PROJECT REPORT ON:

“HEART FAILURE PREDICTION”

अखिल भारतीय रोबोटिक्स एवं ऑटोमेशन परिषद

AICRA

All India Council for Robotics & Automation

Submitted By:

Student Name: Ophelia

Submitted Date By: 15-10-2021

UNDERTAKING

I declare that the work presented in this project titled “**HEART FAILURE PREDICTION**”, submitted to the All India council of robotics and Automation, for the award of the ***Internship*** in **DATA SCIENCE**, is my original work. I have not plagiarized or submitted the same work for the award of any other Internship. In case this undertaking is found incorrect, I accept that my Project may be unconditionally withdrawn.

October, 2021 ______________________

OPHELIA

CERTIFICATE

Certified that the work contained in the project titled “**HEART FAILURE PREDICTION**”, by Ophelia, has been carried out under my supervision

and that this work has not been submitted elsewhere for a Internship..

All India Council of Robotics and Automation
DATA SCIENCE
Delhi-110020

Preface

Heart failure (HF), often referred to as congestive heart failure (CHF), occurs when the heart is unable to pump sufficiently to maintain blood flow to meet the body's needs. The terms chronic heart failure (CHF) or congestive cardiac failure (CCF) are often used interchangeably with congestive heart failure. Signs and symptoms commonly include shortness of breath, excessive tiredness, and leg swelling. The shortness of breath is usually worse with exercise, while lying down, and may wake the person at night.A limited ability to exercise is also a common feature.

Common causes of heart failure include coronary artery disease including a previous myocardial infarction (heart attack), high blood pressure, atrial fibrillation, valvular heart disease, excess alcohol use, infection, and cardiomyopathy of an unknown cause. These cause heart failure by changing either the structure or the functioning of the heart. here are two main types of heart failure: heart failure due to left ventricular dysfunction and heart failure with normal ejection fraction depending on if the ability of the left ventricle to contract is affected, or the heart's ability to relax.

Acknowledgements

I take upon this opportunity to acknowledge the many people who helped me to accomplish this project successfully.

I am deeply indebted to my mentor Sumit Chatterjee who motivated me along the way.

I would like to thank all my teachers to support me throught the completion of project.

My heartfelt thanks to my parents who support me a lot.

I also express my deepest gratitude to the almighty God.

Finally, I would like to wind up by paying my heartfelt thanks to AICRA institute who provided me with this great opportunity.

OPHELIA

TABLE OF CONTENT

1. INTRODUCTION

1.1 PROBLEM DEFINATION

The major challenge in heart disease is its detection. There are instruments available which can predict heart disease but either they are expensive or are not efficient to calculate chance of heart disease in human. Early detection can decrease the mortality rate. However is not possible to monitor any person 24 hours but we can collect data about them , lot of data is available in today's world that can be used to predict their health and notify about it to them in advance.

1.2 OBJECTIVES

The main objectives of developing this project are:

1. To develop machine learning model to predict future possibility of heart disease by implementing Logistic Regression.
2. To determine significant risk based on medical dataset which may lead to heart disease.
3. To analyze feature selection methods and understand their working principle.

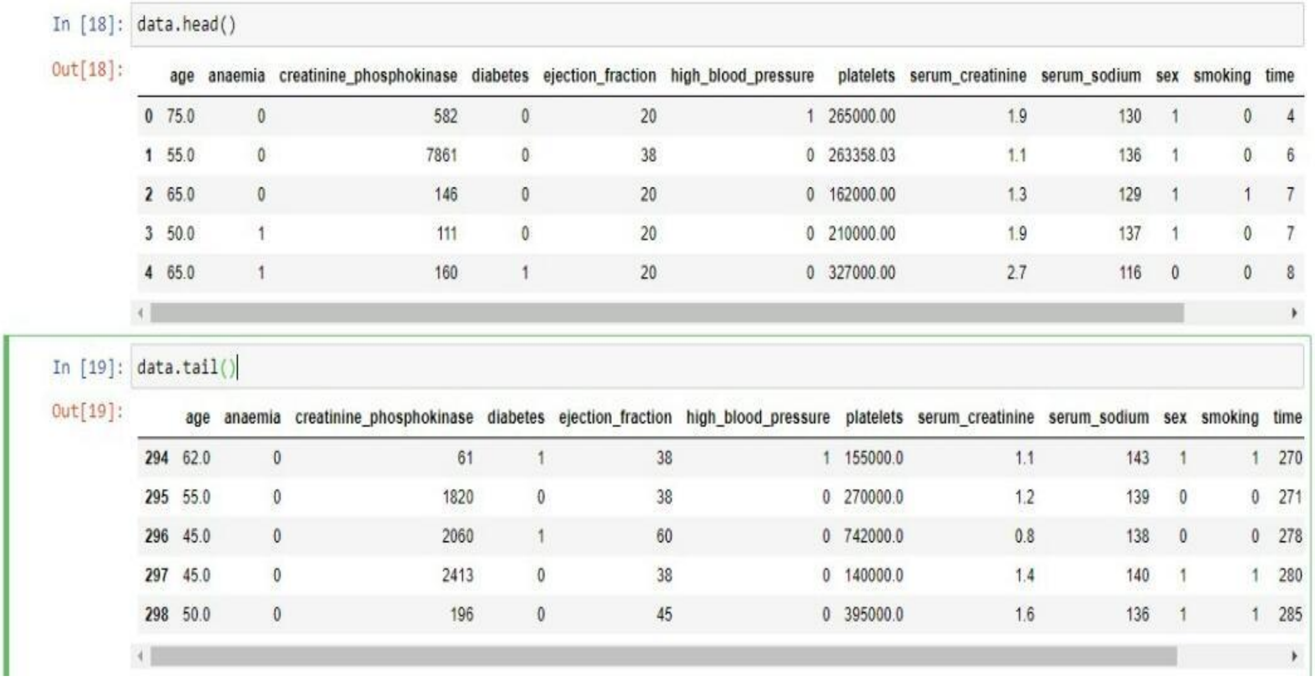
2. DATASET

The dataset is publicly on the kaggle website which is result of ongoing study. Cardiovascular disease are the number 1 cause of death globally, taking an estimate 17.9 million lives each year, which accounts for 31% of all deaths worldwide.

Heart failure is a common event caused by CVDs and the dataset contains 12 features that can be used to predict mortality by heart failure.

Most cardiovascular disease can be prevented by addressing behavioral risk factors such as tobacco use, unhealthy diet and obesity, physical inactivity and harmful use of alcohol using population wide strategies.

People with cardiovascular disease or who are at high cardiovascular risk (due to the presence of one or more risk factors such as hypertension , diabetes, hyperlipidaemia or already established disease) need early detection and management wherein a machine learning model can be of great help.



```
In [18]: data.head()
```

Out[18]:

	age	anaemia	creatinine_phosphokinase	diabetes	ejection_fraction	high_blood_pressure	platelets	serum_creatinine	serum_sodium	sex	smoking	time
0	75.0	0	582	0	20	1	265000.00	1.9	130	1	0	4
1	55.0	0	7861	0	38	0	263358.03	1.1	136	1	0	6
2	65.0	0	146	0	20	0	162000.00	1.3	129	1	1	7
3	50.0	1	111	0	20	0	210000.00	1.9	137	1	0	7
4	65.0	1	160	1	20	0	327000.00	2.7	116	0	0	8

```
In [19]: data.tail()
```

Out[19]:

	age	anaemia	creatinine_phosphokinase	diabetes	ejection_fraction	high_blood_pressure	platelets	serum_creatinine	serum_sodium	sex	smoking	time
294	62.0	0	61	1	38	1	155000.0	1.1	143	1	1	270
295	55.0	0	1820	0	38	0	270000.0	1.2	139	0	0	271
296	45.0	0	2060	1	60	0	742000.0	0.8	138	0	0	278
297	45.0	0	2413	0	38	0	140000.0	1.4	140	1	1	280
298	50.0	0	196	0	45	0	395000.0	1.6	136	1	1	285

3. ALGORITHM USED

3.1 LOGISTIC REGRESSION

Logistic Regression is a supervised classification algorithm. It is predictine analysis algorithm based on the concept of probability. It measures the relstionshio between the dependent variable and the one or more independent variable(risk factor) by estimating probabilities using underlying logistic function (sigmoid function) . sigmoid function is used as accost function to limit the hypothesis of logistic regression between 0 and 1(squashing).

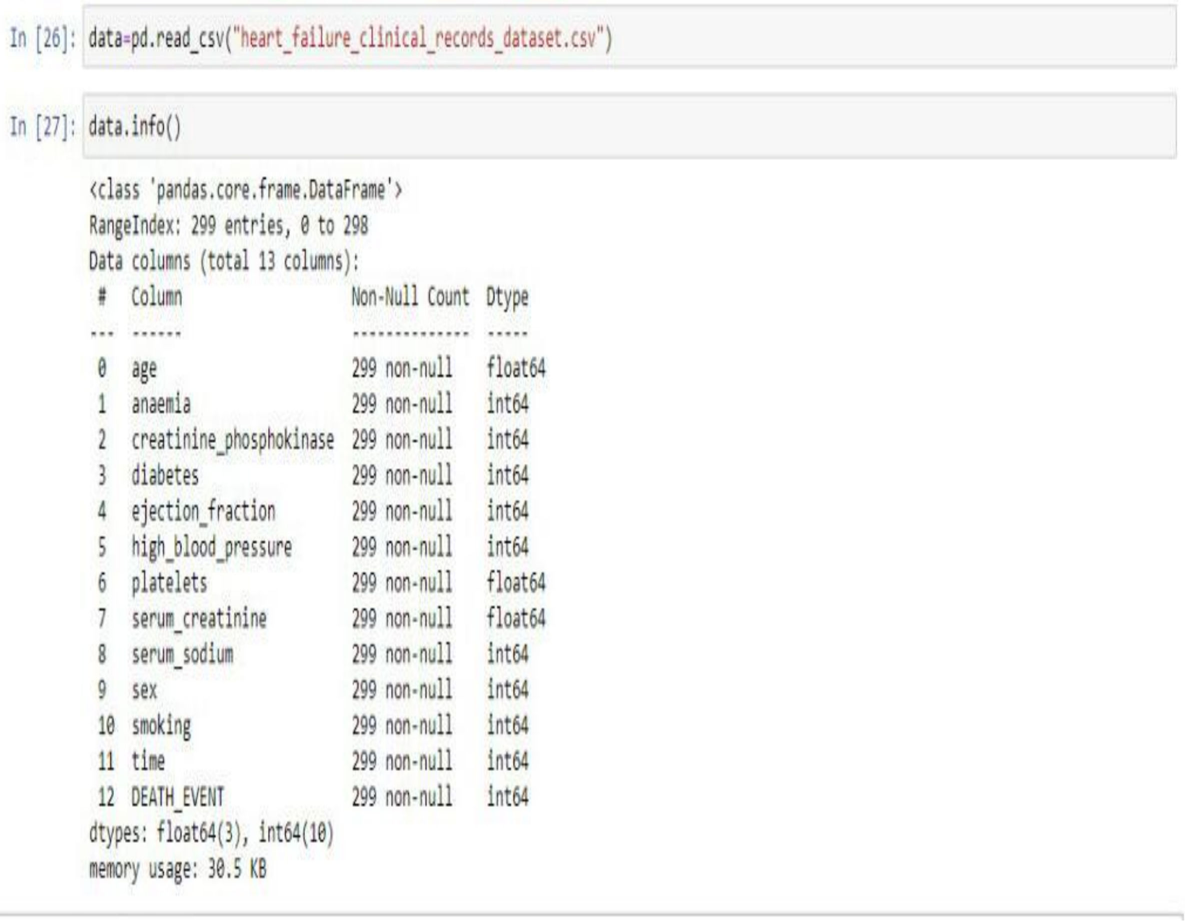
Logistic Regression relies on the proper presentation of data. So, to make the model more powerful, important features from the available data set are selected using Backward elimination and recursive elimination technique.

4. BUILDING PREDICTIVE MODEL

4.1 DATA EXPLORATION

To get the insight of data and to reduce it (so that only required data can be used and useless data can be eliminated)

Proper exploration of data is essential. Many functions can be used to understand data few of them used in these project are :



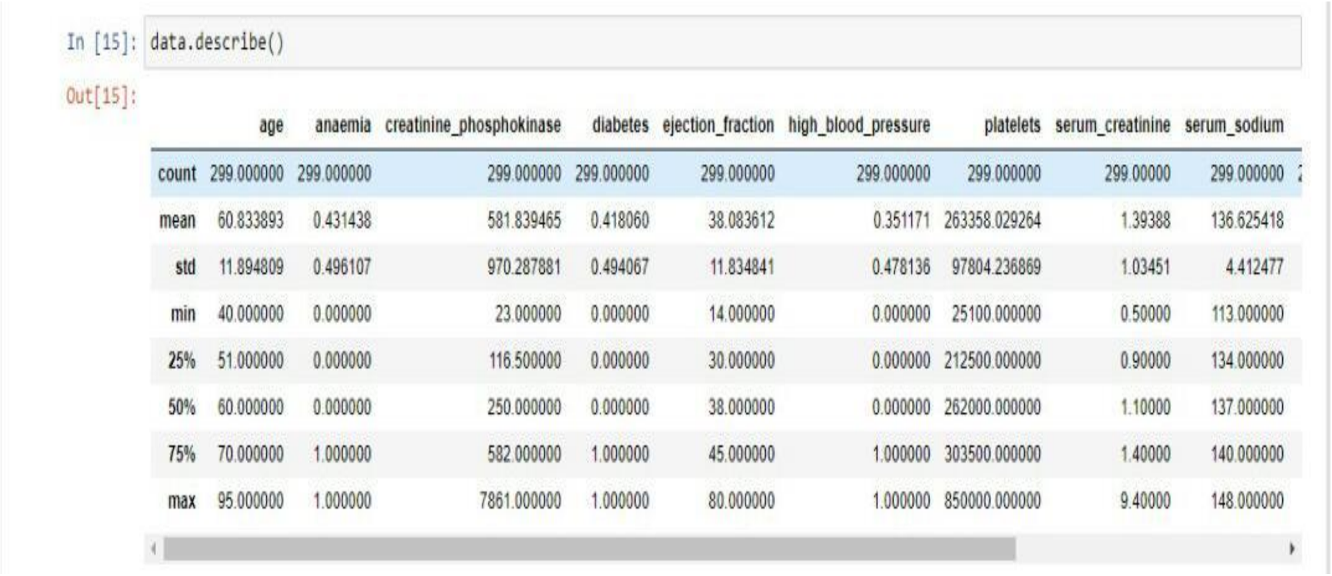
```
In [26]: data=pd.read_csv("heart_failure_clinical_records_dataset.csv")
```

```
In [27]: data.info()
```

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

Data.info() can be used to know number of non null entries, coloumns , what type of data does the data set contains and amount of memory used.

Data.describe() is used to view some basic statistical details like percentile, mean, std etc of a data frame or series of numeric values. It analyzes both numeric and object series and the DataFrame column sets of mixed data types.



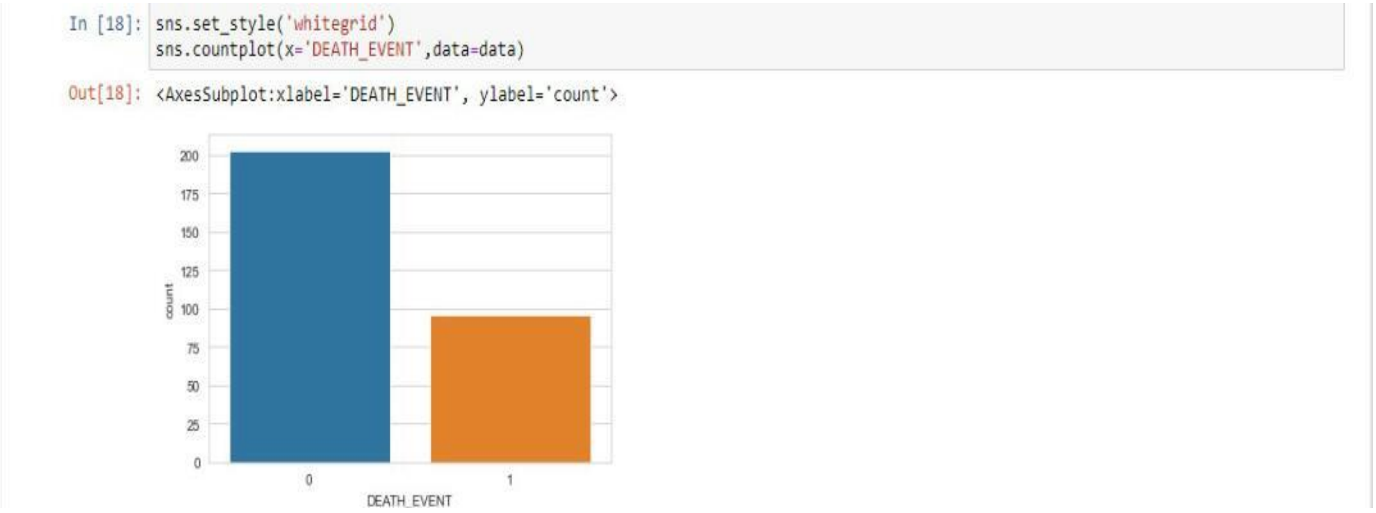
In [15]:
```
data.describe()
```

Out[15]:

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

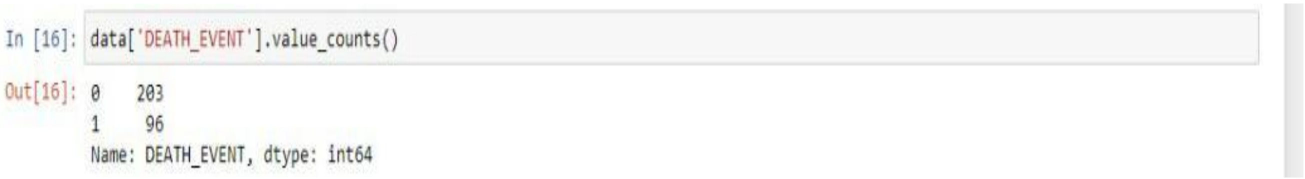
To get the true count of number of deaths due to heart failure we can count it with values_count() where in 0 represents death and 1 represents survival .

For better visualization baar graph can also be used to represent the counts and also to check if the data is balanced or not.



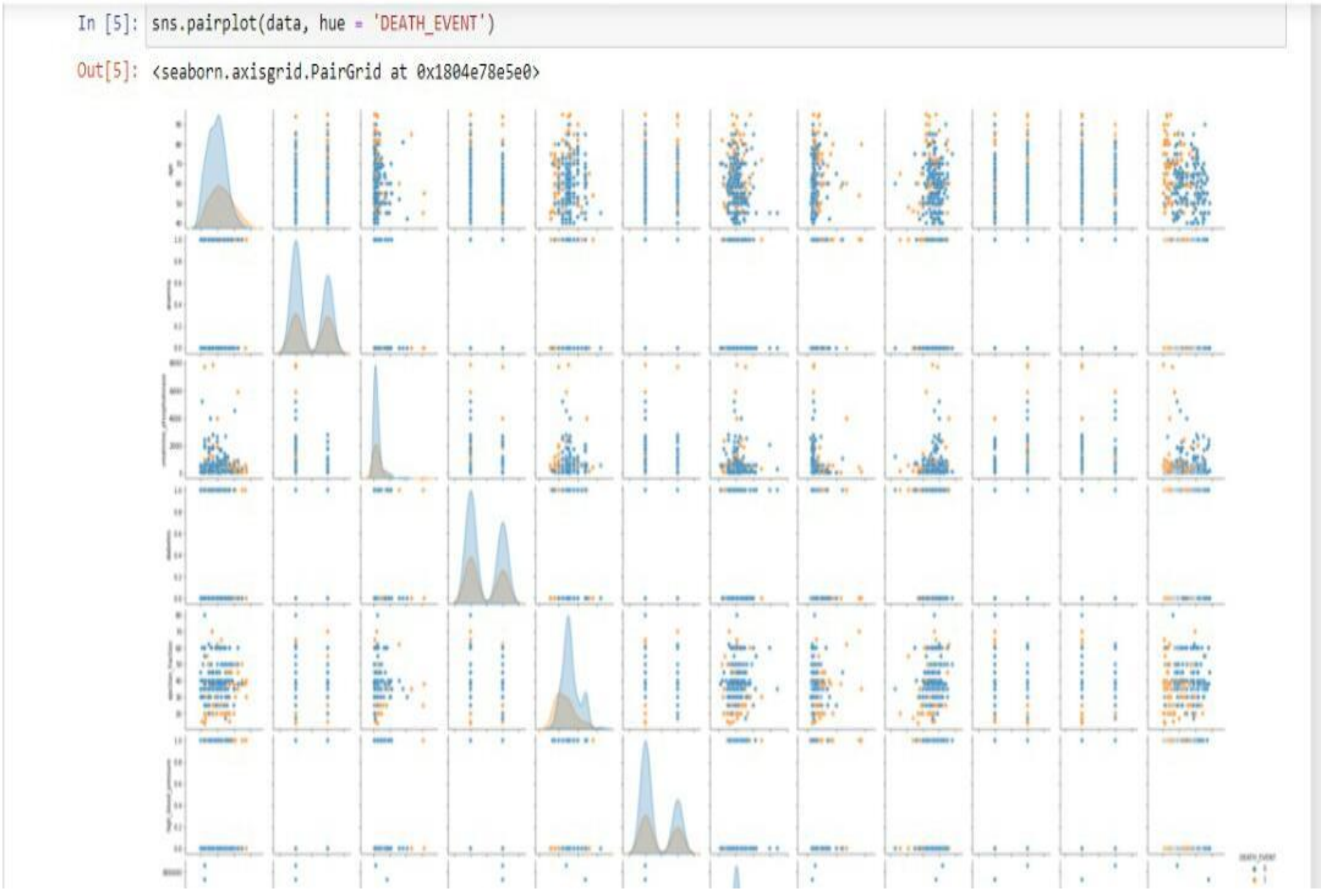
In [18]:
```
sns.set_style('whitegrid')
sns.countplot(x='DEATH_EVENT',data=data)
```

Out[18]:
```
<AxesSubplot:xlabel='DEATH_EVENT', ylabel='count'>
```



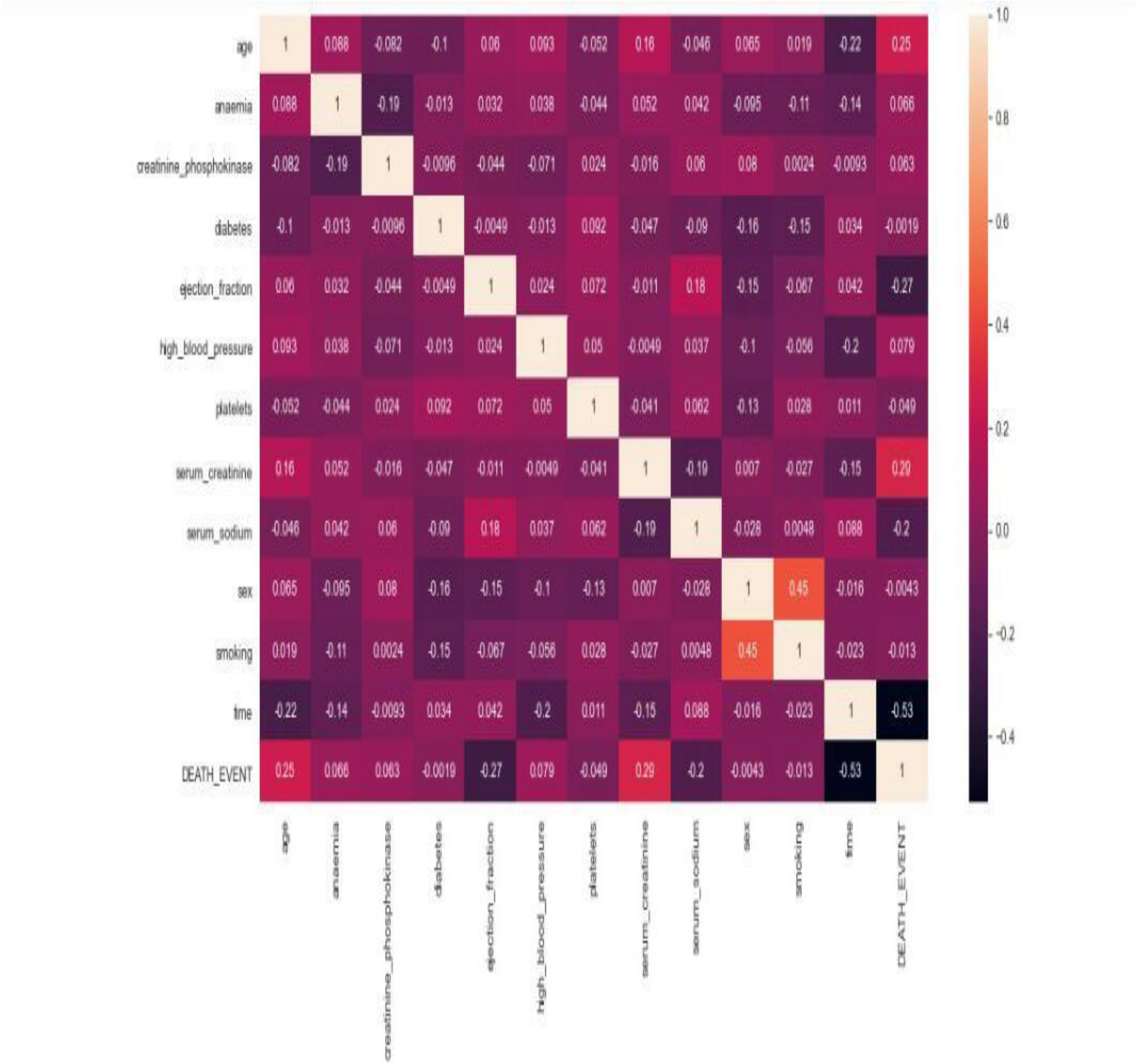
```
In [16]: data['DEATH_EVENT'].value_counts()

Out[16]: 0    203
         1     96
         Name: DEATH_EVENT, dtype: int64
```

A pair plot allows us to see both distribution of single variables and relationships between two variables.pair plots are a great method to identify trends for follow-up analysis and, are easy tp implement in python. Just we need to import seaborn to implement it.

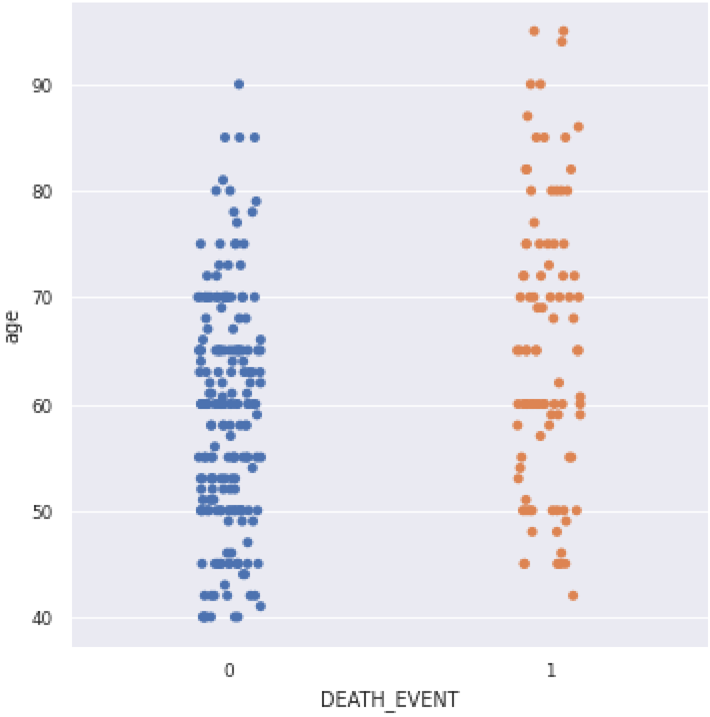


```
In [5]: sns.pairplot(data, hue = 'DEATH_EVENT')

Out[5]: <seaborn.axisgrid.PairGrid at 0x1804e78e5e0>
```

A heatmap is a two dimensional graphical representation of data where the individual values that are contained in a matrix are represented as colors. The seaborn python package allows the creation of annotated heatmaps which can be tweaked using matplotlib tools.



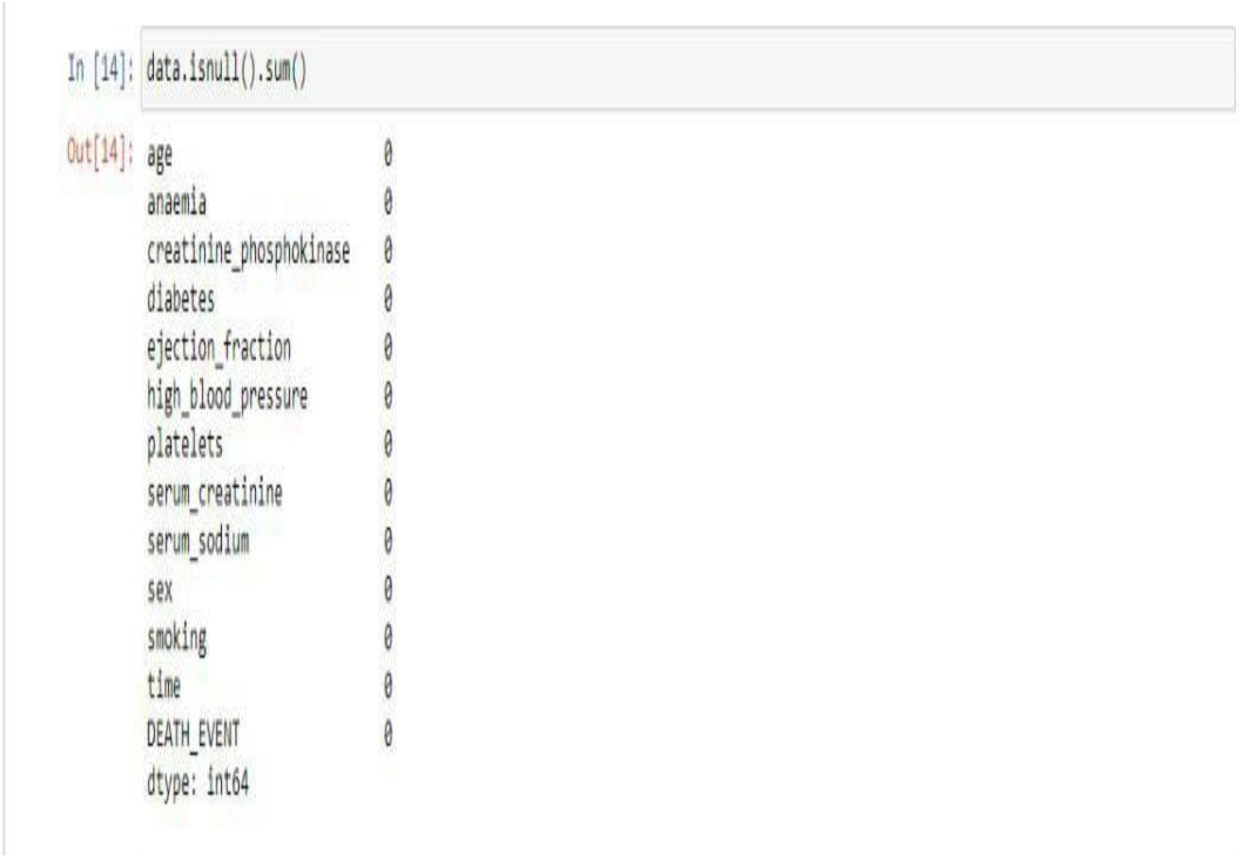
Individually every independent variable can be compared with dependent variable to get understanding of their proportions and use them to build efficient predictive model.



4.2 DATA WRANGLING

Data wrangling is the process of cleansing and unifying messy and complex data sets for easy access and analysis.

If there are any null elements that can be eliminated for simplification . to know if there are any null values isnull() function can be used .



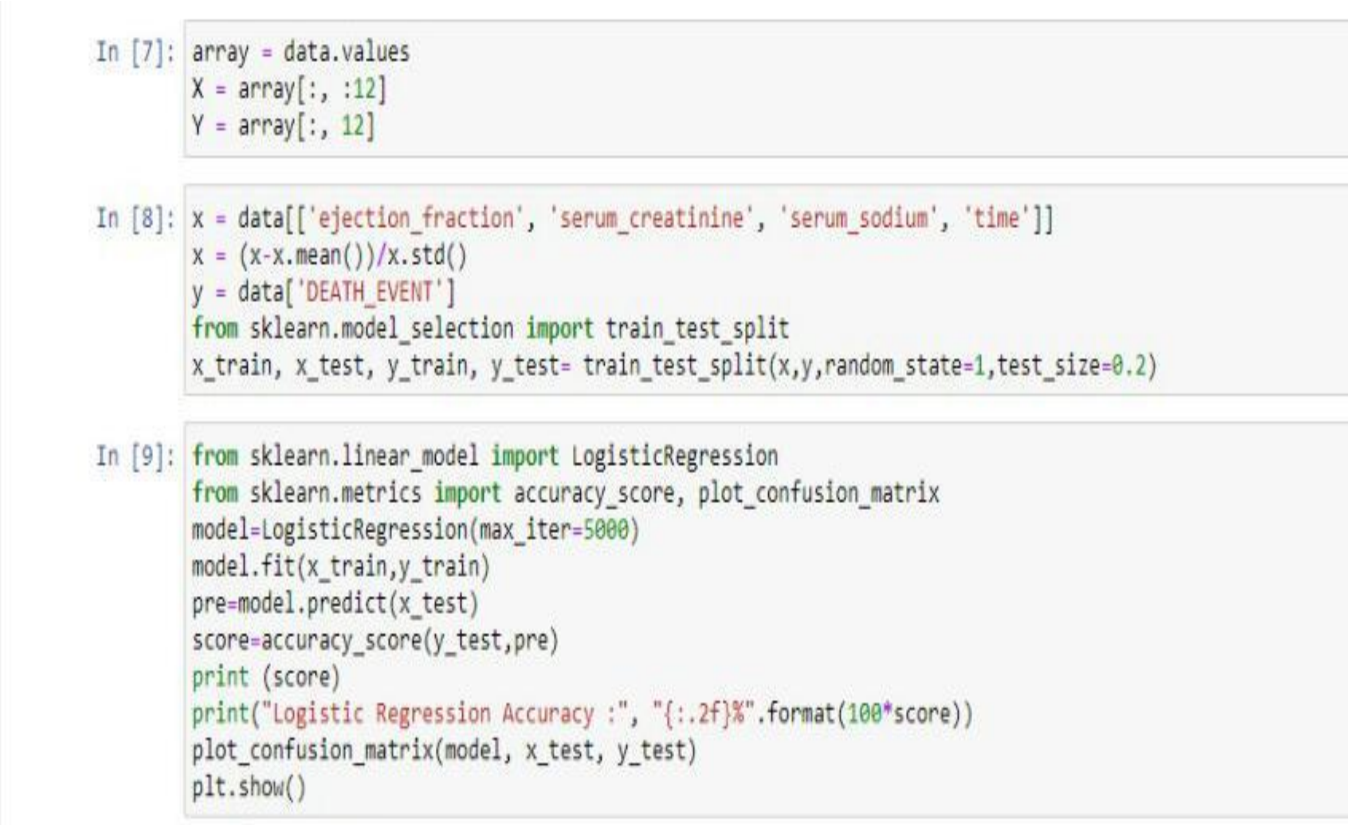
```
In [14]: data.isnull().sum()

Out[14]: age                         0
         anaemia                     0
         creatinine_phosphokinase    0
         diabetes                    0
         ejection_fraction           0
         high_blood_pressure         0
         platelets                   0
         serum_creatinine            0
         serum_sodium                0
         sex                         0
         smoking                     0
         time                        0
         DEATH_EVENT                 0
         dtype: int64
```

Since there are no null values ,data can be left unchanged ,in large data sets scaling is used to remove unwanted data .

4.3 TRAIN AND TEST

In LogisticRegression data is split into x and y variable where x contains all independent variable and y contains dependent variable whose prediction is to be done.

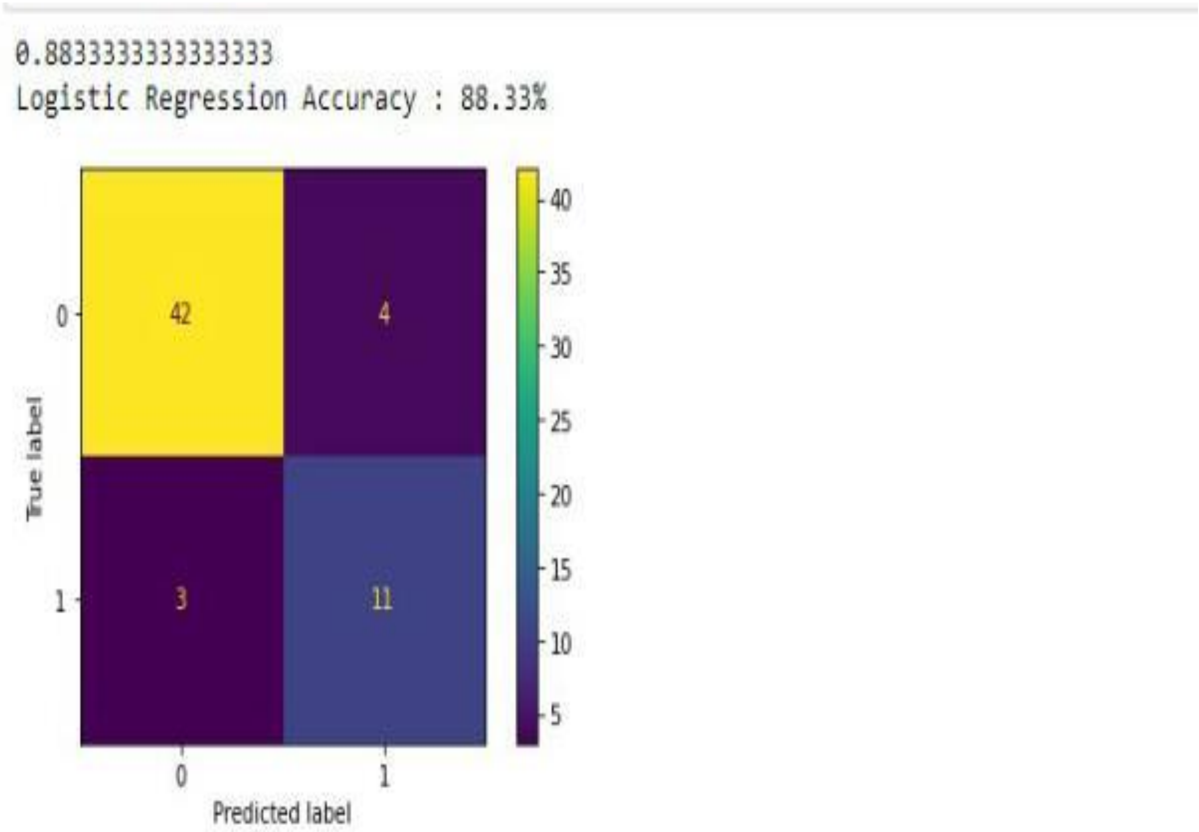
First the data is trained with the data set so that it gets familiar with the data and can predict easily for new data set.



```
In [7]: array = data.values
        X = array[:, :12]
        Y = array[:, 12]

In [8]: x = data[['ejection_fraction', 'serum_creatinine', 'serum_sodium', 'time']]
        x = (x-x.mean())/x.std()
        y = data['DEATH_EVENT']
        from sklearn.model_selection import train_test_split
        x_train, x_test, y_train, y_test= train_test_split(x,y,random_state=1,test_size=0.2)

In [9]: from sklearn.linear_model import LogisticRegression
        from sklearn.metrics import accuracy_score, plot_confusion_matrix
        model=LogisticRegression(max_iter=5000)
        model.fit(x_train,y_train)
        pre=model.predict(x_test)
        score=accuracy_score(y_test,pre)
        print (score)
        print("Logistic Regression Accuracy :", "{:.2f}%".format(100*score))
        plot_confusion_matrix(model, x_test, y_test)
        plt.show()
```

4.4 ACCURACY CHECK

With the accuracy score of trained model we can know how efficient and how much reliable it can be with new dataset.



```
0.8833333333333333
Logistic Regression Accuracy : 88.33%
```

5. DISCUSSION ON RESULTS

In this project by using machine learning Algorithm to detect heart failure chance, based on a dataset. With this model patient can be treated in time based on the prediction.

MODEL	**ACCURACY**
LOGISTICREGRESSION	88.33%

6. REFERENCES

- https://www.kaggle.com/andrewmvd/heart-failure-clinical-data
- https://in.video.search.yahoo.com/search/video;_ylt=AwrPg3UpAmhhcyUAV0e7HAx.;_ylu=Y29sbwNzZzMEcG9zAzEEdnRpZAMEc2VjA3Nj?p=heart+failure+prediction+in+python+youtube&type=E211IN714G0&ei=UTF
- https://www.academia.edu/42249626/Mini_Project_Report_On_Heart_Disease_Prediction_
- https://seaborn.pydata.org/generated/seaborn.pairplot.html
- https://seaborn.pydata.org/generated/seaborn.heatmap.html