Practical No. 1

Aim: Design a bot using AIML.

Source Code:

Step 1: Create the XML file
Open the notepad, write the following code, and save it as std-startup.xml

```
<aiml version="1.0.1" encoding="UTF-8">
  <!-- std-startup.xml -->
  <!-- Category is an atomic AIML unit -->
  <category>
    <!-- Pattern to match in user input -->
    <!-- If user enters "LOAD AIML B" -->
    <pattern>LOAD AIML B</pattern>
    <!-- Template is the response to the pattern -->
    <!-- This learn an aiml file -->
    <template>
      <learn>basic_chat.aiml</learn>
      <!-- You can add more aiml files here -->
      <!--<learn>more_aiml.aiml</learn>-->
    </template>
  </category>
</aiml>
```

Step 2: Create the aiml file
Open the notepad, write the following code, and save it as basic_chat.aiml

```
<aiml version="1.0.1" encoding="UTF-8">
<!-- basic_chat.aiml -->

  <category>
    <pattern>HELLO</pattern>
    <template>
      Well, hello!
    </template>
  </category>

  <category>
    <pattern>WHAT ARE YOU</pattern>
    <template>
      I'm a bot, silly!
    </template>
  </category>


  <category>
    <pattern>MY NAME IS *</pattern>
    <template>
                        <set name = "username"><star/></set> is the nice name.
    </template>
  </category>

  <category>
    <pattern>I LIKE *</pattern>
    <template>
```

```
                    <set name = "liking"><star/></set> is also my favourite.
    </template>
  </category>

        <category>
    <pattern>MY DOG NAME IS *</pattern>
    <template>
                    THAT IS INTERESTING THAT YOU HAVE A DOG NAMED <set name =
"dog"><star/></set> .
    </template>
  </category>

  <category>
    <pattern>Bye</pattern>
    <template>
                    Bye!!! <get name = "username"/> Thanks for talking with me.
    </template>
  </category>
</aiml>
```

Step 3: Install aiml package

```
pip install aiml
or
pip3 install aiml
```

Step 4: Create chatbot.py file

```
Import aiml
# Create the kernel and learn AIML files
kernel = aiml.Kernel()
kernel.learn("std-startup.xml")
kernel.respond("load aiml b")

# Press CTRL-C to break this loop
while True:
 message = input("Enter your message to the bot: ")
 if message == "quit":
     break
 else:
     bot_response = kernel.respond(message)
     print(bot_response)
```

Output:

```
Loading std-startup.xml...done (0.00 seconds)
Loading basic_chat.aiml...done (0.00 seconds)
Enter your message to the bot: Hello
Well, hello!
Enter your message to the bot: what are you
I'm a bot, silly!
Enter your message to the bot: My name is Shraddha
Shraddha is the nice name.
Enter your message to the bot: My dog name is Pacey and Dawson
THAT IS INTERESTING THAT YOU HAVE A DOG NAMED Pacey and Dawson .
Enter your message to the bot: bye
Bye!!! Shraddha Thanks for talking with me.
Enter your message to the bot: quit
```

Reference:
https://www.devdungeon.com/content/ai-chat-bot-python-aiml
https://www.tutorialspoint.com/aiml/index.htm

Practical No. 2

Aim: Design an Expert system using AIML

Source Code:

Step 1: Create the XML file
Open the notepad, write the following code, and save it as std-startup1.xml

```
<aiml version="1.0.1" encoding="UTF-8">
  <!-- std-startup.xml -->
  <!-- Category is an atomic AIML unit -->
  <category>
    <!-- Pattern to match in user input -->
    <!-- If user enters "LOAD AIML B" -->
    <pattern>LOAD AIML B</pattern>
    <!-- Template is the response to the pattern -->
    <!-- This learn an aiml file -->
    <template>
      <learn>basic_chat.aiml</learn>
      <!-- You can add more aiml files here -->
      <!--<learn>more_aiml.aiml</learn>-->
    </template>
  </category>
</aiml>
```

Step 2: Create the aiml file
Open the notepad, write the following code, and save it as basic_chat1.aiml

```
<aiml version="1.0.1" encoding="UTF-8">
<!-- basic_chat.aiml -->
 <category>
    <pattern>HELLO</pattern>
    <template>
      WHAT WOULD YOU LIKE TO DISCUSS? : HEALTH, MOVIES
    </template>
 </category>
 <category>
    <pattern>MOVIES</pattern>
    <template>
      YES <set name = "topic">MOVIES</set>
    </template>
 </category>
 <category>
    <pattern>HEALTH</pattern>
    <template>
      YES <set name = "topic">HEALTH</set>
    </template>
 </category>
<topic name ="MOVIES">
                <category>
                        <pattern>*</pattern>
                        <template>
                                DO YOU LIKE COMEDY MOVIES?
                        </template>
                </category>
                <category>
```

```
                        <pattern>YES</pattern>
                        <template>
                                I TOO LIKE COMEDY MOVIES
                        </template>
                </category>
                <category>
                        <pattern>NO</pattern>
                        <template>
                                BUT I LIKE COMEDY MOVIES
                        </template>
                </category>
        </topic>
        <topic name ="HEALTH">
                <category>
                        <pattern>*</pattern>
                        <template>
                                DO YOU HAVE FEVER?
                        </template>
                </category>
                <category>
                        <pattern>YES</pattern>
                        <template>
                                PLEASE TAKE MEDICINES AND PROPER REST
                        </template>
                </category>
                <category>
                        <pattern>NO</pattern>
                        <template>
                                GO OUT FOR A WALK AND LISTEN MUSIC
                        </template>
                </category>
        </topic>
        <category>
                        <pattern>NICE TALKING TO YOU</pattern>
                        <template>
                                SAME HERE...!!
                        </template>
        </category>
</aiml>
```

Step 3: Install aiml package

```
pip install aiml
or
pip3 install aiml
```

Step 4: Create chatbot.py file

```
import aiml
# Create the kernel and learn AIML files
kernel = aiml.Kernel()
kernel.learn("std-startup1.xml")
kernel.respond("load aiml b")

# Press CTRL-C to break this loop
while True:
 message = input("Enter your message to the bot: ")
```

```
if message == "quit":
    break
else:
    bot_response = kernel.respond(message)
    print(bot_response)
```

Output:

```
Loading std-startup1.xml...done (0.00 seconds)
Loading basic_chat1.aiml...done (0.00 seconds)
Enter your message to the bot: hello
WHAT WOULD YOU LIKE TO DISCUSS? : HEALTH, MOVIES
Enter your message to the bot: health
YES HEALTH
Enter your message to the bot: i am feeling tired
DO YOU HAVE FEVER?
Enter your message to the bot: no
GO OUT FOR A WALK AND LISTEN MUSIC
Enter your message to the bot: movies
YES MOVIES
Enter your message to the bot: i love movies
DO YOU LIKE COMEDY MOVIES?
Enter your message to the bot: yes
I TOO LIKE COMEDY MOVIES
Enter your message to the bot: nice talking to you
SAME HERE...!!
Enter your message to the bot: quit
```

References:

https://www.devdungeon.com/content/ai-chat-bot-python-aiml
https://www.tutorialspoint.com/aiml/aiml_topic_tag.htm

Practical No. 3

Aim: Implement Bayes Theorem using Python

Source Code:

```
# calculate the probability of cancer patient and diagnostic test
 # calculate P(A|B) given P(A), P(B|A), P(B|not A)
def bayes_theorem(p_a, p_b_given_a, p_b_given_not_a):
        # calculate P(not A)
        not_a = 1 - p_a
        # calculate P(B)
        p_b = p_b_given_a * p_a + p_b_given_not_a * not_a
        # calculate P(A|B)
        p_a_given_b = (p_b_given_a * p_a) / p_b
        return p_a_given_b

# P(A)
p_a = 0.0002
# P(B|A)
p_b_given_a = 0.85
# P(B|not A)
p_b_given_not_a = 0.05
# calculate P(A|B)
result = bayes_theorem(p_a, p_b_given_a, p_b_given_not_a)
# summarize
print('P(A|B) = %.3f%%' % (result * 100))
```

Output:

```
P(A|B) = 0.339%
```

References:

https://machinelearningmastery.com/bayes-theorem-for-machine-learning/

Practical No. 4

Aim: Implement Conditional Probability and joint probability using Python

Source Code:

```
import enum, random
class Kid(enum.Enum):
   BOY = 0
   GIRL = 1
def random_kid() -> Kid:
   return random.choice([Kid.BOY, Kid.GIRL])
both_girls = 0
older_girl = 0
either_girl = 0
random.seed(0)
#conditional
for _ in range(10000):
   younger = random_kid()
   older = random_kid()
   if older == Kid.GIRL:
      older_girl += 1
   if older == Kid.GIRL and younger == Kid.GIRL:
      both_girls += 1
   if older == Kid.GIRL or younger == Kid.GIRL:
      either_girl += 1
print("older girl: ", older_girl)
print("both girl: ",both_girls)
print("either girl: ", either_girl)
print("P(both | older):", both_girls / older_girl)   # 0.5007089325501317
print("P(both | either):", both_girls / either_girl) # 0.3311897106109325
```

Output:

```
older girl:  4937
both girl:  2472
either girl:  7464
P(both | older): 0.5007089325501317
P(both | either): 0.3311897106109325
```

References:

https://towardsdatascience.com/conditional-probability-with-python-concepts-tables-code-c23ffe65d110

Practical No. 5

Aim: Write a program for to implement Rule based system. (Prolog)

Source Code:

```
go:-
   hypothesis(Disease),
   write('I believe that the patient have '),
   write(Disease),
   nl,
   write('TAKE CARE '),
   undo.

/*Hypothesis that should be tested*/
hypothesis(cold) :- cold, !.
hypothesis(flu) :- flu, !.
hypothesis(typhoid) :- typhoid, !.
hypothesis(measles) :- measles, !.
hypothesis(malaria) :- malaria, !.
hypothesis(unknown). /* no diagnosis*/

/*Hypothesis Identification Rules*/

cold :-
verify(headache),
verify(runny_nose),
verify(sneezing),
verify(sore_throat),
write('Advices and Sugestions:'),
nl,
write('1: Tylenol/tab'),
nl,
write('2: panadol/tab'),
nl,
write('3: Nasal spray'),
nl,
write('Please wear warm cloths Because'),
nl.

flu :-
verify(fever),
verify(headache),
verify(chills),
verify(body_ache),
write('Advices and Sugestions:'),
nl,
write('1: Tamiflu/tab'),
nl,
write('2: panadol/tab'),
nl,
write('3: Zanamivir/tab'),
nl,
write('Please take a warm bath and do salt gargling Because'),
nl.
```

```
typhoid :-
verify(headache),
verify(abdominal_pain),
verify(poor_appetite),
verify(fever),
write('Advices and Sugestions:'),
nl,
write('1: Chloramphenicol/tab'),
nl,
write('2: Amoxicillin/tab'),
nl,
write('3: Ciprofloxacin/tab'),
nl,
write('4: Azithromycin/tab'),
nl,
write('Please do complete bed rest and take soft Diet Because'),
nl.

measles :-
verify(fever),
verify(runny_nose),
verify(rash),
verify(conjunctivitis),
write('Advices and Sugestions:'),
nl,
write('1: Tylenol/tab'),
nl,
write('2: Aleve/tab'),
nl,
write('3: Advil/tab'),
nl,
write('4: Vitamin A'),
nl,
write('Please Get rest and use more liquid Because'),
nl.

malaria :-
verify(fever),
verify(sweating),
verify(headache),
verify(nausea),
verify(vomiting),
verify(diarrhea),
write('Advices and Sugestions:'),
nl,
write('1: Aralen/tab'),
nl,
write('2: Qualaquin/tab'),
nl,
write('3: Plaquenil/tab'),
nl,
write('4: Mefloquine'),
nl,
write('Please do not sleep in open air and cover your full skin Because'),
nl.
```

```
/* how to ask questions */
ask(Question) :-
write('Does the patient have following symptom:'),
write(Question),
write('? '),
read(Response),
nl,
( (Response == yes ; Response == y)
->
assert(yes(Question)) ;
assert(no(Question)), fail).

:- dynamic yes/1,no/1.
/*How to verify something */
verify(S) :-
(yes(S)
->
true ;
(no(S)
->
fail ;
ask(S))).
/* undo all yes/no assertions*/
undo :- retract(yes(_)),fail.
undo :- retract(no(_)),fail.
undo.
```

Output:

```
?-
% d:/Prolog/daignosis.pl compiled 0.02 sec, 17 clauses
?- go.
Does the patient have following symptom:headache? yes
|: .

Does the patient have following symptom:runny_nose? |: yes.

Does the patient have following symptom:sneezing? |: yes.

Does the patient have following symptom:sore_throat? |: yes.

Advices and Sugestions:
1: Tylenol/tab
2: panadol/tab
3: Nasal spray
Please wear warm cloths Because
I believe that the patient have cold
TAKE CARE
true.
```

```
?- go.
Does the patient have following symptom:headache? yes
|: .

Does the patient have following symptom:runny_nose? |: no.

Does the patient have following symptom:fever? |: yes.

Does the patient have following symptom:chills? |: yes.

Does the patient have following symptom:body_ache? |: yes.

Advices and Sugestions:
1: Tamiflu/tab
2: panadol/tab
3: Zanamivir/tab
Please take a warm bath and do salt gargling Because
I believe that the patient have flu
TAKE CARE
true.
```

```
?- hypothesis(cold).
Does the patient have following symptom:headache? yes
|: .

Does the patient have following symptom:runny_nose? |: yes.

Does the patient have following symptom:sneezing? |: yes.

Does the patient have following symptom:sore_throat? |: yes.

Advices and Sugestions:
1: Tylenol/tab
2: panadol/tab
3: Nasal spray
Please wear warm cloths Because
true.

?- hypothesis(flu).
Does the patient have following symptom:fever? yes.

Does the patient have following symptom:chills? |: no.

false.
```

References:

https://www.cpp.edu/~jrfisher/www/prolog_tutorial/pt_framer.html