Machine learning y Deep learning con Python: Lectura 5

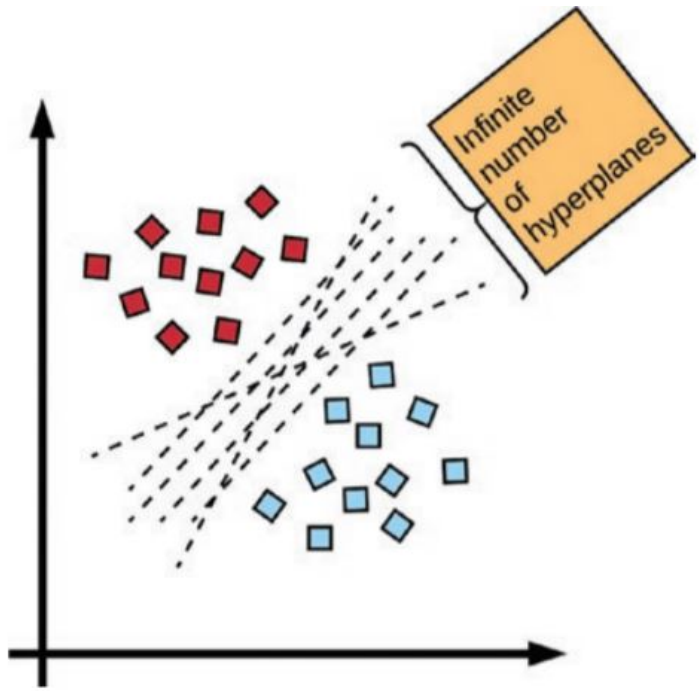
Ing. Pedro Rotta

Universidad de Piura - Vida Universitaria

Enero-2022

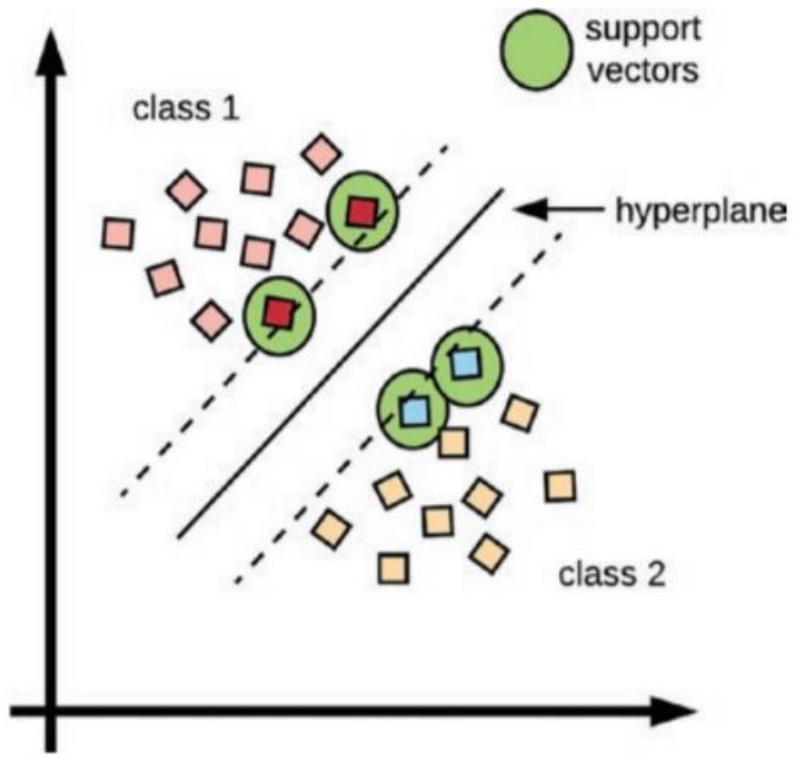
Máquinas de Vector Soporte

Máquinas vector soporte aka SVM(Support vector machine), es un algoritmo para clasificación y regresión. El objetivo en una clasificación binaria, es ajustar el modelo a la mejor línea que separe los grupos. Cuando se tiene más de 2 características, el separador se conoce como hiperplano.



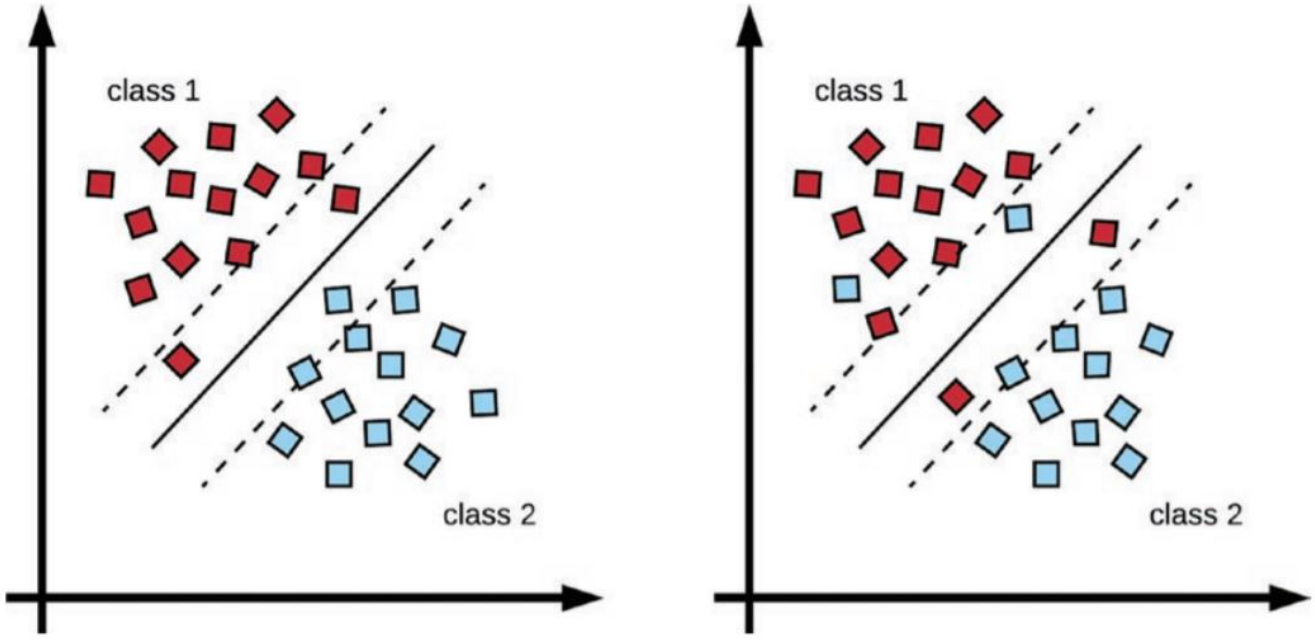
Vectores de soporte

Para obtener el mejor hiperplano, el modelo trata de encontrar los vectores más cercanos a la frontera de las clases. El mejor hiperplano, estará en justamente la mitad exacta de ambas fronteras. Los puntos en el límite que ayudan a definir la frontera son llamados vectores soporte porque definen la máxima distancia entre clases.



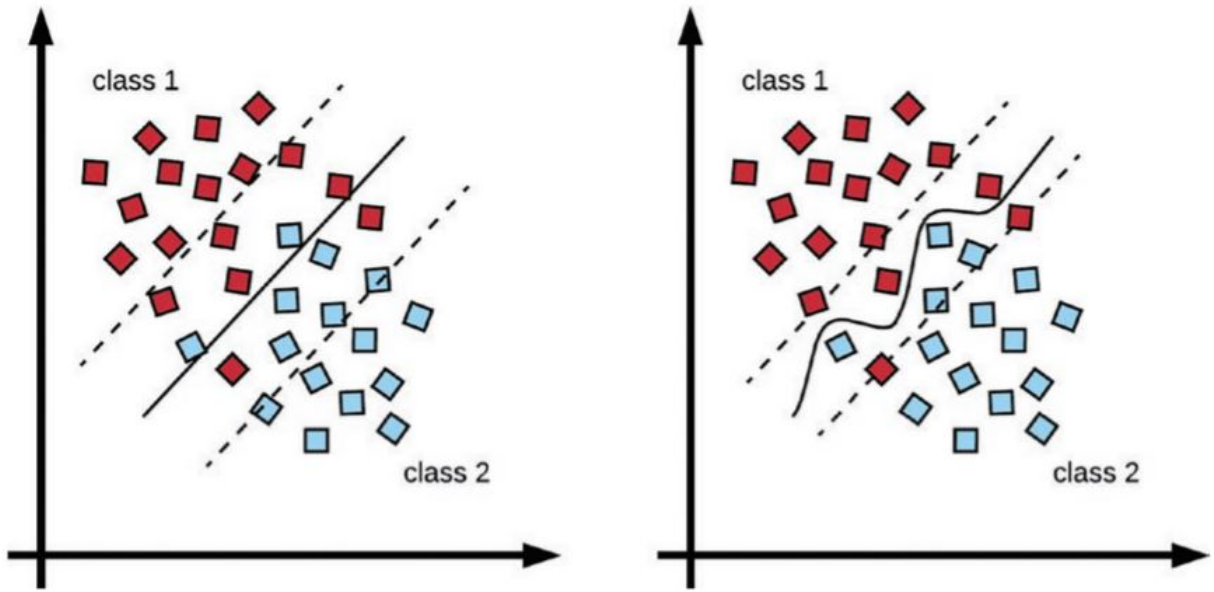
El clasificador basado en vectores de soporte

En realidad es casi imposible en la vida real, separa exactamente con un hiperplano a dos clases perfectamente. Para ello, el **SVC** (support vector clasificator) usa un método que se conoce como márgen suave o **soft margin**. El margen suave, permite que ciertos puntos al momento del entrenamiento sean erróneamente clasificados, pero generaliza el modelo de una mejor manera.



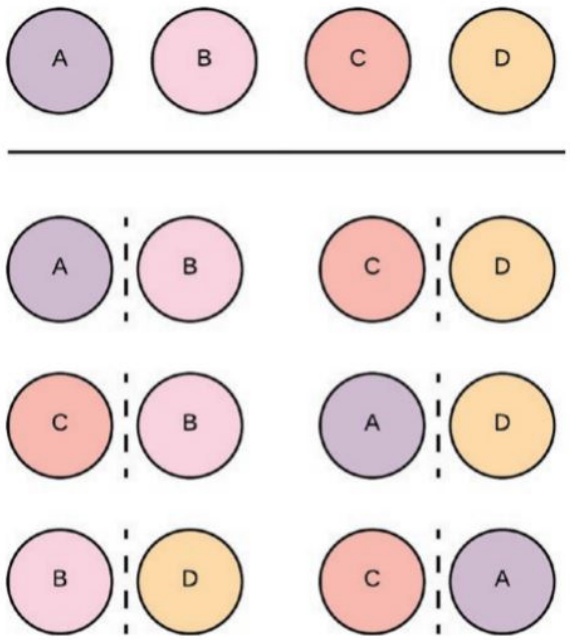
Hiperparámetros en SVC

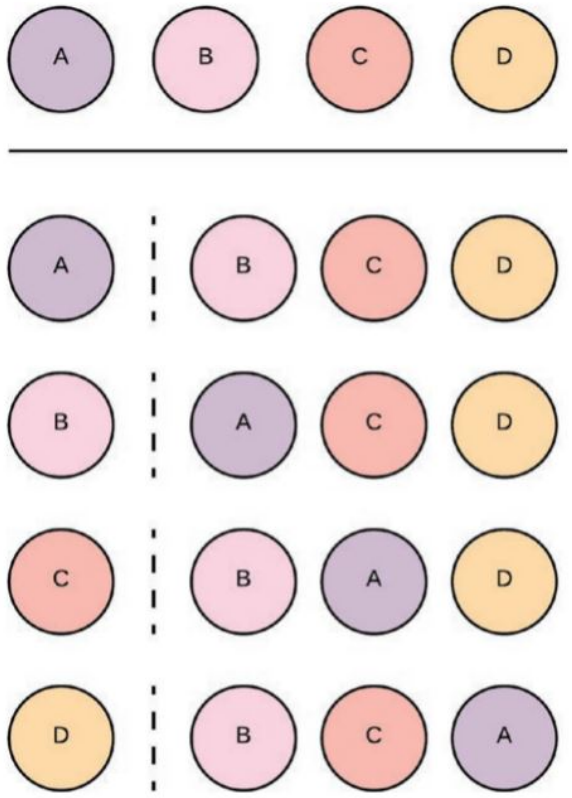
El paramétro C se vincula con el control del grado de violaciones permitidas en la frontera del conjunto de datos. Este parámetro debe ser un número positivo real. Si C es más grande, el algoritmo será más permisivo. Se debe ajustar C para que el algoritmo no alcance ni el sobreajuste ni el subajuste.



Hiperparámetros en SVC

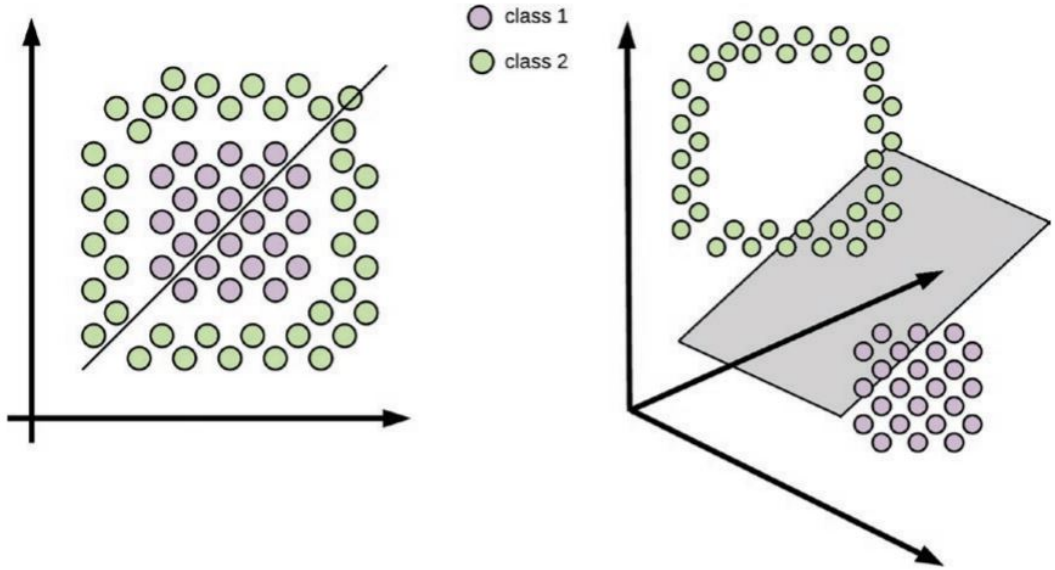
Otro hiperparámetro, es el tipo de método cuándo el algoritmo es multiclass. Existen 2 aproximaciones. OVO (One vs One) que dibuja una frontera para cada par de clases. y OVR(One vs Rest) que dibuja una frontera por cada clase.





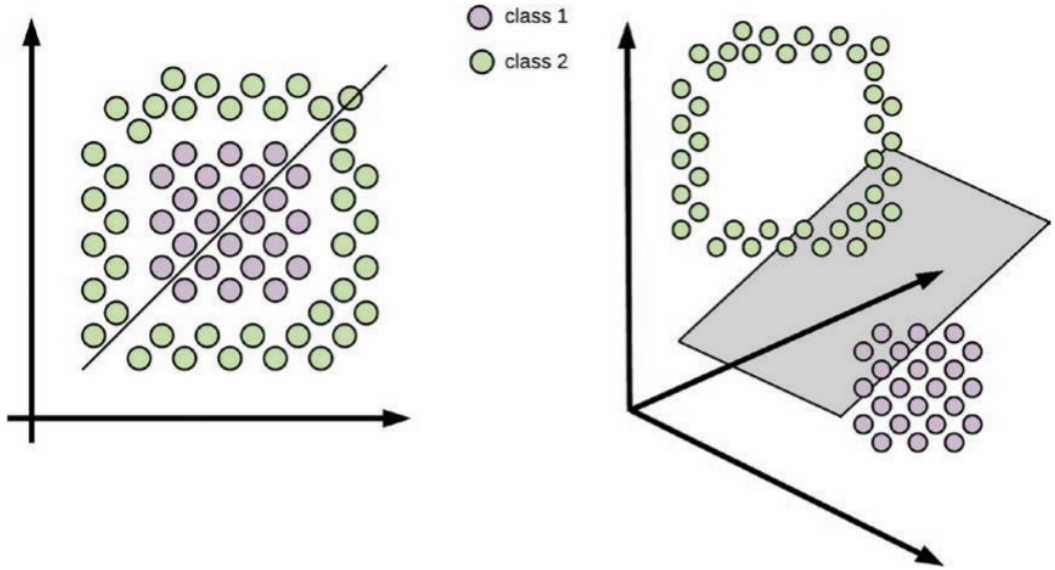
Hiperparámetros en SVC

El siguiente hiperparámetro va relacionado a datos no lineales. Para sistemas no lineales, se utiliza una técnica para añadir un espacio dimensional a este, haciendo que se pueda separar. Esta técnica es llamada Kernel.



Hiperparámetros en SVC

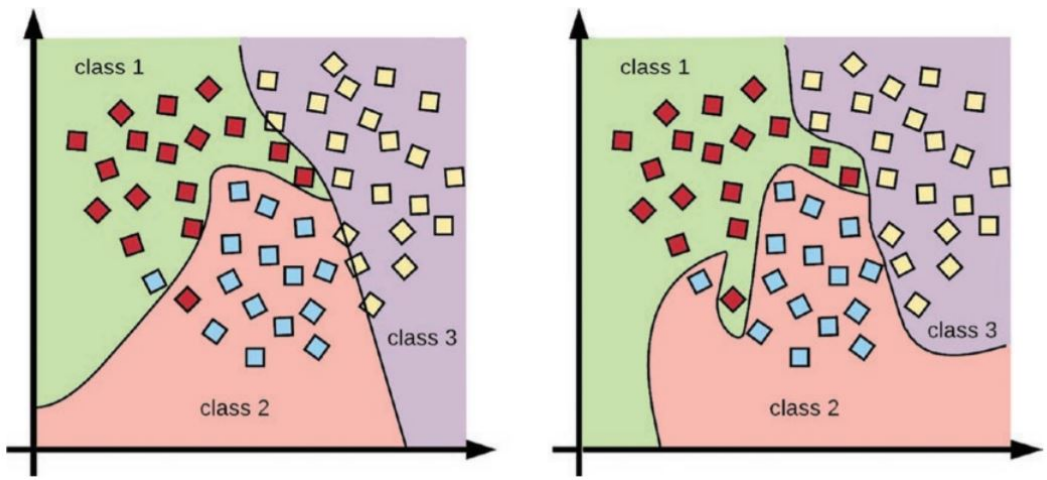
El siguiente hiperparámetro va relacionado a datos no lineales. Para sistemas no lineales, se utiliza una técnica para añadir un espacio dimensional a este, haciendo que se pueda separar. Esta técnica es llamada Kernel.



El añadir espacios dimensionales al conjunto de datos, se logra extendiendo la interacción de los términos a nivel polinomial o añadiendo términos a la interacción. Dependiendo de las dimensiones del dataset esto puede llevar a sobreajustar el modelo.

Hiperparámetros en SVC

El último hiperparámetro para sistemas no lineales es **gamma** cuando el tipo de kernel es radial. El kernel radial es similar a añadir múltiples características al espacio. γ se usa para controlar la flexibilidad de los márgenes de decisión. Un valor bajo de gamma es más flexible respecto a los margenes (No tan exacto, más simple). En cambio un alto valor de gamma vuelve complejo los margenes de decisión.



Computarización

En sklearn existen varios tipos de kernel para SVM, nos enfocaremos en 2 tipos: El kernel lineal y el C-kernel.

El kernel lineal, que no permite incluir kernels polinómicos para data no lineal. Es rápido, tiene solo el hiperparámetro C y se importa como:

```
from sklearn.svm import LinearSVC
```

El C-kernel, permite incluir los hiperparámetros vistos, como el tipo de kernel, el grado polinómico del kernel y el gamma, en el caso radial para data no lineal. Se importa de sklearn como:

```
from sklearn.svm import SVC
```

Ver problema 1.

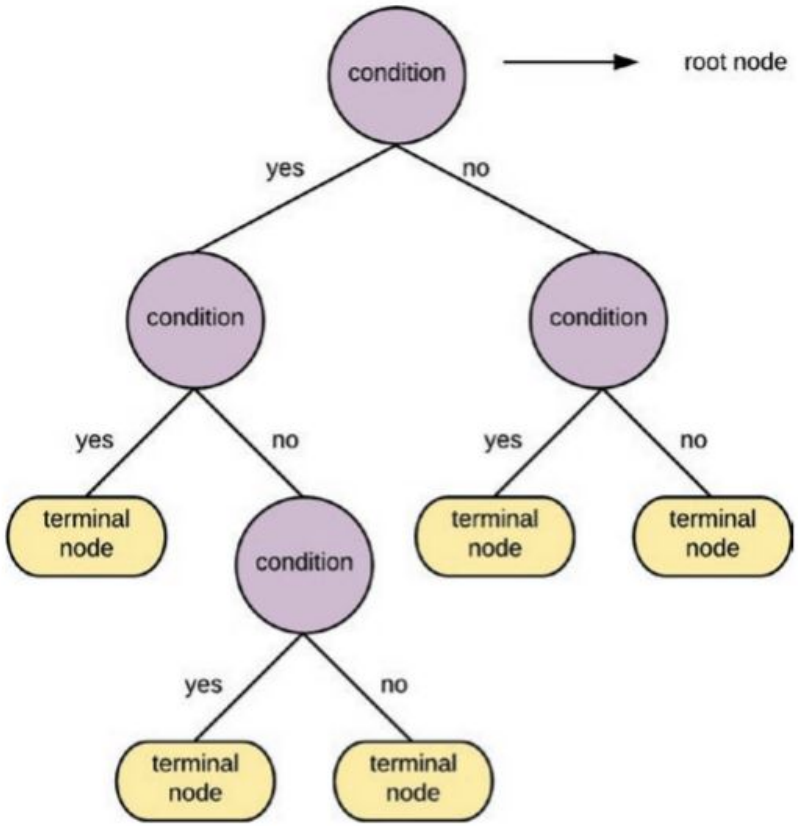
Métodos ensamblados

Los métodos ensamblados o métodos de conjunto son técnicas que combinan varios clasificadores débiles. La metodología dice que si la precisión combinada de estos métodos es mejor que si se trabajaran solos.

Los métodos de ensamblaje se usan tanto para clasificación como para regresión. Ejemplos de estos son **Random Forest(RF)** y **Stochastic Gradient Boosting(SGB)**. Ambos están desarrollados sobre la base de un algoritmo de clasificación debil conocido como **árboles de decisión**

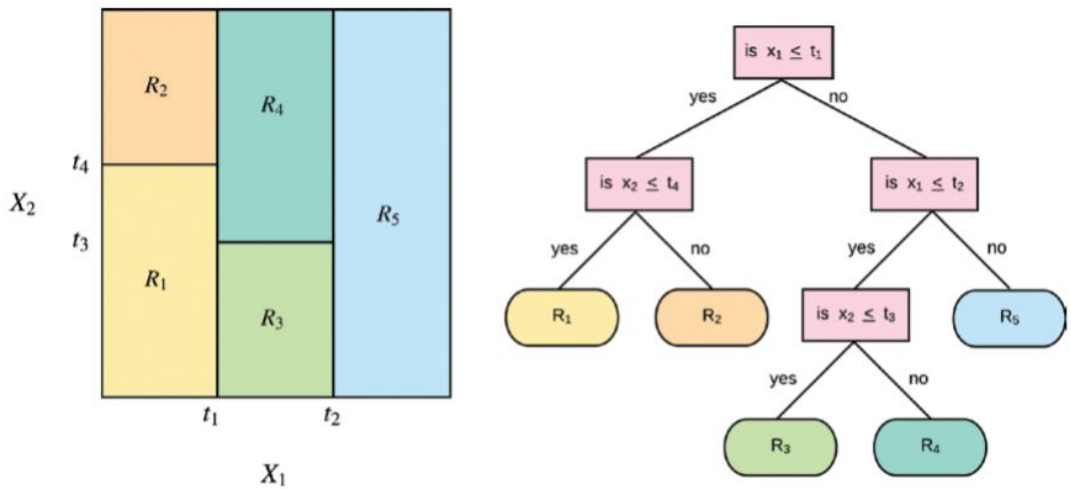
Árboles de decisión

Árboles de decisión aka **CART** (classification and regression trees) pueden ser visualizados como gráficos de flujo de decisión. Una rama conecta nodos en el gráfico, donde el último nodo es llamado un nodo terminal y el nodo superior es conocido como raiz.



Construyendo un árbol

Para construir un árbol de decisión se utiliza un método de separación binaria. Este método primero logra encontrar una primera restricción, dando así una partición binaria. Si se cumple esta restricción, se procede a desarrollar para regresión el promedio de los valores como respuesta, en el caso de clasificación, sería la clase a la que pertenece esta restricción.



Computarización de CART

Para computarizar CART, sklearn ofrece 2 tipos de constructor, tanto para regresión como para clasificación. Si se busca resolver un problema de regresión se usa **DecisionTreeRegressor**:

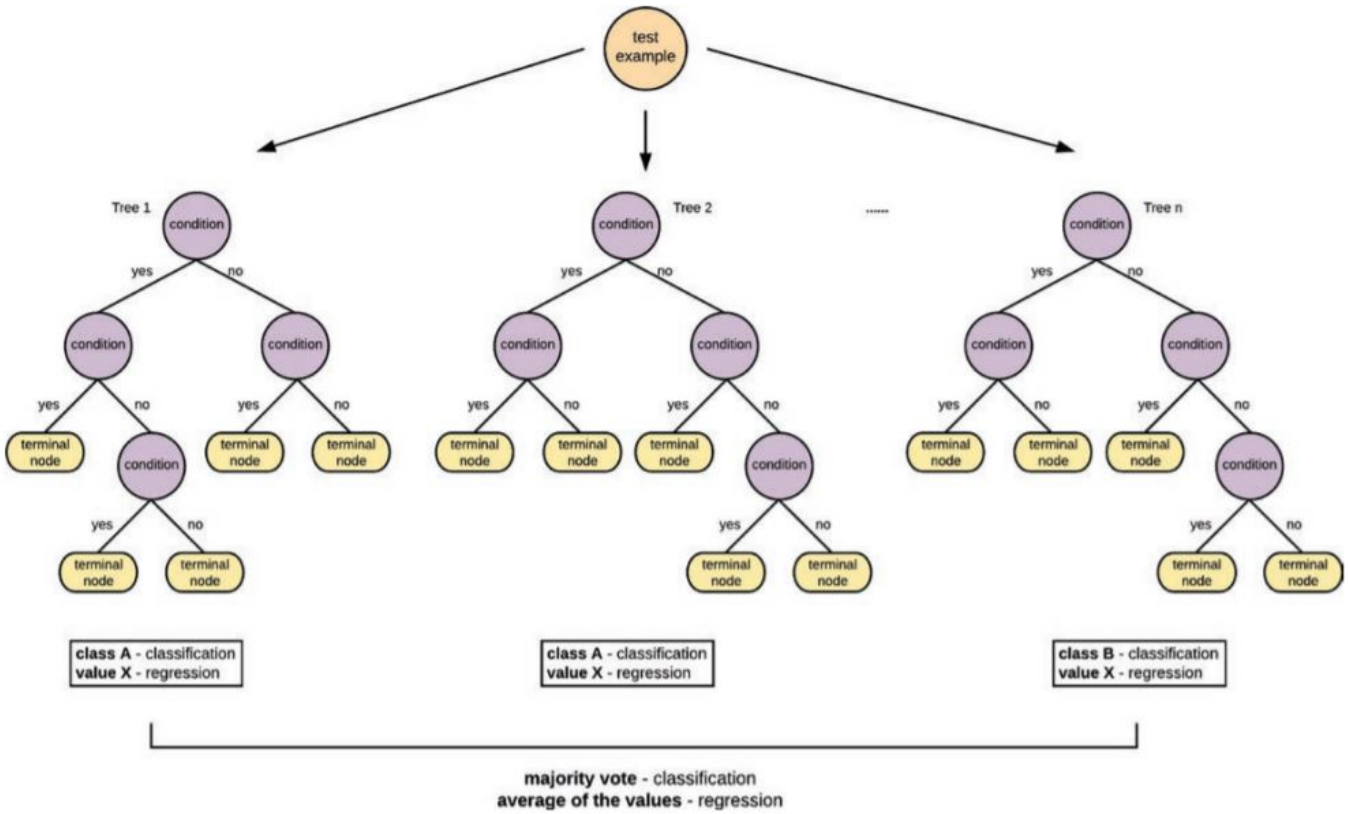
```
from sklearn.tree import DecisionTreeRegressor
```

Si se busca resolver un problema de clasificación se usa **DecisionTreeClassifier**

```
from sklearn.tree import DecisionTreeClassifier
```

Random Forest

Random forest combina varios CART en su metodología, lo que lo convierte en un predictor robusto. La idea tras RF, es hacer una separación de la data en grupos de data aleatorios, cada grupo aleatorio, lleva a cabo una regresión o clasificación mediante CART. Para un nuevo valor, la predicción es el promedio de los diferentes resultados o la cantidad de votos para clasificación.



Computarización e hiperparámetros

RF es un algoritmo robusto, que se puede importar para clasificación desde sklearn como:

```
from sklearn.ensemble import RandomForestClassifier
```

O para regresión como:

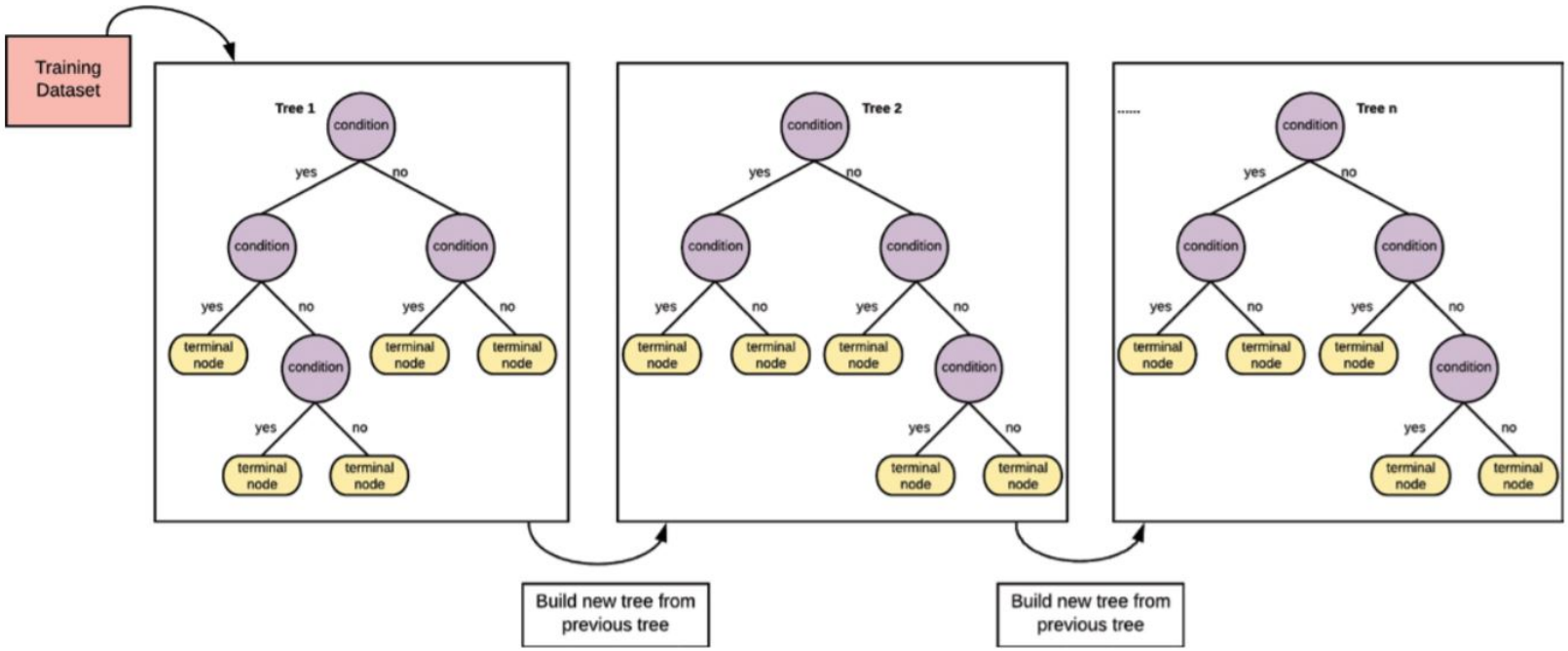
```
from sklearn.ensemble import RandomForestRegressor
```

Los hiperparámteros para RF son:

n_estimators : El número de árboles
max_depth: El máximo de profundidad de un árbol. Si no se brinda, se toma el mínimo de muestras divisibles.
min_samples_split: El mínimo número de muestras dentro de un nodo interno para poder dividir.
boostrap : Si se usará subdatasets creados aleatoriamente o no. Por defecto True. Ver problema 2

Stochastic Gradient Boosting

Es un método robusto, parecido al RF. También genera una partición de datos aleatorio, pero en este caso, el error de un árbol, afecta a los resultados del otro en su función de costo, haciendo que la varianza residual se minimice. Esta información se usa para construir el árbol sucesivo.



Computarización e hiperparámetros

Se puede importar el SGD para clasificación como:

```
from sklearn.ensemble import
GradientBoostingClassifier
```

Se puede importar el SGD para regresión como:

```
from sklearn.ensemble import
GradientBoostingRegressor
```

Los hiperparámteros para SGD son:

learning_rate : Controla el ratio de aprendizaje del modelo. Normalmente es bueno usar entre 0.01 y 0.001.
max_depth: El máximo de profundidad de un árbol.Por defecto es 3