ntic master
revolucionamos la comunicación

UNIVERSIDAD COMPLUTENSE MADRID

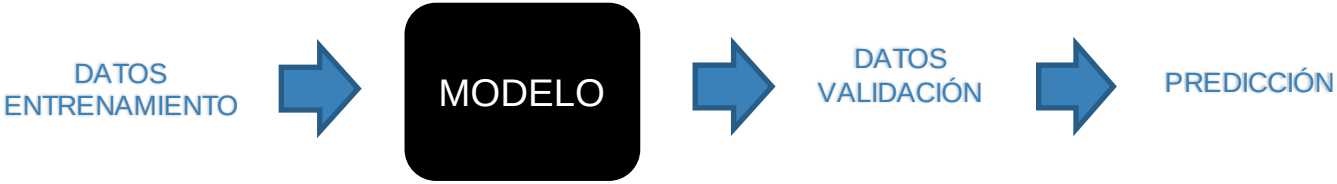
# Machine learning con Python

José Ángel Carballo Sánchez

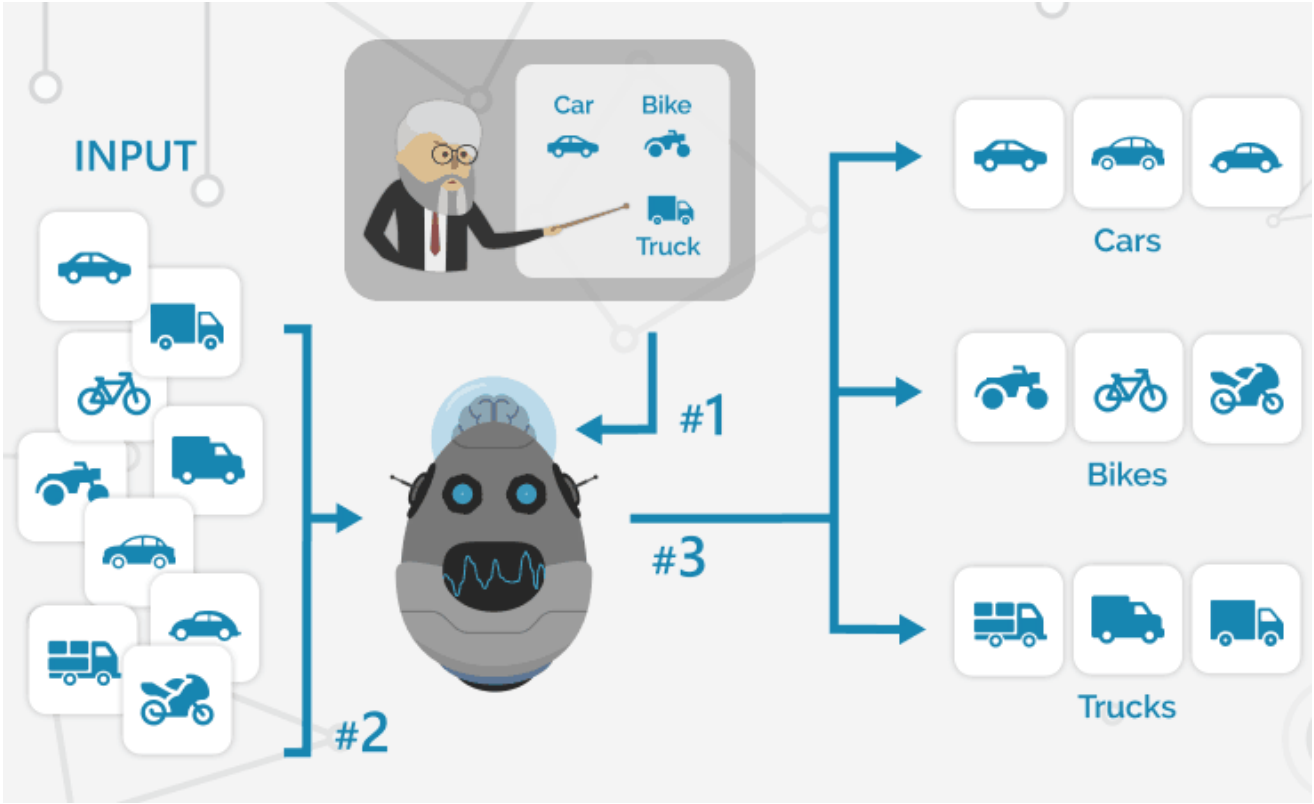
# ¿Qué es un modelo?

- Se trata de algoritmos utilizados para generalizar comportamientos e inferencias para un conjunto más amplio.

  - Utilizan datos de entrada para entrenarse.
  - Un modelo es una abstracción de los datos que se han utilizado.
  - Un caso particular es la memorización completa de los datos.

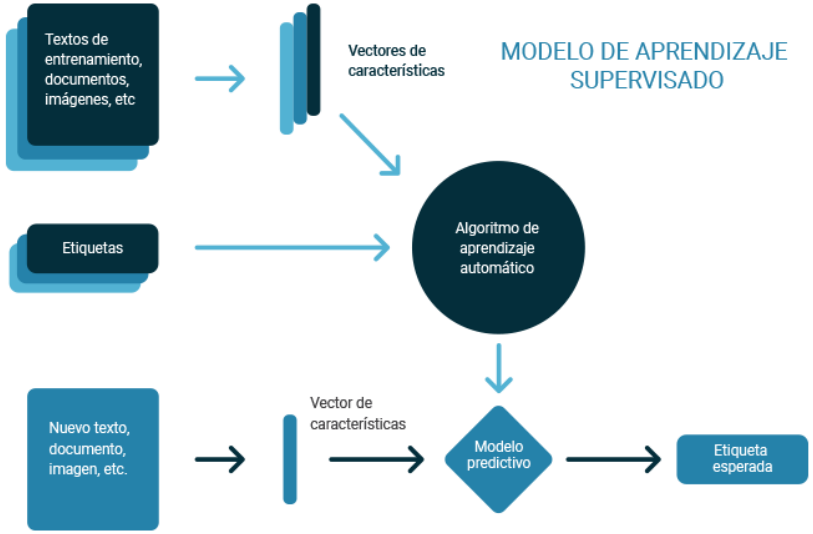


# ¿Qué es un modelo?



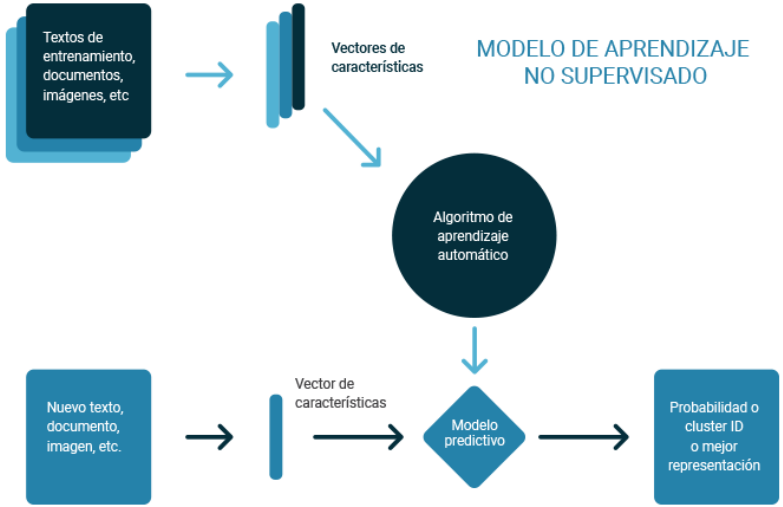
# Tipos de aprendizaje

- Supervisado
  - Se genera un modelo predictivo basado en datos entrada/salida
  - Tenemos datos previamente etiquetados/clasificados (conocimiento a priori)
  - Ejemplos: SVM, decision tree, logistic regression, k-nearest…



# Tipos de aprendizaje

- No supervisado
  - Ajustan su modelo predictivo únicamente con los datos de entrada
  - No tenemos datos previamente etiquetados/clasificados
  - Ejemplos: k-medias, hierarchical clustering, principal component analysis…

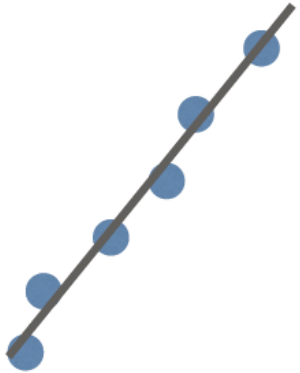


# Problemas en aprendizaje automático

- Los modelos se deben seleccionar en función de los datos y el problema a resolver.

clasificación

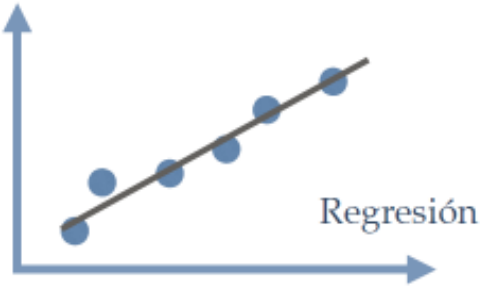
clustering



regresión

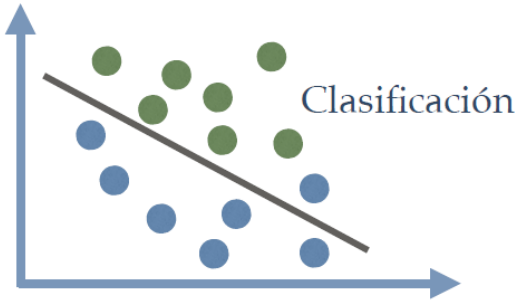
# Problemas en aprendizaje automático

- **Regresión:**
  - Los algoritmos son utilizados para aprender a predecir el valor de una variable continua a partir de una o más variables explicativas.
  - El resultado es concreto y numérico.
  - Algoritmos: regresión lineal/no, random forest, SVM…
  - Ejemplos:
    - Queremos saber el precio de venta de algo (una casa).
    - Queremos saber el tiempo que se tarda en algo (empresa de mercancías).
    - Queremos saber el número de productos que venderemos.



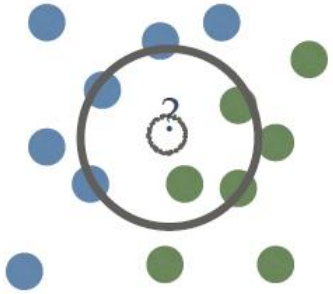
# Problemas en aprendizaje automático

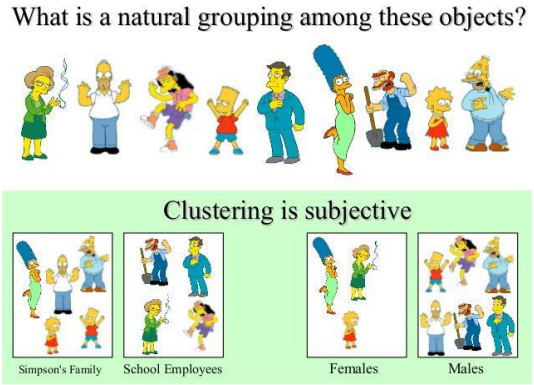
- **Clasificación**:
  - Los algoritmos son utilizados para aprender a predecir valores discretos a partir de una o más variables explicativas.
  - El resultado es una clase (a través de probabilidades…)
  - Algoritmos: igual que en regresión
  - Ejemplos:
    - ¿El cliente se irá de la compañía? SI/NO
    - ¿Tipo de enfermedad? CANCER/GRIPE
    - ¿Quién meterá el gol? DELANTERO/MEDIOCENTRO/DEFENSA



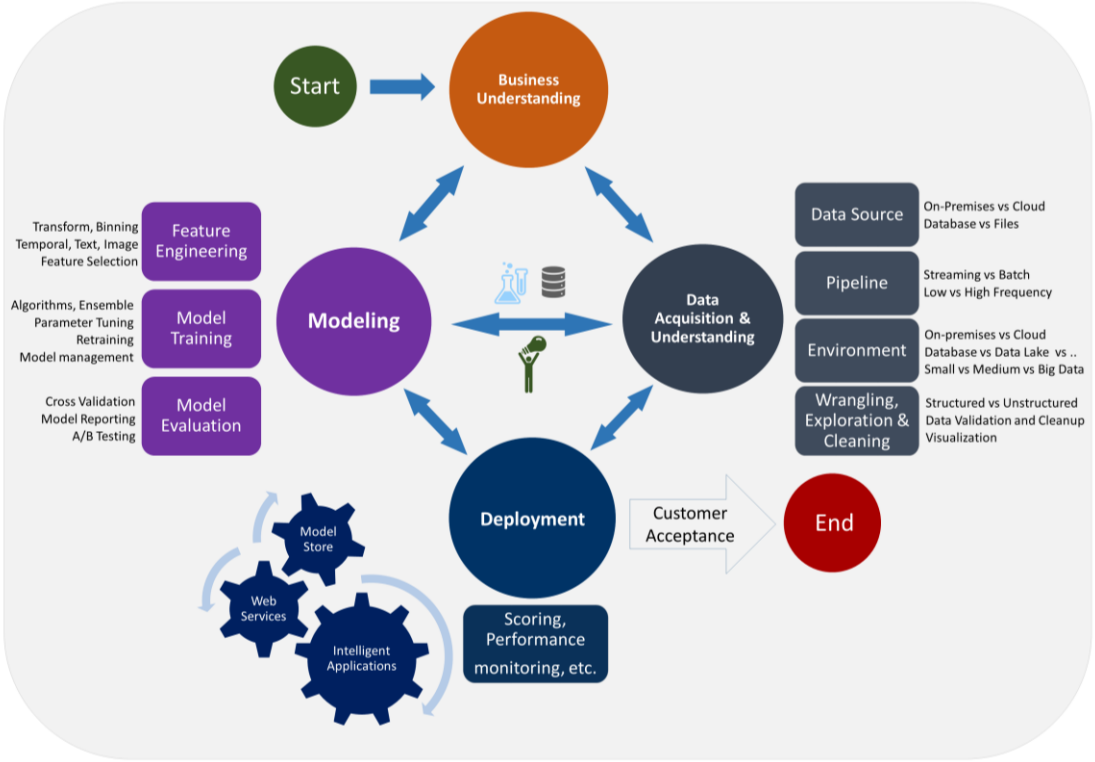
# Problemas en aprendizaje automático

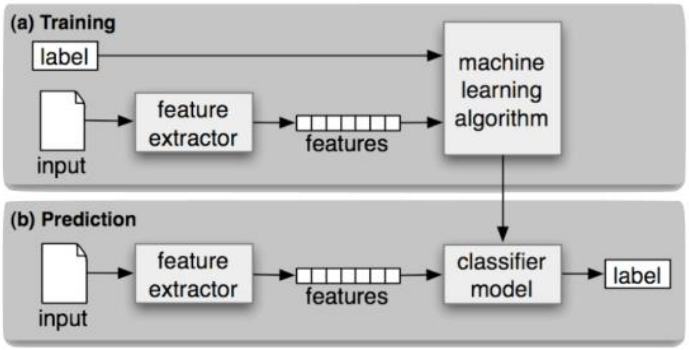
- **Clustering**:
  - Los algoritmos son utilizados para segmentar datos en grupos iguales o similares.
  - El resultado son diferentes clusters/grupos.
  - Algoritmos: kmeans, DBSCAN
  - Ejemplos:
    - ¿Qué clientes se parecen más entre ellos?
    - ¿Qué grupos de clientes tengo en mi compañía?





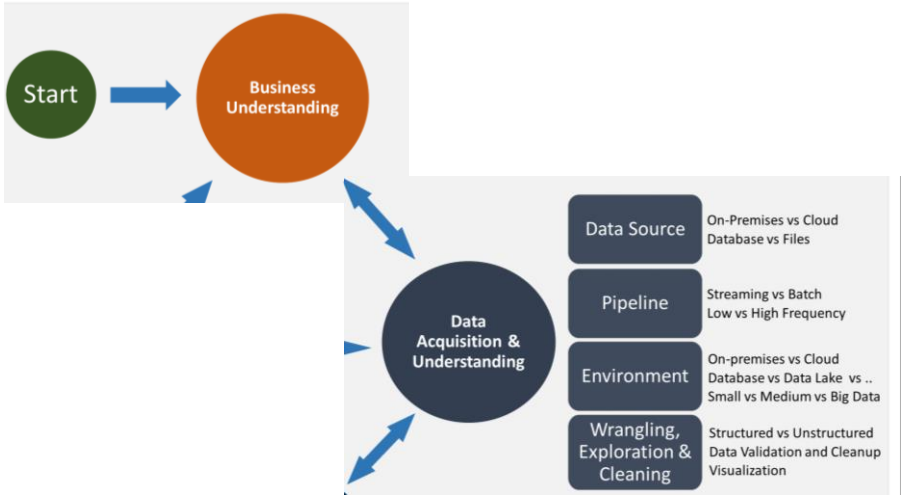
# Proceso de un proyecto de ML



# Proceso de un proyecto de ML: ejercicio

- **Bank Marketing DataSet**
  - https://archive.ics.uci.edu/ml/datasets/bank+marketing
  - bank-full.csv with all examples and 17 inputs, ordered by date



# Proceso de un proyecto de ML: ejercicio

- Features:

**age** (numeric)
**job** : type of job (categorical: "admin.","unknown","unemployed","management","housemaid","entrepreneur","student","blue-collar","self-employed","retired","technician","services")
**marital** : marital status (categorical: "married","divorced","single"; note: "divorced" means divorced or widowed)
**education** (categorical: "unknown","secondary","primary","tertiary")
**default**: has credit in default? (binary: "yes","no")
**balance**: average yearly balance, in euros (numeric)
**housing**: has housing loan? (binary: "yes","no")
**loan**: has personal loan? (binary: "yes","no")
# related with the last contact of the current campaign:
**contact**: contact communication type (categorical: "unknown","telephone","cellular")
**day**: last contact day of the month (numeric)
**month**: last contact month of year (categorical: "jan", "feb", "mar", ..., "nov", "dec")
**duration**: last contact duration, in seconds (numeric)
# other attributes:
**campaign**: number of contacts performed during this campaign and for this client (numeric, includes last contact)
**pdays**: number of days that passed by after the client was last contacted from a previous campaign (numeric, -1 means client was not previously contacted)
**previous**: number of contacts performed before this campaign and for this client (numeric)
**poutcome**: outcome of the previous marketing campaign (categorical: "unknown","other","failure","success")
# Output variable (desired target):
**Y**: has the client subscribed a term deposit? (binary: "yes","no")

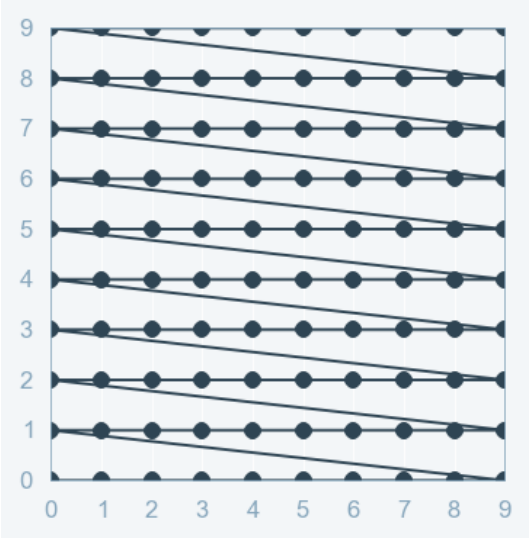
# Mejora de modelos

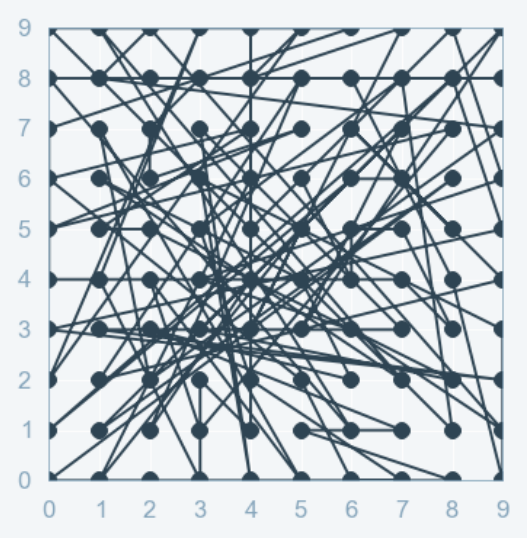
- **Feature Engineering**: en un término utilizado en proyectos de data science a la mejora de modelos a través de la creación de nuevas variables, agrupación de ellas… Este módulo, se podría incluir en el pre procesado de datos, pero normalmente se incluye después de conseguir el score de un modelo con el ánimo de realizar una mejora.

| | y |
|---|---|
| y | 1 |
| duration | 0.394521 |
| poutcome_success | 0.306788 |
| poutcome | 0.213476 |
| poutcome_unknown | 0.167051 |
| contact_unknown | 0.150935 |
| housing | 0.139173 |
| contact_cellular | 0.135873 |
| month_mar | 0.129456 |
| month_oct | 0.128531 |
| month_sep | 0.123185 |
| pdays | 0.103621 |

# Mejora de modelos

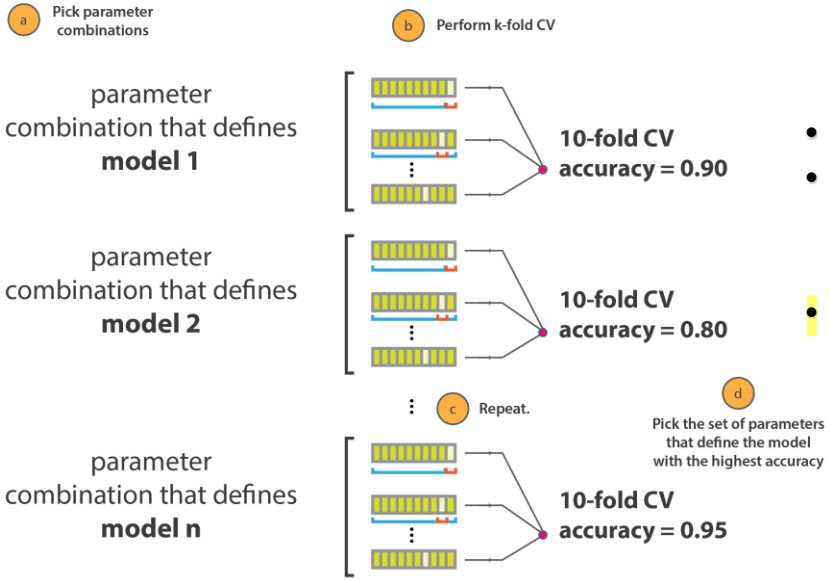
- **Tuneo de hiperparámetros**: realiza una selección de hiper-parámetros



# Mejora de modelos

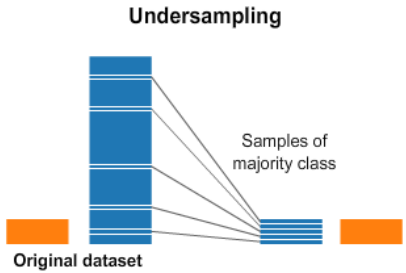
- Grid Search: técnica usada para la selección de hiper-parámetros del entrenamiento de un modelo. Se realiza mediante los siguientes parámetros:

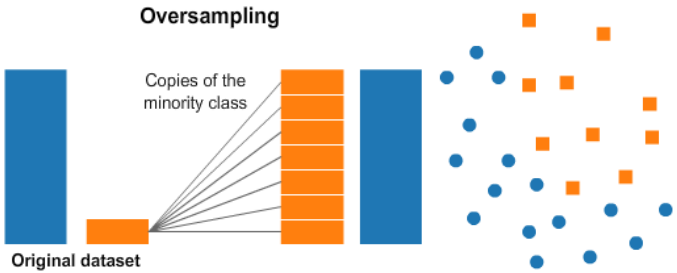


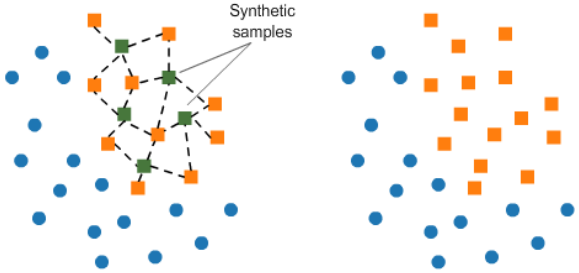

- **estimator**: el modelo que se ha de evaluar
- **param_grid**: un diccionario donde se indican los parámetros a evaluar como clave y el conjunto elementos como valor
- **cv**: el número de conjuntos en los que se divide los datos para la validación cruzada.

# Mejora de modelos

- Técnicas de balanceo de datos: cuando nuestra variable target no tiene una muestra significativa, se pueden utilizar este tipo de técnicas de balanceo de datos. Entre ellas, nos encontramos con:
  - Undersampling: consiste en seleccionar un porcentaje de muestras de la clase mayoritaria. cuando hay muchos datos, elimina los datos de la clase mayoritaria
  - Oversampling: consiste en duplicar un porcentaje de muestras de la clase minoritaria. Genera muestras sintéticas con algoritmos



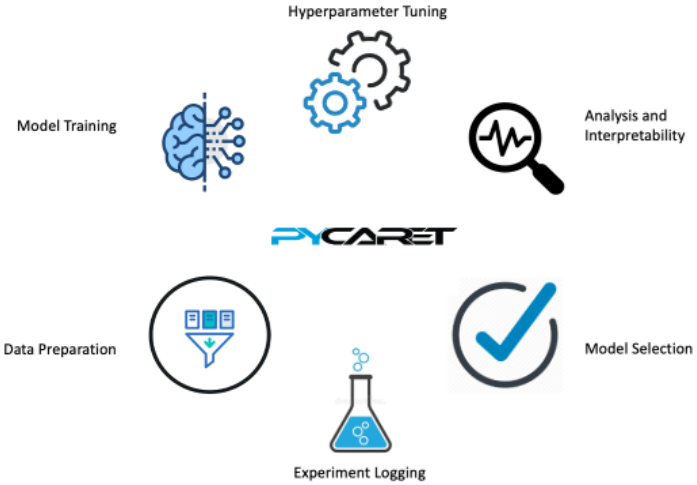



# Herramientas/librerías que pueden ayudar

- **AUTOML:**

**Pycaret:**

- Nos ofrece una primera aproximación rápida al modelo que mejor funcionaría dentro de una amplia lista de algoritmos.
- Permite llevar a cabo desde la preparación de los datos, hasta el despliegue del modelo final en tan solo unos minutos.
- No sustituye al Data Scientist, pero nos da una primera idea del potencial de los datos, de los algoritmos que mejor se ajustan a ese dataset…
- Es compatible con jupyter notebook

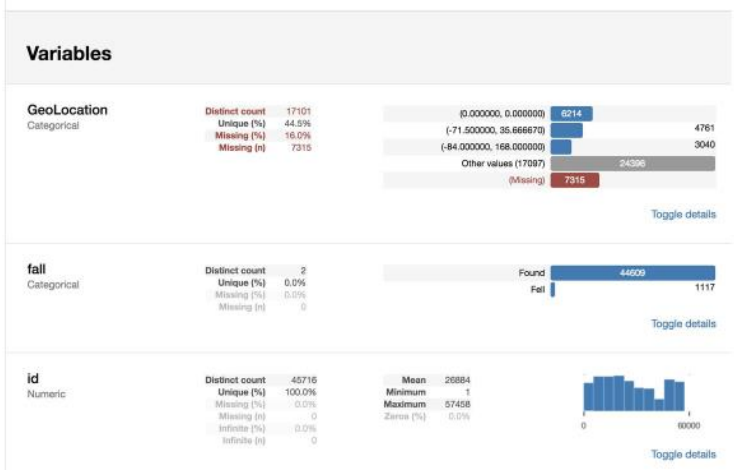


# Herramientas/librerías que pueden ayudar

- **EDA:**

**Pandas profiling:**

- Nos ofrece una primera aproximación rápida al análisis de variables.
- Realiza un EDA en formato html en cuestión de minutos.
- Podemos ver correlación entre variables, número de nulos, tipo de variables e incluso podemos visualizar la distribución de cada una de ellas.

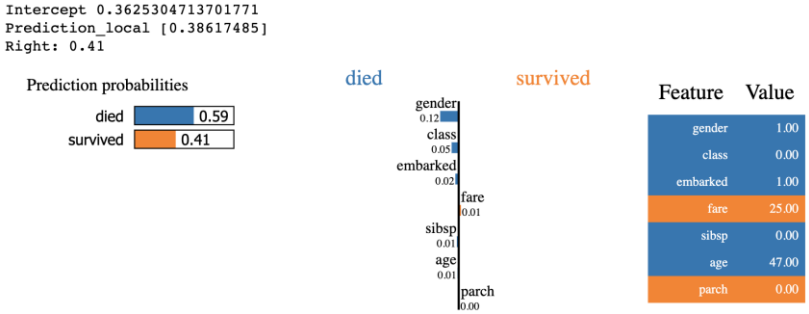


# Herramientas/librerías que pueden ayudar

- **EXPLICACIÓN DE MODELOS:**

**Lime:**

- Conocer bien el comportamiento del modelo, así como las variables más influyentes nos ayudan a poder mejorar el mismo y saber hasta donde puede llegar.
- Además nos permite explicar al resto de personas (no técnicas) el porqué funciona de una determinada manera y respaldar su uso.
- LIME es una librería que nos ayuda a realizar todo esto de manera sencilla y rápida.



# Herramientas/librerías que pueden ayudar

- **DESPLIEGUE DE MODELOS:**

**Mlflow:**

- MLflow es una plataforma de código abierto para gestionar el ciclo de vida de ML de principio a fin.

- MLflow se desarrolló para poder funcionar con cualquier lenguaje de programación, librerías, algoritmos y herramientas de implementación.

- Se basa en una interfaz API REST

# Herramientas/librerías que pueden ayudar

- **Grandes conjuntos de datos**

**H2O:**

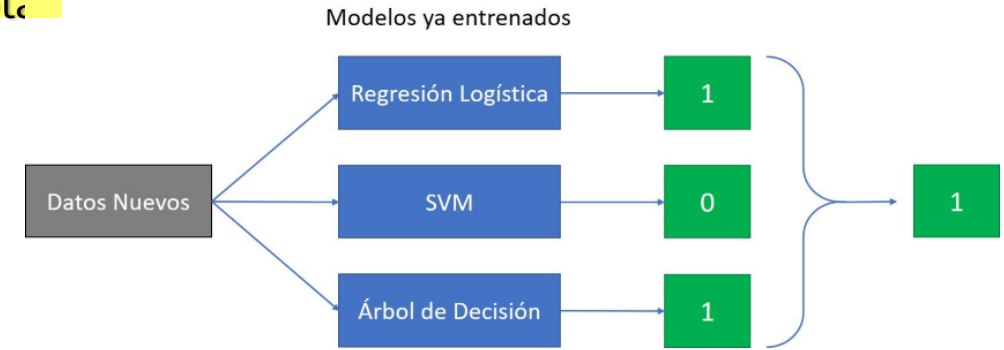
- Es una plataforma machine Learning open-source desarrollada en Java que ofrece un gran conjunto de algoritmos de machine learning con alta capacidad de procesamiento (cluster)

- Gracias a su forma de comprimir y almacenar los datos, H2O es capaz de trabajar con millones de registros en un único ordenador (emplea todos sus cores) o en un cluster de muchos ordenadores

# Ensemble models

Se tratan de métodos combinados que utilizan múltiples algoritmos de machine learning para obtener un mejor rendimiento predictivo.

- Voting: permite entrenar varios modelos con los mismos datos. Cada modelo tiene asociado un voto. La predicción final se obtiene como la más vota

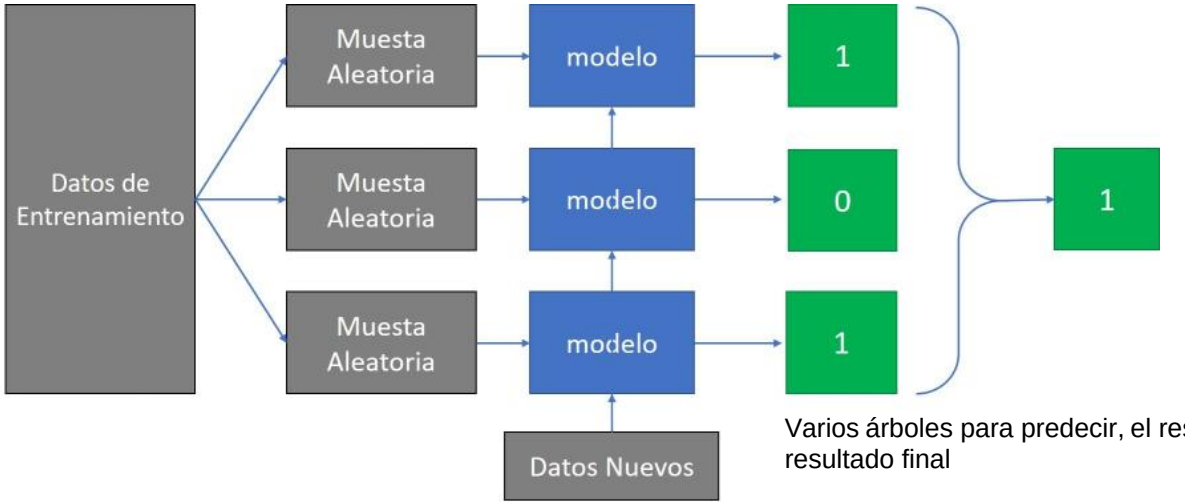



- Stacking: se trata de apilar modelos, es decir, usar la salida de un modelo como entrada de otro.

# Ensemble models

- Bagging: es un meta-algoritmo que consigue combinaciones de modelos a través de una familia inicial, provocando un menos overfitting y varianza. Consigue que los errores se compensen entre sí, entrenando cada modelo con subconjuntos del dataset original. El resultado es una combinación.

Ejemplo: Random Forest (bagging de decission tree)

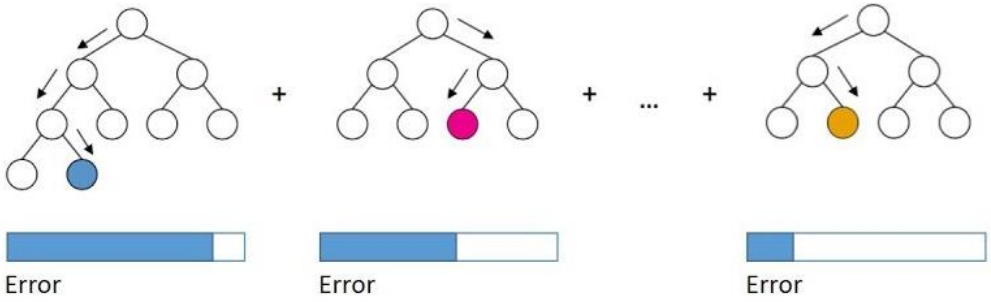



Varios árboles para predecir, el resultado de la mayoría es el resultado final

# Ensemble methods

- Boosting: en este caso, cada modelo intenta arreglar los errores de los modelos anteriores. Tras la primera clasificación, se dará más peso a las muestras mal clasificadas. En el caso de regresión se da más peso al error cuadrático medio para el siguiente modelo.

- Ejemplos: Xgboost, CatBoost, Lightgbm, AdaBoost



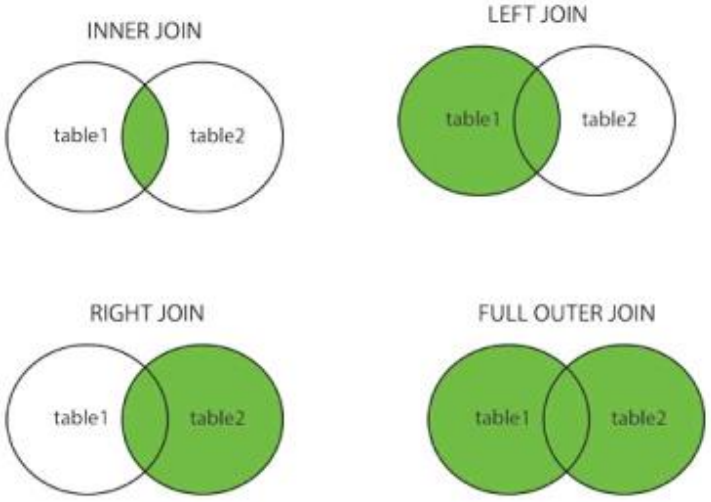

Intentar aplicar un modelo boosting a uno de los modelos realizados anteriormente (ejemplo Clasificación)

# Ejercicio real

Consiste en calcular el churn (clientes que abandonan una compañía) de clientes en una operadora de telecomunicaciones. La compañía ha descubierto que entre diciembre de 2019 y enero de 2020 han tenido bastantes bajas de clientes. El objetivo de la práctica es llegar a construir un modelo analítico que sea capaz de predecir los próximos clientes que son potenciales a marcharse de la operadora para poder lanzar una campaña e intentar fidelizarles antes de su portabilidad. Además, los directivos de la compañía quieren conocer las causas por las cuales los clientes se están fugando. Para ello disponemos de varios datasets con datos que aportan información de los clientes.

# Ejercicio real

1. Cargar los datasets y construir un único tablón analítico con todas las variables que consideremos necesarias para cada una de las cosechas que queremos analizar.



# Ejercicio real

**target**: esta variable no la tenemos. Esto ocurre más a menudo de lo que pensamos en la industria. Una vez que tenemos los dos tablones analíticos construidos (diciembre y enero) hay que pensar como podemos construir esta variable.

¡¡¡ COSECHAS !!!

# Ejercicio real

2. Realizar un pre procesado y una limpieza de los datos, siguiendo la metodología vista en clase.

3. Muestrear los datos, construyendo un dataset de train y test.

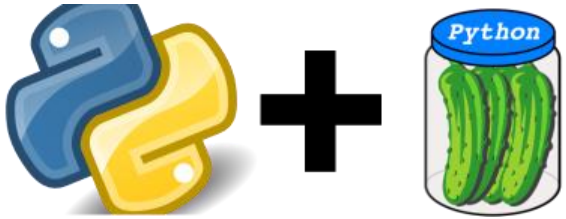
# Ejercicio real

4. Construir un modelo analítico de clasificación que sea capaz de predecir cuando un cliente se fuga de la empresa, argumentando el tipo de algoritmo utilizado, las variables seleccionadas, obteniendo las métricas oportunas… Explicar lo “bueno” o “malo” que es el modelo a través de las métricas obtenidas.

5. Realizar una mejora del modelo utilizando técnicas vistas en clase (feature engineering, análisis de variables, comprobación de overfitting, validación cruzada, tuneado de hiper parámetros a través de la automatización…)

# Ejercicio real

6. Predecir los clientes de la cosecha de enero que más probabilidad tienen de cambiarse de operadora.




| id | probabilidad |
|---|---|
| 86266 | 0.96 |
| 98899 | 0.94 |
| 19111 | 0.93 |
| 14588 | 0.93 |
| 35890 | 0.92 |
| ... | ... |

7. Obtener y explicar las claves (a través de las variables…) de la marcha de los clientes en la compañía.


UNIVERSIDAD
COMPLUTENSE
MADRID