Heart Disease Prediction Model

Team AI Big BabaA colorful circles and circles on a black background

Description automatically generated

**Team Members**

**Ahmad Abdallah, Mohamed Ahmed, Mostafa Mohamed, Youssef Reda, Mohamed Bassam**

**Our Goal is to**

Develop an accurate heart disease prediction model using the XGBoost classifier.

**Made by: Mohamed Ahmed (21511546)**

**Mohamed Bassam (21511646)**

**Youssef Reda (22513303)**

[1.1] Overview:

In the development of our heart disease prediction model, understanding the data is an important task, this section will provide an overview of the dataset and its characteristics.

[1.2] Source:

The dataset used in the model is from:

<https://www.kaggle.com/datasets/fedesoriano/heart-failure-prediction>

[1.3] Data Size and Characteristics:

About 918 rows with 12 columns

[1.4] Data Preprocessing:

Data preprocessing involved a thorough check for null values, ensuring data integrity and consistency, please refer to the accompanying notebook named ‘analyses.

**Heart disease prediction report**

The investigation into heart disease prediction required a thorough examination of a dataset with 12 columns, which included critical parameters such as Age, Sex, ChestPainType, RestingBP, Cholesterol, FastingBS, RestingECG, MaxHR, ExerciseAngina, Oldpeak, ST\_Slope, and heart disease. To help our study, we used essential Python libraries such as:

1. Pandas
2. NumPy
3. Matplotlib
4. Seaborn
5. Scikit-learn, with a focus on the labelencoder, standardscaler, and train\_test\_split modules.

we make an analysis on six column we found that:

m: 725 (m = male)

f: 193 (f = female)

as we know the number of participants is 918

we found that 508 have a disease and 410 have not the disease.

data shape (918,12)

the more get elder the more probability of get the disease.

all data have not null values and datatype (int64, object, float64)

when we used visualization to express about the disease and six

we find that the disease in male is greater than female.

To use model, we cannot use char or string and so that we use labelencoder to convert values to number.

We used the train\_test\_split function to divide our dataset into a 30% training set and a 70% testing set before training the model. The introduction of a random\_state option meant that our results were reproducible, which is necessary when developing and evaluating machine learning models.

**Here is the report of the modeling.**

In this research, we used the accurate XGBoost classifier to predict the risk of heart disease based on a number of patient characteristics. To improve the model's performance, the procedure included extensive data investigation, preprocessing, model development, assessment, and hyperparameter adjustment.

**And I've separated them into multiple things, which will be listed below.**

1. Model Definition and Training:

The first stage was to define the XGBoost classifier model's initial hyperparameters, which included the number of estimators (trees) and maximum depth. The model was created with 100 estimators and a maximum depth of three. After dividing the dataset into training and testing sets, the model was trained on the training set using the fit approach.

1. Model Evaluation:

The accuracy of the model was evaluated using both the training and testing sets. The accuracy score on the test set was roughly 84.42%, suggesting quite strong performance. To explore deeper into the model's efficacy, a confusion matrix and classification report were created. The confusion matrix revealed that the model struck a balance between accuracy and recall for both classes (0: No Heart Disease; 1: Heart Disease). The classification report detailed the accuracy, recall, and F1-score for each class.

1. Cross-Validation Scores:

A 5-fold cross-validation was used to assess the model's performance across several data subsets. The cross-validation results showed consistency, with average accuracy ranging from 80 to 84%.

1. Hyperparameter Tuning:

GridSearchCV was used to tune the model's hyperparameters in order to improve its performance. The hyperparameters under consideration were the number of estimators and the tree's maximum depth. The optimal hyperparameters were found to be a maximum depth of 5 and 50 estimators.

1. Retraining with Tuned Hyperparameters:

The model was again retrained using the found best hyperparameters, with the goal of capturing more complicated correlations in the data.

1. Feature Importance Analysis:

A feature significance analysis was performed to get insight into which features contribute the most to the model's predictions. The research found that characteristics such as 'ST\_Slope,' 'ExerciseAngina,' and 'ChestPainType' were the most influential in predicting the risk of heart disease.

1. Receiver Operating Characteristic (ROC) Curve:

The ROC curve was used to determine the model's ability to differentiate between positive and negative situations. The area under the curve (AUC) was determined to evaluate the model's overall performance. The AUC of about 0.91 indicated a relatively good degree of discriminating skill.  
  
**\*Note: The higher the AUC, the better the model's performance is.**

In our situation, an AUC of 0.91 is pretty good, indicating that our XGBoost classifier is doing well in distinguishing between persons with and without heart disease. It's a strong indication of the model's performance in the classification assignment you've given it.

**To Sum up everything again:**

1. **Model Building and Training:**
   * Created an XGBoost classifier with default hyperparameters.
   * Split the dataset into training and testing sets.
   * Trained the model on the training set and evaluated its accuracy on the test set.
2. **Model Evaluation:**
   * Calculated and printed accuracy, confusion matrix, and classification report on the test set.
   * Conducted cross-validation to assess the model's performance across different subsets of the data.
3. **Hyperparameter Tuning:**
   * Defined a parameter grid for hyperparameter tuning using **GridSearchCV**.
   * Conducted grid search and identified the best hyperparameters.
4. **Retraining the Model with Tuned Hyperparameters:**
   * Re-trained the XGBoost model using the best hyperparameters.
5. **Feature Importance:**
   * Extracted and visualized feature importance using the **feature\_importances\_** attribute.
6. **Receiver Operating Characteristic (ROC) Curve:**
   * Plotted the ROC curve and calculated the area under the curve (AUC) to assess the model's performance.

**Some Definition if you didn’t know them or in what I used it for.**

1. **XGBoost:**

XGBoost stands for eXtreme Gradient Boosting. It is a powerful and efficient machine learning algorithm.

1. **ROC Curve:**

An ROC curve (receiver operating characteristic curve) is a graph showing the performance of a classification model at all classification thresholds.

1. **AUC:**

The Area Under the Curve (AUC) is the measure of the ability of a binary classifier to distinguish between classes and is used as a summary of the ROC curve. The higher the AUC, the better the model's performance at distinguishing between the positive and negative classes.

1. **Cross-Validation:**

Cross-validation is a resampling approach that determines how well a prediction model generalizes to a different dataset. The dataset is divided into numerous subsets, the model is trained on part of them, and the remaining subsets are tested.

1. **Hyperparameter Tuning:**

Hyperparameter tuning is the systematic search for the optimal hyperparameter values for a machine learning model. Hyperparameters are parameters that are established after training and have an important impact on model performance.

Team Information:

1. Ahmad Abdallah [22510183]

2. Mohamed Ahmed [21511546]

3. Mostafa Mohamed [22511123]

4. Youssef Reda [22513303]

5. Mohamed Bassam [21511646]

Contributions:

Mohamed Ahmed for the dataset and analytics

Mohamed Bassam and Ahmad Abdallah for the Model and training

Youssef Reda and Mostafa Mohamed and Mohamed Bassam and Ahmad Abdallah and Mohamed Ahmed for the report