MA2823 : Introduction to Machine Learning
CentraleSupélec
Assignment 2

Geoffroy DUNOYER, Magali MELLON, Jean-Baptiste GOURLET, Guillaume GRIEHSER
Kaggle Team Name : **Club Voile x Raid CS**

December 3, 2018

1 Feature engineering

1.1 Feature Selection and creation

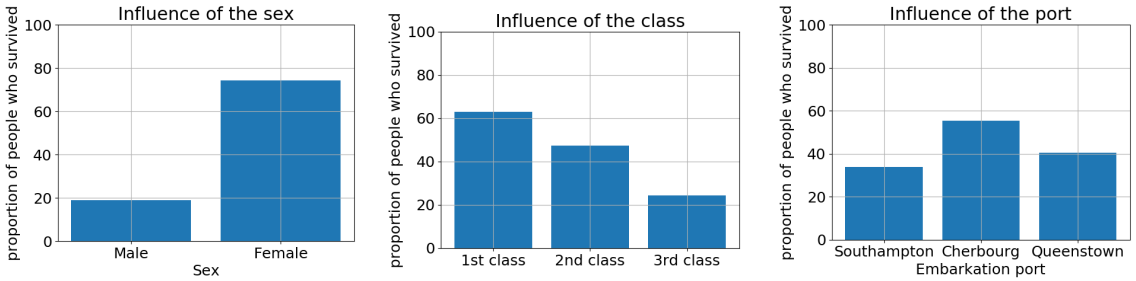
To select or not a feature and to see if it was interesting to create ones, we proceeded label by label, following our intuition to test things on the training data set. We evaluated for each label the influence that it could have on the final result: the death or the living of the people. It is possible that there is an automatic way to it but we figured out our algorithms better performed after a rigorous preprocessing.

First thing to consider is this figure : on our training dataset, 62% of people died. So when we consider a parameter, we have to analyse if it makes a significant difference with that figure.

1.1.1 Features we selected

Here are the figures we decided to select:

- **sex**: being a man or a woman has a significant influence because women took precedence over men.
- **passenger class**: As we thought, the class is important because it represents the place of the cabin.
- **port of embarkation**: It seems that there is a relation between the port of embarkation and the result, even if it appears surprising to us.
- **age**: The age is a good feature because the risk pf surviving is linked to the physical condition. If the age is not known we use the average of the other person.
- **fare**: The fare is a good way to distinguish because it is an important characteristic of the social class.

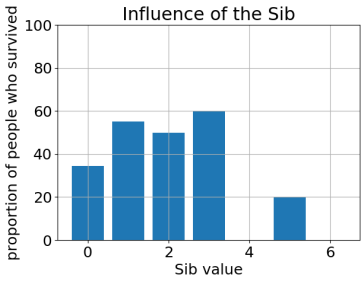


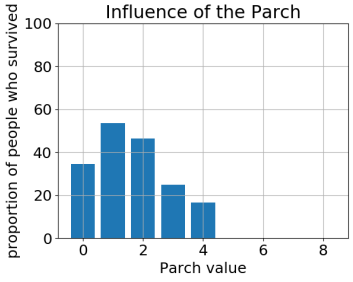
1.1.2 Features we discarded

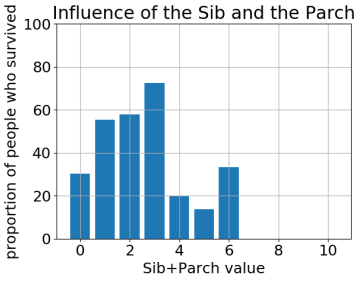
We decided to discard several figures for different reasons : they had no influence on the result, they could not be exploited by a Machine Learning program or there was a redundancy with a feature we already selected. Here are the features we finally decided to discard:

- **passenger id**: given randomly, no influence on the result
- **name**: the name of the person could not be exploited. Nevertheless, the title before the name could be useful, we'll analyse it later.

- **Siblings/Spouses and Parents/children** Alone, they are not very useful, as you can see on the following figures. But maybe, we can do something with these two features.



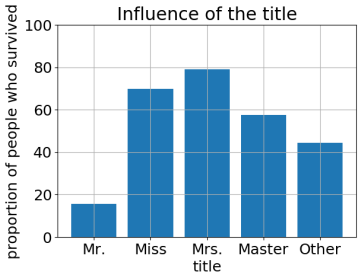


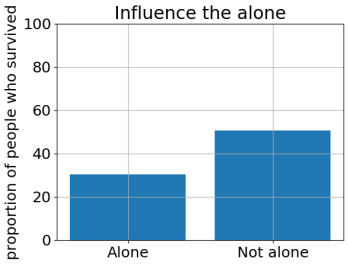


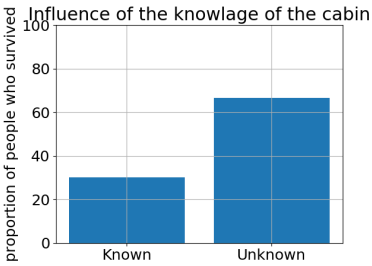
- **ticket Number**: our program cannot anything with it.
- **fare**: Fare and class are highly related. We chose to keep the class, because of its discrete aspect. But in practice, our experiments shown that keeping also the fare did not modified the results of our algorithm.
- **cabin number**: the cabin number is not exploitable, but we still can do something differently with it.

1.1.3 Features we created

- **title**: For the title we only distinguish 4 types and a good part of the title is linked to the sex or the social status.
- **alone**: This feature as been created to synthesize the date from the parch and the siblings as you can see on the figure the difference is huge.
- **cabin known**: Instead of using the cabin number we use only the fact if the number of the cabin in known.







1.2 Influence of features combinations

The task of feature selection selection was principally done testing label by label for each model. Once again, it could have been optimised but considering the size of the data, it has been possible. Here I describe, how the selection has been done, considering all the classifiers we used.

1.2.1 Unavoidable features

Here are the features we used in every model because of their influence on the results.

- **gender**
- **port of embarkation**
- **passenger class**
- **cabin known**

1.2.2 Additional features

The following features are use just for some classifier.

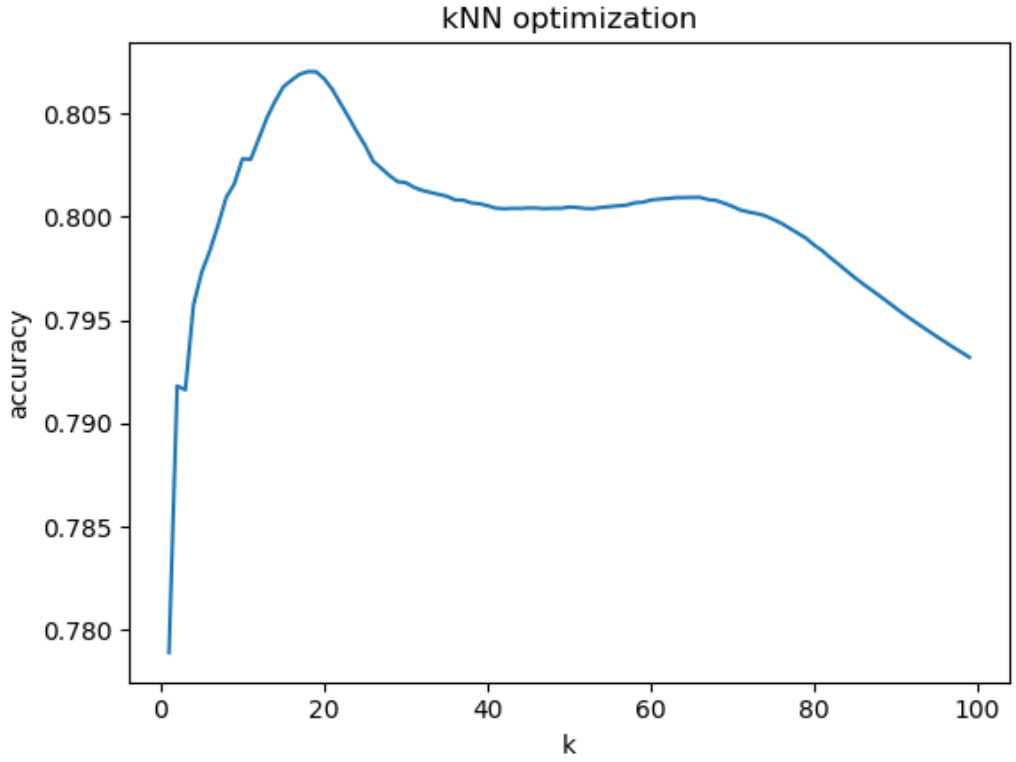
- **title**
- **age**
- **alone**

2 Model Tuning and comparison

2.1 Classifier comparison and optimisations

classifier	features we selected	parameters	cross-validated perfor-mance	Kaggle score
k-nearest neighbors	class - sex - cabin - port - title	k = 19	0.789	0,807
logistic regression	class - sex - age - cabin - port - title	set to default	0,792	0,765
random Forest	class - sex - cabin - port - title	max depth = 5	0,824	0,789
boosting	class - sex - fare - cabin - port - title	max depth = 7	0,829	0,785
decision tree	class - sex - cabin - port - title	max depth = 5	0,829	0,789

2.2 kNN overfitting



Here is the way we selected the number of neighbour. We can see that 19 is the best optimisation. It is an example of our way of working with classifiers.

2.3 Improvements

We can introduce some features we did not use by finding a way to improve there use. For example we can use the age by creating categories.

We try to use a neuron network. The results are correct about 80 %. But the technical is better for more complex problems with more data. In this case it is not useful to take a such powerful method for this exercise.