PROJECT REPORT

(Project Term August-November 2021)

Hearth Failure Predictor

Submitted by

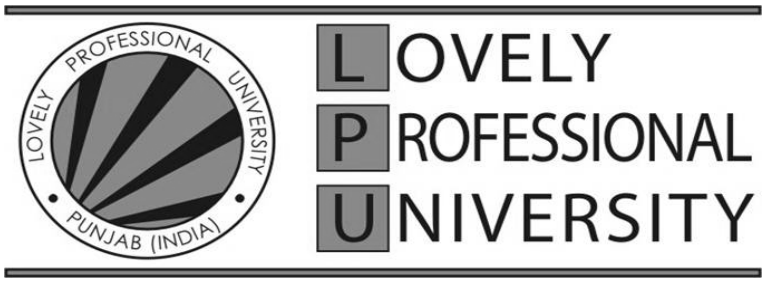
Yash kumar garg **Registration Number: 11906097**
Abhishek Bhadoria **Registration Number: 11905997**

Course Code INT246

Under the Guidance of

Dr. Sagar Pande

School of Computer Science and Engineering



DECLARATION

We hereby declare that the project work entitled Hearth Failure Predictor is an authentic record of our own work carried out as requirements of Project for the award of B.Tech degree in CSE from Lovely Professional University, Phagwara, under the guidance of Dr. Sagar Pande , during August to November 2021. All the information furnished in this project report is based on our own intensive work and is genuine.

Name of Student 1: Yash kumar garg
Registration Number: 11906097

Name of Student 2: Abhishek Bhadoria
Registration Number: 11905997

(Signature of Student 1)
Date:

(Signature of Student 2)
Date:

CERTIFICATE

This is to certify that the declaration statement made by this group of students is correct to the best of my knowledge and belief. They have completed this Project under my guidance and supervision. The present work is the result of their original investigation, effort and study. No part of the work has ever been submitted for any other degree at any University. The Project is fit for the submission and partial fulfillment of the conditions for the award of B.Tech degree in CSE from Lovely Professional University, Phagwara.

Signature and Name of the Mentor

Designation

School of Computer Science and Engineering,
Lovely Professional University,
Phagwara, Punjab.

Date : 20/11/2021

ACKNOWLEDGEMENT

I have taken efforts in this project. However, it would not have been possible without the kind support and help of many individuals and organizations. I would like to extend my sincere thanks to all of them.

I am highly indebted to Mr. Sagar Pande for their guidance and constant supervision as well as for providing necessary information regarding the project & also for their support in completing the project.

I would like to express my gratitude towards my parents & member of GeekForGeek for their kind co-operation and encouragement which help me in completion of this project.I would like to express my special gratitude and thanks to industry persons for giving me such attention and time.

My thanks and appreciations also go to my colleague in developing the project and people who have willingly helped me out with their abilities.

Name of Student 1: Yash Kumar Garg

Registration Number: 11906097

Name of Student 2: Abhishek Bhadoria

Registration Number: 11905997

TABLE OF CONTENTS

Abstract

Heart failure is a worldwide healthy problem affecting more than 550,000 people every year. A better prediction for this disease is one of the key approaches of decreasing its impact. Both linear and machine learning models are used to predict heart failure based on various data as inputs, e.g., clinical features.

Cardiovascular diseases (CVDs) are the number 1 cause of death globally, taking an estimated 17.9 million lives each year, which accounts for 31% of all deaths worlwide. Heart failure is a common event caused by CVDs and this dataset contains 12 features that can be used to predict mortality by heart failure. If you're able to make a machine learning model, then this will help in early detection and people can be saved.

You have to predict a person death event using some features:-

- Age, Gender , blood pressure, smoke, diabetes, ejection fraction, creatinine phosphokinase, serum creatinine, serum sodium, time\

Heart disease is one of the major cause of mortality in the world today. Prediction of cardiovascular disease is a critical challenge in the field of clinical data analysis. With the advanced development in machine learning (ML), artificial intelligence (AI) and data science has been shown to be effective in assisting in decision making and predictions from the large quantity of data produced by the healthcare industry.

ML approaches has brought lot of improvements and broadens the study in medical field which recognizes patterns in the human body by using various algorithms and correlation techniques. One such reality is coronary heart disease, various studies gives impression into predicting heart disease with ML techniques. Initially ML was used to find degree of heart failure, but also used to identify significant features that affects the heart disease by using correlation techniques. There are many features/factors that lead to heart disease like age, blood pressure, sodium creatinine, ejection fraction etc.

In this paper we propose a method to finding important features by applying machine learning techniques. The work is to design and develop prediction of heart disease by feature ranking machine learning. Hence ML has huge impact in saving lives and helping the doctors, widening the scope of research in actionable insights, drive complex decisions and to create innovative products for businesses to achieve key goals.

Introduction

Heart failure is a serious problem which has a huge impact on people's life. With the accelerated pace of life, increased portion sizes and inactivity, most people always ignore their health. Moreover, because of the environmental deterioration, those factors can lead to the issue of heart failure which can become more and more common in the future. If people did not pay attention to the issue of heart failure, it would finally cause the death. In the past years, different researchers used different methods to collect and analyse data with the aim to predict heart failure. These data include electronic health record (EHR) data of patients with heart failure in different hospitals from different countries, Cleveland heart disease dataset, biomedical science datasets from UCI, etc.

Patients paper chart is determined by the device that is a digital version called as an electronic health record (EHR). This information obtained is secured, real time and patient centred record which makes it easy for the medical staff. This helps to recognize hidden data and brings a correlation between patient data which can be used for clinical and research practices. This process helps in eliminating the traditions.

Cardiovascular diseases (CVDs) are the number 1 cause of death globally, taking an estimated 17.9 million lives each year. CVDs are a group of disorders of the heart and blood vessels and include coronary heart disease, cerebrovascular disease, rheumatic heart disease and other conditions. Four out of 5CVD deaths are due to heart attacks and strokes, and one third of these deaths occur prematurely in people under 70 years of age.

Millions of people worldwide struggle to control the risk factors that lead to cardiovascular disease, many others remain unaware that they are at high risk. A large number of heart attacks and strokes can be prevented by controlling major risk factors through lifestyle interventions and drug treatment where necessary. The risk factors for CVD include behavioural factors, such as tobacco use, an unhealthy diet, harmful use of alcohol and inadequate physical activity, and physiological factors, including high blood pressure (hypertension), high blood cholesterol and high blood sugar or glucose which are linked to underlying social determinants and drivers, such as ageing, income and urbanization .

Based on these data, various methods are being applied, e.g., predicting the survival of patients by utilizing classifiers of machine learning, using supervised deep learning and machine learning algorithms, training a boosted decision tree algorithm, utilizing machine intelligence-based statistical model, random under-sampling method and deep neural network models, using bioinformatic explainable deep neural network (BioExpDNN), etc. We uses machine learning model link logistic regressor , Deep forest , Random forest and Xgboost. The work of this paper can fill these inadequacies. We use accuracy as the evaluation metrics. The score reflects the robustness of the model, and the accuracy reflects the overall accuracy. At last the results of all models are compared.

Profile of the Problem. Rationale/Scope of the study

In this section, we give a short review of recent related works:--

Taking advantage of 299 patients who have cardiac failure in 2015. Those data have 13 features for example high blood pressure, sex, and smoking. The authors utilized some different classifiers of Machine Learning to forecast the proportion of survivors, and rank the features corresponding to the most important risk factors. They find serum creatinine and ejection fraction are the most important factor to forecast the proportion of survivors.

The authors use some data which are from electronic health record (EHR) and use some models about Machine Learning The results show novel machine learning models is possible to change the prediction accuracy of model. Finally, the results show age of patients, creatinine, body mass index, and levels of blood pressure were significant factors in predicting mortality within one year among heart failure patients.

The authors collect data from 5,822 hospitalized and ambulatory patients with heart failure and it included eight variables, for example, diastolic blood pressure, white blood cell count, platelets, albumin, and red blood cell distribution width. By training a boosted decision tree algorithm, a model was developed to correlate a subset of 5822 inpatient and outpatient heart failure patients with a very high or very low risk of mortality. As a result, using machine learning and off-the-shelf variables, the authors generated and validated a risk score for death in patients with heart failure that was more accurate than other risk scores compared to it.

This study used three datasets about biomedical science which are from UCI and utilized bioinformatic explainable deep neural network. They found that the results of research paper showed that the classification accuracies about CDS, CCBRDS, and HFCRDS were 92.59 percent, 100 percent, and 78.9 percent.

The authors took advantage of data which is from the medical records about cardiac failure patients. The dataset contains 299 cardiac failure patients in medical records and has clinical and lifestyle data. Different machine learning algorithms are used. The result showed that Machine learning algorithms are tools which are useful and effective to classify the records of medicine of patients with cardiac failure.

Here the scope of the project is that integration of clinical decision support with computer-based patient records could reduce medical errors, enhance patient safety, decrease unwanted practice variation, and improve patient outcome. This suggestion is promising as data modeling and analysis tools, e.g., data mining, have the potential to generate a knowledge-rich environment which can help to significantly improve the quality of clinical decisions .

Objectives

Main Objectives.

The main objective of this research is to develop a heart prediction system. The system can discover and extract hidden knowledge associated with diseases from a historical heart data set . Heart disease prediction system aims to exploit data mining techniques on medical data set to assist in the prediction of the heart diseases.

Specific Objectives.

- Provides new approach to concealed patterns in the data.
- Helps avoid human biasness.
- To implement XGboost Classifier that classifies the disease as per the input of the user.
- Reduce the cost of medical tests.

Justification.

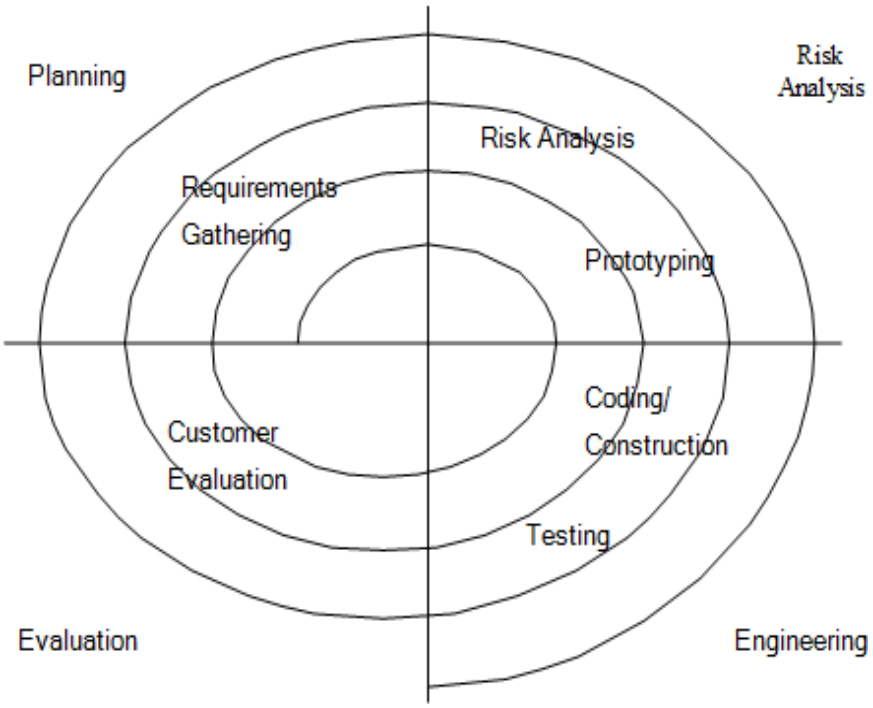
Clinical decisions are often made based on doctor's insight and experience rather than on the knowledge rich data hidden in the dataset. This practice leads to unwanted biases, errors and excessive medical costs which affects the quality of service provided to patients. The proposed system will integrate clinical decision support with computer-based patient records (Data Sets). This will reduce medical errors, enhance patient safety, decrease unwanted practice variation, and improve patient outcome. This suggestion is promising as data modeling and analysis tools, e.g., data mining, have the potential to generate a knowledge rich environment which can help to significantly improve the quality of clinical decisions.

There are voluminous records in medical data domain and because of this, it has become necessary to use data mining techniques to help in decision support and prediction in the field of healthcare. Therefore, medical data mining contributes to business intelligence which is useful for diagnosing of disease

System Development Methodology

The methodology of software development is the method in managing project development. There are many models of the methodology are available such as Waterfall model model, Incremental model, RAD model, Agile model, Iterative model and Spiral model. However, it still need to be considered by developer to decide which is will be used in the project. The methodology model is useful to manage the project efficiently and able to help developer from getting any problem during time of development. Also, it help to achieve the objective and scope of the projects. In order to build the project, it need to understand the stakeholder requirements.

Methodology provides a framework for undertaking the proposed DM modeling. The methodology is a system comprising steps that transform raw data into recognized data patterns to extract knowledge for users.



There are four phases that involve in the spiral model:

1) Planning phase

Phase where the requirement are collected and risk is assessed. This phase where the title of the project has been discussed with project supervisor. From that discussion, Heart Prediction System has been proposed. The requirement and risk was assessed after doing study on existing system and do literature review about another existing research.

2) Risk analysis Phase

Phase where the risk and alternative solution are identified. A prototype are created at the end this phase. If there is any risk during this phase, there will be suggestion about alternate solution.

3) Engineering phase

At this phase, a software are created and testing are done at the end this phase.

4) Evaluation phase

At this phase, the user do evaluation toward the software. It will be done after the system are presented and the user do test whether the system meet with their expectation and requirement or not. If there is any error, user can tell the problem about system.

SYSTEM REQUIREMENT

Tools

For application development, the following Software Requirements are:

Operating System: Windows 7 or MacOS 12
Language: Python, Flask , HTML , CSS
Tools: Terminal, Microsoft Excel, Microsoft Word

Technologies used: Github, Terminal ,Google Chrome , Heroku

Software requirements:

Any OS with clients to access the internet Wi-Fi Internet or cellular Network Create and design Data Flow and Context Diagram Versioning Control Medium to find reference to do system testing, display and run shinyApp.

Hardware Requirements

For application development, the following Software Requirements are: Processor: Intel or high
RAM: 1024 MB
Space on disk: minimum 100mb

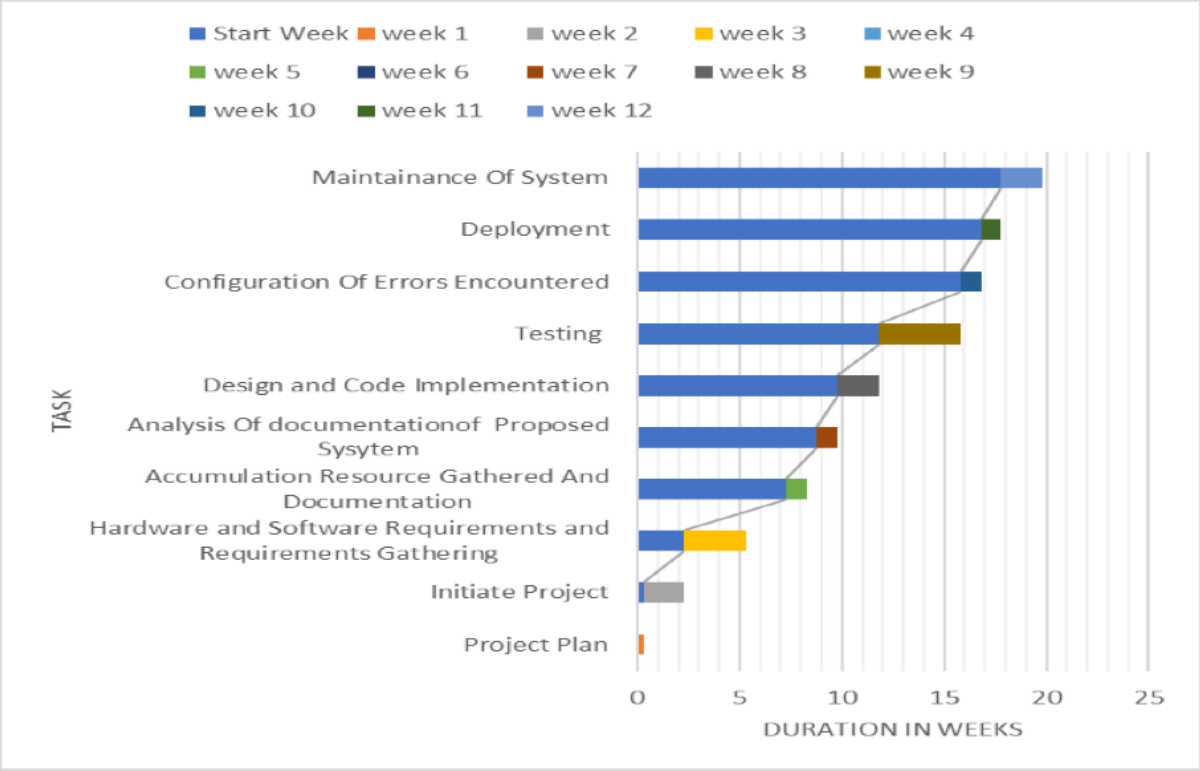
For running the application:
Device: Any device that can access the internet Minimum space to execute: 20 MB

The effectiveness of the proposal is evaluated by conducting experiments with a cluster formed by 3 nodes with identical setting, configured with an Intel CORETM i7-4770 processor (3.40GHZ, 4 Cores, 8GB RAM, running Ubuntu 18.04 LTS with 64-bit Linux 4.31.0 kernel)

Budget

The budget of completion for developing the heart disease prediction system will require various software and hardware devices. The application is averagely expensive to build but if happens to be as successful as the developer sees it to be it will bring forth enough profit to cover the costs undergone.

Work Plan



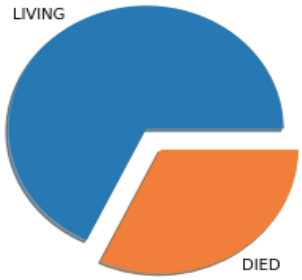
Dataset Description

We utilize a common dataset in public: https://www.kaggle.com/andrewmvd/heart-failure-clinical-data. This data contains the following input characteristics and prediction targets. The target we want to predict is the Death Event. The features have Age, anaemia, high blood pressure, creatinine phosphokinase, diabetes, ejection fraction, platelets, sex, serum creatinine, smoking, and time. Since the number of positive and negative samples is different.

```
Information about data:-
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 299 entries, 0 to 298
Data columns (total 13 columns):
 #   Column                    Non-Null Count  Dtype
---  ------                    --------------  -----
 0   age                       299 non-null    float64
 1   anaemia                   299 non-null    int64
 2   creatinine_phosphokinase  299 non-null    int64
 3   diabetes                  299 non-null    int64
 4   ejection_fraction         299 non-null    int64
 5   high_blood_pressure       299 non-null    int64
 6   platelets                 299 non-null    float64
 7   serum_creatinine          299 non-null    float64
 8   serum_sodium              299 non-null    int64
 9   sex                       299 non-null    int64
 10  smoking                   299 non-null    int64
 11  time                      299 non-null    int64
 12  DEATH_EVENT               299 non-null    int64
dtypes: float64(3), int64(10)
memory usage: 30.5 KB
```

Death event in the dataset:- The Data seems to be less as only 299 entry so it is pretty mush biased and can caused low accuracy.

```
Total No. Of Living Cases :-  203
Total No. Of Died Cases :-  96
```

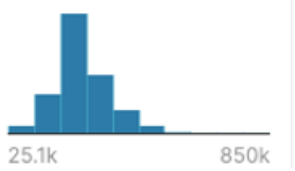


Feature Explanation

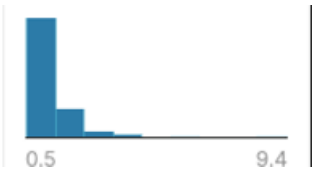
➔ Ejection Fraction- Percentage of blood leaving the heart at each contraction (percentage)

➔ High Blood Pressure- If the patient has hypertension (Boolean)

➔ Platelets- Platelets in the blood (kiloplatelets/mL)



➔ Serum Creatinine- Level of serum creatinine in the blood (mg/dL)

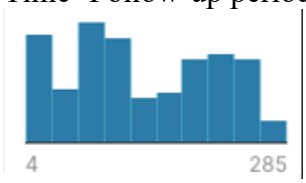


➔ Serum Sodium- Level of serum sodium in the blood (mEq/L)

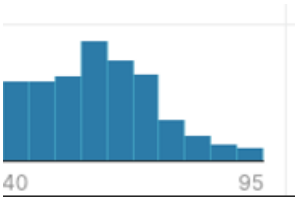
➔ Sex- Women or Men (binary)

➔ Smoking- If the patient smokes or not (Boolean)

➔ Time- Follow-up period (days)

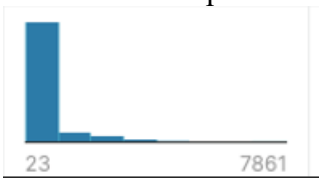


➔ Age

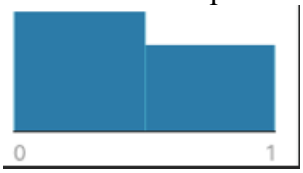


➔ Anaemia- Decrease of red blood cells or hemoglobin (Boolean)

➔ Creatinine Phosphokinase- Level of the CPK enzyme in the blood (mcg/L)



➔ Diabetes- If the patient has diabetes (Boolean)



Correlation Analysis-Pearson

Correlation techniques are done to compare the top features and then to relate how one feature is related to another feature. In this project we have related various featured with each other so that there is precise feature ranking. The first technique we used is Pearson. This technique measures the statistical relationship of the features based on the values obtained.

Followed by null analysis before proceeding the modelling correlation techniques are applied among all the feature of bug analysis using Pearson correlation

```
n square shows the correlation between the variables on each axis. Correlation ranges from -1 to +1. Values cl
close to 1 the correlation is the more positively correlated they are; that is as one increases so does the c
correlation closer to -1 is similar, but instead of both increasing one variable will decrease as the other inc
diagonals are all 1/dark green because those squares are correlating each variable to itself (so it's a perfe
plot is also symmetrical about the diagonal since the same two variables are being paired together in those s

erences:- https://en.wikipedia.org/wiki/Pearson_correlation_coefficient
```


```
can do the same as here
orr().style.background_gradient(cmap='coolwarm')
```

	age	anaemia	creatinine_phosphokinase	diabetes	ejection_fraction	high_blood_pressure	platelets	serum_creatinine	seru
age	1.000000	0.088006	-0.081584	-0.101012	0.060098	0.093289	-0.052354	0.159187	
anaemia	0.088006	1.000000	-0.190741	-0.012729	0.031557	0.038182	-0.043786	0.052174	
ine_phosphokinase	-0.081584	-0.190741	1.000000	-0.009639	-0.044080	-0.070590	0.024463	-0.016408	

We analyse dataset to understand and summarize main characteristics of the data. Here the scale of measurement is taken in a ratio of 1:1, when we correlate, r=1, it's a positive correlation hence ranks the features accordingly. It helps in seeing what the data infers even before applying any modelling or hypothesis testing tasks. Hereby we implement the correlation of the target feature priority with all other feature of the bug so that respective features are selected for the model building.

Correlation Analysis-kendall

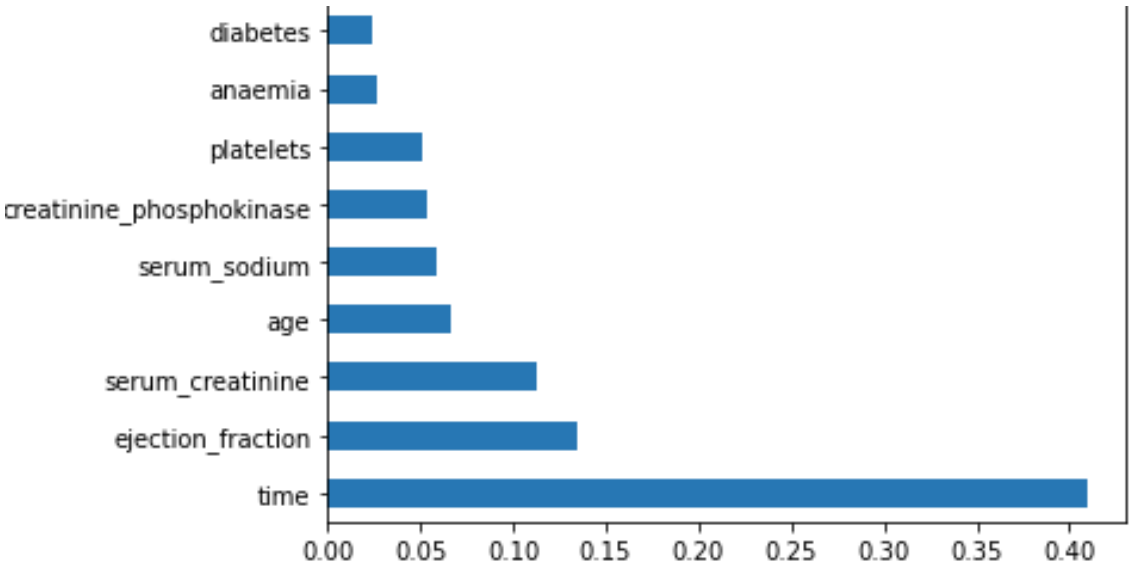
In Kendall rank correlation we use a coefficient to measure the ordinal association between two measured features, here the rank comes high when features have similar rank or features. Followed by null analysis before proceeding the modeling correlation techniques are applied among all the feature of bug analysis using kendall correlation

describing the data:-

	age	anaemia	creatinine_phosphokinase	diabetes	ejection_fraction	high_blood_pressure	platelets	serum_creatinine	serum_sodium	
count	299.000000	299.000000	299.000000	299.000000	299.000000	299.000000	299.000000	299.00000	299.000000	299.00
mean	60.833893	0.431438	581.839465	0.418060	38.083612	0.351171	263358.029264	1.39388	136.625418	0.64
std	11.894809	0.496107	970.287881	0.494067	11.834841	0.478136	97804.236869	1.03451	4.412477	0.47
min	40.000000	0.000000	23.000000	0.000000	14.000000	0.000000	25100.000000	0.50000	113.000000	0.00
25%	51.000000	0.000000	116.500000	0.000000	30.000000	0.000000	212500.000000	0.90000	134.000000	0.00
50%	60.000000	0.000000	250.000000	0.000000	38.000000	0.000000	262000.000000	1.10000	137.000000	1.00
75%	70.000000	1.000000	582.000000	1.000000	45.000000	1.000000	303500.000000	1.40000	140.000000	1.00
max	95.000000	1.000000	7861.000000	1.000000	80.000000	1.000000	850000.000000	9.40000	148.000000	1.00

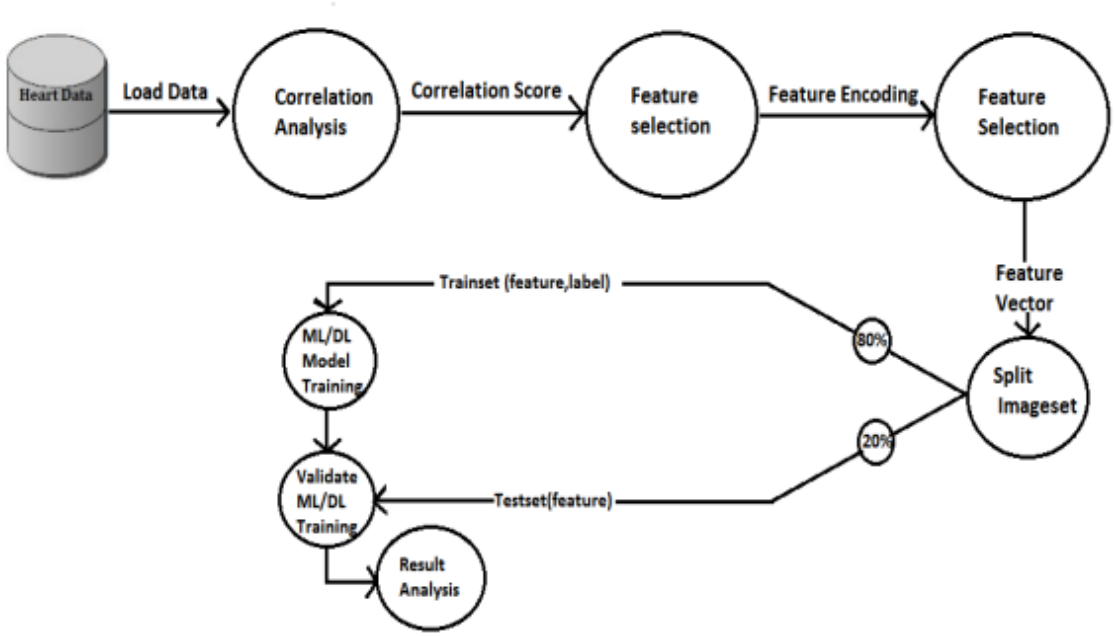
Here every feature is correlated with each other and as show in the fig 7, there is a rank given on how one feature may be responsible for presence of another. Hereby we implement the correlation of the target feature priority Fig 8 with all other feature of the bug so that respective feature can be selected for the model building

Data feature importance

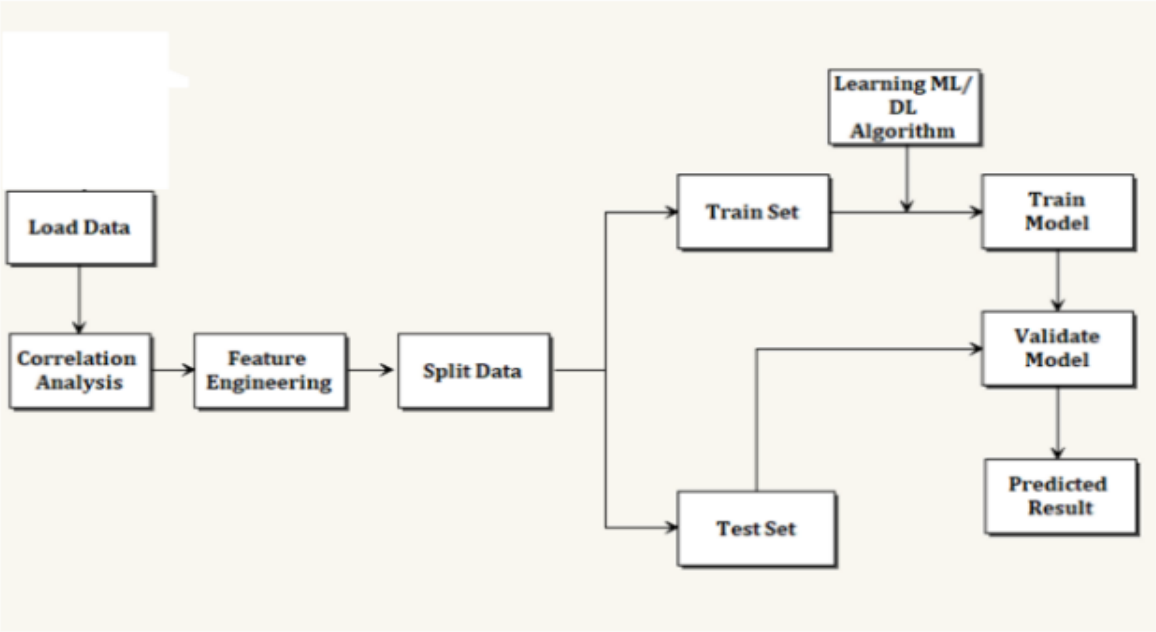


Here time is more dominant over all other features and this is the real problem about the data while all other feature share approx equal participation in the model.

DESIGN AND IMPLEMETATION



The data flow is relatively designed for the proper analysis of data. The data is loaded for correlation analysis, after finding relation between each feature correlation score is found out, based on the correlation score the features are ranked. Machine learning models are applied on the features selected. System design is an important development for any proposed system to be productive. Noteworthy level arrangement gives the system's raised level of perspective and functionality. The proposed framework and the experiment workflow with dataset.



MODELS AND EXPERIMENTS

I want to introduce all machine learning models I use in my research paper. With the development of big data and artificial intelligence, machine learning has been successful in a series of problems. For heart failure prediction, there are also some relevant studies as we introduced in the related work part. However, there lacks a comprehensive comparison between different machine learning models. Logistic Regression (LR): Logistic regression which is a classification model, and it is often used in dichotomy. Logistic Regression is one of algorithms of ML to solving binary (0 or 1) problems, which is used to estimate the likelihood of some things

Linear Regression-

Linear regression is one of the easiest and most popular Machine Learning algorithms. It is a statistical method that is used for predictive analysis. Linear regression makes predictions for continuous/real or numeric variables such as sales, salary, age, product price, etc.

Linear regression algorithm shows a linear relationship between a dependent (y) and one or more independent (y) variables, hence called as linear regression. Since linear regression shows the linear relationship, which means it finds how the value of the dependent variable is changing according to the value of the independent variable.

The linear regression model provides a sloped straight line representing the relationship between the variables. Consider the below image:

Mathematically, we can represent a linear regression as:

$$y= a_0+a_1x+ \varepsilon$$

```
# building logistic regression model as a baseline model

from sklearn.linear_model import LogisticRegression

lr_clf = LogisticRegression(max_iter=1000)
lr_clf.fit(X_train, y_train)

lr_clf_pred = lr_clf.predict(X_test)
```

```
y_pred = lr_clf.predict(X_test)
evaluating_model(y_test, y_pred)
score = lr_clf.score(X_test, y_test)
scores.append(score)
```

```
Accuracy Score:-  0.7888888888888889
Precision Score:-  0.7647058823529411
Recall Score:-  0.4642857142857143
Confusion Matrix:-
 [[58  4]
 [15 13]]
```

Standard Scaler-

In Machine Learning, StandardScaler is used to resize the distribution of values so that the mean of the observed values is 0 and the standard deviation is 1. In this article, I will walk you through how to use StandardScaler in Machine Learning.

StandardScaler is an important technique that is mainly performed as a preprocessing step before many machine learning models, in order to standardize the range of functionality of the input dataset.

Some machine learning practitioners tend to standardize their data blindly before each machine learning model without making the effort to understand why it should be used, or even whether it is needed or not. So you need to understand when you should use the StandardScaler to scale your data.
StandardScaler comes into play when the characteristics of the input dataset differ greatly between their ranges, or simply when they are measured in different units of measure.

StandardScaler removes the mean and scales the data to the unit variance. However, outliers have an influence when calculating the empirical mean and standard deviation, which narrows the range of characteristic values.

These differences in the initial features can cause problems for many machine learning models. For example, for models based on the calculation of distance, if one of the features has a wide range of values, the distance will be governed by that particular characteristic.

The idea behind the StandardScaler is that variables that are measured at different scales do not contribute equally to the fit of the model and the learning function of the model and could end up creating a bias.

```
# building logistic regression with StandardScaler

from sklearn.pipeline import make_pipeline
from sklearn.preprocessing import StandardScaler

lr_clf_pip = make_pipeline(StandardScaler(), LogisticRegression())
lr_clf_pip.fit(X_train, y_train)

y_pred1 = lr_clf_pip.predict(X_test)
evaluating_model(y_test,y_pred1)
score = lr_clf_pip.score(X_test, y_test)
scores.append(score)
```

```
Accuracy Score:-  0.8111111111111111
Precision Score:-  0.7894736842105263
Recall Score:-  0.5357142857142857
Confusion Matrix:-
 [[58  4]
 [13 15]]
```

```
from sklearn.svm import SVC
```

Grid Search CV-

Support Vector Machine or SVM is one of the most popular Supervised Learning algorithms, which is used for Classification as well as Regression problems. However, primarily, it is used for Classification problems in Machine Learning.

The goal of the SVM algorithm is to create the best line or decision boundary that can segregate n-dimensional space into classes so that we can easily put the new data point in the correct category in the future. This best decision boundary is called a hyperplane.

SVM chooses the extreme points/vectors that help in creating the hyperplane. These extreme cases are called as support vectors, and hence algorithm is termed as Support Vector Machine. Consider the below diagram in which there are two different categories that are classified using a decision boundary or hyperplane

```
from sklearn.svm import SVC
from sklearn.model_selection import GridSearchCV

# defining parameter range
param_grid = {'C': [0.1, 1, 10, 100, 1000],
              'gamma': [1, 0.1, 0.01, 0.001, 0.0001],
              'kernel': ['rbf']}
grid = GridSearchCV(SVC(), param_grid, refit=True, verbose =3)
grid.fit(X_train, y_train)
```

```
grid.best_estimator_
```

```
SVC(C=10, gamma=0.0001)
```

```
svc = SVC(C = 10, gamma = 0.0001)
svc.fit(X_train, y_train)
y_pred2 = svc.predict(X_test)
evaluating_model(y_test, y_pred2)
score = svc.score(X_test, y_test)
scores.append(score)
```

```
Accuracy Score:-  0.6777777777777778
Precision Score:-  0.4
Recall Score:-  0.07142857142857142
Confusion Matrix:-
 [[59  3]
 [26  2]]
```

Decision tree-

- Decision Tree is a Supervised learning technique that can be used for both classification and Regression problems, but mostly it is preferred for solving Classification problems. It is a tree-structured classifier, where internal nodes represent the features of a dataset, branches represent the decision rules and each leaf node represents the outcome.
- In a Decision tree, there are two nodes, which are the Decision Node and Leaf Node. Decision nodes are used to make any decision and have multiple branches, whereas Leaf nodes are the output of those decisions and do not contain any further branches.
- The decisions or the test are performed on the basis of features of the given dataset.
- *It is a graphical representation for getting all the possible solutions to a problem/decision based on given conditions.*
- It is called a decision tree because, similar to a tree, it starts with the root node, which expands on further branches and constructs a tree-like structure.
- In order to build a tree, we use the CART algorithm, which stands for Classification and Regression Tree algorithm.
- A decision tree simply asks a question, and based on the answer (Yes/No), it further split the tree into subtrees.

```
ds_clf = DecisionTreeClassifier(max_depth=8, max_features=0.9, max_leaf_nodes=30,
                        min_impurity_decrease=0.05, min_samples_leaf=0.02,
                        min_samples_split=10, min_weight_fraction_leaf=0.005,
                        random_state=2, splitter='random')
ds_clf.fit(X_train, y_train)
pred4 = ds_clf.predict(X_test)
evaluating_model(y_test, pred4)
score = ds_clf.score(X_test, y_test)
scores.append(score)
```

```
Accuracy Score:-  0.8111111111111111
Precision Score:-  0.72
Recall Score:-  0.6428571428571429
Confusion Matrix:-
 [[55  7]
 [10 18]]
```

Random forest-

Random Forest is a popular machine learning algorithm that belongs to the supervised learning technique. It can be used for both Classification and Regression problems in ML. It is based on the concept of **ensemble learning,** which is a process of combining multiple classifiers to solve a complex problem and to improve the performance of the model.

As the name suggests, "Random Forest is a classifier that contains a number of decision trees on various subsets of the given dataset and takes the average to improve the predictive accuracy of that dataset." Instead of relying on one decision tree, the random forest takes the prediction from each tree and based on the majority votes of predictions, and it predicts the final output.

The greater number of trees in the forest leads to higher accuracy and prevents the problem of overfitting.

```
: rf_clf = RandomForestClassifier(max_depth=2, max_features=0.5,
                        min_impurity_decrease=0.01, min_samples_leaf=10,
                        random_state=2)
  rf_clf.fit(X_train, y_train)
  pred5 = rf_clf.predict(X_test)
  evaluating_model(y_test, pred5)
  score = rf_clf.score(X_test, y_test)
  scores.append(score)
```

```
Accuracy Score:-  0.8666666666666667
Precision Score:-  0.9
Recall Score:-  0.6428571428571429
Confusion Matrix:-
 [[60  2]
 [10 18]]
```

Gradient boosting-

Gradient boosting classifiers are a group of machine learning algorithms that combine many weak learning models together to create a strong predictive model. Decision trees are usually used when doing gradient boosting. Gradient boosting models are becoming popular because of their effectiveness at classifying complex datasets, and have recently been used to win many Kaggle data science competitions.

The Python machine learning library, Scikit-Learn, supports different implementations of gradient boosting classifiers, including XGBoost.

In this article we'll go over the theory behind gradient boosting models/classifiers, and look at two different ways of carrying out classification with gradient boosting classifiers in Scikit-Learn.

```
from sklearn.ensemble import GradientBoostingClassifier

gbdt = GradientBoostingClassifier(n_estimators=200, learning_rate=0.1,max_depth=1,random_state=0)
gbdt.fit(X_train, y_train)

pred_gdbt = gbdt.predict(X_test)
evaluating_model(y_test, pred_gdbt)
score = gbdt.score(X_test, y_test)
scores.append(score)
```

```
Accuracy Score:-  0.8555555555555555
Precision Score:-  0.8571428571428571
Recall Score:-  0.6428571428571429
Confusion Matrix:-
 [[59  3]
 [10 18]]
```

Xgboost-

XGBoost is an optimized distributed gradient boosting library designed to be highly **efficient**, **flexible** and **portable**. It implements machine learning algorithms under the Gradient Boosting framework. XGBoost provides a parallel tree boosting (also known as GBDT, GBM) that solve many data science problems in a fast and accurate way. The same code runs on major distributed environment (Hadoop, SGE, MPI) and can solve problems beyond billions of examples.

```
: from xgboost import XGBClassifier
  xgb1 = XGBClassifier(colsample_bytree= 1.0, learning_rate = 0.1,max_depth = 4,n_estimators= 400,subsample= 1.0)
  eval_set  = [(X_test, y_test)]
  xgb1.fit(X_train, y_train)
```

```
: pred6 = xgb1.predict(X_test)
  evaluating_model(y_test, pred6)
  score = xgb1.score(X_test, y_test)
  scores.append(score)
```

```
Accuracy Score:-  0.8666666666666667
Precision Score:-  0.7857142857142857
Recall Score:-  0.7857142857142857
Confusion Matrix:-
 [[56  6]
 [ 6 22]]
```

COMPARE ALL MODELS

Accuracy Comparison :

```
scores
```

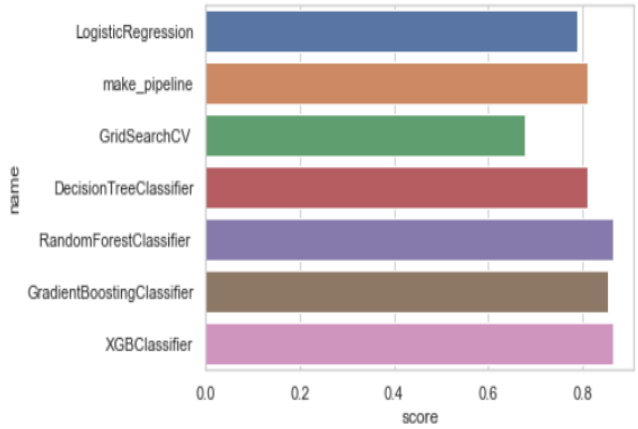
```
[0.7888888888888889,
 0.8111111111111111,
 0.6777777777777778,
 0.8111111111111111,
 0.8666666666666667,
 0.8555555555555555,
 0.8666666666666667]
```

```
df = pd.DataFrame()
df['name'] = names
df['score'] = scores
df
```

	name	score
0	LogisticRegression	0.788889
1	make_pipeline	0.811111
2	GridSearchCV	0.677778
3	DecisionTreeClassifier	0.811111
4	RandomForestClassifier	0.866667
5	GradientBoostingClassifier	0.855556
6	XGBClassifier	0.866667

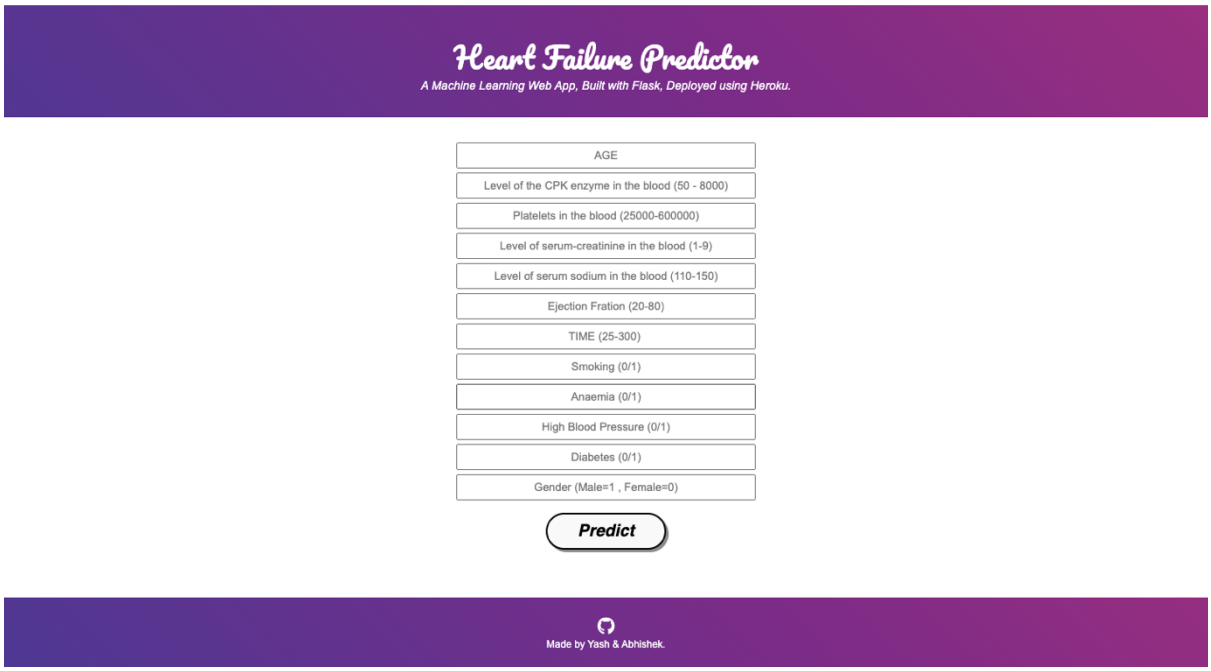
Pictorial Representation:

```
sns.set(style="whitegrid")
ax = sns.barplot(y="name", x="score", data=df)
```



HEART FAILURE PREDICTOR UI(WEB UI)

INPUT SCREEN:



This User Interface is made using HTML5 and CSS3 and backend is made using Flask and web is deployed using Heroku. We made this Website for easy user interaction So every one can use this to get a feedback about their health and to act on it as soon as possible . We are taking data from user and feeding that data to our Machine learning model and according to its learning it predict an answer and the answer is shown shown to user as output screen.

OUTPUT SCREEN:

Source Code for UI:

HTML5: Hypertext Markup Language revision 5 (HTML5) is markup language for the structure and presentation of World Wide Web contents. HTML5 supports the traditional HTML and XHTML-style syntax and other new features in its markup, New APIs, XHTML and error handling.

```
        <meta charset="utf-8">
        <title>Heart Failure Predictor</title>
        <link rel="stylesheet" type="text/css" href="{{ url_for('static', filename='styles.css') }}" >
        <script src="https://kit.fontawesome.com/5f3f547070.js" crossorigin="anonymous"></script>
        <link href="https://fonts.googleapis.com/css2?family=Pacifico&display=swap" rel="stylesheet">
    </head>

    <body>

        <!-- Website Title -->
        <div class="container">
            <h2 class='container-heading'><span class="heading_font">Heart Failure Predictor</span></h2>
            <div class='description'>
                <p>A Machine Learning Web App, Built with Flask, Deployed using Heroku.</p>
            </div>
        </div>

        <!-- Text Area -->
        <div class="ml-container">
            <form action="{{ url_for('predict') }}" method="POST">
                <input class="form-input" type="text" name="age" placeholder="AGE"><br>
                <input class="form-input" type="text" name="CPK" placeholder="Level of the CPK enzyme in the blood (50 - 8000)"><
                <input class="form-input" type="text" name="platelets" placeholder="Platelets in the blood (25000-600000)"><br>
                <input class="form-input" type="text" name="SC" placeholder="Level of serum-creatinine in the blood (1-9)"><br>
                <input class="form-input" type="text" name="SS" placeholder="Level of serum sodium in the blood (110-150)"><br>
                <input class="form-input" type="text" name="EF" placeholder="Ejection Fration(20-80)"><br>
                <input class="form-input" type="text" name="time" placeholder="TIME(25-300)"><br>
                <input class="form-input" type="text" name="Smoking" placeholder="Smoking(0/1)"><br>
                <input class="form-input" type="text" name="anaemia" placeholder="Anaemia(0/1)"><br>
                <input class="form-input" type="text" name="bloodpressure" placeholder="High Blood Pressure (0/1)"><br>
                <input class="form-input" type="text" name="Diabetes" placeholder="Diabetes(0/1)"><br>
                <input class="form-input" type="text" name="Gender" placeholder="Gender(Male=1 , Female=0)"><br>

                <input type="submit" class="my-cta-button" value="Predict">
            </form>
        </div>

        <!-- Footer -->
       <div class='footer'>
           <div class="contact">
               <a target="_blank" href="https://github.com/yash662001garg"><i class="fab fa-github fa-lg contact-icon"></i></a>
           </div>
           <p class='footer-description'>Made by Yash & Abhishek.</p>
       </div>

    </body>
```

CSS3: CSS stands for Cascading Style Sheets . CSS describes how HTML elements are to be displayed on screen, paper, or in other media. It saves a lot of work. It can control the layout of multiple web pages all at once .External stylesheets are stored in CSS files

```html
<!DOCTYPE html>

<html lang="en" dir="ltr">
    <head>
        <meta charset="utf-8">
        <title>Heart Failure Predictor</title>
        <link rel="stylesheet" type="text/css" href="{{ url_for('static', filename='styles.css') }}" >
        <script src="https://kit.fontawesome.com/5f3f547070.js" crossorigin="anonymous"></script>
        <link href="https://fonts.googleapis.com/css2?family=Pacifico&display=swap" rel="stylesheet">
    </head>

    <body>

        <!-- Website Title -->
        <div class="container">
            <h2 class='container-heading'><span class="heading_font">Heart Failure Predictor</span></h2>
            <div class='description'>
                <p>A Machine Learning Web App, Built with Flask, Deployed using Heroku.</p>
            </div>
        </div>

        <!-- Result -->
        <div class="results">
            {% if prediction==1 %}
                <h1>Prediction: <span class='danger'>Contact with nearby Heart doctor </span></h1>
            {% elif prediction==0 %}
                <h1>Prediction: <span class='safe'>Great! You are Fine.</span></h1>
            {% endif %}
        </div>

        <!-- Footer -->
       <div class='footer'>
           <div class="contact">
               <a target="_blank" href="https://github.com/yash662001garg"><i class="fab fa-github fa-lg contact-icon"></i></a>
           </div>
           <p class='footer-description'>Made by Yash & Abhishek.</p>
       </div>

    </body>
</html>

```

Flask: Flask is a web framework. This means flask provides you with tools, libraries and technologies that allow you to build a web application. This web application can be some web pages, a blog, a wiki or go as big as a web-based calendar application or a commercial website.

```python
from flask import Flask, render_template, request
import pickle
import numpy as np
import pandas as pd


filename = 'model.pkl'
classifier = pickle.load(open(filename, 'rb'))



app = Flask(__name__)


@app.route('/')
def home():
    return render_template('index.html')



@app.route('/predict', methods=['POST'])
def predict():
    if request.method == 'POST':
        age = int(request.form['age'])
        cpk = int(request.form['CPK'])
        plate = float(request.form['platelets'])
        sc = float(request.form['SC'])
        ss = int(request.form['SS'])
        ef = int(request.form['EF'])
        time = int(request.form['time'])
        smoke = int(request.form['Smoking'])
        anae = int(request.form['anaemia'])
        pressure = int(request.form['bloodpressure'])
        dia = int(request.form['Diabetes'])
        sex = int(request.form['Gender'])

        data = np.array([[age,anae,cpk,dia,ef,pressure,plate,sc,ss,sex,smoke,time]])
        my_prediction = classifier.predict(data)

        return render_template('result.html', prediction=my_prediction)



if __name__ == '__main__':
    app.run(debug=True)

```

CONCLUSION AND FUTURE WORK

The proposed system is GUI-based, user-friendly, scalable, reliable and an expandable system. The proposed working model can also help in reducing treatment costs by providing Initial diagnostics in time. The model can also serve the purpose of training tool for medical students and will be a soft diagnostic tool available for physician and cardiologist. General physicians can utilize this tool for initial diagnosis of cardio-patients. There are many possible improvements that could be explored to improve the scalability and accuracy of this prediction system. As we have developed a generalized system, in future we can use this system for the analysis of different data sets. The performance of the health's diagnosis can be improved significantly by handling numerous class labels in the prediction process, and it can be another positive direction of research. In DM warehouse, generally, the dimensionality of the heart database is high, so identification and selection of significant attributes for better diagnosis of heart disease are very challenging tasks for future research.

Machine learning techniques were used in this work to process raw data and provide a new and novel discernment towards heart disease. Heart disease prediction is challenging and very important in the medical field. However, the mortality rate can be drastically controlled if the disease is detected at the early stages and preventative measures are adopted as soon as possible.

However, in future we improve single model approach for both priority and severity prediction of the bug reports. Finally, performance evaluation is carried out in terms of accuracy, precision and recall metrics and their performance is comparable. XGboost proved to be quite accurate in the prediction of heart disease. The future course of this research can be performed with diverse mixtures of machine learning techniques to better prediction techniques. Furthermore, new feature selection methods can be developed to get a broader perception of the features to increase the performance of heart failure prediction.

BIBLIOGRAPHY

1. https://machinelearningmastery.com
2. https://ai.google
3. https://towardsdatascience.com
4. https://youtube.com
5. https://www.tensorflow.org
6. https://www.javatpoint.com
7. https://www.stackoverflow.com
8. https://www.quora.com
9. https://www.analyticsvidhya.com