UNIVERSIDAD EAFIT

Escuela de Ciencias Aplicadas e Ingeniería

Artificial Intelligence

Lecture05 – Supervised ML pipeline fundamentals

Contenido

UNIVERSIDAD EAFIT
Escuela de Ciencias Aplicadas e Ingeniería

1. ¿Cómo aprende una computadora?
2. Ciclo de vida de un proyecto de Machine Learning
3. Taxonomía del Machine Learning
4. Métricas de desempeño en aprendizaje supervisado
5. Preparando los datos para aplicar Machine Learning
6. Análisis exploratorio de datos
7. Ingeniería de características
8. Regresión Lineal, Logística

¿Cómo aprende una computadora?

¿Cómo aprende una computadora usando ML?

UNIVERSIDAD EAFIT
Escuela de Ciencias Aplicadas e Ingeniería

Un algoritmo de ML aprende mediante un **modelo** o función de hipótesis (a menudo denotado h) cuyos **parámetros** o reglas se ajustan a los datos.

El modelo h es simplemente una función que recibe unas **características** y devuelve una **predicción.**

Parámetros e hiperparámetros de un modelo de ML

UNIVERSIDAD EAFIT
Escuela de Ciencias Aplicadas e Ingeniería

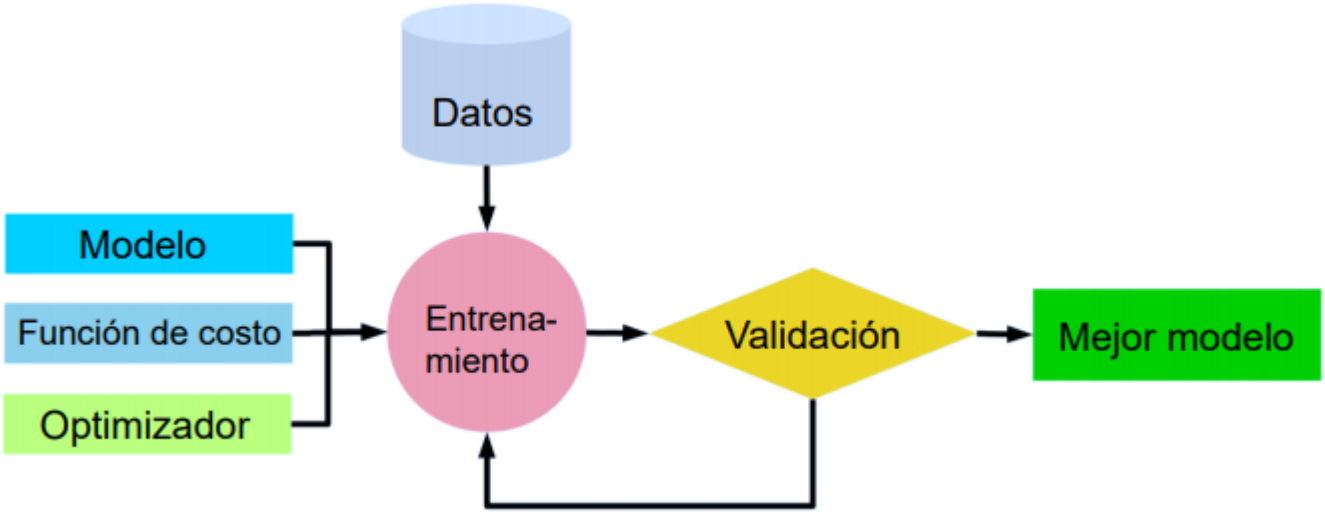
Parámetros:

- Se establecen mediante el entrenamiento
- Se deben guardar como parte del modelo
- **Ejemplos**: Los pesos de una regresión lineal o los puntos de corte de un árbol de decisión

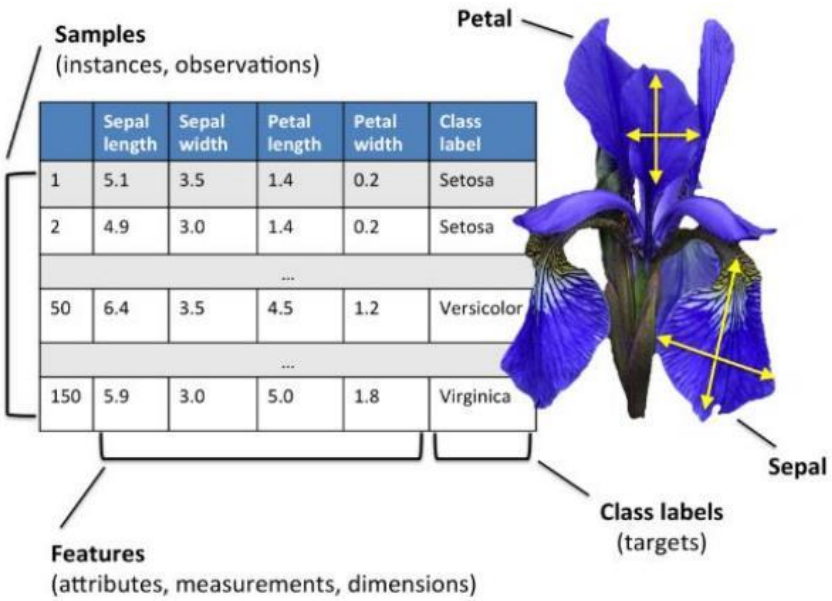
Hiperparámetros:

- Se establecen antes de entrenar (a menudo mediante un proceso de validación)
- Por lo general no es necesario guardarlos para reproducir el modelo
- **Ejemplos**: El parámetro de regularización de una regresión lineal o la profundidad máxima de un árbol de decisión

¿Cómo aprende una computadora usando ML?



¿Cómo Son los Datos que Ingiere un Algoritmo de ML?



	Sepal length	Sepal width	Petal length	Petal width	Class label
1	5.1	3.5	1.4	0.2	Setosa
2	4.9	3.0	1.4	0.2	Setosa
...					
50	6.4	3.5	4.5	1.2	Versicolor
...					
150	5.9	3.0	5.0	1.8	Virginica

Las **características** se suelen denotar con $\mathbf{X}$ y la variable a **predecir** se suele denotar con $\boldsymbol{y}$.

La variable a **predecir** a menudo se denomina **etiqueta.**

UNIVERSIDAD EAFIT

Escuela de Ciencias Aplicadas e Ingeniería

Ciclo de vida de un proyecto de Machine Learning

Pasos para entrenar un modelo de ML

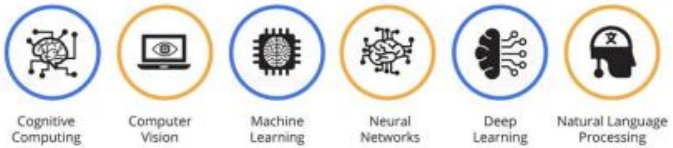
1. **Recolectar los datos**. Podemos recolectar los datos desde muchas fuentes, podemos por ejemplo extraer los datos de un sitio web o obtener los datos utilizando una API, desde una base de datos, desde otros dispositivos...
2. **Preprocesar los datos**. Una vez que tenemos los datos, tenemos que asegurarnos que tiene el formato correcto para nutrir nuestro algoritmo de aprendizaje.
3. **Explorar los datos**. Una vez que ya tenemos los datos y están con el formato correcto, podemos realizar un pre análisis para corregir los casos de valores faltantes o intentar encontrar a simple vista algún patrón en los mismos que nos facilite la construcción del modelo.
4. **Entrenar el Algoritmo.** En esta etapa nutrimos al o a los algoritmos de aprendizaje con los datos que venimos procesando en las etapas anteriores.
5. **Evaluar el Algoritmo**. En esta etapa ponemos a prueba la información o conocimiento que el algoritmo obtuvo del entrenamiento del paso anterior. Evaluamos la precisión del algoritmo en sus predicciones y si no estamos conformes con su rendimiento, podemos volver a la etapa anterior y continuar entrenando.
6. **Utilizar el modelo**. Medimos su rendimiento, lo que tal vez nos obligue a revisar todos los pasos anteriores.

Taxonomía del Machine Learning

¿Qué es inteligencia artificial?

UNIVERSIDAD EAFIT
Escuela de Ciencias Aplicadas e Ingeniería

El objetivo de la IA es lograr que los computadores hagan cosas que requieren inteligencia humana.



"The theory and development of computer systems able to perform tasks that normally require human intelligence" - Merriam Webster

Understanding language, reasoning, speech recognition, decision-making, navigating the visual world, manipulating physical objects, etc.

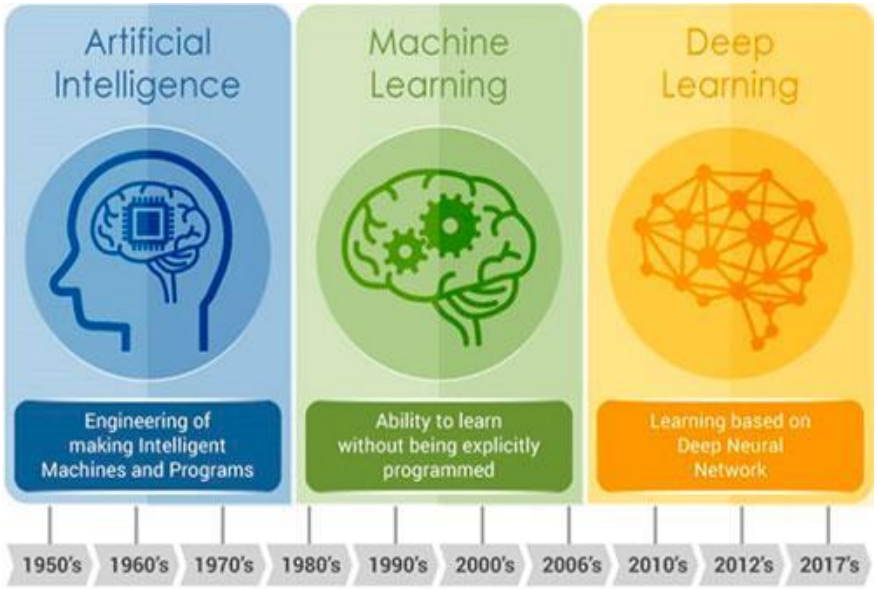
¿Qué es aprendizaje de máquina?

UNIVERSIDAD EAFIT
Escuela de Ciencias Aplicadas e Ingeniería

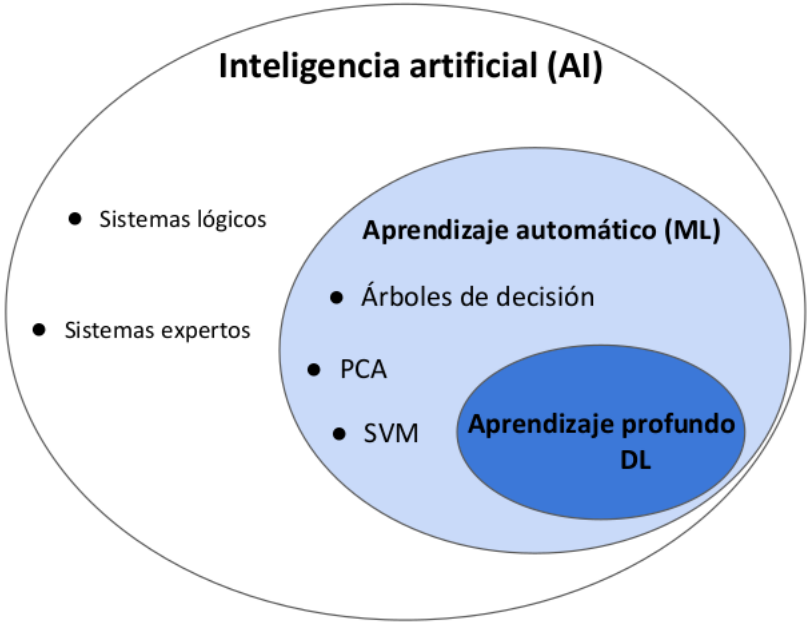
"Machine learning is a branch of artificial intelligence (AI) and computer science which focuses on the use of data and algorithms to imitate the way that humans learn, gradually improving its accuracy"

Los algoritmos de ML se caracterizan por su habilidad para **aprender de los datos** sin estar explícitamente programados.

¡ML se utiliza comúnmente para hacer predicciones!



¿Qué es inteligencia artificial?



Inteligencia artificial: estudia los comportamientos inteligentes en dispositivos.

Machine learning: estudia los algoritmos que mejoran su rendimiento incorporando nuevos datos a un modelo existente.

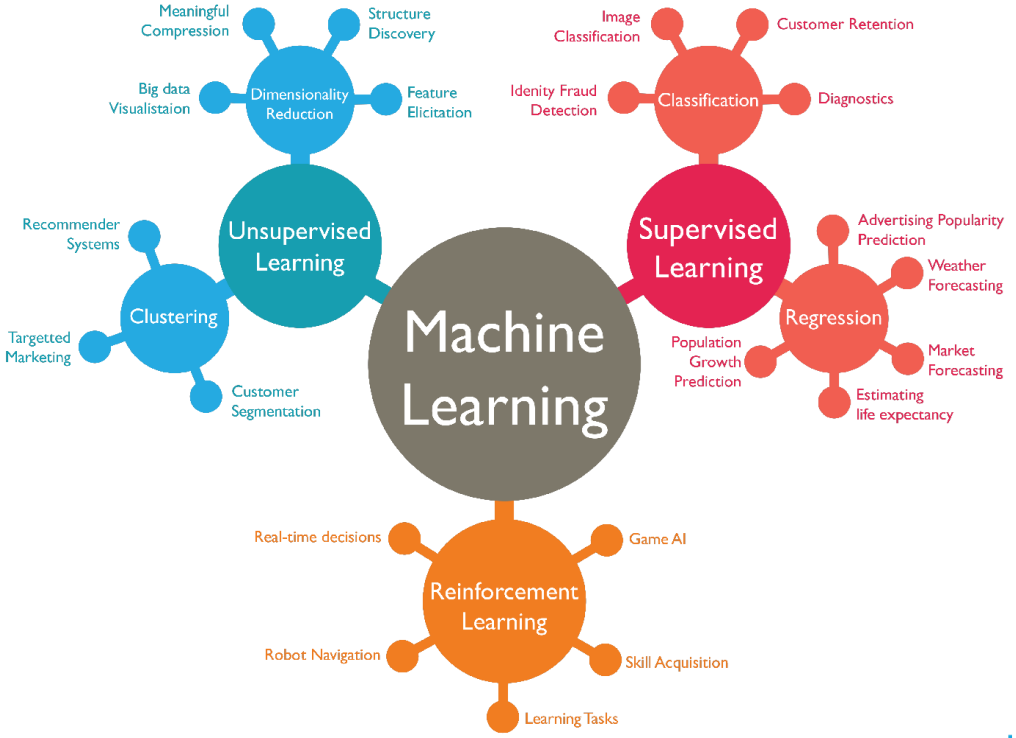
Deep learning: usa algoritmos de redes neuronales artificiales para extraer características cada vez más complejas a partir de los datos de entrada, buscando mejorar su rendimiento

Tipos de Aprendizaje en ML

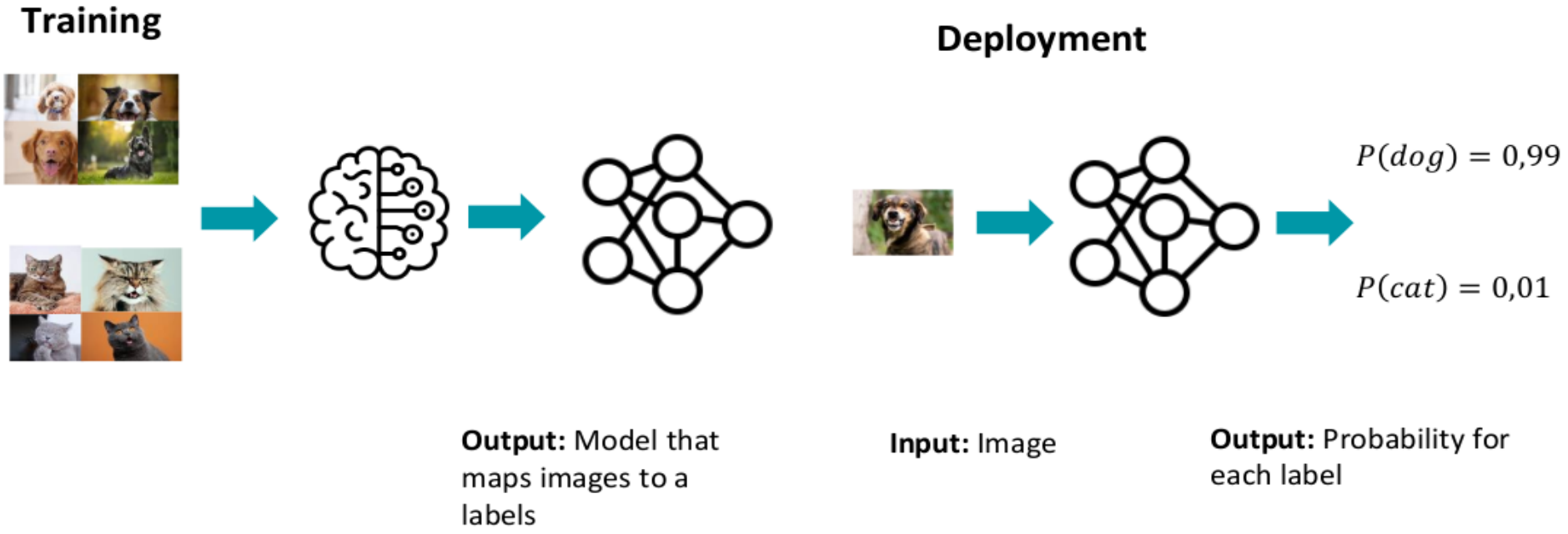
UNIVERSIDAD EAFIT | Escuela de Ciencias Aplicadas e Ingeniería

Existen tres tipos principales de aprendizaje en ML:

- Aprendizaje supervisado
- Aprendizaje no supervisado
- Aprendizaje por refuerzo



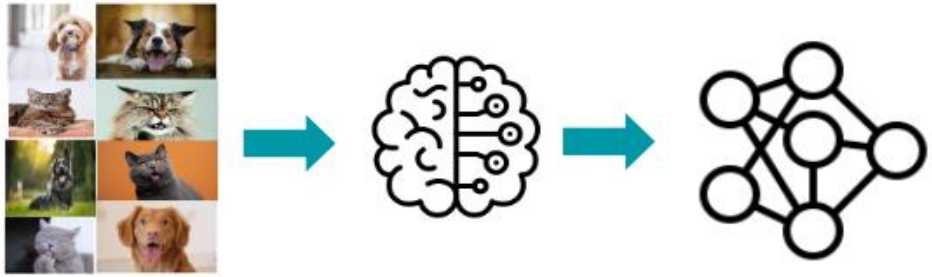
Tipos de ML: Aprendizaje supervisado



Tipos de ML: Aprendizaje no supervisado

UNIVERSIDAD EAFIT | Escuela de Ciencias Aplicadas e Ingeniería

Training



Input: Images (without labels)

Output: Model that finds similar groups (clusters)

Deployment

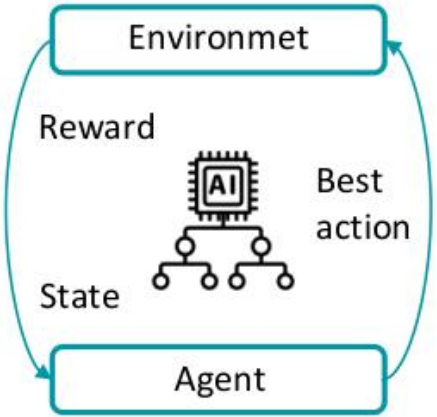
Input: Image

Output: group

Tipos de ML: Aprendizaje por refuerzo

UNIVERSIDAD EAFIT | Escuela de Ciencias Aplicadas e Ingeniería

Training

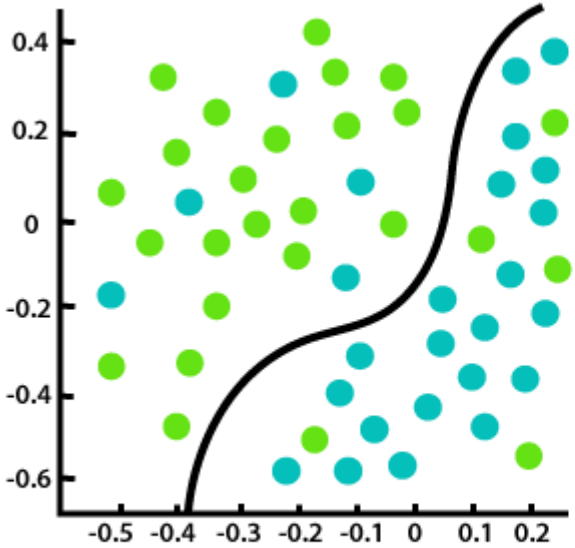


Deployment

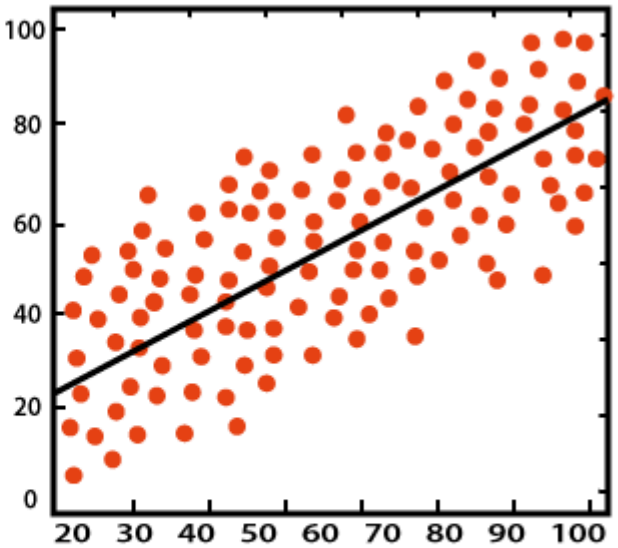
Input: Images (without labels)

Output: best action

Tipos de Aprendizaje Supervisado

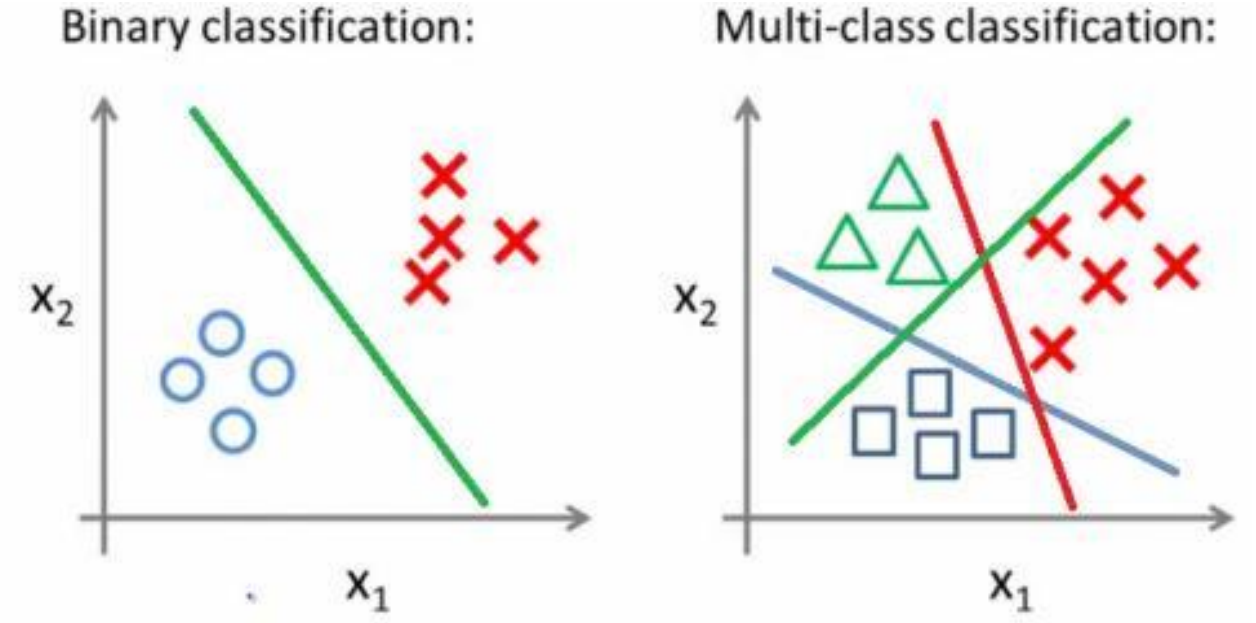


Classification



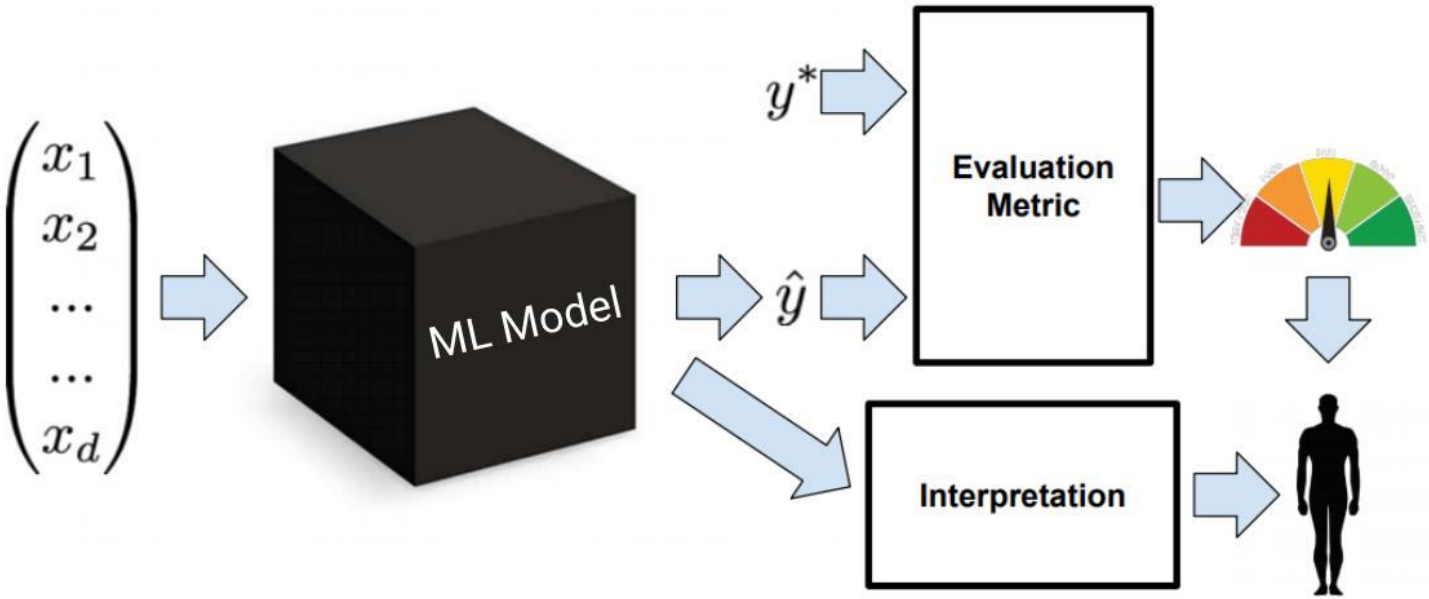
Regression

Tipos de Clasificación



Métricas de desempeño en aprendizaje supervisado

Evaluación de un Modelo de ML



Métricas Comunes en Aprendizaje Supervisado

UNIVERSIDAD EAFIT
Escuela de Ciencias Aplicadas e Ingeniería

Métricas de regresión:

- Error cuadrático medio (MSE)
- Error absoluto medio (MAE)
- Raíz cuadrada del error cuadrático medio (RMSE)
- Error absoluto porcentual medio (MAPE)

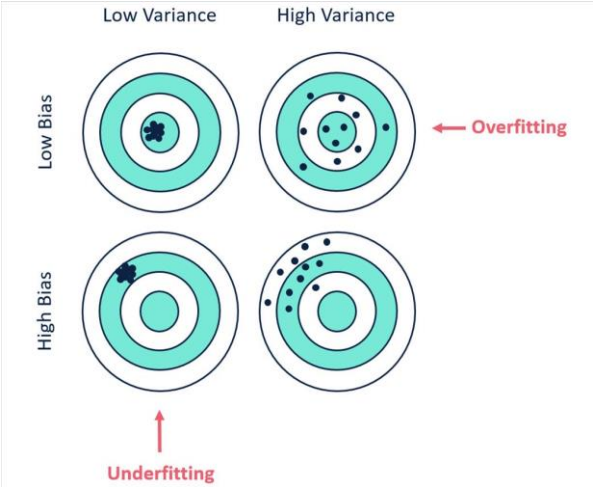
Métricas de clasificación:

- Accuracy
- Precision
- Recall
- F1-score

Sesgo y Varianza

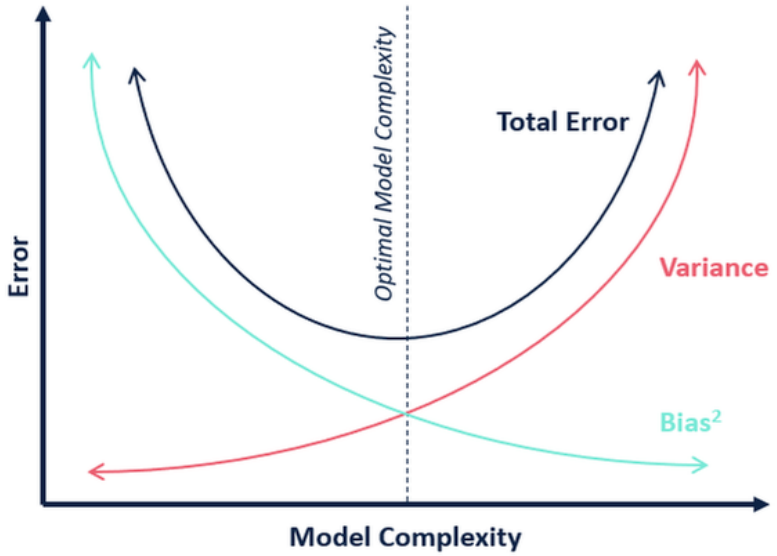
Si no hay errores de código y los datos están limpios y tienen sentido, hay dos fuentes principales de error en los modelos de ML:

- Sesgo (bias)
- Varianza (variance)

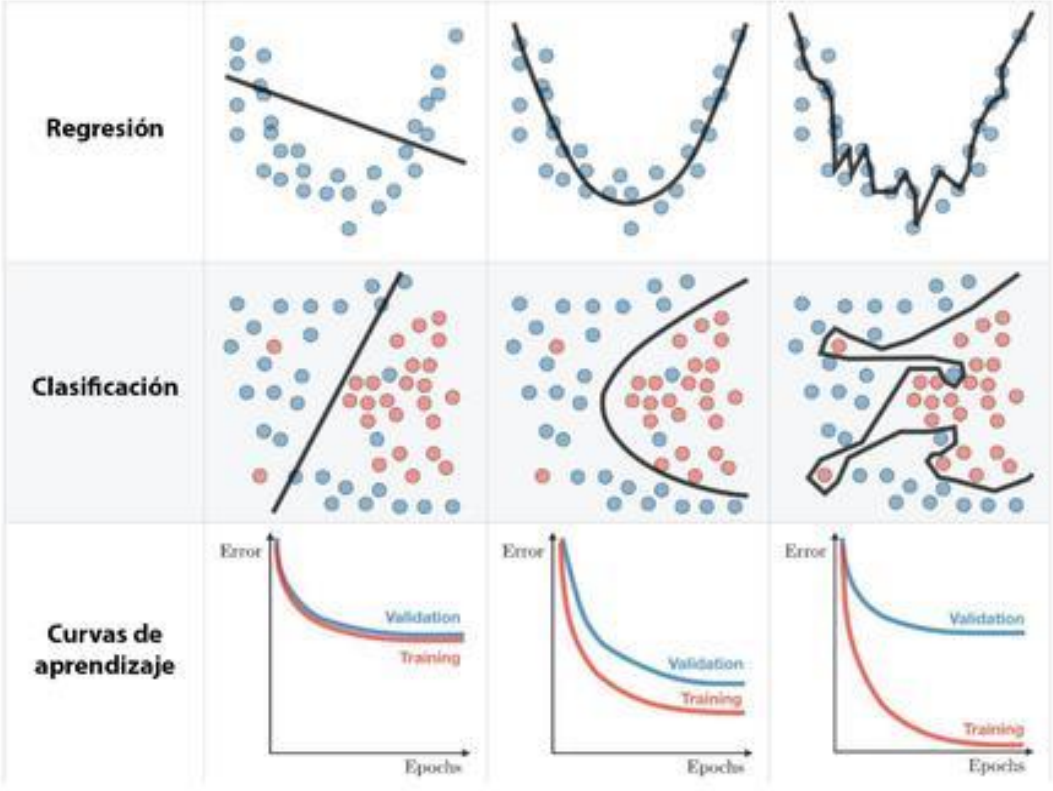


Sesgo y Varianza

Por lo general, asociamos el sesgo alto con modelos muy simples que subajustan (underfit) los datos y asociamos la varianza alta con modelos muy complejos que sobreajustan (overfit) los datos.



Subajuste y Sobreajuste

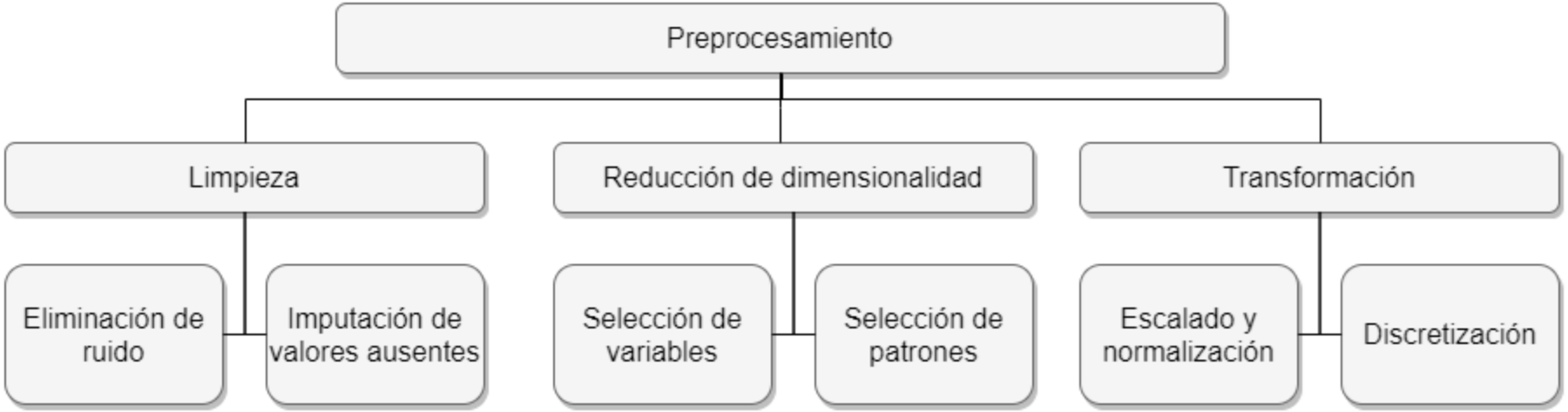


UNIVERSIDAD EAFIT

Escuela de Ciencias Aplicadas e Ingeniería

Preparando los datos para aplicar Machine Learning

Tareas comunes de procesamiento



UNIVERSIDAD EAFIT

Escuela de Ciencias Aplicadas e Ingeniería

Análisis exploratorio de datos

Exploratory Data Analysis (EDA)

UNIVERSIDAD EAFIT
Escuela de Ciencias Aplicadas e Ingeniería

Metodología para analizar bases de datos y encontrar sus principales características.

- **Coleccionar** o concatenar datos
- **Realizar investigaciones iniciales** para descubrir patrones, detectar anomalías, validar hipótesis y verificar suposiciones:
  - Estadísticas descriptivas
  - Representaciones gráficas (histogramas, barras,…)
- **Procesar** los datos para obtener información relevante
- **¡Más un arte que una ciencia!**

Estadística descriptiva

UNIVERSIDAD EAFIT | Escuela de Ciencias Aplicadas e Ingeniería

Estadísticas generales - `df.head(), df.shape, df.info()`

- Número de muestras (filas)
- Número de características (columnas)

Estadística univariada (para cada característica)

- Estadística para variables numéricas (media, varianza, histogramas)

`df.describe(), hist(df[feature])`

- Estadística para variables categóricas (histogramas, moda, valores más/menos frecuentes, porcentajes, número de valores únicos)
  - Histograma de valores - `df[feature].value_counts(), sns.distplot()`

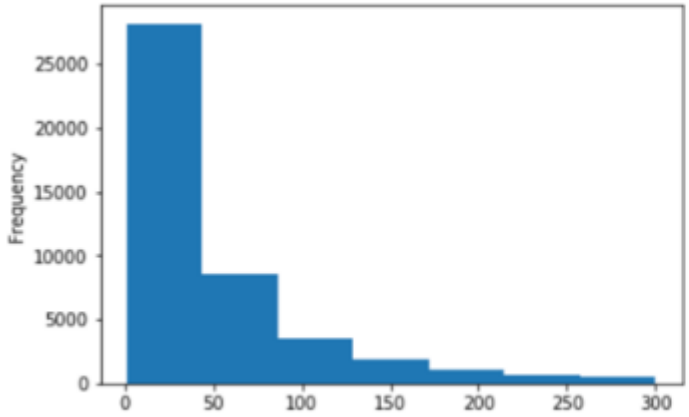
- Estadísticas de la variable objetivo
  - Distribución de clases – `df[target].value_counts(), np.bincount(y)`

Estadística descriptiva

UNIVERSIDAD EAFIT
Escuela de Ciencias Aplicadas e Ingeniería

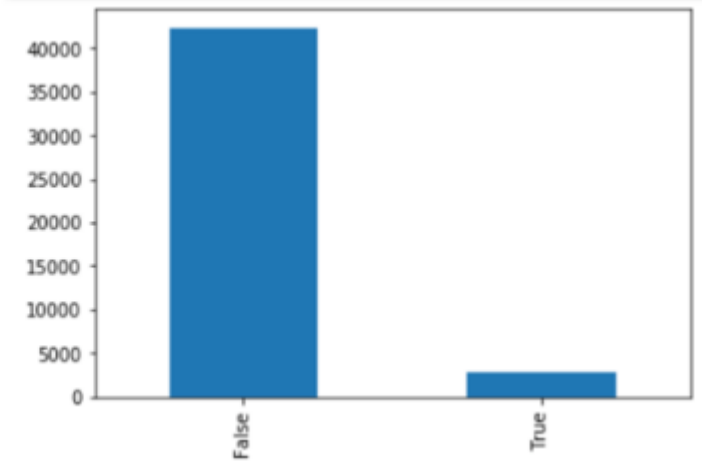
Características numéricas

```
import matplotlib.pyplot as plt

df[num_feature].plot.hist(bins = 7)
plt.show()
```



Características categóricas

```
import matplotlib.pyplot as plt

df[cat_feature].value_counts().plot.bar()
plt.show()
```



Estadística multivariada

UNIVERSIDAD EAFIT
Escuela de Ciencias Aplicadas e Ingeniería

Estadísticas multivariadas (más de una característica)

Correlaciones: Cómo se relacionan (linealmente) pares de características

```
[feature1, feature2]
df[cols].corr()
```

	feature1	feature2
feature1	1	0.0128493
feature2	0.0128493	1

	feature1	feature2
feature1	1	0.882106
feature2	0.882106	1

Los valores de la correlación están entre -1 y 1:

- -1 significa correlación negativa perfecta
- 1 significa correlación positiva perfecta
- 0 significa que no hay correlación entre las variables.

Estadística multivariada

UNIVERSIDAD EAFIT
Escuela de Ciencias Aplicadas e Ingeniería

Las características altamente correlacionadas pueden afectar el rendimiento de algunos modelos de aprendizaje de máquina, como la regresión lineal o logística, cuando su implementación involucra el cálculo de inversas de matrices. Además, puede esconder la relevancia de alguna característica.

- Seleccione una de las características correlacionadas y descarte las demás
- Otros modelos, como los árboles de decisión, son inmunes a este problema

Adicionalmente, características altamente correlacionadas con la variable objetivo
podrían mejorar el rendimiento de modelos lineales.

Imputación de datos faltantes

UNIVERSIDAD EAFIT | Escuela de Ciencias Aplicadas e Ingeniería

Descartar filas y/o columnas con valores faltantes: Remover esas filas y/o columnas de la base de datos.

Una menor cantidad de datos puede generar modelos sobre/sub entrenados

Imputar (completar) los valores faltantes:

- **Reemplazar** los valores faltantes de características numéricas con el promedio o de características categóricas con la moda
  `df['col'].fillna((df['col'].mean())), df['col'].fillna((df['col'].mode()))`
- Placeholder: Asignar un valor común a los valores faltantes
- Técnicas avanzadas: Predecir los valores faltantes usando ML (AWS Datawig usa NNs)

https://github.com/awslabs/datawig

Imputación de datos faltantes

UNIVERSIDAD EAFIT
Escuela de Ciencias Aplicadas e Ingeniería

`SimpleImputer`: **Clase** de **sklearn** para utilizar métodos de imputación, implementa los métodos `.fit()` y `.transform()`

SimpleImputer(missing_values=nan, strategy='mean', fill_value=None)

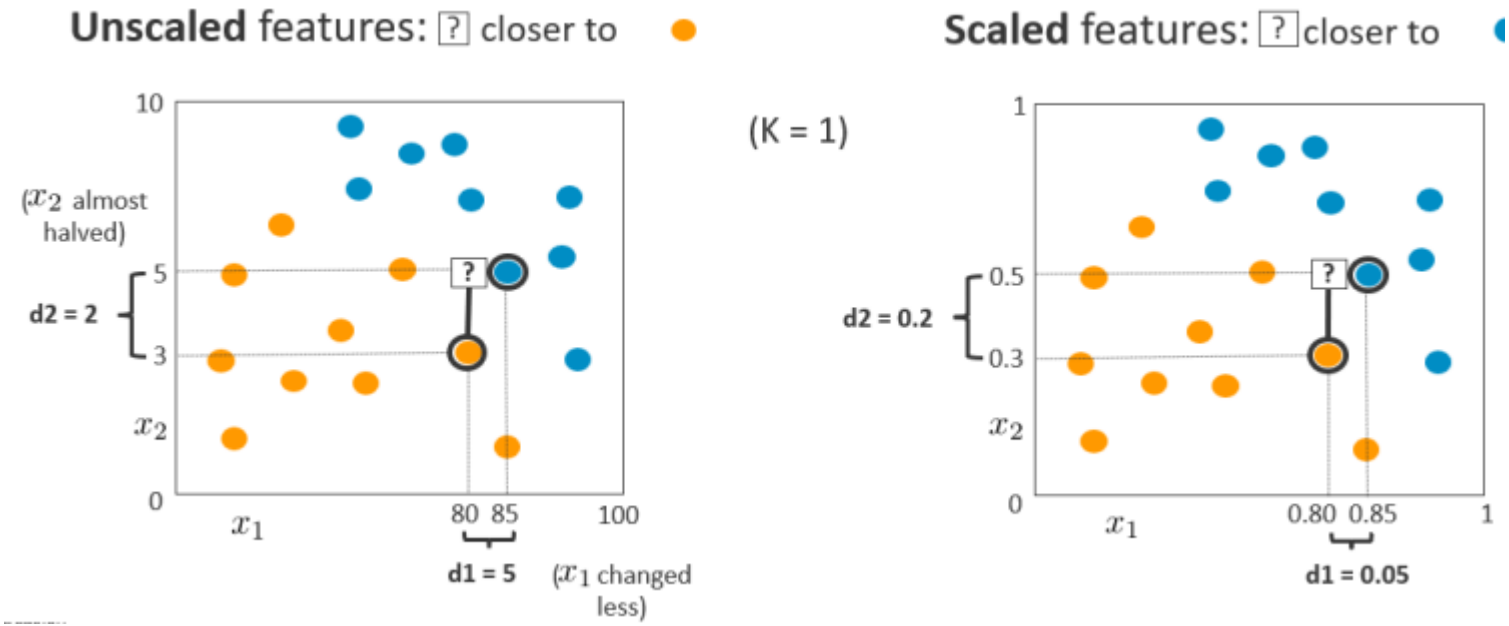
- **numerical data:**
  Strategy = “mean”, replace missing values using the mean along each column
  Strategy = “median”, replace missing values using the median along each column

- **numerical or categorical data:**
  Strategy = “most_frequent”, replace missing using the most frequent value along each column
  Strategy = “constant”, replace missing values with fill_value

Estandarización de características

Motivación: Algoritmos como kNN y redes neuronales cambian su comportamiento cuando las características no están en la misma escala.

Solución: Llevar las características a la misma escala (mean/variance, MinMax)



Estandarización de características

UNIVERSIDAD EAFIT
Escuela de Ciencias Aplicadas e Ingeniería

`StandardScaler` **Clase** de **sklearn**, implementa los métodos `.fit()` y `.transform()` para escalar cada característica (columna) de forma que quede con media y desviación estándar 1.

$$x_{scaled} = \frac{x - x_{mean}}{x_{std}}$$

`MinMAxScaler` **Clase** de **sklearn**, implementa los métodos `.fit()` y `.transform()` para escalar cada característica (columna) de forma que los valores mínimo y máximo sean 0 y 1.

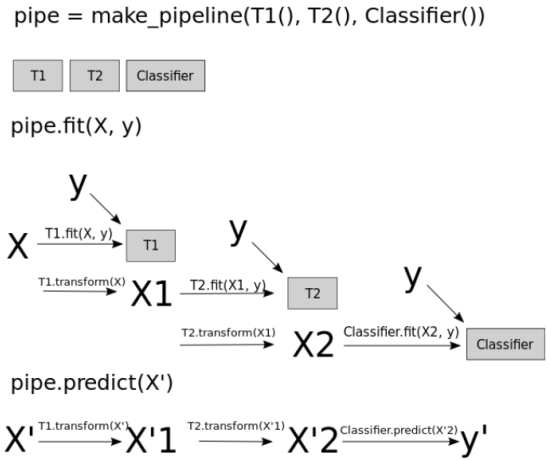
$$x_{scaled} = \frac{x - x_{min}}{x_{max} - x_{min}}$$

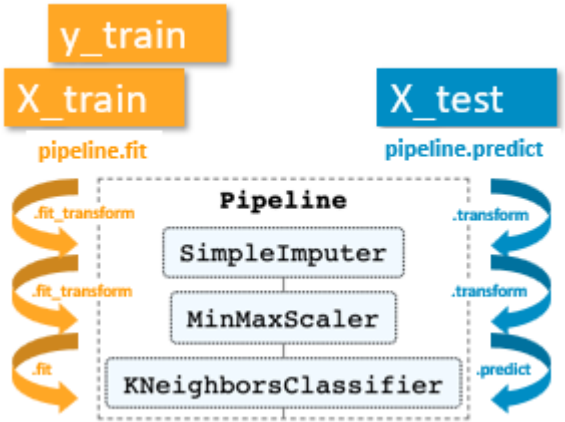
Pipeline (sklearn)

UNIVERSIDAD EAFIT | Escuela de Ciencias Aplicadas e Ingeniería

Pipeline: **secuencia** de transformaciones de datos que, generalmente, finaliza con un estimador, implementa los métodos `.fit()` y `.predict()`. Previene el sesgo a los datos de entrenamiento.

```
pipeline = Pipeline([
    ('imputer', SimpleImputer(strategy='mean')),
    ('scaler', MinMaxScaler()),
    ('clf', KNeighborsClassifier(n_neighbors = 3))
    ])

pipeline.fit(X_train, y_train)
predictions = pipeline.predict(X_test)
```





Ingeniería de características

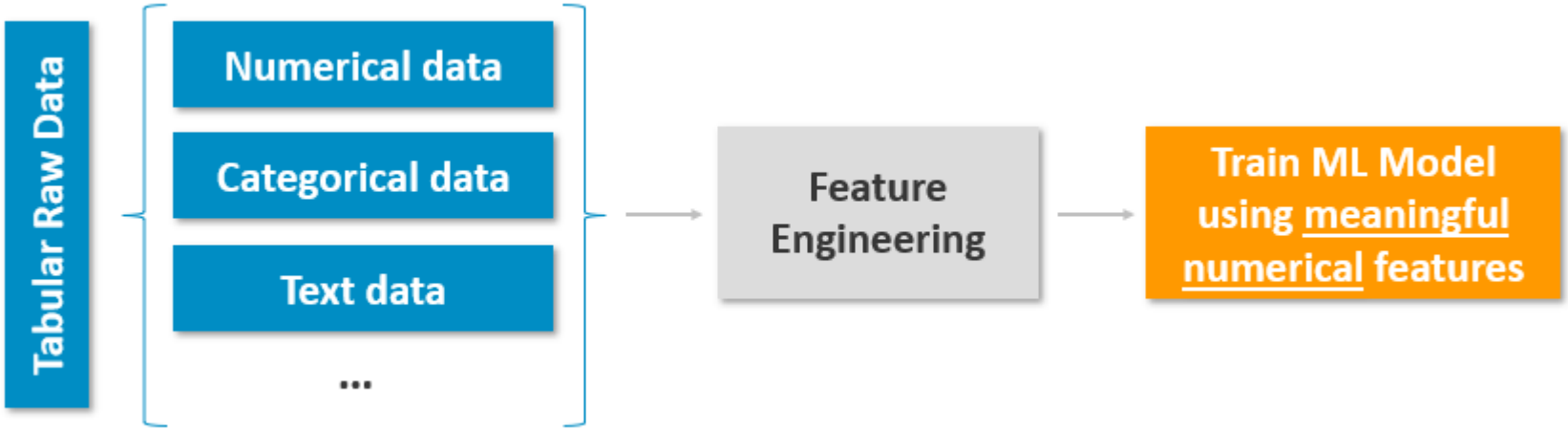
Ingeniería de características

UNIVERSIDAD EAFIT | Escuela de Ciencias Aplicadas e Ingeniería

Utiliza conocimiento del **dominio y de los datos** para crear **nuevas características a partir de los datos crudos** como entradas para los modelos de ML.

Intuición: ¿Qué información utilizaría un humano para hacer predicciones?

¡A menudo, este proceso es más un arte que una ciencia!



Codificación de características

UNIVERSIDAD EAFIT
Escuela de Ciencias Aplicadas e Ingeniería

Características categóricas: No tienen una representación numérica natural.
Ejemplo: color ∈ {green, red, blue}, fraude ∈ {falso, verdadero}
La mayoría de modelos de ML requieren convertir estas categorías a números.

Codificación (encoding): Asignar un número a cada categoría.
Ordinal: Las categorías están ordenadas, ejemplo, tamaño ∈ {L>M>S}. Podemos asignar los valores L->3, M->2, S->1.

Nominal: Las categorías no tienen un orden específico, ejemplo, color ∈ {green, red, blue}. Podemos asignar números de forma aleatoria.

Codificación de características

UNIVERSIDAD EAFIT
Escuela de Ciencias Aplicadas e Ingeniería

`LabelEncoder`: **Clase de sklearn**, codifica variables categóricas con valores entre 0 y n_classes – 1. Implementa los métodos: `.fit() y .transform()`

Codifica las variables objetivo o una sola característica, no la matriz completa X.

	color	size	price	classlabel
0	green	S	10.1	shirt
1	red	M	13.5	pants
2	blue	L	15.3	shirt

```
from sklearn.preprocessing import LabelEncoder

le = LabelEncoder()
df['color'] = le.fit_transform(df['color'])
```

```
   color size  price classlabel
0      1    S   10.1      shirt
1      2    M   13.5      pants
2      0    L   15.3      shirt
```

Codificación de características

UNIVERSIDAD EAFIT
Escuela de Ciencias Aplicadas e Ingeniería

`OrdinalEncoder`: **Clase de sklearn,** codifica variables categóricas con valores entre 0 y n_classes – 1. Implementa los métodos: `.fit() y .transform().`

- Codifica dos o más variables categóricas. No funciona para una sola característica.
- Devuelve una sola columna de enteros por característica.

	color	size	price	classlabel
0	green	S	10.1	shirt
1	red	M	13.5	pants
2	blue	L	15.3	shirt

```
from sklearn.preprocessing import OrdinalEncoder
oe = OrdinalEncoder()

df[['color','size','classlabel']] =
oe.fit_transform(df[['color','size','classlabel']])
```

	color	size	price	classlabel
0	1.0	2.0	10.1	1.0
1	2.0	1.0	13.5	0.0
2	0.0	0.0	15.3	1.0

Codificación de características

UNIVERSIDAD EAFIT
Escuela de Ciencias Aplicadas e Ingeniería

Codificación usando la variable objetivo: Codificar utilizando valores que puedan explicar la variable objetivo.

Ejemplo: Promediar la variable objetivo para cada categoría. Después, reemplazar los valores categóricos con los valores promediados.

x_1	x_2	y
a	c	1
a	d	1
b	c	0
a	d	0
a	d	0
a	d	1
b	d	0

x_1 -> cat a -> 3/5 = 0.6
x_1 -> cat b -> 0/2 = 0
x_2 -> cat c -> 1/2 = 0.5
x_2 -> cat d -> 2/5 = 0.4

x_1	x_2	y
0.6	0.5	1
0.6	0.4	1
0	0.5	0
0.6	0.4	0
0.6	0.4	0
0.6	0.4	1
0	0.4	0

Transformaciones en sklearn

UNIVERSIDAD EAFIT
Escuela de Ciencias Aplicadas e Ingeniería

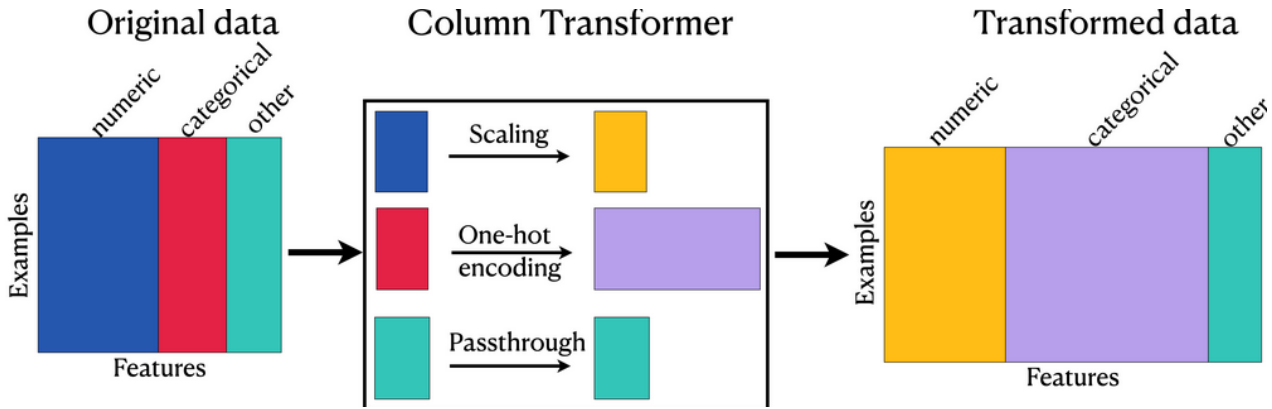
SimpleImputer, StandardScaler, MinMaxScaler, LabelEncoder, OrdinalEncoder, OneHotEncoder, CountVectorizer pertenecen a la clase Transformers de sklearn. Todos tienen los métodos:

- .fit(): aprende la transformación a partir de los datos de entrenamiento
- .transform() aplica la transformación a cualquier conjunto (entrenamiento, validación, prueba).
- En el conjunto de entrenamiento, también se puede aplicar .fit_transform()

ColumnTansformer en sklearn

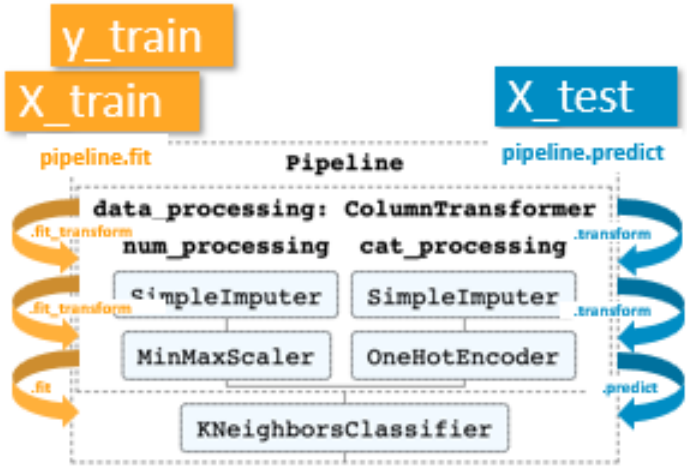
`ColumnTransformer` aplica las transformaciones a las columnas de un arreglo de Numpy o a un DataFrame de Pandas. Implementa los métodos `.fit()` y `.transform()`.

- Permite aplicar transformaciones a subconjuntos de características (numérica, categóricas, texto), de forma individual.
- Las características generadas por cada transformación se concatenarán para conformar un solo grupo de características



ColumnTranformer y Pipeline


```
numerical_processing = Pipeline([
    ('num_imputer', SimpleImputer(strategy='mean')),
    ('num_scaler', MinMaxScaler())])

categorical_processing = Pipeline([
    ('cat_imputer', Imputer(strategy='constant', fill_value='missing')),
    ('cat_encoder', OneHotEncoder(handle_unknown='ignore'))])

processor = ColumnTransformer(transformers =[
     ('num_processing', numerical_processing, ('feature1', 'feature3')),
     ('cat_processing', categorical_processing, ('feature0', 'feature2'))])

pipeline = Pipeline([('data_processing', processor),
                     ('estimator', KNeighborsClassifier())])

pipeline.fit(X_train, y_train)
predictions = pipeline.predict(X_test)
```



UNIVERSIDAD EAFIT

Escuela de Ciencias Aplicadas e Ingeniería

Regresión Lineal, Logística

Notación matemática

UNIVERSIDAD EAFIT | Escuela de Ciencias Aplicadas e Ingeniería

	longitude	latitude	housing_median_age	total_rooms	total_bedrooms	population	households	median_income	median_house_value	ocean_proximity
0	-122.23	37.88	41.0	880.0	129.0	322.0	126.0	8.3252	452600.0	NEAR BAY
1	-122.22	37.86	21.0	7099.0	1106.0	2401.0	1138.0	8.3014	358500.0	NEAR BAY
2	-122.24	37.85	52.0	1467.0	190.0	496.0	177.0	7.2574	352100.0	NEAR BAY
3	-122.25	37.85	52.0	1274.0	235.0	558.0	219.0	5.6431	341300.0	NEAR BAY
4	-122.25	37.85	52.0	1627.0	280.0	565.0	259.0	3.8462	342200.0	NEAR BAY

Vector de características

$$\mathbf{x}^{(i)} = \begin{bmatrix} x_1^{(i)} \\ x_2^{(i)} \\ \vdots \\ x_d^{(i)} \end{bmatrix} \in \Re^{d \times 1}$$

$$y^{(i)} \in \Re$$

$$y^{(i)} \in [0, 1, \cdots, C]$$

Matriz de características

$$\mathbf{X} = \begin{bmatrix} \cdots & \mathbf{x}^{(1)^\top} & \cdots \\ \cdots & \mathbf{x}^{(2)^\top} & \cdots \\ & \vdots & \\ \cdots & \mathbf{x}^{(n)^\top} & \cdots \end{bmatrix} \in \Re^{n \times d} \qquad \mathbf{y} = \begin{bmatrix} y^{(1)} \\ y^{(2)} \\ \vdots \\ y^{(n)} \end{bmatrix}$$

Métodos de aprendizaje de máquina

Formas de aproximar la relación entre $\mathbf{x}^{(i)}$ y $\mathrm{y}^{(i)}$

$$f(\mathbf{x}^{(i)}) = y^{(i)}$$

Función de costo

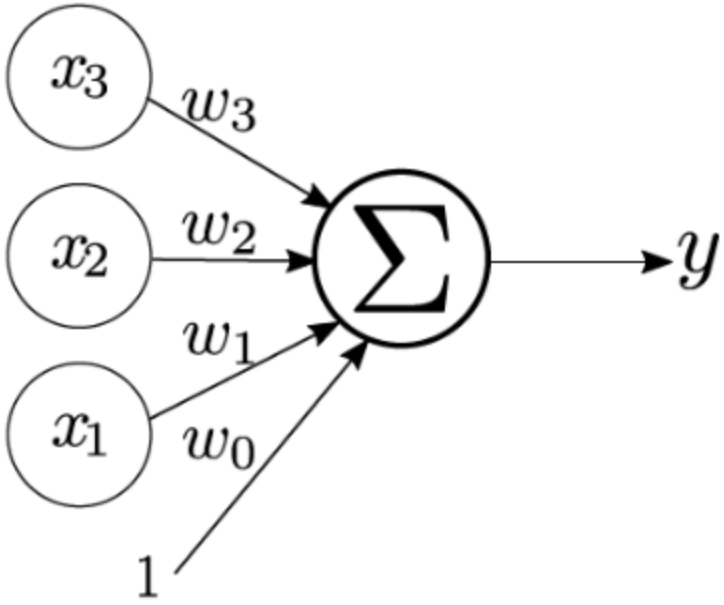
$$\mathcal{L}\left(f(\mathbf{x}^{(i)}), y^{(i)}\right)$$

Regresión Lineal: Modelo

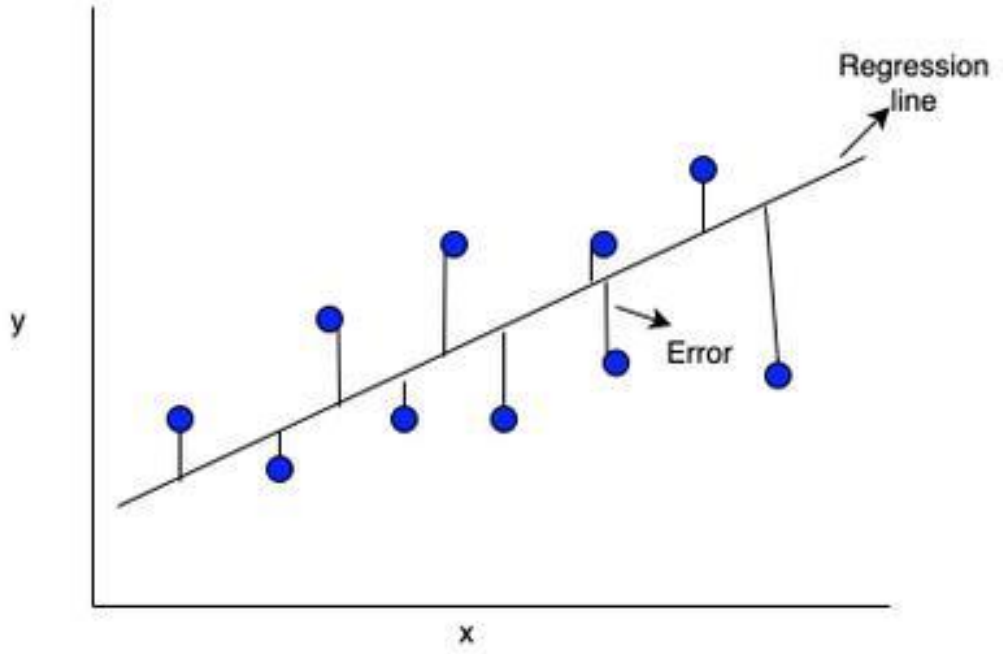
UNIVERSIDAD EAFIT
Escuela de Ciencias Aplicadas e Ingeniería

$$\hat{y} = \hat{f}(\mathbf{x}) = w_0 + \sum_{j=1}^{d} w_j \times x_j$$

$$\hat{y} = \mathbf{w}^T \begin{bmatrix} \mathbf{x} \\ 1 \end{bmatrix}$$



Regresión Lineal: Error en Función Objetivo



Regresión Lineal: Función Objetivo

$$Cost_{\mathbf{w}} = \frac{1}{m} \sum_{i=1}^{n} \left(y_i - \mathbf{w}^T \begin{bmatrix} \mathbf{x}_i \\ 1 \end{bmatrix} \right)^2$$

Regresión Lineal: Entrenamiento

UNIVERSIDAD EAFIT
Escuela de Ciencias Aplicadas e Ingeniería

Descenso por el gradiente:

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

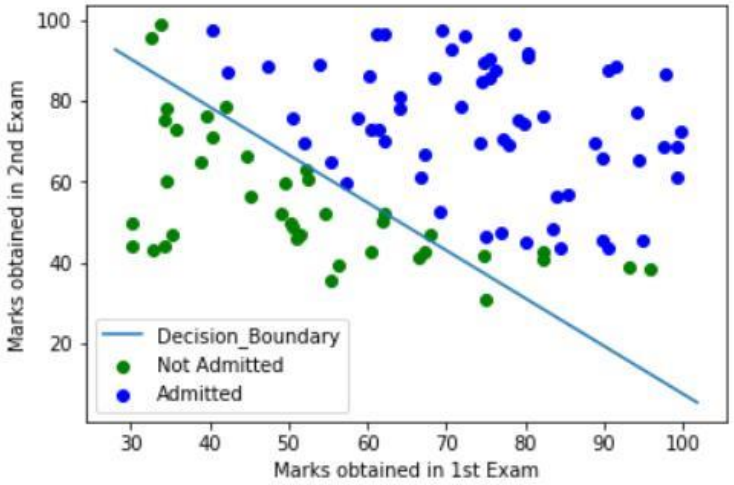
$$\nabla Cost_{\mathbf{w}} = \frac{1}{m} X^T (\hat{y} - y)$$

Ecuación normal:

$$\mathbf{w} = (\mathbf{X}^T \mathbf{X})^{-1} (\mathbf{X}^T Y)$$

Regresión Logística: Nombre

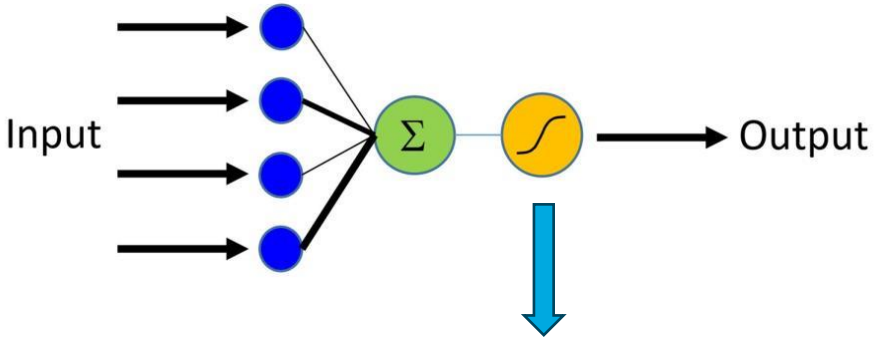
La regresión logística es un tipo de modelo lineal de **clasificación** a pesar de llevar históricamente la palabra regresión en su nombre.

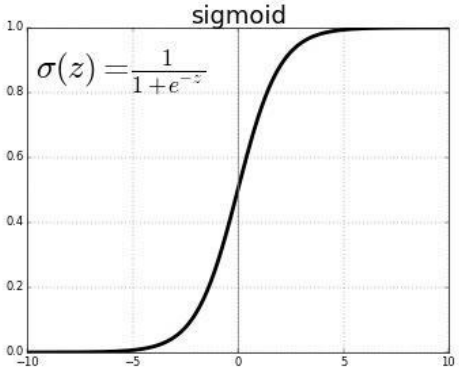


Regresión Logística: Modelo

UNIVERSIDAD EAFIT
Escuela de Ciencias Aplicadas e Ingeniería

$$\hat{y} = h(\mathbf{x}) = \frac{1}{1 + \exp\left(-\mathbf{w}^T \begin{bmatrix} \mathbf{x} \\ 1 \end{bmatrix}\right)}$$





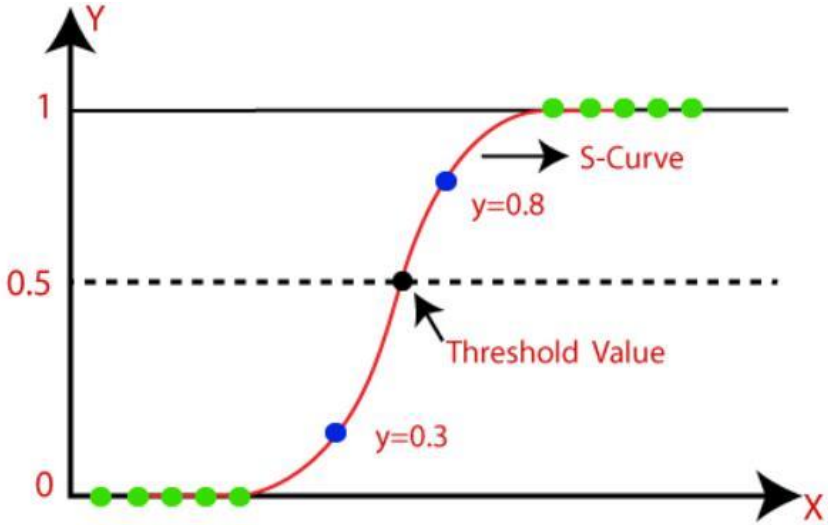
Regresión Logística: Etiqueta

La **etiqueta** con la que se entrena una regresión logística debe ser **binaria**, tomando únicamente los valores 0 o 1.

La clase 0 se suele denominar **clase negativa**, mientras que la clase 1 se suele llamar **clase positiva**.

Regresión Logística: Predicción

La salida que retorna una regresión logística es un número entre 0 y 1 que se interpreta como la probabilidad de que un ejemplo pertenezca a la clase positiva. La predicción final se decide según un umbral (por lo general 0.5)



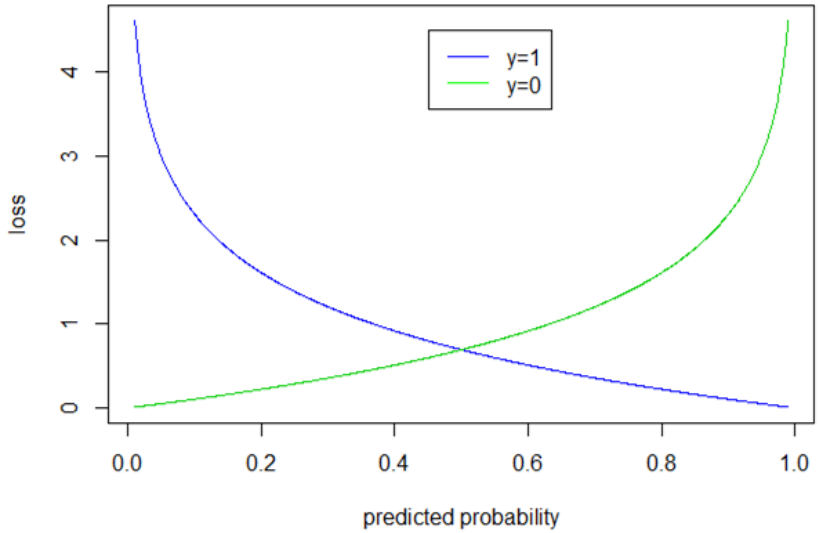
Regresión Logística: Función Objetivo

$$J(\theta) = \frac{1}{m}\sum_{i=1}^{m}[-y^{(i)}log(h_\theta(x^{(i)})) - (1-y^{(i)})log(1-h_\theta(x^{(i)}))]$$

$$\frac{\partial J(\theta)}{\partial \theta_j} = \frac{1}{m}\sum_{i=1}^{m}(h_\theta(x^{(i)}) - y^{(i)})x_j^{(i)}$$

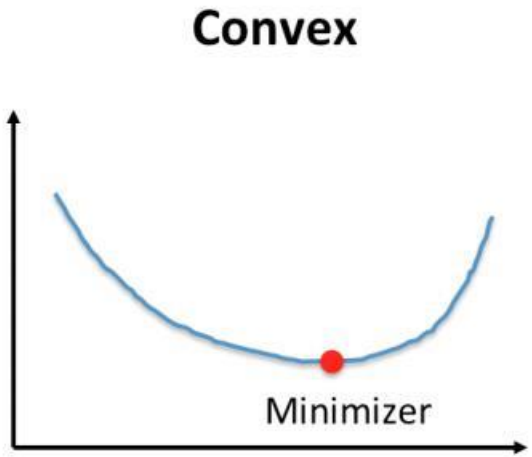
Regresión Logística: Función Objetivo

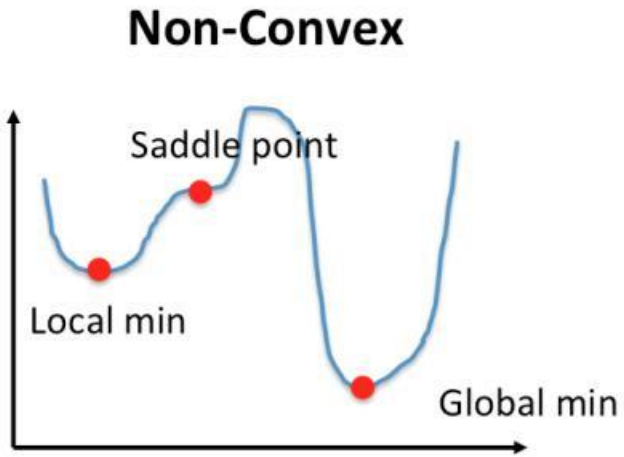
Esta función objetivo se denomina **pérdida de entropía cruzada (cross-entropy loss).**



Regresión Logística: Función Objetivo

La pérdida de entropía cruzada se utiliza porque permite que el problema de optimización para entrenar la regresión logística se vuelva **convexo.**





Regresión Logística: Entrenamiento

La regresión logística sólo se puede entrenar mediante métodos iterativos. Por lo general, se utiliza el descenso por el gradiente.

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

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

GRACIAS