

In [22]: *#Reference : <https://www.geeksforgeeks.org/python-decision-tree-regression-using-sklearn>*

```
# import numpy package for arrays and stuff
import numpy as np

# import matplotlib.pyplot for plotting our result
import matplotlib.pyplot as plt

# import pandas for importing csv files
import pandas as pd

# import dataset
# dataset = pd.read_csv('Data.csv')
# alternatively open up .csv file to read data

dataset = np.array(
[['Asset Flip', 100, 1000],
['Text Based', 500, 3000],
['Visual Novel', 1500, 5000],
['2D Pixel Art', 3500, 8000],
['2D Vector Art', 5000, 6500],
['Strategy', 6000, 7000],
['First Person Shooter', 8000, 15000],
['Simulator', 9500, 20000],
['Racing', 12000, 21000],
['RPG', 14000, 25000],
['Sandbox', 15500, 27000],
['Open-World', 16500, 30000],
['MMOFPS', 25000, 52000],
['MMORPG', 30000, 80000]
])

# print the dataset
print(dataset)
```

```
[['Asset Flip' '100' '1000']
['Text Based' '500' '3000']
['Visual Novel' '1500' '5000']
['2D Pixel Art' '3500' '8000']
['2D Vector Art' '5000' '6500']
['Strategy' '6000' '7000']
['First Person Shooter' '8000' '15000']
['Simulator' '9500' '20000']
['Racing' '12000' '21000']
['RPG' '14000' '25000']
['Sandbox' '15500' '27000']
['Open-World' '16500' '30000']
['MMOFPS' '25000' '52000']
['MMORPG' '30000' '80000']]
```

In [23]: *# select all rows by : and column 2*
by 2 to Y representing labels
X = dataset[:, 1:2].astype(int)
y = dataset[:, 2].astype(int)

```
# print y
print(y)
```

```
[ 1000  3000  5000  8000  6500  7000 15000 20000 21000 25000 27000 30000
 52000 80000]
```

```
In [24]: # 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(X, y)
```

Out[24]: DecisionTreeRegressor(random_state=0)

```
In [25]: # predicting a new value

# test the output by changing values, like 3750
y_pred = regressor.predict([[3750]])

# print the predicted price
print("Predicted price: % d\n"% y_pred)
```

Predicted price: 8000

```
In [26]: # arange for creating a range of values
# from min value of X to max value of X
# with a difference of 0.01 between two
# consecutive values
X_grid = np.arange(min(X), max(X), 0.01)

# reshape for reshaping the data into
# a len(X_grid)*1 array, i.e. to make
# a column out of the X_grid values
X_grid = X_grid.reshape((len(X_grid), 1))

# scatter plot for original data
plt.scatter(X, y, color = 'red')

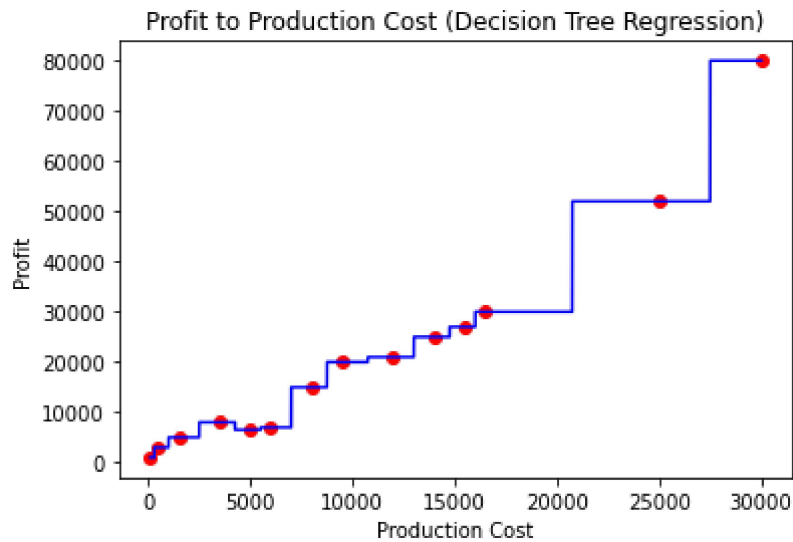
# plot predicted data
plt.plot(X_grid, regressor.predict(X_grid), color = 'blue')

# specify title
plt.title('Profit to Production Cost (Decision Tree Regression)')

# specify X axis label
plt.xlabel('Production Cost')

# specify Y axis label
plt.ylabel('Profit')

# show the plot
plt.show()
```



```
In [27]: import six
import sys
sys.modules['sklearn.externals.six'] = six

from sklearn.externals.six import StringIO
from IPython.display import Image
from sklearn.tree import export_graphviz
import pydotplus
dot_data = StringIO()
export_graphviz(regressor, out_file=dot_data,
                filled=True, rounded=True,
                special_characters=True, feature_names = ['Production Cost'])
graph = pydotplus.graph_from_dot_data(dot_data.getvalue())
graph.write_png('diabetes.png')
Image(graph.create_png())
```

Out[27]:

