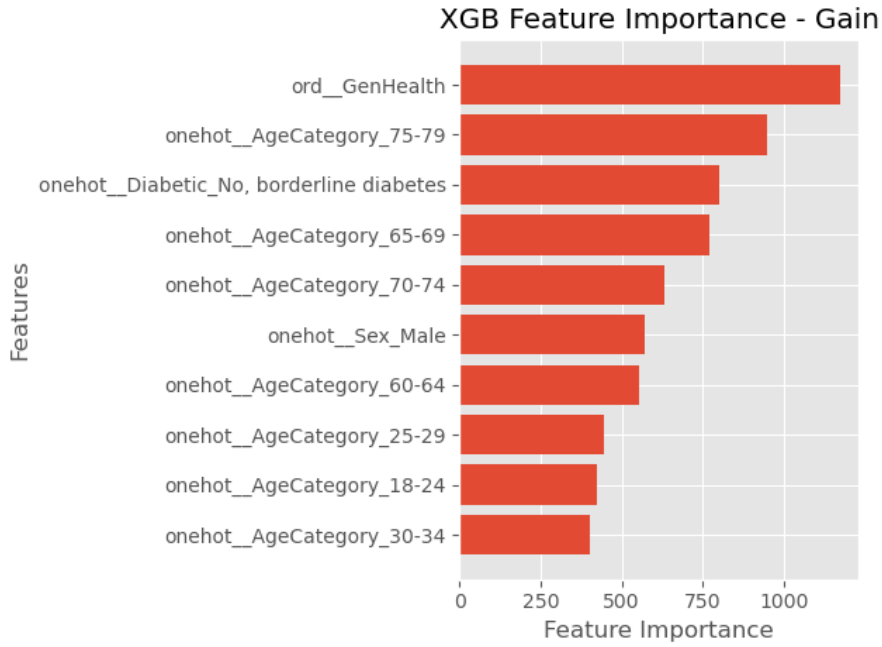

Figure 6. XGBoost “Gain” Feature Importance

Figure 6 shows the XGBoost feature importance using the gain metrics. The “gain” feature importance measures the average contribution of each feature across all splits. [3] The top features include “GenHealth”, “AgeCategory_75-79”, and “Diabetic_No.”

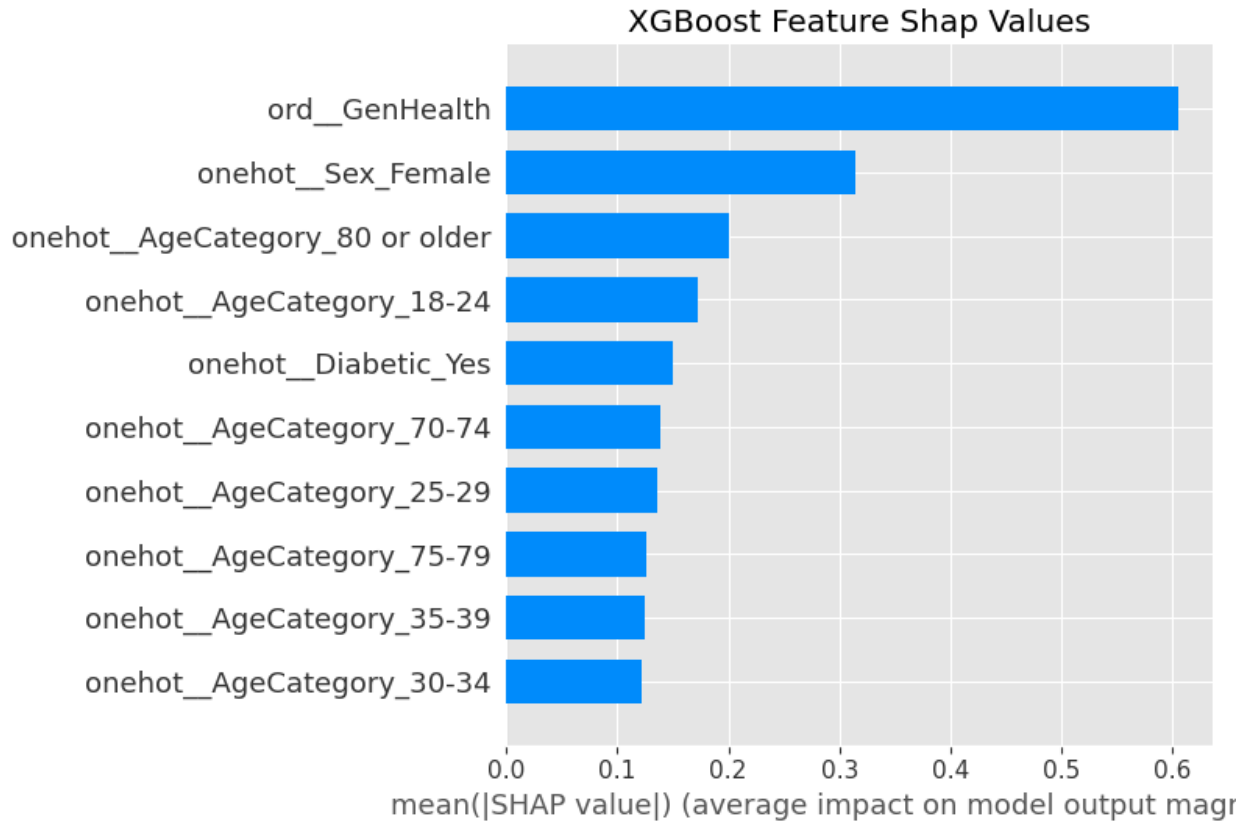

Figure 7. SHAP Global Feature Importance

Figure 7 shows the absolute global SHAP value feature importances. The SHAP value uses cooperative game theory to compute the contribution of each feature in the model’s decision making. We see that the top features include “GenHealth”, “Sex_Female”, and “AgeCategory_80 or older”.

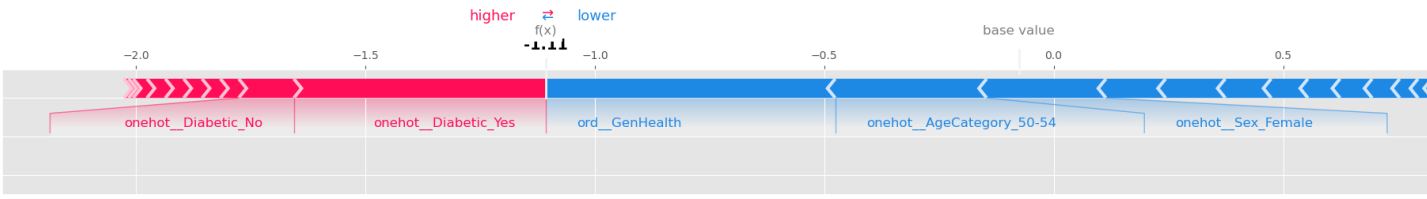

Figure 8. SHAP Local Feature Importance

Figure 8 shows the SHAP local feature importances. We take a random data point and examine which feature plays the most important roles in the model's decision making. The result shows that "GenHealth" contributes the most in influencing the model to predict class 0 while "Diabetic_Yes" influences the model to predict class 1.

Overall, although the three global feature importances show some disparities, all three have "GenHealth" as the most important feature. This is not surprising since the patient's general health status can be significantly influential in predicting whether they have heart diseases. Besides that, age category features also play important roles in the model's decision making. One unexpected result was that although studies have shown that the consumption of alcohol and tobacco can lead to higher chances of heart disease, neither feature appears among the top features. [4]

Outlook

To improve our model's predictive power, we can include additional features, such as a patient's family history of heart diseases, their comorbidities (liver diseases, lung diseases, etc.), and their happiness metrics. Another approach we can adopt is using up-sampling techniques, such as the SMOTE method, to counteract the dataset's imbalance.

References

1. Heart Disease Facts. Centers for Disease Control and Prevention. [2022-10-14]

https://www.cdc.gov/heartdisease/facts.htm .

2. Personal Key Indicators of Heart Disease. Kamil Pytlak. [2022-02-18]

https://www.kaggle.com/datasets/kamilpytlak/personal-key-indicators-of-heart-disease

3. Consistent Individualized Feature Attribution for Tree Ensembles. Scott M. Lundberg, Gabriel G. Erion, Su-In Lee. [2018-02-12]

4. The Effects of Smoking and Drinking on Cardiovascular Disease and Risk Factors. Kenneth J. Mukamal. Alcohol Res Health. [2006]