



Tecnológico de Monterrey

**Instituto Tecnológico y de Estudios Superiores de Monterrey,
Campus Monterrey**

Escuela de ingeniería y ciencias

**Inteligencia Artificial Avanzada para la Ciencia de Datos I
(Grupo 101)**

**Momento de Retroalimentación: Módulo 2 Implementación de una
técnica de aprendizaje máquina sin el uso de un framework
(Portafolio Implementación)**

Alumnos:

Amy Murakami Tsutsumi

A01750185

Profesor:

Iván Mauricio Amaya Contreras

Fecha:

18/09/2022

Dataset

Para este portafolio de implementación se utilizará el dataset de "Weather in Szeged 2006 - 2016" que se obtuvo de la siguiente liga:

<https://www.kaggle.com/datasets/budincsevit/szeged-weather>

Este dataset contiene información con atributos que se utilizarán para determinar y predecir si existe una relación entre temperatura y humedad. Es importante aclarar que se utilizarán únicamente los primeros 100 datos del dataset. Además, los únicos atributos que se utilizarán son:

1. Temperatura: en grados centígrados
2. Humedad: dato que indica la humedad con un valor entre 0 y 1

Librería

Para la implementación del modelo de regresión lineal no se utilizará ninguna librería. Sin embargo, será necesario el uso de librerías para poder graficar y para realizar el subset de entrenamiento. Por lo tanto, las librerías que se utilizarán para estos procesos son:

- pandas : para la creación y operaciones de dataframes.
- matplotlib.pyplot: para la generación de gráficos.
- numpy: Para la creación de vectores y matrices
- sklearn.model_selection - train_test_split : para la división de los datos en subconjuntos de entrenamiento y prueba.

Modelo

Para los modelos se dividió el dataset: el 80% lo conforman los datos de entrenamiento y el 20% son los datos de prueba.

▼ Modularización del data-set

Ahora se realizará la división de datos en subsets de entrenamiento y prueba.

```
✓ [163] #Modularización del data-set
0 s x_train, x_test, y_train, y_test = train_test_split(temp, humidity, test_size = 0.20, train_size = 0.80, random_state= 2)
```

Métrica de desempeño (valor logrado sobre el subset de prueba)

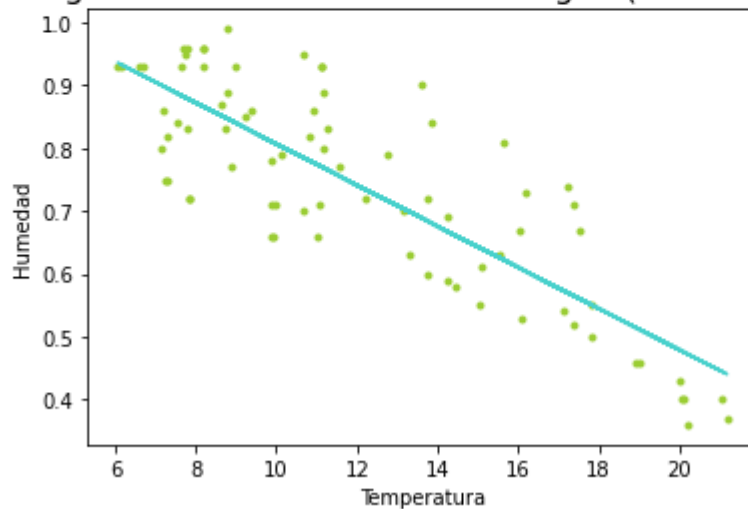
Se realizaron 5 pruebas con distintos valores para poder encontrar el mejor modelo.

✓
7s

```
predic = []  
pred1 = reglin(x_train, y_train, 0.0005, 100000)  
met1 = metricaDesemp(pred1)  
predic.append(met1)  
print("Métrica de desempeño de la prueba 1: ", met1)
```

➡ Theta inicial: [1, 1]
Theta final: [1.134930801501417, -0.03278603245817401]

Regresión lineal del clima en Szeged (2006-2016)



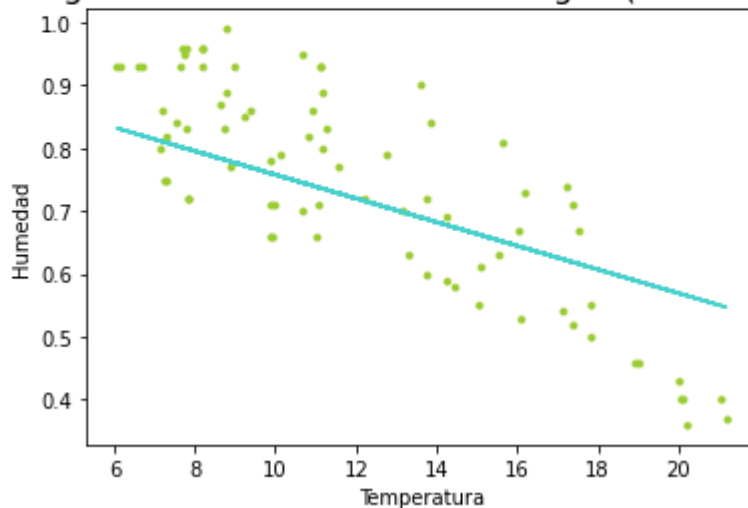
Métrica de desempeño de la prueba 1: 0.6936664029435702

✓
7 s

```
▶ pred2 = regLin(x_train, y_train, 0.00001, 100000)
met2 = metricaDesemp(pred2)
predic.append(met2)
print("Métrica de desempeño de la prueba 2: ", met2)
```

☞ Theta inicial: [1, 1]
Theta final: [0.9464259897430221, -0.018853381066785444]

Regresión lineal del clima en Szeged (2006-2016)



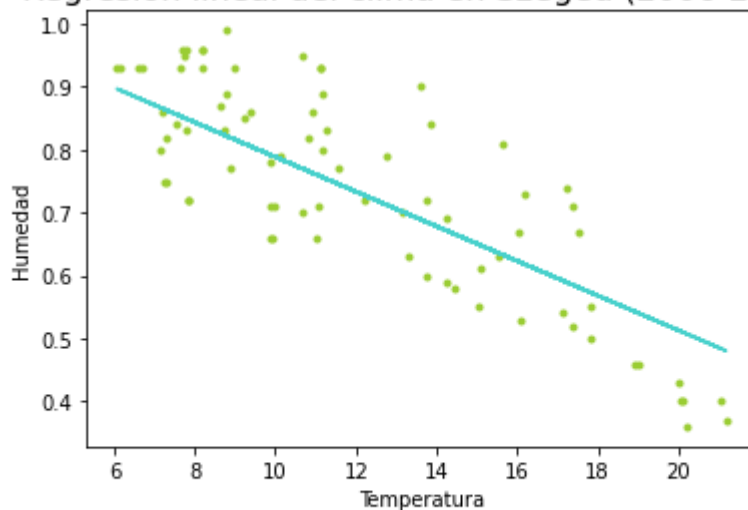
Métrica de desempeño de la prueba 2: 0.6936664029435704

✓
0 s

```
▶ pred3 = regLin(x_train, y_train, 0.001, 10000)
met3 = metricaDesemp(pred3)
predic.append(met3)
print("Métrica de desempeño de la prueba 3: ", met3)
```

☞ Theta inicial: [1, 1]
Theta final: [1.0642278476712483, -0.0275602793388332]

Regresión lineal del clima en Szeged (2006-2016)



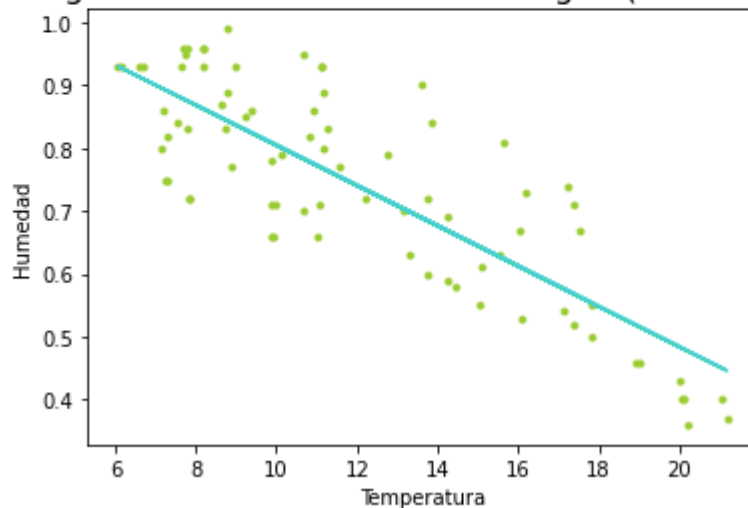
Métrica de desempeño de la prueba 3: 0.6936664029435707

✓
0 s

```
▶ pred4 = reglin(x_train, y_train, 0.005, 5500)
  met4 = metricaDesemp(pred4)
  predic.append(met4)
  print("Métrica de desempeño de la prueba 4: ", met4)
```

↳ Theta inicial: [1, 1]
Theta final: [1.1250707993649236, -0.03205726606115396]

Regresión lineal del clima en Szeged (2006-2016)



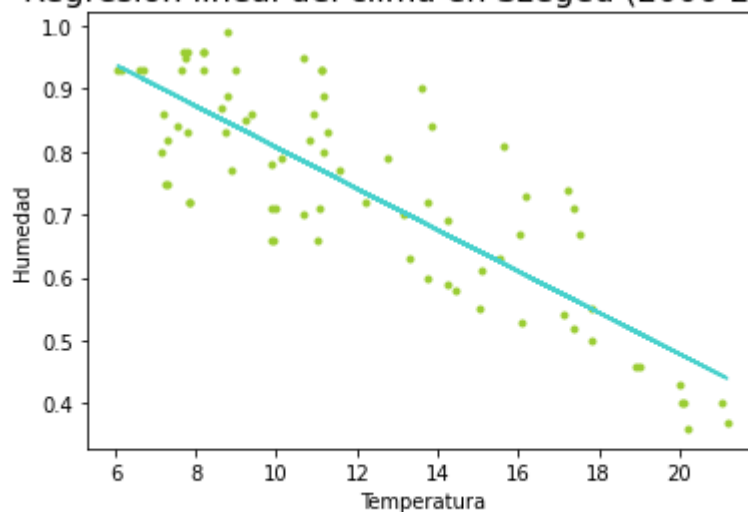
Métrica de desempeño de la prueba 4: 0.6936664029435705

✓
7 s

```
▶ pred5 = reglin(x_train, y_train, 0.0015, 100000)
  met5 = metricaDesemp(pred5)
  predic.append(met5)
  print("Métrica de desempeño de la prueba 5: ", met5)
```

↳ Theta inicial: [1, 1]
Theta final: [1.1358826780390563, -0.03285638696977962]

Regresión lineal del clima en Szeged (2006-2016)



Métrica de desempeño de la prueba 5: 0.6936664029435705

Según las métricas de desempeño obtenidas, la prueba 3 fue la que tuvo mejor desempeño. Por lo tanto se utilizará un valor alpha de 0.001 y 10000 iteraciones.

Predicciones de prueba (entradas, valor esperado, valor obtenido)

Más adelante se realizó la predicción con los valores previamente obtenidos de alpha y las iteraciones.

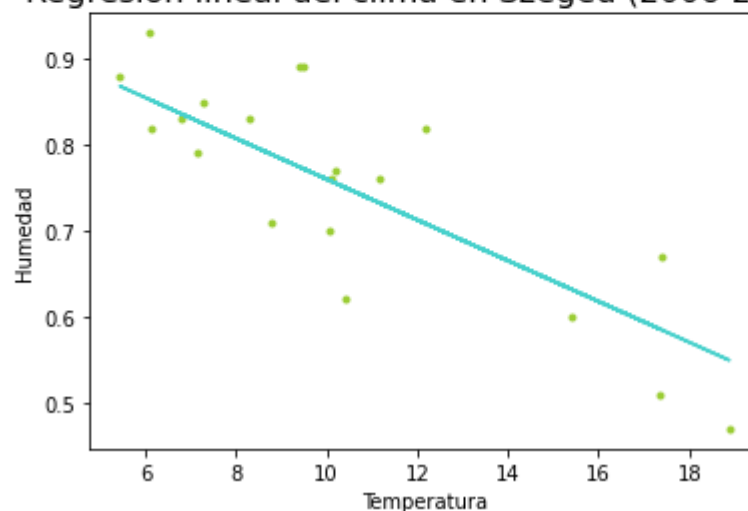
▼ Predicción

```
✓ [206] predF = regLin(x_test, y_test, 0.001, 10000)  
0s
```

Theta inicial: [1, 1]

Theta final: [0.9970759382989637, -0.02369674445443631]

Regresión lineal del clima en Szeged (2006-2016)



Después de realizar las predicciones se generó una tabla que indica las entradas, el valor esperado y el valor obtenido de cada predicción. De esta manera se puede visualizar qué tanto varía el valor real esperado con el valor que se obtuvo en la predicción.

```
✓ [207] pd.set_option('max_columns', None)  
0s [208] dfEntPred = pd.DataFrame()  
[209] dfEntPred["Entrada"] = x_test  
[210] dfEntPred["Valor Real Esperado"] = y_test  
[211] dfEntPred["Predicción"] = predF  
[212] dfEntPred
```

		Entrada	Valor Real Esperado		Predicción
0 s	▶				
📄	93	8.794444		0.71	0.788675
	30	7.261111		0.85	0.825010
	56	12.166667		0.82	0.708766
	24	10.422222		0.62	0.750103
	16	15.388889		0.60	0.632411
	23	10.200000		0.77	0.755369
	2	9.377778		0.89	0.774852
	27	7.155556		0.79	0.827511
	28	6.111111		0.82	0.852260
	13	17.333333		0.51	0.586334
	98	5.438889		0.88	0.868190
	91	10.050000		0.70	0.758923
	88	10.116667		0.76	0.757343
	14	18.877778		0.47	0.549737
	0	9.472222		0.89	0.772614
	21	11.183333		0.76	0.732067

Nombre del archivo a revisar

MomRetroM2.py

MomentoRetroM2SinFramework.pdf