



Como se puede observar dentro del anterior caso se necesitan mas cantidad de muestras que solo 2, ya que estas tan pequeñas muestras hacen que el error cuadratico de validacion sea extremadamente alto en comparacion con el de entrenamiento, esto quiere decir que el modelo no tiene datos suficientes para capturar patrones. Entonces de los casos presentados el mejor es el numero 10 debido a su bajo MSE de Validacion.

```
indices = random.sample(range(len(X_train)), 10)
indices
X_train_subset = X_train.iloc[indices]
y_train_subset = y_train.iloc[indices]

# Crear y entrenar el modelo base SGDRegressor
model = SGDRegressor(learning_rate='constant', eta0=1e-4, max_iter=1000000, random_
model.fit(X_train_subset, y_train_subset)

# Calcular el MSE en el subconjunto de entrenamiento
y_train_subset_pred = model.predict(X_train_subset)
mse_train_subset = mean_squared_error(y_train_subset, y_train_subset_pred)

# Calcular el MSE en el conjunto de validación
y_valid_pred = model.predict(X_valid)
mse_valid = mean_squared_error(y_valid, y_valid_pred)

# Calcular el MSE en el conjunto de prueba
y_test_pred = model.predict(X_test)
mse_test = mean_squared_error(y_test, y_test_pred)

# Imprimir resultados
print(f"MSE en el subconjunto de entrenamiento (10 muestras): {mse_train_subset}")
print(f"MSE en el conjunto de validación: {mse_valid}")
print(f"MSE en el conjunto de prueba: {mse_test}")

➡ MSE en el subconjunto de entrenamiento (10 muestras): 32.88228895569166
MSE en el conjunto de validación: 71.07586302559147
MSE en el conjunto de prueba: 74.50550128199356
```