m07_03_limpeza_base_dados

May 3, 2021

1 Curso de Python do DS ao DEV

Comunidade DS - Meigarom Lopes

2 Modulo 07 - Banco de Dados

3 Limpeza e definicação da granularidade

[1]:
```
import re
import numpy  as np
import pandas as pd
```

```
/Users/meigarom.lopes/.pyenv/versions/3.8.0/envs/pythondsaodev/lib/python3.8/sit
e-packages/pandas/compat/__init__.py:97: UserWarning: Could not import the lzma
module. Your installed Python is incomplete. Attempting to use lzma compression
will result in a RuntimeError.
  warnings.warn(msg)
```

[87]:
```
data = pd.read_csv( '/Users/meigarom.lopes/repos/python-ds-ao-dev/Module01/
 ↪products_hm.csv' )

# product id
data = data.dropna( subset=['product_id'] )
data['product_id'] = data['product_id'].astype( int )

# product name
data['product_name'] = data['product_name'].apply( lambda x: x.replace( ' ',␣
 ↪'_' ).lower() )

# product price
data['product_price'] = data['product_price'].apply( lambda x: x.replace( '$ ',␣
 ↪'' ) ).astype( float )

# scrapy datetime
data['scrapy_datetime'] = pd.to_datetime( data['scrapy_datetime'],␣
 ↪format='%Y-%m-%d %H:%M:%S' )
```

```
# style id
data['style_id'] = data['style_id'].astype( int )

# color id
data['color_id'] = data['color_id'].astype( int )

# color name
data['color_name'] = data['color_name'].apply( lambda x: x.replace( ' ', '_' ).
↪replace( '/', '_' ).lower() if pd.notnull( x ) else x )

# fit
data['fit'] = data['fit'].apply( lambda x: x.replace( ' ', '_' ).lower() if pd.
↪notnull( x ) else x )

# size number
data['size_number'] = data['size'].apply( lambda x: re.search( '\d{3}cm', x ).
↪group(0) if pd.notnull( x ) else x )
data['size_number'] = data['size_number'].apply( lambda x: re.search( '\d+', x␣
↪).group(0) if pd.notnull( x ) else x )

# size model
data['size_model'] = data['size'].str.extract( '(\d+/\\d+)' )

# composition
data = data[~data['composition'].str.contains( 'Pocket lining:', na=False )]
data = data[~data['composition'].str.contains( 'Lining:', na=False )]
data = data[~data['composition'].str.contains( 'Shell:', na=False )]

# drop duplicates
data = data.drop_duplicates( subset=['product_id', 'product_category',␣
↪'product_name', 'product_price',
                                     'scrapy_datetime', 'style_id', 'color_id',␣
↪'color_name', 'fit'], keep='last' )

# reset index
data = data.reset_index( drop=True )

# break composition by comma
df1 = data['composition'].str.split( ',', expand=True )

# cotton | polyester | elastano | elasterell
df_ref = pd.DataFrame( index=np.arange( len( data ) ),␣
↪columns=['cotton','polyester', 'elastane', 'elasterell'] )

# cotton
df_cotton = df1[0]
```

```
df_cotton.name = 'cotton'

df_ref = pd.concat( [df_ref, df_cotton ], axis=1 )
df_ref = df_ref.iloc[:, ~df_ref.columns.duplicated( keep='last')]
df_ref['cotton'] = df_ref['cotton'].fillna( 'Cotton 0%' )

# polyester
df_polyester = df1.loc[df1[1].str.contains( 'Polyester', na=True ), 1]
df_polyester.name = 'polyester'

df_ref = pd.concat( [df_ref, df_polyester], axis=1 )
df_ref = df_ref.iloc[:, ~df_ref.columns.duplicated( keep='last') ]
df_ref['polyester'] = df_ref['polyester'].fillna( 'Polyester 0%' )

# elastano
df_elastane = df1.loc[df1[1].str.contains( 'Elastane', na=True ), 1]
df_elastane.name = 'elastane'

# combine elastane from both columns 1 and 2
df_elastane = df_elastane.combine_first( df1[2] )

df_ref = pd.concat( [df_ref, df_elastane], axis=1 )
df_ref = df_ref.iloc[:, ~df_ref.columns.duplicated( keep='last') ]
df_ref['elastane'] = df_ref['elastane'].fillna( 'Elastane 0%' )

# elasterell
df_elasterell = df1.loc[df1[1].str.contains( 'Elasterell', na=True ), 1]
df_elasterell.name = 'elasterell'
df_ref = pd.concat( [df_ref, df_elasterell], axis=1 )
df_ref = df_ref.iloc[:, ~df_ref.columns.duplicated( keep='last') ]

df_ref['elasterell'] = df_ref['elasterell'].fillna( 'Elasterell-P 0%' )

# final join
data = pd.concat( [data, df_ref], axis=1 )

# format composition data
data['cotton'] = data['cotton'].apply( lambda x: int( re.search( '\d+', x ).
 ↪group(0) ) / 100 if pd.notnull( x ) else x )
data['polyester'] = data['polyester'].apply( lambda x: int( re.search( '\d+', x␣
 ↪).group(0) ) / 100 if pd.notnull( x ) else x )
data['elastane'] = data['elastane'].apply( lambda x: int( re.search( '\d+', x ).
 ↪group(0) ) / 100 if pd.notnull( x ) else x )
data['elasterell'] = data['elasterell'].apply( lambda x: int( re.search(␣
 ↪'\d+',x ).group(0) ) / 100 if pd.notnull( x ) else x )

# Drop columns
```

```
data = data.drop( columns=['size', 'product safety', 'composition'], axis=1 )

# Drop duplicates
data = data.drop_duplicates()
data.shape
```

```
[87]: (308, 15)
```

```
[90]: data.head()
```

```
[90]:    product_id product_category product_name  product_price  \
      0   636207006   men_jeans_slim   slim_jeans          19.99
      1   636207006   men_jeans_slim   slim_jeans          19.99
      2   636207006   men_jeans_slim   slim_jeans          19.99
      3   636207006   men_jeans_slim   slim_jeans          19.99
      4   636207006   men_jeans_slim   slim_jeans          19.99

            scrapy_datetime  style_id  color_id       color_name       fit  \
      0 2021-04-11 17:48:05    636207         6            white  slim_fit
      1 2021-04-11 17:48:05    636207         6  dark_denim_blue  slim_fit
      2 2021-04-11 17:48:05    636207         6  dark_gray_denim  slim_fit
      3 2021-04-11 17:48:05    636207         6             gray  slim_fit
      4 2021-04-11 17:48:05    636207         6            black  slim_fit

        size_number size_model  cotton  polyester  elastane  elasterell
      0         187      31/32    0.99        0.0      0.01         0.0
      1         187      31/30    0.88        0.1      0.02         0.0
      2         187      31/30    0.88        0.1      0.02         0.0
      3         187      31/30    0.88        0.1      0.02         0.0
      4         187      31/30    0.88        0.1      0.02         0.0
```