Height Prediction from Age using Decision Tree Regression

Import Dataset

In [19]:

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
```

Load Dataset from Local directory

In [20]:

```
from google.colab import files
uploaded = files.upload()
```

Choose File **No file selected**

Upload widget is only available when the cell has been executed in the current browser session. Please rerun this cell to enable.

```
Saving Dataset.csv to Dataset.csv
```

Load Dataset

In [29]:

```
dataset = pd.read_csv('dataset.csv')
```

Summarize Dataset

In [30]:

```
print(dataset.shape)
print(dataset.head(5))
```

```
(71, 2)
   Age  Height
0   10     138
1   11     138
2   12     138
3   13     139
4   14     139
```

Segregate Dataset into Input X & Output Y

In [31]:

```
X = dataset.iloc[:, :-1].values
X
```

Out[31]:

```
array([[10],
       [11],
       [12],
       [13],
       [14],
       [15],
       [16],
       [17],
       [18],
       [19],
```

```
       [20],
       [21],
       [22],
       [23],
       [24],
       [25],
       [26],
       [27],
       [28],
       [29],
       [30],
       [31],
       [32],
       [33],
       [34],
       [35],
       [36],
       [37],
       [38],
       [39],
       [40],
       [41],
       [42],
       [43],
       [44],
       [45],
       [46],
       [47],
       [48],
       [49],
       [50],
       [51],
       [52],
       [53],
       [54],
       [55],
       [56],
       [57],
       [58],
       [59],
       [60],
       [61],
       [62],
       [63],
       [64],
       [65],
       [66],
       [67],
       [68],
       [69],
       [70],
       [71],
       [72],
       [73],
       [74],
       [75],
       [76],
       [77],
       [78],
       [79],
       [80]])
```

In [32]:

```
Y = dataset.iloc[:, -1].values
Y
```

Out[32]:

```
array([138, 138, 138, 139, 139, 139, 140, 140, 140, 141, 141, 141, 142,
       142, 142, 143, 143, 143, 144, 144, 145, 145, 146, 146, 147, 147,
       148, 148, 149, 149, 150, 150, 151, 152, 153, 155, 156, 157, 158,
       159, 160, 161, 162, 163, 164, 164, 165, 166, 167, 168, 169, 170
```

```
171, 172, 173, 175, 178, 180, 183, 185, 188, 190, 193, 195, 199,
203, 204, 205, 206, 207, 208])
```

Splitting Dataset for Testing our Model

In [33]:

```
from sklearn.model_selection import train_test_split
x_train, x_test, y_train, y_test = train_test_split(X, Y, test_size = 0.20, random_state
= 0)
```

Training Dataset using DECISION TREE

In [34]:

```
from sklearn.tree import DecisionTreeRegressor
model = DecisionTreeRegressor()
model.fit(x_train, y_train)
```

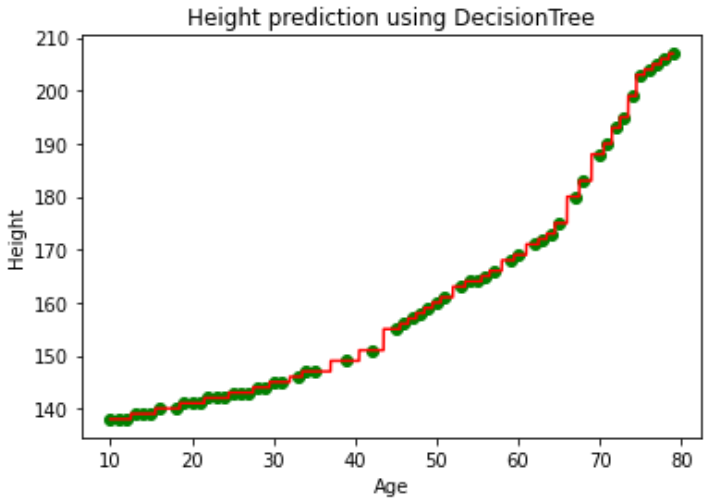
Out[34]:

```
DecisionTreeRegressor(ccp_alpha=0.0, criterion='mse', max_depth=None,
                      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='deprecated',
                      random_state=None, splitter='best')
```

Visualizing Graph

In [35]:

```
X_val = np.arange(min(x_train), max(x_train), 0.01)
X_val = X_val.reshape((len(X_val), 1))
plt.scatter(x_train, y_train, color = 'green')
plt.plot(X_val, model.predict(X_val), color = 'red')
plt.title('Height prediction using DecisionTree')
plt.xlabel('Age')
plt.ylabel('Height')
plt.figure()
plt.show()
```




```
<Figure size 432x288 with 0 Axes>
```

Prediction for all test data for validation

In [36]:

```
ypred = model.predict(x_test)

from sklearn.metrics import r2_score, mean_squared_error
```

```
mse = mean_squared_error(y_test, ypred)
rmse = np.sqrt(mse)
print("Root Mean Square Error: ", rmse)
r2score = r2_score(y_test, ypred)
print("R2Score", r2score*100)
```

```
Root Mean Square Error:  1.3416407864998738
R2Score 99.42815994578109
```