

semana 1

luis

22/6/2021

Contents

¿Qué es la predicción?	1
componentes de una prediccion	1
ejemplo de SPAM	1
Importancia relativa de los pasos	4
Error en muestra y fuera de la muestra	6
Diseño del estudio de predicción	8
Tipos de error	9
para resultados binarios	9
para el caso continuo	12
Medidas de error comunes	13
Característica Operativa del Receptor (curvas)	13
Validacion Cruzada	15
enfoco 1	15
enfoco 2	18
recursos	19

¿Qué es la predicción?

componentes de una prediccion

question -> input data -> features -> algorithm -> parameters -> evaluation

ejemplo de SPAM

question

Comience con una pregunta general

¿Puedo detectar automáticamente correos electrónicos que son SPAM que no lo son?

Hazlo concreto

¿Puedo utilizar características cuantitativas de los correos electrónicos para clasificarlos como SPAM / HAM?

input data

Datos obtenidos de: <http://rss.acs.unt.edu/Rdoc/library/kernlab/html/spam.html>

features

Dear Jeff,

Can you send me your address so I can send you the invitation?

Thanks,

Ben

Frecuencia de “you” = $2/17 = 0.118$

```
library(kernlab)
data(spam)
head(spam)
```

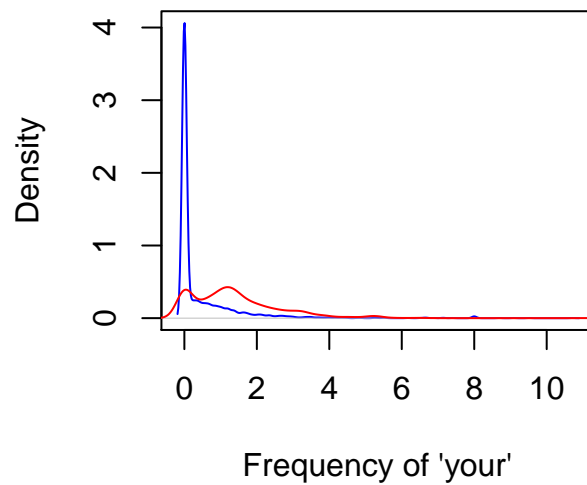
```
##  make address  all num3d  our over remove internet order mail receive will
## 1 0.00      0.64 0.64      0 0.32 0.00      0.00      0.00 0.00 0.00      0.00 0.64
## 2 0.21      0.28 0.50      0 0.14 0.28      0.21      0.07 0.00 0.94      0.21 0.79
## 3 0.06      0.00 0.71      0 1.23 0.19      0.19      0.12 0.64 0.25      0.38 0.45
## 4 0.00      0.00 0.00      0 0.63 0.00      0.31      0.63 0.31 0.63      0.31 0.31
## 5 0.00      0.00 0.00      0 0.63 0.00      0.31      0.63 0.31 0.63      0.31 0.31
## 6 0.00      0.00 0.00      0 1.85 0.00      0.00      1.85 0.00 0.00      0.00 0.00
##  people report addresses free business email  you credit your font num000
## 1 0.00      0.00      0.00 0.32      0.00 1.29 1.93      0.00 0.96      0 0.00
## 2 0.65      0.21      0.14 0.14      0.07 0.28 3.47      0.00 1.59      0 0.43
## 3 0.12      0.00      1.75 0.06      0.06 1.03 1.36      0.32 0.51      0 1.16
## 4 0.31      0.00      0.00 0.31      0.00 0.00 3.18      0.00 0.31      0 0.00
## 5 0.31      0.00      0.00 0.31      0.00 0.00 3.18      0.00 0.31      0 0.00
## 6 0.00      0.00      0.00 0.00      0.00 0.00 0.00      0.00 0.00      0 0.00
##  money hp hpl george num650 lab labs telnet num857 data num415 num85
## 1 0.00 0 0      0      0 0 0 0      0      0 0      0 0
## 2 0.43 0 0      0      0 0 0 0      0      0 0      0 0
## 3 0.06 0 0      0      0 0 0 0      0      0 0      0 0
## 4 0.00 0 0      0      0 0 0 0      0      0 0      0 0
## 5 0.00 0 0      0      0 0 0 0      0      0 0      0 0
## 6 0.00 0 0      0      0 0 0 0      0      0 0      0 0
##  technology num1999 parts pm direct cs meeting original project re edu
## 1      0      0.00      0 0      0.00 0      0      0.00      0 0.00 0.00
## 2      0      0.07      0 0      0.00 0      0      0.00      0 0.00 0.00
## 3      0      0.00      0 0      0.06 0      0      0.12      0 0.06 0.06
## 4      0      0.00      0 0      0.00 0      0      0.00      0 0.00 0.00
## 5      0      0.00      0 0      0.00 0      0      0.00      0 0.00 0.00
## 6      0      0.00      0 0      0.00 0      0      0.00      0 0.00 0.00
##  table conference charSemicolon charRoundbracket charSquarebracket
## 1      0      0      0.00      0.000      0
## 2      0      0      0.00      0.132      0
## 3      0      0      0.01      0.143      0
## 4      0      0      0.00      0.137      0
## 5      0      0      0.00      0.135      0
## 6      0      0      0.00      0.223      0
##  charExclamation charDollar charHash capitalAve capitalLong capitalTotal type
## 1      0.778      0.000      0.000      3.756      61      278 spam
## 2      0.372      0.180      0.048      5.114      101     1028 spam
## 3      0.276      0.184      0.010      9.821      485     2259 spam
## 4      0.137      0.000      0.000      3.537      40      191 spam
## 5      0.135      0.000      0.000      3.537      40      191 spam
```

```
## 6          0.000      0.000      0.000      3.000      15          54 spam
```

En este caso sencilla las características a escoger es la frecuencia de la palabra you

algorithm

```
plot(density(spam$your[spam$type=="nonspam"]),  
     col="blue",main="",xlab="Frequency of 'your'")  
lines(density(spam$your[spam$type=="spam"]),col="red")
```

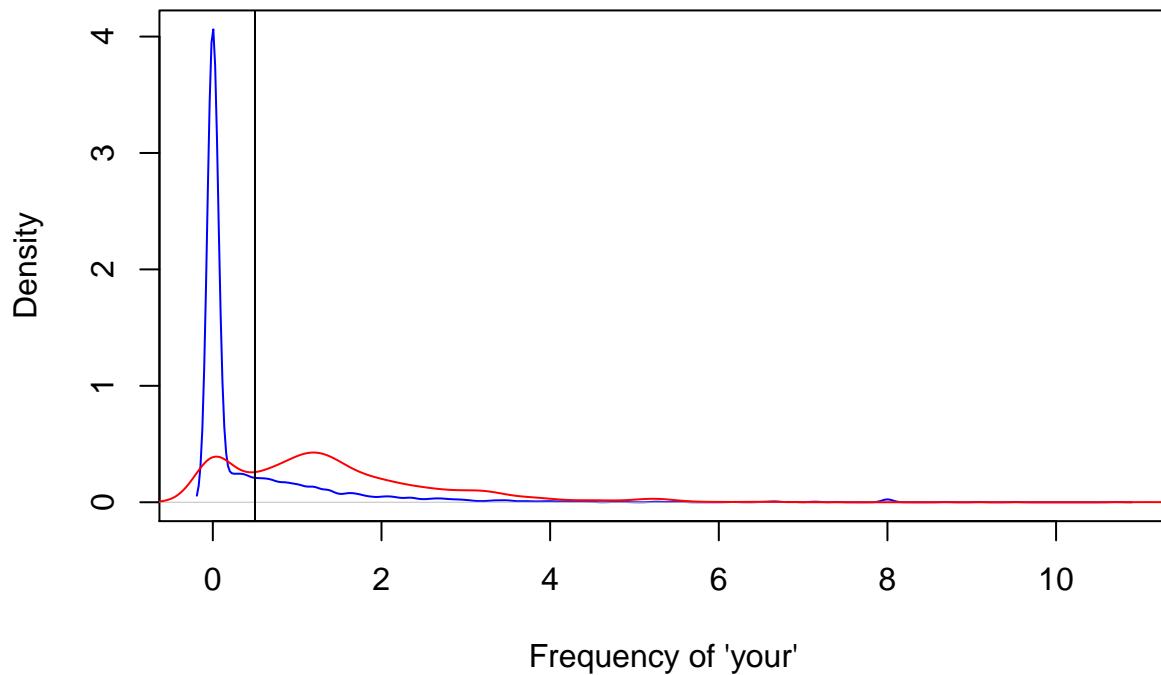


- Encuentra un valor C .
- frecuencia de 'your' > C predice "spam"

patameters

elegimos un valor de c de 0.5

```
plot(density(spam$your[spam$type=="nonspam"]),  
     col="blue",main="",xlab="Frequency of 'your'")  
lines(density(spam$your[spam$type=="spam"]),col="red")  
abline(v=0.5,col="black")
```



evaluation

```
prediction <- ifelse(spam$your > 0.5, "spam", "nonspam")
table(prediction, spam$type) / length(spam$type)
```

```
##
## prediction  nonspam    spam
## nonspam  0.4590306 0.1017170
## spam     0.1469246 0.2923278
```

precision $\approx 0.459 + 0.292 = 0.751$

Importancia relativa de los pasos

“La combinación de algunos datos y el doloroso deseo de una respuesta no garantiza que se pueda extraer una respuesta razonable de un conjunto de datos determinado.”

-John Tukey

input data

1. Puede ser fácil (clasificaciones de películas -> clasificaciones de películas nuevas)
2. Puede ser más difícil (datos de expresión genética -> enfermedad)
3. Depende de lo que sea una “buena predicción”.
4. A menudo, [más datos > mejores modelos] (<http://www.youtube.com/watch?v=yvDCzhbjYWs>)
5. ¡El paso más importante!

¡las características importan!

features

Propiedades de buenas características

- Conducir a la compresión de datos
- Conservar la información relevante
- Se crean en base al conocimiento de aplicaciones de expertos.

Errores comunes

- Intentando automatizar la selección de funciones
- No prestar atención a las peculiaridades específicas de los datos
- Tirar información innecesariamente

algorithm

Los algoritmos importan menos de lo que piensas

Los algoritmos importan menos de lo que piensas, y esto puede ser una fuente de sorpresa y frustración para algunas personas. Así que esta es una tabla en la que intentaron predecir una variedad de diferentes tareas de predicción, por ejemplo, una especie de tarea de segmentación, predicción de votos en la Cámara de Representantes de EE. UU., y predicción de formas de onda y un montón de otras tareas de predicción diferentes. Y así lo hicieron de dos formas diferentes, primero usaron algo llamado análisis discriminante lineal que es una especie de un predictor temprano muy básico que puede aprender. Y luego también probaron para cada dato configurado para encontrar el mejor algoritmo de predicción absoluto podrían haberlo hecho y luego esta tabla muestra el error de predicción de estos dos enfoques diferentes. Y puede ver que el mejor error de predicción es siempre un poco mejor que el error discriminante lineal. Pero, en realidad, no está tan lejos.

TABLE 1
Performance of linear discriminant analysis and the best result we found on ten randomly selected data sets

Data set	Best method e.r.	Lindisc e.r.	Default rule	Prop linear
Segmentation	0.0140	0.083	0.760	0.907
Pima	0.1979	0.221	0.350	0.848
House-votes16	0.0270	0.046	0.386	0.948
Vehicle	0.1450	0.216	0.750	0.883
Satimage	0.0850	0.160	0.758	0.889
Heart Cleveland	0.1410	0.141	0.560	1.000
Splice	0.0330	0.057	0.475	0.945
Waveform21	0.0035	0.004	0.667	0.999
Led7	0.2650	0.265	0.900	1.000
Breast Wisconsin	0.0260	0.038	0.345	0.963

Temas a considerar

The “Best” Machine Learning Method

Interpretable

Simple

Accurate

Fast
(to train and test)

Scalable

La predicción se trata de compensaciones de precisión

- Interpretabilidad versus precisión
- Velocidad versus precisión
- Simplicidad versus precisión
- Escalabilidad versus precisión (adaptación)

La interpretabilidad importa: declaraciones si / entonces pueden ser muy interpretables para algunas personas, y esa es la razón por la que a la gente le gustan cosas como los árboles de decisiones

Error en muestra y fuera de la muestra

En Error de muestra: La tasa de error que obtiene en el mismo conjunto de datos que usó para construir su predictor. Algunas veces llamado error de resustitución. (entrenamiento)

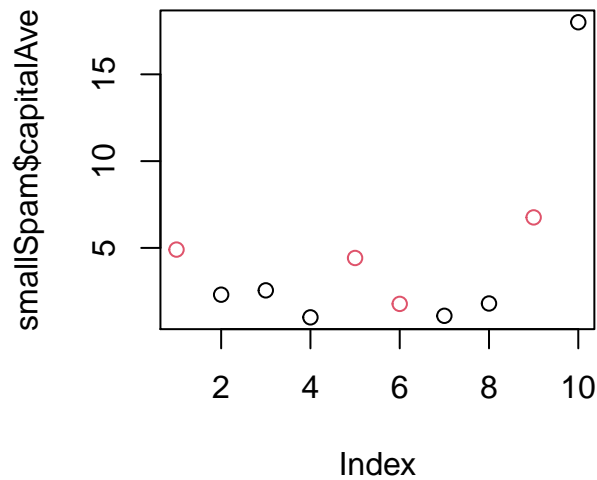
error fuera de la muestra: La tasa de error que obtiene en un nuevo conjunto de datos. A veces se denomina error de generalización. (prueba)

Ideas claves

1. Lo que le importa es el error fuera de muestra
2. En error de muestra < fuera de error de muestra
3. El motivo es el sobreajuste
 - Hacer coincidir su algoritmo con los datos que tiene

ejemplo motivacional

```
library(kernlab); data(spam); set.seed(333)
smallSpam <- spam[sample(dim(spam)[1],size=10),]
spamLabel <- (smallSpam$type=="spam")*1 + 1
plot(smallSpam$capitalAve,col=spamLabel)
```



Regla de predicción 1

- $\text{capitalAve} > 2.7 = \text{"spam"}$
- $\text{capitalAve} < 2.40 = \text{"no spam"}$
- capitalAve entre 2,40 y 2,45 = "spam"
- capitalAve entre 2,45 y 2,7 = "no spam"

Aplicando regla de predicción 1 a smallSpam

```
rule1 <- function(x){
  prediction <- rep(NA,length(x))
  prediction[x > 2.7] <- "spam"
  prediction[x < 2.40] <- "nonspam"
  prediction[(x >= 2.40 & x <= 2.45)] <- "spam"
  prediction[(x > 2.45 & x <= 2.70)] <- "nonspam"
  return(prediction)
}
table(rule1(smallSpam$capitalAve),smallSpam$type)
```

```
##
##           nonspam spam
## nonspam         5    1
## spam            1    3
```

Regla de predicción 2 * $\text{capitalAve} > 2.40 = \text{"spam"}$ * $\text{capitalAve} \leq 2.40 = \text{"nonspam"}$

Aplicando regla de predicción 2 a smallSpam

```
rule2 <- function(x){
  prediction <- rep(NA,length(x))
  prediction[x > 2.8] <- "spam"
  prediction[x <= 2.8] <- "nonspam"
  return(prediction)
}
table(rule2(smallSpam$capitalAve),smallSpam$type)
```

```
##
##           nonspam spam
## nonspam         5    1
## spam           1    3
```

Aplicar para completar los datos de spam

```
table(rule1(spam$capitalAve),spam$type)
```

```
##
##           nonspam spam
## nonspam    2141  588
## spam       647 1225
```

```
table(rule2(spam$capitalAve),spam$type)
```

```
##
##           nonspam spam
## nonspam    2224  642
## spam       564 1171
```

```
mean(rule1(spam$capitalAve)==spam$type)
```

```
## [1] 0.7315801
```

```
mean(rule2(spam$capitalAve)==spam$type)
```

```
## [1] 0.7378831
```

Mira la precisión

```
sum(rule1(spam$capitalAve)==spam$type)
```

```
## [1] 3366
```

```
sum(rule2(spam$capitalAve)==spam$type)
```

```
## [1] 3395
```

que esta pasando?(**sobreajuste**)

- Los datos tienen dos partes
 - Señal
 - Ruido
- El objetivo de un predictor es encontrar la señal
- Siempre puede diseñar un predictor perfecto en la muestra
- Capturas tanto la señal como el ruido cuando haces eso
- El predictor no funcionará tan bien en nuevas muestras

<http://en.wikipedia.org/wiki/Overfitting>

Diseño del estudio de predicción

1. Defina su tasa de error
2. Divida los datos en:
 - Entrenamiento, Pruebas, Validación
3. En el conjunto de entrenamiento, elige características.
 - Usar validación cruzada
4. En la función de predicción de selección del conjunto de entrenamiento
 - Usar validación cruzada

5. Si no hay validación
 - Aplicar 1x al conjunto de prueba
6. Si la validación
 - Aplicar al conjunto de prueba y refinar
 - Aplicar 1x a la validación

Evite tamaños de muestra pequeños

Reglas generales para el diseño de estudios de predicción

- Si tiene un tamaño de muestra grande
 - 60% de formación
 - Prueba del 20%
 - 20% de validación
- Si tiene una muestra de tamaño medio
 - 60% de formación
 - 40% de prueba
- Si tiene un tamaño de muestra pequeño
 - Hacer validación cruzada
 - Informe de advertencia de tamaño de muestra pequeño

algunos principios para recordar

- Deje a un lado la prueba / validación y *no lo mire*
- En general, muestra de entrenamiento y prueba son *aleatorios*
- Sus conjuntos de datos deben reflejar la estructura del problema
 - Si las predicciones evolucionan con el entrenamiento/prueba dividido en el tiempo en fragmentos de tiempo (llamado [backtesting] (<http://en.wikipedia.org/wiki/Backtesting>) en finanzas)
- Todos los subconjuntos deben reflejar la mayor diversidad posible
 - La asignación aleatoria hace esto
 - También puede intentar equilibrar por características, pero esto es complicado

Tipos de error

para resultados binarios

En general, **Positivo** = identificado y **negativo** = rechazado. Por lo tanto:

Verdadero positivo = identificado correctamente

False positivo = identificado incorrectamente

Verdadero negativo = correctamente rechazado

False negative = rechazado incorrectamente

Ejemplo de prueba médica:

Verdadero positivo = Personas enfermas correctamente diagnosticadas como enfermas

Falso positivo = Personas sanas identificadas incorrectamente como enfermas

Verdadero negativo = Personas sanas correctamente identificadas como sanas

False negative = Personas enfermas incorrectamente identificadas como saludables.

- *La sensibilidad* (tasa de verdaderos positivos) mide la proporción de positivos que se identifican correctamente (es decir, la proporción de aquellos que tienen alguna afección (afectados) que se identifican correctamente como portadores de la afección).

- La *especificidad* (tasa de verdaderos negativos) mide la proporción de negativos que se identifican correctamente (es decir, la proporción de aquellos que no tienen la afección (no afectados) que se identifican correctamente como personas que no padecen la afección).

		DISEASE	
		+	-
TEST	+	TP	FP
	-	FN	TN

Sensitivity

→ $\Pr(\text{positive test} \mid \text{disease})$

Specificity

→ $\Pr(\text{negative test} \mid \text{no disease})$

Positive Predictive Value

→ $\Pr(\text{disease} \mid \text{positive test})$

Negative Predictive Value

→ $\Pr(\text{no disease} \mid \text{negative test})$

Accuracy

→ $\Pr(\text{correct outcome})$

		DISEASE	
		+	-
TEST	+	TP	FP
	-	FN	TN

Sensitivity

→ $TP / (TP+FN)$

Specificity

→ $TN / (FP+TN)$

Positive Predictive Value

→ $TP / (TP+FP)$

Negative Predictive Value

→ $TN / (FN+TN)$

Accuracy

→ $(TP+TN) / (TP+FP+FN+TN)$

Ahora pongamos un ejemplo: Suponga que alguna enfermedad tiene una prevalencia del 0,1% en la población. Supongamos que tenemos un kit de prueba para esa enfermedad que funciona con una sensibilidad del 99% y una especificidad del 99%. ¿Cuál es la probabilidad de que una persona tenga la enfermedad dado que el resultado de la prueba es positivo, si seleccionamos al azar un sujeto de

- la población en general?
- una subpoblación de alto riesgo con una prevalencia de enfermedad del 10%?

Población general

		DISEASE	
		+	-
TEST	+	99	999
	-	1	98901

Población general como fracciones

		DISEASE	
		+	-
TEST	+	99	999
	-	1	98901

Sensitivity

$$\rightarrow 99 / (99+1) = 99\%$$

Specificity

$$\rightarrow 98901 / (999+98901) = 99\%$$

Positive Predictive Value

$$\rightarrow 99 / (99+999) \approx 9\%$$

Negative Predictive Value

$$\rightarrow 98901 / (1+98901) > 99.9\%$$

Accuracy

$$\rightarrow (99+98901) / 100000 = 99\%$$

Subpoblación en riesgo

		DISEASE	
		+	-
TEST	+	9900	900
	-	100	89100

Subpoblación en riesgo como fracciones

		DISEASE	
		+	-
TEST	+	9900	900
	-	100	89100

Sensitivity	→ 9900 / (9900+100) = 99%
Specificity	→ 89100 / (900+89100) = 99%
Positive Predictive Value	→ 9900 / (9900+900) ≈ 92%
Negative Predictive Value	→ 89100 / (100+89100) ≈ 99.9%
Accuracy	→ (9900+89100) / 100000 = 99%

para el caso continuo

Error cuadrático medio(MSE) :

$$\frac{1}{n} \sum_{i=1}^n (Predicción_i - Verdad_i)^2$$

Raíz de el error cuadrático medio (RMSE) :

$$\sqrt{\frac{1}{n} \sum_{i=1}^n (Predicción_i - Verdad_i)^2}$$

Medidas de error comunes

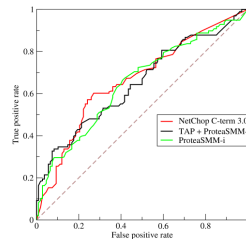
1. Error cuadrático medio (o error cuadrático medio)
 - Datos continuos, sensibles a valores atípicos
2. Desviación absoluta mediana
 - Datos continuos, a menudo más robustos
3. Sensibilidad (recuerdo)
 - Si quieres pocos positivos perdidos
4. Especificidad
 - Si quieres algunos negativos llamados positivos
5. Precisión
 - Pondera los falsos positivos / negativos por igual
6. Concordancia
 - Un ejemplo es [kappa] (http://en.wikipedia.org/wiki/Cohen%27s_kappa)
7. Valor predictivo de un positivo (precisión)
 - Cuando está revisando y el predominio es bajo

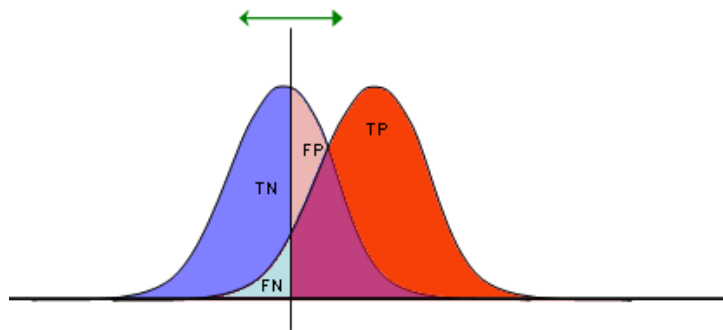
Característica Operativa del Receptor (curvas)

- En la clasificación binaria está prediciendo una de dos categorías
 - Vivo muerto
 - Haga clic en el anuncio / no haga clic
- Pero tus predicciones suelen ser cuantitativas
 - Probabilidad de estar vivo
 - Predicción en una escala del 1 al 10
- El *cutoff* que elijas da resultados diferentes

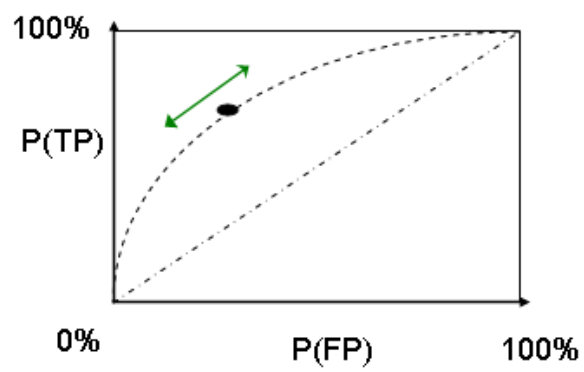
La curva ROC , es un diagrama gráfico que ilustra la capacidad de diagnóstico de un sistema clasificador binario a medida que varía su umbral de discriminación. El método fue desarrollado originalmente para operadores de receptores de radar militares a partir de 1941, lo que llevó a su nombre.

La curva ROC se crea trazando la tasa de verdaderos positivos (TPR) frente a la tasa de falsos positivos (FPR) en varios valores de umbral

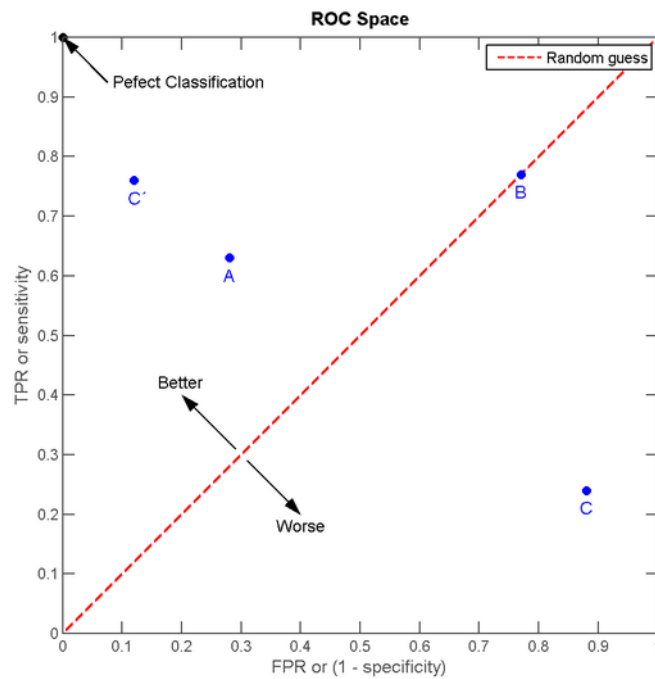




TP	FP
FN	TN
1	1



- $AUC = 0.5$: adivinación aleatoria
- $AUC = 1$: clasificador perfecto
- En general, el AUC superior a 0,8 se considera “bueno”



Validacion Cruzada

enfoque 1

1. La precisión en el conjunto de entrenamiento (precisión de resustitución) es optimista
2. Una mejor estimación proviene de un conjunto independiente (precisión del conjunto de prueba)
3. Pero no podemos usar el conjunto de prueba al crear el modelo o se convierte en parte del conjunto de entrenamiento.
4. Por tanto, estimamos la precisión del conjunto de prueba con el conjunto de entrenamiento.

Acercarse:

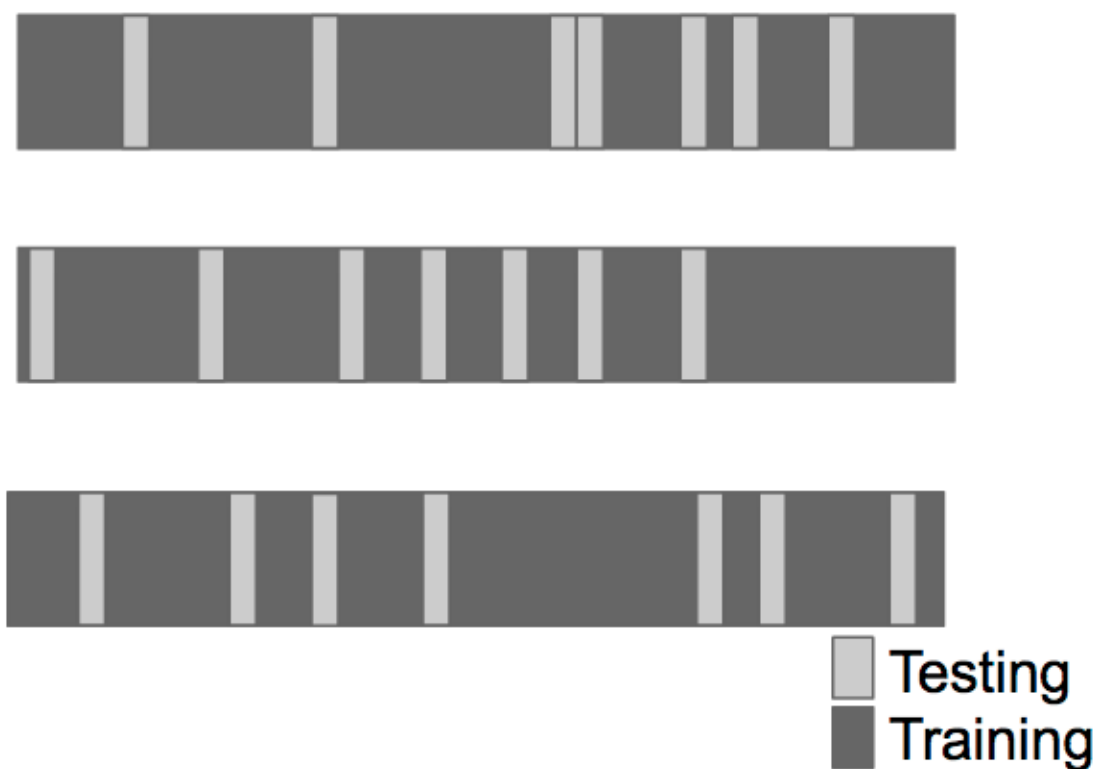
1. Usa el conjunto de entrenamiento
2. Divídalo en conjuntos de entrenamiento/prueba
3. Construya un modelo en el conjunto de entrenamiento.
4. Evaluar en el equipo de prueba
5. Repetir y promediar los errores estimados

Usado para:

1. Seleccionar variables para incluir en un modelo
2. Elegir el tipo de función de predicción que se utilizará
3. Seleccionar los parámetros en la función de predicción
4. Comparación de diferentes predictores

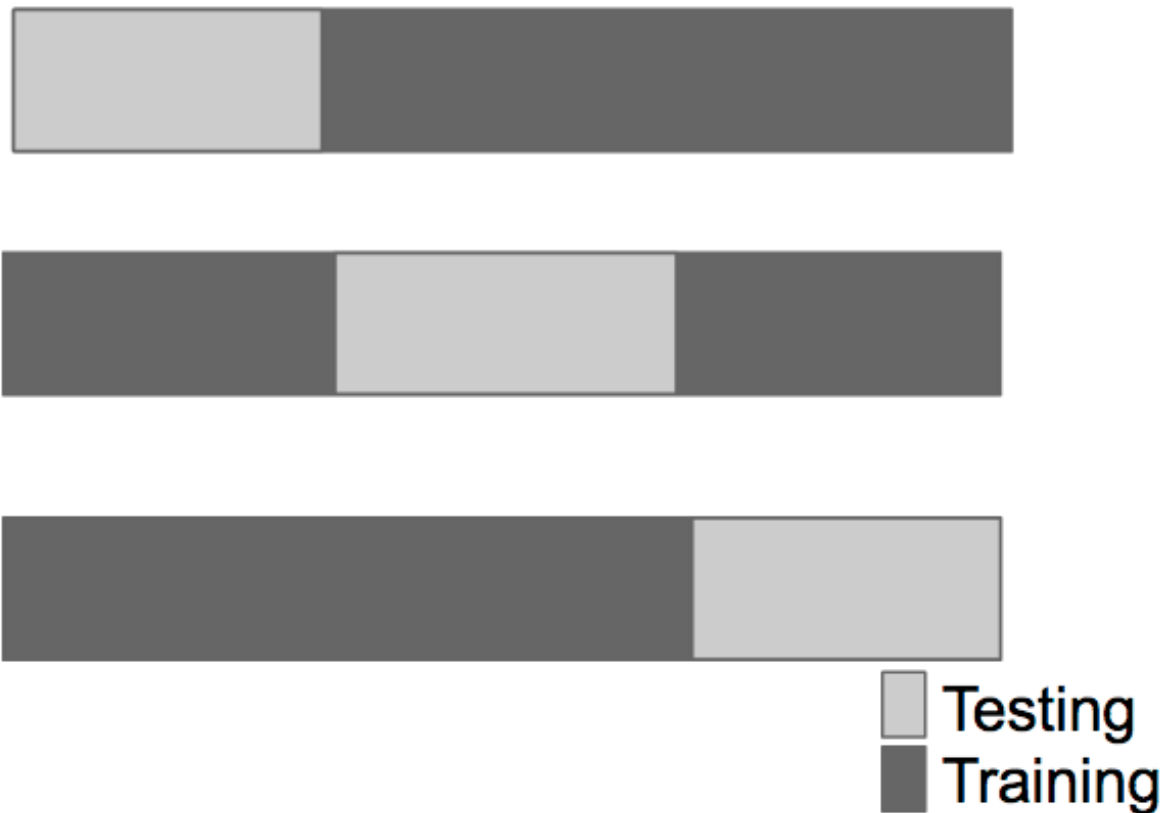
Diferentes formas de dividir los datos

1. Random subsampling



Este método consiste al dividir aleatoriamente el conjunto de datos de entrenamiento y el conjunto de datos de prueba. Para cada división la función de aproximación se ajusta a partir de los datos de entrenamiento y calcula los valores de salida para el conjunto de datos de prueba. El resultado final se corresponde a la media aritmética de los valores obtenidos para las diferentes divisiones. La ventaja de este método es que la división de datos entrenamiento-prueba no depende del número de iteraciones. Pero, en cambio, con este método hay algunas muestras que quedan sin evaluar y otras que se evalúan más de una vez, es decir, los subconjuntos de prueba y entrenamiento se pueden solapar



2. K-fold



Se dividen en K subconjuntos. Uno de los subconjuntos se utiliza como datos de prueba y el resto ($K-1$) como datos de entrenamiento. El proceso de validación cruzada es repetido durante k iteraciones, con cada uno de los posibles subconjuntos de datos de prueba. Finalmente se realiza la media aritmética de los resultados de cada iteración para obtener un único resultado. Este método es muy preciso puesto que evaluamos a partir de K combinaciones de datos de entrenamiento y de prueba, pero aun así tiene una desventaja, y es que, a diferencia del método de retención, es lento desde el punto de vista computacional

3. Leave one out



 Testing
 Training

Implica separar los datos de forma que para cada iteración tengamos una sola muestra para los datos de prueba y todo el resto conformando los datos de entrenamiento. La evaluación viene dada por el error, y en este tipo de validación cruzada el error es muy bajo, pero en cambio, a nivel computacional es muy costoso, puesto que se tienen que realizar un elevado número de iteraciones, tantas como N muestras tengamos y para cada una analizar los datos tanto de entrenamiento como de prueba.

Consideraciones

- Para los datos de series de tiempo, los datos deben usarse en “fragmentos”
- Para validación cruzada de k-fold
 - Mayor k = menos sesgo, más varianza
 - Más pequeño k = más sesgo, menos varianza
- El muestreo aleatorio debe realizarse *sin reemplazo*
- El muestreo aleatorio con reemplazo es el *bootstrap*
 - Subestima el error
 - Se puede corregir, pero es complicado ([0.632 Bootstrap] (<http://www.jstor.org/discover/10.2307/2965703?uid=2&uid=4&sid=21103054448997>))
- Si realiza una validación cruzada para elegir la estimación de predictores, debe estimar los errores en datos independientes.

enfoque 2

cuando hacemos validacion cruzada desde el enfoque 1 existe la posibilidad de que pase el mismo efecto en el conjunto de prueba que el que pasa en el conjunto de entrenamiento, es decir que que haya un sobreajuste generado por el conjunto de prueba

Para solucionar esto, podemos introducir un tercer conjunto, el Conjunto de validación cruzada , para que sirva como conjunto intermedio. Entonces, nuestro conjunto de prueba nos dará un error preciso y no

optimista.

Una forma de ejemplo de dividir nuestro conjunto de datos en tres conjuntos es:

- Conjunto de entrenamiento: 60%
- Conjunto de validación cruzada: 20%
- Equipo de prueba: 20%

Ahora podemos calcular tres valores de error separados para los tres conjuntos diferentes.

1. Optimice los parámetros en Θ utilizando el conjunto de entrenamiento.
2. Encuentre el menor error usando el conjunto de validación cruzada.
3. Estime el error de generalización usando el conjunto de prueba (si es muy grande hay un sobreajuste)

recursos

- The elements of statistical learning
- List of machine learning resources on Quora
- List of machine learning resources from Science
- Advanced notes from MIT open courseware
- Advanced notes from CMU
- Kaggle - machine learning competitions
- ROC