✧ 資料&情境說明：

沿用之前的資料集：

Index	Position	Level	Salary
0	Business Analyst	1	45000
1	Junior Consultant	2	50000
2	Senior Consultant	3	60000
3	Manager	4	80000

情境：人資處根據求職者前公司的薪資級距資料集透過模型來驗證求職者所說的工作 20 年以上、年薪 160K 是否為真
→但是這邊的概念比較像是假設求職者說的是真的，然後我們來用這幾個模型來預測看看哪個模型預測的準

✧ 方法介紹–三種不同的模型(non-linear model)

- Support Vector Regression (SVR)
- Decision Tree Regression
- Random Forest Regression

✧ Support Vector Regression (SVR)

✧ **方法簡介**：找到一個平面能有效區分所有觀測值，且讓所有觀測值離平面的距離最大

✧ **資料載入**：資料載入的步驟和之前都一樣

```
import numpy as np
import matplotlib.pyplot as plt
import pandas as pd
dataset = pd.read_csv('Position_Salaries.csv')
X = dataset.iloc[:, 1:2].values
y = dataset.iloc[:, 2].values
```

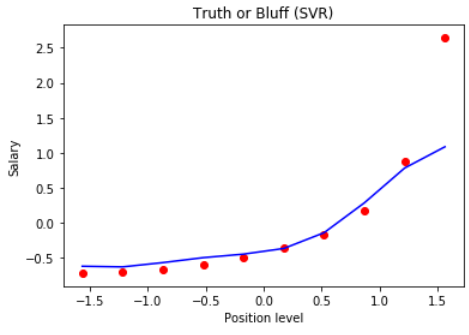
✧ **建立預測模型**

```
from sklearn.svm import SVR
regressor = SVR(kernel = 'rbf')
regressor.fit(X,y)
```

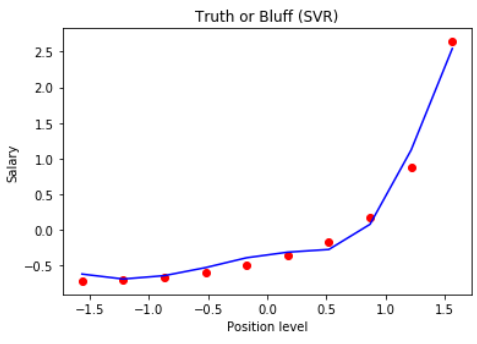
✧ **SVR 說明**

C=1.0,

懲罰係數，就是對誤差的容忍度，C 越大就越不能容忍誤差，反之，太大或太小都不好，會失真



C=1



C=10

cache_size=200, 暫存空間大小，單位 MB

coef0=0.0, rbf 不需要調整，預設 0，只有在使用 poly、sigmoid 才要調

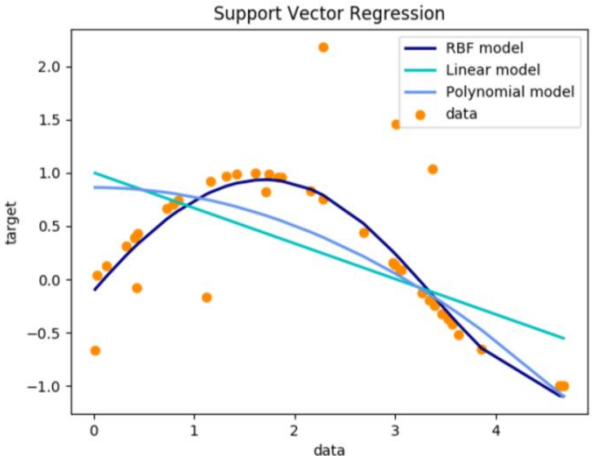
degree=3, rbf 不需要調整，預設 3，只有在使用 poly 才要調

epsilon=0.1,

gamma='auto', 內核係數

kernel='rbf', 核函數，有線性(linear)、多項(poly)、sigmoid、rbf 等

作者提到我們現在要處理的狀況是非線性的，rbf 是大多數人會使用且表現也比較好，所以選它



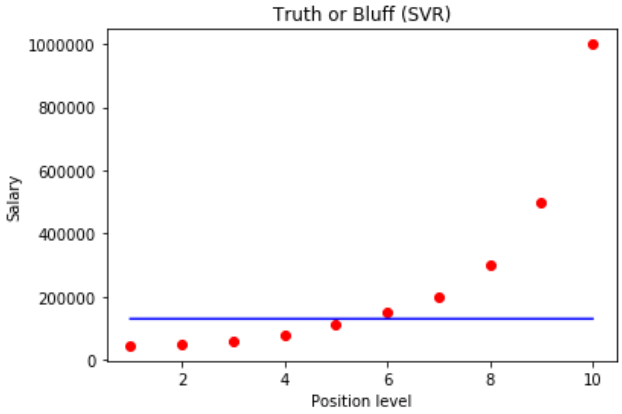
max_iter=-1,

shrinking=True,

tol=0.001,

verbose=False

✧ **未標準化跑出來的圖**



未標準化之前的圖，配適線是水平的，預測值為 130K，顯然有問題

✧ **進行標準化**

```
from sklearn.preprocessing import StandardScaler
sc_X = StandardScaler()
sc_y = StandardScaler()
X = sc_X.fit_transform(X)
y = sc_y.fit_transform(y)
```

使用 SVR 這個方法的時候，記得先做標準化這個動作，**讓 X y 的變動量不會互相影響**

在標準化的過程中會出現 warning，但不用在意因為它是在說 Xy 的值會從整數變成小數

✧ **再次進行預測**

(1)直接跑 y_pred = regressor.predict(6.5)，得到的預測值為 0.01158103

(2)跑 y_pred = regressor.predict(sc_X.transform(np.array([[6.5]]))，預測值為-0.27861589

(3)跑 y_pred = sc_y.inverse_transform(regressor.predict(sc_X.transform(np.array([[6.5]])))),

預測值為 **170370.0204065(正確結果)**

說明：SRV Model 用到的 X y 在剛剛已經進行過標準化，所以 6.5 也要做相同的動作就是標準化，但要注意的是要 transform 的話值必須是 2D array(矩陣)，所以要先用 np.array 處理過後再用 2 個中括號括起來才會是矩陣的形式，最後為了要取得 6.5 所對應的原始薪資，所以要做一個 inverse 的轉換