LP I
Assignment : B2
Artificial Intelligence & Robotics

Title:- ChatBot Date of Completion:- 30.10.20

Problem statement:- Implement any one of the following Expert System:-
1) Medical Diagonasis of 10 diseases based on adequate symptoms
2) Identifying birds of India based on characteristics

OR

Develop elimentary chatbot suggestive investment as per the customers need.

Learning Objectives :- To implement a chatbot
:- Understand nltk.

Learning Outcomes :- Students will able to implement chatbot.

Theory:- Software/Hardware requirements:-
OS (Linux), Python

Theory:-

1. Chatbot is used when an actor has to be given some information about something which is repetative in nature.
2. NLTK is used for creating a chatbot. Following step are followed for training the chatbot model.

a) ~~Text~~ Remove Stopwords
b) Tokenization
c) Text Analysis
d) Lexican normalization
3) Sentiment analysis
4) Text classification
5) Performing sentiment analyse using text classification.

1) Text Analytics and NLP
i) NLP enables the computer to interact with humans in natural manner.
ii) Help machine to understand human language & desire meaning from it.

2) Compare text analytics, NLP and Text Mining:-
i) Text mining is a process to exploring sizeable textual data and find patterns.
ii) NLP processes with the underlying metadata.

3) Text analysis:-
i) tokenization :- Break long text into small chunks for easy processing.

ii) Stop words:-
The unnecessary text may ~~p~~ induce noise hence we remove them from text.

iii) Stemming:-
Finding the root word

iv) Sentiment Analysis:-
Analyzing if the words are positive, negative or neutral.

v) Text Classification:-
Identifying categories or class of given text such as blog, Book, tweet etc.

Test Case

Case	Expected Output	Actual Output	Remark
Information Warren Buffet	Information is showed	Information is showed	Passed
P/E ~~show~~ ratio	explained	explained	Passed
Information on investment types	Showed	not found	failed.
retail ?	not shown	not shown	failed.

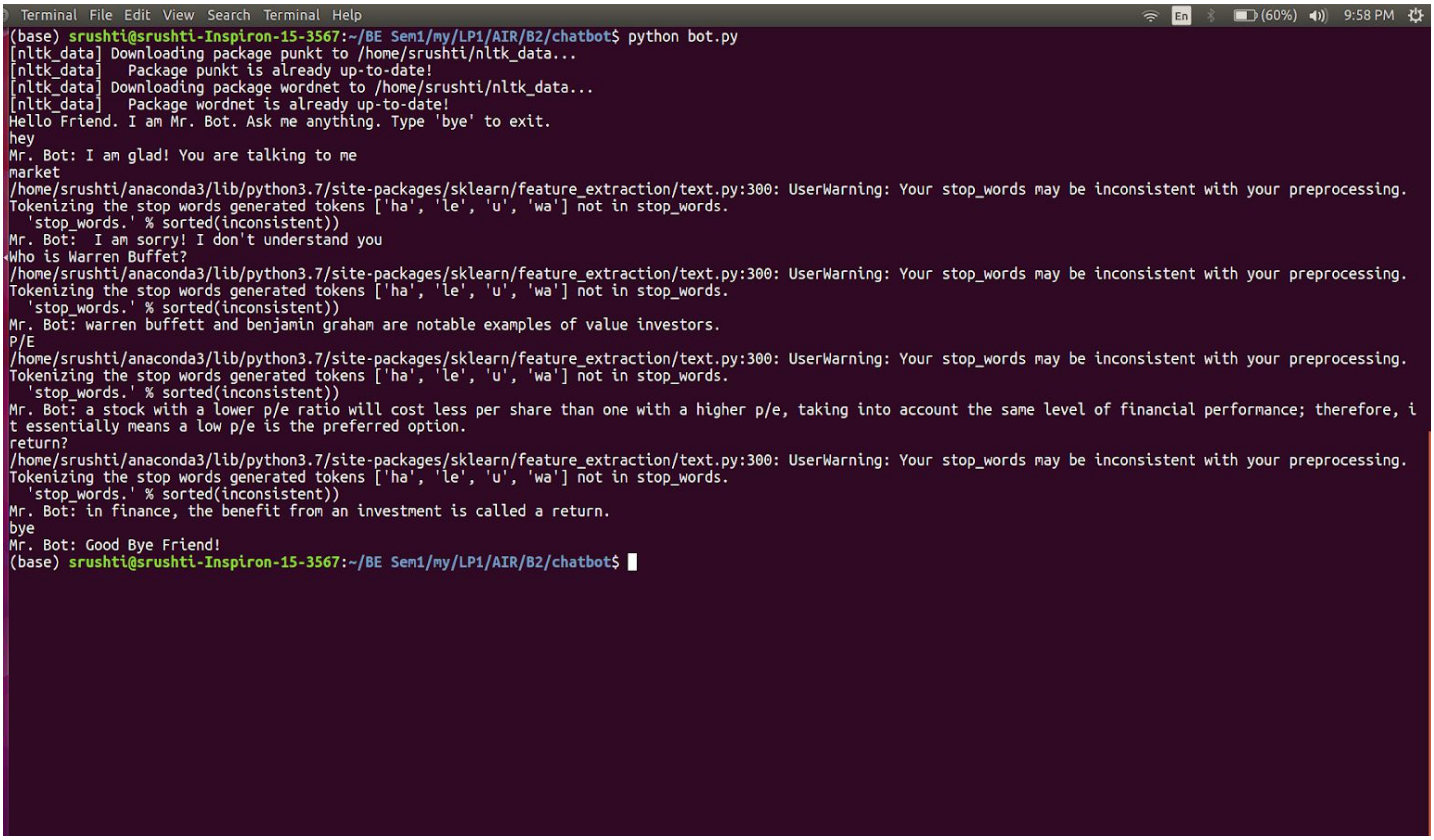
Conclusion:- Thus I implement a chatbot and understood text synthesis and analysis.

CODE

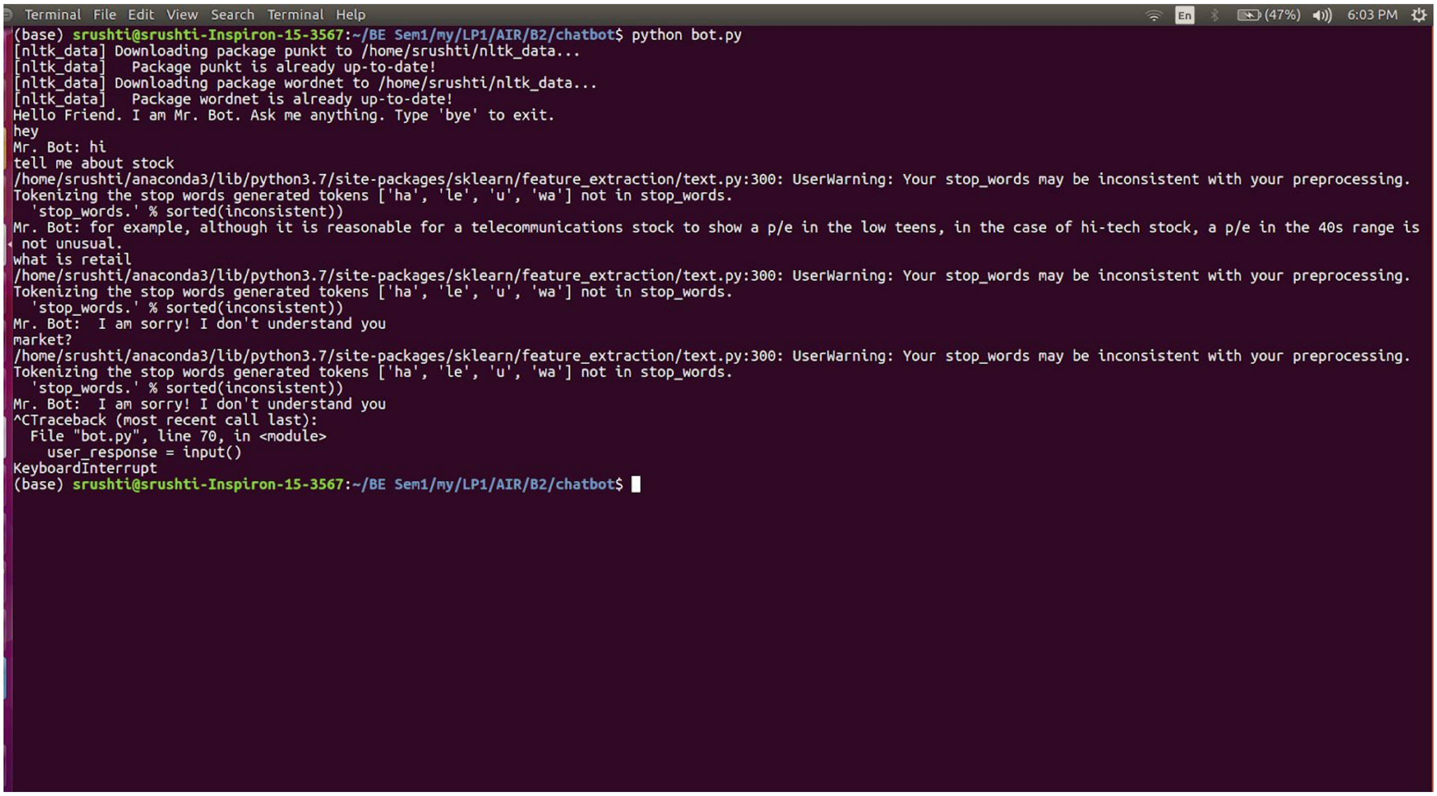
```
import nltk
import numpy as np
import random
import string
from sklearn.feature_extraction.text import TfidfVectorizer
from sklearn.metrics.pairwise import cosine_similarity


with open('data.txt','r',errors = 'ignore') as f:
    raw = f.read()
    raw = raw.lower()

nltk.download('punkt')
nltk.download('wordnet')


sent_tokens = nltk.sent_tokenize(raw)
word_tokens = nltk.word_tokenize(raw)


lemmer = nltk.stem.WordNetLemmatizer()


def LemTokens(tokens):
    return [lemmer.lemmatize(token) for token in tokens]


remove_punct_dict = dict((ord(punct), None) for punct in string.punctuation)


def LemNormalize(text):
    return LemTokens(nltk.word_tokenize(text.lower().translate(remove_punct_dict)))


GREETING_INPUTS = ("hello", "hi", "greetings", "sup", "what's up","hey",)
GREETING_RESPONSES = ["hi", "hey", "*nods*", "hi there", "hello", "I am glad! You are talking
to me"]

def greeting(sentence):
    for word in sentence.split():
        if word.lower() in GREETING_INPUTS:
            return random.choice(GREETING_RESPONSES)
```

```
def response(user_response):
    robo_response = ""

    sent_tokens.append(user_response)
    TfidfVec = TfidfVectorizer(tokenizer=LemNormalize, stop_words='english')
    tfidf = TfidfVec.fit_transform(sent_tokens)
    vals = cosine_similarity(tfidf[-1], tfidf)
    idx=vals.argsort()[0][-2]
    flat = vals.flatten()
    flat.sort()

    req_tfidf = flat[-2]

    if(req_tfidf == 0):
        robo_response = robo_response + " I am sorry! I don't understand you"
        return robo_response
    else:
        robo_response = robo_response+sent_tokens[idx]
        return robo_response


flag = True

print("Hello Friend. I am Mr. Bot. Ask me anything. Type 'bye' to exit. ")
while(flag == True):

    user_response = input()
    user_response = user_response.lower()
    if(user_response != 'bye'):
        if(user_response == 'thanks' or user_response == 'thank you' ):
            flag = False
            print("Mr. Bot: You are welcome")
        else:
            if(greeting(user_response) != None):
                print("Mr. Bot: "+greeting(user_response))
            else:
                print("Mr. Bot: ",end = "")
                print(response(user_response))
                sent_tokens.remove(user_response)
    else:
        flag = False
        print("Mr. Bot: Good Bye Friend!")
```

OUTPUT



```
(base) srushti@srushti-Inspiron-15-3567:~/BE Sem1/my/LP1/AIR/B2/chatbot$ python bot.py
[nltk_data] Downloading package punkt to /home/srushti/nltk_data...
[nltk_data]   Package punkt is already up-to-date!
[nltk_data] Downloading package wordnet to /home/srushti/nltk_data...
[nltk_data]   Package wordnet is already up-to-date!
Hello Friend. I am Mr. Bot. Ask me anything. Type 'bye' to exit.
hey
Mr. Bot: I am glad! You are talking to me
market
/home/srushti/anaconda3/lib/python3.7/site-packages/sklearn/feature_extraction/text.py:300: UserWarning: Your stop_words may be inconsistent with your preprocessing.
Tokenizing the stop words generated tokens ['ha', 'le', 'u', 'wa'] not in stop_words.
  'stop_words.' % sorted(inconsistent))
Mr. Bot:  I am sorry! I don't understand you
Who is Warren Buffet?
/home/srushti/anaconda3/lib/python3.7/site-packages/sklearn/feature_extraction/text.py:300: UserWarning: Your stop_words may be inconsistent with your preprocessing.
Tokenizing the stop words generated tokens ['ha', 'le', 'u', 'wa'] not in stop_words.
  'stop_words.' % sorted(inconsistent))
Mr. Bot: warren buffett and benjamin graham are notable examples of value investors.
P/E
/home/srushti/anaconda3/lib/python3.7/site-packages/sklearn/feature_extraction/text.py:300: UserWarning: Your stop_words may be inconsistent with your preprocessing.
Tokenizing the stop words generated tokens ['ha', 'le', 'u', 'wa'] not in stop_words.
  'stop_words.' % sorted(inconsistent))
Mr. Bot: a stock with a lower p/e ratio will cost less per share than one with a higher p/e, taking into account the same level of financial performance; therefore, i
t essentially means a low p/e is the preferred option.
return?
/home/srushti/anaconda3/lib/python3.7/site-packages/sklearn/feature_extraction/text.py:300: UserWarning: Your stop_words may be inconsistent with your preprocessing.
Tokenizing the stop words generated tokens ['ha', 'le', 'u', 'wa'] not in stop_words.
  'stop_words.' % sorted(inconsistent))
Mr. Bot: in finance, the benefit from an investment is called a return.
bye
Mr. Bot: Good Bye Friend!
(base) srushti@srushti-Inspiron-15-3567:~/BE Sem1/my/LP1/AIR/B2/chatbot$
```



```
(base) srushti@srushti-Inspiron-15-3567:~/BE Sem1/my/LP1/AIR/B2/chatbot$ python bot.py
[nltk_data] Downloading package punkt to /home/srushti/nltk_data...
[nltk_data]   Package punkt is already up-to-date!
[nltk_data] Downloading package wordnet to /home/srushti/nltk_data...
[nltk_data]   Package wordnet is already up-to-date!
Hello Friend. I am Mr. Bot. Ask me anything. Type 'bye' to exit.
hey
Mr. Bot: hi
tell me about stock
/home/srushti/anaconda3/lib/python3.7/site-packages/sklearn/feature_extraction/text.py:300: UserWarning: Your stop_words may be inconsistent with your preprocessing.
Tokenizing the stop words generated tokens ['ha', 'le', 'u', 'wa'] not in stop_words.
  'stop_words.' % sorted(inconsistent))
Mr. Bot: for example, although it is reasonable for a telecommunications stock to show a p/e in the low teens, in the case of hi-tech stock, a p/e in the 40s range is
 not unusual.
what is retail
/home/srushti/anaconda3/lib/python3.7/site-packages/sklearn/feature_extraction/text.py:300: UserWarning: Your stop_words may be inconsistent with your preprocessing.
Tokenizing the stop words generated tokens ['ha', 'le', 'u', 'wa'] not in stop_words.
  'stop_words.' % sorted(inconsistent))
Mr. Bot:  I am sorry! I don't understand you
market?
/home/srushti/anaconda3/lib/python3.7/site-packages/sklearn/feature_extraction/text.py:300: UserWarning: Your stop_words may be inconsistent with your preprocessing.
Tokenizing the stop words generated tokens ['ha', 'le', 'u', 'wa'] not in stop_words.
  'stop_words.' % sorted(inconsistent))
Mr. Bot:  I am sorry! I don't understand you
^CTraceback (most recent call last):
  File "bot.py", line 70, in <module>
    user_response = input()
KeyboardInterrupt
(base) srushti@srushti-Inspiron-15-3567:~/BE Sem1/my/LP1/AIR/B2/chatbot$
```