

Microsoft Excel interface showing a spreadsheet with columns A through W and rows 44 through 73. The data is organized into three columns: A, B, and C. The values in column A range from 52 to 80, and the values in column B range from 162 to 208. The values in column C are calculated as the sum of the values in columns A and B.

Row	A	B	C
44	52	162	214
45	53	163	216
46	54	164	218
47	55	164	219
48	56	165	221
49	57	166	223
50	58	167	225
51	59	168	227
52	60	169	229
53	61	170	231
54	62	171	233
55	63	172	235
56	64	173	237
57	65	175	240
58	66	178	244
59	67	180	247
60	68	183	251
61	69	185	254
62	70	188	258
63	71	190	261
64	72	193	265
65	73	195	268
66	74	199	273
67	75	203	278
68	76	204	280
69	77	205	282
70	78	206	284
71	79	207	286
72	80	208	288

The ribbon includes tabs for Clipboard, Font, Alignment, Number, Styles, Cells, and Editing. The active cell is B72, containing the value 208. The status bar at the bottom shows the active sheet is named "dataset".



```
[1] import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
```

Load Dataset from Local directory

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

Load Dataset

0s completed at 7:11 PM



```
[3] dataset = pd.read_csv('dataset.csv')
```

Summarize Dataset

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

Segregate Dataset into Input X & Output Y

```
X = dataset.iloc[:, 0:11].values
```

✓ 0s completed at 7:12 PM



```
[4] 2 12 138
    3 13 139
    4 14 139
```

Segregate Dataset into Input X & Output Y

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

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

Splitting Dataset for Testing our Model

0s completed at 7:12 PM



Splitting Dataset for Testing our Model

```
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

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



Visualizing Graph

```
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()
```

Prediction for all test data for validation

0s completed at 7:14 PM



+ Code + Text

RAM Disk Editing

Prediction for all test data for validation

```
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
```

0s completed at 7:21 PM

