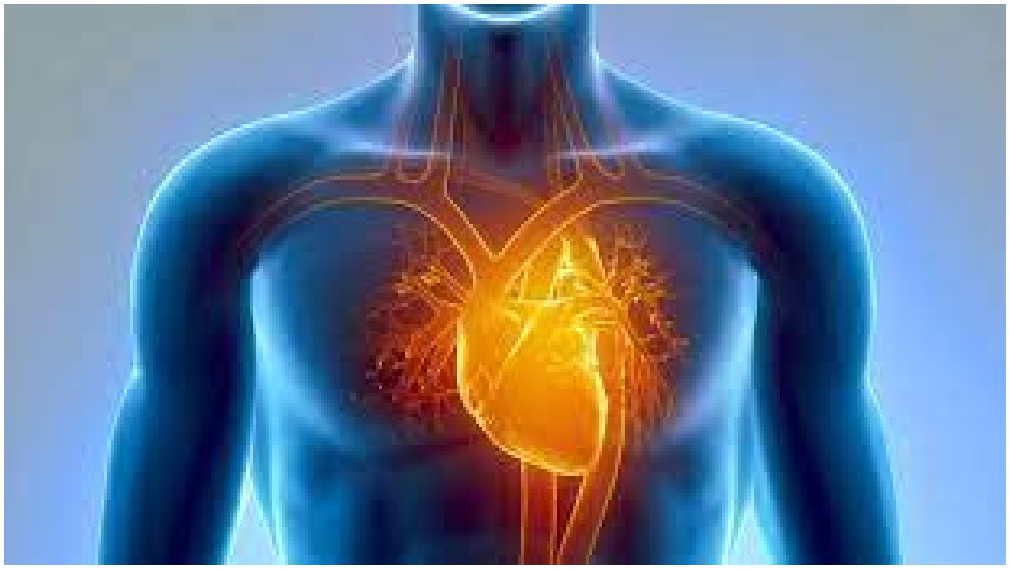
PROJECT REPORT

HEART DISEASE CLASSIFICATION

Predicting if a person is diagnosed with Heart disease or not



TEAM 2

20wh1a6606-J.Akanksha Sharma

20wh1a6632-K.Nikitha Reddy

20wh1a6637-T.Harshitha

20wh1a6638-Sk.Dil Nawaz Farida

20wh1a6646-B.Sreevidya

21wh5a6602-T.Nehasri

INTRODUCTION

Heart diseases are the leading cause of death globally. Each year, 17.9 million deaths occur, that is one death every other second. One third of these deaths occur below the age of 70 . A lot of effort is put in by researchers all over the world to provide prevention, help, relieve, and hopefully one day cure heart diseases.

Due to the lack of knowledge, many people are constantly dying due to sudden heart attacks. If the patient's heart condition is priorly examined then proper medication and prevention of disease could be done,this pioneered our project.

Approach:

The dataset is taken from the hospitals, which includes all the reports of a patient's health conditions.

Using those particular records we predict whether the patient would be diagnosed with Heart disease or not. We can classify the patients into two classes, the patients who have heart disease and those who don't.

Using a classifier the patients are divided into their respective classes.

Relevance to real world:

The classifier would help the patient's heart condition without becoming chronic.

TASK DEFINITION

There are 14 columns

1. Age: The age of the patient.
2. sex: The gender of the patient. (1 = male, 0 = female).
3. Cp: Type of chest pain. (1 = typical angina, 2 = atypical angina, 3 = non — anginal pain,
4. 4 = asymptotic).
5. Trestbps: Resting blood pressure in mmHg.
6. Chol: Serum Cholesterol in mg/dl.
7. Fbs: Fasting Blood Sugar. (1 = fasting blood sugar is more than 120mg/dl, 0 = otherwise).

8. Restecg: Resting ElectroCardioGraphic results (0 = normal, 1 = ST-T wave abnormality, 2 = left ventricular hypertrophy).
9. Thalach: Max heart rate achieved.
10. Exang: Exercise induced angina (1 = yes, 0 = no).
11. Oldpeak: ST depression induced by exercise relative to rest.
12. Slope: Peak exercise ST segment (1 = upsloping, 2 = flat, 3 = downsloping).
13. Ca: Number of major vessels (0–3) colored by fluoroscopy.
14. Thal: Thalassemia (3 = normal, 6 = fixed defect, 7 = reversible defect).
15. Num: Diagnosis of heart disease (0 = absence, 1, 2, 3, 4 = present).

	age	sex	cp	trestbps	chol	fbs	restecg	thalach	exang	oldpeak	slope	ca	thal	target
0	63	male	3	145	233	1	0	150	0	2.3	0	0	1	yes
1	37	male	2	130	250	0	1	187	0	3.5	0	0	2	yes
2	41	female	1	130	204	0	0	172	0	1.4	2	0	2	yes
3	56	male	1	120	236	0	1	178	0	0.8	2	0	2	yes
4	57	female	0		354	0	1	163	1	0.6	2	0	2	yes
5	57	male	0	140	192	0	1	148	0	0.4	1	0	1	yes
6	56	female	1	140	294	0	0	153	0	1.3	1	0	2	yes
7	44	male	1	120	263	0	1	173	0	0	2	0	3	yes
8	52	male	2	172	199	1	1	162	0	0.5	2	0	3	yes
9	57	male	2	150	168	0	1	174	0	1.6	2	0	2	yes
10	54	male	0	140	239	0	1	160	0	1.2	2	0	2	yes
11	48	female	2	130	275	0	1	139	0	0.2	2	0	2	yes
12	49	male	1	130	266	0	1	171	0	0.6	2	0	2	yes
13	64	male	3	110	211	0	0	144	1	1.8	1	0	2	yes
14	58	female	3	150	283	1	0	162	0	1	2	0	2	yes

METHODOLOGY

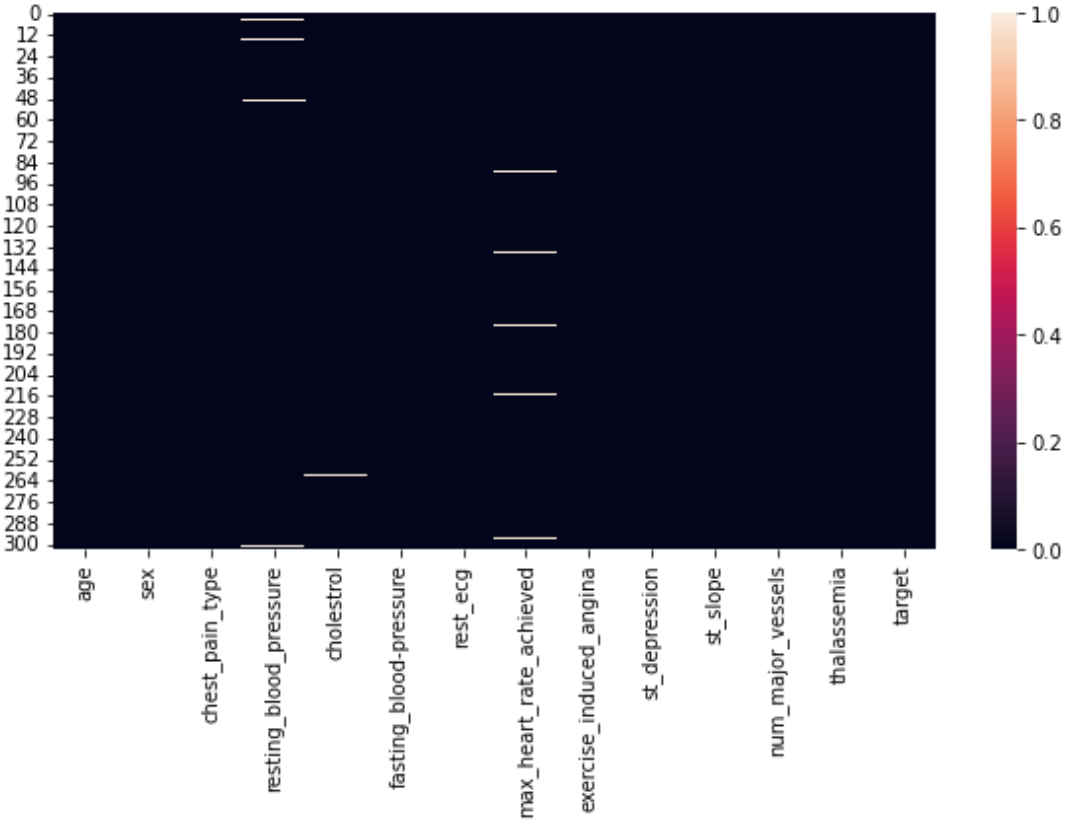
Data preprocessing:

- Downloading the dataset
- Link: https://www.kaggle.com/c/heart-disease-uci/data

  https://github.com/20WH1A6637/Heart-Disease-Classification/blob/main/hdc_dataset.csv
- Analyzing the dataset and column.
- Delete the unproductive columns(Unnamed :0)
- Data exploration and replacing the missing values with mean/median or mode based on the data type(numerical and categorical)

- The next task was splitting the dataset. So this gave us 70 percent of Training Data, 20 percent of Testing Data and 10 percent of Validation Data.
- Using correlation for dropping columns – A heatmap was plotted.

It is used for understanding the data better.Since there are only 14 columns,no column is being dropped.

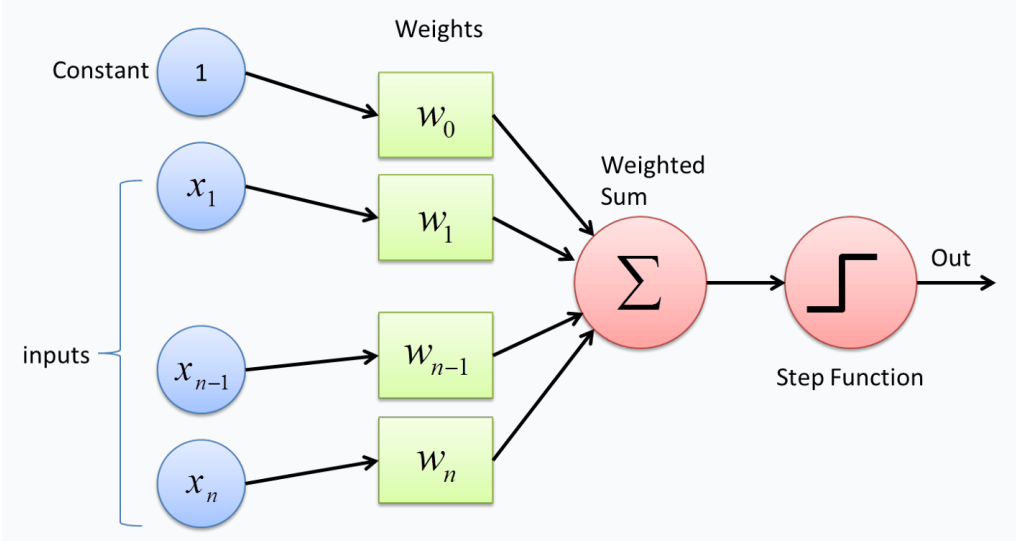


LIBRARIES AND MODELS USED

- Different types of python libraries such as pandas, Sklearn, NumPy, matplotlib are used for implementing the algorithms.
- **Perceptron :** The Perceptron is used to classify data into two parts hence it is called Linear Binary Classifier.

  The perceptron consists of 4 parts:

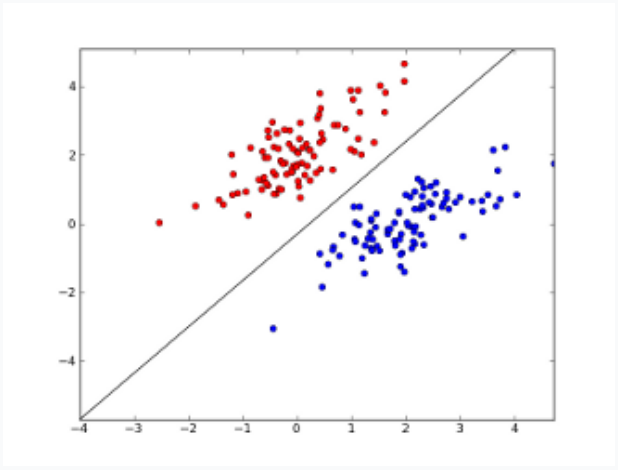
  1. Input values or one input layer
  2. Weights and Bias
  3. Net sum
  4. Activation Function



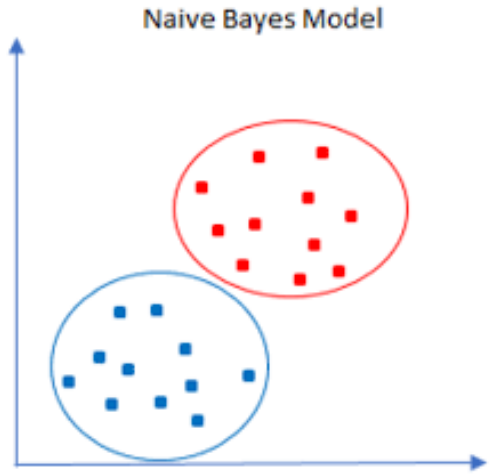
Perceptron works on these simple steps:

1. All the inputs x are multiplied with their weights w .
2. Add all the multiplied values and call them Weighted Sum.
3. Apply that weighted sum to correct Activation function

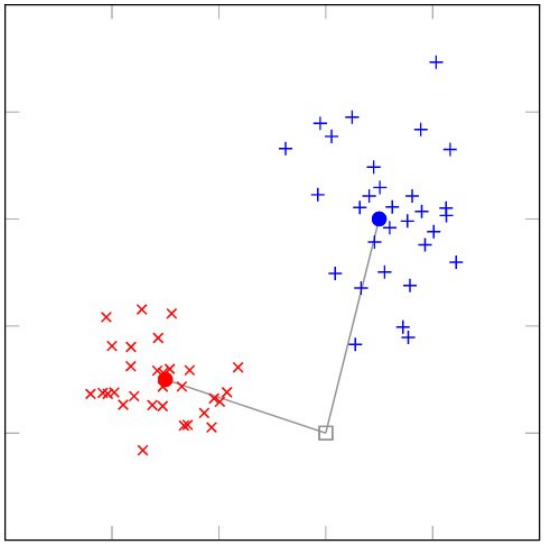
- Weights show the strength of the particular node.
- A bias value allows you to shift the activation function curve up or down.
- Activation functions are used to map the input between the required values .



- **Bernoulli Naive Bayes :** Naive Bayes classifier is a probabilistic classifier based on applying Bayes' theorem with strong(naive) independence assumptions between the features. A Naive Bayesian model is easy to build, with no complicated iterative parameter estimation which makes it particularly useful in the field of medical science for diagnosing heart patients.



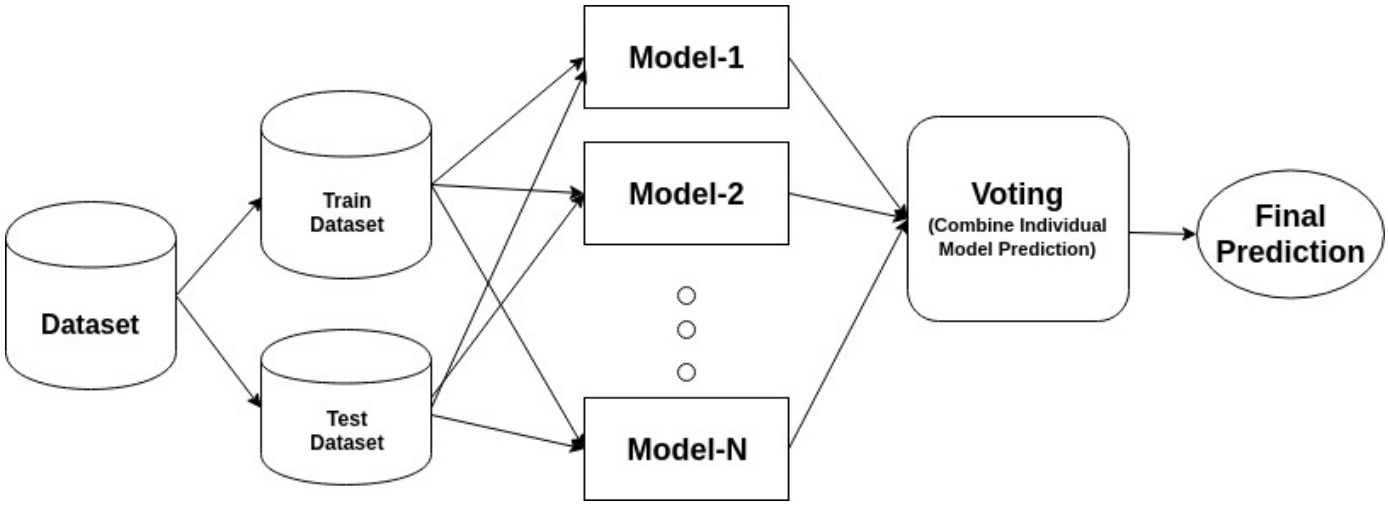
- **Nearest Centroid :** In machine learning, a nearest centroid classifier or nearest prototype classifier is a classification model that assigns to observations the label of the class of training samples whose mean (centroid) is closest to the observation.



- **Calibrated ClassifierCV :** There are two concerns in calibrating probabilities; they are diagnosing the calibration of predicted probabilities and the calibration process itself.
- **Adaboost Classifier** : It combines multiple classifiers to increase the accuracy of classifiers. AdaBoost is an iterative ensemble method. AdaBoost classifier builds a strong classifier by combining multiple poorly performing classifiers so that you will get high accuracy strong classifier. The basic concept behind Adaboost is to set the weights of classifiers and training the data sample in each iteration such that it ensures the accurate predictions of unusual observations. Any machine learning algorithm can be used as base classifier if it accepts weights on the training set. Adaboost should meet two conditions:

  1.The classifier should be trained interactively on various weighed training examples.

  2.In each iteration, it tries to provide an excellent fit for these examples by minimizing training error.



DATA

Model	Validation Score	Test Score
Perceptron	0.89	0.41
Bernoulli Naive Bayes	0.89	0.89
Nearest Centroid	0.89	0.76
AdaBoost Classifier	0.87	0.93
Calibrated Classifier CV	0.87	0.86

RESULTS

Finally, we got highest validation accuracy score of 0.89 for perceptron classifier, Bernoulli Naive Bayes and nearest centroid. Where as AdaBoost Classifier got the highest test accuracy score of about 0.93. The following may be the reasons for change in final test score:

- The Perceptron is a linear machine learning algorithm for binary classification tasks. This means that it learns a decision boundary that separates two classes using a line (called a hyperplane) in the feature space. As such, it is appropriate for those problems where the classes can be separated well by a line or linear model, referred to as linearly separable.
- Adaboost is a meta-learning machine learning (ML) algorithm, i.e., it can be used on top of any other ML algorithm. A perceptron classifier is non meta-learning ML.
- If you have no hidden layer, then perceptron is as good as a linear classifier, if it has one or more hidden layers then it is non-linear classifier. If it is deep (or multiple layers), then hierarchical features can be learned. The output of a perceptron is the linear combination of the feature and their associated weights.
- Adaboost does not learn features as such. In Adaboost, the output of the other learning algorithms ('weak learners') is combined into a weighted sum that represents the final output of the boosted classifier. AdaBoost is adaptive in the sense that subsequent weak learners are tweaked in favor of those instances misclassified by previous classifiers.

DEPLOYMENT

Deployment is the method by which you integrate a machine learning model into an existing production environment to make practical business decisions based on data. It is one of the last stages in the machine learning life cycle and can be one of the most cumbersome.

In order to start using a model for practical decision-making, it needs to be effectively deployed into production. If you cannot reliably get practical insights from your model, then the impact of the model is severely limited.

DEPLOYMENT USING GRADIO:

Gradio is an open-source python library that allows you to quickly create easy-to-use,

customizable UI components for your machine learning model.

Gradio allows you to integrate the GUI directly into your Python notebook making it easier to use.

Enables you to create demos of your machine learning model. These demos can be used to present ideas to clients, users, and team members before the actual application is implemented.

We created a simple interface using gradio with 13 inputs and a single output which gives the predicted output whether a person is diagnosed with a heart disease or not.

GOOGLE COLAB NOTEBOOK:

https://colab.research.google.com/drive/1a-Z5PvWBWTgxD4cGpIULBlpxl2VMjFpy#scrollTo=rcHeNq6OUQoN

CONCLUSION

In conclusion, this kind of model can be very useful for people to predict the heart condition prior and prevention of or treatment for the disease is provided at an early stage.

REFERENCES

- https://www.datacamp.com/tutorial/tutorial-data-cleaning-tutorial
- https://towardsdatascience.com/data-cleaning-how-to-handle-missing-values-in-pandas-cc8570c446ec
- https://medium.com/@work.gracesophie/what-are-the-5-elements-of-a-problem-statement-41576afd6312
- https://www.geeksforgeeks.org/naive-bayes-classifiers/
- http://scikitlearn.org/stable/modules/neighbors.html
- https://gradio.app/docs/