Kahlil Cole

Final Project

For my project I found an IMDB movie review data set. In this data set there was a column for reviews and a column for sentiment of either 0 or 1. The shape of the data set is (50,000, 2). In [**161**]: imdb.shape

Out[**161**]: (50000, 2)

For the project I wanted to figure out if the data set would have a great negative or positive polarity for the reviews. I had to clean up the data set before using it because it had a lot of things in it that would mess up the analysis. I had to remove HTML tags, take out spaces, and remove different punctuation from the data.

I used a package called BeautifulSoup to help remove HTML tags from my data set. To remove the punctuation, I used a formula that I found during my research on the topic. After I cleaned the reviews part of the dataset, I started to work on finding the sentiment polarity. I used the text blob package and the find\_pol function from the Intro to Text Analysis notes. After I found the polarities of the reviews, I made a new data frame of just the cleaned reviews and the polarities.

K = imdb.review

imdb["review"] = [BeautifulSoup(K).get\_text() for K in imdb["review"]]

imdb["review"].head()

df\_imRev = imdb["review"]

def remove\_punctuation(no\_punc):

no\_punc = ''.join([i for i in no\_punc if i not in frozenset(string.punctuation)])

return no\_punc

rev\_clean = df\_imRev.apply(remove\_punctuation)

def find\_pol(review):

return TextBlob(review).sentiment.polarity

rev\_clean['Sentiment\_Polarity'] = rev\_clean.apply(find\_pol)

d\_rev\_clean = rev\_clean.drop("Sentiment\_Polarity", axis = 0)

rev\_sent = rev\_clean['Sentiment\_Polarity']

new\_Rev = pd.DataFrame({"review":d\_rev\_clean, "Sentiment\_Polarity": rev\_sent})

After I made the new data frame, I made variables called pos and neg which separated the polarities of the reviews. I also made variables called pos\_count and neg\_count to count the polarities.

neg\_count = rev\_sent[(rev\_sent >= -1) & (rev\_sent < 0)].count()

In [**163**]: neg\_count

Out[**163**]: 12146

neg = rev\_sent[(rev\_sent >= -1) & (rev\_sent < 0)]

In [**164**]: neg

Out[**164**]:

3 -0.060937

8 -0.142863

15 -0.046667

17 -0.158654

21 -0.154833

...

49982 -0.016590

49990 -0.180093

49994 -0.294737

49996 -0.276190

49998 -0.048663

Name: review, Length: 12146, dtype: float64

pos\_count = rev\_sent[(rev\_sent >= 0) & (rev\_sent <= 1)].count()

In [**165**]: pos\_count

Out[**165**]: 37854

pos = rev\_sent[(rev\_sent >= 0) & (rev\_sent <= 1)]

In [**166**]: pos

Out[**166**]:

0 0.023433

1 0.111490

2 0.346324

4 0.217952

5 0.155294

...

49992 0.095676

49993 0.161840

49995 0.394425

49997 0.056984

49999 0.120000

Name: review, Length: 37854, dtype: float64

Based on the polarity I found that 37,854 out of the 50,000 reviews had a positive polarity. The negative polarities were accounted for 12,146 out of the 50,000 reviews. Based on those number we can confirm that positive reviews have the highest amount of polarity.

I started to tokenize the reviews from the new data set created. I was not sure if you wanted us to try and tokenize the whole data set, so I only did a couple random reviews. I broke the reviews into tokens, tags, and entities.

In [**167**]: rev3 = new\_Rev["review"][3]

In [**168**]: rev3

Out[**168**]: 'Basically theres a family where a little boy Jake thinks theres a zombie in his closet his parents are fighting all the timeThis movie is slower than a soap opera and suddenly Jake decides to become Rambo and kill the zombieOK first of all when youre going to make a film you must Decide if its a thriller or a drama As a drama the movie is watchable Parents are divorcing arguing like in real life And then we have Jake with his closet which totally ruins all the film I expected to see a BOOGEYMAN similar movie and instead i watched a drama with some meaningless thriller spots3 out of 10 just for the well playing parents descent dialogs As for the shots with Jake just ignore them'

In [**169**]: tokens = nltk.word\_tokenize(rev3)

In [**170**]: print(tokens)

['Basically', 'theres', 'a', 'family', 'where', 'a', 'little', 'boy', 'Jake', 'thinks', 'theres', 'a', 'zombie', 'in', 'his', 'closet', 'his', 'parents', 'are', 'fighting', 'all', 'the', 'timeThis', 'movie', 'is', 'slower', 'than', 'a', 'soap', 'opera', 'and', 'suddenly', 'Jake', 'decides', 'to', 'become', 'Rambo', 'and', 'kill', 'the', 'zombieOK', 'first', 'of', 'all', 'when', 'youre', 'going', 'to', 'make', 'a', 'film', 'you', 'must', 'Decide', 'if', 'its', 'a', 'thriller', 'or', 'a', 'drama', 'As', 'a', 'drama', 'the', 'movie', 'is', 'watchable', 'Parents', 'are', 'divorcing', 'arguing', 'like', 'in', 'real', 'life', 'And', 'then', 'we', 'have', 'Jake', 'with', 'his', 'closet', 'which', 'totally', 'ruins', 'all', 'the', 'film', 'I', 'expected', 'to', 'see', 'a', 'BOOGEYMAN', 'similar', 'movie', 'and', 'instead', 'i', 'watched', 'a', 'drama', 'with', 'some', 'meaningless', 'thriller', 'spots3', 'out', 'of', '10', 'just', 'for', 'the', 'well', 'playing', 'parents', 'descent', 'dialogs', 'As', 'for', 'the', 'shots', 'with', 'Jake', 'just', 'ignore', 'them']

In [**171**]: tag = nltk.pos\_tag(tokens)

In [**172**]: print(tag)

[('Basically', 'NNP'), ('theres', 'VBZ'), ('a', 'DT'), ('family', 'NN'), ('where', 'WRB'), ('a', 'DT'), ('little', 'JJ'), ('boy', 'JJ'), ('Jake', 'NNP'), ('thinks', 'VBZ'), ('theres', 'VBZ'), ('a', 'DT'), ('zombie', 'NN'), ('in', 'IN'), ('his', 'PRP$'), ('closet', 'NN'), ('his', 'PRP$'), ('parents', 'NNS'), ('are', 'VBP'), ('fighting', 'VBG'), ('all', 'PDT'), ('the', 'DT'), ('timeThis', 'JJ'), ('movie', 'NN'), ('is', 'VBZ'), ('slower', 'JJR'), ('than', 'IN'), ('a', 'DT'), ('soap', 'NN'), ('opera', 'NN'), ('and', 'CC'), ('suddenly', 'RB'), ('Jake', 'NNP'), ('decides', 'VBZ'), ('to', 'TO'), ('become', 'VB'), ('Rambo', 'NNP'), ('and', 'CC'), ('kill', 'VB'), ('the', 'DT'), ('zombieOK', 'NN'), ('first', 'RB'), ('of', 'IN'), ('all', 'DT'), ('when', 'WRB'), ('youre', 'NN'), ('going', 'VBG'), ('to', 'TO'), ('make', 'VB'), ('a', 'DT'), ('film', 'NN'), ('you', 'PRP'), ('must', 'MD'), ('Decide', 'VB'), ('if', 'IN'), ('its', 'PRP$'), ('a', 'DT'), ('thriller', 'NN'), ('or', 'CC'), ('a', 'DT'), ('drama', 'NN'), ('As', 'IN'), ('a', 'DT'), ('drama', 'NN'), ('the', 'DT'), ('movie', 'NN'), ('is', 'VBZ'), ('watchable', 'JJ'), ('Parents', 'NNS'), ('are', 'VBP'), ('divorcing', 'VBG'), ('arguing', 'VBG'), ('like', 'IN'), ('in', 'IN'), ('real', 'JJ'), ('life', 'NN'), ('And', 'CC'), ('then', 'RB'), ('we', 'PRP'), ('have', 'VBP'), ('Jake', 'VBN'), ('with', 'IN'), ('his', 'PRP$'), ('closet', 'NN'), ('which', 'WDT'), ('totally', 'RB'), ('ruins', 'VBZ'), ('all', 'PDT'), ('the', 'DT'), ('film', 'NN'), ('I', 'PRP'), ('expected', 'VBD'), ('to', 'TO'), ('see', 'VB'), ('a', 'DT'), ('BOOGEYMAN', 'NNP'), ('similar', 'JJ'), ('movie', 'NN'), ('and', 'CC'), ('instead', 'RB'), ('i', 'VB'), ('watched', 'VBD'), ('a', 'DT'), ('drama', 'NN'), ('with', 'IN'), ('some', 'DT'), ('meaningless', 'NN'), ('thriller', 'NN'), ('spots3', 'VBD'), ('out', 'IN'), ('of', 'IN'), ('10', 'CD'), ('just', 'RB'), ('for', 'IN'), ('the', 'DT'), ('well', 'NN'), ('playing', 'VBG'), ('parents', 'NNS'), ('descent', 'JJ'), ('dialogs', 'NNS'), ('As', 'IN'), ('for', 'IN'), ('the', 'DT'), ('shots', 'NNS'), ('with', 'IN'), ('Jake', 'NNP'), ('just', 'RB'), ('ignore', 'VB'), ('them', 'PRP')]

In [**173**]: entity1 = nltk.chunk.ne\_chunk(tag)

In [**174**]: print(entity1)

(S

(GPE Basically/NNP)

theres/VBZ

a/DT

family/NN

where/WRB

a/DT

little/JJ

boy/JJ

Jake/NNP

thinks/VBZ

theres/VBZ

a/DT

zombie/NN

in/IN

his/PRP$

closet/NN

his/PRP$

parents/NNS

are/VBP

fighting/VBG

all/PDT

the/DT

(ORGANIZATION timeThis/JJ)

movie/NN

is/VBZ

slower/JJR

than/IN

a/DT

soap/NN

opera/NN

and/CC

suddenly/RB

(PERSON Jake/NNP)

decides/VBZ

to/TO

become/VB

(PERSON Rambo/NNP)

and/CC

kill/VB

the/DT

(ORGANIZATION zombieOK/NN)

first/RB

of/IN

all/DT

when/WRB

youre/NN

going/VBG

to/TO

make/VB

a/DT

film/NN

you/PRP

must/MD

Decide/VB

if/IN

its/PRP$

a/DT

thriller/NN

or/CC

a/DT

drama/NN

As/IN

a/DT

drama/NN

the/DT

movie/NN

is/VBZ

watchable/JJ

Parents/NNS

are/VBP

divorcing/VBG

arguing/VBG

like/IN

in/IN

real/JJ

life/NN

And/CC

then/RB

we/PRP

have/VBP

Jake/VBN

with/IN

his/PRP$

closet/NN

which/WDT

totally/RB

ruins/VBZ

all/PDT

the/DT

film/NN

I/PRP

expected/VBD

to/TO

see/VB

a/DT

(ORGANIZATION BOOGEYMAN/NNP)

similar/JJ

movie/NN

and/CC

instead/RB

i/VB

watched/VBD

a/DT

drama/NN

with/IN

some/DT

meaningless/NN

thriller/NN

spots3/VBD

out/IN

of/IN

10/CD

just/RB

for/IN

the/DT

well/NN

playing/VBG

parents/NNS

descent/JJ

dialogs/NNS

As/IN

for/IN

the/DT

shots/NNS

with/IN

(PERSON Jake/NNP)

just/RB

ignore/VB

them/PRP)

To conclude everything based on my code the polarity of the reviews was mostly positive. Based on the original sentiment column from the original data set the positive and negative sentiments are split 25,000 apiece.