# import pandas as pd

# import sqlite3

# conn = sqlite3.connect(r"/Users/shubhendu.abhinav/Desktop/sql files/1690174445.164308result.db")

# df = pd.read\_sql("select \* from result", conn)

# df.to\_csv(r"/Users/shubhendu.abhinav/Desktop/outputfiles/25julycontactvalidation.csv", index=False)

import os

import sqlite3

import pandas as pd

dfs = []

# reading all files in the directory

all\_files = os.listdir(r"/Users/shubhendu.abhinav/Desktop/sql files")

print(all\_files)

for file in all\_files:

if file.endswith('.db')==True:

con = sqlite3.connect(r"/Users/shubhendu.abhinav/Desktop/sql files/" + file)

dfs.append(pd.read\_sql("select \* from contacts", con))

print(file)

## final consolidated file

final\_file = pd.concat(dfs,ignore\_index=True)

final\_file.to\_csv(r"/Users/shubhendu.abhinav/Desktop/outputfiles/contactvalidation8august.csv", index=False)

#

#

#

# # Splitting file into 2 files

# file1 = final\_file[['Code', 'Company\_Input', 'Main\_Title', 'Main\_Link']]

# file2 = final\_file[['Code', 'Company\_Input', 'Other\_Title', 'Other\_Link']]

#

# # Renaming columns of file 2

# file2.rename(columns={'Other\_Title':'Main\_Title','Other\_Link':'Main\_Link'},inplace=True)

#

#

# # Consolidating file

# final\_file\_v2 = pd.concat([file1,file2])

#

# # Removing duplicates based on all columns value

# final\_file\_v2.drop\_duplicates(inplace=True)

#

# # Keeping only required columns

# final\_file\_v2 = final\_file\_v2[['Code', 'Company\_Input', 'Main\_Link']]

#

#

# #function to extract account name from zoominfo url

# def company\_extractor(url):

# try:

# url = str(url)

# url = url.split('/')[-1]

# url = url.replace('-',' ')

# url = url.title().strip()

# return url

# except:

# return ".."

#

# url = 'https://www.linkedin.com/company/terminus-account-based-marketing/mycompany/verification/'

# # print(company\_extractor(url))

#

#

# # running company extractor function and adding result in new column

# final\_file\_v2['output\_company']= final\_file\_v2.Main\_Link.apply(company\_extractor)

#

#

#

# # renaming columns to run accounts matching script

# final\_file\_v2.rename(columns={'Code':'code','Company\_Input':'account1','output\_company':'account2'},inplace=True)

# final\_file\_v2.to\_csv(r"/Users/shubhendu.abhinav/Desktop/outputfiles/linkedin.csv", index=False)

#

#

#

#

# # import pandas as pd

# # import numpy as np

# # path = r"/Users/shubhendu.abhinav/Desktop/tactic/new notmatch old6.5m data/Not Match1.csv"

# # df = pd.read\_csv(path)

# # # df2 = df.drop(['ORB\_NUM', 'ENTITY\_TYPE'], axis=1)

# # print(df.columns)

# # # df2.to\_csv(r"/Users/shubhendu.abhinav/Desktop/tactic/old/afterdeletecolumnnewdata.csv")

#

# # import os

# # import sqlite3

# # import pandas as pd

# # dfs = []

# #

# # # reading all files in the directory

# # all\_files = os.listdir(r"/Users/shubhendu.abhinav/Desktop/sql files")

# # print(all\_files)

# # for file in all\_files:

# # if file.endswith('.db')==True:

# # con = sqlite3.connect(r"/Users/shubhendu.abhinav/Desktop/sql files/" + file)

# # dfs.append(pd.read\_sql("select \* from contacts", con))

# #

# # print(file)

# # ## final consolidated file

# # final\_file = pd.concat(dfs,ignore\_index=True)

# # final\_file.to\_csv(r"/Users/shubhendu.abhinav/Desktop/outputfiles/contacts30aprilshu.csv", index=False)

# # # import pandas as pd

# # # final\_file = pd.read\_csv(r"/Users/shubhendu.abhinav/Desktop/untouched zmi/consolidated\_db.csv")

# # # final\_file = pd.concat(dfs,ignore\_index=True)

# # # # print(final\_file)

# # #

# # # # Splitting file into 2 files

# # # file1 = final\_file[['Code', 'Company\_Input', 'Main\_Title', 'Main\_Link']]

# # # file2 = final\_file[['Code', 'Company\_Input', 'Other\_Title', 'Other\_Link']]

# # #

# # # # Renaming columns of file 2

# # # file2.rename(columns={'Other\_Title':'Main\_Title','Other\_Link':'Main\_Link'},inplace=True)

# # #

# # #

# # # # Consolidating file

# # # final\_file\_v2 = pd.concat([file1,file2])

# # #

# # # # Removing duplicates based on all columns value

# # # final\_file\_v2.drop\_duplicates(inplace=True)

# # #

# # # # Keeping only required columns

# # # final\_file\_v2 = final\_file\_v2[['Code', 'Company\_Input', 'Main\_Link']]

# # #

# # #

# # # #function to extract account name from zoominfo url

# # # def company\_extractor(url):

# # # try:

# # # url = str(url)

# # # url = url.split('/')[-2]

# # # url = url.replace('-',' ')

# # # url = url.title().strip()

# # # return url

# # # except:

# # # return ".."

# # #

# # # url = 'https://www.zoominfo.com/c/terminus-software-inc/363561950'

# # # # print(company\_extractor(url))

# # #

# # #

# # # # running company extractor function and adding result in new column

# # # final\_file\_v2['output\_company']= final\_file\_v2.Main\_Link.apply(company\_extractor)

# # # final\_file\_v2.rename(columns={'Code':'code','Company\_Input':'account1','output\_company':'account2'},inplace=True)

# # # final\_file\_v2.to\_csv(r"/Users/shubhendu.abhinav/Desktop/outputfiles/accounts17march.csv", index=False)

import pandas as pd

import numpy as np

from pandasql import sqldf

import matplotlib.pyplot as plt

# df = pd.read\_csv(r"/Users/shubhendu.abhinav/Desktop/Mig\_Batch\_34\_10April\_23.csv",encoding = 'latin-1')

df = pd.read\_csv(r"/Users/shubhendu.abhinav/Desktop/nameage.csv")

# data = df.groupby('REGION')['COUNTRY'].nunique()

# data.plot(kind='piechart')

# plt.xlabel('REGION')

# plt.ylabel('Number of countries')

# plt.title('countries per region')

# plt.show()

# df1 = df.loc[0:5,['ACCOUNT\_NAME']]

# print(df.columns)

# df1 = df.groupby('COUNTRY').nunique().to\_clipboard()

# df1 = df.drop(columns=['LOCATION\_TAG','LOCATION\_ID'])

# filtercountry = df[df['COUNTRY'].isin(['United States'])]

# print(filtercountry)

# # print(df.dtypes)

# # print(df.info())

# # print(df.iloc[0:2])

# columns = ['ACCOUNT\_NAME', 'WEBSITE', 'PHONE']

# df1 = df[columns].nunique().to\_clipboard()

# df1 = df.groupby('REGION')['INDUSTRY','SUB\_INDUSTRY','EMPLOYEE\_RANGE','WEBSITE'].count().to\_clipboard()

# print(df[columns].to\_clipboard())

#

# # df1 = df['ACCOUNT\_NAME'].value\_counts().to\_clipboard()

# print(df[columns].counts().to\_clipboard())

# # column\_names = ['ACCOUNT\_NAME', 'WEBSITE', 'PHONE']

# df['status'] = 'notmatch'

# print(df.head())

# columns = ['United States','Canada']

# columns2 = ['bank']

# # df['new status'] = np.where(df['COUNTRY'].isin(columns),'Found', 'Not Found')

# df1 = df(df['COUNTRY'].isin(columns) & (df['INDUSTRY'].isin(columns2)))

#

# df1.to\_csv(r"/Users/shubhendu.abhinav/Desktop/Mydata1.csv")

# search\_results = df[df['COUNTRY'].str.contains('united', case=False)].to\_clipboard()

# df = df.rename(columns={'TE\_ACCOUNT\_ID': 'newTE\_ACCOUNT\_ID'}).to\_clipboard()

# df1 = df.apply(lambda x: x.str.strip() if x.dtype == "object" else x)

# df1.to\_csv(r"/Users/shubhendu.abhinav/Desktop/Mydata1.csv")

# columns\_to\_clean = ['ACCOUNT\_NAME','WEBSITE']

# df1 = df['ACCOUNT\_NAME'].str.strip()

# for column in columns\_to\_clean:

# df[column] = df[column].str.strip()

# print(df)

# df1 = df[['ACCOUNT\_NAME','ZIP\_CODE']]

# print(df1)

# unique\_count = df[['ACCOUNT\_NAME','WEBSITE']].nunique()

# print(unique\_count)

# df2 = df.drop\_duplicates(subset='ZIP\_CODE', keep="first")

# # print(df2.count())

# unique\_count = df2['ZIP\_CODE'].nunique()

# print(unique\_count)

# df['Label'] = df.duplicated(subset=['ACCOUNT\_NAME', 'WEBSITE'], keep=False)

# df['Label'] = df['Label'].map({True: 'Duplicate', False: 'Unique'})

# df.to\_csv(r"/Users/shubhendu.abhinav/Desktop/Mydata1.csv")

# df\_sorted = df.sort\_values('ACCOUNT\_NAME',ascending=False)

# df\_sorted = df.sort\_values(by=['ACCOUNT\_NAME', 'WEBSITE'])

# df\_sorted = df.sort\_values(by=['ACCOUNT\_NAME', 'WEBSITE'],ascending=[False, True]) #it will sort first column in descending order nd second in ascending ordder

#

# print(df\_sorted.head(10))

# duplicate\_rows = df.duplicated().to\_clipboard()

# df1 = df['TE\_ACCOUNT\_ID'].duplicated()

# idduplicate = list(df1)

# df['idduplicate'] = idduplicate

# print(df.count())

# df1 = df[['ACCOUNT\_NAME','WEBSITE']].drop\_duplicates()

# # removeduplicate = list(df1)

# # df['removeduplicate'] = removeduplicate

# df1 = df.drop\_duplicates(['ACCOUNT\_NAME','WEBSITE'], keep="first")

# print(df.nunique())

# print(df1.nunique())

#

#

# df1.to\_csv(r"/Users/shubhendu.abhinav/Desktop/shu2.csv",index = False)

#

# print(duplicate\_rows)

# missing\_data = df[['ACCOUNT\_NAME','STATE']].isnull().sum()# or df['column\_name'].isnull()

# print(missing\_data)

# missing\_data1 = df.isnull().sum()# or df['column\_name'].isnull()

# print(missing\_data1)

# print(df.count())

# print(missing\_data1)

# print(df)

# df1 = df['ACCOUNT\_NAME'].value\_counts().to\_clipboard()

# print(df1)

# df['ACCOUNT\_NAME'] = df['ACCOUNT\_NAME'].str.upper()

# df['WEBSITE'] = df['WEBSITE'].str.upper()

# df.to\_csv(r"/Users/shubhendu.abhinav/Desktop/shu2.csv",index = False)

# df1 = df['gender'].value\_counts()

# male\_employees = df[df['gender'] == 'M']

# df['maleemployee'] = male\_employees

# female\_employee = df[df['gender'] == 'F']

# print(male\_employees)

# morethen = male\_employees[male\_employees['age'] >20]

# average\_age = morethen['age'].mean()

# average\_age = male\_employees['age'].value\_counts()

# print("Average age of male employees who are grater then 20:", average\_age)

# df1 = df.groupby('first name')['gender','age'].value\_counts()

# average\_age = df['age'].mean()

# df['Average\_age'] = average\_age

# Total\_age = df['age'].sum()

# df['Total\_age'] = Total\_age

# average\_ageman = male\_employees['age'].mean()

# df['maleavergeage'] = average\_ageman

# average\_agefemale = female\_employee['age'].mean()

# df['womenavfage'] = average\_agefemale

# print(df)

# df['add'] = df['maleavergeage']+df['womenavfage']

# print(df['add'].drop)

# df['fullname'] = ''

# df['fullname'] = df['first name'] + ' ' + df['last name']

# morethenavergae = df[df['male\_employees'] > df['maleavergeage']]

# print(morethenavergae)

# print(df)

# result = df.query('age > 20 and (gender == "M" and first\_name == "joe")')

# print(result)

# print(df1)

# print(df1.columns)

# con = ['Oman','United States','Belgium']

# df1['EMPLOYEE\_RANGE'] = pd.to\_numeric(df1['EMPLOYEE\_RANGE'], errors='coerce')

# Filter the DataFrame based on the specified conditions

# filtered\_df = df1.query('(COUNTRY in ["Oman","United States","Belgium"]) and INDUSTRY == "Banks" and EMPLOYEE\_RANGE >= 50 and EMPLOYEE\_RANGE <= 400')

# filtered\_df = df1.query('(COUNTRY in ["Oman","United States","Belgium"]) and INDUSTRY == "Banks" and EMPLOYEE\_RANGE >=10 and EMPLOYEE\_RANGE <= 200')

# Print the filtered DataFrame

# print(filtered\_df.EMPLOYEE\_RANGE)

# conditions = (

# (df1['COUNTRY'].isin(['Oman', 'United States', 'Belgium'])) &

# (df1['INDUSTRY'] == 'Banks') &

# (df1['EMPLOYEE\_RANGE']==40)

# )

# print(conditions)

# print(df.columns)

# df1 = df[['ACCOUNT\_NAME','CITY']]

# df2 = df1.query('CITY == "Harare"')

# df2 = df[df['CITY'] == "Harare"].nunique()

# print(df2)

# df.dropna(inplace=True)

# print(df.to\_string())

# df1 = df.fillna(130)

# # print(df1.to\_string())

# df1.to\_excel(r"/Users/shubhendu.abhinav/Desktop/Mydata1.xlsx")

# print(df.to\_string())

# df1 = df.loc[0:4,['ids','gender']]

# df1 = df.loc[0:3,:]

# print(df1)

# df1 = df[(df['gender']=='M' & (df['age'] < 30)]

df1 = df[(df['gender'] == 'M') & (df['age'] < 30) & (df['ids'] ==1)]

print(df1)

from readinig\_input import readinginput

from parsing\_scraped\_result import htmlparser

from selenium.webdriver.common.by import By

from selenium import webdriver

import time

from webdriver\_manager.chrome import ChromeDriverManager

from selenium.webdriver.support import expected\_conditions as EC

from selenium.webdriver.support.wait import WebDriverWait

import sqlite3

import os

from datetime import datetime

from os.path import join

curr\_dir = os.path.dirname(os.path.abspath(\_\_file\_\_))

current\_time = datetime.now().strftime("%d\_%b\_%Y\_%H\_%M")

def browser\_instance():

""" Starts chrome browser instance to logins to linkedin.

Param:

None

Returns:

object: chrome driver instance.

"""

try:

#driver = webdriver.Chrome()

driver = webdriver.Chrome(ChromeDriverManager().install())

driver.get('https://www.linkedin.com/uas/login')

element = WebDriverWait(driver, 200).until(EC.presence\_of\_element\_located((By.ID, "global-nav-search")))

driver.get("https://www.linkedin.com/sales/search/people?viewAllFilters=true")

print("logged in successfully")

except:

raise Exception ("Could not login")

return driver

def output\_db():

"""Creates Sqlite db and table (Code, Input\_Company, Contact\_Name, Title, Output\_Company, Location, Profile\_url, More\_Employees, Output\_company\_Url) to store scraped result.

Param:

None

Returns:

object: db connetion

"""

con = sqlite3.connect(join(curr\_dir, "database", current\_time +".db"))

cur = con.cursor()

cur.execute("create table if not exists contacts(Code,Input\_Company,Contact\_Name,Title,Output\_Company,Location,Profile\_url,More\_Employees,Output\_company\_Url)")

return con

def scrapingcontacts():

"""Reads input, scrapes contacts and saves in sqlite db.

"""

inputdata = readinginput() # Reading input file from folder

company = inputdata[0] # Reading all companies in the list

filter = inputdata[1] # Reading all filters in the list

code = inputdata[2] # Reading all code in the list

driver = browser\_instance() # Browser instance

db\_connection = output\_db() # sqlite db instace

# cursor = db.cursor()

time.sleep(5)

for eachcompany, eachfilter, eachcode in zip(company,filter,code):

time.sleep(3)

driver.get(eachfilter)

time.sleep(3)

try:

element\_count = len(driver.find\_element(By.ID,"search-results-container").find\_elements(By.CLASS\_NAME,"artdeco-list\_\_item"))

driver.find\_element(By.ID,"search-results-container").find\_elements(By.CLASS\_NAME,"artdeco-list\_\_item")[element\_count-1].location\_once\_scrolled\_into\_view

time.sleep(5)

driver.find\_element(By.ID,"search-results-container").find\_elements(By.CLASS\_NAME,"artdeco-list\_\_item")[0].location\_once\_scrolled\_into\_view

time.sleep(5)

for each\_element in range(0,element\_count):

driver.find\_element(By.ID,"search-results-container").find\_elements(By.CLASS\_NAME,"artdeco-list\_\_item")[each\_element].location\_once\_scrolled\_into\_view

source = driver.find\_element(By.ID,"search-results-container").find\_elements(By.CLASS\_NAME,"artdeco-list\_\_item")[each\_element].get\_attribute('innerHTML')

htmlparser(db\_connection, source,eachcode,eachcompany) # passing html to function to scrape and save into sqlite db.

except:

pass

scrapingcontacts()

df.rename(columns= {'Marital\_Status':'Shaadi'})

# describe() method returns description of the data in the DataFrame (i.e. count, mean, std, etc)

df.describe()

pd.isnull(df).sum()

df.dropna(inplace=True)

df['Amount'] = df['Amount'].astype('int')

df[['Age', 'Orders', 'Amount']].describe()

df.to\_string

df.nunique()

df.isnull().sum()

df.notnull().sum()

df['Weather'].unique()

df.nunique()

df.isnull().sum()

df.notnull().sum()

df.count()

df['Wind Speed\_km/h'].nunique()

df[df['Weather'] == 'Clear']

df.groupby('Weather').get\_group('Clear')

df[df['Wind Speed\_km/h'] ==4]

df['Weather'].value\_counts()

df[df['Weather'].str.contains('Snow')].tail(10)

df[(df['Wind Speed\_km/h']>=24) & (df['Visibility\_km']==25)]

df.groupby('Weather')['Rel Hum\_%'].count()

df.groupby('Weather').count()

df[(df['Weather'] == 'Clear') & (df['Visibility\_km']>=40)].to\_string

df3.drop(columns=['country\_name'],inplace=True)

df3.groupby('violation')['driver\_age','driver\_gender'].count()

df4['Region'].value\_counts()

df4.sample(n=10)

df4.groupby('Region')['Confirmed'].sum().sort\_values(ascending=False)

df5['date'] = pd.to\_datetime(df5['date'])

df5['year'] = df5['date'].dt.year

df5.drop(['year','month'],axis=1)

import pandas as pd

import numpy as np

import xlrd

# path = r"/Users/shubhendu.abhinav/Desktop/AutoRabit\_Contact.csv"

path = r"/Users/shubhendu.abhinav/Desktop/AutoRabit\_Account.csv"

# df = pd.read\_csv(path,encoding = 'unicode\_escape',nrows=2)

# want toprint only 2 columns that is column no 2 and 5

# df = pd.read\_csv(path,encoding = 'unicode\_escape',usecols=[2,5])

# want to skip some rows

# df = pd.read\_csv(path,encoding = 'unicode\_escape',skiprows=[0,5])

# want to make email column as a special index col

# df = pd.read\_csv(path,encoding = 'unicode\_escape',index\_col="Email")

# to get the columns name

# df = pd.read\_csv(path,encoding = 'unicode\_escape')

# print(df.columns)

# sort complete data into desending order

# df = pd.read\_csv(path,encoding = 'unicode\_escape',usecols=['Email','First Name'])

# df1 = df.sort\_index(axis = 0,ascending=False)

# sort specific column

# df1 = df.sort\_values(by = ['First Name'],ascending=False)

# df1.to\_excel(r"/Users/shubhendu.abhinav/Desktop/sort.xlsx")

# want to see only us and india data

# df = pd.read\_csv(path,encoding = 'unicode\_escape',index\_col = "Billing Country" )

# data = df.loc[["United States","India"]]

# replce particular value

df = pd.read\_csv(path,encoding = 'unicode\_escape',usecols=['Account Name','Billing Country','Industry'])

# data = df.replace(to\_replace="United States",value="SouthAfrcia")

# replace specific column value

# df['Billing Country'] = df['Billing Country'].replace(['United States'],'Kenya')

# print(df['Billing Country'])

df1 = df.groupby('Billing Country').value\_counts()

#

df1.to\_excel(r"/Users/shubhendu.abhinav/Desktop/sort.xlsx")

import re

import pandas as pd

path = r"/Users/shubhendu.abhinav/Desktop/email\_list.csv"

data = pd.read\_csv(path, encoding = 'unicode\_escape')

df1 = pd.DataFrame()

df = data.iloc[:, 0]

flat\_emails = df.values.flatten()

emails = list(map(lambda x: x.lower(), flat\_emails))

bad\_emails\_count = 0

bad\_emails = []

for email in emails:

match = re.match('^[\_a-z0-9-]+(\.[\_a-z0-9-]+)\*@[a-z0-9-]+(\.[a-z0-9-]+)\*(\.[a-z]{2,4})$', email)

if match == None:

print("%s has bad syntax" % (email))

bad\_emails\_count += 1

bad\_emails.append("bad")

else:

bad\_emails\_count += 1

bad\_emails.append('ok')

df1["email"] = emails

df1["status"] = bad\_emails

# print("---------------------%d emails are not correct----------------------------" % (bad\_emails\_count))

# print("---------------------------------------------------------------------------")

# print(bad\_emails)

df1.to\_csv(r"/Users/shubhendu.abhinav/Desktop/Mydata1.csv")

# Mydata1 = pd.read\_csv('Mydata1.csv')

import pandas as pd

import time

import numpy as np

from selenium import webdriver

from selenium.webdriver.common.keys import Keys

from webdriver\_manager.chrome import ChromeDriverManager

from selenium.webdriver.common.by import By

path = r"/Users/shubhendu.abhinav/Desktop/url.csv"

df = pd.read\_csv(path, encoding = 'unicode\_escape')

df1 = df['Id'].to\_list()

df2 = df['Name'].to\_list()

list2 = []

# list3 = []

#company\_list = df['Name'].values.tolist()

company\_list = df['Name'].tolist()

#print(", ".join(company\_list))

# driver = webdriver.Chrome(ChromeDriverManager().install())

driver = webdriver.Safari()

for company\_list in company\_list:

print(company\_list)

# driver.get('http://google.com/search?q='+str(company\_list) + ' website')

driver.get('http://google.com/search?q='+str(company\_list) + ' phone')

time.sleep(2)

# outname = driver.find\_element(By.CLASS\_NAME,'yuRUbf').text

outname = driver.find\_element(By.CLASS\_NAME,'LrzXr').text

print(outname)

# url = driver.find\_element(By.CLASS\_NAME, 'yuRUbf').find\_element(By.TAG\_NAME, 'a').get\_attribute('href')

# print(url)

list2.append(outname)

# list3.append(url)

time.sleep(3)

df["Name"] = df2

df['outname'] = list2

# df["url"] = list3

#time.sleep(30)

df.to\_csv(r"/Users/shubhendu.abhinav/Desktop/mydata.csv",index = False)

import phonenumbers

import pandas as pd

path = r"/Users/shubhendu.abhinav/Desktop/phone.csv"

df = pd.read\_csv(path, encoding = 'unicode\_escape')

df1 = pd.DataFrame()

df['result'] = df.apply(lambda x: phonenumbers.is\_valid\_number(phonenumbers.parse("+"+str(x.phone), None)), axis=1)

print(df)

df.to\_csv(r"/Users/shubhendu.abhinav/Desktop/Mydata1.csv")

# # #Mydata1 = pd.read\_csv('Mydata1.csv')

# import pandas as pd

# path = r"/Users/shubhendu.abhinav/Desktop/Relias Contact File.csv"

# df1 = pd.read\_csv(path, encoding = 'unicode\_escape')

# data = df1[df1.Title.isnull()== False]

# data.to\_csv(r"/Users/shubhendu.abhinav/Desktop/title.csv")

import os

import sqlite3

import pandas as pd

dfs = []

# reading all files in the directory

all\_files = os.listdir(r"/Users/shubhendu.abhinav/Desktop/untouched zmi")

print(all\_files)

for file in all\_files:

if file.endswith('.db')==True:

con = sqlite3.connect(r"/Users/shubhendu.abhinav/Desktop/untouched zmi/" + file)

dfs.append(pd.read\_sql("select \* from result", con))

print(file)

## final consolidated file

final\_file = pd.concat(dfs,ignore\_index=True)

print(final\_file)

# Splitting file into 2 files

file1 = final\_file[['Code', 'Company\_Input', 'Main\_Title', 'Main\_Link']]

file2 = final\_file[['Code', 'Company\_Input', 'Other\_Title', 'Other\_Link']]

# Renaming columns of file 2

file2.rename(columns={'Other\_Title':'Main\_Title','Other\_Link':'Main\_Link'},inplace=True)

# Consolidating file

final\_file\_v2 = pd.concat([file1,file2])

# Removing duplicates based on all columns value

final\_file\_v2.drop\_duplicates(inplace=True)

# Keeping only required columns

final\_file\_v2 = final\_file\_v2[['Code', 'Company\_Input', 'Main\_Link']]

#function to extract account name from zoominfo url

def company\_extractor(url):

try:

url = str(url)

url = url.split('/')[-2]

url = url.replace('-',' ')

url = url.title().strip()

return url

except:

return ".."

url = 'https://www.zoominfo.com/c/terminus-software-inc/363561950'

# print(company\_extractor(url))

# running company extractor function and adding result in new column

final\_file\_v2['output\_company']= final\_file\_v2.Main\_Link.apply(company\_extractor)

# renaming columns to run accounts matching script

final\_file\_v2.rename(columns={'Code':'code','Company\_Input':'account1','output\_company':'account2'},inplace=True)

final\_file\_v2.to\_csv(r"/Users/shubhendu.abhinav/Desktop/outputfiles/26apr.csv", index=False)

# import pandas as pd

# path = r"/Users/shubhendu.abhinav/Desktop/Relias Contact File.csv"

#

# df = pd.read\_csv(path, encoding = 'unicode\_escape',chunksize=900000)

# # # df1 = df.temocode.drop\_duplicates()

# # # print(len(df1))

# # data = df['Contact ID'].drop\_duplicates().to\_list()

# # print(len(set(data)))

# x = 1

# for each in df:

# each.to\_csv(r"/Users/shubhendu.abhinav/Desktop/cdpaudit"+str(x)+".csv")

# x = x+1

# import pandas as pd

# # # # import openpyxl

# path = r"/Users/shubhendu.abhinav/Desktop/23augustconsolidate.csv"

# df1 = pd.read\_csv(path)

# # # # # df2 = pd.read\_excel(r"/Users/shubhendu.abhinav/Desktop/relias folder/reliascontact2.xlsx")

# # # # # df3 = pd.read\_excel(r"/Users/shubhendu.abhinav/Desktop/relias folder/reliascontact3.xlsx")

# # # # #

# # # # # dfw = pd.concat([df1,df2,df3])

# print(len(df1))

# # code = []

# # for x in range(1,2501841):

# # code.append('TAIDOOO'+str(x))

# # # print(len(code))

# # df1['temocode'] = code

# # # print(df1.columns)

# # df1 = df1[['temocode','Contact ID', 'First Name', 'Last Name', 'Salutation', 'Email', 'Title','Account ID']]

# # print(len(df1))

# df1.to\_excel(r"/Users/shubhendu.abhinav/Desktop/newconsoli.xlsx",index = False)

# a = [10,20,30,40]

# b = len(a)

# # print(b)

# for n in range(b-1,-1,-1):

# print(a[n])

# a = "welcome"

# b = len(a)

# for n in range(b-1,-1,-1):

# print(a[n])

# else:

# print("program is complete")

#

# # def mydata(a,b):

# # c = a+b

# # return c

# # output = mydata(10,20)

# # print(output)

import pandas as pd

import numpy as np

df = pd.read\_excel(r"/Users/shubhendu.abhinav/Desktop/google/3k\_google.xlsx")

# df1 = df[['result','Company']]

# df1 = df.loc[3,'First line Up']

df1 = df.loc[0:4]

# df1 = df.loc[0:4,["Company","Job Title"]]

# df1 = df.iloc[0:4,1:3]

print(df1)

# df['pass'] = 'Nan'

# df1 = df.replace('Nan','india')

# df = df.drop(columns = 'pass')

# print(df)

# df1 = df[(df["Location"] == "Pakistan") & (df["result"] == "MatchRemovedEntitytype") & (df["Job Title"].str.contains("manager"))]

# df1.to\_excel(r"/Users/shubhendu.abhinav/Desktop/sort.xlsx")