Ciencia de Datos
Modelos de Clasificación

OSCAR ANDRES GASPAR ALVAREZ

Modelos supervisados

Objetivo

- Modelos de Clasificación

Modelos supervisados

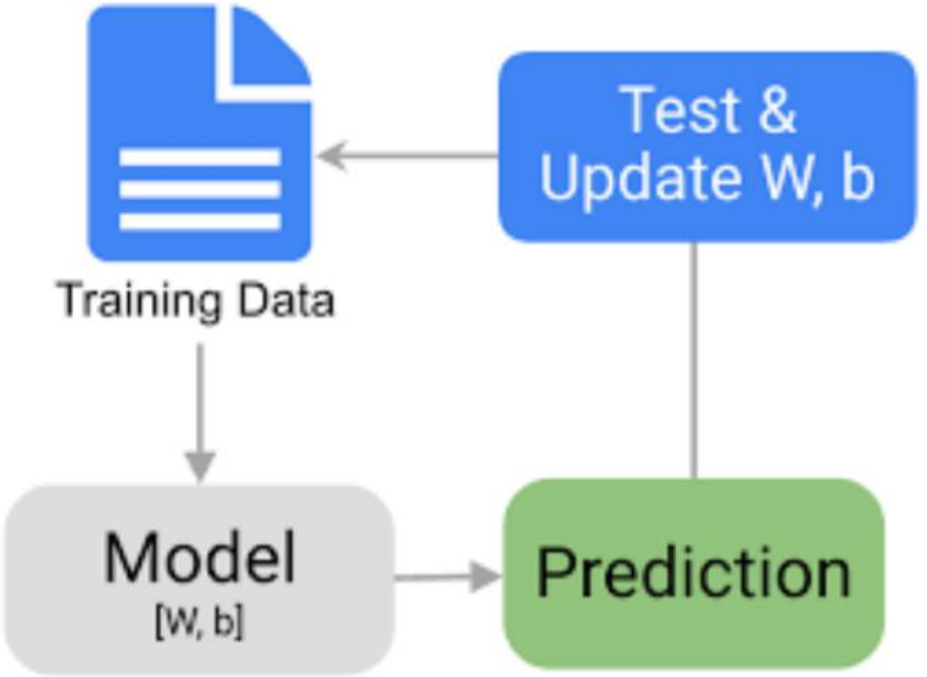
- Modelos supervisado son aquellos en los que producen funciones a partir de un conjunto de ejemplos de los que conocemos la salida deseada (dependiendo del tipo de salida, suele darse una subcategoría que diferencia entre modelos de clasificación, si la salida es un valor categórico, y modelos de regresión, si la salida es un valor de un espacio continuo)

Modelos de clasificación

- La clasificación supervisada es una de las tares que más frecuentemente son llevadas a cabo por los denominados Sistemas Inteligentes.
- Número de paradigmas desarrollados bien por la Estadística (Regresión Logística, Análisis Discriminante) o bien por la Inteligencia Artificial (Redes Neuronales, Inducción de Reglas, Árboles de Decisión, Redes Bayesianas) son capaces de realizar las tareas propias de la clasificación.

Entrenamiento del modelo

- El proceso de entrenamiento de un modelo de ML consiste en proporcionar datos de entrenamiento de los cuales aprender a un algoritmo de ML (es decir, el algoritmo de aprendizaje). El término modelo de ML se refiere al artefacto de modelo que se crea en el proceso de entrenamiento



Procedimiento común

- Dividimos data en : 70% para entrenar, validar y 30% para Probar

Estrategias de validación

Validación simple

El método más sencillo de validación consiste en repartir aleatoriamente las observaciones disponibles en dos grupos, uno se emplea para entrenar al modelo y otro para evaluarlo. Si bien es la opción más simple, tiene dos problemas importantes:

- La estimación del error es altamente variable dependiendo de qué observaciones se incluyan como conunto de entrenamiento y cuáles como conjunto de validación (problema de varianza).

- Al excluir parte de las observaciones disponibles como datos de entrenamiento (generalmente el 20%), se dispone de menos información con la que entrenar el modelo y, por lo tanto, se reduce su capacidad. Esto suele tener como consecuencia una sobrestimación del error comparado al que se obtendría si se emplearan todas las observaciones para el entrenamiento (problema de bias).

Leave One Out Cross-Validation (LOOCV)

- El método LOOCV en un método iterativo que se inicia empleando como conjunto de entrenamiento todas las observaciones disponibles excepto una, que se excluye para emplearla como validación. Si se emplea una única observación para calcular el error, este varía mucho dependiendo de qué observación se haya seleccionado. Para evitarlo, el proceso se repite tantas veces como observaciones disponibles, excluyendo en cada iteración una observación distinta, ajustando el modelo con el resto y calculando el error con dicha observación. Finalmente, el error estimado por el LOOCV es el promedio de todos lo i errores calculados.
- El método LOOCV permite reducir la variabilidad que se origina si se divide aleatoriamente las observaciones únicamente en dos grupos
- La principal desventaja de este método es su coste computacional.
- LOOCV es un método de validación muy extendido ya que puede aplicarse para evaluar cualquier tipo de modelo. Sin embargo, los autores de An Introduction to Statistical Learning consideran que, al emplearse todas las observaciones como entrenamiento, se puede estar cayendo en overfitting, por lo que, aun considerándolo muy aceptable, recomiendan emplear K-Fold Cross-Validation

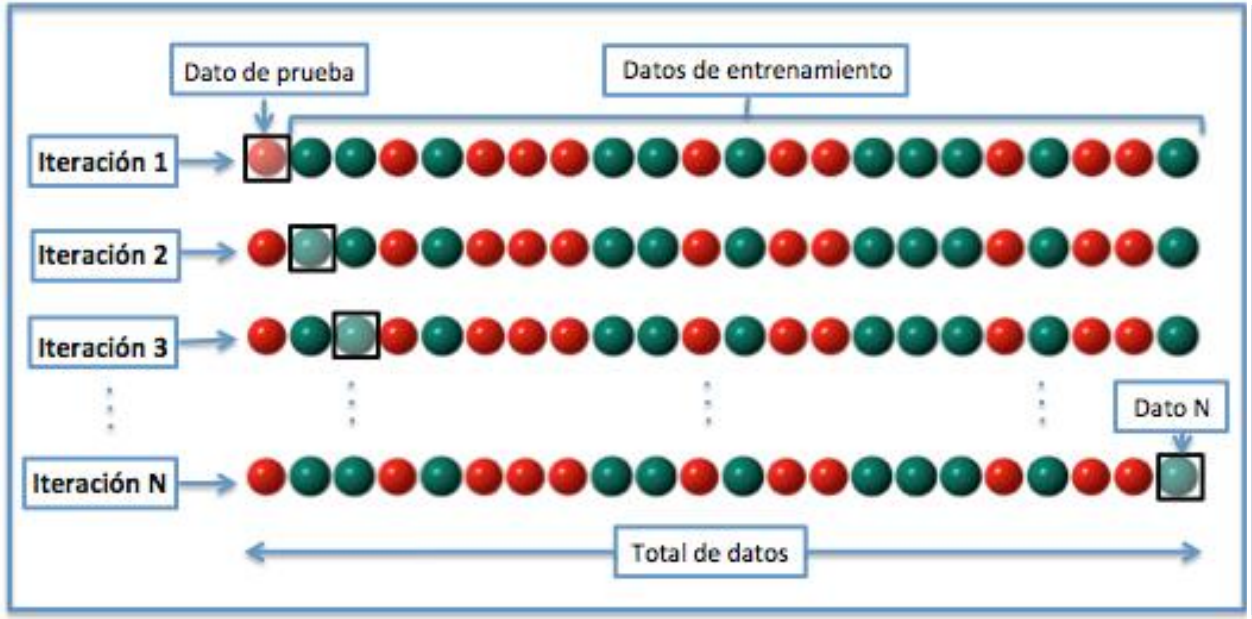
K-Fold Cross-Validation

- El método K-Fold Cross-Validation es también un proceso iterativo. Consiste en dividir los datos de forma aleatoria en k grupos de aproximadamente el mismo tamaño, k-1 grupos se emplean para entrenar el modelo y uno de los grupos se emplea como validación. Este proceso se repite k veces utilizando un grupo distinto como validación en cada iteración. El proceso genera k estimaciones del error cuyo promedio se emplea como estimación final.

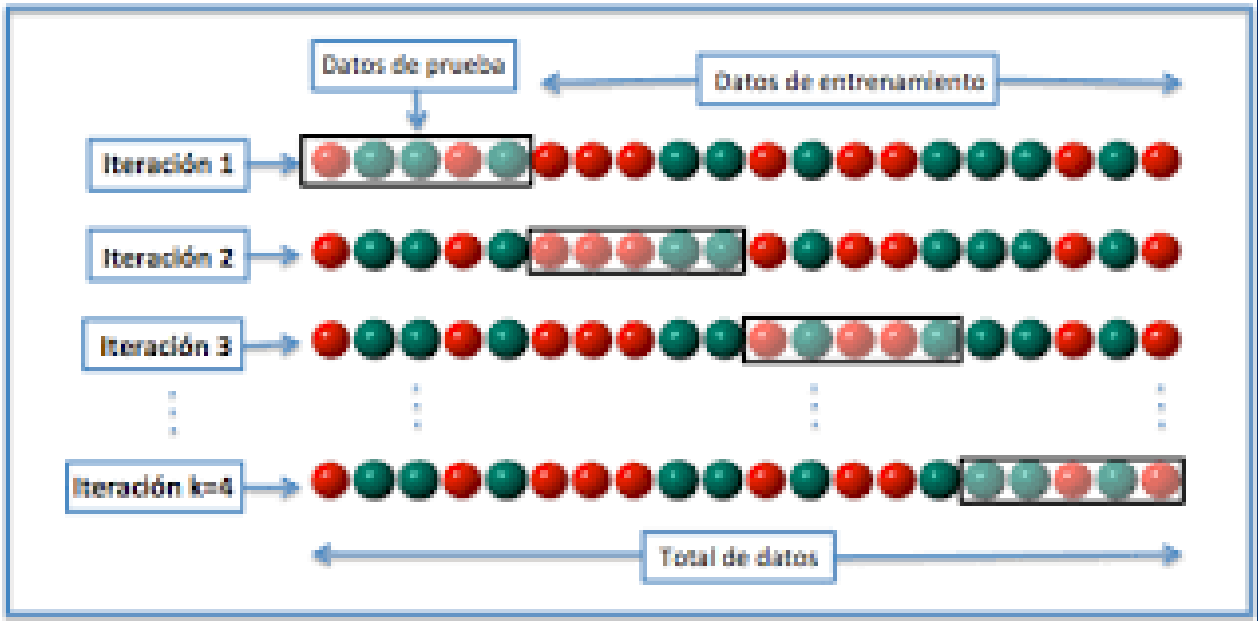
Dos ventajas del método K-Fold Cross-Validation frente al LOOCV:

- Requerimientos computacionales: el número de iteraciones necesarias viene determinado por el valor k escogido. Por lo general, se recomienda un k entre 5 y 10
- Balance entre bias y varianza: la principal ventaja de K-fold CV es que consigue una estimación precisa del error de test gracias a un mejor balance entre bias y varianza

Leave One Out

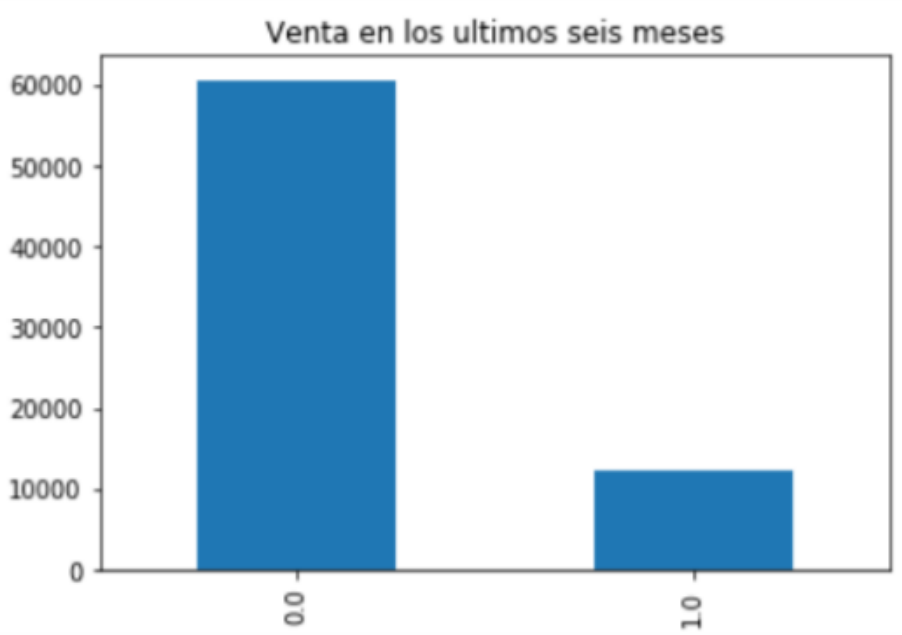


K-Fold Cross-Validation



Clases desbalanceadas

- “solemos encontrar que en nuestro conjunto de datos de entrenamiento contamos con que alguna de las clases de muestra es una clase «minoritaria» es decir, de la cual tenemos muy poquitas muestras. Esto provoca un desbalanceo en los datos que utilizaremos para el entrenamiento de nuestra máquina.”

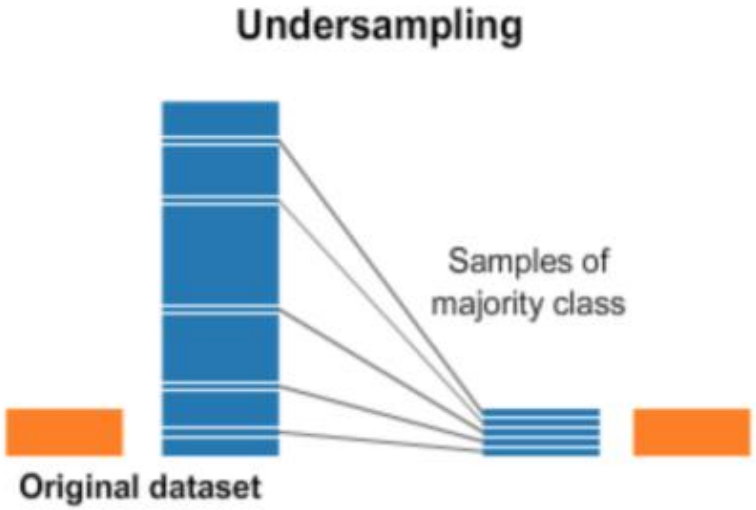


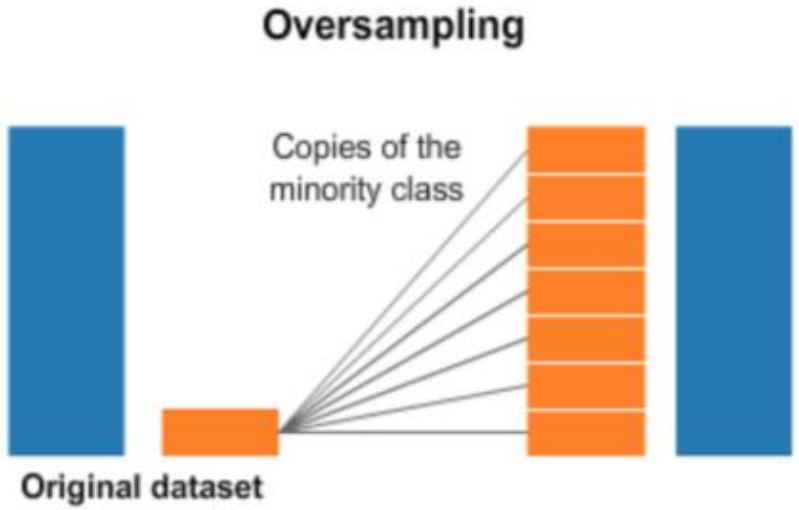
¿Cómo nos afectan los datos desbalanceados?

- “Por lo general afecta a los algoritmos en su proceso de generalización de la información y perjudicando a las clases minoritarias. Esto suena bastante razonable: si a una red neuronal le damos 990 de fotos de gatitos y sólo 10 de perros, no podemos pretender que logre diferenciar una clase de otra. Lo más probable que la red se limite a responder siempre «tu foto es un gato» puesto que así tuvo un acierto del 99% en su fase de entrenamiento.”

Estrategias para el manejo de Datos Desbalanceados:

- Ajuste de Parámetros del modelo: Consiste en ajustar parámetros ó métricas del propio algoritmo para intentar equilibrar a la clase minoritaria penalizando a la clase mayoritaria durante el entrenamiento.
- Modificar el Dataset: podemos eliminar muestras de la clase mayoritaria para reducirlo e intentar equilibrar la situación. Tiene como «peligroso» que podemos prescindir de muestras importantes, que brindan información y por lo tanto empeorar el modelo.
- Muestras artificiales: podemos intentar crear muestras sintéticas (no idénticas) utilizando diversos algoritmos que intentan seguir la tendencia del grupo minoritario. Según el método, podemos mejorar los resultados. Lo peligroso de crear muestras sintéticas es que podemos alterar la distribución «natural» de esa clase y confundir al modelo en su clasificación.
- •Balanced Ensemble Methods: Utiliza las ventajas de hacer ensamble de métodos, es decir, entrenar diversos modelos y entre todos obtener el resultado final (por ejemplo «votando») pero se asegura de tomar muestras de entrenamiento equilibradas.





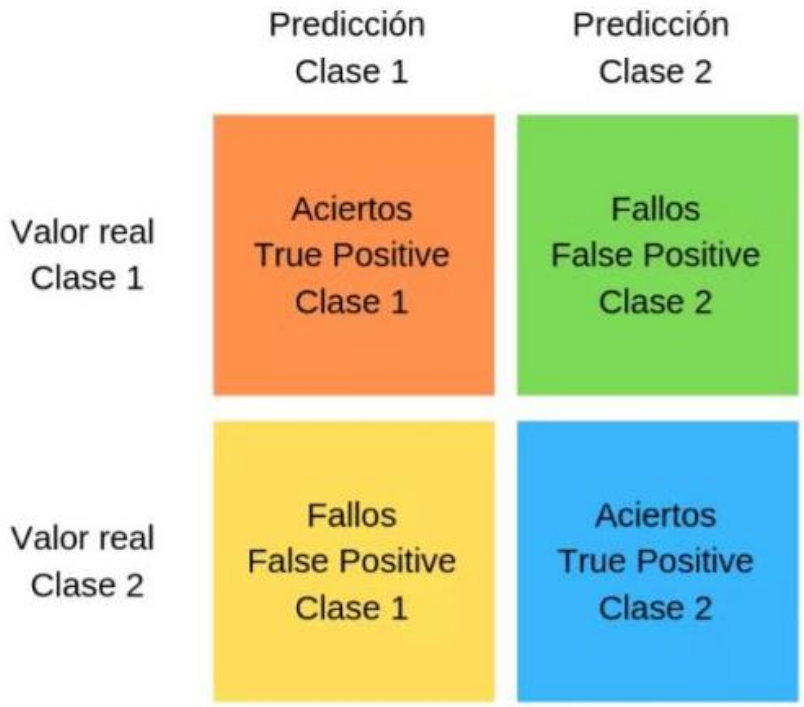
Muestreo (sobremuestreo y submuestreo)

Métricas

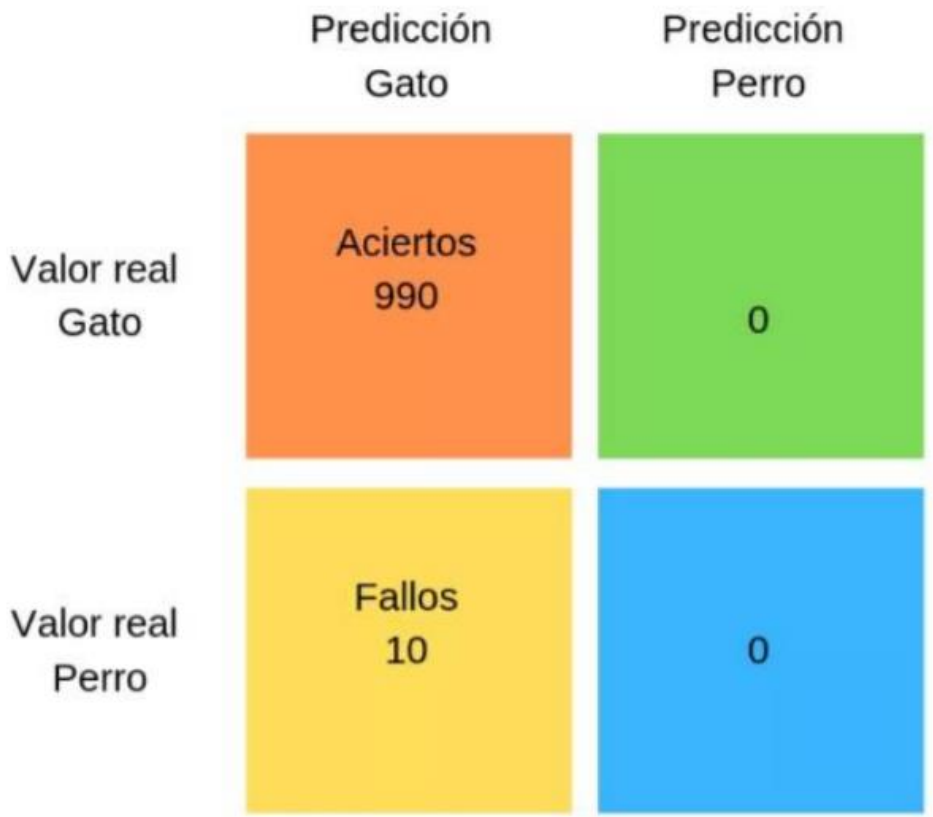
- Accuracy del modelo es básicamente el numero total de predicciones correctas dividido por el número total de predicciones. En este caso da 99% cuando no hemos logrado identificar ningún perro.
- La Precisión de una clase define cuan confiable es un modelo en responder si un punto pertenece a esa clase. Para la clase gato será del 99% sin embargo para la de perro será 0%
- El Recall de una clase expresa cuan bien puede el modelo detectar a esa clase. Para gatos será de 1 y para perros 0.
- El valor F1 se utiliza para combinar las medidas de precisión y recall en un sólo valor. Esto es práctico porque hace más fácil el poder comparar el rendimiento combinado de la precisión y la exhaustividad entre varias soluciones.

Métricas y Confusión Matrix

- Para poder entender esto un poco mejor, utilizaremos la llamada «Confusión matrix» que nos ayudará a comprender las salidas de nuestra máquina:



Ejemplo


Predicción
Gato
Predicción
Perro
Valor real
Gato
Aciertos
990
0
Valor real
Perro
Fallos
10
0

ejemplo

Accuracy

$$\text{Precision} = \frac{\textit{True Positive}}{\textit{True Positive} + \textit{False Positive}}$$

$$\text{Recall} = \frac{\textit{True Positive}}{\textit{True Positive} + \textit{False Negative}}$$

$$\text{Accuracy} \frac{990 + 0}{990 + 0 + 10 + 0}$$

$$\text{Precisión Clase 1} \frac{990}{990 + 10}$$

$$\text{Recall Clase 1} \frac{990}{990 + 0}$$

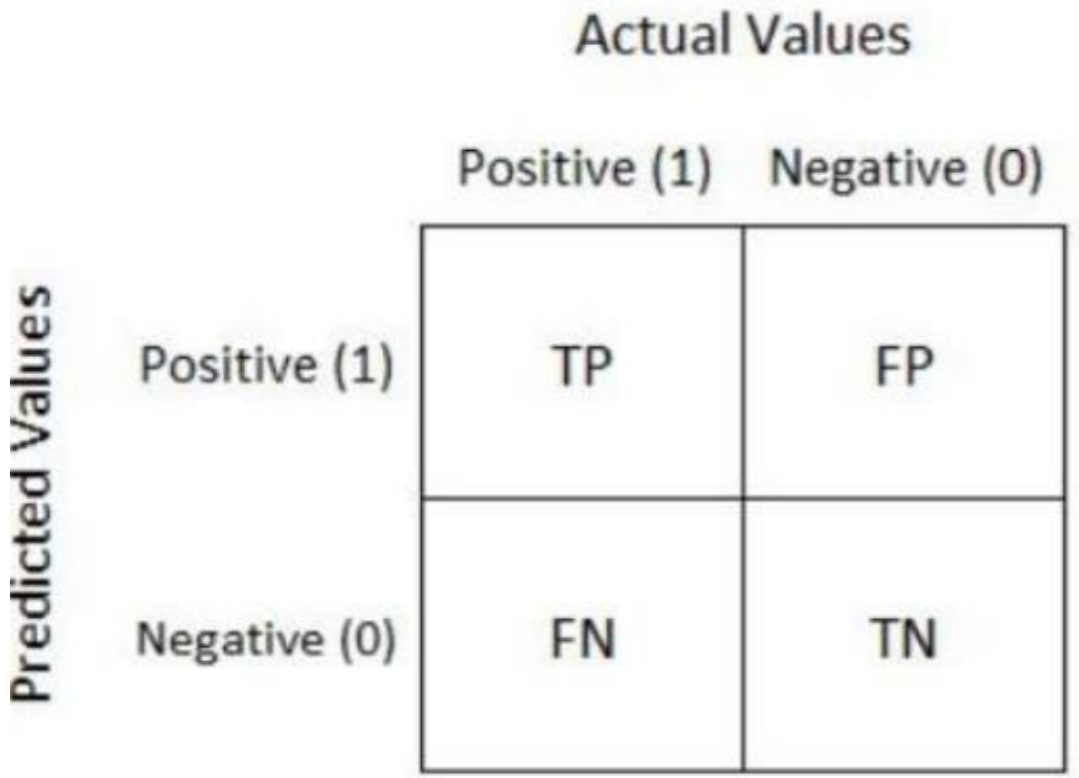
$$\text{Precisión Clase 2} \frac{0}{0 + 0}$$

$$\text{Recall Clase 2} \frac{0}{0 + 10}$$

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

Errores tipo 1 y tipo 2

- TP = Verdadero positivo: el modelo predijo la clase positiva correctamente, para ser una clase positiva. FP = Falso positivo: el modelo predijo la clase negativa incorrectamente, para ser una clase positiva. FN = Falso negativo: el modelo predijo incorrectamente que la clase positiva sería la clase negativa. TN = Verdadero negativo: el modelo predijo la clase negativa correctamente, para ser la clase negativa.



Errores

- Error de tipo 1: el modelo predijo que la instancia sería una clase positiva, pero es incorrecta. Esto es falso positivo (FP).
- •Error de tipo 2: el modelo predijo que la instancia sería la clase Negativa, pero es incorrecta. Esto es falso negativo (FN)

- Recall = TP / (TP + FN)
- Precision = TP / (TP + FP)
- F-Score = (2 * Recall * Precision) / (Recall + Precision)

Tenemos cuatro casos posibles para cada clase:

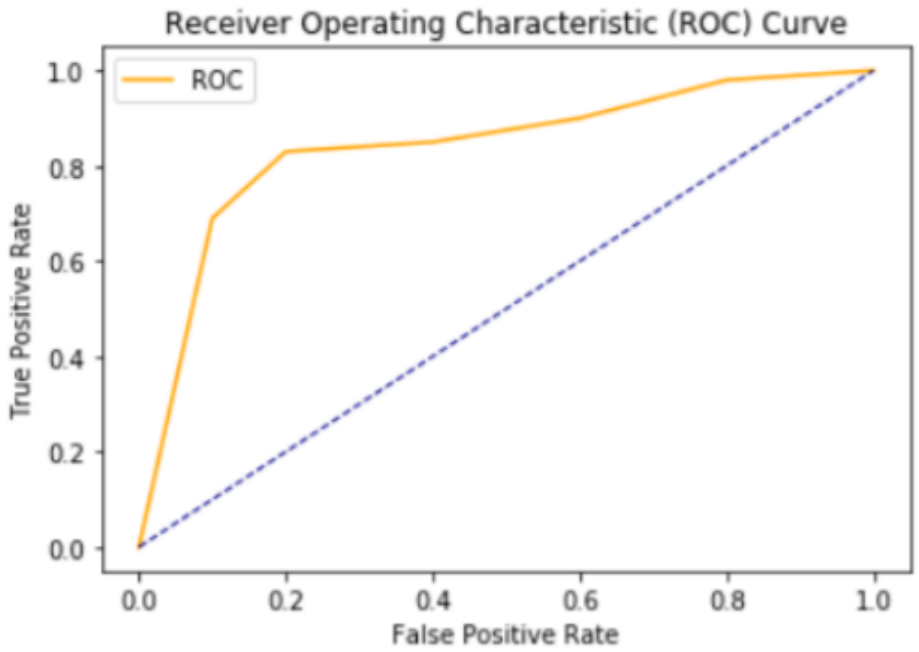
- Alta precisión y alto recall: el modelo maneja perfectamente esa clase
- Alta precisión y bajo recall: el modelo no detecta la clase muy bien, pero cuando lo hace es altamente confiable.
- Baja precisión y alto recall: La clase detecta bien la clase pero también incluye muestras de otras clases.
- Baja precisión y bajo recall: El modelo no logra clasificar la clase correctamente.

Resumen

- Accuracy cantidad de veces que acertaste una afirmación, sobre el total de datos de entrada.
- Esta métrica se define como la cantidad de casos verdaderos positivos sobre la cantidad total de todo lo que dijiste que era positivo.
- Recall calcula cuántos de los positivos reales captura nuestro modelo al etiquetarlo como positivo (verdadero positivo). Aplicando la misma comprensión, sabemos que Recall será la métrica del modelo que usaremos para seleccionar nuestro mejor modelo cuando haya un alto costo asociado con Falso negativo.
- F1 Score podría ser una mejor medida para usar si necesitamos buscar un equilibrio entre Precisión y Recuperación Y hay una distribución de clase desigual (gran cantidad de Negativos Reales)

Introducción a AUC - Curva ROC

- La curva AUC-ROC es la métrica de selección del modelo para el problema de clasificación bi-multi class. ROC es una curva de probabilidad para diferentes clases. ROC nos dice qué tan bueno es el modelo para distinguir las clases dadas, en términos de la probabilidad predicha.
- Una curva ROC típica tiene una tasa de falsos positivos (FPR) en el eje X y una tasa de verdaderos positivos (TPR) en el eje Y.



Árbol de Decisión

- No asume una distribución
- Simple de entender e interpretar
- Ignora variables redundantes
- Las variables de entrada y salida pueden ser categóricas o continuas.
- Divide el espacio de predictores (variables independientes) en regiones distintas y no sobrepuestas.
- Se divide la población o muestra en conjuntos homogéneos basados en la variable de entrada más significativa.
- La construcción del árbol sigue un enfoque de división binaria recursiva.

Desventajas

- Sobreajuste Pérdida de información al categorizar variables continuas
- Precisión: métodos como SVM y clasificadores tipo ensamblador a menudo tienen tasas de error 30% más bajas que CART (Classification and Regression Trees)
- Inestabilidad: un pequeño cambio en los datos puede modificar ampliamente la estructura del árbol. Por lo tanto la interpretación no es tan directa como parece.

Arboles de Regressión vs árboles de clasificación

Regresión	Clasificación
Variable dependiente es continua	Variable dependiente es categórica
Valores de los nodos terminales se reducen a la media de las observaciones en esa región.	El valor en el nodo terminal se reduce a la moda de las observaciones del conjunto de entrenamiento que han “caído” en esa región.

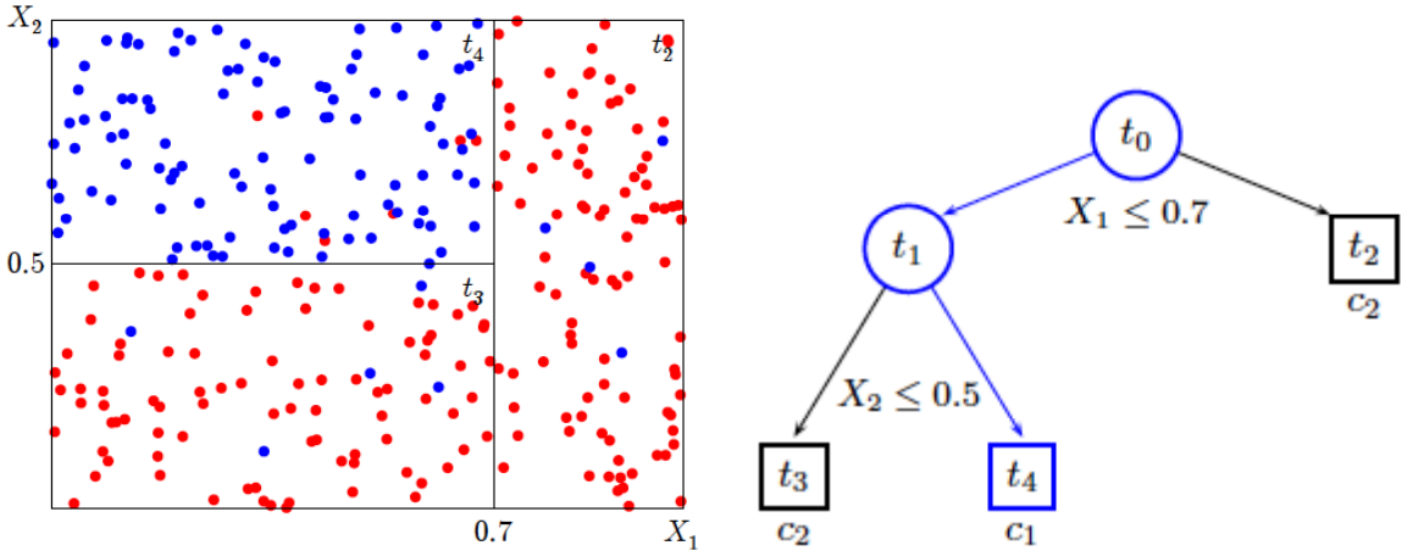


Fig. 36: Recursive partitioning and corresponding decision tree. Louppe (2014).

Extructura de un Arbol

Todos comparten dos tareas

- ¿Cómo dividir un nodo?
- ¿Cuando dejar de dividirlo?

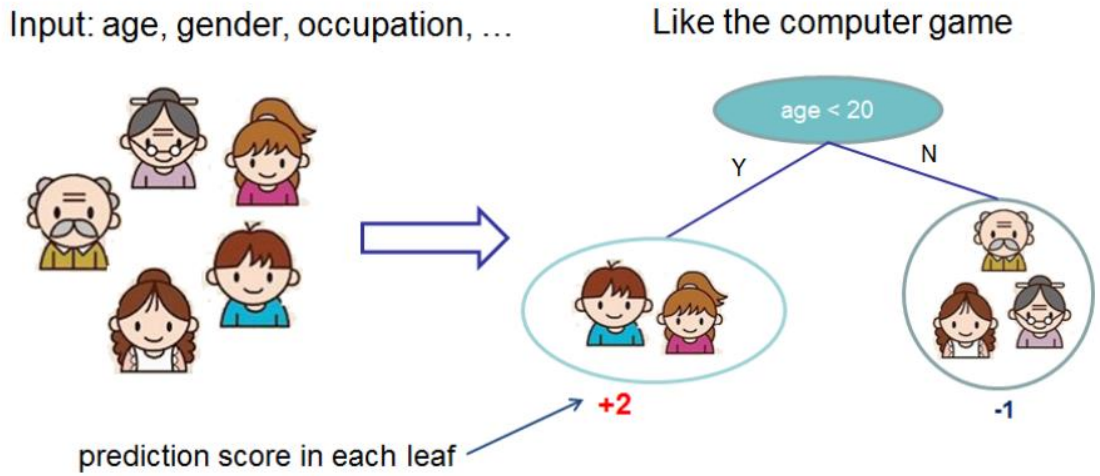
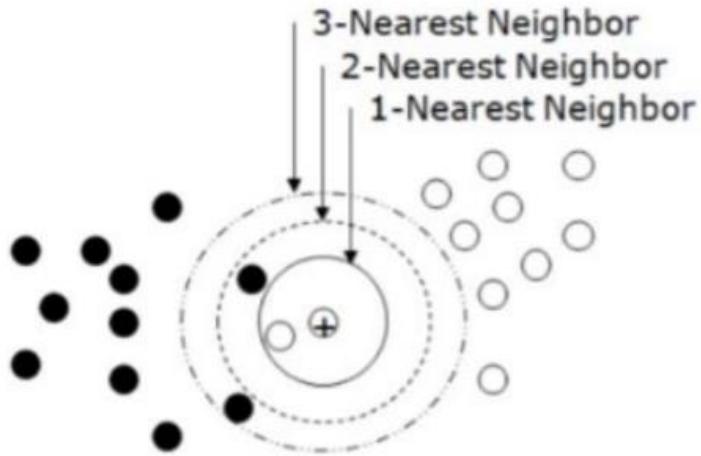


Fig. 37: Hypothetical tree which classifies whether a person likes a game. Chen and Guestrin (2016).

Ejemplo

K vecinos más cercanos o K-NN (K Nearest Neighbours)

- Clasifica cada dato nuevo en el grupo que corresponda, según tenga k vecinos más cerca de un grupo o de otro. Es decir, calcula la distancia del elemento nuevo a cada uno de los existentes, y ordena dichas distancias de menor a mayor para ir seleccionando el grupo al que pertenecer. Este grupo será, por tanto, el de mayor frecuencia con menores distancias

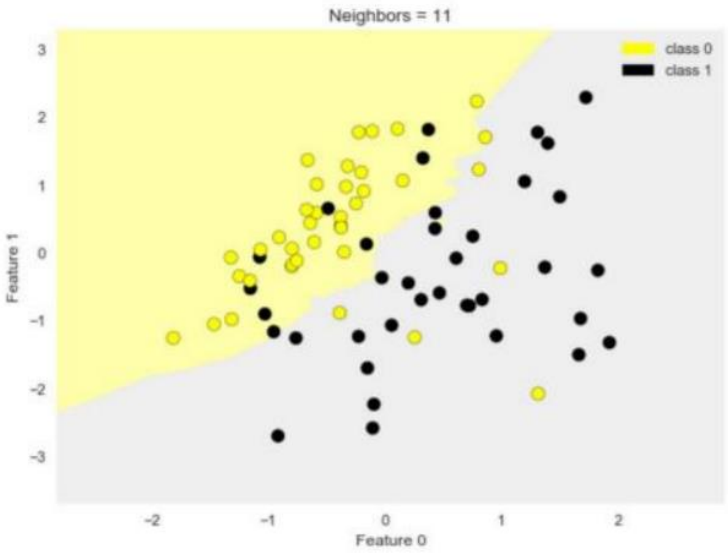


Web Data Mining, de Bing Liu

¿Cómo funciona KNN?

- • Calcular la distancia entre el item a clasificar y el resto de items del dataset de entrenamiento.
- Seleccionar los «k» elementos más cercanos (con menor distancia, según la función que se use)
- Realizar una «votación de mayoría» entre los k puntos: los de una clase/etiqueta que <> decidirán su clasificación final.

Imagen de clasificación



Codificación de columnas de datos

- Varios algoritmos de aprendizaje automático requieren datos de entrada numéricos, por lo que debe representar columnas categóricas en una columna numérica.
- Para codificar estos datos, puede asignar cada valor a un número. Por ejemplo, nublado: 0, lluvioso: 1 y soleado: 2.
- Este proceso se conoce como codificación de etiqueta, y sklearn lo hará convenientemente utilizando Label Encoder

Pros y contras

- Aunque sencillo, se utiliza en la resolución de multitud de problemas, como en sistemas de recomendación, búsqueda semántica y detección de anomalías.
- Como pros tiene sobre todo que es sencillo de aprender e implementar.
- Tiene como contras que utiliza todo el dataset para entrenar «cada punto» y por eso requiere de uso de mucha memoria y recursos de procesamiento (CPU). Por estas razones kNN tiende a funcionar mejor en datasets pequeños y sin una cantidad enorme de features (las columnas).