A PROJECT REPORT

on

"HEART FAILURE PREDICTION"

Submitted to

KIIT Deemed to be University

In Partial Fulfilment of the Requirement for the Award of

BACHELOR'S DEGREE IN
INFORMATION TECHNOLOGY

BY

PRATYUSH AANAND	1905189
ARYAN SARRAF	1905599
ASHUTOSH MISHRA	1905600
SAMBHAV CHOUDHARY	1905634

UNDER THE GUIDANCE OF
SANKALP NAYAK

SCHOOL OF COMPUTER ENGINEERING

KALINGA INSTITUTE OF INDUSTRIAL TECHNOLOGY

BHUBANESWAR, ODISHA - 751024

May 2020

A PROJECT REPORT

on

"HEART FAILURE PREDICTION"

Submitted to

KIIT Deemed to be University

In Partial Fulfilment of the Requirement for the Award of

BACHELOR'S DEGREE IN INFORMATION TECHNOLOGY

BY

PRATYUSH AANAND	1905189
ARYAN SARRAF	1905599
ASHUTOSH MISHRA	1905600
SAMBHAV CHOUDHARY	1905634

UNDER THE GUIDANCE OF

SANKALP NAYAK

SCHOOL OF COMPUTER ENGINEERING

KALINGA INSTITUTE OF INDUSTRIAL TECHNOLOGY

BHUBANESWAE, ODISHA -751024

May 2022

KIIT Deemed to be University

School of Computer Engineering
Bhubaneswar, ODISHA 751024

CERTIFICATE

This is certify that the project entitled

"HEART FAILURE PREDICTION "

submitted by

PRATYUSH AANAND	1905189
ARYAN SARRAF	1905599
ASHUTOSH MISRHA	1905600
SAMBHAV CHOUDHARY	1905634

is a record of bonafide work carried out by them, in the partial fulfilment of the requirement for the award of Degree of Bachelor of Engineering (Computer Science & Engineering OR Information Technology) at KIIT Deemed to be university, Bhubaneswar. This work is done during year 2022-2023, under our guidance.

Date: 18/04/2022

(Guide Name)

SANKALP NAYAK

Acknowledgements

We are profoundly grateful to **SANKALP NAYAK** of ***School of Computer Engineering, KIIT, BBSR*** for his expert guidance and continuous encouragement throughout to see that this project rights its target since its commencement to its completion. .....

PRATYUSH AANAND
ARYAN SARRAF
ASHUTOSH MISHRA
SAMBHAV CHOUDHARY

ABSTRACT

Due to fast pace of modern lifestyle, catching up is given a lot preference over health. Maintaining a healthy life in this day and age is quite an impossible task, in a society of hustle culture, exercise and proper diet being neglected in favour of sitting whole day for work and more readily available junk food. Leading us to a society crippled with heart disease. CVD is taking 17.9 million life per year. As they say prevention is better than cure, so a need for accurate measure to judge the likelihood of a person getting heart attack is the need of the hour.

That's where ML comes in, it has it flaw but for now we can use it to analyse the health record, in order to see what are the condition and features that lead to heart failure. After training classification models on previous records, we can test it on new data. Here we have use many of classification models like Logistic Regression, Naïve Bayes, SVM, Decision Tree, Random Forrest Classification etc on dataset provided by Kaggle having 12 attributes. we compared the algorithm and found Random Forest with hyper tunning to be the best performing model with an accuracy of 87.94%. But there is a lot to be desired as the accuracy we got best represent how well the model trained on the current dataset.

Keywords:
Heart Disease , Machine Learning, classification models, Artificial neural network and xgb boost

Contents

List of Figures

Chapter 1
Introduction

Heart Failure can simply be defined as the inadequacy of the heart to pump blood which causes congestion in lungs thereby reducing the oxygen level required to function properly. Heart failure can be caused by various ailments as well as stress, In fact stress can be attributed as one of the major cause of heart failure today, other factors that contribute towards the rising cases of heart failure is rising obesity rates worldwide and the unhealthy lifestyle and eating habits also contribute to the same. The data suggests that heart diseases are on the rise for every age group and causing millions of deaths and misery .Heart failure can be more fatal with age as recent data suggests that 2 in 10 cardiac arrests results in death for adults over 65.Well this might be the case statistically the chances of survival are dependent on the magnitude of the attack, it's not uncommon that even for a young adult a high intensity heart attack can lead to death

Most of times heart attacks are sudden and unpreparable, this is mostly due to the fact that the patients themselves do not realize they might be in danger. Heart treatments are one of the most expensive procedures and global expenditure exceeds $346 billion. And despite the high cost the chances of survival are slim as compared to other medical operational procedures.

So Due to the above mentioned facts, fast diagnosis ,risk assessments and predictions are very important to preempt the risk of death due to heart failure. And courtesy of the technological advancements great strides have been made in this regard. Since the turn of the decade(2010s) sales of fitness trackers and smartwatches have been on the rise and it is a $36 billion industry today, and even though most of the trackers just have basic features as the heart rate monitors and step trackers and even though the devices are not acutely accurate but there are multiple apps that can predict the chances of heart failure or other ailments using machine learning using data collected via the fitness trackers.

More sophisticated devices are more precise and these devices coupled with apps that sync data regularly and provide real time analysis can be life saving

The idea behind the project is same as the objective behind the apps and health tracking equipment's, analyzing data associated with heart rate, cholesterol levels and various other attributes can help us make somewhat accurate predictions of future impending risks as well as provide the doctors with the idea of the intensity as well as probability of heart attack occurring and based on that they can take measures to advert or minimize the risks.

The purpose of the project is to implement known machine learning models and assess their reliability in determining or predicting the possibility of heart failure when we use 12 distinct attributes based on the health record of the patients taken from kaggle, which we have used to train all the models and then determine their accuracy on the new data set. This has been done to determine which model which when equipped with more tools and more detailing can be used as a valid and reliable predictor which can be used to provide accurate analysis and make vital predictions and save lives.

Our study not only helped us to understand the way the classification models worked on the data set and the accuracy of predictably of each classification model but also the fact that how different health attributes help us to reach more conclusive results and how each attribute affect the model's accuracy

We used many known classification models and after fine tuning the best accuracy we could do was 87.94% which is much below industry standards and performed on a well processed and hand-picked data set. But with more training and more in depth implementation we will able to conceive a model that can make a much more accurate prediction at an industrially satisfactory level.

Chapter 2
Basic Concepts/ Literature Review

Heart failure prediction using ML models is not a niche field,and tech giants are constantly trying to update their instruments and devices in order to better synchronize the data against better prediction modeling.

As a result there exists abundant research as well as data on the matter,and each tries to study different health related attributes that can be used to make better predictions and as a result except for few primary factors different research data contains vastly different attributes.

From the objective of our project,as we were amateurs in the field and required more precise and accurate data that would be compatible enough with the existing models to yield satisfactory result so we decided to get the data from Kaggle,one of the biggest and most reliable data libraries on the internet that consisted of hundreds of libraries containing data extracted from top notch research on heart failure and had different health attributes which we could use to make accurate predictions(the community support on kaggle also helped in this matter).

We decide to go with the data set:https://www.kaggle.com/datasets/fedesoriano/heart-failure-prediction Which had optimal usability and was updated periodically so as to improve the result.This coupled with the fact that the database was license free(generally all datasets on kaggle are open) enabled us to work directly with the data set without any hassles.

Being engineering students we had no prior knowledge of attributes which can be picked over another in order to make accurate predictions so we went with the general concise of the kaggle community.

The dataset picked by us (credit) was created by the combination of 5 existing datasets in a refined manner(taking 11 common attributes present in all the original datasets)which resulted in the largest heart diseases dataset.

The 5 original dataset which were mixed were from Cleveland, Hungarian, Switzerland, Long Beach VA, Stalog (Heart) Data Set This resulted in 918 observations when filtered for duplicates and other outliers The attributes were:

1. Age
2. Sex
3. Chest pain/type
4. Resting blood pressure
5. Cholesterol
6. Fasting Blood Sugar
7. resting electrocardiogram
8. Maximum Heart rate
9. exercise induced angina
10. Oldpeak
11. Heart disease target

Based on the attributes and data extracted from the above mentioned sources we were able to train different types of ML models and study their accuracy.Our objective was to find the most effective known Ml model which could make highly precise predictions and the quality as well as quantity of the dataset helped us to achieve our objective.

Chapter 3
Problem Statement/ Requirement Specifications

Using dataset, training the classification and artificial neural network model in order to find the best performing model, which can predict the likelihood of a person getting a heart attack. So the medical specialist can intervene and recommend necessary lifestyle changes. While also analyzing the dataset in order to find correlation between dependent (target variable) and independent variable. To find pattern in the data like which age, gender group which are more certain of having a heart attack etc.

3.1 Project Planning

As our problem belong to binary classification (either 1 for having a heart attack or 0 for not having a heart attack). First phase should be data cleaning, here we will check for empty cells, redundant tuples etc. Next will be visualizing the data and seeing the correlation between dependent-independent variable and independent-independent variable. Following which we will one-hot encode categorical data and apply the model. Taking confusion matrix into account we will compare the accuracy. And use the best performing model to predict new data

3.2 Project Analysis

The dataset we have contains only 918 observation which could be detrimental for model training, as the more we had the better the model could have performed. Some of the data is also categorical data. ML model have problem processing categorical data so in order to take care of those we have to either label encode the data or one-hot-encoding. And this is also a deciding factor as depending on how we encode data performance of model can be altered. As we are going to apply artificial neural network it would be best if we apply one hot encoding. Using one-hot encoding we may create more than 20 attribute as all one-hot does is create extra column which may be difficult to handle in future.

3.3 System Design

3.3.1 Design Constraints

The whole project is done in Jupiter notebook

Necessary package

1. NumPy :- use to tackle multidimensional array and there processing and operation
2. Pandas :- pandas are used to analyze, clean and manipulate data, in our project we have used to import dataset
3. Matplotlib :- these are visualizing tools for python
4. Seaborn :- its based on matplotlib, it's use is to visualize high level statistical data
5. Scikit-learn :- library for ML processing
6. TensorFlow :- helps to build Artificial Neural Network

PC requirement

1. Intel(R) Core(TM) i5-8250U CPU @ 1.60GHz 1.80 GHz
2. 16.0 GB Ram
3. 64-bit operating system, x64-based processor

3.3.2 System Architecture **OR** Block Diagram

In Preprocessing we started with importing necessary libraries and our dataset, following which data is visualized to find pattern in the data and correlation between attributes. Before starting the ml part certain necessary step that are done like redundancy check, null cell identification, one-hot-encoding categorical data. After all this data is split into test and training set in 3:1 ratio respectively. 75% data is used to then train classification models.

And the remaining 25% of the data is used for testing based on the accurate prediction, false positive and false negative confusion matrix is prepared. And after repeating the process with several models we will find out the best performing model. We will also be trying to hyper tune the best performing model using grid search cv in order to get the best parameters, using this hyper tune model we will finally try to predict a new result.

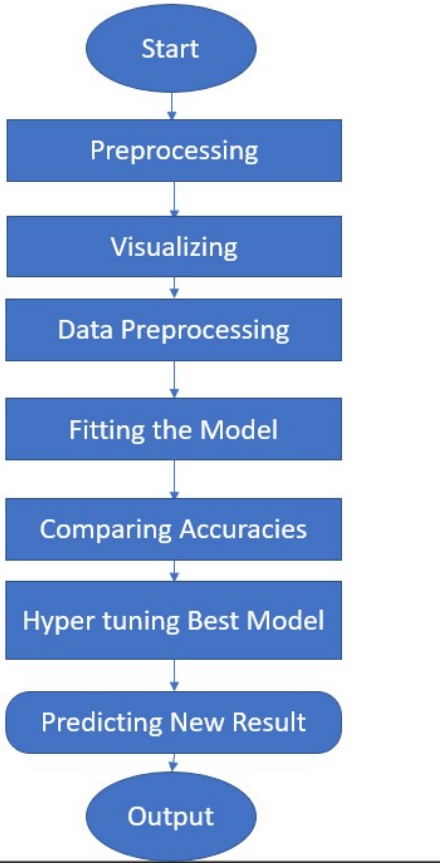

Fig 3.1 : Flow diagram of the Project

Chapter 4
Implementation

Classification basically group certain data into already given classes, as our target variable is a binary output where 0 signifies the person might not have a heart attack and 1 that he/she might. So we will use all classification model we know like

1. Logistic Regression
2. SVM
3. Kernel SVM
4. KNN
5. Decision Tree
6. Random Forrest

Afterward we will try and use Artificial Neural Network (which we can make to predict binary output) and XGBboost for prediction. Next step will be to hypertune best performing model. And finally predict our result

4.1 Methodology OR Proposal

Feature	Description
Age	Age of the Person
Sex	Gender (Male and Female)
Chest Pain	Chest pain type TA :- Typical Angina ATA :- Atypical Angina NAP :- Non-Anginal Pain ASY :- Asymptotic Angina
Resting BP	Normal Blood Pressure [mm/Hg]
Cholesterol	Cholesterol level of Person [mm/dl]
Fasting BS	Blood Sugar before food intake if FastingBS > 120 mg/dl then 1,else 0
Resting ECG	Resting electrocardiogram Normal :- Normal ST :- having ST-T wave abnormality (T wave inversions and/or ST elevation or depression of > 0.05 mV) LVH :- showing probable or definite left ventricular hypertrophy by Estes' criteria]
Max HR	Maximum Heart Rate (60-202)
Exercise Angina	Angina induced by exercising
OldPeak	ST numeric value measured in depression

ST_Slope	Slope of peak exercise ST Up :- Upsloping Flat :- flat Down :- Down slopping
Heart Disease	Output 1 - heart failure 0 - normal

Some of our attributes are categorical like sex, chest pain, fasting bs etc. to deal with them we have either label encoding or One-Hot Encoding. We choose the later, the need for dealing with categorical attributes is necessary as Machine learning algorithm can't work on them. Once this is done we have to split the data for training and test purpose each. The ratio of split is 3:1, in order to keep the split consistent throughout each iteration of code run we have assigned a random state 0 to make our study of accuracies a little easier. Moving on the next most important task is to scale all the attributes so that one attribute doesn't become deciding factor and the distance between them. There is a chance of data leakage through test set so we use different function to train our training and test set

4.1.1 Logistic Regression

Logistic Regression is a type of regression used when the target is categorical. It is one of the most famous model. When we try to fit linear regression model there are chances that it may cross the
Where the actual values lies. And completely fails to fit all the point. Where logistic regression uses a sigmoid graph. Which is easy to accommodate the points. And give a better output then linear regression.

$$p = \frac{1}{1 + e^{-z}}$$

$$z = \beta 0 + \beta 1 * x$$

$$\gamma(threshold) = 0.5$$

fig 4.1 : logisitc regression equation

Above this threshold it would be considered 1 where as below it 0
P=0.3 output=0 and p=0.7 output=1

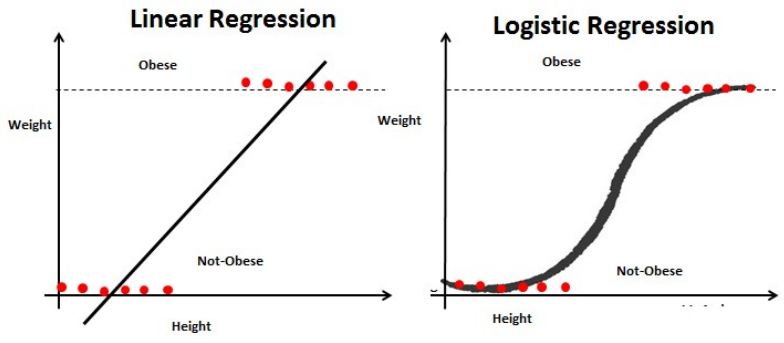

fig 4.2 Linear V Logistic

4.1.2 Naïve Bayes

This one is pure classification model, the idea behind this all the independent variables are independent of each other, for this assumption only its called naïve. This model is based on Bayes Theorem

$$P(A|B) = \frac{P(A) * P(B|A)}{P(B)}$$

Fig 4.3 : Bayes Theorem

Where A is hypothesis whose probability is affected by the data (output). A is the evidence i.e. the unseen data which was not used in computing the prior probability. P(A) is the prior probability, P(A|B) is the posterior probability, P(B|A) is probability of observing B given A.

We have two classes (0 and 1)

$$P(y|x) = \frac{P(y) * P(x|y)}{P(x)}$$

$$x = (x1, x2, x3, x4)$$

$$P(y|x1, x2, x3, x4) = \frac{P(A) * P(y|x1) * P(y|x2) \dots P(y|x4}{P(x1) * P(x2) \dots P(x4)}$$

$$P(y|x1, x2, x3, x4) = \frac{P(y) * P(y|x1) * P(y|x2) \dots P(y|x4)}{P(x1) * P(x2) \dots P(x4)}$$

Fig 4.4 : Bayes thereom with multiple variable

There are 2 classes of y either 0 or 1 we compare them and class with greater probability is the output.

$$P(0|x1, x2, x3, x4) \text{ and } P(1|x1, x2, x3, x4)$$

$$\frac{P(0) * P(0|x1) * P(0|x2) \dots P(0|x4)}{P(x1) * P(x2) \dots P(x4)} \text{ and } \frac{P(1) * P(1|x1) * P(1|x2) \dots P(1|x4)}{P(x1) * P(x2) \dots P(x4)}$$

Fig 4.5 class prob calculation

4.1.3 Support Vector Machine

It plots all the feature on a graph and classify them into two classes(0 or 1) on the basis of a hyperplane

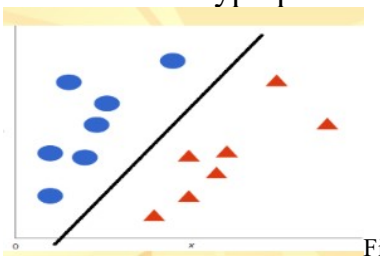

Fig 4.6 Hyperlane dividing two plane

Hyperplane is basically used for dividing in 1D it's a point, 2d a line, 3d a plane and above it a hyperplane. It is such that it maximizes margin between two class. Its drawn by joining two closest point of the two class by a line and perpendicularly bisecting it. It is such that the distance to these point are largest (it helps in robustness of the model lack of which may lead to misclassification.

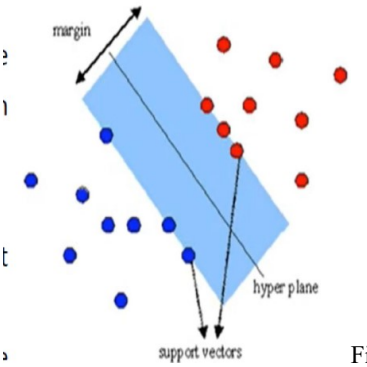

Fig 4.7 how to draw hyperplane

The name of the model is support vector as points here are called vectors, and these 2 point using which hyperplane is drawn are like a support to it hence.

4.1.4 Kernal SVM

Sometimes a situation may arise where the two class are such that they can't be divided by any iteration of SVM in that case we introduce another dimension.

$$Z^2=x^2+y^2$$

The point which are neared to center remains a level below while farther away point get above. This is quite a difficult task to achieve but in python this can be done using kernel

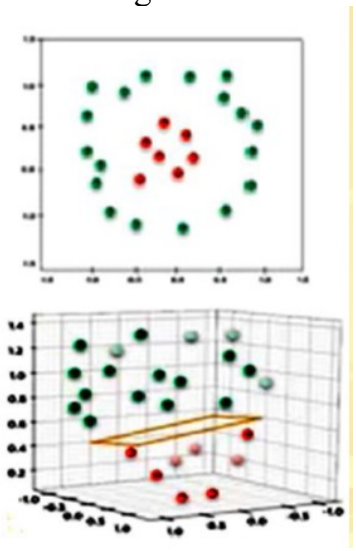

Fig 4.8 kernel SVM

SVM and Kernel SVM both the model a little different in approach from all the other model as

- All the other model tries to compare a fruit with the most apples' apple and orangish orange
- While SVM compare a fruit with the most orangish apple and apples' orange

Sometime this approach backfires but sometimes it becomes the best performing model.

4.1.5 K-nearest Neighbors

In this algorithm, a new data is classified based on it's distance from the two binary classes, in this model we have used the distance as Euclidean. It follow the procedure

1. Choose the k no of neighbors
2. Using Euclidean distance calculate k no nearest points
3. Count how many of these belong to each binary category
4. Assign the new data to the category having more no. of nearest neighbor

Euclidean Distance :-

$$\sqrt{(x_1^2 - x_2^2) * (y_1^2 - y_2^2)}$$

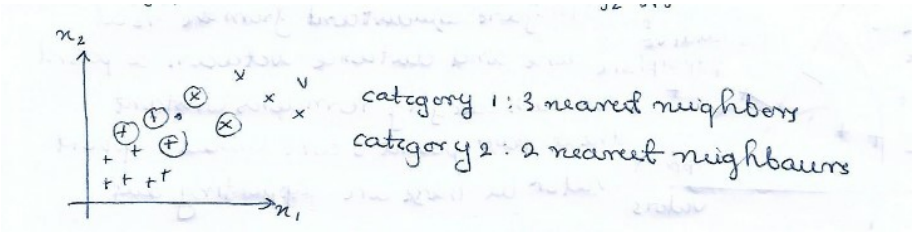

Fig 4.9 K-Nearest Neighbour

4.1.6 Decision Tree

Decision tree works on splitting the data in such way, that a new data can be easily classified by traversing a decision tree. Data points are splitted such that it can maximize the categories.

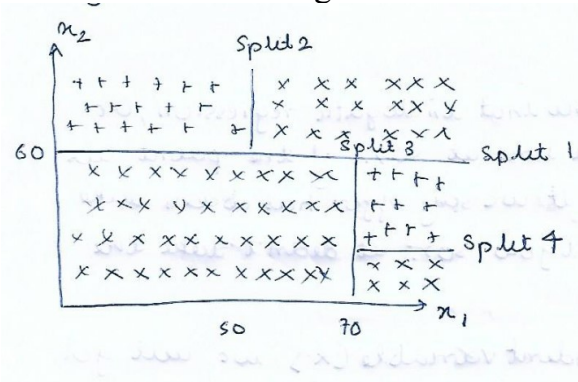

Fig 4.10 splitting done in decision tree

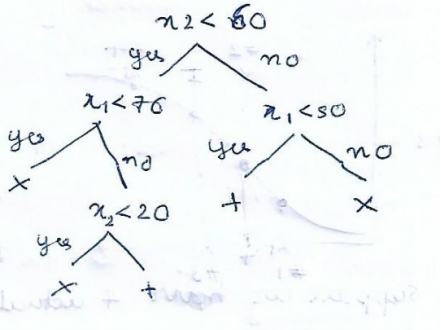

fig 4.11 decision tree

4.1.7 Random Forrest Classification

It comes under Ensemble Learning(using multiple algorithm and putting them together as one big ML model). It works on a principle that multiple guessing may not reveal a real answer but taking there average we can reduce the gap between prediction and actual values

1. Choose random k data point from training set
2. Build a decision tree for them
3. Select the no of random tree you want to build and repeat step 1 and 2
4. Use every one of these decision tree and predict category of a new data, and assign it to the category having the majority vote.

4.1.8 Artificial Neural Network

Neural Network was created with the aim of replicating actual neurons. Which relay the electric signal to brain. Neural network is that there are some inputs which are given to the neuron through weighted synapsis, and neuron gives an output. The output can be continuous or categorical

Steps performed in neurons

1. $A = \sum_{i=1}^{m} w_i x_i$
2. $\varphi(A)$ signal created
3. Signal passed to output

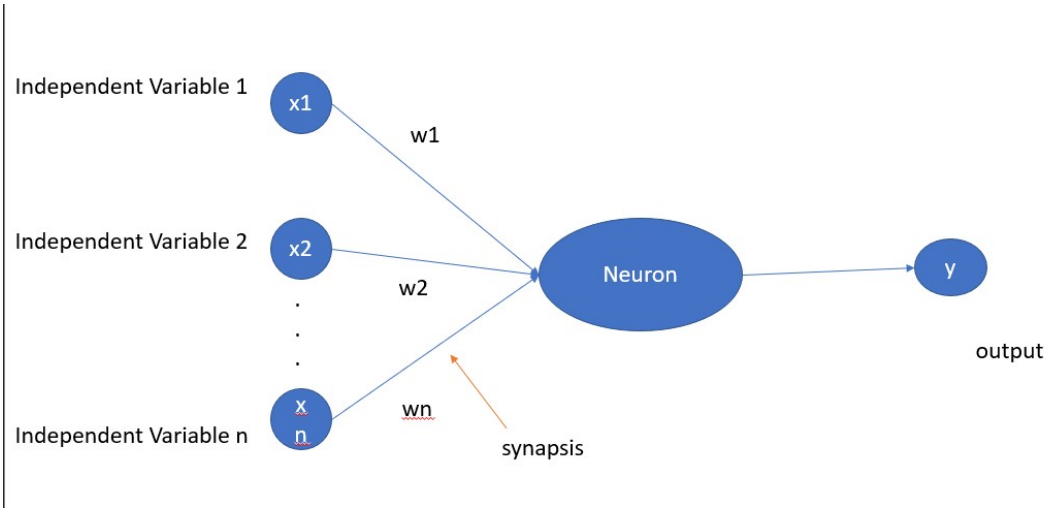

Fig 4.12 : structure of neural network

Signals are called activation function these decide whether neuron will remain on or not. There are four type of activation function

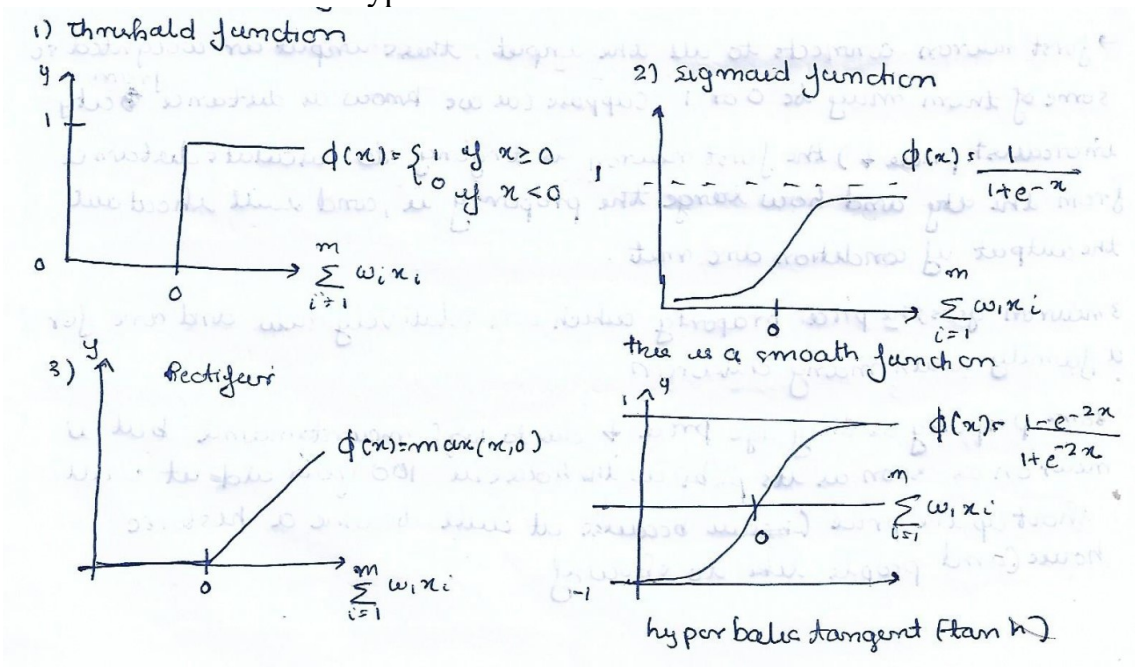

Fig 4.13: activation function

The one advantage neural network have in comparison to other machine learning model is hidden layer.
It tries various combination of independent attributes in order to creates its hidden layer, Neural network also offer back propagation suppose during training part y output hi calculated then compared with the y actual and the difference is back propagated to the neurons so that they can adjust the weight of all the input variable so that predicted value of y can be closer to actual value of y. The steps followed are

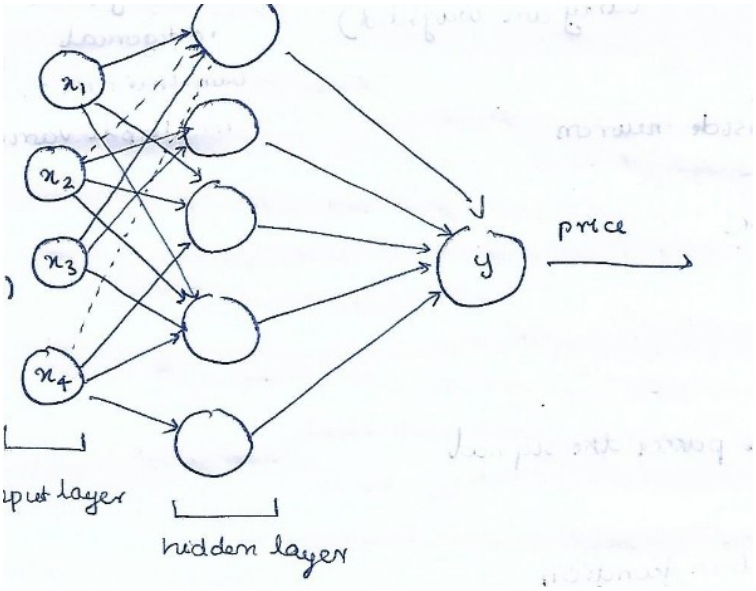

Fig 4.14 : example of a neural network

1. Initialize weight randomly to no. closer to 0
2. Input data
3. Neurons are activated in a way that the impact of each neuron's activation is limited by weights. Continue the activation until y pred is calculated.
4. Generate error by comparing y pred with y actual
5. Back propagate error and adjust weight accordingly
6. Repeat 1-5 with new data and continue to update weight(reinforcement learning) or update after a batch of data(batch learning
7. When done with the whole training set this cycle is repeated. We call this cycle epoch

4.1.9 XGBoost

These are gradient boosted decision trees designed for speed and performance.

4.1.10 Grid Search CV

It helps to find the best parameter to train the model on for highest accuracy the model can give. It loops through all the possible combination of parameters

4.2 Testing OR Verification Plan

After the model is tested on test set we have four condition

1. Actual value of y=0 and predicted value of y=0 (ture negative)
2. Actual value of y=0 and predicted value of y=1 (false positive)
3. Actual value of y=1 and predicted value of y=0 (false negative)
4. Actual value of y=1 and predicted valued of y=1 (true positive)

False negative is much more dangerous of an error in comparison to false positive as, a person not having a heart attack when predicted we would, doesn't bother as much as a person having a heart attack but predicted not to.

Based on these four outcomes we create a confusion matrix in which y actual is assigned row value and y pred columns value and we represent using A_{00} as true negative, A_{01} as false positive, A_{10} is false negative, A_{11} is true positive

n=165	**Predicted: NO**	**Predicted: YES**	
Actual: NO	TN = 50	FP = 10	60
Actual: YES	FN = 5	TP = 100	105
	55	110	

Fig 4.15 : Confusion Matrix

Using this we can calculate the accuracy of our model as
Correct prediction = TN+TP = 50+100 = 150
Total Value = n = TN+FP+FN+TP = 165

$$Accuracy = \frac{Correct\ Prediction}{Total\ Value} = \frac{150}{165} = 90.90\ \%$$

4.3 Result Analysis OR Screenshots

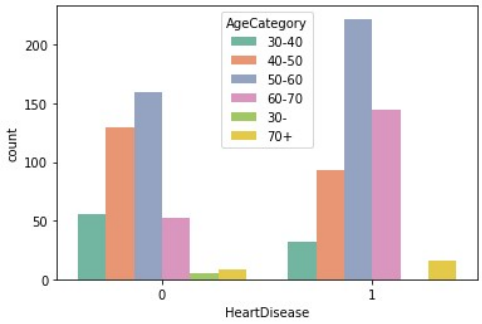

Fig 4.16 Age variation

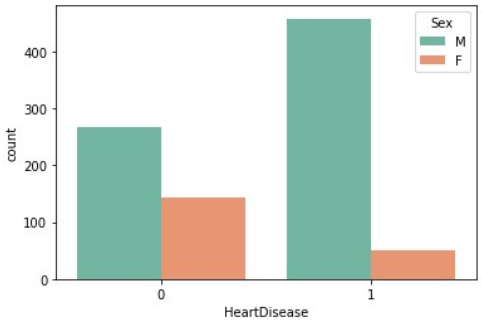

Fig 4.17 Sex variation

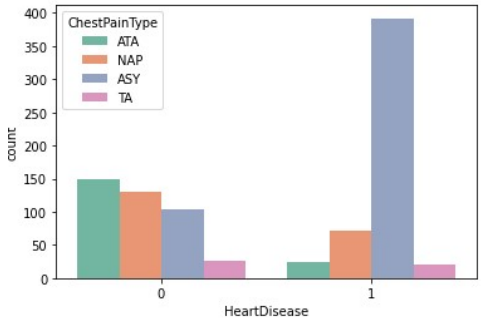

Fig 4.18 Chest pain variation

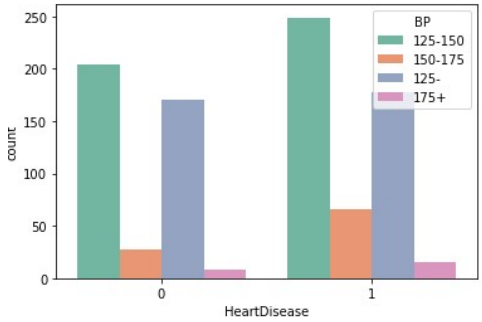

Fig 4.19 Blood Pressure variation

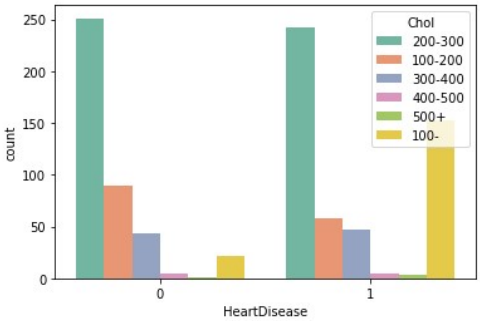

Fig 4.20 cholestrol variation

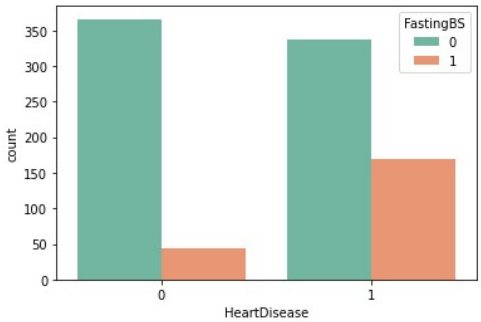

Fig 4.21 FastingBS variation

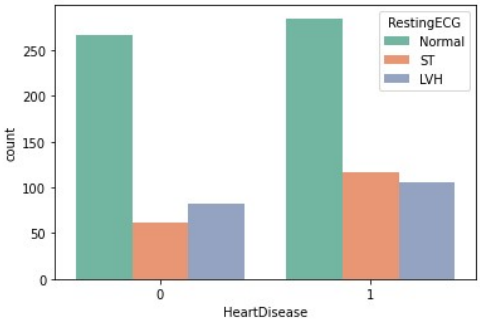

Fig 4.22 RestingECG variation

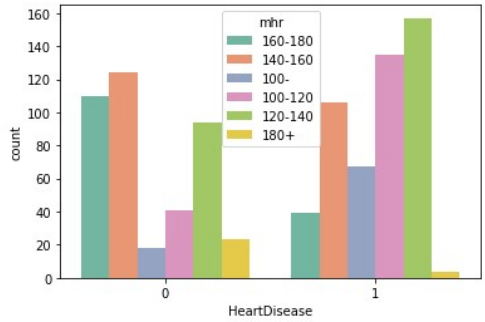

Fig 4.23 Maximum Heart Rate variation

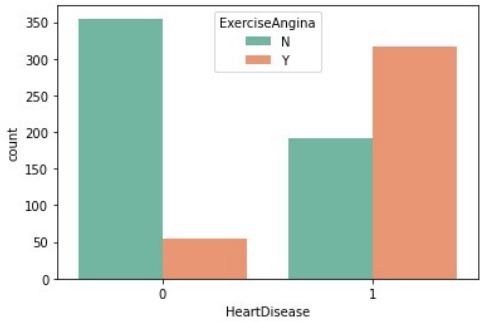

Fig 4.24 ExerciseAngina Variation

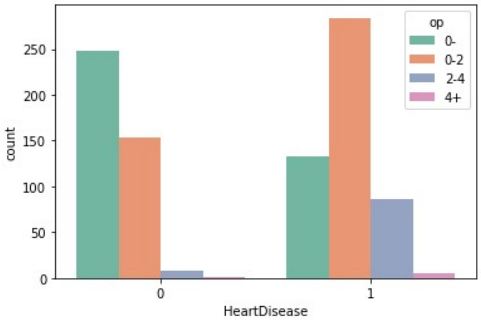

Fig 4.25 Oldpeak Variation

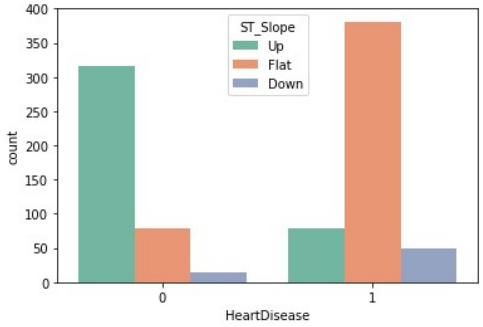

Fig 4.26 ST Slope Variation

	Age	RestingBP	Cholesterol	FastingBS	MaxHR	Oldpeak	HeartDisease
Age	1.000000	0.254399	-0.095282	0.198039	-0.382045	0.258612	0.282039
RestingBP	0.254399	1.000000	0.100893	0.070193	-0.112135	0.164803	0.107589
Cholesterol	-0.095282	0.100893	1.000000	-0.260974	0.235792	0.050148	-0.232741
FastingBS	0.198039	0.070193	-0.260974	1.000000	-0.131438	0.052698	0.267291
MaxHR	-0.382045	-0.112135	0.235792	-0.131438	1.000000	-0.160691	-0.400421
Oldpeak	0.258612	0.164803	0.050148	0.052698	-0.160691	1.000000	0.403951
HeartDisease	0.282039	0.107589	-0.232741	0.267291	-0.400421	0.403951	1.000000

Fig 4.27 Correlation Between feature

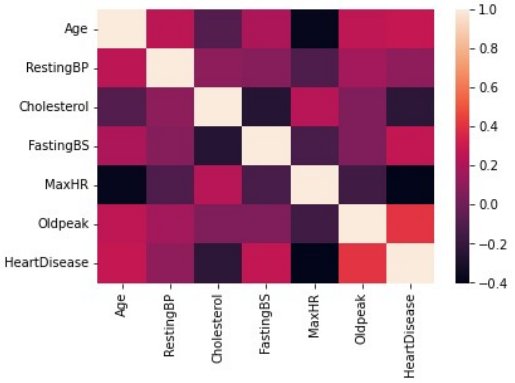

Fig 4.28 Heatmap of Correlation

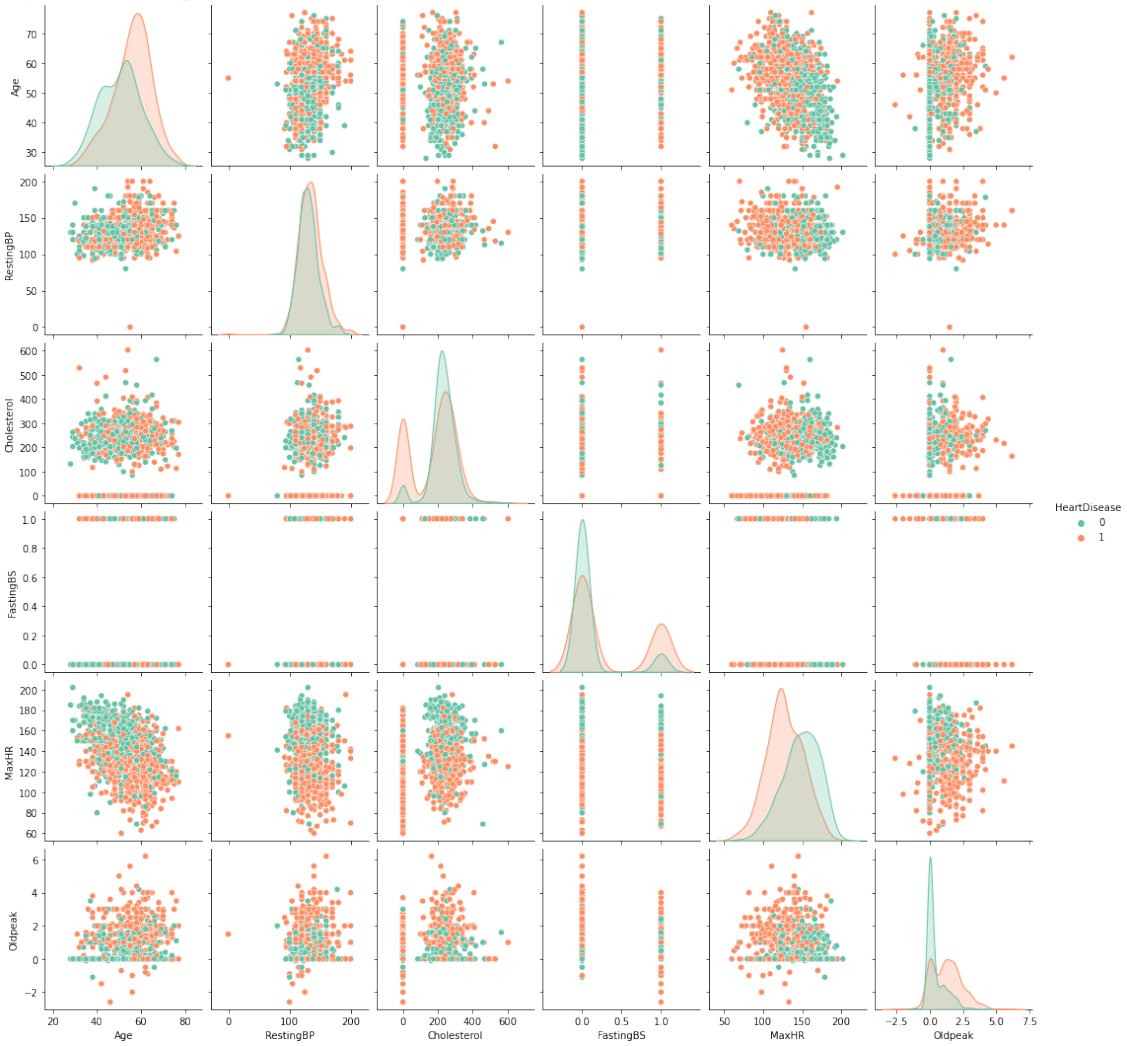

Fig 4.29 Bivariate Comparision

Logistic Regression

```
In [44]:
from sklearn.linear_model import LogisticRegression
LR = LogisticRegression(random_state = 0)
LR.fit(x_train, y_train)

Out[44]: LogisticRegression(random_state=0)

In [45]:
from sklearn.metrics import confusion_matrix, accuracy_score
y_pred = LR.predict(x_test)
cm = confusion_matrix(y_test, y_pred)
sns.heatmap(cm,annot=True,fmt=".1f")
t=accuracy_score(y_test, y_pred)*100
print('Accuracy=',t)
perf.append(['Logistic Regression',t])

Accuracy= 82.6086956521739
```

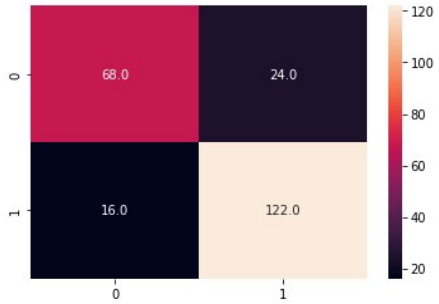

Fig 4.30 Logistic Regression

Naive Bayes

```
In [47]:
from sklearn.naive_bayes import GaussianNB
NB = GaussianNB()
NB.fit(x_train, y_train)

Out[47]: GaussianNB()

In [48]:
from sklearn.metrics import confusion_matrix, accuracy_score
y_pred = NB.predict(x_test)
cm = confusion_matrix(y_test, y_pred)
sns.heatmap(cm,annot=True,fmt=".1f")
t=accuracy_score(y_test, y_pred)*100
print('Accuracy=',t)
perf.append(['Naive Bayes',t])

Accuracy= 84.34782608695653
```

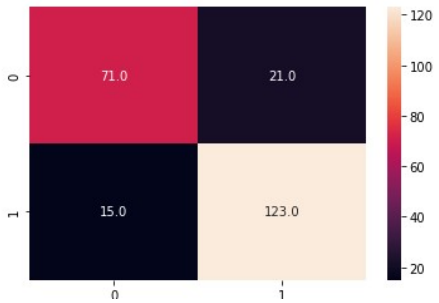

Fig 4.31 Naïve Bayes

SVM

```
In [50]:
from sklearn.svm import SVC
SV = SVC(kernel = 'linear', random_state = 0)
SV.fit(x_train, y_train)

Out[50]: SVC(kernel='linear', random_state=0)

In [51]:
from sklearn.metrics import confusion_matrix, accuracy_score
y_pred = SV.predict(x_test)
cm = confusion_matrix(y_test, y_pred)
sns.heatmap(cm,annot=True,fmt=".1f")
t=accuracy_score(y_test, y_pred)*100
print('Accuracy=',t)
perf.append(['SVM',t])

Accuracy= 81.73913043478261
```

Fig 4.32 SVM

Kernal SVM

```
In [53]:
from sklearn.svm import SVC
KSV = SVC(kernel = 'rbf', random_state = 0)
KSV.fit(x_train, y_train)

Out[53]: SVC(random_state=0)

In [54]:
from sklearn.metrics import confusion_matrix, accuracy_score
y_pred = KSV.predict(x_test)
cm = confusion_matrix(y_test, y_pred)
sns.heatmap(cm,annot=True,fmt=".1f")
t=accuracy_score(y_test, y_pred)*100
print('Accuracy=',t)
perf.append(['Kernal SVM',t])

Accuracy= 85.21739130434783
```

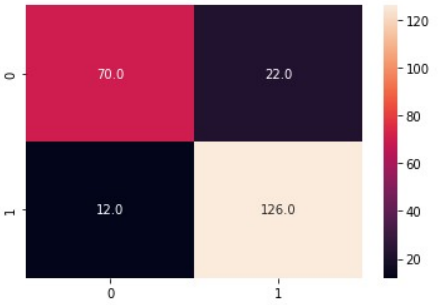

Fig 4.33 Kernal SVM

KNN

```
In [56]:
from sklearn.neighbors import KNeighborsClassifier
```

```
In [57]:
from sklearn.neighbors import KNeighborsClassifier
KNN = KNeighborsClassifier(n_neighbors = 5, metric = 'minkowski', p = 2)
KNN.fit(x_train, y_train)
```

```
Out[57]: KNeighborsClassifier()
```

```
In [58]:
from sklearn.metrics import confusion_matrix, accuracy_score
y_pred = KNN.predict(x_test)
cm = confusion_matrix(y_test, y_pred)
sns.heatmap(cm,annot=True,fmt=".1f")
t=accuracy_score(y_test, y_pred)*100
print('Accuracy=',t)
perf.append(['KNN',t])
```

```
Accuracy= 83.91304347826087
```

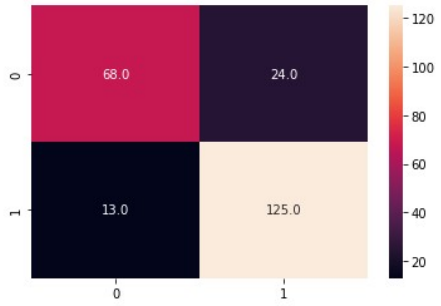

Fig 4.34 KNN

Descision Tree

```
In [60]:
from sklearn.tree import DecisionTreeClassifier
DT = DecisionTreeClassifier(criterion = 'entropy', random_state = 0)
DT.fit(x_train, y_train)
```

```
Out[60]: DecisionTreeClassifier(criterion='entropy', random_state=0)
```

```
In [61]:
from sklearn.metrics import confusion_matrix, accuracy_score
y_pred = DT.predict(x_test)
cm = confusion_matrix(y_test, y_pred)
sns.heatmap(cm,annot=True,fmt=".1f")
t=accuracy_score(y_test, y_pred)*100
print('Accuracy=',t)
perf.append(['Descision Tree',t])
```

```
Accuracy= 83.91304347826087
```

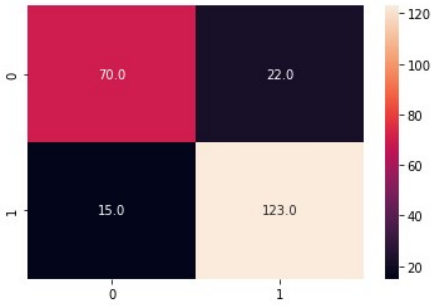

Fig 4.35 Descision Tree

Random Forrest Classification

```
In [63]:
1 from sklearn.ensemble import RandomForestClassifier
2 RF = RandomForestClassifier(n_estimators = 10, criterion = 'entropy', random_state = 0)
3 RF.fit(x_train, y_train)

Out[63]: RandomForestClassifier(criterion='entropy', n_estimators=10, random_state=0)
```

```
In [64]:
1 from sklearn.metrics import confusion_matrix, accuracy_score
2 y_pred = RF.predict(x_test)
3 cm = confusion_matrix(y_test, y_pred)
4 sns.heatmap(cm,annot=True,fmt=".1f")
5 t=accuracy_score(y_test, y_pred)*100
6 print('Accuracy=',t)
7 perf.append(['Random Forrest Classification',t])

Accuracy= 87.39130434782608
```

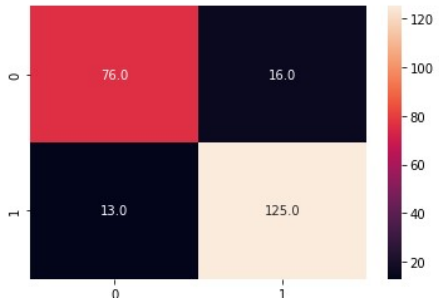

Fig 4.36 Random Forrest Classification

Artificial Neural Network

```
In [77]:
1 from sklearn.metrics import confusion_matrix, accuracy_score
2 cm = confusion_matrix(y_test, y_pred)
3 sns.heatmap(cm,annot=True,fmt=".1f")
4 t=accuracy_score(y_test, y_pred)*100
5 print(t)
6 perf.append(['Artificial Neural Network',t])

84.78260869565217
```

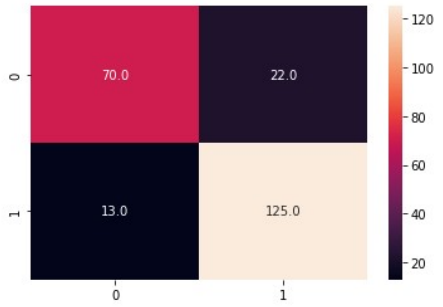

Fig 4.37 ANN

XGB Boost

In [79]:
```
from xgboost import XGBClassifier
classifier = XGBClassifier()
classifier.fit(x_train, y_train)
```

```
C:\Users\19056\anaconda3\lib\site-packages\xgboost\sklearn.py:1224: UserWarning: The use of label encoder in XGBClassifier is d
eprecated and will be removed in a future release. To remove this warning, do the following: 1) Pass option use_label_encoder=F
alse when constructing XGBClassifier object; and 2) Encode your labels (y) as integers starting with 0, i.e. 0, 1, 2, ..., [num
_class - 1].
  warnings.warn(label_encoder_deprecation_msg, UserWarning)

[10:09:56] WARNING: C:/Users/Administrator/workspace/xgboost-win64_release_1.5.1/src/learner.cc:1115: Starting in XGBoost 1.3.
0, the default evaluation metric used with the objective 'binary:logistic' was changed from 'error' to 'logloss'. Explicitly se
t eval_metric if you'd like to restore the old behavior.
```

Out[79]:
```
XGBClassifier(base_score=0.5, booster='gbtree', colsample_bylevel=1,
              colsample_bynode=1, colsample_bytree=1, enable_categorical=False,
              gamma=0, gpu_id=-1, importance_type=None,
              interaction_constraints='', learning_rate=0.300000012,
              max_delta_step=0, max_depth=6, min_child_weight=1, missing=nan,
              monotone_constraints='()', n_estimators=100, n_jobs=8,
              num_parallel_tree=1, predictor='auto', random_state=0,
              reg_alpha=0, reg_lambda=1, scale_pos_weight=1, subsample=1,
              tree_method='exact', validate_parameters=1, verbosity=None)
```

In [80]:
```
from sklearn.metrics import confusion_matrix, accuracy_score
y_pred = classifier.predict(x_test)
cm = confusion_matrix(y_test, y_pred)
sns.heatmap(cm,annot=True,fmt=".1f")
t=accuracy_score(y_test, y_pred)*100
print(t)
perf.append(['XGB boost',t])
```

```
83.91304347826087
```

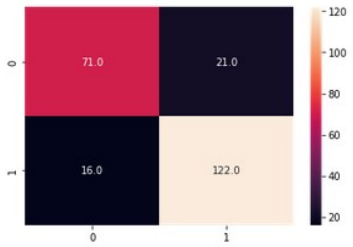

Fig 4.38 XGBoost

Out[81]:

	Names	Acc_Score
6	Random Forrest Classification	87.391304
3	Kernal SVM	85.217391
7	Artificial Neural Network	84.782609
1	Naive Bayes	84.347826
4	KNN	83.913043
5	Descision Tree	83.913043
8	XGB boost	83.913043
0	Logistic Regression	82.608696
2	SVM	81.739130

Fig 4.39 Accuracies

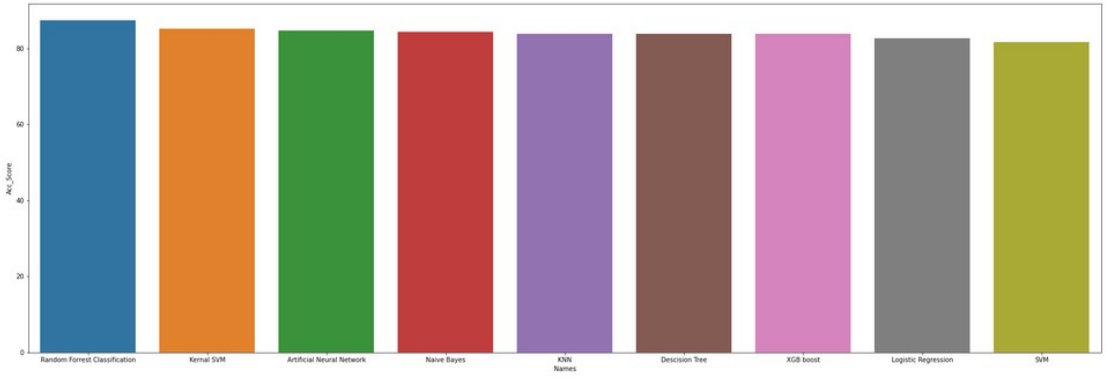

Fig 4.40 Accuracies graph

It shuffles the dataset randomly and then split the data into k groups and on each k group perform the basic steps of ml like splitting data into training and test set. Training the model testing the model and calculating k no of accuracies for each model also calculating standard deviation

	Names	Accuracy	Standard Deviation
0	Logistic Regression	87.350810	3.015892
2	SVM	87.208014	2.835074
3	Kernal SVM	86.918159	4.237970
1	Naive Bayes	86.624041	4.433612
4	KNN	86.187127	3.789120
6	Random Forrest	85.462489	4.958828
5	Descision Tree	78.184143	3.774295

Fig 4.41 K-FOLD Accuracies graph

Applying Grid Search CV on Random Forrest

In [82]:

```
from sklearn.model_selection import GridSearchCV
parameters = [{
    'n_estimators': [10,20,30,40],
    'max_features': ['auto', 'sqrt', 'log2'],
    'max_depth' : [4,5,6,7,8],
    'criterion' :['gini', 'entropy']
}]
Rgs = RandomForestClassifier(random_state = 0)
grid_search = GridSearchCV(estimator = Rgs,
                           param_grid = parameters,
                           scoring = 'accuracy',
                           cv = 10,
                           n_jobs = -1)
grid_search.fit(x_train, y_train)
best_accuracy = grid_search.best_score_
best_parameters = grid_search.best_params_
print("Best Accuracy: {:.2f} %".format(best_accuracy*100))
print("Best Parameters:", best_parameters)
```

```
Best Accuracy: 87.94 %
Best Parameters: {'criterion': 'entropy', 'max_depth': 6, 'max_features': 'auto', 'n_estimators': 40}
```

Fig 4.42 GridSearchCV

Chapter 5

Conclusion and Future Scope

5.1 Conclusion

In our study of the dataset, we found out that male with age above 50 and having asymptomatic chest pain with blood pressure below 175 and cholesterol less than 500 and fasting blood sugar more than 120 are more likely to have a heart attack than the rest.

Using machine learning we also found out that Random Forest is best performing model based on the current dataset with accuracy score of 87.39%. And after hyper tunning its performance increase increases slightly to 87.94%.

5.2 Future Scope

One of the major disadvantages we had was lack of significant no of data. As the accuracy of a model increases the more no and variety of data it is exposed to. That's why it is better to say that random Forrest is the best performing model according to our dataset than random Forrest is the best performing model, as the we had a significant no of observation the result would've been different. This project also lacks skilled ml engineer as we have a basic level of knowledge. The best accuracy we could do 87.94% with hyper tunning.

References

[1] https://www.cdc.gov/heartdisease/facts.htm
[2] https://amj.amegroups.com/article/view/5475/html
[3] https://www.fortunebusinessinsights.com/fitness-tracker-market-103358
[4] https://indianexpress.com/article/technology/gadgets/apple-watch-saves-an-indian-life-thanks-to-its-ecg-feature-report-6808698/
[5] https://www.kaggle.com/datasets/andrewmvd/heart-failure-clinical-data
[6] https://support.apple.com/en-in/HT208931
[7] https://www.kaggle.com/fedesoriano/heart-failure-prediction
[8] https://jupyter.org
[9] https://numpy.org
[10] https://pandas.pydata.org
[11] https://matplotlib.org
[12] https://seaborn.pydata.org
[13] https://scikit-learn.org/stable/
[14] https://www.tensorflow.org
[15] https://www.analyticsvidhya.com/blog/2020/12/beginners-take-how-logistic-regression-is-related-to-linear-regression/
[16] https://www.dataschool.io/simple-guide-to-confusion-matrix-terminology/

Heart Failure Prediction

Aryan Sarraf
1905599

Abstract: Due to the sudden shift in lifestyle, Heart attack is a major cause of death. And patient dying from heart attack is increasing day by day. A sudden need to predict likelihood of a person dying from heart attack is the need of the hour, in medical field. ML has given an edge be it not 100% accurate it is better then nothing. If prediction is done then maybe intervention in the lifestyle can be done by the medical professional in order to save the life of person.

Individual contribution and findings: The Task given to me was analyzing the graph. Keeping that in mind the observation we have are 918. So visualizing them would easy and surely reveal some pattern which we missed in correlation and group the individual most prone to the heart failure. After seeing dataset, it was clear there were already some type of data between the attributes such as categorical data like sex, chestpaintype. Continuous like age and cholesterol. First, I did univariable comparison using count plot for categorical data and distplot for continuous values. Next task on my hand was to variation of heart disease in accordance to another variable. As said, I had two types of data, it was easy to draw the plot of categorical only using count plot again just this time using output as counting condition. But for the continuous value the only comparison seem feasible to me was to categorize them. So I did the same and using constant interval categorized them. Finally again using count plot showed them in variation with target variable. It was clear that some were showing positive correlation while for others there were none, no pattern at all. Using this graph data I prediction the category. For model implementation I had the task to implement SVM and KSVM. These are different then other algorithm, as other algorithm compare with the most well fitted data to test these check their data on outlier. In some scenario these are special algorithm performing better than other. But in accordance to how I got the data encoded they are just doing fine. KSVM is just one step further taken by SVM. To categorize data by implementing another dimension to the mix

Individual contribution to project report preparation: I did the project planning and implementation methodology and obviously wrote my algorithm part. I also did the screen shot of implementation and also assisted in writing conclusion and future scope.

Individual contribution for project presentation and demonstration: I did the same part on ppt also like organizing all the screenshot etc

Full Signature of Supervisor:
…….

Full signature of the student:
……..

INDIVIDUAL CONTRIBUTION REPORT:

Heart Failure Prediction

Ashutosh Mishra
1905600

Abstract: Due to the sudden shift in lifestyle, Heart attack is a major cause of death. And patient dying from heart attack is increasing day by day. A sudden need to predict likelihood of a person dying from heart attack is the need of the hour, in medical field. ML has given an edge be it not 100% accurate it is better then nothing. If prediction is done then maybe intervention in the lifestyle can be done by the medical professional in order to save the life of person.

Individual contribution and findings: The first part done by me was preprocessing involving splitting data into test and training set, as my data set was of only 918. My first challenge was to select a significant no. training data so can my model train of variety of data but doesn't get overfitted. The split of 3:1 was more reasonable to me. Next was doing something about the categorical data, I did choose One-Hot-Encoding as this will make column into binary switches which should help my dataset by giving more and easily processable by model. And also, model performed better in comparison label encoded data. Feature scaling was required by some model so applied it. For model I applied XGBoost, ANN as these model work both on classification and regression I have used them to see if I can take advantage of them over another algorithm, but the score didn't seem impressive. Maybe lack of data is the key. I also tested K-FOLD cross validation on algorithm to check how they performed and result is quite different then accuracies extracted from confusion matrix. Due to lack of knowledge, I used random forest to hyper tune as it was the best performing model in confusion matrix accuracies. After hyper tuning it still is the best performing model. Next was the task to do single prediction here our lack of knowledge showed the most as we just given a random input and extracted the output as 0 and 1. Finally I also showed accuracies in form of barplot

Individual contribution to project report preparation: For Report Writing, I have written abstract, formulated the problem description, project analysis my part of project XGBoost, ann and Testing and Verification Part, Hyper tunning/Grid Search CV

Individual contribution for project presentation and demonstration: I have done all the above part in the ppt also, and also compile the same.

Full Signature of Supervisor: ……

Full signature of the student: ……..

Heart Failure Prediction

Pratyush Aanand
1905189

Abstract: Due to the sudden shift in lifestyle, Heart attack is a major cause of death. And patient dying from heart attack is increasing day by day. A sudden need to predict likelihood of a person dying from heart attack is the need of the hour, in medical field. ML has given an edge be it not 100% accurate it is better then nothing. If prediction is done then maybe intervention in the lifestyle can be done by the medical professional in order to save the life of person.

Individual contribution and findings: I have done the preprocessing of project before the actual goal of is achieved. First task was to import all the libraries necessary for our project like numpy, pandas, seaborn & matplotlib. My next task was to import the data in the data frame and check for inconsistency and what is the status of the data, are there any redundant variable, empty cells and outliers. Moving on the model I Implement Logistic Regression and Naïve Bayes, these are most famous algorithm. My approach was simple first simply import the model from sklearn and fitted them according to binary classification, for naïve bayes it was gaussian. Afterward I trained them on training set and then using the test set drew the confusion matrix and calculated the accuracies

Individual contribution to project report preparation: I did list of figure and gathered all the required image. Analyzed Design constraints. Written the same info about model and their accuracies

Individual contribution for project presentation and demonstration: Did the same gathered all the required image and edited the ppt.

Full Signature of Supervisor: ……

Full signature of the student: ……..

Heart Failure Prediction

Sambhav Choudhary
1905634

Abstract: Due to the sudden shift in lifestyle, Heart attack is a major cause of death. And patient dying from heart attack is increasing day by day. A sudden need to predict likelihood of a person dying from heart attack is the need of the hour, in medical field. ML has given an edge be it not 100% accurate it is better then nothing. If prediction is done then maybe intervention in the lifestyle can be done by the medical professional in order to save the life of person.

Individual contribution and findings: My Task started with finding the correlation between the target and other attributes and also the correlation between all the independent attributes. Seeing that there wasn't much of a correlation in data, we can conclude that some of the attributes are the same (affect the target similarly). To show my findings I drew a Heatmap in which color changes according to the correlation value. Also, I drew the bivariate comparison graph. Coming to the model side of things, I implemented KNN. It uses distance to some nearest point to classify the object, the standard practice of implementing it is to consider 5 neighbors. To measure and understand the distance I chose the Euclidean distance (minkowski,p=2). Next, I decided to implement all the tree related classification models. Starting with decision Tree,I chose accuracy over speed and hence didn't not specify a certain depth upto which the operation needed to be carried,The decision tree true to its value worked very well in train_test_split data set but encountered problem in the k-fold test displaying the problem of overfitting and staying true to its low bias high variance virtue,To tackle the shortcomings of the decision tree we switched to the Random Forest classification using the entropy criterion and having 10 elements(10 separate decision tree would have to be constructed),Random Forest works on the principle of bagging(bootstrap aggregation) where separate decision trees are constructed and worked upon the final prediction are made on the basis of majority vote.Random Forest gave the highest accuracy in our experiment.Then I used the extreme gradient boosting algorithm(xgboost) which also makes use of the decision tree but the entire operation is carried out by first making a random prediction then calculating the residues between the actual result and prediction,these residuals are clubbed with a single attribute to construct a decision tree which keeps on adding trees sequentially,it doesn't require hyper tuning because we have a built in pruning point thus making it a lot quicker as well as efficient.

Individual contribution to project report preparation: I wrote the project introduction, literature review draw block diagram of System architecture. Also my part of the implementation. Assisted in conclusion and future scope. Did the Turnitin test

Individual contribution for project presentation and demonstration: I did the my written code part and helped in editing it

Full Signature of Supervisor: ……

Full signature of the student: ……..

Sambhav Choudhary

ORIGINALITY REPORT

14%	9%	6%	9%
SIMILARITY INDEX	INTERNET SOURCES	PUBLICATIONS	STUDENT PAPERS

PRIMARY SOURCES

1	www.coursehero.com Internet Source	2%
2	www.researchgate.net Internet Source	1%
3	Submitted to University of Western Sydney Student Paper	1%
4	Submitted to Melbourne Institute of Technology Student Paper	1%
5	Shwet Ketu, Pramod Kumar Mishra. "Empirical Analysis of Machine Learning Algorithms on Imbalance Electrocardiogram Based Arrhythmia Dataset for Heart Disease Detection", Arabian Journal for Science and Engineering, 2021 Publication	1%
6	Submitted to Federal University of Technology Student Paper	1%
7	Submitted to University of Sunderland Student Paper	1%

8	"An Introduction to Bivariate Regression", Statistics in Criminal Justice, 2006 Publication	1 %
9	me.me Internet Source	<1 %
10	Submitted to The University of Manchester Student Paper	<1 %
11	Submitted to University of Westminster Student Paper	<1 %
12	Ashutosh Shankhdhar, Pawan Kumar Verma, Prateek Agrawal, Vishu Madaan, Charu Gupta. "Quality analysis for reliable complex multiclass neuroscience signal classification via electroencephalography", International Journal of Quality & Reliability Management, 2022 Publication	<1 %
13	Submitted to University of Western Ontario Student Paper	<1 %
14	dspace.daffodilvarsity.edu.bd:8080 Internet Source	<1 %
15	Submitted to Nanyang Technological University, Singapore Student Paper	<1 %
16	www.maxwell.vrac.puc-rio.br Internet Source	<1 %

17	Submitted to University of Salford Student Paper	<1 %
18	Submitted to Queen Mary and Westfield College Student Paper	<1 %
19	www.eeedept.griet.ac.in Internet Source	<1 %
20	www.slideshare.net Internet Source	<1 %
21	Submitted to Chiang Mai University Student Paper	<1 %
22	www.conference.bonfring.org Internet Source	<1 %
23	"Proceedings of International Conference on Big Data, Machine Learning and their Applications", Springer Science and Business Media LLC, 2021 Publication	<1 %
24	thesai.org Internet Source	<1 %
25	Submitted to London School of Economics and Political Science Student Paper	<1 %
26	www.groundai.com Internet Source	<1 %

27	"Soft Computing in Data Science", Springer Science and Business Media LLC, 2019 Publication	<1%
28	Pasi Luukka, Jouni Lampinen. "Chapter 11 A Classification method based on principal component analysis and differential evolution algorithm applied for prediction diagnosis from clinical EMR heart data sets", Springer Science and Business Media LLC, 2010 Publication	<1%

Exclude quotes	Off	Exclude matches	Off
Exclude bibliography	On		

Sambhav Choudhary

GRADEMARK REPORT

FINAL GRADE

/12

GENERAL COMMENTS

Instructor

PAGE 1

PAGE 2

PAGE 3

PAGE 4

PAGE 5

PAGE 6

PAGE 7

PAGE 8

PAGE 9

PAGE 10

PAGE 11

PAGE 12

PAGE 13

PAGE 14

PAGE 15

PAGE 16

PAGE 17

PAGE 18

PAGE 19

PAGE 20

PAGE 21

PAGE 22

PAGE 23

PAGE 24

PAGE 25

PAGE 26

PAGE 27

PAGE 28

PAGE 29

PAGE 30

PAGE 31

PAGE 32

PAGE 33