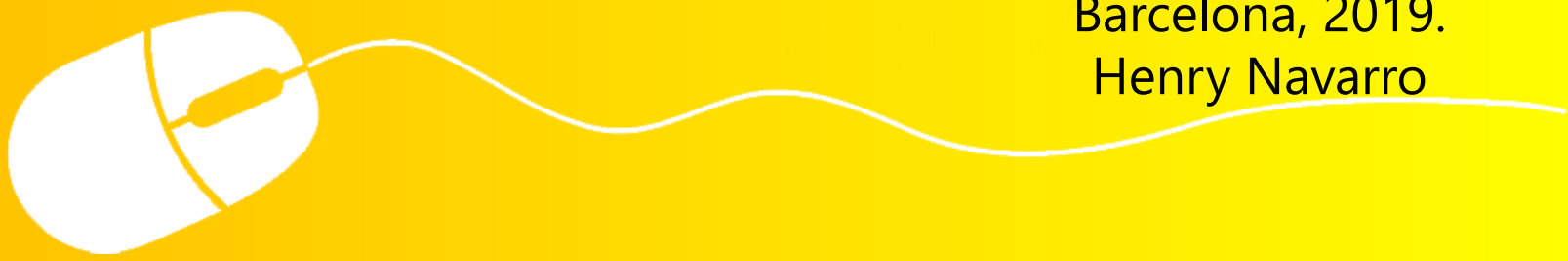
KSCHOOL

Máster Data Science

Barcelona, 2019.

Henry Navarro



KSCHOOL

About me

- Ahora: Lead Data Scientist – Altran Innovation.
- Antes: Data Scientist – Equifax, Solutio, Enefgy.
- Profesor Universidad Carlos III de Madrid – Grupo ML4DS & GISC. (https://goo.gl/9eeHAz)
- Profesor Escuela de Organización Industrial.
- Data Analyst – Ministerio de Justicia (Venezuela).
- Máster en Ingeniería Matemática – UC3M.
- Lienciado en Matemática – UCV.

KSCHOOL

Intro supervised learning

#10YearsChallenge

KSCHOOL

Intro supervised learning

#10YearsChallenge

$$y = \beta X + \varepsilon$$

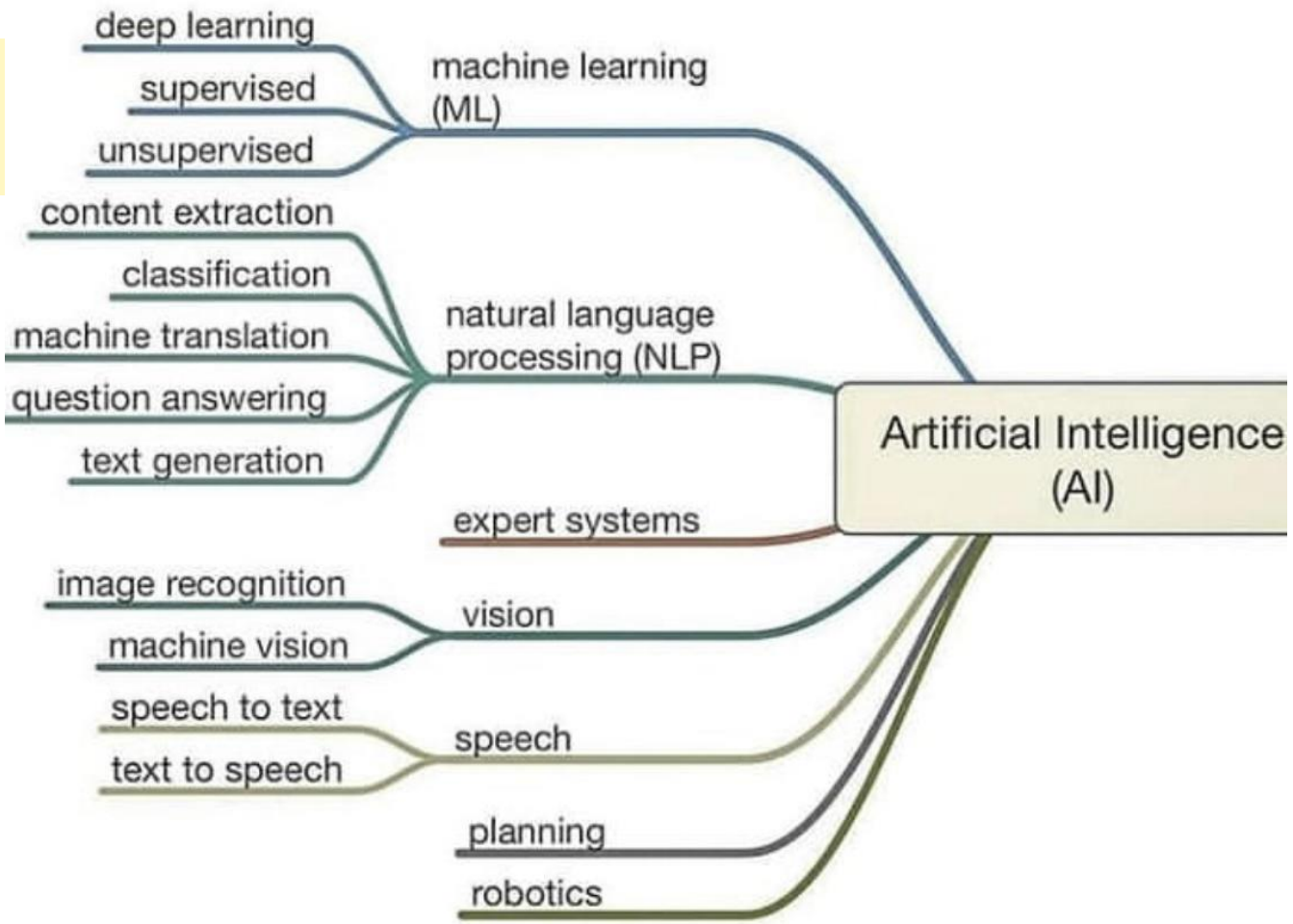
Statistics

$$y = \beta X + \varepsilon$$

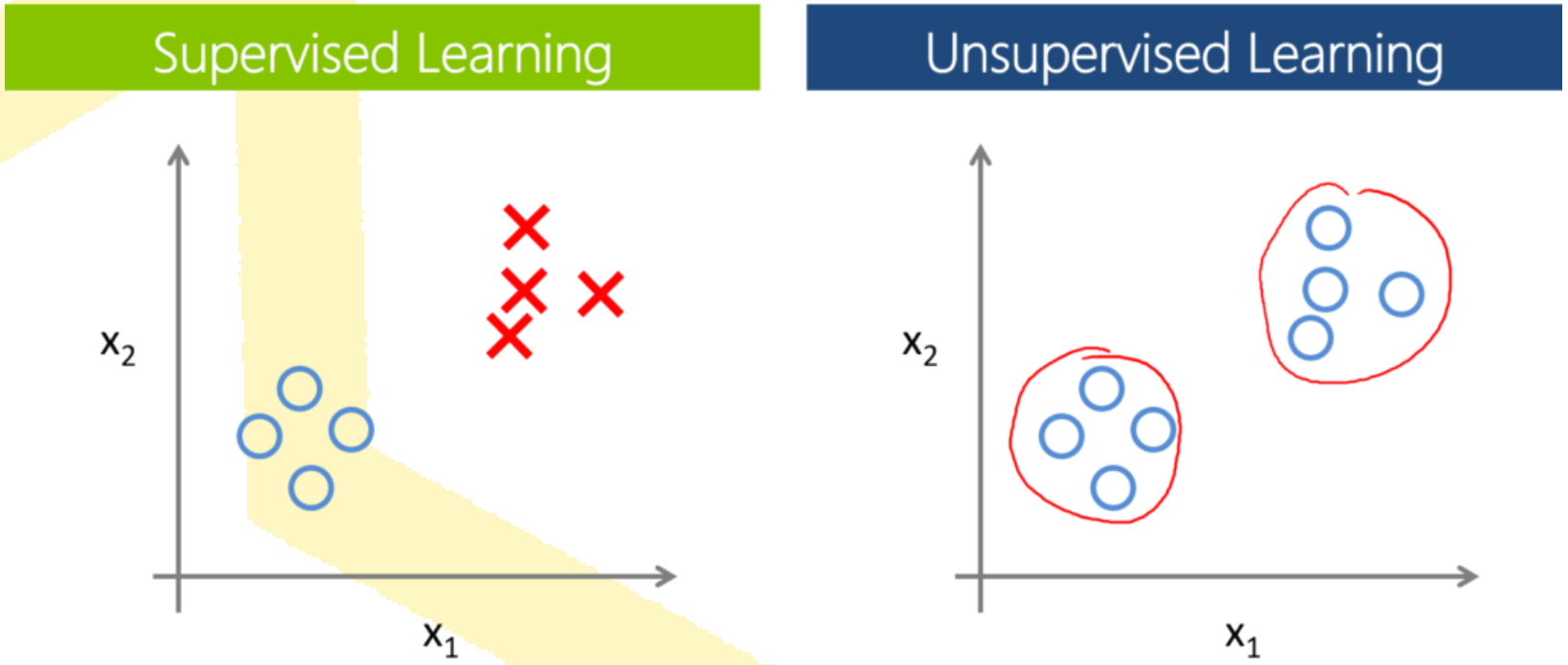
Machine Learning

KSCHOOL

Intro supervised learning



Supervised vs. Unsupervised learning



KSCHOOL

Variable respuesta/objetivo/ dependiente

supervised learning

y	x1	x2	x3	x4	x5	x6
0	186.176767	32.01013	7.389389	56.68727	171.3361	18.03844
1	159.659374	-95.06651	-83.210200	155.31220	-149.0119	-183.90314
0	44.307132	-167.88587	-90.023000	124.17956	170.8277	30.37569
0	129.380781	-83.71101	193.529927	193.97078	135.2245	-157.56599
0	-7.236501	150.92669	-75.665873	58.89800	-114.4337	-58.16047
0	-13.191041	51.07507	168.874093	-73.05704	-179.1995	-178.97354

KSCHOOL

unsupervised learning

x1	x2	x3	x4	x5	x6
186.176767	32.01013	7.389389	56.68727	171.3361	18.03844
159.659374	-95.06651	-83.210200	155.31220	-149.0119	-183.90314
44.307132	-167.88587	-90.023000	124.17956	170.8277	30.37569
129.380781	-83.71101	193.529927	193.97078	135.2245	-157.56599
-7.236501	150.92669	-75.665873	58.89800	-114.4337	-58.16047
-13.191041	51.07507	168.874093	-73.05704	-179.1995	-178.97354

No tenemos variable respuesta

KSCHOOL

Index

- Linear regression.
- Logistic regression.
- Polynomial regression.
- Other generalized linear models.
- Model Diagnosis (along the course).
- Nonlinear Models: splines, decision trees, random forest, k – nearest neighbors, support vector machines.
- Intro to Deep learning.

We won't study...

- Detalles matemáticos
- Forward/Backward propagation.
- Gradient descent algorithm.

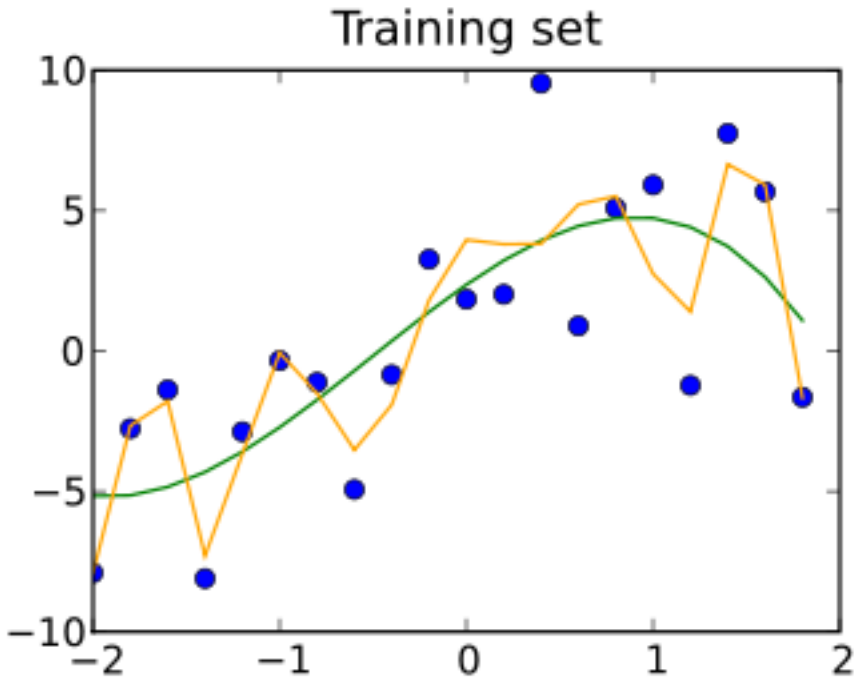
KSCHOOL

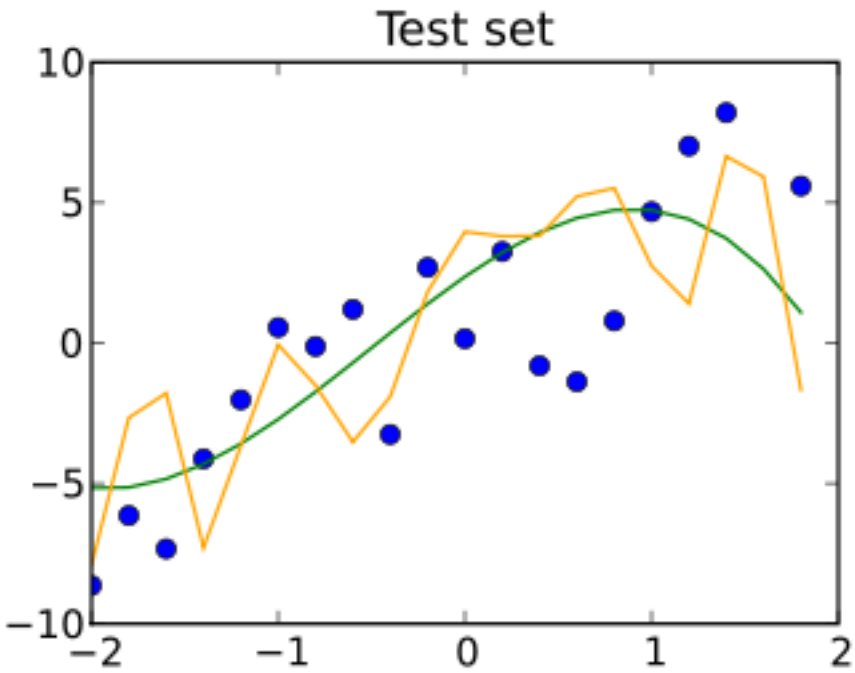
Conceptos previos

- Dataset de train y test.
- Overfitting.
- Matriz de confusión.
- Cross - Validation

KSCHOOL

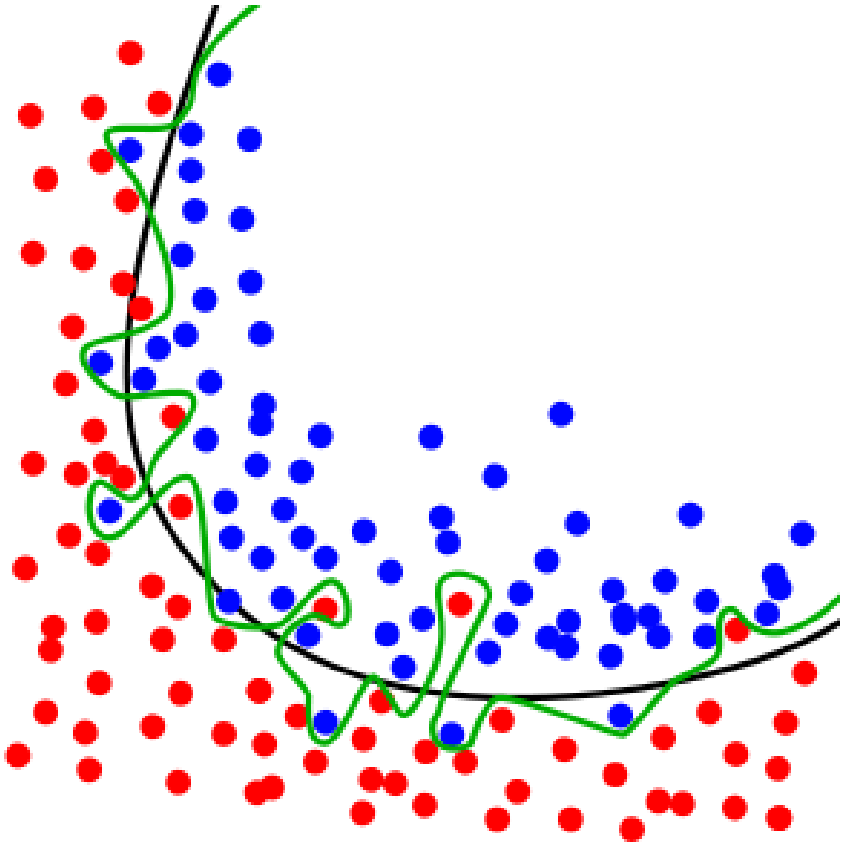
Dataset de train y test.





Overfitting

- Es el efecto de sobreentrenar un algoritmo de aprendizaje con unos ciertos datos para los que se conoce el resultado deseado.
- El algoritmo de aprendizaje debe alcanzar un estado en el que será capaz de predecir el resultado en otros casos a partir de lo aprendido con los datos de entrenamiento, generalizando para poder resolver situaciones distintas a las acaecidas durante el entrenamiento.
- Cuando un sistema se entrena demasiado (se sobreentrena) o se entrena con datos extraños, el algoritmo de aprendizaje puede quedar ajustado a unas características muy específicas de los datos de entrenamiento que no tienen relación con la función objetivo.



KSCHOOL

Matriz de confusión

- Observado vs. Predicho.

		Actual class	
		Cat	**Non-cat**
Predicted class	**Cat**	5 True Positives	2 False Positives
	Non-cat	3 False Negatives	17 True Negatives

KSCHOOL

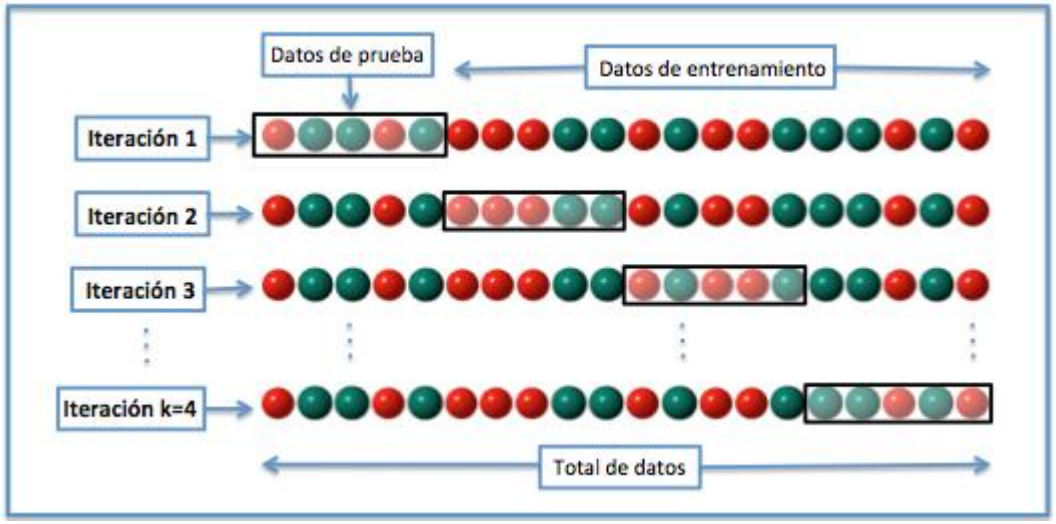
Matriz de confusión

- Observado vs. Predicho.

		True condition	
	Total population	Condition positive	Condition negative
Predicted condition	Predicted condition positive	**True positive**, Power	**False positive**, Type I error
	Predicted condition negative	**False negative**, Type II error	**True negative**

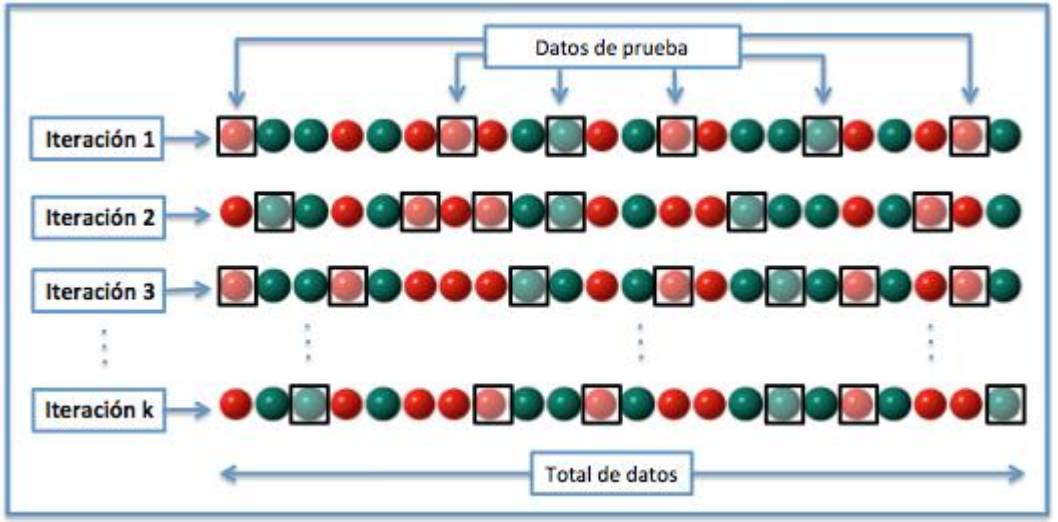
K-fold Cross Validation

- En la *K-fold cross-validation* los datos de muestra se dividen en K subconjuntos. Uno de los subconjuntos se utiliza como datos de prueba y el resto (K-1) como datos de entrenamiento. El proceso de validación cruzada es repetido durante k iteraciones, con cada uno de los posibles subconjuntos de datos de prueba.
- Este método es muy preciso puesto que evaluamos a partir de K combinaciones de datos de entrenamiento y de prueba, pero aun así tiene una desventaja, y es que, a diferencia del método de retención, es lento desde el punto de vista computacional.



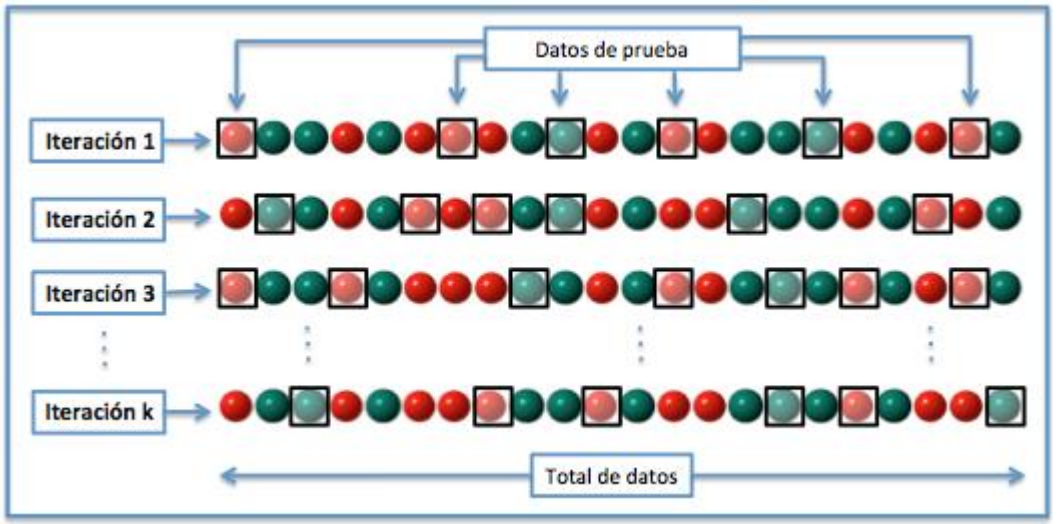
Random Cross Validation

- Este método consiste al dividir aleatoriamente el conjunto de datos de entrenamiento y el conjunto de datos de prueba. Para cada división la función de aproximación se ajusta a partir de los datos de entrenamiento y calcula los valores de salida para el conjunto de datos de prueba.
- La ventaja de este método es que la división de datos entrenamiento-prueba no depende del número de iteraciones. Pero, en cambio, con este método hay algunas muestras que quedan sin evaluar y otras que se evalúan más de una vez, es decir, los subconjuntos de prueba y entrenamiento se pueden solapar.

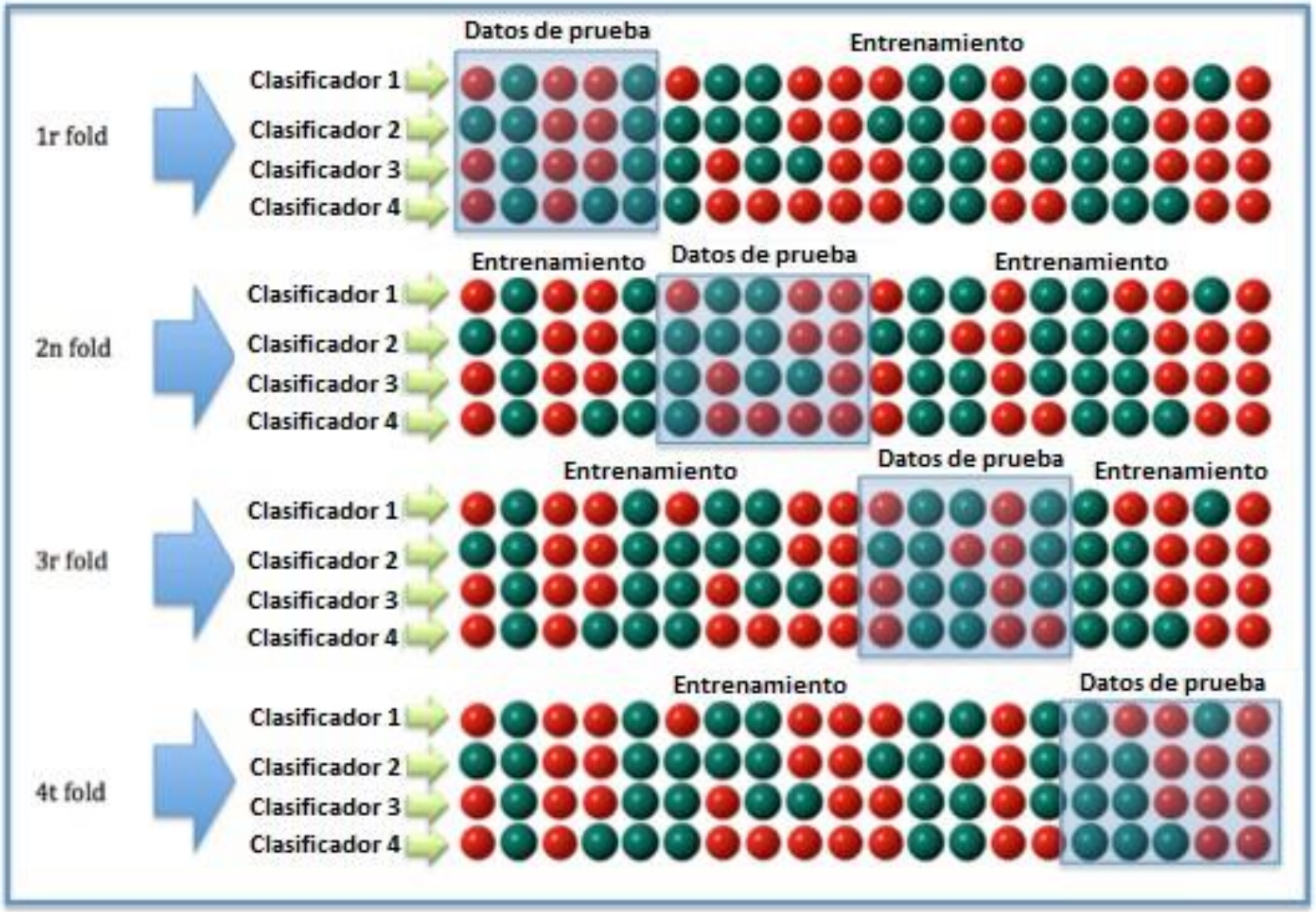


Leave-one-out cross-validation

- La Leave-one-out cross-validation implica separar los datos de forma que para cada iteración tengamos una sola muestra para los datos de prueba y todo el resto conformando los datos de entrenamiento.
- La evaluación viene dada por el error, y en este tipo de validación cruzada el error es muy bajo, pero en cambio, a nivel computacional es muy costoso, puesto que se tienen que realizar un elevado número de iteraciones, tantas como N muestras tengamos y para cada una analizar los datos tanto de entrenamiento como de prueba.

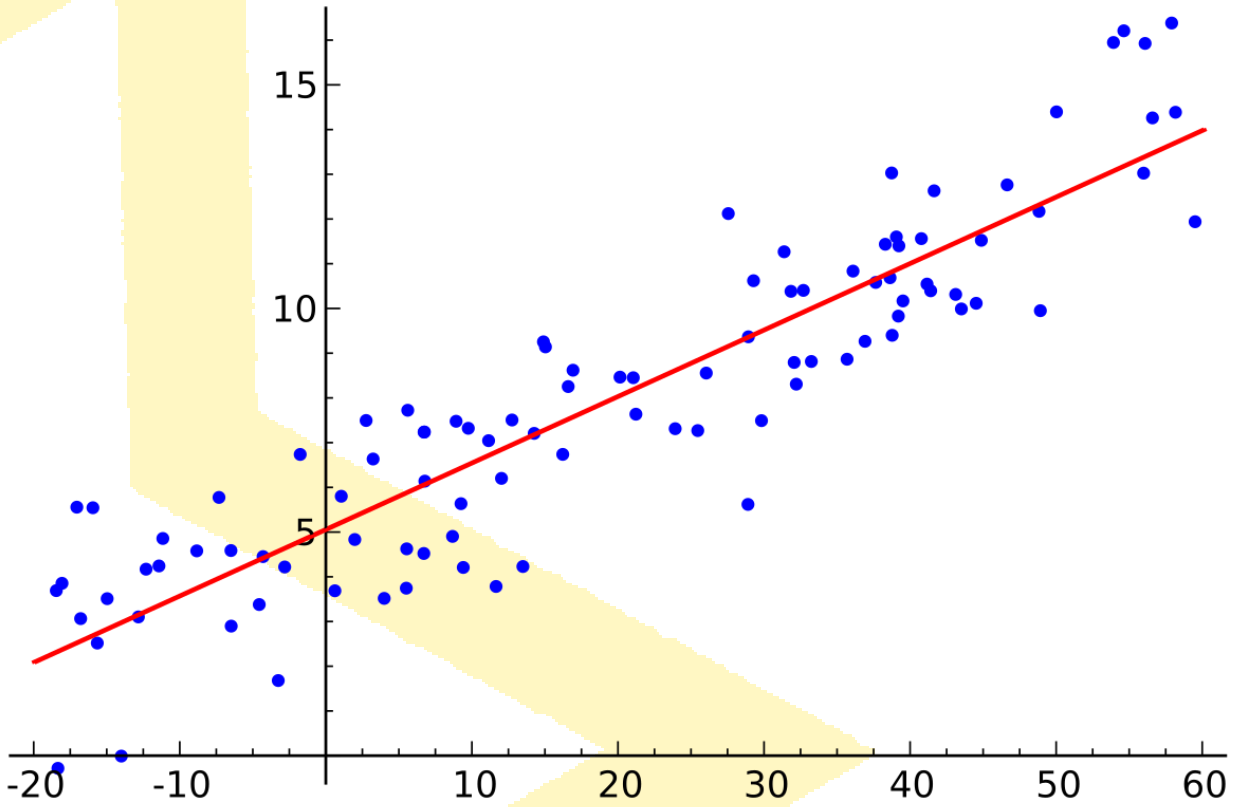


Cross-Validation ejemplo



KSCHOOL

La famosa, simple pero poderosa regresión lineal…



Linear regression

- Analiza la relación entre una variable de respuesta (a menudo llamada y) y una o más variables y sus interacciones (a menudo llamadas x o variables explicativas).
- Es uno de los modelos estadísticos más básicos que existen, sus resultados pueden ser interpretados por casi todos y ha existido desde el siglo XIX. Esto es precisamente lo que hace que la regresión lineal sea tan popular. Es simple.
- Se predice que seguirá siendo el ¡Utilizado en el año 2118!

KSCHOOL

Linear regression

- El modelo busca como objetivo hallar la siguiente ecuación

$$y = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \cdots + \beta_n x_n$$

- Donde $\beta_i \in \mathbb{R}$ para todo $i \in 1, \ldots, n$.
- En notación matricial el modelo sería

$$Y = X\beta + \varepsilon$$

- Donde ε se considera el error, y se asume que es un vector con distrución $N(0,1)$

¿Cómo sabemos si nuestro modelo está bien ajustado?

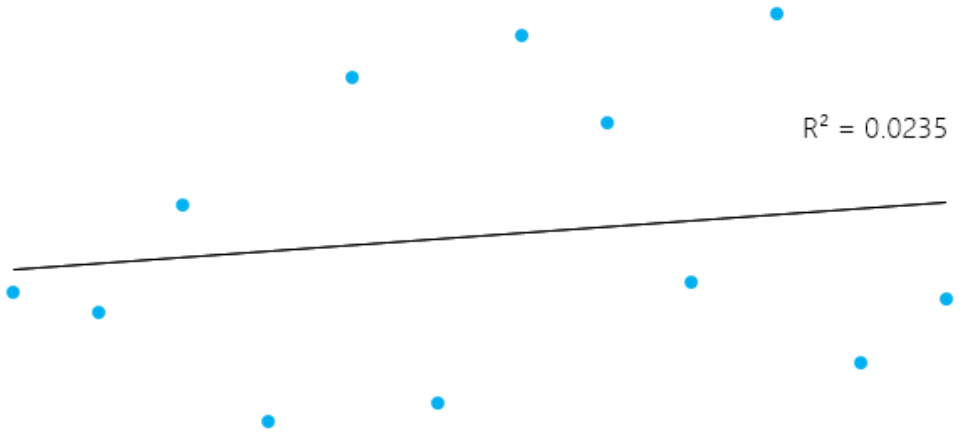
- Una medida muy utilizada para probar qué tan bueno es su modelo es el coeficiente de determinación o R^2, como sigue:

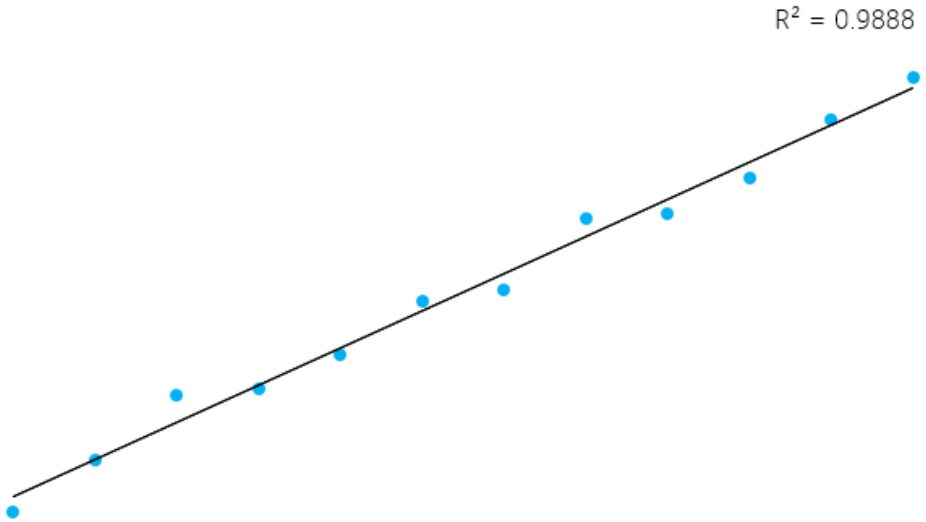
$$R^2 = \frac{Explained\ Variation\ of\ the\ model}{Total\ Variation\ of\ the\ model}$$

- Esto puede parecer un poco complicado, pero en general, para los modelos que se ajustan bien a los datos, R^2 está cerca de 1.
- Los modelos que se ajustan mal a los datos tienen R^2 cerca de 0.

KSCHOOL

El famoso R^2





KSCHOOL

El famoso R^2

```
Call:
lm(formula = height ~ age + no_siblings, data = ageandheight)

Residuals:
     Min       1Q   Median       3Q      Max
-0.28029 -0.22490 -0.02219  0.14418  0.48350

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept) 64.95872    0.55752 116.515 1.28e-15 ***
age          0.63516    0.02254  28.180 4.34e-10 ***
no_siblings -0.01137    0.05893  -0.193    0.851
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 0.2693 on 9 degrees of freedom
Multiple R-squared:  0.9888,    Adjusted R-squared:  0.9863
F-statistic: 397.7 on 2 and 9 DF,  p-value: 1.658e-09
```

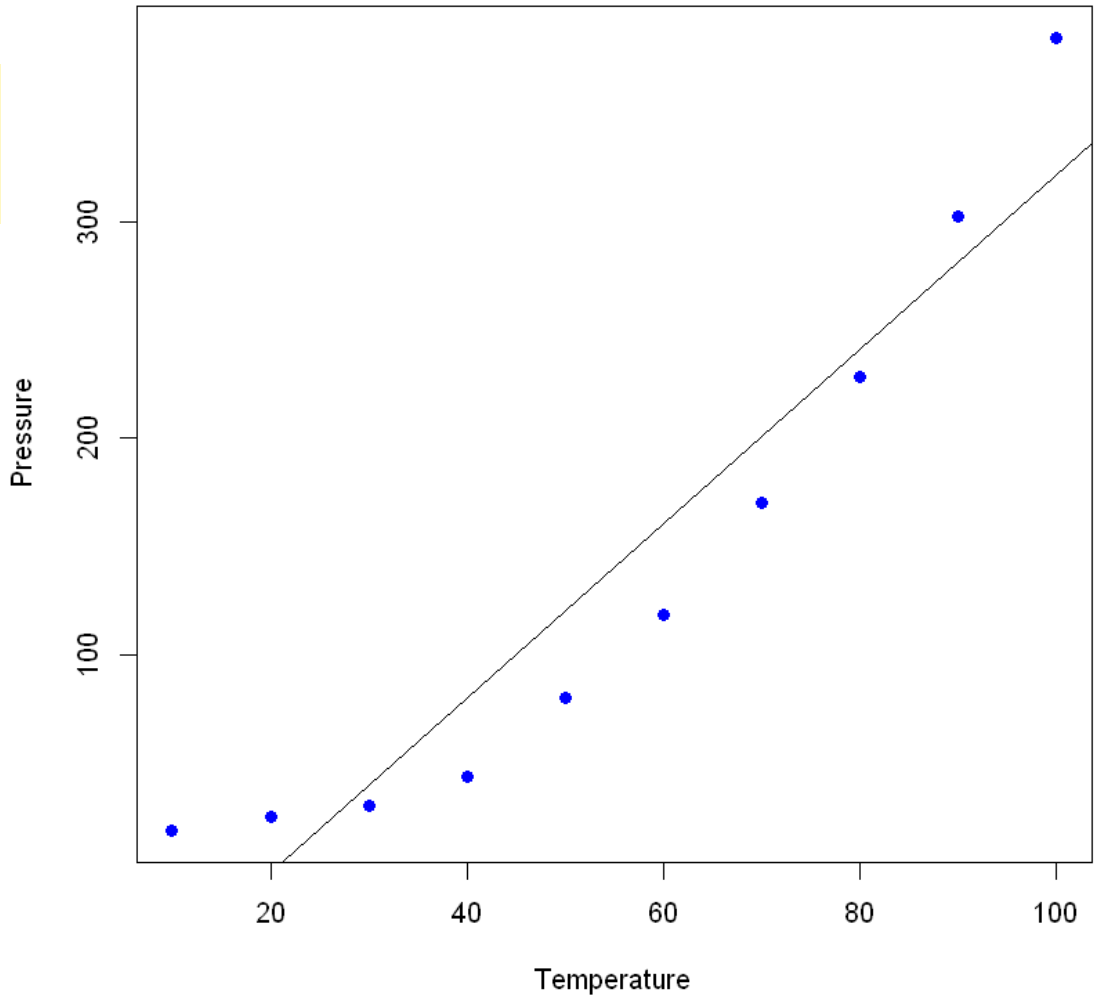
- En el rectángulo azul, observe que hay dos R^2 diferentes, uno múltiple y uno ajustado. El múltiplo es el R^2 que viste anteriormente.
- Un problema con este R^2 es que no puede disminuir a medida que agrega más variables independientes a su modelo, seguirá aumentando a medida que haga que el modelo sea más complejo, incluso si estas variables no agregan nada a sus predicciones.

Residuales

- Puedes tener un R^2 bastante bueno en tu modelo, pero no nos apresuremos a sacar conclusiones aquí.
- Veamos un ejemplo...

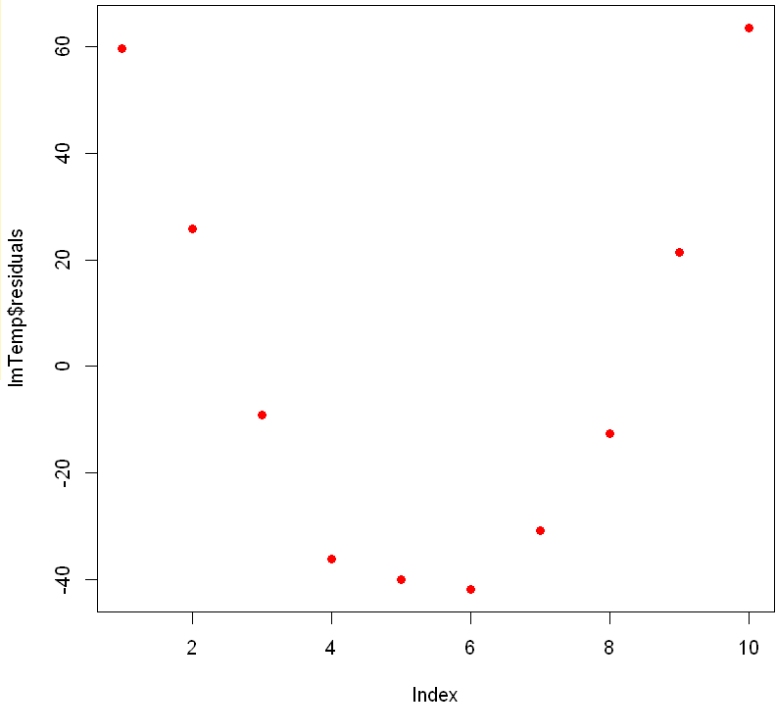
KSCHOOL

Residuales



Residuales

- Idealmente, cuando grafiques los residuos, deberían parecer aleatorios. De lo contrario, significa que tal vez hay un patrón oculto que el modelo lineal no está considerando.



Puntos influyentes (outliers)

- En sus datos, puede tener puntos influyentes que pueden sesgar su modelo, a veces innecesariamente. Piense en un error en la entrada de datos y en lugar de escribir "2.3", el valor fue "23".
- El tipo más común de punto influyente son los valores atípicos, que son puntos de datos donde la respuesta observada no parece seguir el patrón establecido por el resto de los datos.
- Puede detectar puntos influyentes observando el objeto que contiene el modelo lineal, utilizando la distancia de cook.
- Esta distancia se calcula con R con la función cooks.distance y analizando que observaciones tienen una distancia de Cook mayor a 1.

KSCHOOL

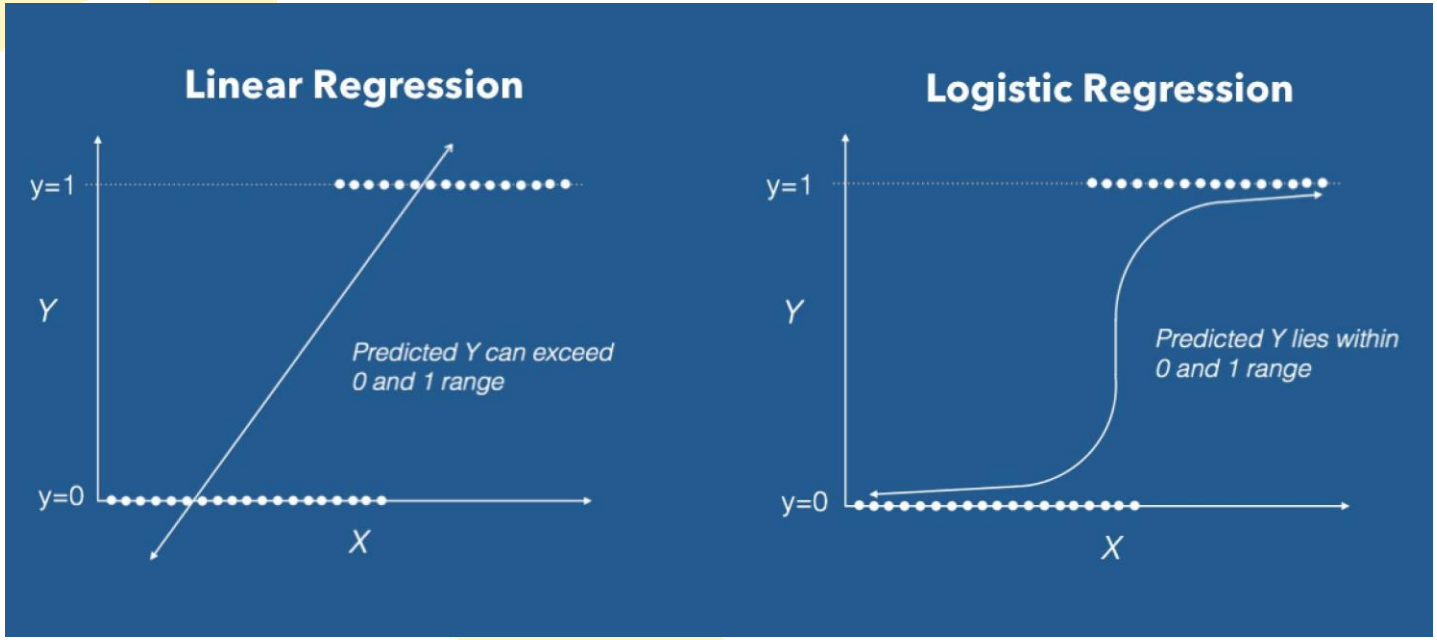
Let's do it...

Conclusiones

- La regresión lineal es un excelente modelo, ha prevalecido durante años.
- Poderoso y simple a la vez.
- Aprendimos a comprender qué hay detrás de este simple modelo estadístico y cómo puede modificarlo según sus necesidades.

KSCHOOL

Logistic Regression, ¿verdadero o falso?



Logistic Regression

- Nuestro objetivo es conseguir una función que modele la probabilidad de que un valor pertenezca a una de las dos categorías, cero o uno, partiendo de la ecuación para la regresión lineal.
- Teniendo esto en cuenta, el rango de nuestra función objetivo deberá estar entre 0 y 1. Para ello tendremos que hacer algunas transformaciones sobre la función de la regresión lineal, y utilizar el concepto de Odds.
- Según esto, podemos hacer nuestro primer movimiento, igualando la ecuación de regresión lineal a una probabilidad:

$$\beta_0 + \beta_1 x_1 + \beta_2 x_2 + \cdots + \beta_n x_n$$

- Función logit

$$ln\left(\frac{p}{1-p}\right)$$

- Función sigmode

$$p = \frac{1}{1 + e^{-(\beta_0 + \beta_1 x_1 + \beta_2 x_2 + \cdots + \beta_n x_n)}}$$

Cutoff o umbral de decisión

- La función que hemos calculado devuelve una estimación de una probabilidad. Para poder utilizarla como clasificador, hay que definir un límite de decisión. En función de si el valor devuelto por la función es mayor o menor que ese límite, clasificaremos el evento.
- Para poder utilizar este modelo, debemos calcular los coeficientes de la ecuación de regresión lineal.
- Tendremos entonces que

$$\hat{y} = \begin{cases} 1, P(\hat{y} = 1 | x_1, \dots, x_n) \geq \varepsilon \\ 0, P(\hat{y} = 1 | x_1, \dots, x_n) < \varepsilon \end{cases}$$

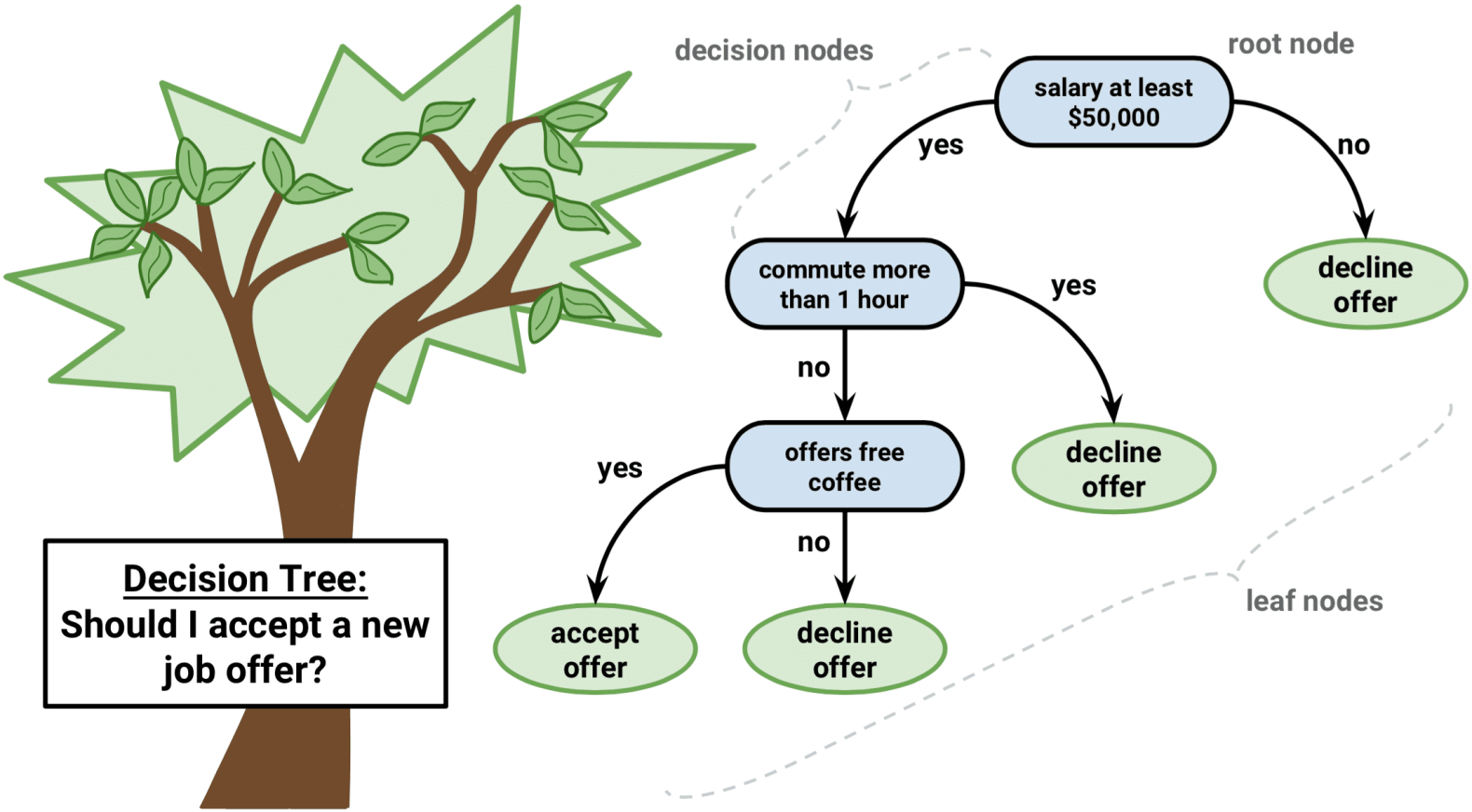
KSCHOOL

Let's do it...

Conclusión

- Así que este es el final de este tutorial de R sobre cómo construir modelos de regresión logística usando la función glm () y configurando la familia como binomial.
- glm () no asume una relación lineal entre variables dependientes e independientes.
- Sin embargo, asume una relación lineal entre la función de enlace y las variables independientes en el modelo logit.

Decision Trees o como un árbol puede clasificar eventos…



Decision Trees Motivación

- Imaginemos que estás jugando un juego de 20 preguntas. Tu oponente ha elegido un tema en secreto, y debes averiguar qué eligió. En cada turno, puedes hacer una pregunta de sí o no, y su oponente debe responder con sinceridad. ¿Cómo descubres el secreto en la menor cantidad de preguntas?
- Debería ser obvio que algunas preguntas son mejores que otras. Por ejemplo, preguntando "¿Puede volar?" ya que es probable que su primera pregunta sea infructuosa, mientras que preguntar "¿Está vivo?" Es un poco más útil. Intuitivamente, queremos que cada pregunta reduzca significativamente el espacio de posibles secretos, lo que eventualmente lo llevará a su respuesta.

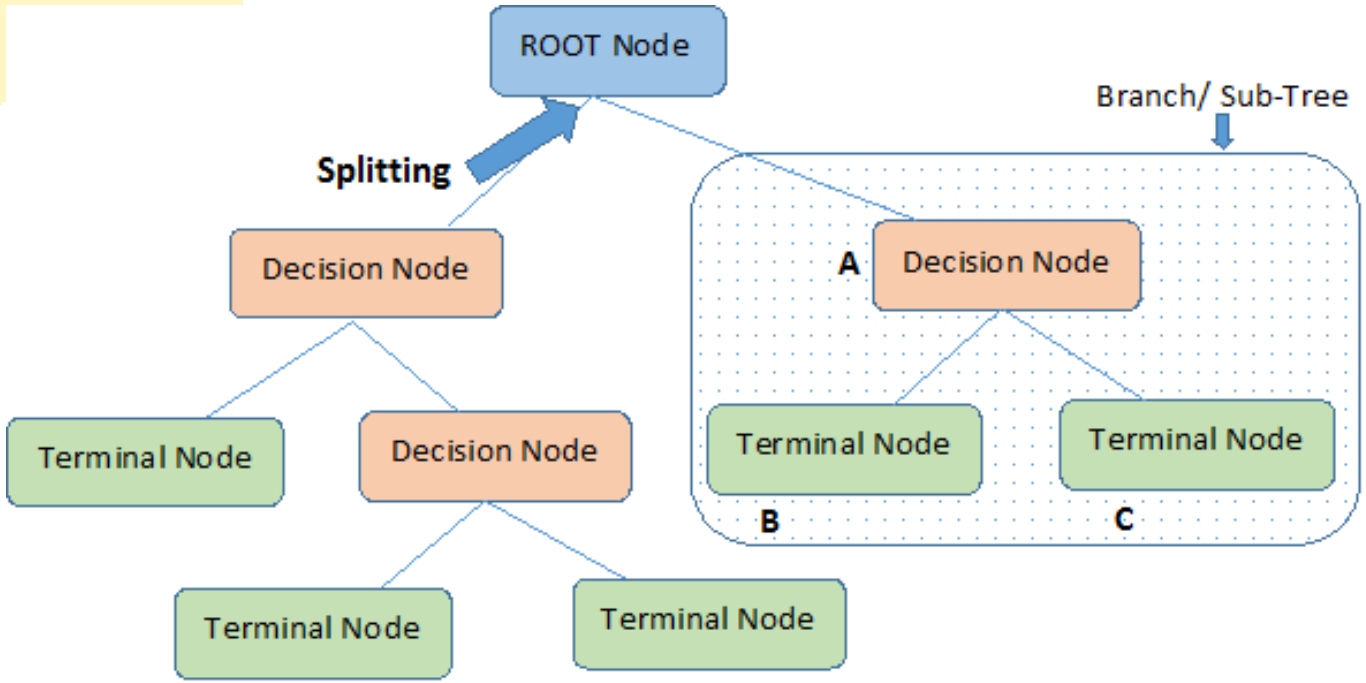
Decision Trees Motivación

- Esa es la idea básica detrás de los árboles de decisión. En cada punto, considera un conjunto de preguntas que pueden particionar tu conjunto de datos.
- Elige la pregunta que proporciona la mejor división y nuevamente encuentra las mejores preguntas para las particiones.
- Simplemente puedes tomar un valor y "lanzarlo" por el árbol. Las preguntas te guiarán a su clase apropiada.

Decision Trees Intro

- El árbol de decisión es un tipo de algoritmo de aprendizaje supervisado que se puede usar tanto en problemas de regresión como de clasificación.
- Funciona para variables de entrada y salida tanto categóricas como continuas.

Decision Trees Intro



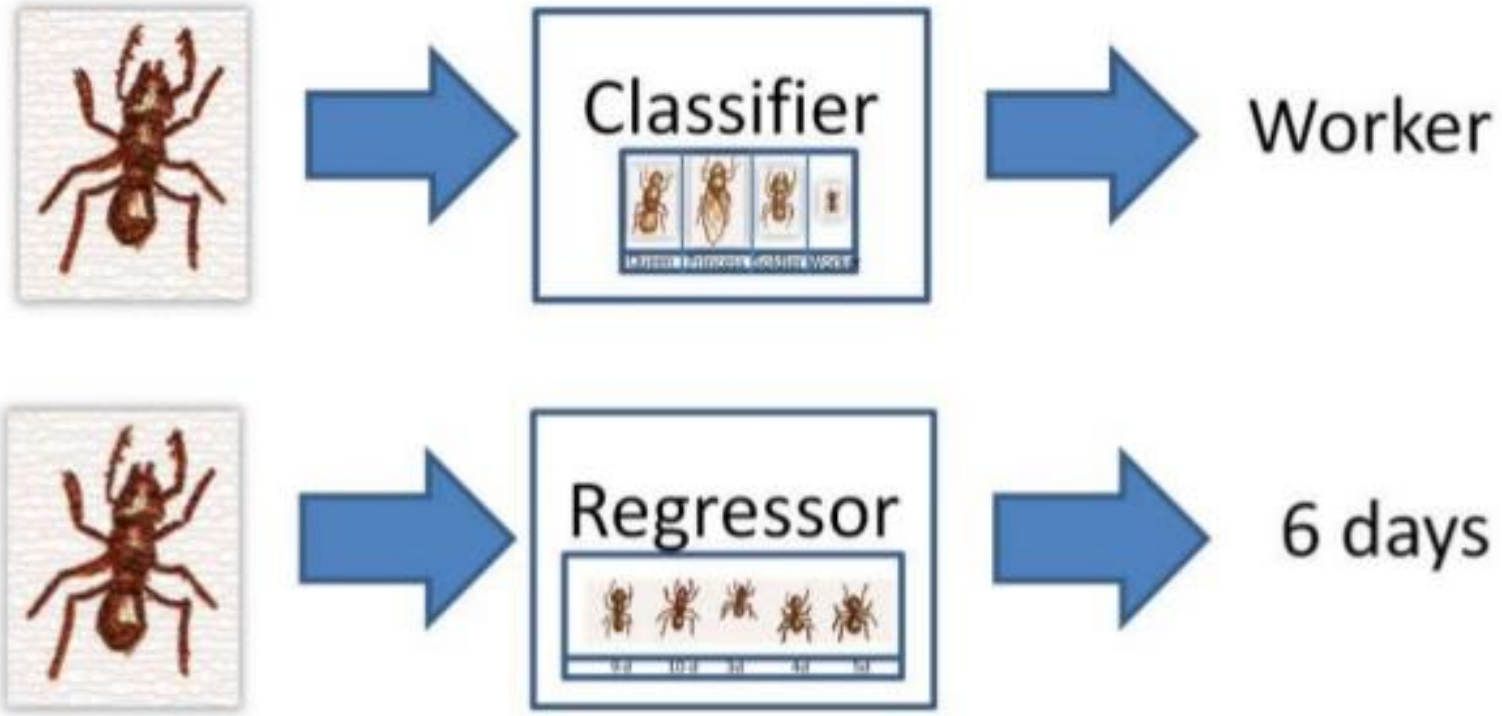
Note:- A is parent node of B and C.

Decision Trees Intro

- Identifiquemos terminologías importantes en el Árbol de decisiones, mirando la imagen de anterior.
- El nodo raíz (Root node) representa a toda la población o muestra. Además se divide en dos o más conjuntos homogéneos.
- El splitting es un proceso de división de un nodo en dos o más subnodos.
- Cuando un subnodo se divide en subnodos adicionales, se denomina nodo de decisión (Decision node).
- Los nodos que no se dividen se denominan nodo terminal (terminal node) o hoja (leaf).
- Cuando elimina subnodos de un nodo de decisión, este proceso se denomina Poda (Pruning). Lo opuesto a la poda es la splitting.
- Una subsección de un árbol entero se llama Rama (Branch).
- Un nodo, que se divide en subnodos, se denomina nodo padre (parent node) de los subnodos; mientras que los subnodos se denominan hijos (child) del nodo padre.

Tipos de Decision Trees

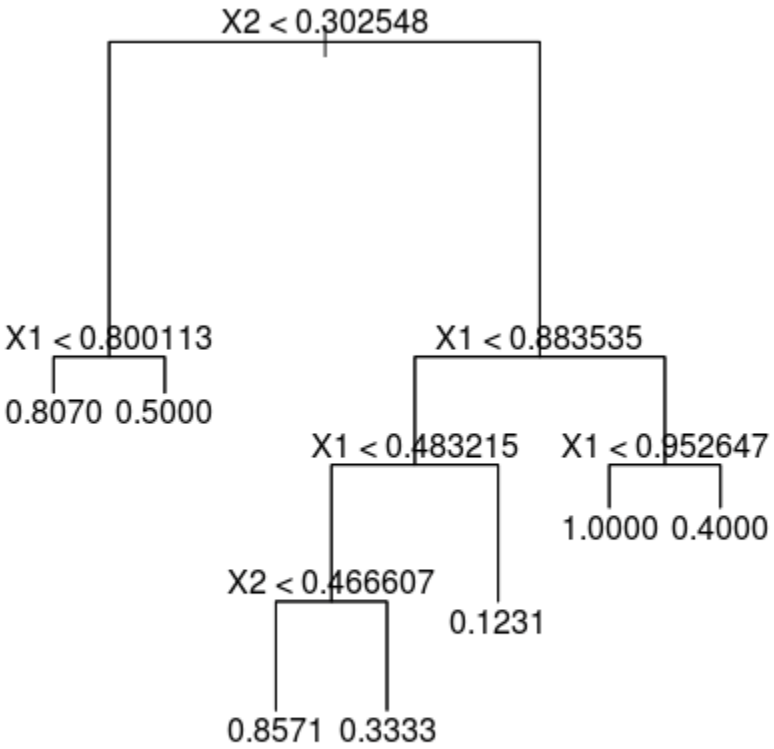
Classification vs. Regression

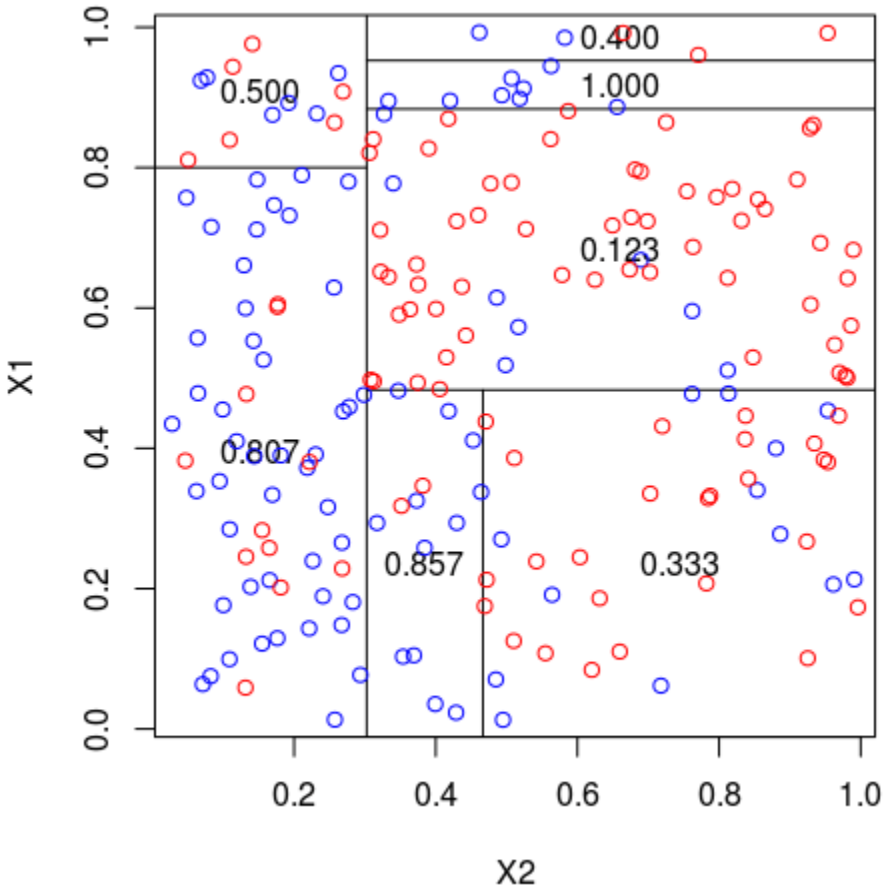


Regression Decision Trees

- Echemos un vistazo a la imagen de la siguiente diapositiva, que ayuda a visualizar la naturaleza de la partición realizada por un árbol de regresión.
- Se observa un árbol de regresión que se ajusta a un conjunto de datos aleatorio.
- Ambas visualizaciones muestran una serie de reglas de división (splitting rules), que comienzan en la parte superior del árbol.
- Observe que cada división del dominio está alineada con uno de los ejes de las variables.
- El concepto de división paralela de ejes se generaliza directamente a dimensiones superiores a dos. Esto es, para un conjunto de características de tamaño p, un subconjunto de Rp, el espacio se divide en M regiones , Rm, cada una de las cuales es un "hiperbloque" p-dimensional.

Regression Decision Trees





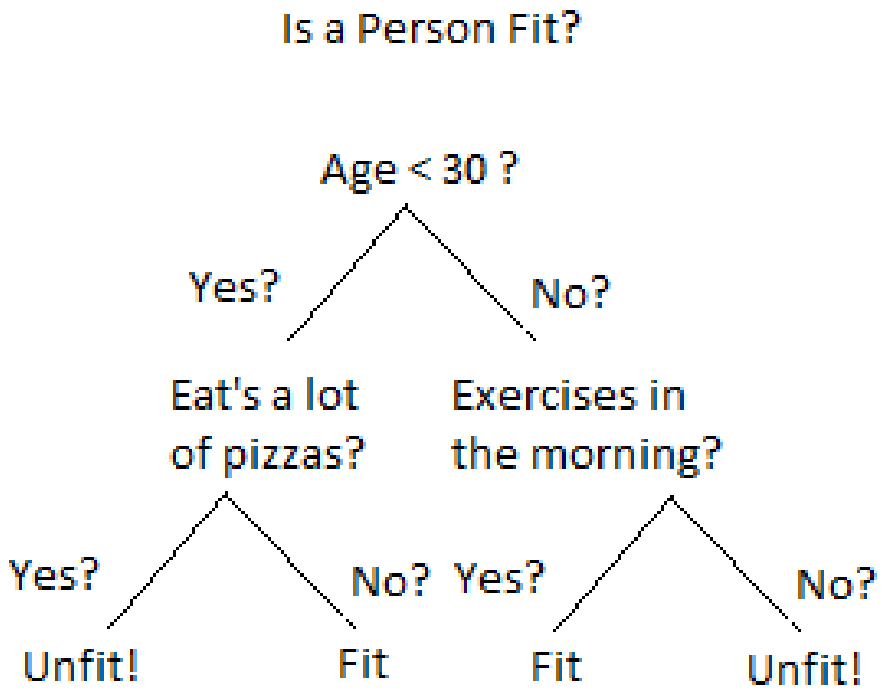
Regression Decision Trees

- Para construir un árbol de regresión, primero se utiliza la división binaria recursiva para hacer crecer un árbol grande en los datos de entrenamiento, deteniéndose solo cuando cada nodo terminal tenga menos de un número mínimo de observaciones.
- La división binaria recursiva es un algoritmo codicioso y de arriba hacia abajo que se utiliza para minimizar la suma de cuadrados residual (RSS), una medida de error que también se usa en la configuración de regresión lineal. El RSS, en el caso de un espacio de características particionado con M particiones está dado por:

$$RSS = \sum_{m=1}^{M} \sum_{i \in R_m} \left(y_i - \hat{y}_{R_m}\right)^2$$

KSCHOOL

Classification Decision Trees



Classification Decision Trees

- Un árbol de clasificación es muy similar a un árbol de regresión, excepto que se usa para predecir una respuesta cualitativa en lugar de cuantitativa.
- Recuerde que para un árbol de regresión, la respuesta predicha para una observación viene dada por la respuesta media de las observaciones de entrenamiento que pertenecen al mismo nodo terminal. En contraste, un árbol de clasificación predice que cada observación pertenece a la clase de observaciones de entrenamiento más comunes en la región a la que pertenece.
- En la interpretación de los resultados de un árbol de clasificación, nos interesa saber no solo la predicción de clase correspondiente a una región de nodo terminal particular, sino también las proporciones de clase entre las observaciones de entrenamiento que caen dentro de esa región.

Ventajas y desventajas

- La principal ventaja de usar árboles de decisión es que son intuitivamente muy fáciles de explicar. Reflejan estrechamente la toma de decisiones humanas en comparación con otros enfoques de regresión y clasificación. Pueden mostrarse gráficamente y pueden manejar fácilmente predictores cualitativos sin la necesidad de crear variables ficticias.
- Sin embargo, los árboles de decisión generalmente no tienen el mismo nivel de precisión predictiva que otros modelos, ya que no son del todo robustos. Un pequeño cambio en los datos puede causar un gran cambio en el árbol estimado final.
- Al agregar muchos árboles de decisión, utilizando métodos como random forest y boosting, se puede mejorar sustancialmente el rendimiento predictivo de los árboles de decisión.

KSCHOOL

Let's do it...

Ensemble Learning

- Ensemble Learning es un tipo de técnica de aprendizaje supervisado en el que la idea básica es generar múltiples modelos en un conjunto de datos de entrenamiento y luego simplemente combinar (promediar) sus reglas de salida o su hipótesis, para generar un modelo fuerte que funcione muy bien y no se adapte a la perfección y que equilibre también la compensación sesgo-varianza.
- La idea es que, en lugar de producir un único Modelo complicado y complejo que pueda tener una gran varianza, lo que llevará a un Overfitting o que sea demasiado simple y tenga un alto sesgo que lleve al Underfitting, generaremos muchos Modelos al entrenar en Training Set. Y al final los combinas. Una técnica de este tipo es Random Forest.

Random Forest

- En Random Forests, la idea es descorrelacionar los diversos árboles que son generados y luego simplemente reducimos la Varianza en los Decision Trees promediándolos.
- Promediar los Decision Trees nos ayuda a reducir la variación y también a mejorar el rendimiento de los árboles de decisión en el conjunto de test y, finalmente, evitar el overfitting.
- La idea es construir muchos árboles de tal manera que la correlación entre los árboles sea más pequeña.
- Otra cuestión importante acerca de los Random Forest es que podemos seguir agregando más y más árboles grandes y tupidos, y eso no nos perjudicará porque al final solo vamos a promediarlos, lo que reducirá la varianza.

KSCHOOL

Random Forest

Bootstrap sampling

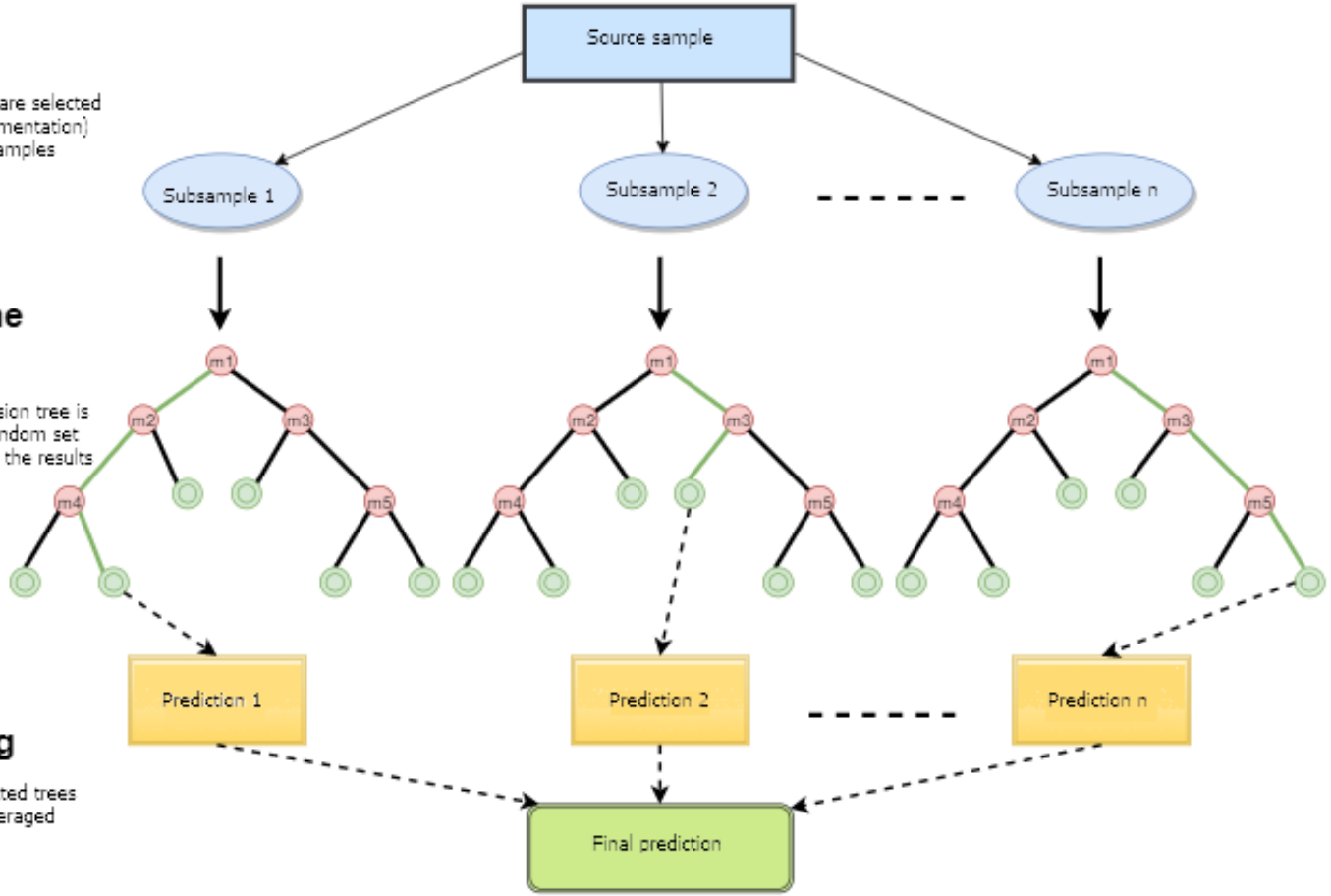
r (percentage) examples are selected (0.63 in classical implementation) in n random subsamples

Building the models

for each subsample, a decision tree is constructed based on a random set of m features (covariants), the results fall into leaves

Bootstrap aggregating

results from all constructed trees are gathered and averaged



KSCHOOL

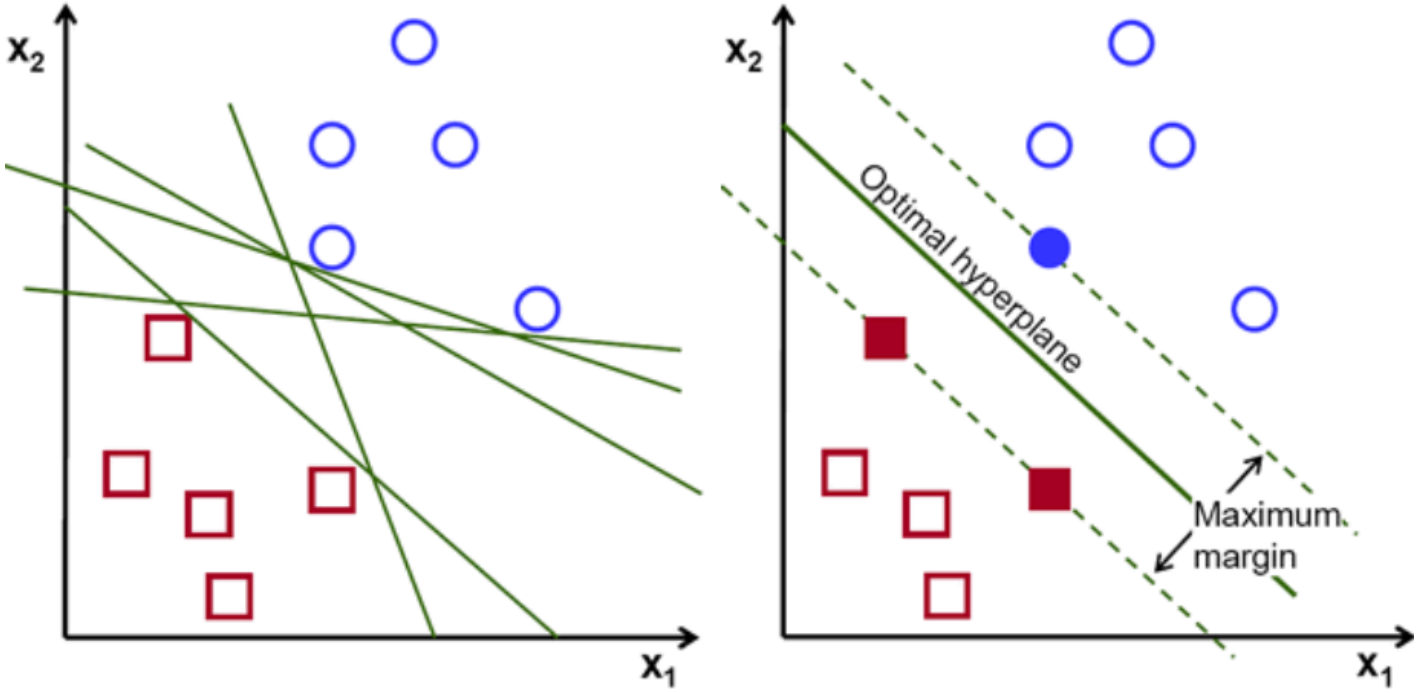
Let's do it...

Conclusión

- Es posible aplicar una técnica de modelado tan compleja y fuerte en R con una simple librería: randomForest.
- Los bosques aleatorios son una técnica muy agradable para ajustar un modelo más preciso al promediar muchos árboles de decisión, reducir la variación y evitar el problema de overfitting que puede haber en los decisión trees.
- Los decisión tres en sí mismos tienen un bajo performance, pero cuando se usan con técnicas de ensamble learning, su rendimiento predictivo se mejora considerablemente.

KSCHOOL

Support Vector Machines o como separar un conjunto datos...

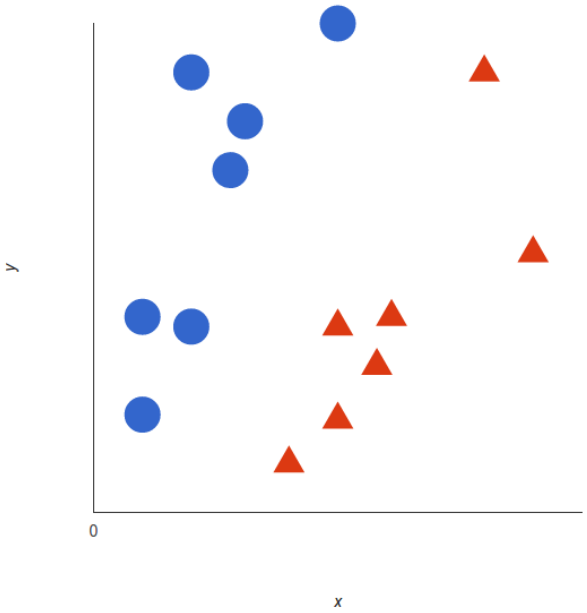


Support Vector Machines

- En el machine learning, Support Vector Machines son modelos de aprendizaje supervisados con algoritmos de aprendizaje asociados que analizan los datos utilizados para la clasificación y el análisis de regresión.
- Al igual que siempre, nos concentraremos en desarrollar la intuición en lugar de el rigor. Lo que significa esencialmente que omitiremos la mayor cantidad de matemáticas posible y desarrollaremos una fuerte intuición desde el principio de funcionamiento.

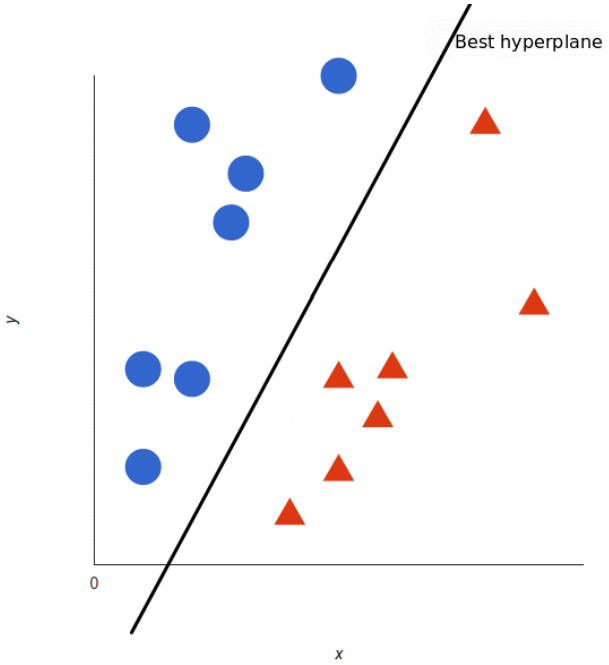
SVM linear data

- Los conceptos básicos de las SVM y como funcionan se entienden mejor con un simple ejemplo. Imaginemos que tenemos dos etiquetas: roja y azul, y nuestros datos tienen dos características: x e y.
- Queremos un clasificador que, dado un par de coordenadas (x, y), determine si es rojo o azul. Trazamos nuestros datos de entrenamiento ya etiquetados en un plano:



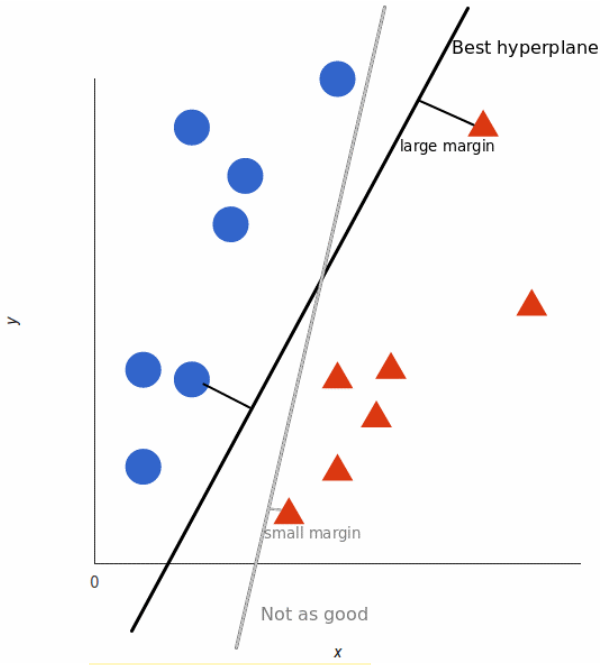
SVM linear data

- Una SVM toma estos puntos de datos y genera un hiperplano (que en dos dimensiones es simplemente una línea) que mejor separa las etiquetas. Esta línea es el límite de la decisión (decisión boundary): cualquier cosa que caiga a un lado de ella, la clasificaremos como azul, y cualquier cosa que caiga al otro como roja.



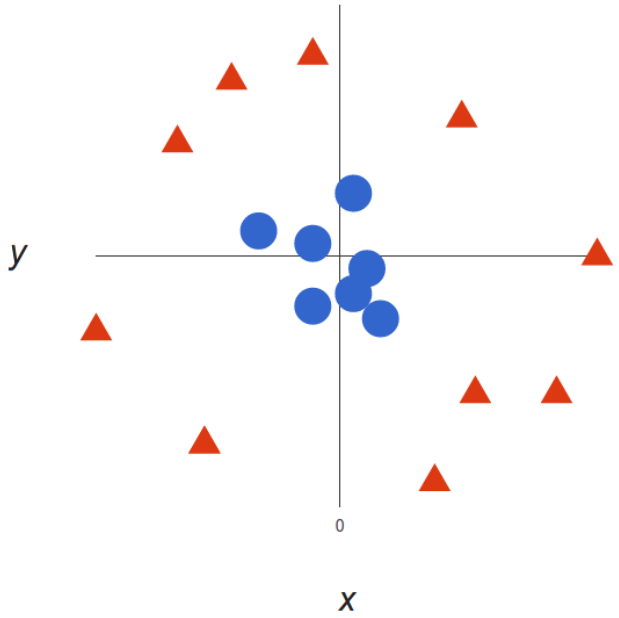
SVM linear data

- Pero, ¿cuál es exactamente el mejor hiperplano? Para las SVM, es aquel que maximiza los las distancias de ambas etiquetas. En otras palabras: el hiperplano (recuerde que es una línea en este caso) cuya distancia al elemento más cercano de cada etiqueta es la más grande.



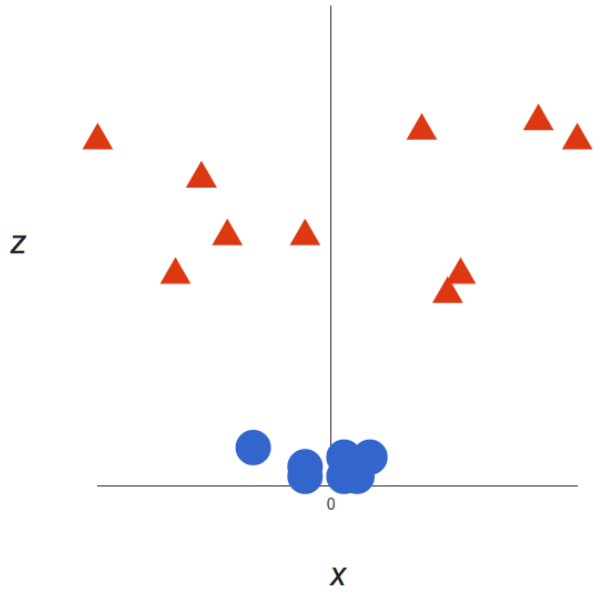
SVM non linear data

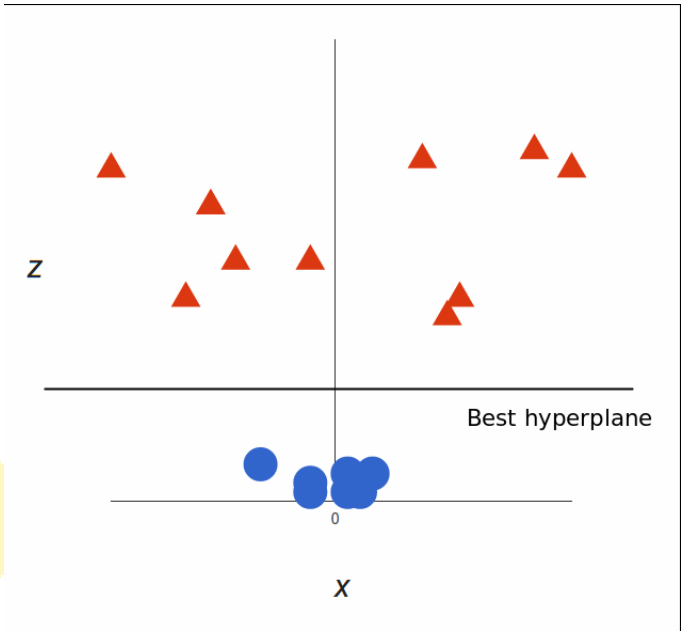
- El ejemplo anterior es fácil de entender, claramente, los datos se podían separar linealmente. Podríamos dibujar una línea recta para separar el rojo y el azul.
- Lamentablemente, por lo general, las cosas no son tan simples siempre. Veamos el siguiente caso



SVM non linear data

- Es más que claro que no hay un límite de decisión lineal (una sola línea recta que separa ambas etiquetas). Sin embargo, los vectores están claramente separados y parece que debería ser fácil separarlos.
- Entonces, esto es lo que haremos: agregaremos una tercera dimensión.
- Esto nos dará un espacio tridimensional. Tomando una porción de ese espacio, se ve así:

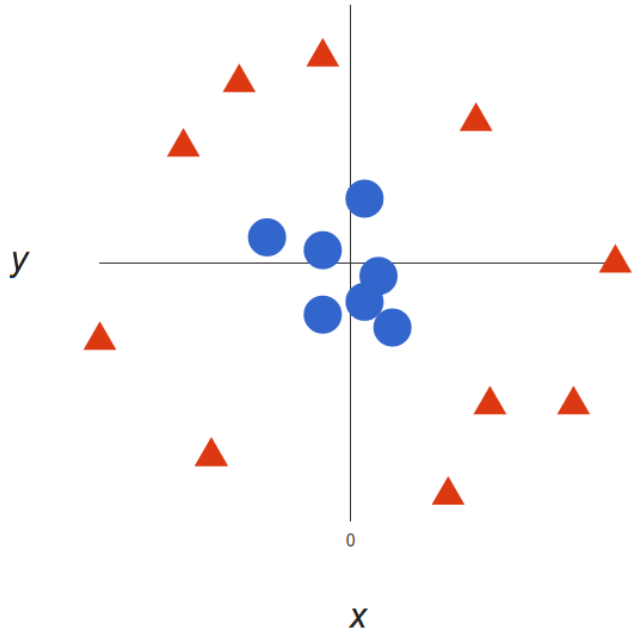




SVM non linear data

- Tenga en cuenta que dado que ahora estamos en tres dimensiones, el hiperplano es un plano paralelo al eje x en una z determinada (digamos z = 1).
- Lo que queda es mapearlo de nuevo a dos dimensiones

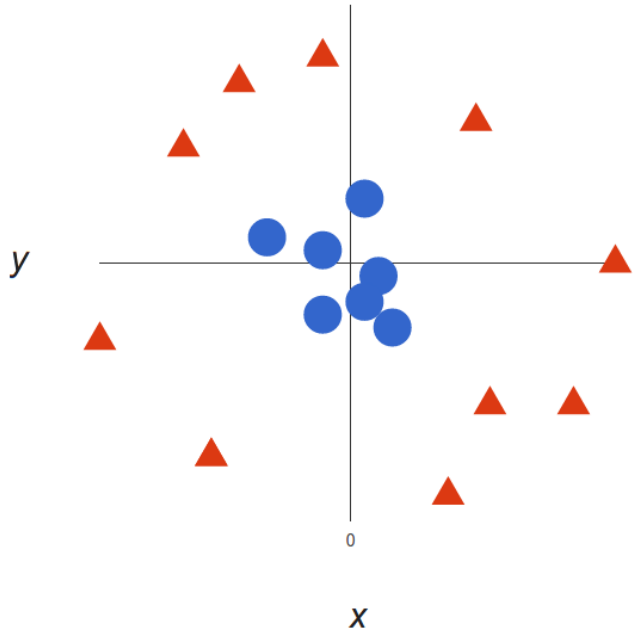
Pasando de esto...



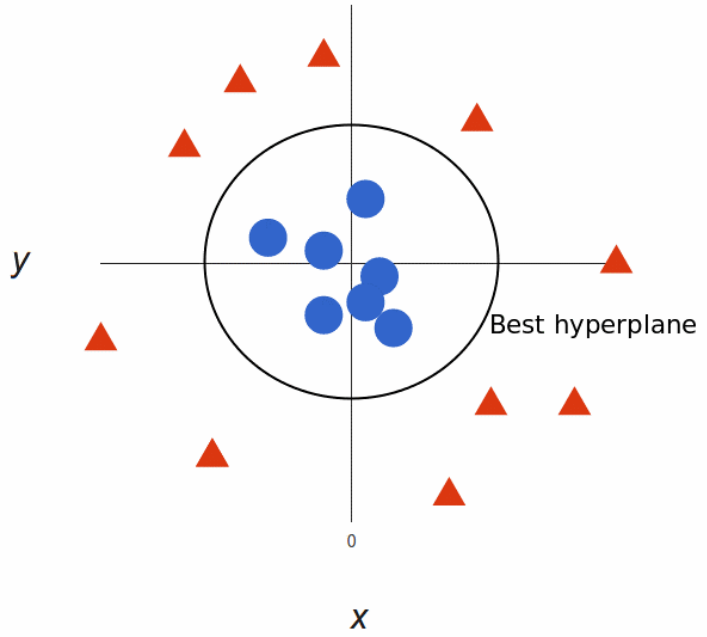
SVM non linear data

- Tenga en cuenta que dado que ahora estamos en tres dimensiones, el hiperplano es un plano paralelo al eje x en una z determinada (digamos z = 1).
- Lo que queda es mapearlo de nuevo a dos dimensiones

Pasando de esto...



¡A esto!



Kernel trick

- En el ejemplo anterior, encontramos una manera de clasificar los datos no lineales mediante la asignación inteligente de nuestro espacio a una dimensión superior.
- Sin embargo, resulta que el cálculo de esta transformación puede ser bastante costoso desde el punto de vista computacional: puede haber muchas dimensiones nuevas, cada una de las cuales posiblemente implique un cálculo complicado. Hacer esto para cada vector en el conjunto de datos puede ser una gran cantidad de trabajo, por lo que sería genial si pudiéramos encontrar una solución más "barata".
- Aquí hay un truco: SVM no necesita los vectores reales para hacer su magia, en realidad solo puede arreglárselas con los productos de puntos entre ellos. ¡Esto significa que podemos esquivar los costosos cálculos de las nuevas dimensiones! Esto es lo que hacemos en su lugar:

Kernel trick

- Imagina el nuevo espacio que queremos:

$$z = x^2 + y^2$$

- En este espacio el producto punto se definiría como sigue:
- $a \cdot b = x_a \cdot x_b + y_a \cdot y_b + z_a \cdot z_b = x_a \cdot x_b + y_a \cdot y_b + \left(x_a^2 + y_b^2\right) \cdot \left(x_b^2 + y_b^2\right)$
- Le decimos a la SVM que haga su trabajo, pero usando el nuevo producto de puntos, llamamos a esto kernel function.
- Esto se conoce como el kernel trick, que amplía el espacio de características para acomodar un límite no lineal entre las clases. Los tipos comunes de núcleos utilizados para separar datos no lineales son los núcleos polinomiales, los núcleos de base radial y los núcleos lineales (que son lo mismo que los clasificadores de vectores de soporte). Simplemente, estos núcleos transforman nuestros datos para pasar un hiperplano lineal y, por lo tanto, clasificar nuestros datos.

SVM: ventajas

- Alta dimensionalidad: SVM es una herramienta eficaz en espacios de alta dimensión, que es particularmente aplicable a la clasificación de documentos y análisis de sentimientos donde la dimensionalidad puede ser extremadamente grande.
- Eficiencia de la memoria: dado que solo un subconjunto de los puntos de entrenamiento se usa en el proceso de decisión real de asignar nuevos registros, solo estos puntos deben almacenarse en la memoria (y calcularse) al tomar decisiones.
- Versatilidad: la separación de clases es a menudo altamente no lineal. La capacidad de aplicar nuevos núcleos permite una flexibilidad sustancial para los límites de decisión, lo que lleva a un mayor rendimiento de clasificación.

SVM: desventajas

- Kernel Parameters Selection: Los SVM son muy sensibles a la elección de los parámetros del kernel. En situaciones donde el número de funciones para cada objeto excede el número de muestras de datos de entrenamiento, los SVM pueden tener un rendimiento pobre. Esto se puede ver intuitivamente como si el espacio de características de alta dimensión fuera mucho más grande que las muestras. Luego, hay vectores soporte menos efectivos en los que se admiten los hiperplanos lineales óptimos, lo que lleva a un rendimiento de clasificación más bajo a medida que se agregan nuevas muestras invisibles.

- No probabilístico: No existe una interpretación probabilística directa para la pertenencia a un grupo. Sin embargo, una métrica potencial para determinar la "efectividad" de la clasificación es qué tan lejos del límite de decisión se encuentra el nuevo punto.

KSCHOOL

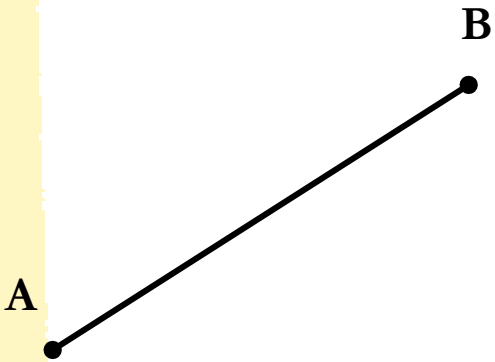
Let's do it...

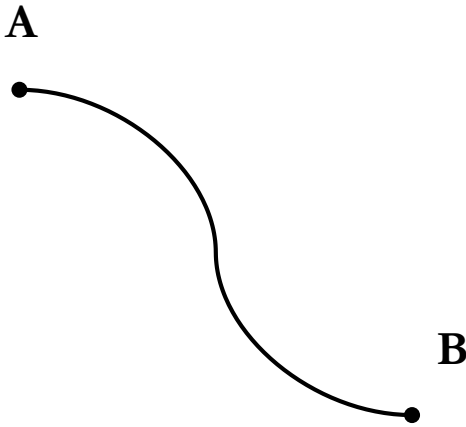
Conclusion

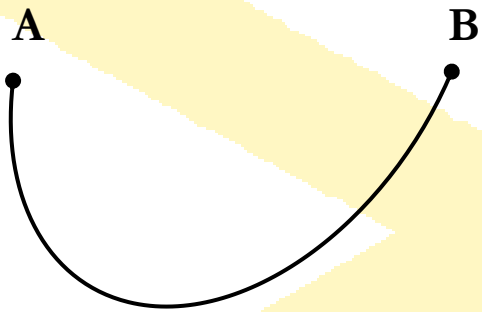
- Para resumir, las SVM son una subclase de clasificadores supervisados que intentan dividir un espacio de características en dos o más grupos. Lo logran al encontrar un medio óptimo para separar dichos grupos según sus etiquetas de clase conocidas:
  - En casos más simples, el "límite" de separación es lineal, lo que lleva a grupos que se dividen por líneas (o planos) en espacios de alta dimensión.
  - En casos más complicados (donde los grupos no están bien separados por líneas o planos), los SVM pueden realizar particiones no lineales. Esto se consigue mediante una función del núcleo.
  - En última instancia, esto los convierte en clasificadores muy sofisticados y capaces, pero a un coste usual y es que pueden ser propensos a overfitting.
- Las SVM son excelentes clasificadores para situaciones específicas cuando los grupos están claramente separados. También hacen un gran trabajo cuando sus datos no están separados linealmente. Puede transformar los datos para separarlos linealmente, o puede hacer que los SVM conviertan los datos y separen linealmente las dos clases directamente de la caja. Esta es una de las principales razones para usar SVMs.

KSCHOOL

Splines cúbicos, o como unir puntos para estimar datos







Splines

- Los splines son una forma suave y flexible de ajustar los modelos no lineales y aprender las interacciones no lineales de los datos. En la mayoría de los métodos en los que ajustamos los modelos no lineales a los datos y aprendemos las no linealidades es mediante la transformación de los datos o las variables aplicando Una transformación no lineal.
- Los splines cúbicos con knots (puntos de corte) en X son un polinomio cúbico de piece-wise con derivadas continuas hasta el orden 2 en cada knot. Tienen 1ª y 2ª derivadas continuas.
- Básicamente lo que realizan es una transformación de las variables x_i aplicando una función Base $b(x)$ y ajustando un modelo utilizando estas variables transformadas, lo que agrega no linealidades al modelo y permite que los splines se ajusten a funciones no lineales más suaves y flexibles.
- Convirtiendo la ecuación de regresión en lo que sigue:

$$f(x) = y_i = \beta_0 + \beta_1 b(x_1) + \beta_2 b(x_2) + \cdots + \beta_n b(x_n)$$

Splines

- Los splines son una forma suave y flexible de ajustar los modelos no lineales y aprender las interacciones no lineales de los datos. En la mayoría de los métodos en los que ajustamos los modelos no lineales a los datos y aprendemos las no linealidades es mediante la transformación de los datos o las variables aplicando Una transformación no lineal.
- Los splines cúbicos con knots (puntos de corte) en X son un polinomio cúbico de piece-wise con derivadas continuas hasta el orden 2 en cada knot. Tienen 1ª y 2ª derivadas continuas.
- Básicamente lo que realizan es una transformación de las variables x_i aplicando una función Base $b(x)$ y ajustando un modelo utilizando estas variables transformadas, lo que agrega no linealidades al modelo y permite que los splines se ajusten a funciones no lineales más suaves y flexibles.
- Convirtiendo la ecuación de regresión en lo que sigue:

$$f(x) = y_i = \beta_0 + \beta_1 b(x_1) + \beta_2 b(x_2) + \cdots + \beta_n b(x_n)$$

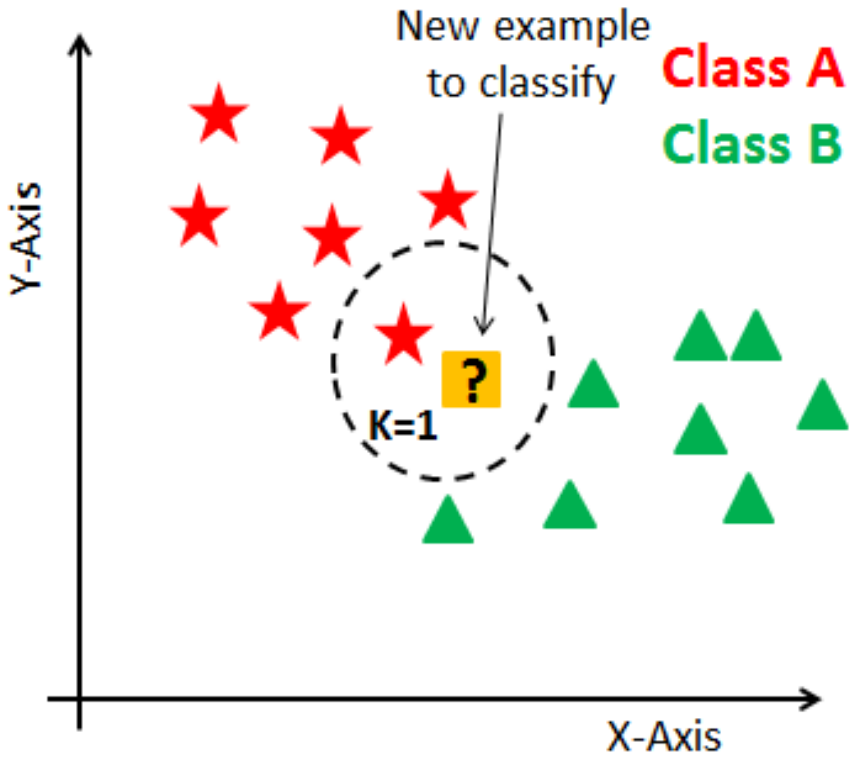
KSCHOOL

Let's do it...

Conclusión

- Hemos visto que lo splines cúbicos transforman las variables y agregan No linealidades al Modelo y son más flexibles y suaves que otras técnicas.
- Es mejor en términos de extrapolación y es más suave.
- Otras técnicas, como la regresión polinomial, son muy malas en la extrapolación y oscilan mucho una vez que se salen de los límites y se vuelven muy onduladas y fluctuantes, lo que da como resultado una alta varianza y en general, overfitting a mayores valores de grado de polinomios.
- Lo principal que debe recordar al ajustar los modelos no lineales a los datos es que necesitamos hacer algunas transformaciones a los datos o las variables para hacer que el modelo sea más flexible y más fuerte en el aprendizaje de interacciones no lineales entre las variables independientes X_i y dependiente Y.

K – Nearest Neighbors (KNN): Somos lo que no rodea.

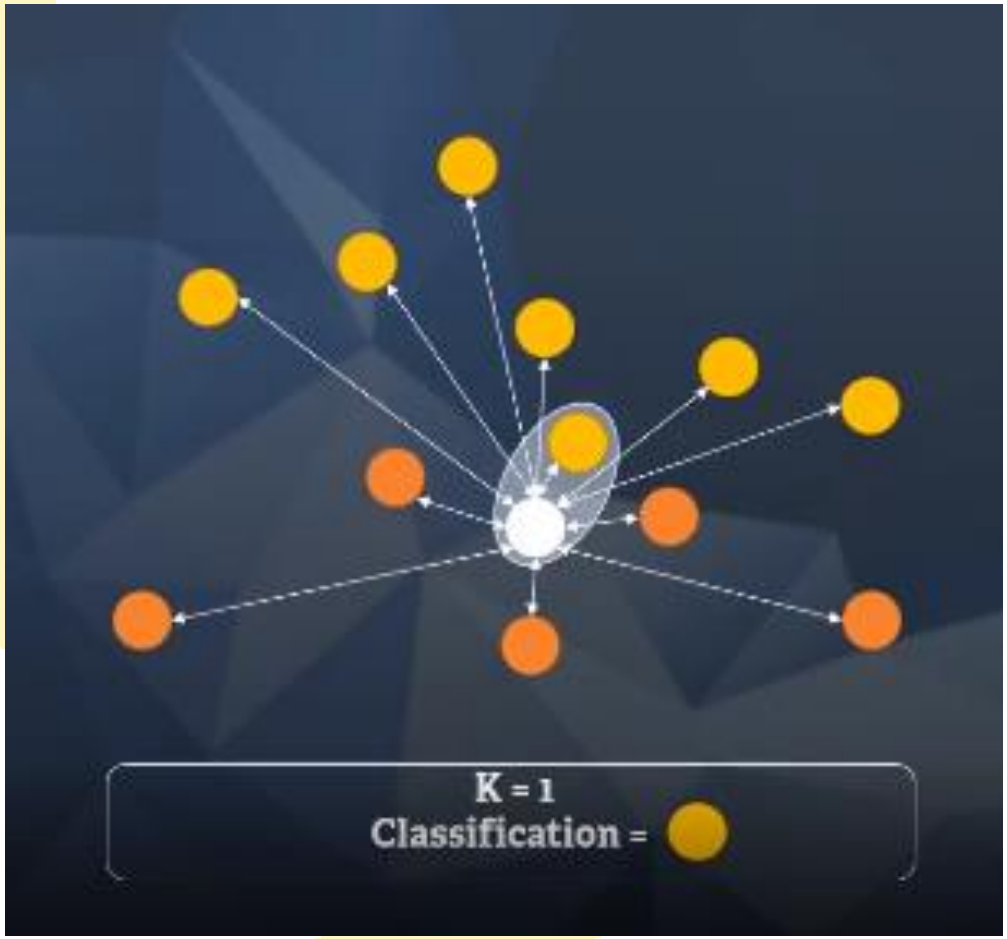


KNN: Intro

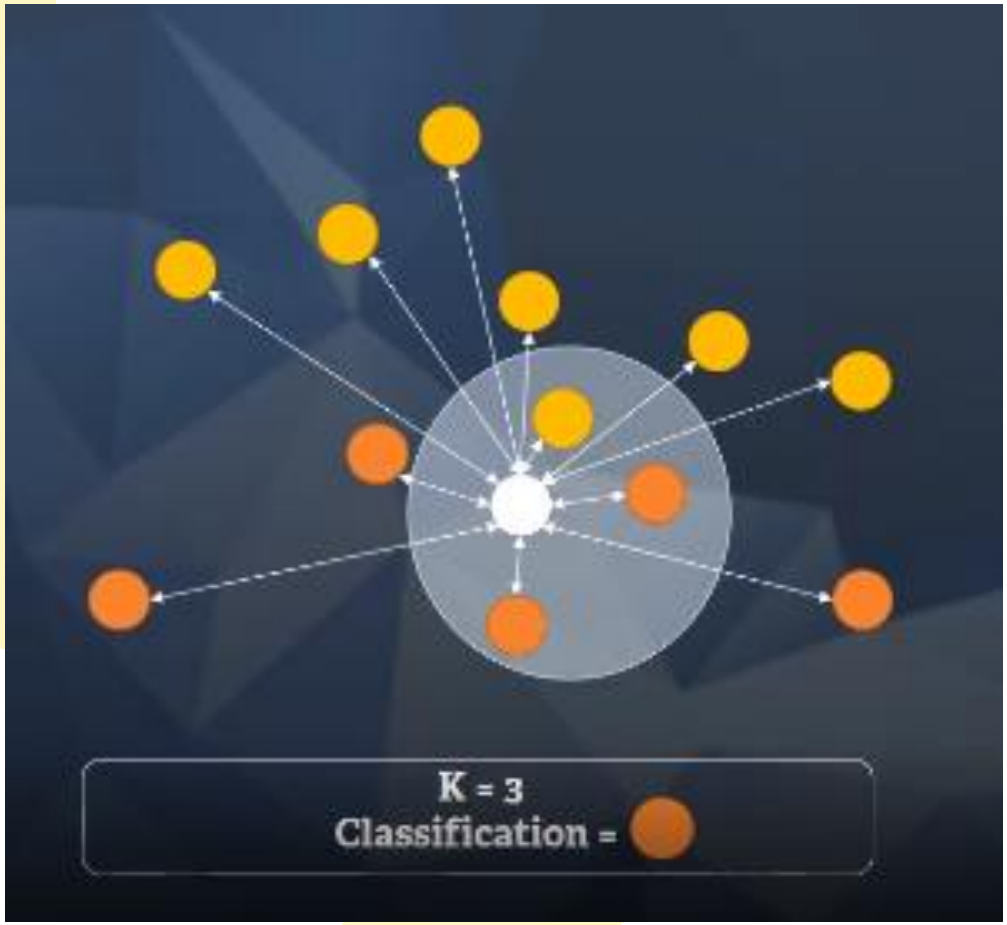
- Tenemos un conjunto de datos ya clasificados y queremos saber a qué clase pertenece otro conjunto del cual no tenemos información.
- Es uno de los algoritmos de aprendizaje automático más simples y es un ejemplo de aprendizaje basado en instancias, donde los datos nuevos se clasifican según las instancias etiquetadas y almacenadas.
- Toma la puntuación más alta de las distancias de la nueva observación a las observaciones establecidas
- Ventajas: Facilidad para interpretar la salida, tiempo de cálculo, poder predictivo.
- Desventajas: No proporciona una probabilidad, no puede usarse para variables independientes categóricas (no siempre).
- ¿puede usarse para palabras? Es decir, clasificación de texto o natural language processing (NLP).

KSCHOOL

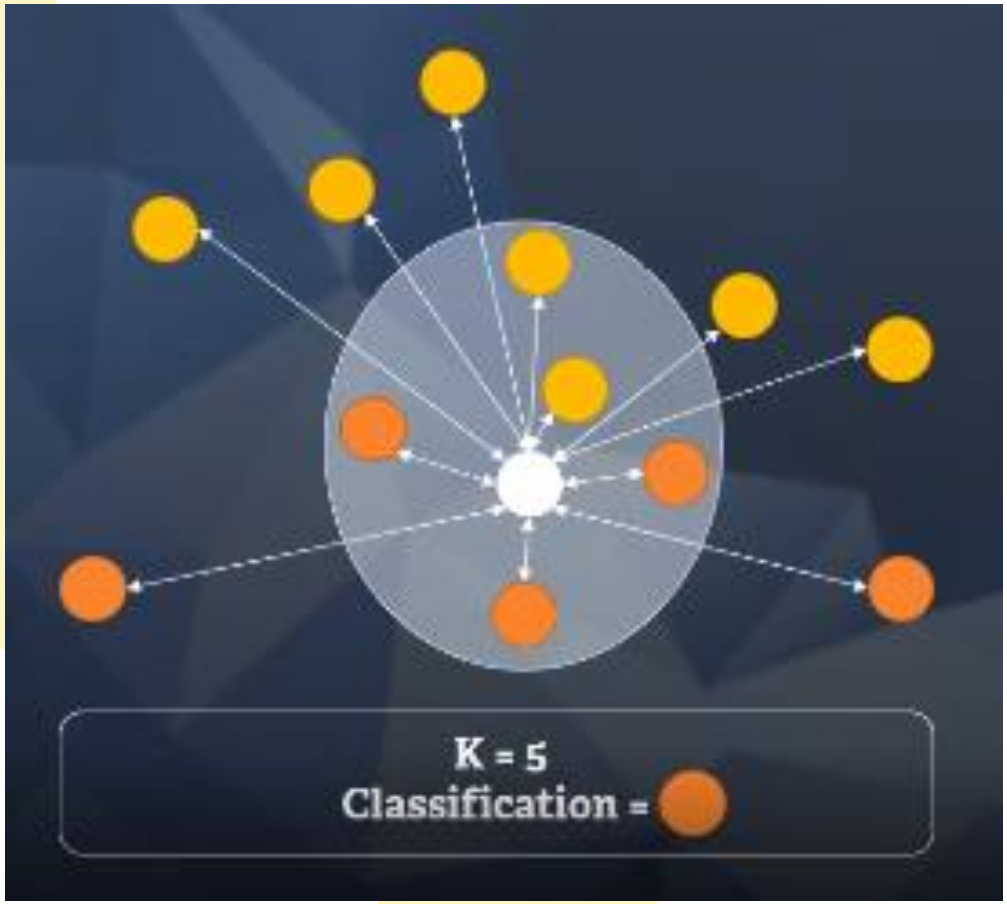
K=1



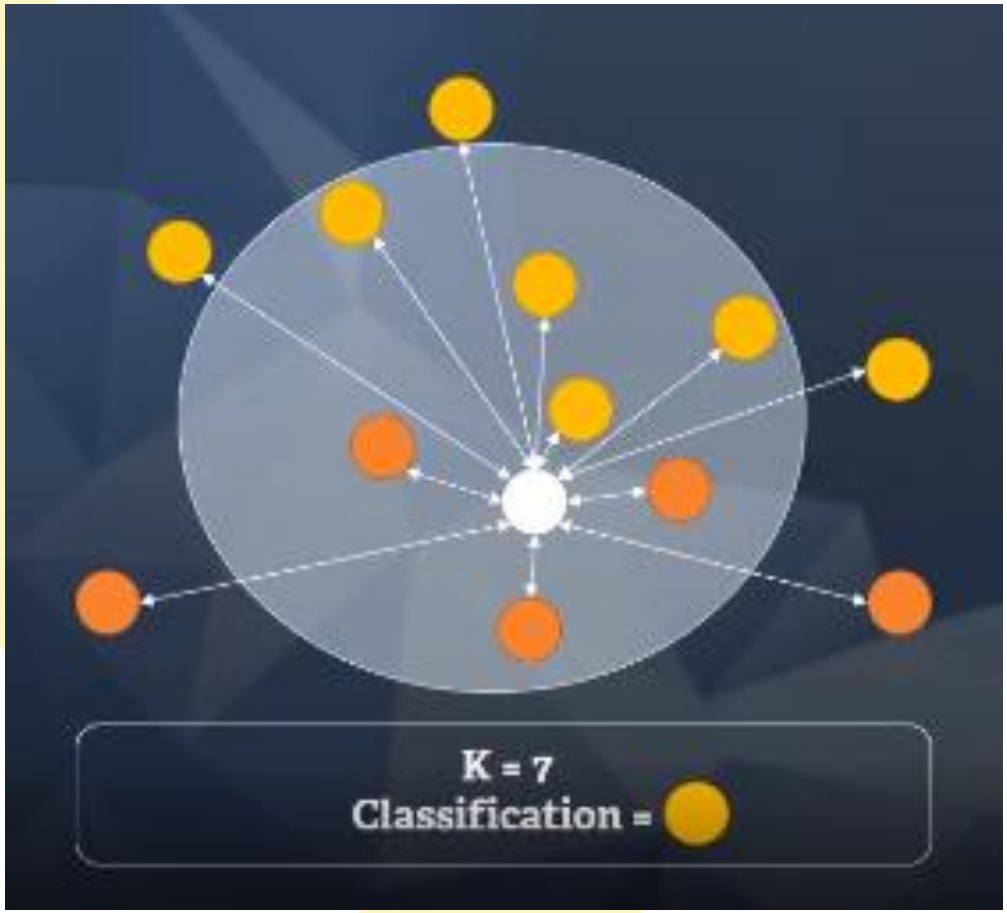
K=3



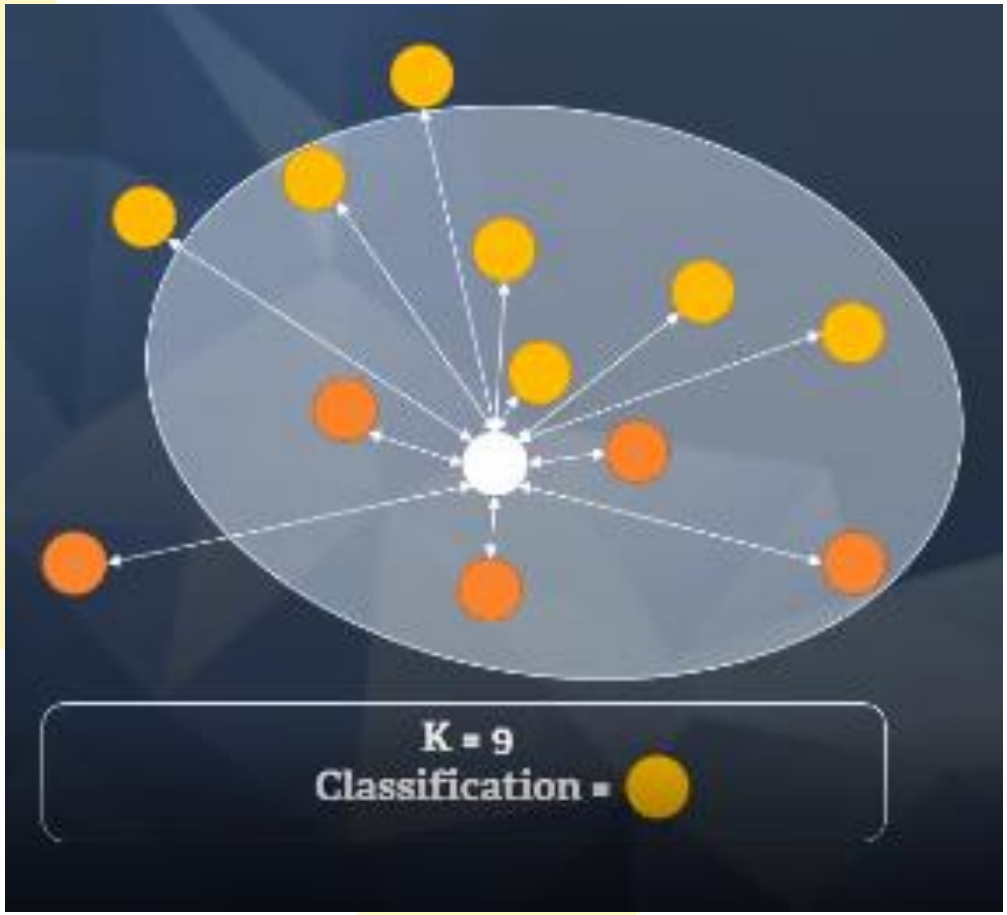
K=5



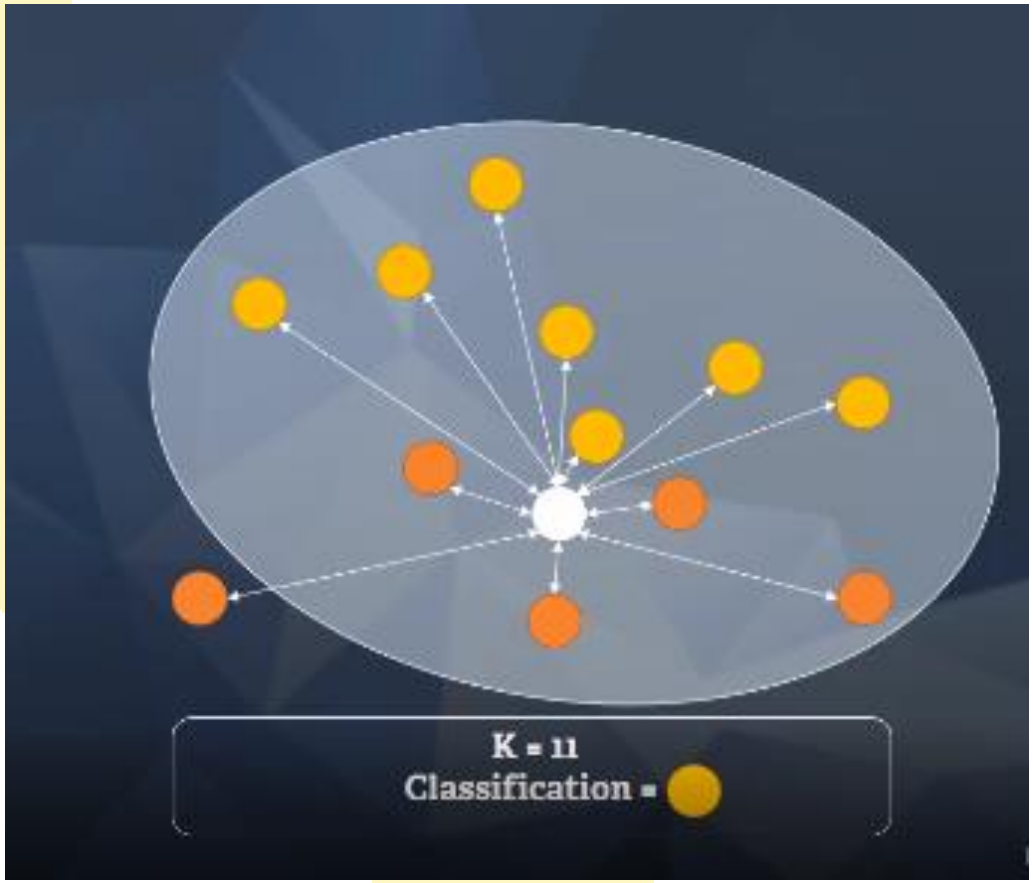
K=7



K=1



K=11



KSCHOOL

Basic Deep Learning

KSCHOOL

Image recognition – Object detection (basic Deep learning)

De esto…

$$p = \frac{1}{1 + e^{-(\beta_0 + \beta_1 x_1 + \beta_2 x_2 + \cdots + \beta_n x_n)}}$$

¡A esto!

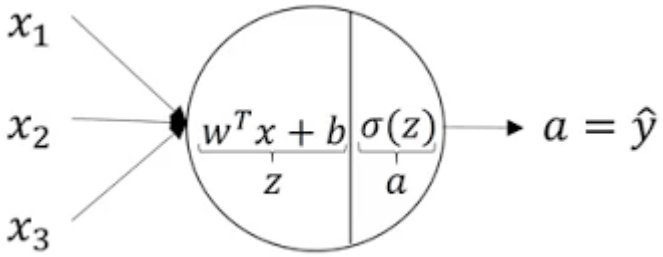
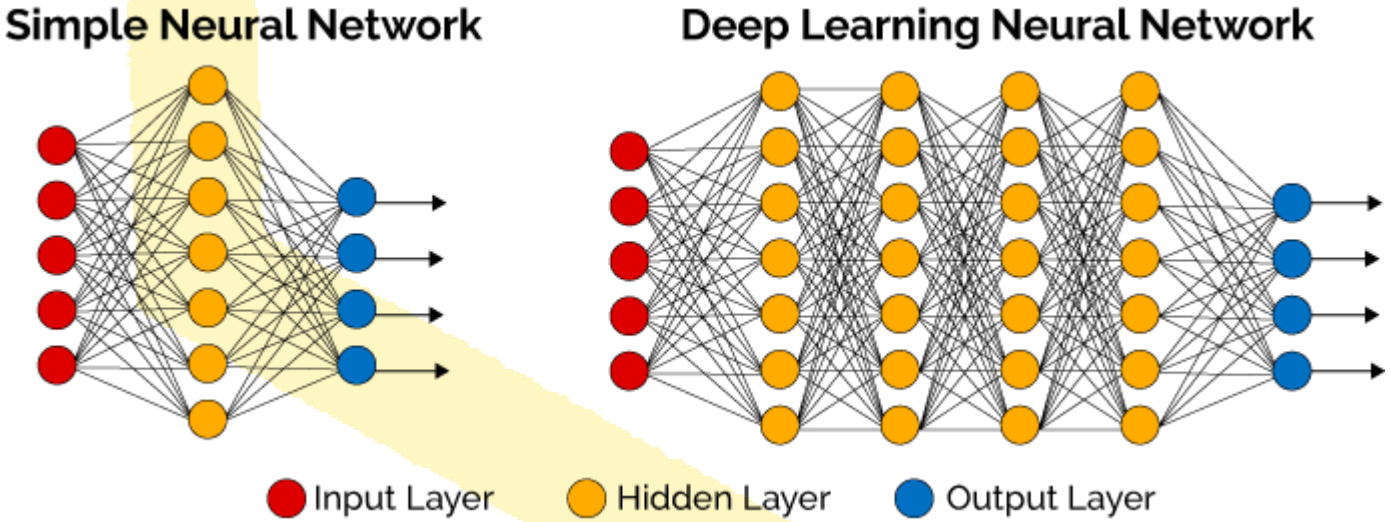


Image recognition – Object detection (basic Deep learning)

¡O esto! 😱



RGB: Red, Green, Blue

- La intensidad de cada una de las componentes se mide según una escala que va del 0 al 255 y cada color es definido por un conjunto de valores (correspondientes a valores "R", "G" y "B").
- El rojo se obtiene con (255,0,0)
- El verde con (0,255,0)
- El azul con (0,0,255)
- Cuando leemos una imagen en R, automáticamente se le asigna una matriz en RGB como vemos a continuación.

De una imagen a una matriz

pixel image

imread

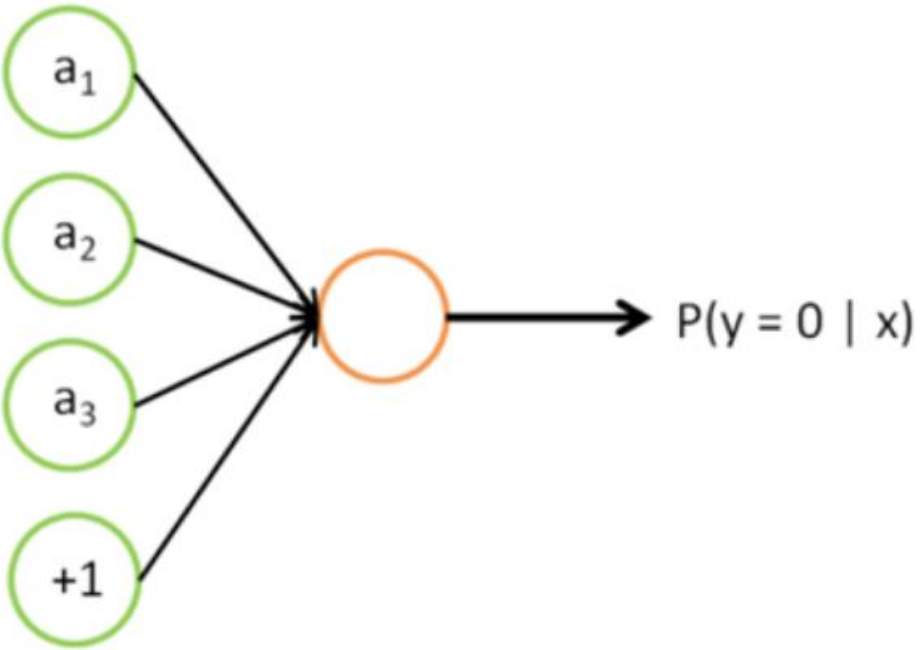
Blue
Green
Red

255	134	93	22
255	134	202	22
255	231	42	22
123	94	83	2
34	44	187	92
34	76	232	124
67	83	194	202

reshaped image vector

$$\begin{pmatrix} 255 \\ 231 \\ 42 \\ 22 \\ 123 \\ 94 \\ \vdots \\ \vdots \\ 92 \\ 142 \end{pmatrix}$$

De una imagen a una matriz



Logistic Regression

KSCHOOL

Let's do it...

KSCHOOL

Supervised Learning Project

Let's do it...

KSCHOOL

Summary Supervised Learning

¿qué debemos saber de cada implementaión?

Summary: Conceptos básicos

- Supervised vs. Unsupervised Learning.
- Dataset de train y test: Entrenamos nuestro modelo, ¿cómo sabemos si es bueno?
- Overfitting: Si intentas reajustar demasiado no predices.
- Matriz de confusión y medidas de rendimiento: ¿cómo mido qué tan bueno es mi modelo?
- Cross – Validation: ¿depende mi modelo de los datos de entrenamiento?

KSCHOOL

Summary: Regresión lineal

- Estimación de valores.
- Modelo más básico y a la vez útil.
- Interpretar los coeficientes

$$y = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \cdots + \beta_n x_n$$

- Variables correladas/correlacionadas disminuyen el poder predictivo.

KSCHOOL

Summary: Regresión lineal

- Distancia de cook.
- Se asume que la variable respuesta/independiente, tiene distribución norma
- R^2 no te confíes, pero tenlo en cuenta siempre.
- ¿Recuerdas los residuales?

Summary: Regresión logística

- Clasificación binaria.
- Interpretar los coeficientes

$$y = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \cdots + \beta_n x_n$$

- Variables correladas/correlacionadas disminuyen el poder predictivo.
- Distancia de cook.
- Se asume que la variable respuesta/independiente, tiene distribución normal.

KSCHOOL

Summary: Regresión logística

- Modelo más básico y a la vez útil.
- Interpretar los coeficientes

$$\frac{p}{1-p} = \frac{1}{1+e^{-(\beta_0+\beta_1 x_1+\beta_2 x_2+\cdots+\beta_n x_n)}}$$

- Variables correladas/correlacionadas disminuyen el poder predictivo.
- Cutoff: decide si el valor es TRUE o FALSE. ¿cómo debes elegirlo?

Summary: Decision Trees

- Estimación y clasificación
- Conceptos claros: Root Node, Splitting, Decision Node, Terminal Node, Parent Node, Child Node, Pruning, Branch.
- Fácil de interpretar.
- Poco robustos.
- No obtenemos una probabilidad.

Summary: Random Forest

- Problemas de clasificación y estimación.
- Ensemble Learning: ¿utilizar al hombre más fuerte del mundo para mover un coche o muchos hombres normales?
- Promedio de muchos árboles de decisión.
- Suelen ser robustos, a diferencia de los Decision Trees.

Summary: SVM

- Problemas de clasificación y estimación (aunque no lo hayamos visto (SVR)).
- En dos dimensiones: una recta separa las clases.
- En tres dimensiones: un plano separar las clases
- En problemas no lineales: Kernel trick.
- No son del todo robustos.

Summary: Splines cúbicos

- Problemas de estimación.
- Polinomio de grado 3 en los intervalos deseados.
- Smoothed splines: polinomio de grado 3 en todo el dominio de acuerdo a cada grado de libertad.
- No uno, ni dos sino tres grados debe ser el polinomio.
- Tampoco cuatro, cinco o más.

KSCHOOL

Fin...