Vidyavardhini's College Of Engineering & Technology
Department of Instrumentation Engineering

"Heart Disease Identification Method Using Machine Learning Classification In E-Healthcare"

Guide name: **Prof. Vishal Pande**

Project By:-

Sujit Shibaprasad Maity.

Title and Content Layout

1. INTRODUCTION
2. LITERATURE REVIEW
3. PROBLEM STATEMENT & IDENTIFICATION
4. REQUIREMENT
5. ALGORITHM & METHODOLOGY
6. BLOCK DIAGRAM
7. FLOW CHART
8. EXECUTION
9. CONCLUSION & REFERENCES

1. INTRODUCTION

- Millions of people today are suffering from various diseases that can prove fatal. Diseases like cancer, heart diseases, diabetes, etc. cause a lot of health problems which may even lead to death. Identification of such diseases in early stages can help solve a lot of conditions related to them. Diagnosis of diseases at the right time will be of great help. Therefore, to help the diagnosis process many data mining and machine learning techniques can be used

1.1. Aim:

To apply machine learning techniques resulting in improving the accuracy in the prediction of Heart disease.

1.2. Existing System:

The obtained results are compared with the results of existing models within the same domain and found to be improved. The data of heart disease patients collected from the UCI laboratory is used to discover patterns with NN, Naive Bayes. The results are compared for performance and accuracy with these algorithms. The proposed hybrid method returns results of 86:8% for F-measure, competing with the other existing methods.

1.3. Proposed System:

ML process starts from a pre-processing data phase followed by feature selection based on data cleaning, classification of modeling, performance evaluation, and the results with improved accuracy. We create a web application through the user input to predict a heart disease.

2. LITERATURE REVIEW

- Detrano et al. developed HD classification system by using machine learning classification techniques and the performance of the system was 77% in terms of accuracy.
- Gudadhe et al. developed a diagnosis system using multi-layer Perceptron and support vector machine (SVM) algorithms for HD classification and achieved accuracy 80.41%.
- Humar et al. designed HD classification system by utilizing a neural network with the integration of Fuzzy logic. The classification system achieved 87.4% accuracy.
- Resul et al. developed an ANN ensemble based diagnosis system for HD along with statistical measuring system enterprise miner (5.2) and obtained the accuracy of 89.01%, sensitivity 80.09%, and specificity 95.91%.
- Liu et al. proposed a HD classification system using relief and rough set techniques. The proposed method achieved 92.32% classification accuracy.
- Palaniappan et al. proposed an expert medical diagnosis system for HD identification. In development of the system the predictive model of machine learning, such as navies bays (NB), Decision Tree (DT), and Artificial Neural Network were used. The 86.12% accuracy was achieved by NB, ANN accuracy 88.12% and DT classifier achieved 80.4% accuracy
- Olaniyi et al. developed a three-phase technique based on the artificial neural network technique for HD prediction in angina and achieved 88.89% accuracy.
- Samuel et al. developed an integrated medical decision support system based on artificial neural network and Fuzzy AHP for diagnosis of HD. The performance of the proposed method in terms of accuracy was 91.10%.

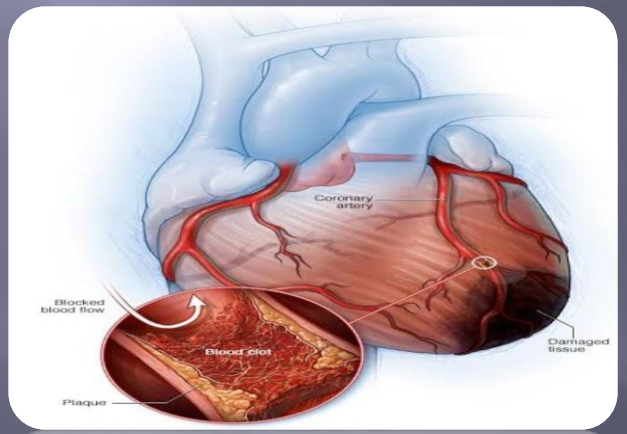
- In literature various machine learning based diagnosis techniques have been proposed by researchers to diagnosis HD. This research study present some existing machine learning based diagnosis techniques in order to explain the important of the proposed work.

Researchers	Techniques Used	Accuracy
Detrano et al.	HD Diagnosis using ML classification techniques.	77
Gudadhe et al.	Multi-Layer Perceptron and Support Vector Machine (SVM) Algorithms.	80.41
Humar et al.	ANN and Fuzzy Logic.	87.4
Resul et al.	ANN Ensemble based Diagnosis techniques.	89.01
Akil et al.	Diagnosis System based on Navies Bays, ANN and Decision Tree.	88.12
Palaniappan et al.	Three Phase Technique based on ANN.	88.89
Olaniyi et al.	ANN and Fuzzy AHP.	91.1
Samuel et al.	Relief Rough Set based Method for HD Detection.	92.32
Liu et al.	Hybrid ML Method.	88.07

3. PROBLEM STATEMENT & IDENTIFICATION

PROBLEM STATMENT

- We have a Data Set which Classified if Patient have a HD or not According to its Feature.
- We have Create a Model which will Predict if a Patients have this HD or not



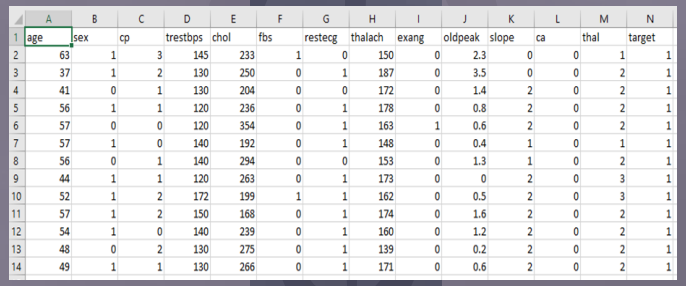
PROBLEM IDENTIFICATION

- Identification of heart diseases in early stages can help solve a lot of conditions related to them.
- To help the diagnosis process many data mining and machine learning techniques can be used.

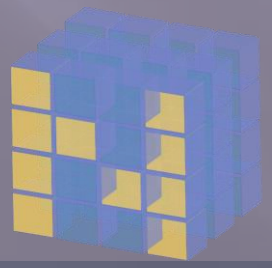
4. REQUIREMENT

SOFTWARE REQUIREMENTS

- Operating system:
  Windows 7,8,10,11(64 bit).
- Software:
  Python 3.7, Anaconda.
- Tools:
  Jupiter Note Book IDE,VS Code.
- Csv File:
  Data Set.



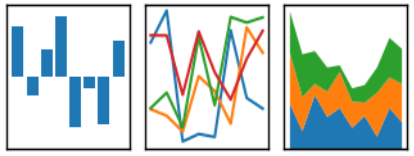
	A	B	C	D	E	F	G	H	I	J	K	L	M	N
1	age	sex	cp	trestbps	chol	fbs	restecg	thalach	exang	oldpeak	slope	ca	thal	target
2	63	1	3	145	233	1	0	150	0	2.3	0	0	1	1
3	37	1	2	130	250	0	1	187	0	3.5	0	0	2	1
4	41	0	1	130	204	0	0	172	0	1.4	2	0	2	1
5	56	1	1	120	236	0	1	178	0	0.8	2	0	2	1
6	57	0	0	120	354	0	1	163	1	0.6	2	0	2	1
7	57	1	0	140	192	0	1	148	0	0.4	1	0	1	1
8	56	0	1	140	294	0	0	153	0	1.3	1	0	2	1
9	44	1	1	120	263	0	1	173	0	0	2	0	3	1
10	52	1	2	172	199	1	1	162	0	0.5	2	0	3	1
11	57	1	2	150	168	0	1	174	0	1.6	2	0	2	1
12	54	1	0	140	239	0	1	160	0	1.2	2	0	2	1
13	48	0	2	130	275	0	1	139	0	0.2	2	0	2	1
14	49	1	1	130	266	0	1	171	0	0.6	2	0	2	1

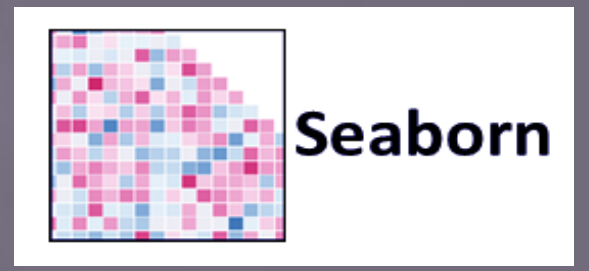


NumPy

pandas

$$y_{it} = \beta' x_{it} + \mu_i + \epsilon_{it}$$



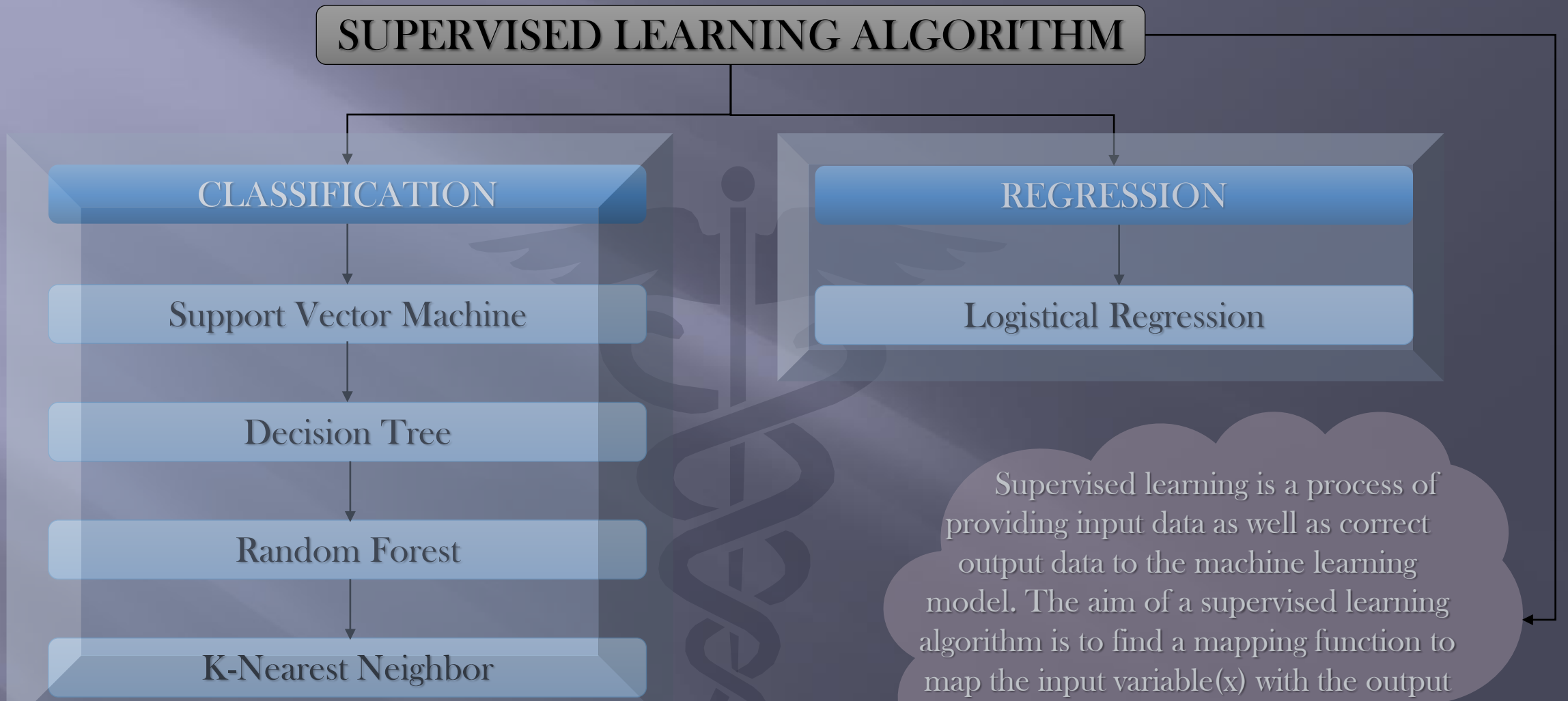


matplotlib

HARDWARE REQUIREMENTS

- Hard Disk:
  500GB & Above
- RAM:
  4GB & Above
- Processor:
  I3 & Above

5. ALGORITHM & METHODOLOGY



Supervised learning is a process of providing input data as well as correct output data to the machine learning model. The aim of a supervised learning algorithm is to find a mapping function to map the input variable(x) with the output variable(y).

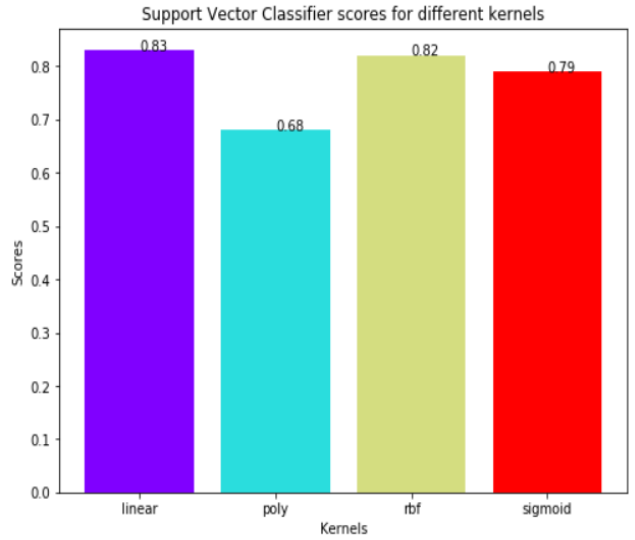
Contnd....

CLASSIFICATION & STEPS

- Classification algorithms are used when the output variable is categorical, which means there are two classes such as Yes-No, Male-Female, True-false, etc.

Support Vector Classifier – 83.0%

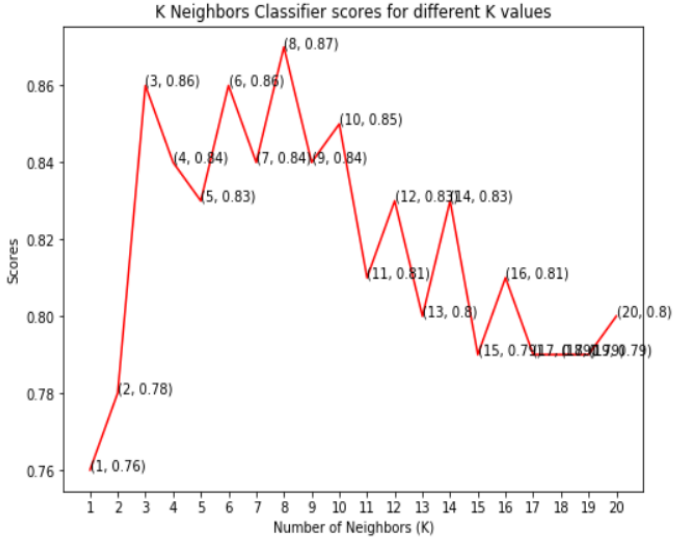
```
In [20]: colors = rainbow(np.linspace(0, 1, len(kernels)))
         plt.bar(kernels, svc_scores, color = colors)
         for i in range(len(kernels)):
             plt.text(i, svc_scores[i], svc_scores[i])
         plt.xlabel('Kernels')
         plt.ylabel('Scores')
         plt.title('Support Vector Classifier scores for different kernels')

Out[20]: Text(0.5, 1.0, 'Support Vector Classifier scores for different kernels')
```




```
In [21]: print("The score for Support Vector Classifier is {}% with {} kernel.".format(svc_scores[0]*100, 'linear'))

         The score for Support Vector Classifier is 83.0% with linear kernel.
```

K-Neighbors Classifier - 87.0%

```
In [17]: plt.plot([k for k in range(1, 21)], knn_scores, color = 'red')
         for i in range(1,21):
             plt.text(i, knn_scores[i-1], (i, knn_scores[i-1]))
         plt.xticks([i for i in range(1, 21)])
         plt.xlabel('Number of Neighbors (K)')
         plt.ylabel('Scores')
         plt.title('K Neighbors Classifier scores for different K values')

Out[17]: Text(0.5, 1.0, 'K Neighbors Classifier scores for different K values')
```




```
In [18]: print("The score for K Neighbors Classifier is {}% with {} nieghbors.".format(knn_scores[7]*100, 8))

         The score for K Neighbors Classifier is 87.0% with 8 nieghbors.
```

Contnd....

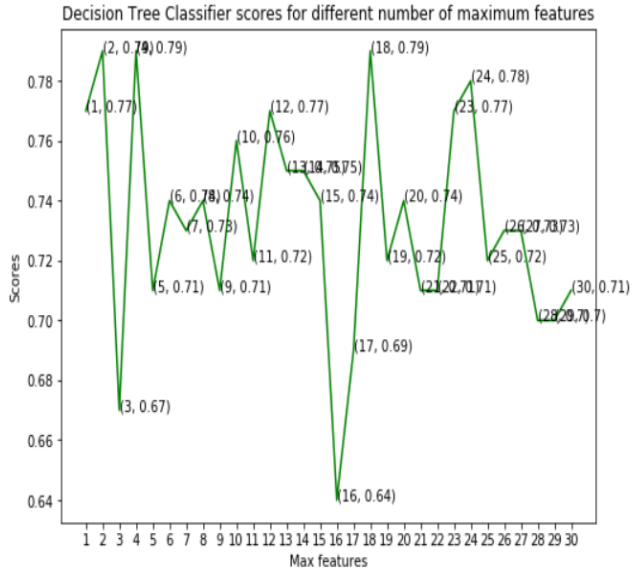
3 step process

Split the dataset → Train the dataset → Compare the Algos

Decision Tree Classifier - 79.0%

```
In [23]: plt.plot([i for i in range(1, len(X.columns) + 1)], dt_scores, color = 'green')
         for i in range(1, len(X.columns) + 1):
             plt.text(i, dt_scores[i-1], (i, dt_scores[i-1]))
         plt.xticks([i for i in range(1, len(X.columns) + 1)])
         plt.xlabel('Max features')
         plt.ylabel('Scores')
         plt.title('Decision Tree Classifier scores for different number of maximum features')
```

```
Out[23]: Text(0.5, 1.0, 'Decision Tree Classifier scores for different number of maximum features')
```



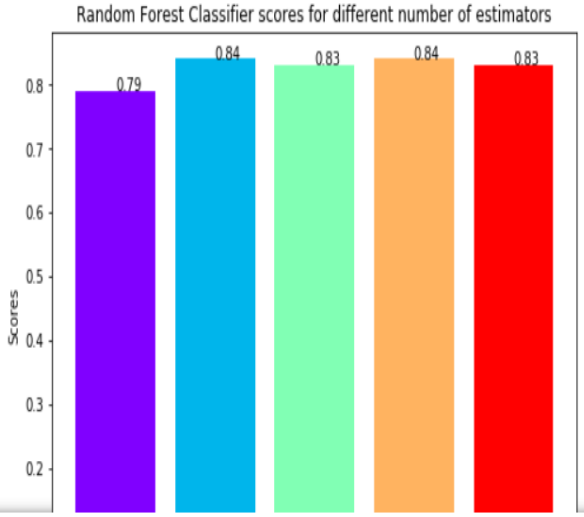

```
In [24]: print("The score for Decision Tree Classifier is {}% with {} maximum features.".format(dt_scores[17]*100, [2,4,18]))
```

```
The score for Decision Tree Classifier is 79.0% with [2, 4, 18] maximum features.
```

Random Forest Classifier - 84.0%

```
In [26]: colors = rainbow(np.linspace(0, 1, len(estimators)))
         plt.bar([i for i in range(len(estimators))], rf_scores, color = colors, width = 0.8)
         for i in range(len(estimators)):
             plt.text(i, rf_scores[i], rf_scores[i])
         plt.xticks(ticks = [i for i in range(len(estimators))], labels = [str(estimator) for estimator in estimators])
         plt.xlabel('Number of estimators')
         plt.ylabel('Scores')
         plt.title('Random Forest Classifier scores for different number of estimators')
```

```
Out[26]: Text(0.5, 1.0, 'Random Forest Classifier scores for different number of estimators')
```




```
In [27]: print("The score for Random Forest Classifier is {}% with {} estimators.".format(rf_scores[1]*100, [100, 500]))
```

```
The score for Random Forest Classifier is 84.0% with [100, 500] estimators.
```

Contnd....

REGRESSION

- Regression algorithms are used if there is a relationship between the input variable and the output variable. It is used for the prediction of continuous variables, such as Weather forecasting, Market Trends, etc.

Logistic Regression:

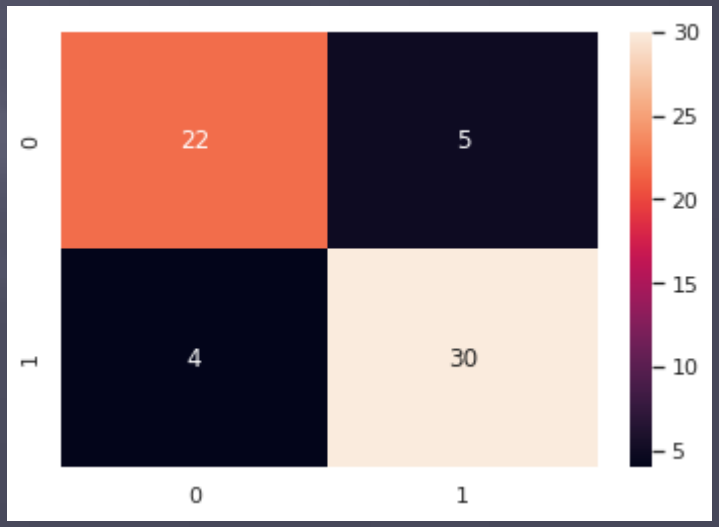
```
from sklearn.linear_model import LogisticRegression
logreg = LogisticRegression()

logreg.fit(X_train, Y_train)

y_pred_lr = logreg.predict(X_test)
print(y_pred_lr)
```

Accuracy score : 85.25 %

Precision: 0.85

Recall is: 0.88

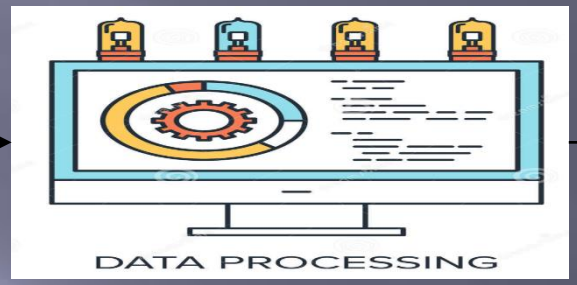
F-Score: 0.86



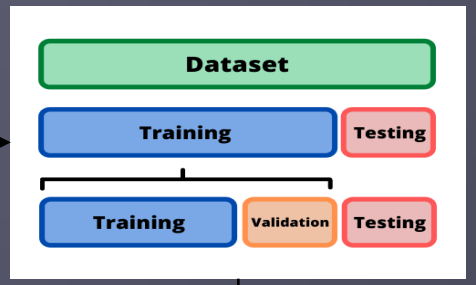
6. BLOCK DIAGRAM

age	sex	cp	trestbps	chol	fbs	restecg	thalach	exang	oldpeak	slope	ca	thal	target
63	1	3	145	233	1	0	150	0	2.3	0	0	1	1
37	1	2	130	250	0	1	187	0	3.5	0	0	2	1
41	0	1	130	204	0	0	172	0	1.4	2	0	2	1
56	1	1	120	236	0	1	178	0	0.8	2	0	2	1
57	0	0	120	354	0	1	163	1	0.6	2	0	2	1
57	1	0	140	192	0	1	148	0	0.4	1	0	1	1
56	0	1	140	294	0	0	153	0	1.3	1	0	2	1
44	1	1	120	263	0	1	173	0	0	2	0	3	1
52	1	2	172	199	1	1	162	0	0.5	2	0	3	1
57	1	2	150	168	0	1	174	0	1.6	2	0	2	1
54	1	0	140	239	0	1	160	0	1.2	2	0	2	1
48	0	2	130	275	0	1	139	0	0.2	2	0	2	1
49	1	1	130	266	0	1	171	0	0.6	2	0	2	1

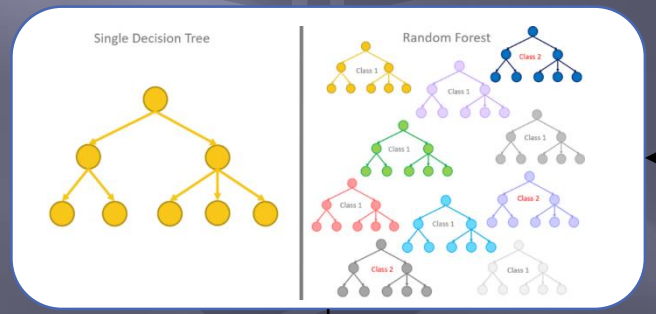
Raw Data Set



Data Preprocessing



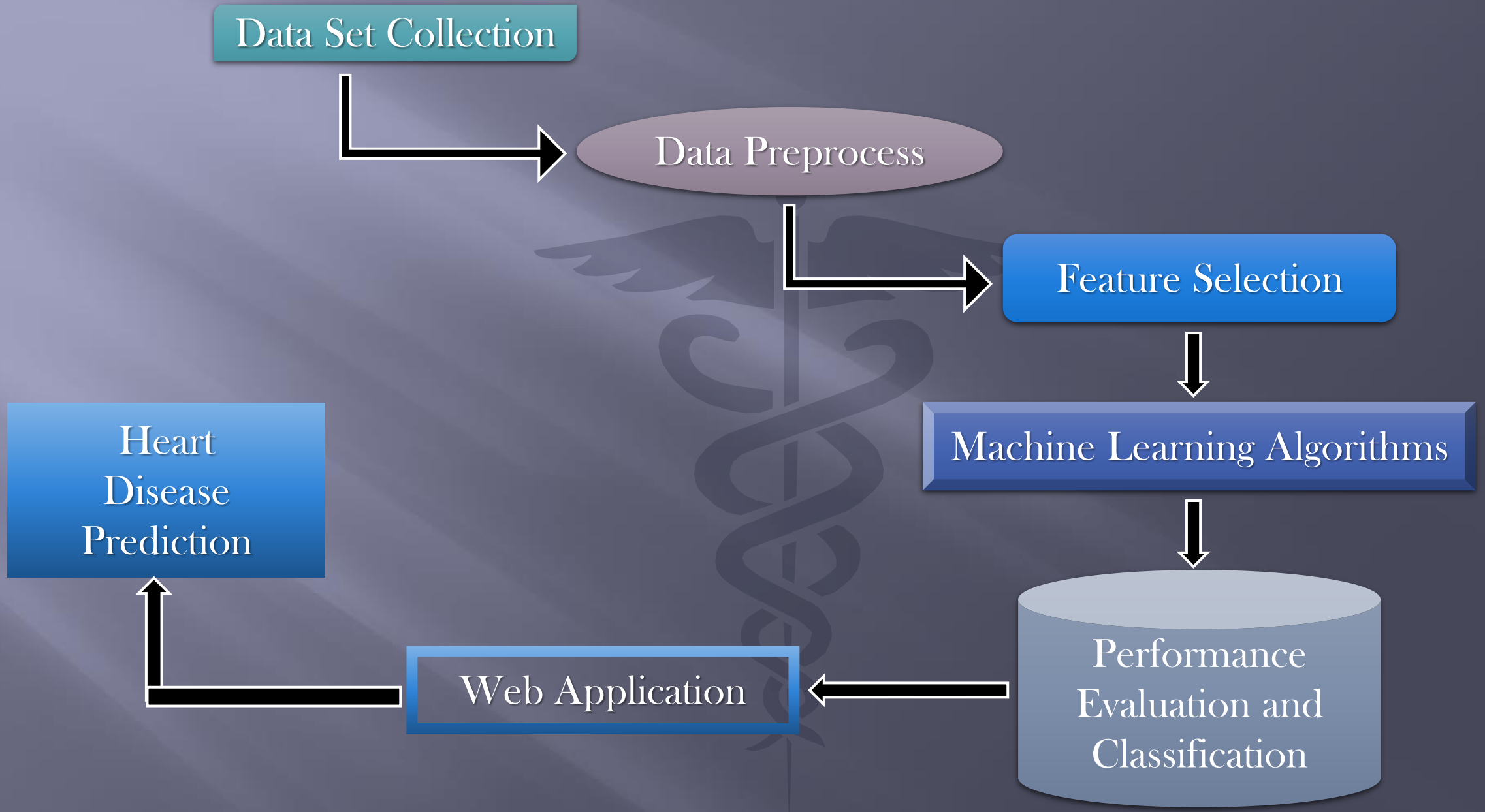
Train Test Split



Random Forest Model

Trained Model

7. FLOW CHART


Data Set Collection
Data Preprocess
Feature Selection
Machine Learning Algorithms
Performance Evaluation and Classification
Web Application
Heart Disease Prediction

8. EXECUTION

Data Preprocess

- We cannot Set Raw Data to the System first we have to process the data to fit or compactable for our Machine Learning Algorithm.

```
(base) C:\Users\sujit maity\Heart Disease Identification Method Using Machine Learning\M1>python preprocess.py
Cleveland data. Size=(302, 14)
Number of missing values
age         0
sex         0
cp          0
trestbps    0
chol        0
fbs         0
restecg     0
thalach     0
exang       0
oldpeak     0
slope       0
ca          4
thal        2
num         0
dtype: int64

Hungary data:. Size=(293, 14)
Number of missing values
age           0
sex           0
cp            0
trestbps      1
chol         23
fbs           8
restecg       1
thalach       1
exang         1
oldpeak       0
slope       189
ca          290
thal        265
num           0
dtype: int64
```

```
age         0
sex         0
cp          0
trestbps    0
chol        0
fbs         0
restecg     0
thalach     0
exang       0
num         0
dtype: int64
     age  sex  cp  trestbps   chol  fbs  restecg  thalach  exang  num
0     67    1   4     160.0  286.0  0.0      2.0    108.0    1.0    2
1     67    1   4     120.0  229.0  0.0      2.0    129.0    1.0    1
2     37    1   3     130.0  250.0  0.0      0.0    187.0    0.0    0
3     41    0   2     130.0  204.0  0.0      2.0    172.0    0.0    0
4     56    1   2     120.0  236.0  0.0      0.0    178.0    0.0    0
..   ...  ...  ..       ...    ...  ...      ...      ...    ...  ...
911   54    0   4     127.0  333.0  1.0      1.0    154.0    0.0    1
912   62    1   1     130.0  139.0  0.0      1.0    140.0    0.0    0
```

Feature Selection

- We have use Four Algorithms to test, which Algorithm has high level Accuracy of Prediction.

jupyter ML algorithms Last Checkpoint: 11/20/2021 (autosaved)

Logout

File Edit View Insert Cell Kernel Widgets Help Trusted Python 3

Run Code

In [1]:
```
import numpy as np
import pandas as pd
```

In [2]:
```
import matplotlib.pyplot as plt
from matplotlib import rcParams
from matplotlib.cm import rainbow
%matplotlib inline
import warnings
warnings.filterwarnings('ignore')
```

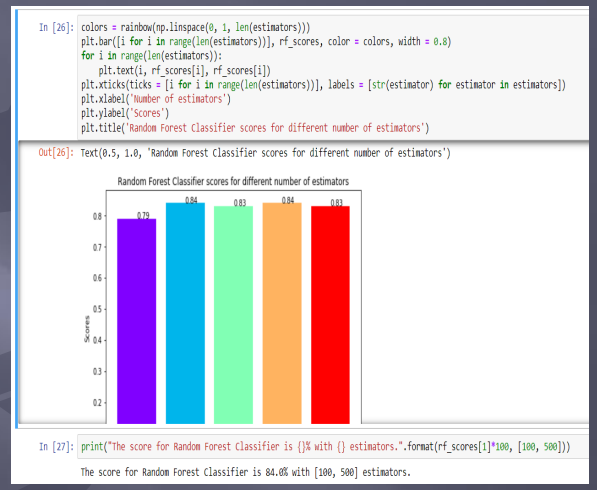
In [3]:
```
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import StandardScaler
```

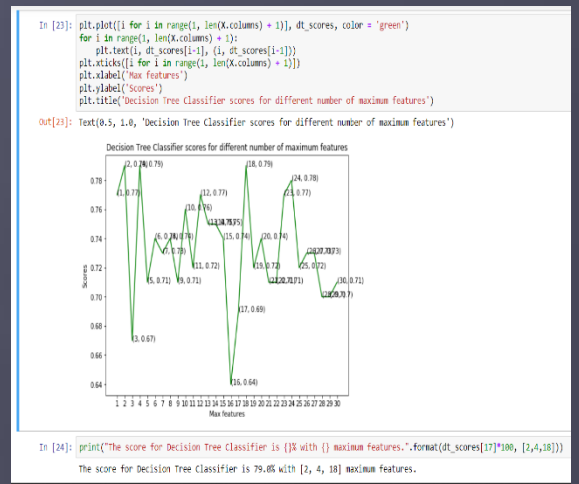
In [4]:
```
from sklearn.neighbors import KNeighborsClassifier
from sklearn.svm import SVC
from sklearn.tree import DecisionTreeClassifier
from sklearn.ensemble import RandomForestClassifier
```

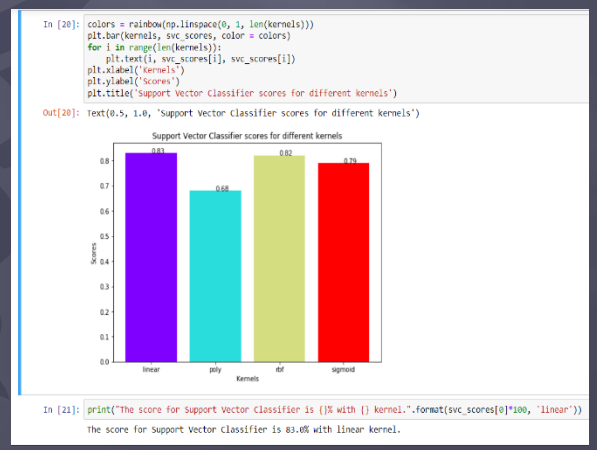
In [5]:
```
dataset = pd.read_csv('dataset.csv')
dataset
```

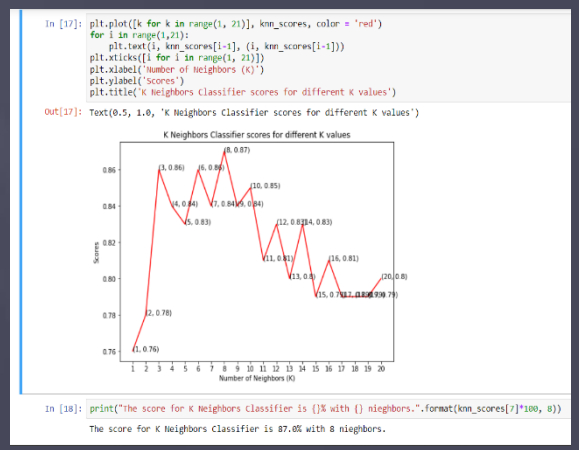
Out[5]:

	age	sex	cp	trestbps	chol	fbs	restecg	thalach	exang	oldpeak	slope	ca	thal	target
0	63	1	3	145	233	1	0	150	0	2.3	0	0	1	1
1	37	1	2	130	250	0	1	187	0	3.5	0	0	2	1
2	41	0	1	130	204	0	0	172	0	1.4	2	0	2	1
3	56	1	1	120	236	0	1	178	0	0.8	2	0	2	1
4	57	0	0	120	354	0	1	163	1	0.6	2	0	2	1
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
298	57	0	0	140	241	0	1	123	1	0.2	1	0	3	0







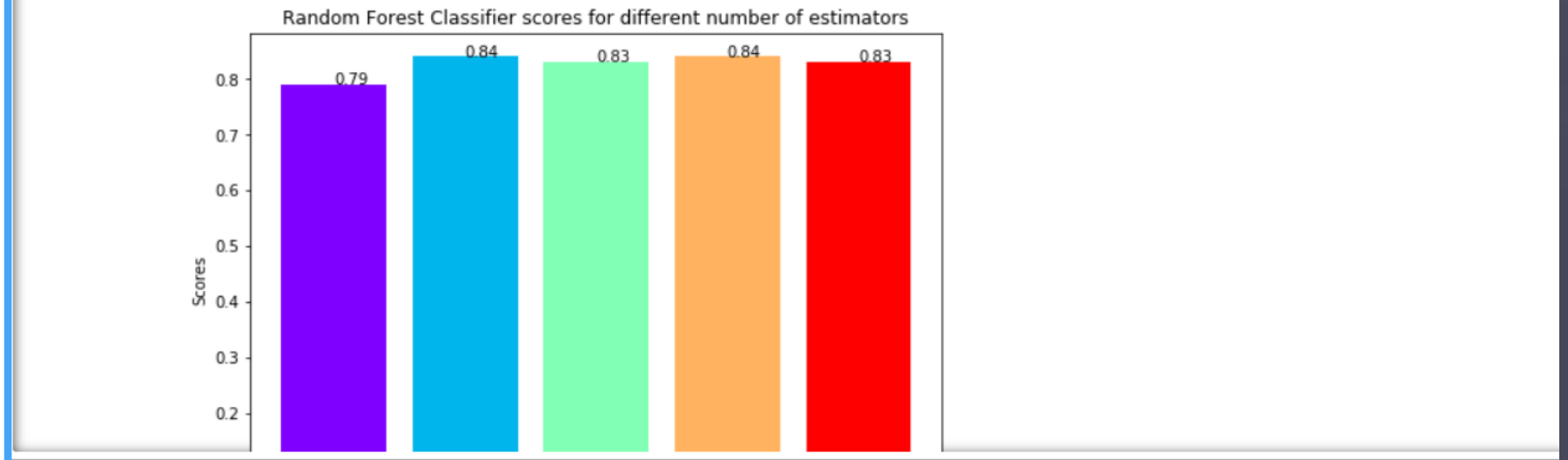


Machine Learning Algorithms

- Out of Four Algorithms, Random Forest Classifier have high level Accuracy of Prediction.

```
In [26]: colors = rainbow(np.linspace(0, 1, len(estimators)))
         plt.bar([i for i in range(len(estimators))], rf_scores, color = colors, width = 0.8)
         for i in range(len(estimators)):
             plt.text(i, rf_scores[i], rf_scores[i])
         plt.xticks(ticks = [i for i in range(len(estimators))], labels = [str(estimator) for estimator in estimators])
         plt.xlabel('Number of estimators')
         plt.ylabel('Scores')
         plt.title('Random Forest Classifier scores for different number of estimators')
```

```
Out[26]: Text(0.5, 1.0, 'Random Forest Classifier scores for different number of estimators')
```

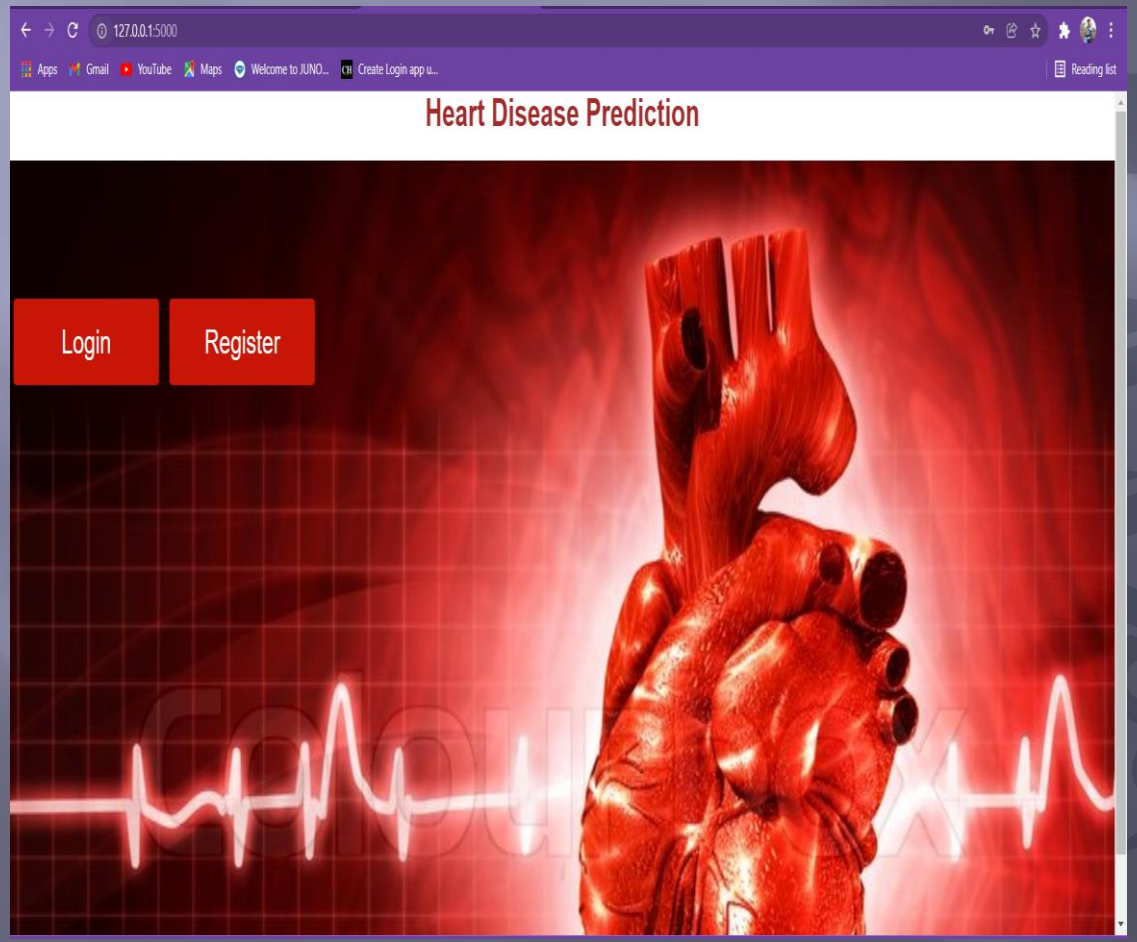


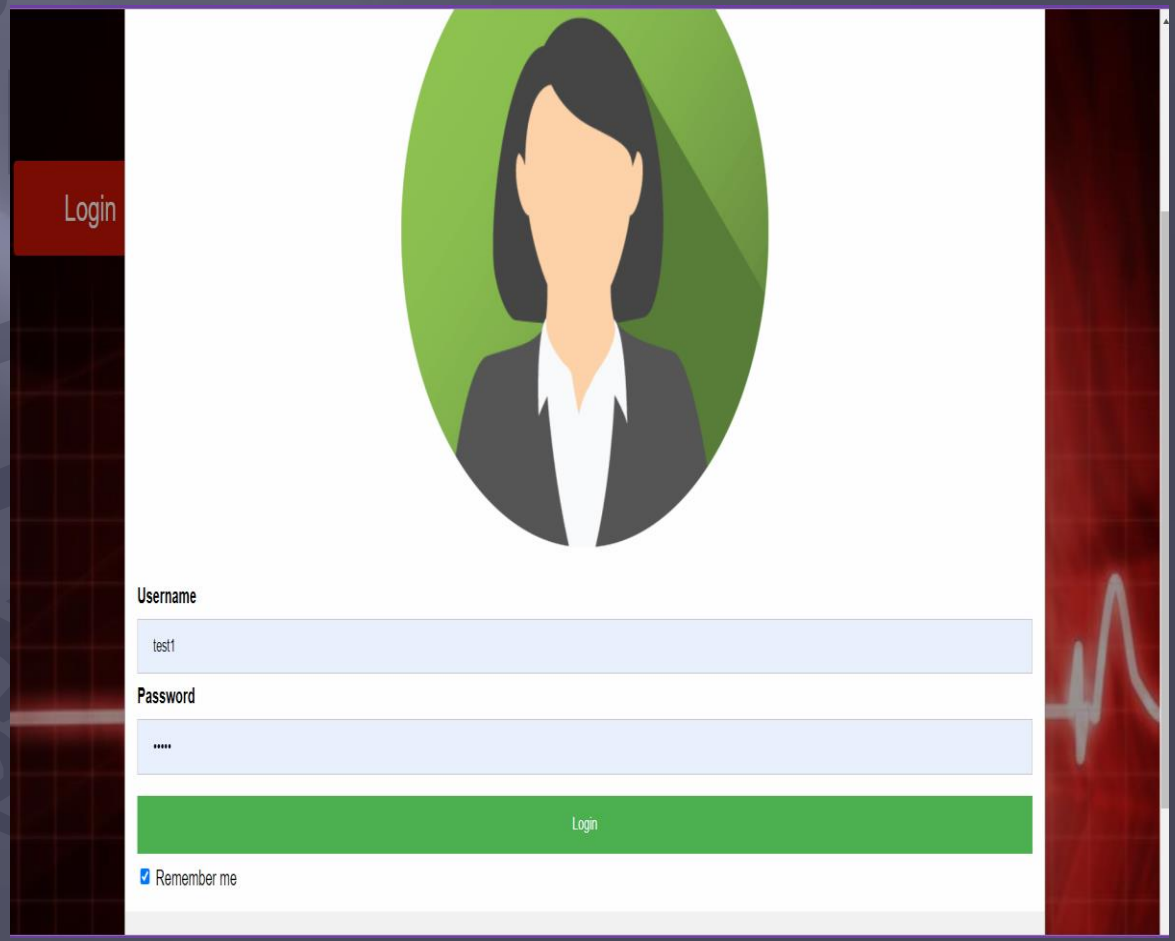

```
In [27]: print("The score for Random Forest Classifier is {}% with {} estimators.".format(rf_scores[1]*100, [100, 500]))
```

```
The score for Random Forest Classifier is 84.0% with [100, 500] estimators.
```

Web Application

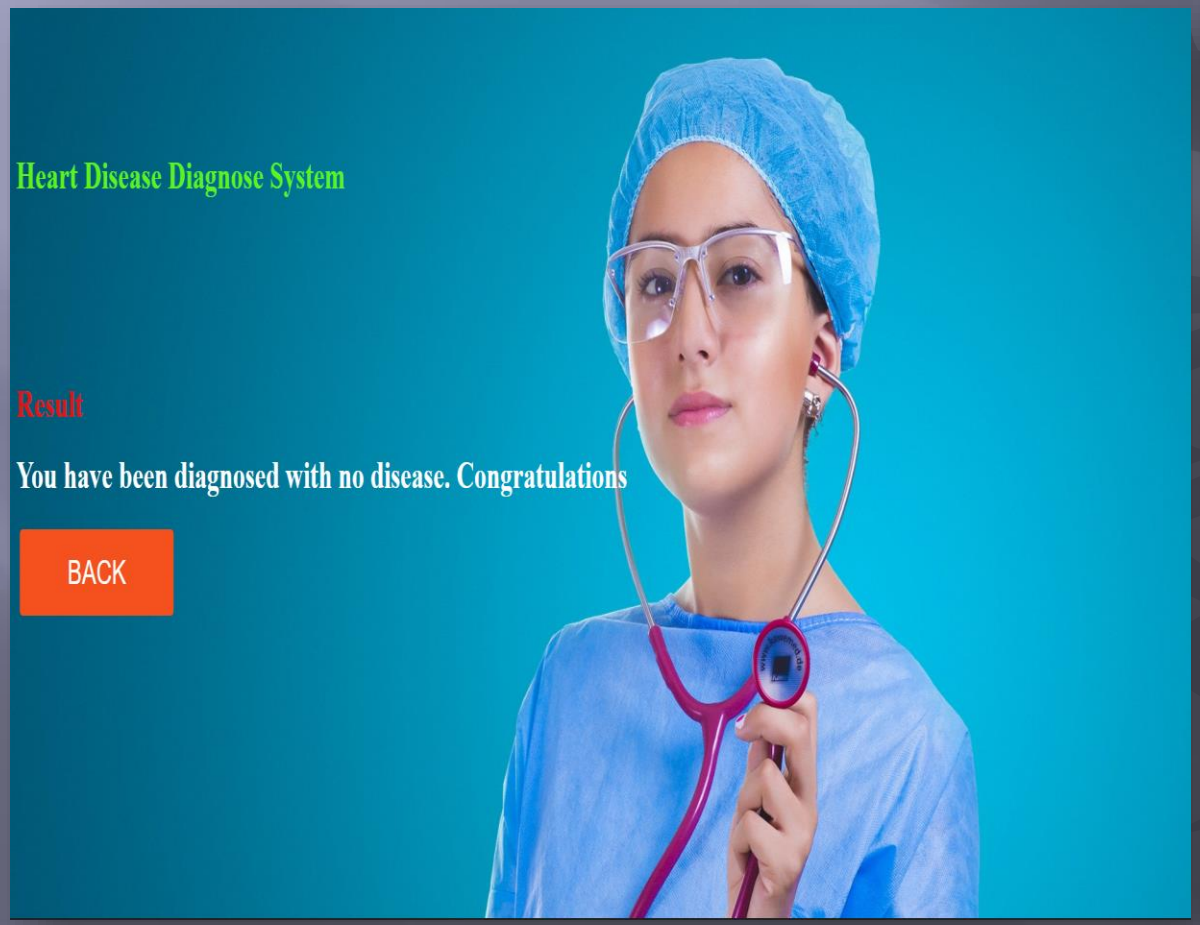
- We developed Web Application for the End User to use our Proposed System in better way and also for Privacy.



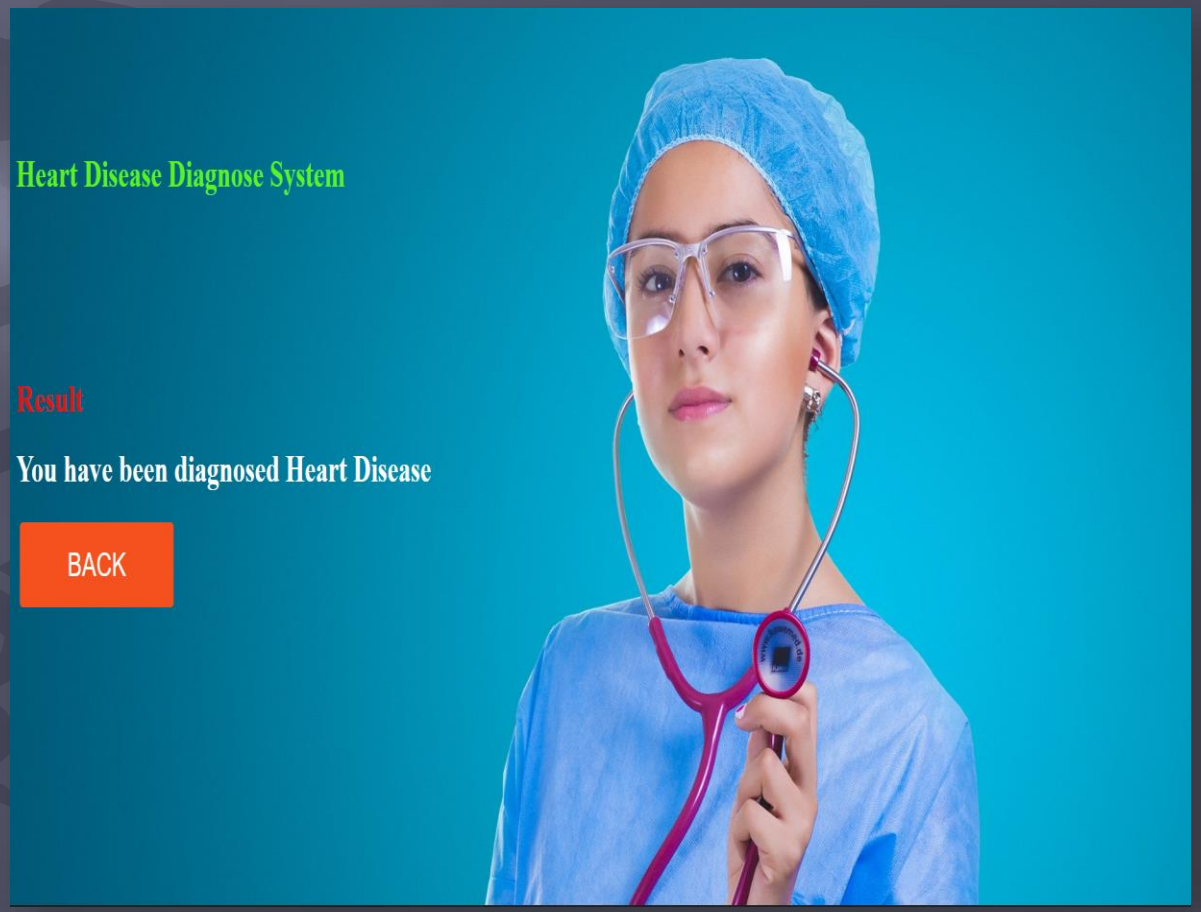


Heart Disease Prediction

IF YES



IF NO



9. CONCLUSION & REFERENCES

CONCLUSION

- In this study, an efficient machine learning based diagnosis system has been developed for the diagnosis of heart disease.
- The experimental results show that the proposed feature selection algorithm is feasible with classifier Random Forest Classifier for designing a high-level intelligent system and with high Accuracy to identify heart disease. The suggested diagnosis system achieved good accuracy as compared to previously proposed methods.
- In this Project we also have developed Web Application to predict HD.

REFERENCES

- A. S. Abdullah and R. R. Rajalaxmi, "A data mining model for predicting the coronary heart disease using random forest classifier," in Proc. Int. Conf. Recent Trends Comput. Methods, Commun. Controls, Apr. 2012, pp. 22–25.
- A. H. Alkeshuosh, M. Z. Moghadam, I. Al Mansoori, and M. Abdar, "Using PSO algorithm for producing best rules in diagnosis of heart disease," in Proc. Int. Conf. Comput. Appl. (ICCA), Sep. 2017, pp. 306– 311.
- N. Al-milli, "Backpropogation neural network for prediction of heart disease," J. Theor. Appl.Inf. Technol., vol. 56, no. 1, pp. 131–135, 2013.
- C. A. Devi, S. P. Rajamhoana, K. Umamaheswari, R. Kiruba, K. Karunya, and R. Deepika, "Analysis of neural networks based heart disease prediction system," in Proc. 11th Int. Conf. Hum. Syst. Interact. (HSI), Gdansk, Poland, Jul. 2018, pp. 233–239.
- P. K. Anooj, "Clinical decision support system: Risk level prediction of heart disease using weighted fuzzy rules," J. King Saud Univ.- Comput. Inf. Sci.,vol.24,no.1,pp.27–40,Jan.2012.doi:10.1016/j.jksuci.2011.09.002.
- L. Baccour, "Amended fused TOPSIS-VIKOR for classification (ATOVIC) applied to some UCI data sets," Expert Syst. Appl., vol. 99, pp. 115–125, Jun. 2018. doi: 10.1016/j.eswa.2018.01.025.
- H. A. Esfahani and M. Ghazanfari, "Cardiovascular disease detection using a new ensemble classifier," in Proc. IEEE 4th Int. Conf. Knowl.Based Eng. Innov. (KBEI), Dec. 2017, pp. 1011–1014.