

Tema 4.

Trabajo Práctico 5.

Tema. IA conexionista.

Propuestas de solución.

Indice

A continuación se menciona una propuesta de abordaje de modelos de Aprendizaje Automático o Machine Learning (ML) / Ciencia de Datos,

- Definiciones de IA
- Tecnologías de IA para modelar y simular toma de decisiones
- Algunos métodos / metodologías / workflows, pipelines (pueden ser utilizados / adaptados en sus proyectos)
- Herramientas de programación (frameworks, librerías para implementar código)
- Recursos, datasets (selección de datasets para experimentar con modelos de ML)
- Métricas (para evaluar problemas de predicción, clasificación, otros)

Definiciones de IA

Sistemas que
piensan como
humanos

“La interesante tarea de lograr que las computadoras piensen... *Máquinas con mente*, en su amplio sentido literal”
(Haugeland, 1985)

“La automatización de] actividades que vinculamos con procesos de pensamiento humano, actividades tales como la toma de decisiones, resolución de problemas, aprendizaje...”
(Bellman, 1978)

Sistemas que
actúan como
humanos

“El arte de crear máquinas con capacidad de realizar funciones que realizadas por personas requieren inteligencia” (Kurzweil, 1990)

“El estudio de cómo lograr que las computadores realicen tareas que, por el momento, los humanos hacen mejor” (Rich y Knight, 1991)

“El estudio de las facultades mentales mediante el uso de modelos computacionales”
(Carniak y McDermott, 1985)

“El estudio de los cálculos que permiten percibir, razonar y actuar”
(Winston, 1992)

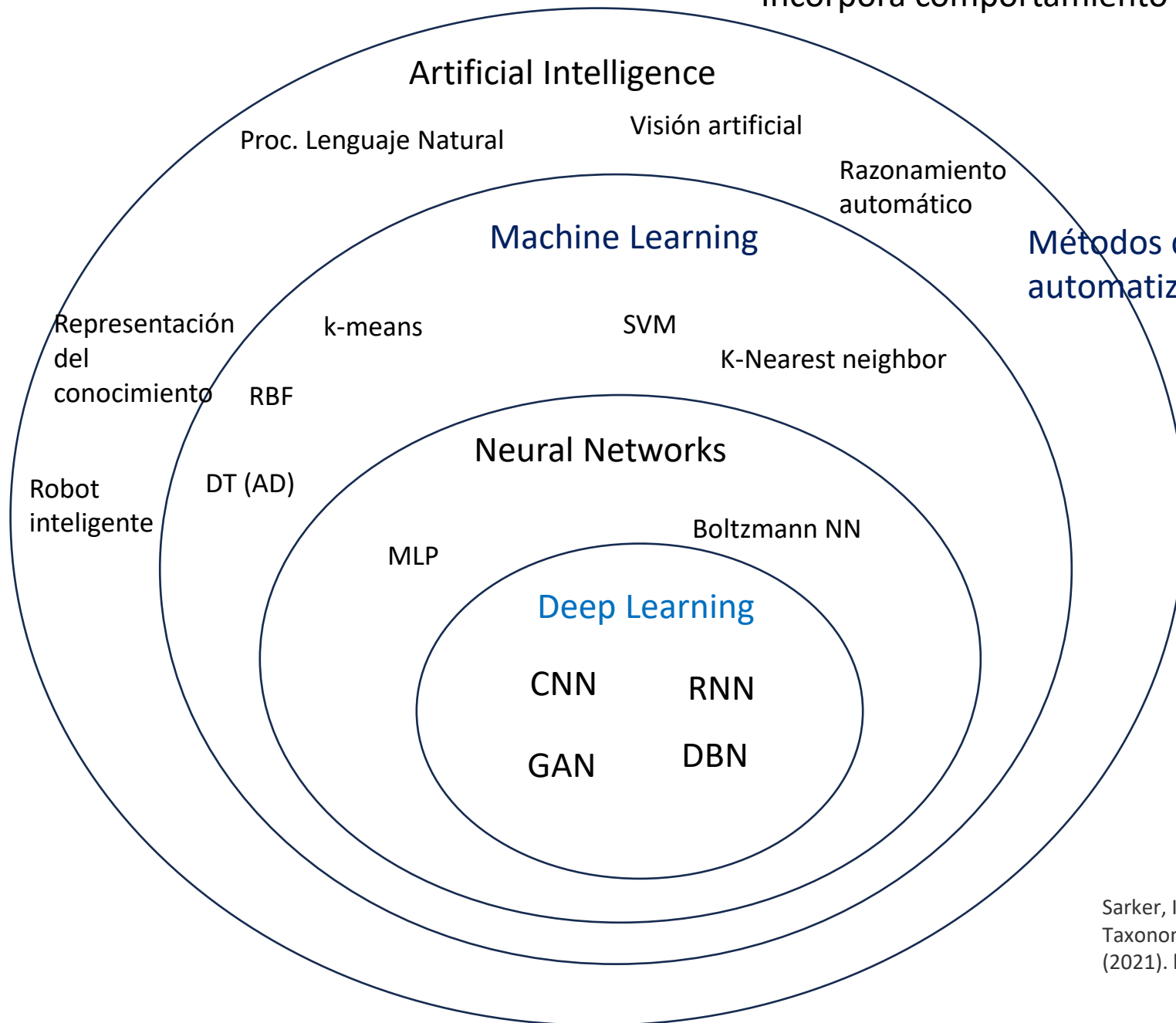
“Un campo de estudio que se enfoca a la explicación y emulación de la conducta inteligente en función de procesos computacionales”
(Schalkoff, 1990).

“La rama de la ciencia de la computación que se ocupa de la automatización de la conducta inteligente”
(Luger y Stubblefield, 1993).

Sistemas que
piensan
racionalmente
(idealmente)

Sistemas que
actúan
racionalmente
(idealmente)

Incorpora comportamiento humano e inteligencia a máquinas o sistemas

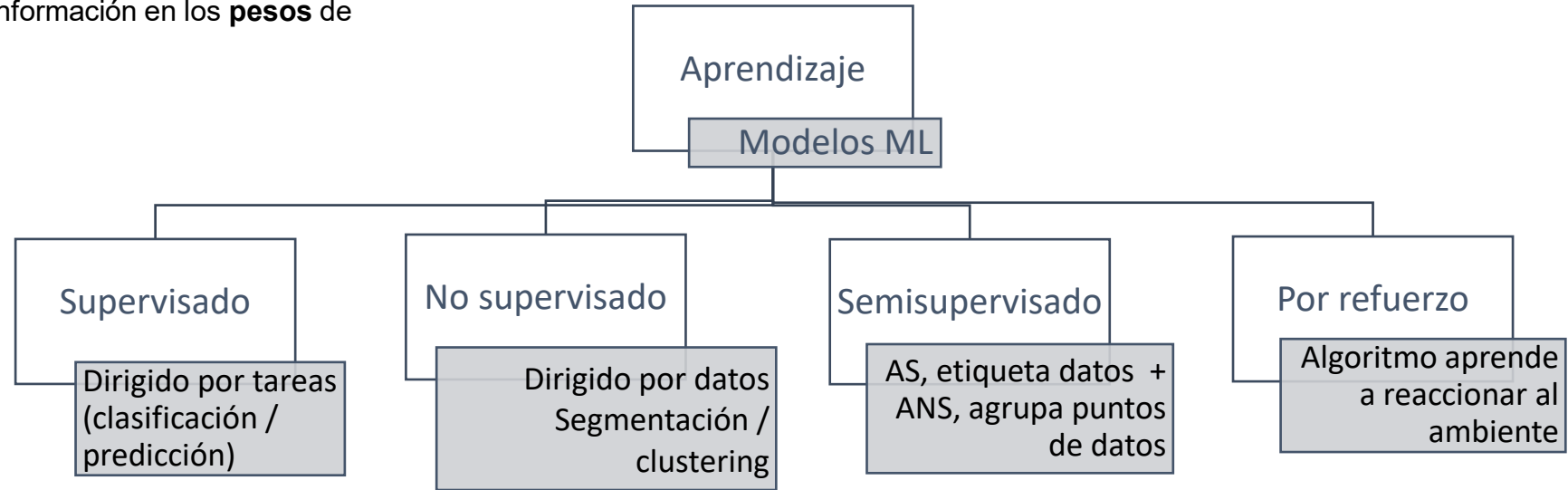


Métodos que aprenden de datos o experiencia, automatización en la construcción de modelos

Computación utilizando RN multi-layer

Conocimiento y Aprendizaje en RNA

- En RNA se denomina **aprendizaje** al proceso que consiste en almacenar la información en los **pesos** de las **conexiones**.



El aprendizaje de una RNA se define en sus algoritmos. Ej. algoritmo retropropagación

Un modelo de RNA se diferencia de otros modelos de la IA, generan su propio **conocimiento**

aprendiendo de los patrones o ejemplos proporcionados. El conocimiento se encuentra en la estructura de la red y los pesos optimizados disponibles entre las neuronas o nodos.

Alternativas en la construcción de modelos de ML

Construir modelos para proponer soluciones a:

- problemas de predicción y
- problemas de clasificación

Construir modelos:

- utilizando todas las variables evidenciales
- utilizando variables evidenciales relevantes
- utilizando distintos clasificadores o técnica de ML [Ej.: Redes Neuronales, Regresión Logística, Máquinas de Vectores de Soporte, Vecinos más Cercanos y Árboles de Decisión Clasificación, Naive Bayes, otros], o combinando clasificadores
- variando los parámetros/hiperparámetros según clasificador o técnica de ML

Comparar los modelos construidos aplicando métricas

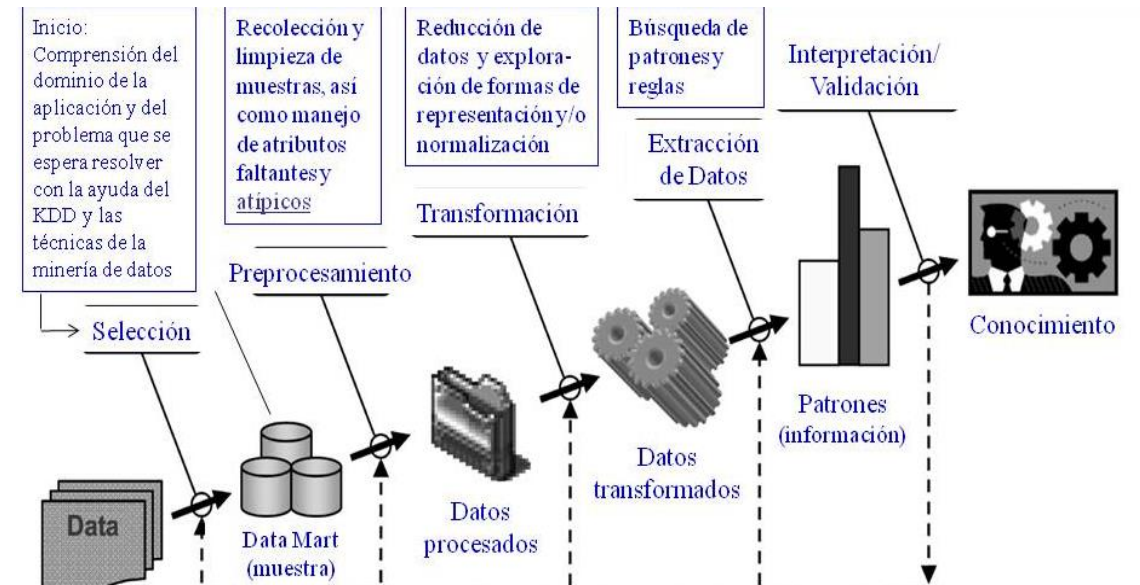
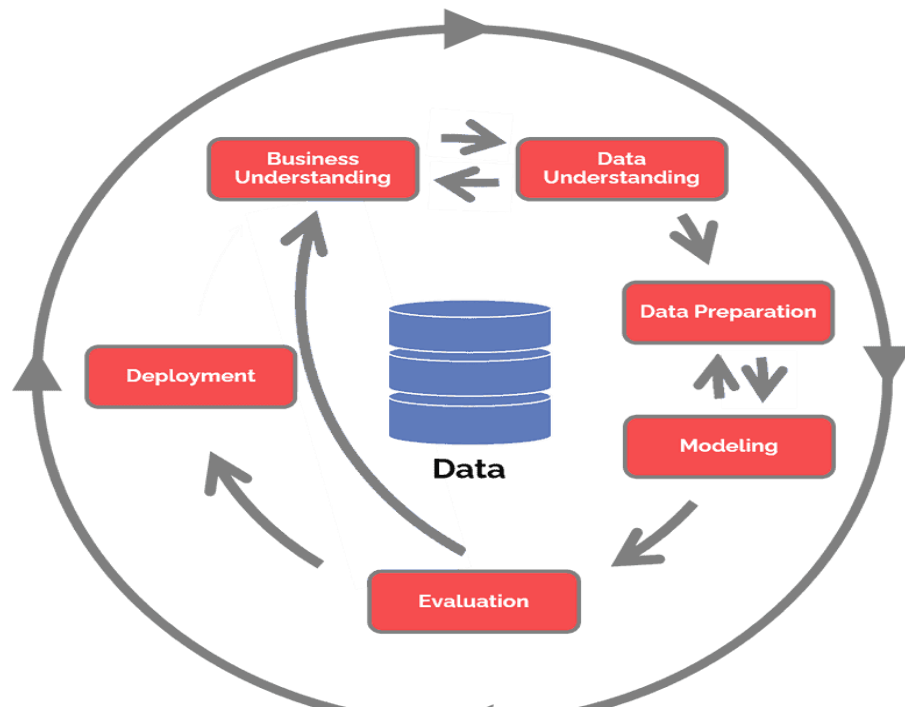
Otras estrategias?

Construcción y validación de modelos de ML o AA

Métodos, metodologías, workflows, pipelines, incorporan esencialmente tres fases asociadas al tratamiento de los datos para transformarlos en información y conocimiento para apoyar toma de decisiones. Se identifican como:

- Preprocesamiento
- Procesamiento
- Posprocesamiento

Construcción y validación de modelos de ML ^{KDD} o AA



1. Consolidate multiple data sources.
 - a. Real-time
 - b. Historian, e.g. OSIsoft PI
 - c. ERP
 - d. Laboratory sample data.
2. Manipulate into appropriate format.
3. Quality check.
4. Cleanse

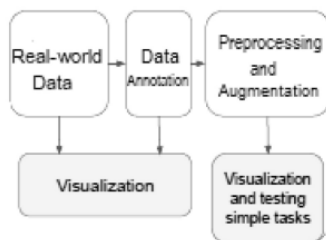
1. Visualisation
2. Collaborate with local subject matter experts.
3. Identify candidate anomalies.
4. Identify data gaps.

1. Where appropriate modify source data.
2. Gather new data and information.

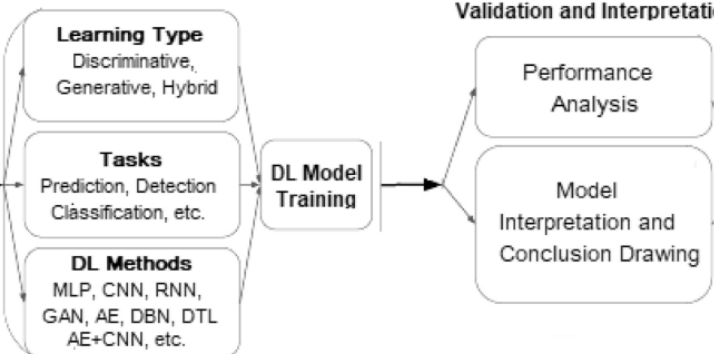
1. Automated Correlation.
2. Identify the valid early indicators of events.
3. Generate predictive model for deployment.
4. Review results with subject matter experts.

1. Automated ongoing monitoring of prediction accuracy.
2. Flag when model fine tuning may be required.

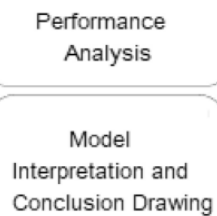
Step 1: Data Understanding and Preprocessing



Step 2: DL Model Building and Training



Step 3: Validation and Interpretation



From: [Deep Learning: A Comprehensive Overview on Techniques, Taxonomy, Applications and Research Directions](#)

A typical DL workflow to solve real-world problems, which consists of three sequential stages (i) data understanding and preprocessing (ii) DL model building and training (iii) validation and interpretation

Construcción y validación de modelos de ML o AA

- Preprocesamiento
- Procesamiento
 - Seleccionar algoritmos de aprendizaje / entrenamiento
 - Seleccionar Hiperparámetros
 - Varían según ML seleccionado y algoritmos de aprendizaje
 - Regularización, métodos para evitar sobreajuste
 - Penalización L1 y L2
 - Dropout
 - Aplicar métricas, Construir representaciones gráficas
 - En problemas de clasificación
 - En problemas de predicción
- Posprocesamiento
 - Análisis comparativo de métricas y representaciones gráficas

Métricas. Evaluación de la calidad de los modelos de ML

Problemas de Regresión

- Error cuadrático medio (MSE)
- Raíz Error cuadrático medio (RMSE)
- Error absoluto medio (MAE)

Problemas de Clasificación

Construir una matriz de confusión

Principales Métricas de clasificación

- Accuracy (promedio general de clasificación)
- Precisión
- Recall
- F1 Score

Métricas

Problemas de Regresión

- Error cuadrático medio (MSE)
- Raíz Error cuadrático medio (RMSE)
- Error absoluto medio (MAE)

$$MSE = \frac{1}{n} \sum_{i=1}^n \left(Y_i - \hat{Y}_i \right)^2.$$

Y_i is the observed value
 \hat{Y}_i is the predicted value
 n is the number of data points.

$$RMSE = \sqrt{MSE}$$

$$MAE = \frac{\sum_{i=1}^n |y_i - x_i|}{n} = \frac{\sum_{i=1}^n |e_i|}{n}.$$

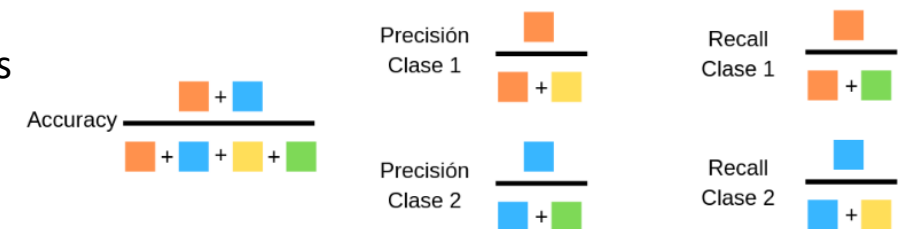
Métricas

- **Accuracy** del modelo, número total de predicciones correctas dividido el número total de predicciones.
- **Precisión** de una clase define cuan confiable es un modelo en responder si un punto pertenece a esa clase.
- **Recall** de una clase expresa cuan bien puede el modelo detectar a esa clase.
- **F1 Score** de una clase es dada por la media armónica de precisión y recall ($2 \times \text{precision} \times \text{recall} / (\text{precision} + \text{recall})$)

Casos posibles para cada clase:

- **alta precisión y alto recall:** el modelo maneja perfectamente esa clase
- **alta precisión y bajo recall:** el modelo no detecta la clase muy bien, si detecta la clase es altamente confiable.
- **baja precisión y alto recall:** detecta bien la clase, e incluye muestras de otras clases.
- **baja precisión y bajo recall:** El modelo no logra clasificar la clase correctamente.

	Predicción Clase 1	Predicción Clase 2
Valor real Clase 1	Aciertos True Positive Clase 1	Fallos False Positive Clase 2
Valor real Clase 2	Fallos False Positive Clase 1	Aciertos True Positive Clase 2



Nota. Si se dispone de un dataset desequilibrado, se podrá obtener **alto valor de PRECISION** en la clase Mayoritaria y **bajo valor RECALL** en la clase Minoritaria

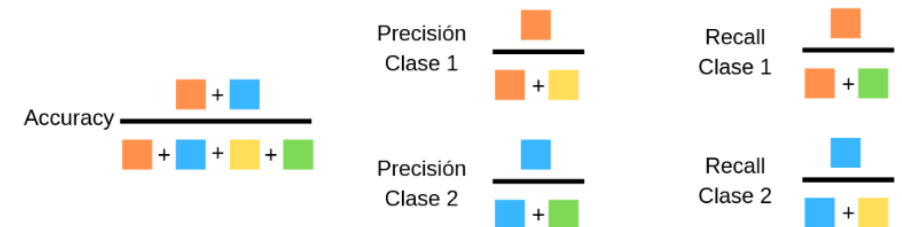
Métricas

Comparar métricas de entrenamiento vs. validación

- Dada la métrica Accuracy, si
- Acc-entrenamiento alta y Acc-validación alta -> Modelo generaliza !
- Acc-entrenamiento alta y Acc-validación baja -> sobreajuste (responde bien al entrenamiento)
- Acc-entrenamiento bajo y Acc-validación -> Modelo subajuste: requiere entrenar

Nota. Si se dispone de un dataset desequilibrado, se podrá obtener **alto valor de PRECISION** en la clase Mayoritaria y **bajo valor RECALL** en la clase Minoritaria

	Predicción Clase 1	Predicción Clase 2
Valor real Clase 1	Aciertos True Positive Clase 1	Fallos False Positive Clase 2
Valor real Clase 2	Fallos False Positive Clase 1	Aciertos True Positive Clase 2



Herramientas de programación

Python

Software



Python is a high-level, interpreted, general-purpose programming language. Its de... +



R



R is a programming language for statistical computing and graphics supported by the R Core Team and the R Foundation for Statistical Computing. Created by statisticians Ross Ihaka and Robert Gentleman...



PyTorch

Machine learning framework

<https://www.python.org/psf-landing/>

Recurros. Datasets

≡ kaggle

+ Create

Home

Competitions

Datasets

Models

Search

Datasets

Explore, analyze, and share quality data. [Learn more](#) about data types, creating, and collaborating.

+ New Dataset

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

Take the power of AI on the go with the free Copilot app
Create images, get help with writing, and search faster

No thanks [Get the Copilot app](#)

Microsoft | Research Our research Programs & events Connect & learn About Register: Research Forum All Microsoft Search

Researcher tools: code, datasets, & models

An index of datasets, SDKs, APIs and other open source code created by Microsoft researchers and shared with the broader academic community. We also maintain a [collection](#) highlighting some of the tools you'll find here.

Current Selections

Showing 1 – 10 of 1017 results

Sort by: Most recent

[Researcher tools: code, datasets, & models - Microsoft Research](#)

Registry of Open Data on AWS

The Registry of Open Data on AWS is now available on AWS Data Exchange
All datasets on the Registry of Open Data are now discoverable on AWS Data Exchange alongside 3,000+ existing data products from category-leading data providers across industries. Explore the catalog to find open, free, and commercial data sets. [Learn more about AWS Data Exchange](#)

[Explore the catalog](#)

About

This registry exists to help people discover and share datasets that are available via AWS resources. See recent additions and learn more about sharing data on AWS.

Get started using data quickly by viewing all tutorials with associated SageMaker Studio Lab notebooks.

See all usage examples for datasets listed in this registry.

See datasets from Allen Institute for Artificial Intelligence (AI2), Digital Earth Africa, Data for Good at Meta, NASA Space Act Agreement, NIH STRIDES, NOAA Open Data Dissemination Program, Space Telescope Science Institute, and Amazon Sustainability Data Initiative.

The Human Sleep Project

[bioinformatics](#) [deep learning](#) [life sciences](#) [machine learning](#) [medicine](#) [neurophysiology](#) [neuroscience](#)

The Human Sleep Project (HSP) sleep physiology dataset is a growing collection of clinical polysomnography (PSG) recordings. Beginning with PSG recordings from from ~15K patients evaluated at the Massachusetts General Hospital, the HSP will grow over the coming years to include data from >200K patients, as well as people evaluated outside of the clinical setting. This data is being used to develop CAISR (Complete AI Sleep Report), a collection of deep neural networks, rule-based algorithms, and signal processing approaches designed to provide better-than-human detection of conventional PSG...

[Details](#)

Usage examples

- The sleep and wake electroencephalogram over the lifespan. Neurobiol Aging. 2023 Jan 19;124:60-70. doi: 10.1016/j.neurobiolaging.2023.01.006. Epub ahead of print. PMID: 36739622. by Sun H, Ye E, Paixao L, Ganglberger W, Chu CJ, Zhang C, et al.
- Insomnia and morning motor vehicle accidents: A decision analysis of the risk of hypnotics versus the risk of untreated insomnia. Journal of Clinical Psychopharmacology. 2014 Jun;34(3):400-402. PMID: PMC6794095. by Bianchi MT, Westover MB.

Search datasets (currently 529 matching datasets)

Search datasets

Add to this registry

If you want to add a dataset or example of how to use a dataset to this

[Registry of Open Data on AWS](#)

Welcome to the UC Irvine Machine Learning Repository

We currently maintain 664 datasets as a service to the machine learning community. Here, you can donate and find datasets used by millions of people all around the world!

[VIEW DATASETS](#) [CONTRIBUTE A DATASET](#)

Popular Datasets

- Iris**
A small classic dataset from Fisher, 1936. One of the earliest known data...
Classification 150 Instances 4 Features
- Heart Disease**
4 databases: Cleveland, Hungary, Switzerland, and the VA Long Beach
Classification 303 Instances 13 Features
- Dry Bean**
Images of 13,611 grains of 7 different registered dry beans were taken w...
Classification 13.61K Instances 16 Features

New Datasets

- PhiUSIIL Phishing URL (Website)**
PhiUSIIL Phishing URL Dataset is a substantial dataset comprising 134,85...
Classification 235.8K Instances 54 Features
- RT-IoT2022**
The RT-IoT2022, a proprietary dataset derived from a real-time IoT infras...
Classification, Re... 123.12K Instances 84 Features
- Regensburg Pediatric Appendicitis**
This repository holds the data from a cohort of pediatric patients with su...
Classification 782 Instances 59 Features

[Home - UCI Machine Learning Repository](#)

Recursos. Datasets

[Datos Argentina](#)

datos.gob.ar

[Datasets](#) [Series](#) [Organizaciones](#) [APIs](#) [Acerca](#) ▾

Datos Argentina

Ponemos a tu alcance datos públicos en formatos abiertos para que puedas usarlos, modificarlos y compartirlos. Estos datos son tuyos. Podés crear visualizaciones, aplicaciones y grandes herramientas con ellos.

1231
DATASETS

040
ORGANIZACIONES
CON DATOS

¿Qué dataset buscás?



Agroganadería, pesca y forestación

Asuntos internacionales

Ciencia y tecnología

Economía y finanzas

Educación, cultura y deportes

Energía

Gobierno y sector público

Justicia, seguridad y legales

Medio ambiente

Población y sociedad

Regiones y ciudades

Salud

Transporte

GOBIERNO DE LA PROVINCIA DE BUENOS AIRES

[Datasets](#) [Sección Nueva](#) [Andino](#) [Organizaciones](#)

Datasets

Acá verás todos los conjuntos de datos que publicamos desde la Provincia de Buenos Aires.

100
DATASETS

Temas



Organizaciones

ARBA (2)
Banco Provincia (BAPRO) (2)
Dirección General de Cultura y Educación (6)
IDEBA (1)
Instituto Cultural de la Provincia de Buenos Aires (1)
Instituto de Previsión Social (2)
IOMA (15)
Junta Electoral (1)
Ministerio de Ambiente (2)
Ministerio de Desarrollo Agrario (2)
Ministerio de Economía (8)

¿Qué datasets buscás?



Ordenar por: Última modificación ▾

Rendimiento de establecimientos de salud

Ministerio de Salud de la Provincia de Buenos Aires. Subsecretaría de...

Datos correspondientes a los rendimientos de los establecimientos de salud provincial

xls csv pdf

Centros de salud

Ministerio de Salud de la Provincia de Buenos Aires.

Centros de testeo de salud. Incluye centros de Hepatitis Virales, centros de testeo de VIH y Sífilis, y centros que garantizan el Acceso ...

xls csv zip

Puertos públicos

Ministerio de Producción Ciencia e Innovación Tecnológica.

Datos correspondientes a los puertos públicos provinciales.

xls csv zip

Portal de Datos Abiertos (ciudaddecorrientes.gov.ar)

Municipalidad de
CORRIENTES

[Datasets](#) [Organizaciones](#) [Acerca](#) ▾

Portal de Datos Abiertos

El Portal de Datos Abiertos es una iniciativa ligada a las políticas de Gobierno Abierto que sirve para almacenar y compartir las bases de datos producidas por el Municipio para facilitar el conocimiento del gobierno, fortalecer el rendimiento de cuentas y mejorar la participación ciudadana.

¿Qué dataset buscás?



070
DATASETS

021
ORGANIZACIONES
CON DATOS

008
TEMAS

Ambiente

Desarrollo Urbano

Economía

Educación

Ejido urbano

Movilidad

Salud pública

Seguridad

<https://catalogo.datos.gba.gob.ar/dataset>

ML y modelos de RNA

ML. Redes Neuronales Artificiales

MLP (multi-perceptrón):

- Modelos con arquitecturas sencillas,
- Requerimientos computacionales: no son elevados y no es necesario el uso de GPUs.
- Uso de librerías, ej.: Scikit-learn

Deep learning:

- Modelos más complejos (redes convolucionales, redes recurrentes,...)
- Mayores requerimientos computacionales hacen necesario el uso de GPUs.
- Uso de frameworks especializados: Tensorflow-Keras o Pytorch.

Modelos de ML. Fases asociadas en aprendizaje y conocimiento. Ej. RNA

- Preprocesamiento
- Procesamiento
 - Algoritmos de entrenamiento (optimización)
 - En conjuntos de datos pequeños: l-bfgs
 - En conjuntos de datos grandes: Adam o Rmsprop
 - Hiperparámetros
 - Número de capas,
 - Número neuronas por capa, neuronas en capa de entrada, neurona en capa de salida,
 - Learning rate, ratio de aprendizaje establece cómo de rápido pueden cambiar los parámetros de un modelo a medida que se optimiza (aprende).
 - Otros hiperparámetros
 - Regularización, métodos para evitar sobreajuste
 - Penalización L1 y L2
 - Dropout
- Posprocesamiento

Fase 1. Preprocesamiento de los datos

- Detectar valores nulos / faltantes
 - Eliminar filas / columnas
 - Reemplazar (con aproximaciones): valores calculados / conocimiento del experto
- Detectar outlier
 - Outlier en una variable se puede detectar en gráficas
 - Opinión del experto sobre valor outlier
- Manejar datos nominales
 - Convertir texto a número.
 - Datos categóricos (Ej. Iris asignar valores: 0,1,2)
 - One hot (tantas etiquetas de salida como valores asuma la variable output, solo 1 activa por vez)
- Estandarizar y escalar datos*
 - Centrado: restar a cada valor la media del predictor al que pertenece. Todos los predictores asumen como valor media el cero, es decir, los valores se centran en torno al origen.
 - Normalizar (estandarizar) los datos. Cada característica debe poseer su rango de datos. Transformar los datos de forma que todos los predictores estén aproximadamente en la misma escala
 - Normalización Z-score (StandardScaler): dividir cada predictor entre su desviación típica después de haber sido centrado, de esta forma, los datos pasan a tener una distribución normal.
 - Estandarización max-min (MinMaxScaler): transformar los datos de forma que estén dentro del rango [0, 1].

* Si los predictores son numéricos, la escala en la que se miden y su varianza pueden influir en el comportamiento del modelo. Si no se igualan de alguna forma los predictores, aquellos que se midan en una escala mayor o que tengan más varianza dominarán el modelo aunque no sean los que más relación tienen con la variable respuesta

Fase 2. Procesamiento de modelos

- Construir, entrenar y validar distintos modelos, con diferentes combinaciones de hiperparámetros.
- Separar en conjuntos de entrenamiento, validación y testeo
- Aplicar validación cruzada, probar con distintos valores
- Según herramienta de programación los hiperparámetros pueden variar.
- Seleccionar el optimizador (Solvers). Ej en RNA: 'l-bfgs', 'sgd', 'adam',

Tipos de solucionadores

Estocástico, actualizan los parámetros para cada punto de datos.

Lote, actualizan los parámetros después de procesar un lote de punto de datos.

Fase 3. Posprocesamiento

Análisis de resultados

- Aplicar métricas definidas para
 - problemas de clasificación
 - problemas de predicción
- Realizar estudios comparativos, analizar resultados. Argumentar las decisiones

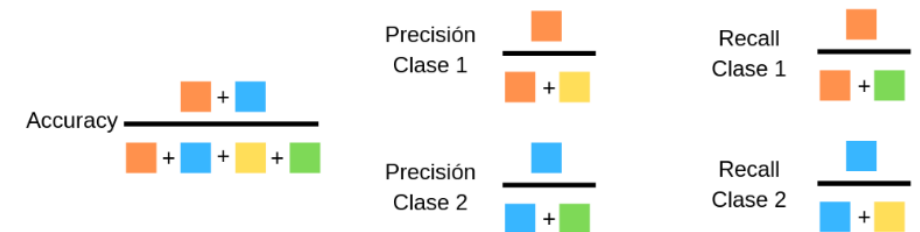
Métricas. Problemas de clasificación

- Accuracy del modelo, numero total de predicciones correctas dividido por el número total de predicciones.
- Precisión de una clase define cuan confiable es un modelo en responder si un punto pertenece a esa clase.
- Recall de una clase expresa cuan bien puede el modelo detectar a esa clase.
- F1 Score de una clase es dada por la media harmonica de precisión y recall ($2 \times \text{precision} \times \text{recall} / (\text{precision} + \text{recall})$)

Casos posibles para cada clase:

- **alta precisión y alto recall:** el modelo maneja perfectamente esa clase
- **alta precisión y bajo recall:** el modelo no detecta la clase muy bien, pero cuando lo hace es altamente confiable.
- **baja precisión y alto recall:** La clase detecta bien la clase, e incluye muestras de otras clases.
- **baja precisión y bajo recall:** El modelo no logra clasificar la clase correctamente.

	Predicción Clase 1	Predicción Clase 2
Valor real Clase 1	Aciertos True Positive Clase 1	Fallos False Positive Clase 2
Valor real Clase 2	Fallos False Positive Clase 1	Aciertos True Positive Clase 2



Nota. Si se dispone de un dataset desequilibrado, se podrá obtener **alto valor de PRECISION en la clase Mayoritaria y bajo valor RECALL en la clase Minoritaria**

Práctica con software: Phytton.

MLPClassifier

[sklearn.neural_network.MLPClassifier](#)

[MLPClassifier — scikit-learn 1.5.0 documentation](#)

```
class sklearn.neural_network.MLPClassifier(hidden_layer_sizes=(100,),  
activation='relu', *, solver='adam', alpha=0.0001, batch_size='auto',  
learning_rate='constant', learning_rate_init=0.001, power_t=0.5, max_iter=200,  
shuffle=True, random_state=None, tol=0.0001, verbose=False, warm_start=False,  
momentum=0.9, nesterovs_momentum=True, early_stopping=False, validation_fraction=0.1,  
beta_1=0.9, beta_2=0.999, epsilon=1e-08, n_iter_no_change=10, max_fun=15000) [source]
```


Práctico 5. Ejercicio 2

- Algoritmos de entrenamiento (solvers) en RNA
 - 'l-bfgs',
 - 'sgd',
 - 'adam',
 - otros ?
- ¿Qué significan estas siglas ?
- ¿En qué conjuntos de datos se aplican ?
- ¿Qué tipos de solvers son?
- Mencionar algunas características de los solvers,

Práctico 5. Ejercicio 3

- Algoritmos de entrenamiento (solvers) en RNA
 - l-bfg',
 - 'sgd',
 - 'adam',
 - Otros ?
- Construye tres modelos distintos, modificando el solver utilizado
- Entrena y valida los modelos
- Aplica métricas y analiza los resultados.

Práctico 5. Ejercicio 4

- Sea uno de los modelos de RNA entrenados y validados. Si se aplican dos configuraciones distintas de validación cruzada ($m = 10$ fols, $m = 5$ fols). Las métricas de evaluación, ¿presentan cambios ?

Caso de análisis. Ejemplo

Caso de análisis


Dataset o conjunto de datos Iris

Largo de sépalos	Ancho de sépalos	Largo de pétalos	Ancho de pétalos	Especies
5.1	3.5	1.4	0.2	<i>I. setosa</i>
4.9	3.0	1.4	0.2	<i>I. setosa</i>
4.7	3.2	1.3	0.2	<i>I. setosa</i>
4.6	3.1	1.5	0.2	<i>I. setosa</i>
5.0	3.6	1.4	0.2	<i>I. setosa</i>
5.4	3.9	1.7	0.4	<i>I. setosa</i>
4.6	3.4	1.4	0.3	<i>I. setosa</i>
5.0	3.4	1.5	0.2	<i>I. setosa</i>
4.4	2.9	1.4	0.2	<i>I. setosa</i>
4.9	3.1	1.5	0.1	<i>I. setosa</i>
5.4	3.7	1.5	0.2	<i>I. setosa</i>
4.8	3.4	1.6	0.2	<i>I. setosa</i>
4.8	3.0	1.4	0.1	<i>I. setosa</i>
4.3	3.0	1.1	0.1	<i>I. setosa</i>
5.8	4.0	1.2	0.2	<i>I. setosa</i>
5.7	4.4	1.5	0.4	<i>I. setosa</i>
5.4	3.9	1.3	0.4	<i>I. setosa</i>
5.1	3.5	1.4	0.3	<i>I. setosa</i>
5.7	3.8	1.7	0.3	<i>I. setosa</i>
5.1	3.8	1.5	0.3	<i>I. setosa</i>
5.4	3.4	1.7	0.2	<i>I. setosa</i>

contiene 50 muestras de cada una de tres especies de *Iris* ([*Iris setosa*](#), [*Iris virginica*](#) e [*Iris versicolor*](#)).

Se midieron (en centímetros) cuatro rasgos de cada muestra: el largo y ancho del [sépalos](#) y del [pétalos](#).

- Basado en la combinación de estos cuatro rasgos (evidencias), Fisher desarrolló un modelo discriminante lineal para distinguir entre una especie y otra.

 Iris Donated on 6/30/1988		
A small classic dataset from Fisher, 1936. One of the earliest known datasets used for evaluating classification methods.		
Dataset Characteristics	Subject Area	Associated Tasks
Tabular	Biology	Classification
Feature Type	# Instances	# Features
Real	150	4

R. Fisher. "Iris," UCI Machine Learning Repository, 1936. [Online]. Available: <https://doi.org/10.24432/C56C76>.

[Iris - UCI Machine Learning Repository](#)

Comprensión
de datos



Preprocesamiento



Procesamiento
(Modelado)



Posprocesamiento
Evaluación



Despliegue

Datos entrenamiento

Largo de sépalos	Ancho de sépalos	Largo de pétalo	Ancho de pétalo	Especies	Specie1	Specie 2	Specie 3
4.6	3.2	1.4	0.2	I. setosa	1	0	0
5.3	3.7	1.5	0.2	I. setosa	1	0	0
5	3.3	1.4	0.2	I. setosa	1	0	0
7	3.2	4.7	1.4	I. versicolor	0	1	0
6.4	3.2	4.5	1.5	I. versicolor	0	1	0
6.9	3.1	4.9	1.5	I. versicolor	0	1	0
5.5	2.3	4	1.3	I. versicolor	0	1	0
6.3	2.5	5	1.9	I. virginica	0	0	1
6.5	3	5.2	2	I. virginica	0	0	1
6.2	3.4	5.4	2.3	I. virginica	0	0	1
5.9	3	5.1	1.8	I. virginica	0	0	1

Algoritmo Aprendizaje

- Modelo predictivo
- Modelo clasificación

Backpropagation
Descenso gradiente

....

Error (optimizar)

Métricas:

- Modelo predictivo
- Modelo clasificación
- Otras

Algoritmos, hiperparámetros
varían según tecnología IA

Datos validación / Nuevos datos

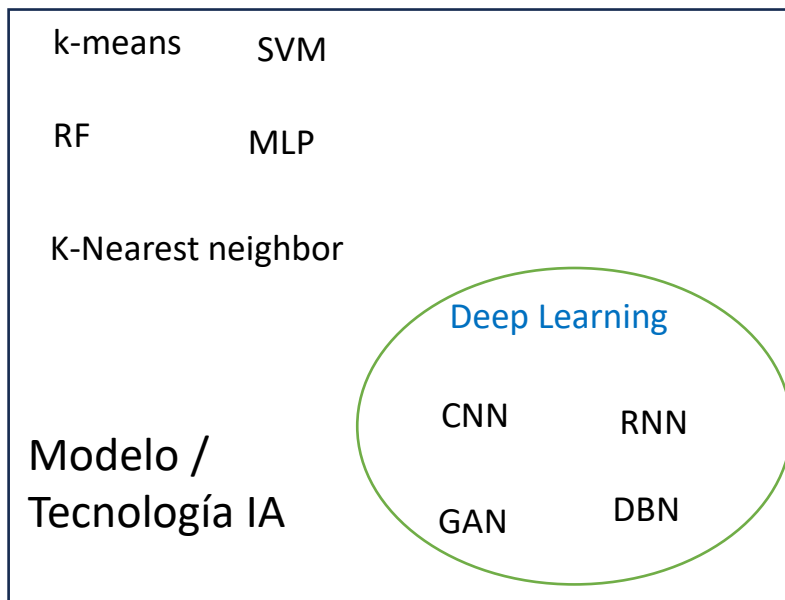
Largo de sépalos	Ancho de sépalos	Largo de pétalo	Ancho de pétalo
4.6	3.2	1.4	0.2
5.3	3.7	1.5	0.2
5	3.3	1.4	0.2
7	3.2	4.7	1.4
6.4	3.2	4.5	1.5
6.9	3.1	4.9	1.5
5.5	2.3	4	1.3
6.3	2.5	5	1.9
6.5	3	5.2	2
6.2	3.4	5.4	2.3
5.9	3	5.1	1.8



Modelo entrenado



Inferencias



Configurar el modelo, Entrenar el modelo, ajuste de parámetros mediante la optimización iterativa. Evitar el sobreajuste / subajuste

Comprensión
de datos



Preprocesamiento



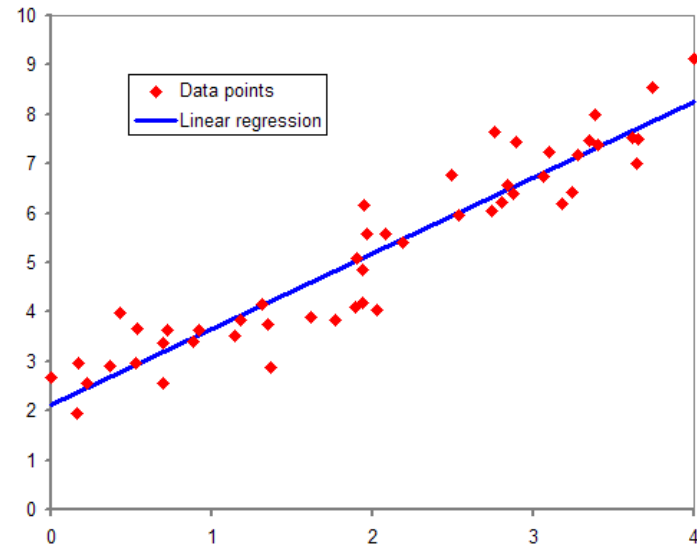
Procesamiento
(Modelado)



Posprocesamiento
Evaluación

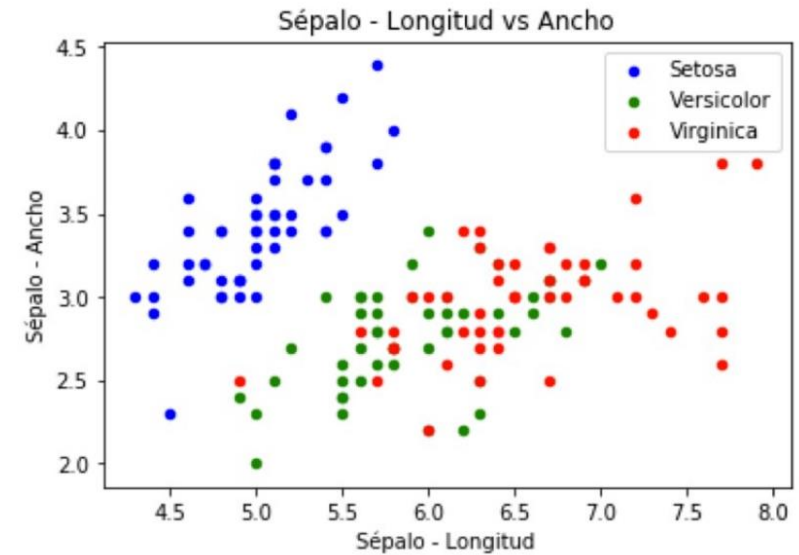


Despliegue



Predecir un valor real

- múltiples variables de entrada, regresión multivariante.
- Variables se ordenan por tiempo, pronóstico de series de tiempo.



Clasificación

- Binaria
- Multiclase

Comprensión
de datos



Preprocesamiento



Procesamiento
(Modelado)



Posprocesamiento
Evaluación



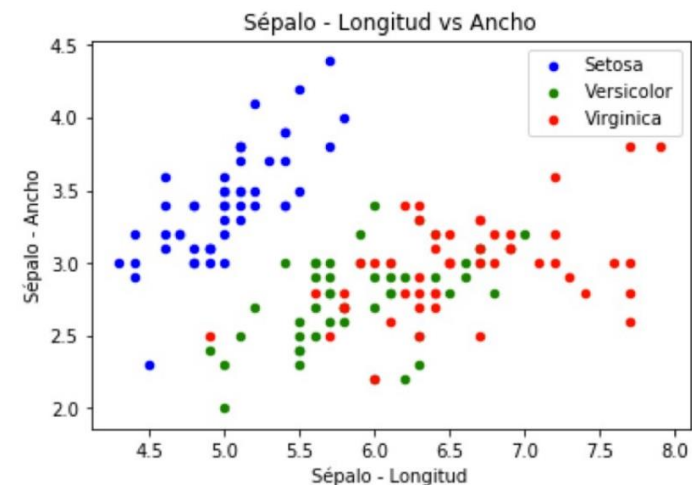
Despliegue

Disponer de fuentes de datos

Archivos

- Fila: registro o ejemplo
- Columna: atributo, variable, característica
- Datos numéricos o nominales
- ML requiere datos numericos

Largo de sépalos	Ancho de sépalos	Largo de pétalo	Ancho de pétalo	Especies
5.1	3.5	1.4	0.2	I. setosa
4.9	3	1.4	0.2	I. setosa
4.7	3.2	1.3	0.2	I. setosa
4.6	3.1	1.5	0.2	I. setosa
5	3.6	1.4	0.2	I. setosa
5.4	3.9	1.7	0.4	I. setosa
4.6	3.4	1.4	0.3	I. setosa
5	3.4	1.5	0.2	I. setosa
4.4	2.9	1.4	0.2	I. setosa



Ap Supervisado
Clasificación multiclase

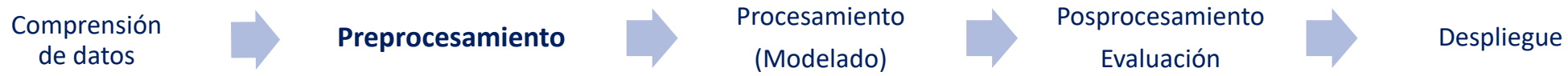
Datos nominales a categórico

Largo de sépalos	Ancho de sépalos	Largo de pétalo	Ancho de pétalo	Especies	Specie
5.1	3.5	1.4	0.2	I. setosa	0
4.9	3	1.4	0.2	I. setosa	0
4.7	3.2	1.3	0.2	I. setosa	0
4.6	3.1	1.5	0.2	I. setosa	0
5	3.6	1.4	0.2	I. setosa	0
5.4	3.9	1.7	0.4	I. setosa	0
4.6	3.4	1.4	0.3	I. setosa	0
5	3.4	1.5	0.2	I. setosa	0

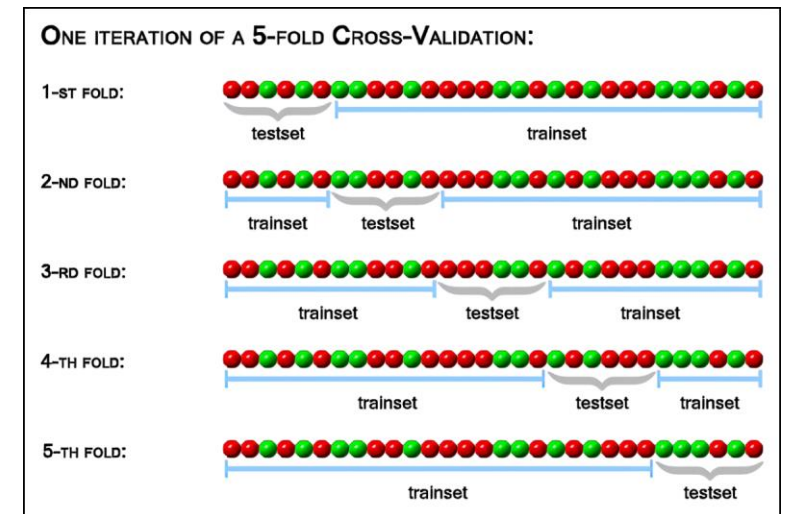
Specie: 0 setosa; 1 versicolor; 2 virginica

Largo de sépalos	Ancho de sépalos	Largo de pétalo	Ancho de pétalo	Especies	Specie1	Specie 2	Specie 3
4.6	3.2	1.4	0.2	I. setosa	1	0	0
5.3	3.7	1.5	0.2	I. setosa	1	0	0
5	3.3	1.4	0.2	I. setosa	1	0	0
7	3.2	4.7	1.4	I. versicolor	0	1	0
6.4	3.2	4.5	1.5	I. versicolor	0	1	0
6.9	3.1	4.9	1.5	I. versicolor	0	1	0
5.5	2.3	4	1.3	I. versicolor	0	1	0
6.3	2.5	5	1.9	I. virginica	0	0	1
6.5	3	5.2	2	I. virginica	0	0	1
6.2	3.4	5.4	2.3	I. virginica	0	0	1
5.9	3	5.1	1.8	I. virginica	0	0	1

Datos nominales a One-hot



- Manejar datos nominales
 - Convertir texto a número.
 - Datos categóricos (Ej. Iris asignar valores: 0,1,2)
 - One hot (tantas etiquetas de salida como valores asuma la variable output, solo 1 activa por vez)
- Detectar valores nulos / faltantes
 - Eliminar filas / columnas
 - Reemplazar (con aproximaciones): valores calculados (media, medana, moda) / conocimiento del experto
- Detectar outlier
 - Outlier, variable detectable en gráfica
 - Opinión del experto sobre valor outlier
- Normalizar datos
 - Cada característica debe poseer su rango de datos
- Dividir los datos en conjuntos de entrenamiento, validación y prueba.



Comprensión
de datos



Preprocesamiento



Procesamiento
(Modelado)



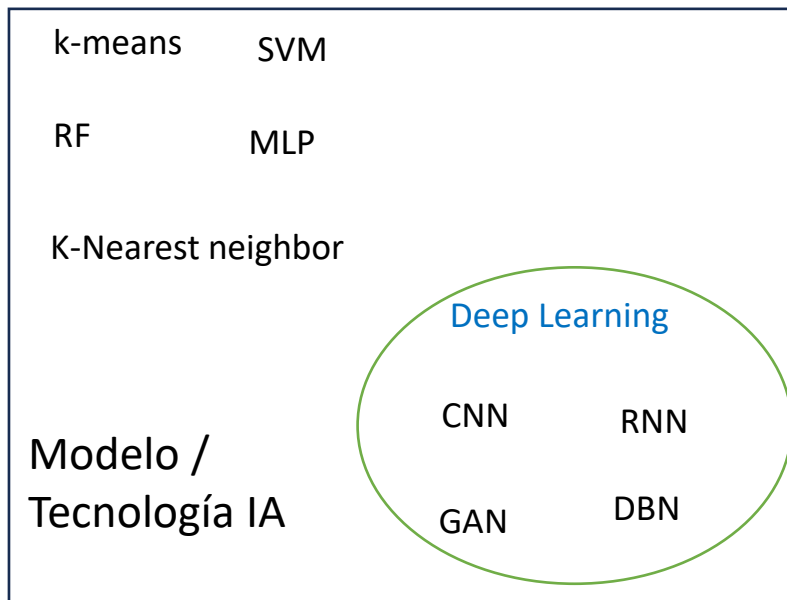
Posprocesamiento
Evaluación



Despliegue

Datos entrenamiento

Largo de sépalos	Ancho de sépalos	Largo de pétalo	Ancho de pétalo	Specie1	Specie 2	Specie 3
4.6	3.2	1.4	0.2	1	0	0
5.3	3.7	1.5	0.2	1	0	0
5	3.3	1.4	0.2	1	0	0
7	3.2	4.7	1.4	0	1	0
6.4	3.2	4.5	1.5	0	1	0
6.9	3.1	4.9	1.5	0	1	0
5.5	2.3	4	1.3	0	1	0
6.3	2.5	5	1.9	0	0	1
6.5	3	5.2	2	0	0	1
6.2	3.4	5.4	2.3	0	0	1
5.9	3	5.1	1.8	0	0	1



Algoritmo Aprendizaje
RNA

Backpropagation
Descenso gradiente

....

Funciones activacion /
transferencia

hiperparámetros

Nro capas / neuronas
F. activación(transferencia)
Learning rate (alfa)
otras

Métricas:

- Modelo predictivo
- Modelo clasificación



Datos validación / Nuevos datos

Largo de sépalos	Ancho de sépalos	Largo de pétalo	Ancho de pétalo
4.6	3.2	1.4	0.2
5.3	3.7	1.5	0.2
5	3.3	1.4	0.2
7	3.2	4.7	1.4
6.4	3.2	4.5	1.5
6.9	3.1	4.9	1.5
5.5	2.3	4	1.3
6.3	2.5	5	1.9
6.5	3	5.2	2
6.2	3.4	5.4	2.3
5.9	3	5.1	1.8



Modelo entrenado



Inferencias

Configurar el modelo, Entrenar el modelo, ajuste de parámetros mediante la optimización iterativa. Evitar el sobreajuste / subajuste

Comprensión
de datos



Preprocesamiento



Procesamiento
(Modelado)



Posprocesamiento
Evaluación



Despliegue

Datos entrenamiento

Largo de sépalos	Ancho de sépalos	Largo de pétalo	Ancho de pétalo	Specie1	Specie 2	Specie 3
4.6	3.2	1.4	0.2	1	0	0
5.3	3.7	1.5	0.2	1	0	0
5	3.3	1.4	0.2	1	0	0
7	3.2	4.7	1.4	0	1	0
6.4	3.2	4.5	1.5	0	1	0
6.9	3.1	4.9	1.5	0	1	0
5.5	2.3	4	1.3	0	1	0
6.3	2.5	5	1.9	0	0	1
6.5	3	5.2	2	0	0	1
6.2	3.4	5.4	2.3	0	0	1
5.9	3	5.1	1.8	0	0	1



Algoritmo Aprendizaje
RNA

Backpropagation
Descenso gradiente

....

Funciones activación /
transferencia

Hiperparámetros en RNA
Nro capas / neuronas
F. activación(transferencia)
Learning rate (alfa)
otras

Métricas:

- Modelo predictivo
- Modelo clasificación



Datos validación / Nuevos datos

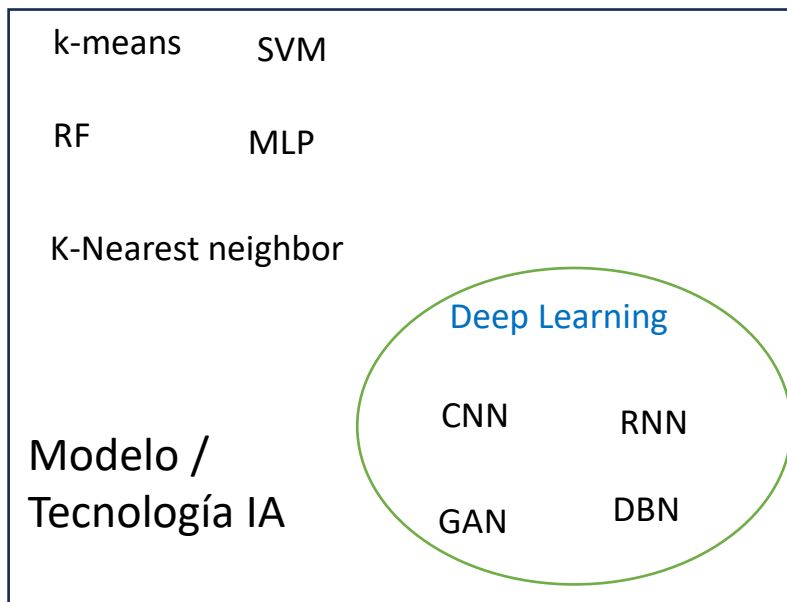
Largo de sépalos	Ancho de sépalos	Largo de pétalo	Ancho de pétalo
4.6	3.2	1.4	0.2
5.3	3.7	1.5	0.2
5	3.3	1.4	0.2
7	3.2	4.7	1.4
6.4	3.2	4.5	1.5
6.9	3.1	4.9	1.5
5.5	2.3	4	1.3
6.3	2.5	5	1.9
6.5	3	5.2	2
6.2	3.4	5.4	2.3
5.9	3	5.1	1.8



Modelo entrenado



Inferencias



Validar resultados. Aplicación de metricas [RMS, MAE / precisión, recall, accuracy, F1-score]. Ajuste si corresponde. Enfatizar la interpretación.

3. Posprocesamiento

Análisis de resultados. Ejemplos

Tabla 5. Resultados utilizando Árboles de Decisión en la base de datos *Acceptors*

Algoritmo	Exactitud	Área bajo la curva ROC	rVP	rVN	
AlternatingDecisionTree (ADTree)		0.711	0.931	0.591	0.96
DecisionStump	0	0.726	0	1	
Id3	0.492	0.73	0.555	0.904	
J48	0.591	0.727	0.548	0.937	
RandomForest					
(10 trees, 8 random features)	0.771	0.847	0.111	0.995	
RandomForest					
(5 trees, 8 random features)	0.613	0.785	0.241	0.975	
RandomTree	0.219	0.544	0.219	0.87	
REPTree	0.674	0.877	0.565	0.955	
SimpleCart	0.687	0.878	0.559	0.958	

Tabla 6. Resultados utilizando Árboles de Decisión en la base de datos *Donors*

Algoritmo	Exactitud	Área bajo la curva ROC	rVP	rVN
AlternatingDecisionTree (ADTree)	0.827	0.967	0.683	0.976
DecisionStump	0	0.706	0	1
Id3	0.695	0.841	0.735	0.946
J48	0.759	0.842	0.745	0.961
RandomForest				
(10 trees, 8 random features)	0.823	0.834	0.116	0.996
RandomForest				
(5 trees, 8 random features)	0.563	0.759	0.211	0.973
RandomTree	0.179	0.522	0.183	0.861
REPTree	0.768	0.929	0.771	0.961
SimpleCart	0.794	0.932	0.763	0.967

El algoritmo *ADTree* resultó el mejor método para la base de datos *Acceptors* según el área bajo la curva ROC y la razón de verdaderos positivos, mientras que en la de *Donors* fue por la exactitud y el área bajo la curva. La mayor razón de verdaderos positivos la obtuvo el método *REPTree* en ambas bases. Los clasificadores basados en árboles de decisión no brindaron resultados significativos puesto que los parámetros medidos fueron bajos.

Fuente: Díaz-Barrios et al. Algoritmos de aprendizaje automático para clasificación de Splice Sites en secuencias genómicas

Ejemplos

Tabla 3. Métricas de desempeño para cada modelo de CNN

Modelo (mejor época/total épocas)	MÉTRICAS DE DESEMPEÑO						
	Precisión	Sensibilidad	Especificidad	F1-score	G_mean	IBA	Tiempo (min)
Wide_resnet101_2 (26/50)	97.75	97.75	96.76	97.75	97.25	94.66	61.45
Resnext101_32x8d (21/50)	97.75	97.75	96.40	97.75	97.06	94.34	76.98
Resnext50_32x4d (10/50)	97.75	97.75	96.07	97.75	96.89	94.04	36.10
Inception_V3 (47/50)	97.69	97.67	97.16	97.67	97.41	94.94	113.28
Mnasnet1_0 (49/50)	97.58	97.50	97.03	97.52	97.26	94.63	95.22
Wide_resnet50_2 (44/50)	97.50	97.50	96.16	97.50	96.82	93.86	111.27
Mobilenet_v2 (9/50)	97.42	97.42	95.64	97.41	96.50	93.29	26.97
Shufflenet_v2_x0_5 (48/50)	95.70	95.67	93.57	95.67	94.57	89.62	83.00

Los resultados de desempeño de los modelos seleccionados son enlistados en la Tabla 3. Cabe resaltar, que todos los modelos fueron entrenados usando 50 épocas de entrenamiento, sin embargo, solo se reporta la época con mejor desempeño de precisión para cada modelo. Además, el tiempo tomado por cada modelo para llegar a la época con mejor desempeño también es reportado.

De acuerdo con los resultados reportados en la Tabla 3, los primeros siete modelos superan el 90% en la métrica IBA y 97% en precisión, lo que nos indica que obtuvieron un excelente desempeño de predicción en todas sus clases con un alto grado de clasificación en cada clase.

Ahora bien, con el fin de determinar cuál es la mejor arquitectura para clasificar imágenes del COVID-19 se requiere de otra métrica, por lo cual utilizaremos la métrica de precisión. En la Fig. 5 se muestra los valores de precisión obtenidos por cada modelo de CNN en función del tiempo de entrenamiento con las imágenes de la base de datos para prueba.

De acuerdo con los resultados presentados en la Tabla 3 y en la Fig. 5, el valor de precisión más alto fue del 97.75%, el cual fue obtenido por tres arquitecturas: Wide_resnet101_2, Resnext101_32x8d y Resnext50_32x4d.

Referencias

W. S. McCulloch y W. Pitts, A logical calculus of the ideas immanent in nervous activity. *The bulletin of mathematical biophysics*, 5(4), 115-133. 1943, <https://doi.org/10.1007/bf02459570>

M. Minsky, S. Papert, Seymour, Perceptrons: An Introduction to Computational Geometry. MIT Press. 1969 ISBN 0-262-63022-2. [Perceptrons; an Introduction to Computational Geometry - Marvin Minsky, Seymour Papert - Google Libros](#)

S. J. Russell y P. Norvig Inteligencia Artificial: Un Enfoque Moderno, 3ra Edición. Prentice Hall, USA, 2009

S. Haykin, Neural Networks and Learning Machines, 3ra Edición, Prentice Hall, USA, 2009.

S. I. Gallant, "Perceptron-based learning algorithms," in *IEEE Transactions on Neural Networks*, vol. 1, no. 2, pp. 179-191, June 1990, doi: 10.1109/72.80230.

Publicaciones de la temática