import os

from sklearn.feature\_extraction.text import TfidfVectorizer

from sklearn.metrics.pairwise import cosine\_similarity

​

student\_files = [doc for doc in os.listdir() if doc.endswith('.txt')]

student\_notes =[open(File).read() for File in student\_files]

​

vectorize = lambda Text: TfidfVectorizer().fit\_transform(Text).toarray()

similarity = lambda doc1, doc2: cosine\_similarity([doc1, doc2])

​

vectors = vectorize(student\_notes)

s\_vectors = list(zip(student\_files, vectors))

​

def check\_plagiarism():

plagiarism\_results = set()

global s\_vectors

for student\_a, text\_vector\_a in s\_vectors:

new\_vectors =s\_vectors.copy()

current\_index = new\_vectors.index((student\_a, text\_vector\_a))

del new\_vectors[current\_index]

for student\_b , text\_vector\_b in new\_vectors:

sim\_score = similarity(text\_vector\_a, text\_vector\_b)[0][1]

student\_pair = sorted((student\_a, student\_b))

score = (student\_pair[0], student\_pair[1],sim\_score)

plagiarism\_results.add(score)

return plagiarism\_results

​for data in check\_plagiarism():

print(data