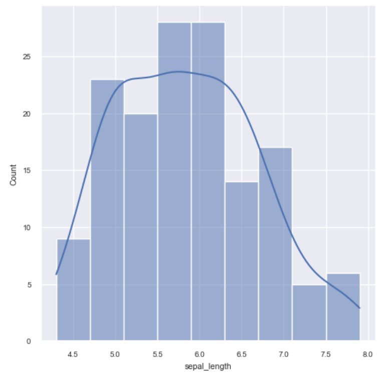
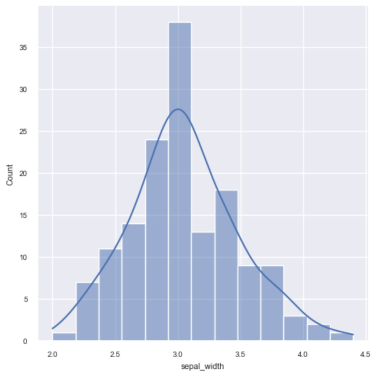
Data Science and AI
Mini Project - Individual
MERLIN Nicolas 73204

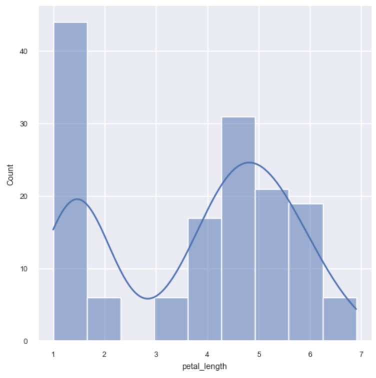
Link to the Github:
https://github.com/Eragon783/PythonIndividualProject

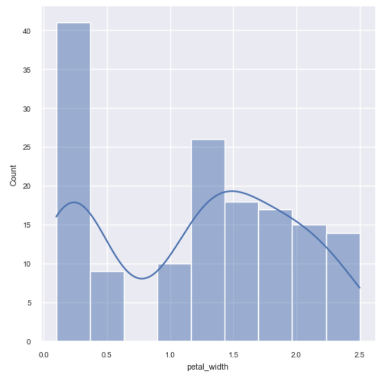
First solution: IRIS Flower

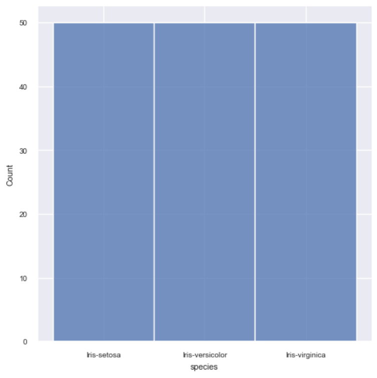
The database is **already perfect** : there are no null or strange values and all the columns are useful for the model. We can directly split the data and start training the model. Here is the result of the data analysis:











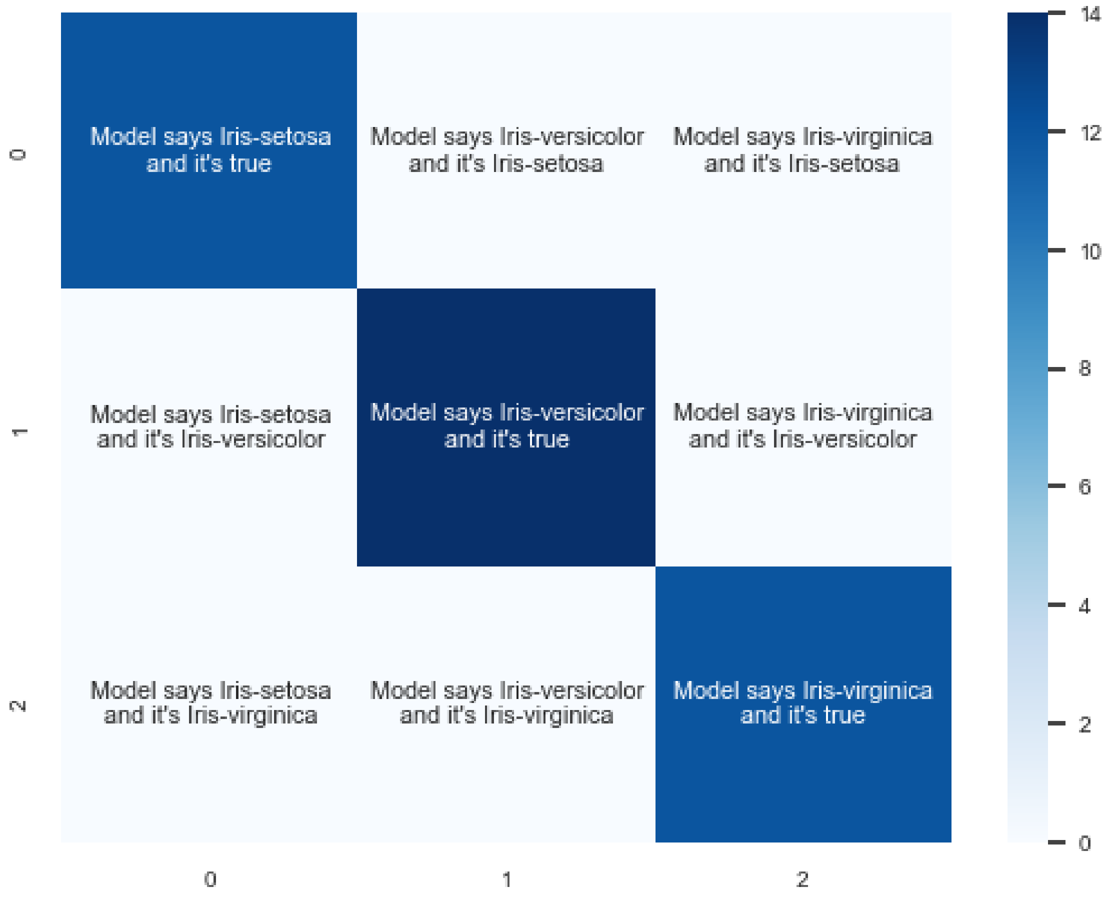
I used the **SVM algorithm** as requested in the statement to build my model because this is the one we use in class. We obtained a score of **0.973**.

To improve the model performance, I choose a test size of **0.25** because it improves the score of the model. Moreover, I test several value for the hyper-parameter:

- **kernel**: I try different values of the kernel type to be used in the model : “rbf” (the default value) and “linear” (the value we use I class)
- **C**: same find for the regularization parameter, I try 1, 10, 100 and 1000

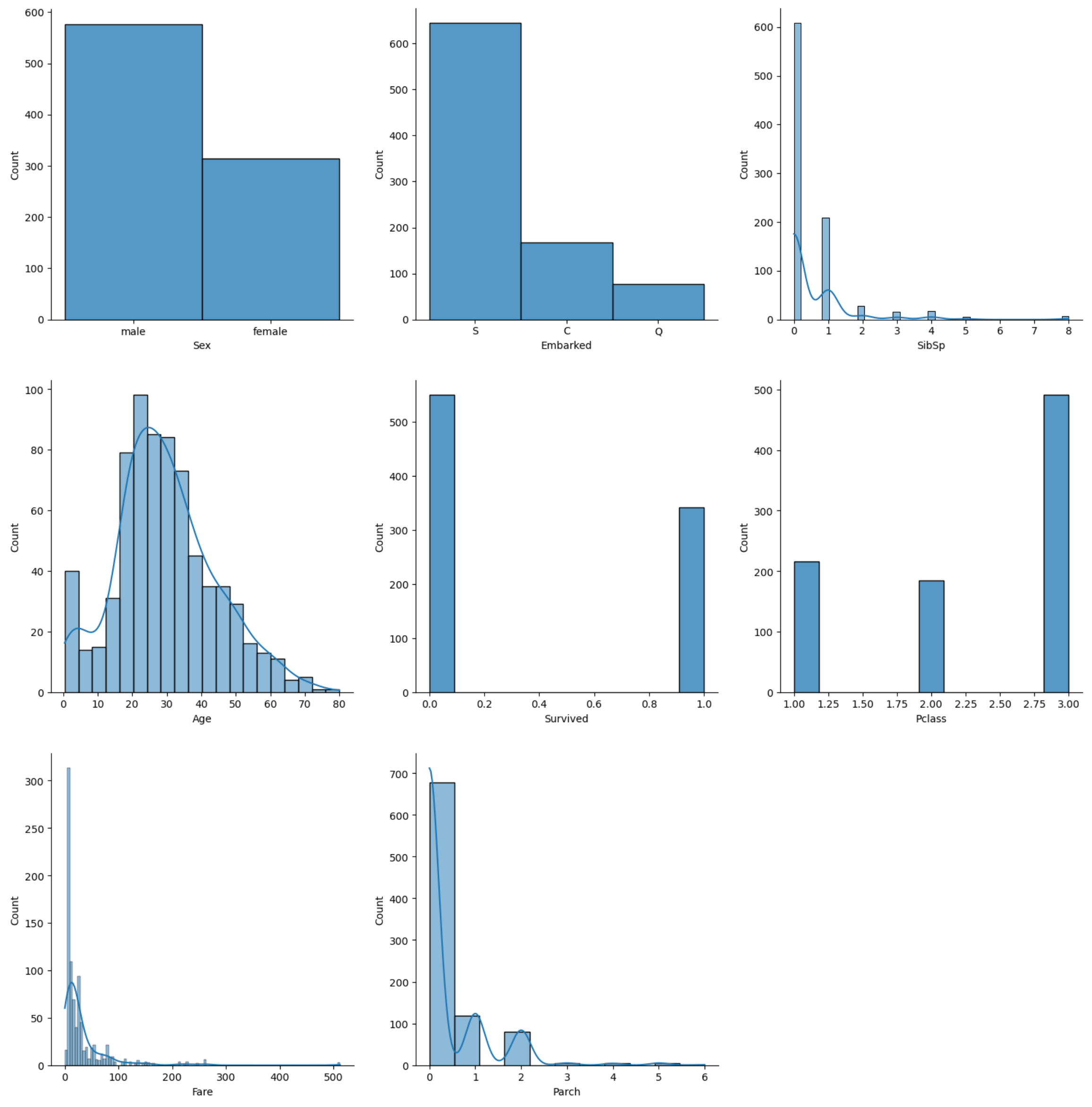
After the performance improvement by hyper-parameter modification, the model has a perfect score of **1**. Here is the confusion matrix:

```
[[12  0  0]
 [ 0 14  0]
 [ 0  0 12]]
```



Second solution: Titanic

Here is the result of the data analysis:



The database **needs cleaning** : there are some null values on the columns “Age” (177), “Cabin” (687) and “Embarked” (2) that we need to consider.

I have deleted the “Cabin” columns because there are too many null values to be interesting. I have also filled the null values of the column “Age” by the median. Finally, I have deleted the two rows where the age is null.

I have also deleted the column “PassengerId”, “Name” and “Ticket” because they don’t help to predict if the passenger has survived. Our database **is now clean**, we can split the data and start training the model.

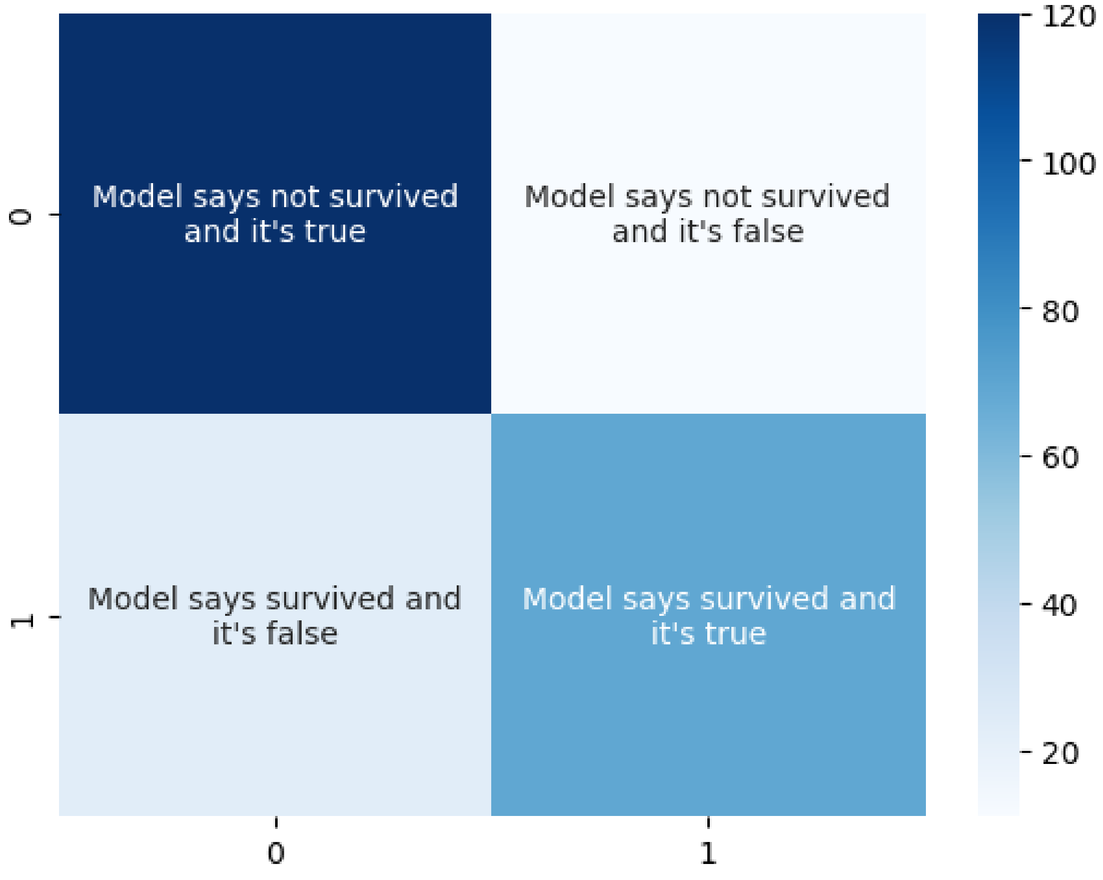
I used the **logistic regression algorithm** as requested in the statement to build my model. We obtained a score of **0.848**.

To improve the model performance, I choose a test size of **0.25** like before and for the same reasons. Again, I test several value for the hyper-parameter:

- **solver**: I try different values for the algorithm to use in the optimization problem : “lbfgs” (the default value) and “liblinear” (the value we use I class)
- **C**: same find for the regularization parameter, I try 1, 10, 100 and 1000

It seems that we already have the best model, the score that we obtained (0.848) is the best. Here is the confusion matrix :

```
[[12  0  0]
 [ 0 14  1]
 [ 0  0 11]]
```


Model says not survived and it's true
Model says not survived and it's false
Model says survived and it's false
Model says survived and it's true