Ayudantía 8

Matriz de Confusión, KNN y Árboles

Contenidos

- Contexto de aprendizaje supervisado
- Motivación: TP, TN, FP, FN.
- Matriz de confusión
- Métricas: Accuracy, recall y f1-score
- KNN
- Árboles de decisión

Un problema clásico: El Spam en el Mail.

Image Source: Gemini AI Pro

¿Cómo lo solucionamos?

Dataset con etiqueta

label	text	label_num
ham	this is a follow up to the note i gave you on monday , 4 / 3 / 00 { preliminary flow data provided by daren } .	0
ham	Subject: hpl nom for january 9 , 2001 (see attached file : hplnol 09 . xls)	0
ham	ho ho ho , we ' re around to that most wonderful time of the year - - - neon leaders retreat time ! i know that this time of year is extremely hectic , and that it ' s tough to think about anything past the holidays , but life does go on past the week of december 25 through january 1 , and that ' s what i ' d like you to think about for a minute . on the calender that i handed out at the beginning of the fall semester , the retreat was scheduled for the weekend of january 5 - 6 . but because of a youth ministers conference that brad and dustin are connected with that week , we ' re going to change the date to the following weekend , january 12 - 13 . now comes the part you need to think about . i think we all agree that it ' s important for us to get together and have some time to recharge our	0
spam	Subject: photoshop , windows , office . cheap . main trending abasements darer prudently fortuitous undergone lighthearted charm orinoco taster railroad affluent pornographic cuvier irvin parkhouse blameworthy chlorophyll robed diagrammatic fogarty clears bayda inconveniencing managing represented smartness hashish academies shareholders unload badness	1
ham	Subject: re : indian springs this deal is to book the teco pvr revenue . it is my understanding that teco just sends us a check , i haven ' t received an answer as to whether there is a predermined price associated with this deal or if teco just lets us know what we are giving . i can continue to chase this deal down if you need .	0
ham	Subject: ehronline web address change this message is intended for ehronline users only . due to a recent change to ehronline , the url (aka " web address ") for accessing ehronline needs to be changed on your computer . the change involves adding the letter " s " to the " http " reference in the url . the url for accessing ehronline should be : https : / / ehronline . enron . com .	0

Source: Kaggle spam mail dataset

Learner

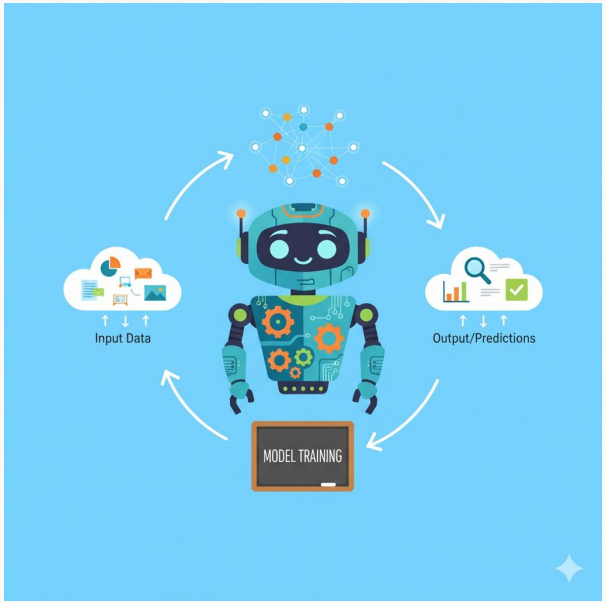


Image Source: Gemini AI Pro

Si te interesa aprender sobre la teoría del aprendizaje automático a nivel matemático te recomiendo el libro:
“Understanding Machine Learning: From Theory to Algorithms” de Shai Shalev-Shwartz y Shai Ben-David.

KNN

Los K vecinos más cercanos y frecuentes son los que determinan mi clase.

$$y = \arg\max_{C_j} \sum_{l \in X_k} I\left(y_l = j\right).$$

¿Bajo qué criterio de distancia, el vecino es el más cercano?

Un ejemplo: la norma euclidiana

$$D\left(x, y\right) = \sqrt{\sum_{i=1}^{d} \left(x_{[i]} - y_{[i]}\right)^2}.$$

Fórmulas extraidas de “A new approach to K-nearest neighbors distance metrics on sovereign country credit rating”por Ali İhsan Çetin a b y Ali Hakan Büyüklü.

KNeighborsClassifier

```
class sklearn.neighbors.KNeighborsClassifier(n_neighbors=5, *,
weights='uniform', algorithm='auto', leaf_size=30, p=2, metric='minkowski',
metric_params=None, n_jobs=None)
```

[source]

KNN

n_neighbors : *int, default=5*

Number of neighbors to use by default for `kneighbors` queries.

weights : *{'uniform', 'distance'}, callable or None, default='uniform'*

Weight function used in prediction. Possible values:

- 'uniform' : uniform weights. All points in each neighborhood are weighted equally.
- 'distance' : weight points by the inverse of their distance. in this case, closer neighbors of a query point will have a greater influence than neighbors which are further away.
- [callable] : a user-defined function which accepts an array of distances, and returns an array of the same shape containing the weights.

Refer to the example entitled Nearest Neighbors Classification showing the impact of the `weights` parameter on the decision boundary.

algorithm : *{'auto', 'ball_tree', 'kd_tree', 'brute'}, default='auto'*

Algorithm used to compute the nearest neighbors:

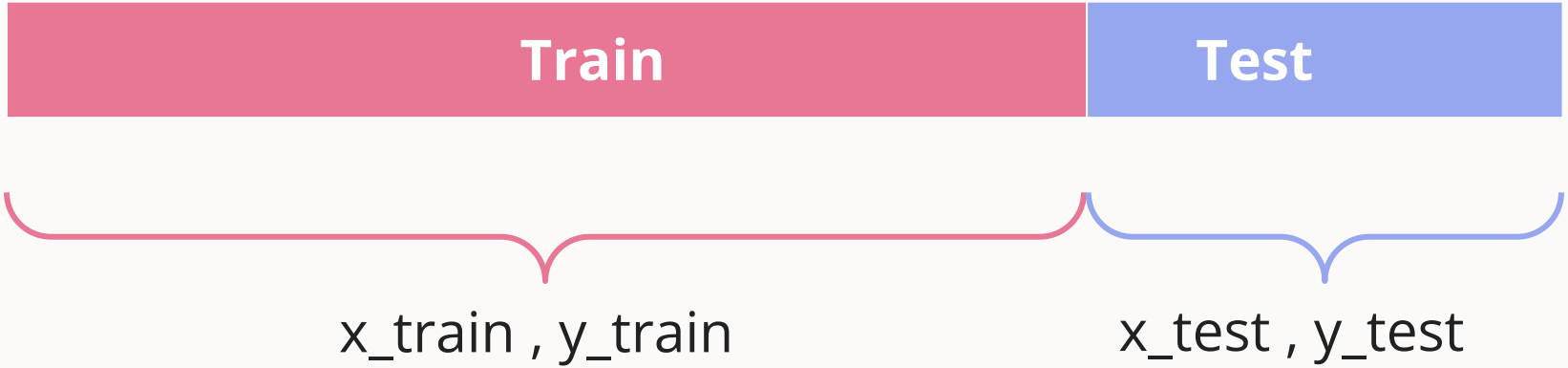
- 'ball_tree' will use `BallTree`
- 'kd_tree' will use `KDTree`
- 'brute' will use a brute-force search.
- 'auto' will attempt to decide the most appropriate algorithm based on the values passed to `fit` method.

Note: fitting on sparse input will override the setting of this parameter, using brute force.

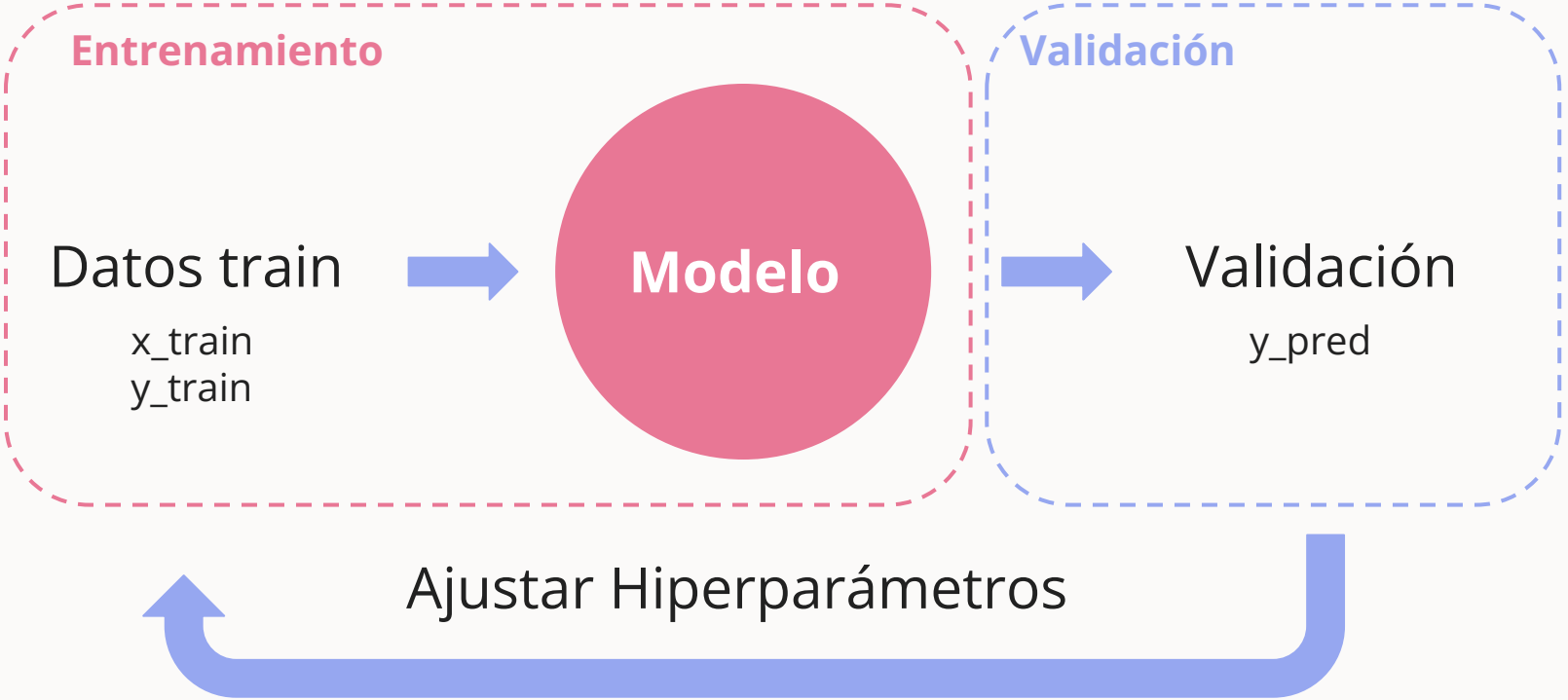
leaf_size : *int, default=30*

Leaf size passed to BallTree or KDTree. This can affect the speed of the construction and query, as well as the memory required to store the tree. The optimal value depends on the nature of the problem.

Separación de los Datos



Entrenamiento y Validación



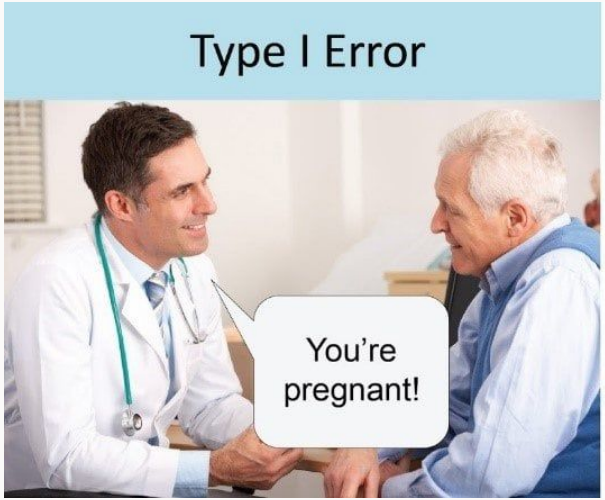
```
>>> from sklearn.neighbors import KNeighborsClassifier
>>>
>>> model = KNeighborsClassifier()
>>> model.fit(x_train, y_train)
>>> y_pred = model.predict(x_test)
```

▼ KNeighborsClassifier ⓘ ?

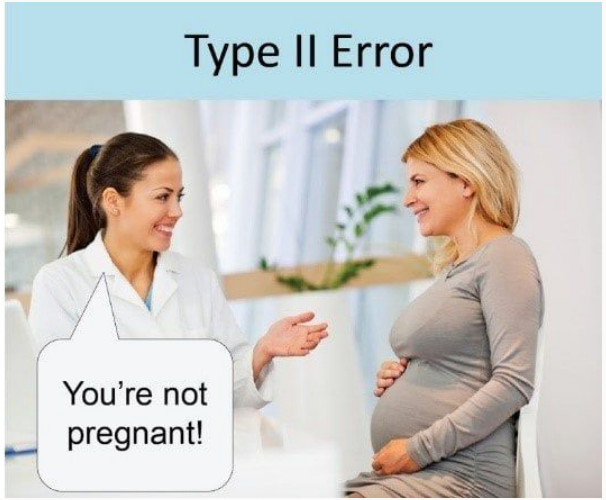
▶ Parameters

Evaluar la calidad del modelo

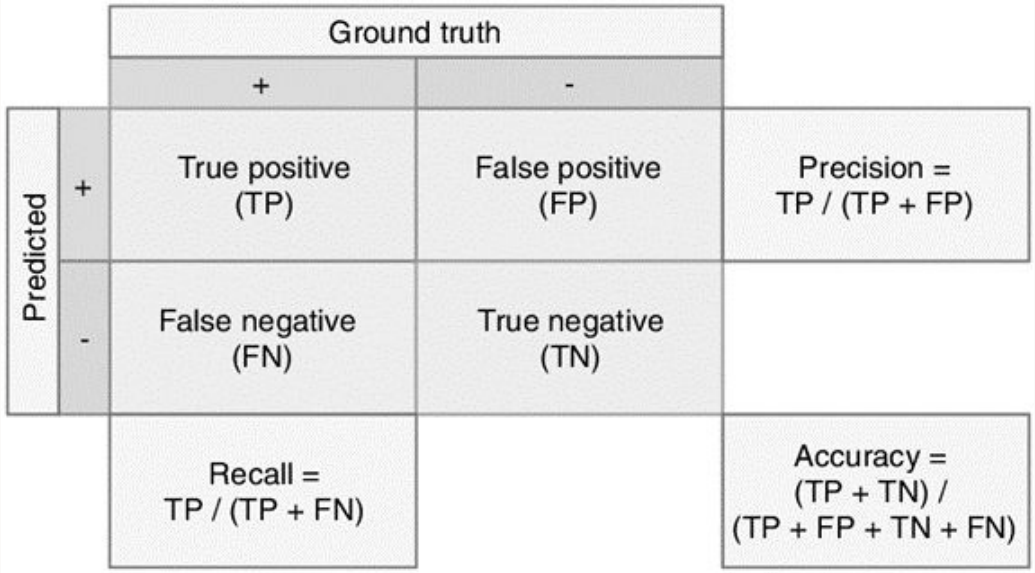
Falsos positivos



Falsos negativos



Errores del modelo: Falsos positivos y falsos negativos.



		Ground truth +	Ground truth -	
Predicted	+	True positive (TP)	False positive (FP)	Precision = TP / (TP + FP)
Predicted	-	False negative (FN)	True negative (TN)	
		Recall = TP / (TP + FN)		Accuracy = (TP + TN) / (TP + FP + TN + FN)

Matriz de confusión extraída de researchgate: Optimization of a soil type prediction method based on the deep learning model and vegetation characteristics.

Árboles de decisión

- Técnica de aprendizaje supervisado
- Usados para clasificación y regresión.
- Organizan los datos para dividirlos en diferentes clases.

Árboles de decisión

Clima	Temperatura	Humedad	Viento	Jugar?
soleado	alta	alta	F	No
soleado	alta	alta	V	No
nublado	alta	alta	F	Si
lluvioso	Agradable	alta	F	Si
lluvioso	frio	normal	F	Si
lluvioso	frio	normal	V	No
nublado	frio	normal	V	Si
soleado	Agradable	alta	F	No
soleado	frio	normal	F	Si
lluvioso	Agradable	normal	F	Si
soleado	Agradable	normal	V	Si
nublado	Agradable	alta	V	Si
nublado	alta	normal	F	Si
lluvioso	Agradable	alta	V	No

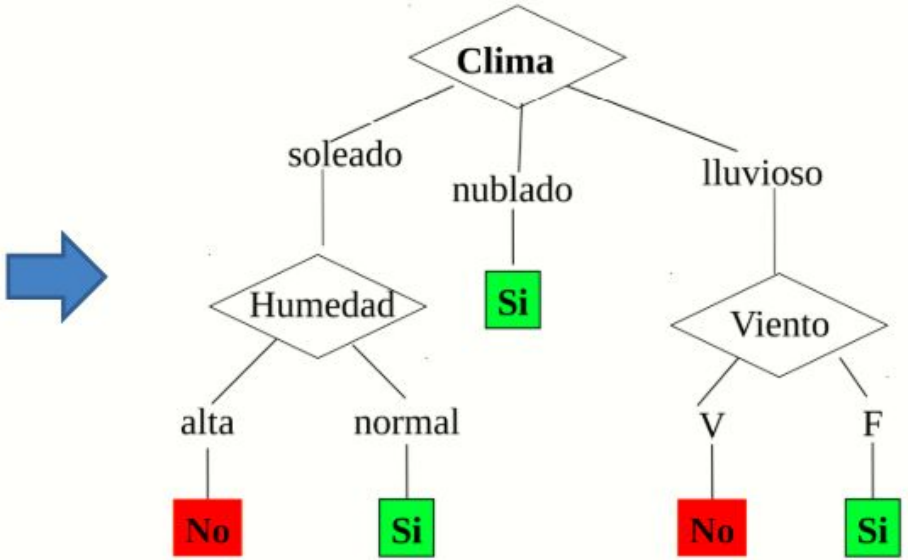


Imagen obtenida de la clase de árboles de decisión del profesor Hans Löbel 2025-1

Árboles de decisión

Estructura del Árbol:

- Raíz: Representa la totalidad de los datos.

- Nodos:
  - Cada nodo (excepto las hojas) es una pregunta o condición sobre alguna característica.
  - Según la respuesta, los datos se dividen en nodos del siguiente nivel.

Árboles de decisión

Estructura del Árbol:

- Para decidir la mejor división en un nodo se utilizan métricas como Gini, Entropía o índice de impureza.
- Elegimos la característica que entrega la mayor ganancia de información (mayor reducción de entropía)
- Estas son calculadas por los modelos automáticamente

Árboles de decisión

DecisionTreeClassifier

```
class sklearn.tree.DecisionTreeClassifier(*, criterion='gini', splitter='best',
max_depth=None, min_samples_split=2, min_samples_leaf=1,
min_weight_fraction_leaf=0.0, max_features=None, random_state=None,
max_leaf_nodes=None, min_impurity_decrease=0.0, class_weight=None, ccp_alpha=0.0,
monotonic_cst=None)
```

[source]

A decision tree classifier.

Read more in the User Guide.

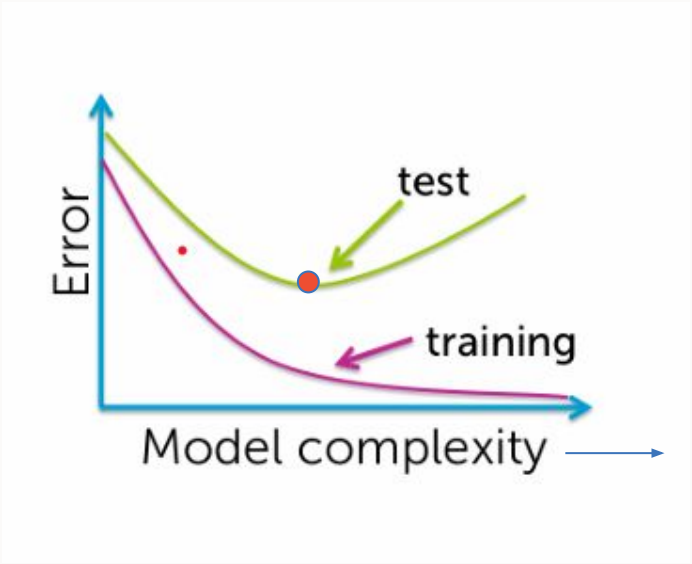
Parameters:

criterion : *{"gini", "entropy", "log_loss"}, default="gini"*

The function to measure the quality of a split. Supported criteria are "gini" for the Gini impurity and "log_loss" and "entropy" both for the Shannon information gain, see Mathematical formulation.

Fuente: DecisionTreeClassifier. (n.d.). Scikit-learnhttps://scikit-learn.org/stable/modules/generated/sklearn.tree.DecisionTreeClassifier.html

Sobreajuste es un problema importante para los árboles de decisión



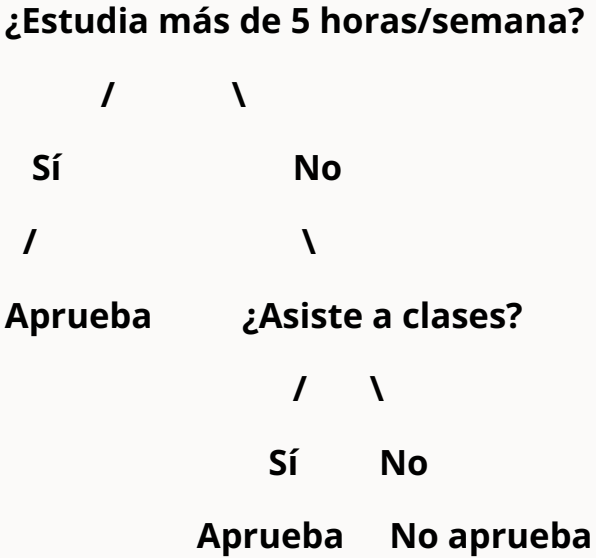
Profundidad del árbol en este caso

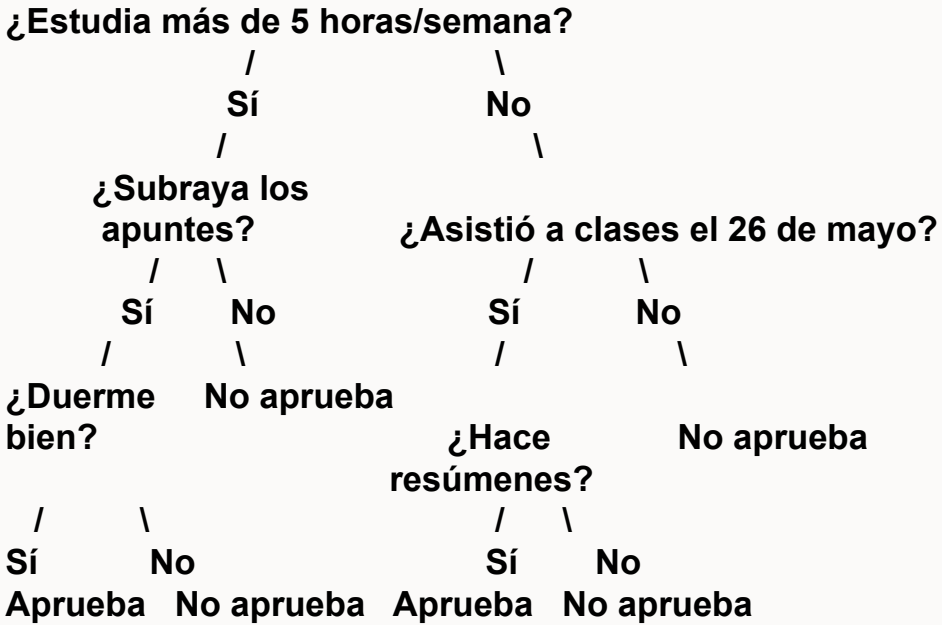
Imagen obtenida de la clase de árboles de decisión del profesor Hans Löbel 2025-1

Técnicas para reducir el sobreajuste

- Usar un conjunto de validación para detener el crecimiento.
- Detener cuando los datos restantes no son estadísticamente relevantes.
- Podar el árbol después de construirlo completamente.
- Penalizar la complejidad al seleccionar atributos.

Ejemplo de sobreajuste en árboles de decisión





Este árbol es poco profundo, tiene reglas generales. Puede que no sea perfecto, pero generaliza bien a nuevos estudiantes.

Este árbol tiene muchas divisiones específicas, como fechas y detalles muy concretos. Esto significa que está "aprendiendo de memoria" los datos del pasado, en vez de aprender patrones generales.

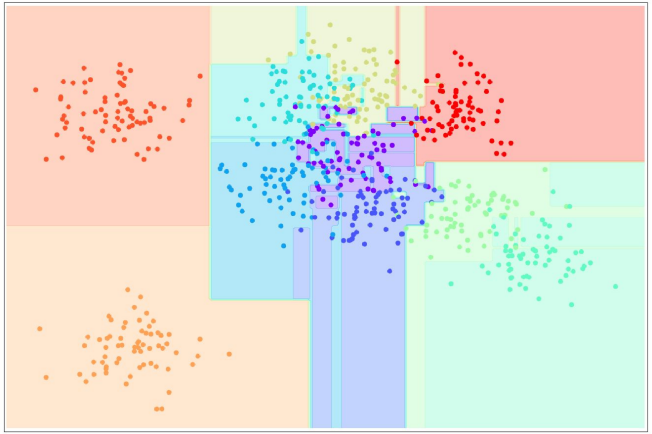
Más modelos

- El **árbol de decisión** tiene una gran ventaja: es simple y fácil de interpretar...
- ... pero puede sufrir serios problemas de sobreajuste

Más modelos

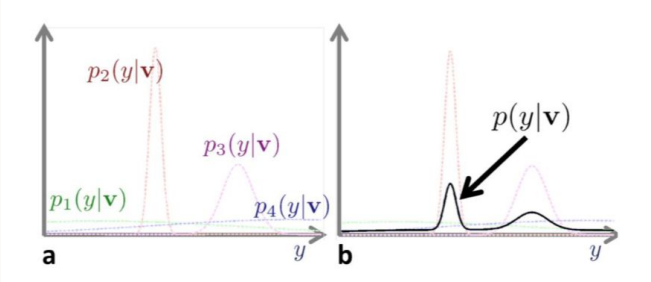
¿De dónde proviene el sobreajuste en un árbol de decisión?

- Al crear nuevos nodos en el entrenamiento (“bajar” en el árbol), la **muestra** se hace cada vez más reducida
- Si hay muchos **atributos**, es altamente probable elegir alguno bueno en los datos de entrenamiento, pero que es “inútil” para generalizar



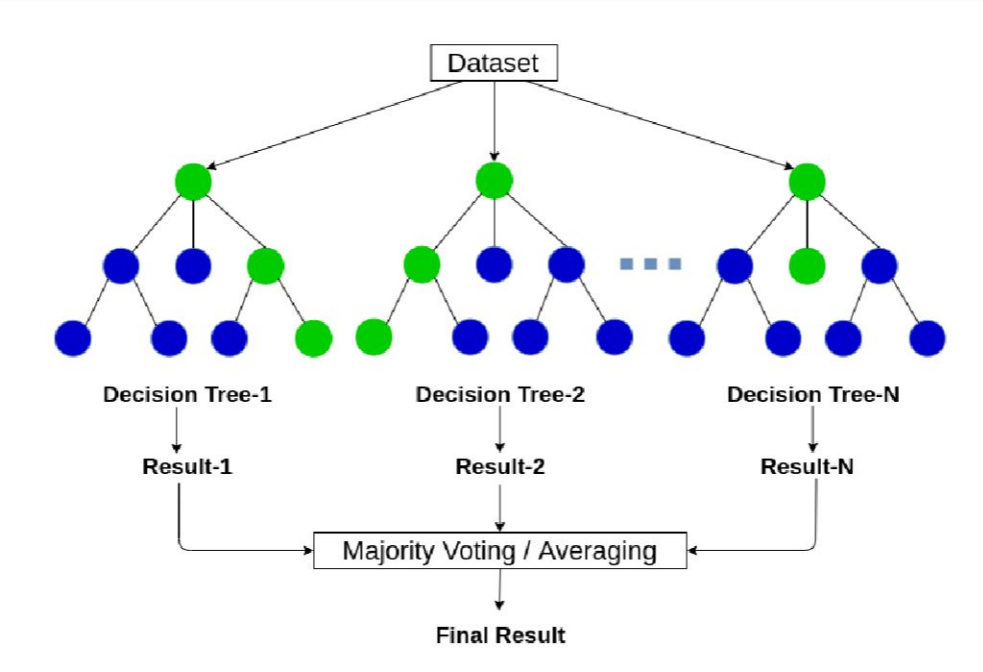
Ensambles de modelos con baja correlación

- Si el **patrón de error** (acierto) es **distinto** para todos, es altamente probable que, en promedio, la respuesta del ensamble sea correcta (los **errores se cancelan**)



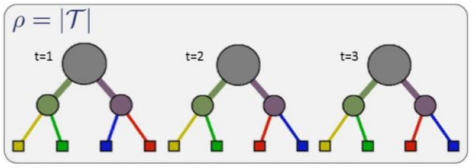
Muestras aleatorias

- Tomemos varios **árboles de decisión** y armemos una gran **“votación”**, donde cada árbol tenga “un voto”.
- Como vimos en la diapositiva pasada, necesitamos que los modelos tengan **baja correlación** para que nuestro plan resulte. **¿Cómo logramos esto?**
- Si cada modelo ve sólo una parte de los datos (elegida aleatoriamente), es probable que la **correlación** entre ellos **disminuya**

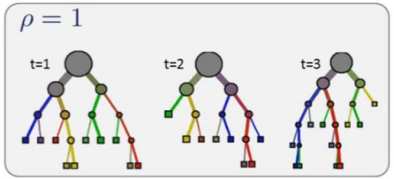


Atributos aleatorias

- Lo anterior no basta: **¿qué pasa si tenemos un atributo muy bueno?**
- La solución es similar: tomamos muestras aleatorias de atributos
- Esto no solo **disminuye la correlación**, sino que también **limita la profundidad del árbol**, reduciendo la complejidad de este



a) Low randomness, high tree correlation

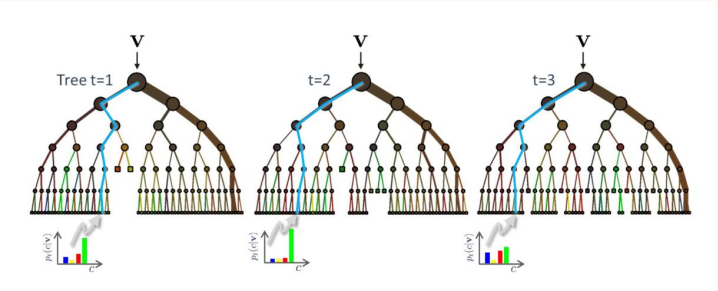


b) High randomness, low tree correlation

Random Forest

Random Forest

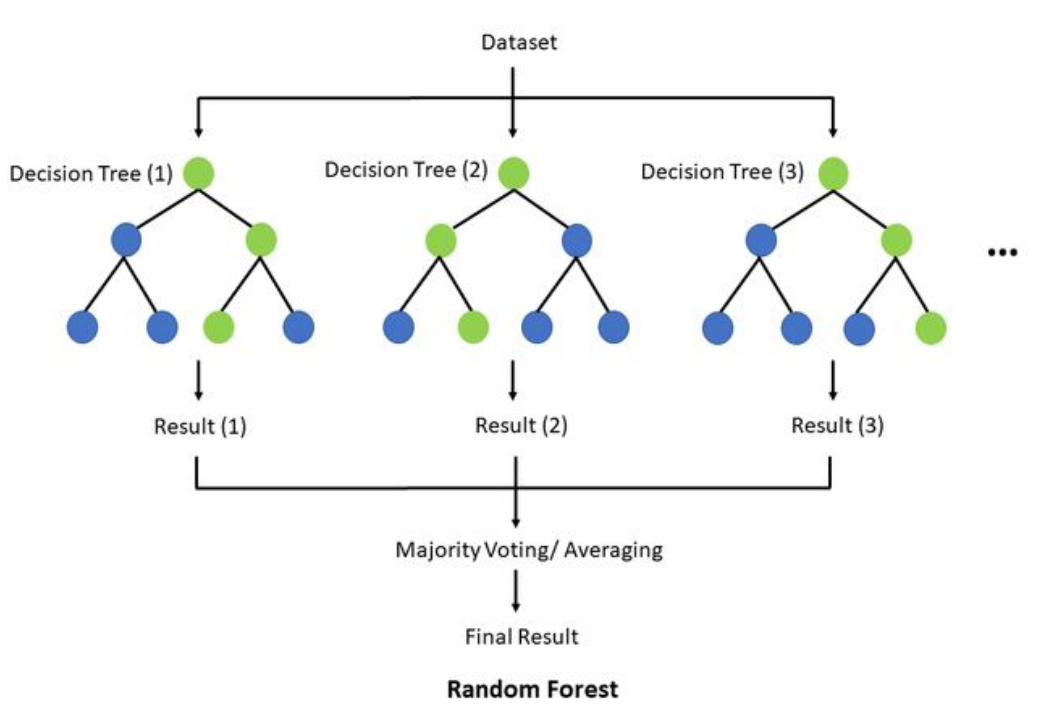
- La introducción de todas estas estrategias basadas en **aleatoriedad**, aplicada a **varios árboles de decisión**, convierten esta técnica en un ensamble llamado ***Random Forest***



Random Forest

- Un árbol para cada muestra aleatoria (subconjunto) de los datos.
- Árboles de **poca profundidad** para evitar el *overfitting*.
- La predicción final se obtiene de un promedio entre las predicciones de los árboles del bosque

Random Forest



Hiperparámetros: RF

- **Los mismos de un árbol**
- **n_estimators:** número de árboles en el bosque

Ventajas de *Random Forest*

- Al igual que los **árboles de decisión**, son **fácilmente interpretables**
- Rendimiento es **altamente competitivo** con datos **tabulados**
- Gracias a la aleatorización en su construcción, son altamente **resistentes** al ***overfitting***

Ventajas de *Random Forest*

- Todas estas ventajas hacen que ***Random Forest*** sea un modelo **altamente utilizado**, a día de hoy, en el mundo de *machine learning*