SAS Code – Book 1

This SAS code uses the dataset that was reduced to only the variables found to be valuable through EDA.

\*Import Data - this is the dataset post-EDA modified variables;

**proc** **import** out=work.bankmodel

datafile="\\smu.edu\Files\users$\jeysenbach\Apps.SMU\Desktop\SAS\simplelogicEU3.csv"

DBMS= csv replace;

getnames=yes;

datarow=**2**;

\*Variables in this dataset: age job education default contact month campaign previous poutcome cons.price.idx cons.conf.idx Subscription;

\*\*\*\*\*\*\*\*\*Test and train set partitions\*\*\*\*\*\*\*\*\*\*;

Separate Yes and No observations into datasets for partitioning into a training and test set;

\*The goal is to get a training set that has approx equal yes and no responses for logistic regression analysis;

**data** bankmodel\_no;

set bankmodel;

if Subscription='no';

row = \_n\_;

**run**;

**data** bankmodel\_yes;

set bankmodel;

if Subscription='yes';

row = \_n\_;

**run**;

\*There are 36548 No and 4640 Yes.;

\*Below will randomly select about half of the yes observations and an approx equal number of no observations;

**data** banktrain\_no banktest\_no;

set bankmodel\_no;

by row;

if first.row then do;

if ranuni(**1234**) < **.065** then destination = 'banktrain\_no';

else destination = 'banktest\_no';

retain destination;

end;

if destination = 'banktrain\_no' then output banktrain\_no;

else output banktest\_no;

drop destination;

**run**;

**data** banktrain\_yes banktest\_yes;

set bankmodel\_yes;

by row;

if first.row then do;

if ranuni(**1234**) < **.5** then destination = 'banktrain\_yes';

else destination = 'banktest\_yes';

retain destination;

end;

if destination = 'banktrain\_yes' then output banktrain\_yes;

else output banktest\_yes;

drop destination;

**run**;

**data** banktrain;

set banktrain\_yes banktrain\_no;

**run**;

**data** banktest;

set banktest\_yes banktest\_no;

**run**;

\*Export test and train sets to csv;

**proc** **export** data=banktrain dbms=csv

outfile="\\smu.edu\Files\users$\jeysenbach\Apps.SMU\Desktop\SAS\banktrain.csv"

replace;

**run**;

**proc** **export** data=banktest dbms=csv

outfile="\\smu.edu\Files\users$\jeysenbach\Apps.SMU\Desktop\SAS\banktest.csv"

replace;

**run**;

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*Logistic Regression Models\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

\*Run 1: This will calculate predictions for both the training and test sets with the "score" option;

**proc** **logistic** data=banktrain plots=ROC;

class Subscription job education default contact month poutcome / param=ref;

model Subscription(event='yes') = job default education contact month campaign previous poutcome euribor3m cons\_price\_idx cons\_conf\_idx / scale=none aggregate lackfit ctable outroc=trainroc;

output out=trainpreds;

score data=banktest out=testpreds outroc=testroc fitstat;

**run**;

\*Run 2: education and "previous" were slightly insignificant so compare models with and without them;

**proc** **logistic** data=banktrain plots=ROC;

class Subscription job default contact month poutcome / param=ref;

model Subscription(event='yes') = job default contact month campaign poutcome euribor3m cons\_price\_idx cons\_conf\_idx / scale=none aggregate lackfit ctable outroc=trainroc;

output out=trainpreds;

score data=banktest out=testpreds outroc=testroc fitstat;

**run**;

\*frequency table of test set results;

**proc** **freq** data=testpreds;

tables I\_Subscription\*Subscription;

**run**;

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*Options for adjusting prior probability when running on test set\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

\*Run 3: set prior prob to .2;

**proc** **logistic** data=banktrain plots=ROC;

class Subscription job default contact month poutcome / param=ref;

model Subscription(event='yes') = job default contact month campaign poutcome euribor3m cons\_price\_idx cons\_conf\_idx / scale=none aggregate lackfit ctable outroc=trainroc;

output out=trainpreds;

score data=banktest out=testpreds outroc=testroc fitstat priorevent=**.2**;

**run**;

**proc** **freq** data=testpreds;

tables I\_Subscription\*Subscription;

**run**;

\*Run 3: set prior prob to .65;

**proc** **logistic** data=banktrain plots=ROC;

class Subscription job default contact month poutcome / param=ref;

model Subscription(event='yes') = job default contact month campaign poutcome euribor3m cons\_price\_idx cons\_conf\_idx / scale=none aggregate lackfit ctable outroc=trainroc;

output out=trainpreds;

score data=banktest out=testpreds outroc=testroc fitstat priorevent=**.65**;

**run**;

**proc** **freq** data=testpreds;

tables I\_Subscription\*Subscription;

**run**;

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

\*Run4 - Back to original model - Run ROC Curve comparisons omtting most influential variables;

**proc** **logistic** data=banktrain plots=ROC;

class Subscription job default contact month poutcome / param=ref;

model Subscription(event='yes') = job default contact month campaign poutcome euribor3m cons\_price\_idx cons\_conf\_idx / scale=none aggregate lackfit ctable outroc=trainroc;

output out=trainpreds;

score data=banktest out=testpreds outroc=testroc fitstat;

roc 'Excluding Euribor3m' job default contact month campaign poutcome cons\_price\_idx cons\_conf\_idx;

roc 'Excluding Euribor3m and poutcome' job default contact month campaign cons\_price\_idx cons\_conf\_idx;

roc 'Only Euribor3m and poutcome' poutcome euribor3m;

roccontrast reference('Excluding Euribor3m and poutcome') /estimate;

**run**;

\*FOR OBJECTIVE 2;

\*Add interaction terms and try selection methods - ONLY with variables from orignal model;

\*This was the last manually iterated code after forward and backward selection - not much improvement;

**proc** **logistic** data=banktrain plots=ROC;

class Subscription job education default contact month poutcome / param=ref;

model Subscription(event='yes') = job default contact campaign poutcome euribor3m month euribor3m\*month / scale=none aggregate lackfit ctable outroc=trainroc;

output out=trainpreds;

score data=banktest out=testpreds outroc=testroc fitstat;

**run**;

**proc** **logistic** data=banktrain plots=ROC;

class Subscription job education default contact month poutcome / param=ref;

model Subscription(event='yes') = job default contact campaign poutcome euribor3m month euribor3m\*month / scale=none aggregate lackfit ctable outroc=trainroc;

output out=trainpreds;

score data=banktest out=testpreds outroc=testroc fitstat;

**run**;

**proc** **freq** data=testpreds;

tables I\_Subscription\*Subscription;

**run**;