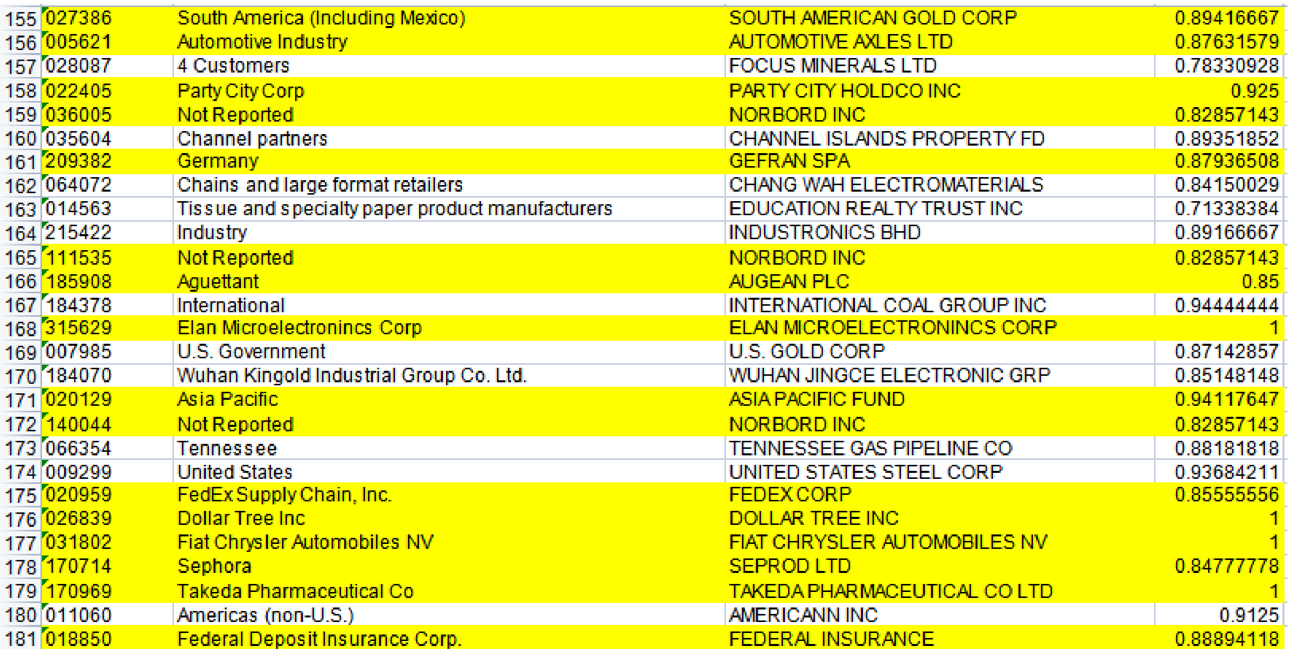
1. 书写并执行`公司名称匹配.py`文件（代码段放最后了），找出缩写与名称匹配度在前三的公司名称写入Match Company name.xlsx表中
2. 在匹配的三个公司中确定匹配无误的**公司缩写和名称**提取出来，单独放一个表中，**使公司与缩写唯一配对**：



155	027386	South America (Including Mexico)	SOUTH AMERICAN GOLD CORP	0.89416667
156	005621	Automotive Industry	AUTOMOTIVE AXLES LTD	0.87631579
157	028087	4 Customers	FOCUS MINERALS LTD	0.78330928
158	022405	Party City Corp	PARTY CITY HOLDCO INC	0.925
159	036005	Not Reported	NORBORD INC	0.82857143
160	035604	Channel partners	CHANNEL ISLANDS PROPERTY FD	0.89351852
161	209382	Germany	GEFRAN SPA	0.87936508
162	064072	Chains and large format retailers	CHANG WAH ELECTROMATERIALS	0.84150029
163	014563	Tissue and specialty paper product manufacturers	EDUCATION REALTY TRUST INC	0.71338384
164	215422	Industry	INDUSTRONICS BHD	0.89166667
165	111535	Not Reported	NORBORD INC	0.82857143
166	185908	Aguettant	AUGEAN PLC	0.85
167	184378	International	INTERNATIONAL COAL GROUP INC	0.94444444
168	315629	Elan Microelectronincs Corp	ELAN MICROELECTRONINCS CORP	1
169	007985	U.S. Government	U.S. GOLD CORP	0.87142857
170	184070	Wuhan Kingold Industrial Group Co. Ltd.	WUHAN JINGCE ELECTRONIC GRP	0.85148148
171	020129	Asia Pacific	ASIA PACIFIC FUND	0.94117647
172	140044	Not Reported	NORBORD INC	0.82857143
173	066354	Tennessee	TENNESSEE GAS PIPELINE CO	0.88181818
174	009299	United States	UNITED STATES STEEL CORP	0.93684211
175	020959	FedEx Supply Chain, Inc.	FEDEX CORP	0.85555556
176	026839	Dollar Tree Inc	DOLLAR TREE INC	1
177	031802	Fiat Chrysler Automobiles NV	FIAT CHRYSLER AUTOMOBILES NV	1
178	170714	Sephora	SEPROD LTD	0.84777778
179	170969	Takeda Pharmaceutical Co	TAKEDA PHARMACEUTICAL CO LTD	1
180	011060	Americas (non-U.S.)	AMERICANN INC	0.9125
181	018850	Federal Deposit Insurance Corp.	FEDERAL INSURANCE	0.88894118

```
# 公司名称匹配.py文件
# -*- coding: utf-8 -*-
"""
Created on Mon Oct 24 23:40:34 2022

@author: jc
"""
import xlrd
import xlwt
import xlsxwriter
import re
import jellyfish



def spl_string(string):
    '''
    以空格为分隔符划分customer name
    消除大小写影响
    :param string:
    :return:
    '''
    string = re.sub(r'[-,/&()]|\'\sBD\.',r'', string) #去除customer name中的字符
    outcome = string.split()
    all_sp = ''
    for sp in outcome:
        all_sp += sp.lower()
    return all_sp



'''
从All Company data.xlsx 中读取公司名清理后的名字
Cleaned Full Name(E:清理后的名字)
'''

Allcompanydata_wb = xlrd.open_workbook(r'C:\Users\jc\Documents\大学'
                                +'\\0大三其他\\_金融数据挖掘科研课题'
                                +'\\PyProgram'
                                +'\All Company Data.xlsx')
Allcompanydata = Allcompanydata_wb.sheet_by_name('sheet1')
cleaned_name = Allcompanydata.col_values(colx = ord('E')-ord('A'), start_rowx = 1)
full_name = Allcompanydata.col_values(colx = 1, start_rowx = 1 )
key = Allcompanydata.col_values(colx = 0, start_rowx = 1 )


'''
从Abbreviation.xlsx中读取 客户缩写 与 上游公司KEY
'''
Abbreviation_wb = xlrd.open_workbook(r'C:\\Users\\jc'
                                      +'\\Documents\\大学\\0大三其他'
                                      +'\\_金融数据挖掘科研课题\\PyProgram'
                                      +'\Abbreviation Data.xlsx')
Abbreviation = Abbreviation_wb.sheet_by_name('sheet1')
clabbr = Abbreviation.col_values(colx = ord('E')-ord('A'), start_rowx = 1)
abbr = Abbreviation.col_values(colx = 1, start_rowx = 1)
upkey = Abbreviation.col_values(colx = 0, start_rowx = 1)
# 上游公司KEY 和 客户缩写 客户清理以后的缩写 组成元组（不可修改，并去掉重复的）
upkey_abbr = list(set(zip(upkey,abbr,clabbr)))
```

```
'''
创建保存匹配结果的表格Match companyname.xlsx,并创建表头
'''
Matchcpname_wb = xlwt.Workbook()
Matchcpname = Matchcpname_wb.add_sheet('sheet1')
name_list = ['Upstream key','customer abbreviation ','Full name','similarity',
              'Full name','similarity','Full name','similarity',]# 0 1

for i in range(len(name_list)):
    Matchcpname.write(0, i , name_list[i])

'''
第一列：Upstrean key ---- upkey_abbr[i][0]
第二列：customer abbreviation ---- upkey_abbr[i][1]
第三列及以后：  匹配出的公司全称 / JW算法算出的相似度
'''

# 1. 写入第一列Upstream key,第二列customer abbreviation
for i in range(len(upkey_abbr)):
    Matchcpname.write(i+1, 0 , upkey_abbr[i][0])
    Matchcpname.write(i+1, 1, upkey_abbr[i][1])

# 2.写入第三列及以后的数据
# (1) 遍历一遍 upkey_abbr 中的 upkey---ukabbr[0], clabbr---ukabbr[2]
for uk_abbr in upkey_abbr:
    indexnow = upkey_abbr.index(uk_abbr)  # 表示读取到第几个缩写了

    matchcp = []    # 用于临时储存匹配相似度大于0.3时：[公司名，相似度]
    # (2) 遍历一遍 cleaned_name 中的 clname
    for clname in cleaned_name:
        similarity = jellyfish.jaro_winkler_similarity(uk_abbr[2], clname)
        # 当相似度大于0.5时，写入临时数组中
        if(similarity > 0.5):
            matchcp.append([clname,similarity])

    #当没有匹配到公司时，跳过本次写入
    if len(matchcp) == 0 :
        continue

       # 按匹配度顺序从大到小排序，并取出前三的[公司名，相似度]
    matchcp = sorted(matchcp, key=lambda matchcp:matchcp[1],reverse =True)

    #如果没有三个的话，就全写进去
    if len(matchcp) < 3:
        col = 0
        for i,j in matchcp:
            Matchcpname.write(indexnow+1, col+2,
                              full_name[cleaned_name.index(i)])
            Matchcpname.write(indexnow+1, col+3, j )
            col += 2
    else:
    # 如果大于3个的话，就只写前三个
        for i,j in zip(range(0,5,2),range(3)):
            Matchcpname.write(indexnow+1, i+2,
                              full_name[cleaned_name.index(matchcp[j][0])])
            Matchcpname.write(indexnow+1, i+3,
                              matchcp[j][1])



Matchcpname_wb.save('Match Company name.xlsx')
```