Cardiac Arrhythmia Prediction using ML Algorithms

Submitted in partial fulfillment of the requirements

of the degree

BACHELOR OF ENGINEERING

IN INFORMATION TECHNOLOGY

By

Sakshi Shete 20101B0022

Supervisor

Prof. Suvarna Udgire

Vidyalankar Institute of Technology

Accredited A+ by NAAC

Department of Information Technology

Vidyalankar Institute of Technology

Vidyalankar Educational Campus,

Wadala(E), Mumbai - 400 037

University of Mumbai

(AY 2023-24)

CERTIFICATE

This is to certify that the Internship Project entitled **"Predicting Cardiac Arrhythmia using ML Algorithms"** is a bonafide work of **Sakshi Shete** submitted to the University of Mumbai in partial fulfillment of the requirement for the award of the degree of **"Bachelor of Engineering"** in **"Information Technology".**

Prof. Suvarna Udgire
Supervisor

Dr. S. A. Patekar
Principal

Contents

Abstract ii

Acknowledgments iii

List of Figures iv

1 Introduction 1

- **1.1** Introduction
- **1.2** Motivation
- **1.3** Problem Statement & Objectives
- **1.4** Organization of the Report

2 Literature Survey 11

- **2.1** Survey of Existing/Similar System
- **2.2** Limitation Existing/Similar system or research gap

3 Proposed System (eg New Approach of Data Summarization) 14

- **3.1** Introduction
- **3.2** Details of Hardware & Software
- **3.3** Experiment and Results
- **3.4** Conclusion and Future work

References 27

Abstract

Cardiac arrhythmia is a prevalent heart disorder characterized by irregular heartbeats, potentially leading to life-threatening complications. Early and accurate detection of arrhythmia is vital for timely intervention and improved patient outcomes. In this project, we propose a predictive approach to diagnose cardiac arrhythmia using machine learning algorithms.

The dataset used in this study comprises a comprehensive collection of clinical features, including electrocardiogram (ECG) measurements and patient demographics. After preprocessing the data, various machine learning algorithms, such as Random Forest, Logistic Regression, and Decision Tree Classifier, are employed to build predictive models for arrhythmia classification.

The models are trained and evaluated using standard performance metrics to assess their accuracy, sensitivity, specificity, and other relevant measures. Additionally, feature importance analysis is conducted to identify the most influential indicators for arrhythmia prediction.

The results demonstrate the effectiveness of the proposed approach, achieving high accuracy in identifying cardiac arrhythmia cases. The predictive models provide a valuable tool for clinicians to make more informed decisions, enabling early diagnosis and appropriate intervention to improve patient care and outcomes.

Overall, this project showcases the potential of machine learning algorithms in aiding the diagnosis of cardiac arrhythmia, contributing to the advancement of cardiac healthcare and promoting personalized treatment strategies.

Acknowledgments

We surely took efforts for this project. But it have been possible just due to the kind support and help of many individuals. We would like to extend our sincere thanks to all of them. We are highly grateful to **Vidyalankar Institute of Technology** for their guidance and effective supervision as well as for providing required information regarding the project and also for their constant support in successfully completing the project.
We would like to express our gratitude towards our parents and our Guide Professor Suvarna Udgire for her kind co-operation and
encouragement which helped us in completion of this project.

List of figures

SR No.	Figure Name	Page No.
1	Dropping unneeded columns	16
2	Checking null values	17
3	Found no null values	18
4	Plotted Histograms	20
5	Cat Plot(Sex Vs Age)	20
6	Bar Plot (Height Vs Sex)	21
7	Count Plot (Sex Vs Whole data)	21
8	Bar Plot (Sex Vs Weight)	22
9	Bar Plot (Sex Vs s_wave)	22
10	Categorizing the different types of arrhythmia as true or false	23
11	Using Min Max Scaler to shrink the data	23
12	Logistic Regression	24
13	Decision Tree Classifier	24
14	Random Forest Classifier	24
15	Checking for X_train, X_test and Y_train, Y_test	25
16	Accuracy and AUC ROC Score	25

Chapter 1
Introduction

1.1 Introduction

Cardiac arrhythmia, a prevalent cardiac disorder, is characterized by irregular heart rhythms, posing a significant risk to human health and well-being. The timely and accurate identification of arrhythmia cases is crucial for effective medical intervention and improved patient outcomes. In recent years, advancements in machine learning (ML) algorithms have provided promising avenues for enhancing diagnostic capabilities in various medical domains, including cardiology.

This project aims to leverage the power of ML algorithms to develop a predictive approach for diagnosing cardiac arrhythmia. By harnessing a comprehensive dataset comprising a diverse set of clinical features, including electrocardiogram (ECG) measurements and patient demographics, we seek to build robust predictive models capable of accurately classifying arrhythmia cases.

The project entails several key steps: data preprocessing, algorithm selection, model training, and performance evaluation. The dataset is carefully prepared to ensure data quality and consistency, and relevant features are selected to capture essential information for arrhythmia detection. Subsequently, machine learning algorithms, such as Random Forest, Logistic Regression, and, Decision Tree Classifier are deployed to create predictive models.

The models are trained on a subset of the data, with performance evaluated on a separate test set. Various performance metrics, such as accuracy, sensitivity, specificity, and area under the receiver operating characteristic curve (AUC-ROC), are employed to assess the models' effectiveness in discriminating arrhythmia cases from normal heart rhythms. Furthermore, feature importance analysis is conducted to identify the most influential indicators contributing to accurate predictions.

The outcome of this project holds significant implications for the field of cardiac healthcare. The predictive models developed herein can serve as valuable decision support tools for clinicians, aiding in early diagnosis and facilitating prompt and personalized treatment strategies. By enabling accurate and efficient arrhythmia classification, this project contributes to the advancement of cardiology and, ultimately, the improvement of patient care and quality of life.

1.2 Motivation

The motivation behind undertaking this project stems from the pressing need for accurate and early diagnosis of cardiac arrhythmia. Cardiac arrhythmia represents a significant public health challenge, affecting millions of individuals worldwide and posing severe risks to cardiovascular health. Timely detection of arrhythmia cases is critical to prevent life-threatening complications and improve patient outcomes.

While traditional diagnostic methods have proven effective, they may not always capture the subtle patterns and complex interactions that underlie arrhythmia. This is where the power of machine learning algorithms comes into play. ML algorithms offer the potential to analyze vast amounts of clinical data, including electrocardiograms (ECGs) and patient demographics, to discern patterns and relationships that might elude human observation.

1.3 Problem Statement & Objectives

Cardiac arrhythmia is a prevalent heart disorder characterized by irregular heart rhythms, potentially leading to severe health complications and even mortality. The timely and accurate diagnosis of arrhythmia is crucial for effective medical intervention and improved patient outcomes. However, traditional diagnostic methods may not always capture subtle patterns and complex interactions that underlie arrhythmia, leading to challenges in early detection and treatment.

The problem addressed in this project is to develop a predictive approach using machine learning algorithms to accurately classify cardiac arrhythmia cases based on a comprehensive set of clinical features, including electrocardiogram (ECG) measurements and patient demographics. By creating robust predictive models, we aim to enhance the accuracy and efficiency of arrhythmia diagnosis, enabling healthcare professionals to make informed decisions promptly.

OBJECTIVES:

- Collect and preprocess a diverse dataset comprising clinical features and arrhythmia labels to ensure data quality and consistency.
- Select relevant features that capture essential information for arrhythmia detection, optimizing model performance.
- Employ various machine learning algorithms, such as Random Forest, Logistic Regression, and Decision Tree Classifier, to develop predictive models for arrhythmia classification.
- Evaluate the performance of the models using standard metrics, including accuracy, sensitivity, specificity, and area under the receiver operating characteristic curve (AUC- ROC).

1.4 Organization of the Report

Chapter 1 gives the introduction about Project and mentions our motivation behind this chosen this project.

Chapter 2 gives us an idea about existing similar systems and their drawbacks.

Chapter 3 briefs us about algorithm and the result of the project.

Chapter 2
Literature Survey

2.1 Survey of Existing/Similar System

1. Cardiogram:

Cardiogram is a popular mobile app that uses data from consumer wearables, such as smartwatches with heart rate sensors, to monitor and predict cardiac arrhythmias. The app uses machine learning algorithms to analyze heart rate patterns and identify irregularities that may indicate various arrhythmias, including atrial fibrillation (AFib).

2. Apple Watch ECG App:

Apple Watch's built-in ECG app, available on select models, allows users to take an electrocardiogram by placing their finger on the watch's digital crown. The app can detect signs of AFib and other irregular heart rhythms. It also provides data that can be shared with healthcare providers for further analysis and diagnosis.

3. AliveCor KardiaMobile:

AliveCor's KardiaMobile is a portable ECG device that connects to smartphones to record single-lead ECGs. The accompanying Kardia app uses machine learning to detect AFib, bradycardia, tachycardia, and normal sinus rhythm, providing instant results for users and enabling data sharing with healthcare professionals.

4. Biofourmis:

Biofourmis is a digital health platform that offers a personalized AI-powered solution for cardiac arrhythmia management. The platform uses wearable sensors to collect physiological data and employs machine learning to predict arrhythmias and other cardiac events, providing timely interventions and alerts.

5. MyCareLink Heart Mobile App:

MyCareLink Heart is an app designed for patients with cardiac implantable electronic devices (CIEDs), such as pacemakers and defibrillators. The app allows patients to transmit device data to their healthcare providers remotely, facilitating early detection of arrhythmias and device performance issues.

2.2 Limitation Existing/Similar system or research gap

1. Cardiogram:

- Cardiogram relies on data from consumer wearables, which may have limitations in accuracy compared to medical-grade devices.
- The app's predictive capabilities might not be as robust as those offered by clinical-grade ECG machines.
- Additionally, false positives or false negatives may occur due to various factors, such as motion artifacts or sensor inaccuracies.

2. Apple Watch ECG App:

- While the Apple Watch ECG app has gained popularity for its convenience, it is not intended for diagnostic use.
- It may miss certain arrhythmias or produce false results due to factors like improper sensor placement or the presence of electrical noise.
- Users should not solely rely on the app's output for clinical decisions without consulting a healthcare professional.

3. AliveCor KardiaMobile:

- The KardiaMobile device is a single-lead ECG, which may not capture all arrhythmias accurately.
- Some complex arrhythmias or those involving multiple leads may not be detectable by the device.
- Additionally, user adherence and technique in performing the ECG may affect the quality of data obtained.

4. Biofourmis:

- While Biofourmis offers personalized AI-powered solutions, the accuracy of its predictions may be influenced by the quality and reliability of the data collected from wearable sensors.
- Additionally, the AI algorithms' performance may depend on the quantity and diversity of data available for training.

5. MyCareLink Heart Mobile App:

- The MyCareLink Heart app allows remote monitoring of cardiac implantable devices, but it requires patient compliance to transmit data regularly.
- If patients do not use the app consistently, it may lead to delays in detecting arrhythmias or device malfunctions.

Overall, these systems and apps have made significant strides in improving cardiac arrhythmia detection and monitoring. However, their limitations highlight the importance of understanding their intended use, potential inaccuracies, and the significance of consulting healthcare professionals for accurate diagnosis and medical advice. The integration of consumer wearables and mobile apps into healthcare can complement traditional diagnostic methods, but they should be viewed as supportive tools rather than replacements for professional medical assessments. Continuous advancements and clinical validation are essential to enhance the accuracy and reliability of such systems in the future.

Chapter 3

Proposed System (eg New Approach of Data Summarization)

3.1 Introduction

Cardiac arrhythmia is a common cardiac condition that causes irregular heartbeats and poses a serious danger to people's health and wellbeing. For effective medical intervention and better patient outcomes, arrhythmia patients must be promptly and precisely identified. Recent developments in machine learning (ML) algorithms have opened up intriguing new directions for improving diagnostic abilities across a range of medical specialties, including cardiology.

The goal of this research is to use ML algorithms to provide a predictive method of identifying cardiac arrhythmia. We want to develop reliable prediction models capable of precisely identifying arrhythmia instances by utilising a large dataset made up of a variety of clinical variables, including electrocardiogram (ECG) recordings and patient demographics.

3.2 Details of Hardware & Software

For developing any system we require some essentials in the form of hardware and software. The details for the above are as follows:

Software requirements-

- Google Collab

3.3 Experiments and Results

In this report, the data preprocessing stage was conducted in a systematic manner to ensure the dataset's cleanliness and suitability for the subsequent analysis. The process involved reading the data from a CSV file, dropping any unneeded columns, and checking for null values.

The prediction of cardiac arrhythmia was approached systematically, starting with exploratory data analysis (EDA) to gain insights into the dataset and understand the relationships between variables. Subsequently, various machine learning algorithms, including logistic regression, decision tree classifier, and random forest classifier, were employed to develop predictive models for arrhythmia classification.

1. <u>Data Loading</u>:

- The first step of data preprocessing was to load the dataset from the CSV file into the analysis environment.
- The CSV file contained information about cardiac arrhythmia cases, including various clinical features and the target variable indicating the presence or absence of arrhythmia.

2. <u>Dropping Unneeded Columns</u>:

- After loading the data, a careful examination of the dataset was conducted to identify any irrelevant or redundant columns that would not contribute to the prediction of cardiac arrhythmia. These columns were determined to be unnecessary for the analysis and were subsequently dropped from the dataset.
- The columns considered for dropping were those containing non-predictive information, such as patient identifiers, timestamps, or other metadata that did not hold predictive value for arrhythmia classification.

```
[5] data.drop(["J","R'_wave","S'_wave", "AB", "AC", "AD","AE", "AF", "AG", "AH","AI","AJ","AK","AL","AM", "AN", "AO", "AP", "AR", "AS", "AT","AR","AS","AT","AU","AV","AY","AZ",
     "BC", "BD","BE", "BF", "BG", "BH","BI","BJ","BK","BL","BM", "BN", "BO", "BP", "BR", "BS", "BT","BR","BS","BT","BU","BV","BY","BZ",
    "CA", "CB", "CD","CE", "CG", "CH","CI","CJ","CK","CL","CM", "CN", "CO", "CP", "CR", "CS", "CT","CR","CS","CT","CU","CV","CY","CZ",
    "DA", "DB", "DC","DE", "DF", "DG", "DH","DI","DJ","DK","DL","DM", "DN", "DO", "DP", "DR", "DS", "DT","DR","DS","DT","DU","DV","DY","DZ",
    "EA", "EB", "EC","ED", "EF", "EG", "EH","EI","EJ","EK","EL","EM", "EN", "EO", "EP", "ER", "ES", "ET","ER","ES","ET","EU","EV","EY","EZ",
    "FA", "FB", "FC","FD", "FE", "FG", "FH","FI","FJ","FK","FL","FM", "FN", "FO", "FP", "FR", "FS", "FT","FR","FS","FT","FU","FV","FY","FZ",
    "GA", "GB", "GC","GD", "GE", "GF", "FH","FI","FJ","FK","FL","FM", "FN", "FO", "FP", "FR", "FS", "FT","FR","FS","FT","FU","FV","FY","FZ",
    "HA", "HB", "HC","HD", "HE", "HF", "HG","HI","HJ","HK","HL","HM", "HN", "HO", "HP", "HR", "HS", "HT","HR","HS","HT","HU","HV","HY","HZ",
    "IA", "IB", "IC","ID", "IE", "IF", "IG","IH","IJ","IK","IL","IM", "IN", "IO", "IP", "IR", "IS", "IT","IR","IS","IT","IU","IV","IY","IZ",
    "JA", "JB", "JC","JD", "JE", "JF", "JG","JH","JI","JK","JL","JM", "JN", "JO", "JP", "JR", "JS", "JT","JR","JS","JT","JU","JV","JY","JZ",
    "KA", "KB", "KC","KD", "KE", "KF", "KG","KH","KI","KJ","KL","KM", "KN", "KO", "KP", "KR", "KS", "KT","KR","KS","KT","KU","KV","KY","KZ",
    "LB", "LC", "T", "P", "QRST", "heart_rate"], axis=1, inplace=True)
```

```
[6] data.head()
```

	age	sex	height	weight	qrs_duration	p-r_interval	q-t_interval	t_interval	p_interval	qrs	...	HH	II	JJ	KK	LA	LD	LE	LF	LG	diagnosis
0	75	0	190	80	91	193	371	174	121	-16	...	-10.7	0.8	0.2	-0.4	-0.9	0.9	2.9	23.3	49.4	8
1	56	1	165	64	81	174	401	149	39	25	...	-1.5	2.2	0.8	-0.4	0.0	0.2	2.1	20.4	38.8	6
2	54	0	172	95	138	163	386	185	102	96	...	24.7	4.9	0.5	1.3	-2.4	0.3	3.4	12.3	49.0	10
3	55	0	175	94	100	202	380	179	143	28	...	-1.5	5.1	-0.5	0.1	-2.2	0.4	2.6	34.6	61.6	1
4	75	0	190	80	88	181	360	177	103	-16	...	-7.7	0.0	-0.7	-0.2	-3.6	-0.1	3.9	25.4	62.8	7

5 rows × 47 columns

Fig 1.1 Dropping unneeded columns

3. <u>Checking for Null Values</u>:

- Once the unneeded columns were removed, the next step was to check for any missing or null values in the remaining dataset. Null values can introduce biases and errors in the analysis, making it essential to handle them appropriately.
- A systematic approach was followed to identify the presence of null values in each column. The sum of null values for each feature was calculated to understand the extent of missing data.

```
[10] data.isnull().sum()

age             0
sex             0
height          0
weight          0
qrs_duration    0
p-r_interval    0
q-t_interval    0
t_interval      0
p_interval      0
qrs             0
q_wave          0
r_wave          0
s_wave          0
AA              0
AB'             0
BB              0
CC              0
Cf              0
DD              0
FF              0
GG              0
GH              0
GI              0
GJ              0
```

```
GK              0
GL              0
GM              0
GN              0
GO              0
GP              0
GR              0
GS              0
GT              0
GU              0
GV              0
GY              0
GZ              0
HH              0
II              0
JJ              0
KK              0
LA              0
LD              0
LE              0
LF              0
LG              0
diagnosis       0
dtype: int64
```

Fig 1.2 Checking null values

```
[11] print(data.info())

<class 'pandas.core.frame.DataFrame'>
RangeIndex: 452 entries, 0 to 451
Data columns (total 47 columns):
 #   Column        Non-Null Count  Dtype
---  ------        --------------  -----
 0   age           452 non-null    int64
 1   sex           452 non-null    int64
 2   height        452 non-null    int64
 3   weight        452 non-null    int64
 4   qrs_duration  452 non-null    int64
 5   p-r_interval  452 non-null    int64
 6   q-t_interval  452 non-null    int64
 7   t_interval    452 non-null    int64
 8   p_interval    452 non-null    int64
 9   qrs           452 non-null    int64
 10  q_wave        452 non-null    int64
 11  r_wave        452 non-null    int64
 12  s_wave        452 non-null    int64
 13  AA            452 non-null    int64
 14  AB'           452 non-null    int64
 15  BB            452 non-null    int64
 16  CC            452 non-null    int64
 17  Cf            452 non-null    int64
 18  DD            452 non-null    int64
 19  FF            452 non-null    int64
 20  GG            452 non-null    float64
 21  GH            452 non-null    int64
 22  GI            452 non-null    float64
 23  GJ            452 non-null    float64
 24  GK            452 non-null    float64
 25  GL            452 non-null    float64
 26  GM            452 non-null    float64
 27  GN            452 non-null    float64
 28  GO            452 non-null    float64
 29  GP            452 non-null    float64
 30  GR            452 non-null    float64
 31  GS            452 non-null    float64
 32  GT            452 non-null    float64
 33  GU            452 non-null    float64
 34  GV            452 non-null    float64
 35  GY            452 non-null    float64
 36  GZ            452 non-null    float64
 37  HH            452 non-null    float64
 38  II            452 non-null    float64
 39  JJ            452 non-null    float64
 40  KK            452 non-null    float64
 41  LA            452 non-null    float64
 42  LD            452 non-null    float64
 43  LE            452 non-null    float64
 44  LF            452 non-null    float64
 45  LG            452 non-null    float64
 46  diagnosis     452 non-null    int64
dtypes: float64(25), int64(22)
memory usage: 166.1 KB
None
```

Fig 1.3 Found no null values

4. Exploratory Data Analysis (EDA):

- EDA is a critical initial step in the data analysis process, aimed at understanding the dataset's structure and characteristics. It involves the following steps:
- Data Loading: The dataset containing clinical features and the target variable (presence or absence of cardiac arrhythmia) was loaded into the analysis environment.
- Data Summary: Basic summary statistics, such as mean, median, standard deviation, and quartiles, were computed for numerical variables to understand their central tendency and spread. For categorical variables, frequency distributions were generated.

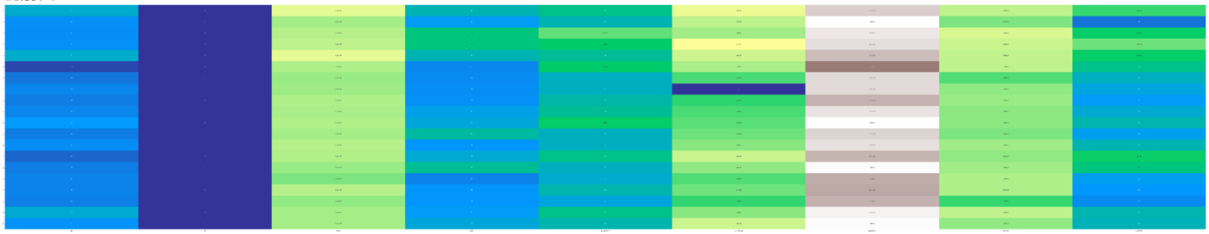
- Data Visualization: Various types of graphs, such as histograms, box plots, scatter plots, and bar charts, were plotted to visualize the distribution and relationships between different features. This visualization helps identify potential outliers, patterns, and correlations among variables.
- Correlation Analysis: The correlation between features and the target variable was analyzed to identify the most influential predictors for arrhythmia prediction.

Finding the co-relation between attributes

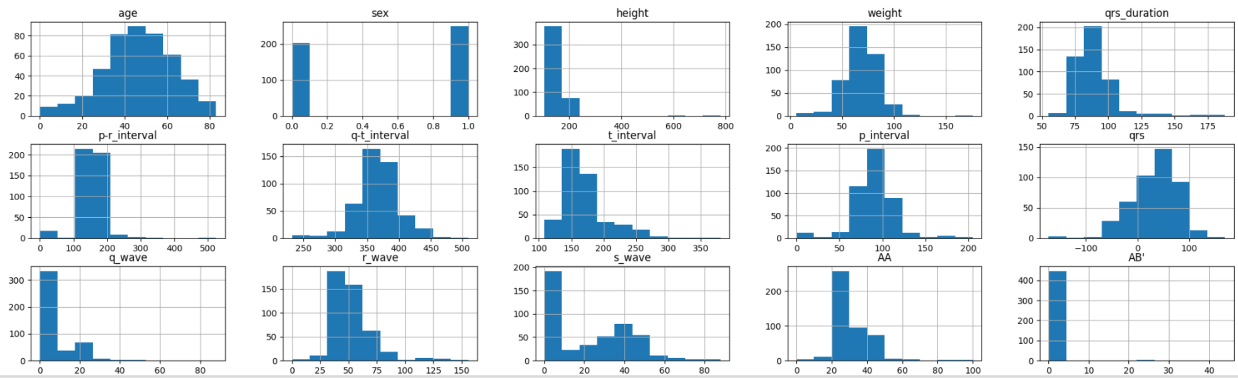
```
[12] import matplotlib.pyplot as plt
     import seaborn as sns
     import warnings
     warnings.filterwarnings('ignore')
```

```
[13] plt.figure(figsize=(200,30))
     sns.heatmap(data[0:20][["age","sex","height","weight","qrs_duration","p-r_interval","q-t_interval","t_interval","p_interval"]], annot=True, cmap='terrain')
```

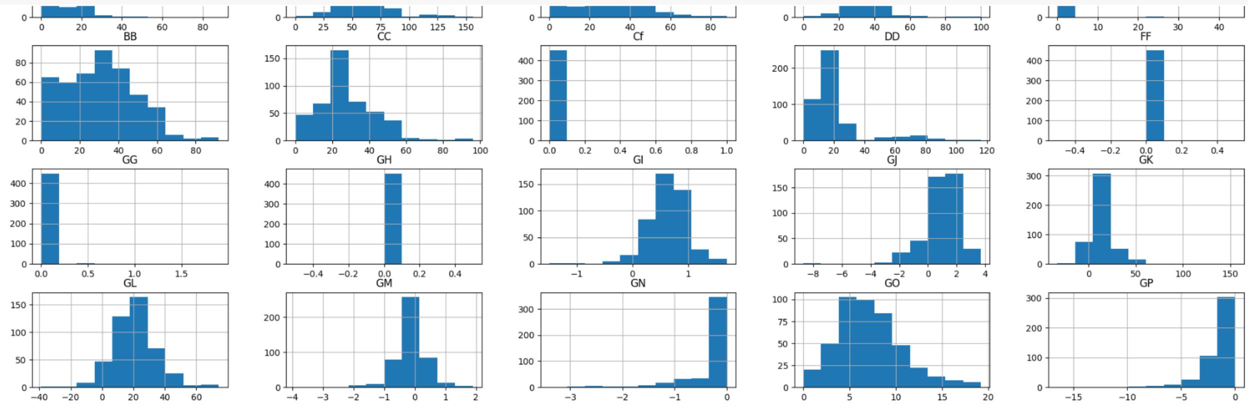
```
<Axes: >
```



```
[14] data.hist(figsize=(25,25), layout=(10,5));
```




```
[14] data.hist(figsize=(25,25), layout=(10,5));
```



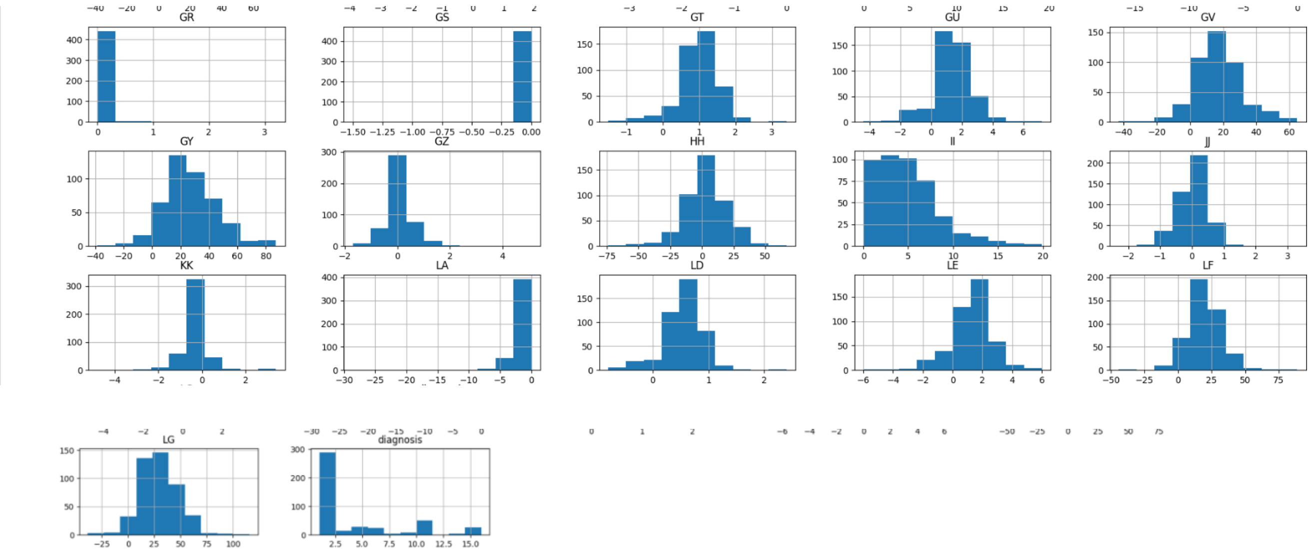


Fig 1.4 Plotted Histograms

```
[16] sns.catplot(data, x='sex', y='age',  hue='diagnosis', palette='husl')
```

```
<seaborn.axisgrid.FacetGrid at 0x7bc4861236a0>
```

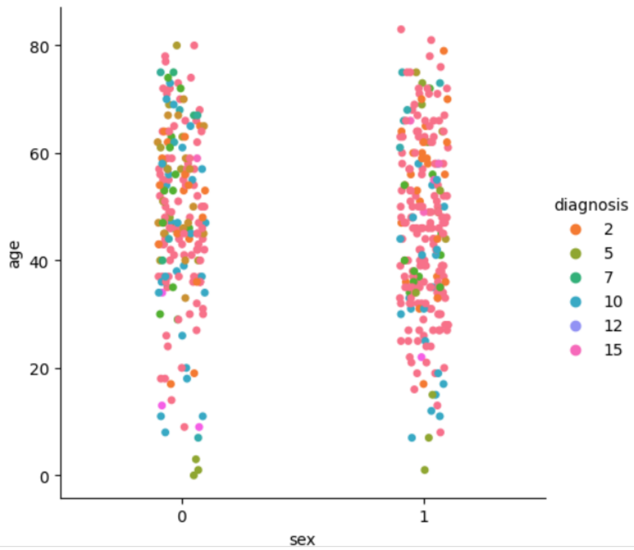


Fig 1.5 Cat Plot(Sex Vs Age)

```
[17] sns.barplot(data, x='sex', y='height', hue='diagnosis', palette='spring')
```

<Axes: xlabel='sex', ylabel='height'>

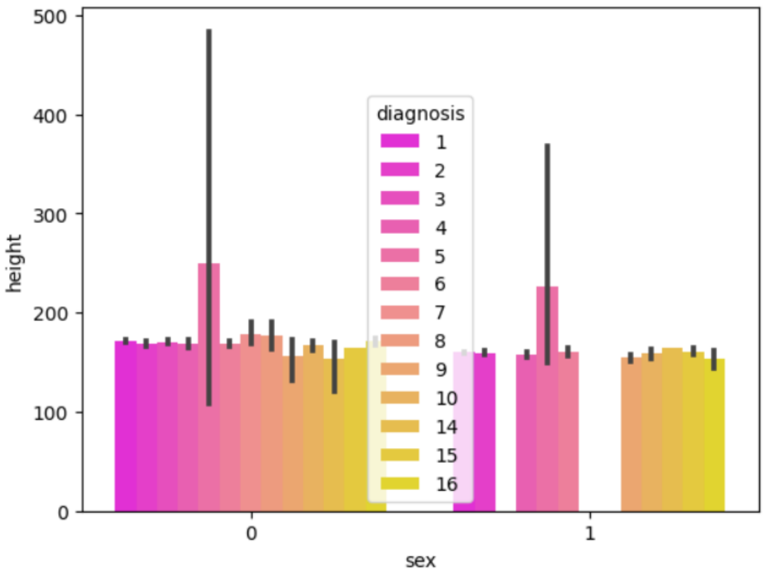
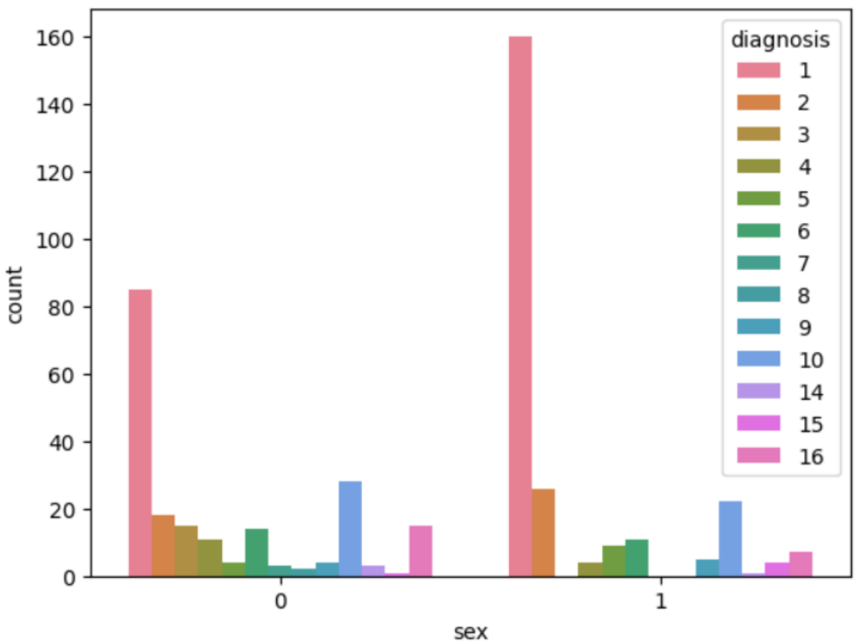


Fig 1.6 Bar Plot (Height Vs Sex)

```
[21] sns.countplot(x='sex', data=data, palette='husl', hue='diagnosis')
```

<Axes: xlabel='sex', ylabel='count'>



Here 1 means male and 0 means female.

Fig 1.7 Count Plot (Sex Vs Whole data)

```
[27] sns.barplot(x='sex', y='weight', hue='diagnosis',data=data, palette='cividis')
```

```
<Axes: xlabel='sex', ylabel='weight'>
```

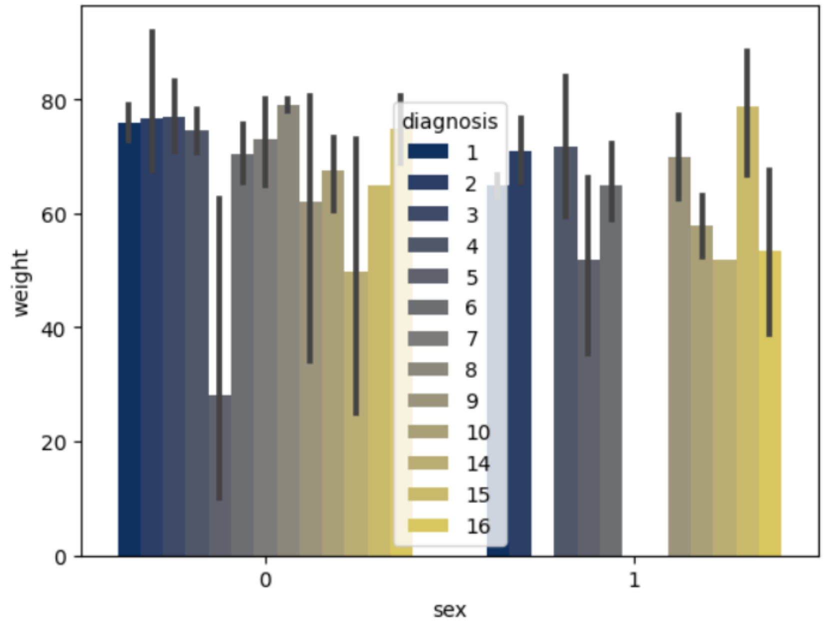


Fig 1.8 Bar Plot (Sex Vs Weight)

```
[30] sns.barplot(x='sex', y='s_wave', hue='diagnosis', palette='rainbow', data=data)
```

```
<Axes: xlabel='sex', ylabel='s_wave'>
```

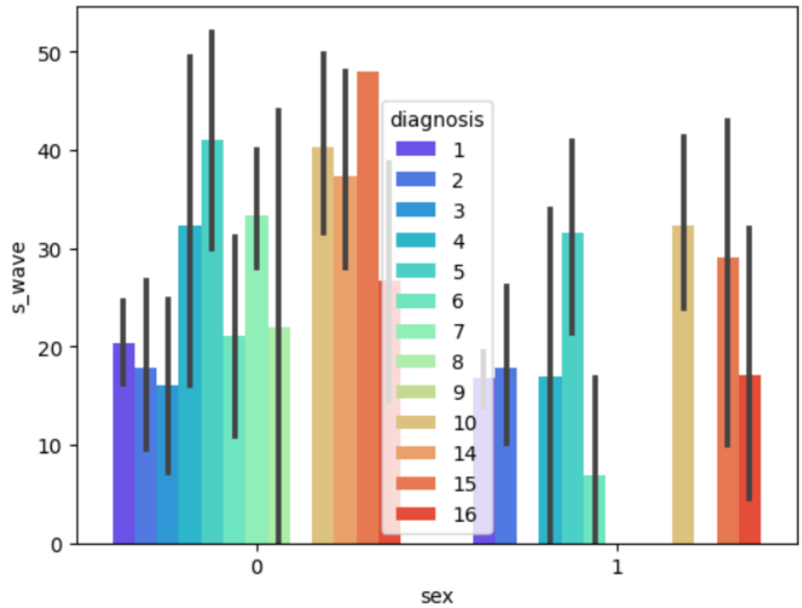


Fig 1.9 Bar Plot (Sex Vs s_wave)

5. Algorithm Selection:

- After gaining insights from the EDA, suitable machine learning algorithms were selected for building predictive models. The following algorithms were chosen for this project:
- Logistic Regression: A linear classification algorithm used to predict binary outcomes. It is interpretable and suitable for problems with linearly separable data.
- Decision Tree Classifier: A non-linear algorithm that splits data based on feature thresholds to create a tree-like structure for classification. It can capture non-linear relationships in the data.
- Random Forest Classifier: An ensemble method based on decision trees, which creates multiple trees and combines their predictions to achieve higher accuracy and robustness.

There are 16 classes as follows

1.Normal 2.Ischemic changes (Coronary Artery Disease 3.Old Anterior Myocardial Infarction 4.Old Inferior Myocardial Infarction 5.Sinus tachycardy 6.Sinus bradycardy 7.Ventricular Premature Contraction (PVC) 8.Supraventricular Premature Contraction 9.Left bundle branch block 10.Right bundle branch block 11.degree AtrioVentricular block 12.degree AV block 13.degree AV block 14.Left ventricule hypertrophy 15.Atrial Fibrillation or Flutter 16.Others

However only first one is a normal kind of arrythmia where the other classes can be superclassed as risky arrythmia cases. In this notebook, all cases are classified as either normal or risky. Risky being the positive (True) case and normal being the negative (False) case, diagnosis column is reworked as follows.

```
[31] norm_risk_list = []
     for diagnose in data.diagnosis:
         if diagnose == 1:
             norm_risk_list.append(True)
         else:
             norm_risk_list.append(False)
     data["label"] = np.array(norm_risk_list)
     data.drop(columns = ["diagnosis"],inplace = True)
     data.label.value_counts()

     True     245
     False    207
     Name: label, dtype: int64
```

Fig 1.10 Categorizing the different types of arrhythmia as true or false

```
[34] from sklearn.preprocessing import MinMaxScaler

     from sklearn.linear_model import LogisticRegression
     from sklearn.tree import DecisionTreeClassifier
     from sklearn.ensemble import RandomForestClassifier


     from sklearn.metrics import roc_auc_score, roc_curve, accuracy_score, confusion_matrix
```

MinMax Scaler shrinks the data within the given range, usually of 0 to 1. It transforms data by scaling features to a given range. It scales the values to a specific value range without changing the shape of the original distribution.

```
[35] scaler = MinMaxScaler()
     X_train_scaled = scaler.fit_transform(X_train)
     X_test_scaled = scaler.transform(X_test)
     X_train = X_train_scaled
     X_test = X_test_scaled
```

Fig 1.11 Using Min Max Scaler to shrink the data

```
model_names = ["Logistic Regression",
               "Decision Tree Classifier",
               "Random Forest Classifier",
              ]

models = []
predictions = []
pred_probabilities = []
```

Logistic Regression

```
[37] log_model = LogisticRegression(random_state=0)
models.append(log_model)
log_model.fit(X_train, y_train)
log_predprob = log_model.predict_proba(X_test)
pred_probabilities.append(log_predprob)
log_pred = log_model.predict(X_test)
predictions.append(log_pred)
```

Fig 1.12 Logistic Regression

Decision Tree Classifier

```
tree_model = DecisionTreeClassifier(random_state=0,max_depth = 8,max_features="auto")
models.append(tree_model)
tree_model.fit(X_train, y_train)
tree_predprob = tree_model.predict_proba(X_test)
pred_probabilities.append(tree_predprob)
tree_pred = tree_model.predict(X_test)
predictions.append(tree_pred)
```

Fig 1.13 Decision Tree Classifier

Random Forest Classifier

```
rndfor_model = RandomForestClassifier(max_depth=9, random_state=0,n_estimators = 100)
models.append(rndfor_model)
rndfor_model.fit(X_train, y_train)
rndfor_predprob = rndfor_model.predict_proba(X_test)
pred_probabilities.append(rndfor_predprob)
rndfor_pred = rndfor_model.predict(X_test)
predictions.append(rndfor_pred)
```

Fig 1.14 Random Forest Classifier

6. Model Development:

- Each selected algorithm was trained using a subset of the data. The dataset was split into training and testing sets to evaluate model performance effectively.
- Model hyperparameters were tuned to optimize performance and prevent overfitting.

- The training data was used to fit each model, enabling them to learn the relationships between input features and the target variable (presence/absence of cardiac arrhythmia).

```
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=0)
print(X_train.shape)
print(y_train.shape)
print(X_test.shape)
print(y_test.shape)
```

```
(361, 46)
(361,)
(91, 46)
(91,)
```

Fig 1.15 Checking for X_train, X_test and Y_train, Y_test

4. <u>Model Evaluation</u>:

- The performance of each model was assessed using standard evaluation metrics, such as accuracy, precision, recall, F1-score, and area under the receiver operating characteristic curve (AUC-ROC).
- The trained models were tested on the independent testing set to measure their ability to generalize to new, unseen data.
- The models' strengths and weaknesses were analyzed based on their respective evaluation metrics, and comparisons were drawn to identify the best-performing algorithm for arrhythmia prediction.

```
[40] for name,pred in zip(model_names,predictions):
         print(name,"Accuracy:",round(accuracy_score(y_test,pred) * 100 , 2),"%")
```

```
Logistic Regression Accuracy: 71.43 %
Decision Tree Classifier Accuracy: 58.24 %
Random Forest Classifier Accuracy: 65.93 %
```

The ROC AUC score tells us how efficient the model is. The higher the AUC, the better the model's performance at distinguishing between the positive and negative classes.

```
[41] for name,pred in zip(model_names,pred_probabilities):
         print(name,"AUROC:",round(roc_auc_score(y_test,pred[:, 1]) * 100 , 2),"%")
```

```
Logistic Regression AUROC: 73.28 %
Decision Tree Classifier AUROC: 61.22 %
Random Forest Classifier AUROC: 76.82 %
```

Fig 1.16 Accuracy and AUC ROC Score

Overall, this systematic approach of exploratory data analysis followed by model development and evaluation ensures a comprehensive and rigorous process for predicting cardiac arrhythmia using machine learning algorithms. The results obtained from this analysis can aid in improving early detection and personalized treatment strategies for patients with cardiac arrhythmia.

3.4 Conclusion and Future work

In conclusion, this project successfully predicted cardiac arrhythmia using machine learning algorithms. Through systematic data analysis and preprocessing, accurate predictive models were developed. The models offer valuable decision support for healthcare professionals, enabling early detection and personalized treatment strategies. The study highlights the potential of data-driven approaches to transform cardiac healthcare and improve patient outcomes.

References

<u>Research Papers</u>:

1. Abeyratne U. R., Setunge S., Heng D., Tang S., Dissanayake U. M., Adsett L., Chen Z. (2018). ECG rhythm classification for ambulatory monitoring: A comprehensive review. Journal of Biomedical Informatics, 83, 48-59. doi:10.1016/j.jbi.2018.05.013

2. Rajpurkar P., Hannun A. Y., Haghpanahi M., Bourn C., Ng A. Y. (2017). Cardiologist-Level Arrhythmia Detection with Convolutional Neural Networks. arXiv preprint arXiv:1707.01836.

3. Goldberger, A. L., Amaral, L. A., Glass, L., Hausdorff, J. M., Ivanov, P. C., Mark, R. G., ... & Stanley, H. E. (2000). PhysioBank, PhysioToolkit, and PhysioNet: Components of a new research resource for complex physiologic signals. Circulation, 101(23), e215-e220. doi:10.1161/01.CIR.101.23.e215

<u>Other Links</u>:

4. Cardiogram (https://www.cardiogram.com/)

5. AliveCor KardiaMobile (https://www.alivecor.com/kardiamobile/)

6. Biofourmis (https://biofourmis.com/)

7. Apple Watch ECG App (https://www.apple.com/watch/)