Machine Learning Project Report

Title : Heart Disease Prediction
Author : Shayna Nicholas Tuscano
Date : 05/01/2024

1. Introduction

1.1 Background

There are different types of coronary heart disease, the majority of individuals only learn they have the disease following symptoms such as chest pain, a heart attack, or sudden cardiac arrest. Through this project we will try to accurately predict heart disease in the population so that preventive measures can be taken.

1.2 Problem Statement

Heart disease stands as one of the leading global causes of mortality. Relying solely on traditional risk factors and clinical assessments may result in delays in accessing essential medical attention, underscoring the need for more advanced and timely diagnostic approaches.

1.3 Objectives

The primary goal of this project is to create a machine learning model capable of accurately predicting the likelihood of heart disease in individuals, facilitating early intervention and personalised healthcare. The integration of machine learning models into healthcare sector can provide an opportunity to leverage historical data to make predictions about future instances.

2. Data Exploration and Analysis

2.1 Data Collection

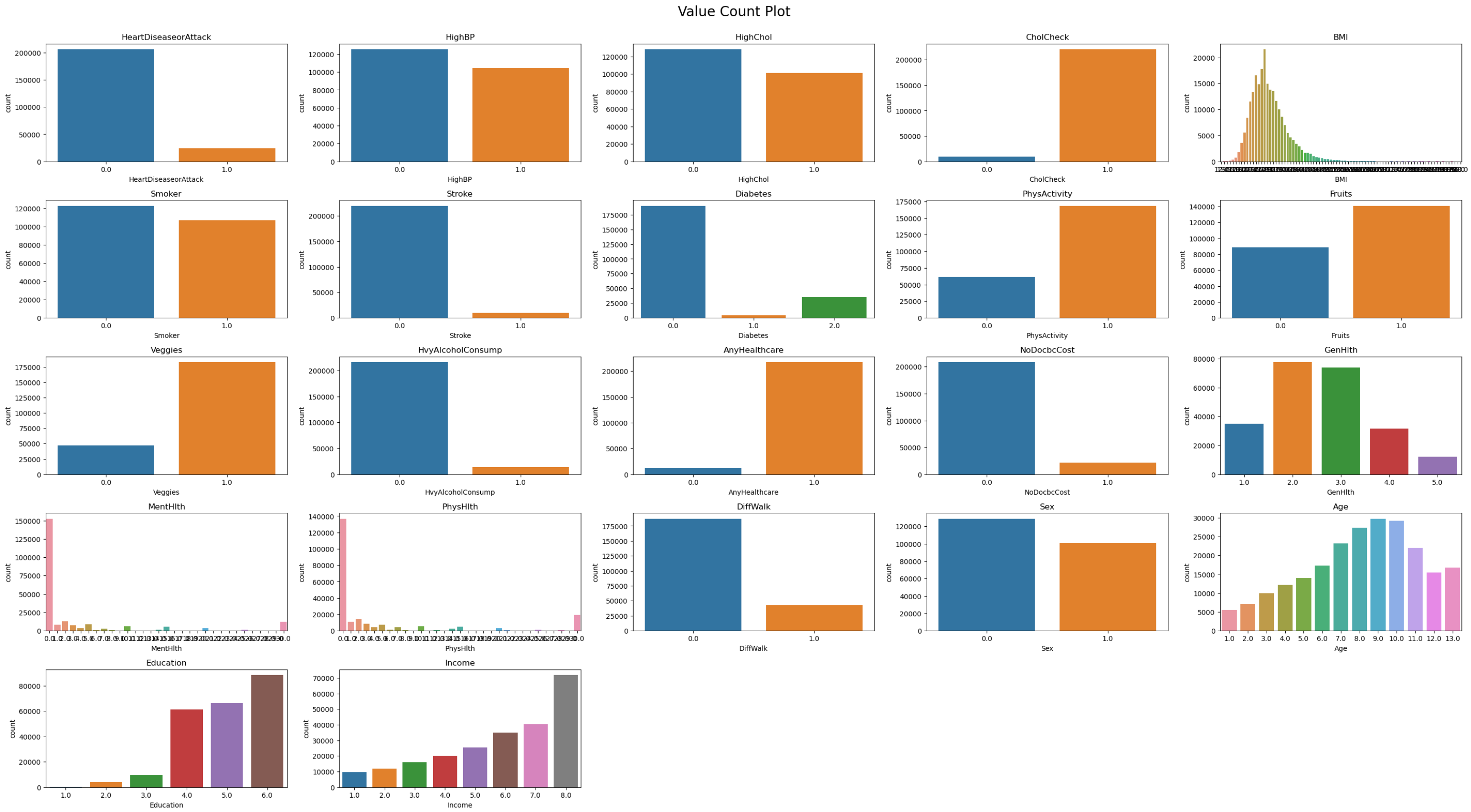
In this project, the dataset utilised is the "Heart Disease Health Indicators Dataset," sourced from Kaggle. This dataset comprises 253,680 survey responses derived from the BRFSS 2015 dataset, which is already available on Kaggle. The Behavioural Risk Factor Surveillance System (BRFSS) conducts an annual health-related telephone survey administered by the CDC. Each year, this survey gathers responses from over 400,000 Americans, covering various aspects such as health-related risk behaviours, chronic health conditions, and the utilisation of preventative services.

2.2 Data Overview

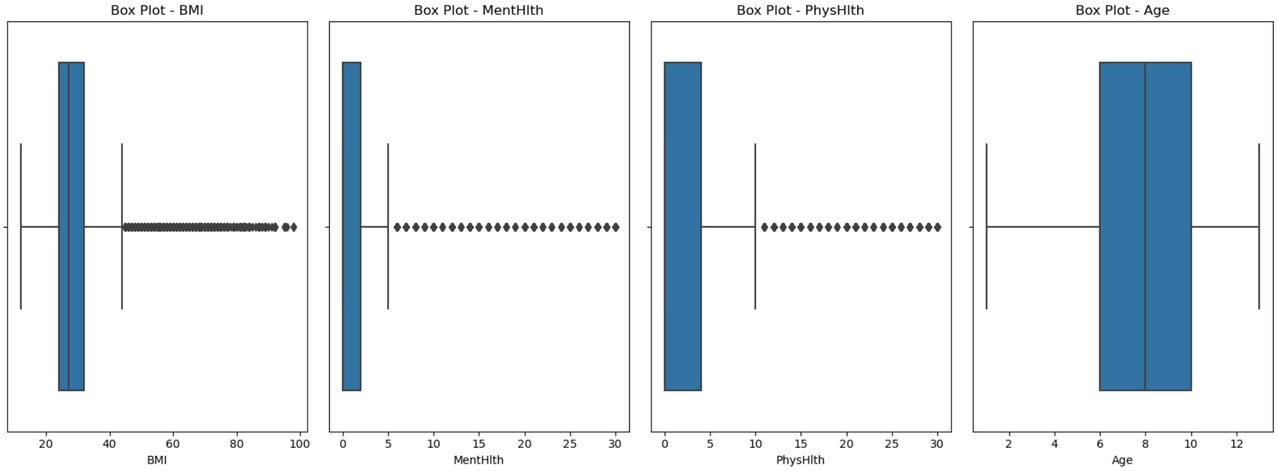
This dataset comprises 253,680 records with 22 columns, encompassing 21 independent features and 1 target variable. The independent features encompass diverse aspects such as health, lifestyle, demographics, and medical conditions. The features include indicators like 'HighBP' for high blood pressure presence, 'CholCheck' for cholesterol check status, 'BMI' for Body Mass Index, 'Smoker' for smoking status, 'Stroke' for history of stroke, 'Diabetes' for diabetes presence, 'PhysActivity' for physical activity level, 'Fruits' and 'Veggies' for fruit and vegetable consumption, 'HvyAlcoholConsump' for heavy alcohol consumption, 'AnyHealthcare' for access to any healthcare, 'NoDocbcCost' for lack of doctor visit due to cost, 'GenHlth', 'MentHlth', and 'PhysHlth' for general health, mental health, and physical health perception, respectively. Additionally, features like 'DiffWalk', 'Sex', 'Age', 'Education', and 'Income' are included. The target variable, 'HeartDiseaseorAttack', denotes the presence or absence of heart disease or heart attack.

2.3 Exploratory Data Analysis (EDA)

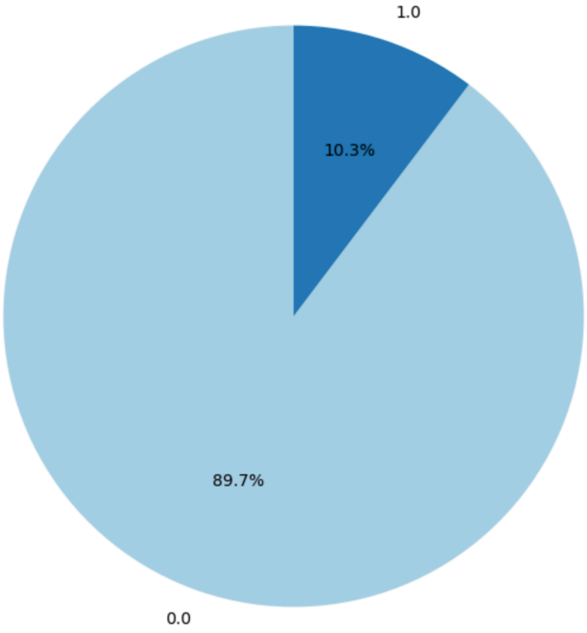
Following an exploratory data analysis (EDA) on the dataset, it was observed that it encompasses both categorical and numerical features. The categorical features include 'HighBP', 'HighChol', 'CholCheck', 'Smoker', 'Stroke', 'Diabetes', 'PhysActivity', 'Fruits', 'Veggies', 'HvyAlcoholConsump', 'AnyHealthcare', 'NoDocbcCost', 'GenHlth', 'DiffWalk', and 'Sex', while numerical features include 'BMI', 'Age', 'Education', and 'Income'.



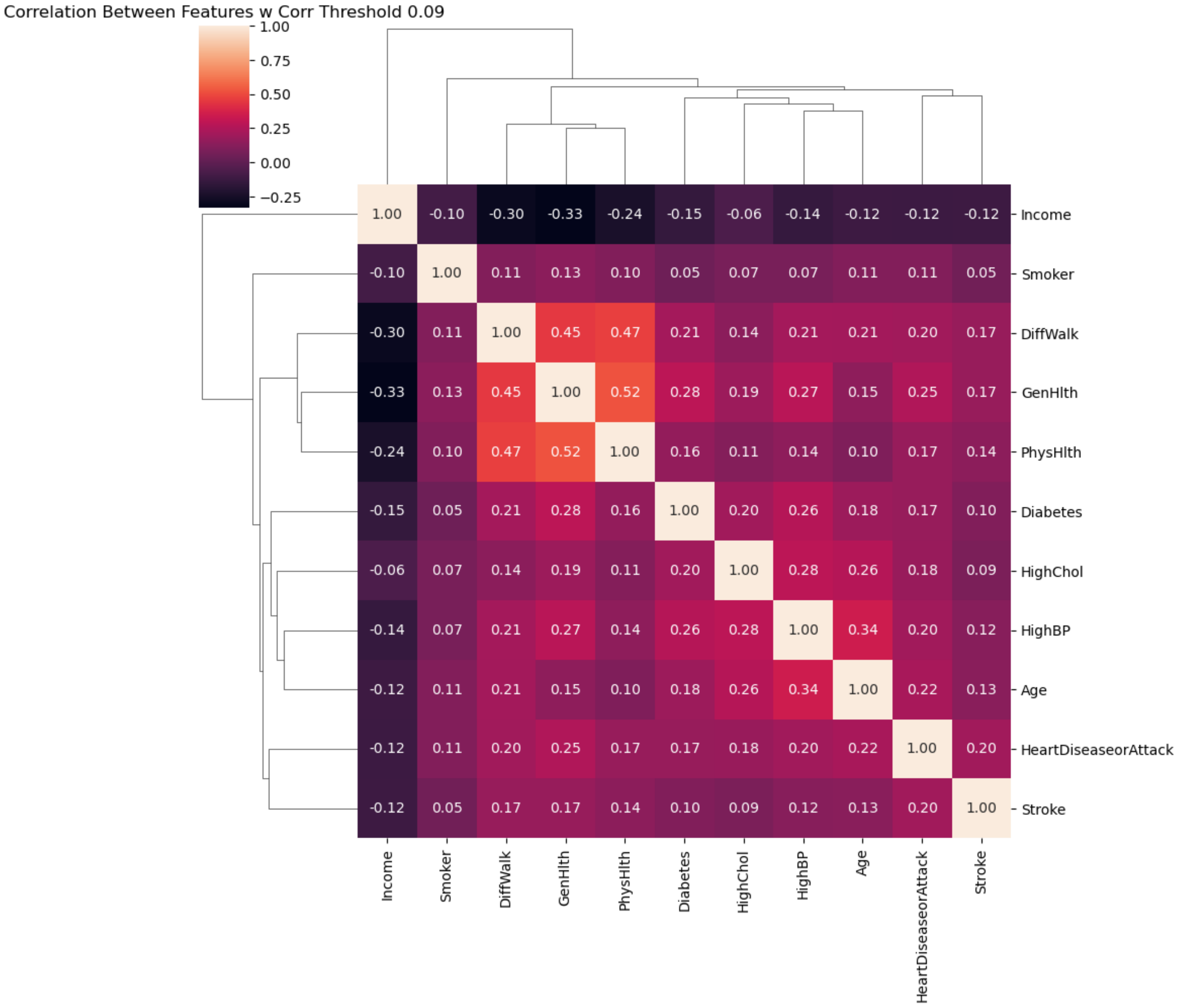
Boxplot analysis revealed outliers in the numerical features 'BMI', 'MentHlth', and 'PhysHlth'.



Moreover, the target variable, 'HeartDiseaseorAttack', is categorical, indicating the presence (1) or absence (0) of heart disease. Within the training set of 253,680 individuals, only 10.3% were found to have heart disease.



To refine the dataset, features with very low correlation to the target variable were identified and subsequently removed.



3. Data Preprocessing

3.1 Handling Missing Data

After throughly analysing the dataset no missing data was found.

3.2 Removing Duplicate Records

A total of 23,899 duplicate records were identified in the dataset and subsequently removed for data cleanliness and to prevent any potential bias in the analysis.

3.3 Encoding Categorical Variables

The utilised dataset includes categorical columns that have already undergone label encoding, where the categorical values were replaced with corresponding numerical labels. The accompanying information provided by the author helped establish the mapping between the original categorical values and their label-encoded representations. For instance:
'HeartDiseaseorAttack' is encoded as 0 for "no heart disease" and 1 for "heart disease."
'HighBP' is encoded as 0 for "No high blood pressure" and 1 for "high blood pressure."
'HighChol' is encoded as 0 for "No" and 1 for "high" cholesterol.
'Smoker' is encoded as 0 for "No" and 1 for "Yes."
'Stroke' is encoded as 0 for "No" and 1 for "Yes."
'Diabetes' is encoded as 0 for "no diabetes or only during pregnancy," 1 for "pre-diabetes or borderline diabetes," and 2 for "yes, diabetes."
'GenHlth' is an ordinal variable, with 1 representing "Excellent" health and 5 representing "Poor" health.
'DiffWalk' is encoded as 0 for "No" and 1 for "Yes."
This information aids in interpreting and understanding the encoded values within the categorical columns of the dataset.

3.4 Handling Imbalanced Data

Given the substantial class imbalance in the dataset, a common strategy is to employ synthetic sample generation techniques to address this issue in machine learning datasets. One widely adopted approach for generating synthetic samples is SMOTE (Synthetic Minority Over-sampling Technique). SMOTE functions by creating artificial instances of the minority class through interpolation between existing minority class instances.

To ensure the effectiveness of these techniques, it is crucial to apply data imbalance adjustments, like oversampling or under-sampling, before the dataset is split into training and testing sets. This precautionary step helps prevent synthetic samples from influencing the test set, preserving the integrity of the evaluation process. By proactively addressing class imbalance through methods such as SMOTE prior to model training, the goal is to enhance the model's capacity to learn and accurately predict instances from the minority class.

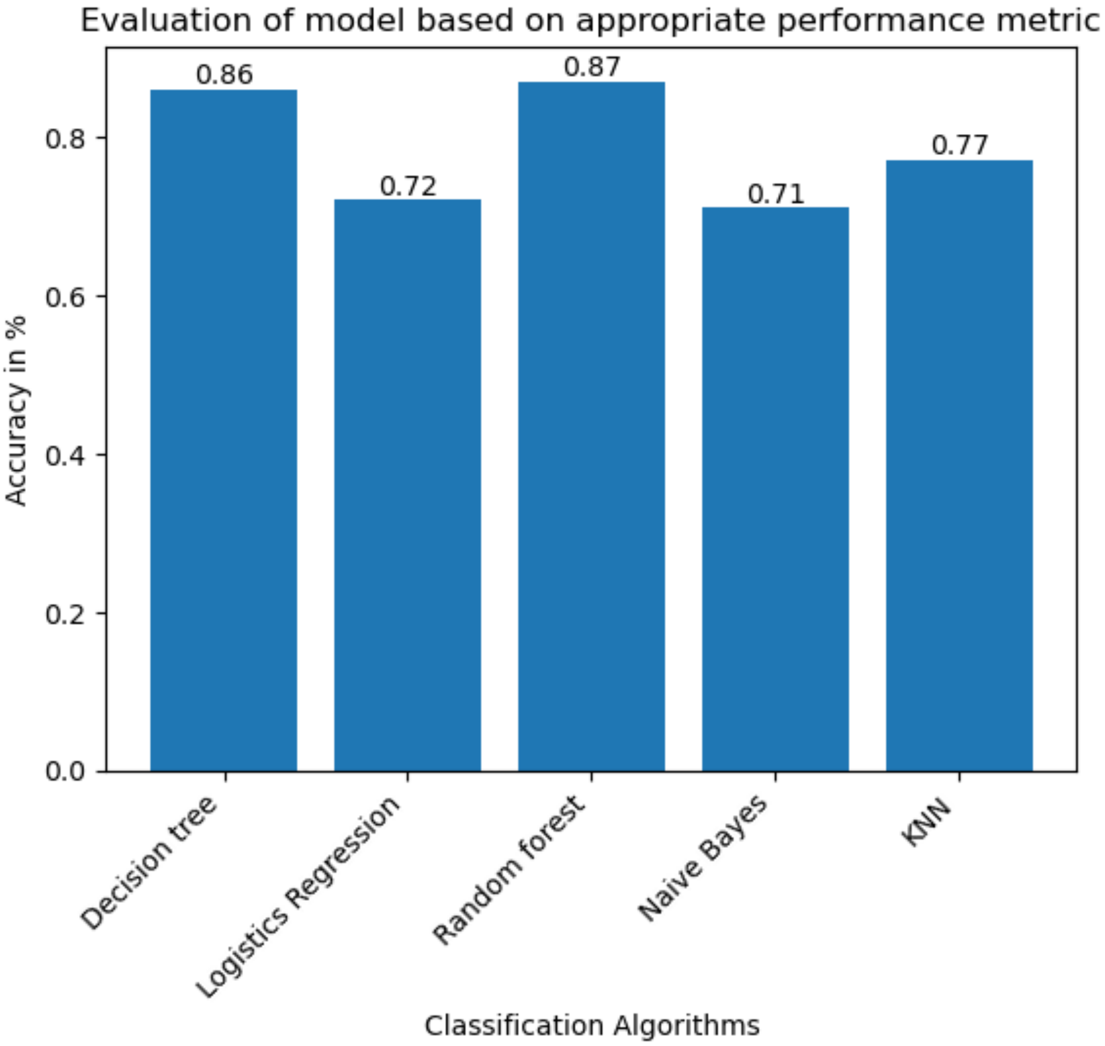
4. Model Development

4.1 Model Selection

In tackling the classification task at hand, we have employed various machine learning algorithms. Specifically, we have explored the efficacy of Decision Trees, Logistic Regression, Random Forest, Naive Bayes, and K-Nearest Neighbors (KNN) in addressing the classification challenge.

4.2 Model Training

The process of model training began by splitting the dataset into features and labels, followed by an 80% training set and a 20% testing set partition. To address the imbalanced data, the Synthetic Minority Over-sampling Technique (SMOTE) was applied to the training dataset. Decision Tree, Logistic Regression, Random Forest, Naive Bayes, and KNN were then trained and cross-validated using k-fold cross-validation. Among these models, Random Forest and Decision Tree exhibited superior accuracy and recall compared to others, prompting their selection for further consideration.



Following the model selection, hyperparameter tuning was conducted using grid search CV. For Random Forest, parameters such as n_estimators (20, 60, 100, 120), max_features (0.2, 0.6, 1.0), max_depth (2, 8, None), and max_samples (0.5, 0.75, 1.0) were explored. For Decision Tree, the hyperparameters included max_depth (3, 5, 7, 10), min_samples_split (2, 5, 10), min_samples_leaf (1, 2, 4), and max_features ('sqrt', 'log2').
It is noteworthy that hyperparameter tuning can significantly impact the model's performance and its ability to make accurate predictions on data. This process aims to optimize the chosen models by fine-tuning their parameters to achieve the best possible results..

4.3 Model Evaluation

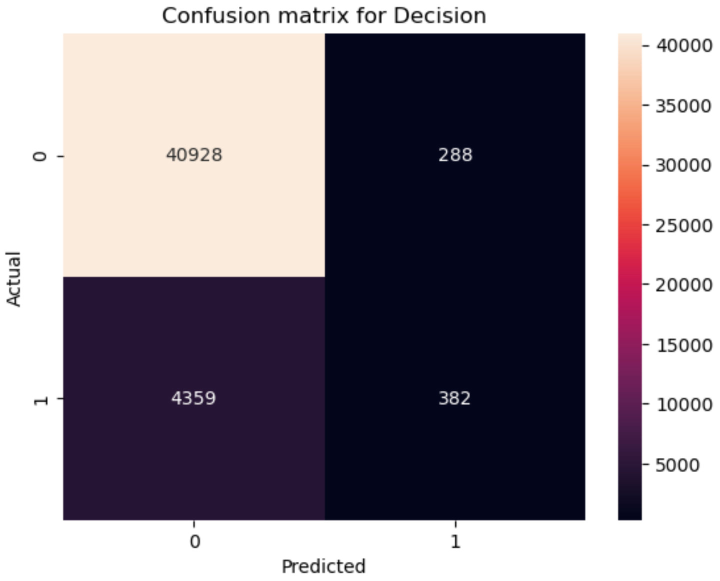
Following hyperparameter tuning, the performance of both models was assessed on the test data. This evaluation involved analyzing various metrics such as accuracy, precision, recall, and F1 score to gauge how well the optimized models generalize to new, unseen data. The goal was to ensure that the selected models not only perform well on the training data but also demonstrate robustness and effectiveness when applied to the independent test set.

Decision Tree performance :

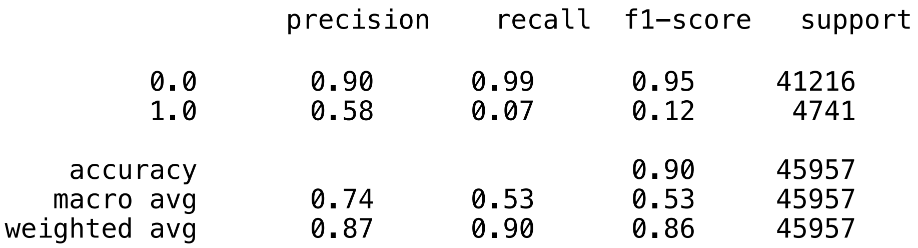
```
              precision    recall  f1-score   support

         0.0       0.90      0.99      0.95     41216
         1.0       0.57      0.08      0.14      4741

    accuracy                           0.90     45957
   macro avg       0.74      0.54      0.54     45957
weighted avg       0.87      0.90      0.86     45957
```



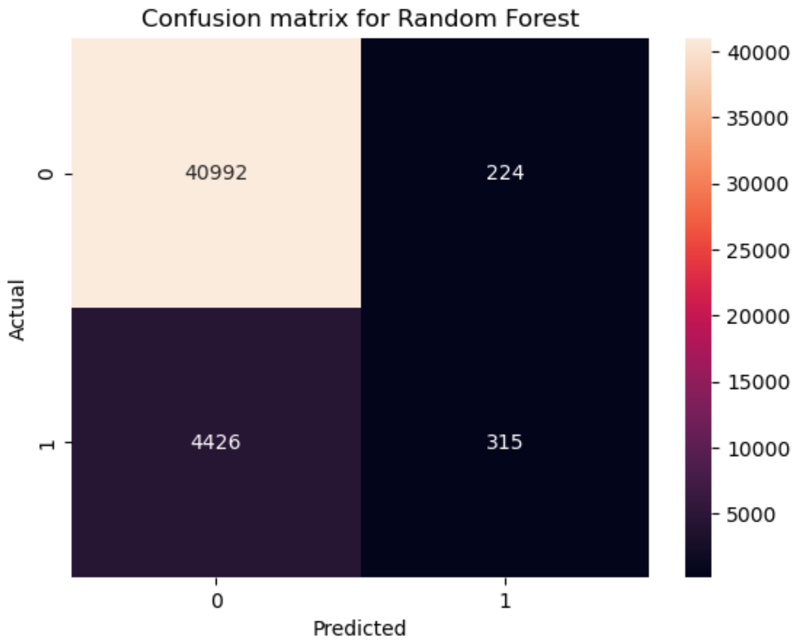
Random Forest performance:



```
              precision    recall  f1-score   support

         0.0       0.90      0.99      0.95     41216
         1.0       0.58      0.07      0.12      4741

    accuracy                           0.90     45957
   macro avg       0.74      0.53      0.53     45957
weighted avg       0.87      0.90      0.86     45957
```



4.4 Model Performance

Post-hyperparameter tuning, there was a notable improvement in the accuracy of both models when evaluated on the test data. This enhancement underscores the effectiveness of fine-tuning model parameters to optimise their performance, resulting in better predictive accuracy on previously unseen data.

5. Conclusion

In this project, both models exhibited similar performance in terms of accuracy and recall. However, the Decision Tree algorithm emerged as more suitable for the task at hand. This preference was driven by the consideration that training a single decision tree is generally faster than training a Random Forest, particularly when dealing with large datasets. The efficiency of the Decision Tree algorithm aligns well with the project's requirements and computational considerations.

6. References

https://www.kaggle.com/code/alexteboul/heart-disease-health-indicators-dataset-notebook
https://www.kaggle.com/datasets/cdc/behavioral-risk-factor-surveillance-system
https://www.kaggle.com/datasets/alexteboul/heart-disease-health-indicators-dataset

7. Appendices

The comprehensive Python code for this project, including all the necessary implementations and configurations, is enclosed in a separate file for your review and reference.