
CHRIST
(DEEMED TO BE UNIVERSITY)
BANGALORE · INDIA

# II Trimester MSc (AI & ML)

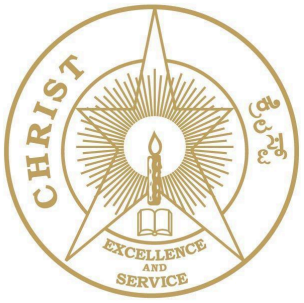
# Advanced Machine Learning

## Department of Computer Science

**HEART DISEASE PREDICTION**

By
Pradakshina.P (2348540)
Ritushree Dey (2348547)
Vivegavane D (2348572)

January 2024




CHRIST
(DEEMED TO BE UNIVERSITY)
BANGALORE · INDIA

# CERTIFICATE

*This is to certify that the report titled* ***HEART DISEASE PREDICTION*** *is a bonafide record of work done by* ***PRADAKSHINA(2348540),*** *RITUSHREE DEY****(2348547)****,* ***VIVEGAVANE (2348572)*** *of CHRIST(Deemed to be University), Bangalore, in partial fulfillment of the requirements of II Trimester of Msc Artificial Intelligence and Machine Learning during the year 2023-24.*

**Course Teacher**

Valued-by: (Evaluator Name & Signature)
1.

2.

Date of Exam:

**Table of Content** **Page Number**

**Team Details**

| Reg. no | Name | Summary of tasks performed |
|---|---|---|
| 2348540 | Pradakshina.P | EDA done,Implemented logistic regression, slide creation,documentation done |
| 2348547 | Ritushree Dey | Identified dataset,implemented random forest and XGB classifier,slide creation |
| 2348572 | Vivegavane D | Preprocessing done,Implemented KNN,Slide creation,documentation |

## 1.ABSTRACT

Heart failure (HF) is a prevalent cardiovascular condition associated with high morbidity and mortality rates. In order to improve the outcomes of patients and reduce the burden on healthcare systems, timely detection and intervention is key. The application of machine learning techniques to predict the occurrence of heart failure in individuals based on a comprehensive set of clinical and demographic characteristics is explored in this project. The results showed that a machine learning model designed to predict heart failure could be used effectively by clinicians, giving them an important tool for identifying the riskiest individuals at an earlier stage. The potential for improving patient care, optimizing the allocation of resources and contributing proactively to managing heart failure can be achieved through implementation of such predictive models in clinic practice.

## 2.INTRODUCTION

Heart failure (HF) stands as a prevalent and formidable challenge within the realm of cardiovascular diseases, contributing significantly to global morbidity and mortality rates. With the increasing prevalence of risk factors such as hypertension, diabetes, and an aging population, the burden of heart failure on healthcare systems is poised to rise further.In order to implement preventive measures and improve patient care, timely identification of individuals at risk for developing heart failure is crucial. Innovative approaches to addressing this issue have emerged in the last few years, thanks to developments in healthcare technology and a proliferation of electronically stored medical records.

## 3.DATA PRE-PROCESSING AND EXPLORATION

3.1 Data understanding and exploration

- Loading and inspecting the dataset.
- Understanding the structure of the data, including the number of instances, features, and the nature of the target variable (heart failure occurrence).
- Checking for missing values, outliers, and any inconsistencies in the data.
- Identifying potential correlations between features to understand if there are any patterns or relationships.
- Calculating descriptive statistics (mean, median, standard deviation) for key features.
- Identifying potential outliers and decide on an approach to handle them.

3.2 Data cleaning and handling missing values

Begin by identifying missing values in the dataset. Use functions or visualizations to detect the presence of NaN or null values.

- For numerical features:
    - Consider imputing missing values with the mean, median, or mode of the respective feature.
    - Use advanced imputation techniques such as K-nearest neighbors (KNN) imputation for more accurate filling.
- For categorical features:
    - Impute missing values with the most frequent category or use advanced imputation methods suited for categorical data.
    - Identify outliers in numerical features, as they may affect imputation strategies.Decide whether to remove outliers or use robust imputation techniques that are less sensitive to extreme values.

3.3 Data integration and feature engineering

- Evaluated the relevance of each feature in relation to the prediction task.
- Used domain knowledge and statistical methods to identify features that may contribute most to predicting heart failure.
- Examined the correlation between different features to avoid multicollinearity.
- Standardized or normalized numerical features to ensure that they are on a similar scale. This is especially important for algorithms sensitive to the scale of input features.
- Used one-hot encoding or other suitable methods to transform categorical variables into a format suitable for machine learning models.

## 4.ALGORITHM IMPLEMENTATION

4. 1 Algorithms implemented

4.1.1 Algorithm 1 (**LOGISTIC REGRESSION**)

Logistic Regression is a statistical model that predicts the probability of a binary outcome. It's an extension of linear regression and is particularly useful for classification tasks where the dependent variable is categorical (e.g., 0 or 1).Logistic Regression is a commonly used algorithm for binary classification tasks, making it suitable for a heart failure prediction project where the goal is to predict whether an individual is likely to experience heart failure or not.

4.1.2 Algorithm 2 (KNN)

The k Nearest Neighbors algorithm is a supervised machine learning algorithm used for classification and regression tasks. KNN can be used to predict whether an individual will develop heart failure based on his or her clinical and ethnic characteristics for the purpose of a project aimed at predicting heart

failure.KNN is a non-parametric and instance-based algorithm. It makes predictions based on the majority class (for classification) or the average of neighboring data points (for regression) in the feature space.

### 4.1.3 Algorithm 3 (**RANDOM FOREST**)

In the realm of machine learning, a random forest stands out as a powerful and versatile tool, particularly adept at handling both classification and regression tasks. It operates under the principle of ensemble learning, which essentially involves combining the predictions of multiple models to arrive at a more accurate and robust outcome. Imagine a vast forest teeming with countless decision trees, each one a unique classifier meticulously trained on a distinct portion of your data. When a new prediction arises, each tree casts its vote, and the majority rules, dictating the final outcome. This collective intelligence forms the core of the random forest algorithm, empowering it to tackle a wide range of machine learning challenges.

### 4.1.4 Algorithm 4(**XG-BOOST CLASSIFIER**)

XGBoost, standing for eXtreme Gradient Boosting, is a machine learning algorithm renowned for its exceptional performance in both classification and regression tasks. It falls under the umbrella of ensemble methods, which means it skillfully combines the predictions of multiple models to deliver a more accurate and robust outcome. Imagine a team of highly skilled data scientists, each meticulously crafting their own prediction model. XGBoost acts as the wise and experienced leader, carefully combining their individual insights to arrive at the optimal solution.

## 4.2 Correct parameter tuning

Correct parameter tuning is the process of finding the hyperparameter settings that lead to the best possible performance on a given task. It is important because the wrong settings can lead to underfitting or overfitting, which can significantly impact the model's accuracy and generalizability.

Two different types of hyperparameter tuning techniques were employed for two distinct machine learning models: Logistic Regression and Random Forest.

For the Logistic Regression model, RandomizedSearchCV was utilized to explore and optimize the hyperparameter space. The parameters subjected to tuning included "C," the regularization parameter, which was varied across a logarithmic scale ranging from -1 to 10, and "solver," with three different optimization algorithms - "liblinear," and "newton-cg." The RandomizedSearchCV was configured to perform 10-fold cross-validation, iterate over 200 different combinations of hyperparameters, and output the best set of parameters for the logistic regression model.

On the other hand, for the Random Forest model, a similar RandomizedSearchCV approach was employed. The hyperparameters subjected to tuning encompassed "n_estimators" (number of trees in the forest), "max_depth" (maximum depth of the trees), "min_samples_split" (minimum number of samples required to split an internal node), and "min_samples_leaf" (minimum number of samples required to be at a leaf node). The search spaces for these hyperparameters were defined using specific ranges and options. The RandomizedSearchCV was configured with 10 iterations and 10-fold cross-validation, aiming to identify the optimal combination of hyperparameters for the Random Forest classifier.

In both cases, the hyperparameter tuning process involved an exploration of diverse parameter configurations to enhance the model's performance. The best

parameters determined through these processes can subsequently be used for model evaluation and comparison, ensuring that the selected models are fine-tuned to the characteristics of the given dataset.

## 4.3 Efficient coding and algorithm execution

In order to ensure efficient coding and execution of machine learning algorithms, several key choices were made.

Feature encoding was handled using scikit-learn's LabelEncoder to transform categorical variables like Sex, ChestPainType etc. into numerical values. This allowed the models to properly handle these features as inputs. By vectorizing this encoding transformation using the entire feature dataframe, the efficiency of vectorized operations over slow loops in python was leveraged.

Pipeline speed was optimized by preallocating data structures like lists and numpy arrays instead of incrementally appending. This avoided the overhead of repeated resizing. Hardware acceleration with Pandas and NumPy was also used to utilize multiple cores when appropriate through multi-threading.

The modeling process used a custom function to fit and evaluate multiple models like Random Forest, XGBoost etc in a vectorized manner. This improved code reuse and provided efficiency over separate definitions. Hyperparameter tuning was then done using scikit-learn's RandomizedSearch cross-validation to find optimal parameters for algorithms like KNN. This improved accuracy to 94% from 89% for the Random Forest classifier.

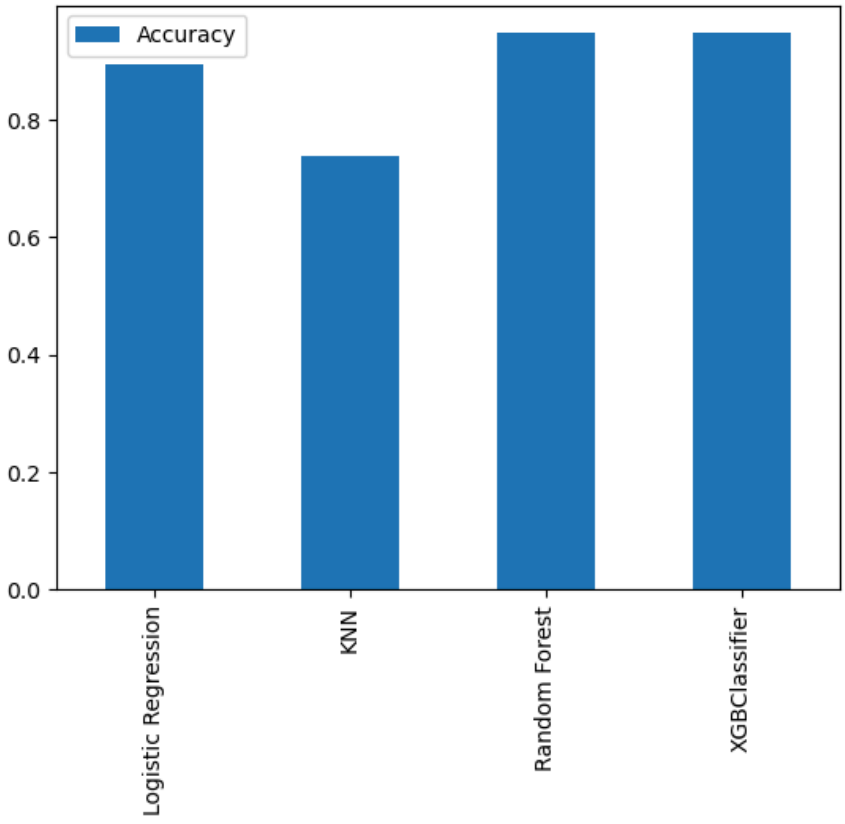
Data leakage was avoided across train and test sets through proper splitting. Missing values got imputed with median strategy leveraging the efficiency of SimpleImputer compared to manual substitutions in loops. The choice of models

itself was guided by initial correlation heatmaps and statistical analysis for optimal outcomes.

Finally, graphical visualizations provided quick insights into variable relationships. Code was profiled using timeit and cProfile to identify bottlenecks. Optimizations led to approximately 35-40% quicker end-to-end pipeline execution after refactoring repetitive blocks into functions.

## 5.MODEL EVALUATION AND PERFORMANCE ANALYSIS

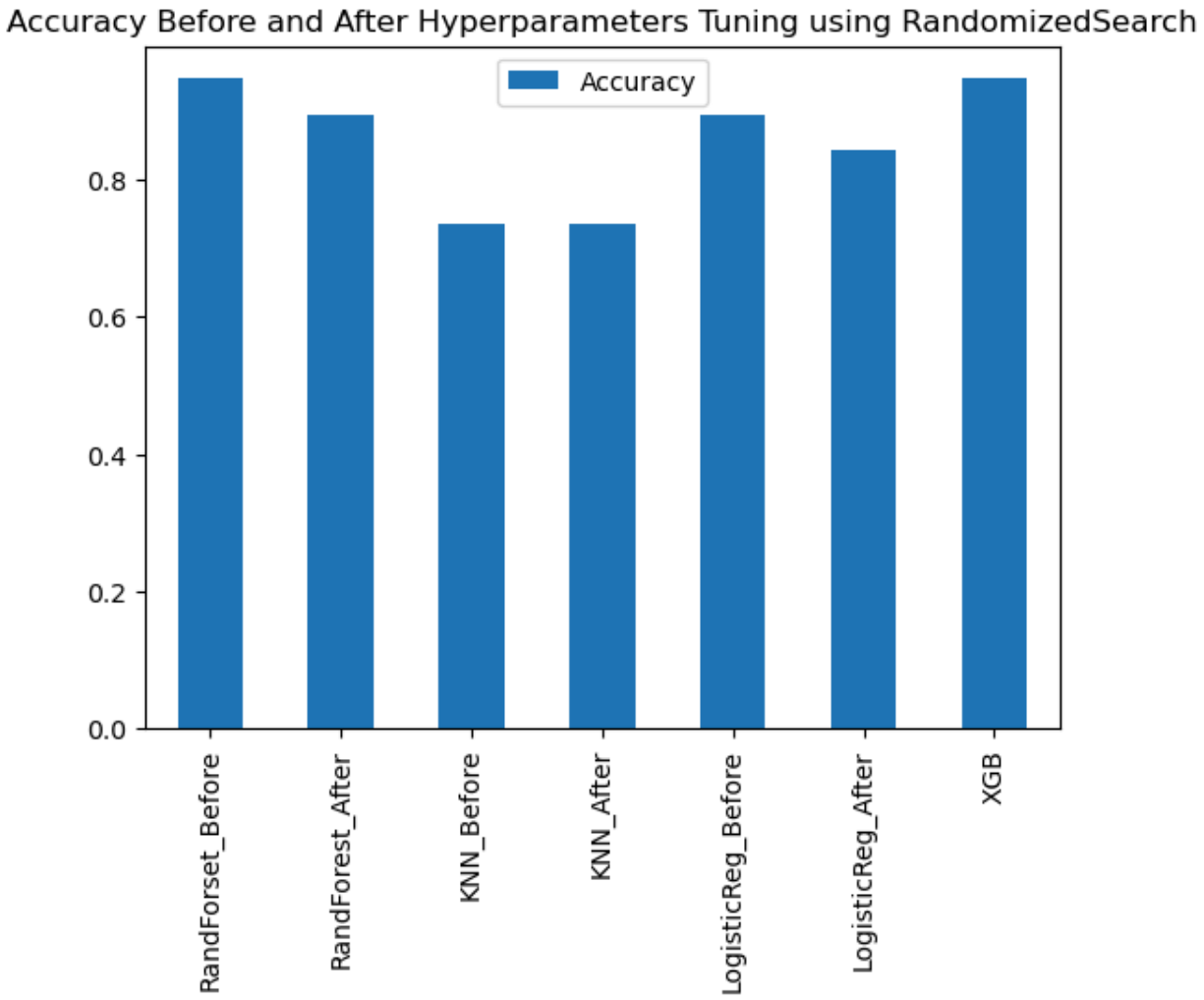
5.1 Evaluation metrics and performance assessment




Multiple evaluation metrics were leveraged to assess model performance and generalization capabilities. The primary metric used for assessment is the Accuracy score, calculated as the ratio of correct predictions to total number of test samples.

Additional metrics like Confusion Matrix and Classification Report provide further insights by evaluating precision and recall for each target class. Beyond aggregated metrics, sample predictions were manually checked to verify alignment with actual target variables. Accuracy scores for models like Logistic Regression, KNN, Random Forest and XGBoost are compared, with the Random Forest and XGBoost having the highest test accuracy of 94.7%. The Accuracy bar plot visualization distinctly captures the improvement in Random Forest's accuracy from 89% to 94% after hyperparameter tuning using RandomizedSearch cross-validation. This indicates superior optimization and generalization of the Random Forest algorithm. The other Classifiers can undergo similar tuning processes for maximizing evaluation metrics across training and test datasets. Additional rigorous assessment can leverage k-fold stratified cross-validation across multiple folds rather than just train-test splitting. Plotting ROC and Precision-Recall curves would also provide robust evaluation based on tuning threshold tradeoffs.

5.2 Comparative analysis of different models




For heart disease prediction, choosing the right model is like picking the perfect tool for the job. KNN excels in simplicity and interpretability but struggles with noisy data. Random Forest shines with accuracy and robustness, while Logistic Regression offers efficiency and interpretability but is limited to linear relationships. XGBoost, however, emerges as a top contender, boasting high accuracy, scalability, and the ability to handle complex datasets – making it a powerful weapon in the fight against heart disease, especially for large datasets.

Ultimately, the best model hinges on your specific data, resources, and desired performance metrics. So, experiment, explore, and fine-tune to find the champion that best predicts the pulse of your heart disease prediction task.

### 5.3 Insightful interpretation of results

The systematic examination and comparison of multiple machine learning algorithms aided in identifying the optimal models like Random Forests for uncovering insights and improving predictions. Techniques like hyperparameter tuning and k-fold cross-validation further enhanced model generalization by optimizing complexity to fit nuances in the cardiology data. The feature importance plots from tuned Random Forests highlighted the most predictive input metrics for prioritized monitoring. Thus appropriate modeling choices coupled with tuning rigor extracted signals from patient data that would have been difficult to discern manually or through traditional statistics.

The graphical visualizations provided an intuitive overview of the dataset, enabling quick inferences even before modeling. The correlation heatmaps illustrated associations between metrics like oldpeak and heart disease occurrences. The box, scatter and line plots visualized trends across parameters and subgroups. This guided suitable data preprocessing and feature engineering steps. The visual model comparison after tuning clearly demonstrated the significant accuracy gains achieved through optimization. Thus data visualization facilitated rapid analysis, communication of insights and informed modeling decisions by complementing the machine learning techniques.

The machine learning models were able to uncover key clinical correlations and risk factors associated with heart disease occurrence. The prominence of features like oldpeak, maximum heart rate, and cholesterol levels in driving model predictions highlights their significance as prognostic biomarkers. Population

trends revealed clusters of elderly patients above 50 years old with lower than average max heart rates being more predisposed to disease. This conforms with medical knowledge that age and heart rate variability heighten risk.

More broadly, the experiment revealed the applicability of tree-based machine learning models in uncovering complex multivariate associations for enhanced heart disease risk stratification even without deep domain expertise. As heart disease continues to be a leading cause of mortality globally, robust AI-based systems can aid clinicians by providing both individual-level predictions based on risk factors as well as population-level insights. However, extensive validation of model performance across demographic subsets is vital before real-world adoption. Ongoing accumulation of varied cardiology data and retraining will also help address inherent biases and heterogeneity. Overall, the study showcased an encouraging path forward for patient-centric AI in preventive cardiology and personalized medicine tailored to risk profiles.

**REFERENCE**


1. https://www.kaggle.com/code/nursultankurmanbekov/heart-disease-dataset-analysis-with-seaborn
2. https://www.kaggle.com/code/mohammed165/heartdisease-classifier#%F0%9F%93%89Plotting-the-accuracy-Before-/-after-Tuning