Decision_tree_regression

October 23, 2021

1 Decision Tree Regression

1.1 Importing the libraries

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

1.2 Importing the dataset

[2]:
```python
dataset = pd.read_csv('Position_Salaries.csv')
X = dataset.iloc[:, 1:-1].values
y = dataset.iloc[:, -1].values
```

1.3 Training the Decision Tree Regression model on the whole dataset

[3]:
```python
from sklearn.tree import DecisionTreeRegressor
regressor = DecisionTreeRegressor(random_state = 0)
regressor.fit(X, y)
```

[3]:
```
DecisionTreeRegressor(random_state=0)
```

1.4 Predicting a new result

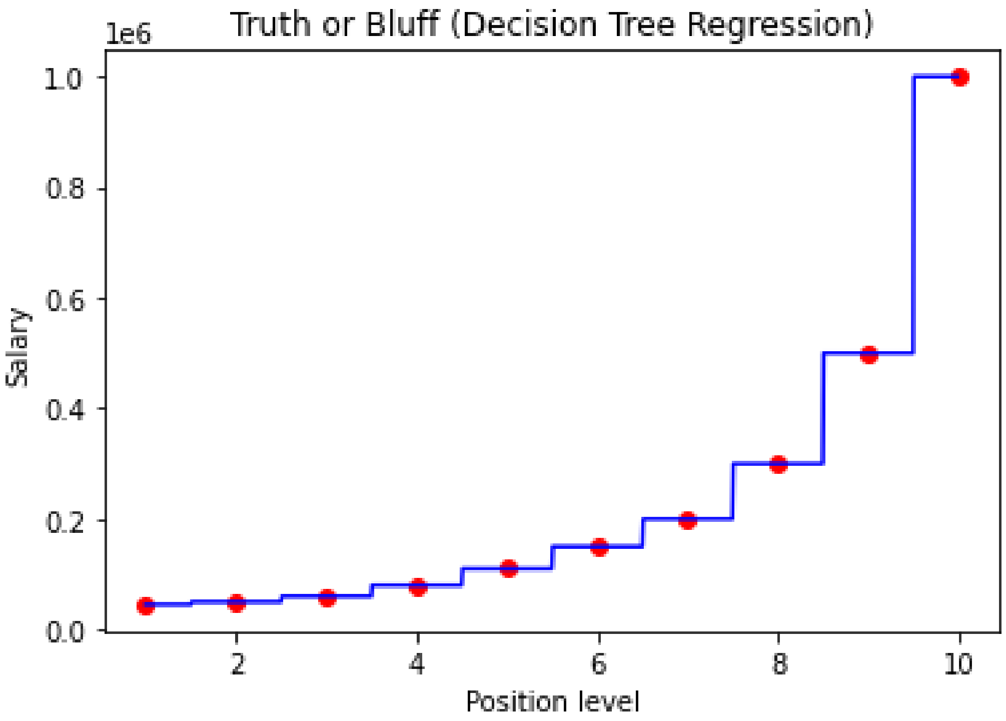
[4]:
```python
regressor.predict([[6.5]])
```

[4]:
```
array([150000.])
```

1.5 Visualising the Decision Tree Regression results (higher resolution)

[5]:

```
X_grid = np.arange(min(X), max(X), 0.01)
X_grid = X_grid.reshape((len(X_grid), 1))
plt.scatter(X, y, color = 'red')
plt.plot(X_grid, regressor.predict(X_grid), color = 'blue')
plt.title('Truth or Bluff (Decision Tree Regression)')
plt.xlabel('Position level')
plt.ylabel('Salary')
plt.show()
```



[6]:

```
from sklearn.tree import export_text,plot_tree
text_representation = export_text(regressor)
print(text_representation)
```

```
|--- feature_0 <= 8.50
|   |--- feature_0 <= 6.50
|   |   |--- feature_0 <= 4.50
|   |   |   |--- feature_0 <= 3.50
|   |   |   |   |--- feature_0 <= 2.50
|   |   |   |   |   |--- feature_0 <= 1.50
|   |   |   |   |   |   |--- value: [45000.00]
|   |   |   |   |   |--- feature_0 >  1.50
|   |   |   |   |   |   |--- value: [50000.00]
|   |   |   |   |--- feature_0 >  2.50
|   |   |   |   |   |--- value: [60000.00]
|   |   |   |--- feature_0 >  3.50
|   |   |   |   |--- value: [80000.00]
|   |   |--- feature_0 >  4.50
|   |   |   |--- feature_0 <= 5.50
|   |   |   |   |--- value: [110000.00]
|   |   |   |--- feature_0 >  5.50
|   |   |   |   |--- value: [150000.00]
|   |--- feature_0 >  6.50
|   |   |--- feature_0 <= 7.50
|   |   |   |--- value: [200000.00]
|   |   |--- feature_0 >  7.50
|   |   |   |--- value: [300000.00]
|--- feature_0 >  8.50
|   |--- feature_0 <= 9.50
|   |   |--- value: [500000.00]
|   |--- feature_0 >  9.50
|   |   |--- value: [1000000.00]
```

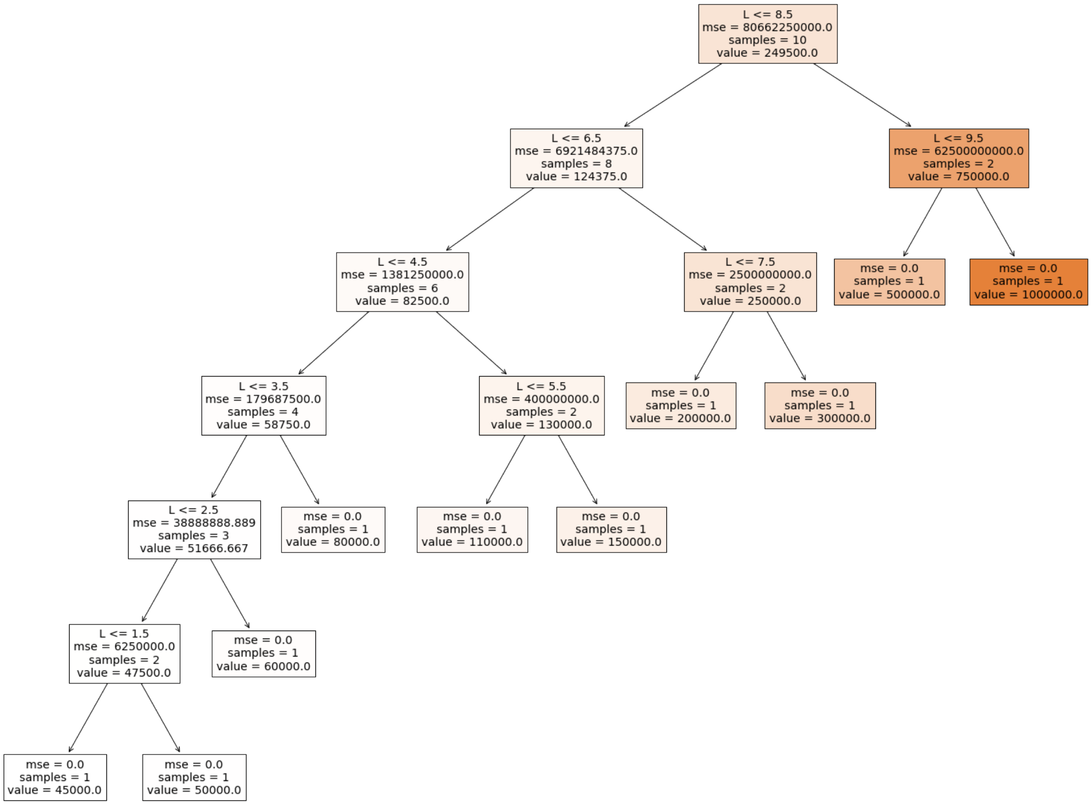
[9]:

```
print(dataset)
```

```
            Position  Level   Salary
0   Business Analyst      1    45000
1  Junior Consultant      2    50000
2  Senior Consultant      3    60000
3            Manager      4    80000
4    Country Manager      5   110000
5     Region Manager      6   150000
6            Partner      7   200000
7     Senior Partner      8   300000
8            C-level      9   500000
9                CEO     10  1000000
```

[13]:

```
fig = plt.figure(figsize=(25,20))
_ = plot_tree(regressor, feature_names='Level', filled=True)
```


L <= 8.5
mse = 80662250000.0
samples = 10
value = 249500.0
L <= 6.5
mse = 6921484375.0
samples = 8
value = 124375.0
L <= 9.5
mse = 62500000000.0
samples = 2
value = 750000.0
L <= 4.5
mse = 1381250000.0
samples = 6
value = 82500.0
L <= 7.5
mse = 2500000000.0
samples = 2
value = 250000.0
mse = 0.0
samples = 1
value = 500000.0
mse = 0.0
samples = 1
value = 1000000.0
L <= 3.5
mse = 179687500.0
samples = 4
value = 58750.0
L <= 5.5
mse = 400000000.0
samples = 2
value = 130000.0
mse = 0.0
samples = 1
value = 200000.0
mse = 0.0
samples = 1
value = 300000.0
L <= 2.5
mse = 38888888.889
samples = 3
value = 51666.667
mse = 0.0
samples = 1
value = 80000.0
mse = 0.0
samples = 1
value = 110000.0
mse = 0.0
samples = 1
value = 150000.0
L <= 1.5
mse = 6250000.0
samples = 2
value = 47500.0
mse = 0.0
samples = 1
value = 60000.0
mse = 0.0
samples = 1
value = 45000.0
mse = 0.0
samples = 1
value = 50000.0