

**HO CHI MINH CITY UNIVERSITY OF TRANSPORT****Kiến thức - Kỹ năng - Sáng tạo - Hội nhập**Sứ mệnh - Tâm nhínTriết lý Giáo dục - Giá trị cốt lõi

## Contents

<b>0 Resources</b>	<b>2</b>
<b>1 Searching Agent</b>	<b>3</b>
<b>2 Constraint Satisfaction Problem (CSP)</b>	<b>9</b>
<b>3 Proposition Logic</b>	<b>10</b>
<b>4 First Order Logic</b>	<b>22</b>
<b>5 Inference in First Order Logic</b>	<b>24</b>
<b>6 Leaner Regression</b>	<b>25</b>
<b>7 Decision Tree</b>	<b>27</b>
<b>8 Naive Bayes Classifier (NBC) and Support Vector Machine (SVM)</b>	<b>29</b>
<b>9 Rosenblatt's Perceptron</b>	<b>31</b>
<b>10 Convolutional Neural Networks (CNN)</b>	<b>32</b>
<b>11 Recurrent Neural Network (RNN)</b>	<b>34</b>
<b>12 Generative Adversarial Network (GAN)</b>	<b>36</b>

## 0 Resources

### 0.1 Searching Agent

- <https://networkx.org/>

### 0.2 Constraint

- <https://pypi.org/project/python-constraint/>
- <https://pycsp.org/>
- <https://developers.google.com/optimization>

### 0.3 Proposition logic & First order logic

- <https://pypi.org/project/z3-solver/>
- <https://microsoft.github.io/z3guide/>
- SymPy

### 0.4 Automated planning

- <https://solver.planning.domains/>
- <https://marketplace.visualstudio.com/items?itemName=jan-dolejsi.pddl>
- <https://github.com/jan-dolejsi/pddl-reference?tab=readme-ov-file>

### 0.5 Machine Learning

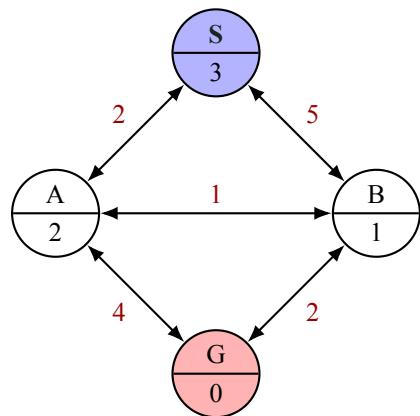
- <https://developers.google.com/machine-learning/crash-course>
- <https://www.kaggle.com/>
- <https://orangedatamining.com/>

### 0.6 Neural Network

- <https://developers.google.com/machine-learning/crash-course/neural-networks>
- <https://alexlenail.me/NN-SVG/index.html>
- <https://playground.tensorflow.org/>
- <http://neurallnetworksanddeeplearning.com/>

# 1 Searching Agent

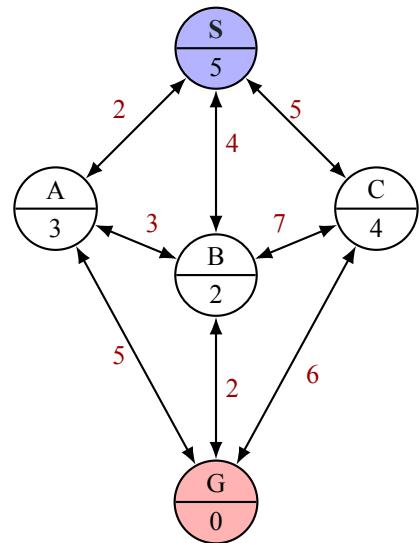
## 1.1 Searching



initial state: S; Goal state: G

- a) Best-first search.
- b) Breadth-first search.
- c) Depth-first search.
- d) A\* search.

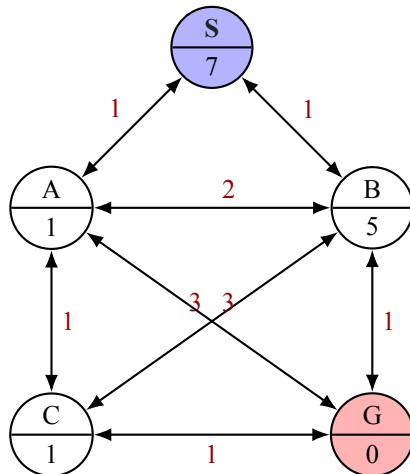
## 1.2 Searching



initial state: S; Goal state: G

- a) Best-first search.
- b) Breadth-first search.
- c) Depth-first search.
- d) A\* search.

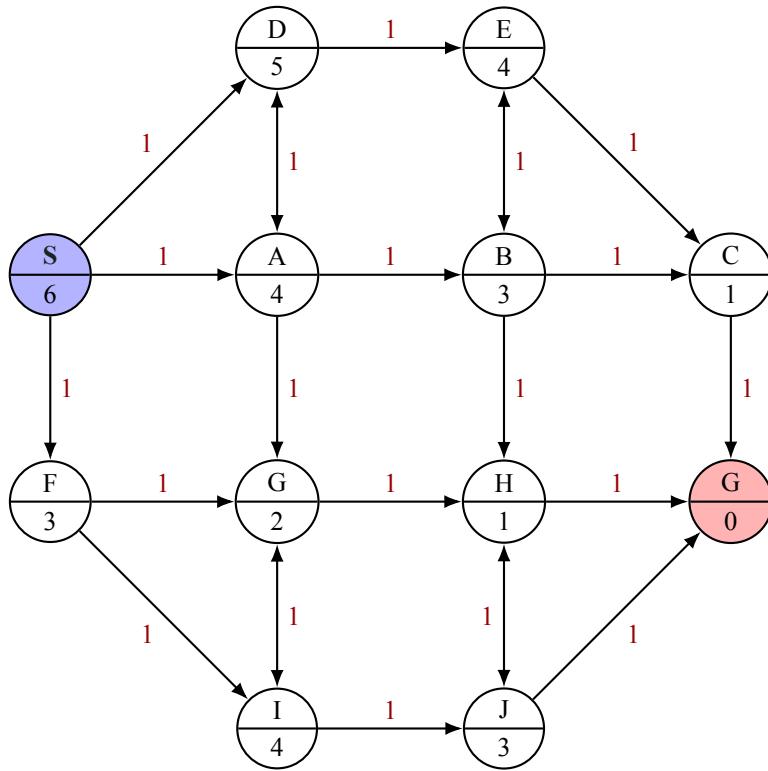
## 1.3 Searching



initial state: S; Goal state: G

- a) Best-first search.
- b) Breadth-first search.
- c) Depth-first search.
- d) A\* search.

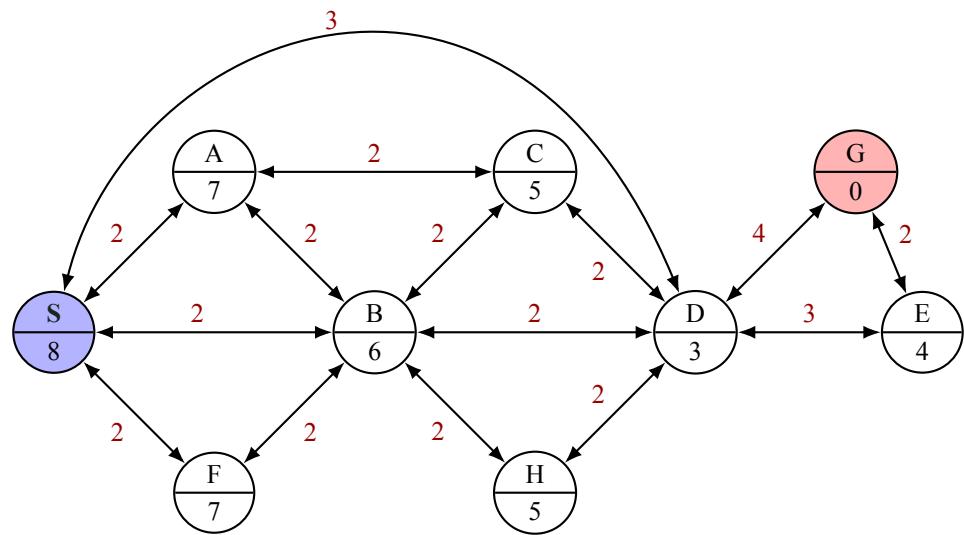
## 1.4 Searching



initial state: S; Goal state: G

- a) Best-first search.
- b) Breadth-first search.
- c) Depth-first search.
- d) A\* search.

### 1.5 Searching (Turtle)

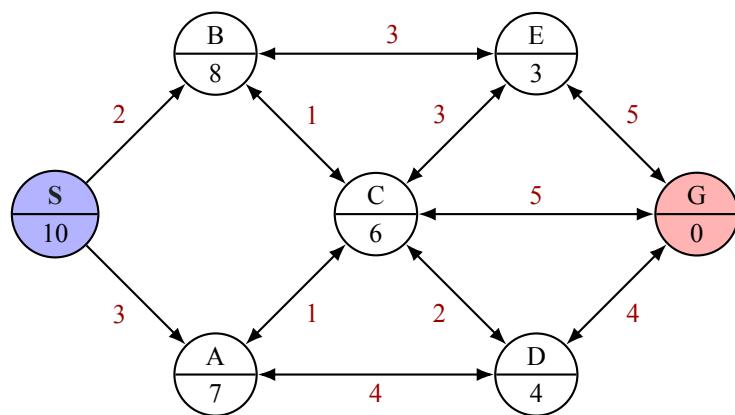


**Turtle**

Initial state: S; Goal state: G

- Best-first search.
- Breadth-first search.
- Depth-first search.
- A\* search.

### 1.6 Searching



Initial state: S; Goal state: G

- Best-first search.
- Breadth-first search.
- Depth-first search.
- A\* search.

### 1.7 The 8-puzzle problem

**Initial state:**

1	2	3
8		4
7	6	5

**Goal state:**

1	2	3
4	5	6
7	8	

$h_1$ : Number of misplaced tiles.

$h_2$ : Sum of Euclidian distances of the tiles from their goal positions.

$h_3$ : Sum of Manhattan distances of the tiles from their goal positions.

$h_4$ : Number of tiles out of row + Number of tiles out of column.

$g$ : Depth of node - How many steps that have already done from the initial state or the current state.

$$f(n) = g(n) + h(n)$$

Draw tree searching by use:

- a) Breadth-first search.
- b) Depth-first search.
- c) Greedy Best-first with  $h_1, h_2, h_3, h_4$ .
- d) A\* with  $h_1, h_2, h_3, h_4$ .

### 1.8 The 8-puzzle problem

**Initial state:**

2	8	3
1	6	4
7		5

**Goal state:**

1	2	3
8		4
7	6	5

Draw tree searching by use:

- a) Breadth-first search.
- b) Depth-first search.
- c) Greedy Best-first with  $h_1, h_2, h_3, h_4$ .
- d) A\* with  $h_1, h_2, h_3, h_4$ .

### 1.9 The 8-puzzle problem

Initial state:

7	2	4
5		6
8	3	1

Goal state:

	1	2
3	4	5
6	7	8

Draw tree searching by use:

- a) Breadth-first search.
- b) Depth-first search.
- c) Greedy Best-first with  $h_1, h_2, h_3, h_4$ .
- d) A\* with  $h_1, h_2, h_3, h_4$ .

## 2 Constraint Satisfaction Problem (CSP)

### 2.1 Scheduling 3 tasks

We have three tasks: A, B, and C. Each task needs to be assigned a time slot from a set of possible time slots: 1, 2, or 3. However, there are constraints on when certain tasks can be scheduled:

- Task A cannot be scheduled at the same time as task B.
- Task B must be scheduled before task C.

### 2.2 Scheduling Classes

Let's consider a scheduling problem for a university with multiple courses and classrooms. The goal is to schedule classes in a way that satisfies various constraints:

#### Variables:

- Courses: Math (M), Physics (P), Chemistry (C), Biology (B)
- Classrooms: R1, R2, R3

#### Constraints:

1. Each course must be assigned to exactly one classroom.
2. No two courses can be held in the same classroom at the same time.
3. There are specific time constraints for certain courses:
  - Math and Physics cannot be scheduled at the same time.
  - Chemistry and Biology cannot be scheduled at the same time.
4. Each course has preferred time slots:
  - Math prefers morning sessions.
  - Physics prefers afternoon sessions.
  - Chemistry prefers morning sessions.
  - Biology prefers afternoon sessions.

### 2.3 Scheduling problem

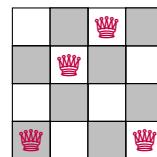
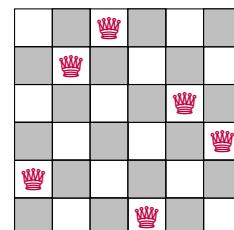
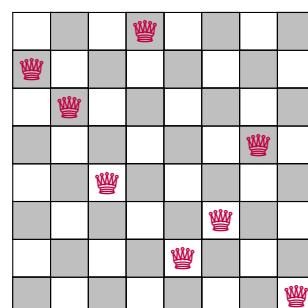
You're tasked with scheduling a series of events at a conference. The conference has multiple sessions and each session can accommodate different talks. However, there are various constraints and preferences to consider:

#### Variables:

- Talks: A, B, C, D, E, F
- Sessions: Morning Session (M), Afternoon Session (A), Evening Session (E)

**Constraints:**

1. Each talk must be assigned to exactly one session.
2. No two talks can be scheduled in the same session simultaneously.
3. Certain talks have special requirements:
  - Talk A and B are co-dependent and must be scheduled in the same session.
  - Talk C and D cannot be scheduled in the same session.
  - Talk E must be scheduled before talk F.

**2.4 4-Queens****2.5 6-Queens****2.6 8-Queens****2.7 Wolf, Goat and Cabbage problem**

Một nông dân có 3 mặt hàng: một con sói, một con dê và một quả bắp cải phải băng qua sông bằng thuyền.

Thuyền chỉ có thể mang theo nông dân và một mặt hàng duy nhất.

Nếu ở cùng nhau và không có người nông dân thì sói sẽ ăn dê, hoặc dê sẽ ăn bắp cải.

Làm thế nào họ có thể qua sông mà không có gì bị ăn?

**Lưu ý:**

Đây là bài toán cho cả CSP và Proposition Logic.

**3 Proposition Logic****3.1 Compute the Truth tables**

a)  $(\neg P) \Leftrightarrow (\neg Q)$  (compare to  $P \Leftrightarrow Q$ )

b)  $[P \vee (\neg Q)] \Leftrightarrow P$

c)  $\neg[P \wedge (Q \vee R)]$

d)  $\neg(P \wedge Q)$

e)  $P \wedge (\neg P) \Rightarrow Q$

f)  $(P \wedge Q) \Rightarrow P$

g)  $P \Rightarrow (P \vee Q)$

### 3.2 Truth tables method

a)  $(P \Rightarrow Q) \vee (P \Rightarrow \neg Q)$  : valid?

b)  $(\neg P \vee Q) \wedge (Q \Rightarrow \neg R \wedge \neg P) \wedge (P \vee R)$  : satisfiable?

c) Determine whether the  $(P \wedge \neg Q \Rightarrow P \wedge Q)$  is a logical consequence of the  $\neg P$

d) Determine whether  $P \Rightarrow (Q \wedge \neg Q)$  and  $\neg P$  are logically equivalent

### 3.3 Truth tables method verify whether the following logical consequences and equivalences are correct

a)  $(P \Rightarrow Q) \models \neg P \Rightarrow \neg Q$

b)  $(P \Rightarrow Q) \wedge \neg Q \models \neg P$

c)  $P \Rightarrow (Q \wedge R) \models (P \Rightarrow Q) \Rightarrow R$

d)  $P \vee (\neg Q \wedge R) \models Q \vee \neg R \Rightarrow P$

e)  $\neg(P \wedge Q) \equiv \neg P \vee \neg Q$

f)  $(P \vee Q) \wedge (\neg P \Rightarrow \neg Q) \equiv Q$

g)  $(P \wedge Q) \vee R \models (P \Rightarrow \neg Q) \Rightarrow R$

h)  $(P \vee Q) \wedge (\neg P \Rightarrow \neg Q) \equiv P$

i)  $((P \Rightarrow Q) \Rightarrow Q) \Rightarrow Q \equiv P \Rightarrow Q$

### 3.4 Which of the following are correct?

a)  $\text{False} \models \text{True}$

b)  $\text{True} \models \text{False}$

c)  $(A \wedge B) \models (A \Leftrightarrow B)$

d)  $A \Leftrightarrow B \models A \vee B$

e)  $A \Leftrightarrow B \models \neg A \vee B$

f)  $(A \wedge B) \Rightarrow C \models (A \Rightarrow C) \vee (B \Rightarrow C)$

g)  $(C \vee (\neg A \wedge \neg B)) \equiv ((A \Rightarrow C) \wedge (B \Rightarrow C))$

- h)  $(A \vee B) \wedge (\neg C \vee \neg D \vee E) \models (A \vee B)$
- i)  $(A \vee B) \wedge (\neg C \vee \neg D \vee E) \models (A \vee B) \wedge (\neg D \vee E)$
- j)  $(A \vee B) \wedge \neg(A \rightarrow B)$  is satisfiable.
- k)  $(A \Leftrightarrow B) \wedge (\neg A \vee B)$  is satisfiable.
- l)  $(A \Leftrightarrow B) \Leftrightarrow C$  has the same number of models as  $(A \Leftrightarrow B)$   
for any fixed set of proposition symbols that includes  $A, B, C$ .

### 3.5 Prove each of the following assertions:

- a)  $\alpha$  is valid if and only if  $\text{True} \models \alpha$
- b) For any  $\alpha$ ,  $\text{False} \models \alpha$
- c)  $\alpha \models \beta$  if and only if the sentence  $(\alpha \Rightarrow \beta)$  is valid.
- d)  $\alpha \equiv \beta$  if and only if the sentence  $(\alpha \Leftrightarrow \beta)$  is valid.
- e)  $\alpha \models \beta$  if and only if the sentence  $(\alpha \wedge \neg\beta)$  is unsatisfiable.

### 3.6 Prove, or find a counterexample to, each of the following assertions:

- a) If  $\alpha \models \gamma$  or  $\beta \models \gamma$  (or both) then  $(\alpha \wedge \beta) \models \gamma$
- b) If  $(\alpha \wedge \beta) \models \gamma$  then  $\alpha \models \gamma$  or  $\beta \models \gamma$  (or both).
- c) If  $\alpha \models (\beta \vee \gamma)$  then  $\alpha \models \beta$  or  $\alpha \models \gamma$  (or both).
- d) If  $\alpha \models (\beta \wedge \gamma)$  then  $\alpha \models \beta$  and  $\alpha \models \gamma$

### 3.7 Valid, unsatisfiable, or neither

Verify your decisions using truth tables or the equivalence rules of Table.

- a)  $\text{Smoke} \Rightarrow \text{Smoke}$
- b)  $\text{Smoke} \Rightarrow \text{Fire}$
- c)  $(\text{Smoke} \Rightarrow \text{Fire}) \Rightarrow (\neg \text{Smoke} \Rightarrow \neg \text{Fire})$
- d)  $\text{Smoke} \vee \text{Fire} \vee \neg \text{Fire}$
- e)  $((\text{Smoke} \wedge \text{Heat}) \models \text{Fire}) \Leftrightarrow ((\text{Smoke} \models \text{Fire}) \vee (\text{Heat} \Rightarrow \text{Fire}))$
- f)  $(\text{Smoke} \Rightarrow \text{Fire}) \Rightarrow ((\text{Smoke} \wedge \text{Heat}) \models \text{Fire})$
- g)  $\text{Big} \vee \text{Dumb} \vee (\text{Big} \Rightarrow \text{Dumb})$

### 3.8 Convert into CNF

- a)  $P \Leftrightarrow Q$
- b)  $(P \Rightarrow Q) \Rightarrow R$
- c)  $(P \vee Q) \wedge (P \wedge R)$

- d)  $(P \Rightarrow Q) \Leftrightarrow (P \Rightarrow R)$
- e)  $(A \wedge \neg\neg B) \vee (\neg C \wedge D)$
- f)  $(P \vee (Q \wedge \neg A)) \wedge (\neg P \vee A \vee \neg B)$
- g)  $\neg(P \Rightarrow Q) \vee ((R \vee S) \Rightarrow (Q \vee T)) \vee (\neg P \Rightarrow \neg V)$
- h)  $P \Rightarrow (Q \wedge R)$
- i)  $(P \vee Q) \Rightarrow R$
- j)  $\neg(\neg P \vee Q) \vee (R \Rightarrow \neg S)$
- k)  $\neg((P \Rightarrow (Q \Rightarrow R))) \Rightarrow ((P \Rightarrow Q) \Rightarrow (P \Rightarrow R))$
- l)  $P \vee (\neg Q \wedge (R \Rightarrow \neg P))$
- m)  $\neg(((A \Rightarrow B)) \Rightarrow A) \Rightarrow A$
- n)  $\neg(A \vee (A \Rightarrow B))$
- o)  $\neg P \wedge Q \vee \neg(\neg R \vee \neg Q)$

### 3.9 Prove with resolution that the following formulas are tautologies

- a)  $P \Rightarrow (Q \Rightarrow P)$
- b)  $(P \wedge (P \Rightarrow Q)) \Rightarrow Q$
- c)  $((P \Rightarrow Q) \wedge \neg Q) \Rightarrow \neg P$
- d)  $((P \Rightarrow Q) \wedge \neg Q) \Rightarrow \neg Q$

### 3.10 Prove using resolution that

$$P \Rightarrow (Q \wedge R) \models (\neg Q \vee \neg R) \Rightarrow \neg P$$

### 3.11 Prove using resolution that

$P \Rightarrow Q$  is a logical consequence of  $((T \Rightarrow Q) \wedge (\neg R \Rightarrow \neg S) \wedge (P \Rightarrow U) \wedge (\neg T \Rightarrow \neg R) \wedge (U \Rightarrow S))$

### 3.12 Prove with resolution

- a) 1.  $(P \Rightarrow Q) \Rightarrow Q$   
2.  $(P \Rightarrow P) \Rightarrow R$   
3.  $(R \Rightarrow S) \Rightarrow \neg(S \Rightarrow Q)$   
 $\therefore R$
- b) 1.  $\neg P \vee Q$   
2.  $\neg R \Rightarrow \neg Q$   
 $\therefore P \Rightarrow R$
- c) 1.  $(P \Rightarrow Q) \Rightarrow Q$   
2.  $(P \Rightarrow P) \Rightarrow R$   
3.  $(R \Rightarrow S) \Rightarrow \neg(S \Rightarrow Q)$   
 $\therefore R$

### 3.13 Forward chaining and Backward chaining

- a) 1.  $G \Rightarrow A$
2.  $(A \wedge B \wedge C) \Rightarrow D$
3.  $(E \wedge B) \Rightarrow C$
4.  $D \Rightarrow H$
5.  $(C \wedge H) \Rightarrow F$
6.  $G \Rightarrow B$
7.  $G$
8.  $E$
- $\therefore H$

### 3.14 The Lady or the Tiger

Frank Stockton's story:

Nhà vua muốn tha thứ cho một tù nhân thông minh nên để ra trò chơi: Trò chơi có một số phòng và anh tù nhân đến cửa phòng và dựa trên các biển báo của các phòng để mở cửa phòng. Tùy theo luật chơi, nếu tù nhân mở cửa phòng của công chúa, anh ta sẽ được tha tội và được cưới công chúa, nhưng nếu có hổ trong phòng, anh ta sẽ bị hổ ăn thịt.

a) Phiên bản 1 (2 phòng).

- Có 2 phòng; Mỗi phòng hoặc là có công chúa hoặc là chúa hổ. Có thể cả hai phòng đều chúa hổ hay cả hai phòng đều có công chúa và có thể có một phòng có công chúa, một phòng chúa hổ.
- Biển báo trên cửa phòng này đúng thì biển báo trên cửa phòng kia sai và ngược lại.
- Biển báo trên cửa phòng 1: Phòng này có công chúa và phòng kia có hổ.
- Biển báo trên cửa phòng 2: Một trong hai phòng này có công chúa và phòng còn lại có hổ.

♡ Anh tù nhân phải mở cửa phòng nào để được tha tội và được cưới công chúa.

b) Phiên bản 2 (2 phòng).

- Có 2 phòng; Mỗi phòng hoặc là có công chúa hoặc là chúa hổ. Có thể cả hai phòng đều chúa hổ hay cả hai phòng đều có công chúa và có thể có một phòng có công chúa, một phòng chúa hổ.
- Cả hai biển báo đều đúng hoặc đều sai.
- Biển báo trên cửa phòng 1: Ít nhất có một phòng có công chúa.
- Biển báo trên cửa phòng 2: Phòng kia có hổ.

♡ Anh tù nhân phải mở cửa phòng nào để được tha tội và được cưới công chúa.

c) Phiên bản 3 (2 phòng).

- Có 2 phòng; Mỗi phòng hoặc là có công chúa hoặc là chúa hổ. Có thể cả hai phòng đều chúa hổ hay cả hai phòng đều có công chúa và có thể có một phòng có công chúa, một phòng chúa hổ.
- Cả hai biển báo đều đúng hoặc đều sai.
- Biển báo trên cửa phòng 1: Hổ ở phòng này hay công chúa ở phòng kia.
- Biển báo trên cửa phòng 2: Công chúa ở phòng kia.

♡ Anh tù nhân phải mở cửa phòng nào để được tha tội và được cưới công chúa.

## d) Phiên bản 4 (2 phòng).

- Có 2 phòng; Mỗi phòng hoặc là có công chúa hoặc là chứa hổ. Có thể cả hai phòng đều chứa hổ hay cả hai phòng đều có công chúa và có thể có một phòng có công chúa, một phòng chứa hổ.
  - Ở phòng 1: Nếu có công chúa thì biển báo của phòng là **đúng** nhưng nếu có hổ trong phòng thì biển báo của phòng là **sai**.
  - Ở phòng 2: Ngược lại với phòng 1, nếu có công chúa thì biển báo của phòng là **sai** nhưng nếu có hổ trong phòng thì biển báo của phòng là **đúng**.
  - Biển báo trên cửa phòng 1: Cả 2 phòng đều có công chúa.
  - Biển báo trên cửa phòng 2: Cả 2 phòng đều có công chúa.
- ♡ *Anh tù nhân phải mở cửa phòng nào để được tha tội và được cưới công chúa.*

## e) Phiên bản 5 (2 phòng).

- Có 2 phòng; Mỗi phòng hoặc là có công chúa hoặc là chứa hổ. Có thể cả hai phòng đều chứa hổ hay cả hai phòng đều có công chúa và có thể có một phòng có công chúa, một phòng chứa hổ.
  - Ở phòng 1: Nếu có công chúa thì biển báo của phòng là **đúng** nhưng nếu có hổ trong phòng thì biển báo của phòng là **sai**.
  - Ở phòng 2: Ngược lại với phòng 1, nếu có công chúa thì biển báo của phòng là **sai** nhưng nếu có hổ trong phòng thì biển báo của phòng là **đúng**.
  - Biển báo trên cửa phòng 1: Có ít nhất 1 phòng có công chúa.
  - Biển báo trên cửa phòng 2: Phòng kia có công chúa.
- ♡ *Anh tù nhân phải mở cửa phòng nào để được tha tội và được cưới công chúa.*

## f) Phiên bản 6 (2 phòng).

- Có 2 phòng; Mỗi phòng hoặc là có công chúa hoặc là chứa hổ. Có thể cả hai phòng đều chứa hổ hay cả hai phòng đều có công chúa và có thể có một phòng có công chúa, một phòng chứa hổ.
  - Ở phòng 1: Nếu có công chúa thì biển báo của phòng là **đúng** nhưng nếu có hổ trong phòng thì biển báo của phòng là **sai**.
  - Ở phòng 2: Ngược lại với phòng 1, nếu có công chúa thì biển báo của phòng là **sai** nhưng nếu có hổ trong phòng thì biển báo của phòng là **đúng**.
  - Biển báo trên cửa phòng 1: Không có sự khác biệt giữa 2 phòng.
  - Biển báo trên cửa phòng 2: Phòng kia có công chúa.
- ♡ *Anh tù nhân phải mở cửa phòng nào để được tha tội và được cưới công chúa.*

## g) Phiên bản 7 (2 phòng).

- Có 2 phòng; Mỗi phòng hoặc là có công chúa hoặc là chứa hổ. Có thể cả hai phòng đều chứa hổ hay cả hai phòng đều có công chúa và có thể có một phòng có công chúa, một phòng chứa hổ.
  - Ở phòng 1: Nếu có công chúa thì biển báo của phòng là **đúng** nhưng nếu có hổ trong phòng thì biển báo của phòng là **sai**.
  - Ở phòng 2: Ngược lại với phòng 1, nếu có công chúa thì biển báo của phòng là **sai** nhưng nếu có hổ trong phòng thì biển báo của phòng là **đúng**.
  - Biển báo trên cửa phòng 1: Có sự khác biệt giữa 2 phòng.
  - Biển báo trên cửa phòng 2: Bạn nên chọn phòng khác thì tốt hơn.
- ♡ *Anh tù nhân phải mở cửa phòng nào để được tha tội và được cưới công chúa.*

## h) Phiên bản 8 (3 phòng).

- Có 3 phòng; Một phòng có công chúa còn 2 phòng kia chứa hổ.
- Nhiều nhất là 1 phòng có biển báo đúng.
- Biển báo trên cửa phòng 1: Hổ có trong phòng này.
- Biển báo trên cửa phòng 2: Công chúa ở phòng này.
- Biển báo trên cửa phòng 3: Hổ ở phòng số 2.

♡ *Anh tù nhân phải mở cửa phòng nào để được tha tội và được cưới công chúa.*

## i) Phiên bản 9 (3 phòng).

- Có 3 phòng; Một phòng có công chúa còn 2 phòng kia chứa hổ.
- Biển báo ở phòng có công chúa là đúng và có ít nhất 1 biển báo của 2 phòng còn lại là sai.
- Biển báo trên cửa phòng 1: Hổ có trong phòng số 2.
- Biển báo trên cửa phòng 2: Hổ có trong phòng này.
- Biển báo trên cửa phòng 3: Hổ ở phòng số 1.

♡ *Anh tù nhân phải mở cửa phòng nào để được tha tội và được cưới công chúa.*

## j) Phiên bản 10 (3 phòng).

- Có 3 phòng; Một phòng có công chúa, một phòng khác có hổ và phòng còn lại trống.
- Biển báo ở phòng có công chúa là đúng; Biển báo ở phòng có hổ là sai và biển báo ở phòng trống thì có thể đúng hoặc sai.
- Biển báo trên cửa phòng 1: Phòng số 3 trống.
- Biển báo trên cửa phòng 2: Hổ có trong phòng số 1.
- Biển báo trên cửa phòng 3: Phòng này trống.

♡ *Anh tù nhân phải mở cửa phòng nào để được tha tội và được cưới công chúa.*

k) Phiên bản 11 (**9 phòng**).

- Có 3 phòng; Một phòng có công chúa, các phòng khác có thể có hổ hoặc trống.
- Biển báo ở phòng có công chúa là đúng; Biển báo ở phòng có hổ là sai và biển báo ở phòng trống thì có thể đúng hoặc sai.
- Biển báo trên cửa phòng 1: Công chúa ở phòng số lẻ.
- Biển báo trên cửa phòng 2: Phòng này trống.
- Biển báo trên cửa phòng 3: Biển báo ở phòng số 5 là đúng hay biển báo phòng số 7 là sai.
- Biển báo trên cửa phòng 4: Biển báo ở phòng số 1 là sai.
- Biển báo trên cửa phòng 5: Biển báo ở phòng số 2 hay biển báo ở phòng số 4 là đúng.
- Biển báo trên cửa phòng 6: Biển báo ở phòng số 3 là sai.
- Biển báo trên cửa phòng 7: Công chúa không có trong phòng số 1.
- Biển báo trên cửa phòng 8: Phòng này chứa hổ và phòng số 9 thì trống.
- Biển báo trên cửa phòng 9: Phòng này chứa hổ và biển báo phòng số 6 thì sai.

♡ Anh tù nhân phải mở cửa phòng nào để được tha tội và được cưới công chúa.

### 3.15 What do the Potters dream of?

*Let's meet the Potters: Mummy, Daddy, their son Peter, a schoolboy, and his younger sister Betty. There is also Aunt Polly, who often visits them. Each family member has his or her hobbies, favorite desserts, and dreams!*

*Conditions of the puzzle:*

1. Mummy Potter attends yoga classes on Mondays and Thursdays.
2. A person loving ice cream dreams of visiting Paris.
3. Betty likes only marmalade.
4. Mummy eats only marshmallows.
5. The Potters have three money boxes for their dreams at their home: one for a trip to the sea, one for a ticket to the Swan Lake ballet, one for a new album for the collection of coins.
6. Aunt Polly has a sewing machine and a collection of sewing materials at home. She made a ballet suit for Betty for her classes.
7. Peter often goes fishing with his dad, but he quickly becomes bored of it and begins to walk down the shore looking for rare coins for his collection.
8. Peter doesn't like anything with cream.
9. Peter and Betty's parents have made the same New Year wish both.
10. On holidays, Mummy prepares the family's favorite desserts: Napoleon cake, marmalade, and waffles.

*Fill in the table and say who likes the Napoleon cake and who dreams of going to Paris.*

Person	Daddy	Mummy	Peter	Betty	Aunt Polly
Hobby					
Dessert					
Dreams					

### 3.16 One Evening

Once Donald Smith came to visit his schoolmate Bill Simpson to do their homework together and learn William Shakespeare's sonnets. When he entered Simpsons' apartment, he could see five pairs of shoes: sandals, office shoes, lace-up boots, and two pairs of sneakers.

Try to guess which room each of the Simpsons' family members (Bill, Mummy, Daddy, Granny, and Bill's sister Melanie) is in, what they do, and what kind of shoes each of them was wearing outdoors before coming home.

Conditions of the puzzle:

1. Daddy is in the kitchen alone.
2. Melanie is preparing for her philosophy college exam.
3. Granny prefers to wear open-toe shoes.
4. A person preparing dinner is also watching the news on TV.
5. There are several things being read by the Simpsons' at the moment: a philosophy textbook, a newspaper, and a book of Shakespeare's sonnets.
6. There is no one in the bedroom.
7. Daddy has just returned home from work. He is a civil servant and wears business-style clothes and shoes.
8. Mummy is looking after flowers.
9. There are three people in the living room.
10. Bill has brought a book to the children's room.
11. Mummy and Melanie came after jogging an hour ago.

Fill solution in the table:

Person	Daddy	Mummy	Granny	Melanie	Bill
Shoes					
Doings					
Rooms					

### 3.17 Granny's Birthday

*Granny Simpson's birthday is coming soon. The rest of the family have prepared gifts for her, many of them at the same time! Granny will have a birthday show, bouquets from everyone, and many tasty things for her anniversary.*

*What gifts will Granny receive? Read the tips and find out the answer!*

Conditions of the puzzle:

1. The Simpsons are preparing a concert show with tricks, a guitar song, and one family member's own poem.
2. A person who wrote the poem will give a bread machine as a present and buy irises.
3. Mummy will buy tulips.
4. Melanie has learned to bake cinnamon buns and remembered guitar chords for Granny's birthday.
5. The trickster has prepared a notebook for recipes and a fruit salad.
6. Melanie knows that Granny likes daisies.
7. A person who will give a rocking chair will also prepare homemade candies for Granny.
8. Bill has a special deck of cards and a box with a double bottom.
9. Mummy and Melanie have had rehearsals for two for a week.
10. Daddy will prepare orange juice.
11. Granny will also be given a bouquet of roses and a ticket to the theatre play.

*Who will give a theatre ticket and who will buy roses?*

Person	Bill	Melanie	Daddy	Mummy
Gift				
Show				
Flower for a bouquet				
Tasty thing				

### 3.18 What to wear

*Ben and Mary spent time at a health camp from Monday till Friday. Each of them took such clothes as shorts, jeans, a zipped jacket, a polo, and a baseball cap.*

*What did the kids wear every day at the camp? It is known that they haven't changed their clothes during the day.*

Conditions of the puzzle:

1. Ben was wearing the same clothes as Mary twice — on Wednesday and Friday. It was very hot on Wednesday, so they wore the lightest clothes they had.
2. They both wore jeans on Monday.
3. Mary wore the same clothes on Thursday and Friday.
4. Each of them wore a baseball cap once on different days of the week with even numbers.
5. Mary wore a zipped jacket on Monday, Thursday, and Friday.
6. On Tuesday, Ben wore a zipped jacket and jeans. Overall, Ben wore jeans three times, and he also wore a zipped jacket twice.
7. Mary didn't put on her baseball cap on Thursday.
8. On departure day, Mary put on her zipped jacket and jeans.
9. Mary wore jeans one day more than shorts.

Fill solution in the table:

Person	Ben	Mary
Monday		
Tuesday		
Wednesday		
Thursday		
Friday		

### 3.19 Einstein Riddle

The legend says that this problem was created by Albert Einstein in the last century. Einstein said that only 2% of the world could solve it.

There are five houses of different colors next to each other. In each house lives a man. Each man has a unique nationality, an exclusive favorite drink, a distinct favorite brand of cigarettes and keeps specific pets.

1. The Brit lives in the Red house.
2. The Swede keeps Dogs as pets.
3. The Dane drinks Tea.
4. The Green house is exactly to the left of the White house.
5. The owner of the Green house drinks Coffee.
6. The person who smokes Pall Mall rears Birds.
7. The owner of the Yellow house smokes Dunhill.
8. The man living in the centre house drinks Milk.
9. The Norwegian lives in the first house.
10. The man who smokes Blends lives next to the one who keeps Cats.
11. The man who keeps Horses lives next to the man who smokes Dunhill.
12. The man who smokes Blue Master drinks Beer.
13. The German smokes Prince.
14. The Norwegian lives next to the Blue house.
15. The man who smokes Blends has a neighbour who drinks Water.

#### Who owns the fish?

## 4 First Order Logic

- 4.1** Consider a knowledge base containing just two sentences:  $P(a)$  and  $P(b)$ .

Does this knowledge base entail  $\forall x P(x)$ ?

Explain your answer in terms of models.

- 4.2** Which of the following are valid (necessarily true) sentences?

- a)  $(\exists x x = x) \Rightarrow (\forall y \exists z y = z)$
- b)  $\forall x P(x) \vee \neg P(x)$
- c)  $\forall x Smart(x) \vee (x = x)$

**4.3** Does the fact  $\neg \text{Spouse}(\text{George}, \text{Laura})$  follow from the facts  $\text{Jim} = \text{George}$  and  $\text{Spouse}(\text{Jim}, \text{Laura})$ ?

If so, give a proof; if not, supply additional axioms as needed. What happens if we use Spouse as a unary function symbol instead of a binary predicate?

**4.4** Predicates:  $\text{Mapcolor}$ ,  $\text{In}(x, y)$ ,  $\text{Borders}(x, y)$  and  $\text{Country}(x)$

For each of the logical expressions, state whether it:

- Correctly expresses the English sentence.
- Is syntactically invalid and therefore meaningless.
- Is syntactically valid but does not express the meaning of the English sentence.

a) Paris and Marseilles are both in France.

1.  $\text{In}(\text{Paris} \wedge \text{Marseilles}, \text{France})$
2.  $\text{In}(\text{Paris}, \text{France}) \wedge \text{In}(\text{Marseilles}, \text{France})$
3.  $\text{In}(\text{Paris}, \text{France}) \vee \text{In}(\text{Marseilles}, \text{France})$

b) There is a country that borders both Iraq and Pakistan.

1.  $\exists c \text{Country}(c) \wedge \text{Border}(c, \text{Iraq}) \wedge \text{Border}(c, \text{Pakistan})$
2.  $\exists c \text{Country}(c) \Rightarrow [\text{Border}(c, \text{Iraq}) \wedge \text{Border}(c, \text{Pakistan})]$
3.  $[\exists c \text{Country}(c)] \Rightarrow [\text{Border}(c, \text{Iraq}) \wedge \text{Border}(c, \text{Pakistan})]$
4.  $\exists c \text{Border}(\text{Country}(c), \text{Iraq} \wedge \text{Pakistan})$

c) All countries that border Ecuador are in South America.

1.  $\forall c \text{Country}(c) \wedge \text{Border}(c, \text{Ecuador}) \Rightarrow \text{In}(c, \text{SouthAmerica})$
2.  $\forall c \text{Country}(c) \Rightarrow [\text{Border}(c, \text{Ecuador}) \Rightarrow \text{In}(c, \text{SouthAmerica})]$
3.  $\forall c [\text{Country}(c) \Rightarrow \text{Border}(c, \text{Ecuador})] \Rightarrow \text{In}(c, \text{SouthAmerica})$
4.  $\forall c \text{Country}(c) \wedge \text{Border}(c, \text{Ecuador}) \wedge \text{In}(c, \text{SouthAmerica})$

d) No region in South America borders any region in Europe.

1.  $\neg [\exists c, d \text{In}(c, \text{SouthAmerica}) \wedge \text{In}(d, \text{Europe}) \wedge \text{Borders}(c, d)]$
2.  $\forall c, d [\text{In}(c, \text{SouthAmerica}) \wedge \text{In}(d, \text{Europe})] \Rightarrow \neg \text{Borders}(c, d)$
3.  $\neg \forall c \text{In}(c, \text{SouthAmerica}) \Rightarrow \exists d \text{In}(d, \text{Europe}) \wedge \neg \text{Borders}(c, d)$
4.  $\forall c \text{In}(c, \text{SouthAmerica}) \Rightarrow \forall d \text{In}(d, \text{Europe}) \Rightarrow \neg \text{Borders}(c, d)$

e) No two adjacent countries have the same map color.

1.  $\forall x, y \neg \text{Country}(x) \vee \neg \text{Country}(y) \vee \neg \text{Borders}(x, y) \vee \neg (\text{MapColor}(x) = \text{MapColor}(y))$
2.  $\forall x, y (\text{Country}(x) \wedge \text{Country}(y) \wedge \text{Borders}(x, y) \wedge \neg(x = y)) \Rightarrow \neg (\text{MapColor}(x) = \text{MapColor}(y))$
3.  $\forall x, y \text{Country}(x) \wedge \text{Country}(y) \wedge \text{Borders}(x, y) \wedge \neg (\text{MapColor}(x) = \text{MapColor}(y))$
4.  $\forall x, y (\text{Country}(x) \wedge \text{Country}(y) \wedge \text{Borders}(x, y)) \Rightarrow \text{MapColor}(x \neq y)$

## 5 Inference in First Order Logic

### 5.1 Cây phả hệ

- Cho các sự kiện sau:

$\checkmark \text{Parent(John, Mary)}$  (John là cha của Mary)  
 $\checkmark \text{Parent(Mary, Peter)}$  (Mary là mẹ của Peter)  
 $\checkmark \forall x \forall y (\text{Parent}(x, y) \rightarrow \text{Ancestor}(x, y))$  (Nếu x là cha/mẹ của y thì x là tổ tiên của y)  
 $\checkmark \forall x \forall y \forall z (\text{Ancestor}(x, y) \wedge \text{Parent}(y, z) \rightarrow \text{Ancestor}(x, z))$  (Nếu x là tổ tiên của y và y là cha/mẹ của z thì x là tổ tiên của z)

- Hỏi: Có thể suy ra được  $\text{Ancestor}(\text{John}, \text{Peter})$  (John là tổ tiên của Peter) hay không? Giải thích.

### 5.2 Sử dụng Forward Chaining

- Cho cơ sở tri thức (Knowledge Base - KB) sau:

1.  $\forall x (A(x) \rightarrow B(x))$
2.  $\forall x (B(x) \wedge C(x) \rightarrow D(x))$
3.  $A(\text{John})$
4.  $C(\text{John})$

- Áp dụng thuật toán Forward Chaining để suy ra các mệnh đề mới từ KB.

### 5.3 Chuyển đổi sang dạng chuẩn tắc (Clausal Form)

- Chuyển đổi câu sau sang dạng chuẩn tắc:

$$\forall x (P(x) \rightarrow \exists y (Q(x, y) \wedge R(y)))$$

### 5.4 Chứng minh bằng Resolution

- Cho các mệnh đề sau:

$$\begin{aligned} &\checkmark P \rightarrow Q \\ &\checkmark \neg Q \vee R \\ &\checkmark P \end{aligned}$$

- Chứng minh R bằng phương pháp Resolution.

### 5.5 Kiểm tra tính hợp lệ

- Cho cơ sở tri thức KB như sau:

$\checkmark \forall x (\text{Bird}(x) \rightarrow \text{CanFly}(x))$  (Mọi con chim đều có thể bay)  
 $\checkmark \text{Penguin(Tweety)} \wedge \text{Bird(Tweety)}$  (Tweety là chim cánh cụt và là một con chim)  
 $\checkmark \forall x (\text{Penguin}(x) \rightarrow \neg \text{CanFly}(x))$  (Chim cánh cụt không thể bay)

- Hỏi: KB có suy ra được  $\neg \text{CanFly}(\text{Tweety})$  (Tweety không thể bay) hay không? Giải thích bằng cách sử dụng các quy tắc suy luận (ví dụ: Modus Ponens, Universal Elimination).

### 5.6 Các hiệp sĩ và người hùng

- Một hòn đảo có hai loại cư dân: hiệp sĩ (luôn nói thật) và người hùng (luôn nói dối). Bạn gặp hai người tên A và B.

✓ A nói rằng: "Hoặc tôi là người hùng, hoặc B là hiệp sĩ."

- Hỏi A và B là hiệp sĩ hay người hùng?

### 5.7 Ba vị thần

- Có ba vị thần: Chân (luôn nói thật), Dối (luôn nói dối), và Ngẫu nhiên (nói thật hoặc dối ngẫu nhiên).
- Bạn được phép hỏi mỗi vị thần một câu hỏi có/không.
- Hỏi làm thế nào để xác định được danh tính của từng vị thần?

### 5.8 Chứng minh bằng Resolution

- Cho các mệnh đề sau:

1.  $\forall x(P(x) \rightarrow Q(x))$
2.  $\forall x(Q(x) \rightarrow R(x))$
3.  $\exists xP(x)$
4.  $\neg R(a)$

- Chứng minh rằng mệnh đề trên là mâu thuẫn bằng phương pháp Resolution.

### 5.9 Các hộp quà

- Có ba hộp quà:

✓ Một hộp chứa hai đồng vàng,  
 ✓ Một hộp chứa một đồng vàng và một đồng bạc,  
 ✓ Và một hộp chứa hai đồng bạc.

- Các hộp được dán nhãn sai.
- Bạn được phép lấy một đồng xu từ một hộp bất kỳ.
- Hỏi làm thế nào để xác định được nội dung của mỗi hộp?

## 6 Leaner Regression

### 6.1 Consider the following set of training examples:

$$X = \{1, 3, 2, 4, 3, 6\};$$

$$y = \{1.50, 2, 3.50, 2.50, 1, 1.50\}$$

a) Calculate:  $\beta$

b) Predict:  $x = 2.50$

### 6.2 Consider the following set of training examples:

$$X = \{(1, 0), (2, 3.50), (2.50, 4), (3.50, 4), (3, 1), (5, 1.50)\};$$

$$y = \{1.50, 2, 3.50, 2.50, 1, 1.50\}$$

a) Calculate:  $\beta$

b) Predict:  $x = (1.50, 4)$

**6.3** Consider the following set of training examples:

$X = \{(1, 0, 1), (2, 3.50, 3), (2.50, 4, 1.50), (3.50, 2.50, 4), (3, 1, 1.50), (5, 2.50, 1.50)\};$

$y = \{1.50, 2, 3.50, 2.50, 1, 1.50\}$

a) Calculate:  $\beta$

b) Predict:  $x = (1.50, 4.50, 3.50)$

**6.4 House Price Prediction****Mô tả:**

Sử dụng Linear Regression để dự đoán giá nhà dựa trên các yếu tố như diện tích, số phòng, vị trí, v.v.

**Dữ liệu:**

Tập dữ liệu có thể được thu thập từ các trang web bất động sản, từ cơ quan quản lý bất động sản hoặc từ các dịch vụ môi giới bất động sản. Dữ liệu này bao gồm các thông tin như diện tích, số phòng, vị trí, tiện ích xung quanh và giá bán thực tế.

**6.5 Advertising Sales Prediction****Mô tả:**

Sử dụng Linear Regression để dự đoán doanh số bán hàng dựa trên các chiến lược quảng cáo.

**Dữ liệu:**

Tập dữ liệu có thể được thu thập từ các chiến dịch quảng cáo trước đó, từ các công ty quảng cáo hoặc từ các công ty phân tích thị trường.

**6.6 Student Performance Prediction****Mô tả:**

Sử dụng Linear Regression để dự đoán hiệu suất học tập của sinh viên dựa trên các yếu tố như thời gian học, số giờ ngủ, điểm số trước đó, v.v.

**Dữ liệu:**

Tập dữ liệu có thể được thu thập từ các học kỳ trước, từ các hệ thống quản lý học tập trường đại học hoặc từ các cuộc khảo sát sinh viên. Dữ liệu này bao gồm các thông tin về sinh viên như thời gian học, số giờ ngủ, điểm số trước đó và điểm cuối kỳ.

**6.7 Stock Price Prediction****Mô tả:**

Sử dụng Linear Regression để dự đoán giá cổ phiếu của một công ty dựa trên các yếu tố như giá trị của các chỉ số tài chính, tin tức thị trường, và dữ liệu lịch sử giá cổ phiếu.

**Dữ liệu:**

Tập dữ liệu có thể được thu thập từ các trang web tài chính như Yahoo Finance hoặc Alpha Vantage, hoặc từ các công ty dịch vụ dữ liệu tài chính như Bloomberg. Dữ liệu này bao gồm các chỉ số tài chính như P/E ratio, EPS, lợi nhuận ròng, tin tức thị trường và dữ liệu lịch sử giá cổ phiếu.

**6.8 Traffic Volume Prediction**

**Mô tả:**

Sử dụng Linear Regression để dự đoán lưu lượng giao thông trên một tuyến đường cụ thể dựa trên các yếu tố như thời tiết, giờ trong ngày, ngày trong tuần, và các sự kiện đặc biệt.

**Dữ liệu:**

Tập dữ liệu có thể được thu thập từ các hệ thống giao thông hoặc từ các tổ chức quản lý giao thông. Dữ liệu này bao gồm các thông tin như thời tiết (nhiệt độ, độ ẩm, tốc độ gió), giờ trong ngày, ngày trong tuần và lưu lượng giao thông thực tế được đo lường từ cảm biến hoặc camera giao thông.

**6.9 Retail Sales Prediction****Mô tả:**

Sử dụng Linear Regression để dự đoán doanh số bán lẻ của một cửa hàng hoặc chuỗi cửa hàng dựa trên các yếu tố như thời gian, quảng cáo, giá cả, v.v.

**Dữ liệu:**

Tập dữ liệu có thể được thu thập từ các hệ thống bán lẻ hoặc từ các công ty phân tích thị trường. Dữ liệu này bao gồm các thông tin về doanh số bán lẻ hàng ngày, hàng tuần hoặc hàng tháng và các yếu tố như quảng cáo, giá cả, sự kiện đặc biệt, v.v.

**6.10 Crop Yield Prediction****Mô tả:**

Sử dụng Linear Regression để dự đoán sản lượng của một loại nông sản (ví dụ: lúa, hành, cà phê) dựa trên các yếu tố như điều kiện thời tiết, loại đất, phân bón, v.v.

**Dữ liệu:**

Tập dữ liệu có thể được thu thập từ các trang web nông nghiệp, cơ quan nông nghiệp hoặc từ các nghiên cứu nông nghiệp. Dữ liệu này bao gồm các thông tin về điều kiện thời tiết (nhiệt độ, lượng mưa, độ ẩm), loại đất, loại cây trồng, lượng phân bón sử dụng và sản lượng thực tế.

**Ghi chú:**

Trong Python, thư viện phổ biến để triển khai các mô hình hồi quy tuyến tính (linear regression) là scikit-learn. Thư viện này cung cấp một lớp LinearRegression để xây dựng và huấn luyện mô hình hồi quy tuyến tính.

## 7 Decision Tree

**7.1 Calculate and draw Decision Tree Regressor with **max depth = 3**:**

- a)  $X = \{ 2, 1, 3, 5 \} ; y = \{ 1.5, 2.5, 3, 2 \}$
- b)  $X = \{ 2.5, 1.5, 2, 3, 3.5 \} ; y = \{ 1, 2.5, 3, 2.5, 1.5 \}$
- c)  $X = \{ (2, 5), (1, 5), (2, 0), (0, 3), (3, 5) \} ; y = \{ 1, 1.5, 3, 3.5, 1.5 \}$
- d)  $X = \{ (1, 2, 3), (0, 1, 2), (2, 4, 1) \} ; y = \{ 1, 1.5, 2 \}$

**7.2 Calculate and draw Decision Tree Classifier with **max depth = 3**:**

- a)  $X = \{ 0, 1, 0, 2 \} ; y = \{ 2, 0, 1, 1 \}$
- b)  $X = \{ (2,0), (1,2), (0,1), (2,1) \} ; y = \{ 1, 0, 1, 2 \}$
- c)  $X = \{ (2,0,1), (0,1,2), (2,2,1) \} ; y = \{ 1, 0, 2 \}$
- d)  $X = \{ (2,1,1), (1,1,0), (2,0,1), (0,0,1) \} ; y = \{ 1, 0, 2, 1 \}$

### 7.3 Iris Flower Classification

#### Mô tả:

Sử dụng Decision Tree để phân loại loài hoa Iris dựa trên các đặc điểm hình học của hoa như chiều dài và chiều rộng của cánh hoa và đài hoa.

#### Dữ liệu:

Tập dữ liệu Iris là một tập dữ liệu phổ biến trong Machine Learning, chứa thông tin về ba loài hoa Iris: Iris Setosa, Iris Versicolor và Iris Virginica. Mỗi loài có 50 mẫu, với các đặc điểm như chiều dài và chiều rộng của cánh hoa và đài hoa.

Tập dữ liệu Iris có sẵn trong thư viện scikit-learn, có thể sử dụng hàm load\_iris() để tải tập dữ liệu này.

### 7.4 Titanic Survival Prediction

#### Mô tả:

Trong thảm họa Titanic, có nhiều yếu tố ảnh hưởng đến khả năng sống sót của hành khách, bao gồm giới tính, tuổi tác, loại vé, v.v. Sử dụng Decision Tree để dự đoán xem một hành khách có sống sót hay không dựa trên các thông tin như giới tính, tuổi tác, loại vé, và nơi lên tàu.

#### Dữ liệu:

Tập dữ liệu Titanic có sẵn và được sử dụng rộng rãi trong cộng đồng Machine Learning. Có thể tìm thấy tập dữ liệu này trên nhiều trang web, bao gồm Kaggle. Tập dữ liệu này bao gồm các thông tin như: PassengerId, Survived (0: không sống sót, 1: sống sót), Pclass (loại vé), Sex (giới tính), Age (tuổi tác), SibSp (số lượng anh chị em hoặc vợ/chồng đi cùng), Parch (số lượng bố mẹ hoặc con cái đi cùng), Fare (giá vé), Embarked (nơi lên tàu).

### 7.5 Weather prediction

#### Mô tả:

Sử dụng Decision Tree để dự đoán điều kiện thời tiết dựa trên các yếu tố khác nhau như nhiệt độ, độ ẩm, áp suất không khí và tốc độ gió.

#### Dữ liệu:

Có thể sử dụng dữ liệu thời tiết lịch sử từ các trang web hoặc API dịch vụ thời tiết như OpenWeatherMap. Dữ liệu này bao gồm các yếu tố thời tiết như nhiệt độ, độ ẩm, áp suất không khí, tốc độ gió, v.v. tại mỗi thời điểm.

### 7.6 Medical Diagnosis

#### Mô tả:

Sử dụng Decision Tree để chẩn đoán các bệnh y tế dựa trên các biểu hiện lâm sàng và kết quả các xét nghiệm.

#### Dữ liệu:

Tập dữ liệu có thể được thu thập từ các cơ sở dữ liệu y tế hoặc từ các nghiên cứu y học. Dữ liệu này bao gồm các đặc điểm lâm sàng của bệnh nhân như tuổi, giới tính, triệu chứng, kết quả xét nghiệm, lịch sử y tế và một cột nhãn cho biết

liệu bệnh nhân có mắc bệnh hay không.

### 7.7 Sentiment Analysis

#### Mô tả:

Sử dụng Decision Tree để phân loại văn bản vào các nhóm cảm xúc khác nhau như tích cực, tiêu cực hoặc trung tính.

#### Dữ liệu:

Tập dữ liệu có thể được thu thập từ các nguồn như trang web đánh giá sản phẩm, các diễn đàn trực tuyến hoặc các mạng xã hội. Dữ liệu này bao gồm các văn bản và một nhãn cho biết cảm xúc của mỗi văn bản (ví dụ: positive, negative, neutral).

Có thể sử dụng các kỹ thuật như TF-IDF hoặc Word Embedding để biểu diễn văn bản và làm cho mô hình hiểu được nội dung của văn bản.

### 7.8 Customer Segmentation

#### Mô tả:

Sử dụng Decision Tree để phân loại khách hàng thành các nhóm khác nhau dựa trên hành vi mua hàng, thông tin địa lý, đặc điểm cá nhân, v.v.

### 7.9 Wine Quality Prediction

#### Mô tả:

Sử dụng Decision Tree để dự đoán chất lượng của rượu dựa trên các yếu tố hóa học như nồng độ cồn, axit, độ pH, và các thuộc tính khác.

#### Dữ liệu:

Tập dữ liệu có thể được thu thập từ các nguồn như UCI Machine Learning Repository hoặc từ các tổ chức nghiên cứu rượu. Dữ liệu này bao gồm các đặc điểm hóa học của rượu như nồng độ cồn, axit, độ pH, tỷ lệ đường, v.v. và một điểm số hoặc nhãn cho biết chất lượng của rượu.

#### Ghi chú:

Trong Python, thư viện phổ biến để triển khai Decision Tree là scikit-learn. Thư viện này cung cấp một lớp `DecisionTreeClassifier` để xây dựng và huấn luyện mô hình Decision Tree cho bài toán phân loại, và một lớp `DecisionTreeRegressor` cho bài toán hồi quy.

## 8 Naive Bayes Classifier (NBC) and Support Vector Machine (SVM)

### 8.1 Email Spam Detection

#### Mô tả:

Sử dụng NBC và SVM để phân loại các email thành hai loại: "spam" và "không phải spam". Chúng ta sẽ xây dựng một mô hình dựa trên các đặc điểm của email như từ khóa, độ dài của email, sự xuất hiện của các ký tự đặc biệt, v.v.

#### Dữ liệu:

Tập dữ liệu có thể được thu thập từ các nguồn như SpamAssassin Public Corpus hoặc Enron Email Dataset. Dữ liệu này bao gồm các email đã được gán nhãn là "spam" hoặc "không phải spam".

## 8.2 Text Classification

### Mô tả:

Sử dụng NBC và CVM để phân loại các văn bản vào các danh mục khác nhau dựa trên nội dung của chúng. Chẳng hạn, phân loại các bài báo, email hoặc tweet vào các danh mục như "Thể thao", "Chính trị", "Giải trí", v.v.

### Dữ liệu:

Tập dữ liệu có thể được thu thập từ các nguồn như trang tin tức, blog, hoặc từ các tập dữ liệu đã được gán nhãn. Dữ liệu này bao gồm các văn bản đã được phân loại vào các danh mục mong muốn.

## 8.3 Sentiment Analysis

### Mô tả:

Sử dụng NBC và CVM để phân tích cảm xúc của văn bản, tức là xác định xem một đoạn văn bản là tích cực, tiêu cực hoặc trung tính.

### Dữ liệu:

Tập dữ liệu có thể được thu thập từ các nguồn như trang web đánh giá sản phẩm, mạng xã hội, hoặc từ các tập dữ liệu đã được gán nhãn. Dữ liệu này bao gồm các văn bản (ví dụ: bài đánh giá sản phẩm, bình luận trên mạng xã hội) đã được gán nhãn là tích cực, tiêu cực hoặc trung tính.

## 8.4 Document Classification

### Mô tả:

Sử dụng NBC và CVM để phân loại các văn bản vào các danh mục khác nhau dựa trên nội dung của chúng. Chẳng hạn, phân loại các văn bản thành các danh mục như "Thể thao", "Chính trị", "Khoa học", "Giải trí", v.v.

### Dữ liệu:

Tập dữ liệu có thể được thu thập từ các nguồn như trang tin tức, blog, hoặc từ các tập dữ liệu đã được gán nhãn. Dữ liệu này bao gồm các văn bản đã được phân loại vào các danh mục mong muốn.

## 8.5 Weather Prediction

### Mô tả:

Sử dụng NBC và CVM để dự đoán thời tiết, tức là xác định điều kiện thời tiết như "nắng", "mưa", "âm u", v.v. dựa trên các yếu tố như nhiệt độ, độ ẩm, tốc độ gió, v.v.

### Dữ liệu:

Tập dữ liệu có thể được thu thập từ các trạm quan trắc thời tiết hoặc các nguồn dữ liệu thời tiết trực tuyến. Dữ liệu này bao gồm các thông tin như nhiệt độ, độ ẩm, tốc độ gió và điều kiện thời tiết thực tế.

## 8.6 Medical Diagnosis

### Mô tả:

Sử dụng NBC và CVM để hỗ trợ trong quá trình chẩn đoán y tế. Mục tiêu là dự đoán xem một bệnh nhân có mắc phải một loại bệnh cụ thể hay không dựa trên các dữ liệu y tế và các chỉ số sinh học.

### Dữ liệu:

Tập dữ liệu có thể được thu thập từ các bệnh viện hoặc cơ sở y tế. Dữ liệu này bao gồm các thông tin về các triệu chứng, kết quả xét nghiệm, lịch sử y tế và kết quả chẩn đoán của các bệnh nhân.

## 8.7 Product Recommendation

### Mô tả:

Sử dụng NBC và CVM để đề xuất sản phẩm cho người dùng dựa trên lịch sử mua hàng hoặc sở thích của họ. Mục tiêu là dự đoán xem một người dùng có quan tâm đến một sản phẩm cụ thể hay không dựa trên thông tin về sản phẩm và hành vi mua hàng của người dùng.

### Dữ liệu:

Tập dữ liệu có thể được thu thập từ các trang web thương mại điện tử hoặc cơ sở dữ liệu về lịch sử mua hàng của khách hàng. Dữ liệu này bao gồm thông tin về sản phẩm, hành vi mua hàng của người dùng (ví dụ: lịch sử mua hàng, xếp hạng sản phẩm, đánh giá), và các thông tin đặc trưng của người dùng (ví dụ: tuổi, giới tính, v.v.).

## 8.8 Breast Cancer test

### Mô tả:

Sử dụng NBC và CVM để phát hiện ung thư vú dựa trên các yếu tố y tế của một bệnh nhân, như kích thước khối u, độ đặc trưng, tình trạng của các nốt nhỏ, v.v.

### Dữ liệu:

Tập dữ liệu có thể được thu thập từ các cơ sở dữ liệu y tế hoặc bệnh viện. Dữ liệu này bao gồm các thông tin về kích thước khối u, đặc trưng của nó và kết quả kiểm tra xác nhận ung thư.

# 9 Rosenblatt's Perceptron

## 9.1 Phân loại Iris Dataset

### Mô tả:

Sử dụng Rosenblatt's Perceptron để phân loại các loài hoa Iris dựa trên các đặc trưng của chúng.

### Dữ liệu:

Tập dữ liệu Iris Dataset chứa các thông tin về kích thước cánh hoa và lá hoa của các loài Iris.

## 9.2 Phân loại điểm dữ liệu 2D

### Mô tả:

Tạo ra một tập dữ liệu đơn giản với hai đặc trưng và hai lớp tuyến tính, sau đó sử dụng Perceptron để phân loại chúng.

### Dữ liệu:

Tập dữ liệu được tạo tự động với hai đặc trưng và hai lớp tuyến tính.

## 9.3 Dự đoán Diabetic Patients

### Mô tả:

Sử dụng Rosenblatt's Perceptron để dự đoán liệu một bệnh nhân có mắc bệnh tiểu đường hay không dựa trên các chỉ số sinh học.

### Dữ liệu:

Tập dữ liệu chứa các thông tin về chỉ số sinh học của bệnh nhân và kết quả xác định có mắc tiểu đường hay không.

## 9.4 Phân loại Email Spam

**Mô tả:**

Sử dụng Perceptron để phân loại các email thành hai loại: spam và không phải spam, dựa trên nội dung của email.

**Dữ liệu:**

Tập dữ liệu chứa các email đã được gán nhãn là spam hoặc không spam.

**9.5 Dự đoán Phân loại Tumor****Mô tả:**

Sử dụng Rosenblatt's Perceptron để dự đoán xem một khối u là ác tính hay lành tính dựa trên các đặc trưng của nó.

**Dữ liệu:**

Tập dữ liệu chứa thông tin về kích thước và tính chất của khối u cùng với kết quả xác nhận liệu nó là ác tính hay lành tính.

## 10 Convolutional Neural Networks (CNN)

**10.1 Phân loại ảnh CIFAR-10****Mô tả:**

Sử dụng CNN để phân loại các ảnh từ tập dữ liệu CIFAR-10. CIFAR-10 là một tập dữ liệu phổ biến trong lĩnh vực máy học.

**Dữ liệu:**

Tập dữ liệu CIFAR-10 chứa 60,000 ảnh màu 32x32 pixel, được chia thành 10 lớp, mỗi lớp chứa 6,000 ảnh.

**10.2 Face recognition****Mô tả:**

Sử dụng CNN để nhận dạng và phân loại các khuôn mặt từ ảnh.

**Dữ liệu:**

Có thể sử dụng tập dữ liệu có sẵn như LFW (Labeled Faces in the Wild) hoặc CK+ (Cohn-Kanade Extended Facial Expression Dataset) để thực hiện bài tập này.

**10.3 Detect objects****Mô tả:**

Sử dụng CNN để phát hiện và định vị các đối tượng trong ảnh. Mục tiêu là xác định các vị trí của các đối tượng cụ thể trong một ảnh và vẽ các hộp giới hạn xung quanh chúng.

**Dữ liệu:**

Có thể sử dụng tập dữ liệu COCO (Common Objects in Context) hoặc VOC (Visual Object Classes) để thực hiện bài tập này. Cả hai tập dữ liệu này đều chứa các ảnh được gắn nhãn với các đối tượng cụ thể được định danh.

**10.4 Categorize dog and cat photos****Mô tả:**

Sử dụng CNN để phân loại ảnh chó và mèo từ tập dữ liệu được gắn nhãn.

**Dữ liệu:**

Có thể sử dụng tập dữ liệu có sẵn như tập dữ liệu "Dogs vs. Cats" trên Kaggle hoặc tập dữ liệu có sẵn trong TensorFlow datasets.

## 10.5 Classification of handwritten text images

### Mô tả:

Sử dụng CNN để phân loại các hình ảnh chứa chữ số viết tay từ tập dữ liệu MNIST hoặc EMNIST.

### Dữ liệu:

Có thể sử dụng tập dữ liệu MNIST (chứa các chữ số từ 0 đến 9) hoặc tập dữ liệu EMNIST (mở rộng của MNIST, bao gồm cả chữ cái và chữ số) để thực hiện bài tập này.

### Ghi chú:

Dưới đây là một số loại mạng CNN phổ biến:

1. **LeNet**: Là một trong những mạng CNN đầu tiên được phát triển bởi Yann LeCun cho bài toán nhận dạng ký tự trong dãy số, như mã vạch hoặc số viết tay.
2. **AlexNet**: Một trong những mạng CNN đầu tiên trở nên phổ biến sau khi giành chiến thắng trong cuộc thi ImageNet năm 2012. AlexNet sử dụng một kiến trúc sâu với nhiều lớp tích chập và lớp pooling.
3. **VGGNet**: Mạng CNN với một kiến trúc đơn giản nhưng sâu, bao gồm nhiều lớp tích chập và lớp pooling. VGGNet nổi tiếng với kiến trúc đồng nhất, mỗi lớp tích chập có kernel size là 3x3 và stride là 1.
4. **GoogLeNet (InceptionNet)**: Sử dụng một kiến trúc phức tạp với nhiều modules Inception, bao gồm các lớp tích chập với các kích thước kernel khác nhau và các lớp pooling.
5. **ResNet**: Được giới thiệu bởi Kaiming He et al., ResNet sử dụng kiến trúc "skip connections" để tránh vấn đề biến mất gradient trong quá trình huấn luyện mạng CNN sâu.
6. **MobileNet**: Mạng CNN được thiết kế đặc biệt để chạy trên các thiết bị di động và tài nguyên có hạn. MobileNet sử dụng các lớp tích chập Depthwise Separable Convolution để giảm lượng tính toán và số lượng tham số.
7. **Xception**: Một biến thể của GoogLeNet, Xception sử dụng các lớp tích chập depthwise separable convolution để giảm số lượng tham số và tăng hiệu suất mô hình.
8. **DenseNet**: Sử dụng kết nối mật độ (dense connections) giữa các lớp, mỗi lớp nhận đầu vào từ tất cả các lớp trước đó, giúp tăng tính khái quát và hiệu suất của mạng.
9. **EfficientNet**: Một dòng mạng CNN hiệu suất cao được tối ưu hóa cho cả hai yếu tố: hiệu suất và kích thước. EfficientNet sử dụng phương pháp tìm kiếm đa mục tiêu (compound scaling) để điều chỉnh đồng thời chiều sâu, chiều rộng và độ phân giải của mạng.
10. **Capsule Network (CapsNet)**: Một loại mạng CNN có khả năng tự động phát hiện và tái tạo cấu trúc không gian đa chiều của đối tượng trong hình ảnh, giúp cải thiện khả năng tự tin trong phân loại đối tượng.

Trong Python, một số thư viện phổ biến cho mạng Convolutional Neural Networks (CNNs) bao gồm:

1. **TensorFlow và Keras**: TensorFlow là một thư viện mã nguồn mở phát triển bởi Google, còn Keras là một API được tích hợp sẵn trong TensorFlow giúp xây dựng và huấn luyện mạng nơ-ron một cách dễ dàng và linh hoạt.

2. **PyTorch:** PyTorch là một thư viện máy học mã nguồn mở phát triển bởi Facebook, có cùng một cấu trúc với TensorFlow nhưng được yêu thích bởi sự linh hoạt và dễ sử dụng.
3. **Caffe:** Caffe là một thư viện mã nguồn mở cho việc phát triển các mô hình deep learning được phát triển bởi Caffe2 trước khi được hợp nhất vào PyTorch.
4. **MXNet:** MXNet là một thư viện deep learning mã nguồn mở và hiệu quả cao, được phát triển bởi dự án Apache.

## 11 Recurrent Neural Network (RNN)

### 11.1 Time Series Prediction

#### Mô tả:

Sử dụng mạng RNN để dự đoán các giá trị trong một chuỗi thời gian tiếp theo dựa trên dữ liệu lịch sử.

#### Dữ liệu:

Có thể sử dụng tập dữ liệu chuỗi thời gian như giá cổ phiếu, doanh số bán hàng, nhiệt độ, v.v.

### 11.2 Text classification

#### Mô tả:

Sử dụng mạng RNN để phân loại các đoạn văn bản vào các danh mục khác nhau, chẳng hạn như phân loại tin tức, phân loại cảm xúc hoặc phân loại ngôn ngữ.

#### Dữ liệu:

Có thể sử dụng các tập dữ liệu văn bản có sẵn như tập dữ liệu IMDb để phân loại cảm xúc của đánh giá phim, hoặc tập dữ liệu 20 newsgroups để phân loại các tin tức.

### 11.3 Machine translation

#### Mô tả:

Sử dụng mạng RNN dạng mã hóa - giải mã (Encoder-Decoder) để thực hiện dịch máy từ một ngôn ngữ sang ngôn ngữ khác.

#### Dữ liệu:

Có thể sử dụng các bộ dữ liệu dịch máy như bộ dữ liệu dịch Anh-Việt.

### 11.4 Create a title for the text

#### Mô tả:

Sử dụng một mạng RNN để tạo tiêu đề cho các đoạn văn bản. Mục tiêu là tạo ra một tiêu đề ngắn, hấp dẫn và mô tả nội dung của văn bản.

#### Dữ liệu:

Có thể sử dụng các tập dữ liệu văn bản có sẵn hoặc thu thập dữ liệu từ các nguồn trực tuyến như các trang web tin tức hoặc các blog.

#### Ghi chú:

Dưới đây là một số loại mạng RNN phổ biến:

1. **Vanilla RNN (Basic RNN)**: Là mạng RNN cơ bản với các đơn vị chuỗi đơn giản. Tuy nhiên, nó thường gặp vấn đề về việc truyền ngược gradient qua nhiều bước thời gian.
2. **LSTM (Long Short-Term Memory)**: LSTM giải quyết vấn đề của sự biến mất gradient trong RNN thông thường bằng cách sử dụng các cổng (gates) để kiểm soát thông tin được lưu trữ và truyền qua thời gian.
3. **GRU (Gated Recurrent Unit)**: Tương tự như LSTM, GRU cũng giải quyết vấn đề biến mất gradient bằng cách sử dụng các cổng, nhưng nó có một kiến trúc đơn giản hơn bao gồm các cổng cập nhật và cổng khôi phục.
4. **Bidirectional RNN**: Mạng RNN này kết hợp hai lớp RNN chạy cùng một chuỗi, một theo chiều thuận và một theo chiều ngược. Điều này giúp mạng có khả năng hiểu ngữ cảnh từ cả hai phía của chuỗi.
5. **Deep RNN**: Một mạng RNN với nhiều lớp RNN xếp chồng lên nhau. Các lớp này có thể được kết nối theo cấu trúc nối tiếp hoặc kết nối ngược lại.
6. **Clockwork RNN**: Một dạng của RNN mà thời gian được chia thành các "kích hoạt" khác nhau cho từng phần của mạng, giúp giảm bớt độ phức tạp tính toán.
7. **Attention-based RNN**: Sử dụng cơ chế attention để tập trung vào các phần quan trọng của chuỗi đầu vào, giúp mạng RNN có thể xử lý các chuỗi có độ dài khác nhau một cách hiệu quả.
8. **Echo State Network (ESN)**: Một loại mạng RNN với cấu trúc thứ cấp đặc biệt, trong đó mạng nơ-ron được khởi tạo một cách ngẫu nhiên và chỉ các trọng số giữa lớp ẩn và lớp đầu ra được điều chỉnh trong quá trình huấn luyện.
9. **Hierarchical RNN**: Mạng RNN với kiến trúc phân cấp, trong đó mỗi lớp RNN được sử dụng để xử lý các cấp độ khác nhau của dữ liệu chuỗi.
10. **Recurrent Convolutional Neural Network (RCNN)**: Kết hợp giữa kiến trúc tích chập và kiến trúc RNN để xử lý dữ liệu chuỗi có cấu trúc lướt.

Trong Python, có một số thư viện phổ biến cho việc triển khai mạng Neural Networks (NNs) và cụ thể là mạng Recurrent Neural Networks (RNNs). Dưới đây là một số thư viện phổ biến và mạnh mẽ để xây dựng và huấn luyện mạng RNN:

1. **TensorFlow và Keras**: TensorFlow là một thư viện mã nguồn mở phát triển bởi Google, trong khi Keras là một API mạnh mẽ và dễ sử dụng được tích hợp sẵn trong TensorFlow. Keras cung cấp một cách tiếp cận dễ dàng cho việc xây dựng và huấn luyện mạng nơ-ron, bao gồm cả mạng RNNs.
2. **PyTorch**: PyTorch là một thư viện deep learning mã nguồn mở được phát triển bởi Facebook. Nó cung cấp một cấu trúc tương tự như TensorFlow nhưng thường được ưa chuộng bởi sự linh hoạt và dễ sử dụng hơn.
3. **MXNet**: MXNet là một thư viện deep learning hiệu suất cao được phát triển bởi Apache. Nó cung cấp một API để sử dụng cho việc xây dựng và huấn luyện mạng RNNs.
4. **Theano**: Theano là một thư viện deep learning mã nguồn mở cho Python, cho phép bạn định nghĩa, tối ưu hóa và đánh giá các mạng nơ-ron.

5. **CNTK (Microsoft Cognitive Toolkit)**: CNTK là một thư viện deep learning mã nguồn mở của Microsoft. Nó cung cấp một cách tiếp cận hiệu quả và mạnh mẽ cho việc xây dựng và huấn luyện mạng RNNs cũng như các loại mạng nơ-ron khác.

## 12 Generative Adversarial Network (GAN)

### 12.1 Create fake face photos

#### Mô tả:

Sử dụng một mạng GAN để tạo ra các hình ảnh khuôn mặt mới và chân thực từ một tập dữ liệu hình ảnh khuôn mặt đã có.

#### Dữ liệu:

Có thể sử dụng các tập dữ liệu khuôn mặt như CelebA hoặc LFW.

### 12.2 Create animated images from real photos

#### Mô tả:

Sử dụng một mạng GAN để chuyển đổi các hình ảnh thực tế thành ảnh hoạt hình hoặc vẽ tranh. Mục tiêu là tạo ra các hình ảnh mới với phong cách hoạt hình từ các bức ảnh thực tế.

#### Dữ liệu:

Có thể sử dụng bất kỳ tập dữ liệu hình ảnh nào chứa các đối tượng, cảnh quan hoặc bất kỳ đối tượng nào bạn muốn chuyển đổi thành hình ảnh hoạt hình.

#### Lưu ý:

Huấn luyện GAN để tạo ra hình ảnh hoạt hình từ hình ảnh thực tế có thể đòi hỏi nhiều tài nguyên tính toán và thời gian. Cần nhắc sử dụng mô hình đã được huấn luyện trước hoặc dịch vụ trên đám mây nếu gặp rắc rối về tài nguyên.

### 12.3 Create portrait photos from different styles

#### Mô tả:

Sử dụng một mạng GAN để tạo ra các ảnh chân dung từ các phong cách nghệ thuật khác nhau, chẳng hạn như phong cách của các họa sĩ nổi tiếng. Mục tiêu là tạo ra các ảnh chân dung mới với các đặc điểm và phong cách nghệ thuật khác nhau.

#### Dữ liệu:

Cần một tập dữ liệu ảnh chân dung chất lượng cao và đa dạng. Đảm bảo rằng tập dữ liệu bao gồm các bức ảnh chân dung từ các phong cách nghệ thuật khác nhau.

### 12.4 Create images of different moods

#### Mô tả:

Sử dụng một mạng GAN để tạo ra các ảnh miền tâm trạng khác nhau. Mục tiêu là tạo ra các hình ảnh thể hiện các tâm trạng khác nhau, chẳng hạn như vui vẻ, buồn bã, lo lắng, hạnh phúc, và nhiều hơn nữa.

#### Dữ liệu:

Có thể sử dụng các tập dữ liệu chứa các hình ảnh miền tâm trạng khác nhau hoặc các tập dữ liệu đa dạng chứa các hình ảnh con người có thể biểu lộ các tâm trạng khác nhau.

## 12.5 Create fake celebrity photos

### Mô tả:

Sử dụng một mạng GAN để tạo ra các ảnh giả mạo của những người nổi tiếng. Mục tiêu là tạo ra các hình ảnh mô phỏng những người nổi tiếng, dựa trên ảnh của họ đã được sử dụng để huấn luyện mô hình.

### Dữ liệu:

Cần một tập dữ liệu chứa các hình ảnh của những người nổi tiếng. Đảm bảo rằng tập dữ liệu bao gồm các hình ảnh có chất lượng cao và đa dạng của những người nổi tiếng khác nhau.

## 12.6 Create new documents automatically

### Mô tả:

Sử dụng một mạng GAN để tạo ra các đoạn văn bản mới tự động. Mục tiêu là tạo ra các đoạn văn bản có tính sáng tạo và độc đáo, giống như những gì có thể được viết bởi con người.

### Dữ liệu:

Có thể sử dụng các tập dữ liệu văn bản lớn, chẳng hạn như các tập dữ liệu từ sách, báo cáo, blog hoặc các tài liệu trên internet.

### Ghi chú:

Dưới đây là một số loại mạng GAN phổ biến:

1. **Vanilla GAN (GAN cơ bản)**: Là mô hình GAN ban đầu được giới thiệu bởi Ian Goodfellow và các đồng nghiệp. Nó bao gồm một mạng sinh (generator) và một mạng phân biệt (discriminator) cùng với hàm mất mát adversarial.
2. **DCGAN (Deep Convolutional GAN)**: Sử dụng mạng tích chập sâu trong cả generator và discriminator. DCGAN đã cải thiện khả năng sinh ra hình ảnh chất lượng cao.
3. **CGAN (Conditional GAN)**: Một phiên bản của GAN mà mạng sinh nhận thêm thông tin điều kiện, chẳng hạn như nhãn lớp, để tạo ra hình ảnh có điều kiện.
4. **WGAN (Wasserstein GAN)**: Sử dụng khoảng cách Wasserstein thay vì khoảng cách Jensen-Shannon để đo lường sự tương đồng giữa phân phối của dữ liệu thật và dữ liệu giả mạo, giúp cải thiện sự ổn định và chất lượng của mô hình.
5. **LSGAN (Least Squares GAN)**: Thay vì sử dụng hàm mất mát log likelihood, LSGAN sử dụng bình phương sai số giữa đầu ra của mạng phân biệt và giá trị đích mong muốn.
6. **CycleGAN**: Một dạng đặc biệt của GAN được sử dụng để chuyển đổi hình ảnh từ một miền sang một miền khác mà không cần cặp dữ liệu huấn luyện đồng bộ.
7. **StarGAN**: Mở rộng của CycleGAN cho phép chuyển đổi hình ảnh giữa nhiều lớp nhãn khác nhau trong một mạng duy nhất.
8. **StyleGAN**: Sử dụng kiến trúc hiệu quả cao để tạo ra hình ảnh kỹ thuật số với chất lượng cao và sự đa dạng lớn.
9. **BigGAN**: Một phiên bản của GAN sử dụng kiến trúc mạng sâu và kỹ thuật kiểm soát tương tự như Conditional GAN để tạo ra hình ảnh có độ phân giải cao và chất lượng tốt.

10. **Progressive GAN**: Một kiến trúc GAN mở rộng với việc huấn luyện từng bước một, bắt đầu từ hình ảnh thô và ngày càng tăng độ phân giải và chi tiết của hình ảnh được tạo ra.

Trong Python, có một số thư viện phổ biến để triển khai mạng Generative Adversarial Networks (GANs). Dưới đây là một số thư viện phổ biến cho mạng GAN:

1. **TensorFlow và Keras**: TensorFlow là một thư viện mã nguồn mở phát triển bởi Google, trong khi Keras là một API được tích hợp sẵn trong TensorFlow giúp xây dựng và huấn luyện mạng nơ-ron một cách dễ dàng và linh hoạt. Bạn có thể sử dụng TensorFlow và Keras để triển khai các kiến trúc GAN phổ biến như DCGAN (Deep Convolutional GAN).
2. **PyTorch**: PyTorch là một thư viện deep learning mã nguồn mở được phát triển bởi Facebook. Nó cung cấp một cấu trúc tương tự như TensorFlow nhưng thường được ưa chuộng bởi sự linh hoạt và dễ sử dụng hơn.
3. **TensorFlow Probability**: TensorFlow Probability là một thư viện TensorFlow dành cho xác suất và thống kê. Nó cung cấp một cách tiếp cận dễ dàng cho việc xây dựng các mô hình GAN probabilistic.
4. **Chainer**: Chainer là một thư viện deep learning mã nguồn mở dành cho Python và C++, cho phép bạn xây dựng các mô hình GAN và huấn luyện chúng một cách linh hoạt.