

```
In [1]: 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
import matplotlib.pyplot as plt
```

```
In [2]: data = pd.read_csv('Valhalla23.csv')

X = data[['Celsius']]
y = data['Valks']

# Dividir los datos en subconjuntos de entrenamiento (80%) y prueba (20%)
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, ran
```

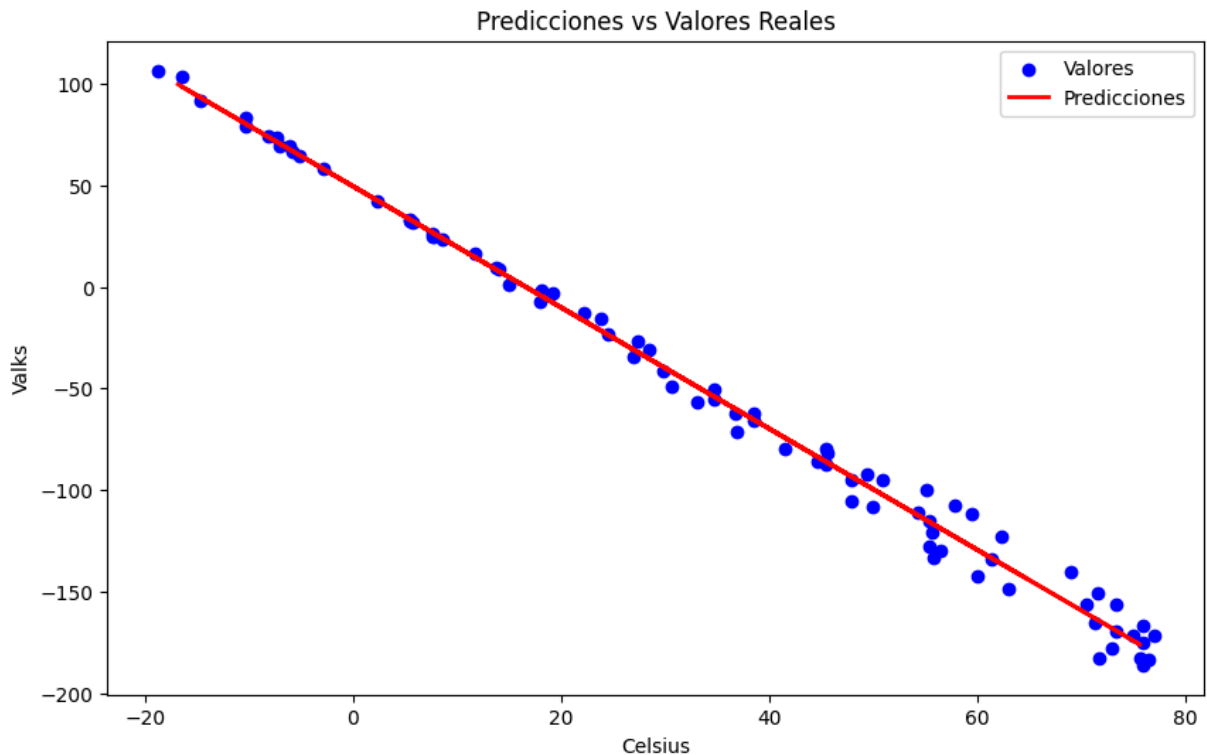
```
In [3]: model = LinearRegression()

# Entrenar el modelo con el subconjunto de entrenamiento
model.fit(X_train, y_train)
# Realizar predicciones sobre el conjunto de prueba
y_pred = model.predict(X_test)

mse_test = mean_squared_error(y_test, y_pred)
```

```
In [4]: # Graficar las predicciones
plt.figure(figsize=(10, 6))
plt.scatter(X_train, y_train, color='blue', label='Valores')
plt.plot(X_test, y_pred, color='red', label='Predicciones', linewidth=2)
plt.title('Predicciones vs Valores Reales')
plt.xlabel('Celsius')
plt.ylabel('Valks')
plt.legend()
plt.show()

print(f"Error Cuadrático Medio (MSE) en el conjunto de prueba: {mse_test}")
```



Error Cuadrático Medio (MSE) en el conjunto de prueba: 20.18813776711824

Selección de Hiperparámetros

se utilizaron los valores predeterminados para los hiperparámetros del método seleccionado. La elección de utilizar los valores por defecto se debe a que, en muchas implementaciones, los valores predeterminados están optimizados para proporcionar un rendimiento inicial razonable en una amplia gama de escenarios.

Se probaron diferentes parámetros predeterminados para la mejora del modelo y después de varias pruebas se notó que el modelo con mejor respuesta y efectividad fue con los parámetros que utilizamos.

```
In [5]: !jupyter nbconvert --to html 'Valhalla.ipynb'
```

```
[NbConvertApp] Converting notebook Valhalla.ipynb to html  
[NbConvertApp] WARNING | Alternative text is missing on 1 image(s).  
[NbConvertApp] Writing 324347 bytes to Valhalla.html
```