



UNIVERSIDAD DE CHILE

# Deep Learning

Deeper, Better, \_\_\_\_\_, Stronger than Machine Learning

---

Valentin Barriere

Universidad de Chile – DCC

CC6204, Primavera 2024

# Deep Learning

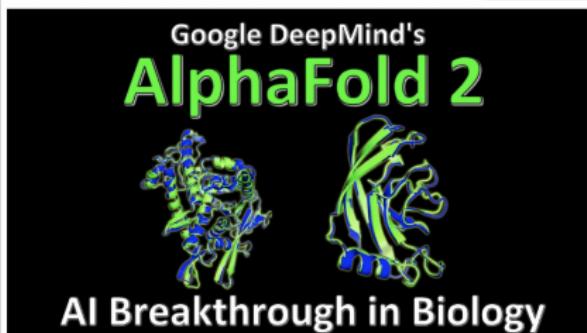
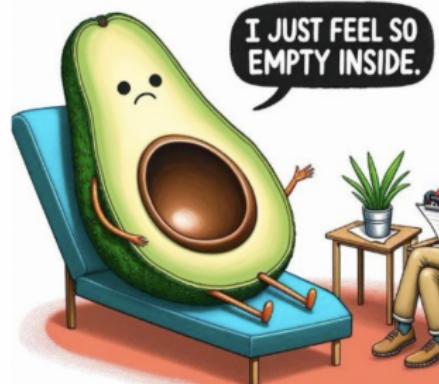
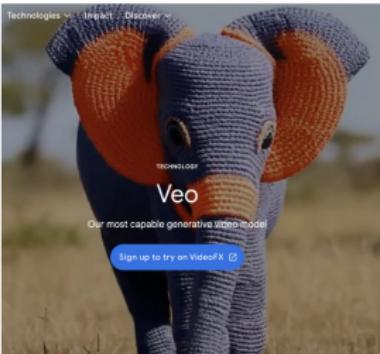


Figure 1: Conversational Agent, Image/Video generation, Protein structure

# Outline : Significacion de las termas

Significacion de las termas

Aplicaciones

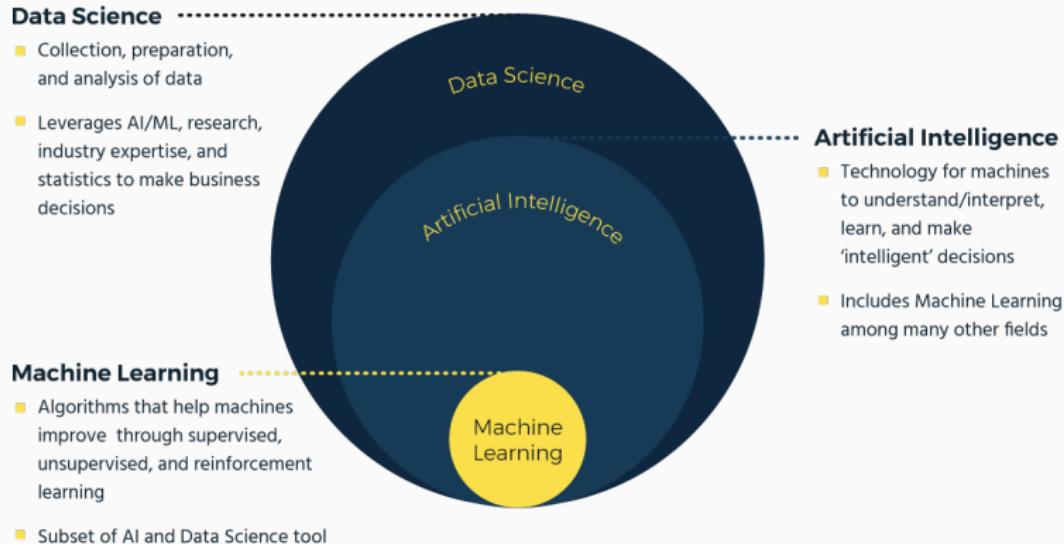
Historia

Los recursos

Principe

El curso

## AI vs. Data Science vs. Machine Learning



**Figure 2:** Diferencias entre campos

## En resumen

Data Science se centra en el análisis de datos para extraer conocimiento, Machine Learning utiliza algoritmos para hacer predicciones y tomar decisiones basadas en datos, y Artificial Intelligence se refiere al desarrollo de sistemas que pueden realizar tareas inteligentes de manera autónoma.

Definición (sobre) simplista:

- Data mining genera entendimiento.
- Machine learning genera predicciones.
- Artificial intelligence genera acciones.

# Ejemplo en plataforma de música

## Data Scientist

Recopila y analiza datos de usuarios de plataformas de música para identificar patrones y preferencias musicales.

## Machine Learner

Desarrolla y optimiza un modelo de recomendación de música utilizando algoritmos de aprendizaje automático para predecir las preferencias de los usuarios.

## Artificial Intelligence

Implementa un agente social que puede interagir con el usuario, para mejorar la personalización de las recomendaciones musicales y proporcionar una experiencia más precisa y contextualizada.

# Datos

---

Diferentes tipos de datos:

- Datos estructurados:
  - Datos sociales: Edad, Salario, Color de piel, Lugar de residencia
  - Datos métricos: *Likes* de una publicación, Tiempo pasado en una página, Número de amigos en común
- Datos no estructurados:
  - Texto: Frase, Párrafo, Documento
  - Sonido: Canción, Discurso
  - Imagen: Foto, Vídeo

## Diferentes tipos de Minería:

- Exploración de datos: Detectar valores simples, sesgos
- Tarea de clasificación/regresión: Alimentarse de datos para caracterizar nuevos datos **por clase o con un valor**, de manera supervisada
- Tarea de agrupamiento: Caracterizar datos por clase de manera no supervisada
- Reducción de dimensiones: Desarrollar estructuras comunes para representaciones comprimidas de datos

## Diferentes tipos de Minería:

- Exploración de datos: Detectar valores simples, sesgos
- Tarea de clasificación/regresión: Alimentarse de datos para caracterizar nuevos datos **por clase o con un valor**, de manera supervisada
- Tarea de agrupamiento: Caracterizar datos por clase de manera no supervisada
- Reducción de dimensiones: Desarrollar estructuras comunes para representaciones comprimidas de datos
- **Generacion: generar datos segun una distribucion particular**  
(mas complejo)

# Outline : Historia

Significacion de las termas

**Historia**

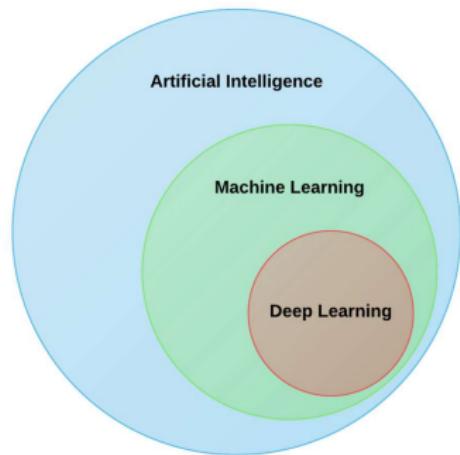
Principe

Aplicaciones

Los recursos

El curso

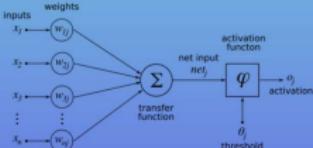
## Qué es Deep learning?



## Redes neuronales no siempre fueron profundas

**1943** El primer modelo matemático de una neurona (McCulloch-Pitts)

**1957** El perceptrón



**1959** Hubel y Wiesel descubrieron las células simplex y complex en sistemas de visión biológica.

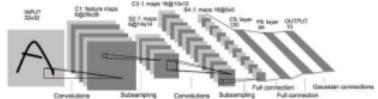
**1965** Primera red profunda (8 capas) – Ivakhnenko and Lapa

**1979** Red neuronal para reconocer patrones visuales (Neocognitron) – Fukushima

**1982** Primera red recurrente – Hopfield

**1986** Algoritmo Backpropagation

**1989** Redes neuronales convolucionales – Handwritten recognition



**1989** Reinforcement learning – Q-learning

## Redes neuronales no siempre fueron profundas (no tanto)



**2015 ~ Explosión de la industria DL**

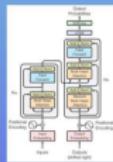


# Historia

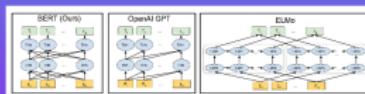
## Redes neuronales profundas

**2014 Mecanismo de atención – Bahdanau et al.**

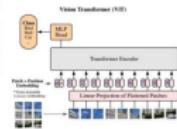
**2017 Arquitectura Transformer – Machine translation.**



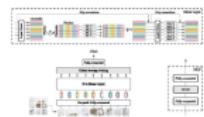
**2018-2020 Modelos de NLP: BERT, GPTs, XLM**



**2020 Vision Transformer**



**2021 MLP Mixer**



**2022 Dall-e, Imagen, LaMDA**

# Outline : Principe

Significacion de las termas

Aplicaciones

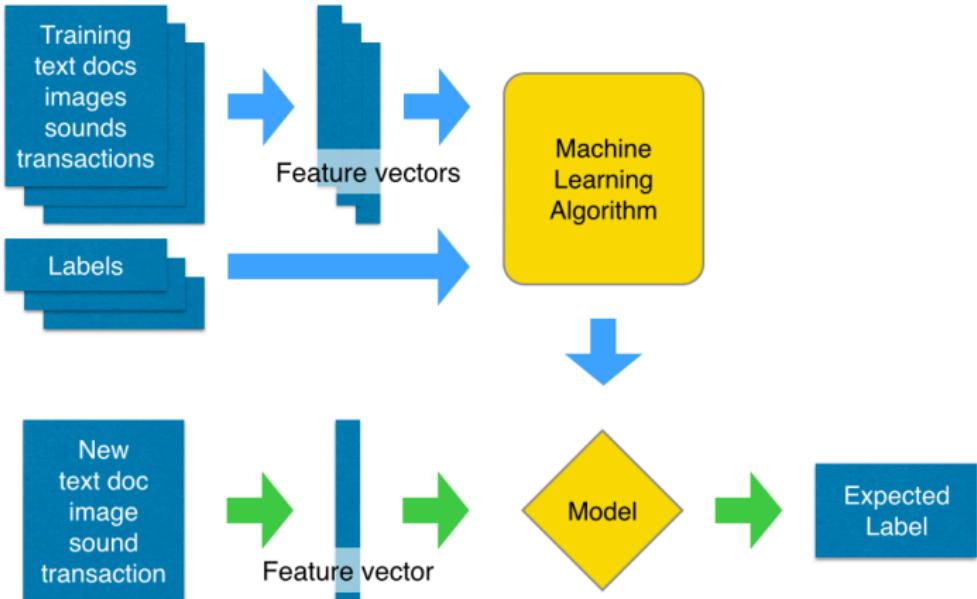
Historia

Los recursos

Principe

El curso

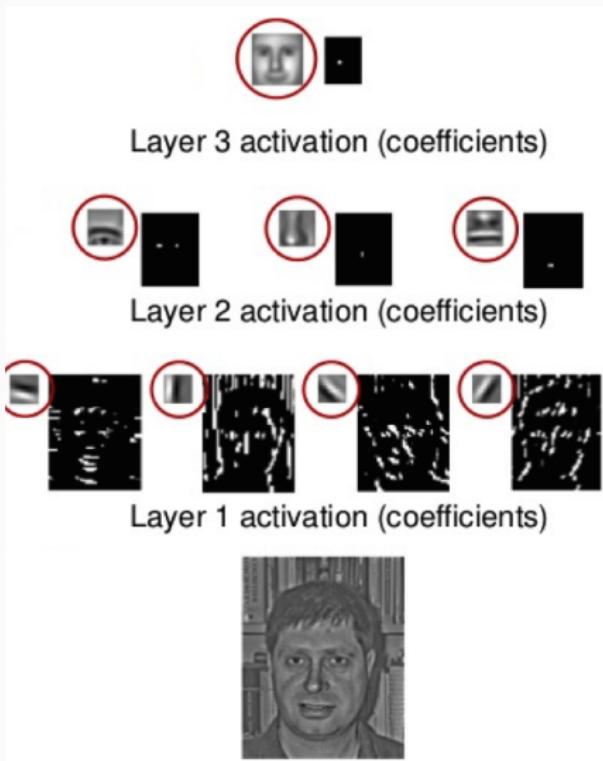
# Supervised Machine Learning



Predictive Modeling Data Flow

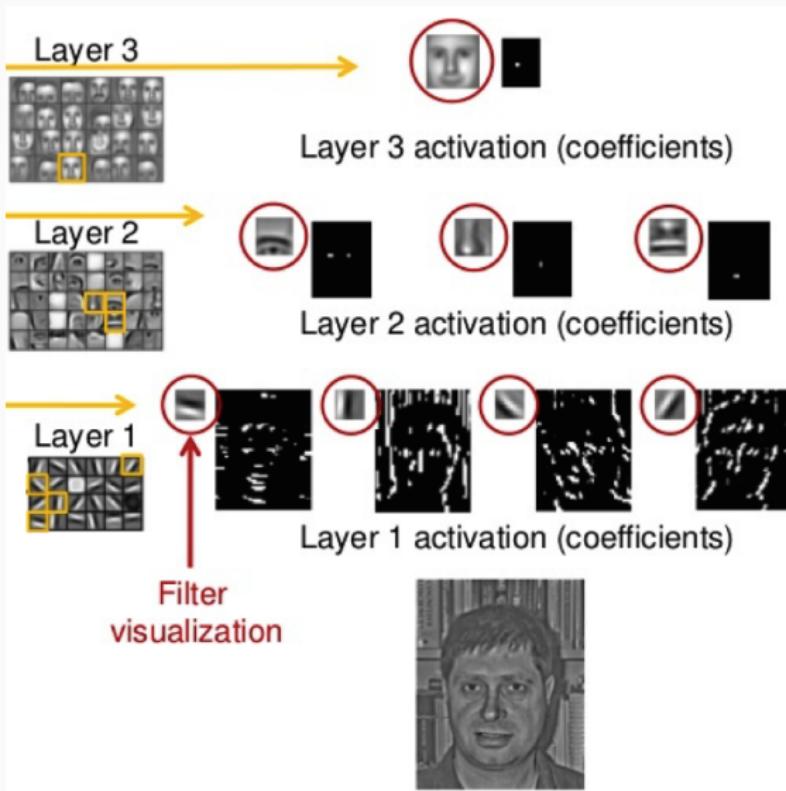
ML: Need to **create the feature vectors** to use them as input.

## Complejidad de las representaciones: Ejemplo



DL **extracts the feature vectors from raw data**

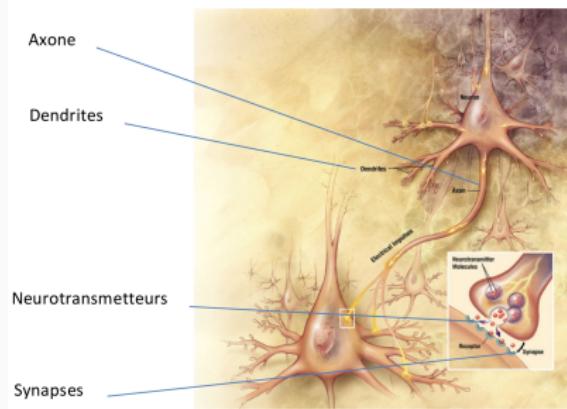
# Complejidad de las representaciones: Composicionalidad



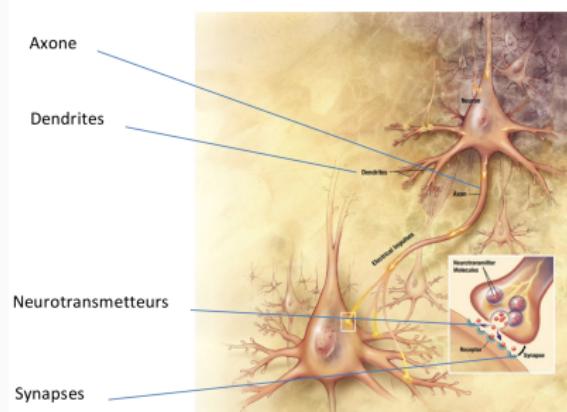
DL extracts the feature vectors from raw data and **compose with them**

# Primero, ¿por qué este nombre?

Una neurona recibe varias informaciones (neurotransmisores) a nivel de sus **dendritas**. Estos neurotransmisores son liberados por las **sinapsis**. Cuando la cantidad de información supera un cierto **umbral**, la neurona se "activa", enviando una corriente eléctrica a través de su axón, lo que le permite **emitir a su vez** neurotransmisores a través de sus sinapsis.



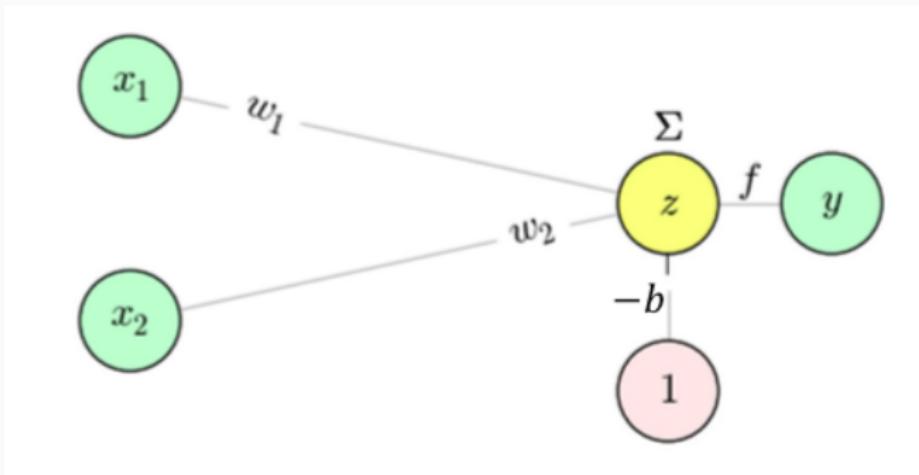
# Primero, ¿por qué este nombre?



## Para resumir:

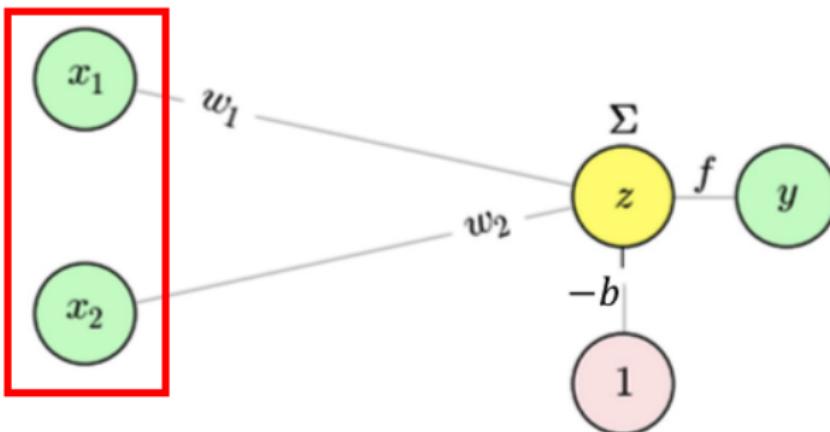
- Las salidas de una neurona son las entradas de otra.
- Una neurona emite cuando recibe una cantidad de información que supera un umbral.
- La cantidad de información emitida a la siguiente neurona es gestionada por las sinapsis, que se activan a partir de un umbral.

# Perceptrón – Presentación



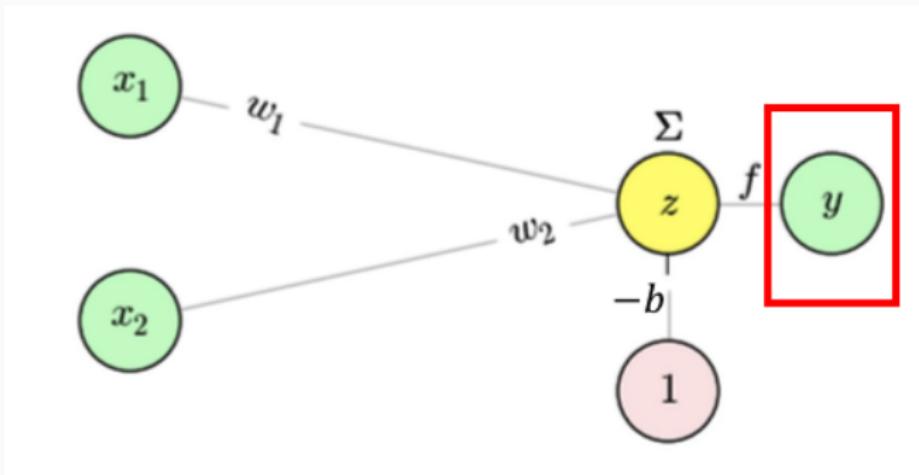
$$y = f(x_1 w_1 + x_2 w_2 - b)$$

# Perceptrón – Presentación



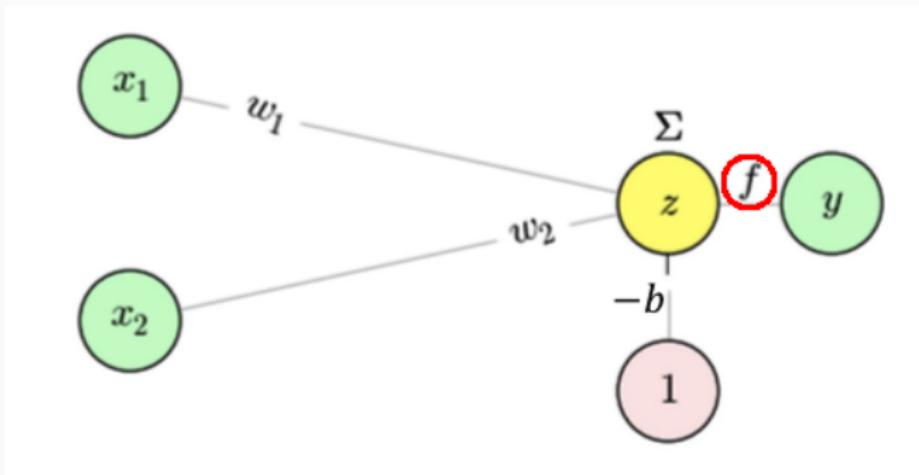
$$y = f(x_1 w_1 + x_2 w_2 - b)$$

# Perceptrón – Presentación



$$y = f(x_1 w_1 + x_2 w_2 - b)$$

## Perceptrón – Presentación



$$y = f(x_1 w_1 + x_2 w_2 - b)$$

En teoría  $f$  es la función de Heaviside:  $f(z) = \mathbb{1}_{\mathbb{R}_+} = \begin{cases} 1 & \text{si } z \geq 0 \\ 0 & \text{si } z < 0 \end{cases}$

# Outline : Aplicaciones

Significacion de las termas

Historia

Principe

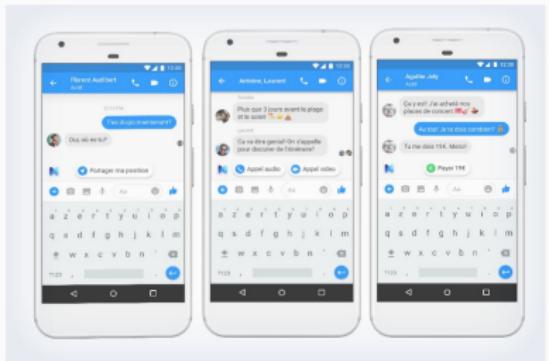
Aplicaciones

Los recursos

El curso

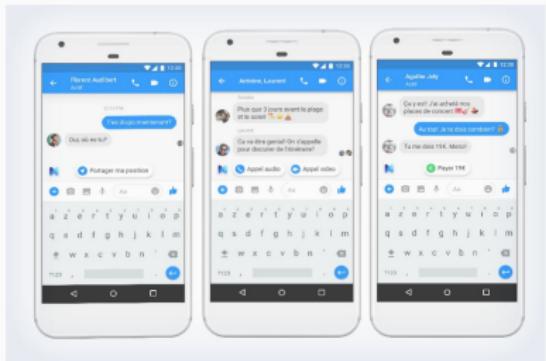
# Aplicaciones (I/II)

- Detección de eventos en un texto



# Aplicaciones (I/II)

- Detección de eventos en un texto

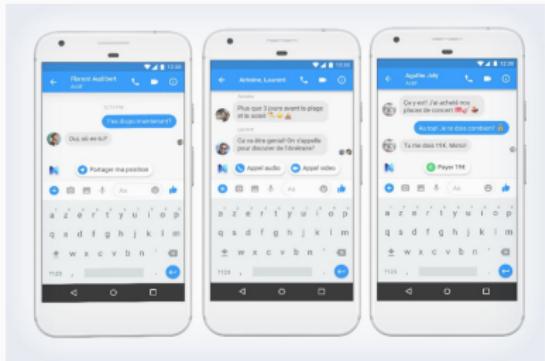


- Procesamiento automático de opiniones de usuarios



# Aplicaciones (I/II)

- Detección de eventos en un texto



- Procesamiento automático de opiniones de usuarios



Les connaissez-vous ?



**Marina Dunion**  
Digital Marketing @Air France  
& Co-Founder @FlexiFly  
● Teddy Viraye-Chevalier et 3 autres relations



**Salvatore Anzalone**  
Post-Doc at ISIR, University Pierre et Marie Curie, Paris  
● Thomas Janssone et 2 autres relations

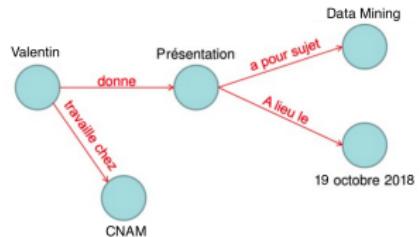


**Halla Olafsdottir**  
Medical Solutions Project Manager | Chef de Projet  
● Télécom ParisTech

- Propuesta de recomendaciones a un usuario

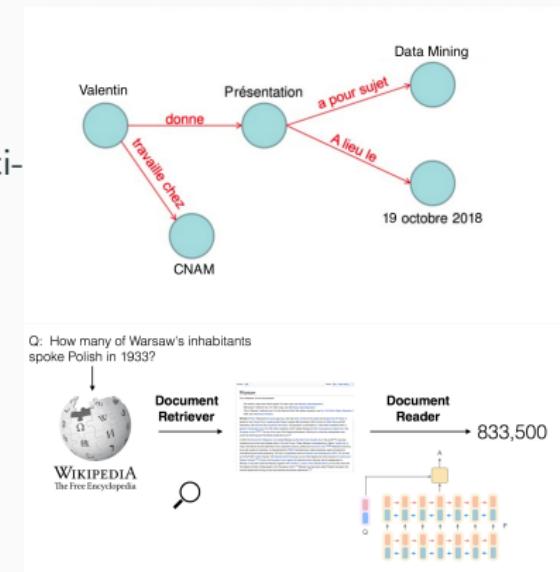
## Aplicaciones (II/II)

- Detección de relaciones entre entidades en un texto



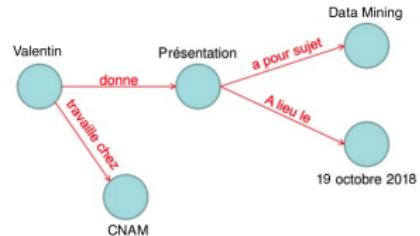
# Aplicaciones (II/II)

- Detección de relaciones entre entidades en un texto
- Respuesta a una pregunta

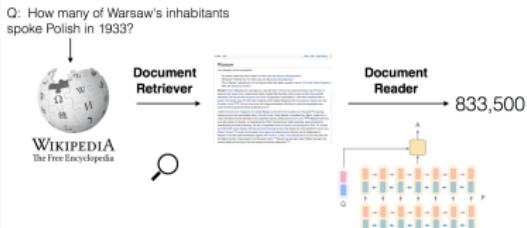


# Aplicaciones (II/II)

- Detección de relaciones entre entidades en un texto



- Respuesta a una pregunta

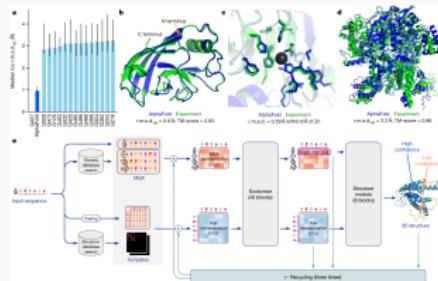


- Módulo de IE para un agente conversacional



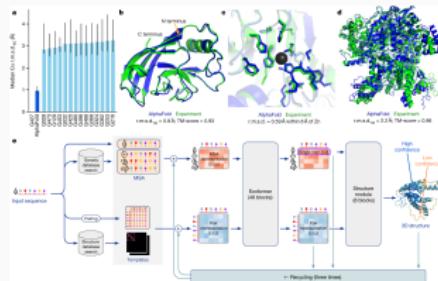
# Significación del trabajo: Porque hacer eso?

- Avance científico



# Significación del trabajo: Porque hacer eso?

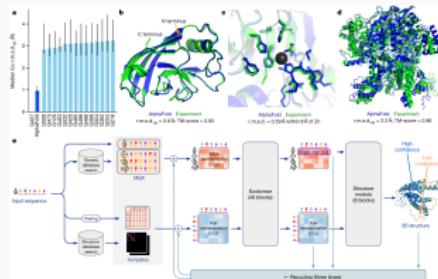
- Avance científico



- Prevención y gestión de desastres naturales

# Significación del trabajo: Porque hacer eso?

- Avance científico



- Prevención y gestión de desastres naturales

- Impacto en la salud pública

**Open Chronic**

Améliorer la prise en charge des malades chroniques

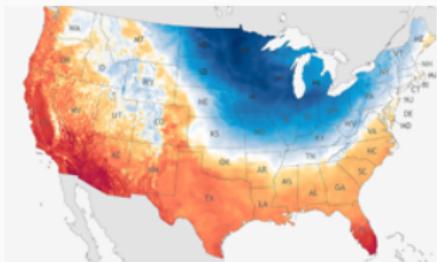
Santé    Promotion 3

Ministère de la santé, Direction de la recherche, des études, de l'innovation et des statistiques

Paris    Data science

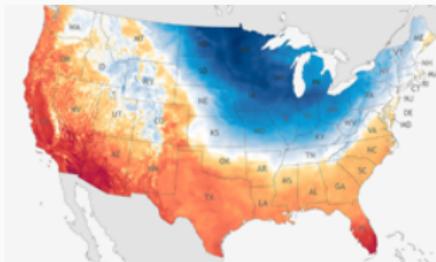
# Significación del trabajo: Porque hacer eso?

- Sostenibilidad ambiental

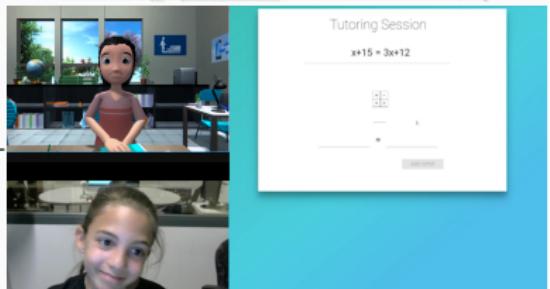


# Significación del trabajo: Porque hacer eso?

- Sostenibilidad ambiental

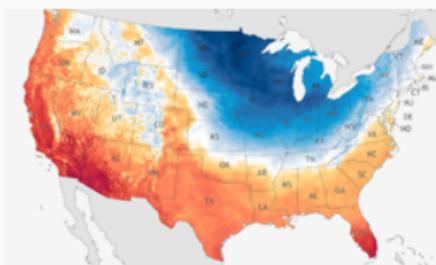


- Impulso a la educación y la investigación

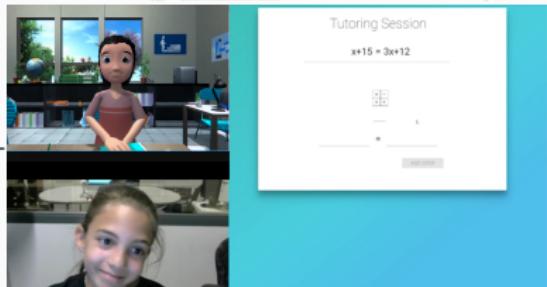


# Significación del trabajo: Porque hacer eso?

- Sostenibilidad ambiental



- Impulso a la educación y la investigación



- Democracia participativa



# Outline : Los recursos

Significacion de las termas

Historia

Principe

Aplicaciones

**Los recursos**

El curso

# Material

- Computadora
- Jupyter Notebook y Anaconda:  
<https://www.anaconda.com/download/>
- Los notebooks y las cheatsheets disponibles online:

**Python For Data Science Cheat Sheet**  
NumPy Basics

Learn Python for Data Science interactively at [www.DataCamp.com](http://www.DataCamp.com)

**NumPy**

The NumPy library is the core library for scientific computing in Python. It provides a high-performance multidimensional array object, and tools for working with these arrays.

Use the following import convention:  
`>>> import numpy as np`

**NumPy Arrays**

1D array      2D array      3D array

**Creating Arrays**

**Python For Data Science Cheat Sheet**  
Matplotlib

Learn Python interactively at [www.DataCamp.com](http://www.DataCamp.com)

**Matplotlib**

Matplotlib is a Python 2D plotting library which produces publication-quality figures in a variety of hardcopy formats and interactive environments across platforms.

**1) Prepare the Data** Also see Lists & NumPy

**1D Data**

```
>>> import numpy as np  
>>> x = np.linspace(0, 10, 100)  
>>> y = np.cos(x)  
>>> z = np.sin(x)
```

**2D Data or Images**

```
>>> data = 2 * np.random.random((100, 100))  
>>> data -= data.mean() / 100  
>>> Y, X = np.mgrid[-3:3:100j, -3:3:100j]  
>>> Z = np.exp(-X**2 - Y**2)  
>>> Z = Z / Z.max()
```

**Python For Data Science Cheat Sheet**  
Scikit-Learn

Learn Python for data science interactively at [www.DataCamp.com](http://www.DataCamp.com)

**Scikit-learn**

Scikit-learn is an open source Python library that implements a range of machine learning, preprocessing, cross-validation and visualization algorithms using a unified interface.

**A Basic Example**

```
>>> from sklearn import neighbors, datasets, preprocessing  
>>> digits = datasets.load_digits()  
>>> X_train = digits.data[0:-100] # 100 target  
>>> X_test = digits.data[-100:] # 100 target  
>>> y_train = digits.target[0:-100]  
>>> y_test = digits.target[-100:]  
>>> knn = neighbors.KNeighborsClassifier(n_neighbors=5)  
>>> knn.fit(X_train, y_train)  
>>> y_pred = knn.predict(X_test)  
>>> accuracy_score(y_test, y_pred)
```

**Creat**  
**Sup**

Lines  
>>> t  
>>> l  
>>> r  
>>> N  
>>> E  
>>> S  
>>> K  
>>> KNN  
>>> k  
>>> k  
**Unst**  
**Princ**  
>>> p  
**K Me**  
>>> t  
>>> k  
**Mod**

**Python For Data Science Cheat Sheet**  
Pandas Basics

Learn Python for Data Science interactively at [www.DataCamp.com](http://www.DataCamp.com)

**Pandas**

The Pandas library is built on NumPy and provides easy-to-use data structures and data analysis tools for the Python programming language.

Use the following import convention:  
`>>> import pandas as pd`

**Pandas Data Structures**

**Series**

A one-dimensional labeled array capable of holding any data type

**Python For Data Science Cheat Sheet**  
Jupyter Notebook

Learn More Python for Data Science interactively at [www.DataCamp.com](http://www.DataCamp.com)

**Saving/Loading Notebooks**

Create new notebook

Open an existing notebook

Make a copy of the current notebook

Save current notebook and record checkpoint

Preview of the printed notebook

Close notebook & stop running any scripts

**Writing Code And Text**

**Python For Data Science Cheat Sheet**  
Keras

Learn Python for data science interactively at [www.DataCamp.com](http://www.DataCamp.com)

**Keras**

Keras is a powerful and easy-to-use deep learning library for Theano and TensorFlow that provides a high-level neural networks API to develop and evaluate deep learning models.

**A Basic Example**

```
>>> import numpy as np  
>>> from keras.models import Sequential  
>>> from keras.layers import Dense  
>>> data = np.random.randint(1000,10000)  
>>> data = np.reshape(data,(1000,1000))  
>>> model = Sequential([  
>>>     Dense(128, activation='relu',  
>>>         input_dim=1000),  
>>>     Dense(10, activation='softmax')])  
>>> model.compile(optimizer='adam',  
>>>                 loss='binary_crossentropy',  
>>>                 metrics=['accuracy'])  
>>> model.fit(data,labels,epochs=10,batch_size=32)
```

**Mod**  
**Sequ**  
>>> m  
>>> ms  
>>> m  
**Mult**  
**Grid**  
>>> fsi  
>>> msi  
>>> m  
>>> m  
**Multi-C**  
>>> fci  
>>> mci  
>>> m  
>>> m  
**Regress**  
>>> msc  
>>> m  
**Conv**

# Recursos

- Libros
  - Deep learning, Goodfellow, Bengio, Courville (  
<https://www.deeplearningbook.org/>)
  - Neural networks and Deep learning, Nielsen (  
<http://neuralnetworksanddeeplearning.com/>)
  - Dive into Deep Learning. Zhang, Lipton, Li, Smola (  
<https://d2l.ai/>)
- Otros
  - CS231n - Stanford: Deep Learning for Computer Vision (  
<http://cs231n.stanford.edu/>)
  - CS224N: Natural Language Processing with Deep Learning (  
<https://web.stanford.edu/class/cs224n/>)
  - Deep learning – New York University (  
<https://atcold.github.io/pytorch-Deep-Learning/>)
  - CS224W: Machine Learning with Graphs (  
<http://web.stanford.edu/class/cs224w/>)
- Canal Discord: <https://discord.gg/SZD3EqHZ>

Biblioteca de Python para Deep Learning (Redes Neuronales Profundas).  
Es de alto nivel que corre sobre Tensorflow y otros backends.



- Visión: [Tutorial de Keras para afinar un VGG16 preentrenado](#), que puede ser utilizado con:  
[Diferentes modelos de CNN preentrenados disponibles en Keras](#)
- Texto y audio:  
[Tutorial RNN-LSTM Seq2seq para traducción automática](#)
- Texto: [Uso de embeddings de palabras preentrenados](#)

Biblioteca de Python para Deep Learning (Redes Neuronales Profundas)  
concurrente de Tensorflow



- Visión: [Tutorial para afinar un ResNet18 preentrenado](#)
- Audio: [Reconocimiento de voz con Wav2Vec2](#)
- Texto: [Tutorial RNN-GRU Seq2seq para traducción automática](#)

Biblioteca de Python para **Transformers** (Tipo de Modelos de Redes Neuronales Profundas)



## Hugging Face

- Muchos modelos ya pre-entrenados, para imágenes, audio, multimodal, texto, ...
- Modelos clásicos, modelos generativos, embeddings, ... hasta los más grandes y nuevos (tipo LLaMA3-70B)
- Otras librerías: Diffuser, Datasets, Accelerate, PEFT, bitsandbytes, TRL, ...

# Outline : El curso

---

Significacion de las termas

Historia

Principe

Aplicaciones

Los recursos

**El curso**

# Evaluaciones

---

Ya no esta definido!

- Tareas?
- Proyecto en grupos?
- Quizz?

# El programa

Todo no esta definido! Que quieren ver después las bases?

- **Classical language modeling:** from Shannon to LLM-based Chatbot
- **Audio models:** audio speech recognition, source separation, audio scene classification, general models
- **Multimodal models:** classical ones, CLIP, Diffusion, LLM-based, generative ones for vision or audio
- **Large text generative models (LLM):** Instructions, RLHF, Agent using tools, Retrieved Augmented Generation
- **Efficiency:** how to train/deploy an LLM on a small budget (parameter efficient fine tuning, quantization, optimized specific libraries like vLLM)

**Les voy a mandar un encuesta!**

**Questions?**

## References i