import pandas as pd

from datetime import datetime

import warnings

warnings.filterwarnings('ignore')

'''

Gera as series de Ideb por ano para serem usadas na analise cluster que vou fazer

'''

class Preprocessing(object):

def \_\_init\_\_(self, file\_name, encoding='utf-8'):

self.file = file\_name

self.encoding = encoding

self.dados = None

self.dados\_temporais = None

@property

def campos\_validos(self):

return ['Co\_UF', 'Cod\_Municipio\_Completo', 'Cod\_Escola\_Completo', 'Rede',

'Ideb2005', 'Ideb2007', 'Ideb2009','Ideb2011', 'Ideb2013', 'Ideb2015', 'Ideb2017',

'TaxaAprovacao2005\_1ao5ano','TaxaAprovacao2007\_1ao5ano','TaxaAprovacao2009\_1ao5ano','TaxaAprovacao2011\_1ao5ano','TaxaAprovacao2013\_1ao5ano','TaxaAprovacao2015\_1ao5ano','TaxaAprovacao2017\_1ao5ano',

'TaxaAprovacao2005\_1ano','TaxaAprovacao2007\_1ano','TaxaAprovacao2009\_1ano','TaxaAprovacao2011\_1ano','TaxaAprovacao2013\_1ano','TaxaAprovacao2015\_1ano','TaxaAprovacao2017\_1ano',

'TaxaAprovacao2005\_2ano','TaxaAprovacao2007\_2ano','TaxaAprovacao2009\_2ano','TaxaAprovacao2011\_2ano','TaxaAprovacao2013\_2ano','TaxaAprovacao2015\_2ano','TaxaAprovacao2017\_2ano',

'TaxaAprovacao2005\_3ano','TaxaAprovacao2007\_3ano','TaxaAprovacao2009\_3ano','TaxaAprovacao2011\_3ano','TaxaAprovacao2013\_3ano','TaxaAprovacao2015\_3ano','TaxaAprovacao2017\_3ano',

'TaxaAprovacao2005\_4ano','TaxaAprovacao2007\_4ano','TaxaAprovacao2009\_4ano','TaxaAprovacao2011\_4ano','TaxaAprovacao2013\_4ano','TaxaAprovacao2015\_4ano','TaxaAprovacao2017\_4ano',

'TaxaAprovacao2005\_5ano','TaxaAprovacao2007\_5ano','TaxaAprovacao2009\_5ano','TaxaAprovacao2011\_5ano','TaxaAprovacao2013\_5ano','TaxaAprovacao2015\_5ano','TaxaAprovacao2017\_5ano',

'IndicadorRendimento\_2005','IndicadorRendimento\_2007','IndicadorRendimento\_2009','IndicadorRendimento\_2011','IndicadorRendimento\_2013','IndicadorRendimento\_2015','IndicadorRendimento\_2017',

'NotaProvaBrasil\_MT\_2005','NotaProvaBrasil\_MT\_2007','NotaProvaBrasil\_MT\_2009','NotaProvaBrasil\_MT\_2011','NotaProvaBrasil\_MT\_2013','NotaProvaBrasil\_MT\_2015','NotaProvaBrasil\_MT\_2017',

'NotaProvaBrasil\_LP\_2005','NotaProvaBrasil\_LP\_2007','NotaProvaBrasil\_LP\_2009','NotaProvaBrasil\_LP\_2011','NotaProvaBrasil\_LP\_2013','NotaProvaBrasil\_LP\_2015','NotaProvaBrasil\_LP\_2017',

'NotaProvaBrasil\_NotaMedia\_2005','NotaProvaBrasil\_NotaMedia\_2007','NotaProvaBrasil\_NotaMedia\_2009','NotaProvaBrasil\_NotaMedia\_2011','NotaProvaBrasil\_NotaMedia\_2013','NotaProvaBrasil\_NotaMedia\_2015','NotaProvaBrasil\_NotaMedia\_2017']

def load\_file(self):

df = pd.read\_csv(self.file, encoding=self.encoding)

df = df[self.campos\_validos].copy()

df.columns = ['uf', 'cod\_mun', 'cod\_escola', 'tp\_rede',

'i05', 'i07', 'i09', 'i11', 'i13', 'i15', 'i17',

'tx\_aprov\_1ao5ano\_2005','tx\_aprov\_1ao5ano\_2007','tx\_aprov\_1ao5ano\_2009','tx\_aprov\_1ao5ano\_2011','tx\_aprov\_1ao5ano\_2013','tx\_aprov\_1ao5ano\_2015','tx\_aprov\_1ao5ano\_2017',

'tx\_aprov\_1ano\_2005','tx\_aprov\_1ano\_2007','tx\_aprov\_1ano\_2009','tx\_aprov\_1ano\_2011','tx\_aprov\_1ano\_2013','tx\_aprov\_1ano\_2015','tx\_aprov\_1ano\_2017',

'tx\_aprov\_2ano\_2005','tx\_aprov\_2ano\_2007','tx\_aprov\_2ano\_2009','tx\_aprov\_2ano\_2011','tx\_aprov\_2ano\_2013','tx\_aprov\_2ano\_2015','tx\_aprov\_2ano\_2017',

'tx\_aprov\_3ano\_2005','tx\_aprov\_3ano\_2007','tx\_aprov\_3ano\_2009','tx\_aprov\_3ano\_2011','tx\_aprov\_3ano\_2013','tx\_aprov\_3ano\_2015','tx\_aprov\_3ano\_2017',

'tx\_aprov\_4ano\_2005','tx\_aprov\_4ano\_2007','tx\_aprov\_4ano\_2009','tx\_aprov\_4ano\_2011','tx\_aprov\_4ano\_2013','tx\_aprov\_4ano\_2015','tx\_aprov\_4ano\_2017',

'tx\_aprov\_5ano\_2005','tx\_aprov\_5ano\_2007','tx\_aprov\_5ano\_2009','tx\_aprov\_5ano\_2011','tx\_aprov\_5ano\_2013','tx\_aprov\_5ano\_2015','tx\_aprov\_5ano\_2017',

'ind\_rend\_2005','ind\_rend\_2007','ind\_rend\_2009','ind\_rend\_2011','ind\_rend\_2013','ind\_rend\_2015','ind\_rend\_2017',

'nota\_prova\_MT\_2005','nota\_prova\_MT\_2007','nota\_prova\_MT\_2009','nota\_prova\_MT\_2011','nota\_prova\_MT\_2013','nota\_prova\_MT\_2015','nota\_prova\_MT\_2017',

'nota\_prova\_LP\_2005','nota\_prova\_LP\_2007','nota\_prova\_LP\_2009','nota\_prova\_LP\_2011','nota\_prova\_LP\_2013','nota\_prova\_LP\_2015','nota\_prova\_LP\_2017',

'nota\_prova\_nota\_media\_2005','nota\_prova\_nota\_media\_2007','nota\_prova\_nota\_media\_2009','nota\_prova\_nota\_media\_2011','nota\_prova\_nota\_media\_2013','nota\_prova\_nota\_media\_2015','nota\_prova\_nota\_media\_2017']

df['cod\_escola'] = df['cod\_escola'].astype(int)

df['cod\_mun'] = df['cod\_mun'].astype(int)

if self.dados is None:

self.dados = df

def fill\_missing(self):

lista\_col = ['i05', 'i07', 'i09', 'i11', 'i13', 'i15', 'i17',

'tx\_aprov\_1ao5ano\_2005','tx\_aprov\_1ao5ano\_2007','tx\_aprov\_1ao5ano\_2009','tx\_aprov\_1ao5ano\_2011','tx\_aprov\_1ao5ano\_2013','tx\_aprov\_1ao5ano\_2015','tx\_aprov\_1ao5ano\_2017',

'tx\_aprov\_1ano\_2005','tx\_aprov\_1ano\_2007','tx\_aprov\_1ano\_2009','tx\_aprov\_1ano\_2011','tx\_aprov\_1ano\_2013','tx\_aprov\_1ano\_2015','tx\_aprov\_1ano\_2017',

'tx\_aprov\_2ano\_2005','tx\_aprov\_2ano\_2007','tx\_aprov\_2ano\_2009','tx\_aprov\_2ano\_2011','tx\_aprov\_2ano\_2013','tx\_aprov\_2ano\_2015','tx\_aprov\_2ano\_2017',

'tx\_aprov\_3ano\_2005','tx\_aprov\_3ano\_2007','tx\_aprov\_3ano\_2009','tx\_aprov\_3ano\_2011','tx\_aprov\_3ano\_2013','tx\_aprov\_3ano\_2015','tx\_aprov\_3ano\_2017',

'tx\_aprov\_4ano\_2005','tx\_aprov\_4ano\_2007','tx\_aprov\_4ano\_2009','tx\_aprov\_4ano\_2011','tx\_aprov\_4ano\_2013','tx\_aprov\_4ano\_2015','tx\_aprov\_4ano\_2017',

'tx\_aprov\_5ano\_2005','tx\_aprov\_5ano\_2007','tx\_aprov\_5ano\_2009','tx\_aprov\_5ano\_2011','tx\_aprov\_5ano\_2013','tx\_aprov\_5ano\_2015','tx\_aprov\_5ano\_2017',

'ind\_rend\_2005','ind\_rend\_2007','ind\_rend\_2009','ind\_rend\_2011','ind\_rend\_2013','ind\_rend\_2015','ind\_rend\_2017',

'nota\_prova\_MT\_2005','nota\_prova\_MT\_2007','nota\_prova\_MT\_2009','nota\_prova\_MT\_2011','nota\_prova\_MT\_2013','nota\_prova\_MT\_2015','nota\_prova\_MT\_2017',

'nota\_prova\_LP\_2005','nota\_prova\_LP\_2007','nota\_prova\_LP\_2009','nota\_prova\_LP\_2011','nota\_prova\_LP\_2013','nota\_prova\_LP\_2015','nota\_prova\_LP\_2017',

'nota\_prova\_nota\_media\_2005','nota\_prova\_nota\_media\_2007','nota\_prova\_nota\_media\_2009','nota\_prova\_nota\_media\_2011','nota\_prova\_nota\_media\_2013','nota\_prova\_nota\_media\_2015','nota\_prova\_nota\_media\_2017']

for el in lista\_col:

self.dados[el] = self.dados.apply(lambda x : '0' if x[el]=='-' else x[el], axis=1)

def formata\_serie(self):

frames = []

for escola in self.dados['cod\_escola'].tolist():

dfesc = self.dados[self.dados['cod\_escola'] == escola]

\_df = dfesc.iloc[:, 4:].T

\_df.columns = ['ideb']

lista\_ideb = \_df['ideb'].tolist()

frames.append(pd.DataFrame({'cod\_escola' : [dfesc['cod\_escola'].iloc[0]]\*7,

'cod\_mun' : [dfesc['cod\_mun'].iloc[0]]\*7,

'uf' : [dfesc['uf'].iloc[0]]\*7,

'tp\_rede' : [dfesc['tp\_rede'].iloc[0]]\*7,

'ano' : [2005, 2007, 2009, 2011, 2013, 2015, 2017],

'ideb' : lista\_ideb}))

resp = pd.concat(frames)

resp['ideb'] = resp.apply(lambda x : float(x['ideb']), axis=1)

if self.dados\_temporais is None:

self.dados\_temporais = resp

def dados\_finais(self):

self.load\_file()

self.fill\_missing()

self.formata\_serie()

return self.dados\_temporais

if \_\_name\_\_ == '\_\_main\_\_':

pp = Preprocessing('ideb\_escolas\_anosiniciais2005\_2017.csv', 'latin-1')

resp2 = pp.dados\_finais()