



Centro de Investigaciones para el Desarrollo – CID
Programa de Educación permanente - PEC
Facultad de Ciencias Económicas
Sede Bogotá

educación continua



UNIVERSIDAD
NACIONAL
DE COLOMBIA

DIPLOMADO EN CIENCIA DE DATOS

APRENDIZAJE AUTOMÁTICO

Mónica Ramírez Bernal

2020-2

Centro de Investigaciones para el Desarrollo – CID
Programa de Educación permanente - PEC
Facultad de Ciencias Económicas
Sede Bogotá

educación continua 



UNIVERSIDAD
NACIONAL
DE COLOMBIA

SOBRE MI

- Ingeniera de Sistemas
- Programación y Bases de Datos
- Certificación Data Scientist
- Data Scientist: *Programador que sabe matemáticas*
- Meetups y compartir conocimiento
- Lead Data Scientist en TAK
- Python!



@monita_rb

monicaramirezrb

github.com/monitarb

monita.ramirezrb@gmail.com

METODOLOGÍA

- Quiz!
- Un poquito de teoría...
- Labs que pueden seguir conmigo (R o Python) x2
- Receso
- Anuncios o Tips adicionales
- Links útiles de cada tema
- Proyecto
- Dudas
- Un favor...

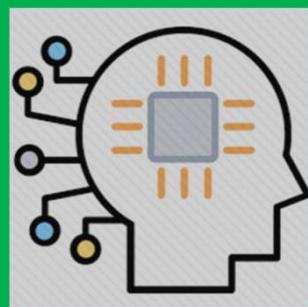
PROYECTO!

- Mismos grupos
- Elegir 3 datasets
- Qué pregunta de negocio vamos a solucionar?
- Presentación final de 10 minutos
 - Desafíos
 - Mejoras al modelo
- Manejo del tiempo
- Participación

¿QUÉ VAMOS A VER AQUÍ?

INTELIGENCIA ARTIFICIAL

Máquinas y Programas
“Inteligentes”



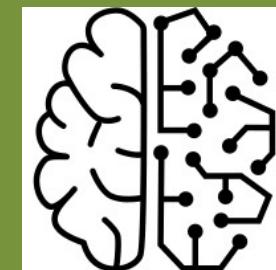
MACHINE LEARNING

“Aprender” sin estar
explícitamente programadas



DEEP LEARNING

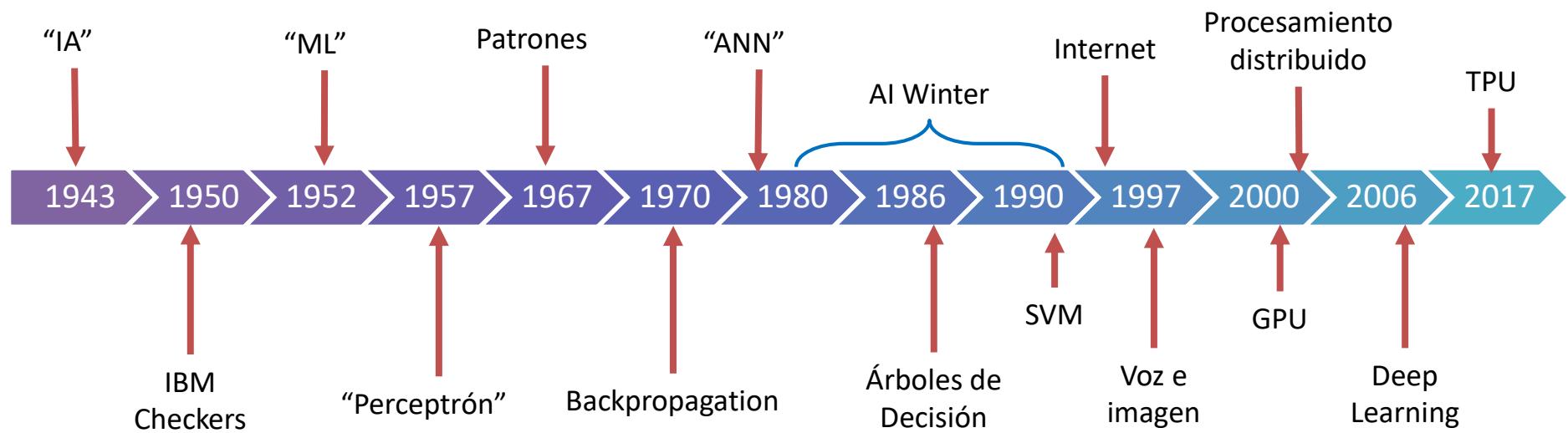
Redes Neuronales
Profundas



¿QUÉ VAMOS A VER AQUÍ?

- **MÓDULO 1:** Introducción, Conceptos y Problemas
- **MÓDULO 2:** Clasificación y Predicción
- **MÓDULO 3:** Validación de Modelos
- **MÓDULO 4:** Redes Neuronales: Conceptos
- **MÓDULO 5:** Tipos de Redes Neuronales
- **MÓDULO 6:** Implementación de Redes Neuronales
- **MÓDULO 7:** Proyecto

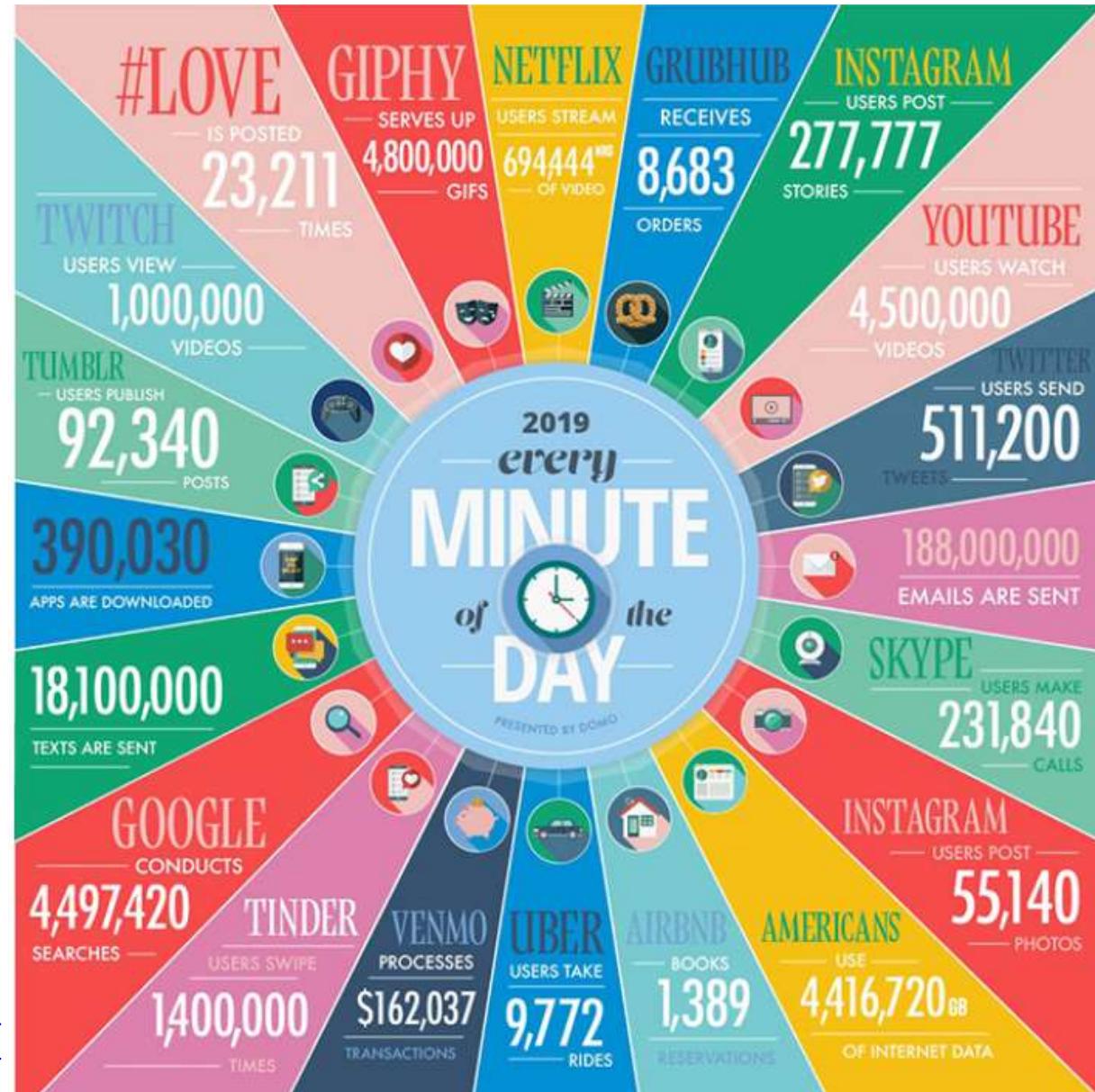
ESTO ES NUEVO ¿CIERTO?



¿QUÉ ES?

- “The science of getting computers to act without being explicitly programmed.” Standford
- Sistemas que encuentren patrones, relaciones y regularidades sobre los datos y aplicarlos para construir modelos descriptivos y predictivos
- Son matemáticas! (Bajo un increíble poder de procesamiento)
- No van a reemplazar al ser humano:
 - Conocimiento del Negocio
 - Evaluar descubrimientos
 - Mejora Contínua

EN UN MINUTO...



<https://www.domo.com/learn/data-never-sleeps-7>
<https://www.domo.com/learn/data-never-sleeps-8>

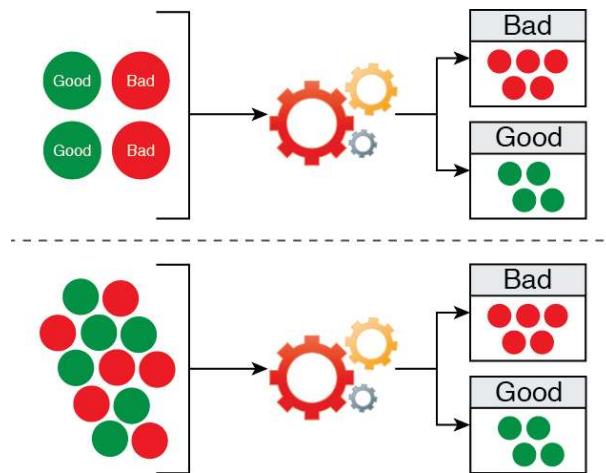
¿HASTA DONDE PUEDEN LLEGAR?



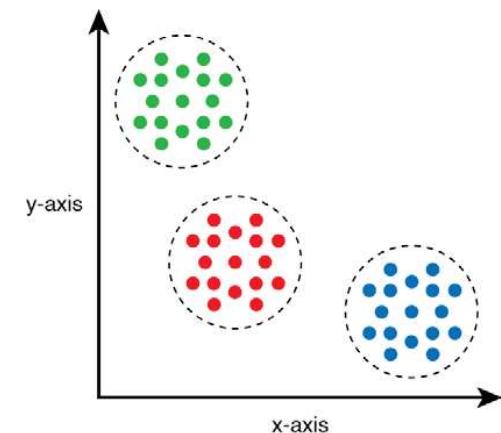
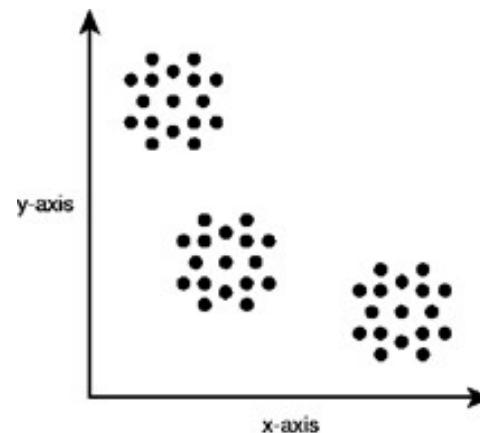
¿QUÉ PUEDEN HACER?

Aprendizaje Supervisado

vs Aprendizaje No Supervisado



- Clasificación
- Predicción



- Segmentación
- Análisis de Componentes Principales

MACHINE LEARNING?

Traditional programming



Machine learning



¿QUÉ NECESITAMOS?

DATOS
VARIABLES
ALGORITMO
AFINAMIENTO
MODELO!



¿MÁS VARIABLES?

- Estandarizar/Normalizar
 - Variables en el mismo rango
 - Común: Restar la media:
 - Media = 0 y Varianza = 1
 - Definir un rango [0,1], [-1,1]
 - Ya existen librerías para esto (Python, R)

¿MÁS VARIABLES?

- Variables Categóricas

Fecha Producción	Producción	Temperatura ...	Mes
2015-02-01	515	73 ...	Febrero
2001-11-01	629	76 ...	Septiembre
...
2010-11-01	603	74 ...	Noviembre

Fecha Producción	Producción	Mes[Enero]	Mes[Febrero] ...	Mes[Noviembre]
2015-02-01	515	0	1 ...	0
2001-11-01	629	0	0 ...	0
...
2010-12-01	603	0	0 ...	0

¿MÁS VARIABLES?

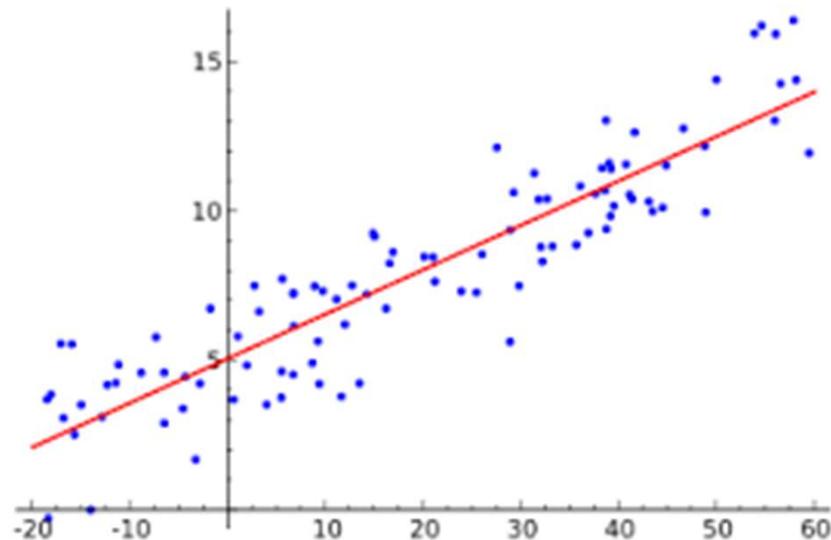
- Series de Tiempo

	month_dt	Produccion	prev_month
1	2010-02-01	648.0	515.0
2	2010-03-01	629.0	648.0
3	2010-04-01	647.0	629.0
4	2010-05-01	822.0	647.0
5	2010-06-01	780.0	822.0

BREAK – PERO ANTES:

```
MÓDULO 5 APRENDIZAJE AUTOMATICO
└── Python
    ├── LabXXpy_Nombre.ipynb
    └── LabXXpy_Nombre_solved.ipynb
└── R
    ├── LabXXr_Nombre.Rmd
    └── LabXXr_Nombre_solved.html
└── data
    └── ...
└── UN_Aprendizaje_Automatico.pdf
```

¿QUÉ ES UN “MODELO”?



- **Datos vs Modelo**
- Cada atributo es una dimensión
- En A. Supervisado, el atributo a predecir, también es una dimensión
- En A. No Supervisado, se busca la cercanía de los “puntos”
- Se divide el set de datos en 2: entrenamiento y prueba

¿ENTRENAMIENTO?

Histórico de ventas (base de datos actual)

Producto	Categoría	Precio U.	Día	Tienda	...	Unidades
001	Gaseosa	\$1000	Lunes	A	...	100
001	Gaseosa	\$900	Viernes	A	...	250
002	Lácteos	\$1200	Sábado	A	...	150
002	Lácteos	\$2000	Lunes	B	...	110
003	Aseo	\$1000	Domingo	B	...	200
003	Aseo	\$1500	Lunes	B	...	120
...

Nuevo análisis producto...

Prodcuto	Categoría	Precio U.	Día	Tienda	...	Unidades
002	Lácteos	\$1500	Lunes	C	...	?

The diagram shows a yellow robot with a speech bubble containing a question mark. A vertical line connects the robot to three horizontal arrows labeled 'Entrenar' (Train), 'Probar' (Test), and 'Predecir' (Predict). The 'Entrenar' arrow points up to a gear icon. The 'Probar' arrow points down from the robot. The 'Predecir' arrow points down to the question mark in the table.

¿ENTRENAMIENTO?

Entrenamiento



¿ENTRENAMIENTO?

Entrenamiento

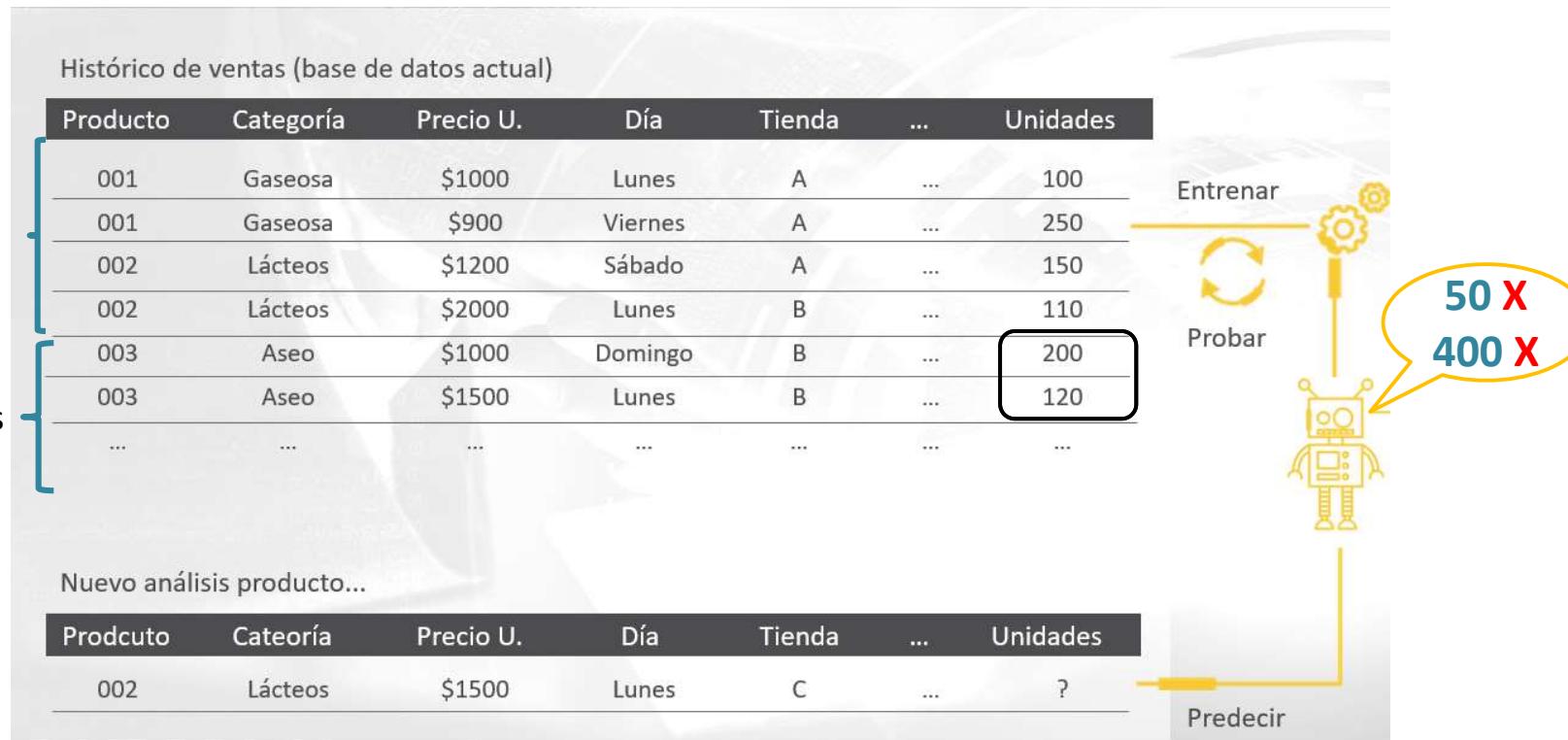
Pruebas



¿ENTRENAMIENTO?

Entrenamiento

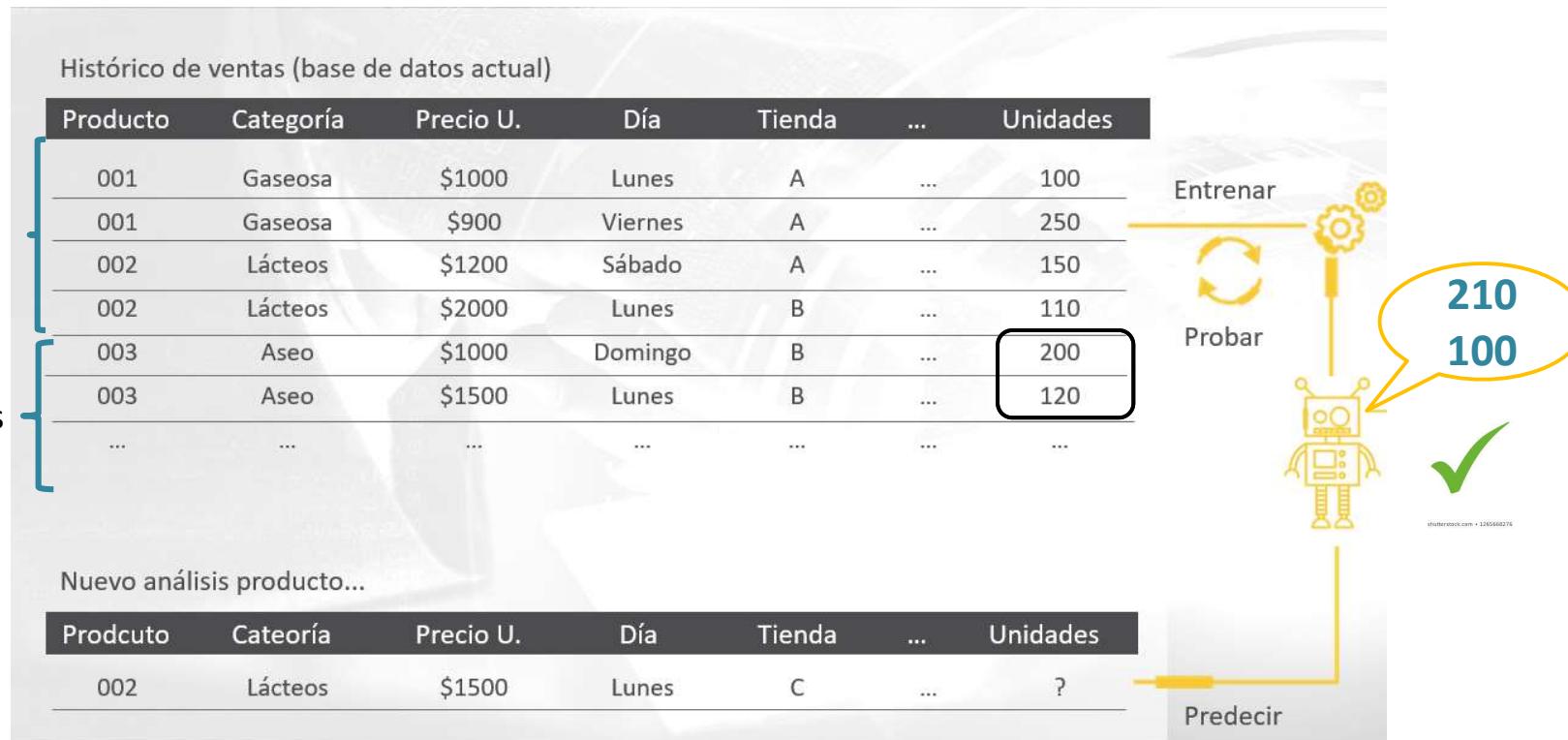
Pruebas



¿ENTRENAMIENTO?

Entrenamiento

Pruebas



¿ENTRENAMIENTO?

Entrenamiento

Pruebas

Histórico de ventas (base de datos actual)						
Producto	Categoría	Precio U.	Día	Tienda	...	Unidades
001	Gaseosa	\$1000	Lunes	A	...	100
001	Gaseosa	\$900	Viernes	A	...	250
002	Lácteos	\$1200	Sábado	A	...	150
002	Lácteos	\$2000	Lunes	B	...	110
003	Aseo	\$1000	Domingo	B	...	200
003	Aseo	\$1500	Lunes	B	...	120
...

Nuevo análisis producto...

Prodcuto	Categoría	Precio U.	Día	Tienda	...	Unidades
002	Lácteos	\$1500	Lunes	C	...	?

Entrenar

Probar

200!

Predecir

LAB 1 - ¿QUÉ VAMOS A HACER?

- Conocer los datos!
- Es lo más importante antes de entrenar un modelo
- Si el modelo se alimenta con “basura”, entregará.... Basura
- **Calidad de Datos:**
 - Datos Faltantes
 - Errores
 - Atípicos
- Correlación

¿QUÉ MÁS SE PUEDE HACER?

- Predicción y Clasificación, pero ¿Qué más?
- Encontrar Patrones y Estructura:
 - Sistemas de Recomendación
 - Procesamiento de Lenguaje Natural
 - Extracción de Entidades
 - Análisis de Sentimiento
 - Generación Corrección de textos
- Análisis de Imágenes
 - Detección de Objetos



¿DÓNDE PUEDO LEER MÁS?

- <https://www.dataversity.net/a-brief-history-of-machine-learning>
- <https://www.doc.ic.ac.uk/~jce317/history-machine-learning.html>
- <https://www.viatech.com/en/2018/05/history-of-artificial-intelligence/>
- <https://www.domo.com/learn/data-never-sleeps-8>
- <https://www.educba.com/supervised-learning-vs-unsupervised-learning/>

APRENDIZAJE AUTOMÁTICO

M2. CLASIFICACIÓN Y PREDICCIÓN

Mónica Ramírez Bernal

2020-2

Centro de Investigaciones para el Desarrollo – CID
Programa de Educación permanente - PEC
Facultad de Ciencias Económicas
Sede Bogotá

educación continua 



UNIVERSIDAD
NACIONAL
DE COLOMBIA

¿SON DIFERENTES?... NO TANTO...

PREDICCIÓN

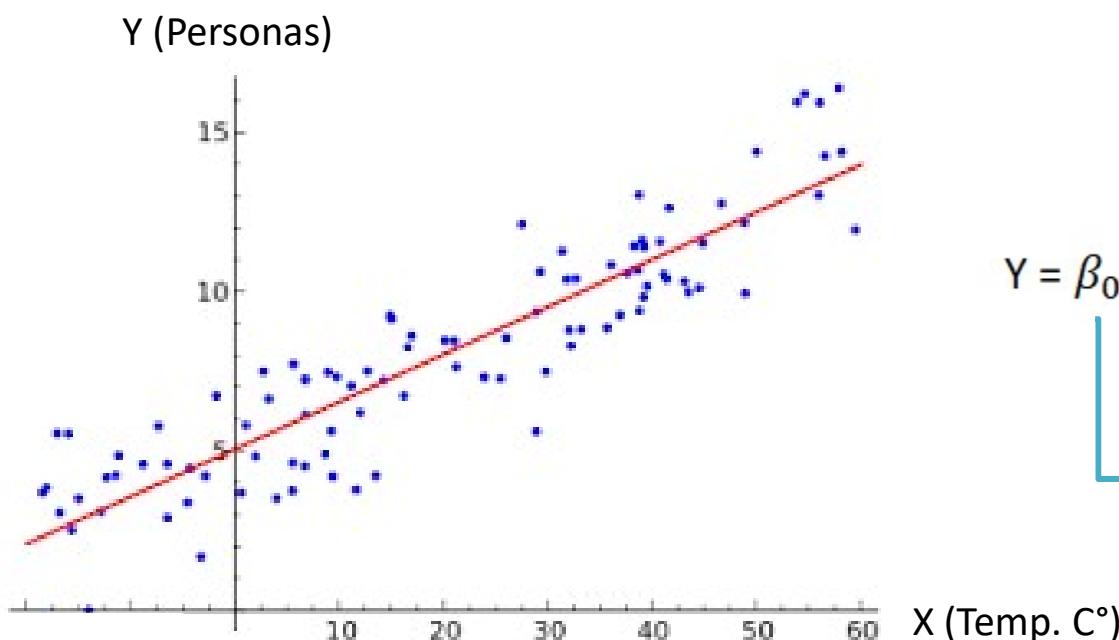
- Sobre variables continuas:
 - Temperatura mañana: 27 °C
 - Monto préstamo: \$50M
 - Cantidad de nominaciones
 - Precio de una casa
 - Demanda productos
 - Unidades de Producción

CLASIFICACIÓN

- Sobre variables categóricas:
 - Lloverá mañana? SI/NO
 - Dar el préstamo? SI/NO
 - Va a ganar el Óscar? SI/NO
 - Estrato de una casa (1-6)
 - Medio de Transporte
 - Reconocimiento de imagen

CASO MÁS SIMPLE...

PREDICCIÓN



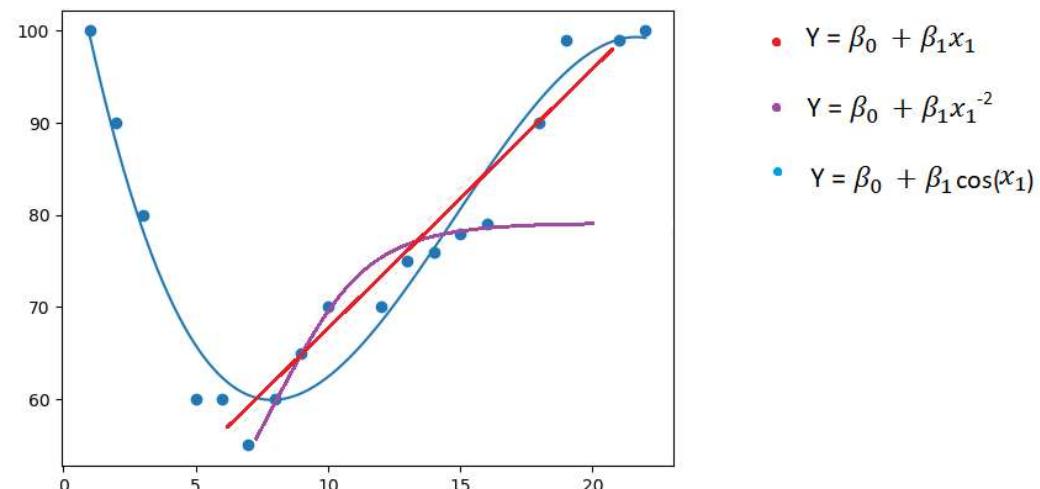
$Y = \beta_0 + \beta_1 x_1$ → Variable
→ Peso/Coeficiente
→ Sesgo/Intercepto

PERO EN LA VIDA REAL...

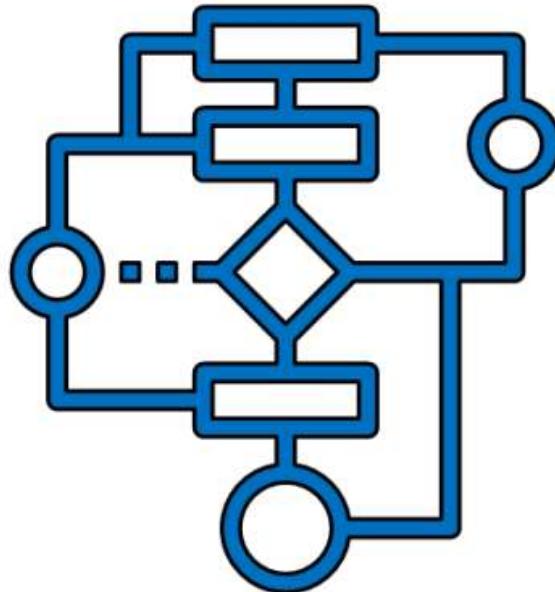
- Qué pasa si son muchas variables?

$$Y = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \dots + \beta_n x_n$$

- Siempre es una relación lineal?



¿QUÉ ES UN ALGORITMO?



- Un **Algoritmo** es un método estandarizado para entrenar un **Modelo**
- Un **Modelo** es una función que mapea Input a un set de predicciones (Output) usando un **Algoritmo**
- Los datos son usados para construir el **Modelo** de manera que el input se transforme en el output
- Los **Algoritmos** definen las reglas de construcción del **Modelo**

LAB 2 - ¿QUÉ VAMOS A HACER?

- **Predicción (score)**

- Regresión Lineal
- OLS

- **Clasificación (high_quality)**

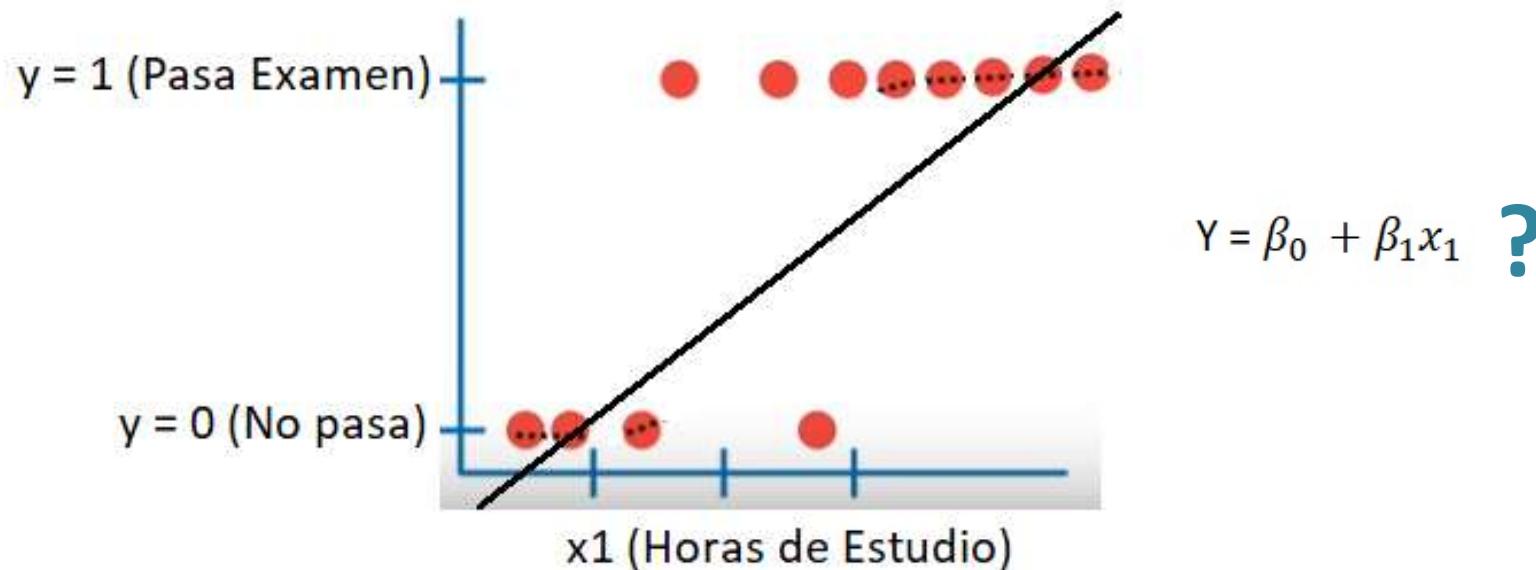
- Regresión Logística
- SVM
- kNN

- **Ensembles**

- Árboles de Decisión
- Bosques Aleatorios

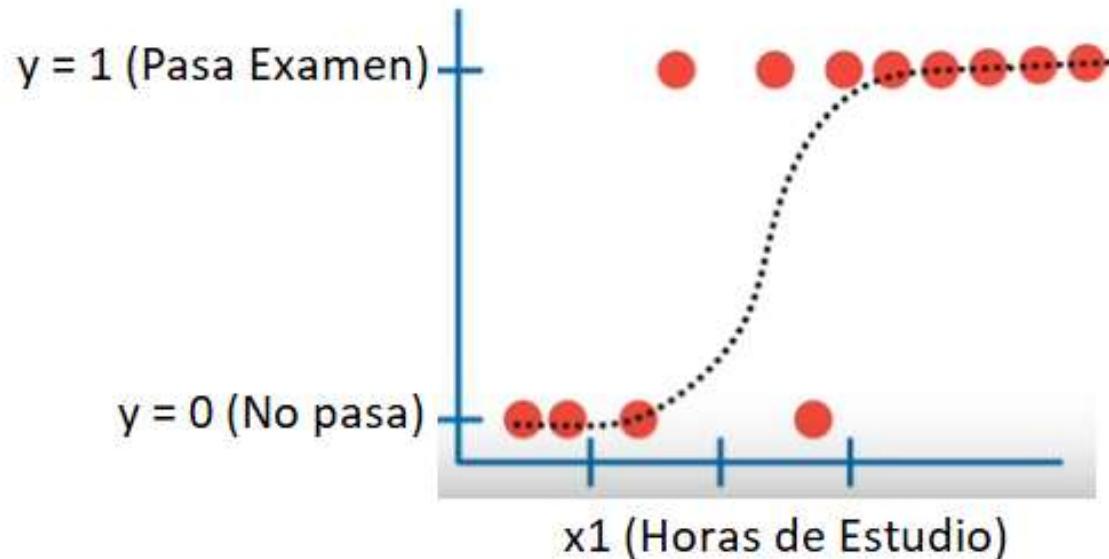
CASO MÁS SIMPLE...

CLASIFICACIÓN



CASO MÁS SIMPLE...

CLASIFICACIÓN



$$y_{\beta}(x) = \frac{1}{1 + e^{-(\beta_0 + \beta_1 x + \varepsilon)}}$$

- Término conocido!
- Valor [0-1] = Probabilidad
- Más variables?

¿Y SI SON VARIAS CLASES?

- **Algunos Algoritmos lo pueden manejar**
- **Binario - One versus All (OVA)**

- Bueno vs No Bueno (Regular + Malo)
- Regular vs No Regular (Bueno + Malo)
- Malo vs No Malo (Bueno + Regular)

} N modelos

- **Binario - One versus One (OVO)**

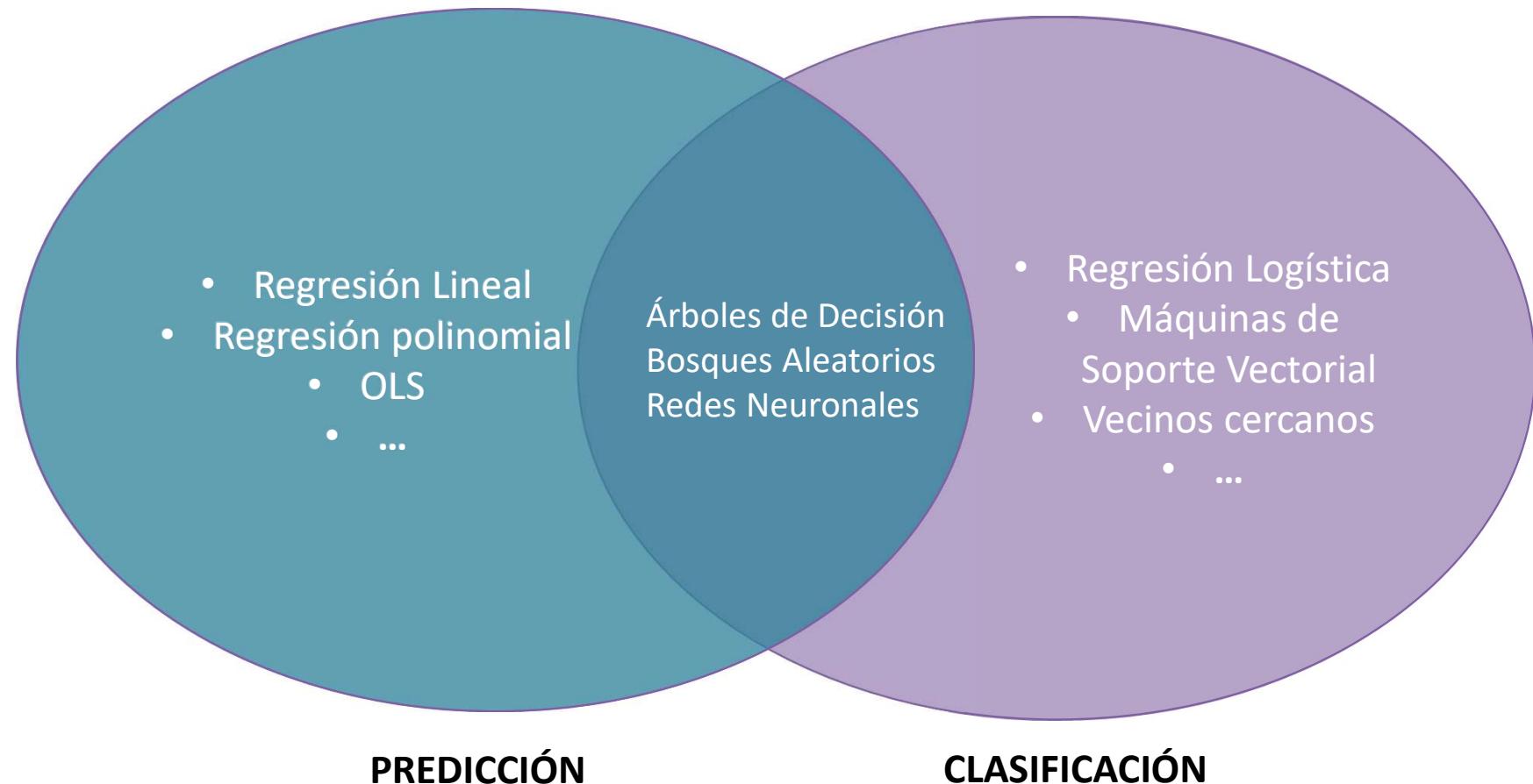
- Bueno vs Regular
- Bueno vs Malo
- Regular vs Malo

} Nx(N-1)/2 modelos

LAB 2 - ¿QUÉ VAMOS A HACER?

- **Predicción (score)**
 - Regresión Lineal
 - OLS
- **Clasificación (high_quality)**
 - Regresión Logística
 - SVM
 - kNN
- **Ensembles**
 - Árboles de Decisión
 - Bosques Aleatorios

EN RESUMEN...



UNA ÚLTIMA COSA...

Regresión Polinomial

Grado Polinomio

Regresión Logística

Solver

Árboles de Decisión

Profundidad Max.

Cant. Ejemplos Min/Hoja

Cant. Máx. Hojas

kNN

#Vecinos

SVM

Kernell

Grado Poly

C (Penalidad)

Gamma

Árboles Aleatorios

Profundidad Max.

Cant. Ejemplos Min/Hoja

Cant. Máx. Hojas

Árboles

UNA ÚLTIMA COSA...

Regresión Polinomial

Grado Polinomio

Regresión Logística

Solver

HIPERPARÁMETROS

Árboles de Decisión

Profundidad Max.

Cant. Ejemplos Min/Hoja

Cant. Máx. Hojas

kNN

#Vecinos

Árboles Aleatorios

Profundidad Max.

Cant. Ejemplos Min/Hoja

Cant. Máx. Hojas

Árboles

SVM

Kernell

Grado Poly

C (Penalidad)

Gamma

¿DÓNDE PUEDO LEER MÁS?

- <https://www.iartificial.net/clasificacion-o-regresion/>
- <https://machinelearningmind.com/2019/11/25/logistic-regression-explained/>
- <https://statisticsbyjim.com/regression/ols-linear-regression-assumptions>
- <https://data-flair.training/blogs/svm-support-vector-machine-tutorial>
- <https://towardsdatascience.com/interpretability-and-random-forests-4fe13a79ae34>
- <https://www.mathworks.com/videos/applied-machine-learning-part-3-hyperparameter-optimization-1547849445386.html>
- <https://machinelearning-blog.com/2018/04/23/logistic-regression-101/>
- <https://www.dezyre.com/article/top-10-machine-learning-algorithms/202>

APRENDIZAJE AUTOMÁTICO

M3. VALIDACIÓN DE MODELOS

Mónica Ramírez Bernal

2020-2

Centro de Investigaciones para el Desarrollo – CID
Programa de Educación permanente - PEC
Facultad de Ciencias Económicas
Sede Bogotá

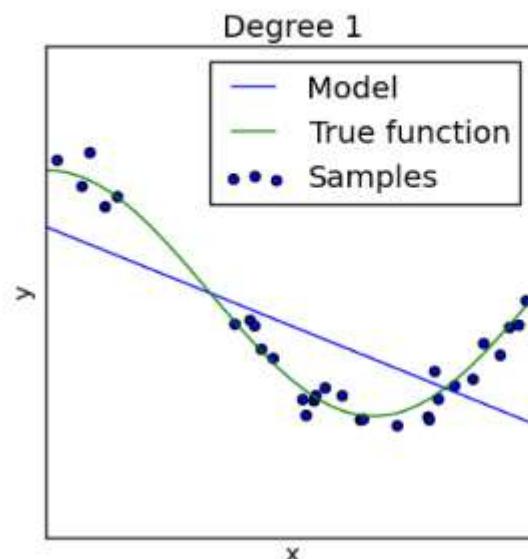
educación continua 



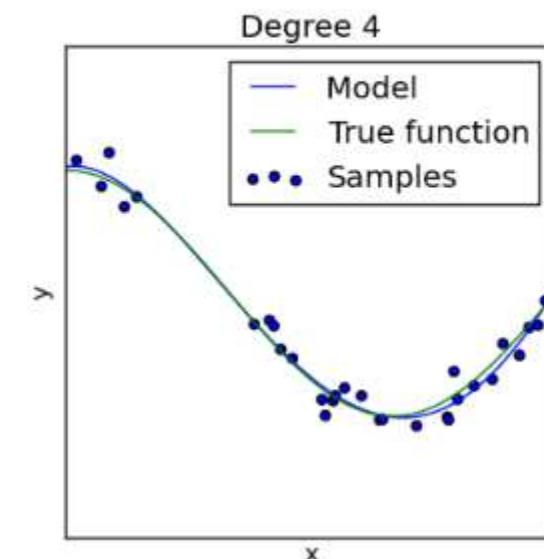
UNIVERSIDAD
NACIONAL
DE COLOMBIA

CÓMO SABER QUÉ UN MODELO ES BUENO?

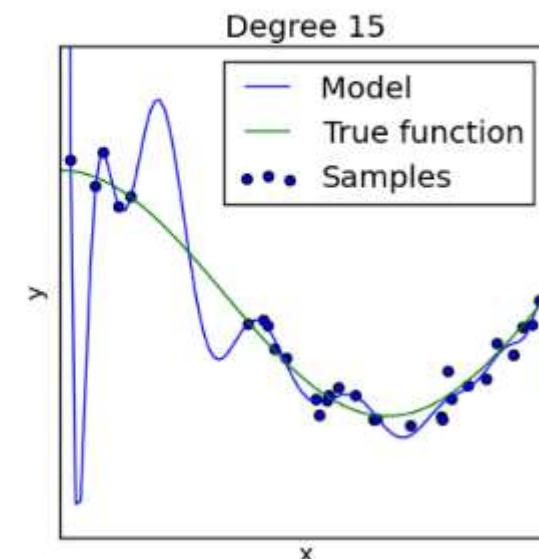
Underfitting



Just Right



Overfitting

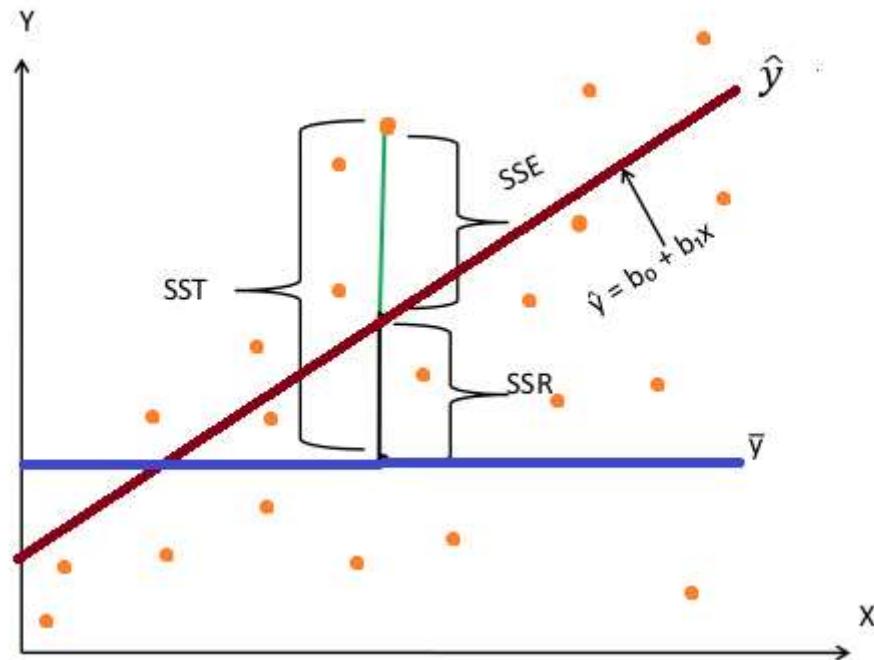


Subajuste

Modelo justo

Sobreajuste

Y... CÓMO MEJORARLO? (PREDICCIÓN)



$$SSE = \sum (y_i - \hat{y})^2$$

$$SST = \sum (y_i - \bar{y})^2$$

$$SSR = \sum (\hat{y}_i - \bar{y})^2$$

$$R^2 = 1 - \frac{SSE}{SST}$$

CASTIGAR AL MODELO!

- Si se equivoca (Error):

$$SSE = \sum (y_i - \hat{y})^2$$

- Si se demora

- Complejidad = Cantidad de Variables $f(c)$

Ecuación Costo Final:

$$J = \sum (y_i - \hat{y})^2 + f(c)$$

Subajuste

Alto + Bajo

Modelo justo

Bajo + Bajo

Sobreajuste

Bajo + Alto

CASTIGAR COMPLEJIDAD (REGULARIZACIÓN)

- **Lasso (L1):** Bajar Coeficientes a 0

$$f(\mathbf{c}) = \lambda \sum_{j=1}^k |\beta_j|$$

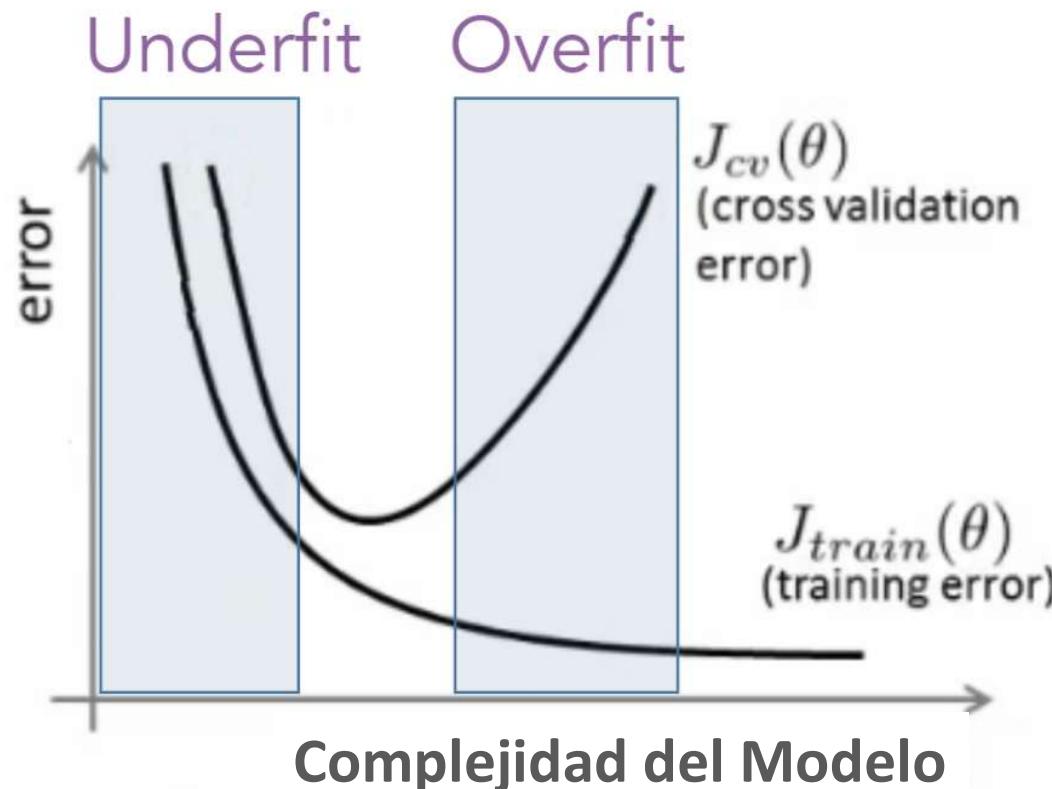
- **Ridge (L2):** Disminuir coeficientes

$$f(\mathbf{c}) = \lambda \sum_{j=1}^k \beta_j^2$$

- **Otros:**

- ElasticNet

Y CÓMO VISUALIZAR SI VAMOS BIEN?



LAB 3 - ¿QUÉ VAMOS A HACER?

- **Predicción (score)**

- Curvas de Validación
- Regularización

- **Clasificación (high_quality)**

- Matriz de Confusión
- Curva ROC
- Área bajo la Curva (AUC)
- Ajuste del umbral

SOBREALAJSTE EN CLASIFICACIÓN. ES IGUAL?

- **Modelo de Mercadeo (Compra o no?)**

- Datos Reales:

1	2	3	4	5	6	7	8	9	10
NO	NO	SI	SI	NO	SI	NO	NO	SI	NO

- Predicción:

1	2	3	4	5	6	7	8	9	10
NO	SI	NO	SI	NO	NO	NO	NO	SI	SI

✗ ✗ ✗ ✗

Ok = 6/10
Score= 60%

- Modelo más "bobo":

1	2	3	4	5	6	7	8	9	10
NO									

✗ ✗ ✗ ✗

Ok = 6/10
Score= 60%

SOBREALJUSTE EN CLASIFICACIÓN. ES IGUAL?

- **Modelo de Detección de Cáncer**

- Datos Reales:

1	2	3	4	5	6	7	8	9	10
NO	NO	NO	SI	NO	NO	NO	NO	NO	NO

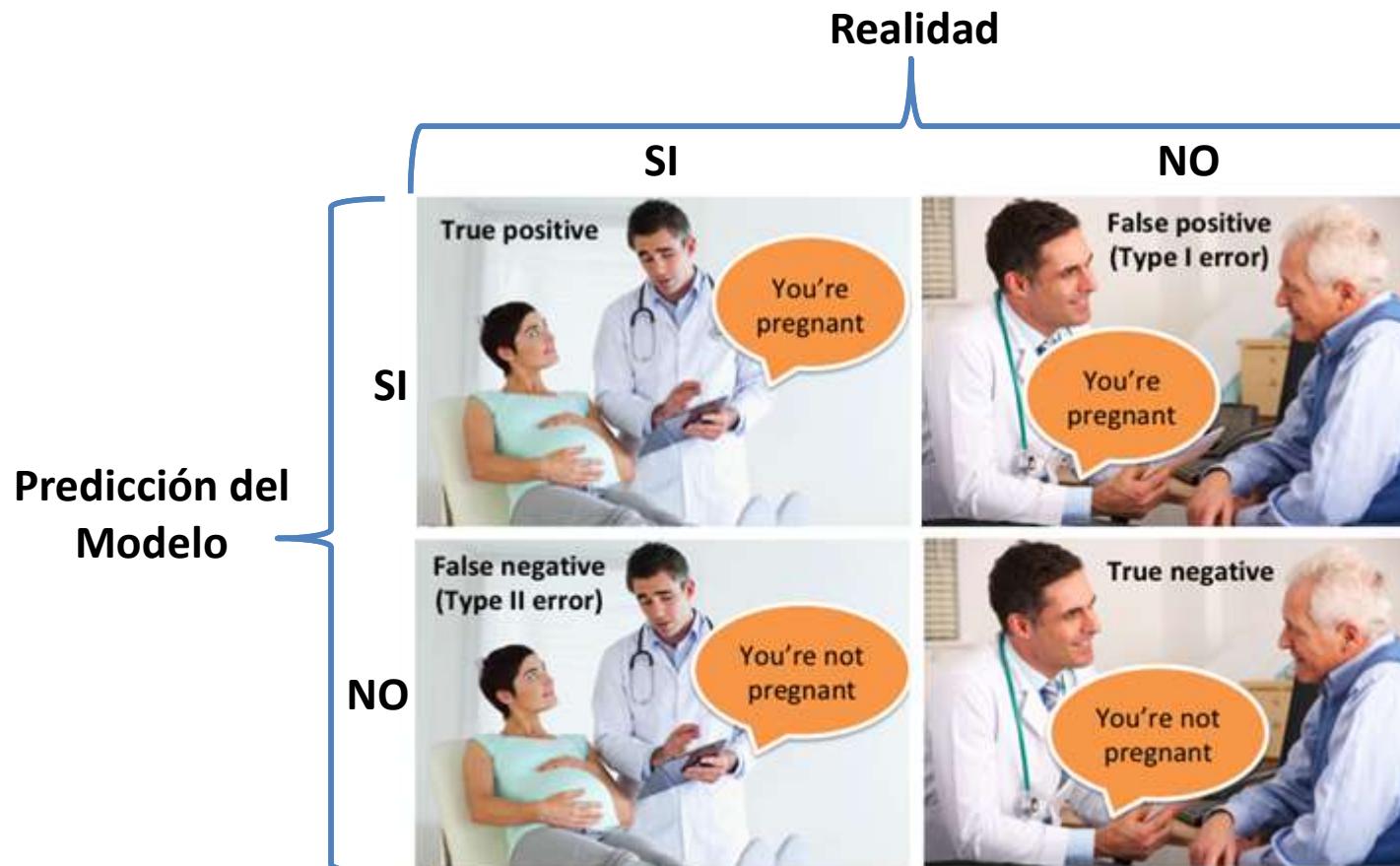
- Modelo más “bobo”:

1	2	3	4	5	6	7	8	9	10
NO									

X

Ok = 9/10
Score= 90%

HAY OTRO TIPO DE MÉTRICAS:



<https://data-flair.training/blogs/data-science-interview-questions/>

HAY OTRO TIPO DE MÉTRICAS:

- Accuracy

- (Score)

$$\frac{TP + TN}{TP + FP + TN + FN}$$

		Realidad	
		SI	NO
Modelo	SI	TP	FP
	NO	FN	TN

- % de predicciones correctas
(Positivas o Negativas)

HAY OTRO TIPO DE MÉTRICAS:

- Sensibilidad

- (Recall, TPR)

$$\frac{TP}{TP + FN}$$

		Realidad	
		SI	NO
Modelo	SI	TP	FP
	NO	FN	TN

- De todos los casos positivos, cuántos encontré?

HAY OTRO TIPO DE MÉTRICAS:

- Precisión

- (Positive Predicted Value)

		Realidad	
		SI	NO
Modelo	SI	TP	FP
	NO	FN	TN

$$\frac{TP}{TP + FP}$$

- De todos los que dije "SI". Cuántos eran ciertos?

HAY OTRO TIPO DE MÉTRICAS:

- **Especificidad**

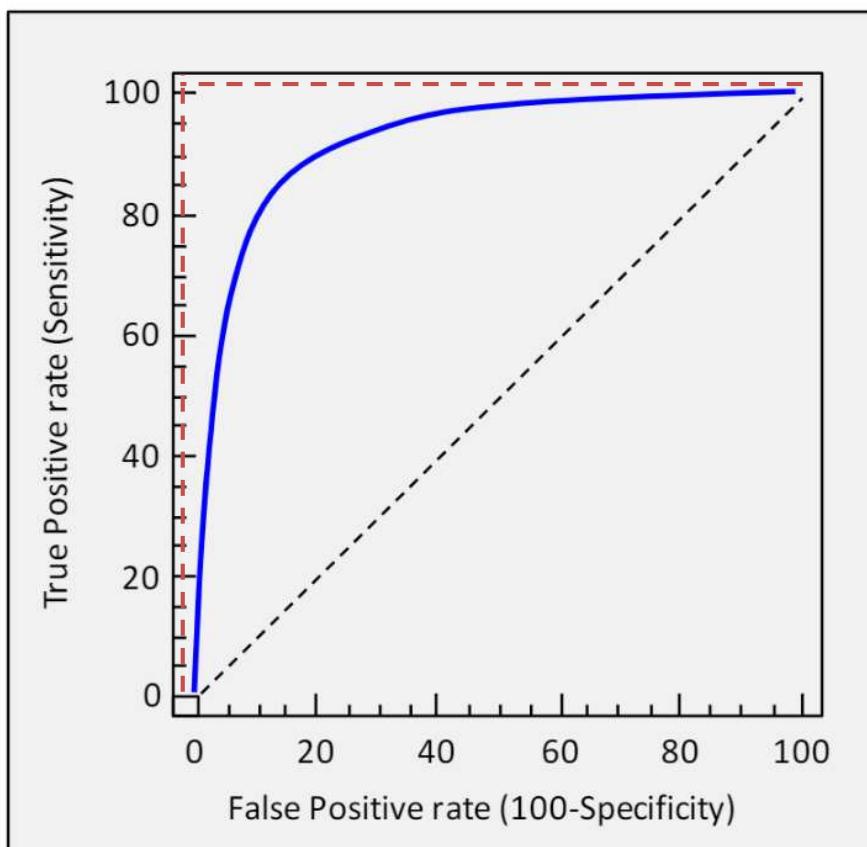
- (Selectivity, TNR)

$$\frac{TN}{TN + FP}$$

		Realidad	
		SI	NO
Modelo	SI	TP	FP
	NO	FN	TN

- De todos los casos negativos, cuántos acerté?

VALIDACIÓN GRÁFICA:



CURVA ROC:

----- Modelo "Perfecto"

----- Modelo Aleatorio

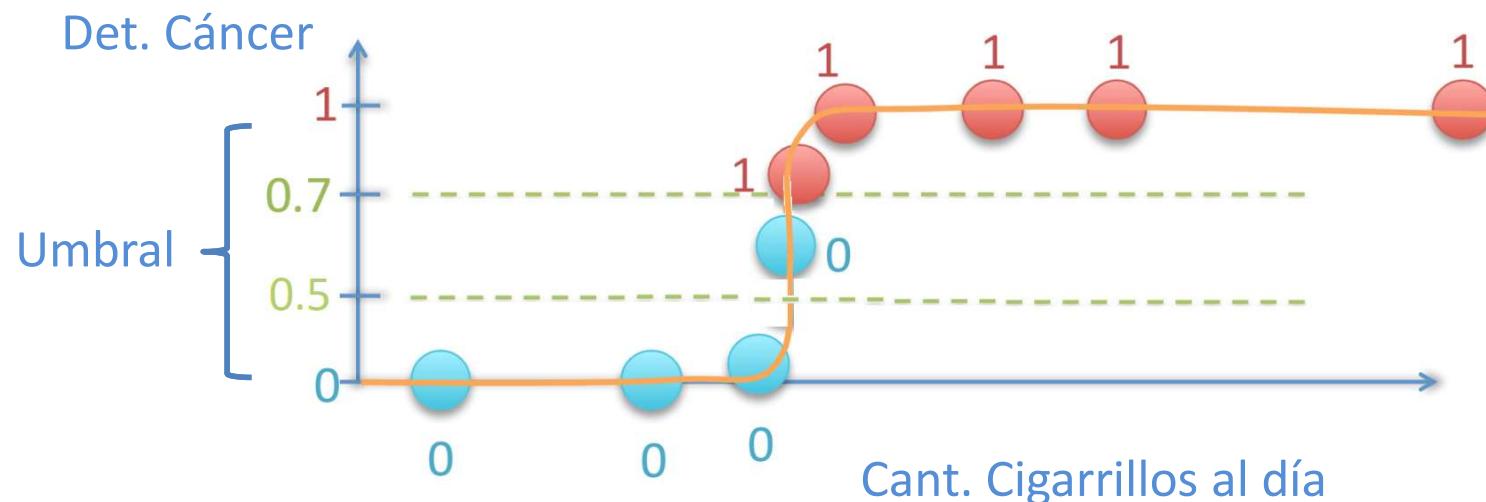
— Mi Modelo

ÁREA BAJO LA CURVA:

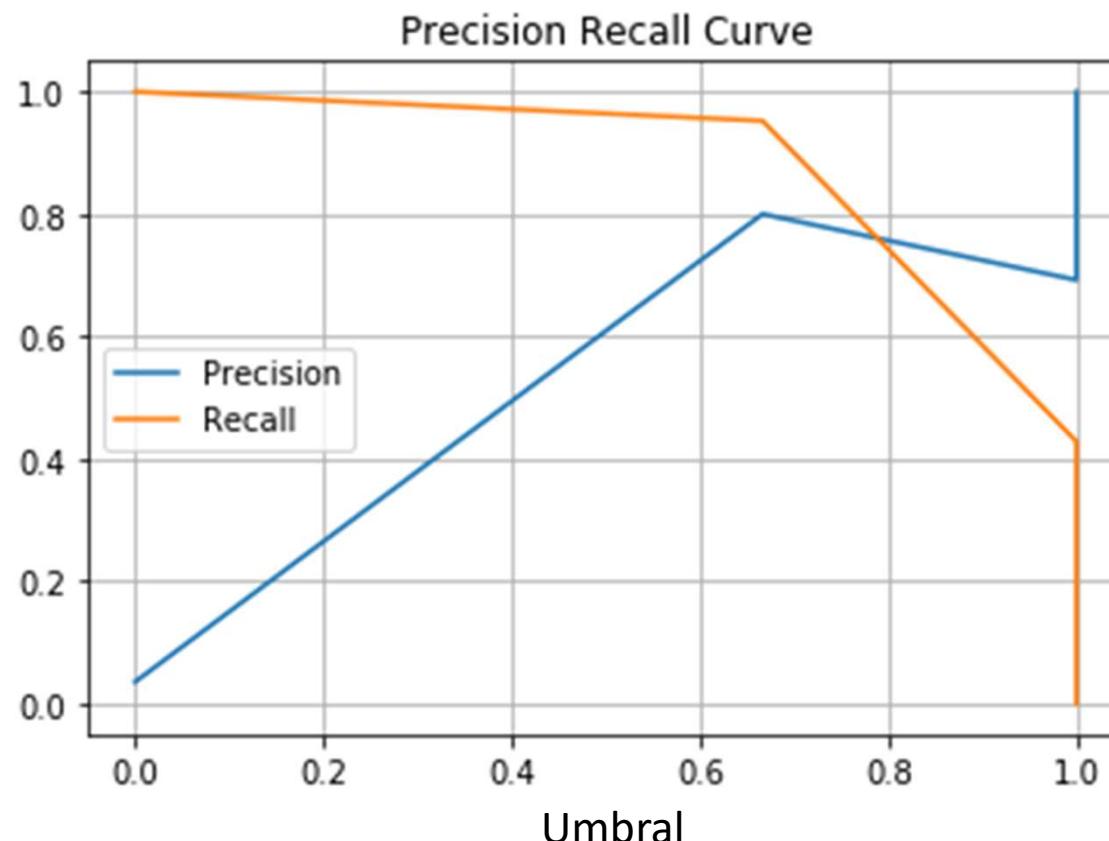
AUC ≈ 100

SUFICIENTE EN EL CASO DESBALANCEADO?

- $Y = f(x)$ es una probabilidad.
- Desde dónde decir SI/NO? (Umbral)



VALIDACIÓN GRÁFICA:



Y SI SON VARIAS CLASES?

- One vsOne (OVO)
- One vs Rest (OVR), One Vs All o One against All:

		Realidad			
		Bajo	Medio	Moderado	Alto
Modelo	Bajo	TP 14	89	FP 8	90
	Medio	45	278	23	5
	Moderado	FN 8	23	TN 14	12
	Alto	20	56	20	389

LAB 3 - ¿QUÉ VAMOS A HACER?

- Predicción (score)
 - Curvas de Validación
 - Regularización
- Clasificación (high_quality)
 - Matriz de Confusión
 - Curva ROC
 - Área bajo la Curva (AUC)
 - Ajuste del umbral

EL MODELO SIGUE MALO, QUÉ HAGO?

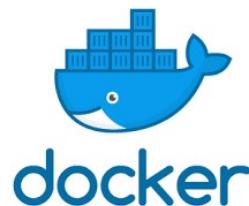
- Aumentar el Dataset
- Usar o generar más variables
- Probar varios modelos e Hiperparámetros
- Regularización o Polinomios
- Estandarizar datos
- Usar un conjunto más pequeño de variables
(Eliminar ruido)
- Hacer Sacrificios

¿DÓNDE PUEDO LEER MÁS?

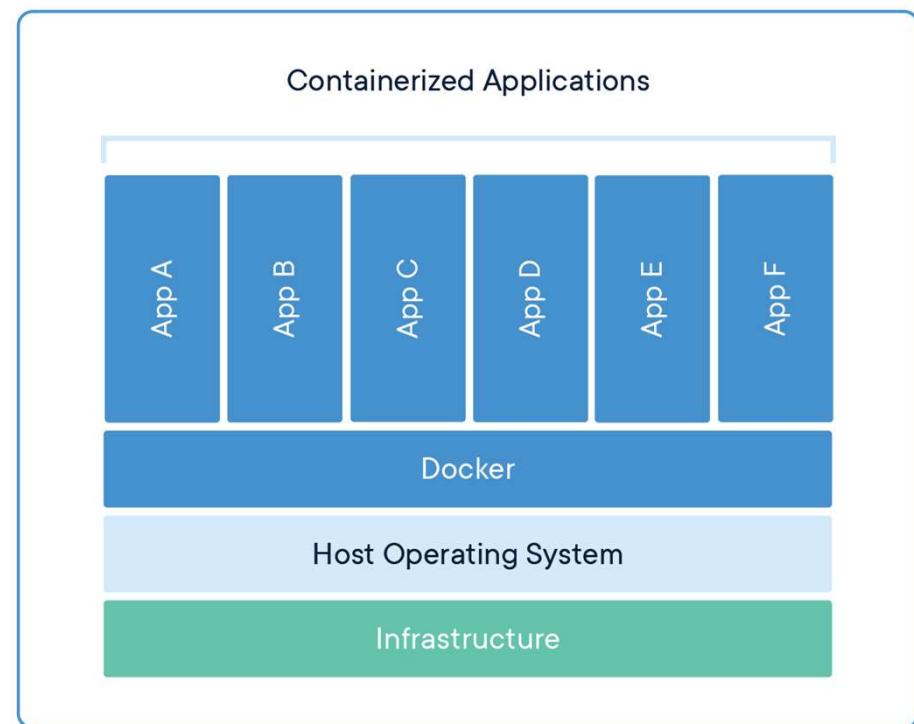
- <https://www.aprendemachinelearning.com/que-es-overfitting-y-underfitting-y-como-solucionarlo/>
- <https://elitedatascience.com/bias-variance-tradeoff>
- <https://www.ritchieng.com/machinelearning-learning-curve/>
- <https://machinelearningmastery.com/finalize-machine-learning-models-in-r/>
- <https://www.datascienceblog.net/post/machine-learning/interpreting-roc-curves-auc/>

PRÓXIMA SESIÓN: REDES NEURONALES!

- Librería keras:
 - Python = Jupyter/Colab
 - R = Rstudio/ **Docker**



```
library(keras)  
install_keras()  
model <- keras_model_sequential()
```



APRENDIZAJE AUTOMÁTICO

M4. REDES NEURONALES

Mónica Ramírez Bernal

2020-2

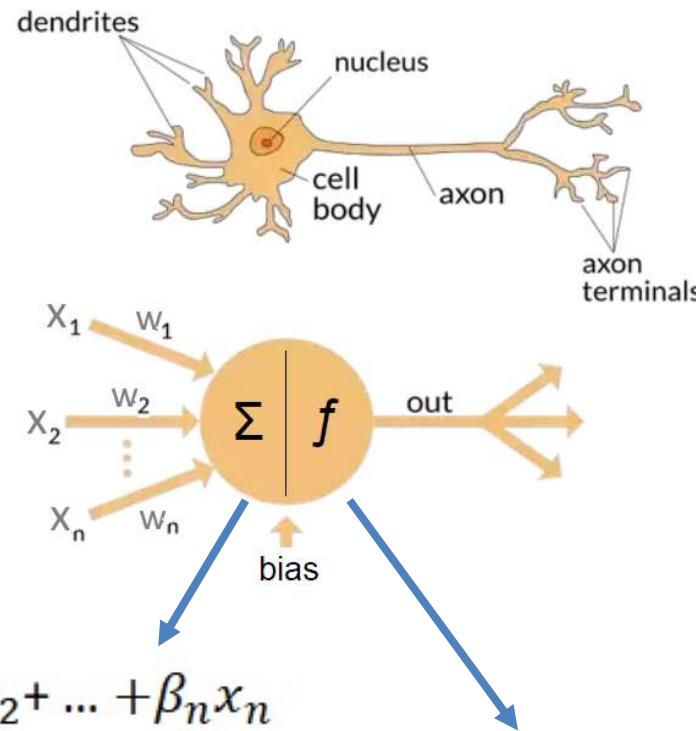
Centro de Investigaciones para el Desarrollo – CID
Programa de Educación permanente - PEC
Facultad de Ciencias Económicas
Sede Bogotá

educación continua 



UNIVERSIDAD
NACIONAL
DE COLOMBIA

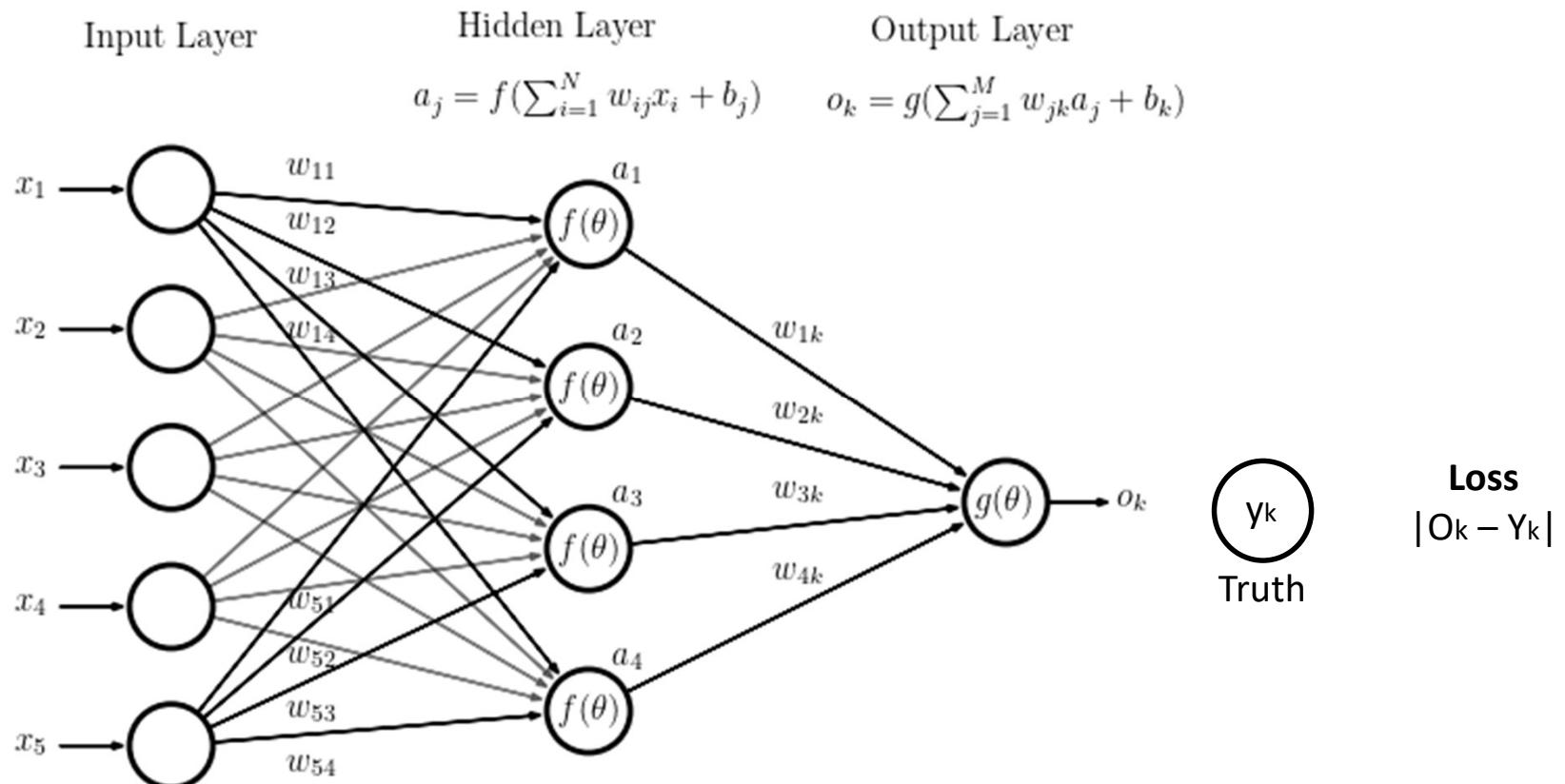
CASO MÁS SIMPLE...



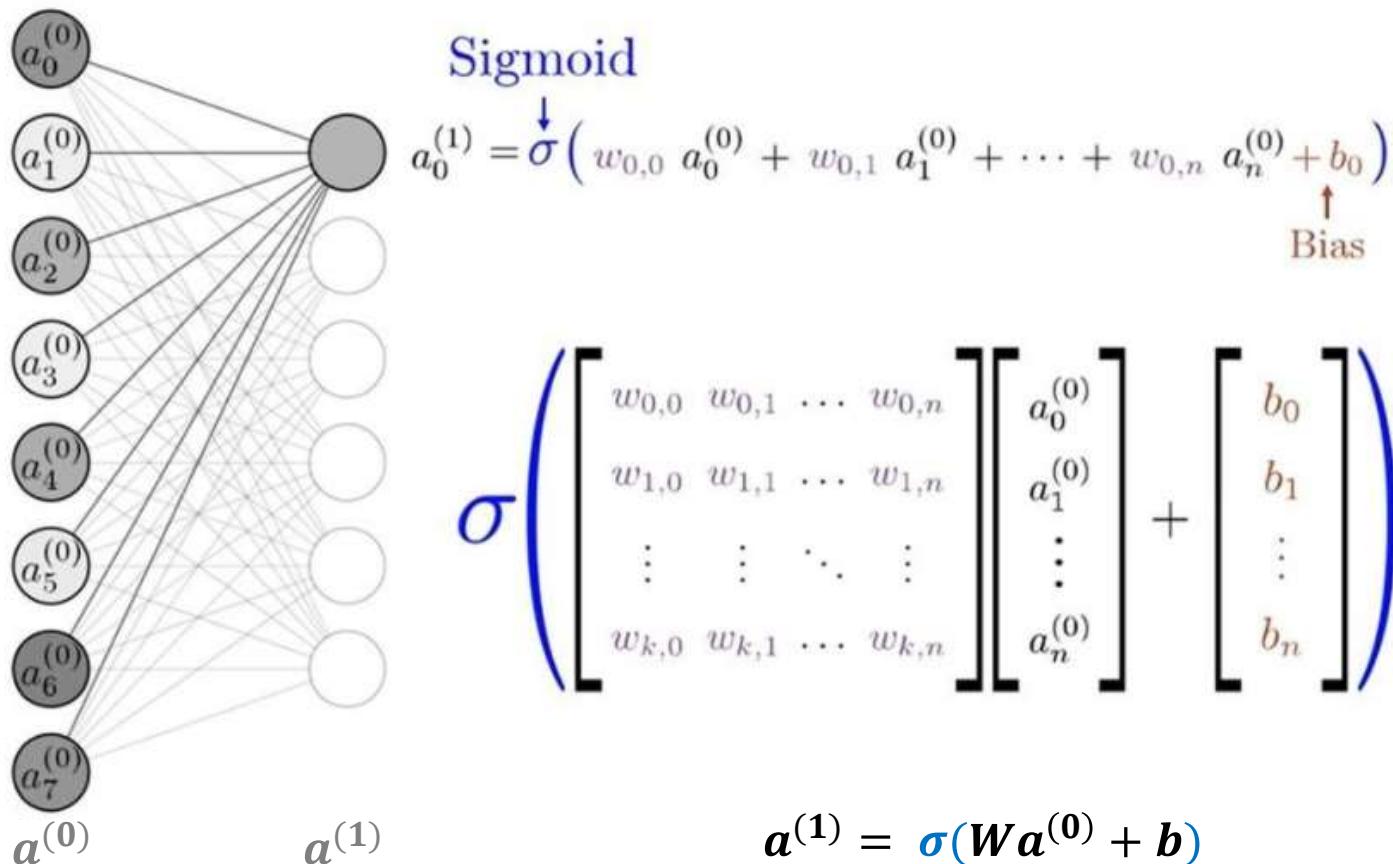
$$f(x) = w_0 + \sum_{i=1}^n w_i x_i$$

$$y = \varphi(w_0 + \sum_{i=1}^n w_i x_i)$$

CASO MÁS COMPLEJO...



PERO AL FIN Y AL CABO...



QUÉ SABEMOS HASTA AHORA?

- Son matemáticas pero muy complejas! (No explicativas)
- Aprendizaje Supervisado: Función de error
- Algunos Hiperparámetros:
 - # Capas Ocultas
 - # Nodos en cada Capa
- Función de Activación
- Extremadamente complejas (Computo y Tiempo)

FUNCIÓN DE ACTIVACIÓN

https://en.wikipedia.org/wiki/Activation_function

Name	Plot	Equation
Identity		$f(x) = x$
Binary step		$f(x) = \begin{cases} 0 & \text{for } x < 0 \\ 1 & \text{for } x \geq 0 \end{cases}$
Logistic (a.k.a. Sigmoid or Soft step)		$f(x) = \sigma(x) = \frac{1}{1 + e^{-x}}^{[1]}$
TanH		$f(x) = \tanh(x) = \frac{(e^x - e^{-x})}{(e^x + e^{-x})}$
Rectified linear unit (ReLU) ^[11]		$f(x) = \begin{cases} 0 & \text{for } x < 0 \\ x & \text{for } x \geq 0 \end{cases}$
Softmax		$f_i(\vec{x}) = \frac{e^{x_i}}{\sum_{j=1}^J e^{x_j}} \quad \text{for } i = 1, \dots, J$

Regresión

(0 o 1) Clasificación Binaria

[0, 1] Probabilidad (RegLog!)

[-1, 1] Clasificación Binaria

(0, ∞) Regresión

Clasificación Multiclas

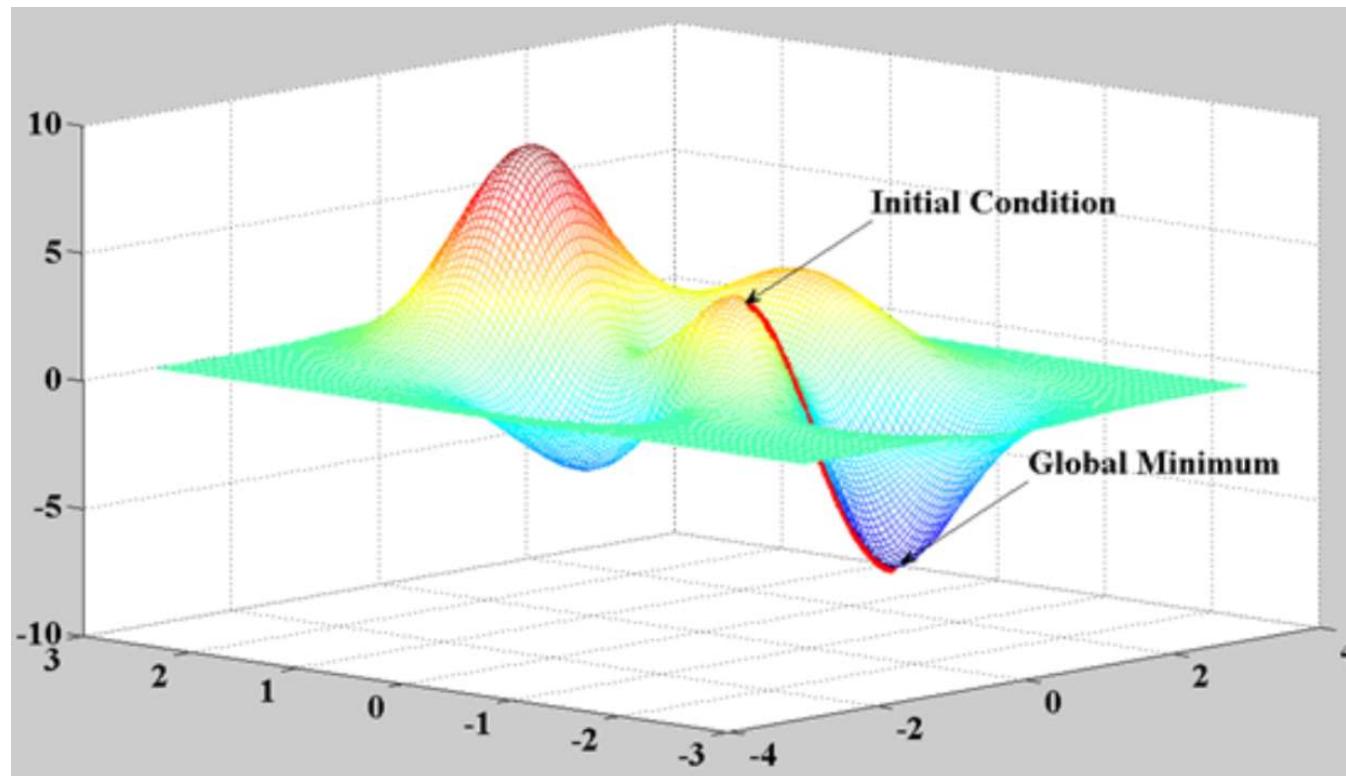
ESA ECUACIÓN TAN GRANDE, CÓMO SE OPTIMIZA?

No esta $a^{(1)} = \sigma(Wa^{(0)} + b)$ sino la COMPLETA:

- Inicio: W = Valores random
- Calcular la probabilidad de cada Neurona final con un caso de prueba
- Calcular la distancia de esa probabilidad al valor real = Costo
- Calcular el Costo promedio de todos los casos de prueba = Esto es lo que minimizo!

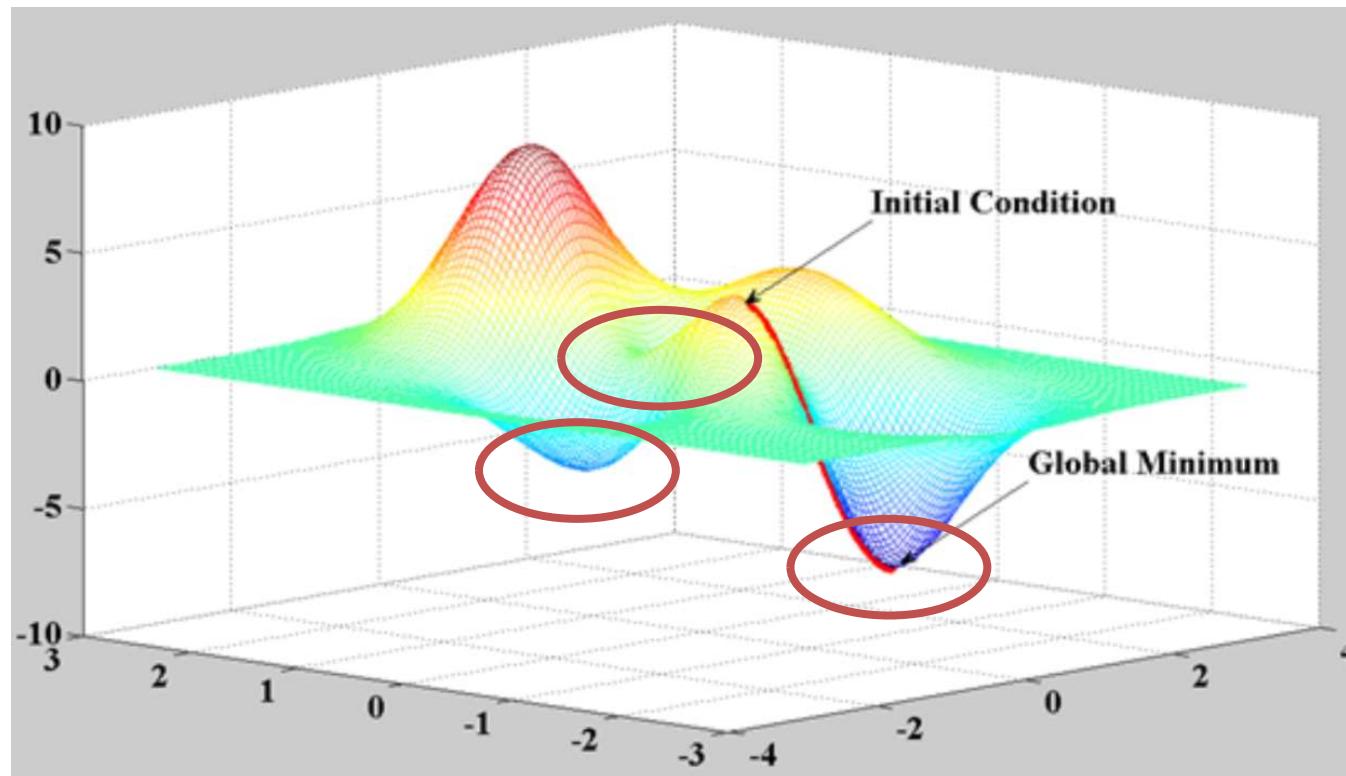
DESCENSO DE GRADIENTE

$$J(b_0, w_0) = \sum((b_0 + x_i w_0) - y_i)^2$$



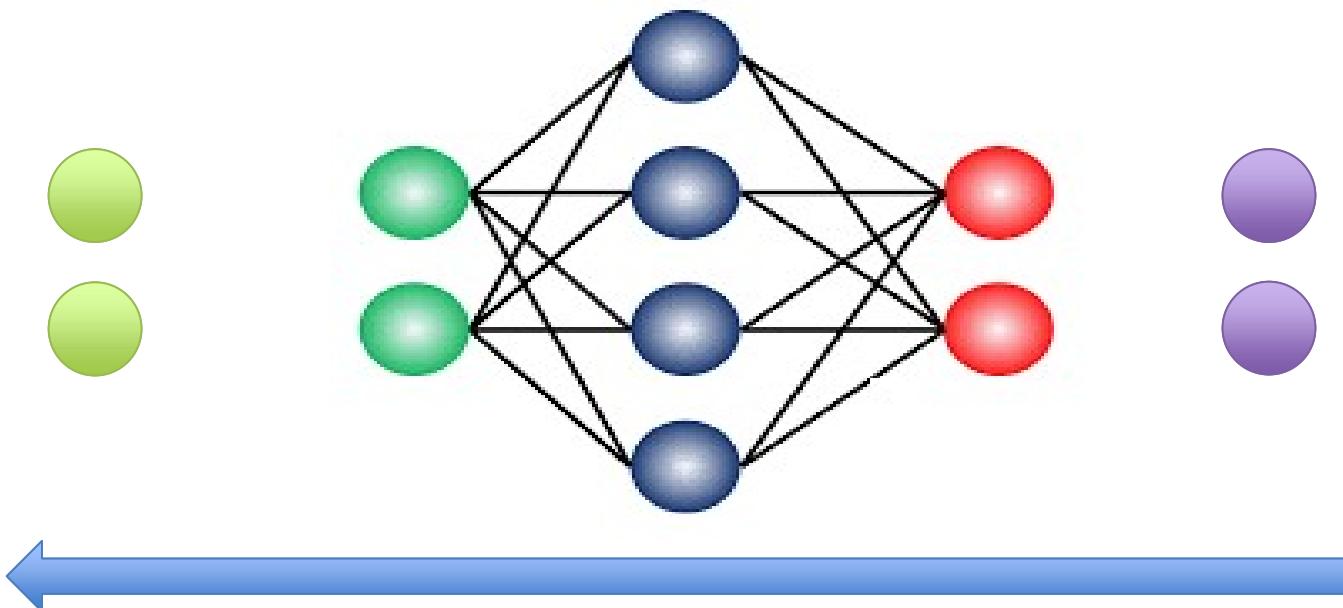
DESCENSO DE GRADIENTE

- Las Redes Neuronales son NO determinísticas!



OTRA FORMA: RETROPROPAGACIÓN

Input Real Input Modelo Modelo Salida Realidad/Labels



4. Calcular la “distancia” de la primera capa a las entradas reales

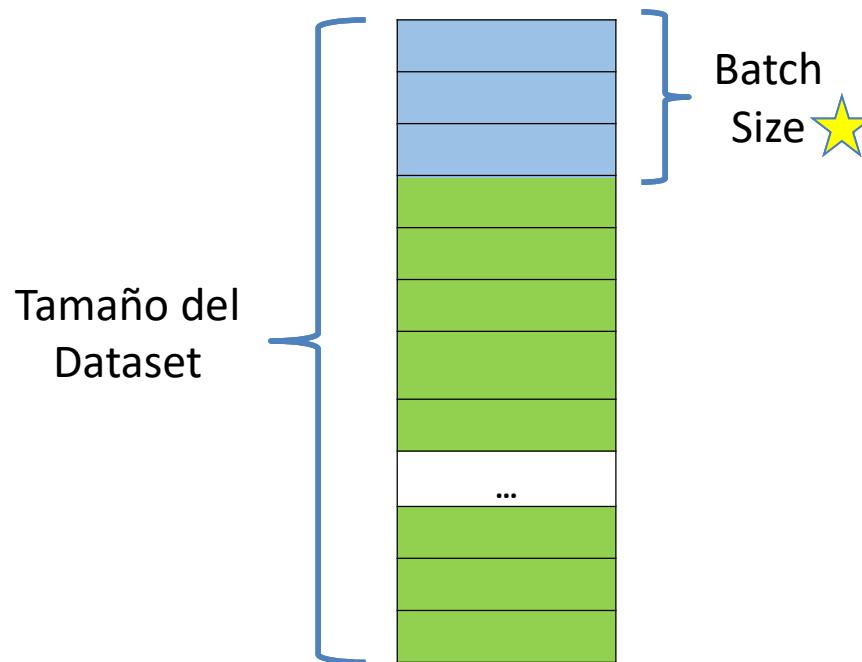
3. ... cómo hay que modificar la capa anterior? Y la anterior...

2. Para que la última neurona diga lo que queremos...

1. Partir de lo que queremos obtener

Y PARA QUE NO SE DEMORE TANTO?

- Descenso de Gradiente Estocástico: Hacer el mismo proceso pero con “porciones” de datos (Batches)



- # Pasos: Entrenar el modelo en 1 batch★
- # Épocas: Entrenar el modelo en todo el dataset★
- Learning Rate (LR): Tamaño del “paso” del descenso de gradiente★

★ HIPERPARÁMETROS!

ANTES DEL BREAK...

- R:

```
library(keras)  
install_keras()  
model <- keras_model_sequential()
```



- Python:

```
import keras  
model = keras.Sequential()
```



LAB 4 - ¿QUÉ VAMOS A HACER?

- **TensorFlow Playground!**
 - Clasificación
 - Regresión
- **Mi primera Red Neuronal**

Y CÓMO SE HACE TODO ESO EN CÓDIGO?

1. Definir el Modelo

2. Compilar el Modelo

3. Entrenar el Modelo

1. DEFINIR EL MODELO

- Construir la Red = Definir cada capa (# Neuronas y F. Activación) y secuencia
- Cada Capa es una función que transforma la salida de la anterior
- Cantidad de Neuronas en cada capa = Prueba y Error
- Cantidad de Neuronas Salida: Cuántas clases quiero predecir

1. DEFINIR EL MODELO

• OTRAS CAPAS INTERMEDIAS

- BatchNormalization: Normaliza los datos
- DropOut: Elimina salidas aleatoriamente
- Pooling: Disminuye el tamaño de salidas (Avg, Máx, ...)
- Reshape, Flatten: Manejo del tamaño del Dataset

```
model = keras.Sequential([
    keras.layers.Dense(50, activation='relu'),
    keras.layers.Dropout(rate=0.4)
    keras.layers.Dense(30, activation='relu'),
    keras.layers.Dropout(rate=0.2)
    keras.layers.Dense(2, activation='softmax')
])
```

2. COMPILAR EL MODELO

- “Cómo Aprende el Modelo?”
- **Optimizador:** Cómo ajustar los valores para mejorar el modelo (Descendiente de Gradiente, DG. Estocástico, Adam) ...
- **Función de pérdida:** Qué tanto se esta equivocando: Loss, Error Residual (SSE), Error Promedio Absoluto (MAE)...
- **Métricas:** Otras medidas para optimizar (Accuracy, Recall, Precisión, ...)

3. ENTRENAR EL MODELO (.fit)

- Definir: Épocas, Batch Size y Callbacks
- Callbacks:
 - **ModelCheckpoint:** Guardar el mejor modelo en caso de que empeore
 - **EarlyStopping:** Detener el Entrenamiento cuando las métricas dejen de mejorar
 - **ReduceLROnPlateau:** Disminuir la taza de Aprendizaje LR cuando las métricas dejen de mejorar

LAB 4 - ¿QUÉ VAMOS A HACER?

- TensorFlow Playground!
 - Clasificación
 - Regresión
- **Mi primera Red Neuronal**

¿DÓNDE PUEDO LEER MÁS?

- <https://towardsdatascience.com/the-differences-between-artificial-and-biological-neural-networks-a8b46db828b7>
- <https://medium.com/swlh/simple-neural-network-from-scratch-130b175eb1e6>
- <https://algorithmia.com/blog/introduction-to-optimizers>
- https://www.youtube.com/playlist?list=PLZHQB0WTQDNU6R1_67000Dx_ZCJB-3pi
- <https://playground.tensorflow.org/>

APRENDIZAJE AUTOMÁTICO

M5. TIPOS DE REDES NEURONALES

Mónica Ramírez Bernal

2020-2

Centro de Investigaciones para el Desarrollo – CID
Programa de Educación permanente - PEC
Facultad de Ciencias Económicas
Sede Bogotá

educación continua 

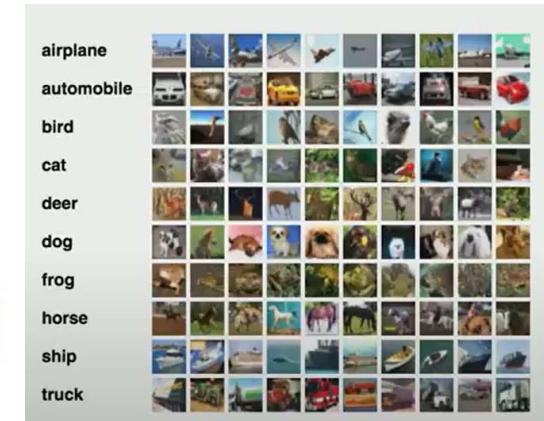
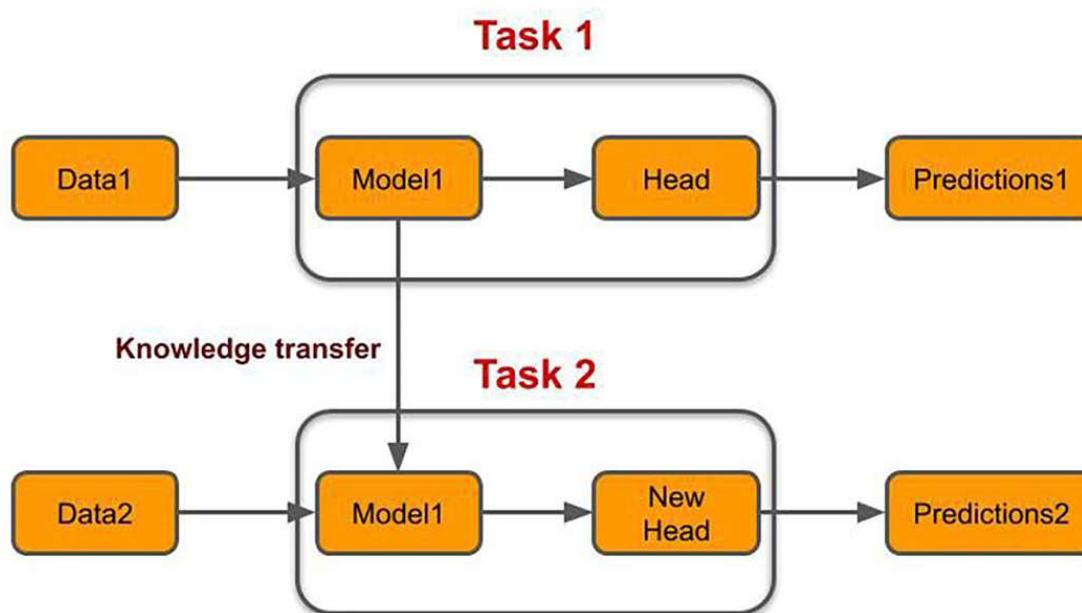


UNIVERSIDAD
NACIONAL
DE COLOMBIA

ANTES DE ESO, POR DÓNDE EMPEZAR?

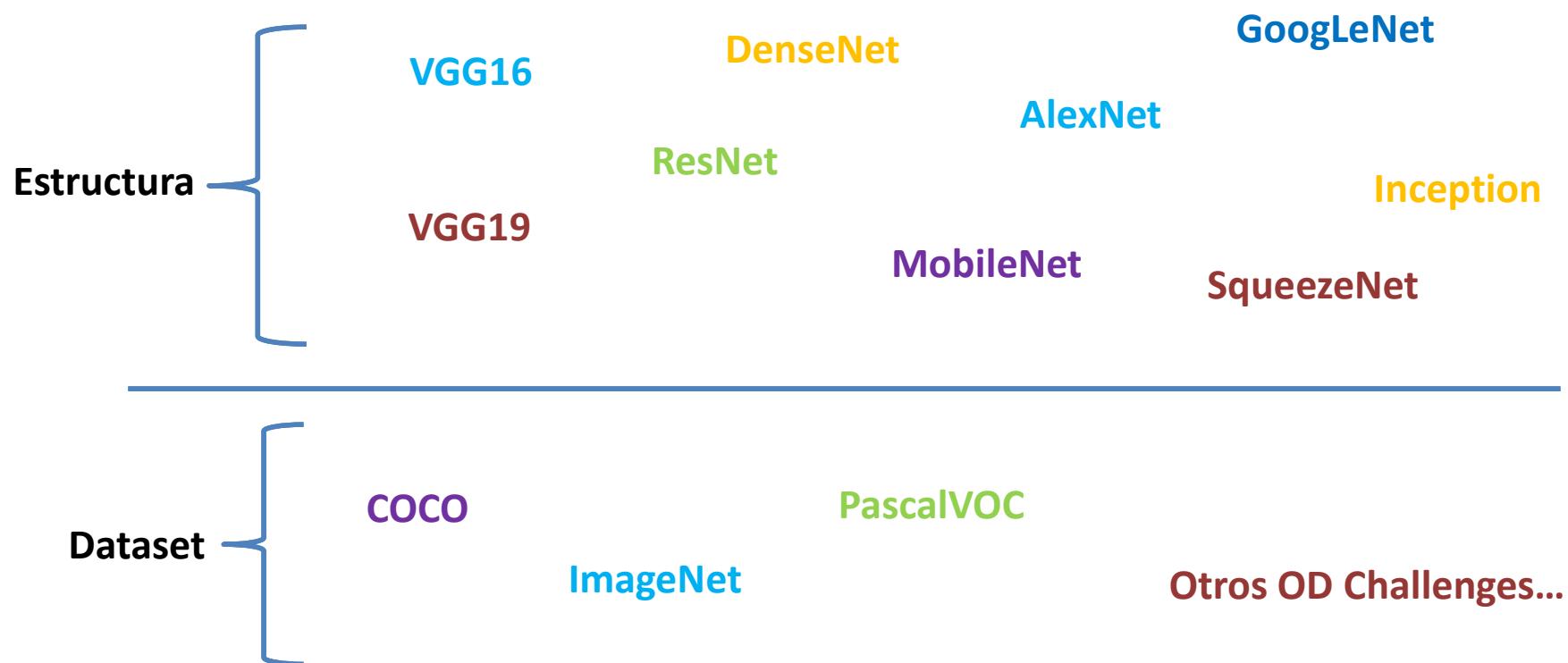


TRANSFER LEARNING



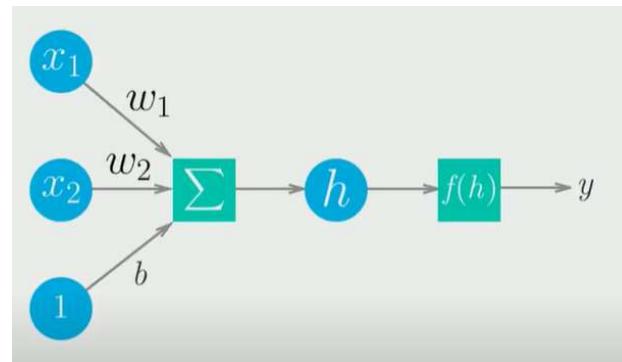
TRANSFER LEARNING

• QUÉ REDES SE PUEDEN USAR?

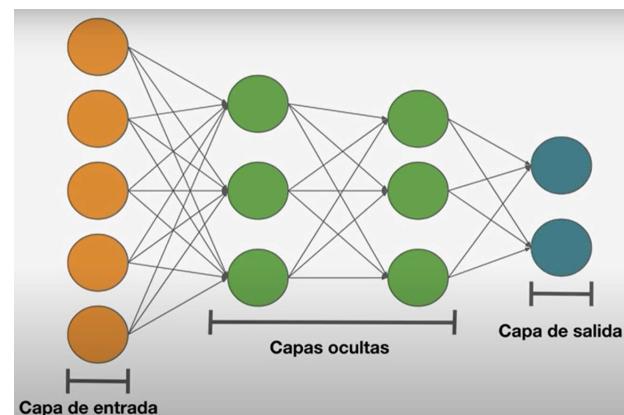


CLASIF. POR # CAPAS OCULTAS

- PERCEPTRÓN SIMPLE



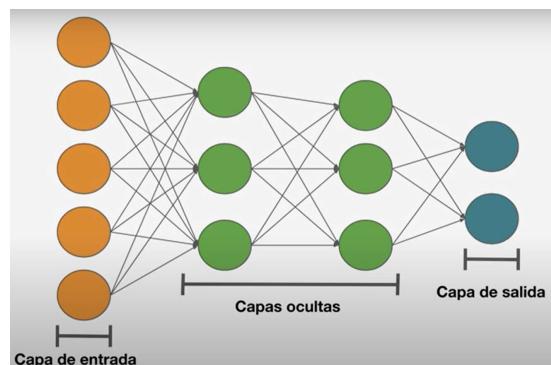
- PERCEPTRÓN MULTICAPA



CLASIF. POR TOPOLOGÍA

• 1. PROFUNDAS (DNN o FNN)

- Muy versátil (Excepto texto e imágenes...)
- “Completamente Conectada” = Cada neurona recibe todas las entradas de la capa anterior y retorna salidas para la capa siguiente

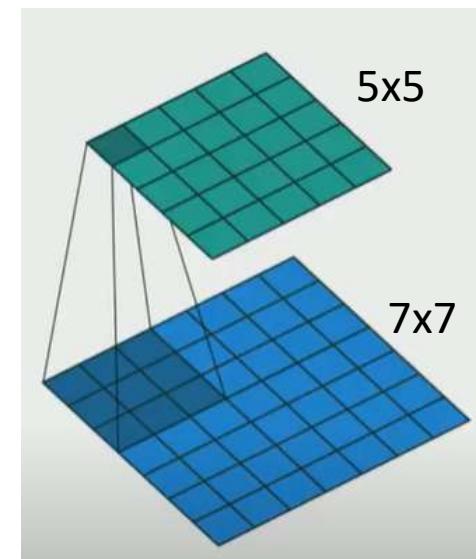
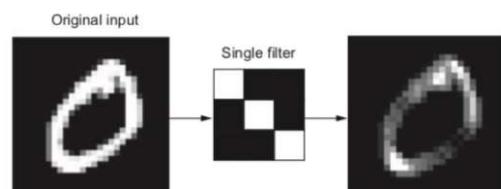


```
model = keras.Sequential([
    keras.layers.Dense(10, activation='relu'),
    keras.layers.Dense(2, activation='softmax')
])
```

CLASIF. POR TOPOLOGÍA

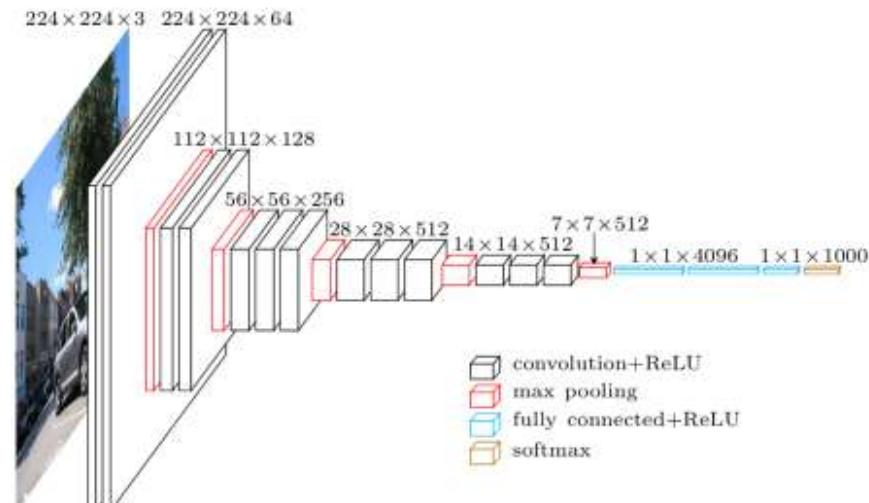
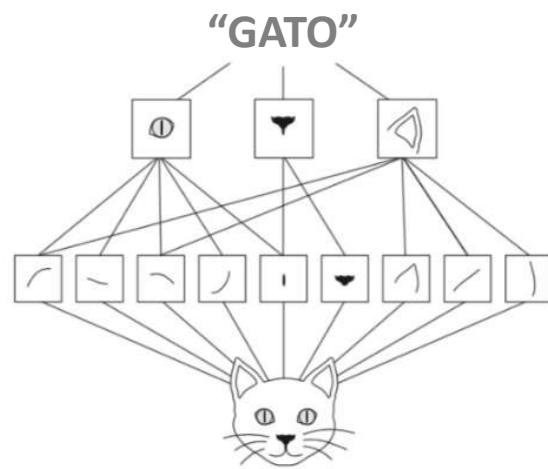
• 2. CONVOLUCIONAL (CNN)

- Ideal para datos espaciales, imágenes, videos, sonido...
- Convolución = Filtro (< parámetros!)



CLASIF. POR TOPOLOGÍA

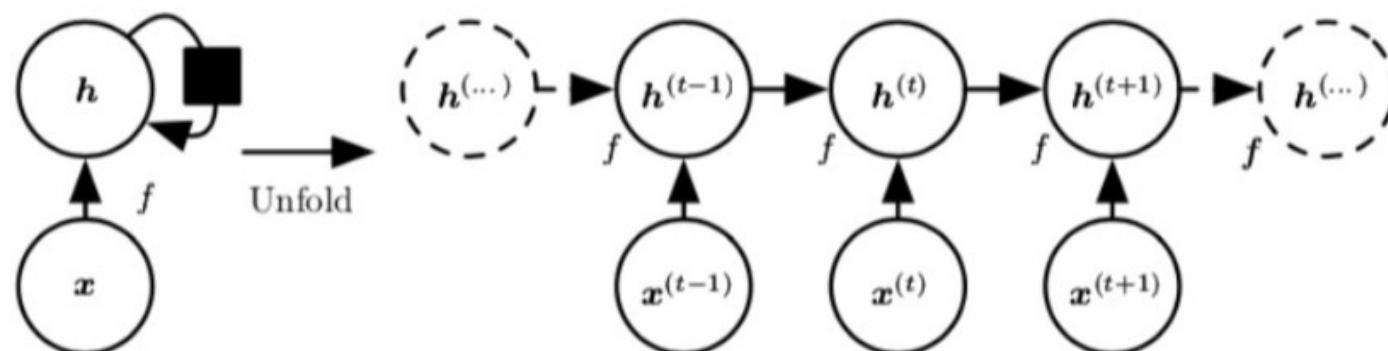
- **2. CONVOLUCIONAL (CNN)**
 - Cada capa generaliza datos de la capa anterior:



CLASIF. POR TOPOLOGÍA

• 3. RECURRENTE (RNN)

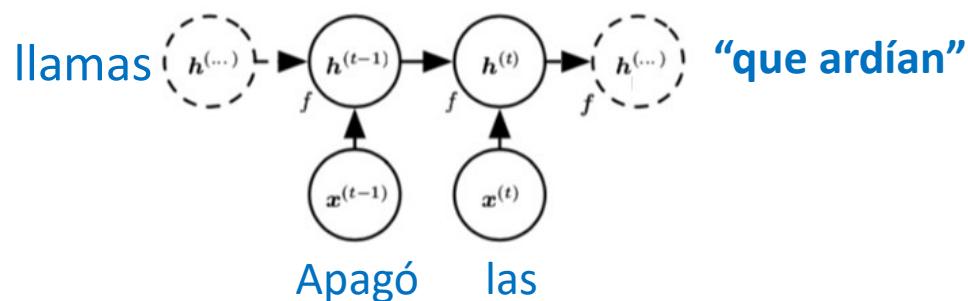
- Ideal para texto o series de tiempo
- “Repetir la capa anterior”



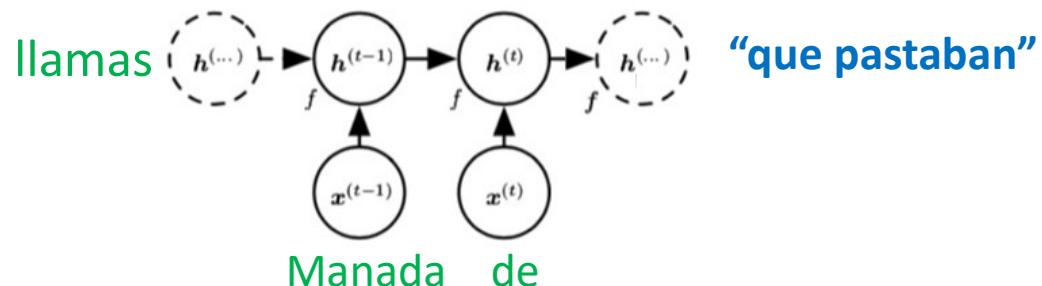
CLASIF. POR TOPOLOGÍA

• 3. RECURRENTE (RNN)

“Apagó las llamas”



“Manada de llamas”



LAB 5 - ¿QUÉ VAMOS A HACER?

- **Redes Convolucionales (Imágenes!)**
 - Construir la red!
 - Data Augmentation
- **Transfer Learning!**
 - Callbacks

QUÉ MÁS SE PUEDE HACER?

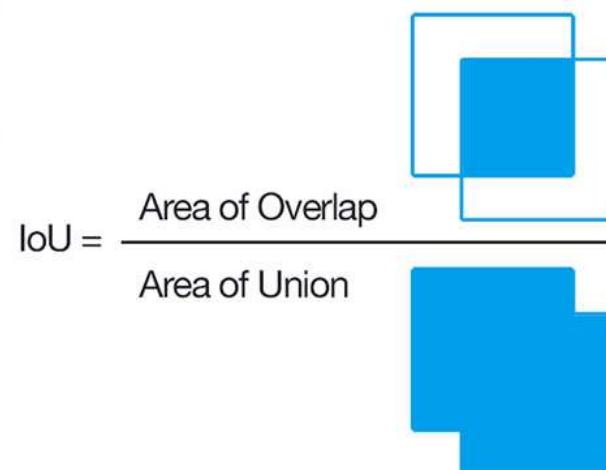
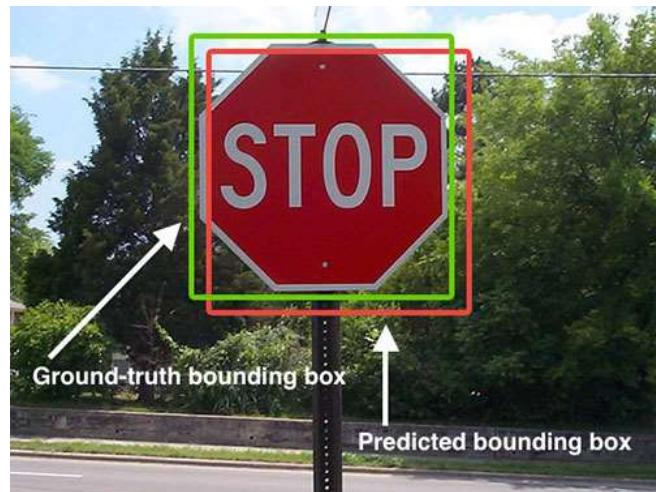
- DETECCIÓN DE OBJETOS (OD)



- Base Network
 - (Las que ya conocemos!)
- OD Framework
 - (R-CNN, SSD, YOLO)
- Sliding Window
- Image Pyramid
- Non-Max Suppression

QUÉ MÁS SE PUEDE HACER?

- DETECCIÓN DE OBJETOS (VALIDACIÓN)



www.pyimagesearch.com/2018/05/14/a-gentle-guide-to-deep-learning-object-detection/

QUÉ MÁS SE PUEDE HACER?

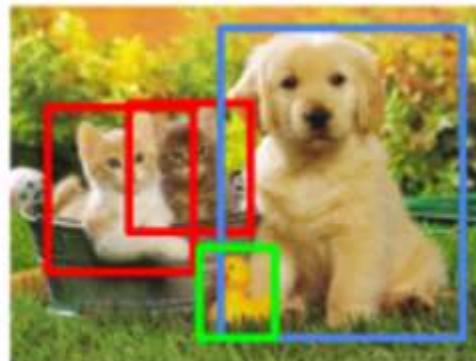
- **SEGMENTACIÓN DE OBJETOS**

Clasificación



CAT

Detección



CAT, DOG, DUCK

Segmentación

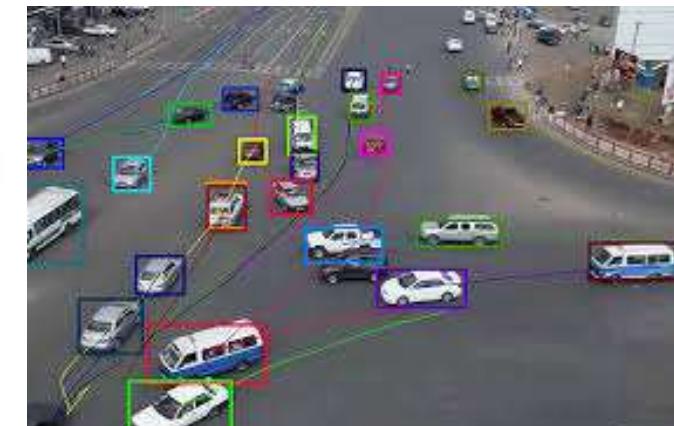


CAT, DOG, DUCK

QUÉ MÁS SE PUEDE HACER?

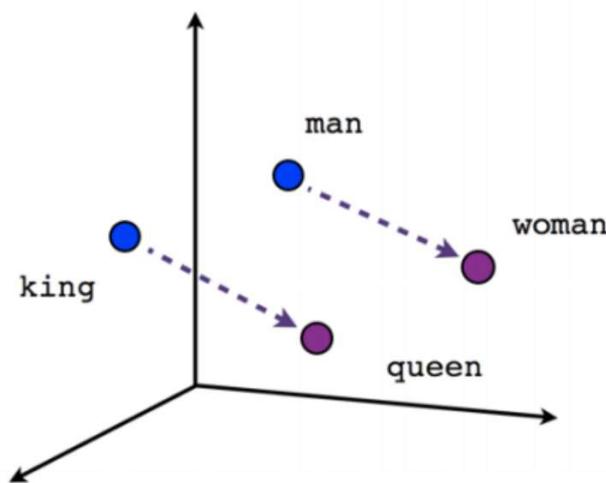
- **SEGUIMIENTO DE OBJETOS (OT)**

- Recibe (x,y) de un objeto a seguir y “predice” su ubicación
(Algoritmos que hacen eso por similitud o correlación)
- OD es mucho más \$\$\$ en tiempo y recursos
- OD busca un objeto nuevo y se lo entrega al OT
- OT solo se ejecuta cada N frames
- OT sigue los objetos que detecta el OD
- OT se actualiza cada N frames

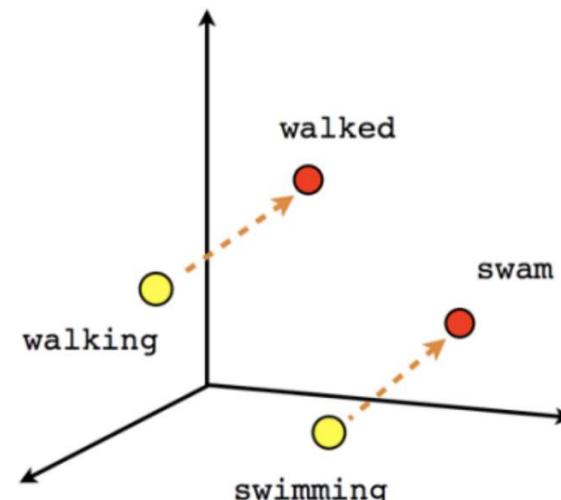


QUÉ MÁS SE PUEDE HACER?

• VECTORES DE PALABRAS



Male-Female

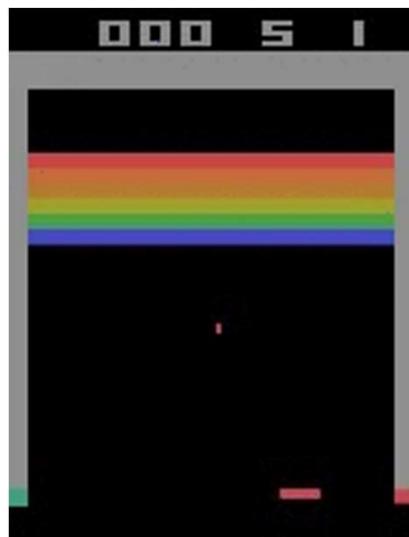


Verb tense

https://www.marekrei.com/pub/Constructing_and_Evaluating_Word_EMBEDDINGS.pdf

QUÉ MÁS SE PUEDE HACER?

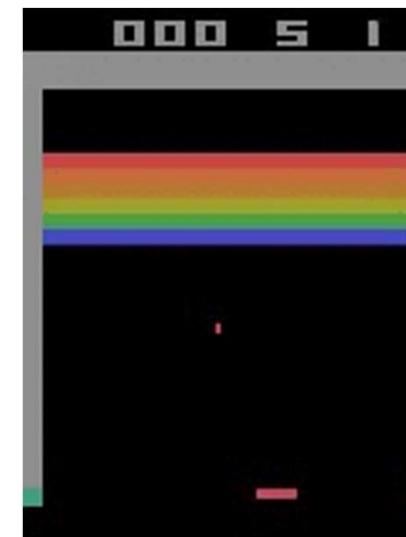
- APRENDIZAJE POR REFUERZO



Inicial



Entrenamiento



Final

<https://deepsense.ai/what-is-reinforcement-learning-the-complete-guide/>

DOS REDES NEURONALES!

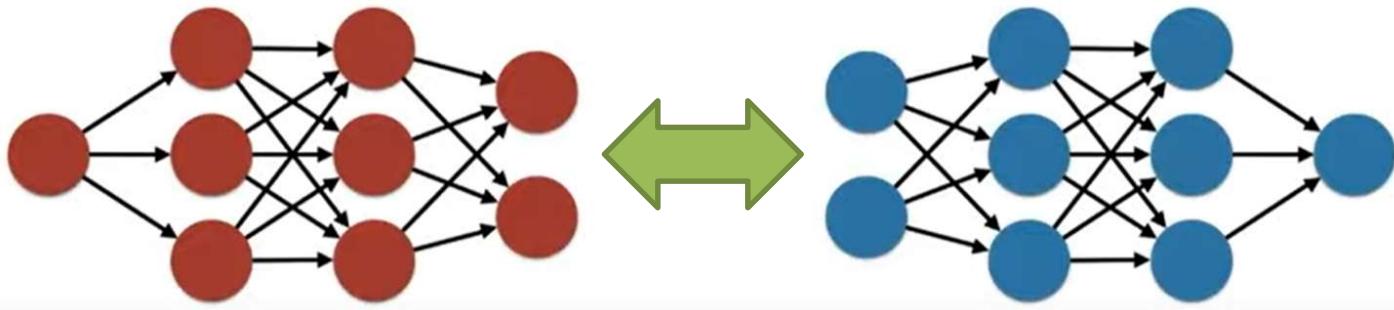
- REDES ADVERSARIALES GENERATIVAS (GANs)



<https://thispersondoesnotexist.com/>

DOS REDES NEURONALES!

- **REDES ADVERSARIALES GENERATIVAS (GANs)**



GENERADOR

DISCRIMINADOR

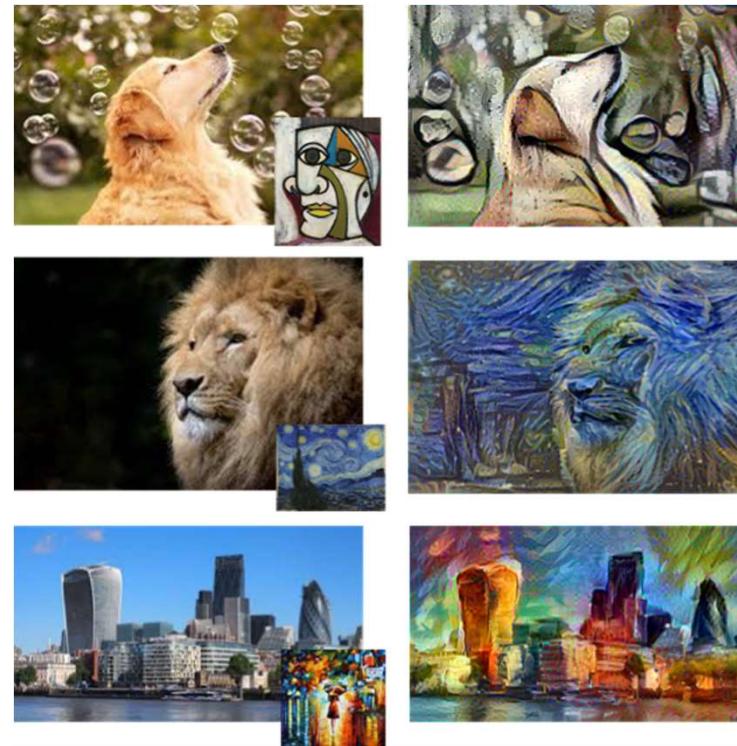


<https://github.com/luisguiserrano/gans>

GENERAR IMÁGENES A PARTIR DE OTRAS

- **STYLE TRANSFER**

$$Loss = \alpha Loss_content + \beta Loss_style$$



<https://www.fritz.ai/style-transfer/>

¿DÓNDE PUEDO LEER MÁS?

- <https://www.youtube.com/watch?v=V5BYRPJThjE>
- <https://www.youtube.com/watch?v=tSjciYOKNdw>
- <https://medium.com/towards-artificial-intelligence/main-types-of-neural-networks-and-its-applications-tutorial-734480d7ec8e>
- <https://machinelearningmastery.com/how-to-configure-image-data-augmentation-when-training-deep-learning-neural-networks/>
- <https://towardsdatascience.com/a-comprehensive-hands-on-guide-to-transfer-learning-with-real-world-applications-in-deep-learning-212bf3b2f27a>

APRENDIZAJE AUTOMÁTICO

M6. IMPLEMENTACIÓN

Mónica Ramírez Bernal

2020-2

OTRAS LIBRERÍAS:



Caffe



neuralnet
nnet
darch
deepnet



TensorFlow

PyTorch

theano

¿Y AHORA QUÉ?



“El 80% de los Modelos de IA se quedan en el laptop de los Científicos de Datos”

“GUARDAR” EL MODELO

- Estructura + Pesos
- Modelos sencillos (ML):
 - Python: .pkl .joblib
 - R : .rda .rds
- Redes Neuronales: Cada librería lo hace
- Se pueden volver a cargar
- Reentrenar desde checkpoint

¿EN DÓNDE?

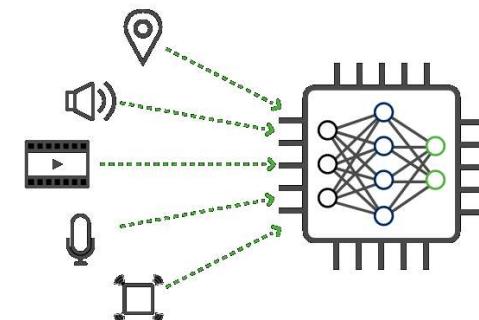
- Complejidad del Modelo (GPU?)
- Capacidad a soportar
- Velocidad de Respuesta



OnSite



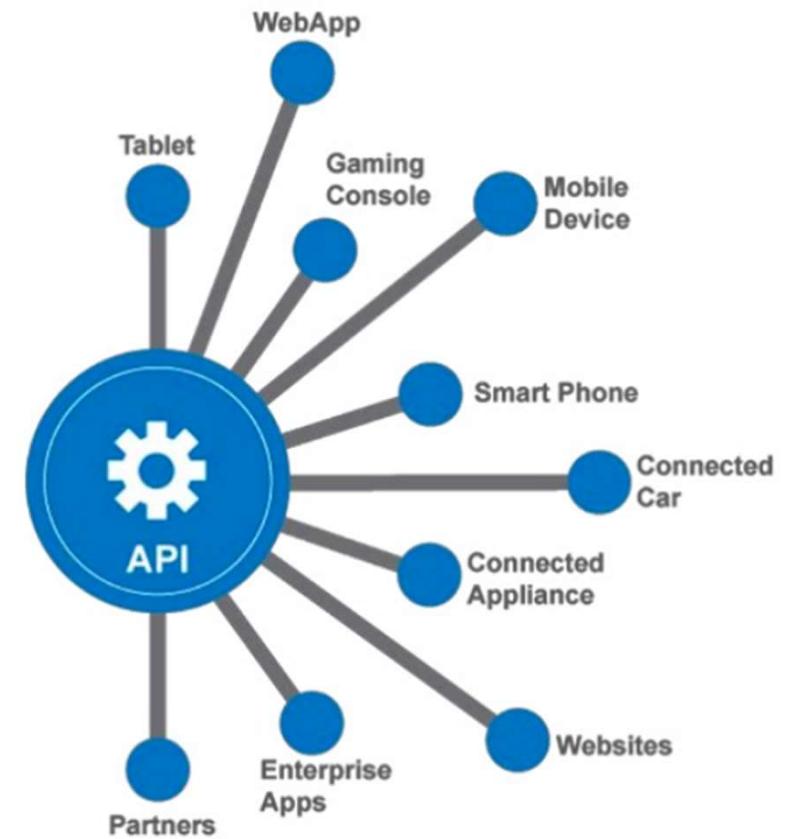
Cloud



Edge

¿CÓMO?

- API
 - Encapsular un servicio para que pueda ser usado por otros
 - Estandariza envío de información
 - En nuestro caso el servicio a encapsular es el modelo!



¿CÓMO?

- API (Frameworks)

Densidad:

Alcohol:

pH:

Azucar:

Predecir Score de Vino



Flask



heroku



Glitch



¿CÓMO?

- **Cloud**
 - Aprovisionamiento de Servicios por Demanda
 - Pay as you Go (Pago lo que Consumo)

Servicios

- Almacenamiento
- Instancias Cómputo
- Bases de Datos
- Notebooks
- Entrenamiento Modelos
- Despliegue de Modelos

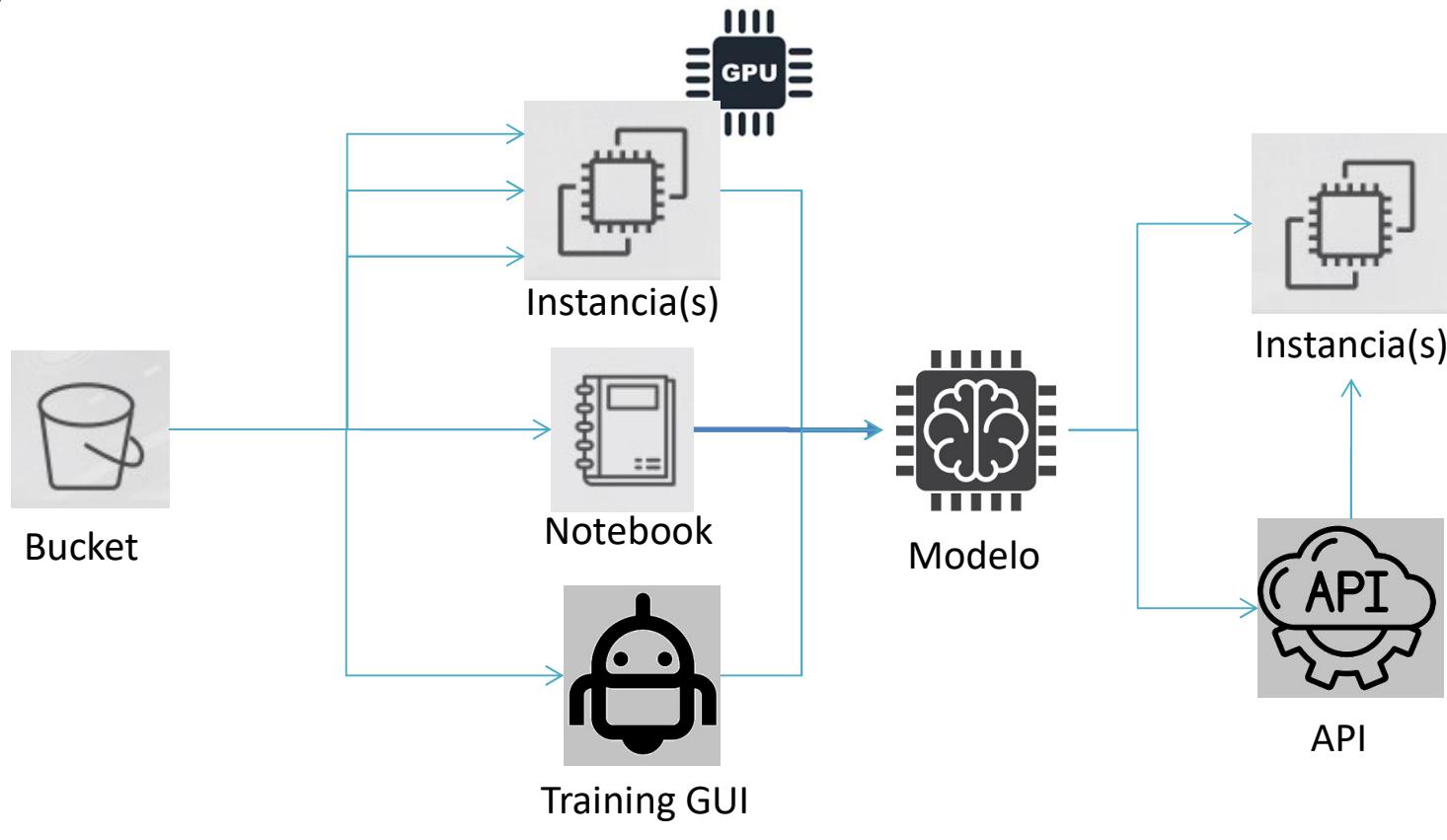
Proveedores



Google Cloud



¿CLOUD?

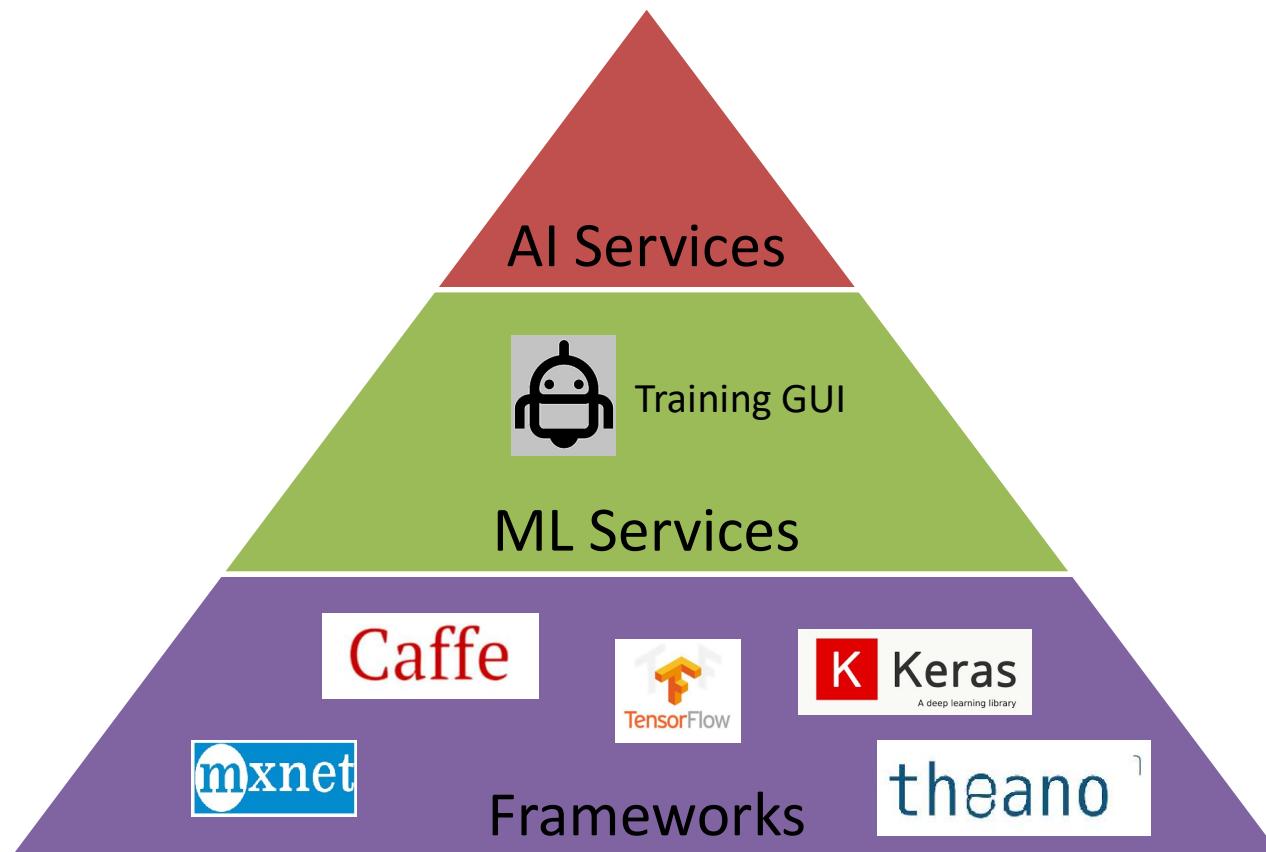


Almacenamiento

Entrenamiento

Despliegue

¿CLOUD?



LAB 6 - ¿QUÉ VAMOS A HACER?

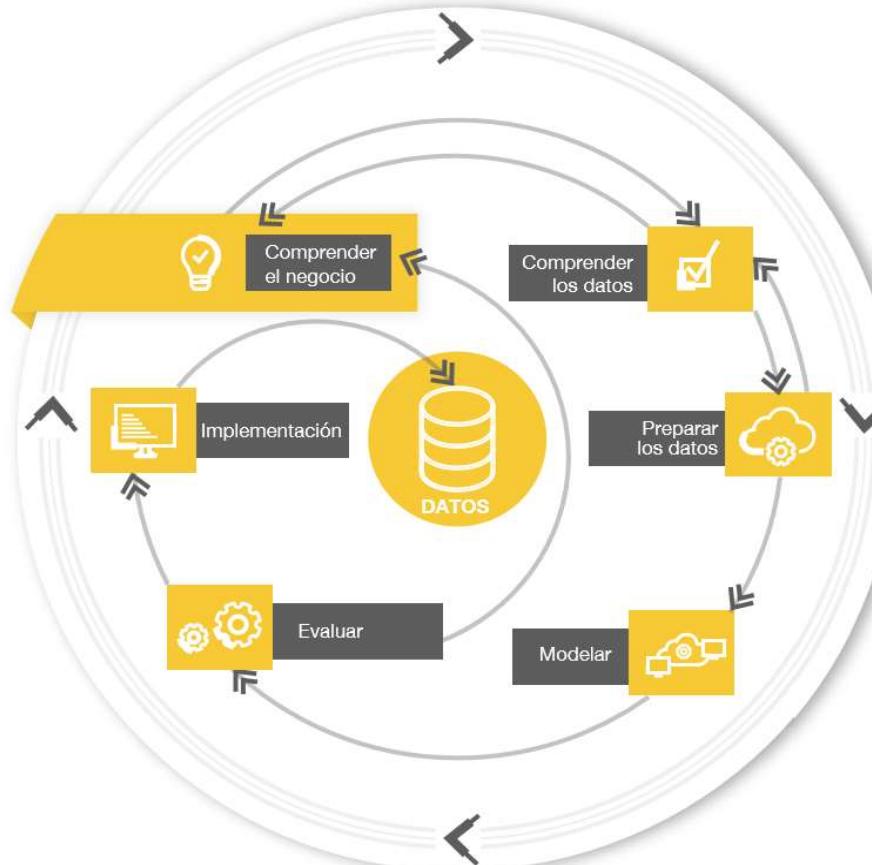
- APIs Preentrenadas

- Detección de Objetos
- Análisis Facial
- Comparación de Rostros
- Texto en imágenes
- Reconocimiento de Celebridades



¿PROYECTO REAL?

- Desarrollo:
CRISP-DM



PROYECTO REAL?

- Ciclo de Vida Modelo:



¿DÓNDE PUEDO LEER MÁS?

- <https://www.youtube.com/watch?v=sRQrxoTDsGw>
- <https://anderfernandez.com/blog/como-poner-en-produccion-un-modelo-de-machine-learning-de-r/>
- <https://www.datacamp.com/community/tutorials/machine-learning-models-api-python>
- <https://aws.amazon.com/machine-learning/>
- <https://cloud.google.com/ai-platform>
- <https://www.coursera.org/lecture/big-data-machine-learning/crisp-dm-icJH6>

PROYECTO!

- Mismos grupos
- Qué pregunta de negocio vamos a solucionar?
 - ML o DL
- Presentación final de 10 minutos
 - Desafíos
 - Mejoras al modelo
- Manejo del tiempo
- Participación
- Nota por grupo = Elimina el peor Quiz!

PROYECTO!

ACTIVIDAD	FECHA
Asignación de Datasets	JUEVES (19 de Noviembre)
Clase Práctica – Trabajo en el Proyecto A. Automático	SÁBADO (21 de Noviembre)
Primera Clase Curso: Semántica de Datos y Visualización (Julián Cruz)	MARTES (24 de Noviembre)
Presentaciones Proyecto A. Automático	JUEVES (26 de Noviembre)