

Introducción a la Inteligencia Artificial

Facultad de Ingeniería

Universidad de Buenos Aires

Ing. Lautaro Delgado
(lautarodc@unops.org)



Índice

1. Repaso general Clase 1-2
2. Más Ingeniería de Features
3. Aprendizaje Estadístico - Regresión Lineal
 - a. Concepto
 - b. Demostración Matemática
 - i. Error Cuadrático Medio
 - ii. Máxima Verosimilitud
4. Ejercicio de Aplicación



Más Ingeniería de Features

Ejercicio #1 | 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.

Dado un dataset X de n muestras y m features, implementar un método en numpy para normalizar con z-score. Pueden utilizar `np.mean()` y `np.std()`.

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



Missing Values - 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



Ejercicio #2 | Remover filas y columnas con NaNs en un dataset

Dado un dataset, hacer una función que, utilizando numpy, filtre las columnas y las filas que tienen NaNs.



Missing Values - 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

Ejercicio #3 | Reemplazar NaNs por la media de la columna.

Dado un dataset, hacer una función que utilizando numpy reemplace los NaNs por la media de la columna.



Missing Values - 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)**

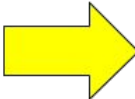
1. Se trata cada columna con missing values como la variable dependiente de un problema de regresión.
2. Se van haciendo los fits de cada columna de manera secuencial.
3. Se utiliza la regresión para completar los missing values.

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

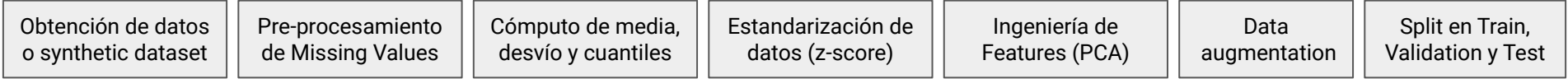
Machine Learning Terminology

- Raw vs. Tidy Data
- Training vs. Holdout Sets
- Baseline
- Parameters vs. Hyperparameters
- Classification vs. Regression
- Model-Based vs. Instance-Based Learning
- Shallow vs. Deep Learning



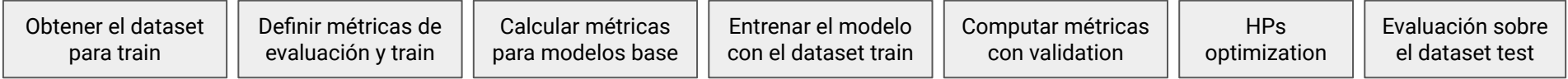
Dataset pipeline

Acciones que generalmente se ejecutan sobre los datasets.

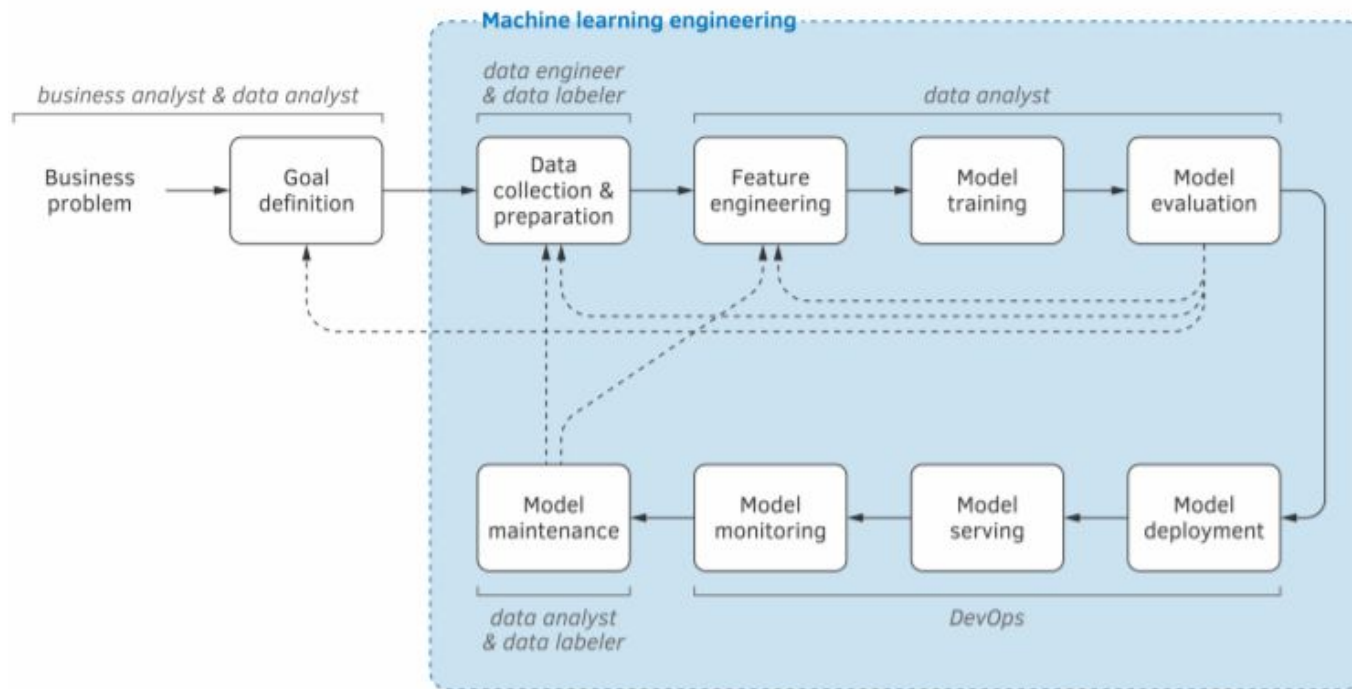


Model pipeline

Pasos involucrados al entrenar un modelo de Machine Learning



Machine Learning Pipeline



Ejercicio #4 | Dado un dataset X separarlo en 70 / 20 / 10

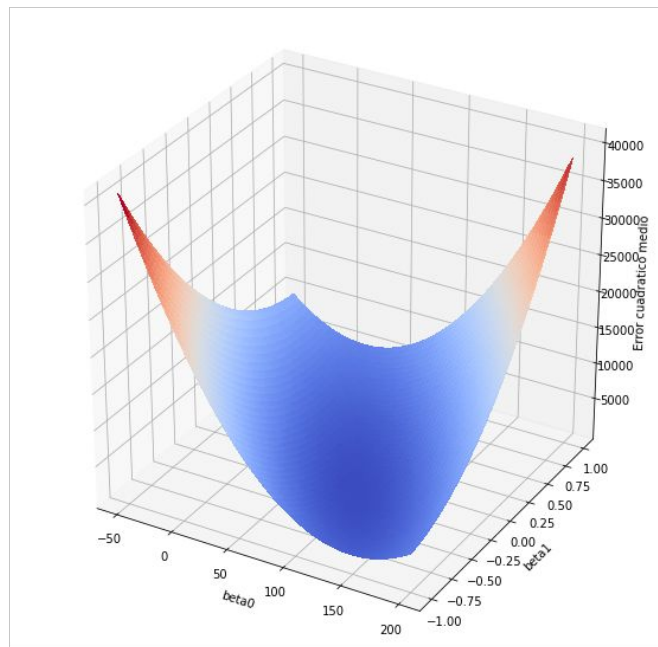
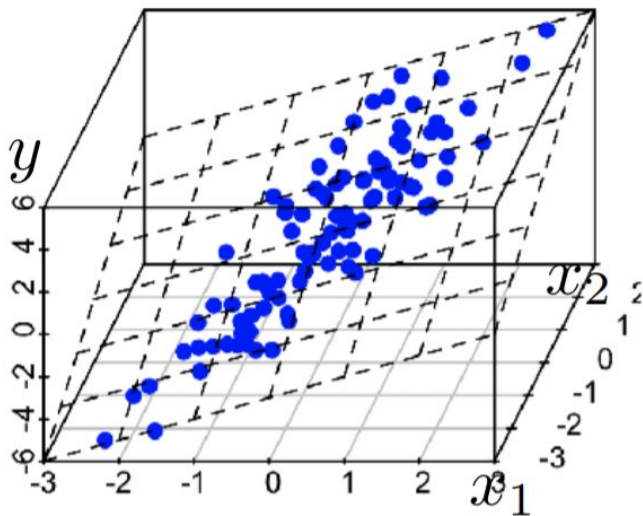
Como vimos en el ejercicio integrador, en problemas de Machine Learning es fundamental que separemos los datasets de n muestras, en 3 datasets de la siguiente manera:

- **Training dataset:** los datos que utilizaremos para entrenar nuestros modelos. Ej: 70% de las muestras.
- **Validation dataset:** los datos que usamos para calcular métricas y ajustar los hiperparámetros de nuestros modelos. Ej: 20% de las muestras.
- **Testing dataset:** una vez que entrenamos los modelos y encontramos los hiperparámetros óptimos de los mismos, el testing dataset se lo utiliza para computar las métricas finales de nuestros modelos y analizar cómo se comporta respecto a la generalización. Ej: 10% de las muestras.

A partir de utilizar `np.random.permutation`, hacer un método que dado un dataset, devuelva los 3 datasets como nuevos numpy arrays.

Regresión Lineal

En ésta clase vamos a ver el framework teórico detrás de la gran mayoría de los modelos de Machine Learning, tal como es el aprendizaje estadístico. Para ello, vamos a utilizar como modelo base la regresión lineal.



Jamboard - Desarrollo Matemático Regresión Lineal



Ejercicio Integrador Regresión Lineal

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

