

DIPLOMADO EN CIENCIA DE DATOS

# MODELACIÓN SUPERVISADA SEMANA 4

Facultad de Estudios Superiores Acatlán

# OUTLINE

## **Módulo 2 - Semana 4**

- 1.- Vecinos más cercanos
- 2.- Bayes Ingenuo
- 3.- Árboles de decisión
- 4.- Ensamblés de modelos

# QUIZ 3



**<https://b.socrative.com/login/student/>**  
**Room name: IRENE2290**

# PREGUNTAS ABIERTAS QUIZ 2



## Quiz 2

- ¿Cómo se puede incrementar la complejidad de un modelo de Regresión lineal?
- En pocas palabras, ¿qué es el sobreajuste de datos? y ¿cómo se diferencia del desajuste de datos?
- Describe de manera breve el algoritmo de K-fold Cross-Validation

# PROYECTO FINAL



## Presentaciones (10%)

- 6 minutos cada uno más 1.5 minutos de preguntas
- Presentación debe contener lo más relevante de su proyecto final (a lo más 6 slides)
- Van a pasar por bloques de personas (Más información en classroom)

## Reporte final (30%)

- Reporte escrito (de preferencia en Latex)
- Máximo 15 cuartillas (sin contar lo del módulo 1)
- Portada
- Índice
- Contenido de reporte final: Describir los pasos y principales resultados de desarrollar un modelo supervisado con el set de datos que eligieron
- Una sección para cada parte de la metodología CRISP-DM (Si están utilizando el mismo set de datos que en el módulo uno, presentar esos resultados en el reporte en lugar de rehacerlos)
- Ideal: Probar varios modelos supervisados y seleccionar el mejor
- Citas bibliográficas, Índice de figuras
- Subir reporte y notebook ya ejecutado con todos los resultados del proyecto (solo módulo dos)

# 1.- K-Vecinos más cercanos



# K-VECINOS MÁS CERCANOS

## CONCEPTOS BÁSICOS



Este modelo utiliza métodos de similitud, pues intenta hacer una predicción con base en el valor de la variable respuesta que tienen sus vecinos más cercanos



### Definición

- Target numérica/catógorica → Regresión/Clasificación
- Un set de datos está compuesto por el par  $(x_n, y_n)$
- NO tiene un problema de optimización
- Intenta ver la similitud de cada observación con los vecinos más cercanos

#### Clasificación

Para cada nueva observación a predecir, asignará el valor de la variable target más frecuente del conjunto de vecinos más cercanos

#### Regresión

Para cada nueva observación a predecir, asignará el valor del promedio de las variables target del conjunto de vecinos más cercanos

### Características

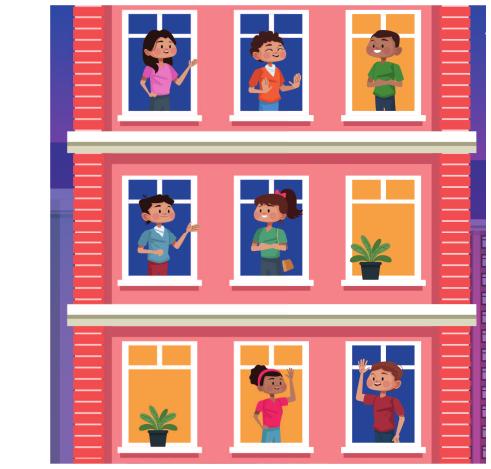
- Es un método de similitudes
- Funciona bien con sets de datos con pocas dimensiones
- Suposición: observaciones que están cerca una de la otra son parecidas, por lo tanto pueden tomar el mismo valor de la variable target
- Hyper parámetro a optimizar: Número de vecinos más cercanos K
- Calcula la distancia entre observaciones, dependiendo de el valor de K y el tipo de problema supervisado, asignará una etiqueta con base en la similitud con los K vecinos más cercanos
- Afectado por la maldición de las dimensiones

# K-VECINOS MÁS CERCANOS

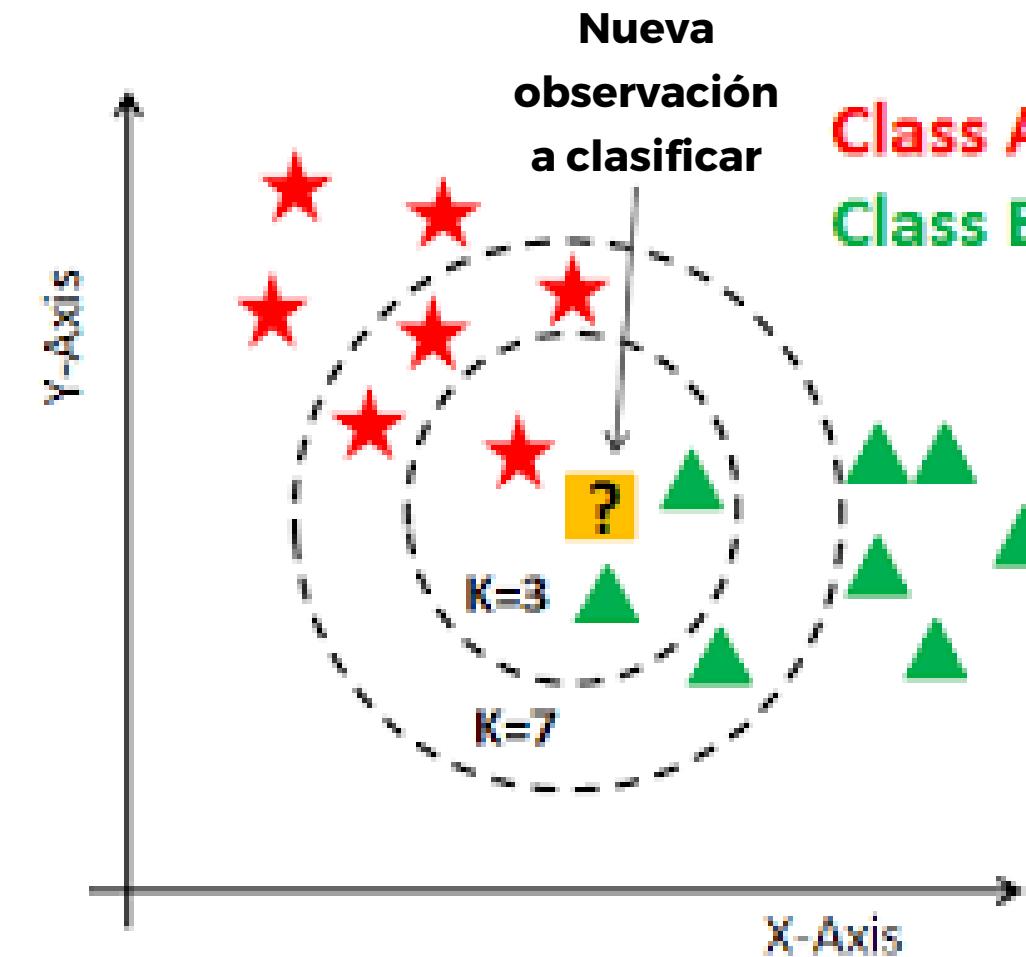
## INTUICIÓN - MODELO PARA CLASIFICAR DATOS



Este modelo intenta asignar una etiqueta con base en la etiqueta más frecuente que tienen los vecinos más cercanos de cada observación



### Intuición gráfica\*



Medimos la distancia entre la observación a predecir y el resto de observaciones. Después, determinamos los K vecinos más cercanos

**¿Qué pasa si K=3?**

Se asigna la  
etiqueta de la  
clase B

**¿Qué pasa si K=7?**

Se asigna la  
etiqueta de la  
clase A

**¿Cómo sería si el problema fuera de Regresión?**

# K-VECINOS MÁS CERCANOS

## FORMAS DE MEDIR LA DISTANCIA ENTRE OBSERVACIONES



Existen diferentes maneras de medir distancias entre observaciones, a continuación se enlistan las más importantes



### Métricas para medir distancias

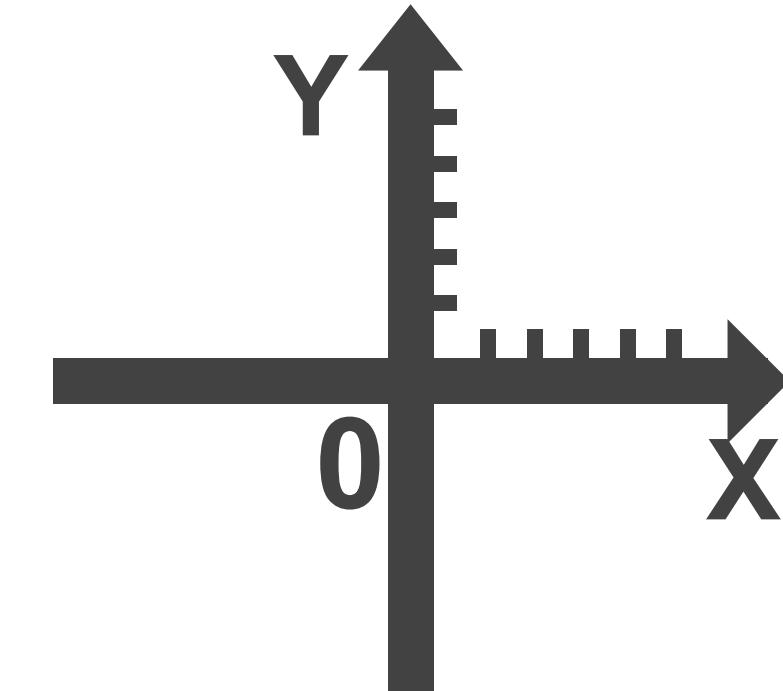
Distancia Euclideana (Norma L2)

Distancia Manhattan (Norma L1)

Distancia Jacard

Distancia Coseno (mide el ángulo de separación entre dos vectores)

Distancia Levenshtein (para medir similitud en strings)



# K-VECINOS MÁS CERCANOS

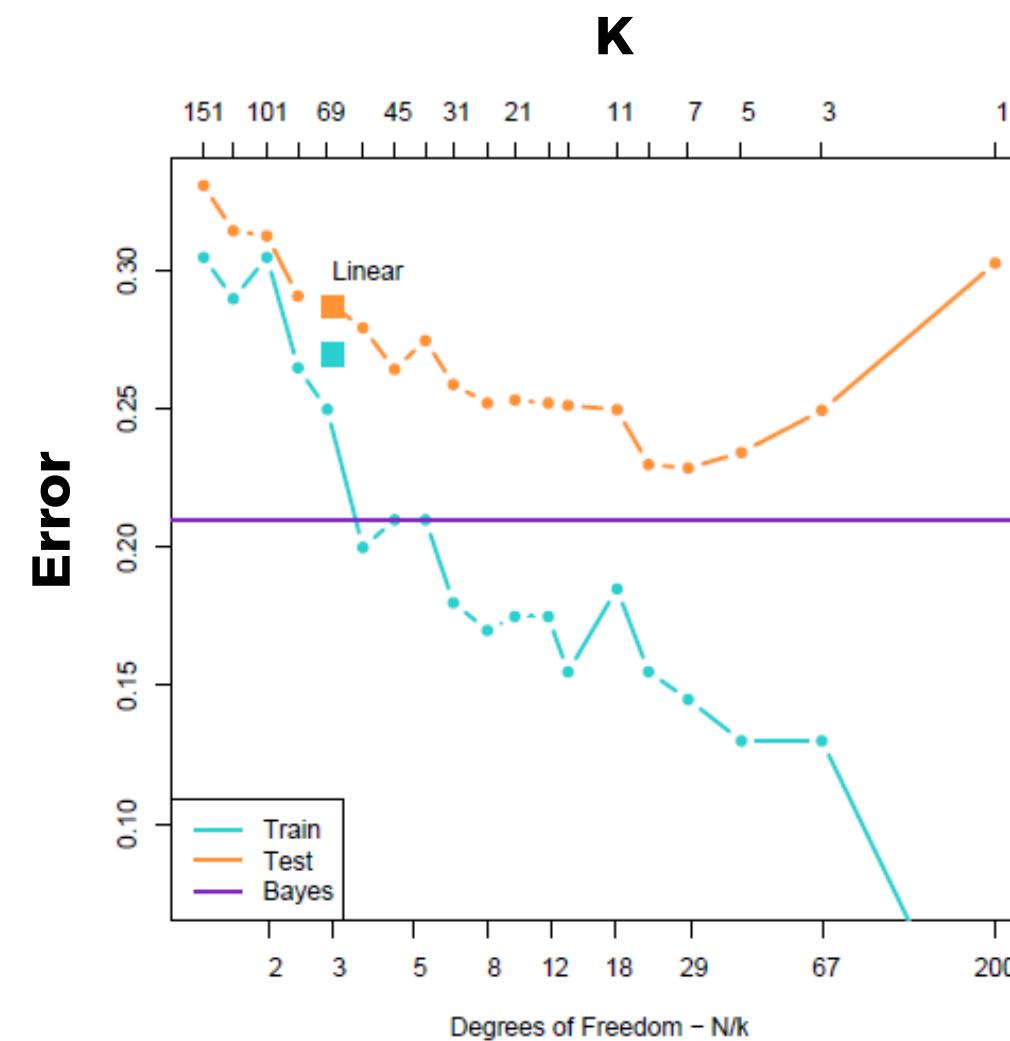
## TRADE-OFF ENTRE SESGO Y VARIANZA



Dependiendo del valor de K (número de vecinos), nuestro modelo será más o menos complejo. Dicho hyper parámetro debe ser optimizado



### Error en el set de datos de entrenamiento y validación versus el número de vecinos más cercano (K)



- Si K toma valores grandes:
  - El modelo estará tomando el valor más frecuente de un conjunto grande de observaciones. Por lo tanto, entre más grande sea K, más simple será nuestro modelo
  - Podríamos tener alto sesgo pero baja varianza
- Si K toma valores pequeños:
  - Entre más pequeña será K, más complejo será el modelo. Lo anterior sucede pues el modelo asignará una clasificación con base en un número reducido de observaciones (no tantos patrones aprendidos)
  - El modelo podría tener bajo sesgo pero alta varianza

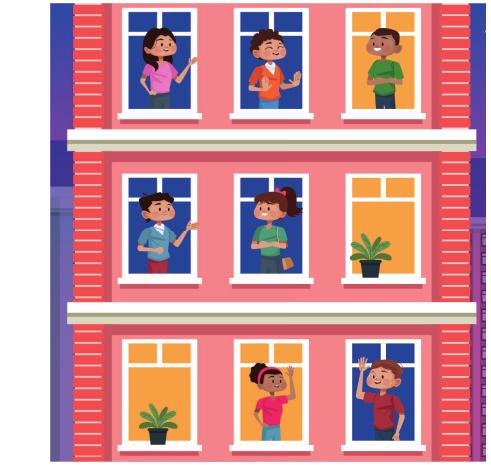
¿Por qué la función del error en el set de datos de validación tiene forma de U?

# K-VECINOS MÁS CERCANOS

## COSAS QUE PUEDEN AFECTAR ESTE MODELO



Al ser un modelo que utiliza técnicas de similitud entre observaciones, se puede ver afectado por la escala de las variables y el número de dimensiones en el set de datos



### La maldición de las dimensiones

- **¿Qué es?**
  - Hacer que un modelo generalice de manera adecuada se vuelve exponencialmente difícil conforme va creciendo la dimensionalidad del set de datos
  - Tendremos más ruido conforme aumentemos las dimensiones
- **Como solucionarlo:**
  - Reducir dimensiones con las siguientes técnicas:
    - PCA (Principal Components Analysis)
    - Selección de Variables
    - SVD (Singular Value Decomposition)

### Escalas diferentes de variables

- **¿Qué es?**
  - Variables con diferentes escalas pueden confundir al método de similitud que este modelo utiliza
  - Las métricas para medir distancias se verán inclinadas a las variables con mayor magnitud
- **Como solucionarlo:**
  - Normalizar variables
  - Estandarizar variables

## 2.- Bayes Ingenuo

# BAYES INGENUO

## CONCEPTOS BÁSICOS

$$P(A|B) = \frac{P(A, B)}{P(B)}$$



Este modelo asume independencia de las variables  $X_i$  dada la variable target  $y$

### Definición

- Target categórica → Clasificación
- Un set de datos está compuesto por el par  $(x_n, y_n)$
- No se tiene que resolver un problema de optimización
- En vez de eso se calcula la siguiente probabilidad

$$\mathbb{P}(y_n | x_n)$$

### Características

- Clasificador muy simple
- Asume que las variables de entrada  $X_i$  son independientes dada las clases de la variable target, de ahí el nombre de "Ingenuo"
- Toma en cuenta la información de todas las variables de entrada
- Es muy eficiente en términos de almacenamiento y tiempo de ejecución pues el entrenamiento solo consiste en almacenar conteos de ocurrencias de clases y de variables de entrada
- A pesar de ser un clasificador ingenuo, tiene un buen performance en tareas de clasificación, como lo es el clasificar si un email es spam o no

# BAYES INGENUO

## INTUICIÓN

$$P(A|B) = \frac{P(A, B)}{P(B)}$$



Este modelo asume independencia de las variables  $X_i$  dada la variable target  $y$

### Forma de calcular la probabilidad de ocurrencia para cada clase de la variable target $y$

$$P(y_n | x_n) = \frac{P(x_n | y_n) P(y_n)}{P(x_n)}$$

Puede ser estimada al contar la proporción de ejemplos de entrenamiento en la clase  $y$  en donde aparece la variable de entrada  $X_i$

Asume independencia de las variables de entrada

$$= \frac{P(x_1 | y_n) P(x_2 | y_n) \dots P(x_D | y_n) P(y_n)}{P(x_1) P(x_2) \dots P(x_D)}$$

Puede ser estimada al contar el número de ocurrencias de la clase con respecto a todos los ejemplos de entrenamiento

# 3.- Árboles de decisión



# ÁRBOLES DE DECISIÓN

## CONCEPTOS BÁSICOS



Este modelo utiliza un algoritmo de tipo avaro (greedy) para partir el espacio de variables de entrada, tal que se reduzca la impureza de las particiones



### Definición

- Target numérica/catógorica → Regresión/Clasificación
- Un set de datos está compuesto por el par  $(x_n, y_n)$
- Utiliza la función de costo del algoritmo de entrenamiento CART (Classification and Regression Tree). Ver referencia

$$J(k, t_k) = \frac{m_{\text{left}}}{m} G_{\text{left}} + \frac{m_{\text{right}}}{m} G_{\text{right}}$$

where  $\begin{cases} G_{\text{left/right}} \end{cases}$  measures the impurity of the left/right subset,  
 $\begin{cases} m_{\text{left/right}} \end{cases}$  is the number of instances in the left/right subset.

### Características

- Es un algoritmo de tipo greedy
- Es un componente fundamental de los bosques aleatorios
- Suelen ser llamados modelos de caja blanca, pues son muy sencillos de interpretar
- Son ampliamente usados
- Para problemas de clasificación, se utiliza la métrica Gini o la métrica de entropía para medir la impureza de las particiones creadas en cada nivel del árbol de decisión
- CART siempre partirá el espacio en dos nodos hijo
- Existe otro algoritmo llamado ID3 que tiene la capacidad de partir el espacio en más de dos nodos hijos

# ÁRBOLES DE DECISIÓN

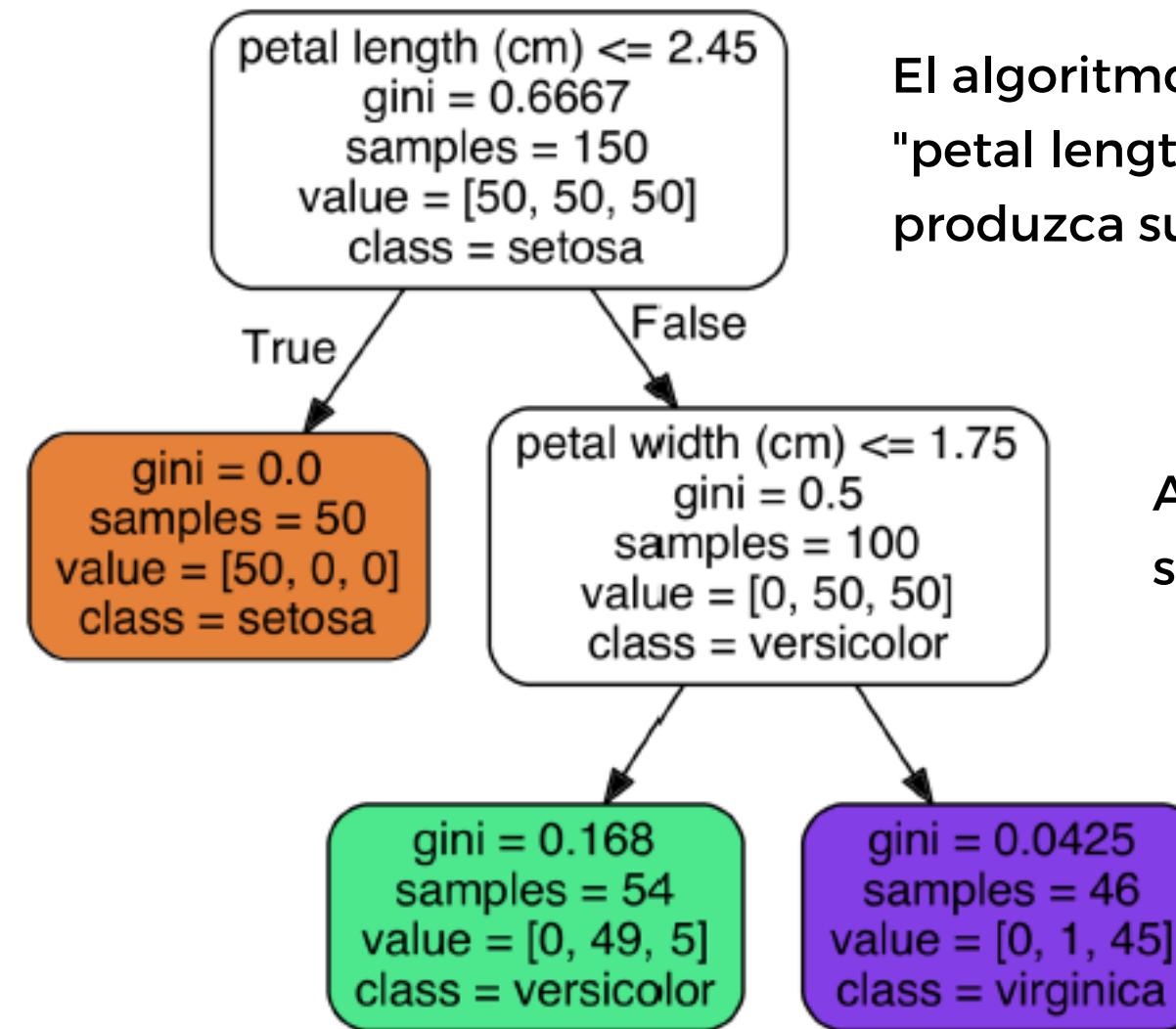
## INTUICIÓN



Este modelo utiliza un algoritmo de tipo avaro (greedy) para partir el espacio de variables de entrada, tal que se reduzca la impureza de las particiones



### Árbol de decisión sencillo (CART)



El algoritmo primero parte los datos en dos subconjuntos utilizando la variable "petal length" y el threshold "2.45". Elige el par (petal length, 2.45) tal que produzca subconjuntos puros

Ahora parte cada uno de los subconjuntos... y así sucesivamente



Criterio para parar: una vez que llega a la máxima profundidad establecida, o si ya no existe una partición tal que los subconjuntos resultantes tengan mínima métrica de impureza

# ÁRBOLES DE DECISIÓN

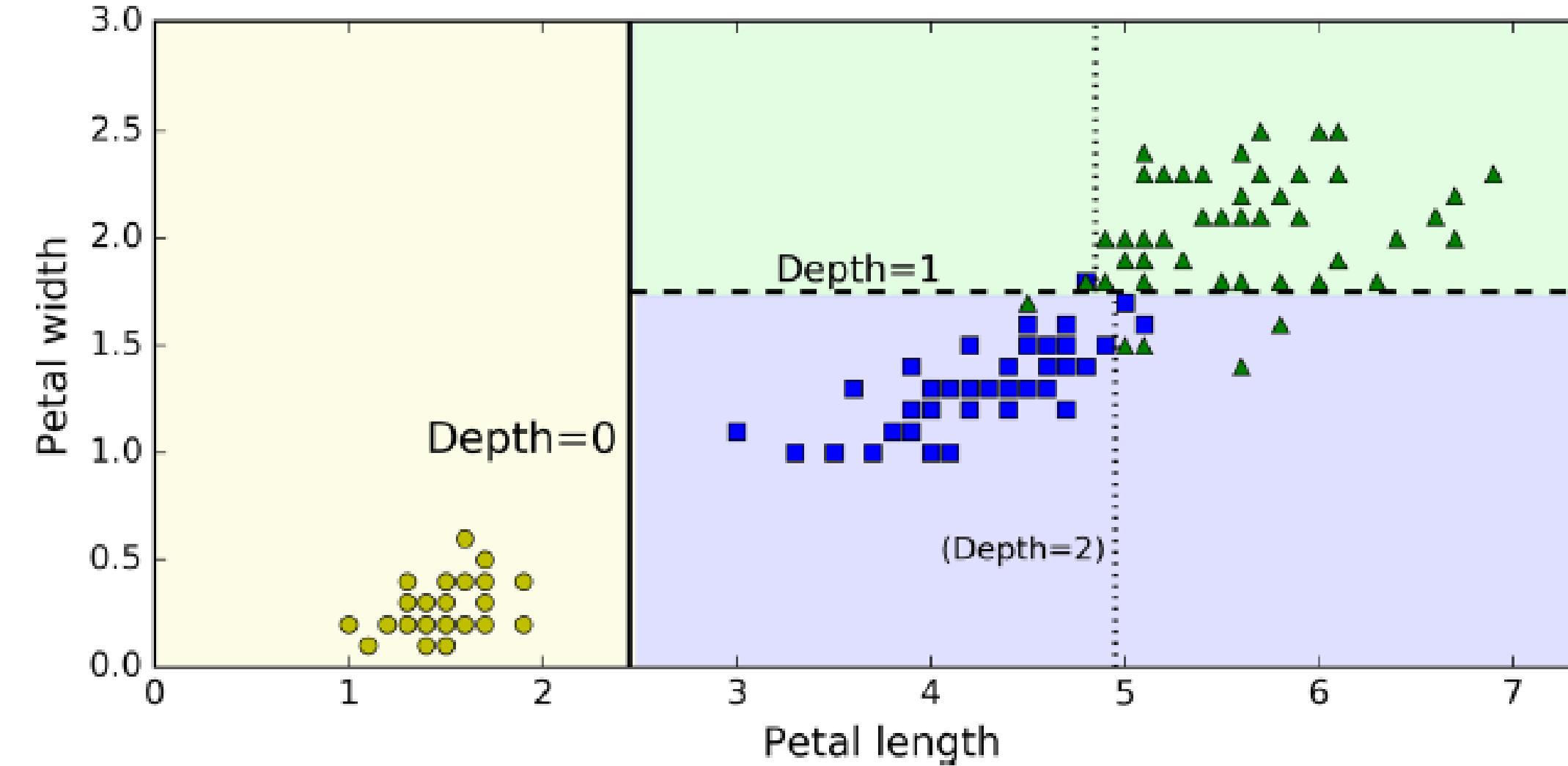
## INTUICIÓN



Este modelo utiliza un algoritmo de tipo avaro (greedy) para partir el espacio de variables de entrada, tal que se reduzca la impureza de las particiones



### Árbol de decisión sencillo (CART)



# ÁRBOLES DE DECISIÓN

## COSAS QUE PUEDEN AFECTAR - SOBREAJUSTE A LOS DATOS



Los árboles de decisión sin ningún tipo de restricción son propensos a sobreajustar a los datos de entrenamiento



### Formas de regularización

- Si un árbol de decisión se entrena sin ningún tipo de restricción, entonces el modelo se adaptará por completo al set de datos de entrenamiento (sobreajuste)
- Este modelo es usualmente llamado "modelo no paramétrico". Lo anterior no es porque el modelo no tenga parámetros, si no que el número de parámetros no se puede determinar antes de entrenar un modelo. **¿Qué pasa si un modelo de aprendizaje tiene muchos parámetros?**
- En contraste, un modelo paramétrico, tiene muy pocos parámetros. **¿Qué pasa si un modelo de aprendizaje de máquina tiene pocos parámetros?**
- Para prevenir el sobreajuste, se tienen que restringir los grados de libertad del árbol de decisión (regularización)
- Los hyper parámetros de regularización dependen del tipo de modelo, pero algo comúnmente usado es restringir la máxima profundidad del modelo
- Reducir la máxima profundidad que un árbol de decisión puede tener, ayuda a reducir el riesgo de sobreajustar a los datos de entrenamiento
- Otras formas de prevenir el sobreajuste es al podar el árbol de decisión

# ÁRBOLES DE DECISIÓN

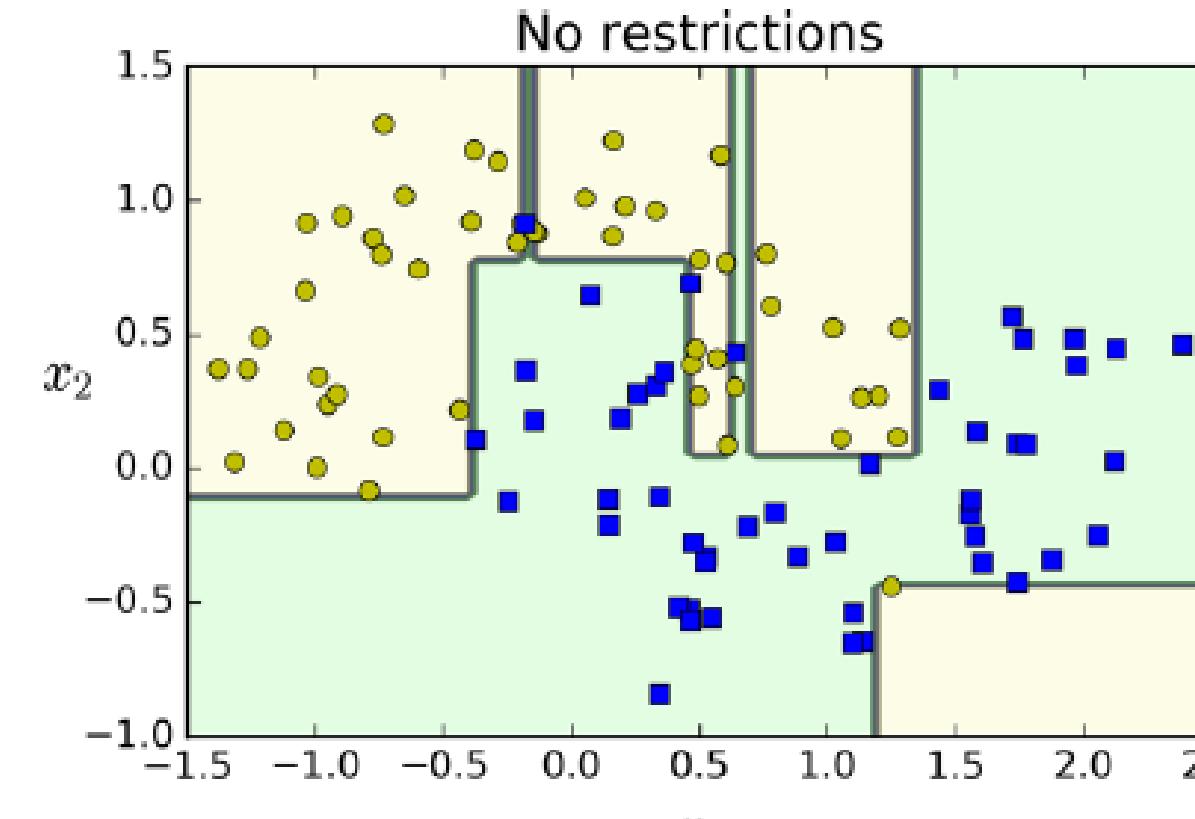
## SOBREAJUSTE A LOS DATOS DE ENTRENAMIENTO



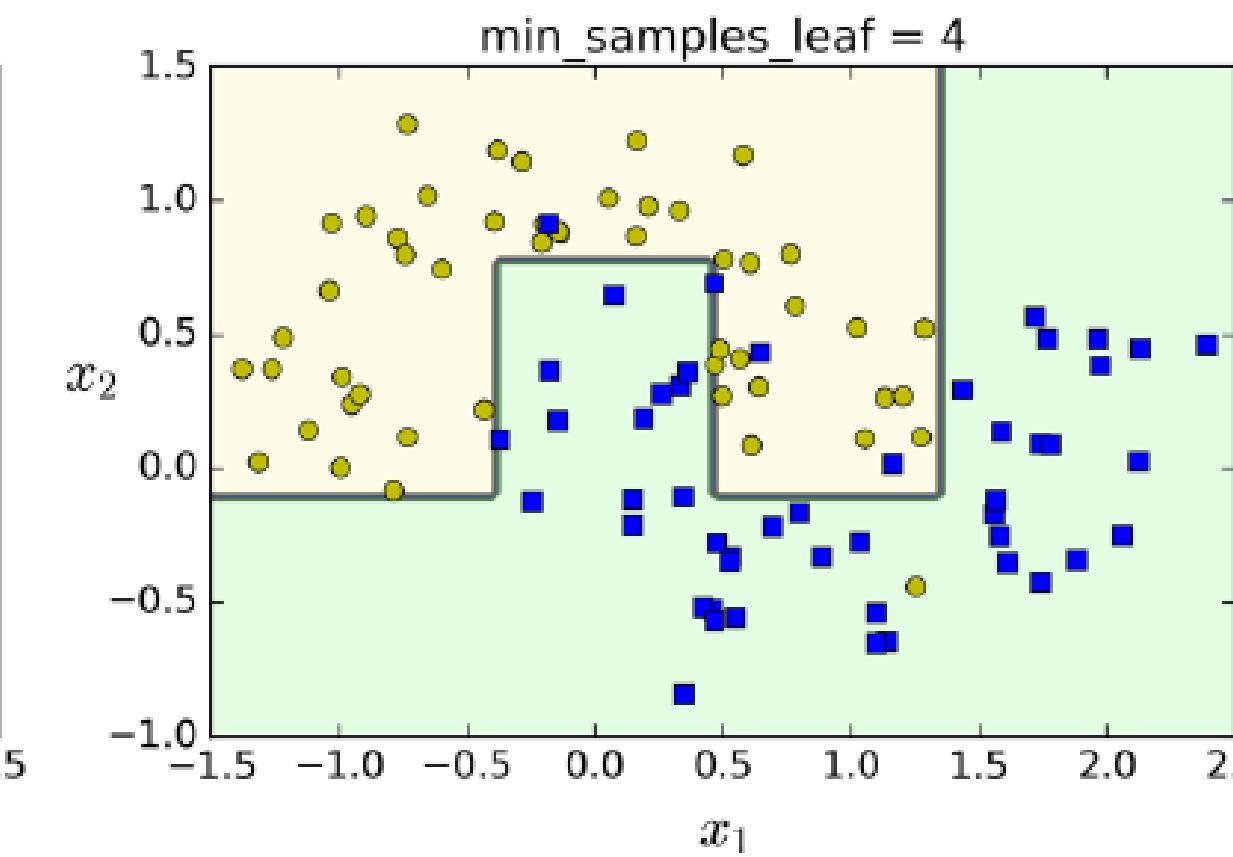
Los árboles de decisión sin ningún tipo de restricción son propensos a sobreajustar a los datos de entrenamiento



### Árboles de decisión sin y con restricciones



Sobreajuste



Modelo generaliza mejor

# ÁRBOLES DE DECISIÓN

## COSAS QUE PUEDEN AFECTAR - INESTABILIDAD DEL SET DE ENTRENAMIENTO

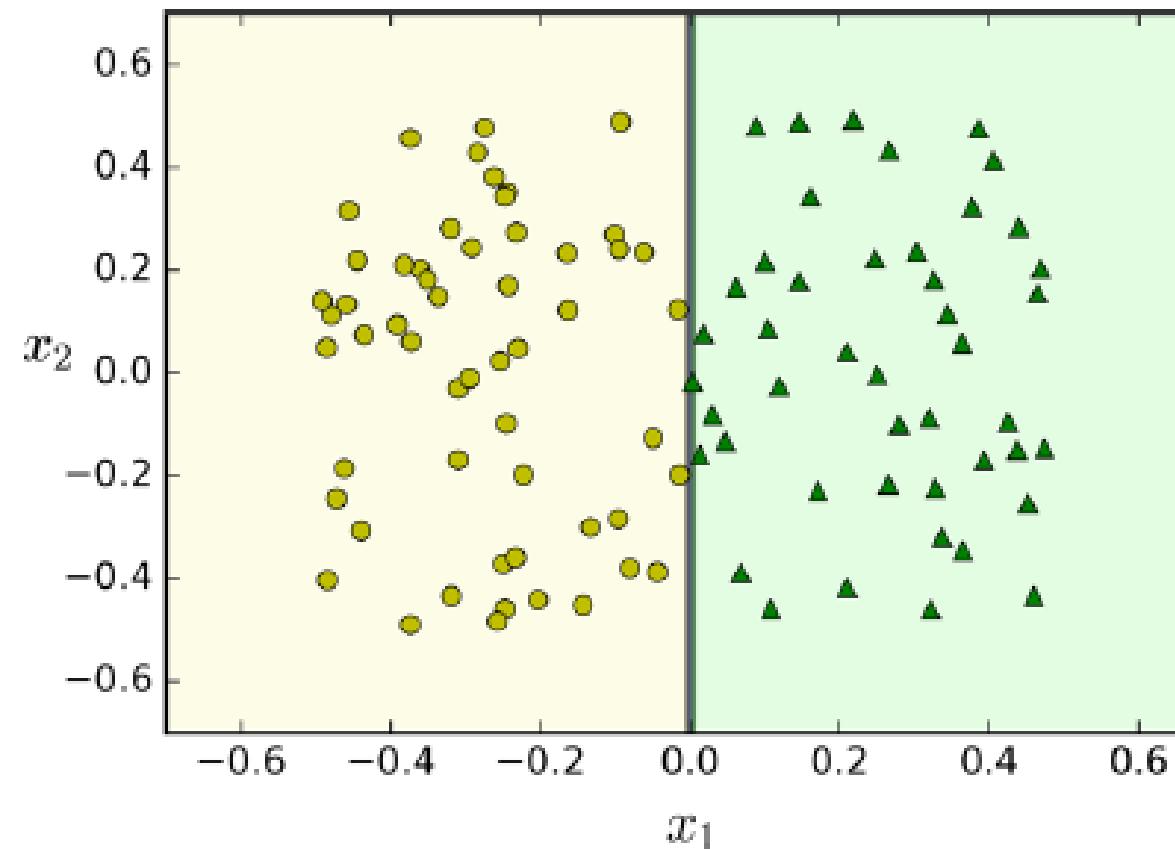


Los árboles de decisión son muy sensibles a pequeñas variaciones en el set de datos de entrenamiento

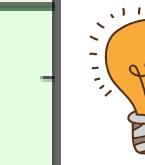
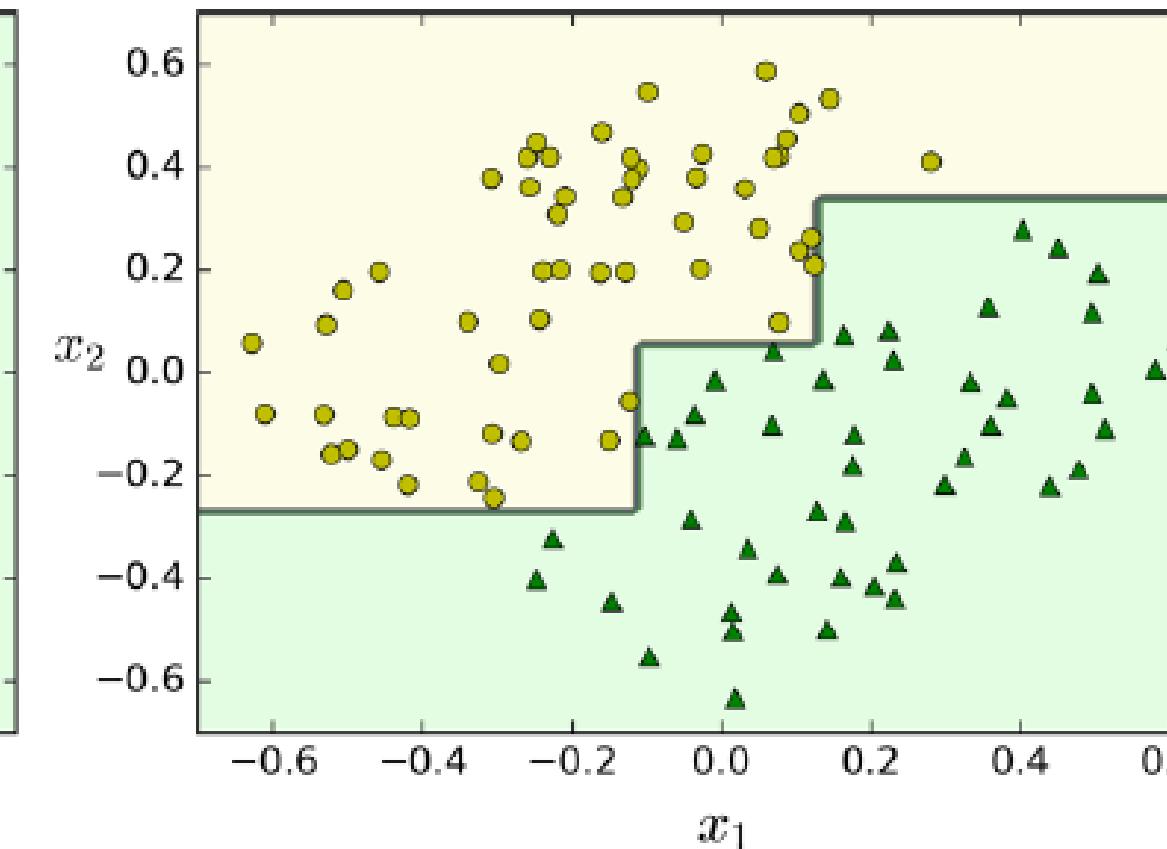


### Problemas de estabilidad

Modelo ama los espacios de decisión ortogonales



Si rotamos el set de datos 45 grados, el espacio de decisión se ve afectado



Forma de mitigarlo:  
Usando PCA, que normalmente hace que los datos tengan una mejor orientación

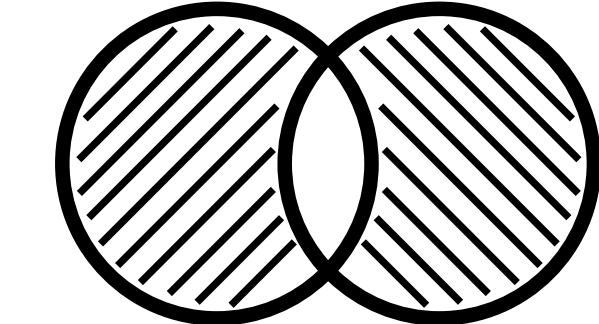
## 4.- Ensamblles de modelos

# ENSAMBLES

## CONCEPTOS BÁSICOS



Este método agrega diferentes clases de modelos para buscar maximizar el performance (minimizar el error) de un problema supervisado



### Definición

- Target numérica/categórica → Regresión/Clasificación
- Un set de datos está compuesto por el par  $(x_n, y_n)$
- No existe una función de costo, pero intenta agregar los resultados de diferentes modelos de predicción



**Objetivo:** Obtener modelos diversos que al ser agregados mejoren el performance total del problema supervisado a resolver

### Características

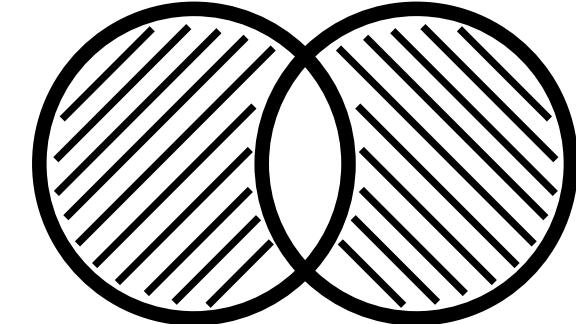
- Utiliza la sabiduría de la multitud para hacer que el modelo obtenga buenas métricas de performance
- Funcionan mejor cuando los modelos son independientes entre ellos
- Un Ensamble de árboles de decisión es llamado un modelo de Bosque aleatorio
- Existen diferentes tipos de ensambles:
  - Bagging, Boosting, Stacking
- Para los clasificadores, lleva a cabo una votación acerca de la predicción de cada modelo. Asigna la que fue más votada
- Normalmente un ensamble de  $k$  modelos logra mejores métricas de performance que el modelo individual con mejor performance

# ENSAMBLES

## INTUICIÓN



Este método agrega diferentes clases de modelos para buscar maximizar el performance (minimizar el error) de un problema supervisado



### Sabiduría de las multitudes

- Supongamos que queremos tener una solución a una pregunta compleja
- Supongamos que hacemos la pregunta a miles de personas al azar y utilizamos una función de agregación **¿Cómo cuál?** para determinar la solución
- Dicha solución resultará ser mejor que la solución dada por un experto en el tema
- Lo mismo sucede si utilizamos una función de agregación para las predicciones de diferentes modelos supervisados

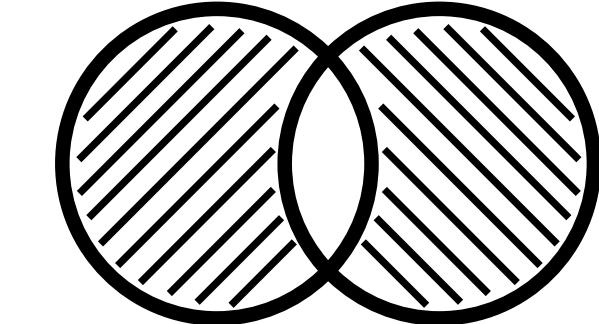


# ENSAMBLLES

## INTUICIÓN

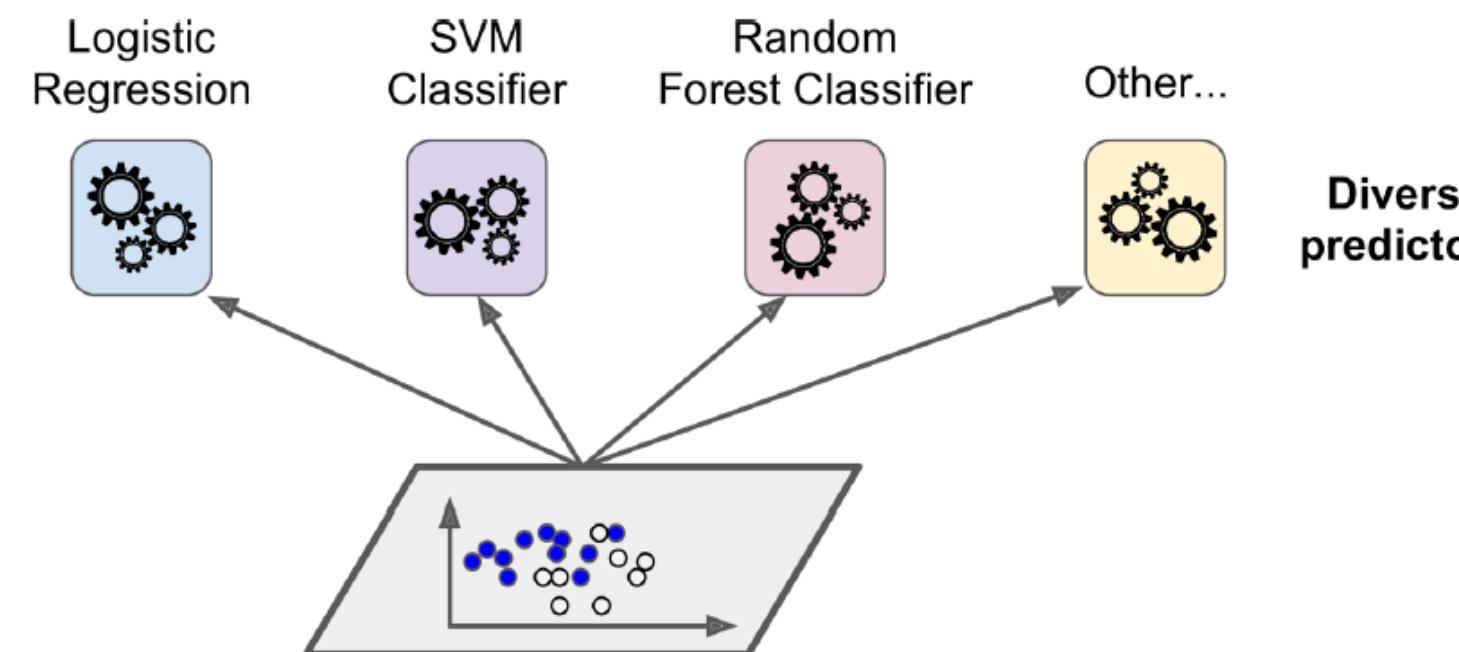


Este método agrega diferentes clases de modelos para buscar maximizar el performance (minimizar el error) de un problema supervisado



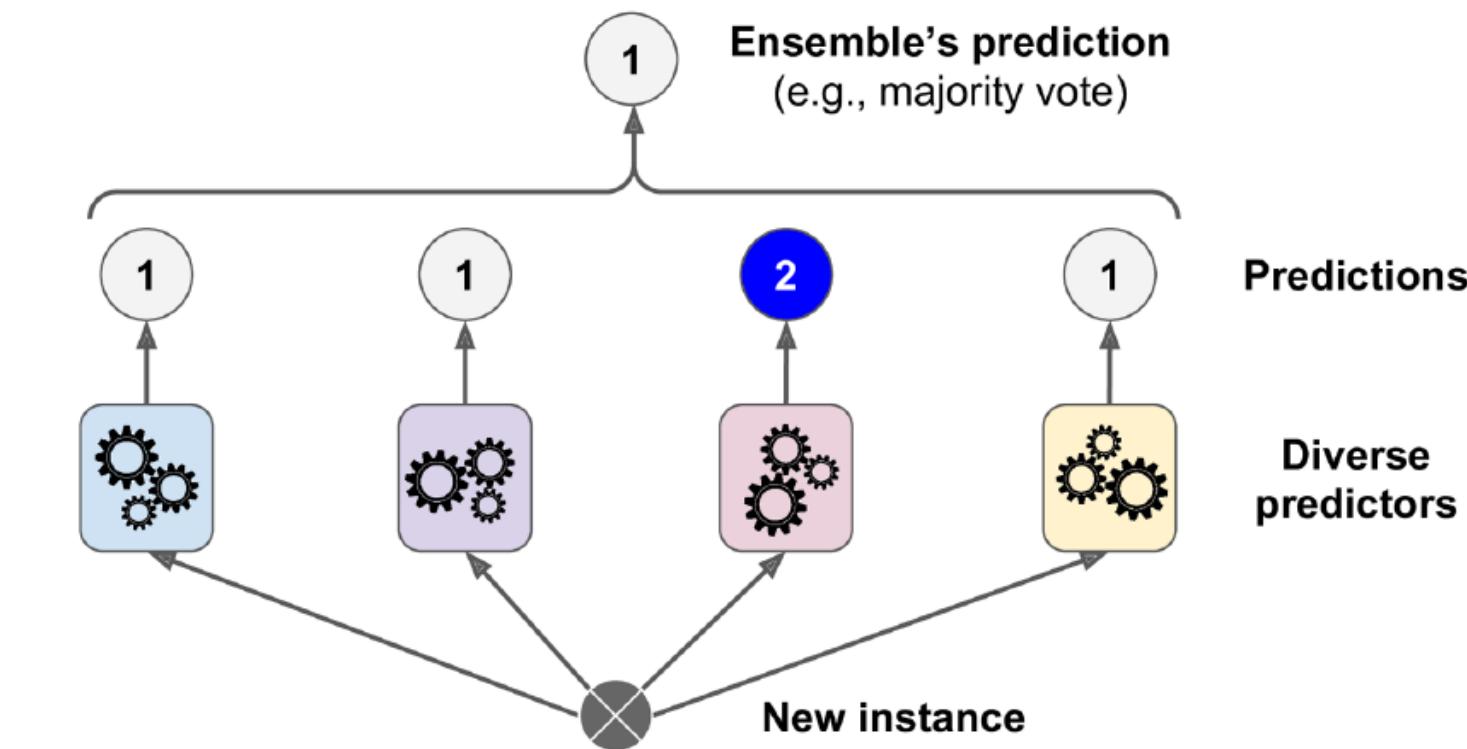
### Predicciones diversas

- Consiste en agregar las predicciones de modelos de aprendizaje de máquina ya entrenados para determinar las predicciones óptimas



### Voto mayoritario

- También llamado votación dura
- Predice la clase más votada por los modelos

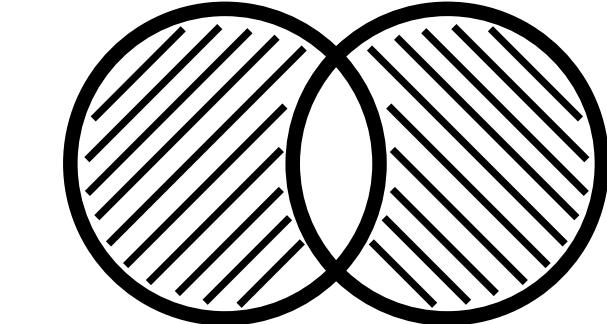


# ENSAMBLÉS

## INTUICIÓN

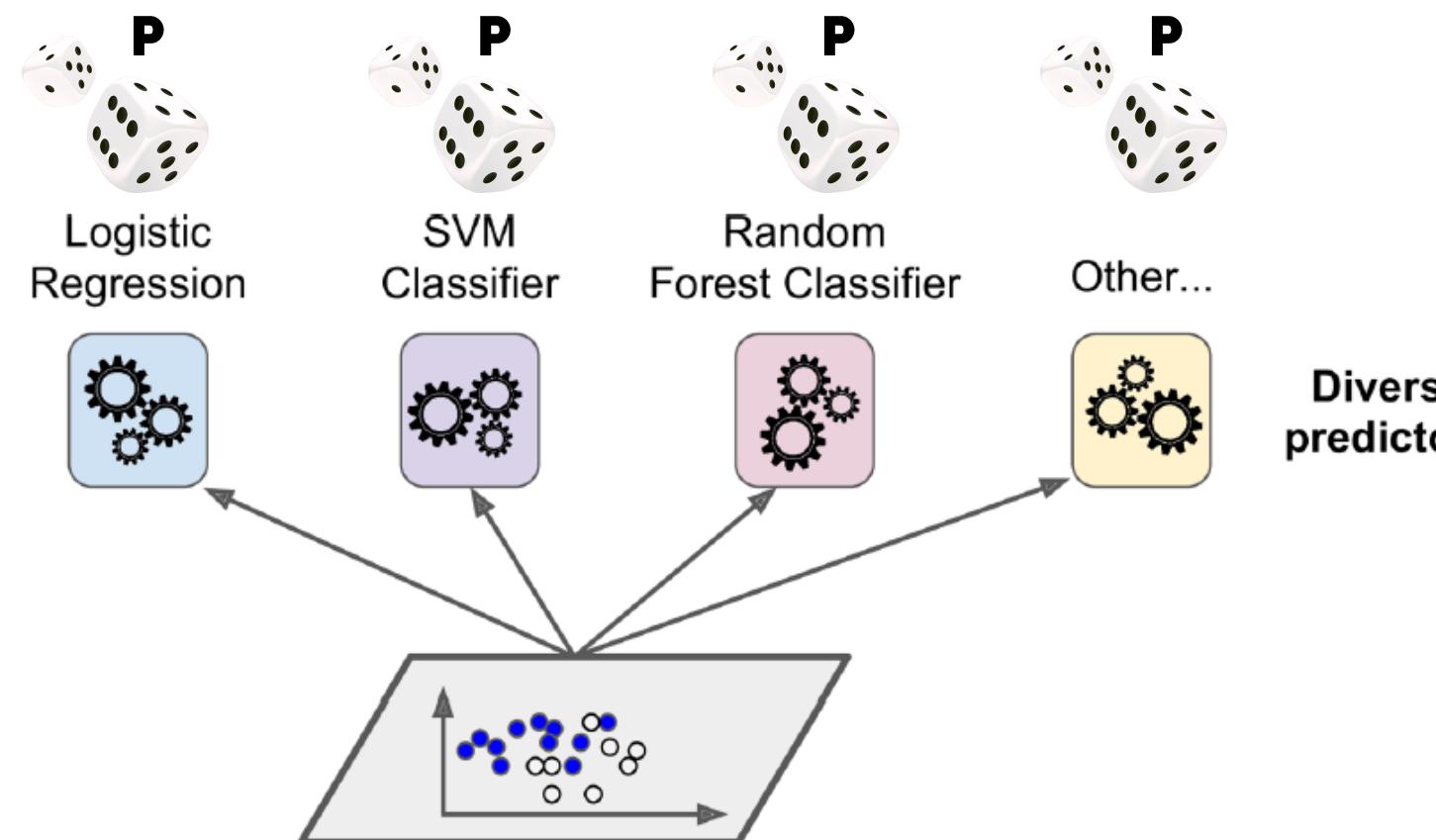


Este método agrega diferentes clases de modelos para buscar maximizar el performance (minimizar el error) de un problema supervisado



### Voto suave

- Si los modelos entrenados permiten obtener un score de probabilidad, entonces se calculará el promedio del score de probabilidad de todos los modelos

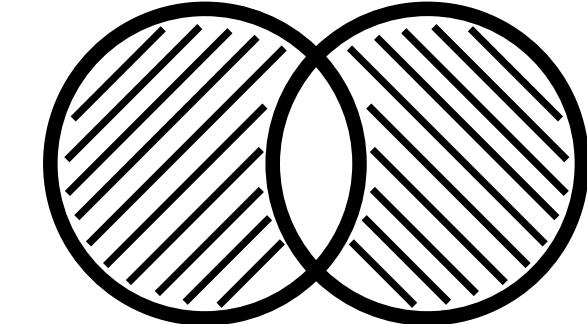


# ENSAMBLES

## TIPOS DE ENSAMBLES



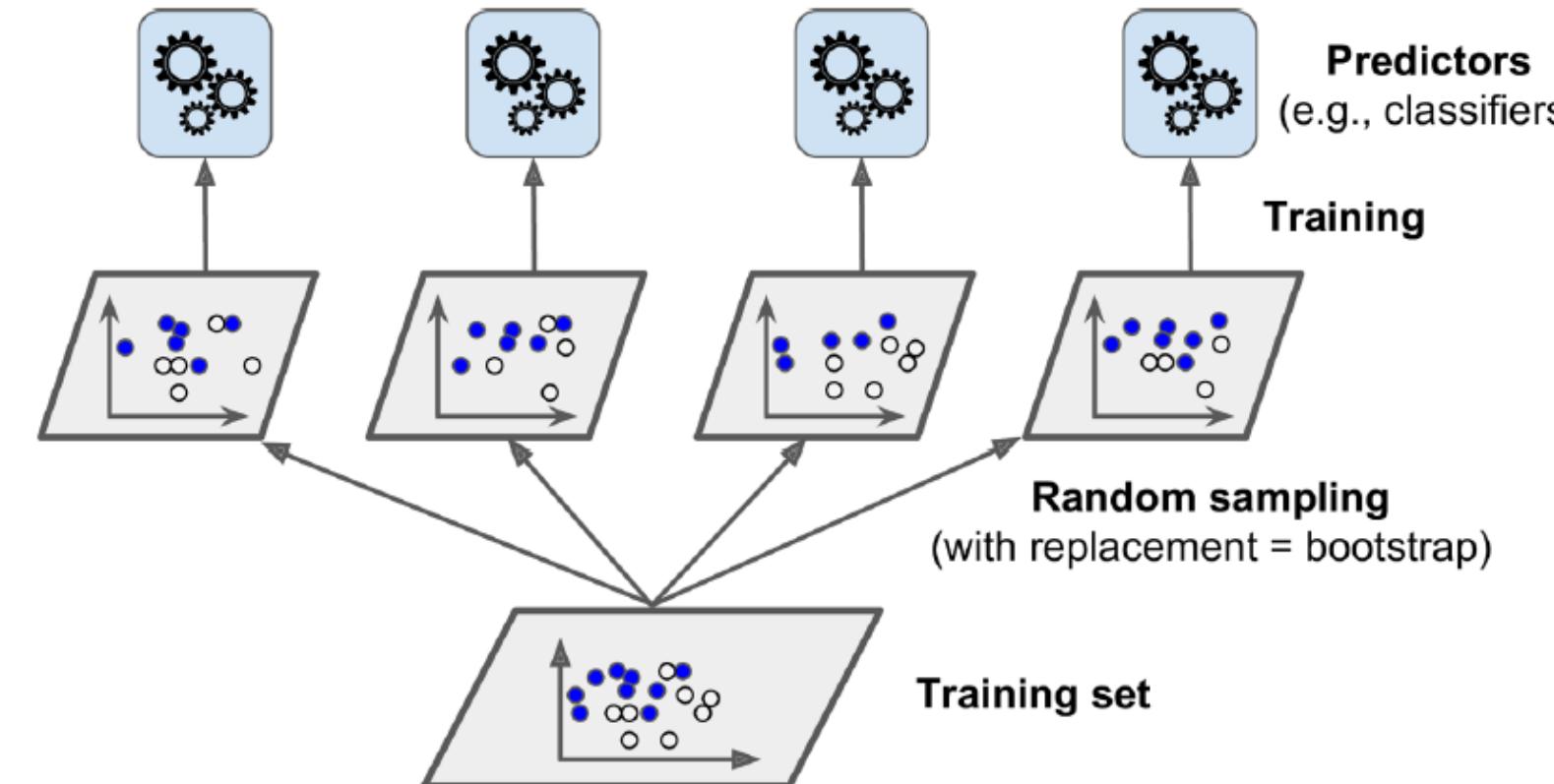
Estos métodos consisten en entrenar un único modelo con diferentes sub-muestras aleatorias del set de datos de entrenamiento



### Bagging

- Cuando el muestreo aleatorio es llevado a cabo con reemplazo
- Bagging es el nombre corto de Bootstrap Bagging

Para realizar predicciones se pueden utilizar las votaciones duras y suaves, es decir se utiliza una función de agregación para determinar la predicción óptima



### Pasting

- Cuando el muestreo es ejecutado sin reemplazo

El agregar las predicciones de cada modelo hace que se reduzca el sesgo y la varianza

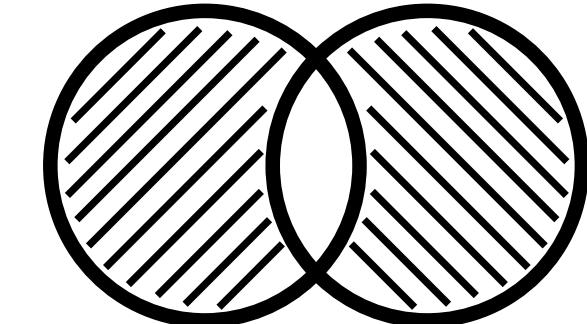


# ENSAMBLES

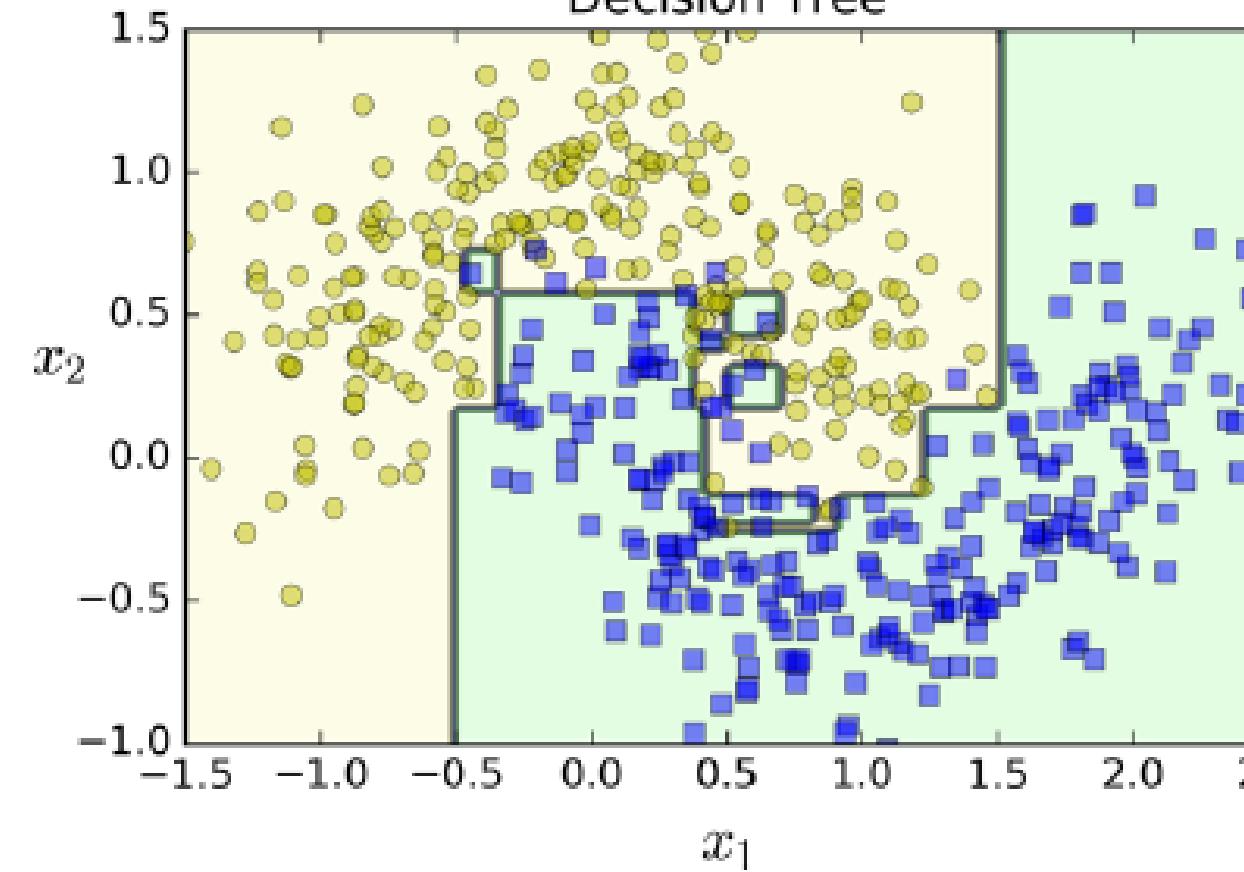
## TIPOS DE ENSAMBLES - EJEMPLO DE BAGGING



Estos métodos consisten en entrenar un único modelo con diferentes sub-muestras aleatorias del set de datos de entrenamiento



Árbol de decisión sin  
restricciones  
Decision Tree



Bagging Ensamble de 500  
árboles de decisión  
Decision Trees with Bagging

