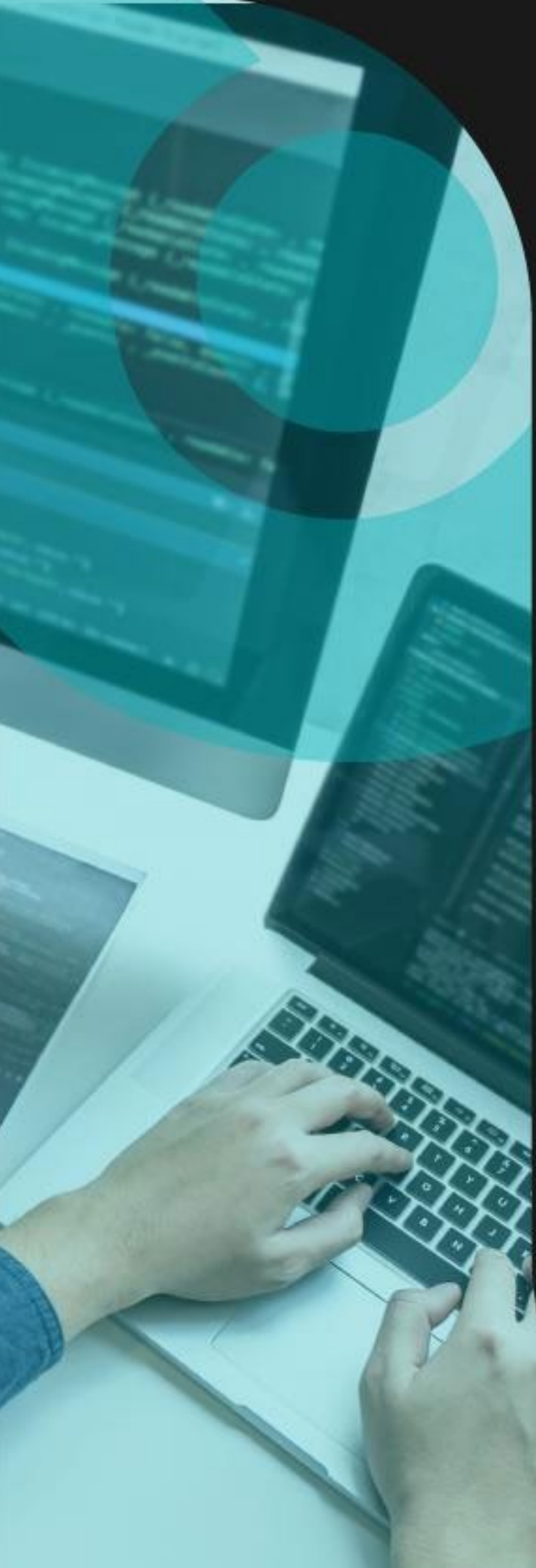


AWAKELAB

BASECAMP

Ciencia de Datos

Módulo: Aprendizaje de Máquina Supervisado

Aprendizaje Esperado

---

4. Elaborar un modelo predictivo aplicando el algoritmo de Regresión Logística para resolver un problema de clasificación utilizando el lenguaje Python.

Algoritmos de Clasificación

La clasificación se define como el proceso de reconocimiento, comprensión y agrupación de objetos e ideas en categorías preestablecidas, también conocidas como "subpoblaciones". Con la ayuda de estos conjuntos de datos de entrenamiento categorizados previamente, la clasificación en los programas de aprendizaje automático aprovecha una amplia gama de algoritmos para clasificar futuros conjuntos de datos en categorías respectivas y relevantes.

Problemas que requieren clasificación

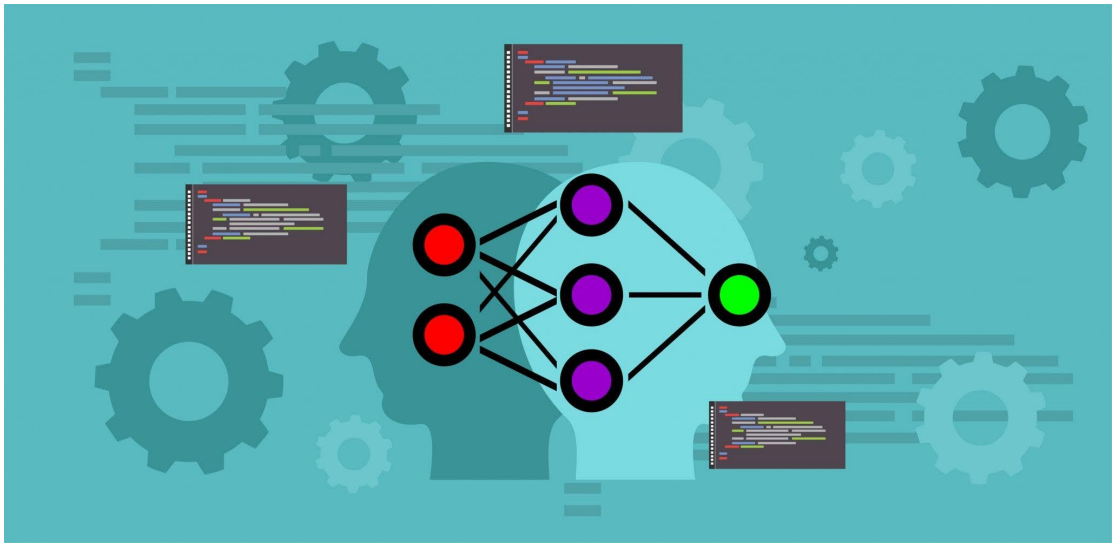
Basado en datos de entrenamiento, el algoritmo de clasificación es una técnica de aprendizaje supervisado que se utiliza para categorizar nuevas observaciones. En la clasificación, un programa utiliza el conjunto de datos o las observaciones proporcionadas para aprender a categorizar nuevas observaciones en varias clases o grupos. Por ejemplo, 0 o 1, rojo o azul, sí o no, spam o no spam, etc. Se pueden usar objetivos, etiquetas o categorías para describir clases. El algoritmo de clasificación utiliza datos de entrada etiquetados porque es una técnica de aprendizaje supervisado y comprende información de entrada y salida. Una función

de salida discreta (y) se transfiere a una variable de entrada en el proceso de clasificación (x).

En palabras simples, la clasificación es un tipo de reconocimiento de patrones en el que se realizan algoritmos de clasificación en datos de entrenamiento para descubrir el mismo patrón en nuevos conjuntos de datos.

Algoritmos más usados

Al realizar una clasificación en nuestros datos se puede realizar en datos estructurados o no estructurados. La clasificación es una técnica en la que categorizamos los datos en un número determinado de clases. El objetivo principal de un problema de clasificación es identificar la categoría/clase a la que pertenece un nuevo dato.



Algunos de los tópicos que estudiaremos en este módulo son:

- Regresión Logística
- K-Nearest Neighbors *
- Naives Bayes *
- Support Vector Machine *
- Decision Tree

- Random Forest *
- Gradient Boosting *

(*) Estos temas se estudiarán en profundidad en los aprendizajes siguientes de este módulo

Modelos Lineales Generalizados

Un Modelo Lineal Generalizado (GLM) es una generalización de una regresión lineal que permite utilizar variables provenientes de distribuciones distintas a la Normal como variable respuesta.

En un GLM la variable respuesta se conecta con un modelo lineal mediante una función llamada *función de enlace*, la cual se relaciona con la esperanza de la distribución. En términos simples, se busca una función para el valor esperado de la variable respuesta , tal que se pueda retornar a un modelo lineal clásico.

En particular, cuando el modelo tiene una respuesta que sigue una distribución Bernoulli y se utiliza la función de enlace logit, se denomina Regresión Logística. Este tiene por objetivo modelar cómo influye en la probabilidad de éxito de un suceso, la presencia o ausencia de otros factores.

Árbol de decisión

Corresponde a una técnica de Machine Learning de tipo Supervisado. Al final del entrenamiento, se obtienen secuencias de preguntas de los datos que llevan a un resultado o predicción. Un plus que tiene esta técnica es que es posible visualizar las secuencias de preguntas en forma de árbol, siendo fácil de explicar y comprender para la audiencia.

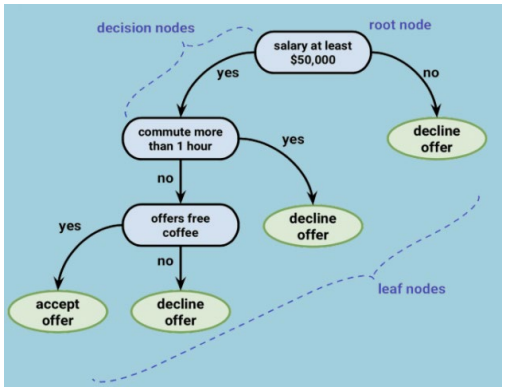
Los árboles de decisión se clasifican en:

- Árbol de Regresión: Cuando la variable output es numérica. Ejemplo: Determinar las secuencias de preguntas adecuadas de modo de predecir el salario de un colaborador.
- Árbol de Clasificación: Cuando la variable output es categórica. Ejemplo: Determinar las secuencias de preguntas adecuadas de modo de predecir el fallo o no fallo de un producto.

Ejemplo Árbol de Clasificación: ¿Debería aceptar un nuevo empleo?

Podemos observar las siguientes partes del árbol:

- Cada nodo de decisión (decision node) representa una pregunta sobre un atributo. El nodo raíz (root node) corresponde a la primera pregunta.
- Cada rama representa una posible respuesta a las preguntas.
- Cada hoja o nodo terminal (leaf node) representa la clase o predicción al seguir ese camino.

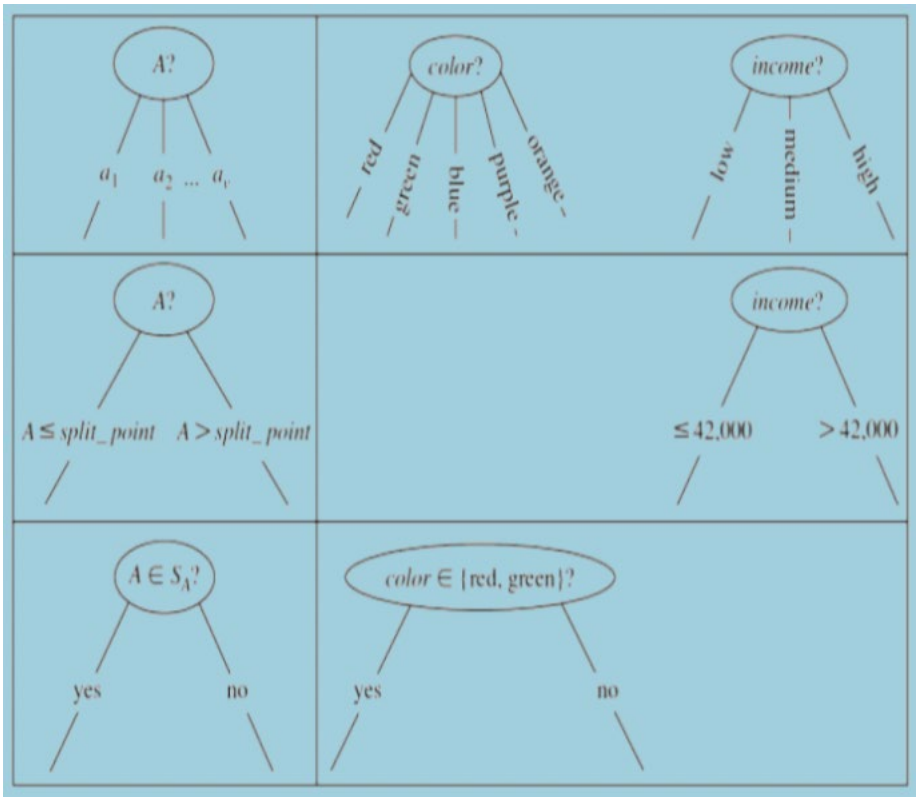


Árbol de clasificación: ¿Cómo se obtienen las preguntas? (*)

1. Si todas las observaciones pertenecen a la misma clase o categoría, se crea un nodo terminal con dicha categoría.
2. Si no, se recurre a algún método para determinar la primera pregunta o criterio de separación, la idea es obtener particiones puras (que los grupos estén en la menor medida posible mezclados, es decir más homogéneos).
3. Se crea el primer nodo el criterio de separación encontrado, este proceso se itera de forma de añadir ramas y nodos de forma anidada hasta que ocurra alguno de los siguientes casos:

- Todas las tuplas pertenecen a la misma clase.
- No hay más atributos para particionar.
- Ya no hay más datos.

(*) Se profundizará más de esto en el aprendizaje 6

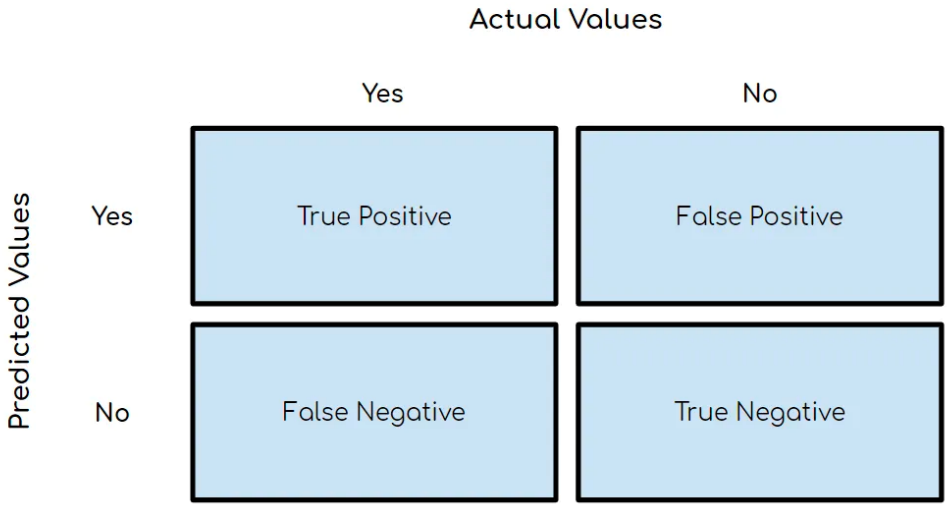


Métricas de Desempeño

¿Qué es una matriz de confusión?

Una matriz de confusión, también conocida como matriz de error, es una tabla resumida. Se utiliza para evaluar el rendimiento de un modelo de clasificación. El número de predicciones correctas e incorrectas se resume con valores de conteo y se desglosa por cada clase.

A continuación se muestra una imagen de la estructura de una matriz de confusión de 2x2. Para dar un ejemplo, digamos que hubo diez casos en los que un modelo de clasificación predijo 'Sí' en el que el valor real fue 'Sí'. Luego, el número diez iría en la esquina superior izquierda en el cuadrante Verdadero Positivo. Esto nos lleva a algunos términos clave:



- **Positivo (P) :** La observación es positiva (por ejemplo, es un perro).
- **Negativo (N) :** la observación no es positiva (por ejemplo, no es un perro).
- **Verdadero positivo (VP) :** resultado en el que el modelo predice correctamente la clase positiva.
- **Verdadero negativo (VN) :** resultado en el que el modelo predice correctamente la clase negativa.
- **Falso positivo (FP) :** también llamado error de tipo 1 , un resultado en el que el modelo predice incorrectamente la clase positiva cuando en realidad es negativa.
- **Falso negativo (FN) :** también llamado error de tipo 2 , un resultado en el que el modelo predice incorrectamente la clase negativa cuando en realidad es positiva.

Precisión

La precisión también se conoce como valor predictivo positivo y es la proporción de instancias relevantes entre las instancias recuperadas. En otras palabras, responde a la pregunta "¿Qué proporción de identificaciones positivas fue realmente correcta?"

$$precisión = \frac{VP}{VP + FP}$$

Exactitud

Esto es simplemente igual a la proporción de predicciones que el modelo clasificó correctamente.

$$accuracy = \frac{\#\, predicciones\ correctas}{\#\, todas\ las\ predicciones} = \frac{VP + VN}{VP + FP + VN + FN}$$

Sensibilidad

Es la proporción de casos positivos que fueron correctamente identificadas por el algoritmo.

$$sensitivity = \frac{VP}{VP + FN}$$

Especificidad

Se trata de los casos negativos que el algoritmo ha clasificado correctamente. Expresa cuán bien puede el modelo detectar esa clase.

$$specificity = \frac{VN}{FP + VN}$$

Regresión logística

Si una variable cualitativa con dos niveles se codifica como *1* y *0*, matemáticamente es posible ajustar un modelo de regresión lineal por mínimos cuadrados $\beta_0 + \beta_1 x$. El problema de esta aproximación es que, al tratarse de una recta, para valores extremos del predictor, se obtienen valores de Y menores que 0 o mayores que 1, lo que entra en contradicción con el hecho de que las probabilidades siempre están dentro del rango [0,1]. Entonces Para evitar estos problemas, la regresión logística transforma el valor devuelto por la regresión lineal $\beta_0 + \beta_1 x$ empleando una función cuyo resultado está siempre comprendido entre *0* y *1*.

La función de enlace que nos permite relacionar la esperanza de la distribución Bernoulli con los predictores es:

$$logit(\pi_{x_i}) = ln\ ln\left(\frac{\pi_{x_i}}{1-\pi_{x_i}}\right) = \beta_0 + \beta_1 x_1 + \cdots + \beta_k x_k$$

Si en lo anterior aplicamos la función exponencial en ambos lados tenemos que:

$$\left(\frac{\pi_{x_i}}{1-\pi_{x_i}}\right) = exp(\beta_0 + \beta_1 x_1 + \cdots + \beta_k x_k)$$

El término $\frac{\pi_{x_i}}{1-\pi_{x_i}}$ se conoce como chance (odds). Las chances representan la relación entre la probabilidad de ocurrencia y no ocurrencia de un éxito.

- Si las chances son mayores a 1 la probabilidad de ocurrencia del éxito es mayor a la probabilidad de fracaso (más probable que ocurra el éxito).
- Si las chances son menores a 1 la probabilidad de ocurrencia del éxito es menor a la probabilidad del fracaso (menos probable que ocurra el éxito).

La razón entre dos chances se conoce como Odds Ratio (OD), o simplemente razón de chances. Esta cuantifica cuánto más probable es la ocurrencia de un evento al aumentar en una unidad o categoría una variable X_i específica.

Sean $x_* = (1, x_1, \ldots, x_i + 1, \ldots, x_k)$ y $x_\circ = (1, x_1, \ldots, x_i + 1, \ldots, x_k)$ entonces la razón de chances está dada por:

$$OR = \left(\frac{\frac{\pi_{x_*}}{1-\pi_{x_*}}}{\frac{\pi_{x_\circ}}{1-\pi_{x_\circ}}}\right) = \frac{exp\ exp\left(\beta_0 + \beta_1 x_1 + \cdots + \beta_i(x_i+1) + \cdots + \beta_k x_k\right)}{exp\ exp\left(\beta_0 + \beta_1 x_1 + \cdots + \beta_i x_i + \cdots + \beta_k x_k\right)}$$
$$= exp\ exp\left(\beta_i\right)$$

AWAKELAB

Con esto los coeficientes se interpretan de la siguiente forma:

- Si $\beta_i > 0$, entonces $OR > 1$ y por tanto X_i es un factor de riesgo.
- Si $\beta_i = 0$, entonces $OR = 0$ y por tanto no hay efecto del factor X_i.
- Si $\beta_i < 0$, entonces $OR < 1$ y por tanto X_i es un factor protector.

Notar que si $\beta_i = 0$, entonces la probabilidad se mantendrá constante para cualquier valor de X_i.

¿Qué es la función sigmoidea?

La función sigmoidea es una función matemática que tiene una curva característica en forma de "S", que transforma los valores entre el rango 0 y 1. La función sigmoidea también se llama curva sigmoidea o función logística. Es una de las funciones de activación no lineal más utilizadas.
La expresión matemática para sigmoide:

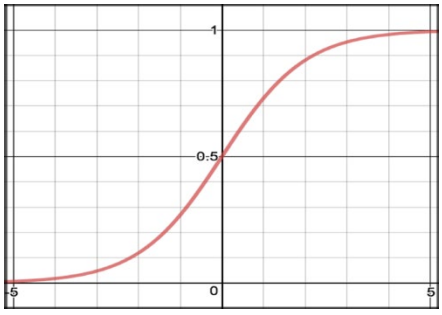
$$y = \frac{1}{1 + e^{-x}}$$

Luego la derivada es:

$$\frac{dy}{dx} = \frac{e^x}{(1 - e^x)^2}$$

Y la integral queda como:

$$\int y\,dx = x + ln\,ln\,(1 + e^{-x}) = ln\,ln\,(1 + e^x)$$



En el gráfico anterior, si el valor de x va a infinito positivo, entonces el valor predicho de y se convertirá en 1 y si va a infinito negativo, entonces el valor predicho de y se convertirá en 0.

Suponga que conoce la regresión logística, que es el algoritmo común utilizado para la clasificación binaria o cuando el valor de la variable objetivo es de naturaleza categórica. La función logit o sigmoidea se utiliza para predecir las probabilidades de un resultado binario.

Ventajas	Desventajas
La regresión logística es más fácil de implementar, interpretar y muy eficiente de entrenar.	Si el número de observaciones es menor que el número de características, no se debe utilizar la regresión logística, de lo contrario, puede provocar un sobreajuste.
No hace suposiciones acerca de las distribuciones de clases en el espacio de características.	Construye límites lineales.
Puede extenderse fácilmente a múltiples clases (regresión multinomial) y una vista probabilística natural de las predicciones de clase.	La principal limitación de la regresión logística es la suposición de linealidad entre la variable dependiente y las variables independientes.

No solo proporciona una medida de cuán apropiado es un predictor (tamaño del coeficiente), sino también su dirección de asociación (positiva o negativa).	Solo se puede usar para predecir funciones discretas. Por lo tanto, la variable dependiente de la regresión logística está ligada al conjunto de números discretos.
Es muy rápido en la clasificación de registros desconocidos.	Los problemas no lineales no se pueden resolver con regresión logística porque tiene una superficie de decisión lineal. Los datos linealmente separables rara vez se encuentran en escenarios del mundo real.
Buena precisión para muchos conjuntos de datos simples y funciona bien cuando el conjunto de datos es linealmente separable.	La regresión logística requiere multicolinealidad media o nula entre variables independientes.
Puede interpretar los coeficientes del modelo como indicadores de la importancia de las características.	Es difícil obtener relaciones complejas utilizando la regresión logística. Los algoritmos más potentes y compactos, como las redes neuronales, pueden superar fácilmente a este algoritmo.

La regresión logística es menos propensa al sobreajuste, pero puede hacerlo en conjuntos de datos de gran dimensión. Se pueden considerar técnicas de regularización (L1 y L2) para evitar el sobreajuste en estos escenarios.	En la regresión lineal, las variables independientes y dependientes se relacionan linealmente. Pero la regresión logística necesita que las variables independientes se relacionan linealmente con las probabilidades logarítmicas (log(p/(1-p)).

Implementación en Python

```
# import the class

from sklearn.linear_model import LogisticRegression


# instantiate the model (using the default
parameters)

logreg = LogisticRegression(random_state=16)


# fit the model with data

logreg.fit(X_train, y_train)


y_pred = logreg.predict(X_test)
```

Referencias

[1] Cassis, A. (2015). Inteligencia artificial 101.

https://inteligenciaartificial101.wordpress.com/2015/10/20/aprendizaje supervisado/

[2] Métodos de Clasificación

https://bookdown.org/content/2274/metodos-de-clasificacion.html

[3] Regresión Logística

https://conceptosclaros.com/que-es-regresion-logistica/

[4] Función Sigmoidea

https://ichi.pro/es/que-es-la-funcion-sigmoidea-como-se-implementa-en-regresion-logistica-77981969140323

Complementario:

[1] Algoritmos de clasificación - PDF

chrome-extension://efaidnbmnnnibpcajpcglclefindmkaj/viewer.html?pdfurl=https%3A%2F%2Fdata.unimooc.com%2Fmateriales-cursos%2Fmachine-learning%2FMachine-Learning-5.pdf&clen=175693&chunk=true

[2] Comparación entre algoritmos de clasificación y de regresión

https://www.youtube.com/watch?v=u9kchxQAelM

[3] Ventajas y desventajas de algoritmos de calcificación

https://www.youtube.com/watch?v=T8aCfSBlrqU