

Cirrhosis: Liver transplant

NARVAEZ CAO ROMERO ALDO

The goal of this project is taking as target the variable Status, and use it to predict if a person is candidate to receive a transplant or not

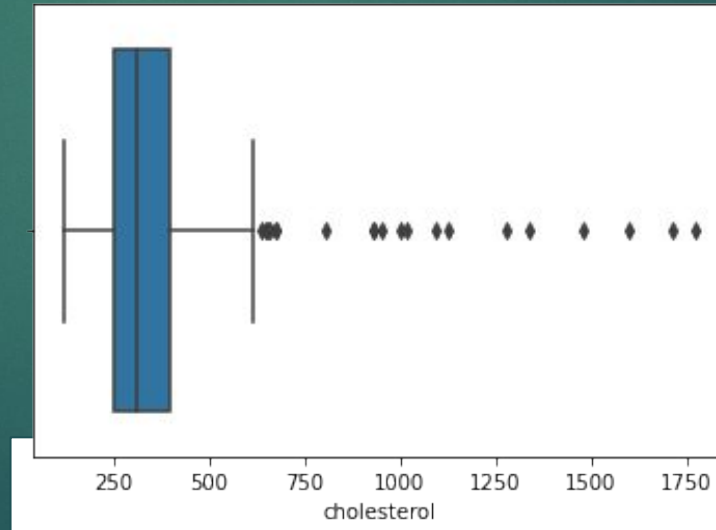
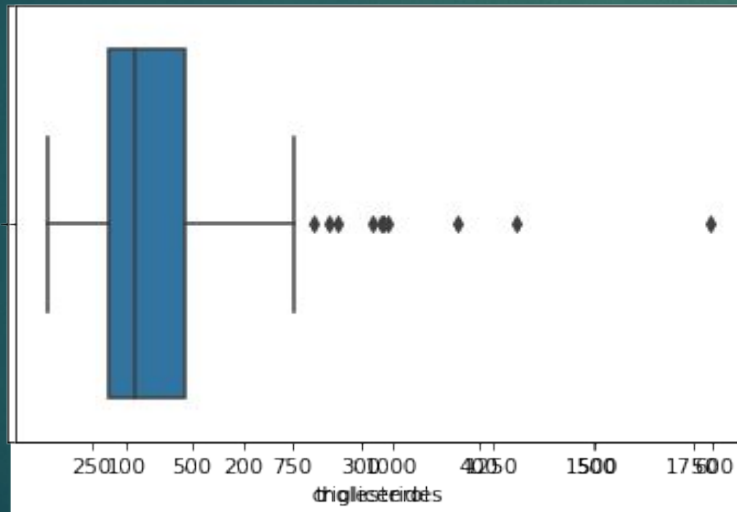
duration	status	drug	age	sex	ascites	hepatomology	spiders	edema	bilirubin	cholesterol	albumin	copper	phosphatase	SGOT	triglicerides	platelets	prothrombin	stage
400	2	1	21464	1	1	1	1	1.0	14.5	261.0	2.60	156.0	1718.0	137.95	172.0	190.0	12.2	4
4500	0	1	20617	1	0	1	1	0.0	1.1	302.0	4.14	54.0	7394.8	113.52	88.0	221.0	10.6	3
1012	2	1	25594	0	0	0	0	0.5	1.4	176.0	3.48	210.0	516.0	96.10	55.0	151.0	12.0	4
1925	2	1	19994	1	0	1	1	0.5	1.8	244.0	2.54	64.0	6121.8	60.63	92.0	183.0	10.3	4
1504	1	2	13918	1	0	1	1	0.0	3.4	279.0	3.53	143.0	671.0	113.15	72.0	136.0	10.9	3

There was a total of 312 rows and 20 columns

Cleaning and EDA

3

- Get rid off null values
- Replace the cholesterol and the triglycerides for their median and mean values
- Eliminate the rest of the Null



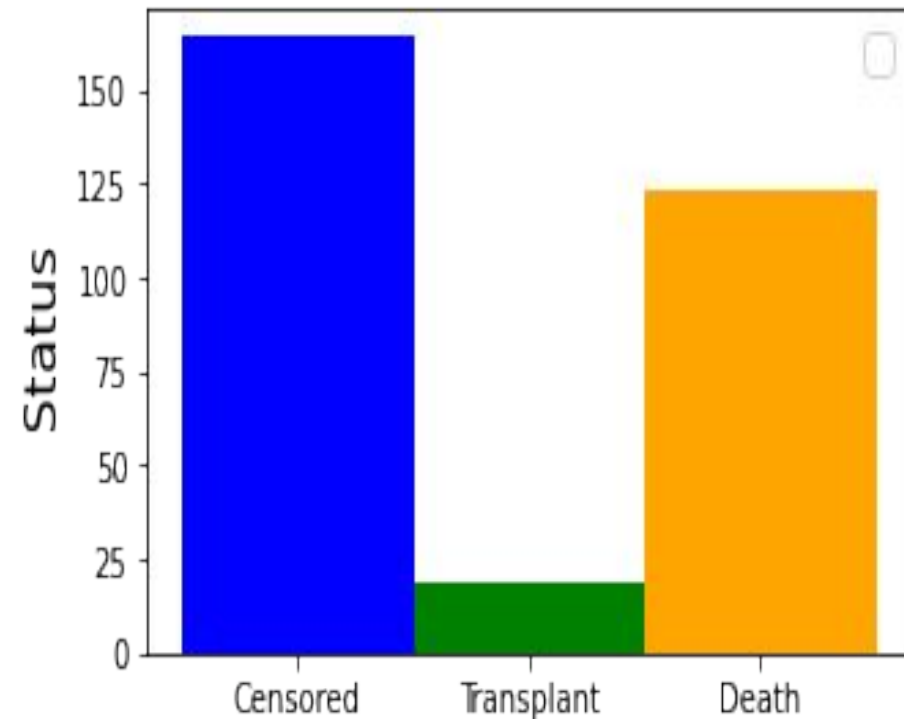
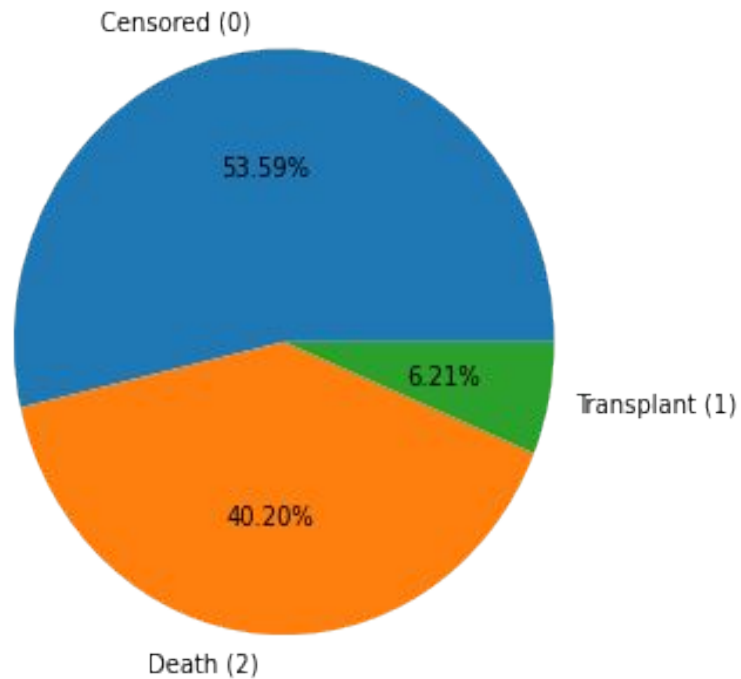
```
df.isnull().sum()
```

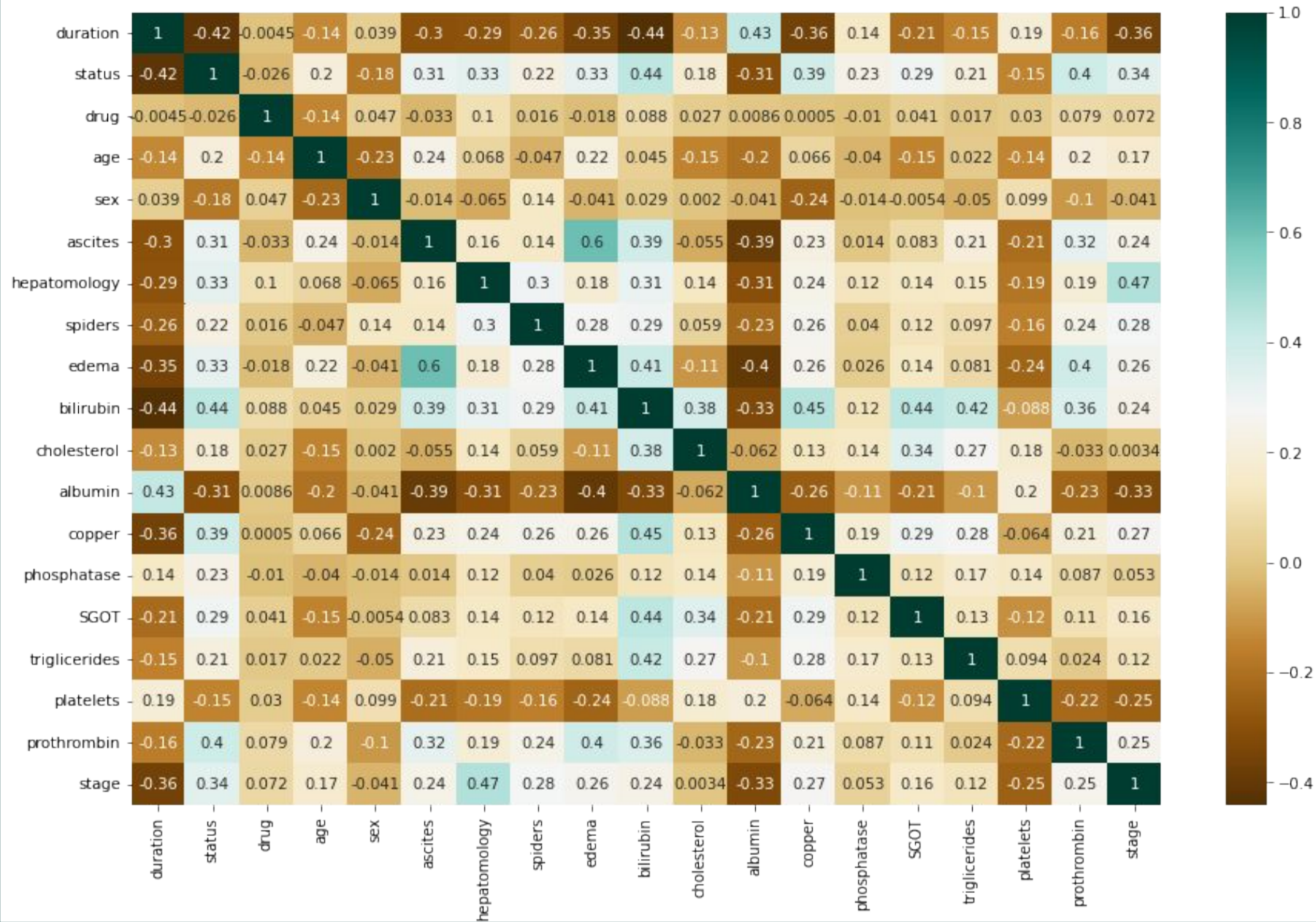
duration	0
status	0
drug	0
age	0
sex	0
ascites	0
hepatomology	0
spiders	0
edema	0
bilirubin	0
cholesterol	28
albumin	0
copper	2
phosphatase	0
SGOT	0
triglycerides	30
platelets	4
prothrombin	0
stage	0

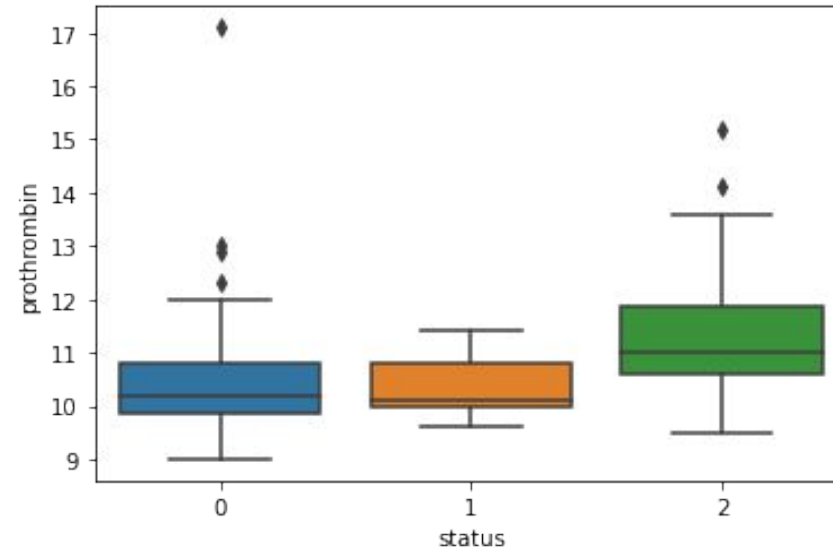
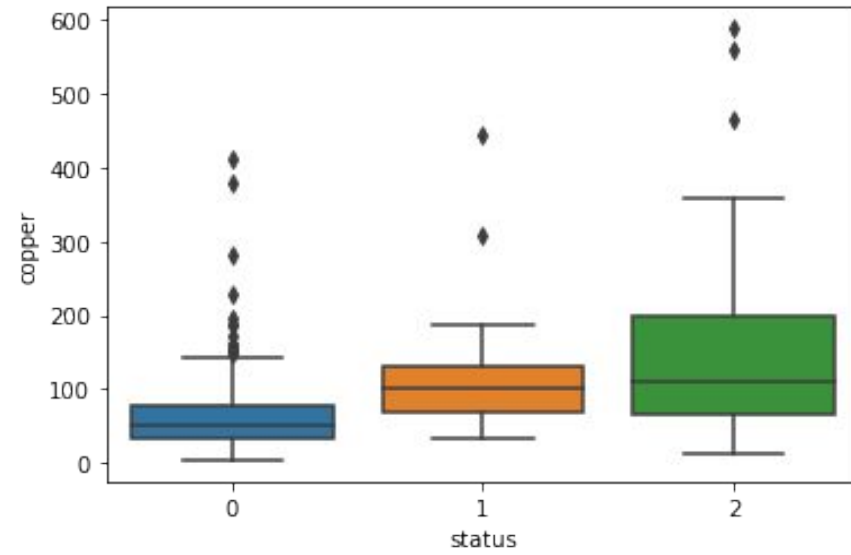
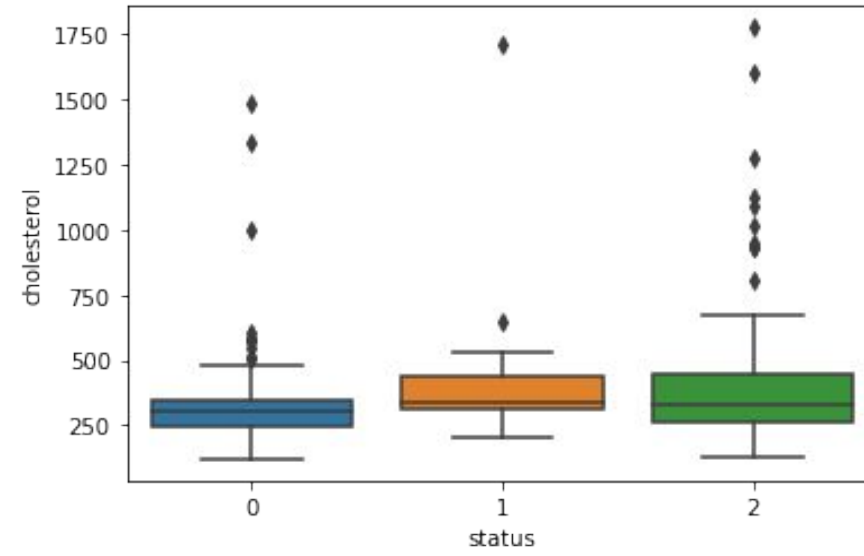
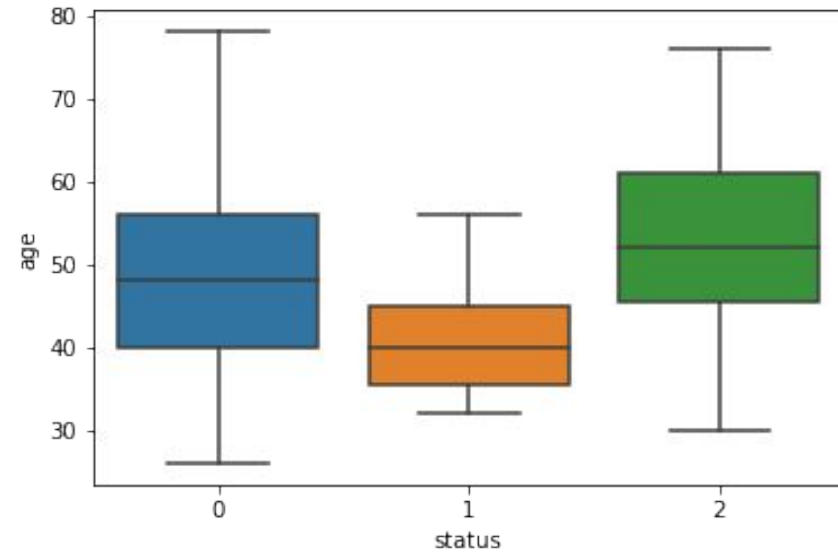
dtype: int64

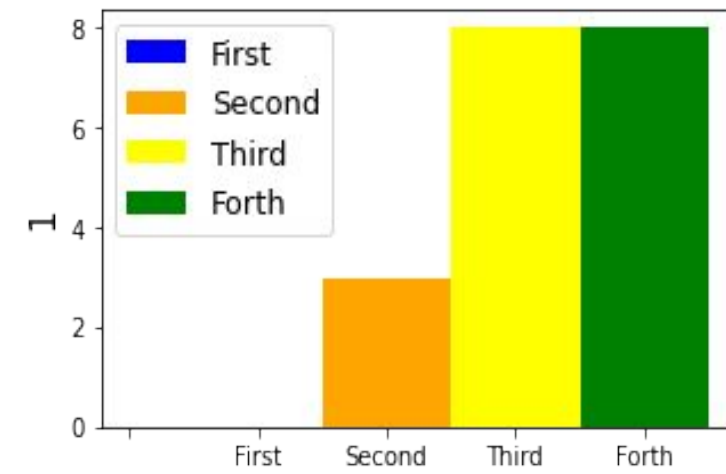
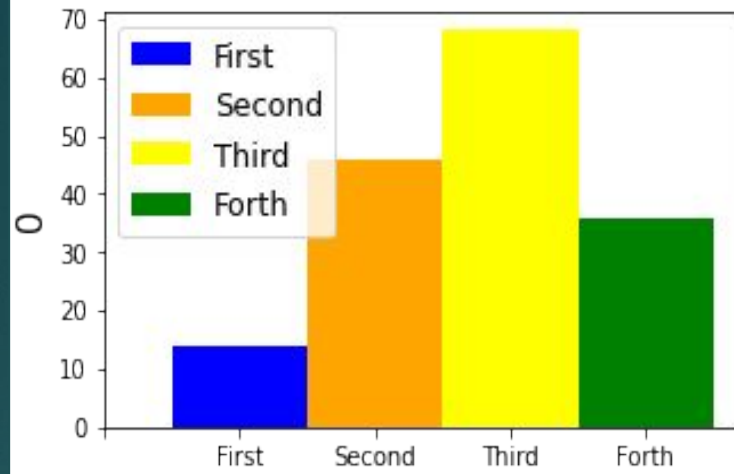
Taking a look to the target

4





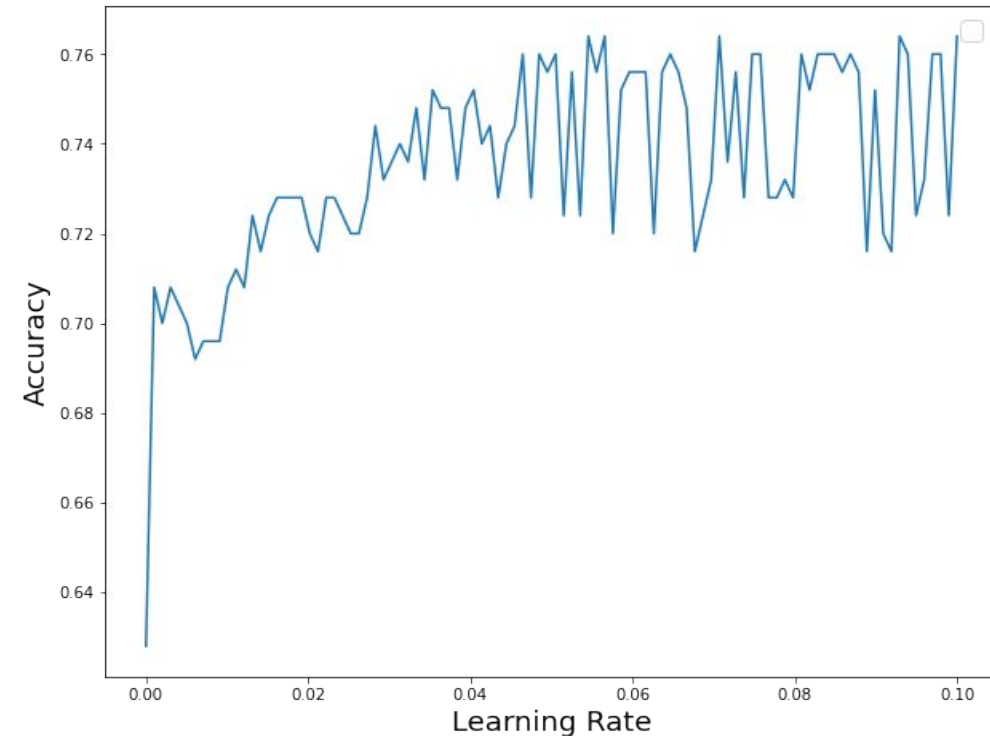
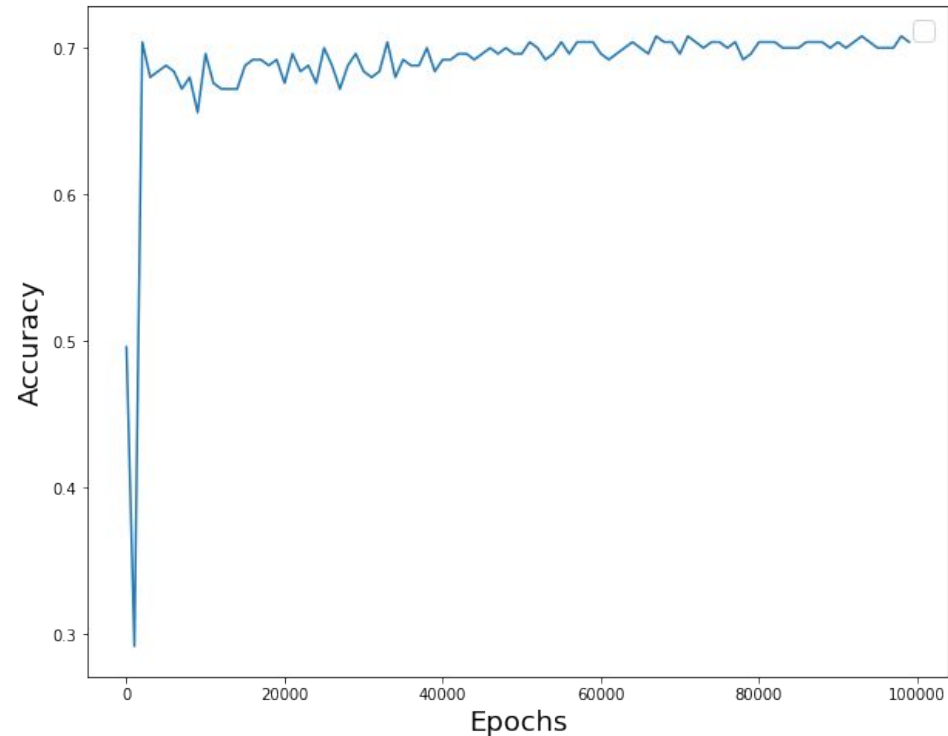




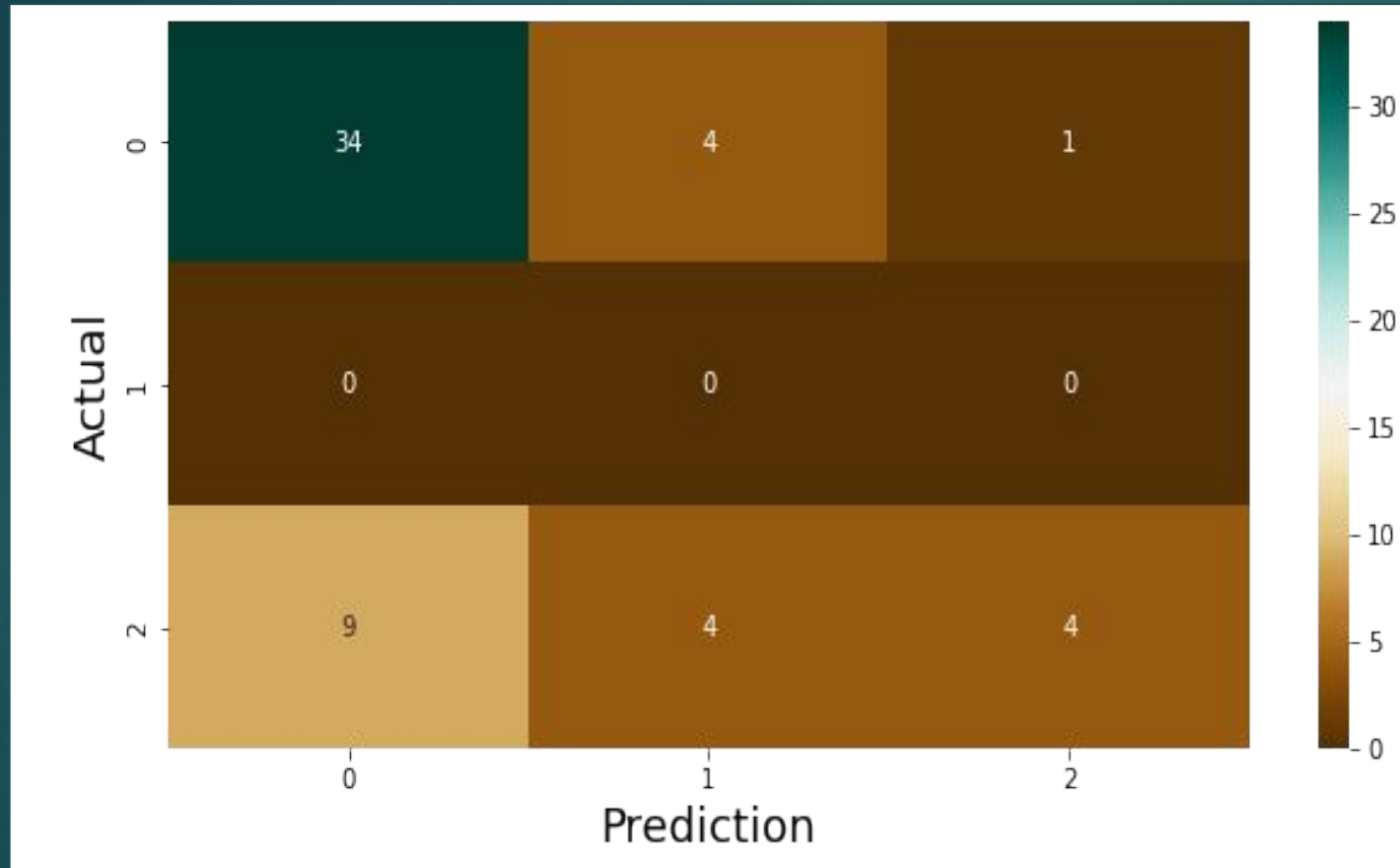
Training the model

8

It was used the Shallow ANN algorithm, and it was trained with 85% of the data. Along with the columns : hepatomology, copper, ascites, edema, bilirubin, albumin and prothrombin



It was obtained an accuracy of 72% for the Training data and 68% for the test data



	Actual	prediction
0	0	0
1	0	0
2	0	0
3	0	2
4	0	2
5	0	0
6	1	0
7	1	2

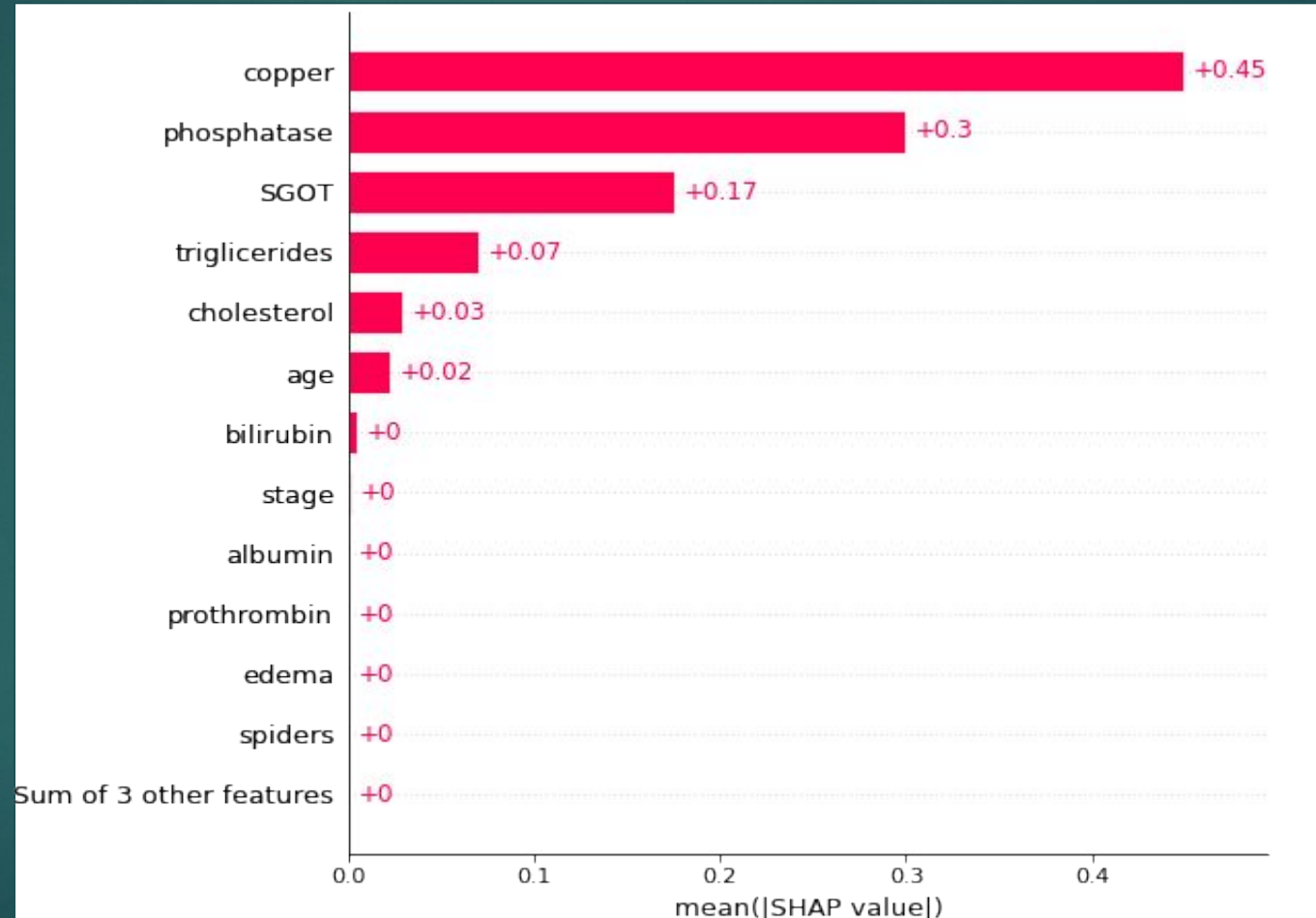
We didn't get the best results. Let's take a look to the probabilities

	0	1	2
0	0.839933	0.033979	0.126088
1	0.715131	0.048004	0.236865
2	0.726844	0.040524	0.232632
3	0.089604	0.040857	0.869539
4	0.356984	0.057773	0.585243
5	0.666264	0.046141	0.287596
6	0.755494	0.039824	0.204682
7	0.155768	0.055069	0.789163
8	0.340955	0.054416	0.604628
9	0.809370	0.038302	0.152327

Additional remarks

11

What is the effect of each variable in our model



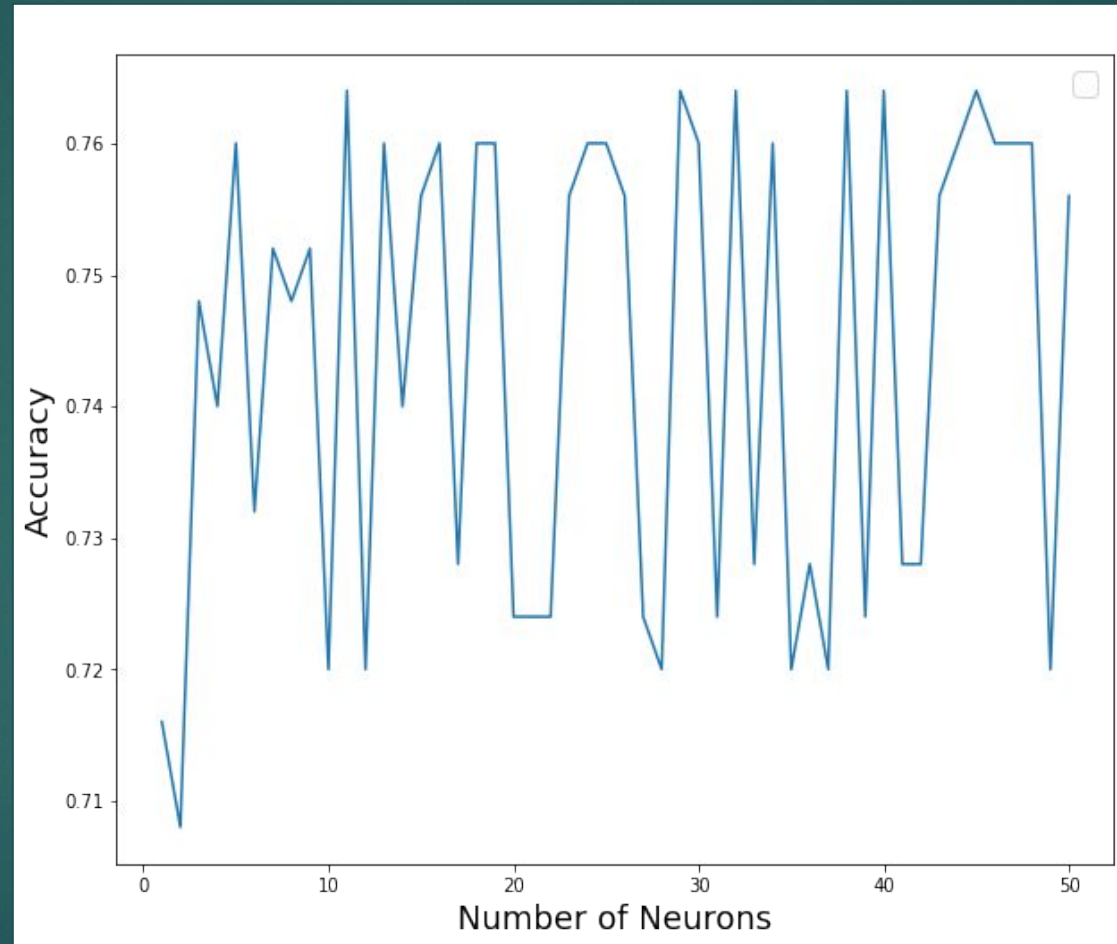
Conclusions

12

- ▶ It was found really interesting relations between the status target and some other variables during the EDA
- ▶ It was understood, by observing the confusion matrix and the probabilities dataframe, why the accuracy of the model was not very precise. We can be able of getting a better one by increasing the amount of essays.
- ▶ We were able to observe which variables are the ones which has more impact in our model

Appendix

13



```
def predict(self, X):  
    self.__forward__(X)#Printing the probabilities too  
    return self.Z[2].argmax(axis=1),self.Z[2]
```

```
#Prediction and probabilities with the test data  
y_hat_test, probs_test = ann.predict(X_test)
```

```
#Accuracy of the test data  
accuracy(y_test,y_hat_test)
```

```
0.6785714285714286
```

```
#Creation of a data frame of the probabilities  
result=pd.DataFrame(probs_test)  
result.head(10)
```