Predicting Mortality in Heart Failure Patients Using Logistic Regression

1 PROBLEM FORMULATION

1.1 PROBLEM

How accurately can we predict the chances of a patient getting a heart disease?

1.2 DATASET

The dataset of different patient's information contains 13 features, along with 299 datapoints (without missing values). Each datapoint represents a patient's medical data.

The properties – or features of the dataset are the following:

- presence of anaemia, presence of diabetes, if the patient has high blood pressure, gender, if they smoke, and if the patient has passed away are all **binary** data.
- age, platelets amount in blood and serum sodium amount in blood are **continuous** data.
- CPK enzyme amount in blood, ejection fraction of heart (percentage), serum creatine amount in blood, serum sodium amount in blood, time follow-up period (of diagnosis, in days) are **numerical** data.

My project aims to prove and explore that heart failure can be predicted from a selected number of features alone, investigating risk factors of heart disease, while also attempting to forecast a patient's chances of dying from heart failure. The dataset can be found on Kaggle [1] and is additionally used in another related research paper [2].

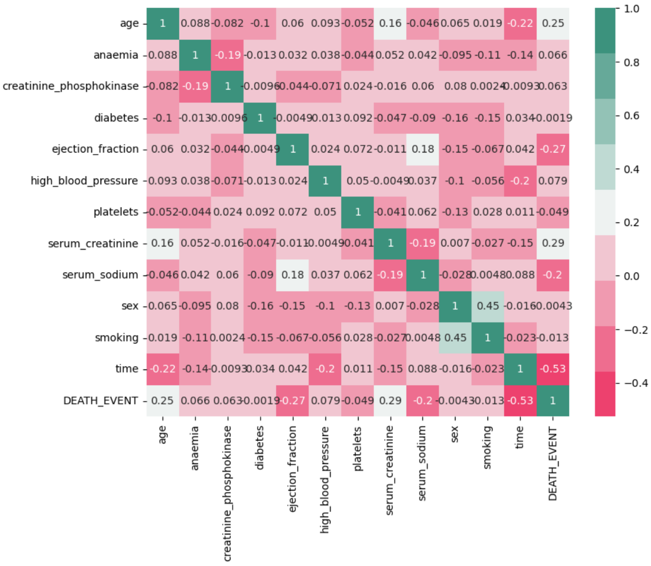
1.3 FEATURES AND LABELS

The target variable or label is the feature called 'DEATH_EVENT', which indicates if the patient has passed away or not. Therefore, I will be aiming to use a supervised model.

2 METHODS

I will be using Logistic Regression, a binary classification model, falling under the category of linear predictor maps, to solve the problem, as the data points can be classified to two categories, whether the patient has died (1) or is alive (0), which is why I will be picking the 'DEATH_EVENT' to be the deciding label of my model. The model itself is ideal due to its simplicity, and since it assumes that there is a linear relationship between features and labels, it is sufficient for my dataset and goal.

Having looked at the related research paper [2], I've selected features that would correlate the best with my model, having dropped redundant ones onward. The research indicated: "[...] **serum creatinine** and **ejection fraction** are sufficient to predict survival of heart failure patients from medical records [...]". To examine this claim, I have used a correlation matrix further and analysed feature importance to see which features should be retained. Notably, the research paper’s findings and the correlation matrix’ results are mostly equivalent, as shown in Figure 1 below.



1. Figure The correlation of features, shown in a matrix. In the ’DEATH_EVENT’ column, we can observe that time, serum_sodium and ejection_fraction play an important part in whether death incurs.

The dataset will be split according to the 70-30 ratio for training-testing, as I aspire the model to be more generalized and robust, unlike the case with the 80-20 ratio. Commonly, the 70-30 is also more used on smaller datasets [3], such as the current one, although the chosen ratio can vary a lot depending on context and model. Given the simplicity of the model, I will not be dividing the dataset according to cross validation. The data splitting can be done with the function ‘train_test_split()’ from the scikit-learn library.

For choosing the adequate loss function, I have chosen logistic loss, as that would fit with my model more, and is more commonly used in binary classification problems. Additionally, the loss function is chosen as it allowed the use of a ready-made library for logistic regression with scikit-learn.

3 REFERENCES, SOURCES

[1] Heart Failure Prediction Dataset (2020) sourced from UC Irvine Machine Learning Repository - https://www.kaggle.com/datasets/andrewmvd/heart-failure-clinical-data

[2] ‘Machine learning can predict survival of patients with heart failure from serum creatinine and ejection fraction alone’ (2020 February 3) by D. Chicco, Giuseppe Jurman - https://www.semanticscholar.org/paper/Machine-learning-can-predict-survival-of-patients-Chicco-Jurman/e64579d8593140396b518682bb3a47ba246684eb

[3] Article on ‘What is data splitting?’ by Alexander S. Gillis ’https://www.techtarget.com/searchenterpriseai/definition/data-splitting#:~:text=Data%20should%20be%20split%20so,optimal%20for%20small%20data%20sets.

Appendix

September 20, 2024

[84]:

```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns  #data visualization library
from sklearn.linear_model import LogisticRegression
from sklearn.metrics import accuracy_score, confusion_matrix  # evaluation␣
 ↪metrics
from sklearn.model_selection import train_test_split


df = pd.read_csv('heart_failure_clinical_records_dataset.csv')
df.head(5)
```

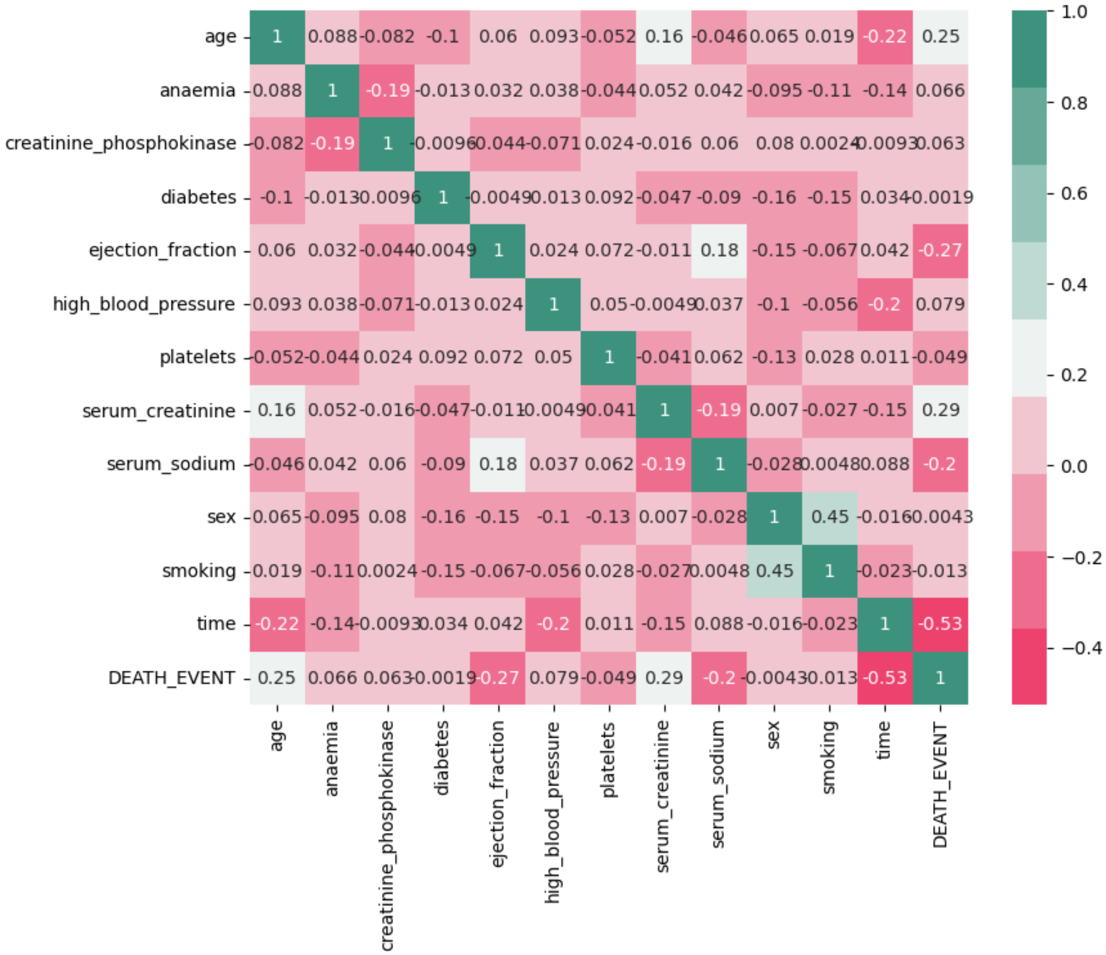
[84]:

```
    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

   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

   smoking  time  DEATH_EVENT
0        0     4            1
1        0     6            1
2        1     7            1
3        0     7            1
4        0     8            1
```

[83]:

```
# Correlation matrix
corr_matrix = df.corr()
```

```
plt.figure(figsize=(9,7))
sns.heatmap(corr_matrix, annot=True, cmap=cmap)
plt.show()
```



[78]:
```
#only taking into account the serum_creatine, ejection fraction and time factors
```

[79]:
```
features = ['time', 'ejection_fraction','serum_creatinine']
X = df[features]
y = df['DEATH_EVENT']
```

[80]:
```
X_train, X_test, y_train, y_test = train_test_split(X,y, test_size=0.3,␣
↪random_state=4)
```

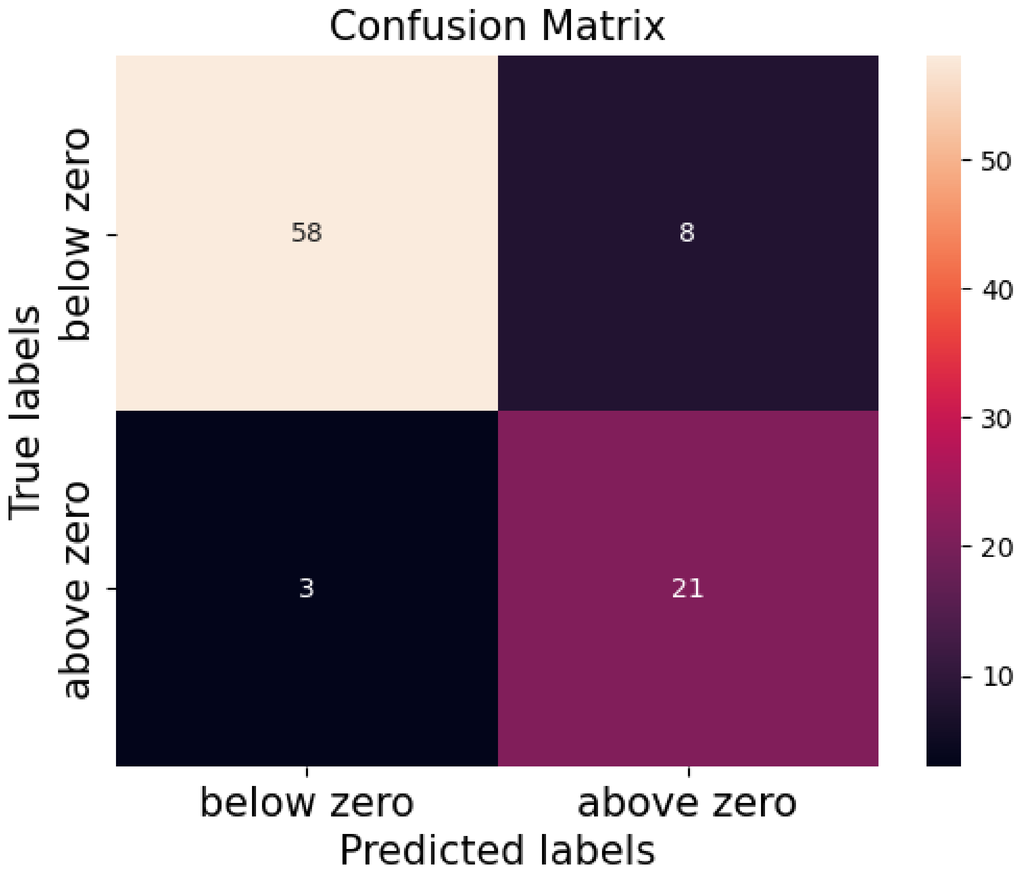
[81]:
```
logregmodel = LogisticRegression()
logregmodel.fit(X_train, y_train)
y_pred = logregmodel.predict(X_test)
acc = accuracy_score(y_test, y_pred)
```

```
confmat = confusion_matrix(y_test, y_pred)
print("Accuracy: ", acc)
print("Confusion Matrix:\n", confmat)
```

```
Accuracy:  0.8777777777777778
Confusion Matrix:
 [[58  8]
 [ 3 21]]
```

[82]:
```
ax= plt.subplot()

sns.heatmap(confmat, annot=True, fmt='g', ax=ax)

ax.set_xlabel('Predicted labels',fontsize=15)
ax.set_ylabel('True labels',fontsize=15)
ax.set_title('Confusion Matrix',fontsize=15)
ax.xaxis.set_ticklabels(['below zero', 'above zero'],fontsize=15)
ax.yaxis.set_ticklabels(['below zero', 'above zero'],fontsize=15)
```

[82]:
```
[Text(0, 0.5, 'below zero'), Text(0, 1.5, 'above zero')]
```


Confusion Matrix
below zero
above zero
True labels
below zero
above zero
Predicted labels