Homework 5

Run the following cell to load `heart_processed.csv`, which has log-predictors from the Heart Failure Clinical Records Dataset for predicting `DEATH_EVENT`.

In [24]:
```
import pandas as pd
import numpy as np
dataset = pd.read_csv("heart_processed.csv")
X = dataset.drop("DEATH_EVENT", axis=1)
y = dataset["DEATH_EVENT"]
X=X.iloc[:,1:]
```

Write a naive Bayes classifier with priors inferred from the dataset, and class-conditional densities inferred using `scipy.stats.gaussian_kde` with default bandwidth. Print the accuracy on the dataset.

In [25]:
```
import scipy.stats as stats
indices_0 = y.index[y==0]
indices_1 = y.index[y==1]
X_0 = X.iloc[indices_0, :]
X_1 = X.iloc[indices_1, :]
prior_0 = X_0.shape[0] / X.shape[0]
prior_1 = X_1.shape[0] / X.shape[0]

lst0=[]
for i in range(0,X_0.shape[1]):
    kde0=stats.gaussian_kde(X_0.iloc[:,i])
    res0=kde0.evaluate(X.iloc[:,i])
    lst0.append(res0)
lst1=[]
for i in range(0,X_1.shape[1]):
    kde1=stats.gaussian_kde(X_1.iloc[:,i])
    res1=kde1.evaluate(X.iloc[:,i])
    lst1.append(res1)

def Naive(X):
    likelihood_0=np.prod(lst0, axis=0)
    likelihood_1=np.prod(lst1, axis=0)
    posterior_0 = prior_0 * likelihood_0
    posterior_1 = prior_1 * likelihood_1
    return (posterior_1 > posterior_0).astype(int)
# Accuracy
np.sum(Naive(X) == y) / len(y)
```

Out[25]:
```
0.7892976588628763
```