

# Desempeño de Modelos

Jorge Gallego

Facultad de Economía, Universidad del Rosario

Junio 27 de 2017

# Introducción

- Hemos visto algunas técnicas para evaluar el desempeño de un modelo
- Naturalmente, nos interesa que los algoritmos pronostiquen “bien” en datos futuros
- Pero existen diferentes concepciones, o métricas, para evaluar buenos pronósticos
- Veremos algunas de estas medidas de desempeño de los algoritmos

# Precisión Total del Modelo

- Quizás la manera más obvia de medir el desempeño de un algoritmo es a través de su precisión total
- Representa la proporción total de casos clasificados correctamente, sobre la proporción total de casos a clasificar
- Sin embargo, hay que tener precaución con esta medida
- Consideremos el siguiente caso: queremos predecir un defecto genético en bebés
- Supongamos que 1 de cada 1000 bebés vienen con el defecto genético

# Precisión Total del Modelo

- Consideremos el siguiente algoritmo: clasificar cada nuevo bebé como sano
- ¿Cuál sería la precisión de este algoritmo?
- Se equivocaría una en 1000 veces. Una precisión del 99.9%
- Lo evaluaríamos con un excelente algoritmo. ¿Lo es?
- Naturalmente no lo es. En este ejemplo, nos interesa predecir correctamente a los bebés que nacen con el defecto

# Precisión Total del Modelo

- Es decir, queremos minimizar la tasa de “falsos negativos”
- ¡Pero en este caso nos equivocamos en el 100% de bebés que nacen con el defecto!
- El algoritmo no es muy útil en este caso
- Necesitamos otras métrica para evaluar el desempeño del modelo
- Cuál sea más útil depende del problema que se esté buscando resolver

# Matrices de Confusión

- En tareas de clasificación, son varias las cantidades de interés
- **Verdaderos Positivos:** casos correctamente clasificados en la categoría de interés
- **Verdaderos Negativos:** casos correctamente clasificados fuera de la categoría de interés
- **Falsos Positivos:** casos incorrectamente clasificados en la categoría de interés
- **Falsos Negativos:** casos incorrectamente clasificados fuera de la categoría de interés

## Ejemplo: Predicción de SMS Spam

		Predicted to be Spam	
		no	yes
Actually Spam	no	<div>TN</div> <div>True Negative</div>	<div>FP</div> <div>False Positive</div>
	yes	<div>FN</div> <div>False Negative</div>	<div>TP</div> <div>True Positive</div>

Fuente: Lantz (2015)

# Medidas Básicas de Desempeño

De la matriz de confusión podemos definir dos medidas básicas de desempeño:

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN}$$

$$Error\ Rate = 1 - Accuracy = 1 - \frac{TP + TN}{TP + TN + FP + FN}$$

En principio, buscamos maximizar la precisión y minimizar el error



# Otras Medidas de Desempeño

- Pero como lo refleja el ejemplo introductorio, la precisión no siempre es la mejor medida
- Estudiaremos otras métricas para medir desempeño:
- El estadístico Kappa, la sensibilidad y la especificidad
- Así como algunas medidas gráficas:
- La curva ROC y el área AUC

# Estadístico Kappa

- El estadístico kappa es un ajuste de la precisión total
- Ajusta por la probabilidad de acertar simplemente por cuestiones del azar
- Es particularmente útil con bases de datos con imbalance de clase grande
- En los que es fácil tener una precisión alta simplemente asignando todos los casos a la clase más popular

# Estadístico Kappa

- Se define como:

$$\kappa = \frac{Pr(a) - Pr(e)}{1 - Pr(e)}$$

- donde  $P(a)$  es la proporción de concordancia actual
- $P(e)$  es la concordancia esperada entre el clasificador y los valores verdaderos, asumiendo que se escogen al azar

# Estadístico Kappa

- El estadístico kappa es un número entre 0 y 1
- Cuánto más grande sea, mayor es la precisión
- Suele usarse la siguiente convención:
  - ▶  $\kappa \leq 0.2 \rightarrow$  concordancia muy baja
  - ▶  $0.2 < \kappa \leq 0.4 \rightarrow$  concordancia baja
  - ▶  $0.4 < \kappa \leq 0.6 \rightarrow$  concordancia moderada
  - ▶  $0.6 < \kappa \leq 0.8 \rightarrow$  concordancia alta
  - ▶  $0.8 < \kappa \leq 1 \rightarrow$  concordancia muy alta

## Ejemplo: Predicción de SMS Spam

sms_results\$actual_type	sms_results\$predict_type		Row Total
	ham	spam	
ham	1203	4	1207
	16.128	127.580	
	0.997	0.003	0.868
	0.975	0.026	
	0.865	0.003	
spam	31	152	183
	106.377	841.470	
	0.169	0.831	0.132
	0.025	0.974	
	0.022	0.109	
Column Total	1234	156	1390
	0.888	0.112	

## Ejemplo: Predicción de SMS Spam

En este ejemplo:

$$P(a) = \frac{1203 + 152}{1390} = 0.974$$

$$\begin{aligned} P(e) &= Pr(\text{tipo real ham}) * Pr(\text{pred ham}) \\ &\quad + Pr(\text{tipo real spam}) * Pr(\text{pred spam}) \\ &= \left( \frac{1207}{1390} \right) * \left( \frac{1234}{1390} \right) + \left( \frac{183}{1390} \right) * \left( \frac{156}{1390} \right) \\ &= 0.868 * 0.888 + 0.132 * 0.112 \\ &= 0.786 \end{aligned}$$

## Ejemplo: Predicción de SMS Spam

De esta forma, el estimador kappa en este ejemplo es:

$$\kappa = \frac{0.974 - 0.786}{1 - 0.786} = 0.879$$

Que representa una concordancia muy alta. Los pronósticos son más que simple azar

# Sensibilidad y Especificidad

- Dos medidas adicionales capturan un trade-off inherente a estos problemas
- Algoritmos muy conservadores o muy agresivos
- Un filtro de spam muy agresivo clasificaría como spam casi todo, incluyendo correos buenos. Muchos falsos positivos
- Un filtro muy conservador clasificaría poco como spam, protegiendo los buenos. Muchos falsos negativos
- Las medidas de sensibilidad y especificidad capturan este trade-off



# Sensibilidad y Especificidad

- Los definimos de la siguiente forma:

$$sensitivity = \frac{TP}{TP + FN}$$

$$specificity = \frac{TN}{TN + FP}$$

- La sensibilidad mide la fracción positivos correctamente clasificados. Qué tanto acertamos con los positivos
- La especificidad mide la fracción de negativos correctamente clasificados. Qué tanto acertamos con los negativos

# Sensibilidad y Especificidad

- En el ejemplo de los SMS:

$$sensitivity = \frac{152}{183} = 0.831$$

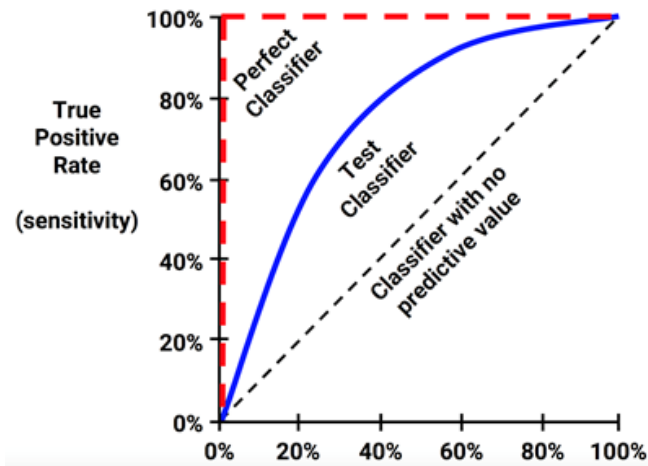
$$specificity = \frac{1203}{1207} = 0.997$$

- La idea es balancear estas dos medidas
- Cuál sea el punto ideal depende del problema en específico

# Curvas ROC

- El trade-off entre sensibilidad y especificidad lo podemos expresar gráficamente
- Por medio de las curvas ROC (Receiver Operating Characteristic)
- Se busca aumentar la detección de verdaderos positivos, sin aumentar los falsos positivos
- Ubicamos los verdaderos positivos en el eje  $y$  y los falsos positivos en el  $x$

# Curvas ROC



Fuente: Lantz (2015)

# Curvas ROC

- La recta de 45 grados representa el clasificador inútil:
- El que aumenta el número de verdaderos positivos aumentando los falsos positivos
- La línea punteada es el clasificador perfecto: alcanza el máximo de verdaderos positivos al mínimo de falsos positivos
- La curva se construye ordenando las predicciones del modelo de la clase positiva, empezando por los valores más altos
- Se seleccionan umbrales para clasificar como un positivo. Se calculan las tasas de verdaderos positivos y falsos positivos
- Y se grafican los diferentes umbrales

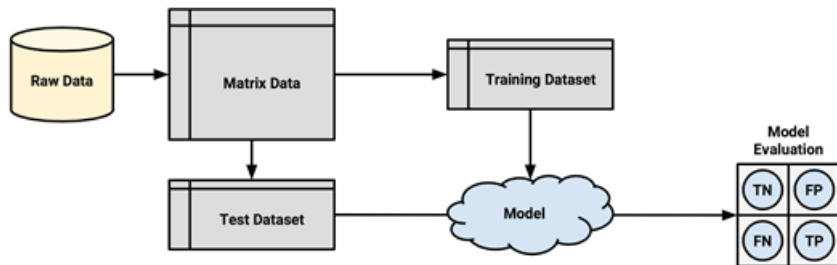
# Area Bajo la Curva (AUC)

- El área bajo la curva mide el desempeño del clasificador bajo el criterio ROC
- El clasificador inútil tiene un  $AUC=0.5$ . Recoge el 50% del área total del cuadrado
- El clasificador perfecto tiene  $AUC=1$ . Recoge el 100% del área
- Los clasificadores reales tendrán un AUC entre 0.5 y 1. Cuánto más grande sea AUC, mejor será el clasificador

# Desempeño Futuro

- Hemos visto un método simple para evaluar el desempeño futuro de un algoritmo
- Dividir aleatoriamente la base total en dos: entrenamiento y prueba
- Esta técnica se conoce como el *holdout method*
- Qué proporción se destine a cada grupo ha de depender de cuántos datos se tengan

# Desempeño Futuro



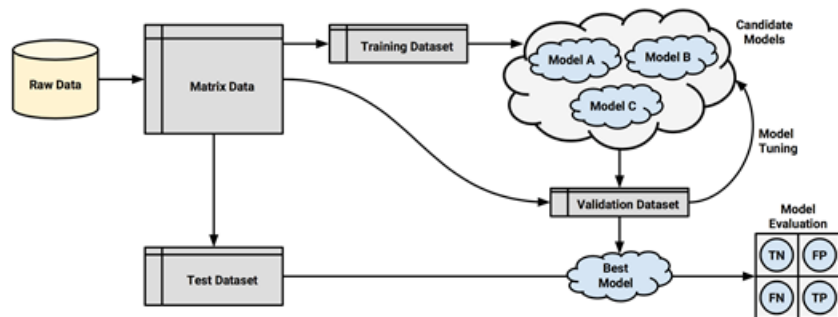
Fuente: Lantz (2015)



# Desempeño Futuro

- En la práctica, el resultado del modelo en la base de prueba no debería afectar el tipo de modelo utilizado
- Hemos violado este principio en algunos ejemplos. Cuando cambiamos el modelo luego de ver su desempeño en la prueba
- ¿Cómo evitar el sesgo que implica esta práctica?
- Dividir los datos en tres: entrenamiento, validación y prueba
- El modelo se afina con los datos de validación. Luego se pone a prueba con los de prueba

# Desempeño Futuro



Fuente: Lantz (2015)

# Holdout Repetido

- Pero el método de *holdout* puede tener problemas
- Por ejemplo, las muestras de entrenamiento, validación y prueba pueden quedar desbalanceadas
- En especial en el outcome a predecir. Esto es particularmente probable con muestras pequeñas
- Todo lo cual puede conducir a sesgos en las predicciones
- Las técnicas de holdouts repetidos permiten apaciguar algunos de estos problemas

## Cross-Validation

- El caso más utilizado de holdout repetido es la validación cruzada (*cross validation*)
- La técnica más popular es la de *k-fold cross validation*, y en particular, la de 10-fold
- Se dividen los datos en 10 partes iguales
- Cada parte es usada una vez como base de prueba, luego de entrenar el modelo con las 9 partes restantes
- Al final, se promedian las medidas de desempeño que se tienen para los 10 ejercicios realizados