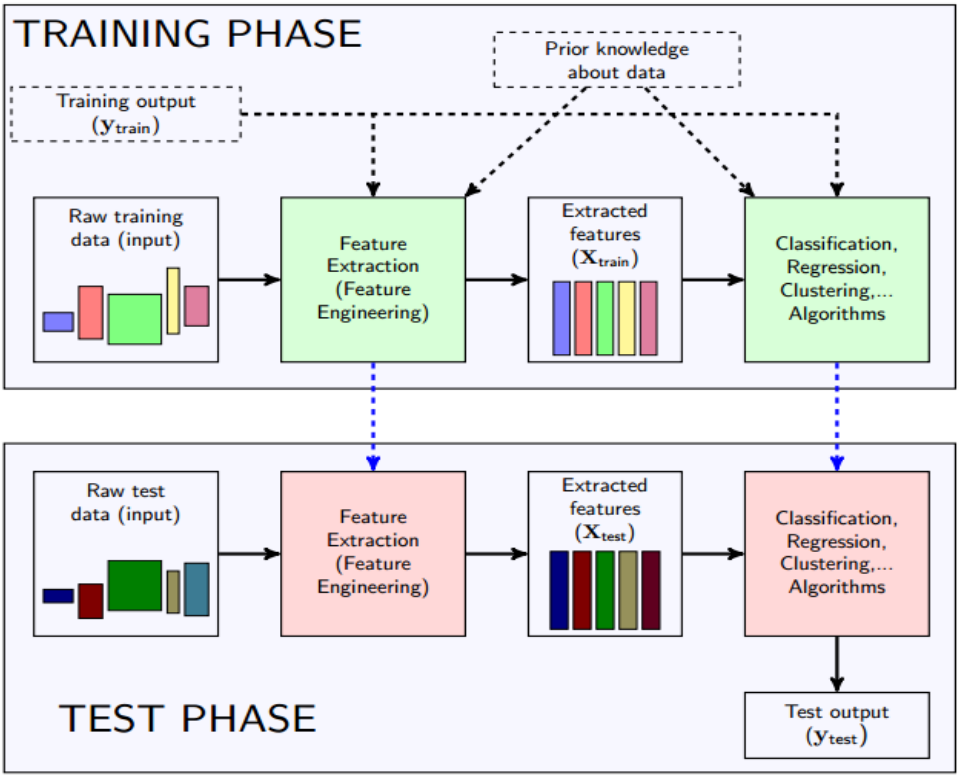
Đặc trưng Feature

Feature engineering là tác vụ cần thiết trong quá trình xây dựng predictive model => rất tốn thời gian và công sức đòi hỏi phải có kiến thức.

Lý do: ta không thể đưa dữ liệu thô (raw data) trực tiếp vào bất kỳ mô hình Machine Learning nào => nên mục tiêu cần làm là rút trích các đặc trưng (features) từ dữ liệu thô ban đầu này.

Feature Engineering là quá trình xây dựng các features cần thiết cho việc training các mô hình Machine Learning dựa trên các dữ liệu thô truyền vào.

Mô hình chung cho bài toán Machine Learning



Thuật toán Linear Regression

Hồi qui tuyến tính là một thuật toán hồi qui mà đầu ra là một hàm số tuyến tính của đầu vào.

$$y = b_1x_1 + b_2x_2 + b_3x_3 + \cdots + b_kx_k + e$$

Biến mục đích — Hệ số hồi quy — Biến số giải thích — Sai số

Ví dụ: sử dụng thuật toán Linear Regression để dự đoán cân nặng dựa vào chiều cao

Dữ liệu:

Chiều cao (cm)	Cân nặng (kg)	Chiều cao (cm)	Cân nặng (kg)
147	49	168	60
150	50	170	72
153	51	173	63
155	52	175	64
158	54	178	66
160	56	180	67
163	58	183	68
165	59		

Note: Sinh viên sửa lại dữ liệu người cao 170cm nặng 62 kg

1. Import thư viện: numpy, matplotlib

```
import numpy as np
import matplotlib.pyplot as plt
from sklearn import datasets, linear_model
```

2. Đưa các đặc trưng vào từ bảng dữ liệu ở trên:

```
: #height (cm) - features
features = np.array([[147, 150, 153, 158, 163, 165, 168, 170, 173, 175, 178, 180, 183]]).T
print(features)
```

```
[[147]
 [150]
 [153]
 [158]
 [163]
 [165]
 [168]
 [170]
 [173]
 [175]
 [178]
 [180]
 [183]]
```

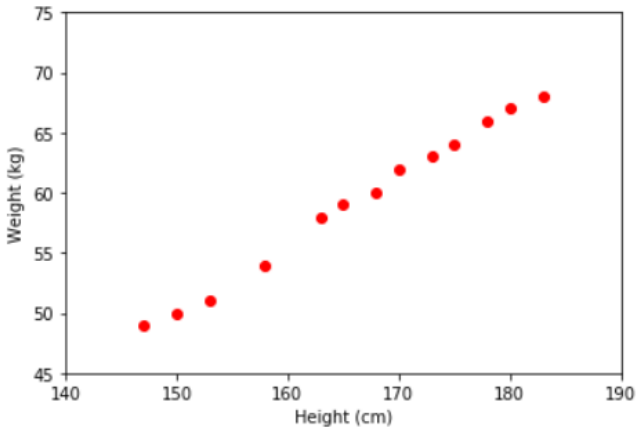
3. Đưa target (đích) là cân nặng vào:

```
: #weight (kg) - targets
targets = np.array([[ 49, 50, 51,  54, 58, 59, 60, 62, 63, 64, 66, 67, 68]]).T
print(targets)
```

```
[[49]
 [50]
 [51]
 [54]
 [58]
 [59]
 [60]
 [62]
 [63]
 [64]
 [66]
 [67]
 [68]]
```

4. Vẽ đồ thị chiều cao-cân nặng dùng matplotlib:

```
# Visualize data - show features and targets relationships
plt.plot(features, targets, 'ro')
plt.axis([140, 190, 45, 75])
plt.xlabel('Height (cm)')
plt.ylabel('Weight (kg)')
plt.show()
```



5. Xây dựng mô hình (model) dùng thuật toán hồi qui tuyến tính:

```
#Fit the model by Linear Regression
classifier = linear_model.LinearRegression()
classifier.fit(features, targets)
```

```
LinearRegression(copy_X=True, fit_intercept=True, n_jobs=None,
         normalize=False)
```

6. Tính các hệ số của mô hình và in ra:

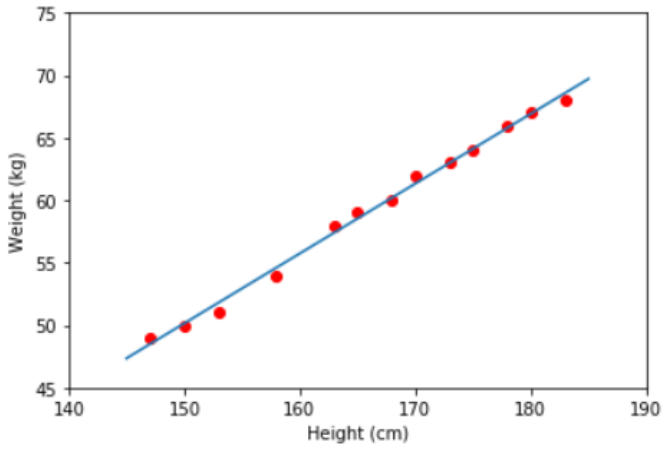
```
#y = w0 + w1 * x1
#target = coef_ * feature + intercept_
w = classifier.coef_[0]
b = classifier.intercept_
print("coef_ w = ", w)
print("intercept_ b = ", b)
```

```
coef_ w =  [0.55920496]
intercept_ b =  [-33.73541021]
```

7. Vẽ đường thẳng (màu Blue) đi qua các điểm dữ liệu:

```
#Fiting line
x0 = np.linspace(145, 185, 2, endpoint=True)
y0 = (w * x0) + b

# Visualize data
plt.plot(features, targets, 'ro')
plt.plot(x0, y0)
plt.axis([140, 190, 45, 75])
plt.xlabel('Height (cm)')
plt.ylabel('Weight (kg)')
plt.show()
```



8. Sử dụng mô hình được xây dựng bởi thuật toán hồi qui tuyến tính để dự đoán:

```
: #Prediction 150
print("With 150 cm height, you will weight", classifier.predict([[150]])[0][0], "kg.")
```

```
With 150 cm height, you will weight 50.145334085142366 kg.
```

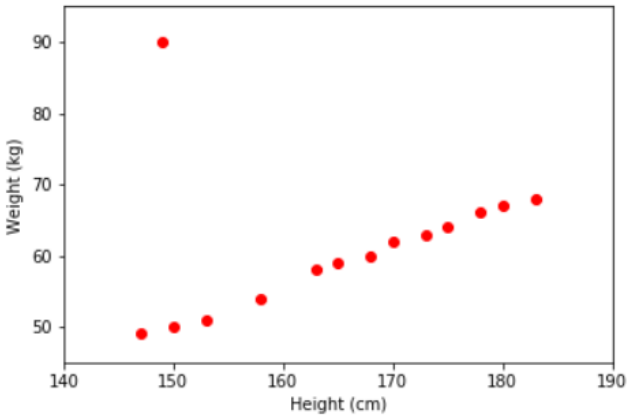
```
: #Prediction 155
print("With 155 cm height, you will weight", classifier.predict([[155]])[0][0], "kg.")
```

```
With 155 cm height, you will weight 52.941358894840704 kg.
```

Sinh viên đánh giá mô hình: so sánh với bảng chiều cao cân nặng ở trên và cho nhận xét về kết quả dự đoán của mô hình với chiều cao là 150 và 155.

9. Khi cân nặng là một số rất lớn so với bảng trên:

```
#Be careful with the noise
#height (cm) - features
features_noise = np.array([[147, 150, 153, 158, 163, 165, 168, 170, 173, 175, 178, 180, 183, 149]]).T
#weight (kg) - targets
targets_noise = np.array([[ 49, 50, 51,  54, 58, 59, 60, 62, 63, 64, 66, 67, 68, 90]]).T
# Visualize data - show features and targets relationships
plt.plot(features_noise, targets_noise, 'ro')
plt.axis([140, 190, 45, 95])
plt.xlabel('Height (cm)')
plt.ylabel('Weight (kg)')
plt.show()
```



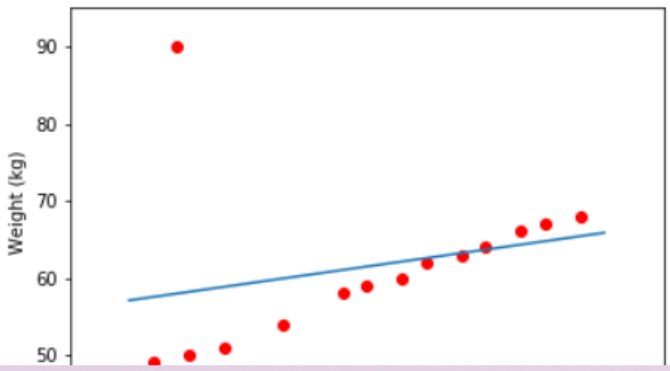
```
: #Fit the model by Linear Regression
classifier1 = linear_model.LinearRegression()
classifier1.fit(features_noise, targets_noise)

#y = w*x + b
#target = coef_ * feature + intercept_
w = classifier1.coef_[0]
b = classifier1.intercept_
print("coef_ w = ", w)
print("intercept_ b = ", b)

#Fiting line
x0 = np.linspace(145, 185, 2, endpoint=True)
y0 = (w * x0) + b

# Visualize data
plt.plot(features_noise, targets_noise, 'ro')
plt.plot(x0, y0)
plt.axis([140, 190, 45, 95])
plt.xlabel('Height (cm)')
plt.ylabel('Weight (kg)')
plt.show()
```

```
coef_ w =  [0.21901073]
intercept_ b =  [25.33194279]
```



Sinh viên quan sát đường kẻ màu xanh (blue) và so sánh với kết quả ở mục 7.

Bài tập 1: Sinh viên xây dựng mô hình dựa vào thuật toán hồi qui tuyến tính với tập dữ liệu về số giờ học và kết quả như sau:

	A	B
1	Hours_Studied	Test_Grade
2	2	57
3	3	66
4	4	73
5	5	76
6	6	79
7	7	81
8	8	90
9	9	96
10	10	100
11		

.