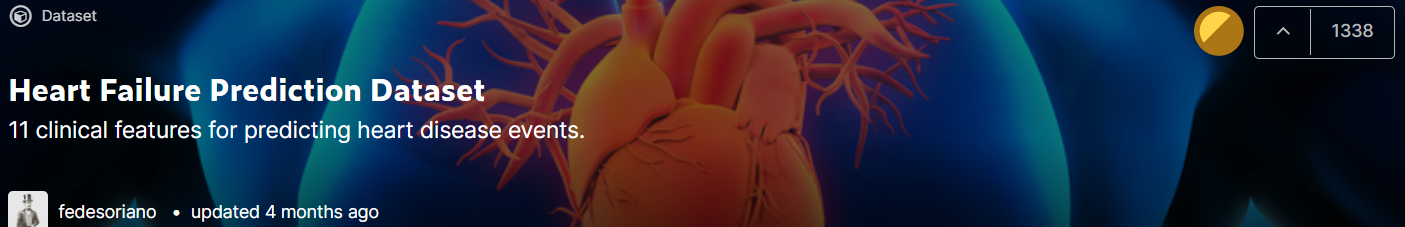
**Heart Failure Diseases Prediction**



[Heart Failure Prediction Dataset | Kaggle](https://www.kaggle.com/fedesoriano/heart-failure-prediction/code)

Our team names:

ندي ايمن عبدالعزيز حسن نصر يمني عبدالمنعم حسني الغنام

عبير محمد فتحي المرشدي ندي ناصر بدوى احمد

In this project:

we want to predict whether the patient has heart failure or not.

Using some of the characteristics we took from a patient survey.

918 questionnaires were collected from patients and a predictor was made and used to detect this disease.

After applying this model, disease was predicted by 86% Accuracy.

Below is a detailed view of the steps we have taken:

**Steps**

1. Importing libraries
2. Import the dataset
3. Knowing information about dataset
4. Check for missing values
5. Cleaning dataset

* Check for missing values(what we just need)

1. See the categorical values
2. Visualizing dataset
3. Extracting Independent & dependent Variable
4. Encode the Categorical Variable
5. Using Feature selection technique [RFE]
6. Splitting the data-set (Training & Test Set)
7. Feature Scaling
8. Training Decision Tree Classifier
9. Comparing Results
10. Checking for Accuracy
11. Checking for overfitting
12. Importing libraries

We use these libraries:



1. Import the dataset

-We import our dataset which has 918 rows & 11 features.



3-Knowing information about dataset

-In this step, we know our features and their names, we also know which feature is important and which is not (for example, Age and sex is not very important )

-in this step, we also know our type of values (float, int , object[we must fix it in this project])

Graphical user interface, text, application

Description automatically generated

4-Check for missing values

-In this step, we are looking for if there are missing, error or non-existent values.

Graphical user interface, application, Word

Description automatically generated

5-Cleaning dataset

-In this step , we just need to drop rows (3 rows) which have missing values to clean our dataset.

Graphical user interface, application, Word

Description automatically generated

6-See the categorical values

-In this step, By using [.head()] method we can easily know our categorical values to can modify it later.

A picture containing graphical user interface

Description automatically generated

7-Visualizing dataset

-In this step , we are trying to show ranges of values for every features in our datasets (Not equal )

Chart, histogram

Description automatically generated

-This plot shows The number of those who were classified as sick(1) and the number of healthy ones(0).

Chart, bar chart

Description automatically generated

8-Extracting Independent & dependent Variable

-In this step, we select our independent variable (all columns except the last one),and dependent one (last column).



9-Encode the Categorical Variable

-In this step, we modify our values of column [1,2,6,8,10],to turn them to numeric variables to can our model deal with them (we cannot do that if they were categoric ones ).

Graphical user interface, text, application, email

Description automatically generated

-we know show our new features, we notice that the 11 features turn into 20 features and their values are not equal (have not equal ranges of features)

A picture containing text, crossword puzzle

Description automatically generated

Chart, histogram

Description automatically generated

10-Using Feature selection technique [RFE]

-In this step, we are trying to reduce our features from 20 to 5 features by identify which feature is more important rather than the others , we do this by Recursive Feature Elimination Technique.

Graphical user interface, text, application, email

Description automatically generated

-we can now see that our features reduced from 20 to just the most important 5 features which affect more in our prediction .

A picture containing text, crossword puzzle

Description automatically generated

11-Splitting the dataset (Training & Test Set)

-In this step, we split our dataset to training and test dataset , as we training the model on (80%)of data and (20%) we test our model in them.

Graphical user interface, text, application, Word

Description automatically generated

12-Feature Scaling

-In this step, as we saw our ranges of values need to be equal to make our model efficient to predict well.

Graphical user interface, text, application

Description automatically generated

13-Training Decision Tree Classifier

-In this step, we finally finish preprocessing our dataset and ready to start training our model

We choose Decision tree as it is a common-sense technique to find the best solutions to problems with uncertainty .The time & effort required by an analyst to develop and tune a classifier can be reduced , Fast prediction, Ease of coding and good to classify our dataset ..

So we import our decision tree classifier to train our model.

Graphical user interface, text, application

Description automatically generated

14-Comparing Results

-In this step, we just show our actual results and predicted ones.

A computer screen capture

Description automatically generated with medium confidence

15-Checking for Accuracy

-In this step, we can see that there are a lot of ways by which we can know the accuracy , we found it 86% , (without feature selection step we found it 78% ) , so it is more accurate

A picture containing text

Description automatically generated

Chart

Description automatically generated

16-Checking for overfitting

As we know in Decision Tree Classifier , we may be have overfitting (it is disadvantage ) , so …

-In this step, we use Lasso model to can check if we have overfitting or not and fix it too.

Graphical user interface, text, application, email

Description automatically generated

-Fortunately, Our model is predict well in train and test dataset , so we can celebrate ….

ENDING OUR PROJECT