INSTITUTO POLITÉCNICO NACIONAL

ESCUELA SUPERIOR DE CÓMPUTO

Minería de Datos

Practica
ARBOL DE REGRESIÓN

Integrantes:

- **Sánchez Zarazúa Jesua Antonio**

Profesora: Ocampo Botello Fabiola
Grupo: 3CM9
Fecha de entrega: 03/04/2020

Practica. Arboles de Clasificación

INTRODUCCION

Los árboles de decisión son un de la familia de modelos de aprendizaje automático más utilizados. Se pueden utilizar tanto para resolver problemas de clasificación como de regresión. Una de sus principales ventajas es la facilidad con la que se puede interpretar los resultados en base a reglas. Permitiendo no solo obtener un resultado, sino que inspeccionar los motivos por los que se llega a una predicción dada.

Los modelos de Árbol de Regresión y fueron introducidos en la Estadística por Breiman (1984). Los autores utilizan el término “modelos de árbol de regresión” cuando la variable respuesta es cuantitativa y el de “modelos de árbol de clasificación” cuando ésta es cualitativa.

Esta practica se realizo usando el lenguaje de programación Python. Además de las librerías para machine learning Scikit Learn y otras librerias especificas como Pandas para la lectura del dataset .CSV y Graphviz para construir una imagen del modelo del árbol resultado del aprendizaje

También se ha de recalcar que se uso Jupyter Notebook como herramienta de desarrollo, por lo que el resultado es un Notebook que posee cada línea de código y resultado obtenido, visible sin la necesidad de compilar el archivo.

INTENCION RECOPILACION DE DATOS

Este conjunto de datos contiene información recopilada por el Servicio de Censos de EE. UU. Sobre viviendas en el área de Boston Mass. Se obtuvo del archivo StatLib (http://lib.stat.cmu.edu/datasets/boston), y se ha utilizado ampliamente en todo. La literatura para los algoritmos de referencia. Sin embargo, estas comparaciones se realizaron principalmente fuera de Delve y, por lo tanto, son algo sospechosas. El conjunto de datos es de tamaño pequeño con solo 506 casos.

Fuente:

https://www.cs.toronto.edu/~delve/data/boston/bostonDetail.html

Boston Housing dataset

Diccionario de Datos

- CRIM - tasa de criminalidad per cápita por ciudad
- ZN: proporción de tierra residencial dividida en zonas para lotes de más de 25,000 pies cuadrados.
- INDUS: proporción de acres de negocios no minoristas por ciudad.
- CHAS: variable ficticia Charles River (1 si el tramo limita con el río; 0 en caso contrario)
- NOX - concentración de óxidos nítricos (partes por 10 millones)
- RM - número promedio de habitaciones por vivienda
- EDAD - proporción de unidades ocupadas por el propietario construidas antes de 1940
- DIS - distancias ponderadas a cinco centros de empleo de Boston
- RAD - índice de accesibilidad a carreteras radiales
- IMPUESTO: tasa impositiva sobre el valor total de la propiedad por cada $ 10,000
- PTRATIO - relación alumno-profesor por localidad
- B - 1000 (Bk - 0.63) ^ 2 donde Bk es la proporción de negros por ciudad
- LSTAT -% de la poblacion con un menor status economico de la población
- MEDV: valor medio de las viviendas ocupadas por sus propietarios en $ 1000

Numero de Datos: 506 filas

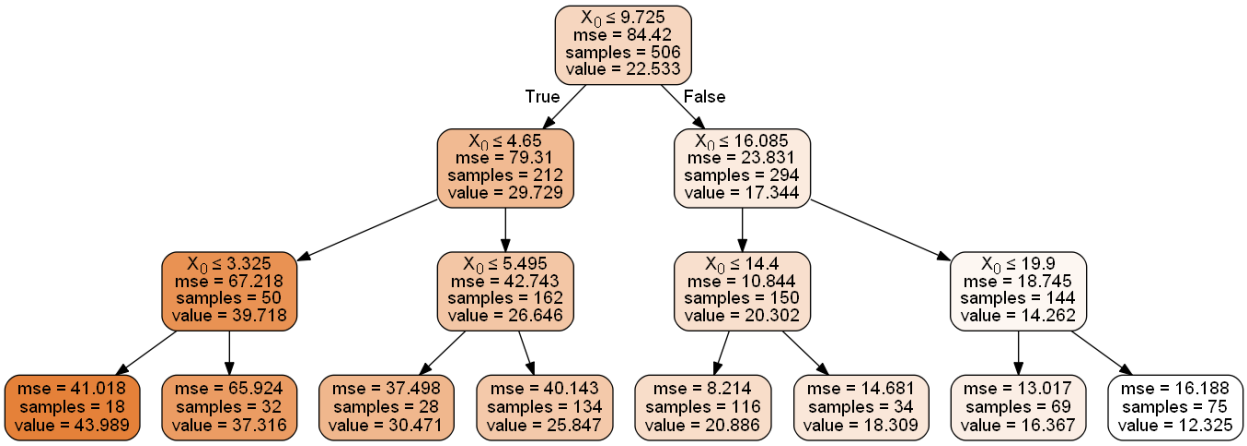
Numero de Columnas: 14 Columnas

Tipos de Datos contenidos: Numericos(int)

OBJETIVO DE ANALISIS DE DATOS

Se entiende que los valores contenidos son validos para todos los casos puesto que el dataset esta incluido en los conjuntos disponibles para la librería Sci Kit Learn. Para resolver el ejercicio se decidio tomar como variable independiente los datos que corresponden a la columna LSTAT la cual ilustra el porcentaje de la poblacion con un estatus social bajo.

WORKFLOW ARBOL



MEDIDAS

In [48]:
```
tree = DecisionTreeRegressor(criterion='mse',     # Initialize and fit regressor
                             max_depth=3)
tree.fit(X, y)
#x_pred = tree.predict(X)
```

Out[48]:
```
DecisionTreeRegressor(criterion='mse', max_depth=3, max_features=None,
           max_leaf_nodes=None, min_impurity_decrease=0.0,
           min_impurity_split=None, min_samples_leaf=1,
           min_samples_split=2, min_weight_fraction_leaf=0.0,
           presort=False, random_state=None, splitter='best')
```

ANEXO: Cuaderno Completo de Jupyter Notebook

jupyter ArbolRegresor_BostonDataSet Last Checkpoint: hace unos segundos (autosaved)

Logout

File Edit View Insert Cell Kernel Help Trusted Python 3

In [2]:
```
import pandas as pd
from sklearn import datasets
from sklearn.tree import DecisionTreeRegressor
from matplotlib import pyplot as plt
from sklearn import metrics
from sklearn.tree import export_graphviz
from sklearn.externals.six import StringIO
import pydotplus
from IPython.display import Image
```

In [3]:
```
boston = datasets.load_boston()
df = pd.DataFrame(boston.data[:, 12])      # Create DataFrame using only the LSAT feature
df.columns = ['LSTAT']
df['MEDV'] = boston.target                 # Create new column with the target MEDV
df.head()
```

Out[3]:

	LSTAT	MEDV
0	4.98	24.0

In [4]:
```
X = df[['LSTAT']].values                        # Asignamos la columna de la variable independiente
y = df['MEDV'].values
sort_idx = X.flatten().argsort()
X = X[sort_idx]
y = y[sort_idx]
```

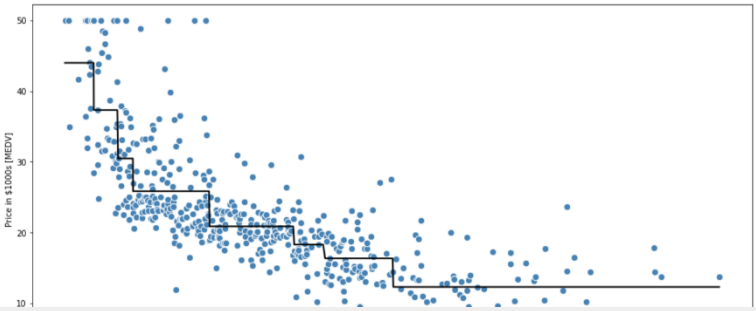
In [5]:
```
tree = DecisionTreeRegressor(criterion='mse',      # Inicializamos el arbol en modo regresor
                             max_depth=3)

tree.fit(X, y)
#x_pred = tree.predict(X)
```

Out[5]:
```
DecisionTreeRegressor(criterion='mse', max_depth=3, max_features=None,
           max_leaf_nodes=None, min_impurity_decrease=0.0,
           min_impurity_split=None, min_samples_leaf=1,
           min_samples_split=2, min_weight_fraction_leaf=0.0,
           presort=False, random_state=None, splitter='best')
```

In [12]:
```
x_pred = tree.predict(X)
```

In [13]:
```
plt.figure(figsize=(16, 8))
plt.scatter(X, y, c='steelblue',
```




```
graph = pydotplus.graph_from_dot_data(dot_data.getvalue())
graph.write_png('boston.png')
Image(graph.create_png())
```

Out[14]: