FTML REPORT

Alexandre Lemonnier - Sarah Gutierez - Victor Simonin - Alexandre Poignant

2023 promotion

Table des matières

1 Bayes estimator and Bayes risk

We choose to consider the following setting :

- $\mathcal{X} = \{2, 4, 6\}$
- $\mathcal{Y} = \{1, 3\}$
- X follows an uniform law on $\mathcal{X}$
-

$$Y = \begin{cases} C(1/4) \text{ if } X = 2 \\ C(3/5) \text{ if } X = 4 \\ C(3/4) \text{ if } X = 6 \end{cases}$$

With C(p) a custom law with parameter p, such as :

$$P(X = x) = \begin{cases} p \text{ if } x = 3 \\ 1 - p \text{ if } x = 1 \\ 0 \text{ else} \end{cases}$$

We use the "0-1" loss $l(y, z) = 1_{y \neq z}$.

1.1 Bayes estimator

We have seen that the Bayes estimator is defined by :

$$f^*(x) = \arg\min E[l(y, z)|X = x]$$

Hence :

$$\begin{aligned} f^*(x) &= \arg\min E[l(y, z)|X = x] \\ &= \arg\min P(Y \neq z|X = x) \\ &= 1 - \arg\min P(Y = z|X = x) \\ &= \arg\max P(Y = z|X = x) \end{aligned} \tag{1.1}$$

The optimal classifier selects the most probable output given $X = x$.

1.1.1 Application

We have found f^*, here are some values we can find with it :
- $f^*(2) = 1$
- $f^*(4) = 3$
- $f^*(6) = 3$

1.2 Bayes risk

Let's now compute the Bayes Risk with the with the "0-1" loss :

$$\begin{aligned} R^*(x) &= E[l(Y, f^*(X))] \\ &= E_X[E_Y(l(Y \neq f^*(X)|X))] \\ &= E_X[P(Y \neq f^*(X)|X)] \end{aligned} \tag{1.2}$$

But we have :

$$P(Y \neq f^*(X)|X = x) = P(Y \neq f^*(x))$$

We note $\eta(x) = P(Y = 3|X = x)$. Then :
- If $\eta(x) > \frac{1}{2},$ then $f^*(x) = 3,$ and $P(Y \neq f^*(x)) = P(Y = 1) = 1 - \eta(x)$
- If $\eta(x) < \frac{1}{2},$ then $f^*(x) = 1,$ and $P(Y \neq f^*(x)) = P(Y = 3) = \eta(x)$

In both cases, $P(Y \neq f^*(x)) = min(\eta(x), 1 - \eta(x))$.
We conclude that

$$R^* = E_X[min(\eta(X), 1 - \eta(X)]$$

1.2.1 Application

In this settings :

$$\begin{aligned} R^* &= \frac{1}{3}\frac{1}{4} + \frac{1}{3}\frac{2}{5} + \frac{1}{3}\frac{1}{4} \\ &= \frac{1}{3}(\frac{1}{2} + \frac{2}{5}) \\ &= \frac{1}{3}(\frac{5}{10} + \frac{4}{10}) \\ &= \frac{1}{3}\frac{9}{10} \\ &= \frac{3}{10} \end{aligned} \tag{1.3}$$

2 Bayes risk with absolute loss

2.1 Question 1

In this question, we need to propose a setting where the Bayes predictor is different for the square loss and the absolute loss.

With the following setting :
- $\mathcal{X} = [0,1] / \frac{1}{2}$
- $\mathcal{Y} = \mathbb{R}$
- X follows an Uniform law on $\mathcal{X}$
- Y follows an Bernoulli law on X

With the first loss, which is the square loss, we obtain a Bayes predictor of :

$$f^*(x) = X$$

But with the second loss, which is the absolute loss, the Bayes predictor is :

$$f^*(x) = \begin{cases} 0 \text{ if } & x < \frac{1}{2} \\ 1 \text{ if } & x > \frac{1}{2} \end{cases}$$

2.2 Question 2

We need here to determine the Bayes predictor for a general case. Which means we need to determine :

$$\begin{aligned} f^*(x) &= argminE[|y-z||X=x] \\ &= argmin(g(z)) \end{aligned} \tag{2.1}$$

with

$$g(z) = \int_{y \in \mathbb{R}} |y-z| P_{Y|X=x}(y) dy$$

Let's try to develop g(z) :

$$\begin{aligned} g(z) &= \int_{y\in\mathbb{R}} |y-z|P_{Y|X=x}(y)dy \\ &= \int_{-\infty}^{z} |z-y|P_{Y|X=x}(y)dy + \int_{z}^{+\infty} |y-z|P_{Y|X=x}(y)dy \\ &= z\int_{-\infty}^{z} P_{Y|X=x}(y)dy - \int_{-\infty}^{z} yP_{Y|X=x}(y)dy + \int_{z}^{+\infty} yP_{Y|X=x}(y)dy - z\int_{z}^{+\infty} P_{Y|X=x}(y)dy \end{aligned} \tag{2.2}$$

We need now to find the derivative of g(z) to get the minimum :

$$\begin{aligned} g'(z) &= \int_{-\infty}^{z} P_{Y|X=x}(y)dy - \int_{z}^{+\infty} P_{Y|X=x}(y)dy \\ &= \int_{-\infty}^{z} |z-y|P_{Y|X=x}(y)dy + \int_{z}^{+\infty} |y-z|P_{Y|X=x}(y)dy \end{aligned} \tag{2.3}$$

We need to find the point where the derivative is null :

$$g'(z) = 0 \leftrightarrow \int_{-\infty}^{z} P_{Y|X=x}(y)dy = \int_{z}^{+\infty} P_{Y|X=x}(y)dy$$

We know here that the sum of the equation must be 1, so we obtain the minimum when the left part and the right part are equal to $\frac{1}{2}$. Here we can conclude that the Bayes predictor is equal to the median.

3 Expected value of empirical risk

3.1 Step 1

$$E[R_n(\overset{\wedge}{\theta})] = E_\epsilon[\frac{1}{n}||(I_n - X(X^TX)^{-1}X^T)\epsilon||^2]$$

$$\begin{aligned}
E[R_n(\overset{\wedge}{\theta})] &= E[\frac{1}{n}||y - X\overset{\wedge}{\theta}||^2] \\
&= E[\frac{1}{n}||y - X(X^TX)^{-1}X^Ty||^2] \\
&= E[\frac{1}{n}||(I_n - X(X^TX)^{-1}X^T)y||^2] \\
&= E[\frac{1}{n}||(X\theta^* + \epsilon - X(X^TX)^{-1}X^TX\theta^* + (I_n - X(X^TX)^{-1}X^T)\epsilon||^2] \\
&= E[\frac{1}{n}||(X\theta^* + \epsilon - XI_d\theta^* - X(X^TX)^{-1}X^T)\epsilon||^2] \\
&= E_\epsilon[\frac{1}{n}||(I_n - X(X^TX)^{-1}X^T)\epsilon||^2]
\end{aligned} \tag{3.1}$$

3.2 Step 2

Let A $\in \mathbb{R}^{n,n}$. Show that :

$$\sum_{(i,j)\in[1,n]} A_{ij}^2 = tr(A^TA)$$

$$\begin{aligned}
tr(A^TA) &= \sum_{i\in[1,n]}\sum_{j\in[1,n]} (A^TA)_{ij} \\
&= \sum_{i\in[1,n]}\sum_{j\in[1,n]} A_{ji}^TA_{ij} \\
&= \sum_{(i,j)\in[1,n]} A_{ij}A_{ij} \\
&= \sum_{(i,j)\in[1,n]} A_{ij}^2
\end{aligned} \tag{3.2}$$

3.3 Step 3

We need to show :

$$E_\epsilon[\frac{1}{n}||A\epsilon||^2] = \frac{\sigma^2}{n}tr(A^TA)$$

In the previous step we proved that $\sum\limits_{(i,j)\in[1,n]} A_{ij}^2 = tr(A^TA)$. So we have :

$$\begin{aligned}\frac{\sigma^2}{n}tr(A^TA) &= \frac{1}{n}E_\epsilon[\epsilon^2] \sum_{(i,j)\in[1,n]} A_{ij}^2 \\ &= \frac{1}{n}E_\epsilon[\epsilon^2 \sum_{(i,j)\in[1,n]} A_{ij}^2] \\ &= E_\epsilon[\frac{1}{n} \sum_{(i,j)\in[1,n]} (A_{ij}\epsilon)^2] \\ &= E_\epsilon[\frac{1}{n}||A\epsilon||^2]\end{aligned} \tag{3.3}$$

3.4 Step 4

Let's show that with $A = I_n - X(X^TX)^{-1}X^T$ then $A^TA = A$

$$\begin{aligned}A^TA &= (I_n - X(X^TX)^{-1}X^T)^T(I_n - X(X^TX)^{-1}X^T) \\ &= (I_n^T - X^{T^T}(X((X^TX)^{-1})^T)(I_n - X(X^TX)^{-1}X^T) \\ &= (I_n^T - X((X^TX)^{-1})^TX^T)(I_n - X(X^TX)^{-1}X^T) \\ &= I_n - X(X^TX)^{-1}X^T - X((X^TX)^{-1})^TX^T + X((X^TX)^{-1})^TX^TX(X^TX)^{-1}X^T \\ &= I_n - X(X^TX)^{-1}X^T - X((X^TX)^{-1})^TX^T + X((X^TX)^{-1})^TX^T \\ &= I_n - X(X^TX)^{-1}X^T \\ &= A\end{aligned} \tag{3.4}$$

3.5 Step 5

What we want to prove is

$$E[R_X(\overset{\wedge}{\theta})] = \frac{n-d}{n}\sigma^2$$

From the previous steps we know that $A = I_n - X(X^TX)^{-1}X^T$, which is the projection matrix of X. And we also showed that : $E_\epsilon[\frac{1}{n}||A\epsilon||^2] = \frac{\sigma^2}{n}tr(A^TA)$.
This results are going to help us to prove what we want :

$$\begin{aligned}
E_\epsilon[\frac{1}{n}||A\epsilon||^2] &= \frac{\sigma^2}{n}tr(A^TA) \\
&= \frac{\sigma^2}{n}tr(A) \qquad \text{(see step 4)} \\
&= \frac{\sigma^2}{n}tr(I_n - X(X^TX)^{-1}X^T) \\
&= \frac{\sigma^2}{n}tr(I_n - X(X^TX)^{-1}X^T) \\
&= \frac{\sigma^2}{n}(tr(I_n) - tr(X(X^TX)^{-1}X^T)) \qquad \text{because tr(A + B) = tr(A) + tr(B)} \\
&= \frac{\sigma^2}{n}(tr(I_n) - tr(X^TX(X^TX)^{-1})) \qquad \text{because trace is circular} \\
&= \frac{\sigma^2}{n}(tr(I_n) - tr(I_d)) \\
&= \frac{n-d}{n}\sigma^2
\end{aligned} \tag{3.5}$$

4 Regression

In this section we had to perform a regression on a given dataset, with our inputs stored in a *inputs.npy* file and our label in a *labels.npy* file.

We were free to choose our regression method and free to implement what we wanted, so we decided to test several algorithms exposed by the Sklearn Api and compare them.

The seven tested algorithms are :

- Support Vector Regressor
- Gradient Boosting Regressor
- ElasticNet
- Bayesian Ridge
- Kernel Ridge
- Linear Regression
- XGBoost Regressor

We decided to split our dataset to have a test dataset and a train dataset, to run these algorithms with default hyper-parameters and to evaluate them with the R2 score. The objective was to obtain a R2 score superior than 0.84 on the test subset. With our basic implementation, we had great results that were superior than 0.84 on several of our tests.

Here is the benchmark results :

Regression Algorithms tested	R2 score obtained
BayesianRidge	0.8923
KernelRidge	0.8921
LinearRegression	0.8915
XGBoost	0.8409
SVR	0.7022
GradientBoosting	0.8688
ElasticNet	0.8917

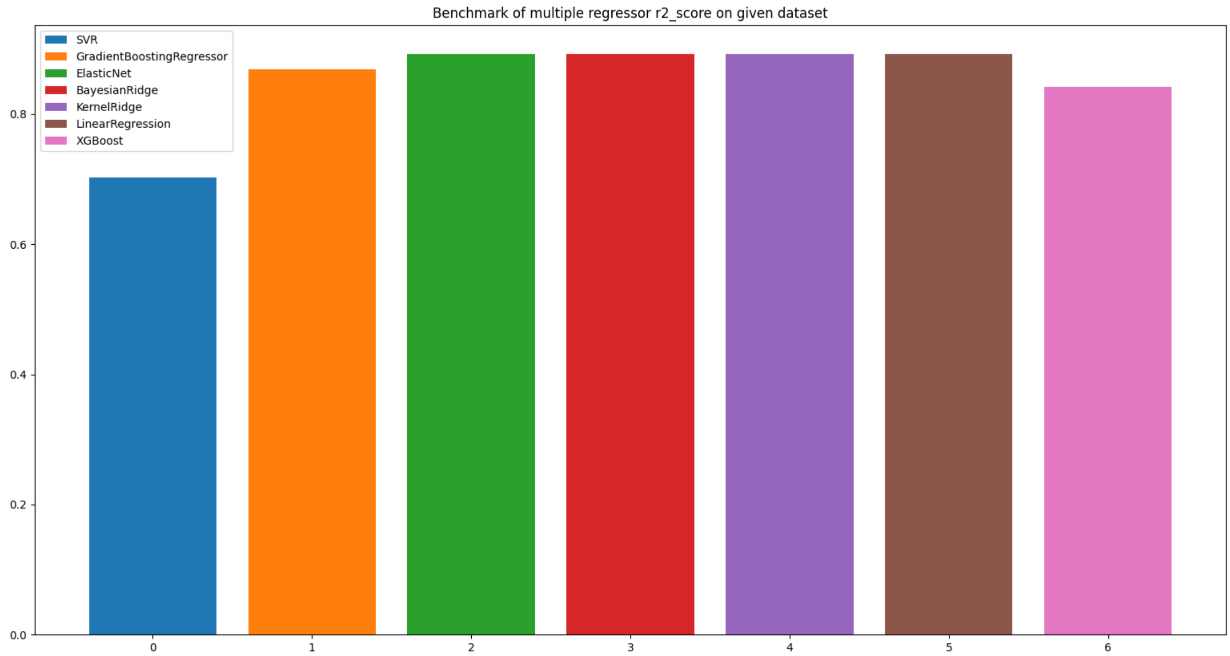

FIGURE 4.1 – Caption

5 Classification

For this exercice, we had to perform classification on a given dataset, with our inputs stored in a *inputs.npy* file and our label in a *labels.npy* file as in the last part.

We were again free to choose our classifier model and free to implement what we wanted, so we decided to test several algorithms exposed by the Sklearn Api and compare them.

The seven tested algorithms are :

- Support Vector Classifier
- Linear Support Vector Classifier
- Logistic Regression
- Random Forest
- Perceptron
- Stochastic Gradient Descent
- Decision Tree

We decided to split our dataset to have a test dataset and a train dataset, to run these algorithms with default hyper-parameters and to evaluate them with their accuracy. The objective was to obtain an accuracy superior than 0.85 on the test subset. With our basic implementation, we had great results that were superior than 0.85 on several of our tested classifier.

Here is the benchmark results :

Regression Algorithms tested	R2 score obtained
Support Vector Classifier	0.883
Linear Support Vector Classifier	0.889
Logistic Regression	0.891
Random Forest	0.858
Perceptron	0.855
Stochastic Gradient Descent	0.879
Decision Tree	0.807

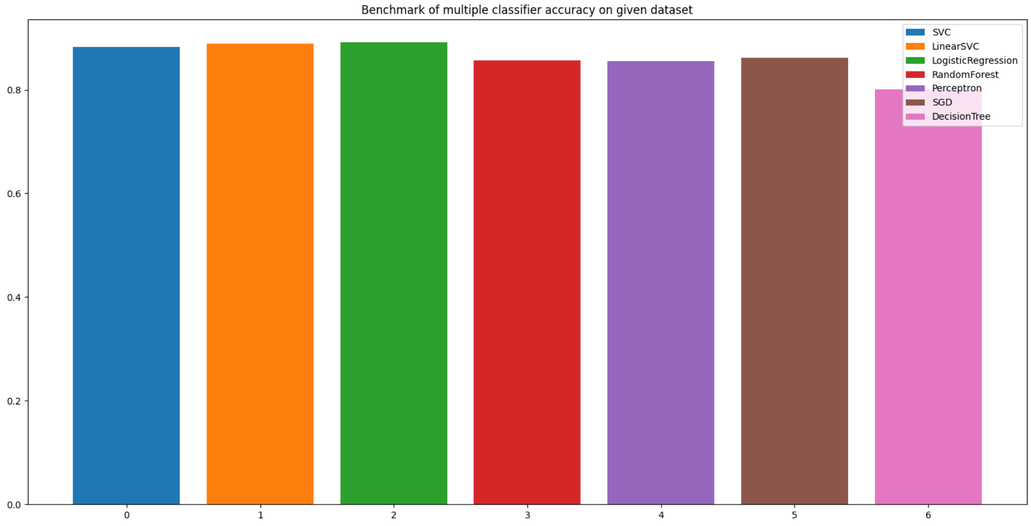

FIGURE 5.1 – Benchmark of multiple classifer accuracy trained on a given dataset