Predicting Outcomes in Cardiovascular Disease

Anyan Liu, Wenxin Ni, Kate Wasmer, You Wu

2024-12-17

Abstract

Heart disease is closely related to morbidity, mortality and high medical cost. Due to its high prevalence and severe risk, accurately predicting heart disease is essential. Machine learning algorithms have proven effective in predicting disease. Our study aimed to access and summarize the performance of several machine learning models in predicting heart disease. In this study, heart disease medical records were retrieved from the Cardiovascular Disease dataset (Ulianova, 2018) on Kaggle with a sample size 70,000. The dataset was randomly divided into 80% training and 20% testing sets. With 14 potential predictors, four machine learning models (SVM, Random Forest, XGBoost, HistGradientBoostingClassifier) were employed and used hyperparameter optimization in our study. For predictive performance, XGBoost had the highest Accuracy of 0.74, the highest Recall of 0.78 and the best AUROC of 0.80. RF had the fastest computation time of 17.51 minutes. HistGradientBoostingClassifier had the highest precision of 0.76 comparable to other models. We conclude that the XGBoost model outperforms other algorithms in predicting heart disease. However, there exists heterogeneity among ML algorithms in multiple parameters. Clinicians can leverage these differences to apply the most suitable algorithm for their dataset, thereby optimizing predictive performance.

Introduction

Heart disease is the leading cause of death in the United States and does not discriminate against sex or race. According to the Center for Disease Control and Prevention, one person dies from cardiovascular disease every 33 seconds. This translates to roughly 2,600 deaths per day, which actuates the need for correctly identifying and diagnosing this condition. Furthermore, individuals who are less likely to develop heart problems (e.g., those under the age of 50) often fall under the radar, and therefore do not receive the proper medical care. The lack of preventative measures in cases like these lead to fatal consequences, resulting in orphaned children and wrongful deaths. However, age is only one of many factors that play a role in cardiovascular disease.

By analyzing datasets with a wide range of predictors and ensembling machine learning models, our objective in this project was to determine whether or not an individual will develop heart disease with the greatest possible accuracy.[1] The alarming statistics for this condition propelled us to engineer and fine-tune different classifiers to obtain not only a high level of accuracy, but also desirable precision and recall rates. Alongside the overarching goal of achieving optimal performance metrics, we also investigated the strength of each feature in our dataset, determining which factors are the most integral to developing heart disease.

Methods

Collecting and handling data for our research question provided a computational challenge and prompted us to reconsider our methodology. One of the greatest obstacles was establishing an acceptable sample size for

[1]The link to our GitHub is listed in the References section, underneath Liu, A. et. al. For a direct link, please access https://github.com/KatherineWasmer/biostat_625_final

our data set. As is the case with any machine learning model, generalizability and robustness are essential for obtaining reproducible results. After weighing the benefits and limitations of a variety of Kaggle datasets, we decided on the Cardiovascular Disease dataset (Ulianova, 2018). With a sample size of 70,000 patients and a variety of features that captured demographic, clinical, and diagnostic measurements, we considered this data conducive with our research objectives.

Before implementing any machine learning algorithms on the data, we spent a day on preprocessing. Kate utilizes the pandas library in Python to conduct "feature engineering". The following changes[2] were made to the original dataset to maximize understanding of the given data, and to stay consistent with scientific facts:

1. For the patients' age, we modified the units from days to years for better readability. We developed a simple apply() method in pandas that returned a new column called "age_years", by dividing each value in the "age" column by 365. We then dropped the "age" column, since it was no longer necessary for our analysis.

2. We implemented a lambda function that assigned each individual to a blood pressure category based on their systolic and diastolic readings. For different groups, we relied on the American Heart Association's most recent diagnostic criteria, yielding 5 different categories: normal, elevated, high blood pressure stage 1, high blood pressure stage 2, and hypertensive crisis. This new feature was labelled "BP Category".

3. We also feature engineered a column for the BMI, using an analogous lambda function as the one in (2). Given that the author of this dataset was based in Toronto, the height and weight were measured in centimeters and kilograms. We simply used the formula (kilograms/meters^2) to compute the BMI for each individual.

4. In the sex column, we modified the encoded data (where 1=female and 2=male) to 'F' and 'M' for increased comprehension.

5. The reported systolic and diastolic blood pressures caused a problem in the scientific integrity of our research. Many of the measurements were not recorded accurately; examples included negative blood pressures (which is not possible), or cases where the systolic number was lower than the diastolic, which cannot realistically happen. To avoid making false assumptions about this erroneous data, we replaced these blood pressure values with null values, which would later be imputed, natively used in a classifier, or just deleted from the dataset.

In the next 4 sections, we describe the classifiers that each individual used.

Support Vector Machine

Support Vector Machines(SVM) are powerful models widely used for classification problems. SVMs are especially effective handling high-dimensional data, although our dataset only includes 14 variables, they are still suitable because SVMs still can find optimal decision boundaries. Also, the correlation matrix plot shown before shows weak linear relationship, which means that compared to linear model, SVM can be a good choice. Besides, SVMs have good resistance to overfitting, especially for low-dimensional datasets like the one we used.

The issue of missing values was particularly important for the SVM. Since the records containing null values accounted for less than 5 percent of the sample size, we decided to directly remove the null values.

Initially, we used the Linear kernel for training the SVM model. However, the training process took an exceptionally long time, making it computationally inefficient. To address this, we switched to the Radial kernel,

[2]The source code for changes 1-4 can be found at https://github.com/KatherineWasmer/biostat_625_final/blob/main/initial_filter_cardio.ipynb, and the output csv file can be found at cleaned_data_v1.csv underneath the main branch of our GitHub. For change (5), please refer to cleaning_irrational_data.ipynb and cleaned_data_v2.csv.

which significantly reduced the training time. Additionally, we compared the accuracy of two groups and found that they are comparable. Therefore, we proceeded with the Radial kernel for further hyperparameter tuning.

Then, we performed 10-fold cross-validation grid search to tune the hyperparameters cost and gamma. After comparing the results, we selected the combination of these two hyperparameters that yielded the highest accuracy.

Random Forest

To predict cardiovascular disease classification, the Random Forest algorithm was employed as an ensemble method consisting of multiple decision trees. Each tree was built using bootstrap samples, and at each node, a randomly selected subset of features was evaluated for optimal splitting based on the Gini impurity criterion, enhancing model diversity and reducing overfitting. Final predictions were determined by majority voting across all trees.

To rigorously evaluate the model's performance, the dataset was randomly divided into training and testing sets, with missing values imputed using the na.omit method. To optimize the model performance, hyperparameter tuning was conducted using a 5-fold cross-validation strategy. By defining the hyperparameter search space for the Random Forest, including the number of randomly selected features, the impurity criterion for node splitting, and the tree depth, the optimal parameter combination was determined based on cross-validated performance metrics.

Additionally, the importance of features in RF was assessed using the Mean Decrease Gini (MDG) index. Variables with higher MDG values were considered to have greater predictive significance for cardiovascular disease. Then, we use the final RF on the testing sets.

XG Boost Classifier

The XGBoost algorithm was used to build and evaluate a predictive model for cardiovascular disease (CVD) using a gradient boosting algorithm. We began with the data preprocessing, where missing values were removed and categorical variables were numerically encoded. The dataset was split into 80% training and 20% testing subsets using stratified sampling to maintain class balance.

A 5-fold cross-validation with a grid search was applied for hyperparameter tuning in order to optimize model performance. The search space included parameters such as the learning rate, tree depth, row sampling ratio, column sampling ratio and minimum loss reduction. Using the best-tuned hyperparameters, the final XGBoost model was trained and evaluated on the test set. Performance metrics, including accuracy, precision, recall, and the ROC-AUC score, were calculated to assess model effectiveness. Additionally, feature importance was analyzed using the xgb.importance function to identify the most influential variables in predicting heart disease.

HistGradientBoostingClassifier

The HistGradientBoostingClassifier (HGBC) is seen as an extension of the XGBoostClassifier, with native support for both null data and categorical data. These two properties accelerated our workflow and resulted in quick computation. The former indicated that imputation wasn't necessary for the dataset. Given that most data is not missing completely at random (MCAR), the imputation methods could lead to some potential bias in the prediction, so having the ability to retain null values is especially effective. Moreover, the native support for categorical data eliminated the need to encode non-numerical columns, which is often a crucial step in the machine learning pipeline. This classifier is also adaptable to large data sets; it is considered a fast estimator that doesn't require a large amount of memory. It is often difficult to find a balanced tradeoff between speed and memory when working with big data, so the HGBC is an excellent choice for fixing this problem.

We ran two different sets with the HistGradientBoostingClassifier; the first set included all 70,000 records due to the native support of missing values, but the second removed the null values to provide a baseline comparison with the other classifiers. For the latter subset, there were 68,680 samples. The procedure for both sets was consistent: first, 20% of the records were randomly assigned as test data, and 80% were used to train the model, with a 10-fold cross-validation. A pseudo-random seed was chosen to maintain the same test-train split, while also remaining unbiased in the selection.

Results

Performance of heart disease prediction models

	20% Testing Dataset				
Algorithm	**AUROC**	**Accuracy**	**Recall**	**Precision**	**Speed_minutes**
SVM	0.76	0.72	0.69	0.72	32.09
Random Forest	0.79	0.72	0.76	0.72	16.56
eXtreme gradient boosting	0.80	0.74	0.78	0.72	18.11
HistGradientBoostingClassifier	0.80	0.73	0.69	0.76	42.39

Figure 1: Table 1: Comparison of metrics across different classifiers (null values dropped)

The performance metrics of the four models are shown in Table 1. In our study, XGBoost demonstrated the best performance, attaining the highest AUROC value of 0.80 and the highest accuracy of 0.74, while having a relatively fast computation time of 17.51 minutes, presenting its robust predictive ability. Random Forest followed closely with an AUROC of 0.79, a recall of 0.76, and an accuracy of 0.72. Its computation time was 16.56 minutes which was the fastest. In contrast, the HistGradientBoostingClassifier (n = 68,680) achieved the highest precision of 0.76 and the highest AUROC of 0.80, but it required the longest runtime of 42.39 minutes, indicating a disparity between prediction precision and computational efficiency. The efficiency and precision are supported by the comparison of the HGBC on the entire dataset (n=70,000); with the null values included, it had a precision of 0.76, a recall rate of 0.7, an accuracy of 0.734, and an AUROC of 0.8. These metrics were essentially identical, which can be attributed to the large sample size. However, the slight increase in recall rate suggests that retaining the null values in a model that natively accepts them is the best approach in general. The SVM model performed moderately across all aspects. Overall, XGBoost was the most effective model in the study.

Feature Importance Rankings

Variable importance was determined by the coefficient effect size for machine learning models. For XGBoost, the most important 3 features are physical activity, age and blood pressure. For HGBC, the most important 3 features are physical activity, cholesterol and age. For Random Forest, the most important 3 features are BMI, physical activity and age. For SVM, the most important 3 features are physical activity, age and alcohol intake. Additionally, we found that age and physical activity are always important in our models, so we generally deduce maybe they are important predictors which provide effective guidance for clinicians.

Conclusion

In this study, we investigated the performance of four machine learning models-**XGBoost, Random Forest, SVM, and HistGradientBoostingClassifier**—to predict heart disease outcomes using a dataset of 70,000

sample sizes with 14 features. We also conducted a feature importance analysis to detect the most important predictors in each model so that generally identify the most significant predictors of heart disease.

Our study shows that XGBoost provides the best overall performance, achieving the highest AUROC (0.80) and accuracy (0.74), while maintaining a relatively fast computation time of 17.51 minutes. This highlights its effectiveness as a robust and efficient classifier. However, in real-world scenarios, clinicians should further consider the actual data quality and diagnostic requirements to select the optimal model. Additionally, we find that age and physical activity are always the most important predictors of heart disease.

References

American Heart Association. (2024, May 17). *Understanding blood pressure readings.* American Heart Association. https://www.heart.org/en/health-topics/high-blood-pressure/understanding-blood-pressure-readings

CDC. (2024, April 29). *Heart Disease Facts.* CDC. https://www.cdc.gov/heart-disease/data-research/facts-stats/index.html

HistGradientBoostingClassifier. (2024). Scikit-Learn. https://scikit-learn.org/1.5/modules/generated/sklearn.ensemble.HistGradientBoostingClassifier.html

Liu, A. et. al. (2024, December 17). *biostat_625_final.* [Github Repository] https://github.com/KatherineWasmer/biostat_625_final

Python Machine Learning - AUC - ROC Curve. (n.d.). Www.w3schools.com. https://www.w3schools.com/python/python_ml_auc_roc.asp

Ulianova, S. (2018). *Cardiovascular Disease dataset* [Dataset]. Ryerson University. https://www.kaggle.com/datasets/sulianova/cardiovascular-disease-dataset