La Muerte del Overfitting

Gustavo Denicolay

2019-11

I am **tormented**
with an everlasting itch
for things remote.

I love to sail
forbidden seas.

– Herman Melville, Moby-Dick.

Afirmación I

Algunos algoritmos fallaban para ciertos datasets, el establishment declaró culpable al dataset y se le diagnosticó la patología “dataset con clase desbalanceada”, el dataset debia ser *corregido*. Pensamientos confusos y fraudes piadosos.

La soberbia evitó ver que el verdadero problema estaba en esos algoritmos, que no estaban haciendo el test estadístico correcto.

Afirmación II

La mayoría de los modelos fallaban en datos nuevos, el establishment declaró culpable a “la complejidad del modelo” y diagnosticó la patología “overfitting” algo místico relacionado con “ajustar el modelo al ruido”.

La soberbia evitó ver que el verdadero problema estaba en esos algoritmos, que no estaban haciendo el test estadístico correcto para las múltiples comparaciones que efectuaban en su construcción.

Afirmación III

Es una práctica ABERRANTE y de resultados sub optimos en pos de controlar el overfitting, limitar la “complejidad” del arbol por :

- max_depth
- min_split
- min_bucket

cuando simplemente alcanza con hacer el test estadístico adecuado para las multiples comparaciones.

Afirmación IV

El modelo va a generalizar bien en datos nuevos si durante su construcción antes de agregar un nuevo elemento al modelo respondo esta pregunta :

Es esto realmente un patrón o es un simple producto del azar (dadas todas las variables que tengo) ?

Bibliography

Jensen, D. D., & Cohen, P. R. *Multiple comparisons in induction algorithms.* Machine Learning, 38(3): 309-338, 2000

Ojala, M. and Garriga, G.C. *Permutation tests for studying classifier performance*. Journal of Machine Learning Research, 11:1833–1863, 2010

Errors in adding components to a model,
usually called overfitting,
are probably the best known pathology
of induction algorithms.

Overfitting occurs when a
Multiple Comparison Procedure
is applied to model components

Errors in adding components to a model,
usually called overfitting,
are probably the best known pathology
of induction algorithms.

Overfitting occurs when a
Multiple Comparison Procedure
is applied to model components

Induction algorithms vary widely in how they generate and evaluate components, but all algorithms that decide whether to add c_{max} to a model make implicit or explicit statistical hypothesis tests.[3] One common form of the test asks: "Under the null hypothesis that a component c will not improve the predictive power of the model m, what is the probability of a score at least as large as x?" When this probability is very small, algorithms reject the null hypothesis and infer that adding c will improve the predictive power of m. This form of the test is usually *incorrectly* applied to the component c_{max} and its associated score x_{max}.

Under the null hypothesis that a component *c*
will not improve the predictive power of model *m*,
what is the probability of a score at least as large *x* ?

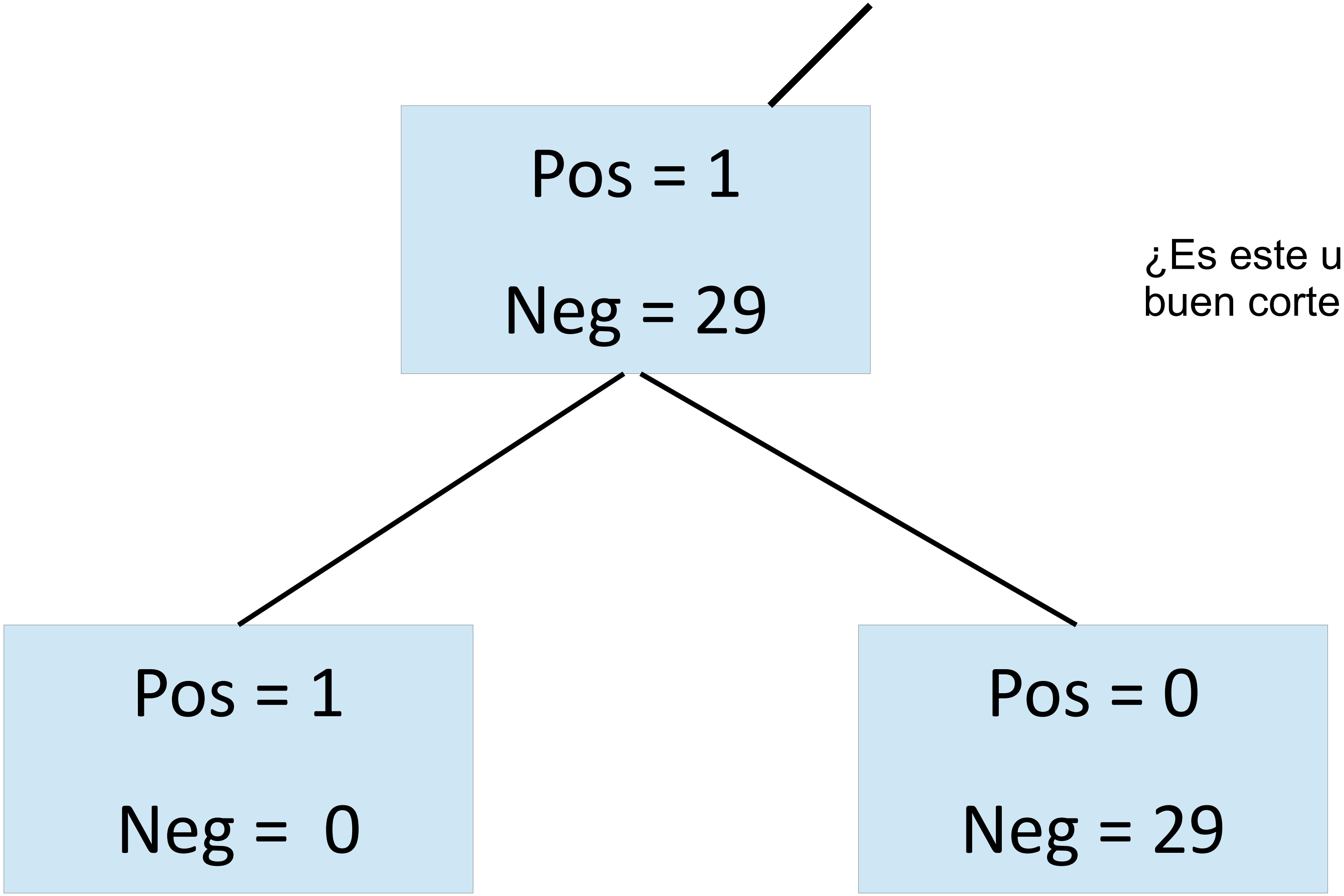
The test is incorrect because it does not adjust for n,
the number of components examined.

Overfitting occurs because
the wrong form of test
is used

Some decision tree algorithms are far more likely
to construct models that use discrete variables with
many values (e.g. home town)
rather than discrete variables with relatively
few values (e.g. gender)

A third pathology reveals for induction algorithms that efficiently search extremely large spaces of models. Paradoxically, these algorithms produce models that are often less accurate on new data that models produced by algorithms that search only a fraction of the same space.

¿Qué es el Procedimiento de Multiples Comparaciones ?



Pos = 1

Neg = 29

Pos = 1

Neg = 0

Pos = 0

Neg = 29

¿Es este un
buen corte ?

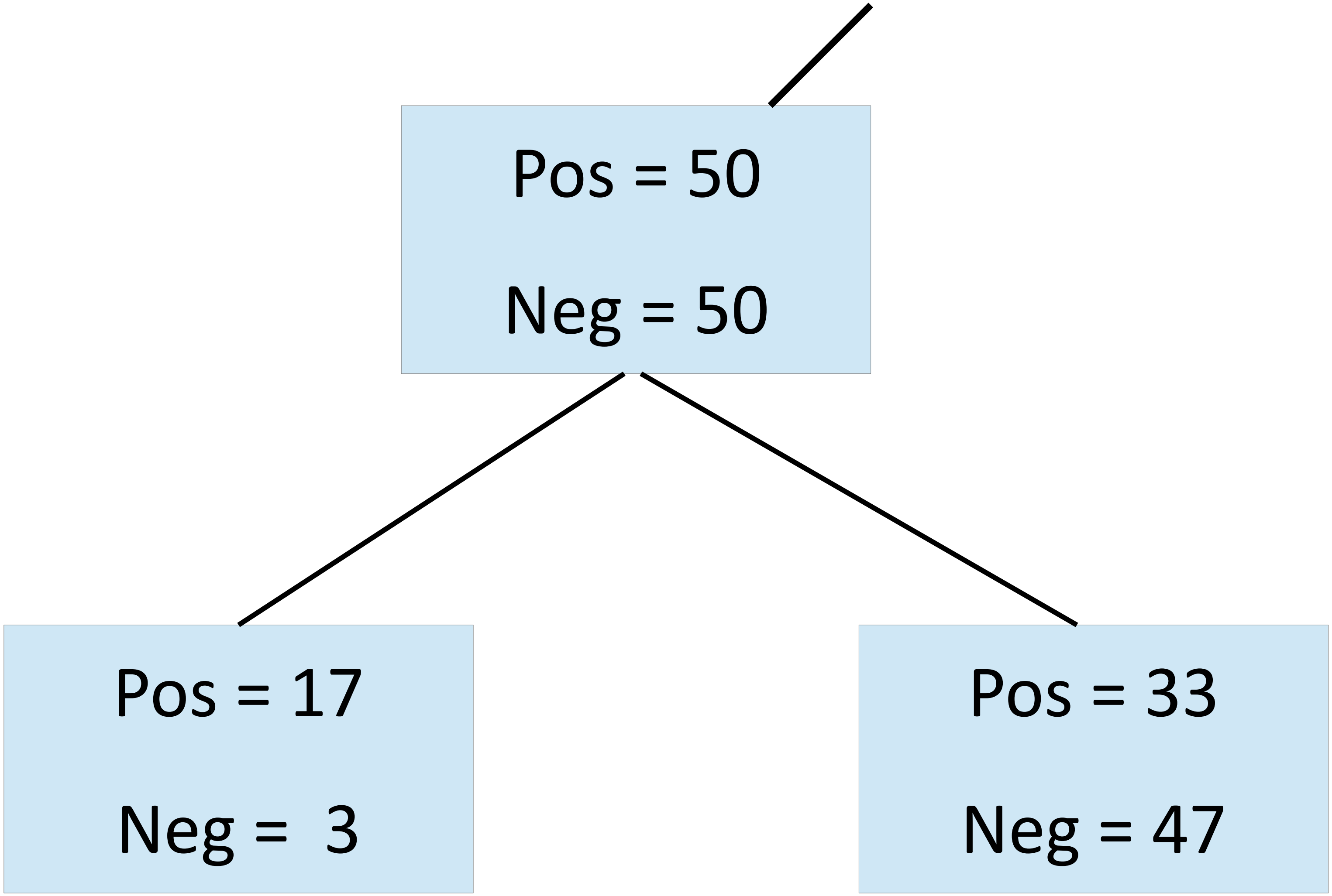
Si hubiera una sola variable
el p-value = 0.03333 < 0.05

Si tengo 2 atributos independientes entre si en el dataset la probabilidad de ese corte con alguna de las dos variables será :

0.065555555 > 0.05 NO es PATRON

Si tuviera 100 atributos independientes entre si en el dataset

p-value = 0.9662966 >> 0.05 NO es PATRON



Pos = 50

Neg = 50

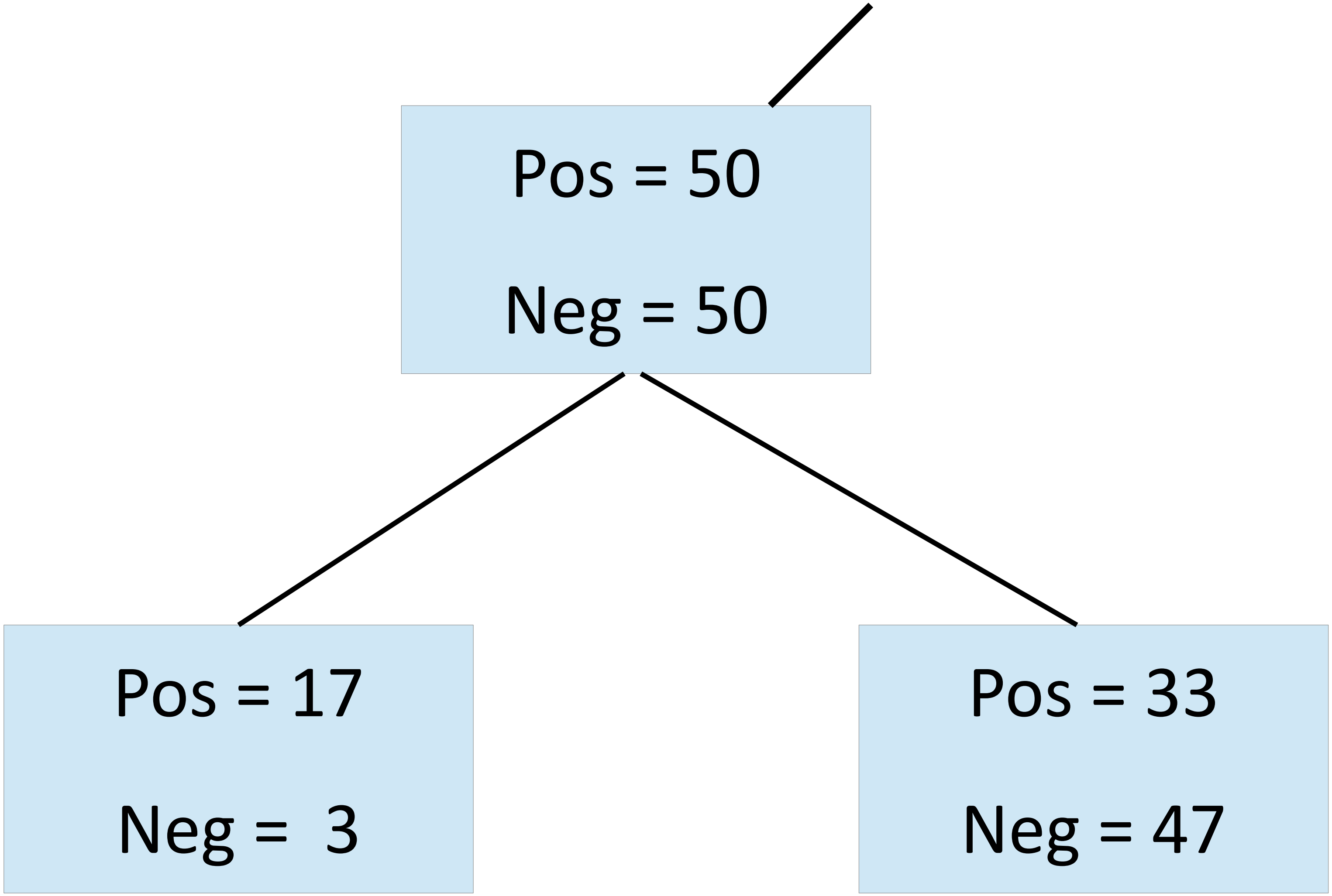
Pos = 17

Neg = 3

Pos = 33

Neg = 47

p-value = 0.0004 < 0.05



Pos = 50

Neg = 50

Pos = 17

Neg = 3

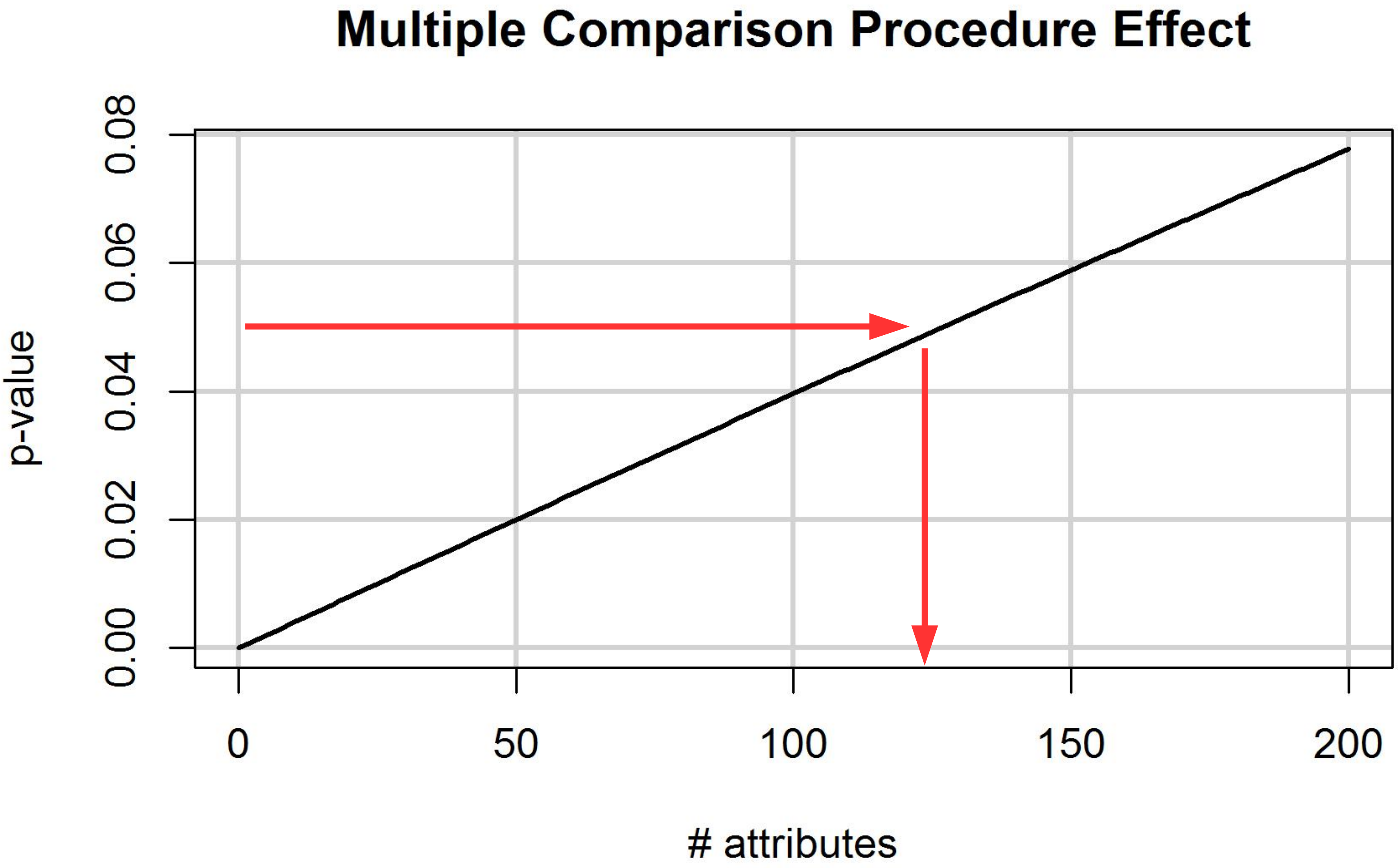
Pos = 33

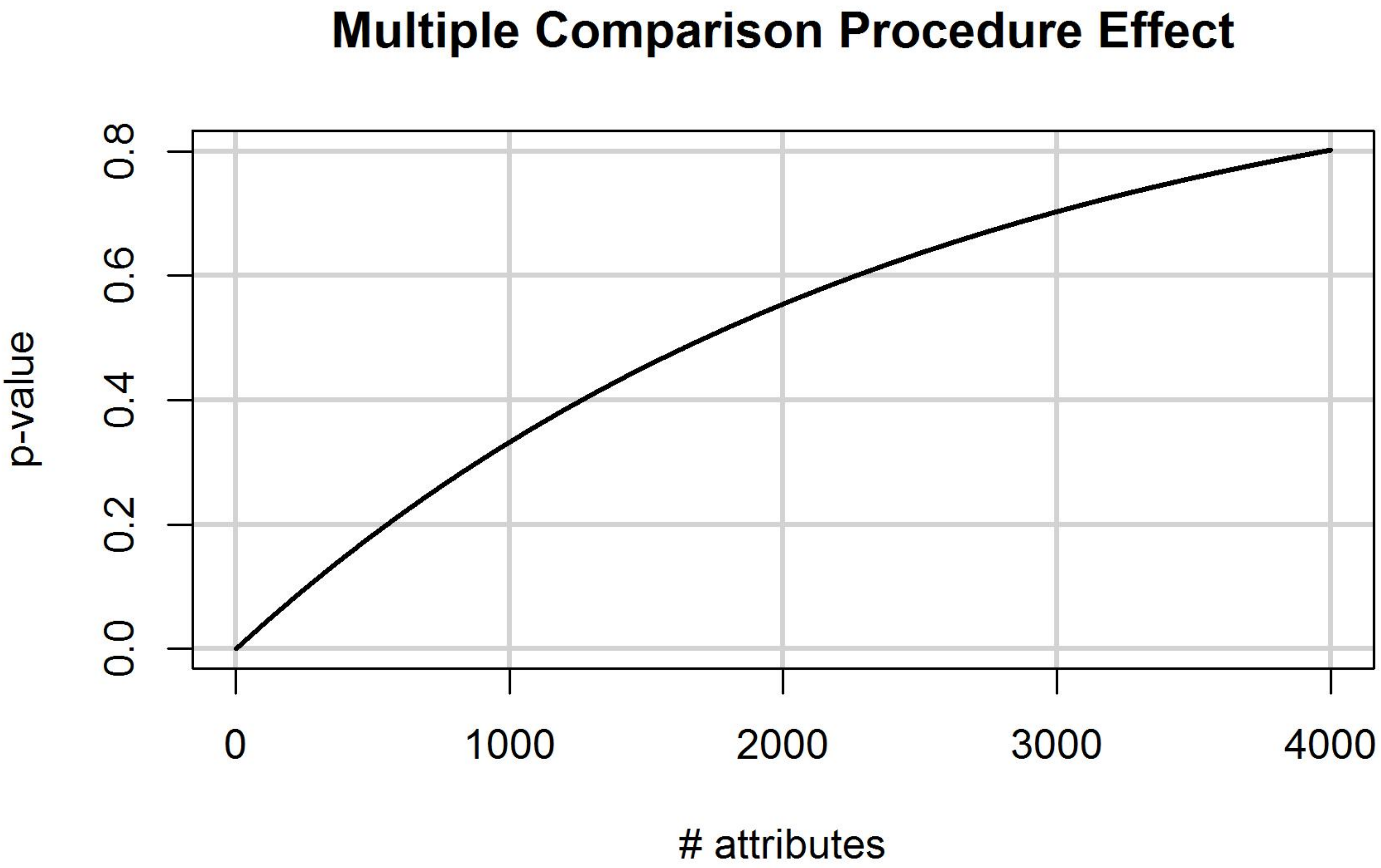
Neg = 47

p-value = 0.0004 < 0.05

Cual es el p-value que luego de
n variables independientes en si

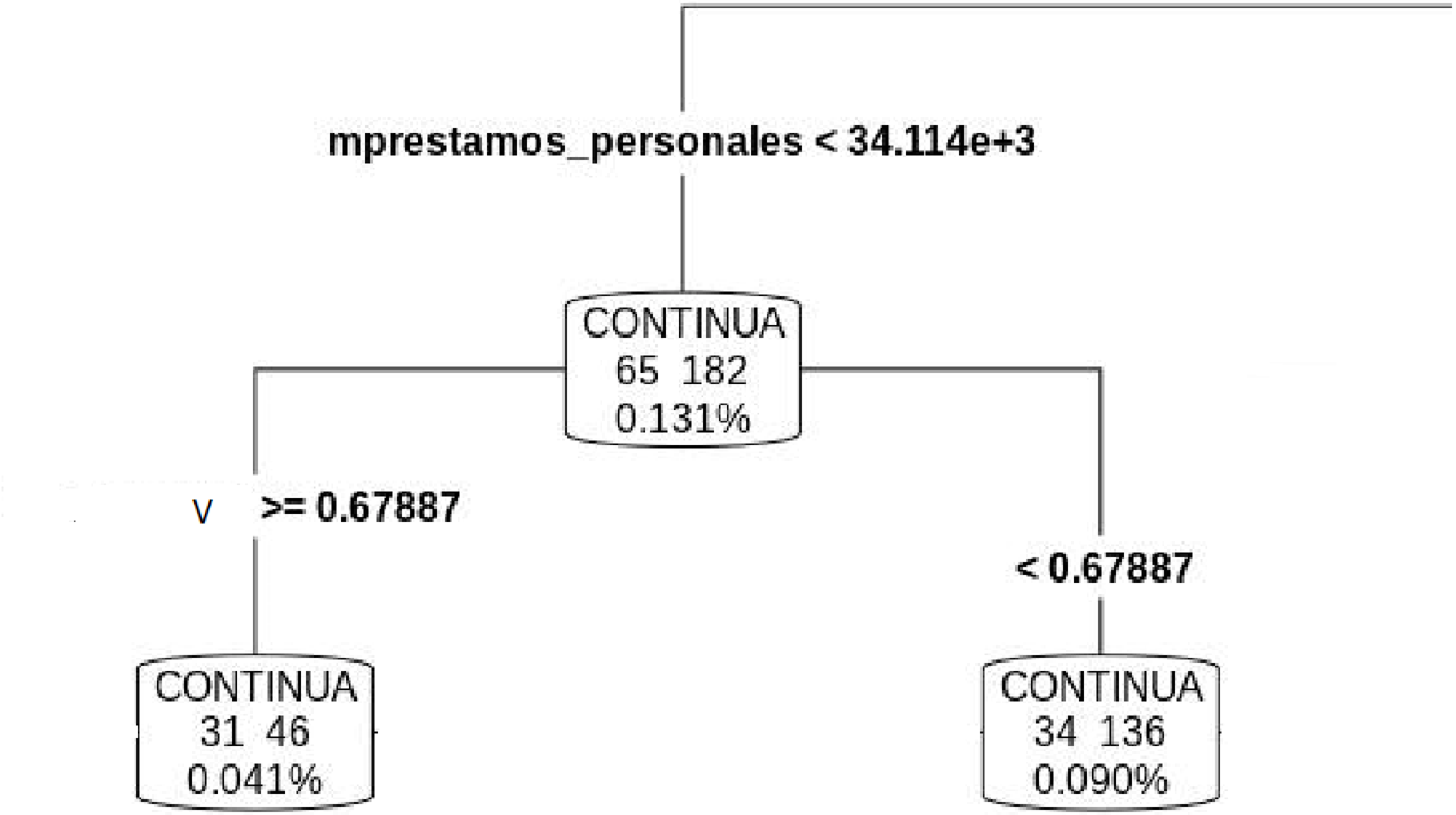
$$p\_value_n = 1 – (1 - p\_value_1)^n$$


Multiple Comparison Procedure Effect
p-value
0.00
0.02
0.04
0.06
0.08
0
50
100
150
200
# attributes


Multiple Comparison Procedure Effect
p-value
0.0
0.2
0.4
0.6
0.8
0
1000
2000
3000
4000
# attributes

¿Qué tal un ejemplo real ?

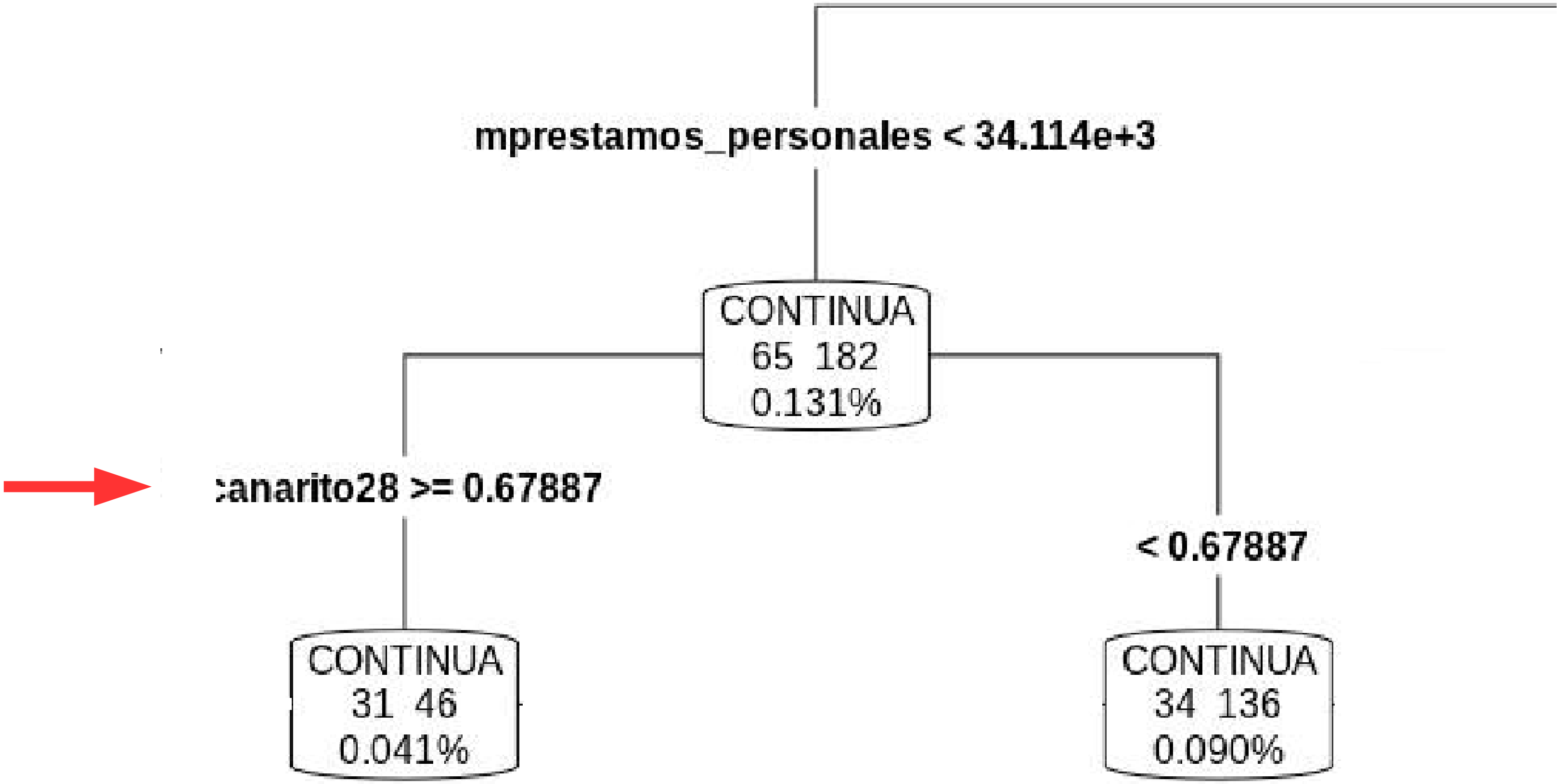
rpart hizo este corte ...



1-variable p-value = 0.00085 < 0.05 PATRON

150-variables p-value = 0.081 > 0.05 NO PATRON

El verdadero corte era por ...



1-variable p-value = 0.00085 < 0.05 PATRON

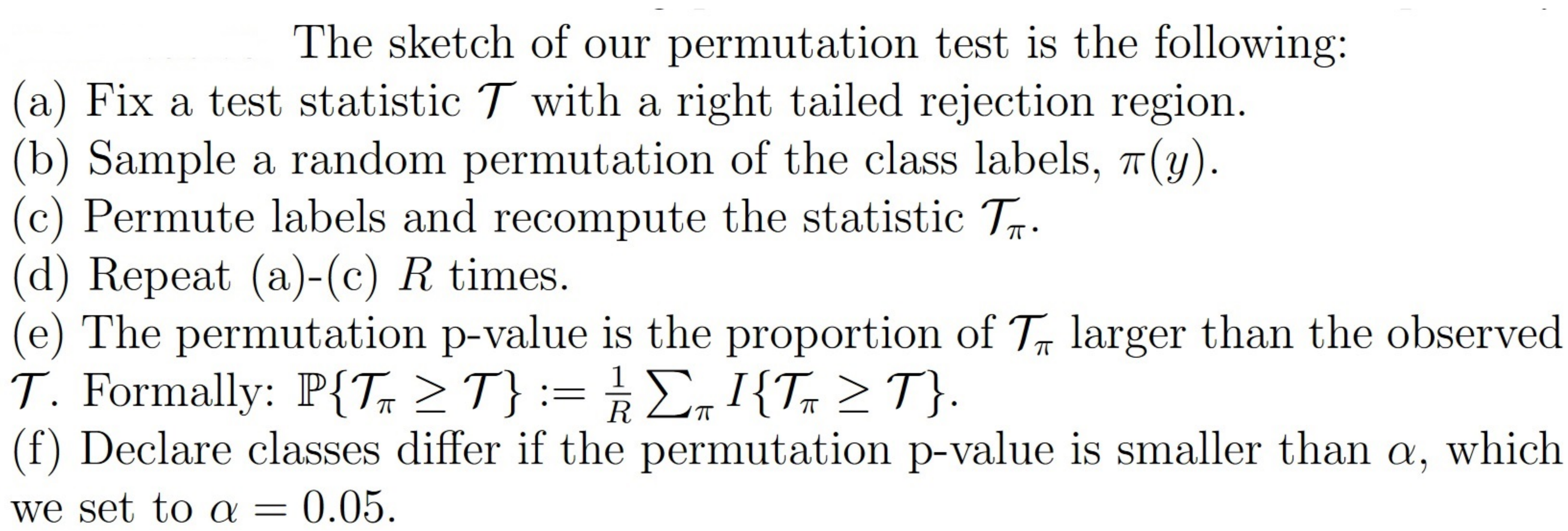
150-variables p-value = 0.081 > 0.05 NO PATRON

Task	**Possible Pathology**
decide whether to add a component to a model (whether to split or not a node in a decision tree, weather to add a new tree in gradient boosting)	Overfitting
which of several attributes to use in a model component (which attribute to split a node in a decision tree)	Selection error
select among different models	Oversearching

Solution for MCPs

- New Sample Data
- Cross-Validation
- Bonferroni Adjustment
- Class Randomization (Permutation)

Permutation Test



The sketch of our permutation test is the following:

(a) Fix a test statistic $\mathcal{T}$ with a right tailed rejection region.

(b) Sample a random permutation of the class labels, $\pi(y)$.

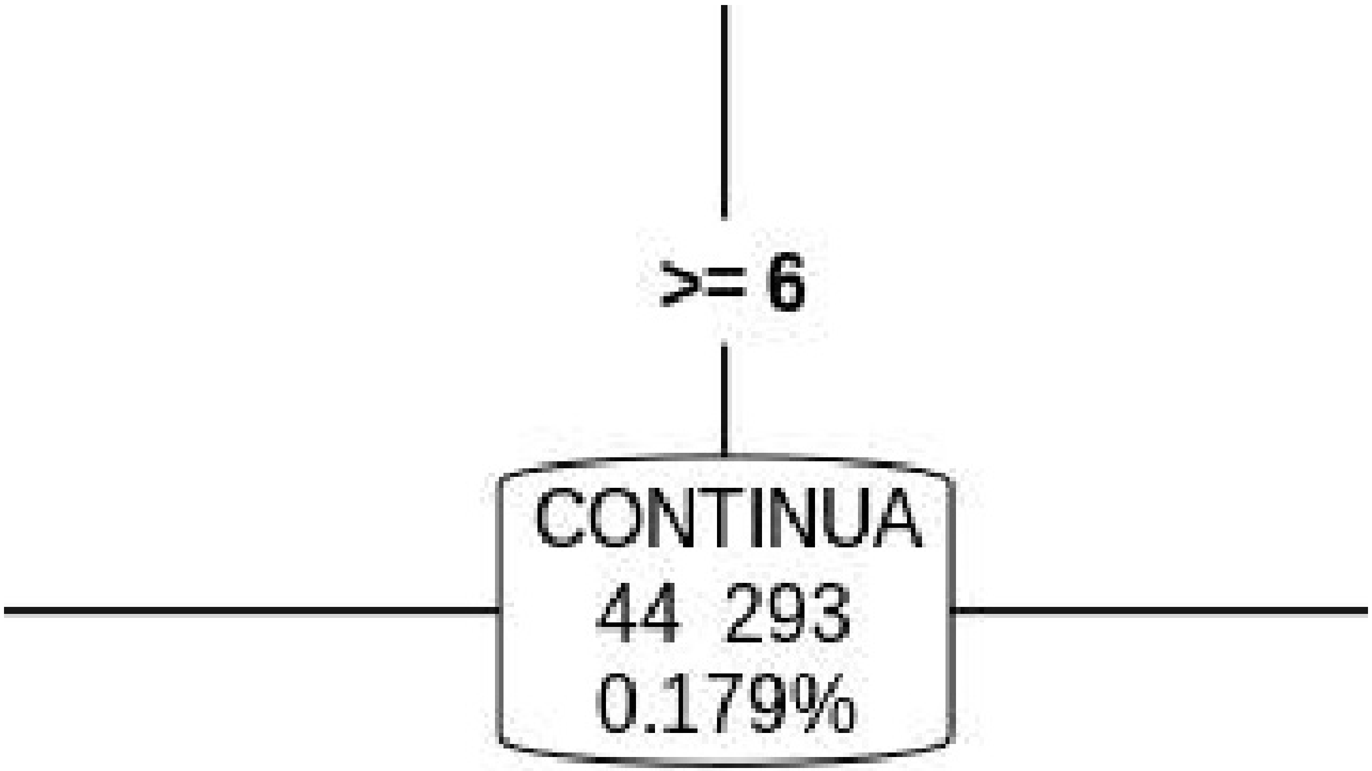
(c) Permute labels and recompute the statistic $\mathcal{T}_\pi$.

(d) Repeat (a)-(c) R times.

(e) The permutation p-value is the proportion of $\mathcal{T}_\pi$ larger than the observed $\mathcal{T}$. Formally: $\mathbb{P}\{\mathcal{T}_\pi \geq \mathcal{T}\} := \frac{1}{R}\sum_\pi I\{\mathcal{T}_\pi \geq \mathcal{T}\}$.

(f) Declare classes differ if the permutation p-value is smaller than α, which we set to $\alpha = 0.05$.

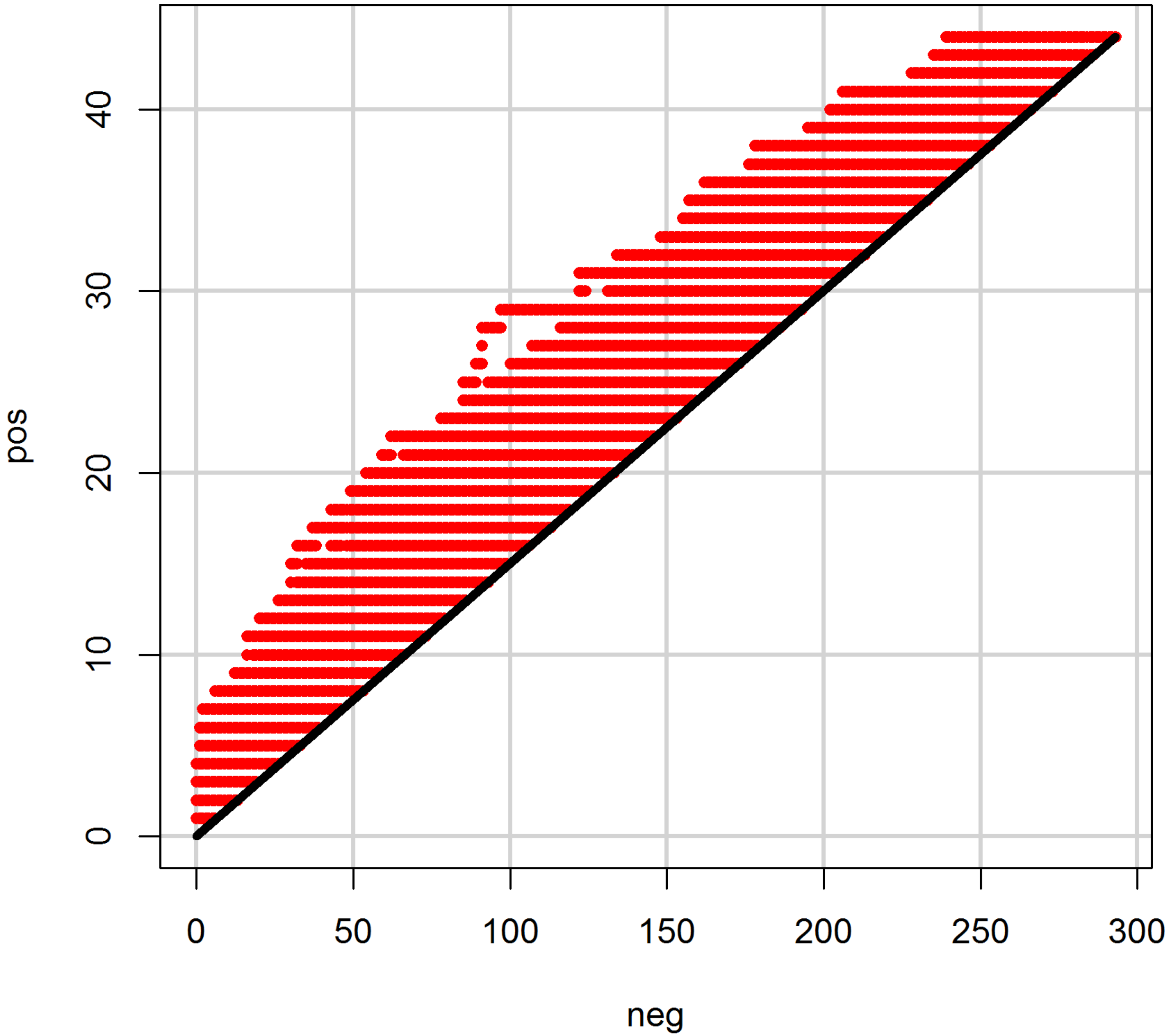
(explicación en pizarrón de
Class Randomization)



>= 6
CONTINUA
44 293
0.179%

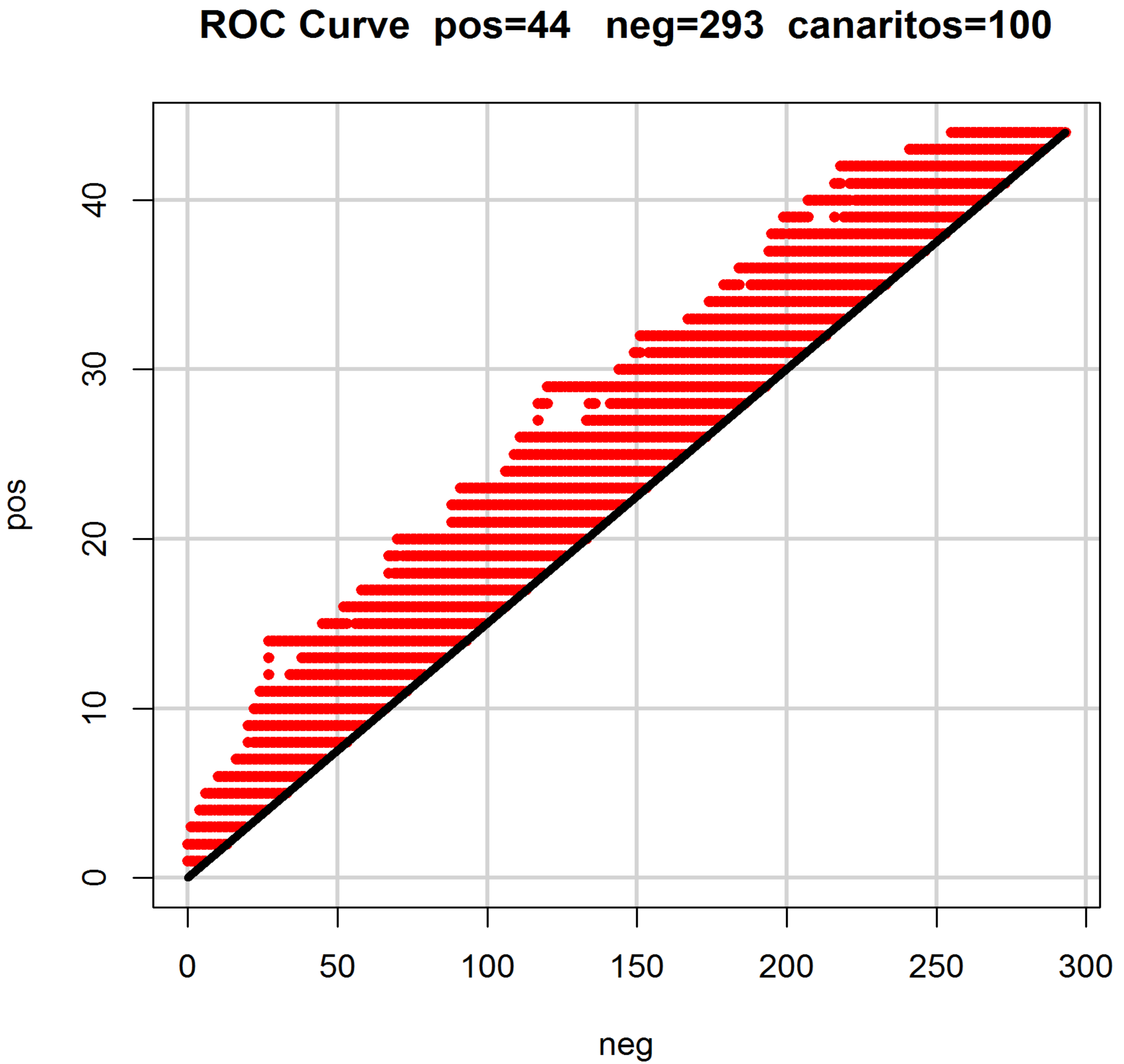
(pizzarrón + R real time)

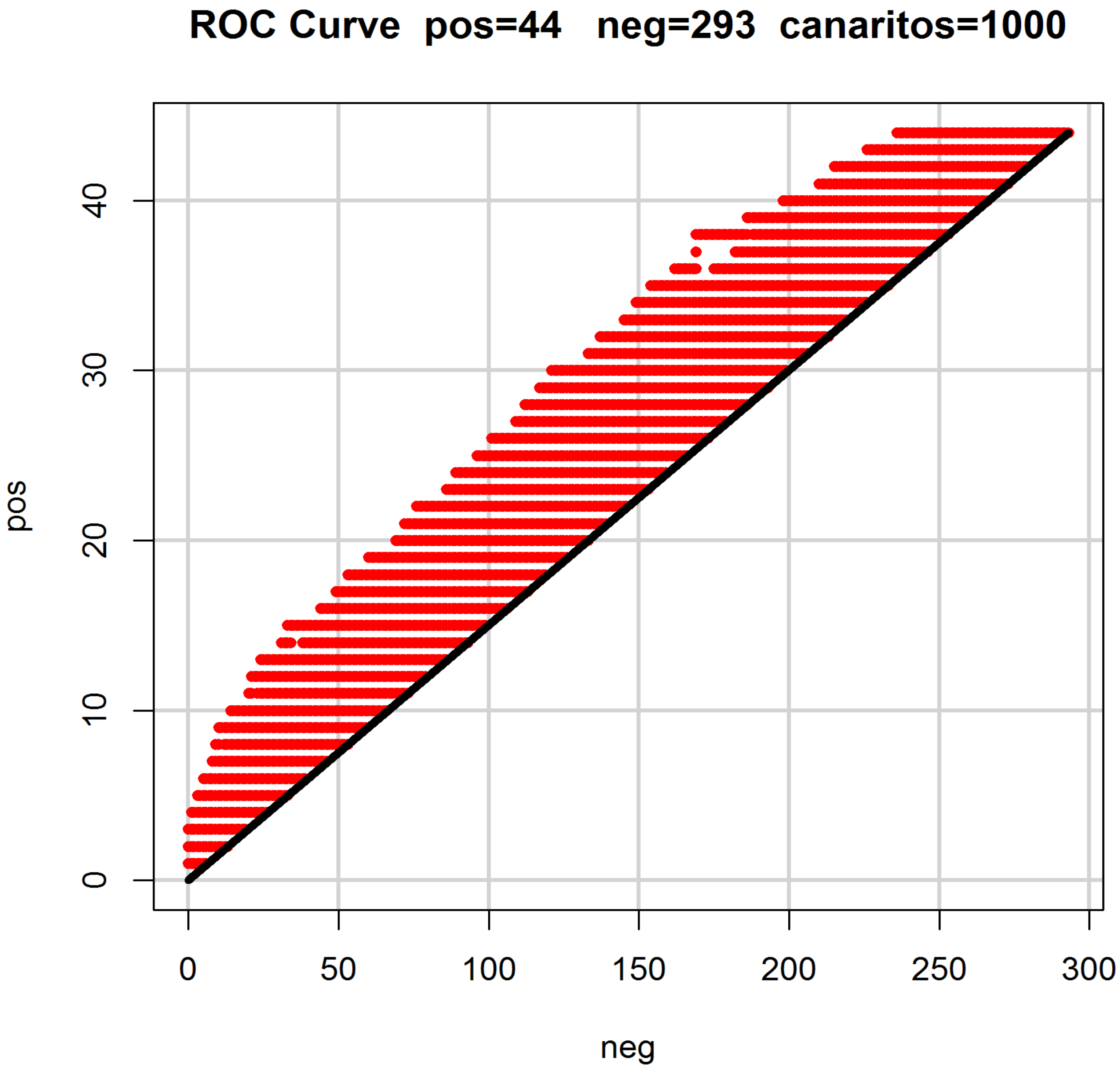
ROC Curve pos=44   neg=293 permutaciones=20



Los canaritos se han utilizado en las minas de carbon para detectar monóxido de carbono y otros gases tóxicos antes que afecten al ser humano.
Los canaritos actuan como una especie sentinela :
un animal más sensible a los inodoros e incoloros venenosos gases.

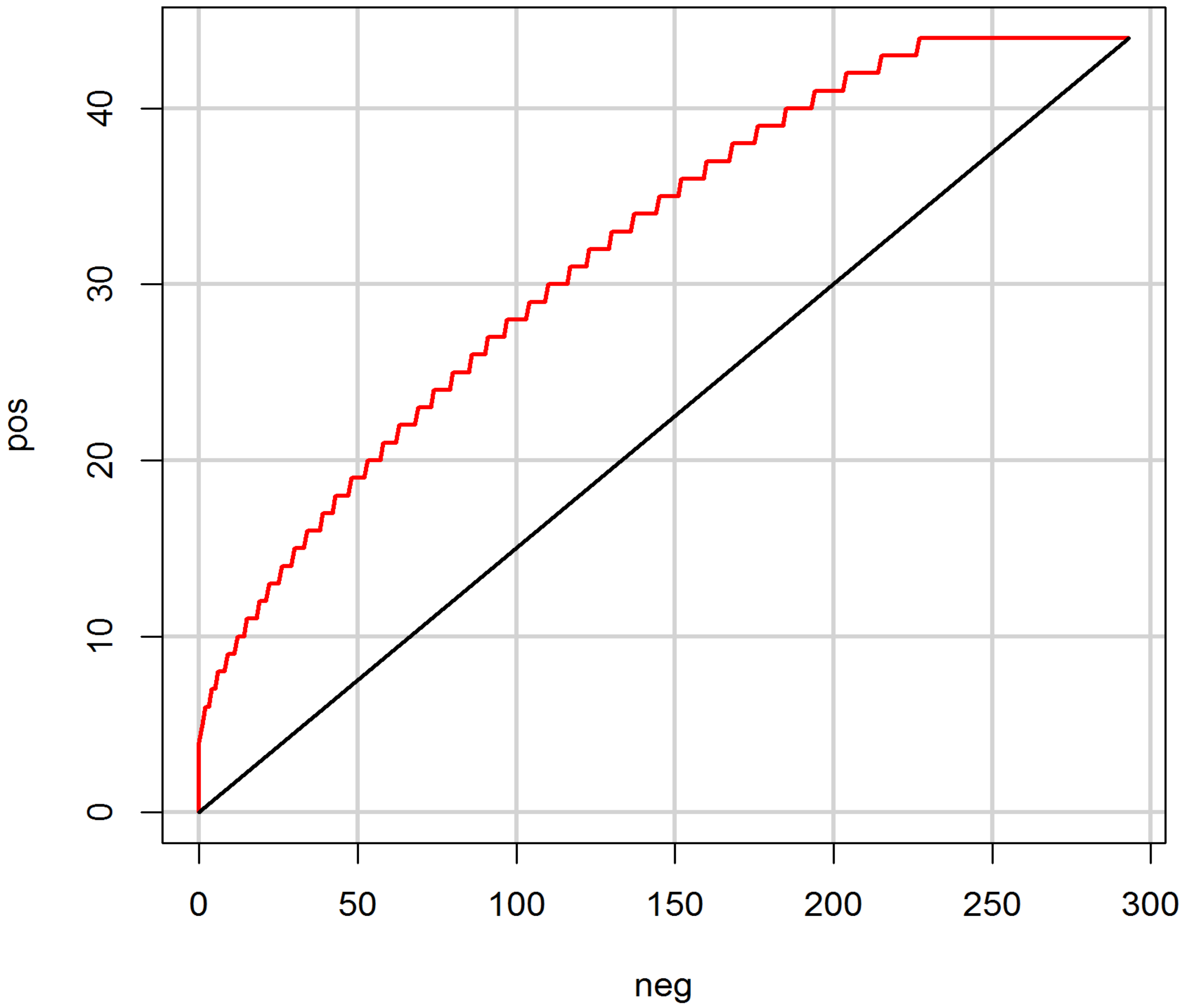
Si el canarito se desvanecía o moría, eso alertaba a los mineros a evacuar la mina inmediatamente.

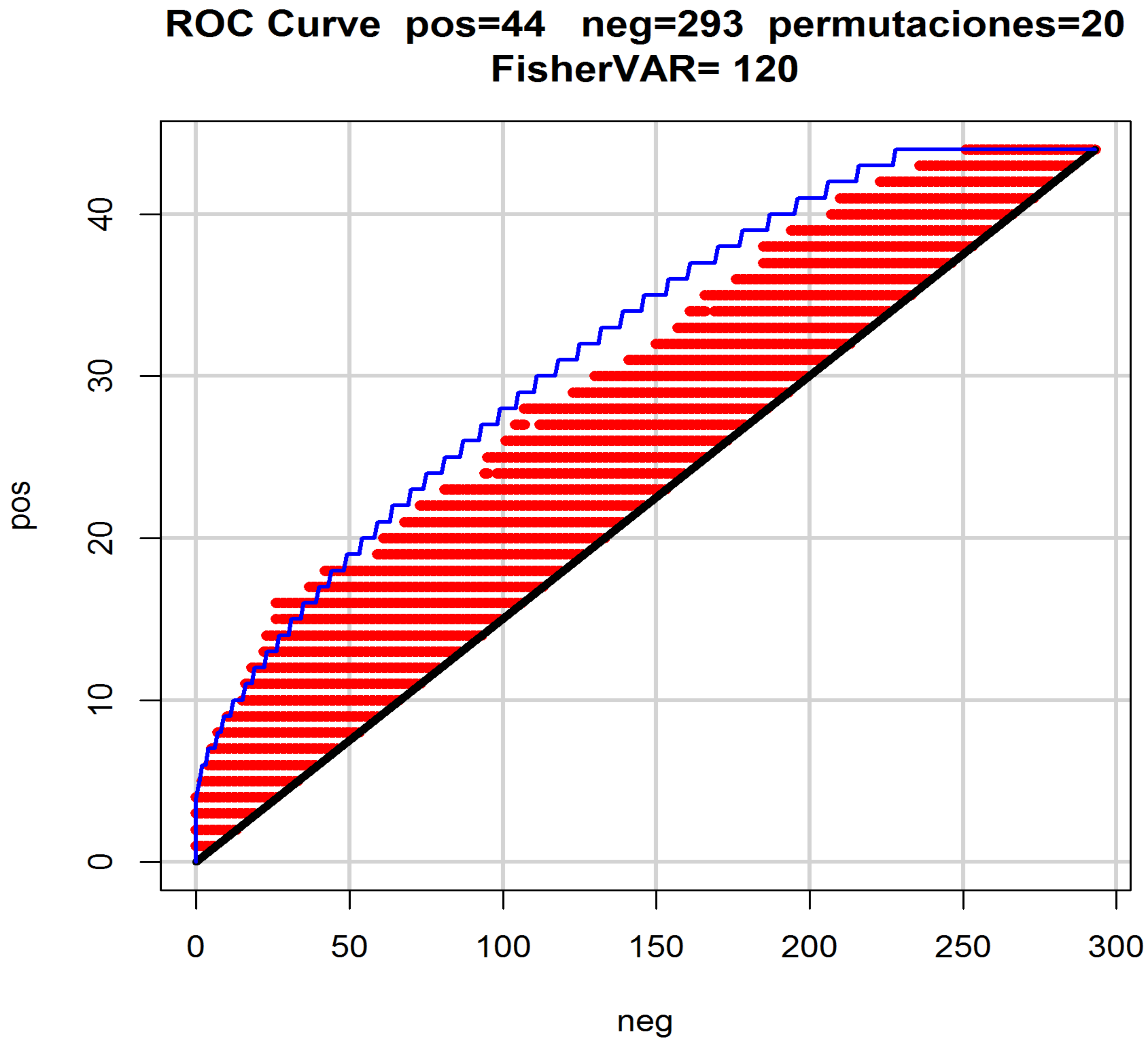

ROC Curve pos=44 neg=293 canaritos=100
pos
0
10
20
30
40
neg
0
50
100
150
200
250
300


ROC Curve pos=44   neg=293 canaritos=1000
pos
0
10
20
30
40
neg
0
50
100
150
200
250
300

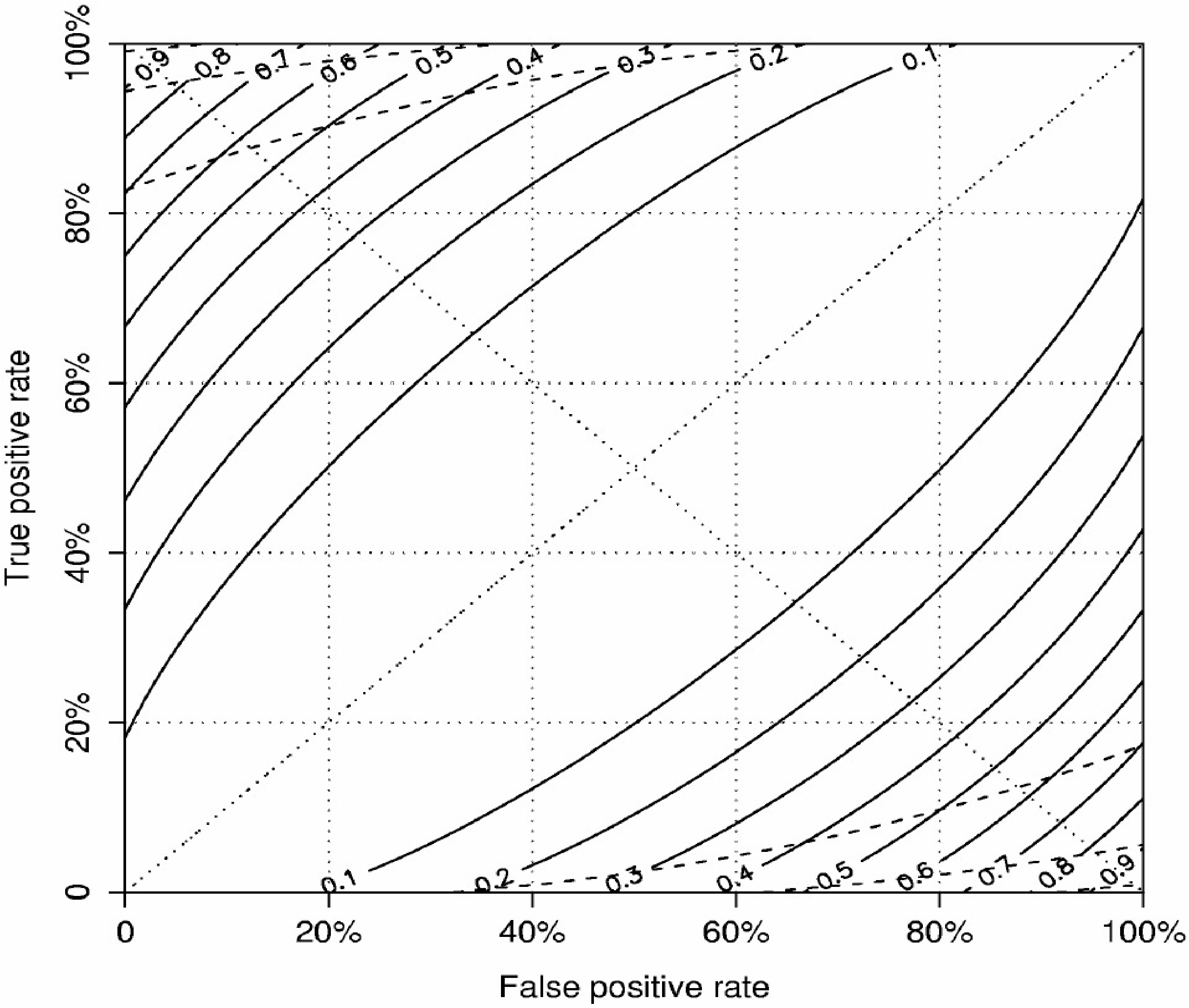
Ahora la curva teórica

ROC Curve teoricapos=44 neg=293 variables=150 alpha=0.0



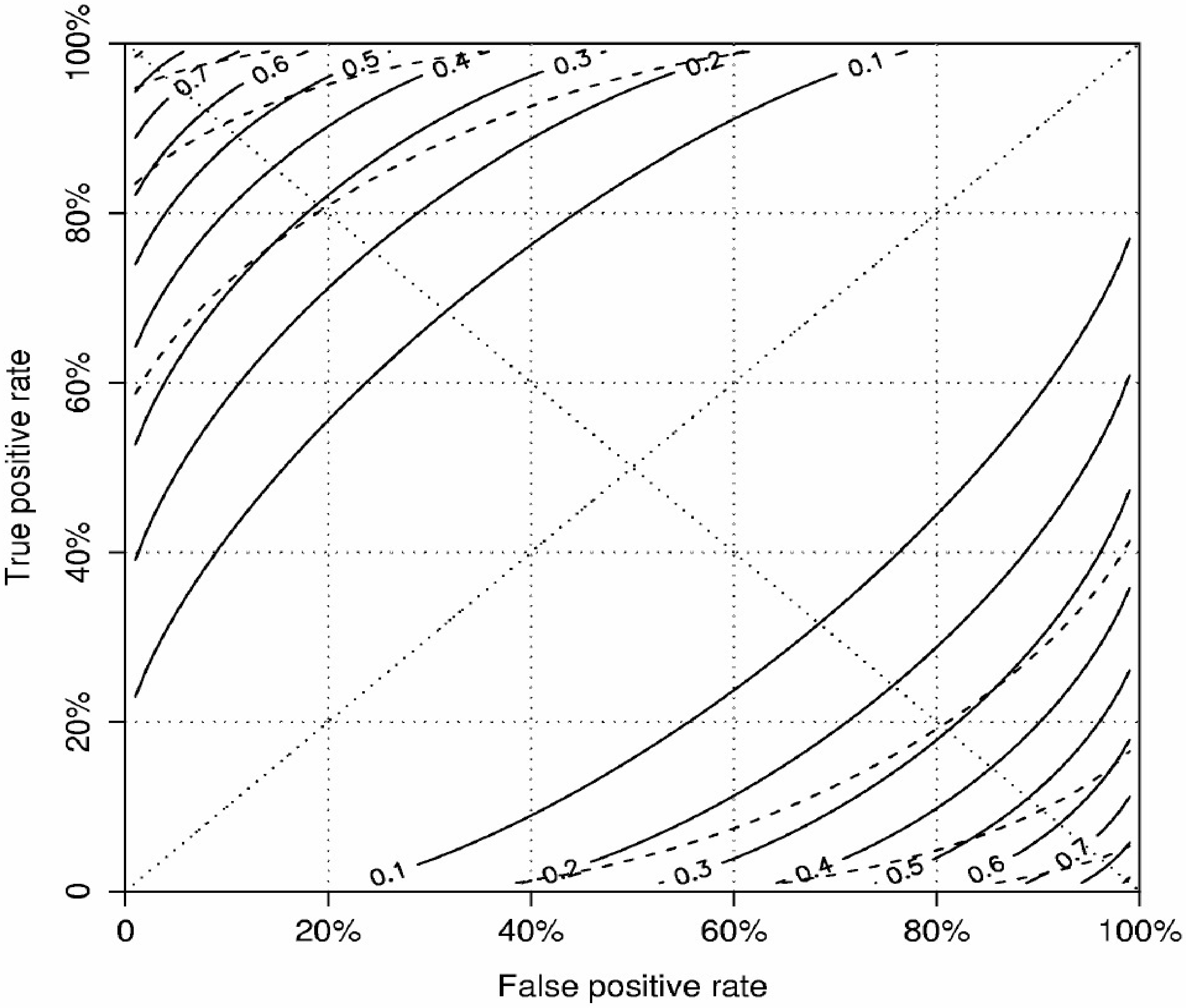

ROC Curve pos=44 neg=293 permutaciones=20
FisherVAR= 120
pos
0
10
20
30
40
neg
0
50
100
150
200
250
300

Impurity	$Imp(p,n)$	Relative impurity
Entropy	$-p\log p - n\log n$	
Gini	$4pn$	$\dfrac{1+c}{tpr + c\cdot fpr}\, tpr \cdot fpr$
DKM	$2\sqrt{pn}$	$\sqrt{tpr \cdot fpr}$

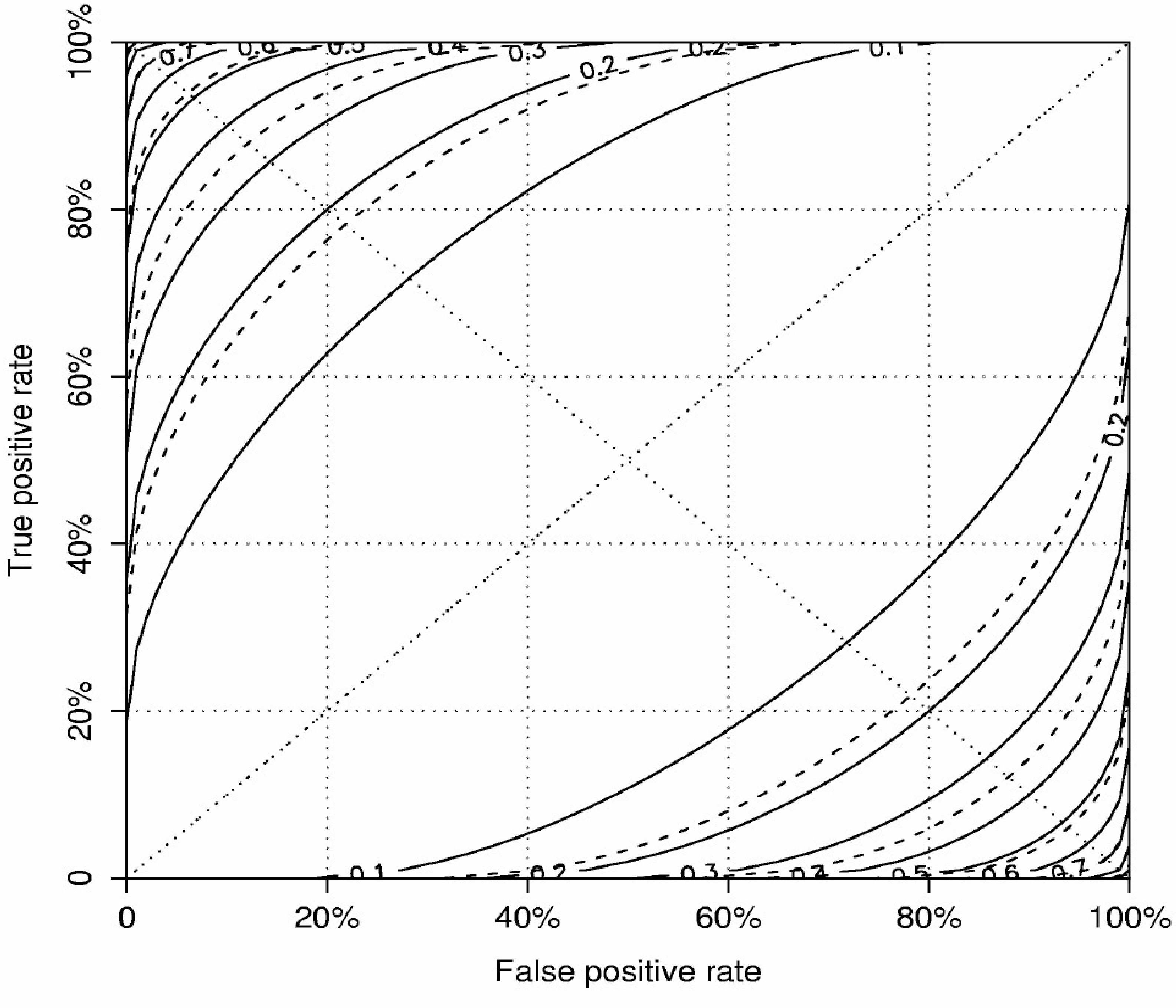


True positive rate
False positive rate
0
20%
40%
60%
80%
100%
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9

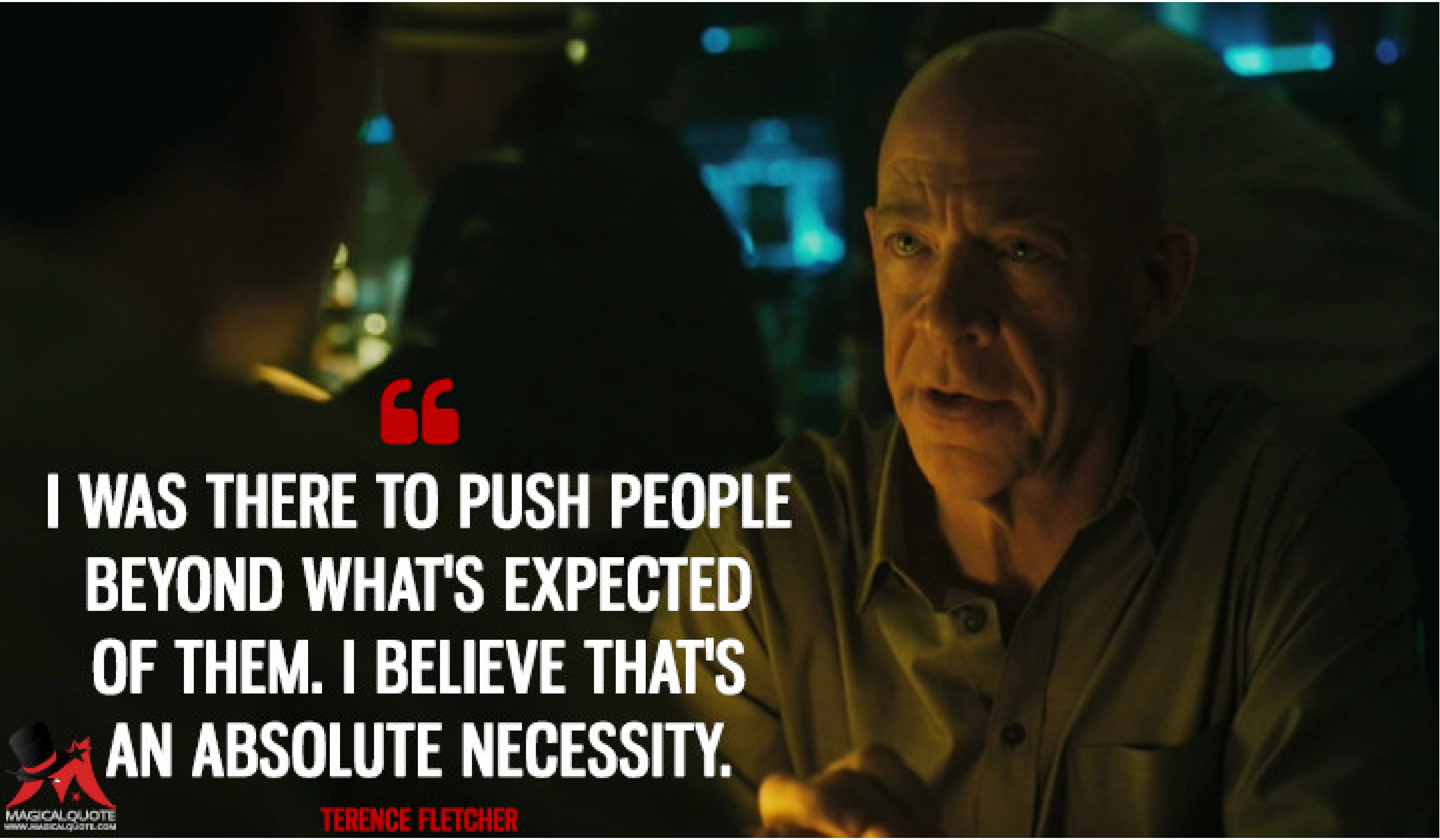
Gini-split



Information gain


True positive rate
False positive rate
0
20%
40%
60%
80%
100%
0
20%
40%
60%
80%
100%
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.2

DKM-split



"I WAS THERE TO PUSH PEOPLE BEYOND WHAT'S EXPECTED OF THEM. I BELIEVE THAT'S AN ABSOLUTE NECESSITY.

TERENCE FLETCHER

MAGICALQUOTE
WWW.MAGICALQUOTE.COM