CONCEPTOS BÁSICOS DE MACHINE LEARNING

Martín De la Fuente

OBJETIVOS

- Hacer un resumen más teórico de los contenidos.
- Lograr entender bien qué estamos haciendo en la tarea.
- Manejar conceptos para poder responder las preguntas de la última parte de la tarea.
- Avanzar en su tarea y resolver dudas.

Clase 10: Preprocesamiento y PCA

Ayudantía 5: RD y Clustering

Documentación scikit-learn

CLASIFICADORES

CLASIFICADORES

¿Cuándo queremos usar clasificadores?

- Usamos clasificadores cuando necesitamos detectar mediante un sistema automático la categoría (o clase) de un input.
- Por ejemplo:
  - Clasificar imágenes según su contenido
  - Clasificar canciones según su género musical
  - Clasificar dígitos escritos a mano
  - Clasificar vinos según su calidad

CLASIFICADORES

¿Qué clasificadores existen?

- Redes Neuronales (ANN)
- Vectores de Soporte (SVM)
- Regresión Logística (LR)
- Árboles de Decisión (DT)
- Vecinos Más Cercanos (KNN)
- Bayesiano Ingenuo (NB)

CLASIFICADORES

¿Cómo construimos un clasificador?

- Debemos buscar la forma de "entrenar" a nuestro clasificador.
- Usamos datos "etiquetados" para que el clasificador aprenda a reconocer patrones (aprendizaje supervisado).

CLASIFICADORES

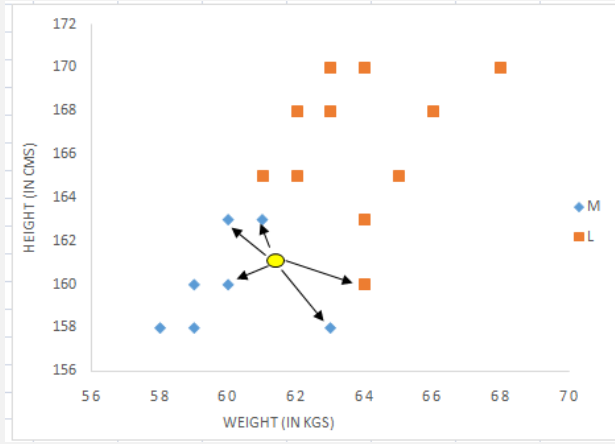
MESA

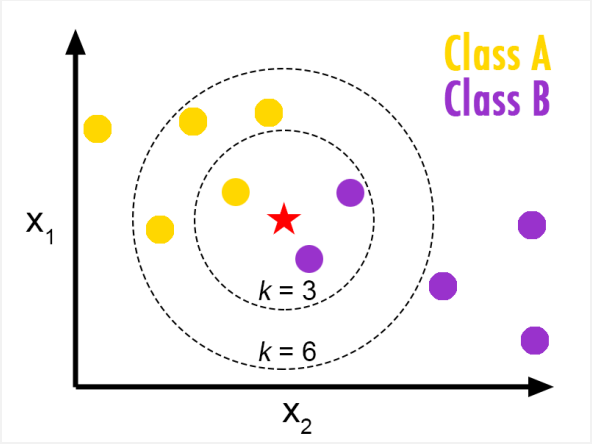
SILLA

= ?

K-NEAREST NEIGHBOURS

- Se almacenan los datos etiquetados en un espacio de N dimensiones.
- Para clasificar una instancia nueva, vemos a qué clase corresponden los K vecinos más cercanos.
- Generalmente se usa distancia euclidiana.
- Es importante normalizar los datos.
- Parámetros importantes: K





DECISION TREES

- A partir de los datos etiquetados se construye un árbol.
- Para clasificar una instancia nueva vamos avanzando por el árbol (según los atributos de esa instancia) hasta llegar a un nodo hoja que nos dice su clase.
- Se puede construir el árbol bajo distintas reglas y criterios.
- Los atributos con valores numéricos continuos deben tratarse por intervalos.
- Es importante entrenar con datos variados.
- Parámetros importantes: profundidad máxima del árbol y muestras mínimas para ramificar.

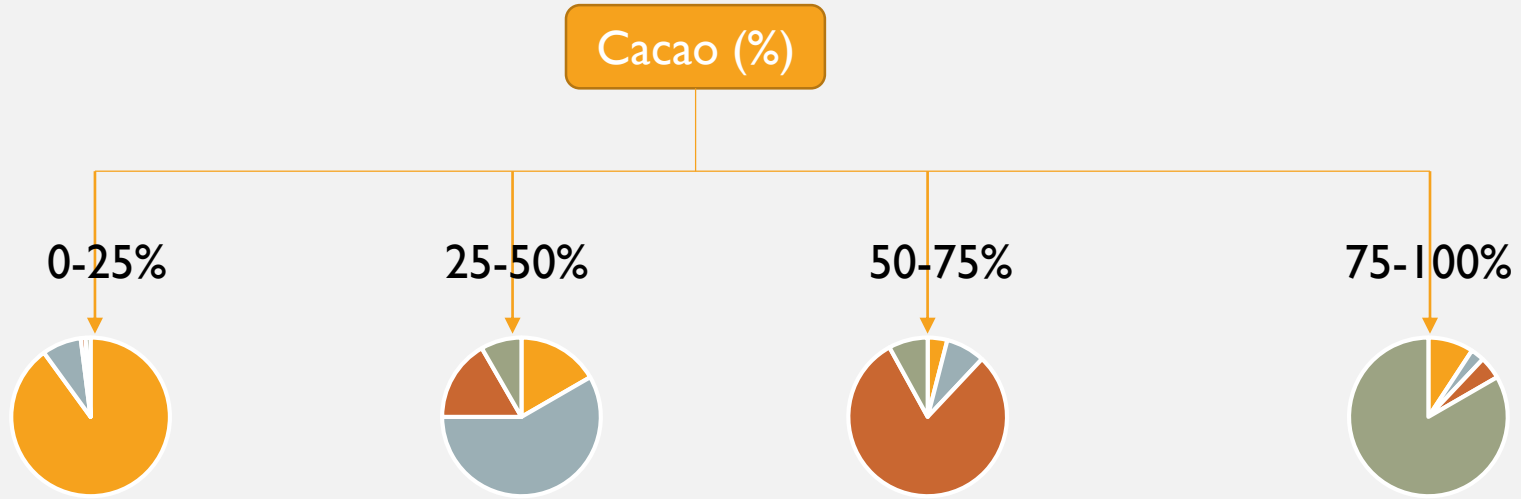
Cacao (%)	Leche (%)	Color	Precio (por 100gr)	Calidad
30	50	Café neutro	4312	Muy bueno
50	20	Café oscuro	4602	Regular
80	5	Café oscuro	8160	Muy bueno
20	40	Café neutro	2569	Malo
10	60	Café claro	1420	Malo
10	70	Café claro	1032	Bueno
30	20	Café neutro	4926	Bueno
60	10	Café oscuro	8741	Regular
55	5	Café oscuro	8423	Muy bueno
62	5	Café oscuro	9851	Muy bueno
20	40	Café neutro	4563	Malo
5	75	Café claro	5102	Regular
20	20	Café neutro	2036	Malo
15	30	Café claro	2471	Malo
20	30	Café neutro	3625	Regular
10	60	Café claro	1359	Malo
90	2	Café oscuro	10465	Muy bueno
30	20	Café neutro	2512	Regular

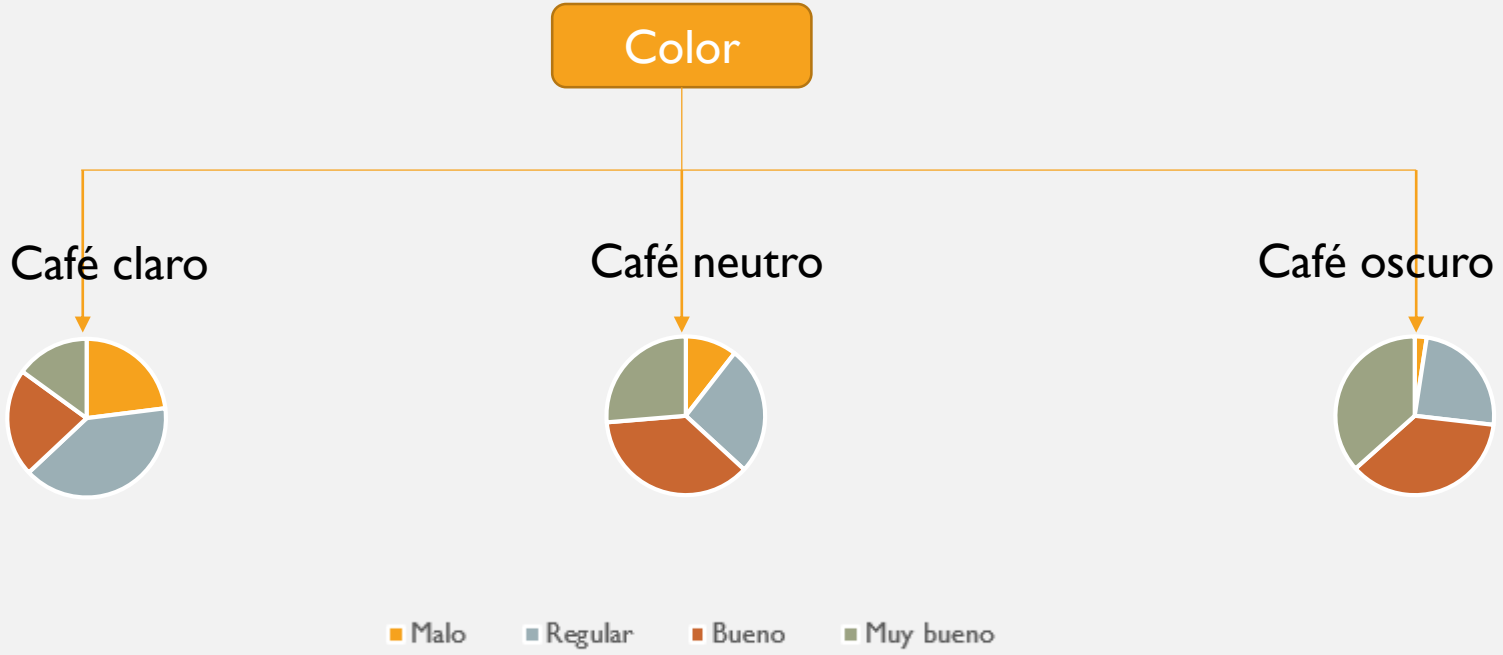
¿Qué atributo tiene mayor relación con la calidad?

¿Hay forma de medir/cuantificar esta relación?

¿En qué parte del árbol situamos a los que mejor separan las clases?

¿Hasta que punto seguimos ramificando el árbol?



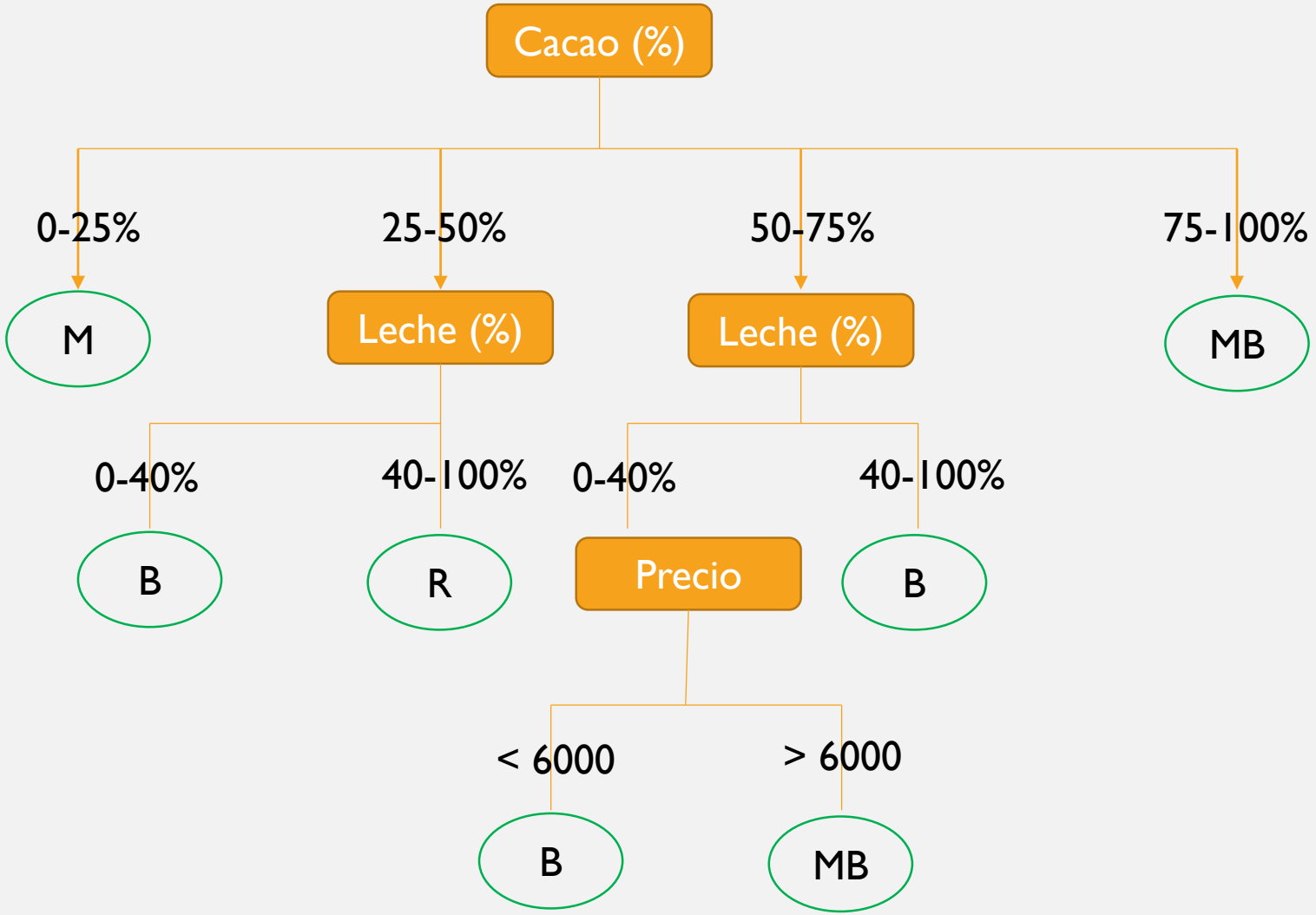


Cacao (%)	Leche (%)	Color	Precio (por 100gr)	Calidad
30	50	Café neutro	4312	Muy bueno
50	20	Café oscuro	4602	Regular
80	5	Café oscuro	8160	Muy bueno
20	40	Café neutro	2569	Malo
10	60	Café claro	1420	Malo
10	70	Café claro	1032	Bueno
30	20	Café neutro	4926	Bueno
60	10	Café oscuro	8741	Regular
55	5	Café oscuro	8423	Muy bueno
62	5	Café oscuro	9851	Muy bueno
20	40	Café neutro	4563	Malo
5	75	Café claro	5102	Regular
20	20	Café neutro	2036	Malo
15	30	Café claro	2471	Malo
20	30	Café neutro	3625	Regular
10	60	Café claro	1359	Malo
90	2	Café oscuro	10465	Muy bueno
30	20	Café neutro	2512	Regular

Para cuantificar qué atributos separan mejor las clases podemos usar la entropía o la impureza de Gini

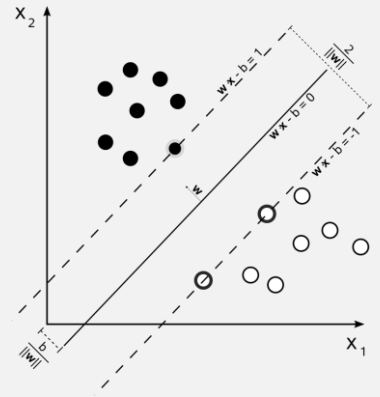
Situamos más cerca del nodo raíz a aquellos atributos que mejor separan las clases.

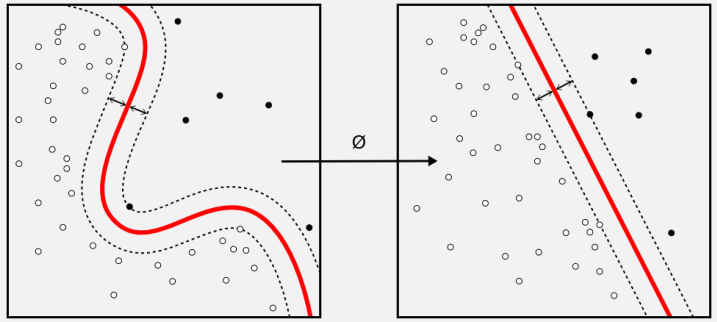
Mientras más profundo estamos en el árbol, menos ejemplos estaremos considerando. No es buena idea tomar la decisión cuando tenemos pocos ejemplos.



SUPPORT VECTOR MACHINES

- Se encuentra un hiperplano que separa las distintas clases lo mejor posible.
- Para clasificar una instancia nueva, vemos en qué lado del hiperplano se encuentra.
- Se usan funciones de kernel para aumentar la dimensionalidad y así lograr una separación lineal.





CLASIFICADORES

¿Cómo medimos su rendimiento?

- Vimos que existen distintos clasificadores, y que para entrenarlos debemos entregarles datos etiquetados.
- Al finalizar el entrenamiento nos gustaría darles un dato nuevo (que no tiene etiqueta) y ver si son capaces de clasificarlo correctamente.
- Por esta razón el conjunto de datos etiquetados los dividimos en dos: el set de entrenamiento y el set de prueba.

SCORE O ACCURACY

Nos dice qué porcentaje de los datos que probamos fueron clasificados de manera correcta.

MATRIZ DE CONFUSIÓN

Versión detallada del score, por clase.

		Predicted			
		Iris-setosa	**Iris-versicolor**	**Iris-virginica**	**Σ**
Actual	**Iris-setosa**	100.0 %	0.0 %	0.0 %	**50**
	Iris-versicolor	0.0 %	88.7 %	6.4 %	**50**
	Iris-virginica	0.0 %	11.3 %	93.6 %	**50**
	Σ	**50**	**53**	**47**	**150**

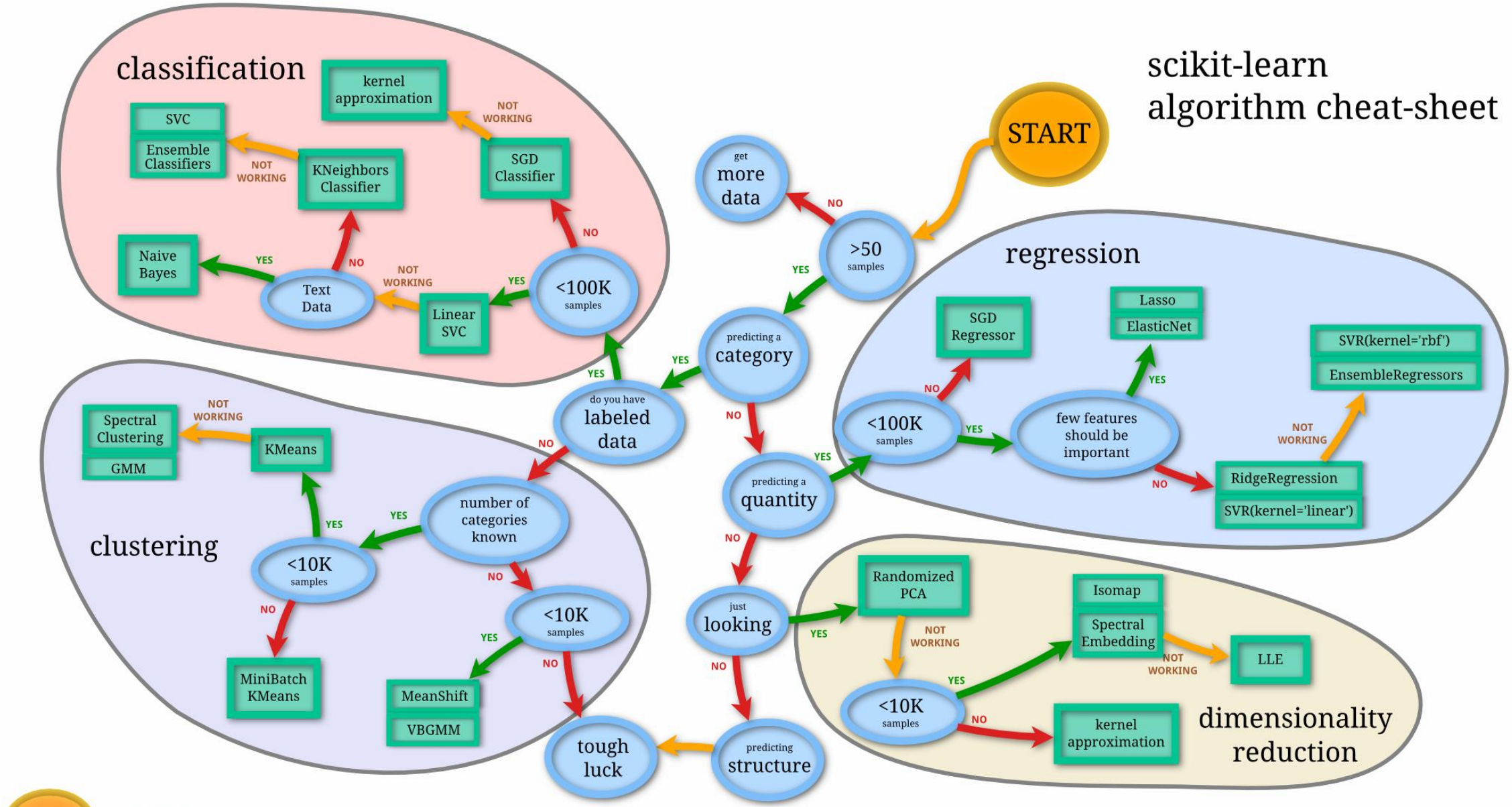
- Es importante que el set de pruebas tenga varios ejemplos de cada clase, y a su vez, la cantidad de ejemplos sea homogénea en todas las clases.
  - Se puede representar como valor numérico o como porcentaje.
- La MC permite ver qué clases son identificadas mejor y peor.
- La MC permite ver qué clases suelen confundirse más.

CLASIFICADORES

¿Qué otras métricas existen?

- Funciones de pérdida (cuánto varía el valor predicho del valor real):
  - Logistic loss o crossentropy loss
- Métricas específicas para clasificación binaria (dos clases).
  - Precision
  - Recall (Sensitivity)
  - F1 Score
  - F-Beta Score
  - AUC-ROC

OTRAS ÁREAS DE MACHINE LEARNING


scikit-learn
algorithm cheat-sheet
START
classification
kernel approximation
NOT WORKING
SVC
Ensemble Classifiers
NOT WORKING
KNeighbors Classifier
SGD Classifier
NO
Naive Bayes
YES
NO
Text Data
NOT WORKING
Linear SVC
YES
<100K samples
NO
get more data
NO
>50 samples
YES
regression
SGD Regressor
Lasso
ElasticNet
SVR(kernel='rbf')
EnsembleRegressors
NO
<100K samples
YES
few features should be important
YES
NOT WORKING
NO
RidgeRegression
SVR(kernel='linear')
predicting a category
YES
NO
do you have labeled data
YES
NO
predicting a quantity
YES
NO
clustering
Spectral Clustering
GMM
NOT WORKING
KMeans
YES
number of categories known
YES
NO
<10K samples
NO
MiniBatch KMeans
<10K samples
YES
NO
MeanShift
VBGMM
just looking
YES
NO
Randomized PCA
NOT WORKING
Isomap
Spectral Embedding
NOT WORKING
LLE
YES
<10K samples
NO
kernel approximation
dimensionality reduction
tough luck
predicting structure
Back
scikit learn

REDUCCIÓN DE DIMENSIONALIDAD

REDUCCIÓN DE DIMENSIONALIDAD

¿Para qué sirve?

- Usamos técnicas de reducción de dimensionalidad cuando queremos visualizar los datos.
- Cuando tenemos datos en muchas dimensiones no es posible ver su distribución en el espacio, por lo tanto hacemos una reducción de dimensiones.
- Generalmente hacemos la reducción a 2 o 3 dimensiones.
- Al hacer la reducción hay una pérdida de información asociada.
- Generalmente buscamos que esa pérdida se distribuya de forma equitativa entre todos los atributos.

REDUCCIÓN DE DIMENSIONALIDAD

¿Qué técnicas existen?

- t-SNE
  - Busca preservar distancias locales.
- PCA
  - Busca capturar las estructuras en los atributos.
  - Captura bien las estructuras lineales, pero no sirve mucho para estructuras no lineales.

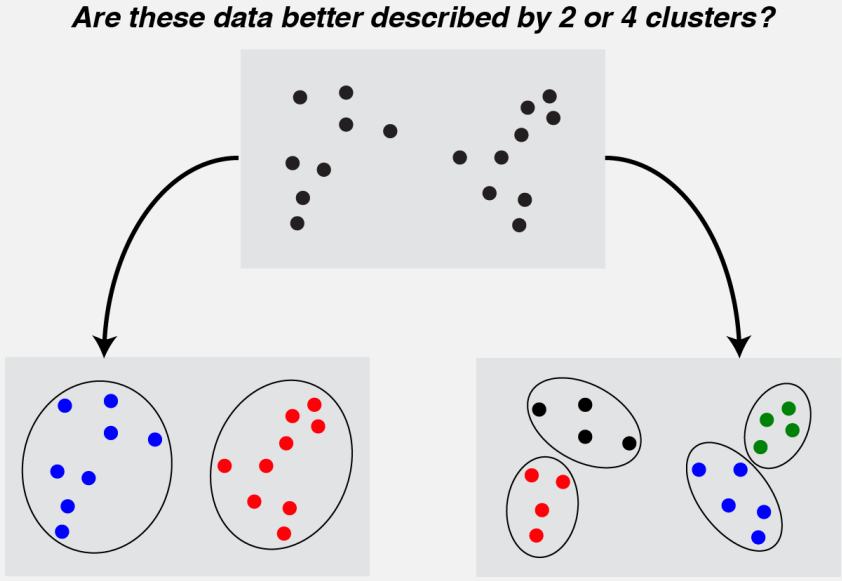
CLUSTERING

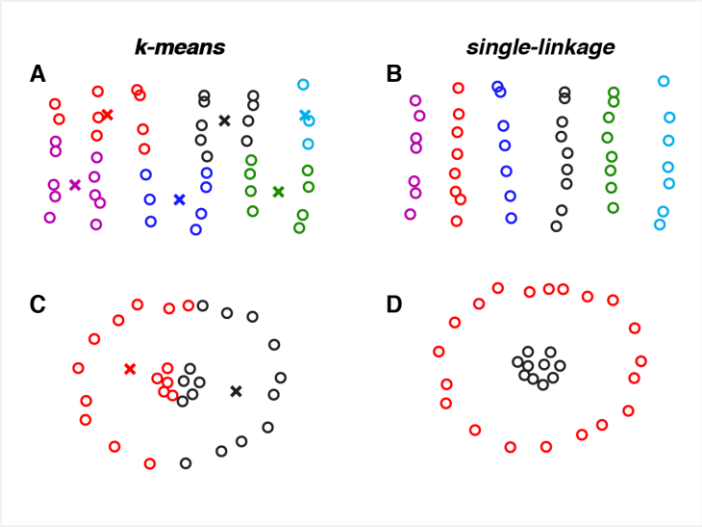
CLUSTERING

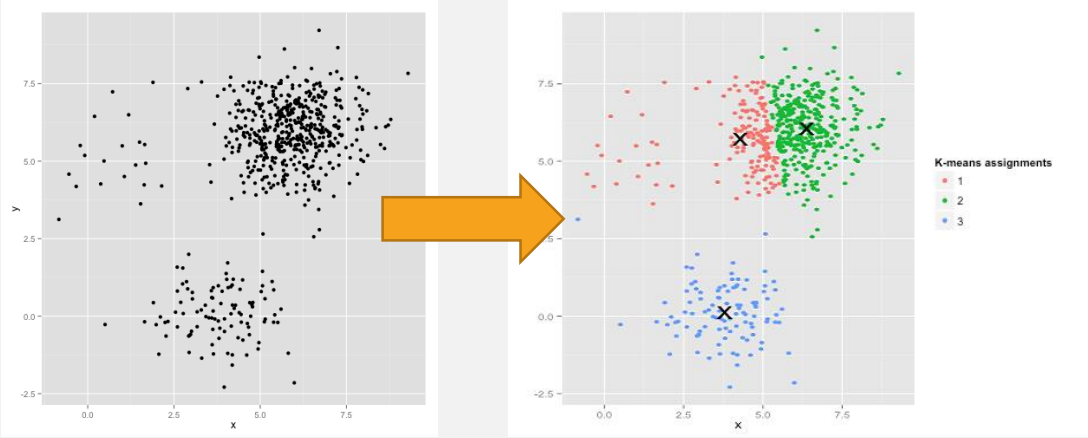
¿Para qué sirve?

- Usamos técnicas de clustering cuando tenemos datos no etiquetados y queremos encontrar las clases o categorías existentes.
- Si lo vemos desde una perspectiva visual puede ser un problema fácil de resolver para un humano, pero suele ser un problema difícil.
- Dependiendo del caso particular pueden haber varias soluciones posibles, es un tema de perspectiva.
- Aunque generalmente visualizamos las técnicas de clustering en 2D, estos pueden ser aplicados en una cantidad arbitraria de dimensiones.

- Los computadores no “ven” como los humanos, solo tienen una “lista de puntos”.
- Una solución podría encontrar más clases que otras. En general en estos problemas no se sabe el número de clases por lo tanto ambas opciones podrían ser correctas.
- Hay algoritmos que funcionan bien en algunos casos y otros que funcionan mejor en otros. Algo que puede parecer obvio para los humanos puede no serlo para un algoritmo.







CLUSTERING

¿Qué algoritmos existen?

- Basados en conectividad
  - Single-linkage, complete-linkage, UPGMA
- Basados en centroide
  - K-Means y variaciones
- Basados en distribución
  - Gaussian mixture model
- Basados en densidad
  - Meanshift, DBSCAN y variaciones

CLUSTERING

¿Cómo medimos su rendimiento?

- El desempeño de un algoritmo de clustering puede ser bueno o malo dependiendo de lo que se busca, por lo tanto es algo bastante subjetivo.
- Existen métricas que intentan medir con un puntaje el rendimiento del algoritmo. Se conocen como métricas de evaluación interna:
  - Índice de Davies-Bouldin, Índice de Dunn, Coeficiente de Silhouette
- Generalmente, estas métricas buscan ven qué tanto se parecen los puntos de un mismo cluster y qué tanto difieren puntos de clusters distintos. Esto hace que evalúen mejor cierto tipo de algoritmos.

CLUSTERING

¿Cómo medimos su rendimiento?

- También existen métricas de evaluación externa donde una se entrega una solución al problema de clustering y luego se compara qué tan parecida es la respuesta entregada por el algoritmo.

CONCEPTOS BÁSICOS DE MACHINE LEARNING

Martín De la Fuente