



Bài giảng môn học:

Học Máy (Machine Learning)

CHƯƠNG 3: HỌC CÓ GIÁM SÁT (tiếp) (Supervised Learning)

Giảng viên: Đặng Văn Nam

Email: dangvannam@hmg.edu.vn

Nội dung chương 3 – phần 1

1. Giới thiệu Thuật toán KNN
2. Xây dựng mô hình KNN trên dữ liệu Titanic
3. Đánh giá độ chính xác của mô hình phân lớp
 1. Tổng số mẫu dự đoán đúng
 2. Độ chính xác % (Accuracy)
 3. Ma trận nhầm lẫn (Confusion matrix)



1. Thuật toán KNN (K – Nearest Neighbors)

Giới thiệu thuật toán KNN

K-Nearest neighbors(k-NN) là một trong những thuật toán đơn giản nhất và phổ biến trong học máy. Một số tên gọi khác:

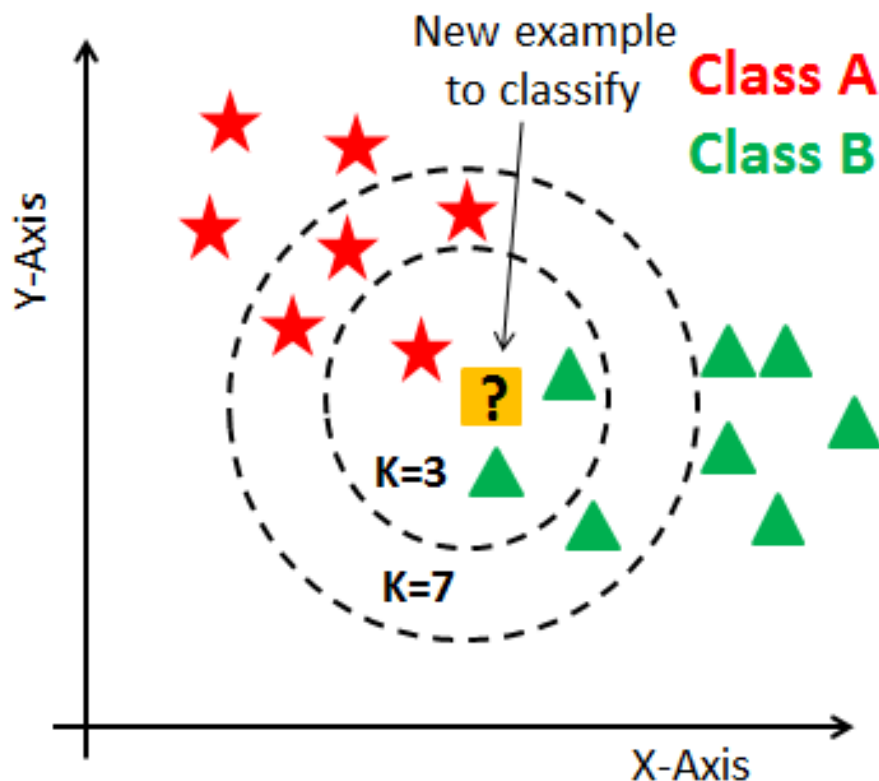
- **Instance-based learning**
- **Lazy learning**
- **Memory-based learning**

Ý tưởng của thuật toán này là nó không học một điều gì từ tập dữ liệu học (nên KNN được xếp vào loại lazy learning), mọi tính toán được thực hiện khi cần dự đoán nhãn của dữ liệu mới.

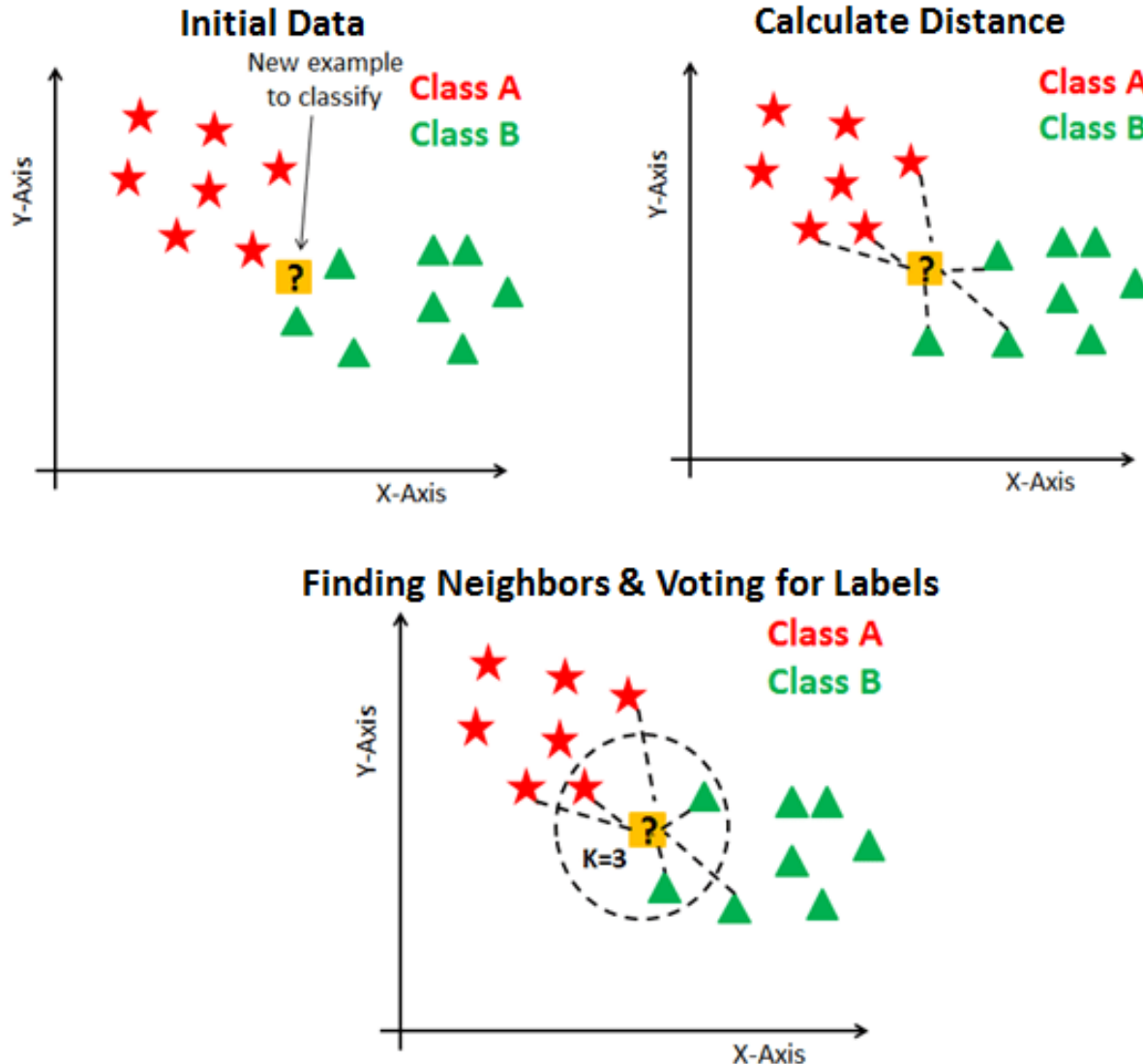


Giới thiệu thuật toán KNN

Bản chất, KNN là thuật toán đi tìm đầu ra của một điểm dữ liệu mới bằng cách chỉ dựa trên thông tin của K điểm dữ liệu trong tập huấn luyện gần nó nhất (K - lân cận)



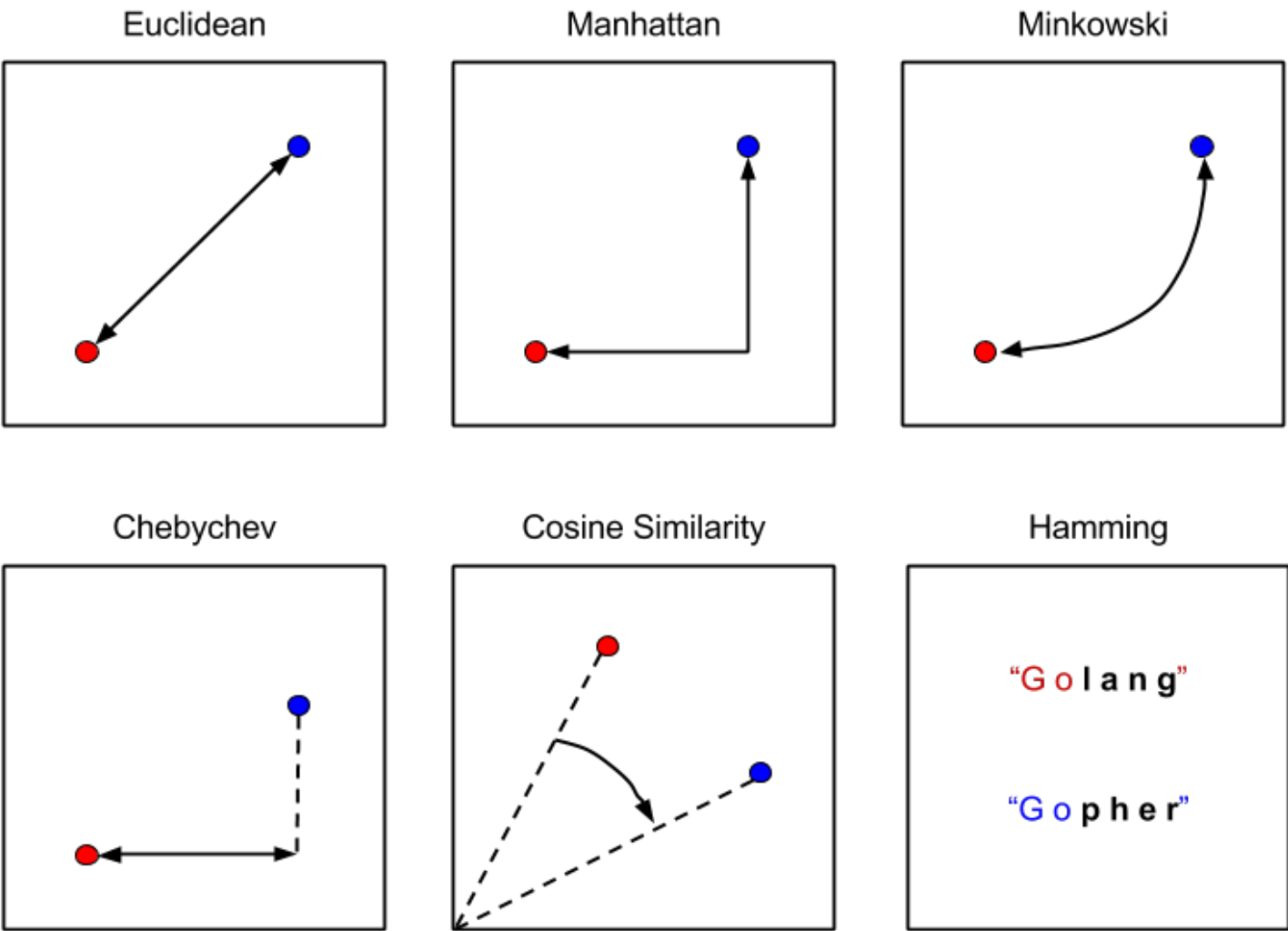
Giới thiệu thuật toán KNN



Những hàng xóm nào sẽ được sử dụng cho việc dự đoán?

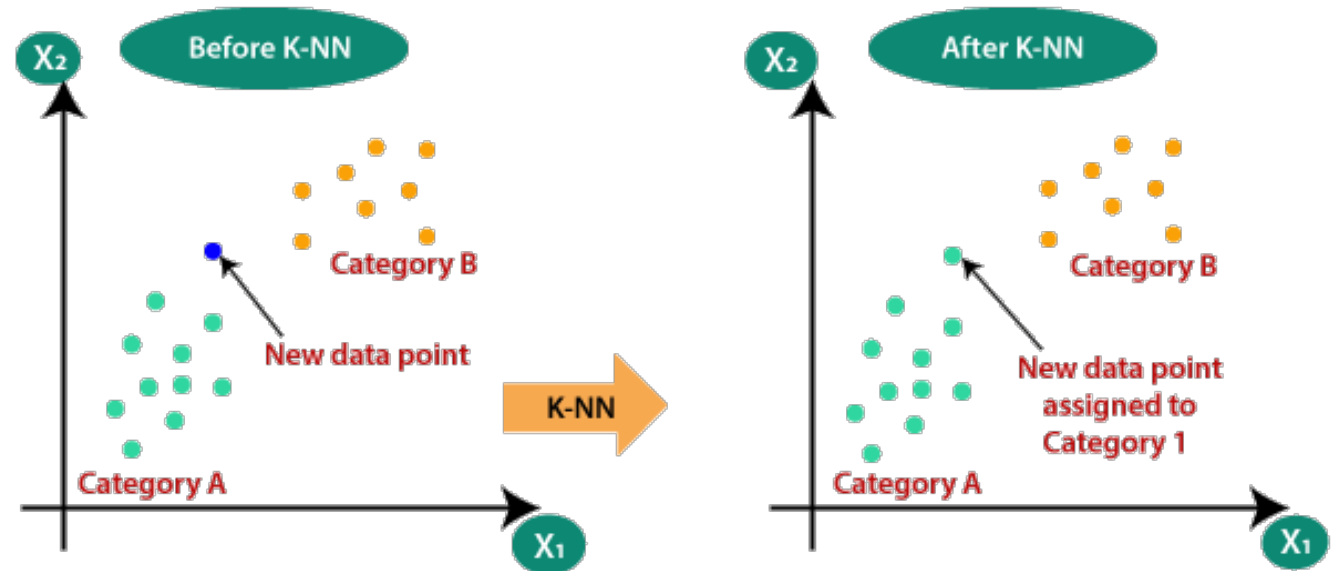
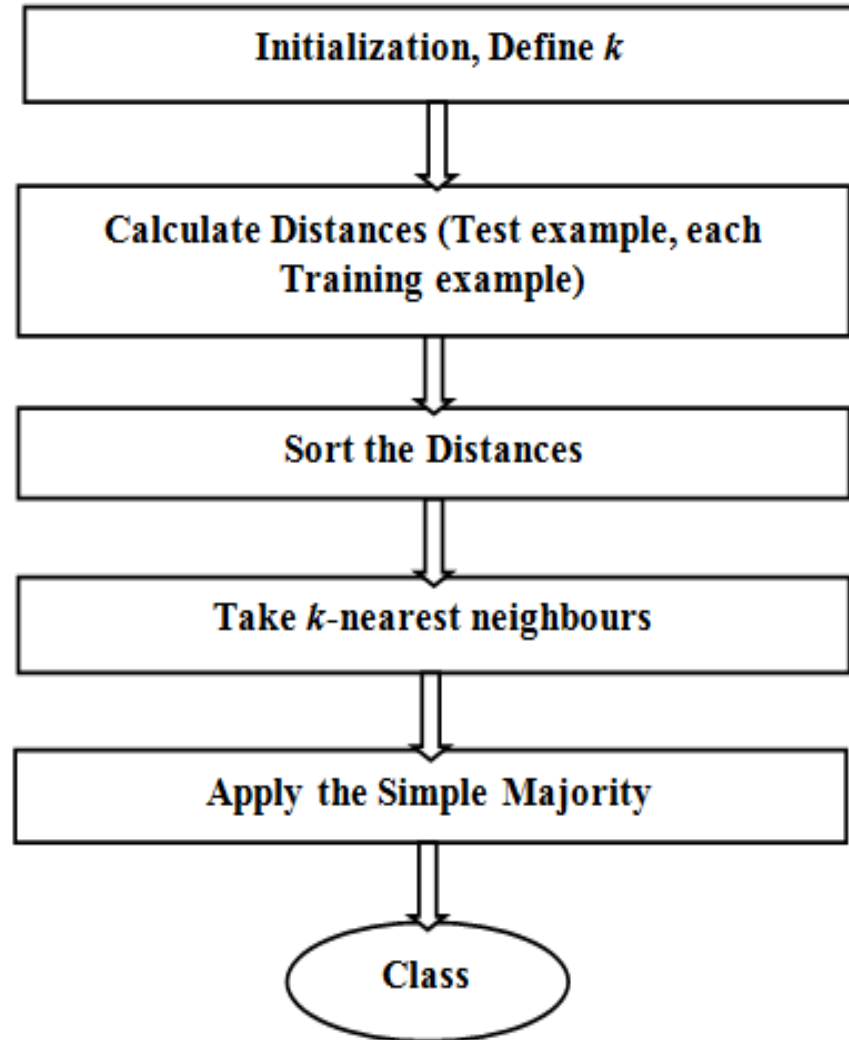
Giới thiệu thuật toán KNN

Tính khoảng cách giữa hai điểm A - B



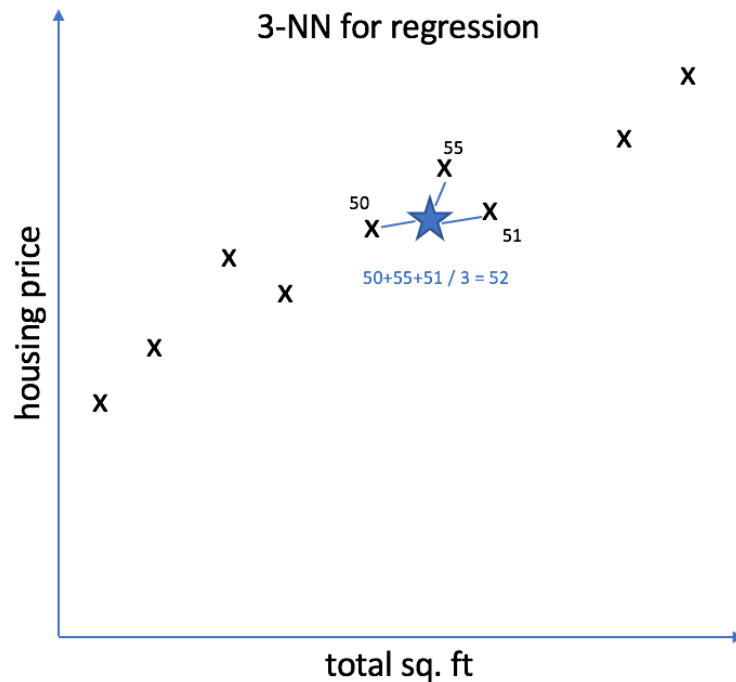
Euclidean	$\sqrt{\sum_{i=1}^k (x_i - y_i)^2}$
Manhattan	$\sum_{i=1}^k x_i - y_i $
Minkowski	$\left(\sum_{i=1}^k (x_i - y_i)^q \right)^{1/q}$

Các bước thực hiện thuật toán KNN

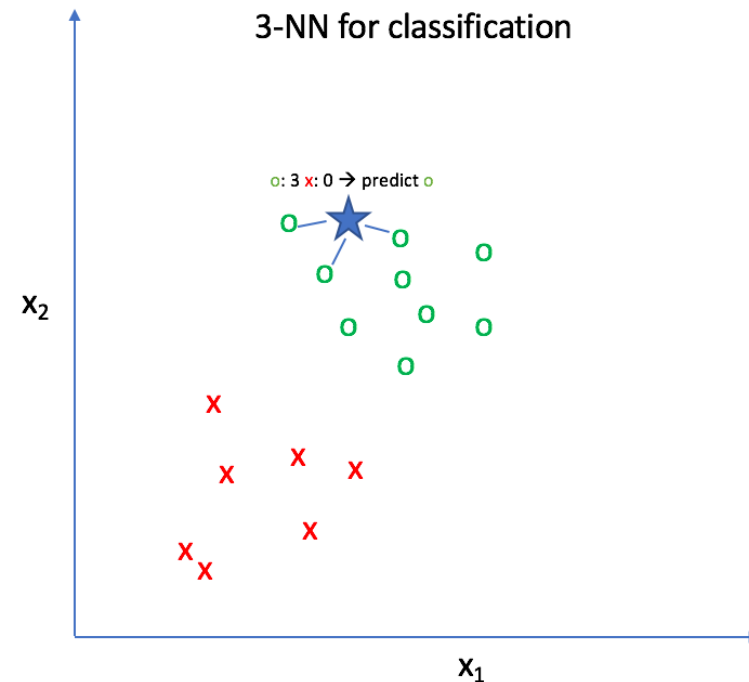


KNN cho bài toán phân lớp và hồi quy

KNN được sử dụng cho cả bài toán phân loại và hồi quy



Hồi quy (regression): nhãn của điểm dữ liệu mới được là nhãn của điểm dữ liệu đã biết gần nhất ($K=1$) hoặc trung bình có trọng số của những điểm gần nhất.



Phân loại (classification): nhãn của điểm dữ liệu mới được suy ra trực tiếp từ K điểm dữ liệu gần nhất.

Ưu nhược điểm của KNN

Ưu điểm:

- Độ phức tạp tính toán trong quá trình huấn luyện bằng 0
- Việc dự đoán kết quả của dữ liệu mới rất đơn giản
- Không cần giả sử gì về phân phối của các class

Nhược điểm:

- KNN rất nhạy với nhiễu khi K nhỏ.
- Tính toán khoảng cách tới từng điểm dữ liệu trong tập huấn luyện tốn rất nhiều thời gian, đặc biệt với các CSDL có số chiều lớn và có nhiều điểm dữ liệu. K càng lớn thì độ phức tạp càng tăng.
- Lưu toàn bộ dữ liệu trong bộ nhớ ảnh hưởng tới hiệu năng của KNN

....

- Nếu chơi với 5 người tự tin, bạn sẽ là người thứ 6.
- Nếu chơi với 5 người thông minh, bạn sẽ là người thứ 6.
- Nếu chơi với 5 triệu phú, bạn sẽ là người thứ 6.
- Nếu chơi với 5 kẻ ngốc, bạn sẽ là người thứ 6.
- Nếu chơi với 5 kẻ cháy túi, bạn cũng sẽ là người thứ 6.

MUỐN THÀNH CÔNG

Hãy chơi
với
**NGƯỜI GIỎI
HƠN BẠN**



"HÃY GIAO DU VỚI NHỮNG NGƯỜI TỐT HƠN BẠN. HÃY CHỌN BÊN CẠNH NHỮNG NGƯỜI CÓ CÁCH HÀNH XỬ TỐT HƠN MÌNH. NHƯ VẬY BẠN SẼ TỐT LÊN.."

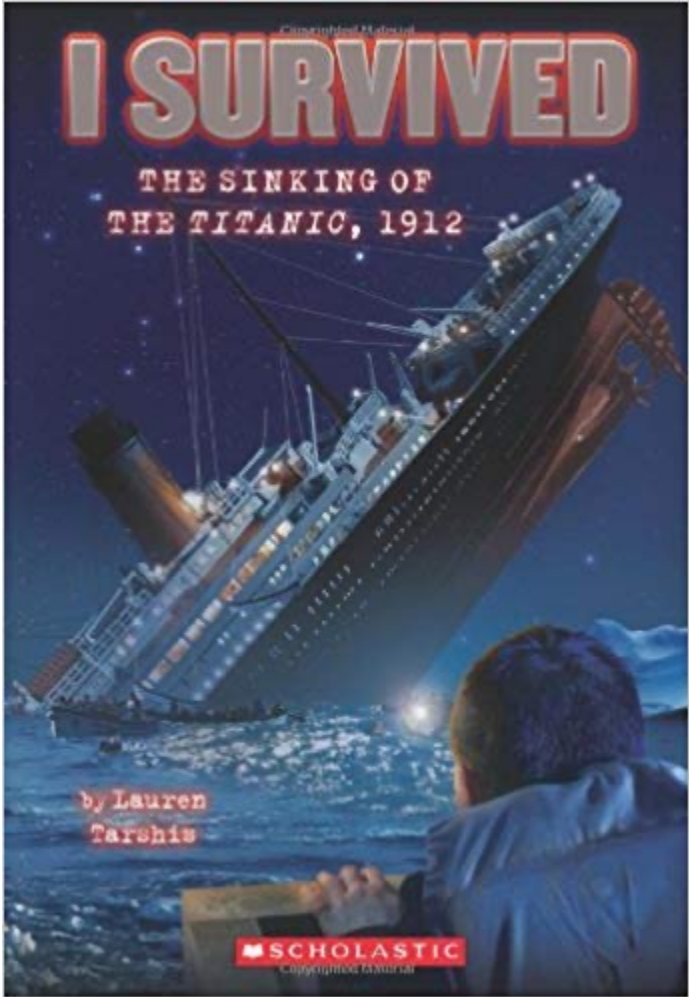
- Warren Buffett -

2. Thực hành với Bài toán Titanic

Bài toán Titanic

- Xây dựng model học máy sử dụng KNN dự đoán khả năng được cứu (1), Không được cứu (0) của hành khách trên tập dữ liệu đã được chuẩn bị ở chương 2:

	A	B	C	D	E	F	G
1	Survived	Pclass	Sex	Age	SibSp	Parch	Embarked
2	0	3	0	1	1	0	0
3	1	1	1	2	1	0	1
4	1	3	1	1	0	0	0
5	1	1	1	2	1	0	0
6	0	3	0	2	0	0	0
7	0	3	0	1	0	0	2
8	0	1	0	3	0	0	0
9	0	3	0	0	3	1	0
10	1	3	1	1	0	2	0
11	1	2	1	0	1	0	1
12	1	3	1	0	1	1	0
13	1	1	1	3	0	0	0
14	0	3	0	1	0	0	0
15	0	3	0	2	1	5	0



- Sinh viên làm trên Jupyter Notebook

Bài toán Titanic

1. Thuộc tính phụ thuộc (Nhãn - y):

* Survived: 0: Không được cứu - 1: Được cứu

2. Thuộc tính độc lập (Đầu vào - X):

* Pclass: Hạng vé(1 - hạng nhất, 2 - hạng 2, 3 - hạng 3)

* Sex: Giới tính (0 - Male, 1 - Nữ)

* Age: Độ tuổi

* 0: Tuổi từ 0 - 16 tuổi

* 1: Tuổi từ 17 - 32 tuổi

* 2: Tuổi từ 32 - 48 tuổi

* 3: Tuổi từ 48 - 64 tuổi

* 4: Tuổi từ 64 tuổi trở lên

* SibSp: Số lượng anh chị em đi cùng

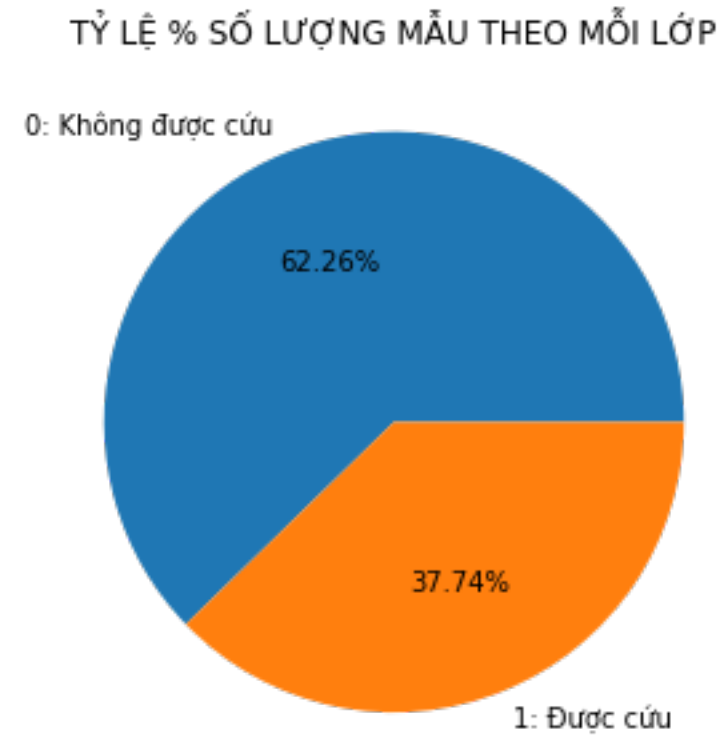
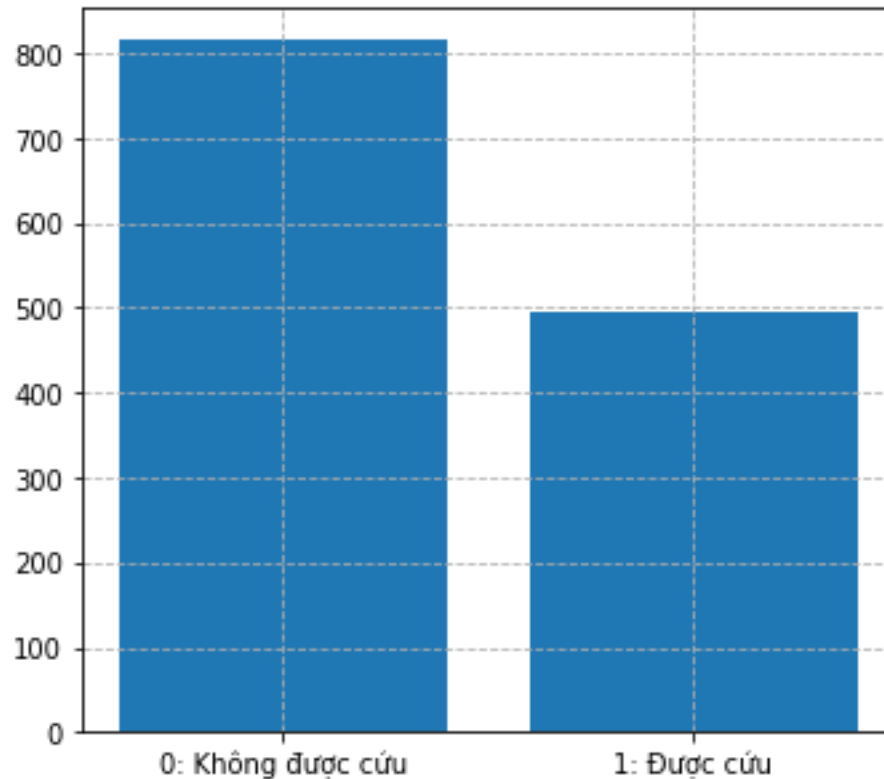
* Parch: Số lượng con cái, ba mẹ đi cùng

* Embarked: Cảng lên tàu (0: Cảng S, 1: Cảng C, 2: Cảng Q)

	A	B	C	D	E	F	G
1	Survived	Pclass	Sex	Age	SibSp	Parch	Embarked
2	0	3	0	1	1	0	0
3	1	1	1	2	1	0	1
4	1	3	1	1	0	0	0
5	1	1	1	2	1	0	0
6	0	3	0	2	0	0	0
7	0	3	0	1	0	0	2
8	0	1	0	3	0	0	0
9	0	3	0	0	3	1	0
10	1	3	1	1	0	2	0
11	1	2	1	0	1	0	1
12	1	3	1	0	1	1	0
13	1	1	1	3	0	0	0
14	0	3	0	1	0	0	0
15	0	3	0	2	1	5	0

Bài toán Titanic

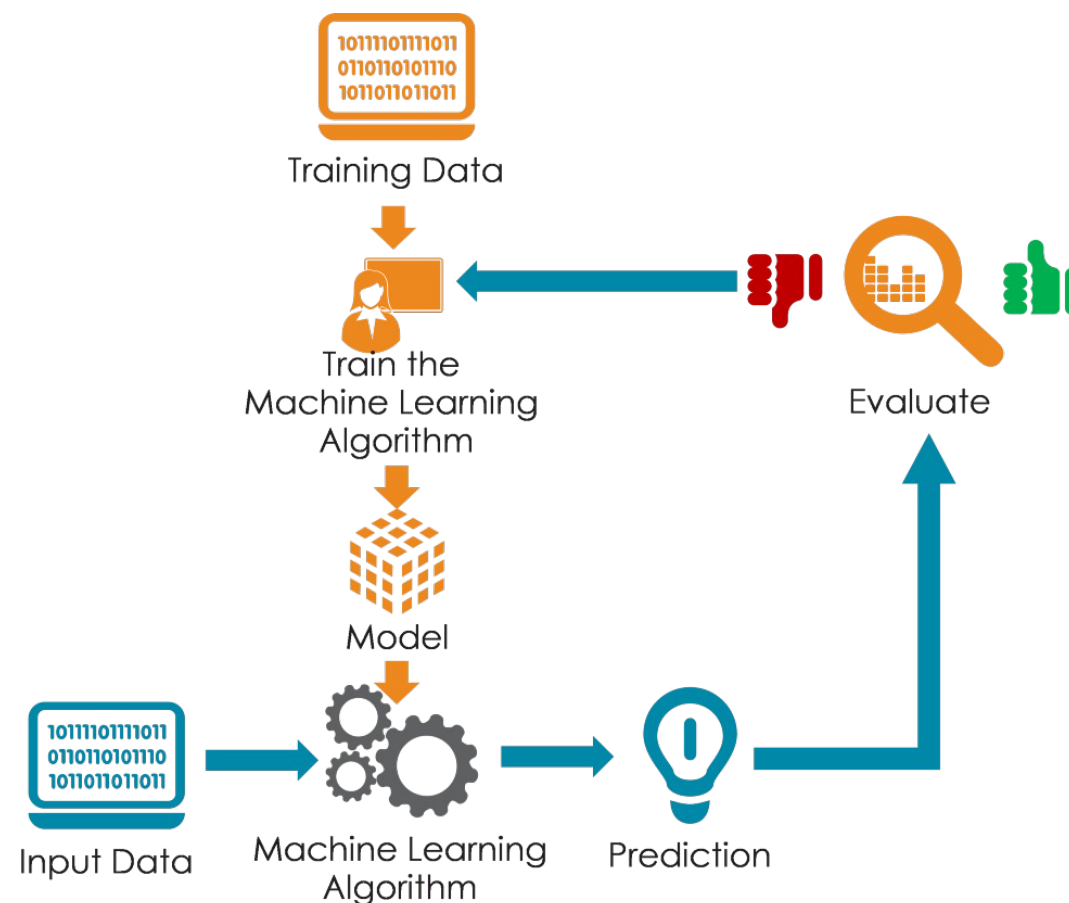
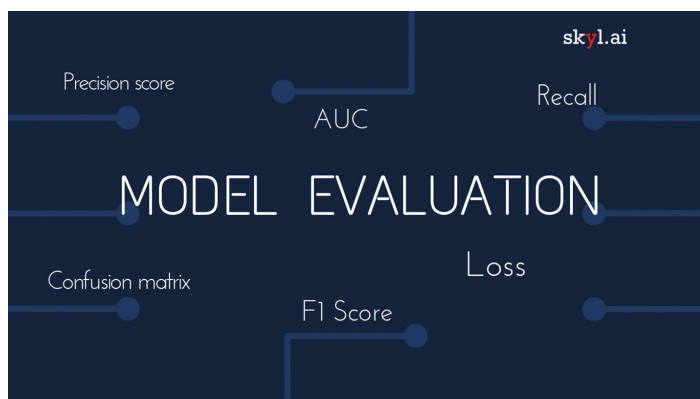
Đánh giá mức độ cân bằng của dữ liệu: Tập dữ liệu mất cân bằng nhẹ → Không cần xử lý



3. Đánh giá độ chính xác của mô hình phân lớp

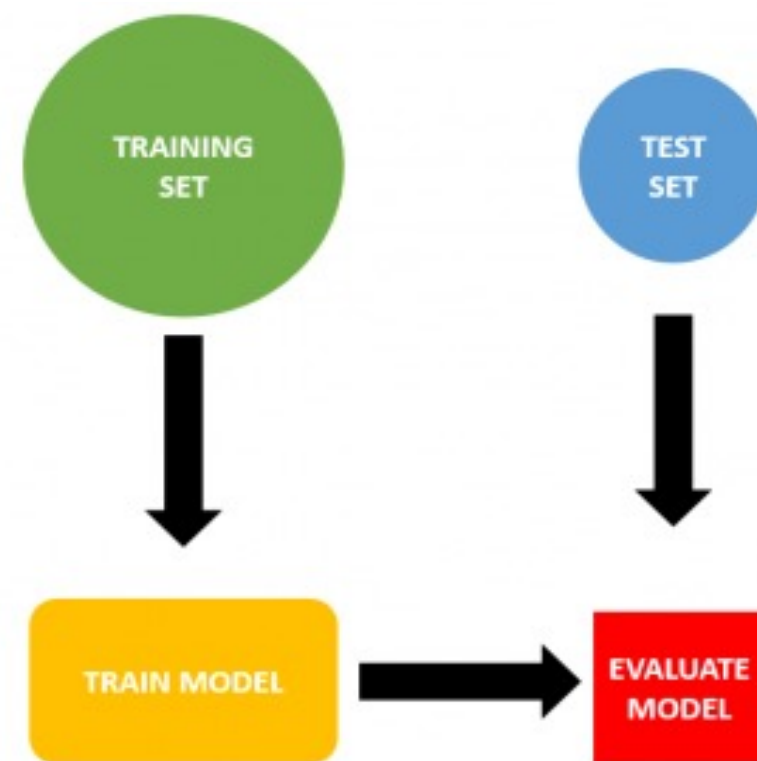
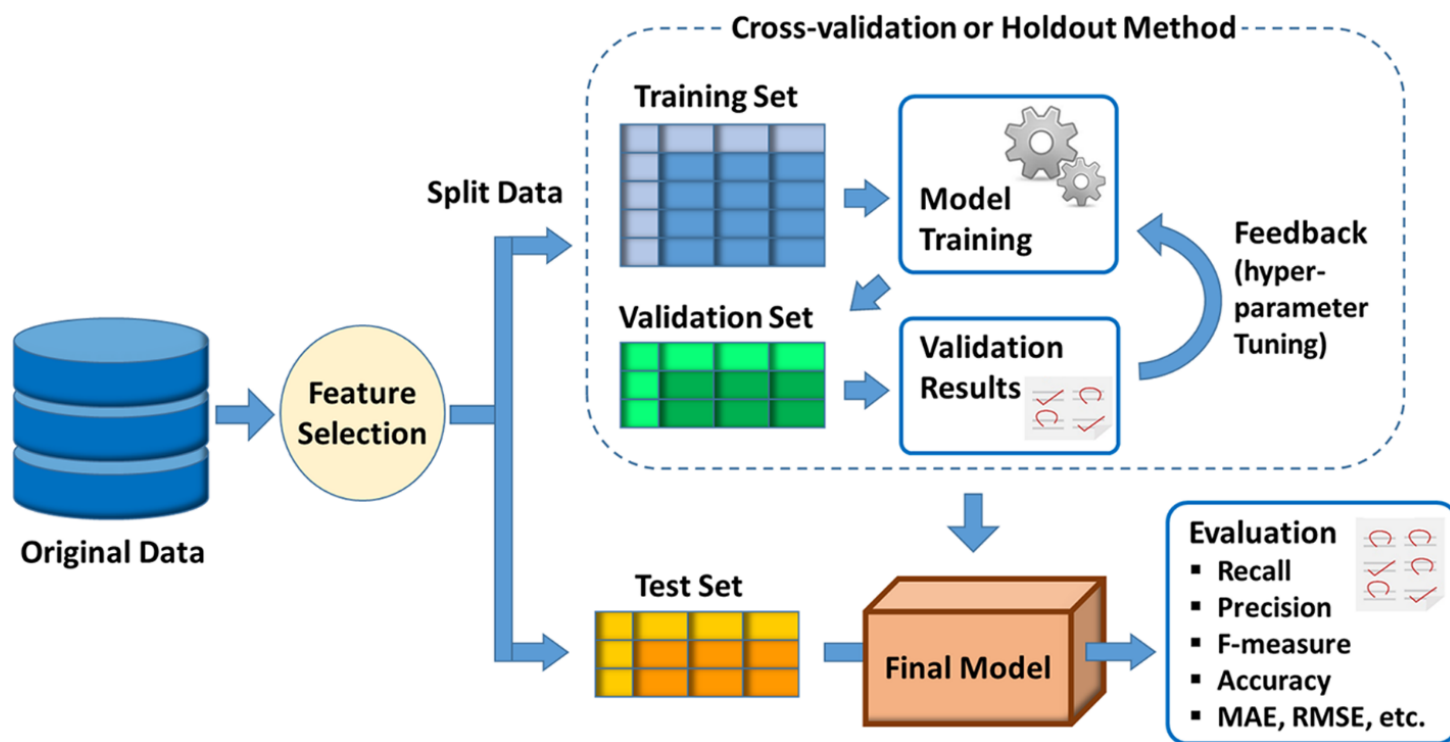
Đánh giá mô hình học máy

Khi xây dựng một mô hình học máy, chúng ta cần một phép đánh giá để xem mô hình sử dụng có hiệu quả không và để so sánh độ chính xác của các mô hình học máy khác nhau cho bài toán cần giải quyết.



Đánh giá mô hình học máy

Hiệu năng của một mô hình thường được đánh giá dựa trên tập dữ liệu kiểm thử (Test Data)



Đánh giá mô hình học máy

- **y_predict**: Kết quả dự đoán của mô hình học máy với tập dữ liệu kiểm thử (Test Data)
- **y_test (y_groundtruth)**: Nhãn thật (target) của tập dữ liệu kiểm thử (Test Data)

Phân lớp (Classification)

y_predict	y_groundtruth
0	0
0	0
1	0
0	0
0	1
1	1
0	0
0	0
1	0
1	1
1	0
0	1
0	0

Hồi quy (Regression)

y_predict	y_groundtruth
22 890	23 432
19 120	18 850
9 590	10 500
20 231	22 567
7 498	5 235
13 675	11 563
22 453	25 005
24 645	19 214
30 654	27 087
5 643	8 675
14 087	13 675
8 000	7 465
25 986	29 875

Đánh giá mô hình học máy

- Để đánh giá một hệ thống phân lớp thường sử dụng các tham số:
 - Accuracy
 - Confusion matrix
 - Precision
 - Recall
 - F1-score

Evaluation Metrics for Classification Models

Accuracy, Confusion Matrix, Precision, Recall, Specificity, F1-Score

n = 165	Predicted: No	Predicted: Yes
Actual: No	50	10
Actual: Yes	5	100

Đánh giá mô hình học máy

- Cách đơn giản và hay được sử dụng nhất là Accuracy: Tính tỷ lệ giữa số điểm được dự đoán đúng và tổng số điểm trong tập dữ liệu kiểm thử.

y_predict	y_groundtruth
0	0
0	0
1	0
0	0
0	1
1	1
0	0
0	0
1	0
1	1
1	0
0	1
0	0

```
1 #Sử dụng module accuracy_score trong thư viện sklearn để đánh giá độ chính xác:
2 from sklearn.metrics import accuracy_score
3
4 #1. Đếm tổng số mẫu dự đoán đúng trên tập Test
5 acc1 = accuracy_score(y_test, y_pred, normalize=False)
6 print('Tổng số mẫu dự đoán đúng:', acc1, ' / ', len(y_test))
```

Tổng số mẫu dự đoán đúng: 216 / 262

```
1 #2.Độ chính xác theo tỷ lệ % số mẫu dự đoán đúng/tổng số mẫu của tập test (Accuracy)
2 acc2 = accuracy_score(y_test, y_pred)
3 print('Độ chính xác của mô hình (k=5 default):', acc2)
4 print('Độ chính xác theo %:', round(acc2*100,2))
```

Độ chính xác của mô hình (k=5 default): 0.8244274809160306


Độ chính xác theo %: 82.44

Đánh giá mô hình học máy

- Cách tính sử dụng Accuracy như ở trên chỉ cho chúng ta biết được bao nhiêu phần trăm lượng dữ liệu được phân loại đúng (hoặc tổng có bao nhiêu mẫu phân loại đúng) mà không chỉ ra được cụ thể mỗi loại được phân loại như thế nào, lớp nào được phân loại đúng nhiều nhất, và dữ liệu thuộc lớp nào thường bị phân loại nhầm vào lớp khác. Để có thể đánh giá được các giá trị này, chúng ta sử dụng một ma trận được gọi là ma trận nhầm lẫn - *confusion matrix*.

Making sense of the confusion matrix

	Predicted: NO	Predicted: YES
Actual: NO	50	10
Actual: YES	5	100



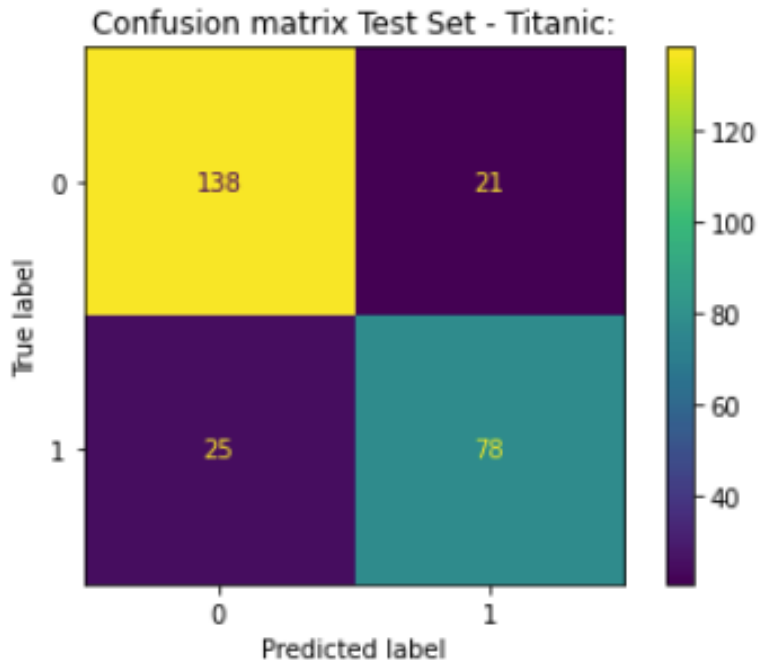
Đánh giá mô hình học máy

- Confusion matrix là một ma trận vuông với kích thước mỗi chiều bằng số lượng lớp dữ liệu. Giá trị tại hàng i , cột j là số lượng điểm lẽ ra thuộc vào lớp i nhưng lại được dự đoán là thuộc vào lớp j .
- Tổng số phần tử của toàn ma trận chính là số điểm trong tập kiểm thử. Các phần tử trên đường chéo của ma trận là số điểm được phân loại đúng của mỗi lớp dữ liệu.

		Predicted			
		A	B	C	
True labels	A	2	2	0	4
	B	1	2	0	3
	C	0	0	3	3
		3	4	3	Total

Đánh giá mô hình học máy

```
1 #Sử dụng ma trận confusion matrix kiểm tra kết quả:  
2 from sklearn.metrics import confusion_matrix, ConfusionMatrixDisplay  
3 cnf_matrix_titanic = confusion_matrix(y_test, y_pred)  
4  
5 ConfusionMatrixDisplay.from_predictions(y_test, y_pred)  
6 plt.title('Confusion matrix Test Set - Titanic:')  
7 plt.show()
```



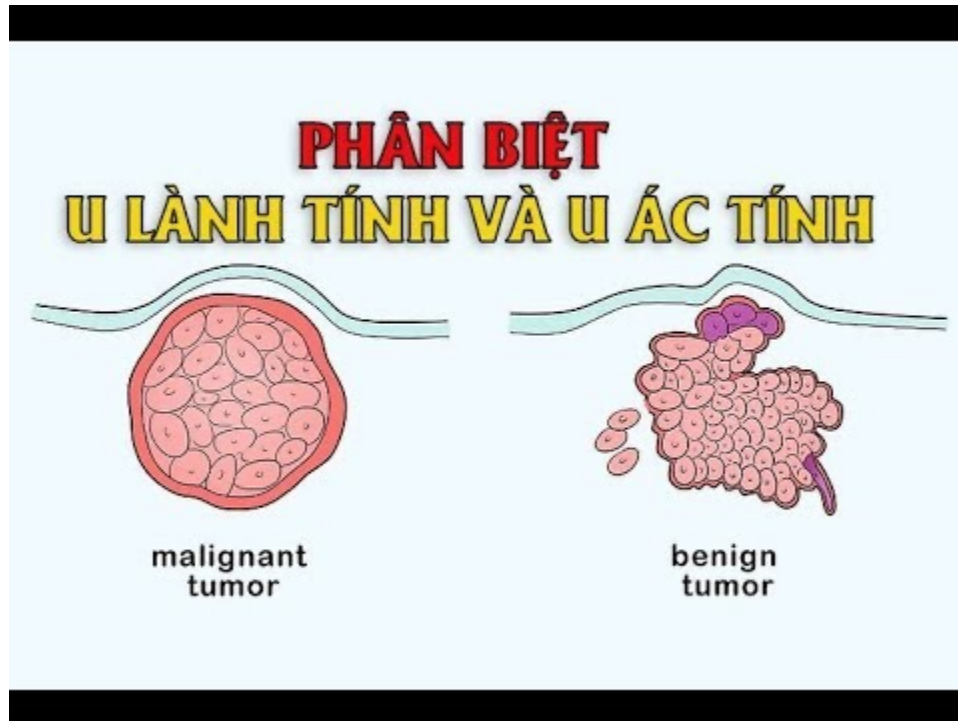
- Tổng số mẫu trong tập Test: $138 + 78 + 21 + 25 = 262$
- Số mẫu trong tập Test thuộc lớp 0: $138 + 21 = 159$
- Số mẫu trong tập Test thuộc lớp 1: $25 + 78 = 103$
- Trong lớp 0: mô hình dự đoán đúng được **138** mẫu dự đoán sai **21** mẫu
- Trong lớp 1: mô hình dự đoán đúng được **78** mẫu sai **25** mẫu:
- Tổng số mẫu dự đoán đúng: $138 + 78 = 216$
- Tổng số mẫu dự đoán sai: $21 + 25 = 46$

4. Bài tập

Bài tập thực hành

Xây dựng mô hình học máy với thuật toán KNN dự đoán một bệnh nhân u vú là lành tính hay ác tính. Sinh viên sử dụng tập 961 bệnh nhân u vú đã xử lý ở Chương 2;

1. Kiểm tra mức độ cân bằng dữ liệu
2. Phân tách các biến Độc lập (X) - Phụ thuộc (Y) tương ứng
3. Chia tập dữ liệu thành 2 phần Train - Test với tỷ lệ 75% - 25%



Bài tập thực hành

4. Xây dựng model dự đoán bệnh nhân bị bệnh u vú lành tính - ác tính với thuật toán K người láng giềng gần nhất (KNN). Tùy chỉnh tham số `n_neighbors`, `weight`, `p` để thu được mô hình có độ chính xác tốt nhất trên tập Train và Test.

Hiển thị các kết quả sau của model:

- a) Độ chính xác của model (accuracy) trên tập Train – Test
- b) Tổng số mẫu dự đoán đúng - sai trên tập Test
- c) Ma trận confusion matrix trên tập Test

Bài tập thực hành

5. Dự đoán với mô hình xây dựng được cho 2 bệnh nhân sau:

- Bệnh nhân 1:

- * Age: 20 tuổi;
- * Weight: 65 Kg
- * Shape: Round;
- * Margin: Circumscribed
- * Density: Low

- Bệnh nhân 2 :

- * Age: 64 tuổi;
- * Weight: 75 kg
- * Shape: Round;
- * Margin: Circumscribed
- * Density: High

Sinh viên làm tiếp vào file bài tập chương 2, sau đó đặt lại tên:
Manhom_MaSV_Hoten_Baitap_KNN và nộp bài theo link:

→ Link nộp bài: <https://forms.gle/aqEtp4sc2W8AFm7z5>



Thank you!