VIDYA JYOTHI
INSTITUTE OF TECHNOLOGY
AN AUTONOMOUS INSTITUTION

DEPARTMENT OF ARTIFICIAL INTELLIGENCE

Heart Disease Prediction Using ANN

NAME : V. Sriya Chaitanya

ROLL NO. : 21911A3560

SECTION : AI-A

FACULTY SIGN

CONENT

Project Abstract –

Day by day the cases of heart diseases are increasing at a rapid rate and it's very Important and concerning to predict any such diseases beforehand. This diagnosis is a difficult task i.e. it should be performed precisely and efficiently. The research paper mainly focuses on which patient is more likely to have a heart disease based on various medical attributes. We prepared a heart disease prediction system to predict whether the patient is likely to be diagnosed with a heart disease or not using the medical history of the patient. To predict and classify the patient with heart disease we have use ANN models. A quite Helpful approach was used to regulate how the model can be used to improve the accuracy of prediction of Heart Attack in any individual. The strength of the proposed model was quiet satisfying and was able to predict evidence of having a heart disease in a particular individual.

So a quiet significant amount of pressure has been lift off by using the given model in finding the probability of the classifier to correctly and accurately identify the heart disease. The Given heart disease prediction system enhances medical care and reduces the cost. This project gives us significant knowledge that can help us predict the patients with heart disease It is implemented on the

Introduction–

Heart Disease Prediction using ANN. The term "heart disease" is often used interchangeably with the term "cardiovascular disease". Cardiovascular disease generally refers to conditions that involve narrowed or blocked blood vessels that can lead to a heart attack, chest pain (angina) or stroke. Other heart conditions, such as those that affect your Hea's muscle, valves, or rhythm, also are considered forms of heart disease.

Cardiovascular diseases are the number 1 cause of death globally, taking an estimated 17.9 million lives each year, which accounts for 31% of all deaths worldwide. Early detection, and management of cardiovascular diseases can be a great way to manage the fatality rate associated with cardiovascular diseases, and this is where a machine learning model comes in.

We will use the following steps to predict the Heart Disease Understanding the Problem Reading and understanding the data Exploratory Data Analysis and visualisation Modelling Generate Insight ant features

Description On Project–

Individuals with heart disease can face various physical, emotional, and lifestyle challenges. The specific problems experienced may vary depending on the type and severity of the heart condition
Many people across the world are facing heart disease, which causes individual damage.

Heart disease prediction is our point of view in the issue of heart disease and finding a way in reducing life damage. Predicting heart conditions typically involves a combination of risk assessment, medical tests, and clinical evaluation by a healthcare provider

Heart diseases, also known as cardiovascular diseases, can have a wide range of effects on an individual's health and overall well-being. The specific effects experienced can vary depending on the type of heart disease, its severity, and how well it is managed
Hypertension (High Blood Pressure)
Heart Failure
Risk factors for heart disease include smoking, high blood pressure, high cholesterol levels.
Lifestyle modifications such as a healthy diet, regular exercise, smoking cessation, and stress management can help reduce the risk of heart disease.

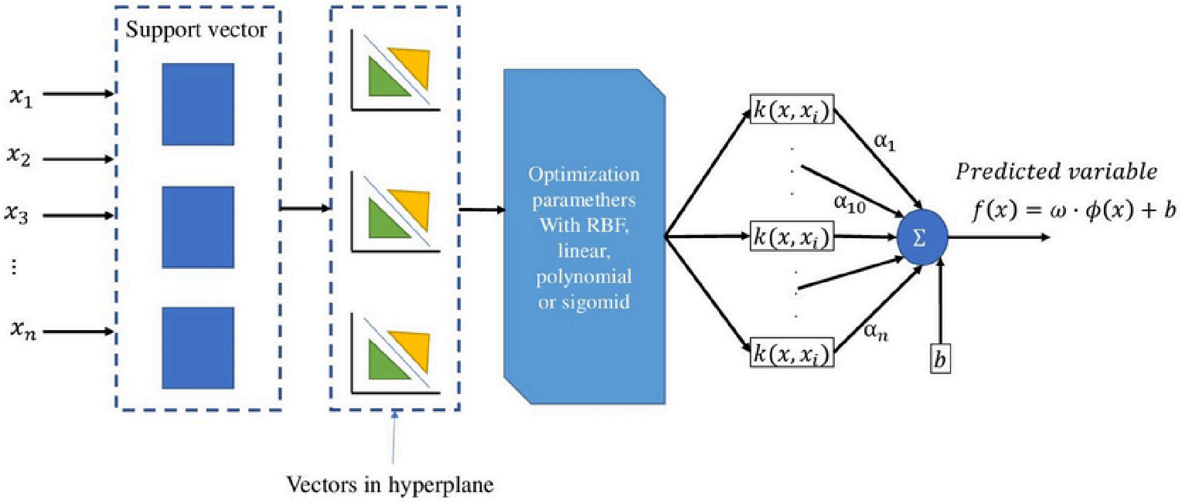
Algorithm and Architecture –

To predict heart disease using Support Vector Machine (SVM), follow these steps:

1. Exploratory Data Analysis (EDA): Use the Heart Disease Cleveland Dataset, focusing on 14 key attributes such as age, sex, and cholesterol.
2. Divide the dataset into training and testing sets.
3. Apply the Support Vector Machine algorithm.
4. Evaluate the model’s accuracy, precision, score.

SVM stands for Support Vector Machine are a set of supervised learning methods used for classification, regression and outliers detection. SVM can be imagined as a surface that maximizes the boundaries between various types of points of data that is represented in multi-dimensional space also known as hyperplane. It can be used for both binary classification and multi-class classification

Hence we have taken SVM algorithm for this project which will allow us to predict if there exists heart disease or not



Causing Factors -

Age:

The risk of heart disease increases with age. As people get older, their arteries may become narrower and less flexible, which can lead to conditions like atherosclerosis.

Sex:

Men are generally at higher risk for heart disease compared to premenopausal women. However, after menopause, a woman's risk increases and becomes like that of men.

High Blood Pressure (Hypertension):

Uncontrolled high blood pressure puts extra strain on the heart and blood vessels, increasing the risk of heart disease. high blood pressure puts extra strain on the heart and blood vessels, increasing the risk of heart disease.

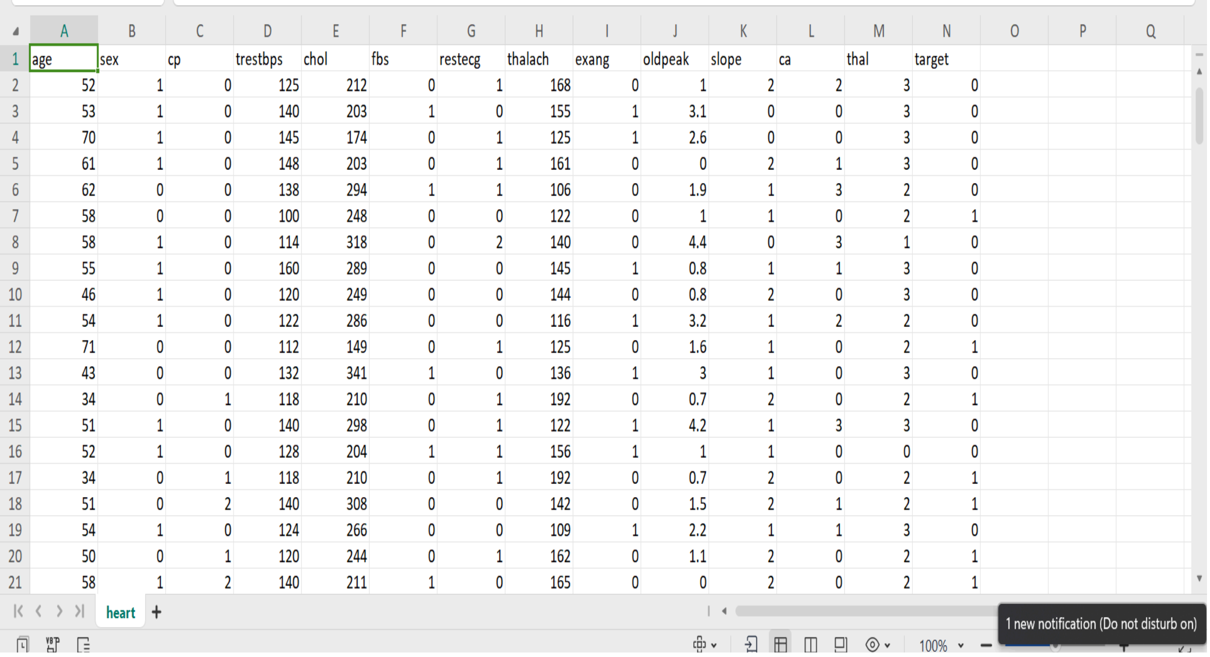
Cholesterol:

Elevated levels of LDL ("bad") cholesterol and low levels of HDL ("good") cholesterol

Can lead to a buildup of plaque in the arteries, increasing the risk.

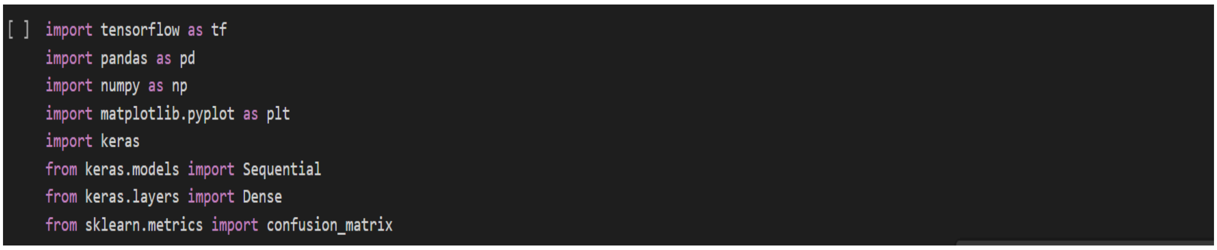
These are few of the factors there are many more that can cause heart failure or disease.

Sample DAta -

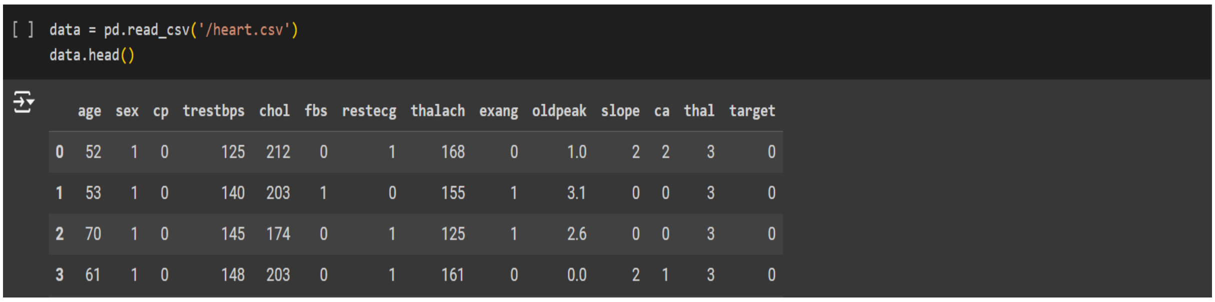


	A	B	C	D	E	F	G	H	I	J	K	L	M	N
1	age	sex	cp	trestbps	chol	fbs	restecg	thalach	exang	oldpeak	slope	ca	thal	target
2	52	1	0	125	212	0	1	168	0	1	2	2	3	0
3	53	1	0	140	203	1	0	155	1	3.1	0	0	3	0
4	70	1	0	145	174	0	1	125	1	2.6	0	0	3	0
5	61	1	0	148	203	0	1	161	0	0	2	1	3	0
6	62	0	0	138	294	1	1	106	0	1.9	1	3	2	0
7	58	0	0	100	248	0	0	122	0	1	1	0	2	1
8	58	1	0	114	318	0	2	140	0	4.4	0	3	1	0
9	55	1	0	160	289	0	0	145	1	0.8	1	1	3	0
10	46	1	0	120	249	0	0	144	0	0.8	2	0	3	0
11	54	1	0	122	286	0	0	116	1	3.2	1	2	2	0
12	71	0	0	112	149	0	1	125	0	1.6	1	0	2	1
13	43	0	0	132	341	1	0	136	1	3	1	0	3	0
14	34	0	1	118	210	0	1	192	0	0.7	2	0	2	1
15	51	1	0	140	298	0	1	122	1	4.2	1	3	3	0
16	52	1	0	128	204	1	1	156	1	1	1	0	0	0
17	34	0	1	118	210	0	1	192	0	0.7	2	0	2	1
18	51	0	2	140	308	0	0	142	0	1.5	2	1	2	1
19	54	1	0	124	266	0	0	109	1	2.2	1	1	3	0
20	50	0	1	120	244	0	1	162	0	1.1	2	0	2	1
21	58	1	2	140	211	1	0	165	0	0	2	0	2	1

Code And Output For The Project -

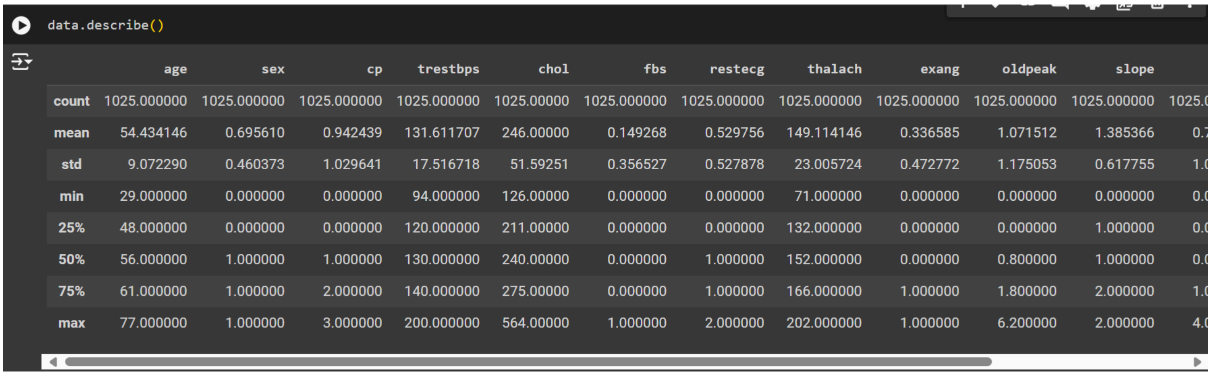


```
[ ] import tensorflow as tf
    import pandas as pd
    import numpy as np
    import matplotlib.pyplot as plt
    import keras
    from keras.models import Sequential
    from keras.layers import Dense
    from sklearn.metrics import confusion_matrix
```



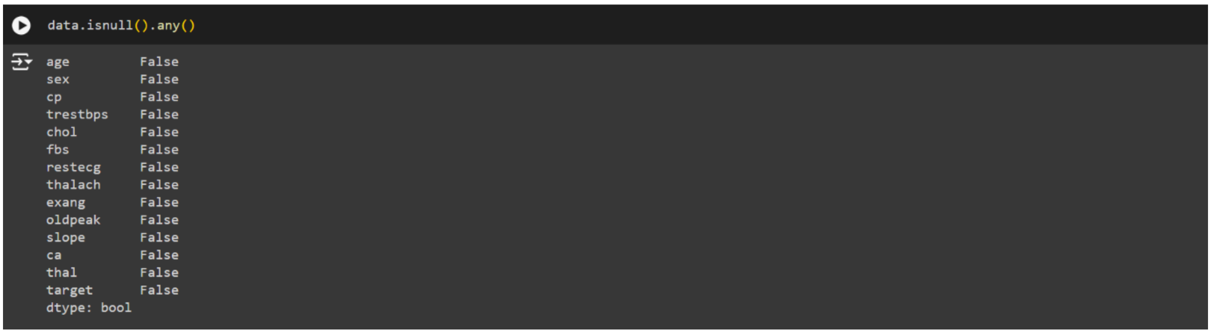
```
[ ] data = pd.read_csv('/heart.csv')
    data.head()
```

```
   age  sex  cp  trestbps  chol  fbs  restecg  thalach  exang  oldpeak  slope  ca  thal  target
0   52    1   0       125   212    0        1      168      0      1.0      2   2     3       0
1   53    1   0       140   203    1        0      155      1      3.1      0   0     3       0
2   70    1   0       145   174    0        1      125      1      2.6      0   0     3       0
3   61    1   0       148   203    0        1      161      0      0.0      2   1     3       0
```



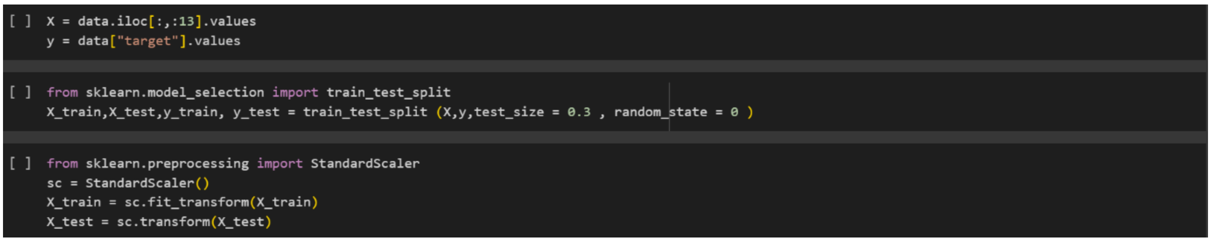
```
data.describe()
```

	age	sex	cp	trestbps	chol	fbs	restecg	thalach	exang	oldpeak	slope	
count	1025.000000	1025.000000	1025.000000	1025.000000	1025.00000	1025.000000	1025.000000	1025.000000	1025.000000	1025.000000	1025.000000	1025.0
mean	54.434146	0.695610	0.942439	131.611707	246.00000	0.149268	0.529756	149.114146	0.336585	1.071512	1.385366	0.7
std	9.072290	0.460373	1.029641	17.516718	51.59251	0.356527	0.527878	23.005724	0.472772	1.175053	0.617755	1.0
min	29.000000	0.000000	0.000000	94.000000	126.00000	0.000000	0.000000	71.000000	0.000000	0.000000	0.000000	0.0
25%	48.000000	0.000000	0.000000	120.000000	211.00000	0.000000	0.000000	132.000000	0.000000	0.000000	1.000000	0.0
50%	56.000000	1.000000	1.000000	130.000000	240.00000	0.000000	1.000000	152.000000	0.000000	0.800000	1.000000	0.0
75%	61.000000	1.000000	2.000000	140.000000	275.00000	0.000000	1.000000	166.000000	1.000000	1.800000	2.000000	1.0
max	77.000000	1.000000	3.000000	200.000000	564.00000	1.000000	2.000000	202.000000	1.000000	6.200000	2.000000	4.0



```
data.isnull().any()
```

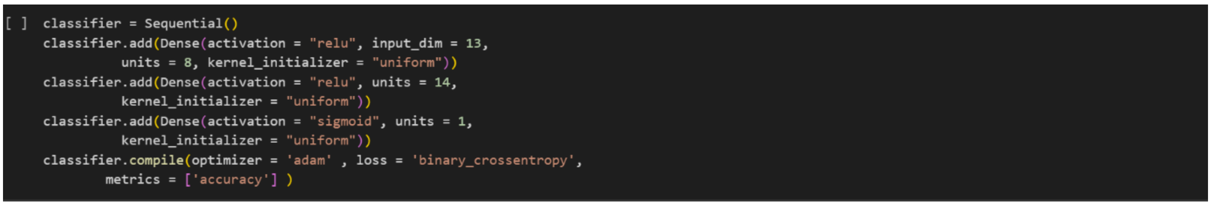
```
age         False
sex         False
cp          False
trestbps    False
chol        False
fbs         False
restecg     False
thalach     False
exang       False
oldpeak     False
slope       False
ca          False
thal        False
target      False
dtype: bool
```



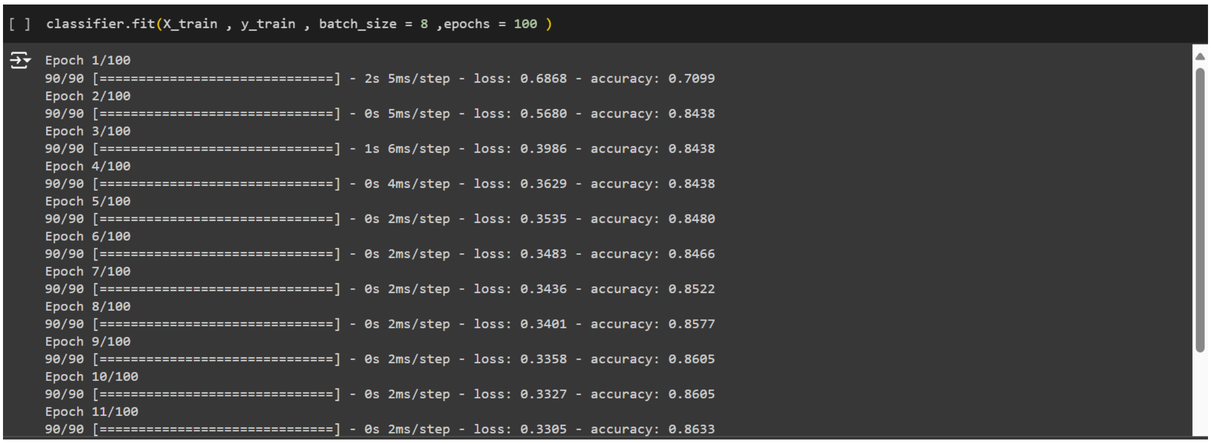
```
X = data.iloc[:,:13].values
y = data["target"].values
```

```
from sklearn.model_selection import train_test_split
X_train,X_test,y_train, y_test = train_test_split (X,y,test_size = 0.3 , random_state = 0 )
```

```
from sklearn.preprocessing import StandardScaler
sc = StandardScaler()
X_train = sc.fit_transform(X_train)
X_test = sc.transform(X_test)
```

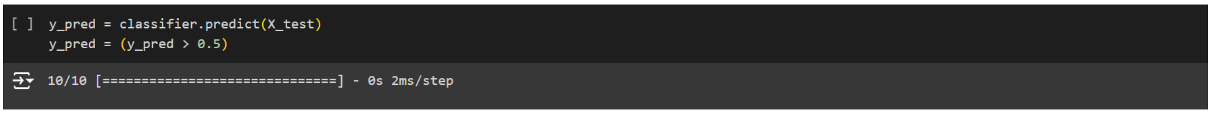


```
classifier = Sequential()
classifier.add(Dense(activation = "relu", input_dim = 13,
          units = 8, kernel_initializer = "uniform"))
classifier.add(Dense(activation = "relu", units = 14,
          kernel_initializer = "uniform"))
classifier.add(Dense(activation = "sigmoid", units = 1,
          kernel_initializer = "uniform"))
classifier.compile(optimizer = 'adam' , loss = 'binary_crossentropy',
        metrics = ['accuracy'] )
```



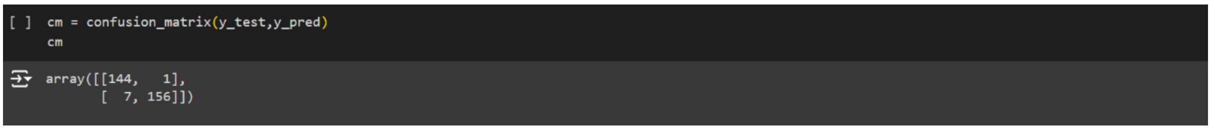
```
classifier.fit(X_train , y_train , batch_size = 8 ,epochs = 100 )
```

```
Epoch 1/100
90/90 [==============================] - 2s 5ms/step - loss: 0.6868 - accuracy: 0.7099
Epoch 2/100
90/90 [==============================] - 0s 5ms/step - loss: 0.5680 - accuracy: 0.8438
Epoch 3/100
90/90 [==============================] - 1s 6ms/step - loss: 0.3986 - accuracy: 0.8438
Epoch 4/100
90/90 [==============================] - 0s 4ms/step - loss: 0.3629 - accuracy: 0.8438
Epoch 5/100
90/90 [==============================] - 0s 2ms/step - loss: 0.3535 - accuracy: 0.8480
Epoch 6/100
90/90 [==============================] - 0s 2ms/step - loss: 0.3483 - accuracy: 0.8466
Epoch 7/100
90/90 [==============================] - 0s 2ms/step - loss: 0.3436 - accuracy: 0.8522
Epoch 8/100
90/90 [==============================] - 0s 2ms/step - loss: 0.3401 - accuracy: 0.8577
Epoch 9/100
90/90 [==============================] - 0s 2ms/step - loss: 0.3358 - accuracy: 0.8605
Epoch 10/100
90/90 [==============================] - 0s 2ms/step - loss: 0.3327 - accuracy: 0.8605
Epoch 11/100
90/90 [==============================] - 0s 2ms/step - loss: 0.3305 - accuracy: 0.8633
```



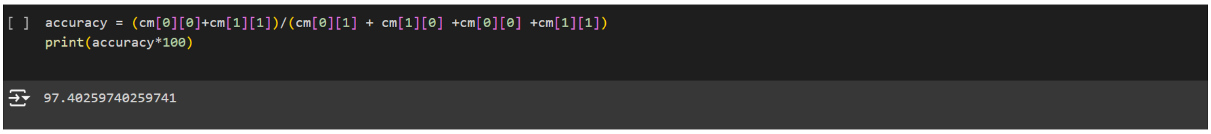
```
y_pred = classifier.predict(X_test)
y_pred = (y_pred > 0.5)
```

```
10/10 [==============================] - 0s 2ms/step
```



```
cm = confusion_matrix(y_test,y_pred)
cm
```

```
array([[144,   1],
       [  7, 156]])
```



```
accuracy = (cm[0][0]+cm[1][1])/(cm[0][1] + cm[1][0] +cm[0][0] +cm[1][1])
print(accuracy*100)
```

```
97.40259740259741
```

Conclusion -

A cardiovascular disease detection model has been developed using three ANN classification modelling techniques. This project predicts people with cardiovascular disease by extracting the patient medical history that leads to a fatal heart disease from a dataset that includes patients' medical history such as chest pain, sugar level, blood pressure, etc. This Heart Disease detection system assists a patient based on his/her clinical information of them been diagnosed with a previous heart disease. The accuracy of our model is 97.4%.

Use of more training data ensures the higher chances of the model to accurately predict whether the given person has a heart disease or not. By using these, computer aided techniques we can predict the patient fast and better and the cost can be reduced very much.

There are several medical databases that we can work on as these techniques are better and they can predict better than a human being which helps the patient as well as the doctors. Therefore, in conclusion this project helps us predict the patients who are diagnosed with heart diseases by cleaning the dataset and applying these algorithm to get an accuracy of an average of 97.4% on our model which is better than other models having less accuracy.