We have provided some synthetic (fake, semi-randomly generated) twitter data in a csv file named project\_twitter\_data.csv which has the text of a tweet, the number of retweets of that tweet, and the number of replies to that tweet. We have also words that express positive sentiment and negative sentiment, in the files positive\_words.txt and negative\_words.txt.

Your task is to build a sentiment classifier, which will detect how positive or negative each tweet is. You will create a csv file, which contains columns for the Number of Retweets, Number of Replies, Positive Score (which is how many happy words are in the tweet), Negative Score (which is how many angry words are in the tweet), and the Net Score for each tweet. At the end, you upload the csv file to Excel or Google Sheets, and produce a graph of the Net Score vs Number of Retweets.

To start, define a function called strip\_punctuation which takes one parameter, a string which represents a word, and removes characters considered punctuation from everywhere in the word. (Hint: remember the .replace() method for strings.)

punctuation\_chars = ["'", '"', ",", ".", "!", ":", ";", '#', '@']

Next, copy in your strip\_punctuation function and define a function called get\_pos which takes one parameter, a string which represents one or more sentences, and calculates how many words in the string are considered positive words. Use the list, positive\_words to determine what words will count as positive. The function should return a positive integer - how many occurrences there are of positive words in the text. Note that all of the words in positive\_words are lower cased, so you’ll need to convert all the words in the input string to lower case as well.

punctuation\_chars = ["'", '"', ",", ".", "!", ":", ";", '#', '@']

# list of positive words to use

positive\_words = []

with open("positive\_words.txt") as pos\_f:

for lin in pos\_f:

if lin[0] != ';' and lin[0] != '\n':

positive\_words.append(lin.strip())

Next, copy in your strip\_punctuation function and define a function called get\_neg which takes one parameter, a string which represents one or more sentences, and calculates how many words in the string are considered negative words. Use the list, negative\_words to determine what words will count as negative. The function should return a positive integer - how many occurrences there are of negative words in the text. Note that all of the words in negative\_words are lower cased, so you’ll need to convert all the words in the input string to lower case as well.

punctuation\_chars = ["'", '"', ",", ".", "!", ":", ";", '#', '@']

negative\_words = []

with open("negative\_words.txt") as pos\_f:

for lin in pos\_f:

if lin[0] != ';' and lin[0] != '\n':

negative\_words.append(lin.strip())

Finally, copy in your previous functions and write code that opens the file project\_twitter\_data.csv which has the fake generated twitter data (the text of a tweet, the number of retweets of that tweet, and the number of replies to that tweet). Your task is to build a sentiment classifier, which will detect how positive or negative each tweet is. Copy the code from the code windows above, and put that in the top of this code window. Now, you will write code to create a csv file called resulting\_data.csv, which contains the Number of Retweets, Number of Replies, Positive Score (which is how many happy words are in the tweet), Negative Score (which is how many angry words are in the tweet), and the Net Score (how positive or negative the text is overall) for each tweet. The file should have those headers in that order. Remember that there is another component to this project. You will upload the csv file to Excel or Google Sheets and produce a graph of the Net Score vs Number of Retweets. Check Coursera for that portion of the assignment, if you’re accessing this textbook from Coursera.

punctuation\_chars = ["'", '"', ",", ".", "!", ":", ";", '#', '@']

# lists of words to use

positive\_words = []

with open("positive\_words.txt") as pos\_f:

for lin in pos\_f:

if lin[0] != ';' and lin[0] != '\n':

positive\_words.append(lin.strip())

negative\_words = []

with open("negative\_words.txt") as pos\_f:

for lin in pos\_f:

if lin[0] != ';' and lin[0] != '\n':

negative\_words.append(lin.strip())

**solution:**

projectTwitterDataFile = open("project\_twitter\_data.csv","r")

resultingDataFile = open("resulting\_data.csv","w")

punctuation\_chars = ["'", '"', ",", ".", "!", ":", ";", '#', '@']

# lists of words to use

positive\_words = []

with open("positive\_words.txt") as pos\_f:

for lin in pos\_f:

if lin[0] != ';' and lin[0] != '\n':

positive\_words.append(lin.strip())

def get\_pos(strSentences):

strSentences = strip\_punctuation(strSentences)

listStrSentences= strSentences.split()

count=0

for word in listStrSentences:

for positiveWord in positive\_words:

if word == positiveWord:

count+=1

return count

negative\_words = []

with open("negative\_words.txt") as pos\_f:

for lin in pos\_f:

if lin[0] != ';' and lin[0] != '\n':

negative\_words.append(lin.strip())

def get\_neg(strSentences):

strSentences = strip\_punctuation(strSentences)

listStrSentences = strSentences.split()

count=0

for word in listStrSentences:

for negativeWord in negative\_words:

if word == negativeWord:

count+=1

return count

def strip\_punctuation(strWord):

for charPunct in punctuation\_chars:

strWord = strWord.replace(charPunct, "")

return strWord

def writeInDataFile(resultingDataFile):

resultingDataFile.write("Number of Retweets, Number of Replies, Positive Score, Negative Score, Net Score")

resultingDataFile.write("\n")

linesPTDF = projectTwitterDataFile.readlines()

headerDontUsed= linesPTDF.pop(0)

for linesTD in linesPTDF:

listTD = linesTD.strip().split(',')

resultingDataFile.write("{}, {}, {}, {}, {}".format(listTD[1], listTD[2], get\_pos(listTD[0]), get\_neg(listTD[0]), (get\_pos(listTD[0])-get\_neg(listTD[0]))))

resultingDataFile.write("\n")

writeInDataFile(resultingDataFile)

projectTwitterDataFile.close()

resultingDataFile.close()