Chatbot Project 2

```
from google.colab import files
uploaded = files.upload()
```

users.xml

- **users.xml**(text/xml) - 17 bytes, last modified: 5/2/2024 - 100% done

```
Saving users.xml to users.xml
```

```
import pandas as pd
from sklearn.model_selection import train_test_split
from sklearn.feature_extraction.text import TfidfVectorizer
from sklearn.pipeline import Pipeline
from sklearn.naive_bayes import MultinomialNB
from sklearn.neural_network import MLPClassifier
import spacy
from bs4 import BeautifulSoup
import re
```

Train Machine Learning Models

Text Classifier

The following is basesd on the github scikit examples from class.

Read in questions.csv

```
data = pd.read_csv('questions.csv',header=0,encoding='latin-1')
```

Remove puctuation, numbers

```
data['Question'].replace('[\d][\d]+', ' num ', regex=True, inplace=True)
data['Question'].replace('[!@#*][!@#*]+', ' punct ', regex=True, inplace=True)
```

Set x and y

```
x=data.Question
y=data.Category
```

I will not be splitting the data into train and test because I have a very small dataset and would like to train the data on the entire dataset.

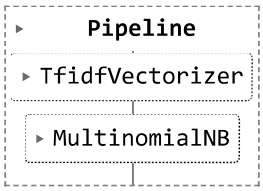
Create the vectorizer

```
vectorizer_a = TfidfVectorizer(decode_error='replace', lowercase=True, stop_words='english', max_df=0.8,
```

The ngram range here was chosen based on https://dylancastillo.co/text-classification-using-python-and-scikit-learn/#train-and-evaluate-the-model.

Create and train the naive bayes classifier

```
question_model = Pipeline([('tfidf', vectorizer_a), ('multinb',MultinomialNB())])
question_model.fit(x,y)
```



Sentiment Classifier

The following is based on the scikit-learn examples from class. The code is mostly taken from my code for assignment 3.

Read the csv file

```
sentiment_data = pd.read_csv('sentiment-emotion-labelled_Dell_tweets.csv', header = 0,
                  usecols=[3,5],encoding='latin-1')
```

Remove user tags, urls, numbers, and punctuation.

```
sentiment_data['Text'].replace('[\d][\d]+', ' num ', regex=True, inplace=True)
sentiment_data['Text'].replace('[!@#*][!@#*]+', ' punct ', regex=True, inplace=True)
sentiment_data['Text'].replace("http[A-Za-z0-9$/-_.+!*'(),;/?:@=&]",'', regex=True, inplace=True)
sentiment_data['Text'].replace("@[a-zA-Z0-9_]",'',regex=True, inplace=True)
sentiment_data['Text'].replace("  ", " ", regex=True,inplace=True)
```

I will be using the same vectorizer as before.

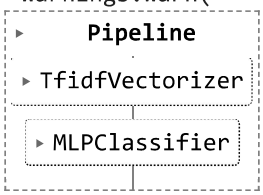
```
vectorizer = TfidfVectorizer(decode_error='replace', lowercase=True, stop_words='english', max_df=0.8, n
```

```
sent_x = sentiment_data.Text
sent_y = sentiment_data.sentiment
```

I've chosen to keep the same details from Homework 3 because this was the combination that previously gave the best results that I was able to achieve.

```
sent_model = Pipeline([('tfidf',vectorizer),
                       ('neuralnet',MLPClassifier(solver='sgd',alpha=1e-5,
                                                  hidden_layer_sizes=(15,7),
                                                  random_state=1,
                                                  max_iter=200))])
sent_model.fit(sent_x, sent_y)
```

```
/usr/local/lib/python3.10/dist-packages/sklearn/neural_network/_multilayer_perceptron.py
  warnings.warn(
```



Create Chatbot

Read in files

```
kb = pd.read_csv('plantInfo-clean (2).csv', header = 0, encoding='latin-1', index_col='Name')

with open('users.xml', 'r') as x:
  udata = x.read()

user_data = BeautifulSoup(udata, 'xml')
uroot = user_data.find('users')
```

Get name of user

The SpaCy code is based on the github SpaCy examples.

Identify the name

```
print("Hi, I'm a chatbot that knows about kitchen gardening. What's your name? I'd like to make a profile
username = ''
ner = spacy.load('en_core_web_sm')

while username == '':
  user_text = input()
  spacy_text = ner(user_text)
  if spacy_text.ents:
    name = ''
    for ent in spacy_text.ents:
      name = name + ent.text
```

```
      name = name + ent.text
    print('I understand that your name is '+name+'. Is that correct?')
    user_text = input()
    if 'yes' in user_text or 'Yes' in user_text:
      username = username + name
    else:
      print("I'm sorry, I must have misunderstood. What is your name?")
  else:
    print("I'm sorry, I didn't hear a name. Could you tell me your name?")
```

```
 Hi, I'm a chatbot that knows about kitchen gardening. What's your name? I'd like to make a profile f
 Adrianna
 I understand that your name is Adrianna. Is that correct?
 Yes
```

Add the name to the user models or find the name in the user models.

```
if user_data.find(username):
  user = user_data.find(username)
else:
  user = user_data.new_tag(username)
  uroot.append(user)
```

Get topic

```
def get_topic(user_text, spacy_text, previous_topic):
  user_text = user_text.lower()
  topic = ''
  row = ''
  row_count = 0

  while topic == '' and row_count < kb.shape[0]:
    if kb.index[row_count].lower() in user_text:
      topic = kb.index[row_count]
      row = kb.loc[topic]
    elif isinstance(kb.iloc[row_count][0], str):
      alts = kb.iloc[row_count][0].lower().split(", ")
      for alt in alts:
        if alt in user_text:
          topic = kb.index[row_count]
          row = kb.loc[topic]
    row_count = row_count + 1

  if topic == '':
    pronouns = [token for token in spacy_text if token.pos_=='PRON']

  if topic == '' and len(pronouns) > 0 and previous_topic != '':
    topic = previous_topic
    row = kb.loc[topic]
  elif topic == '':
    topic = 'None'

  return topic, row
```

Get Question

```
def get_id(initial):
  id = ''
  if initial == 'name':
    id = 'Name'
  elif initial == 'alternatename':
    id = 'alternateName'
  elif initial == 'sowinstructions':
    id = 'sowInstructions'
  elif initial == 'spaceinstructions':
    id = 'spaceInstructions'
  elif initial == 'harvestinstructions':
    id = 'harvestInstructions'
  elif initial == 'compatibleplants':
    id = 'compatiblePlants'
  elif initial == 'avoidinstructions':
    id = 'avoidInstructions'
  elif initial == 'culinaryhints':
    id = 'culinaryHints'
  elif initial == 'culinarypreservation':
    id = 'culinaryPreservation'
  else:
    id = 'url'

  return id
```

Conversation logic

Main conversation loop

```
print("Hi "+ username +", what would you like to talk about today? I can tell you about how to grow, harve
user_text = input()
previous_topic = ''

while user_text != 'Goodbye' and user_text != 'goodbye':
  if user_text == '':
    print("I'm sorry, I didn't catch that. Could you try again?")
  else:
    spacy_text = ner(user_text)
    topic,row = get_topic(user_text, spacy_text, previous_topic)
    if topic == 'None':
      print("I'm sorry, I don't know the plant that you're referring to. Could you ask about another plant
    else:
      previous_topic = topic
      user_text = re.sub('[\d][\d]+', ' num ', user_text)
      user_text = re.sub('[!@#*][!@#*]+', ' punct ', user_text)
      # get the sentiment
      sent = sent_model.predict([user_text])
      if sent[0] == 'positive':
        like = user_data.new_tag('1')
        like.string = topic
```

```
        user.append(like)
      elif sent[0] == 'negative':
        dislike = user_data.new_tag('d')
        dislike.string = topic
        user.append(dislike)
      # get the question
      df = pd.DataFrame()
      info = [[user_text,'unknown']]
      df = pd.DataFrame(info,columns=['Question','Category'])
      question = df.Question
      qpred = question_model.predict(question)
      # get the response
      id = get_id(qpred[0])
      response = row[id]
      if pd.isna(response):
        print("I'm sorry, I don't know about that for "+topic+".")
      else:
        print(response)
  user_text = input()
```

```
Hi Adrianna, what would you like to talk about today? I can tell you about how to grow, harvest, and
How do I cook peas?
Raw straight from the pod in the garden is best! Raw in salads. Steamed lightly. Small pods can be s
How do I grow them?
Peas are easy to grow. Sow in the garden. Sow the seed at a depth approximately three times the diam
Do they need to be spaced far apart?
Space plants 5 - 8 cm apart.
When should I harvest them?
Harvest them in 9-11 weeks. Pick the pods every day to increase production.
What about potatoes?
Peeled or unpeeled and scrubbed, potatoes can be boiled, baked, fried and roasted. - The only way th
Can I freeze potatoes?
I'm sorry, I don't know about that for Potato.
Where can I learn more about potatoes?
Peeled or unpeeled and scrubbed, potatoes can be boiled, baked, fried and roasted. - The only way th
Goodbye
```

End the conversation

```
print("I hope this conversation has been helpful! Happy planting!")
file = open("users.xml","w")
file.write(user_data.prettify())
file.close()
```

```
I hope this conversation has been helpful! Happy planting!
```