STAT 656 Homework 3

E. Lee Rainwater – lee.rainwater@tamu.edu

1 Executive Summary

The supplied data, from *credithistory.xlsx*, contained 1000 observations, the details of which are listed in Appendix 1. The response variable was taken to be the feature, *good_bad*, which is a binary classification of whether the applicant is expected to have good or bad credit.

A total of ten logistic regression models were run:

- Logistic with all features with model validation
- Logistic with stepwise feature selection
- Logistic with regularization
    - Regularization coefficients ranging from 1.0 x 10^{-4} to +∞

For each model, the dataset was randomly split 70%/30% into test/validation datasets. Metrics for each of the validation cases are shown in Table 1:

Model	Precision	Sensisivity	Validation Specificity	F1	MISC
All Features with Validation	0.8186	0.8739	0.4487	0.8453	23.7%
Logistic with Stepwise	0.8238	0.9054	0.4487	0.8627	21.3%
Regularization Logistic, C = 0.0001	0.7492	0.9955		0.8549	25.0%
Regularization Logistic, C = 0.01	0.7762	0.9685		0.8617	23.0%
Regularization Logistic, C = 0.1	0.8118	0.9324		0.8679	21.0%
Regularization Logistic, C = 1	0.8139	0.8468		0.8300	25.7%
Regularization Logistic, C = 5	0.8162	0.8604		0.8377	24.7%
Regularization Logistic, C = 10	0.8194	0.8378		0.8285	25.7%
Regularization Logistic, C = 50	0.8178	0.8694		0.8428	24.0%
Regularization Logistic, C = ∞	0.8248	0.8694		0.8465	23.3%

Table 1 - Model Validation Metrics

On cursory investigation, the models appear to be very close in quality. Given that the target variable is the binary determination of the customer's credit (good/bad), it may be surmised that the effects are of false negatives vs. false positives. A false negative will result in a loan being not granted to a qualified customer, resulting in the loss of associated profit. A false positive, on the other hand, *may* result in the loss of the entire value of the loan, plus expenses. Thus, it is surmised that the negative consequences of a false positive are greater than those of a false negative.

For this reason, *sensitivity* is chosen as the primary metric for evaluation, as it more strongly considers the effect of false positives. As can be seen in Table 1, *Regularization Logistic, C = 0.0001*provides the highest degree of sensitivity.

A more in-depth analysis would include consideration of the actual amount of profit loss due to false negatives, as the lower *precision* value for this same model indicates that more applicants may be falsely rejected.

2 Appendix 1 – Python Output

2.1 Imputation & Outliers

```
............... Missing  Outliers
age.......        35        6
amount....        12        9
checking..         0        0
coapp.....        12        0
depends...         0        0
duration..        42        0
employed..         0        6
existcr...         0        0
foreign...         0        0
history...         0        0
housing...         0        0
installp..         0        0
job.......         0        0
marital...         9        5
other.....         0        0
property..         0        0
purpose...       564        0
resident..        11        0
savings...         4        2
telephon..        19        0
good_bad..         0        0
```

2.2 Logistic Regression – All Features with Validation

```
      Python 3.7.6 (default, Jan  8 2020, 13:42:34)
Type "copyright", "credits" or "license" for more information.

IPython 7.13.0 -- An enhanced Interactive Python.

runfile('/Users/edwardrainwater/OneDrive - Texas A&M University/Summer-2020/STAT 656
Applied Analytics/hw-03/rainwater-stat656-hw03.py', wdir='/Users/edwardrainwater/OneDrive
- Texas A&M University/Summer-2020/STAT 656 Applied Analytics/hw-03')
   checking  duration  history purpose  ...  depends  telephon  foreign  good_bad
0         1       NaN        4       3  ...        1       NaN        1      good
1         2      48.0        2     NaN  ...        1       NaN        1       bad
2         4      12.0        4     NaN  ...        2       NaN        1      good
3         1      42.0        2     NaN  ...        2       NaN        1      good
4         1      24.0        3     NaN  ...        2       NaN        1       bad

[5 rows x 21 columns] (1000, 21)
checking      int64
duration    float64
history       int64
purpose      object
amount      float64
savings     float64
employed      int64
installp      int64
marital     float64
```

```
coapp       float64
resident    float64
property      int64
age         float64
other         int64
housing       int64
existcr       int64
job           int64
depends       int64
telephon    float64
foreign       int64
good_bad     object
dtype: object

********** Data Preprocessing ***********
Features Dictionary Contains:
3 Interval,
4 Binary,
13 Nominal, and
1 Excluded Attribute(s).

Data contains 1000 observations & 21 columns.

/Users/edwardrainwater/opt/anaconda3/envs/Stat656MacOSX/lib/python3.7/site-
packages/AdvancedAnalytics/ReplaceImputeEncode.py:338: UserWarning: purpose:has more than
50% missing.Recommend setting Data Type set to DT.Ignore.
  "Recommend setting Data Type set to DT.Ignore.")

Attribute Counts
............... Missing  Outliers
age.......        35        6
amount....        12        9
checking..         0        0
coapp.....        12        0
depends...         0        0
duration..        42        0
employed..         0        6
existcr...         0        0
foreign...         0        0
history...         0        0
housing...         0        0
installp..         0        0
job.......         0        0
marital...         9        5
other.....         0        0
property..         0        0
purpose...       564        0
resident..        11        0
savings...         4        2
telephon..        19        0
good_bad..         0        0

******************************************************************************
*************  Running StatsModel - All Features with Validation   *************
```

```
******************************************************************************
Warning: Maximum number of iterations has been exceeded.
         Current function value: 0.466458
         Iterations: 35
Printing results summary...
                           Logit Regression Results
==============================================================================
Dep. Variable:               good_bad   No. Observations:                  700
Model:                          Logit   Df Residuals:                      654
Method:                           MLE   Df Model:                           45
Date:                Thu, 11 Jun 2020   Pseudo R-squ.:                  0.2533
Time:                        16:42:50   Log-Likelihood:                -326.52
converged:                      False   LL-Null:                       -437.29
Covariance Type:            nonrobust   LLR p-value:                 3.670e-25
==============================================================================
                 coef    std err          z      P>|z|      [0.025      0.975]
------------------------------------------------------------------------------
const         19.7128   1.95e+04      0.001      0.999   -3.82e+04    3.83e+04
age            0.0203      0.011      1.856      0.063      -0.001       0.042
amount     -9.472e-05   4.91e-05     -1.929      0.054      -0.000     1.5e-06
duration      -0.0246      0.011     -2.282      0.022      -0.046      -0.003
depends       -0.3219      0.305     -1.054      0.292      -0.921       0.277
foreign        1.4928      0.832      1.794      0.073      -0.138       3.124
telephon       0.2279      0.235      0.969      0.333      -0.233       0.689
checking1     -2.0323      0.276     -7.352      0.000      -2.574      -1.491
checking2     -1.4167      0.285     -4.979      0.000      -1.974      -0.859
checking3     -1.0766      0.425     -2.532      0.011      -1.910      -0.243
coapp1        -0.6176      0.462     -1.336      0.182      -1.524       0.289
coapp2        -1.3099      0.636     -2.060      0.039      -2.556      -0.064
employed1      0.2766      0.513      0.539      0.590      -0.728       1.282
employed2     -0.2890      0.357     -0.811      0.418      -0.988       0.410
employed3      0.1929      0.301      0.640      0.522      -0.398       0.784
employed4      0.4120      0.355      1.160      0.246      -0.284       1.108
existcr1     -15.3745   1.95e+04     -0.001      0.999   -3.83e+04    3.82e+04
existcr2     -15.7521   1.95e+04     -0.001      0.999   -3.83e+04    3.82e+04
existcr3     -15.9180   1.95e+04     -0.001      0.999   -3.83e+04    3.82e+04
history0      -1.8710      0.492     -3.803      0.000      -2.835      -0.907
history1      -1.6593      0.562     -2.951      0.003      -2.762      -0.557
history2      -1.0100      0.328     -3.081      0.002      -1.653      -0.367
history3      -0.8362      0.400     -2.088      0.037      -1.621      -0.051
housing1      -0.5449      0.568     -0.960      0.337      -1.658       0.568
housing2      -0.2042      0.544     -0.376      0.707      -1.270       0.861
installp1      0.8683      0.340      2.552      0.011       0.201       1.535
installp2      0.9388      0.276      3.396      0.001       0.397       1.481
installp3      0.5217      0.308      1.697      0.090      -0.081       1.124
job1          -0.7188      0.756     -0.951      0.342      -2.200       0.763
job2          -0.2276      0.410     -0.555      0.579      -1.031       0.576
job3          -0.3471      0.342     -1.016      0.310      -1.017       0.323
marital1      -0.5331      0.583     -0.914      0.361      -1.677       0.611
marital2      -0.5999      0.391     -1.535      0.125      -1.366       0.166
marital3      -0.0772      0.390     -0.198      0.843      -0.842       0.687
other1        -0.7518      0.293     -2.562      0.010      -1.327      -0.177
other2        -0.9835      0.444     -2.213      0.027      -1.855      -0.112
property1      0.3555      0.500      0.711      0.477      -0.625       1.336
```

```
property2      0.2710      0.489      0.554      0.579      -0.688       1.230
property3      0.3735      0.479      0.779      0.436      -0.566       1.313
resident1      0.4934      0.354      1.392      0.164      -0.201       1.188
resident2     -0.4395      0.266     -1.655      0.098      -0.960       0.081
resident3     -0.1930      0.318     -0.608      0.543      -0.815       0.429
savings1      -0.6255      0.296     -2.114      0.035      -1.206      -0.045
savings2      -0.4164      0.410     -1.017      0.309      -1.219       0.386
savings3      -0.4319      0.548     -0.789      0.430      -1.505       0.641
savings4       0.5030      0.750      0.671      0.502      -0.966       1.972
==============================================================================
****** Training Model Metrics *****

Model Metrics
Observations...............       700
Accuracy...................    0.7814
Precision..................    0.8083
Sensitivity (Recall).......    0.8912
Specificity (Selectivity)..    0.5450
F1-Score...................    0.8478
MISC (Misclassification)...     21.9%
     class 0...............     45.5%
     class 1...............     10.9%


     Confusion
       Matrix     Class 0   Class 1
Class 0.....       121       101
Class 1.....        52       426

******** Validation Metrics *******

Model Metrics
Observations...............       300
Accuracy...................    0.7633
Precision..................    0.8186
Sensitivity (Recall).......    0.8739
Specificity (Selectivity)..    0.4487
F1-Score...................    0.8453
MISC (Misclassification)...     23.7%
     class 0...............     55.1%
     class 1...............     12.6%


     Confusion
       Matrix     Class 0   Class 1
Class 0.....        35        43
Class 1.....        28       194
**********************************************


******************************************************************************
***************  Running StatsModel - Logistic with Stepwise   ***************
******************************************************************************
```

```
/Users/edwardrainwater/opt/anaconda3/envs/Stat656MacOSX/lib/python3.7/site-
packages/statsmodels/base/model.py:568: ConvergenceWarning: Maximum Likelihood
optimization failed to converge. Check mle_retvals
  "Check mle_retvals", ConvergenceWarning)
Add  checking1                        with p-value 1.54079e-15
Add  checking2                        with p-value 1.00836e-14
Add  duration                         with p-value 1.04097e-09
Add  savings1                         with p-value 0.000857473
Add  history1                         with p-value 0.000921171
Add  employed4                        with p-value 0.0028786
Add  savings2                         with p-value 0.00643012
Add  marital3                         with p-value 0.00758101
Add  history0                         with p-value 0.00807971
Add  history2                         with p-value 0.00770107
Add  property1                        with p-value 0.0230649
Add  foreign                          with p-value 0.0379187
Add  resident2                        with p-value 0.0280212
Add  housing2                         with p-value 0.0380405
Add  employed2                        with p-value 0.0557519
Add  other1                           with p-value 0.0620334
Add  checking3                        with p-value 0.0834477
Add  history3                         with p-value 0.0981458
Add  existcr1                         with p-value 0.0760241

Final selected attributes:
checking1
checking2
duration
savings1
history1
employed4
savings2
marital3
history0
history2
property1
foreign
resident2
housing2
employed2
other1
checking3
history3
existcr1
******************************************************************************
Optimization terminated successfully.
         Current function value: 0.494159
         Iterations 7

******************************************************************************
******************************* Training Model ********************************
                              Target: good_bad
                           Logit Regression Results
==============================================================================
```

```
Dep. Variable:               good_bad   No. Observations:                  700
Model:                          Logit   Df Residuals:                      680
Method:                           MLE   Df Model:                           19
Date:                Thu, 11 Jun 2020   Pseudo R-squ.:                  0.2090
Time:                        16:42:57   Log-Likelihood:                -345.91
converged:                       True   LL-Null:                       -437.29
Covariance Type:            nonrobust   LLR p-value:                 8.870e-29
==============================================================================
                 coef    std err          z      P>|z|      [0.025      0.975]
------------------------------------------------------------------------------
const          3.2929      0.426      7.737      0.000       2.459       4.127
checking1     -1.9458      0.259     -7.509      0.000      -2.454      -1.438
checking2     -1.1974      0.262     -4.562      0.000      -1.712      -0.683
duration      -0.0341      0.008     -4.200      0.000      -0.050      -0.018
savings1      -0.4201      0.241     -1.741      0.082      -0.893       0.053
history1      -1.8927      0.530     -3.569      0.000      -2.932      -0.853
employed4      0.2742      0.276      0.994      0.320      -0.266       0.815
savings2      -0.3846      0.357     -1.077      0.282      -1.084       0.315
marital3       0.2309      0.197      1.169      0.242      -0.156       0.618
history0      -1.7448      0.468     -3.730      0.000      -2.662      -0.828
history2      -0.9706      0.310     -3.129      0.002      -1.579      -0.363
property1      0.1400      0.220      0.637      0.524      -0.290       0.570
foreign        1.6184      0.809      2.001      0.045       0.033       3.203
resident2     -0.5057      0.210     -2.409      0.016      -0.917      -0.094
housing2       0.2714      0.211      1.288      0.198      -0.142       0.684
employed2     -0.5503      0.250     -2.200      0.028      -1.041      -0.060
other1        -0.5345      0.277     -1.932      0.053      -1.077       0.008
checking3     -0.8739      0.402     -2.175      0.030      -1.661      -0.087
history3      -0.6779      0.377     -1.797      0.072      -1.417       0.062
existcr1       0.4480      0.272      1.645      0.100      -0.086       0.982
==============================================================================

******************************************************************************
*************************** Training Model Metrics ***************************
********************************************************************************

Model Metrics
Observations...............        700
Accuracy...................     0.7586
Precision..................     0.7932
Sensitivity (Recall).......     0.8745
Specificity (Selectivity)..     0.5090
F1-Score...................     0.8318
MISC (Misclassification)...      24.1%
     class 0...............      49.1%
     class 1...............      12.6%


     Confusion
       Matrix     Class 0   Class 1
Class 0.....       113       109
Class 1.....        60       418

******************************************************************************
```

```
**************************** Validation Metrics ****************************
****************************************************************************

Model Metrics
Observations...............       300
Accuracy...................    0.7867
Precision..................    0.8238
Sensitivity (Recall).......    0.9054
Specificity (Selectivity)..    0.4487
F1-Score...................    0.8627
MISC (Misclassification)...     21.3%
     class 0...............     55.1%
     class 1...............      9.5%


     Confusion
       Matrix     Class 0   Class 1
Class 0.....        35        43
Class 1.....        21       201
****************************************************************************
*******Regularization Logistic Regression*****

Logistic Regression Model using C= 0.0001


Model Metrics..........       Training     Validation
Observations...........            700            300
Coefficients...........             46             46
DF Error...............            654            254
Iterations.............             99             99
Mean Absolute Error....         0.4161         0.3915
Avg Squared Error......         0.2064         0.1835
Accuracy...............         0.6771         0.7500
Precision..................     0.6886         0.7492
Recall (Sensitivity).......     0.9623         0.9955
F1-score...................     0.8028         0.8549
Total Misclassifications...        226             75
MISC (Misclassification)...      32.3%          25.0%
     class 0...............      93.7%          94.9%
     class 1...............       3.8%           0.5%


Training                 Class     Class
Confusion Matrix             0         1
Class 0..............       14       208
Class 1..............       18       460


Validation               Class     Class
Confusion Matrix             0         1
Class 0..............        4        74
Class 1..............        1       221

Logistic Regression Model using C= 0.01
```

Model Metrics..........	Training	Validation
Observations...........	700	300
Coefficients...........	46	46
DF Error...............	654	254
Iterations.............	50	50
Mean Absolute Error....	0.3977	0.3765
Avg Squared Error......	0.1926	0.1753
Accuracy...............	0.7114	0.7700
Precision..................	0.7184	0.7762
Recall (Sensitivity).......	0.9498	0.9685
F1-score...................	0.8180	0.8617
Total Misclassifications...	202	69
MISC (Misclassification)...	28.9%	23.0%
class 0...............	80.2%	79.5%
class 1...............	5.0%	3.2%

Training Confusion Matrix	Class 0	Class 1
Class 0..............	44	178
Class 1..............	24	454

Validation Confusion Matrix	Class 0	Class 1
Class 0..............	16	62
Class 1..............	7	215

Logistic Regression Model using C= 0.1

Model Metrics..........	Training	Validation
Observations...........	700	300
Coefficients...........	46	46
DF Error...............	654	254
Iterations.............	126	126
Mean Absolute Error....	0.3548	0.3396
Avg Squared Error......	0.1676	0.1587
Accuracy...............	0.7557	0.7900
Precision..................	0.7707	0.8118
Recall (Sensitivity).......	0.9142	0.9324
F1-score...................	0.8364	0.8679
Total Misclassifications...	171	63
MISC (Misclassification)...	24.4%	21.0%
class 0...............	58.6%	61.5%
class 1...............	8.6%	6.8%

Training Confusion Matrix	Class 0	Class 1
Class 0..............	92	130
Class 1..............	41	437

Validation Confusion Matrix	Class 0	Class 1
Class 0.............	30	48
Class 1.............	15	207

Logistic Regression Model using C= 1.0

Model Metrics..........	Training	Validation
Observations...........	700	300
Coefficients...........	46	46
DF Error...............	654	254
Iterations.............	134	134
Mean Absolute Error....	0.3282	0.3197
Avg Squared Error......	0.1614	0.1588
Accuracy...............	0.7571	0.7433
Precision..................	0.7906	0.8139
Recall (Sensitivity).......	0.8766	0.8468
F1-score...................	0.8313	0.8300
Total Misclassifications...	170	77
MISC (Misclassification)...	24.3%	25.7%
class 0...............	50.0%	55.1%
class 1...............	12.3%	15.3%

Training Confusion Matrix	Class 0	Class 1
Class 0.............	111	111
Class 1.............	59	419

Validation Confusion Matrix	Class 0	Class 1
Class 0.............	35	43
Class 1.............	34	188

Logistic Regression Model using C= 5.0

Model Metrics..........	Training	Validation
Observations...........	700	300
Coefficients...........	46	46
DF Error...............	654	254
Iterations.............	848	848
Mean Absolute Error....	0.3118	0.3135
Avg Squared Error......	0.1552	0.1649
Accuracy...............	0.7757	0.7533
Precision..................	0.8104	0.8162
Recall (Sensitivity).......	0.8766	0.8604
F1-score...................	0.8422	0.8377
Total Misclassifications...	157	74
MISC (Misclassification)...	22.4%	24.7%

class 0...............	44.1%	55.1%
class 1...............	12.3%	14.0%

Training Confusion Matrix	Class 0	Class 1
Class 0..............	124	98
Class 1..............	59	419

Validation Confusion Matrix	Class 0	Class 1
Class 0..............	35	43
Class 1..............	31	191

Logistic Regression Model using C= 10.0

Model Metrics..........	Training	Validation
Observations...........	700	300
Coefficients...........	46	46
DF Error...............	654	254
Iterations.............	154	154
Mean Absolute Error....	0.3232	0.3161
Avg Squared Error......	0.1613	0.1600
Accuracy...............	0.7571	0.7433
Precision..................	0.7928	0.8194
Recall (Sensitivity).......	0.8724	0.8378
F1-score...................	0.8307	0.8285
Total Misclassifications...	170	77
MISC (Misclassification)...	24.3%	25.7%
class 0...............	49.1%	52.6%
class 1...............	12.8%	16.2%

Training Confusion Matrix	Class 0	Class 1
Class 0..............	113	109
Class 1..............	61	417

Validation Confusion Matrix	Class 0	Class 1
Class 0..............	37	41
Class 1..............	36	186

Logistic Regression Model using C= 50.0

Model Metrics..........	Training	Validation
Observations...........	700	300
Coefficients...........	46	46
DF Error...............	654	254
Iterations.............	691	691

```
Mean Absolute Error....        0.3135         0.3150
Avg Squared Error......        0.1558         0.1649
Accuracy...............        0.7757         0.7600
Precision..................    0.8046         0.8178
Recall (Sensitivity).......    0.8870         0.8694
F1-score...................    0.8438         0.8428
Total Misclassifications...       157             72
MISC (Misclassification)...     22.4%          24.0%
     class 0...............     46.4%          55.1%
     class 1...............     11.3%          13.1%


Training                  Class     Class
Confusion Matrix              0         1
Class 0..............       119       103
Class 1..............        54       424


Validation                Class     Class
Confusion Matrix              0         1
Class 0..............        35        43
Class 1..............        29       193

Logistic Regression Model using C= inf


Model Metrics..........       Training     Validation
Observations...........            700            300
Coefficients...........             46             46
DF Error...............            654            254
Iterations.............            484            484
Mean Absolute Error....         0.3108         0.3122
Avg Squared Error......         0.1558         0.1650
Accuracy...............         0.7757         0.7667
Precision..................     0.8081         0.8248
Recall (Sensitivity).......     0.8808         0.8694
F1-score...................     0.8428         0.8465
Total Misclassifications...        157             70
MISC (Misclassification)...      22.4%          23.3%
     class 0...............      45.0%          52.6%
     class 1...............      11.9%          13.1%


Training                  Class     Class
Confusion Matrix              0         1
Class 0..............       122       100
Class 1..............        57       421


Validation                Class     Class
Confusion Matrix              0         1
Class 0..............        37        41
Class 1..............        29       193
```

```
** Cross-Validation for Regularization Logistic Regression **
0.000100..    0.8171    0.0144
0.010000..    0.8285    0.0177
0.100000..    0.8324    0.0341
1.000000..    0.8260    0.0274
5.000000..    0.8335    0.0288
10.000000.    0.8266    0.0278
50.000000.    0.8275    0.0271
inf.......    0.8315    0.0321
```

3 Appendix – Python Code Listing

```
"""
Created 09 JUN 2020

@author: el-rainwater, Rainwater Center for Neolithic Computing
"""

import pandas as pd
import numpy as np
from AdvancedAnalytics.ReplaceImputeEncode import ReplaceImputeEncode, DT
from AdvancedAnalytics.Regression import logreg, stepwise
import statsmodels.api as sm
from sklearn.model_selection import train_test_split, cross_val_score
from sklearn.linear_model import LogisticRegression

filepath = '/Users/edwardrainwater/OneDrive - Texas A&M University/' \
    'Summer-2020/STAT 656 Applied Analytics/hw-03/'
file = 'credithistory.xlsx'

def run_all_features(encoded_df,target):
    print("\n" + ("*"*78))
    print("*"*13 + "  Running StatsModel - All Features with Validation   " + "*"*13)
    print(("*"*78))
    y = encoded_df[target].astype(int)
    X = encoded_df.drop(target, axis=1)
    Xt, Xv, yt, yv = train_test_split(X, y, train_size=0.7, random_state=12345)
    Xtc     = sm.add_constant(Xt)
    Xvc     = sm.add_constant(Xv)
    model   = sm.Logit(yt, Xtc)
    results = model.fit()
    print('Printing results summary...')
    print(results.summary())

    print("****** Training Model Metrics *****")
    mat = results.pred_table(threshold=0.5)
    logreg.display_confusion(mat)

    print("\n******** Validation Metrics *******")
    predv  = results.predict(Xvc)
    sv     = np.where(predv<0.5, 0, 1)
    logreg.display_confusion(pd.crosstab(sv, yv))
    print("*********************************************\n")
```

```
    return()


def run_stepwise(df_encoded,target):
    print("\n" + ("*"*78))
    print("*"*15 + "  Running StatsModel - Logistic with Stepwise   " + "*"*15)
    print(("*"*78))

    # Set up stepwise feature selection
    df_encoded[target] = df_encoded[target].astype(int) # Do this to make target int
    y = df_encoded[target]
    sw = stepwise(df_encoded, target, reg='logistic', verbose=True)

    selected = sw.fit_transform()

    print('\nFinal selected attributes:')
    print(*selected,sep='\n')
    print(("*"*78))

    # Split the model 70/30 for training/validation

    X_train, X_validate, y_train, y_validate =  \
        train_test_split(df_encoded[selected], y, train_size=0.7,
                         random_state = 12345)

    Xc_train    = sm.add_constant(X_train)
    Xc_validate = sm.add_constant(X_validate)

    model    = sm.Logit(y_train, Xc_train)
    results  = model.fit()
    print("\n" + ("*"*78))
    print("*"*31 + " Training Model " + "*"*31)
    print("                              Target: " + target)
    print(results.summary())

    print("\n" + ("*"*78))
    print("*"*27 + " Training Model Metrics " + "*"*27)
    print(("*"*80))

    mat = results.pred_table(threshold=0.5)
    logreg.display_confusion(mat)

    print("\n" + ("*"*78))
    print("*"*29 + " Validation Metrics " + "*"*29)
    print(("*"*78))

    predv  = results.predict(Xc_validate)
    sv     = np.where(predv<0.5, 0, 1)
    logreg.display_confusion(pd.crosstab(sv, y_validate))
    print(("*"*78))

    return()

def run_reglr_logistic_regression(encoded_df, target):
```

```
    print("*******Regularization Logistic Regression*****")
    y = encoded_df[target].astype(int)
    X = encoded_df.drop(target, axis=1)
    X_train, X_validate, y_train, y_validate =  \
    train_test_split(X, y, train_size=0.7, random_state = 12345)

    C_list = [1e-4, 1e-2, 1e-1, 1.0, 5.0, 10.0, 50.0, np.inf]
    for c in C_list:
        lr = LogisticRegression(C=c, tol=1e-4, solver='lbfgs', max_iter=5000)
        lr = lr.fit(X_train, y_train)
        print("\nLogistic Regression Model using C=", c)
        logreg.display_split_metrics(lr, X_train, y_train, X_validate,
                                     y_validate, target_names=['Bad', 'Good'])

    print("\n** Cross-Validation for Regularization Logistic Regression **")
    for c in C_list:
        lr = LogisticRegression(C=c, tol=1e-4, solver='lbfgs', max_iter=5000)
        lrc = cross_val_score(lr, X, y, cv=10, scoring='f1', n_jobs=3)
        mean = lrc.mean()
        std  = lrc.std()
        print("{:.<10.6f}{:>10.4f}{:>10.4f}".format(c, mean, std))
    return()



def main(): ##########################################################

    df = pd.read_excel(filepath + file)
    print(df.head(), df.shape)
    print(df.dtypes)

    attribute_map = {
        'age':[DT.Interval, (19,120)],
        'amount':[DT.interval, (0,20000)],
        'checking':[DT.nominal, (list(range(1,5)))],
        'coapp':[DT.nominal, (1,2,3)],
        'depends':[DT.Binary, (1,2)],
        'duration':[DT.interval, (1,72)],
        'employed':[DT.Nominal, (list(range(1,6)))],
        'existcr':[DT.Nominal, (1,2,3,4)],
        'foreign':[DT.Binary, (1,2)],
        'history':[DT.nominal, (0,1,2,3,4)],
        'housing':[DT.Nominal, (1,2,3)],
        'installp':[DT.nominal, (1,2,3,4)],
        'job':[DT.nominal, (1,2,3,4)],
        'marital':[DT.Nominal, (1,2,3,4)],
        'other':[DT.Nominal, (1,2,3)],
        'property':[DT.nominal, (1,2,3,4)],
      'purpose':[DT.Ignore, ('0', '1', '2', '3', '4', '5', \
                            '6', '8', '9', 'X') ],
        'resident':[DT.Nominal, (1,2,3,4)],
        'savings':[DT.nominal,(1,2,3,4,5)],
        'telephon':[DT.Binary, (1,2)],
            'good_bad':[DT.Binary , ('bad', 'good') ]
```

```
        }

    target = 'good_bad'

    # One-hot encode and impute missing values
    rie = ReplaceImputeEncode(data_map=attribute_map, nominal_encoding='one-hot',
                              binary_encoding='one-hot', no_impute=[target],
                              interval_scale=None, drop=True,
                              display=True)
    df_encoded = rie.fit_transform(df).dropna() #drop rows with missing values

    run_all_features(df_encoded,target)
    run_stepwise(df_encoded,target)
    run_reglr_logistic_regression(df_encoded, target)



if __name__=='__main__':
    main()
```