

Actividad 3 - Métodos de aprendizaje supervisado

Realizado por:

Cristhian Camilo Lara Londoño - Código Banner: 100102964

Presentado a:

Jorge Castañeda



Corporación universitaria Iberoamericana

Facultad de ingeniería

Ingeniería de software virtual

Inteligencia Artificial

13 de abril de 2025

De acuerdo con lo solicitado, en el siguiente link puede encontrar el video correspondiente a la Actividad 3 - Métodos de aprendizaje supervisado, en la cual se aplican técnicas de aprendizaje automático supervisado para predecir el tiempo de viaje entre estaciones dentro de un sistema de transporte masivo simulado. Así como el link del repositorio.

Video: https://www.youtube.com/watch?v=1gKfqUY0Y_4

Repositorio: <https://github.com/Cristhianlara7/Actividad-3---M-todos-de-aprendizaje-supervisado.git>

Evidencias:

1. Importación de librerías

```
1 import pandas as pd
2 import numpy as np
3 from sklearn.model_selection import train_test_split
4 from sklearn.preprocessing import LabelEncoder
5 from sklearn.ensemble import RandomForestRegressor
6 from sklearn.metrics import mean_squared_error
7
```

2. Crear un dataset simulado

```
# Crear dataset simulado
np.random.seed(0)
n = 20
data = pd.DataFrame({
    'origen': np.random.choice(['Est. A', 'Est. B', 'Est. C'], size=n),
    'destino': np.random.choice(['Est. D', 'Est. E', 'Est. F'], size=n),
    'hora': np.random.randint(5, 23, size=n),
    'dia_semana': np.random.choice(['Lunes', 'Martes', 'Miércoles', 'Jueves', 'Viernes'], size=n),
    'clima': np.random.choice(['Soleado', 'Lluvia', 'Nublado'], size=n),
    'tiempo_viaje': np.random.normal(loc=15, scale=3, size=n).round(2)
})
```

3. Preprocesamiento:

```
# Preprocesamiento
le_origen = LabelEncoder()
le_destino = LabelEncoder()
le_dia = LabelEncoder()
le_clima = LabelEncoder()

data['origen'] = le_origen.fit_transform(data['origen'])
data['destino'] = le_destino.fit_transform(data['destino'])
data['dia_semana'] = le_dia.fit_transform(data['dia_semana'])
data['clima'] = le_clima.fit_transform(data['clima'])

# Separar variables predictoras y objetivo
X = data.drop('tiempo_viaje', axis=1)
y = data['tiempo_viaje']
```

4. Entrenamiento y prueba.

```
# Entrenamiento y prueba
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)

# Modelo
modelo = RandomForestRegressor(n_estimators=100, random_state=42)
modelo.fit(X_train, y_train)

# Evaluación
y_pred = modelo.predict(X_test)
error = mean_squared_error(y_test, y_pred) ** 0.5
print(f"Error promedio (RMSE): {error:.2f} minutos")
```

5. Ejecución:

```
C:\Users\CRSLA\OneDrive\Documentos\Backup Angie\Documentos\Disco D\Semestre febrero - Junio\Inteligencia artificial>python modelo_transporte.py
Error promedio (RMSE): 3.03 minutos

C:\Users\CRSLA\OneDrive\Documentos\Backup Angie\Documentos\Disco D\Semestre febrero - Junio\Inteligencia artificial>
```

- Este resultado dependerá de los valores ingresados.