ENSE 411

Artificial Intelligence

Final Report

Winter Semester 2025

Predicting the Risk Of Heart Disease using XGBoost

Instructor: Dr. Kin-Choong Yow

Tolani Oke-Steve
200429404

1.0 Intro

1.1 PROJECT SUMMARY

This project will focus on the development of a prediction model that will be able to effectively predict the risk of having heart disease based on certain factors such as Cholesterol level, Blood pressure, chest pain type and more. Using the UCI dataset obtained from kaggle. This model will be implemented using XGBoost, a fast gradient boosting algorithm known for its accuracy. To enhance this model's prediction I will also be applying the Synthetic Minority Over-Sampling technique (SMOTE) to address class imbalance so that when data is fed into this model it does not focus on the majority of the data which is patients without heart diseases instead it focuses on patients with heart diseases the minority data in order to regularly give accurate predictions. To evaluate the performance of this model I will be using performance metrics such as accuracy, F1-Score, AUC-ROC curves and more.The primary goal here is to produce a model with at least 85% accuracy that provides health care professionals with a stable tool for early detection of heart diseases.

1.2 Problem Description

Heart disease remains one of the leading causes of death worldwide. Early detection of heart problems can greatly increase patient chances in the event of successful intervention. Predicting the risk of heart disease can be challenging due to the complex nature of datasets. While various machine learning models exist that can assist in detecting the likelihood of heart diseases effectively. Training these models do take up alot of time and resources. To address this issue I would be using XGBoost, a fast gradient boosting algorithm combined with Synthetic Minority Over-Sampling technique (SMOTE) to handle data imbalance. This approach aims to improve prediction accuracy while maintaining efficiency.

1.3 PROJECT OBJECTIVE

The objective of this project is to implement the XGBoost model to help health professionals in the early detection of heart diseases enabling appropriate intervention to improve patient outcomes.

2.0 Approaches and Methodologies Investigated

Throughout the entirety of this project various methodologies were used to achieve the set objectives. This section discusses the various methodologies I explored throughout the entirety of this project.

2.1 Linear regression

Initially, during the start of this project the first approach I investigated was linear regression which involved a statistical method to model relationship between dependent and independent variables and this model assumes relationships are entirely linear. But this idea was dropped due to the simplicity of the model which made it unfit for this project.

2.2 Youden's index

Youden's index is a statistical measure used to evaluate the performance of tests that differentiates two groups e.g presence or absence of a disease. This approach combines specificity and sensitivity into one value using this formula: Youden index = (Specificity + Sensitivity - 1). This approach was used to find the optimal cutoff point to ensure proper balance for this project's dataset.

2.3 XGBoost

This is the approach I used, XGBoost (Extreme gradient boosting), an optimized gradient boosting algorithm that is designed to be portable, efficient and flexible. XGBoost builds trees sequentially where each new tree built attempts to correct errors of the previous trees. The concept behind XGBoost is boosting where multiple weak decision trees are combined to create a strong predictive model. XGBoost stands out among traditional gradient boosting techniques by introducing features such as regularization (to reduce overfitting), parallel processing for faster training, and handling missing values automatically. Due to its high predictive capacity, XGBoost is used in real world services such as Amazon web services (AWS) sagemaker for fraud detection. Below I have provided a simplified diagram of the XGBoost algorithm.

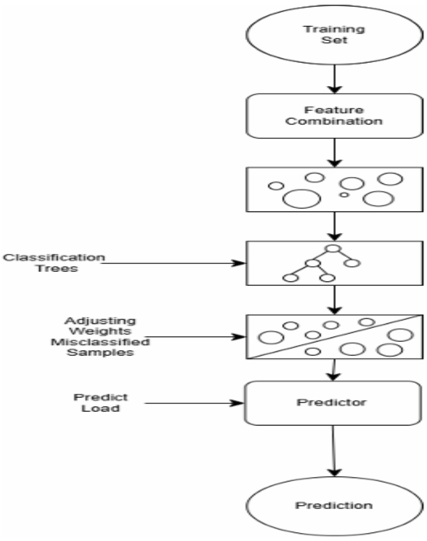


Figure 1: Diagram of XGBoost algorithm

3.0 Methods

3.1 Data

The UCI heart disease dataset used in this project was obtained from kaggle (Link: UCI Heart Disease Data), this dataset contained attributes such as age, sex, cholesterol, blood pressure, chest pain type and so on. These attributes served as predictors, while the target variable num was used to indicate the presence or absence of heart diseases. The dataset was cleaned using the mean to encode numerical columns and mode to encode categorical columns. Additionally, I applied one hot encoding to this dataset to allow easy processing of the values within this set. Outliers were identified and capped using the IQR (Interquartile Range) method to improve data quality. After which Synthetic Minority Over-Sampling technique (SMOTE) was applied to the training set of the data to generate samples for the minority class to address class imbalance. Below is a diagram showing the application of SMOTE to the training set as described above.

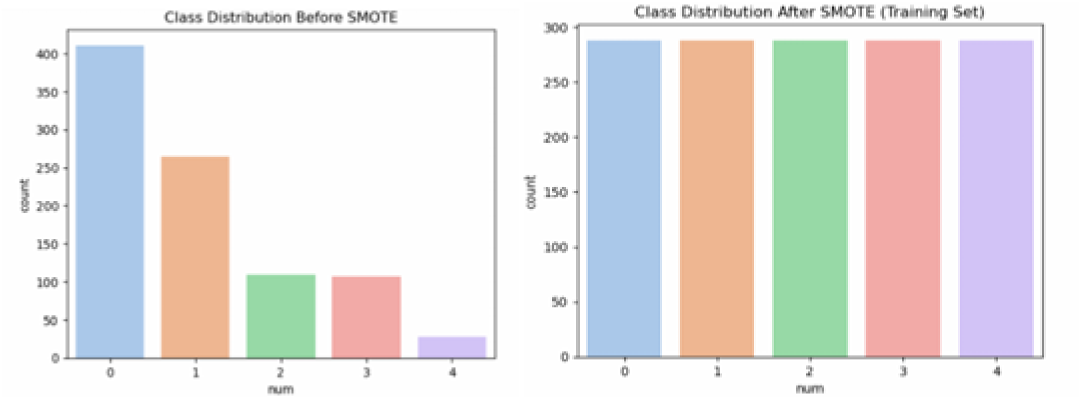


Figure 2: Application of SMOTE to training set of data

3.2 Correlating Features

Before building any model, the first step is to evaluate data within the provided dataset. While some features had obvious correlations to target value for the presence of heart disease (num). The feature with the most correlation was oldpeak as shown in figure 1. Other features such as number of major vessels (ca) and age showed great correlation with heart disease. While features such as thallium stress (thalch), cholesterol level (chol), and resting blood pressure displayed weaker correlations than expected.

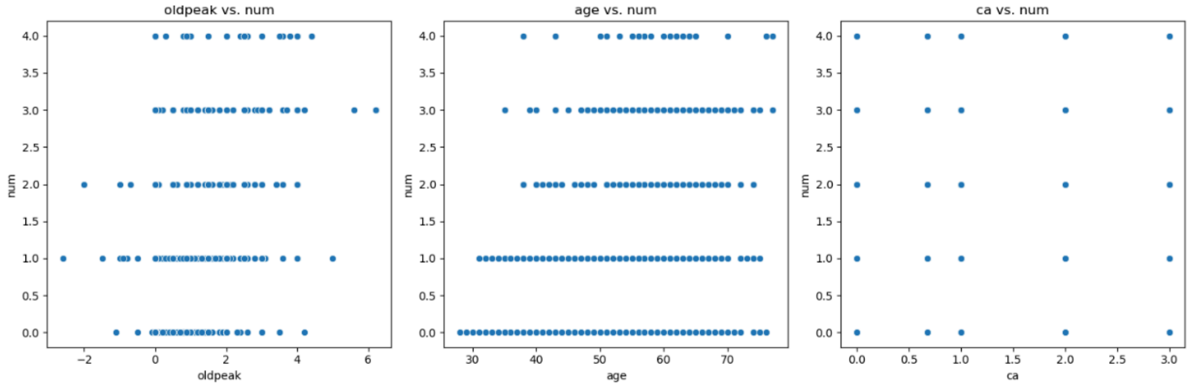


Figure 3:The target variable, num, represents the presence of heart disease. Oldpeak, age, and ca correlated most with num

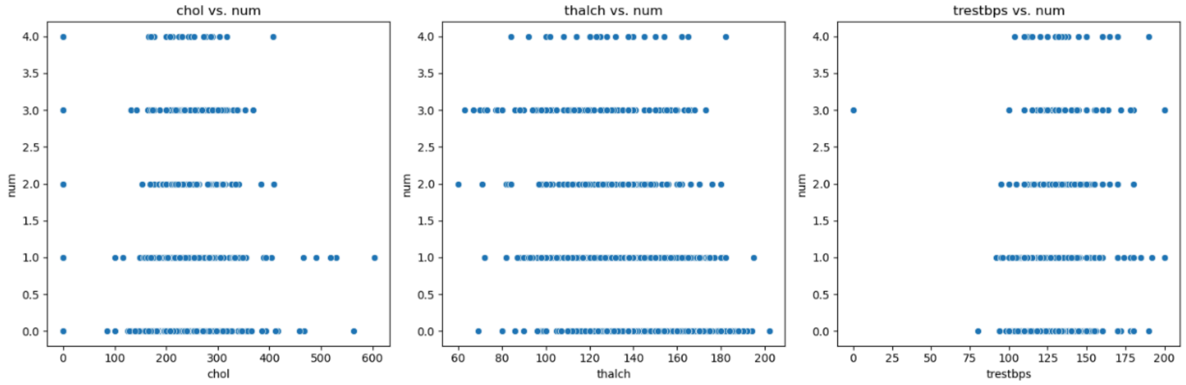


Figure 4: With the target variable, num, representing the presence of heart disease. Chol, Thalch and Trestbps are weakly correlated with num.

While individual factors such as cholesterol level, resting blood pressure may not exhibit strong standalone correlation with heart disease risk. Combining these features together during the training process did help the model identify underlying patterns which helped enhance this model's predictive capacity.

4.0 Implementation details

For this project the XGBClassifier library was imported and used to solve the problem of heart disease prediction. Using this basic idea: Detection and classification of heart disease data. The foundation for this was achieved using libraries such as matplotlib and seaborn for graphs, numpy for computing data and pandas for data manipulation. XGBoost was trained using the training set, and then I applied hyperparameter tuning manually, where I performed a sweep across all parameters to find the values that will best fit this model. An attempt was made to use the validation set to adjust the hyperparameters but that failed likely due to incorrect configuration of the required parameters. Hence, that idea was abandoned, instead the hyperparameters were tuned manually for the remainder of this project.

The hyperparameters I used in Building, and training of this model are as follows:

1. eval_metric = "mlogloss"
2. max_depth = 5
3. learning_rate = 6.9%
4. n_estimators = 201 (This gives us more trees for better performance)
5. gamma = 98%
6. subsample = 80%
7. colsample_bytree = 80% (This helps with overfitting of the model)

Additionally, In order to simplify the amount of classes I have to deal with, I applied target transformation which is a subset of feature engineering, to simplify the prediction classes into two main classes (classes 2, 3, 4 representing severity of heart disease) combined into class 1 signifying the presence of disease and (class 1 mild severity of heart disease) was combined to class 0 which signifies the absence of heart diseases. After that, I applied Youden's index to determine the ultimate threshold for classifying between disease (1) and no disease (0). I have provided the formula I used for Youden's index below.

$$\text{Youden's Index} = \text{Recall} + \text{Precision} - 1$$

Figure 5: Formula for Youden's index

5.0 Results

The dataset set I used in this project contained 920 data points which were divided into 70% training and 30% testing. Additionally, the training data was split into 90% training, 10% validation which gave me an overall training and test split of 9:1:4. After which XGBoost was applied and testing was carried out, which resulted in an accuracy of **84.1%** which was less than the target of 85%. Initially four classes were used during the initial prediction process but these classes were later combined into two major classes (0 absence of disease, 1 presence of disease) which were evaluated using F1 score, recall, precision and support with the results shown in the table (table 1) below. In the table below, macro average represents the mean of each performance metric for the disease and no disease class. Weighted average represents the weighted sum of the metrics given the number of instances of values in each class.

Table 1: Model performance on the test set using various evaluation methods.

	Precision	Recall	F1-Score	Support	Accuracy
No disease (0)	0.94	0.68	0.79	123	0.84
Disease (1)	0.79	0.97	0.87	153	0.84
Macro avg	0.87	0.83	0.83	276	
Weighted avg	0.86	0.84	0.84	276	

When my model was evaluated with the confusion matrix heatmap (figure 3), it correctly predicted 143 cases of true positive (TP) heart disease, my model also correctly predicted 86 cases of no disease true negatives (TN) as shown in the diagram below. This model also incorrectly classified 37 cases of no disease as heart disease false positives (FP) while also incorrectly classifying 10 cases of actual heart disease as no disease false negatives (FN).

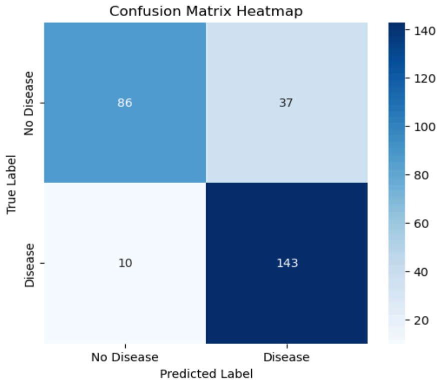
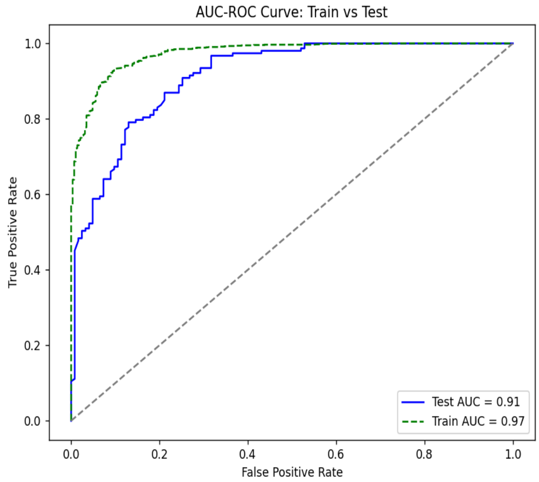


Figure 6: Confusion matrix heatmap

Further evaluation was conducted on this model’s performance using AUC-ROC and the precision recall plot. The AUC-ROC curve provided in (figure 4) below, compares the sensitivity of true positive(TP) rate against the false negative (FN) rate which shows my model's ability to differentiate between classes. The AUC scores 0.97 training set, 0.91 test set indicate a strong preference in differentiating cases with, without heart disease. The slight drop in the AUC indicates that my model might be overfitting in some cases, but the high scores confirms that my model is reliable in classifying cases correctly.

The precision recall plot (PR) plot in figure 5 below with an optimal threshold of 0.39 indicates that a large portion of diseases for this model are identified without overly increasing the amount of false positives.



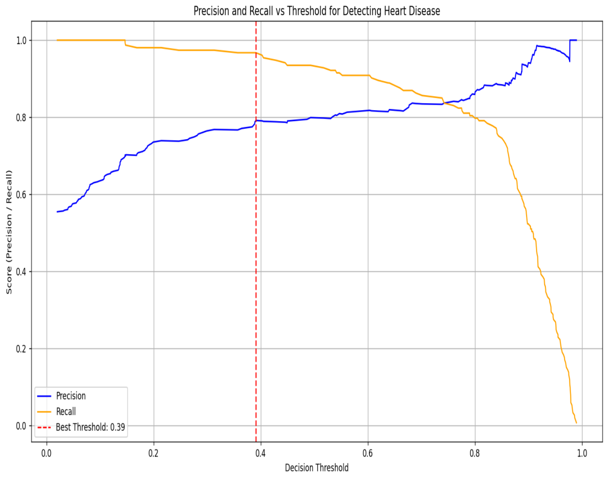


Figure 7: AUC-ROC curve

Figure 8: PR plot

6.0 Conclusion and Future work

While my implementation was just short of the set target of 85%, when it comes to heart disease prediction the work done by others in this field using XGBoost proves that this approach is sufficient for detection of heart diseases. Despite falling short of my placed target for this project I feel like I have learned a lot about machine learning (XGBoost) and classifying data, and how difficult it can be to solve issues in a complex model given a limited dataset compared to my prior knowledge in this area.

In terms of future work, I believe that I can certainly improve the predictive power of my model if I had access to a larger dataset with similar data points. The current data set of 920 data points has limited my model's capacity to generalize effectively leading to the shortfall in accuracy. Additionally, If I had to reassess this project I would explore alternative approaches such as deep neural network (DNN), or ensemble methods which could further improve the predictive performance of this model. Lastly, since this model was developed to be used in a clinical setting, I would re-evaluate this model's ability to interpret medical records in order to determine its clinical relevance.

References:

1. Codezup. A step-by-step guide to building a predictive model with Python and XGBoost. A Step-by-Step Guide to Building a Predictive Model with Python and XGBoost
2. Diego Taquiri (n.d.). HeartDiseaseMLInterpretation. diego-taquiri/HeartDiseaseMLInterpretation: Heart disease prediction model using XGBoost, with hyperparameter tuning, model evaluation, and interpretability through SHAP.
3. Crossman-Smith, Jamie. "XGBoost Explained: A Beginner's Guide." *Low Code for Data Science*, Medium,XGBoost Explained: A Beginner's Guide | by Jamie Crossman-Smith | Low Code for Data Science | Medium
4. Budholiya, Kartik, et al. "An Optimized XGBoost Based Diagnostic System for Effective Prediction of Heart Disease." *Journal of King Saud University - Computer and Information Sciences*, vol. 34, no. 7, 2022, pp. 4514–4523, An optimized XGBoost based diagnostic system for effective prediction of heart disease - ScienceDirect
5. Heart disease UCI dataset. Kaggle: UCI Heart Disease Data
6. Yadav, Amit. "XGBoost vs Gradient Boosting." *Medium*, 7 Jan. 2025, XGBoost vs Gradient Boosting. I understand that learning data science… | by Amit Yadav | Medium
7. Najeebuddinm98. "Prediction on UCI Heart Disease Dataset Using the XGBoost Library.", xgboost_heartdisease_pred/prototyping.ipynb at main · najeebuddinm98/xgboost_heartdisease_pred
8. Learn Statistics Easily "What is Youden Index - Understanding Its Importance.", What is: Youden Index - Understanding Its Importance