# Algoritmos de regresión y métricas

Esther Aguilar Hervás

# ÍNDICE

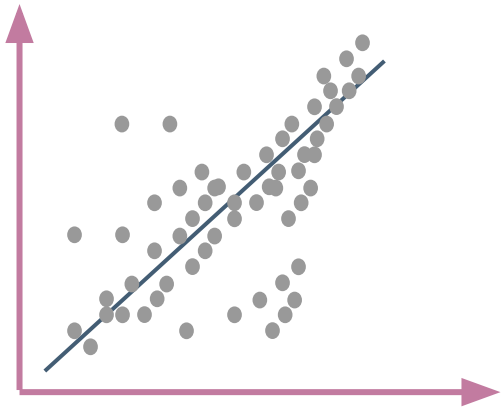
# Objetivos

- Conocer la metodología a seguir para construir modelos de ML
- Conocer diferentes algoritmos de regresión
- Aprender a evaluar algoritmos de regresión en base a métricas

# 01

# Introducción

# Regresión



**Objetivo**: establecer un método para la relación entre un cierto número de características y una variable objetivo continua

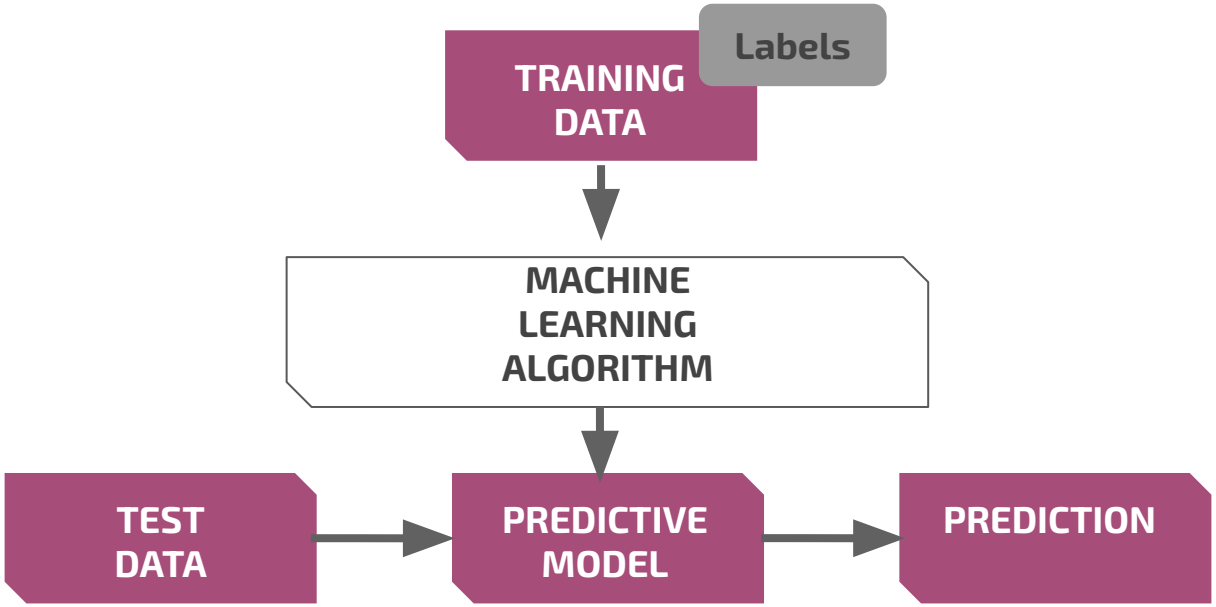
*variable dependiente* — **Y = mX + b** — *variable explicatoria*

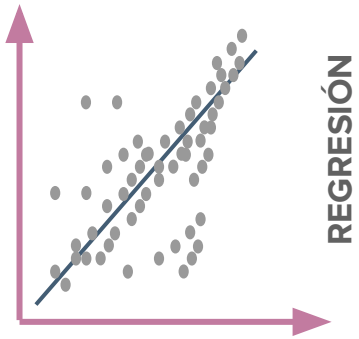
**Tipos de algoritmos supervisados**:

- Regresión lineal simple
- Regresión lineal múltiple
- Regresión polinómica

**Ejemplo**: Predicción del coste de la vivienda en función del número de habitaciones y localidad

# Regresión



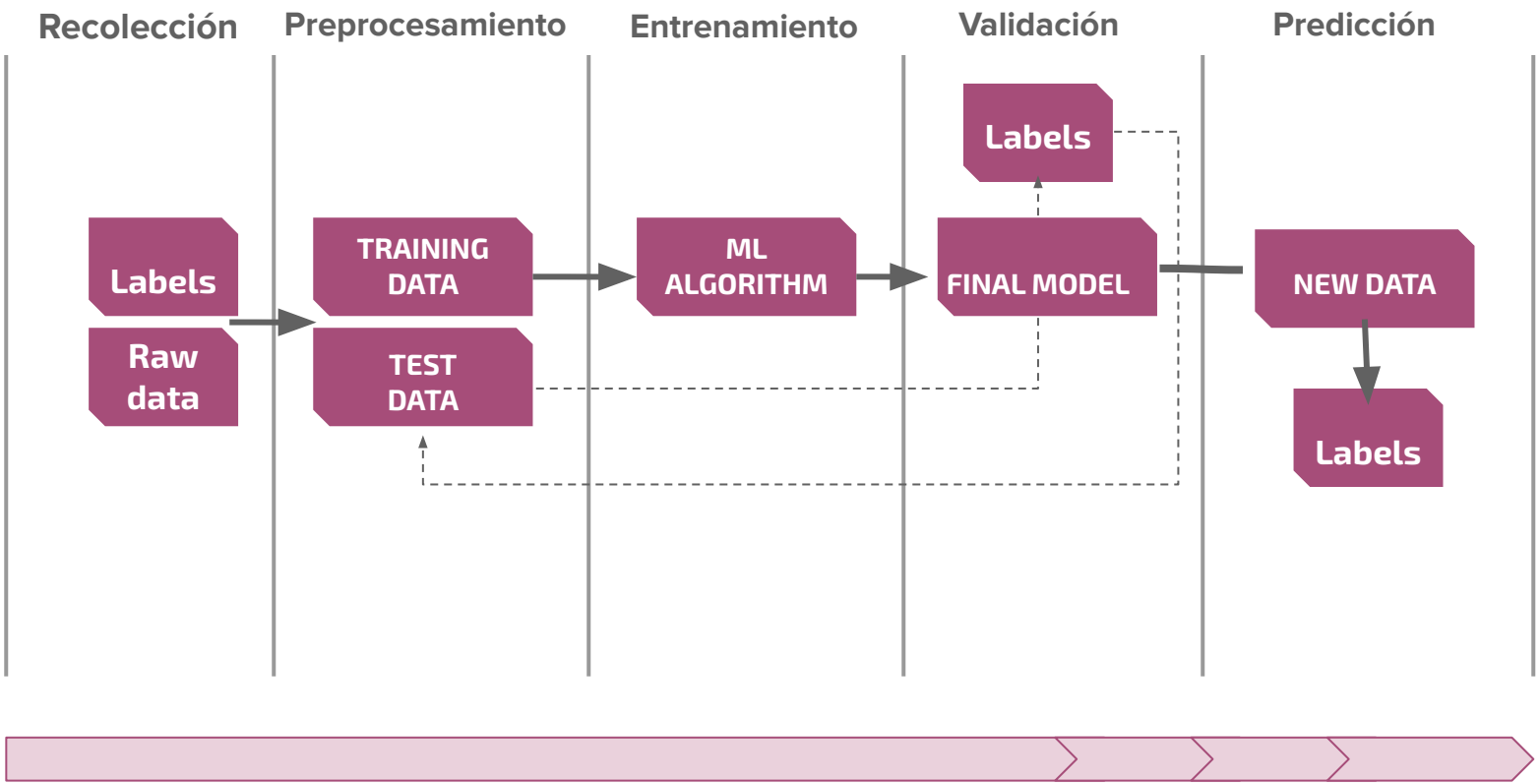




$Y = mX + b$

# 02

# Metodología

# Introducción

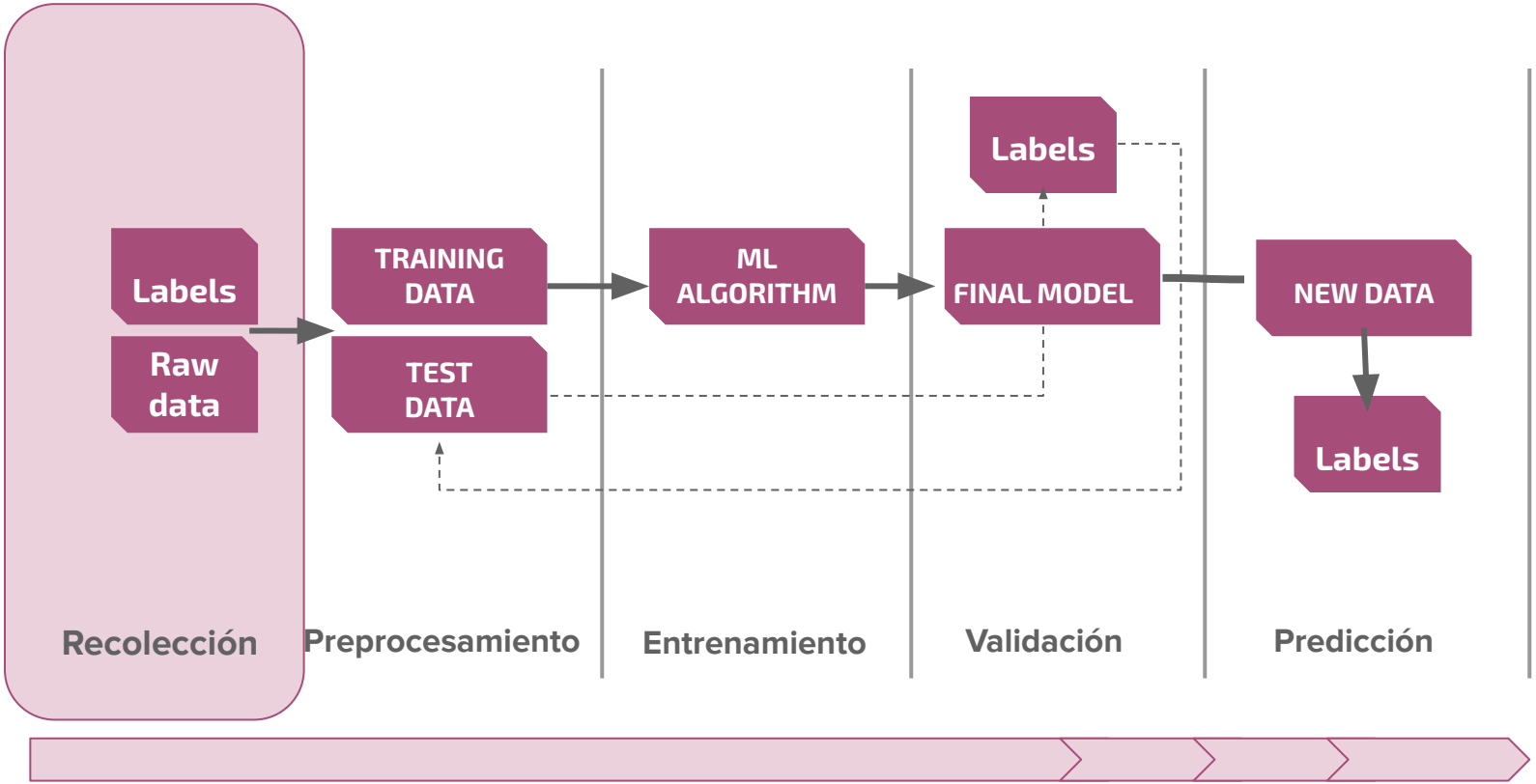


# Definición del objetivo

- ¿Qué deseamos hacer?
- ¿Cómo podremos hacerlo?
- ¿Es posible lo que deseo dada los dat
- os de los que dispongo?

**Predecir el precio de la vivienda**

# Recolección de datos



# Recolección de datos

**Google Colab**
**Fichero local**

```
# importamos fichero local
from google.colab import files
uploaded = files.upload()

# análisis exploratorio
import pandas as pd

# creamos variable con dataset
import io
house_data = pd.read_csv(io.BytesIO(uploaded['data.csv']))
```

# Recolección de datos

**Google Colab**
**Fichero Drive**

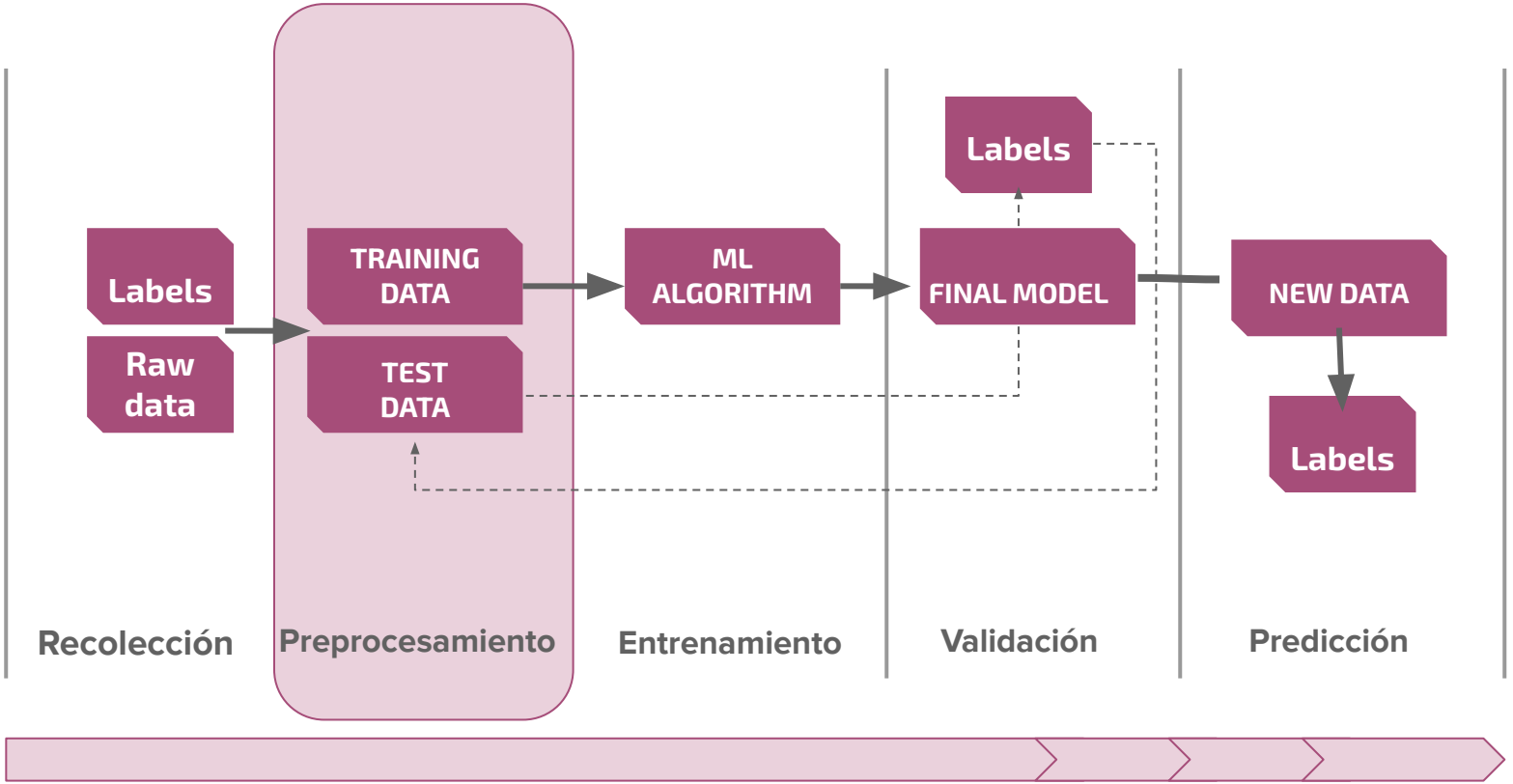
```
# configuramos importación desde Drive
!pip install -U -q PyDrive
from pydrive.auth import GoogleAuth
from pydrive.drive import GoogleDrive
from google.colab import auth
from oauth2client.client import GoogleCredentials

# autenticación y creación del cliente PyDrive
auth.authenticate_user()
gauth = GoogleAuth()
gauth.credentials = GoogleCredentials.get_application_default()
drive = GoogleDrive(gauth)

# creamos variable con el link de nuestro csv
link = 'https://drive.google.com/open?id=1BVaXEFrRIjtHgfOxLpTd01T2doUbtQ_L'
# cargamos variable id con el identificador del documento
fluff, id = link.split( '=')

# descargamos contenido del fichero
downloaded = drive.CreateFile({ 'id':id})
downloaded.GetContentFile( 'data.csv')
```

# Recolección de datos

Google Colab
Kaggle

```
# configuramos importación desde kaggle
from google.colab import files
!pip install -q kaggle

# subimos kaggle.json con el cliente
uploaded = files.upload()

# descargamos dataset
!kaggle datasets download -d shree1992/housedata

# análisis exploratorio
import pandas as pd

# creamos variable con dataset
house_data=pd.read_csv("./data.csv")
```

# Preprocesamiento



# Análisis exploratorio de datos

```
# exploración del dataset
house_data

# exploración de las primeras líneas dataset
house_data.head()

# consultamos datos estadísticos generales:num de elementos, media, desviación, valor min.,
percentiles
house_data.describe().transpose()

# consultamos el número de filas y columnas
house_data.shape

# consultamos si tenemos celdas sin valor
house_data.isnull().sum()

# nombre de las columnas
house_data.columns
```

# Análisis exploratorio de datos

```
# visualización de datos
import matplotlib.pyplot as plp
import seaborn as sns

# visualización regplot
sns.set(style="darkgrid")
sns.regplot(x=house_data['yr_built'], y= house_data["price"])

# visualización displot
sns.displot(house_data["bathrooms"],bins=10)

# visualización histogramas
house_data.hist(figsize=(15,20))
plp.show()

# visualización barras
plp.figure(figsize=(15,20))
sns.barplot(data=house_data,y='statezip',x='price',orient="h3")
```

# Preprocesamiento

```
# consultar tipo variables
house_data.dtypes

# eliminar variables
house_data = house_data.drop('date',axis=1)
house_data = house_data.drop('statezip',axis=1)

# convertir float a int
house_data["price"]=house_data["price"].astype(int)

# convertir string en int
from sklearn.preprocessing import LabelEncoder
LE=LabelEncoder()
house_data["street"]=LE.fit_transform(house_data["street"])
```

# Preprocesamiento

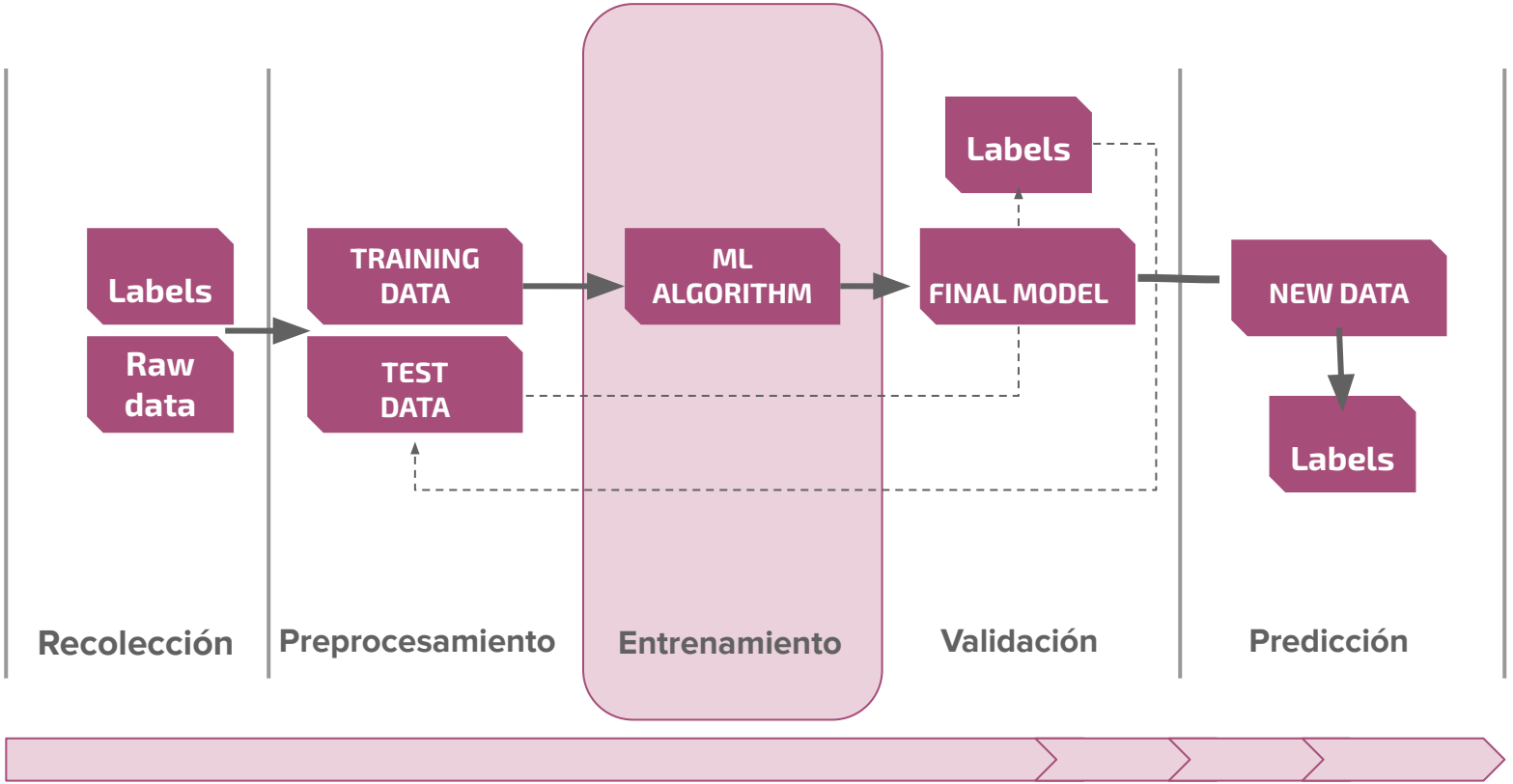
```
# obtener valores nulos
house_data.isnull().sum()

# completar valores nulos con la media
house_data.fillna(house_data.mean())

# completar valores nulos con el valor más frecuente
house_data['bedrooms'] = house_data['bedrooms'].fillna(house_data['bedrooms'].mode()[0])
```

# Elección del algoritmo

- Supervisado??
- No supervisado??
- Clasificación??
- Regresión??

**Regresión lineal**

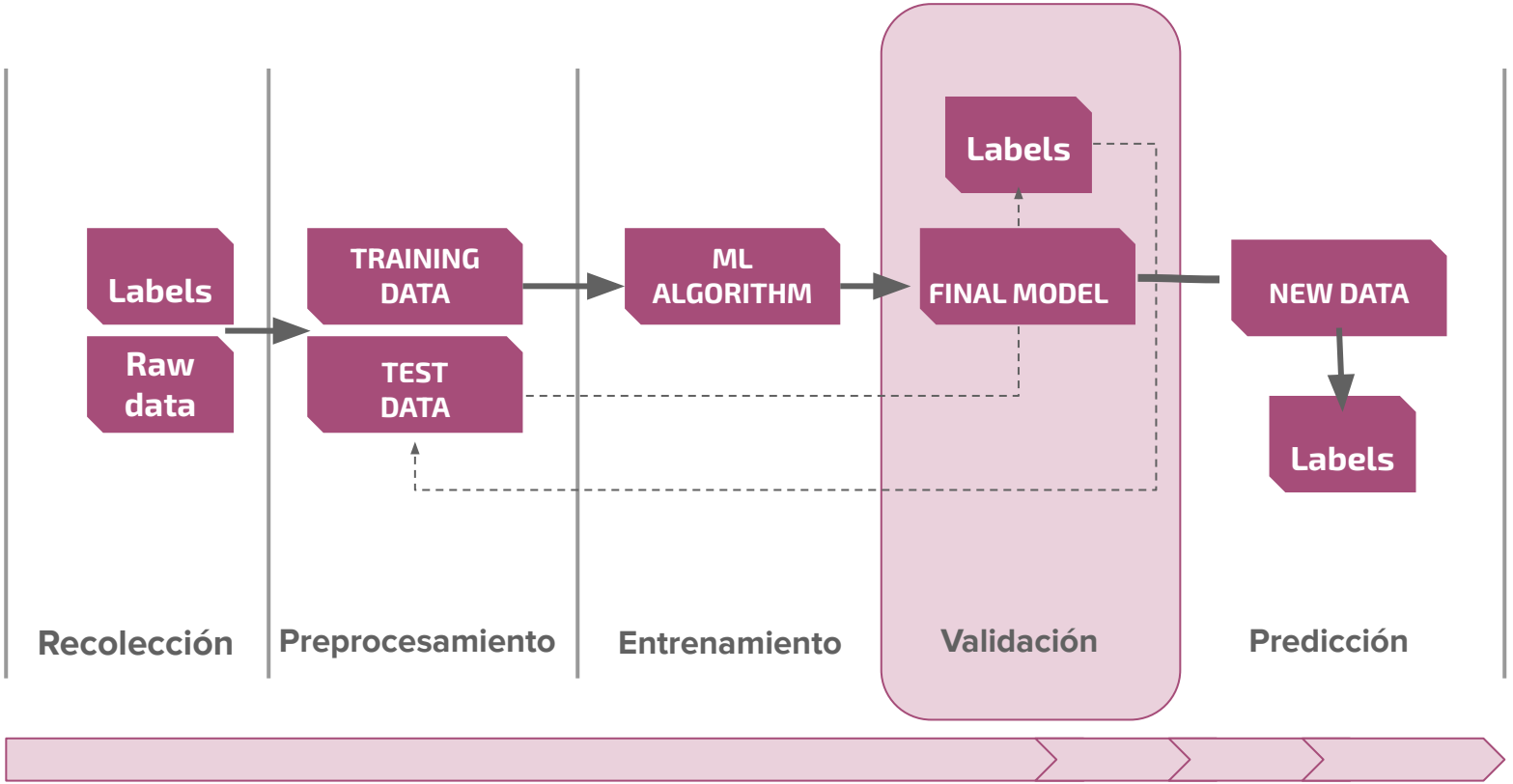
# Entrenamiento



# Entrenamiento

```
# indicamos variable objetivo
X = house_data.drop('price',axis =1).values
y = house_data['price'].values

# preparamos train data y test data
from sklearn.model_selection import train_test_split
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.20, random_state=43)

# regresión lineal
from sklearn.linear_model import LinearRegression
lin_reg=LinearRegression()
lin_reg.fit(X_train, y_train)
```

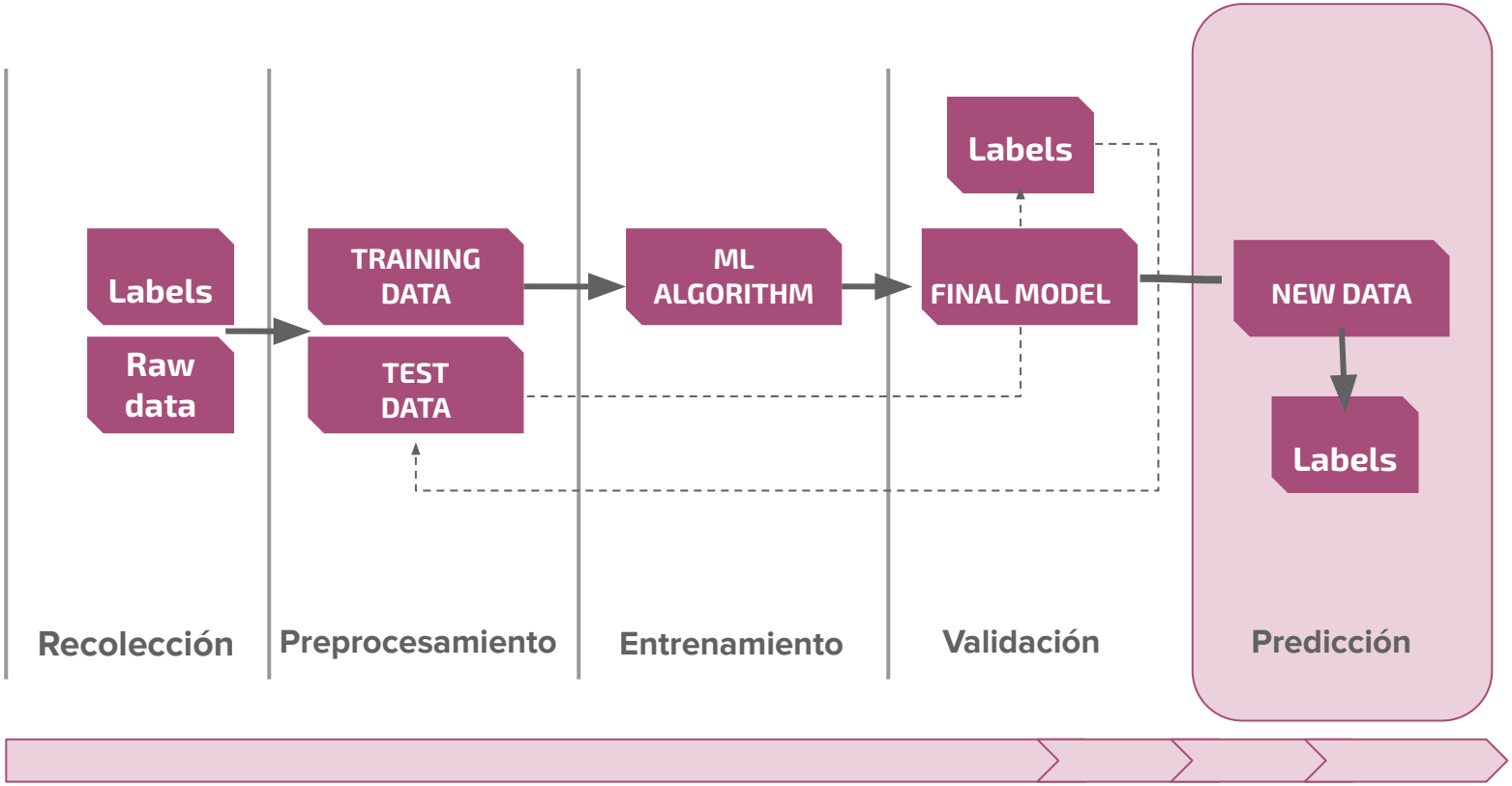
# Validación del modelo



# Validación del modelo

```
# regresión lineal
y_pred=lin_reg.predict(X_test)

# consultar accuracy
print("Training acc >> ", lin_reg.score(X_train, y_train))
print("Testing acc >>  ", lin_reg.score(X_test, y_test))

# validación cruzada
from sklearn.model_selection import cross_val_score
lin_reg = LinearRegression()
cvscores_10 = cross_val_score(lin_reg, X, y, cv= 10)
print(np.mean(cvscores_10))

# calculamos el Error Cuadrático Medio (MSE = Mean Squared Error)
from sklearn.metrics import mean_squared_error
mse_hipot1_train = mean_squared_error(y_true = y_hipot1_train, y_pred = prediccion_entrenamiento)
print('Error Cuadrático Medio (MSE) TRAIN= ' + str(mse_hipot1_train))
```

# Predicción


Labels
Raw data
TRAINING DATA
TEST DATA
ML ALGORITHM
Labels
FINAL MODEL
NEW DATA
Labels
Recolección
Preprocesamiento
Entrenamiento
Validación
Predicción

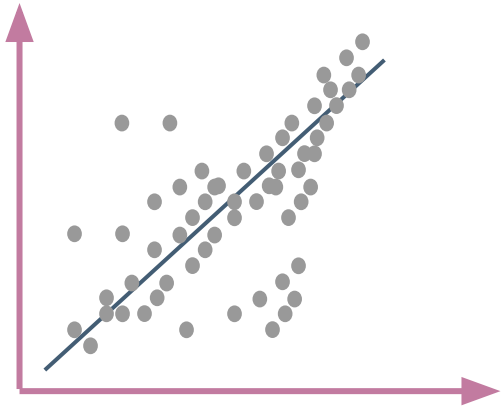
# Predicción

```
# regresión lineal
from sklearn.linear_model import LinearRegression
lin_reg=LinearRegression()
lin_reg.fit(X_train, y_train)

y_pred=lin_reg.predict(X_test)
```

# 03

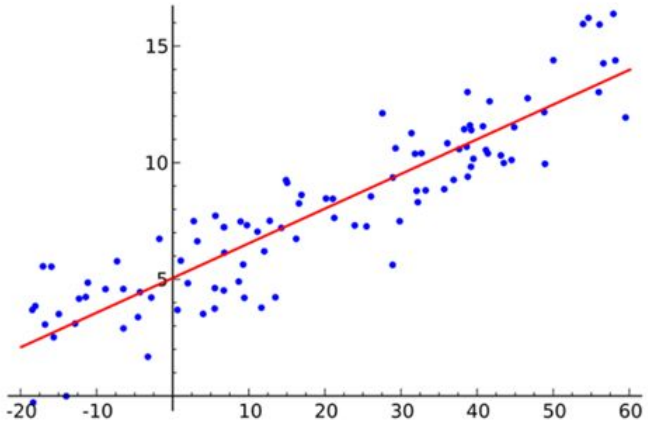
# Algoritmos de regresión

# Algoritmos de regresión



- Regresión lineal simple
- Regresión lineal múltiple
- Regresión polinómica

# Regresión Lineal Simple




$$Y = mX + b + e$$

**Variable predictora**
**Variable independiente**
→ **MODELO** →
**Variable a predecir**
**Variable dependiente**

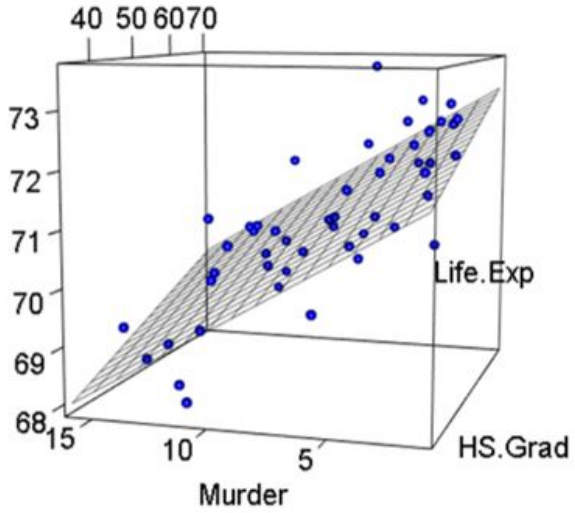
En este modelo se fija:

- Y – Variable que se quiere predecir (variable dependiente)
- X – Variable predictora (variable independiente)

Determina:

- m – Pendiente
- b – Término independiente
- e – Error cometido en la predicción

# Regresión Lineal Múltiple




$$Y = m_1X_1 + m_2X_2 + b + e$$

**Variables predictoras**
**Variables independientes**

→ **MODELO** →

**Variable a predecir**
**Variable dependiente**

En este modelo se fija:

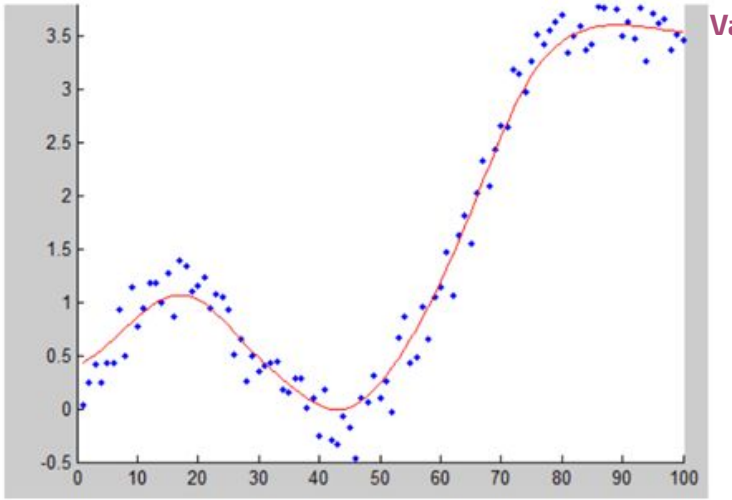
- Y – Variable que se quiere predecir (variable dependiente)
- X1 , X2... X n – Variables predictoras (variables independientes)

Determina:

- X1 , X2... X n – Término independiente
- e – Error cometido en la predicción

# Regresión Polinómica

$$Y = aX^2 + bX + c + e$$




**Variables predictoras**
**Variables independientes**

→ **MODELO** →

**Variable a predecir**
**Variable dependiente**

En este modelo se fija:

- Y – Variable que se quiere predecir (variable dependiente)
- X – Variable predictora (variable independiente)

Determina:

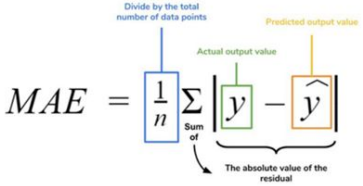
- a y b - Pendientes
- c – Término independiente
- e – Error cometido en la predicción

# 04

# Métricas

# Métricas

## Error medio absoluto (MAE)



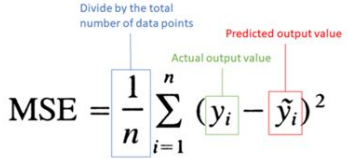

Media de los errores absolutos entre los valores observados y los predichos

## Raíz del error cuadrático medio (RMSE)

$$\text{RMSE} = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2}$$

Es la raíz cuadrada del error cuadrático medio

## Error cuadrático medio (MSE)




$$\text{MSE} = \frac{1}{n}\sum_{i=1}^{n}(y_i - \tilde{y}_i)^2$$

Es la media de las diferencias cuadradas entre los valores observados y los predichos.

## Coeficiente R^2

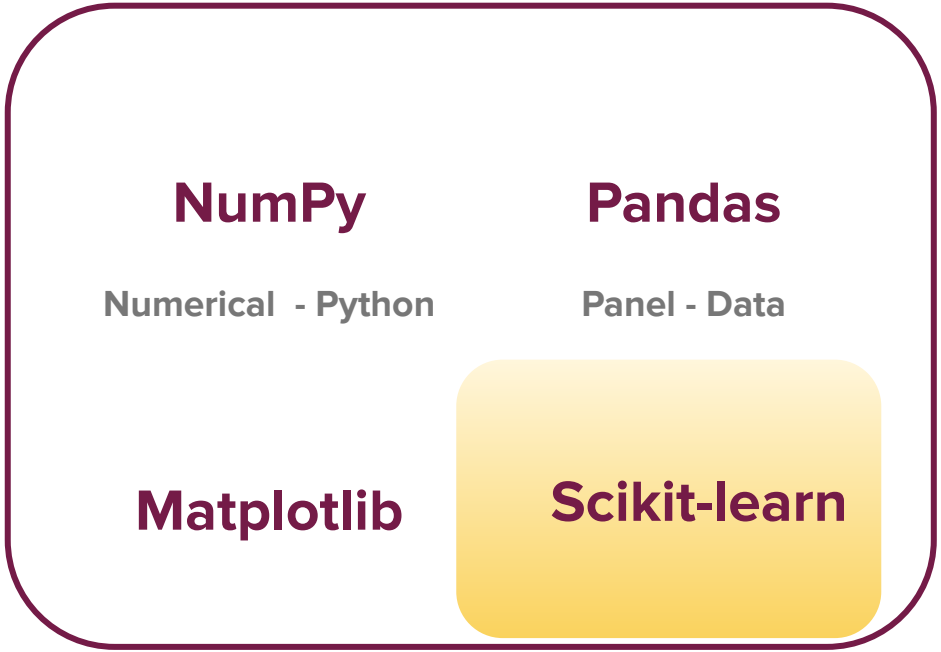
$$R^2 = 1\frac{SS_{RES}}{SS_{TOT}} = 1\frac{\Sigma_i (y_i - \hat{y}_i)^2}{\Sigma_i (y_i - \bar{y}_i)^2}$$

Porcentaje de variación entre la variable predecida y las variables componentes del modelo

# 05

# Ejemplos prácticos

# Librerías Python



# Librerías Python - Scikit-learn

```
# importar libreria sklearn.linear_model
from sklearn.linear_model import LinearRegression

x_train=variables_independientes_train_set
y_train=variables_dependientes_train_set

x_test=variables_independientes_test_set
y_test=variables_dependientes_test_set

# Regresion lineal
regresion_lineal=LinearRegression()

# Entrenamos el modelo
regresion_lineal.fit(x_train, y_train)

# Realizamos la predicción
regresion_lineal.predict(x_test)
```

# Librerías Python - Scikit-learn

```
# Regresion lineal
regresion_lineal=LinearRegression()
regresion_lineal.fit(x_train, y_train)
prediccion=regresion_lineal.predict(x_test)

# Fase de validación
# importamos el cálculo del error cuadrático medio (MSE)
from sklearn.metrics import mean_squared_error

# calculamos el Error Cuadrático Medio (MSE = Mean Squared Error)
mse_train = mean_squared_error(y_true = y_test, y_pred = prediccion)
```

# Ejemplos Regresión Lineal

- **Regresión Lineal Simple**
- **Regresión Lineal Múltiple**

# ¡Gracias!

LinkedIn: in/estheraguilarhervas/