%web\_drop\_table(WORK.IMPORT);

FILENAME REFFILE '/folders/myfolders/Chapter 3.4.xlsx';

PROC IMPORT DATAFILE=REFFILE

DBMS=XLSX

OUT=WORK.IMPORT;

GETNAMES=YES;

RUN;

PROC CONTENTS DATA=WORK.IMPORT; RUN;

%web\_open\_table(WORK.IMPORT);

data cleanup;

set import;

date\_dt=substr(date,1,2);

date\_m=substr(date,4,2);

date\_y=substr(date,7,4);

format date\_1 date9.;

date\_1=mdy(date\_m, date\_dt, date\_y);

dflt\_dt=substr(default\_date,1,2);

dflt\_m=substr(default\_date,4,2);

dflt\_y=substr(default\_date,7,4);

format dflt\_dt\_1 date9.;

dflt\_dt\_1=mdy(dflt\_m,dflt\_dt,dflt\_y);

cure\_dt=substr(cure\_date,1,2);

cure\_m=substr(cure\_date,4,2);

cure\_y=substr(cure\_date,7,4);

format cure\_dt\_1 date9.;

cure\_dt\_1=mdy(cure\_m,cure\_dt,cure\_y);

re\_dflt\_dt=substr(re\_default\_date,1,2);

re\_dflt\_m=substr(re\_default\_date,4,2);

re\_dflt\_y=substr(re\_default\_date,7,4);

format re\_dflt\_dt\_1 date9.;

re\_dflt\_dt\_1=mdy(re\_dflt\_m,re\_dflt\_dt,re\_dflt\_y);

if default\_date ne "." then do;

dflt=1;

end;

else do;

dflt=0;

end;

run;

data cleaned (drop=dflt\_dt\_1 cure\_dt\_1 re\_dflt\_dt\_1 date\_1);

set cleanup (drop=default\_date cure\_date re\_default\_date dflt\_dt dflt\_m dflt\_y cure\_dt cure\_m cure\_y re\_dflt\_dt re\_dflt\_m re\_dflt\_y

date date\_dt date\_m date\_y);

format default\_date date9.;

default\_date=dflt\_dt\_1;

format cure\_date date9.;

cure\_date=cure\_dt\_1;

format re\_default\_dt date9.;

re\_default\_dt=re\_dflt\_dt\_1;

format date date9.;

date=date\_1;

run;

/\*1. Selecting the last customer record for modelling\*/

proc sort data=cleaned;

by customer date;

run;

data model\_latest\_record temp;

set cleaned;

by customer date;

if last.customer and last.date then output model\_latest\_record;

else output temp;

run;

proc genmod data=model\_latest\_record descending;

Class collateral\_type customer\_type;

Model dflt = utilisation ltv collateral\_type borrowing\_portfolio\_ratio postcode\_index customer\_type arrears/ dist=binomial;

Output out=preds p=pred l=lower u=upper;

Run;

Proc sql;

select count(customer) from cleaned where default\_date ne .; quit;

/\*for example\*/

proc logistic data=preds descending;

model dflt = / nofit;

roc "Genmod model" pred=pred;

run;

/\*2. Selecting the defaulted record information and last record for non-defaulted customers\*/

data defaulted non\_defaulted;

set cleaned;

if default\_date ne . then output defaulted;

else output non\_defaulted;

run;

proc sql;

create table non\_defaulted\_filter as/\*remove the customers who are part of the defaulted dataset\*/

select \*

from non\_defaulted where customer not in (select customer from defaulted)

order by customer, date

;

quit;

data non\_defaulted\_select temp1;/\*select the last record of the non-defaulted\*/

set non\_defaulted\_filter;

by customer date;

if last.customer and last.date then output non\_defaulted\_select;

else output temp1;

run;

proc sql;/\*find relationship length at customer level\*/

create table vars as

select customer, count(distinct(year)) as relationship\_length, max(arrears) as arrears\_flag

from cleaned

group by 1

;

quit;

proc sql;

create table model\_rerun\_selection as

select distinct a.\*, b.relationship\_length, b.arrears\_flag

from

(select \* from defaulted union

select \* from non\_defaulted\_select) as a left join vars as b

on a.customer=b.customer

order by dflt

;

quit;

proc surveyselect data=defaulted method=srs n=5 out=dflt\_validation;

run;

proc surveyselect data=non\_defaulted\_select method=srs n=5 out=non\_dflt\_validation;

run;

proc sql;

create table validation as

select \* from dflt\_validation union

select \* from non\_dflt\_validation

;

quit;

proc sql;

create table model\_validation as

select distinct a.\*, case when b.customer eq . then 1 else 0 end as validation\_sample

from model\_rerun\_selection as a left join validation as b

on a.customer=b.customer

order by customer

;

quit;

proc genmod data=model\_validation descending;

Weight validation\_sample;

Model dflt = utilisation ltv borrowing\_portfolio\_ratio postcode\_index arrears\_flag relationship\_length/ dist=binomial;

Output out=preds(where=(validation\_sample=0)) p=pred l=lower u=upper;

Run;

Proc sort data=model\_validation;

By dflt;

Run;

Proc Means Data=model\_validation;

By dflt;

Vars utilisation ltv borrowing\_portfolio\_ratio postcode\_index arrears\_flag relationship\_length;

Run;

/\*Record the data in the Validation Dataset as everytime surveyselect is run randomly the training and validation dataset will change\*/

proc sql;

Title "Customers in validaiton dflt selection";

select distinct customer from dflt\_validation;

Title "Customers in validaiton non dflt selection";

select distinct customer from non\_dflt\_validation;

quit;

proc sql;

Title "Prediction of validation dataset";

select customer, dflt as observed\_default\_status, pred as pred\_value format 8.5

from preds

;

quit;

proc logistic data=preds descending;

model dflt = / nofit;

roc "Genmod model" pred=pred;

run;

/\*3. Logistic\*/

Data validation\_customer;

Input Customer;

Datalines;

/\*Dflt\*/

3342349

6161840

8697888

12095275

35234232

/\*Live\*/

4232324

7567563

10870633

11228652

24123211

;

run;

proc sql;

create table logistic as

select \*

from model\_rerun\_selection

where customer not in (select customer from validation\_customer)

;

quit;

Proc logistic data=logistic;

Model dflt=utilisation ltv borrowing\_portfolio\_ratio postcode\_index arrears\_flag

relationship\_length;

Run;

/\*4 Proc Genmod Probit\*/

proc sql;

create table overall as

select distinct a.\*, case when b.customer ne . then 0 else 1 end as validation\_sample

from model\_rerun\_selection as a left join validation\_customer as b

on a.customer=b.customer

;

quit;

proc genmod data=overall descending;

Weight validation\_sample;

model dflt = utilisation borrowing\_portfolio\_ratio postcode\_index arrears\_flag

/ dist=binomial link=probit;

output out=preds(where=(validation\_sample=0)) p=pred;

ods output parameterestimates=parms;

run;

proc print data=parms noobs;

format estimate 12.10;

var parameter level: estimate;

run;

Proc logistic data=preds descending;

Model dflt= / nofit;

roc "Genmod model" pred=pred;

Run;

Proc corr data=overall;

Var utilisation ltv borrowing\_portfolio\_ratio postcode\_index arrears\_flag relationship\_length;

Run;