TEC | Tecnológico de Costa Rica

Maestría en Ingeniería Electrónica

Reconocimiento de patrones

Apuntes de clase 2

Estudiante: Esteban Martínez Valverde

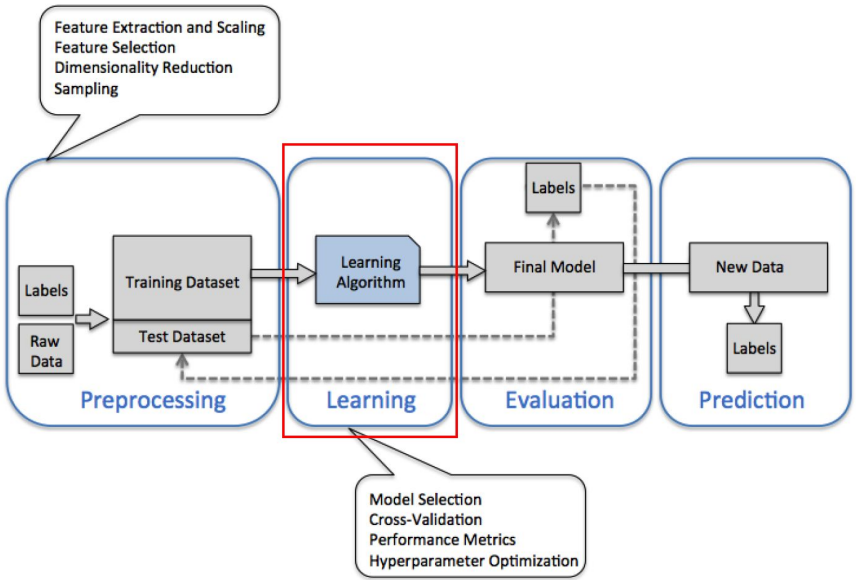
Profesor: Felipe Meza Obando

Julio 2018

Métodos de Aprendizaje Automático

Metodología de Diseño

- **Preprocesado:**
  - **Pr**imero se realiza una extracción, selección de características de escala, Reducción de dimensiones y muestreo
  - Luego se separa el Training DataSet del Test Dataset
- **Entrenamiento:**
  - Selección de modelo
  - Validación Cruzada
  - Métricas de funcionamiento
  - Optimización de hiperparametros
- **Evaluación:**
  - Obtención del modelo final
  - Obtención de las etiquetas
- **Predicción:**
  - Pruebas con nuevos datos
  - Obtención de etiquetas

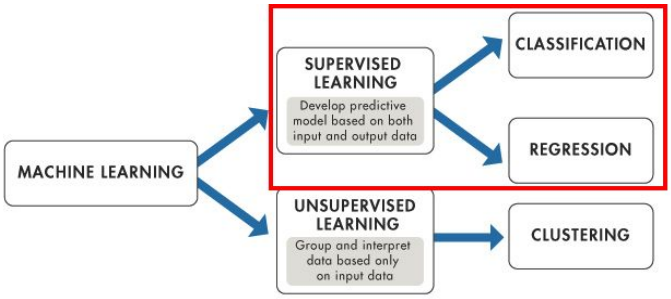


Conceptos importantes

Aprendizaje supervisado

Se predicen salidas **(clases)** a partir de entradas **(atributos)**, a través de un entrenamiento de pares entrada/salida **(instancias)**

atributos -> instancias -> clases



Clasificación

- El objetivo principal es **predecir una clase o varias clases**. Existe:
  - *Clasificación binaria*: yes/no, spam/no-spam
  - *Clasificación multiclase*: eg.iris

Regresión

- El objetivo es **predecir un número continuo**, en un **rango** establecido.
  - Se distingue de la clasificación en el sentido de que hay una continuidad a la salida

Generalización

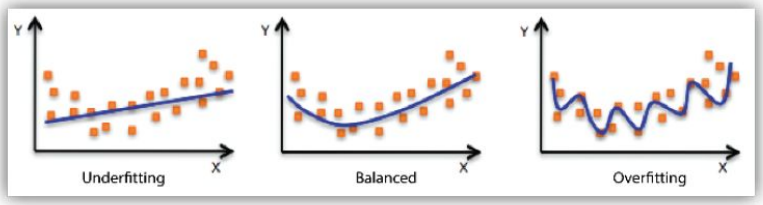
- **Capacidad de hacer predicciones correctas** con el conjunto de pruebas a partir del conjunto de entrenamiento.
  - Si las predicciones son correctas -> el modelo “generaliza" del conjunto de entrenamiento al conjunto de pruebas

Overfitting (training>testing)

- Modelo de aprendizaje que sea **muy complejo**, **funciona muy bien** con el conjunto de entrenamiento pero **no generaliza** con nuevos datos.

Underfitting (training<testing)

- Modelo de aprendizaje que sea **muy simple**, **funciona mal** con el conjunto de entrenamiento y con nuevos datos no será capaz generalizar



Aprendizaje Supervisado

1. kNN (Clasificación/regresión
2. Modelos Lineales:
    - Linear Regression (Clasificación).
    - Logistic Regression y SVM (Regresión).
    - Modelos Lineales para clasificación multiclase.
3. Naive Bayes.
4. Decision Trees.
5. Random Forest.
6. Kernel SVM.

kNN (k Nearest Neighbors)

Parámetros:

➔ Requiere de ajuste del valor k y de la forma de medir la distancia, se asumirá la distancia euclidiana para efectos prácticos.

Ventajas:

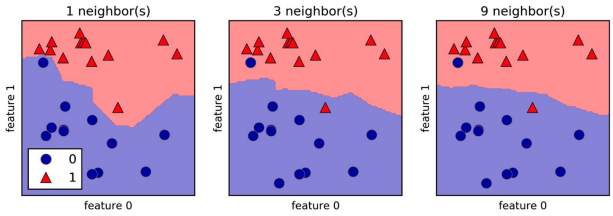
➔ Muy fácil de entender.
➔ Generalmente entrega buenos resultados sin excesivo ajuste.

Desventajas:

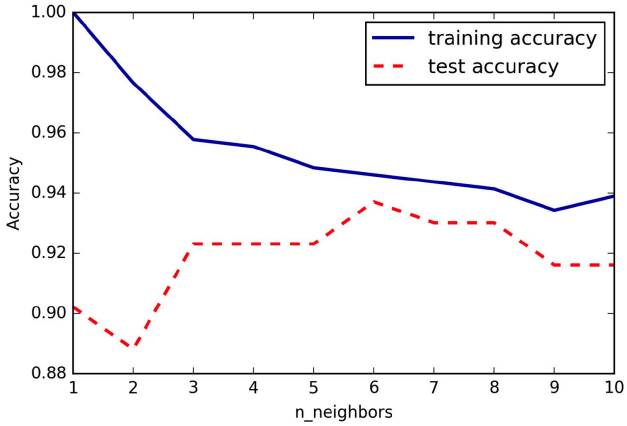
➔ Cuando el conjunto de entrenamiento es muy grande, se puede volver lento.
➔ No da buenos resultados cuando hay muchos *features* o muchos son ceros.

Clasificación

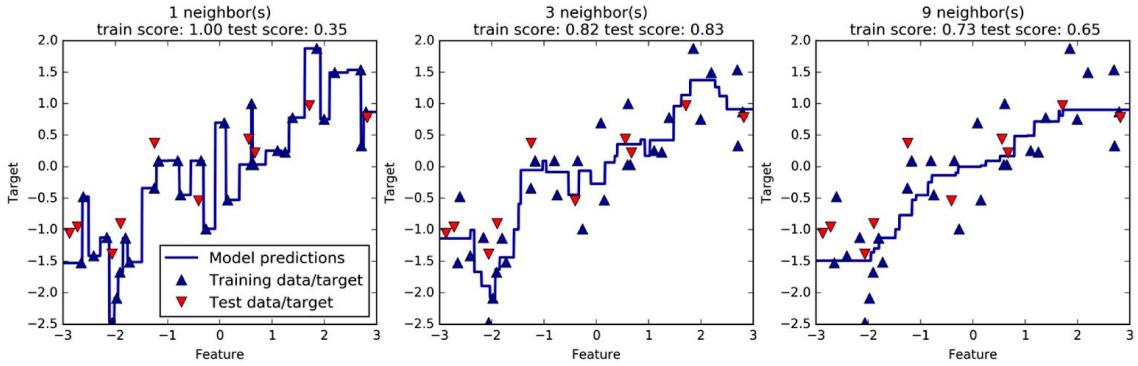
- Para predecir un nuevo dato, **busca el/los punto(s) más cercanos**
- Es el más simple de los algoritmos de aprendizaje.
    - **Caso elemental (k=1):** calculado a partir de 1 vecino cercano
    - **Caso voto (k=3)** cuenta la cantidad de vecinos cercanos pertenecientes a una clase
- Puede ser aplicado a conjuntos de datos de **múltiples clases**.
- Se crea una **línea de decisión** que separa las clases
    - K Alto -> Baja complejidad
    - K bajo-> Alta complejidad



- Se trata de encontrar el punto en el cual: el conjunto de pruebas **entregue la mejor precisión** de la mano de una precisión del conjunto de entrenamiento "buena" (i.e no perfecta pero tampoco pésima).



Regresión



Modelos Lineales - Regresión

Regresión Lineal Ordinaria (OLS)

- Consiste en encontrar los **parámetros w y b** que minimizan el **MSE (mean square error)** entre los valores reales y la predicción.
- En la mayoría de herramientas computacionales se le conoce como ***Linear Regression*.**

```
from sklearn.linear_model import LinearRegression

lr = LinearRegression().fit(X_train, y_train)

lr.score(X_train, y_train)
lr.score(X_test, y_test)
```

scikit-learn demo

RIDGE

- Se basa en ***Linear Regression*****,** con la diferencia de que se persigue obtener valores de **w cercanos a cero**, con el n de minimizar el efecto de los atributos mientras se mantiene un buen desempeño.
- Se recurre al uso del parámetro para **balancear la simplicidad** del modelo con su desempeño.
- El valor óptimo de depende de las características del conjunto de datos.

LASO

Se pueden obtener valores de w iguales a cero.

Se denomina **REGULARIZACIÓN** al efecto de influir en el modelo de aprendizaje con el n de evitar *overfitting*, en el caso de RIDGE se llama regularización L2 y en el caso de LASSO regularización L1.

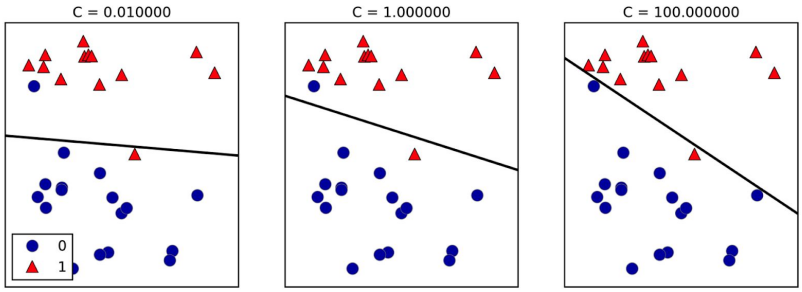
Modelos Lineales - Clasificación

- En el caso de la clasificación se usa la misma base matemática, con la diferencia de que el límite al valor por predecir se ubica en cero.
- Valores menores a cero corresponden a una clase y valores mayores a cero corresponden a la otra clase.
- La salida o clase por predecir corresponde a una línea denominada **límite de decisión**.
- Existen varios tipos de algoritmos de clasificación por modelos lineales pero en general difieren en dos aspectos:
    - Que tan bien la combinación de coeficientes (w) ajuste a los datos de entrenamiento.
    - El tipo de regularización utilizada.

Logistic Regression

- Se utiliza el parámetro C para controlar la regularización.
- Un valor alto de C implica menos regularización, el modelo trata de ajustarse a los datos de training lo mayor posible.
- Un valor bajo de C implica encontrar coeficientes w cercanos a cero.
- Logistic Regression se usa para la clasificación.

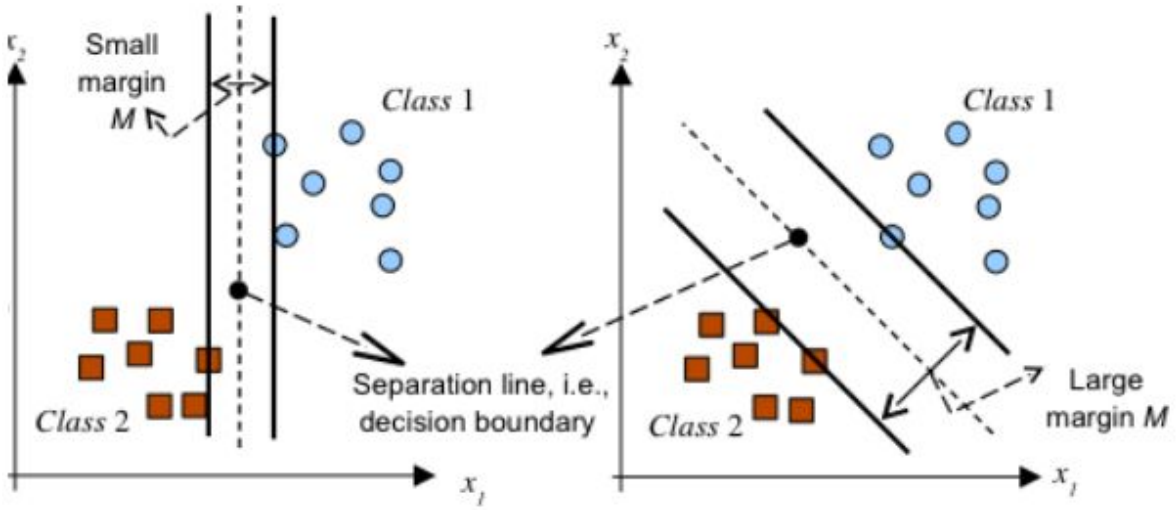
```
from sklearn.linear_model import LogisticRegression
```

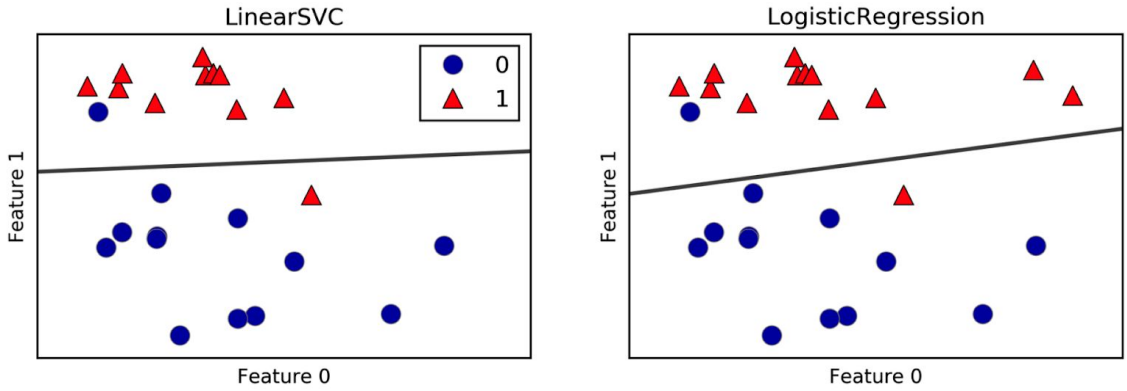


SVM

- Maquina de vector de soporte o Clasificador de vector de soporte.

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





Modelos Lineales - Multiclase

- Consiste en usar modelos lineales usados comúnmente en clasificación binaria a clasificación con múltiples clases
- Se recurre al algoritmo uno vrs. el resto, el cual consiste en separar cada clase del resto como si fuera clasificación binaria.
- Los datos de las zonas centrales (en este caso el triángulo central) se asocian con la clase de la línea más cercana, debiendo as una nueva área por clase.

Resumen

- Los parámetros de regularización corresponden a para los modelos de regresión lineal y a C para la regresión logística y SVM.
- Modelos simples implican un alto y un C bajo.
- Otro factor importante es la regularización, L1 se escoge cuando se tiene claro que solo algunos atributos son importantes.
- Los modelos lineales son rápidos de entrenar y de hacer predicciones.
- Escalan muy bien para conjuntos de datos grandes o dispersos.
- Son de fácil comprensión.
- Un detalle importante es que los coeficientes no son fácilmente interpretables.

Naive Bayes

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

donde A es la hipótesis, $P(A)$ la probabilidad a priori, $P(B|A)$ la probabilidad condicional, $P(B)$ la probabilidad marginal y $P(A|B)$ la probabilidad a posteriori.

- Por ejemplo, se va fabricar una tarjeta electrónica y se requiere de un componente B, las partes se compran a 2 proveedores distintos.
- Un 30% se le compra al proveedor A1 y un 70% al proveedor A2.
- Históricamente el 6% de las partes compradas al proveedor A1 son defectuosas, en el caso del proveedor A2 solo el 4% son defectuosas.

- $P(A_1) = 0.3$
- $P(A_2) = 0.7$
- $P(B|A_1) = 0.06$
- $P(B|A_2) = 0.04$
- $P(B) = 6\%$ de $30 = \mathbf{1.8\%} + 4\%$ de $70 = \mathbf{2.8\%} == \mathbf{4.6\%}$

$$P(A_1|B) = \frac{P(A_1) \times P(B|A_1)}{P(B)} = \frac{0.3 \times 0.06}{0.046} = 0.39$$

$$P(A_2|B) = \frac{P(A_2) \times P(B|A_2)}{P(B)} = \frac{0.7 \times 0.04}{0.046} = 0.61$$

- Se dice ***naive*** por el hecho de que supone una idea simple: independencia de los atributos entre sí.
- • Se puede encontrar en 3 tipos, dependiendo de las características de los datos:
  - ***BernoulliNB***: Datos son binarios.
  - ***MultinomialNB:*** Datos representan un conteo.
  - ***GaussianNB***: Distribución gaussiana de los datos.
- Es eficiente porque aprende a través de la recolección de estadísticas por clase de cada atributo.

- MultinomialNB: Valor promedio de cada atributo para cada clase.
- GaussianNB: Valor promedio de cada atributo y STD para cada clase.

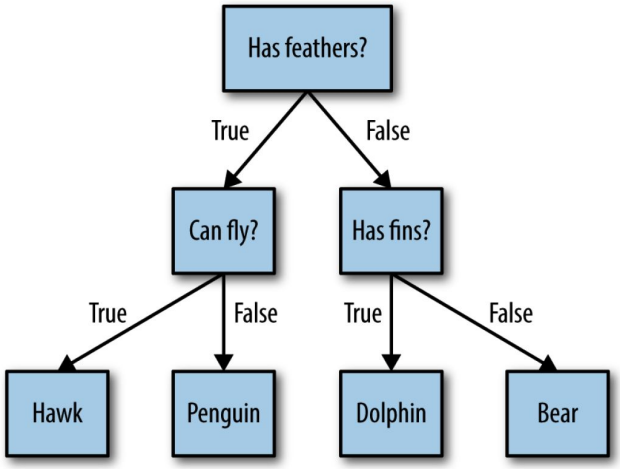
- Algoritmo suele ser muy rápido en el entrenamiento y predicción.
- Se suele usar mucho en grandes conjuntos de datos.
- MultinomialNB y GaussianNB, usan el parametro para suavizado de los datos, resultando en modelos menos complejos.

```
from sklearn.naive_bayes import MultinomialNB
clf = MultinomialNB(alpha=1.0)
clf.fit(X, Y)
```

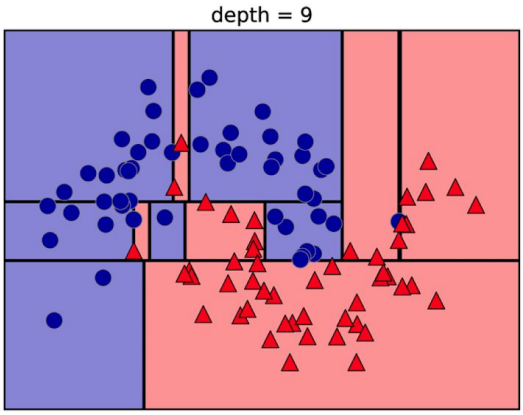
GaussianNB se usa con datos de grandes dimensiones, mientras ***MultinomialNB*** y BernoulliNB con datos dispersos.

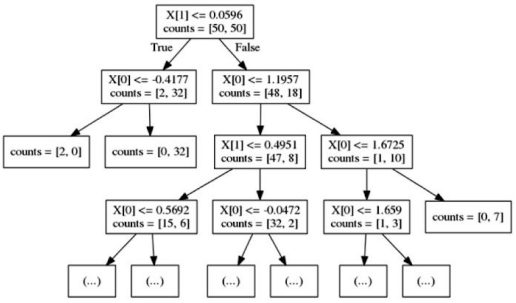
Decision Trees

- Esquema de aprendizaje basado en preguntas tipo if/else, seguidas de una decisión.



- A cada cuadro, sea una pregunta o nodo terminal, se le llama hoja.
- A cada pregunta se le conoce como prueba.
- Al primer nodo se le llama raz y es el que parte los datos de la manera más significativa.





- La hoja cuyos datos pertenecen solamente a una clase se llama pura.

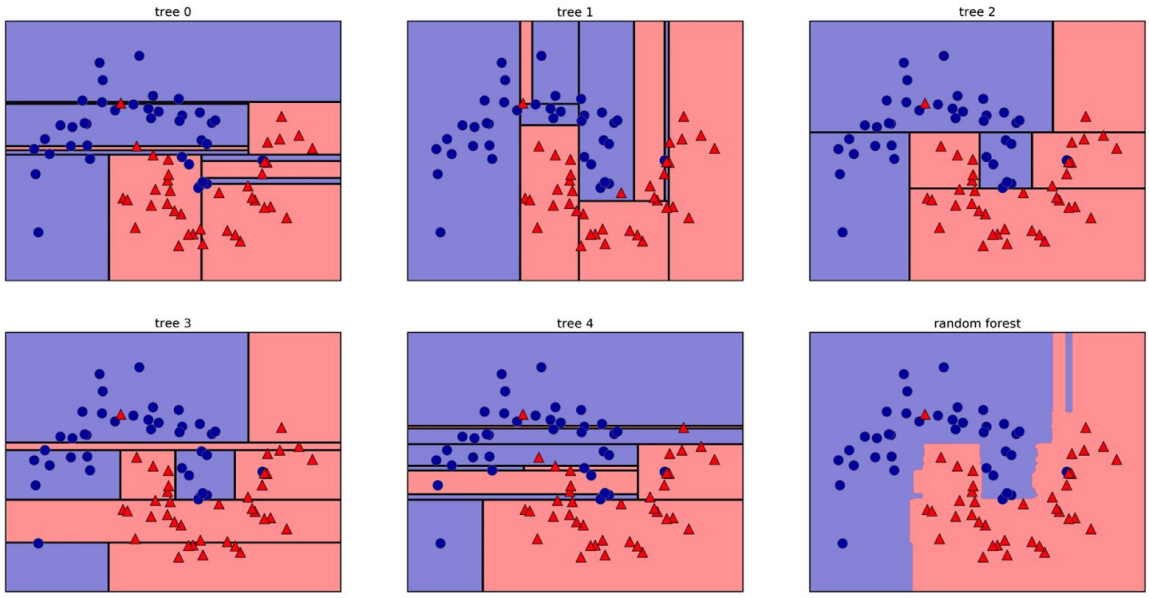
- Un árbol de decisión con solo hojas puras implica que esta 100%
- ajustado al conjunto de entrenamiento, en decir: overfitting. Existen dos técnicas para evitarlo:
    - pre-pruning: Detener la creación del árbol tempranamente (e.g limitar máximo de hojas o la profundidad del árbol).
    - pruning (post-pruning): Remover o colapsar nodos que contienen poca información.
- • Se puede tener una idea de la importancia de los atributos en el árbol, usando el comando:
    - 1 tree.feature_importances_
- O bien, mediante algún método de interpretación gráfica.

- Los árboles de decisión pueden ser usados para clasificación o
- regresión:
    - ***DecisionTreeClassifier***
    - ***DecisionTreeRegressor***
- Es muy recomendado el uso de técnicas para evitar overfitting:
    - ***max depth***
    - ***max leaf nodes***
    - ***min samples leaf***
- Son modelos fáciles de visualizar.
- No requieren pre-procesado por que se desempeña bien con datos en múltiples escalas.

Random Forest

- Algoritmo que pertenece a la familia de métodos en “conjunto" donde se recurre al uso de múltiples modelos de aprendizaje automático operando como uno solo.
- Propuesta para evitar caer en ***overfitting*** usando árboles de decisión convencionales.
- Consiste en una colección de árboles de decisión, donde cada árbol es diferente.
- En conjunto se asume algún overfitting para cada árbol, sin embargo al promediar los resultados se reduce el ***overfitting***.

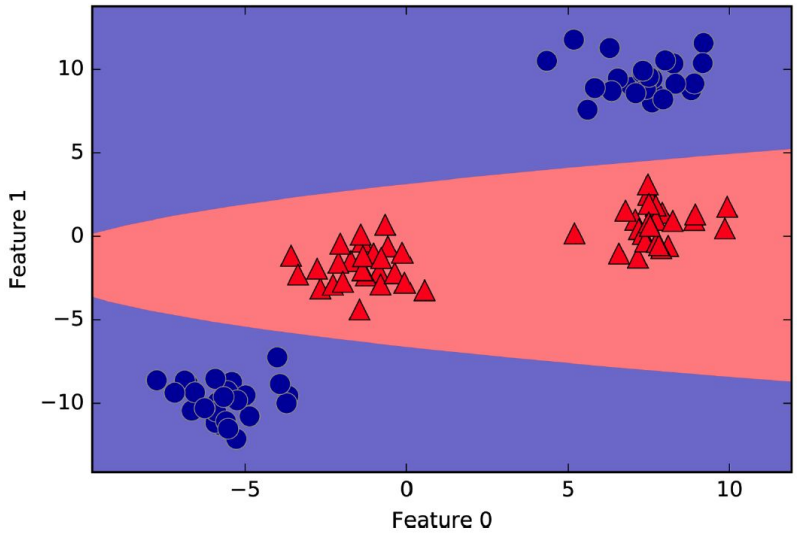
- Su nombre proviene de la forma en la cual se construye cada árbol para mantenerlos diferentes, es una forma aleatoria.
- Para construir el árbol, se debe indicar el numero de arboles
- mediante el parámetro n estimators.
- Es posible usar el método tanto para la clasificación como para la regresión:
    - ***RandomForestRegressor***
    - ***RandomForestClassifier***
- El parámetro max features dene que tan diferentes son los árboles que se crearan. Un valor alto implica similitud en los árboles, uno bajo signica amplia diferencia.
- Para la predicción usando Random Forest, primero se lleva a cabo la predicción para cada árbol y luego si es clasificación se hace un conteo del resultado de cada árbol (gana la mayora) y si es regresión se estima el promedio.



Kernel SVM

- Extensión del modelo de clasificador para conjuntos de datos más complejos con más dimensiones.
- Los modelos lineales pueden enfrentar limitaciones en su flexibilidad, dadas las limitaciones de las líneas o hiperplanos.

- Una forma de mejorar la flexibilidad es agregando más atributos,
- bajo criterios técnicos.
- Por ejemplo: agregar en set de datos anterior un tercer atributo llamado (feature1 ** 2).
- Un plano en 3D es ahora capaz de separar adecuadamente las clases.



- Si llevamos el límite de decisión como una función de los 2 atributos originales, vemos como la linealidad se pierde y más bien se tiene una especie de elipse.
- Agregar atributos no lineales puede hacer que los modelos lineales
- sean más efectivos.
- Hay que tener cuidado con agregar más atributos, en especial en términos del costo computacional.
- La clave: El uso de un **kernel**.
- Encontramos dos tipos de kernels muy comunes:
    - ***Polinomial***: Se hace un cálculo de todos los posibles polinomios hasta un cierto grado de los atributos originales.
    - ***RBF:*** También conocido como kernel gaussiano, considera todos los posible polinomios de todos los grados pero a mayor grado del polinomio menor relevancia tienen el atributo.
- Se dispone de varios parámetros de ajuste, donde los más comunes
- son:
    - ***gamma***: Controla el ancho del kernel gaussiano, determina la escala que determina la cercana entre dos puntos.
    - ***C***: Es el parámetro de regularización, limita el nivel de importancia de cada punto.

```
from sklearn.svm import SVC
X, y = % dataset %
svm = SVC(kernel='rbf', C=10, gamma=0.1).fit(X, y)
```

- Un gamma bajo implica radio alto del kernel gaussiano, lo que implica que los puntos se consideran cercanos y se re eja como un límite de decisión menos complejo y más suave.
- Un gamma alto implica radio pequeño del kernel gaussiano, lo que implica que los puntos se consideran lejanos y se re eja como un límite de decisión más complejo.
- Un C bajo implica que menos puntos tienen influencia en el límite de decisión.
- Un C alto implica que más puntos tienen influencia en el límite de decisión.

- Se desempeñan muy bien en la mayoría de conjuntos de datos.
- Trabajan bien en conjuntos de pocas o muchas dimensiones.
- Con muchas instancias o ejemplos de datos puede consumir muchos recursos computacionales.
  El preprocesamiento de los datos es muy común en kSVM, factor por el que random forest es muy usado.
- Son más complejos de explicar y de inspeccionar con detalle durante la operación.

Métodos de Aprendizaje NO Supervisado

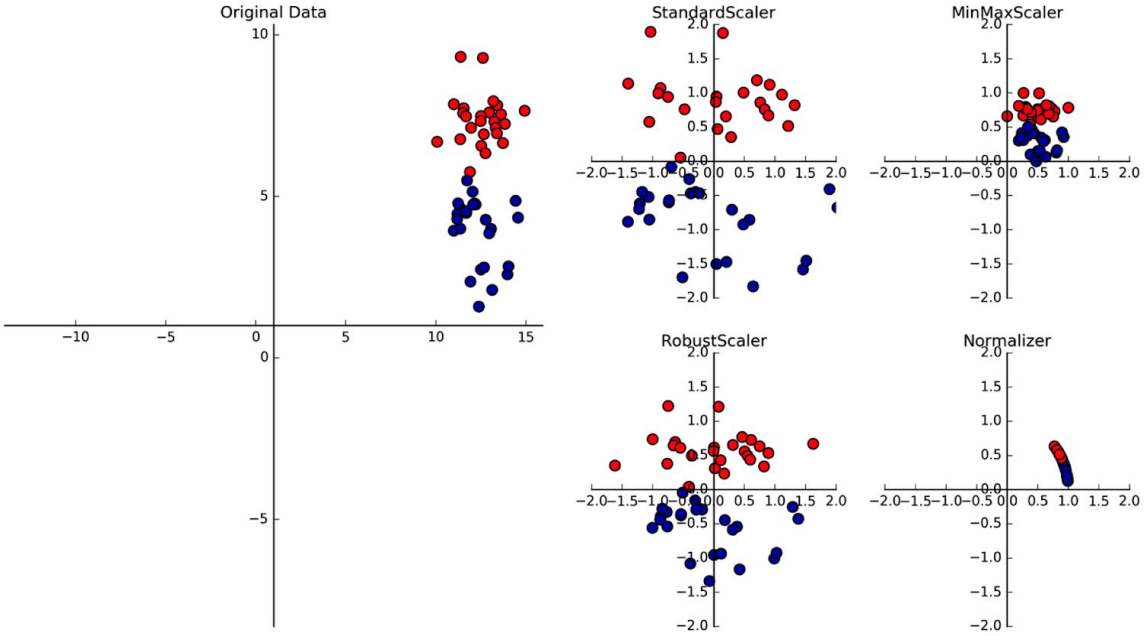
- Se caracteriza por que no hay un conocimiento previo de la salidas (clases).
- Al algoritmo solo se le presentan los datos de entrada y de ah se extrae conocimiento como salida.
- Encontramos dos tipos de aprendizaje:
  - ***Transformaciones NO-supervisadas:*** Creación de nuevas representaciones de datos que hacen más fácil su representación al ojo humano e.g reducir dimensiones.
  - ***Algoritmos de agrupamiento (clustering):*** Partición de los datos en grupos distintos compuestos por datos \similares" e.g agrupar rostros.

Desafíos

- Cómo evaluar que el algoritmo aprendió algo que sea de valor?
- No hay forma de indicarle al algoritmo que es lo que estamos buscando, generalmente la forma de validarlo es manualmente e.g visualizando los rostros.
- Este tipo de aprendizaje se usa mucho a nivel exploratorio para entender mejor los datos.
- Otro uso muy común es el pre-procesado de algoritmos supervisados:
  - ANN y SVM son muy sensibles a la escala de los datos.

Tipos de Preprocesado

- ***StandardScaler:*** Para cada atributo; la media es 0 y la varianza es 1. Los atributos quedan en rangos de la misma magnitud.
- ***RobustScaler***: Usa la media y cuartiles, en vez de media y varianza. Ignora puntos que son outliers.
- ***MaxMinScaler:*** Se ajustan los datos entre 0 y 1.
- ***Normalizer:*** Se proyectan los datos en un círculo (o esfera para más dimensiones) de radio 1.



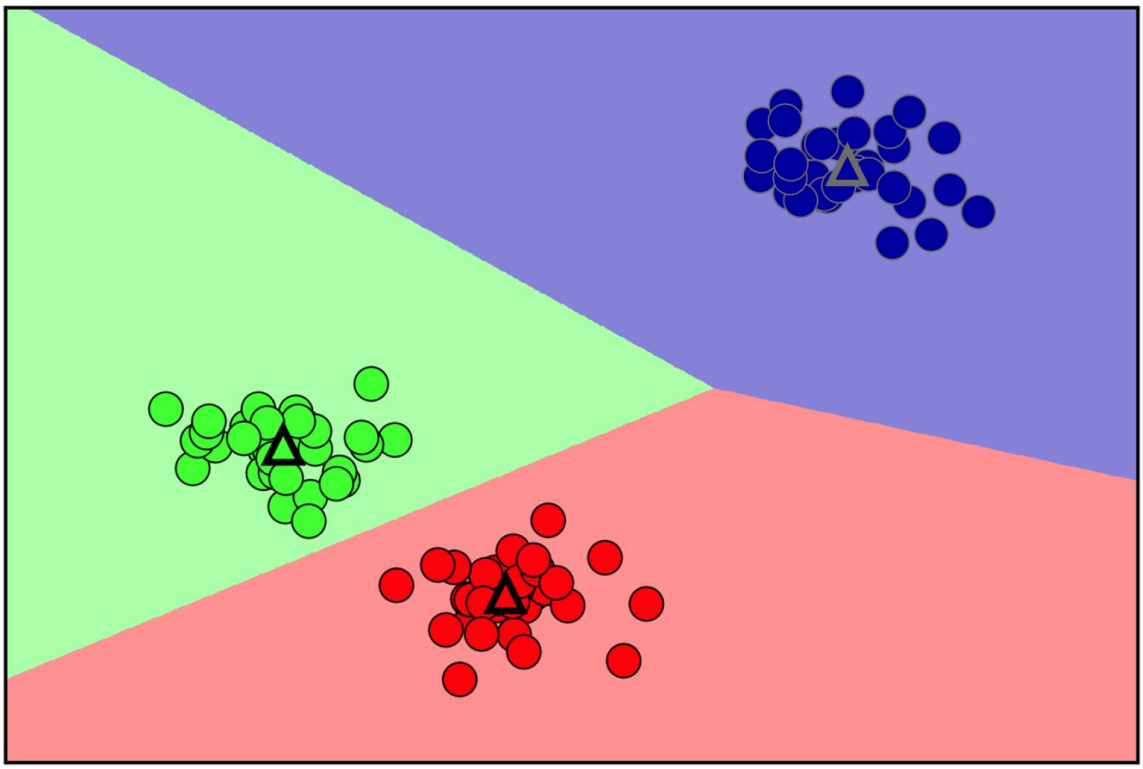
Algoritmos de agrupamiento (clustering)

- Partición del conjunto de datos en grupos (clusters), los datos de
- cada cluster son “similares".
- La tarea de predecir consiste en asignar el nuevo punto a un grupo.
- Se mencionan 3 tipos:
    - k-Means clustering.
    - Aglomeración.
    - DBSCAN.

K-means clustering

- Consiste en buscar “centros de grupo" que sean representativos.
- Incluye los siguientes pasos:
    - Se eligen centros aleatorios.
    - Se asignan los puntos al centro más cercano.

    - Se ajusta el centro de cada grupo a la media de los puntos que pertenecen a dicho grupo.
    - El algoritmo se detiene cuando no hay más movimientos en el centro de los grupos.
- Se debe especificar el número de grupos.

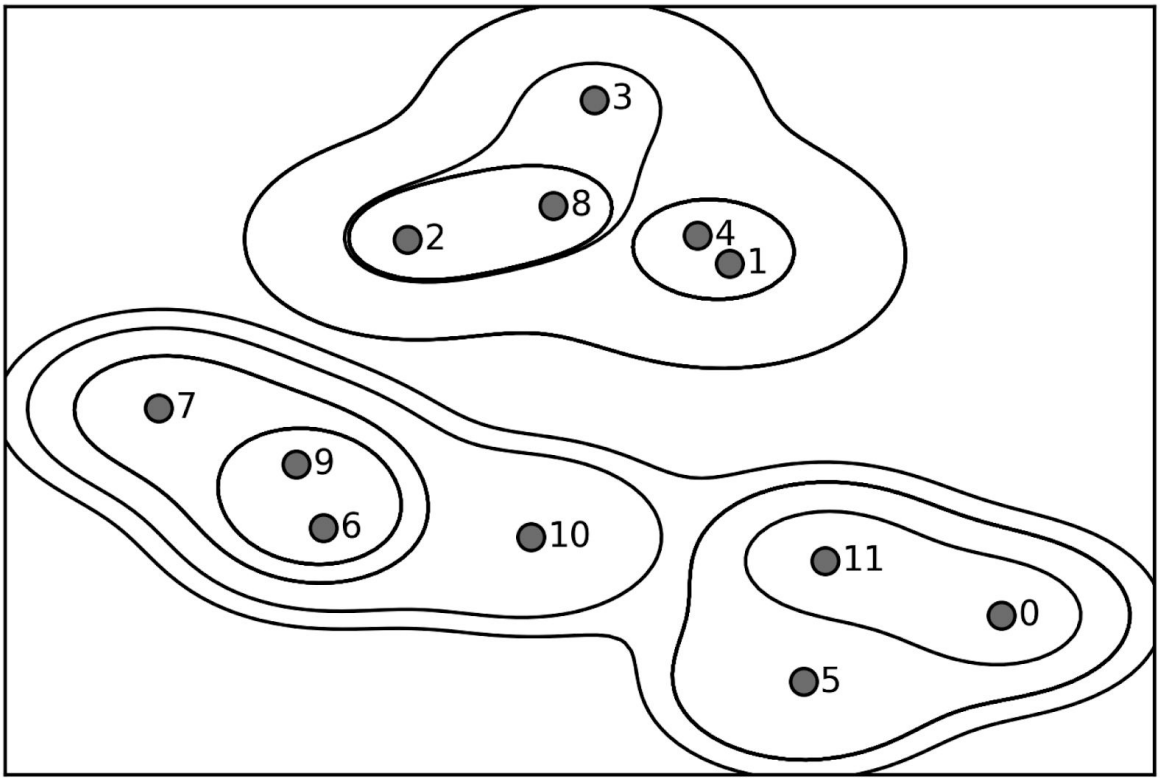


- Algunos detalles del algoritmo a considerar:
    - Solo puede venir regiones con formas relativamente simples.
    - Asume que todos los grupos tienen el mismo diámetro.
    - El límite entre grupos se ubica generalmente justo en el medio de dos grupos.
    - Asume que todas las direcciones son importantes para cada grupo.

Aglomeración

- Se fundamenta en el principio de asumir que cada punto es un grupo, posteriormente los más próximos se unen para formar un nuevo grupo hasta que algún criterio de "**stop**" se cumpla (número de grupos).
- Existe un parámetro para especificar qué tan parecidos son los grupos:
- ***ward***: Se escoge como grupo los puntos que incrementen lo mínimo la varianza del grupo. Se obtienen grupos balanceados.

    - muy bueno para la mayoría de conjuntos de datos, average y complete buenos para conjuntos no balanceados.
- ***average***: Se escogen como grupo los puntos que tengan la mínima distancia promedio.
- ***complete***: Se escogen como grupo los puntos que tengan la mínima distancia máxima.

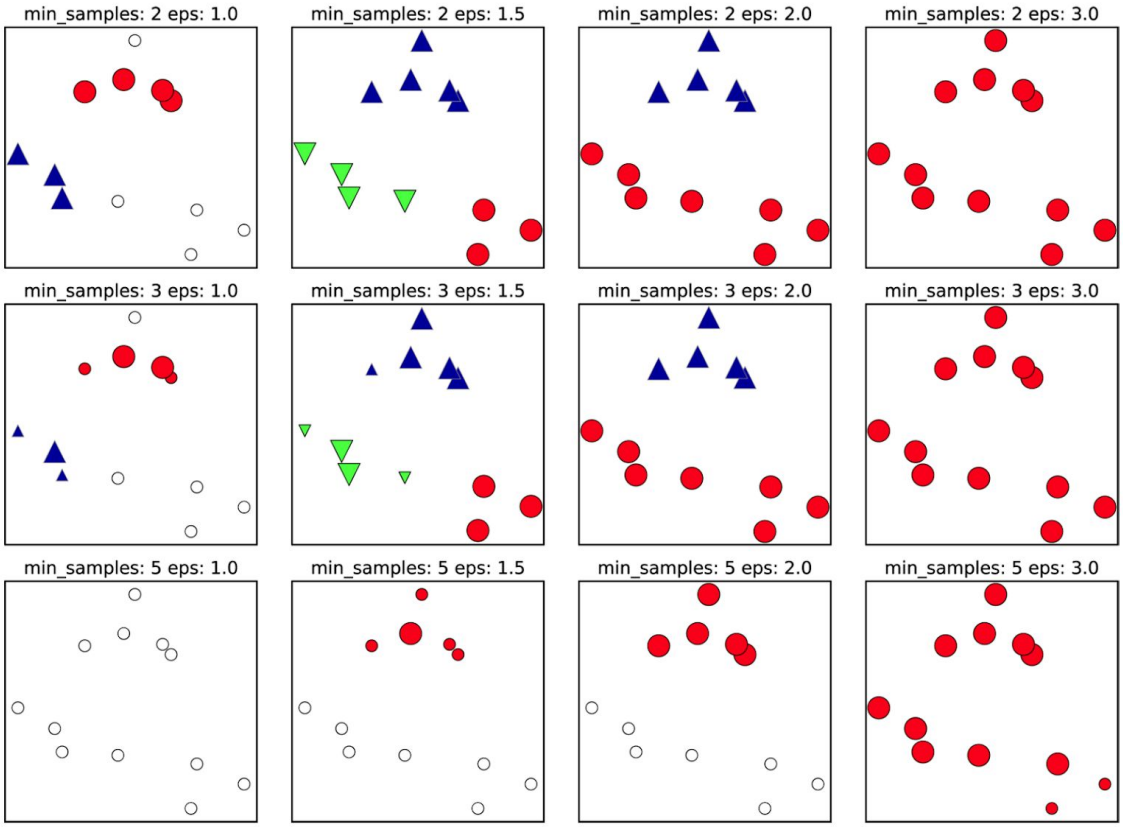


- Es posible tener una idea del número de grupos óptimo usando agrupamiento jerárquico y los dendrogramas.
- Desafortunadamente el agrupamiento jerárquico sólo se puede usar con dos features (2D).
- No existen en scikit-learn pero si en SciPy.
- Se crea un árbol a partir de los puntos.
- Donde se obtenga la máxima distancia, se marca la zona de grupos recomendados.
- Aun así, este método no es muy bueno para formas complejas.

DBSCAN

- Density based spatial clustering of applications with noise.
- No requiere conocimiento de número de grupos a priori.
- Trabaja bien para conjunto de datos complejos.
- Pese a ser más lento que k-Means o por aglomeración, escala muy bien para grandes conjuntos de datos.

- Opera de la siguiente manera:
    - Idéntica zonas densas, separadas de zonas vacas.
    - Los puntos que pertenecen a un grupo, se llaman \core samples".
    - Los puntos que NO pertenecen a un grupo, se llaman \noise".
    - Se definen dos parámetros min samples y eps.
    - Si hay menos puntos que min samples en la distancia [eps] para un punto X) X = noise
    - Si hay más puntos que min samples en la distancia [eps] para un punto X) X = core sample
- • El grupo crece hasta que no hayan mas \core samples" dentro de la distancia eps en el grupo.



Incrementando eps significa que más puntos serán incluidos en el grupo.

Incrementando min samples significa que menos puntos serán \core samples" y más serán \noise".

\Métodos