ClaimItems_items_VF

October 2, 2020

[1]:
```
# import necessaries modules:
import numpy as np
import pandas as pd
import datetime
import gc
```

0.1 Step 1. Reading the data related to tblClaimItems

0.1.1 Step 1.1: Reading tblClaimItems

[2]:
```
# csv file related to the tblClaimItems:
filename = 'openIMIS csv/claim_items2020.csv'

# selection of columns (the entire table has 30 columns)
cols = ['ClaimItemID', 'ClaimID', 'ItemID','ProdID','PolicyID',\
        'ClaimItemStatus','RejectionReason',\
       ␣
 ↪'QtyProvided','QtyApproved','PriceAsked','PriceApproved','PriceValuated',\
       'Explanation','Justification',
       'ValidityFromReview','AuditUserIDReview']

# read the csv file
df_claim_items_raw = pd.read_csv(filename,low_memory=False,usecols=cols,\
                                 parse_dates = ['ValidityFromReview'])
df_claim_items_raw = df_claim_items_raw.iloc[:-2,:]

# add a column 'ItemServiceType' with all values equal to 'Medication':
df_claim_items_raw['ItemServiceType']='Medication'

memStats_claim_items = (df_claim_items_raw.memory_usage()/1024/1024).sum()
shape_claim_items = df_claim_items_raw.shape
```

0.1.2 Step 1.2: Reading tblItems

[5]:
```
# csv file related to the tblItems:
filename = 'openIMIS csv/items2020.csv'

# selection of columns
cols = ['ItemID', 'ItemCode', 'ItemName', 'ItemType', \
        'ItemPrice', 'ItemCareType', 'ItemFrequency', 'ItemPatCat',\
        'ItemUUID','ValidityFrom','ValidityTo']

# read the csv file
df_items_raw = pd.read_csv(filename,low_memory=False,usecols=cols,\
                           parse_dates = ['ValidityFrom','ValidityTo'])
df_items_raw = df_items_raw.iloc[:-2,:]

# rename the columns in order to have similar name as the claimservices related␣
 ↪dataset
df_items_raw.rename(columns = {'ValidityFrom': 'ItemValidityFrom',
                               'ValidityTo': 'ItemValidityTo'}, inplace =␣
 ↪True)

df_items_raw['ItemID'] = df_items_raw['ItemID'].astype(float)

# add a column 'ItemLevel' in order o have coherence with the concatenation of␣
 ↪the
# tblClaimServices and tblServices:
df_items_raw['ItemLevel'] = 'M'

memStats_items = (df_items_raw.memory_usage()/1024/1024).sum()
shape_items = df_items_raw.shape
```

Step 1.3: Concatention of the tblClaimItems with tblItems (based on ItemID column)

[6]:
```
df_concat = pd.merge(df_claim_items_raw,df_items_raw,on='ItemID')

memStats_claim_items_c = (df_concat.memory_usage()/1024/1024).sum()
shape_claim_items_c = df_concat.shape
```

[7]:
```
# deleting dataframes no longer necessary
del [[df_claim_items_raw,df_items_raw]]
df_claim_items_raw=pd.DataFrame()
df_items_raw=pd.DataFrame()
gc.collect()
```

[7]: 28

0.1.3 Step 2: Reading the claim related file

This is an already concatenated file with all the fields related to other tables (tblClaimAdmins, tblHF, tblLocations, tblInsuree, tblFamilies, tblDiagnosis)

[8]:
```
# open the concatenated file related to claims
df_claims_raw=pd.read_pickle('openIMIS csv/ClaimsPlus2020_sel.pkl')

memStats_claims = (df_claims_raw.memory_usage()/1024/1024).sum()
shape_claims = df_claims_raw.shape
```

0.1.4 Step 3: Merge all dataframes

[9]:
```
df_citems_sel = pd.merge(df_concat,df_claims_raw,on='ClaimID')
```

[11]:
```
# Verify if item valid
cond1 = (df_citems_sel['DateFrom']>df_citems_sel['ItemValidityFrom'])&\
(df_citems_sel['ItemValidityTo'].isnull())
cond2 = (df_citems_sel['DateFrom']>df_citems_sel['ItemValidityFrom'])&\
(df_citems_sel['ItemValidityTo'].notnull())&\
(df_citems_sel['DateTo']>df_citems_sel['ItemValidityTo'])

# create a list of Valid/Not Valid items
validcond = cond1|cond2

# create a new column
df_citems_sel.loc[validcond,('ValidItem')] = 1
df_citems_sel.loc[~validcond,('ValidItem')] = 0
```

[12]:
```
# Verify if ClaimAdmin valid
cond1 = (df_citems_sel['DateFrom']>df_citems_sel['ClaimAdminValidityFrom'])&\
(df_citems_sel['ClaimAdminValidityTo'].isnull())
cond2 = (df_citems_sel['DateFrom']>df_citems_sel['ClaimAdminValidityFrom'])&\
(df_citems_sel['ClaimAdminValidityTo'].notnull())&\
(df_citems_sel['DateTo']>df_citems_sel['ClaimAdminValidityTo'])

# create a list of Valid/Not Valid items
validcond = cond1|cond2

# create a new column
df_citems_sel.loc[validcond,('ValidClaimAdmin')] = 1
df_citems_sel.loc[~validcond,('ValidClaimAdmin')] = 0
```

[13]:
```
# Verify if Insuree valid
cond1 = (df_citems_sel['DateFrom']>df_citems_sel['InsureeValidityFrom'])&\
(df_citems_sel['InsureeValidityTo'].isnull())
cond2 = (df_citems_sel['DateFrom']>df_citems_sel['ClaimAdminValidityFrom'])&\
(df_citems_sel['InsureeValidityTo'].notnull())&\
(df_citems_sel['DateTo']>df_citems_sel['InsureeValidityTo'])
```

```
# create a list of Valid/Not Valid items
validcond = cond1|cond2

# create a new column
df_citems_sel.loc[validcond,('ValidInsuree')] = 1
df_citems_sel.loc[~validcond,('ValidInsuree')] = 0
```

[20]:
```
# Verify if Family valid
cond1 = (df_citems_sel['DateFrom']>df_citems_sel['FamilyValidityFrom'])&\
(df_citems_sel['FamilyValidityTo'].isnull())
cond2 = (df_citems_sel['DateFrom']>df_citems_sel['FamilyValidityFrom'])&\
(df_citems_sel['FamilyValidityTo'].notnull())&\
(df_citems_sel['DateTo']>df_citems_sel['FamilyValidityTo'])

# create a list of Valid/Not Valid items
validitem = cond1|cond2

# create a new column
df_citems_sel.loc[validitem,('ValidFamily')] = 1
df_citems_sel.loc[~validitem,('ValidFamily')] = 0
```

[21]:
```
# Verify if Location valid
cond1 = (df_citems_sel['DateFrom']>df_citems_sel['LocationValidityFrom'])&\
(df_citems_sel['LocationValidityTo'].isnull())
cond2 = (df_citems_sel['DateFrom']>df_citems_sel['LocationValidityFrom'])&\
(df_citems_sel['LocationValidityTo'].notnull())&\
(df_citems_sel['DateTo']>df_citems_sel['LocationValidityTo'])

# create a list of Valid/Not Valid items
validitem = cond1|cond2

# create a new column
df_citems_sel.loc[validitem,('ValidLocation')] = 1
df_citems_sel.loc[~validitem,('ValidLocation')] = 0
```

[22]:
```
# Verify if ICD valid
cond1 = (df_citems_sel['DateFrom']>df_citems_sel['ICDValidityFrom'])&\
(df_citems_sel['ICDValidityTo'].isnull())
cond2 = (df_citems_sel['DateFrom']>df_citems_sel['ICDValidityFrom'])&\
(df_citems_sel['ICDValidityTo'].notnull())&\
(df_citems_sel['DateTo']>df_citems_sel['ICDValidityTo'])

# create a list of Valid/Not Valid items
validitem = cond1|cond2

# create a new column
df_citems_sel.loc[validitem,('ValidICD')] = 1
```

```
df_citems_sel.loc[~validitem,('ValidICD')] = 0
```

[23]:
```
df_citems_sel.drop(['ItemValidityFrom','ItemValidityTo',\
                    'ClaimAdminValidityFrom','ClaimAdminValidityTo',\
                   'InsureeValidityFrom','InsureeValidityTo',\
                   'FamilyValidityFrom','FamilyValidityTo',\
                   'LocationValidityFrom','LocationValidityTo',\
                   'ICDValidityFrom','ICDValidityTo'], axis=1, inplace=True)
```

[27]:
```
memStats = (df_citems_sel.memory_usage()/1024/1024).sum()
shape_ciems_sel = df_citems_sel.shape
```

[24]:
```
# save the results:
df_citems_sel.to_pickle('openIMIS csv/ClaimItems_Items2020_sel.pkl')
#df_concat.to_csv('openIMIS csv/ClaimItems_Items2000_sel.csv')
```

0.2 Summary

[28]:
```
# Summary concerning the concatenation
print(f'''Summary of the concatenation process:
- tblClaimItems has : {shape_claim_items[0]} rows ; {shape_claim_items[1]}␣
 ↪columns ; {round(memStats_claim_items,2)} memory consumption;
- tblItems has : {shape_items[0]} rows ; {shape_items[1]} columns ;␣
 ↪{round(memStats_items,2)} memory consumption;
- Concatenation of tblClaimItems and tblItems has : {shape_claim_items_c[0]}␣
 ↪rows ;
{shape_claim_items_c[1]} columns ; {round(memStats_claim_items_c,2)} memory␣
 ↪consumption;
- tblClaims has : {shape_claims[0]} rows ; {shape_claims[1]} columns ;␣
 ↪{round(memStats_claims,2)} memory consumption;
- Concatenation of previous tables has : {shape_ciems_sel[0]} rows ;
{shape_ciems_sel[1]} columns ;{round(memStats,2)} memory consumption;
''')
```

```
Summary of the concatenation process:
- tblClaimItems has : 12414014 rows ; 17 columns ; 1610.09 memory consumption;
- tblItems has : 1528 rows ; 12 columns ; 0.14 memory consumption;
- Concatenation of tblClaimItems and tblItems has : 12414014 rows ;
28 columns ; 2746.63 memory consumption;
- tblClaims has : 5953640 rows ; 61 columns ; 2816.21 memory consumption;
- Concatenation of previous tables has : 12371992 rows ;
82 columns ;7834.44 memory consumption;
```