Code with Two New Features (Total_Income and EMI)

```
data train;

infile '/home/ermindedic0/train.txt' DSD MISSOVER FIRSTOBS=2;

input Loan_ID$ Gender$ Married$ Dependents$ Education$ Self_Employed$ Applicant Co_Applicant
Loan_Amount L_Amount_Term Credit_History Property$ Loan_Status$;

run;


data test;

infile '/home/ermindedic0/test.txt' DSD MISSOVER FIRSTOBS=2;

input Loan_ID$ Gender$ Married$ Dependents$ Education$ Self_Employed$ Applicant Co_Applicant
Loan_Amount L_Amount_Term Credit_History Property$;

run;


proc contents data=train;

run;


proc print data=train (obs=20);

run;


proc gchart data=train;

vbar Gender;

run;


proc univariate data=train;

var Applicant;

histogram/normal;

run;
```

```
proc boxplot data=train;
plot Applicant*Education;
run;

proc gchart data=train;
vbar Married/subgroup=Loan_Status type=percent;
run;

proc gchart data=train;
vbar Applicant/subgroup=Loan_Status;
run;

data appandco;
set train;
Total_Income=Applicant + Co_Applicant;
run;

proc datasets;
modify train;
run;

proc format;
value incomegrp
    0 -< 2750 = 'Low'
 2750 -< 4000 = 'Average'
 4000 -< 6000 = 'High'
6000 -  high = 'Very High'
;
run;
```

```
proc gchart data=train;

format Applicant incomegrp.;

  vbar Applicant /

  coutline=black

  subgroup=Loan_Status

 legend=legend1

  type=freq

  width=8

  maxis=axis1

  raxis=axis2

  discrete;

run;

proc format;
value incomegrp
   0 -< 1000 = 'Low'
```

```
 1000 -< 3000 = 'Average'
 3000 -< 42000 = 'High'
;
run;


proc gchart data=train;

format Co_Applicant incomegrp.;

  vbar Co_Applicant /

  coutline=black

  subgroup=Loan_Status

 legend=legend1

  type=freq

  width=8

  maxis=axis1

  raxis=axis2

  discrete;

run;
```

```
/*Combining Applicant and Co-Applicant Income*/
proc format;
value incomegrp
   0 -< 2750 = 'Low'
 2750 -< 4000 = 'Average'
 4000 -< 6000 = 'High'
6000 -  high = 'Very High'
;
run;


legend1 label=('Loan_Status') frame;

axis2 label=('Percentage' angle=90) order=(0 to .25 by .05) value=('0.0' '0.2' '0.4' '0.6' '0.8' '1.0');

axis1 label=('Applicant');



pattern1 V=msolid  C=lightblue;

pattern2 V=msolid  C=orange;



proc gchart data=appandco;
```

```
format Total_Income incomegrp.;

  vbar Total_Income /

  coutline=black

  subgroup=Loan_Status

  legend=legend1

  type=percent

  width=8

  maxis=axis1

  raxis=axis2

  outside=pct

  inside=freq

  discrete;

run;

proc format;
value incomegrp
   0 -< 101 = 'Low'
```

```
 101 -< 201 = 'Average'
 201 - high = 'High'
;
run;


proc gchart data=train;

format Loan_Amount incomegrp.;

  vbar Loan_Amount/

  coutline=black

  subgroup=Loan_Status

 legend=legend1

  type=freq

  width=8

  maxis=axis1

  raxis=axis2

  discrete;

run;
```

```
/* Turning Loan_Status into binary, and getting rid of characters in the Dependents variable*/
data traint;
set train (rename=(Loan_Status=L_Status));
if L_Status = 'N' then L_Status = 0; else L_Status = 1;
drop Loan_Status;
Lo_Status = input(L_Status, 8.);
drop L_Status;
N_Dependents = compress(dependents, '+');
drop Dependents;
Loan_Amount=log(Loan_Amount);
drop Loan_ID;
Total_Income=Applicant + Co_Applicant;
EMI=Loan_Amount/L_Amount_Term;
Total_Income=log(Total_Income);
drop Applicant Co_Applicant Loan_Amount L_Amount_Term;
run;

/* Partition data*/
data traind valid;
set traint;
if ranuni(7) <=.6 then output traind; else output valid;
run;


/*No imputation, just considering the missing data patterns*/
proc mi data=traind nimpute=0;
class Gender Married N_Dependents Self_Employed Property Education;
```

```
fcs;
var Credit_History Gender Married N_Dependents Self_Employed Property Education;
run;



/*Imputation of our missing values for traind*/
proc mi nimpute=9 data=traind seed=55 out=traindd;
 class Gender Married N_Dependents Self_Employed Property Education;
 fcs plots=trace nbiter=20 logistic (N_Dependents)
   discrim(Gender Married Self_Employed
   /classeffects=include) regression (Total_Income EMI);
 var Credit_History N_Dependents Self_Employed Gender Married Property Education Total_Income
EMI;
run;

/* Fitting logistic model with ROC and constrast of different ROC Curves*/
ods graphics on;
proc logistic data=traindd outmodel=model1 plots(only)=roc(ID=Cutpoint);
class Gender Married Self_Employed Property Education N_Dependents (ref='1')/ param = ref;
model Lo_Status(event='1') = Credit_History N_Dependents Self_Employed Gender Married Property
Education Total_Income EMI;
by _imputation_;
ods output ParameterEstimates=train_parms;
run;
ods graphics off;

/*Combining parameter estimates*/
proc mianalyze parms=train_parms;
class Gender Married N_Dependents Self_Employed Property Education;
```

```
MODELEFFECTS Intercept Credit_History Total_Income EMI Gender Married N_Dependents
Self_Employed Property Education;

STDERR;

run;


/*Imputation of our missing values for valid*/

proc mi nimpute=9 data=valid seed=55 out=valida;

 class Gender Married N_Dependents Self_Employed Property Education;

 fcs plots=trace nbiter=20 logistic (N_Dependents)

   discrim(Gender Married Self_Employed

   /classeffects=include) regression (Total_Income EMI);

 var Credit_History N_Dependents Self_Employed Gender Married Property Education Total_Income
EMI;

run;


proc logistic inmodel=model1;

score data=valida out=validacc fitstat;

by _imputation_;

run;


/* Use PROC LOGISTIC and output the predicted probabilities.

 */

proc logistic data=validacc noprint;

class Gender Married Property Education;

   model Lo_Status(event='1') = Gender Credit_History Married Property Education Total_Income EMI;

   output out=validout predicted=predprob; /* save predicted probabilities in data set */

run;


/* To construct the decile calibration plot, identify deciles of the predicted prob. */
```

```
proc rank data=validout out=validdecile groups=10;
  var predprob;
  ranks Decile;
run;

/* Then compute the mean predicted prob and the empirical proportions (and CI) for each decile */
proc means data=validdecile noprint;
  class Decile;
  types Decile;
  var Lo_Status predprob;
  output out=LogiDecileOut mean=yMean PredProbMean
       lclm=yLower uclm=yUpper;
run;

title "Calibration Plot";
proc sgplot data=LogiDecileOut noautolegend aspect=1;
  lineparm x=0 y=0 slope=1 / lineattrs=(color=grey pattern=dash);

  series x=PredProbMean y=yMean;  /* if you to connect the deciles */
  scatter x=PredProbMean y=yMean / yerrorlower=yLower yerrorupper=yUpper;
  yaxis label="Observed Probability of Outcome";
  xaxis label="Predicted Probability of Outcome";
run;
```