SAS Code – Book 2

This SAS code uses the original dataset with all of the variables included.

\*Import Data - this is the raw dataset with no variable modifications;

**proc** **import** out=work.bankraw

datafile="\\smu.edu\Files\users$\jeysenbach\Apps.SMU\Desktop\SAS\bank-additional-full.csv"

DBMS= csv replace;

getnames=yes;

datarow=**2**;

\*Rename "y" to "Subscription";

**data** bankraw;

set bankraw;

rename y = Subscription;

**run**;

\*\*\*\*\*\*\*\*\*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** bankraw\_no;

set bankraw;

if Subscription='no';

row = \_n\_;

**run**;

**data** bankraw\_yes;

set bankraw;

if Subscription='ye';

row = \_n\_;

**run**;

\*There are 36548 No and 4640 Yes.;

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

**data** rbanktrain\_no rbanktest\_no;

set bankraw\_no;

by row;

if first.row then do;

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

else destination = 'rbanktest\_no';

retain destination;

end;

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

else output rbanktest\_no;

drop destination;

**run**;

**data** rbanktrain\_yes rbanktest\_yes;

set bankraw\_yes;

by row;

if first.row then do;

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

else destination = 'rbanktest\_yes';

retain destination;

end;

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

else output rbanktest\_yes;

drop destination;

**run**;

**data** banktrain\_raw;

set rbanktrain\_yes rbanktrain\_no;

**run**;

**data** banktest\_raw;

set rbanktest\_yes rbanktest\_no;

**run**;

\*Export test and train sets to csv;

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

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

replace;

**run**;

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

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

replace;

**run**;

/\*\*\*\*\*\*\*\*\*\*\*\*\*Logisitic Regression Models\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

\*FOR OBJECTIVE 1 - comparing to EDA model using LASSO on all variables;

\*Using LASSO on the training set for variable selection;

**proc** **hpgenselect** data=banktrain\_raw;

class Subscription marital housing loan contact day\_of\_week job education default month poutcome;

model Subscription(event='ye') = marital previous contact campaign job education default day\_of\_week housing loan month poutcome age cons\_price\_idx cons\_conf\_idx euribor3m emp\_var\_rate/ dist=binary;

selection method=lasso details=all;

**run**;

\*Lasso selected "previous contact campaign default month poutcome age cons\_price\_idx cons\_conf\_idx euribor3m";

\*Run 2: Run with variables selected by LASSO;

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

class Subscription contact default month poutcome / param=ref;

model Subscription(event='ye') = previous contact campaign default month poutcome age cons\_price\_idx cons\_conf\_idx euribor3m / scale=none aggregate lackfit outroc=rtrainroc;

output out=rtrainpreds;

score data=banktest\_raw out=rtestpreds outroc=rtestroc fitstat;

**run**;

\*Run 3: Remove insig variables and run again;

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

class Subscription contact default month poutcome / param=ref;

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

output out=rtrainpreds;

score data=banktest\_raw out=rtestpreds outroc=rtestroc fitstat;

**run**;

**proc** **freq** data=rtestpreds;

tables I\_Subscription\*Subscription;

**run**;

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

\*FOR OBJECTIVE 2 - adding 2-way interaction terms and running LASSO selection again;

**proc** **hpgenselect** data=banktrain\_raw;

class Subscription contact default month poutcome;

model Subscription(event='ye') = previous | contact | campaign | default | month | poutcome | age | cons\_price\_idx | cons\_conf\_idx | euribor3m @**2** / dist=binary;

selection method=lasso details=all;

**run**;

\*Lasso selected: previous\*cons\_price\_idx cons\_price\_idx\*contact campaign\*cons\_price\_idx cons\_price\_idx\*default cons\_price\_idx\*month cons\_price\_idx\*poutcome age\*cons\_price\_idx age\*cons\_conf\_idx cons\_price\_idx\*cons\_conf\_idx age\*euribor3m cons\_price\_idx\*euribor3m;

\*Run proc logistic with LASSO or manually selected variables;

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

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

model Subscription(event='ye') = job contact campaign default month poutcome cons\_price\_idx cons\_conf\_idx euribor3m\*month job\*age month\*campaign/ scale=none aggregate lackfit outroc=rtrainroc;

output out=rtrainpreds;

score data=banktest\_raw out=rtestpreds outroc=rtestroc fitstat;

**run**;

**proc** **freq** data=rtestpreds;

tables I\_Subscription\*Subscription;

**run**;

\*Iterating through LASSO selected variables - no improvement;

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

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

model Subscription(event='ye') = job contact month cons\_price\_idx cons\_conf\_idx euribor3m cons\_price\_idx\*contact cons\_price\_idx\*month cons\_price\_idx\*cons\_conf\_idx cons\_price\_idx\*euribor3m / scale=none aggregate lackfit outroc=rtrainroc;

output out=rtrainpreds;

score data=banktest\_raw out=rtestpreds outroc=rtestroc fitstat;

**run**;

**proc** **freq** data=rtestpreds;

tables I\_Subscription\*Subscription;

**run**;