

Major Project

K N MONISH KUMAR PRESIDENCY UNIVERSITY BENGALURU 3rd Year

Code:

```
import numpy as np
import pandas as pd
from sklearn.model_selection import train_test_split
from sklearn.metrics import accuracy_score
import matplotlib.pyplot as plt
import seaborn as sns
from sklearn.neighbors import KNeighborsClassifier
from sklearn.tree import DecisionTreeClassifier
```

df=pd.read_csv('https://raw.githubusercontent.com/knmonishgh/dataset/main/heart_failure_cl
inical_records_dataset.csv')
df

	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
294	62.0	0	61	1	38	1	155000.00	1.1	143	1	1	270
295	55.0	0	1820	0	38	0	270000.00	1.2	139	0	0	271
296	45.0	0	2060	1	60	0	742000.00	0.8	138	0	0	278
297	45.0	0	2413	0	38	0	140000.00	1.4	140	1	1	280
298	50.0	0	196	0	45	0	395000.00	1.6	136	1	1	285
299 rows × 13 columns												

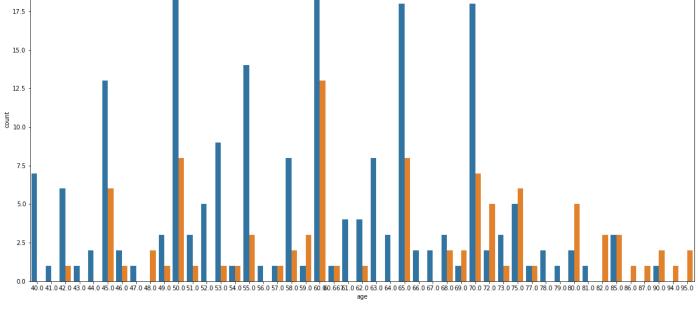
df.describe()

	age	anaemia	creatinine_phosphokinase	diabetes	ejection_fraction	high_blood_pressure	platelets	serum_creatinine	serum_sodium	
count	299.000000	299.000000	299.000000	299.000000	299.000000	299.000000	299.000000	299.00000	299.000000	299.0
mean	60.833893	0.431438	581.839465	0.418060	38.083612	0.351171	263358.029264	1.39388	136.625418	0.6
std	11.894809	0.496107	970.287881	0.494067	11.834841	0.478136	97804.236869	1.03451	4.412477	0.4
min	40.000000	0.000000	23.000000	0.000000	14.000000	0.000000	25100.000000	0.50000	113.000000	0.0
25%	51.000000	0.000000	116.500000	0.000000	30.000000	0.000000	212500.000000	0.90000	134.000000	0.0
50%	60.000000	0.000000	250.000000	0.000000	38.000000	0.000000	262000.000000	1.10000	137.000000	1.0
75%	70.000000	1.000000	582.000000	1.000000	45.000000	1.000000	303500.000000	1.40000	140.000000	1.0
max	95.000000	1.000000	7861.000000	1.000000	80.000000	1.000000	850000.000000	9.40000	148.000000	1.0

df.hist(figsize = (20,20))plt.show() anaemia creatinine_phosphokinase diabetes age 250 -0 -0 -0 -0.2 0.8 0.2 0.4 0.8 0.4 0.6 0.6 0.0 0.0 ejection_fraction high_blood_pressure platelets serum_creatinine 0 -0 -0.0 400000 600000 800000 0.2 0.4 0.8 0.6 serum_sodium sex smoking time 0 -0.2 0.4 0.8 0.2 0.4 0.6 0.8 DEATH_EVENT 0.2 0.4 0.6

plt.figure(figsize=(20,20)) p=sns.heatmap(df.corr(), annot=True,cmap='RdYlGn',square=True) 0.088 -0.082 -0.1 0.06 0.093 -0.052 0.16 -0.046 0.065 0.019 0.25 age - 0.8 -0.013 0.032 0.038 0.052 -0.095 -0.11 0.066 -0.082 -0.0096 -0.044 -0.071 0.024 -0.016 0.06 0.08 0.0024 -0.0093 0.063 creatinine_phosphokinase - 0.6 -0.0049 -0.013 -0.0096 -0.013 0.092 -0.047 -0.09 0.034 -0.0019 diabetes -0.1 -0.044 0.024 0.072 -0.011 0.18 -0.067 0.042 0.032 -0.0049 0.06 ejection fraction - 0.4 0.038 -0.071 0.05 -0.1 0.079 0.093 -0.013 0.024 -0.0049 0.037 -0.056 high_blood_pressure --0.052 0.072 0.05 -0.041 -0.13 0.028 0.011 -0.049 platelets - 0.2 serum_creatinine 0.16 0.052 -0.016 -0.047 -0.011 -0.0049 -0.041 0.007 -0.027 0.29 -0.046 0.042 0.06 -0.09 0.18 0.037 0.062 -0.028 0.0048 0.088 serum_sodium - 0.0 -0.13 0.45 -0.016 -0.095 0.08 -0.1 0.007 -0.028 sex 0.065 -0.0043 -0.11 0.0024 -0.056 0.028 -0.027 0.45 -0.023 -0.013 smoking 0.019 -0.067 0.0048 - -0.2 -0.0093 0.034 0.011 0.088 -0.016 -0.023 DEATH_EVENT --0.0043 -0.013 - -0.4 diabetes š smoking age ejection_fraction serum_sodium time DEATH_EVENT creatinine_phosphokinase serum_creatinine high blood pressure platelets





```
X = np.array(df.drop(['DEATH_EVENT'], axis=1))
y = np.array(df['DEATH_EVENT'])
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=0)
print(X_train.shape)
print(X_test.shape)
 (239, 12)
 (60, 12)
#using Decision tree
DT clf = DecisionTreeClassifier()
DT_clf.fit(X_train, y_train)
y_pred = DT_clf.predict(X_test)
y_pred
array([0, 0, 0, 0, 0, 0, 1, 0, 1, 0, 0, 1, 0, 0, 1, 1, 0, 0, 0, 1, 1,
        0, 0, 1, 0, 1, 0, 0, 0, 0, 1, 1, 0, 0, 0, 0, 0, 0, 1, 0, 1, 0, 1,
        1, 0, 0, 1, 0, 1, 1, 0, 0, 0, 1, 1, 0, 1, 0, 0], dtype=int64)
DT_clf.predict([[68,1,220,0,35,1,289000,0.9,140,1,1,20]])
 array([1], dtype=int64)
KNN_clf = KNeighborsClassifier()
KNN_clf.fit(X_train, y_train)
y_pred = KNN_clf.predict(X_test)
KNN_clf.predict([[60,0,2656,1,30,0,305000,2.3,137,1,0,30]])
array([0], dtype=int64)
```