Modelos predictivos en analítica de datos

Francisco García

Analítica Predictiva

Es el uso de modelos estadísticos y algoritmos de machine learning sustentados en datos previos.

Objetivo: Encontrar patrones en los datos y usarlos para hacer predicciones

Toma de decisiones basada en datos en lugar de intuición.

Optimización de procesos → predicción de fallas, detección de fraudes, segmentación de clientes.

Aplicaciones en física → Modelado de fenómenos, clasificación de eventos, detección de anomalías en experimentos, reconstrucción de parámetros iniciales.

Índice

Fundamentos

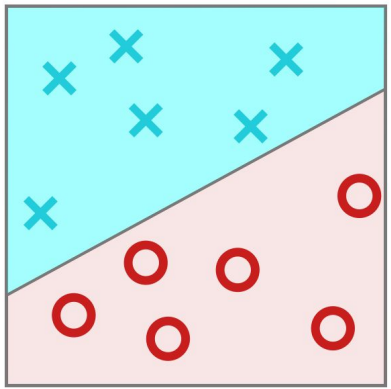
- Clasificación y regresión
- Métricas de desempeño
- Validación cruzada

Modelos:

- Regresión lineal y logística
- KNN
- Naïve Bayes
- Árboles de decisión
- Support Vector Machines

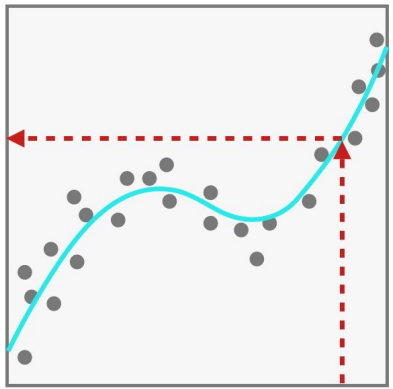
Clasificación vs. Regresión

Classification Groups observations into "classes"



Here, the line classifies the observations into X's and O's

Regression predicts a numeric value



Here, the fitted line provides a predicted output, if we give it an input

Clasificación:

- Variable objetivo categórica
- Si tiene 2 clases: Binaria/Dicotómica
- Ejemplos:
  - Hay o no hay fraude fiscal?
  - Tipo de partícula detectada.
  - Morfología de galaxias.

Regresión:

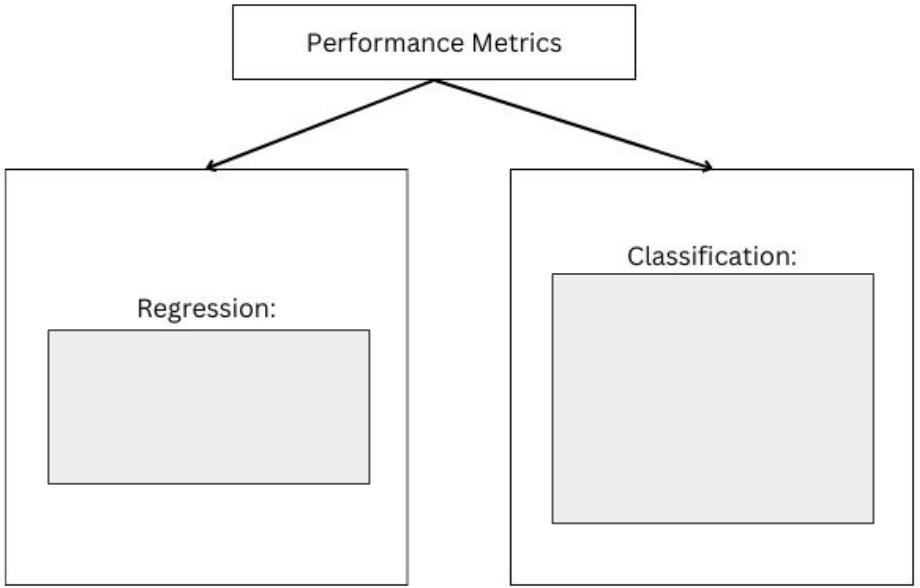
- Variable objetivo numérica
- Ejemplos:
  - Precio de venta de una propiedad inmueble.
  - Reconstrucción de energía primaria.
  - Conductividad eléctrica de un material a diferentes temperaturas.

Métricas de rendimiento

Nos ayudan a evaluar qué tan bien está funcionando un modelo.

Dependiendo del tipo de problema (clasificación o regresión), usamos métricas diferentes.

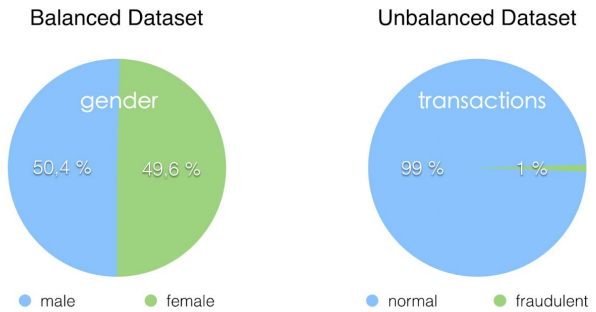
Elegir la métrica correcta es clave para interpretar los resultados correctamente.



Métricas de desempeño (Clasificación)

Accuracy/Tasa de precisión

- Proporción de predicciones correctas respecto al total.
- Es útil cuando las clases están correctamente balanceadas en el dataset.
- Desventaja: Pierde significado si una clase es más frecuente que otra, dando lugar a una mala interpretación de la validez de un modelo.



$$\text{Accuracy} = \frac{\text{Correct predictions}}{\text{All predictions}}$$

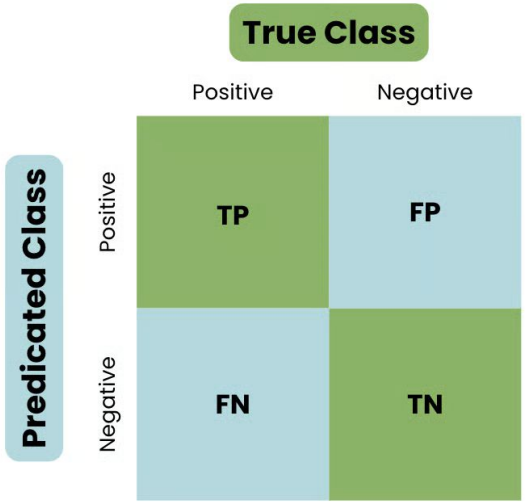
Métricas de desempeño (Clasificación)

Matriz de Confusión

- Compara las predicciones del modelo con los valores reales.

- Requiere definir cuál clase es la de mayor interés (crítica) y cuáles son de menor interés.
- Ejemplo: Diagnóstico de una enfermedad
  - Positivo: El paciente tiene la enfermedad.
  - Negativo: El paciente está sano.

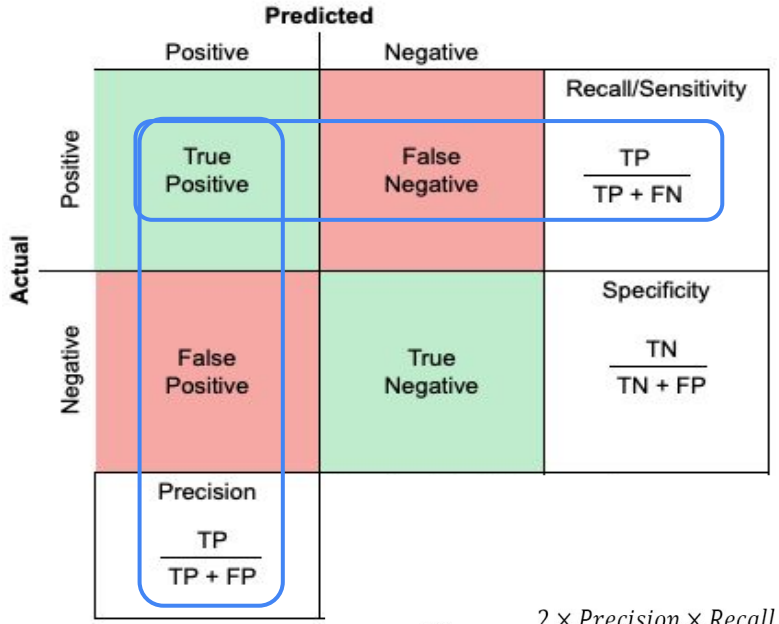
- Las predicciones correctas, las dividiremos entre verdaderos positivos y verdaderos negativos. Las incorrectas se dividen entre falsos positivos y falsos negativos.



Métricas de desempeño (Clasificación)

Métricas derivadas de la matriz de Confusión

- Precision:
  - Mide cuántos de los casos predichos como positivos realmente lo son.
  - No toma en cuenta casos negativos
  - Útil cuando los FP son costosos
- Recall (Sensitivity):
  - Mide qué porcentaje de los casos positivos fueron detectados.
  - Sensible a que el modelo simplemente clasifique todo como positivo
  - Útil cuando los FN son costosos
- Specificity:
  - Mide qué tan bien el modelo clasifica los casos negativos.
- F1 score
  - Media armónica entre Precisión y Recall.
  - Útil cuando nos interesan ambas métricas.
  - Funciona bien en data desbalanceada



$$F1 = \frac{2 \times Precision \times Recall}{Precision + Recall}$$

Métricas de desempeño (Clasificación)

ROC-AUC

En algunos modelos no obtenemos una clase u otra, sino la probabilidad de obtener una clasificación positiva.

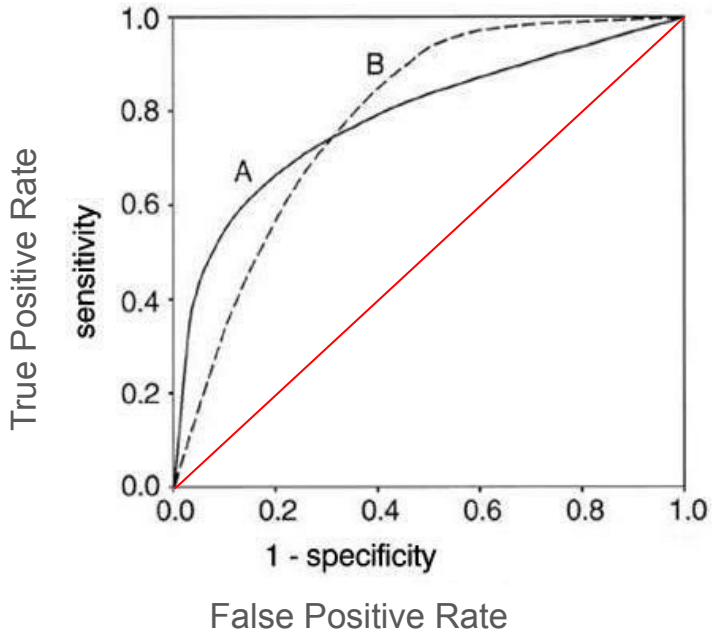
Podríamos definir que una probabilidad $P \geq 0.5$ equivale a clasificar un positivo, y $P < 0.5$ equivale a un negativo.

El umbral podría ser otro valor. y en función de eso cambiará la matriz de confusión del modelo.

Podemos caracterizar un modelo según los pares (1 - Specificity ; Sensitivity) que se obtienen al variar el umbral entre 0 y 1. → Curva ROC

El área bajo esta curva (AUC) se puede usar como métrica.

- AUC = 1 : Modelo clasifica bien
- AUC = 0.5 : Modelo equivalente a adivinar
- AUC = 0 : Modelo clasifica completamente mal



Métricas de desempeño (Regresión)

MSE, RMSE y MAE

MAE:

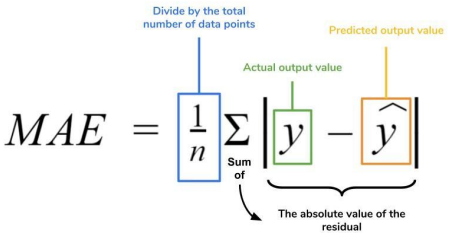
- Promedio de los errores absolutos.
- No amplifica grandes errores.

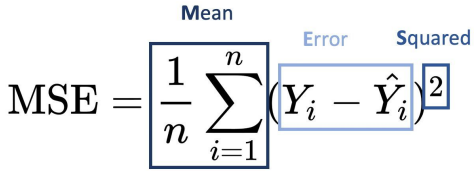
MSE:

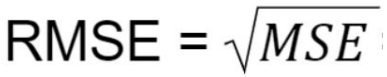
- Promedio de los errores al cuadrado.
- Penaliza más los errores grandes, útil cuando estos son críticos.
- Las unidades de esta métrica son el cuadrado de la variable objetivo.

RMSE:

- Raíz cuadrada del MSE.
- Tiene las mismas ventajas que el MSE.
- Tiene las mismas unidades que la variable objetivo.







Métricas de desempeño (Regresión)

R2 y Adjusted-R2

- Evalúan qué tan bien un modelo de regresión explica la variabilidad de los datos.
- Comparan el modelo con una referencia básica: la media de los datos.
- R2 = 1 significa que el modelo explica perfectamente la variabilidad de los datos.

$$R^2 = 1 - \frac{\sum_{i=1}^{n} (Y_i - \hat{Y}_i)^2}{\sum_{i=1}^{n} (Y_i - \overline{Y}_i)^2}$$

- Podríamos agregar variables irrelevantes a la regresión y que con eso, el modelo busque usarlas para reducir el error. Subiendo artificialmente el R2.
- Para tener esto en consideración se usa el R2-Ajustado.

$$R^2_{ajust} = 1 - (1 - R^2)\frac{n-1}{n-p-1}$$

n: total de datos
p: grados de libertad

En general deberíamos usar estas dos métricas para comparar modelos con la misma estructura y no para determinar si un modelo es bueno o malo por sí solo.

Validación Cruzada

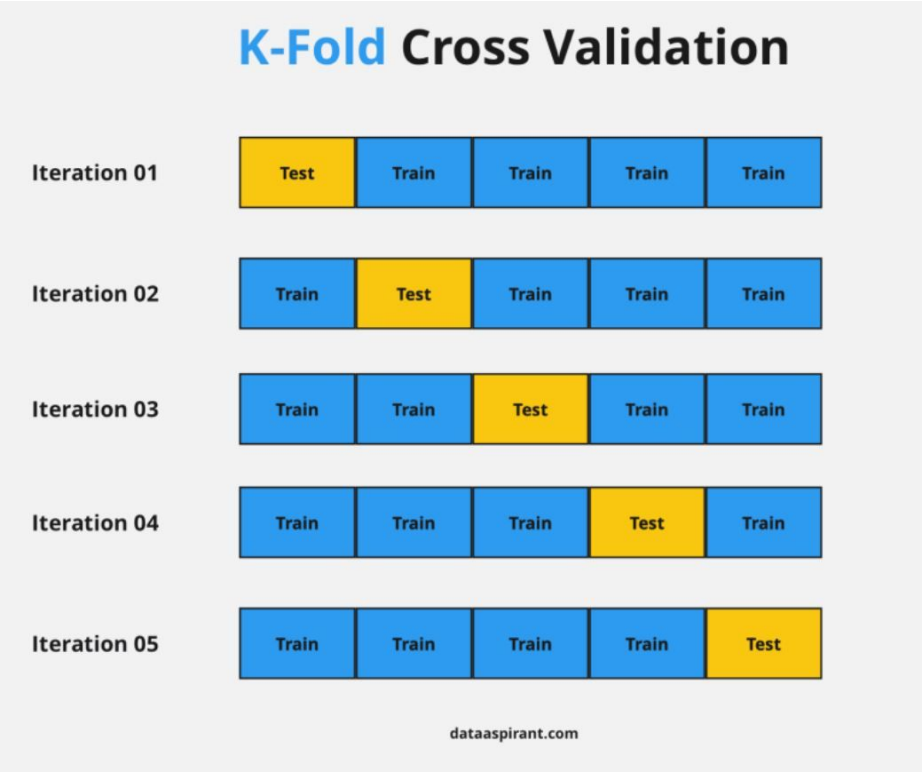
A veces podemos caer en el error de entrenar a un modelo para funcionar perfectamente con nuestro dataset. Pero esto no garantiza que funcione bien con datos nuevos. → Sobreajuste

La validación cruzada permite evitar esto.

Consiste en partir el dataset, entrenar a varios modelos y usarlos para construir un modelo final.

Esto además permite considerar la variabilidad de las métricas de rendimiento.

Hay diferentes tipos de validación cruzada: K-Fold, Leave-One-Out, Stratified K-Fold, …



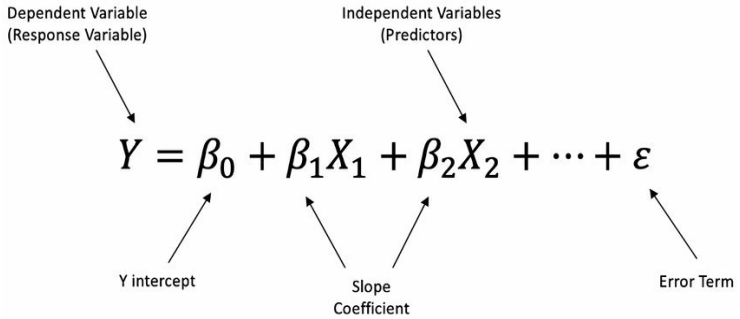
Regresión lineal múltiple

Definición

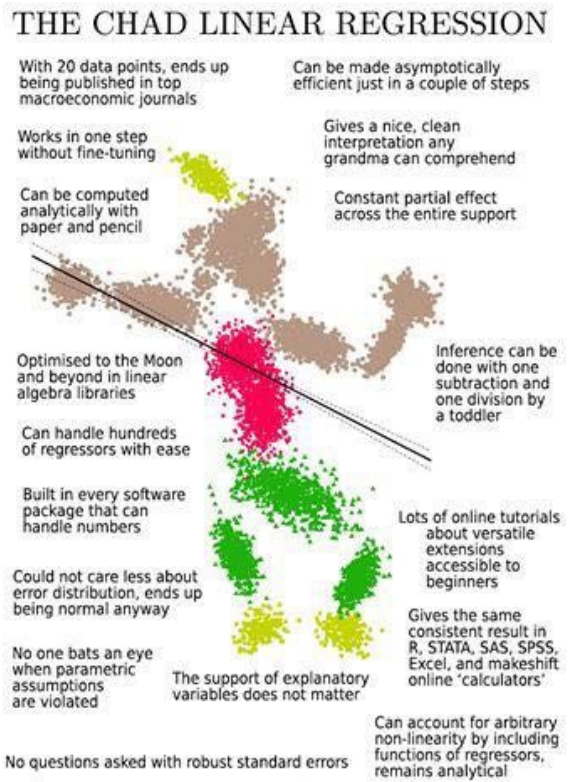
Target numérica.

Es una extensión de la regresión lineal simple donde se usan múltiples variables predictoras para estimar la variable target.

El objetivo es encontrar una combinación lineal de las variables predictoras que minimice el error y describa la relación entre estas y la target.

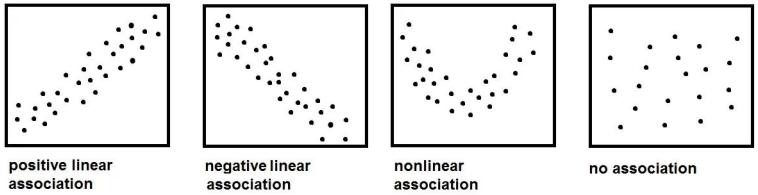


$$\hat{Y} = \beta_0 + \beta_1 X_1 + \beta_2 X_2 + \cdots$$



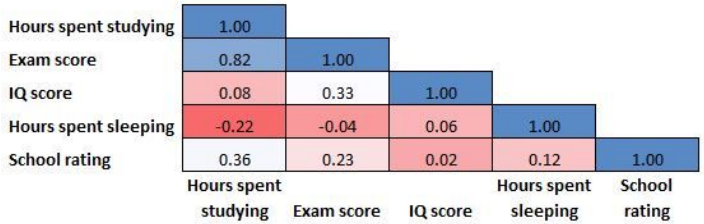
Regresión lineal múltiple

Suposiciones de validez del modelo



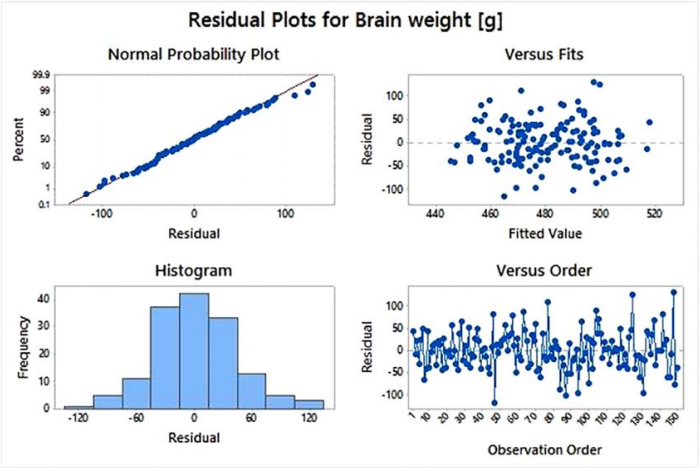
Variables

- La relación entre la variable dependiente y las variables predictoras debe ser lineal.
- No Colinealidad entre Variables Predictoras.



Errores:

- Normalidad: Los errores deben seguir una distribución normal con media 0.
- Independencia: Los errores deben ser independientes entre sí / No debe haber autocorrelación.
- Homocedasticidad: La distribución de errores debe tener varianza constante



Regresión lineal múltiple

Pros y Contras

:)

Es fácil de interpretar.

Ayuda a identificar relaciones funcionales entre múltiples variables y la variable respuesta.

Puede manejar predictores continuos y categóricos.

Es eficiente computacionalmente.

Funciona con datasets pequeños.

Se pueden aplicar técnicas como eliminación hacia atrás (Backward Elimination) o selección progresiva (Forward Selection) para elegir las variables más relevantes (P-value < 0.05).

:(

No siempre existe una relación lineal entre las predictoras y la target. No puede capturar interacciones no lineales ni efectos complejos en los datos.

Es muy sensible a valores atípicos, ya que estos pueden afectar significativamente los coeficientes estimados.

Muchos supuestos por validar.

Regresión logística

Definición

Target categórica binaria. ~~Regresión.~~ Clasificación

Busca predecir la probabilidad (P) de clasificar la target como positiva.

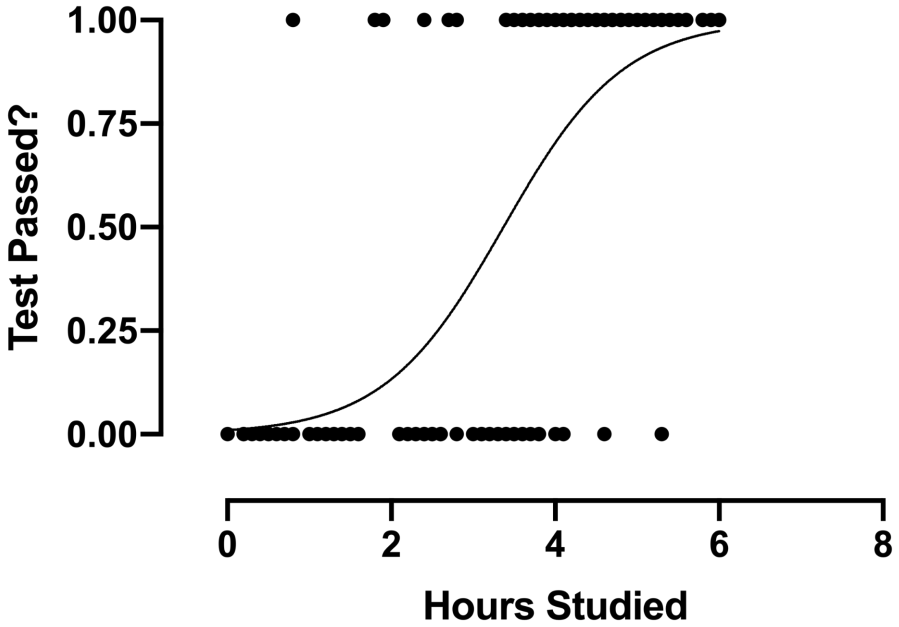
Parte de una transformación de la regresión lineal.

- Definimos los odds como: $odds = \frac{P}{1-P}$
- Asumimos que el Ln(odds) sigue una relación lineal con las variables predictoras

$$log(\frac{p}{1-p}) = \beta_0 + \beta_1 X$$

- Despejando P se obtiene el sigmoide:

$$P = \frac{1}{1 + e^{-(\beta_0 + \beta_1 X)}}$$



Regresión logística

Suposiciones de validez

No debe haber colinealidad entre variables predictoras.

Las observaciones deben ser independientes.

Regresión logística

Pros y Contras

:)

- Los mismos que en la regresión lineal + el hecho de tener menos supuestos para validar.

:(

- Supone que la relación entre cada predictora y el log(odds) es lineal.
- No maneja bien datos desbalanceados.
- No captura relaciones complejas en los datos.
- Sensible a outliers.

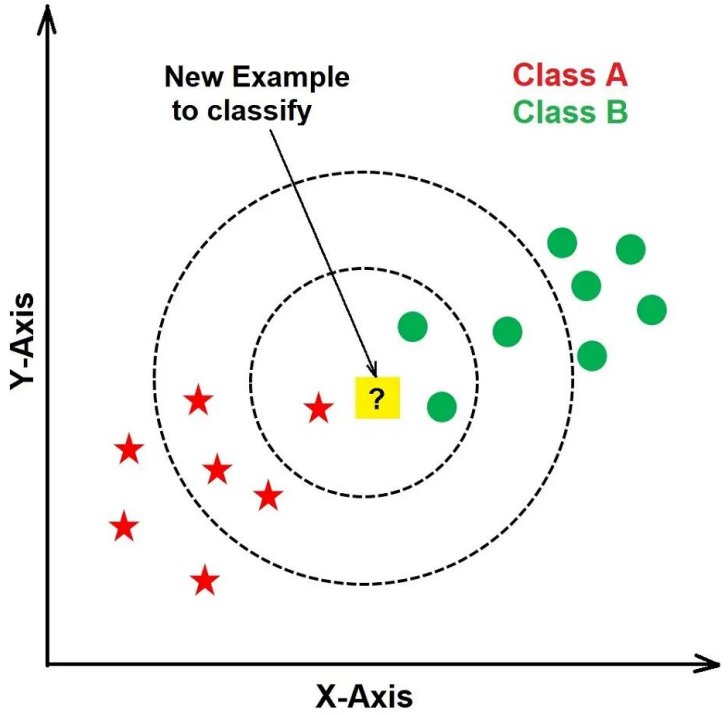
K-Nearest-Neighbors

Definición

Modelo de clasificación y regresión

Se basa en predecir según las k muestras más cercanas en el espacio de características.

No asume una relación funcional entre la variable de entrada y la salida, sino que asume que el nuevo datapoint se comportará como los datapoints que se le parecen.



K-Nearest-Neighbors

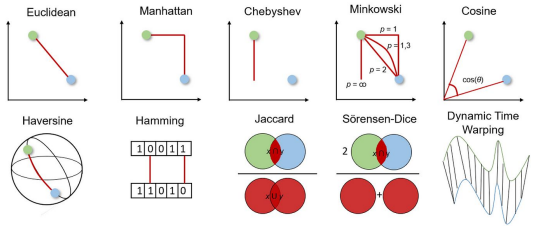
Algoritmo

1️⃣ Seleccionar K: Número de vecinos a considerar.

- K impar para evitar empates.
- Un K pequeño captura mejor patrones locales, pero es sensible a ruido y valores atípicos.
- un K grande reduce el ruido, pero lleva a pérdida de detalles.
- Como regla general k = sqrt(n).
- Es mejor usar probar varios K hasta minimizar el error.

2️⃣ Calcular la distancia entre el nuevo punto y todos los puntos de entrenamiento.

- Distancias comunes:
  - Euclidiana
  - Manhattan
  - Distancia Coseno



3️⃣ Seleccionar los K vecinos más cercanos.

- Escoger las observaciones con menor distancia al nuevo datapoint

4️⃣ Predecir

- Clasificación:
  - Se toma la clase más frecuente entre los vecinos.
- Regresión:
  - Se toma la media de los vecinos.

Existe una variación que pondera utilizando las distancias al momento de predecir.

K-Nearest-Neighbors

Pros y contras

:)

Simple e intuitivo.

No requiere entrenamiento (solo almacena datos).

No asume ninguna relación previa entre los predictores y la target.

Funciona bien en problemas con distribución clara y bien separada.

:(

Costoso computacionalmente. Debe calcular distancias con todos los puntos.

Para un buen desempeño hay que seleccionar bien K.

Sensible a datos desbalanceados.

Afectado por la escala de los datos.

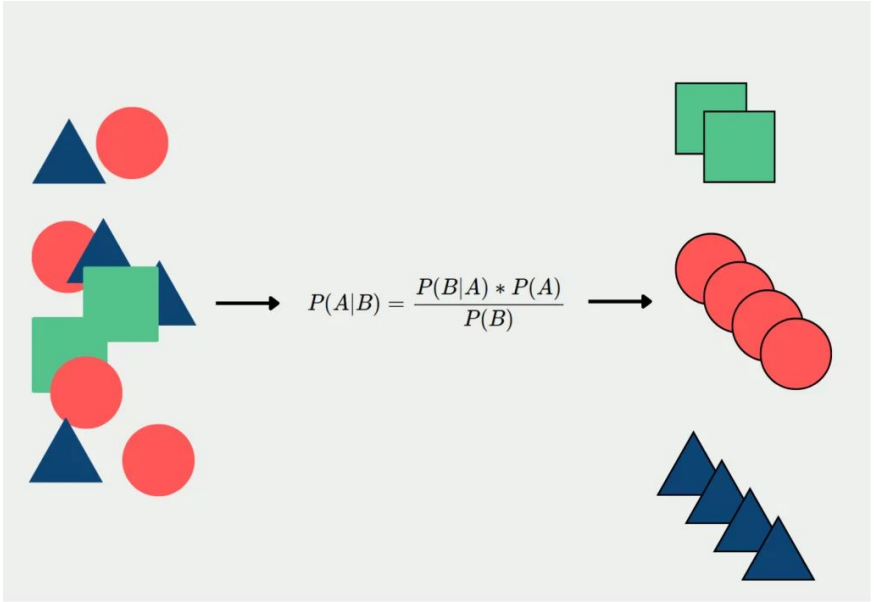
Naïve Bayes

Definición

Modelo de clasificación basado en el teorema de Bayes.

Asume que cada variable predictora contribuye de manera independiente a la probabilidad de la variable target. (Naïve)

Cual es la probabilidad de que mi nueva observación sea clasificada de una forma dado que tiene estas características?



Naïve Bayes

Teorema de bayes y algoritmo

Determina la probabilidad de algo, dadas ciertas condiciones.

Likelihood Prior

$$P(C|X) = \frac{P(X|C)P(C)}{P(X)}$$

Posterior Evidence

Ya que asumimos independencia. El likelihood se puede calcular como un producto de varios likelihoods.

$$P(X_1, X_2, \ldots, X_n|C) = P(X_1|C)P(X_2|C)\ldots P(X_n|C)$$

Como es imposible calcular la evidencia del vector de características, y ya que este es el mismo para cualquier clase de la variable objetivo, podemos tomarlo como un factor de proporcionalidad y obviarlo.

$$P(C|X) \propto P(C)\prod_{i=1}^{n} P(X_i|C)$$

El algoritmo de predicción consiste en calcular P(Ck|X) para todas las clases Ck y escoger la que haga máximo este valor. AKA la que tenga mayor probabilidad condicional.

$$\hat{C} = \arg\max_{C_k} P(C_k)\prod_{i=1}^{n} P(X_i|C_k)$$

Naïve Bayes

Ejemplo

Variable target

- **C: Enfermo/ Sano**

Variables predictoras

- **X1: Fiebre/No fiebre**
- **X2: Sarpullido/ No sarpullido**

Para este ejemplo, el sujeto no tiene fiebre, pero sí sarpullido

Calculamos las probabilidades de estar enfermo y de no estar enfermo

Probabilidad de estar enfermo:

P(Enfermo | No fiebre, Sarpullido) ∝ P(Enfermo) P(No fiebre| Enfermo) P(Sarpullido| Enfermo)

Probabilidad de estar sano:

P(Sano | No fiebre, Sarpullido) ∝ P(Sano) P(No fiebre| Sano) P(Sarpullido| Sano)

Y escogemos la clase con mayor probabilidad condicional!

Naïve Bayes

Pros y contras

:)

Funciona bien con grandes volúmenes de datos y es computacionalmente ligero.

Se entrena rápido, incluso con muchos atributos.

Funciona muy bien con datos categóricos.

La suposición de independencia es un poco flexible.

Maneja bien características irrelevantes.

:(

En la vida real no es tan común encontrar data que valide el supuesto de independencia.

Para predictoras numéricas se requiere un preprocesamiento.

Es sensible a datos poco representativos.

Es sensible en el caso de encontrar una likelihood con valor 0

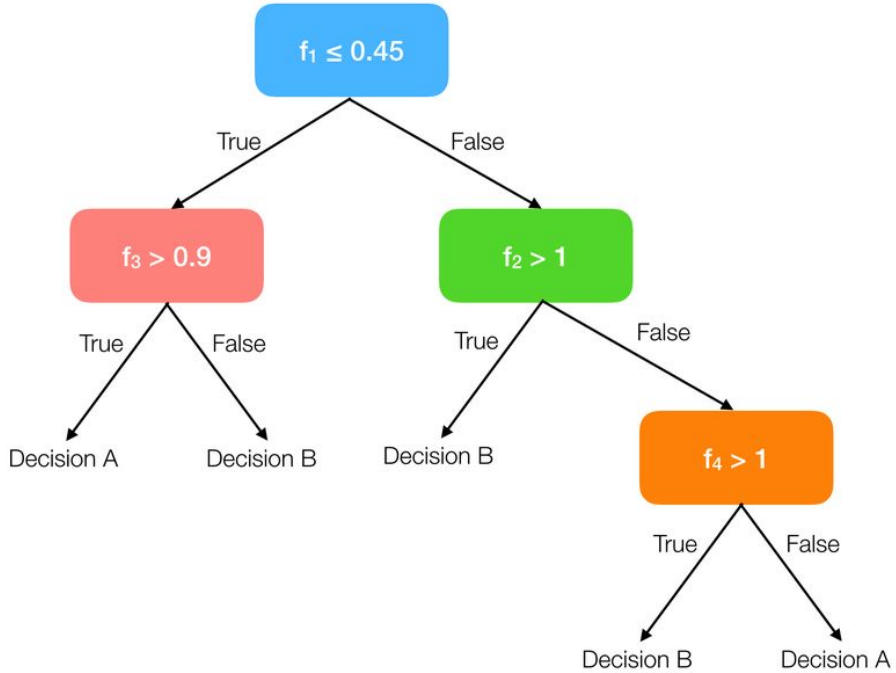
Decision Trees

Definición

Modelo de aprendizaje supervisado que divide iterativamente los datos en subconjuntos más pequeños hasta alcanzar una decisión final.

Estructura:

- Nodo raíz: Punto de inicio del árbol.
- Ramas: Divisiones de los nodos.
- Nodos: Representan decisiones basadas en condiciones sobre las variables.
- Hojas: Resultados finales de la clasificación o predicción.



Decision Trees

Algoritmo

En un nodo, escoger alguna variable y punto de corte para dividir los datos. (Ejemplo: llueve/no llueve, edad>23/edad<=23,etc.)

Evaluar esta división en función de alguna métrica. La división siempre debe buscar hacer más puros los nodos resultantes.

Probar con otras variables y puntos de corte, y evaluar.

Seleccionar la mejor forma de dividir los datos.

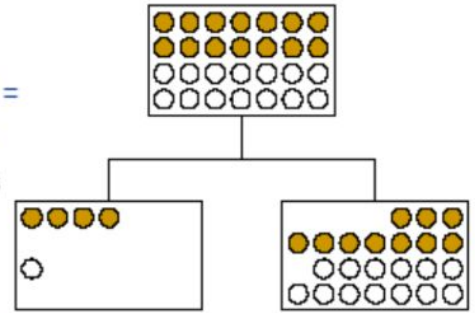
Dividir los datos

Repetir

Comparing the Splits with Entropy

Second Split:

- Entropy for left child is
  -1*((1/5)*log(1/5) + (4/5)log(4/5)) =
  -1*(-0.4644 + -0.2575) = 0.7219.
- Entropy for the right child is
  -1*((10/23)*log(10/23) +
  (13/23)log(13/23)) =
  -1*(-0.5225 + -0.4652) = 0.9877.
- Entropy for the split is
  (5/28)*Entropy$_{left}$+(23/28)*Entropy$_{right}$ =
  0.9402.



$$Var(Y) = \frac{1}{N}\sum_{i=1}^{N}(y_i - \bar{y})^2$$

$$Var_{split} = \frac{N_L}{N}Var_L + \frac{N_R}{N}Var_R$$

$$Gini = 1 - \sum_{i=1}^{C} p_i^2$$

$$Gini_{split} = \frac{N_L}{N}Gini_L + \frac{N_R}{N}Gini_R$$

$$S = -\sum_{i=1}^{C} p_i \log_2(p_i)$$

$$S_{split} = \frac{N_L}{N}S_L + \frac{N_R}{N}S_R$$

Decision Trees

Pros y contras

:)

Fácil de interpretar y visualizar.

No necesita normalización de datos.

Maneja bien datos mixtos.

No requiere supuestos estadísticos fuertes

Selecciona automáticamente las características más relevantes.

La aplicación es eficiente computacionalmente.

:(

Alto riesgo de sobreajuste. (Se soluciona con la poda y bosques aleatorios)

Sensibles a pequeñas variaciones en los datos.

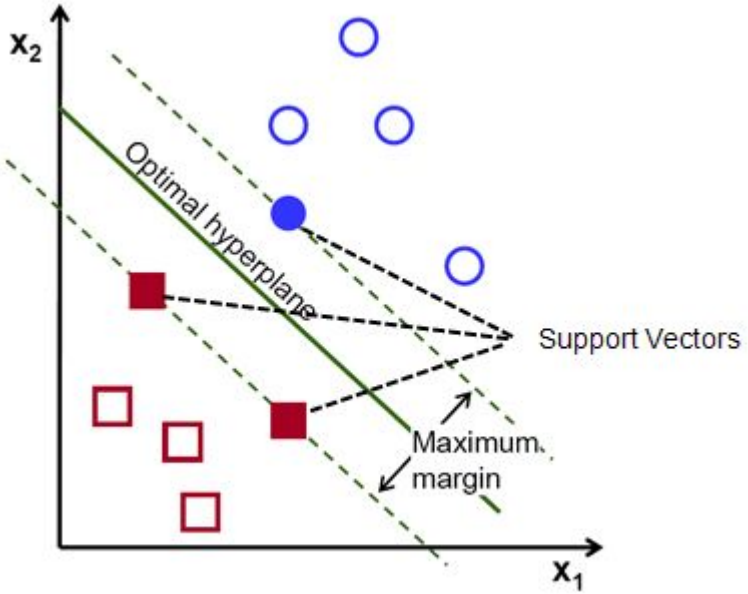
No destaca por su precisión.

Support Vector Machines

Definición

Busca encontrar un hiperplano óptimo que separe las clases en el espacio de características con el mayor margen posible.

Los puntos de datos más cercanos al hiperplano se llaman vectores de soporte, y son los que determinan la frontera de decisión.

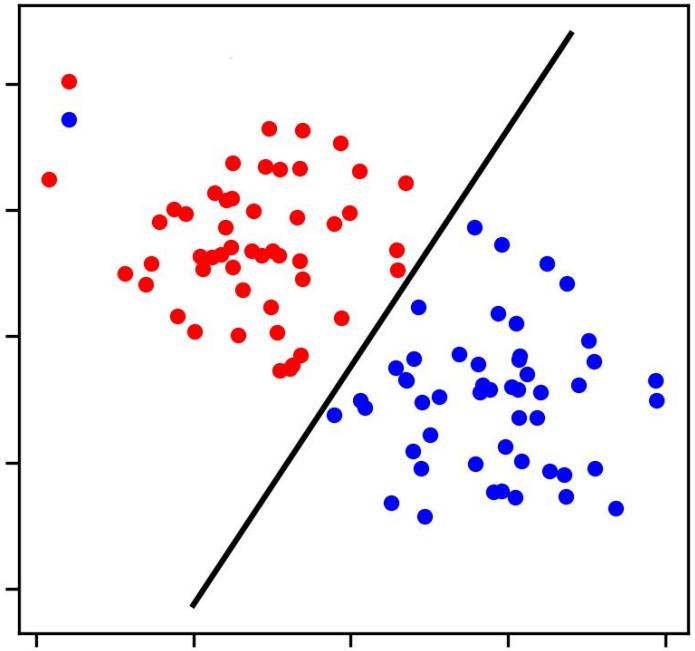


Support Vector Machines

Casos Linealmente (No) Separables

Caso ideal (Separabilidad lineal): Si los datos son perfectamente separables, el hiperplano se define claramente.

Caso realista (Datos traslapados): Es imposible hallar un hiperplano que separe perfectamente las clases, por lo cual se admite una cierta tolerancia (Soft Margin SVM).



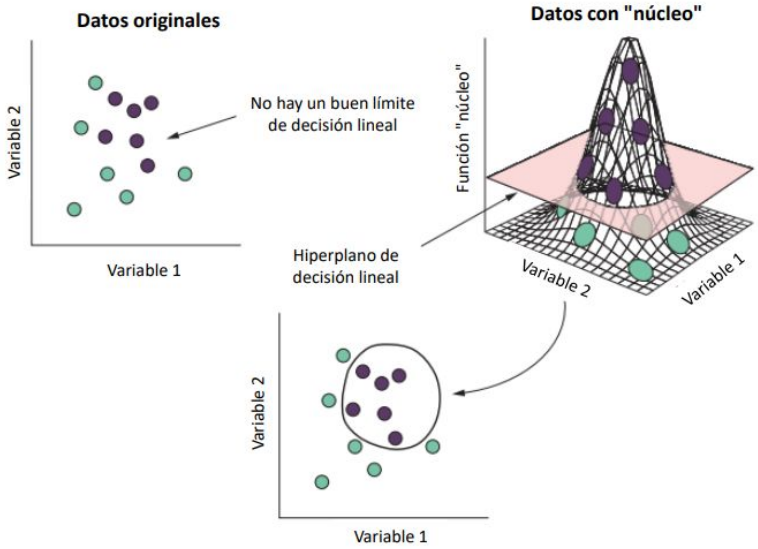
Support Vector Machines

Kernel

Agregar una nueva dimensión a los datos. Permite separar clases cuando no son linealmente separables (Igual puede ser necesario un margen suave).

Tipos de kernel:

Kernel	Fórmula
Lineal	$K(\mathbf{x}_i, \mathbf{x}_j) = \mathbf{x}_i \cdot \mathbf{x}_j$
Polinómico	$K(\mathbf{x}_i, \mathbf{x}_j) = (\mathbf{x}_i \cdot \mathbf{x}_j + c)^d$
Radial (RBF)	$K(\mathbf{x}_i, \mathbf{x}_j) = \exp\left(-\gamma \lVert \mathbf{x}_i - \mathbf{x}_j \rVert^2\right)$
Sigmoidal	$K(\mathbf{x}_i, \mathbf{x}_j) = \tanh(\alpha \mathbf{x}_i \cdot \mathbf{x}_j + \beta)$



Support Vector Machines

Hiperparámetros

Parámetro de kernel

Define la transformación del espacio de características.

Parámetro de grado

Grado del kernel polinomial

Parámetro gamma

- Controla el balance entre maximizar el margen y minimizar los errores.
- Valores altos → Límite de decisión granular.
- Valores bajos → Límite de decisión menos granular.

Parámetro de costo

- Controla la influencia de un solo punto en la clasificación. No existe para el kernel lineal.
- Valores altos → Penaliza errores más fuertemente.
- Valores bajos → Margen más suave y amplio.

Support Vector Machines

Pros y contras

:)

Funciona bien para datasets con muchas variables predictoras.

No requiere mucha data de entrenamiento.

No sobreajusta (con los hiperparámetros adecuados).
y no es sensible al ruido.

Útil para datos con fronteras definidas.

:(

No funciona bien cuando las clases están mezcladas en el espacio de características.

Es costoso a nivel computacional.

No ofrece una interpretación probabilística de la clasificación.