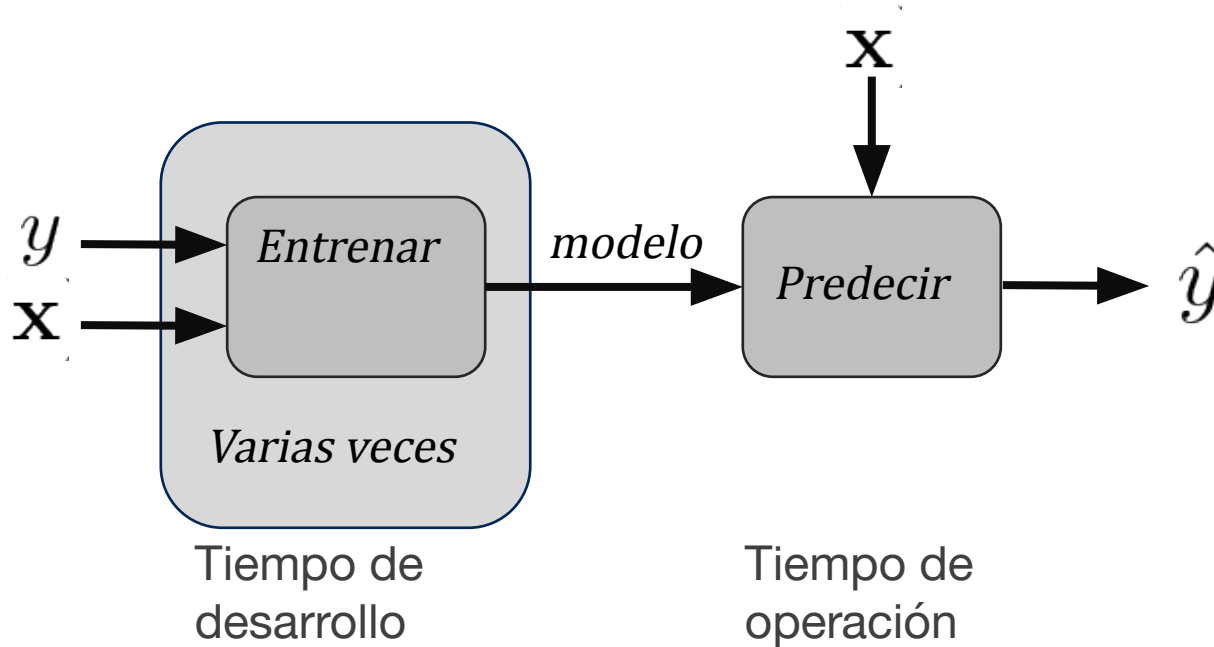


Clasificación Básica

Clasificación

Clasificación *también* es aprendizaje supervisado.



Variable a predecir $y \in \{1, 2, \dots, C\}$

Variables predictores $\mathbf{x} \in \mathbb{R}^d \longrightarrow \langle x_1, x_2, \dots, x_j, \dots, x_d \rangle$

Aplicaciones

Aprobación de créditos



Aplicaciones

Lectura automática de caracteres



Aplicaciones

Reconocimiento de objetos



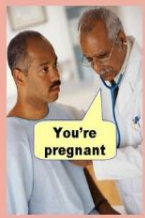

Aplicaciones

Segmentación semántica



Clasificación - Métricas

Métricas para evaluar un problema de clasificación

		CLASE REAL	
		P	N
PREDICCIÓN	P	Correct conclusion	 False positive (type 1 error)
	N	 False negative (type 2 error)	Correct conclusion

Calcula la cantidad de aciertos sobre el total de muestras

$$Accuracy = \frac{CorrectPredictions}{TotalSamples}$$

Clasificación - Binaria

Métricas: matriz de confusión*

		Predicted class	
		P	N
Actual class	P	True positives (TP)	False negatives (FN)
	N	False positives (FP)	True negatives (TN)

Resume de manera gráfica el rendimiento del modelo, permite hacer un diagnóstico más preciso, y definir nuevas métricas.

**Nota: Siempre revisar en qué eje están las predicciones*

Clasificación - Métricas

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

De los que predije positivos,
cuáles en realidad lo eran

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

De los que debía seleccionar como
positivos, cuántos logré clasificar bien

$$F1 = 2 \times \frac{precision \times recall}{precision + recall}$$

Media armónica de las dos métricas

Clasificación - Métricas

Existen muchas métricas más a partir de la matriz de confusión, sin embargo las anteriores son las más comunes.*

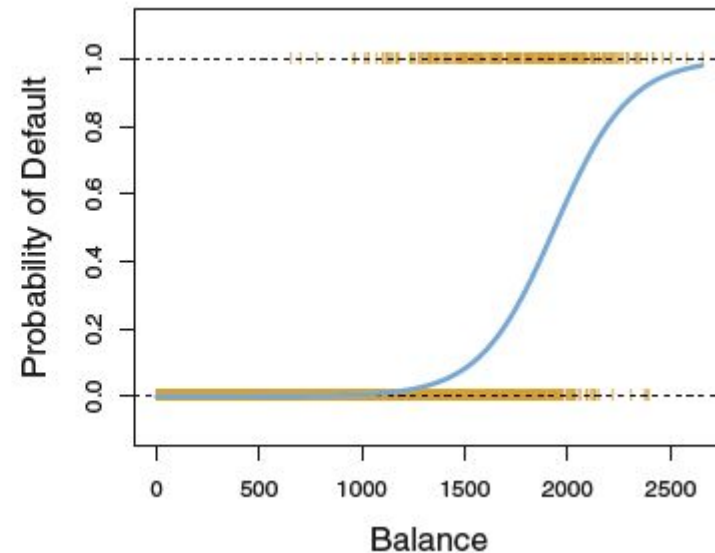
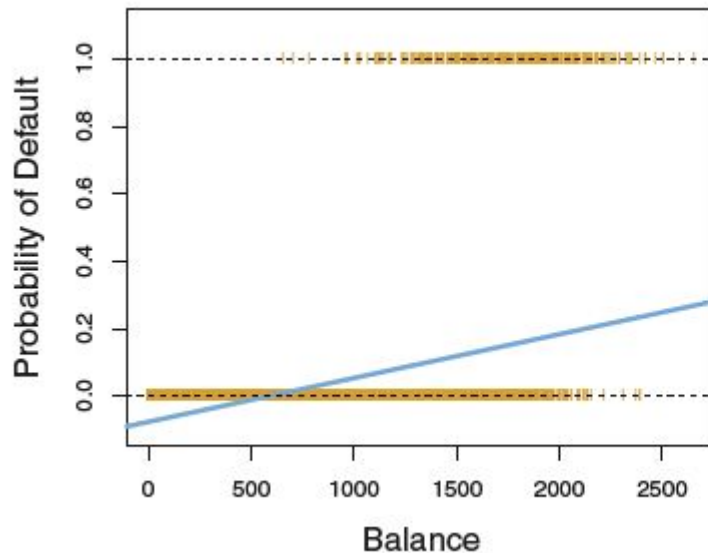
		True condition				
		Total population	Condition positive	Condition negative	Prevalence = $\frac{\Sigma \text{ Condition positive}}{\Sigma \text{ Total population}}$	Accuracy (ACC) = $\frac{\Sigma \text{ True positive} + \Sigma \text{ True negative}}{\Sigma \text{ Total population}}$
Predicted condition	Predicted condition positive	True positive	False positive, Type I error	Positive predictive value (PPV), Precision = $\frac{\Sigma \text{ True positive}}{\Sigma \text{ Predicted condition positive}}$	False discovery rate (FDR) = $\frac{\Sigma \text{ False positive}}{\Sigma \text{ Predicted condition positive}}$	
	Predicted condition negative	False negative, Type II error	True negative	False omission rate (FOR) = $\frac{\Sigma \text{ False negative}}{\Sigma \text{ Predicted condition negative}}$	Negative predictive value (NPV) = $\frac{\Sigma \text{ True negative}}{\Sigma \text{ Predicted condition negative}}$	
		True positive rate (TPR), Recall, Sensitivity, probability of detection, Power $= \frac{\Sigma \text{ True positive}}{\Sigma \text{ Condition positive}}$	False positive rate (FPR), Fall-out, probability of false alarm $= \frac{\Sigma \text{ False positive}}{\Sigma \text{ Condition negative}}$	Positive likelihood ratio (LR+) $= \frac{\text{TPR}}{\text{FPR}}$	Diagnostic odds ratio (DOR) $= \frac{\text{LR+}}{\text{LR-}}$	$F_1 \text{ score} = 2 \cdot \frac{\text{Precision} \cdot \text{Recall}}{\text{Precision} + \text{Recall}}$
		False negative rate (FNR), Miss rate $= \frac{\Sigma \text{ False negative}}{\Sigma \text{ Condition positive}}$	Specificity (SPC), Selectivity, True negative rate (TNR) $= \frac{\Sigma \text{ True negative}}{\Sigma \text{ Condition negative}}$	Negative likelihood ratio (LR-) $= \frac{\text{FNR}}{\text{TNR}}$		

*Se recomienda evaluar cuál métrica tiene más sentido para cada problema

Clasificación

¿Por qué no usar regresión lineal?

- Valores menores que cero o mayores que 1.
- No se puede interpretar como probabilidades.
- Sensible a outliers

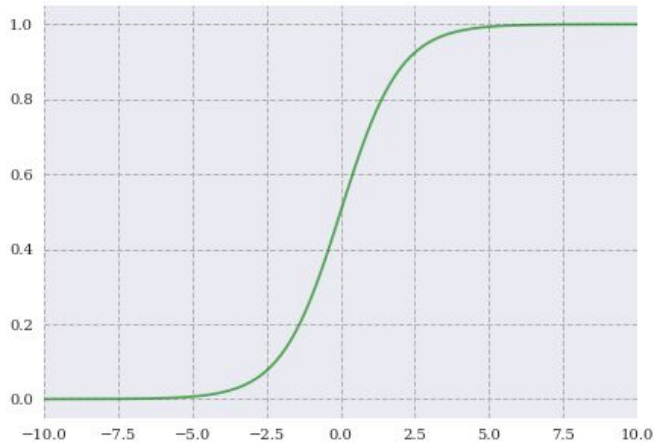


Regresión logística

$$y = \sigma(\mathbf{x}^T \mathbf{w})$$



La relación entre las variables de entrada y parámetros (pesos) del modelo es lineal



$$\sigma(x) = \frac{1}{1 + e^{-x}}$$



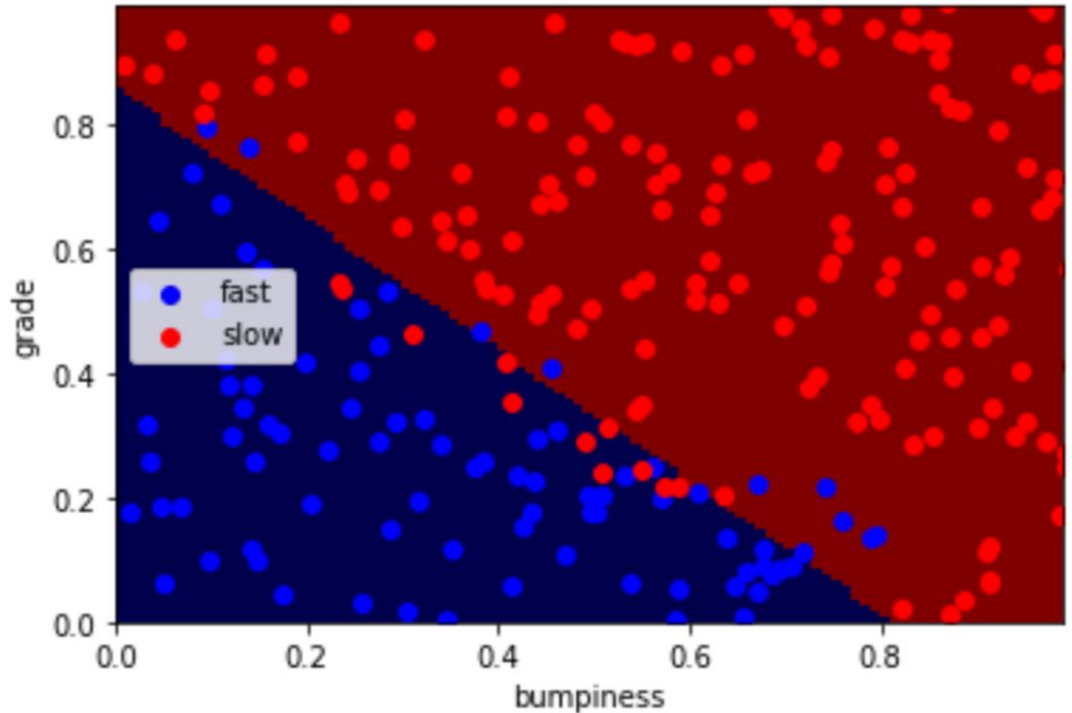
La función *sigmoide* permite *mapear* el resultado de la combinación lineal a probabilidades.

Podemos asignar una etiqueta usando el criterio dado por:

$$\begin{cases} 1 & \text{si } \sigma(\mathbf{x}^T \mathbf{w}) \geq 0.5 \\ 0 & \text{si } \sigma(\mathbf{x}^T \mathbf{w}) < 0.5 \end{cases}$$

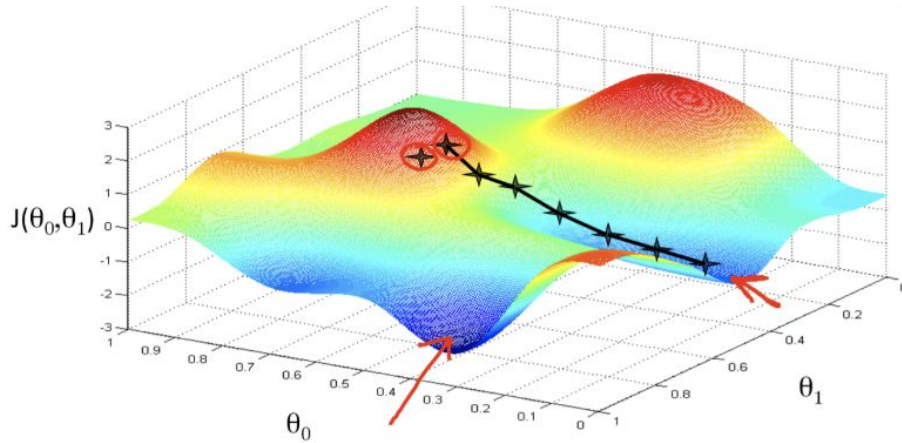
Regresión logística

Una vez escojo el **umbral**, el espacio de clasificación se divide por una línea (plano).



Regresión logística

No tiene solución cerrada como la regresión lineal



Usamos un método iterativo para descender por la superficie de la función de costo hasta llegar al valor más pequeño de esta (el mínimo)

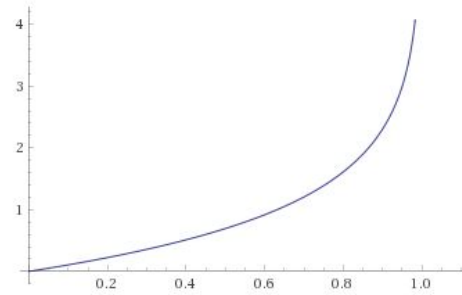
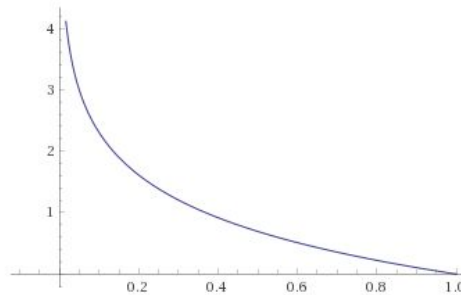
$$\mathbf{w} := \mathbf{w} - \alpha \times \nabla Cost_{\mathbf{w}}$$

Es posible usar como función de error el MSE, pero la función de costo no sería convexa, hay mejores alternativas.

Regresión logística

Para cada elemento del conjunto de entrenamiento, definimos la función de costo

$$\mathcal{L}(y_i, \hat{y}_i) = -(y_i \ln(\hat{y}_i) + (1 - y_i) \ln(1 - \hat{y}_i))$$

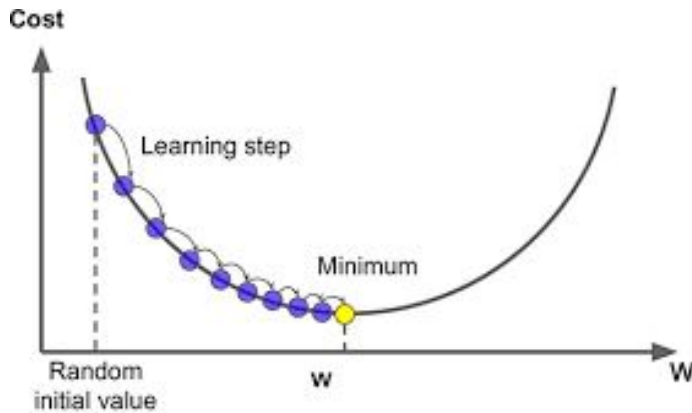


El modelo de regresión logística es aquel que minimiza el costo a lo largo de todos los datos de entrenamiento

$$\mathbf{w} = \arg \min_{\mathbf{w}} -\frac{1}{n} \sum_{i=1}^n y_i \ln(\hat{y}_i) + (1 - y_i) \ln(1 - \hat{y}_i)$$

Regresión logística

Descenso por el gradiente



1. Calculamos la predicción para todos los elementos del conjunto de entrenamiento.*

$$\hat{y} = \sigma(X\mathbf{w}) = \frac{1}{1 + e^{-X\mathbf{w}}}$$

2. Actualizamos los parámetros usando la fórmula de descenso por el gradiente

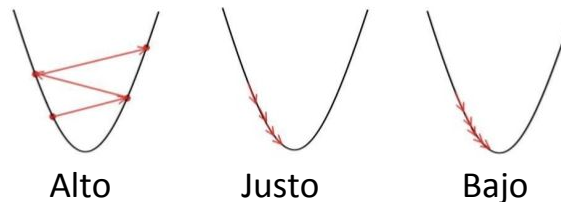
$$\mathbf{w} := \mathbf{w} - \alpha \times \nabla Cost_{\mathbf{w}}$$

donde

$$\nabla Cost_{\mathbf{w}} = \frac{1}{m} X^T (\hat{y} - y)$$

3. Se repite este proceso hasta que se cumpla algún criterio de convergencia, por ejemplo número de iteraciones, diferencia en la norma de gradientes consecutivos, diferencia en norma de parámetros consecutivos, etc.

Note que α no se actualiza en el algoritmo, éste debe ser escogido por el científico. Lo llamaremos un hiper parámetro.



*Note que no hay consenso de escribir las X como filas o columnas, sea cuidadoso al usar una librería

Clasificación - Multiclase

Clasificación de múltiples clases: uno vs el resto



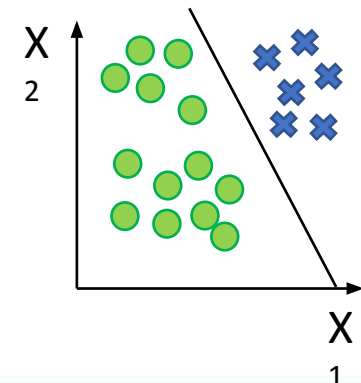
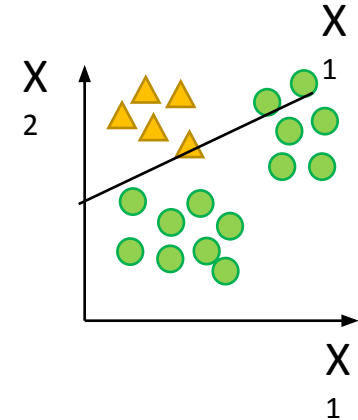
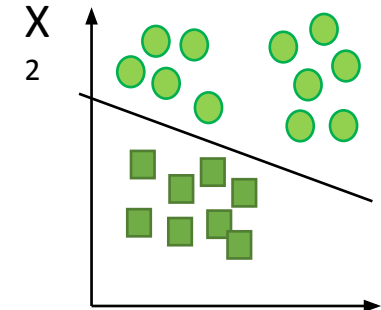
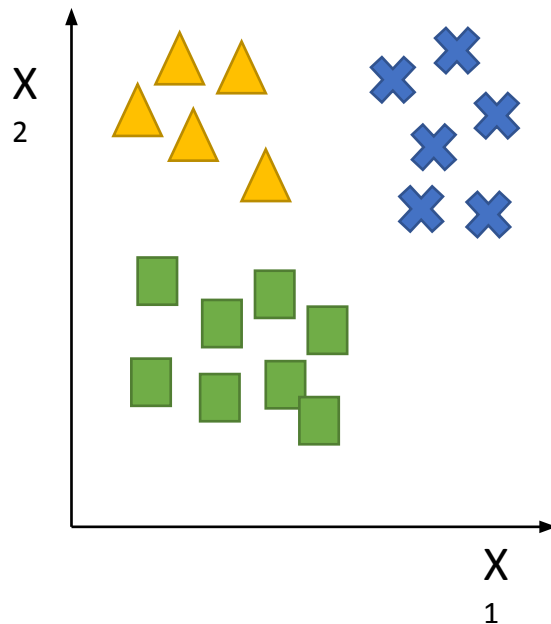
Dividir el problema en k subproblemas de clasificación binaria: es o no es de la clase k

Crear un clasificador \hat{f}_k par cada subproblema: la clase k se define como la clase positiva y la unión de las otras clases se define como la clase negativa

La predicción está basada en el modelo \hat{f}_k con mayor *certeza* para alguna observación x : a la observación se le asigna la clase k que produce mayor *certeza*.

Clasificación - Multiclase

Clasificación de múltiples clases: uno vs el resto



Clasificación - Multiclase

Clasificación de múltiples clases: uno vs uno



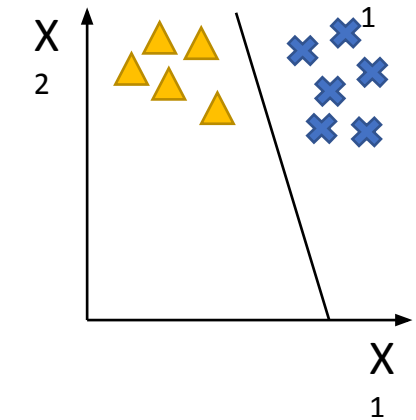
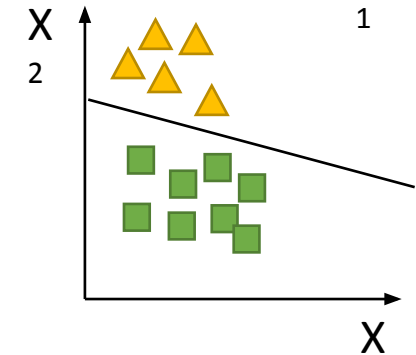
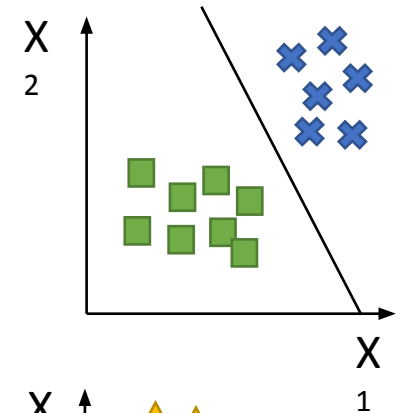
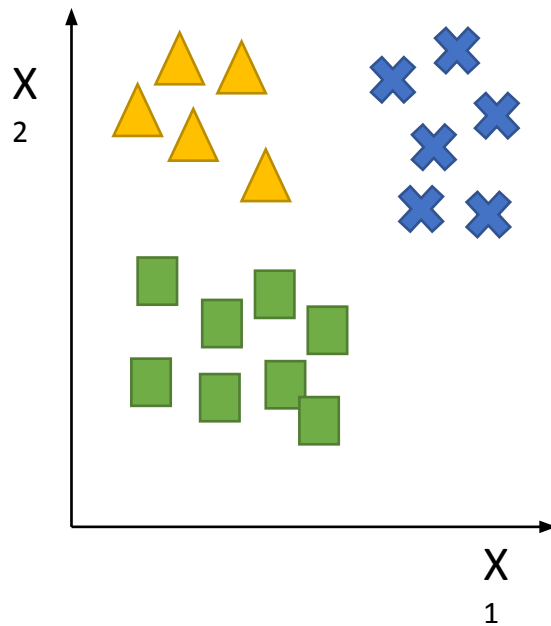
Dividir el problema en $\binom{k}{2}$ subproblemas de clasificación binaria: es de la clase k_i o de la clase $k_j \forall j \neq i$

Crear un clasificador \widehat{f}_{ki} par cada subproblema: la clase k_i se define como la clase positiva y la k_j se define como la clase negativa.

La predicción está basada en cuántos modelos \widehat{f}_{ki} determinan que la observación x , pertenece a su clase: cada clasificador vota si una observación pertenece o no a su clase y al final se cuentan todos los votos

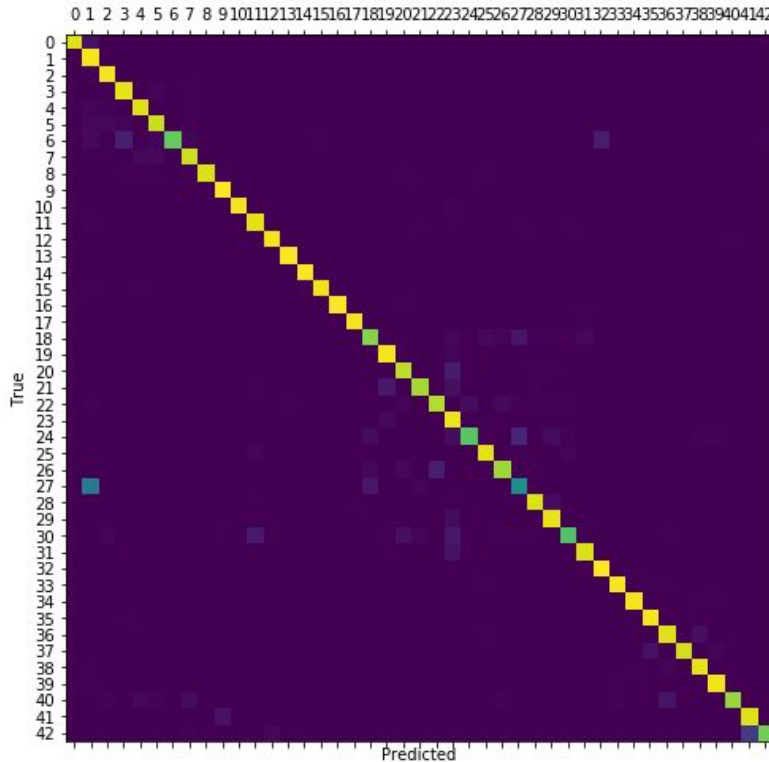
Clasificación

Clasificación de múltiples clases: uno vs uno



Clasificación - Multiclase

¿Qué hacer con la matriz de confusión cuando hay más de dos clases?



- Calcular las métricas por clase y sacar la media simple.
- Calcular las métricas por clase y sacar la media con pesos, i.e. multiplicar por el número de muestras cada métrica y dividir
- Olvidarse de las clases y calcular el accuracy de todo el modelo