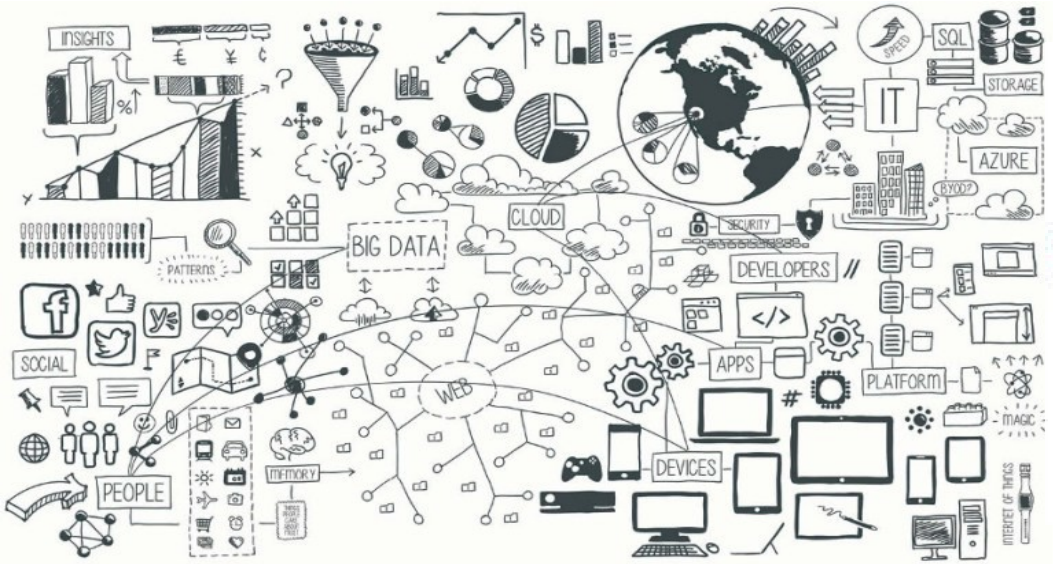
Data Mining (Minería de Datos)

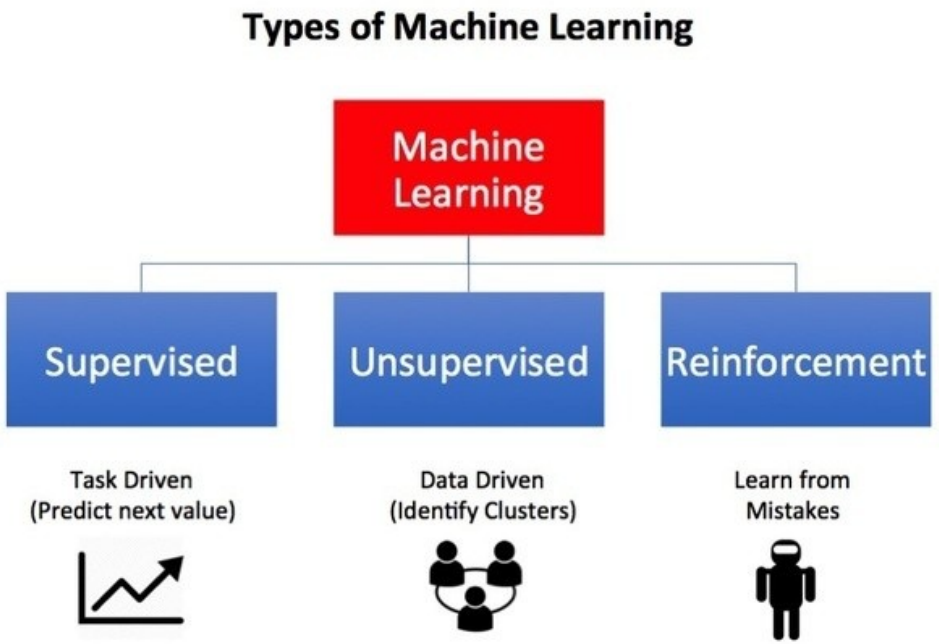
PARADIGMS, CANONICAL PROBLEMS AND DATASETS



Malialen Iturbide
José Manuel Gutiérrez

Grupo de Meteorología

Univ. de Cantabria – CSIC
MACC / IFCA



NOTA: Las líneas de código de R en esta presentación se muestran sobre un fondo gris.

Oct	30	Aplazada (sesión de refuezo)
Nov	6	Presentación, introducción y perspectiva histórica
	8	**Paradigmas, problemas canonicos y data challenges**
	13	Reglas de asociación
	15	Practica: Reglas de asociación
	20	Evaluación, sobrejuste y crossvalidacion
	22	Practica: Crossvalidacion
	27	Arboles de clasificacion y decision
	29	Practica: Arboles de clasificación
		T01. Datos discretos
Dic	4	Técnicas de vecinos cercano (k-NN)
	11	Práctica: Vecinos cercanos
	13	Reducción de dimensión lineal
	18	Practica: LDA y PCA
	20	Reducción no lineal
		T02. Clasificación
Ene	8	Arboles de clasificación y regresion (CART)
	10	Practica: CART
	15	Ensembles: Bagging and Boosting
	17	Practica Random Forests
		T03. Prediccion
	22	Practica Gradient boosting
	24a	Técnicas de agrupamiento
	24b	Practica: Técnicas de agrupamiento
	29a	Practica: El paquete CARET
	29b	Examen

Sectores de aplicación

Financiero
Seguros

Comercio y
marketing

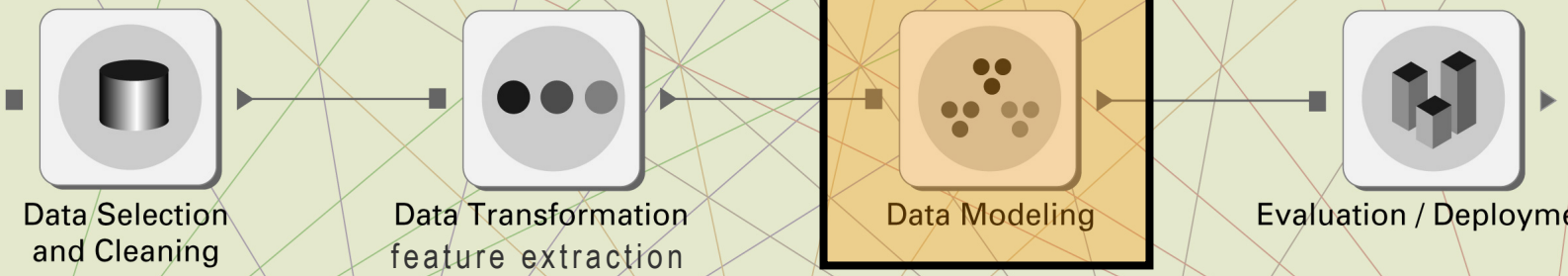
Industria y
empresarial

Tecnologías
información y
comunicación

Sanitario y
farmacéutico

Meteorología
Medio Ambiente

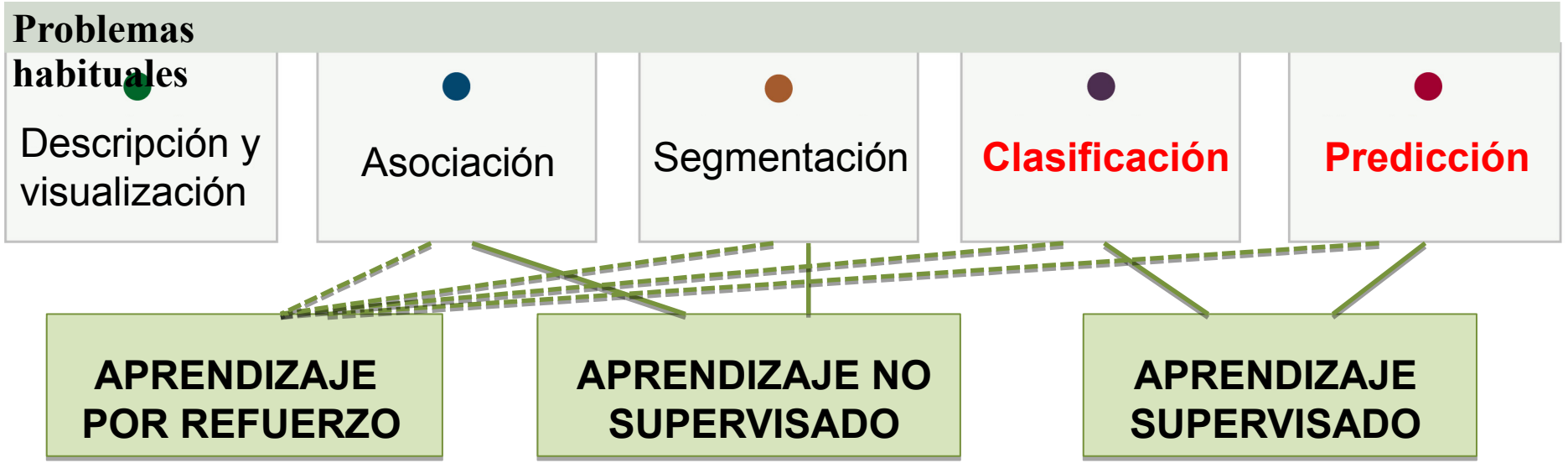
Proceso de Minería de Datos



Problemas habituales (canónicos):

- Asociación
- Reduccion de dimensión
- Segmentación
- Clasificación
- Predicción

Machine learning develop methods for data modelling and prognosis.



Datos de entrada *(X)*: $(X_1, X_2, \ldots, X_n)$

Aprendizaje supervisado: Se entrena con datos *(X)* que han sido etiquetados ("label") $(y_1, y_2, \ldots)$. Las etiquetas clasifican cada punto de datos en uno o más grupos, como "manzanas" o "naranjas". El sistema aprende cómo se estructuran estos datos, se entrena de manera que minimiza el error de predicción del sistema. El objetivo es predecir las categorías de datos nuevos o de "test".

Aprendizaje NO supervisado: Se trata de agrupar e interpretar los datos sólo con los datos de entrada *(X)*.

Aprendizaje por refuerzo: Se encuentra entre el aprendizaje supervisado y no supervisado. Se centra en ir aprendiendo de la experiencia. Recive recompensas o castigos $(r_1, r_2, \ldots)$ de las acciones $(a_1, a_2, \ldots)$ que realiza. El objetivo es maximizar las recompensas.

Problemas habituales

Descripción y visualización

Asociación

Segmentación

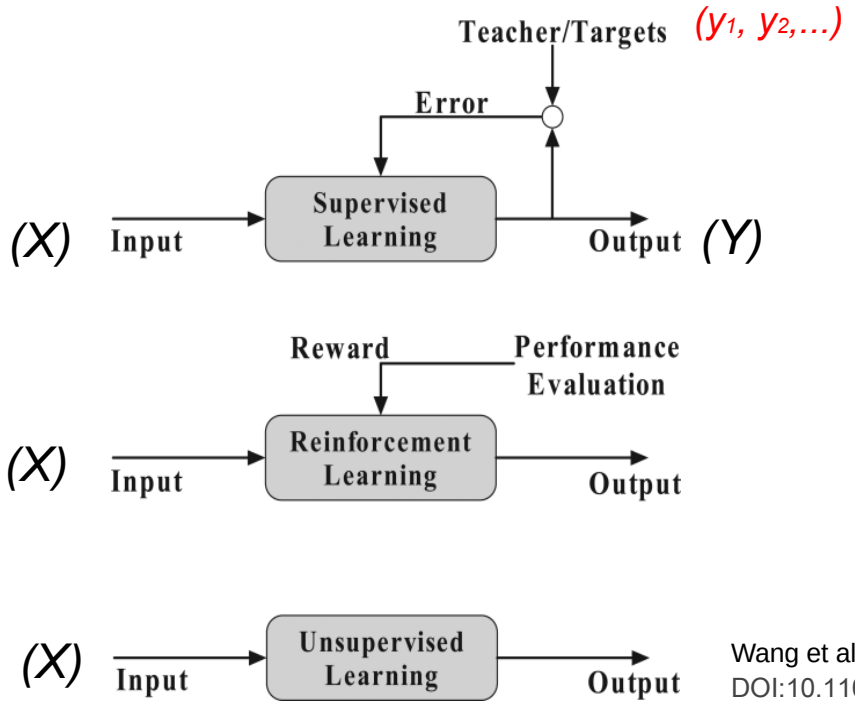
Clasificación

Predicción

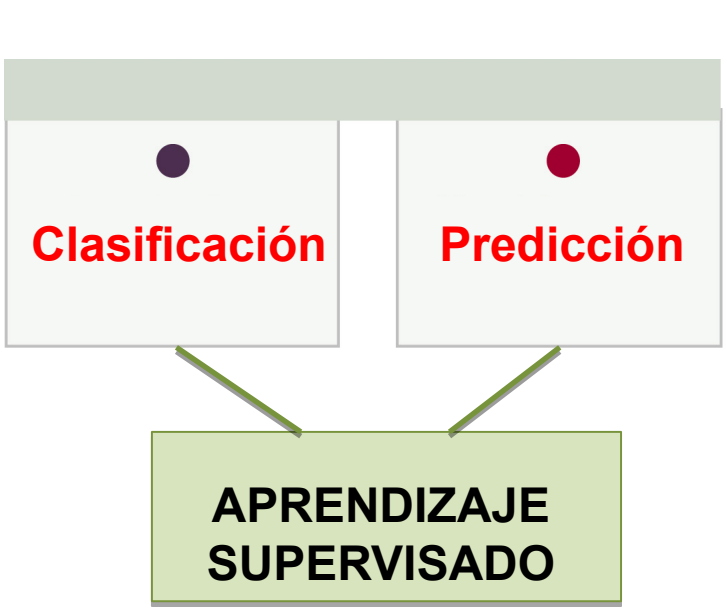
APRENDIZAJE POR REFUERZO

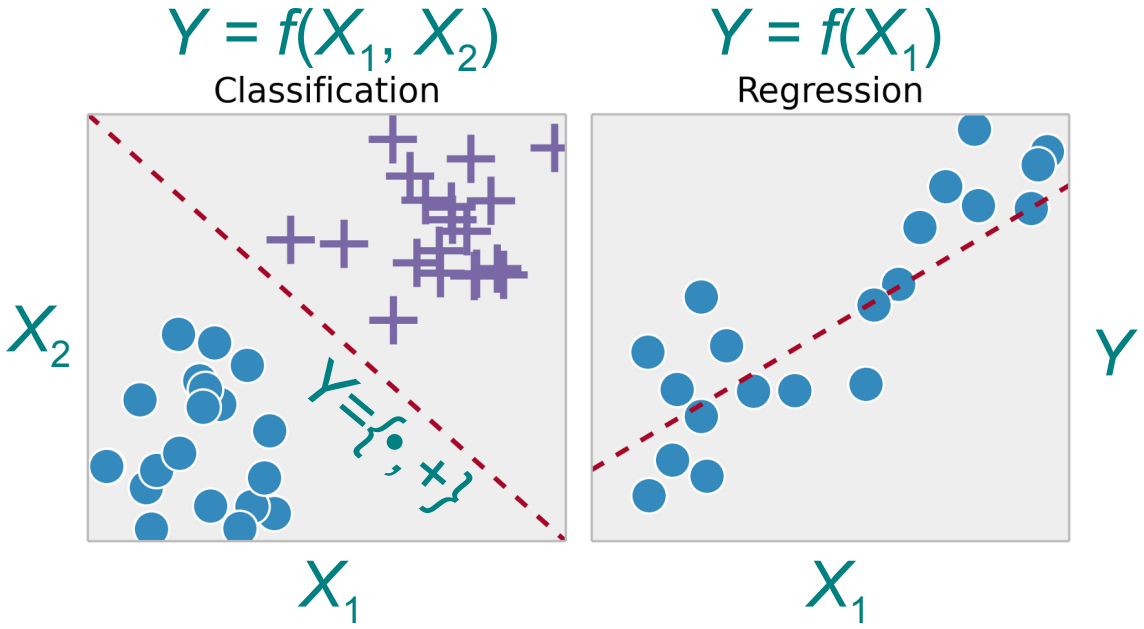
APRENDIZAJE NO SUPERVISADO

APRENDIZAJE SUPERVISADO

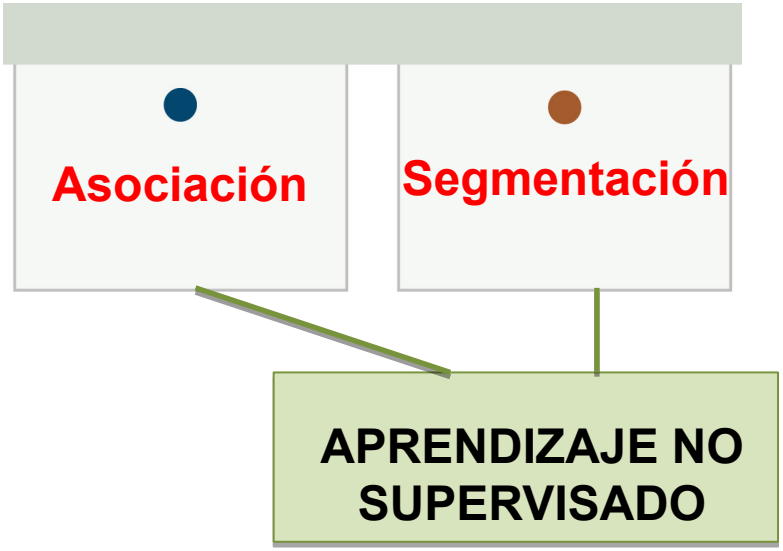


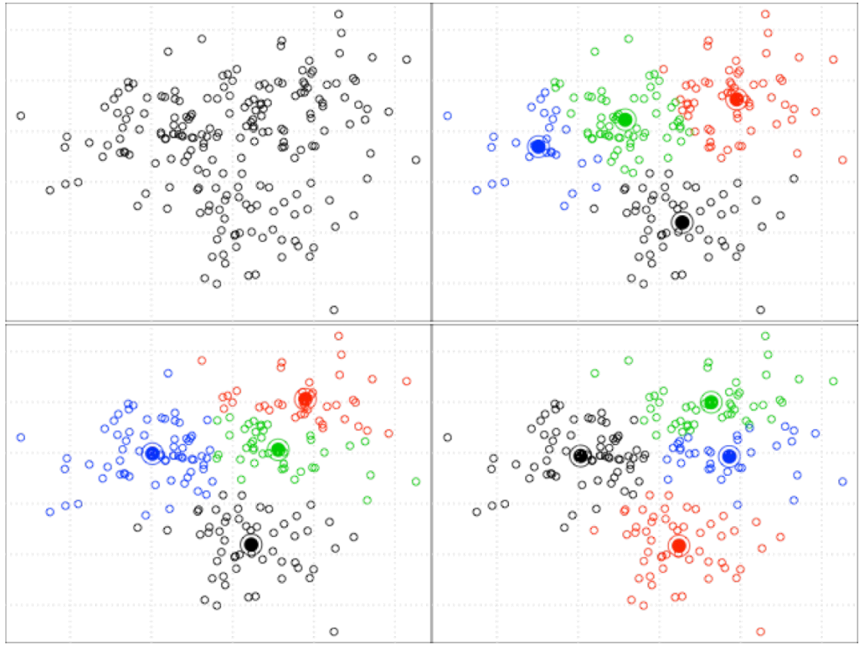
Wang et al. 2012
DOI:10.1109/TSMCC.2012.2186565





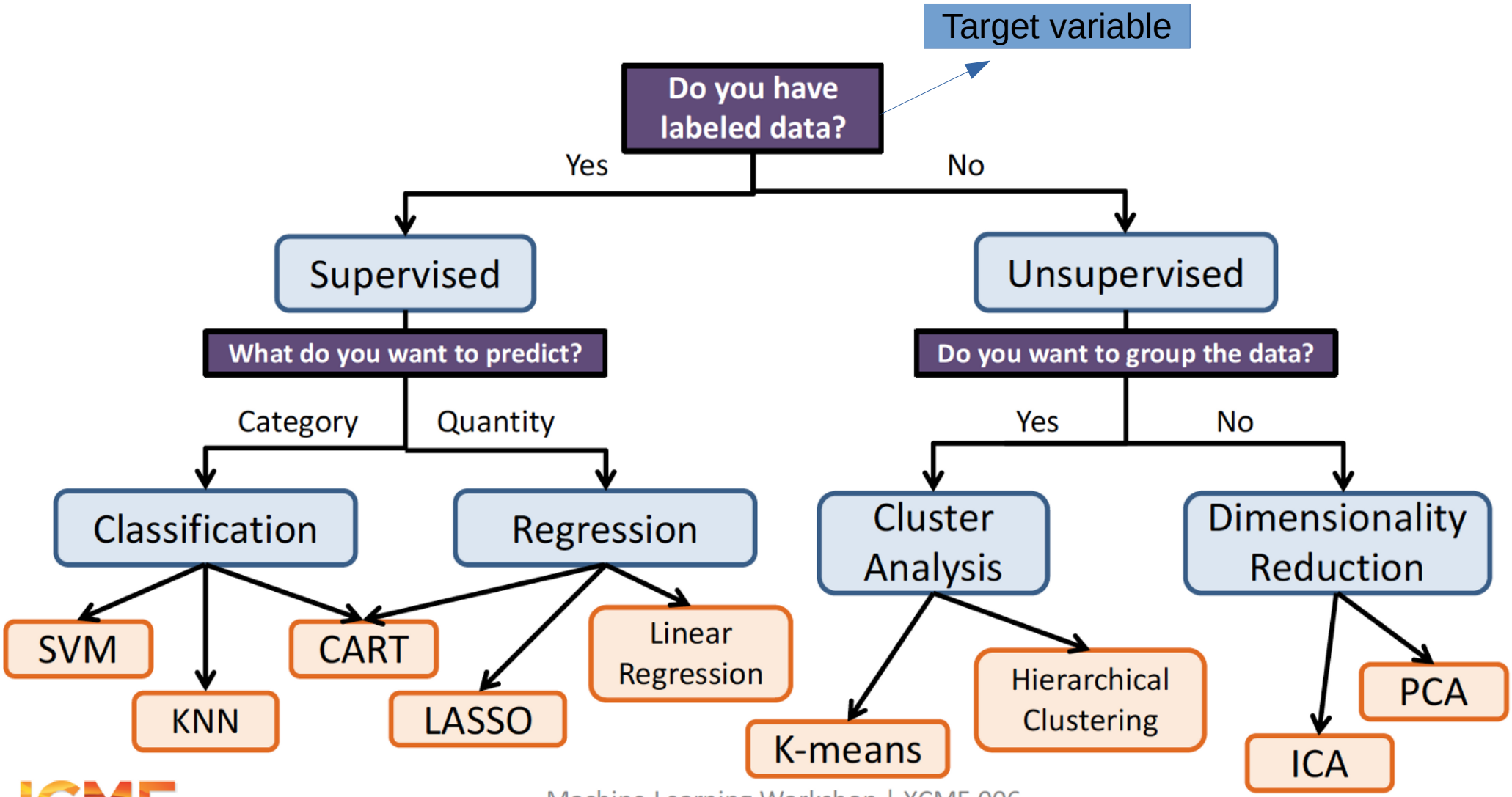
- Target Variable: Y : *categorical/factor* or *continuous*
  - What we are trying to predict.
- Predictive Variables: $\{X_1, X_2, \ldots, X_N\}$: *continuous or factor*
  - "Covariates" used to make predictions.
- Predictive Model: $Y = f(X_1, X_2, \ldots, X_N)$
  - "Learning engine" that estimates the f (or the parameters defining f).



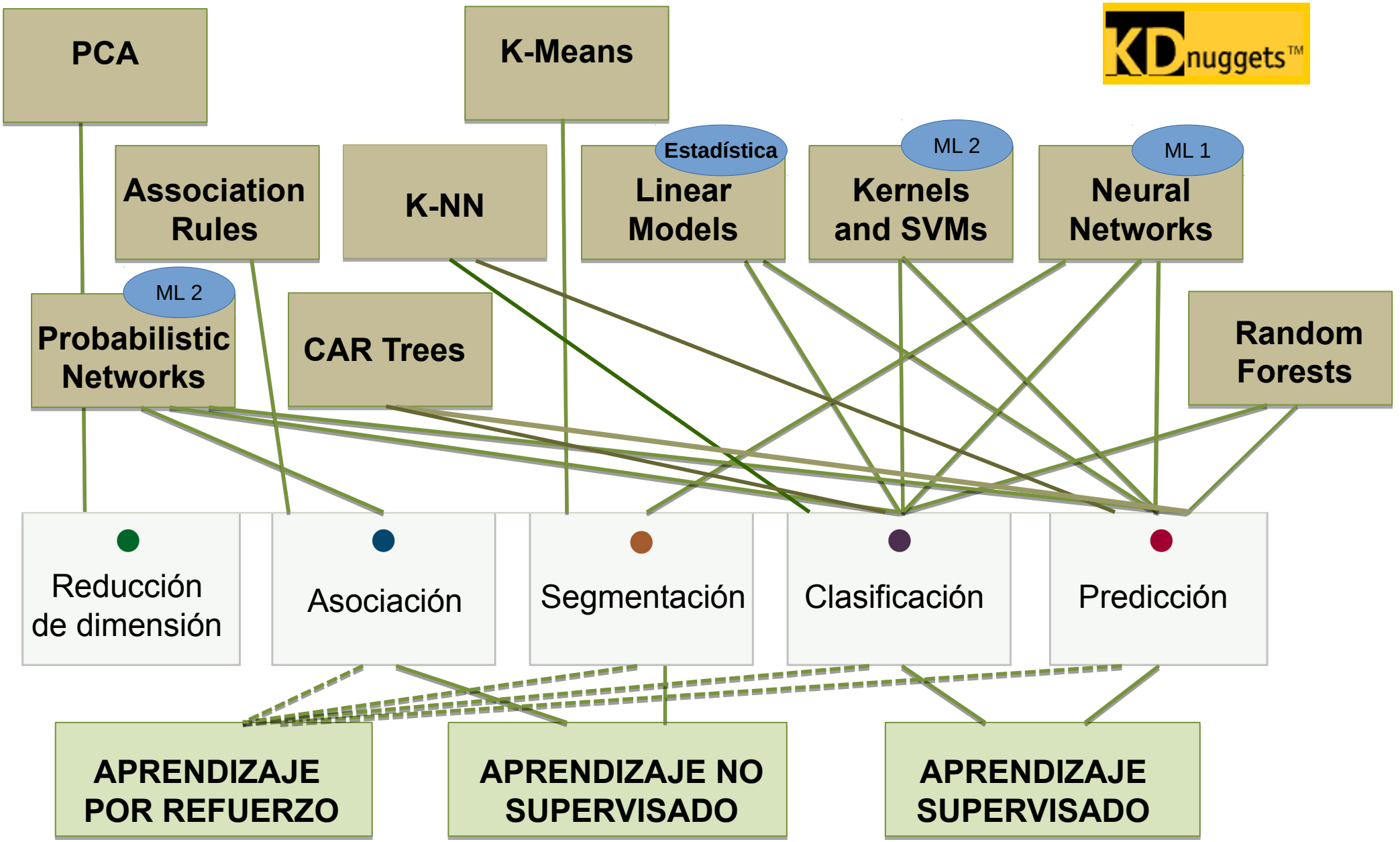


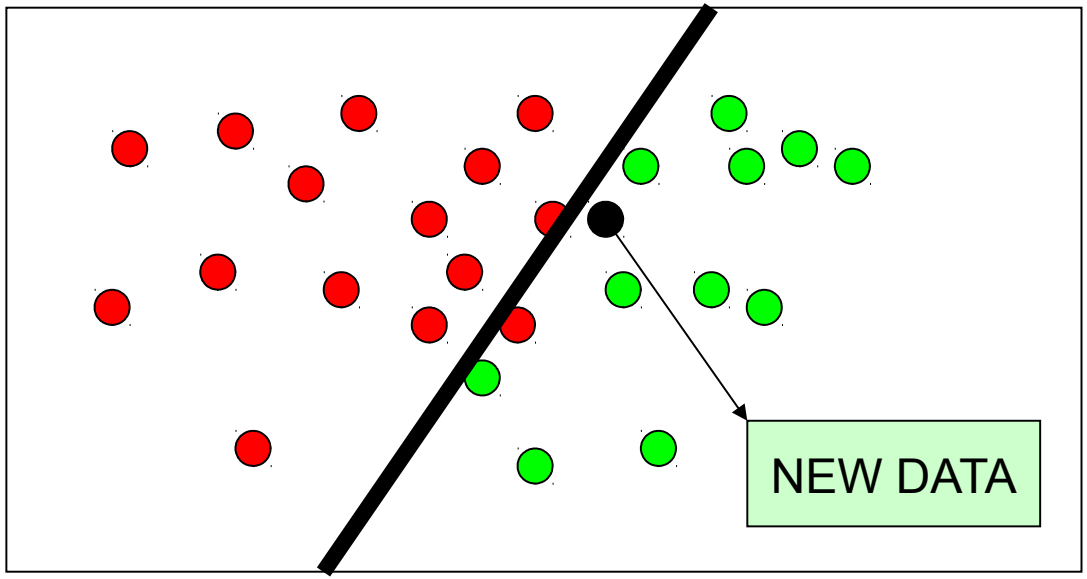
discrete: #clusters

- Target Variable: *There is no target variable*
- Variables: $\{X_1, X_2, \ldots, X_N\}$: *continuous* or *factor*
  - "Covariates" used to make predictions.
- Predictive Model: Algorithmic, based on $(X_1, X_2, \ldots, X_N)$.
  - Ad-hoc "learning" and "prediction" engine.


Target variable
Do you have labeled data?
Yes
No
Supervised
Unsupervised
What do you want to predict?
Do you want to group the data?
Category
Quantity
Yes
No
Classification
Regression
Cluster Analysis
Dimensionality Reduction
SVM
KNN
CART
LASSO
Linear Regression
K-means
Hierarchical Clustering
ICA
PCA

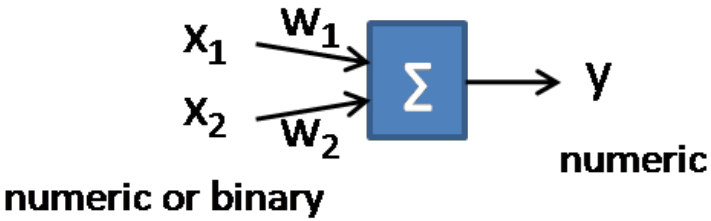
https://www.kdnuggets.com/2017/10/top-10-machine-learning-algorithms-beginners.html





GENERATIVE METHODS:

Linear models are the simplest family for machine learning and have good generalization properties.

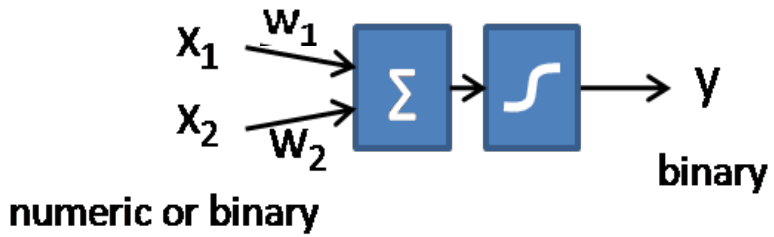


$$y = w_0 + w_1x_1 + w_2x_2$$

$$y = f(\mathbf{X},\mathbf{W}) = \mathbf{X}^T.\mathbf{W}$$

$$\mathbf{W} = \begin{pmatrix} w_0 \\ w_1 \\ w_2 \end{pmatrix} \quad \mathbf{X} = \begin{pmatrix} 1 \\ x_1 \\ x_2 \end{pmatrix}$$

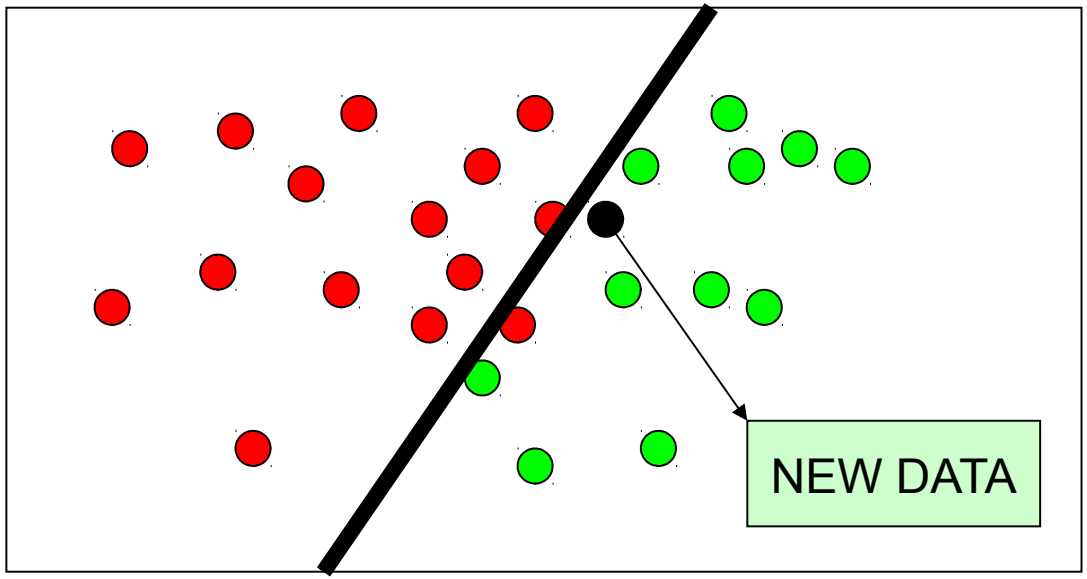
LINEAR REGRESSION



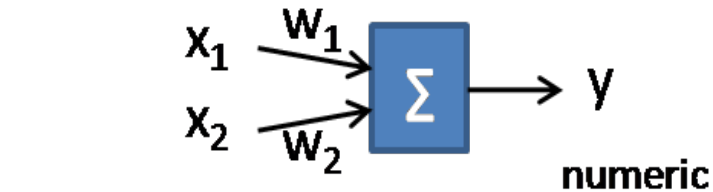
$$y = \text{sigmod}(w_0 + w_1x_1 + w_2x_2)$$
$$\ldots \text{ where sigmod}(k) = 1 / (1 + e^{-k})$$

$$y = f(\mathbf{X},\mathbf{W}) = \text{sigmod}(\mathbf{X}^T.\mathbf{W})$$

LOGISTIC REGRESSION

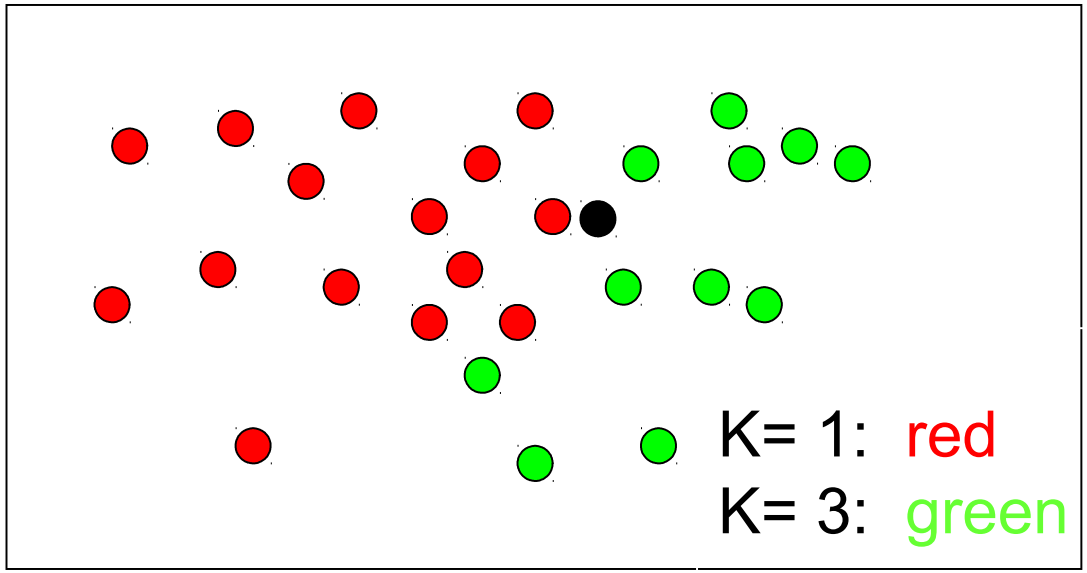


GENERATIVE METHODS:
Linear models are the simplest family for machine learning and have good generalization properties.



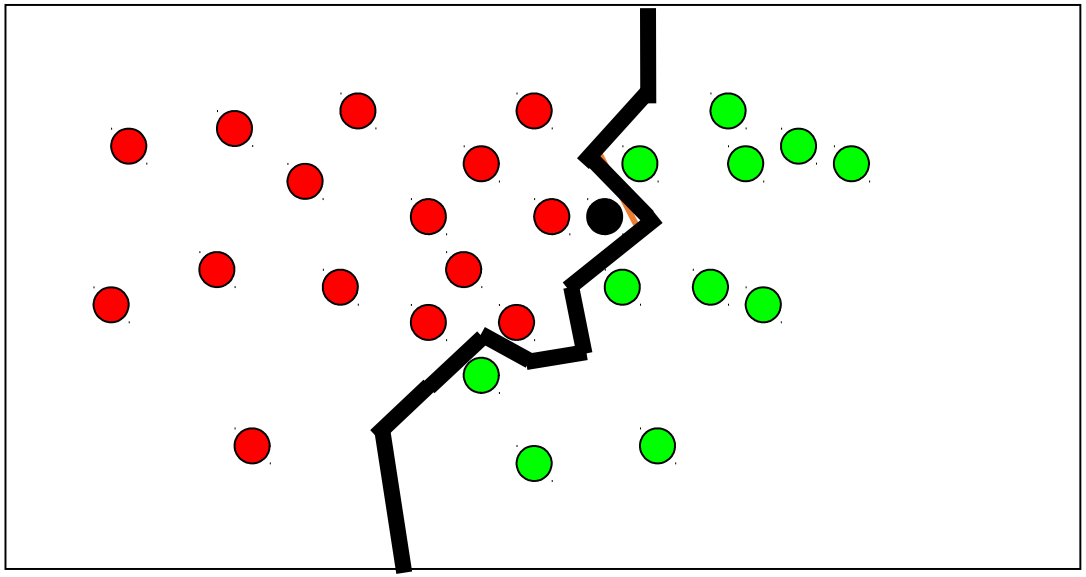
$$y = w_0 + w_1x_1 + w_2x_2$$

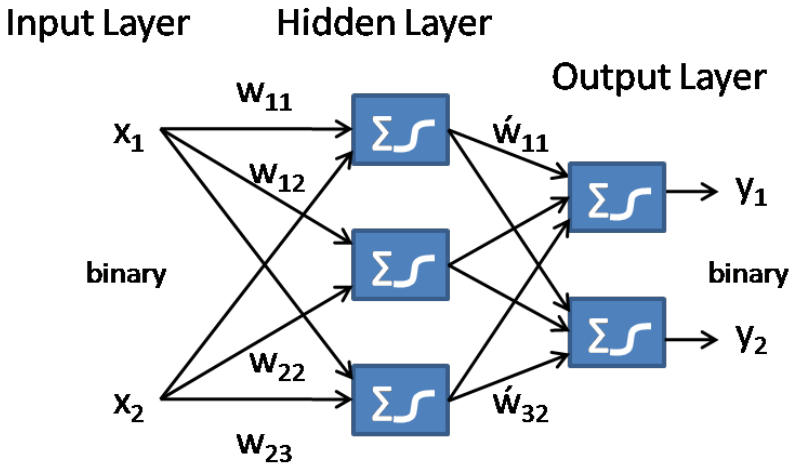
$$y = f(\mathbf{X},\mathbf{W}) = \mathbf{X}^T.\mathbf{W}$$



NON-GENERATIVE (OR ALGORITHMIC)
K Nearest Neighbours is the simples non-generative method. It depends on a single parameter (K) to be tunned (generalization depends on K).

Increasing model complexity (e.g. number of parameters) can result in **overfitting** (lack of generalization).





Existe una distinción clave entre los sistemas de aprendizaje en línea (**online**) y fuera de línea (**offline**):

- **Offline**

Los sistemas de aprendizaje **se entrenan y validan offline** y **se "congelan"** antes de empezar a ser utilizados por los usuarios. **Posteriores entrenamientos del sistema se realizarán de nuevo offline** para congelar una actualización que da lugar a una **nueva versión**.

Este proceso es el más común ya que posibilita la **verificación humana del sistema antes de que el sistema interactúe con el usuario.**

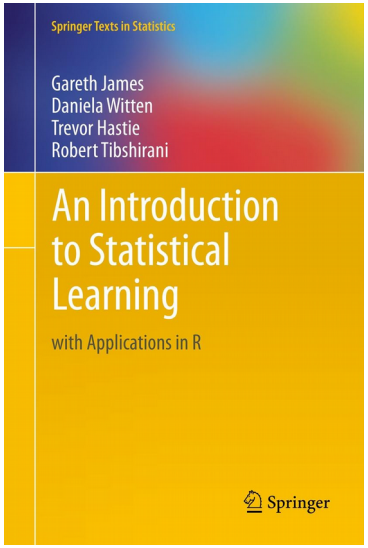
- **Online**

Los sistemas de aprendizaje se entrenan y validan offline, pero se actualizan **a medida que se tiene nueva información. El funcionamiento de los algoritmos de aprendizaje pueden mejorar en tiempo real**.

Un ejemplo son los sistemas de detección de spam que se entrenan en respuesta a los patrones del correo entrante y al feedback que da el usuario sobre la precisión del sistema.

Bibliografía/Repositorios de datos

1



An Introduction to Statistical Learning: With Applications in R

James, G., Witten, D., Hastie, T., Tibshirani, R.

Springer (2013)

http://www-bcf.usc.edu/~gareth/ISL

```
install.packages("ISLR")
library("ISLR")
library(help = "ISLR")
```

2

```
library(help = "datasets")
```

3

kaggle

https://www.kaggle.com/datasets

4

UCI Machine Learning Repository
Center for Machine Learning and Intelligent Systems

https://archive.ics.uci.edu/ml/datasets.html

kaggle Search kaggle Competitions Datasets Kernels Discussion Jobs ...

Overview Data Kernels Discussion Leaderboard Rules **Late Submission**

Overview

Description
Evaluation
Prizes
Timeline

In this competition, Kaggle is challenging you to build a model that predicts the total ride duration of taxi trips in New York City. Your primary dataset is one released by the NYC Taxi and Limousine Commission, which includes pickup time, geo-coordinates, number of passengers, and several other variables.

Longtime Kagglers will recognize that this competition objective is similar to the ECML/PKDD trip time challenge we hosted in 2015. But, this challenge comes with a twist. Instead of awarding prizes to the top finishers on the leaderboard, this playground competition was created to reward collaboration and collective learning.

https://www.kaggle.com/headsortails/nyc-taxi-eda-update-the-fast-the-curious/notebook

Listado de datasets utilizados en el curso

EN FUNCIÓN DE LA NATURALEZA DE LOS DATOS PODEMOS CLASIFICARLAS COMO

SÓLO CATEGÓRICAS (FACTORES)

- **Groceries.** Disponible en kaggle y en el paquete {arulesViz} de R.

- **Mushroom.** Disponible en kaggle y UCI.

MIXTOS (CONTINUOS Y FACTORES)

- **Iris.** Disponible en kaggle, UCI y el paquete {datasets} de R.

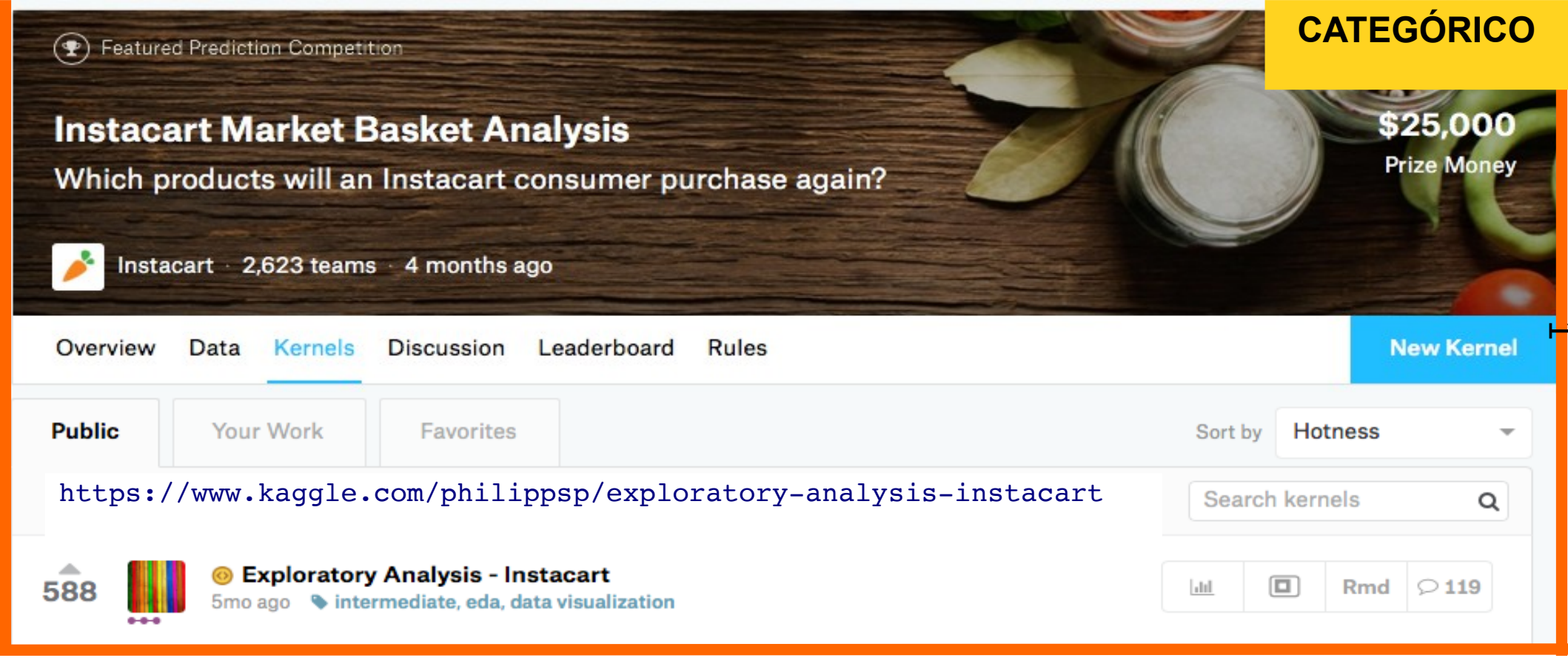
- **MNIST.** Disponible en https://pireddie.com/projects/mnist-in-csv/

Todos estos datasets están disponibles en **gitHub**:

https://github.com/SantanderMetGroup/Master-Data-Science

- **Meteo** (Santander Meteorology Group).

- **The fruits dataset by Dr. Iain Murray.** Disponible en



En el curso utilizaremos un dataset más pequeño, “Groceries”, disponible en el paquete de R **arulesViz**.

Attribute characteristics	Categorical
Number of instances	9835
Number of attributes	169

```
install.packages("arulesViz")
data(“Groceries”)
```

CATEGÓRICO

http://archive.ics.uci.edu/ml/datasets/Mushroom

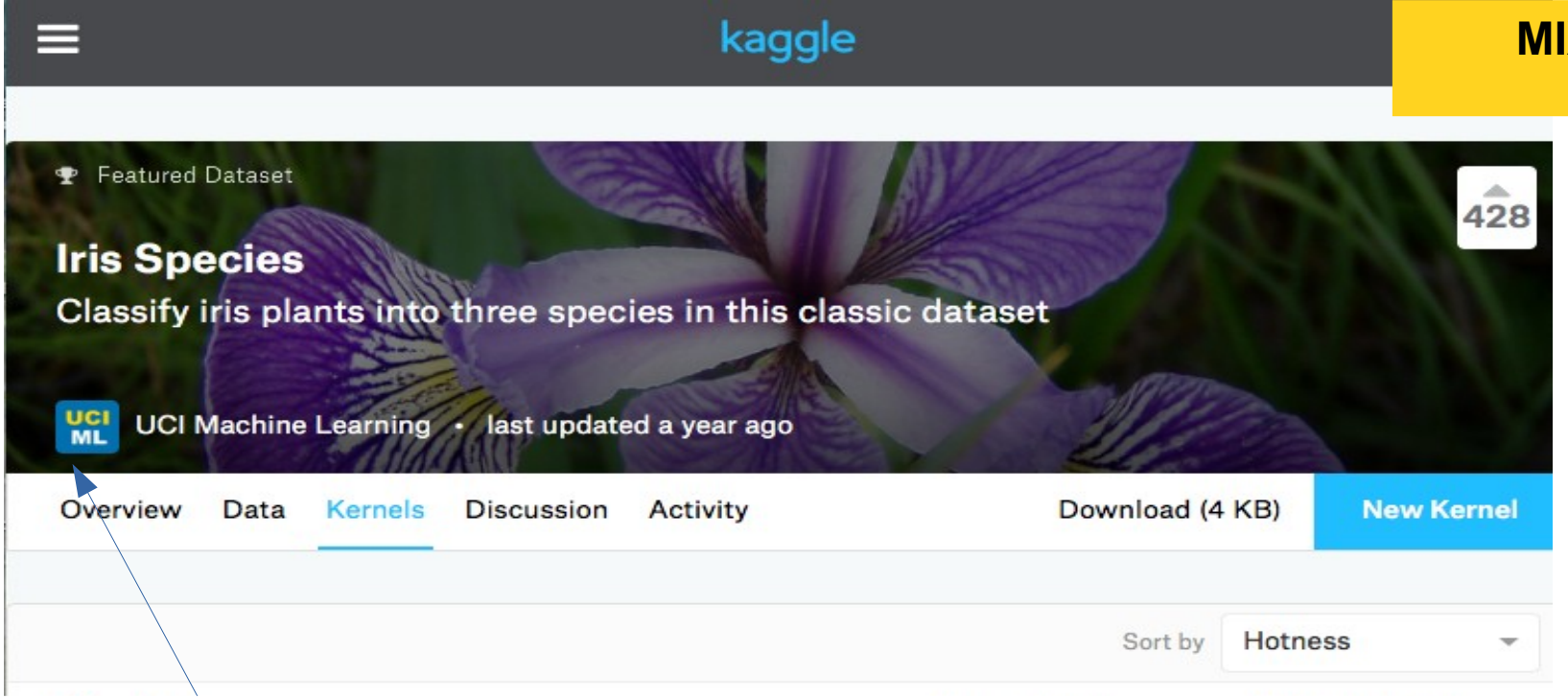
Data Set Characteristics:	Multivariate	**Number of Instances:**	8124	**Area:**	Life
Attribute Characteristics:	Categorical	**Number of Attributes:**	22	**Date Donated**	1987-04-27
Associated Tasks:	Classification	**Missing Values?**	Yes	**Number of Web Hits:**	298439

Attribute Information: (classes: edible=e, poisonous=p)

cap-shape: bell=b,conical=c,convex=x,flat=f, knobbed=k,sunken=s
cap-surface: fibrous=f,grooves=g,scaly=y,smooth=s
cap-color: brown=n,buff=b,cinnamon=c,gray=g,green=r,pink=p,purple=u,...
bruises: bruises=t,no=f
odor: almond=a,anise=l,creosote=c,fishy=y,foul=f,musty=m,none=n,...
...

```
mush <- read.csv("Data_mining/datasets/mushrooms.csv")
str(mush)
```

```
'data.frame':   8124 obs. of  23 variables:
 $ class                   : Factor w/ 2 levels "e","p": 2 1 1 2 1 1 1 1 2 1 ...
 $ cap.shape               : Factor w/ 6 levels "b","c","f","k",..: 6 6 1 6 6 6 1 1 6 1 ...
 $ cap.surface             : Factor w/ 4 levels "f","g","s","y": 3 3 3 4 3 4 3 4 4 3 ...
 $ cap.color               : Factor w/ 10 levels "b","c","e","g",..: 5 10 9 9 4 10 9 9 9 10 ...
 $ bruises                 : Factor w/ 2 levels "f","t": 2 2 2 2 1 2 2 2 2 2 ...
 $ odor                    : Factor w/ 9 levels "a","c","f","l",..: 7 1 4 7 6 1 1 4 7 1 ...
...
```



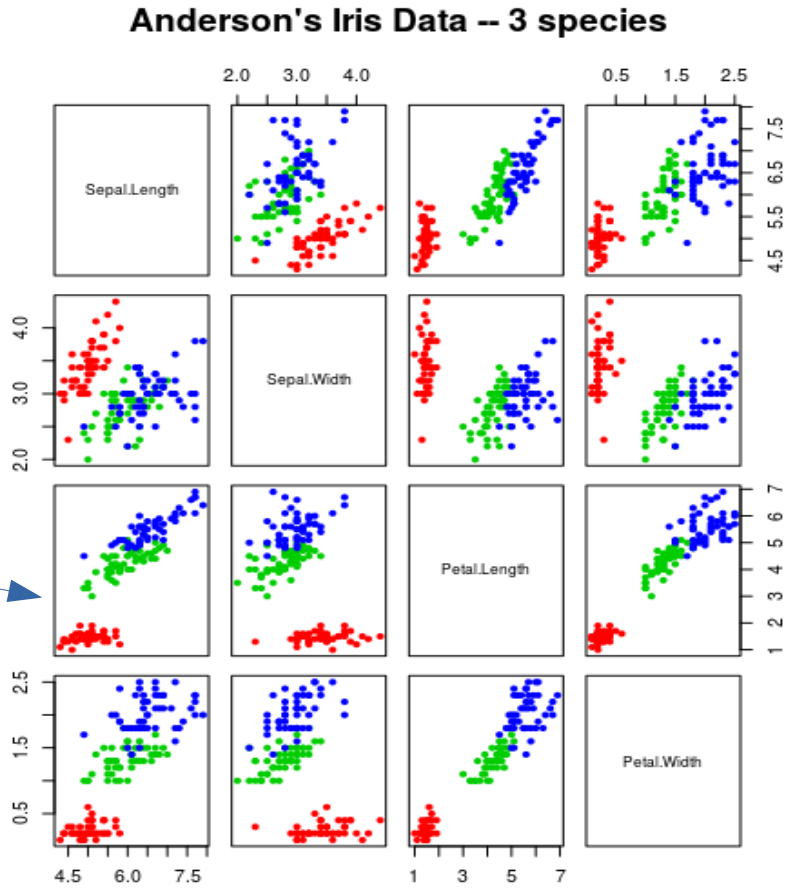
http://archive.ics.uci.edu
/ml/datasets/Iris

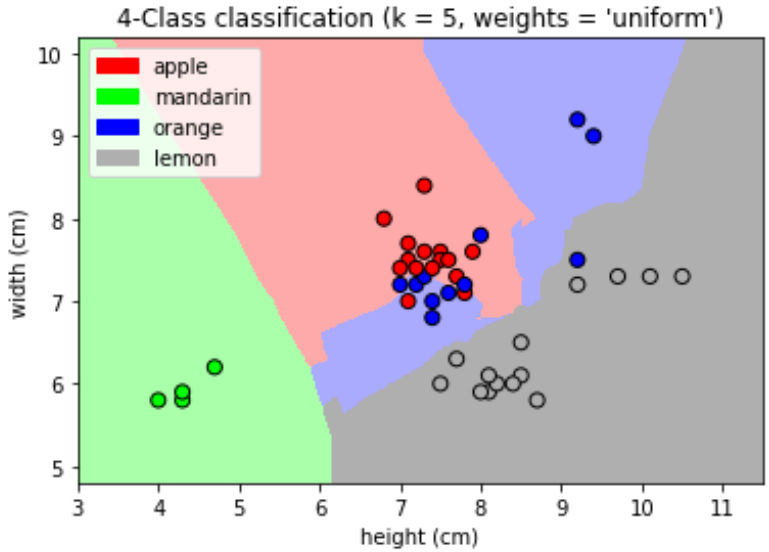
Associated Tasks:	Classification	Missing Values?	No	Number of Web Hits:	1549312


```
data("iris")
str(iris)
```

```
'data.frame':   150 obs. of  5 variables:
 $ Sepal.Length: num  5.1 4.9 4.7 4.6 5 5.4 4.6 5 4.4 4.9 ...
 $ Sepal.Width : num  3.5 3 3.2 3.1 3.6 3.9 3.4 3.4 2.9 3.1 ...
 $ Petal.Length: num  1.4 1.4 1.3 1.5 1.4 1.7 1.4 1.5 1.4 1.5 ...
 $ Petal.Width : num  0.2 0.2 0.2 0.2 0.2 0.4 0.3 0.2 0.2 0.1 ...
 $ Species     : Factor w/ 3 levels "setosa","versicolor",..: 1 1 ...
```

```
library(graphics)
pairs(iris[1:4],
      main = "Anderson's Iris species",
      pch = 20,
      col = c("red", "green3", "blue")
[unclass(iris$Species)])
```





https://towardsdatascience.com/solving-a-simple-classification-problem-with-python-fruits-lovers-edition-d20ab6b071d2

```
fruits <- read.table("Data_mining/datasets/fruits.txt",
header = TRUE)
str(fruits)
```

```
$ fruit_label  : int  1 1 1 2 2 2 2 2 1 1 ...
$ fruit_name   : Factor w/ 4 levels "apple","lemon",..: 1 1 1 3 3 3 3 3 1 1 ...
$ fruit_subtype: Factor w/ 10 levels "braeburn","cripps_pink",..: 4 4 4 5 5 5 5 5 1 1 ...
$ mass         : int  192 180 176 86 84 80 80 76 178 172 ...
$ width        : num  8.4 8 7.4 6.2 6 5.8 5.9 5.8 7.1 7.4 ...
$ height       : num  7.3 6.8 7.2 4.7 4.6 4.3 4.3 4 7.8 7 ...
$ color_score  : num  0.55 0.59 0.6 0.8 0.79 0.77 0.81 0.81 0.92 0.89 ...
```

MIXTO

Reviewed Dataset

Gene expression dataset (Golub et al.)

Molecular Classification of Cancer by Gene Expression Monitoring

Chris Crawford • last updated 4 months ago

Overview | Data | Kernels | Discussion | Activity | Download (1 MB) | New Kernel

Optimization Based Tumor Classification from Microarray Gene Expression Data

Onur Dagliyan[1], Fadime Uney-Yuksektepe[2], I. Halil Kavakli[1], Metin Turkay[3]*

An important use of data obtained from microarray measurements is the classification of tumor types with respect to genes that are either up or down regulated in specific cancer types.

Table 1. Cancer data sets used in this study.

Data set	Samples	Genes	Classes	Reference
Leukemia	72	7129	2	Golub et al. (1999)
Prostate cancer	102	12600	2	Singh et al. (2002)
Prostate outcome	21	12600	2	Singh et al. (2002)

MIXTO

Reviewed Dataset

Gene expression dataset (Golub et al.)

Molecular Classification of Cancer by Gene Expression Monitoring

Chris Crawford • last updated 4 months ago

Overview | Data | Kernels | Discussion | Activity | Download (1 MB) | New Kernel

Optimization Based Tumor Classification from Microarray Gene Expression Data

Onur Dagliyan[1], Fadime Uney-Yuksektepe[2], I. Halil Kavakli[1], Metin Turkay[3]*

```
gene <- read.csv("Data_mining/datasets/gene_trainDF.csv")
str(gene)
```

```
'data.frame':   38 obs. of  7130 variables:
 $ X1   : num  1.1314 1.3258 -2.0812 0.8449 -0.0963 ...
 $ X2   : num  0.459 0.48 -0.332 1.156 0.844 ...
...
 $ X7129: num  -0.16 0.412 -0.26 -1.504 0.139 ...
 $ label: Factor w/ 2 levels "ALL","AML": 1 1 1 1 1 1 1 1 1 1 ...
```

The highest accuracy is obtained with the optimal gene set consisting of 4 genes:

- Myeloperoxidase (M19507-at),
- adipsin (M84526-at),
- CD33 antigen and
- TCF3 transcription factor 3.

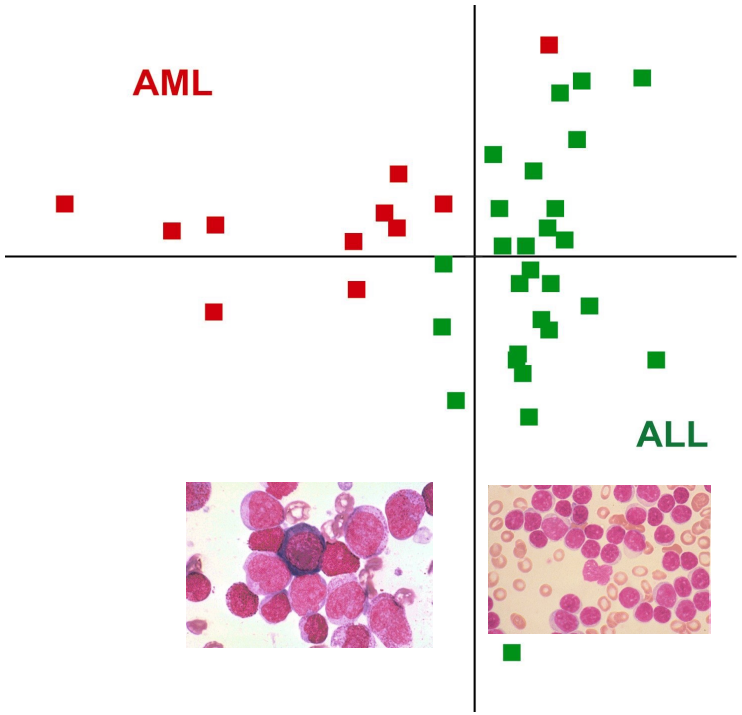
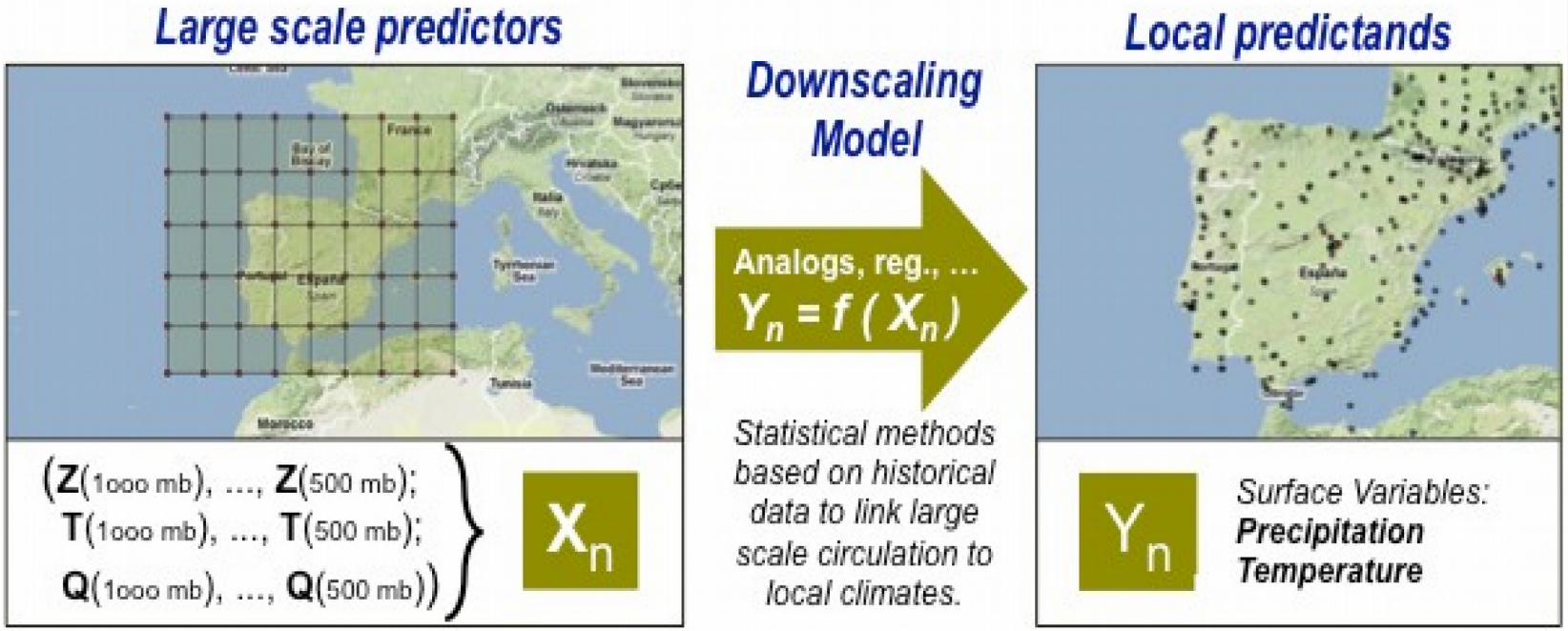


Table 2. Classification results of leukemia (AML-ALL) data set.

Classifier	Test Set	10-CV	LOOCV
HBE	100	97.146 0.903	98.61
BayesNet	94.12	95.71	95.83
LibSVM	58.82	86.576 10.44	91.67
SMO	97.06	93.146 0.571	94.44
Logistic Regression	91.18	96.866 1.67	98.61
RBF Network	97.06	97.43± 1.07	97.22
IBk	97.06	96.006 1.40	95.83
J48	94.12	89.146 1.94	90.28
Random Forest	94.12	93.146 1.07	90.2

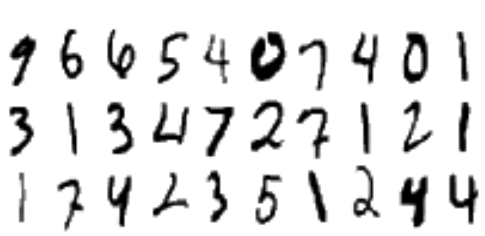


Predictors: Z500,T850,T700,T500,2T,Q850,Q500,SLP
lonLim = (-10,4),
latLim = (36,44),
years = 1979:2008

Predictand: precipitation in Lisboa.
LonLim = -9.15
LatLim = 38.7
years = 1979:2008

```
meteo <- read.csv("Data_mining/datasets/meteo.csv")
str(meteo)
```

```
'data.frame':   10958 obs. of  321 variables:
 $ y   : num  10.9 0.6 13 0 0 1.2 1.1 0 0 0.7 ...
 $ X1  : num  57043 56963 56523 54628 53584 ...
 $ X2  : num  56535 56493 55971 53980 53391 ...
 $ X3  : num  55884 55931 55304 53494 53310 ...
 $ X4  : num  55176 55340 54498 53073 53293 ...
```



Digit Recognizer

Learn computer vision fundamentals with the famous MNIST data

1,996 teams · 2 years to go

MIXTO

Overview Data Kernels Discussion Leaderboard Rules

https://www.kaggle.com/c/digit-recognizer#tutorial

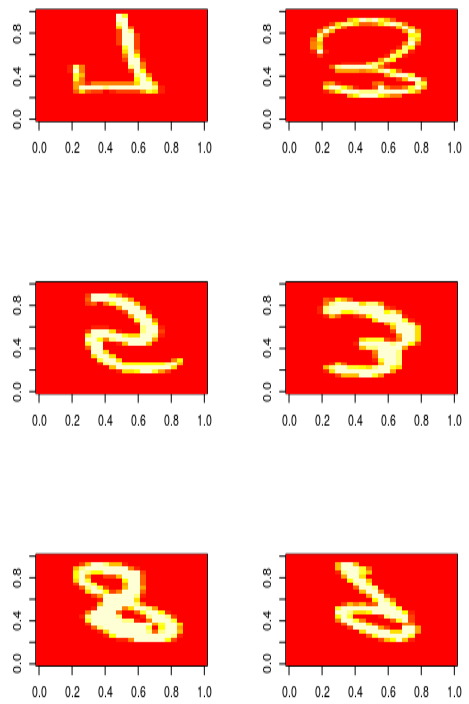
```
mydatadir <- paste0(getwd(), "/MNIST_train.csv")
train <- read.csv(mydatadir)
str(train)
```

```
'data.frame':	42000 obs. of  785 variables:
 $ label  : int  1 0 1 4 0 0 7 3 5 3 ...
 $ pixel0 : int  0 0 0 0 0 0 0 0 0 0 ...
 $ pixel1 : int  0 0 0 0 0 0 0 0 0 0 ...
 $ pixel2 : int  0 0 0 0 0 0 0 0 0 0 ...
 ...
```

```
# split data into response variable (y) and predictors (x)
y <- train[,1]; x <- train[,-1]
dim(x)
```

```
[1] 42000   784
```

```
par(mfrow = c(3,2))
image(matrix(as.matrix(x[7,]), nrow = sqrt(784), ncol = sqrt(784)))
for (i in 8:12) {
      image(matrix(as.matrix(x[i,]), nrow = sqrt(784), ncol = sqrt(784)))
}
```

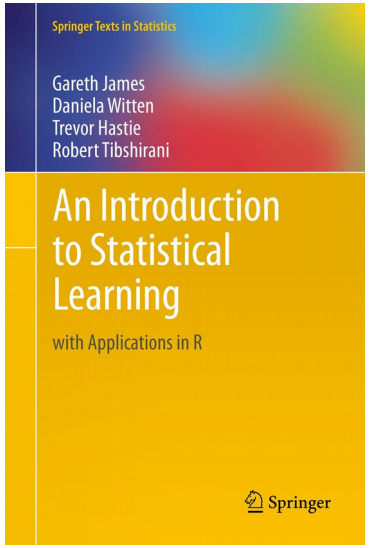


```
Y[7:12]
```

```
[1] 7 3 5 3 8 9
```

Tareas

1 30-60mins



Echa un vistazo a los datasets que hay en el paquete ISLR.

```
install.packages("ISLR")
library("ISLR")
library(help = "ISLR")
```

Analiza la estructura de los datasets: ¿de qué tipo son? ¿para qué tipo de problemas serían adecuados? e.g.

```
data("Hitters")
str(Hitters)
```

2 60-90mins

Lee con calma el siguiente notebook de kaggle sobre las duraciones de los trayectos de taxi en Nueva York:

https://www.kaggle.com/headsortails/nyc-taxi-eda-update-the-fast-the-curious/notebook