Guillermo García Hernández

2ºDAW 25/26

Tipos de IA, historia y rol de los datos

Índice

1. Objetivo del documento

El presente informe tiene como objetivo dar respuesta a una serie de cuestiones relacionadas con los fundamentos de la Inteligencia Artificial (IA), sus tipos de aprendizaje, la importancia de los datos, y la aplicación práctica de algoritmos de clasificación sobre datasets reales como *Titanic* e *Iris*.

2. Problemas planteados, razonamientos y soluciones

2.1. Tipos de IA: simbólica, ML clásico y Deep Learning

- Sistema experto médico de los 80 → **IA simbólica**
- Reconocimiento facial en el móvil → **Deep Learning**
- Chatbot basado en reglas → **IA simbólica**
- Netflix recomendando series → **ML clásico**
- AlphaGo → **Deep Learning + Aprendizaje por refuerzo**

2.2. Aprendizaje supervisado, no supervisado y por refuerzo

- Clasificar correos spam/no spam → **Supervisado**
- Agrupar clientes según hábitos → **No supervisado**
- Dron que evita obstáculos → **Refuerzo**
- Predecir precios de casas → **Supervisado**

2.3. Enfoque más usado en aplicaciones web

El enfoque más empleado actualmente en aplicaciones web es el **aprendizaje supervisado**, ya que permite entrenar modelos con datos etiquetados y obtener predicciones fiables en problemas prácticos como recomendadores, filtros de spam, personalización de contenido o detección de fraude.

2.4. Dataset Titanic: problemas y limpieza básica

- **Problemas detectados:**
    - Valores nulos en columnas (*Age, Cabin, Embarked*).
    - Variables categóricas sin transformar.
    - Desbalanceo en las clases (más fallecidos que supervivientes).
- **Riesgo de entrenar con datos incompletos:**
    - Se reduce la precisión del modelo.
    - Aumenta el sesgo.
- **Ejemplo de limpieza con Pandas:**

```
import pandas as pd

df = pd.read_csv("titanic.csv")
df["Age"].fillna(df["Age"].median(), inplace=True)
df.drop(columns=["Cabin"], inplace=True)
df["Embarked"].fillna(df["Embarked"].mode()[0], inplace=True)
```

2.5. “Los datos son el nuevo petróleo”

La expresión se refiere a que los datos son el recurso fundamental que impulsa el desarrollo de la IA. Igual que el petróleo fue clave en la revolución industrial, los datos lo son en la revolución digital.

2.6. Riesgos de sesgo en los datos

- **Contrataciones:** riesgo de discriminar por género o raza.
- **Justicia:** modelos entrenados con sentencias previas pueden reforzar prejuicios.
- **Sanidad:** diagnósticos menos fiables en minorías por falta de datos representativos.

2.7. Sistema de recomendación de restaurantes

- **Datos necesarios:** ubicación, tipo de comida, precio, reseñas, valoraciones.
- **Recopilación:** apps móviles, formularios, APIs externas (Google Maps, Yelp).
- **Problemas éticos:** privacidad de usuarios, sesgo hacia grandes cadenas, manipulación de reseñas.

2.8. Iris dataset

Cargar el dataset Iris con scikit-learn

Para cargar el dataset Iris utilizamos la librería scikit-learn:

```
from sklearn.datasets import load_iris

from sklearn import datasets

iris = datasets.load_iris()

X, y = iris.data, iris.target
```

El dataset contiene un total de **150 muestras** distribuidas en **3 clases**: *Setosa*, *Versicolor* y *Virginica*.

Cada muestra está descrita por **4 características (features)**:

1. Longitud del sépalo.
2. Anchura del sépalo.
3. Longitud del pétalo.
4. Anchura del pétalo.

Exploración inicial

Mostrar las primeras 5 filas.

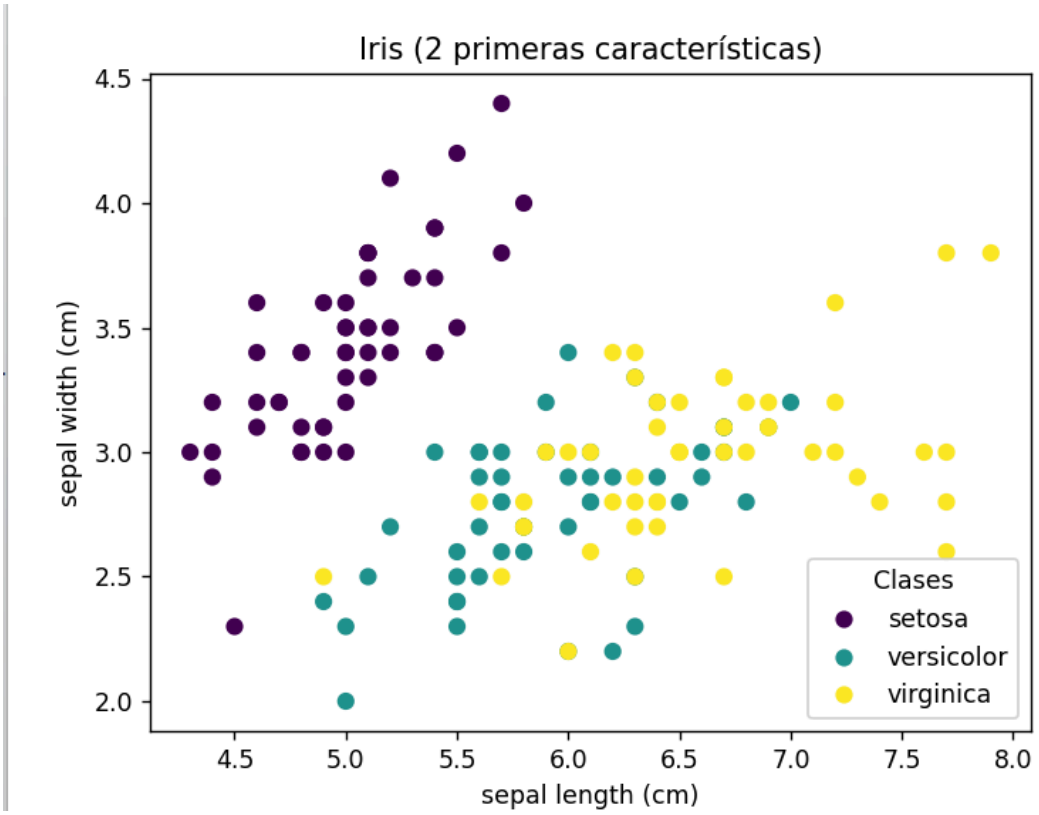
df.head()

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

Graficar un *pairplot* (usando seaborn.pairplot) coloreado por especie.

```
import seaborn as sns

sns.pairplot(df, hue="species")
```



División de datos

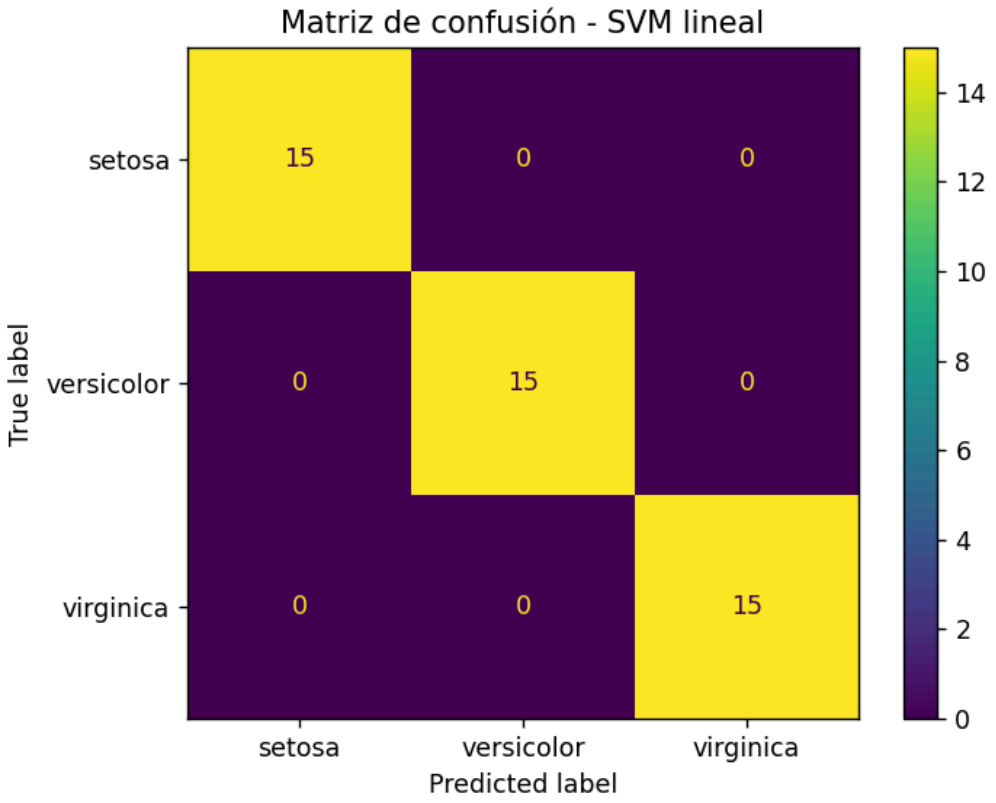
Los datos se separaron en entrenamiento (70%) y prueba (30%):

```
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, stratify=y, random_state=42
)
```

Entrenamiento con KNN y SVM

- **Logistic Regression (LogReg): Accuracy = 0.933**
- **KNN (k=5): Accuracy = 0.978**
- **SVM (lineal): Accuracy = 1.000**



Comparación de modelos

El modelo SVM lineal obtuvo 100% de precisión, KNN 97.8% y Logistic Regression 93.3%. En este dataset sencillo, SVM es el más robusto y no cometió errores, mientras que en problemas más complejos KNN tiende a confundirse entre versicolor y virginica.