

Lab Regression 2 (22/4/2022)

ລະຫັດນັກສຶກສາ :205Q001019

ຊື່ ແລະ ນາມສະກຸນ:ທ້າວ ນຸຊົ່ວ ເຮີ 3CW1

ຈົ່ງຕອບຄໍາຖາມຕໍ່ນີ້ໃຫ້ສໍາເລັດດ້ວຍການນໍາໃຊ້ຄໍາສັ່ງຂອງ Python:

1. Multiple Linear Regression

1.1. ຈາກຊຸດຂໍ້ມູນ (Data Set) **50_Startups.csv**, ຈົ່ງບອກຈຳນວນຖັນ , ແຖວ (shape) ແລະ ເພີ່ຂໍ້ມູນໃສ່ຕາຕະລາງໃຫ້ສໍາເລັດ :

ຖັນ	5
ແຖວ	50

	R&D Spend	Administration	Marketing Spend	State	Profit
0	165349.20	136897.80	471784.10	New York	192261.83
1	162597.70	151377.59	443898.53	California	191792.06
2	153441.51	101145.55	407934.54	Florida	191050.39

1.2. ຈົ່ງກຳນົດຕົວປ່ຽນເອກະລາດ (Independent Variables X) ໃຫ້ເປັນ R&D Spend, Administration, Marketing Spend ແລະ State. ກຳນົດຕົວປ່ຽນຕາມ (Dependent Variables y) ໃຫ້ເປັນ Profit.

```
dataset = pd.read_csv('50_Startups.csv')
X = dataset.iloc[:, :-1].values
y = dataset.iloc[:, -1].values
```

✓ 0.4s

1.3. ຈົ່ງທຳການຕຽມຂຸດຂໍ້ມູນ (Preprocessing) ດ້ວຍການເຮັດ OneHotEncoder ຂອງຕົວປ່ຽນອິດສະຫຼະ X.

```
from sklearn.compose import ColumnTransformer
from sklearn.preprocessing import OneHotEncoder
ct = ColumnTransformer(transformers=[('encoder', OneHotEncoder(), [3])], remainder='passthrough')
X = np.array(ct.fit_transform(X))
```

1.4 ຫຼັງຈາກຕຽມຂຸດຂໍ້ມູນສາເລັດ, ຈົ່ງຂຽນຄຳສັ່ງເພື່ອແຍກຂຸດຂໍ້ມູນເປັນສອງພາກສ່ວນຄື: ຂຸດຮຽນ 85 ແລະ ຂຸດທົດສອ 15 ?

```
from sklearn.model_selection import train_test_split
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size = 0.15, random_state = 0)
```

✓ 0.1s

1.5. ຈົ່ງສ້າງໂມແດວ Linear Regression ແລະ ທຳການປະມວນຜົນ (fit) ຂຸດຂໍ້ມູນຮຽນຈາກຂໍ້ 1.4

```
from sklearn.linear_model import LinearRegression
regressor = LinearRegression()
regressor.fit(X_train, y_train)
```

✓ 0.2s

1.6. ຈົ່ງຄິດໄລ່ຄ່າ intercept_ ແລະ coef_ ຂອງໂມແດວຫຼັງການປະມວນຜົນ (fit).

```
regressor.intercept_
```

✓ 0.3s

42190.78496234339

```
regressor.coef_
```

✓ 0.4s

array([2.12734825e+02, -1.14601240e+03, 9.33277580e+02, 7.73051855e-01,
3.50926156e-02, 3.60827279e-02])

1.7. ຈົ່ງຄິດໄລ່ຄ່າ R-squared ຂອງໂມເດວຫຼັງການປະມວນຜົນ (fit) ຊຸດຂໍ້ມູນຮຽນຈາກ ຂໍ້ 1.4

```
regressor.score(X_train, y_train) #R squared
```

✓ 0.4s

0.9522700739455511

1.8. ຈົ່ງທົດສອບໂມເດວຂອງຊຸດຂໍ້ມູນ 15 ຈາກຂໍ້ 1.4.

```
y_pred = regressor.predict(X_test)
np.set_printoptions(precision=2)
print(np.concatenate((y_pred.reshape(len(y_pred),1), y_test.reshape(len(y_test),1)),1))
```

✓ 0.7s

1.9. ຈົ່ງຄິດໄລ່ຄ່າ R-squared ຂອງໂມເດວຫຼັງການທົດສອບຊຸດຂໍ້ມູນທົດສອບຈາກຂໍ້ 1.8.

```
regressor.score(X_test, y_test) #R squared
```

✓ 0.2s

0.9190315130435046

2. Polynomial Regression

2.1 ຈາກຊຸດຂໍ້ມູນ (Data Set) Position_salaries.csv, ຈົ່ງບອກຈຳນວນຖັນ, ແຖວ (shape) ແລະ ເພີ່ມຂໍ້ມູນໃສ່ຕາຕະລາງໃຫ້ສໍາເລັດ:

	Position	Level	Salary
0	Business Analyst	1	45000
1	Junior Consultant	2	50000
2	Senior Consultant	3	60000

ແຖວ	10
ຖັນ	3

2.2 ຈົ່ງກຳນົດ Level ໃຫ້ເປັນຕົວປ່ຽນເອກະລາດ (Independent Variables X). ກຳນົດ Salary ໃຫ້ເປັນຕົວປ່ຽນຕາມ (Dependent Variables y)

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

✓ 0.5s

2.3 ຈົ່ງປ່ຽນຕົວປ່ຽນເອກະລາດໃຫ້ເປັນ Polynomial Features ໃຫ້ degree = 3 ແລະທຳການ (fit) ໂມເດວ Linear Regression ດ້ວຍ .

```
from sklearn.preprocessing import PolynomialFeatures
poly_reg = PolynomialFeatures(degree = 3)
X_poly = poly_reg.fit_transform(X)
lin_reg_2 = LinearRegression()
lin_reg_2.fit(X_poly, y)
```

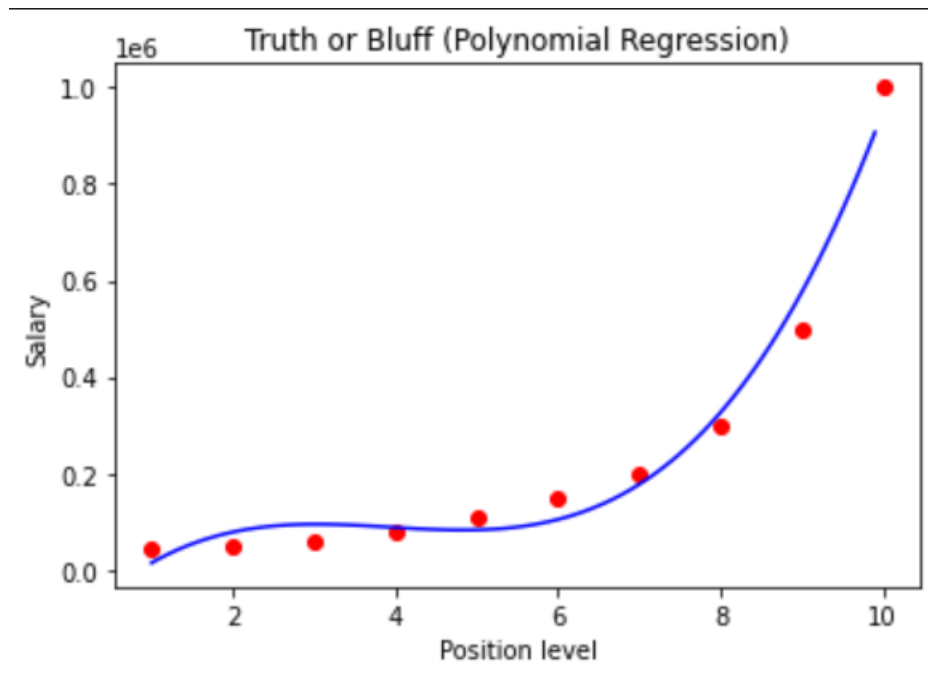
✓ 0.2s

2.4 ຈົ່ງສະແດງຂໍ້ມູນ (X, y) ດ້ວຍ scatter Graph ບົນພື້ນຖານຂຸດຄຳສັ່ງ matplotlib

ການສະແດງຜົນຂອງ Polynomial Regression (ສຳລັບຄວາມລະອຽດທີ່ສູງຂຶ້ນແລະເສັ້ນໂຄ້ງທີ່ລຽບກວ່າ)

```
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, lin_reg_2.predict(poly_reg.fit_transform(X_grid)), color = 'blue')
plt.title('Truth or Bluff (Polynomial Regression)')
plt.xlabel('Position level')
plt.ylabel('Salary')
plt.show()
```

✓ 0.1s



2.5 ຈົ່ງພະຍາກອນຄາດເດົາເງິນເດືອນ (Salary) ຂອງພະນກງານທີ່ມີຕຳແໜ່ງລະດັບ (Level) 7.5?

ຄາດຄະເນຜົນໄດ້ຮັບໃຫມ່ດ້ວຍການ Polynomial Regression

```
lin_reg_2.predict(poly_reg.fit_transform([[7.5]]))
```

✓ 0.5s

array([240915.35547786])

2.6 ຈົ່ງຄິດໄລ່ R squared (ຄວາມຖືກຕ້ອງໃນການຄາດເດົາ)

Polynomial Regression

```
lin_reg_2.score(X_poly,y)
```

✓ 0.2s

0.9812097727913367