ENGR 1330

Final Project: Predicting the death due to heart failure

Timur Abdilov
Mohammed Uddin

TASK

list of tools:

- Numpy
- Pandas
- Matplotlib
- Sklearn
- heart_failure_clinical_records_dataset.csv
- jupyter notebook

Cardiovascular diseases (CVDs) are the number 1 cause of death globally, taking an estimated 17.9 million lives each year, which accounts for 31% of all deaths worlwide.

Heart failure is a common event caused by CVDs and this dataset contains 12 features that can be used to predict mortality by heart failure.

There are some factors that affects Death Event. This dataset contains person's information like age ,sex , blood pressure, smoke, diabetes,ejection fraction, creatinine phosphokinase, serum_creatinine, serum_sodium, time and we have to predict their DEATH EVENT.

Predictions are made using classification algorithm

About Dataset

```
 ---  ------                    --------------  -----
  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
```

	age	anaemia	creatinine_phosphokinase	diabetes
0	75.0	0	582	0
1	55.0	0	7861	0
2	65.0	0	146	0
3	50.0	1	111	0
4	65.0	1	160	1
5	90.0	1	47	0
6	75.0	1	246	0

ejection_fraction	high_blood_pressure	platelets	serum_creatinine
20	1	265000.00	1.9
38	0	263358.03	1.1
20	0	162000.00	1.3
20	0	210000.00	1.9
20	0	327000.00	2.7
40	1	204000.00	2.1
15	0	127000.00	1.2
60	0	454000.00	1.1

Extension: .csv (comma separated values)

299 entries (from 0 to 298)

As mentioned in a previous slide, there 12 values (predictors)

The algorithm will determine whether or not the death will occur based on these predictors.

Dataset contains binary values !!!

- **anaemia (0-not diagnosed, 1-diagnosed)**
- **diabetes (0-not diagnosed, 1-diagnosed)**
- **high blood pressure (0-not diagnosed, 1-diagnosed)**
- **sex (0-Female, 1-Male)**
- **smoking (0-No, 1-Yes)**
- **DEATH EVENT (1-patient died, 0-patient survived)**

More info about dataset

serum_sodium	sex	smoking	time	DEATH_EVENT
130	1	0	4	1
136	1	0	6	1
129	1	1	7	1
137	1	0	7	1
116	0	0	8	1
132	1	1	8	1

Feature	Explanation
Age	Age of the patient
Anaemia	Decrease of red blood cells or hemoglobin
High blood pressure	If a patient has hypertension
Creatinine phosphokinase (CPK)	Level of the CPK enzyme in the blood
Diabetes	If the patient has diabetes
Ejection fraction	Percentage of blood leaving

Sex	Woman or man
Platelets	Platelets in the blood
Serum creatinine	Level of creatinine in the blood
Serum sodium	Level of sodium in the blood
Smoking	If the patient smokes
Time	Follow-up period
(target) death event	If the patient died during the follow-up perio

Quantitative values in dataset

- Age (Years)
- Creatinine phosphokinase (CPK) (mcg/L)
- Ejection fraction (% of blood leaving)
- Platelets (kiloplatelets/mL)
- Serum creatinine (mg/dL)
- Serum sodium (mEq/L)
- Time (follow-up period in days)

*mcg/L: micrograms per liter
*mEq/L: milliequivalents per litre
*mL: microliter

BMC Medical Informatics and Decision Making

	age	anaemia	creatinine_phosphokinase	diabetes	ejection_fraction	high_blood_pressure	platelets	serum_creatinine	serum_sodium	sex	smoking	time	DEATH_EVENT
count	299.000000	299.000000	299.000000	299.000000	299.000000	299.000000	299.000000	299.00000	299.000000	299.000000	299.00000	299.000000	299.00000
mean	60.833893	0.431438	581.839465	0.418060	38.083612	0.351171	263358.029264	1.39388	136.625418	0.648829	0.32107	130.260870	0.32107
std	11.894809	0.496107	970.287881	0.494067	11.834841	0.478136	97804.236869	1.03451	4.412477	0.478136	0.46767	77.614208	0.46767
min	40.000000	0.000000	23.000000	0.000000	14.000000	0.000000	25100.000000	0.50000	113.000000	0.000000	0.00000	4.000000	0.00000
25%	51.000000	0.000000	116.500000	0.000000	30.000000	0.000000	212500.000000	0.90000	134.000000	0.000000	0.00000	73.000000	0.00000
50%	60.000000	0.000000	250.000000	0.000000	38.000000	0.000000	262000.000000	1.10000	137.000000	1.000000	0.00000	115.000000	0.00000
75%	70.000000	1.000000	582.000000	1.000000	45.000000	1.000000	303500.000000	1.40000	140.000000	1.000000	1.00000	203.000000	1.00000
max	95.000000	1.000000	7861.000000	1.000000	80.000000	1.000000	850000.000000	9.40000	148.000000	1.000000	1.00000	285.000000	1.00000

we also performed extra model testing with normalized values, however it didn;t affect the accuracy, so normalization in this case is not necessary

for this purpose we utilize sklearn.preprocessing.StandardScaler()
z = (x-u)/s

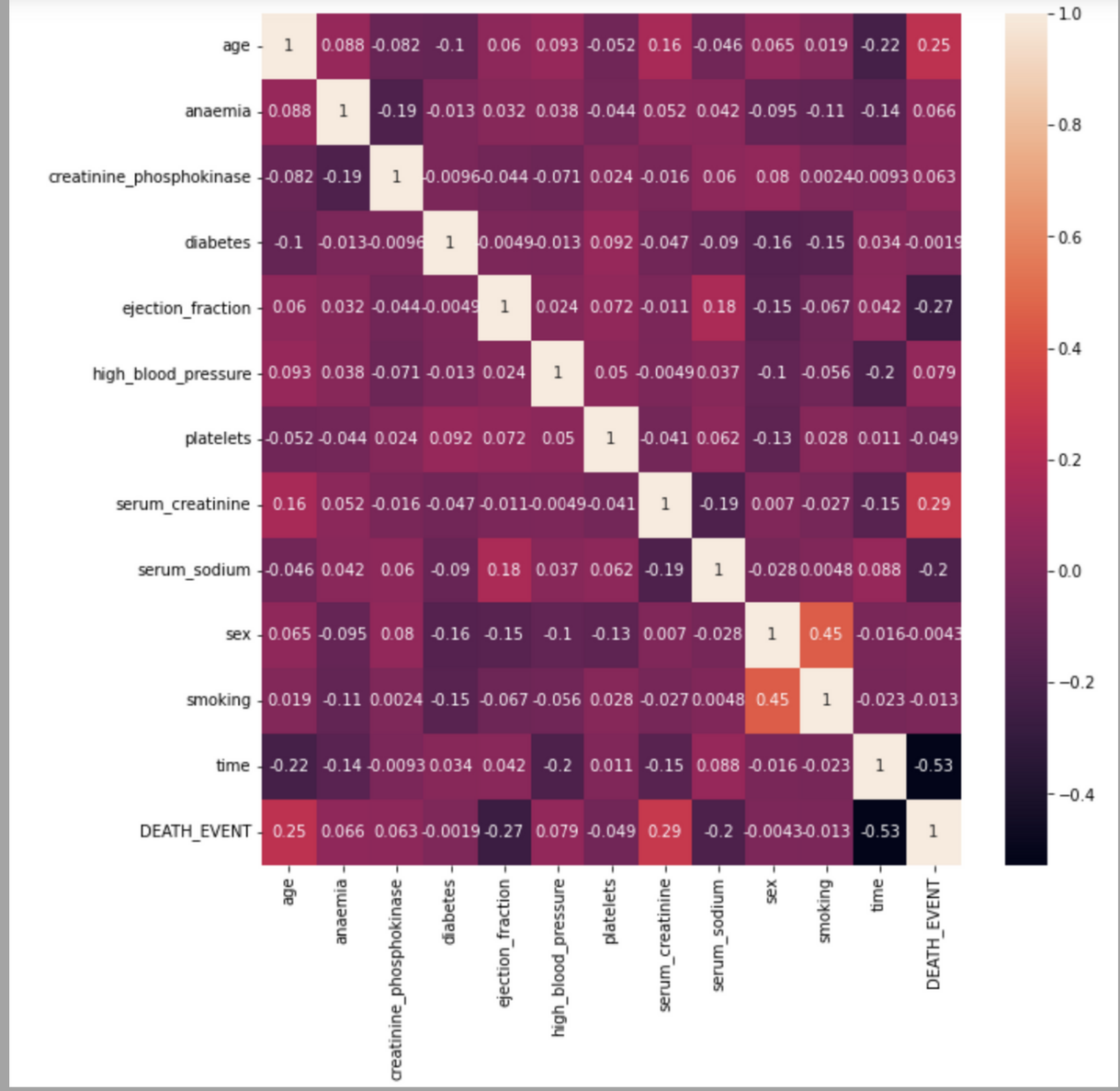
Some Descriptive Statistics

Seaborn Heatmap

Heatmap creates a correlation matrix based on our .corr() values of our dataframe

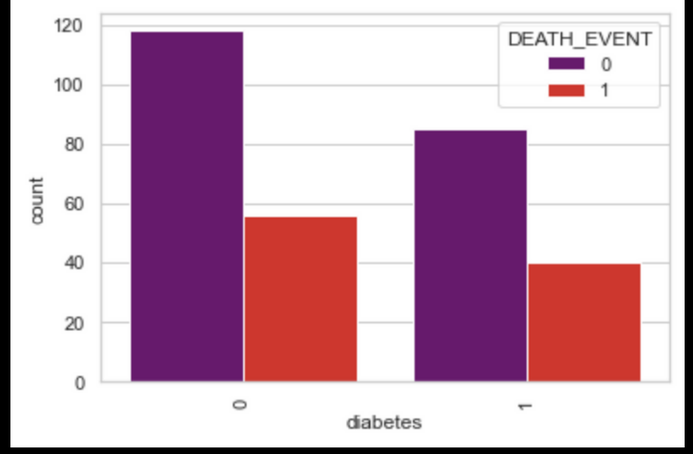
---

it is convenient to express correlation using color grading



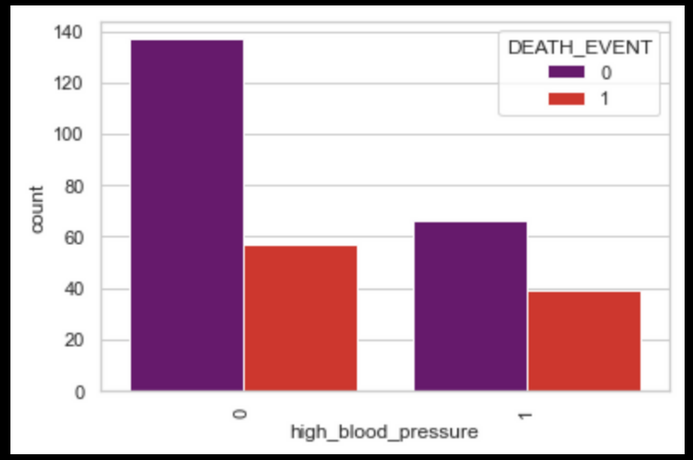

```
plt.figure(figsize=(10,10))
sns.heatmap(df_heart.corr(),annot=True)
plt.show()
```

Visualisation relationships (binary)



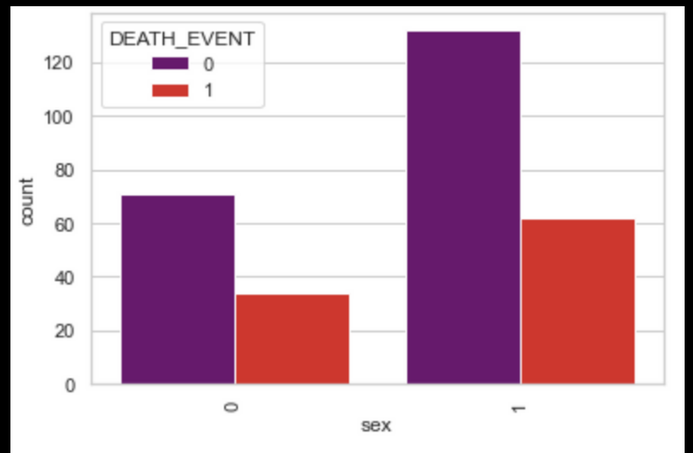
A bar graph displays the number of deaths and survivorships associated with and without diabetes diagnosis.

people without diabetes tend to survive in more cases



A bar graph displays the number of deaths and survivorships associated with and without high blood pressure (hypertension).

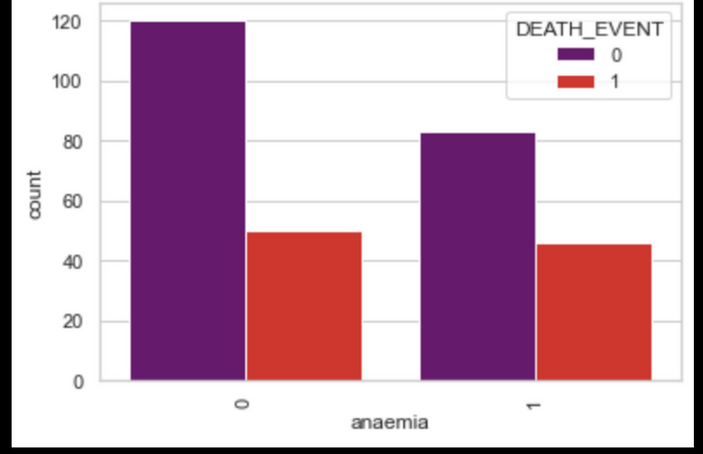
people without hypertension survive more often



A bar graph displays the number of deaths and survivorships associated with sex

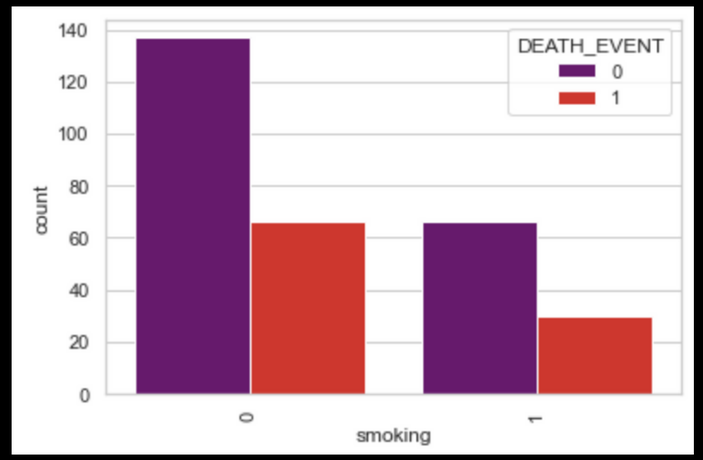
survival rate is higher for men, the number of death for men is also higher

A bit more visualisation (binary)



A bar graph displays the number of deaths and survivorships associated with and without anaemia diagnosis.

death cases here arer almost equal, however people without anaemia tend to survive in more cases

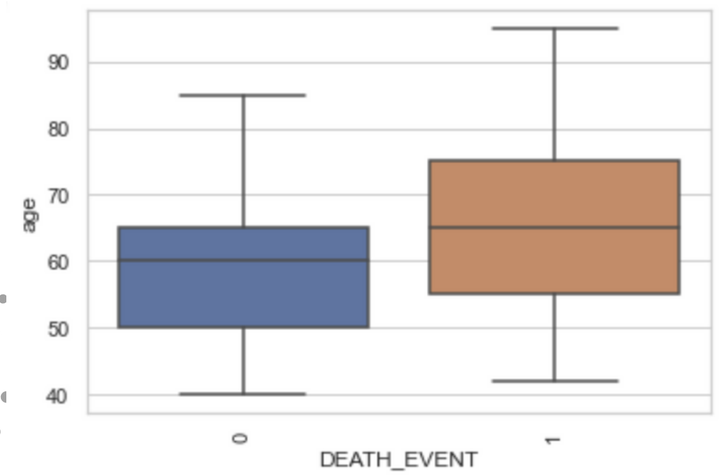


A bar graph displays the number of deaths and survivorships associated with smoking habit

of course, people who don't smoke survive more often

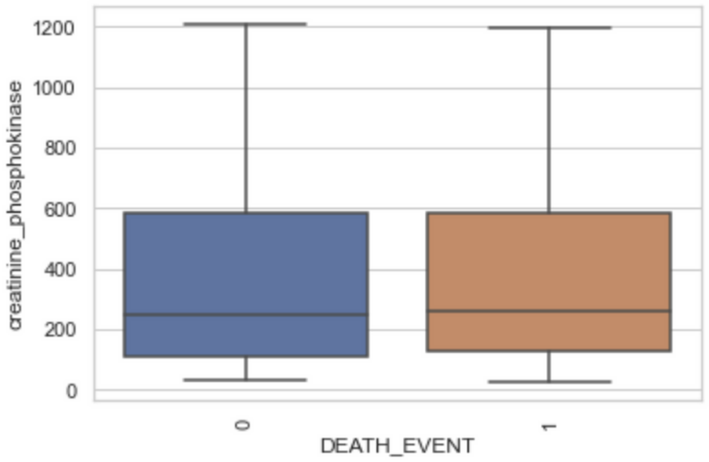
age vs Death

50% of people who died were over 65 years old
people who are yonger than 60 have more chance to stay alive



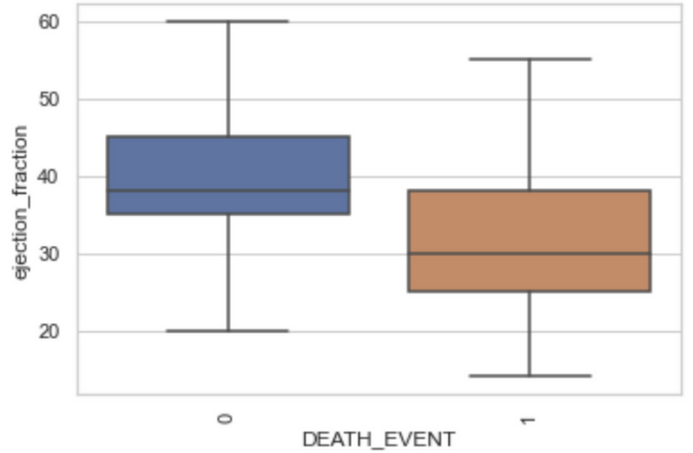
creatinine phosphokinase vs Death

this feature shows us that there is not significant difference in creatinine phosphokinase indicators between patients who died or stayed alive. In both cases 50% of population had CPK over +-240 mcg/L



ejection fraction vs Death

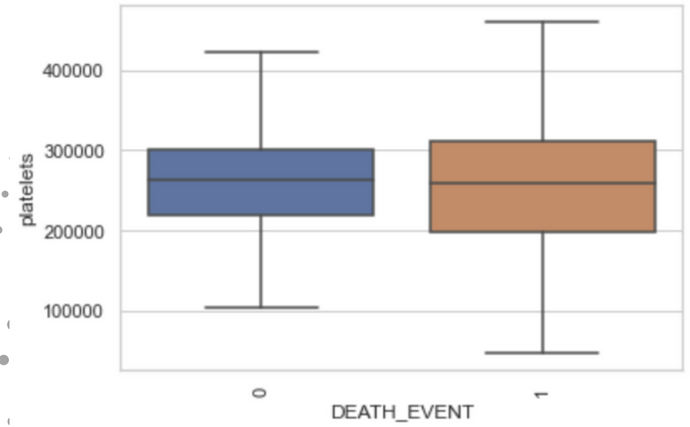
50% of population who didn't die due to heart disease have ejection fraction over 39%, while 50% patients who died had only 30% (very low)



Relationship between numerical values and death

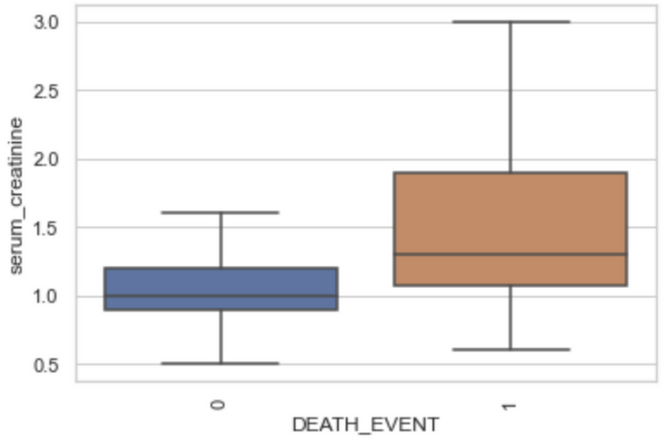
platelets vs Death

Though both, died and survived population have same medain number of platelets +-260k kiloplatelets/mL. But died population has higher IQR, and 25% had of died fraction had lower concentration level, means bad blood clotting and higher risk of internal bleeding



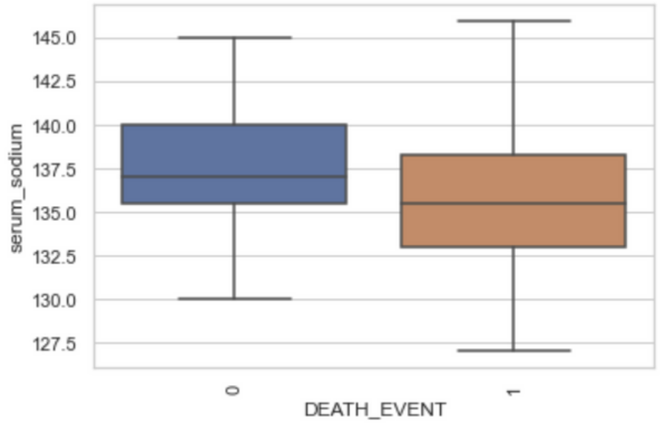
serum creatinine vs Death

overall, patients who died have higher level of serum creatinine. 50% had over 1.27 mcg/dL (kidney failure)



serum sodium vs Death

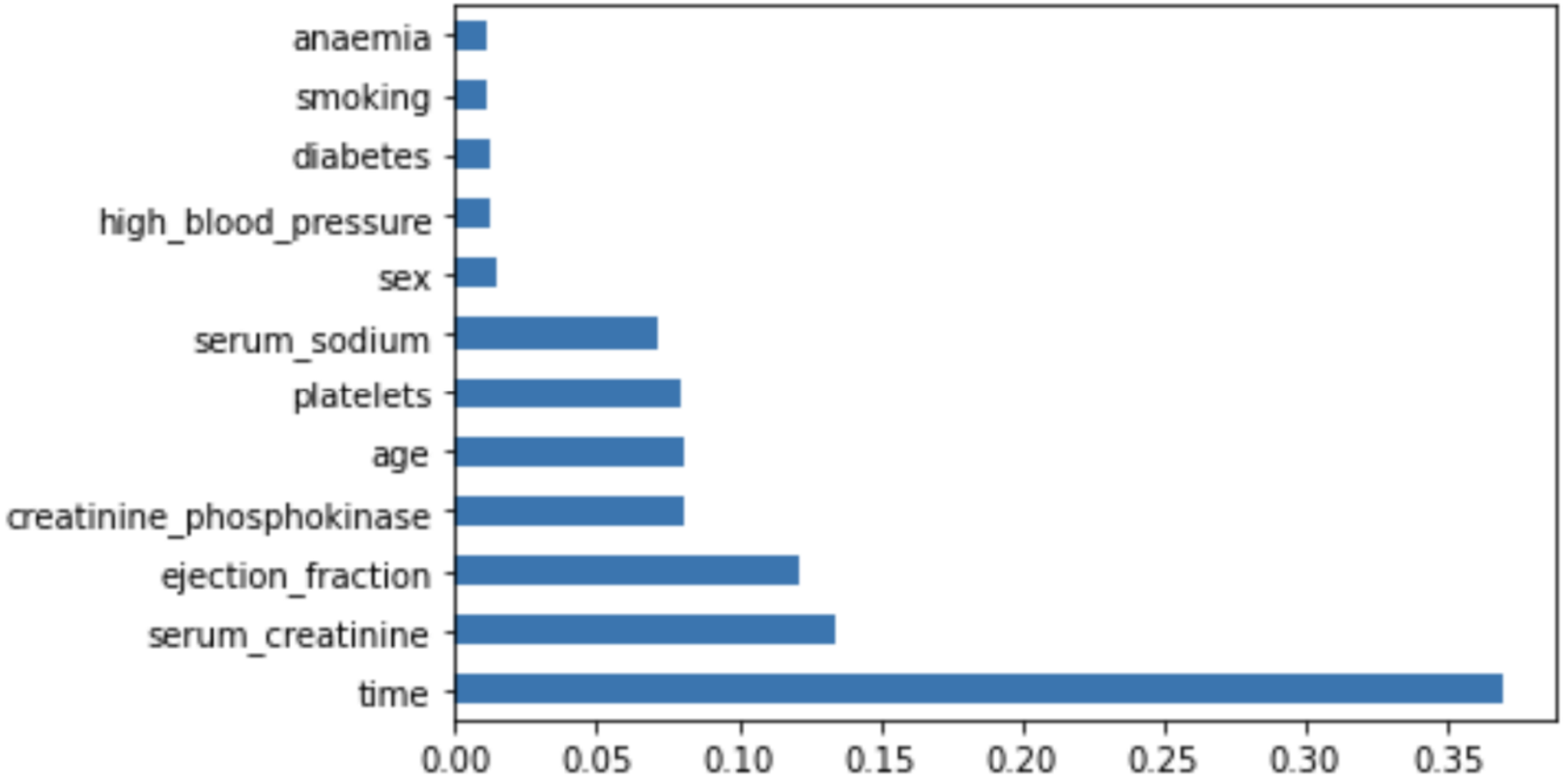
Here opposite, most of died patients have lower serum sodium level. It affects cardiac output by either decreasing heart rate



Penn Medicine

a bit more

feature importance



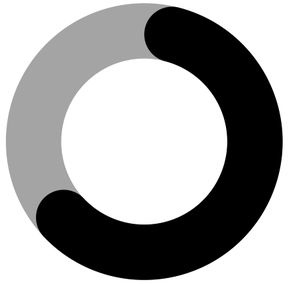

```
print(model.feature_importances_)
```

Numeriacal values have more importance rather than binaries

RandomForestClassifier() function

Back to Agenda Page

```
x_train, x_test, y_train, y_test = train_test_split(x, y, test_size=0.2, random_state=0)
```



80/20 ratio is used to divide the original dataset.

80% will go for train set, 20% for validation

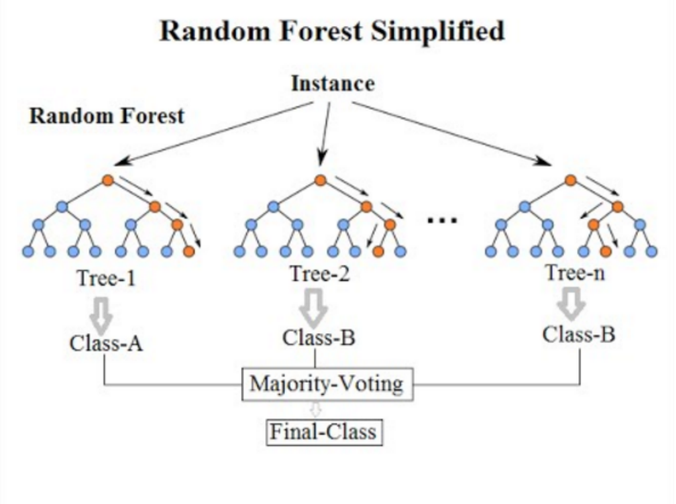
sklearn train_test_split() function is used

x_train - predictors train set
y_train - train set of value we need to predict
x_test - test set that algorithm will take
y_test - will be added to x_test dataframe to compare predicted values

Some possible ways to solve problem of classification

- Naive Bayes
- SVM (support vector machines)
- RandomForestClassifier (most efficient)
- KNeighborsClassifier
- DecisionTreeClassifier
- LogisticRegression

dividing the dataset into the train and validation sets



RandomForestClassifier

ensemble learning method for classification

Random Forest grows multiple decision trees

trees are merged together for a more accurate prediction

Prediction is based on Vote

algorithm chooses the classification with the majority of the “votes

mostly used in Healthcare

usually maintains its accuracy

Missing data is not a problem

also can be used for regression

Prediction of illnesses

.fit() just takes training data as arguments for algorithm

.predict() indicates what varieble will be an output for prediction

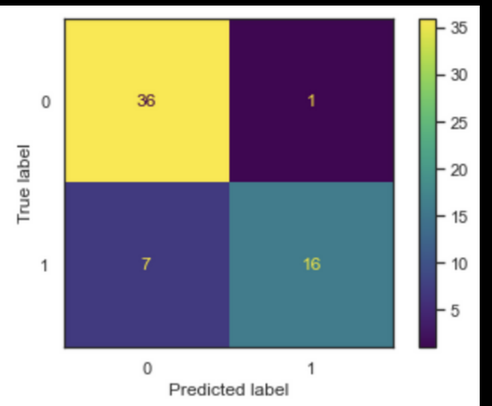
```
from sklearn.ensemble import RandomForestClassifier
y_pred = RandomForestClassifier().fit(x_train, y_train).predict(x_val)
```

Results

You can observe first rows of test sample with predicted values

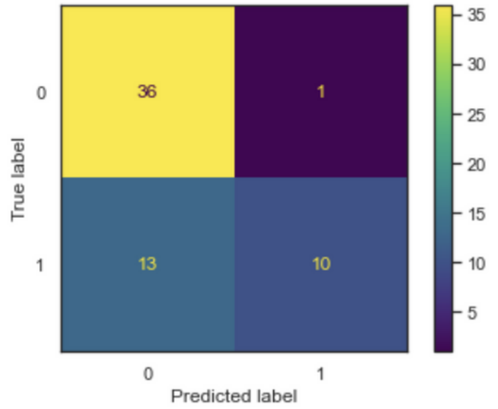
```
from sklearn.metrics import confusion_matrix, ConfusionMatrixDisplay
cm = confusion_matrix(y_val, y_pred_2)
disp = ConfusionMatrixDisplay(confusion_matrix=cm)
disp.plot()
plt.show()
```

87% accuracy is a good result.

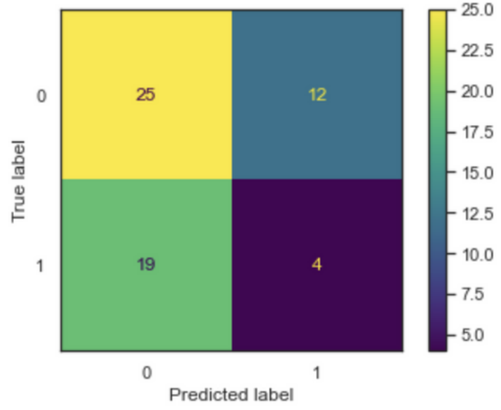


	age	anaemia	creatinine_phosphokinase	diabetes	ejection_fraction	high_blood_pressure	platelets	serum_creatinine	serum_sodium	sex	smoking	time	real	pred
206	40.000	1	101	0	40	0	226000.00	0.80	141	0	0	187	0	0
188	60.667	1	151	1	40	1	201000.00	1.00	136	0	0	172	0	0
12	45.000	1	981	0	30	0	136000.00	1.10	137	1	0	11	1	1
219	55.000	0	582	1	35	1	371000.00	0.70	140	0	0	197	0	0
237	70.000	0	232	0	30	0	173000.00	1.20	132	1	0	210	0	0
136	65.000	1	59	1	60	0	172000.00	0.90	137	0	0	107	0	0
228	65.000	0	56	0	25	0	237000.00	5.00	130	0	0	207	0	0
205	50.000	1	167	1	45	0	362000.00	1.00	136	0	0	187	0	0
52	60.000	0	3964	1	62	0	263358.03	6.80	146	0	0	43	1	1
108	63.000	0	936	0	38	0	304000.00	1.10	133	1	1	88	0	0
240	70.000	0	81	1	35	1	533000.00	1.30	139	0	0	212	0	0
15	82.000	1	379	0	50	0	47000.00	1.30	136	1	0	13	1	1

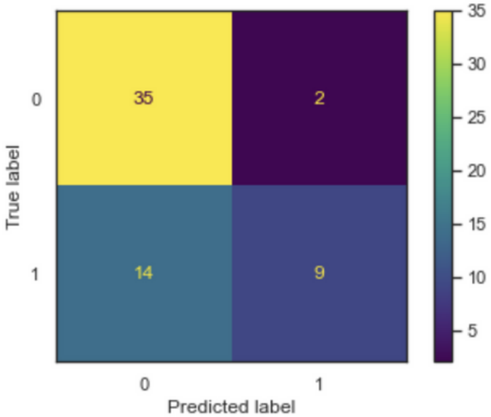
Comparison in accuracy



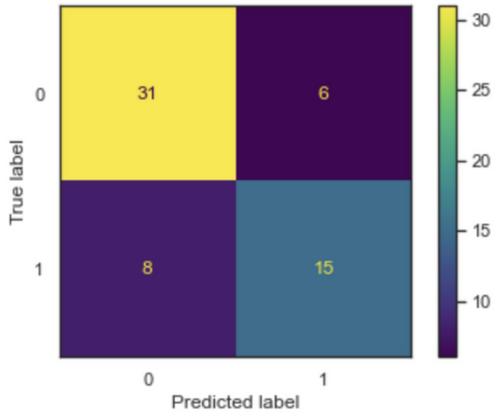
Naive Bayes



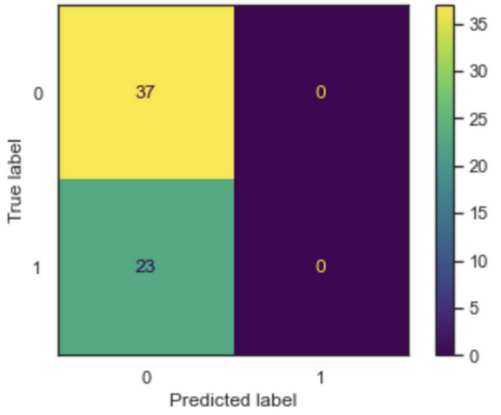
KNeighboursClassifier



LogisticRegression



DecisionTreeClassifier



SVC

Team 25

Thank you for listening

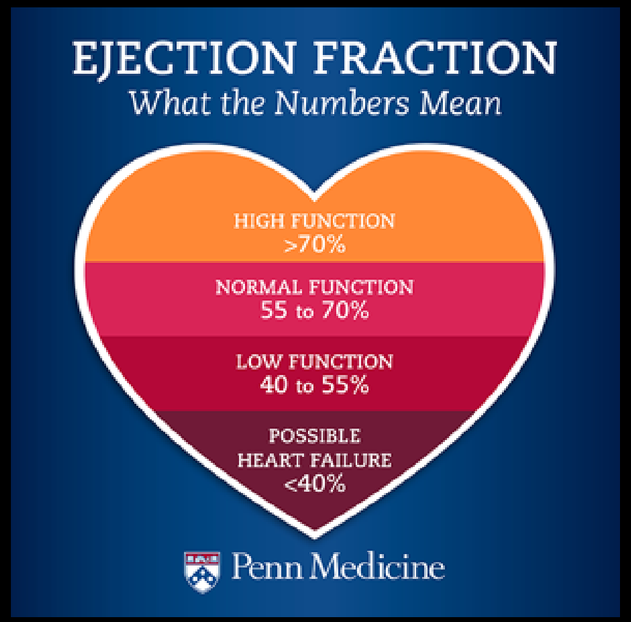
You are more than welcome to ask questions?

Email

tabdilov@ttu.edu

Strive for Honor
Evermore
Long Live the Matadors

Materials used



Penn Medicine

Duke Medicine

BMC research

UC Irvine statistics