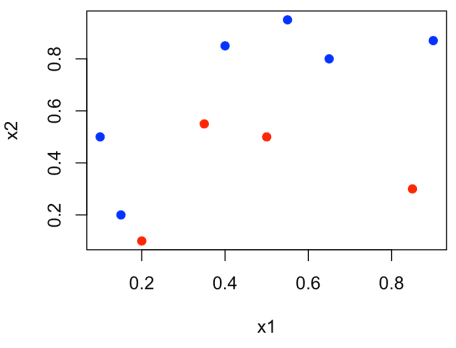
Last Friday, we discussed the use of ROC curves to describe the goodness of a classifier. I did say that I will post a brief paragraph on the interpretation of the diagonal. If you look around some say that it describes the “*strategy of randomly guessing a class*“, that it is obtained with “*a diagnostic test that is no better than chance level*“, even obtained by “*making a prediction by tossing of an unbiased coin*“.

Let us get back to ROC curves to illustrate those points. Consider a very simple dataset with 10 observations (that is not linearly separable)

|  |  |
| --- | --- |
| 1  2  3  4 | x1 = **c**(.4,.55,.65,.9,.1,.35,.5,.15,.2,.85)  x2 = **c**(.85,.95,.8,.87,.5,.55,.5,.2,.1,.3)  y = **c**(1,1,1,1,1,0,0,1,0,0)  **df** = **data.frame**(x1=x1,x2=x2,y=**as.factor**(y)) |

here we can check that, indeed, it is not separable

|  |  |
| --- | --- |
| 1 | **plot**(x1,x2,**col**=**c**("red","blue")[1+y],pch=19) |



Consider a logistic regression (the course is on linear models)

|  |  |
| --- | --- |
| 1 | reg = **glm**(y~x1+x2,**data**=**df**,**family**=**binomial**(link = "logit")) |

but any model here can be used… We can use our own function

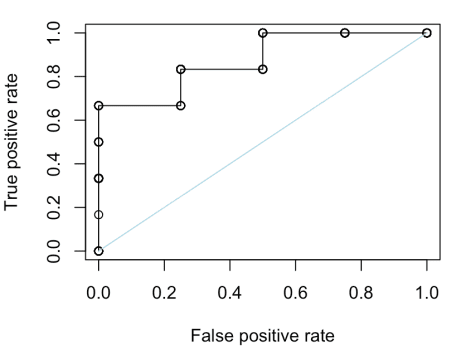
|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10 | Y=**df**$y S=**predict**(reg) roc.curve=**function**(s,**print**=FALSE){  Ps=(S&gt;=s)\*1  FP=**sum**((Ps==1)\*(Y==0))/**sum**(Y==0)  TP=**sum**((Ps==1)\*(Y==1))/**sum**(Y==1)  **if**(**print**==TRUE){  **print**(**table**(Observed=Y,Predicted=Ps))  }  vect=**c**(FP,TP)  **names**(vect)=**c**("FPR","TPR")  **return**(vect) } |

or any R package actually

|  |  |
| --- | --- |
| 1  2 | **library**(ROCR)  perf=performance(prediction(S,Y),"tpr","fpr") |

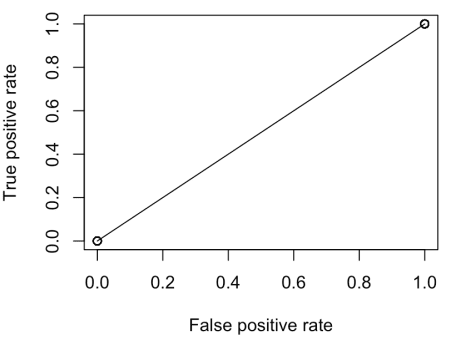
We can plot the two simultaneously here

|  |  |
| --- | --- |
| 1  2  3  4 | **plot**(performance(prediction(S,Y),"tpr","fpr"))  V=**Vectorize**(roc.curve)(**seq**(-5,5,**length**=251))  **points**(V[1,],V[2,])  **segments**(0,0,1,1,**col**="light blue") |

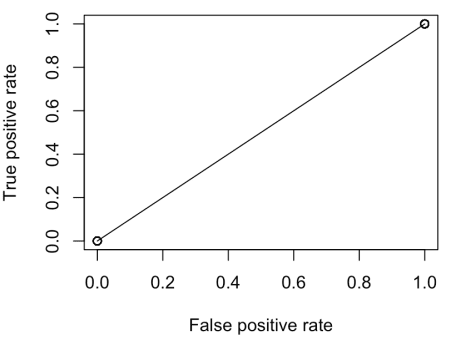
So our code works just fine, here. Let us consider various strategies that should lead us to the diagonal.

The first one is : everyone has the same probability (say 50%)

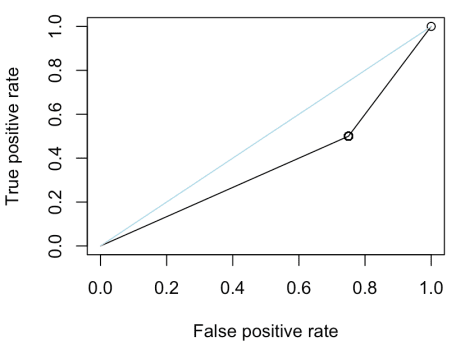
|  |  |
| --- | --- |
| 1  2  3  4 | S=**rep**(.5,10)  **plot**(performance(prediction(S,Y),"tpr","fpr"))  V=**Vectorize**(roc.curve)(**seq**(0,1,**length**=251))  **points**(V[1,],V[2,]) |

Indeed, we have the diagonal. But to be honest, we have only two points here : and . Claiming that we have a straight line is not very satisfying… Actually, note that we have this situation whatever the probability we choose

|  |  |
| --- | --- |
| 1  2  3  4 | S=**rep**(.2,10)  **plot**(performance(prediction(S,Y),"tpr","fpr"))   V=**Vectorize**(roc.curve)(**seq**(0,1,**length**=251))  **points**(V[1,],V[2,]) |

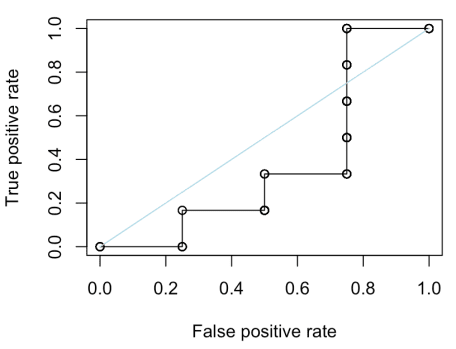
We can try another strategy, like “*making a prediction by tossing of an unbiased coin*“. This is what we obtain

|  |  |
| --- | --- |
| 1  2  3  4  5  6 | **set.seed**(1)  S=**sample**(0:1,size=10,**replace**=TRUE)  **plot**(performance(prediction(S,Y),"tpr","fpr"))  V=**Vectorize**(roc.curve)(**seq**(0,1,**length**=251))  **points**(V[1,],V[2,])  **segments**(0,0,1,1,**col**="light blue") |



We can also try some sort of “random classifier”, where we choose the score randomly, say uniform on the unit interval

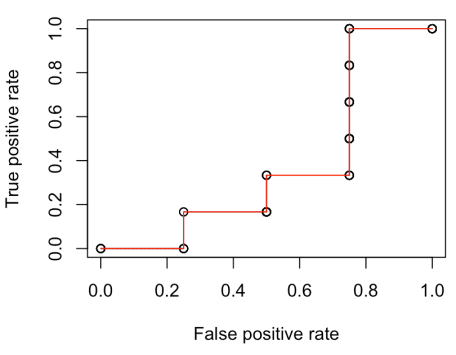
|  |  |
| --- | --- |
| 1  2  3  4  5  6 | **set.seed**(1)   S=**runif**(10)  **plot**(performance(prediction(S,Y),"tpr","fpr"))  V=**Vectorize**(roc.curve)(**seq**(0,1,**length**=251))  **points**(V[1,],V[2,])  **segments**(0,0,1,1,**col**="light blue") |

Let us try to go further on that one. For convenience, let us consider another function to plot the ROC curve

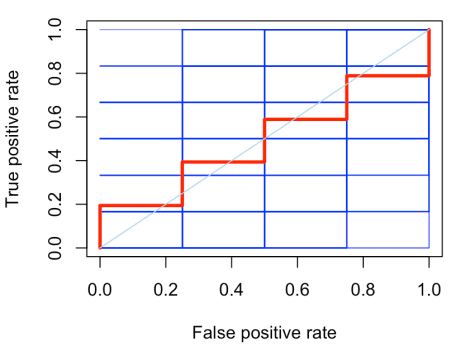
|  |  |
| --- | --- |
| 1  2 | V=**Vectorize**(roc.curve)(**seq**(0,1,**length**=251))   roc\_curve=**Vectorize**(**function**(x) **max**(V[2,**which**(V[1,]&lt;=x)])) |

We have the same line as previously

|  |  |
| --- | --- |
| 1  2  3 | x=**seq**(0,1,**by**=.025)  y=roc\_curve(x)  **lines**(x,y,type="s",**col**="red") |

But now, consider many scoring strategies, all randomly chosen

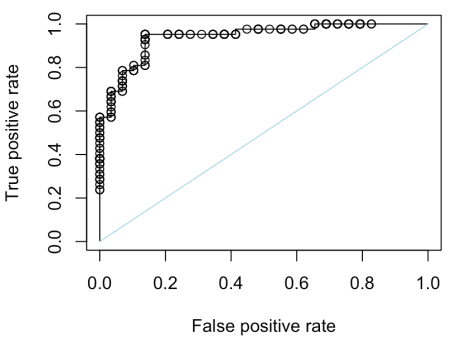
|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9 | MY=**matrix**(NA,500,**length**(y))  **for**(i **in** 1:500){  S=**runif**(10)  V=**Vectorize**(roc.curve)(**seq**(0,1,**length**=251))  MY[i,]=roc\_curve(x)  }  **plot**(performance(prediction(S,**df**$y),"tpr","fpr"),**col**="white")  **for**(i **in** 1:500){  **lines**(x,MY[i,],**col**=**rgb**(0,0,1,.3),type="s") }  **lines**(**c**(0,x),**c**(0,**apply**(MY,2,**mean**)),**col**="red",type="s",lwd=3) **segments**(0,0,1,1,**col**="light blue") |



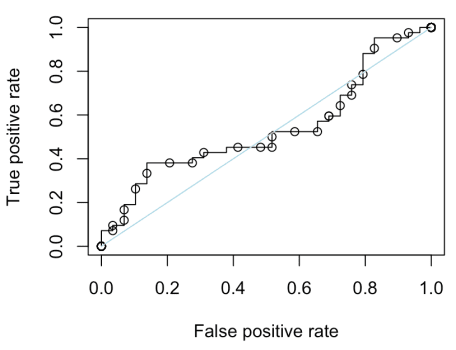
The red line is the average of all random classifiers. It is not a straight line, be we observe oscillations around the diagonal.

Consider a dataset with more observations

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9 | myocarde = **read.table**("http://freakonometrics.free.fr/myocarde.csv",**head**=TRUE, sep=";")  myocarde$PRONO = (myocarde$PRONO=="SURVIE")\*1  reg = **glm**(PRONO~.,**data**=myocarde,**family**=**binomial**(link = "logit"))  Y=myocarde$PRONO  S=**predict**(reg)  **plot**(performance(prediction(S,Y),"tpr","fpr"))   V=**Vectorize**(roc.curve)(**seq**(-5,5,**length**=251))  **points**(V[1,],V[2,])  **segments**(0,0,1,1,**col**="light blue") |

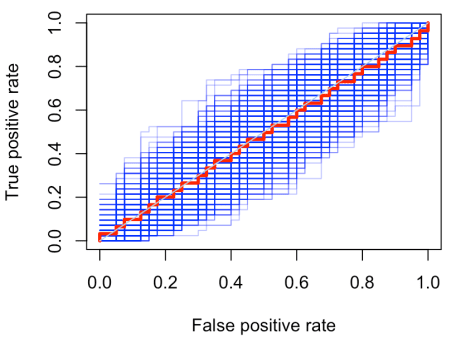
Here is a “random classifier” where we draw scores randomly on the unit interval

|  |  |
| --- | --- |
| 1  2  3  4  5 | S=**runif**(**nrow**(myocarde)  **plot**(performance(prediction(S,Y),"tpr","fpr"))  V=**Vectorize**(roc.curve)(**seq**(-5,5,**length**=251))  **points**(V[1,],V[2,])  **segments**(0,0,1,1,**col**="light blue") |



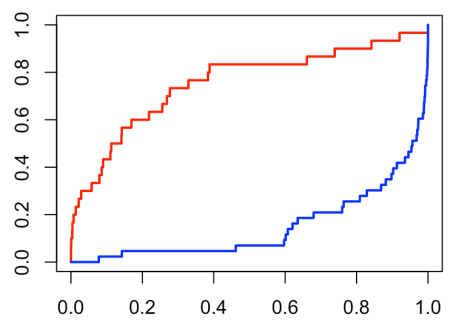
And if we do that 500 times, we obtain, on average

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9 | MY=**matrix**(NA,500,**length**(y))  **for**(i **in** 1:500){  S=**runif**(**length**(Y))  V=**Vectorize**(roc.curve)(**seq**(0,1,**length**=251))  MY[i,]=roc\_curve(x) }  **plot**(performance(prediction(S,Y),"tpr","fpr"),**col**="white")  **for**(i **in** 1:500){  **lines**(x,MY[i,],**col**=**rgb**(0,0,1,.3),type="s")  }  **lines**(**c**(0,x),**c**(0,**apply**(MY,2,**mean**)),**col**="red",type="s",lwd=3) **segments**(0,0,1,1,**col**="light blue") |

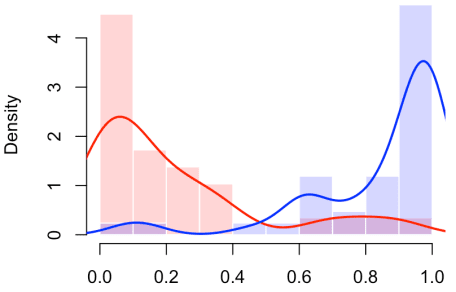
So, it looks like me might say that the diagonal is what we have, on average, when drawing randomly scores on the unit interval…

I did mention that an interesting visual tool could be related to the use of the Kolmogorov Smirnov statistic on classifiers. We can plot the two empirical cumulative distribution functions of the scores, given the response

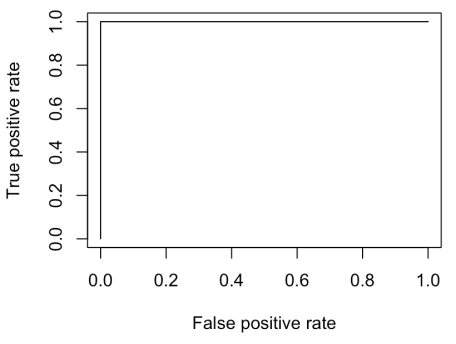
|  |  |
| --- | --- |
| 1  2  3  4  5 | score=**data.frame**(yobs=Y,  ypred=**predict**(reg,type="response"))   f0=**c**(0,**sort**(score$ypred[score$yobs==0]),1)  f1=**c**(0,**sort**(score$ypred[score$yobs==1]),1)  **plot**(f0,(0:(**length**(f0)-1))/(**length**(f0)-1),**col**="red",type="s",lwd=2,xlim=0:1)  **lines**(f1,(0:(**length**(f1)-1))/(**length**(f1)-1),**col**="blue",type="s",lwd=2) |

we can also look at the distribution of the score, with the histogram (or density estimates)

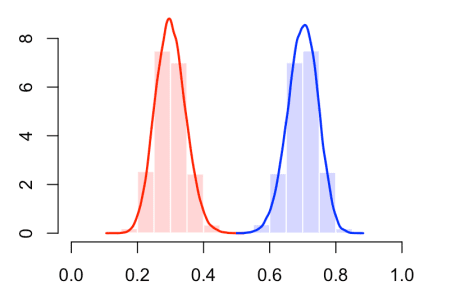
|  |  |
| --- | --- |
| 1  2  3  4  5 | S=score$ypred  **hist**(S[Y==0],**col**=**rgb**(1,0,0,.2),  probability=TRUE,breaks=(0:10)/10,border="white")  **hist**(S[Y==1],**col**=**rgb**(0,0,1,.2),  probability=TRUE,breaks=(0:10)/10,border="white",add=TRUE)  **lines**(**density**(S[Y==0]),**col**="red",lwd=2,xlim=**c**(0,1))  **lines**(**density**(S[Y==1]),**col**="blue",lwd=2) |



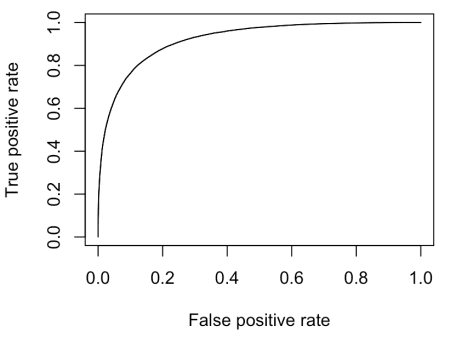
The underlying idea is the following : we do have a “perfect classifier” (top left corner)



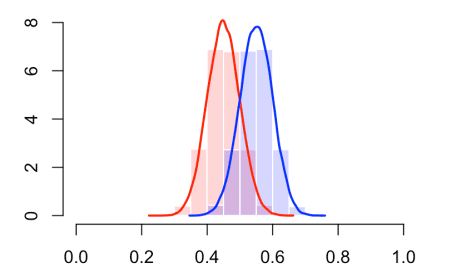
is the supports of the scores do not overlap



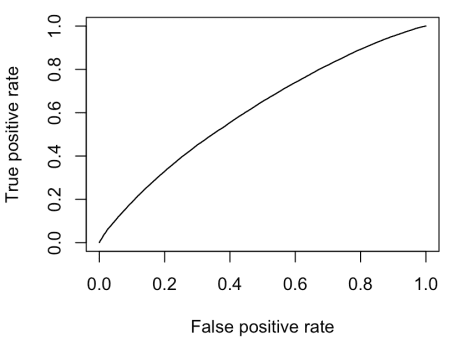
otherwise, we should have errors. That the case below

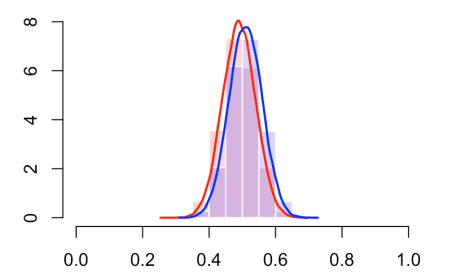


we in 10% of the cases, we might have misclassification



or even more missclassification, with overlapping supports

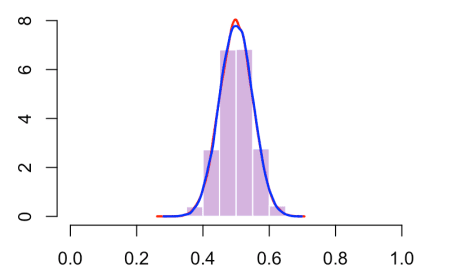




Now, we have the diagonal



when the two conditional distributions of the scores are identical



Of course, that only valid when is very large, otherwise, it is only what we observe on average….