



# ESCUELA POLITÉCNICA NACIONAL

## ESCUELA DE FORMACIÓN DE TECNÓLOGOS



### FUNDAMENTOS INTELIGENCIA ARTIFICIAL

ASIGNATURA:

Fundamentos de Inteligencia Artificial

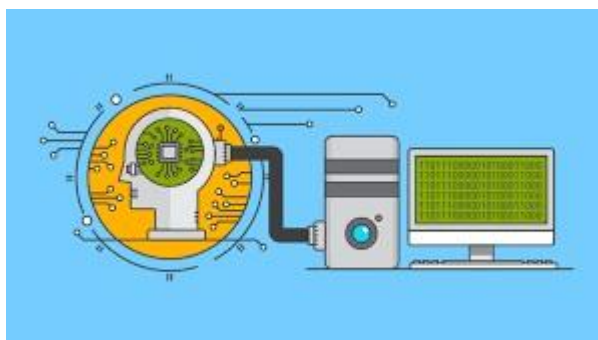
PROFESOR:

Ing. Yadira Franco R PERÍODO ACADÉMICO: 2025-A

### TAREA SEMANA 3

TÍTULO:

**Introducción a la Inteligencia Artificial IA**



Nombre: Joshua Morocho

**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**

**Algoritmos de aprendizaje supervisado.**

Es un tipo de aprendizaje automático en el que el algoritmo aprende a partir de un conjunto de datos etiquetado (es decir, ya sabemos cuál es la respuesta correcta para cada ejemplo). El objetivo es que, una vez entrenado, el modelo pueda predecir resultados para nuevos datos.

**Modelos Supervisados.**

Son las técnicas o métodos utilizados para entrenar modelos con datos etiquetados como lo son:

**Regresión lineal:** para predecir valores numéricos (como precios).

**Árboles de decisión:** para clasificar o predecir.

**K-NN** (K-vecinos más cercanos).

**Máquinas de soporte vectorial** (SVM).

**Redes neuronales** supervisadas.

**Usos:**

**Clasificación:** predecir categorías (¿spam o no spam?).

**Regresión:** predecir valores numéricos (¿cuánto costará?).

**Reconocimiento de voz o imagen** (cuando se tiene un conjunto de entrenamiento etiquetado).

**Diagnóstico médico** (predecir enfermedades a partir de síntomas).

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

**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.

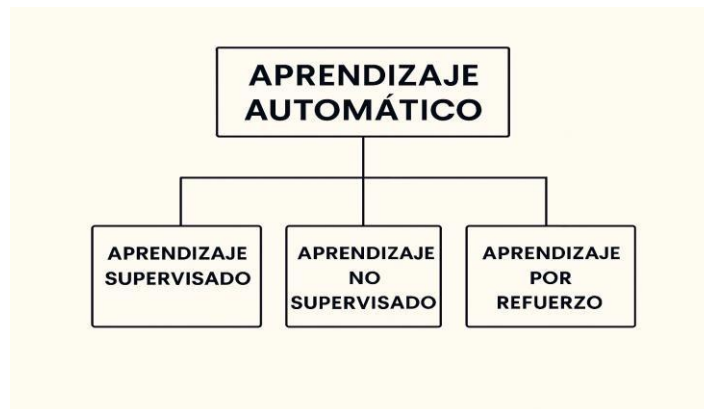
4. **Promedio de Tareas:** Promedio de calificaciones de las tareas o ejercicios realizados.
5. **Tipo de Estudiante:** Si es un estudiante que estudia solo (1) o en grupo (0).

**Salida (Etiqueta):**

- **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.

Se hace uso de la librería sklearn para el entrenamiento y generación de las etiquetas al modelo:

```
import pandas as pd
import numpy as np
from sklearn.model_selection import train_test_split
from sklearn.tree import DecisionTreeClassifier
from sklearn.preprocessing import StandardScaler
from sklearn.metrics import classification_report, accuracy_score, confusion_matrix
```

Se genera el dataset con datos ficticios definiendo sus valores

```
data = { #Datasets fictiops
    'Horas_Estudio': [2, 5, 1, 8, 6, 7, 3, 4, 9, 10],
    'Conocimiento_Previo': [3, 7, 2, 9, 6, 8, 4, 5, 9, 10],
    'Asistencia': [60, 90, 50, 100, 85, 95, 65, 70, 100, 100],

    'Promedio_Tareas': [60, 80, 50, 95, 85, 90, 70, 75, 100, 98],
    'Tipo_Estudiente': [1, 0, 1, 0, 1, 1, 1, 0, 1, 0],
    'Resultado': [0, 1, 0, 1, 1, 0, 0, 1, 1, 1] #1 = Aprobado, 0 = No Aprobado
}
```

Se hace la operación y generación de etiquetas de entrada para el entrenamiento y prueba

```
#Se convierte a DataFrame
df = pd.DataFrame(data)
X = df.drop('Resultado', axis=1)
y = df['Resultado']

#Escalacion de caracteristicas
escala = StandardScaler()
X_escalado = escala.fit_transform(X)

#División en conjuntos de entrenamiento y prueba
X_entrenado, X_prueba, y_entrenado, y_prueba = train_test_split(X_escalado, y, test_size=0.3, random_state=42)
```

Se genera el tipo de modelo y sus predicciones

```
#modelo arbol de decisión
model = DecisionTreeClassifier(random_state=42)
model.fit(X_entrenado, y_entrenado)

#predicciones
y_prediccion = model.predict(X_prueba)
```