In [2]:

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
%matplotlib inline
```

In [3]:

```
hrv=pd.read_csv("/Users/baskaran/Downloads/HRV csv.csv")
```

In [4]:

```
hrv
```

	> 1	Sepsis3	Mean.rate	Coefficient.of.variation	Poincar..SD1	Poincar..SD2	LF.HF.ratio.LombScargle	LF.HF
0	0	1	89.411160	0.130259	0.045349	0.115000	1.996490	
1	0	1	86.805760	0.026824	0.009504	0.024437	0.879037	
2	0	1	85.507670	0.035203	0.009715	0.033555	2.542225	
3	0	1	95.127480	0.013182	0.006829	0.009571	1.596207	
4	0	1	93.370410	0.016777	0.006921	0.013585	1.128488	
...	...	...	...	...	...	...	...	
4309	0	0	111.683756	0.031518	0.015239	0.018471	0.495158	
4310	0	0	111.409998	0.048524	0.021231	0.030251	0.468523	
4311	0	0	114.402754	0.053378	0.009238	0.038498	3.641941	
4312	0	0	120.986143	0.049403	0.012527	0.032305	1.138312	
4313	0	0	109.532809	0.023737	0.007142	0.016945	3.862068	

In [5]:

```
hrv.head()
```

Out[5]:

	SI > 1	Sepsis3	Mean.rate	Coefficient.of.variation	Poincar..SD1	Poincar..SD2	LF.HF.ratio.LombSca
0	0	1	89.41116	0.130259	0.045349	0.115000	1.99(
1	0	1	86.80576	0.026824	0.009504	0.024437	0.87!
2	0	1	85.50767	0.035203	0.009715	0.033555	2.54:
3	0	1	95.12748	0.013182	0.006829	0.009571	1.59(
4	0	1	93.37041	0.016777	0.006921	0.013585	1.12!

5 rows × 59 columns

In [6]:

```
hrv.describe()
```

Out[6]:

	SI > 1	Sepsis3	Mean.rate	Coefficient.of.variation	Poincar..SD1	Poincar..SD2
count	4314.000000	4314.000000	4314.000000	4314.000000	4314.000000	4314.000000
mean	0.056328	0.124710	89.521243	0.038975	0.012284	0.036459
std	0.230581	0.330428	16.664347	0.023929	0.009302	0.026529
min	0.000000	0.000000	56.485260	0.006346	0.001003	0.004482
25%	0.000000	0.000000	75.815800	0.022457	0.005034	0.018612
50%	0.000000	0.000000	88.548570	0.032952	0.009728	0.028810
75%	0.000000	0.000000	101.961156	0.049546	0.016937	0.046719
max	1.000000	1.000000	183.984092	0.272920	0.063498	0.305502

8 rows × 59 columns

In [7]:

```
hrv.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 4314 entries, 0 to 4313
Data columns (total 59 columns):
 #   Column                                Non-Null Count  Dtype
---  ------                                --------------  -----
 0   SI > 1                                4314 non-null   int64
 1   Sepsis3                               4314 non-null   int64
 2   Mean.rate                             4314 non-null   float64
 3   Coefficient.of.variation              4314 non-null   float64
 4   Poincar..SD1                          4314 non-null   float64
 5   Poincar..SD2                          4314 non-null   float64
 6   LF.HF.ratio.LombScargle               4314 non-null   float64
 7   LF.Power.LombScargle                  4314 non-null   float64
 8   HF.Power.LombScargle                  4314 non-null   float64
 9   DFA.Alpha.1                           4314 non-null   float64
 10  DFA.Alpha.2                           4314 non-null   float64
 11  Largest.Lyapunov.exponent             4314 non-null   float64
 12  Correlation.dimension                 4314 non-null   float64
 13  Power.Law.Slope.LombScargle           4314 non-null   float64
 14  Power.Law.Y.Intercept.LombScargle     4314 non-null   float64
 15  DFA.AUC                               4314 non-null   float64
 16  Multiscale.Entropy                    4314 non-null   float64
 17  VLF.Power.LombScargle                 4314 non-null   float64
 18  Complexity                            4314 non-null   float64
 19  eScaleE                               4314 non-null   float64
 20  pR                                    4314 non-null   float64
 21  pD                                    4314 non-null   float64
 22  dlmax                                 4314 non-null   float64
 23  sedl                                  4314 non-null   float64
 24  pDpR                                  4314 non-null   float64
 25  pL                                    4314 non-null   float64
 26  vlmax                                 4314 non-null   float64
 27  sevl                                  4314 non-null   float64
 28  shannEn                               4314 non-null   float64
 29  PSeo                                  4314 non-null   float64
 30  Teo                                   4314 non-null   float64
 31  SymDp0_2                              4314 non-null   float64
 32  SymDp1_2                              4314 non-null   float64
 33  SymDp2_2                              4314 non-null   float64
 34  SymDfw_2                              4314 non-null   int64
 35  SymDse_2                              4314 non-null   float64
 36  SymDce_2                              4314 non-null   float64
 37  formF                                 4314 non-null   float64
 38  gcount                                4314 non-null   float64
 39  sgridAND                              4314 non-null   float64
 40  sgridTAU                              4314 non-null   float64
 41  sgridWGT                              4314 non-null   float64
 42  aFdP                                  4314 non-null   float64
 43  fFdP                                  4314 non-null   float64
 44  IoV                                   4314 non-null   float64
 45  KLPE                                  4314 non-null   float64
 46  AsymI                                 4314 non-null   float64
 47  CSI                                   4314 non-null   float64
 48  CVI                                   4314 non-null   float64
 49  ARerr                                 4314 non-null   float64
 50  histSI                                4314 non-null   float64
 51  MultiFractal_c1                       4314 non-null   float64
```

```
 52  MultiFractal_c2                  4314 non-null   float64
 53  SDLEalpha                        4314 non-null   float64
 54  SDLEmean                         4314 non-null   float64
 55  QSE                              4314 non-null   float64
 56  Hurst.exponent                   4314 non-null   float64
 57  mean                             4314 non-null   float64
 58  median                           4314 non-null   float64
dtypes: float64(56), int64(3)
memory usage: 1.9 MB
```

In [107]:

```
corr=hrv.corr()
corr
```

Correlation.dimension	0.009734	-0.032485	-0.166158	-0.114705	0.244667	-0.11
Power.Law.Slope.LombScargle	-0.042618	0.225378	0.018384	0.035461	0.481598	-0.05
Power.Law.Y.Intercept.LombScargle	-0.020171	0.209579	0.071411	-0.023391	0.323432	-0.08
DFA.AUC	-0.250401	0.255221	-0.660435	0.715520	0.733504	0.74
Multiscale.Entropy	-0.051426	0.023695	-0.325469	-0.220344	0.060554	-0.12
VLF.Power.LombScargle	0.083420	-0.233340	0.347033	-0.149871	-0.701779	-0.09
Complexity	0.106864	-0.177325	0.389119	-0.515126	-0.733795	-0.45
eScaleE	0.018142	-0.049849	0.001698	-0.149347	-0.137451	-0.11
pR	-0.032106	0.059559	0.196255	0.269682	0.183284	0.13
pD	-0.004502	-0.001569	0.298086	0.368837	0.013232	0.25
dlmax	-0.046364	0.032961	0.143503	0.465837	0.098844	0.35
sedl	0.011871	-0.057758	0.326715	0.330435	-0.032931	0.21
pDpR	0.024064	-0.086480	0.108515	0.169715	-0.098566	0.17

SIMPLE FEATURE SCALING

In [110]:

```
hrv=hrv/hrv.max()
hrv
```

Out[110]:

	SI > 1	Sepsis3	Mean.rate	Coefficient.of.variation	Poincar..SD1	Poincar..SD2	LF.HF.ratio.Lom
0	0.0	1.0	0.485972	0.477279	0.714180	0.376430	
1	0.0	1.0	0.471811	0.098285	0.149674	0.079990	
2	0.0	1.0	0.464756	0.128987	0.152997	0.109836	
3	0.0	1.0	0.517042	0.048300	0.107547	0.031329	
4	0.0	1.0	0.507492	0.061472	0.108996	0.044468	
...	...	...	...	...	...	...	
4309	0.0	0.0	0.607029	0.115484	0.239992	0.060461	
4310	0.0	0.0	0.605541	0.177796	0.334357	0.099021	
4311	0.0	0.0	0.621808	0.195581	0.145485	0.126016	
4312	0.0	0.0	0.657590	0.181016	0.197282	0.105744	
4313	0.0	0.0	0.595338	0.086974	0.112476	0.055466	

4314 rows × 59 columns

MIN MAX NORMALISATION

In [111]:

```
hrv_mmn=pd.read_csv("/Users/baskaran/Downloads/HRV csv.csv")
hrv_mmn=(hrv_mmn-hrv_mmn.min())/(hrv_mmn.max()-hrv_mmn.min())
hrv_mmn
```

Out[111]:

	SI > 1	Sepsis3	Mean.rate	Coefficient.of.variation	Poincar..SD1	Poincar..SD2	LF.HF.ratio.LombScargle	LF.Pow
0	0.0	1.0	0.258245	0.464835	0.709593	0.367145	0.040955	
1	0.0	1.0	0.237810	0.076819	0.136027	0.066291	0.017583	
2	0.0	1.0	0.227629	0.108251	0.139403	0.096582	0.052369	
3	0.0	1.0	0.303079	0.025644	0.093223	0.016906	0.032583	
4	0.0	1.0	0.289298	0.039130	0.094696	0.030241	0.022800	
...	...	...	...	...	...	...	...	
4309	0.0	0.0	0.432933	0.094428	0.227794	0.046472	0.009554	
4310	0.0	0.0	0.430786	0.158222	0.323674	0.085606	0.008996	
4311	0.0	0.0	0.454259	0.176431	0.131771	0.113002	0.075370	

Z-SCORE NORMALISATION

In [112]:

```
hrv_zs=pd.read_csv("/Users/baskaran/Downloads/HRV csv.csv")
hrv_zs=(hrv_zs-hrv_zs.mean())/hrv_zs.std()
hrv_zs
```

	SI > 1	Sepsis3	Mean.rate	Coefficient.of.variation	Poincar..SD1	Poincar..SD2	LF.HF.ratio.LombScargle
0	-0.244288	2.648955	-0.006606	3.814704	3.554740	2.960603	-0.151281
1	-0.244288	2.648955	-0.162952	-0.507799	-0.298894	-0.453155	-0.546556
2	-0.244288	2.648955	-0.240848	-0.157645	-0.276210	-0.109453	0.041760
3	-0.244288	2.648955	0.336421	-1.077893	-0.586479	-1.013526	-0.292873
4	-0.244288	2.648955	0.230982	-0.927659	-0.576588	-0.862219	-0.458318
...	...	...	...	...	...	...	...
4309	-0.244288	-0.377420	1.329936	-0.311639	0.317666	-0.678042	-0.682344
4310	-0.244288	-0.377420	1.313508	0.399034	0.961855	-0.233997	-0.691766
4311	-0.244288	-0.377420	1.493099	0.601881	-0.327491	0.076872	0.430761
4312	-0.244288	-0.377420	1.888157	0.435767	0.026103	-0.156572	-0.454843
4313	-0.244288	-0.377420	1.200861	-0.636804	-0.552829	-0.735564	0.508626

In [113]:

```
import pandas as pd
from sklearn import preprocessing
hrv=pd.read_csv("/Users/baskaran/Downloads/HRV csv.csv")
```

In [114]:

```
hrv.columns
```

Out[114]:

```
Index(['SI > 1', 'Sepsis3', 'Mean.rate', 'Coefficient.of.variation',
       'Poincar..SD1', 'Poincar..SD2', 'LF.HF.ratio.LombScargle',
       'LF.Power.LombScargle', 'HF.Power.LombScargle', 'DFA.Alpha.1',
       'DFA.Alpha.2', 'Largest.Lyapunov.exponent', 'Correlation.dimens
ion',
       'Power.Law.Slope.LombScargle', 'Power.Law.Y.Intercept.LombScarg
le',
       'DFA.AUC', 'Multiscale.Entropy', 'VLF.Power.LombScargle', 'Comp
lexity',
       'eScaleE', 'pR', 'pD', 'dlmax', 'sedl', 'pDpR', 'pL', 'vlmax',
'sevl',
       'shannEn', 'PSeo', 'Teo', 'SymDp0_2', 'SymDp1_2', 'SymDp2_2',
       'SymDfw_2', 'SymDse_2', 'SymDce_2', 'formF', 'gcount', 'sgridAN
D',
       'sgridTAU', 'sgridWGT', 'aFdP', 'fFdP', 'IoV', 'KLPE', 'AsymI',
'CSI',
       'CVI', 'ARerr', 'histSI', 'MultiFractal_c1', 'MultiFractal_c2',
       'SDLEalpha', 'SDLEmean', 'QSE', 'Hurst.exponent', 'mean', 'medi
an'],
      dtype='object')
```

In [103]:

```
hrv["Sepsis3"].corr(hrv["Sepsis3"])
for i in hrv.columns:
    c=hrv[i].corr(hrv["Sepsis3"])
    print(c)
```

```
-0.02527187721937068
1.0
-0.16339079489084518
0.1677841726215487
0.22365614440654025
0.21044018286202543
-0.07323413923349596
0.08020929373715716
0.057125647101376034
-0.04022189066112259
-0.1225839432399076
-0.04395880818673318
-0.03248454585523562
0.2253782217742344
0.20957850071545472
0.2552208940628494
0.023695335837259166
-0.23334038308966018
-0.17732538232419662
-0.04984913736416616
0.05955896505753054
-0.0015692246137141163
0.032961392744447794
-0.057757892311287716
-0.0864795918199675
0.01647207525861744
0.1104639748136656
-0.014202152841250166
-0.03753856244824713
0.22920153852827524
0.2359670597116792
-0.16131169746197865
0.18964678425672235
0.10120606015202153
-0.15140349293731684
0.18099042437881668
0.13612354757955894
-0.12117184926613388
-0.2580152720895615
-0.26290505085522325
0.16172489536943382
0.16790082520811114
0.3400543647300645
0.3531992587766918
0.1796164525390422
-0.16968590868391098
-0.1183767046913884
-0.12070166527698939
0.2148044585830974
0.11565868773292132
-0.09680280266447794
0.11245506699536517
-0.04091915207988212
```

```
0.050061979673419806
0.04620081878268628
0.17120638583682043
0.17146838911164766
0.1763162886773732
0.1539233794535976
```

In [116]:

```
hrv.columns
```

Out[116]:

```
Index(['SI > 1', 'Sepsis3', 'Mean.rate', 'Coefficient.of.variation',
       'Poincar..SD1', 'Poincar..SD2', 'LF.HF.ratio.LombScargle',
       'LF.Power.LombScargle', 'HF.Power.LombScargle', 'DFA.Alpha.1',
       'DFA.Alpha.2', 'Largest.Lyapunov.exponent', 'Correlation.dimens
ion',
       'Power.Law.Slope.LombScargle', 'Power.Law.Y.Intercept.LombScarg
le',
       'DFA.AUC', 'Multiscale.Entropy', 'VLF.Power.LombScargle', 'Comp
lexity',
       'eScaleE', 'pR', 'pD', 'dlmax', 'sedl', 'pDpR', 'pL', 'vlmax',
'sevl',
       'shannEn', 'PSeo', 'Teo', 'SymDp0_2', 'SymDp1_2', 'SymDp2_2',
       'SymDfw_2', 'SymDse_2', 'SymDce_2', 'formF', 'gcount', 'sgridAN
D',
       'sgridTAU', 'sgridWGT', 'aFdP', 'fFdP', 'IoV', 'KLPE', 'AsymI',
'CSI',
       'CVI', 'ARerr', 'histSI', 'MultiFractal_c1', 'MultiFractal_c2',
       'SDLEalpha', 'SDLEmean', 'QSE', 'Hurst.exponent', 'mean', 'medi
an'],
      dtype='object')
```

K-NEAREST NEIGHBOUR

In [96]:

```
dummies=pd.get_dummies(hrv)
```

In [97]:

```
import cufflinks as cf
```

In [98]:

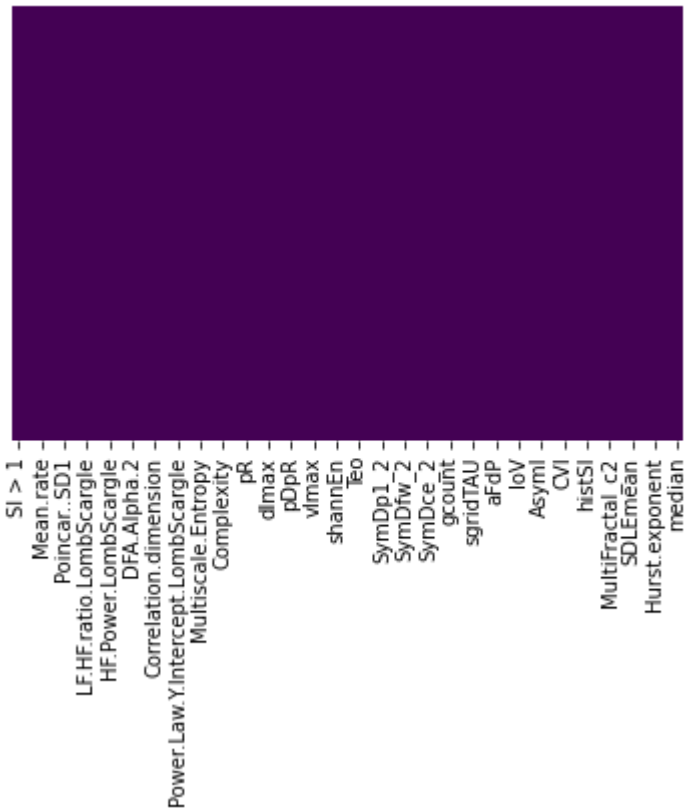
```
cf.go_offline()
```

In [99]:

```
sns.heatmap(hrv.isnull(),yticklabels=False, cbar=False, cmap='viridis')
```

Out[99]:

```
<AxesSubplot:>
```



In [101]:

```
hrv.keys()
```

Out[101]:

```
Index(['SI > 1', 'Sepsis3', 'Mean.rate', 'Coefficient.of.variation',
       'Poincar..SD1', 'Poincar..SD2', 'LF.HF.ratio.LombScargle',
       'LF.Power.LombScargle', 'HF.Power.LombScargle', 'DFA.Alpha.1',
       'DFA.Alpha.2', 'Largest.Lyapunov.exponent', 'Correlation.dimens
ion',
       'Power.Law.Slope.LombScargle', 'Power.Law.Y.Intercept.LombScarg
le',
       'DFA.AUC', 'Multiscale.Entropy', 'VLF.Power.LombScargle', 'Comp
lexity',
       'eScaleE', 'pR', 'pD', 'dlmax', 'sedl', 'pDpR', 'pL', 'vlmax',
'sevl',
       'shannEn', 'PSeo', 'Teo', 'SymDp0_2', 'SymDp1_2', 'SymDp2_2',
       'SymDfw_2', 'SymDse_2', 'SymDce_2', 'formF', 'gcount', 'sgridAN
D',
       'sgridTAU', 'sgridWGT', 'aFdP', 'fFdP', 'IoV', 'KLPE', 'AsymI',
'CSI',
       'CVI', 'ARerr', 'histSI', 'MultiFractal_c1', 'MultiFractal_c2',
       'SDLEalpha', 'SDLEmean', 'QSE', 'Hurst.exponent', 'mean', 'medi
an'],
      dtype='object')
```

In [ ]:

```

```

In [186]:

```
from sklearn.preprocessing import StandardScaler
```

In [187]:

```
scaler= StandardScaler()
```

In [188]:

```
scaler.fit(hrv.drop('Sepsis3',axis=1))
```

Out[188]:

```
StandardScaler()
```

In [189]:

```
hrv['Sepsis3']
```

Out[189]:

```
0       1
1       1
2       1
3       1
4       1
       ..
4309    0
4310    0
4311    0
4312    0
4313    0
Name: Sepsis3, Length: 4314, dtype: int64
```

In [190]:

```
arr= hrv['Sepsis3'].to_numpy()
arr
```

Out[190]:

```
array([1, 1, 1, ..., 0, 0, 0])
```

In [191]:

```
arr.reshape(-1,1)
```

Out[191]:

```
array([[1],
       [1],
       [1],
       ...,
       [0],
       [0],
       [0]])
```

In [195]:

```
from sklearn.model_selection import train_test_split
```

In [196]:

```
x = hrv
y = np.ravel(arr)
x_train,x_test,y_train,y_test = train_test_split(x,y,test_size=0.30,random_state=58)
```

In [197]:

```
x=hrv.iloc[:,2:].values
y=hrv.iloc[:,1:].values
x
```

Out[197]:

```
array([[8.94111600e+01, 1.30259000e-01, 4.53490000e-02, ...,
        2.93162000e-01, 1.46666670e-02, 1.00000000e-02],
       [8.68057600e+01, 2.68240000e-02, 9.50400000e-03, ...,
        1.89769000e-01, 1.84000000e-02, 1.00000000e-02],
       [8.55076700e+01, 3.52030000e-02, 9.71500000e-03, ...,
        8.42615000e-01, 7.56666700e-03, 1.00000000e-02],
       ...,
       [1.14402754e+02, 5.33780000e-02, 9.23800000e-03, ...,
        1.06483000e-01, 6.01333330e-02, 3.00000000e-02],
       [1.20986143e+02, 4.94030000e-02, 1.25270000e-02, ...,
        5.01250000e-02, 6.96666670e-02, 4.00000000e-02],
       [1.09532809e+02, 2.37370000e-02, 7.14200000e-03, ...,
        8.47800000e-01, 2.31666670e-02, 2.00000000e-02]])
```

In [198]:

```
from sklearn.neighbors import KNeighborsClassifier
```

In [199]:

```
knn= KNeighborsClassifier(n_neighbors=1)
```

In [200]:

```
knn.fit(x_train,y_train)
```

Out[200]:

```
KNeighborsClassifier(n_neighbors=1)
```

In [201]:

```
pred=knn.predict(x_test)
```

In [202]:

```
from sklearn.metrics import classification_report,confusion_matrix,accuracy_score
```

In [203]:

```
print(confusion_matrix(y_test,pred))
```

```
[[1074   60]
 [  92   69]]
```

In [204]:

```
print(classification_report(y_test,pred))
```

```
              precision    recall  f1-score   support

           0       0.92      0.95      0.93      1134
           1       0.53      0.43      0.48       161

    accuracy                           0.88      1295
   macro avg       0.73      0.69      0.70      1295
weighted avg       0.87      0.88      0.88      1295
```

In [205]:

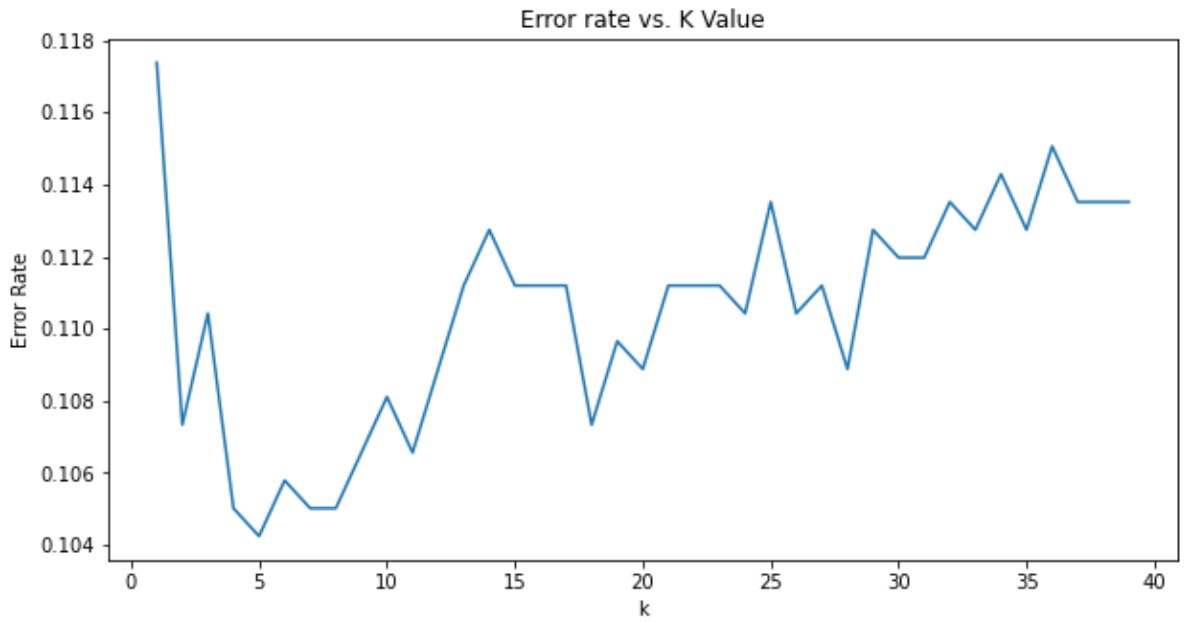
```
error_rate = []

for k in range(1,40):

    knn= KNeighborsClassifier(n_neighbors=k)
    knn.fit(x_train,y_train)
    pred_k = knn.predict(x_test)
    error_rate.append(np.mean(pred_k != y_test))
```

In [206]:

```
plt.figure(figsize=(10,5))
plt.plot(range(1,40),error_rate)
plt.title('Error rate vs. K Value')
plt.xlabel('k')
plt.ylabel('Error Rate')
```

Out[206]:

```
Text(0, 0.5, 'Error Rate')
```



In [207]:

```
knn=KNeighborsClassifier(n_neighbors=38)

knn.fit(x_train,y_train)
pred = knn.predict(x_test)


print('with k=38')
print('\n')
print(confusion_matrix(y_test,pred))
print('\n')
print(classification_report(y_test,pred))
```

```
with k=38


[[1131    3]
 [ 144   17]]


              precision    recall  f1-score   support

           0       0.89      1.00      0.94      1134
           1       0.85      0.11      0.19       161

    accuracy                           0.89      1295
   macro avg       0.87      0.55      0.56      1295
weighted avg       0.88      0.89      0.85      1295
```

In [208]:

```
knn_accuracy = accuracy_score(y_test, pred)
knn_accuracy
```

Out[208]:

```
0.8864864864864865
```

SVM

In [139]:

```
dummies=pd.get_dummies(hrv)
```

In [140]:

```
hrv
```

Out[140]:

	SI > 1	Sepsis3	Mean.rate	Coefficient.of.variation	Poincar..SD1	Poincar..SD2	LF.HF.ratio.LombScargle	LF.Pov
0	0	1	89.411160	0.130259	0.045349	0.115000	1.996490	
1	0	1	86.805760	0.026824	0.009504	0.024437	0.879037	
2	0	1	85.507670	0.035203	0.009715	0.033555	2.542225	
3	0	1	95.127480	0.013182	0.006829	0.009571	1.596207	
4	0	1	93.370410	0.016777	0.006921	0.013585	1.128488	
...	...	...	...	...	...	...	...	
4309	0	0	111.683756	0.031518	0.015239	0.018471	0.495158	
4310	0	0	111.409998	0.048524	0.021231	0.030251	0.468523	
4311	0	0	114.402754	0.053378	0.009238	0.038498	3.641941	

In [141]:

```
hrv.keys()
```

Out[141]:

```
Index(['SI > 1', 'Sepsis3', 'Mean.rate', 'Coefficient.of.variation',
       'Poincar..SD1', 'Poincar..SD2', 'LF.HF.ratio.LombScargle',
       'LF.Power.LombScargle', 'HF.Power.LombScargle', 'DFA.Alpha.1',
       'DFA.Alpha.2', 'Largest.Lyapunov.exponent', 'Correlation.dimens
ion',
       'Power.Law.Slope.LombScargle', 'Power.Law.Y.Intercept.LombScarg
le',
       'DFA.AUC', 'Multiscale.Entropy', 'VLF.Power.LombScargle', 'Comp
lexity',
       'eScaleE', 'pR', 'pD', 'dlmax', 'sedl', 'pDpR', 'pL', 'vlmax',
'sevl',
       'shannEn', 'PSeo', 'Teo', 'SymDp0_2', 'SymDp1_2', 'SymDp2_2',
       'SymDfw_2', 'SymDse_2', 'SymDce_2', 'formF', 'gcount', 'sgridAN
D',
       'sgridTAU', 'sgridWGT', 'aFdP', 'fFdP', 'IoV', 'KLPE', 'AsymI',
'CSI',
       'CVI', 'ARerr', 'histSI', 'MultiFractal_c1', 'MultiFractal_c2',
       'SDLEalpha', 'SDLEmean', 'QSE', 'Hurst.exponent', 'mean', 'medi
an'],
      dtype='object')
```

In [210]:

```
from sklearn.preprocessing import StandardScaler
```

In [211]:

```
scaler= StandardScaler()
```

In [212]:

```
scaler.fit(hrv.drop('Sepsis3',axis=1))
```

Out[212]:

```
StandardScaler()
```

In [213]:

```
hrv['Sepsis3']
```

Out[213]:

```
0       1
1       1
2       1
3       1
4       1
       ..
4309    0
4310    0
4311    0
4312    0
4313    0
Name: Sepsis3, Length: 4314, dtype: int64
```

In [214]:

```
arr = hrv['Sepsis3'].to_numpy()
arr
```

Out[214]:

```
array([1, 1, 1, ..., 0, 0, 0])
```

In [215]:

```
arr.reshape(-1,1)
```

Out[215]:

```
array([[1],
       [1],
       [1],
       ...,
       [0],
       [0],
       [0]])
```

In [216]:

```
from sklearn.model_selection import train_test_split
```

In [217]:

```
x = hrv
y = np.ravel(arr)
x_test,x_train,y_test,y_train = train_test_split(x,y, test_size=0.5, random_state=20
```

In [218]:

```
from sklearn.svm import SVC
```

In [219]:

```
model = SVC()
```

In [220]:

```
model.fit(x_train,y_train)
```

Out[220]:

```
SVC()
```

In [221]:

```
predictions = model.predict(x_test)
```

In [222]:

```
from sklearn.metrics import classification_report,confusion_matrix
```

In [223]:

```
print(confusion_matrix(y_test,predictions))
```

```
[[1897    0]
 [ 260    0]]
```

In [224]:

```
print(classification_report(y_test,predictions))
```

```
              precision    recall  f1-score   support

           0       0.88      1.00      0.94      1897
           1       0.00      0.00      0.00       260

    accuracy                           0.88      2157
   macro avg       0.44      0.50      0.47      2157
weighted avg       0.77      0.88      0.82      2157


/opt/anaconda3/lib/python3.9/site-packages/sklearn/metrics/_classifica
tion.py:1248: UndefinedMetricWarning:

Precision and F-score are ill-defined and being set to 0.0 in labels w
ith no predicted samples. Use `zero_division` parameter to control thi
s behavior.

/opt/anaconda3/lib/python3.9/site-packages/sklearn/metrics/_classifica
tion.py:1248: UndefinedMetricWarning:

Precision and F-score are ill-defined and being set to 0.0 in labels w
ith no predicted samples. Use `zero_division` parameter to control thi
s behavior.

/opt/anaconda3/lib/python3.9/site-packages/sklearn/metrics/_classifica
tion.py:1248: UndefinedMetricWarning:

Precision and F-score are ill-defined and being set to 0.0 in labels w
ith no predicted samples. Use `zero_division` parameter to control thi
s behavior.
```

In [225]:

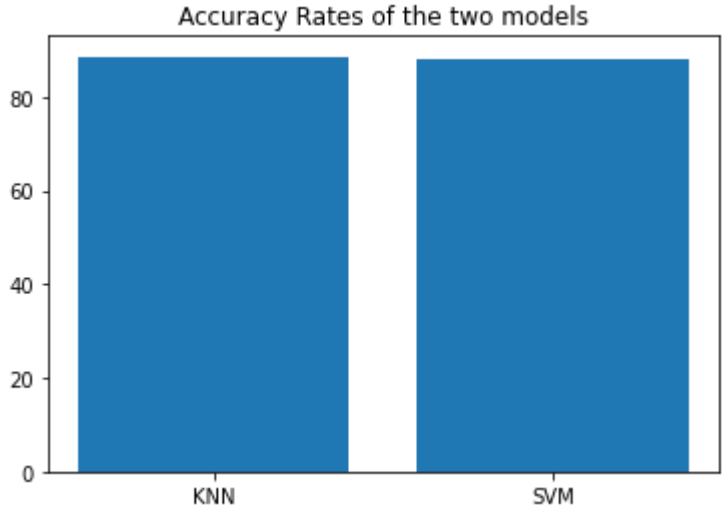
```
svm_accuracy=accuracy_score(y_test, predictions)
svm_accuracy
```

Out[225]:

```
0.8794622160407974
```

In [228]:

```
model = ['KNN', 'SVM']
acc = [88.64864864864865,87.94622160407974]
plt.bar(model,acc)
plt.title("Accuracy Rates of the two models")
plt.show()
```



In [ ]:

```
```

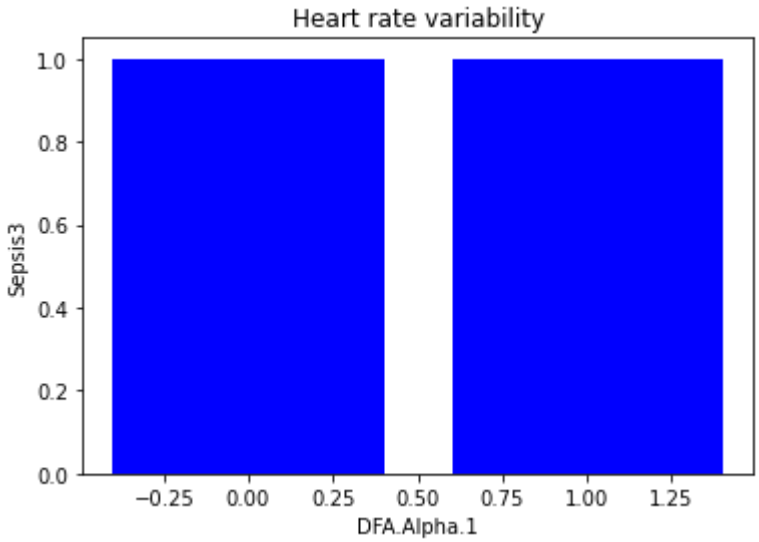
DATA VISUALISATION

In [8]:

```
data = pd.read_csv('/Users/baskaran/Downloads/HRV csv.csv')
```

In [9]:

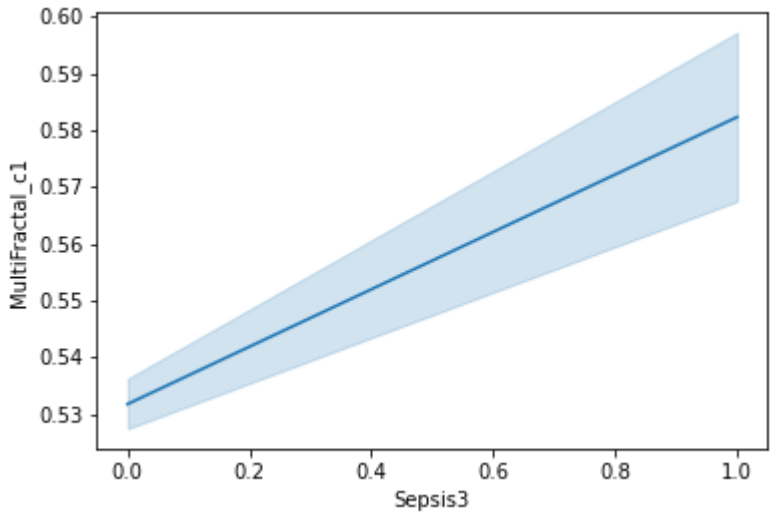
```
df = pd.DataFrame(data)

X = list(df.iloc[:, 0])
Y = list(df.iloc[:, 1])


plt.bar(X, Y, color='b')
plt.title("Heart rate variability")
plt.xlabel("DFA.Alpha.1")
plt.ylabel("Sepsis3")


plt.show()
```



In [ ]:

In [11]:

```
import seaborn
import pandas
import matplotlib.pyplot as plt
res = seaborn.lineplot(x="Sepsis3", y="MultiFractal_c1", data=data)
plt.show()
```



In [8]:

```
import seaborn
import pandas
import matplotlib.pyplot as plt
csv = pandas.read_csv("/Users/baskaran/Downloads/HRV csv.csv")
res = seaborn.violinplot(x=csv['Sepsis3'])
plt.show()
```