SUPERVISED MACHINE LEARNING FINAL PROJECT


# Heart Failure Clinical Records

KAVITHA SUNDARAM

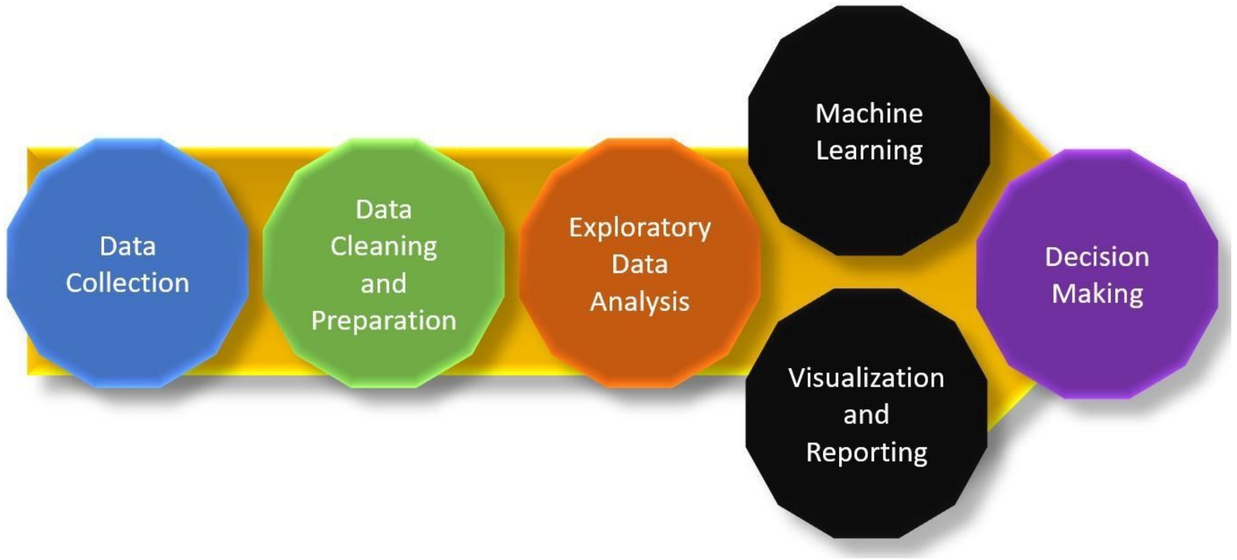
# DATA SOURCE

- Heart failure is a serious condition and there is no cure for this disease. It is a situation in which the patient's heart is not pumping the blood well as the normal heart pumps. Heart Failure prediction is a complex task in the medical field. The rates of heart failure have been increasing day by day as the rate of population is also increasing day by day.
- This paper aims at analysing the machine learning algorithms based on the percentage of various performance metrics (such as, Accuracy, Precision and Recall). The machine learning methodology is proposed. The most suitable algorithm for each metrics is predicted. It is analyzed using the specific variables in the dataset by using the python programming as well as different supervised machine learning algorithms which include, Decision Tree, Logistic Regression, KNN and Random Forest. Anaconda jupyter notebook is used for implementing python scripting.

DataSource: https://archive.ics.uci.edu/ml/datasets/Heart+failure+clinical+records

# CONTENTS





# DESCRIPTION

- This dataset contains the medical records of 299 patients who had heart failure, collected during their follow-up period, where each patient profile has 13 clinical features.

- Provide the names, email addresses, institutions, and other contact information of the donors and creators of the data set.The original dataset version was collected by Tanvir Ahmad, Assia Munir, Sajjad Haider Bhatti, Muhammad Aftab, and Muhammad Ali Raza (Government College University, Faisalabad, Pakistan) and made available by them on FigShare under the Attribution 4.0 International (CC BY 4.0: freedom to share and adapt the material) copyright in July 2017.

- HF: Heart Failure is medical term.

# DESCRIPTION

```
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
```

hf_rec.info()

| | 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 |

hf_rec.head()

# DESCRIPTION

| | age | anaemia | creatinine_phosphokinase | diabetes | ejection_fraction | high_blood_pressure | platelets |
|---|---|---|---|---|---|---|---|
| **count** | 299.000000 | 299.000000 | 299.000000 | 299.000000 | 299.000000 | 299.000000 | 299.000000 |
| **mean** | 60.833893 | 0.431438 | 581.839465 | 0.418060 | 38.083612 | 0.351171 | 263358.029264 |
| **std** | 11.894809 | 0.496107 | 970.287881 | 0.494067 | 11.834841 | 0.478136 | 97804.236869 |
| **min** | 40.000000 | 0.000000 | 23.000000 | 0.000000 | 14.000000 | 0.000000 | 25100.000000 |
| **25%** | 51.000000 | 0.000000 | 116.500000 | 0.000000 | 30.000000 | 0.000000 | 212500.000000 |
| **50%** | 60.000000 | 0.000000 | 250.000000 | 0.000000 | 38.000000 | 0.000000 | 262000.000000 |
| **75%** | 70.000000 | 1.000000 | 582.000000 | 1.000000 | 45.000000 | 1.000000 | 303500.000000 |
| **max** | 95.000000 | 1.000000 | 7861.000000 | 1.000000 | 80.000000 | 1.000000 | 850000.000000 |

| ure | platelets | serum_creatinine | serum_sodium | sex | smoking | time | DEATH_EVENT |
|---|---|---|---|---|---|---|---|
| 000 | 299.000000 | 299.00000 | 299.000000 | 299.000000 | 299.00000 | 299.000000 | 299.00000 |
| 71 | 263358.029264 | 1.39388 | 136.625418 | 0.648829 | 0.32107 | 130.260870 | 0.32107 |
| 36 | 97804.236869 | 1.03451 | 4.412477 | 0.478136 | 0.46767 | 77.614208 | 0.46767 |
| 000 | 25100.000000 | 0.50000 | 113.000000 | 0.000000 | 0.00000 | 4.000000 | 0.00000 |
| 000 | 212500.000000 | 0.90000 | 134.000000 | 0.000000 | 0.00000 | 73.000000 | 0.00000 |
| 000 | 262000.000000 | 1.10000 | 137.000000 | 1.000000 | 0.00000 | 115.000000 | 0.00000 |
| 000 | 303500.000000 | 1.40000 | 140.000000 | 1.000000 | 1.00000 | 203.000000 | 1.00000 |
| 000 | 850000.000000 | 9.40000 | 148.000000 | 1.000000 | 1.00000 | 285.000000 | 1.00000 |

Df Description of all features including DEATH_EVENT.

# EXPLORATORY DATA ANALYSIS(EDA)

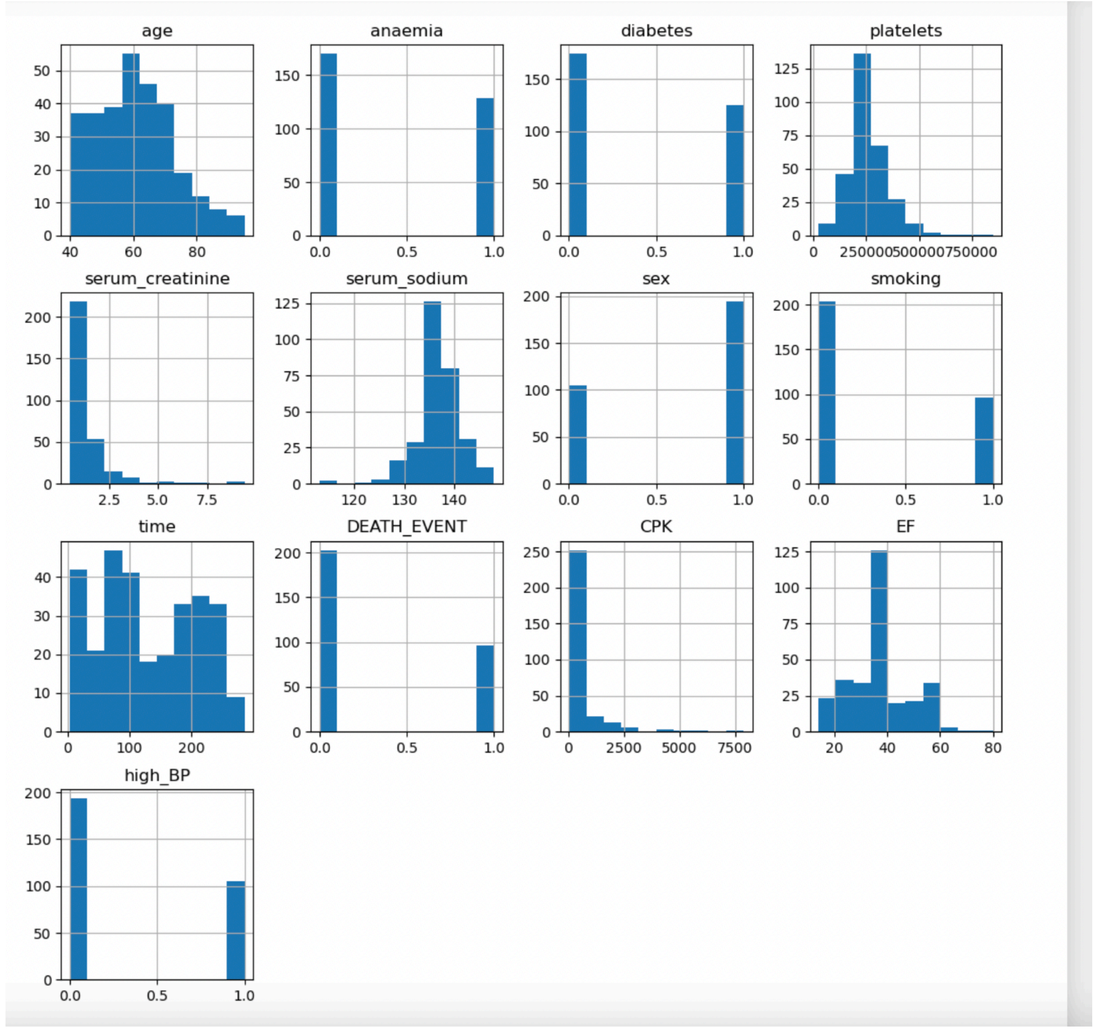
- For future analysis , am going to rename some variables in short form to predict the analysis way better.
- Lets Check null values and data types of all variables for model analysis.

```
age                   float64
anaemia                 int64
diabetes                int64
platelets             float64
serum_creatinine      float64
serum_sodium            int64
sex                     int64
smoking                 int64
time                    int64
DEATH_EVENT             int64
CPK                     int64
EF                      int64
high_BP                 int64
dtype: object
```

```
age                   0
anaemia               0
diabetes              0
platelets             0
serum_creatinine      0
serum_sodium          0
sex                   0
smoking               0
time                  0
DEATH_EVENT           0
CPK                   0
EF                    0
high_BP               0
dtype: int64
```

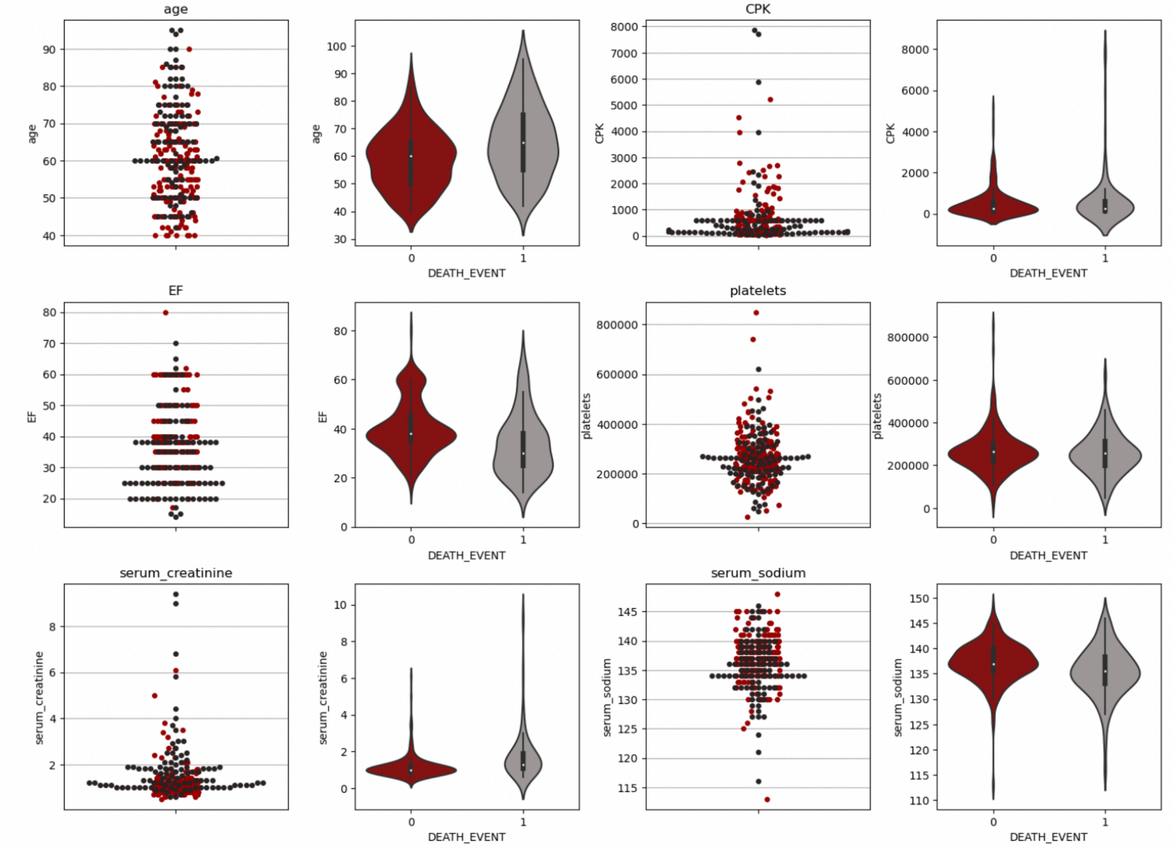
# EXPLORATORY DATA ANALYSIS(EDA)

- After analysing above histograms, we can easily divide our variables into
- categorical(anaemia, diabetes, sex, smoking, high_BP)
- numerical(age, platelets, serum_creatinine, serum_sodium, time, CPK, EF)



# DATA PREPROCESSING

- Look at the structure of EF and serum creatinine, both are having difference in violi plots.
- Lets analyse more of categorical datatypes.
- After looking up with **serum_creatinine** values its over normal for patients with high level serum are vulnerable to heart failure.
- EF **ejection_fraction** values its under norma for patients with high level ejection fraction a also vulnerable to heart failure.

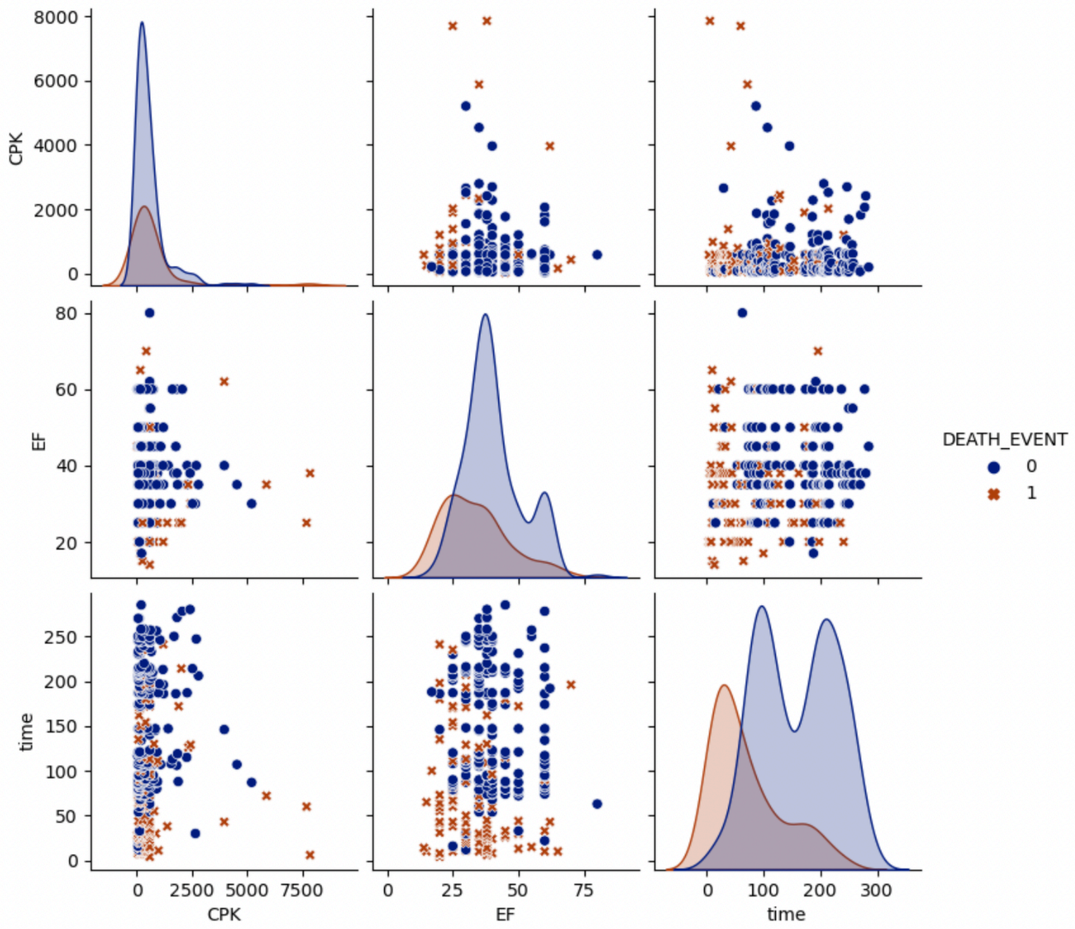



Violin plots

# DATA PREPROCESSING

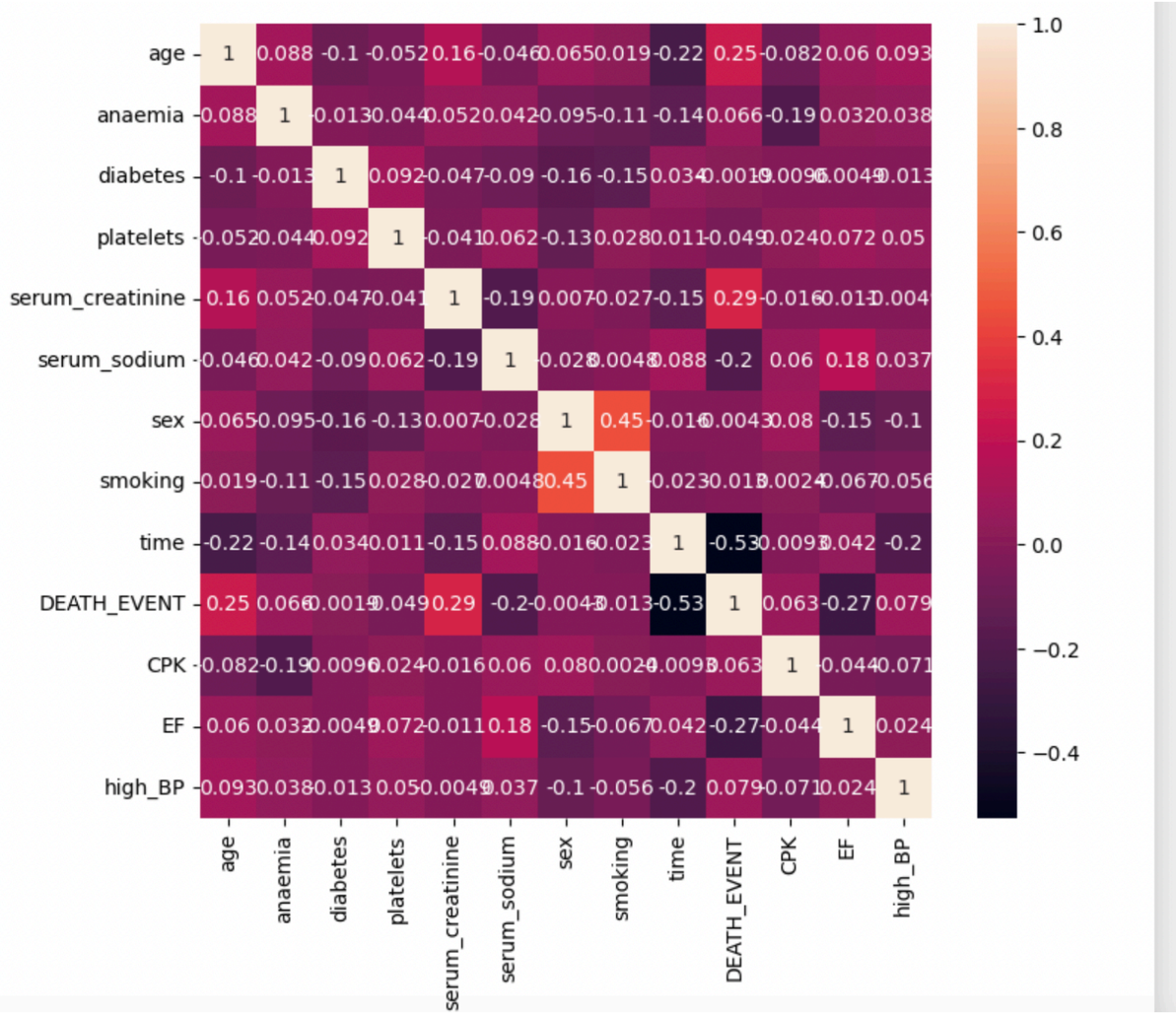
- Normal medical range for **Creatine phosphokinase**: 2 - 210 mcg/L .
- Normal medical range for **Ejection fraction** : 50 %.
- Total of 299 patients, approximately 92% of patients had heart failure and passed away due to high level of CPK(which is more than 210mcg/L and EF ).
- Dataset has some unbalanced data with values.

```
142 patients of 299 has not normal value for each feature. Representing 4
7.49 percent of patients and 147.91666666666666 percent of total death.
```



# DATA PREPROCESSING

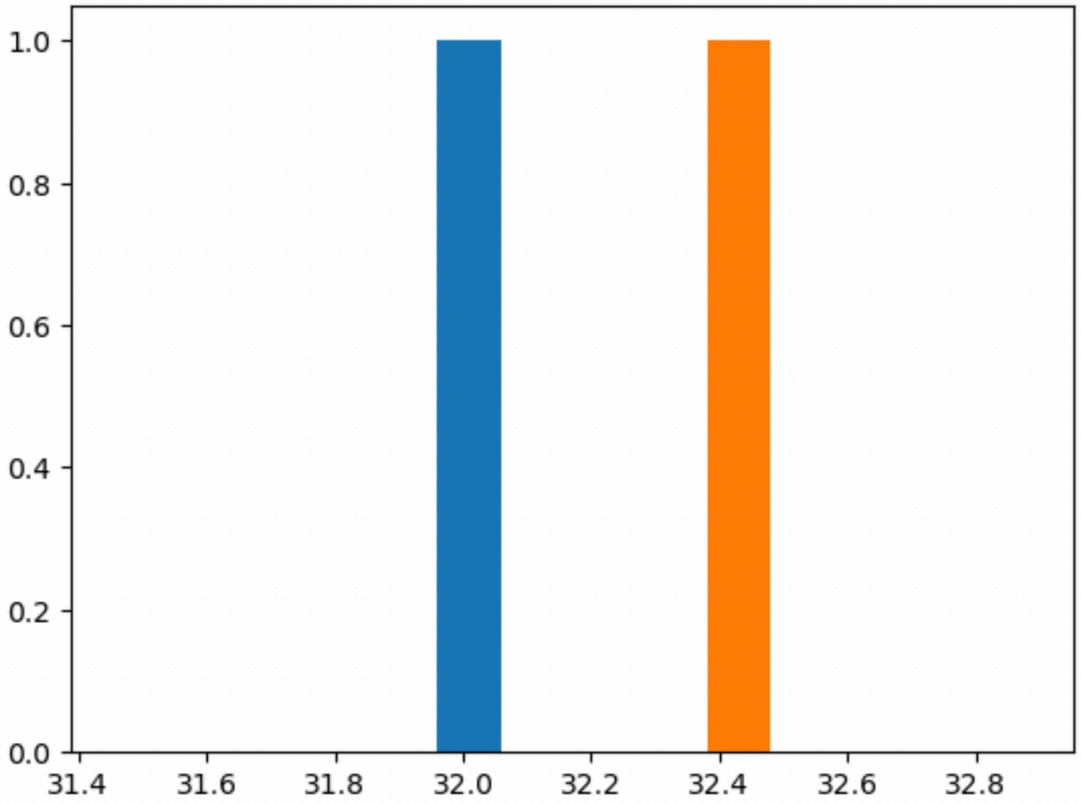
- Most of the variables are uncorrelated. as you can see **sex** and **smoking** are positively correlated.
- The color coding indicates the strength of correlation between variables, with darker shades indicating higher positive correlation and lighter shades indicating lower correlation or negative correlation.
- Age is positively correlated with serum creatinine, serum sodium, and ejection fraction, indicating that older individuals tend to have higher levels of these variables.
- diabetes with age>60 is more vulnerable to heart failure than non-diabetes aged < 50.



# DATA PREPROCESSING

```
hf_rec.corrwith(hf_rec["DEATH_EVENT"])
```

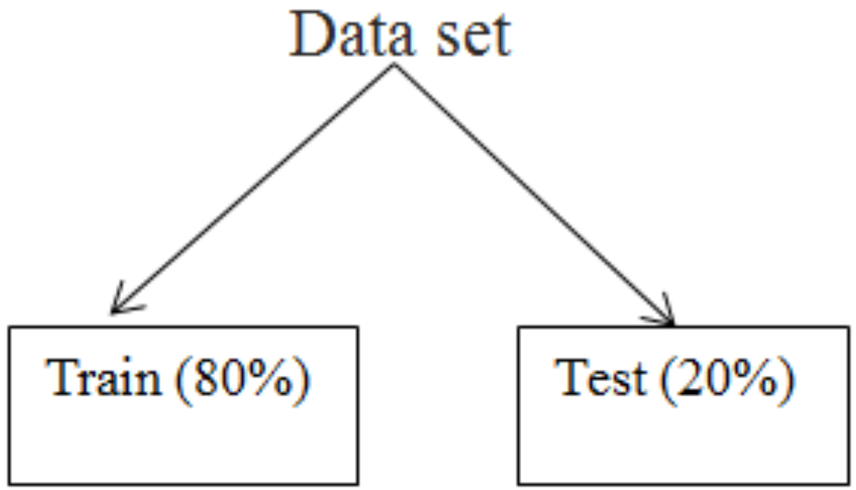
```
age                       0.253729
anaemia                   0.066270
diabetes                 -0.001943
platelets                -0.049139
serum_creatinine          0.294278
serum_sodium             -0.195204
sex                      -0.004316
smoking                  -0.012623
time                     -0.526964
DEATH_EVENT               1.000000
CPK                       0.062728
EF                       -0.268603
high_BP                   0.079351
dtype: float64
```



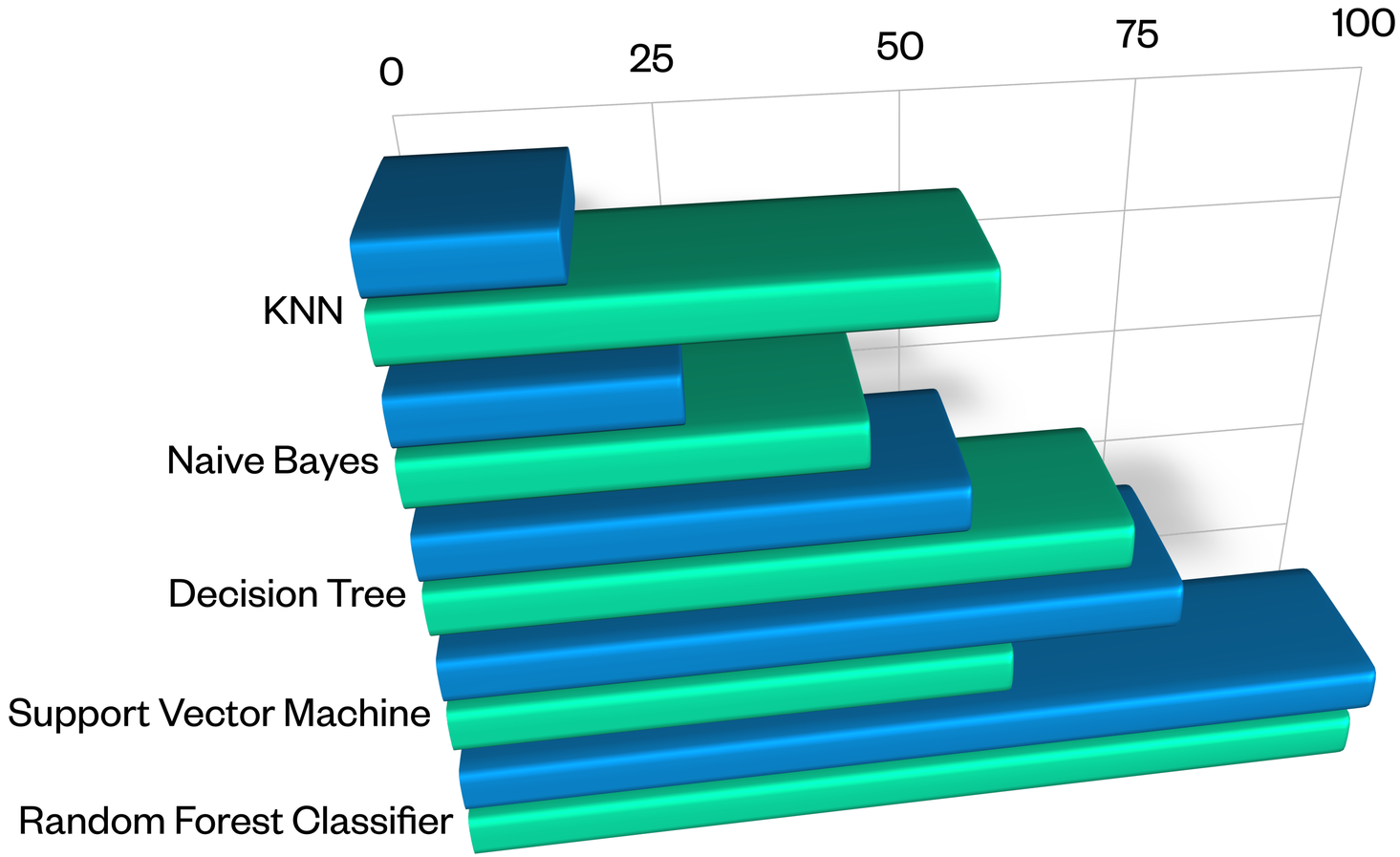

1. 32% mens and 32.4% womens are vulnerable to heart failure and death
2. 68% mens and 67.6% womens are not vulnerable to heart failure.

# VALIDATION AND SPLITTING DATA

- Splitting data into train and test samples with .80:.20 ratio.
- scaling data for future model classification .
- The normalisation has been done to make all the attribute values between zero and one (0-1) to reach better accuracy.

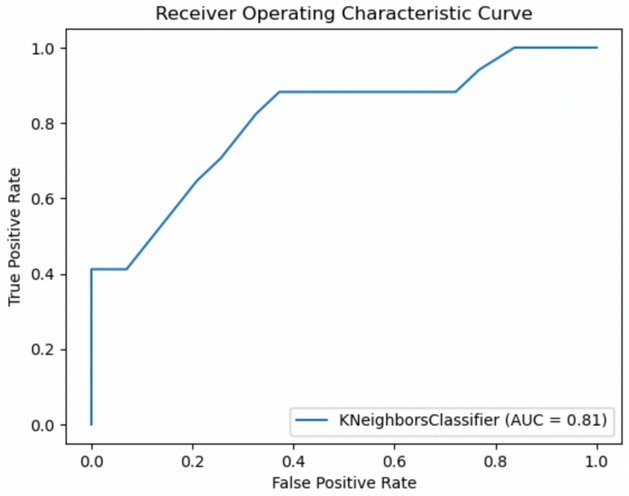


# MODEL CLASSIFICATION

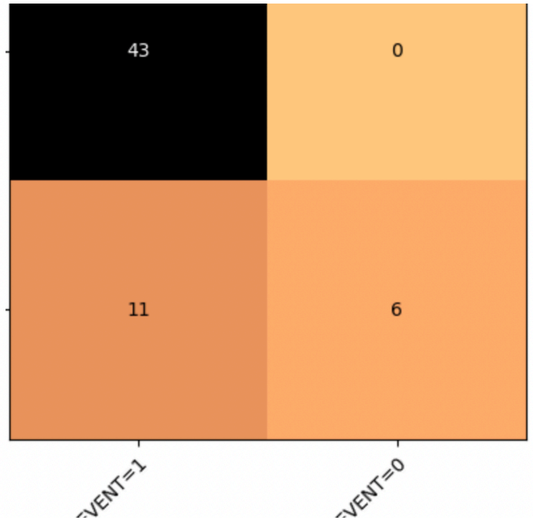


# MODEL CLASSIFICATION

- We can tell the average accuracy for this **KNN** is the average of the F1-score for both labels, which is 0.78 in our case
- accuracy predicted is this case = 81%
- 43 out of 43 death count was predicted exactly
- 6 out of 17 values were predicted correctly and 11 was incorrectly forecasted




ROC CURVE



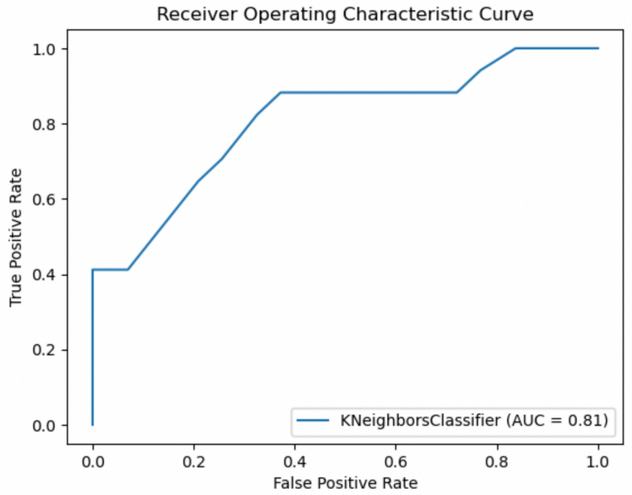

CONFUSION MATRIX

| | precision | recall | f1-score | support |
|---|---|---|---|---|
| 0 | 0.80 | 1.00 | 0.89 | 43 |
| 1 | 1.00 | 0.35 | 0.52 | 17 |
| accuracy | | | 0.82 | 60 |
| macro avg | 0.90 | 0.68 | 0.70 | 60 |
| weighted avg | 0.85 | 0.82 | 0.78 | 60 |

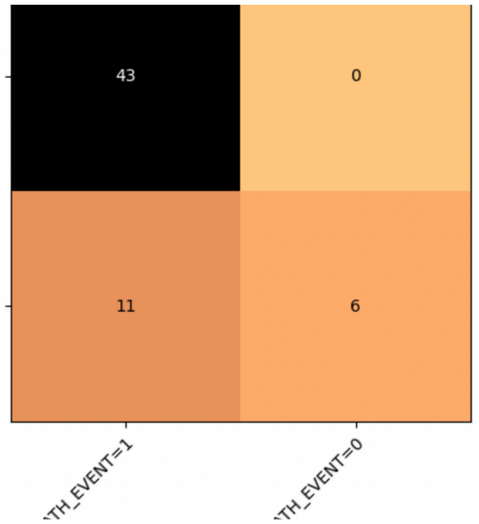
KNN ACCURACY SCORE

# MODEL CLASSIFICATION

- We can tell the average accuracy for this **NAIVE BAYES** is the average of the F1-score for both labels, which is 0.57 in our case
- 2. accuracy predicted is this case = 67%
- 3. **40 out of 43 death count was predicted exactly and 3 was incorrectly forecsated**
- 4. **17 out of 17 values were incorrectly forecasted**
-




ROC CURVE



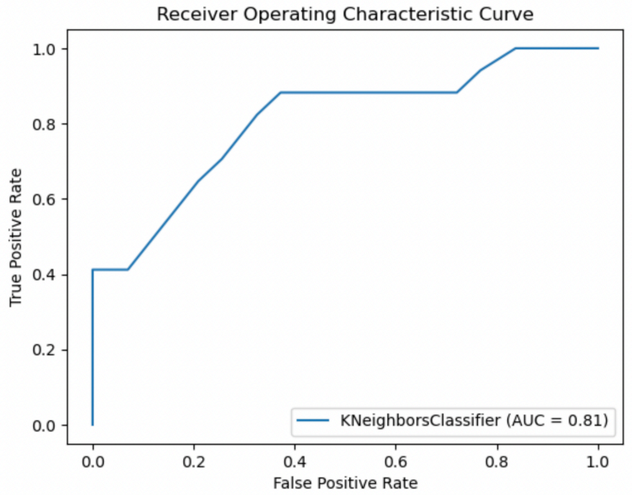

CONFUSION MATRIX

| | precision | recall | f1-score | support |
|---|---|---|---|---|
| 0 | 0.80 | 1.00 | 0.89 | 43 |
| 1 | 1.00 | 0.35 | 0.52 | 17 |
| accuracy | | | 0.82 | 60 |
| macro avg | 0.90 | 0.68 | 0.70 | 60 |
| weighted avg | 0.85 | 0.82 | 0.78 | 60 |

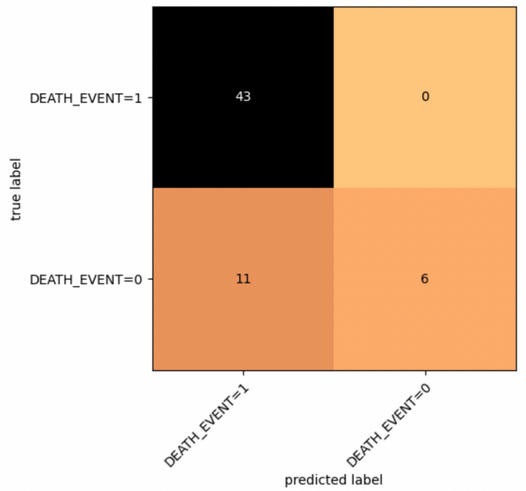
ACCURACY SCORE

# MODEL CLASSIFICATION

- We can tell the average accuracy for this **Decision Tree** is the average of the F1-score for both labels, which is 0.82 in our case
- accuracy predicted is this case = 81.67%
- 35 out of 43 death count was predicted exactly and 8 was incorrectly forecsated
- 13 out of 17 values were predicted exactly and 4 was incorrectly forecasted




ROC CURVE



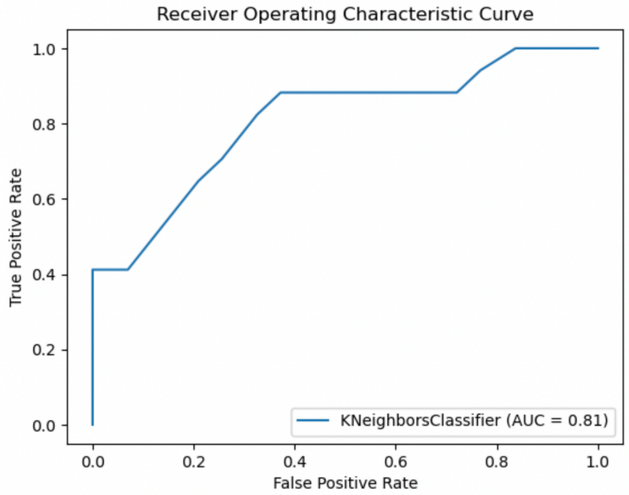

CONFUSION MATRIX

| | precision | recall | f1-score | support |
|---|---|---|---|---|
| 0 | 0.80 | 1.00 | 0.89 | 43 |
| 1 | 1.00 | 0.35 | 0.52 | 17 |
| accuracy | | | 0.82 | 60 |
| macro avg | 0.90 | 0.68 | 0.70 | 60 |
| weighted avg | 0.85 | 0.82 | 0.78 | 60 |

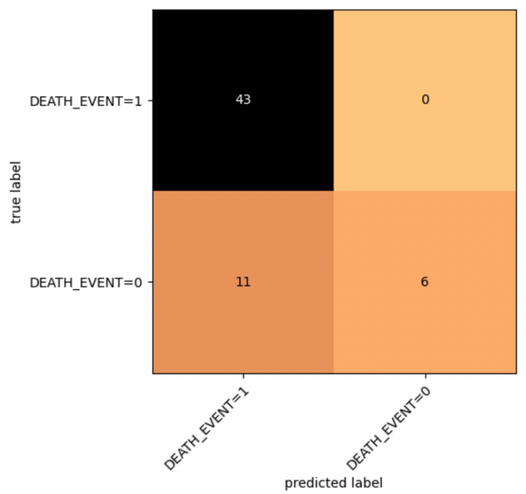
ACCURACY SCORE

# MODEL CLASSIFICATION

- We can tell the average accuracy for this **SVM** is the average of the F1-score for both labels, which is 0.60 in our case
- accuracy predicted is this case = 71.67%
- 43 out of 43 death count was predicted exactly
- 17 out of 17 values were incorrectly forecasted




ROC CURVE



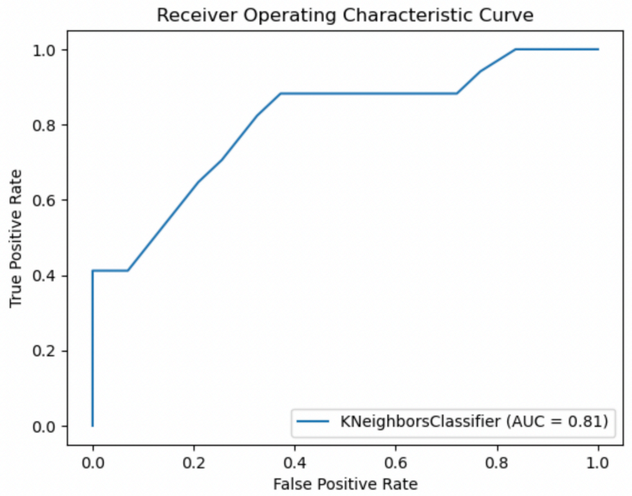

CONFUSION MATRIX

| | precision | recall | f1-score | support |
|---|---|---|---|---|
| 0 | 0.80 | 1.00 | 0.89 | 43 |
| 1 | 1.00 | 0.35 | 0.52 | 17 |
| accuracy | | | 0.82 | 60 |
| macro avg | 0.90 | 0.68 | 0.70 | 60 |
| weighted avg | 0.85 | 0.82 | 0.78 | 60 |

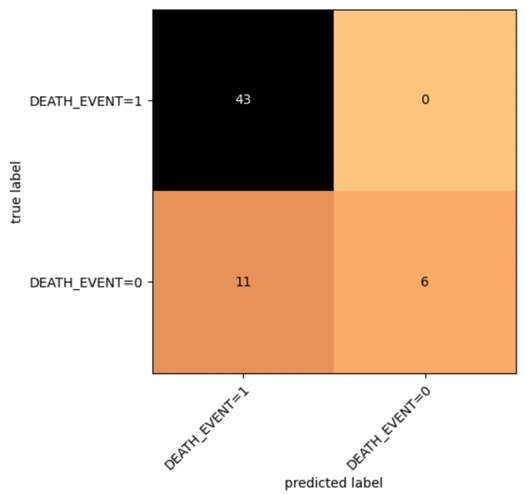
ACCURACY SCORE

# MODEL CLASSIFICATION

- We can tell the average accuracy for this **RANDOM FOREST** is the average of the F1-score for both labels, which is 0.90 in our case
- accuracy predicted is this case = 90%
- 42 out of 43 death count was predicted exactly and 1 were incorrectly forecasted * *
- 11 out of 17 values was predicted exactly and 6 were incorrectly forecasted* *




ROC CURVE

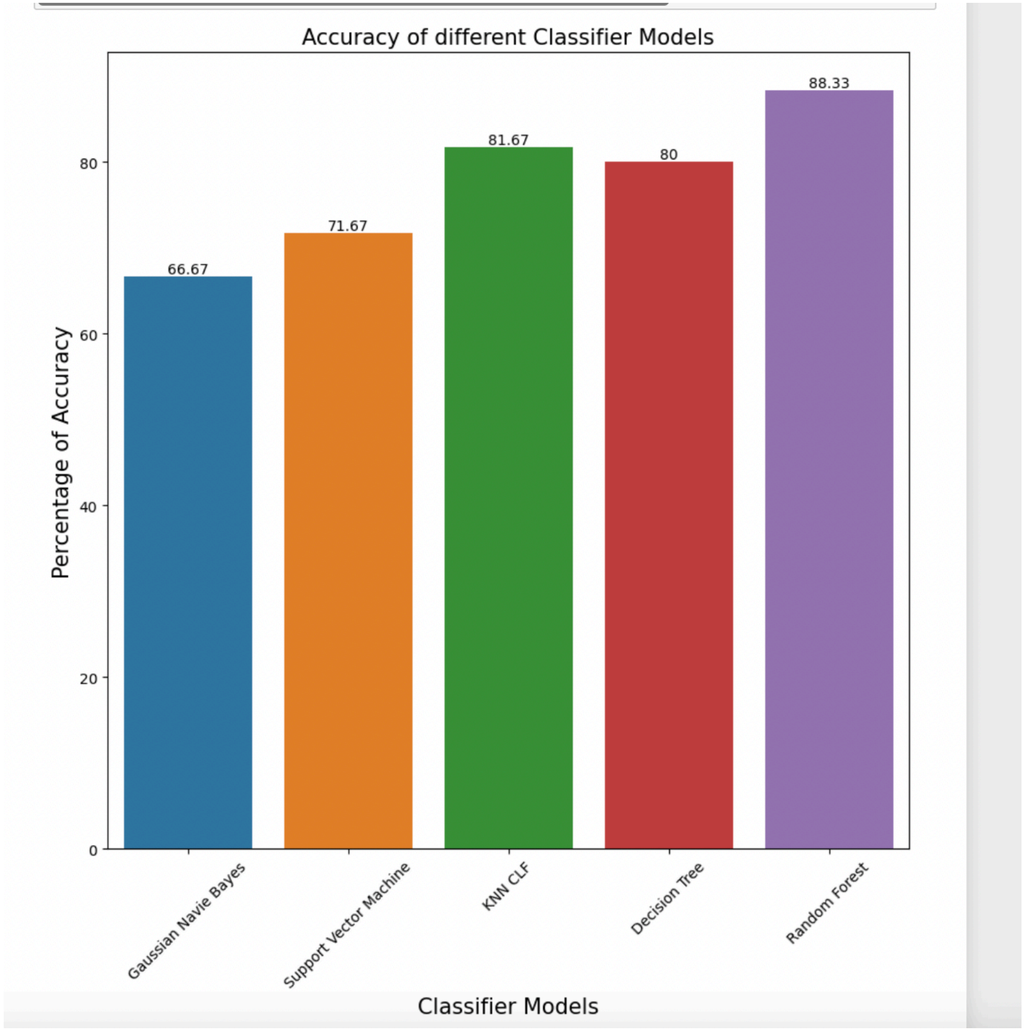



CONFUSION MATRIX

| | precision | recall | f1-score | support |
|---|---|---|---|---|
| 0 | 0.80 | 1.00 | 0.89 | 43 |
| 1 | 1.00 | 0.35 | 0.52 | 17 |
| accuracy | | | 0.82 | 60 |
| macro avg | 0.90 | 0.68 | 0.70 | 60 |
| weighted avg | 0.85 | 0.82 | 0.78 | 60 |

ACCURACY SCORE

# RESULT AND ANALYSIS



# CONCLUSION

- Best overall model seems to be the random forest trained on the oversampled dataset, that delivers the best results in terms of accuracy and f1 score.
- For the models that allow it, it's possible to evaluate the ROC curve to select a threshold according to the main goal (minimise false positives or maximise true positives) but the results in the table are obtained by fixing the threshold at 0.5.
- I have used almost all classification algorithm models to predict the accuracy of heart failure according to the feature provided with dataset.
- Random-forest makes the best model out of all.as 90% accuracy.
- I also want to look into feature selection for logistic regression algorithms. I focused mainly on tuning my random forest algorithm here, but maybe I could get more consistent results from my logistic regression by applying feature selection beyond collinearity corrections.

# REFERENCES


kaggle references:
https://www.kaggle.com/datasets/andrewmvd/heart-failure-clinical-data
medium:
https://medium.com/@ammar.j.alashhab/using-machine-learning-algorithm-in-heart-failure
gridDB:
https://griddb.net/en/blog/heart-failure-prediction-using-machine-learning-python-and-griddb
stack-overflow:
https://stackoverflow.com/questions/54084452/how-to-plot-seaborn-pairplot-as-subplot
plotly:
https://plotly.com/python/violin/


# GITHUB REPOSITORY LINK

https://github.com/kavishant87/Supervised_Final_Project_5509


# THANKS FOR WATCHING...

KAVITHA SUNDARAM