

Proiect Învățare Automată

Titlu tema: Heart Failure Prediction – Prezicerea bolilor de inima



Student: Draghici Andreea-Maria

Grupa: CR4.S1 A

Anul de studiu: IV

Specializarea: Calculatoare Romana

Link problema: <https://www.kaggle.com/datasets/andrewmvd/heart-failure-clinical-data?page=2>

Link github: <https://github.com/AndreeaDraghici/Machine-Learning-Project.git>

Stim cu totii ca bolile cardiovasculare sunt cauza numarul 1 de deces la nivel global. Mi-am dorit sa analizez acest topic, pentru a vedea ce rezultate pot obtine.

Initial am importat in Jupyter librariile si datasetul pentru a il incarca si citii datele din el.

```
#importare librarii
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
%matplotlib inline
import seaborn as sns
sns.set(style='darkgrid')
import warnings
warnings.filterwarnings('ignore')
import time

#importare functii
from sklearn.model_selection import train_test_split
from sklearn.metrics import accuracy_score
from sklearn.preprocessing import LabelEncoder
from sklearn.model_selection import cross_val_score
from sklearn.model_selection import StratifiedKFold

#importare modele
from sklearn.ensemble import RandomForestClassifier
from sklearn.linear_model import LogisticRegression
from sklearn.tree import DecisionTreeClassifier
from sklearn.neighbors import KNeighborsClassifier
from sklearn.discriminant_analysis import LinearDiscriminantAnalysis
from sklearn.naive_bayes import GaussianNB
from sklearn.svm import SVC

df=pd.read_csv("C:\\Users\\user\\Desktop\\Proiect_IA\\heart_failure_clinical_records_dataset.csv") #incarc datasetul si il citesc
```

Pentru acest lucru, pe baza unui dataset, am utilizat mai multe metode pentru a obtine statistici reprezentative ca mai jos. Pentru a vedea procentul deceselor si al bolii decesului atat la femei, cat si la barbati.

```
human_df = df.copy()

human_df["diabetes"] = human_df["diabetes"].replace(0, "non-diabetes")
human_df["diabetes"] = human_df["diabetes"].replace(1, "diabetes")

human_df["sex"] = human_df["sex"].replace(0, "male")
human_df["sex"] = human_df["sex"].replace(1, "female")

human_df["smoking"] = human_df["smoking"].replace(0, "non-smoking")
human_df["smoking"] = human_df["smoking"].replace(1, "smoking")

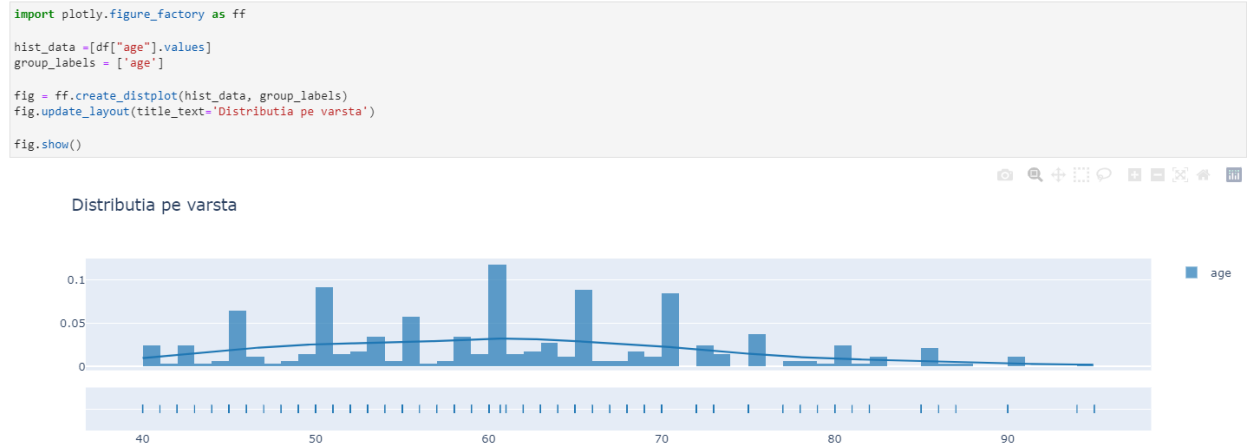
human_df["DEATH_EVENT"] = human_df["DEATH_EVENT"].replace(0, "dead")
human_df["DEATH_EVENT"] = human_df["DEATH_EVENT"].replace(1, "alive")

fig = px.sunburst(
    human_df,
    path=['sex', 'smoking', 'diabetes', 'DEATH_EVENT'],
)

fig.show()
```



Pe baza atributului age, am dorit sa vad o distributie pe varsta a deceselor.



Am impartit datele pentru antrenament si test, apoi am evaluat modelele pentru a obtine acuratetea.

```
: #separam variabila de raspuns din setul de date
X=df.drop('DEATH_EVENT',axis=1)
y=df['DEATH_EVENT']
#datele de antrenament
X_train, X_val, y_train, y_val = train_test_split(X, y, test_size=0.25, random_state=2)

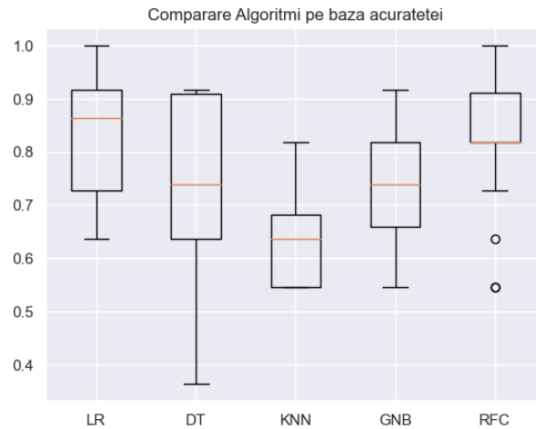
#modelele
models=[]
models.append(('LR',LogisticRegression(solver='liblinear',multi_class='ovr')))
models.append(('DT',DecisionTreeClassifier()))
models.append(('KNN',KNeighborsClassifier()))
models.append(('GNB',GaussianNB()))
models.append(('RFC',RandomForestClassifier()))

#evaluarea modelului
results=[]
names=[]
for name,model in models:
    kfold=StratifiedKFold(n_splits=20)#random_state=1
    cv_results=cross_val_score(model, X_train, y_train, cv=kfold, scoring='accuracy')
    results.append(cv_results)
    names.append(name)
    print(' %s: Acuratete: %f' %(name,cv_results.mean()))

LR: Acuratete: 0.838258
DT: Acuratete: 0.743561
KNN: Acuratete: 0.638258
GNB: Acuratete: 0.753409
RFC: Acuratete: 0.824621
```

Apoi am comparat algoritmi folositi in functie de acuratetea obtinuta anterior si am observat ca cei mai performanti algoritmi sunt Logistic Regression si Random Forest Classifier.

```
#comparatie modele folosite
plt.boxplot(results,labels=names)
plt.title("Comparare Algoritmi pe baza acuratetei") #observam ca Logistic Regression si Random Forest sunt algoritmi performanti, obtinand o acuratete cat mai buna cu ei
plt.show()
```



Apoi pentru cei doi algoritmi vom antrena modelele, datele de test si antrenament, pentru a obtine acuratetea intre valorile de test si cele prezise.

```
#antrenam modelele
lrc = LogisticRegression(solver='liblinear',multi_class='ovr')
start_time = time.time()
lrc.fit(X_train,y_train) #antrenare date
pred_y=lrc.predict(X_val) #prezicerea

#Accuracy
print('Acuratete Logistic Regression: ',accuracy_score(y_val.values,pred_y)) #date test , antrenament

rf = RandomForestClassifier()
start_time = time.time()
rf.fit(X_train,y_train) #antrenare date
pred_y=rf.predict(X_val) #prezicerea

#Accuracy
print('Acuratete Random Forest Classifier: ',accuracy_score(y_val.values,pred_y))
```

```
Acuratete Logistic Regression:  0.8933333333333333
Acuratete Random Forest Classifier:  0.8933333333333333
```

In final am vrut sa am un raport general asupra datasetului si astfel pe baza unei librarii importate, am putut genera un raport sub format html.

```
from pandas_profiling import ProfileReport

profile = ProfileReport(df, title="General Report")
profile.to_file("general_report.html")

profile
```

Rezultate:

Overview

Overview

Alerts 2

Reproduction

Dataset statistics

Number of variables	13
Number of observations	299
Missing cells	0
Missing cells (%)	0.0%
Duplicate rows	0
Duplicate rows (%)	0.0%
Total size in memory	30.5 KiB
Average record size in memory	104.4 B

Variable types

Numeric	7
Categorical	6

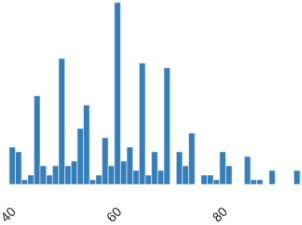
Variables

Select Columns

age

Real number (R)

Distinct	47	Minimum	40
Distinct (%)	15.7%	Maximum	95
Missing	0	Zeros	0
Missing (%)	0.0%	Zeros (%)	0.0%
Infinite	0	Negative	0
Infinite (%)	0.0%	Negative (%)	0.0%
Mean	60.833893	Memory size	2.5 KiB



More details

Referinte

1. <https://www.kaggle.com/datasets/andrewmvd/heart-failure-clinical-data>
2. <https://bmcmedinformdecismak.biomedcentral.com/articles/10.1186/s12911-020-1023-5>
3. <https://www.ncbi.nlm.nih.gov/pmc/articles/PMC5863635/>
4. <https://www.kaggle.com/datasets/andrewmvd/heart-failure-clinical-data/discussion>
5. <https://www.kaggle.com/datasets/andrewmvd/heart-failure-clinical-data/discussion/181241?page=2>

Concluzii

Facand research pe internet pentru acest proiect, am observat ca exista foarte multe metode de prezicere a datasetului.

Consider ca am invatat multe lucruri interesante si cunostinte ce m-au ajutat, pentru ca tema aleasa m-a facut sa fiu curioasa si sa aflu mai multe moduri de prezicere, rezolvare a problemei.

Desi complexitatea rezolvarii este una foarte mica, slaba, consider totusi ca m-a ajutat sa acumulez mai multe informatii despre partea de prezicere, obtinere a unor statistici si a unei acurateti cat mai optime.