Blandel Mathilde

Mercier Léa

ESIR 2 - Systèmes d'information

Université de Rennes 1

Ecole Supérieure d'Ingénieurs de Rennes

Année 2020 - 2021

Report of TP - Data Mining

TP 1 to 3

ECOLE SUPERIEURE D'INGENIEURS DE RENNES

UNIVERSITÉ DE RENNES 1

TP1 : A quick introduction

a) Introduction

This first part had the goal to establish our workspace environment. So we created a repository on gitlab and made our groups of TPs.

b) Tools and library which we are going to use

The interesting thing about this part is that we were able to use and test the different libraries we are going to use in the rest of the module.
And this tools and libraries are :

- **Anaconda**, this is the platform we used in the whole course which simplifies the setup of the whole environment of data analytics that we are going to use.
- **Python**, the language we used in the whole course. We used it since the beginning of the practicals with Jupyter Notebook. And to familiarize ourself with it we used the resources on the moodle, the official site of Seaborn, Matlplotlib, Scikit-learn and Pandas.
- **Jupyter Notebook**, more efficient for data analysis than an IDE. Thanks to it you run simple analysis, write your explanations in textual comments, and go through your dataset in an efficient and simple way.
- **Numpy**, it allow us to use multidimensional arrays and to do some algebra on it.
- **Pandas**, is a library that supports the use of structured data (dataframe). We are going to use it with Numpy along all our practicals.
- **Visualization**, more specifically Matlplotlib and Seabornwhich are the two libraries we are going to use for plots and data visualization.
- **Scikit-learn**, is a useful tool which while be helpful for implementing basic algorithms

But I am not going to give too much details because we are going to see it in the next TPs.

c) Conclusion

This first TP was really quick in comparison to the two others. We had already used some of the librairies during other modules like DataEngineering (Pandas, Numpy) and the language Python during our formation. But without that TP, we could not make the others, and it gives us a good overview of the tools in our hands.

TP2 : Prediction on Titanic dataset

a) Introduction

The goal of these practicals was to develop some techniques of data mining with whom we could predict if an individual could survive the sinking of the Titanic or not.

b) Data exploration, infrastructure and statistics

In this part we are going to familiarise ourself with the Pandas library. This one allows us to transform a dataset and to handle it.

At first we should import our data set into a global variable thanks to the library Pandas we saw earlier. We wanted to see all our available data but for that we should at first change the options of the data frame we just created. And after that, we could easily see all of them.
We could also want to display only the first few lines of our data frame, for this purpose we are going to use the ***head(n)*** command.

After that a quick look at our data set shows us that we have different types of data inside it. And with the command ***info()***, we can see that we actually have int64 for integer, object for character string, and float64 for decimal numbers. We also learn that our dataset has 891 entry/lines and twelves columns.

This time we want to know the percentage and the number or survivors in fonction of their caracteristics : age, sex and their class on the boat. This gave us this tablature :

	Type	Number	Pourcentage
0	Female	233	26.15
1	Male	109	12.23
2	Children	39	4.38
3	Adolescent	83	9.32
4	Adult	220	24.69
5	1st class	136	15.26
6	2nd class	87	9.76
7	3rd class	119	13.36

We can see that the female is the category which has the most luck to survive. This can maybe be explained by the fact that they were evacuated in priority.

We can also make some simple queries which can allow us to know the number of survivors who were under the age of fire when the accident occurred. And the answer to this
We have some missing values too. To be more specific we have 177 missing people who did not give their age and 687 who did not give their cabin.

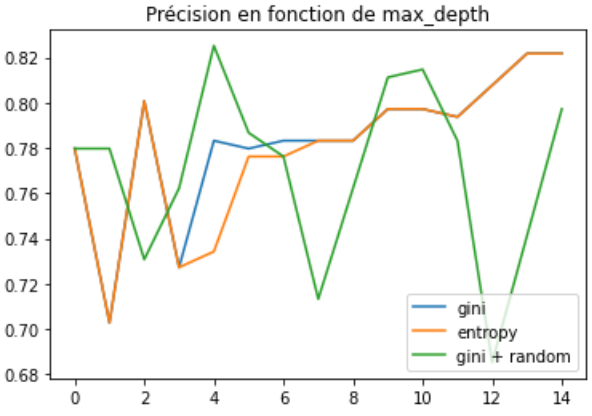
c) Features selection and Classification

Now that we have all the data we want to analyse them and create a subset of data to predict the life or death of some potential subjects.

The first step will be to create a data train to train our model on it. And since the more importantes datas to survive the sinking of the Titanic are the age, the gender, the ticket class of the personne and if this person survived or not.
We also change the gender column into numerical values to be able to reevaluate them. So from now on, 1 is for men and 0 for women.
And for the case of null values we just discard them and their lines.

We are going to use a decision tree. For this, we start to divide our data in two parts. The column 'Survived' will be the information to predict and the three others the criterias.
Testing our model on the datas we used to train it is a bad idea, because we will always predict the right result. This is why we divide our datas a second time, the subset X_train will be used to train our decision tree and the subset X_test (40% of the data) will be used later to test it.

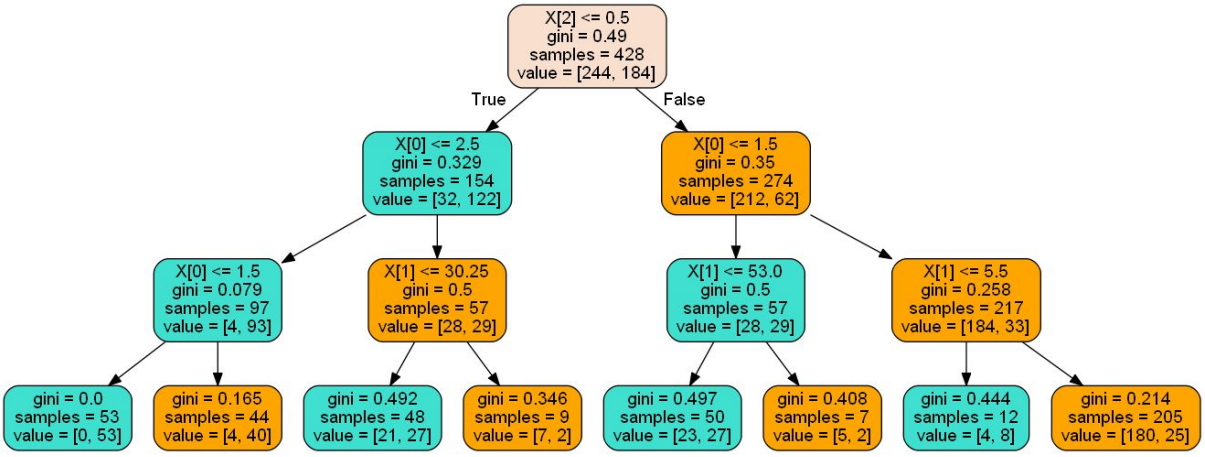
After that, we can use our model to predict the data. To be able to compute the accuracy of our model, we use it on the subset that we kept apart, and we compare the result with the expected one.



Parameters of the decision tree classifier :

- if we increase max_depth, our model became more complex
- for this set, gini is more accurate than entropy
- Splitter is the strategy we chose to create the nodes. The default strategy is "best" and chooses the best division. Random on another hand, choose the best random division. Globally, "best" is better when max_depth increases.

We used pydotplus to print our decision tree



On each node, we can see :

- the rule which had been used
- samples : the individus number
- value [nb dead/nb living] in our population

The first node corresponds to the gender of the individu, and we are going to divide the individuals at this stage. The men (1>0.5) will be at the right (False) of the tree and the women at the left (True). When we went down to the left, we can see that there were 154 women and how many of them died (32) and survived(122). We continue with the ticket class, and the new rule which divides the firsts class from the seconds and thirds ones. We kept going until we reached the leaf of the tree.
These rules have been learned by the tree during the training. The rules of classifications at the top are the most significant ones.

d) K-fold cross validation

Beforehand we decided to divide the data set into two different subsets. In the k-fold cross validation the datas are divided into k parts, the k-1 parts will be used to train the model and the last one to test it.
To get the accuracy of this method, we will make the mean of each result computed in the loop which will allow us to get our final result.
And in the end the average accuracy of DTC is 88.22, of LR is 78.71, of GNB is 78.14, and of RFC is 82.37.

e) Improving the predictions

At this point, our goal was to improve the accuracy of our model and for that, we had two potentials answers :

Missing values :
The precision of LR depends on the quality of the data so we can try to increase the quality of our data by using the lines with missing values instead of dropping them. We replace the missing values by the average age and we start again the tests. And then we discovered that the precision of LR and GNB had increased but the one of DTC and RFC had lowered.

Finding new rules:
The second option was to find new rules so we decided to add Embarked. We observed that the port of embarkation also influenced the probability of surviving the sinking. This addition has an impact on accuracy of all models.

f) Classify the test set

This time, we are going to use the whole model to train the Logistic Regression model and another set of data for the tests. As we should predict the results for all the passengers, we have decided to keep the method which replaces the missing values with the mean of all the values in the same column. And since we discovered that we have a low accuracy, we decided to add the Embarked rule. And all of that gives us an accuracy of 94.26.

g) Conclusion

In conclusion, this practical was the first real one, since the first one was just a familiarisation with the different tools and the environment.

TP3 : Prediction of a person income

a) Introduction

This last TP has a lot in common with the Titanic one. Thanks to that it was quicker to do and we could focus more on the analytic part. So, you would find out that we use a lot of techniques which came for the previous TP.

b) The Census Income Dataset

In this first part, we downloaded the census data, also known as the Census income dataset, to be able to use it in the rest of the TP with Pandas like before.

We did almost the same as the precedent part, so we are not going to repeat ourself.

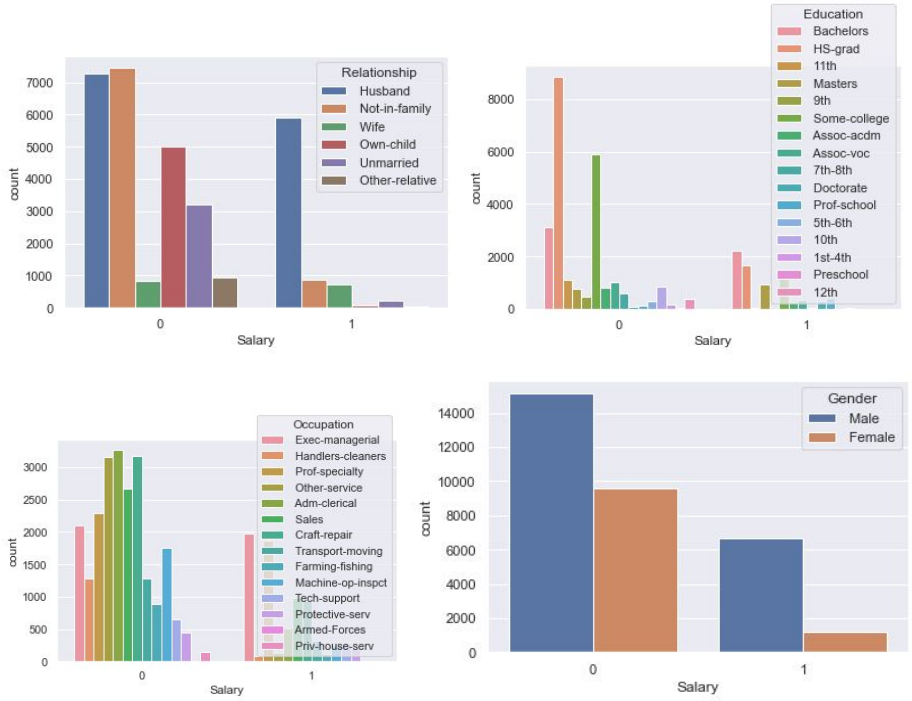
c) Data exploration and analysis

Now, we are going to explore a little bit more of our data. So we put each set of data into a dataframe and read it.

Thanks to that, we can say that the train-data has 488400 elements, and the test_data has 244200 elements. Both of them are composed with int64 for numericals values and object for the string of characters. We can also add that in the train_data set we have 1836 missing values for the column WorkClass, 1843 for the column Occupation, and 583 for the NativeCountry column. When for the test_data we have 963 missing values in the WorkClass column, 966 in the Occupation column, 274 in the NativeCountry column. What gave us 2399 rows with missing values in the train_data set against 1221 in the test_data. But that gave almost 7% of missing values in both of the data sets.

After that, we created a new column “Salary” which is going to act as a label for the “Income” column. When the column “Income” is equal to “<= 50K” Salary take the value 0, 1 elsewhere (so when “Income” is equal to “> 50K”).

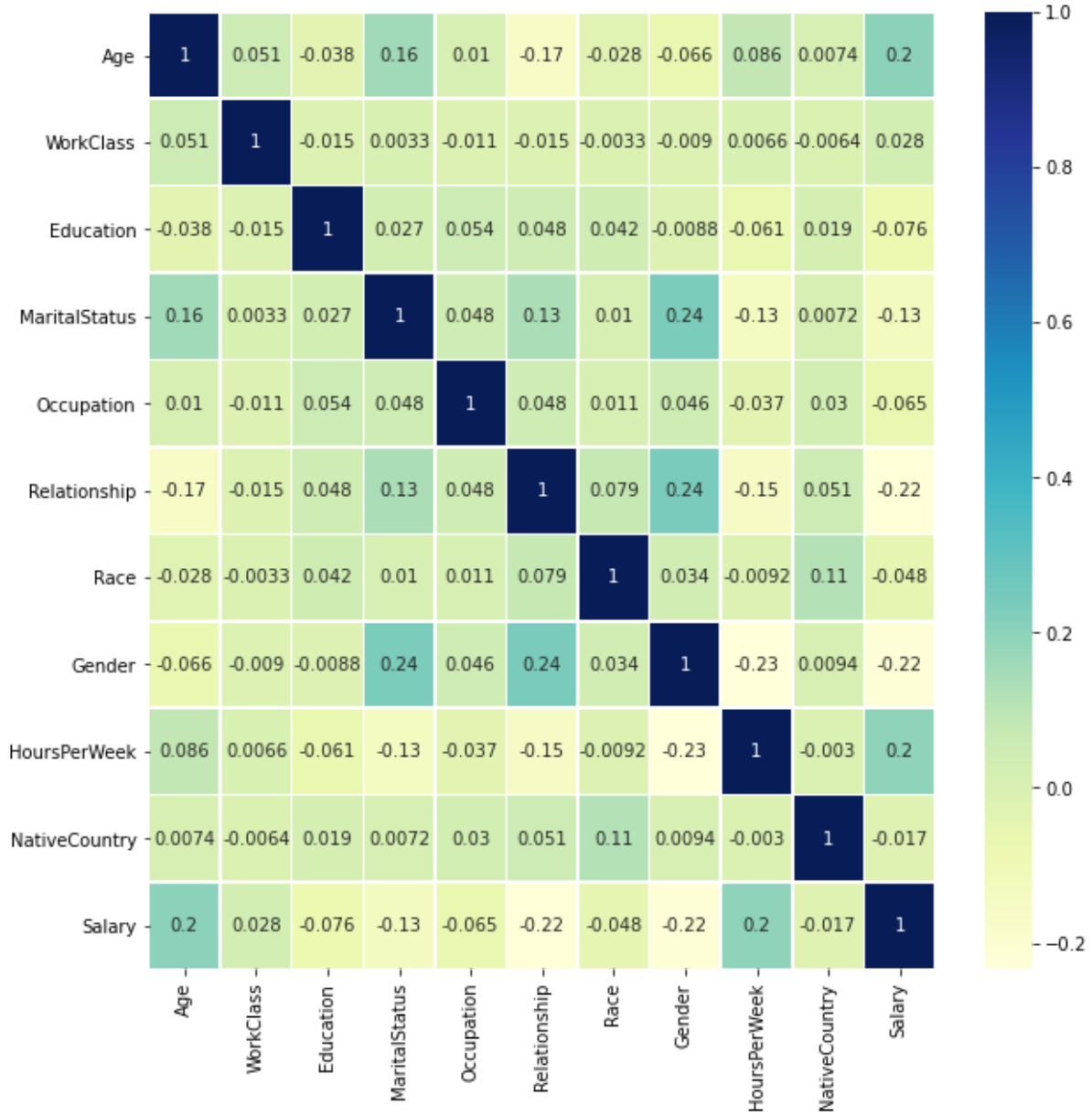
When we have this new column we plot all our attribute, like you can see on our result and on some examples below :



We can see in these few graphs that the people who win more than 50k.

d) Feature Engineering

Now, we are going to adjust the variables of our data set to make their use more easy. For this purpose we had converted continuous attributes into categorical attributes and string of characters into numericals values. We also drop the rows with missing values, because it was the easiest and quicker way to do it. We also drop the column of the Income, the Index, the, and . Then we plot the Person correlation for all the attributes using Seaborn heatmap. Which gave us the graph below :



By reading this graph we can see the occurrence of the correlation of the different variables of our data set.

e) Model selection

For this part we used the k-fold cross validation to evaluate various classifiers. We used for of them :

- the decision tree classifier
- the logistic regression classifier
- the gaussian NB classifier
- and the random forest classifier

We trained all for them.

f) Classification on the test dataset

In this part, we did exactly the same as in the previous practical. But this time we removed the fnlwgt column to increase the accuracy of our result. And this was quite effective. That allowed us to discover that the decision tree classifier and the random forest classifier were more accurate. When on another hand, the Gaussian NB was quite ineffective.

g) Conclusion

In conclusion, in this TP we used some of the methods we saw in class to achieve a prediction like in the previous practical. But we used new features like the heatmap, which allow us to describe our data and analyses in a better way.