import pandas as pd #импорт модуля пандас под именем пд, необходимого для работы с таблицей из экселя  
import nltk #импорт пакета nltk для NLP  
from nltk.corpus import stopwords #импорт объекта для удаления лишних (стоп) слов из модуля corpus из пакета nltk для NLP  
from string import punctuation #импорт объекта для удаления знаков препинания между словами  
  
survey = pd.read\_excel("D:\Docs\Excel2.xlsx",  
 usecols='A,J') #выбираем только колонки A, J для идентификации и анализа  
  
comments = survey["Комментарий клиента"].astype(str).tolist() #создаем массив только с комментариями  
clients = survey["Id шаблона"].astype(str).tolist() #создаем массив только с id клиента  
  
def matchrus(text, alphabet=set('абвгдеёжзийклмнопрстуфхцчшщъыьэюя')):  
 return not alphabet.isdisjoint(text.lower()) #функция для проверки, если комментарий написан на русском  
def matcheng(text, alphabet=set('abcdefghijklmnopqrstuvwxyz')):  
 return not alphabet.isdisjoint(text.lower()) #функция для проверки, если комментарий написан на английском  
  
russiancomments = [] #создаем три пустых массива, чтобы в них записать, соответственно, комментарии на русском  
englishcomments = [] #на английском  
nulcomments = [] #и "нулевые"  
for i in range(0,len(clients)): #цикл, который исполняется для каждого id клиента  
 a=[] #пустой массив  
 a.append(i) #в него записываем порядковый номер строки, начиная с 0  
 a.append(clients[i]) #следом в него записываем id клиента  
 a.append(comments[i]) #и его комментарий  
 if matchrus(comments[i]):  
 russiancomments.append(a) #если комментарий на русском, записываем в массив для руссских комментариев  
 elif matcheng(comments[i]):  
 englishcomments.append(a) #аналогично для английского  
 else:  
 nulcomments.append(a) #оставшиеся попадают в "нулевой" массив  
#получился массив в массиве на самом деле, но для простоты понимания ниже буду называть массивы в массиве строчками  
  
print(russiancomments) #печатаем все три массива по очереди  
print(englishcomments) #так их можно наглядно увидеть  
print(nulcomments) #и при необходимости скорректировать алгоритм  
print(len(russiancomments)) #печатаем количество строк в каждом массиве  
print(len(englishcomments)) #так их можно сложить и ерепроверить, что сумма сошлась верно  
print(len(nulcomments)) #можно это и булевой функцией записать коротко, но пока нужна наглядность для корректировки  
  
russian\_stopwords = stopwords.words("russian")  
print(russian\_stopwords) #печатаем русские лишние (стоп) слова из пакета nltk, чтобы понимать, что будем дальше выкидывать  
english\_stopwords = stopwords.words("english")  
print(english\_stopwords) #для английского отдельно  
  
for i in range(0, len(russiancomments)): #делим тексты на токены, то есть на отдельные слова, выкидывая знаки препинания и стоп-слова  
 text = russiancomments[i][2]  
 tokens = nltk.word\_tokenize(text.lower())  
 tokens = [token for token in tokens if token not in russian\_stopwords and token not in punctuation]  
 print(tokens)  
  
for i in range(0, len(englishcomments)): #аналогично для английского  
 text = englishcomments[i][2]  
 tokens = nltk.word\_tokenize(text.lower())  
 tokens = [token for token in tokens if token not in english\_stopwords and token not in punctuation]  
 print(tokens)  
  
for i in range(0, len(nulcomments)): #аналогично для нулевого  
 text = nulcomments[i][2]  
 tokens = nltk.word\_tokenize(text.lower())  
 print(tokens)