Regresión logística

Regresión logística

- No todas las etiquetas son continuas, a veces es necesario predecir categorías, esto se conoce como clasificación.
- La regresión logística es una de las formas básicas para realizar la clasificación (no se confunda por la palabra "regresión")

Lectura sugerida

Secciones 4-4.3 de

Introduction to Statistical Learning

Por Gareth James

Regresión logística

- Si desea comprender completamente algunos de los conceptos detrás de los métodos de evaluación y las métricas detrás de la clasificación, ¡la lectura es muy recomendable!

Importante

- Queremos aprender sobre Regresión logística como un método para la clasificación.
- Algunos ejemplos de problemas de clasificación:
  - Spam versus correos electrónicos legítimos
  - Préstamo Predeterminado (sí / no)
  - Diagnóstico de la enfermedad
- Todos los anteriores fueron ejemplos de clasificación binaria

Importante

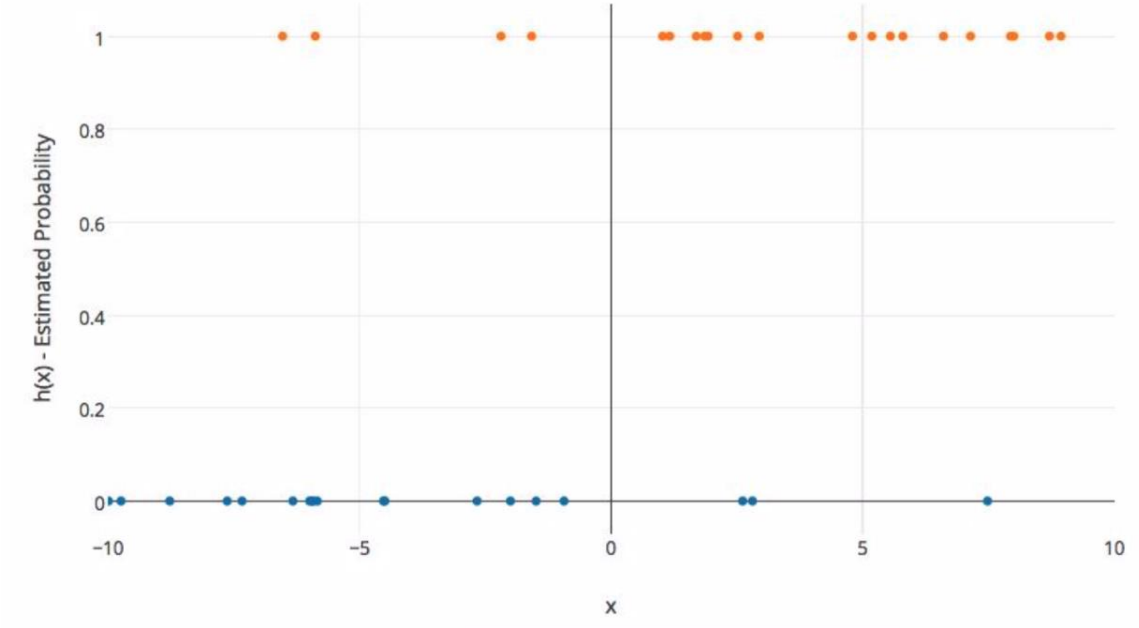
- Hasta ahora solo hemos visto problemas de regresión en los que intentamos predecir un valor continuo.
- Aunque el nombre puede ser confuso al principio, la regresión logística nos permite resolver problemas de clasificación, donde estamos tratando de predecir categorías discretas.

Importante

- La convención para la clasificación binaria es tener dos clases 0 y 1.
- Vayamos a través de la idea básica para la regresión logística.
- También explicaremos por qué tiene el término regresión, ¡aunque se utilice para la clasificación!

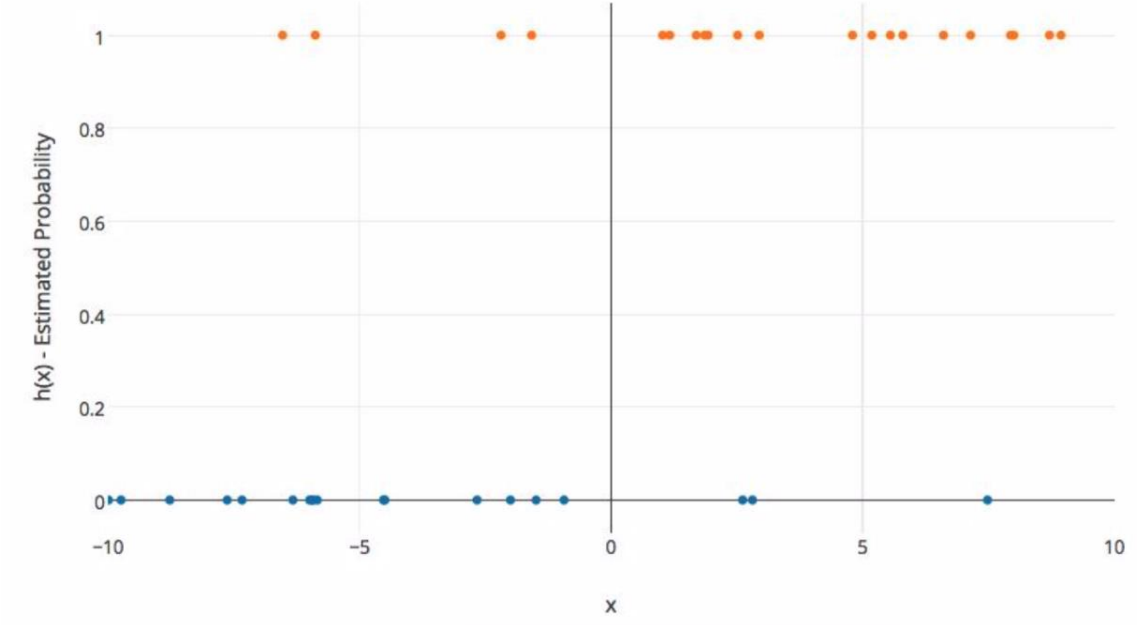
Background

- Imagina que trazamos algunos datos categóricos contra una característica.



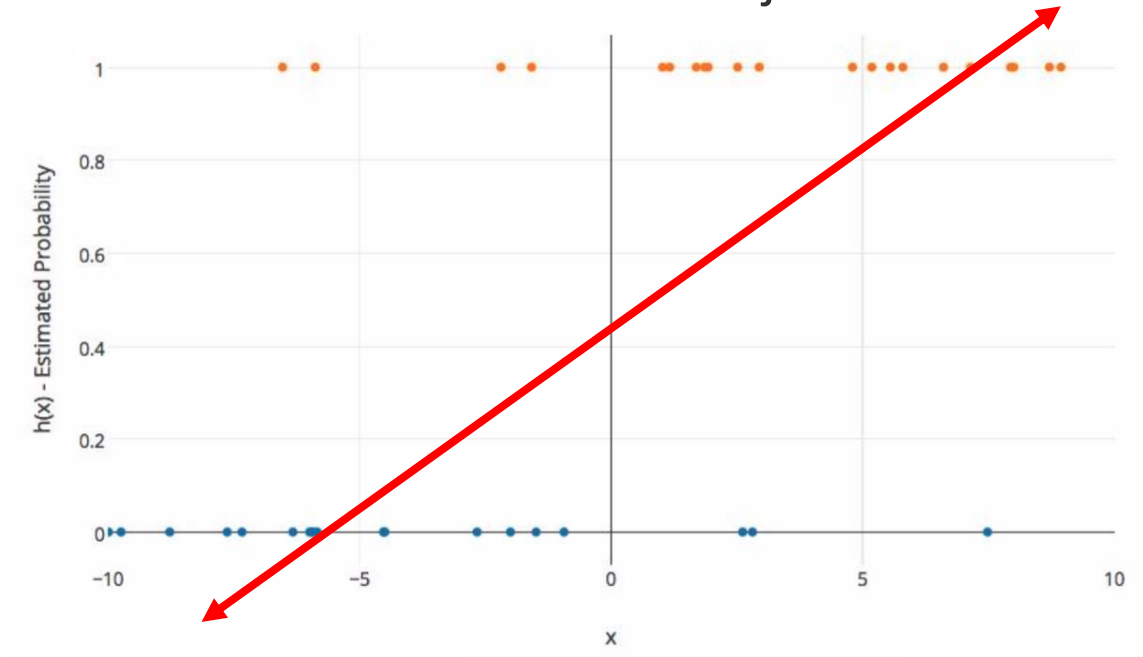
Background

- El eje X representa un valor de característica y el eje Y representa la probabilidad de pertenecer a la clase 1.



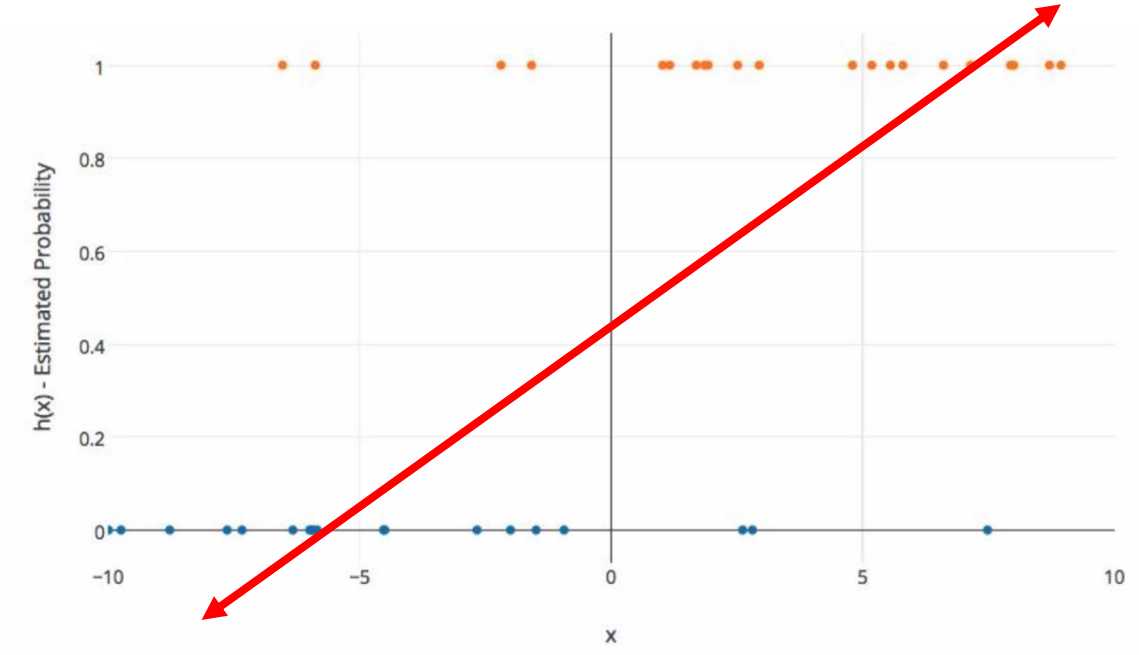
Background

- No podemos usar un modelo de regresión lineal normal en grupos binarios. No conducirá a un buen ajuste:



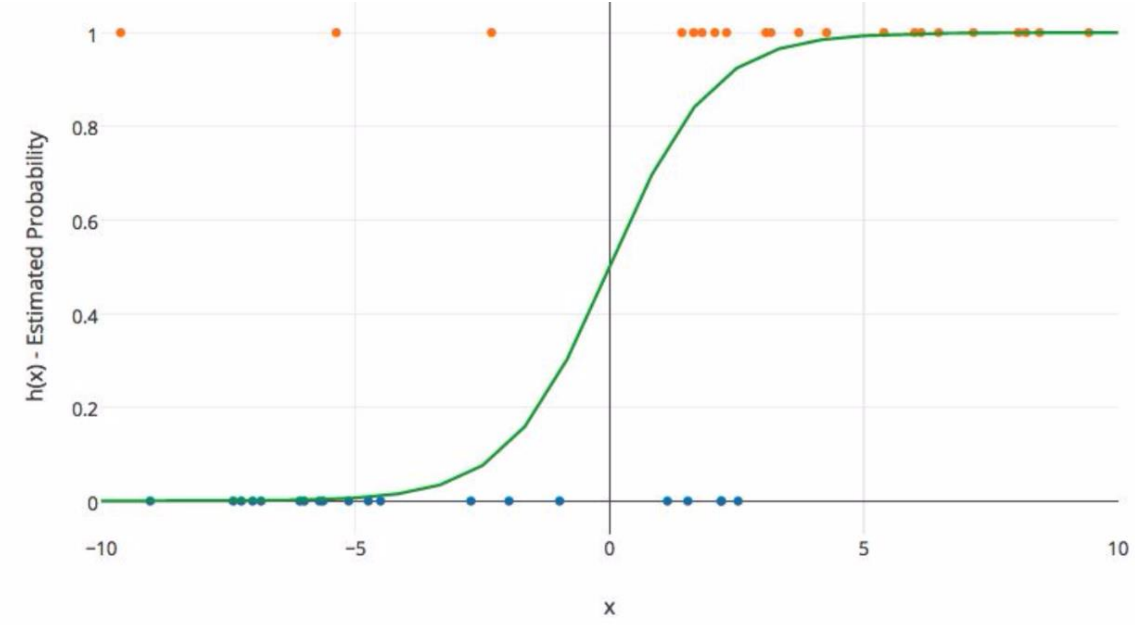
Background

- Necesitamos una función que se ajuste a los datos categóricos binarios!



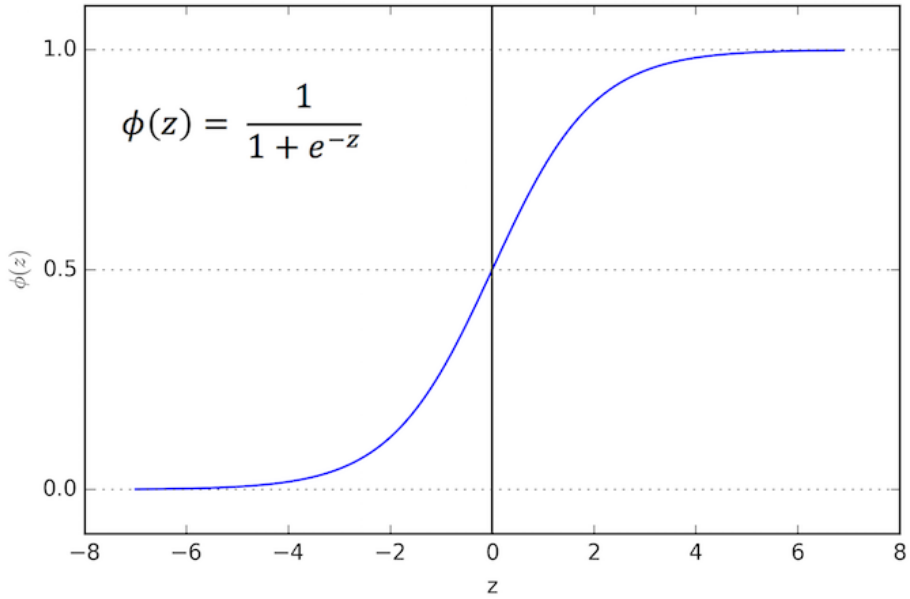
Background

- Sería genial si pudiéramos encontrar una función con este tipo de comportamiento:



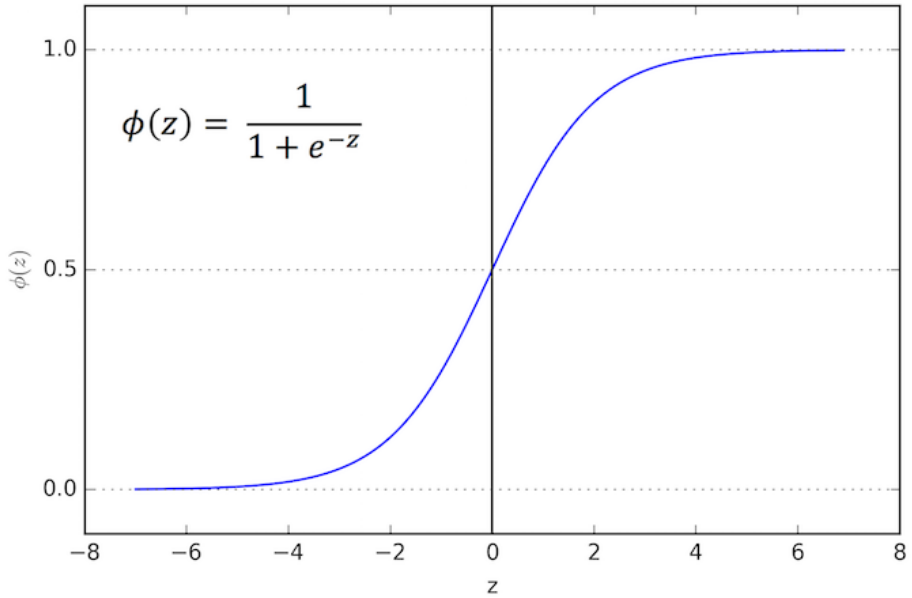
Función sigmoidea

- La función sigmoide (también conocida como logística) toma cualquier valor y genera una salida entre 0 y 1.



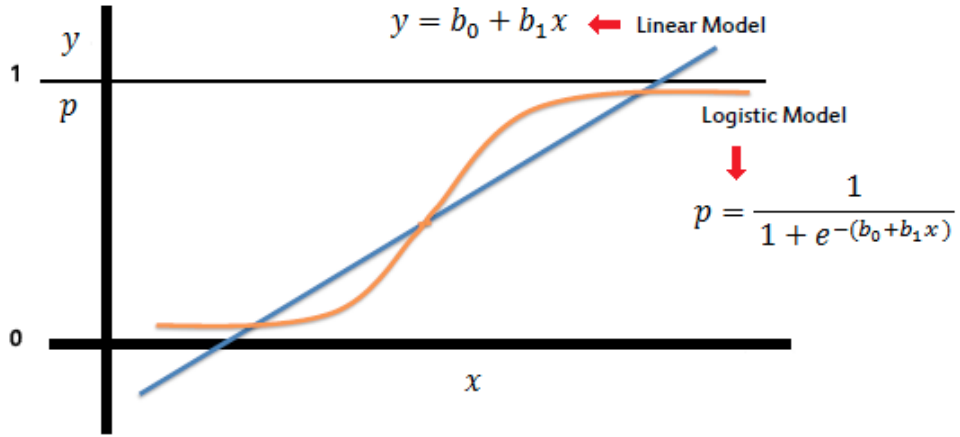
Función sigmoidea

- Esto significa que podemos tomar nuestra Solución de Regresión Lineal y colocarla en la Función Sigmoide.



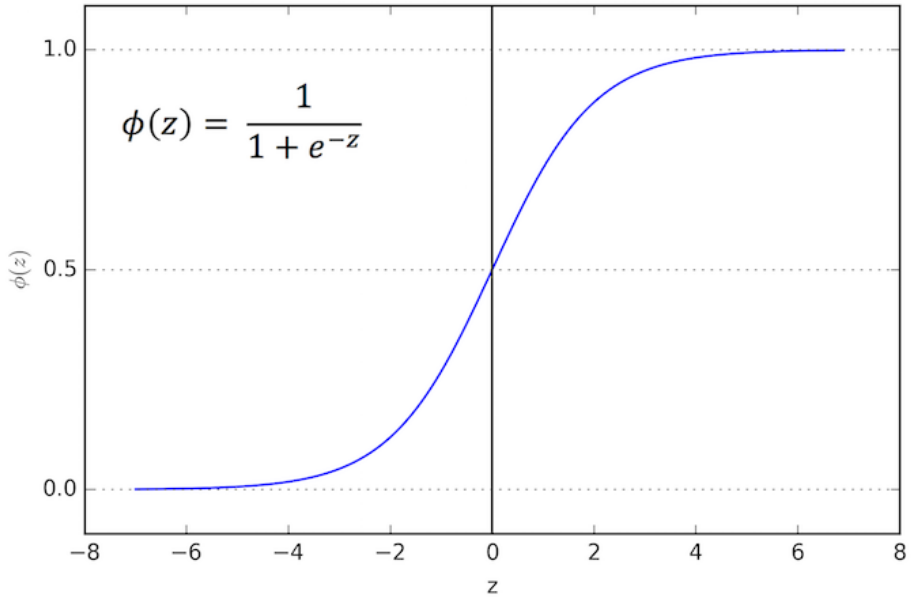
Función sigmoidea

- Esto significa que podemos tomar nuestra Solución de Regresión Lineal y colocarla en la Función Sigmoide.



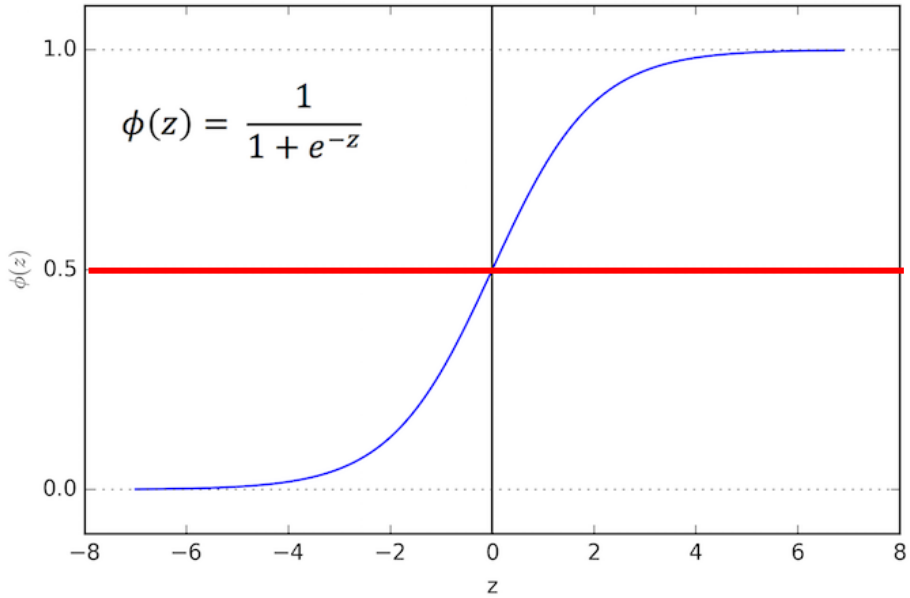
Función sigmoidea

- Esto da como resultado una probabilidad de 0 a 1 de pertenencia a la clase 1.



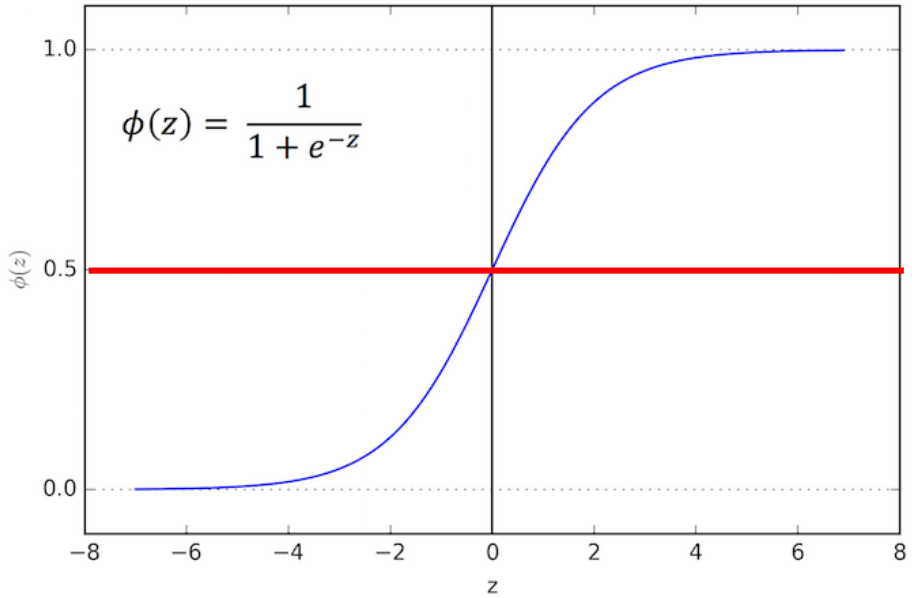
Función sigmoidea

- Podemos establecer un punto de corte en 0.5, cualquier cosa debajo de esto resulta en la clase 0, cualquier cosa arriba es la clase 1.



Repaso

- Usamos la función logística para generar un valor que va de 0 a 1. En función de esta probabilidad, asignamos una clase.



Evaluación del modelo

- Después de entrenar un modelo de regresión logística con algunos datos de entrenamiento, evaluará el rendimiento de su modelo con algunos datos de prueba.
- Puedes usar una matriz de confusión para evaluar los modelos de clasificación.

Matriz de confusión

		predicted condition *(condición predicha)*	
	total population	prediction positive *(predicción positiva)*	prediction negative *(predicción negativa)*
true condition *(condición verdadera)*	condition positive	**Verdadero Positivo** **True Positive (TP)**	**Falso Negativo** **False Negative (FN)** (type II error) **(Error Tipo II)**
	condition negative *(condición negativa)*	**Falso Positivo** **False Positive (FP)** (Type I error) **(Error Tipo I)**	**Verdadero Negativo** **True Negative (TN)**

Matriz de confusión

		predicted condition		
	total population	prediction positive	prediction negative	Prevalence $= \frac{\Sigma \text{ condition positive}}{\Sigma \text{ total population}}$
true condition	condition positive	**True Positive (TP)**	**False Negative (FN)** (type II error)	True Positive Rate (TPR), Sensitivity, Recall, Probability of Detection $= \frac{\Sigma \text{ TP}}{\Sigma \text{ condition positive}}$
	condition negative	**False Positive (FP)** (Type I error)	**True Negative (TN)**	False Positive Rate (FPR), Fall-out, Probability of False Alarm $= \frac{\Sigma \text{ FP}}{\Sigma \text{ condition negative}}$
	Accuracy $= \frac{\Sigma \text{ TP} + \Sigma \text{ TN}}{\Sigma \text{ total population}}$	Positive Predictive Value (PPV), Precision $= \frac{\Sigma \text{ TP}}{\Sigma \text{ prediction positive}}$	False Omission Rate (FOR) $= \frac{\Sigma \text{ FN}}{\Sigma \text{ prediction negative}}$	Positive Likelihood Ratio (LR+) $= \frac{\text{TPR}}{\text{FPR}}$
		False Discovery Rate (FDR) $= \frac{\Sigma \text{ FP}}{\Sigma \text{ prediction positive}}$	Negative Predictive Value (NPV) $= \frac{\Sigma \text{ TN}}{\Sigma \text{ prediction negative}}$	Negative Likelihood Ratio (LR−) $= \frac{\text{FNR}}{\text{TNR}}$

Evaluación del modelo

- El punto principal a recordar con la matriz de confusión y las diversas métricas calculadas es que todas son fundamentalmente formas de comparar los valores predichos con los valores reales.
- ¡Lo que constituye una métrica "buena" dependerá realmente de la situación específica!

Evaluación del modelo

- Podemos utilizar una matriz de confusión para evaluar nuestro modelo.
- Por ejemplo, imagine pruebas para detectar enfermedades.

n=165	**Predicted: NO**	**Predicted: YES**
Actual: NO	50	10
Actual: YES	5	100

Ejemplo: prueba de presencia de enfermedad
NO = prueba negativa = falso = 0
SÍ = prueba positiva = Verdadero = 1

Evaluación del modelo

n=165	**Predicted: NO**	**Predicted: YES**	
Actual: NO	TN = 50	FP = 10	60
Actual: YES	FN = 5	TP = 100	105
	55	110	

Terminología básica:

- Verdaderos positivos (TP)
- Negativos Verdaderos (TN)
- Falsos positivos (FP)
- Falsos negativos (FN)

Matriz de confusión

n=165	**Predicted: NO**	**Predicted: YES**	
Actual: NO	TN = 50	FP = 10	60
Actual: YES	FN = 5	TP = 100	105
	55	110	

Exactitud:

- En general, ¿con qué frecuencia es correcto?
- (TP + TN) / total = 150/165 = 0.91

Matriz de confusión

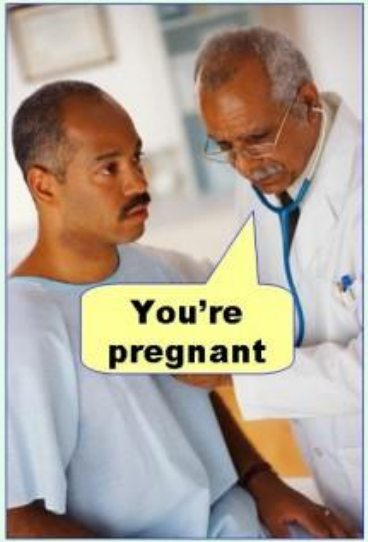
n=165	**Predicted: NO**	**Predicted: YES**	
Actual: NO	TN = 50	FP = 10	60
Actual: YES	FN = 5	TP = 100	105
	55	110	

Tasa de clasificación errónea (Tasa de error):

- En general, ¿con qué frecuencia está mal?
- (FP + FN) / total = 15/165 = 0.09

Matriz de confusión

Type I error
(false positive)



Type II error
(false negative)

Evaluación del modelo

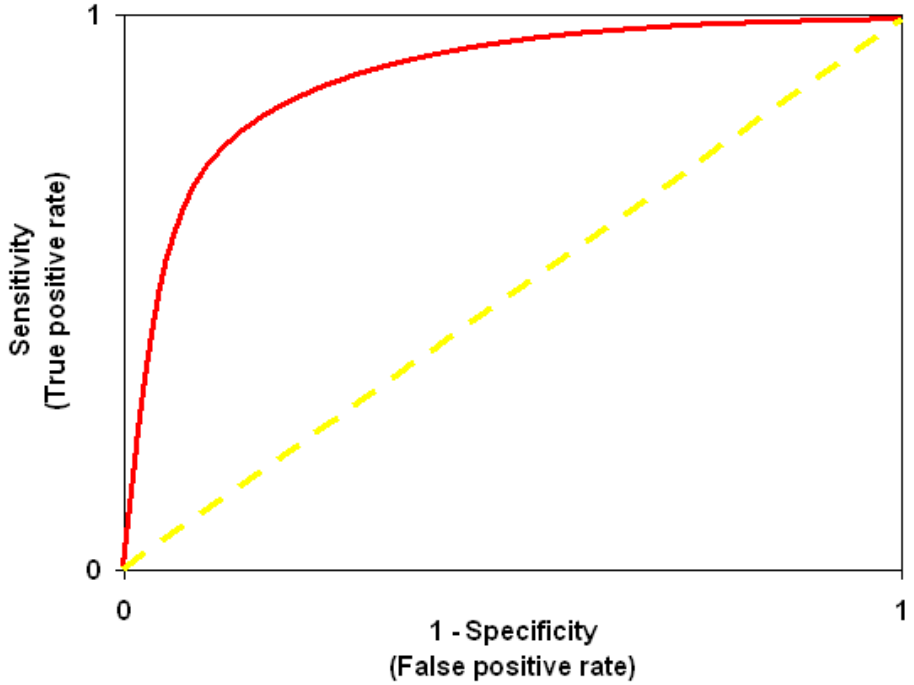
- ¿Todavía confundido con la matriz de confusión?
- ¡No hay problema! Echa un vistazo a la página de Wikipedia para ver si tiene un diagrama realmente bueno con todas las fórmulas para todas las métricas.
- A lo largo del curso, por lo general solo imprimimos métricas (por ejemplo, precisión).

Evaluación del modelo

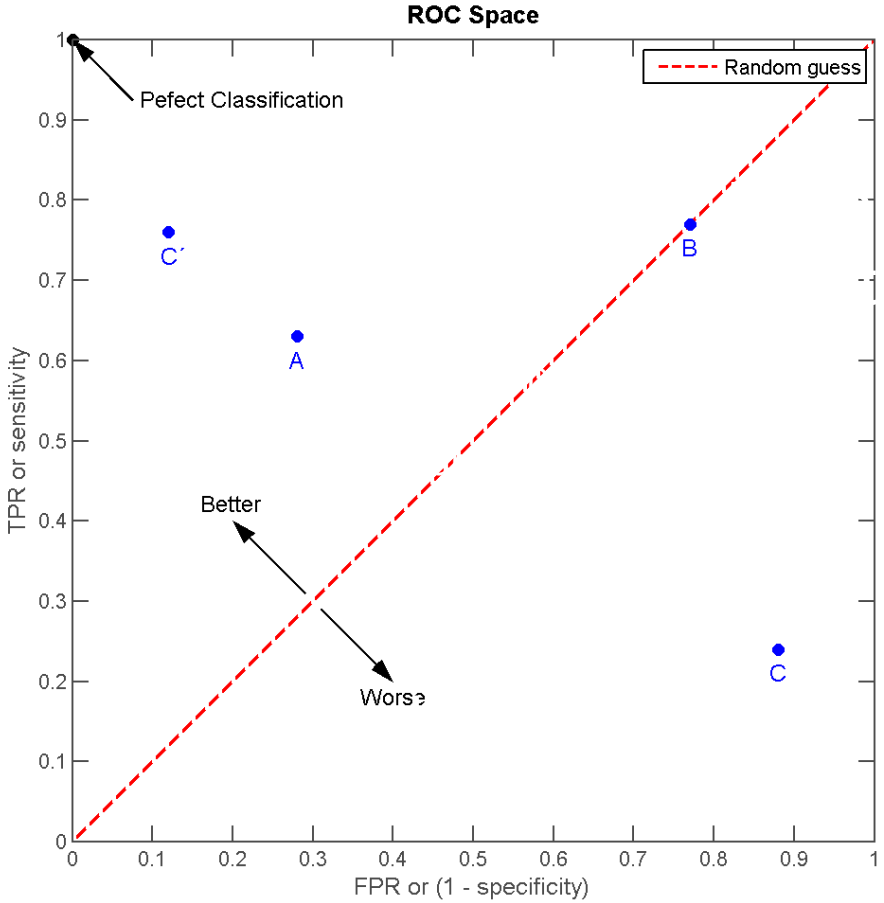
- La clasificación binaria tiene algunas de sus propias métricas de clasificación especial.
- Estos incluyen visualizaciones de métricas de la matriz de confusión.
- La curva de la curva del operador receptor (ROC) se desarrolló durante la Segunda Guerra Mundial para ayudar a analizar los datos del radar.

Evaluación del Modelo

- La curva ROC:



Evaluación del Modelo



Evaluación del Modelo

- Una discusión completa de la curva ROC está más allá del alcance de este curso, pero la lectura sugerida entra en mucho más detalle.
- Por ahora, solo necesita saber que el área debajo de la curva es una métrica de qué tan bien un modelo se ajusta a los datos.

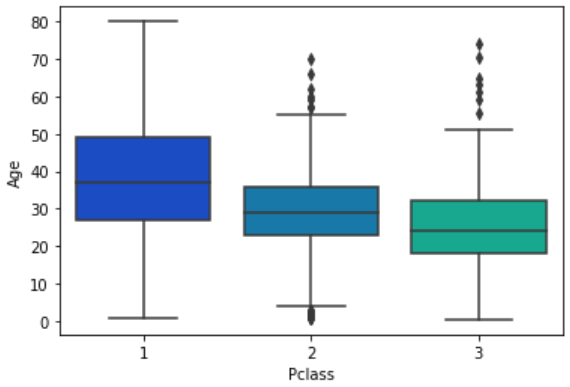
Calculo de los Cuartiles a las edades por clase en el problema del Titanic

In [1]:
```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
%matplotlib inline
```

In [2]:
```
datos = pd.read_csv('titanic_train.csv')
```

In [3]:
```
sns.boxplot(x='Pclass',y='Age',data=datos,palette='winter')
```

Out[3]: <AxesSubplot:xlabel='Pclass', ylabel='Age'>



In [4]:
```
pasaj_c1 = datos[(datos['Pclass']==3) & (datos['Age'].isnull()==False)]
pasaj_c2 = datos[datos['Pclass']==2]
pasaj_c3 = datos[datos['Pclass']==3]
```

In [5]:
```
edades_c1 = pasaj_c1['Age']
edades_c2 = pasaj_c2['Age'].dropna()
edades_c3 = pasaj_c3['Age'].dropna()
```

In [6]:
```
Cuartiles_c1=pd.DataFrame(edades_c1.quantile([0.25,0.50,0.75]))
Cuartiles_c1.columns = ["Edad"]
Cuartiles_c1.index = ["Q1","Q2","Q3"]
Cuartiles_c2=pd.DataFrame(edades_c2.quantile([0.25,0.50,0.75]))
Cuartiles_c2.columns = ["Edad"]
Cuartiles_c2.index = ["Q1","Q2","Q3"]
Cuartiles_c3=pd.DataFrame(edades_c3.quantile([0.25,0.50,0.75]))
Cuartiles_c3.columns = ["Edad"]
Cuartiles_c3.index = ["Q1","Q2","Q3"]
```

In [7]:
```
print("Primera Clase")
print(Cuartiles_c1)
print("Segunda Clase")
print(Cuartiles_c2)
print("Tercera Clase")
print(Cuartiles_c3)
```

```
Primera Clase
    Edad
Q1  18.0
Q2  24.0
Q3  32.0
Segunda Clase
    Edad
Q1  23.0
Q2  29.0
Q3  36.0
Tercera Clase
    Edad
Q1  18.0
Q2  24.0
Q3  32.0
```

In [8]:
```
print("Promedio edad 1ra Clase: ",np.round(edades_c1.mean(),2))
print("Promedio edad 2da Clase: ",np.round(edades_c2.mean(),2))
print("Promedio edad 3ra clase: ",np.round(edades_c3.mean(),2))
```

```
Promedio edad 1ra Clase:  25.14
Promedio edad 2da Clase:  29.88
Promedio edad 3ra clase:  25.14
```

In [9]:
```
psj_por_clase = datos.groupby(['Pclass'])
```

In [11]:
```
psj_por_clase['Age'].quantile([0.25,0.50,0.75])
```

Out[11]:
```
Pclass
1       0.25    27.0
        0.50    37.0
        0.75    49.0
2       0.25    23.0
        0.50    29.0
        0.75    36.0
3       0.25    18.0
        0.50    24.0
        0.75    32.0
Name: Age, dtype: float64
```