30.01 Semana 3

> Machine Learning

Introducción al machine learning

> J. Antonio García Ramírez

jose.ramirez@cimat.mx

Agenda: Aprendizaje Supervisado

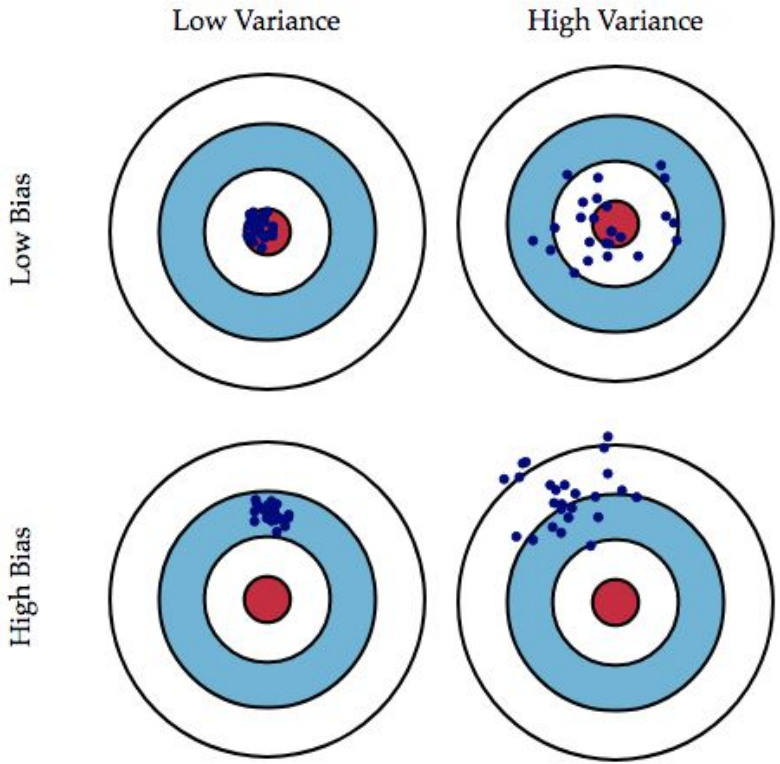
- Review, PCA y lab (40)
- Aprendizaje supervisado, def. (5)
- Regresión
  - Lineal y multiple (15) sys.sleep()
- Localidad, knn (15)
- Árboles (20)
  - Más árboles, un bosque

- Bonus: LDA, DA y FLD

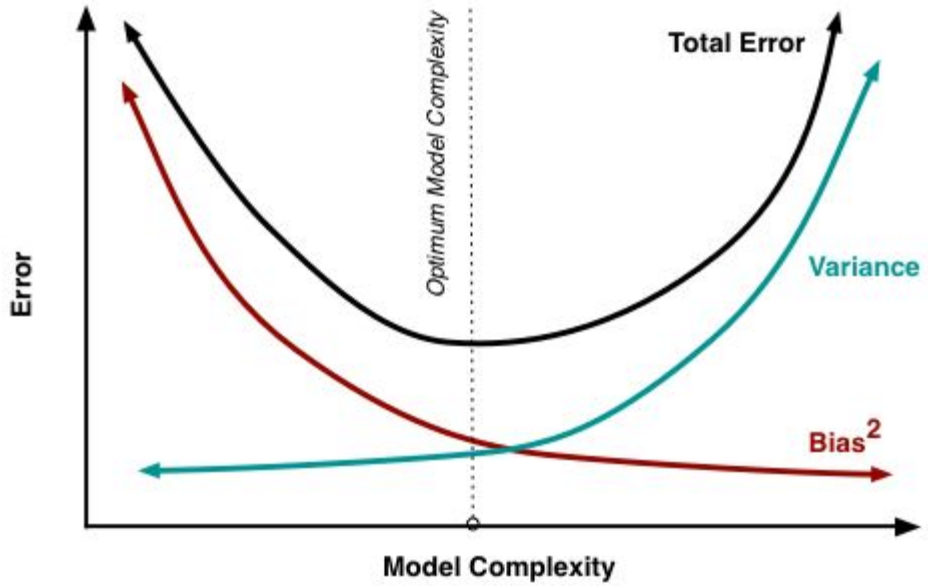
Agenda: Aprendizaje supervisado

- Clasificación (10) sys.sleep()
  - Regresión logística (20)
  - El famoso SVM (25)
- Manos a la obra (60)
- Cierre de tecnicas y metodologia en ML

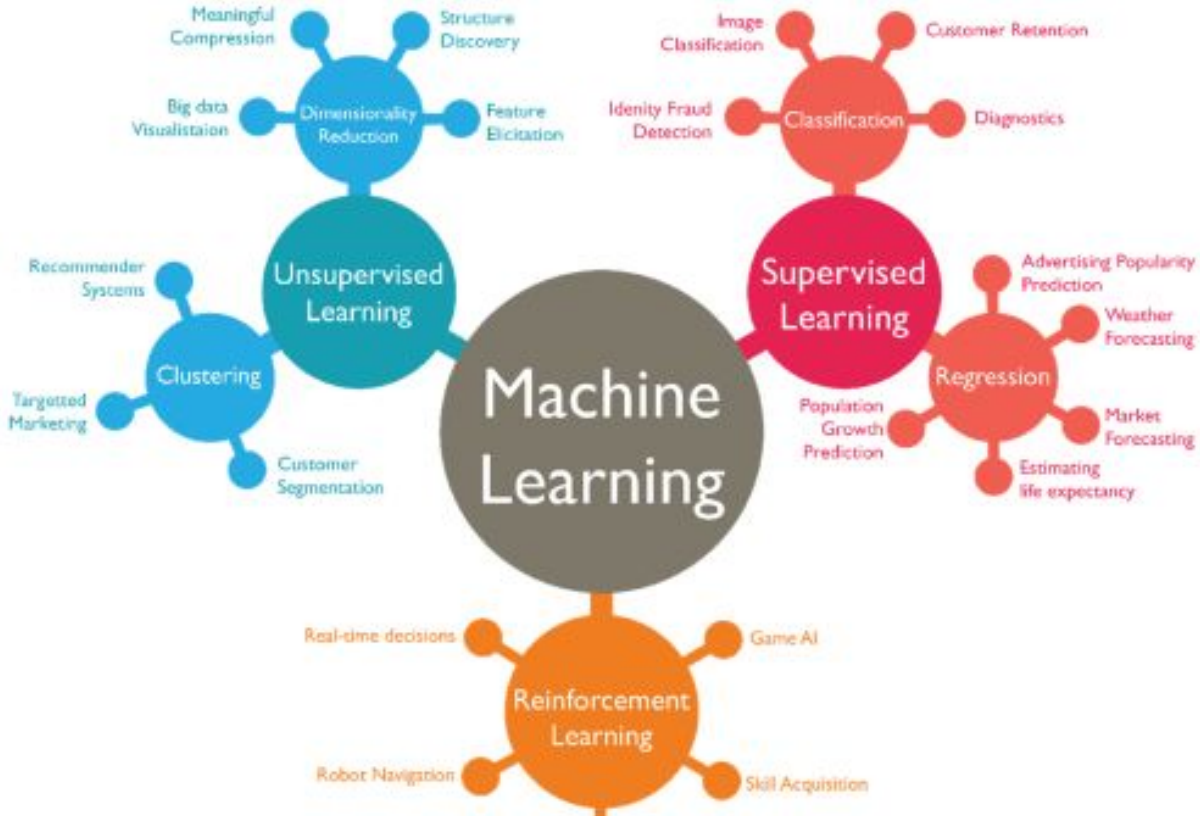
Variance vs Bias tradeoff



Variance vs Bias tradeoff



El árbol

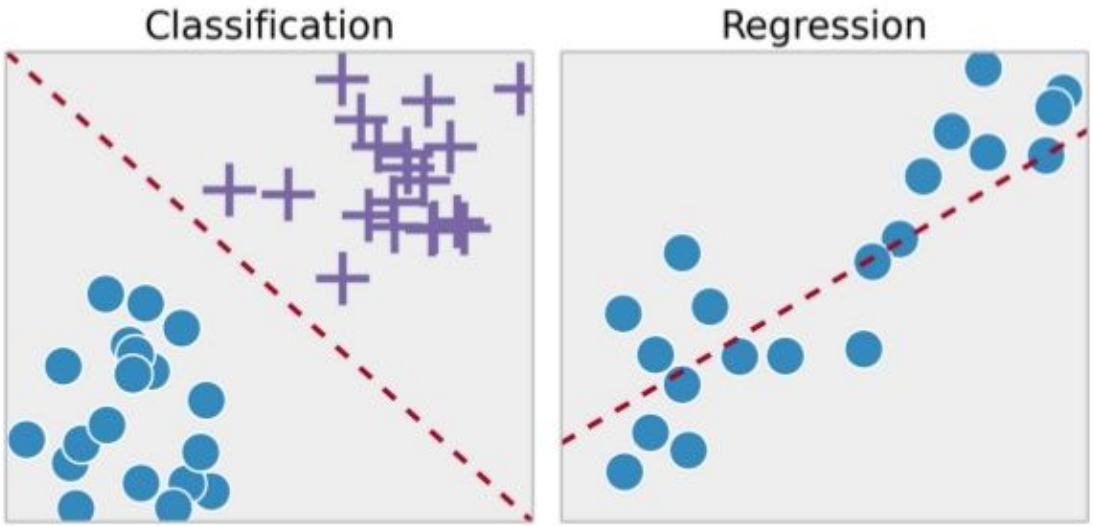

Machine Learning
Unsupervised Learning
Meaningful Compression
Structure Discovery
Dimensionality Reduction
Big data Visualistaion
Feature Elicitation
Recommender Systems
Clustering
Targetted Marketing
Customer Segmentation
Supervised Learning
Image Classification
Customer Retention
Classification
Idenity Fraud Detection
Diagnostics
Regression
Advertising Popularity Prediction
Weather Forecasting
Population Growth Prediction
Market Forecasting
Estimating life expectancy
Reinforcement Learning
Real-time decisions
Game AI
Robot Navigation
Skill Acquisition

Aprendizaje supervisado

- Déf:
- Es la tarea de aprendizaje automático de aprender una **función** que mapea una entrada a una salida basada en pares de entrada-salida de ejemplo.

$$f(x_i, y) \longrightarrow \hat{y}$$

Aprendizaje Supervisado



Regresión (valores continuos)

- Predicción
- Regresión lineal, multiple
- knn
- Árboles
  - Más árboles, un bosque

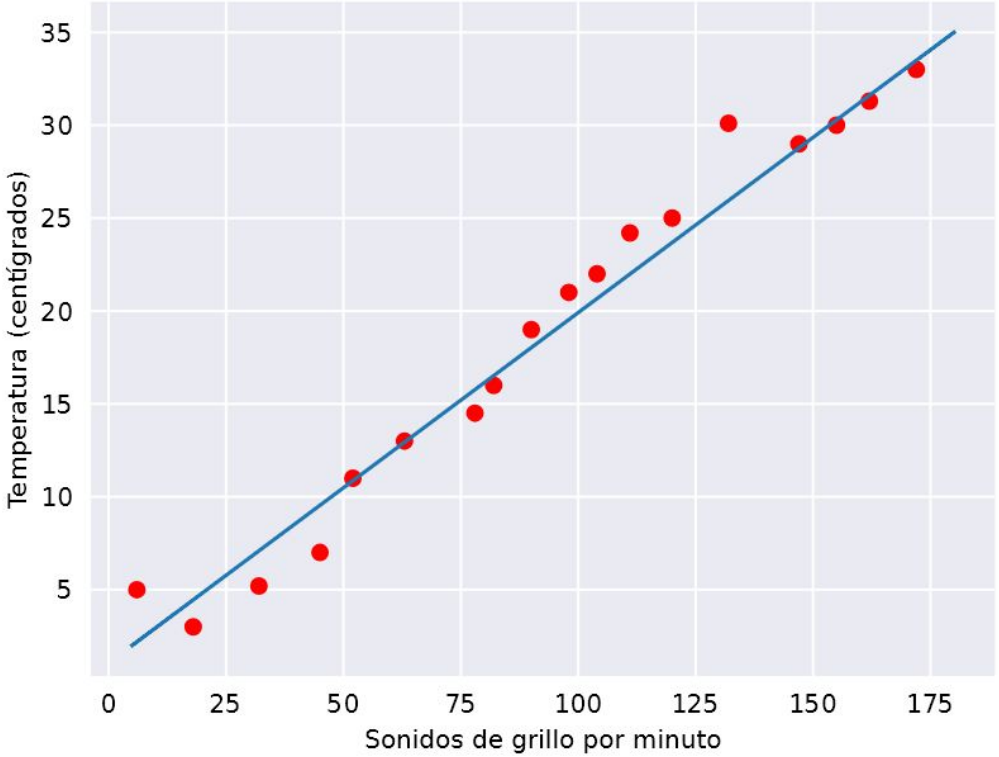
Regresión lineal. Pilar

- La regresión lineal es un método para encontrar la línea recta o el hiperplano que mejor se adapta a un conjunto de ***puntos***

$$\hat{Y} = \hat{\beta}_0 + \sum_{j=1}^{p} X_j \hat{\beta}_j$$

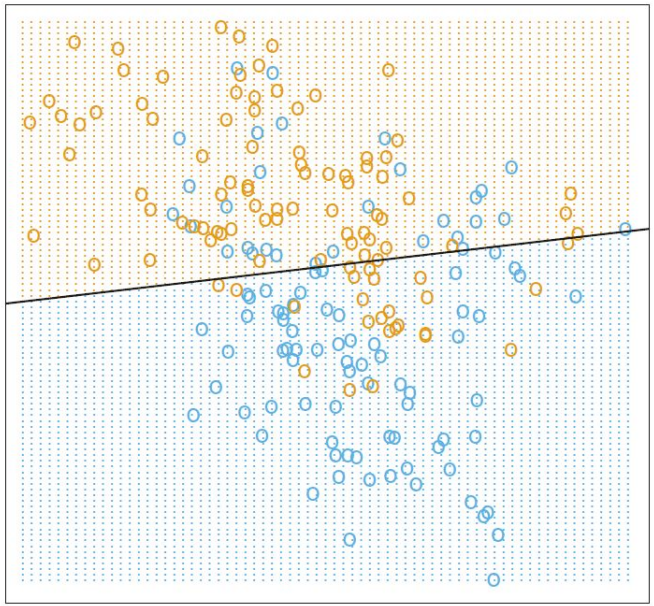
$$\mathrm{RSS}(\beta) = \sum_{i=1}^{N} (y_i - x_i^T \beta)^2$$

Regresión lineal



$$y' = b + w_1 x_1$$

Regresión lineal. Pilar



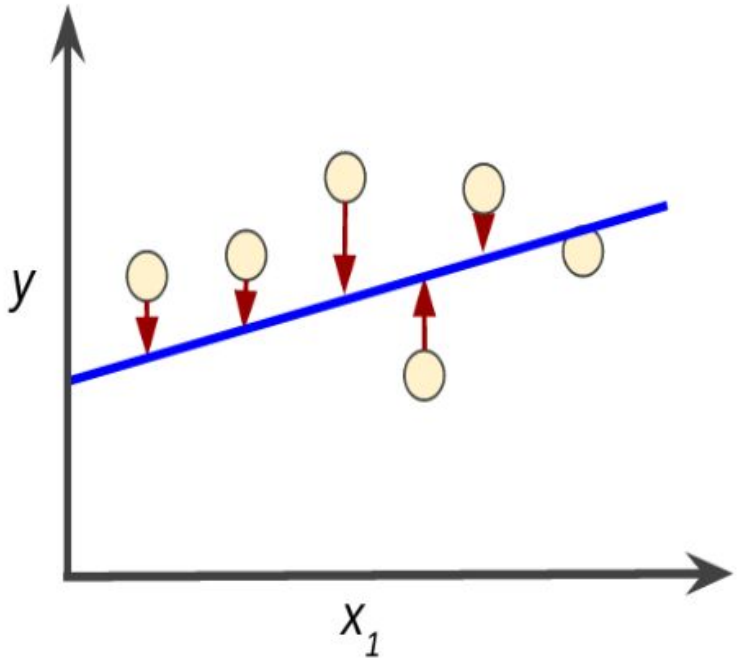
Ejemplito:
https://fou-foo.shinyapps.io/likelihoodratiotest/

Regresión

- Por cada registro: Predictores y una salida numérica
- El modelo es una función **lineal** de los predictores hacia la salida

Regresión (función de error)

$$\mathrm{RSS}(\beta) = \sum_{i=1}^{N} (y_i - x_i^T \beta)^2$$



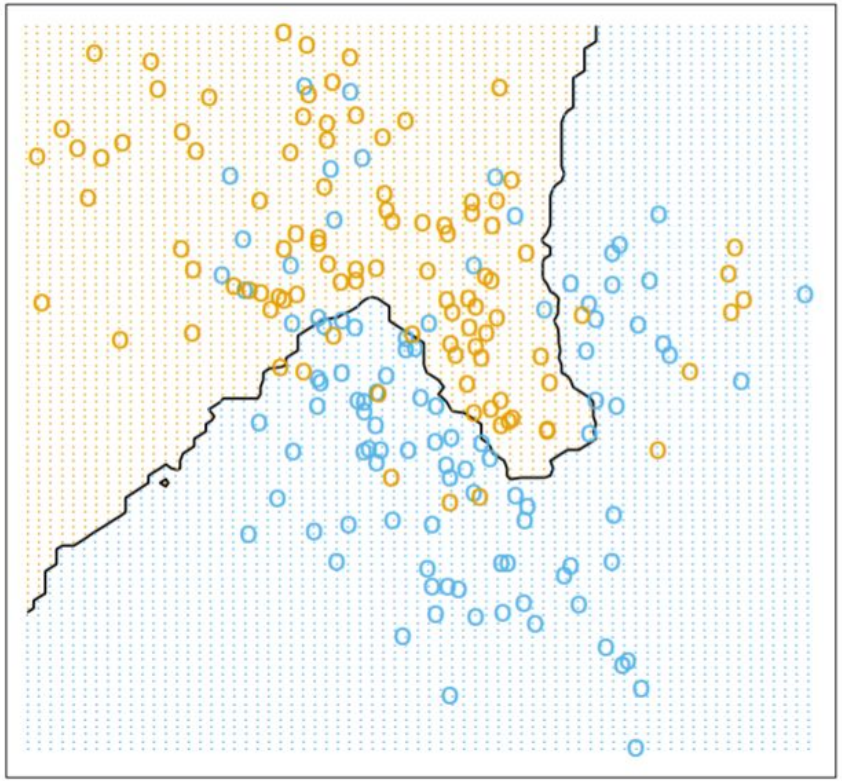
Localidad: *knn*

$$\hat{Y}(x) = \frac{1}{k} \sum_{x_i \in N_k(x)} y_i$$

Algoritmo 4: knn

1) Fije una vecindad $N_k(x)$ donde x es cualquier punto de su conjunto de entrenamiento y k es el número de puntos del conjunto más cercanos a él sin contemplar a él mismo x
   La cercanía implica una métrica, como la distancia euclidiana.
2) Promedie sus respuestas

Regresión local knn (k=15)



Regresión menos local (árboles)

Qué hacen ?

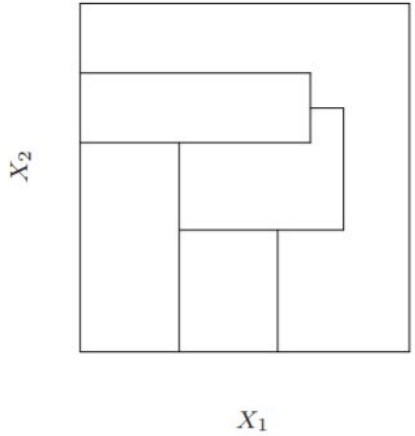
- Estratifican o segmentan el espacio del predictor en una serie de regiones simples
- Para hacer una predicción en una observación, utilizamos la media de las observaciones en la región a la que pertenece
- Dado que el conjunto de reglas de división utilizado para segmentar el espacio del predictor se puede resumir en un árbol, pues les decimos **árbol de decisión.**

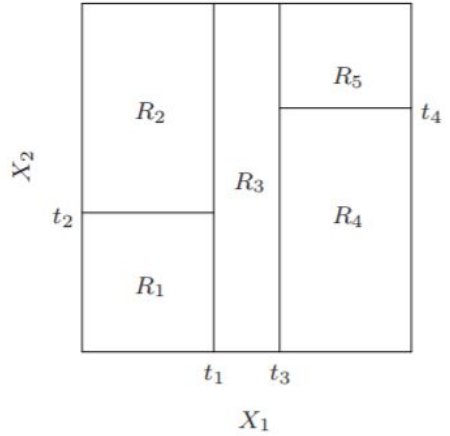
Árboles (aka CART por Breiman, 1984)

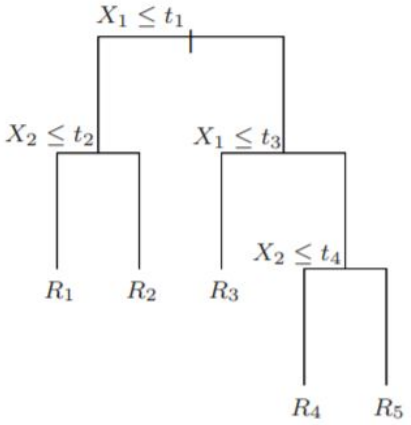
Y luego ?

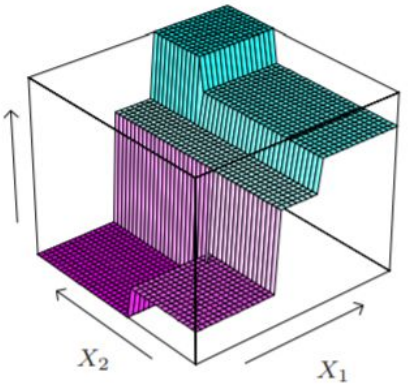
- Son simples y útiles para la interpretación, aunque poco potentes se puede mejorar con boosting y bagging
- Se pueden aplicar tanto a problemas de **regresión** como de **clasificación**

Árboles


X_2
X_1


R_1
R_2
R_3
R_4
R_5
t_1
t_2
t_3
t_4
X_1
X_2


$X_1 \leq t_1$
$X_2 \leq t_2$
$X_1 \leq t_3$
$X_2 \leq t_4$
R_1
R_2
R_3
R_4
R_5


X_2
X_1

Árboles

Algoritmo 5: Construcción de árbol de decisión

1) Compruebe los casos base anteriores
2) Para cada atributo a, encuentre la relación de ganancia de información, desde la división hasta a_{best}
3) Crear un nodo de decisión que se divide en
4) Recurra en las listas secundarias obtenidas dividiendo en a_{best}, y agregue esos nodos como hijos de nodo hasta que cada partición tenga *k* elementos

Árboles (puntos a considerar)

- Número mínimo de elementos en cada subpartición
- Número total de nodos de decisión (profundidad del árbol número de diferentes a)
- Son muy fáciles de explicar a la gente. De hecho más fáciles de explicar que regresión lineal
- Pueden manejar fácilmente predictores cualitativos sin la necesidad de crear variables ficticias (dummies)

- Cuidado con el sobreajuste!
  - Pueden ser muy poco robustos. Una pequeño cambio en los datos puede causar un gran cambio en la estimación final

Árboles (subproducto)

- Los a_{best} inducen un rankeo en las variables !

Árboles (puntos a considerar)

- En cada nodo a se aproxima una solución buscando optimizar:

- Regresión $\min_{j,\,s}\left[\min_{c_1}\sum_{x_i\in R_1(j,s)}(y_i-c_1)^2+\min_{c_2}\sum_{x_i\in R_2(j,s)}(y_i-c_2)^2\right].$

- Clasificación

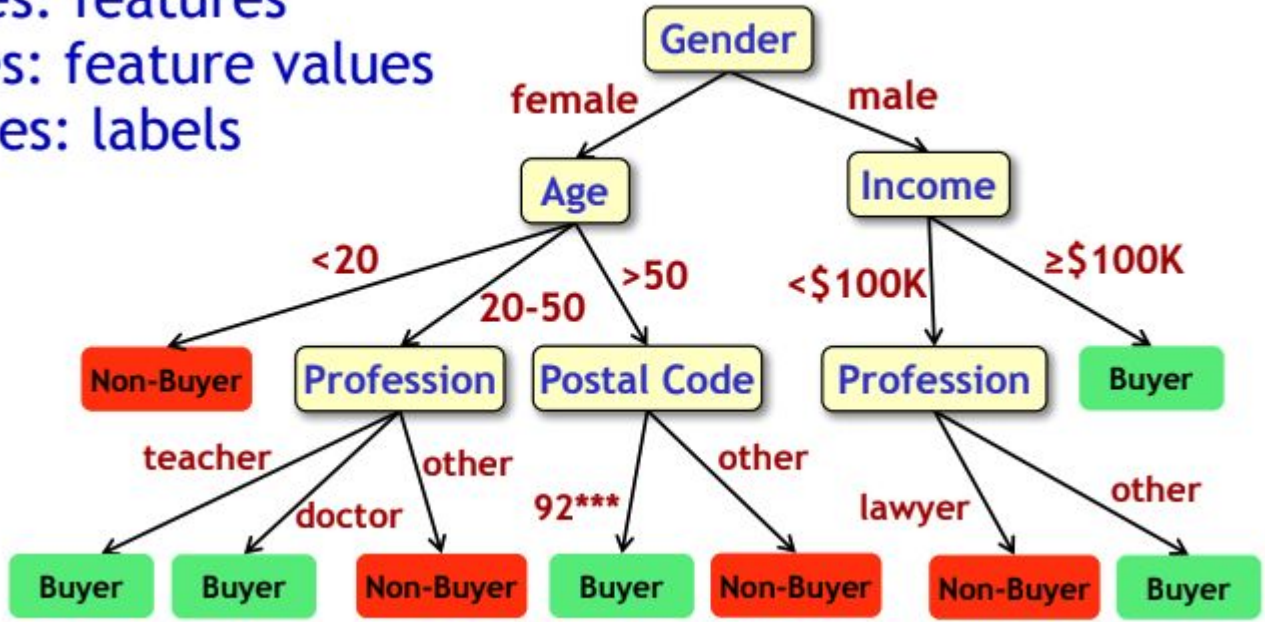
Misclassification error: $\frac{1}{N_m}\sum_{i\in R_m} I(y_i\neq k(m)) = 1-\hat{p}_{mk(m)}$

Gini index: $\sum_{k\neq k'}\hat{p}_{mk}\hat{p}_{mk'} = \sum_{k=1}^{K}\hat{p}_{mk}(1-\hat{p}_{mk})$.

Cross-entropy or deviance: $-\sum_{k=1}^{K}\hat{p}_{mk}\log\hat{p}_{mk}$.

Árboles de decisión

Nodes: features
Edges: feature values
Leaves: labels



Más árboles, un bosque

Qué hacen ?

- La idea esencial es promediar muchos ruidos con modelos aproximadamente insesgados para reducir la varianza

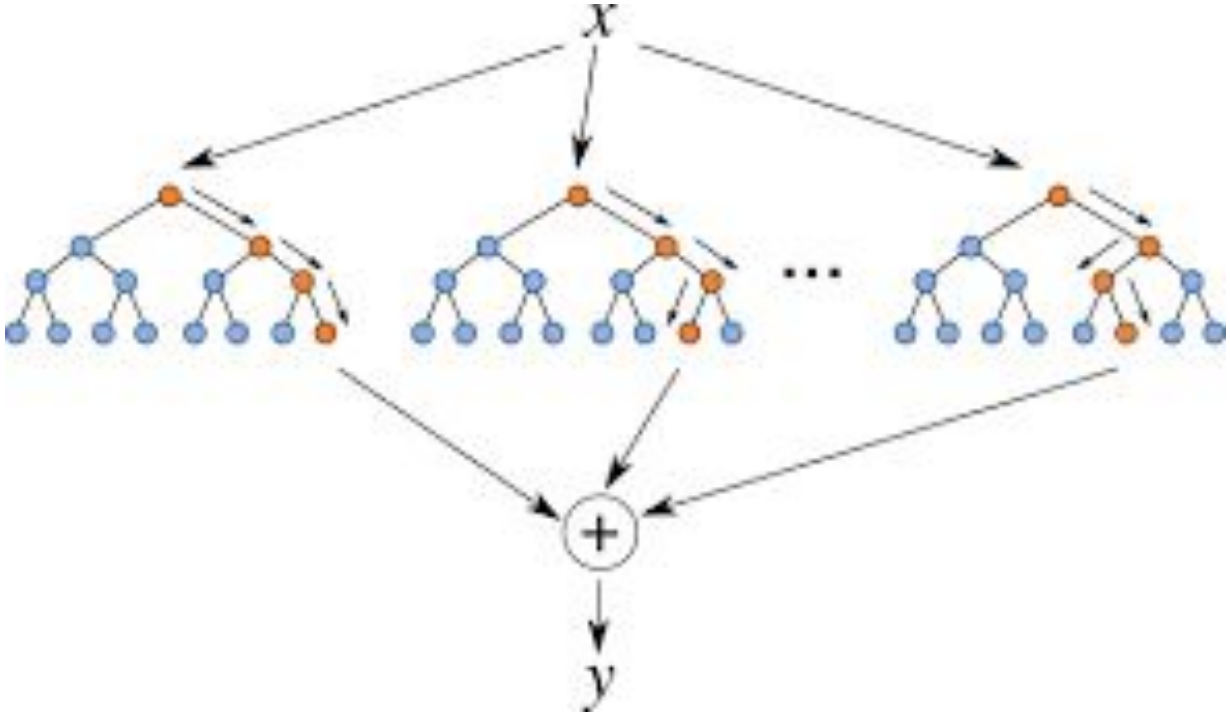
Random forest

Algoritmo 6: Random forest

1) Para $b = 1, \ldots, B$:
   a) Tome una muestra con reemplazo de tamaño N
   b) Construya un árbol T_b y obtenga una predicción

2) La predicción final se obtiene promediando $\{T_i\}_b^B$

Random forest



Clasificación (valores discretos)

- Separación
  - knn
  - Regresión logística
  - El famoso SVM y su primo DWD

Regresión logística

- Supongamos que la probabilidad de un valor puede ser modelada linealmente:

$$p(X) = \beta_0 + \beta_1 X$$

- El *detalle* es que esto no siempre vive en $[0, 1]$ por eso se considera el modelo:

$$p(X) = \frac{e^{\beta_0 + \beta_1 X}}{1 + e^{\beta_0 + \beta_1 X}}$$

Regresión logística

- Que es equivalente (un poquito de álgebra) a

$$\log\left(\frac{p(X)}{1-p(X)}\right) = \beta_0 + \beta_1 X.$$

Regresión logística

1) Resuelva la ecuación anterior, como dios le permita

Regresión logística

Regresión logística

1) Resuelva la ecuación anterior, o con un algoritmo iterativo como Newton-Raphson

$$\log\left(\frac{p(X)}{1-p(X)}\right) = \beta_0 + \beta_1 X_1 + \cdots + \beta_p X_p$$

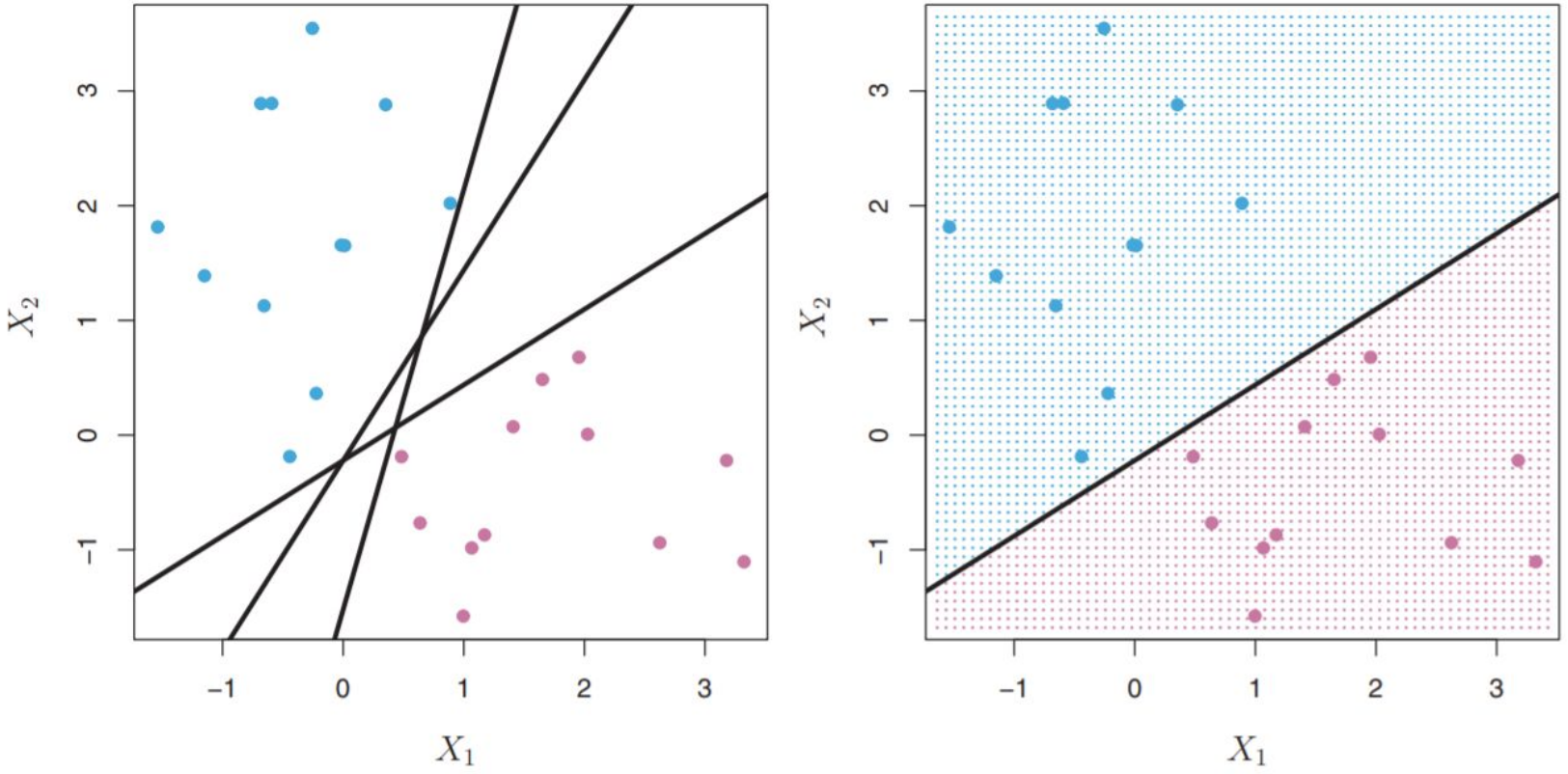
Support vector machine

Qué hacen ?

- La idea es encontrar el mejor hiperplano separador

- Nuevecito de Hava Siegelmann y Vladimir Vapnik (2010)

Support vector machine



Support vector machine

SVM y DWD

1) Resuelva el problema de optimización, como dios le permita

https://joseramirezcimat.shinyapps.io/DWD1/

> Manos a la obra

Sigue el Jupyter Notebook