**Predicting Boston Housing Prices using Decision Tree Regression and Adaptive Boosting Regression**

**The demonstrated Python code of predicting Boston Housing Prices using Decision Tree Regression is as below:**

# Import the necessary libraries

import numpy

import seaborn as sns

import matplotlib.pyplot as plot

import pandas

from sklearn import metrics

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

from sklearn.model\_selection import ShuffleSplit

from sklearn.model\_selection import KFold

from sklearn.metrics import accuracy\_score

from sklearn.tree import DecisionTreeRegressor

# Import the dataset

dataset = pandas.read\_csv('BostonHousing.csv')

#Explore the dataset

print(dataset.shape)

print(dataset.head(5))

# Differentiate attribute and target columns

Y = dataset['medv'].values

X = dataset[['rm','ptratio','lstat']].values

predicted\_y = []

expected\_y = []

#Applying K-fold cross validation

kf = KFold(n\_splits=4,shuffle=True)

for train\_index, test\_index in kf.split(X):

x\_train, x\_test = X[train\_index], X[test\_index]

y\_train, y\_test = Y[train\_index], Y[test\_index]

regressor = DecisionTreeRegressor()

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

predicted\_y.extend(regressor.predict(x\_test))

expected\_y.extend(y\_test)

df = pandas.DataFrame({'Actual': expected\_y, 'Predicted': predicted\_y})

print(df.head(20))

#showing bar chart

df1 = df.head(30)

df1.plot(kind='bar')

plot.show()

#showing plot

x\_ax = range(len(expected\_y))

plot.scatter(x\_ax, expected\_y, s=5, color="blue", label="original")

plot.plot(x\_ax, predicted\_y, lw=0.8, color="red", label="predicted")

plot.legend()

plot.show()

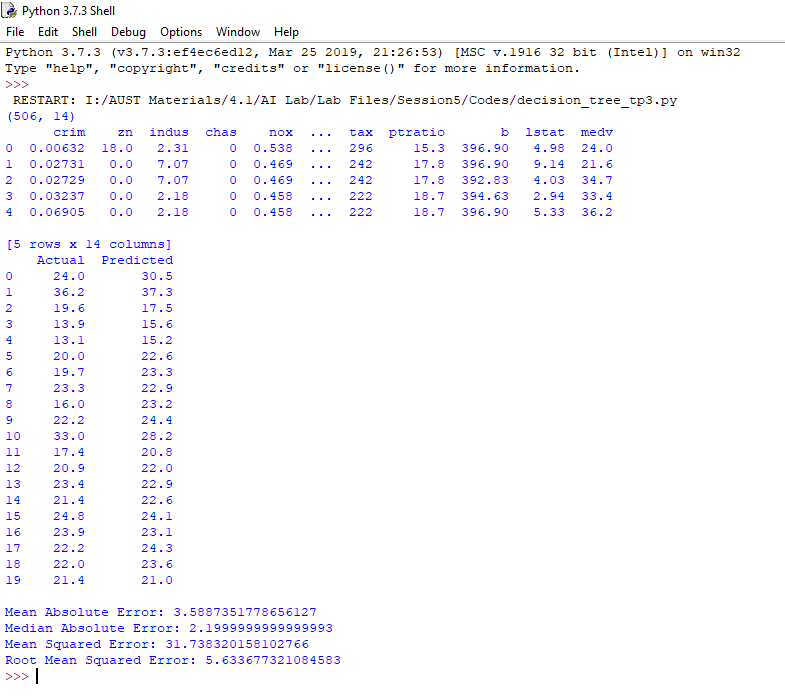
print('\nMean Absolute Error:', metrics.mean\_absolute\_error(expected\_y, predicted\_y))

print('Median Absolute Error:', metrics.median\_absolute\_error(expected\_y, predicted\_y))

print('Mean Squared Error:', metrics.mean\_squared\_error(expected\_y, predicted\_y) )

print('Root Mean Squared Error:', numpy.sqrt(metrics.mean\_squared\_error(expected\_y, predicted\_y)))

**Sample input and output is as below:**



**The demonstrated Python code of predicting Boston Housing Prices using Adaptive Boosting Regression is as below:**

# Import the necessary libraries

import numpy

import seaborn as sns

import matplotlib.pyplot as plot

import pandas

from sklearn import metrics

from sklearn.model\_selection import ShuffleSplit

from sklearn.model\_selection import KFold

from sklearn.ensemble import AdaBoostRegressor

from sklearn.datasets import make\_regression

# Import the dataset

dataset = pandas.read\_csv('BostonHousing.csv')

#Explore the dataset

print(dataset.shape)

print(dataset.head(5))

# Differentiate attribute and target columns

Y = dataset['medv'].values

X = dataset[['rm','ptratio','lstat']].values

predicted\_y = []

expected\_y = []

#Applying K-fold cross validation

kf = KFold(n\_splits=4,shuffle=True)

for train\_index, test\_index in kf.split(X):

x\_train, x\_test = X[train\_index], X[test\_index]

y\_train, y\_test = Y[train\_index], Y[test\_index]

ada\_reg = AdaBoostRegressor(n\_estimators=100)

ada\_reg.fit(x\_train, y\_train)

predicted\_y.extend(ada\_reg.predict(x\_test))

expected\_y.extend(y\_test)

df = pandas.DataFrame({'Actual': expected\_y, 'Predicted': predicted\_y})

print(df.head(20))

#showing bar chart

df1 = df.head(30)

df1.plot(kind='bar')

plot.show()

#showing plot

x\_ax = range(len(expected\_y))

plot.scatter(x\_ax, expected\_y, s=5, color="blue", label="original")

plot.plot(x\_ax, predicted\_y, lw=0.8, color="red", label="predicted")

plot.legend()

plot.show()

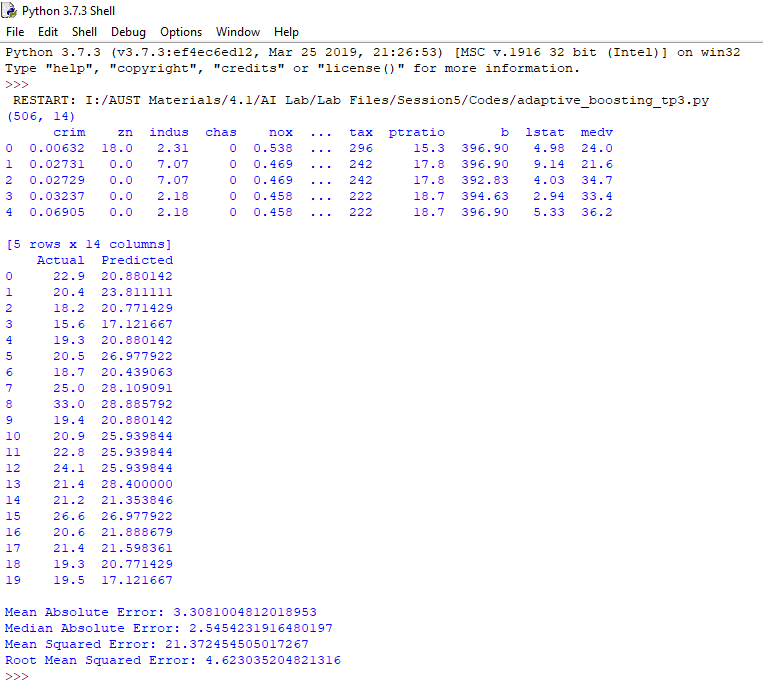
print('\nMean Absolute Error:', metrics.mean\_absolute\_error(expected\_y, predicted\_y))

print('Median Absolute Error:', metrics.median\_absolute\_error(expected\_y, predicted\_y))

print('Mean Squared Error:', metrics.mean\_squared\_error(expected\_y, predicted\_y) )

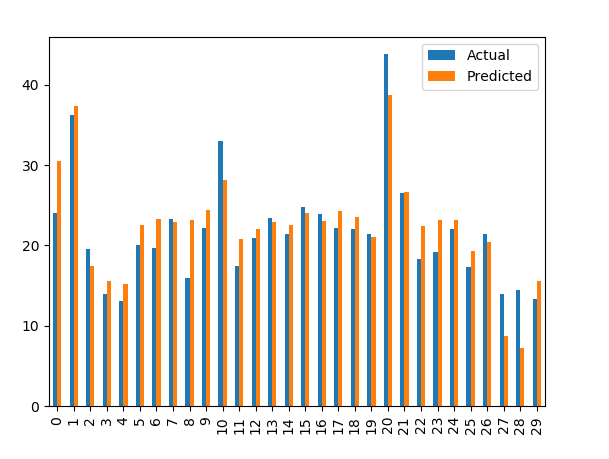
print('Root Mean Squared Error:', numpy.sqrt(metrics.mean\_squared\_error(expected\_y, predicted\_y)))

**Sample input and output is as below:**

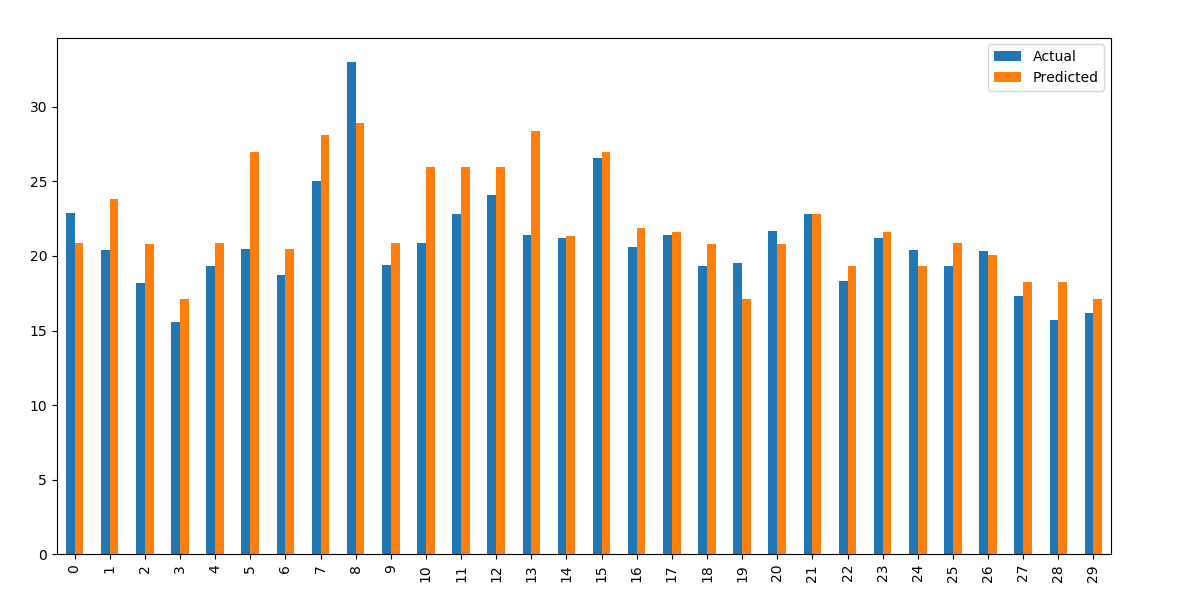


**Comparison between two approaches:**

1. Bar Chart

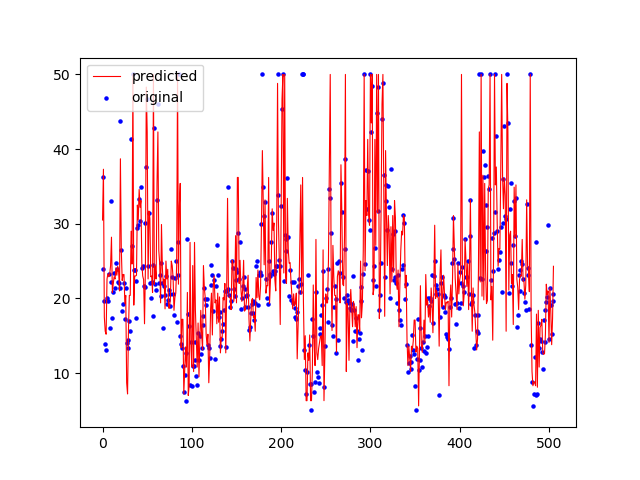


**Fig: Bar Chart for Decision Tree Regression**

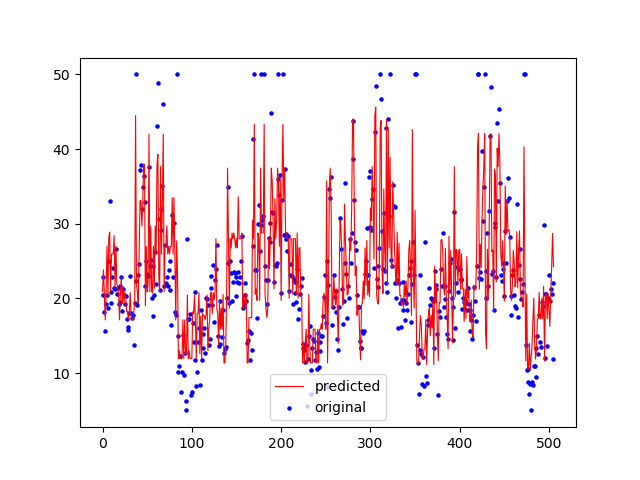


**Fig: Bar Chart for Adaptive Boosting Regression**

1. **Scatter Plot:**

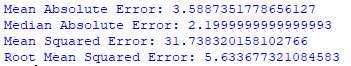


**Fig: Scatter Plot for Decision Tree Regression**

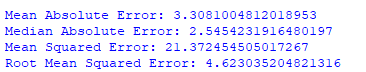


**Fig: Scatter Plot for Adaptive Boosting Regression**

1. **Performance Metric Scores:**
2. **Decision Tree Regression**



1. **Adaptive Boosting Regression**



By analyzing the above bar charts, scatter plots and performance metric score it seems that Adaptive Boosting Regression Model gives better performance in comparison to the decision tree regression model for predicting Boston Housing prices.