The code is built on a jupyter notebook on gcp. It it about the file modification. And it solves the problem of memory out when reading csv file and modifying data frame by reading by lines and split file into chunks

```
import pandas as pd
import gcsfs
import csv
import google.auth
import numpy as np
from datetime import datetime


currentMonth = datetime.now().month
currentYear = datetime.now().year
currentDay = datetime.now().day

#import boto3
#import dask.dataframe as dd

project = google.auth.default()[1]
fs = gcsfs.GCSFileSystem(project=project)
#file contains 10000003 rows,46columns

#load and separate file into 30 chunks

## read file from gcs
file_path="gs://bucketpath/filename"
csvfile=fs.open(file_path)

lines=list()

with fs.open(file_path, 'r', encoding='latin1') as readFile:

    reader = csv.reader(readFile)

    for row in reader:

        lines.append(row)

nrow = len(lines)
nrow_without_header=nrow-1
```

```
#for each chunk:
### split content array into 30 chunks
### turn split arrays into dataframe and modify it(use the same name for dataframes to save
memory)
### save dataframe into a csv file
df=pd.DataFrame(lines)
df=df.iloc[:,0]
headers = df.iloc[0]
content = df.values[1:]
# split content array into 30 chunks (0 to 29)
import numpy as np
a=np.array_split(content,30)
#put the header into csvfile
header_final = [ ['CustomerID',
'FIRSTNAME','LASTNAME','ADDRESS1','TOWN','POSTCODE','EMAIL1','MOBILE1','LAN
DLINE1','Client','Prospect','PossesseurVN','PossesseurVO','PossesseurVP','PossesseurVU','Posse
ssionInf12Mois','Possession30a36Mois','Possession42a48Mois','PossessionInf4ans','Possession4
a8ans','PossessionSup8ans','Proprietaire','Locataire','PossesseurRenault','PossesseurAutreMarque
PSA','PossesseurSegmentA','PossesseurSegmentB','PossesseurSegmentC','PossesseurSegmentD','
PossesseurBerline','PossesseurSUV','PossesseurSW','PossesseurMonospace','PossesseurEssence','
PossesseurDiesel','PossesseurElectrique','PossesseurHybride','PossesseurPremium','PossesseurM
ainstream','PossesseurEntry','PossessionProcheDateAnniv','PossesseurContratEntretien']]
df_header=pd.DataFrame(header_final)
df_header
csvnew="/home/jupyter/filename"
df_header.to_csv(csvnew, sep='|',index = False, header=False)

##file modificaiton for each chunk (30 loops)
def dataframe_modification():
    for i in range(30):
        ## modification of dataframe
        # turn split arrays into dataframe and modify it (a[0] to a[29])
        #select first 46 columns (ignore extra columns of error lines)
        df_chunk=(pd.DataFrame(pd.Series(a[i]).str.split('|').tolist())).iloc[:,0:46]
        # group columns 4,5,6,7 and put the new column to position 4
        df_chunk[df_chunk.columns[4]]=df_chunk[df_chunk.columns[4:8]].apply(lambda x: '
'.join(x.dropna().astype(str)),axis=1)
        # deletd: column
CIVILITE,ADRESSE_LIGNE_2,ADRESSE_LIGNE_3,ADRESSE_LIGNE_4
```

```
df_chunk=df_chunk.drop([df_chunk.columns[1],df_chunk.columns[5],df_chunk.columns[6],df_
chunk.columns[7]],axis=1)
      # reorder: change the column order according to the header order
      order = [0,3,2,4,9,8,10,12,11,13, 14, 15, 16, 17, 18, 19, 20,21, 22, 23, 24, 25, 26, 27, 28, 29,
30, 31, 32, 33, 34, 35, 36, 37,38, 39, 40, 41, 42, 43, 44, 45]
      df_chunk = df_chunk[order]
      #df_chunk.head()
      # put the chunk into a csv file
      df_chunk.to_csv("/home/jupyter/Jin_ok.psv", sep='|', mode='a',index = False, header=False)
      CPL(df_chunk)

##file modificaiton for each chunk (30 loops)
def dataframe_modification():
   for i in range(30):
      ## modification of dataframe
      # turn split arrays into dataframe and modify it (a[0] to a[29])
      #select first 46 columns (ignore extra columns of error lines)
      df_chunk=(pd.DataFrame(pd.Series(a[i]).str.split('|').tolist())).iloc[:,0:46]
      # group columns 4,5,6,7 and put the new column to position 4
      df_chunk[df_chunk.columns[4]]=df_chunk[df_chunk.columns[4:8]].apply(lambda x: '
'.join(x.dropna().astype(str)),axis=1)
      # deletd: column
CIVILITE,ADRESSE_LIGNE_2,ADRESSE_LIGNE_3,ADRESSE_LIGNE_4

df_chunk=df_chunk.drop([df_chunk.columns[1],df_chunk.columns[5],df_chunk.columns[6],df_
chunk.columns[7]],axis=1)
      # reorder: change the column order according to the header order
      order = [0,3,2,4,9,8,10,12,11,13, 14, 15, 16, 17, 18, 19, 20,21, 22, 23, 24, 25, 26, 27, 28, 29,
30, 31, 32, 33, 34, 35, 36, 37,38, 39, 40, 41, 42, 43, 44, 45]
      df_chunk = df_chunk[order]
      #df_chunk.head()
      # put the chunk into a csv file
      df_chunk.to_csv("/home/jupyter/filename", sep='|', mode='a',index = False, header=False)
      CPL(df_chunk)

#Execution (modification + CPL)
dataframe_modification()

#Final CPL

cplfile_csv="/home/jupyter/CPL_final_%s%s%s.csv" % (currentDay,currentMonth,currentYear)
df_cpl=pd.read_csv(cplfile_csv)
```

```
droplist=[]
for i in range(1,len(df_cpl)):
   if  i % 5 ==0:
      droplist.append(i-1)
#print(droplist)

df_cpl=df_cpl.drop(df_cpl.index[droplist])

for col in range (2,8):
   df_cpl[df_cpl.columns[col]]=pd.to_numeric(df_cpl[df_cpl.columns[col]])

dtype=df_cpl.groupby(by=df_cpl.columns[0])['dtype'].unique()
cpl_No_notzero_notnull=df_cpl.groupby(by=df_cpl.columns[0])
['cpl_No_notzero_notnull'].sum()
cpl_No_notnull=df_cpl.groupby(by=df_cpl.columns[0])['cpl_No_notzero_notnull'].sum()
cpl_percen_notnull=cpl_No_notnull/nrow_without_header*100.0
cpl_percen_notzero_notnull=cpl_No_notzero_notnull/nrow_without_header*100
MaxLen=df_cpl.groupby(by=df_cpl.columns[0])['MaxLen'].max()
MinLen=df_cpl.groupby(by=df_cpl.columns[0])['MinLen'].min()

df_cpl_final=pd.DataFrame(dtype)
df_cpl_final[1]=pd.DataFrame(cpl_No_notzero_notnull)
df_cpl_final[2]=pd.DataFrame(cpl_percen_notzero_notnull)
df_cpl_final[3]=pd.DataFrame(cpl_No_notnull)
df_cpl_final[4]=pd.DataFrame(cpl_percen_notnull)
df_cpl_final[5]=pd.DataFrame(MaxLen)
df_cpl_final[6]=pd.DataFrame(MinLen)
df_cpl_final.columns
=['dtype','cpl_No_notzero_notnull','cpl_percen_notzero_notnull','cpl_No_notnull','cpl_percen_no
tnull','MaxLen','MinLen']
df_cpl_final.index=['TOWN','POSTCODE','EMAIL1','MOBILE1']
cplfile_final_csv="/home/jupyter/CPL_final_%s%s%s.csv" %
(currentDay,currentMonth,currentYear)
df_cpl_final.to_csv(cplfile_final_csv,mode='a',index = True, header= True)

###pb0:to a single file(not supported by gcp)-failed
#df_final.to_csv(csvnew,sep='|',index = False,encoding='latin1',single_file = True,header=True)

###pb1: can't process all lines at once, otherwise memoryout error;
###solution1: cut the file into 30 chunks, modify each;

###pb2: can't append dataframe to csv file in gcs (file in gcs is immutable)
```

###solution2: put the file into jupyternotebook first and then upload it in gcs

###pb3: can't install boto3 (send file to aws);

!gsutil cp /home/jupyter/filename gs://bucketpath/

!gsutil cp /home/jupyter/CPL_final_02022020.csv gs://bucketpath/