"In Pursuit of Global Competitiveness"

THESIS

On

"AI Based Heart Disease Detection System"

Submitted By

Uday Khade	**BE20F04F038**
Akansha Lokhande	**BE20F04F042**
Pritam Nikumbh	**BE20F04F048**
Shaurya Singh	**BE20F04F056**

Guided By

Prof. Sangeeta Bharkad

For the Degree of

Bachelor of Technology

(Electronics & Telecommunication Engineering)

DR. BABASAHEB AMBEDKAR MARATHWADA UNIVERSITY,

AURANGABAD

Department of Electronics and Telecommunication Engineering,

Government College of Engineering, Chhatrapati Sambhaji Nagar

(An Autonomous Institute of Government of Maharashtra) (2022-2023)

CERTIFICATE

This is to certify that, the thesis report entitled "AI Based Heart Disease Detection" submitted by Uday Khade (BE20F04F038), Akansha Lokhande (BE20F04F042), Pritam Nikumbh (BE20F04F048) & Shaurya Singh (BE20F04F056) is a bonafide work completed under my supervision and guidance in partial fulfillment for award of Bachelor of Technology (Electronics & Telecommunication), Degree of Government College of Engineering (An Autonomous Institute of Government of Maharashtra), affiliated to Dr. Babasaheb Ambedkar Marathwada University, Aurangabad.

Place: Chhatrapati Sambhajinagar

Date:

Prof. Sangeeta Bharkad
Guide
Department of Electronics &
Telecommunications Engineering

Prof. Dr. S. R. Hirekhan
Head of the Department
Department of Electronics &
Telecommunications Engineering

Prof. Dr. S. S. Dambhare
Principal
Government College of Engineering ,
Chhatrapati Sambhajinagar

THESIS APPROVAL SHEET

Uday Khade (BE20F04F038), Akansha Lokhande (BE20F04F042), Pritam Nikumbh (BE20F04F048) & Shaurya Singh (BE20F04F056) have done the appropriate work related to "AI Based Heart Disease Detection" for the award of Bachelor of Technology (Electronics & Telecommunication), is being submitted to Government College of Engineering, Aurangabad.

Guide: Prof. Sangeeta Bharkad

Place: Government College of Engineering, Chhatrapati Sambhajinagar

Date:

DECLARATION

We hereby declare that we have formed, completed and written the thesis entitled “AI Based Heart Disease Detection”. It has not previously been submitted for the basis of the award for any degree or diploma or other similar title for any other diploma / examining body or university.

Place : Chhatrapati Sambhajinagar

Date :

Uday Khade	**BE20F04F038**
Akansha Lokhande	**BE20F04F042**
Pritam Nikumbh	**BE20F04F048**
Shaurya Singh	**BE20F04F056**

CONTENT

LIST OF ABBREVIATIONS

Abbreviations	Illustrations
ECG	Electrocardiogram
KNN	K-Nearest Neighbour
CNN	Convolutional Neural Network
SVM	Support Vector Machine
LR	Logistic Regression
XG Boost	Extreme Gradient Boosting
UI	User Interface
RDBMS	Relational Database Management System
PCA	Principal Component Analysis

ABSTRACT

Heart disease is a leading cause of mortality worldwide, emphasizing the need for accurate and timely diagnosis. In this study, we propose an AI-based approach for heart disease detection using electrocardiogram (ECG) images. Our model predicts the patient's heart condition into four categories: myocardial infarction, normal heart, person with myocardial history, and abnormal heartbeat. We employ a deep learning architecture trained on a comprehensive dataset of ECG images, incorporating image processing techniques for feature extraction. Through rigorous experimentation and evaluation, our model demonstrates promising performance in accurately classifying heart conditions. Our findings suggest that this AI-based approach has the potential to assist healthcare professionals in early diagnosis and intervention, ultimately improving patient outcomes and reducing healthcare costs.

ACKNOWLEDGEMENT

Completion of our thesis was a task which would have not been accomplished without cooperation and help from our guide. We wish to express our deep sense of gratitude to our guide **Prof. Sangeeta Bharkad** for her guidance and constant encouragement, without which it would have not been possible. We would like to thank **Prof. Dr. S. R. Hirekhan**, Head, Electronics & Telecommunication Department for his encouragement, guidance and allowing us to use all the facilities in the department. We are also very much grateful to Principal **Prof**. **Dr. S. S. Dambhare** who has been a constant source of inspiration. We are also thankful to my friends who stood with us throughout the completion of this thesis. Lastly, we also thank my parents for their love and support.

Uday Khade	**BE20F04F038**
Akansha Lokhande	**BE20F04F042**
Pritam Nikumbh	**BE20F04F048**
Shaurya Singh	**BE20F04F056**

CHAPTER 1: INTRODUCTION

1.1 INTRODUCTION TO ARTIFICIAL INTELLIGENCE IN CARDIOVASCULAR DISEASE

Heart disease is a leading cause of mortality worldwide, emphasizing the need for accurate and timely diagnosis. In this study, we propose an AI-based approach for heart disease detection using electrocardiogram (ECG) images. Our model predicts the patient's heart condition into four categories: Myocardial Infarction, normal heart, person with Myocardial history, and abnormal heartbeat. We employ a deep learning architecture trained on a comprehensive dataset of ECG images, incorporating image processing techniques for feature extraction. Through rigorous experimentation and evaluation, our model demonstrates promising performance in accurately classifying heart conditions. Our findings suggest that this AI-based approach has the potential to assist healthcare professionals in early diagnosis and intervention, ultimately improving patient outcomes and reducing healthcare costs.

1.2 KEY FEATURE OF HEART DISEASE DETECTION

Deep Learning Model Architecture:

Our project utilizes a deep learning model architecture specifically designed for heart disease detection using ECG images. The model is based on convolutional neural networks (CNNs) which are well-suited for image classification tasks.

Multi-Class Classification:

The model is trained to classify ECG images into four distinct categories:
Myocardial infarction: Indicates a heart attack.
Normal heart: Represents a healthy heart condition.
Person with Myocardial history: Indicates a patient with a history of myocardial infarction.
Abnormal heartbeat: Indicates irregularities in the heart rhythm.

ECG Image Processing:

Prior to model input, ECG images undergo preprocessing steps to enhance their quality and extract relevant features. These steps may include noise reduction, baseline correction, and normalization.

Comprehensive Dataset:
We have curated a comprehensive dataset of ECG images covering a wide range of heart conditions. This dataset ensures that the model learns to generalize well across different variations in ECG signals.

Training Process:
The model undergoes extensive training on the dataset using state-of-the-art optimization techniques and hyperparameter tuning to achieve optimal performance.
Interpretability and Explainability:
Our model provides insights into its decision-making process, allowing healthcare professionals to interpret and understand the factors contributing to the classification results.

Real-Time Prediction:
Once deployed, the model can make predictions on new ECG images in real-time, enabling quick and efficient diagnosis of heart conditions.

Scalability and Adaptability:
The architecture of our model allows for scalability, making it adaptable to varying levels of data and computational resources. It can also be easily integrated into existing healthcare systems.

Evaluation Metrics:
Performance evaluation of the model is conducted using various metrics such as accuracy, precision, recall, and F1-score to ensure its effectiveness in heart disease detection.

Clinical Validation:
The model's predictions are validated against clinical data and expert opinions to ensure its reliability and accuracy in real-world scenarios.

Potential Impact:
Our AI-based heart disease detection system has the potential to revolutionize cardiac healthcare by enabling early detection of heart conditions, leading to timely intervention and improved patient outcomes.

1.3 MOTIVATION

Heart disease remains a significant public health concern worldwide, with its prevalence and associated mortality rates continuing to rise. Early detection and intervention are crucial in mitigating the impact of heart-related conditions, yet current diagnostic methods often rely on costly and time-consuming procedures, leading to delays in treatment and poorer patient outcomes. The motivation behind our project stems from the urgent need for an accurate, efficient, and accessible solution for heart disease detection. By leveraging the advancements in artificial intelligence and medical imaging, we aim to develop a system capable of analyzing electrocardiogram (ECG) images to identify various heart conditions swiftly and accurately. This project seeks to empower healthcare professionals with a tool that not only enhances diagnostic accuracy but also streamlines the diagnostic process, ultimately improving patient care and reducing the burden on healthcare systems.

1.4 NEED OF WORK

Early Detection Saves Lives: Early detection of heart disease significantly improves patient outcomes by enabling timely intervention and treatment. AI-based systems have the potential to detect subtle abnormalities in ECG signals that may not be apparent to human observers, thus facilitating early diagnosis.

Rising Healthcare Costs: Heart disease places a significant financial burden on healthcare systems globally. By streamlining the diagnostic process and reducing unnecessary tests and procedures, AI-based detection systems can help alleviate this burden and make healthcare more cost-effective.

Limited Access to Expertise: In many regions, there is a shortage of cardiologists and specialists trained in interpreting ECGs. An automated system for heart disease detection can bridge this gap by providing accurate assessments even in areas with limited access to medical expertise.

Growing Data Complexity: With the increasing volume and complexity of medical data, there is a need for automated tools that can efficiently process and analyze large datasets. AI algorithms excel at extracting meaningful patterns and insights from complex data, making them ideal for interpreting ECG images and identifying subtle indicators of heart diseases.

Personalized Medicine: AI-based systems can enable personalized medicine by tailoring treatment plans to individual patients based on their unique cardiac profiles. By accurately categorizing heart conditions, these systems can assist healthcare providers in developing personalized treatment strategies for better patient outcomes.

1.5 SCOPE OF SYSTEM

ECG Image Analysis: The system will focus on the analysis of electrocardiogram (ECG) images to detect various heart conditions, including myocardial infarction, normal heart, persons with myocardial history, and abnormal heartbeat. The analysis will involve preprocessing of ECG images, feature extraction, and Training with Machine Learning Algorithms and classification using deep learning techniques.

Multi-Class Classification: Our system will be capable of classifying ECG images into multiple categories representing different heart conditions. This includes distinguishing between normal and abnormal heart rhythms, as well as identifying specific cardiac abnormalities such as myocardial infarction.

Real-Time Prediction: The system will provide real-time predictions of heart conditions based on input ECG images. This feature enables healthcare professionals to make rapid diagnostic decisions and initiate appropriate interventions promptly.

Interpretability: Our system will provide insights into its decision-making process, allowing healthcare professionals to understand the factors contributing to the classification results. Interpretability is essential for gaining trust in the system's predictions and facilitating collaboration between the AI system and human experts.

Scalability: The system will be designed to handle large volumes of ECG data efficiently, making it suitable for deployment in diverse healthcare settings. It will be scalable to accommodate future expansions in data size and computational resources.

Integration with Healthcare Systems: Our system will be designed for seamless integration with existing healthcare systems, allowing healthcare providers to access and utilize its capabilities within their workflow. This integration ensures that the AI-based heart disease detection system becomes an integral part of routine clinical practice.

Validation and Clinical Use: The system will undergo rigorous validation against clinical data and expert opinions to ensure its reliability and accuracy in real-world scenarios. Once validated, it will be ready for clinical use, supporting healthcare professionals in making informed decisions regarding patient care.

User Interface: The system will feature a user-friendly interface that enables healthcare professionals to interact with the system easily facilitating quick interpretation and decision-making.

1.6 OBJECTIVES

- To address and acknowledge Public health concerns specifically about Cardiovascular Health.
- To minimize the gap between current limitations and shortcomings of Diagnostic methods.
- To utilize advancements in Artificial Intelligence in Imagining and diagnosis of cardiovascular disease.
- Our specific focus is on ECG which is a widely used diagnostic method for detection of heart disease.
- Aiming for high potential impact of the project on improving patient care by enabling faster diagnosis and intervention, leading to better treatment outcomes.
- Overall Goal of this project is to develop an AI-powered solution for heart disease detection that is accurate, efficient, and accessible, ultimately benefiting both healthcare professionals and patients.

1.7 STRUCTURE OF THESIS FOR PROJECT

- **Chapter 1** includes the Introduction, Objective, Need, Thesis Organization of the AI Based Heart Disease Detection System and its technical and use. It concisely elaborates the importance of the mentioned topic.

- **Chapter 2** Describes Literature Review. It explains briefly about previously used methods available and then their use in the Medical field .

- **Chapter 3** Explains the algorithms, software and libraries used to implement the project with the integration of frontend and backend components.

- **Chapter 4** Deals with the Experimental Analysis along with the flow of the system.

- **Chapter 5** Concludes the work done which directs for future scope in the particular applications.

CHAPTER 2: LITERATURE REVIEW

2.1 PROJECT BACKGROUND

Cardiovascular disease detection system involves use of Machine Learning Algorithms to train a system which will predict the result using ECG images. The training of a model involves data acquisition , data preprocessing , extraction of signals and finally training and testing with various algorithms.
The user end application interface is built with python and steam-lit which allows users to enter ECG Image and also see the step by step processing with the concluded result.

2.2 REVIEW OF WORK

Machine learning and deep learning have been widely employed in various studies to predict and identify heart diseases.
1] Obenshain demonstrated the potential of data mining methods in healthcare, highlighting the successful application of automated surveillance for the transmission of infectious diseases. This underscores the broad range of applications for machine learning in the medical field .

2] Sellappan Palaniappan and Rafiah Awang developed a technique to extract buried knowledge from past heart disease databases. They used the Naive Bayes algorithm, achieving an accuracy rate of 86%. This approach aimed to assist healthcare professionals in making more informed decisions and reducing malpractice claims .

3] S. Nikhar and A. Karandikar studied the prediction of cardiac illnesses using Decision Tree and Naive Bayes classifiers. Decision Tree classifiers outperformed Naive Bayes in terms of accuracy, attributed to information gain calculations. Additionally, they proposed using a Selective Naive Bayes classifier to enhance accuracy further.

4] Shadman Nashif evaluated the effectiveness of various machine learning algorithms, including Support Vector Machine (SVM), Naive Bayes, Artificial Neural Networks, Random Forest, and Simple Logistic Regression. Their research found that SVM consistently provided the best outcomes, particularly in terms of precision, recall, and sensitivity. They developed a real-time tracking system using Arduino and SVM algorithm, which effectively detected cardiac illnesses .

5] V V Ramalingam examined the performance of different machine learning algorithms for cardiovascular disease prediction. Random Forest models performed remarkably well due to their ability to address overfitting. Models built on Naive Bayes classifiers also showed good performance with low computational time. SVM consistently performed well in their evaluation .

6] Chithambaram T e investigated machine learning methods for predicting cardiovascular disease. When tested, K-Nearest Neighbors, Random Forest, Decision Tree, Correlation, and SVM achieved proportional accuracy rates of 63.4%, 68.4%, 71.4%, and 86.2%, respectively.

7] In a study by A. H. Khan , various heart diseases were classified using 12-lead ECG images with SSD MobileNet V2, achieving a success rate of 98.33% .

8] T. Anwar and S. Zakir discussed the effects of picture augmentation on ECG readings. Efficient Net B3 was used to classify images into four categories: Normal, COVID, Myocardial Infarction, and History of Myocardial Infarction. However, they found that picture augmentation techniques might alter data patterns, reducing effectiveness .

9] J. Aspuru presented a system based on linear regression algorithms for identifying ECG waves' fiducial spots, achieving an average sensitivity of 99.5%.

10] U. R. Acharya used CNNs to detect Myocardial Infarction in ECG signals, achieving average accuracy rates of 93.53% and 95.22% with and without noise reduction, respectively.

In summary, machine learning and deep learning methods have shown significant promise in predicting heart diseases with high accuracy. These techniques offer opportunities for creating individualized prevention and treatment plans. However, there is a need for more diverse datasets to verify results and improve model precision.

2.3 INDEX TERMS OF ECG

An Electrocardiogram (ECG or EKG) is a diagnostic test that measures the electrical activity of the heart over time. It is a non-invasive procedure commonly used to detect and monitor various heart conditions.

During an ECG, electrodes are placed on specific points on the body, typically on the chest, arms, and legs. These electrodes detect the electrical impulses generated by the heart as it beats. The impulses are recorded as waves on a graph, which represent the different phases of the heart's activity. The basic technical details of an ECG include:

Electrodes: Small, adhesive patches that are placed on the skin to detect the electrical signals from the heart.

Leads: Wires that connect the electrodes to the ECG machine. The number of leads used can vary depending on the type of ECG being performed.

ECG Machine: A device that records and displays the electrical signals from the heart as waves on a graph. Modern ECG machines may also have digital capabilities for storing and analyzing data.

Waves and Intervals**:** The ECG graph typically displays several waves and intervals, including the P wave (atrial depolarization), QRS complex (ventricular depolarization), and T wave (ventricular repolarization). These components provide information about the heart's rhythm and function.

Overall, an ECG provides valuable insights into the electrical activity of the heart and helps healthcare professionals diagnose various heart conditions, such as arrhythmias, heart attacks, and abnormal heart rhythms.

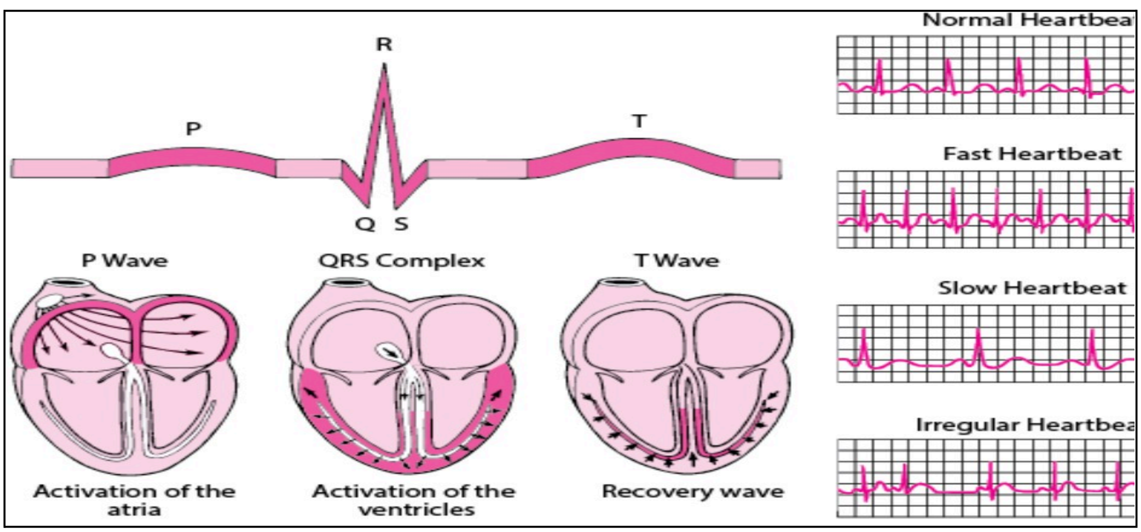


Fig 1

2.4 ANALYSIS OF ECG GRAPH BASED ON ABNORMALITIES

Normal ECG:
A normal electrocardiogram (ECG) exhibits regular patterns of electrical activity in a healthy heart, demonstrating typical P waves, QRS complexes, and T waves. These waves represent the sequential depolarization and repolarization of the atria and ventricles, indicating a normal heart rhythm and function. In a normal ECG, the P wave reflects atrial depolarization, the QRS complex signifies ventricular depolarization and contraction, and the T wave represents ventricular repolarization. Absence of significant deviations from these patterns indicates a healthy cardiac status.

Myocardial Infarction (Heart Attack) ECG:
An ECG during a myocardial infarction (heart attack) reveals specific abnormalities indicative of heart muscle damage and ongoing ischemia. These abnormalities include ST segment elevation, T wave inversion, or the presence of pathological Q waves. ST segment elevation typically indicates acute myocardial injury, while T wave inversion can suggest ischemia or injury to the myocardium. Pathological Q waves, which are wide and deepened, may appear during or after an acute myocardial infarction and indicate irreversible myocardial damage. These ECG changes play a crucial role in diagnosing acute myocardial infarction and guiding immediate medical intervention.

Abnormal Heartbeat (Arrhythmia) ECG:
An ECG of a patient with arrhythmia displays irregular patterns of electrical activity, reflecting abnormal heart rhythms such as tachycardia, bradycardia, or atrial fibrillation. These irregularities are observed as deviations from the normal morphology of the P waves, irregular R-R intervals, or absence of certain waves. Tachycardia manifests as rapid heartbeats, with shortened R-R intervals and possibly absent P waves. Bradycardia, on the other hand, exhibits slow heart rates, with prolonged R-R intervals and abnormal P wave morphology. Atrial fibrillation is characterized by chaotic and irregular electrical activity, resulting in irregularly spaced R-R intervals and absent P waves in many cases. These ECG findings provide valuable information for diagnosing and managing arrhythmias.

History of Myocardial Infarction ECG:
An ECG in a patient with a history of myocardial infarction may show persistent changes reflecting prior heart muscle damage. These changes include widened and deepened Q waves, T wave abnormalities, or alterations in the ST segment. Widened and deepened Q

waves often signify transmural myocardial infarction, indicating significant myocardial damage. T wave abnormalities, such as inversion or flattening, may persist after an acute event and can indicate ongoing ischemia or scar tissue formation. ST segment alterations, such as elevation or depression, may also persist and reflect myocardial ischemia or remodeling. These ECG findings provide insights into the patient's cardiac history and ongoing cardiac health, guiding long-term management and monitoring.

2.5 DATASET AND IT'S CLASSIFICATION

We utilized a dataset comprising four categories for image classification of ECG images:

1. **Normal**: This category represents ECG images from individuals with normal cardiac function and rhythm.
2. **Myocardial Infarction**: ECG images in this category depict patients diagnosed with acute myocardial infarction, commonly known as a heart attack.
3. **Abnormal Heartbea**t: This category includes ECG images showing irregular patterns of electrical activity, indicating arrhythmias such as tachycardia, bradycardia, or atrial fibrillation.
4. **History of Myocardial Infarction**: ECG images in this category belong to patients with a documented history of myocardial infarction, showing persistent changes indicative of prior heart muscle damage.

The dataset was obtained from ECG Images dataset of Cardiac Patients - Mendeley Data, specifically from the ECG images dataset of Cardiac Patients created under the auspices of Ch. Pervaiz Elahi Institute of Cardiology Multan, Pakistan. This dataset aims to support the scientific community in conducting research on cardiovascular diseases.

The dataset contains a diverse collection of ECG images, each labeled with one of the four categories mentioned above. The images were collected using standard ECG recording equipment, ensuring consistency and quality across the dataset.

CHAPTER 3: SYSTEM DEVELOPMENT

3.1 PROBLEM DEFINITION

Heart disease remains a leading cause of mortality worldwide, necessitating early detection for effective intervention. However, current diagnostic methods often involve costly and time-consuming procedures, leading to delays in treatment and poorer patient outcomes. There is a critical need for an accurate, efficient, and accessible solution for heart disease detection. Our project aims to address this need by developing an AI-powered system capable of swiftly analyzing electrocardiogram (ECG) images to identify various heart conditions. By automating the analysis process, we aim to enhance diagnostic accuracy, streamline the diagnostic workflow, and ultimately improve patient care while reducing the burden on healthcare systems.

3.2 HARDWARE REQUIREMENT

	SYSTEM REQUIREMENT	SPECIFICATION
1)	PROCESSOR	INTEL CORE i-5
2)	RAM	8 GB
3)	HARD DRIVE	512 GB (SSD RECOMMENDED)
4)	GRAPHICS PROCESSOR	NVIDIA GeForce MX250
5)	CPU	2.4 GHz

3.3 SOFTWARE

	SOFTWARE	VERSIONS
1)	ANACONDA DISTRIBUTIONS	2023.09
2)	PYTHON	3.11
3)	JUPYTER NOTEBOOK	7.0
4)	VISUAL STUDIO CODE	17.7
5)	PYCHARM COMMUNITY	2024.1

3.4 ARCHITECTURE / BLOCK DIAGRAM

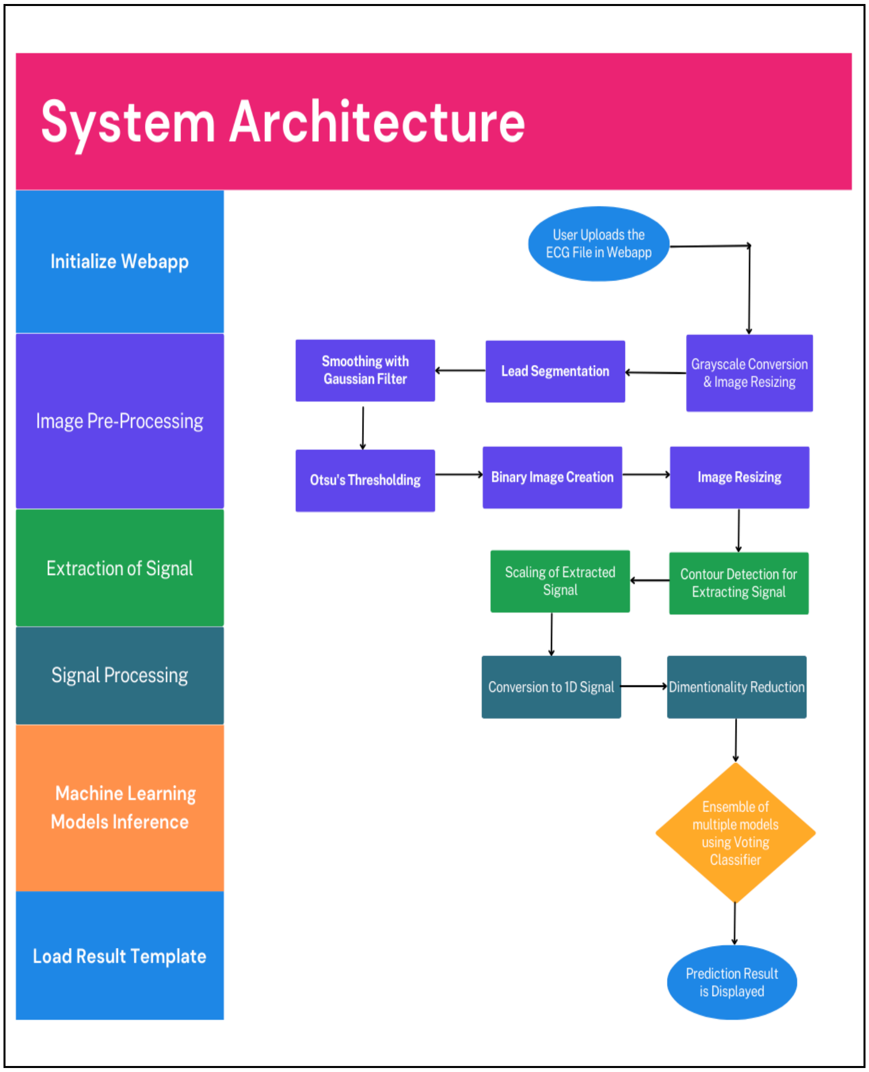


Fig 2

3.5 DATA COLLECTION

We will use our existing dataset and the specifications of that is as following

SPECIFICATION	NO OF IMAGES
NORMAL	284
ABNORMAL HEARTBEAT	233
MYOCARDIAL INFARCTION	239
HISTORY OF MYOCARDIAL INFARCTION	179
TOTAL	928

3.6 DATA PREPARATION AND PREPROCESSING

1) Image Acquisition and Grayscale Conversion

The uploaded ECG input is first passed through and color conversion and image resizing function.

Colour Conversion: It converts the user-provided image, which might be in RGB format (containing red, green, and blue channels), into grayscale format. Grayscale images use a single channel to represent intensity levels, simplifying subsequent processing steps.

Image Resizing: It resizes the grayscale image to a standard dimension of 1572x2213 pixels. This ensures consistency in the size of the analyzed ECG image across different datasets.

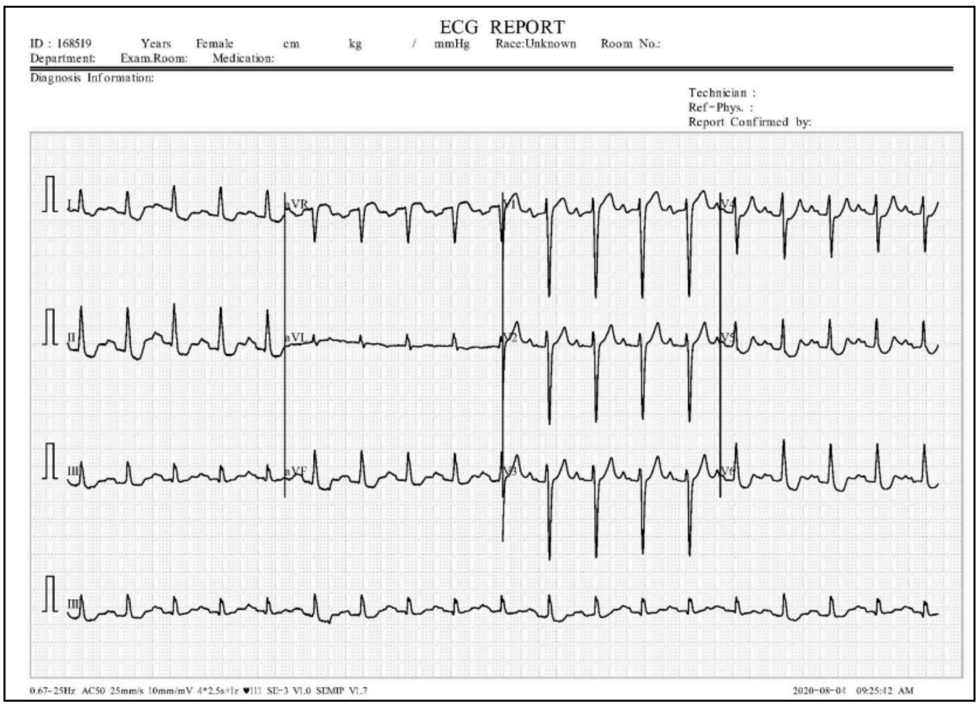


Fig 3

2) Lead Segmentation

After this a function segments the pre-processed grayscale image into the individual leads (electrocardiogram measurement points) relevant for ECG analysis. In this case, it extracts 13 leads, including:

Bipolar Limb Leads (Leads 1, 2, 3): These leads measure the voltage difference between two limb placements.

Augmented Unipolar Limb Leads (aVR, aVL, aVF): These leads estimate the voltage at a specific limb relative to an average reference created from all limbs.

Unipolar Chest Leads (V1 to V6): These leads measure the voltage at specific locations on the chest wall relative to a reference electrode (often the right leg).

The function achieves segmentation by relying on predefined coordinates within the image. It extracts rectangular image regions corresponding to each lead's location. The function returns a list containing all 13 segmented leads as separate grayscale images.

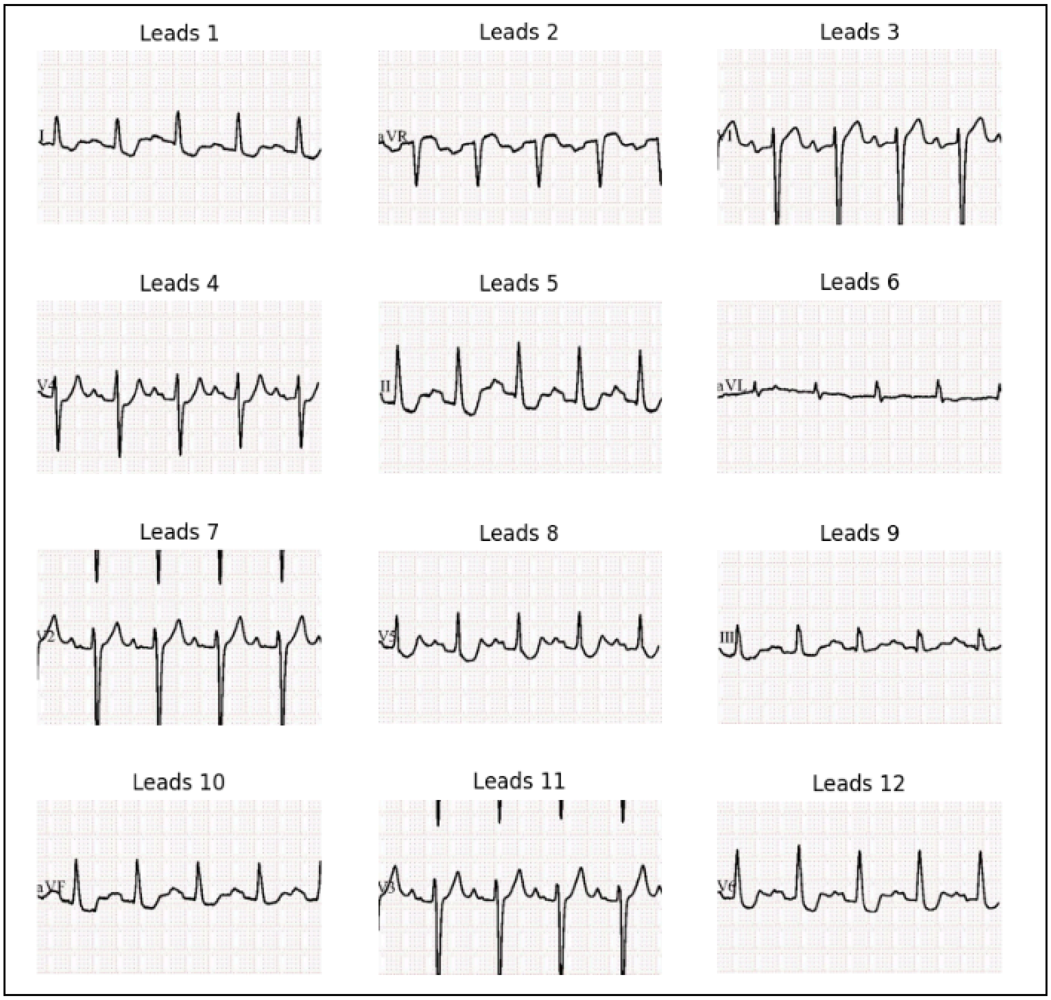


Fig 4

3) Leads Pre-processing

·This function focuses on further pre-processing each of the individually segmented leads to enhance the ECG signal and prepare it for feature extraction or analysis. The pre-processing steps involve:

Smoothing with Gaussian Filter: A Gaussian filter is applied to the lead image to blur or smoothen out noise and high-frequency components that might interfere with the ECG signal.

Otsu's Thresholding: This technique automatically determines a threshold value to separate the foreground (ECG signal) from the background in the image. Pixels with intensity values above the threshold are considered to represent the signal, while those below are considered background noise.

Binary Image Creation: Based on the Otsu's threshold, a binary image is created. This image has only two-pixel values: black (representing background) and white (representing the foreground or the ECG signal).

Image Resizing: The binary image is resized to a standard dimension (often 300x450 pixels) to ensure consistency in the size of the processed leads.

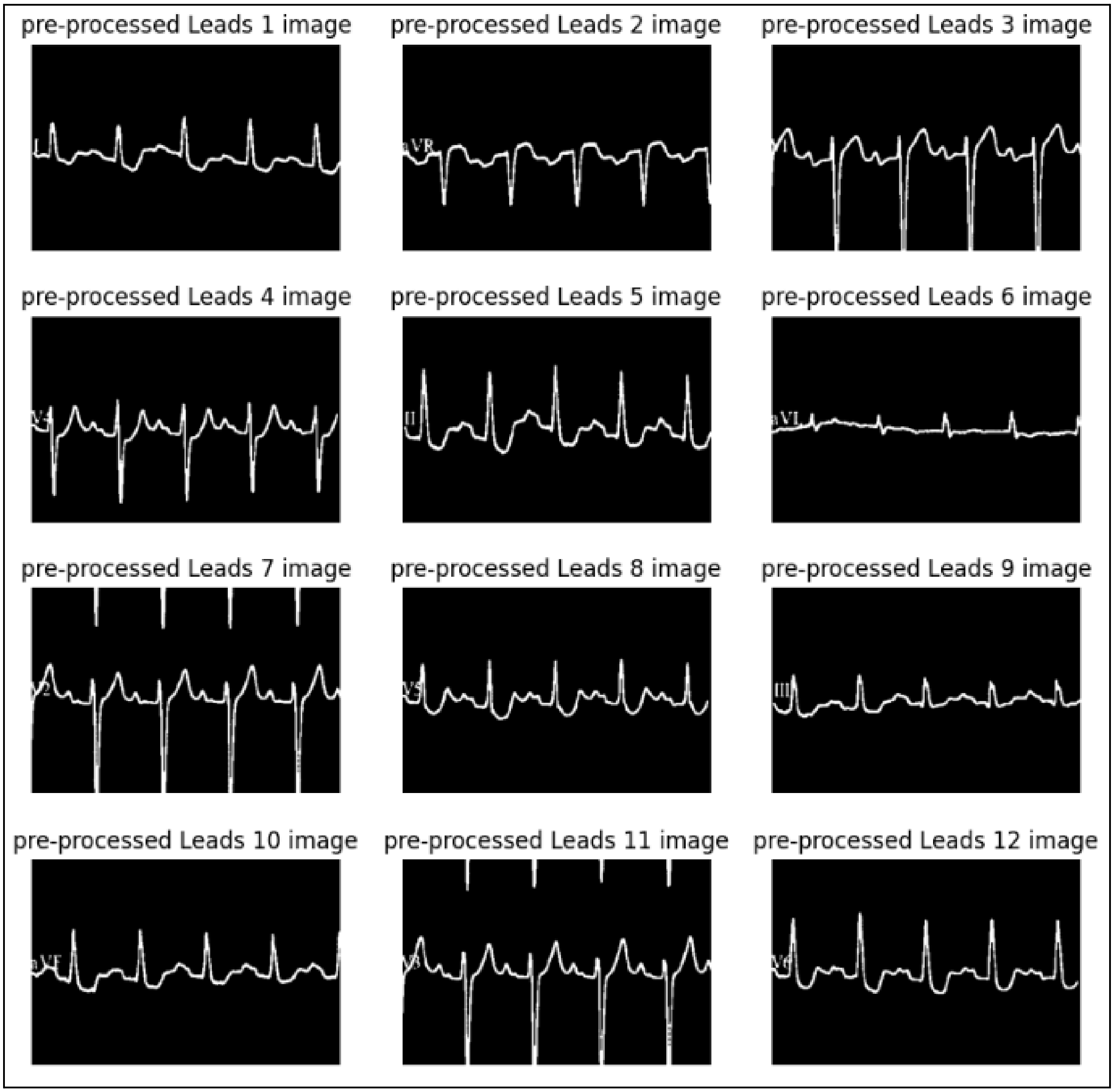


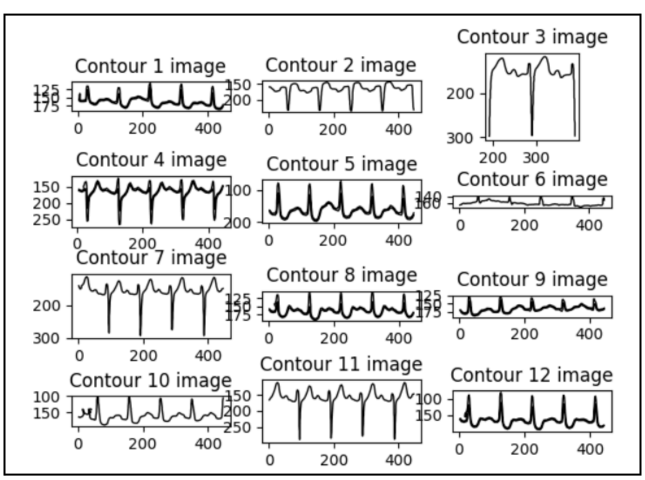
Fig 5

4) Signal Extraction and Scaling

This function employs contour detection algorithms to identify connected regions of foreground pixels within the binary image. Ideally, the largest and most prominent contour will correspond to the ECG signal, as it represents a continuous path of connected pixels with similar intensity levels.

Here the assumption is taken that the largest contour detected in the binary image represents the ECG signal. This assumption is valid under the condition that the pre-processing steps effectively isolated the signal and minimized the presence of other significant objects in the image.

- ❖ The extracted contour, representing the isolated ECG signal, is resized to a standard dimension. This standardization ensures consistency in the size of the extracted signal across different leads, facilitating further processing and analysis.
- ❖ The next step applies MinMaxScaler to the intensity values of the extracted signal data. MinMaxScaler scales the data to a specific range, typically between 0 and 1. This normalization is crucial for several reasons:
- ❖ Improved Model Performance: Many machine learning algorithms and analysis techniques perform better when the data features are on a similar scale. Normalization ensures that all features contribute equally during analysis, preventing features with larger scales from dominating the results.
- ❖ Enhanced Interpretability: When features are on a common scale, the analysis results become easier to interpret. The scaled values can directly represent the relative strength or intensity of the ECG signal at different points
- ❖ The function saves the scaled 1D signal data (representing the intensity values of the ECG signal) for each lead (excluding the long lead) as a separate CSV file. This creates a structured and easily accessible format for the model.



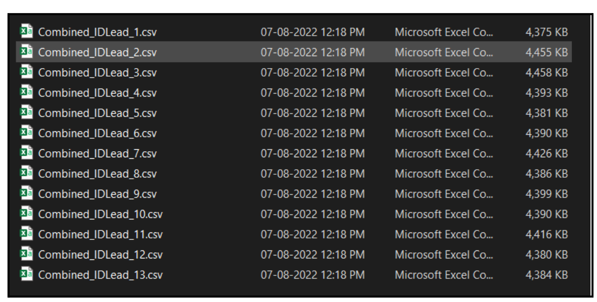


Fig 6 ,7

3.7 CHOOSING APPROPRIATE ALGORITHM FOR MODEL TRAINING

We have implemented various algorithms and we selected appropriate algorithms for model training. Here are the algorithms implemented.

1) **KNN(K - Nearest Neighbour):**

- ❖ The K-Nearest Neighbors (KNN) algorithm stands as a fundamental technique in machine learning, renowned for its simplicity and effectiveness in classification and regression tasks.
- ❖ When confronted with a new instance, KNN identifies the 'k' nearest neighbors from the training set based on a chosen distance metric.
- ❖ The hyperparameter 'k' serves as a critical determinant in KNN, dictating the number of neighbors to consider.
- ❖ Selection of an appropriate 'k' value significantly impacts the algorithm's performance, with smaller values leading to increased model complexity and potential overfitting, while larger 'k' values result in smoother decision boundaries and susceptibility to underfitting.

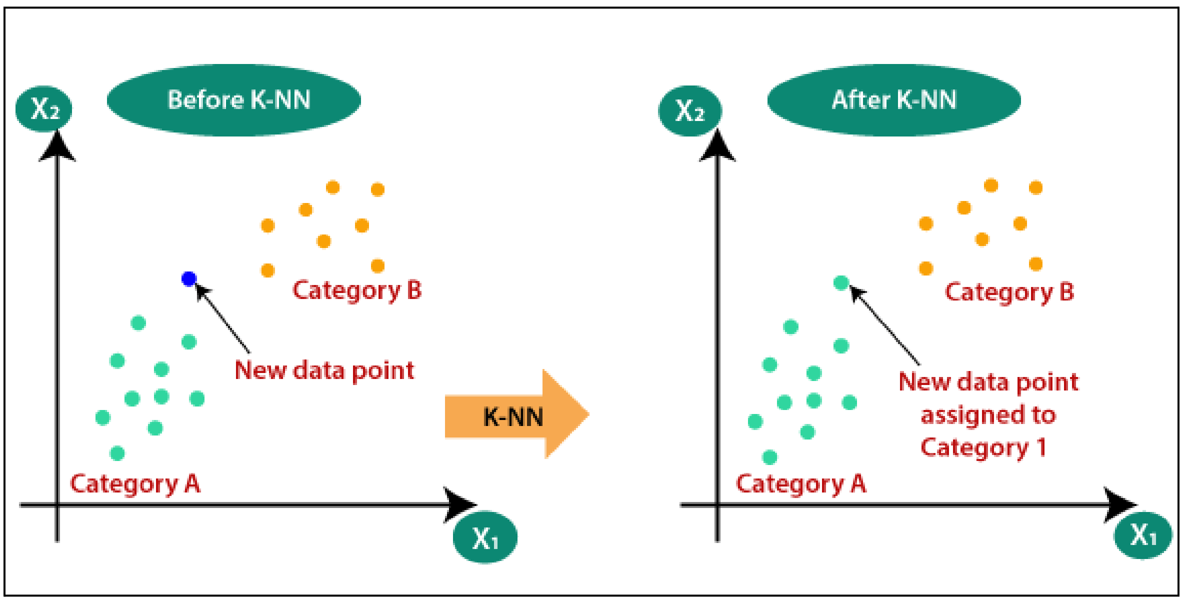


Fig 8

2) **Logistic Regression**

- ❖ Logistic Regression models the relationship between the independent variables (features) and the categorical dependent variable using the logistic function.
- ❖ The logistic function maps the weighted sum of the input features to a probability score, which is then interpreted as the likelihood of belonging to a particular class.
- ❖ During training, the model parameters (weights) are iteratively adjusted to maximize the likelihood of the observed labels given the input data, typically using optimization techniques like gradient descent.

- ❖ The decision boundary generated by Logistic Regression separates the feature space into regions corresponding to different class labels.
- ❖ Logistic Regression outputs probabilities rather than hard class labels. A threshold value (usually 0.5) is applied to these probabilities to make binary decisions
- ❖ To mitigate overfitting and improve generalization, Logistic Regression often incorporates regularization techniques like L1 or L2 regularization.

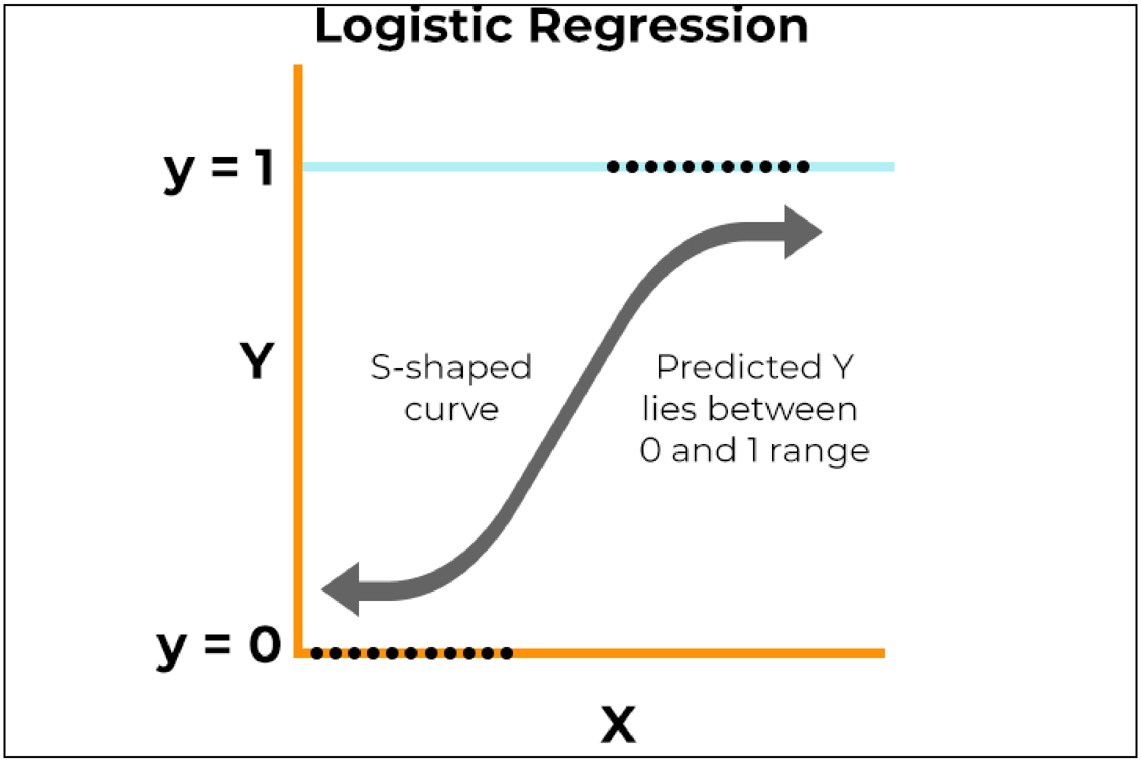


Fig 9

3) Support Vector Machine (SVM)

- ❖ Support Vector Machine (SVM) is a powerful supervised learning algorithm utilized for classification, regression, and outlier detection tasks.
- ❖ SVM operates by mapping the input data points into a high-dimensional feature space using a kernel function.
- ❖ The kernel trick enables SVM to efficiently handle non-linearly separable data by implicitly mapping it into a higher-dimensional space.
- ❖ SVM aims to find the hyperplane that not only separates the classes but also maximizes the margin between them.
- ❖ This margin maximization strategy leads to a robust decision boundary that generalizes well to unseen data and is less susceptible to overfitting.
- ❖ SVM incorporates regularization parameters (e.g., C parameter) to control the trade-off between maximizing the margin and minimizing classification errors.

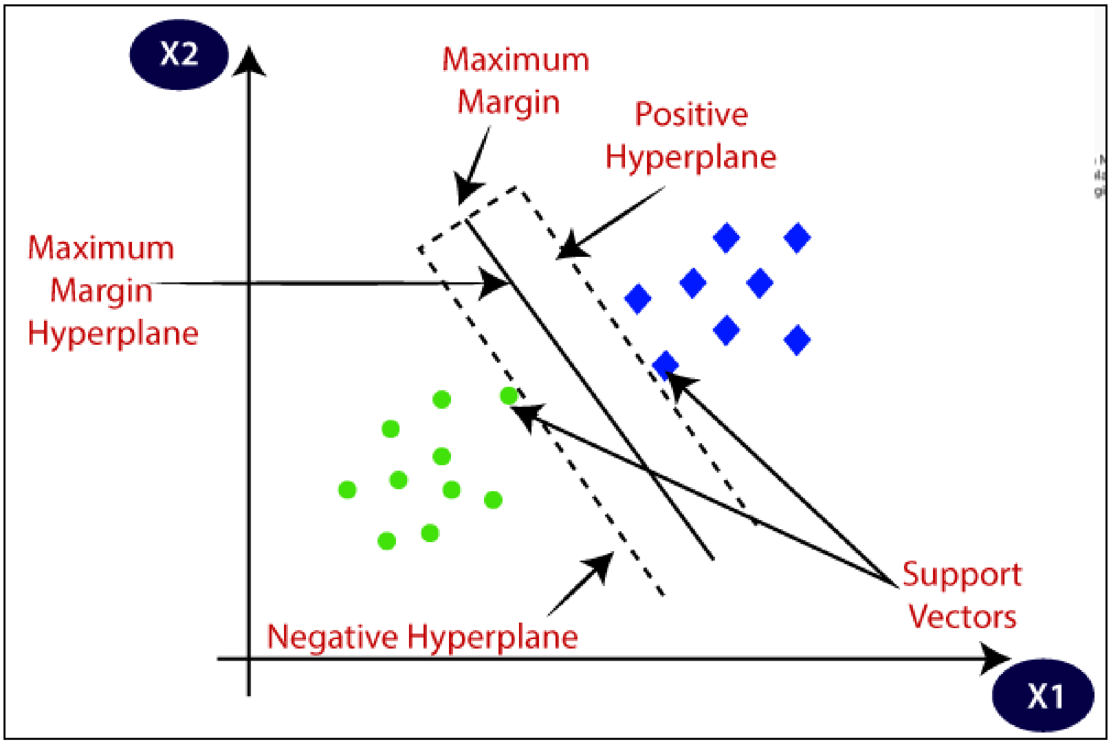


Fig 10

4) Random Forest

❖ Random Forest is an ensemble learning algorithm that operates by constructing a multitude of decision trees during training and outputting the mode (classification) or mean (regression) prediction of the individual trees.

❖ Random Forest builds each decision tree using a subset of the training data and a random selection of features at each node.

❖ During training, each tree is grown to its maximum depth without pruning, resulting in a collection of diverse and relatively uncorrelated trees.

❖ During prediction, the final output is determined by aggregating the predictions of all trees, typically through averaging (regression) or voting (classification).

❖ Key hyperparameters in Random Forest include the number of trees in the forest, the maximum depth of each tree, and the number of features to consider at each split.

❖ Tuning these hyperparameters is essential to optimize the model's performance and prevent overfitting.

❖ The ensemble approach improves robustness and generalization by reducing variance and mitigating overfitting.

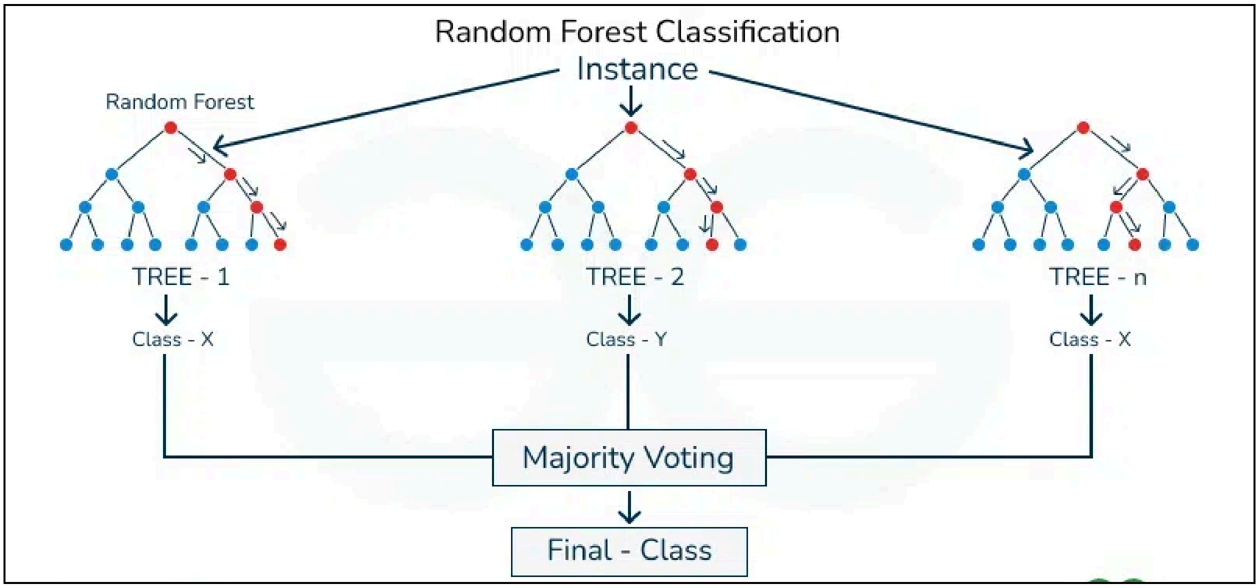


Fig 11

5) **Gaussian Naive Bayes (GNB)**

❖ Gaussian Naive Bayes (GNB) is a variant of the Naive Bayes algorithm, which is based on Bayes' theorem and assumes that features are conditionally independent given the class label.

❖ GNB calculates the probability of each class given a set of input features using Bayes' theorem, which states that the probability of a class given the features is proportional to the product of the probability of each feature given the class and the prior probability of the class.

❖ In GNB, the likelihood of each feature given the class is modeled as a Gaussian distribution, with parameters estimated from the training data (mean and variance for each feature in each class)

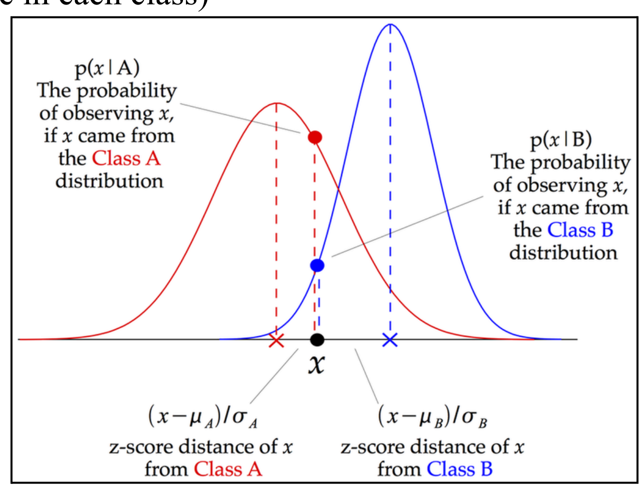


Fig 12

6) XGBoost (Extreme Gradient Boosting)

- ❖ XGBoost is an optimized implementation of gradient boosting machines, which are ensemble learning techniques renowned for their effectiveness in supervised learning tasks.
- ❖ XGBoost builds a predictive model by combining the predictions of multiple weak learners, typically decision trees, in an additive manner.
- ❖ During training, XGBoost sequentially adds decision trees to the ensemble, with each subsequent tree correcting the errors made by the previous ones.
- ❖ It employs a gradient boosting framework, optimizing an objective function by minimizing the loss function and adding new trees that minimize the residual errors.
- ❖ XGBoost provides valuable insights into feature importance through techniques like gain, cover, and frequency.

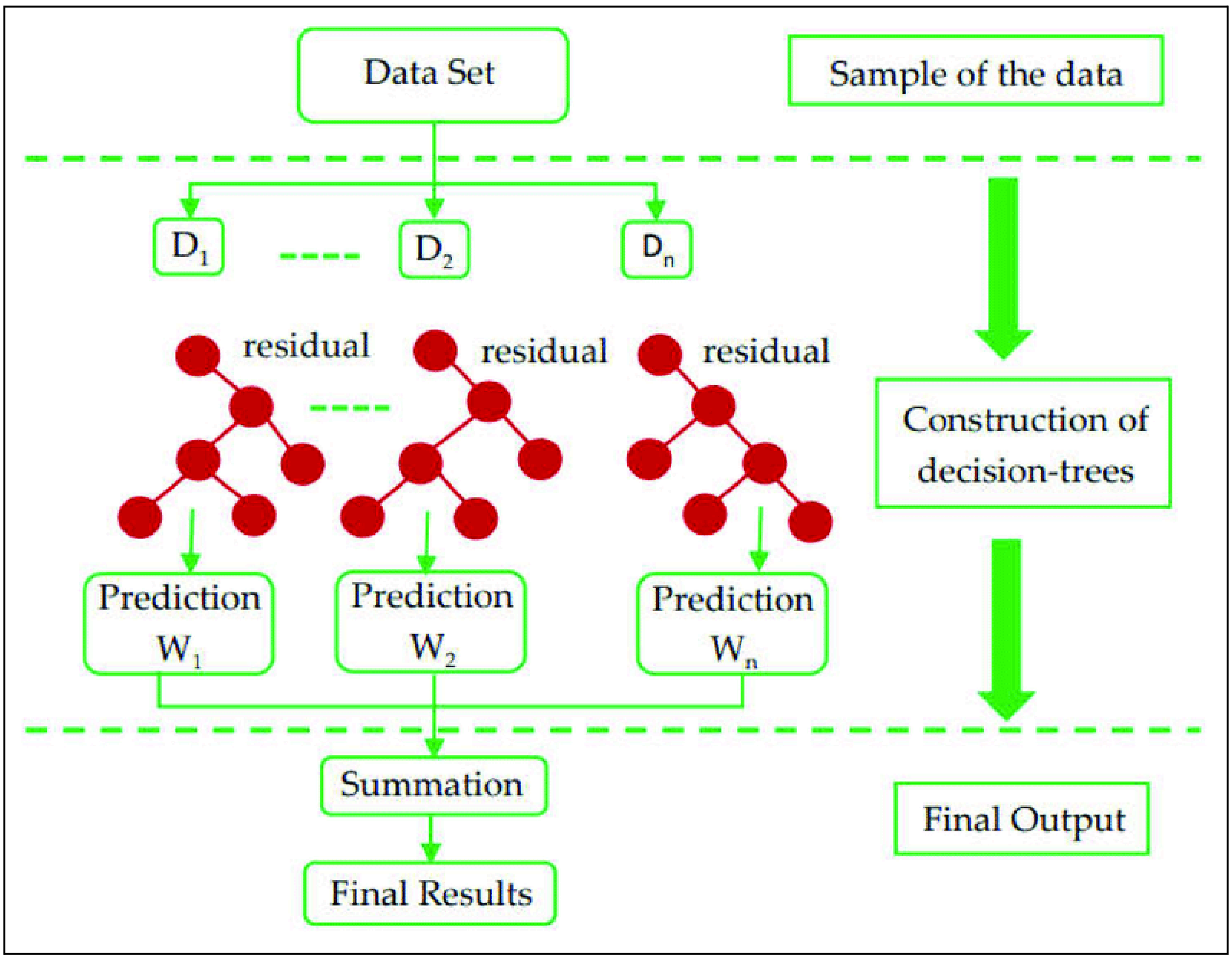


Fig 13

3.8 MODEL TRAINING

1) Using KNN

```
k_range = list(range(1, 30))
parameters = dict(knn__n_neighbors=k_range)

#input
X = final_result_df.iloc[:,:-1]

#target
y=final_result_df.iloc[:,-1]

# Create train and test sets
X_train, X_test, y_train, y_test = train_test_split(X,y,test_size=0.4,random_state=42)

#increasing cv score takes lot of time in gooogle colab, so kept it just 2.
cv = GridSearchCV(pipeline,parameters,cv=2)

cv.fit(X_train,y_train)

# Predict the labels of the test set: y_pred
y_pred = cv.predict(X_test)

Knn_Accuracy = cv.score(X_test, y_test)
```

Fig 14

2) Using Logistic Regression

```
#input
X = final_result_df.iloc[:,:-1]

#target
y=final_result_df.iloc[:,-1]

#parameters for gridsearchcv
c_space = np.logspace(-4, 4, 10)
parameters = {'lr__C': c_space,'lr__penalty': ['l2']}

# Create train and test sets
X_train, X_test, y_train, y_test = train_test_split(X,y,test_size=0.4,random_state=42)

#call GridSearchCV and set crossvalscore to 2
cv = GridSearchCV(pipeline,parameters,cv=2)

cv.fit(X_train,y_train)

# Predict the labels of the test set: y_pred
y_pred = cv.predict(X_test)
LR_Accuracy = cv.score(X_test, y_test)
```

Fig 15

3) Using SVM

```
#input
X = final_result_df.iloc[:,:-1]

#target
y=final_result_df.iloc[:,-1]

# Specify the hyperparameter space, if we increase the penalty(c) and gamma value the accurancy can be increased.
#since it takes lots of time in google colab provided only a single value
parameters = {'SVM__C':[1, 10, 100],
              'SVM__gamma':[0.1, 0.01]}

# Create train and test sets
X_train, X_test, y_train, y_test = train_test_split(X,y,test_size=0.5,random_state=21)

cv = GridSearchCV(pipeline,parameters,cv=3)
cv.fit(X_train,y_train)

y_pred = cv.predict(X_test)
SVM_Accuracy = cv.score(X_test, y_test)

# Compute and print metrics
SVM_Accuracy=cv.score(X_test, y_test)
```

Fig 16

4) Using XGBoost

```
from xgboost import XGBClassifier
from sklearn.metrics import accuracy_score

model = XGBClassifier()
model.fit(X_train, y_train)

y_pred = model.predict(X_test)
predictions = [round(value) for value in y_pred]

# evaluate predictions
accuracy = accuracy_score(y_test, predictions)
```

Fig 17

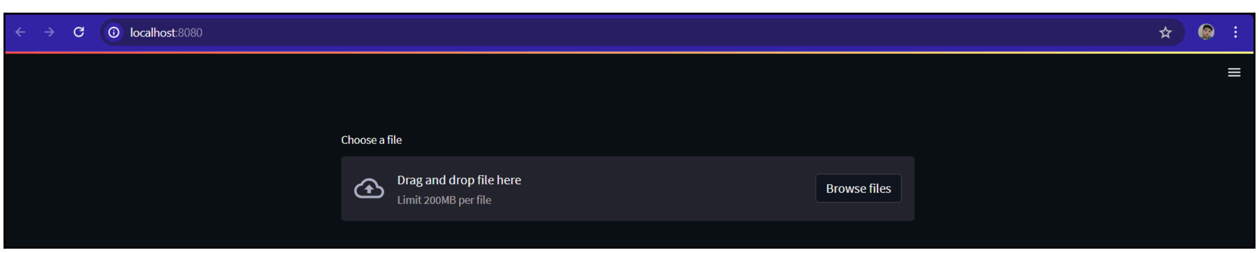
❖ Ensemble all Models

```
eclf = VotingClassifier(estimators=[
    ('SVM', SVC(probability=True)),
    ('knn', KNeighborsClassifier()),
    ('rf', ensemble.RandomForestClassifier()),
    ('bayes',GaussianNB()),
    ('logistic',LogisticRegression()),
    ], voting='soft')

# Hyperparameter Tuning using gridSearch
params = {'SVM__C':[1, 10, 100],
          'SVM__gamma':[0.1, 0.01],
          'knn__n_neighbors': [1,3,5],
          'rf__n_estimators':[300, 400],
          }

grid = GridSearchCV(estimator=eclf, param_grid=params, cv=5)
voting_clf = grid.fit(X_train, y_train)

print(grid.best_params_)
y_pred = voting_clf.predict(X_test)

# Compute and print metrics
Voting_Accuracy=voting_clf.score(X_test, y_test)
```

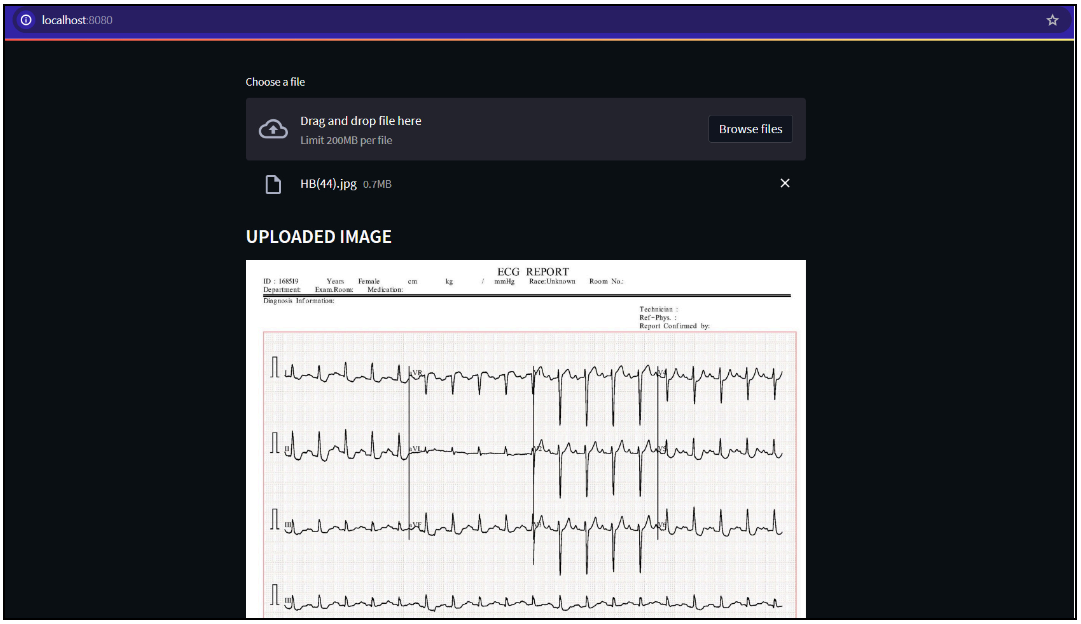
Fig 18

3.9 Integration of Frontend with Backend

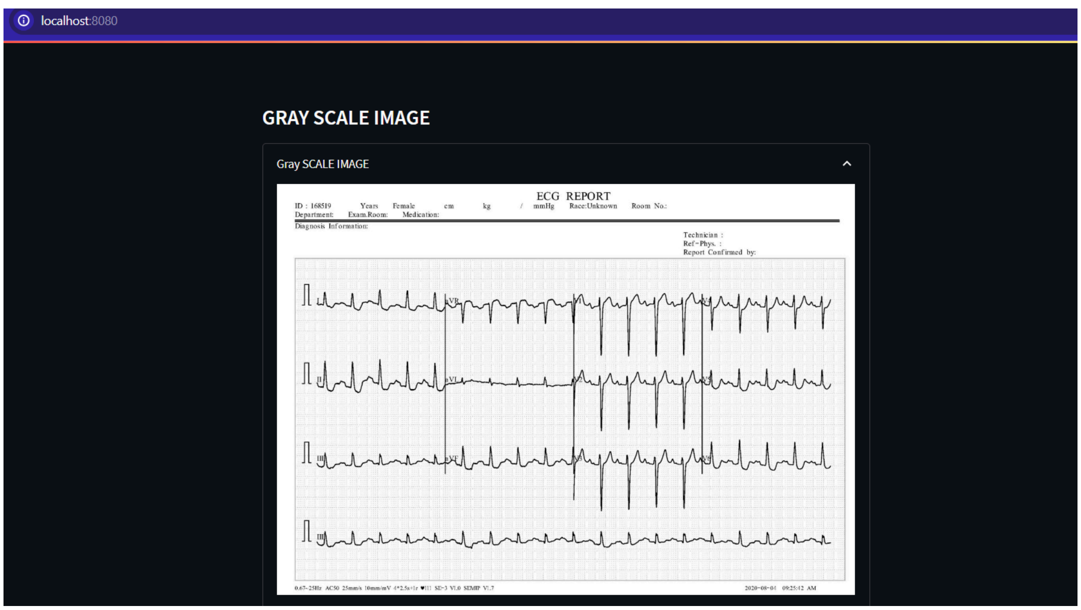
Simplified Framework: This project utilizes Streamlit, a Python library known for its ease of use in building web apps. Compared to Flask, Streamlit requires less code for creating the user interface, making development faster.
Model Loading: Similar to the Flask app, pre-trained models (scaler and heart disease classifier) are loaded using joblib. These models handle data normalization and prediction.
Data Preparation Function: The predict_heart_disease function encapsulates user input processing. It converts the data into a NumPy array, applies the loaded scaler for normalization, and makes a prediction using the loaded model.
Prediction Trigger: Clicking the button created with st.button initiates the prediction process based on the user-provided data.
Conditional Output: The code checks the prediction outcome. It shows the tentative prediction of ECG Image . Otherwise, a warning message is shown using st.warning.
Flexibility: Streamlit allows for easy customization. You can modify the app by adding more informative messages based on the predicted stage or incorporating a section to display past predictions.
Frontend Integration: Unlike Flask, Streamlit doesn't directly utilize HTML templates. The user interface is built entirely within the Python script using Streamlit functions.
Focus on Logic: This Streamlit approach prioritizes the core logic of data processing, prediction, and result display. While offering less UI flexibility compared to Flask with HTML templates, Streamlit simplifies development.



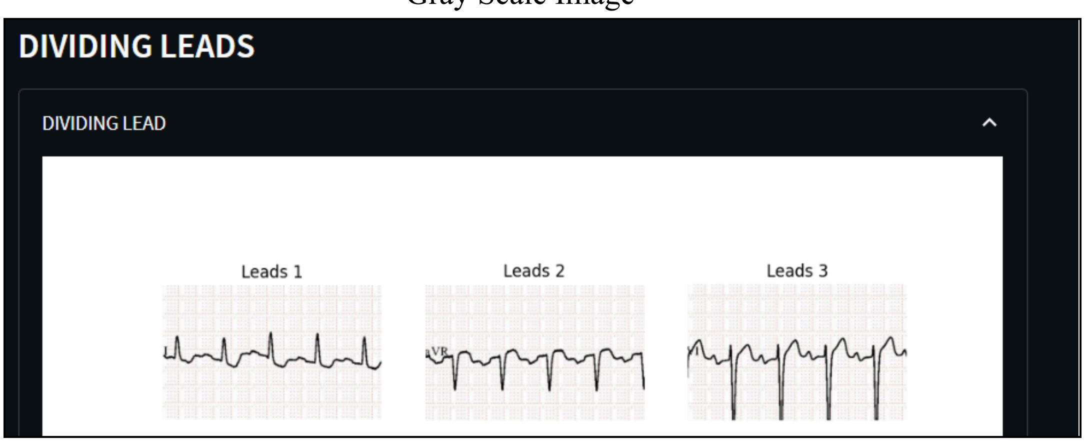
Basic Input Prompt to upload ECG Image

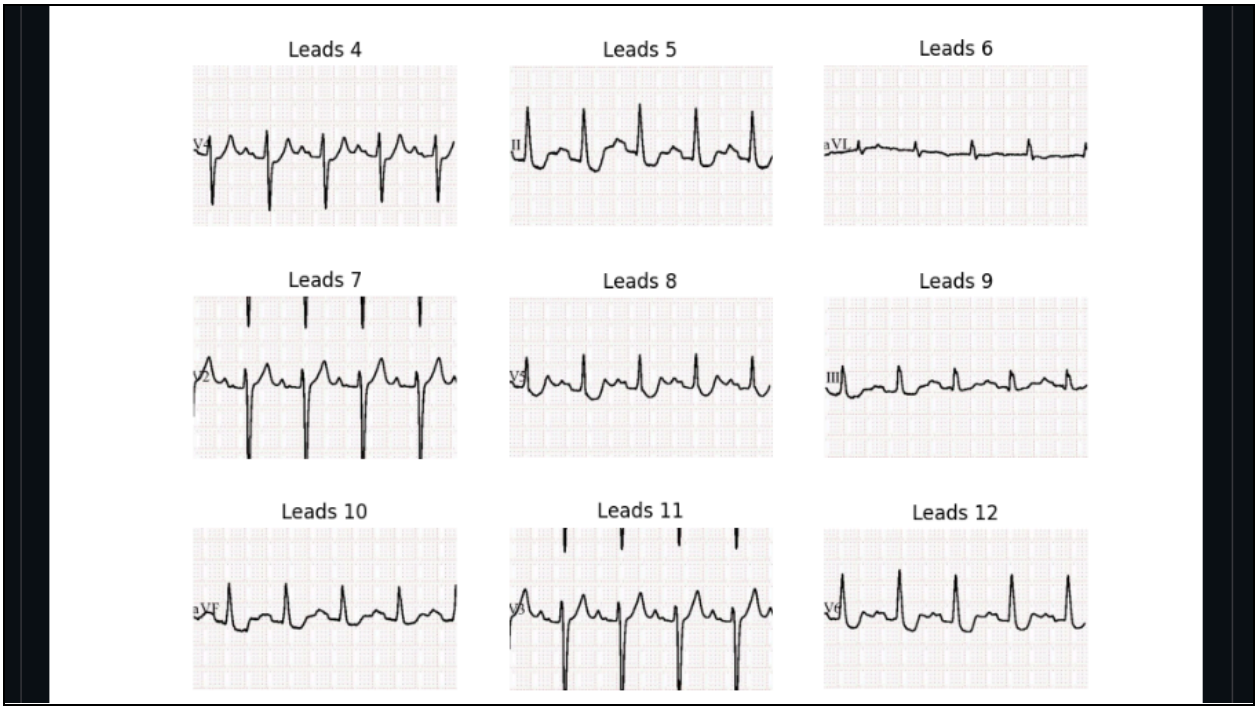


Uploaded Image



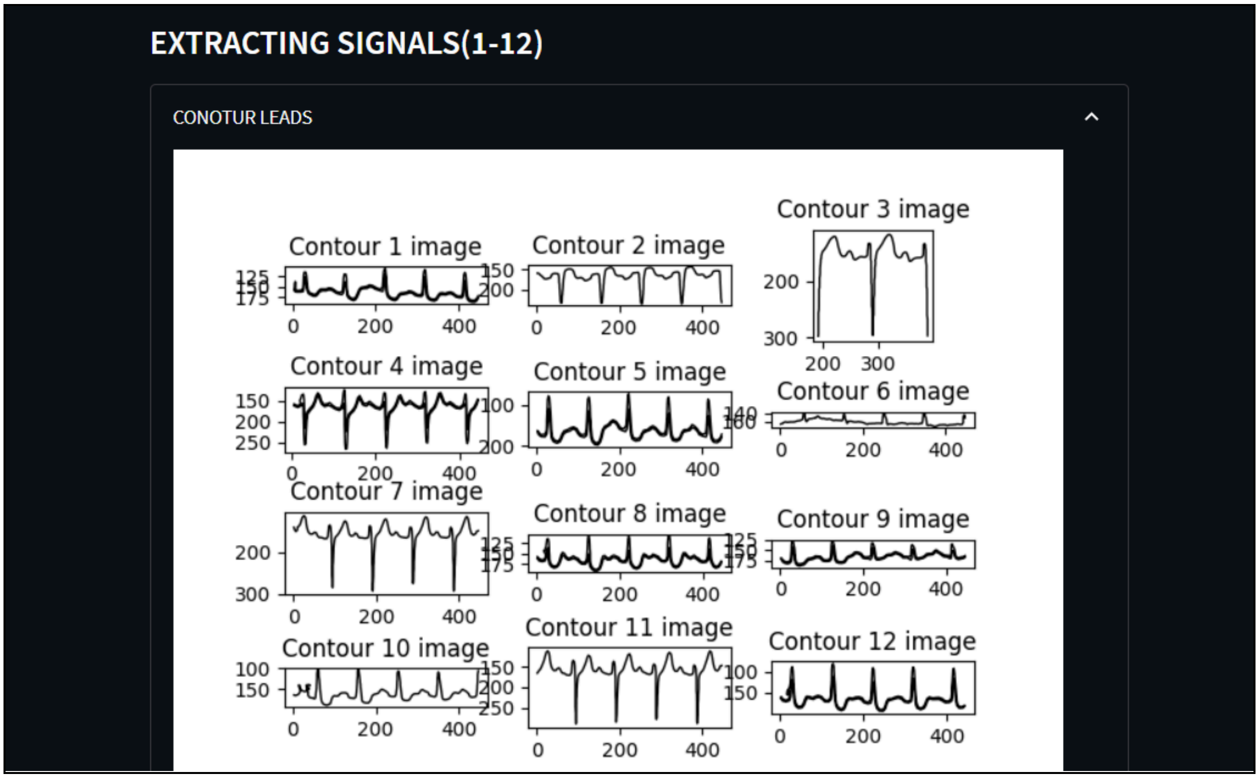
Gray Scale Image



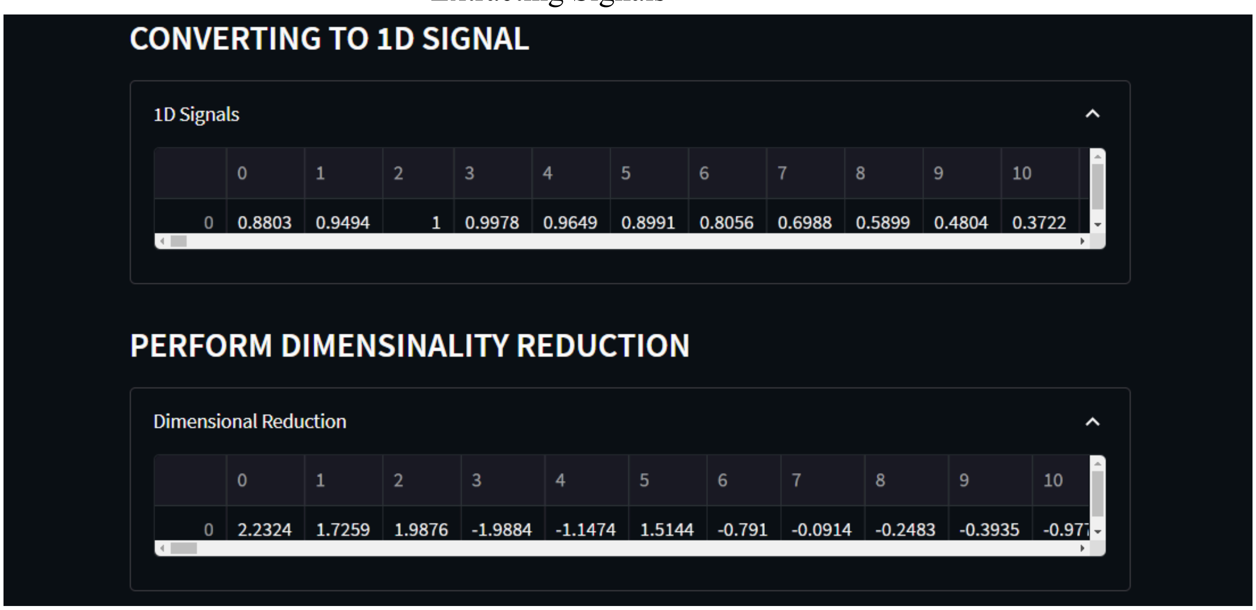


Dividing Leads

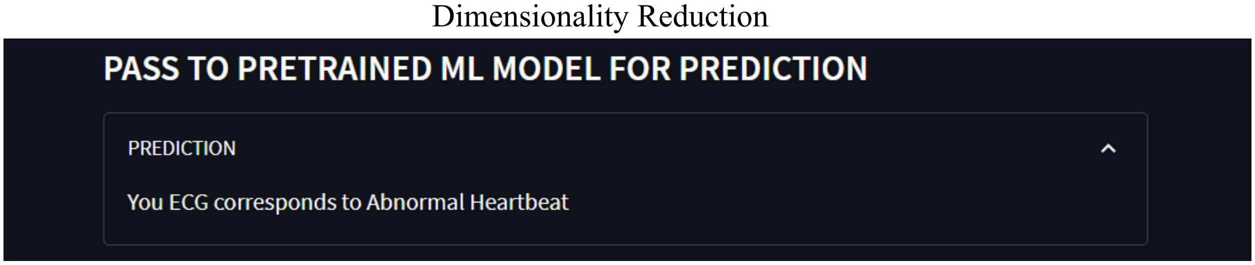
Preprocessing of Leads



Extracting Signals



Dimensionality Reduction



CHAPTER 4: EXPERIMENTAL ANALYSIS

4.1 ACCURACY ASSESSMENT

1) POST DIMENSIONALITY REDUCTION KNN :-

Accuracy: 0.782258064516129

	precision	recall	f1-score	support
0	0.87	0.63	0.73	105
1	0.91	0.91	0.91	94
2	0.72	0.88	0.79	112
3	0.63	0.67	0.65	61
accuracy			0.78	372
macro avg	0.78	0.77	0.77	372
weighted avg	0.80	0.78	0.78	372

Tuned Model Parameters: {'knn__n_neighbors': 1}

Type *Markdown* and LaTeX: α^2

Accuracy Achieved :0.7822580

Percentage Accuracy : 78.2%

2) LOGISTIC REGRESSION

Accuracy: 0.543010752688172

	precision	recall	f1-score	support
0	0.36	0.33	0.35	105
1	0.73	0.91	0.81	94
2	0.56	0.58	0.57	112
3	0.38	0.26	0.31	61
accuracy			0.54	372
macro avg	0.51	0.52	0.51	372
weighted avg	0.52	0.54	0.53	372

Tuned Model Parameters: {'lr__C': 10000.0, 'lr__penalty': 'l2'}

Accuracy Achieved :0.5430107

Percentage Accuracy : 54.3%

3) SVM (SUPPORT VECTOR MACHINE)

Accuracy: 0.8225806451612904

	precision	recall	f1-score	support
0	0.58	1.00	0.74	93
1	1.00	1.00	1.00	99
2	1.00	0.61	0.76	117
3	1.00	0.68	0.81	63
accuracy			0.82	372
macro avg	0.90	0.82	0.83	372
weighted avg	0.90	0.82	0.83	372

Accuracy Achieved :0.8225806

Percentage Accuracy : 82.3%

4) XG-BOOST

```
Accuracy: 0.853448275862069
              precision    recall  f1-score   support

           0       0.79      0.70      0.74       119
           1       0.98      1.00      0.99       125
           2       0.82      0.87      0.84       140
           3       0.80      0.82      0.81        80

    accuracy                           0.85       464
   macro avg       0.85      0.85      0.85       464
weighted avg       0.85      0.85      0.85       464
```

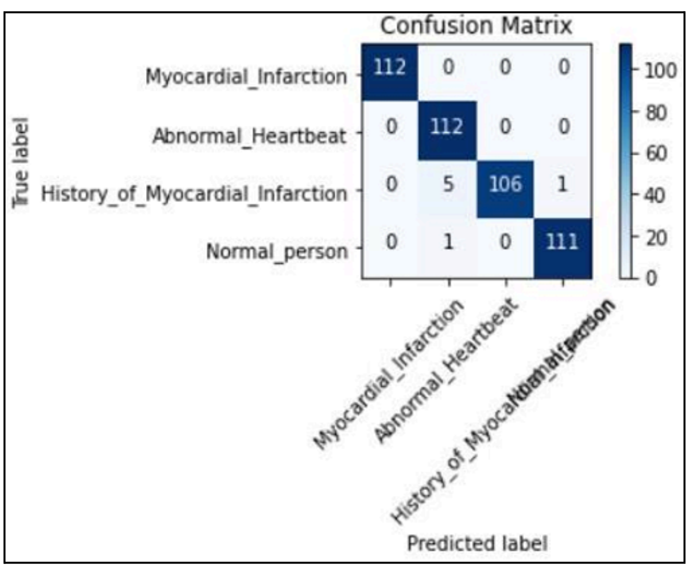
Accuracy Achieved :0.8534482
Percentage Accuracy : 85.34%

5) AFTER ENSEMBLE ALL MODELS

```
{'SVM__C': 1, 'SVM__gamma': 0.1, 'knn__n_neighbors': 1, 'rf__n_estimators': 300}
Accuracy: 0.9247311827956989
              precision    recall  f1-score   support

           0       0.89      0.96      0.92        80
           1       1.00      1.00      1.00        72
           2       0.92      0.92      0.92        79
           3       0.88      0.75      0.81        48

    accuracy                           0.92       279
   macro avg       0.92      0.91      0.91       279
weighted avg       0.92      0.92      0.92       279

{'SVM__C': 1, 'SVM__gamma': 0.1, 'knn__n_neighbors': 1, 'rf__n_estimators': 300}
```

FINAL ACCURACY ACHIEVED :- 0.9247311
FINAL PERCENTAGE ACCURACY :- 92.47%

4.2 CONFUSION MATRIX



The study of the confusion matrix for each of the four dataset types that we took into account in our particular scenario, the cardiac patient ECG image dataset as follows.

Fig 19

4.3 PERFORMANCE ANALYSIS

1) Accuracy achieved with the K nearest Neighbor algorithm tested after post dimensionality reduction is 78.2%.
2) Accuracy achieved with the Logistic Regression algorithm tested after training the model is 54.3% with precision of 0.52.
3) Accuracy achieved with the Support Vector Machine algorithm tested on the model is 82.3% with precision around 0.90.
4) Accuracy achieved with the XGBOOST algorithm tested on the model is 85.34% with precision around 0.79.
5) After Ensemble all the model stack and Hyperparameter tuning the final accuracy achieved is 92.47% with weighted average of 0.92.
6) Therefore it can be said that the system is 92 % accurate while predicting results from analyzing ECG images uploaded by the user.

CHAPTER 5: FUTURE WORK AND CONCLUSION

5.1 FUTURE SCOPE

5.1.1 Migrating To A Flask Based Web App

- **Flask** is a lightweight and versatile web framework for Python, designed to create web applications quickly and efficiently. It follows a simple and minimalistic approach, offering essential tools for building web applications without imposing strict architectural constraints. With its modular structure and extensive ecosystem of extensions, Flask enables developers to create scalable and customizable web solutions, making it popular for projects ranging from simple APIs to complex web applications.

- **Web Application Structure:** The Flask-based web application is structured with routes for different pages. It consists of routes for the home page ('/'), the result page ('/result'), and an about page ('/about'). Each route is associated with a specific function that renders the corresponding HTML template using the render_template function.

- **Machine Learning Integration:** The web application integrates a machine learning model for predicting heart disease. It loads a trained Random Forest Classifier (rfc.sav) using the joblib library. The input data from the user's form submission is preprocessed using a scaler loaded from a pickle file (scaler.pkl). The model predicts the likelihood of heart disease based on the user's input features.

- **Dynamic Rendering:** Depending on the prediction result, the application dynamically renders different HTML templates. If the model predicts no heart disease, it renders the 'nodisease.html' template. Otherwise, it renders the 'heartdisease.htm' template, passing the predicted stage of heart disease to be displayed to the user. This dynamic rendering enhances user experience by providing tailored feedback based on the prediction outcome.

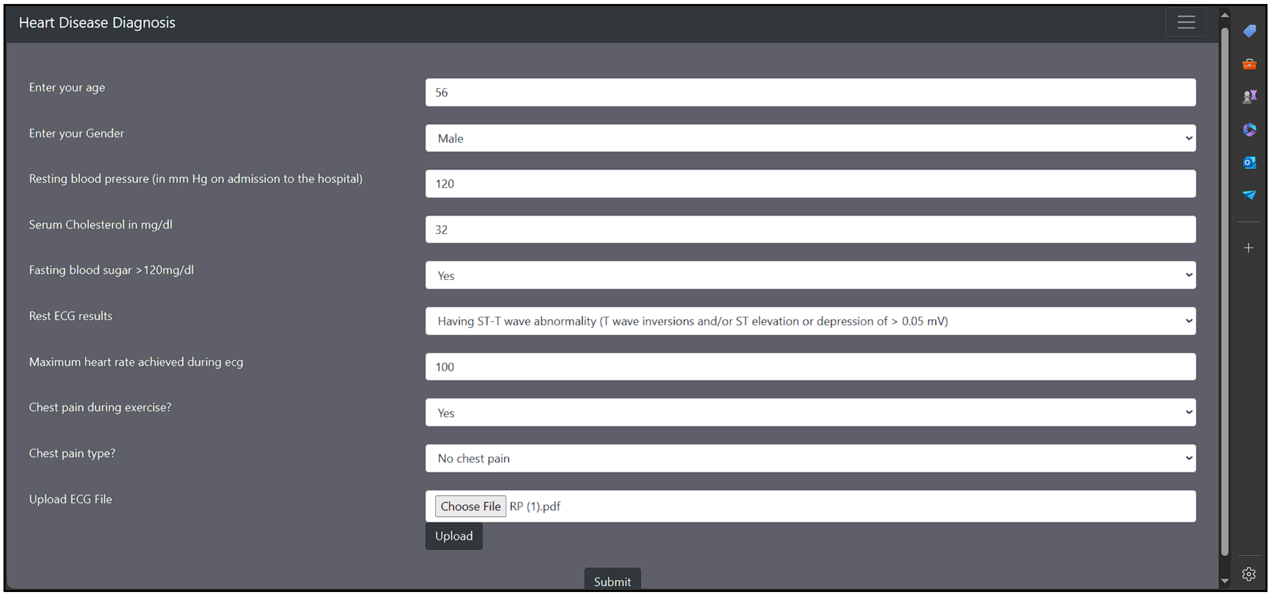


Fig 1: User Input Interface

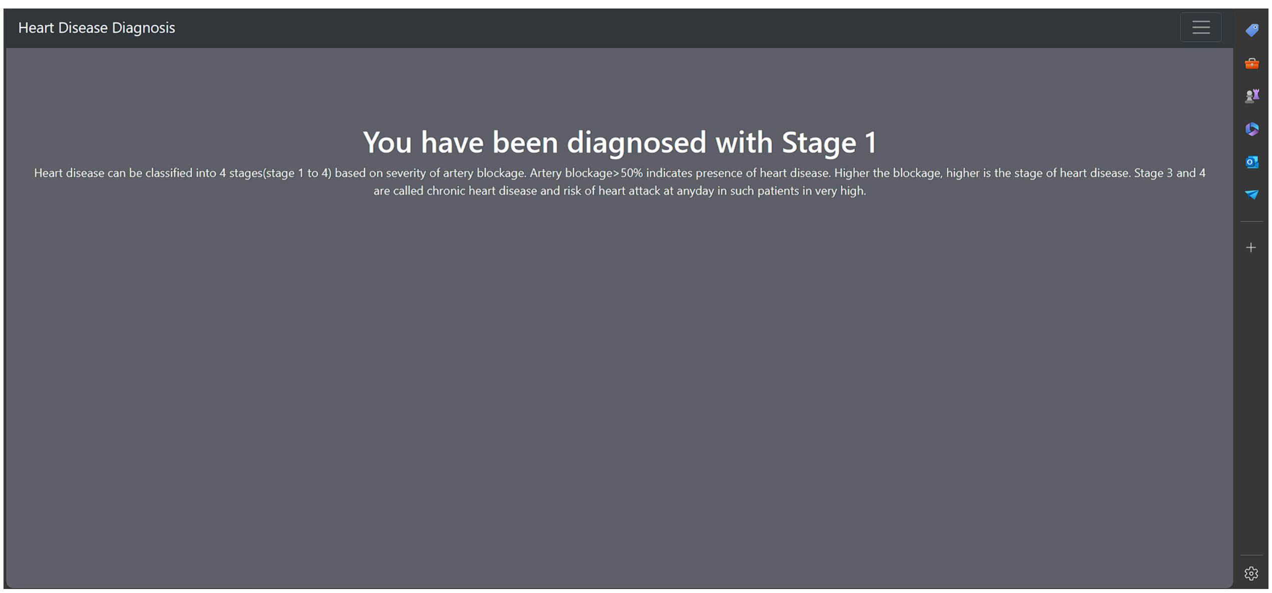


Fig 2: Prediction Result

5.1.2 Using A Mysql Database To Store Data

- **MySQL** is a popular open-source relational database management system (RDBMS) that stores data in a structured way using tables. This makes it easy to access, manage, and manipulate large amounts of information. MySQL is known for its speed, reliability, and ease of use, making it a great choice for both small and large applications. Additionally, it's compatible with many programming languages and operating systems.
- **Data Extraction and Transformation**: Periodically, a tool can extract the relevant data from the MySQL database. This tool could be a custom script or a third-party data integration platform. The extracted data may also be transformed into a format suitable for Snowflake.
- **Data Loading into Snowflake:** The extracted and transformed patient data is then loaded into Snowflake. Snowflake's cloud-based architecture and processing power can handle large data volumes efficiently.
- **Webapp Data Access:** The web app can be designed to query data from Snowflake to dynamically populate its content. This involves storing the patient information and their reports on the cloud database.

5.1.3 Addition Of A Cloud Database To Web App Using Sqlalchemy

- **SQLAlchemy** is the Python SQL toolkit and Object Relational Mapper that gives application developers the full power and flexibility of SQL.It provides a full suite of well known enterprise-level persistence patterns, designed for efficient and high-performing database access, adapted into a simple and Pythonic domain language.
- **Data Persistence and Retrieval:** Within the application, functions were implemented to interact with the database using SQL Alchemy's ORM. This included inserting new patient records, querying existing data for analysis, and updating or deleting records as needed. SQL Alchemy's query language constructs facilitated complex database operations, enhancing data management capabilities.
- **Integration with Visualization:** SQL Alchemy was seamlessly integrated with Matplotlib to visualize patient data directly from the database. By retrieving relevant data using SQLAlchemy queries, Matplotlib was able to generate informative graphs, charts, and statistical analyses. This integration streamlined the process of deriving visual insights from stored patient reports, facilitating data-driven decision-making in healthcare research or clinical practice

5.2 CONCLUSION

- ❖ In this project, we have developed a comprehensive pipeline for electrocardiogram (ECG) analysis, aiming to assist healthcare professionals in efficiently diagnosing cardiac conditions. Leveraging the power of Python and libraries like Streamlit and scikit-image, we have created an intuitive web application for ECG processing and classification.
- ❖ Our system begins by allowing users to upload ECG images through a user-friendly interface built with Streamlit. Once uploaded, the image undergoes a series of preprocessing steps, including conversion to grayscale, smoothing, and thresholding to enhance signal clarity. The ECG image is then segmented into individual leads, facilitating focused analysis.
- ❖ Each lead is subjected to further processing to extract and scale the ECG signal. Through techniques such as contouring and normalization, we obtain one-dimensional representations of the ECG signals. These signals are combined into a cohesive dataset for dimensional reduction using Principal Component Analysis (PCA), reducing computational complexity while preserving essential information.
- ❖ Finally, the preprocessed signals are fed into a pre-trained machine learning model for classification. Our model, trained on a diverse dataset of ECG signals, can accurately classify ECGs into categories such as myocardial infarction, abnormal heartbeat, normal, or history of myocardial infarction. This classification aids clinicians in making informed decisions regarding patient care.
- ❖ The strength of our system lies in its versatility and efficiency. By automating ECG analysis, we reduce the burden on healthcare professionals, allowing them to focus more on patient care. Additionally, our web-based interface makes the tool accessible from anywhere, facilitating remote diagnosis and consultation.
- ❖ In conclusion, this project represents a significant step towards enhancing the efficiency and accuracy of ECG analysis through automation and machine learning. As we continue to refine and expand our model with larger datasets and advanced techniques with a database integration and migration to a more powerful framework such as Flask, we envision it becoming an indispensable tool in the realm of cardiovascular healthcare.

APPENDIX

❖ Libraries and Modules

Sr.no	Libraries	Modules
1)	Numpy	asarray
2)	Pandas	read_csv
3)	skimage.segmentation	slic
4)	skimage.filters	threshold_otsu,gaussian
5)	scipy.ndimage	ndimage
6)	sklearn.preprocessing	MinMaxScaler
7)	sklearn.decomposition	PCA
8)	Matplotlib	matplotlib.pyplot
9)	sklearn.neighbors	KNeighborsClassifier
10)	sklearn.model_selection	train_test_split,GridSearchCV
11)	sklearn.metrics	confusion_matrix,classification_report
12)	sklearn.linear_model	LogisticRegression
13)	sklearn.svm	SVC
14)	xgboost	XGBClassifier
15)	sklearn.metrics	confusion_matrix, classification_report
16)	sklearn	linear_model, tree, ensemble
17)	sklearn.ensemble	VotingClassifier
18)	sklearn.naive_bayes	GaussianNB
19)	sklearn.metrics	accuracy_score

❖ List of Figures

❖ References

1. https://www.ncbi.nlm.nih.gov/pmc/articles/PMC8898839/
2. https://www.nature.com/articles/s41597-020-0386-x
3. https://ieeexplore.ieee.org/document/9848279
4. https://ieeexplore.ieee.org/document/4493524
5. https://www.researchgate.net/publication/317821702_Application_of_Deep_Convolutional_Neural
6. _Network_for_Automated_Detection_of_Myocardial_Infarction_Using_ECG_Signals https://www.ncbi.nlm.nih.gov/pmc/articles/PMC9206502/
7. https://journalofbigdata.springeropen.com/articles/10.1186/s40537-023-00817-1
8. https://data.mendeley.com/datasets/gwbz3fsgp8/2