Utilizing CDC data to predict heart Disease in the US

Akhzar Farhan, DSBA, *akhzarfarhan@iisc.ac.in*
Arun Kumar A, DSBA, *aruna@iisc.ac.in*
Gajam Venu Gopal, AI, *gajamvenu@iisc.ac.in*
Venkata Ajay Kolla, AI, *ajaykolla@iisc.ac.in*

1. Abstract

Heart disease is one of the leading causes of mortality worldwide, resulting in high medical costs and significant loss of life. For healthcare providers and insurers, this leads to increased hospitalization costs, resource strain, and economic burdens. This project aims to develop a machine learning model to predict the likelihood of heart disease in individuals based on various health and lifestyle factors. Using a dataset that includes features such as blood pressure, cholesterol, smoking, diabetes status, obesity (high BMI), physical inactivity, and excessive alcohol consumption, the project applies several machine learning algorithms, including logistic regression,random forests, XGboost, DNN, etc. The goal is to identify the most effective model for predicting heart disease risk with high accuracy. The model is trained, validated, and tested on the dataset, and performance metrics are evaluated to assess its effectiveness. The findings highlight the importance of specific features in heart disease prediction and suggest that machine learning can be a valuable tool for early diagnosis and preventive healthcare. This approach could potentially assist healthcare professionals in making informed decisions, improving patient outcomes, and reducing the burden of heart disease.

2. Introduction

Heart disease remains a critical public health challenge in the United States, representing a significant threat to population health and national healthcare systems. According to the Centers for Disease Control and Prevention (CDC), heart disease is the leading cause of death in the United States, accounting for approximately 695,000 deaths annually and affecting nearly 20% of adults. The economic impact is equally staggering, with heart disease and stroke costing the United States healthcare system an estimated $235 billion each year in direct medical costs and lost productivity.

The complexity of heart disease prevention and early detection has long challenged healthcare professionals, as the condition results from a multifaceted interplay of genetic, lifestyle, and environmental factors. Traditional diagnostic approaches often rely on retrospective analysis and reactive medical interventions, which can be both costly and less effective in preventing adverse health outcomes.

Machine learning represents a transformative approach to addressing these challenges, offering the potential to develop predictive models that can identify individuals at high risk of heart disease before critical symptoms emerge. By leveraging advanced computational techniques and comprehensive health datasets, machine learning algorithms can analyze complex, interconnected health parameters to generate nuanced risk assessments with unprecedented accuracy and efficiency.

This project aims to develop and validate a machine learning model capable of predicting heart disease risk by integrating multiple health and lifestyle indicators. By examining key factors such as blood pressure, cholesterol levels, smoking status, diabetes, obesity, physical activity, and alcohol consumption, the study seeks to create a robust predictive framework that can support healthcare providers in making more informed, proactive clinical decisions.

Through a comprehensive approach involving multiple machine learning algorithms, including logistic regression, random forests, and XGBoost, this research will systematically evaluate predictive performance and identify the most effective methodological approach for heart disease risk assessment. The ultimate goal is to develop a tool that can provide timely, accurate risk stratification, potentially enabling earlier interventions and more targeted preventive care.

3. Methodology

3.1. Dataset

The dataset used in this project is sourced from Kaggle and is titled **Indicators of Heart Disease (2022 UPDATE)**. It was collected and made publicly available by Kamil Pytlak, and the dataset aims to explore various health indicators related to heart disease prediction.

The CSV file considered for the project is **2022/heart_2022_with_nans.csv.** The dataset consists of **445132 rows (246,022 rows without NANs)** and **40 columns**, each representing specific attributes about individuals' health conditions, which can potentially be used to predict the likelihood of heart disease.

The dataset includes the following columns:

1. **Sex**: The gender of the individual (binary: male and female).
2. **GeneralHealth**: Self-reported general health status (categorical: 1 = Excellent, 2 = Very good, 3 = Good, 4 = Fair, 5 = Poor).
3. **PhysicalHealthDays**: Number of days in the past month the individual felt physically unhealthy (numerical).
4. **MentalHealthDays**: Number of days in the past month the individual felt mentally unhealthy (numerical).
5. **LastCheckupTime**: Time since the last checkup (categorical: Within past year (anytime less than 12 months ago), Within past 2 years (1 year but less than 2 years ago), Within past 5 years (2 years but less than 5 years ago), 5 or more years ago).
6. **PhysicalActivities**: Whether the individual participates in physical activities (binary: yes, no).
7. **SleepHours**: Average number of hours of sleep per day (numerical).
8. **RemovedTeeth**: Whether the individual has had teeth removed (categorical: 'None of them', '1 to 5', '6 or more, but not all', 'All')
9. **HadHeartAttack**: Whether the individual has had a heart attack (binary: yes, no).
10. **HadAngina**: Whether the individual has had angina (binary: yes, no).
11. **HadStroke**: Whether the individual has had a stroke (binary: yes, no).
12. **HadAsthma**: Whether the individual has asthma (binary: yes, no).
13. **HadSkinCancer**: Whether the individual has had skin cancer (binary: yes, no).
14. **HadCOPD**: Whether the individual has Chronic Obstructive Pulmonary Disease (COPD) (binary: yes, no).
15. **HadDepressiveDisorder**: Whether the individual has had a depressive disorder (binary: yes, no).
16. **HadKidneyDisease**: Whether the individual has kidney disease (binary: yes, no).
17. **HadArthritis**: Whether the individual has arthritis (binary: yes, no).
18. **HadDiabetes**: Whether the individual has diabetes (binary: yes, no).
19. **DeafOrHardOfHearing**: Whether the individual is deaf or has hearing difficulty (binary: yes, no).

20. **BlindOrVisionDifficulty**: Whether the individual is blind or has vision difficulty (binary: yes, no).
21. **DifficultyConcentrating**: Whether the individual has difficulty concentrating (binary: yes, no).
22. **DifficultyWalking**: Whether the individual has difficulty walking (binary: yes, no).
23. **DifficultyDressingBathing**: Whether the individual has difficulty dressing or bathing (binary: yes, no).
24. **DifficultyErrands**: Whether the individual has difficulty running errands (binary: yes, no).
25. **SmokerStatus**: Whether the individual is a current smoker (categorical: 'Never smoked', 'Former smoker', 'Current smoker - now smokes some days', 'Current smoker - now smokes every day').
26. **ECigaretteUsage**: Whether the individual uses e-cigarettes (categorical: 'Never used e-cigarettes in my entire life', 'Not at all (right now)','Use them some days', 'Use them every day').
27. **ChestScan**: Whether the individual has had a chest scan (binary: yes, no).
28. **RaceEthnicityCategory**: Race/ethnicity category of the individual (categorical: 'White only, Non-Hispanic', 'Black only, Non-Hispanic', 'Other race only, Non-Hispanic', 'Multiracial, Non-Hispanic', 'Hispanic').
29. **AgeCategory**: Age group category (categorical: e.g., 18-29, 30-39, etc.).
30. **HeightInMeters**: Height of the individual in meters (numerical).
31. **WeightInKilograms**: Weight of the individual in kilograms (numerical).
32. **BMI**: Body Mass Index (numerical).
33. **AlcoholDrinkers**: Whether the individual drinks alcohol (binary: yes, no).
34. **HIVTesting**: Whether the individual has undergone HIV testing (binary: yes, no).
35. **FluVaxLast12**: Whether the individual received a flu vaccination in the last 12 months (binary: yes, no).
36. **PneumoVaxEver**: Whether the individual has ever had a pneumococcal vaccination (binary: yes, no).
37. **TetanusLast10Tdap**: Whether the individual received a tetanus or Tdap vaccination in the past 10 years (categorical: 'Yes, received Tdap', 'No, did not receive any tetanus shot in the past 10 years', 'Yes, received tetanus shot but not sure what type', 'Yes, received tetanus shot, but not Tdap').
38. **HighRiskLastYear**: Whether the individual was considered high risk for health complications in the last year (binary: yes, no).
39. **CovidPos**: Whether the individual tested positive for COVID-19 (binary: yes, no).
40. **State**: The state in which the individual resides (categorical).

3.2. Data preprocessing

3.2.1. Imputation techniques for missing values

In our heart attack prediction study using a medical dataset, we encountered missing crucial values in several records. After evaluating the extent and distribution of missing data, we decided to handle them by removing rows with missing values. This approach was justified as the proportion of missing data was almost the same as the overall dataset, by removing them we ensured that the meaning of the dataset remained intact. Moreover, the missing values were distributed randomly, minimizing the risk of introducing bias through this process. By eliminating incomplete records, we ensured that the dataset used for training and evaluating the predictive model was complete, thereby avoiding potential inaccuracies that could arise from imputation assumptions or distortions in the data. This decision supported the development of a reliable and robust predictive model for heart attack risk.

3.2.2. Under-sampling of majority class

In our study, we observed a significant class imbalance in the dataset, with considerably fewer samples for the

HadHeartAttack = True class, which reflects the real-world prevalence of such cases. To address this imbalance and improve the model's ability to accurately identify heart attack instances, we applied downsampling to the majority class (HadHeartAttack = False) by discarding a proportion of its samples. This method was chosen to create a more balanced dataset, ensuring that the model does not become biased towards the majority class during training. By retaining comparable class counts, we aimed to enhance the model's sensitivity and overall performance for predicting the minority class. While this approach reduced the total number of samples, it was carefully executed to maintain representativeness and minimize the loss of critical information. This adjustment was integral to building a model that can effectively predict heart attack cases in real-world scenarios.

Before Under-samping:

HadHeartAttack	
0	232578
1	13435

After Under-sampling:

HadHeartAttack	
0	100000
1	13435

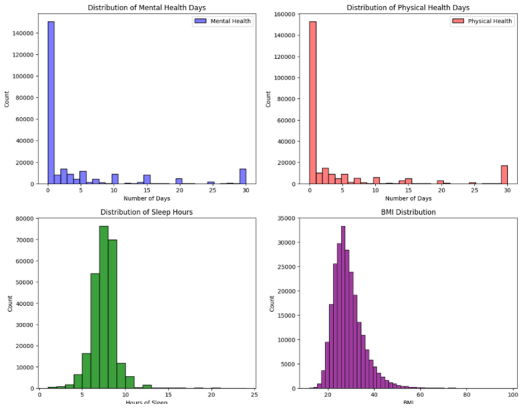
3.3. Exploratory Data Analysis (EDA)

Exploratory Data Analysis (EDA) is a critical preliminary step in data science and machine learning research that involves systematically investigating and visualizing datasets to uncover underlying patterns, detect anomalies, test hypotheses, and verify assumptions before building predictive models. The following subsections highlights the EDA done as part of this project.

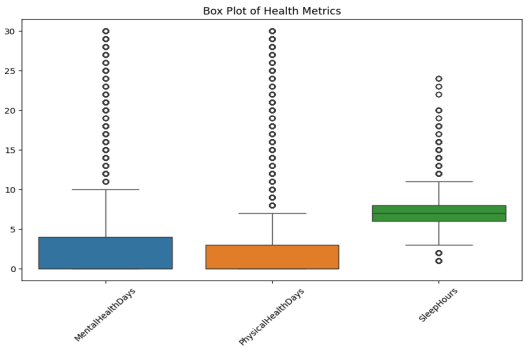
3.3.1. Numerical Variables Analysis

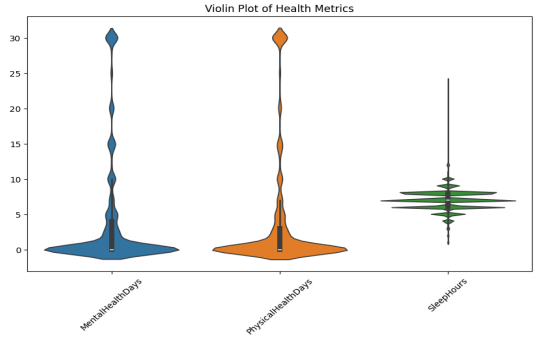
Numerical variables were examined through multiple visualization techniques:

Distribution Analysis: Bar graphs were created to illustrate the distribution of numerical health metrics, providing insights into the spread and central tendencies of key physiological measurements.

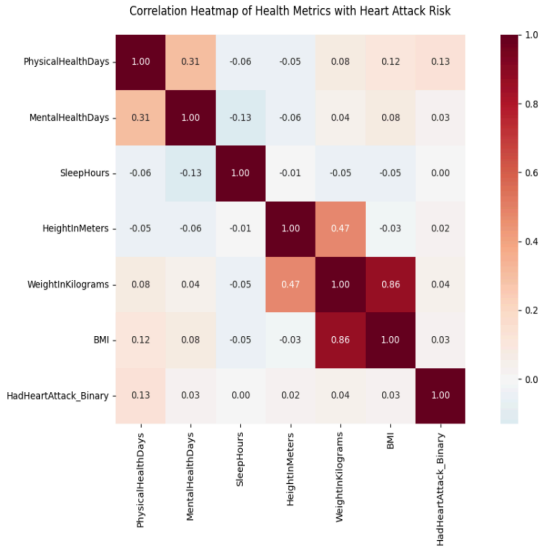


Statistical Visualization: Box plots and violin plots were employed to represent the statistical distribution of health metrics, highlighting median values, quartiles, and potential outliers. These visualizations offer a nuanced view of data variability and underlying statistical characteristics.





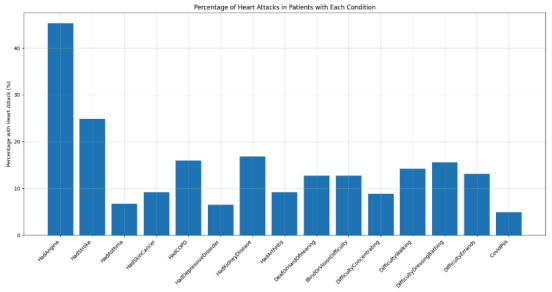
Correlation Assessment: A correlation map was generated to explore the relationship between numerical columns and the target variable of heart attack occurrence, quantifying the strength and direction of potential predictive features.



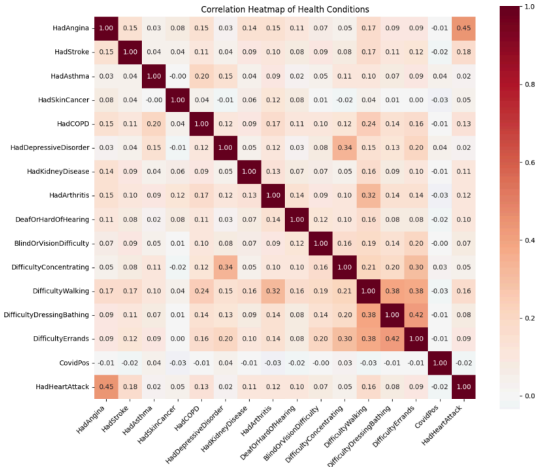
3.3.2. Categorical Variable Analysis

The investigation of categorical variables followed a systematic approach:

Heart Attack Percentage Analysis: Graphs were constructed to calculate and visualize the percentage of heart attacks corresponding to different symptom categories, providing a clear representation of categorical variable impacts.



Correlation Exploration: A correlation heatmap was developed to illustrate the relationships between the target variable and various categorical symptoms, revealing potential predictive patterns.

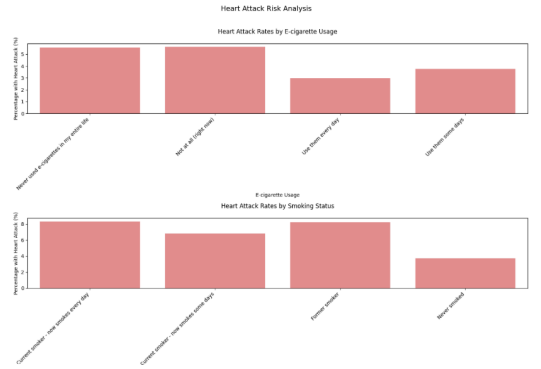


3.3.3. Demographic and Lifestyle Insights

Additional analyses focused on demographic and lifestyle factors:

Heart Attack Rates by Demographics: Visualizations were created to explore heart attack rates across different age groups and ethnic backgrounds, uncovering potential demographic risk variations.

Lifestyle Factor Comparison: A comparative analysis was conducted to examine heart attack rates among smokers versus non-smokers, highlighting the potential impact of lifestyle choices.



3.4. Evaluation parameters

Evaluation parameters are critical for measuring the performance of machine learning models. In this project, several metrics are used to assess the model's accuracy and reliability:

Accuracy: The proportion of correctly predicted instances over the total instances. While useful for balanced datasets, it may not be reliable in the case of imbalanced classes.

Precision: Measures the number of true positive predictions relative to the total predicted positives. It is particularly useful when the cost of false positives is high.

Recall (Sensitivity): Indicates how many actual positive cases were correctly identified by the model. It is crucial when missing a positive instance has severe consequences, such as in medical diagnoses.

F1 Score: The harmonic mean of precision and recall, providing a balance between the two metrics. It is particularly useful in situations where both false positives and false negatives are important.

Confusion Matrix: A table showing true positives, true negatives, false positives, and false negatives. It provides a clear visualization of the model's classification performance.

By using these evaluation metrics, we can comprehensively assess and compare the performance of different models, ensuring the selection of the most effective model for heart disease prediction.

4. Machine learning techniques implemented

Logistic regression:
Logistic regression is a statistical model used for binary classification. It predicts the probability of an outcome (0 or 1) based on input features, using the logistic function to map predictions to a value between 0 and 1. The model is trained to find the best-fitting parameters using maximum likelihood estimation.

Decision Tree Classifier:
A Decision Tree Classifier is a machine learning algorithm used for classification tasks. It splits the data into subsets based on feature values, creating a tree structure with decision nodes and leaf nodes. The tree is built by selecting the feature that best separates the data, often using metrics like Gini impurity or Information Gain.

AdaBoost Classifier:
AdaBoost (Adaptive Boosting) is an ensemble learning algorithm that combines multiple weak classifiers to create a strong classifier. It works by training classifiers sequentially, with each new classifier focusing more on the misclassified instances from the previous ones.

Gradient Boosting Classifier:
Gradient Boosting Classifier is an ensemble learning algorithm that builds a strong classifier by combining multiple weak classifiers, typically decision trees. It trains models sequentially, with each new model correcting the errors (residuals) of the previous one using gradient descent.

Random Forest Classifier:
Random Forest Classifier is an ensemble learning algorithm that combines multiple decision trees to make a final prediction. It builds a collection of decision trees (a forest) using random subsets of data and features, and then aggregates their predictions. The training data considered has been sampled before it was trained. The best value for hyper-parameter is found out using GridSearchCV with 5-fold cross validation technique. Thereafter we considered the range of estimators for which there is performance raise on the cv data and tried out all the values in that range to obtain best performance on the test data.

XGBoost Classifier:
XGBoost (Extreme Gradient Boosting) is an optimized and scalable version of gradient boosting, designed to be highly efficient, flexible, and portable. It is a powerful ensemble learning algorithm that builds decision trees sequentially, with each tree correcting the errors of the previous one using gradient descent. The training data considered has been sampled before it was trained. The best value for hyper-parameter is found out using GridSearchCV with 5-fold cross validation technique. Thereafter we considered the range of estimators for which there is performance raise on the cv data and tried out all the values in that range to obtain best performance on the test data

DNN Classifier:
DNN Classifier is a deep learning model built using the Keras library for classification tasks. Keras provides a high-level interface to create, train, and evaluate neural networks. It simplifies the process of developing models by abstracting many low-level details.The training data considered has been sampled before it was trained. The best value for hyper-parameter is found out using GridSearchCV with 5-fold cross validation technique. Thereafter we considered the range of estimators for which there is performance raise on the cv data and tried out all the values in that range to obtain best performance on the test data.

5. Hyperparameters Tuning with Cross-Validation

In this project, hyperparameter tuning was conducted to optimize the performance of the machine learning models. The goal of hyperparameter tuning is to identify the best combination of model parameters that lead to the best predictive performance. Cross-validation (CV) was used to ensure that the model generalizes well to unseen data by splitting the dataset into multiple training and validation sets. The following techniques were employed for tuning the hyperparameters of different classifiers:

5.1. Randomized Search with Cross-Validation for RandomForestClassifier and XGBoost Classifier

For the **RandomForestClassifier** and **XGBoost Classifier**, we used **RandomizedSearchCV** with **5-fold cross-validation** to search over a wide range of hyperparameters. RandomizedSearchCV is an efficient approach that randomly samples from a specified hyperparameter space, making it more computationally efficient than exhaustive search methods like Grid Search. It evaluates the performance of different combinations of hyperparameters and selects the best-performing set.

The key hyperparameters tuned for **RandomForestClassifier** and **XGBoost** included:

- **n_estimators**: The number of trees in the forest (RandomForest) or the boosting rounds (XGBoost).
- **max_depth**: The maximum depth of each tree to control overfitting.
- **learning_rate**: For XGBoost, the learning rate determines the contribution of each tree to the final prediction.
- **min_samples_split** (RandomForest): The minimum number of samples required to split an internal node.
- **subsample** (XGBoost): The fraction of samples used for fitting each tree.

By using **5-fold cross-validation**, we were able to ensure that the models were evaluated on different subsets of the dataset, which helped mitigate any potential overfitting and provided a more reliable estimate of model performance. This approach helped in selecting the best hyperparameters that resulted in the highest validation performance.

Following is the parameter grid used for Random Forest.

```
param_grid = {
    'n_estimators': [50, 100, 200],
    'max_depth': [None, 10, 20],
    'min_samples_split': [2, 5, 10],
    'max_features': ['sqrt', 'log2', None]
}
```

Following is the set of best hyperparameters.

```
'n_estimators': 100,
'max_depth': 20,
'min_samples_split': 10,
'max_features': 'sqrt',
```

Following is the Parameter grid used for XGBoost.

```
param_dist = {
    'n_estimators': [100, 200, 300],
    'learning_rate': [0.01, 0.1, 0.2],
    'max_depth': [3, 5, 7],
    'subsample': [0.5, 0.7, 0.9]
}
```

Following are the best hyper-parameters for XGBoost.

```
'n_estimators': 300,
'learning_rate': 0.01,
'max_depth': 3,
'subsample': 0.7,
```

5.2. Grid Search with Cross-Validation for Deep Neural Network Classifier

For the **Deep Neural Network (DNN) Classifier**, we used **GridSearchCV** with **3-fold cross-validation** to tune the hyperparameters. Unlike RandomizedSearchCV, GridSearchCV performs an exhaustive search over a manually specified hyperparameter grid. While this method can be more computationally expensive, it guarantees that all combinations of the specified

hyperparameters are tested, ensuring the best possible configuration.

The key hyperparameters tuned for the **Deep Neural Network** included:

- **Number of neurons per layer**: The number of neurons in each hidden layer, which directly affects the capacity of the model.
- **Batch size**: The number of samples used per gradient update.
- **Epochs**: The number of complete passes through the training dataset during model training. This is an essential hyperparameter that affects the model's ability to learn effectively.

The **3-fold cross-validation** was used here to split the dataset into three parts, ensuring that the model's performance was evaluated on multiple validation sets while minimizing computational cost. Given the complexity of neural networks, this method helped in identifying the optimal network architecture and training parameters.

Following is the parameter grid used for DNN.

```
param_grid = {
    'model__neurons': [128, 256],
    'batch_size': [512, 1024],
    'epochs': [10, 20],
}
```

Following is the best hyper-parameters for DNN.

```
'model__neurons': 256,
'batch_size': 1024,
'epochs': 10,
```

6. Model Evaluation and Results

After hyperparameter tuning, the best hyperparameters selected for each classifier were used to train the final model. The performance of each model was evaluated using various metrics (e.g., accuracy, precision, recall, F1 score) on a separate test set to gauge its ability to generalize to unseen data. The following table shows the performance of the different models w.r.t to the evaluation metrics that were achieved.

Model	Accuracy	Precision	Recall	F1-Score
Logistic Regression	0.83	0.89	0.83	0.85
Decision Tree	0.87	0.86	0.87	0.87
Random Forest	0.90	0.88	0.90	0.89
AdaBoost	0.91	0.90	0.91	0.90
Gradient Boosting	0.91	0.90	0.91	0.90
Random Forest (Hyper-parameter tuned)	0.91	0.90	0.91	0.90
XGBoost (Hyper-parameter tuned)	0.91	0.90	0.91	0.90
Artificial Neural Network (Hyper-parameter tuned)	0.91	0.90	0.91	0.90

A notable observation from the hyperparameter tuning process was the convergence of most models towards similar performance levels. This suggests that with appropriate parameter optimization, different machine learning algorithms can achieve comparable effectiveness in heart disease classification.

The consistent performance across models implies that the underlying feature set provides strong predictive signals, and the relationship between the selected health indicators and heart disease risk can be

effectively captured by various machine learning approaches.

The results underscore the potential of machine learning techniques in developing robust predictive models for heart disease risk assessment, highlighting the importance of both feature selection and careful model optimization.

Challenges	Solutions
Missing / Incomplete values	Dropped data points with missing values
Imbalance Dataset	Under-Sampling of majority class
Large set of features	Feature selection using EDA techniques such as correlation matrix
Bias in Dataset	Used Ensemble Learning such as Random Forest, Adaboost, Gradient Boosting, XGBoost,
Over-Fitting	Multiple models with multiple metrics such as accuracy, precision, recall and F1-score
Model Improvement	Hyperparameter tuning techniques such as
	Randomised Search and Grid Search with cross-validation

7. References

1. https://www.kaggle.com/datasets/kamilpytlak/personal-key-indicators-of-heart-disease/data
2. https://scikit-learn.org/1.5/modules/generated/sklearn.linear_model.LogisticRegression.html
3. https://xgboost.readthedocs.io/en/stable/get_started.html
4. https://scikit-learn.org/1.5/modules/generated/sklearn.ensemble.RandomForestClassifier.html
5. https://adriangb.com/scikeras/stable/
6. Keras: Deep Learning for humans
7. TensorFlow