1_Create_Training_Dataset

May 3, 2021

[1]:
```
import modin.pandas as pd
```

[2]:
```
col_names = pd.read_csv('./DATA/columns.csv', sep='\t')
col_names.head()
```

```
UserWarning: Ray execution environment not yet initialized. Initializing…
To remove this warning, run the following python code before doing dataframe
operations:

    import ray
    ray.init()
```

[2]:

	Orig	\
0	Credit Score	
1	First Payment Date	
2	First Time Homebuyer Flag	
3	Maturity Date	
4	Metropolitan Statistical Area (MSA) Or Metropo…	

	Svcg
0	Loan Sequence Number
1	Monthly Reporting Period
2	Current Actual UPB
3	Current Loan Delinquency Status
4	Loan Age

[3]:
```
orig_prefix = './DATA/sample_orig_'
svcg_prefix = './DATA/sample_svcg_'
years = [y for y in range(2010,2021)]
```

[4]:
```
deldf_subs = []
for year in years:
    print("Retrieving monthly data for loans of year " + str(year))
    month_df = pd.read_csv(svcg_prefix+str(year)+'.
 ↪txt',sep='|',names=list(col_names['Svcg']))

    tf1 = month_df['Loan Age'] < 12
```

```
    tf2 = month_df['Delinquency Due to Disaster'] != 'Y'
    tf3 = ~month_df['Current Loan Delinquency Status'].isin(['R','XX','  '])
    tf = tf1 & tf2 & tf3
    month_df = month_df[tf].copy()

    month_df['Current Loan Delinquency Status'] = month_df['Current Loan␣
↪Delinquency Status'].astype(int)
    agg_dict = {'Current Loan Delinquency Status':sum,
            'Monthly Reporting Period':'count'}
    deldf_sub = month_df.groupby('Loan Sequence Number').agg(agg_dict)
    col_dct = {'Monthly Reporting Period':'NumMonths'}
    deldf_sub.reset_index(inplace=True)
    deldf_sub.rename(columns=col_dct, inplace=True)

    deldf_sub = deldf_sub[deldf_sub['NumMonths']==12].copy()
    if len(deldf_sub) == 0:
        continue

    deldf_sub['Loan Deliquency Within Year'] = True

    tf = deldf_sub['Current Loan Delinquency Status'] == 0
    deldf_sub.loc[tf, 'Loan Deliquency Within Year'] = False
    deldf_subs.append(deldf_sub)
```

```
Retrieving monthly data for loans of year 2010
Retrieving monthly data for loans of year 2011
Retrieving monthly data for loans of year 2012
Retrieving monthly data for loans of year 2013
Retrieving monthly data for loans of year 2014
Retrieving monthly data for loans of year 2015
Retrieving monthly data for loans of year 2016
Retrieving monthly data for loans of year 2017
Retrieving monthly data for loans of year 2018
Retrieving monthly data for loans of year 2019
Retrieving monthly data for loans of year 2020
```

```
UserWarning: `DataFrame.copy` for empty DataFrame defaulting to pandas
implementation.
To request implementation, send an email to feature_requests@modin.org.
UserWarning: Distributing <class 'pandas.core.frame.DataFrame'> object. This may
take some time.
```

[5]:
```
delinq_df = pd.concat(deldf_subs)[['Loan Sequence Number','Loan Deliquency␣
↪Within Year']]
delinq_df.head()
```

[5]:

	Loan Sequence Number	Loan Deliquency Within Year
0	F110Q1000008	False
2	F110Q1000064	False
3	F110Q1000072	False
4	F110Q1000080	False
5	F110Q1000096	False

[6]:
```
orig_subs = []
for year in years:
    print("Retrieving origination data for loans of year " + str(year))
    orig_sub = pd.read_csv(orig_prefix+str(year)+'.
 ↪txt',sep='|',names=list(col_names['Orig']))
    orig_subs.append(orig_sub)
```

```
Retrieving origination data for loans of year 2010
Retrieving origination data for loans of year 2011
Retrieving origination data for loans of year 2012
Retrieving origination data for loans of year 2013
Retrieving origination data for loans of year 2014
Retrieving origination data for loans of year 2015
Retrieving origination data for loans of year 2016
Retrieving origination data for loans of year 2017
Retrieving origination data for loans of year 2018
Retrieving origination data for loans of year 2019
Retrieving origination data for loans of year 2020
```

[7]:
```
orig_df = pd.concat(orig_subs)
orig_df.head()
```

[7]:

	Credit Score	First Payment Date	First Time Homebuyer Flag	Maturity Date	\
0	804	201004	N	204003	
1	786	201004	9	204003	
2	816	201003	N	204002	
3	783	201003	N	204002	
4	667	201003	N	204002	

	Metropolitan Statistical Area (MSA) Or Metropolitan Division	\
0	17860.0	
1	NaN	
2	44220.0	
3	45820.0	
4	17900.0	

	Mortgage Insurance Percentage (MI %)	Number of Units	Occupancy Status	\
0	0	1	P	
1	0	1	P	
2	0	1	P	

```
3                                    0                1                P
4                                    0                1                P

   Original Combined Loan-to-Value (CLTV)  \
0                                      39
1                                      50
2                                      56
3                                      80
4                                      80

   Original Debt-to-Income (DTI) Ratio  …  Original Loan Term  \
0                                   20  …                 360
1                                   43  …                 360
2                                   27  …                 360
3                                   36  …                 360
4                                   20  …                 360

  Number of Borrowers    Seller Name    Servicer Name Super Conforming Flag  \
0                   2  Other sellers   U.S. BANK N.A.                   NaN
1                   1  Other sellers  Other servicers                   NaN
2                   1  Other sellers  Other servicers                   NaN
3                   2  Other sellers   U.S. BANK N.A.                   NaN
4                   1  Other sellers  Other servicers                   NaN

  Pre-HARP Loan Sequence Number Program Indicator  HARP Indicator  \
0                           NaN                 9             NaN
1                           NaN                 9             NaN
2                           NaN                 9             NaN
3                           NaN                 9             NaN
4                           NaN                 9             NaN

   Property Valuation Method  Interest Only (I/O) Indicator
0                          9                            NaN
1                          9                            NaN
2                          9                            NaN
3                          9                            NaN
4                          9                            NaN

[5 rows x 31 columns]
```

[8]:
```
delinq_df['Loan Sequence Number'].nunique()
```

[8]: 396338

[9]:
```
orig_df['Loan Sequence Number'].nunique()
```

[9]: 525000

[10]:
```
training_dat = delinq_df.merge(orig_df, on='Loan Sequence Number', how='left')
training_dat.head()
```

[10]:

	Loan Sequence Number	Loan Deliquency Within Year	Credit Score	\
0	F110Q1000008	False	804	
1	F110Q1000064	False	816	
2	F110Q1000072	False	783	
3	F110Q1000080	False	667	
4	F110Q1000096	False	799	

	First Payment Date	First Time Homebuyer Flag	Maturity Date	\
0	201004	N	204003	
1	201003	N	204002	
2	201003	N	204002	
3	201003	N	204002	
4	201003	N	204002	

	Metropolitan Statistical Area (MSA) Or Metropolitan Division	\
0	17860.0	
1	44220.0	
2	45820.0	
3	17900.0	
4	NaN	

	Mortgage Insurance Percentage (MI %)	Number of Units	Occupancy Status	\
0	0	1	P	
1	0	1	P	
2	0	1	P	
3	0	1	P	
4	0	1	P	

	…	Original Loan Term	Number of Borrowers	Seller Name	\
0	…	360	2	Other sellers	
1	…	360	1	Other sellers	
2	…	360	2	Other sellers	
3	…	360	1	Other sellers	
4	…	360	1	Other sellers	

	Servicer Name	Super Conforming Flag	Pre-HARP Loan Sequence Number	\
0	U.S. BANK N.A.	NaN	NaN	
1	Other servicers	NaN	NaN	
2	U.S. BANK N.A.	NaN	NaN	
3	Other servicers	NaN	NaN	
4	U.S. BANK N.A.	NaN	NaN	

	Program Indicator	HARP Indicator	Property Valuation Method	\
0	9	NaN	9	

```
1                  9            NaN                          9
2                  9            NaN                          9
3                  9            NaN                          9
4                  9            NaN                          9

  Interest Only (I/O) Indicator
0                           NaN
1                           NaN
2                           NaN
3                           NaN
4                           NaN

[5 rows x 32 columns]
```

[11]:
```
len(training_dat)
```

[11]:
```
396338
```

[12]:
```
training_dat['Loan Deliquency Within Year'].value_counts()
```

```
UserWarning: value_counts defaulting to pandas implementation.
```

[12]:
```
False    387901
True       8437
Name: Loan Deliquency Within Year, dtype: int64
```

[13]:
```
training_dat['Program Indicator'] = training_dat['Program Indicator'].
↪astype(str)
```

[15]:
```
training_dat.to_pickle('./PROCESSED/training_dat.pkl')
```

```
UserWarning: `DataFrame.to_pickle` defaulting to pandas implementation.
UserWarning: Distributing <class 'pandas.core.frame.DataFrame'> object. This may
take some time.
```

[ ]:
```
# Drop month diff > 12
```

[ ]:
```

```