A. How to import datasets

To import a data from CSV document we use the function **read_csv** from **pandas** library

On Python:

Import pandas as pd

Data=pd.read_csv('file.csv')

Remark: we use the same method to import Excel docs

B. How to remove data from pandas dataframe

To do it we use the function **drop ()** and the **parameter axis such axis=1 to remove a columns and axis=0 to remove lines**

Remark : This is a **very common pattern** when preparing data for a machine learning model:

On Python:

X = data.drop('target', axis=1) X contains **the features** (inputs to the model)

y = data['target'] y contains **the target variable** (the output we want to predict)

C. How to predict using regression model on python:

Regression is a type **of supervised** machine learning used to **predict a continuous value**. Like: Predict someone's **salary** from their **age** and **experience**

The general linear regression formula is

$$\hat{y}=a_1x_1+a_2x_2+...+a_nx_n+b$$

Where:

$\hat{y}$: the predicted value (for example, predicted **salary**)

$x_1,x_2,...,x_n$ the input features (for example,**age** and **experience**)

$a_1,a_2,...,a_n$: the coefficients (weights) learned by the model

b : the intercept (constant term)

The goal is to find a **relationship** between *input variables (features)* (**age** and **experience)** and *output variable (target) (***salary)**.

Steps:

1. **Import the Libraries** import pandas and sklearn
2. **Create a Simple Dataset (CSV, EXCEL, txt,...)**
3. **Separate Inputs and Output** X = what we use to predict and y = what we want to predict
4. **Split the Data into:** Training set (to teach the model) and Test set (to check how good it is) to do so we use **train_test_split** function from **scikit-learn library**

On python:

```
from sklearn.model_selection import train_test_split
X_train, X_test, y_train, y_test = train_test_split( X, y, test_size=0.3, random_state=42)
print("Training set size:", len(X_train))
print("Testing set size:", len(X_test))
```

Remark: the parameters:

- random_state=1000 to use the same random_state (same split) 1000 times
- len() is a function that means “length.”
- test_size=0.3 Means that 30% of the samples go to the test set, and the remaining 70% go to training.
-

5. Create the Regression Model

we create a **Linear Regression** model.

On python :

```
from sklearn.linear_model import LinearRegression

model = LinearRegression()
```

6. **Train the Model** we train (fit) the model using our training data (X_train).

On python :

```
model.fit(X_train, y_train)

print("Coefficients:", model.coef_)

print("Intercept:", model.intercept_)
```

7. **Make Predictions** we use the trained model to make predictions on the test data.

On python :

```
y_pred = model.predict(X_test)

print("Predicted values:", y_pred)
```

8. **Evaluate the Model:** we use Mean Squared Error (MSE) metric.

On python :

```
from sklearn.metrics import mean_squared_error

mse = mean_squared_error(y_test, y_pred)

print("Mean Squared Error:", mse)
```

Remark Mean Squared Error measures **how far** your predictions are from the real values.

$$\text{MSE}=\frac{1}{n}\sum_{i=0}^{n}(y-\hat{y})^2$$

The smaller the MSE, the **better** your model fits the data.