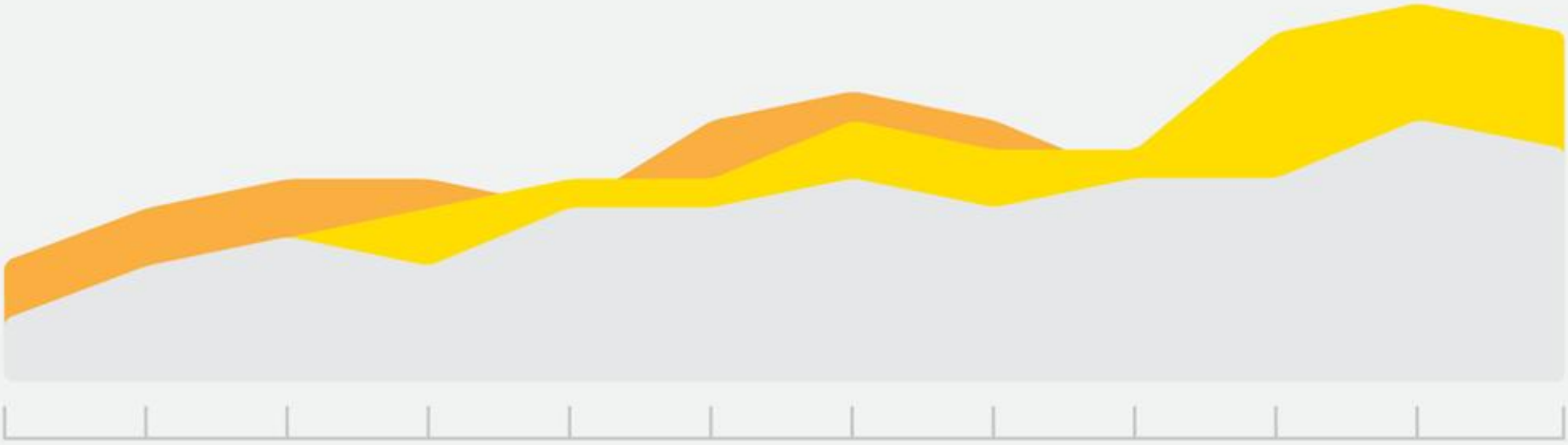
oldemar rodríguez
CONSULTOR en MINERIA DE DATOS

Aprendizaje Supervisado

K - Vecinos más cercanos
Knn-Method



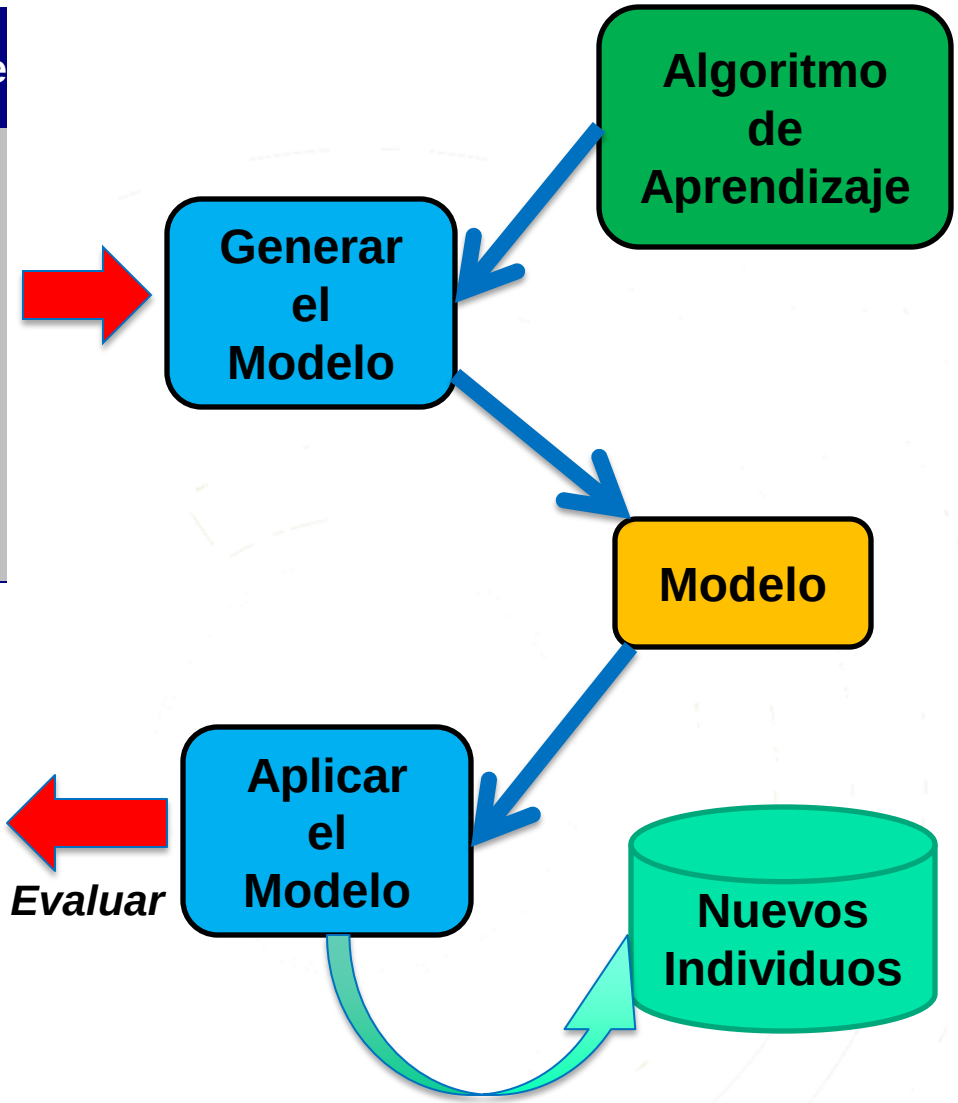
Modelo general de los métodos de Clasificación

Id	Reembolso	Estado Civil	Ingresos Anuales	Fraude
1	Sí	Soltero	125K	No
2	No	Casado	100K	No
3	No	Soltero	70K	No
4	Sí	Casado	120K	No
5	No	Divorciado	95K	Sí
6	No	Casado	60K	No

Tabla de Aprendizaje

Id	Reembolso	Estado Civil	Ingresos Anuales	Fraude
7	No	Soltero	80K	No
8	Si	Casado	100K	No
9	No	Soltero	70K	No

Tabla de Testing



Clasificación: Definición

- Dada una colección de registros (conjunto de entrenamiento) cada registro contiene un conjunto de variables (atributos) denominado x, con un variable (atributo) adicional que es la clase denominada y.

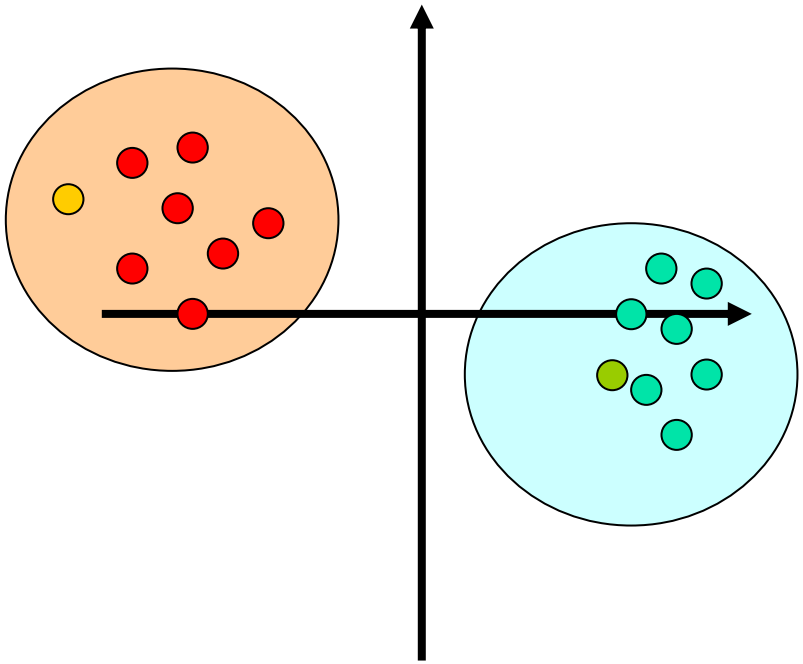
- El objetivo de la ***clasificación*** es encontrar un modelo (una función) para predecir la clase a la que pertenecería cada registro, esta asignación una clase se debe hacer con la mayor precisión posible.

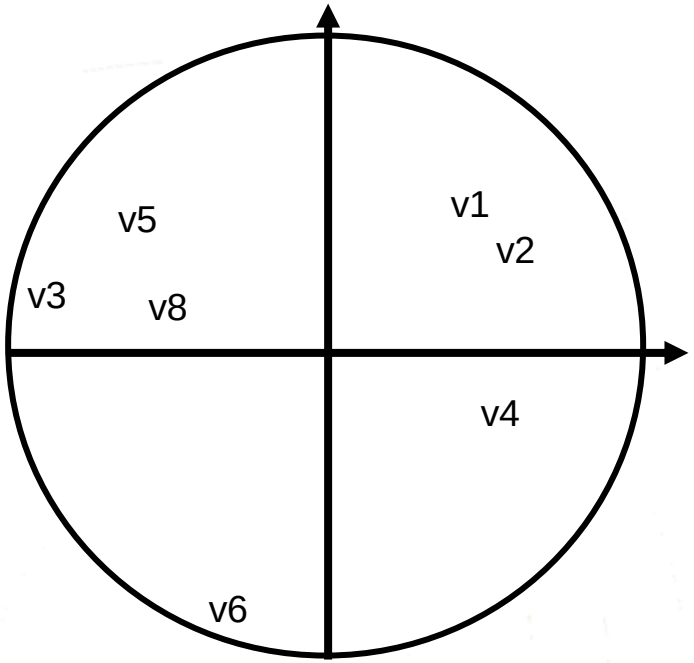
- Un conjunto de prueba (tabla de testing) se utiliza para determinar la precisión del modelo. Por lo general, el conjunto de datos dado se divide en dos conjuntos al azar de el de entrenamiento y el de prueba.

Definición de Clasificación

- Dada una base de datos $D = \{t_1, t_2, \dots, t_n\}$ de tuplas o registros (individuos) y un conjunto de clases $C = \{C_1, C_2, \dots, C_m\}$, el **problema de la clasificación** es encontrar una función $f: D \to C$ tal que cada t_i es asignada una clase C_j.

- $f: D \to C$ podría ser una Red Neuronal, un Árbol de Decisión, un modelo basado en Análisis Discriminante, o una Red Beyesiana.




v5
v1
v2
v3
v8
v4
v6

Ejemplo: Créditos en un Banco

Tabla de Aprendizaje

Variable Discriminante

OLDEMARRR.DMEx...ditoViviendaPeq

	Id	MontoCredito	IngresoNeto	CoeficienteCre...	MontoCuota	GradoAcademico	BuenPagador
▶	1	2	4	3	1	4	1
	2	2	3	2	1	4	1
	3	4	1	1	4	2	2
	4	1	4	3	1	4	1
	5	3	3	1	3	2	2
	6	3	4	3	1	4	1
	7	4	2	1	3	2	2
	8	4	1	3	3	2	2
	9	3	4	3	1	3	1
	10	1	3	2	2	4	1
*	NULL	NULL	NULL	NULL	NULL	NULL	NULL

Con la Tabla de Aprendizaje se entrena (aprende) el modelo matemático de predicción, es decir, a partir de esta tabla se calcula la función ***f*** de la definición anterior.

Ejemplo: Créditos en un Banco
Tabla de Testing

Variable Discriminante

OLDEMARRR.DME...iviendaPeqPRED | OLDEMARRR.DMEx...ditoViviendaPeq

	Id	MontoCredito	IngresoNeto	CoeficienteCre...	MontoCuota	GradoAcademico	BuenPagador
▶	11	3	3	3	3	1	2
	12	2	2	2	2	1	1
	13	2	2	3	2	1	1
	14	1	3	4	3	2	2
	15	1	2	4	2	1	1
*	NULL	NULL	NULL	NULL	NULL	NULL	NULL

- Con la Tabla de Testing se valida el modelo matemático de predicción, es decir, se verifica que los resultados en individuos que no participaron en la construcción del modelo es bueno o aceptable.
- Algunas veces, sobre todo cuando hay pocos datos, se utiliza la Tabla de Aprendizaje también como de Tabla Testing.

Ejemplo: Créditos en un Banco

Nuevos Individuos

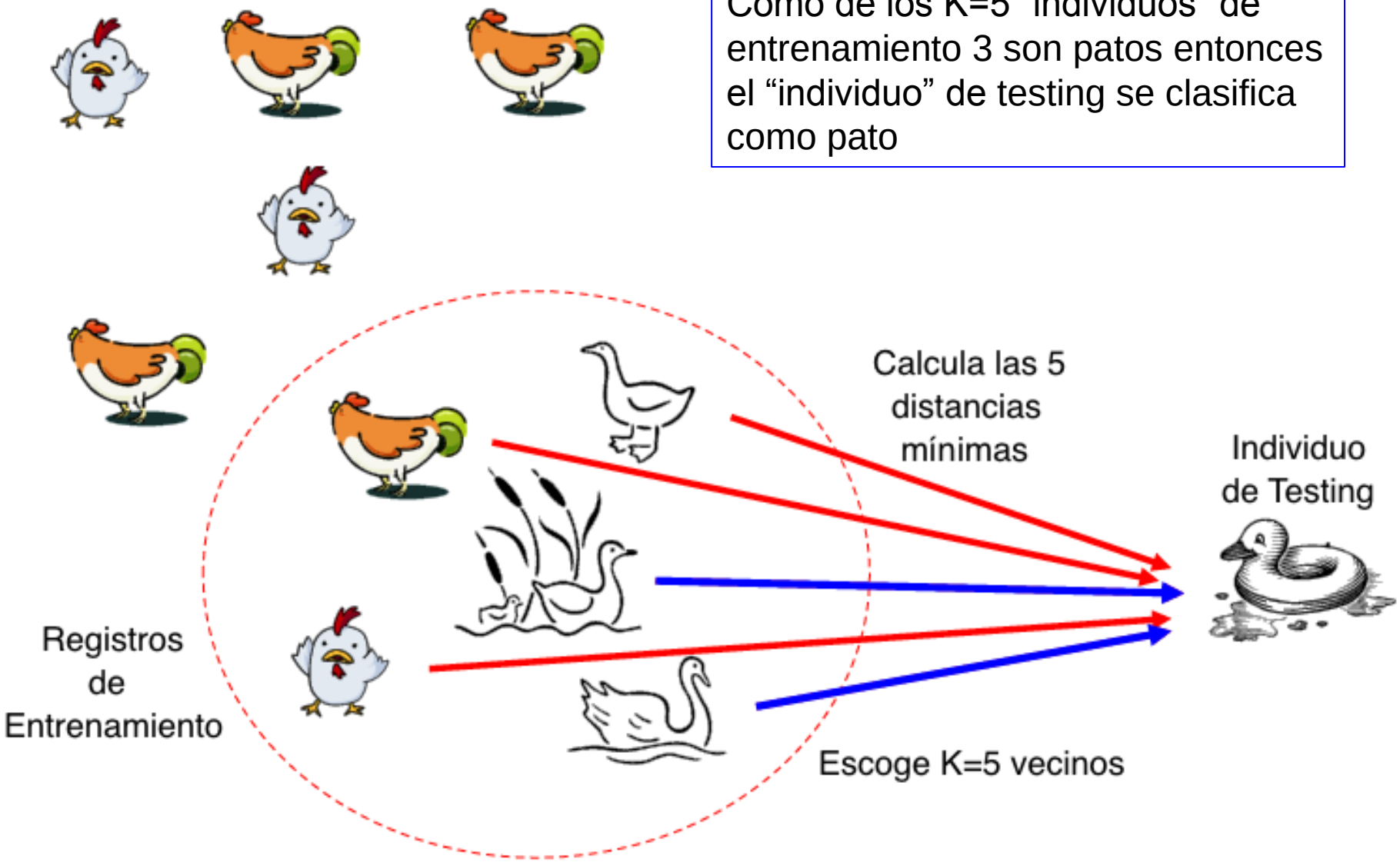
Variable Discriminante

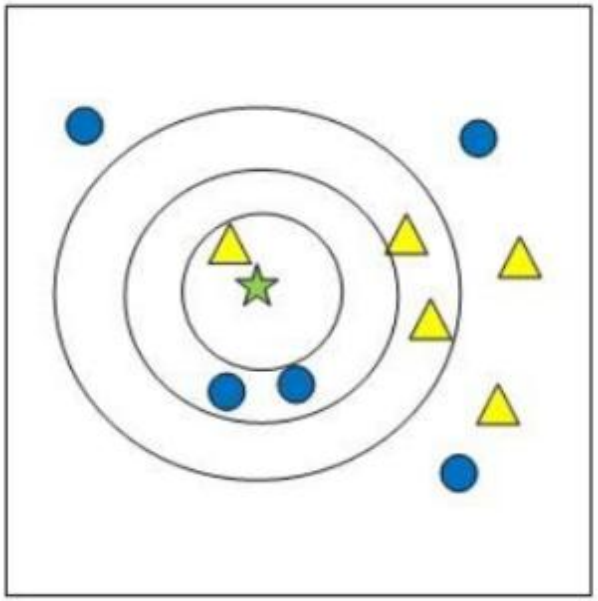
OLDEMARRR.DMEx ...editoViviendaNI

	Id	MontoCredito	IngresoNeto	CoeficienteCre...	MontoCuota	GradoAcademico	BuenPagador
	100	4	4	2	2	3	?
	101	1	4	3	2	4	?
	102	3	2	3	4	2	?
▶*	NULL	NULL	NULL	NULL	NULL	NULL	NULL

Con la Tabla de Nuevos Individuos se predice si estos serán o no buenos pagadores.

Como de los K=5 “individuos” de entrenamiento 3 son patos entonces el “individuo” de testing se clasifica como pato





Para K=1 (círculo más pequeño), la clase de la nueva instancia sería la Clase 1, ya que es la clase de su vecino más cercano, mientras que para K=3 la clase de la nueva instancia sería la Clase 2 pues habrían dos vecinos de la Clase 2 y solo 1 de la Clase 1

Algoritmo

COMIENZO

Entrada: $D = \{(\mathbf{x}_1, c_1), \ldots, (\mathbf{x}_N, c_N)\}$

$\mathbf{x} = (x_1, \ldots, x_n)$ nuevo caso a clasificar

PARA todo objeto ya clasificado (x_i, c_i)

calcular $d_i = d(\mathbf{x}_i, \mathbf{x})$

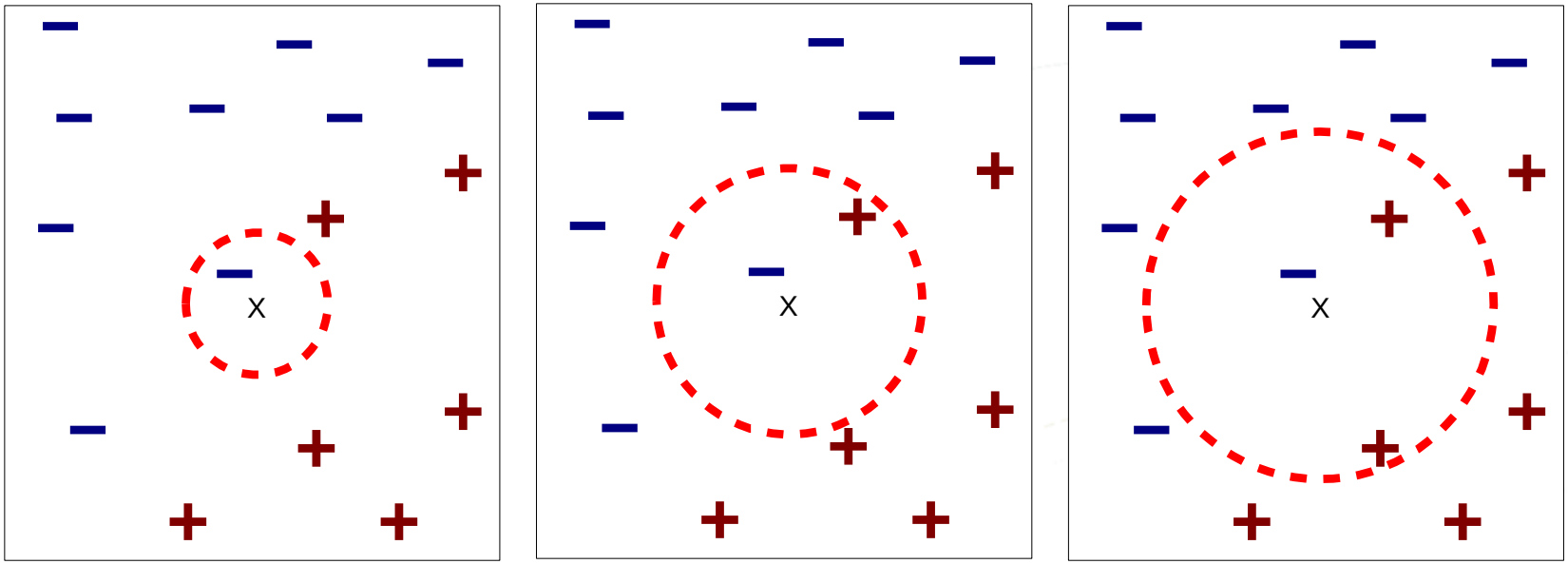
Ordenar $d_i (i = 1, \ldots, N)$ en orden ascendente

Quedarnos con los K casos $D_{\mathbf{x}}^K$ ya clasificados más cercanos a $\mathbf{x}$

Asignar a $\mathbf{x}$ la clase más frecuente en $D_{\mathbf{x}}^K$

FIN

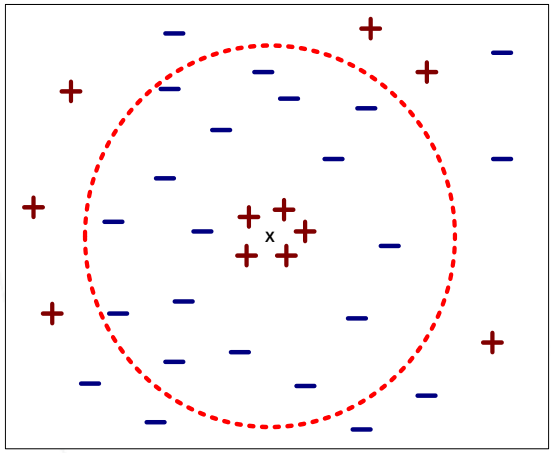
¿Cómo escoger K?



(a) 1-nearest neighbor (b) 2-nearest neighbor (c) 3-nearest neighbor

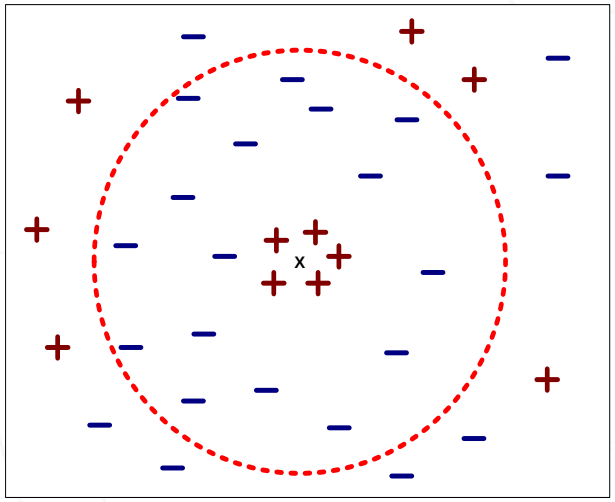
¿Cómo escoger K?

- Escogiendo el valor de K:
  - Si K es muy pequeño el modelo será muy sentitivo a puntos que son atípicos o que son ruido (datos corruptos)
  - Si K es muy grande, el modelo tiende a asignar siempre a la clase más grande.



¿Cómo escoger K?

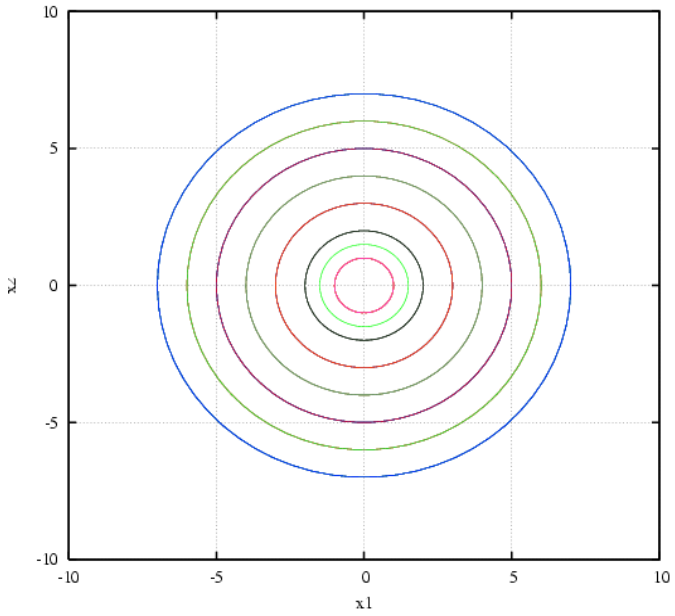
- Escogiendo el valor de K:
  - Mediante la Tabla de Aprendizaje el modelo escogerá el valor de K que mejor clasificación logre en esta tabla, es decir, prueba con K=1, K=2, ….
  - Esto pude ser muy caro computacionalmente.



¿Cómo escoger la distancia?

$$d(A,B) \equiv \sqrt{\sum_{i=1}^{n} (A_i - B_i)^2} = \sqrt{(A-B)^T (A-B)}$$

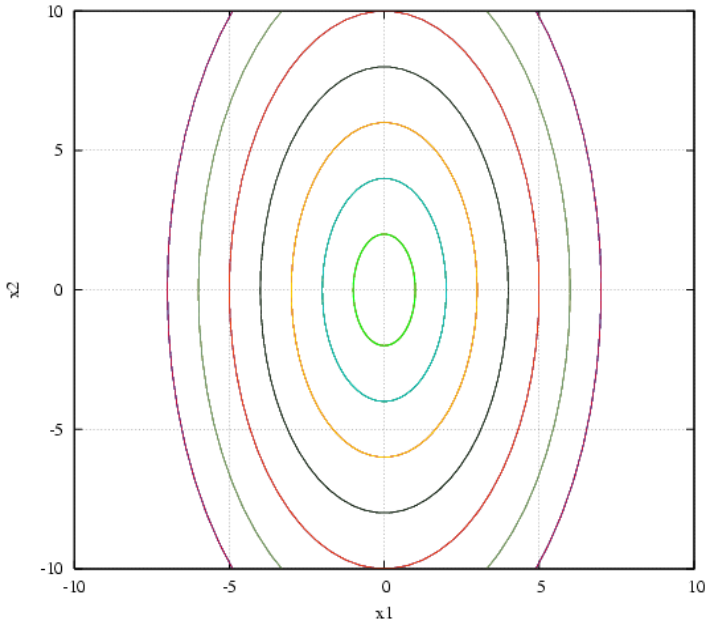
Distancia Euclídea



¿Cómo escoger la distancia?

$$d(A,B) \equiv \sqrt{(A-B)^T M^T M (A-B)}$$

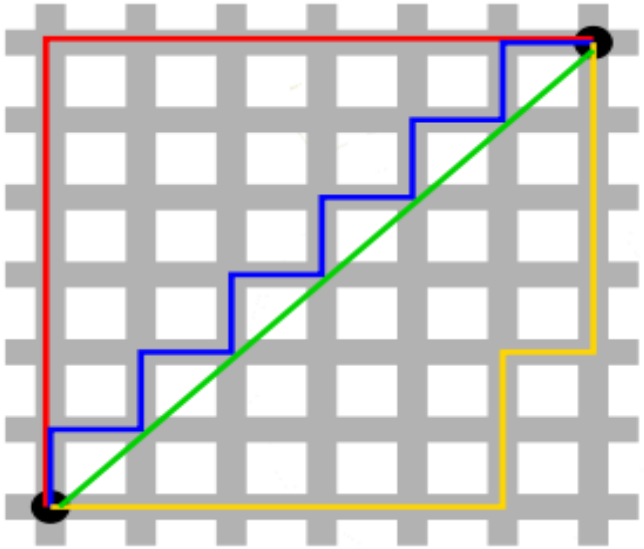
Distancia Euclídea Ponderada



¿Cómo escoger la distancia?

$$d(x, y) = \sum_{i=1}^{n} |x_i - y_i|$$

Distancia Manhattan (city-block)



Matriz de confusión

- La ***Matriz de Confusión*** contiene información acerca de las predicciones realizadas por un ***Método o Sistema de Clasificación***, comparando para el conjunto de individuos en de la tabla de aprendizaje o de testing, la predicción dada versus la clase a la que estos realmente pertenecen.
- La siguiente tabla muestra la matriz de confusión para un clasificador de dos clases:

		Predicción	
		Negativo	Positivo
Valor Real	Negativo	a	b
	Positivo	c	d

Ejemplo: Matriz de confusión

		Predicción	
		Mal Pagador	Buen Pagador
Valor Real	Mal Pagador	800	200
	Buen Pagador	500	1500

- 800 predicciones de Mal Pagador fueron realizadas correctamente, para un 80%, mientras que 200 no, para un 20%.
- 1500 predicciones de Buen Pagador fueron realizadas correctamente, para un 75%, mientras que 500 no (para un 25%).
- En general 2300 de 3000 predicciones fueron correctas para un 76,6% de efectividad en las predicciones. ***Cuidado***, este dato es a veces engañoso y debe ser siempre analizado en la relación a la dimensión de las clases.

Matriz de confusión

		Predicción	
		Negativo	Positivo
Valor Real	Negativo	a	b
	Positivo	c	d

- La Precisión ***P*** de un modelo de predicción es la proporción del número total de predicciones que son correctas respecto al total. Se determina utilizando la ecuación: ***P = (a+d)/(a+b+c+d)***

- ***Cuidado***, este índice es a veces engañoso y debe ser siempre analizado en la relación a la dimensión de las clases.

Ejemplo: Matriz de confusión

		Predicción	
		Fraude	No Fraude
Valor Real	Fraude	0	8
	No Fraude	3	989

- ***Cuidado***, este índice es a veces engañoso y debe ser siempre analizado en la relación a la dimensión de las clases.

- En la Matriz de Confusión anterior la Precisión ***P*** es del 98,9%, sin embargo, el modelo no detectó ningún fraude.

Matriz de confusión

		Predicción	
		Negativo	Positivo
Valor Real	Negativo	a	b
	Positivo	c	d

- La Precisión Positiva (***PP***) es la proporción de casos positivos que fueron identificados correctamente, tal como se calcula usando la ecuación: ***PP = d/(c+d)***

- En el ejemplo anterior Precisión Positiva ***PP*** es del 99,6% .

Matriz de confusión

		Predicción	
		Negativo	Positivo
Valor Real	Negativo	a	b
	Positivo	c	d

- La Precisión Negativa (***PN***) es la proporción de casos negativos que fueron identificados correctamente, tal como se calcula usando la ecuación: ***PN = a/(a+b)***

- En el ejemplo anterior Precisión Negativa ***PN*** es del 0% .

Matriz de confusión

		Predicción	
		Negativo	Positivo
Valor Real	Negativo	a	b
	Positivo	c	d

- Falsos Positivos (***FP***) es la proporción de casos negativos que fueron clasificados incorrectamente como positivos, tal como se calcula utilizando la ecuación: ***FP = b/(a+b)***

- Falsos Negativos (***FN***) es la proporción de casos positivos que fueron clasificados incorrectamente como negativos, tal como se calcula utilizando la ecuación: ***FN = c/(c+d)***

Matriz de confusión

		Predicción	
		Negativo	Positivo
Valor Real	Negativo	a	b
	Positivo	c	d

- Acertividad Positiva (***AP***) indica la proporción de buena predicción para los positivos, tal como se calcula utilizando la ecuación: ***FN = d/(b+d)***
- Acertividad Negativa (***AN***) indica la proporción de buena predicción para los negativos, tal como se calcula utilizando la ecuación: ***FN = a/(a+c)***

Matriz de confusión para más de 2 clases

- La Matriz de Confusión puede calcularse en general para un problema con ***p*** clases.
- En la matriz ejemplo que aparece a continuación, de 8 alajuelenses reales, el sistema predijo que 3 eran heredianos y de 6 heredianos predijo que 1 era un limonense y 2 eran alajuelenses. A partir de la matriz se puede ver que el sistema tiene problemas distinguiendo entre alajuelenses y heredianos, pero que puede distinguir razonablemente bien entre limonenses y las otras provincias.

		Predicción		
		Alajuelense	Herediano	Limonense
Valor Real	alajuelense	5	3	0
	Herediano	2	3	1
	Limonense	0	2	11

K - Vecinos más cercano en R

Package 'kknn'

October 30, 2012

Title Weighted k-Nearest Neighbors

Version 1.2-1

Date 2012-30-10

Author Klaus Schliep & Klaus Hechenbichler

Description Weighted k-Nearest Neighbors Classification, Regression and Clustering

Maintainer Klaus Schliep <klaus.schliep@gmail.com>

Depends R (>= 2.10), igraph (>= 0.6)

Imports Matrix, stats

License GPL (>= 2)

Repository CRAN

Date/Publication 2012-10-30 12:23:58

Ejemplo con IRIS.CSV

Ejemplo con la tabla de datos IRIS
IRIS Información de variables:
1.sepal largo en cm
2.sepal ancho en cm
3.petal largo en cm
4.petal ancho en cm
5.clase:

- Iris Setosa
- Iris Versicolor
- Iris Virginica

	A	B	C	D	E
1	s.largo	s.ancho	p.largo	p.ancho	tipo
2	5.1	3.5	1.4	0.2	setosa
3	4.9	3.0	1.4	0.2	setosa
4	4.7	3.2	1.3	0.2	setosa
5	4.6	3.1	1.5	0.2	setosa
6	5.0	3.6	1.4	0.2	setosa
7	5.4	3.9	1.7	0.4	setosa
8	4.6	3.4	1.4	0.3	setosa
9	5.0	3.4	1.5	0.2	setosa
10	4.4	2.9	1.4	0.2	setosa
11	4.9	3.1	1.5	0.1	setosa
12	5.4	3.7	1.5	0.2	setosa
13	4.8	3.4	1.6	0.2	setosa
14	4.8	3.0	1.4	0.1	setosa
15	4.3	3.0	1.1	0.1	setosa
16	5.8	4.0	1.2	0.2	setosa
17	5.7	4.4	1.5	0.4	setosa
18	5.4	3.9	1.3	0.4	setosa
19	5.1	3.5	1.4	0.3	setosa
20	5.7	3.8	1.7	0.3	setosa
21	5.1	3.8	1.5	0.3	setosa
22	5.4	3.4	1.7	0.2	setosa
23	5.1	3.7	1.5	0.4	setosa
24	4.6	3.6	1.0	0.2	setosa
25	....	...	...	...	....

Ejemplo con IRIS.CSV en R

```
rm(list=ls(all=TRUE)) # BORRA TODAS LAS VARIABLES DE MEMORIA
datos=read.csv("iris.csv",sep = ";",dec='.',header=T)
install.packages('kknn')
library(kknn)
muestra = sample(1:150,50)
ttesting = datos[muestra,]
taprendizaje = datos[-muestra,]
modelo=train.kknn(taprendizaje$tipo~.,data=taprendizaje,K=8)
prediccion=predict(modelo,ttesting[,-5])
## Matriz de Confusion
table(ttesting[,5],prediccion)
# Procentaje de error y de buena clasificacion
error = sum(prediccion != ttesting$tipo) / nrow(ttesting)
error
acierto=(1-error)*100
acierto
```

Ejemplo con IRIS.CSV en R

```
> table(ttesting[,5],prediccion)
            prediccion
             setosa versicolor virginica
  setosa         18          0         0
  versicolor      0         20         1
  virginica       0          0        11

> # Porcentaje de error y de buena clasificacion
>> error
[1] 0.02

> acierto
[1] 98
```

Ejemplo 2: Credit-Scoring

MuestraAprendizajeCredito2500.csv
MuestraTestCredito2500.csv

```
> setwd("C:/Users/Oldemar/Google Drive/Curso Mineria Datos II - Optativo/Datos")
> taprendizaje<-read.csv("MuestraAprendizajeCredito2500.csv",sep = ";",header=T)
> taprendizaje
      MontoCredito IngresoNeto CoefCreditoAvaluo MontoCuota GradoAcademico BuenPagador
1                1           1                 1          1              1          Si
2                3           1                 1          1              1          Si
3                2           1                 1          1              1          Si
4                1           2                 1          1              1          Si
5                1           1                 1          1              1          Si
6                2           1                 1          1              1          Si
7                4           1                 1          1              1          Si
8                1           2                 1          1              1          Si
9                1           2                 1          1              1          Si
10               3           2                 1          1              1          Si
11               1           1                 1          1              1          Si
12               1           2                 1          1              1          Si
13               3           1                 1          1              1          Si
14               3           1                 1          1              1          Si
15               2           1                 1          1              1          Si
16               3           1                 1          1              1          Si
17               3           1                 1          1              1          Si
```

Descripción de Variables

MontoCredito

1=Muy Bajo
2=Bajo
3=Medio
4=Alto

MontoCuota

1=Muy Bajo
2=Bajo
3=Medio
4=Alto

IngresoNeto

1=Muy Bajo
2=Bajo
3=Medio
4=Alto

GradoAcademico

1=Bachiller
2=Licenciatura
3=Maestría
4=Doctorado

CoeficienteCreditoAvaluo

1=Muy Bajo
2=Bajo
3=Medio
4=Alto

BuenPagador

1=NO
2=Si

K - Vecinos más cercano en R

```
rm(list=ls(all=TRUE))
taprendizaje<-
read.csv("MuestraAprendizajeCredito2500.csv",sep =
";",header=T)
## Usamos una nueva tabla de testing para validar
ttesting<-read.csv("MuestraTestCredito2500.csv",sep
= ";",header=T)
modelo=train.kknn(taprendizaje$BuenPagador~.,data=ta
prendizaje,K=6)
prediccion=predict(modelo,ttesting[,-6])
## Matriz de Confusion
table(ttesting[,6],prediccion)
# Procentaje de error y de buena clasificacion
error = sum(prediccion != ttesting$BuenPagador) /
nrow(ttesting)
acierto=(1-error)*100
```

Calidad de la Predicción

- Matriz de Confusión:

	No	***Si***
No	***232***	***113***
Si	***33***	***2122***

- Precisión Global
  - ***P=0.94 (En Árboles P=0.95) (en Bayes fue de 0.83)***

- Precisión en cada variable:
  - ***P(No)= 0.67 (en Bayes fue 0.12)***
  - ***P(Sí)=0.98 (en Bayes fue 0.95)***

Laboratorio

- Construya en RStudio un modelo predictivo para la variable “party” en el archivo ***us2011votes.csv***
  - Genere una tabla de 2/3 de las filas para aprendizaje y el resto para testing. (OJO Total de filas = 426)
  - Genere el modelo (use K=5)
  - Genere una predicción usando la Tabla de Testing
  - Calcule la matriz de confusión y los porcentajes de acierto.

Gracias….

oldemar rodríguez

CONSULTOR en M1NƐR14 DƐ D4T0S