20BCE204

IRS Practical 4

Extract features and opinions from the product review to enhance the performance of the traditional recommender system. To extract features, consider frequent nouns and to extract opinions consider nearer words from a frequent noun. Use Part Of Speech tagging as a preprocessing technique before extracting features and opinions from product reviews.

In [15]:
```
from sklearn.feature_extraction.text import CountVectorizer
from nltk import word_tokenize
import nltk
import numpy as np
import pandas as pd
from nltk.tag import DefaultTagger
from nltk.corpus import stopwords
```

In [ ]:
```
# import spacy
# nlp= spacy.load("en_core_web_sm")
```

In [ ]:
```
# import spacy
# # nlp= spacy.load("en_core_web_sm")
```

In [8]:
```
t=['today is friday']
tagged = nltk.pos_tag(t[0].split(" "))
print(type(tagged[0]))
```

```
<class 'tuple'>
```

In [6]:
```
data= pd.read_csv('Reviews.csv')
data
```

Out[6]:

	Id	ProductId	UserId	ProfileName	HelpfulnessNumerator	H
0	1	B001E4KFG0	A3SGXH7AUHU8GW	delmartian	1	
1	2	B00813GRG4	A1D87F6ZCVE5NK	dll pa	0	
2	3	B000LQOCH0	ABXLMWJIXXAIN	Natalia Corres "Natalia Corres"	1	
3	4	B000UA0QIQ	A395BORC6FGVXV	Karl	3	
4	5	B006K2ZZ7K	A1UQRSCLF8GW1T	Michael D. Bigham "M. Wassir"	0	
...	...	...	...	...	...	
568449	568450	B001EO7N10	A28KG5XORO54AY	Lettie D. Carter	0	
568450	568451	B003S1WTCU	A3I8AFVPEE8KI5	R. Sawyer	0	
568451	568452	B004I613EE	A121AA1GQV751Z	pksd "pk_007"	2	
568452	568453	B004I613EE	A3IBEVCTXKNOH	Kathy A. Welch "katwel"	1	
568453	568454	B001LR2CU2	A3LGQPJCZVL9UC	srfell17	0	

568454 rows × 10 columns

In [11]:
```
file = pd.read_csv('Reviews.csv')
data = file['Text']
print(data)
```

```
0         I have bought several of the Vitality canned d...
1         Product arrived labeled as Jumbo Salted Peanut...
2         This is a confection that has been around a fe...
3         If you are looking for the secret ingredient i...
4         Great taffy at a great price.  There was a wid...
                                ...
568449    Great for sesame chicken..this is a good if no...
568450    I'm disappointed with the flavor. The chocolat...
568451    These stars are small, so you can give 10-15 o...
568452    These are the BEST treats for training and rew...
568453    I am very satisfied ,product is as advertised,...
Name: Text, Length: 568454, dtype: object
```

In [9]:
```
# import pandas as pd
# from nltk.tag import pos_tag

# data = {'comments':['Daniel is really cool', 'Daniel is the most amazing

# df = pd.DataFrame(data)
```

In [16]:
```
# generate one list that have all words and its pos_tag (dictionary for ea
pos = []

# if we are taking those words we save it (preprocessing)
Taken_words = []

for i in range(20):
    special = [':', ',', '.', '``', "''", '"', '%', '&', "'s", "?", "''", "
    words = nltk.word_tokenize(data[i])
    words = [word for word in words if word not in set(stopwords.words('en
    words = [word for word in words if word not in special]

    Taken_words.append(words)

    taggedtext = {}
    temp = nltk.pos_tag(words)
    for i in range(len(temp)):
        taggedtext[temp[i][0]] = temp[i][1]
    pos.append(taggedtext)

print(len(pos))
print(pos[0])
```

```
20
{'I': 'PRP', 'bought': 'VBD', 'several': 'JJ', 'Vitality': 'NNP', 'canned':
'VBD', 'dog': 'RP', 'food': 'NN', 'products': 'NNS', 'found': 'VBD', 'good'
: 'JJ', 'quality': 'NN', 'The': 'DT', 'product': 'NN', 'looks': 'VBZ', 'lik
e': 'IN', 'stew': 'NN', 'processed': 'VBN', 'meat': 'NN', 'smells': 'NNS',
'better': 'RBR', 'My': 'NNP', 'Labrador': 'NNP', 'finicky': 'JJ', 'apprecia
tes': 'NNS'}
```

In [17]:
```
# for each word that we take in pos_tag we find noun in it and create dict

frequentNoun = []
for i in range(len(pos)):

    keys = [x for x in pos[i] if (pos[i][x]=='NN' or pos[i][x]=='NNS')]
    tempdict = {}
    for i in range(len(keys)):
        tempdict[keys[i]]=''
    frequentNoun.append(tempdict)

# for those each noun we check near 10 words and if it is adjective then we
for i in range(len(frequentNoun)):
    for j in frequentNoun[i].keys():

        tkey = list(pos[i].keys())
        for x in range(len(tkey)):

            if(tkey[x]==j):

                tempvalues=[]
                for m in range(1,10):
                    try:
                        if(pos[i][tkey[x+m]] == 'NN' or pos[i][tkey[x+m]] =
                            break
                        if(pos[i][tkey[x+m]] == 'JJ'):
                            tempvalues.append(tkey[x+m])
                    except:
                        print("", end='')
                for m in range(1,10):
                    if(x-m <0): # (it will start taking words from back si
                        break
                    try:
                        if(pos[i][tkey[x-m]] == 'NN' or pos[i][tkey[x-m]] =
                            break
                        if(pos[i][tkey[x-m]] == 'JJ'):
                            tempvalues.append(tkey[x-m])
                    except:
                        print("", end='')



                frequentNoun[i][j] = tempvalues

# print(frequentNoun[0])

print("Noun - adjective")
for i in range(len(frequentNoun)):
    for key, value in frequentNoun[i].items():
        for i in range(len(value)):
            print(f"{key} {value[i]}")
```

```
Noun - adjective
food several
products good
quality good
smells finicky
appreciates finicky
error sure
error unsalted
error small
vendor represent
product represent
centuries pillowy
citrus pillowy
gelatin nuts
case tiny
case nuts
squares powdered
squares tiny
sugar mouthful
sugar powdered
heaven chewy
heaven flavorful
heaven mouthful
treat familiar
treat flavorful
treat chewy
story familiar
ingredient secret
addition good
cherry good
price wide
price great
price taffy
assortment wide
hair wild
pound enjoyable
pound many
flavors many
flavors enjoyable
complaint much
red/black licorice-flavored
red/black much
pieces particular
pieces licorice-flavored
favorites particular
brand delightful
treat delightful
saltwater great
flavors soft
flavors great
chewy soft
candies happen
candies expensive
version beach-themed
version expensive
version happen
party beach-themed
taffy soft
flavors soft
dog healthy
digestion good
digestion small
puppies small
puppies good
tequila unique
tequila cactus
combination unique
ingredients hot
```

```
sauce hot
city anywhere
br magic
internet magic
case ecstatic
it. ecstatic
sauce. personal
sauce. incredible
service incredible
service personal
weight lose
guy protein-rich
by-product protein-rich
years new
bag new
food different
food bowls
sit full
sit bowls
sit different
kitties touch
kitties similar
kitties full
reviews similar
reviews touch
flavor fresh
flavor delicious
flavor good
love delicious
love fresh
pleasure guilty
twizzlers shipment
pounds shipment
TV good
baggie fresh
time fresh
Twizzlers favorite
candy favorite
place cool
place dry
Pounds Record-Breaking
price reasonable
bound unable
get unable
```