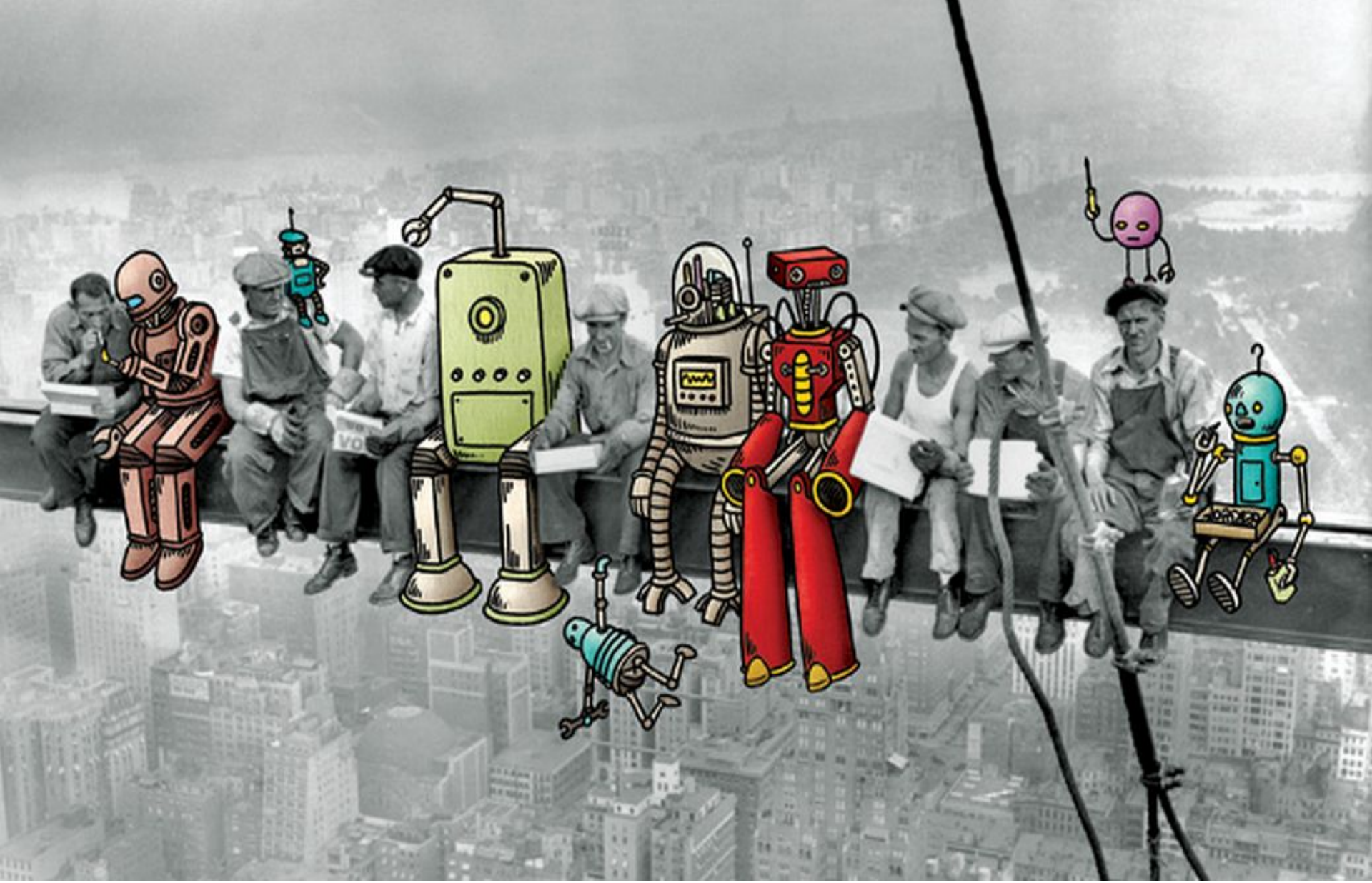


2. Algoritmos Aprendizaje Supervisado

DA07 - Machine Learning (I)

DA - Data Advanced - Data Analytics Journey

DA07 - Machine Learning (I)

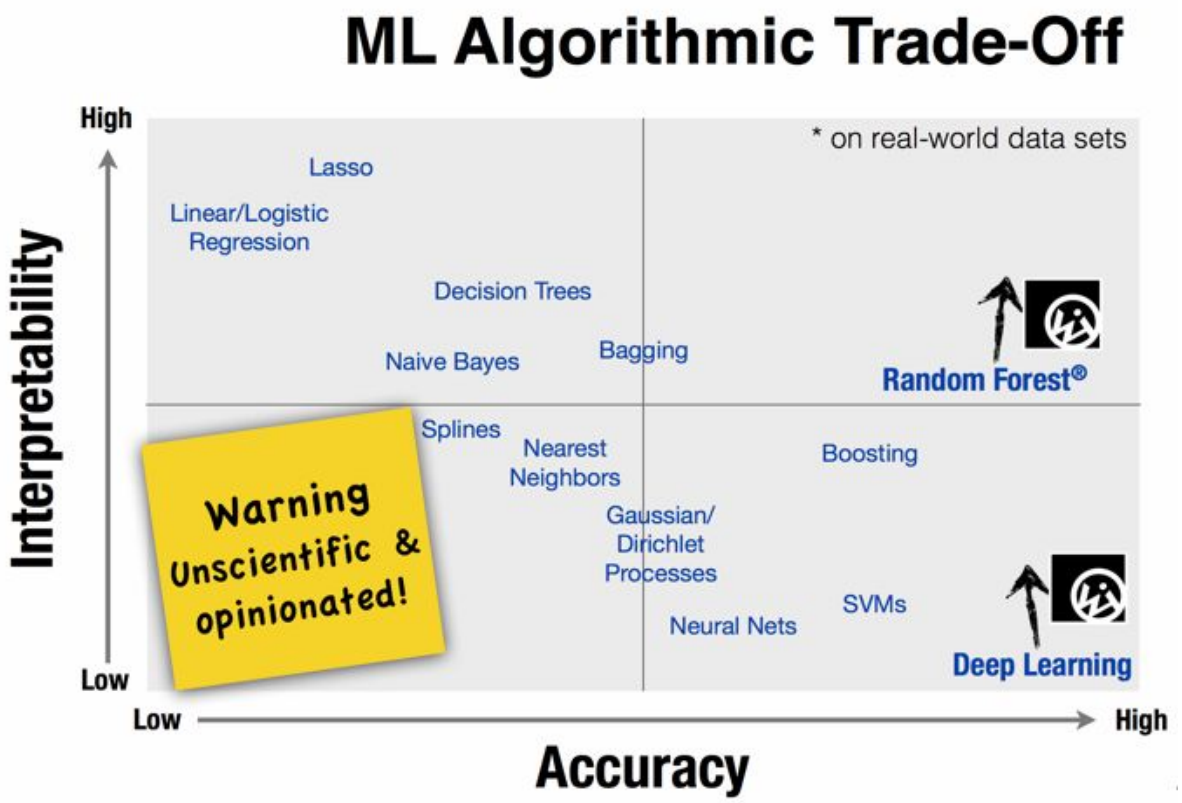
2. Algoritmos Aprendizaje Supervisado

- **Objetivos de aprendizaje**
  - Identificar las diferentes ventajas y desventajas que tiene cada algoritmo
  - Conocer y aplicar cada algoritmo para las características de los datos más apropiados para sus ventajas

2. Algoritmos Aprendizaje Supervisado

2. Algoritmos Aprendizaje Supervisado
Introducción

- Se pueden desarrollar modelos supervisados (clasificación y regresión) con diferentes algoritmos
  - Deberemos elegir en función de qué queramos priorizar y nuestro trade-off

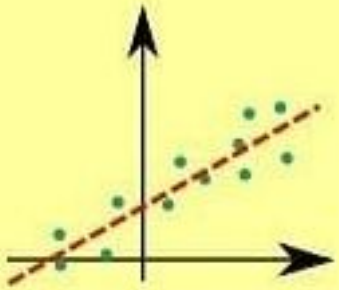


2. Algoritmos Aprendizaje Supervisado

Regresión

LINEAR REGRESSION

1. Econometric modelling
2. Marketing Mix Model
3. Customer Lifetime Value



Continuous ⇒ Continuous

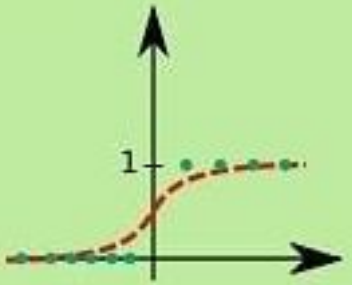
$$y = \alpha_0 + \sum_{i=1}^{N} \alpha_i x_i$$

lm(y ~ x1 + x2, data)

1 unit increase in x increases y by α

LOGISTIC REGRESSION

1. Customer Choice Model
2. Click-through Rate
3. Conversion Rate
4. Credit Scoring



Continuous ⇒ True/False

$$y = \frac{1}{1 + e^{-z}}$$

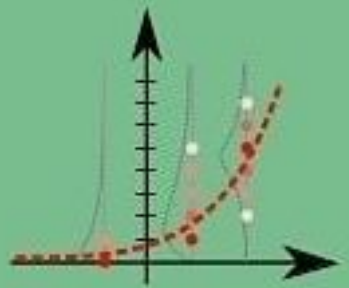
$$z = \alpha_0 + \sum_{i=1}^{N} \alpha_i x_i$$

glm(y ~ x1 + x2, data, family=binomial( ))

1 unit increase in x increases log odds by α

POISSON REGRESSION

1. Number of orders in lifetime
2. Number of visits per user



Continuous ⇒ 0,1,2,...

$$y \sim Poisson(\lambda)$$

$$ln\lambda = \alpha_0 + \sum_{i=1}^{N} \alpha_i x_i$$

glm(y ~ x1 + x2, data, family=poisson( ))

1 unit increase in x multiplies y by e^{α}

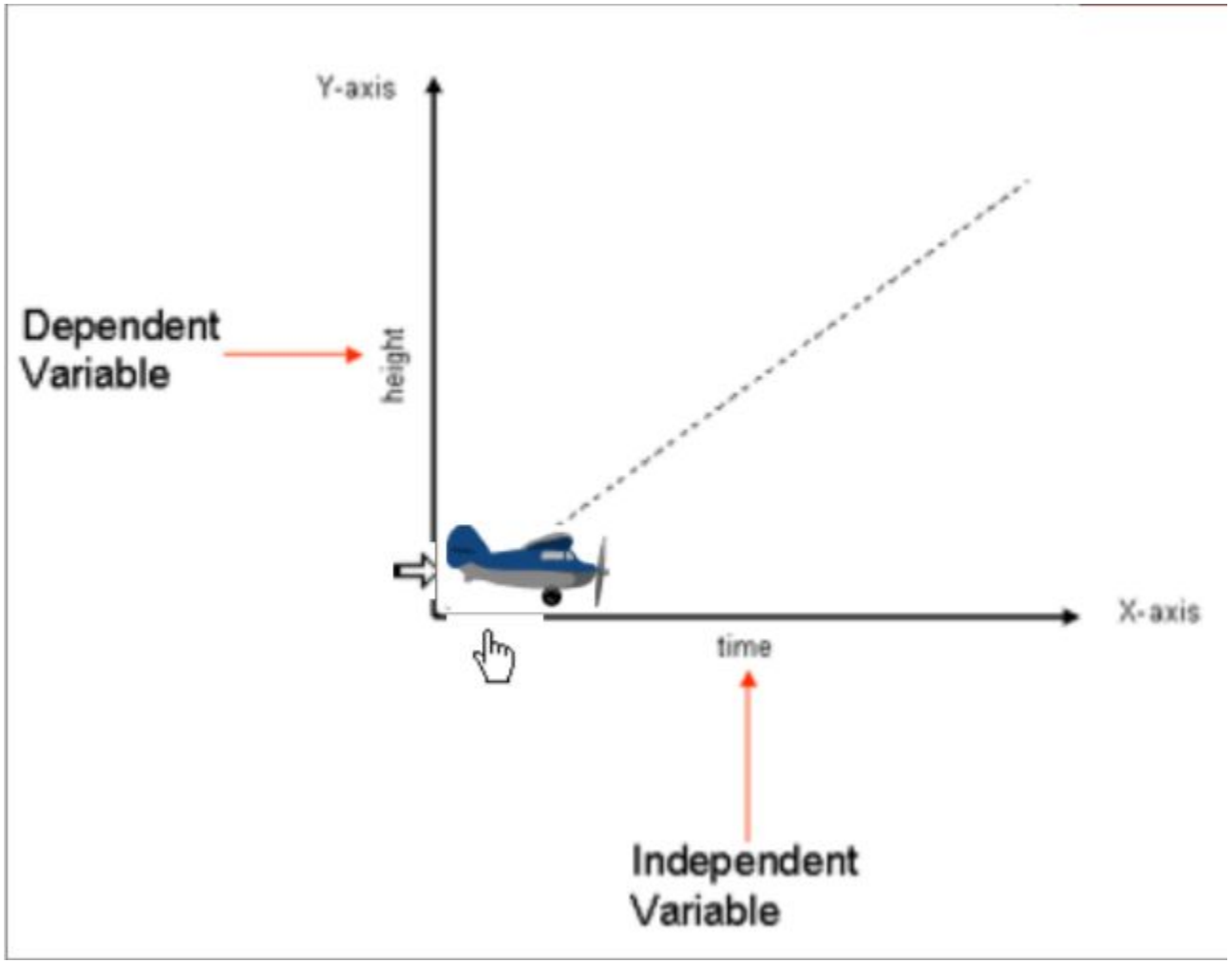
2. Algoritmos Aprendizaje Supervisado

Regresión - Regresión lineal

- Para poder crear un modelo de regresión lineal, es necesario que se cumpla con los siguientes supuestos:
  - La relación entre las variables es lineal.
  - Los errores son independientes.
  - Los errores tienen varianza constante.
  - Los errores tienen una esperanza matemática igual a cero.
  - El error total es la suma de todos los errores.
- Hay dos tipos de variables:
  - *Variables dependientes*: Son las variables de respuesta que se observan en el estudio y que podrían estar influidas por los valores de las variables independientes.
  - *Variables independientes*: Son las que se toman para establecer agrupaciones en el estudio, clasificando intrínsecamente a los casos del mismo.

2. Algoritmos Aprendizaje Supervisado

Regresión - Regresión lineal (II)



2. Algoritmos Aprendizaje Supervisado

Regresión - Regresión lineal (III)

- Construir un modelo o fórmula que predice la variable respuesta a través de las variables explicativas.
- Podemos suponer que la relación sea lineal

$$Y = a_1 X_1 + \cdots + a_m X_m$$

- O podemos suponer que la relación sea a través de transformaciones no lineales de las variables explicativas o incluso de la respuesta

$$Y = a_1 \phi(X_1) + \cdots + a_m \phi(X_m)$$

- La relación sigue siendo lineal. ¿Cómo se interpretan los parámetros?

2. Algoritmos Aprendizaje Supervisado

Regresión - Regresión lineal (II)

- No hemos introducido ninguna restricción en los parámetros
  - Por tanto pueden entran muchas variables en el modelo, aunque sea con efectos pequeños
- Si queremos forzar a que no entren variables en el modelo, podemos intentar forzar que los parámetros sea 0 o cercanos a él
  - Es lo que se llaman modelos de regresión penalizada
- Fácil de calcular computacionalmente los parámetros
- Más fácil de interpretar que otros modelos más sofisticados
- Un modelo restringido puede funcionar mejor cuando hay "mucho ruido"

2. Algoritmos Aprendizaje Supervisado

Regresión - Regresión lineal (III)

- Queremos explicar el empleo en función del PIB

$$Y = \beta_0 + \sum_{j=1}^{p} \beta_j X_j + \epsilon$$

$$\text{Empleo} = \beta_0 + \beta_1 \quad \text{PIB} \quad + \epsilon$$

```
Console ~/
> m1 <- lm(formula=  Employed~GNP, data=datos)
> summary(m1)

Call:
lm(formula = Employed ~ GNP, data = datos)

Residuals:
     Min       1Q   Median       3Q      Max
-0.77958 -0.55440 -0.00944  0.34361  1.44594

Coefficients:
             Estimate Std. Error t value Pr(>|t|)
(Intercept) 51.843590   0.681372   76.09  < 2e-16 ***
GNP          0.034752   0.001706   20.37 8.36e-12 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 0.6566 on 14 degrees of freedom
Multiple R-squared:  0.9674, Adjusted R-squared:  0.965
F-statistic: 415.1 on 1 and 14 DF,  p-value: 8.363e-12
```

2. Algoritmos Aprendizaje Supervisado

Regresión - Regresión lineal (IV)

- Información en el resumen del modelo
  - **Formula**: variable respuesta ~ variable(s) explicativas
  - **Residuals**: cuantiles de los errores cometidos por el modelo
  - **Coefficients**: parámetros estimados
  - **Estimate**: valor estimado del parámetro
  - **std err**: error estándar cometido en la estimación del
  - **t**: el valor del t-statistic. Medida de cómo de estadísticamente significativo es el coeficiente.
  - **P > |t|:** P-value de la hipótesis nula de que el coeficiente =0 es verdad.Si es menor que el nivel de confianza, a menudo 0.05, indica que hay una relación significativa entre esa variable y la variable respuesta.

2. Algoritmos Aprendizaje Supervisado

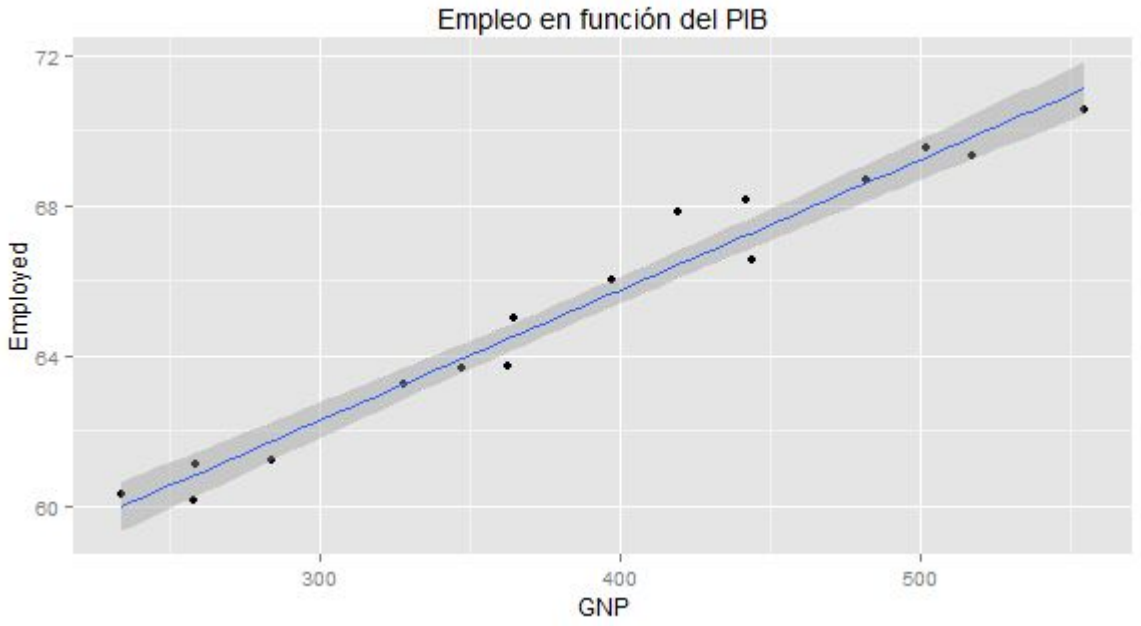
Regresión - Regresión lineal (V)

- **R-squared**: Medida estadística de cómo la regresión se aproxima a los datos. Muy optimista, crece con el número de variables.
- **Adj. R-squared**: Valor anterior ajustado por el número de observaciones y los grados de libertad de los residuos. No siempre crece con el número de variables.
- **F-statistic**: Medida estadística de cómo de significativo es el ajuste.
- **Prob (F-statistic)**: P-value de la hipótesis nula de que todos los coeficientes =0 es verdad.
- **Interpretación del modelo**

```
> coef(m1)
(Intercept)         GNP
51.84358978  0.03475229
```

2. Algoritmos Aprendizaje Supervisado

Regresión - Regresión lineal (VI)



Varios componentes del gráfico:

- Gráfico de dispersión
- Modelo ajustado
- Intervalo de confianza del 95%

2. Algoritmos Aprendizaje Supervisado

Regresión - Regresión lineal (VII)

- **Multicolinealidad y soluciones**
  - En muchas ocasiones existen relaciones lineales (exactas o no) entre los regresores.
  - Ejemplo: peso y tamaño de un coche para predecir el consumo de combustible de un coche.
    - ¿Cómo afecta a la estimación de los modelos?
    - Los coeficientes aparecen con el signo cambiado
    - Grandes varianzas en los estimadores
    - Los contrastes de signficatividad no son válidos
  - Detección de la multicolinealidad:
    - Analizar visualmente las relaciones entre las variables explicativas
    - Analizar la matriz de correlaciones
  - Eliminación de variables (justificada)
    - Regresión en cresta (ridge regression) Se encogen los coeficientes para que sean cercanos a 0, no exactamente 0. En R lm.ridge
    - ¿Cómo? Penalizando por la suma de los cuadrados de los coeficientes
    - Si sólo penalizamos por la norma, criterio Lasso
    - Regresión en componentes principales

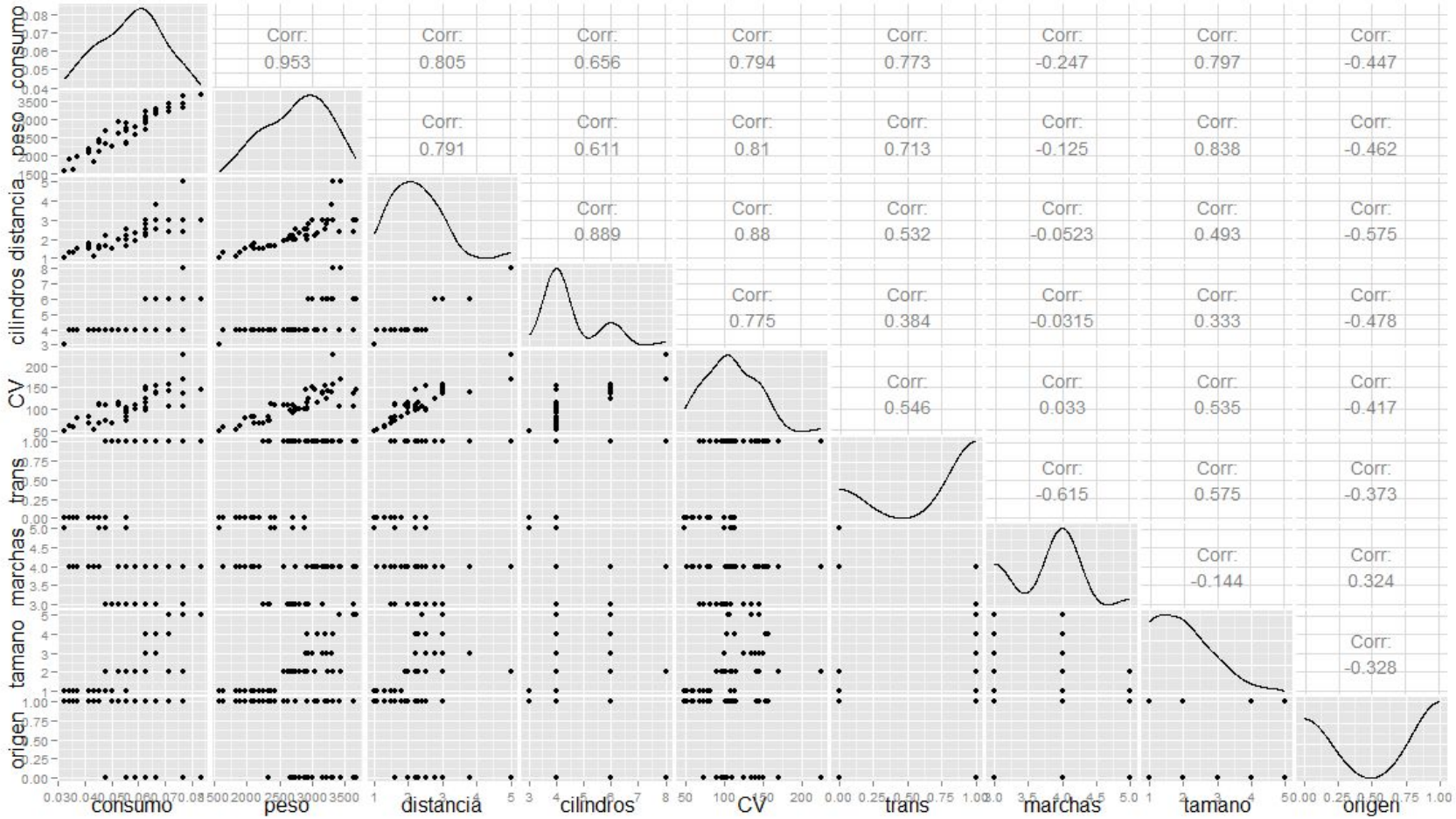
2. Algoritmos Aprendizaje Supervisado

Regresión - Regresión lineal (VIII)

- El caso:
  - Tenemos unos datos de 45 coches de USA para los que queremos predecir el consumo de combustible en función del:
    - Peso
    - Distancia recorrida
    - Caballos de potencia
    - Tipo de transmisión (1=auto, 0=manual)
    - Número de marchas
    - Tamaño (1=pequeño, 2=compacto, 3=medio, 4=wagon, 5=minivan)
    - Origen (1=extranjero, 0=nacional)
- **Preguntas:** ¿cómo pensáis que afectan estas variables al consumo?

2. Algoritmos Aprendizaje Supervisado

Regresión - Regresión lineal (IX)



Santander

2. Algoritmos Aprendizaje Supervisado

Regresión - Regresión lineal (X)

- Para ver si es necesario transformar las variables predictoras, se observan las relaciones de estas con la variable respuesta de manera gráfica. **¿Creéis que hay que transformar alguna?**
- Dentro de las variables variables predictoras tenemos varias variables categóricas. **¿Cuáles son?**
- Recordemos que dentro de los supuestos del modelo de regresión, hemos supuesto que las variables predictoras no estaban relacionadas entre sí. **¿Estáis de acuerdo con ese supuesto?**

2. Algoritmos Aprendizaje Supervisado

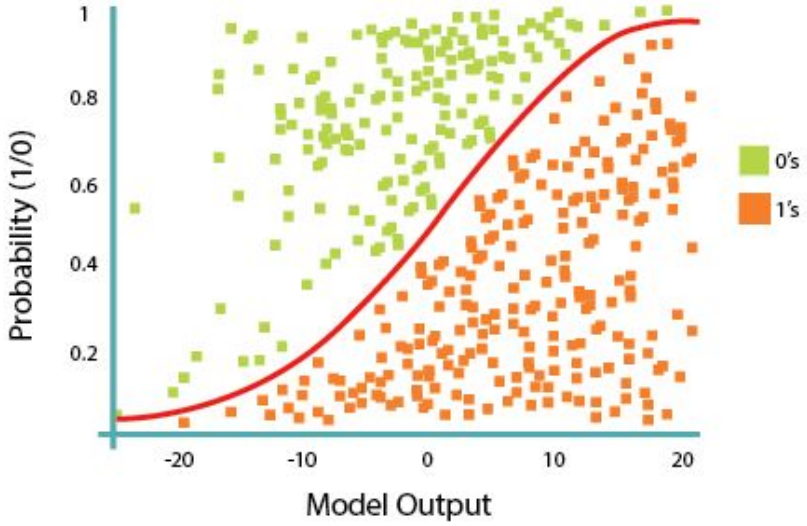
Regresión - Regresión logística

- Uno de los principales problemas en la clasificación ocurre cuando el algoritmo nunca converge en la actualización de los pesos mientras está siendo entrenado.
- Esto ocurre cuando las clases no son perfectamente separables linealmente
  - Por tanto, para tratar con problemas de clasificación binaria la regresión logística es uno de los algoritmos más usados.
- La regresión logística es un algoritmo de clasificación (a pesar de su nombre) simple, pero potente
  - Funciona muy bien en clases linealmente separables y se puede extender a clasificación multiclase

2. Algoritmos Aprendizaje Supervisado

Regresión - Regresión logística (II)

- Por lo tanto, se aplica cuando la variable dependiente es dicotómica o politómica y no numérica
- Para poder aplicar una regresión se asocia la variable dependiente a su probabilidad de ocurrencia.
- Por lo tanto el resultado de un regresión logística es la probabilidad de ocurrencia del suceso



2. Algoritmos Aprendizaje Supervisado

Regresión - Regresión logística (II)

- La proporción de probabilidades es un concepto importante para entender la idea que subyace tras la regresión logística.
- Es la probabilidad de que ocurra un cierto evento
- Se puede escribir como:

$$\text{Proporción de Probabilidades} = \frac{P}{1 - P}$$

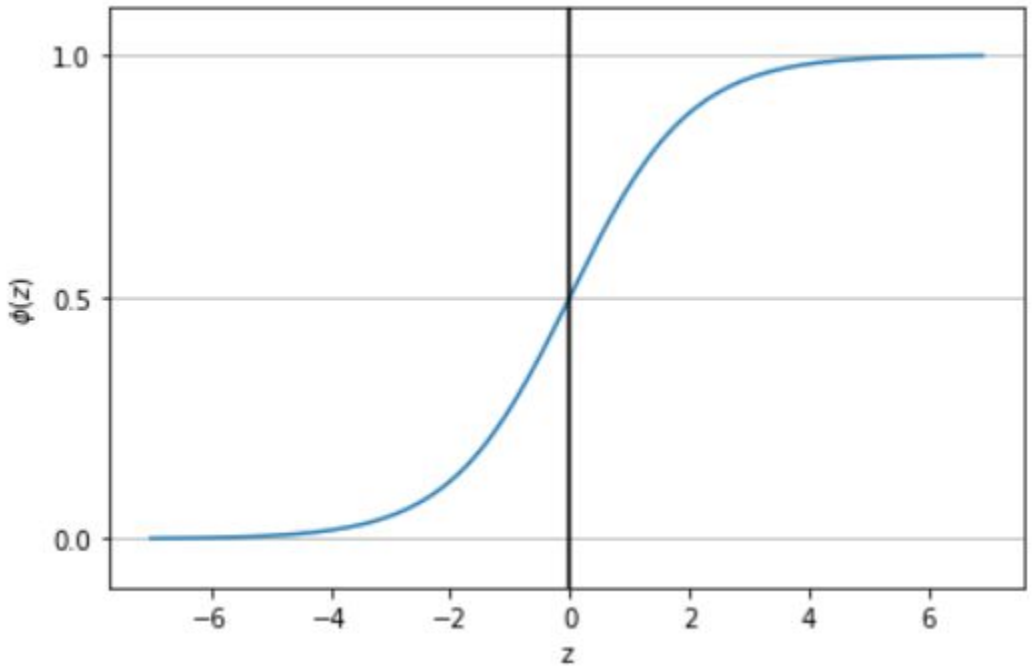
- Nuestra verdadera motivación tras esto es predecir la probabilidad de que una muestra pertenezca a una clase determinada
- Esta es la inversa de la función logit, y se denomina frecuentemente la función sigmoide.

$$\phi(z) = \frac{1}{1 + e^{-z}}$$

2. Algoritmos Aprendizaje Supervisado

Regresión - Regresión logística (III)

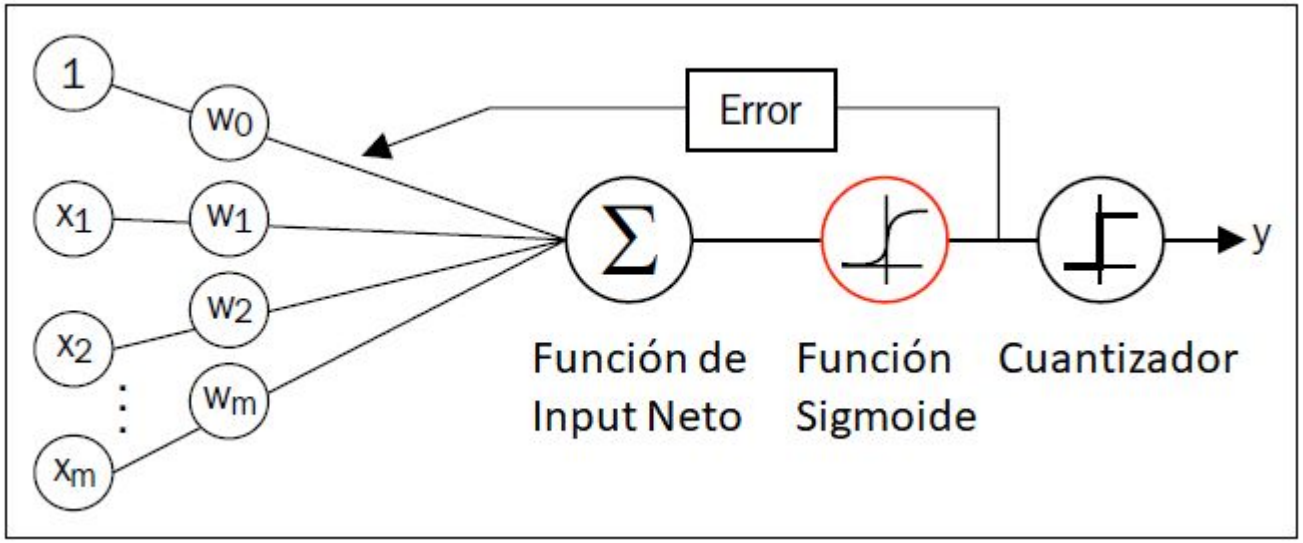
- Cuando se representa en una gráfica, adopta la siguiente forma:



2. Algoritmos Aprendizaje Supervisado

Regresión - Regresión logística (IV)

- Podemos ver que hay dos valores límite en el eje Ø(z), 1 y 0. Significa que la función se aproxima a 1 si z tiende a infinito y o a 0 si z tiende a menos infinito.
- De forma que toma valores reales y los transforma en el rango [0,1], interceptando Ø(z) en 0.5.



2. Algoritmos Aprendizaje Supervisado

Regresión - Regresión logística (V)

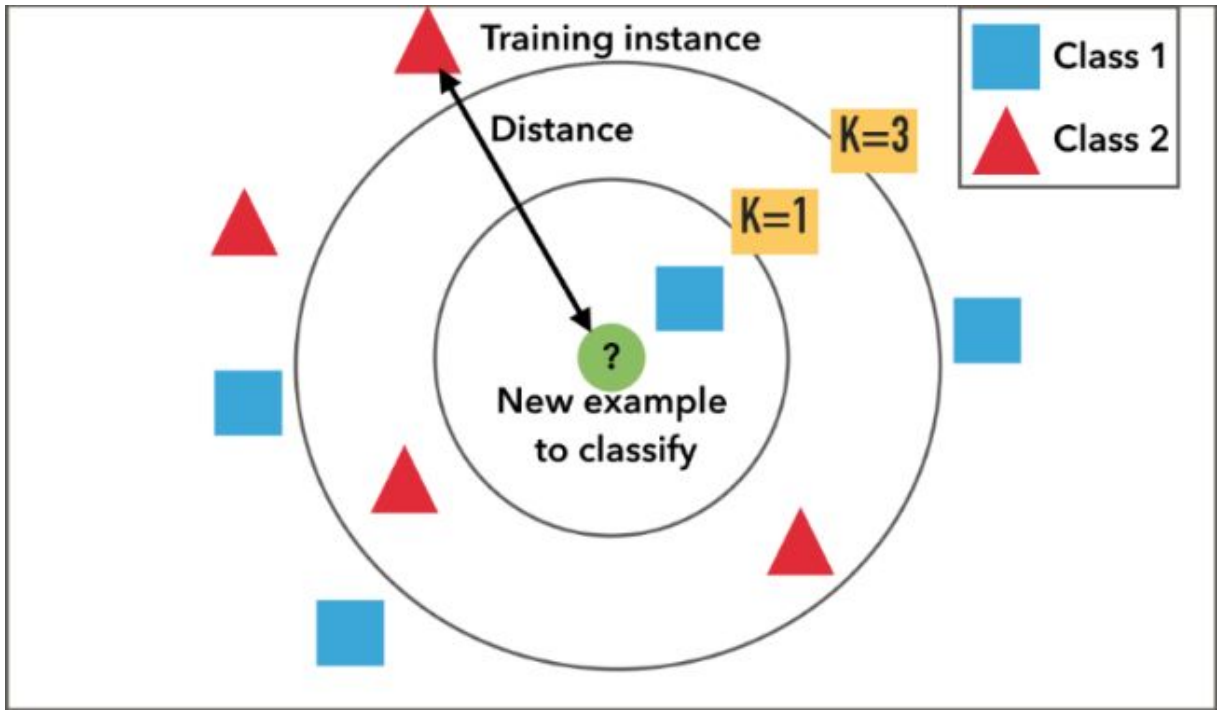
- Esta es una de las razones principales por las que el algoritmo de regresión es tan popular, porque devuelve la probabilidad (como valor entre 0 y 1) de una cierta muestra que pertenece a una clase particular.
- Esto es extremadamente útil en casos como la predicción meteorológica, con la cual no solo nos gustaría saber si el tiempo va a ser lluvioso sino también las probabilidades de que llueva
- En el ámbito médico, por ejemplo, la probabilidad de que un paciente tenga cierta enfermedad.
- ¡Y en muchos otros campos!

2. Algoritmos Aprendizaje Supervisado
kNN

- Los K-vecinos más cercanos, o KNN, pertenecen a un tipo especial de modelos de machine learning que se llaman frecuentemente “algoritmos perezosos”.
- Reciben este nombre porque no aprenden cómo discriminar el conjunto de datos con una función optimizada, en su lugar memorizan el conjunto de datos.
- El nombre también se refiere a la clase de algoritmos llamados “no paramétricos”
    - Estos son algoritmos basados en instancia, que se caracterizan por memorizar el conjunto de datos de entrenamiento, y el aprendizaje perezoso es un caso particular de estos algoritmos, asociados con coste computacional cero durante el aprendizaje.

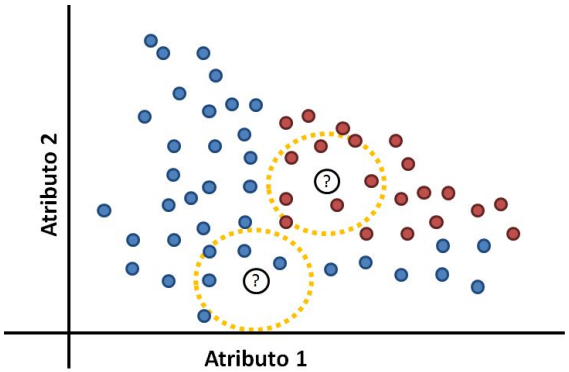
2. Algoritmos Aprendizaje Supervisado
kNN (II)

- El proceso que sigue el algoritmo es:
  - (1) Escoge el número de k y la distancia.
  - (2) Encuentra el k vecino más cercano de la muestra que se pretende clasificar.
  - (3) Asigna la etiqueta de clase por votación mayoritaria.



2. Algoritmos Aprendizaje Supervisado
kNN (III)

- Principio "*Dime con quién andas…*"
  - Se basa en calcular la clasificación directamente a partir de los ejemplos
- Clasificar a partir de los ejemplos más "cercanos"
  - Concepto de distancia
  - Asigna la clase mayoritaria entre los K (parámetro a ajustar por el data scientist) más cercanos al nuevo ejemplo
- Adaptable a Regresión
  - Promedio, mediana,…
  - De los K más cercanos



2. Algoritmos Aprendizaje Supervisado
kNN (IV)

- El algoritmo encuentra las k muestras que son más cercanas al punto que se quiere clasificar, basando sus predicciones en la distancia métrica.
- La principal ventaja es que se adapta a los nuevos datos de entrenamiento, al ser un algoritmo basado en la memoria
- La desventaja es que el coste computacional se incrementa linealmente con el tamaño de los datos de entrenamiento.

2. Algoritmos Aprendizaje Supervisado
kNN (V)

- Puntos a tener en cuenta:
  - Si el algoritmo se enfrenta a un bucle, preferirá los vecinos con menor distancia a la muestra de clasificación
  - Si la distancia es similar, el KNN elegirá la etiqueta de clase que esté primero en el conjunto de datos.
  - Es fundamental elegir el valor k correcto para tener un buen balance entre el sobreajuste y el subajuste.
  - También es crucial establecer una distancia métrica apropiada. Normalmente se usa la distancia 'Minkowski' que es una generalización de las distancias "Euclídea" y "Manhattan"

2. Algoritmos Aprendizaje Supervisado
Näive Bayes

- Naive Bayes es un método de clasificación tanto para problemas binarios como multi-clase
  - Es Naive (ingenuo) porque asume que todas las entradas son independientes entre sí
  - Que unos atributos no dependen de otros (ni un poco)
    - Esto no suele ser así, cuando tratamos datos reales
    - No obstante, funciona “sorprendentemente” bien
- Para el proceso de aprendizaje, solo tenemos que estimar las probabilidades:
  - $P(v_j)$ ➔ probabilidades a priori
    - Probabilidad de que una muestra sea de una determinada clase, independientemente del valor de sus atributos
  - $P(a_i|v_j)$ ➔ probabilidades condicionadas
    - Probabilidad de que una muestra tenga un determinado valor de un atributo, conociendo el valor de la clase
- Simplemente contamos frecuencias

2. Algoritmos Aprendizaje Supervisado

Näive Bayes (II)

- **Calculamos $P(vj)$**
  - P(SI) = 9/14
  - P(NO) = 5/14
- **Calculamos $P(ai|vj)$**
  - P(Cielo=soleado|SI) = 2/9
  - P(Cielo=soleado|NO) = 3/5
  - P(Cielo=nublado|SI) = ...
  - P(Cielo=nublado|NO) = ...
  - P(Temp=alta|SI) = ...
  - P(Temp=alta|NO) = ...
  - P(Temp=suave|SI) = 5/9
  - P(Temp=suave|NO) = 1/5

EJ	Cielo	Temp.	Humedad	Viento	Jugar Tenis
D_1	Soleado	Alta	Alta	Débil	NO
D_2	Soleado	Alta	Alta	Fuerte	NO
D_3	Nublado	Alta	Alta	Débil	SI
D_4	Lluvia	Suave	Alta	Débil	SI
D_5	Lluvia	Baja	Normal	Débil	SI
D_6	Lluvia	Baja	Normal	Fuerte	NO
D_7	Nublado	Baja	Normal	Fuerte	SI
D_8	Soleado	Suave	Alta	Débil	SI
D_9	Soleado	Baja	Normal	Débil	NO
D_{10}	Lluvia	Suave	Normal	Débil	SI
D_{11}	Soleado	Suave	Normal	Fuerte	SI
D_{12}	Nublado	Suave	Alta	Fuerte	SI
D_{13}	Nublado	Alta	Normal	Débil	SI
D_{14}	Lluvia	Suave	Alta	Fuerte	NO

2. Algoritmos Aprendizaje Supervisado

Näive Bayes (III)

- Devolvemos aquella clase que obtenga mayor producto de las probabilidades asociadas
- Limitaciones
  - Clases con frecuencias muy imbalanceadas
  - Aparición de ceros
- Extensiones
  - Cálculo de la distribución de probabilidades para atributos numéricos
  - Corrección de Laplace para evitar problemas con ceros
  - Uso de logaritmos y sumatorios

$$P(v_j)\prod_i P(a_i|v_j)$$

2. Algoritmos Aprendizaje Supervisado
Näive Bayes (IV)

- Son un ejemplo de modelos generativos: intentan modelizar la distribución de los datos
  - ¿Cómo funciona? Ejemplo de Text Mining
    - El caso: queremos clasificar 4 documentos en 2 temáticas (economía y tecnología)
    - Para ellos disponemos de la lista de palabras que aparecen en cada documento
    - Contamos cuántas veces aparece cada palabra en cada documento
  - El segundo documento se representa por (0,1,2,0,0,1)
    - También podríamos considerar únicamente si aparece o no cada una de las palabras (0,1,1,0,0,1)

	market	stock	price	application	mobile	google
document 1('economics')	1	2	3	0	0	0
document 2('economics')	0	1	2	0	0	1
document 3('technology')	0	0	0	2	3	1
document 4('technology')	1	0	1	2	3	0

2. Algoritmos Aprendizaje Supervisado

Näive Bayes (V)

- **Ventajas**
  - Fácil y muy efectivo
  - Funciona bien con outliers y missing data
  - Fácil de obtener las predicciones
  - Funciona bien con datos pequeños y con grandes
- **Desventajas**
  - Se basa en la asunción de independencia
  - No es ideal cuando tenemos muchas variables numéricas (hay que discretizarlas)

2. Algoritmos Aprendizaje Supervisado

Näive Bayes (VI)

Deusto Data

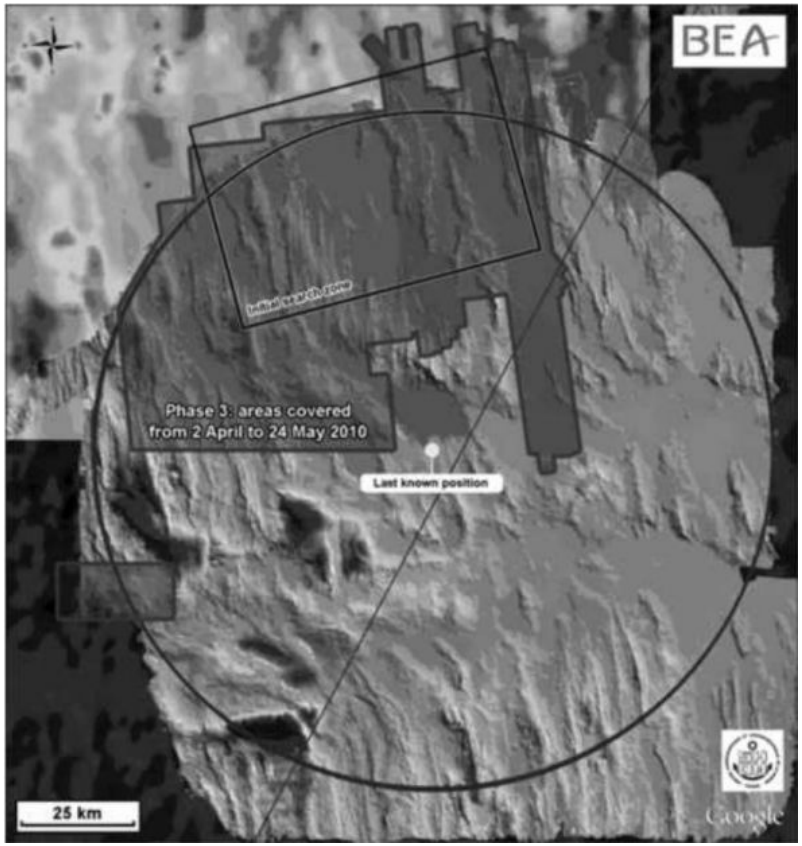
BAYES Y LA INTELIGENCIA COLECTIVA PARA PREDECIR SUCESOS (FÚTBOL, CATÁSTROFES AÉREAS, POLÍTICA, ETC.)

6 MAYO, 2016 ÁLEX RAYÓN DEJAR UN COMENTARIO EDITAR

Kenneth Arrow, premio Nobel de Economía en 1972, y experto en predicciones económicas dijo aquello de:

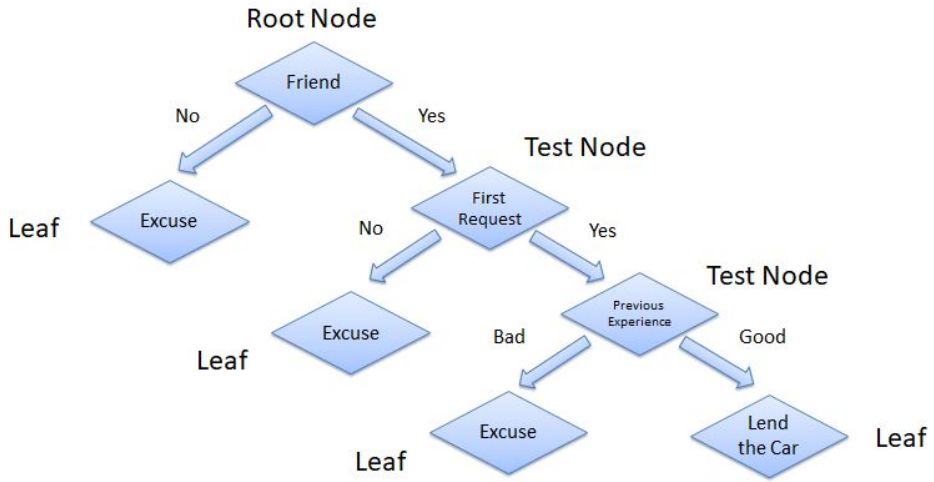
"El buen pronóstico no es el que te dice que lloverá, sino el que te da las probabilidades".

Esto es algo que suelo comentar a la hora de hablar de predicciones. No tienen más que abrir muchos titulares de periódicos para darse cuenta que **la ausencia de la estimación de probabilidades es palpable**. Y eso a pesar que **nada es seguro hasta que ocurre y que la probabilidad cero no existe**. La certeza y la magia debieran quedar excluidas de nuestra manera de ver el mundo.



2. Algoritmos Aprendizaje Supervisado
Árboles

- Los algoritmos de árbol de decisión desglosan el conjunto de datos mediante la formulación de preguntas hasta conseguir el fragmento de datos adecuado para hacer una predicción.
- Para verlo de una manera gráfica, consideremos un ejemplo de un árbol de decisión para determinar si es adecuado prestarle el automóvil a alguien:



2. Algoritmos Aprendizaje Supervisado
Árboles (II)

- Se basan en el principio “divide y vencerás”
  - Toman los datos de entrada y los van partiendo en pedazos para los que sea “más fácil“ ajustar un modelo
  - Los datos se van partiendo en función de una condición aplicada sobre uno (o más) atributos
- Se apoya en
  - Dos proceso iterativos
    - ¿Cómo elegir la manera de partir los datos?
    - ¿Cuándo dejar de partir los datos?
  - Post-proceso
    - ¿Puedo cortar algunas ramas? (poda)
- Son un típico de clasificadores no lineales

2. Algoritmos Aprendizaje Supervisado
Árboles (III)

- Basado en las características de los datos de entrenamiento, el árbol de decisión “aprende” una serie de factores para inferir las etiquetas de clase de los ejemplos.
- El nodo de comienzo es la raíz del árbol, y el algoritmo dividirá de forma iterativa el conjunto de datos en la característica que contenga la máxima ganancia de información, hasta que los nodos finales (hojas) sean puros.

2. Algoritmos Aprendizaje Supervisado
Árboles (IV)

- **Hiperparámetros de los árboles de decisión**
  - **(a) Máxima profundidad**
    - La máxima profundidad es la mayor longitud desde la raíz a las hojas
    - Una gran profundidad puede causar sobreajuste, y pequeña profundidad puede causar subajuste
    - Para evitar sobreajuste, 'podaremos' el árbol de decisión estableciendo un hiperparámetro con la máxima longitud.

$$\text{Max Depth} = K \rightarrow \text{at most } 2^K \text{ Leaves}$$

  - **(b) Máximo número de muestras**
    - Cuando cortamos un nodo, se podría tener el problema de conseguir 99 muestras en uno de los cortes y 1 muestra en el otro, lo que sería un mal uso de los recursos. para evitarlo, podemos establecer un máximo para el número de muestras que permitimos para cada hoja. Esto se puede especificar como un entero o como un número flotante.
    - Un pequeño número de muestras caerá en sobreajuste, mientras que un gran número de muestras caerá en subajuste.

2. Algoritmos Aprendizaje Supervisado
Árboles (V)

- **Hiperparámetros de los árboles de decisión**
  - **(c) Mínimo número de muestras**
    - Análogo al anterior, pero con valores mínimos.
  - **(d) Máximo número de características**
    - Muy frecuentemente tendremos muchas características (columnas) para construir un árbol
    - En cada corte, tendremos que hacer revisar todo el conjunto de datos en cada una de las características, lo que puede ser muy costoso.
    - Una solución a este problema es limitar el número de características que se buscan en cada corte
    - Si este número es suficientemente alto, es probable que encontramos una buena característica entre aquellas que buscamos (aunque pueda no ser la perfecta)
    - Sin embargo, si no es tan alto como el número total de características, la velocidad de los cálculos se elevará de manera significativa.

2. Algoritmos Aprendizaje Supervisado
Árboles (VI)

- Algunas consideraciones a tener en cuenta:
  - El árbol tiene que elegir las “mejores variables” para hacer los cortes ➔ **Significatividad**
  - Las variables continuas se discretizan.
    - ¿Qué ha pasado con la edad en el ejemplo anterior?
    - ¿Cómo las discretizamos? Umbrales para cada variable.
  - Las hojas hacen la predicción final:
    - Si las hojas son “puras”, ya tenemos la etiqueta.
    - Si son impuras, se guarda la frecuencia de cada clase.
  - Puede haber **más de un árbol que funcione para los mismos datos**. De todos ellos nos gustaría quedarnos con el árbol que tenga el **mínimo número de nodos**

2. Algoritmos Aprendizaje Supervisado
Árboles (VII)

- Metodología para construir el árbol: Necesitamos
  1. Un criterio para evaluar la ventaja derivada de la división de un nodo. ¿Qué nodo dividir en cada etapa?
  2. ¿Cómo estimar la tasa de mala clasificación (o varianza de predicción en el caso de árboles de regresión)?
  3. ¿Cuando detenemos el crecimiento del árbol, o cuando podamos un árbol que ha crecido en exceso? Árboles son modelos muy expresivos y pueden "aprender" perfectamente cualquier muestra de entrenamiento y fácilmente hacer overfitting.
  4. Un criterio para asignar un valor (o etiqueta de clase) a cada hoja.

2. Algoritmos Aprendizaje Supervisado
Árboles (VIII)

- Dos posibles estrategias de elección del árbol
  - Dejar de crecer el árbol cuando el corte no es estadísticamente significativo.
  - Crear un árbol completo y podarlo. ¿Cómo?
  - Por ejemplo dividiendo los datos en muestras de entrenamiento y validación
  - Crear un árbol completo en la muestra de entrenamiento
  - Evaluar en la muestra de validación el impacto de podar cada nodo hasta que seguir podando sea dañino
  - Otro: eligiendo el subárbol que tenga la menor tasa de error penalizada (CART)
- Algoritmos de creación de árboles: AID, CHAID, C4, C5, CART, QUEST, etc.

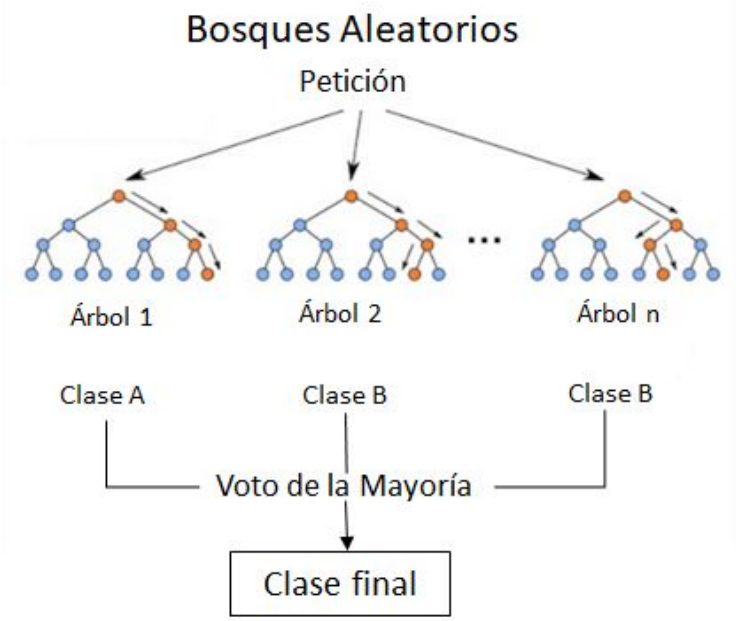
2. Algoritmos Aprendizaje Supervisado
Árboles (VIII)

- **Ventajas**
  - Clasificador válido siempre
  - Muy automatizado, todo tipo de inputs y missing values
  - Descarta variables no importantes
  - Fácil de interpretar
- **Desventajas**
  - Los cortes se suelen hacer en base a variables que tengan muchos niveles
  - Fácil overfitting
  - Sensibles a cambios en los datos de entrenamiento
  - Árboles frondosos difíciles de interpretar y pueden repetir variables

2. Algoritmos Aprendizaje Supervisado

Random Forest

- Si tenemos un conjunto de datos con muchas características (columnas), el algoritmo del árbol de decisión tiende a sobreajustar, añadiendo complejidad al modelo y el proceso de aprendizaje.
- Podemos solventar este punto seleccionando cada columna de forma aleatoria y realizando árboles de decisión para cada conjunto de columnas.



2. Algoritmos Aprendizaje Supervisado
Random Forest (II)

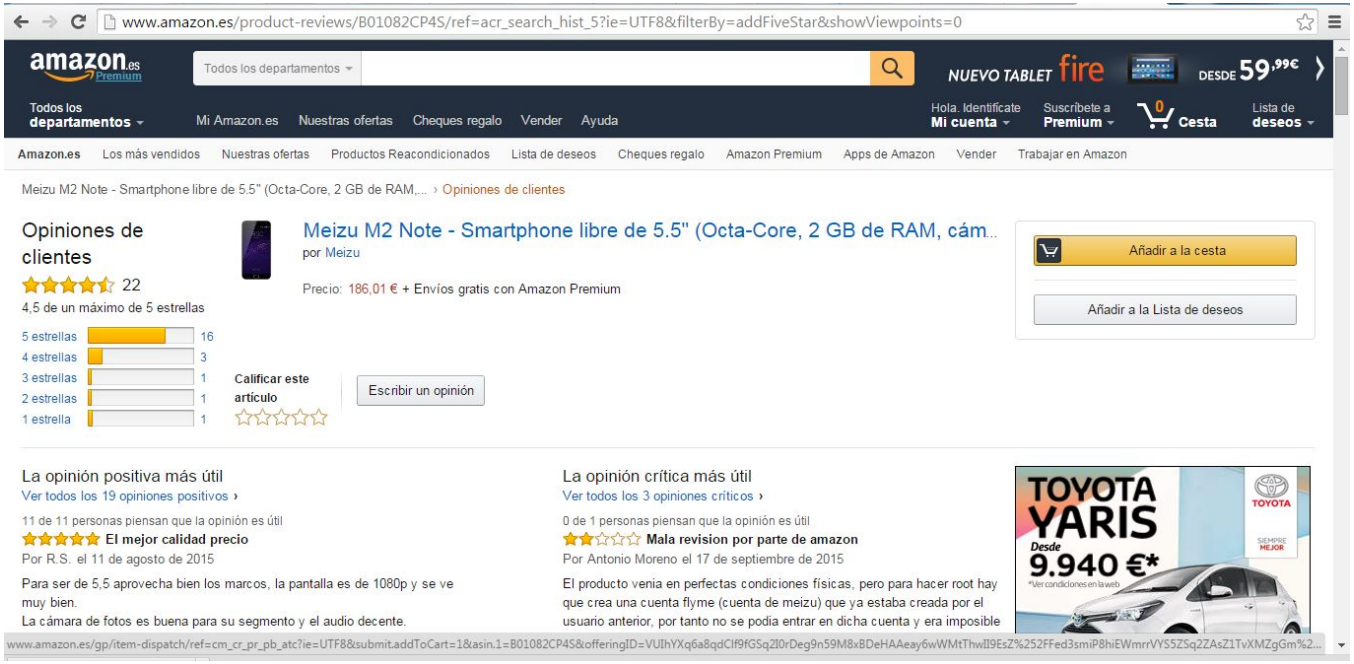
- Los métodos basados en árboles son fáciles de interpretar
- Pero no suelen ser competitivos en términos de precisión con los mejores métodos supervisados
- ¿Cómo mejorar los árboles? Agregando muchos árboles
  - De esta forma, se desarrolla un algoritmo de agrupación de aprendizaje que combinará una serie de modelos más débiles para crear otro más robusto.
- El algoritmo realizará los siguientes pasos:
  - Diseñar una muestra de arranque de tamaño n.
  - Desarrollar un árbol de decisión desde la muestra de arranque. En cada nodo habrá características seleccionadas aleatoriamente sin reemplazamiento y el nodo se cortará maximizando la ganancia de información.
  - El proceso previo se repetirá K veces.
  - Agregar la predicción hecha para cada árbol, asignando la etiqueta de clase por votación mayoritaria

2. Algoritmos Aprendizaje Supervisado
Random Forest (III)

- La principal ventaja de este método es que normalmente no necesitaremos podar el bosque aleatorio (ya que el modelo es muy resistente al ruido)
  - Sin embargo, es mucho menos interpretable que los árboles de decisión.
- El único hiperparámetro que necesitaremos ajustar es el número de árboles K. Normalmente, cuanto más grande es K , mejor se comportará el modelo, pero esto incrementará drásticamente el esfuerzo de computación (y por tanto, el coste).

2. Algoritmos Aprendizaje Supervisado
Random Forest (IV)

- ¿Qué hace una persona antes de comprar un producto? Buscar entre las opiniones de distintos expertos



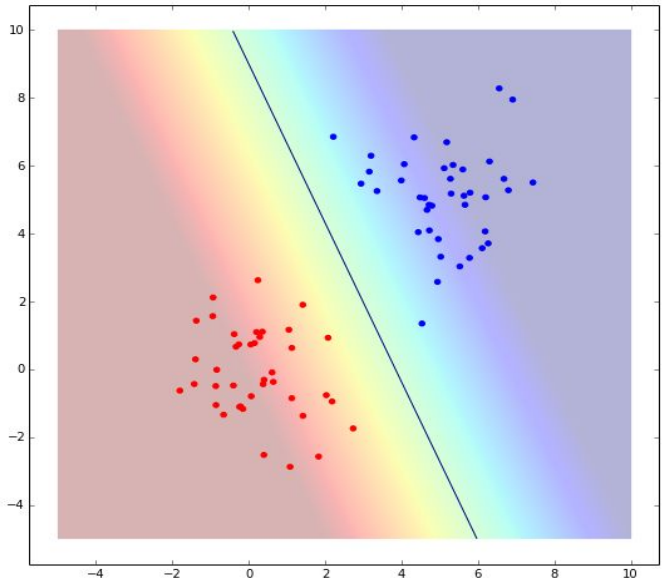
2. Algoritmos Aprendizaje Supervisado
Random Forest (V)

- Imaginemos que quiero una sistema de predicción de peso
  - ¿Cuánto pesan estas naranjas?
  - Puedo tantear a toda la clase a ver
    - ¿Quién cocina? ¿Quién come fruta? ¿Quién hace la compra normalmente? ¿Quién compra fruta al peso? ¿A quién le gustan las naranjas? ¿Alguien ha trabajado en una frutería?...
  - Con el fin de escoger a la persona (sistema) de predicción
  - También puedo preguntaros a todos y tomar el promedio
- ¿Ventajas e inconvenientes de cada enfoque?

2. Algoritmos Aprendizaje Supervisado

Máquinas de Vector Soporte (SVM)

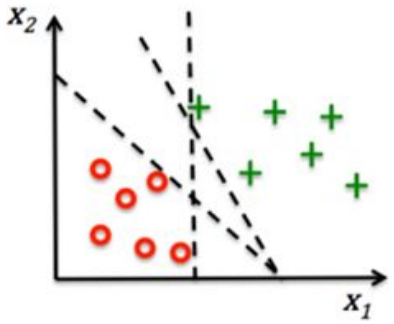
- Con Naive Bayes estimamos la distribución de los datos
- Con SVM vamos a buscar una regla de discriminación, modelizar la frontera entre las clases
  - 2 clases, pero se puede generalizar
  - Vamos a modelar esa frontera con un modelo lineal
- En los problemas de clasificación primero calculan la probabilidad de pertenencia a cada una de las clases de la variable cualitativa
  - Las variables deben ser numéricas



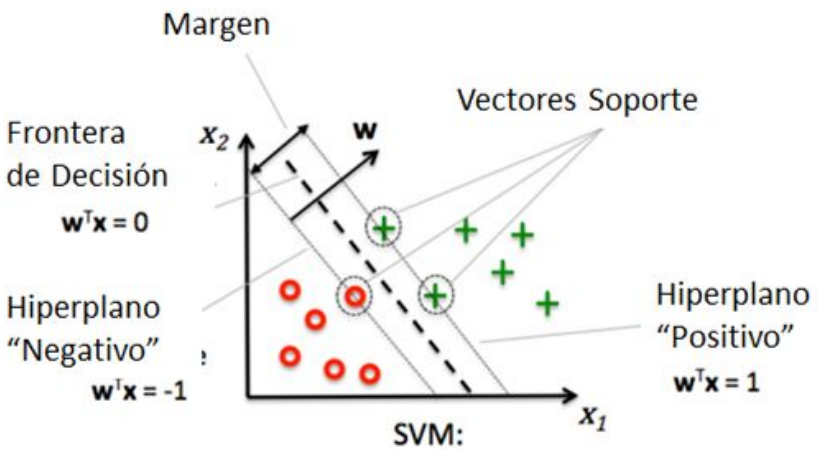
2. Algoritmos Aprendizaje Supervisado

Máquinas de Vector Soporte (SVM) (II)

- Este algoritmo puede ser considerado una extensión del algoritmo “perceptron”
- En SVM el objetivo de la optimización es establecer una línea de decisión que separe las clases maximizando el margen entre esta línea y los puntos de muestra cercanos a este hiperplano
  - Estos puntos se llaman **vectores soporte**



¿Qué Hiperplano?



SVM:
Maximizar el margen

2. Algoritmos Aprendizaje Supervisado
Máquinas de Vector Soporte (SVM) (III)

- Un hiperplano en R^d se define como
  - Divide el espacio en dos partes
  - Valores positivos a los que quedan a un lado del plano
  - Valores negativos a los que quedan al otro
- Como queremos el hiperplano con máximo margen ("*Maximizar la distancia de los puntos más cercanos a la frontera definida por el plano*"), se añaden dos rectas paralelas (márgenes) e intentamos maximizar sus distancias a la línea de decisión original
  - Tendremos en cuenta los puntos sin clasificar (errores) y los que quedan entre los márgenes de la línea
  - Normalmente, las líneas de decisión con márgenes grandes tienden a tener un error de generalización menor
    - Por otro lado, los modelos con márgenes pequeños tienen menor tendencia al "sobreajuste" (overfitting).

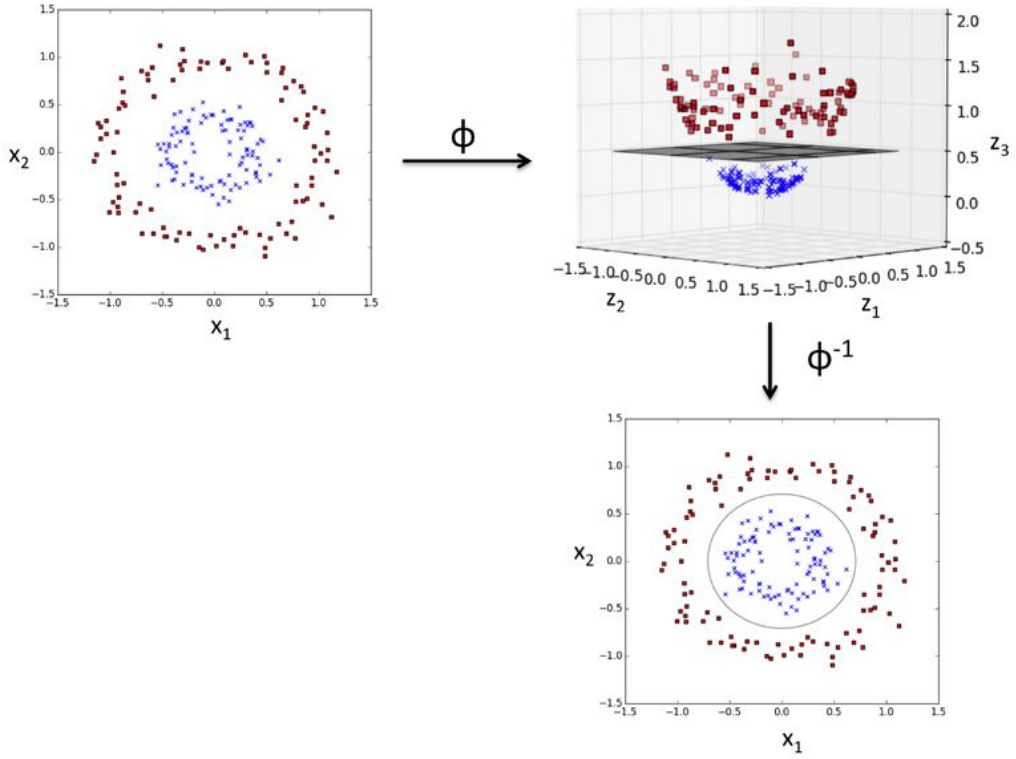
2. Algoritmos Aprendizaje Supervisado
Máquinas de Vector Soporte (SVM) (IV)

- SVM es una rama popular de algoritmos, ya que se pueden utilizar para resolver problemas de clasificación no lineales.
  - Esto se realiza con un método denominado “kernelización” (kernelizing).
- La idea básica de uso de kernel, cuando tratamos con combinaciones no lineales de las características originales, es proyectarlas en un espacio con más dimensiones vía una función de correspondencia Ø, de forma que los datos sean linealmente separables.
- Intuitivamente, el conjunto de datos original se transforma en otro de más dimensiones, y después se aplica una proyección para hacer las clases separables.

2. Algoritmos Aprendizaje Supervisado

Máquinas de Vector Soporte (SVM) (V)

- Después se aplica el algoritmo, se separan las clases y se aplica la función inversa a la que provoca la proyección para volver a la distribución original de los datos.



2. Algoritmos Aprendizaje Supervisado

Máquinas de Vector Soporte (SVM) (VI)

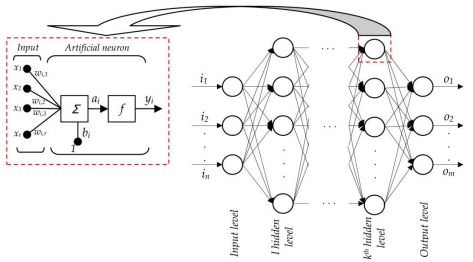
- **Ventajas**
  - Sirve para problemas de clasificación y regresión
  - No está muy afectado por outliers
  - No es propenso al overfitting
  - Buena precisión a la hora de predecir
- **Desventajas**
  - Caja negra: difícil de interpretar
  - Entrenamiento lento
  - Requiere probar con distintas configuraciones

2. Algoritmos Aprendizaje Supervisado
Redes neuronales

- Imitan funcionamiento del cerebro
- **Neuronas** ➔ subunidades de una red de neuronas en el cerebro; funcionamiento:
  - Señales eléctricas de distintas variables llegan a las dendritas
  - Esas señales se acumulan en el cuerpo celular de la neurona
  - Si la señal acumulada>límite se genera una señal de salida que se traslada a través del axón
- **Ventajas**
  - Métodos paramétricos que ajustan bien relaciones no lineales
- **Desventajas**
  - Difíciles de interpretar
- **¿Cuándo se utilizan las redes neuronales?**
  - Cuando no conocemos la relación entre las variables input y las output
  - Es más importante la precisión en la predicción que la explicación del modelo

2. Algoritmos Aprendizaje Supervisado

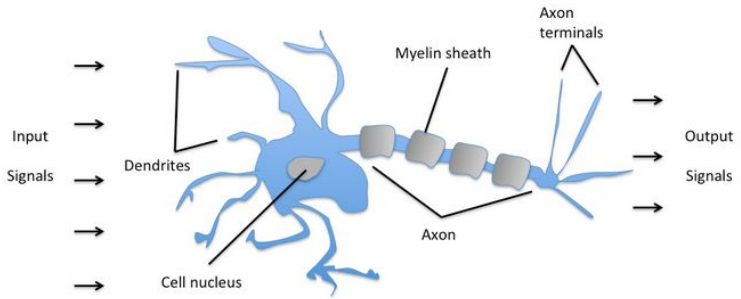
Redes neuronales (II)



- Funcionamiento de una “neurona”
  - Cada neurona recibe diferentes entradas (x_1,x_2, x_3,...)
  - Cada señal se multiplica por un valor, peso, que da una idea de la fuerza de esa conexión (w_1,w_2, w_3,...)
  - La neurona calcula una salida, usando una función de Transferencia (F)
- Una red neuronal no es más que un conjunto de neuronas (normalmente) organizadas en capas
- La respuesta de la neurona de salida se compara con la respuesta deseada y el error cometido se utiliza para modificar los pesos
- El error se propaga hacia neuronas predecesoras, según la influencia sobre ésta neurona
- El proceso sigue hacia atrás

2. Algoritmos Aprendizaje Supervisado
Redes neuronales (III)

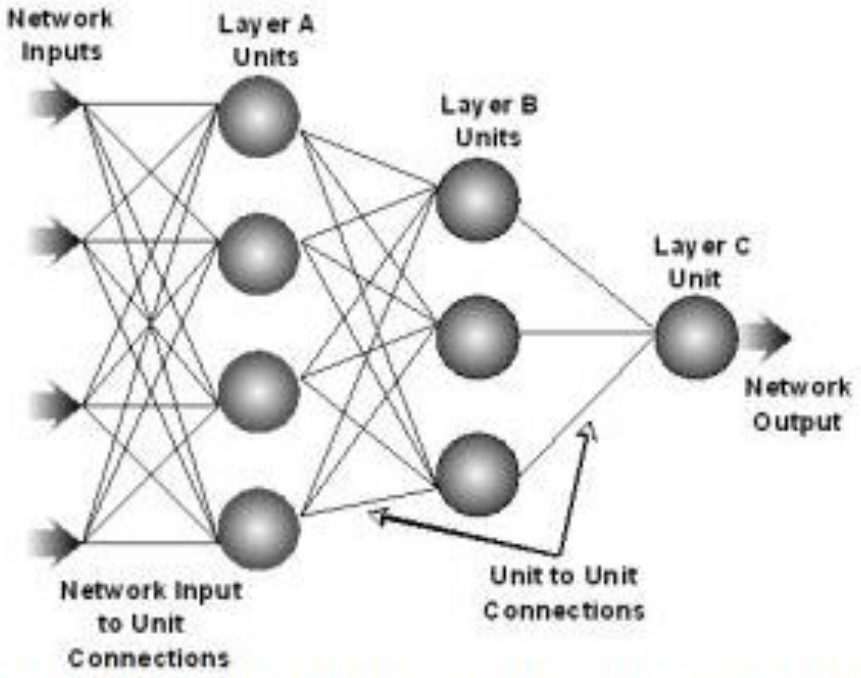
- **Feedforward Neural Networks**
  - Una red neuronal consiste de unidades (neuronas y conexiones entre ellas).
  - Tipos de unidades:
    - Unidades input: reciben los valores de las variables input y pueden estandarizarlos
    - Unidades ocultas: cálculos internos, proveen no linearidad
    - Unidades output: calculan los valores predichos y comparan esos valores predichos con los valores reales objetivo
  - ¿Cómo se pasa la información de unas unidades a otras?
    - A través de conexiones.
    - La mayoría de las conexiones son hacia adelante (SAS, por ejemplo, sólo permite feedforward networks)
  - Bishop, C.M. (1995), Neural Networks for Pattern Recognition

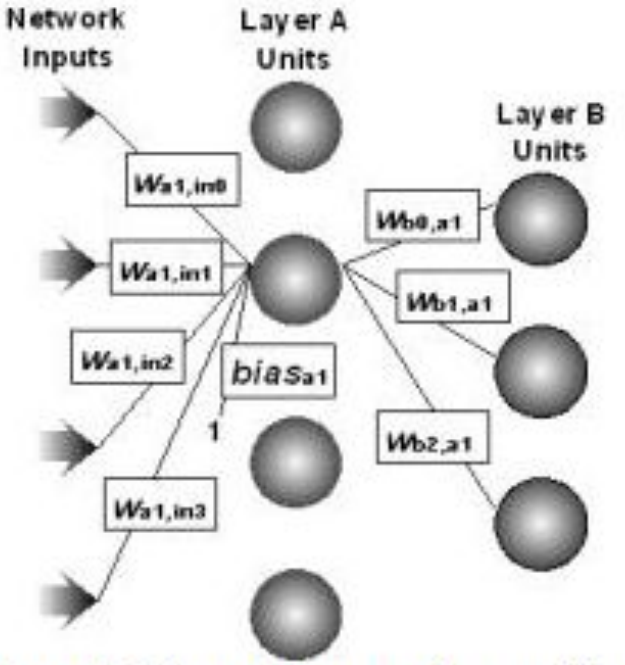


Schematic of a biological neuron.

2. Algoritmos Aprendizaje Supervisado
Redes neuronales (IV)

- **Arquitectura** de las redes neuronales





2. Algoritmos Aprendizaje Supervisado
Redes neuronales (VI)

- Feedback network o recurrent network
  - Permiten las señales ir hacia detrás y hacia delante
  - Mucha mayor complejidad
  - Pueden servir para comprender secuencias de eventos en el tiempo (series temporales)
  - No se utilizan ni están implementadas en SAS/R

2. Algoritmos Aprendizaje Supervisado

Redes neuronales (VII)

- Parámetros:
  - La mayoría de las conexiones tienen un valor numérico llamado peso
  - La red se entrena minimizando la función de error ajustando esos pesos
  - Muchas unidades tienen también dos valores asociados: sesgo y altitud

2. Algoritmos Aprendizaje Supervisado
Redes neuronales (VIII)

- Las unidades ocultas y las output usan dos funciones para producir sus resultados:
  - **Función de combinación**: agrega los resultados de la capa anterior (utiliza los pesos, el sesgo y la altitud). 2 tipos
  - **Lineales**: combinación lineal de los pesos y los valores más el sesgo (intercepto)
  - **Radiales**: (distancia euclídea entre los pesos y los valores)*sesgo^2
- El valor producido por la función de combinación se transforma por la **función de activación**

2. Algoritmos Aprendizaje Supervisado

Redes neuronales (IX)

- Algunas funciones de activación
- Lo que las diferencia es el rango de Valores permitidos que van a la Señal de output

FUNCTION	RANGE	FUNCTION OF NET INPUT g
Identity	$(-\infty, +\infty)$	g
Exponential	$(0, \infty)$	$\exp(g)$
Reciprocal	$(0, \infty)$	$1/g$
Square	$[0, +\infty)$	g^2
Logistic	$(0, 1)$	$\frac{1}{1+\exp(-g)}$
Softmax	$(0, 1)$	$\frac{\exp(g)}{\sum exponentials}$
Gauss	$(0, 1]$	$\exp(-g^2)$
Sine	$[-1, 1]$	$\sin(g)$
Cosine	$[-1, 1]$	$\cos(g)$
Elliott	$(0, 1)$	$\frac{g}{1+\lvert g\rvert}$
Tanh	$(-1, 1)$	$\tanh(g) = 1 - \frac{2}{1+\exp(2g)}$
Arctan	$(-1, 1)$	$\frac{2}{\pi}\arctan(g)$

2. Algoritmos Aprendizaje Supervisado
Redes neuronales (X)

- Las redes neuronales no son sino agrupaciones de capas de neuronas
- Todas las neuronas de una capa comparten características:
  - En la capa de input, todas tienen el mismo método de estandarización
  - Todas las neuronas en una capa oculta tienen las mismas funciones de combinación y activación
- Todas las neuronas en una capa output tienen las mismas funciones de combinación, activación y error

2. Algoritmos Aprendizaje Supervisado
Redes neuronales (XI)

- **Entrenamiento de las redes neuronales**
  - Las redes neuronales aprenden con la experiencia (nuevos datos)
  - A medida que se procesan los input, las conexiones entre neuronas se fortalecen o debilitan (pesos)
  - Los pesos se ajustan para adaptarse a los nuevos datos
  - Entrenar una red neuronal: computacionalmente intensivo

2. Algoritmos Aprendizaje Supervisado
Redes neuronales (XII)

- **Entrenamiento de las redes neuronales**
  - **Backpropagation**: método de cálculo del gradiente de error en una red neuronal feedforward
  - **Backprop network**: red neuronal feedforward entrenada con métodos del gradiente descendiente (técnicas de optimización).
  - **Distintos tipos**: Batch backprop, Incremental backprop, Quickprop, RPROP
    - http://sebastianraschka.com/Articles/2015_singlelayer_neurons.html

2. Algoritmos Aprendizaje Supervisado

Redes neuronales (XIII)

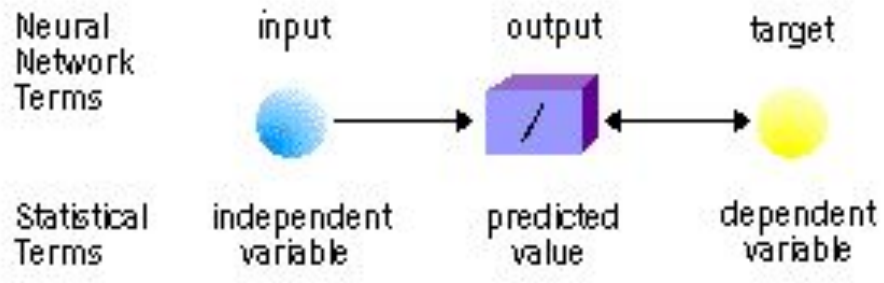
- **Backpropagation**
  - La red empieza fijando pesos de forma aleatoria
  - Proceso iterativo hasta alcanzar un criterio de parada
  - Ciclos (épocas) de dos procesos:
    - **Fase forward**
      - Neuronas activadas en secuencia desde la capa de inputs a la outputs.
      - Aplicar a cada neurona sus pesos y función de activación
      - Al llegar a la capa final se produce una señal de output
    - **Fase backward**: calcular el error y reajustar pesos
      - Comparar la diferencia entre la señal de output y los valores reales.
      - Se propaga el error hacia atrás para recalcular los pesos entre las neuronas

2. Algoritmos Aprendizaje Supervisado
Redes neuronales (XIV)

- ¿Cuánto hay que cambiar los pesos?
  - Técnica del gradiente descendiente:
    - Buscar mínimos siguiendo la dirección con la mayor pendiente descendiente
    - Utiliza las derivadas de la función de activación de las neuronas

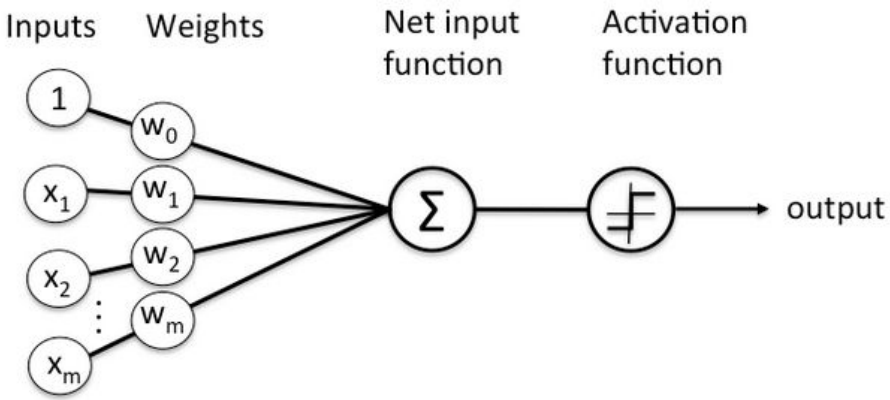
2. Algoritmos Aprendizaje Supervisado
Redes neuronales (XV)

- Ejemplos de redes neuronales más sencillas
  - Regresión
    - Función de activación: identidad o logística
    - Función de combinación: lineal
    - No hay capas ocultas



2. Algoritmos Aprendizaje Supervisado
Redes neuronales (XVI)

- Perceptrón (Análisis discriminante lineal)
  - No hay capas escondidas
  - Función de combinación: lineal
  - Función de activación:escalón



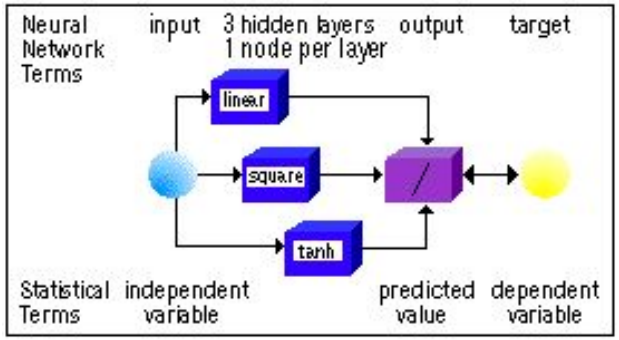
$$g(\mathbf{z}) = \begin{cases} 1 & \text{if } \mathbf{z} \geq 0 \\ -1 & \text{otherwise.} \end{cases}$$

$$\mathbf{z} = w_0 x_0 + w_1 x_1 + \cdots + w_m x_m = \sum_{j=0}^{m} x_j w_j$$

$$= \mathbf{w}^T \mathbf{x}.$$

2. Algoritmos Aprendizaje Supervisado
Redes neuronales (XVII)

- **Multilayer Perceptrón (MLP)**
  - Múltiples transformaciones vía capas ocultas
  - Cada unidad input conectada a una unidad de la capa oculta
  - Cada unidad de la capa oculta conectada a la capa output
  - Capas ocultas combinan los input y aplican una función de activación (lineal o no) MLP: sigmoidal
  - Los resultados se combinan en las unidades output, donde se puede aplicar otra función de activación



2. Algoritmos Aprendizaje Supervisado
Redes neuronales (XVIII)

- **¿Cuántas capas ocultas?**
  - Con muchos datos una sóla capa es suficiente en un MLP
  - No hay teoría sobre cuántas capas ocultas son necesarias
- **Deep neural network**: Redes neuronales con varias capas ocultas
  - "*It has been proven that a neural network with at least one hidden layer of sufficient neurons is a universal function approximator. This means that neural networks can be used to approximate any continuous function to an arbitrary precision over a finite interval*"

2. Algoritmos Aprendizaje Supervisado
Redes neuronales (XIX)

- **¿Cuántas neuronas en las capas ocultas?**
  - Entrenar varias redes con distinto número de neuronas ocultas y estimar el error generalizado de cada red
  - Procedimiento simple: empezar sin neuronas ocultas y añadir una neurona cada vez.
    - Calcular el error de generalización de cada re
    - Dejar de añadir neuronas si aumenta la generalización del error

2. Algoritmos Aprendizaje Supervisado

Redes neuronales (XX)

- **Radial Basis Function (RBF) Networks**
  - Una capa oculta
  - Utiliza como función de activación en las capas ocultas funciones radiales
  - Las funciones de activación en las capas ocultas son la exponencial o la softmax
  - En la capa de output se utilizan funciones de combinación lineales

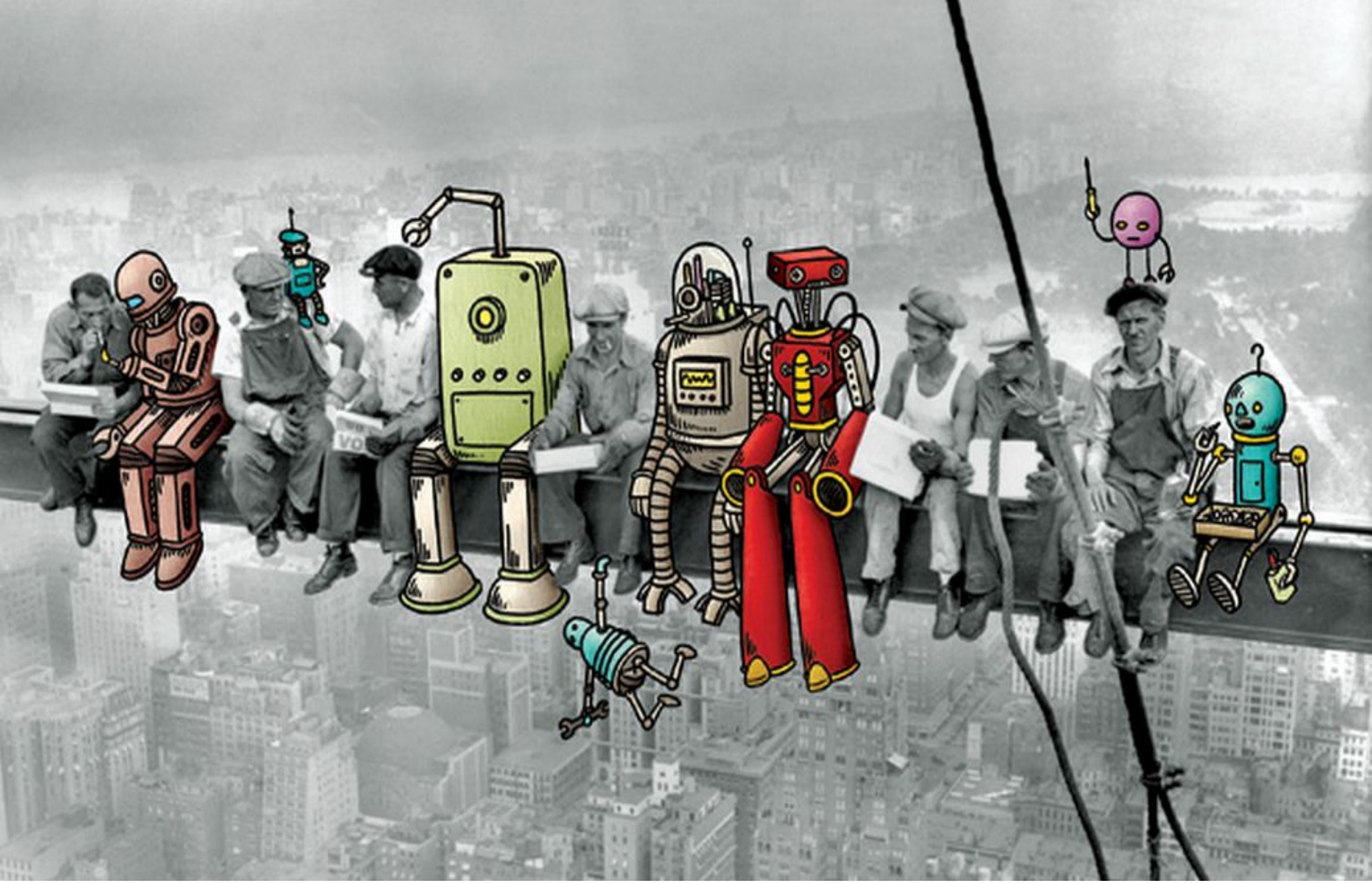
2. Algoritmos Aprendizaje Supervisado

Redes neuronales (XXI)

- **Un consejo previo**
  - Al igual que con otros modelos, estandarizar antes las variables input
  - **Ventajas**
    - Pocas presunciones sobre relaciones entre las variables
    - Valen para clasificación y regresión
    - Modelizan cualquier problema complejo
  - **Desventajas**
    - Caja negra, difícil interpretación
    - Fácil overfitting
    - Computacionalmente intensivas

2. Algoritmos Aprendizaje Supervisado
Redes neuronales (XXII)

- Algunas opciones:
  - Cuántas capas “escondidas”
  - Cuántas neuronas en cada capa
  - Direct Connection si queremos que además haya conexiones directas entre los inputs y los outputs además de las conexiones vía capas intermedias



2. Algoritmos Aprendizaje Supervisado

DA07 - Machine Learning (I)

DA - Data Advanced - Data Analytics Journey