5_Life_Expectancy_Feature_Engineering

January 29, 2023

Table of Contents

1 Life_Expectancy_WHO_UN_Analysis_Modeling

1.1 Feature_Engineering

To: Magnimind

From: Matt Curcio, matt.curcio.ri@gmail.com

Date: 2023-01-29

Re: NOTEBOOK #5

Categorize Countires into Regions

This file takes input of `Clean_LE_Data_Post_EDA_3.csv` and produces output of `Clean_LE_Data_FEng_4.csv`

- This list of countries and their regions was found on the site:

  https://www.thoughtco.com/official-listing-of-countries-world-region-1435153

1.2 Load and verify data integrity

[1]:

```
# Common Python Libraries
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
%matplotlib inline
import seaborn as sns
```

[2]:

```
!ls *.csv
```

```
Clean_LE_Data_FEng_4.csv      Life_Expectancy_Data.csv  y_test.csv
Clean_LE_Data_Post_EDA_3.csv  x_test.csv                y_train.csv
Clean_LE_Data_w_Means_2.csv   x_train.csv
```

[3]:

```
filename = 'Clean_LE_Data_Post_EDA_3.csv'

df = pd.read_csv(filename, header=0)

# Convert object 'Status' to categorical
df["Status"] = pd.Categorical(df["Status"])

df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 2928 entries, 0 to 2927
Data columns (total 17 columns):
 #   Column          Non-Null Count  Dtype
---  ------          --------------  -----
 0   Country         2928 non-null   object
 1   Year            2928 non-null   int64
 2   Status          2928 non-null   category
 3   LifeExpectancy  2928 non-null   float64
 4   AdultMort       2928 non-null   float64
 5   EtOH            2928 non-null   float64
 6   PercExpen       2928 non-null   float64
 7   Measles         2928 non-null   int64
 8   BMI             2928 non-null   float64
 9   lt5yD           2928 non-null   int64
 10  Polio           2928 non-null   float64
 11  TotalExpen      2928 non-null   float64
 12  DTP             2928 non-null   float64
 13  HIV             2928 non-null   float64
 14  Thin1_19y       2928 non-null   float64
 15  Income          2928 non-null   float64
 16  Education       2928 non-null   float64
dtypes: category(1), float64(12), int64(3), object(1)
memory usage: 369.1+ KB
```

1.3 Categorize Countries into Regions

[4]:

```
lst_countries = df.Country.unique()

print('\nNumber of countries evaluated in dataset:', len(lst_countries))

lst_countries
```

```
Number of countries evaluated in dataset: 183
```

```
[4]: array(['Afghanistan', 'Albania', 'Algeria', 'Angola',
            'Antigua and Barbuda', 'Argentina', 'Armenia', 'Australia',
            'Austria', 'Azerbaijan', 'Bahamas', 'Bahrain', 'Bangladesh',
            'Barbados', 'Belarus', 'Belgium', 'Belize', 'Benin', 'Bhutan',
            'Bolivia (Plurinational State of)', 'Bosnia and Herzegovina',
            'Botswana', 'Brazil', 'Brunei Darussalam', 'Bulgaria',
            'Burkina Faso', 'Burundi', "Côte d'Ivoire", 'Cabo Verde',
            'Cambodia', 'Cameroon', 'Canada', 'Central African Republic',
            'Chad', 'Chile', 'China', 'Colombia', 'Comoros', 'Congo',
            'Costa Rica', 'Croatia', 'Cuba', 'Cyprus', 'Czechia',
            "Democratic People's Republic of Korea",
            'Democratic Republic of the Congo', 'Denmark', 'Djibouti',
            'Dominican Republic', 'Ecuador', 'Egypt', 'El Salvador',
            'Equatorial Guinea', 'Eritrea', 'Estonia', 'Ethiopia', 'Fiji',
            'Finland', 'France', 'Gabon', 'Gambia', 'Georgia', 'Germany',
            'Ghana', 'Greece', 'Grenada', 'Guatemala', 'Guinea',
            'Guinea-Bissau', 'Guyana', 'Haiti', 'Honduras', 'Hungary',
            'Iceland', 'India', 'Indonesia', 'Iran (Islamic Republic of)',
            'Iraq', 'Ireland', 'Israel', 'Italy', 'Jamaica', 'Japan', 'Jordan',
            'Kazakhstan', 'Kenya', 'Kiribati', 'Kuwait', 'Kyrgyzstan',
            "Lao People's Democratic Republic", 'Latvia', 'Lebanon', 'Lesotho',
            'Liberia', 'Libya', 'Lithuania', 'Luxembourg', 'Madagascar',
            'Malawi', 'Malaysia', 'Maldives', 'Mali', 'Malta', 'Mauritania',
            'Mauritius', 'Mexico', 'Micronesia (Federated States of)',
            'Mongolia', 'Montenegro', 'Morocco', 'Mozambique', 'Myanmar',
            'Namibia', 'Nepal', 'Netherlands', 'New Zealand', 'Nicaragua',
            'Niger', 'Nigeria', 'Norway', 'Oman', 'Pakistan', 'Panama',
            'Papua New Guinea', 'Paraguay', 'Peru', 'Philippines', 'Poland',
            'Portugal', 'Qatar', 'Republic of Korea', 'Republic of Moldova',
            'Romania', 'Russian Federation', 'Rwanda', 'Saint Lucia',
            'Saint Vincent and the Grenadines', 'Samoa',
            'Sao Tome and Principe', 'Saudi Arabia', 'Senegal', 'Serbia',
            'Seychelles', 'Sierra Leone', 'Singapore', 'Slovakia', 'Slovenia',
            'Solomon Islands', 'Somalia', 'South Africa', 'South Sudan',
            'Spain', 'Sri Lanka', 'Sudan', 'Suriname', 'Swaziland', 'Sweden',
            'Switzerland', 'Syrian Arab Republic', 'Tajikistan', 'Thailand',
            'The former Yugoslav republic of Macedonia', 'Timor-Leste', 'Togo',
            'Tonga', 'Trinidad and Tobago', 'Tunisia', 'Turkey',
            'Turkmenistan', 'Uganda', 'Ukraine', 'United Arab Emirates',
            'United Kingdom of Great Britain and Northern Ireland',
            'United Republic of Tanzania', 'United States of America',
            'Uruguay', 'Uzbekistan', 'Vanuatu',
            'Venezuela (Bolivarian Republic of)', 'Viet Nam', 'Yemen',
            'Zambia', 'Zimbabwe'], dtype=object)
```

1.3.1 NOTE 1:

- This list of countries and their **EIGHT (8) regions** was found on the site:
- https://www.thoughtco.com/official-listing-of-countries-world-region-1435153

[5]:
```
Asia =␣
 ↪['Bangladesh','Bhutan','Brunei','Cambodia','China','India','Indonesia','Japan','Kazakhstan'
        "Democratic People's Republic of Korea",'South Korea','Kyrgyzstan',"Lao␣
 ↪People's Democratic Republic",'Malaysia','Maldives','Mongolia','Myanmar',
        'Nepal','Philippines','Singapore','Sri␣
 ↪Lanka','Taiwan','Tajikistan','Thailand','Turkmenistan',
        'Uzbekistan','Viet Nam']
```

[6]:
```
M_East_N_Africa = ['Afghanistan','Algeria','Azerbaijan','Bahrain','Egypt',"Iran␣
 ↪(Islamic Republic of)",
                  ␣
 ↪'Iraq','Israel','Jordan','Kuwait','Lebanon','Libya','Morocco','Oman','Pakistan','Qatar',
                   'Saudi Arabia','Somalia','Syrian Arab␣
 ↪Republic','Tunisia','Turkey',
                   'United Arab Emirates','Yemen']
```

[7]:
```
Europe = ['Albania','Andorra','Armenia','Austria','Belarus','Belgium','Bosnia␣
 ↪and Herzegovina',
          'Bulgaria','Croatia','Cyprus','Czech␣
 ↪Republic','Denmark','Estonia','Finland','France',
         ␣
 ↪'Georgia','Germany','Greece','Hungary','Iceland','Ireland','Italy','Kosovo','Latvia',
          'Liechtenstein','Lithuania','Luxembourg','Yugoslav republic of␣
 ↪Macedonia','Malta',
          'Republic of Moldova','Monaco','Montenegro',
          'Netherlands','Norway','Poland','Portugal','Romania','Russia','San␣
 ↪Marino','Serbia',
          'Slovakia','Slovenia','Spain','Sweden','Switzerland','Ukraine',
          'United Kingdom of Great Britain and Northern Ireland','Vatican City']
```

[8]:
```
N_America = ['Canada','Greenland','Mexico','United States of America']
```

[9]:
```
C_America_Caribbean = ['Antigua and␣
 ↪Barbuda','Bahamas','Barbados','Belize','Costa Rica','Cuba',
                       'Dominica','Dominican Republic','El␣
 ↪Salvador','Grenada','Guatemala',
                       'Haiti','Honduras','Jamaica','Nicaragua','Panama','Saint␣
 ↪Kitts and Nevis',
                       'Saint Lucia','Saint Vincent and the␣
 ↪Grenadines','Trinidad and Tobago']
```

[10]:

```
S_America =␣
 ↪['Argentina','Bolivia','Brazil','Chile','Colombia','Ecuador','Guyana',
             'Paraguay','Peru','Suriname','Uruguay',"Venezuela (Bolivarian␣
 ↪Republic of)"]
```

[11]:

```
Sub_Saharan_Africa = ['Angola','Benin','Botswana','Burkina␣
 ↪Faso','Burundi','Cameroon','Cape Verde',
                      'The Central African Republic','Chad','Comoros','Republic␣
 ↪of the Congo',
                      'Democratic Republic of the Congo','Côte␣
 ↪d\'Ivoire','Djibouti',
                      'Equatorial␣
 ↪Guinea','Eritrea','Ethiopia','Gabon','Gambia','Ghana',
                     ␣
 ↪'Guinea','Guinea-Bissau','Kenya','Lesotho','Liberia','Madagascar',
                     ␣
 ↪'Malawi','Mali','Mauritania','Mauritius','Mozambique','Namibia','Niger',
                      'Nigeria','Rwanda','Sao Tome and␣
 ↪Principe','Senegal','Seychelles','Sierra Leone',
                      'South Africa','South␣
 ↪Sudan','Sudan','Swaziland','Tanzania','Togo','Uganda',
                      'Zambia','Zimbabwe']
```

[12]:

```
Oceania = ['Australia','Timor-Leste','Fiji','Kiribati','Marshall Islands',
           'Micronesia (Federated States of)','Nauru','Niue','New␣
 ↪Zealand','Palau',
           'Papua New Guinea','Samoa','Solomon␣
 ↪Islands','Tonga','Tuvalu','Vanuatu']
```

[13]:

```
def country_2_region(country):
    """Assign a country name a region. There are EIGHT regions:
    {'Asia':1,
    'M_East_N_Africa':2,
    'S_America':3,
    'N_America':4,
    'Europe':5,
    'Oceania':6,
    'Sub_Saharan_Africa':7,
    'C_America_Caribbean':8}
    """

    region = ''
    if country in Asia:
        region=1
    elif country in M_East_N_Africa:
        region=2
    elif country in S_America:
        region=3
    elif country in N_America:
        region=4
    elif country in Europe:
        region=5
    elif country in Oceania:
        region=6
    elif country in Sub_Saharan_Africa:
        region=7
    else:
        region=8

    return region



# Test with assertions

assert 1==country_2_region('Kazakhstan')

assert 6==country_2_region('Samoa')

assert 2==country_2_region('Algeria')

assert 3==country_2_region('Bolivia')

assert 4==country_2_region('Canada')

assert 5==country_2_region('Andorra')

assert 7==country_2_region('Botswana')

assert 8==country_2_region('Bahamas')

print('Good Job!')
```

```
Good Job!
```

[14]:

```
# Save Files with date-time stamp

# def NamePlusDateTime(file_description, suffix):
#     """This function takes in a file name or desctiption
#     and returns a filename with a date-time stamp suffixed to it."""

#     from datetime import datetime
#     current_datetime = datetime.now()
#     str_current_datetime = str(current_datetime)
#     file_name_DT = file_description + '-' + str_current_datetime + "." + suffix
#     return file_name_DT
```

```
[15]: df['Region'] = df['Country'].apply(country_2_region)

      lst_regions = df.Region.unique()
      lst_regions
```

```
[15]: array([2, 5, 7, 8, 3, 6, 1, 4])
```

1.4 Save engineered data for modeling

```
[16]: file_name = 'Clean_LE_Data_FEng_4.csv'

      df.to_csv(file_name, index=False)

      df.head(3)
```

```
[16]:        Country  Year Status  LifeExpectancy  AdultMort  EtOH  PercExpen  \
      0  Afghanistan  2015      0            65.0      263.0  0.01  71.279624
      1  Afghanistan  2014      0            59.9      271.0  0.01  73.523582
      2  Afghanistan  2013      0            59.9      268.0  0.01  73.219243

         Measles   BMI  lt5yD  Polio  TotalExpen   DTP  HIV  Thin1_19y  Income  \
      0     1154  19.1     83    6.0        8.16  65.0  0.1       17.2   0.479
      1      492  18.6     86   58.0        8.18  62.0  0.1       17.5   0.476
      2      430  18.1     89   62.0        8.13  64.0  0.1       17.7   0.470

         Education  Region
      0       10.1       2
      1       10.0       2
      2        9.9       2
```

```
[17]: !ls *.csv
```

```
Clean_LE_Data_FEng_4.csv      Life_Expectancy_Data.csv  y_test.csv
Clean_LE_Data_Post_EDA_3.csv  x_test.csv                y_train.csv
Clean_LE_Data_w_Means_2.csv   x_train.csv
```

```
[ ]:
```