hw

June 13, 2023

[2]:
```
import pandas as pd
import numpy as np
import statsmodels.api as sm
from sklearn import metrics
import matplotlib.pyplot as plt
```

[3]:
```
data = pd.read_excel('data.xlsx')
data
```

[3]:

	recid	black	alcohol	drugs	married	felon	educ	rules	age
0	0.0	0	1	0	1	0	7	2	441
1	0.0	1	0	0	0	1	12	0	307
2	1.0	0	0	1	0	1	9	3	253
3	0.0	0	0	1	0	0	9	0	244
4	0.0	1	0	0	0	0	12	0	277
…	…	…	…	…	…	…	…	…	…
1006	NaN	0	0	0	0	0	10	0	231
1007	NaN	0	0	0	0	0	9	2	290
1008	NaN	0	0	1	0	0	12	5	236
1009	NaN	0	1	1	0	0	12	0	393
1010	NaN	0	0	0	0	0	8	2	252

[1011 rows x 9 columns]

[4]:
```
age_22 = []
age_30 = []
for i in data['age']:
    if (i // 12 <= 22):
        age_22.append(1)
        age_30.append(0)
    elif (i // 12 > 30):
        age_22.append(0)
        age_30.append(1)
    else:
        age_22.append(0)
        age_30.append(0)
```

[5]:
```
YVar = data[['recid']]
XVar = data.drop(['recid', 'age'], axis=1)
XVar['age_22'] = age_22
XVar['age_30'] = age_30
XVar
```

[5]:

	black	alcohol	drugs	married	felon	educ	rules	age_22	age_30
0	0	1	0	1	0	7	2	0	1
1	1	0	0	0	1	12	0	0	0
2	0	0	1	0	1	9	3	1	0
3	0	0	1	0	0	9	0	1	0
4	1	0	0	0	0	12	0	0	0
…	…	…	…	…	…	…	…	…	…
1006	0	0	0	0	0	10	0	1	0
1007	0	0	0	0	0	9	2	0	0
1008	0	0	1	0	0	12	5	1	0
1009	0	1	1	0	0	12	0	0	1
1010	0	0	0	0	0	8	2	1	0

[1011 rows x 9 columns]

[6]:
```
# OLS

linearModel = sm.OLS(YVar, XVar, missing='drop').fit()
print(linearModel.summary())
yhat1 = linearModel.predict(XVar)
print('Predicted recidive\n', yhat1)
```

```
                                 OLS Regression Results
================================================================================
=======
Dep. Variable:                   recid   R-squared (uncentered):
0.399
Model:                             OLS   Adj. R-squared (uncentered):
0.393
Method:                  Least Squares   F-statistic:
70.21
Date:                 Tue, 13 Jun 2023   Prob (F-statistic):
4.83e-99
Time:                         14:50:24   Log-Likelihood:
-649.86
No. Observations:                 961   AIC:
1318.
Df Residuals:                     952   BIC:
1362.
Df Model:                           9
Covariance Type:            nonrobust
================================================================================
```

```
                 coef    std err          t      P>|t|      [0.025      0.975]
------------------------------------------------------------------------------
black          0.1610      0.031      5.161      0.000       0.100       0.222
alcohol        0.1301      0.039      3.336      0.001       0.054       0.207
drugs          0.1001      0.036      2.802      0.005       0.030       0.170
married       -0.0309      0.036     -0.857      0.392      -0.102       0.040
felon          0.0044      0.035      0.126      0.900      -0.064       0.073
educ           0.0169      0.003      5.206      0.000       0.011       0.023
rules          0.0255      0.007      3.700      0.000       0.012       0.039
age_22         0.1342      0.036      3.749      0.000       0.064       0.204
age_30         0.0194      0.036      0.536      0.592      -0.052       0.090
==============================================================================
Omnibus:                     7757.653   Durbin-Watson:                   2.022
Prob(Omnibus):                  0.000   Jarque-Bera (JB):              117.465
Skew:                           0.437   Prob(JB):                     3.11e-26
Kurtosis:                       1.527   Cond. No.                         28.7
==============================================================================

Notes:
[1] R² is computed without centering (uncentered) since the model does not
contain a constant.
[2] Standard Errors assume that the covariance matrix of the errors is correctly
specified.
Predicted recidive
 0       0.287626
1        0.367929
2        0.466948
3        0.386146
4        0.363525
           …
1006     0.302969
1007     0.202810
1008     0.564102
1009     0.452068
1010     0.320150
Length: 1011, dtype: float64
```

[7]:
```
# LOGIT

logitModel = sm.Logit(YVar, XVar, missing='drop').fit()
print(logitModel.summary())
print(logitModel.cov_params())
yhat2 = logitModel.predict(XVar)
print('\nPredicted recidive\n', yhat2, '\n\n')

# NO REGRESSORS
```

```
Xconst = YVar.drop(['recid'], axis=1)
Xconst = sm.add_constant(Xconst)
const_model = sm.Logit(YVar, Xconst, missing='drop').fit()
print(const_model.summary())
```

```
Optimization terminated successfully.
         Current function value: 0.637887
         Iterations 5
                           Logit Regression Results
==============================================================================
Dep. Variable:                   recid   No. Observations:                  961
Model:                           Logit   Df Residuals:                      952
Method:                            MLE   Df Model:                            8
Date:                Tue, 13 Jun 2023   Pseudo R-squ.:                 0.03704
Time:                        14:50:25   Log-Likelihood:                -613.01
converged:                       True   LL-Null:                       -636.59
Covariance Type:            nonrobust   LLR p-value:                 1.433e-07
==============================================================================
                 coef    std err          z      P>|z|      [0.025      0.975]
------------------------------------------------------------------------------
black          0.5539      0.139      3.987      0.000       0.282       0.826
alcohol        0.3971      0.172      2.308      0.021       0.060       0.734
drugs          0.2427      0.157      1.548      0.122      -0.065       0.550
married       -0.3612      0.165     -2.191      0.028      -0.684      -0.038
felon         -0.1355      0.156     -0.867      0.386      -0.442       0.171
educ          -0.0861      0.015     -5.825      0.000      -0.115      -0.057
rules          0.0990      0.033      3.035      0.002       0.035       0.163
age_22         0.1906      0.156      1.224      0.221      -0.115       0.496
age_30        -0.3911      0.163     -2.393      0.017      -0.711      -0.071
==============================================================================
            black   alcohol     drugs   married     felon      educ     rules  \
black    0.019297  0.002682  0.002698 -0.000153 -0.000647 -0.001052 -0.000060
alcohol  0.002682  0.029598  0.000978 -0.002798  0.003307 -0.000707  0.000087
drugs    0.002698  0.000978  0.024594  0.000087  0.000678 -0.000620 -0.000363
married -0.000153 -0.002798  0.000087  0.027186 -0.001766 -0.000577  0.000044
felon   -0.000647  0.003307  0.000678 -0.001766  0.024430 -0.000553 -0.001343
educ    -0.001052 -0.000707 -0.000620 -0.000577 -0.000553  0.000219 -0.000071
rules   -0.000060  0.000087 -0.000363  0.000044 -0.001343 -0.000071  0.001064
age_22   0.000730  0.001243 -0.000895  0.004329  0.000524 -0.001129 -0.000307
age_30  -0.001371 -0.006406 -0.003290 -0.002924 -0.000829 -0.000549  0.000184

           age_22    age_30
black    0.000730 -0.001371
alcohol  0.001243 -0.006406
drugs   -0.000895 -0.003290
married  0.004329 -0.002924
felon    0.000524 -0.000829
educ    -0.001129 -0.000549
```

```
rules   -0.000307  0.000184
age_22   0.024250  0.007694
age_30   0.007694  0.026714

Predicted recidive
 0       0.318581
1        0.350857
2        0.455021
3        0.415343
4        0.382300
           …
1006     0.338330
1007     0.359565
1008     0.473687
1009     0.313231
1010     0.425441
Length: 1011, dtype: float64


Optimization terminated successfully.
         Current function value: 0.662421
         Iterations 4
                           Logit Regression Results
==============================================================================
Dep. Variable:                   recid   No. Observations:                  961
Model:                           Logit   Df Residuals:                      960
Method:                            MLE   Df Model:                            0
Date:                Tue, 13 Jun 2023   Pseudo R-squ.:               1.027e-11
Time:                        14:50:25   Log-Likelihood:                -636.59
converged:                       True   LL-Null:                       -636.59
Covariance Type:            nonrobust   LLR p-value:                       nan
==============================================================================
                 coef    std err          z      P>|z|      [0.025      0.975]
------------------------------------------------------------------------------
const         -0.5036      0.067     -7.565      0.000      -0.634      -0.373
==============================================================================
```

[13]:
```
# PROBIT

probitModel = sm.Probit(YVar, XVar, missing='drop').fit()
print(probitModel.summary())
yhat3 = probitModel.predict(XVar)
print('Predicted recidive\n', yhat3)
```

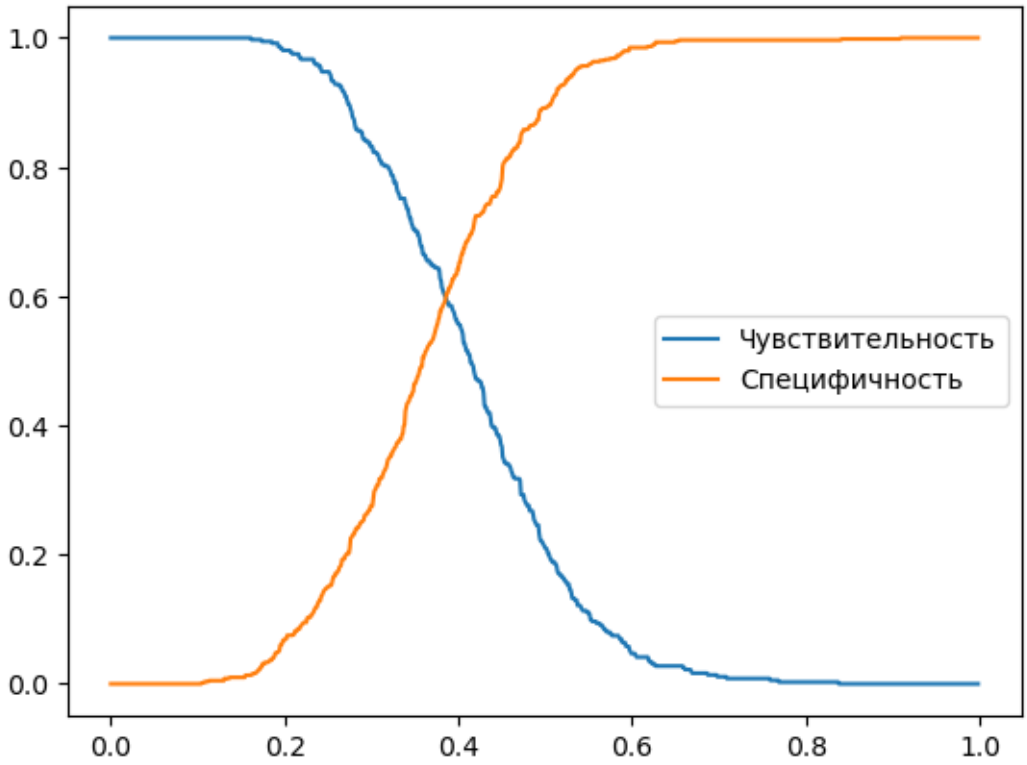
```
Optimization terminated successfully.
         Current function value: 0.638014
         Iterations 5
                          Probit Regression Results
```

```
==============================================================================
Dep. Variable:                  recid   No. Observations:                  961
Model:                         Probit   Df Residuals:                      952
Method:                           MLE   Df Model:                            8
Date:                Tue, 13 Jun 2023   Pseudo R-squ.:                 0.03684
Time:                        15:01:44   Log-Likelihood:                -613.13
converged:                       True   LL-Null:                       -636.59
Covariance Type:            nonrobust   LLR p-value:                 1.595e-07
==============================================================================
                 coef    std err          z      P>|z|      [0.025      0.975]
------------------------------------------------------------------------------
black          0.3430      0.085      4.039      0.000       0.177       0.509
alcohol        0.2443      0.105      2.331      0.020       0.039       0.450
drugs          0.1490      0.096      1.549      0.121      -0.040       0.338
married       -0.2254      0.100     -2.254      0.024      -0.421      -0.029
felon         -0.0794      0.095     -0.836      0.403      -0.266       0.107
educ          -0.0529      0.009     -5.923      0.000      -0.070      -0.035
rules          0.0572      0.018      3.147      0.002       0.022       0.093
age_22         0.1132      0.096      1.177      0.239      -0.075       0.302
age_30        -0.2361      0.099     -2.388      0.017      -0.430      -0.042
==============================================================================
Predicted recidive
 0       0.318180
1        0.355349
2        0.451700
3        0.415441
4        0.385341
           …
1006     0.338903
1007     0.358896
1008     0.465685
1009     0.316611
1010     0.422581
Length: 1011, dtype: float64
```

[14]:
```python
# P(low_i=1)>c

c = []
yhat = []
for i in range(999):
    c.append(i / 1000)
    yhat_i = []
    for j in range(961):
        if (yhat3[j] > c[i]):
            yhat_i.append(1)
        else:
            yhat_i.append(0)
```

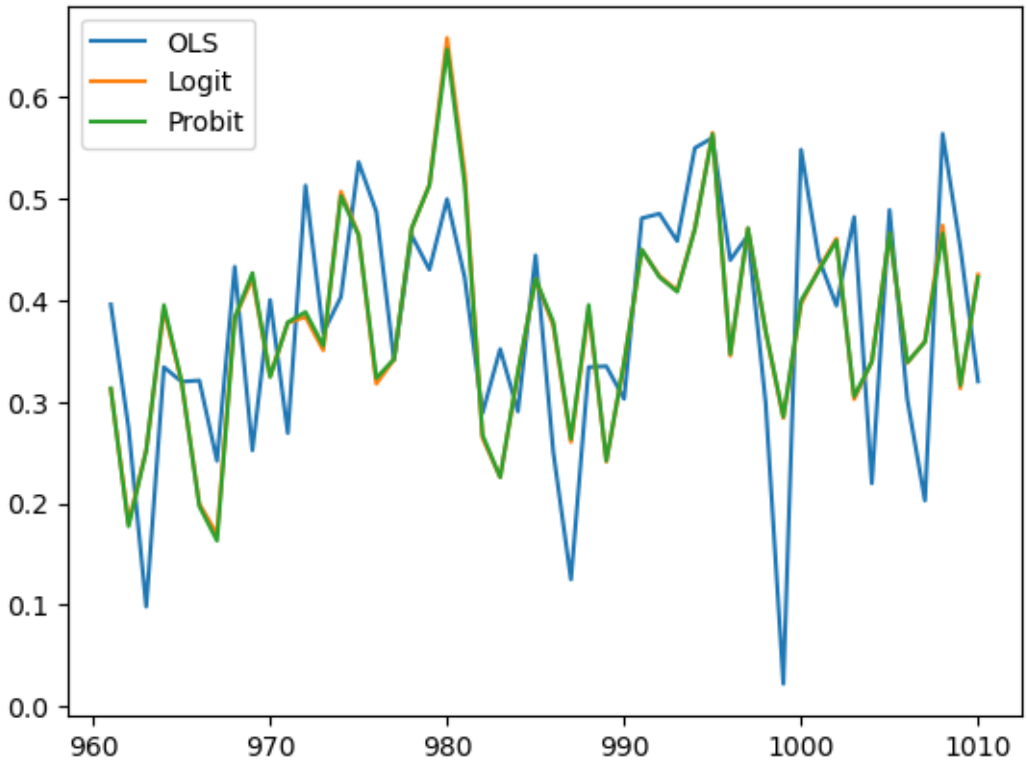
```
    yhat.append(yhat_i)

metric1 = []
metric2 = []
for i in range(999):
    count1 = 0
    count2 = 0
    count_ones = 0
    count_zeroes = 0
    for j in range(961):
        if (data['recid'][j] == 1):
            count_ones += 1
            if (data['recid'][j] == yhat[i][j]):
                count1 += 1
        else:
            count_zeroes += 1
            if (data['recid'][j] == yhat[i][j]):
                count2 += 1
    metric1.append(count1/count_ones)
    metric2.append(count2/count_zeroes)
plt.plot(c, metric1)
plt.plot(c, metric2)
plt.legend(['        ', '       '])
```

[14]: <matplotlib.legend.Legend at 0x133a9827640>



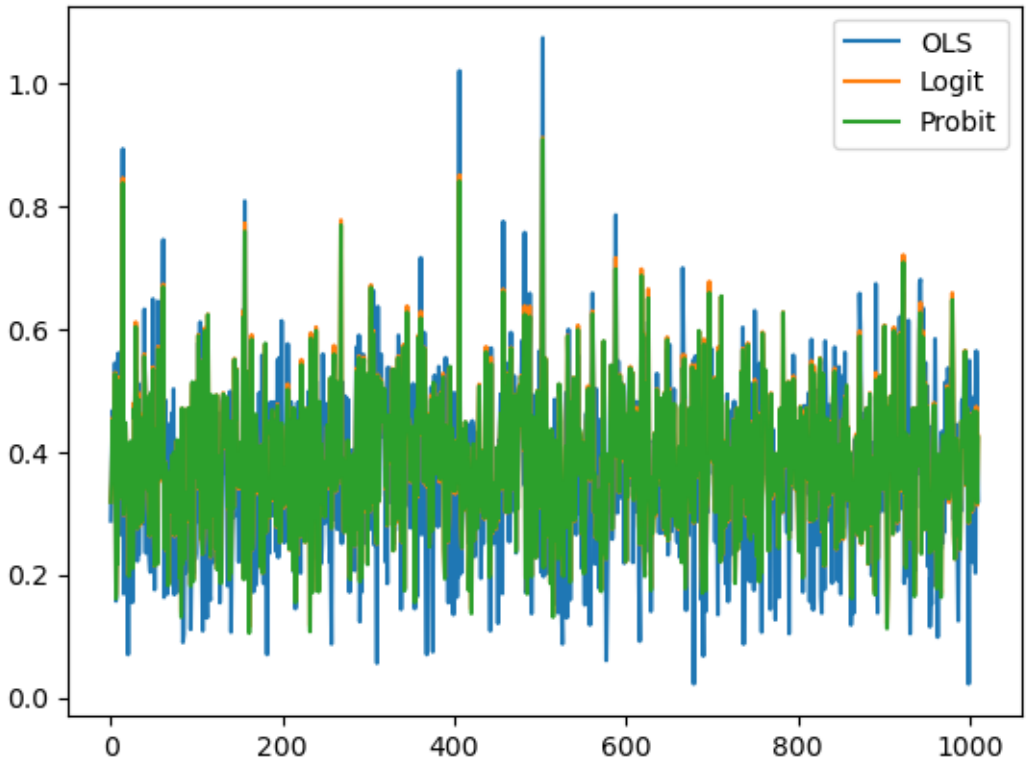
[15]:
```
m1 = max(metric1)
m2 = max(metric2)
c_i = 0
k = 0
for i in range(999):
    if (metric1[i] < 0.7):
        k = i
        c_i = i / 1000
        break
plt.plot(yhat1[-50:])
plt.plot(yhat2[-50:])
plt.plot(yhat3[-50:])
plt.legend(['OLS', 'Logit', 'Probit'])
m1,c_i,metric2[k]
```

[15]: (1.0, 0.353, 0.4757929883138564)



[18]:
```
plt.plot(yhat1)
plt.plot(yhat2)
plt.plot(yhat3)
plt.legend(['OLS', 'Logit', 'Probit'])
```

[18]: <matplotlib.legend.Legend at 0x133a9910250>



[16]:
```
# max in new data

m = max(yhat2[-50:])
k = 0
for i in range(960, 1011):
    if (yhat2[i] == m):
        k = i
        break
m, k
```

[16]: (0.6585270963176593, 980)

[17]:
```
# in all data

attention = []
m = 0
max_i = -1
for i in range(961):
    if (data['recid'][i] < 1 and yhat2[i] >= 0.5):
        attention.append(i)
        if (yhat2[i] >= m ):
            m = yhat2[i]
```

```
            max_i = i
print(attention)
m, i
```

```
[5, 9, 10, 39, 100, 102, 110, 113, 144, 146, 158, 164, 222, 239, 263, 294, 295,
315, 328, 330, 335, 361, 376, 395, 406, 459, 463, 466, 476, 478, 479, 502, 503,
522, 532, 539, 561, 574, 591, 593, 619, 624, 626, 636, 648, 666, 677, 683, 685,
703, 708, 711, 742, 762, 783, 791, 829, 842, 857, 891, 896, 919, 943, 946, 949]
```

[17]: (0.9114130126717576, 960)