## 1. Librerías necesarias:

# Librerías

from sklearn import metrics, model\_selection, preprocessing

from sklearn.svm import SVC, SVR

from sklearn.model\_selection import train\_test\_split

from sklearn.metrics import mean\_absolute\_error, mean\_squared\_error, r2\_score, accuracy\_score, confusion\_matrix, classification\_report

from sklearn.preprocessing import MinMaxScaler, StandardScaler

import numpy as np

import pandas as pd

import matplotlib.pyplot as plt

import seaborn as sns

## 3. Asignación de variables dependientes e independientes:

# Para regresión

X = df.drop(columns=['Salary']) # X: Variables independientes

y = df['Salary'] # y: Variable dependiente (objetivo)

# Para clasificación

# X = df.drop(columns=['output']) # Variables independientes

# y = df['output'] # Variable dependiente (objetivo)

## 

## **4. Reshape de la variable dependiente** (solo en caso de ser necesario):

y = y.values.reshape(-1, 1) # Si la variable y es unidimensional, se puede hacer reshape

## 5. Escalado de datos:

scaler = MinMaxScaler()

X = scaler.fit\_transform(X) # Escala X (variables independientes)

# Si es necesario escalar también Y (aunque generalmente no es recomendable en regresión):

# y = scaler.fit\_transform(y)

## 6. División de datos en conjunto de entrenamiento y test:

X\_train, X\_test, y\_train, y\_test = train\_test\_split(X, y, test\_size=0.1, random\_state=42)

## 

## 7. SVM para regresión (SVR - Support Vector Regression)

## Creación del modelo y entrenamiento:

# Crear el modelo de Support Vector Regression

svr\_model = SVR(kernel='linear') # kernel puede ser 'linear', 'rbf', 'poly', etc.

# Entrenar el modelo

svr\_model.fit(X\_train, y\_train)

## Predicciones y evaluación:

# Realizar predicciones

y\_pred\_svr = svr\_model.predict(X\_test)

# Cálculo de las métricas de regresión

mae\_svr = mean\_absolute\_error(y\_test, y\_pred\_svr) # Error Absoluto Medio

mse\_svr = mean\_squared\_error(y\_test, y\_pred\_svr) # Error Cuadrático Medio

rmse\_svr = np.sqrt(mse\_svr) # Raíz del MSE

r2\_svr = r2\_score(y\_test, y\_pred\_svr) # Coeficiente de Determinación

# Mostrar los resultados

print(f'Error Absoluto Medio (MAE): {mae\_svr}')

print(f'Error Cuadrático Medio (MSE): {mse\_svr}')

print(f'Raíz del Error Cuadrático Medio (RMSE): {rmse\_svr}')

print(f'Coeficiente de Determinación (R²): {r2\_svr}')

## 8. SVM para clasificación (SVC - Support Vector Classifier)

## Creación del modelo y entrenamiento:

# Crear el modelo de Support Vector Classifier

svc\_model = SVC(kernel='linear') # kernel puede ser 'linear', 'rbf', 'poly', etc.

# Entrenar el modelo

svc\_model.fit(X\_train, y\_train)

Predicciones y evaluación:

# Realizar predicciones

y\_pred\_svc = svc\_model.predict(X\_test)

# Cálculo de las métricas de clasificación

accuracy\_svc = accuracy\_score(y\_test, y\_pred\_svc)

print(f'Accuracy: {accuracy\_svc:.2f}')

# Matriz de confusión

cm\_svc = confusion\_matrix(y\_test, y\_pred\_svc)

print('Confusion Matrix:')

print(cm\_svc)

# Reporte de clasificación

print('Classification Report:')

print(classification\_report(y\_test, y\_pred\_svc))