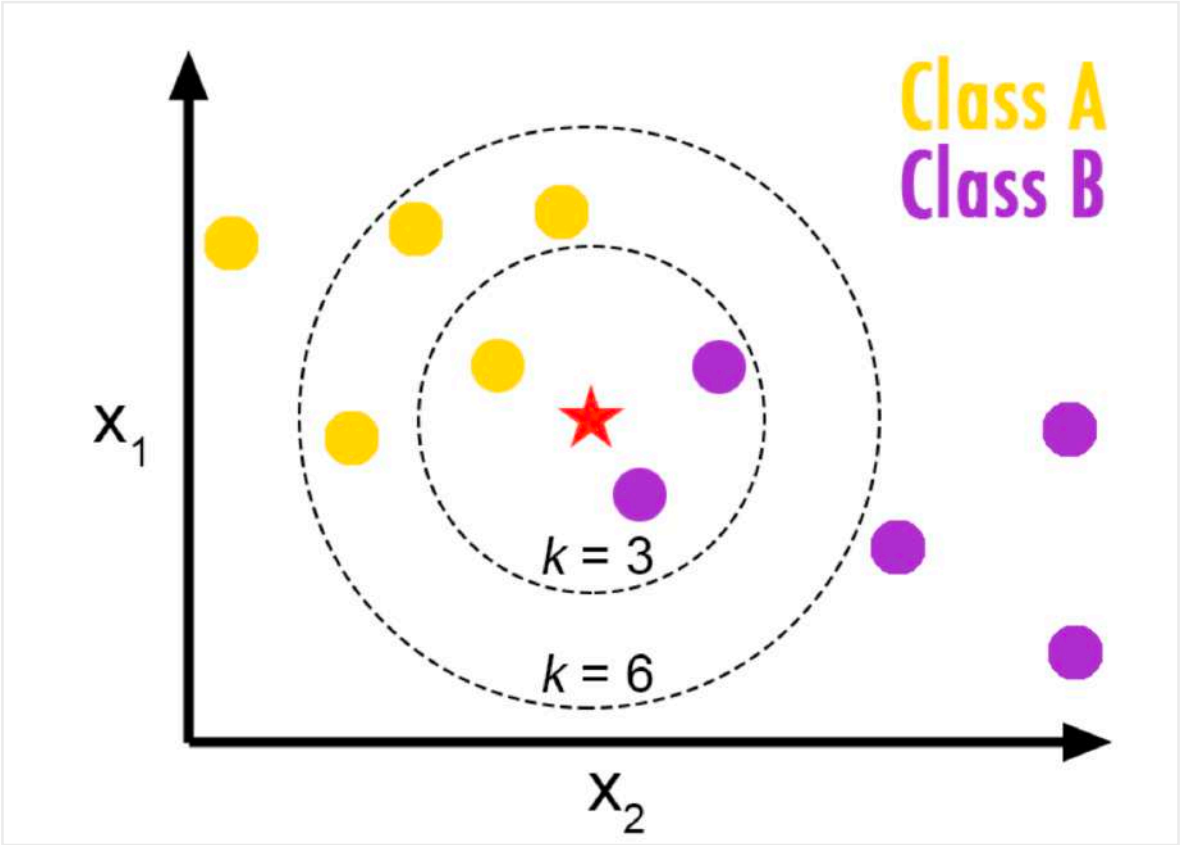
👨‍💻 IBM - DATA SCIENCE - MACHINE LEARNING WITH PYTHON 🐍🧠

In this Lab you will load a customer dataset, fit the data, and use K-Nearest Neighbors to predict a data point. But what is **K-Nearest Neighbors**?

K-Nearest Neighbors is a supervised learning algorithm. Where the data is 'trained' with data points corresponding to their classification. To predict the class of a given data point, it takes into account the classes of the 'K' nearest data points and chooses the class in which the majority of the 'K' nearest data points belong to as the predicted class.



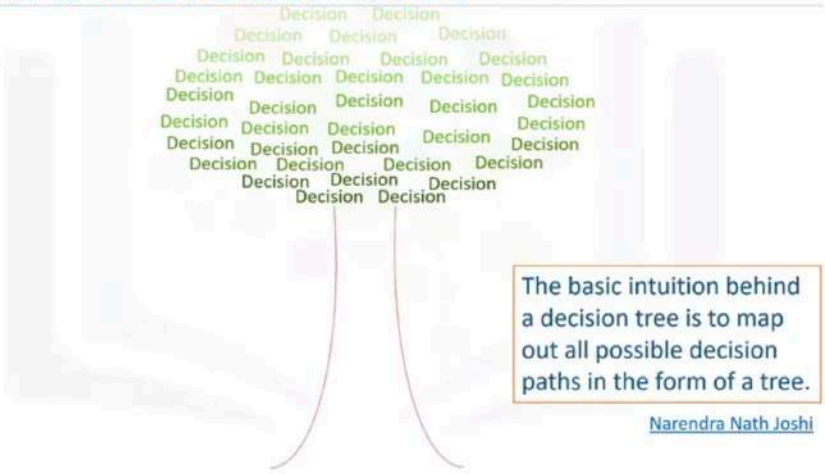
In this case, we have data points of Class A and B. We want to predict what the star (test data point) is. If we consider a k value of 3 (3 nearest data points), we will obtain a prediction of Class B. Yet if we consider a k value of 6, we will obtain a prediction of Class A.

In this sense, it is important to consider the value of k. Hopefully from this diagram, you should get a sense of what the K-Nearest Neighbors algorithm is. It considers the 'K' Nearest Neighbors (data points) when it predicts the classification of the test point.

DECISION TREES

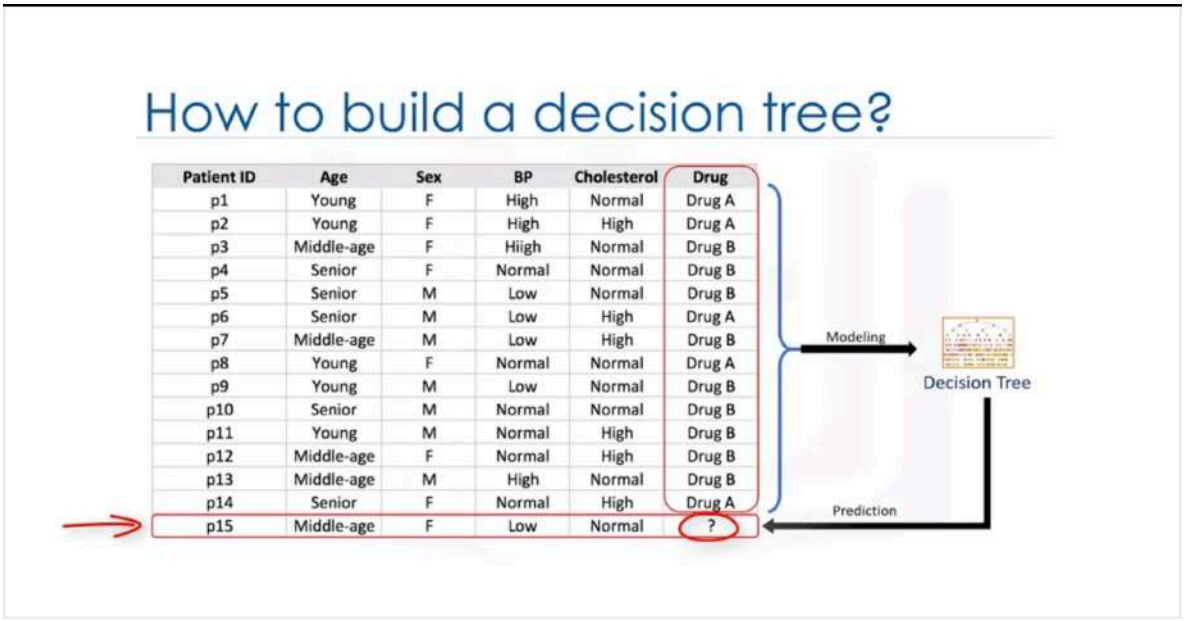
Introduction to Decision Trees

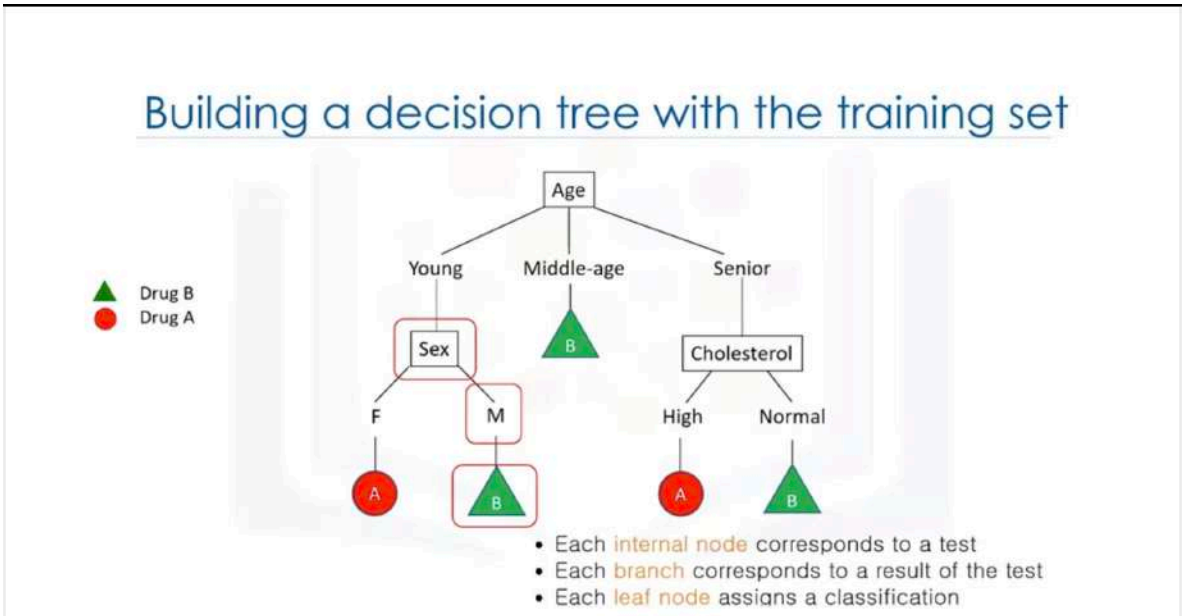
What is a decision tree?



The basic intuition behind a decision tree is to map out all possible decision paths in the form of a tree.

Narendra Nath Joshi

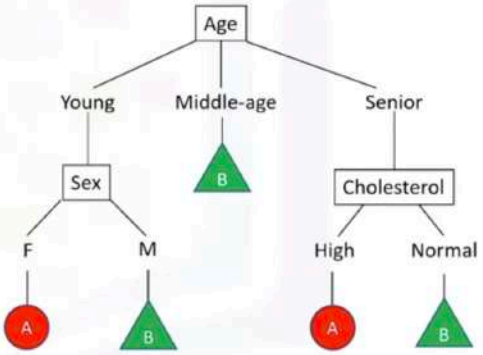




Ok muy rico todo, pero como construimos uno?

Decision tree learning algorithm

1. Choose an attribute from your dataset.
2. Calculate the significance of attribute in splitting of data.
3. Split data based on the value of the best attribute.
4. Go to step 1.

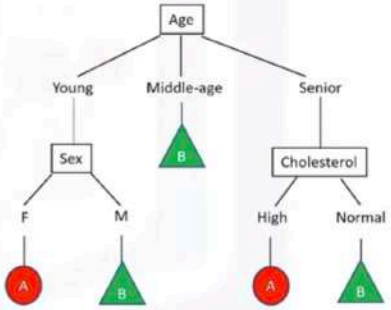


En el siguiente video vemos cómo calcular la significancia de un atributo. La joda es ir priorizándolos por este indicador e iterar así por cada nodo.

Building Decision Trees

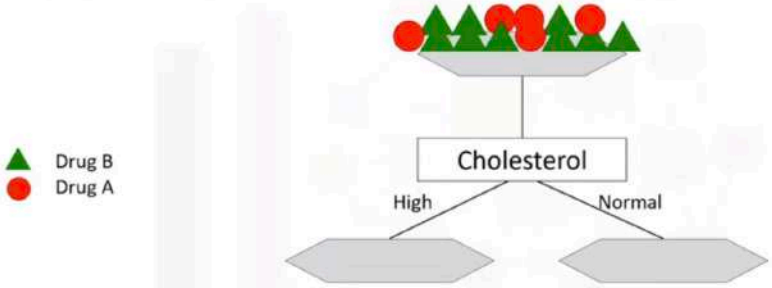
How do you build a decision tree?

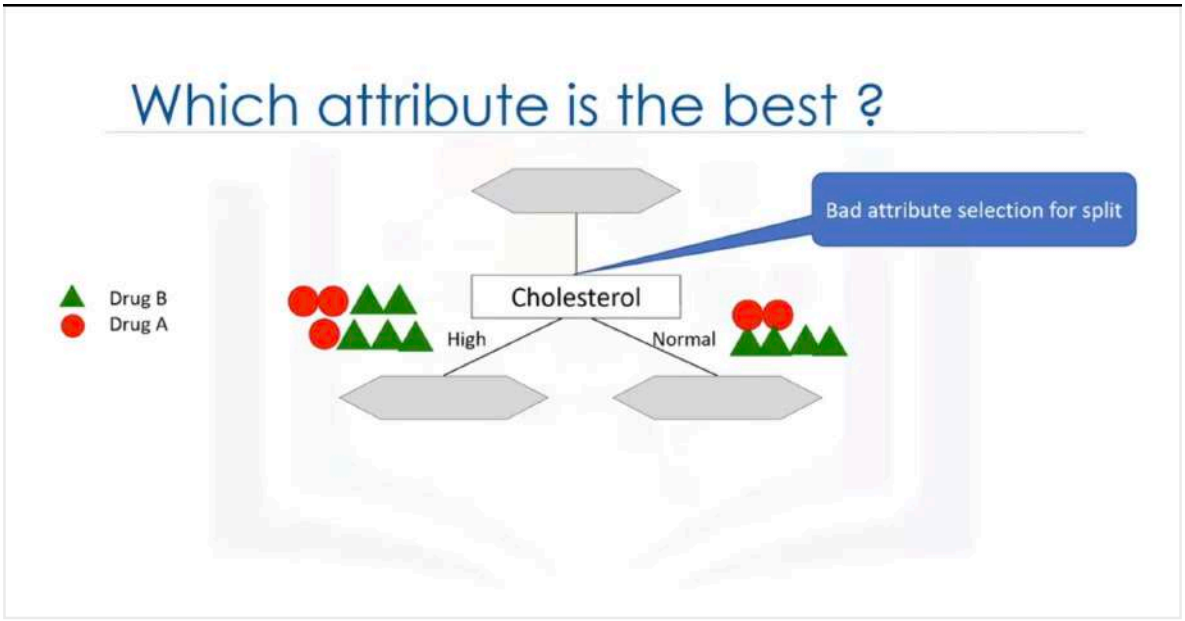
Patient ID	Age	Sex	BP	Cholesterol	Drug
p1	Young	F	High	Normal	Drug A
p2	Young	F	High	High	Drug A
p3	Middle-age	F	Hiigh	Normal	Drug B
p4	Senior	F	Normal	Normal	Drug B
p5	Senior	M	Low	Normal	Drug B
p6	Senior	M	Low	High	Drug A
p7	Middle-age	M	Low	High	Drug B
p8	Young	F	Normal	Normal	Drug A
p9	Young	M	Low	Normal	Drug B
p10	Senior	M	Normal	Normal	Drug B
p11	Young	M	Normal	High	Drug B
p12	Middle-age	F	Normal	High	Drug B
p13	Middle-age	M	High	Normal	Drug B
p14	Senior	F	Normal	High	Drug A
p15	Middle-age	F	Low	Normal	?

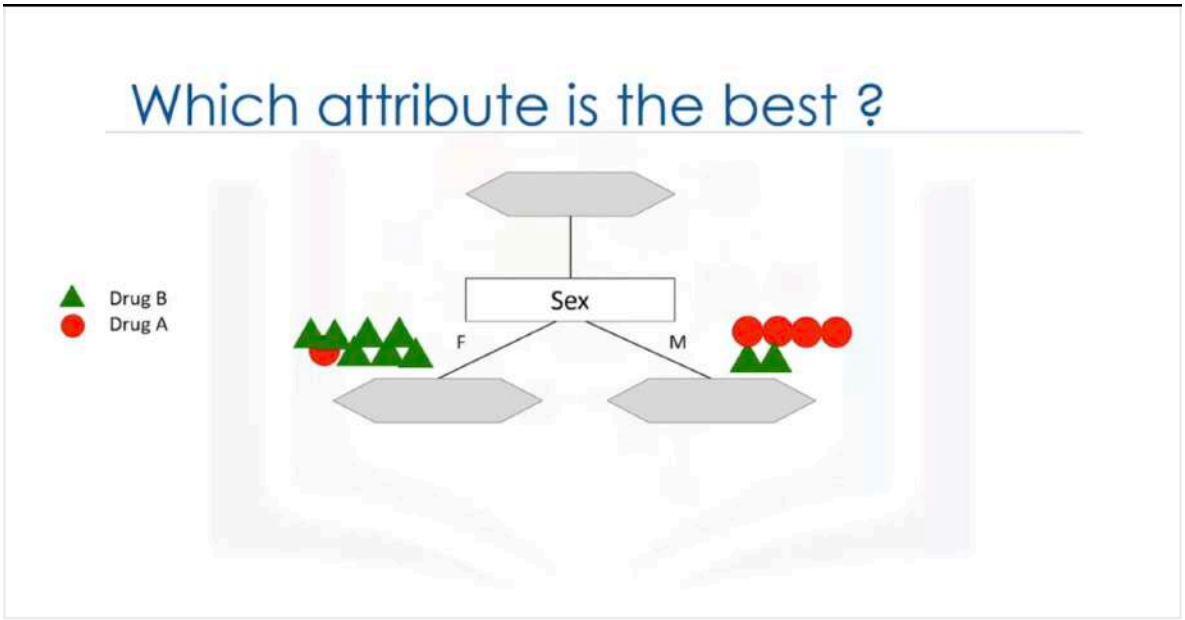


Se construyen haciendo lo que se conoce como ***recursive partitioning*****.**

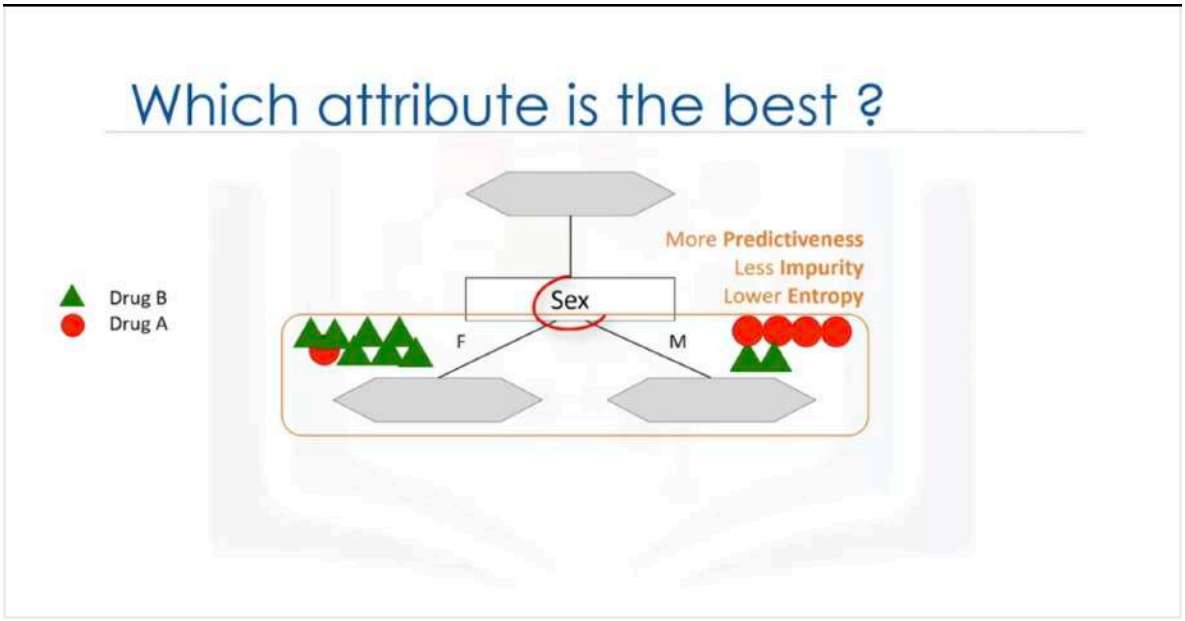
Which attribute is the best ?



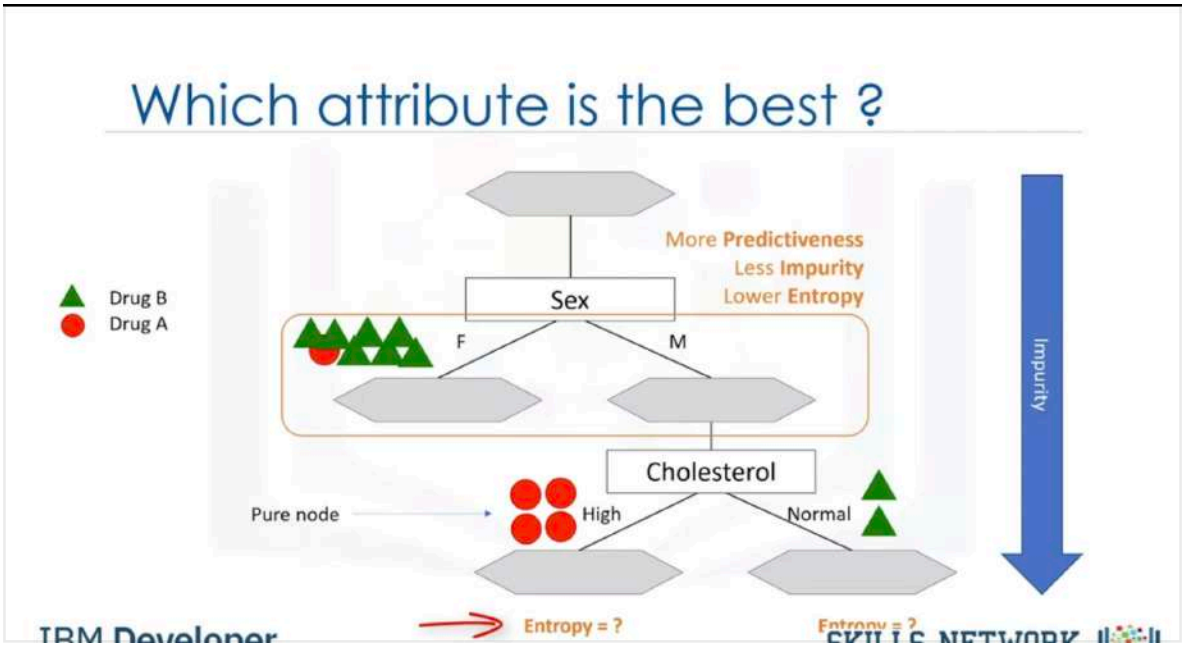




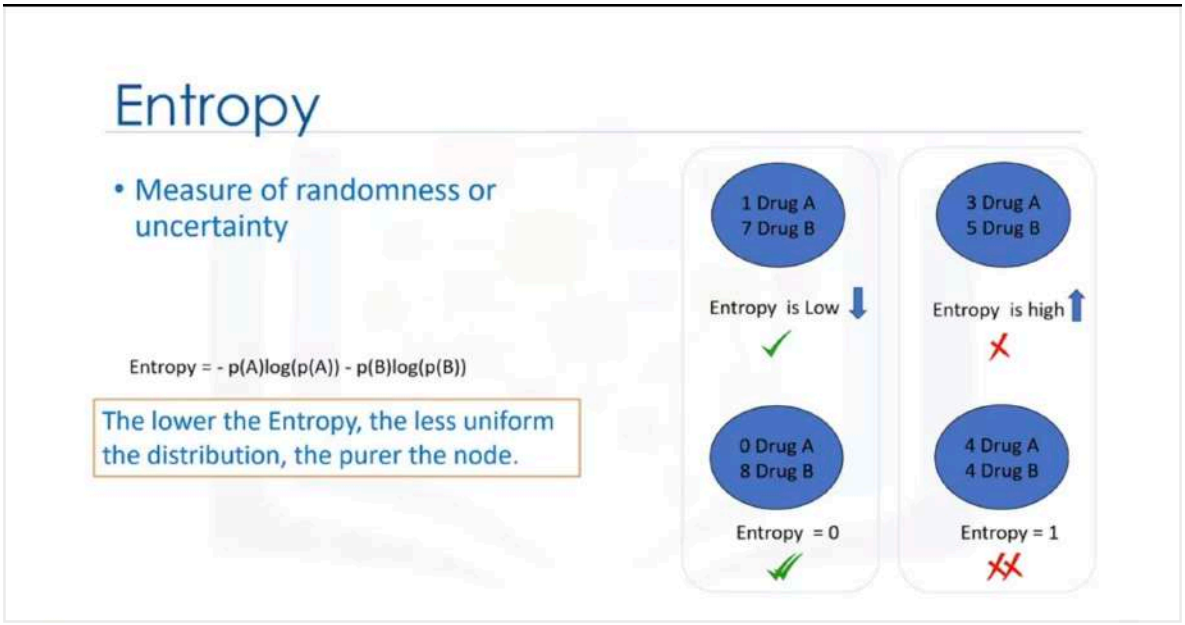
Si bien este segundo es bueno para F no lo es para M. Sin embargo sigue siendo una mejor selección de atributo que colesterol, ya que separo mejor la data, los resultados son mas puro



La joda es seguir desarmando en otros atributos a medida que bajas por el árbol asi obtenes resultados mas puros.



Puro <—> 100% mismo resultado.
La impureza es calculada segun la entropia de la data en los nodos....
Como calculamos la entropia?



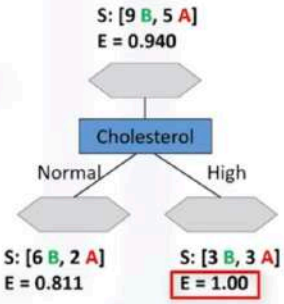
Como ejemplo, calculemos la entropia de mi dataset para construir un arbolito:

Patient ID	Age	Sex	BP	Cholesterol	Drug
p1	Young	F	High	Normal	Drug A
p2	Young	F	High	High	Drug A
p3	Middle-age	F	Hiigh	Normal	Drug B
p4	Senior	F	Normal	Normal	Drug B
p5	Senior	M	Low	Normal	Drug B
p6	Senior	M	Low	High	Drug A
p7	Middle-age	M	Low	High	Drug B
p8	Young	F	Normal	Normal	Drug A
p9	Young	M	Low	Normal	Drug B
p10	Senior	M	Normal	Normal	Drug B
p11	Young	M	Normal	High	Drug B
p12	Middle-age	F	Normal	High	Drug B
p13	Middle-age	M	High	Normal	Drug B
p14	Senior	F	Normal	High	Drug A

Vemos que la entropia de los datos **antes** de separar es 0.94. ahora podemos ir probando distintos atributos para encontrar el que tenga mas productividad, osea el que reduzca mas la entropia.

Is 'Cholesterol' the best attribute?

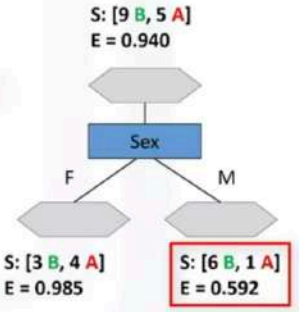
Patient ID	Age	Sex	BP	Cholesterol	Drug
p1	Young	F	High	Normal	Drug A
p2	Young	F	High	High	Drug A
p3	Middle-age	F	Hiigh	Normal	Drug B
p4	Senior	F	Normal	Normal	Drug B
p5	Senior	M	Low	Normal	Drug B
p6	Senior	M	Low	High	Drug A
p7	Middle-age	M	Low	High	Drug B
p8	Young	F	Normal	Normal	Drug A
p9	Young	M	Low	Normal	Drug B
p10	Senior	M	Normal	Normal	Drug B
p11	Young	M	Normal	High	Drug B
p12	Middle-age	F	Normal	High	Drug B
p13	Middle-age	M	High	Normal	Drug B
p14	Senior	F	Normal	High	Drug A



Asi deberíamos ir por todos los atributos...

What about 'Sex'?

Patient ID	Age	Sex	BP	Cholesterol	Drug
p1	Young	F	High	Normal	Drug A
p2	Young	F	High	High	Drug A
p3	Middle-age	F	Hiigh	Normal	Drug B
p4	Senior	F	Normal	Normal	Drug B
p5	Senior	M	Low	Normal	Drug B
p6	Senior	M	Low	High	Drug A
p7	Middle-age	M	Low	High	Drug B
p8	Young	F	Normal	Normal	Drug A
p9	Young	M	Low	Normal	Drug B
p10	Senior	M	Normal	Normal	Drug B
p11	Young	M	Normal	High	Drug B
p12	Middle-age	F	Normal	High	Drug B
p13	Middle-age	M	High	Normal	Drug B
p14	Senior	F	Normal	High	Drug A



Which attribute is the best?

Patient ID	Age	Sex	BP	Cholesterol	Drug
p1	Young	F	High	Normal	Drug A
p2	Young	F	High	High	Drug A
p3	Middle-age	F	High	Normal	Drug B
p4	Senior	F	Normal	Normal	Drug B
p5	Senior	M	Low	Normal	Drug B
p6	Senior	M	Low	High	Drug A
p7	Middle-age	M	Low	High	Drug B
p8	Young	F	Normal	Normal	Drug A
p9	Young	M	Low	Normal	Drug B
p10	Senior	M	Normal	Normal	Drug B
p11	Young	M	Normal	High	Drug B
p12	Middle-age	F	Normal	High	Drug B
p13	Middle-age	M	High	Normal	Drug B
p14	Senior	F	Normal	High	Drug A

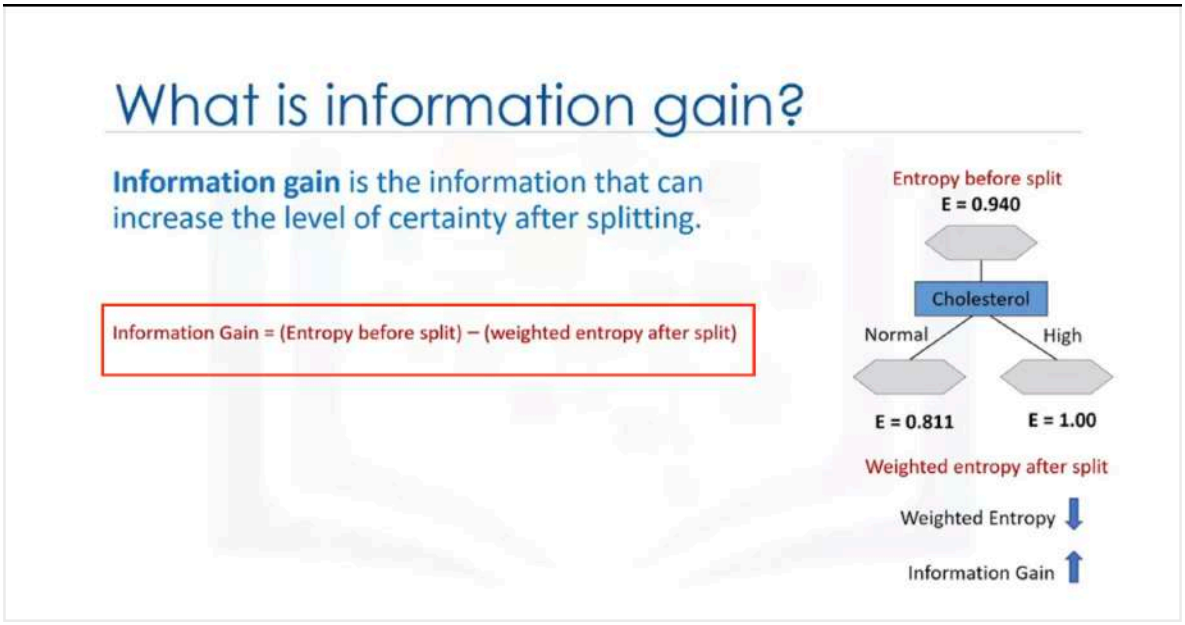
?

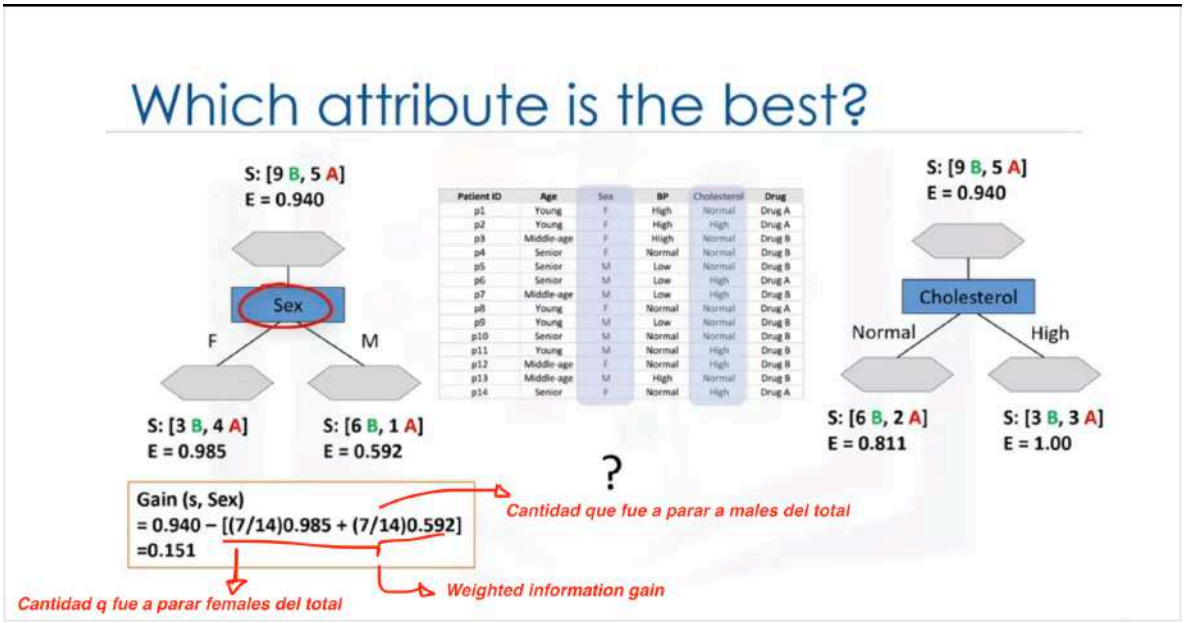
Which attribute is the best?

Patient ID	Age	Sex	BP	Cholesterol	Drug
p1	Young	F	High	Normal	Drug A
p2	Young	F	High	High	Drug A
p3	Middle-age	F	High	Normal	Drug B
p4	Senior	F	Normal	Normal	Drug B
p5	Senior	M	Low	Normal	Drug B
p6	Senior	M	Low	High	Drug A
p7	Middle-age	M	Low	High	Drug B
p8	Young	F	Normal	Normal	Drug A
p9	Young	M	Low	Normal	Drug B
p10	Senior	M	Normal	Normal	Drug B
p11	Young	M	Normal	High	Drug B
p12	Middle-age	F	Normal	High	Drug B
p13	Middle-age	M	High	Normal	Drug B
p14	Senior	F	Normal	High	Drug A

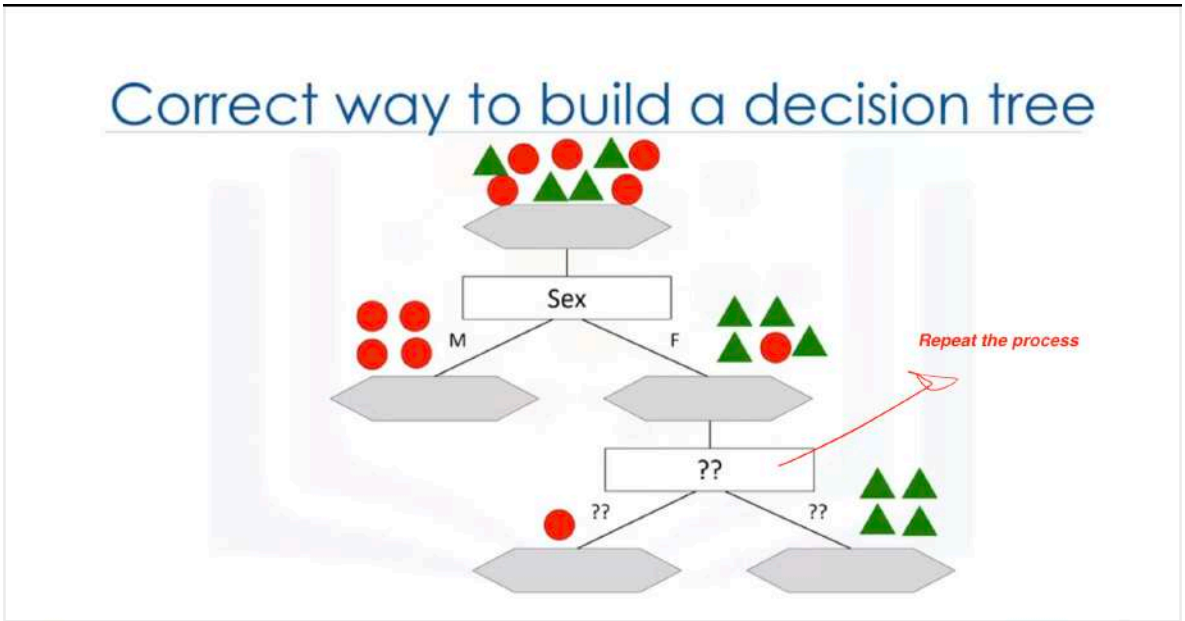
?

The tree with the higher Information Gain after splitting.



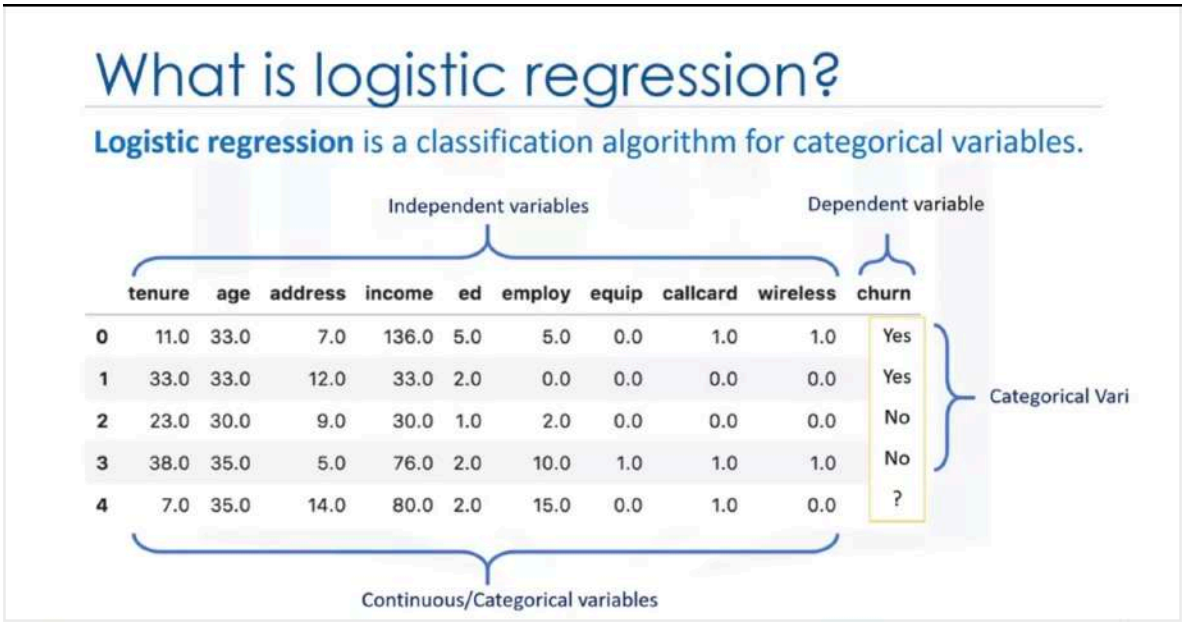


Por lo tanto, claramente gano el del sex. Tiene muchísimo mas Information gain.



LOGISTIC REGRESSION

Se usa para classification. Cuando lo usamos? Que es?...

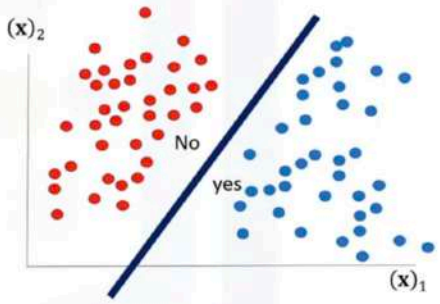


Es análogo a la regresión lineal, con la diferencia de que no predice valores numéricos sino categóricos. La variable dependiente que queremos predecir tiene que ser binaria (positivo negativo).
Las variables independientes tienen que ser continuas. Si son categóricas las tenés que convertir transformando la data. (Dummy o indicators).
Logistic regression en realidad puede ser usada para binar classification o mulitclass classification, pero en este video solo vemos variables dependiente binarias.
Ejemplos de aplicación:

Notar que ademas se entrega la probabilidad de que se pertenezca a la clase.

When is logistic regression suitable?

- If your data is binary
  - 0/1, YES/NO, True/False
- If you need probabilistic results
- When you need a linear decision boundary
- If you need to understand the impact of a feature



$\theta_0 + \theta_1 x_1 + \theta_2 x_2 > 0$

Building a model for customer churn

$\mathbf{x}$ $\quad$ y

	tenure	age	address	income	ed	employ	equip	callcard	wireless	churn
0	11.0	33.0	7.0	136.0	5.0	5.0	0.0	1.0	1.0	1.0
1	33.0	33.0	12.0	33.0	2.0	0.0	0.0	0.0	0.0	1.0
2	23.0	30.0	9.0	30.0	1.0	2.0	0.0	0.0	0.0	0.0
3	38.0	35.0	5.0	76.0	2.0	10.0	1.0	1.0	1.0	0.0

$X \in \mathbb{R}^{m \times n}$

$y \in \{0,1\}$

$\hat{y} = \text{P(y=1|x)}$

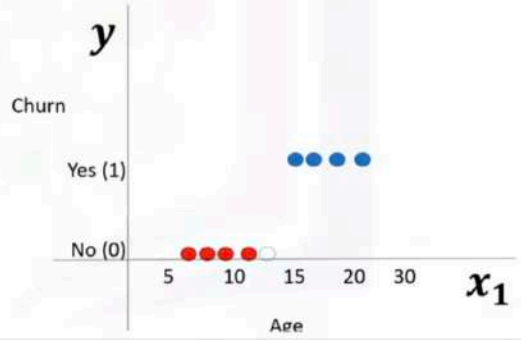
P(y=0|x) = 1- P(y=1|x)

Logistic Regression vs Linear Regression

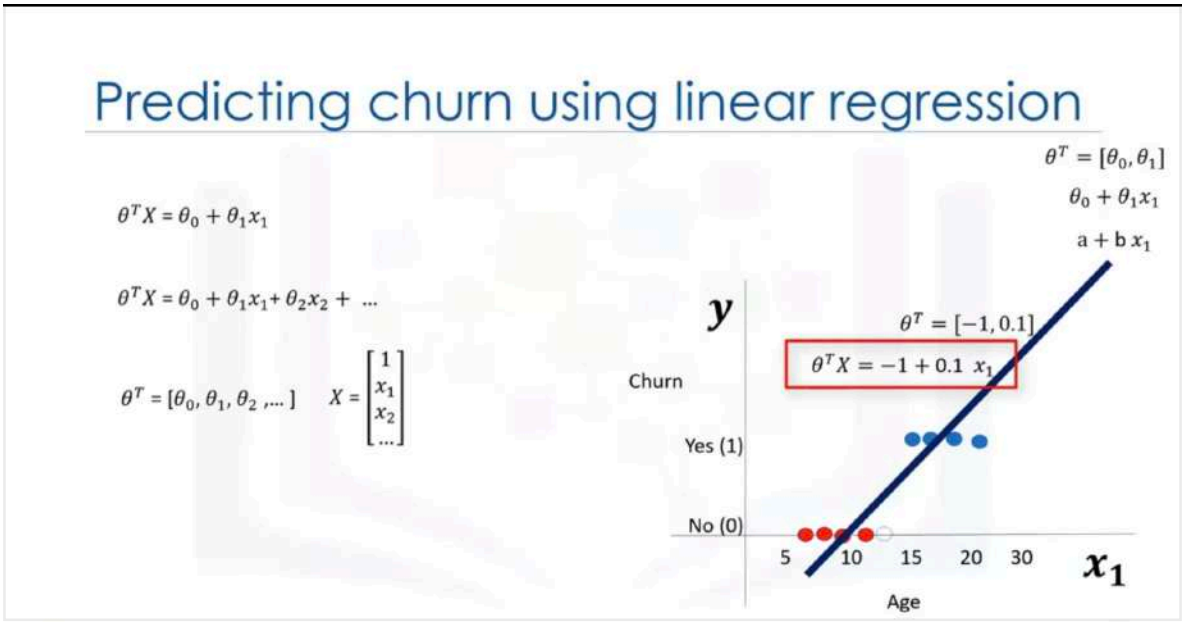
Vamos a ver la funcion sigmoidal.. que es "la parte principal de logístic regression"... osea que las neural networks usan logístic.

Predicting churn using linear regression

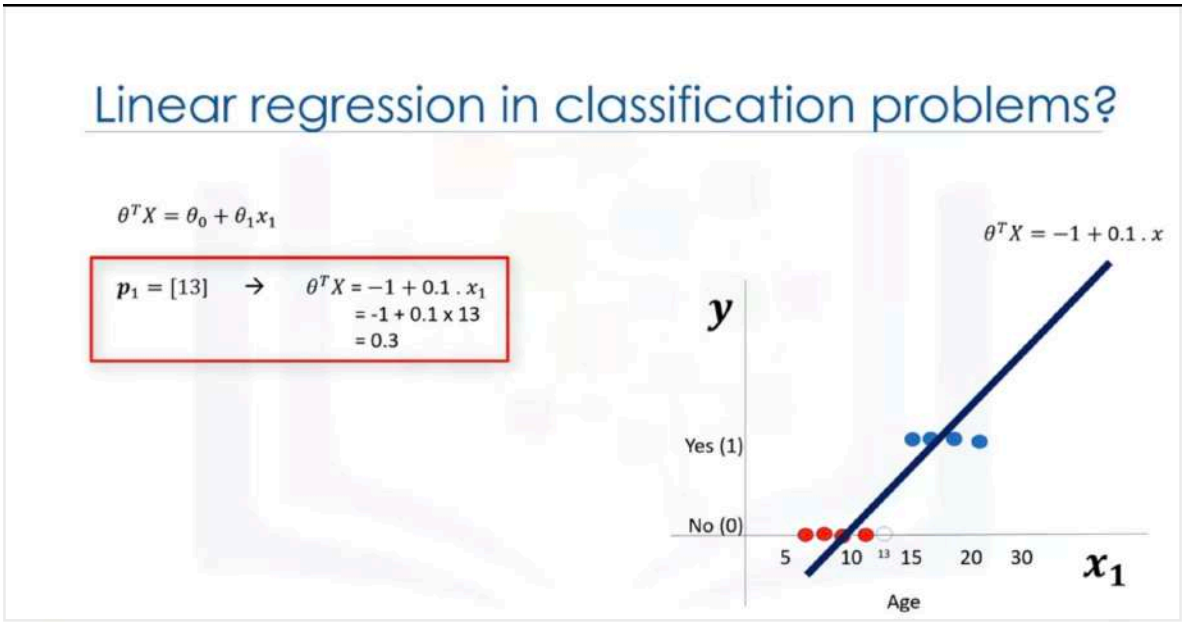
	tenure	age	address	income	ed	employ	equip	callcard	wireless	churn
0	11.0	33.0	7.0	136.0	5.0	5.0	0.0	1.0	1.0	1
1	33.0	33.0	12.0	33.0	2.0	0.0	0.0	0.0	0.0	1
2	23.0	30.0	9.0	30.0	1.0	2.0	0.0	0.0	0.0	0
3	38.0	35.0	5.0	76.0	2.0	10.0	1.0	1.0	1.0	0
4	7.0	35.0	14.0	80.0	2.0	15.0	0.0	1.0	0.0	0



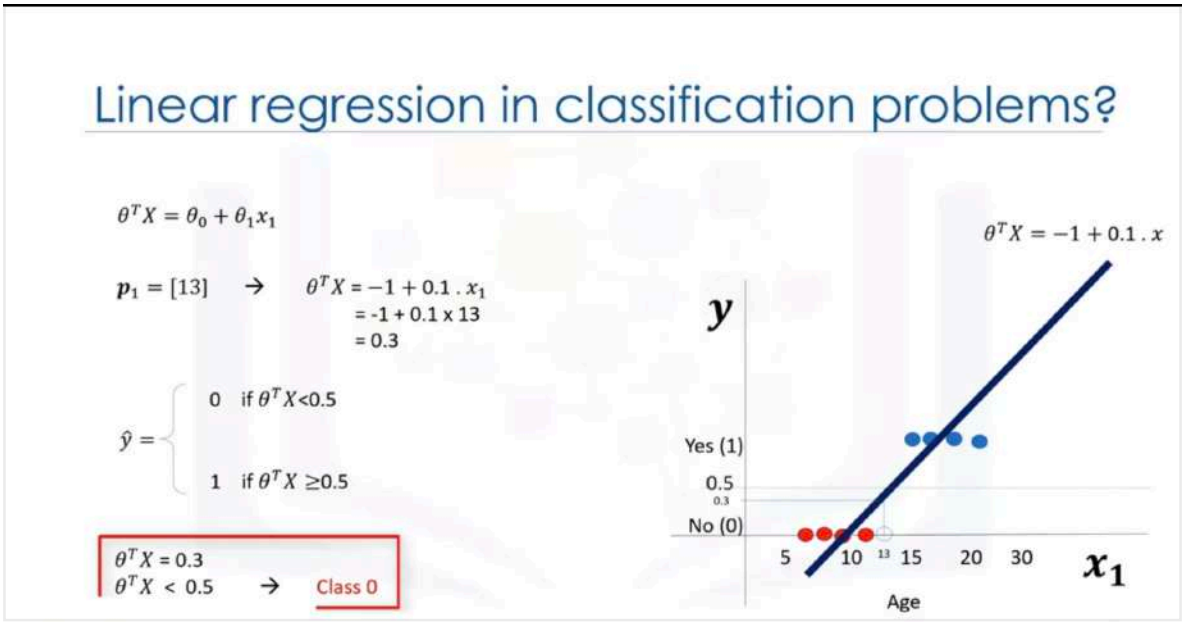
Que pasa si intentamos solucionar este problema con la misma técnica que usábamos con LR?



Que pasa si queremos usar esta linea de regresión obtenida para predecir el churn de mis nuevos customers?

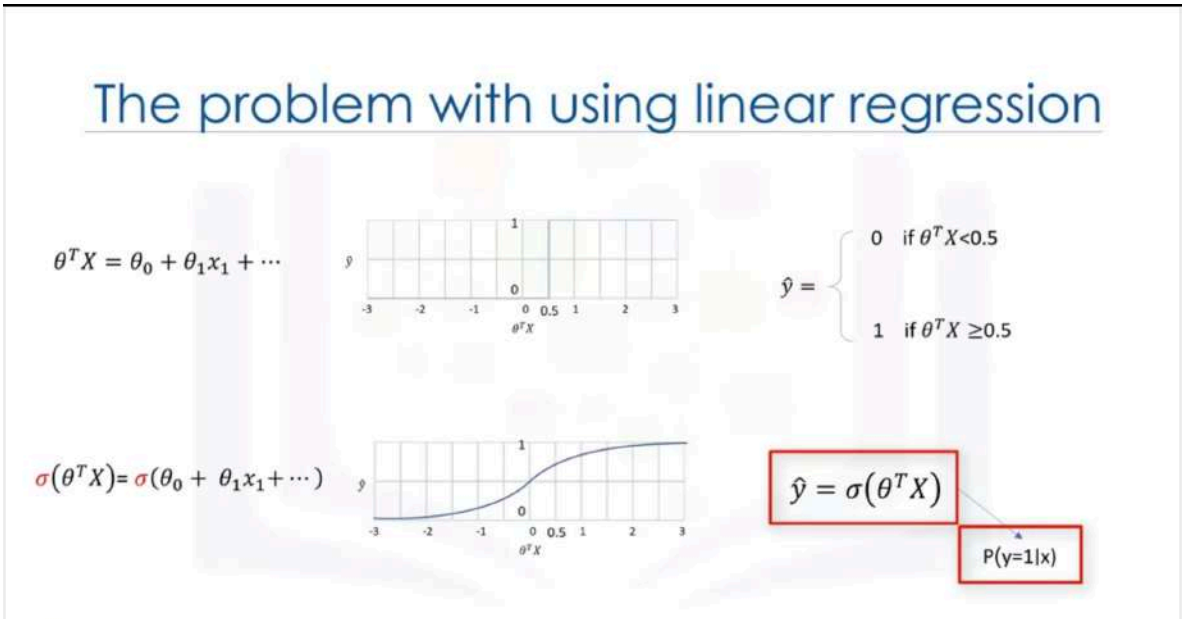


Pones un valor frontera que te permita decidir a que clase pertenece.



Pero tenemos un problema... cuál es la probabilidad de que ese punto de data pertenezca? Es baja.... No es la mejor forma de resolver este problema. Ademas hay otros problemas que verifican que linear regression no es el mejor método para clasificar (no los menciona)

Aca es donde entra la funcion sigmoidal, para cambiar la frontera



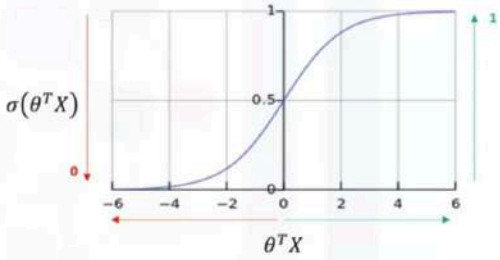
Sigmoid function in logistic regression

- Logistic Function

$$\sigma(\theta^T X) = \frac{1}{1 + e^{-\theta^T X}}$$

$\sigma(\theta^T X) = 1$

$\sigma(\theta^T X) = 0$



Clarification of the customer churn model

What is the output of our model?

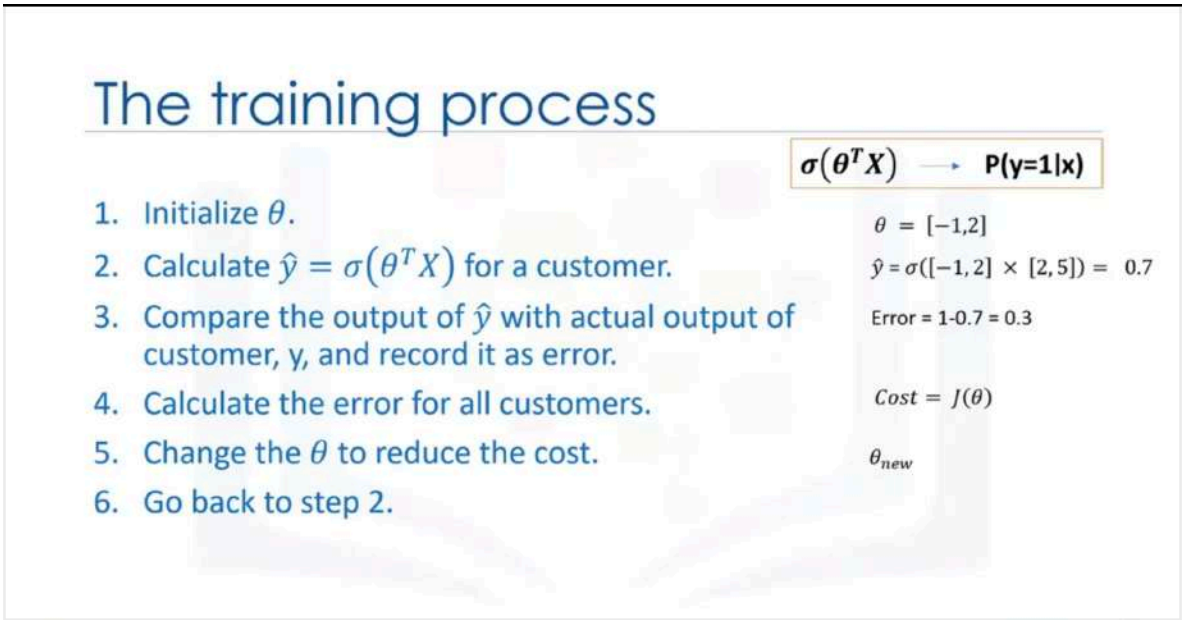
- P(Y=1|X)
- P(y=0|X) = 1 - P(y=1|x)

- P(Churn=1|income,age) = 0.8
- P(Churn=0|income,age) = 1 - 0.8 = 0.2

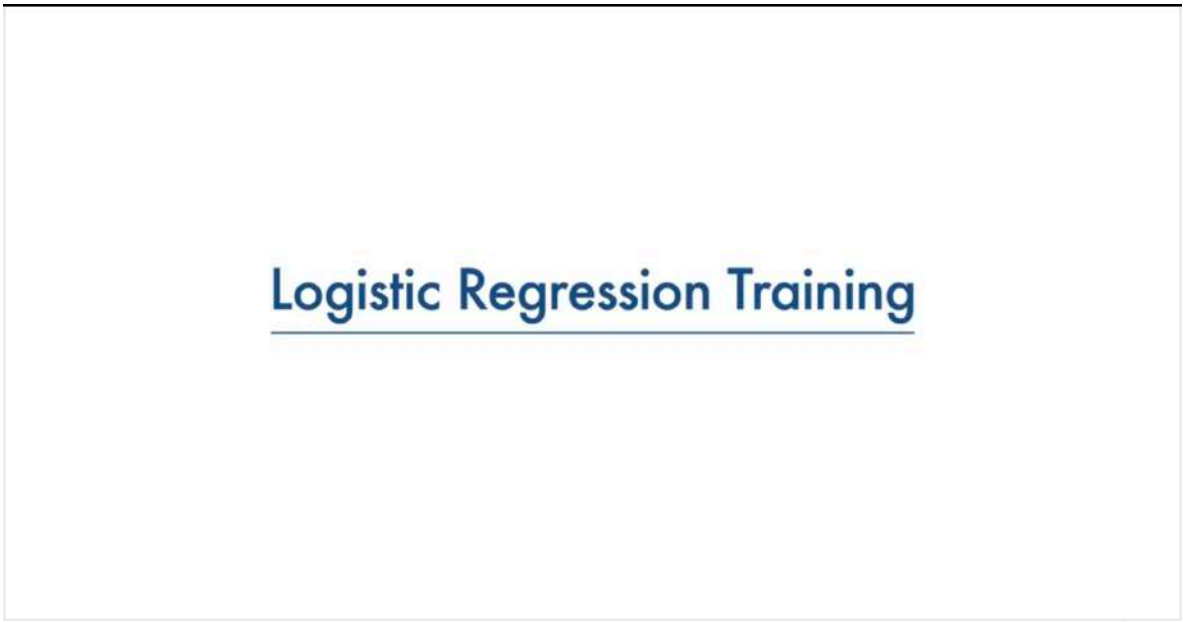
$\sigma(\theta^T X) \rightarrow$ P(y=1|x)

$1 - \sigma(\theta^T X) \rightarrow$ P(y=0|x)

Como logramos eso? Con el training process.

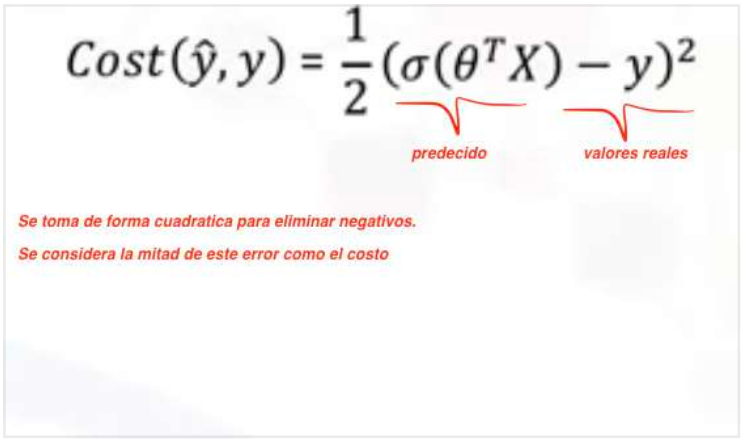


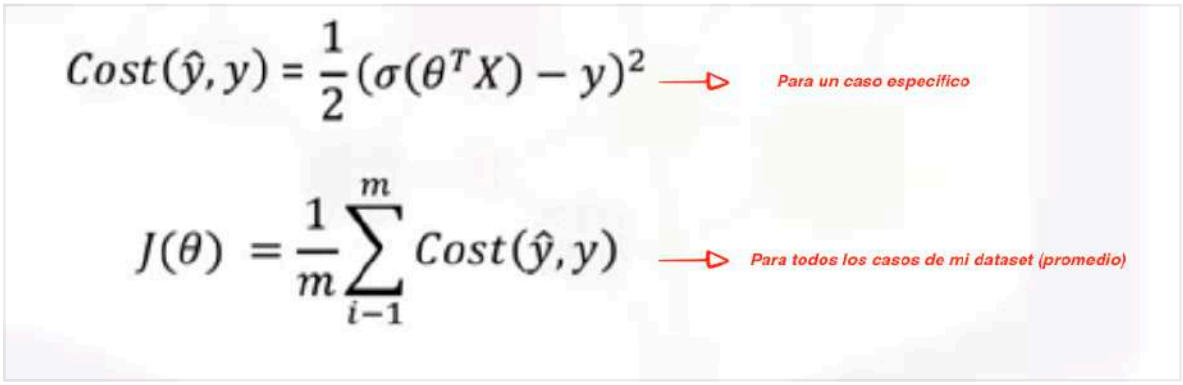
como cambiamos los valores de thita? La forma mas popular es el descenso del gradiente.
La funcion sigmoidal es una constante, predefinida. No cambia en el proceso de entrenamiento.



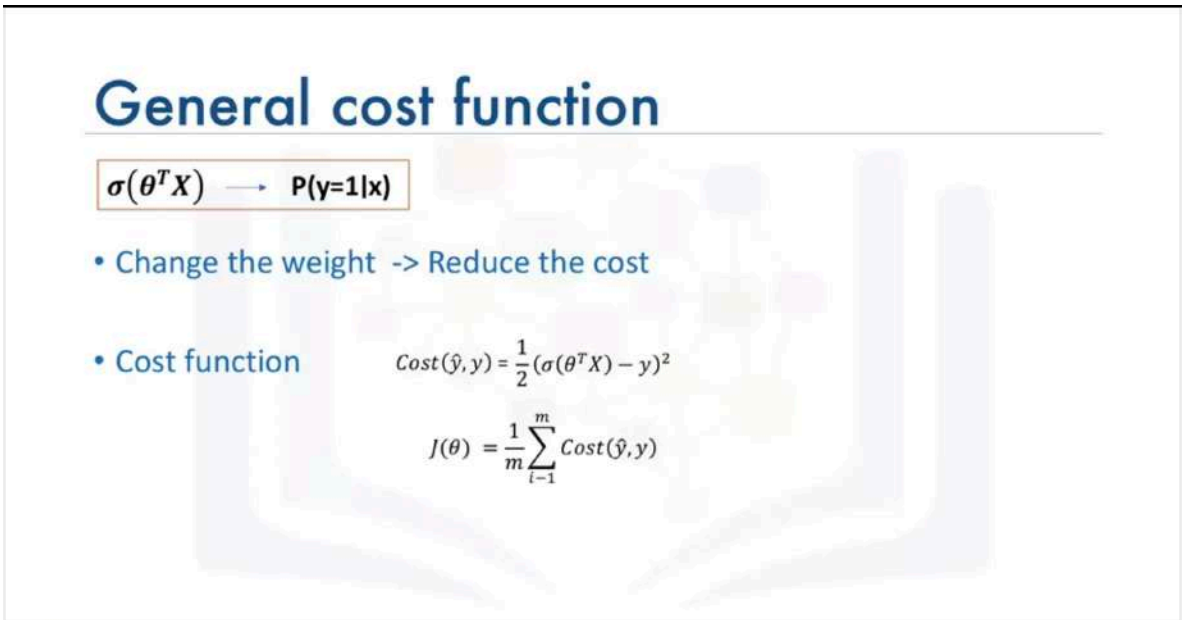
- Entrenamiento de LR
- Cost function
- Tunning de parametros
- Gradient Descent

La joda de todo esto es ir cambiando los parámetros para que el modelo rinda mejor. Esto lo hacemos a través de la cost función. Usando la derivada de la función de costo (gradiente) vamos a ir poder cambiando los parámetros para reducirlo.





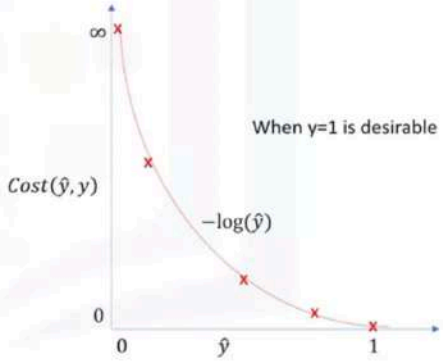
Como calculamos cómo hacer mínimo este error? Debemos encontrar el mínimo de la función. Como es una función multivariable se convierte en un gradiente.



Lo que dice el curso es que "difícil" encontrar el mínimo de esta función. Entonces lo que haces es proponer otra función de costo.

Plotting the cost function of the model

- Model $\hat{y}$
- Actual Value y=1 or 0
- If Y=1, and $\hat{y}$=1 → cost = 0
- If Y=1, and $\hat{y}$=0 → cost = large



La podemos reescribir como:

Logistic regression cost function

- So, we will replace cost function with:

$$Cost(\hat{y}, y) = \frac{1}{2}(\sigma(\theta^T X) - y)^2$$

$$Cost(\hat{y}, y) = \begin{cases} -\log(\hat{y}) & if\ y = 1 \\ -\log(1 - \hat{y}) & if\ y = 0 \end{cases}$$

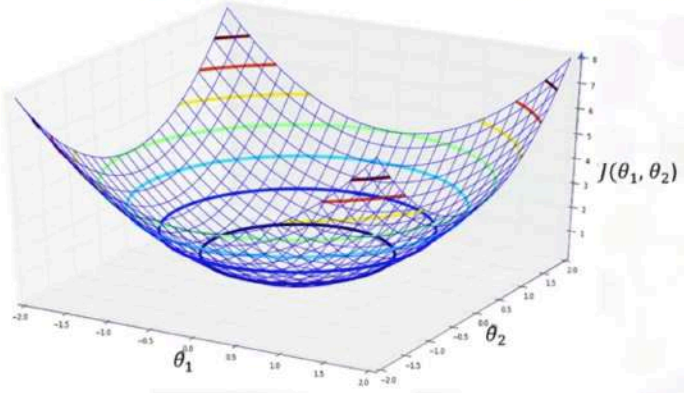
$$J(\theta) = \frac{1}{m}\sum_{i-1}^{m} Cost(\hat{y}, y)$$

$$J(\theta) = -\frac{1}{m}\sum_{i-1}^{m} y^i \log(\hat{y}^i) + (1 - y^i)\log(1 - \hat{y}^i)$$

Minimizing the cost function of the model

- How to find the best parameters for our model?
  - Minimize the cost function
- How to minimize the cost function?
  - Using Gradient Descent
- What is gradient descent?
  - A technique to use the derivative of a cost function to change the parameter values, in order to minimize the cost

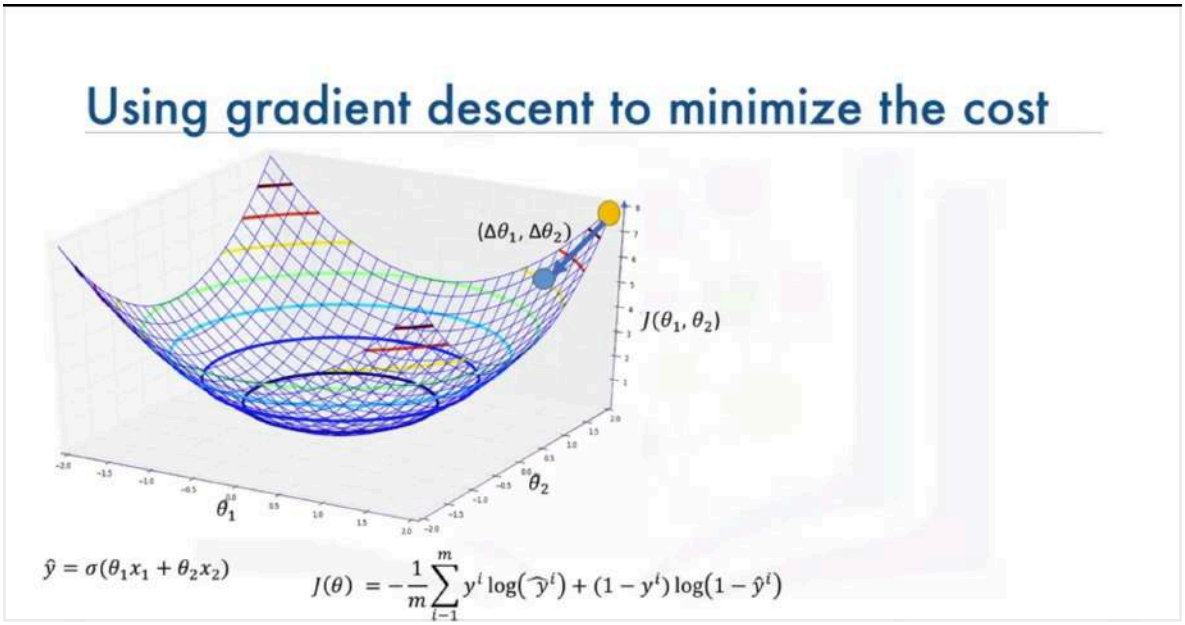
Using gradient descent to minimize the cost



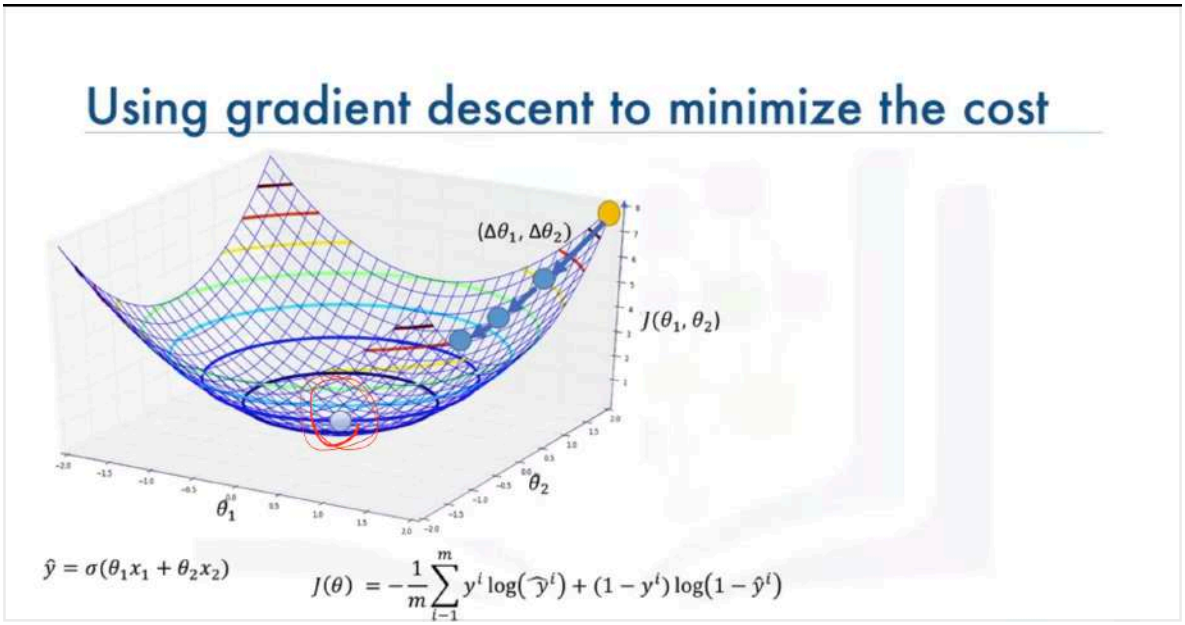
$\hat{y} = \sigma(\theta_1 x_1 + \theta_2 x_2)$

$$J(\theta) = -\frac{1}{m}\sum_{i-1}^{m} y^i \log(\hat{y}^i) + (1 - y^i)\log(1 - \hat{y}^i)$$

Empezas con un punto random y vas iterando cambiando de a diferenciales (deltas) de los parámetros Tita sub n:

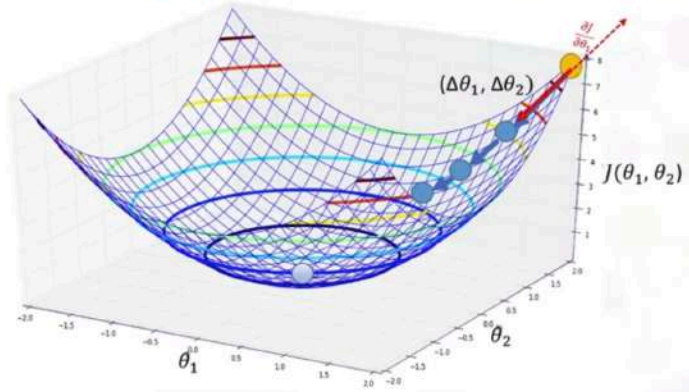


Vas caminando por la superficie y análisis los valores del gradiente.



Para encontrar la dirección y tamaño de estos pasos, necesitamos ir calculando el gradiente de la función de costo. El gradiente es la “pendiente” de la superficie en cada punto y la dirección del gradiente apunta en hacia donde crece la función. Nos tenemos que mover en el sentido contrario del gradiente para estar seguros de que vamos hacia donde la función tiene su mínimo!
El valor del gradiente (numérico) nos indica tambien que tamaño de paso tenemos que tomar. Este va a ir disminuyendo a medida que aumenten las iteraciones.

Using gradient descent to minimize the cost



$$\frac{\partial J}{\partial \theta_1} = -\frac{1}{m}\sum_{i=1}^{m}(y^i - \hat{y}^i)\,x_1^i$$

$$\nabla J = \begin{bmatrix} \frac{\partial J}{\partial \theta_1} \\ \frac{\partial J}{\partial \theta_2} \\ \frac{\partial J}{\partial \theta_3} \\ \cdots \\ \cdots \\ \frac{\partial J}{\partial \theta_k} \end{bmatrix}$$

Learning rate. lo manejamos nosotros

$$New\theta = old\theta - \eta\,\nabla J$$

$$\hat{y} = \sigma(\theta_1 x_1 + \theta_2 x_2)$$

$$J(\theta) = -\frac{1}{m}\sum_{i-1}^{m} y^i \log(\hat{y}^i) + (1 - y^i)\log(1 - \hat{y}^i)$$

Training algorithm recap

1. initialize the parameters randomly.
2. Feed the cost function with training set, and calculate the error.
3. Calculate the gradient of cost function.
4. Update weights with new values.
5. Go to step 2 until cost is small enough.
6. Predict the new customer X.

$$\theta^T = [\theta_0, \theta_1, \theta_2, \ldots.]$$

$$J(\theta) = -\frac{1}{m}\sum_{i-1}^{m} y^i \log(\hat{y}^i) + (1 - y^i)\log(1 - \hat{y}^i)$$

$$\nabla J = [\frac{\partial J}{\partial \theta_1}, \frac{\partial J}{\partial \theta_2}, \frac{\partial J}{\partial \theta_3}, \ldots, \frac{\partial J}{\partial \theta_k}]$$

$$\theta_{new} = \theta_{prv} - \eta\nabla J$$

$$P(y=1|x) = \sigma(\theta^T X)$$

SUPPORT VECTOR MACHINE (S.V.M)

Support Vector Machine

Classification with SVM

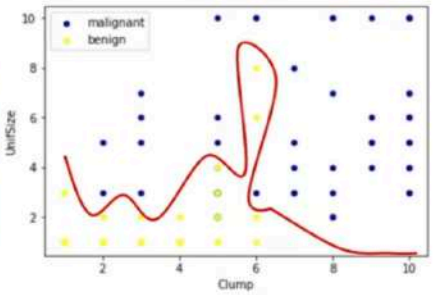
Necesitamos un separator. El tema es que la mayoría de los datasets del mundo real no aceptan separaciones lineales, sino con curvas

What is SVM?

SVM is a supervised algorithm that classifies cases by finding a separator.

1. Mapping data to a **high-dimensional** feature space
2. Finding a **separator**

Clump	UnifSize	UnifShape	MargAdh	SingEpiSize	BareNuc	BlandChrom	NormNucl	Mit	Class
5	1	1	1	2	1	3	1	1	benign
5	4	4	5	7	10	3	2	1	benign
3	1	1	1	2	2	3	1	1	malignant
6	8	8	1	3	4	3	7	1	benign
4	1	1	3	2	1	3	1	1	benign
8	10	10	8	7	10		7	1	malignant
1	1	1	1	2	10	3	1	1	benign
2	1	2	H	2	1	3	1	1	benign
2	1	1	1	2	1	1	1	5	benign
4	2	1	1	2	1	2	1	1	benign



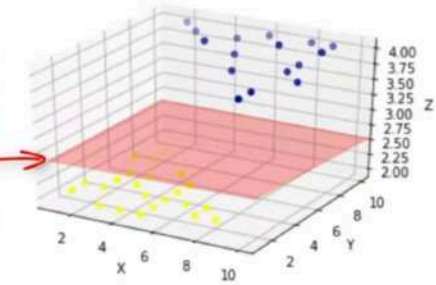
Si transferimos la data a un espacio de R3:

What is SVM?

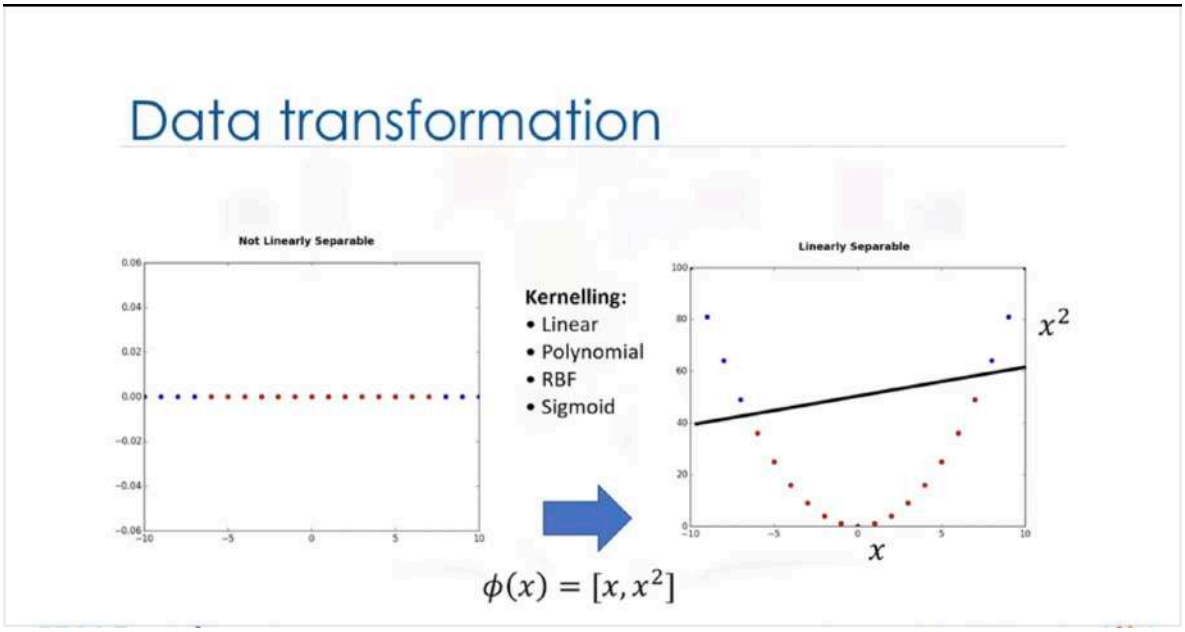
SVM is a supervised algorithm that classifies cases by finding a separator.

1. Mapping data to a **high-dimensional** feature space
2. Finding a **separator**

Clump	UnifSize	UnifShape	MargAdh	SingEpiSize	BareNuc	BlandChrom	NormNucl	Mit	Class
5	1	1	1	2	1	3	1	1	benign
5	4	4	5	7	10	3	2	1	benign
3	1	1	1	2	2	3	1	1	malignant
6	8	8	1	3	4	3	7	1	benign
4	1	1	3	2	1	3	1	1	benign
8	10	10	8	7	10		7	1	malignant
1	1	1	1	2	10	3	1	1	benign
2	1	2	H	2	1	3	1	1	benign
2	1	1	1	2	1	1	1	5	benign
4	2	1	1	2	1	2	1	1	benign

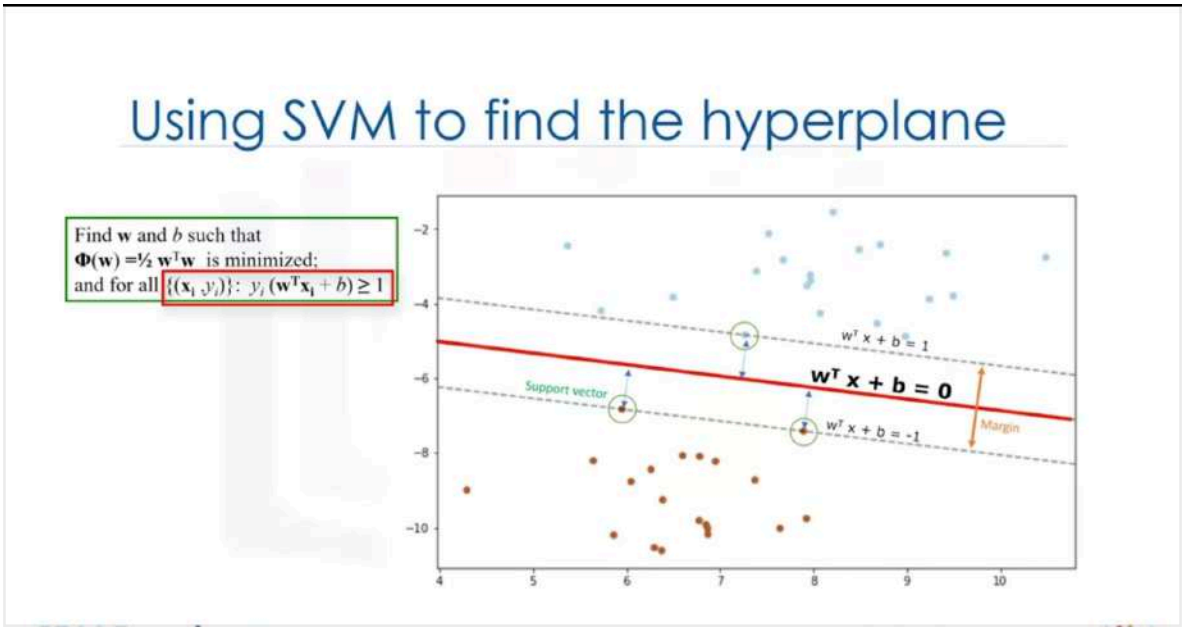


El output de SVM es un hiperplane optimizado para separar de la mejor forma los casos de la clasificación.
Como transferimos data de una manera tal que podamos dibujar un separador como un hyperplane? Y como podemos encontrar el mejor de estos planos post transformación?
Mapping data into a higher dimensional space is called Kernelling



El Fi de la slide de arriba es lo que se conoce como Kernell Function

La siguiente pregunta es como encontramos el separador mas optimo después de haber transformado mis datos?
Un primer approach seria elegir el que represente el mayor margen entre las clases
Lo que esta cerca de la frontera es lo que llamamos support vector.



No entra en detalle con la matemática :(
Este problema de optimización dice que también se puede resolver por descenso del gradiente, pero que no lo vemos en este video.
Como usan solamente los support vectores para el algoritmo, son eficientes. Pero para datasets chiquitos. (<1000 filas)
Lo malo es que muy común overfitteen si la cantidad de features es mayor a la cantidad de **Samples**.

Pros and cons of SVM

- Advantages:
  - Accurate in high-dimensional spaces
  - Memory efficient
- Disadvantages:
  - Prone to over-fitting
  - No probability estimation
  - Small datasets

SVM applications

- Image recognition
- Text category assignment
- Detecting spam
- Sentiment analysis
- Gene Expression Classification
- Regression, outlier detection and clustering