iJRASET
International Journal For Research in
Applied Science and Engineering Technology

INTERNATIONAL JOURNAL FOR RESEARCH
IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Volume: 13 Issue: IX Month of publication: September 2025

DOI: https://doi.org/10.22214/ijraset.2025.73976

www.ijraset.com

Call: ✆08813907089 | E-mail ID: ijraset@gmail.com

International Journal for Research in Applied Science & Engineering Technology (IJRASET)
ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.538
Volume 13 Issue IX Sep 2025- Available at www.ijraset.com

Prediction Heart Disease Using Machine Learning Techniques

Mahesh V N[1], Basavesha D[2]
[1]*M.Tech Scholar, Department of CSE, Shridevi Institute of Engineering and Technology Tumkur*
[2]*Professor, Department of CSE, Shridevi Institute of Engineering and Technology Tumkur*

Abstract: This study gives efficient approach to classify ECG signals with high accuracy. Each heartbeat is a combination of action impulse waveforms produced by different specialized cardiac heart tissues. Heartbeats classification faces some difficulties because these waveforms differ from person to another, they are described by some features. These features are the inputs of machine learning algorithm.

Multiples classifiers are proposed for ECG classification, these classifiers are used mostly in Big Data and Machine Learning fields by the weighted voting principle. Each classifier influences the final decision according to its performance on the training data.

Parameters of each classifier are adjusted on the basis of an individual classifier's performance on the training data by applying the pseudoinverse technique.

Keywords: pseudoinverse, stenosis , nontraumatic , Spark–Scala , artefact removal.

I. INTRODUCTION

Millions of people suffer from irregular heartbeats which can be lethal in some cases. Therefore, accurate and low-cost diagnosis of arrhythmic heartbeats is highly desirable. Many studies have developed arrhythmia classification approaches that use automatic analysis and diagnosis systems based on ECG signals. The most important factors for the analysis and diagnosis of cardiac diseases are features extraction and beats classification.

An electrocardiogram (ECG) is a complete representation of the electrical activity of the heart on the surface of the human body, and it is extensively applied in the clinical diagnosis of heart diseases, it can be reliably used as a measure to monitor the functionality of the cardiovascular system. ECG signals have been widely used for detecting heart diseases due to its simplicity and non-invasive nature. Features of ECG signals can be computed from ECG samples and extracted using software.

A. Problem statement

This study gives efficient approach to classify ECG signals with high accuracy. Each heartbeat is a combination of action impulse waveforms produced by different specialized cardiac heart tissues. Heartbeats classification faces some difficulties because these waveforms differ from person to another, they are described by some features. These features are the inputs of machine learning algorithm.

B. Objectives

Multiples classifiers are proposed for ECG classification, these classifiers are used mostly in Big Data and Machine Learning fields by the weighted voting principle. Each classifier influences the final decision according to its performance on the training data. Parameters of each classifier are adjusted on the basis of an individual classifier's performance on the training data by applying the pseudoinverse technique. The classification performance in this approach was validated on a set of ECG records with different temporal length. Our work is distinguished by:

1) Number of tested records (205,146 records of 51 patients).
2) Complexity of heartbeat types in training and testing (training records contains Normal and Abnormal beats).
3) Using Machine learning algorithms for classification.
4) Using big data tool (Spark–Scala).
5) Using local host pc (according to the lack of requirements).

C. Proposed system

Classification is a process of categorizing a given set of data into classes, It can be performed on both structured or unstructured data. The process starts with predicting the class of given data points.

The classes are often referred to as target, label or categories. The classification predictive modeling is the task of approximating the mapping function from input variables to discrete output variables. The main goal is to identify which class/category the new data will fall into.

Heart disease detection can be identified as a classification problem, this is a binary classification since there can be only two classes i.e has heart disease or does not have heart disease. The classifier, in this case, needs training data to understand how the given input variables are related to the class. And once the classifier is trained accurately, it can be used to detect whether heart disease is there or not for a particular patient. Since classification is a type of supervised learning, even the targets are also provided with the input data.

II. LITERATURE REVIEW

In-hospital cardiac arrest (IHCA) is a potentially catastrophic adverse outcome and one that can cause substantial stress to medical staff. Anticipating a patient's physiological deterioration prior to cardiac arrest allows at risk patients to be identified. Clinical deterioration of respiratory, cardiac, and/or cerebral function without appropriate response to abnormal physiological variables can lead to cardiac arrest. Early Warning Score (MEWS) has been widely adopted as a useful clinical tool to identify those patients at risk of deterioration who require attention. MEWS is a summed score of routinely recorded physiological data, which includes measurements of systolic blood pressure, heart rate, respiratory rate, body temperature, and level of consciousness (the latter represented as either alert, reacting to voice, reacting to pain, or unresponsive).2 MEWS is a simple, quick bedside tool that can be applied by nursing staff.

There was no significant difference in MEWS at triage between the two outcome groups. Each points increase in periarrest MEWS reduced the chance of survival to discharge; periarrest MEWS was also an independent predictor along with other risk factors, such as sex, ED diagnosis, cardiac arrest rhythm, cardiac arrest cause, and underlying comorbidities. Over the period of 30 months, 234 nontraumatic adult patients suffered from IHCA during an ED stay and received resuscitation, and 99 patients with periarrest MEWS were included in the final analysis. The limitation of our study is that all cardiac arrests occurred in the ED and in just one medical center; whether the result is applicable in the ward or in the intensive care unit is questionable. In our study, not all patients had periarrest vital signs recorded. Only 99 patients were eligible for final analyses, which is due to the limitations of a retrospective study. Selection bias may also exist because only those with the most critical status had complete vital signs checked before cardiac arrest.

III. SYSTEM ANALYSIS

A. Proposed System

A proposed system for early detection of cardiac arrest using a machine learning approach would leverage the latest advancements in neural network architectures, data pre-processing techniques, and real-time monitoring capabilities. The system would integrate multiple modalities of cardiovascular data, including ECG signals, heart rate variability (HRV), blood pressure, and possibly other physiological parameters obtained from wearable sensors or medical devices. This multi-modal approach enables a more comprehensive analysis of cardiac health and improves the accuracy of early detection.The proposed system would feature a sophisticated machine learning model, potentially combining CNNs, RNNs, and attention mechanisms to capture both spatial and temporal patterns in cardiovascular data.

The model would be trained on large-scale, annotated datasets, incorporating diverse patient demographics and pathological conditions to ensure robustness and generalization. Transfer learning techniques could be employed to leverage pre-trained models and adapt them to specific cardiac arrest detection tasks, accelerating model development and deployment to enhance real-time monitoring capabilities, the system would integrate with wearable devices or remote monitoring platforms, allowing continuous data acquisition and analysis. Advanced signal processing algorithms would be implemented for real-time data preprocessing, artifact removal, and feature extraction to improve the quality and reliability of predictions. The system would incorporate intelligent alarm systems to provide timely alerts to healthcare providers or patients in the event of detected abnormalities, facilitating prompt intervention and potentially preventing cardiac events.

IV. SYSTEM DESIGN

A. System Architecture

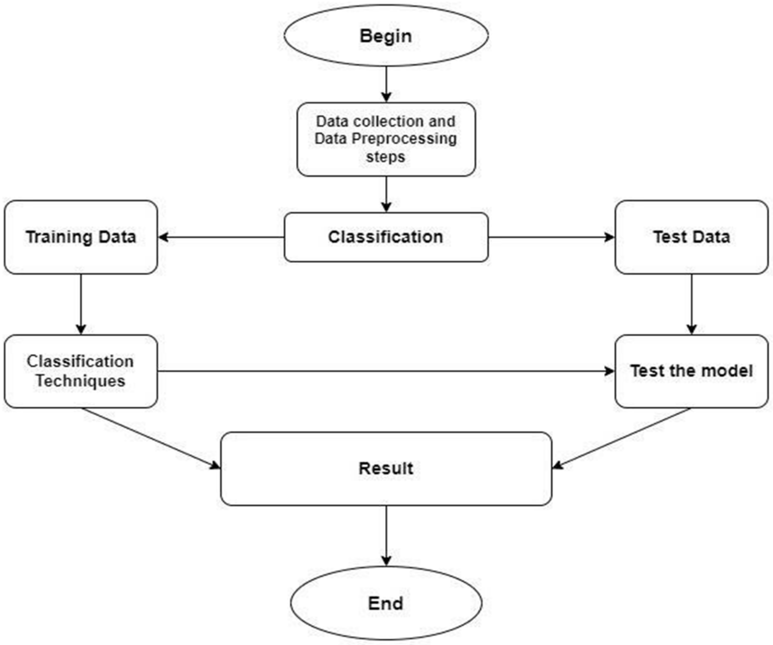

Begin
Data collection and Data Preprocessing steps
Classification
Training Data
Test Data
Classification Techniques
Test the model
Result
End

Fig-1: system architecture of heart disease prediction

In this Fig-1, user use the heartbeat dataset taken as an input and then it is visualized by drawing graphs. The data that is obtained after visualization is ready for preprocessing i.e., acquiring the heartbeat data, importing necessary libraries and so on. Then the process of model building is started by taking the respective data model information our model has been created. Here we are using five important classification techniques to build our model. After all these processes finale output called resultant graph is obtained.

1) Use Case Diagram

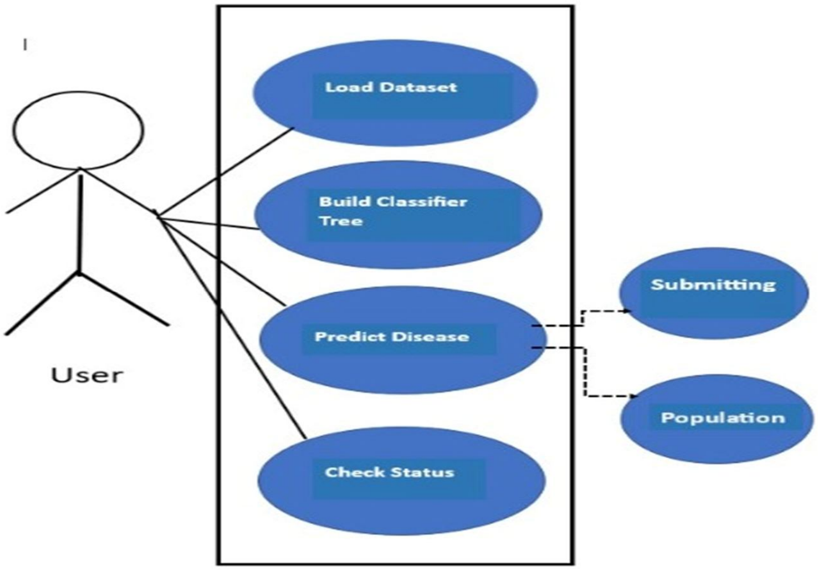


Fig-2: Use case diagram for heart disease prediction

In this Fig-2 user can interact with model and can give input and access output. Input given by user goes to model building phase and then after set of values are trained to get the output. The output can be visualized by user through user interface which has predicted output. Admin has rights to extract information about patient database which consists of patient's health history and admin can also access database.

2) Dataflow Diagram

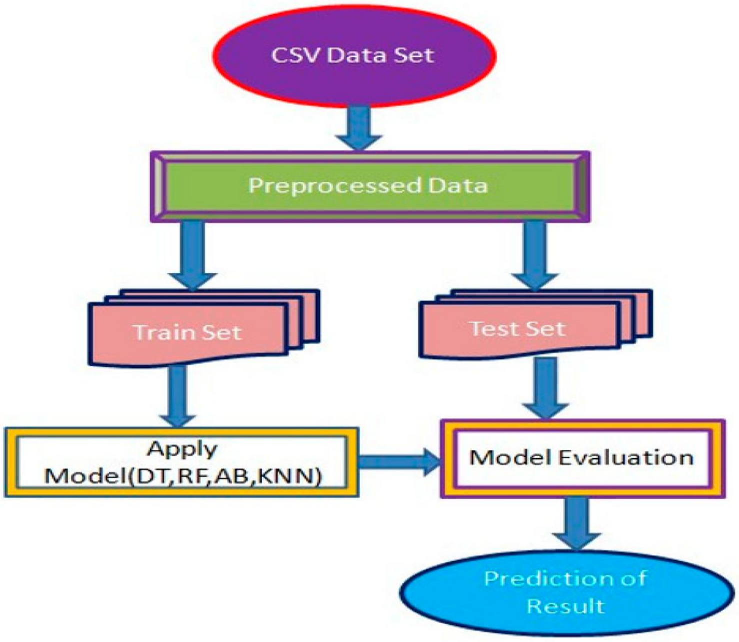


Fig-3: Dataflow diagram for heart disease prediction

In this Fig-3 dataflow diagram the input of heart features are taken as input, this raw data is preprocessed. The preprocessed data is then split into sub datasets one is train data and other is test data. Train data is given to model for training purpose and test data is directly given to classifier for further process.

3) Sequence diagram

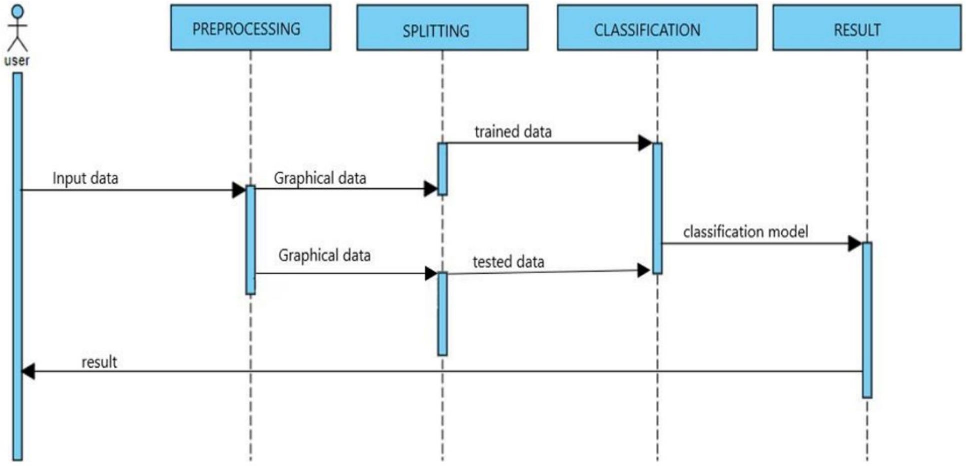
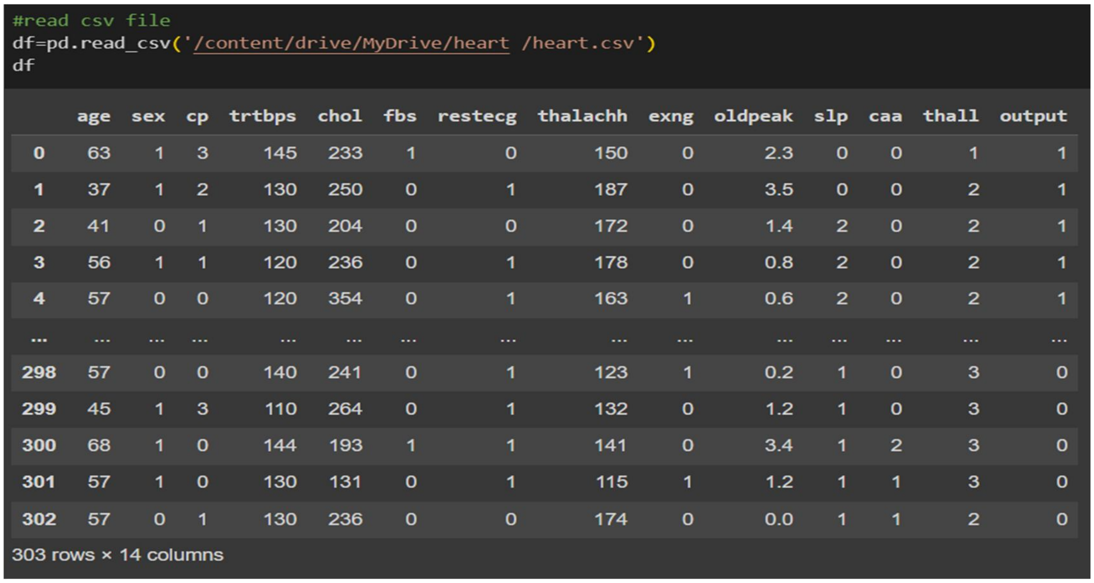


Fig-4: sequence diagram for heart disease prediction

In this Fig-4 shows sequence diagram which explain the prediction process sequentially. User which has input data i.e. heart disease dataset is given to preprocessing phase, the raw data is processed to graphical data and den it is sent to splitting phase. Here data set is split into train and test data. This data is used for model building phase, model is built, and the output is generated this result is used by user for further use.

V. RESULTS



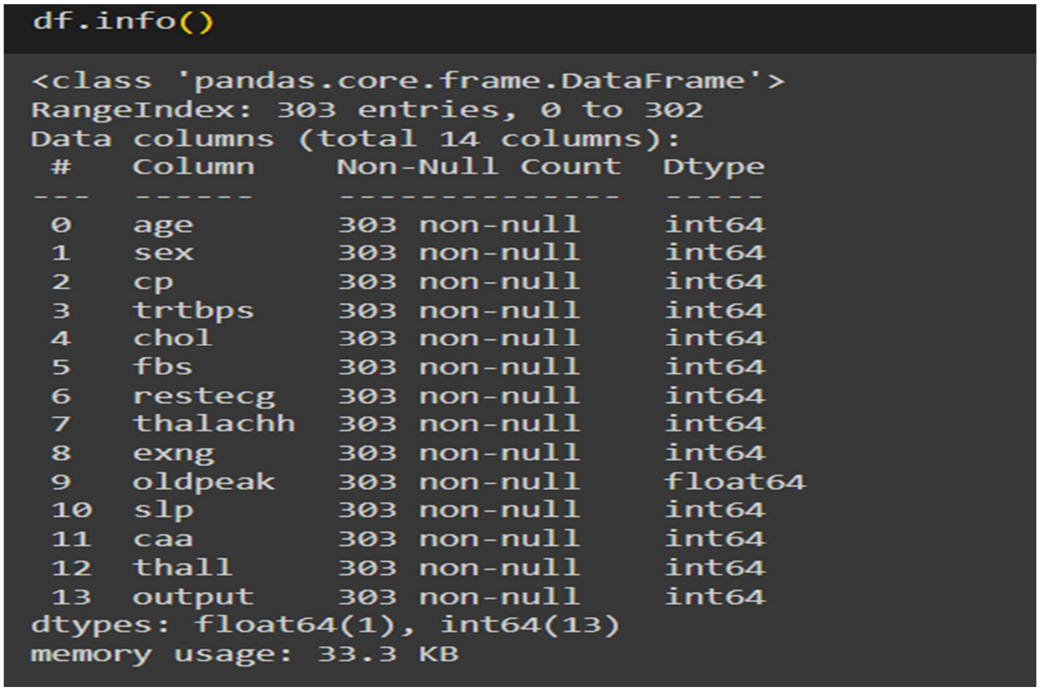
```
#read csv file
df=pd.read_csv('/content/drive/MyDrive/heart /heart.csv')
df
```

	age	sex	cp	trtbps	chol	fbs	restecg	thalachh	exng	oldpeak	slp	caa	thall	output
0	63	1	3	145	233	1	0	150	0	2.3	0	0	1	1
1	37	1	2	130	250	0	1	187	0	3.5	0	0	2	1
2	41	0	1	130	204	0	0	172	0	1.4	2	0	2	1
3	56	1	1	120	236	0	1	178	0	0.8	2	0	2	1
4	57	0	0	120	354	0	1	163	1	0.6	2	0	2	1
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
298	57	0	0	140	241	0	1	123	1	0.2	1	0	3	0
299	45	1	3	110	264	0	1	132	0	1.2	1	0	3	0
300	68	1	0	144	193	1	1	141	0	3.4	1	2	3	0
301	57	1	0	130	131	0	1	115	1	1.2	1	1	3	0
302	57	0	1	130	236	0	0	174	0	0.0	1	1	2	0

303 rows × 14 columns

Fig-5: Datasets for input

Dataset of given CSV file is shown in the fig-5.



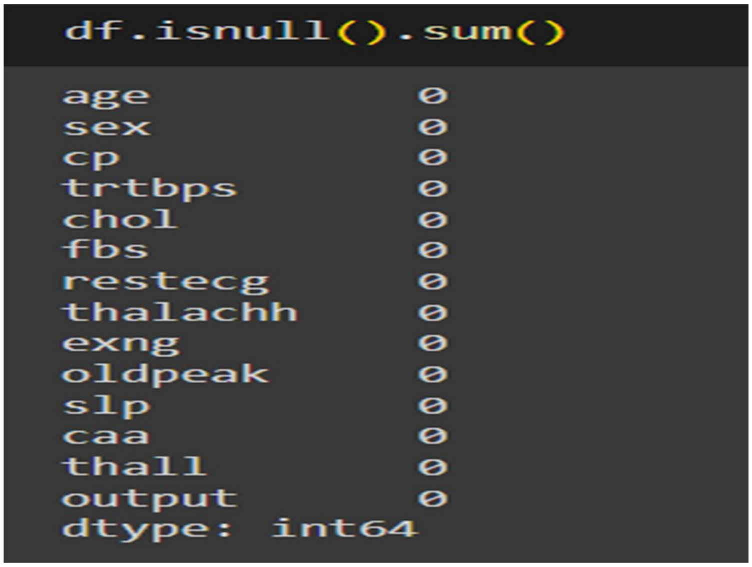
```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 303 entries, 0 to 302
Data columns (total 14 columns):
 #   Column    Non-Null Count  Dtype
---  ------    --------------  -----
 0   age       303 non-null    int64
 1   sex       303 non-null    int64
 2   cp        303 non-null    int64
 3   trtbps    303 non-null    int64
 4   chol      303 non-null    int64
 5   fbs       303 non-null    int64
 6   restecg   303 non-null    int64
 7   thalachh  303 non-null    int64
 8   exng      303 non-null    int64
 9   oldpeak   303 non-null    float64
 10  slp       303 non-null    int64
 11  caa       303 non-null    int64
 12  thall     303 non-null    int64
 13  output    303 non-null    int64
dtypes: float64(1), int64(13)
memory usage: 33.3 KB
```

Fig-6: DataFrame

In this Fig-6 the info() method applied to a pandas DataFrame. It provides a summary of the DataFrame's structure, including the number of non-null values in each column, the data type of each column, and memory usage.

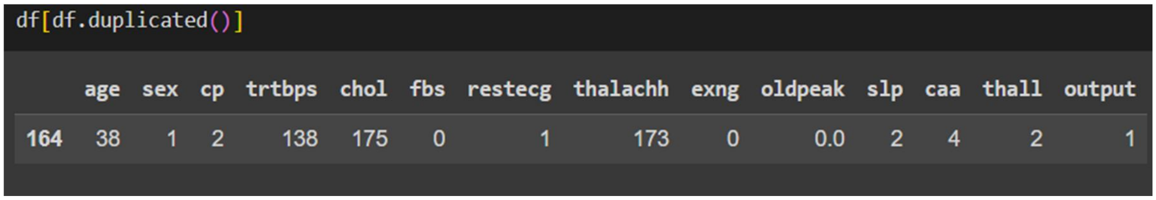
Here's what each section means:



```
df.isnull().sum()

age         0
sex         0
cp          0
trtbps      0
chol        0
fbs         0
restecg     0
thalachh    0
exng        0
oldpeak     0
slp         0
caa         0
thall       0
output      0
dtype:  int64
```

Fig-7: Output of null values

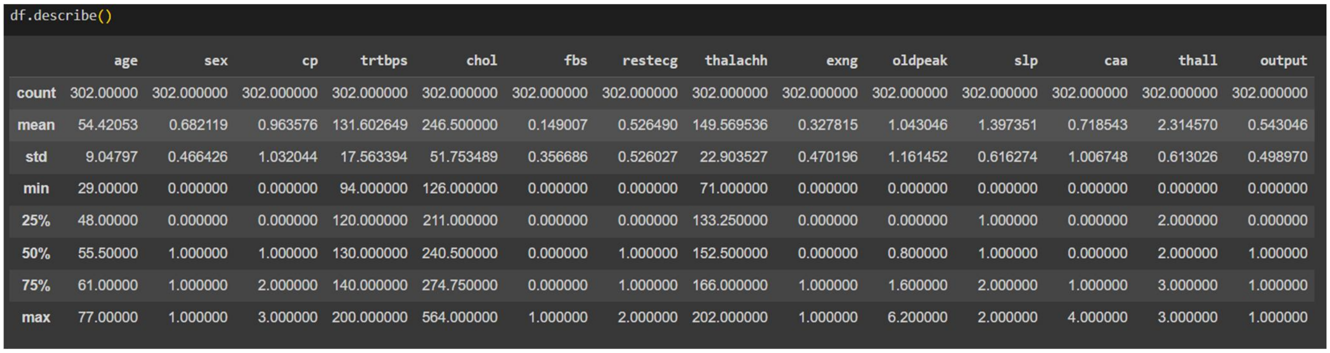
In this Fig-7 output of df.isnull().sum() indicates that there are no missing values (null values) in any of the columns of the DataFrame. Each column has 0 null values, meaning all values are present.



```
df[df.duplicated()]
```

	age	sex	cp	trtbps	chol	fbs	restecg	thalachh	exng	oldpeak	slp	caa	thall	output
164	38	1	2	138	175	0	1	173	0	0.0	2	4	2	1

Fig-8: Output of duplicated value

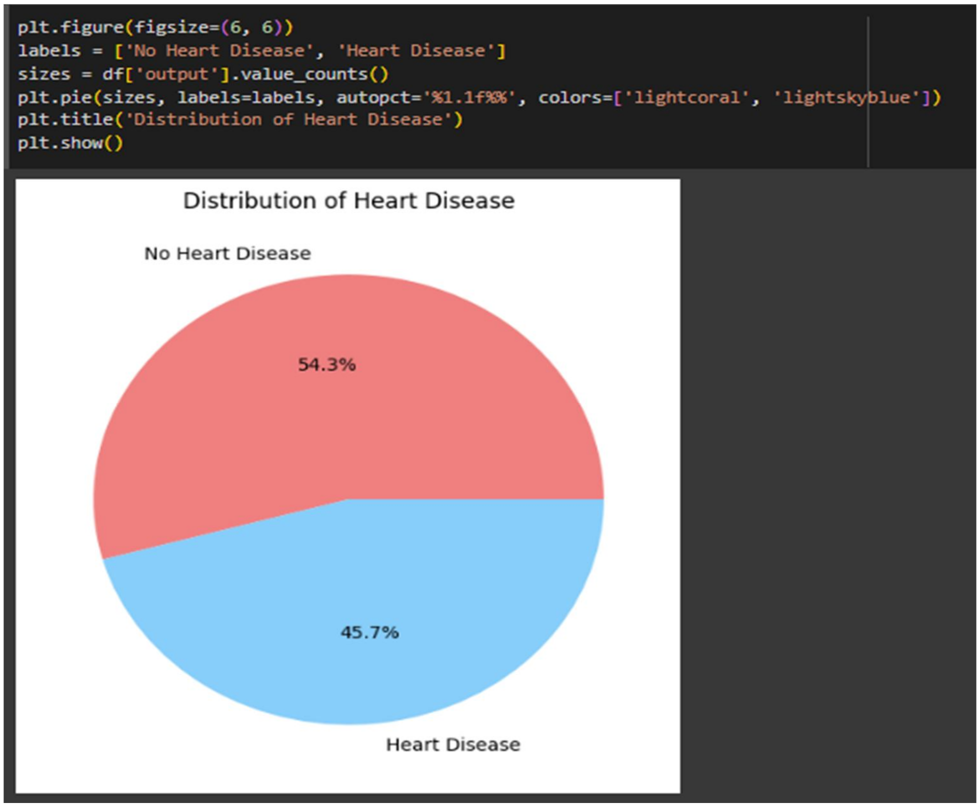


```
df.describe()
```

	age	sex	cp	trtbps	chol	fbs	restecg	thalachh	exng	oldpeak	slp	caa	thall	output
count	302.00000	302.000000	302.000000	302.000000	302.000000	302.000000	302.000000	302.000000	302.000000	302.000000	302.000000	302.000000	302.000000	302.000000
mean	54.42053	0.682119	0.963576	131.602649	246.500000	0.149007	0.526490	149.569536	0.327815	1.043046	1.397351	0.718543	2.314570	0.543046
std	9.04797	0.466426	1.032044	17.563394	51.753489	0.356686	0.526027	22.903527	0.470196	1.161452	0.616274	1.006748	0.613026	0.498970
min	29.00000	0.000000	0.000000	94.000000	126.000000	0.000000	0.000000	71.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000
25%	48.00000	0.000000	0.000000	120.000000	211.000000	0.000000	0.000000	133.250000	0.000000	0.000000	1.000000	0.000000	2.000000	0.000000
50%	55.50000	1.000000	1.000000	130.000000	240.500000	0.000000	1.000000	152.500000	0.000000	0.800000	1.000000	0.000000	2.000000	1.000000
75%	61.00000	1.000000	2.000000	140.000000	274.750000	0.000000	1.000000	166.000000	1.000000	1.600000	2.000000	1.000000	3.000000	1.000000
max	77.00000	1.000000	3.000000	200.000000	564.000000	1.000000	2.000000	202.000000	1.000000	6.200000	2.000000	4.000000	3.000000	1.000000

Fig-9: Output of database

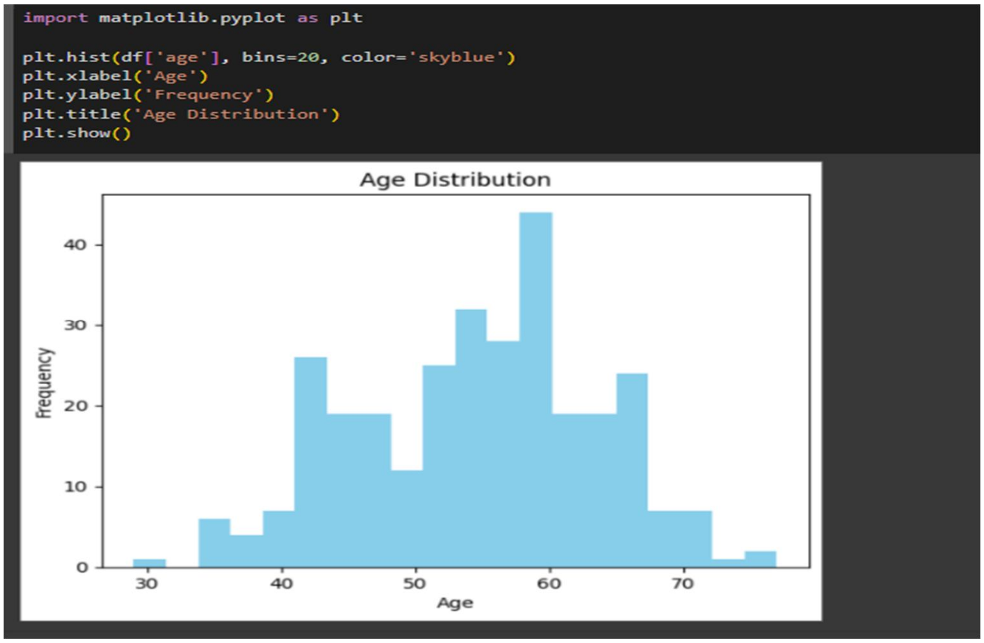
The describe() method applied to the DataFrame. It provides various statistics for each numeric column in the DataFrame.Here's what each part of the output represents:



```
plt.figure(figsize=(6, 6))
labels = ['No Heart Disease', 'Heart Disease']
sizes = df['output'].value_counts()
plt.pie(sizes, labels=labels, autopct='%1.1f%%', colors=['lightcoral', 'lightskyblue'])
plt.title('Distribution of Heart Disease')
plt.show()
```

Fig-10: Distribution of heart disease

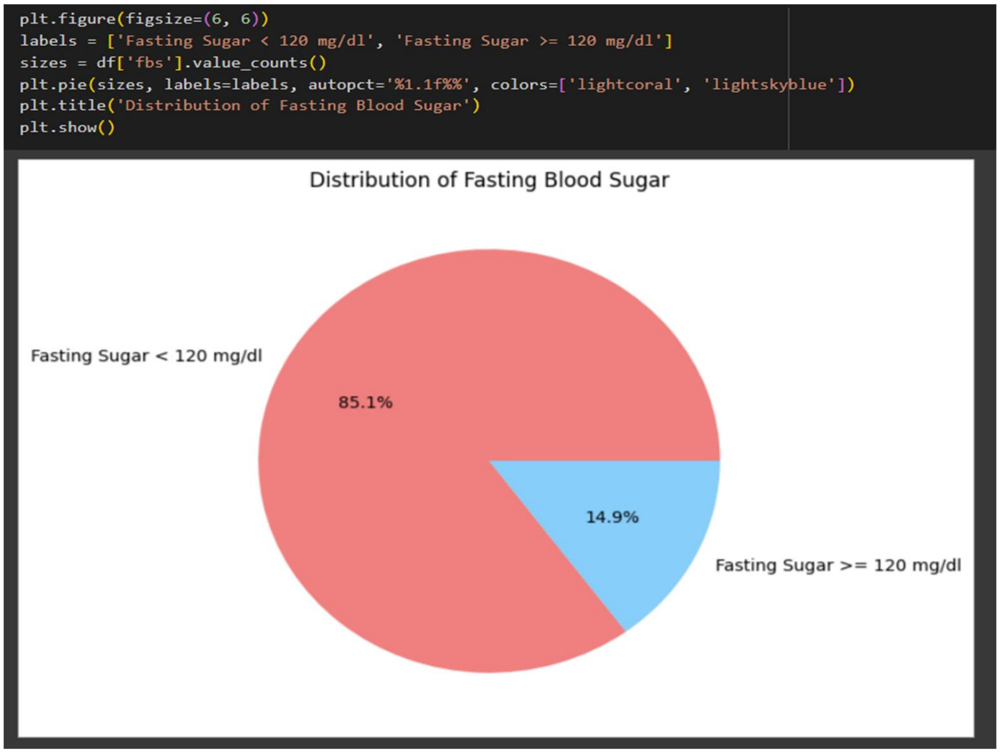
This Fog-10 shows that “Distribution of Heart Disease” with two categories: “No Heart Disease” and “Heart Disease”. The chart shows that 54.3% of the dataset does not have heart disease, while 45.7% does have heart disease.



```
import matplotlib.pyplot as plt

plt.hist(df['age'], bins=20, color='skyblue')
plt.xlabel('Age')
plt.ylabel('Frequency')
plt.title('Age Distribution')
plt.show()
```

Fig-11: Age distribution

This Fig-11 shows that “Age Distribution of Heart Disease”. We see that most people who are suffering are of the age of 58-60.

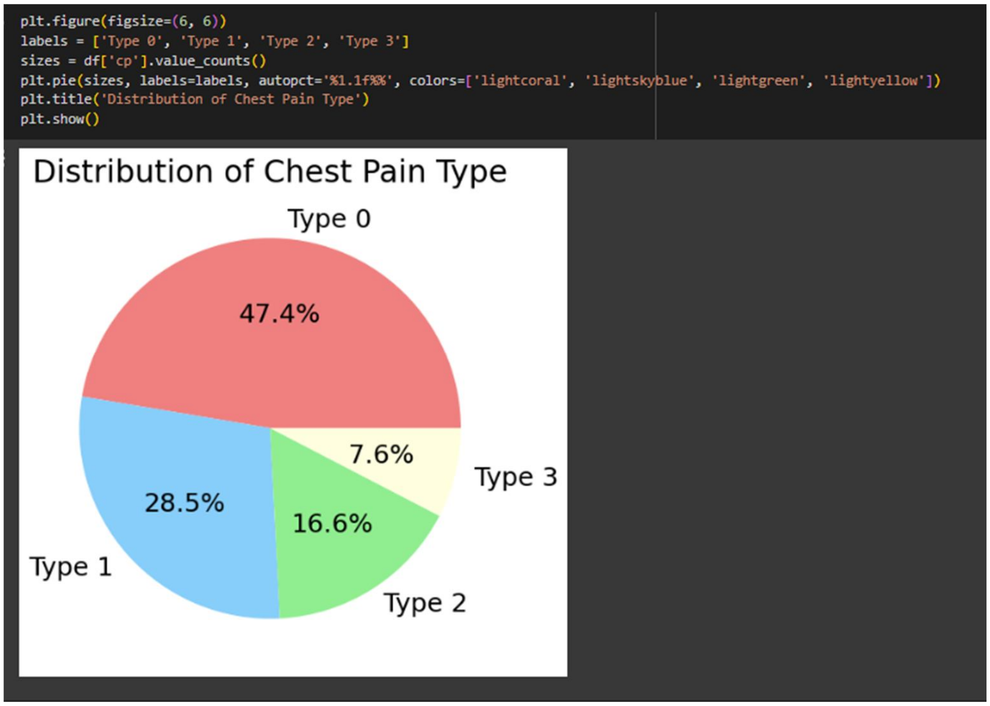


```
plt.figure(figsize=(6, 6))
labels = ['Fasting Sugar < 120 mg/dl', 'Fasting Sugar >= 120 mg/dl']
sizes = df['fbs'].value_counts()
plt.pie(sizes, labels=labels, autopct='%1.1f%%', colors=['lightcoral', 'lightskyblue'])
plt.title('Distribution of Fasting Blood Sugar')
plt.show()
```

Distribution of Fasting Blood Sugar
Fasting Sugar < 120 mg/dl
85.1%
14.9%
Fasting Sugar >= 120 mg/dl

Fig-12: Distribution of fasting blood sugar

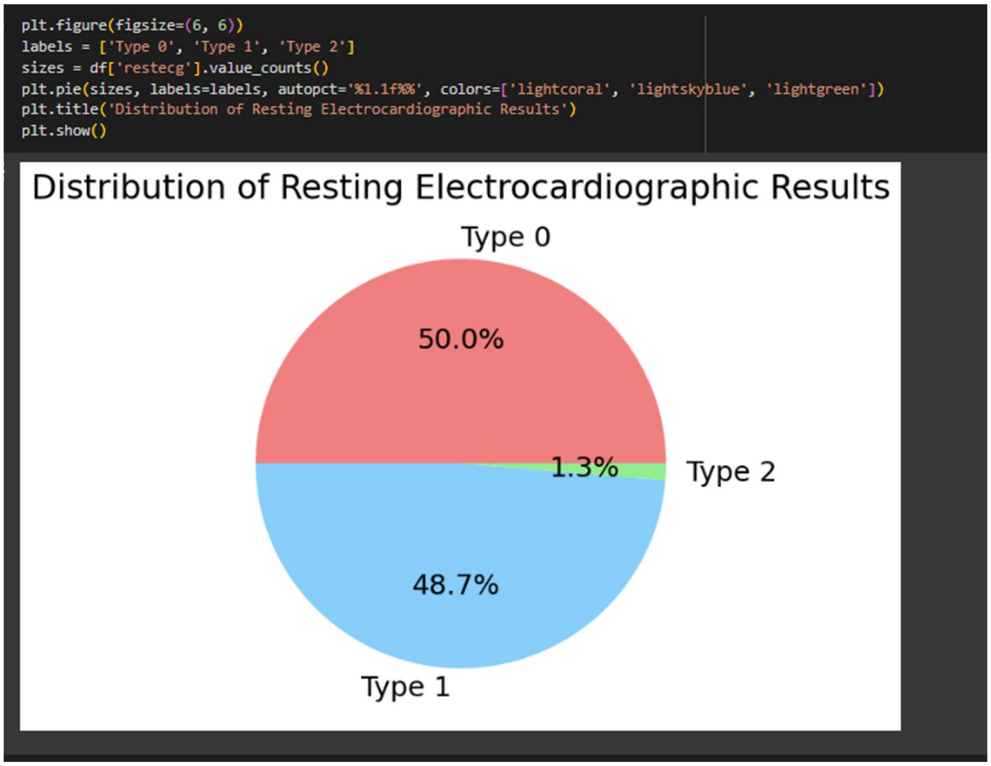
This Fig-12 shows that “Distribution of fasting blood sugar”. If fasting blood sugar > 120mg/dl then : 1 (true) else fasting blood sugar < 120mg/dl : 0 (false).



```
plt.figure(figsize=(6, 6))
labels = ['Type 0', 'Type 1', 'Type 2', 'Type 3']
sizes = df['cp'].value_counts()
plt.pie(sizes, labels=labels, autopct='%1.1f%%', colors=['lightcoral', 'lightskyblue', 'lightgreen', 'lightyellow'])
plt.title('Distribution of Chest Pain Type')
plt.show()
```

Fig-13: Distribution of chest pain type

This Fig-13 shows that “Distribution of chest pain type”. Type 0=Typical Angina has 47.4%, Type 1= Atypical Angina 28.5%, Type 2= Non-Anginal Pain 16.6%, Type 3=Asymptomatic 7.6%.



```
plt.figure(figsize=(6, 6))
labels = ['Type 0', 'Type 1', 'Type 2']
sizes = df['restecg'].value_counts()
plt.pie(sizes, labels=labels, autopct='%1.1f%%', colors=['lightcoral', 'lightskyblue', 'lightgreen'])
plt.title('Distribution of Resting Electrocardiographic Results')
plt.show()
```

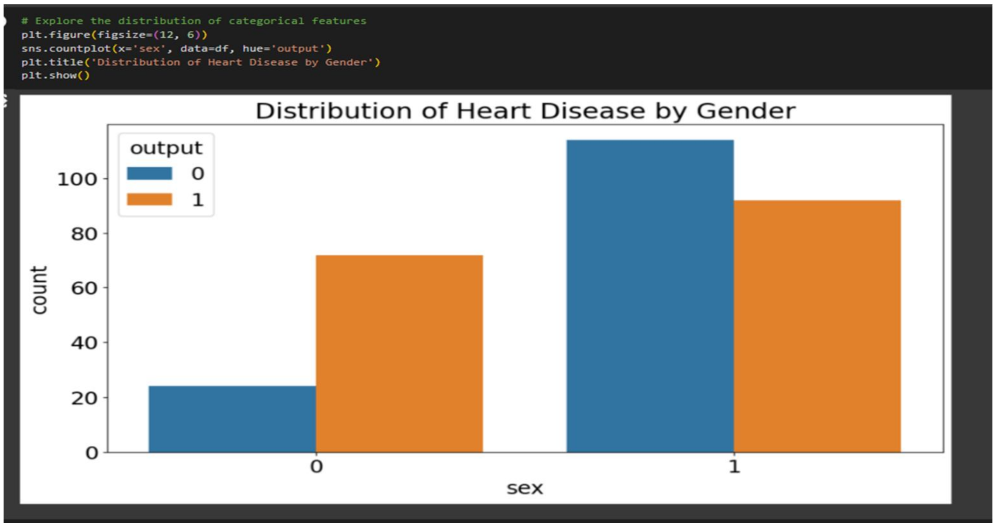
Fig-14: Distribution of resting electrocardiographic results

This Fig-14 shows that “Distribution of Resting Electrocardiographic Results”

1. Type 0 = 50%: This indicates that 50% of the observed cases had a normal ECG reading, suggesting no significant abnormalities or signs of heart disease in half of the cases.

2. 2. Type 1 = 48.7%: This suggests that 48.7% of the observed cases showed ST-T wave abnormalities on the ECG. This abnormality can be indicative of various heart conditions, such as coronary artery disease or myocardial infarction.

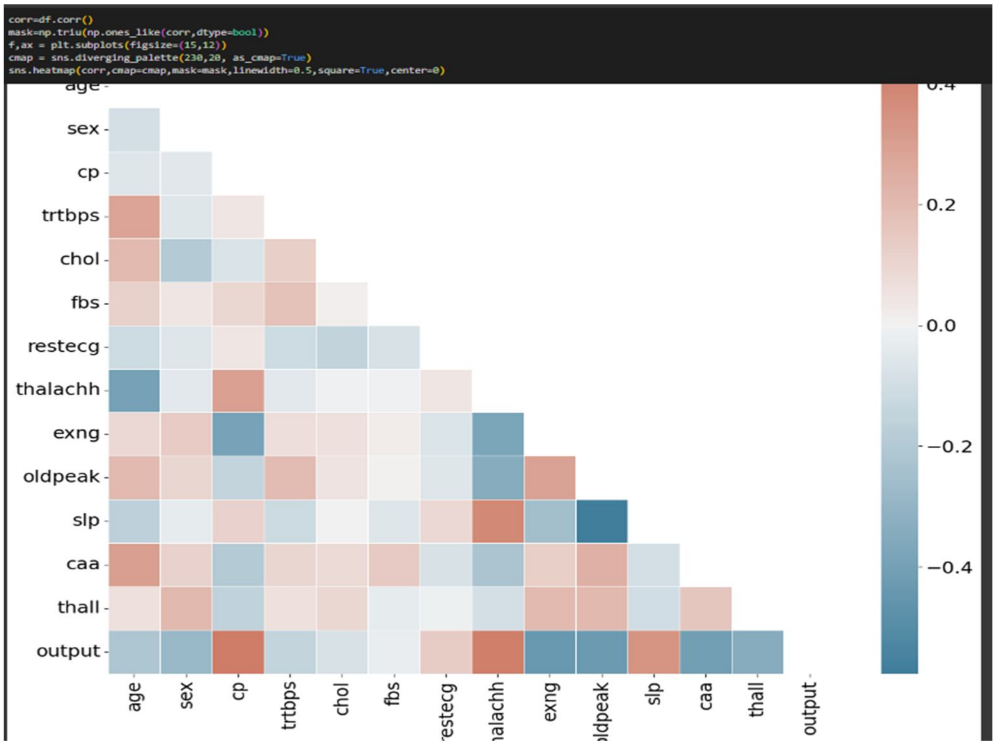
3.Type 2 = 1.3%: This indicates that only 1.3% of the observed cases exhibited left ventricular hypertrophy on the ECG. Left ventricular hypertrophy often suggests an underlying heart condition, such as hypertension or other diseases affecting the heart muscle.



```
# Explore the distribution of categorical features
plt.figure(figsize=(12, 6))
sns.countplot(x='sex', data=df, hue='output')
plt.title('Distribution of Heart Disease by Gender')
plt.show()
```

Fig-15: Distribution of exercise-include angina

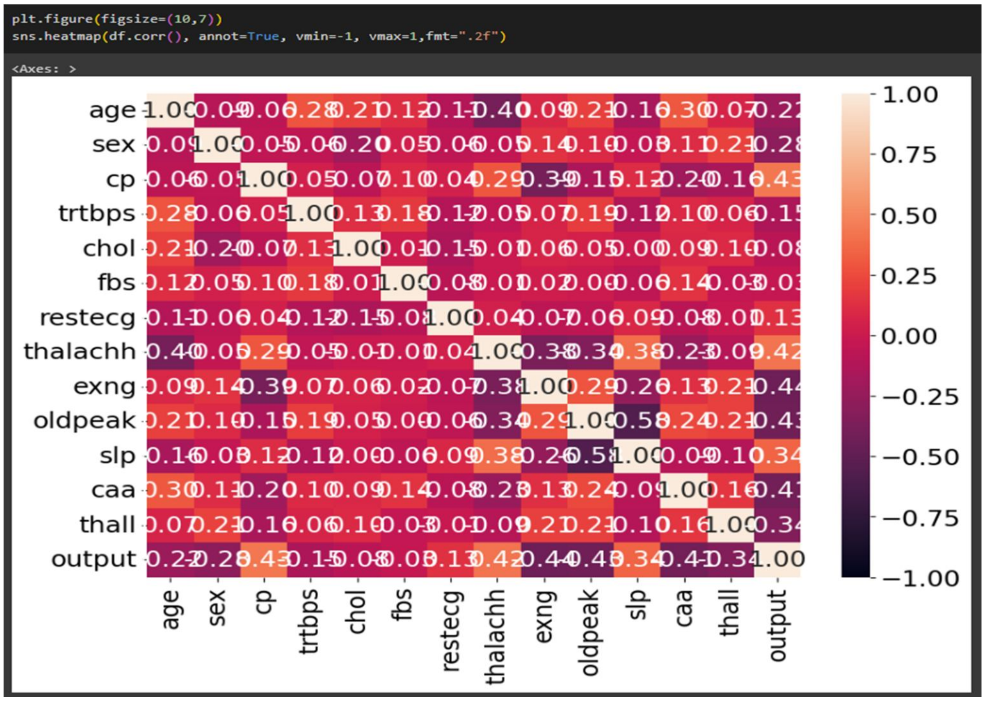
This Fig-15 shows that “Distribution of heart disease by gender”.We see that for females who are suffering from the disease are older than males.



```
corr=df.corr()
mask=np.triu(np.ones_like(corr,dtype=bool))
f,ax = plt.subplots(figsize=(15,12))
cmap = sns.diverging_palette(230,20, as_cmap=True)
sns.heatmap(corr,cmap=cmap,mask=mask,linewidth=0.5,square=True,center=0)
```

Fig-16:Diagonal correlation matrix graph

This Fig-16 shows diagonal correlation matrix of heart disease features. Typically a correlation matrix is "square" with the same variables show rows and columns. This graph also shows correlation between stated importance of various things to people. Each cell indicates correlation between 2 variables



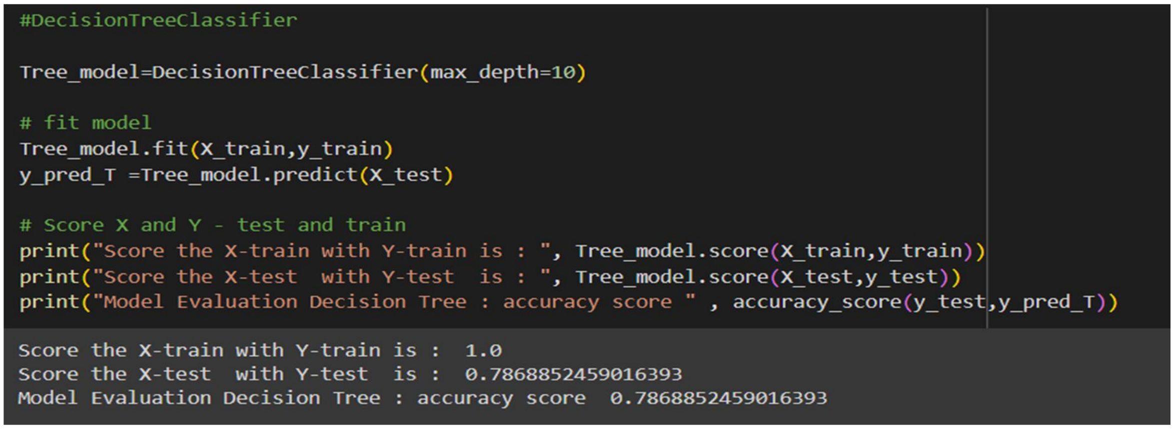
```
plt.figure(figsize=(10,7))
sns.heatmap(df.corr(), annot=True, vmin=-1, vmax=1,fmt=".2f")
```

```
<Axes: >
```

Fig-17: Heatmap of heart disease 2

This Fig-17 A figure with a size of 10x7 inches is created using plt.figure(figsize=(10,7)).

Heatmap Creation: The seaborn heatmap function is used to create a heatmap of the correlation matrix of the DataFrame df.

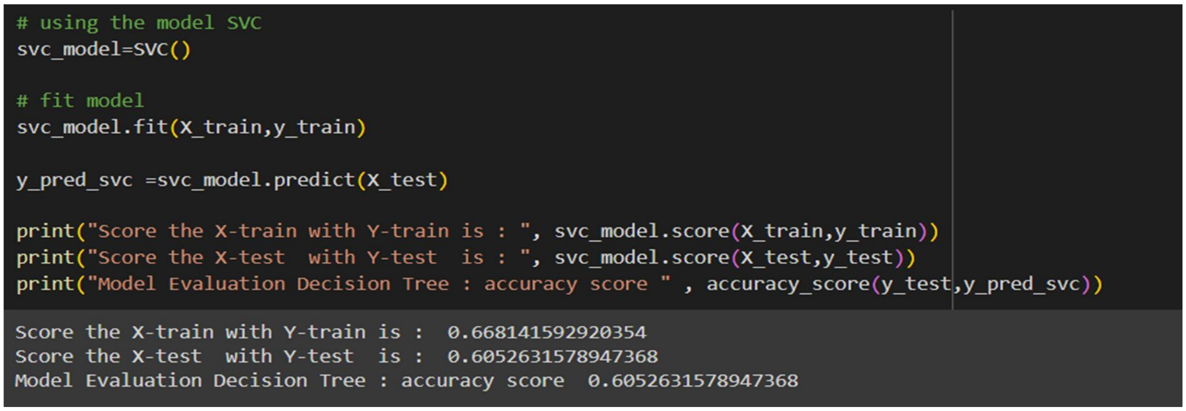


```
#DecisionTreeClassifier

Tree_model=DecisionTreeClassifier(max_depth=10)

# fit model
Tree_model.fit(X_train,y_train)
y_pred_T =Tree_model.predict(X_test)

# Score X and Y - test and train
print("Score the X-train with Y-train is : ", Tree_model.score(X_train,y_train))
print("Score the X-test  with Y-test  is : ", Tree_model.score(X_test,y_test))
print("Model Evaluation Decision Tree : accuracy score " , accuracy_score(y_test,y_pred_T))
```

```
Score the X-train with Y-train is :  1.0
Score the X-test  with Y-test  is :  0.7868852459016393
Model Evaluation Decision Tree : accuracy score  0.7868852459016393
```

Fig-18: Result of Decision Tree

This Fig-18 shows A Decision Tree Classifier model and evaluated its performance. The model achieved an accuracy score of around 78.6% on both the training and test datasets.



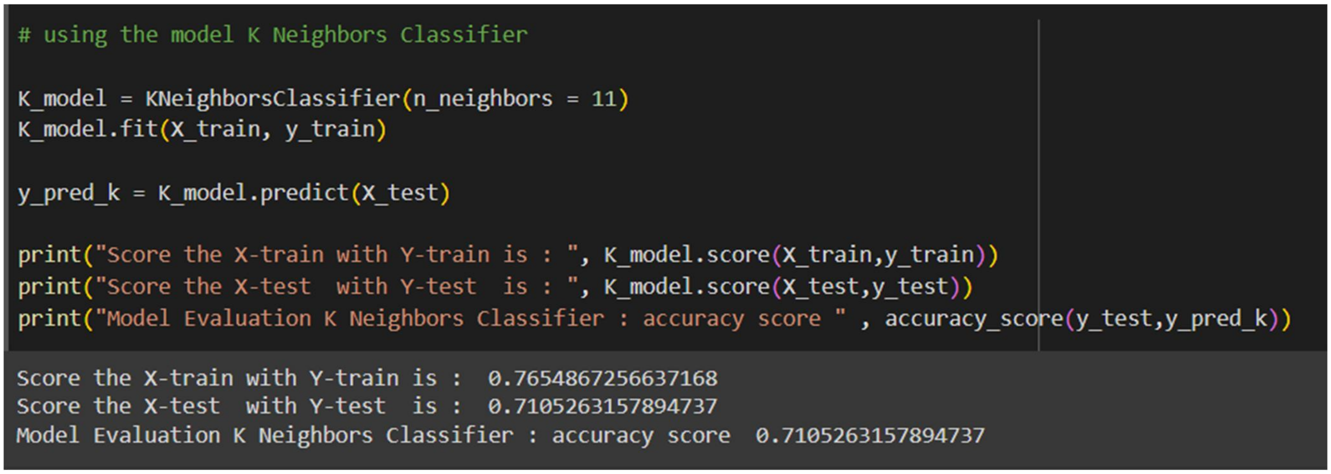
```
# using the model SVC
svc_model=SVC()

# fit model
svc_model.fit(X_train,y_train)

y_pred_svc =svc_model.predict(X_test)

print("Score the X-train with Y-train is : ", svc_model.score(X_train,y_train))
print("Score the X-test  with Y-test  is : ", svc_model.score(X_test,y_test))
print("Model Evaluation Decision Tree : accuracy score " , accuracy_score(y_test,y_pred_svc))
```

```
Score the X-train with Y-train is :  0.668141592920354
Score the X-test  with Y-test  is :  0.6052631578947368
Model Evaluation Decision Tree : accuracy score  0.6052631578947368
```

Fig-19: Result of Support Vector Classifier

In this Fig-19,appears to be implementing a Support Vector Classifier (SVC) model and evaluating its performance, the model achieved an accuracy score of around 60.5% on both the training and test datasets.

Model Selection: The code is using Support Vector Regression (SVR), which is a type of Support Vector Machine (SVM) model used for regression tasks. The output shows that the SVR model achieved a score of around 60.5% on both the training and test datasets

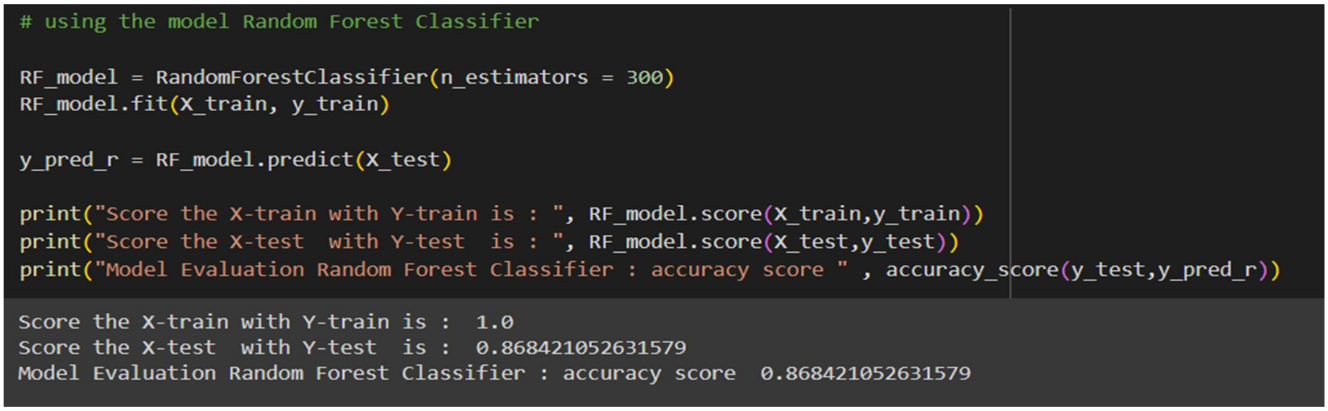


```
# using the model K Neighbors Classifier

K_model = KNeighborsClassifier(n_neighbors = 11)
K_model.fit(X_train, y_train)

y_pred_k = K_model.predict(X_test)

print("Score the X-train with Y-train is : ", K_model.score(X_train,y_train))
print("Score the X-test  with Y-test  is : ", K_model.score(X_test,y_test))
print("Model Evaluation K Neighbors Classifier : accuracy score " , accuracy_score(y_test,y_pred_k))
```

```
Score the X-train with Y-train is :  0.7654867256637168
Score the X-test  with Y-test  is :  0.7105263157894737
Model Evaluation K Neighbors Classifier : accuracy score  0.7105263157894737
```

Fig-20: Result of K Neighbors Classifier

The K Nearest Neighbors Classifier (KNN) algorithm for classification tasks: The output shows that the KNN model achieved a score of around 71.1% on both the training and test datasets.



```
# using the model Random Forest Classifier

RF_model = RandomForestClassifier(n_estimators = 300)
RF_model.fit(X_train, y_train)

y_pred_r = RF_model.predict(X_test)

print("Score the X-train with Y-train is : ", RF_model.score(X_train,y_train))
print("Score the X-test  with Y-test  is : ", RF_model.score(X_test,y_test))
print("Model Evaluation Random Forest Classifier : accuracy score " , accuracy_score(y_test,y_pred_r))
```

```
Score the X-train with Y-train is :  1.0
Score the X-test  with Y-test  is :  0.868421052631579
Model Evaluation Random Forest Classifier : accuracy score  0.868421052631579
```

Fig-21: Result of Random Forest Classifier

This code snippet employs the Random Forest Classifier algorithm for a classification task:
The output indicates that the Random Forest model achieved a perfect score of 100% on the training data, but a slightly lower score of around 86.8% on the test data.

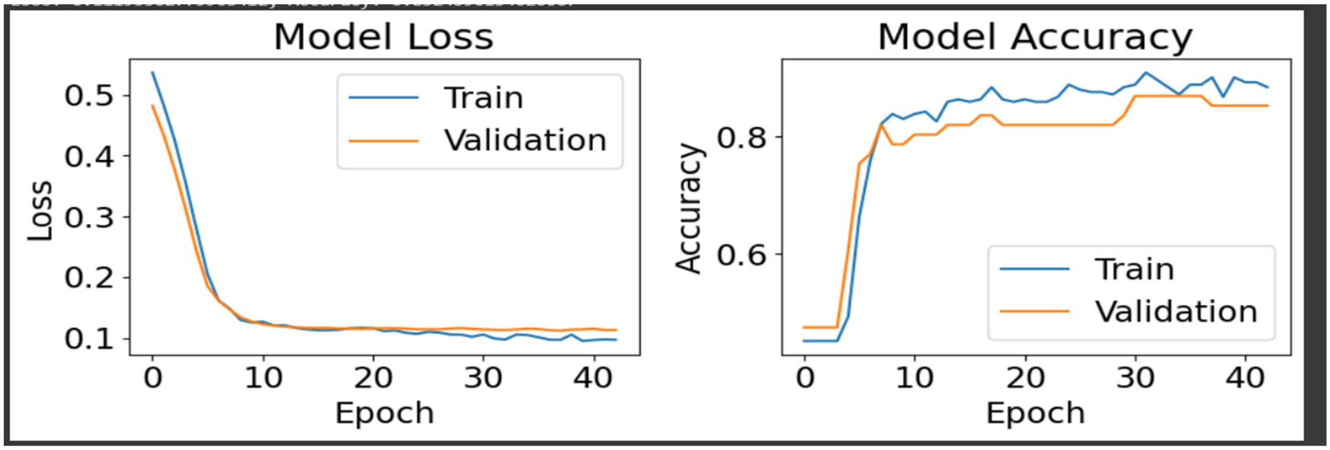
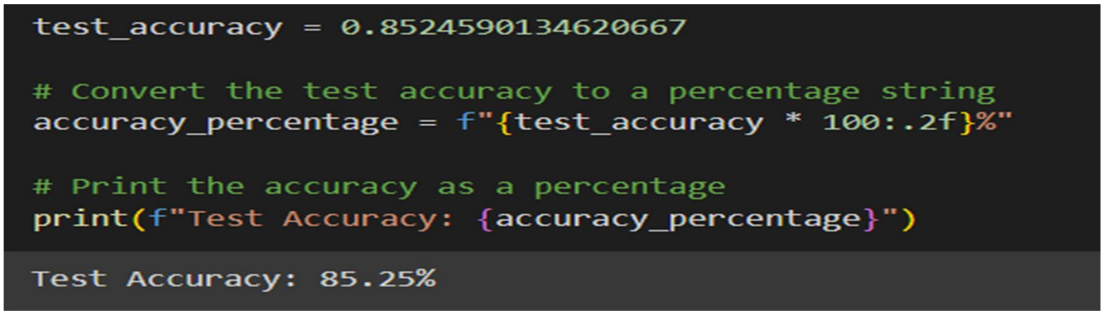


Fig-22: Graph of model loss and odel accuracy

Model Loss and Accuracy:
The y-axis represents the model loss and accuracy values.
The x-axis represents the number of epochs, indicating the training iterations.
Loss and Accuracy Trends:
The loss and accuracy are plotted against the number of epochs.
The plot shows how both training and validation loss and accuracy change over each epoch.
Interpretation:



```
test_accuracy = 0.8524590134620667

# Convert the test accuracy to a percentage string
accuracy_percentage = f"{test_accuracy * 100:.2f}%"

# Print the accuracy as a percentage
print(f"Test Accuracy: {accuracy_percentage}")
```

```
Test Accuracy: 85.25%
```

Fig 6.18: Accuracy of model

This code calculates the test accuracy and converts it into a percentage string. Test Accuracy Calculation: The variable test_accuracy stores the test accuracy value, which is approximately 0.8524590134620667.

VI. CONCLUSION

In conclusion, the utilization of machine learning approaches for the early detection of cardiac arrest presents a promising avenue for improving patient outcomes and reducing the burden of cardiovascular diseases. The proposed system harnesses the power of advanced neural network architectures, multi-modal data integration, and real-time monitoring capabilities to enable timely detection, personalized risk assessment, and proactive intervention. By continuously analyzing cardiovascular data in real-time and providing early alerts to healthcare providers or patients, the system facilitates prompt clinical response and potentially prevents adverse cardiac events. Furthermore, the system prioritizes interpretability, reliability, and scalability, ensuring its suitability for diverse healthcare settings and patient populations. Through ongoing validation, refinement, and collaboration with medical experts and stakeholders, the proposed system has the potential to revolutionize cardiac care, ultimately saving lives and improving the quality of life for individuals at risk of cardiac arrest.

REFERENCES

[1] An-Yi Wang 1,Cheng-Chung Fang 2, Shyr-Chyr Chen 2, Shin-Han Tsai 3, Wei-Fong Kao 4 Affiliations expand Epub 2015 Dec 24.

[2] Yun Gi Kim # 1, Kyongjin Min # 2, Joo Hee Jeong 1, Seung-Young Roh 3, Kyung-Do Han 4, Jaemin Shim 1, Jong-Il Choi 5, Young-Hoon Kim 1 2024 Jan 27;14(1):2289.doi: 10.1038/s41598-024-52859-x.

[3] Vahid HOUSHYARIFAR, Mehdi Chehel AMIRANI,Turk J Elec Eng & Comp Sci (2017) 25: 1541 – 155⃝3c TUB¨ ˙ITAK doi:10.3906/elk-1509-14,https://journals.tubitak.gov.tr/elektrik.

[4] Justin Chan, Thomas Rea, Shyamnath Gollakota & Jacob E. Sunshine npj Digital Medicine volume 2, Article number: 52 (2019).

[5] Ryo Ueno,Liyuan Xu, Wataru Uegami,Hiroki Matsui, Jun Okui, Hiroshi Hayashi, Toru Miyajima, Yoshiro Hayashi,David Pilcher, Daryl Jones,Published online 2020 Jul 13.

[6] Joon-Myoung Kwon 1,Youngnam Lee 2, Yeha Lee 2, Seungwoo Lee 2, Jinsik Park 2018 Jun 26;7(13):e008678. doi: 10.1161/JAHA.118.008678.

[7] Apeksha Shah1 , Swati Ahirrao1 , Sharnil Pandya1 *, Ketan Kotecha2 and Suresh Rathod1,published: 22 October 2021 doi: 10.3389/fpubh.2021.762303.

[8] Minsu ChaeHyo-Wook GilNam-Jun ChoHwamin Lee,Mathematics 2022, 10(12), 2049; https://doi.org/10.3390/math10122049 Submission received: 2 May 2022 / Revised: 1 June 2022 / Accepted: 9 June 2022 / Published: 13 June 2022.