

```
#INSTALLING ELI5
pip install eli5
```

```
Collecting eli5
  Downloading eli5-0.13.0.tar.gz (216 kB)
    216.2/216.2 kB 5.8 MB/s eta 0:00:00
  Preparing metadata (setup.py) ... done
Requirement already satisfied: attrs>17.1.0 in /usr/local/lib/python3.10/dist-packages (from eli5) (23.1.0)
Requirement already satisfied: Jinja2>=3.0.0 in /usr/local/lib/python3.10/dist-packages (from eli5) (3.1.2)
Requirement already satisfied: numpy>=1.9.0 in /usr/local/lib/python3.10/dist-packages (from eli5) (1.23.5)
Requirement already satisfied: scipy in /usr/local/lib/python3.10/dist-packages (from eli5) (1.11.3)
Requirement already satisfied: six in /usr/local/lib/python3.10/dist-packages (from eli5) (1.16.0)
Requirement already satisfied: scikit-learn>=0.20 in /usr/local/lib/python3.10/dist-packages (from eli5) (1.2.2)
Requirement already satisfied: graphviz in /usr/local/lib/python3.10/dist-packages (from eli5) (0.20.1)
Requirement already satisfied: tabulate>=0.7.7 in /usr/local/lib/python3.10/dist-packages (from eli5) (0.9.0)
Requirement already satisfied: MarkupSafe>=2.0 in /usr/local/lib/python3.10/dist-packages (from Jinja2>=3.0.0->eli5) (2.1.3)
Requirement already satisfied: joblib>=1.1.1 in /usr/local/lib/python3.10/dist-packages (from scikit-learn>=0.20->eli5) (1.3.2)
Requirement already satisfied: threadpoolctl>=2.0.0 in /usr/local/lib/python3.10/dist-packages (from scikit-learn>=0.20->eli5) (3.2.0)
Building wheels for collected packages: eli5
  Building wheel for eli5 (setup.py) ... done
  Created wheel for eli5: filename=eli5-0.13.0-py2.py3-none-any.whl size=107719 sha256=48a1bbae3a5d53f18d60f22dce8419283b7b4ee0100119f77
  Stored in directory: /root/.cache/pip/wheels/b8/58/ef/2cf4c306898c2338d51540e0922c8e0d6028e07007085c0004
Successfully built eli5
Installing collected packages: eli5
Successfully installed eli5-0.13.0
```

```
#INSTALLING LIME
pip install lime
```

```
Collecting lime
  Downloading lime-0.2.0.1.tar.gz (275 kB)
    275.7/275.7 kB 9.3 MB/s eta 0:00:00
  Preparing metadata (setup.py) ... done
Requirement already satisfied: matplotlib in /usr/local/lib/python3.10/dist-packages (from lime) (3.7.1)
Requirement already satisfied: numpy in /usr/local/lib/python3.10/dist-packages (from lime) (1.23.5)
Requirement already satisfied: scipy in /usr/local/lib/python3.10/dist-packages (from lime) (1.11.3)
Requirement already satisfied: tqdm in /usr/local/lib/python3.10/dist-packages (from lime) (4.66.1)
Requirement already satisfied: scikit-learn>=0.18 in /usr/local/lib/python3.10/dist-packages (from lime) (1.2.2)
Requirement already satisfied: scikit-image>=0.12 in /usr/local/lib/python3.10/dist-packages (from lime) (0.19.3)
Requirement already satisfied: networkx>=2.2 in /usr/local/lib/python3.10/dist-packages (from scikit-image>=0.12->lime) (3.2.1)
Requirement already satisfied: pillow!=7.1.0,!=7.1.1,!=8.3.0,>=6.1.0 in /usr/local/lib/python3.10/dist-packages (from scikit-image>=0.12->lime) (9.4.0)
Requirement already satisfied: imageio>=2.4.1 in /usr/local/lib/python3.10/dist-packages (from scikit-image>=0.12->lime) (2.31.6)
Requirement already satisfied: tifffile>=2019.7.26 in /usr/local/lib/python3.10/dist-packages (from scikit-image>=0.12->lime) (2023.9.26)
Requirement already satisfied: PyWavelets>=1.1.1 in /usr/local/lib/python3.10/dist-packages (from scikit-image>=0.12->lime) (1.4.1)
Requirement already satisfied: packaging>=20.0 in /usr/local/lib/python3.10/dist-packages (from scikit-image>=0.12->lime) (23.2)
Requirement already satisfied: joblib>=1.1.1 in /usr/local/lib/python3.10/dist-packages (from scikit-learn>=0.18->lime) (1.3.2)
Requirement already satisfied: threadpoolctl>=2.0.0 in /usr/local/lib/python3.10/dist-packages (from scikit-learn>=0.18->lime) (3.2.0)
Requirement already satisfied: contourpy>=1.0.1 in /usr/local/lib/python3.10/dist-packages (from matplotlib->lime) (1.2.0)
Requirement already satisfied: cycler>=0.10 in /usr/local/lib/python3.10/dist-packages (from matplotlib->lime) (0.12.1)
Requirement already satisfied: fonttools>=4.22.0 in /usr/local/lib/python3.10/dist-packages (from matplotlib->lime) (4.44.3)
Requirement already satisfied: kiwisolver>=1.0.1 in /usr/local/lib/python3.10/dist-packages (from matplotlib->lime) (1.4.5)
Requirement already satisfied: pyparsing>=2.3.1 in /usr/local/lib/python3.10/dist-packages (from matplotlib->lime) (3.1.1)
Requirement already satisfied: python-dateutil>=2.7 in /usr/local/lib/python3.10/dist-packages (from matplotlib->lime) (2.8.2)
Requirement already satisfied: six>=1.5 in /usr/local/lib/python3.10/dist-packages (from python-dateutil>=2.7->matplotlib->lime) (1.16.0)
Building wheels for collected packages: lime
  Building wheel for lime (setup.py) ... done
  Created wheel for lime: filename=lime-0.2.0.1-py3-none-any.whl size=283834 sha256=482fa731ae72ec020ac39c4d03a03a8d40f6076f5a812f469a5
  Stored in directory: /root/.cache/pip/wheels/fd/a2/af/9ac0a1a85a27f314a06b39e1f492bee1547d52549a4606ed89
Successfully built lime
Installing collected packages: lime
Successfully installed lime-0.2.0.1
```

```
#INSTALLING SHAP
pip install shap
```

```
Collecting shap
  Downloading shap-0.43.0-cp310-cp310-manylinux_2_12_x86_64.manylinux2010_x86_64.manylinux_2_17_x86_64.manylinux2014_x86_64.whl (532 kB)
    532.9/532.9 kB 4.1 MB/s eta 0:00:00
Requirement already satisfied: numpy in /usr/local/lib/python3.10/dist-packages (from shap) (1.23.5)
Requirement already satisfied: scipy in /usr/local/lib/python3.10/dist-packages (from shap) (1.11.3)
Requirement already satisfied: scikit-learn in /usr/local/lib/python3.10/dist-packages (from shap) (1.2.2)
Requirement already satisfied: pandas in /usr/local/lib/python3.10/dist-packages (from shap) (1.5.3)
Requirement already satisfied: tqdm>=4.27.0 in /usr/local/lib/python3.10/dist-packages (from shap) (4.66.1)
Requirement already satisfied: packaging>20.9 in /usr/local/lib/python3.10/dist-packages (from shap) (23.2)
Collecting slicer==0.0.7 (from shap)
  Downloading slicer-0.0.7-py3-none-any.whl (14 kB)
```

```
Requirement already satisfied: numba in /usr/local/lib/python3.10/dist-packages (from shap) (0.58.1)
Requirement already satisfied: cloudpickle in /usr/local/lib/python3.10/dist-packages (from shap) (2.2.1)
Requirement already satisfied: llvmlite<0.42,>=0.41.0dev0 in /usr/local/lib/python3.10/dist-packages (from numba->shap) (0.41.1)
Requirement already satisfied: python-dateutil>=2.8.1 in /usr/local/lib/python3.10/dist-packages (from pandas->shap) (2.8.2)
Requirement already satisfied: pytz>=2020.1 in /usr/local/lib/python3.10/dist-packages (from pandas->shap) (2023.3.post1)
Requirement already satisfied: joblib>=1.1.1 in /usr/local/lib/python3.10/dist-packages (from scikit-learn->shap) (1.3.2)
Requirement already satisfied: threadpoolctl>=2.0.0 in /usr/local/lib/python3.10/dist-packages (from scikit-learn->shap) (3.2.0)
Requirement already satisfied: six>=1.5 in /usr/local/lib/python3.10/dist-packages (from python-dateutil>=2.8.1->pandas->shap) (1.16.0)
Installing collected packages: slicer, shap
Successfully installed shap-0.43.0 slicer-0.0.7
```

```
#IMPORTING NECESSARY LIBRARIES
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
%matplotlib inline
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LogisticRegression
from sklearn.tree import DecisionTreeClassifier
from sklearn.ensemble import RandomForestClassifier
from xgboost import XGBClassifier
from sklearn.model_selection import GridSearchCV
from sklearn.metrics import accuracy_score, roc_auc_score
import eli5
from eli5.sklearn import PermutationImportance
import lime
import lime.lime_tabular
import shap
import warnings
warnings.filterwarnings("ignore")
```

```
from google.colab import drive
drive.mount('/content/drive')
```

Mounted at /content/drive

```
#READING THE DATASET
df=pd.read_csv("/content/heart_failure_clinical_records_dataset.csv")
```

```
df.head()
```

	age	anaemia	creatinine_phosphokinase	diabetes	ejection_fraction	high_blood_pressure	platelets	serum_creatinine	serum_sodium
0	75.0	0	582	0	20	1	265000.00	1.9	130
1	55.0	0	7861	0	38	0	263358.03	1.1	136
2	65.0	0	146	0	20	0	162000.00	1.3	129
3	50.0	1	111	0	20	0	210000.00	1.9	137
4	65.0	1	160	1	20	0	327000.00	2.7	116

```
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
```

```
df.isnull().sum()

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
```

```
# CHECKING FOR DUPLICATES
dups = df.duplicated()
print('Number of duplicate rows = %d' % (dups.sum()))
df[dups]

Number of duplicate rows = 0
age anaemia creatinine phosphokinase diabetes ejection fraction high blood pressure platelets serum creatinine serum sodium se
```

```
df.describe(include="all")
```

	age	anaemia	creatinine_phosphokinase	diabetes	ejection_fraction	high_blood_pressure	platelets	serum_creatini
count	299.000000	299.000000	299.000000	299.000000	299.000000	299.000000	299.000000	299.0000
mean	60.833893	0.431438	424.214883	0.418060	38.033445	0.351171	259163.714883	1.2345
std	11.894809	0.496107	385.449328	0.494067	11.685643	0.478136	81478.304369	0.4400
min	40.000000	0.000000	23.000000	0.000000	14.000000	0.000000	76000.000000	0.5000
25%	51.000000	0.000000	116.500000	0.000000	30.000000	0.000000	212500.000000	0.9000
50%	60.000000	0.000000	250.000000	0.000000	38.000000	0.000000	262000.000000	1.1000
75%	70.000000	1.000000	582.000000	1.000000	45.000000	1.000000	303500.000000	1.4000
max	95.000000	1.000000	1280.250000	1.000000	67.500000	1.000000	440000.000000	2.1500

```
# MALE AND FEMALE COUNT
df.sex.value_counts()

1    194
0    105
Name: sex, dtype: int64
```

```
df['DEATH_EVENT'].value_counts()

0    203
1     96
Name: DEATH_EVENT, dtype: int64
```

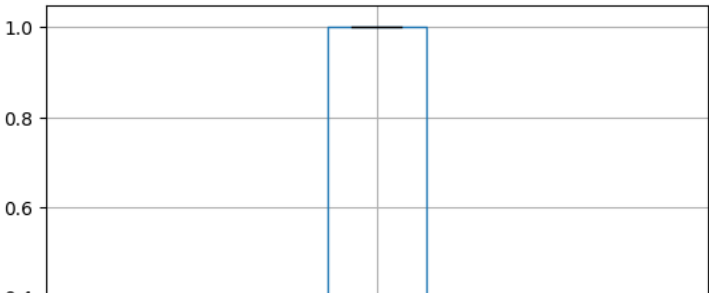
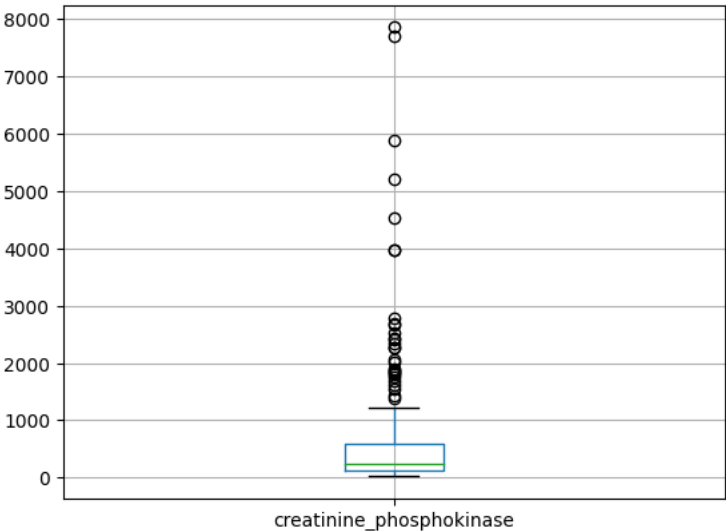
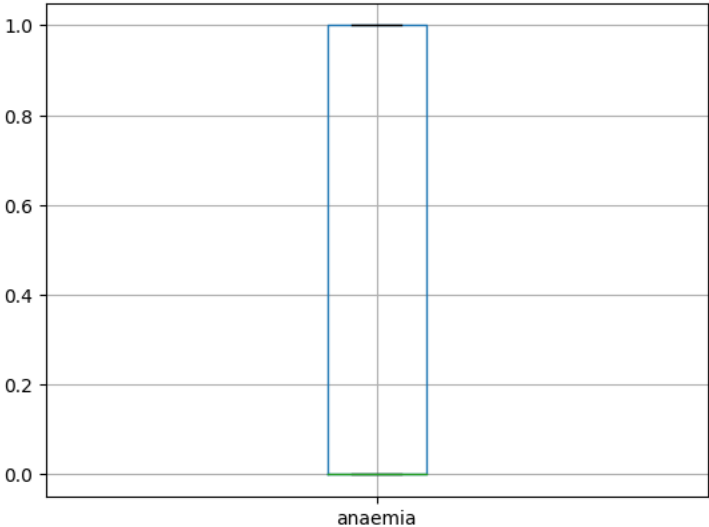
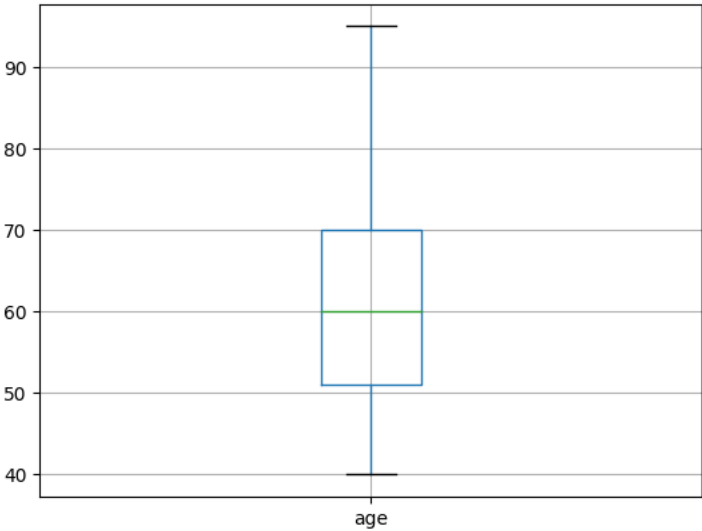
```
# EACH CLASS PROPORTION
df['DEATH_EVENT'].value_counts(normalize=True)

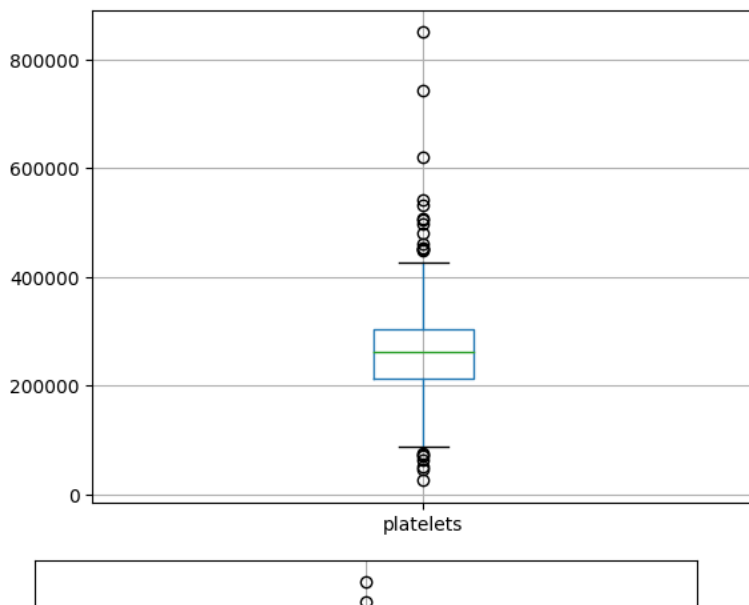
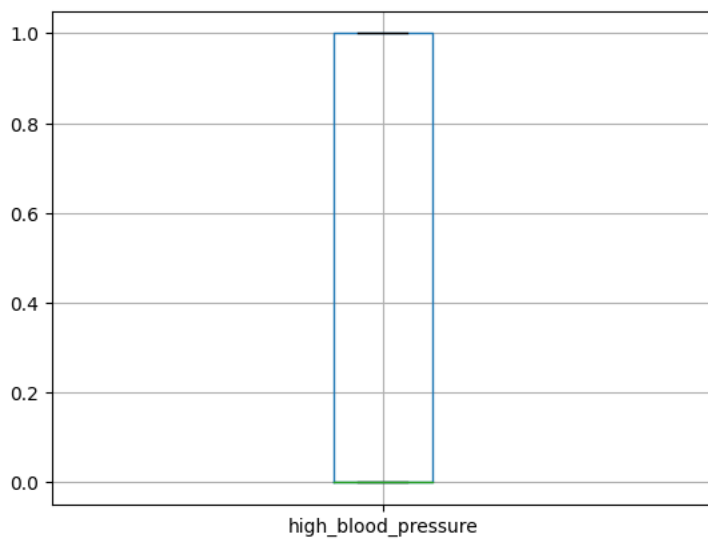
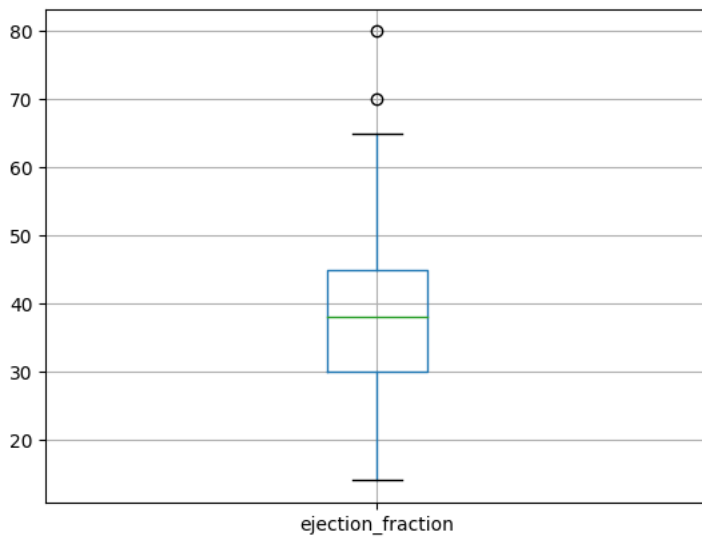
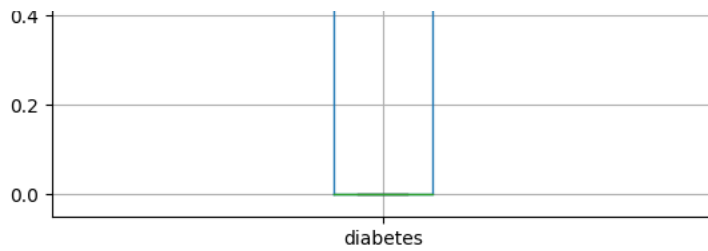
0    0.67893
1    0.32107
Name: DEATH_EVENT, dtype: float64
```

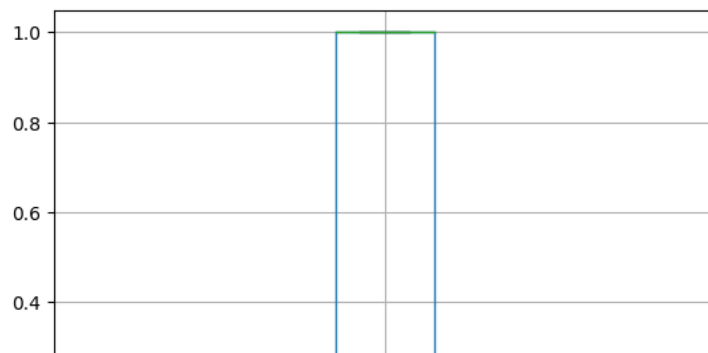
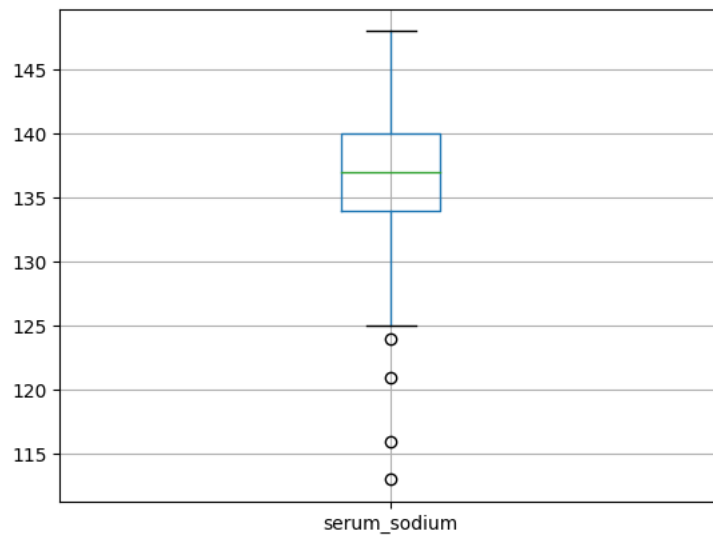
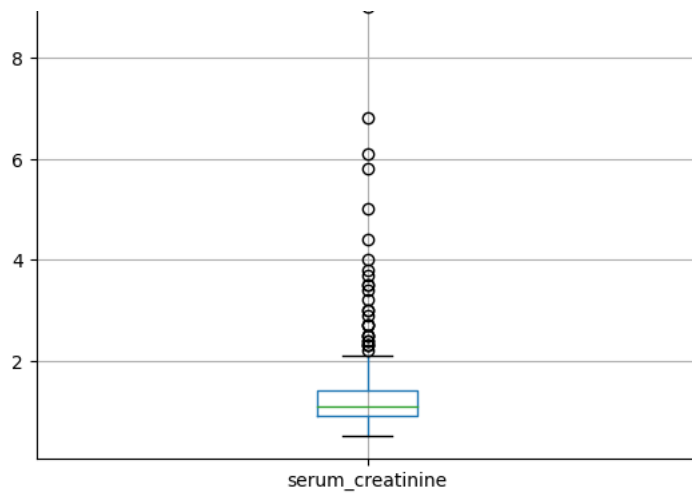
```
df.columns

Index(['age', 'anaemia', 'creatinine_phosphokinase', 'diabetes',
      'ejection_fraction', 'high_blood_pressure', 'platelets',
      'serum_creatinine', 'serum_sodium', 'sex', 'smoking', 'time',
      'DEATH_EVENT'],
      dtype='object')
```

```
for column in df:  
    plt.figure()  
    df.boxplot([column])
```

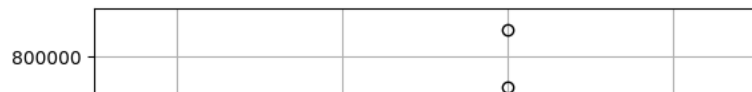






```
# BOXPLOT FOR CONTINUOUS VARIABLES
df.boxplot(column=['creatinine_phosphokinase', 'ejection_fraction', 'platelets', 'serum_creatinine'])
```

&lt;Axes: &gt;



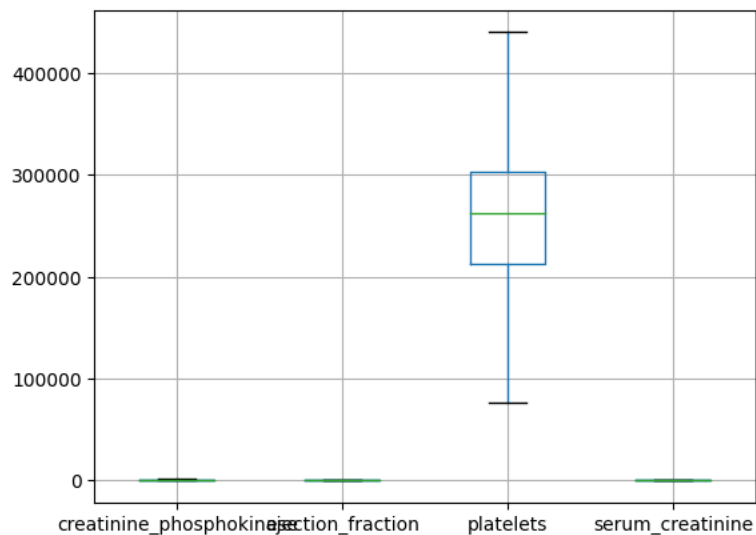
# CALCULATING LOWER RANGE AND UPPER RANGE

```
def treat_outlier(col):
    sorted(col)
    Q1,Q3=np.percentile(col,[25,75])
    IQR=Q3-Q1
    lower_range= Q1-(1.5 * IQR)
    upper_range= Q3+(1.5 * IQR)
    return lower_range, upper_range
```

```
for feature in df[['creatinine_phosphokinase', 'ejection_fraction', 'platelets', 'serum_creatinine']]:
    lr,ur=treat_outlier(df[feature])
    df[feature]=np.where(df[feature]>ur,ur,df[feature])
    df[feature]=np.where(df[feature]<lr,lr,df[feature])
```

```
df.boxplot(column=['creatinine_phosphokinase', 'ejection_fraction', 'platelets', 'serum_creatinine'])
```

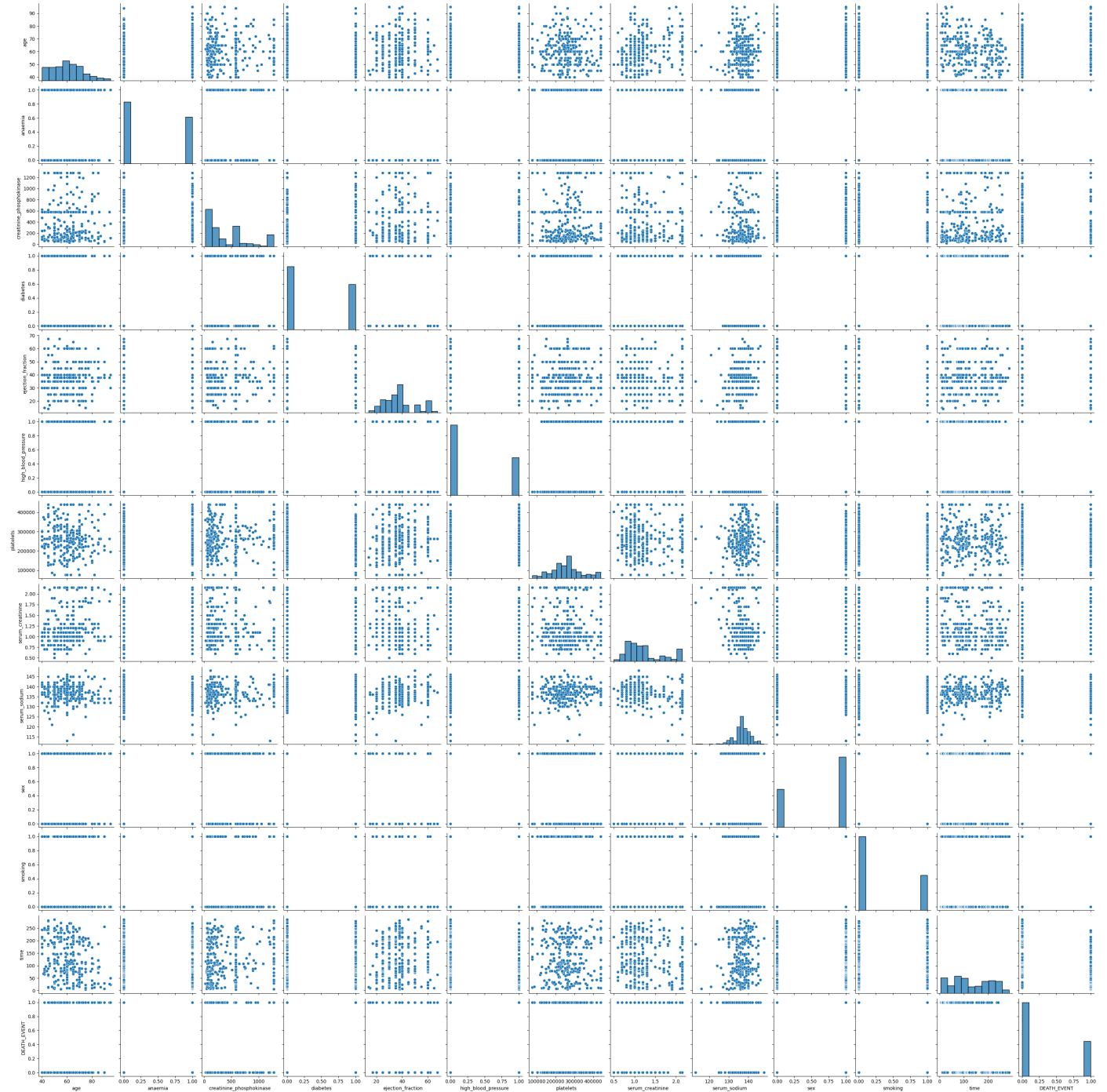
&lt;Axes: &gt;



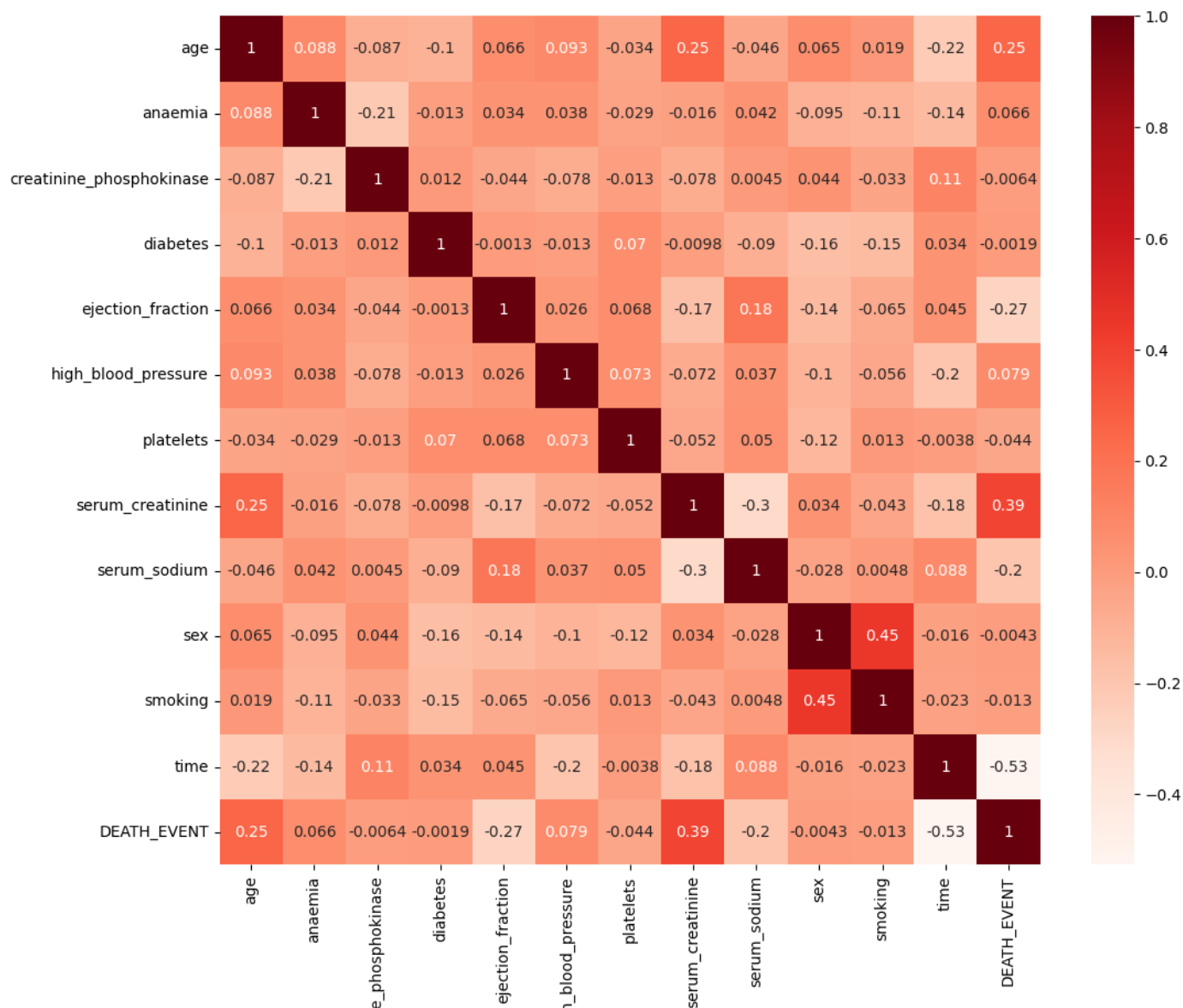
```
sns.pairplot(df)
```



```
<seaborn.axisgrid.PairGrid at 0x7912284d04c0>
```



```
#PEARSON CORRELATION
plt.figure(figsize=(12,10))
cor = df.corr()
sns.heatmap(cor, annot=True, cmap=plt.cm.Reds)
plt.show()
```



```
# SPLITTING DATA INTO TRAIN AND TEST SETS
X = df.drop('DEATH_EVENT', axis=1)
y = df['DEATH_EVENT']
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)
```

```
#LOGISTIC REGRESSION
lr = LogisticRegression()
lr_params = {'C': [0.001, 0.01, 0.1, 1, 10]}
lr_gscv = GridSearchCV(lr, lr_params, scoring='roc_auc', cv=5)
lr_gscv.fit(X_train, y_train)
print('Best LR params:', lr_gscv.best_params_)
lr_best = lr_gscv.best_estimator_
lr_best.fit(X_train, y_train)
lr_preds = lr_best.predict(X_test)
lr_probs = lr_best.predict_proba(X_test)
print('LR accuracy:', accuracy_score(y_test, lr_preds))
print('LR AUC:', roc_auc_score(y_test, lr_preds))
```

```
Best LR params: {'C': 0.01}
LR accuracy: 0.8166666666666667
LR AUC: 0.7914285714285714
```

```
# INTERPRETING LR MODEL
lr_perm = PermutationImportance(lr_best).fit(X_test, y_test)
eli5.show_weights(lr_perm, feature_names = X.columns.tolist())
```