DEPARTAMENTO
DE COMPUTACION

Facultad de Ciencias Exactas y Naturales - UBA

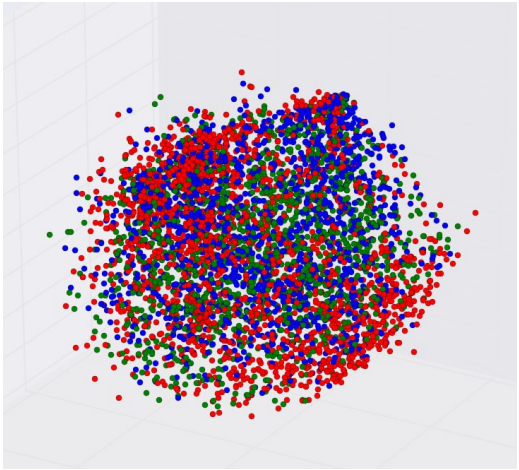
Aprendizaje Automático

Clase 10:

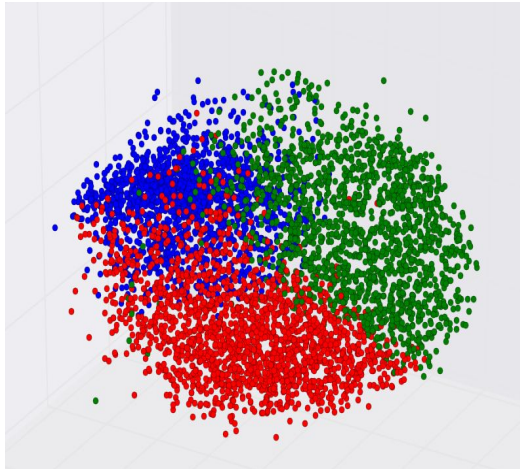
Ingeniería de Atributos

Garbage in... garbage out

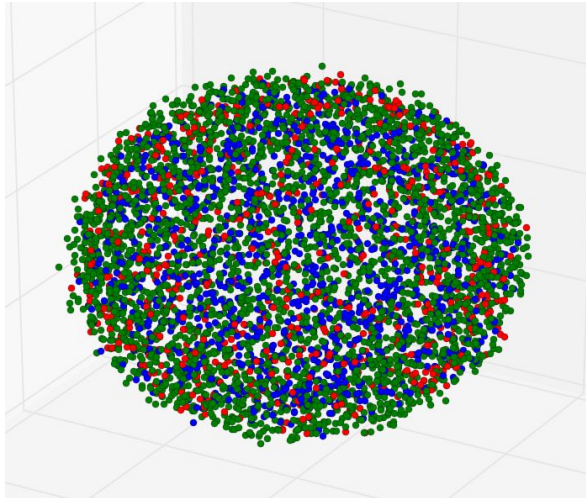
(Peso, Color, #Ruedas)



(Peso, #Patas, Material)



(Material, #Ojos, #Ruedas)



¿Por qué trabajar sobre los atributos?

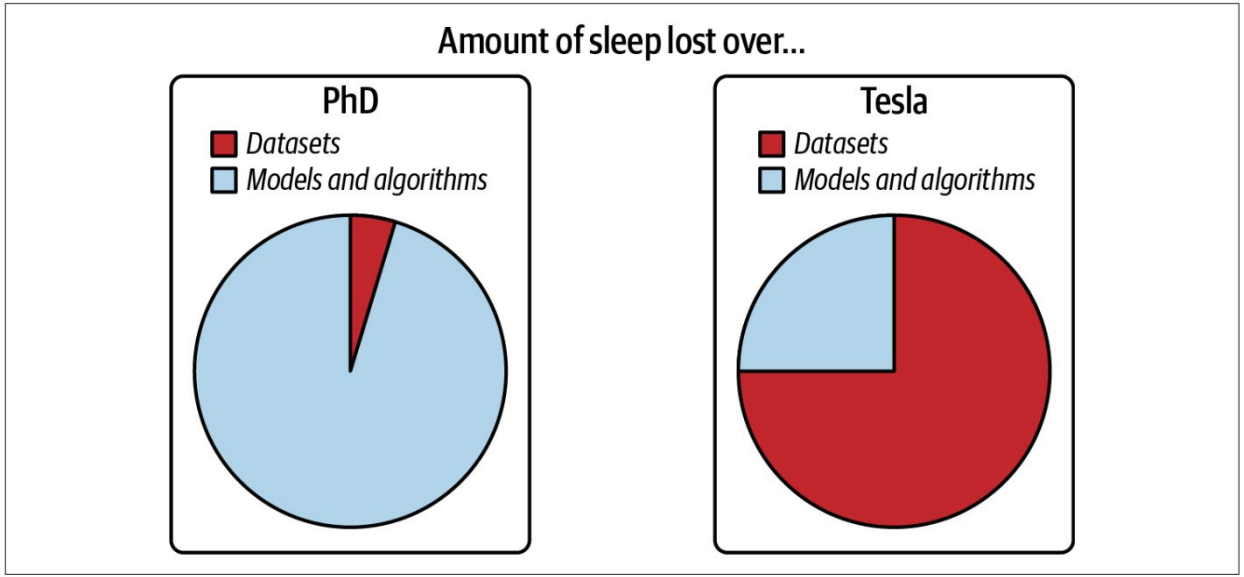


Figure 1-5. Data in research versus data in production. Source: Adapted from an image by Andrej Karpathy[24]

[Designing Machine Learning Systems: An Iterative Process for Production-Ready Applications" (Chip Huyen).]

¿Por qué trabajar sobre los atributos?

Ingeniería de Atributos: Tareas relacionadas a diseñar un conjunto de features. Por ejemplo:

- Tratamiento de **valores faltantes**.
- **Conversión** de atributos.
- **Normalización.**
- **Codificación**
- etc

"Buenos atributos permiten que un modelo simple supere a un modelo complejo" (Peter Norvig)

"Es incalculable el tiempo que se pierde por subestimar este tema" (Pablo Brusco :P)

ID	Age	Gender	Annual income	Marital status	Number of children	Job	Buy?
1		A	150,000		1	Engineer	No
2	27	B	50,000			Teacher	No
3		A	100,000	Married	2		Yes
4	40	B			2	Engineer	Yes

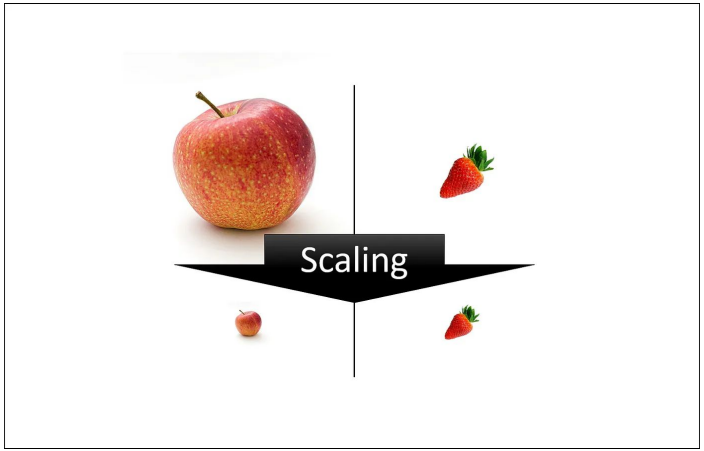
→

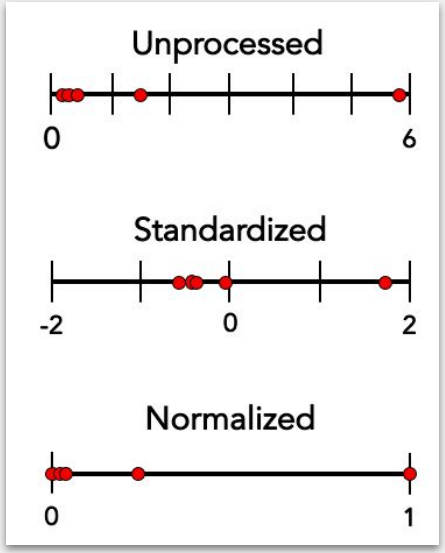
$$\begin{bmatrix} 1.23 \\ 2.34 \\ 3.45 \\ 4.56 \\ 5.67 \end{bmatrix} \quad \begin{bmatrix} 2.13 \\ 3.24 \\ 4.35 \\ 5.46 \\ 6.57 \end{bmatrix} \quad \begin{bmatrix} 3.12 \\ 4.23 \\ 5.34 \\ 6.45 \\ 7.56 \end{bmatrix} \quad \begin{bmatrix} 4.21 \\ 5.32 \\ 6.43 \\ 7.54 \\ 8.65 \end{bmatrix}$$

[Huyen, C. (2022). *Designing machine learning systems*. " O'Reilly Media, Inc.".]

Operaciones comunes de ingeniería de atributos:

Escalado de atributos

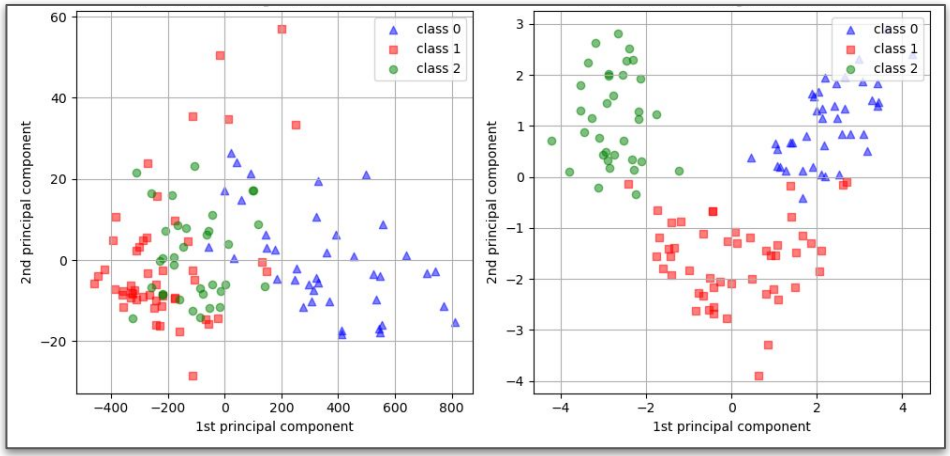




Escalado de atributos

Sin escalar

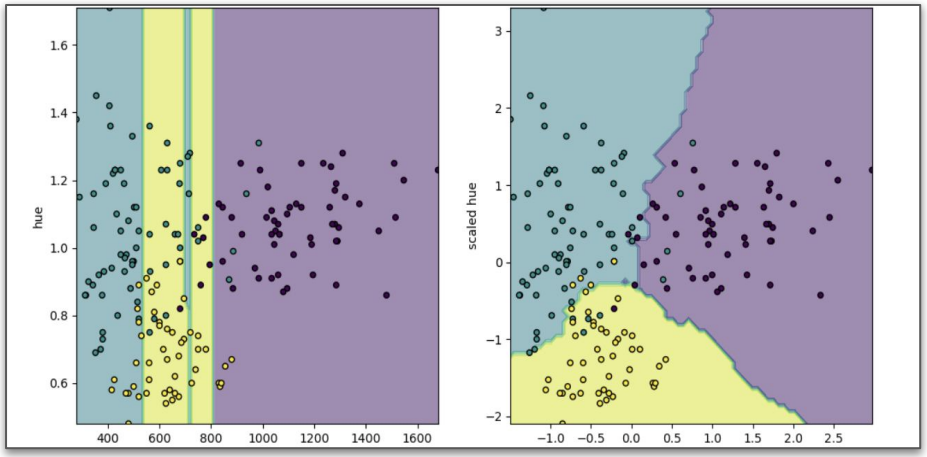
Escalando



PCA

Sin escalar

Escalando



KNN

- Contribución equitativa de atributos (¿deseado?).
- Velocidad de convergencia (SGD)
- Algoritmos basados en distancia (KNN, SVM, K-Means, etc),
- Adaptabilidad a nuevas distribuciones (reutilización de modelos existentes en datos nuevos)
- etc.

https://scikit-learn.org/stable/auto_examples/preprocessing/plot_scaling_importance.html

Escalado de atributos

Normalización (Min-Max Scaling):
Se reescalan los atributos a un rango fijo, generalmente entre $a=0$ y $b=1$.

Estandarización ("standardization", z-scores):
Se centra la distribución en media 0 y con una desvío estándar 1. Los $z\text{-scores}$ se pueden interpretar como cuántas desviaciones estándar está un punto de datos por encima o por debajo de la media del conjunto.

Escalado robusto (Robust Scaling): Utiliza la mediana y el rango intercuartil en lugar de la media y la desviación estándar, lo que lo hace más robusto frente a valores atípicos.

```
from sklearn.preprocessing import MinMaxScaler
```

$$X'_j = a + \frac{(X_j - X_j^{\min})(b-a)}{X_j^{\max} - X_j^{\min}}$$

```
from sklearn.preprocessing import StandardScaler
```

$$X'_j = \frac{X_j - \mu_j}{\sigma_j}$$

```
from sklearn.preprocessing import RobustScaler
```

$$X'_j = \frac{X_j - \text{mediana}_j}{\text{IQR}_j}$$

IQR = rango entre el primer cuartil y el tercer cuartil (percentil 25 - percentil 75).

Escalado de atributos

Normalización (Min-Max Scaling):

- ✗ Sensible a Outliers
- ✗ Datos nuevos pueden caer fuera de rango

```
from sklearn.preprocessing import MinMaxScaler
```

$$X'_j = a + \frac{(X_j - X_j^{\min})(b-a)}{X_j^{\max} - X_j^{\min}}$$

Estandarización ("standardization", z-scores):

- ✗ Supone distribución Normal.
- ✗ Sensible a Outliers (no tanto como min-max)

```
from sklearn.preprocessing import StandardScaler
```

$$X'_j = \frac{X_j - \mu_j}{\sigma_j}$$

Escalado robusto (Robust Scaling):

- ✗ Ignora extremos (cuyo valor puede afectar negativamente a los algoritmos)

```
from sklearn.preprocessing import RobustScaler
```

$$X'_j = \frac{X_j - \text{mediana}_j}{\text{IQR}_j}$$

IQR = rango entre el primer cuartil y el tercer cuartil (percentil 25 - percentil 75).

Escalado de atributos

El escalado de atributos es una fuente común de data leakage.

¿**Sobre qué datos computar** min, max, mu, sigma, IQR, etc?

```
from sklearn.preprocessing import
StandardScaler

scaler = StandardScaler()

scaler.fit(X_train)
X_train_scaled = scaler.transform(X_train)
X_test_scaled = scaler.transform(X_test)
```

```
from sklearn.preprocessing import MinMaxScaler
```

$$X'_j = a + \frac{(X_j - X_j^{\min})(b-a)}{X_j^{\max} - X_j^{\min}}$$

```
from sklearn.preprocessing import StandardScaler
```

$$X'_j = \frac{X_j - \mu_j}{\sigma_j}$$

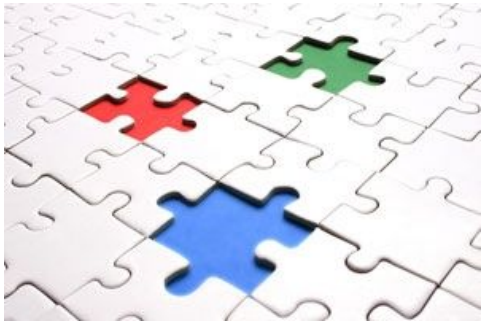
```
from sklearn.preprocessing import RobustScaler
```

$$X'_j = \frac{X_j - \text{mediana}_j}{\text{IQR}_j}$$

IQR = rango entre el primer cuartil y el tercer cuartil (percentil 25 - percentil 75).

Operaciones comunes de ingeniería de atributos:

Tratar Valores Faltantes (missing values)



Nombre	Edad	Profesión
Emanuel Ginobili	45.0	Basquetbolista
Ada Lovelace	NaN	Programadora
Juan Pablo Galeotti	NaN	Director del DC
Chuck Norris	83.0	Todas
Mirta Legrand	NaN	Conductora

"Obviously, the best way to treat missing data is not to have them".

Manejo de valores faltantes

Posibles soluciones

- **Aceptarla y utilizar algoritmos** diseñados para datos faltantes
- **Eliminar** datos con problemas
  - ¿Eliminar **filas** o **columnas**? ¿Y cuando llega un dato nuevo?
- Convertir missing en una **categoría** más (para categóricas)
- **Rellenar** (Imputers)
  - Generales:
    - Media / Mediana / Moda / Constante
    - Random Forest Imputer, KNN imputer, MICE, etc
  - Para series temporales:
    - Last Observation Carried Forward (LOCF)
    - Next Observation Carried Backward (NOCB)
    - Interpolación (lineal, pesada, splines, etc)
- **Imputación + columnas indicadoras** (dummy variables)

En **scikit-learn:** (ojo porque son especialmente malos con este tema) https://scikit-learn.org/stable/modules/tree.html#missing-values-support

Nombre	Edad	Profesión
Emanuel Ginobili	45.0	Basquetbolista
Ada Lovelace	NaN	Programadora
Juan Pablo Galeotti	NaN	Director del DC
Chuck Norris	83.0	Todas
Mirta Legrand	NaN	Conductora

99% accuracy

Manejo de valores faltantes

Posibles soluciones

- **Aceptarla y utilizar algoritmos** diseñados para datos faltantes
- **Eliminar** datos con problemas
  - ¿Eliminar **filas** o **columnas**? ¿Y cuando llega un dato nuevo?
- Convertir missing en una **categoría** más (para categóricas)
- **Rellenar** (Imputers)
  - Generales:
    - Media / Mediana / Moda / Constante
    - Random Forest Imputer, KNN imputer, MICE, etc
  - Para series temporales:
    - Last Observation Carried Forward (LOCF)
    - Next Observation Carried Backward (NOCB)
    - Interpolación (lineal, pesada, splines, etc)
- **Imputación + columnas indicadoras** (dummy variables)

Nombre	Edad	Profesión
Emanuel Ginobili	45.0	Basquetbolista
Ada Lovelace	NaN	Programadora
Juan Pablo Galeotti	NaN	Director del DC
Chuck Norris	83.0	Todas
Mirta Legrand	NaN	Conductora

Manejo de valores faltantes

Posibles soluciones

- **Aceptarla y utilizar algoritmos** diseñados para datos faltantes
- **Eliminar** datos con problemas
  - ¿Eliminar **filas** o **columnas**? ¿Y cuando llega un dato nuevo?
- Convertir missing en una **categoría** más (para categóricas)
- **Rellenar** (Imputers)
  - Generales:
    - Media / Mediana / Moda / Constante
    - Random Forest Imputer, KNN imputer, MICE, etc
  - Para series temporales:
    - Last Observation Carried Forward (LOCF)
    - Next Observation Carried Backward (NOCB)
    - Interpolación (lineal, pesada, splines, etc)
- **Imputación + columnas indicadoras** (dummy variables)

Nombre	Edad	Profesión
Emanuel Ginobili	45.0	Basquetbolista
Ada Lovelace	NaN	Programadora
Juan Pablo Galeotti	NaN	Director del DC
Chuck Norris	83.0	Todas
Mirta Legrand	NaN	Conductora

Manejo de valores faltantes

Posibles soluciones

- **Aceptarla y utilizar algoritmos** diseñados para datos faltantes
- **Eliminar** datos con problemas
  - ¿Eliminar **filas** o **columnas**? ¿Y cuando llega un dato nuevo?
- Convertir missing en una **categoría** más (para categóricas)
- **Rellenar** (Imputers)
  - Generales:
    - Media / Mediana / Moda / Constante
    - Random Forest Imputer, KNN imputer, MICE, etc
  - Para series temporales:
    - Last Observation Carried Forward (LOCF)
    - Next Observation Carried Backward (NOCB)
    - Interpolación (lineal, pesada, splines, etc)
- **Imputación + columnas indicadoras** (dummy variables)

Manejo de valores faltantes

Pausa: ¿pero por qué no están?

Nombre	Edad	Profesión	¿Vive?
Emanuel Ginobili	45.0	Basquetbolista	True
Ada Lovelace	NaN	Programadora	False
Juan Pablo Galeotti	NaN	Director del DC	True
Chuck Norris	83.0	Todas	True
Mirta Legrand	NaN	Conductora	True

¿Por qué puede faltar un valor?

Entender por qué faltan ayuda a pensar cuál es la manera **correcta** de trabajar con ellos.

Manejo de valores faltantes

Pausa: ¿pero por qué no están?

Nombre	Edad	Profesión	¿Vive?
Emanuel Ginobili	45.0	Basquetbolista	True
Ada Lovelace	NaN	Programadora	False
Juan Pablo Galeotti	NaN	Director del DC	True
Chuck Norris	83.0	Todas	True
Mirta Legrand	NaN	Conductora	True

1- Faltante al azar (MCAR - Missing Completely at Random)
Los datos faltantes no están relacionados con ningún dato observado o no observado.
Ej: no se entiende la letra de la persona que hizo la encuesta.

2- Faltante con dependencias (MAR - Missing at Random)
Los datos faltantes están relacionados con los datos observados, pero no con los datos faltantes en sí.
Ej: Falta la nota del recu, pero la persona aprobó el parcial.

3- Faltante sospechoso (MNAR - Missing Not at Random):
Los datos faltantes están relacionados con el motivo por el cual faltan.
Ej: ¿Cuánto gana? (si es muy alto quizás no contestan)

[Rubin, Donald B. 1976]

Manejo de valores faltantes

Pausa: ¿pero por qué no están?

Nombre	Edad	Profesión	¿Vive?
Emanuel Ginobili	45.0	Basquetbolista	True
Ada Lovelace	NaN	Programadora	False
Juan Pablo Galeotti	NaN	Director del DC	True
Chuck Norris	83.0	Todas	True
Mirta Legrand	NaN	Conductora	True

1- Faltante al azar (MCAR - Missing Completely at Random)
Los datos faltantes no están relacionados con ningún dato observado o no observado.
Ej: no se entiende la letra de la persona que hizo la encuesta.

2- Faltante con dependencias (MAR - Missing at Random)
Los datos faltantes están relacionados con los datos observados, pero no con los datos faltantes en sí.
Ej: Falta la nota del recu, pero la persona aprobó el parcial.

3- Faltante sospechoso (MNAR - Missing Not at Random):
Los datos faltantes están relacionados con el motivo por el cual faltan.
Ej: ¿Cuánto gana? (si es muy alto quizás no contestan)

[Rubin, Donald B. 1976]

Manejo de valores faltantes

Pausa: ¿pero por qué no están?

Nombre	Edad	Profesión	¿Vive?
Emanuel Ginobili	45.0	Basquetbolista	True
Ada Lovelace	NaN	Programadora	False
Juan Pablo Galeotti	NaN	Director del DC	True
Chuck Norris	83.0	Todas	True
Mirta Legrand	NaN	Conductora	True

1- Faltante al azar (MCAR - Missing Completely at Random)
Los datos faltantes no están relacionados con ningún dato observado o no observado.
Ej: no se entiende la letra de la persona que hizo la encuesta.

2- Faltante con dependencias (MAR - Missing at Random)
Los datos faltantes están relacionados con los datos observados, pero no con los datos faltantes en sí.
Ej: Falta la nota del recu, pero la persona aprobó el parcial.

3- Faltante sospechoso (MNAR - Missing Not at Random):
Los datos faltantes están relacionados con el motivo por el cual faltan.
Ej: ¿Cuánto gana? (si es muy alto quizás no contestan)

[Rubin, Donald B. 1976]

Manejo de valores faltantes

Posibles soluciones

- **Aceptarla y utilizar algoritmos** diseñados para datos faltantes
- **Eliminar** datos con problemas
  - ¿Eliminar **filas** o **columnas**? ¿Y cuando llega un dato nuevo?
- Convertir missing en una **categoría** más (para categóricas)
- **Rellenar** (Imputers)
  - Generales:
    - Media / Mediana / Moda / Constante
    - Random Forest Imputer, KNN imputer, MICE, etc
  - Para series temporales:
    - Last Observation Carried Forward (LOCF)
    - Next Observation Carried Backward (NOCB)
    - Interpolación (lineal, pesada, splines, etc)
- **Imputación + columnas indicadoras** (dummy variables)

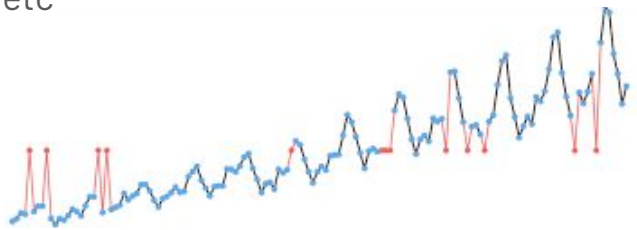
Nombre	Edad	Profesión
Emanuel Ginobili	45.0	Basquetbolista
Ada Lovelace	NaN	Programadora
Juan Pablo Galeotti	NaN	Director del DC
Chuck Norris	83.0	Todas
Mirta Legrand	NaN	Conductora

Nombre	Edad	Profesión
Emanuel Ginobili	45	Basquetbolista
Ada Lovelace	45	Programadora
Juan Pablo Galeotti	45	Director del DC
Chuck Norris	83	Todas
Mirta Legrand	83	Conductora

Manejo de valores faltantes

Posibles soluciones

- **Aceptarla y utilizar algoritmos** diseñados para datos faltantes
- **Eliminar** datos con problemas
  - ¿Eliminar **filas** o **columnas**? ¿Y cuando llega un dato nuevo?
- Convertir missing en una **categoría** más (para categóricas)
- **Rellenar** (Imputers)
  - Generales:
    - Media / Mediana / Moda / Constante
    - Random Forest Imputer, KNN imputer, MICE, etc
  - Para series temporales:
    - Last Observation Carried Forward (LOCF)
    - Next Observation Carried Backward (NOCB)
    - Interpolación (lineal, pesada, splines, etc)
- **Imputación + columnas indicadoras** (dummy variables)



Manejo de valores faltantes

Posibles soluciones

- **Aceptarla y utilizar algoritmos** diseñados para datos faltantes
- **Eliminar** datos con problemas
  - ¿Eliminar **filas** o **columnas**? ¿Y cuando llega un dato nuevo?
- Convertir missing en una **categoría** más (para categóricas)
- **Rellenar** (Imputers)
  - Generales:
    - Media / Mediana / Moda / Constante
    - Random Forest Imputer, KNN imputer, MICE, etc
  - Para series temporales:
    - Last Observation Carried Forward (LOCF)
    - Next Observation Carried Backward (NOCB)
    - Interpolación (lineal, pesada, splines, etc)
- **Imputación + columnas indicadoras** (dummy variables)

Edad	edad_was_none
45	False
45	True
45	True
83	False
83	True

Manejo de valores faltantes

KNN-imputter

Cada atributo faltante X_m se imputa utilizando los valores de los **n-vecinos** más cercanos que tienen un valor para X_m, promediando uniformemente o ponderando por distancia.

Por defecto, se utiliza una métrica de distancia euclidiana que soporta valores faltantes, **nan_euclidean_distances**, para encontrar los vecinos más cercanos.

Ej, la distancia entre [3, na, na, 6] y [1, na, 4, 5] es :

$$\sqrt{\frac{4}{2}((3-1)^2+(6-5)^2)}$$

En donde $4/2$ es la fracción: #total / #definidos.

```python
from sklearn.impute import KNNImputer

X = [[1, 2, np.nan],
     [3, 4, 3],
     [np.nan, 6, 5],
     [8, 8, 7]]

imputer = KNNImputer(n_neighbors=2)
imputer.fit_transform(X)

# devuelve:
          [[1. , 2. , 4. ],
           [3. , 4. , 3. ],
           [5.5, 6. , 5. ],
           [8. , 8. , 7. ]]
```

Manejo de valores faltantes

Imputación por iteraciones

Repetir #iteraciones (o hasta cumplir criterio de convergencia):

1. **Fijar valores iniciales** para los valores faltantes con cualquier otro método (promedios, knn, etc)
2. **Seleccionar una variable con valores faltantes**.
   a. Crear un modelo que prediga valores a partir del valor del resto de los atributos.
      El modelo puede ser regresión lineal, de regresión logística, árboles de decisión, etc, dependiendo de la naturaleza de los datos.
   b. Utilizar el modelo para predecir los faltantes.
3. Continuar el proceso el resto de las variables, utilizando en cada paso la versión imputada de las otras variables.

```
from sklearn.impute import IterativeImputer

X = [[1, 2, np.nan],
     [3, 4, 3],
     [np.nan, 6, 5],
     [8, 8, 7]]

imputer = IterativeImputer(
     random_state=0,
     max_iter=10
     initial_strategy='mean',
     add_indicator=True
)

imputer.fit_transform(X)

# devuelve:
          [[1. , 2. , 1.2, 0. , 1. ],
           [3. , 4. , 3. , 0. , 0. ],
           [5.5, 6. , 5. , 1. , 0. ],
           [8. , 8. , 7. , 0. , 0. ]]
```

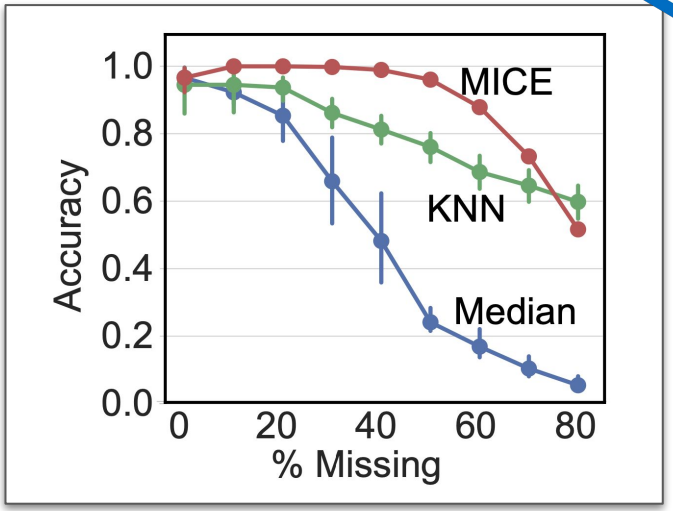
Manejo de valores faltantes

Imputación Múltiple - midiendo incertidumbre

En la comunidad estadística, es común realizar "imputaciones múltiples" para poder medir incertidumbre:

MICE: Multivariate Imputation by Chained Equations

1. Generar **m** versiones del dataset imputado (utilizando subsets de la data para imputar cada vez, a lo Bagging).
2. Calcular **m** resultados finales del análisis que se esté realizando (o de modelos finales)
3. Obtener, además de los **resultados agregados** a partir de los m sub-resultados (ej, la media) alguna medida de varianza. Permite responder:
   "¿cómo difieren los resultados debido a la **incertidumbre** inherente causada por los valores faltantes?"



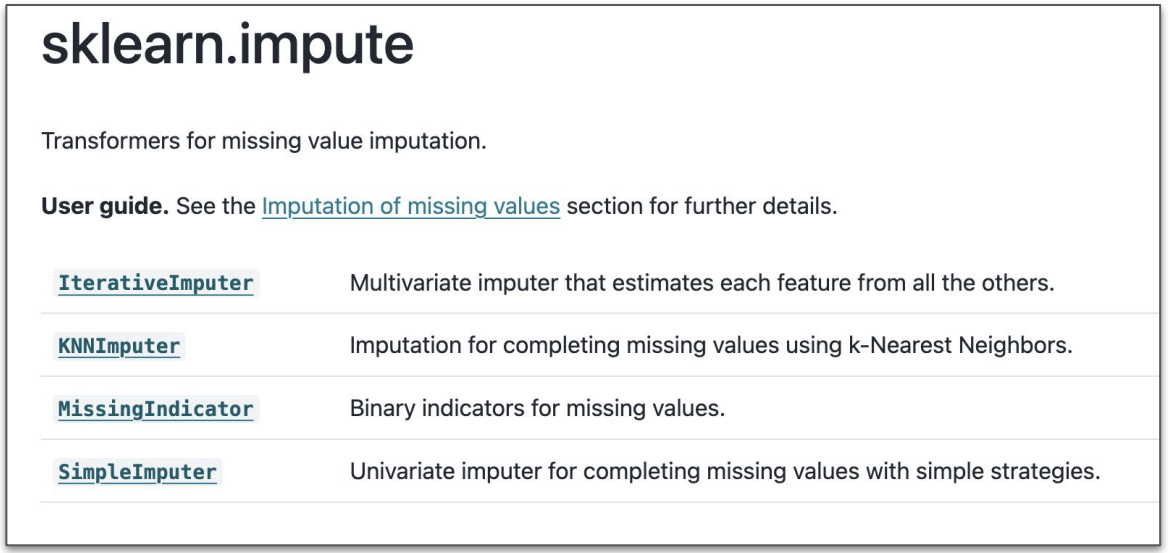
[Samad, M. D., Abrar, S., & Diawara, N. (2022). **Missing value estimation using clustering and deep learning within multiple imputation framework**. *Knowledge-based systems*, *249*, 108968.]

En el paper hay análisis cuando las variables son MCAR, MAR y MNR)

Ver: https://www.youtube.com/watch?v=LMsULWGtP2c

Técnicas de imputación

Warning: Puede crear combinaciones inexistentes (ej., niño de 1 año, altura 1.80cm)
Warning: Destruir relaciones determinísticas (ej., suma de notas de examen y las notas individuales)
Warning: Puede crear datos sin sentido (ej., lugar de trabajo para un desocupado).
Warning: La imputación de valores faltantes es una de las maneras más comunes de **filtrar información**



sklearn.impute

Transformers for missing value imputation.

User guide. See the Imputation of missing values section for further details.

`IterativeImputer`	Multivariate imputer that estimates each feature from all the others.
`KNNImputer`	Imputation for completing missing values using k-Nearest Neighbors.
`MissingIndicator`	Binary indicators for missing values.
`SimpleImputer`	Univariate imputer for completing missing values with simple strategies.

fit / transform / fit_transform

```
from sklearn.experimental import enable_iterative_imputer
from sklearn.impute import IterativeImputer
import numpy as np

data = np.array([[1, 2, np.nan],
                 [3, np.nan, 5],
                 [np.nan, 6, 9]])

imputer = IterativeImputer(max_iter=10, random_state=0)

imputed_data = imputer.fit_transform(data)
print(imputed_data)
```

```
[[ 1.          2.          6.24850451]
 [ 3.          0.18502855  5.        ]
 [-3.40757173  6.          9.        ]]
```

```
data2 = np.array([[np.nan, 0, np.nan],
                  [0, np.nan, 1],
                  [np.nan, 0, 0]])
imputer.transform(data2)
```

```
[[3.2035835  0.         4.87282628]
 [0.         0.51338127 1.        ]
 [5.71040122 0.         0.        ]]
```

fit / transform / fit_transform

fit

Calcula los parámetros necesarios para la transformación a partir de los datos de entrenamiento.

fit_transform

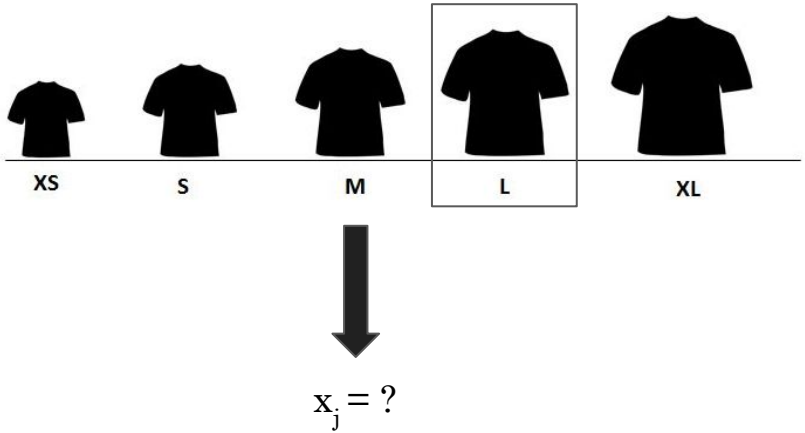
Ajusta y transforma los datos de entrenamiento en una sola llamada.

transform

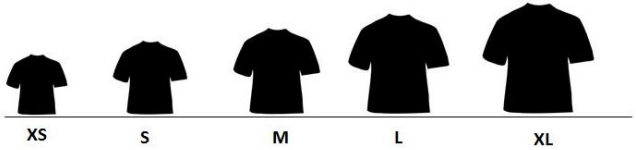
Aplica la transformación a datos nuevos utilizando los parámetros calculados por **fit**.

Operaciones comunes de ingeniería de atributos:

Conversión de Variables



Conversión de variables



Los modelos sólo entienden vectores de números. ¿Si las features no son números? Muchas veces estos atributos contienen mucha información comprimida.

Tipos de variables a tratar

- Numéricas: *edad.*
- Nominales: *nacionalidad.*
- Ordinales: *tamaño remera.*
- Fechas / Hora del día
- Ubicaciones (lat/long)

<3.2, 1, 0, 2.4, 6.6, 4.6, 5.3, 14.3>

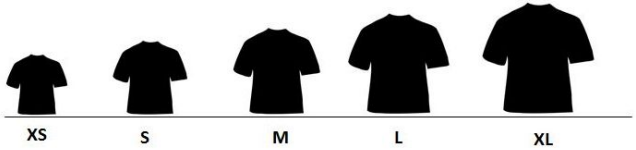
¿Cómo convertir Categóricas? (Parte I)

- **Label encoding:** Asignar números 1,2,3,4, etc.
  Para ordinales puede funcionar.
  nominales: en general, mala idea (¿rojo > verde?)

- **One-Hot encoding:** Por cada valor posible, crear un atributo binarios. Ej: valor "remera_roja" = <0, 0, 0, 1, 0, 0>

  Variante 1: *Dummy Variables*, 1 o 0 por todas las categorías menos una. La ausencia de todas significa presencia de la categoría no encodeada.
  Variante 2: *Effect Coding Scheme* (-1 en vez de todo 0) para ayudar a los modelos.
  Cardinalidad: ¿Cuántas dimensiones se agregan?

- **Target encoding, count encoding** (googlearlas)
- **Categorical Embeddings** (próxima slide)



Conversión de variables

Los modelos sólo entienden vectores de números. ¿Si las features no son números? Muchas veces estos atributos contienen mucha información comprimida.

Tipos de variables a tratar

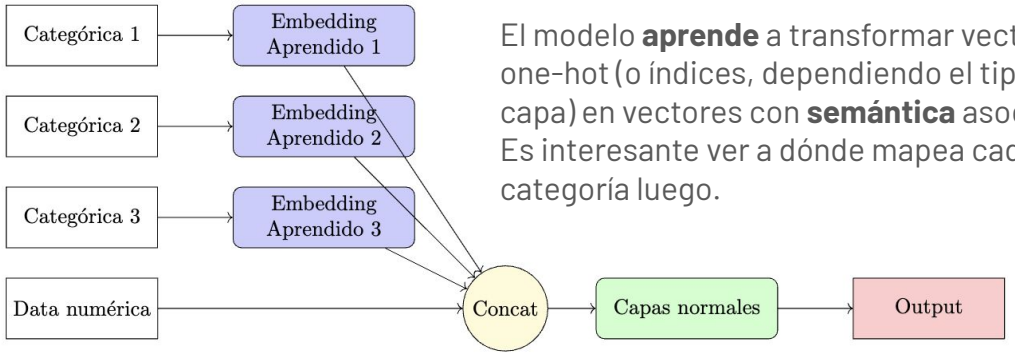
- Numéricas: *edad.*
- Nominales: *nacionalidad.*
- Ordinales: *tamaño remera.*
- Fechas / Hora del día
- Ubicaciones (lat/long)

<3.2, 1, 0, 2.4, 6.6, 4.6, 5.3, 14.3>

Categorical Embeddings

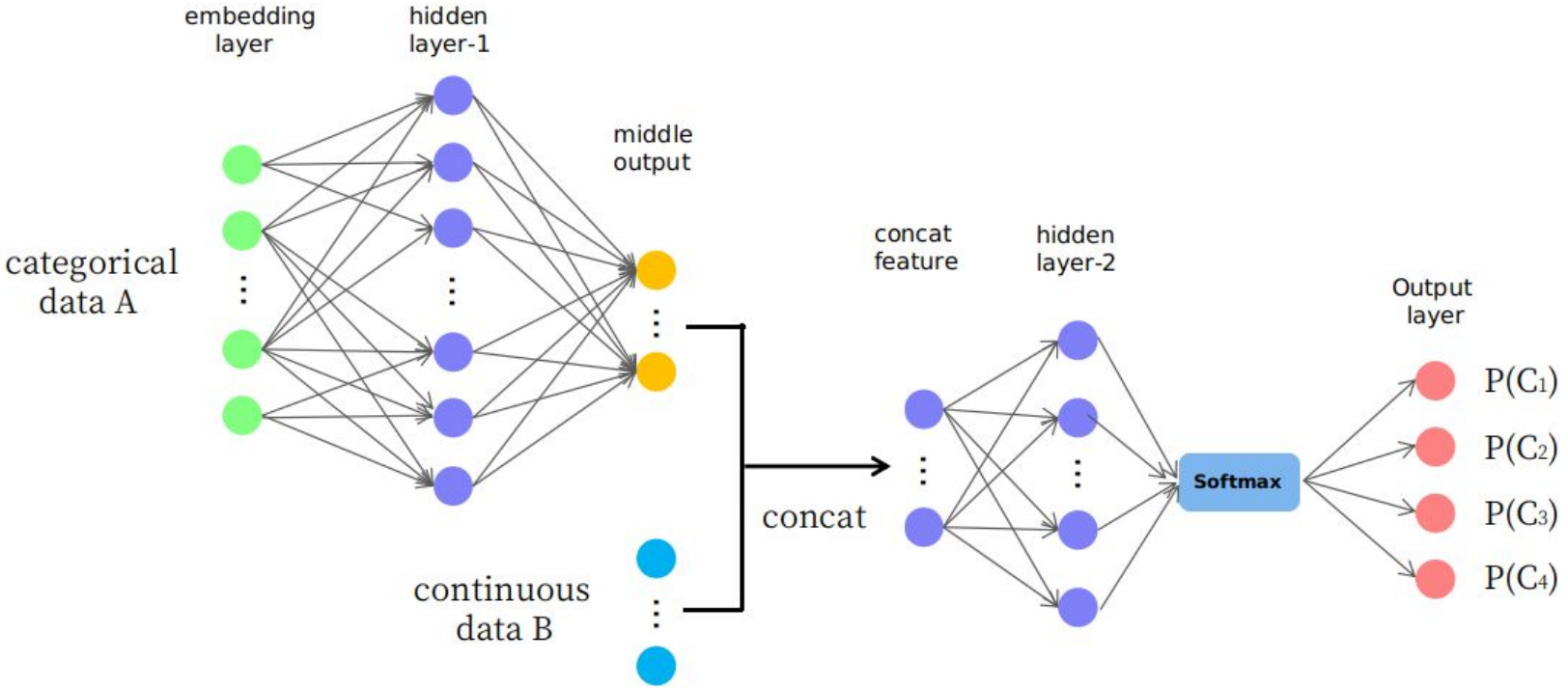
Convertir cada categoría en un **vector**. En donde dos categorías cercanas estarán cerca en el espacio de embeddings.

- **Opción 1**: Si los nombres de las categorías son palabras conocidas. Usar un embedder pre-entrenado (word2vec).
- **Opción 2**: Entrenar el modelo por separado (no supervisado).
- **Opción 3**: Entrenarla como parte de una red más grande)



El modelo **aprende** a transformar vectores one-hot (o índices, dependiendo el tipo de capa) en vectores con **semántica** asociada. Es interesante ver a dónde mapea cada categoría luego.

Categorical Embeddings: Esquema más realista



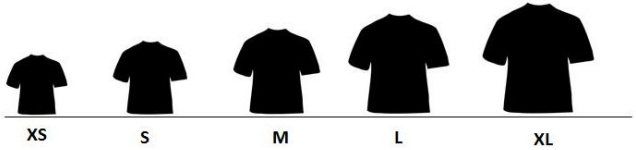
Fuente:

Categorical Embeddings: Ejemplo pytorch

```
class ModeloEjemplo(nn.Module):  # para 2 variables categóricas
    def __init__(self, num_categories1, num_categories2, num_numerical_features, output_dim):
        # en Pytorch “Embedding” espera indices. “Linear” espera vectores (one-hot)
         embedding_dim1 = embedding_dim2 = 3  # Cada attr será representado con 3 números.

        self.embedding1 = nn.Embedding(num_categories1, embedding_dim1)
        self.embedding2 = nn.Embedding(num_categories2, embedding_dim2)

        total_embedding_dim = embedding_dim1 + embedding_dim2
        total_input_dim = total_embedding_dim + num_numerical_features

        # capas Fully Connected:
        self.fc1 = nn.Linear(total_input_dim, 64)
        self.fc2 = nn.Linear(64, output_dim)

    def forward(self, categorical_data, numerical_data):
        cat1_embed = self.embedding1(categorical_data[:, 0])
        cat2_embed = self.embedding2(categorical_data[:, 1])

        # Concatenar embeddings y features numéricos
        x = torch.cat([cat1_embed, cat2_embed, numerical_data], dim=1)

        # Pasada foward por el resto de la red.
        x = relu(self.fc1(x))
        x = self.fc2(x)
        return x
```

Conversión de variables

Datos con estructura oculta



Los modelos sólo entienden vectores de números. ¿Si las features no son números? Muchas veces estos atributos contienen mucha información comprimida.

Tipos de variables a tratar

- Numéricas: *edad.*
- Nominales: *nacionalidad.*
- Ordinales: *tamaño remera.*
- Fechas / Hora del día
- Ubicaciones (lat/long)

`<3.2, 1, 0, 2.4, 6.6, 4.6, 5.3, 14.3>`

¿Cómo usar fechas / horas?

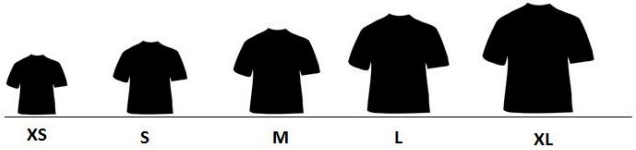
- Timestamp (segs desde 1/1/1970)
- Día de semana vs fin de semana
- ¿Horario laboral? ¿Principio de mes?
- Estación del año / Número de semana / Mes / Año
- Tiempo desde algún evento de interés
- Cantidad de días para que sea navidad
- etc.

¿Cómo usar geolocations?

- Convertir a zonas (knn si no se conocen de antemano)
- Distancia geodésica a algún punto de interés
- País / Región / Provincia / Ciudad / Barrio.
- etc

Conversión de variables

Extracción a partir de texto



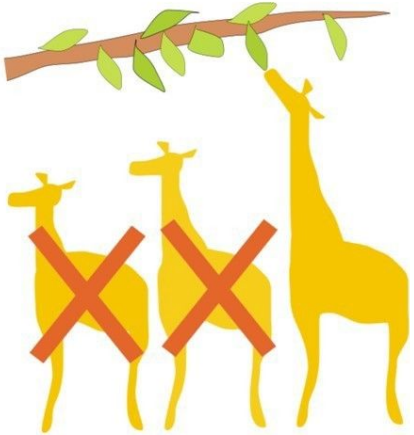

```python
import re

titulos = X['Name'].str.extract(r',([A-Za-z ]{1,20})\.', expand=False)
titulos = titulos.apply(lambda x: x.strip())

# Contar la frecuencia de títulos
contador_titulos = titulos.value_counts()
print(contador_titulos)
```

```
Mr              457
Miss            167
Mrs             114
Master           36
Dr                7
Rev               6
Major             2
Mlle              2
Col               2
Capt              1
Mme               1
Lady              1
the Countess      1
Ms                1
Jonkheer          1
Don               1
Sir               1
Name: Name, dtype: int64
```

Basado en: http://rstudio-pubs-static.s3.amazonaws.com/24969_894d890964fd4308ab537bfde1f784d2.html

Operaciones comunes de ingeniería de atributos:

Selección de variables



- Reduce la dimensión
- Favorece a la generalización
- Acelera la velocidad
- Mejora la interpretabilidad

Selección de Features

Idea: Reducir la dimensión mediante eliminación de variables poco útiles

Métodos univariados
Ranking generado a través de algún método estadístico.

Métodos iterativos
Considera el problema de selección de features como un problema de búsqueda.

Basados en importancia de atributos
Rankear variables según métodos internos de cada algoritmo

Métodos univariados
Objetivo: **¿Qué variable afecta más valor a predecir?**

Test **univariados** (suponen independencia condicional) para ver la relación entre los atributos y el target:

- T-test / Anova (para datos continuos).
- Chi-cuadrado, Information Gain (para datos categóricos).
- Pearson's correlation con Y.
- Gini Index.
- Correlación entre pares de variables.
- Un modelo por variable
- etc

Selección de Features

Idea: Reducir la dimensión mediante eliminación de variables poco útiles

Métodos univariados
Ranking generado a través de algún método estadístico.

Métodos iterativos
Considera el problema de selección de features como un problema de búsqueda.

Basados en importancia de atributos
Rankear variables según métodos internos de cada algoritmo

Métodos iterativos
Objetivo: **¿Cuál es la mejor combinación de variables?**

Se preparan **combinaciones**, se evalúan y se comparan a través de entrenar un modelo por combinación y luego, medir su performance.

- Heurísticas Greedy:
  - **Forward** stepwise selection (próxima slide)
  - **Backward** stepwise selection (próxima slide)
  - Random feature selection
- Best first search
- Random Climbing

Selección de Features

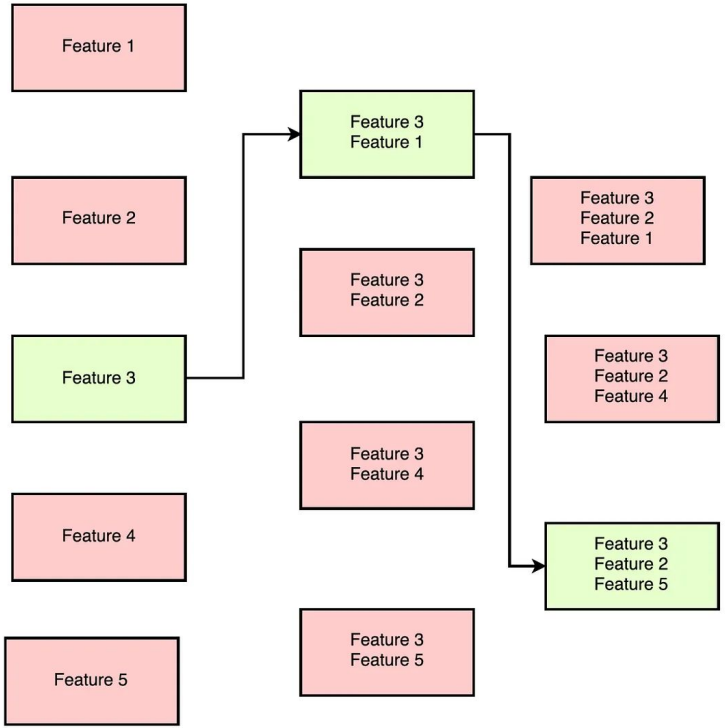
Idea: Reducir la dimensión mediante eliminación de variables poco útiles

Métodos univariados
Ranking generado a través de algún método estadístico.

Métodos iterativos
Considera el problema de selección de features como un problema de búsqueda.

Basados en importancia de atributos
Rankear variables según métodos internos de cada algoritmo

Forward stepwise selection



https://medium.com/@sagar.rawale3/feature-selection-methods-in-machine-learning-eaeef12019cc

Selección de Features

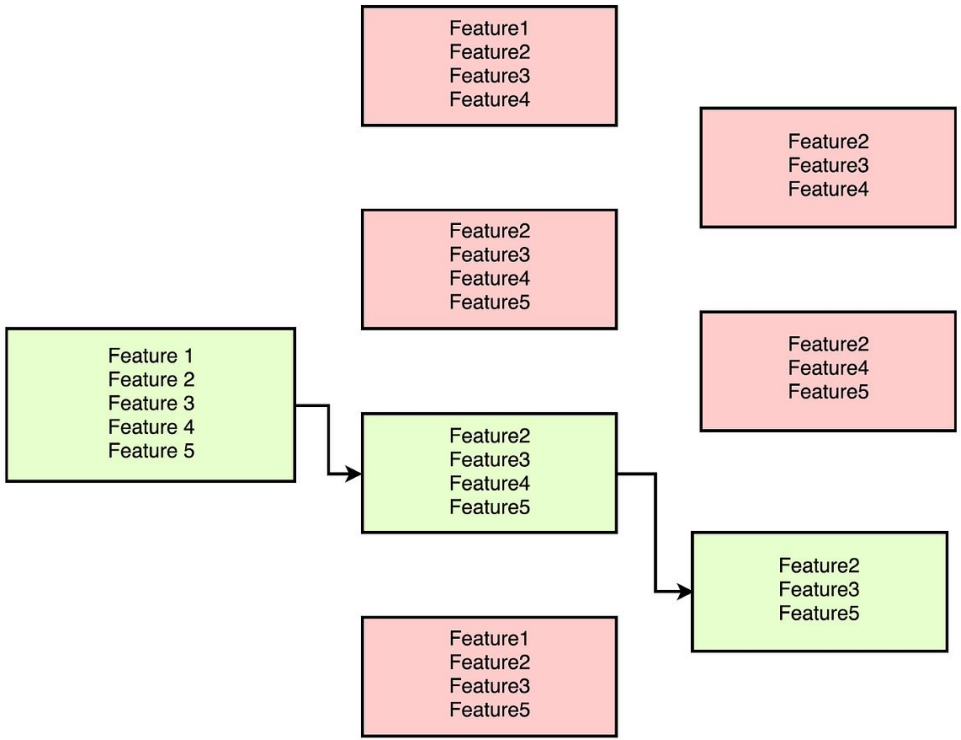
Idea: Reducir la dimensión mediante eliminación de variables poco útiles

Métodos univariados
Ranking generado a través de algún método estadístico.

Métodos iterativos
Considera el problema de selección de features como un problema de búsqueda.

Basados en importancia de atributos
Rankear variables según métodos internos de cada algoritmo

Backward stepwise selection



https://medium.com/@sagar.rawale3/feature-selection-methods-in-machine-learning-eaeef12019cc

Selección de Features

Idea: Reducir la dimensión mediante eliminación de variables poco útiles

Métodos univariados
Ranking generado a través de algún método estadístico.

Métodos iterativos
Considera el problema de selección de features como un problema de búsqueda.

Basados en importancia de atributos
Rankear variables según métodos internos de cada algoritmo

Basados en importancia de atributos

Rankear según **la importancia** que un modelo atribuye al atributo

- En árboles: para un árbol, calcular importancia de permutación o importancia Gini.
- Ensambles: combinar las importancias.
- En regresiones: Utilizar regularización (lasso, ridge, etc) y mirar los pesos.

Riesgo 1: ¿Estamos eliminando los features menos informativos o los más usados por el modelo particular?

Riesgo 2: ¿Si elimin——áramos un atributo malo, el resto quedaría igual?

Selección de Features

Idea: Reducir la dimensión mediante eliminación de variables poco útiles

Métodos univariados
Ranking generado a través de algún método estadístico.

Métodos iterativos
Considera el problema de selección de features como un problema de búsqueda.

Basados en importancia de atributos
Rankear variables según métodos internos de cada algoritmo

Basados en importancia de atributos

Algoritmo **RFE (Recursive Feature Elimination)**

- Entrenar un modelo (árbol de decisión por ejemplo)
- Obtener importancias a partir de un modelo.
- Eliminar **la / las variables menos importantes**
- Repetir desde el paso 1 incluido.

Si un grupo de variables comparte información, este mecanismo irá eliminando variables correlacionadas hasta dejar la más informativa.

Pipelines

https://scikit-learn.org/stable/modules/compose.html

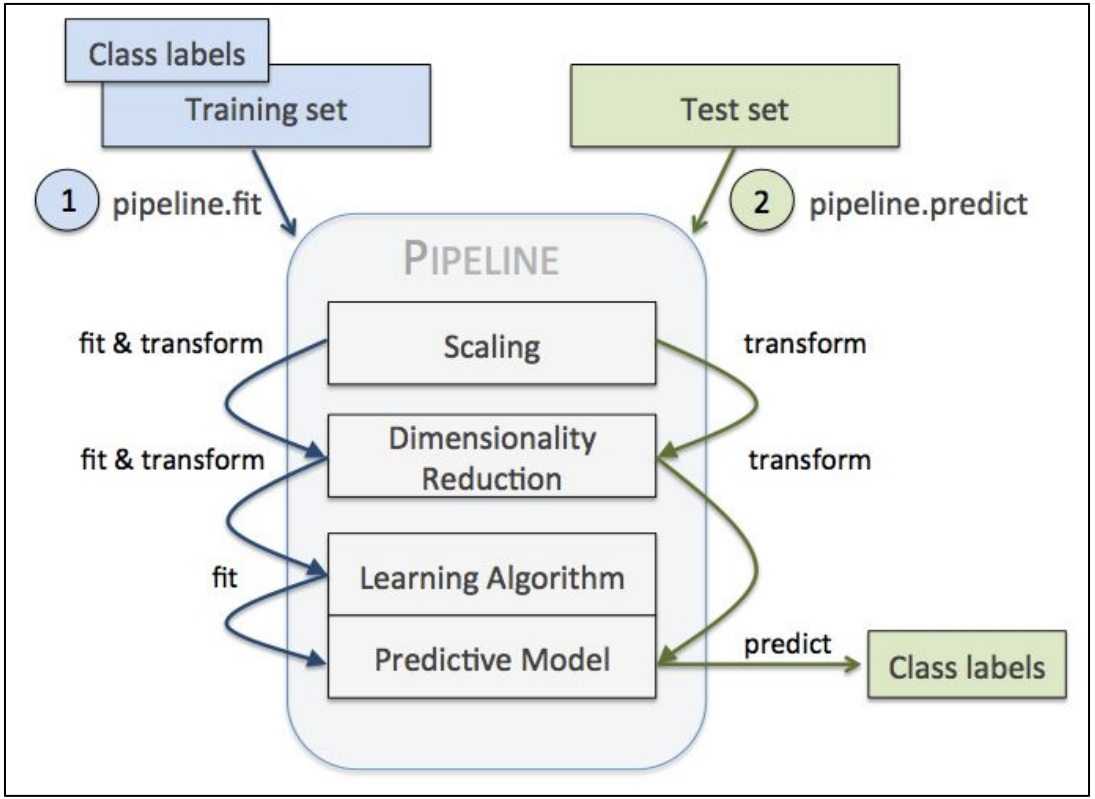
Pipelines

```
from sklearn.feature_selection import SelectKBest, f_classif

# Crear un pipeline con selección de atributos, un escalador y modelo de
regresión logística
pipeline = Pipeline([
    ('select', SelectKBest(score_func=f_classif, k=2)),
    ('scaler', StandardScaler()),
    ('log_reg', LogisticRegression())
])

# Ajustar el pipeline en los datos de entrenamiento
pipeline.fit(X_train, y_train)

# Predecir en los datos de prueba
y_pred = pipeline.predict(X_test)

# Evaluar el modelo
accuracy = pipeline.score(X_test, y_test)
print(f'Precisión (con selección de atributos): {accuracy}')
```

Permiten encadenar múltiples pasos de procesamiento en un solo objeto que se puede usar para ajustar y predecir.

Esto es particularmente útil para preprocesar datos, ajustar modelos y ajustar hiperparámetros de una manera limpia y repetible.

Pipelines



Raschka, Sebastian. Python machine learning. Birmingham, UK: Packt Publishing, 2015. Print)

Permiten encadenar múltiples pasos de procesamiento en un solo objeto que se puede usar para ajustar y predecir.

Esto es particularmente útil para preprocesar datos, ajustar modelos y ajustar hiperparámetros de una manera limpia y repetible.

TAREA

- Leer capítulo 5 (feature engineering) del libro "Designing Machine Learning Systems: An Iterative Process for Production-Ready Applications" (Chip Huyen). Todo lo que no hable de embeddings

- Leer capítulo 6 (Algorithm Chains and Pipelines) del libro "Introduction to machine learning with Python: a guide for data scientists" (Müller, Andreas C., and Sarah Guido)

- Cuestionario