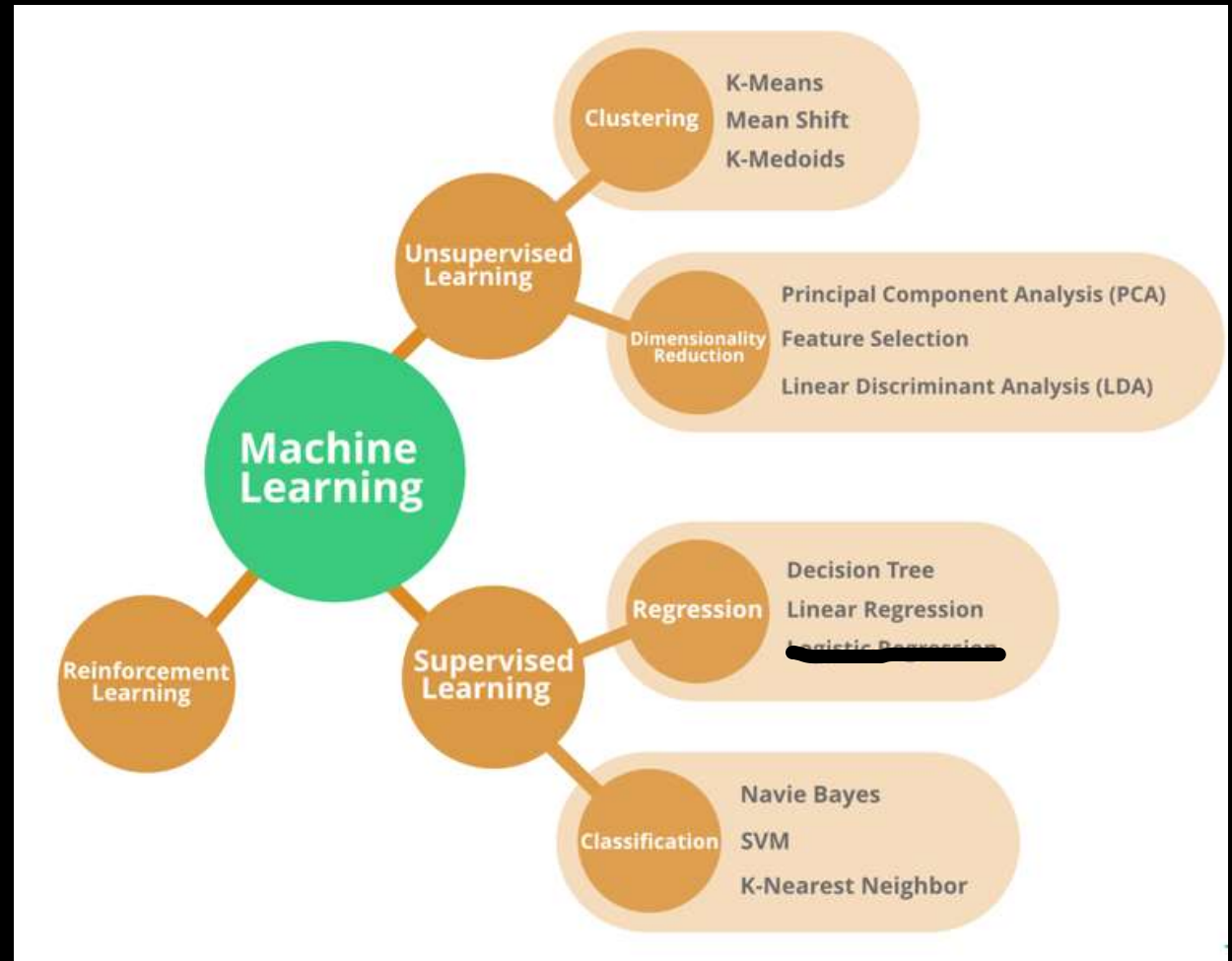


# Machine Learning - Clasificación

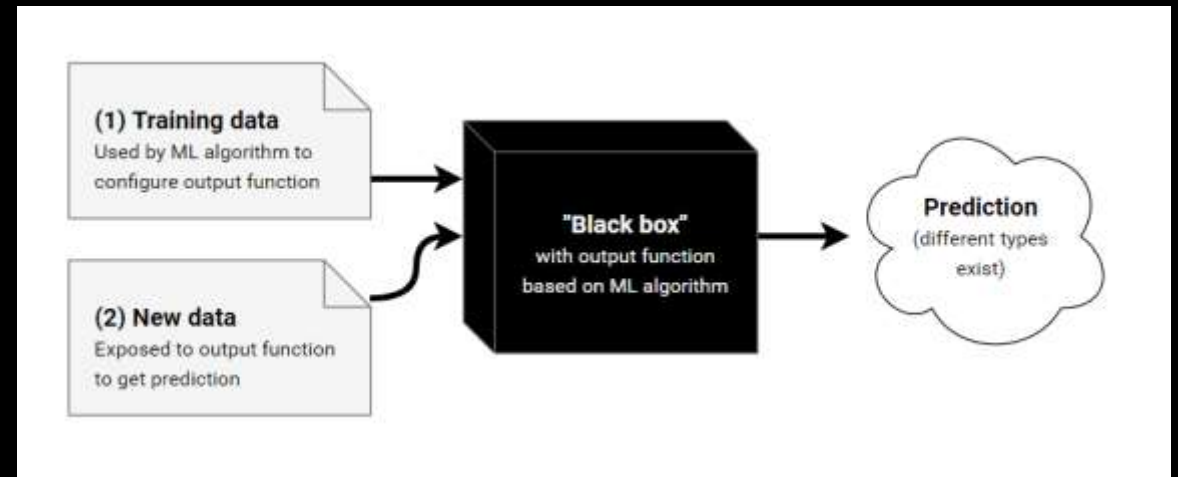
# Algoritmo de clasificación

- Aprendizaje supervisado:
  - Regresión
  - Clasificación
- Aprendizaje no supervisado:
  - Clusterización
  - Reducción de dimensionalidad
- Aprendizaje por refuerzo



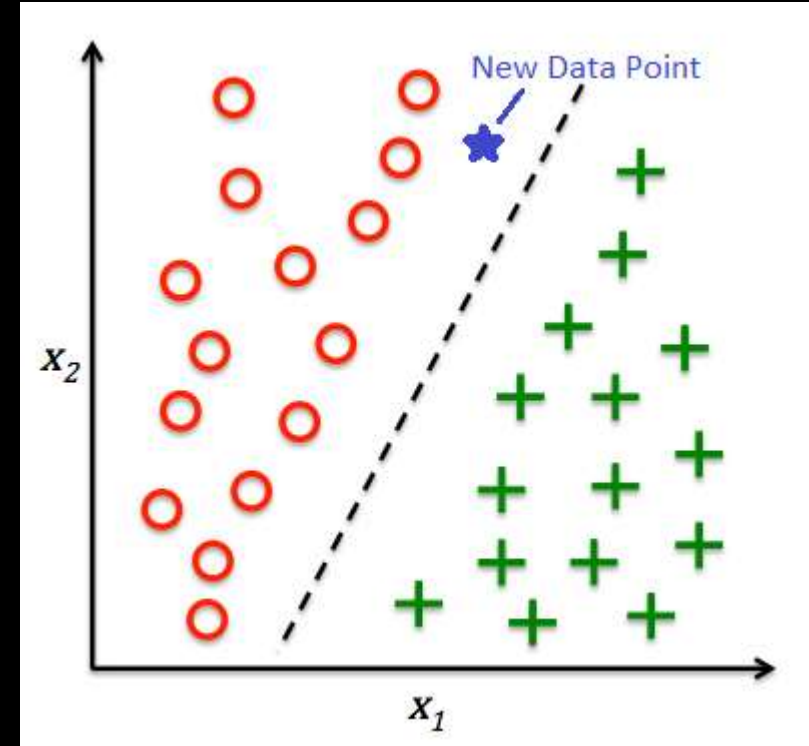
# Modelo de Machine Learning

- Un modelo de ML son un conjunto de parámetros y operaciones que permiten una entrada con datos (input) y, a partir de un proceso específico, obtiene una salida.
- La salida normalmente son uno o más números que representan algo específico para el usuario.
- Se le concede el nombre de “caja negra” habitualmente.



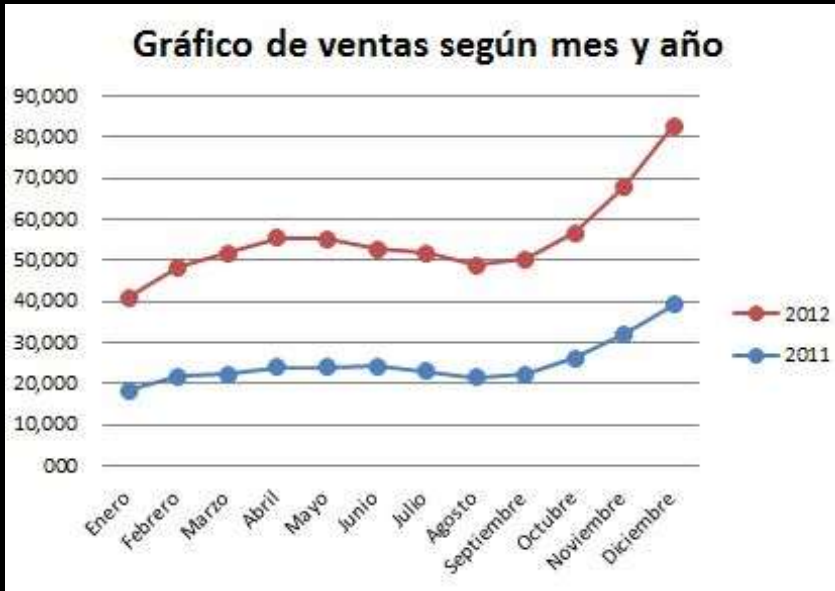
# Algoritmos de clasificación

- Los algoritmos de clasificación son algoritmos de aprendizaje supervisado cuyo objetivo es predecir etiquetas de clase categóricas de las nuevas instancias.
- Dos tipos principales:
  - *Clasificación binaria*: solo hay dos clases posibles. Ejemplo: correo spam o no spam (1 o 0)
  - *Clasificación multi-clase*: más de dos clases. Ejemplo: identificación de dígitos (0 a 9)



# Tipos de técnicas principales en ML

## Algoritmos de regresión



Regresión no lineal

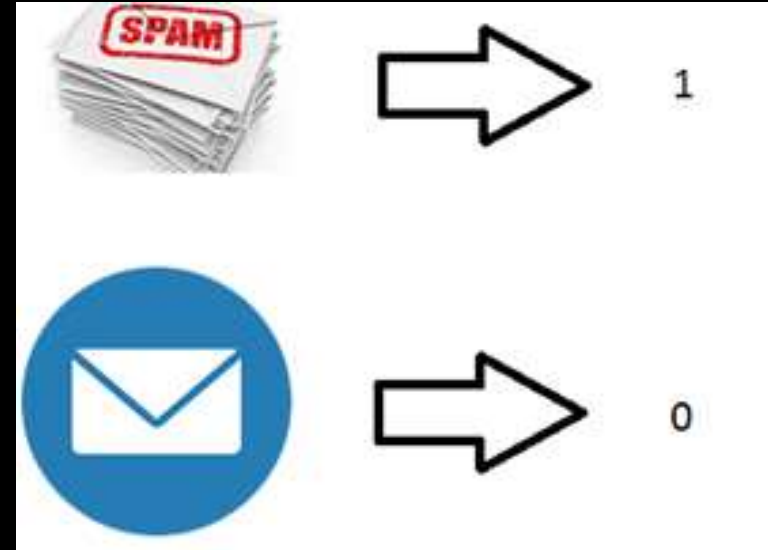
## Algoritmos de clasificación



Árbol de decisión

# Nomenclatura en clasificación con palabras

- Las palabras han de transformarse a números.
- Las clases de los objetos se representan mediante vectores 2D.
- El vector debe contener a 0 todas las clases de objetos disponibles y a 1 el que representa al objeto *en su posición*. No pueden coexistir dos valores 1.
- Ejemplo para correo spam:
  - Definimos que la primera posición del vector 2D se corresponde con el correo SPAM y la segunda se corresponde con el correo no SPAM. Para definir el ejemplo de la imagen, tendríamos el vector  $[1, 0]$ .



# Algoritmos de clasificación más comunes

- Regresión logística
- Árbol de decisión
- KNN
- Naive Bayes
- SVC
- Random Forest
- Deep Learning

# Métricas



# Accuracy

Simplemente cantidad de aciertos vs fallos.

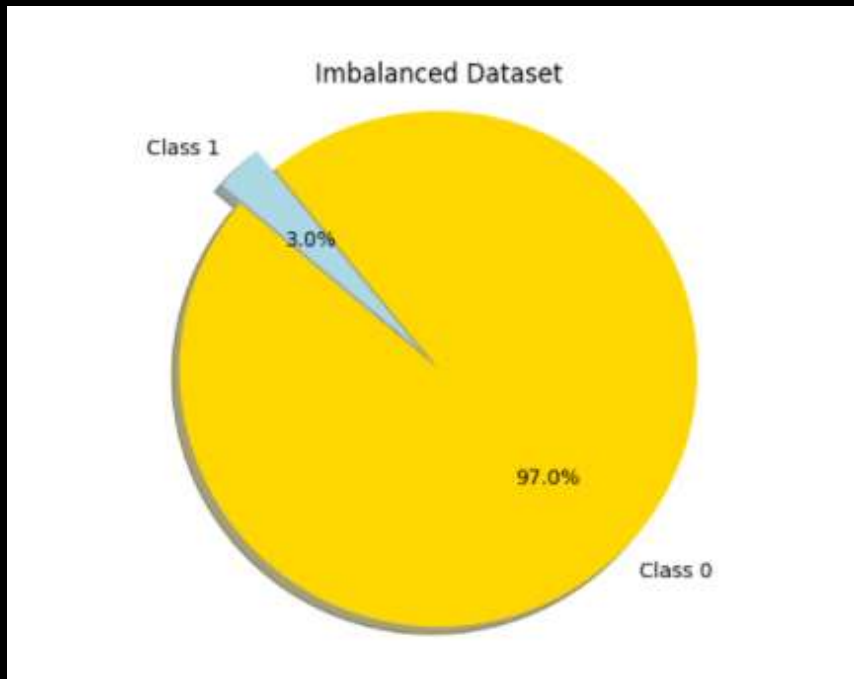
Accuracy = nº aciertos en predicción / total muestras predicción

$$\text{Accuracy} = \frac{\text{correct predictions}}{\text{all predictions}}$$

¿Cómo sé qué clasificador es el mejor? El que tenga un accuracy más alto... Veamos si es así

# La importancia de la métrica

Imagina que tienes clientes de un banco y el objetivo es clasificar si van a pagar o no. El % de los que pagan vs los que no pagan tiene esta pinta.



Calculamos el accuracy: 97% de precisión. Que modelo más bueno!!!

El objetivo del clasificador es que diferencie bien entre las dos clases

## ¿Posibles soluciones?

1. Cambiar la métrica
2. Conseguir más datos :)
3. Resampling: o bien ponemos copias de los elementos de la clase desfavorecida, o eliminamos registros de la más poblada
4. Generar datos sintéticos

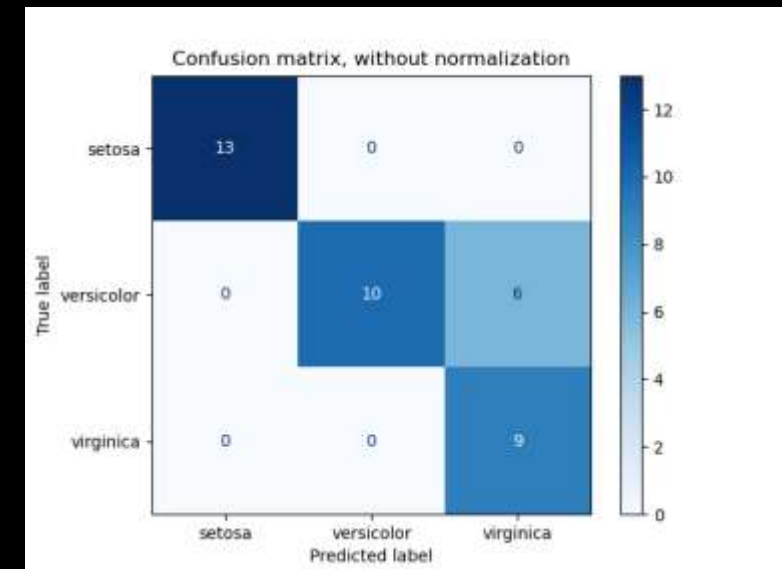
# Matriz de confusión

Muy útil sobre todo en problemas de clasificación binaria. Vemos en una tabla qué tal se comporta el modelo para cada clase (filas son las clases actuales y columnas las predichas). Primero una pequeña aclaración sobre la notación:

- Hay que tener claro qué es 1 y qué es 0. 1 es la pregunta que queremos resolver en el target. ¿Quién me impaga? ¿Quién sobrevive en el Titanic? ¿Quién da positivo en CV?  
0 es si no se da el caso
- Por tanto, positivo es 1, y negativo es 0

Aclarado esto, definimos su matriz de confusión:

	Predicted 0	Predicted 1
Actual 0	TN	FP
Actual 1	FN	TP



Para problemas multiclase

## Accuracy

Los que ha clasificado bien vs todas las muestras a clasificar

$$\text{Accuracy} = \frac{\text{True Positive} + \text{True Negative}}{\text{Total}}$$

	Predicted 0	Predicted 1
Actual 0	TN	FP
Actual 1	FN	TP

## Precision

De los que ha predicho como 1, cuántos en realidad ha acertado

$$\text{Precision} = \frac{\text{True Positive}}{\text{Actual Results}} \quad \text{or} \quad \frac{\text{True Positive}}{\text{True Positive} + \text{False Positive}}$$

	Predicted 0	Predicted 1
Actual 0	TN	FP
Actual 1	FN	TP

## Recall o Sensibilidad

Los positivos que he clasificado bien vs todos los positivos que había

$$\text{Recall} = \frac{\text{True Positive}}{\text{Predicted Results}} \quad \text{or} \quad \frac{\text{True Positive}}{\text{True Positive} + \text{False Negative}}$$

	Predicted 0	Predicted 1
Actual 0	TN	FP
Actual 1	FN	TP

# Algunos ejemplos

## Clasificador de videos buenos para niños



No quieres que se te cuele ningun video malo (0) como video bueno (1) -> FP muy bajos -> precisión alta

Por otro lado, no te va a importar perder algún video bueno (1) y clasificarlo como malo -> FN alto -> mal recall

¿Prioridad? Precision

## Clasificador de ladrones en tienda mediante imágenes



No se me puede escapar ni un ladrón (1), y que se clasifique como no ladrón (0) -> FN bajo -> recall alto

Por otro lado, no me importa clasificar algún cliente como ladrón y realizar registros de vez en cuando -> FP altos -> precisión baja

¿Prioridad? Recall

# F1-Score

Combinación de las métricas de precision y recall

F1 is calculated as follows:

$$F_1 = 2 * \frac{precision * recall}{precision + recall}$$

where:

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

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

In "macro" F1 a separate F1 score is calculated for each `species` value and then averaged.

# Escoger métrica

## Accuracy

Elegir cuando el problema esté balanceado. NO usar nunca cuando la mayor parte de los datos caiga del lado de una sola clase.

Si intentamos predecir cáncer entre 100 personas, y 5 tienen cáncer. Siendo el modelo muy malo, predecirá todos los casos como no cáncer, y tendrá un accuracy del 95%, cuando está prediciendo muy mal en realidad.

## Precision

No me importa que se me escape algún 1, mientras no se me cuele ningún 0 (FP). Que cuando prediga como 1, de verdad sea 1. El foco hay que ponerlo en minimizar los FP

## Recall

Lo que me importa es que los 1s me los capture bien. No me importa que se me cuele algún 0, pero los 1s no se me pueden escapar. como 0s (FN). Por tanto el objetivo es minimizar los FN

# Predicción probabilística

Cuando realizamos predicciones con los modelos, el modelo nos devuelve una probabilidad, no la predicción en sí, y somos nosotros los encargados de interpretar esa probabilidad.

Por ejemplo, si quiero intentar predecir si una persona va a tener o no cierto cáncer, el modelo devuelve una probabilidad entre 0 y 1, y nosotros establecemos un threshold (o umbral) para determinar si es un 0 (no tiene cáncer) o un 1 (tiene cáncer).

	X		Y	
	Fumador	Edad	¿Tendrá cáncer?	
	Si	57	Si 0.7	No 0.3
	No	32	Si 0.1	No 0.9
	Si	39	Si 0.4	No 0.6
	Si	60	Si 0.15	No 0.85

**¿Dónde establecemos el threshold?** Normalmente en 0.5. Si el SI tiene más de 0.5 de posibilidades, lo consideramos como un 1.

Si se desea se puede modificar. Dependerá de la aplicación de negocio.

Si lo pongo por encima de 0.5, estoy siendo más restrictivo con los 1s, entonces tendré más FN (1s clasificados como 0s).

Si lo pongo por debajo de 0.5, seré más flexible con los 1s, y por tanto aumentarán mis FP (0s clasificados como 1s)



# Curva ROC

Curva que nos indica cómo de bueno es nuestro modelo para distinguir las clases.

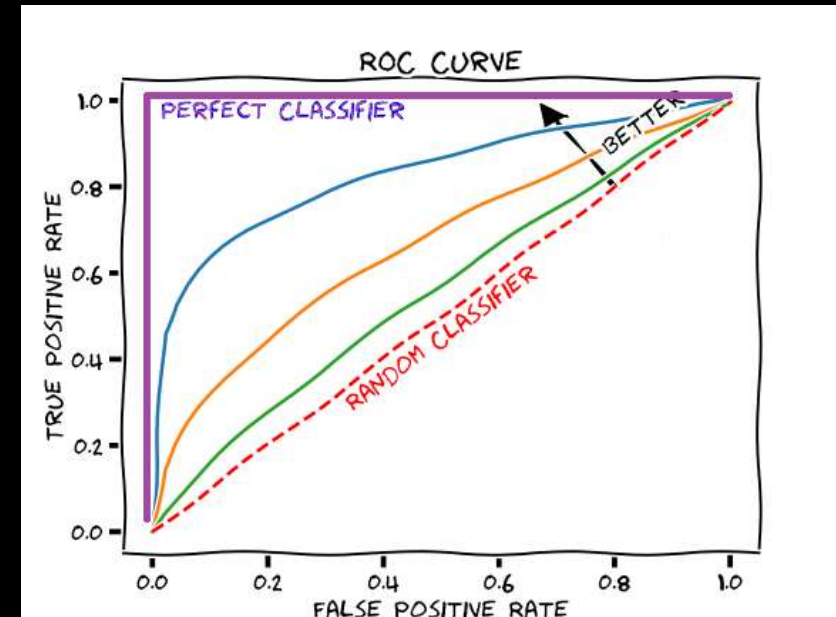
ROC (Receiver Operating Characteristic) es una curva de probabilidad, que va de 0 a 1.

## ¿Qué elementos la componen?

1. Eje X: FPR (False Positive Rate) =  $FP / (FP + TN)$   
Os identificados erróneamente como 1s
1. Eje Y: TPR (True Positive Rate) =  $TP / (TP + FN)$   
O lo que es lo mismo, el Recall -> Los positivos que he clasificado bien vs todos los positivos que había
1. AUC (Area Under the Curve) se trata del área de la curva ROC.  
Va de 0 a 1.

## ¿Cómo se interpreta?

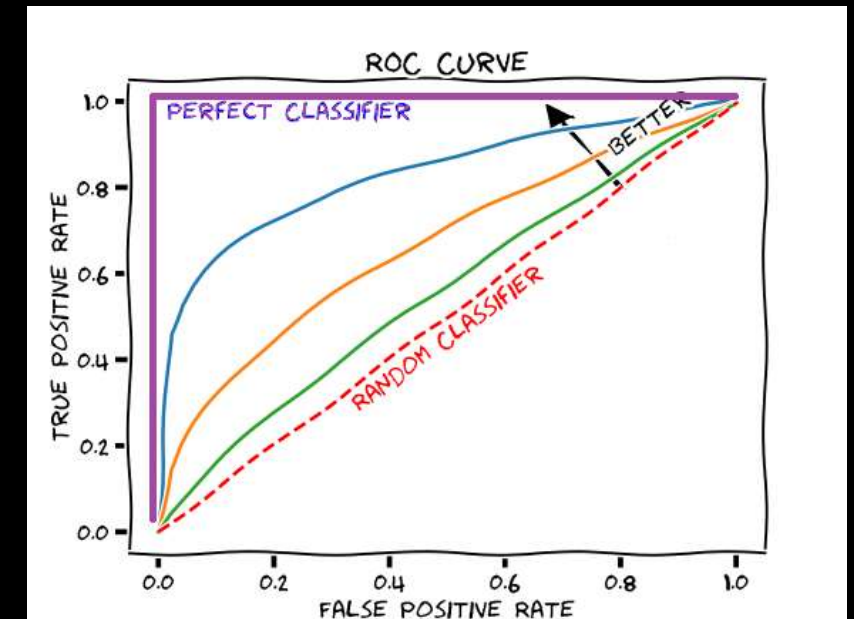
1. **Cuanto mayor es el AUC, más se acerca la curva a la esquina superior izquierda, mejor es el clasificador.**
2. La línea recta del medio representa un clasificador aleatorio. Por tanto, cuanto más cerca de esa línea, peor.
3. Si la curva queda por debajo del random classifier quiere decir que nuestro modelo lo está haciendo peor que un clasificador aleatorio.
4. Si la curva forma un ángulo recto, tienes un clasificador perfecto...sospecha si has hecho algo mal.



# Entendiendo la ROC Curve

- Como sabes, los modelos devuelven **probabilidades en sus predicciones**. Con un threshold (por defecto es 0.5), escogemos entre una clase u otra.
- Si modifico el threshold, cambiarán mis predicciones, y por tanto mi matriz de confusión.
- El threshold es una probabilidad, por lo que podré variarlo de 0 a 1.
- **Cada punto de la curva es cómo quedan mis FPR vs TPR probando varios thresholds.**
- Un punto de la curva (0.10, 0.6), se interpreta como FPR = 0.10, es decir, el 10% me identifica 1s como si fuesen 0s, y el 60% me está identificando bien los 1s.
- **¿Cómo interpreto la zona superior derecha de la curva?** threshold bajo, por lo que soy más flexible con los 1s, se me cuelan más 0s como 1s:
  - FP aumentan
  - TN disminuyen
  - $FPR = FP / (TN + FP)$  -> FPR se aproxima a 1
  - Caso extremo: detecto todo como 1s, por lo que no hay TN. ¿Resultado?  $FPR = 1$
  - TP aumenta. Si dejo entrar a todos los 0s y 1s como 1s, voy a acertar los 1s siempre.
  - FN disminuye
  - $TPR = TP / (TP + FN)$  -> TPR se aproxima a 1

	Predicted	
	0	1
Actual 0	TN	FP
Actual 1	FN	TP



Preguntas