

CONVERTIR CELSIUS A FAHRENHEIT

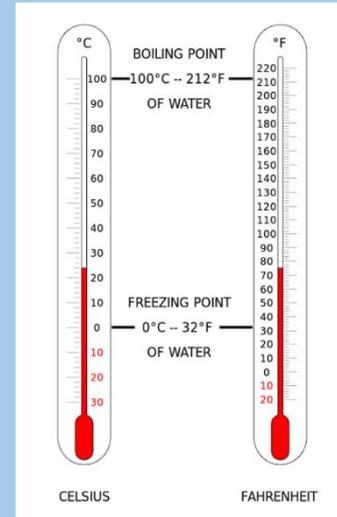
- En este proyecto, construiremos un simple modelo de aprendizaje de máquinas para convertir las temperaturas de Celsius a Fahrenheit.

- La ecuación es la siguiente:

- $T(^{\circ}\text{F}) = T(^{\circ}\text{C}) \times 9/5 + 32$

- Por ejemplo, convirtamos la temperatura de 0°C en Fahrenheit:

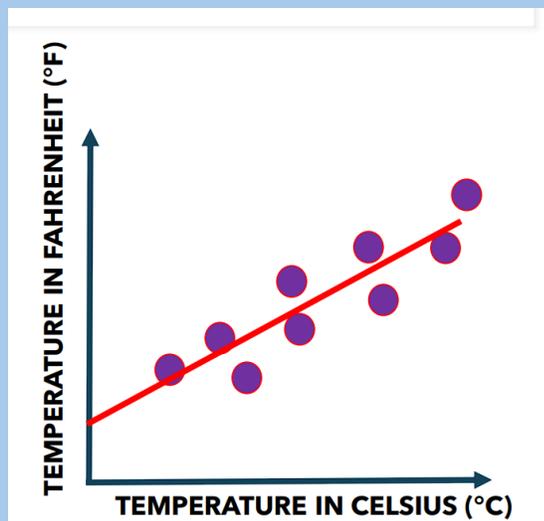
- $T(^{\circ}\text{F}) = (0^{\circ}\text{C} \times 9/5) + 32 = 32^{\circ}\text{F}$



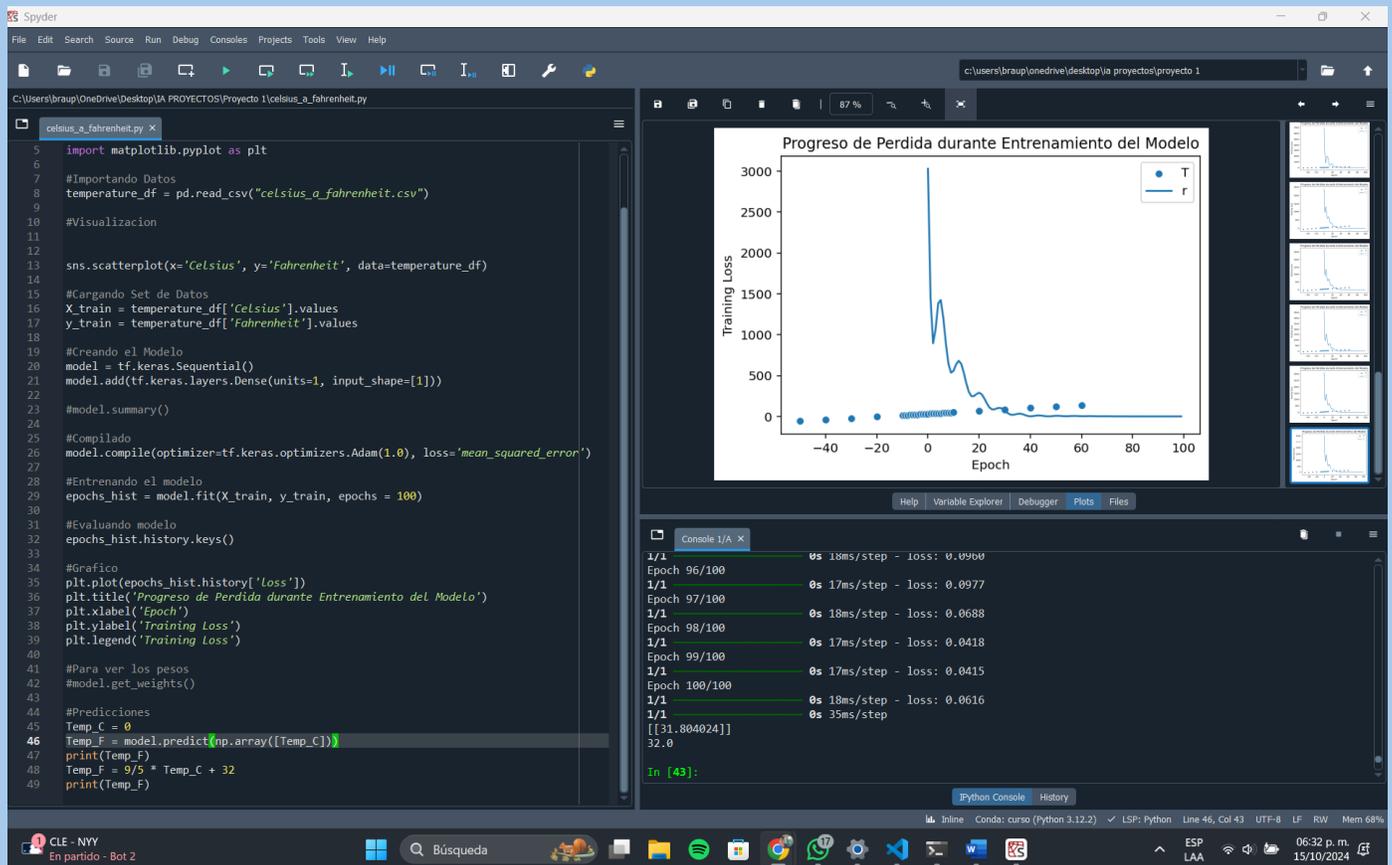
El objetivo es predecir el valor de una variable Y en base a otra variable X

X se llama la variable independiente e Y se llama la variable dependiente.

Esto se llama "Regresión" y se tratará con mucho más detalle en secciones posteriores del curso.



CONVERTIR CELSIUS A FAHRENHEIT



Anaconda es una distribución de Python y R que se utiliza principalmente en ciencia de datos, análisis de datos y aprendizaje automático. Incluye ambos lenguajes de programación y una gran cantidad de bibliotecas y herramientas necesarias para la ciencia de datos. Utiliza conda, un gestor de paquetes y entornos que facilita la instalación y actualización de bibliotecas, así como la creación de entornos de desarrollo aislados.

TensorFlow (import tensorflow as tf): Es una biblioteca de código abierto para el aprendizaje automático y la inteligencia artificial. Se utiliza para construir y entrenar modelos de machine learning y deep learning.

Pandas (import pandas as pd): Es una biblioteca para la manipulación y análisis de datos. Proporciona estructuras de datos como DataFrames, que facilitan el manejo de datos tabulares.

NumPy (import numpy as np): Es una biblioteca fundamental para la computación científica en Python. Ofrece soporte para arrays multidimensionales y funciones matemáticas de alto nivel.

CONVERTIR CELSIUS A FAHRENHEIT

Seaborn (`import seaborn as sns`): Es una biblioteca para la visualización de datos basada en Matplotlib. Facilita la creación de gráficos estadísticos atractivos y con estilo.

Matplotlib (`import matplotlib.pyplot as plt`): Es una biblioteca para la creación de gráficos en 2D. Se utiliza para generar una amplia variedad de visualizaciones, desde gráficos de líneas hasta histogramas y gráficos de dispersión.vb

Se carga un archivo CSV que contiene datos de temperaturas en Celsius y Fahrenheit.

Este proyecto utiliza técnicas de aprendizaje automático para predecir la temperatura en Fahrenheit a partir de la temperatura en Celsius. Aunque la conversión entre estas dos unidades de medida es sencilla y se puede realizar con una fórmula matemática directa, este ejercicio sirve para ilustrar cómo se puede aplicar el aprendizaje automático a problemas de regresión lineal.

Primero, se importan varias librerías de Python, como TensorFlow para construir y entrenar el modelo de aprendizaje automático, pandas para manejar los datos, NumPy para operaciones numéricas, y seaborn y matplotlib para la visualización de datos. Luego, se carga un archivo CSV que contiene datos de temperaturas en Celsius y Fahrenheit. Estos datos se utilizan para entrenar el modelo.

Se crea un gráfico de dispersión para visualizar la relación entre las temperaturas en Celsius y Fahrenheit. Esto ayuda a entender mejor los datos y a verificar que la relación entre las dos variables es lineal. A continuación, se extraen las columnas de Celsius y Fahrenheit del DataFrame y se convierten en arrays de NumPy. Estos arrays se utilizan como datos de entrada (Celsius) y salida (Fahrenheit) para entrenar el modelo.

Se construye un modelo secuencial de TensorFlow con una sola capa densa (neurona). Esta capa tiene una unidad y una entrada de una dimensión, lo que es adecuado para este problema de regresión lineal simple. El modelo se compila especificando el optimizador (Adam) y la función de pérdida (error cuadrático medio). El optimizador ajusta los pesos del modelo para minimizar la pérdida durante el entrenamiento.

CONVERTIR CELSIUS A FAHRENHEIT

El modelo se entrena con los datos de entrenamiento durante un número determinado de épocas (en este caso, 100). Durante el entrenamiento, el modelo ajusta sus pesos para aprender la relación entre las temperaturas en Celsius y Fahrenheit. Se imprime y grafica la pérdida del modelo durante el entrenamiento. La pérdida es una medida de qué tan bien el modelo está aprendiendo a predecir la temperatura en Fahrenheit a partir de la temperatura en Celsius.

Finalmente, se utiliza el modelo entrenado para predecir la temperatura en Fahrenheit para un valor dado de Celsius (por ejemplo, 0 grados Celsius). Además, se calcula la temperatura en Fahrenheit usando la fórmula matemática directa para comparación.

Referencias <https://youtu.be/TWk0YtF7pDM?si=bbBazA41utbGgSXk>

CODIGO:

```
import tensorflow as tf
import pandas as pd
import numpy as np
import seaborn as sns
import matplotlib.pyplot as plt

#Importando Datos
temperature_df = pd.read_csv("celsius_a_fahrenheit.csv")

#Visualizacion

sns.scatterplot(x='Celsius', y='Fahrenheit', data=temperature_df)

#Cargando Set de Datos
X_train = temperature_df['Celsius'].values
y_train = temperature_df['Fahrenheit'].values

#Creando el Modelo
model = tf.keras.Sequential()
model.add(tf.keras.layers.Dense(units=1, input_shape=[1]))

#model.summary()

#Compilado
model.compile(optimizer=tf.keras.optimizers.Adam(1.0), loss='mean_squared_error')

#Entrenando el modelo
epochs_hist = model.fit(X_train, y_train, epochs = 100)

#Evaluando modelo
epochs_hist.history.keys()

#Grafico
plt.plot(epochs_hist.history['loss'])
plt.title('Progreso de Perdida durante Entrenamiento del Modelo')
plt.xlabel('Epoch')
plt.ylabel('Training Loss')
plt.legend('Training Loss')

#Para ver los pesos
#model.get_weights()

#Predicciones
Temp_C = 0
Temp_F = model.predict(np.array([Temp_C]))
print(Temp_F)
Temp_F = 9/5 * Temp_C + 32
print(Temp_F)
```