Instituto Politécnico Nacional
Escuela Superior de Cómputo

JCP_helper, aplicación web para evaluar el aprovechamiento académico en fundamentos de programación.

ARTEFACTO PREPARACIÓN Y LIMPIEZA DE DATOS

Presentan:

- **Chavarría Vázquez Luis Enrique**
- **Machorro Vences Ricardo Alberto**
- **Juarez Espinosa Ulises**

Directoras:

- Hernández Jaime Josefina
- Rivera de la Rosa Mónica

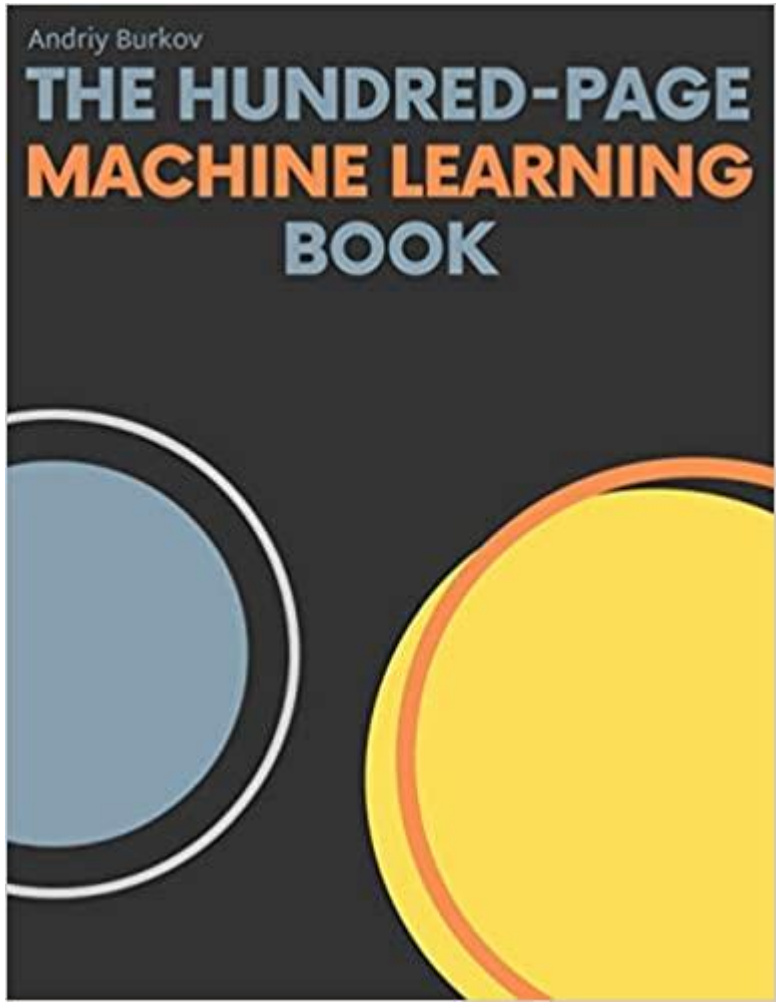
Artefacto: preparación y limpieza de datos.

¿QUÉ ES EL MACHINE LEARNING?

ALGORITMOS CAPACES DE TOMAR DECISIONES SIN NECESIDAD DE INTERVENCIÓN HUMANA

VIEJA Y NUEVA **CIENCIA DE LOS DATOS**

- **Pascal y Fermat (1654). Teoría de la probabilidad**
- **Probabilidad condicional. Thomas Bayes (1702-1761)**
- **Recta de regresión. Francis Galton (1822-1911)**
- **Normal multiv, matriz de correlación. Edgeworth (1845-1926)**
- **Contraste chi-2 de homogeneidad. Karl Pearson (1857-1936)**
- **Componentes principales. Hotelling (1933)**
- **Análisis factorial. Charles Spearman (1863-1945)**
- **Análisis discriminante (Clasificación en categorías). Fisher (1933)**
- Redes Neuronales Artificiales. Rosenblatt (1956)
- Densidad Kernel. Parzen (1962)
- Árboles de decisión (1960). Quinlan (1983)
- Nearest Neighbors. Friedman (1975)
- Algoritmos Genéticos. Holland (1975)
- Bootstrapping. Efron (1979)
- Support Vector Machines. Vapnik (1995)



METODOLOGÍA MACHINE LEARNING

1. **No imponer especificación** ni teoría.
2. Dejar que **hablen los datos por si mismos**
3. Un **algoritmo** encuentra la **relación input-output**.

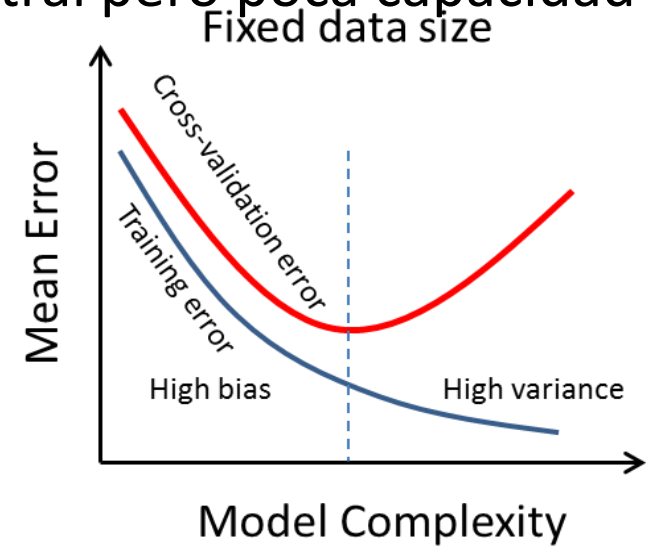
Artefacto: Marco teórico, peligros del manejo de datos.

LOS PELIGROS

- **DATA SNOOPING (FISGONEO DE DATOS)**
- **FALSOS POSITIVOS.**
- **OVERFITTING (SOBRE AJUSTE)**
- **LA MALDICIÓN DE LA DIMENSIONALIDAD**

EL OVERFITTING

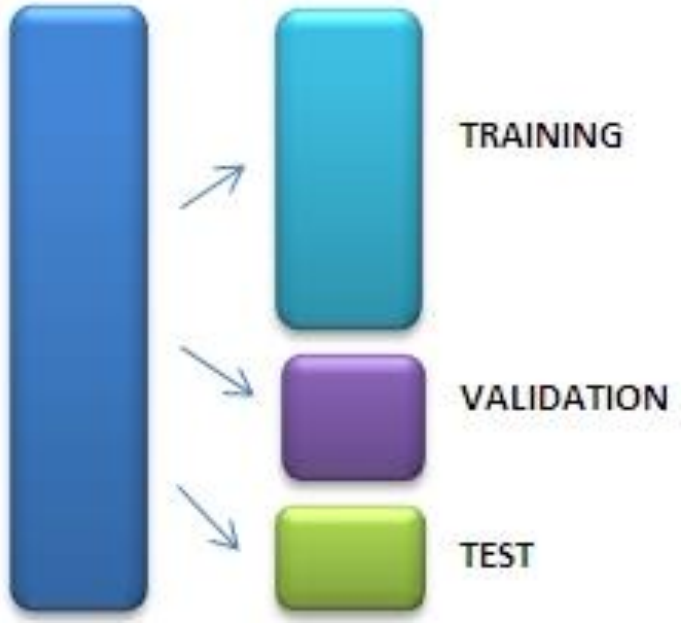
- **SOBRE AJUSTE DE DATOS**
- **Modelos con demasiados parámetros** consigue un perfecto ajuste intra-muestral pero poca capacidad predictiva



- Cross-validation: validar el modelo en un conjunto de datos diferente del que se ha entrenado.

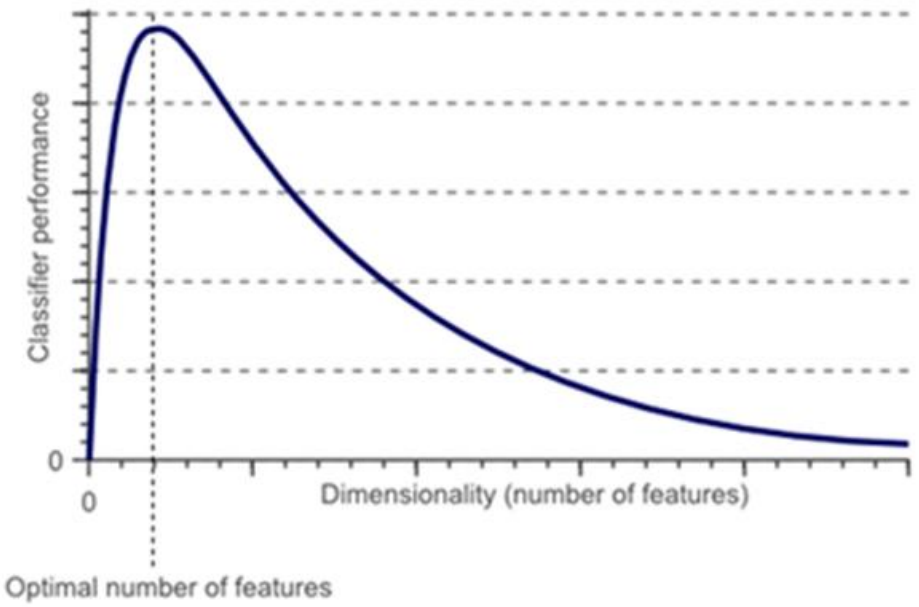
NO SELECCIONAR EL MEJOR MODELO SOBRE LOS DATOS DE ENTRENAMIENYO

- Dividir la base de datos en tres subconjuntos
- **Conjunto de entrenamiento**:
  - Ajustar todos los modelos
- **Conjunto de validación**:
  - Seleccionar el mejor de modelo
- **Conjunto test**
  - Error real cometido del modelo seleccionado



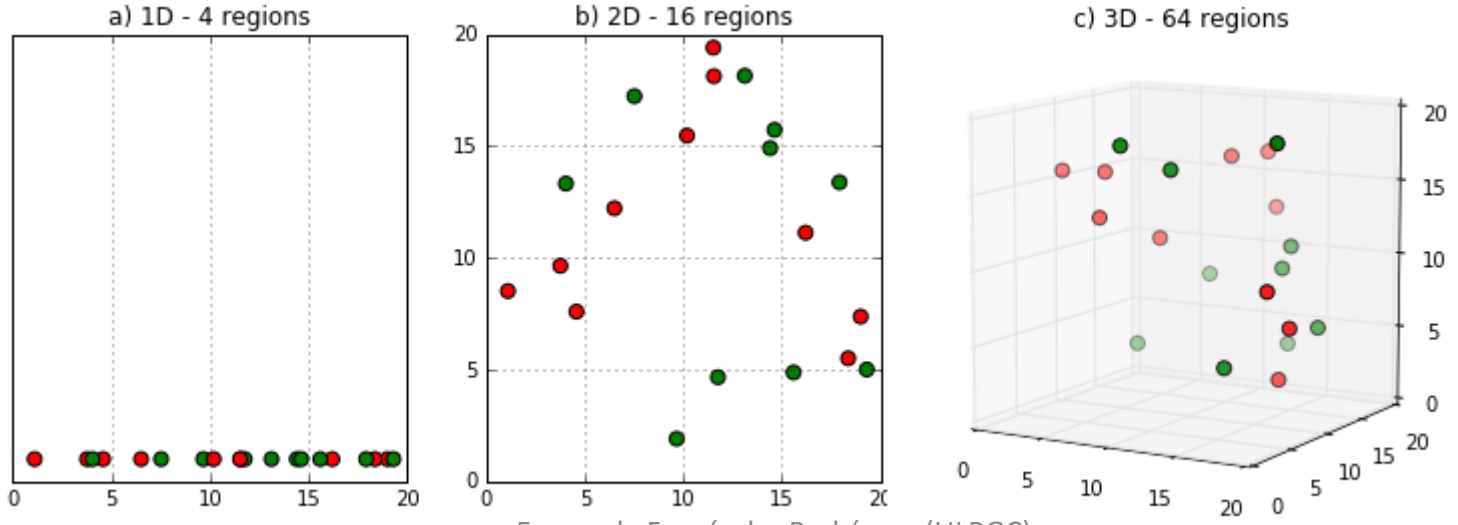
LA MALDICIÓN DE LA DIMENSIONALIDAD

- **Cuando aumenta el número de variables se requiere un aumento exponencial del número de observaciones para conseguir significatividad estadística.**

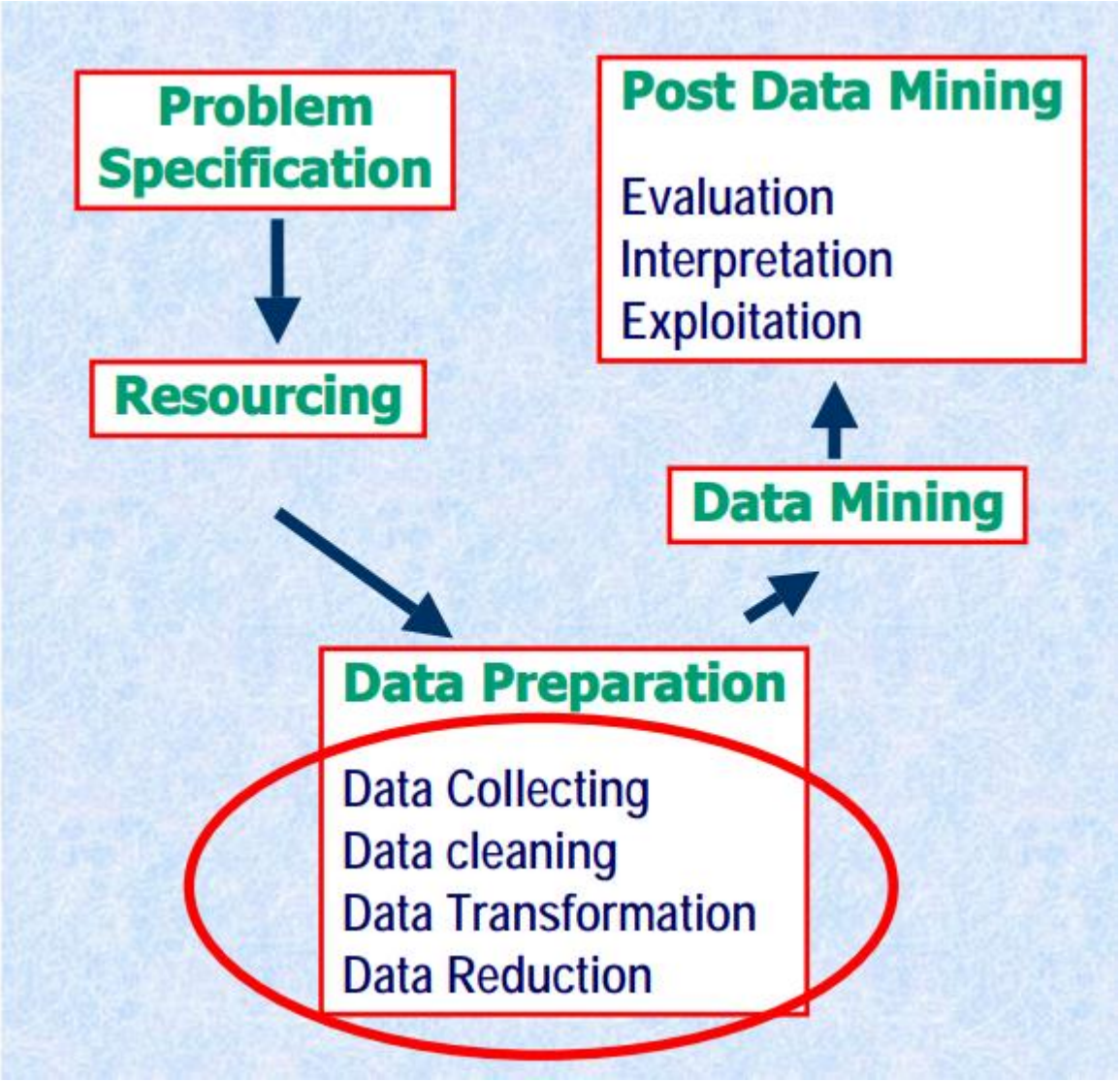


LA MALDICIÓN DE LA DIMENSIONALIDAD

- **Los datos disponibles se vuelven dispersos al aumentar el número de variables (dimensión)**
- Número medio de objetos en un hipercubo de lado 5
- Intervalo: 20/4, Cuadrado: D=20/4^2=1.25
- Cubo : 20/4^3=0.3125



Artefacto: Preprocesamiento de la data.(Preparar y limpiar los datos)



•Nos encontramos en esta parte del proceso.

•**Limpieza de datos**. Quitar o corregir, de los datos sin procesar, aquellos registros con valores dañados o no válidos, y quitar registros en los que falten muchas columnas.

•**Selección y partición de instancias**. Seleccionar datos desde los conjuntos de datos de entrada para crear los conjuntos de entrenamiento, evaluación (validación) y prueba. Este proceso incluye técnicas para las muestras repetidas aleatorias, el sobremuestreo de clases minoritarias y la partición estratificada.

•**Ajuste de los atributos**. Mejorar la calidad de un atributo para el AA, que consiste en el escalamiento y normalización de valores numéricos, ingreso de valores faltantes, recorte de valores atípicos y ajuste de valores con desviaciones en las distribuciones.

•**Transformación de la representación**. Convertir un atributo numérico en un atributo categórico (a través del agrupamiento en depósitos) y convertir un atributo de clasificación en una representación numérica (a través de la codificación one-hot, el aprendizaje con conteo, las incorporaciones de atributos dispersos y demás). Algunos modelos funcionan solo con atributos numéricos o categóricos, y otros admiten la combinación de tipos de atributos. Incluso cuando los modelos admiten los dos tipos, pueden beneficiarse con representaciones diferentes (numéricas y de clasificación) del mismo atributo.

•**Extracción de atributos**. Reducir la cantidad de atributos mediante la creación de representaciones de datos más pequeñas, pero más potentes, a través de técnicas como PCA, incorporación, extracción y hashing.

•**Selección de los atributos**. Seleccionar un subconjunto de los atributos de entrada para el modelo de entrenamiento y omitir aquellos que sean irrelevantes o redundantes, a través de métodos de filtrado o métodos wrapper. Esto también puede implicar que se descarten atributos en los que falte una gran cantidad de valores.

•**Creación de los atributos**. Crear atributos nuevos a través de técnicas comunes como la expansión polinómica (con el uso de funciones matemáticas de una variable) o la combinación de atributos (para capturar interacciones de los atributos). Los atributos también pueden crearse con la lógica empresarial para el dominio del caso práctico del AA.

Ejemplo

	Country	Age	Salary	Purchased
2	France	44	72000	No
3	Spain	27	48000	Yes
4	Germany	30	54000	No
5	Spain	38	61000	No
6	Germany	40		Yes
7	France	35	58000	Yes
8	Spain		52000	No
9	France	48	79000	Yes
10	Germany	50	83000	No
11	France	37	67000	Yes

Data Preprocessing Tools

Importing the libraries

```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
```

Importing the dataset

// Nos deja solucionar solo las primeras 3 columna

```
dataset = pd.read_csv('Data.csv')
x = dataset.iloc[:, :-1].values
y = dataset.iloc[:, -1].values
```

```
print(x)
print()
print(y)
```

```
[['France' 44.0 72000.0]
 ['Spain' 27.0 48000.0]
 ['Germany' 30.0 54000.0]
 ['Spain' 38.0 61000.0]
 ['Germany' 40.0 nan]
 ['France' 35.0 58000.0]
 ['Spain' nan 52000.0]
 ['France' 48.0 79000.0]
 ['Germany' 50.0 83000.0]
 ['France' 37.0 67000.0]]

['No' 'Yes' 'No' 'No' 'Yes' 'Yes' 'No' 'Yes' 'No' 'Yes']
```

Taking care of missing data

```
from sklearn.impute import SimpleImputer

imputer = SimpleImputer(missing_values = np.nan, strategy='mean')
imputer.fit(x[:,1:3])
x[:,1:3] = imputer.transform(x[:,1:3])

print(x)
```

```
[['France' 44.0 72000.0]
 ['Spain' 27.0 48000.0]
 ['Germany' 30.0 54000.0]
 ['Spain' 38.0 61000.0]
 ['Germany' 40.0 63777.77777777778]
 ['France' 35.0 58000.0]
 ['Spain' 38.77777777777778 52000.0]
 ['France' 48.0 79000.0]
 ['Germany' 50.0 83000.0]
 ['France' 37.0 67000.0]]
```

Encoding categorical data

Encoding the Independent Variable

```
from sklearn.compose import ColumnTransformer
from sklearn.preprocessing import OneHotEncoder
ct = ColumnTransformer(transformers=[('encoder', OneHotEncoder(), [0])], remainder='passthrough')
x = ct.fit_transform(x)
```

```
print(x)
```

```
[[0.0 1.0 0.0 0.0 44.0 72000.0]
 [1.0 0.0 0.0 1.0 27.0 48000.0]
 [1.0 0.0 1.0 0.0 30.0 54000.0]
 [1.0 0.0 0.0 1.0 38.0 61000.0]
 [1.0 0.0 1.0 0.0 40.0 63777.77777777778]
 [0.0 1.0 0.0 0.0 35.0 58000.0]
 [1.0 0.0 0.0 1.0 38.77777777777778 52000.0]
 [0.0 1.0 0.0 0.0 48.0 79000.0]
 [1.0 0.0 1.0 0.0 50.0 83000.0]
 [0.0 1.0 0.0 0.0 37.0 67000.0]]
```

Encoding the Dependent Variable

```
from sklearn.preprocessing import LabelEncoder
le = LabelEncoder()
y = le.fit_transform(y)
```

```
print('valores codificados para nuestras VARS DEPENDIENTES')
print(y)
```

```
valores codificados para nuestras VARS DEPENDIENTES
[0 1 0 0 1 1 0 1 0 1]
```

Splitting the dataset into the Training set and Test set

```
from sklearn.model_selection import train_test_split
x_train,x_test,y_train,y_test = train_test_split(x,y, test_size = 0.2, random_state = 1)
"""El 20 prociento de los valores ira a los test y el resto a el entrenamiento"""
```

```
'El 20 prociento de los valores ira a los test y el resto a el entrenamiento'
```

```
print('Impresión de los x_train')
print(x_train)
print('Impresión de los x_test')
print(x_test)
print('Impresión de los y_train')
print(y_train)
print('Impresión de los y_test')
print(y_test)
```

```
Impresión de los x_train
[[1.0 0.0 0.0 1.0 38.77777777777778 52000.0]
 [1.0 0.0 1.0 0.0 40.0 63777.77777777778]
 [0.0 1.0 0.0 0.0 44.0 72000.0]
 [1.0 0.0 0.0 1.0 38.0 61000.0]
 [1.0 0.0 0.0 1.0 27.0 48000.0]
 [0.0 1.0 0.0 0.0 48.0 79000.0]
 [1.0 0.0 1.0 0.0 50.0 83000.0]
 [0.0 1.0 0.0 0.0 35.0 58000.0]]
Impresión de los x_test
[[1.0 0.0 1.0 0.0 30.0 54000.0]
 [0.0 1.0 0.0 0.0 37.0 67000.0]]
Impresión de los y_train
[0 1 0 0 1 1 0 1]
Impresión de los y_test
[0 1]
```

Feature Scaling

```
from sklearn.preprocessing import  StandardScaler
sc = StandardScaler()
x_train[:, 3:] = sc.fit_transform(x_train[:, 3:])
x_test[:, 3:] = sc.transform(x_test[:, 3:])
```

```
print('Impresión de los x_train escalados')
print(x_train)
print()
print('Impresión de los x_test escalados')
print(x_test)
```

```
Impresión de los x_train escalados
[[1.0 0.0 0.0 1.2909944487358056 -0.19159184384578545 -1.0781259408412425]
 [1.0 0.0 1.0 -0.7745966692414834 -0.014117293757057777
  -0.07013167641635372]
 [0.0 1.0 0.0 -0.7745966692414834 0.566708506533324 0.633562432710455]
 [1.0 0.0 0.0 1.2909944487358056 -0.30453019390224867
  -0.30786617274297867]
 [1.0 0.0 0.0 1.2909944487358056 -1.9018011447007988 -1.420463615551582]
 [0.0 1.0 0.0 -0.7745966692414834 1.1475343068237058 1.232653363453549]
 [1.0 0.0 1.0 -0.7745966692414834 1.4379472069688968 1.5749910381638885]
 [0.0 1.0 0.0 -0.7745966692414834 -0.7401495441200351 -0.5646194287757332]]

Impresión de los x_test escalados
[[1.0 0.0 1.0 -0.7745966692414834 -1.4661817944830124 -0.9069571034860727]
 [0.0 1.0 0.0 -0.7745966692414834 -0.44973664397484414 0.2056403393225306]]
```

Artefacto: Formas de presentar la data.

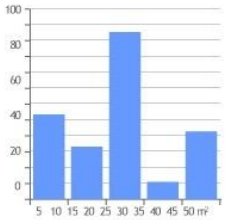


Gráfico de columna

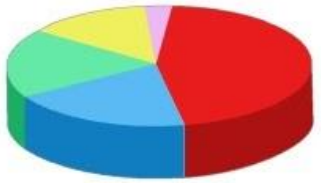


Gráfico por sectores

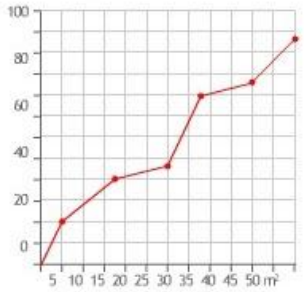


Gráfico lineal

- ¿Cuánto saben sobre el tema?
  - El docente es consciente de lo que requiere.
- ¿Cuánta información van a necesitar?

- ¿Qué datos recordarán?
  - Los datos deben ser presentados de forma minimalista.

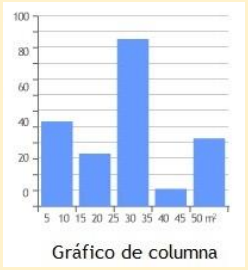


Gráfico de columna

•Comparación de promedios.
•Comparación de rendimiento
•Comparación de peores/mejores
•Comparación de grupos
•Comparación de cuestionarios
•Comparación directa de resultados de un alumno

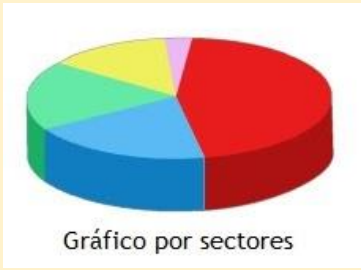


Gráfico por sectores

•Comparación de promedios.
•Comparación de rendimiento
•Comparación de peores/mejores
•Comparación de grupos
•Comparación de cuestionarios
•Comparación directa de resultados de un alumno

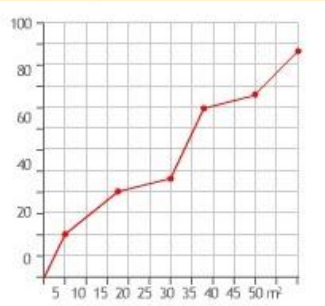


Gráfico lineal

•Determinación de tendencias
  •Grupos
  •Alumnos
  •Cuestionarios (resultados historicos)
•Tendencias de la comunidad
  •Veces respondido