



UNIVERSIDAD  
**NACIONAL**  
DE COLOMBIA

# APRENDIZAJE DE MÁQUINAS

**JOHN W. BRANCH**

Profesor Titular

Departamento de Ciencias de la Computación y de la Decisión

Director del Grupo de I+D en Inteligencia Artificial – GIDIA

[jwbranch@unal.edu.co](mailto:jwbranch@unal.edu.co)

<https://github.com/jwbranch/AprendizajeDeMaquinas>

<https://www.coursera.org/programs/unal-iuukt>

# METODOLOGÍA ENSEÑANZA – APRENDIZAJE

El aprendizaje sincrónico involucra estudios online a través de una plataforma. Este tipo de aprendizaje sólo ocurre en línea. Al estar en línea, el estudiante se mantiene en contacto con el docente y con sus compañeros. Se llama aprendizaje sincrónico porque la plataforma permite que los estudiantes pregunten al docente o compañeros de manera instantánea a través de herramientas como el chat o el video chat.

## Sesiones Remotas vía Google.Meet Sincrónicas y Asincrónicas

El aprendizaje asincrónico puede ser llevado a cabo online u offline. El aprendizaje asincrónico implica un trabajo de curso proporcionado a través de la plataforma o el correo electrónico para que el estudiante desarrolle, de acuerdo a las orientaciones del docente, de forma independiente. Un beneficio que tiene el aprendizaje asincrónico es que el estudiante puede ir a su propio ritmo.

# Descripción del Curso

El curso introduce los conceptos fundamentales y los métodos más utilizados en el campo del aprendizaje de máquinas enfocados desde las perspectivas de la naturaleza del problema que se requiere resolver, esto es, aprendizaje supervisado orientado a los problemas de clasificación y regresión para aplicaciones de predicción o pronóstico. Aprendizaje no supervisado orientado a tareas de agrupar o etiquetar un conjunto de datos, También se incluyen la aproximación general de técnicas modernas de aprendizaje tales como el aprendizaje por refuerzo y aprendizaje profundo.

# Contenido

1. Introducción.
2. Los datos en Aprendizaje de Máquinas.
3. **Aprendizaje Supervisado.**
4. Aprendizaje NO Supervisado.
5. Aprendizaje por Refuerzo.
6. Aprendizaje con Clases Desbalanceadas y Combinación de Modelos.
7. Aplicaciones y Casos de Éxito.

# Bibliografía Recomendada

**Oswaldo Simeone (2018)**, “A Brief Introduction to Machine Learning for Engineers”, Foundations and TrendsR in Signal Processing: Vol. 12, No. 3-4, pp 200–431. DOI: 10.1561/20000000102.

**Goodfellow, I., Bengio, Y. y Courville, A. (2016)** Deep Learning, MIT Press.

**Murphy, K. (2012)**. Machine Learning: A Probabilistic Perspective, MIT Press .

**Hastie, T., Tibshirani, R. y Friedman, J. (2011)**. The Elements of Statistical Learning. Springer.  
(Available for download on the authors' web-page.)

**Szepesvári, C. (2010)**. Algorithms for Reinforcement Learning. Morgan and Claypool.

**Haykin, S. (2008)**. Neural Networks and Learning Machines. Pearson.

**Sutton, R. y Barto, A. (1998)**. Reinforcement Learning: An Introduction. MIT Press.

**Mitchell, T. M. (1997)**. Machine Learning. 1st. McGraw-Hill Higher Education. (Chapter 1)

# EVALUACIÓN



## Certificado Coursera

*Sesenta años de inteligencia artificial – UNAM (Obligatorio)*

*IA para todos – Andrew Ng (Obligatorio)*

*Structuring Machine Learning Projects – Andrew Ng (Obligatorio)*

Machine Learning - University of Washington → **Curso #1: Machine Learning Foundations: A Case Study Approach** (Obligatorio)

## Informe de Lectura (Individual)

Machine Learning Algorithms: A Review

Machine Learning aspects and its Applications Towards Different Research Areas

## Trabajo Final (Debe ser en Grupo 3 ó 5 personas)

*Obtener el conjunto de datos (texto o audio o video o imagen) de los siguientes repositorios o cualquier otro disponible:*

<http://www.ics.uci.edu/~mllearn/databases/>

<https://www.kaggle.com/datasets>

*Origen, atributos, clases*

*"Scatter plot" de los datos*

*Visualización del conjunto en 2D (PCA o MDS)*

*Seleccionar un método de Entrenamiento y Evaluar el Desempeño.*

**20%**

**(Máx. 31 de Dic/2020)**

**20%**

**(Máx. 21 de Nov/2020)**

**60%**

**(Máx. 21 de Dic/2020)**

# APRENDIZAJE DE MÁQUINAS

## “Aprendizaje Supervisado”

**JOHN W. BRANCH**

Profesor Titular

Departamento de Ciencias de la Computación y de la Decisión

Director del Grupo de I+D en Inteligencia Artificial – GIDIA

[jwbranch@unal.edu.co](mailto:jwbranch@unal.edu.co)

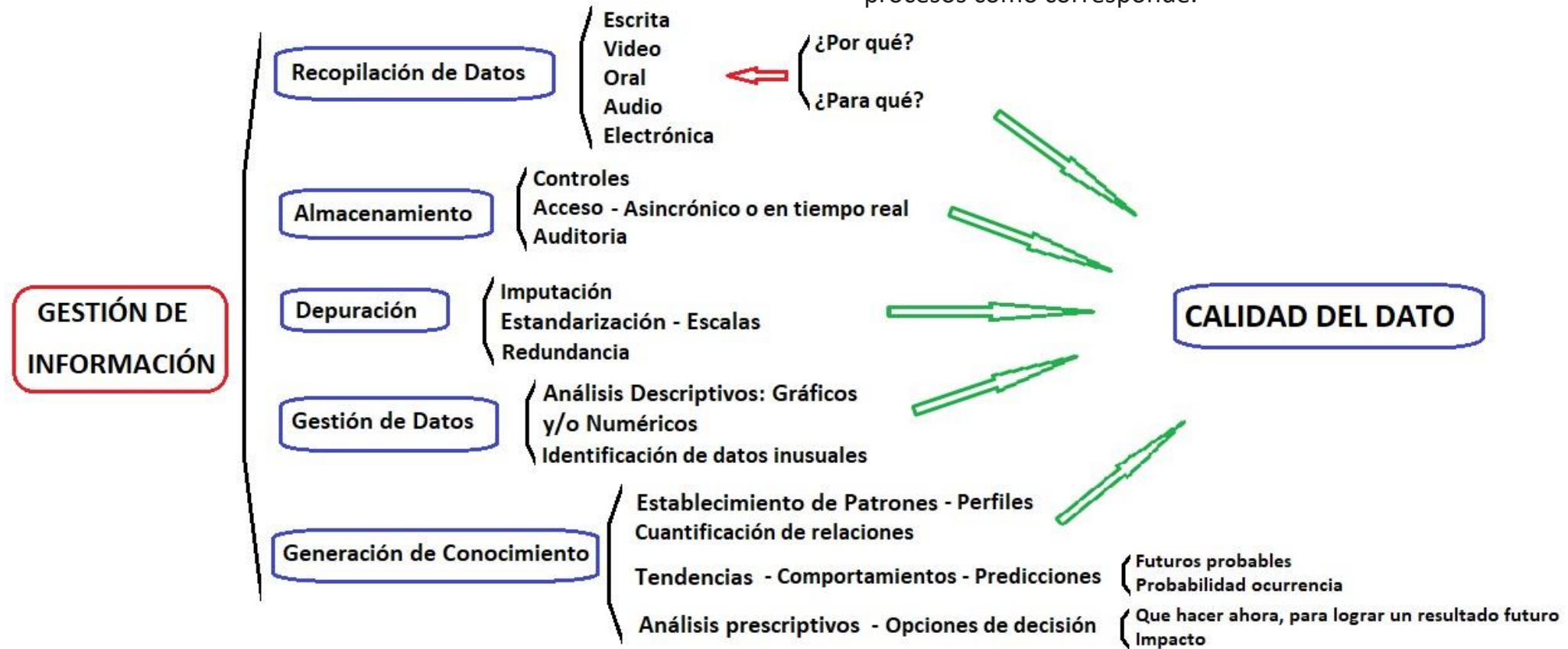
<https://github.com/jwbranch/AprendizajeDeMaquinas>

<https://www.coursera.org/programs/unal-iuukt>



# Cultura y Calidad del Dato

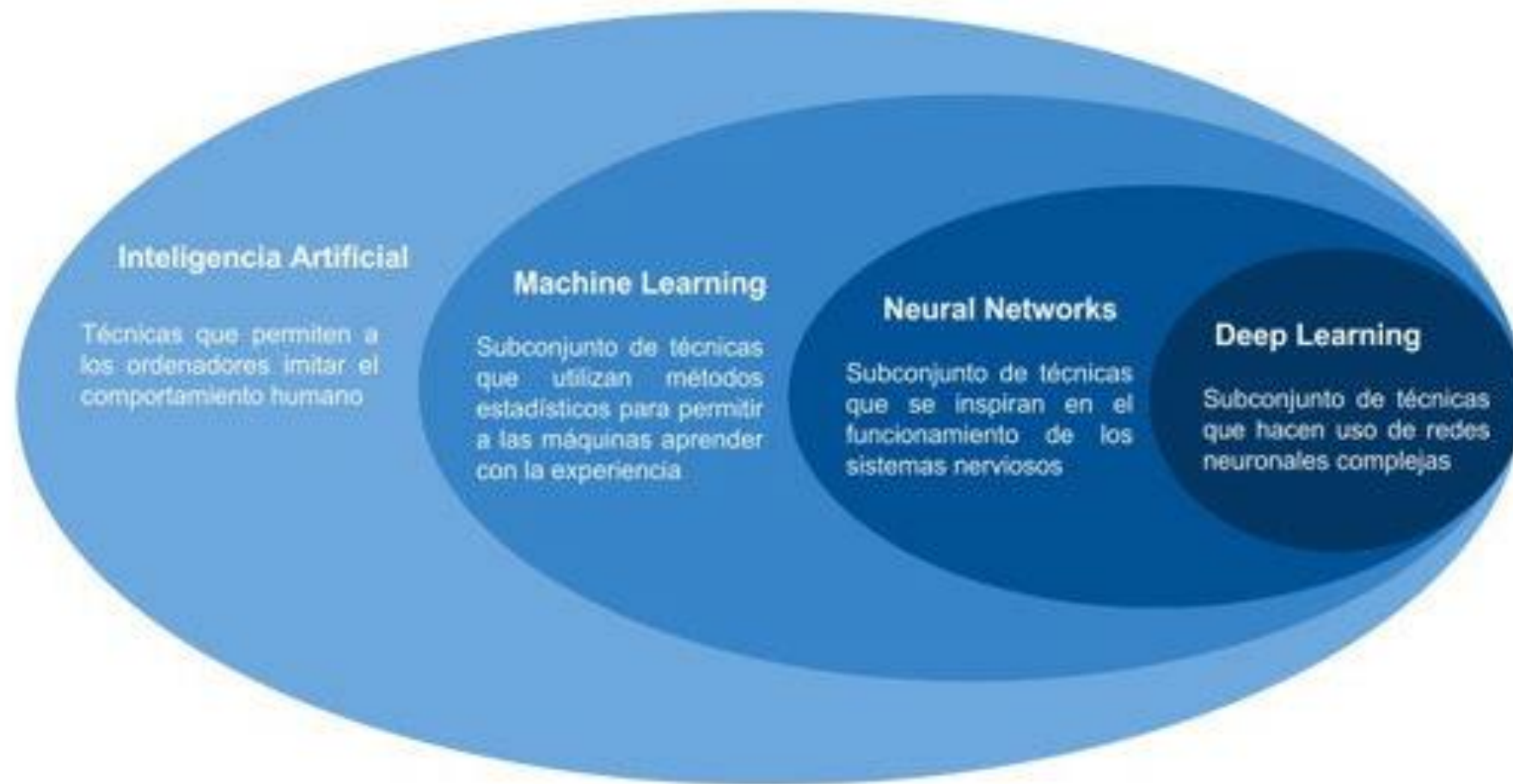
Un estudio realizado por Harvard Business Review, menos del 50% de las decisiones estructuradas de las empresas se basan en datos. Este resultado refleja la importancia de implementar una nueva cultura de datos y, sobre todo, llevar los procesos como corresponde.



*"...las organizaciones basadas en datos tienen 23 veces más probabilidades de adquirir clientes, seis veces más probabilidades de retener a esos clientes y 19 veces más probabilidades de ser rentables."*

McKinsey Global Institute

# Aprendizaje vs Inteligencia



**Las máquinas pueden:**

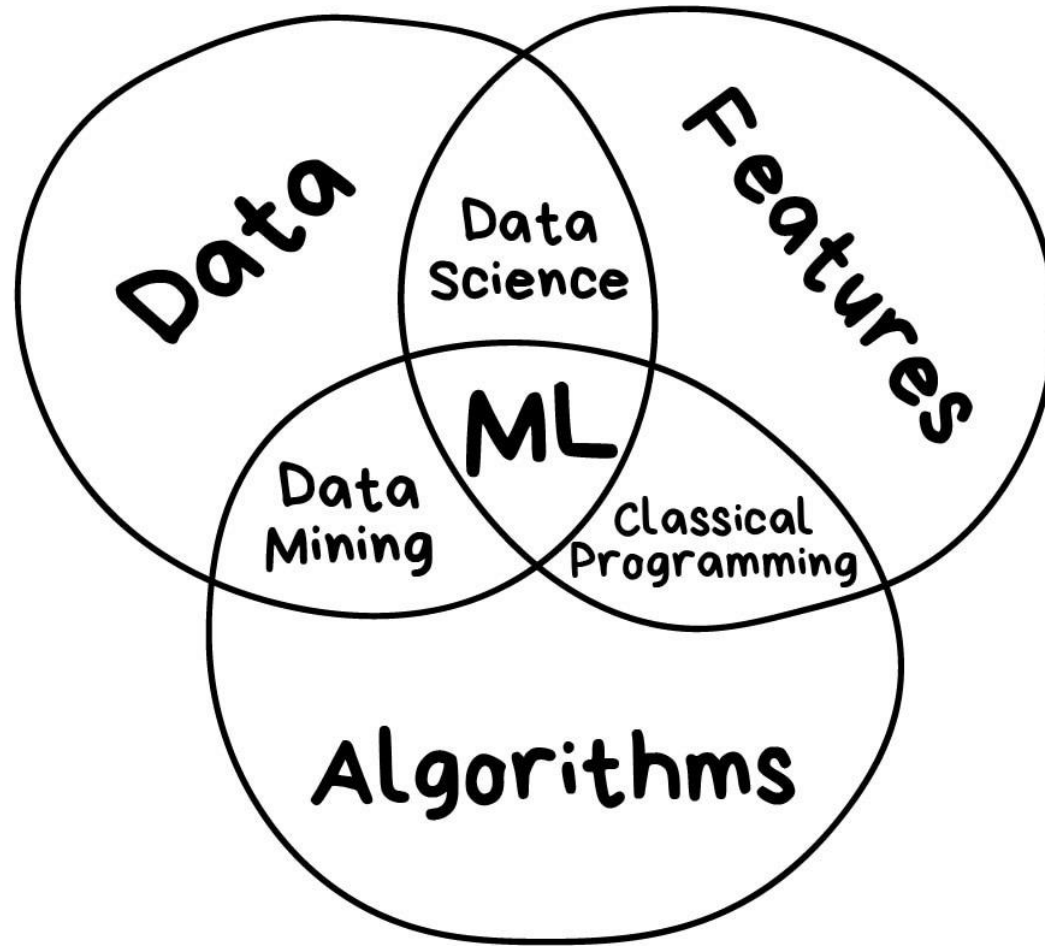
**Hacer Pronósticos**

**Memorizar**

**Reproducir**

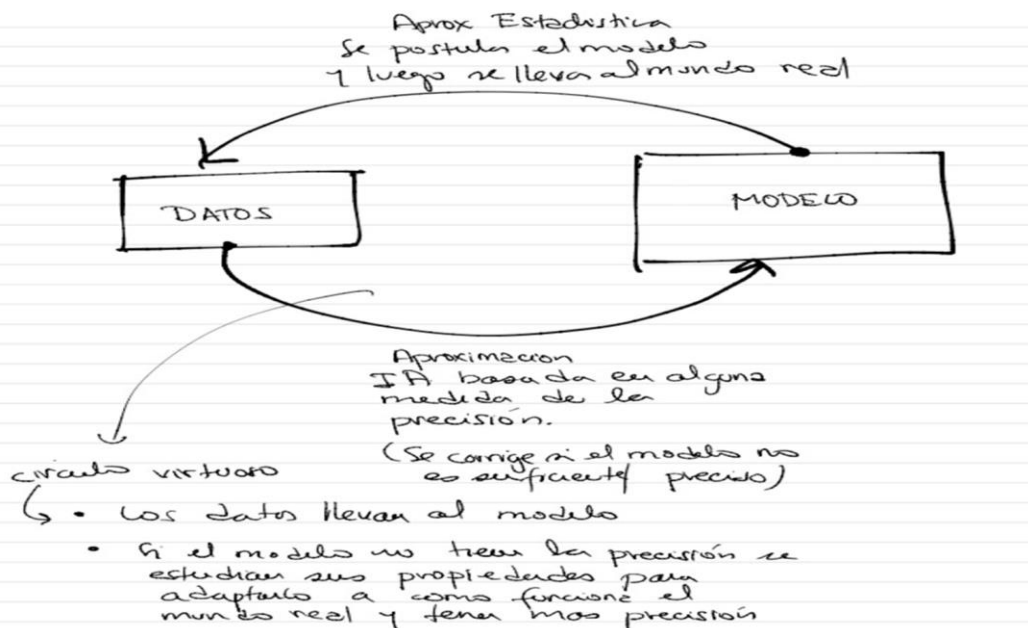
**Elegir el mejor artículo**

## Lo tres componentes del aprendizaje de máquinas



# Tipos de Aprendizaje de Máquinas

No existe una única forma de resolver un problema en el mundo del aprendizaje de máquinas. Siempre hay varios algoritmos que se ajustan, y la habilidad del científico de datos está en elegir cuál se adapta mejor.



## TIPOS PRINCIPALES DE APRENDIZAJE AUTOMÁTICO



# APRENDIZAJE AUTOMÁTICO CLÁSICO





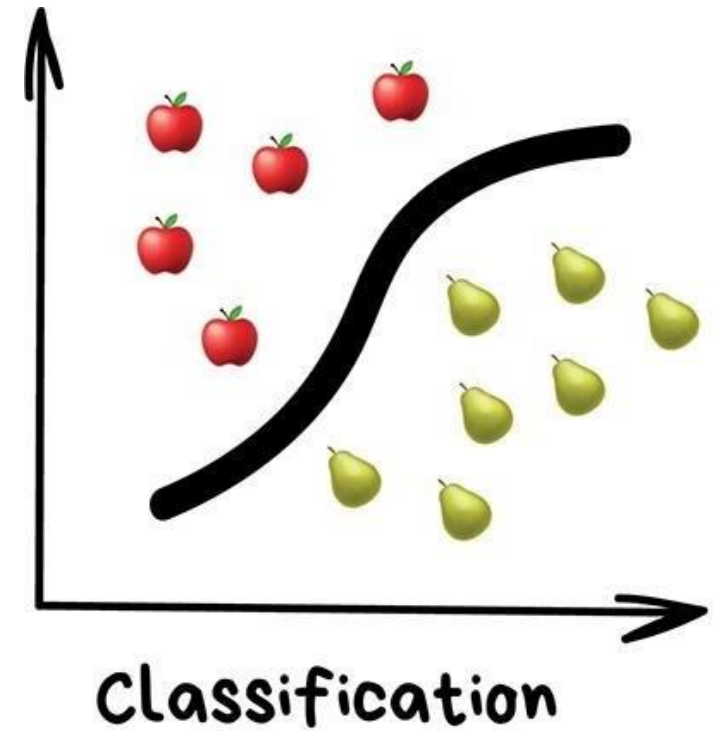
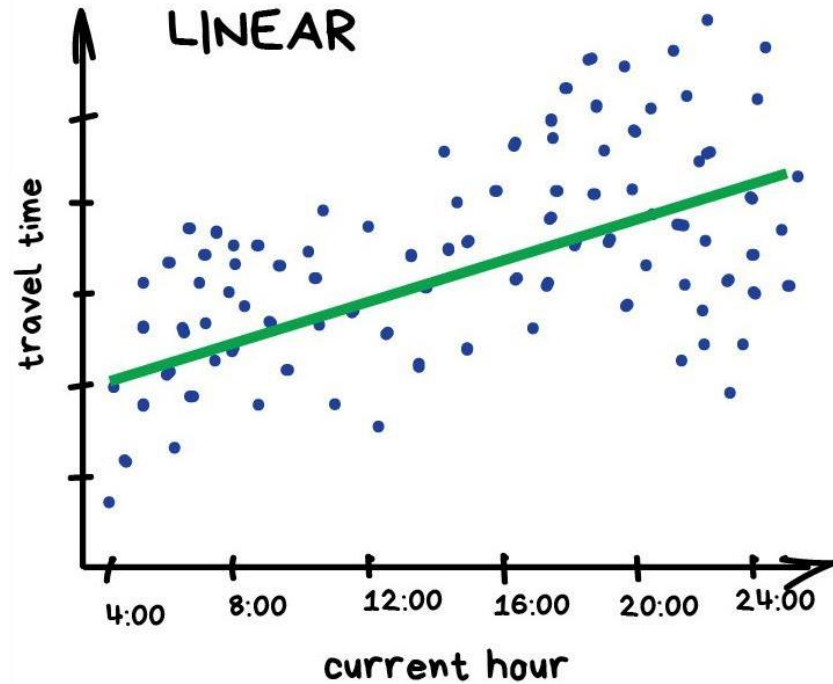


# Aprendizaje Supervisado

**Aprendizaje Supervisado:** requiere de un conjunto de datos conocidos a partir del cual se crea un **modelo** para predecir el valor de una variable de salida. El aprendizaje supervisado se puede usar en dos tareas:

- **Regresión:** en este caso los algoritmos de aprendizaje buscan predecir el valor de una **variable continua** a partir de los datos de entrada. Un ejemplo de una tarea de regresión es el de estimar la longitud de un salmón en función de su edad y su peso.
- **Clasificación:** en este caso la variable de salida es una etiqueta que determina la clase a la que pertenecen los datos de entrada, es decir, la variable de salida es una **variable discreta**.

# Tipos de Aprendizaje





# Contenido

## 1. Regresión

- a. Regresión Lineal
- b. Regresión Polinomial
- c. Regresión Lasso, Ridge y Elastic-net
- d. Red Neuronal para Regresión

## 2. Clasificación

- a. Regresión Logística
- b. K-nn
- c. Máquina de Vectores de Soporte
- d. Algoritmo XGBoost para Clasificación
- e. Red Neuronal para Clasificación

# Regresión Lineal

## CORRELACIÓN ENTRE DOS VARIABLES:

Se considera que dos variables cuantitativas ( $x$  e  $y$ ) están **correlacionadas** cuando una de ellas ( $y$ ) varía sistemáticamente con respecto a los valores de la otra ( $x$ ).

**Por ejemplo:**

- ✓ ¿Hay una correlación entre la Temperatura y el número de Helados Vendidos en una Heladería?
- ✓ ¿Puede identificar otras correlaciones?

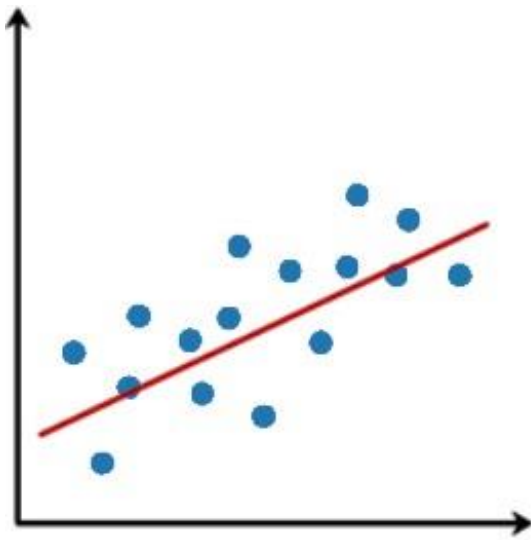
Claro está, si sabemos que la variable  $x$  está correlacionada con  $y$ , quiere decir que podemos **predecir** la variable  $y$  a partir de  $x$ .

**Estamos en el terreno de la PREDICCIÓN!**

# Regresión Lineal

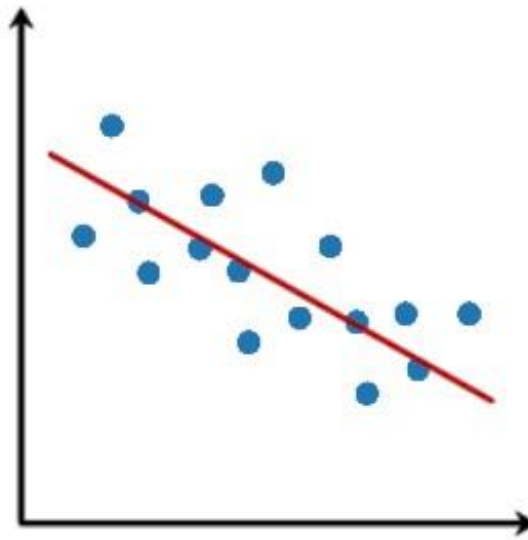
## TIPOS DE CORRELACIÓN:

Hay tres tipos básicos de correlación: positiva, negativa y nula (sin correlación).



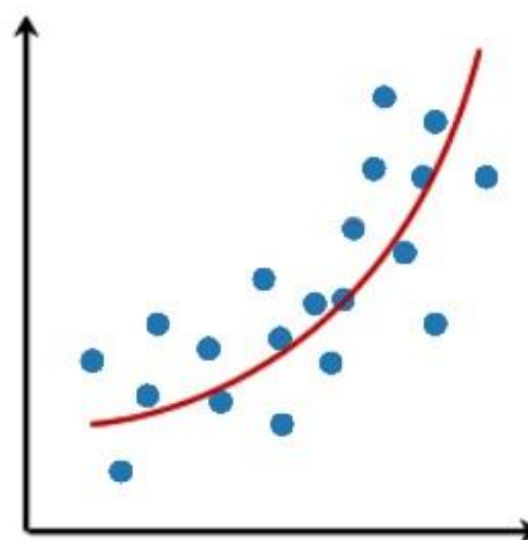
**Correlación Lineal Positiva**

Ocurre cuando una variable aumenta y la otra también

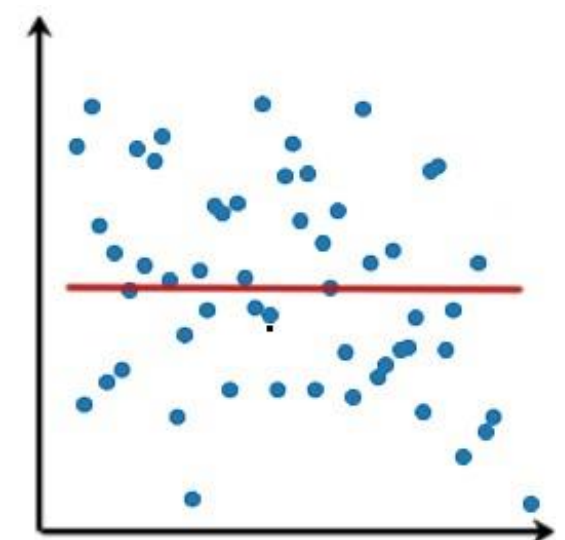


**Correlación Lineal Negativa**

Ocurre cuando una variable aumenta y la otra disminuye



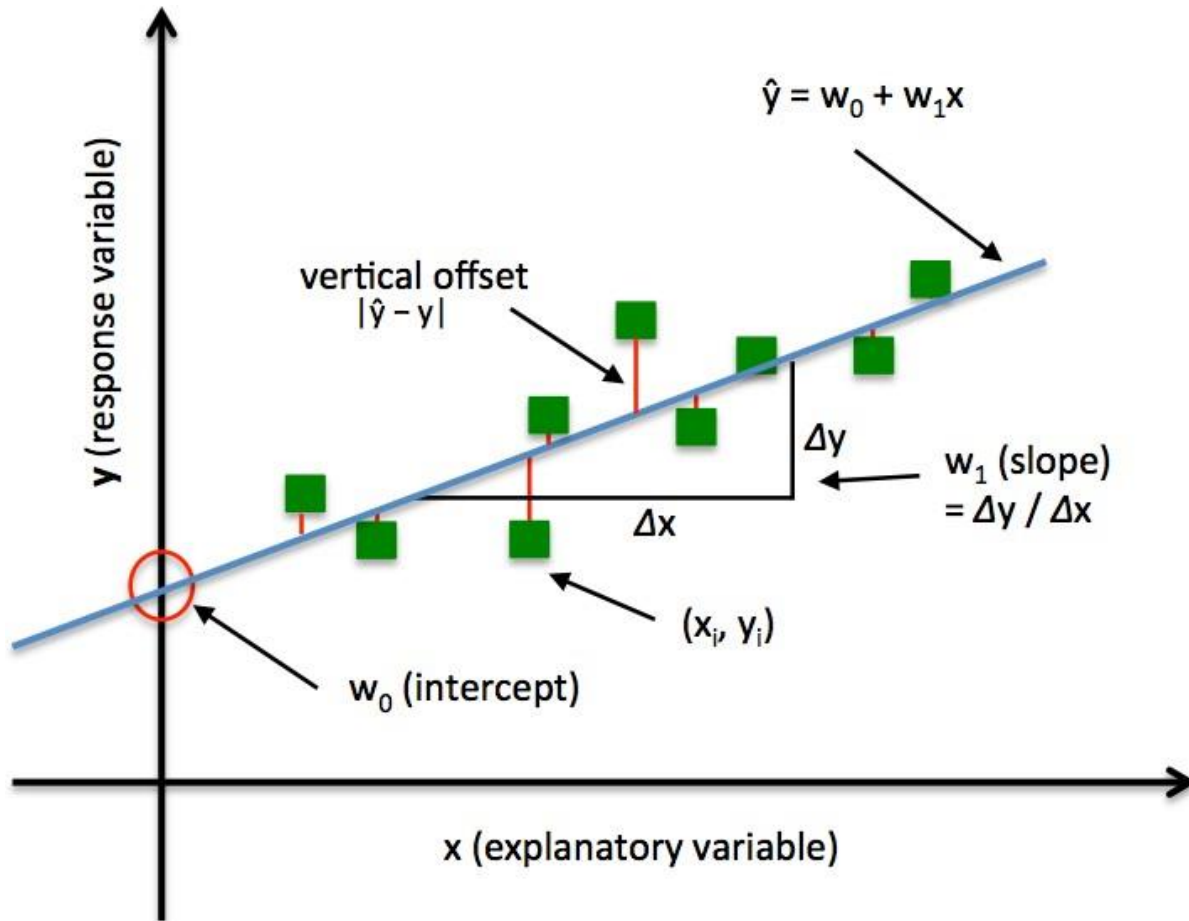
**Correlación No Lineal**



**Sin Correlación**

No hay una relación aparente entre las variables

# Regresión Lineal



## ■ Generalidades:

- Los métodos de regresión buscan modelar la relación entre 2 variables.
- El modelo se ajusta usando una **medida de error** sobre las predicciones que éste hace.
- En la **Regresión Lineal** el modelo a ajustar es una línea recta:

$$\hat{y} = w_0 + w_1x$$

Puede haber múltiples líneas rectas dependiendo de los valores de intercepción y pendiente. Básicamente, lo que hace el algoritmo de regresión lineal es ajustar varias líneas y retornar la línea que produce el menor error.

# Regresión Polinomial

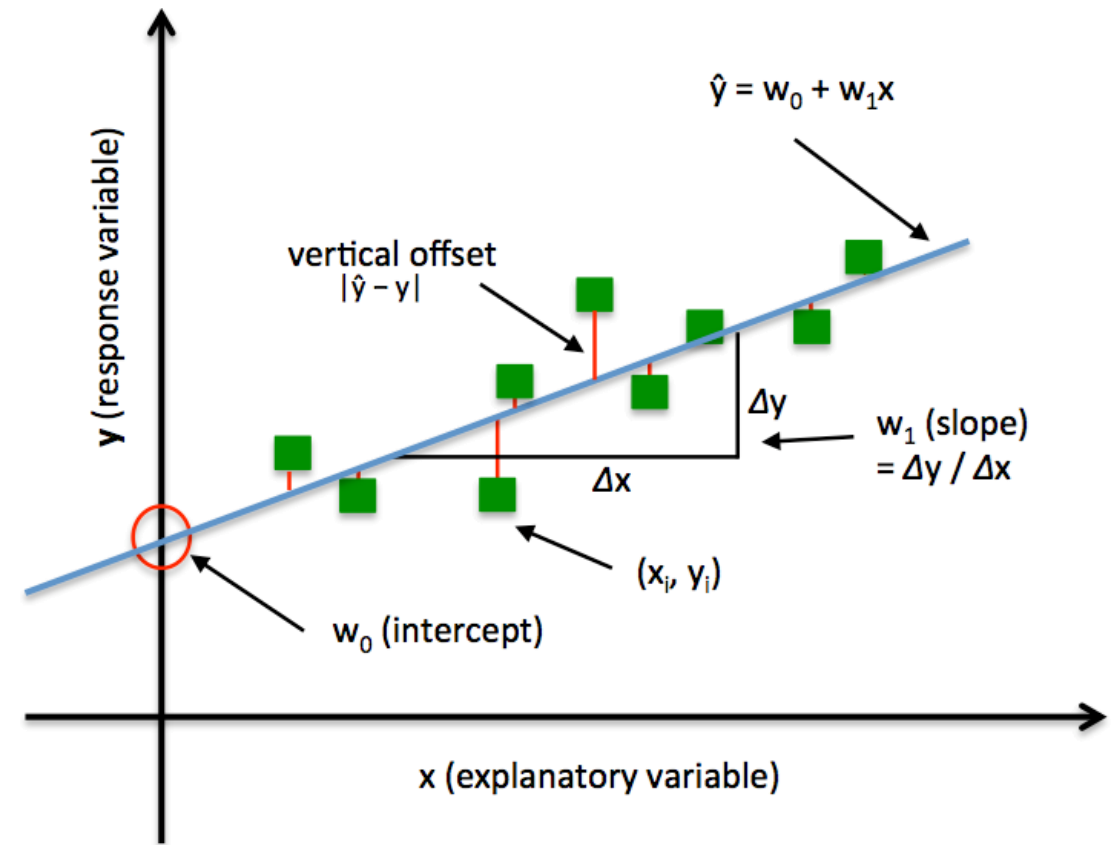
El procedimiento está diseñado para construir un modelo estadístico que describa el impacto de un solo factor cuantitativo  $x$  en una variable dependiente  $y$ . Se ajusta a los datos un modelo polinomial que involucra a  $x$  y potencias de  $x$ . Se realizan pruebas para determinar el orden apropiado del polinomio. Se puede graficar el modelo ajustado con intervalos de confianza y/o predicción. También se pueden graficar residuos e identificar observaciones influyentes.

La dependencia entre la *variable de respuesta* y la *regresora* frecuentemente no es lineal. ¿Cómo determinar la significancia de la desviación del supuesto de linealidad?

Una de las maneras más sencillas es usando la **regresión polinomial**, donde:

$$y = a + bx + cx^2 + dx^3 \dots$$

Por esta razón, la regresión polinomial se considera un caso especial de regresión lineal múltiple.



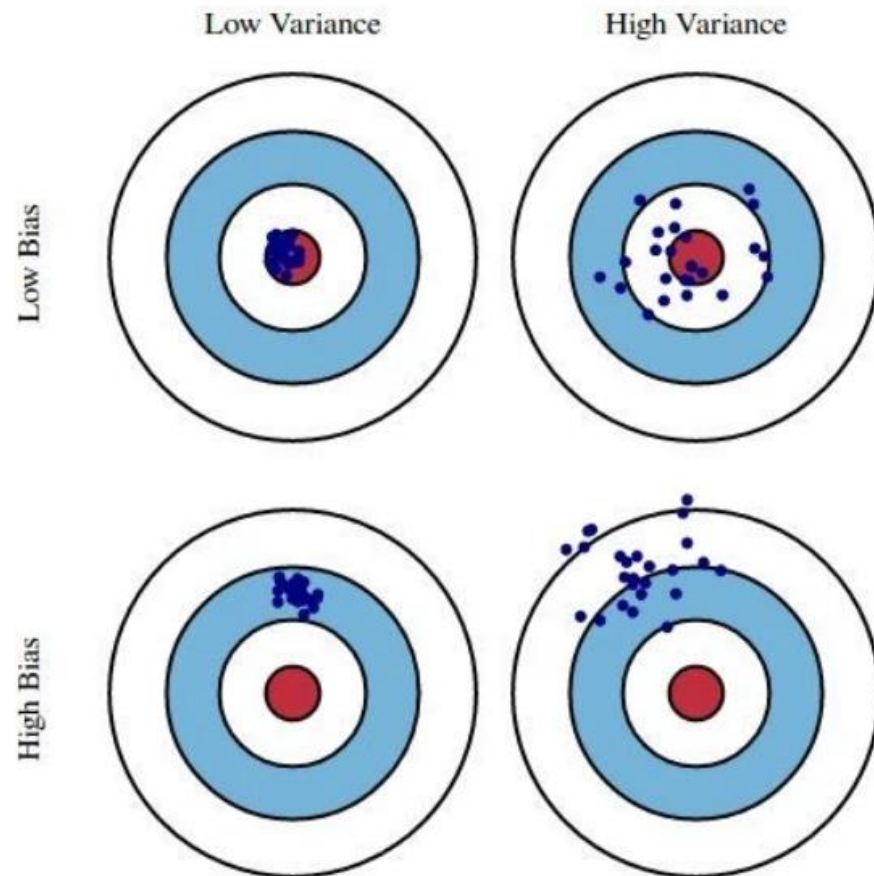
# Regresión Ridge, Lasso y Elastic-Net

## ¿Qué es el Bias?

El sesgo (bias) es la diferencia entre la predicción media de nuestro modelo y el valor correcto que intentamos predecir. Los modelos con alto sesgo prestan muy poca atención a los datos de entrenamiento y simplifican en exceso el modelo. Siempre lleva a un alto error en los datos de entrenamiento y de prueba.

## ¿Qué es la Varianza?

La varianza es la variabilidad de la predicción del modelo para un punto de datos dado o un valor que nos dice la dispersión de nuestros datos. Los modelos con alta varianza prestan mucha atención a los datos de entrenamiento y no generalizan sobre los datos que no han visto antes. Como resultado, tales modelos funcionan muy bien en los datos de entrenamiento pero tienen altas tasas de error en los datos de prueba.



Variance/bias trade off (KDnuggets.com)

Links de interés:

- <https://hackernoon.com/an-introduction-to-ridge-lasso-and-elastic-net-regression-cca60b4b934f>
- <https://www.datacamp.com/community/tutorials/tutorial-ridge-lasso-elastic-net>

<https://diegokoz.github.io/EEA/clase%2010/regularizacion.nb.html>

# Regresión Ridge, Lasso y Elastic-Net

## Regresión Ridge

Utiliza la regularización **L2** como forma de penalidad para el ajuste en la ecuación objetiva (Regresión lineal/polinomial).

El término L2 es igual al cuadrado de la magnitud de los coeficientes. En este caso si  $\lambda$  es cero entonces la ecuación es la básica, pero si es mayor que cero entonces añadimos una restricción a los coeficientes. Esta restricción da como resultado unos coeficientes minimizados que tienden a cero cuanto mayor sea el valor de  $\lambda$ .

$$+ \lambda \sum_{j=0}^p w_j^0$$

L2 regularization penalty term

# Regresión Ridge, Lasso y Elastic-Net

## Regresión Lasso

Utiliza la regularización **L1** como forma de penalidad para el ajuste en la ecuación objetiva (Regresión lineal/polinomial).

Similar a la regresión Ridge, un valor lambda de cero nos da la ecuación básica. Sin embargo, dado un valor lambda adecuado la regresión de lazo puede llevar algunos coeficientes a cero. Cuanto mayor sea el valor de lambda, más características se reducen a cero.

$$+ \lambda \sum_{j=0}^p |w_j|$$

L1 regularization penalty term



# Regresión Ridge, Lasso y Elastic-Net

## Regresión Elastic-Net

Utiliza la regularización **L1** y **L2** como forma de penalidad para el ajuste en la ecuación objetiva (Regresión lineal/polinomial).

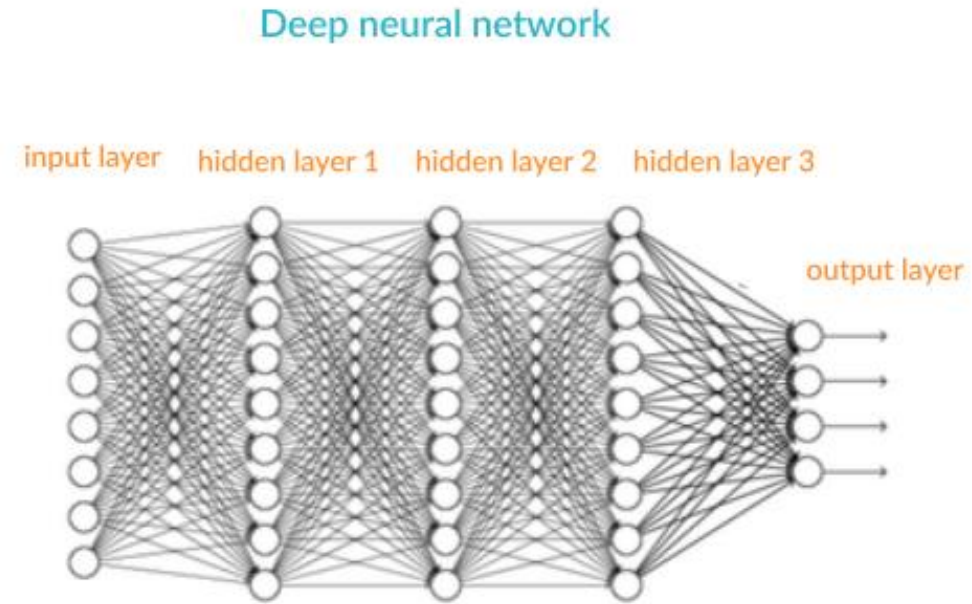
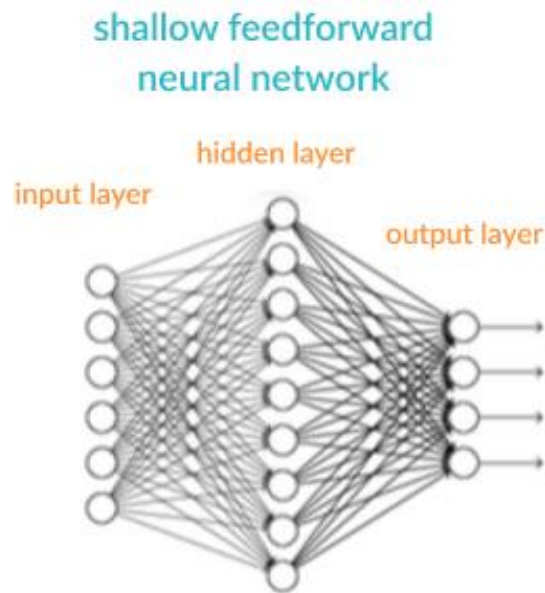
Además de configurar y elegir una Elastic-Net de valor lambda también nos permite afinar el parámetro alfa donde  $\alpha = 0$  corresponde a Ridge y  $\alpha = 1$  a Lasso. En pocas palabras, si ponemos 0 para alfa, la función de penalización se reduce al término L1 (Ridge) y si ponemos alfa en 1 obtenemos el término L2 (Lasso).

$$\frac{\sum_{i=1}^n (y_i - x_i^J \hat{\beta})^2}{2n} + \lambda \left( \frac{1 - \alpha}{2} \sum_{j=1}^m \hat{\beta}_j^2 + \alpha \sum_{j=1}^m |\hat{\beta}_j| \right)$$

Elastic net regularization

# Redes Neuronales para Regresión

Las Redes Neuronales Artificiales (ANN) están compuestas por elementos simples, llamados neuronas, cada una de las cuales puede tomar decisiones matemáticas simples. Juntas, las neuronas pueden analizar problemas complejos, emular casi cualquier función incluyendo las más complejas, y proporcionar respuestas precisas. Una red neuronal poco profunda tiene tres capas de neuronas: una capa de entrada, una capa oculta y una capa de salida. Una red neuronal profunda (DNN) tiene más de una capa oculta, lo que aumenta la complejidad del modelo y puede mejorar significativamente el poder de predicción.



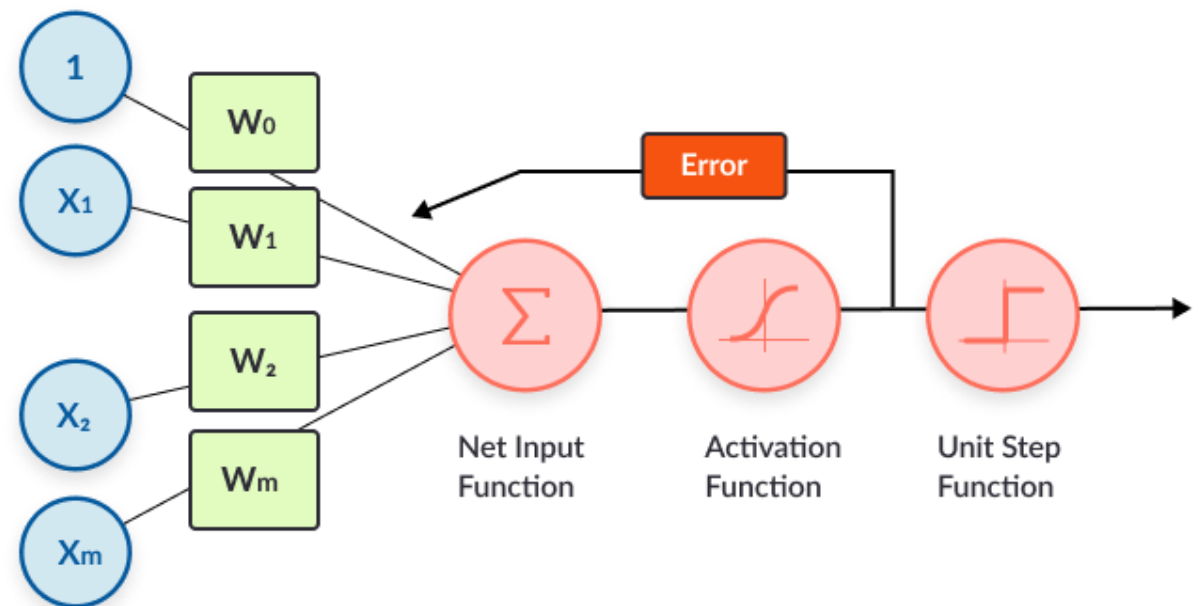
Links de interés:

- <https://missinglink.ai/guides/neural-network-concepts/neural-networks-regression-part-1-overkill-opportunity/>

# Redes Neuronales para Regresión

Las redes neuronales son reducibles a modelos de regresión: una red neuronal puede "fingir" ser cualquier tipo de modelo de regresión. Por ejemplo, esta red neuronal muy simple, con una sola neurona de entrada, una neurona oculta y una neurona de salida, es equivalente a una regresión logística. Toma varias variables dependientes igual al número de parámetros de entrada, las multiplica por sus coeficientes igual a sus pesos, los cuales pasan por una función de activación "sigmoidea" y una función de paso unitario, que se asemeja mucho a la función de regresión logística con su término de error.

La red aprenderá a través del descenso de gradiente (**backpropagation**) para encontrar coeficientes que sean mejores y se ajusten a los datos, hasta llegar a los coeficientes óptimos de regresión (o, en términos de red neuronal, los pesos óptimos para el modelo).



# Ejemplo de Regresión

Links de interés:

- <https://github.com/MGCodesandStats/datasets/blob/master/cars.csv>
- <https://datascienceplus.com/keras-regression-based-neural-networks/>

Se necesita construir un modelo capaz de estimar la cantidad de dinero que un cliente podría gastar en la compra de un vehículo a través de diferentes atributos de la persona.

- Edad
- Género
- Promedio de millas conducidas al día
- Deuda personal
- Ingreso mensual

	age	gender	miles	debt	income	sales
2	28	0	23	0	4099	620
3	26	0	27	0	2677	1792
4	30	1	58	41576	6215	27754
5	26	1	25	43172	7626	28256
6	20	1	17	6979	8071	4438
7	58	1	18	0	1262	2102
8	44	1	17	418	7017	8520
9	39	1	28	0	3282	500
10	44	0	24	48724	9980	22997

# Ejemplo de Regresión

Luego de realizar el respectivo pre-procesamiento de datos, se presenta el modelo propuesto. En este ejemplo, se muestra un red neuronal artificial poco profunda.

```
model = Sequential()  
model.add(Dense(12, input_dim=5, kernel_initializer='normal', activation='relu'))  
model.add(Dense(8, activation='relu'))  
model.add(Dense(1, activation='linear'))  
model.summary()
```

Para esto, se crea un modelo de 3 capas más 1 de entrada. Su composición es de 12 neuronas en su primera capa, 8 en su segunda capa y 1 en su capa de salida. Nótese que la capas diferentes a la neurona de salida utilizan la función de activación 'relu', a diferencia de la ultima capa que maneja una función de activación lineal. Se compila el modelo utilizando el optimizador ADAM y la función de pérdida MSE (Mean Squared Error).

```
model.compile(loss='mse', optimizer='adam', metrics=['mse', 'mae'])
```

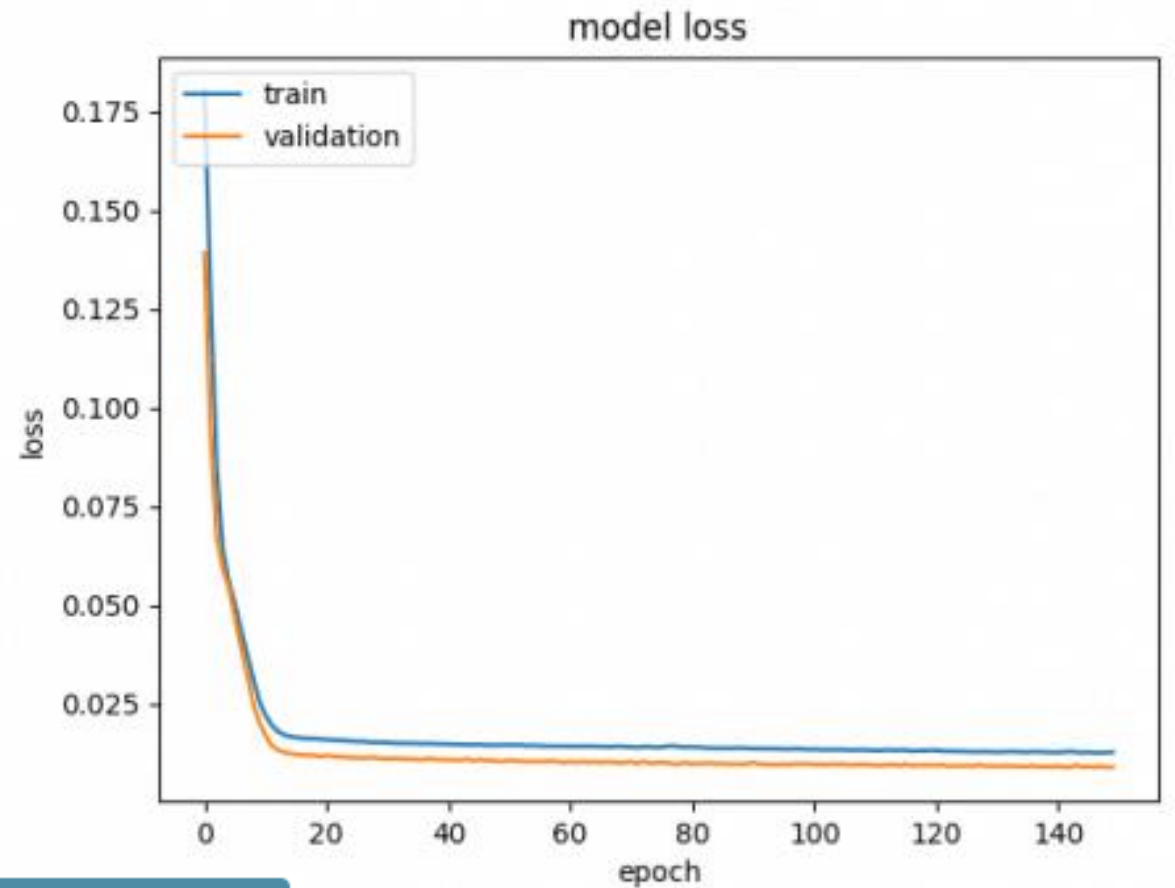
# Ejemplo de Regresión

En su gráfica de entrenamiento se puede observar como el modelo fue minimizando la función de pérdida (MSE) elegida anteriormente.

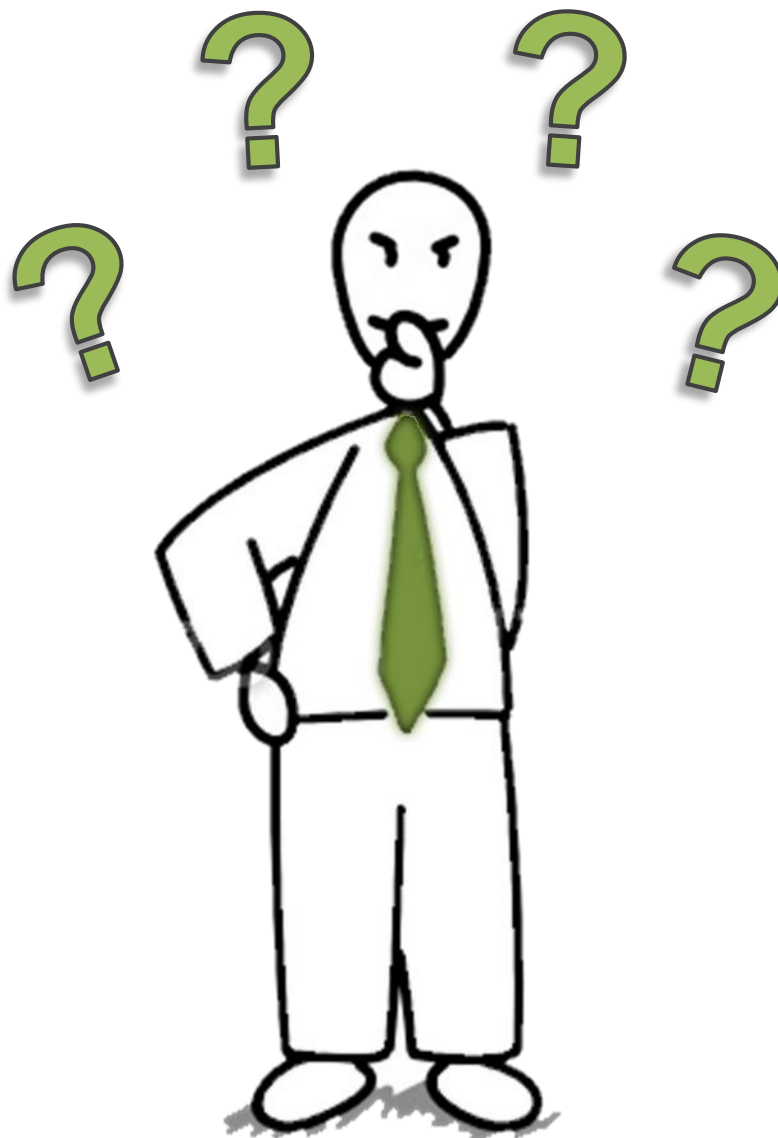
Aquí podemos observar un ejemplo nuevo que nunca ha visto el modelo, junto con su valor estimado

```
Xnew = np.array([[40, 0, 26, 9000, 8000]])
Xnew= scaler_x.transform(Xnew)
ynew= model.predict(Xnew)
#invert normalize
ynew = scaler_y.inverse_transform(ynew)
Xnew = scaler_x.inverse_transform(Xnew)
print("X=%s, Predicted=%s" % (Xnew[0], ynew[0]))

X=[ 40.    0.    26. 9000. 8000.], Predicted=[13686.491]
```






# Preguntas





# Motivación

## OBSERVE EL VIDEO Y RESPONDA A LAS SIGUIENTES PREGUNTAS:

-  ¿Cuántos datos se requieren para entrenar un sistema de visión artificial?
-  ¿Es posible decir que los computadores ya sobrepasaron la capacidad humana?
-  ¿Qué problemas evidencian los sistemas de visión artificial, y en general de los sistemas de Reconocimiento de Patrones?



[https://www.ted.com/talks/fei\\_fei\\_li\\_how\\_we\\_re\\_teaching\\_computers\\_to\\_understand\\_pictures?language=es](https://www.ted.com/talks/fei_fei_li_how_we_re_teaching_computers_to_understand_pictures?language=es)





UNIVERSIDAD  
**NACIONAL**  
DE COLOMBIA