roalimentaci195179n-m195179dulo-2

August 26, 2024

[14]:

```
import numpy as np
import pandas as pd
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LinearRegression
from sklearn.metrics import mean_squared_error, r2_score
import matplotlib.pyplot as plt
```

[15]:

```
# Leyendo los datos
df = pd.read_csv("Valhalla23.csv")
df.head()
```

[15]:

```
   Celsius    Valks
0  61.4720 -139.740
1  70.5790 -156.600
2  -7.3013   73.269
3  71.3380 -165.420
4  43.2360  -75.835
```

[16]:

```
# Dividimos los datos en variable dependiente e independiente
X = df['Celsius']
y = df['Valks']
```

[17]:

```
# Dividimos los datos en prueba y entrenamiento
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2,␣
 ↪random_state=42)
```

[18]:

```
# Armamos el modelo
model = LinearRegression()
model.fit(X_train.values.reshape(-1, 1), y_train)
```

[18]:

```
LinearRegression()
```

[19]:

```
# Realizamos predicciones
y_train_pred = model.predict(X_train.values.reshape(-1, 1))
y_test_pred = model.predict(X_test.values.reshape(-1, 1))
```

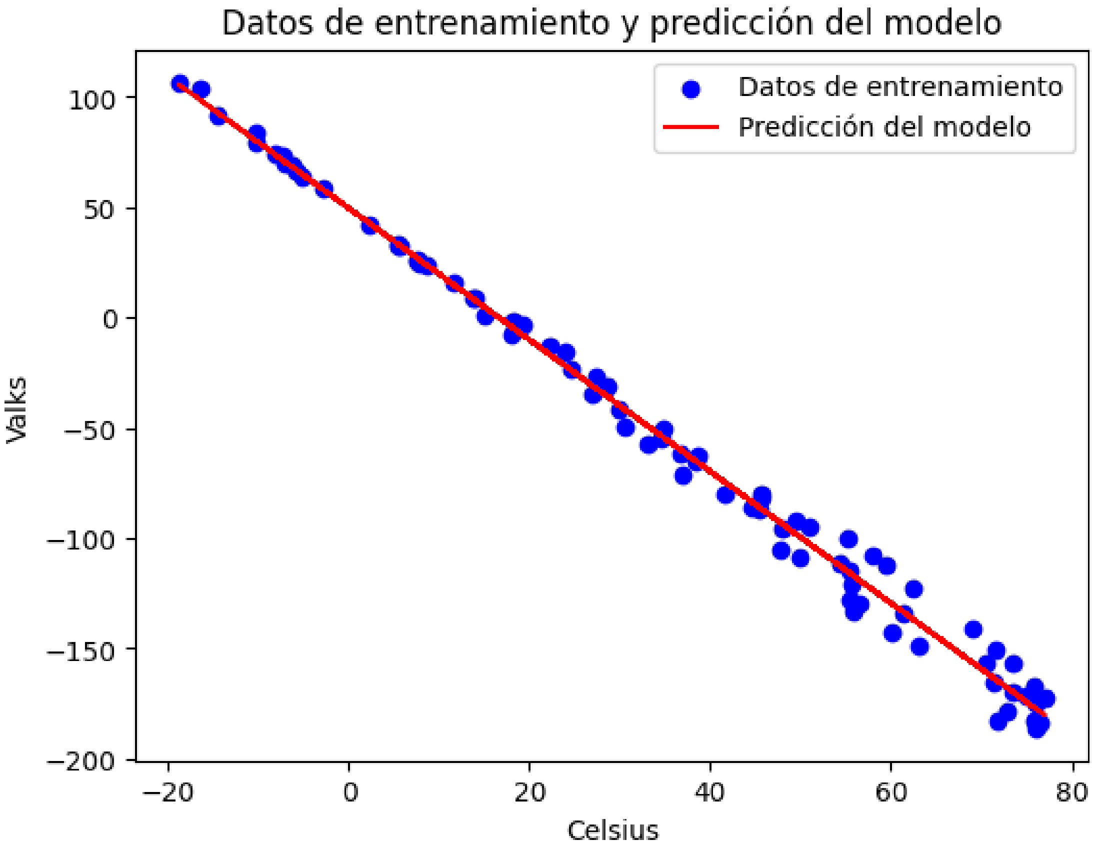
[20]:

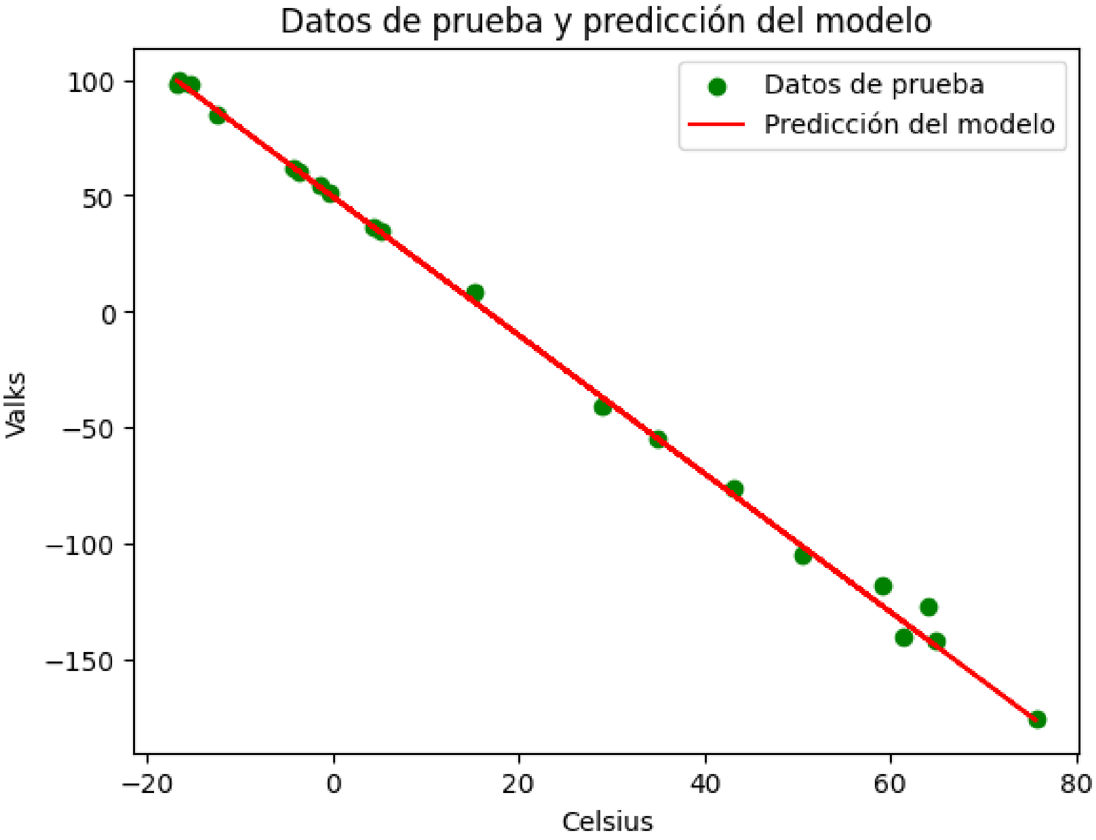
```python
# Métricas de entrenamiento
train_mse = mean_squared_error(y_train, y_train_pred)
train_r2 = r2_score(y_train, y_train_pred)

# Métricas de prueba
test_mse = mean_squared_error(y_test, y_test_pred)
test_r2 = r2_score(y_test, y_test_pred)

print(f'Training MSE: {train_mse}, R^2: {train_r2}')
print(f'Test MSE: {test_mse}, R^2: {test_r2}')
```

```
Training MSE: 50.48822005413272, R^2: 0.993035561819186
Test MSE: 20.18813776711824, R^2: 0.9976240798987097
```

[21]:

```python
# Gráfico para datos de entrenamiento
plt.scatter(X_train, y_train, color='blue', label='Datos de entrenamiento')
plt.plot(X_train, y_train_pred, color='red', label='Predicción del modelo')
plt.xlabel('Celsius')
plt.ylabel('Valks')
plt.title('Datos de entrenamiento y predicción del modelo')
plt.legend()
plt.show()

# Gráfico para datos de prueba
plt.scatter(X_test, y_test, color='green', label='Datos de prueba')
plt.plot(X_test, y_test_pred, color='red', label='Predicción del modelo')
plt.xlabel('Celsius')
plt.ylabel('Valks')
plt.title('Datos de prueba y predicción del modelo')
plt.legend()
plt.show()
```


Datos de entrenamiento y predicción del modelo
Datos de entrenamiento
Predicción del modelo
Valks
100
50
0
−50
−100
−150
−200
Celsius
−20
0
20
40
60
80


Datos de prueba y predicción del modelo
● Datos de prueba
Predicción del modelo
Valks
100
50
0
−50
−100
−150
−20
0
20
40
60
80
Celsius