A Comparative Study of Machine Learning Models for Heart Disease Diagnosis Prediction.

Victor Odoh

School of Computing, Engineering & Digital Technologies
Teesside University, Middlesbrough
C2397722@live.tees.ac.uk

Abstract

This project aims to build and compare different machine-learning (ML) models for heart disease diagnosis prediction, using a dataset of key indicators of heart disease. The dataset consists of 18 attributes and over 320,000 instances, with a highly imbalanced response variable among others. Four machine learning algorithms were selected for the study: LightGBM, XGBoost, Random Classifier, and Logistic Regression with built-in cross-validation. Previous studies have reported high accuracy and precision in predicting heart disease diagnosis using similar ML models on different datasets. The results from the experiments in this study will provide insights into the effectiveness of different ML models in predicting heart disease diagnosis on the chosen dataset, and contribute to the development of accurate and effective prediction models for heart disease diagnosis in general.

Keywords: Machine_learning algorithms, Heart disease, LightGBM, XGBoost, Random Forest, Logistic RegressionCV

1. Introduction

Early detection and diagnosis are crucial for successful treatment and prevention of heart disease, a leading cause of morbidity and mortality worldwide. With the emergence of ML as a powerful tool for predicting heart disease diagnosis using clinical and demographic data, this study aims to develop and compare various ML models for heart disease diagnosis prediction. The models are trained on the 'Key Indicators of Heart Disease' dataset obtained from Kaggle.

In this study, the problem is to build for heart disease diagnosis prediction using the chosen dataset and to perform a comparative study on their performances. The dataset presents several challenges, including a significant class imbalance in the response variable, and an imbalanced protected group (race) in the dataset among others.

Early detection and diagnosis of heart disease are essential for effective treatment and prevention. The use of machine learning models for predicting heart disease diagnosis based on clinical and demographic data has the potential to enable early intervention and prevention, which can have a significant impact on reducing morbidity and mortality due to heart disease. Therefore, developing an accurate and effective predictive model for heart disease diagnosis is of critical importance in healthcare, and this study can contribute towards achieving this goal.

Four ML algorithms were selected for this study; LightGBM, XGBoost, Random Classifier, and Logistic RegressionCV. I chose these algorithms based on their ability to handle imbalanced datasets, their performance in predicting binary classification problems, and their widespread use in medical diagnosis prediction tasks.

2. Literature Review

Previous studies have demonstrated the potential of ML in predicting cardiovascular disease diagnosis based on clinical and demographic data. Karthick et al. (2022) used the chi-square statistical test to select specific attributes from the Cleveland heart disease dataset and employed several ML algorithms, including SVM, Gaussian Naive Bayes, logistic regression, LightGBM, XGBoost, and random forest, for developing a heart disease risk prediction model. The study reported that the random forest algorithm achieved the highest accuracy of 88.5% during validation, using 303 data instances with 13 selected features of the Cleveland HD dataset.

Ghasemi et al. (2023) highlight the development of new ML approaches, such as deep neural networks and gradient boosting machines, which are able to classify data with high precision and accuracy, even in cases of imbalanced data and nonlinear dependencies in high dimensional spaces. The study employed an ensemble method that combined three different models, including deep neural networks, LightGBM, and XGBoost, to detect heart disease using an imbalanced medical dataset. The results showed high accuracy of 91.75% and f1_score 94.4, indicating the effectiveness and robustness of the approach. The study emphasizes the potential of ensemble methods for achieving high accuracy in heart disease diagnosis using ML algorithms.

3. Methodolgy:

Experiments to be done: The goal is to compare the model performances for the following scenarios:

- Unsampled data
- Oversampled data with outliers (Using SMOTE Oversampling Technique)
- Oversampled data without outliers (SMOTE)
- Oversampled data with outliers (Using ADASYN Oversampling Technique)
- Oversampled without outliers (ADASYN)

Based on the extreme class imbalance of the response variable, undersampling the majority class will result in too much data loss, therefore the technique was not adopted. I decided to adopt 2 oversampling techniques (for comparisons later on) to ensure that the model has enough data to learn from.

I decided to first split the data into the training and test sets before applying some preprocessing steps for the following reasons:

- Some preprocessing steps, such as scaling, involve using information about the distribution of the data. If the entire dataset is scaled before splitting, the scaling parameters are learned from the entire dataset, including the test set. This can result in data leakage, where information from the test set is unintentionally incorporated into the training process, leading to overly optimistic evaluation metrics.

  The same goes for oversampling. Hence, the training data is scaled separately and the testing data is also scaled separately but using the same scaling parameters as the training data. This ensures that the input features are in the same range and distribution as the ones the model was trained on. Only the training data sets were oversampled.

- Preprocessing steps may introduce biases or artifacts in the data that affect the performance of the model. Applying preprocessing steps separately to the training and testing sets can ensure that the evaluation metrics are representative of the true performance of the model on unseen data.

Below is my workflow:

I. Explore the data

II. Sort and encode the categorical attributes.

III. Concatenate the dataframes: Encoded categorical attributes and numerical predictors.

IV. Save a copy of the new concatenated dataframe.

V. Remove Outliers from the dataframe copy.

VI. Split the dataframes (with and without outliers) to their training and test sets.

VII. Apply preprocessing steps of Scaling, and oversampling (Using SMOTE, and ADASYN) to the training data sets (with and without outliers). I Oversampled the minority class in the training data sets alone, to ensure that the model sees enough examples of the minority class. The test data is kept separate and not manipulated, so that the performance of the model can be accurately assessed on unseen data.

1. Train the models on the respective training sets and evaluate their performance on the respective testing sets for comparisons.

3.1. Data Exploration

The data is analysed for possible errors, missing values, outliers, and class imbalance, especially in my chosen response variable (*HeartDisease)* to guide my preprocessing steps. The **Y-Data Profiling** library is used for the exploratory analysis. It is a great tool that helps you extract important information from dataframes in one shot and saves the stress of calling df.head(). df.describe(), df.info() etc. multiple times to perform EDA.

Observations:

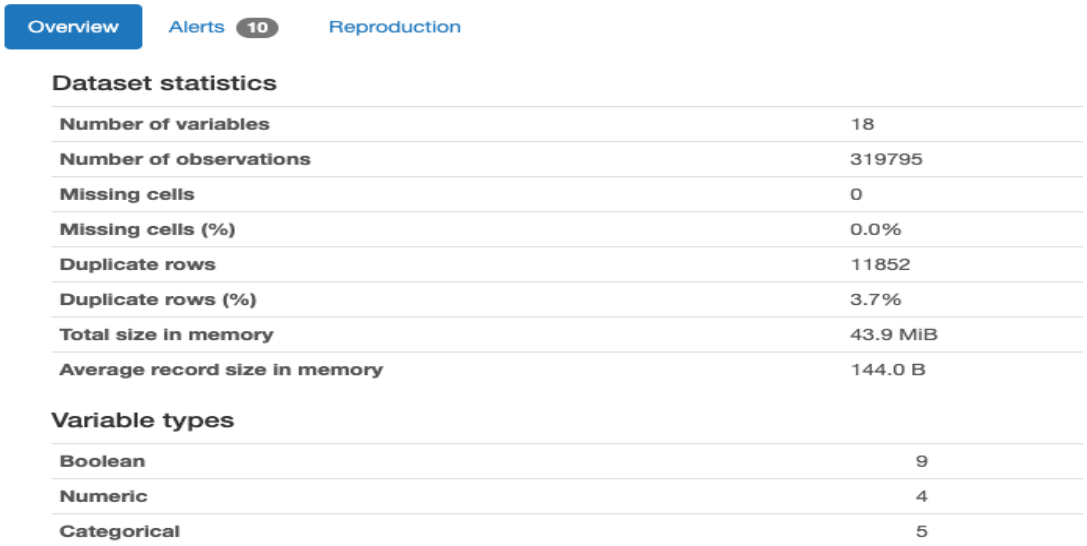

Figure 1.0 Data Overview

- The data is made up of 18 attributes and about 320k instances.
- Of the 18 attributes, 4 are numerical and 14 are categorical (9 binary and 5 non-binary); 17 predictor attributes.
- There are no missing values in the data
- There has been no evidence of errors
- There a 11,856 duplicate rows (3.75% of the data)

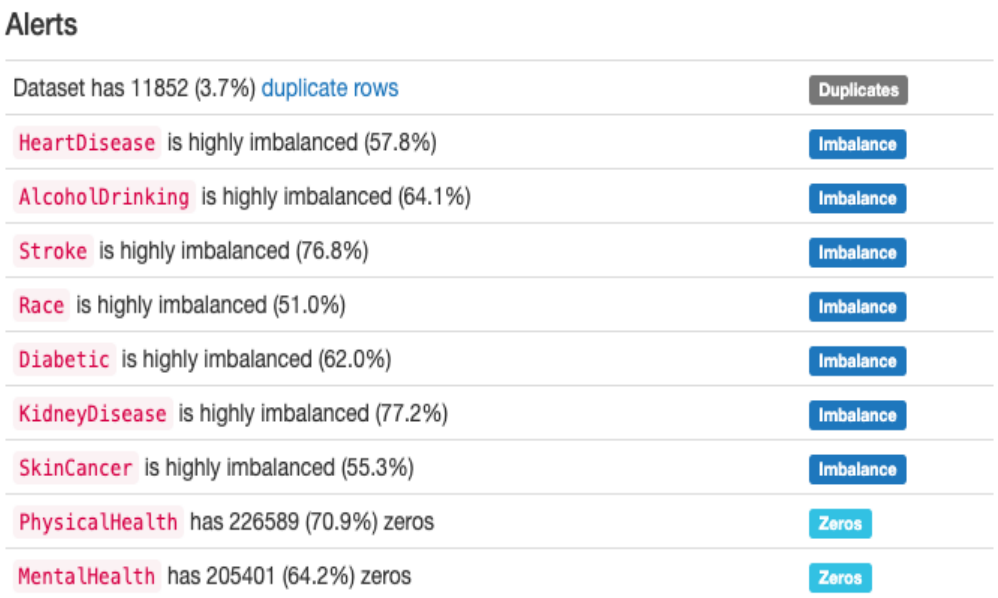

Figure 1.1 Imbalanced attributes

- 7 categorical attributes appear to be highly imbalanced.

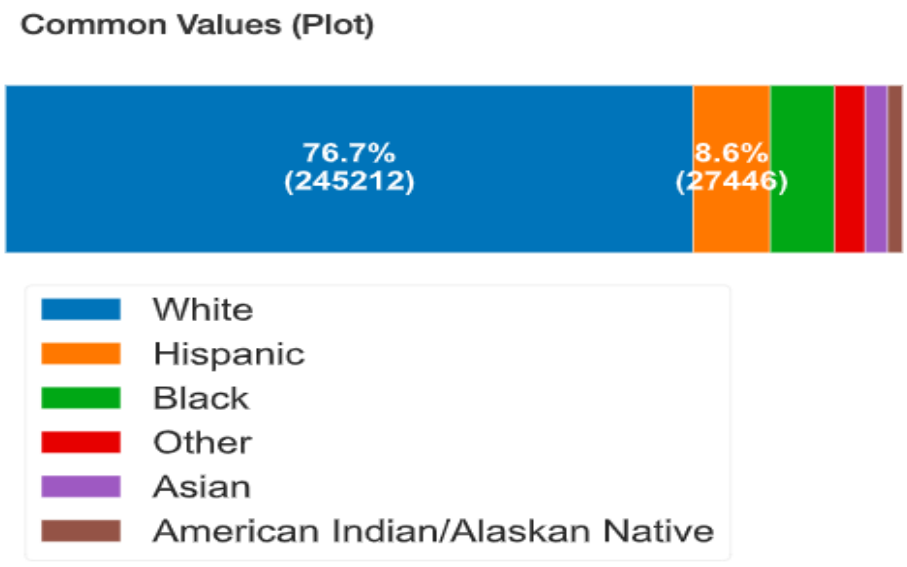

Figure 1.2 Highly Imbalanced Protected Group (Race)

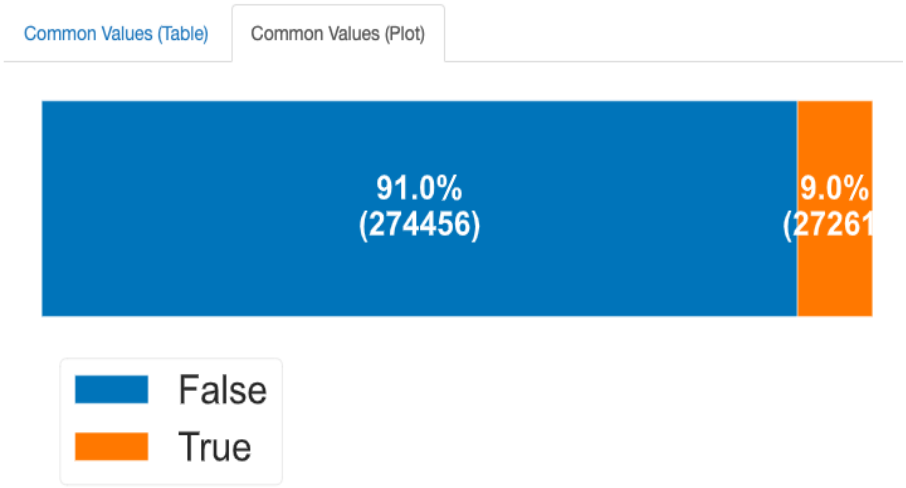

Figure 1.3 Class imbalance in *HeartDisease* variable

- The majority class ('No') of the response variable, *HeartDisease*, constitutes 91% of the data and if this is not balanced, would result in a bias in the model predictions as the algorithms will tend to classify into the class with more instances.

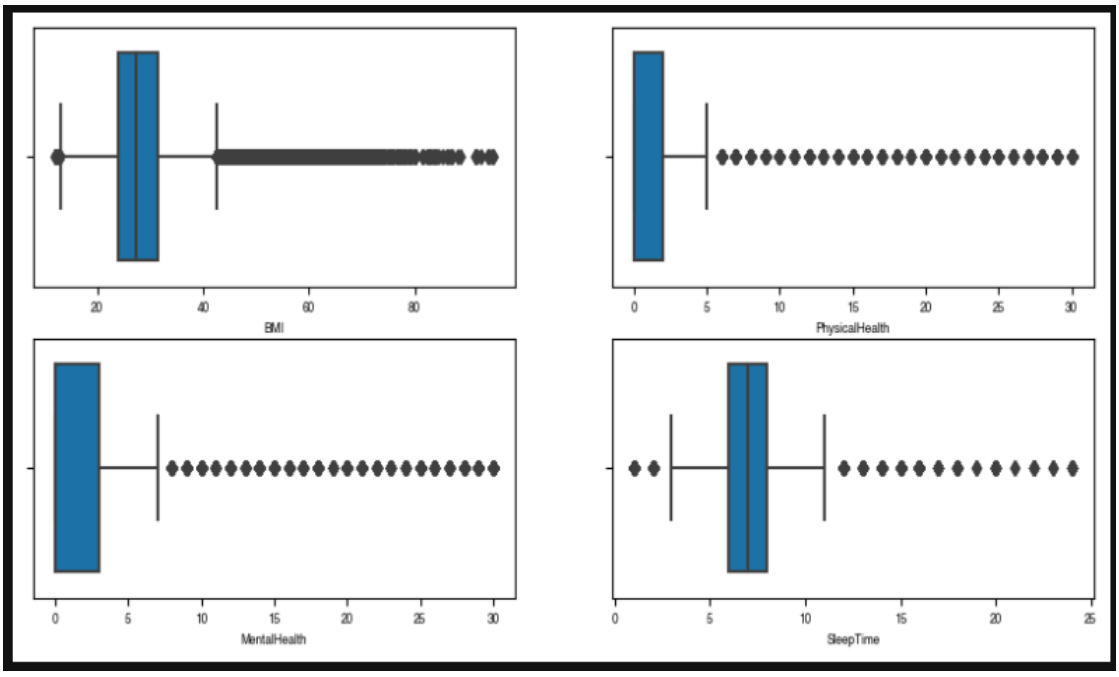

Figure 1.4 Before removing outliers in data

- The box plots expose the outliers in the numerical predictors

3.2 Data Preprocessing; Feature Engineering.

The dataset was preprocessed based on the exploratory analysis carried out in the previous segments of the code.

- The categorical attributes were sorted into nominal and ordinal variables and encoded using the **OnehotEncoder()** and **OrdinalEncoder()** respectively

- The Encoded Categorical Attributes and Numerical Predictors were concatenated to a new dataframe
- Dealing with Outliers in Numerical Predictors of the new dataframe copy:

  From figure 1.1 we saw that about 71% of the *PhysicalHealth* values and 65% of the *MentalHealth* values are 0s (from a range of 0 to 30 indicating the number of sick days within the last 30 days). I attempted this using the IQR method to remove all extreme outliers beyond the whiskers of all boxes of all numerical variables (including 'PhysicalHealth' and 'MentalHealth') and this resulted in about 28% loss of the data in the training data. This was too much). Instead, I applied a combination of Logarithmic transformation and square root transformation to further reduce the skewness in these variables without trimming the dataset further.

  I decided to apply the IQR method only to the *BMI* and *SleepTime* columns as they were less skewed. Only about 5% of data was lost this time.

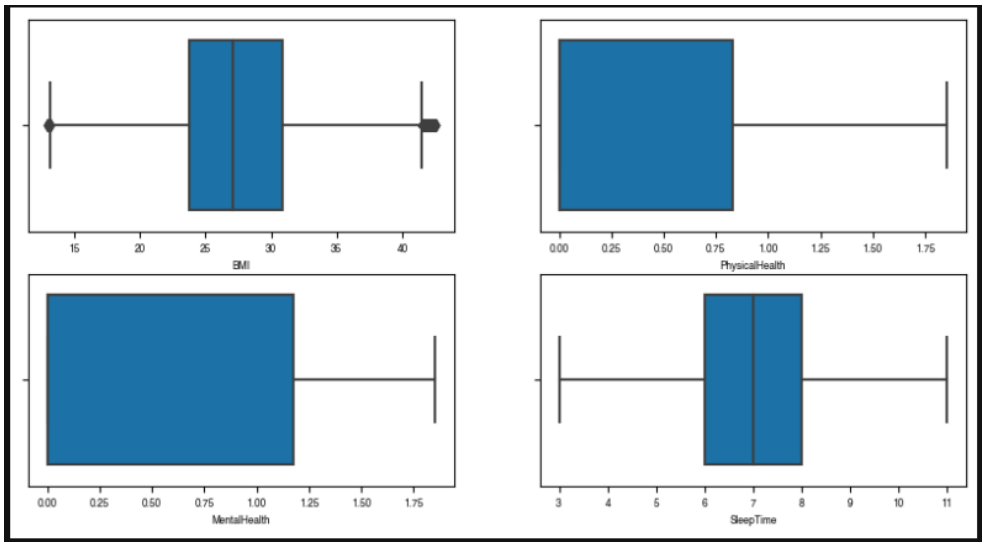

Figure 1.5 After removing outliers

- The Train data sets were scaled using the **StandardScaler()** and the test sets were transformed using the fitted scaler
- For the response variable, the minority class of both training sets (original and copy) was oversampled using SMOTE (The optimal k value was determined) and ADASYN techniques for comparison.

3.3 Model Designs, Evaluations

The four models used for this study were chosen based on the following reasons;

LightGBM and XGBoost are both gradient-boosting algorithms that are frequently used for binary classification problems like predicting heart disease diagnosis. They can handle imbalanced datasets, which is significant for this study since the dataset used is highly imbalanced across some features. These algorithms work by creating decision trees iteratively to predict the class label of the target variable. They can handle a large number of input features with high accuracy and efficiency, hence, they can handle my chosen data well.

The Random Classifier is a simple but effective ML algorithm that is suitable for classification problems like in this study. This algorithm works by randomly selecting a subset of features and creating a decision tree on the selected features. This procedure is repeated several times to create an ensemble of decision trees, which are then combined to generate the final prediction. The Random

Classifier can handle noisy and imbalanced datasets and can produce accurate predictions with little computational cost.

Lastly, the Logistic RegressionCV algorithm is a variation of logistic regression that uses cross-validation to estimate the regularization parameter, which aids in reducing overfitting. Logistic regression is a linear classification algorithm that predicts the probability of an instance belonging to a particular class based on the input features. It can handle both categorical and numerical input features and is widely used for binary classification problems like this.

The modes were trained on the training sets and evaluated on the test sets.

Hyperparameter Tunings:

- LightGBM: random_state = 50, learning_rate = 0.05, n_estimators = 100, max_depth = 5, num_leaves = 31
- XGBoost: n_estimators = 100, max_depth = 5, learning_rate = 0.05, random_state = 50
- Random Forest: n_estimators = 100, max_depth = 10, random_state = 50
- Logistic Regression with built-in Cross Validation: Cs = 10, cv = 5, class_weight = 'balanced', dual = False, fit_intercept = True, intercept_scaling = 1, l1_ratios = None, max_iter = 100, multi_class = 'auto', n_jobs = None, penalty = 'l2', random_state = 91, solver = 'liblinear', tol = 0.0001, verbose = 0

The evaluation metrics used are; Accuracy, Precision, Recall, F-score, and AUC.

4. Results; Discussions and Limitations

	Model	Data Sample	Accuracy	Precision	Recall	F1-score	AUC-ROC
0	LightGB	Initial Data	0.916274	0.584986	0.075434	0.133635	0.843966
1	LightGB	SMOTE with Outliers	0.880939	0.340012	0.415342	0.373921	0.834235
2	LightGB	SMOTE w/o Outliers	0.890664	0.351692	0.367295	0.359324	0.835198
3	LightGB	ADASYN with Outliers	0.843353	0.282167	0.537534	0.370072	0.828642
4	LightGB	ADASYN w/o Outliers	0.849565	0.280653	0.513153	0.362854	0.828601
5	XGBoost	Initial Data	0.916321	0.597464	0.068858	0.123485	0.842852
6	XGBoost	SMOTE with Outliers	0.884035	0.347135	0.402740	0.372876	0.834256
7	XGBoost	SMOTE w/o Outliers	0.888190	0.344821	0.377110	0.360244	0.835544
8	XGBoost	ADASYN with Outliers	0.842680	0.280841	0.536804	0.368758	0.827644
9	XGBoost	ADASYN w/o Outliers	0.850220	0.280633	0.508049	0.361554	0.828710
10	Random Forest	Initial Data	0.915821	0.623306	0.042009	0.078713	0.841765
11	Random Forest	SMOTE with Outliers	0.817915	0.263037	0.625571	0.370350	0.833643
12	Random Forest	SMOTE w/o Outliers	0.819593	0.256243	0.610326	0.360945	0.833216
13	Random Forest	ADASYN with Outliers	0.774230	0.232500	0.711598	0.350486	0.832870
14	Random Forest	ADASYN w/o Outliers	0.775183	0.227454	0.706517	0.344122	0.830874
15	Logistic RegressionCV	Initial Data	0.752685	0.225694	0.777169	0.349803	0.842019
16	Logistic RegressionCV	SMOTE with Outliers	0.749558	0.223615	0.778995	0.347482	0.840591
17	Logistic RegressionCV	SMOTE w/o Outliers	0.746800	0.217593	0.783274	0.340574	0.840512
18	Logistic RegressionCV	ADASYN with Outliers	0.722432	0.210506	0.815342	0.334620	0.841316
19	Logistic RegressionCV	ADASYN w/o Outliers	0.718844	0.204960	0.822536	0.328151	0.841158

Figure 2.0 Evaluation of all models in all cases

Based on the table, it appears that no single model consistently outperforms the others across all metrics. However, we can still make some observations:

- Across all models, the initial dataset has the highest accuracy, while ADASYN with outliers has the lowest accuracy (Except for the Logistic Regression).
- The precision of all models is generally low, ranging from 0.204960 to 0.623306, indicating that many false positives are being classified as positive.
- The recall of all models is also generally low, ranging from 0.042009 to 0.815342, indicating that many true positives are being classified as false negatives.
- The F1-score is generally low, ranging from 0.078713 to 0.370072, indicating a poor balance between precision and recall.
- The AUC-ROC scores are relatively consistent across all models, ranging from 0.828601 to 0.843966.

Closer observation reveals the following:

- Both Gradient Boosting variants (LightGB and XGBoost) recorded the highest accuracies (above 83%), excluding only the Random forest model on Initial Data. However, they recorded the worst Recalls (True positive rates) on the average across all cases. Hence these models cannot be recommended.
- For all cases, only the Logistic RegressionCV model appear to have the poorest predictive accuracies (less than 80% but above 70%), but at the same time in all case, it records the highest recall (above 75%). It recorded higher recall values on the data without outliers (for both the SMOTE and ADASYN variants). The only model that came close was the Random Forest (ADASYN with outliers) at 71%.

This means that the chance that a model CORRECTLY predicts / diagnoses a patient with heart disease is higher for the Logistic RegressionCV model (ADASYN without outliers). The recall (True positive rate) is a very vital metric of the model performance as it records the number of times the model correctly predicts heart disease, for this reason, I will recommend adopting the Logistic RegressionCV model (ADASYN without outliers) for the purpose of this project application even though it has lower predictive accuracy.

In summary, there is no clear winner among the models in terms of overall performance. The AUC-ROC scores indicate that all models are able to distinguish between positive and negative cases to a similar degree. However, the precision, recall, and F1-score are generally low, indicating that there is still room for improvement in the models.

5. Conclusion and Future Research

The performances across all models are generally not great and need to be improved upon as they cannot be reliable for predictions in real applications.

Future research can consider updating the dataset or using a dataset that has a more even distribution of classes within its features, especially in protected groups, to ensure a less biased model. More robust state-of-the-art models, preprocessing steps, and/or hyperparameter tunings can be explored to achieve state-of-the-art performance.

6. References

Ghasemi, A., Hormozan, S., Zahedi, E., & Yazdinejad, M. (2023). Coronary Artery Disease Diagnosis with Deep Neural Network, Lightgbm and XGBoost. International Journal of Hospital Research, 11(4). http://ijhr.iums.ac.ir/article_163790.html

Karthick, K., Aruna, S. K., Samikannu, R., Kuppusamy, R., Teekaraman, Y., & Thelkar, A. R. (2022). Implementation of a Heart Disease Risk Prediction Model Using Machine Learning. Computational and Mathematical Methods in Medicine, 2022, e6517716. https://doi.org/10.1155/2022/6517716

Song, Y., Jiao, X., Qiao, Y., Liu, X., Qiang, Y., Liu, Z., & Zhang, L. (2019). Prediction of Double-High Biochemical Indicators Based on LightGBM and XGBoost. Proceedings of the 2019 International Conference on Artificial Intelligence and Computer Science, 189–193. https://doi.org/10.1145/3349341.3349400

Song, Y., Jiao, X., Yang, S., Zhang, S., Qiao, Y., Liu, Z., & Zhang, L. (2019). Combining Multiple Factors of LightGBM and XGBoost Algorithms to Predict the Morbidity of Double-High Disease. In R. Mao, H. Wang, X. Xie, & Z. Lu (Eds.), Data Science (pp. 635–644). Springer. https://doi.org/10.1007/978-981-15-0121-0_50