MACHINE LEARNING PROJECT

HEART FAILURE PREDICTION

MELIS HARMANTEPE

DATASET

Attributes (13 clinical features):

- age: age of the patient (years)
- anaemia: decrease of red blood cells or hemoglobin (boolean)
- high blood pressure: if the patient has hypertension (boolean)
- creatinine phosphokinase (CPK): level of the CPK enzyme in the blood (mcg/L)
- diabetes: if the patient has diabetes (boolean)
- ejection fraction: percentage of blood leaving the heart at each contraction (percentage)
- platelets: platelets in the blood (kiloplatelets/mL)
- sex: woman or man (binary)
- serum creatinine: level of serum creatinine in the blood (mg/dL)
- serum sodium: level of serum sodium in the blood (mEq/L)
- smoking: if the patient smokes or not (boolean)
- time: follow-up period (days)
- [target] death event: if the patient deceased during the follow-up period (boolean)

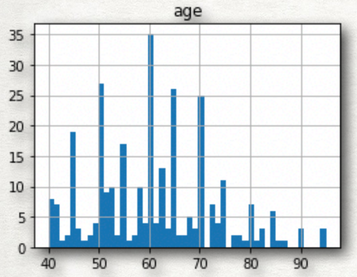
299 Patients

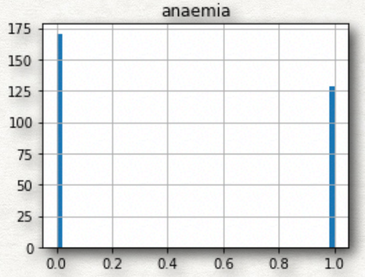
```
data.info()
```

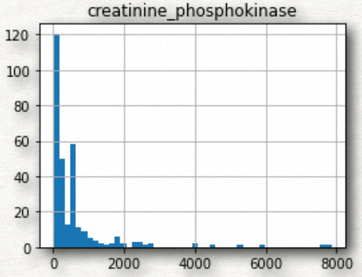
```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 299 entries, 0 to 298
Data columns (total 13 columns):
 #   Column                    Non-Null Count  Dtype
---  ------                    --------------  -----
 0   age                       299 non-null    float64
 1   anaemia                   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
```

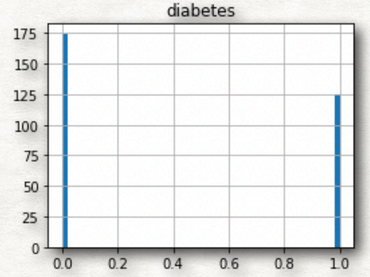
Problem statement: I want to build a classification model which estimates a person's probability of deceasing from heart failure based on all of the 12 risk factors (clinical features).

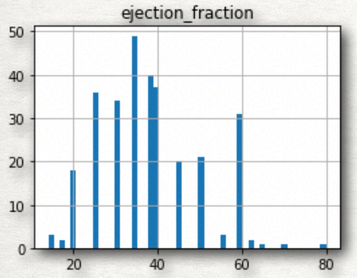
DATA VISUALIZATION & DATA PREPROCESSING

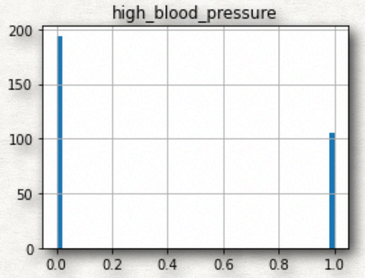


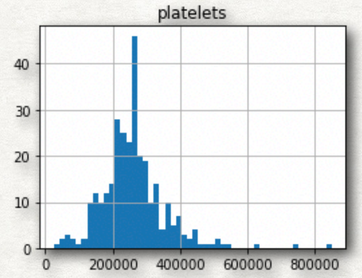


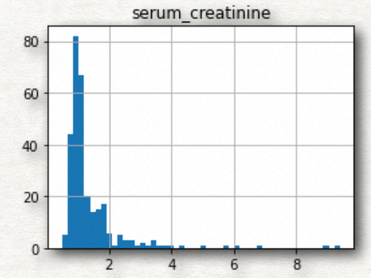


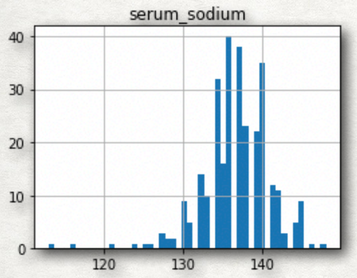


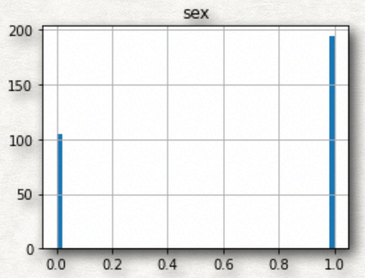


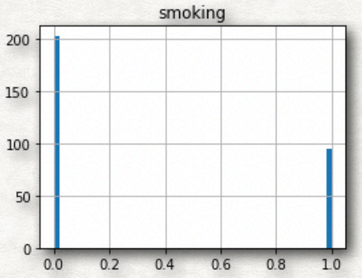


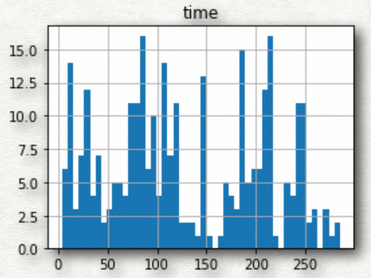


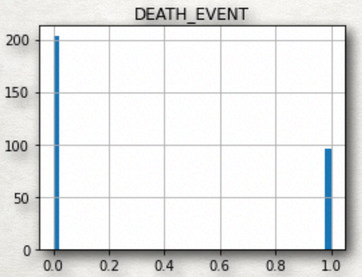


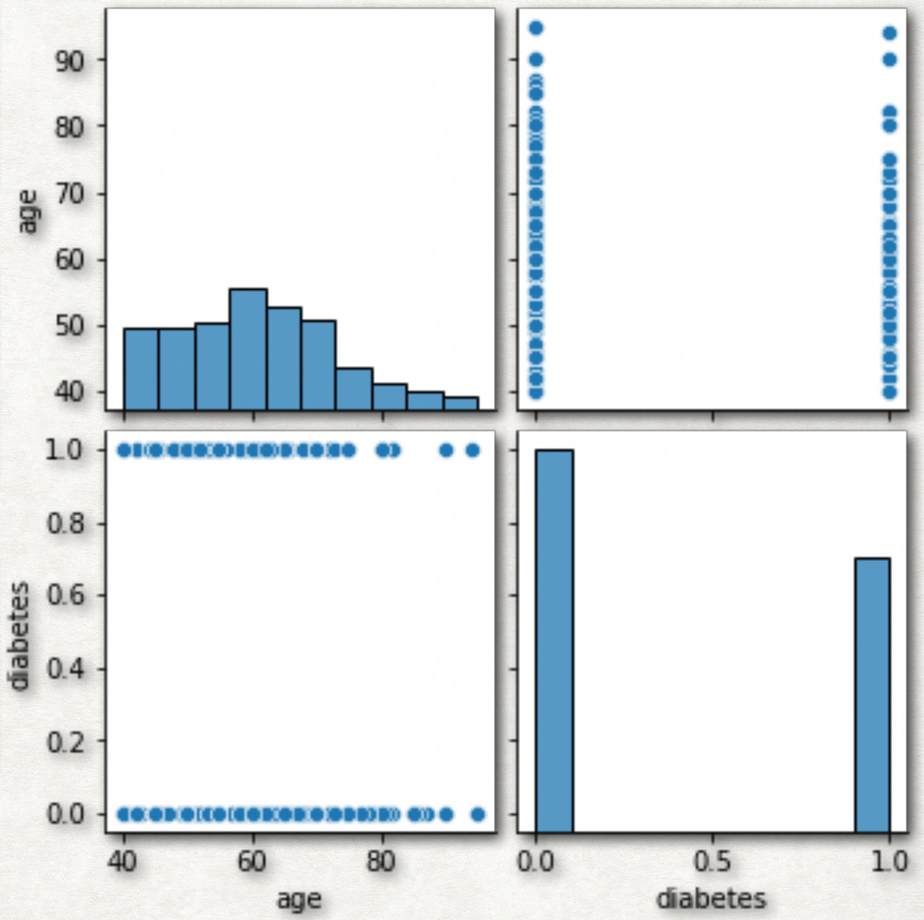






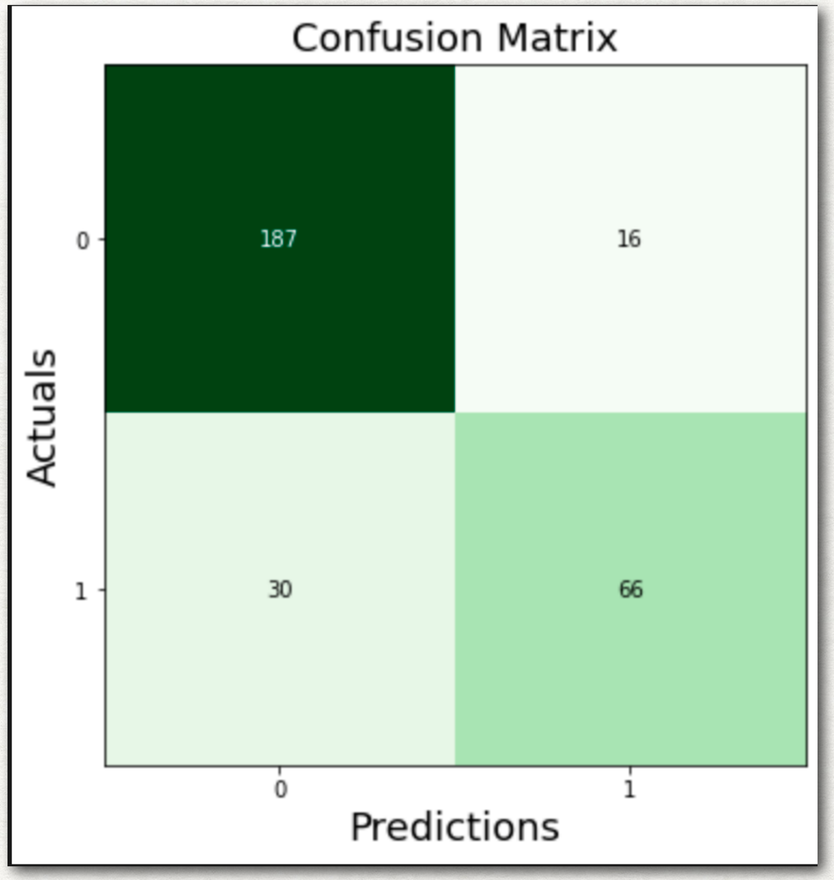






LOGISTIC REGRESSION

```
Training accuracy = 82.27%
Test set accuracy = 28.33%
Execution time is: 0.07134466599999989 seconds
```



	precision	recall	f1-score	support
0.0	0.87	0.89	0.88	203
1.0	0.76	0.71	0.73	96
accuracy			0.83	299
macro avg	0.81	0.80	0.80	299
weighted avg	0.83	0.83	0.83	299

```
#ROC
roc_auc_score(y, y_pred)
```

```
0.7999794745484401
```

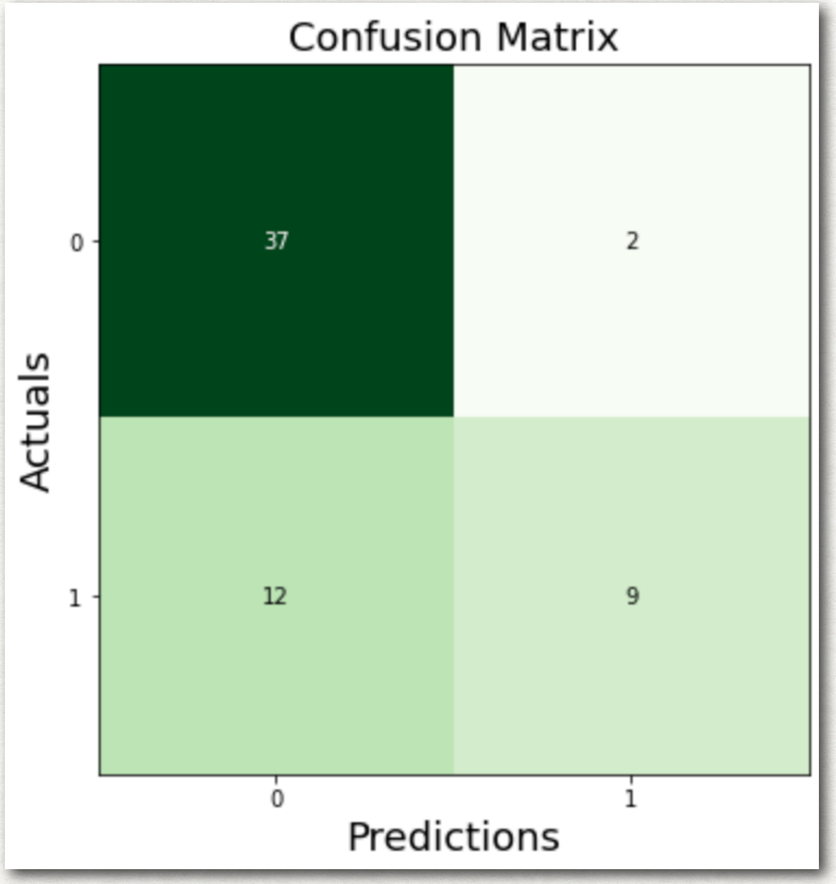
5.1 Confusion Matrix

```
cm_LR = confusion_matrix(y, y_pred)
print(cm_LR)
```

```
[[181  22]
 [ 28  68]]
```

SVM

```
Training accuracy = 81.94%
Test set accuracy = 25.00%
Execution time is: 51.198486577 seconds
```



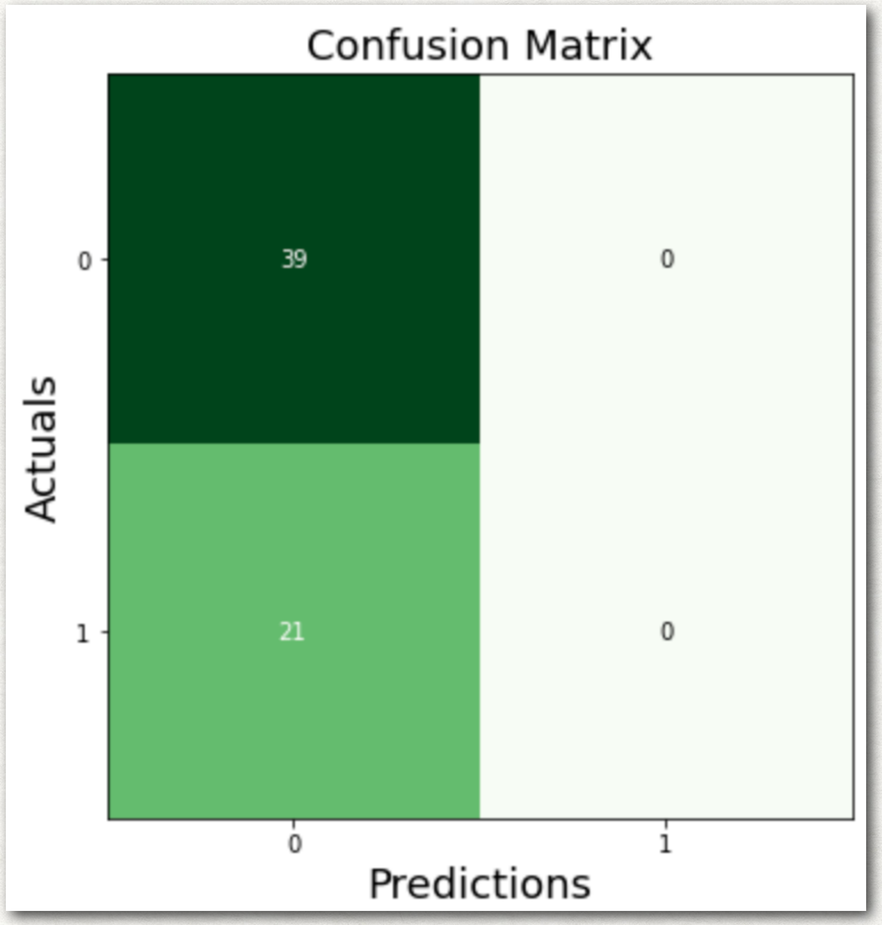
	precision	recall	f1-score	support
0.0	0.76	0.95	0.84	39
1.0	0.82	0.43	0.56	21
accuracy			0.77	60
macro avg	0.79	0.69	0.70	60
weighted avg	0.78	0.77	0.74	60

```
#ROC
roc_auc_score(y_test, y_pred_svc)
```

```
0.6886446886446886
```

```
Training accuracy = 68.56%
Test set accuracy = 15.00%
Execution time is: 0.14908029699995495 seconds
```

NEURAL NETWORKS



	precision	recall	f1-score	support
0.0	0.65	1.00	0.79	39
1.0	0.00	0.00	0.00	21
accuracy			0.65	60
macro avg	0.33	0.50	0.39	60
weighted avg	0.42	0.65	0.51	60

```
#ROC
roc_auc_score(y_test, y_pred_nn)
```

```
0.5
```

```
cm_NN = confusion_matrix(y_test, y_pred_nn)
print(cm_NN)
```

```
[[39  0]
 [21  0]]
```

CONCLUSION

	Logistic Regression	SVM	Neural Networks
Accuracy	0.83	0.82	0.68
F1 Score	0.73	0.56	0.0
AUC	0.80	0.68	0.5
Execution time (secs)	0.07	0.14	51

Logistic Regression > SVM > Neural Networks