print(os.getcwd())

**Text preprocessing:**

**import** pandas **as** pd

data**=**pd.read\_csv('Training.tsv',sep**=**'\t')

english\_stopwords **=** set([

"i", "me", "my", "myself", "we", "our", "ours", "ourselves", "you", "your",

"yours", "yourself", "yourselves", "he", "him", "his", "himself", "she",

"her", "hers", "herself", "it", "its", "itself", "they", "them", "their",

"theirs", "themselves", "what", "which", "who", "whom", "this", "that",

"these", "those", "am", "is", "are", "was", "were", "be", "been", "being",

"have", "has", "had", "having", "do", "does", "did", "doing", "a", "an",

"the", "and", "but", "if", "or", "because", "as", "until", "while", "of",

"at", "by", "for", "with", "about", "against", "between", "into", "through",

"during", "before", "after", "above", "below", "to", "from", "up", "down",

"in", "out", "on", "off", "over", "under", "again", "further", "then", "once"

])

**def** preprocess(text):

*#lower case*

text**=**to\_lower\_case(text)

*#remove contraction*

text **=** text.replace("’", "'") *#replace curly aposrophe with straight aposrophe*

text**=**remove\_contraction(text)

*# Replace digits with words*

text **=** replace\_digits\_with\_words(text)

*#remove url's*

text**=**remove\_urls(text)

*# Remove special characters (punctuation)*

text **=** remove\_special\_characters(text)

*# Tokenize the text manually (splitting by whitespace)*

tokens **=** text.split()

tokens **=** [token **for** token **in** tokens **if** token **not** **in** english\_stopwords]

*# Join tokens back into a string*

processed\_text **=** " ".join(tokens)

**return** processed\_text

**def** remove\_contraction(text):

*# Expand contractions manually (you can add more as needed)*

contractions\_map **=** {

"isn't": "is not",

"aren't": "are not",

"wasn't": "was not",

"weren't": "were not",

"haven't": "have not",

"hasn't": "has not",

"hadn't": "had not",

"won't": "will not",

"wouldn't": "would not",

"don't": "do not",

"doesn't": "does not",

"didn't": "did not",

"can't": "cannot",

"couldn't": "could not",

"shouldn't": "should not",

"mightn't": "might not",

"mustn't": "must not",

"'s": " is",

"'re": " are",

"'m": " am",

"'ll": " will"

}

**for** contraction, expansion **in** contractions\_map.items():

text **=** text.replace(contraction,expansion)

**return** text

**def** replace\_digits\_with\_words(text):

*# Map digits to their word representations (0-9 for simplicity)*

digit\_map **=** {

'0': 'zero', '1': 'one', '2': 'two', '3': 'three', '4': 'four',

'5': 'five', '6': 'six', '7': 'seven', '8': 'eight', '9': 'nine'

}

*# Replace digits with their word representations*

**for** digit, word **in** digit\_map.items():

text **=** text.replace(digit, word)

**return** text

​

**def** remove\_special\_characters(text):

*# Define a set of special characters to remove*

special\_characters **=** set("!\"#$%&'()\*+,-./:;<=>?@[\\]^\_`{|}~")

*# Remove special characters from the text*

cleaned\_text **=** "".join(char **for** char **in** text **if** char **not** **in** special\_characters)

**return** cleaned\_text

**def** to\_lower\_case(text):

lower\_text **=** ""

**for** char **in** text:

*# Check if character is uppercase*

**if** 'A' **<=** char **<=** 'Z':

*# Convert uppercase character to lowercase*

lower\_text **+=** chr(ord(char) **+** 32)

**else**:

lower\_text **+=** char

**return** lower\_text

**def** remove\_urls(text):

*# Split text into words*

words **=** text.split()

*# Filter out words that do not start with 'http' or 'https'*

filtered\_words **=** [word **for** word **in** words **if** **not** (word.startswith('http://') **or** word.startswith('https://'))]

*# Join the filtered words back into a string*

**return** ' '.join(filtered\_words)

​

​

*# Load your DataFrame from 'training.tsv' assuming it contains text data*

df **=** pd.read\_csv('training.tsv', sep**=**'\t')

​

*# Apply the preprocess function to each element in the DataFrame*

df['cleaned\_text'] **=** df['text'].apply(preprocess)

**import** emoji

*#import demoji*

​

*#demoji.download\_codes()*

**def** emo(text):

temp**=**emoji.demojize(text,delimiters**=**(" "," "))

temp**=**temp.replace("\_"," ")

**return** temp

data['emo']**=**data["text"].apply(**lambda** x:emo(x))

data["clean\_text"]**=**data['emo'].apply(**lambda** X: preprocess(X))

data.head(20)

​

​

Out[4]:

|  | **tweet\_id** | **text** | **label** | **emo** | **clean\_text** |
| --- | --- | --- | --- | --- | --- |
| **0** | 1382343793341575169 | @IrvineWelsh I don’t know about you Irvine but... | 0 | @IrvineWelsh I don’t know about you Irvine but... | irvinewelsh not know irvine keep told covid no... |
| **1** | 1377631738692796417 | I bet money if i went n took a covid test righ... | 0 | I bet money if i went n took a covid test righ... | bet money went n took covid test right now imm... |
| **2** | 1386448010029240326 | @JamesMelville My wife received a POSITIVE Cov... | 0 | @JamesMelville My wife received a POSITIVE Cov... | jamesmelville wife received positive covid tes... |
| **3** | 1361342676340211717 | Out of the 180,000+ people who have had the tw... | 0 | Out of the 180,000+ people who have had the tw... | oneeightzerozerozerozero people two vaccine sh... |
| **4** | 1386757983254765569 | My whole family is sick af and here I am now i... | 0 | My whole family is sick af and here I am now i... | whole family sick af here now hospital heart p... |
| **5** | 1382001700853125122 | @renfrew1962 @PeakePolly @J\_Deliciouso I'm not... | 0 | @renfrew1962 @PeakePolly @J Deliciouso I'm no... | renfrewoneninesixtwo peakepolly j deliciouso n... |
| **6** | 1383272654212272136 | Test came back positive, no surprise. I have c... | 1 | Test came back positive, no surprise. I have c... | test came back positive no surprise covid full... |
| **7** | 1374479299047084035 | My Pawpaw has been in the hospital a few days.... | 0 | My Pawpaw has been in the hospital a few days.... | pawpaw hospital few days got special approval ... |
| **8** | 1354020426620547072 | @MattHancock 4 people I know had covid and rec... | 0 | @MattHancock 4 people I know had covid and rec... | matthancock four people know covid recovered e... |
| **9** | 1362671045136809985 | I’m going to sound like I have lost my marbles... | 1 | I’m going to sound like I have lost my marbles... | going sound like lost marbles not felt well si... |
| **10** | 1373077971658022918 | I just tested positive for Covid-19 from two s... | 1 | I just tested positive for Covid-19 from two s... | just tested positive covidonenine two separate... |
| **11** | 1373076070312730633 | Someone I love very much was diagnosed with co... | 0 | Someone I love very much was diagnosed with co... | someone love very much diagnosed covid onenine... |
| **12** | 1331292744489250816 | Dear @PeterHotez ,Serious question about Vacci... | 0 | Dear @PeterHotez ,Serious question about Vacci... | dear peterhotez serious question vaccinein one... |
| **13** | 1379116535726415874 | As I was in the ER last night I overheard a co... | 1 | As I was in the ER last night I overheard a co... | er last night overheard conversation someone w... |
| **14** | 1357823724192296960 | @jrlsilverman @dilleradollar @OregonGovBrown 💡... | 0 | @jrlsilverman @dilleradollar @OregonGovBrown ... | jrlsilverman dilleradollar oregongovbrown ligh... |
| **15** | 1376702269991890945 | Today: 1 person (after overnight code) in ICU ... | 0 | Today: 1 person (after overnight code) in ICU ... | today one person overnight code icu wcovid fiv... |
| **16** | 1377607714524831745 | @Asilverlining20 @TVpsychologist @NHSuk It’s b... | 0 | @Asilverlining20 @TVpsychologist @NHSuk It’s b... | asilverliningtwozero tvpsychologist nhsuk year... |
| **17** | 1388377556097904640 | These side effects are CRAZY!!! Got the second... | 0 | These side effects are CRAZY!!! Got the second... | side effects crazy got second pfizer vaccine t... |
| **18** | 1377509820425723912 | @BethMooreLPM Continuing to #Pray for Healing ... | 0 | @BethMooreLPM Continuing to #Pray for Healing ... | bethmoorelpm continuing pray healing miracles ... |
| **19** | 1383382420234280964 | @mstranack @GillianMcKeith Just spoke to Hayde... | 0 | @mstranack @GillianMcKeith Just spoke to Hayde... | mstranack gillianmckeith just spoke hayden mor... |

**Reading multiple files from multiple folders:**

import os

def read\_text\_files\_from\_folders(root\_folder):

file\_contents = []

for folder\_name, subfolders, filenames in os.walk(root\_folder):

for filename in filenames:

if filename.endswith('.txt'):

file\_path = os.path.join(folder\_name, filename)

try:

with open(file\_path, 'r') as file:

content = file.read()

file\_contents.append((folder\_name, filename, content))

except Exception as e:

print(f"Error reading file {file\_path}: {e}")

return file\_contents

root\_folder = "C:\\Users\\PRATHIKSHA\\1234" # Ensure the backslashes are properly escaped

texts = read\_text\_files\_from\_folders(root\_folder)

# Print the contents

for folder\_name, filename, content in texts:

print(f"Folder: {folder\_name}")

print(f"Filename: {filename}")

print(content)

print('---' \* 10)

Folder: C:\Users\PRATHIKSHA\1234

Filename: abc.txt

aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa

bbbbbbbbbbbbbbbbbbbbbbbbbbbbbbbbbbbbbbbb

------------------------------

Folder: C:\Users\PRATHIKSHA\1234\123

Filename: untitled.txt

------------------------------

**Reading multiple files from a folder:**

import os

folder\_path = input()

for filename in os.listdir(folder\_path):

if os.path.isfile(os.path.join(folder\_path , filename)):

with open(os.path.join(folder\_path , filename), 'r') as file:

content=file.read()

print(f'Content of [filename]:{content}')

C:\\\\Users\\\\PRATHIKSHA\\\\000

Content of [filename]:NAme Age

AAA 12

ddd 13

**Mean Median mode ( with and without built in function**)

# using built in function

import numpy

from scipy import stats

numList=[]

n=int(input("Enter the total number:"))

for num in range(n):

a=int(input("Enter the numbers:"))

numList.append(a)

x=numpy.mean(numList)

y=numpy.median(numList)

z=stats.mode(numList)

print("Mean:",x)

print("Median:",y)

print(z)

# without using built in function

def calculate\_mean(data):

total = 0

for num in data:

total += num

return total / len(data)

def calculate\_median(data):

sorted\_data = sorted(data)

n = len(sorted\_data)

if n % 2 == 0:

return (sorted\_data[n // 2 - 1] + sorted\_data[n // 2]) / 2

else:

return sorted\_data[n // 2]

def calculate\_mode(data):

frequency = {}

for num in data:

if num in frequency:

frequency[num] += 1

else:

frequency[num] = 1

max\_freq = max(frequency.values())

modes = [key for key, value in frequency.items() if value == max\_freq]

return modes[0]

def get\_user\_input():

data = input("Enter a list of numbers separated by spaces: ").split()

data = [int(num) for num in data]

return data

# Main program

user\_data = get\_user\_input()

print("Mean:", calculate\_mean(user\_data))

print("Median:", calculate\_median(user\_data))

print("Mode:", calculate\_mode(user\_data))

Enter the total number:7

Enter the numbers:2

Enter the numbers:2

Enter the numbers:4

Enter the numbers:2

Enter the numbers:3

Enter the numbers:5

Enter the numbers:1

Mean: 2.7142857142857144

Median: 2.0

ModeResult(mode=array([2]), count=array([3]))

Enter a list of numbers separated by spaces: 2 2 4 2 3 5 1

Mean: 2.7142857142857144

Median: 2

Mode: 2

**Range, variance ,standard deviation and iqr(without using built in function)**

def calculate\_range(data):

return max(data) - min(data)

def calculate\_variance(data):

mean = sum(data) / len(data)

return sum((x - mean) \*\* 2 for x in data) / len(data)

def calculate\_standard\_deviation(data):

return calculate\_variance(data) \*\* 0.5

def calculate\_interquartile\_range(data):

sorted\_data = sorted(data)

n = len(sorted\_data)

Q1 = sorted\_data[n // 4]

Q3 = sorted\_data[(3 \* n) // 4]

return Q3 - Q1

def get\_user\_input():

data = input("Enter a list of numbers separated by spaces: ").split()

return [float(num) for num in data]

data = get\_user\_input()

print("Without using built-in functions:")

print("Range:", calculate\_range(data))

print("Variance:", calculate\_variance(data))

print("Standard Deviation:", calculate\_standard\_deviation(data))

print("Interquartile Range:", calculate\_interquartile\_range(data))

Enter a list of numbers separated by spaces: 3 3 5 6 33 2 1 10

Without using built-in functions:

Range: 32.0

Variance: 97.109375

Standard Deviation: 9.854408911751126

Interquartile Range: 7.0

Range, variance ,standard deviation and iqr:( using built in function):

import math

import numpy as np

def calculate\_range\_builtin(data):

return sum(data) / len(data)

def calculate\_variance\_builtin(data):

mean = calculate\_range\_builtin(data)

variance = sum((x - mean) \*\* 2 for x in data) / len(data)

return variance

def calculate\_standard\_deviation\_builtin(data):

variance = calculate\_variance\_builtin(data)

return math.sqrt(variance)

def calculate\_quartiles\_builtin(data):

Q1 = numpy.percentile(data, 25, interpolation='lower')

Q3 = numpy.percentile(data, 75, interpolation='higher')

return Q1, Q3

def calculate\_interquartile\_range\_builtin(data):

Q1, Q3 = calculate\_quartiles\_builtin(data)

return Q3 - Q1

# Main program

def get\_user\_input():

data = input("Enter a list of numbers separated by spaces: ").split()

return [float(num) for num in data]

data = get\_user\_input()

print("\nUsing built-in functions:")

print("Range:", calculate\_range\_builtin(data))

print("Variance:", calculate\_variance\_builtin(data))

print("Standard Deviation:", calculate\_standard\_deviation\_builtin(data))

print("Interquartile Range:", calculate\_interquartile\_range\_builtin(data))

Enter a list of numbers separated by spaces: 3 3 5 6 33 2 1 10

Using built-in functions:

Range: 7.875

Variance: 97.109375

Standard Deviation: 9.854408911751126

Interquartile Range: 8.0