



ESCUELA DE INGENIERÍA
FACULTAD DE INGENIERÍA

EDUCACIÓN
PROFESIONAL

Diplomado en Programación y Aplicaciones de Python

Aplicaciones en Ciencia
de Datos e Inteligencia Artificial

Profesor:

Francisco Pérez Galarce





Evaluaciones

Evaluación escrita de conceptos

20%

- 2 controles (contenido teórico e implementación)
- Prueba final de finalización del curso

10%

1 de 2 ok

10%

Desarrollo de tareas de programación

80%

- 2 actividades de implementación en clases
- 2 Mini proyectos
- Repositorio en Github

20%

1 de 2 ok

40%

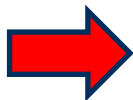
Hoy

20%



Fechas de evaluaciones

Fecha	Actividad/Evaluación
29-10-24	Introducción al aprendizaje de máquina: exploración y procesamiento de datos con Python Actividad 1 (No evaluada)
05-11-24	Aprendizaje supervisado con Python : regresiones Actividad 2 (Evaluada)
12-11-24	Actividad 2 (Evaluada) Control 1
19-11-24	Aprendizaje supervisado con Python naive Bayes y métricas de evaluación Mini Proyecto 1
26-11-24	Aprendizaje supervisado con Python : decision tree, random forest, KNN Actividad 3 (No Evaluada)
03-12-24	Aprendizaje no supervisado con Python: k-means y reducción de dimensionalidad Actividad 4 (Evaluada) – Control 2
10-12-24	Redes Neuronales I Mini Proyecto 2
17-12-24	Redes Neuronales II Prueba Final / Portafolio en Github





Objetivos clase

Identificar el problema de clasificación binaria y múltiple.

Comprender el funcionamiento de los clasificadores naïve Bayes y regresión logística.

Aplicar e implementar clasificadores estudiados con datos reales.



Clasificadores

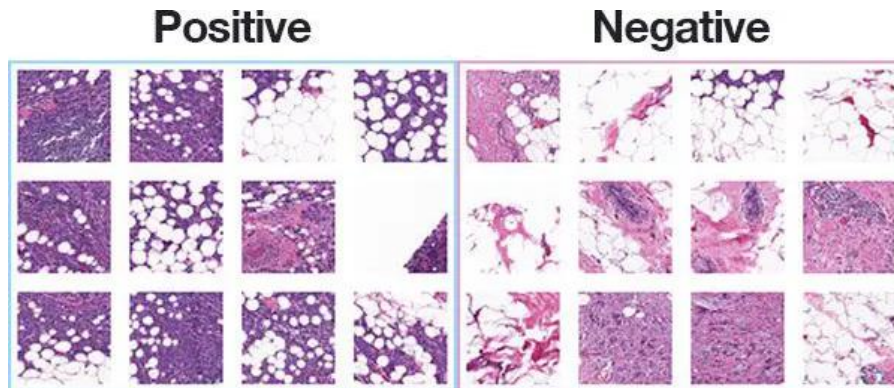
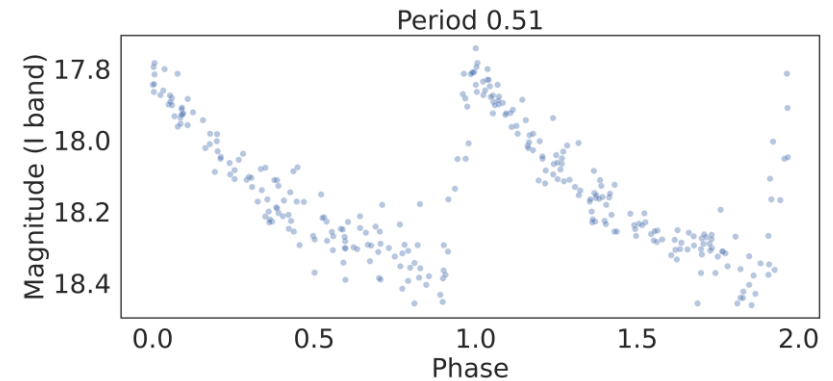
Queremos predecir una **categoría**.

Ejemplos:

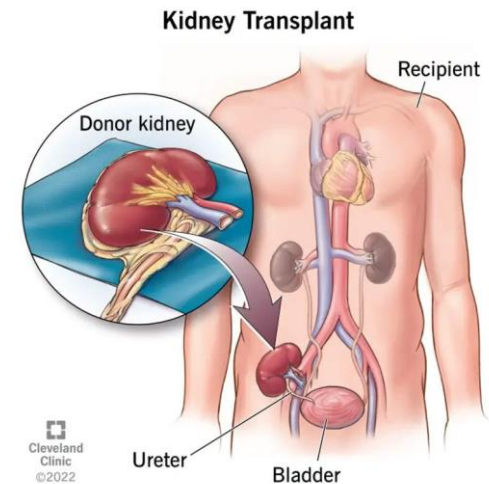
- Tipo de animal (gato, perro)
- Comportamiento de clientes
- Tipos de estrellas o galaxias
- Estados de ánimo desde texto o imágenes



Clasificadores:



<https://pyimagesearch.com/2019/02/18/breast-cancer-classification-with-keras-and-deep-learning/>



Desafío

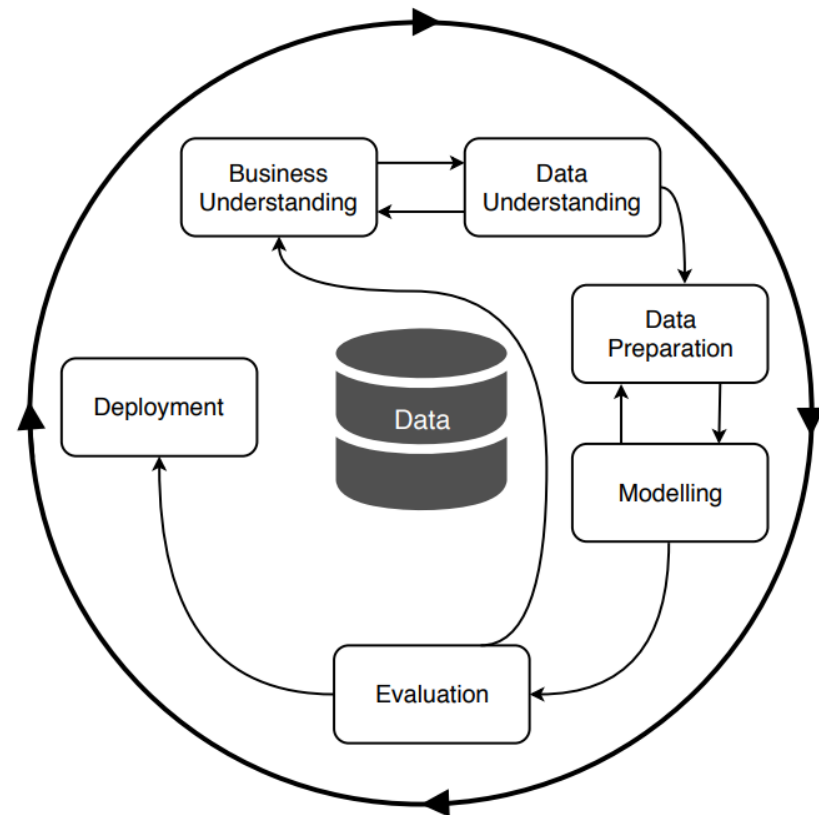


$$\mathcal{D} = \{(x_1, y_1), \dots, (x_N, y_N)\}$$

$$y_i \in \{0, 1\} \forall i \in 1, 2, \dots, N$$

- Falla o no falla
- Compra o no compra
- Paga o no paga

Antes de pasar al
clasificador,
¿Qué procesamiento
debemos aplicar a
los descriptores?



CRoss-Industry Standard Process for Data Mining (CRISP-DM)



Antes de pasar al clasificador, ¿Qué procesamiento debemos aplicar a los descriptores?

Datos categóricos

Datos continuos

Categóricos
ordinales

Categóricos no
ordinales



Procesamiento de descriptores

Datos categóricos

Categóricos
ordinales

Categóricos no
ordinales



Categoricos ordinales

Ejemplos:

Escala de Likert 1 – Totalmente en desacuerdo.
 2 – En desacuerdo.
 3 – Neutral o ni de acuerdo ni en desacuerdo.
 4 – De acuerdo.
 5 – Totalmente de acuerdo.

Nivel de estudios → Educación media
 → Profesional
 → Magíster
 → Doctorado



Categoricos ordinales

Ejemplos:

Escala de Likert

Nivel de estudios → Educación media
→ Profesional
→ Magíster
→ Doctorado

```
from sklearn.preprocessing import MinMaxScaler
Import pandas as pd

df = pd.read_csv('data.csv')

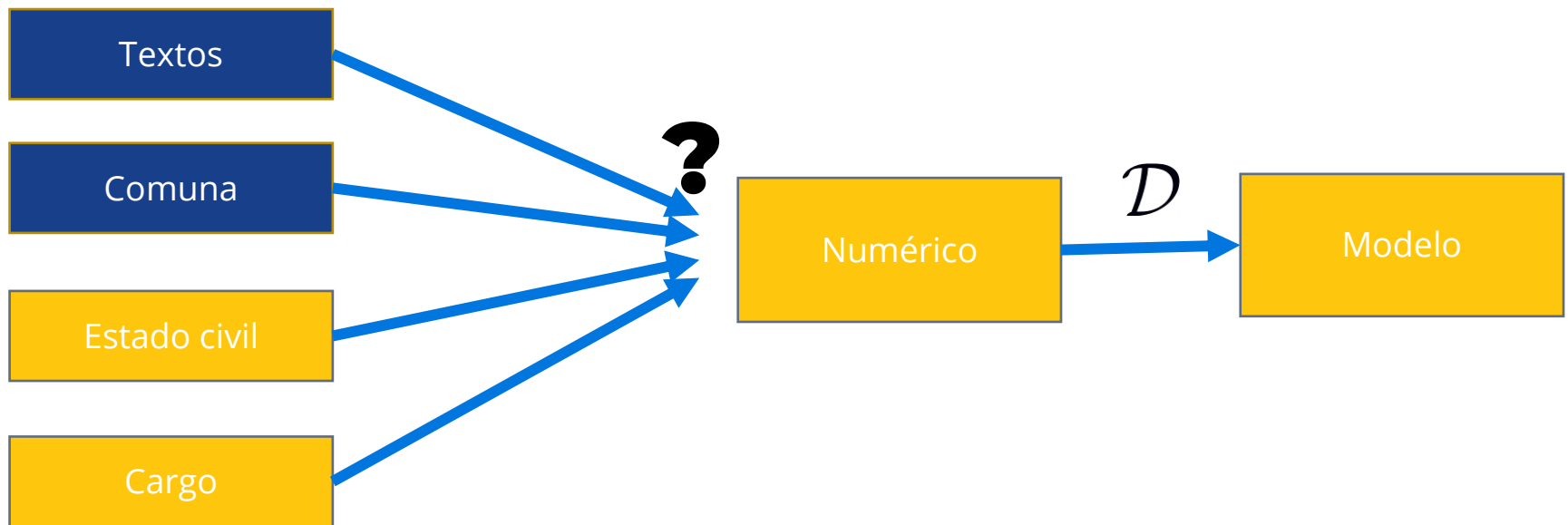
#Mapear
mapping = {'Educación media': 1, 'Profesional': 2, 'Magíster': 3,
'Doctorado': 4}
df['Education_Level_Ordinal'] = df['Education_Level'].map(mapping)

#Escalar
scaler = MinMaxScaler(feature_range=(0, 1))
df['Education_Level_Scaled'] = scaler.fit_transform(df[['Education_Level_Ordinal']])
```

Escalamiento minmax

$$x' = \frac{x - \min}{\max - \min} * (\text{newMax} - \text{newMin}) + \text{newMin}$$

Categóricos no ordinales





Categoricos no ordinales

One Hot Encoder

[0, 0, 0, 1]
[0, 0, 1, 0]
[1, 0, 0, 0]
[0, 1, 0, 0]

Label encoding

[3,
2,
0,
1]



Categoricos no ordinales

One Hot Encoder

```
import pandas as pd

# Sample dataset
data = {'Color': ['Red', 'Blue', 'Green', 'Blue']}
df = pd.DataFrame(data)

# One-Hot Encoding
df_encoded = pd.get_dummies(df, columns=['Color'], prefix='Color')

print(df_encoded)
```

	Color_Blue	Color_Green	Color_Red
0	0	0	1
1	1	0	0
2	0	1	0
3	1	0	0

Categoricos no ordinales

Label encoding

```
from sklearn.preprocessing import LabelEncoder

# Sample dataset
data = {'Color': ['Red', 'Blue', 'Green', 'Blue']}
df = pd.DataFrame(data)

# Label Encoding
encoder = LabelEncoder()
df['Color_Encoded'] = encoder.fit_transform(df['Color'])

print(df)
```

	Color	Color_Encoded
0	Red	2
1	Blue	0
2	Green	1
3	Blue	0

Precaución: no usar en algoritmos que son sensibles a la magnitud de los descriptores, por ejemplo: regresión logística o redes neuronales.

Cuando el número de categorías es muy grande y el modelo no considera la magnitud se puede usar, por ejemplo: árboles de decisión.



Procesamiento de descriptores

Datos continuos



Escalar/estandarizar atributos

Escalamiento MinMax

Normalización z-score

Escalamiento MinMax

1

Realiza una transformación lineal en los datos originales.

2

Si los rangos iniciales son \min y \max , y los rangos finales son newMin y newMax , la transformación pasa de un valor x a un valor x' .

$$x' = \frac{x - \min}{\max - \min} * (\text{newMax} - \text{newMin}) + \text{newMin}$$



Normalización z-score

Los valores para un atributo X son normalizados en base a la media y la desviación estándar.

$$x' = \frac{x - \overbrace{\bar{X}}^{\text{Media}}}{\underbrace{\sigma_X}_{\text{Desviación estándar}}}$$



ESCUELA DE INGENIERÍA
FACULTAD DE INGENIERÍA

EDUCACIÓN
PROFESIONAL

Otros procesamientos en datos continuos

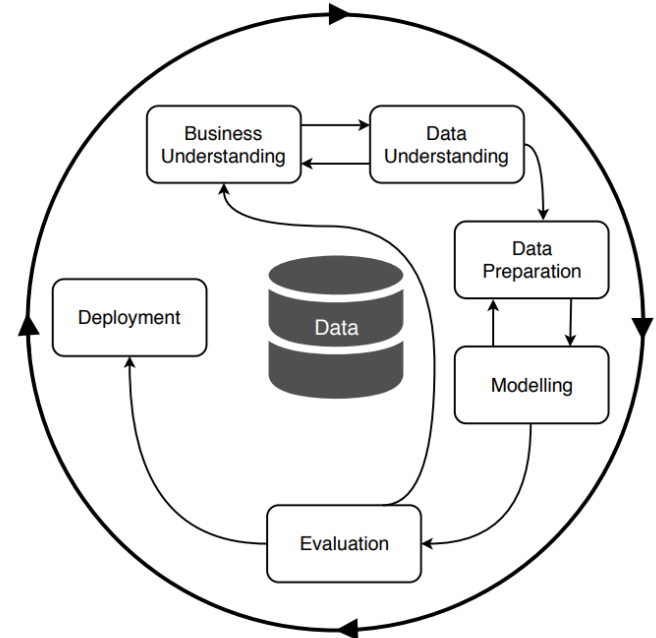
- Discretizar
- Cálculo de nuevos descriptores
- Manejo de datos atípicos
- Datos perdidos



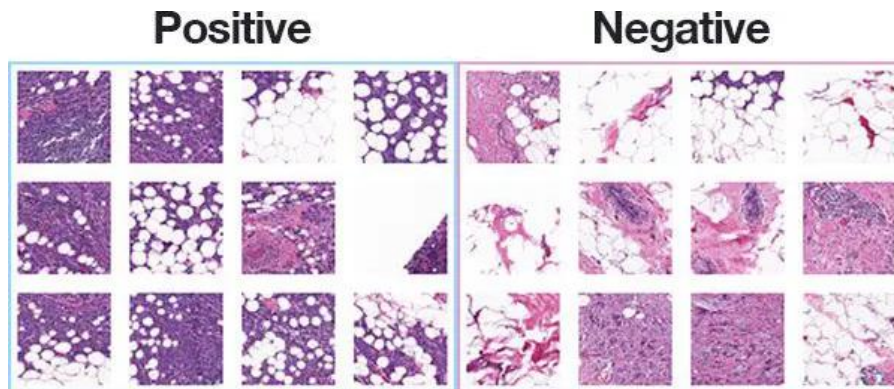
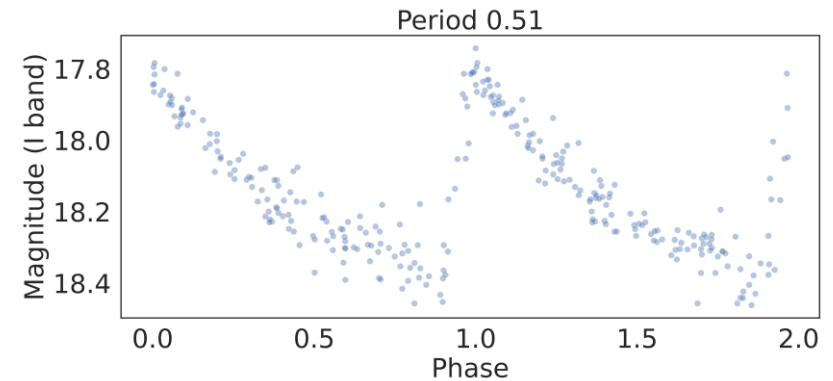
ESCUELA DE INGENIERÍA
FACULTAD DE INGENIERÍA

EDUCACIÓN
PROFESIONAL

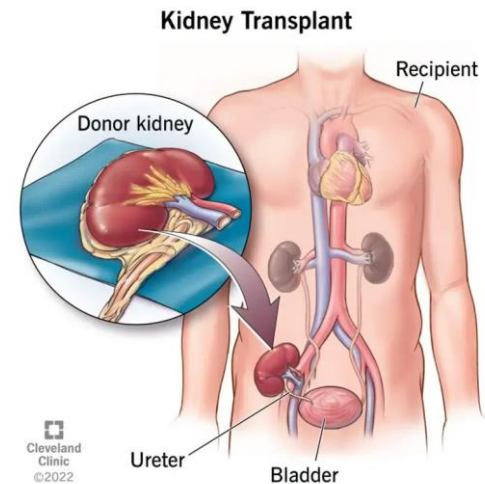
Volvamos a los clasificadores



Clasificadores:



<https://pyimagesearch.com/2019/02/18/breast-cancer-classification-with-keras-and-deep-learning/>



Desafío

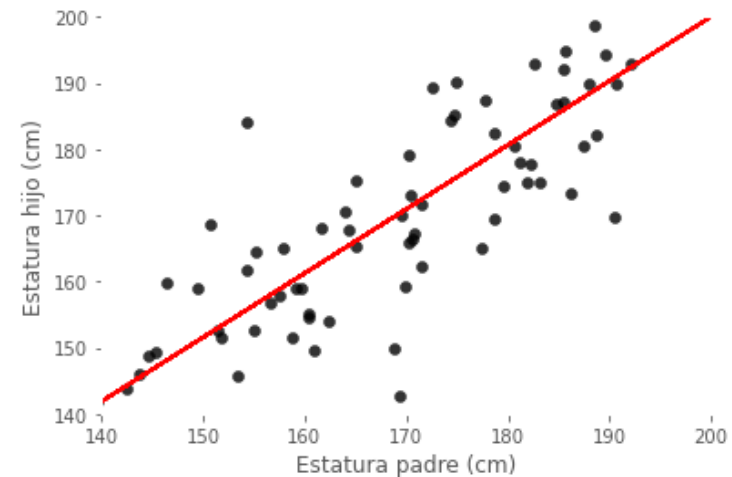


$$\mathcal{D} = \{(x_1, y_1), \dots, (x_N, y_N)\}$$

$$y_i \in \{0, 1\} \quad \forall i \in 1, 2, \dots, N$$

- Falla o no falla
- Compra o no compra
- Paga o no paga

¿Cómo pasamos de una regresión lineal a un problema de clasificación binaria?





ESCUELA DE INGENIERÍA
FACULTAD DE INGENIERÍA

EDUCACIÓN
PROFESIONAL

Regresión logística

Regresión logística

Usando una regression lineal, se puede obtener un valor numérico a partir de un conjunto de predictores.

$$\hat{y} = \hat{\theta}_0 + \hat{\theta}_1 x_1 + .. + \hat{\theta}_i x_i + .. + \hat{\theta}_n x_n$$

n = número de descriptores

$\hat{\theta}_j$ = j -ésimo parámetro del modelo

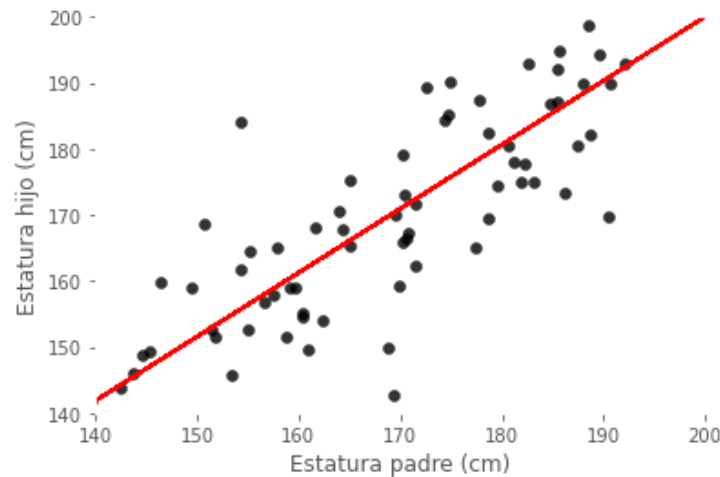
(incluyendo el intercepto θ_0 y los pesos que ponderan los descriptores $\theta_1, \theta_2, \dots, \theta_3$)

Regresión logística

Usando una regression lineal, se puede obtener un valor a partir de un conjunto de predictores.

Los parámetros aprendidos se obtienen de

$$\hat{\theta} = (X^T X)^{-1} X^T y$$





Regresión logística

¿Qué ocurre cuando la variable a predecir tiene respuesta binaria?

Es decir:

$$D = \{(\mathbf{x}_1, y_1), \dots, (\mathbf{x}_n, y_n)\}$$

$$y_i \in \{0,1\} \forall i \in 1, \dots, N$$

¿Se puede usar la misma regression lineal?

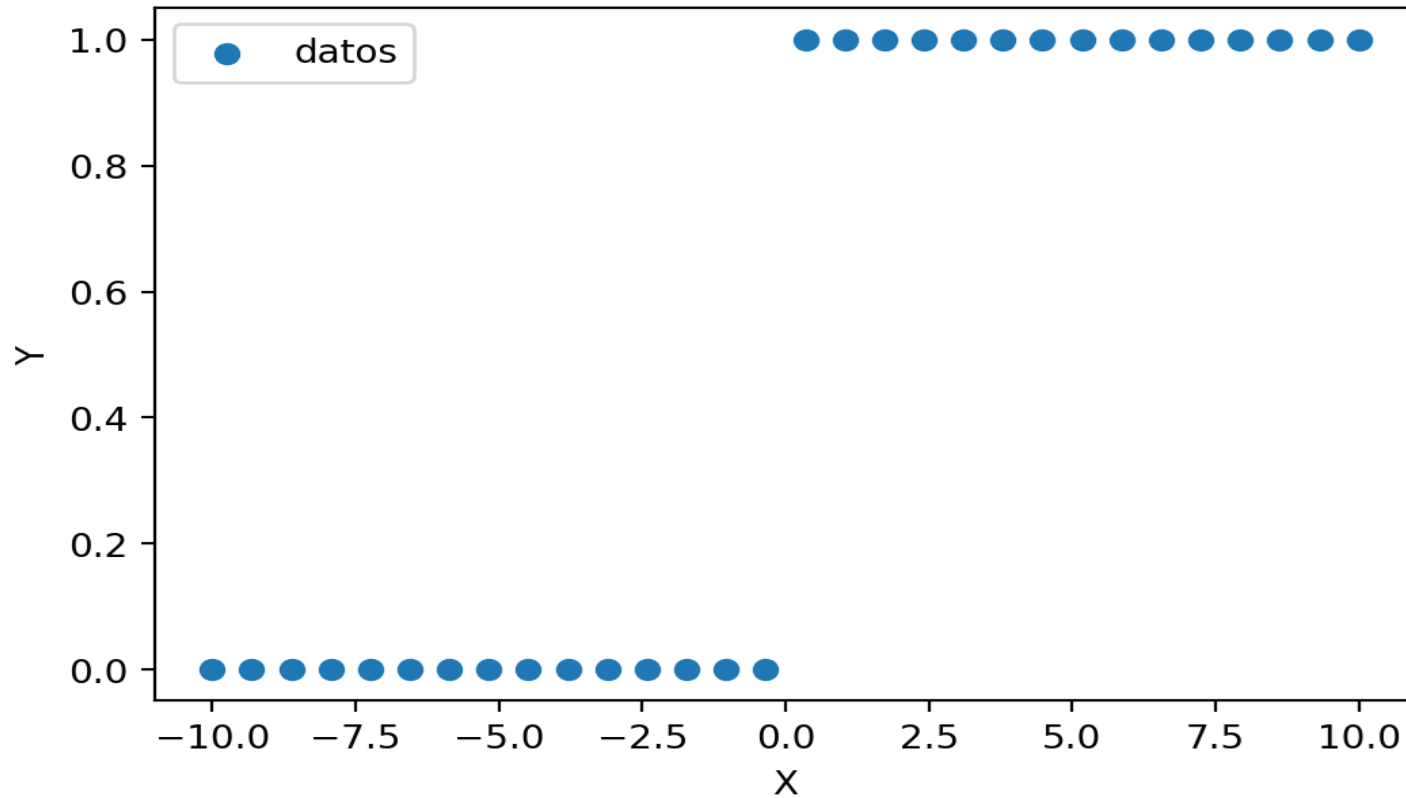


Figura 1. Datos para una clasificación binaria con un descriptor



Regresión logística

$$\hat{y} = \hat{\theta}_0 + \hat{\theta}_1 x_1 + .. + \hat{\theta}_i x_i + .. + \hat{\theta}_n x_n$$

$$\hat{y} = \hat{\theta}^T \cdot \mathbf{x} = h_{\hat{\theta}}(\mathbf{x})$$

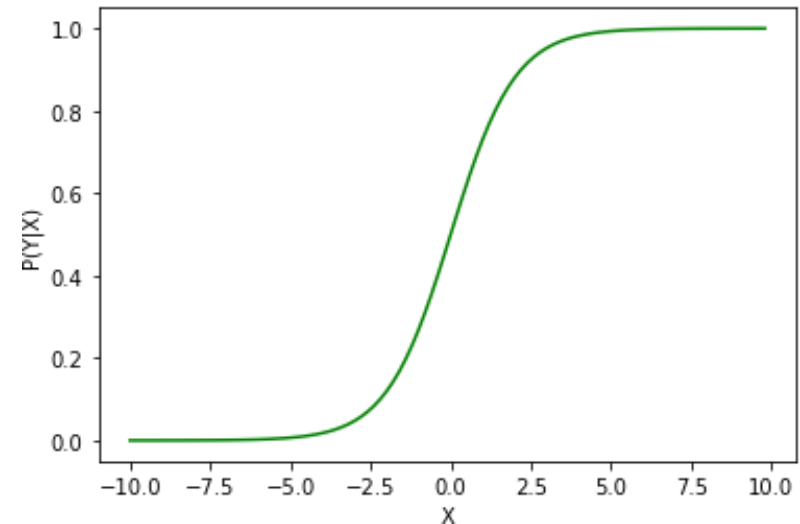
Regresión logística

Parámetros aprendidos

$$h_{\theta}(\mathbf{x}) = \sigma(\theta^T \cdot \mathbf{x})$$

Descriptores/Input

$$\sigma(\theta^T \cdot \mathbf{x}) = \frac{1}{1 + \exp(\theta^T \cdot \mathbf{x})}$$





¿Cómo encontramos los parámetros de la regresión logística?



¿Cómo encontramos los parámetros de la regresión logística?

$$p(y_i|x_i, \theta) = \underbrace{\sigma(\theta^T \cdot x_i)}_{p(y_i=1)}^{y_i} \underbrace{(1 - \sigma(\theta^T \cdot x_i))}_{p(y_i=0)}^{1-y_i}$$

$$p(y_i = 1)$$

$$p(y_i = 0)$$

$$\text{Ber}(y_i|\sigma(x_i, \theta))$$

¿Cómo encontramos los parámetros de la regresión logística?

$$J(\theta) = -\frac{1}{N} \sum_i^N y_i \log \sigma(\theta^T \cdot x_i) + (1 - y_i) \log(1 - \sigma(\theta^T \cdot x_i))$$

Función de pérdida = - Logaritmo de función de verosimilitud

Parámetros óptimos $\hat{\theta} = \arg \min_{\theta \in \Theta} J(\theta)$



Ejemplo

	income	credit_score	default
0	30	600	1
1	50	650	0
2	20	500	1
3	70	700	0
4	90	750	0
5	40	620	1
6	60	670	0
7	80	720	0
8	25	580	1
9	35	610	1

```
# Define predictors and target
X = df[['income', 'credit_score']]
y = df['default']

# Split the data into train and test sets
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.3, random_state=42)

# Fit a logistic regression model
model = LogisticRegression()
model.fit(X_train, y_train)
```


Ejemplo: predecir

$$\hat{y} = \frac{1}{1 + e^{-(111.5689 - 0.0707 \times \text{income} - 0.1689 \times \text{credit_score})}}$$

	income	credit_score	default
0	30	600	1
1	50	650	0
2	20	500	1
3	70	700	0
4	90	750	0
5	40	620	1
6	60	670	0
7	80	720	0
8	25	580	1
9	35	610	1

Test

```
import numpy as np

def predict_logistic_regression(coefficients, intercept, features, scaler=None):
    # Apply scaling if scaler is provided
    if scaler is not None:
        features = scaler.transform(features)

    # Compute the linear combination: z = intercept + sum(coeff * feature)
    z = np.dot(features, coefficients) + intercept

    # Apply the sigmoid function to compute probabilities
    probabilities = 1 / (1 + np.exp(-z))

    # Convert probabilities to class predictions (threshold at 0.5)
    predictions = (probabilities >= 0.5).astype(int)

    return probabilities, predictions
```



Ejemplo: veamos el código



ESCUELA DE INGENIERÍA
FACULTAD DE INGENIERÍA

EDUCACIÓN
PROFESIONAL

Naive Bayes



Naïve Bayes

Clasificador probabilístico, donde obtenemos la probabilidad para cada clase, dadas las variables de entrada.

$$p(Clase|X_1, X_2, X_3) = \frac{p(X_1, X_2, X_3|Clase)p(Clase)}{p(X_1, X_2, X_3)}$$



Teorema de Bayes

Probabilidad de los datos dada la hipótesis (likelihood)

Probabilidad de la hipótesis dado los datos (posterior)

$$\left\{ p(h|\mathcal{D}) = \frac{\overbrace{p(\mathcal{D}|h)} p(h)}{\underbrace{p(\mathcal{D})}} \right\}$$

Probabilidad previa de las hipótesis (prior)

Probabilidad de los datos (marginal likelihood)



Teorema de Bayes / Ejemplo numérico

$$p(h) = 0.08$$

Probabilidad de tener cáncer

$$p(+|h) = 0.98$$

Probabilidad de examen positivo dado que tiene cáncer

$$p(+|\neg h) = 0.03$$

Probabilidad de examen positivo dado que no tiene cáncer

$$p(h|\mathcal{D}) = \frac{p(\mathcal{D}|h)p(h)}{p(\mathcal{D})}$$

Teorema de Bayes / Ejemplo numérico

$$p(h) = 0.08$$

Probabilidad de tener cáncer

$$p(+|h) = 0.98$$

Probabilidad de examen positivo dado que tiene cáncer

$$p(+|\neg h) = 0.03$$

Probabilidad de examen positivo dado que no tiene cáncer

$$p(h|\mathcal{D}) = \frac{p(\mathcal{D}|h)p(h)}{p(\mathcal{D})}$$

$$p(h|+) = \frac{p(+|h)p(h)}{p(+|h)p(h) + p(+|\neg h)p(\neg h)}$$

Teorema de Bayes / Ejemplo numérico

$$p(h|\mathcal{D}) = \frac{p(\mathcal{D}|h)p(h)}{p(\mathcal{D})}$$

$$p(h) = 0.08$$

Probabilidad de tener cáncer

$$p(+|h) = 0.98$$

Probabilidad de examen positivo dado que tiene cáncer

$$p(+|\neg h) = 0.03$$

Probabilidad de examen positivo dado que no tiene cáncer

$$p(\neg h) = 0.992 \quad p(-|h) = 0.02 \quad p(-|\neg h) = 0.97$$

$$p(h|+) = \frac{p(+|h)p(h)}{p(+|h)p(h) + p(+|\neg h)p(\neg h)} = \frac{0.0078}{0.0078 + 0.298} = 0.21$$



ESCUELA DE INGENIERÍA
FACULTAD DE INGENIERÍA

EDUCACIÓN
PROFESIONAL

Clasificador naive Bayes

Teorema de Bayes

$$p(h|\mathcal{D}) = \frac{p(\mathcal{D}|h)p(h)}{p(\mathcal{D})}$$

Diagram illustrating the Bayes' Theorem formula with labels:

Descriptores (above the numerator's condition)

Probabilidad a priori de la clase (above the prior term $p(y)$)

Probabilidad a posteriori de la clase (below the posterior term $p(y|x_1, \dots, x_n)$)

$$\underbrace{p(y|x_1, \dots, x_n)}_{\text{Probabilidad a posteriori de la clase}} = \frac{p(x_1, \dots, x_n|y) \overbrace{p(y)}^{\text{Probabilidad a priori de la clase}}}{p(x_1, \dots, x_n)}$$

Naïve Bayes

Supuesto de independencia condicional

$$p(x_i|y, x_1, \dots, x_{i-1}, x_{i+1}, \dots, x_n) = p(x_i|y) \quad \forall i \in 1, \dots, n$$

$$p(y|x_1, \dots, x_n) = \frac{p(x_1, \dots, x_n|y)p(y)}{p(x_1, \dots, x_n)}$$



$$p(y|x_1, \dots, x_n) = \frac{\prod_{i=1}^n p(x_i|y)p(y)}{p(x_1, \dots, x_n)}$$

Naïve Bayes

$$p(y|x_1, \dots, x_n) = \frac{\prod_{i=1}^n p(x_i|y)p(y)}{p(x_1, \dots, x_n)}$$

← **Constante**

↓

$$p(y|x_1, \dots, x_n) \propto \prod_{i=1}^n p(x_i|y)p(y)$$

↓

$$\hat{y} = \arg \max_y \prod_{i=1}^n p(x_i|y)p(y)$$

← **Clasificador**

Naïve Bayes

$$\hat{y} = \arg \max_y \prod_{i=1}^n \underbrace{p(x_i|y)p(y)}$$

Supuesto respecto al tipo de distribución de los descriptores

- Gaussiana
- Bernoulli
- Categórica
- Mixta



ESCUELA DE INGENIERÍA
FACULTAD DE INGENIERÍA

EDUCACIÓN
PROFESIONAL

Evaluación de clasificadores



Matriz de confusión

Muestra de forma gráfica el resultado del clasificador por clase.

		Predicción		
		A	B	C
Etiqueta	A	80	15	5
	B	6	89	5
	C	21	3	76



ESCUELA DE INGENIERÍA
FACULTAD DE INGENIERÍA

EDUCACIÓN
PROFESIONAL

**¿Qué métricas podemos
obtener desde la matriz de
confusión?**

Accuracy

Medida rendimiento a nivel general del clasificador. No toma en cuenta desbalance de las clases.

Etiqueta

		Predicción		
		A	B	C
Etiqueta	A	80	15	5
	B	6	89	5
	C	21	3	76

$$Acc = \frac{80 + 89 + 76}{80 + 15 + 5 + 6 + 89 + 5 + 21 + 3 + 76} = 81.66\%$$

Precision

Medida de falsos positivos.
Implica qué tanto se debe
confiar en la predicción.

Etiqueta

		Predicción		
		A	B	C
Etiqueta	A	80	15	5
	B	6	89	5
	C	21	3	76

$$Precision = \frac{80}{80 + 6 + 21} = 74.76\%$$

Recall

Medida de falsos negativos.
Implica qué fracción del total
de elementos el clasificador
puede rescatar.

Etiqueta

		Predicción		
		A	B	C
Etiqueta	A	80	15	5
	B	6	89	5
	C	21	3	76

$$Recall = \frac{80}{80 + 15 + 5} = 80\%$$

F1-Score

Medida que integra precision y recall en un solo valor.

Etiqueta

	Predicción		
	A	B	C
A	80	15	5
B	6	89	5
C	21	3	76

$$F = \frac{2}{\frac{1}{recall} + \frac{1}{precision}} = \frac{2}{\frac{1}{0.80} + \frac{1}{0.74}} \approx 0.769$$



ESCUELA DE INGENIERÍA
FACULTAD DE INGENIERÍA

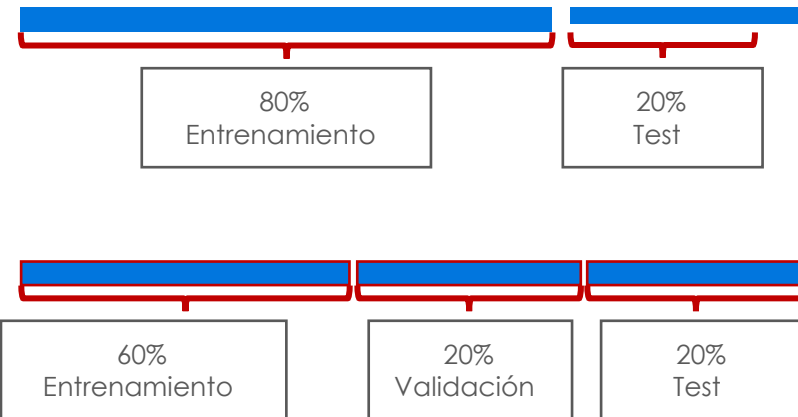
EDUCACIÓN
PROFESIONAL

¿Cómo validamos estas métricas?

Hold-out

Un clasificador debe ser evaluado en un set de datos distinto al de entrenamiento.

Realizar muestreo **estratificado**, asegurando la proporción de clases es crucial y siempre debe hacerse.

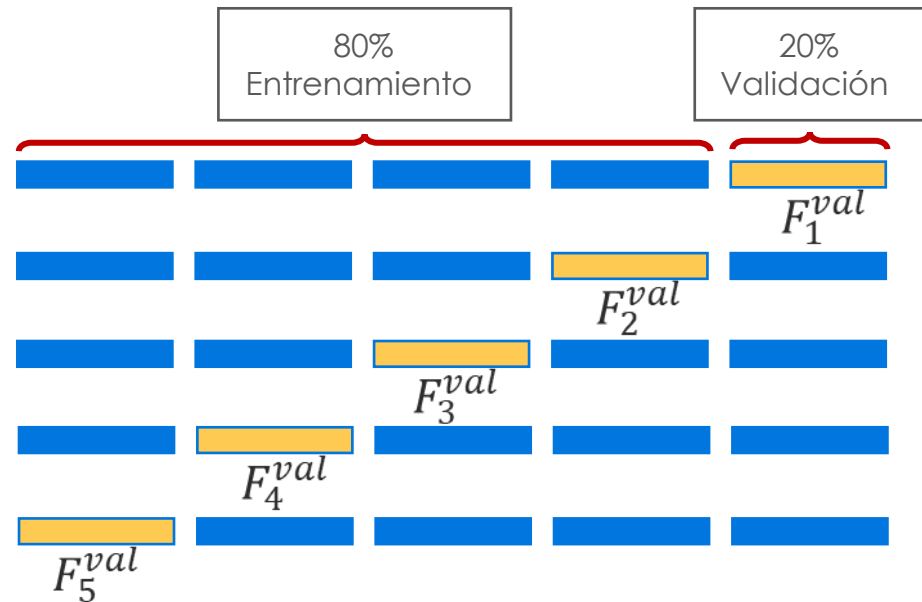


Validación cruzada

La validación cruzada consiste en dividir el set de datos en un número fijo de partes.

Una de ellas se usa para evaluar y el resto para entrenar.

El proceso se repite hasta que se agotan los sets de evaluación.



$$F^{val} = \frac{1}{5} \sum_{k=1}^5 F_i^{val}$$



ESCUELA DE INGENIERÍA
FACULTAD DE INGENIERÍA

EDUCACIÓN
PROFESIONAL



www.educacionprofesional.ing.uc.cl



¿Qué son las probabilidades?

Probabilidad

Es un cálculo de posibilidad de que algo suceda.

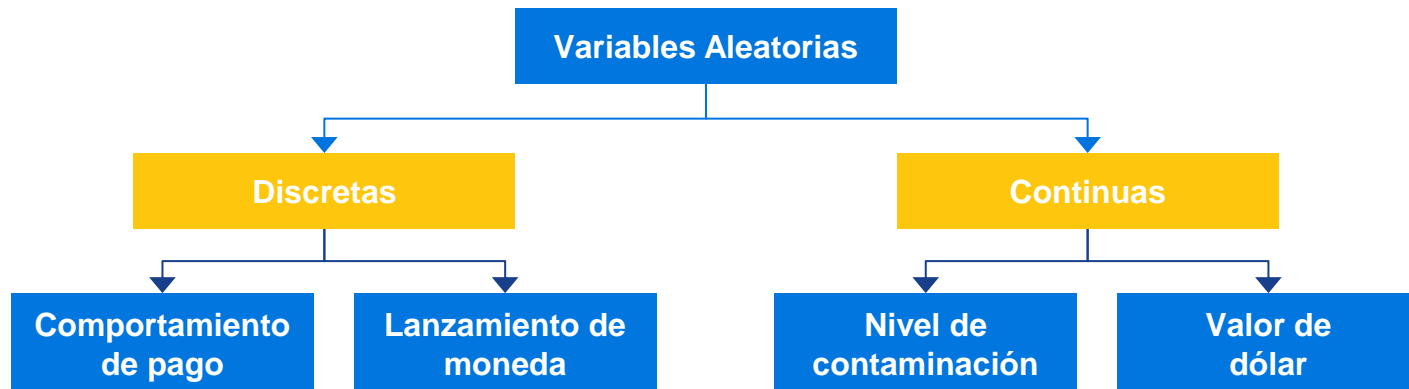
Al no estar seguros sobre el resultado de un evento

Se puede hablar sobre las probabilidades de ciertos resultados.

Variable aleatoria

Variable cuyos valores posibles son resultados numéricos de un fenómeno aleatorio.

Variables aleatorias





Continua vs Discreta

Regla de suma

$$p(A \vee B) = p(A) + p(B) - p(A \wedge B)$$

$$p(A) + p(B)$$

Regla del producto

$$p(A, B) = p(A \wedge B) = p(A|B)p(B)$$

Probabilidad Marginal

$$p(A) = \sum_b p(A, B = b) = \sum_b p(A|B = b)p(B = b)$$



Continua vs Discreta

Continuas

$$\int_x p(x) dx = 1$$

Función de densidad de
probabilidad

Discretas

$$\sum_x p(x) = 1$$

Función de cuantía de
probabilidad