Árboles de decisión - Práctica

En esta entrada explicaremos la parte práctica del algoritmo de Árbol de Decisión Regresión, en donde desarrollaremos un modelo para predecir el precio de las casas en Boston de acuerdo al número de habitaciones que cuenta la vivienda.

Para este análisis vamos a utilizar el dataset disponible en la librería scikit-learn correspondiente al Boston Housing o las casas de Boston. Este es el mismo dataset que utilizamos para la práctica de los anteriores algoritmos que hemos desarrollados por lo que no explicare cómo obtener los datos, su análisis ni su preprocesamiento, todo esto está explicado es las primeras entradas de práctica, por lo que si no los haz visto te recomiendo hacerlo para entender mejor esta parte.

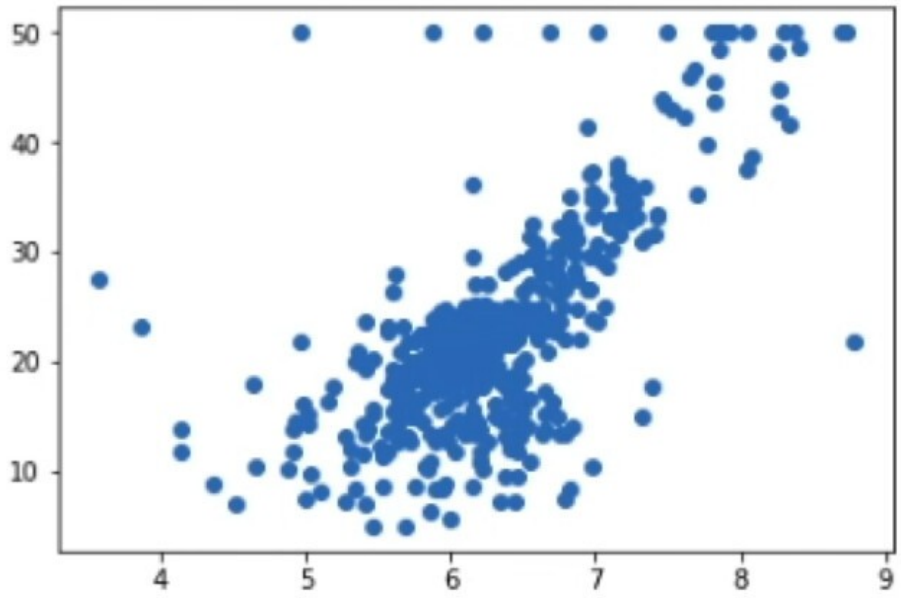
Acá vamos arrancar a preparar los datos para implementar el algoritmo de Árboles de Decisión Regresión.

Como todo proyecto de Machine Learning una vez que se haya realizado el respectivo preprocesamiento a los datos lo primero que debemos haces es separar las variables independientes y la dependiente para convertirlos en lo que en nuestro programa llamaremos X_adr y y_adr. La primera corresponde a los datos ubicados en la columna 6 del dataset y la segunda variable corresponde a la columna de target.

```
#Seleccionamos solamente la columna 6 del dataset
X_adr = boston.data[:, np.newaxis, 5]
#Defino los datos correspondientes a las etiquetas
y_adr = boston.target
```

Si graficamos estos datos tenemos lo siguiente. Como podemos observar es un conjunto de datos que se encuentra distribuido de manera uniforme.

```
#Graficamos los datos correspondientes
plt.scatter(X_adr, y_adr)
plt.show()
```



Árboles de decisión - Práctica

Ahora si empecemos a crear el modelo, lo primero que debemos hacer es separar los datos en entrenamiento y prueba.

```
from sklearn.model_selection import train_test_split
#Separo los datos de "train" en entrenamiento y prueba para probar los algoritmos
X_train, X_test, y_train, y_test = train_test_split(X_adr, y_adr, test_size=0.2)
```

Seguidamente importamos el algoritmo, por lo que definimos from sklearn.tree, importamos DecisionTreeRegressor, con esto ya podemos empezar a utilizar todas las instrucciones contenida en esta librería.

```
from sklearn.tree import DecisionTreeRegressor
#Defino el algoritmo a utilizar
adr = DecisionTreeRegressor(max_depth = 5)
```

Por lo cual definimos el algoritmo indicando adr será igual a DecisionTreeRegressor y procedemos a realizar la configuración respectiva al mismo.

De acuerdo a lo visto anteriormente, tanto en la teoría como en la descripción de este algoritmo en Scikit Learn, esta instrucción cuenta con varias configuraciones que pueden ser modificadas, pero acá la que mas no interesa es max_depth ya que queremos evitar un sobreajuste en el modelo por lo que debemos detener que el árbol sea muy profundo. Este valor lo puedes modificar, para nuestro ejemplo vamos a utilizar max_depth igual a 5.

Definido esto, ahora si podemos entrenar el algoritmo utilizando la instrucción fit y los datos de entrenamiento.

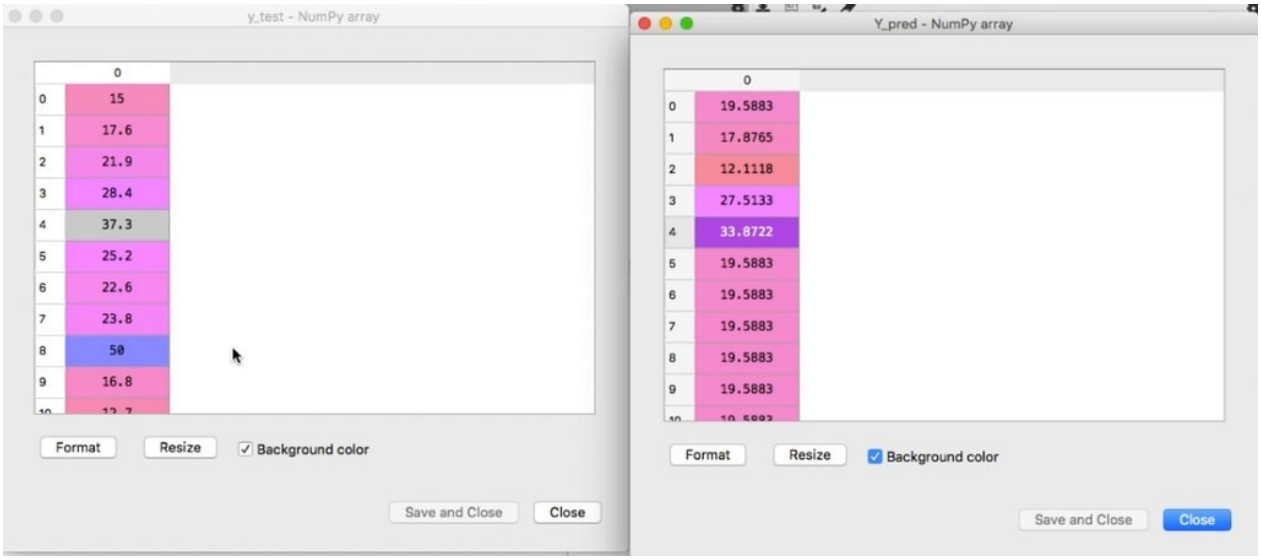
```
#Entreno el modelo
adr.fit(X_train, y_train)
```

Seguidamente realizamos una predicción utilizando los datos de prueba.

```
#Realizo una predicción
Y_pred = adr.predict(X_test)
```

Realizado todo esto comparemos los valores obtenidos en nuestra predicción con los valores reales, para ver que tal es el comportamiento de nuestro modelo.

Árboles de decisión - Práctica



Como podemos observar en este recuadro tenemos los datos reales mientras que en este otro están los datos obtenidos de nuestra predicción implementando el modelo. Si observamos con detenimiento estos datos nos podemos dar cuenta que en ciertos momentos los datos reales son muy parecidos con los datos que se han calculados mientras que en otros casos los datos son totalmente distintos.

Visualicemos los datos de entrenamiento junto con el modelo, para ello vamos a requerir hacer un pequeño truco para que de esta forma el modelo se grafique de manera adecuada.

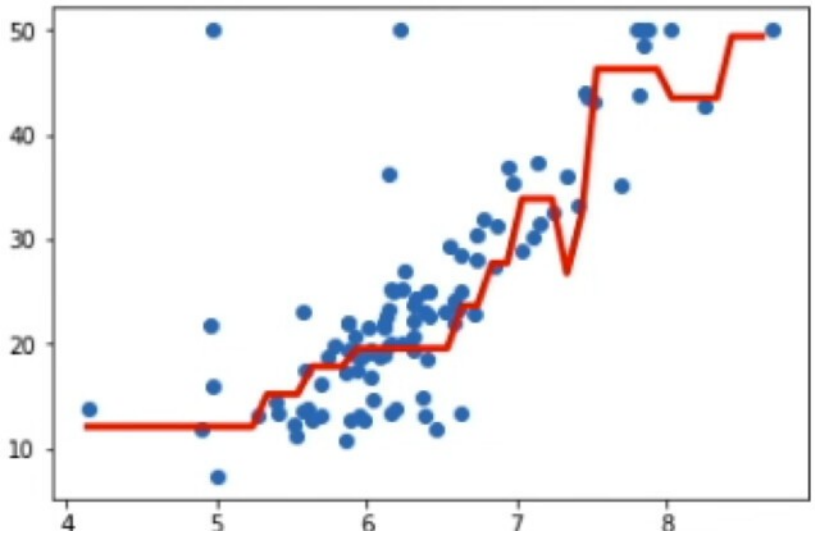
```
#Graficamos los datos de prueba junto con la predicción
X_grid = np.arange(min(X_test), max(X_test), 0.1)
X_grid = X_grid.reshape((len(X_grid), 1))
plt.scatter(X_test, y_test)
plt.plot(X_grid, adr.predict(X_grid), color='red', linewidth=3)
```

Lo primero que vamos hacer es utilizar la instrucción arange para espaciar de manera uniforme los datos, el intervalo a utilizar será el valor mínimo y máximo de los datos de prueba y los pasos para el espaciado será de 0.1.

Una vez realizado esto, procedemos a utilizar la instrucción reshape para darle una nueva forma a la matriz de los datos y de esta forma puedan ser graficados.

Finalizado esto ya podemos graficar los datos pero ahora deberemos utilizar los datos X_grid y su respectiva predicción.

Árboles de decisión - Práctica



Como podemos observar acá el modelo no es una línea recta como la habiamos visto en los anteriores algoritmos, acá podemos observar que el modelo cuenta con subidas y bajadas.

Pero ahora veamos que tal es la precisión del mismo utilizando la instrucción score. El resultado obtenido acá es de 0,669.

```
print('DATOS DEL MODELO ÁRBOLES DE DECISIÓN REGRESION')
print()
print('Precisión del modelo:')
print(adr.score(X_train, y_train))
```

```
DATOS DEL MODELO ÁRBOLES DE DECISIÓN REGRESION

Precisión del modelo:
0.6696045715320864
```

Te aconsejo a que hagas modificaciones a la configuración del algoritmo para que veas el comportamiento del modelo y ver si existe mejora en el mismo o no. De esta forma puedes practicar tus conocimientos en Machine Learning.

Árboles de decisión - Práctica

El programa completo es el siguiente:

```
"""
Árboles de Decisión Regresión
"""
########## LIBRERÍAS A UTILIZAR ##########
#Se importan la librerias a utilizar
import numpy as np
import matplotlib.pyplot as plt
from sklearn import datasets
########## PREPARAR LA DATA ##########
#Importamos los datos de la misma librería de scikit-learn
boston = datasets.load_boston()
print(boston)
print()
########## ENTENDIMIENTO DE LA DATA ##########
#Verifico la información contenida en el dataset
print('Información en el dataset:')
print(boston.keys())
print()
#Verifico las características del dataset
print('Características del dataset:')
print(boston.DESCR)
#Verifico la cantidad de datos que hay en los dataset
print('Cantidad de datos:')
print(boston.data.shape)
print()
#Verifico la información de las columnas
print('Nombres columnas:')
print(boston.feature_names)
########## PREPARAR LA DATA ÁRBOLES DE DECISIÓN REGRESIÓN
##########
#Seleccionamos solamente la columna 6 del dataset
X_adr = boston.data[:, np.newaxis, 5]
#Defino los datos correspondientes a las etiquetas
y_adr = boston.target
#Graficamos los datos correspondientes
plt.scatter(X_adr, y_adr)
plt.show()
########## IMPLEMENTACIÓN DE ÁRBOLES DE DECISIÓN REGRESIÓN
##########
from sklearn.model_selection import train_test_split
#Separo los datos de "train" en entrenamiento y prueba para probar los algoritmos
X_train, X_test, y_train, y_test = train_test_split(X_adr, y_adr, test_size=0.2)
from sklearn.tree import DecisionTreeRegressor
```

Árboles de decisión - Práctica

```
#Defino el algoritmo a utilizar
adr = DecisionTreeRegressor(max_depth = 5)
#Entreno el modelo
adr.fit(X_train, y_train)
#Realizo una predicción
Y_pred = adr.predict(X_test)
#Graficamos los datos de prueba junto con la predicción
X_grid = np.arange(min(X_test), max(X_test), 0.1)
X_grid = X_grid.reshape((len(X_grid), 1))
plt.scatter(X_test, y_test)
plt.plot(X_grid, adr.predict(X_grid), color='red', linewidth=3)
plt.show()
print('DATOS DEL MODELO ÁRBOLES DE DECISIÓN REGRESION')
print()
print('Precisión del modelo:')
print(adr.score(X_train, y_train))
```