



UNIVERSIDAD DE CHILE

# Minería de Datos

Welcome to the Machine Learning class

---

Valentin Barriere

Universidad de Chile – DCC

CC5205, Otoño 2024

# Datos

# Esquema: Datos

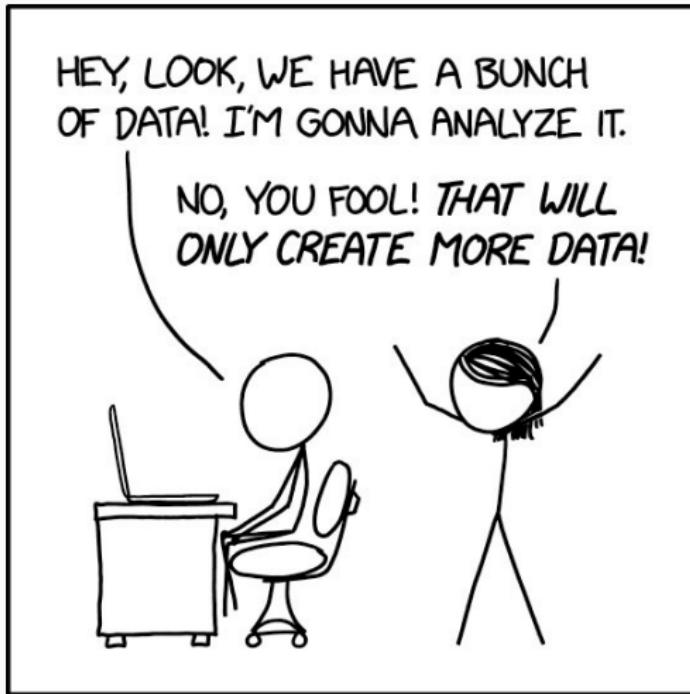
Generalidades

Tipos de datos

Calidad

Limpiar y pre-processar

## Datos



# Outline : Generalidades

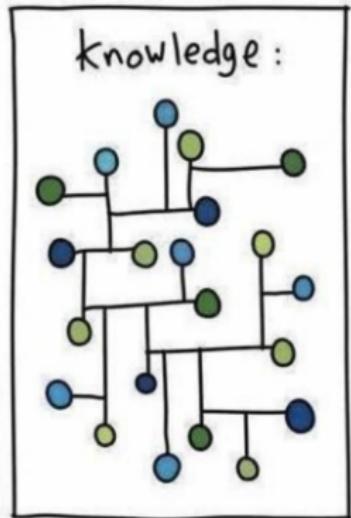
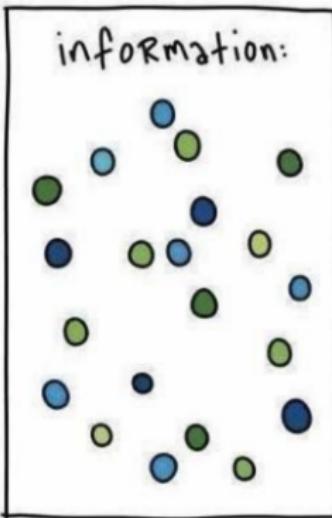
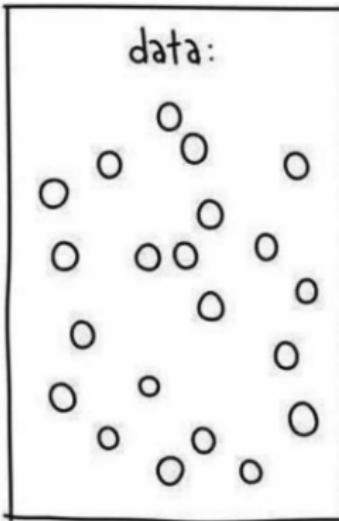
Generalidades

Tipos de datos

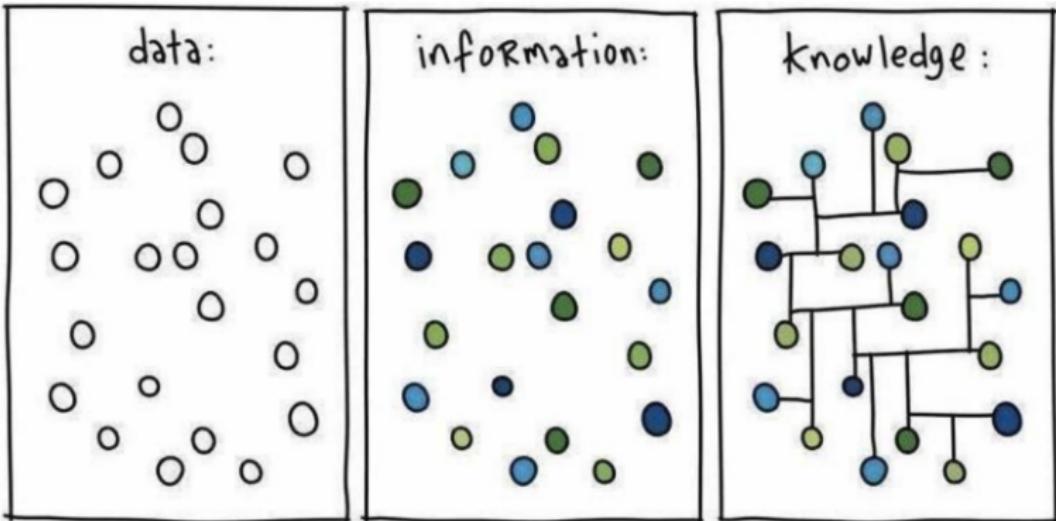
Calidad

Limpiar y pre-processar

# Datos: Que entendemos por datos?



# Datos: Que entendemos por datos?

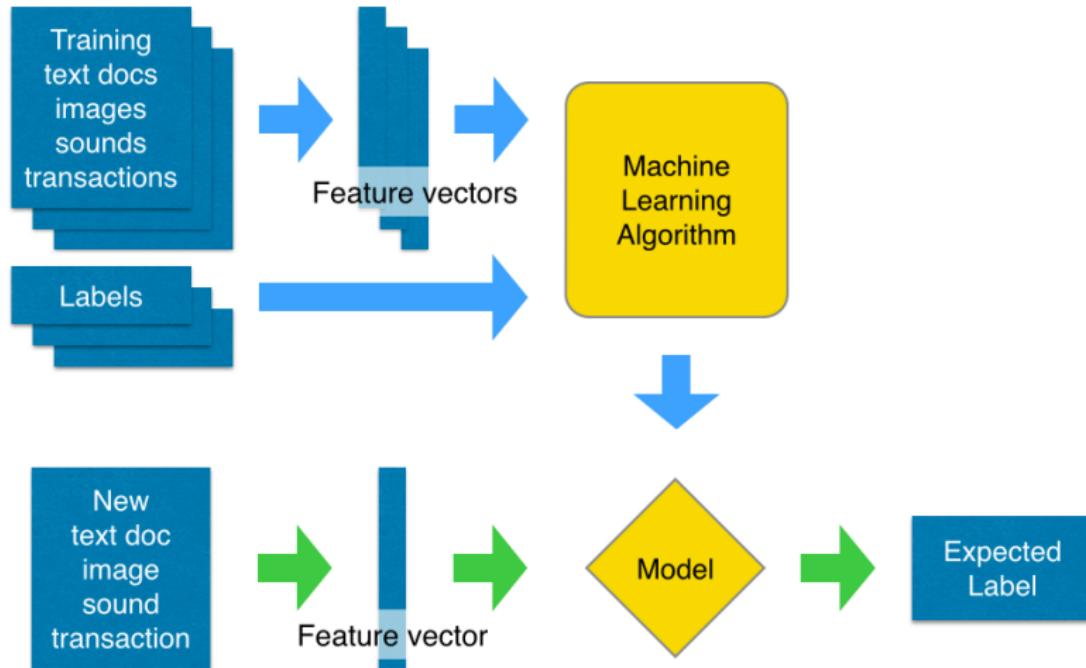


Juan Pérez obtuvo un 5.8 en la segunda prueba de historia

Juan Pérez obtuvo mejor nota que el 75% del resto del curso

Juan Pérez es un buen alumno

# Aprendizaje supervisado



Predictive Modeling Data Flow

## Scikit-learn: biblioteca de ML en Python

Biblioteca de Aprendizaje Automático muy fácil de usar:

<https://scikit-learn.org/>

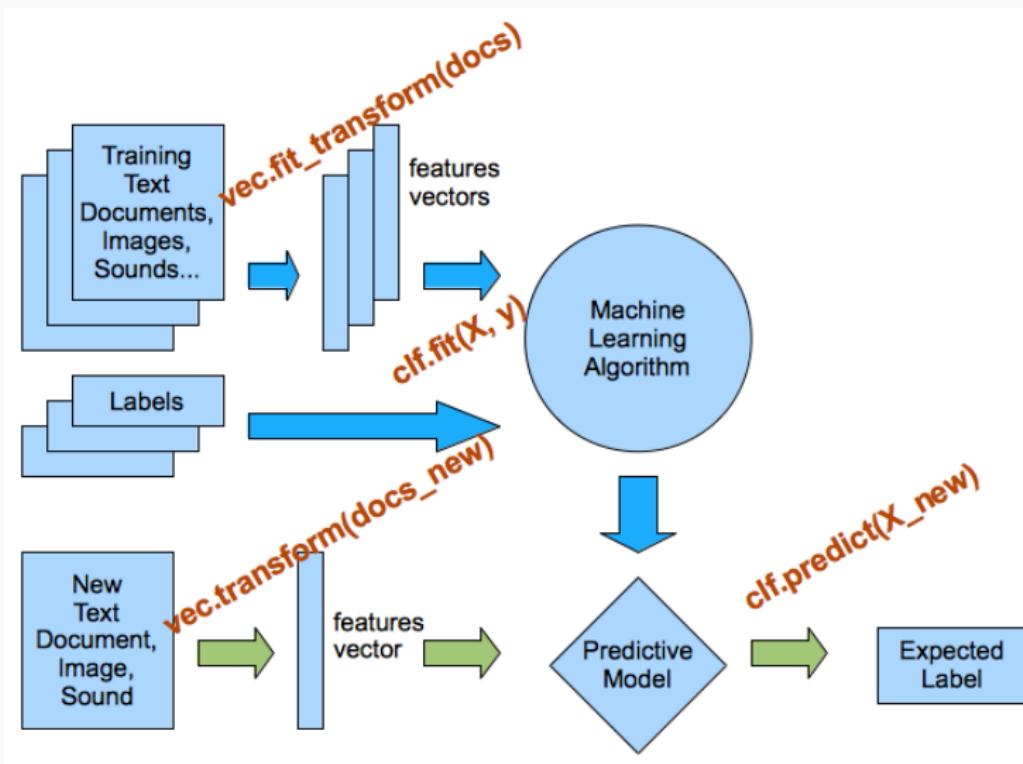


Un manual de usor bacan:

[https://scikit-learn.org/stable/user\\_guide.html](https://scikit-learn.org/stable/user_guide.html)

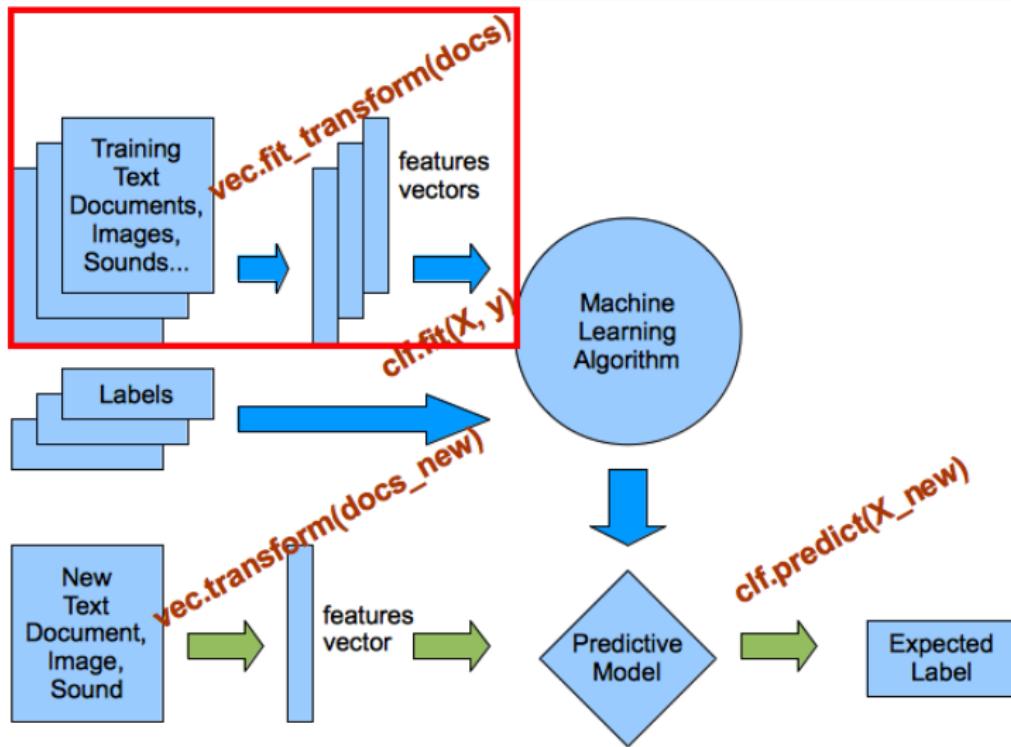
# Scikit-learn: workflow general

Funciones normalizadas para apprender rápido con claridad en el código:



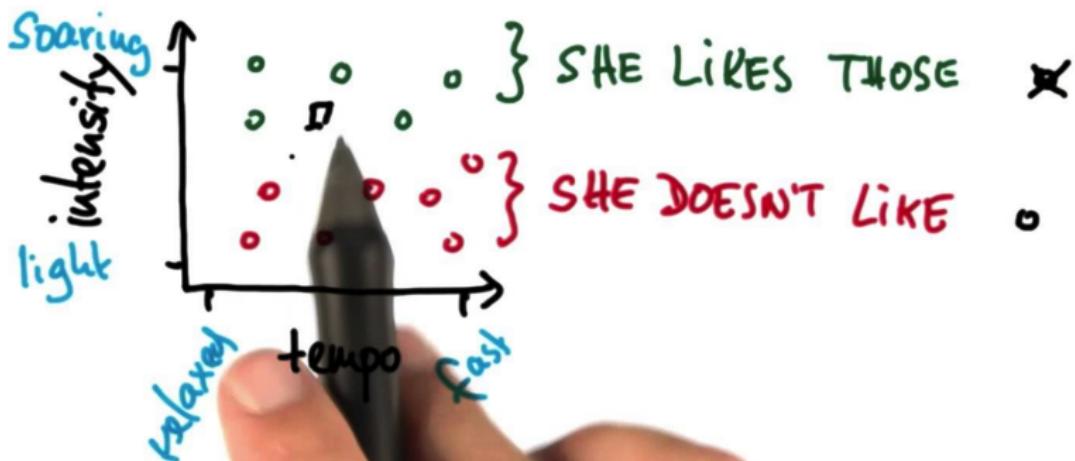
# Scikit-learn: workflow general

Funciones normalizadas para apprender rápido con claridad en el código:



## Descriptores y etiquetas

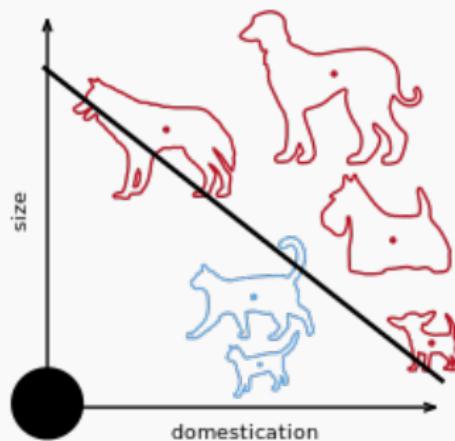
FEATURES AND LABELS



# Aprendizaje automático y Reconocimiento de formas

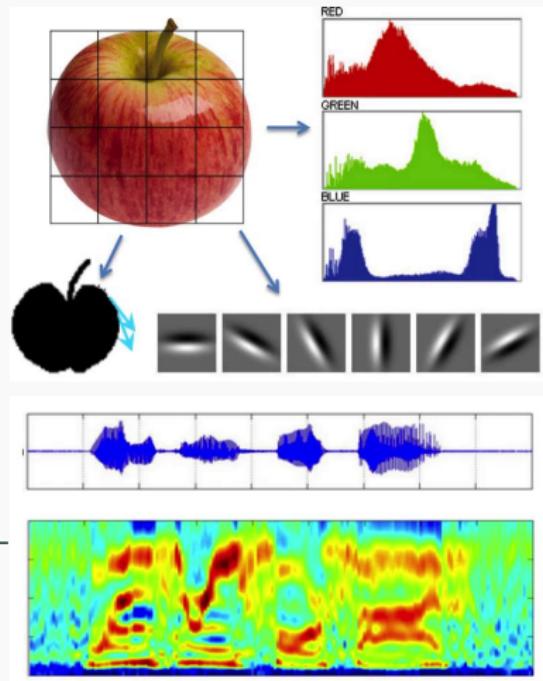
## Vectorización

- Para detectar estructuras en un espacio, es necesario que los datos estén en forma de vectores, es decir, un conjunto de variables numéricas (una por dimensión).
- Se necesita **extraer representaciones cifradas** de cada uno de los documentos para transformar los documentos en vectores.



# Descriptores

- Imágenes: Histograma de colores



- Sonidos: Representación tiempo-frecuencia

# Datos tabulares

Hoy, vamos a hablar principalmente de **datos estructuradas**, i.e. en la forma de datos tabulares.

- Colección de hechos tales como números, palabras, mediciones, o solo descripción de cosas.
- Utilizado generalmente para análisis.

# Outline : Tipos de datos

Generalidades

Tipos de datos

Calidad

Limpiar y pre-processar

# Datos cuantitativos vs. cualitativos

Data

Qualitative  
Descriptive information

"I drink coffee every day"

Quantitative

Numerical information

Discrete  
(Counted)



"I drink 4 coffees every day"

Continuous  
(Measured)



"I drink 80grs of coffee every day"

Cuales son los intereses de datos cualitativos vs cuantitativos?

# Datos cuantitativos vs. cualitativos



Cuales son los intereses de datos cualitativos vs cuantitativos?

Mas facil de interpretar (pueden estar higher-level) pero perdida de informacion

# Un ejemplo: Titanic dataset

PassengerId	Survived	Pclass	Name	Sex	Age	SibSp	Parch	Ticket	Fare	Cabin	Embarked
0	1	0	Braund, Mr. Owen Harris	male	22.0	1	0	A/5 21171	7.2500	NaN	S
1	2	1	Cumings, Mrs. John Bradley (Florence Briggs Th... Cumings, Mrs. John Bradley (Florence Briggs Th...)	female	38.0	1	0	PC 17599	71.2833	C85	C
2	3	1	Heikkinen, Miss. Laina	female	26.0	0	0	STON/O2. 3101282	7.9250	NaN	S
3	4	1	Futrelle, Mrs. Jacques Heath (Lily May Peel) Futrelle, Mrs. Jacques Heath (Lily May Peel)	female	35.0	1	0	113803	53.1000	C123	S
4	5	0	Allen, Mr. William Henry	male	35.0	0	0	373450	8.0500	NaN	S

# Tipos de datos tabulares

Attribute Type	Description	Examples	Operations
Categorical (Qualitative)	Nominal  The values of a nominal attribute are just different names; i.e., nominal values provide only enough information to distinguish one object from another. $(=, \neq)$	zip codes, employee ID numbers, eye color, gender	mode, entropy, contingency correlation, $\chi^2$ test
	Ordinal  The values of an ordinal attribute provide enough information to order objects. $(<, >)$	hardness of minerals, $\{good, better, best\}$ , grades, street numbers	median, percentiles, rank correlation, run tests, sign tests
Numeric (Quantitative)	Interval  For interval attributes, the differences between values are meaningful, i.e., a unit of measurement exists. $(+, -)$	calendar dates, temperature in Celsius or Fahrenheit	mean, standard deviation, Pearson's correlation, $t$ and $F$ tests
	Ratio  For ratio variables, both differences and ratios are meaningful. $(\times, /)$	temperature in Kelvin, monetary quantities, counts, age, mass, length, electrical current	geometric mean, harmonic mean, percent variation

# Extracción de atributos

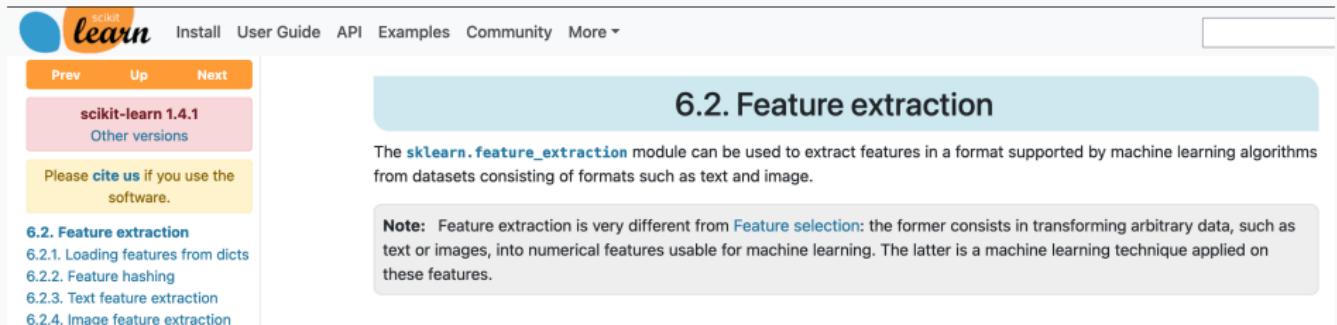
## El concepto de transformar los datos en vectores

Diferentes tipos de extracción posible:

- es nominal (una clase, multclases: un tag)
- es un valor ordinal (normalización)

```
>>> measurements = [  
...     {'city': 'Dubai', 'temperature': 33.},  
...     {'city': 'London', 'temperature': 12.},  
...     {'city': 'San Francisco', 'temperature': 18.},  
... ]  
  
>>> from sklearn.feature_extraction import DictVectorizer  
>>> vec = DictVectorizer()  
  
>>> vec.fit_transform(measurements).toarray()  
array([[ 1.,  0.,  0., 33.],  
       [ 0.,  1.,  0., 12.],  
       [ 0.,  0.,  1., 18.]])  
  
>>> vec.get_feature_names_out()  
array(['city=Dubai', 'city=London', 'city=San Francisco', 'temperature'], ...)
```

# Extraccion de atributos en scikit



The screenshot shows the scikit-learn documentation website. At the top, there's a navigation bar with a blue circle icon, the "scikit-learn" logo, and links for "Install", "User Guide", "API", "Examples", "Community", and "More". Below the navigation bar, there's a horizontal menu with "Prev", "Up", and "Next" buttons. A pink sidebar on the left contains the text "scikit-learn 1.4.1" and "Other versions", followed by a yellow box with the text "Please cite us if you use the software." The main content area has a light blue header with the title "6.2. Feature extraction". The text below the header states: "The `sklearn.feature_extraction` module can be used to extract features in a format supported by machine learning algorithms from datasets consisting of formats such as text and image." A note box below this text says: "Note: Feature extraction is very different from `Feature selection`: the former consists in transforming arbitrary data, such as text or images, into numerical features usable for machine learning. The latter is a machine learning technique applied on these features."

scikit-learn 1.4.1  
Other versions

Please cite us if you use the software.

6.2. Feature extraction

- 6.2.1. Loading features from dicts
- 6.2.2. Feature hashing
- 6.2.3. Text feature extraction
- 6.2.4. Image feature extraction

## 6.2. Feature extraction

The `sklearn.feature_extraction` module can be used to extract features in a format supported by machine learning algorithms from datasets consisting of formats such as text and image.

**Note:** Feature extraction is very different from `Feature selection`: the former consists in transforming arbitrary data, such as text or images, into numerical features usable for machine learning. The latter is a machine learning technique applied on these features.

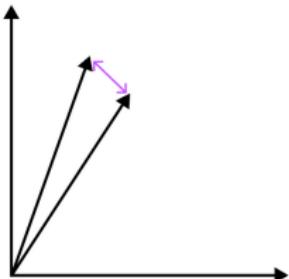
Clase de scikit-learn:

[https://scikit-learn.org/stable/modules/feature\\_extraction.html](https://scikit-learn.org/stable/modules/feature_extraction.html)

# Distancias y similaridades

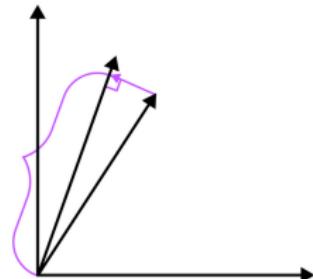
Euclidean Distance

$$d(\mathbf{p}, \mathbf{q}) = \sqrt{\sum_{i=1}^n (q_i - p_i)^2}$$



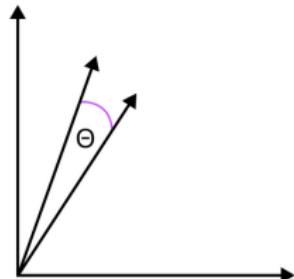
Inner Product

$$\mathbf{a} \cdot \mathbf{b} = \sum_{i=1}^n a_i b_i$$



Cosine Similarity

$$\cos(\theta) = \frac{\mathbf{A} \cdot \mathbf{B}}{\|\mathbf{A}\| \|\mathbf{B}\|} = \frac{\sum_{i=1}^n A_i B_i}{\sqrt{\sum_{i=1}^n A_i^2} \sqrt{\sum_{i=1}^n B_i^2}}$$



# Correlación de Pearson y producto escalar

Se puede hacer una medida de similitud con un producto escalar:

$$\cos(\mathbf{X}, \mathbf{X}') = \frac{\langle \mathbf{X} | \mathbf{X}' \rangle}{\|\mathbf{X}\| \cdot \|\mathbf{X}'\|} = \frac{1}{\|\mathbf{X}\| \cdot \|\mathbf{X}'\|} \sum_{i=1}^N X_i * X'_i$$

## Big Five Personality Traits

5 dimensiones: Apertura a la experiencia, Conciencia, Extraversión, Amabilidad, y Neuroticismo. Note desde -1 hasta 1 en cada dimensión.  
Mas info [sobre el Big 5](#)

## User Recommendation

Cada usuario será representado como un vector de  $N$  dimensiones, siendo cada dimensión un producto. Por cada producto comprado/gustado/visto, el vector va a tener un 1 en la dimensión.

Correlación de Pearson: coseno de los datos centrados:  $\cos(\mathbf{X}_{\text{cent}}, \mathbf{X}'_{\text{cent}})$

# Correlación de Pearson y producto escalar

Se puede hacer una medida de similitud con un producto escalar:

$$\cos(\mathbf{X}, \mathbf{X}') = \frac{\langle \mathbf{X} | \mathbf{X}' \rangle}{\|\mathbf{X}\| \cdot \|\mathbf{X}'\|} = \frac{1}{\|\mathbf{X}\| \cdot \|\mathbf{X}'\|} \sum_{i=1}^N X_i * X'_i$$

## Big Five Personality Traits

5 dimensiones: Apertura a la experiencia, Conciencia, Extraversión, Amabilidad, y Neuroticismo. Note desde -1 hasta 1 en cada dimensión.  
Mas info [sobre el Big 5](#)

## User Recommendation

Cada usuario será representado como un vector de  $N$  dimensiones, siendo cada dimensión un producto. Por cada producto comprado/gustado/visto, el vector va a tener un 1 en la dimensión.

Correlación de Pearson: cosinus de los datos centrados:  $\cos(\mathbf{X}_{\text{cent}}, \mathbf{X}'_{\text{cent}})$

**Piensen a más, que podríamos hacer sobre el texto?**

## Datos non structurados

### Structured Data



What you find in a DB  
(typically)

### Unstructured Data



What you find in the 'wild'  
(text, images, audio, video)

# Datos non structurados

Nombre mascota	Especie	Raza	Edad	Sexo	Peso
Doris	Canino	Schnauzer	3	Hembra	8
Clotilde	Canino	Mestizo	14	Hembra	7



Se ha ingresado el paciente **Doris**, de especie **canino**. Nació aproximadamente el 2017, por lo que tiene **3 años**. Su manto es color **sal y pimienta** y patas blancas. Este es su primer control, pesando **8kg** aproximadamente.



# Características Generales

## Dimensionalidad

nro. de atributos, maldición de la dimensionalidad (curse of dimensionality) tiene que ver con problemas al trabajar con muchas dimensiones (preprocesamiento: reducción de dimensionalidad)

## Dispersión

mayoría de las dimensiones son 0 para los datos, puede tener ventajas, como no necesitar almacenar los valores 0, sólo los 1s. (Ej. Grafo de la Web y sus enlaces).

## Resolución

El nombre habla por ello (ej. variaciones de presión atmosférica en horas es notoria, pero en meses no se detecta).

# Outline : Calidad

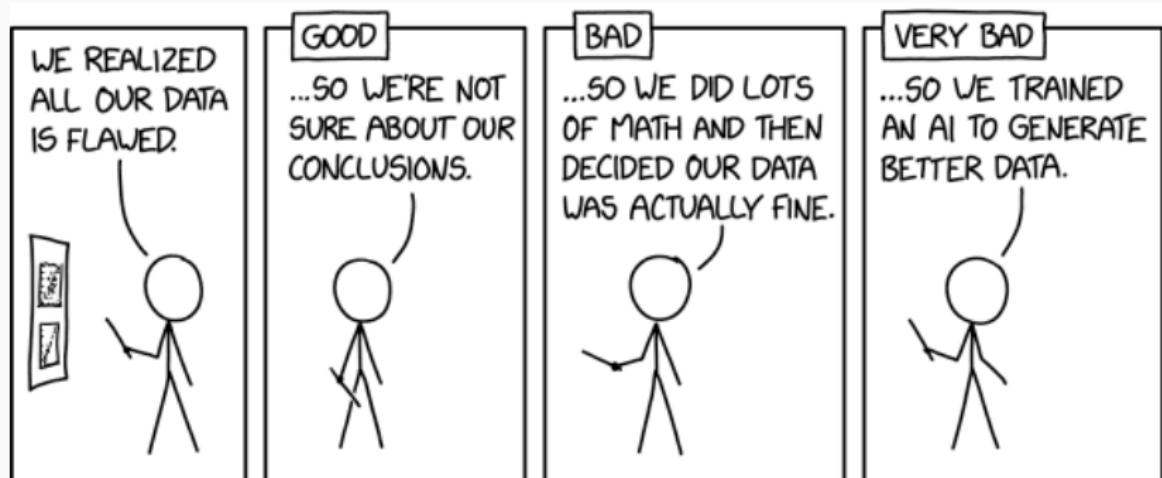
Generalidades

Tipos de datos

**Calidad**

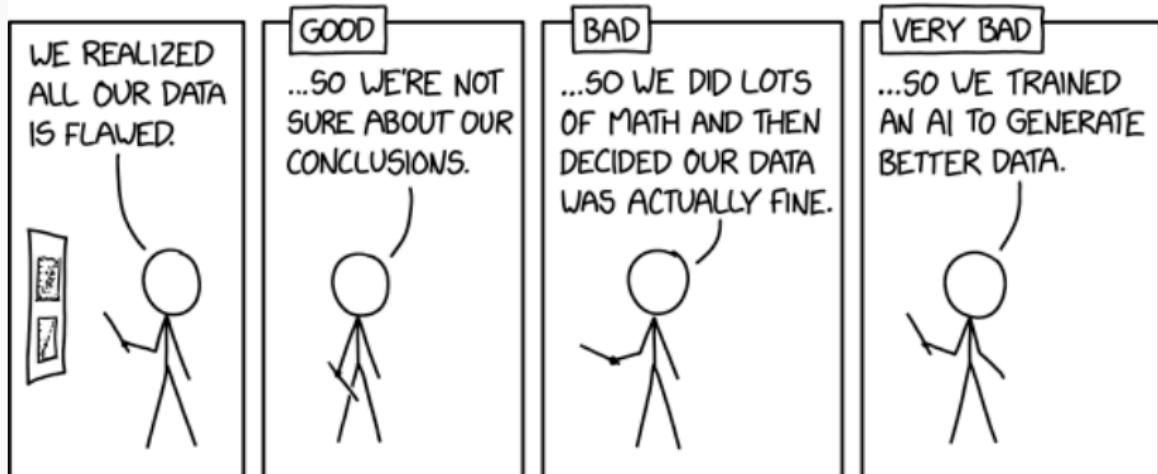
Limpiar y pre-processar

# Calidad de los datos



Tipos de errores: Ruido y outliers, Valores faltantes, Datos duplicados

# Calidad de los datos



Tipos de errores: Ruido y outliers, Valores faltantes, Datos duplicados

**Hay soluciones!**

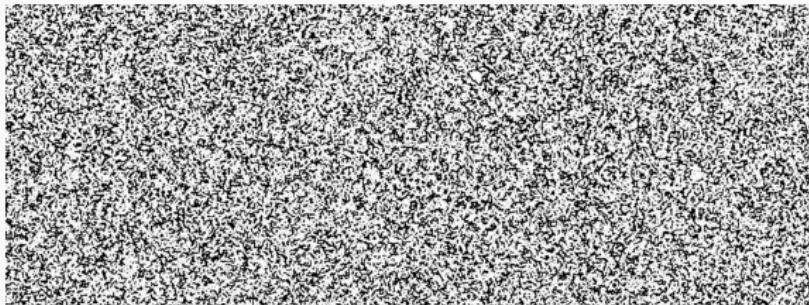
# Ruido estadístico

## Definición

Irregularidad **aleatoria** que encontramos en cualquier dato de la **vida real**. No tienen ningún patrón. Estos errores suelen ser inevitables e imprevisibles.

Consiste generalmente en errores y residuos:

- Los errores pueden ser de medición o de muestreo, son evitables y se repiten, distorsionando los datos. Viene de los datos.
- El residuo viene de la modelización de los datos



## Residuo

El residuo de los datos observados es la diferencia entre tu valor observado (una vez más, ese punto de datos que has medido) y el valor predicho; no el "valor verdadero" en sí mismo, sino el punto en el espacio que tu teoría te dice que el punto de datos debería estar.

- Es difícil (o imposible) de representar toda la realidad con un ensamble finito de observaciones
- Vamos a representar una cosa con un vector de tamaño finito, lo que puede ser reductible al fenómeno inicial, es una aproximación de la realidad
- **Se va a quedar un componente de ruido que no se puede modelizar**

## Ejemplo de ruido

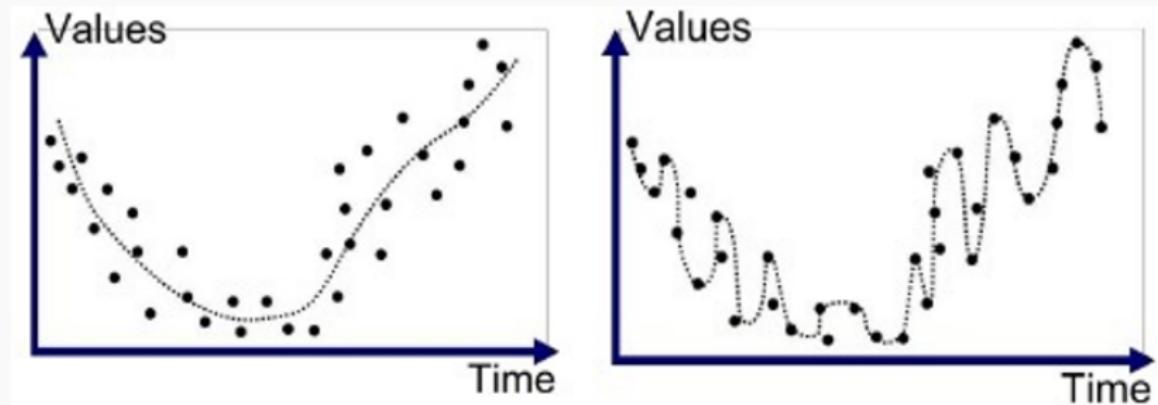


Figure 1: Ruido en una función

Si queremos modelizar  $Y = 3 * X_1 - 2 * X_2^2 + \epsilon$  con  $X_1$  y  $X_2$ , no lo vamos a lograr.

## Denoising Example

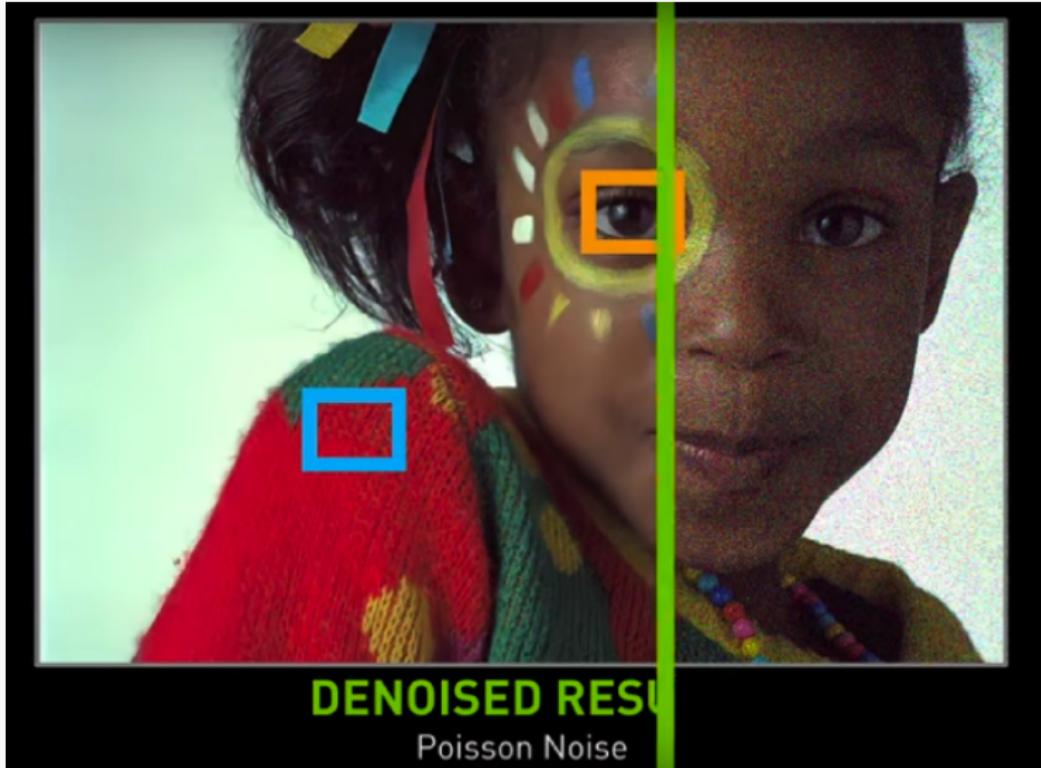
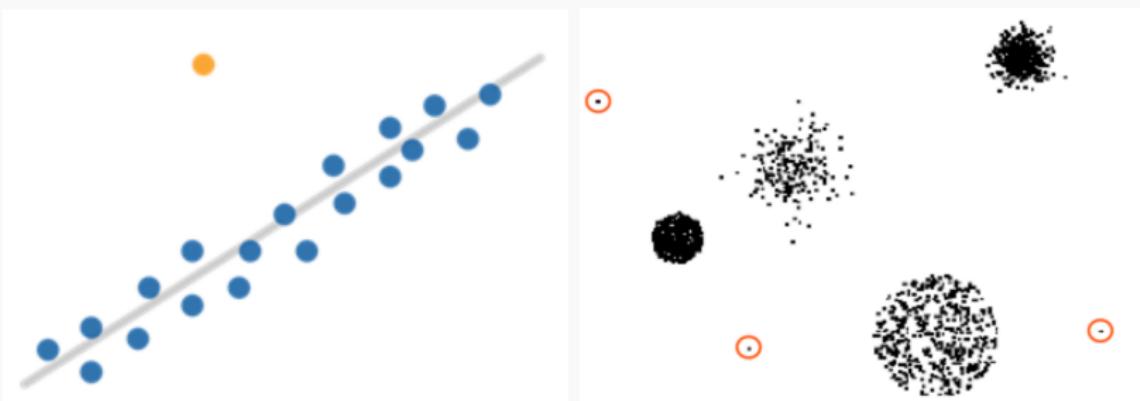


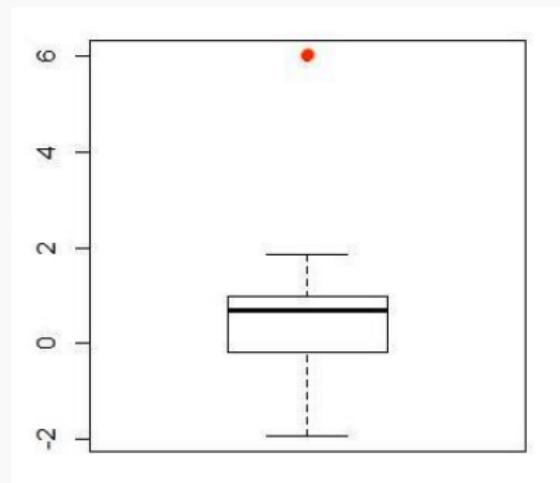
Figure 2: Una aplicación de [Image Denoiser](#)

# Outliers



**Figure 3:** Outliers data points en varias distribuciones

# Outliers



**Figure 3:** Outliers data points en varias distribuciones

Objetos con características considerablemente diferentes a la mayoría

## Valores atípicos ruidosos

- Variabilidad natural: cola de la distribución
- Errores de medición: malo sensor, tipografía, mala entrada, ...
- Eventos excepcionales: eventos inusuales o extremos que no siguen el patrón típico de los datos (y no es interesante de modelizar este distribución)

## Valores atípicos "buenos"

Evento raro tipo anomalía que queremos predecir! Terremoto, crisis epiléptica, mantención de machina, anomalía en un producto, etc...

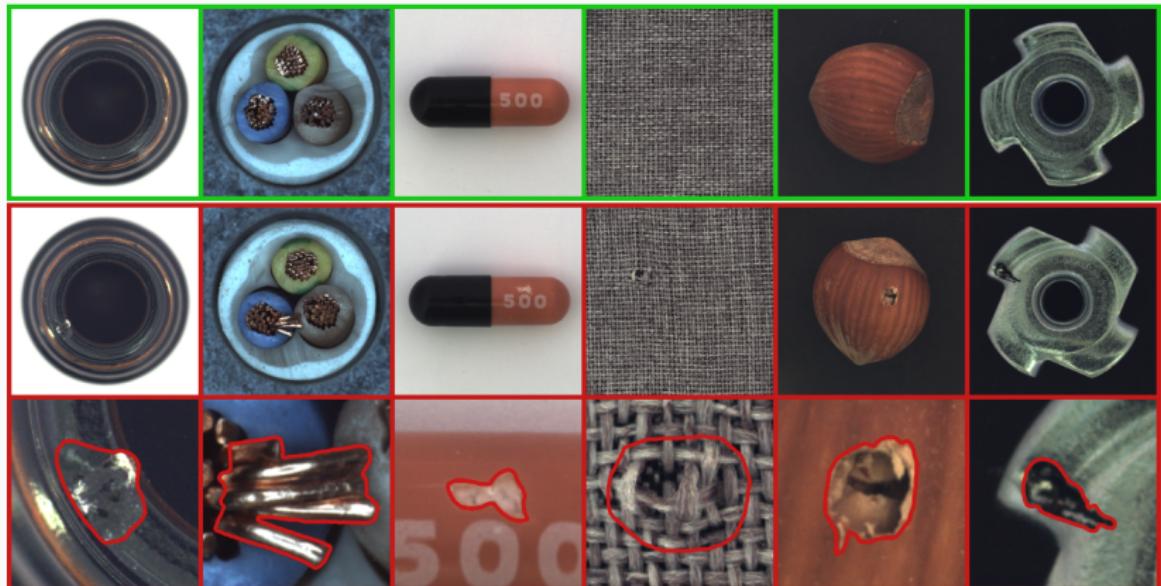
## Detectar un outlier

---

- Visualización de datos (débil)
- Métodos estadísticos: como el rango intercuartílico (IQR), la desviación estándar o los z-scores
- Técnicas de aprendizaje automático: Algoritmos de detección de anomalías
- Métodos basados en el modelo: Mirar los puntos donde hay el error es grande
- Técnicas de agrupamiento: Los valores atípicos pueden surgir como puntos que no se agrupan bien con otros puntos en un algoritmo de agrupamiento. Observar los puntos que quedan solos o están en grupos pequeños puede ayudar a identificar los outliers
- Validación del dominio: **la validación del dominio es crucial.** Algunas veces, lo que parece un outlier puede ser en realidad un dato válido pero poco común. Es importante considerar el contexto y la naturaleza de los datos al interpretar los valores atípicos detectados

# Detección de anomalía

Puede ser mas preciso que detectar un sample entero, pero solo un parte de este ejemplo



**Figure 4:** Ejemplos del MVTEC Anomaly Detection Dataset

# CleanLab va a detectar errores en labels



## Cleanlab Open-Source

cleanlab helps you **clean** data and **labels** by automatically detecting issues in a ML dataset. To facilitate **machine learning with messy, real-world data**, this data-centric AI package uses your **existing** models to estimate dataset problems that can be fixed to train even *better* models.

```
# cleanlab works with **any classifier**. Yup, you can use PyTorch/TensorFlow/OpenAI/XGBoost/etc
cl = cleanlab.classification.CleanLearning(sklearn.YourFavoriteClassifier())

# cleanlab finds data and label issues in **any dataset**... in ONE line of code!
label_issues = cl.find_label_issues(data, labels)

# cleanlab trains a robust version of your model that works more reliably with noisy data.
cl.fit(data, labels)

# cleanlab estimates the predictions you would have gotten if you had trained with *no* label
cl.predict(test_data)

# A universal data-centric AI tool, cleanlab quantifies class-level issues and overall data quality
cleanlab.dataset.health_summary(labels, confident_joint=cl.confident_joint)
```

**Figure 5:** Outlier Label detection. CleanLab helped to detect many of the label error in Imagenet

## Valores faltantes

---

There are often missing values, and that you have to take care of it:

- Eliminando el objeto
- Estimando valores: por una constante, por media, utilizando los de los vecinos (KNN), u otros (aprendiendo los)
- Ignorar: hay algoritmos que pueden manejar valores faltantes

<https://scikit-learn.org/stable/modules/impute.html>

## Valores faltantes: con scikit

Funciones SimpleImputer, o KNNImputer:

```
>>> import numpy as np
>>> from sklearn.impute import SimpleImputer
>>> imp_mean = SimpleImputer(missing_values=np.nan, strategy='mean')
>>> # imp_mean = SimpleImputer(missing_values=np.nan,
# strategy='constant', fill_value=0)
>>> imp_mean.fit([[7, 2, 3], [4, np.nan, 6], [10, 5, 9]])
SimpleImputer()
>>> X = [[np.nan, 2, 3], [4, np.nan, 6], [10, np.nan, 9]]
>>> print(imp_mean.transform(X))
[[ 7.   2.   3. ]
 [ 4.   3.5  6. ]
 [10.  3.5  9. ]]
```

KNNImputer va a tomar los valores de los atributos de los  $N$  ejemplos los mas cercanos en el espacio de representación

# Valores faltantes: estimadores que les manejan

## 6.4.7. Estimators that handle NaN values

Some estimators are designed to handle NaN values without preprocessing. Below is the list of these estimators, classified by type (cluster, regressor, classifier, transform):

- **Estimators that allow NaN values for type cluster:**
  - [HDBSCAN](#)
- **Estimators that allow NaN values for type regressor:**
  - [BaggingRegressor](#)
  - [DecisionTreeRegressor](#)
  - [HistGradientBoostingRegressor](#)
  - [RandomForestRegressor](#)
  - [StackingRegressor](#)
  - [VotingRegressor](#)
- **Estimators that allow NaN values for type classifier:**
  - [BaggingClassifier](#)
  - [DecisionTreeClassifier](#)
  - [HistGradientBoostingClassifier](#)
  - [RandomForestClassifier](#)
  - [StackingClassifier](#)
  - [VotingClassifier](#)

**Figure 6:** Existen modelos manejando los missing values

## Valores faltantes: un ejemplo

Podemos utilizar algoritmos complejos para reconstruir una muestra con valores faltantes

## Valores faltantes: un ejemplo

Podemos utilizar algoritmos complejos para reconstruir una muestra con valores faltantes: **tienen ideas?**

## Valores faltantes: un ejemplo

Podemos utilizar algoritmos complejos para reconstruir una muestra con valores faltantes: tienen ideas?



Figure 7: Un modelo de Image Reconstruction basando en redes neuronales

## Datos duplicados

---

Es muy común de encontrar datos duplicados porque:

- Error humano durante la entrada de datos.
- Este es la realidad (pero no necesariamente sirve)
- Hay una falla en el sistema de recolección: herramientas de extracción, transformación y carga
- Agregación de diferentes bases de datos que tienen duplicaciones

Necesitan controles de calidad de los datos (al tema de las duplicaciones) durante la recolección y procesamiento.

# Use case: Large Language Models

## Scraping the whole web!

Para entrenar un LLM desde zero, es necesario de colectar una grande cantidad de datos! Eso no puede ser datos limpios!

- Data scrapped from the web is pretty noisy and you need to clean it
- Markups, syntax breaks, etc... all that gives non NL text is harmful, and can prevent convergence!
- Deduplication has been identified as playing a significant role in improving language models (Allamanis, 2019; Lee et al., 2022)
- Repeated data has been shown to be increasingly harmful to model quality as parameter count increases (Hernandez et al., 2022):
  - for a 1B parameters model, a hundred duplicates are harmful;
  - at 175B, even a few duplicates could have a disproportionate effect.

# Outline : Limpieza y pre-processar

---

Generalidades

Tipos de datos

Calidad

Limpieza y pre-processar

# Pre-procesamiento de datos

---

- Creación de atributos
- Selección de un subconjunto de atributos
- Agregación
- Normalización
- Muestreo
- Reducción de dimensionalidad
- Discretización y binarización
- Transformación

# Transformaciones de los datos: sobre la distribucion

## Standardization

Estimator might behave badly if the individual features do not more or less look like standard normally distributed data: Gaussian with **zero mean and unit variance**

## Scaling features to a range

Generally between 0 and 1. The motivation to use this scaling include robustness to very small standard deviations of features and preserving zero entries in sparse data.

## Normalization

Normalization is the process of **scaling individual samples to have unit norm**. Por ejemplo

# Transformaciones de los datos: Histogram Stretching

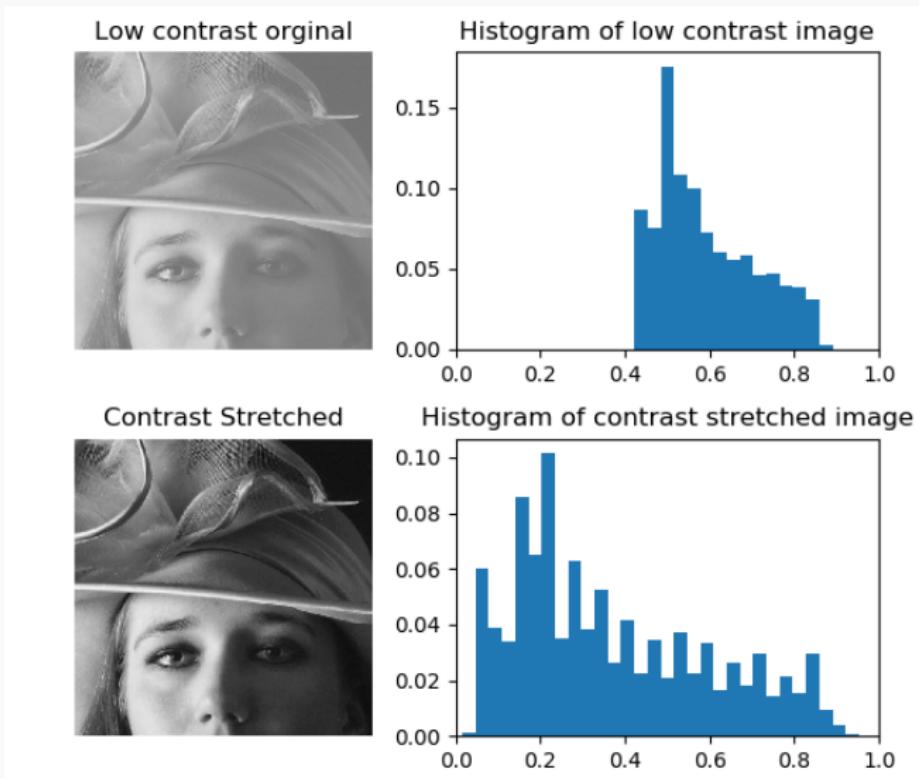


Figure 8: Cambiar el rango de los valores del histograma ayuda a ver la imagen

# Transformaciones de los datos: otras

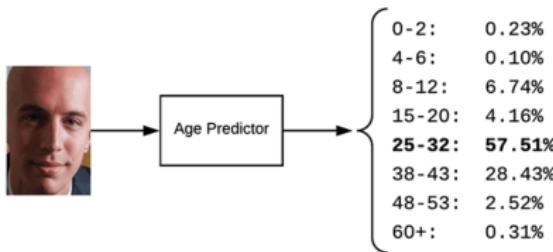
## Discretizacion

Discretization (otherwise known as quantization or binning) provides a way to partition continuous features into discrete values (classes). One-hot encoded discretized features can make a model more expressive, while maintaining interpretability.

### Age Prediction via Regression



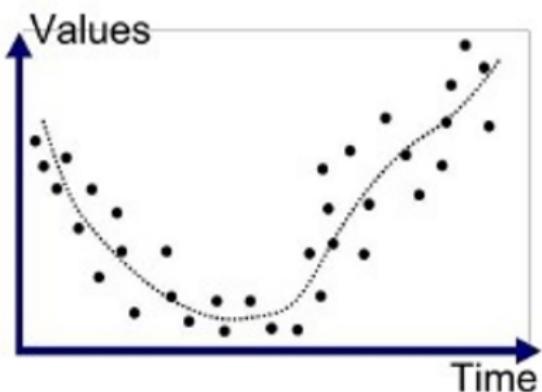
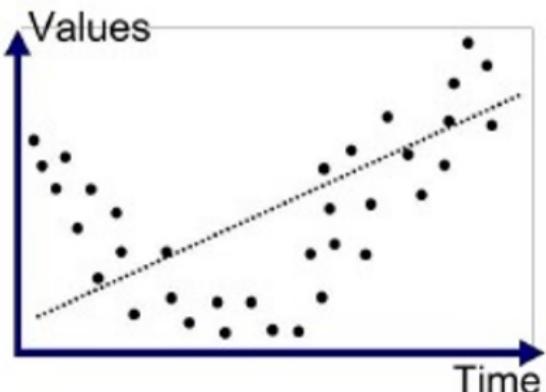
### Age Prediction via Classification



# Transformaciones de los datos: otras

## Polynomial features

Often it's useful to add complexity to a model by considering nonlinear features of the input data.



Mas informaciones [aca](#)

# Transformaciones de los datos: ONEHOTENCODER

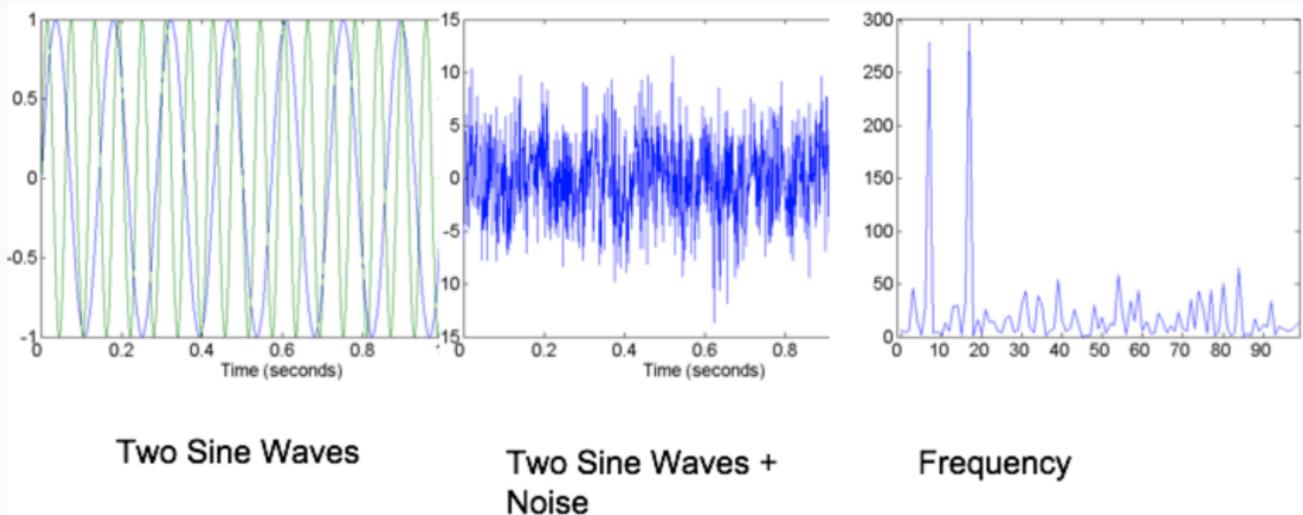
```
>>> genders = ['female', 'male']
>>> locations = ['from Africa', 'from Asia', 'from Europe', 'from US']
>>> browsers = ['uses Chrome', 'uses Firefox', 'uses IE', 'uses Safari']
>>> enc = preprocessing.OneHotEncoder(categories=[genders, locations, browsers])
>>> # Note that there are missing categorical values for the 2nd and 3rd
>>> # feature
>>> X = [['male', 'from US', 'uses Safari'],
>>> ['female', 'from Europe', 'uses Firefox']]
>>> enc.fit(X)
OneHotEncoder(categories=[[['female', 'male'],
                           ['from Africa', 'from Asia', 'from Europe',
                            'from US'],
                           ['uses Chrome', 'uses Firefox', 'uses IE',
                            'uses Safari']]])
>>> enc.transform([['female', 'from Asia', 'uses Chrome']]).toarray()
array([[1., 0., 0., 1., 0., 0., 1., 0., 0., 0.]])
```

Porque:

```
>>> enc.categories_
[array(['female', 'male'], dtype=object),
 array(['from Europe', 'from US'], dtype=object),
 array(['uses Firefox', 'uses Safari'], dtype=object)]
```

Mas sobre el encoding de atributos categoricales

# Transformaciones especiales I



**Figure 9:** Mapear a un nuevo espacio que tiene mas sentido puede ayudar a caracterizar mejor los datos

## 6.3.8. Custom transformers

Often, you will want to convert an existing Python function into a transformer to assist in data cleaning or processing. You can implement a transformer from an arbitrary function with `FunctionTransformer`. For example, to build a transformer that applies a log transformation in a pipeline, do:

```
>>> import numpy as np
>>> from sklearn.preprocessing import FunctionTransformer
>>> transformer = FunctionTransformer(np.log1p, validate=True)
>>> X = np.array([[0, 1], [2, 3]])
>>> # Since FunctionTransformer is no-op during fit, we can call transform directly
>>> transformer.transform(X)
array([[0.          ,  0.69314718],
       [1.09861229,  1.38629436]])
```

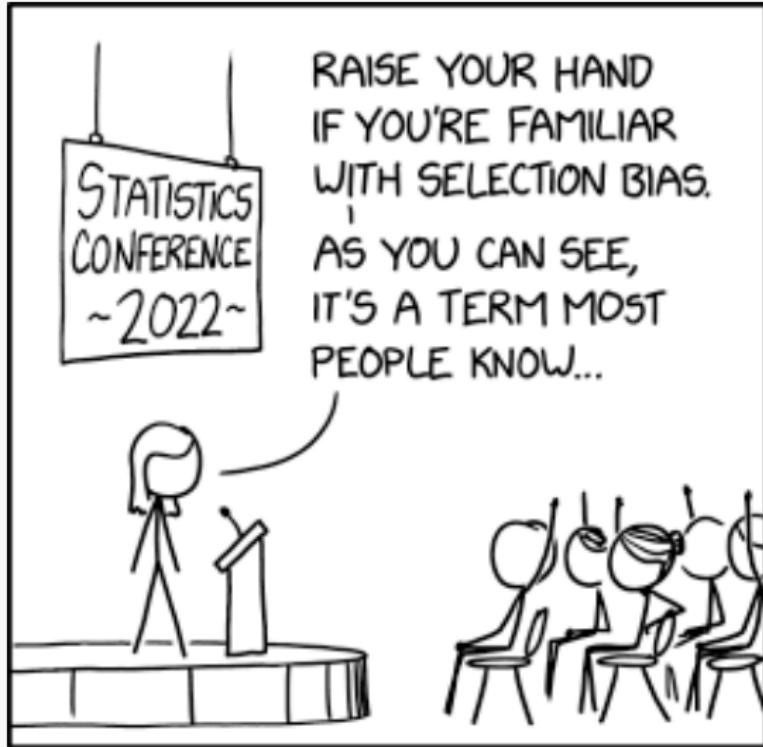
Figure 10: Se puede utilizar cualquier cosa para transformar los datos

<https://scikit-learn.org/stable/modules/preprocessing.html#custom-transformers>

De la misma manera:

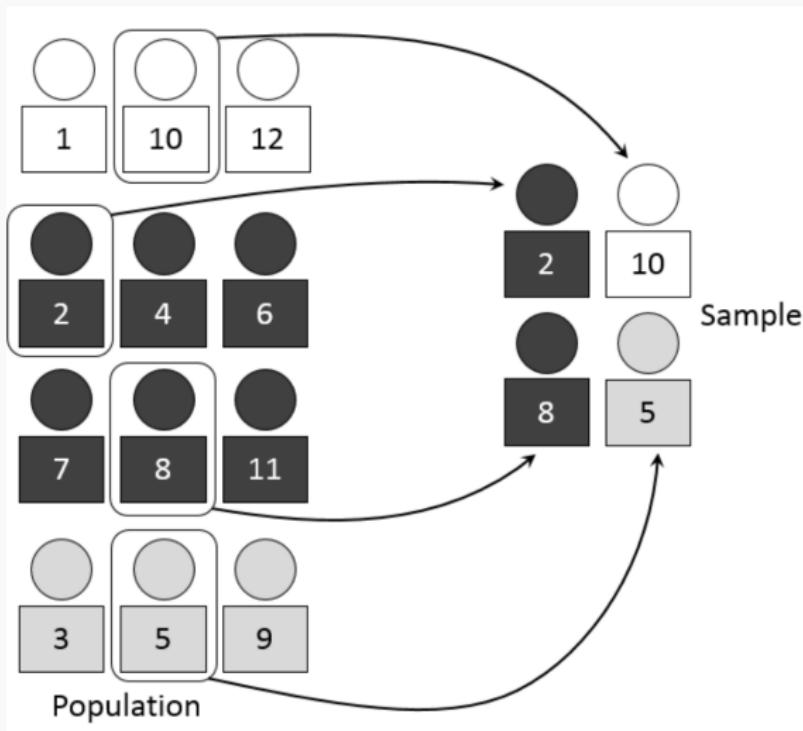
- Si las observaciones que tenemos no están suficientes para representar la realidad, tenemos un problema
- Si los muestreros que tenemos en los datos no representen bien la realidad, tenemos un problema

## Sampling II



**Figure 11:** Mal muestrear los datos puede dar lugar a ruido o sesgos

## Sampling III: Stratified



**Figure 12:** Muestreo estratificado

## Sampling III: Stratified

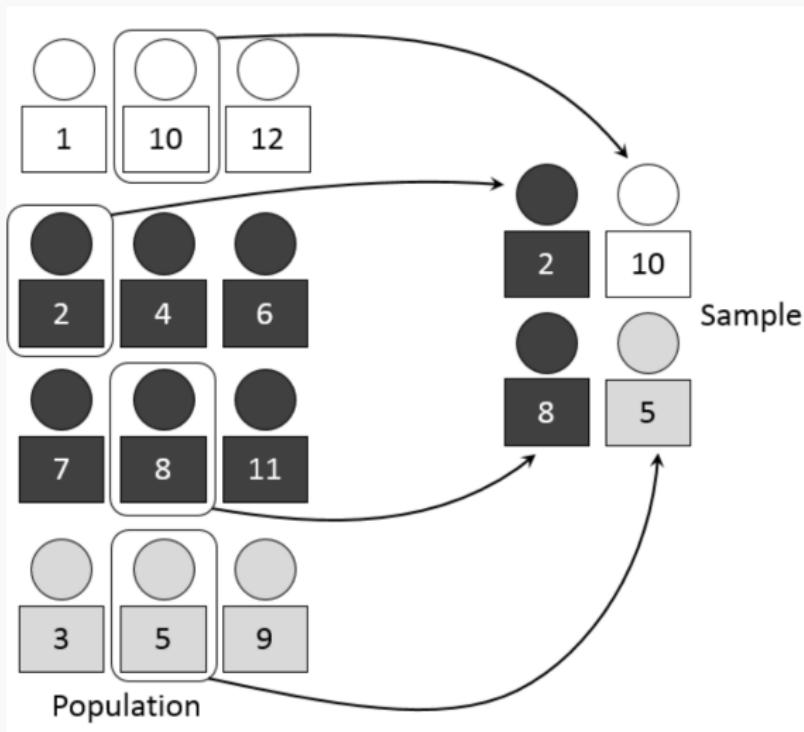
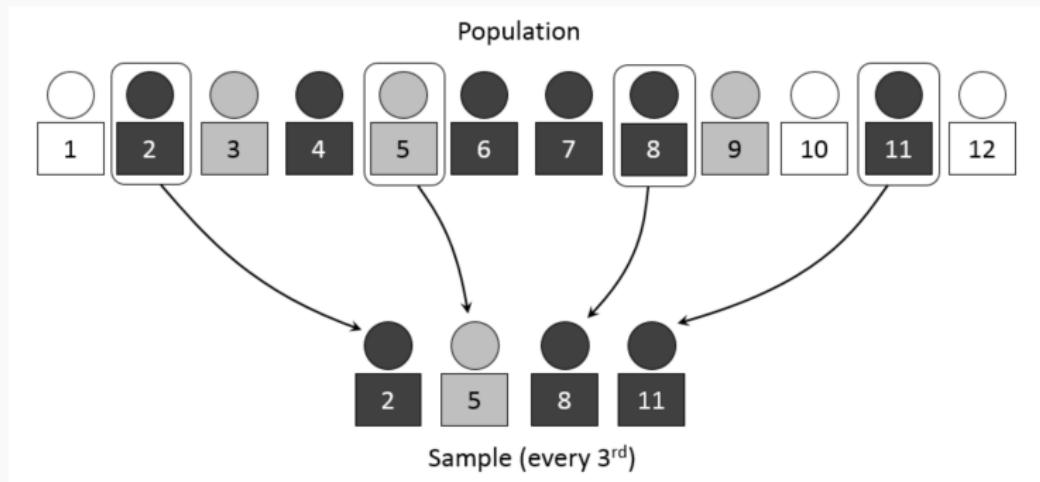


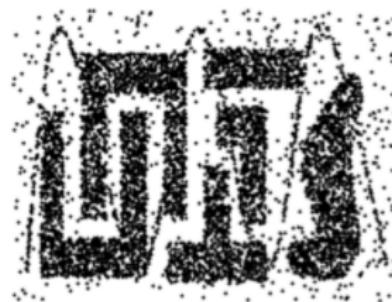
Figure 12: Muestreo estratificado: **Cuales son las ventajas y desventajas?**

## Sampling IV: Systematic Random Sampling



**Figure 13:** Selecciona elementos de una población a intervalos regulares desde un punto de partida aleatorio

## Sampling V



8000 points



2000 Points



500 Points

**Figure 14:** Que sampling se ve mejor para remover el ruido?

# Agregaciones

Combinar 2 o más atributos (o objetos) en un único atributo (o objeto):

- Reducción de datos: menos
- Cambio de escala: ventanas para alinear datos temporalmente
- Datos más estables: menos ruido

## Intuición

Reducción de la variabilidad

# Aggregaciones: Positive Emotion Analysis on Twitter

Average Happiness for Twitter

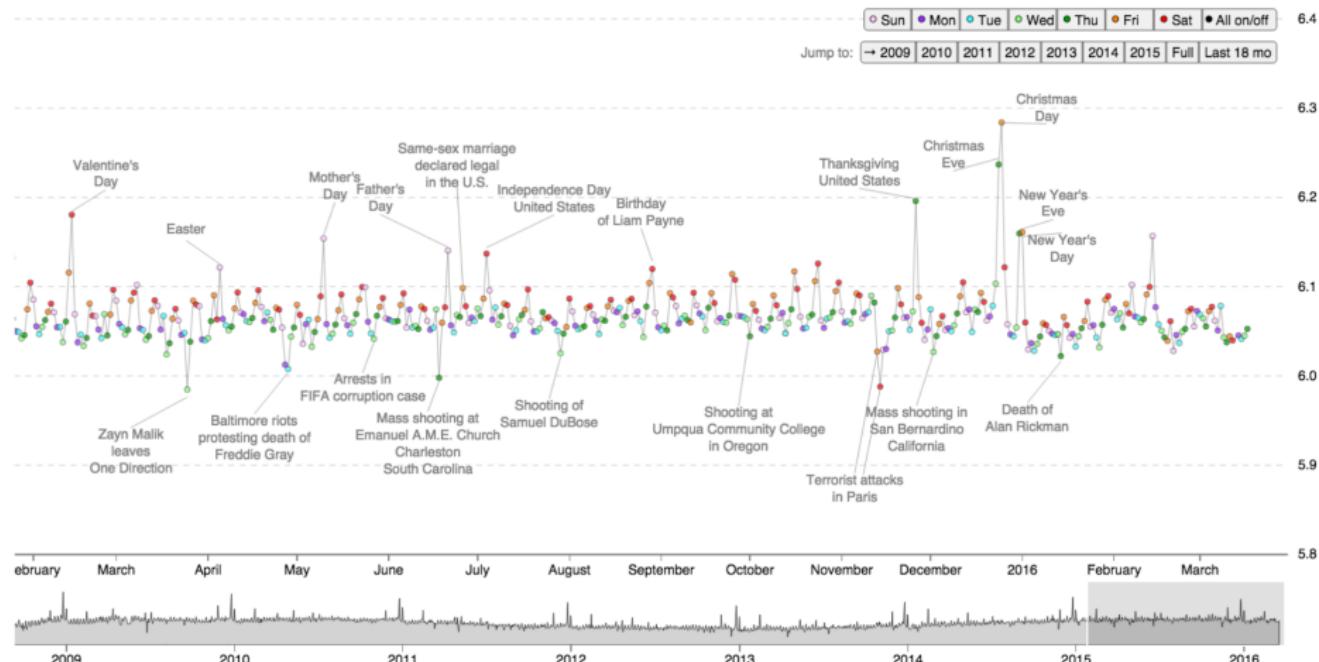


Figure 15: Agregación sobre un día para medir opinión general

# Aggregaciones: Positive Emotion Analysis on Twitter

Average Happiness for Twitter

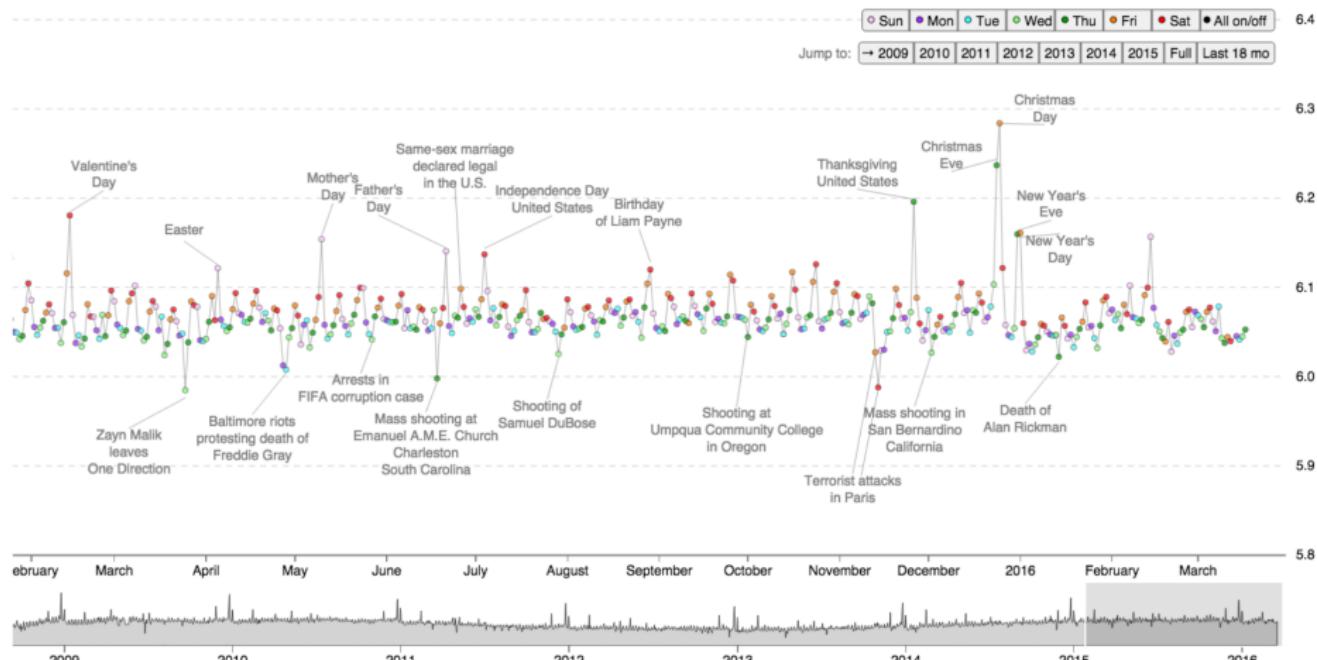
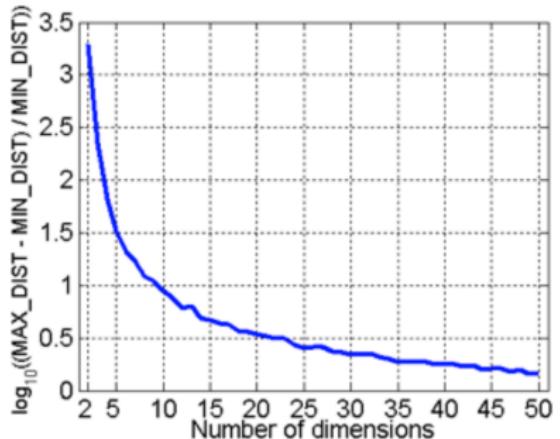


Figure 15: Agregación sobre un día para medir opinión general

Ventajas? Desventajas?

# Cursa de la dimensión

- Al aumentar dimension, los datos se vuelven más dispersos en el espacio
- Pierden significado las medidas, i.e. densidad y distancia entre puntos (clustering y detección de outliers)



## Intuición

Cuanto más aumente la dimensión, más probable será que los puntos se acerquen unos a otros. Mas dimensiones, mas ruido. Vamos a agregar mas cosas que no sirven para la tarea, y que pueden resultar en correlaciones espurias.

# Reducción de dimensión

Hay técnicas para reducir la dimensión de los datos:

- Selección de atributos: los mas "buenos", pertinentes, con menos correlaciones entre ellos
- Análisis en componentes principales: buscar nuevos atributos (abstractos) que capturan la máxima variación de los datos

## A que sirven?

- (i) Evitar curse of dimensionality, (ii) Reducir costos asociados a aplicar el algoritmos (tiempo, memoria), (iii) Mejor visualización de los datos, (iv) Ayuda a quitar atributos irrelevantes o ruidoso

**Lo vamos a ver mas adelante en el curso!**

Algoritmos con que se puede hacer la selección de atributos [en este lasso](#)

More about Unsupervised dimensionality reduction [here](#)

## Ejemplo de selección de atributos

Se puede utilizar la función `VarianceThreshold`, que es un selector de atributos que elimina todas los atributos de baja varianza.

```
>>> from sklearn.feature_selection import VarianceThreshold  
>>> X = [[0, 2, 0, 3], [0, 1, 4, 3], [0, 1, 1, 3]]  
>>> selector = VarianceThreshold()  
>>> selector.fit_transform(X)  
array([[2, 0],  
       [1, 4],  
       [1, 1]])
```

**Questions?**