

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

Oskar Arturo Gamboa Reyes

```
In [1]: import matplotlib.pyplot as plt
import pandas as pd
import numpy as np
colnames=['Celsius', 'Valks']
df = pd.read_csv('Valhalla23.csv', names=colnames, skiprows=1)
```

```
In [2]: #Separar datos de prueba y entrenamiento

test_size = round(0.7*len(df))

df_train = df[0:test_size]
df_test = df[test_size:len(df)]

print(len(df))
print(len(df_train))
print(len(df_test))

100
70
30
```

```
In [3]: # Crear Lista con Los hiper-parámetros iniciales (thetas)
theta0 = 1
theta1 = 1
# Cargar el valor del Learning rate (alpha)
# Para poder determinar alpha tuve que hacer prueba y error, hasta que me dio un va
alpha = 1e-4
# Crear función Lambda para la función de hipótesis
hyp = lambda m,b,x: m*x + b

# Calcular el total de muestras a partir de los datos (n)
n = len(df_train)

for y in range(100000):
    # Calcular delta
    delta = hyp(theta1, theta0, df_train['Celsius']) - df_train['Valks']

    # Calcular sumatorias
    sum0 = np.sum(delta)
    sum1 = np.sum(delta * df_train['Celsius'])

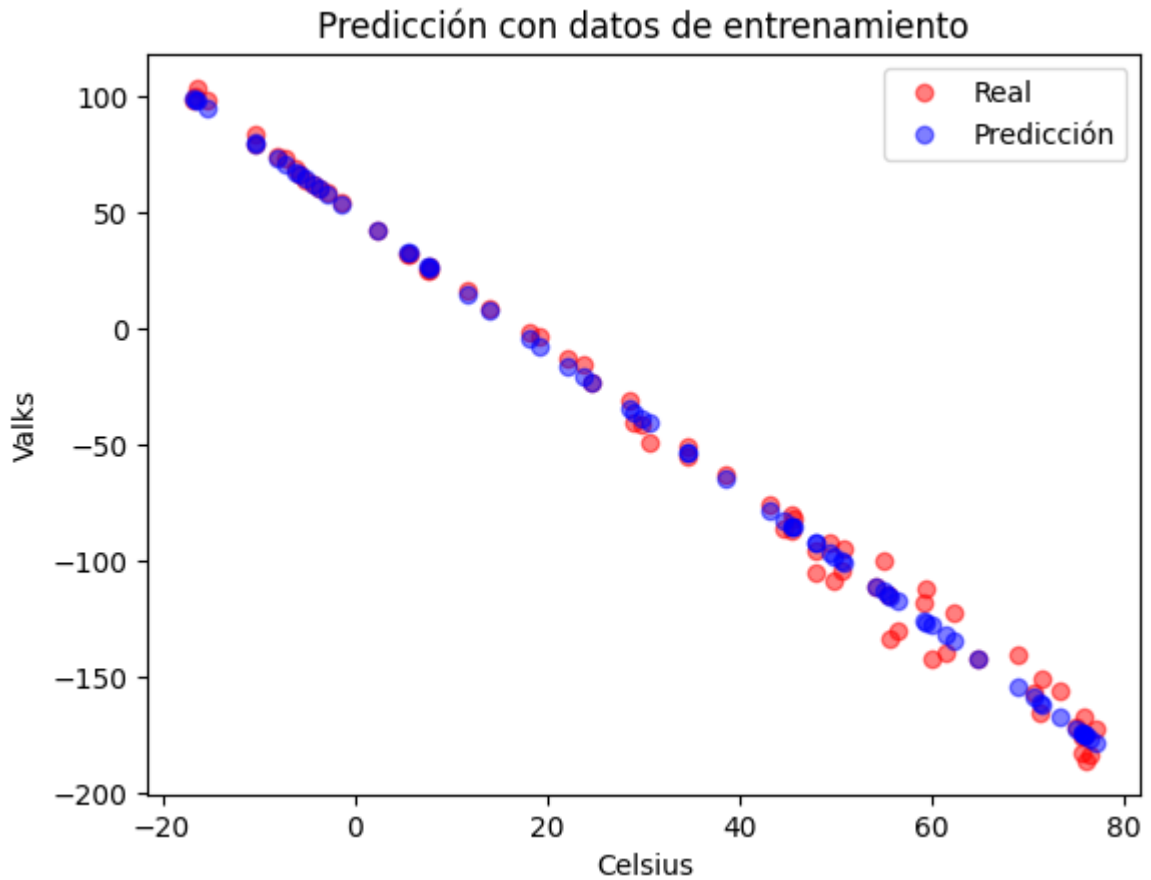
    promedio0 = sum0 / n
    promedio1 = sum1 / n

    # Actualizar theta0 y theta1
    theta0 = theta0 - alpha * promedio0
    theta1 = theta1 - alpha * promedio1

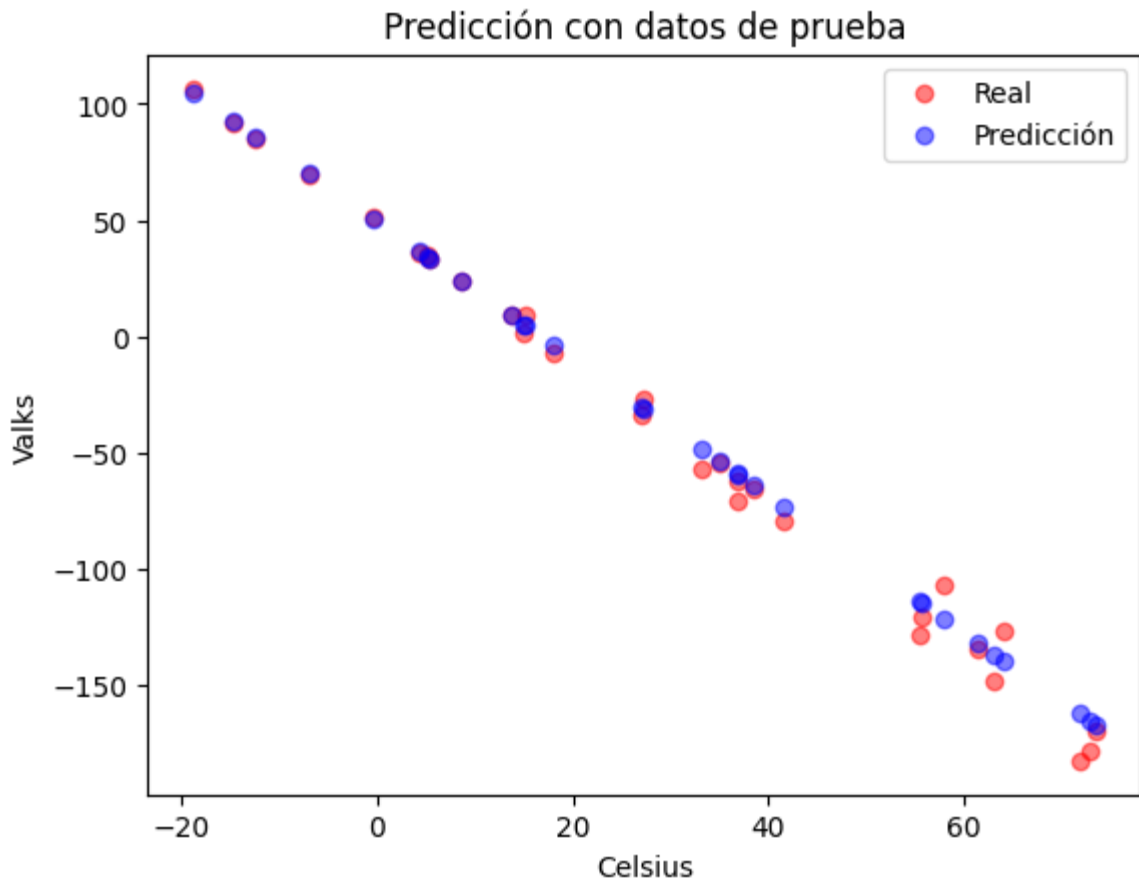
print(theta0)
print(theta1)
```

49.33967282772319  
-2.9508142866852873

```
In [4]: plt.plot(df_train['Celsius'],df_train['Valks'] , 'ro', label="Real", alpha=0.5)
plt.plot(df_train['Celsius'],df_train['Celsius']*theta1+theta0, 'bo', label="Predicci
plt.title("Predicción con datos de entrenamiento")
plt.xlabel('Celsius')
plt.ylabel('Valks')
plt.legend()
plt.show()
```



```
In [5]: plt.plot(df_test['Celsius'],df_test['Valks'], 'ro', label="Real" , alpha=0.5)
plt.plot(df_test['Celsius'],df_test['Celsius']*theta1+theta0, 'bo', label="Predicci
plt.title("Predicción con datos de prueba")
plt.xlabel('Celsius')
plt.ylabel('Valks')
plt.legend()
plt.show()
```



In [6]: *#Función de costos entrenamiento y prueba*

```
costoTrain = np.sum(((df_train['Celsius']*theta1+theta0) - (df_train['Valks']))) ** 2)
costoTest = np.sum(((df_test['Celsius']*theta1+theta0) - (df_test['Valks']))) ** 2)/2

print("Costo Train")
print(costoTrain)
print("Costo Test")
print(costoTest)
```

```
Costo Train
41.03348676508766
Costo Test
55.07102328049679
```

In [7]: `!jupyter nbconvert --to html "/content/drive/MyDrive/ColabNotebooks/Vallhala.ipynb"`

```
[NbConvertApp] Converting notebook /content/drive/MyDrive/ColabNotebooks/Vallhala.ipynb to html
[NbConvertApp] Writing 681292 bytes to /content/drive/MyDrive/ColabNotebooks/Vallhala.html
```