pranav_thing

July 11, 2021

[1]:
```
import pandas as pd
import sklearn

import surgeo
```

1 With NaNs

[2]:
```
# Instatiate your model
fsg = surgeo.BIFSGModel()

# Create pd.Series objects to analze (or load them)
first_names = pd.Series(['HECTOR', 'UNCOMMON_FORENAME', 'JANICE'])
surnames = pd.Series(['DIAZ', 'UNCOMMON_SURNAME', 'WASHINGTON'])
zctas = pd.Series(['65201', '00000', '63110'])

# Get results using the get_probabilities() function
fsg_results = fsg.get_probabilities(first_names, surnames, zctas)

# Show Surgeo BIFSG results
fsg_results
```

[2]:

	zcta5	first_name	surname	white	black	api	native	multiple	hispanic
0	65201	HECTOR	DIAZ	0.003834	0.000225	0.000688	0.000000	0.000000	0.995253
1	00000	UNCOMMONFORENAME	UNCOMMONSURNAME	NaN	NaN	NaN	NaN	NaN	NaN
2	63110	JANICE	WASHINGTON	0.017885	0.968260	0.000059	0.000356	0.013339	0.000101

2 Imputation functions

[3]:
```
def give_tables_placeholder_value(fsg_model):
    '''Add values for impute to the data tables

       These need to have no spaces due to the way the model mangles/
↪standardizes names
    '''
    forename_table = fsg_model._PROB_FIRST_NAME_GIVEN_RACE
    forename_table.loc['IMPUTEVALUE'] = forename_table.loc['ALL OTHER FIRST␣
↪NAMES'].values
    surname_table = fsg_model._PROB_RACE_GIVEN_SURNAME
    surname_table.loc['IMPUTEVALUE'] = surname_table.loc['ALL OTHER NAMES'].
↪values
    zcta_table = fsg_model._PROB_ZCTA_GIVEN_RACE
    # From: https://www.census.gov/prod/cen2010/briefs/c2010br-02.pdf
    white = .640
    black = .123
    api = .048
    native = .007
    multiple = .019
    hispanic = .163
    zcta_table.loc['IMPUTEVALUE'] = [white, black, api, native, multiple,␣
↪hispanic]
    return fsg_model
```

[4]:
```
def populate_dummies(series, model, datatype):
    '''This takes your data and fills in placeholders where no lookup data␣
↪exists'''
    dtype_lookup = {
        'surname': model._PROB_RACE_GIVEN_SURNAME,
        'first_name': model._PROB_FIRST_NAME_GIVEN_RACE,
        'zcta5': model._PROB_ZCTA_GIVEN_RACE,
    }
    table = dtype_lookup[datatype]
    not_present_bixer = ~(series.isin(table.index))
    series.loc[not_present_bixer] = 'IMPUTEVALUE'
    return series
```

3 With imputed info

[5]:
```
# Instatiate your model
fsg = surgeo.BIFSGModel()

# Add placeholders to data tbles
fsg = give_tables_placeholder_value(fsg)

# Create pd.Series objects to analze (or load them)
first_names = pd.Series(['HECTOR', 'UNCOMMON_FORENAME', 'JANICE'])
surnames = pd.Series(['DIAZ', 'UNCOMMON_SURNAME', 'WASHINGTON'])
zctas = pd.Series(['65201', '00000', '63110'])

# Feed copies of your series for dummy values
first_names_dummies = populate_dummies(first_names.copy(), fsg, 'first_name')
surnames_dummies = populate_dummies(surnames.copy(), fsg, 'surname')
zctas_dummies = populate_dummies(zctas.copy(), fsg, 'zcta5')

# Get results using the get_probabilities() function
fsg_results = fsg.get_probabilities(
    first_names_dummies,
    surnames_dummies,
    zctas_dummies,
)

# Put original input values back in
fsg_results['zcta5'] = zctas.values
fsg_results['first_name'] = first_names.values
fsg_results['surname'] = surnames.values

# Show Surgeo BIFSG results
fsg_results
```

[5]:

	zcta5	first_name	surname	white	black	api	native	multiple	hispanic
0	65201	HECTOR	DIAZ	0.003834	0.000225	0.000688	0.000000	0.000000	0.995253
1	00000	UNCOMMON_FORENAME	UNCOMMON_SURNAME	0.878154	0.057865	0.016068	0.000254	0.001871	0.045788
2	63110	JANICE	WASHINGTON	0.017885	0.968260	0.000059	0.000356	0.013339	0.000101

4 Explanation

Surgeo requires valid data points for surnames, forenames, and zctas in order to work. This is because each name/zcta/surname value is associated with a particular probability in the following lookup tables:

`model._PROB_FIRST_NAME_GIVEN_RACE`
`model._PROB_RACE_GIVEN_SURNAME`
`model._PROB_ZCTA_GIVEN_RACE`

In other words, if we don't have data for a last name like `Beeblebrox` surgeo cannot properly calculate what that persons name is because it doesn't know what values to look up.

To get around this we can put placeholder values for the data we want to impute for a given step. We do this by putting in imputation values in the underlying surname, forename, and zcta tables. For forename and surname this data is already supplied under the "ALL OTHER NAMES" and "ALL OTHER FIRST NAMES" rows. For ZCTA we can calculate this fairly easily by looking at race on a national basis instead of by ZIP/ZCTA5.

Note: we cannot simply use "ALL OTHER NAMES" and "ALL OTHER FIRST NAMES" because surgeo mangles your inputs to remove numbers and spaces in a manner similar to the academic papers it is based on. In other words, "ALL OTHER NAMES" becomes "ALLOTHERNAMES" which will return NaN because the lookup table has "ALL OTHER NAMES" (admittedly an oversight on my part).

In summary … to make this work, we: 1. Put in a known imputation value in each of our lookup tables (add impute value to _PROB_RACE_GIVEN_SURNAME) 2. Do a pre-lookup to see which of our input data is missing (chewck is Beeblebrox in _PROB_RACE_GIVEN_SURNAME") 3. Replace non-present values with placeholder (change Beeblebrox to IMPUTEVALUE) 4. Run the calculation 5. Change your names back (change IMPUTEVALUE to Beeblebrox)

It's inelegant but it *should* work. I'd check the math to make sure.