

# Evaluación de Modelos de Aprendizaje Automático

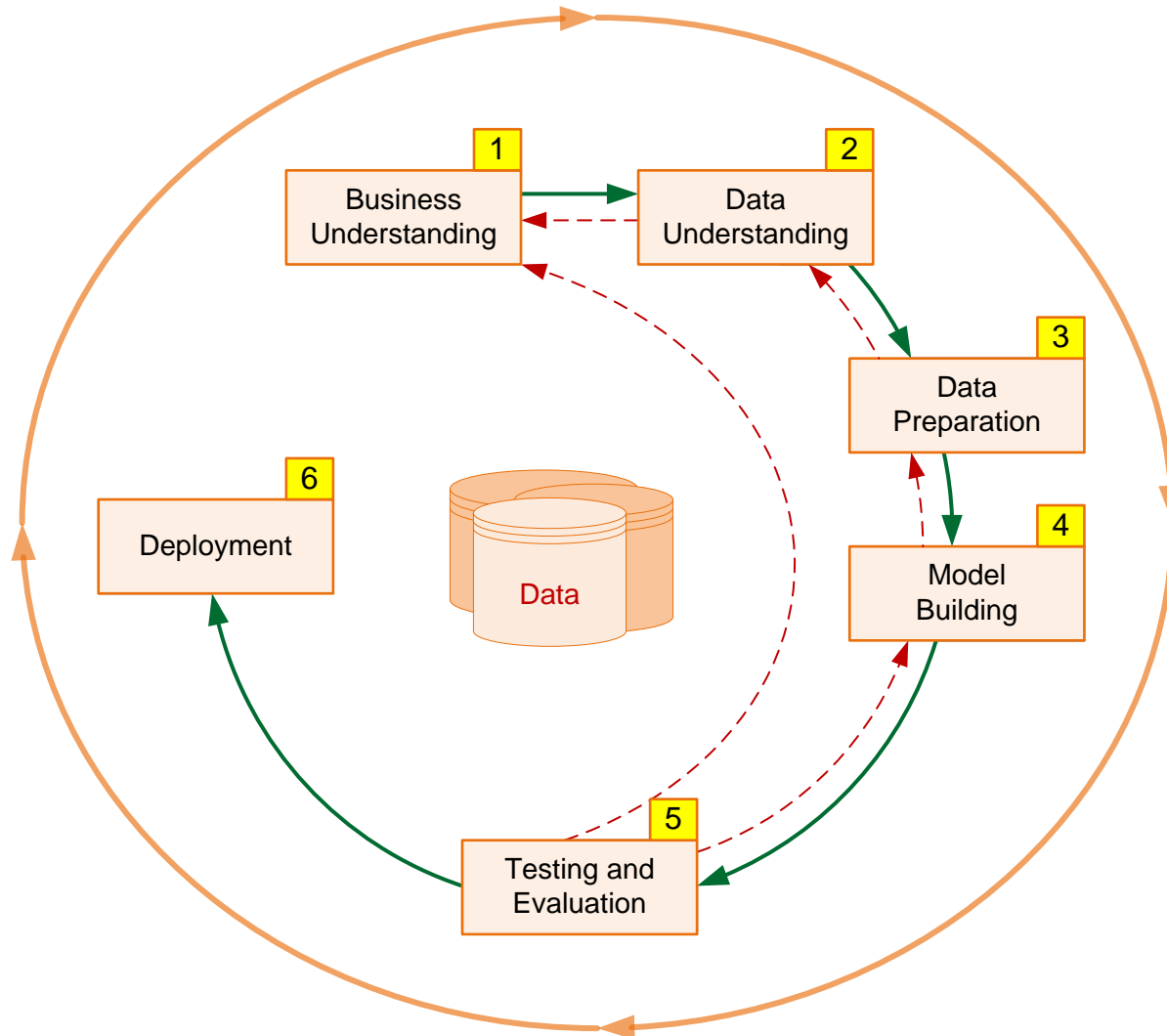
MARIA DE LOS ANGELES CONSTANTINO GONZALEZ



Tecnológico  
de Monterrey



# Introducción CRISP-DM / Evaluación de modelos



# Evaluación de Modelos

- Es una de las fases principales en todo el proceso de análisis de datos
- La calidad de un modelo aprendido se evalúa mediante una o varias métricas que cuantifican el desempeño del modelo
  - La más simple: Exactitud (accuracy) proporción de aciertos en la clasificación dada
- ¿Para qué evaluar?
  - Comparar distintos modelos, para elegir el mejor
  - Estimar cómo se comportará el modelo, una vez puesto “en producción”.
  - Convencer al “cliente” de que el modelo cumplirá su propósito

# Evaluación - Introducción

- Un algoritmo específico no es el mejor para todos los conjuntos de datos y en todos los casos.
- Se necesita una forma de elegir entre modelos:
  - Existen diferentes tipos de modelos, cada uno con hiperparámetros de ajuste
- Para encontrar la mejor solución, es necesario llevar a cabo muchos experimentos, **evaluar diferentes algoritmos de aprendizaje**, ajustar sus hiperparámetros y comparar su desempeño

# Evaluación del Modelo: Metodologías y Métricas

## ■ Metodologías

- ¿Cómo diseñamos el experimento de evaluación del modelo?
- Es fundamental no hacer la evaluación final del desempeño del modelo sobre conjuntos de datos que:
  - Se hayan usado para el aprendizaje del modelo
  - Se hayan usado para el ajuste del modelo
- Utilizar un **procedimiento de evaluación** para estimar qué tan bien se generalizará un modelo a los datos fuera de la muestra.  
Ejemplo: Validación cruzada.

## ■ Métricas

¿Cómo medimos el desempeño de un modelo?

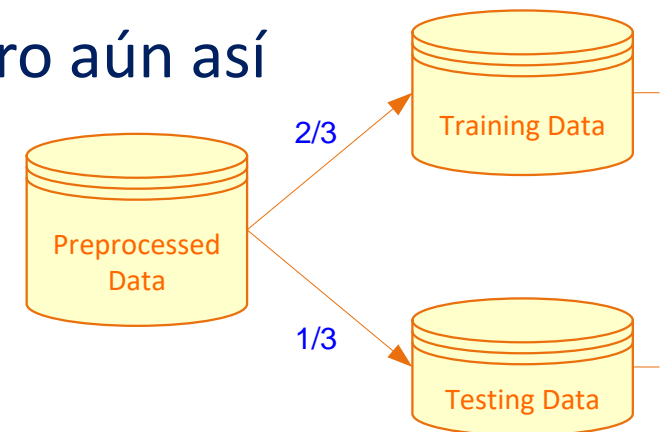
Requiere **métricas de evaluación** para cuantificar el desempeño del modelo.

Uso de métricas en paquete sklearn (metrics)

# Procedimientos de evaluación

## División Entrenamiento-Prueba (Holdout)

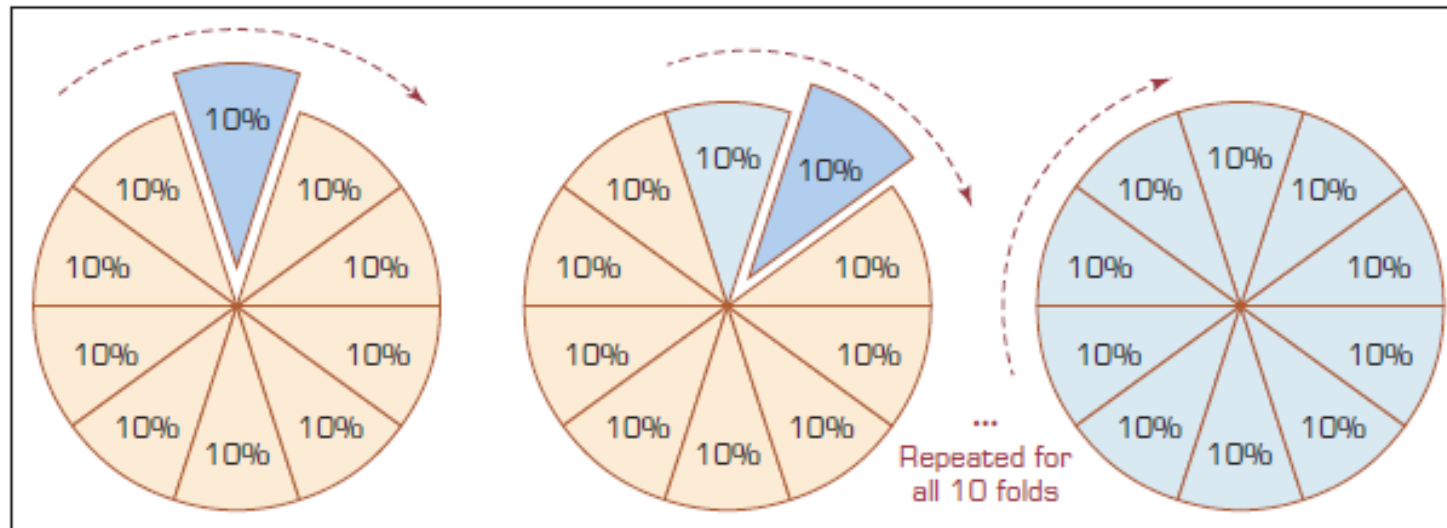
- El conjunto de datos disponible  $D$  se divide en dos subconjuntos independientes,
  - el conjunto de entrenamiento  $D_{train}$  (para aprender un modelo)
  - el conjunto de prueba  $D_{test}$  (para probar el modelo también llamado **holdout set**)
- **Importante:** el conjunto de entrenamiento no debe utilizarse en las pruebas y el conjunto de pruebas no debe utilizarse en el aprendizaje. El conjunto de pruebas no vistas proporciona una estimación sin sesgo de la precisión
- Este método se utiliza principalmente cuando el conjunto de datos  $D$  es grande.
- Mejor estimación del desempeño fuera de la muestra, pero aún así es una estimación de "alta varianza"
- Útil debido a su velocidad, simplicidad y flexibilidad



# Procedimientos de evaluación

## Validación cruzada en k partes (k-fold cross validation)

- Los datos disponibles se particionan en k partes independientes de igual tamaño (Ej.  $k=5$ ,  $k=10$ ).
- Se utiliza cada subconjunto como conjunto de prueba y combina los subconjuntos  $k-1$  restantes como conjunto de entrenamiento.
- Es una manera de evaluar qué tan bueno sería (en términos de generalización) un algoritmo de aprendizaje sobre un conjunto de entrenamiento dado

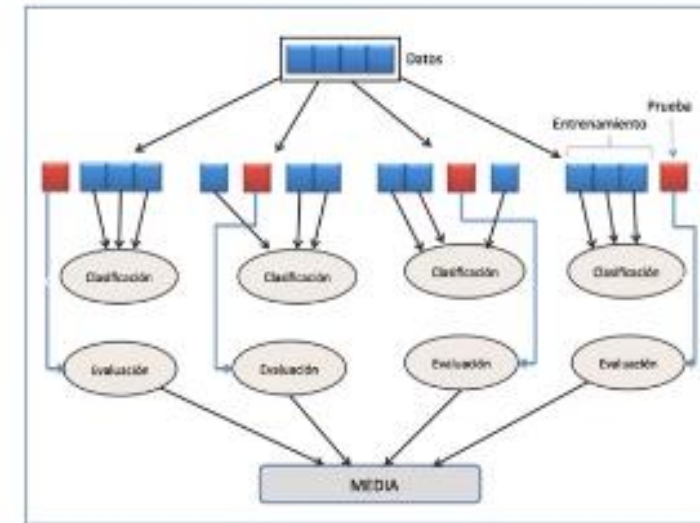


# Procedimientos de evaluación

## Validación cruzada - Observaciones

- El procedimiento se ejecuta  $k$  veces, lo que da  $k$  evaluaciones de acuerdo a una métrica (exactitud).
- La precisión estimada final del aprendizaje es el promedio de las  $k$  evaluaciones.
- Este método se utiliza cuando no se tienen muchos datos.
- Se ejecuta “ $k$ ” veces más lento que la división entrenamiento/prueba
- Es demasiado caro, tiene que crear  $k$  modelos

Validación cruzada o Cross Validation  
Ejemplo de  $k$ -fold Cross Validation con  $k = 4$  y un clasificador





# Evaluación del Modelo: Metodologías y Métricas

## ■ Metodologías

- ¿Cómo diseñamos el experimento de evaluación del modelo?
- Es fundamental no hacer la evaluación final del desempeño del modelo sobre conjuntos de datos que:
  - Se hayan usado para el aprendizaje del modelo
  - Se hayan usado para el ajuste del modelo
- Utilizar un **procedimiento de evaluación** para estimar qué tan bien se generalizará un modelo a los datos fuera de la muestra.  
Ejemplo: validación cruzada

## → ■ Métricas

¿Cómo medimos el desempeño de un modelo?

Requiere **métricas de evaluación** para cuantificar el desempeño del modelo.

Uso de métricas en paquete sklearn (metrics)

# Métricas de evaluación del modelo

- **Modelos de regresión:**

- Error Absoluto Medio (MAE)

- Error Cuadrático Medio (SME)

- Raiz del Error Cuadrático Medio (RSME)

- **Modelos de clasificación:**

- Exactitud (accuracy) de clasificación

- Matriz de confusión: precisión, sensibilidad (recall), especificidad.

- ROC-AUC

# Métricas de Evaluación para Problemas de Clasificación

## ■ Exactitud

- **Exactitud de Clasificación:** Porcentaje de predicciones correctas (entre más alto mejor)
- **Error de Clasificación:** Porcentaje de predicciones incorrectas (entre más pequeño mejor)

## ■ Matriz de Confusión

- Mejor entendimiento de cómo se desempeña el clasificador.
- Permite calcular sensibilidad,, precisión, especificidad , score f1, que podrían coincidir mejor con el objetivo del negocio que la medida de exactitud

## • Curvas ROC y Área Bajo la Curva (Area Under the Curve AUC)

- Visualizar el desempeño del clasificador en todos los umbrales de clasificación posibles, lo que ayuda a elegir un umbral que equilibre adecuadamente la sensibilidad y la especificidad
- Sigue siendo útil cuando hay alto desbalance en las clases (a diferencia de la exactitud de clasificación / error)
- Más difícil de usar cuando hay más de dos clases de respuesta

# Medidas de clasificación

## Exactitud (Accuracy)

- Provee una medida de la eficiencia general del modelo.
- Es la **métrica de clasificación más sencilla**.

*Exactitud(accuracy) =  $\frac{\text{Número de clasificaciones correctas}}{\text{Número total de casos de prueba}}$*

- No indica el tipo de error en la clasificación (positivos como negativos o negativos como positivos).

- **La exactitud no es adecuada en algunas aplicaciones**, principalmente en la clasificación de **datos altamente desbalanceados**, por ejemplo la detección de fraude.
  - La alta exactitud no significa que se detecte la mayoría de los fraudes





Datos	y	ypred
Balanceados	0	0
3 0s,	1	1
4 1s	0	0
	1	1
	1	0
	0	0
	1	1
	y	ypred
Datos	0	0
Desbalanceados	0	0
5 0s,	0	0
2 1s	0	0
	1	0
	0	0
	1	1

=6/7 =86%

# Matriz de Confusión

- Proporciona una **imagen más completa** de cómo funciona el clasificador.
- Para dos clases, se produce una matriz 2x2

		ACTUAL VALUES	
		POSITIVE	NEGATIVE
PREDICTED VALUES	POSITIVE	TP	FP
	NEGATIVE	FN	TN

		Actual Values	
		1	0
Predicted Values	1	<b>TRUE POSITIVE</b> 	<b>FALSE POSITIVE</b> 
	0	<b>FALSE NEGATIVE</b> 	<b>TRUE NEGATIVE</b> 

Confusion Matrix [Image 3] (Image courtesy: My Photoshopped Collection)

*TP*: the number of correct classifications of the positive examples (**true positive**),

*FN*: the number of incorrect classifications of positive examples (**false negative**),

*FP*: the number of incorrect classifications of negative examples (**false positive**), and

*TN*: the number of correct classifications of negative examples (**true negative**).

# Matriz de Confusión - Ejemplo

		ACTUAL VALUES	
		POSITIVE	NEGATIVE
PREDICTED VALUES	POSITIVE	TP	FP
	NEGATIVE	FN	TN

		ACTUAL VALUES	
		POSITIVE	NEGATIVE
PREDICTED VALUES	POSITIVE	560	60
	NEGATIVE	50	330

TP: 560: Hay 560 que son positivas y se predijeron positivas

FP: 60: Hay 60 que son negativas y se predijeron positivas – Error Tipo I

FN: 50: Hay 50 que son positivas y se predijeron negativas - Error Tipo II

TN: 330: Hay 330 que son negativas y se predijeron negativas

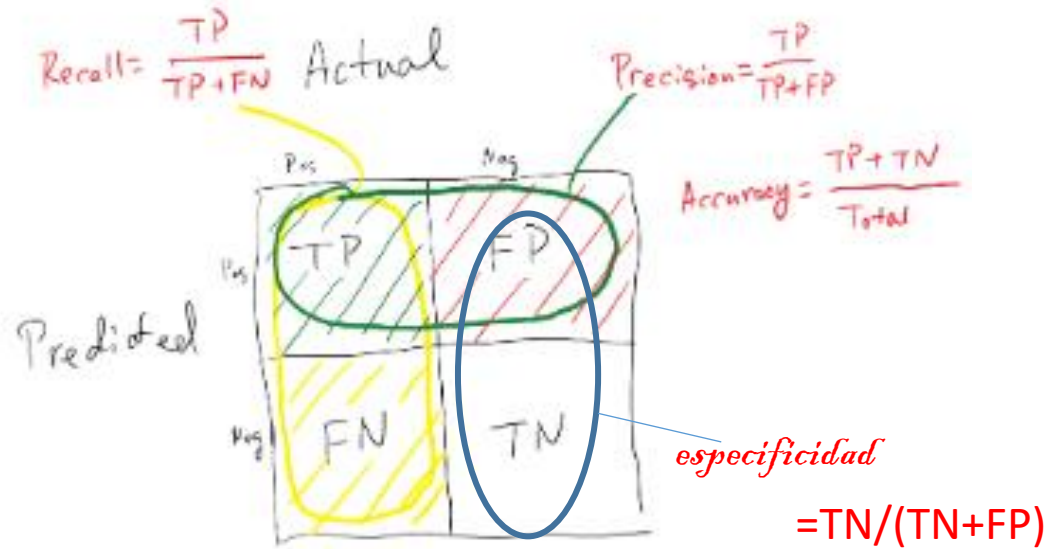
# Métricas a partir de la matriz de confusión

Varias métricas se pueden obtener a partir de la matriz de confusión, tales como:

- Exactitud (Accuracy)** - ¿Cuántos se clasifican correctamente?
- Precisión (Precision)** - ¿Cuántos de los que se predicen positivos son positivos?
- Sensitividad (Recall)** - ¿Cuántos de los que son positivos, se predicen positivos?
- Especificidad** - ¿Cuántos de los que son negativos, se predicen negativos?

Ejemplo:

y	y pred	output for threshold 0.6	Recall	Precision	Accuracy
0	0.5	0	<b>1/2</b>  0.5	<b>2/3</b>  0.67	<b>4/7</b>  0.57
1	0.9	1			
0	0.7	1			
1	0.7	1			
1	0.3	0			
0	0.4	0			
1	0.5	0			



Confusion Matrix [Image 5 and 6] (Image 5 courtesy: My Photoshopped Collection) (Image 6 courtesy: I can not find the source. If you know please comment. I will provide appropriate citations. :D)

# Matriz de confusión – Módulo Sklearn

Original

		ACTUAL VALUES	
		POSITIVE	NEGATIVE
PREDICTED VALUES	POSITIVE	TP	FP
	NEGATIVE	FN	TN

Ejemplo Dataset Diabetes:

0: No tiene

1: Si tiene

**Módulo SKLearn**

n=192	Predicted: 0	Predicted: 1
Actual: 0	TN = 118	FP = 12
Actual: 1	FN = 47	TP = 15

**Verdaderos positivos** (True positives (TP)):

Se predijo correctamente que tenían diabetes: 15

▪ **Verdaderos negativos** (True Negatives (TN)):

Se predijo correctamente que no tenían diabetes: 118

▪ **Falsos positivos** (False Positives (FP)):

Se predijo en forma incorrecta que tenían diabetes cuando en realidad no tenían (**error "Tipo I"**): 12

▪ **Falsos negativos** (False Negatives (FN)):

Se predijo en forma incorrecta que no tenían diabetes cuando en realidad si tenían (**error "Tipo II"**): 47



# Matriz de confusión- Usando librería metrics de sklearn

```
from sklearn import metrics
```

```
confusion = metrics.confusion_matrix(y_test, y_pred)
```

```
print(confusion)
```

```
[[118  12]
 [ 47  15]]
```

Cada valor particular se  
calcula a partir de la matriz:

```
TP = confusion[1, 1]
```

```
TN = confusion[0, 0]
```

```
FP = confusion[0, 1]
```

```
FN = confusion[1, 0]
```

**IMPORTANTE:**

Primer argumento: valores verdaderos, y\_test

Segundo argumento: valores predichos, y\_pred

n=192	Predicted: 0	Predicted: 1	
Actual: 0	TN = 118	FP = 12	130
Actual: 1	FN = 47	TP = 15	62
	165	27	

# Métricas calculadas a partir de la matriz de confusión: Exactitud de clasificación

# usar float para realizar la división, no división entera

Dos formas:

1) `print((TP + TN) / float(TP + TN + FP + FN))`

2) `print(accuracy_score(y_test, y_pred))`

Ejemplo:  $(118+15)/118+15+12+47)$

Exactitud =  $133/192$

0.692708333333

0.692708333333

	n=192		
	Predicted: 0	Predicted: 1	
Actual: 0	TN = 118	FP = 12	130
Actual: 1	FN = 47	TP = 15	62
	165	27	

# Métricas calculadas a partir de la matriz de Confusión: Error de clasificación

**Error de clasificación:** ¿Qué tan seguido el clasificador se equivoca?  
También se le conoce como "Misclassification Rate"

Python:

Dos formas:

- 1) `classification_error = (FP + FN) / float(TP + TN + FP + FN)`  
`print(classification_error)`
- 2) `print(1 - metrics.accuracy_score(y_test, y_pred))`

Ejemplo:

```
classif_error = (12+47)/118+15+12+47  
classif_error = 59/192
```

n=192	Predicted: 0	Predicted: 1	
Actual: 0	TN = 118	FP = 12	130
Actual: 1	FN = 47	TP = 15	62
	165	27	

# Métricas calculadas a partir de la matriz de confusión: Precisión

- Cuando se predice un valor positivo, ¿qué tan frecuentemente la precisión es correcta?
- ¿Qué tan "preciso" es el clasificador para predecir instancias positivas?
- **Precision  $p$**  es el número de **ejemplos positivos clasificados correctamente** divididos entre el número total de ejemplos clasificados como positivos.

$$p = \frac{TP}{TP + FP}$$

Python: Dos formas:

- 1) `precision = TP / float(TP + FP)`  
`print(precision)`  
0.5555555555555556
- 2) `print(metrics.precision_score(y_test, y_pred))`  
0.5555555555555556

$$= 15 / (15 + 12) \\ = 15 / 27$$

n=192	Predicted: 0	Predicted: 1	
	Actual: 0	Actual: 1	
	TN = 118	FP = 12	130
	FN = 47	TP = 15	62
	165	27	

# Métricas calculadas a partir de la matriz de confusión: Sensibilidad - recall

- De los reales positivos ¿cuántos se predicen correctamente?
- ¿Que tan "sensible" es el clasificador para detectar las instancias positivas?
- También se le conoce como “**Razón de Verdaderos Positivos**” TPR(True positive rate) o "Recall"

- Python. Dos formas:

1) Sensitividad =  $TP / \text{float}(TP+FN)$

```
print(sensitivity)
```

0.241935483871

$r = 15 / (15 + 47)$

$r = 15 / 62$

2) `print(metrics.recall_score(y_test, y_pred))`

0.241935483871

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

n=192		Predicted: 0	Predicted: 1	
Actual: 0		TN = 118	FP = 12	130
Actual: 1		FN = 47	TP = 15	62
		165	27	

# Métricas calculadas a partir de la matriz de confusión: Especificidad

- Especificidad: De los reales negativos, ¿cuántos se predicen correctamente?
- Que tan “específico” (o “selectivo”) es el clasificador en predecir instancias negativas?
- También se le conoce como **Razón de Verdaderos Negativos**

Especificidad =  $TN / (TN + FP)$

```
print(specificity)  
0.907692307692
```

$esp = 118 / (118 + 12)$   
 $esp = 118 / 130$

	n=192		
	Predicted: 0	Predicted: 1	
Actual: 0	TN = 118	FP = 12	130
Actual: 1	FN = 47	TP = 15	62
	165	27	

# Métricas a partir de la matriz de confusión - Sklearn

n=192		Predicted: 0	Predicted: 1	
Actual: 0	TN = 118	FP = 12	130	<b>Especificidad</b> <b>Razón de Verdaderos Negativos</b> $e = TN / (TN + FP)$
Actual: 1	FN = 47	TP = 15	62	<b>Sensibilidad</b> <b>Razón de Verdaderos Positivos</b> $r = TP / (TP + FN)$
	165	27		<b>Exactitud</b> $a = (TP + TN) / (TP + TN + FN + FP)$

**Precisión:**  
**Razón de Predicciones Positivas Correctas**  
 $p = TP / (TP + FP)$

# Métricas calculadas a partir de la matriz de confusión: Valor $F_1$ (también llamado $F_1$ -score)

- Es difícil comparar dos clasificadores usando dos medidas.

$F_1$  score combina las medidas de precision y recall en una sola medida, de manera que se tenga precision y sensibilidad similar.

$$F_1 = \frac{2pr}{p+r}$$

$F_1$ -score is the harmonic mean of precision and recall.

$$F_1 = \frac{2}{\frac{1}{p} + \frac{1}{r}}$$

- La media armónica de dos números tiene a ser más cercana a la más pequeña de los dos.
- Para que  $F_1$ -value sea grande, ambos  $p$  y  $r$  deben ser grandes.



# Matriz de Confusión

## Ejemplo Clasificando SPAM (Kelleher et al.)

- Supongamos que tenemos un modelo para detectar posibles correos SPAM (la clase positiva), que aplicamos a un conjunto de 20 correos (cuya clasificación conocemos), con los siguientes resultados:

ID	Clase	Pred.	ID	Clase	Pred.
1	spam	ham	11	ham	ham
2	spam	ham	12	spam	ham
3	ham	ham	13	ham	ham
4	spam	spam	14	ham	ham
5	ham	ham	15	ham	ham
6	spam	spam	16	ham	ham
7	ham	ham	17	ham	spam
8	spam	spam	18	spam	spam
9	spam	spam	19	ham	ham
10	spam	spam	20	ham	spam

		Predicted:		
		0	1	
Actual:	0	TN	FP	<b>Especificidad</b> Razón de Verdaderos Negativos $e = TN/(TN+FP)$
	1	FN	TP	<b>Sensibilidad</b> Razón de Verdaderos Positivos $r = TP/(TP+FN)$
				<b>Exactitud</b> $a = (TP+TN)/(TP+TN+FN+FP)$
				<b>Precisión:</b> Razón de Predicciones Positivas Correctas $p = TP/(TP+FP)$
				$F_1 = \frac{2pr}{p+r}$

# Matriz de Confusión

## Ejemplo Clasificando SPAM (Kelleher et al.)

a) Calcule la matriz de confusión:  
TP, FP, TN, FN

b) Calcule la tasa de aciertos  
(exactitud/accuracy)

c) Calcule la Precisión:

¿Qué proporción de los clasificados como SPAM lo son realmente?

d) Calcule la Sensibilidad/Recall

¿Qué proporción de los que son SPAM se clasifican como tal?

e) Calcule la medida F1-score

# ¿En qué métricas debe centrarse?

- La elección de la métrica depende del **objetivo de negocio**
- Identificar qué es más importante reducir, ¿FP o FN?
  - Elegir la métrica con la variable relevante (FP o FN en la ecuación)

- **Ejemplos:**

- 1) **Filtro de spam** (clase positiva es "spam"):

- ¿Qué reducir?

- a) Correo spam se quede en la bandeja de entrada (falsos negativos)
    - b) Correo que no es spam se quite de la bandeja de entrada por el filtro (falsos positivos)

> Reducir FP => usar **precisión (tiene FP como variable)**

$$p = TP/(TP+FP)$$

# ¿Qué métrica usar?

**2) Detector de transacciones fraudulentas** (clase positiva es "fraude"):

¿Qué reducir?

- a) Transacciones fraudulentas no detectadas (falsos negativos)
- b) Transacciones que no son fraude, se indican como fraude (falsos positivos)

➤ Optimizar FN: Usar **sensibilidad (recall)**, tiene FN como variable

$$r = \frac{TP}{TP + FN}.$$

# Scikit Learn - functions

Scoring	Function
<b>Classification</b>	
'accuracy'	<a href="#"><u>metrics.accuracy_score</u></a>
'f1'	<a href="#"><u>metrics.f1_score</u></a>
'precision'	<a href="#"><u>metrics.precision_score</u></a>
'recall'	<a href="#"><u>metrics.recall_score</u></a>
'roc_auc'	<a href="#"><u>metrics.roc_auc_score</u></a>

# Conclusion

- Matriz de confusión da una **imagen más completa** de cómo funciona un clasificador
- También permite calcular varias **métricas de clasificación**, y estas métricas pueden guiar la selección del modelo.

# Referencias

- Ramesh Sharda; Dursun Delen; Efraim Turban. . Chapter 4. Predictive Analysis. In Business, Analytics and Data Science, 4ª. Ed. Pearson, 2017
- <https://www.ritchieng.com/machine-learning-evaluate-classification-model/>
- [https://scikit-learn.org/stable/modules/model\\_evaluation.html](https://scikit-learn.org/stable/modules/model_evaluation.html)
- <https://www.ibm.com/garage/method/practices/reason/evaluate-and-select-machine-learning-algorithm/>
- Evaluación de modelos: <https://www.cs.us.es/cursos/rac-2018/temas/tema-07.pdf>
- Análisis exploratorio de los datos [http://rstudio-pubs-static.s3.amazonaws.com/423338\\_5b4dc6a938144a3b8ab2ce01fe8be14f.html](http://rstudio-pubs-static.s3.amazonaws.com/423338_5b4dc6a938144a3b8ab2ce01fe8be14f.html)