Machine Learning Workshop

Mentor: Nicolas Känzig

Email: nkaenzig@gmail.com
Workshop Repository: https://github.com/nkaenzig/ml-workshop

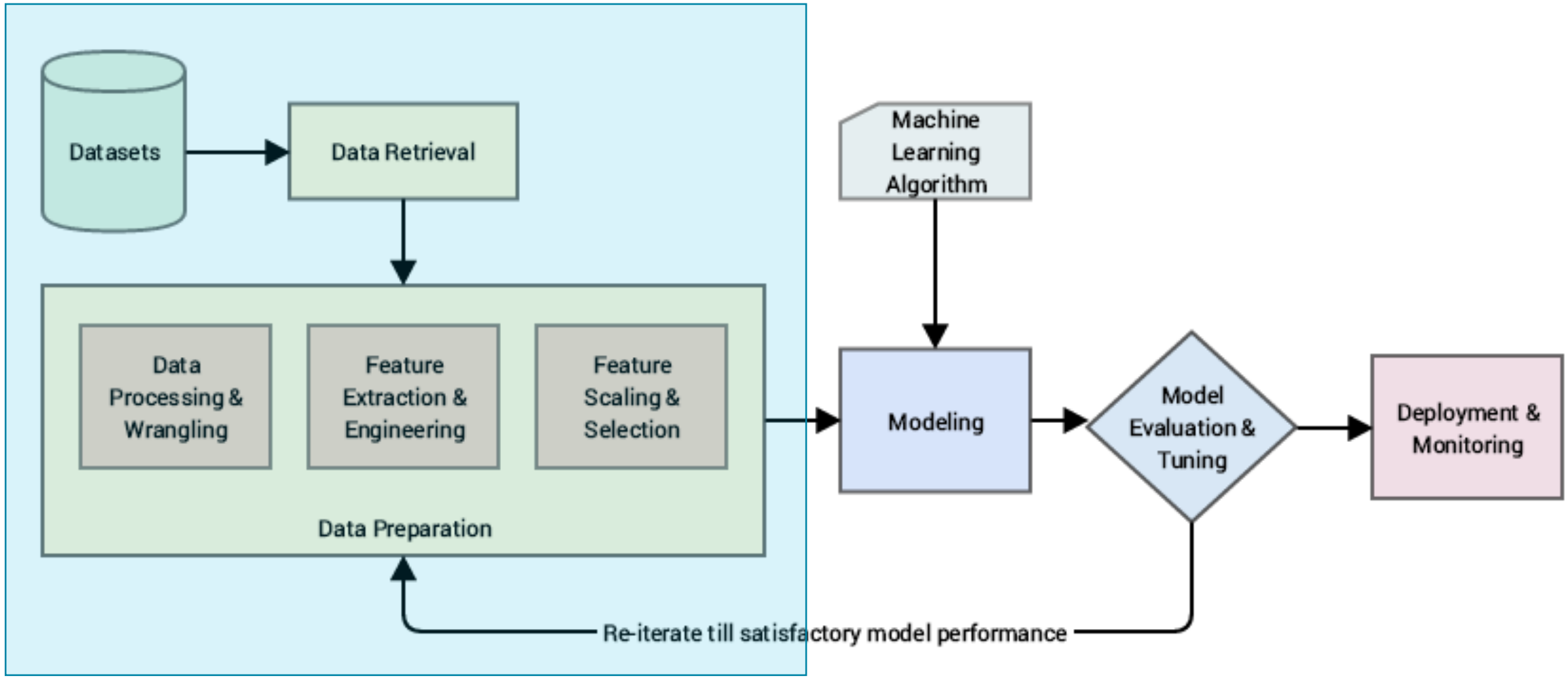
Contenido

Modulo 1
- Introducción ML
- Python crashcourse

Modulo 2
- Análisis de datos
- Preprocesamiento de datos
- Ejemplo ML

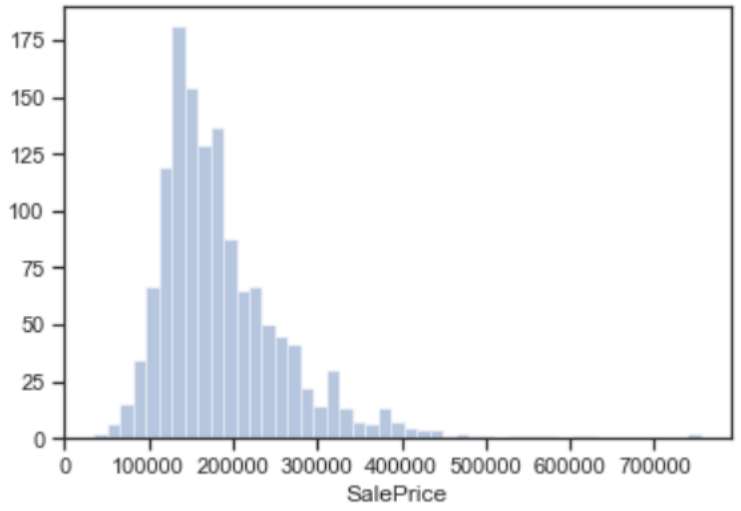
Modulo 3
- Modelos de ML
- Técnicas de evaluación
- Ejemplos ML

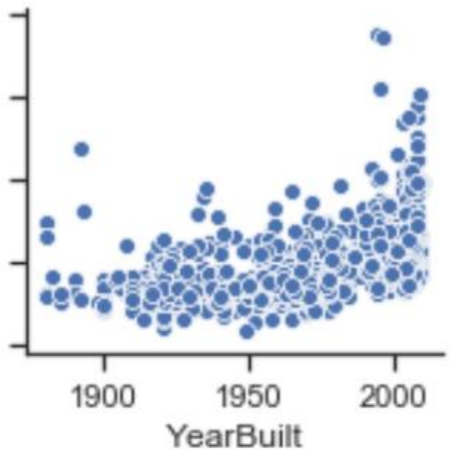
Análisis exploratorio de datos

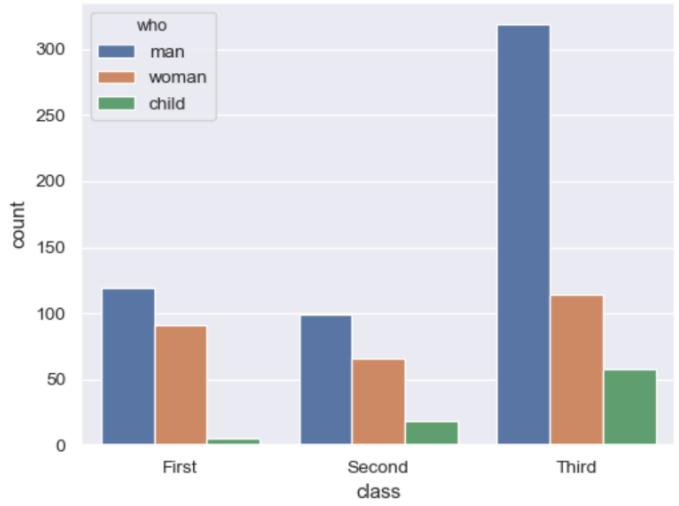

Datasets
Data Retrieval
Machine Learning Algorithm
Data Processing & Wrangling
Feature Extraction & Engineering
Feature Scaling & Selection
Data Preparation
Modeling
Model Evaluation & Tuning
Deployment & Monitoring
Re-iterate till satisfactory model performance

Análisis exploratorio de datos

- Crear visualizaciones y calcular medidas estadísticas para mejor entender los datos



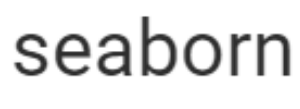




Instrumentos

- Programación

- Liberarías



- IDEs

Numpy

- Liberaría para computación científica
- Algebra Lineal:
  - Objetos para vectores/matrices (i.e. arrays)
- Estadistica:
  - Operaciones básicas: e.g. mean, median, std, percentiles, …

- Muchas de las operaciones son implementados en C

  → Mucho mas rápido que Python sin Numpy

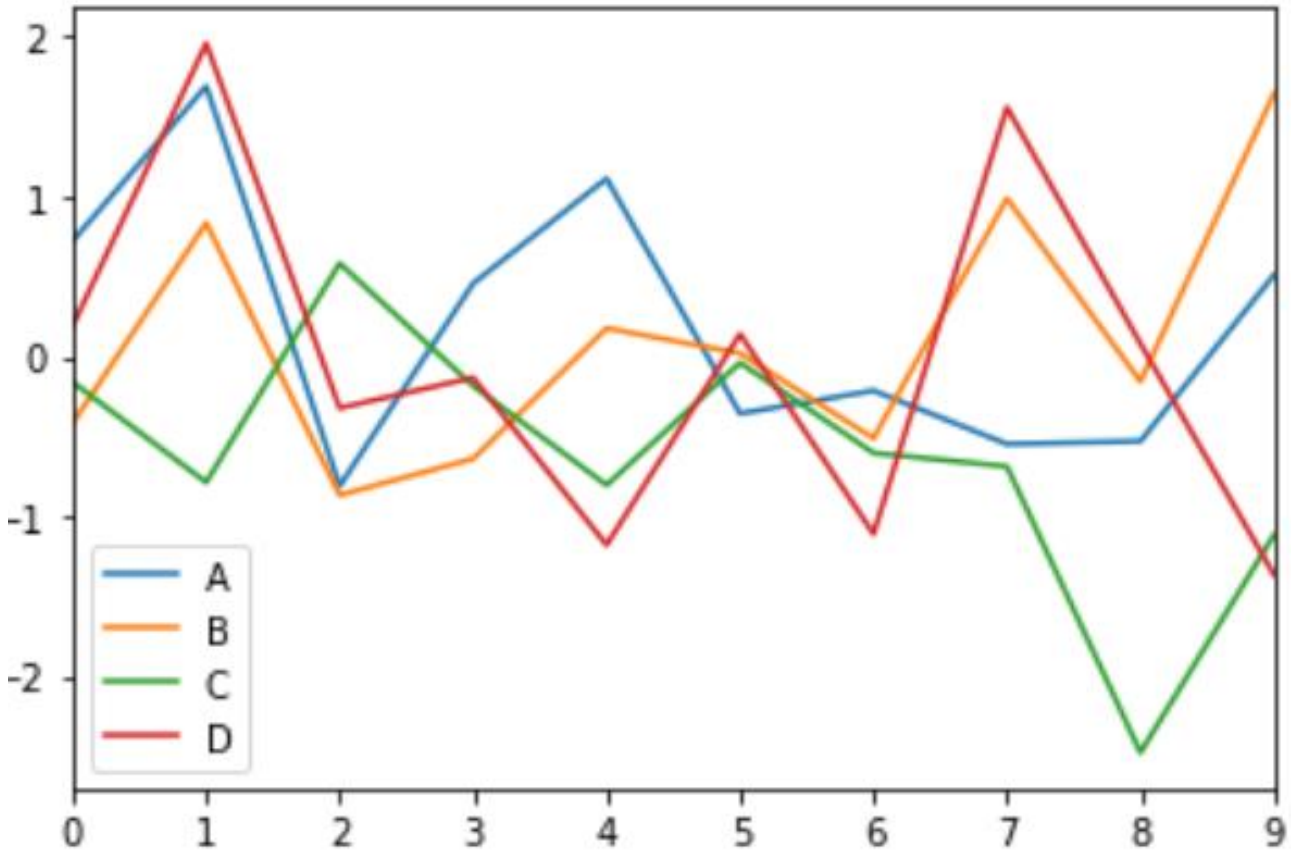
Pandas

	A	B	C	D
0	1.283449	0.405647	0.633235	-0.633953
1	-0.137045	-0.498740	-0.966406	-0.720781
2	-1.066049	0.458651	-1.384483	-0.174038
3	-0.823852	0.250134	0.973628	-0.174436
4	0.762657	-0.056813	1.097659	-0.449781
5	0.755400	-1.310918	0.146741	-0.315770
6	-0.523010	-0.438491	-1.010650	0.097777

- **DataFrame** class
  - “Una Tabula con Index y Columna”
  - Valores son un numpy.array
- Viene con muchas funciones útiles para análisis y preprocesamiento de datos
  - read_csv(), read_excel(), …
  - Encontrar missing data: isna()
  - Borrar filas/columnas donde faltan valores: dropna()
  - Llenar valores que faltan: fillna()
  - Slicing, reshaping, sampling, shuffling, concatenating, …
  - Tiene funciones de matplotlib ya integrado: plot.scatter

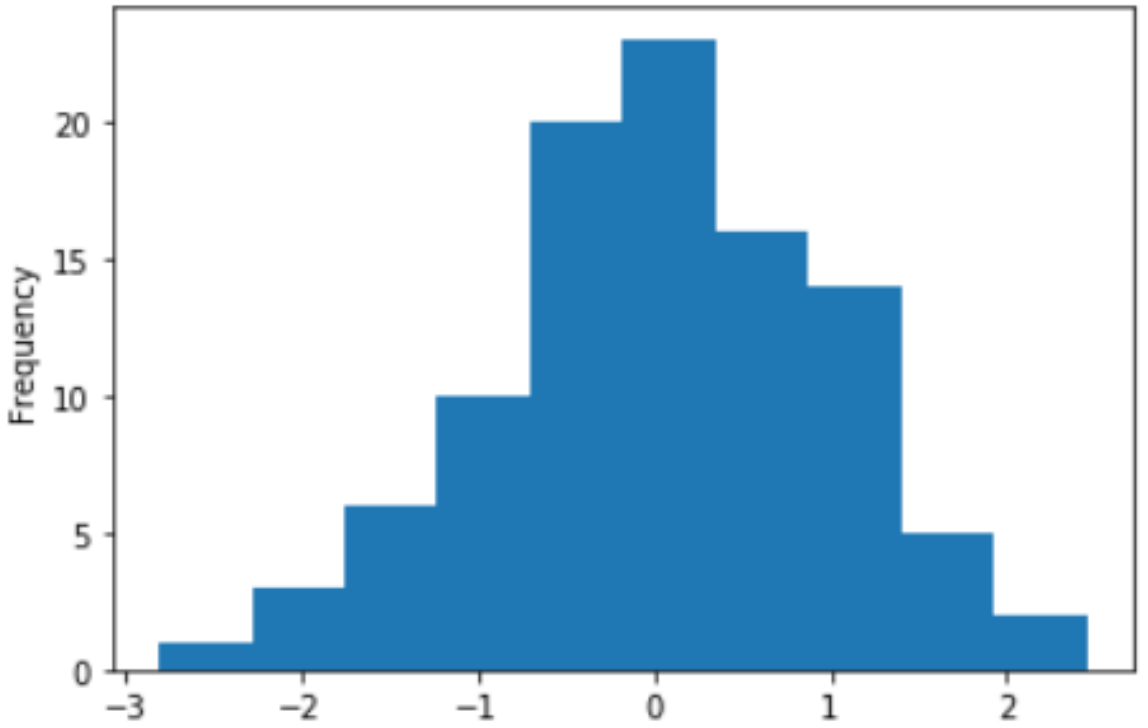
Matplotlib

- Simple line plot:
  - **plt.plot**()



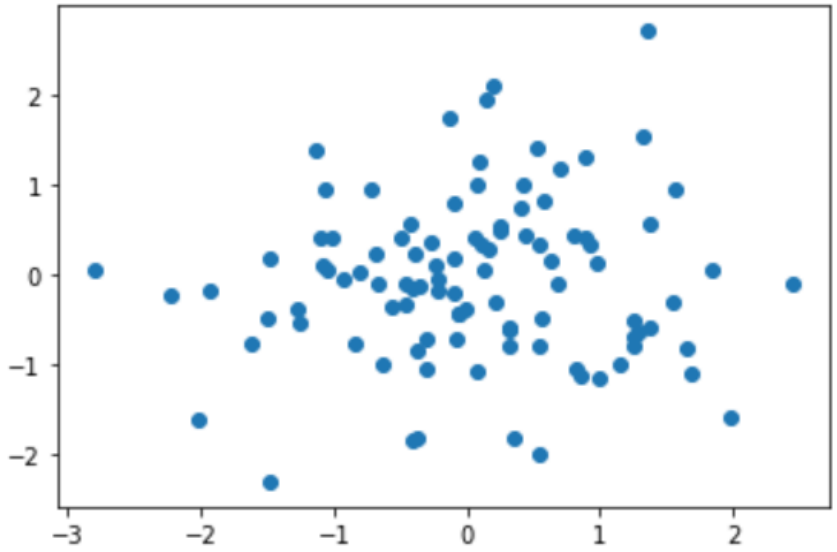
Matplotlib

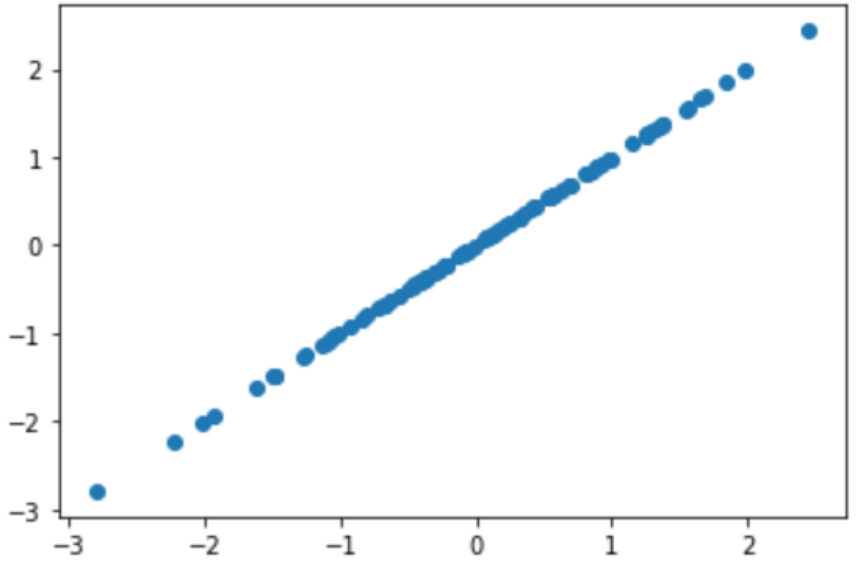
- Histograms:
  - **plt.hist**()
  - Útil para visualizar distribuciones



Matplotlib

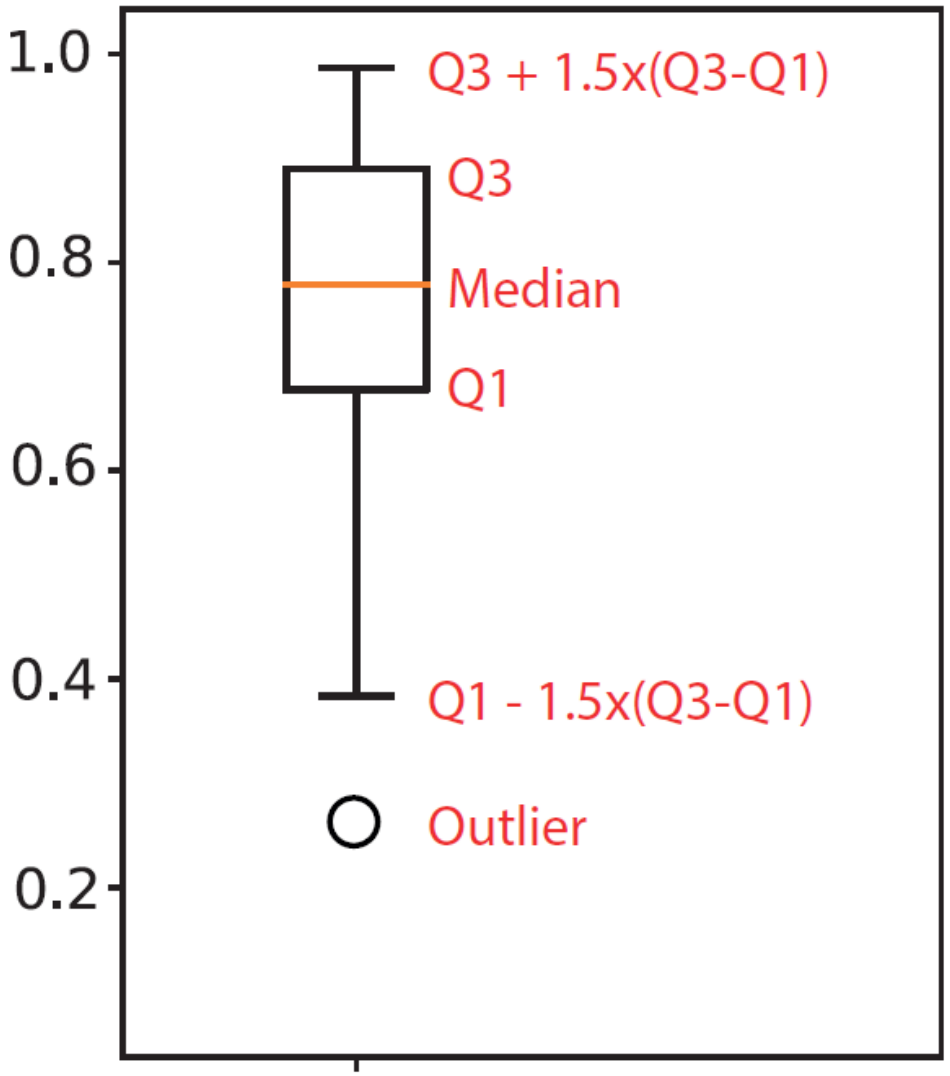
- Scatter plot:
  - **plt.scatter**()
  - Útil para visualizar correlaciones





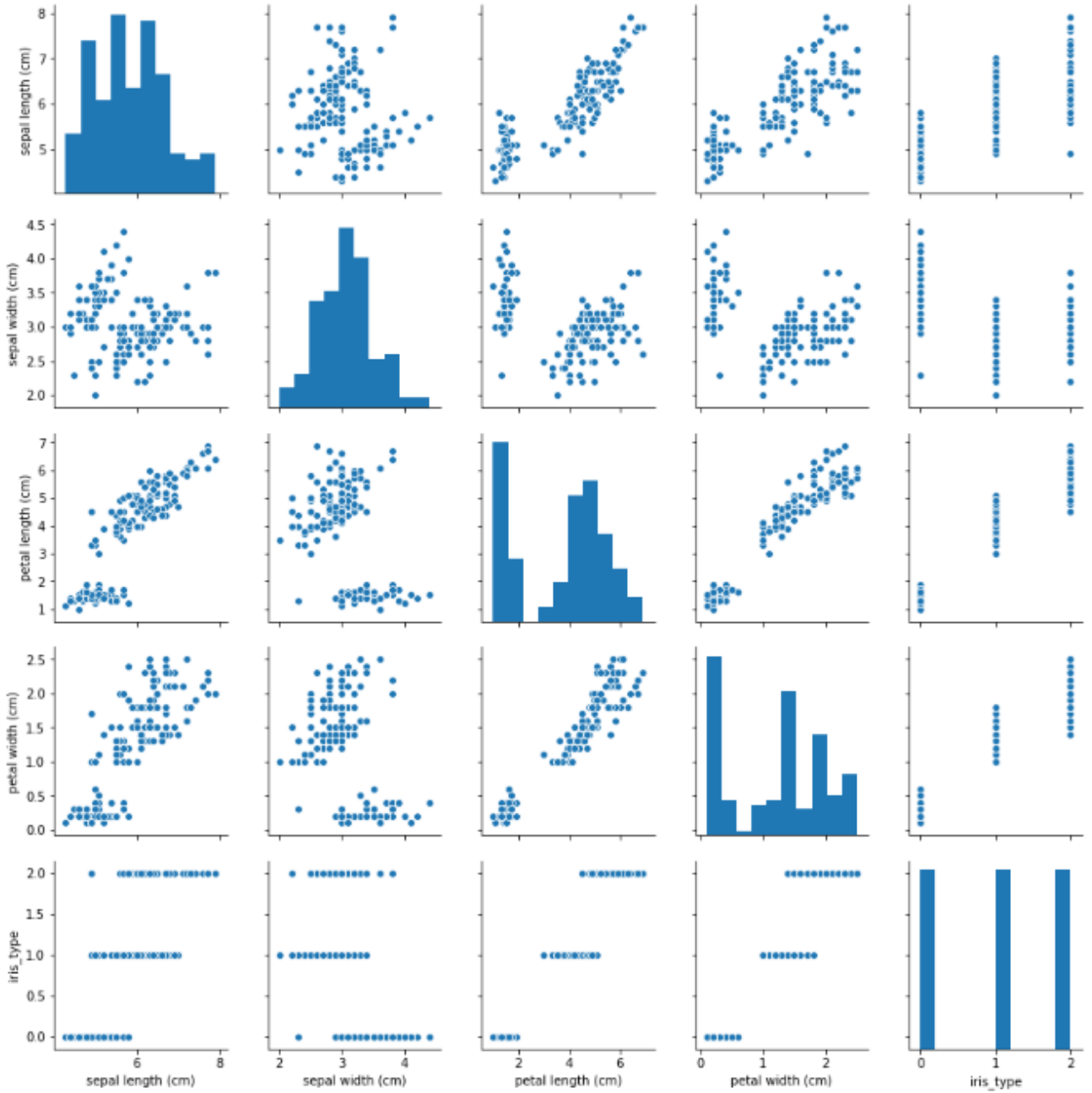
Matplotlib

- Boxplots
  - **plt.boxplot()**



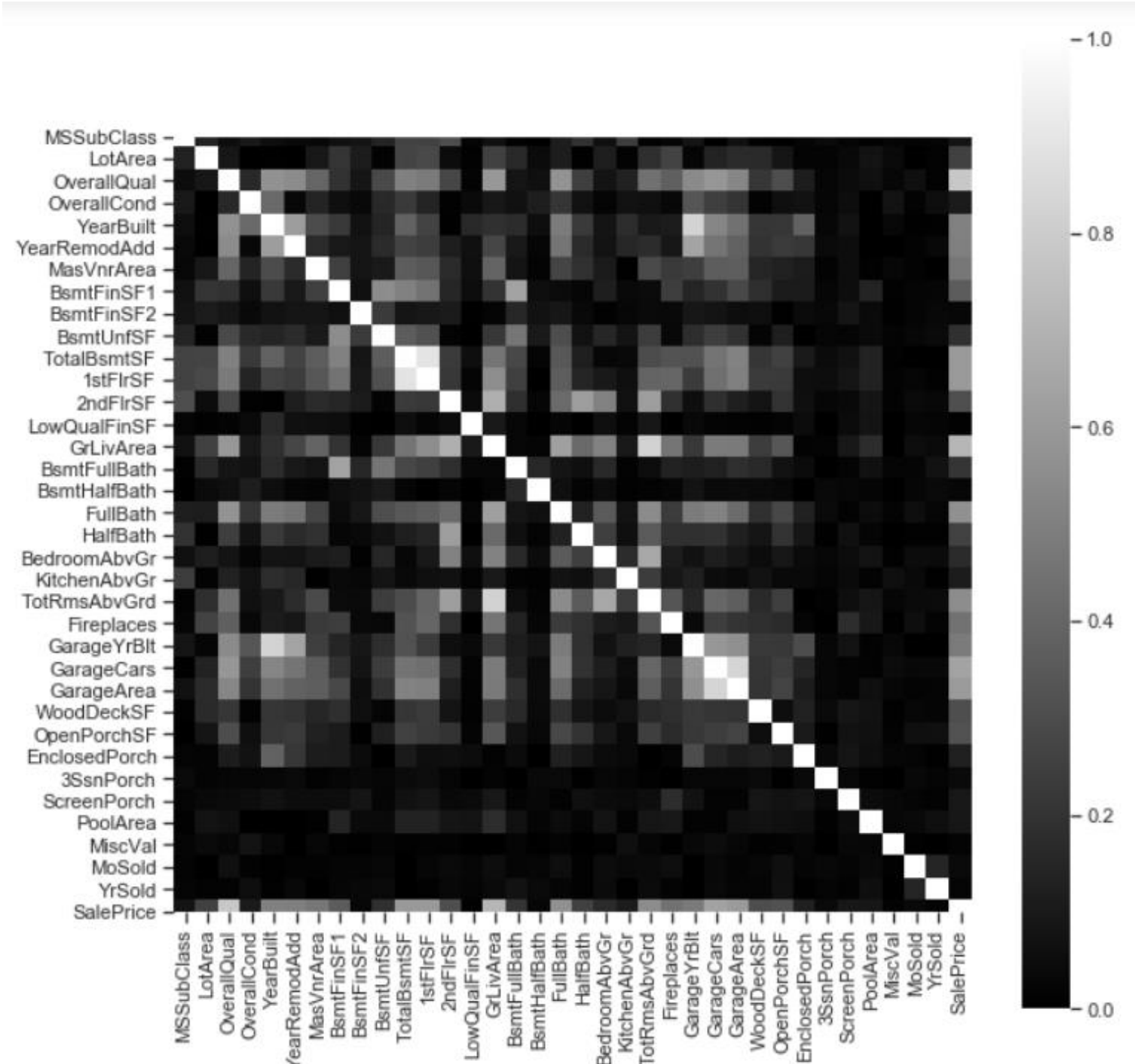
Seaborn

- Pairplots
  - **sns.pairplot()**



Seaborn

- Heatmap
  - **sns.heatmap**(df.corr().abs())



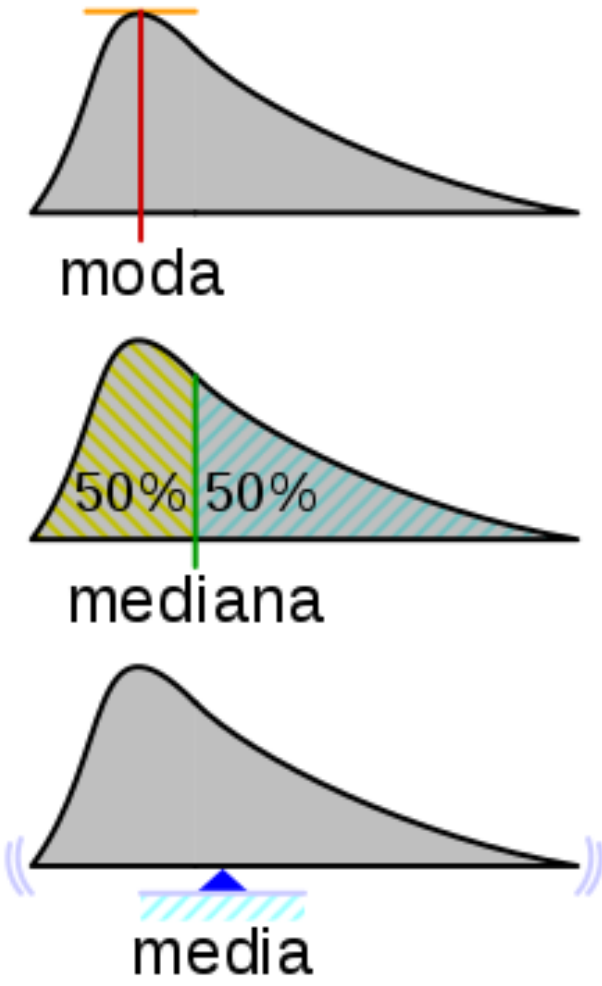
Estadística

- La media

$$\mu = \frac{1}{n}\left(\sum_{i=1}^{n} x_i\right) = \frac{x_1 + x_2 + \cdots + x_n}{n}$$

- Desviación estándar

$$\sigma = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(x_i - \mu)^2}$$

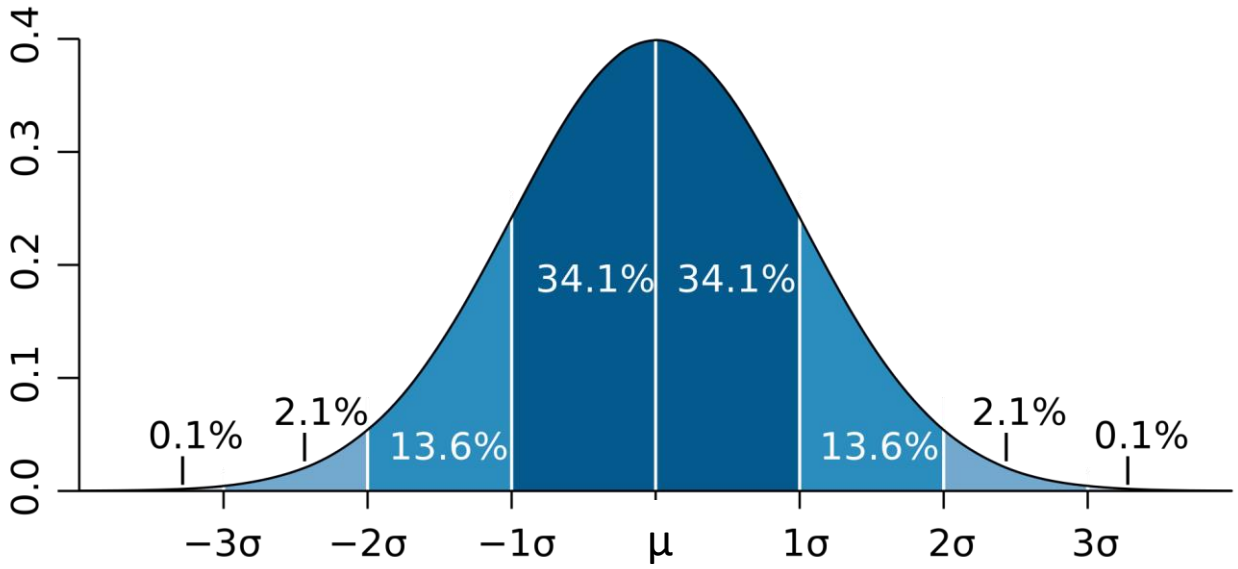


Estadística

- Distribución normal

$$f(x \mid \mu, \sigma^2) = \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left(-\frac{(x-\mu)^2}{2\sigma^2}\right)$$

  - μ : *La media (valor promedio)*
  - σ : *Desviación estándar*



Estadística v.s. ML/Data Science

- Estadística
  - Pocos datos
    - Difícil tomar conclusiones sobre la distribución original
    - → Pruebas de hipótesis, Intervalos de confianza, resultados significantes, ...

- Data Science / Machine Learing
  - Muchos datos
    - Mucho mas fácil tener confianza que los resultados obtenidos son validos para la distribución original