

RINEX

MAJOR PROJECT

PRESIDENCY UNIVERSITY BENGALURU

- Yashaswini M
B.Tech

```
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/BCKeerthi/Dataset/main/heart_fa
ilure_clinical_records_dataset.csv')
df
```

	age	anaemia	creatinine_phosphokinase	diabetes	ejection_fraction	high_blood_pressure	platelets	serum_creatinine	serum_sodium	sex
0	75.0	0	582	0	20	1	265000.00	1.9	130	1
1	55.0	0	7861	0	38	0	263358.03	1.1	136	1
2	65.0	0	146	0	20	0	162000.00	1.3	129	1
3	50.0	1	111	0	20	0	210000.00	1.9	137	1
4	65.0	1	160	1	20	0	327000.00	2.7	116	0
...
294	62.0	0	61	1	38	1	155000.00	1.1	143	1
295	55.0	0	1820	0	38	0	270000.00	1.2	139	0
296	45.0	0	2060	1	60	0	742000.00	0.8	138	0
297	45.0	0	2413	0	38	0	140000.00	1.4	140	1
298	50.0	0	196	0	45	0	395000.00	1.6	136	1

299 rows x 13 columns

```
df.info
```

Output exceeds the [size limit](#). Open the full output [data in a text editor](#)

```
<bound method DataFrame.info of
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
...
296         0  278          0
297         1  280          0
298         1  285          0

```

```
df.describe
```

Output exceeds the [size limit](#). Open the full output data [in a text editor](#)

```

<bound method NDFrame.describe of
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
...
296         0  278          0
297         1  280          0
298         1  285          0

```

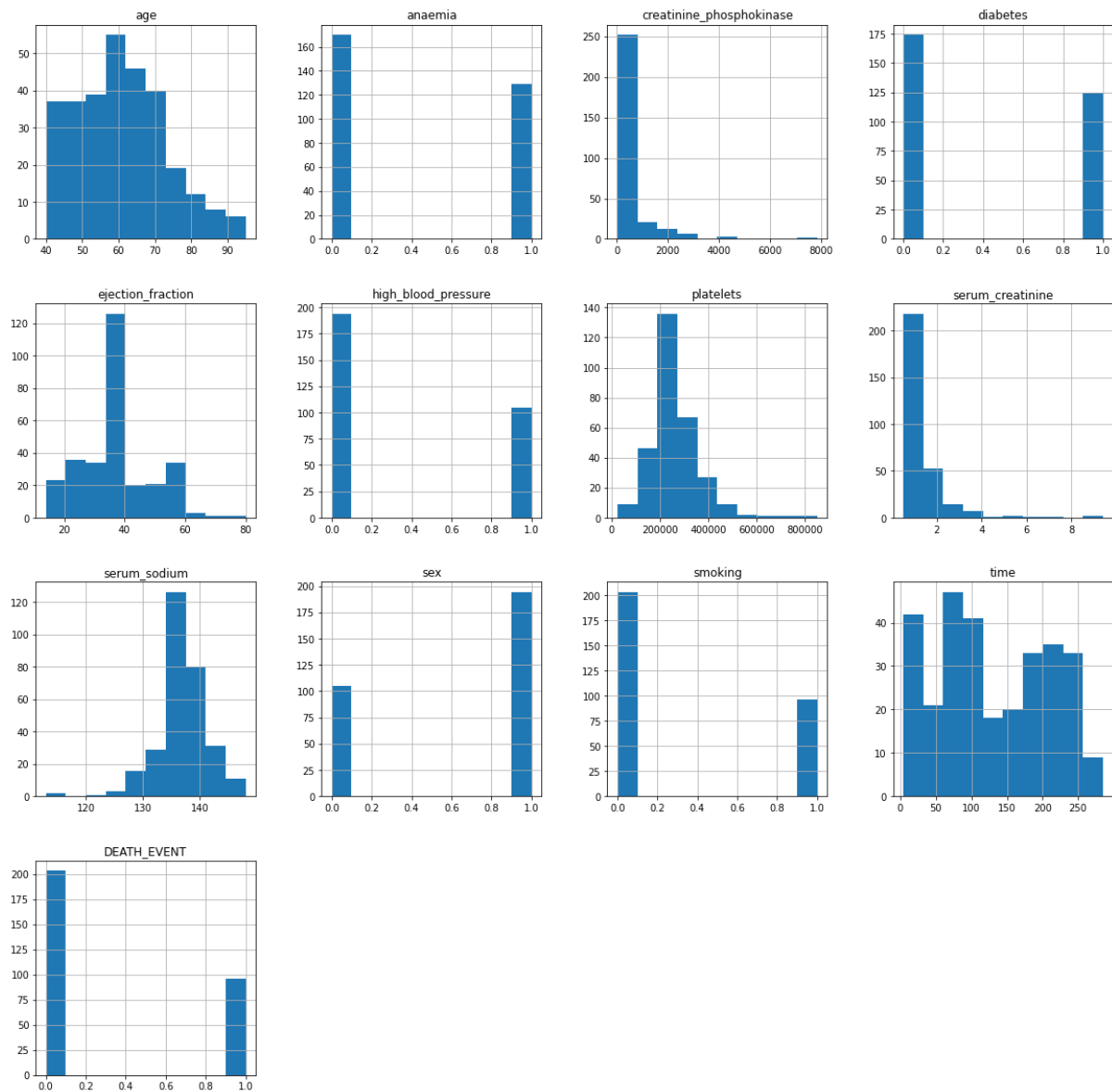
```
df.size
```

```
3887
```

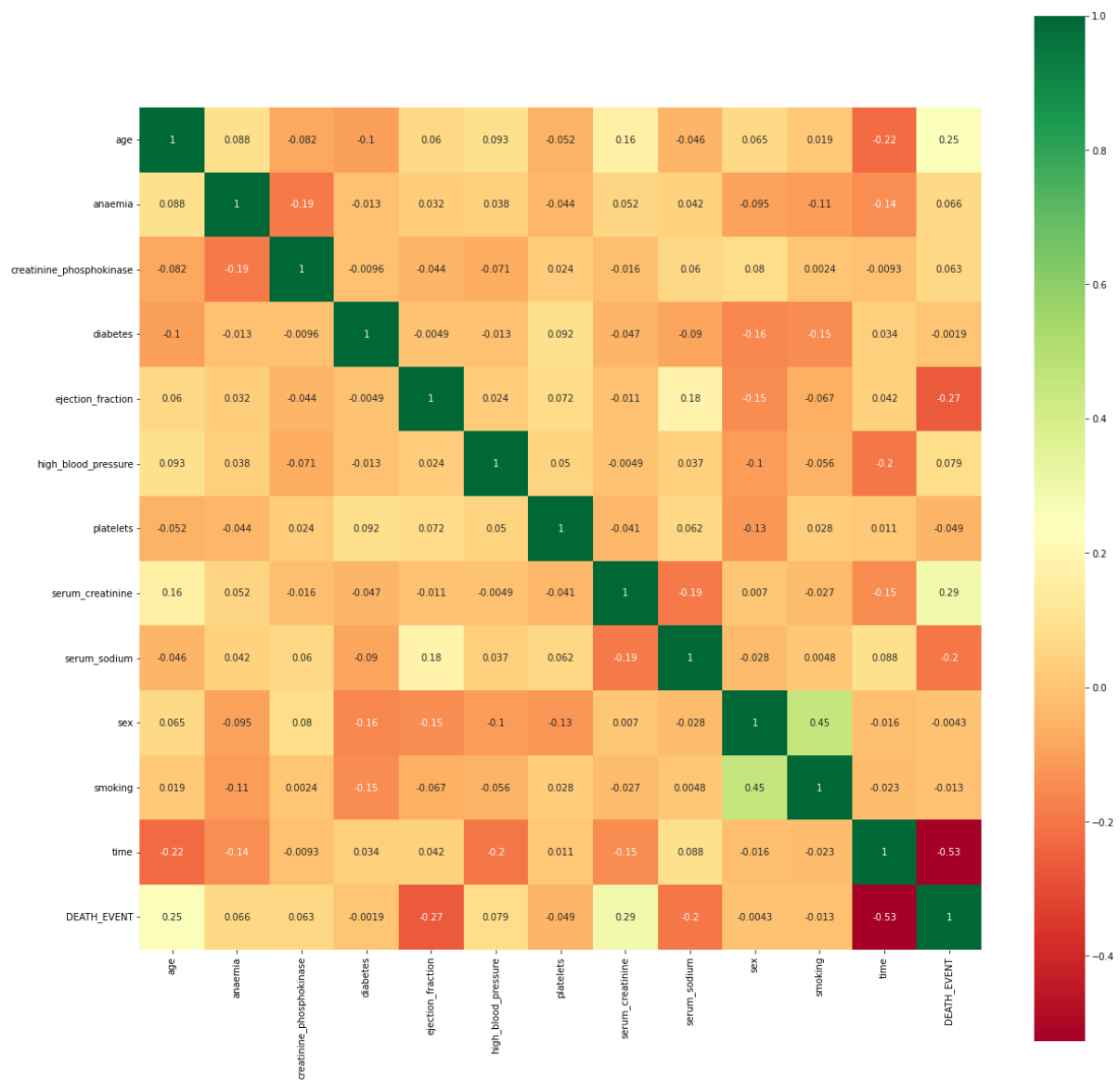
```
df.shape
```

```
(299, 13)
```

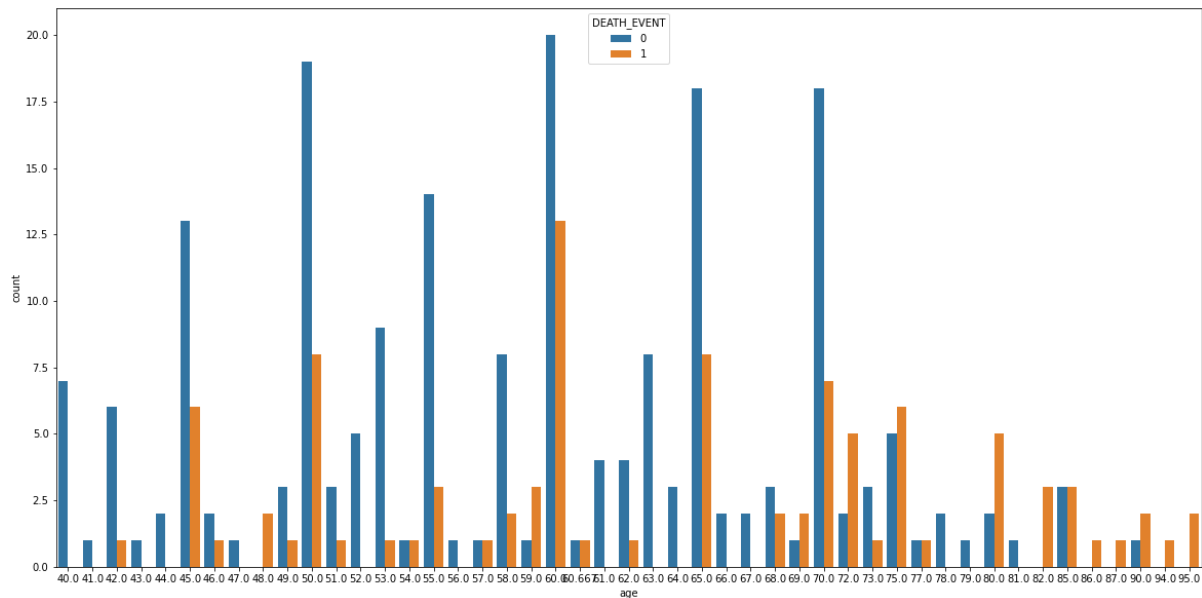
```
df.hist(figsize = (20,20))  
plt.show()
```



```
plt.figure(figsize=(20,20))
p=sns.heatmap(df.corr(), annot=True,cmap='RdYlGn',square=True)
```



```
plt.figure(figsize=(20,10))
sns.countplot(x="age", data=df, hue="DEATH_EVENT");
```



```
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)
```

```
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, 1, 0, 1, 1, 0, 0, 0, 0, 1, 1, 0, 0,
1, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 1, 0, 1, 0, 1, 1, 0, 0, 0, 0, 1,
0, 0, 1, 0, 1, 1, 0, 0, 0, 0], dtype=int64)
```

```
DT_clf.predict([[55,0,7861,0,38,0,263358.03,1.1,136,1,0,6]])
```

```
array([1], dtype=int64)
```

```
KNN_clf = KNeighborsClassifier()  
KNN_clf.fit(X_train, y_train)  
y_pred = KNN_clf.predict(X_test)
```

```
KNN_clf.predict([[75,0,582,1,30,1,263358.03,1.83,134,0,0,23]])
```

```
array([1], dtype=int64)
```