

Redes Neuronales

Tarea 4: Random Forest

Realizado por: David Cevallos

Fecha: 2023-07-27

Enlace Google Colab: https://colab.research.google.com/drive/1TjaF2EfA5GJzp_5tPnFU9qQMSTCS9lks#scrollTo=KQs9zA4husbB

En la presente tarea analizaremos mediante Random Forests los datasets Computer Hardware, carsmall y Tratamiento Regresion para compararlos con los modelos de regresión logrados a través de un único árbol de decisión.

Para correr el presente notebook, procure cargar en la carpeta home los siguientes ficheros:

- machine.data
- carsmall.txt
- TratamientoRegresion.xlsx

```
1 # Importación de librerías
2 import pandas as pd
3 import numpy as np
4 import sklearn.metrics as mt
5 from sklearn import tree
6 from sklearn.model_selection import KFold
7 import matplotlib.pyplot as plt
8 from sklearn.ensemble import RandomForestRegressor
9 from sklearn.preprocessing import LabelEncoder

1 # Función para cálculo de MSE medio a través de árbol de regresión y Random Forest
2 # y estrategia K-folding
3 # Parámetros:
4 #   X           Valores para predicción
5 #   Y           Valores de regresión
6 #   k           Número de folds
7 #   min_samples_leaf Número mínimo de observaciones para una hoja
8 #   graph       Bandera que determina si mostrar diagrama de árbol de
9 #               decisión o no
10 def calcularMSE(X, Y, k=10, min_samples_leaf=10, graph=False):
11     kfold = KFold(n_splits=k, shuffle=True, random_state=0)
12     total_mse = 0;
13     total_mse_rf = 0;
14
15     for i, (train_index, test_index) in enumerate(kfold.split(X)):
16         print("-----")
17         print("Resultados para Fold Nro.",str(i+1))
18         print("-----")
19         # Definición y entrenamiento de árbol de regresión
20         modelo = tree.DecisionTreeRegressor(criterion="squared_error",
21                                             random_state=0,
22                                             min_samples_leaf=min_samples_leaf
23                                             )
24         modelo.fit(X[train_index,:], Y[train_index])
25         # Obtención de valores predichos para el fold (iteración)
26         Ypred = modelo.predict(X[test_index,:])
27         # Obtenemos el valor de MSE para el fold
28         fold_mse = mt.mean_squared_error(Y[test_index], Ypred)
29         print("Valor MSE (Decision Tree) = ", fold_mse)
30         total_mse = total_mse + fold_mse;
31
32         # Definición y entrenamiento de Random Forest
33         modelo_rf = RandomForestRegressor(n_estimators=200, criterion="squared_error", random_state=0)
34         modelo_rf.fit(X[train_index,:], Y[train_index])
35         Ypred_rf = modelo_rf.predict(X[test_index,:])
36         fold_mse_rf = mt.mean_squared_error(Y[test_index], Ypred_rf)
37         print("Valor MSE (Random Forest) = ", fold_mse_rf)
38         total_mse_rf = total_mse_rf + fold_mse_rf;
39
40         # Presentar diagrama de valores reales vs predichos
41         plt.figure(figsize=(8,4))
```

```

42 plt.scatter(np.arange(1,len(Y[test_index])+1), Y[test_index], c="blue", marker="o", label="Reales")
43 plt.scatter(np.arange(1,len(Ypred)+1), Ypred, c="red", marker="*", label="Predecidos (Decision Tree)")
44 plt.scatter(np.arange(1,len(Ypred_rf)+1), Ypred_rf, c="green", marker="*", label="Predecidos (Random Forest)")
45 plt.title("Valores reales vs Valores predecidos (Test)")
46 plt.legend(loc="best")
47 plt.xticks(np.arange(1,len(Ypred)+1))
48 plt.grid()
49 plt.show()
50
51 # Presentar diagrama de árbol de decisión
52 if (graph):
53     plt.figure(figsize=(15,8))
54     tree.plot_tree(modelo, fontsize=7)
55     plt.show()
56
57 # Error Cuadrático Medio final
58 mean_mse = total_mse/k;
59 mean_mse_rf = total_mse_rf/k;
60 print("-----")
61 print("Error Cuadrático Medio final (Decision Tree): ", mean_mse);
62 print("Error Cuadrático Medio final (Random Forest): ", mean_mse_rf);
63 print("-----")

```

▼ Dataset Computer Hardware

```

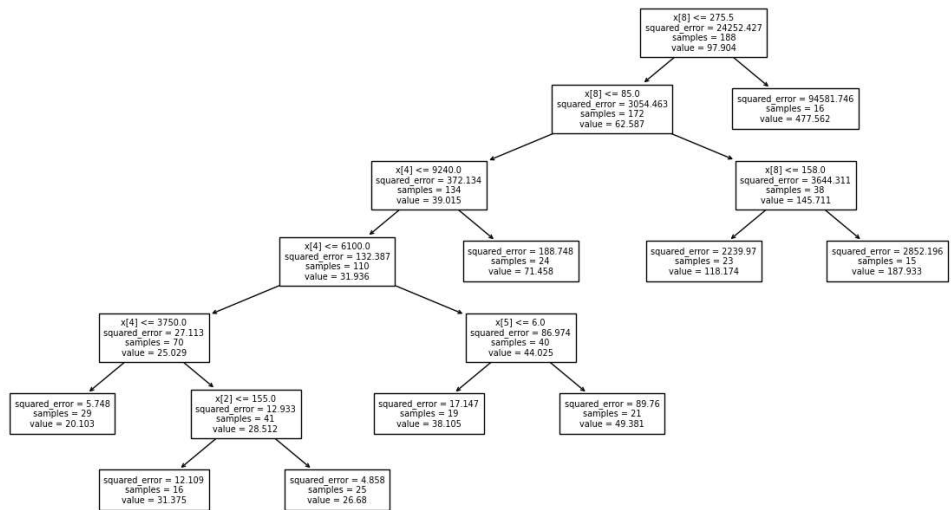
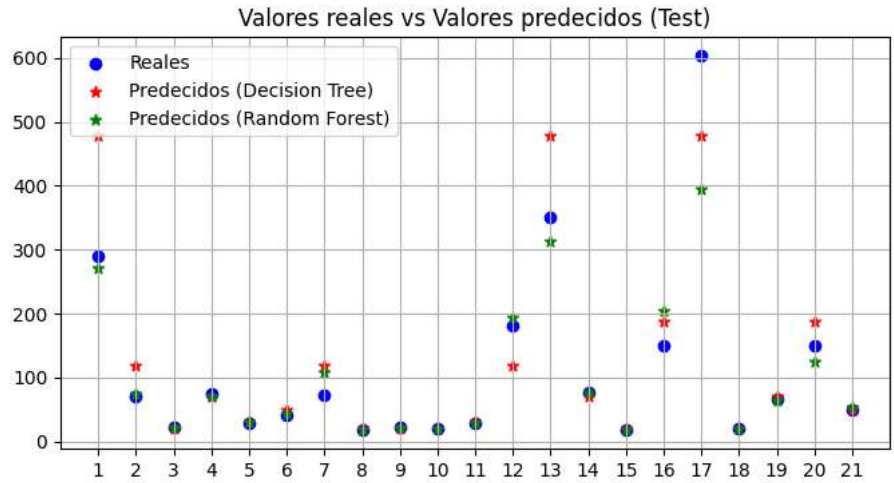
1 # Ejercicio 1: Dataset Computer Hardware
2 print("-----")
3 print("Ejercicio 1: Dataset Computer Hardware")
4 print("-----")
5
6 data = pd.read_csv("/home/machine.data", header=None)
7
8 data[0] = LabelEncoder().fit_transform(data[0])
9 data[1] = LabelEncoder().fit_transform(data[1])
10
11 X = data.drop(columns=[9]).to_numpy()
12 Y = data[9].to_numpy()
13
14 calcularMSE(X, Y, 10, 15, True)

```

Ejercicio 1: Dataset Computer Hardware

Resultados para Fold Nro. 1

Valor MSE (Decision Tree) = 3743.6531862307434
Valor MSE (Random Forest) = 2408.2837869047617



Resultados para Fold Nro. 2

Valor MSE (Decision Tree) = 1097.2532665391157
Valor MSE (Random Forest) = 1251.4945583333338

