

▼ Connect to Google Drive

```
# Import drive to connect and interact with Google Drive (so we can import the data)
# Note: This may take a while, but remember to give permission
from google.colab import drive

drive.mount("/content/gdrive")
!pwd # Print working directory

Drive already mounted at /content/gdrive; to attempt to forcibly remount, call drive.mount("/content/gdrive",
/content/gdrive/My Drive/IA avanzada para la ciencia de datos/Machine learning
```

```
# Navigate to the path where the dataset is stored and read the csv file
%cd "/content/gdrive/My Drive/IA avanzada para la ciencia de datos/Machine learning"
!ls # List files located in defined folder
```

```

/ content/gdrive/My Drive/IA avanzada para la ciencia de datos/Machine learning
actividad5_over_under_fitting.ipynb  'log_reg_gd_V06_alumno-1 (1).ipynb'
hypothesis_function.ipynb            'log_reg_multiclasa_alumnos (1).ipynb'
Iris.csv                             'perceptron_and_or_xor-1 (2).ipynb'
'linear_reg_gd1_alumno (1).ipynb'    Presentaciones
loading_dataset.ipynb
```

▼ Load Dataset

1. Descargar el dataset de iris de Kaggle de la cuenta UCI MACHINE LEARNING
<https://www.kaggle.com/datasets/uciml/iris>
2. Guardar el dataset .data o .csv en mi Drive
3. Cargar el dataset como DataFrame usando Pandas en Python
4. Visualizar el dataset de manera tabular

```
import pandas as pd

# Add names to columns if required
columns = ["sepal length", "sepal width", "petal length", "petal width", "class"]
df = pd.read_csv('Iris.csv', names=columns)

print(df)
```

	sepal length	sepal width	petal length	petal width	class
0	5.1	3.5	1.4	0.2	Iris-setosa
1	4.9	3.0	1.4	0.2	Iris-setosa
2	4.7	3.2	1.3	0.2	Iris-setosa
3	4.6	3.1	1.5	0.2	Iris-setosa
4	5.0	3.6	1.4	0.2	Iris-setosa
...
145	6.7	3.0	5.2	2.3	Iris-virginica
146	6.3	2.5	5.0	1.9	Iris-virginica
147	6.5	3.0	5.2	2.0	Iris-virginica
148	6.2	3.4	5.4	2.3	Iris-virginica
149	5.9	3.0	5.1	1.8	Iris-virginica

[150 rows x 5 columns]

▼ Explore Dataset

1. Imprimir nombre de columnas
2. Seleccionar una sola columna
3. Seleccionar multiples columnas

```
# See columns names with 'columns' property
df.columns
```

```
Index(['sepal length', 'sepal width', 'petal length', 'petal width', 'class'], dtype='object')
```

```
# Select a single column based on column name
df['sepal length']
```

```
0      5.1
1      4.9
2      4.7
3      4.6
4      5.0
...
145    6.7
146    6.3
147    6.5
148    6.2
149    5.9
Name: sepal length, Length: 150, dtype: float64
```

```
# Select multiple columns
df[["sepal length", "sepal width"]].head(6)
```

	sepal length	sepal width
0	5.1	3.5
1	4.9	3.0
2	4.7	3.2
3	4.6	3.1
4	5.0	3.6
5	5.4	3.9

▼ Define X and y (features and labels)

1. Definir X(features) y y(labels) del dataset y guardarlos en una variable cada uno
2. Visualizar X y y
3. Crear nuevas columnas basadas en columnas existentes (sepal_proportion & petal proportion)
4. Re-definir mi X con las nuevas columnas (sepal_proportion & petal_proportion)
5. Visualizar nuevas features (X)

```
# Define X(features) & y(labels) from the dataset
df_X = df[["sepal length", "sepal width", "petal length", "petal width"]]
df_y = df[["class"]]
```

```
df_X.head()
```

	sepal length	sepal width	petal length	petal width
0	5.1	3.5	1.4	0.2
1	4.9	3.0	1.4	0.2
2	4.7	3.2	1.3	0.2
3	4.6	3.1	1.5	0.2
4	5.0	3.6	1.4	0.2



```
df_y.head()
```

	class
0	Iris-setosa
1	Iris-setosa
2	Iris-setosa
3	Iris-setosa
4	Iris-setosa

```
# Make new columns with indicators (based on values of other columns) [Adding features]
# Get proportion of length vs width for both sepal and petal
df_X["sepal proportion"] = df_X["sepal width"]/df_X["sepal length"]
df_X["petal proportion"] = df_X["petal width"]/df_X["petal length"]
df_X.head()
```

	sepal length	sepal width	petal length	petal width	sepal proportion	petal proportion
0	5.1	3.5	1.4	0.2	0.686275	0.142857
1	4.9	3.0	1.4	0.2	0.612245	0.142857
2	4.7	3.2	1.3	0.2	0.680851	0.153846
3	4.6	3.1	1.5	0.2	0.673913	0.133333

```
# Define New X(features)
df_X_new = df_X[["sepal length", "sepal width"]]
df_X_new
```

	sepal length	sepal width	
0	5.1	3.5	
1	4.9	3.0	
2	4.7	3.2	
3	4.6	3.1	

df_y

	class	
0	Iris-setosa	
1	Iris-setosa	
2	Iris-setosa	
3	Iris-setosa	
4	Iris-setosa	
...	...	
145	Iris-virginica	
146	Iris-virginica	
147	Iris-virginica	
148	Iris-virginica	
149	Iris-virginica	

Quiz

1. ¿Por qué separamos las features de los labels en el dataset?

Nosotros separamos datos en dos partes: "features" y "labels". Ya que esto simplifica el proceso y nos ayuda a entender mejor nuestro problema.

Primero, al dividirlos, queda claro cuáles son las entradas y qué estamos tratando de predecir. Esto hace que el problema sea más fácil de entender.

Segundo, nos permite aplicar transformaciones específicas a cada conjunto si es necesario. A veces, las características y las etiquetas necesitan un tratamiento diferente, como normalización o codificación, y esta separación facilita este proceso.

Reporte

En esta clase vimos lo que es la IA, que es el ML y el DL, así como los tipos de Machine Learning, vimos que es la data que es una database y un dataset, así mismo aprendimos para que nos sirven los datos en ML, las diferencias entre labels y features.

Para esta actividad tenemos un dataset que obtuvimos de kaggle, el famoso Iris, separamos los labels de los features para tener mejor un entendimiento de los datos.

En este dataset tenemos la clasificacion de flores de 3 tipos, donde estas la "Clase", serian los labels y lo que queremeos predecir con los features que posee cada una.

✓ 0s completed at 7:31 PM

