Course2_Module2

Training and Test Splits & Polynomial Regression

1. Training and Test Splits

- Ý tưởng: chia dữ liệu thành **2 phần**.
  - Training set: dùng để huấn luyện mô hình.
  - Test set: dùng để đánh giá hiệu suất mô hình.
- Mục đích:
  - Giúp chọn mô hình có khả năng **tổng quát hoá** tốt hơn.
  - Tránh tình trạng **overfitting** (mô hình chỉ học thuộc dữ liệu huấn luyện).

Training vs Test Error

- Khi mô hình phức tạp hơn:
  - Training error thường giảm.
  - Nhưng Test error có thể tăng (dấu hiệu overfit).

Syntax (ví dụ Python sklearn)

```
from sklearn.model_selection import train_test_split

X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_stat
e=42)
```

---

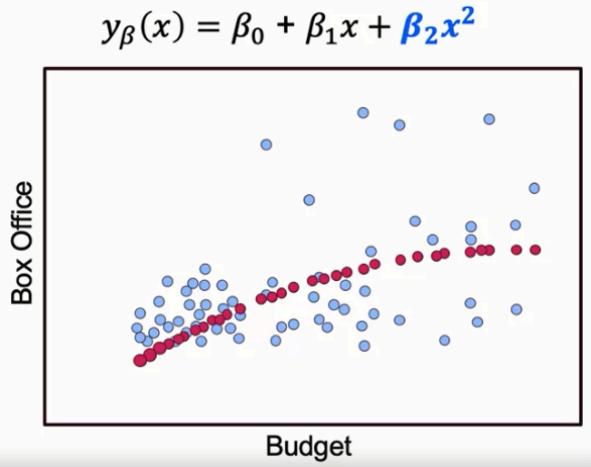
2. Polynomial Regression

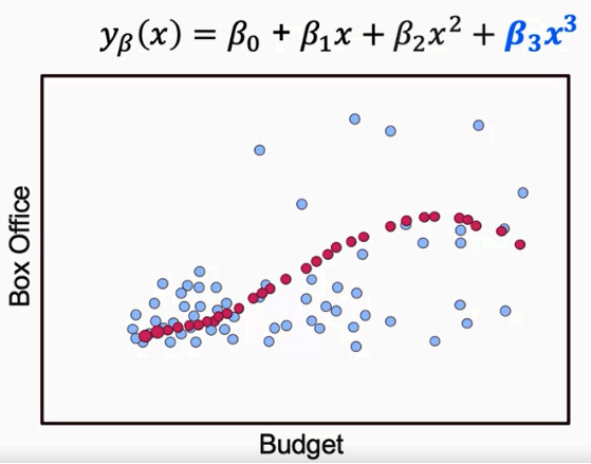
Learning Goals

- Mở rộng Linear Regression bằng Polynomial Features.
- Nắm cách dùng để bắt các quan hệ phi tuyến (nonlinear effects).
- Biết thêm các mô hình khác có thể áp dụng cho regression và classification.

Polynomial Features

- Thêm bậc cao của feature:
  - Ví dụ: thay vì chỉ x, thêm x^2, x^3.
- Công thức:
- Lưu ý: vẫn là **linear regression** nhưng trên không gian feature mở rộng.





Khi nào dùng?

- Khi mối quan hệ giữa biến độc lập và biến phụ thuộc không tuyến tính.
- Phải kiểm tra mối quan hệ trước để chọn bậc đa thức hợp lý.

Prediction vs Interpretation

- Prediction: tập trung vào độ chính xác dự đoán.
- Interpretation: khó hơn vì nhiều bậc, dễ mất ý nghĩa.
- Lưu ý: mô hình càng mạnh thì thường càng khó giải thích.

Các mô hình mở rộng khác

Ngoài Polynomial Regression, có nhiều mô hình để giải quyết cả regression và classification:

- Logistic Regression.
- K-Nearest Neighbors (KNN).
- Decision Trees.
- Support Vector Machines (SVM).
- Random Forests.
- Ensemble Methods.
- Deep Learning Approaches.

Syntax (Python sklearn)

```
from sklearn.preprocessing import PolynomialFeatures
from sklearn.linear_model import LinearRegression

poly = PolynomialFeatures(degree=2)
X_poly = poly.fit_transform(X)

LR = LinearRegression()
LR.fit(X_poly, y)
y_pred = LR.predict(X_poly)
```