



**Módulo 2 Implementación de una técnica de aprendizaje máquina sin el
uso de un framework**

Alumno:

Julián Cisneros Cortés A01747363

Grupo 502

TC3006C.101 Inteligencia artificial avanzada para la ciencia de datos I

Profesores:

Jorge Adolfo Ramírez Uresti

Campus Estado de México

09 de septiembre

Esta primera parte es para leer el excel, y separar entre las variables dependiente y, y la variable independiente x

```
6 # Leer y barajar datos
7 df = pd.read_csv('Salary_dataset.csv', index_col=0).sample(frac=1).reset_index(drop=True)
8
9 # Dividir entre mi variable destino, la variables dependiente y, y la variable independiente x
10 x, y = df['YearsExperience'], df['Salary']
11
```

Después dividimos el dataset entre entrenamiento, prueba y validación

```
# Dividir en conjuntos de entrenamiento, validación y prueba
# 70% entrenamiento, 15% validación y 15% testing
training_size = int(len(df) * 0.7)
validation_size = int(len(df) * 0.15)

X_train, X_valid, X_test = x.iloc[:training_size], x.iloc[training_size:training_size + validation_size], x.iloc[training_size + validation_size:]
y_train, y_valid, y_test = y.iloc[:training_size], y.iloc[training_size:training_size + validation_size], y.iloc[training_size + validation_size:]
```

Realizar el cálculo de la regresión lineal

```
# Calcular parámetros de la regresión lineal
# Sacamos la media de nuestros datos de entrenamiento
x_mean, y_mean = X_train.mean(), y_train.mean()
# Calculamos la pendiente, indicando cómo cambia el salario de acuerdo a la experiencia
m1 = np.sum((X_train - x_mean) * (y_train - y_mean)) / np.sum((X_train - x_mean) ** 2)
# Calculamos la intersección, el valor de y cuando x = 0
m0 = y_mean - m1 * x_mean
```

Realizamos predicciones, y creamos un umbral para poder hacer clasificación, y poder tomar mejores métricas como la matriz de confusión

```
# Predicciones
y_pred_valid = m0 + m1 * X_valid
y_pred_test = m0 + m1 * X_test

# Definir un umbral para convertir a clasificación binaria, y hacer una matriz de confusión
threshold = 100000
y_test_class = np.where(y_test >= threshold, 1, 0)
y_pred_class = np.where(y_pred_test >= threshold, 1, 0)
```

Calculamos diferentes métricas, para medir el desempeño

```
# Calcular métricas, precisión, recall, f1 y la matriz de confusión
precision = precision_score(y_test_class, y_pred_class)
recall = recall_score(y_test_class, y_pred_class)
f1 = f1_score(y_test_class, y_pred_class)
conf_matrix = confusion_matrix(y_test_class, y_pred_class)
```

Mostramos la gráfica con la regresión lineal, y mapeamos los valores de entrenamiento, validación y testing

Se puede ver que sí hay una relación entre la experiencia y el salario, entonces este modelo es bueno y rápido, para este problema