**Algorithm**

**(Code)**

**2) Decision Tree Regression:**

* import pandas as pd

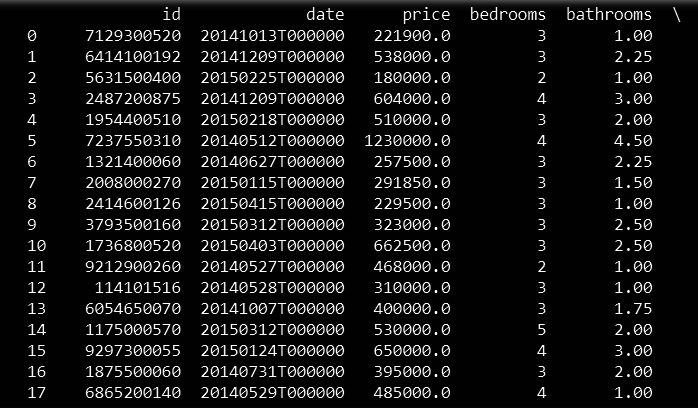
import numpy as np

import matplotlib.pyplot as plt

import seaborn as sns

* data = pd.read\_csv("housingnew.csv")

print(data)



* data.describe()



* input = data.drop(["id","date","price"], axis = 1)

outcome = data.price

print(input)

print(outcome)



* ***# import the regressor***

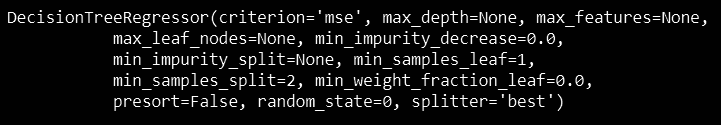
from sklearn.tree import DecisionTreeRegressor

***# create a regressor object***

regressor = DecisionTreeRegressor(random\_state = 0)

***# fit the regressor with X and Y data***

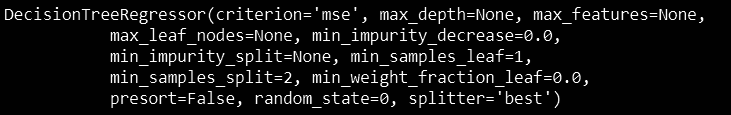
regressor.fit(input, outcome)



* from sklearn.model\_selection import train\_test\_split

x\_train,x\_test,y\_train,y\_test = train\_test\_split(input,outcome,test\_size = 0.2,random\_state=7)

* regressor.fit(x\_train,y\_train)



* regressor.score(x\_test,y\_test)



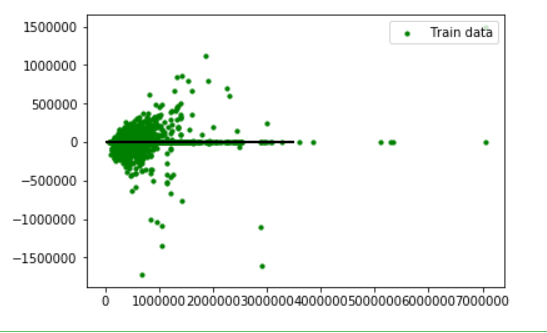
* plt.scatter(regressor.predict(input), regressor.predict(input) - outcome,

color = "green", s = 10, label = 'Train data')

plt.hlines(y = 0, xmin = 0, xmax = 3500000, linewidth = 2)

plt.legend(loc = 'upper right')

plt.show()



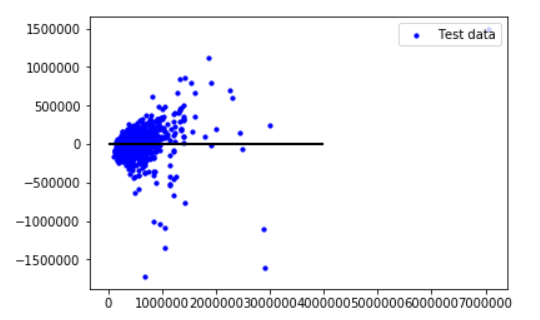
* plt.scatter(regressor.predict(x\_test), regressor.predict(x\_test) - y\_test,

color = "blue", s = 10, label = 'Test data')

plt.hlines(y = 0, xmin = 0, xmax = 4000000, linewidth = 2)

plt.legend(loc = 'upper right')

plt.show()



* plt.scatter(regressor.predict(input), regressor.predict(input) - outcome,

color = "green", s = 10, label = 'Train data')

***## plotting residual errors in test data***

plt.scatter(regressor.predict(x\_test), regressor.predict(x\_test) - y\_test,

color = "blue", s = 10, label = 'Test data')

***## plotting line for zero residual error***

plt.hlines(y = 0, xmin = 0, xmax = 3500000, linewidth = 2)

plt.legend(loc = 'upper right')

***## plot title***

plt.title("Residual errors")

***## function to show plot***

plt.show()

