

Introducción a la Inteligencia Artificial
Facultad de Ingeniería
Universidad de Buenos Aires



Índice

1. Terminology
2. Pipeline
3. Train-test-validation
4. Feature engineering
5. Regresión lineal

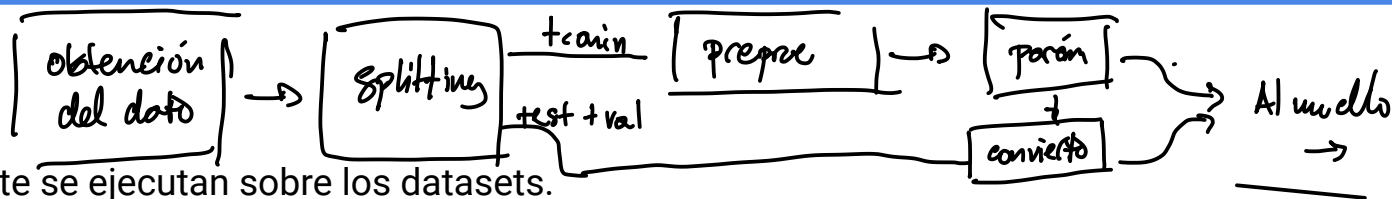


Machine Learning Terminology

- ^{preproc?}
- Raw vs. Tidy Data → **crudos (raw data): sin procesar / tidy: son datos preprocesados**
- Training vs. Holdout Sets → **train, test, holdout / validation | dev (Deep learning)**
- Baseline → (ref. al modelo) refiere al modelo de partida (Modelo sencillo). Es nuestro modelo a ganar
- Parameters vs. Hyperparameters → **parámetros son los "stidors" del pipeline | hp. son los "stidors" del modelo.**
- Classification vs. Regression → **diferenciando la naturaleza de la variable de salida** → $y \in K \rightarrow$ clasif
↳ $y \in \mathbb{R} \rightarrow$ regre.
- Model-Based vs. Instance-Based Learning → **enfogue modelocentrico - enfogue datacentrico.**
- Shallow vs. Deep Learning → **clásico vs deep learning**
+ clásico vs transfer learning.
+ Deep learning
- transfer learning → **Enfogue (Actual) de trabajo con modelos MUY profundos y complejos que entrenados**
- (Active) task specific training → **Ajornar un modelo de trans. learning y especificarlo.**
- Zero-shot → **posarlo por mi tarea.**
- LLM → **large language models. (+ 1MM)**
- bAGI → **baby Artificial General Intelligence.**
- Embedding (latent space, dense space)



Dataset pipeline



Obtención de datos
o synthetic dataset

Pre-procesamiento
de Missing Values

Cómputo de media,
desvío y cuantiles

Estandarización de
datos (z-score)

Ingeniería de
Features (PCA)

Data
augmentation

Split en Train,
Validation y Test

① ↖

↗ ②

Model pipeline

Pasos involucrados al entrenar un modelo de Machine Learning

Obtener el dataset
para train

Definir métricas de
evaluación y train

Calcular métricas
para modelos base

Entrenar el modelo
con el dataset train

Computar métricas
con validation

HPs
optimization

Evaluación sobre
el dataset test

no debs:

- baseline

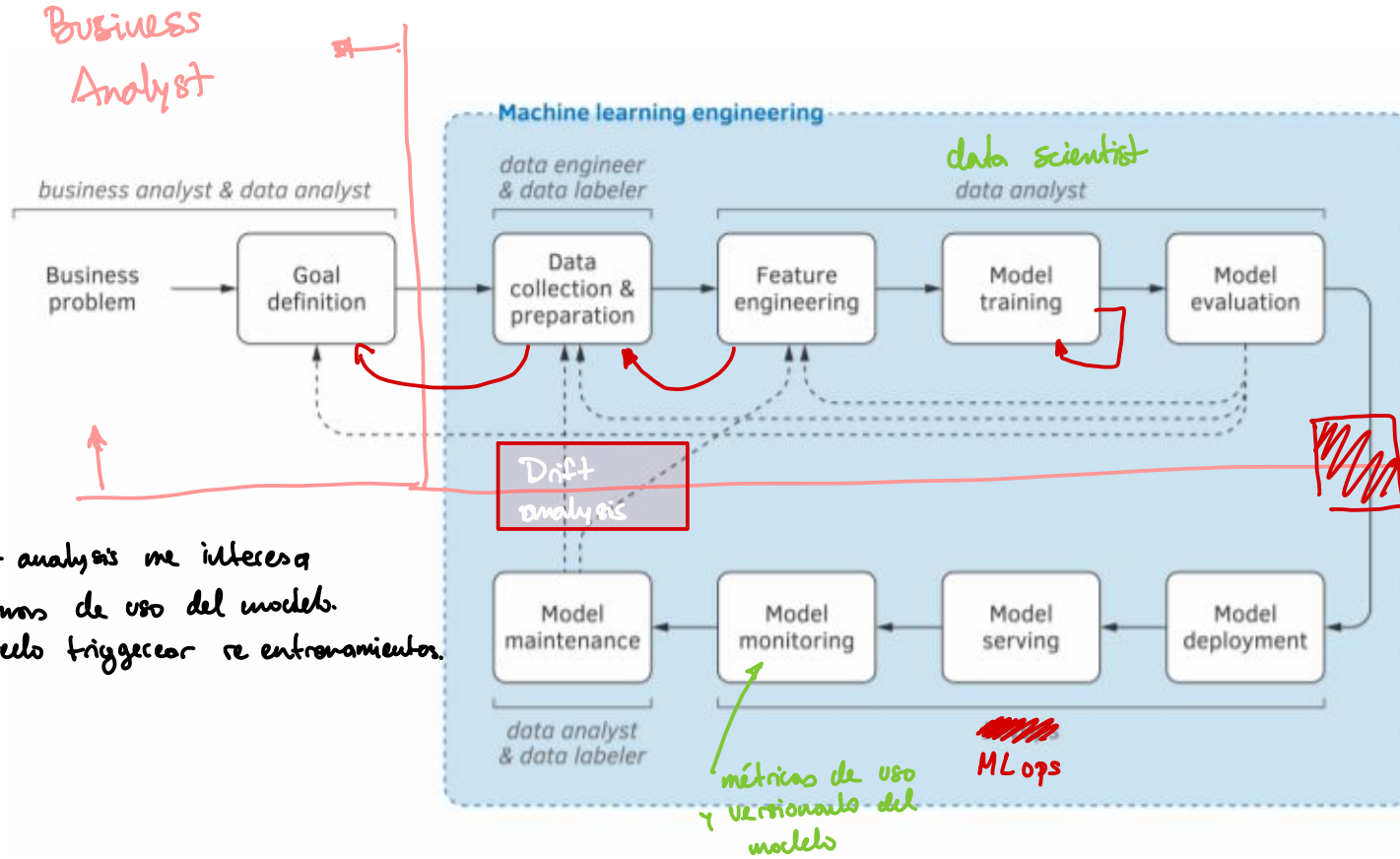
- mc1 → 1+2

- mc2 → 1+7

- mc3 → 1+3



Machine Learning Pipeline



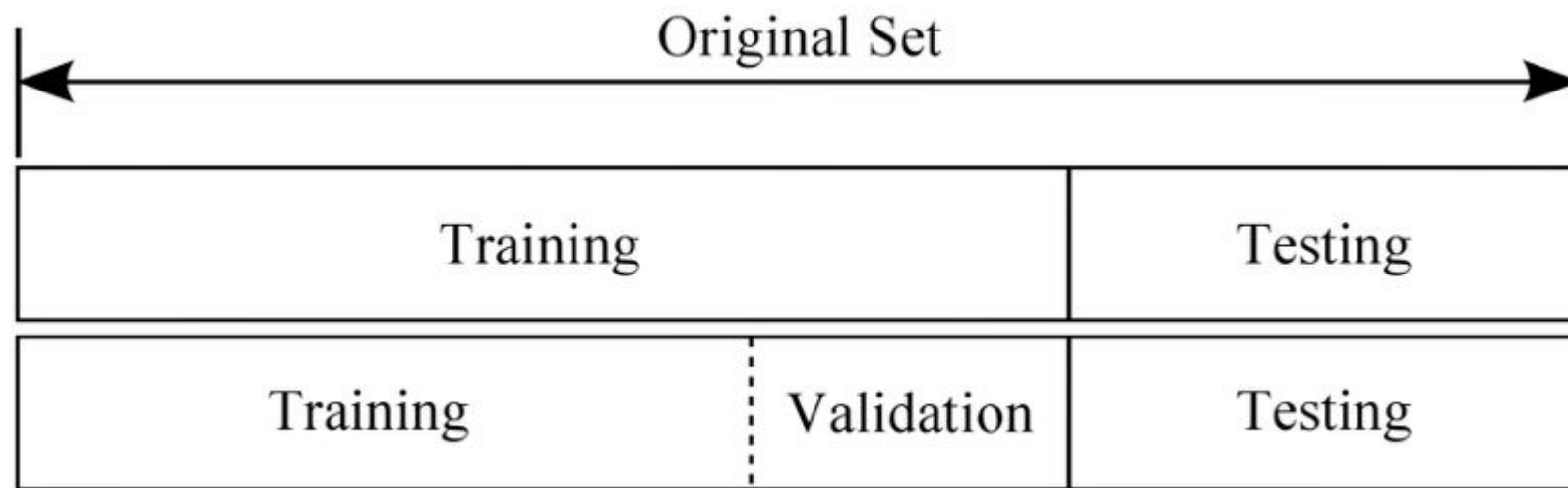
Se mueve
en notebooks
↓
Desarrolla
productividad
↓
productivo
↓
Se mueve
en apps



Ingeniería de Features

Train - test - validation

train es para calular ϕ 's de los pipelines.
test " " validar ϕ 's " " "



train-test: 80-20, 70-30, 90-10

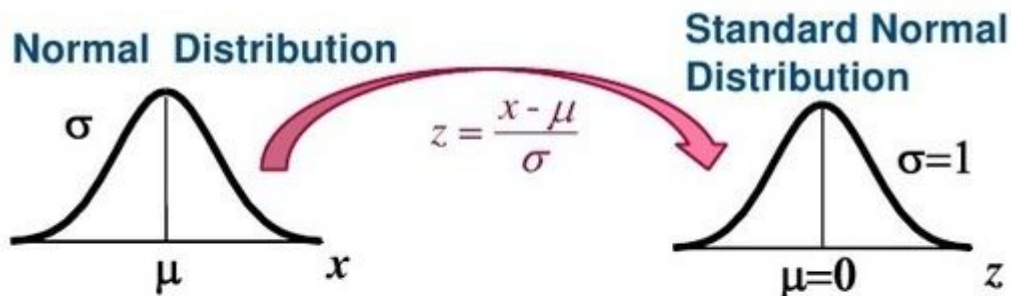
train, test, val: 70-15-15, 80-15-5, 70-25-5, 90-55
20-10

¿cómo separamos?

buscar muestrear de
manera de generar muestras
representativas.

Normalización

Muchos algoritmos de Machine Learning necesitan datos de entrada centrados y normalizados. Una normalización habitual es el z-score, que implica restarle la media y dividir por el desvío a cada feature de mi dataset.



Missing Values

Es muy común en la práctica, recibir como datos de entrada, datasets que tienen información incompleta ("NaN").

ID	City	Degree	Age	Salary	Married ?
1	Lisbon	NaN	25	45,000	0
2	Berlin	Bachelor	25	NaN	1
3	Lisbon	NaN	30	NaN	1
4	Lisbon	Bachelor	30	NaN	1
5	Berlin	Bachelor	18	NaN	0
6	Lisbon	Bachelor	NaN	NaN	0
7	Berlin	Masters	30	NaN	1
8	Berlin	No Degree	NaN	NaN	0
9	Berlin	Masters	25	NaN	1
10	Madrid	Masters	25	NaN	1



Solución 1

Una forma de solucionar el problema es remover las filas y las columnas que contienen dichos valores.

ID	City	Degree	Age	Salary	Married ?
1	Lisbon	NaN	25	45,000	0
2	Berlin	Bachelor	25	NaN	1
3	Lisbon	NaN	30	NaN	1
4	Lisbon	Bachelor	30	NaN	1
5	Berlin	Bachelor	18	NaN	0
6	Lisbon	Bachelor	NaN	NaN	0
7	Berlin	Masters	30	NaN	1
8	Berlin	No Degree	NaN	NaN	0
9	Berlin	Masters	25	NaN	1
10	Madrid	Masters	25	NaN	1

¿Filas luego columnas
ó
Columnas luego filas?



Solución 2

En columnas donde el % de NaNs es relativamente bajo, es aceptable reemplazar los NaNs por la media o mediana de la columna.

Average_Age = 26.0

ID	City	Age	Married ?
1	Lisbon	25	0
2	Berlin	25	1
3	Lisbon	30	1
4	Lisbon	30	1
5	Berlin	18	0
6	Lisbon	NaN	0
7	Berlin	30	1
8	Berlin	NaN	0
9	Berlin	25	1
10	Madrid	25	1



ID	City	Age	Married ?
1	Lisbon	25	0
2	Berlin	25	1
3	Lisbon	30	1
4	Lisbon	30	1
5	Berlin	18	0
6	Lisbon	26	0
7	Berlin	30	1
8	Berlin	26	0
9	Berlin	25	1
10	Madrid	25	1

Solución avanzada

Las técnicas mencionadas producen distorsiones en la distribución conjunta del vector aleatorio. Estas distorsiones pueden ser muy considerables y afectar en gran medida el entrenamiento del modelo. Para reducir este efecto se puede utilizar **MICE (Multivariate Imputation by Chained Equation)**

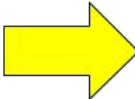
1. Se trata cada columna con missing values como la variable dependiente de un problema de regresión.
2. Se van haciendo los fits de cada columna de manera secuencial.
3. Se utiliza la regresión para completar los missing values.

One hot encoding

En muchos problemas de Machine Learning, puedo tener como dato de entrada variables categóricas. Por ejemplo, una columna con información sobre el color: {rojo, amarillo, azul}

Para este tipo de información, donde no existe una relación ordinal natural entre las categorías, no sería correcto asignar números a las categorías.

Una forma más expresiva de resolver el problema es utilizar “one hot encoding” y transformar la información en binaria de la siguiente manera.



Color	Red	Yellow	Green
Red	1	0	0
Red	1	0	0
Yellow	0	1	0
Green	0	0	1
Yellow	0	0	1

Bibliografía

- The Elements of Statistical Learning | Trevor Hastie | Springer
- An Introduction to Statistical Learning | Gareth James | Springer
- Deep Learning | Ian Goodfellow | <https://www.deeplearningbook.org/>
- Stanford | CS229T/STATS231: Statistical Learning Theory | <http://web.stanford.edu/class/cs229t/>
- Mathematics for Machine Learning | Deisenroth, Faisal, Ong
- Artificial Intelligence, A Modern Approach | Stuart J. Russell, Peter Norvig

