logistic_regression

April 25, 2023

[ ]:
```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import statsmodels.api as sm
import statsmodels.formula.api as smf
import matplotlib as mpl
import matplotlib.cm as cm
```

[ ]:
```
assessor_dir = 'datasets/assessorSequential.csv'
df = pd.read_csv(assessor_dir)
```

[ ]:
```
df = df.drop(columns=['Unnamed: 0','X11th.Draw','PIN','Township␣
↪Code','Neighborhood Code','Age','Longitude','Latitude','ZIP'])
```

[ ]:
```
draw_col = df.columns[2:12]
draw_col
```

[ ]:
```
Index(['X1st.Draw', 'X2nd.Draw', 'X3rd.Draw', 'X4th.Draw', 'X5th.Draw',
       'X6th.Draw', 'X7th.Draw', 'X8th.Draw', 'X9th.Draw', 'X10th.Draw'],
      dtype='object')
```

[ ]:
```
df['avg_draw'] = df[draw_col].mean(axis=1)
df = df.drop(columns=draw_col)
```

[ ]:
```
df_full = df.copy()
```

[ ]:
```
df_reduced = df.copy()
df_full.head(5)
```

[ ]:
```
  Date.Sampled              Address  Sale Price  Tract Median Income  \
0     9/4/2019     100XX S BELL AVE    280000.0             122727.0
1    7/16/2016   100XX S CALHOUN AVE         NaN              44423.0
2   12/17/2019   100XX S CALUMET AVE         NaN              40612.0
3   12/14/2019  100XX S CARPENTER ST         NaN              37207.0
4    7/14/2021  100XX S CARPENTER ST         NaN              37207.0

   avg_draw
0     2.433
```

```
1     9.866
2    10.399
3     8.663
4     9.280
```

[ ]:
```
df_full[~df_full.isin(['NaN', 'NaT']).any(axis=1)]
df_full.shape
df_full.dropna(inplace=True)
df_full.shape
```

[ ]:
```
(149, 5)
```

[ ]:
```
#assign 0 to <7.5 and 1 to >=7.5
df_full['avg_lead_target'] = np.where(df_full['avg_draw']<8.55, 0, 1)
```

[ ]:
```
df_full.head(10)
```

[ ]:
```
   Date.Sampled                  Address  Sale Price  Tract Median Income  \
0      9/4/2019          100XX S BELL AVE    280000.0              122727.0
17   12/20/2016      102XX S ARTESIAN AVE    219000.0               98281.0
20    9/17/2019       102XX S OGLESBY AVE    147400.0               30069.0
21    9/28/2016      103XX S HAMILTON AVE    330000.0              110344.0
36   10/10/2019  105XX S CENTRAL PARK AVE    280000.0              100361.0
42    11/4/2021        105XX S KEDZIE AVE    290000.0               91924.0
44    8/30/2021        105XX S SEELEY AVE    464000.0              110344.0
49   11/30/2021      106XX S EBERHART AVE     86000.0               45273.0
65    12/6/2021     108XX S EGGLESTON AVE    146000.0               41167.0
67     3/2/2022        108XX S HAMLIN AVE    273000.0              118640.0

    avg_draw  avg_lead_target
0      2.433                0
17     5.192                0
20    27.250                1
21     6.315                0
36    10.505                1
42    14.775                1
44     5.901                0
49    15.010                1
65     2.018                0
67     4.998                0
```

[ ]:
```
df_full['avg_lead_target'].value_counts()
```

[ ]:
```
1    76
0    73
Name: avg_lead_target, dtype: int64
```

[ ]:
```
df_full['sale_price'] = df_full['Sale Price']
df_full['tract_income'] = df_full['Tract Median Income']
df_full = df_full.drop(columns=['Sale Price','Tract Median Income'])
```

[ ]:
```
class_0 = df_full[df_full['avg_lead_target']==0]
print(class_0.shape)
class_1 = df_full[df_full['avg_lead_target']==1]

class_1_under = class_1.sample(class_0.shape[0])

df_balanced = pd.concat([class_0, class_1_under], axis=0)
```

```
(73, 6)
```

[ ]:
```
df_balanced['avg_lead_target'].value_counts()
```

[ ]:
```
0    73
1    73
Name: avg_lead_target, dtype: int64
```

[ ]:
```
df_balanced
```

[ ]:
```
     Date.Sampled                Address  avg_draw  avg_lead_target  \
0        9/4/2019       100XX S BELL AVE     2.433                0
17     12/20/2016   102XX S ARTESIAN AVE     5.192                0
21      9/28/2016   103XX S HAMILTON AVE     6.315                0
44      8/30/2021     105XX S SEELEY AVE     5.901                0
65      12/6/2021  108XX S EGGLESTON AVE     2.018                0
…             …                      …       …                …
1511     3/1/2022      74XX N ORIOLE AVE    10.418                1
948     12/9/2019    49XX W WAVELAND AVE    10.626                1
1371    1/22/2020   64XX N FAIRFIELD AVE    12.522                1
163     9/24/2019     121XX S WALLACE ST     8.895                1
1237    6/15/2021    59XX S KILBOURN AVE    29.810                1

      sale_price  tract_income
0       280000.0      122727.0
17      219000.0       98281.0
21      330000.0      110344.0
44      464000.0      110344.0
65      146000.0       41167.0
…            …             …
1511    415600.0       95990.0
948     424900.0       56917.0
1371    400000.0       41308.0
163      20000.0       44436.0
1237    350000.0       48678.0
```

[146 rows x 6 columns]

[ ]:
```
log_reg = smf.logit(formula='avg_lead_target ~ tract_income + sale_price',␣
↪data=df_balanced).fit()
```

```
Optimization terminated successfully.
         Current function value: 0.684165
         Iterations 4
```

[ ]:
```
print(log_reg.summary())
```

```
                           Logit Regression Results
==============================================================================
Dep. Variable:        avg_lead_target   No. Observations:                  146
Model:                          Logit   Df Residuals:                      143
Method:                           MLE   Df Model:                            2
Date:                Tue, 25 Apr 2023   Pseudo R-squ.:                 0.01296
Time:                        01:16:04   Log-Likelihood:                -99.888
converged:                       True   LL-Null:                       -101.20
Covariance Type:            nonrobust   LLR p-value:                    0.2694
================================================================================
                   coef    std err          z      P>|z|      [0.025      0.975]
--------------------------------------------------------------------------------
Intercept       -0.1875      0.417     -0.450      0.653      -1.005       0.630
tract_income -3.737e-06   7.44e-06     -0.502      0.615   -1.83e-05    1.08e-05
sale_price    1.166e-06   8.08e-07      1.443      0.149   -4.18e-07    2.75e-06
================================================================================
```

[ ]:
```
log_reg_reduced = smf.logit(formula='avg_lead_target ~ sale_price',␣
↪data=df_full).fit()
```

```
Optimization terminated successfully.
         Current function value: 0.683988
         Iterations 5
```

[ ]:
```
log_reg_reduced.summary()
```

[ ]:
```
<class 'statsmodels.iolib.summary.Summary'>
"""
                           Logit Regression Results
==============================================================================
Dep. Variable:        avg_lead_target   No. Observations:                  149
Model:                          Logit   Df Residuals:                      147
Method:                           MLE   Df Model:                            1
Date:                Tue, 25 Apr 2023   Pseudo R-squ.:                 0.01293
Time:                        01:16:05   Log-Likelihood:                -101.91
converged:                       True   LL-Null:                       -103.25
Covariance Type:            nonrobust   LLR p-value:                    0.1023
==============================================================================
```

```
                 coef    std err          z      P>|z|      [0.025      0.975]
------------------------------------------------------------------------------
Intercept     -0.3247      0.282     -1.152      0.249      -0.877       0.228
sale_price  9.644e-07   6.14e-07      1.571      0.116   -2.39e-07    2.17e-06
==============================================================================
"""
```

[ ]:
```
df_reduced = df_reduced[['Tract Median Income','avg_draw']]
df_reduced['avg_lead_target'] = np.where(df_reduced['avg_draw']<9, 0, 1)
df_reduced['tract_income'] = df_reduced['Tract Median Income']
df_reduced = df_reduced.drop(columns=['Tract Median Income'])
```

[ ]:
```
class_0 = df_reduced[df_reduced['avg_lead_target']==0]
print(class_0.shape)
class_1 = df_reduced[df_reduced['avg_lead_target']==1]

class_1_under = class_1.sample(class_0.shape[0])

df_balanced1 = pd.concat([class_0, class_1_under], axis=0)
```

```
(816, 3)
```

[ ]:
```
df_balanced1['avg_lead_target'].value_counts()
```

[ ]:
```
0    816
1    816
Name: avg_lead_target, dtype: int64
```

[ ]:
```
log_reg1 = smf.logit(formula='avg_lead_target ~ tract_income',␣
↪data=df_balanced1).fit()
```

```
Optimization terminated successfully.
         Current function value: 0.692576
         Iterations 3
```

[ ]:
```
log_reg1.summary()
```

[ ]:
```
<class 'statsmodels.iolib.summary.Summary'>
"""
                           Logit Regression Results
==============================================================================
Dep. Variable:        avg_lead_target   No. Observations:                 1109
Model:                          Logit   Df Residuals:                     1107
Method:                           MLE   Df Model:                            1
Date:                Tue, 25 Apr 2023   Pseudo R-squ.:               0.0006130
Time:                        01:16:05   Log-Likelihood:                -768.07
converged:                       True   LL-Null:                       -768.54
Covariance Type:            nonrobust   LLR p-value:                    0.3317
```

```
==============================================================================
                 coef    std err          z      P>|z|      [0.025      0.975]
------------------------------------------------------------------------------
Intercept        0.0982      0.149      0.658      0.510      -0.194       0.391
tract_income -1.997e-06   2.06e-06     -0.970      0.332   -6.03e-06    2.04e-06
==============================================================================
"""
```

[ ]: