Week 6 - Session 1: Performance metrics

CM 1h, TP 2h

-- **Performance metrics**

Suppose we have this dataset with features from several clients from a bank:

- Is the client a student? (Qualitative variable)
- Monthly Credit Card Balance
- Annual Income

And we want to predict whether each client will **default**, that is, if they are going to pay their credit card bills or not.

> Show PowerPoint slide with the dataset

We can train a **logistic regression** that will give us an approximation to the Bayes classifier and use it to assign classes to data points as follows:

$$\text{If} \quad \mathbb{P}(Y_i = 1 \mid X = x_i) > 0.5 \quad \text{then} \quad \hat{y}_i = 1 \quad \text{else} \quad \hat{y}_i = 0$$

> Show the results with R

We estimate the accuracy of our estimation with the most intuitive quantity:

$$\texttt{Acc} = \frac{1}{N} \sum_{i=1}^{N} \mathbf{1}_{y_i = \hat{y}_i}$$

Which for the present dataset gives a seemingly high value of 97.1% !

Note, however, that the dataset has **much more** examples of class `NO` than `YES` and if we had designed a dummy classifier that gave `NO` to all observations, then we would have 96% of accuracy.

The initial result looks much less impressive, right?

How to analyse this situation and quantify what is going on?

We should recur to what people call a **confusion matrix**.

-- **Confusion Matrix**

In our present case, we have only two classes, so the confusion matrix is written as follows:

	Reference	
Prediction	No	Yes
No	9627	228
Yes	40	105

and we define four quantities:

- TN: True negative (9627 of them)
- FN: False negative (228 of them)
- TP: True positive (105 of them)

- FP: False positive (40 of them)

The **accuracy** that we had obtained before is defined simply as:

$$\texttt{Acc} = \frac{\text{TN} + \text{TP}}{\text{TN} + \text{FN} + \text{TP} + \text{FP}} = 97.32\%$$

Note, however, that if we were interested in checking how well the classifier **detected clients** that were really **in default**, we would prefer to check the quantity called "True Positive Rate", i.e.

$$\texttt{TPR} = \frac{\text{TP}}{\text{TP} + \text{FN}}$$

We note that for our current example, this value is rather small, only 31.5% For a bank that wishes to avoid losing too much money, this is rather poor performance. This quantity is also often called **sensititivy** or **recall** in other references from the literature.

Another quantity that might be of interest is **how many clients** the bank think will be **in default** when in fact **they are not**. This is what we call the "False Positive Rate", defined as

$$\texttt{FPR} = \frac{\text{FP}}{\text{TN} + \text{FP}}$$

which in our case is only 0.41%. This indicates how much the bank will loose for not accepting certain clients when in fact they would not have cause much problem.

> - In the end of the day, our goal is to have a **large** TPR and a **small** FPR.
> - Note, however, that in the case of banks, it is much **more costly** to have a weak true positive rate (TPR) than letting a false positive rate happen. This means that the bank would prefer to **control** this quantity when assessing the quality of its classifier implementation
> - In the case of radars in a nuclear war, a high false positive rate could have **catastrophic** consequences. For instance, in 1983 a radar from Soviet Union detected a US missile coming directly to Moscow. This could have lead to a quick retaliation and the beginning a full nuclear war. However, the soviet military in charge of the readings of the radar believed it was a false alarm (i.e. a false positive). His main argument was that if the americans were really going to attack the USSR, they would not do that with just a single bomb...
> Reference: https://www.bbc.com/news/world-europe-24280831

A **very important** thing to notice in our classification procedure is that we have used the value of 50% as threshold for deciding whether we should consider the output of our classifier 1 or 0. What if instead we had chosen a threshold of γ and let it vary between 0 and 100%?

- For lower values of γ the classifier will have more tendency to consider data points as from class 1 (i.e. TRUE), meaning that we would more easily detect clients that defaulted. Thus, we would have a higher TPR. However, this comes at a cost: we also get a larger FPR. We need **therefore** to choose a balance between these two quantities.
- In our example, we have that:
  - For $\gamma = 0.2$, $\texttt{TPR} = 60.9\%$ and $\texttt{FPR} = 2.86\%$
  - For $\gamma = 0.8$, $\texttt{TPR} = 9\%$ and $\texttt{FPR} = 0.04\%$

> Plot a curve for several choices of γ we get

Note that each one of this quantities is describing something different from the confusion matrix. It's as if they were **two independent axis** providing complementary information about the classifier.

> This is the ROC curve: present it and say that the area under this curve gives us a rather good measurement of error. Intuition for the random classifier.

In some contexts, one might want to work in terms of two other quantities:

- **Sensitivity** which is the same as the true positive rate, i.e.

$$\text{sensitivity} = \texttt{TPR} = \frac{\text{TP}}{\text{TP} + \text{FN}}$$

- **Specificity** which is defined as

$$\text{specificity} = \frac{\text{TN}}{\text{TN} + \text{FP}} = 1 - \texttt{FPR}$$

An example of such context would be **medical diagnosis** where a false negative (i.e. a patient that has cancer which is not detected) has important consequences.

- In the end of the day, we want to have **both** a large sensitivity and a large specificity.

> Show figure with the confusion matrix for COVID autotest

-- **Precision and recall**

Finally, in the context of **information retrieval** (i.e. document mining) it is very common to use two other metrics, called:

- **Precision** which is defined as

$$\text{precision} = \frac{\text{TP}}{\text{TP} + \text{FP}}$$

  and tells among all of the positive predictions from our classifier, how many of them were correct.

- **Recall** which is defined as

$$\text{recall} = \frac{\text{TP}}{\text{TP} + \text{FN}} = \texttt{TPR}$$

  and tells how many of the truly positive observations were indeed detected by our classifier.

It is important to note that as opposed to the two previous examples of metrics, we have a clear **tradeoff** relation when talking about precision and recall. For instance, take the example of the bank:

- If we use a classifier that labels **all of the clients** as positive (i.e. will default) then the recall of our classifier will be 1, since we recover all of the positive examples. However, this will lead to a very low precision, since there will definetly be several false positives among our findings.

> Show Figure with the curve for recall and precision

It is very common to **merge** recall and precision into a single number called the F-score, define as the harmonic mean of the two:

$$F = \left(\frac{1}{2} \times \left(\frac{1}{R} + \frac{1}{P} \right) \right)^{-1} = 2 \times \frac{RP}{P+R}$$

- Any **intuition on why** the harmonic mean? I will let you guys take a look, but this a more natural way of averaging rates than the usual arithmetic mean. For instance, suppose we have: $R = 90\%$ and $P = 30\%$. The arithmetic mean would give us 60% whereas the harmonic mean is 45%.
- Note that the F score (and precision and recall) can be sometime criticised for not using any information about the true negatives.

-- **Challenge Data ENS**

> Present Stéphane Mallat's course at Collège de France
>
> Talk about the course and the competitions (Why this one and not Kaggle?)
>
> Show some examples and the kinds of metrics that people use