



ESCUELA POLITÉCNICA NACIONAL

ESCUELA DE FORMACIÓN DE TECNÓLOGOS



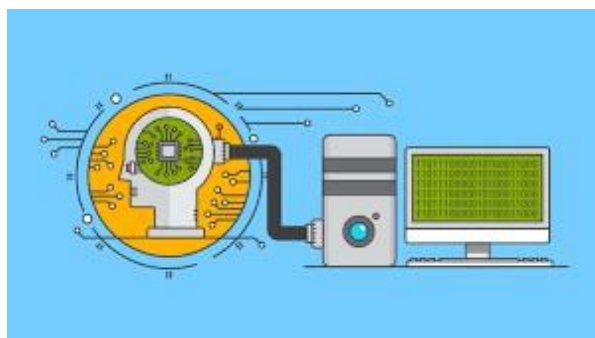
FUNDAMENTOS INTELIGENCIA ARTIFICIAL

ASIGNATURA: Fundamentos de Inteligencia Artificial
PROFESOR: Ing. Yadira Franco R
PERÍODO ACADÉMICO: 2025-A

TAREA SEMANA 3

TÍTULO:

Introducción a la Inteligencia Artificial IA



Nombre: Edwin Sarango

Objetivo de la clase: Introducir los conceptos de **aprendizaje supervisado**, enseñar a aplicar modelos de clasificación en Python y evaluar su rendimiento.

Tarea: Crear un conjunto de datos, entrenar y evaluar modelos de clasificación, y comparar los resultados obtenidos.

1. Que es algoritmos de aprendizaje supervisado y modelos supervisados, diferencias y uso

El *Machine Learning (ML)* es una rama de la inteligencia artificial que permite a las computadoras aprender de datos, identificar patrones y realizar predicciones sin ser programadas explícitamente. Los algoritmos de ML se entrenan con grandes volúmenes de datos históricos y se dividen principalmente en dos tipos: **aprendizaje supervisado** y **aprendizaje no supervisado**, según el tipo de datos utilizados durante el entrenamiento.

El aprendizaje supervisado implica el uso de un conjunto de datos de entrenamiento donde cada entrada está emparejada con una salida deseada (es decir, los datos están *etiquetados*). El algoritmo aprende a asociar entradas con salidas correctas y luego puede hacer predicciones sobre nuevos datos similares.

Características principales:

- Utiliza datos de entrada con sus respectivas salidas conocidas.
- El objetivo es predecir resultados precisos en función de los datos.
- Ideal para problemas de clasificación y regresión.

Técnicas comunes de aprendizaje supervisado:

Regresión Logística

Predice resultados categóricos (clasificación).

Útil para problemas de clasificación binaria (sí/no, aprobado/desaprobado) o multiclase (gato/perro/conejo).

Ejemplo: predecir si un estudiante aprobará una unidad basándose en su actividad en una plataforma educativa.

Regresión Lineal

Predice un valor numérico en una escala continua.

Ejemplo: estimar el precio de una casa en función de características como ubicación, tamaño o antigüedad.

Árbol de Decisiones

Usa una estructura jerárquica de condiciones (tipo “si... entonces...”) para predecir resultados.

Ejemplo: predecir la pérdida de un cliente según su actividad en una aplicación.

Redes Neuronales

Imitan el funcionamiento del cerebro humano con capas de procesamiento.

Realizan transformaciones matemáticas sobre los datos para obtener un resultado.

Ejemplo: reconocer números escritos a mano a partir de imágenes.

2. Cómo funciona aprendizaje supervisado en Python, utilizando el conjunto

El aprendizaje supervisado es una tarea de aprendizaje automático en la que se entrena un algoritmo para encontrar patrones utilizando un conjunto de datos. El algoritmo de aprendizaje supervisado utiliza este entrenamiento para realizar inferencias de entrada y salida sobre conjuntos de datos futuros. Es una categoría de aprendizaje automático que utiliza conjuntos de datos etiquetados para entrenar algoritmos que predicen resultados y reconocen patrones. A diferencia del aprendizaje no supervisado, los algoritmos de aprendizaje supervisado reciben entrenamiento etiquetado para comprender la relación entre la entrada y la salida.

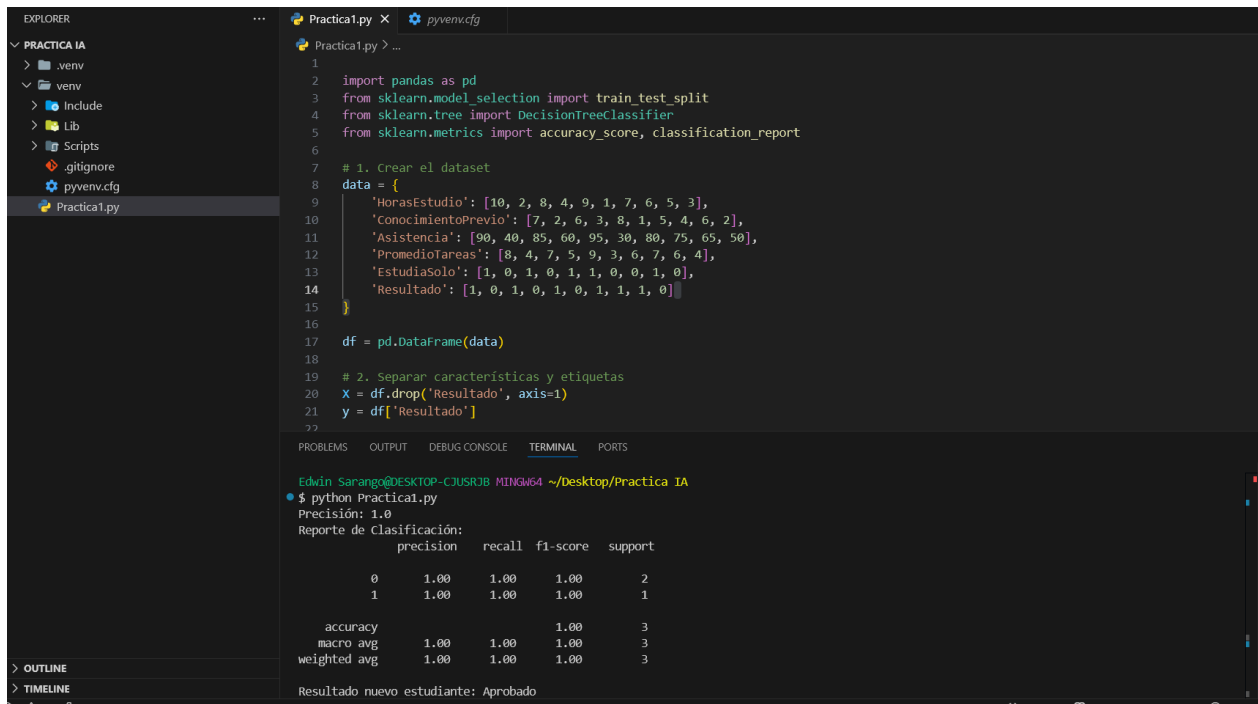
PRACTICA #1

identifique si un estudiante **aprobó o no aprobó** en función de características como el número de horas de estudio, nivel de conocimiento previo, entre otras, puedes seguir esta estructura.

Características (Entradas) para el Dataset:

1. **Horas de Estudio:** Número de horas que el estudiante dedicó al estudio.
2. **Nivel de Conocimiento Previo:** Un valor de 1 a 10, donde 1 es bajo conocimiento y 10 es un nivel alto de conocimiento previo.
3. **Asistencia a Clases:** Porcentaje de clases asistidas, entre 0 y 100.
4. **Promedio de Tareas:** Promedio de calificaciones de las tareas o ejercicios realizados.
5. **Tipo de Estudiante:** Si es un estudiante que estudia solo (1) o en grupo (0).

Salida (Etiqueta):



```
1
2 import pandas as pd
3 from sklearn.model_selection import train_test_split
4 from sklearn.tree import DecisionTreeClassifier
5 from sklearn.metrics import accuracy_score, classification_report
6
7 # 1. Crear el dataset
8 data = {
9     'HorasEstudio': [10, 2, 8, 4, 9, 1, 7, 6, 5, 3],
10    'ConocimientoPrevio': [7, 2, 6, 3, 8, 1, 5, 4, 6, 2],
11    'Asistencia': [90, 40, 85, 60, 95, 30, 80, 75, 65, 50],
12    'PromedioTareas': [8, 4, 7, 5, 9, 3, 6, 7, 6, 4],
13    'Estudiasolo': [1, 0, 1, 0, 1, 1, 0, 0, 1, 0],
14    'Resultado': [1, 0, 1, 0, 1, 0, 1, 1, 1, 0]
15 }
16
17 df = pd.DataFrame(data)
18
19 # 2. Separar características y etiquetas
20 X = df.drop('Resultado', axis=1)
21 y = df['Resultado']
22
```

Edwin Sarango@DESKTOP-CJUSRJB MINGW64 ~/Desktop/Practica IA

```
$ python Practica1.py
Precisión: 1.00
Reporte de Clasificación:
              precision    recall  f1-score   support
0               1.00        1.00        1.00         2
1               1.00        1.00        1.00         1

 accuracy          1.00          1.00          1.00          3
 macro avg          1.00          1.00          1.00          3
weighted avg          1.00          1.00          1.00          3

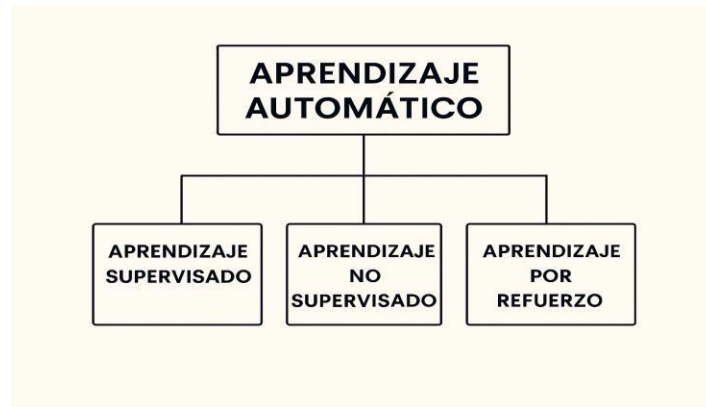
Resultado nuevo estudiante: Aprobado
```

- **Resultado:** "Aprobado" o "No Aprobado" (0 = No Aprobado, 1 = Aprobado).

Entrenamiento de un modelo: Este dataset puede ser utilizado para entrenar un modelo de clasificación supervisado (por ejemplo, Regresión Logística, Árboles de Decisión, etc.) para predecir

si un estudiante aprobará o no en función de las características dadas.

Próxima clase TEST SERÁ NOTA DE LAS TAREAS



Resumen de Clasificación:

Tipo de Aprendizaje	Definición	Ejemplo de Tareas	Algoritmos Comunes
Supervisado	Datos etiquetados para predicción de resultados.	Clasificación, Regresión	Regresión Logística, SVM, Árboles de Decisión, KNN, Redes Neuronales
No Supervisado	Datos no etiquetados para encontrar patrones y estructuras.	Clustering, Reducción de Dimensionalidad	K-means, PCA, Agrupamiento Jerárquico
Refuerzo	Aprende mediante recompensas y penalizaciones en decisiones secuenciales.	Juegos, Control de Robots, Optimización de Estrategias	Q-learning, DQN, Algoritmos de Política

Lo que se necesita del Dataset para entrenar y evaluar estos modelos:

1. **Datos Etiquetados:** Necesitamos que el dataset esté correctamente etiquetado, es decir, que tengamos las columnas de características (entradas) y la etiqueta de salida (si el estudiante aprobó o no).
2. **División en Conjuntos de Entrenamiento y Prueba:** Para evaluar el rendimiento de los modelos, dividimos el dataset en **dos partes**: una para entrenar los modelos (usualmente 70-80% del total) y otra para probarlos (usualmente 20-30% del total).
3. **Datos Limpios:** Es importante que el dataset no tenga valores faltantes o datos irrelevantes que puedan afectar la precisión de los modelos.
4. **Escalado de Datos (si es necesario):** Algunos modelos, como KNN y SVM, pueden beneficiarse de la **normalización** o **escalado** de los datos para asegurar que todas las características tengan la misma escala.

Subir el archivo y código al git hub **INDIVIDUAL LA TAREA**