

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

- 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.

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

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

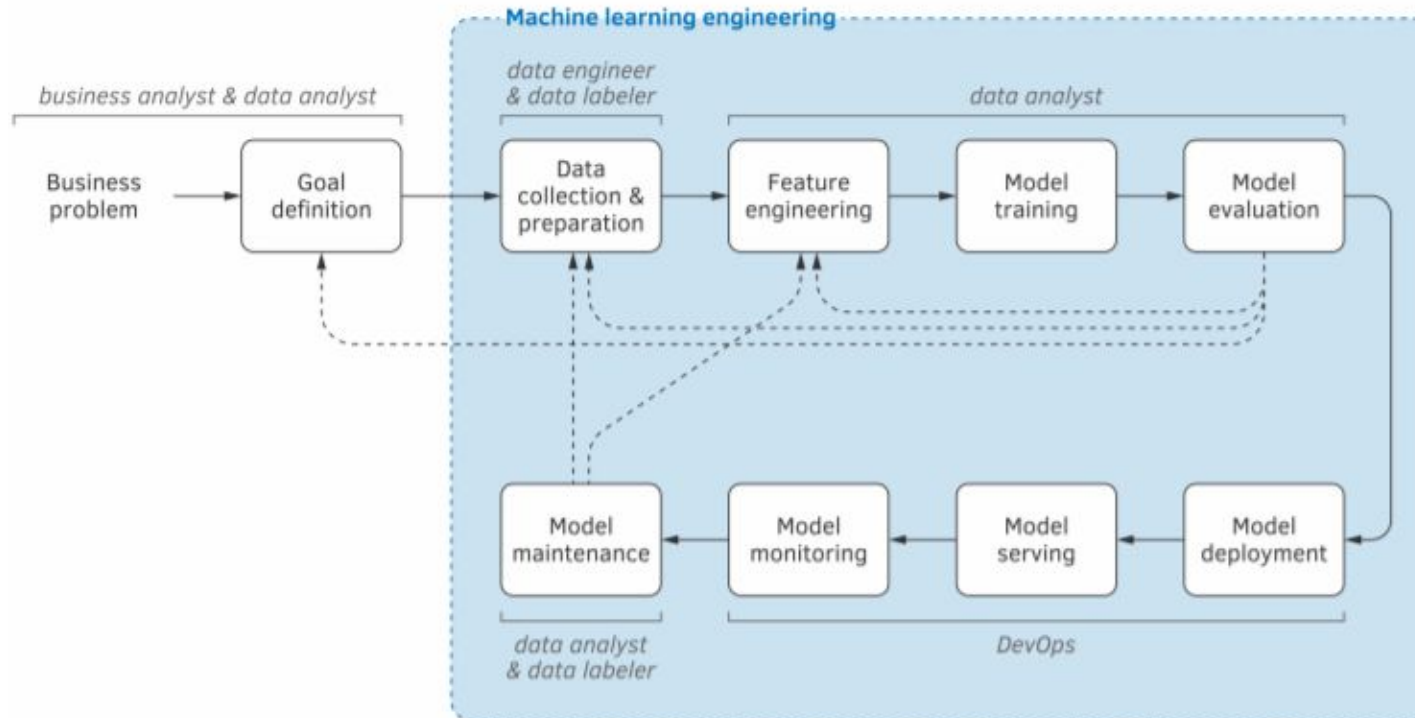
Computar métricas
con validation

HPs
optimization

Evaluación sobre
el dataset test

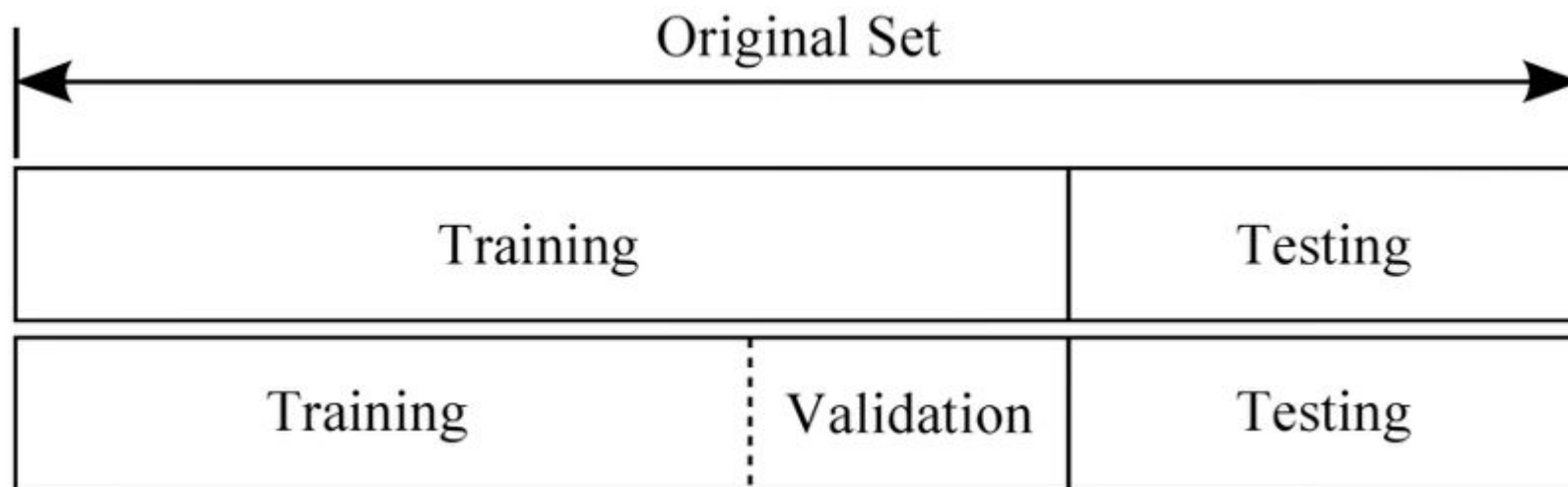


Machine Learning Pipeline



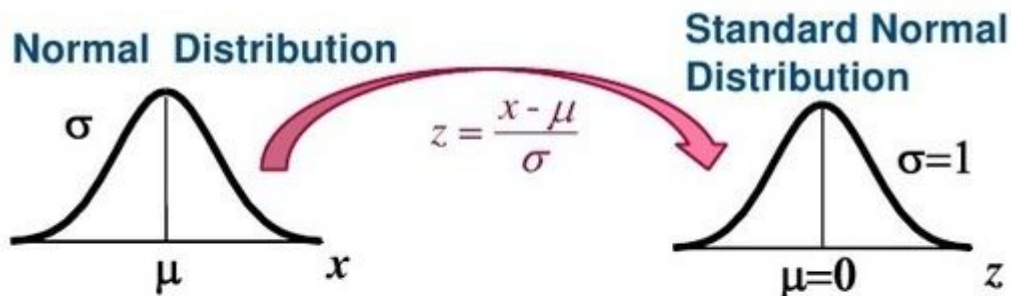
Ingeniería de Features

Train - test - validation



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)**

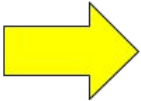
1. Se trata cada columna con missing values como la variable dependiente de un problema de regresión.
2. Se van haciendo los fits de cada columna de manera secuencial.
3. Se utiliza la regresión para completar los missing values.

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

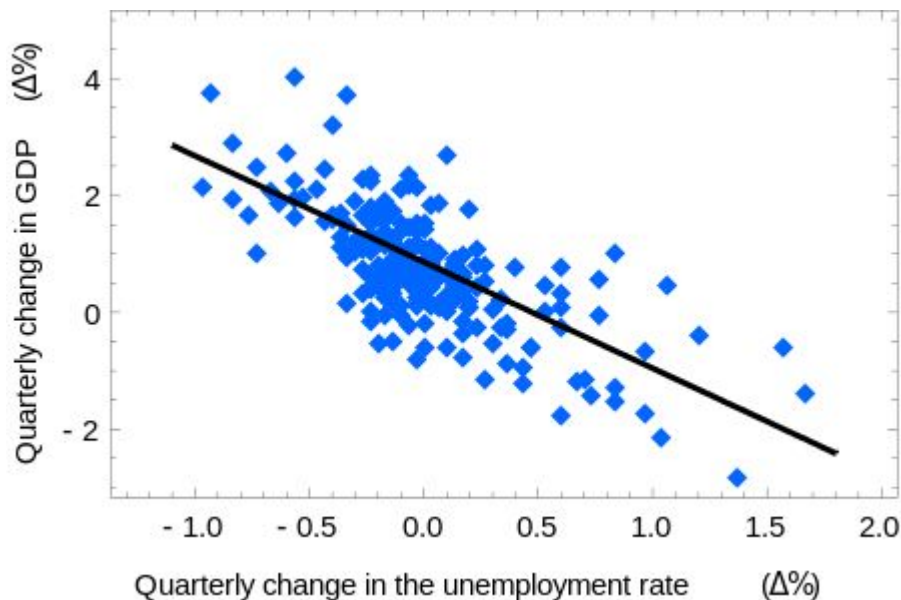
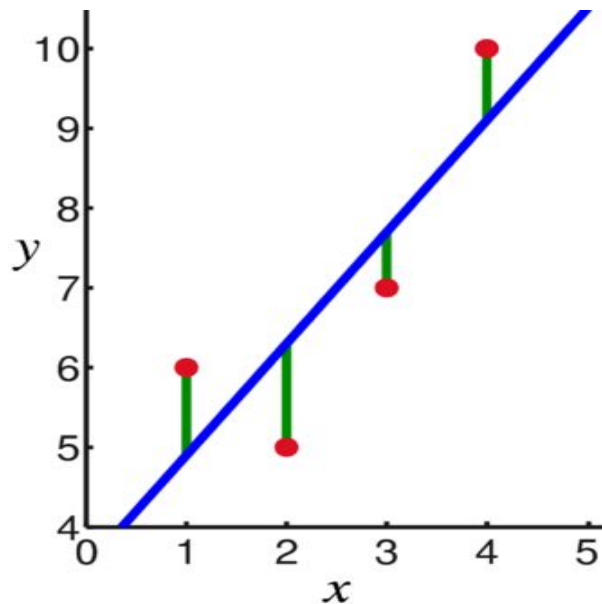
Regresión lineal

$$y_i = \beta_0 + \beta_1 x_{i1} + \cdots + \beta_p x_{ip} + \varepsilon_i = \mathbf{x}_i^\top \boldsymbol{\beta} + \varepsilon_i, \quad i = 1, \dots, n,$$

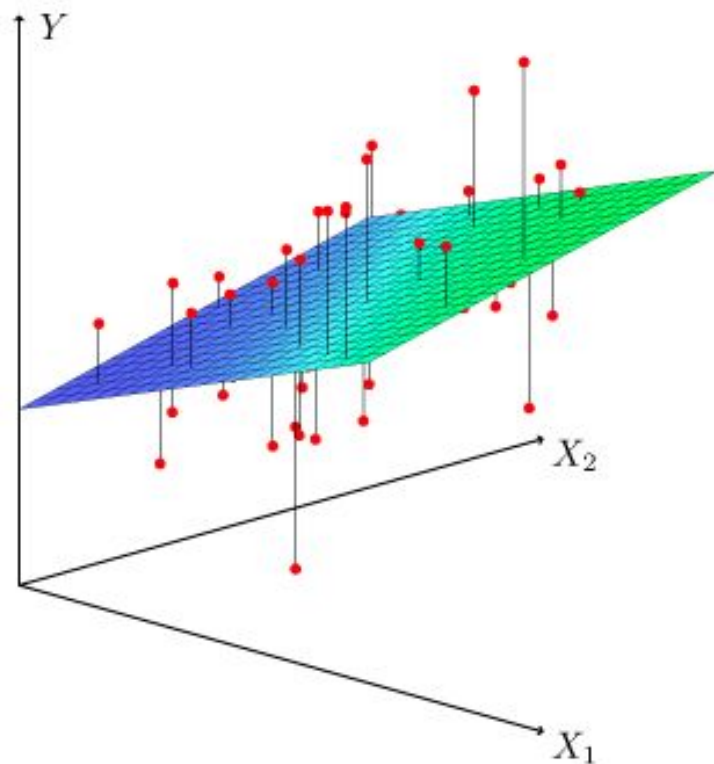


Regresión Lineal $y_i = \beta_0 + \beta_1 x_{i1} + \dots + \beta_p x_{ip} + \varepsilon_i = \mathbf{x}_i^T \boldsymbol{\beta} + \varepsilon_i, \quad i = 1, \dots, n,$

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



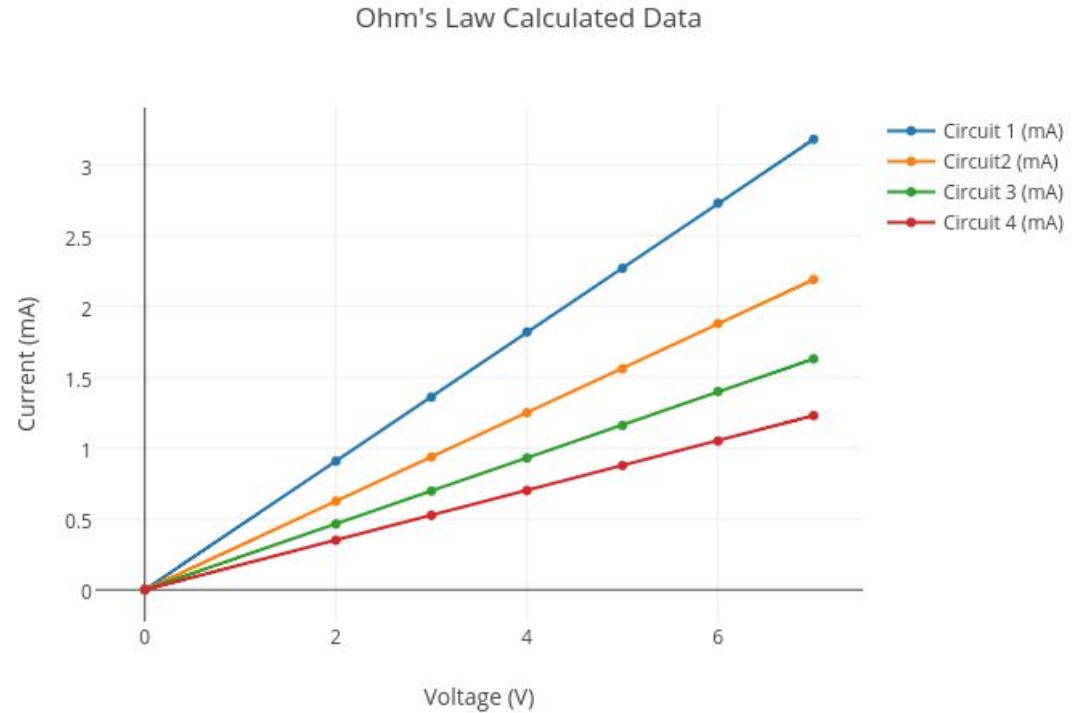
Regresión Lineal $y_i = \beta_0 + \beta_1 x_{i1} + \cdots + \beta_p x_{ip} + \varepsilon_i = \mathbf{x}_i^T \boldsymbol{\beta} + \varepsilon_i, \quad i = 1, \dots, n,$



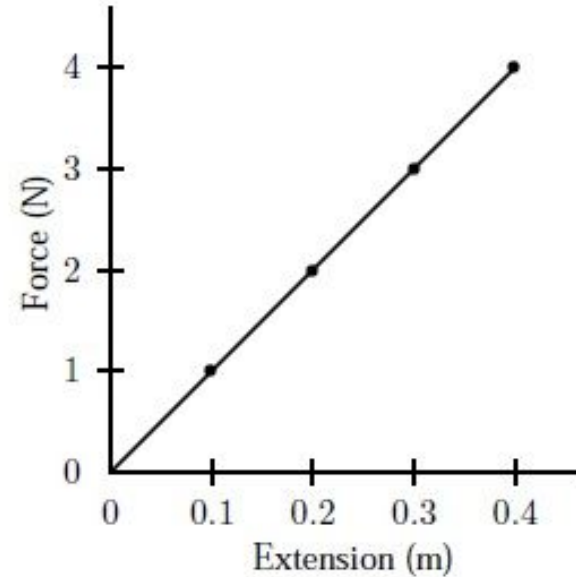
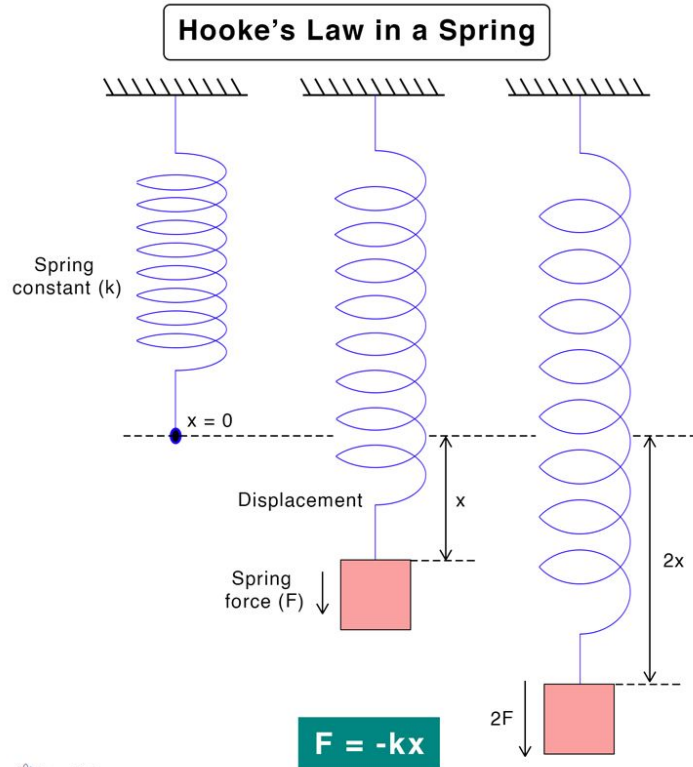
Ley de Ohm

$$I = V/R$$

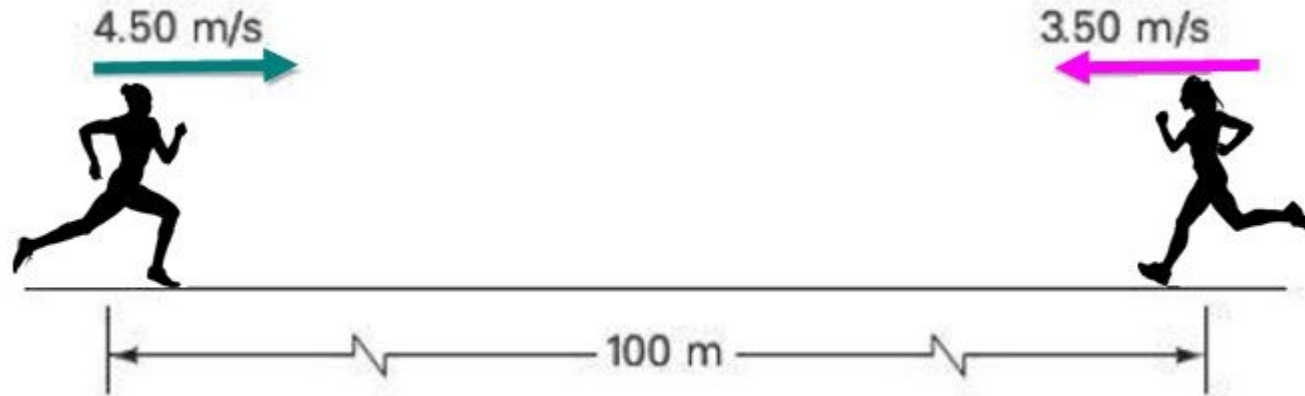
R constante



Ley de Hooke



Movimiento rectilíneo uniforme



$$x(t) = x(t_0) + V * t$$

Población de parásitos



Ejemplo: En un estudio sobre la población de un parásito se hizo un recuento de parásitos en 15 localizaciones con diversas condiciones ambientales.

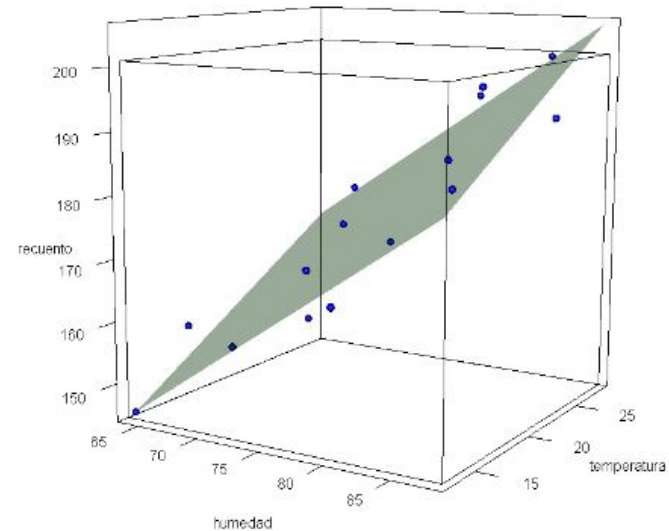
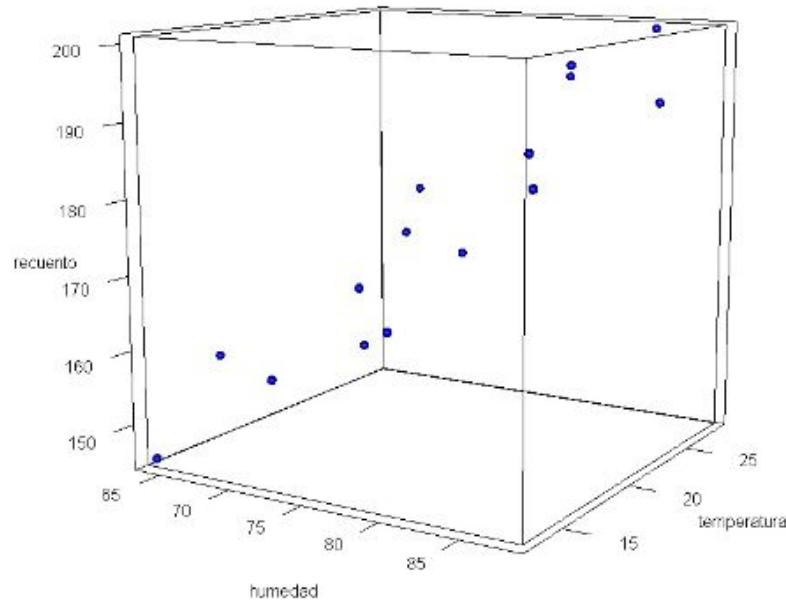
Los datos obtenidos son los siguientes:

Temperatura	15	16	24	13	21	16	22	18	20	16	28	27	13	22	23
Humedad	70	65	71	64	84	86	72	84	71	75	84	79	80	76	88
Recuento	156	157	177	145	197	184	172	187	157	169	200	193	167	170	192

Población de parásitos

$$\text{Recuento} = \beta_0 + \beta_1 \text{Temperatura} + \beta_2 \text{Humedad} + \epsilon$$

$$\text{Recuento} = 25.7115 + 1.5818 \text{Temperatura} + 1.5424 \text{Humedad}$$



Jamboard



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

