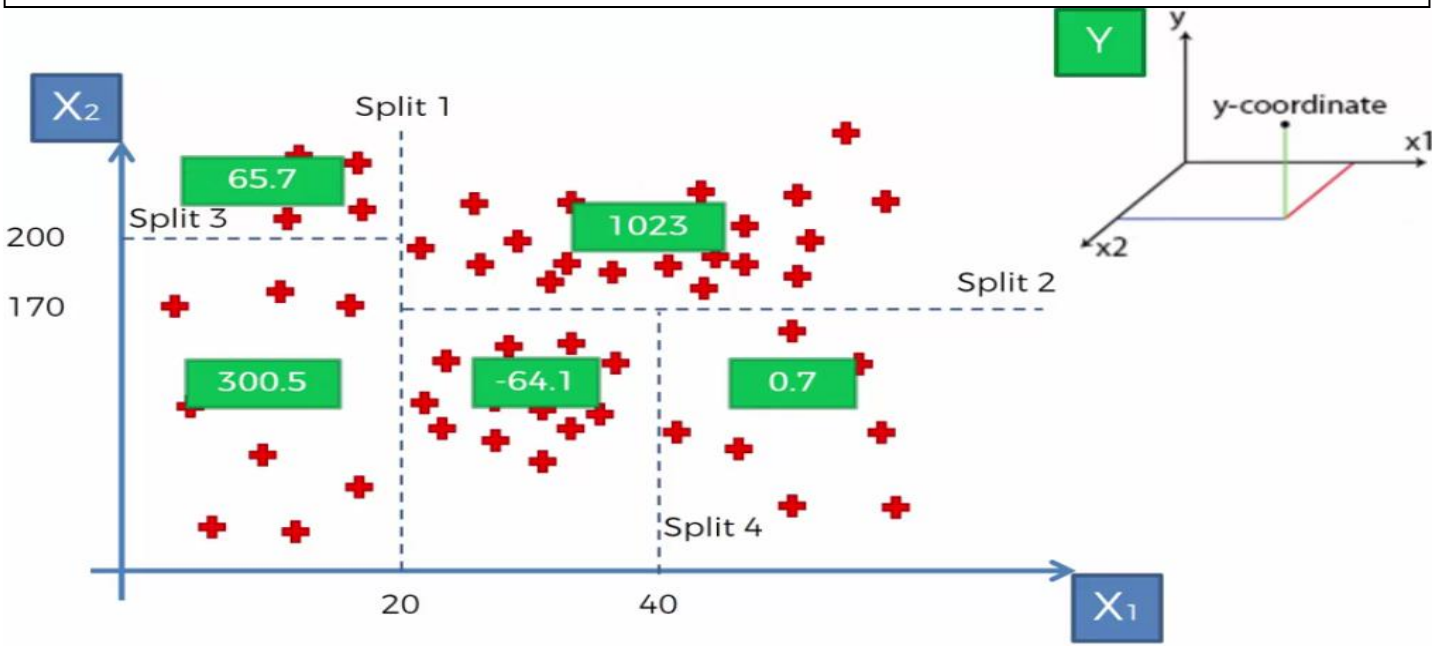
✧ Decision Tree Regression

✧ 方法簡介

決策樹有：
分類樹分析，依變數 y 可能為離散類別 eg. 蘭花品種
回歸樹分析，依變數 y 可能是連續型變數 eg. 薪資
結合以上兩種合稱：CART(Classification And Regression Trees)

✧ 建置步驟

所有 Training Set 樣本會一起在根節點(無抽樣的動作)
可用像單因子變異數分析，找到變異量最大的變項(資訊獲利的概念)作為分割點
(ID3、C4.5 、C5.0、CHAID 及 CART 是決策樹演算法的代表)
如果判斷結果的正確率或涵蓋率不滿足條件(自訂)，則再依最大變異量條件長出分支，簡單來說就是這棵樹會一直長大，直到符合某個我們可以接受條件，就會停止
(條件&這棵樹要怎麼長我都可以透過參數來訂定)



每個分支的終結點都會得到一個群內依變數 y 的平均值，若一新資料要作預測，就會依照該資料落點位置的平均數作為預測結果

✧ DecisionTreeRegressor 說明

```
from sklearn.tree import DecisionTreeRegressor
regressor = DecisionTreeRegressor(random_state = 0)
regressor.fit(X, y)
```

max_depth : int or None, optional (default=None)
決策樹的深度，"None"是盡可能開展枝葉或是每個節點內樣本數都達最小分枝樣本數
min_samples_split : int, float, optional (default=2)
最小分支樣本數(最少要有多少樣本數才能進行分支)
min_samples_leaf : int, float, optional (default=1)
最小節點樣本數(單一分支(葉)中的最小樣本數)

random_state : int, RandomState instance or None, optional (default=None)

seed 的概念

min_impurity_split : float,

停止分支的門檻值

presort : bool, optional (default=False)

是否要預排序

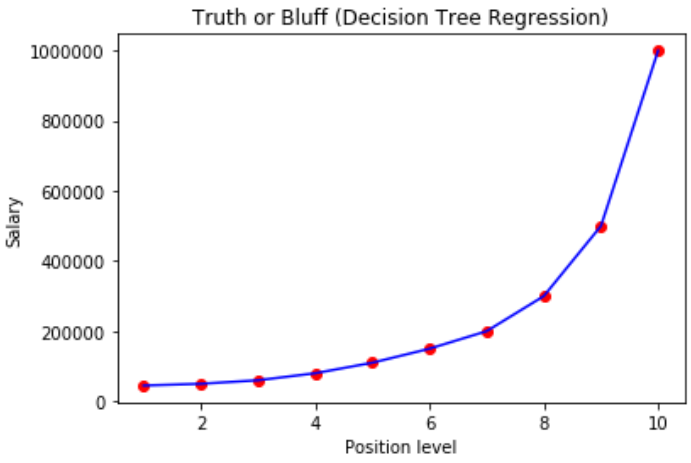
✧ 預測&畫圖

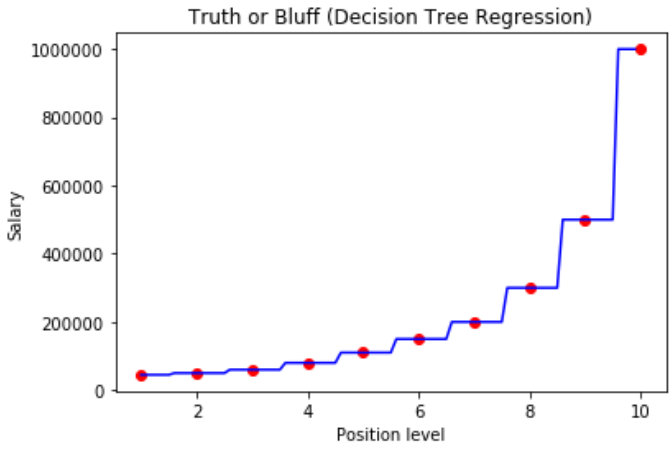
```
y_pred = regressor.predict(6.5)
```

預測結果為 150000

```
plt.scatter(X, y, color = 'red')
plt.plot(X, regressor.predict(X), color = 'blue')
plt.title('Truth or Bluff (Decision Tree Regression)')
plt.xlabel('Position level')
plt.ylabel('Salary')
plt.show()
```

```
X_grid = np.arange(min(X), max(X), 0.1)
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()
```





說明：剛剛提到決策樹執行的結果在每個分支內的預測值都會一樣，區間內的配適線應該要是水平的，所以左圖的圓滑曲線就不太合理，這時候就要使用上次介紹到的 arange 來進行等距切割，隨著距離給定的越小，配適線的階梯形狀會越明顯。