

Quiz Regression

python code

```
import numpy as np
from sklearn.linear_model import LinearRegression
from sklearn.metrics import mean_squared_error, r2_score
from sklearn.preprocessing import PolynomialFeatures
import matplotlib.pyplot as plt
import pandas as pd
plt.style.use("seaborn")

df = pd.read_csv("data/data_quiz.csv")
x_train = df[['X']]
y_train = df['Y']

model = LinearRegression()
model.fit(x_train, y_train)
b = model.coef_
a = model.intercept_
y_pred = model.predict(x_train)
r2 = r2_score(y_train, y_pred)
mse = mean_squared_error(y_train, y_pred)
print("Simple linear")
print("(a)\t\t:", a)
print("(b)\t\t:", b)
print("R_squared\t: %.2f"%(r2))
print("MSE\t\t: %.2f\n"%(mse))
plt.title("Simple linear ")
plt.scatter(x_train, y_train, color="green", label="Actual data")
plt.plot(x_train, y_pred, color="blue", label="Y = %.2fX + %.2f"%(b, a))
plt.legend()
plt.show()

s = ["Y = %.2fX + %.2f, (R2 = %.2f)",
      "Y = %.2fX + %.2fX^2 + %.2f, (R2 = %.2f)",
      "Y = %.2fX + %.2fX^2 + %.2fX^3 + %.2f, (R2 = %.2f)",
      "Y = %.2fX + %.2fX^2 + %.2fX^3 + %.2fX^4 + %.2f, (R2 = %.2f)",
      "Y = %.2fX + %.2fX^2 + %.2fX^3 + %.2fX^4 + %.2fX^5 + %.2f, (R2 = %.2f)"]

def calculator(d):
    poly = PolynomialFeatures(degree=d)
    x_poly = poly.fit_transform(x_train)
    model_poly = LinearRegression()
    model_poly.fit(x_poly, y_train)

    b = model_poly.coef_
    a = model_poly.intercept_
    y_pred = model_poly.predict(x_train)
    r2 = r2_score(y_train, y_pred)
    mse = mean_squared_error(y_train, y_pred)
    print("degree", d)
    print("(a)\t\t:", a)
    print("(b)\t\t:", b)
    print("R_squared\t: %.2f"%(r2))
    print("MSE\t\t: %.2f\n"%(mse))

    x = model_poly.predict(poly.fit_transform(x_train))
    plt.title("Polynomial, degree %d"%(d))
    plt.scatter(x_train, y_train, color="green", label="Actual data")
    if(d==1):
        plt.plot(x_train, x, color="blue", label=s[d-1%(b[1], a, r2)])
    elif(d==2):
        plt.plot(x_train, x, color="blue", label=s[d-1%(b[1], b[2], a, r2)])
    elif(d==3):
        plt.plot(x_train, x, color="blue", label=s[d-1%(b[1], b[2], b[3], a, r2)])
    elif(d==4):
        plt.plot(x_train, x, color="blue", label=s[d-1%(b[1], b[2], b[3], b[4], a, r2)])
    elif(d==5):
        plt.plot(x_train, x, color="blue", label=s[d-1%(b[1], b[2], b[3], b[4], b[5], a, r2)])
    plt.legend()
    plt.show()
    return x

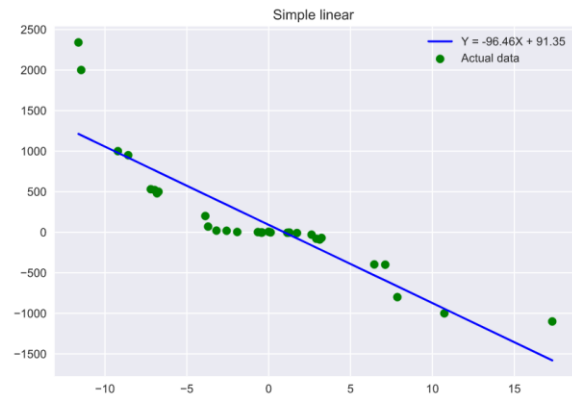
for i in range(5):
    calculator(i+1)

new_x = np.array([20, -20, 25, -25]).reshape(-1, 1)
poly = PolynomialFeatures(degree=4)
x_poly = poly.fit_transform(x_train)
model_poly = LinearRegression()
model_poly.fit(x_poly, y_train)
new_pred = model_poly.predict(poly.fit_transform(new_x))
print("predict reponses for new_x:\n", new_pred)
```

กราฟรูปแบบต่างๆ ที่ใช้ในการวิเคราะห์

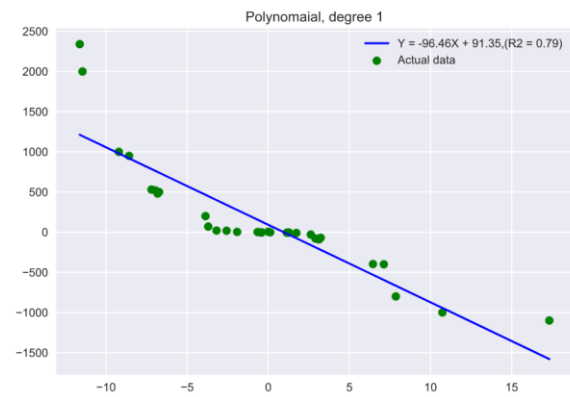
Simple linear

(a) : 91.35160879723867
(b) : [-96.46104138]
R_squared : 0.79
MSE : 105416.91



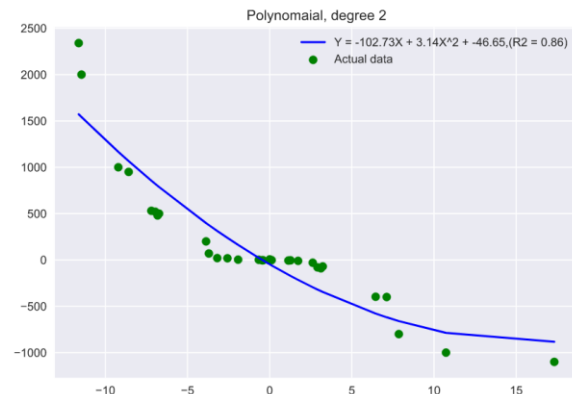
degree 1

(a) : 91.35160879723867
(b) : [0, -96.46104138]
R_squared : 0.79
MSE : 105416.91



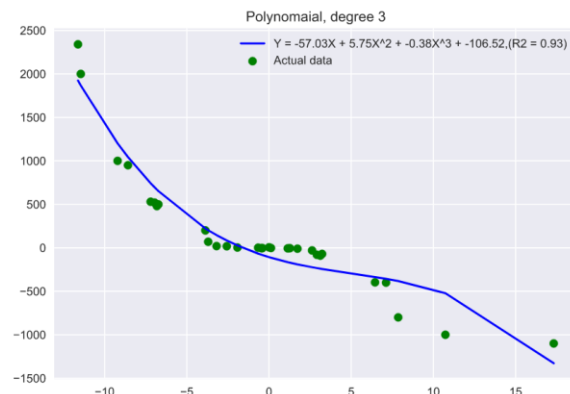
degree 2

(a) : -46.64978649116093
(b) : [0, -102.72911929, 3.14470147]
R_squared : 0.86
MSE : 69105.40



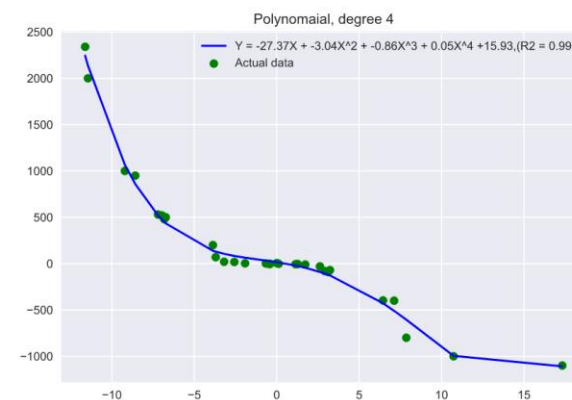
degree 3

(a) : -106.51864604541296
(b) : [0, -57.03226761, 5.7525466, -0.37703167]
R_squared : 0.93
MSE : 37169.91



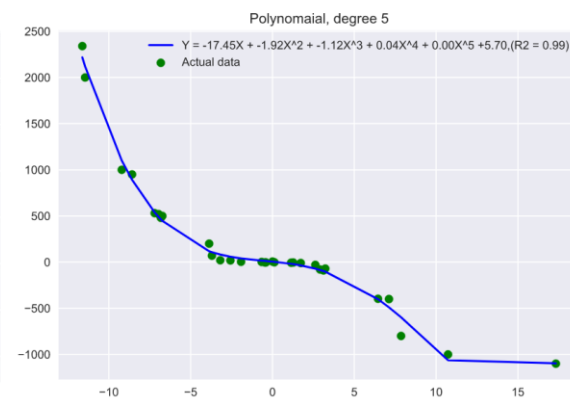
degree 4

(a) : 15.933025038910017
(b) : [0, -27.3661294, -3.03891859, -0.86490249, 0.05284339]
R_squared : 0.99
MSE : 4503.18



degree 5

(a) : 5.703486962592592
(b) : [0.00000000e+00, -1.74480732e+01, -1.92213058e+00, -1.11725461e+00, 4.18246219e-02, 1.16597903e-03]
R_squared : 0.99
MSE : 3992.33



อธิบายการวิเคราะห์เหตุผลและการเลือกใช้โมเดลรูปแบบนั้น

จากกราฟที่ได้ทำการ train ด้วยข้อมูลเดียวกันแต่ใช้คนละโมเดล

1. Simple linear พบว่า

R2 : 0.79 นั้นหมายความว่า ข้อมูลจะเข้าใกล้กับเส้นที่คาดการณ์อยู่ระดับความถูกต้องที่ 79%

MSE : 105416.91 เป็นค่าเฉลี่ยของข้อมูลจริงและค่าคาดการณ์ซึ่งเฉลี่ยออกมาแล้วมีค่าที่สูงมาก

2. polynomial, degree 1 เหมือนกับ Simple linear

3. polynomial, degree 2

R2 : 0.86 นั้นหมายความว่า ข้อมูลจะเข้าใกล้กับเส้นที่คาดการณ์อยู่ระดับความถูกต้องที่ 86%

MSE : 69105.40 เป็นค่าเฉลี่ยของข้อมูลจริงและค่าคาดการณ์ซึ่งเฉลี่ยออกมาแล้วมีค่าที่สูงมาก

4. polynomial, degree 3

R2 : 0.93 นั้นหมายความว่า ข้อมูลจะเข้าใกล้กับเส้นที่คาดการณ์อยู่ระดับความถูกต้องที่ 93%

MSE : 37169.91 เป็นค่าเฉลี่ยของข้อมูลจริงและค่าคาดการณ์ซึ่งเฉลี่ยออกมาแล้วมีค่าที่สูงมาก

5. polynomial, degree 4

R2 : 0.99 นั้นหมายความว่า ข้อมูลจะเข้าใกล้กับเส้นที่คาดการณ์อยู่ระดับความถูกต้องที่ 99%

MSE : 4503.18 เป็นค่าเฉลี่ยของข้อมูลจริงและค่าคาดการณ์ซึ่งเฉลี่ยออกมาแล้วมีค่าที่ดีเมื่อเทียบกับ model 1-4

6. polynomial, degree 5

R2 : 0.99 นั้นหมายความว่า ข้อมูลจะเข้าใกล้กับเส้นที่คาดการณ์อยู่ระดับความถูกต้องที่ 99%

MSE : 3992.33 เป็นค่าเฉลี่ยของข้อมูลจริงและค่าคาดการณ์ซึ่งเฉลี่ยออกมาแล้วมีค่าที่ดีเมื่อเทียบกับ model 1-5

ดังนั้น มี polynomial, degree 4 กับ . polynomial, degree 5 โดยค่า MSE ของ degree 4 ต่างกับ degree 5 เล็กน้อยจริง
เลือก model polynomial, degree 4 เพราะว่า degree น้อยกว่า polynomial, degree 5 ซึ่งค่า R2 เท่ากันและค่า MSE
ต่างกันเล็กน้อย

output ค่า Y ที่ได้จากการพยากรณ์ new_X โดยใช้โมเดลที่เหมาะสมที่สุด

จากข้อมูล new_X = [20,-20,25,-25]

เมื่อแทนในสมการ ของ Polynomial, degree 4

$$Y = -27.37X + -3.04X^2 + -0.86X^3 + 0.05X^4 + 15.93$$

จะได้ค่าเท่ากับ

predict reponses for new_x:

[-211.23415523 14721.85089312 4560.30438021 32956.81372589]