HeartFailure Prediction

August 30, 2020

```
[1]: import pandas as pd
     import numpy as np
```

```
[2]: df=pd.read_csv("datasets_727551_1263738_heart_failure_clinical_records_dataset.
      ↪csv")
```

```
[3]: df.shape
```

```
[3]: (299, 13)
```

```
[4]: df
```

```
[4]:       age  anaemia  creatinine_phosphokinase  diabetes  ejection_fraction  \
     0    75.0        0                       582         0                 20
     1    55.0        0                      7861         0                 38
     2    65.0        0                       146         0                 20
     3    50.0        1                       111         0                 20
     4    65.0        1                       160         1                 20
     ..    …        …                       …        …                …
     294  62.0        0                        61         1                 38
     295  55.0        0                      1820         0                 38
     296  45.0        0                      2060         1                 60
     297  45.0        0                      2413         0                 38
     298  50.0        0                       196         0                 45

          high_blood_pressure  platelets  serum_creatinine  serum_sodium  sex  \
     0                      1  265000.00               1.9           130    1
     1                      0  263358.03               1.1           136    1
     2                      0  162000.00               1.3           129    1
     3                      0  210000.00               1.9           137    1
     4                      0  327000.00               2.7           116    0
     ..                   …        …               …           …  …
     294                    1  155000.00               1.1           143    1
     295                    0  270000.00               1.2           139    0
     296                    0  742000.00               0.8           138    0
     297                    0  140000.00               1.4           140    1
     298                    0  395000.00               1.6           136    1
          smoking  time  DEATH_EVENT
     0          0     4            1
     1          0     6            1
     2          1     7            1
     3          0     7            1
     4          0     8            1
     ..       …   …          …
     294        1   270            0
     295        0   271            0
     296        0   278            0
     297        1   280            0
     298        1   285            0

     [299 rows x 13 columns]
```

[5]:
```
df.describe()
```

[5]:
```
              age     anaemia  creatinine_phosphokinase    diabetes  \
count  299.000000  299.000000                299.000000  299.000000
mean    60.833893    0.431438                581.839465    0.418060
std     11.894809    0.496107                970.287881    0.494067
min     40.000000    0.000000                 23.000000    0.000000
25%     51.000000    0.000000                116.500000    0.000000
50%     60.000000    0.000000                250.000000    0.000000
75%     70.000000    1.000000                582.000000    1.000000
max     95.000000    1.000000               7861.000000    1.000000

       ejection_fraction  high_blood_pressure      platelets  \
count         299.000000           299.000000     299.000000
mean           38.083612             0.351171  263358.029264
std            11.834841             0.478136   97804.236869
min            14.000000             0.000000   25100.000000
25%            30.000000             0.000000  212500.000000
50%            38.000000             0.000000  262000.000000
75%            45.000000             1.000000  303500.000000
max            80.000000             1.000000  850000.000000

       serum_creatinine  serum_sodium         sex    smoking        time  \
count         299.00000    299.000000  299.000000  299.00000  299.000000
mean            1.39388    136.625418    0.648829    0.32107  130.260870
std             1.03451      4.412477    0.478136    0.46767   77.614208
min             0.50000    113.000000    0.000000    0.00000    4.000000
25%             0.90000    134.000000    0.000000    0.00000   73.000000
50%             1.10000    137.000000    1.000000    0.00000  115.000000
75%             1.40000    140.000000    1.000000    1.00000  203.000000
max             9.40000    148.000000    1.000000    1.00000  285.000000
```

```
       DEATH_EVENT
count    299.00000
mean       0.32107
std        0.46767
min        0.00000
25%        0.00000
50%        0.00000
75%        1.00000
max        1.00000
```

[6]:
```
df.isnull().sum()
```

[6]:
```
age                         0
anaemia                     0
creatinine_phosphokinase    0
diabetes                    0
ejection_fraction           0
high_blood_pressure         0
platelets                   0
serum_creatinine            0
serum_sodium                0
sex                         0
smoking                     0
time                        0
DEATH_EVENT                 0
dtype: int64
```

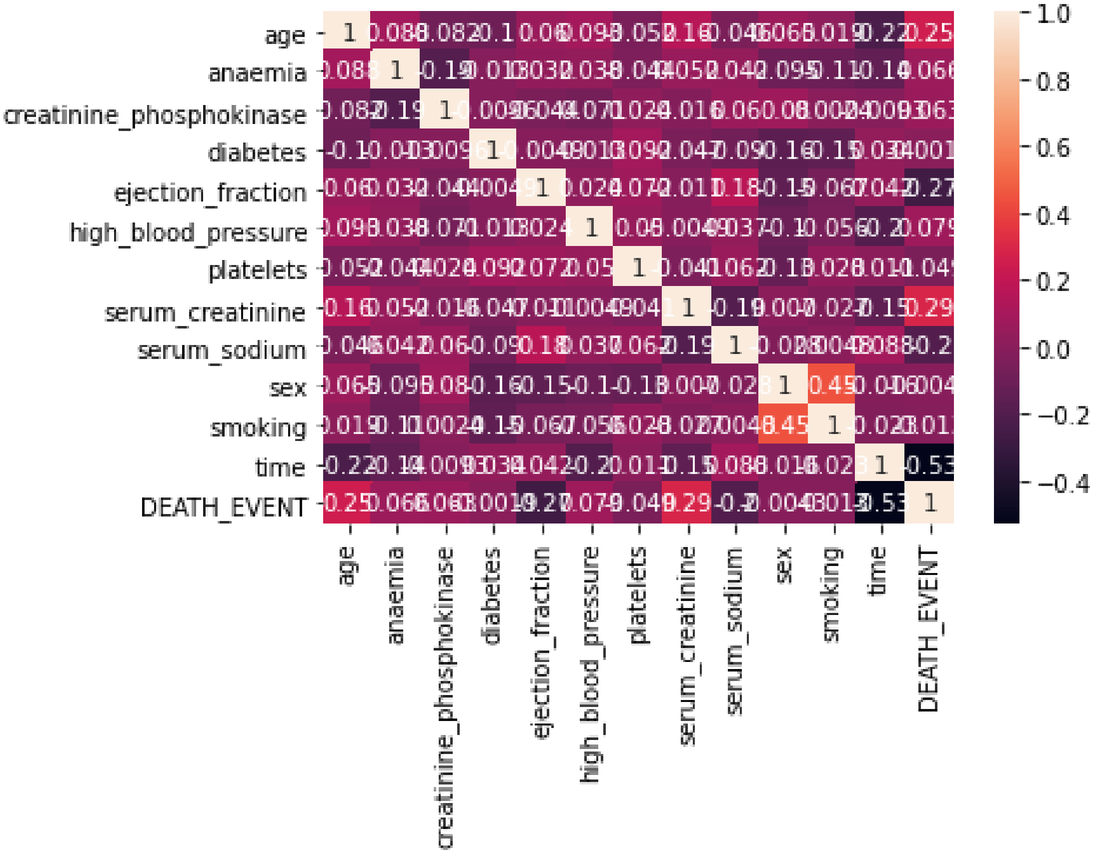
[7]:
```
df.info()
```

```
<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
```

[8]:
```
import matplotlib.pyplot as plt
import seaborn as sns
```

[9]:
```
sns.heatmap(df.corr(),annot=True)
```

[9]:
```
<AxesSubplot:>
```



[10]:
```
df.corr()
```

[10]:
```
                               age   anaemia  creatinine_phosphokinase  \
age                       1.000000  0.088006                 -0.081584
anaemia                   0.088006  1.000000                 -0.190741
creatinine_phosphokinase -0.081584 -0.190741                  1.000000
diabetes                 -0.101012 -0.012729                 -0.009639
ejection_fraction         0.060098  0.031557                 -0.044080
high_blood_pressure       0.093289  0.038182                 -0.070590
```

```
platelets                -0.052354 -0.043786                  0.024463
serum_creatinine          0.159187  0.052174                 -0.016408
serum_sodium             -0.045966  0.041882                  0.059550
sex                       0.065430 -0.094769                  0.079791
smoking                   0.018668 -0.107290                  0.002421
time                     -0.224068 -0.141414                 -0.009346
DEATH_EVENT               0.253729  0.066270                  0.062728

                          diabetes  ejection_fraction  high_blood_pressure  \
age                      -0.101012           0.060098             0.093289
anaemia                  -0.012729           0.031557             0.038182
creatinine_phosphokinase -0.009639          -0.044080            -0.070590
diabetes                  1.000000          -0.004850            -0.012732
ejection_fraction        -0.004850           1.000000             0.024445
high_blood_pressure      -0.012732           0.024445             1.000000
platelets                 0.092193           0.072177             0.049963
serum_creatinine         -0.046975          -0.011302            -0.004935
serum_sodium             -0.089551           0.175902             0.037109
sex                      -0.157730          -0.148386            -0.104615
smoking                  -0.147173          -0.067315            -0.055711
time                      0.033726           0.041729            -0.196439
DEATH_EVENT              -0.001943          -0.268603             0.079351

                          platelets  serum_creatinine  serum_sodium       sex  \
age                       -0.052354          0.159187     -0.045966  0.065430
anaemia                   -0.043786          0.052174      0.041882 -0.094769
creatinine_phosphokinase   0.024463         -0.016408      0.059550  0.079791
diabetes                   0.092193         -0.046975     -0.089551 -0.157730
ejection_fraction          0.072177         -0.011302      0.175902 -0.148386
high_blood_pressure        0.049963         -0.004935      0.037109 -0.104615
platelets                  1.000000         -0.041198      0.062125 -0.125120
serum_creatinine          -0.041198          1.000000     -0.189095  0.006970
serum_sodium               0.062125         -0.189095      1.000000 -0.027566
sex                       -0.125120          0.006970     -0.027566  1.000000
smoking                    0.028234         -0.027414      0.004813  0.445892
time                       0.010514         -0.149315      0.087640 -0.015608
DEATH_EVENT               -0.049139          0.294278     -0.195204 -0.004316

                           smoking      time  DEATH_EVENT
age                       0.018668 -0.224068     0.253729
anaemia                  -0.107290 -0.141414     0.066270
creatinine_phosphokinase  0.002421 -0.009346     0.062728
diabetes                 -0.147173  0.033726    -0.001943
ejection_fraction        -0.067315  0.041729    -0.268603
high_blood_pressure      -0.055711 -0.196439     0.079351
platelets                 0.028234  0.010514    -0.049139
serum_creatinine         -0.027414 -0.149315     0.294278
```

```
serum_sodium               0.004813  0.087640    -0.195204
sex                        0.445892 -0.015608    -0.004316
smoking                    1.000000 -0.022839    -0.012623
time                      -0.022839  1.000000    -0.526964
DEATH_EVENT               -0.012623 -0.526964     1.000000
```

```
[11]: X=df.drop('DEATH_EVENT',axis=1)
```

```
[12]: y=df['DEATH_EVENT']
```

```
[13]: from sklearn.model_selection import train_test_split
```

```
[14]: X_train,X_test,y_train,y_test=train_test_split(X,y,test_size=0.
       ↪20,random_state=42)
```

```
[15]: print(X_train.shape)
      print(X_test.shape)
      print(y_train.shape)
      print(y_test.shape)
```

```
      (239, 12)
      (60, 12)
      (239,)
      (60,)
```

```
[16]: from sklearn.linear_model import LogisticRegression
```

```
[17]: lrg=LogisticRegression()
```

```
[18]: lrg.fit(X_train,y_train)
```

```
[18]: LogisticRegression(C=1.0, class_weight=None, dual=False, fit_intercept=True,
                         intercept_scaling=1, l1_ratio=None, max_iter=100,
                         multi_class='auto', n_jobs=None, penalty='l2',
                         random_state=None, solver='lbfgs', tol=0.0001, verbose=0,
                         warm_start=False)
```

```
[19]: y_pred=lrg.predict(X_test)
      y_pred
```

```
[19]: array([0, 0, 0, 1, 0, 0, 1, 0, 1, 0, 0, 0, 0, 0, 1, 1, 0, 0, 0, 1, 0, 0,
             0, 1, 1, 0, 0, 0, 0, 0, 1, 0, 1, 1, 0, 0, 0, 0, 0, 0, 1, 1, 0, 0,
             0, 0, 0, 0, 0, 0, 1, 0, 1, 0, 0, 0, 0, 1, 0, 1])
```

```
[20]: from sklearn.metrics import accuracy_score
```

```
[21]: acc_lrg=accuracy_score(y_pred=y_pred,y_true=y_test)*100
      acc_lrg                                                    ## Acc_score␣
       ↪based on Logistic Regression Classifier
```

```
[21]: 80.0
```

```
[22]: from sklearn.metrics import confusion_matrix
```

```
[23]: confusion_matrix(y_pred=y_pred,y_true=y_test)
```

```
[23]: array([[33,  2],
             [10, 15]])
```

```
[24]: from sklearn.neighbors import KNeighborsClassifier
```

```
[25]: knn=KNeighborsClassifier(n_neighbors=5)
```

```
[26]: knn.fit(X_train,y_train)
```

```
[26]: KNeighborsClassifier(algorithm='auto', leaf_size=30, metric='minkowski',
                           metric_params=None, n_jobs=None, n_neighbors=5, p=2,
                           weights='uniform')
```

```
[27]: y_pred=knn.predict(X_test)
      y_pred
```

```
[27]: array([1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0,
             0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 1,
             0, 0, 0, 0, 0, 1, 0, 1, 0, 0, 1, 0, 0, 0, 0, 0])
```

```
[28]: confusion_matrix(y_pred=y_pred,y_true=y_test)
```

```
[28]: array([[30,  5],
             [23,  2]])
```

```
[29]: y.value_counts()
```

```
[29]: 0    203
      1     96
      Name: DEATH_EVENT, dtype: int64
```

```
[30]: acc_knn=accuracy_score(y_pred=y_pred,y_true=y_test)*100
      acc_knn
```

```
[30]: 53.333333333333336
```

```
[31]: from sklearn.naive_bayes import MultinomialNB
```

[32]:
```
clf=MultinomialNB()
```

[33]:
```
clf.fit(X_train,y_train)
```

[33]:
```
MultinomialNB(alpha=1.0, class_prior=None, fit_prior=True)
```

[34]:
```
y_pred=clf.predict(X_test)
```

[35]:
```
from sklearn.metrics import accuracy_score
```

[38]:
```
acc_Naive_Bayes=accuracy_score(y_pred=y_pred,y_true=y_test)*100
acc_Naive_Bayes
```

[38]:
```
71.66666666666667
```

[39]:
```
from sklearn.tree import DecisionTreeClassifier
```

[40]:
```
Dec_tree=DecisionTreeClassifier()
```

[41]:
```
Dec_tree.fit(X_train,y_train)
```

[41]:
```
DecisionTreeClassifier(ccp_alpha=0.0, class_weight=None, criterion='gini',
                       max_depth=None, max_features=None, max_leaf_nodes=None,
                       min_impurity_decrease=0.0, min_impurity_split=None,
                       min_samples_leaf=1, min_samples_split=2,
                       min_weight_fraction_leaf=0.0, presort='deprecated',
                       random_state=None, splitter='best')
```

[42]:
```
y_pred=Dec_tree.predict(X_test)
```

[43]:
```
from sklearn.metrics import accuracy_score
```

[47]:
```
acc_Dec_Tree=accuracy_score(y_pred=y_pred,y_true=y_test)*100
acc_Dec_Tree
```

[47]:
```
68.33333333333333
```

[53]:
```
models=pd.DataFrame({'Model':['Logistic␣
 ↪Regression','KNN','Naive-Bayes','Decision Tree'],
                     'Score':[acc_lrg,acc_knn,acc_Naive_Bayes,acc_Dec_Tree]})
models.sort_values(by='Score')
```

[53]:
```
                 Model      Score
1                  KNN  53.333333
3        Decision Tree  68.333333
2          Naive-Bayes  71.666667
0  Logistic Regression  80.000000
```

[ ]:

```
```