```
PROC IMPORT DATAFILE='/home/pareshbits90/Credit.csv'
    DBMS=CSV
    OUT=Data1
    replace;
    GETNAMES=YES;
RUN;

Data data2;
Set data1;
drop Var12;
if numberofdependents ne "Go" and numberofdependents ne "Ba";
run;

**Checking the Frequency of NPA_Status in a sample with dependent values missing and in a complete sample;
Proc sort data=data2;
by Numberofdependents;
run;

Proc Freq data=data2;
tables NPA_status;
where NumberofDependents ne "NA";
run;

Proc Freq data=data2;
tables NPA_status;
run;

*As the frequency distribution is very similar in both results we can ignore missing dependents values as it
will not create any bias in our training or validation dataset;

**Checking the Frequency of NPA_Status in a sample with income values missing and in a complete sample;
Proc sort data=data2;
by MonthlyIncome;
run;

Proc Freq data=data2;
tables NPA_status;
where MonthlyIncome ne "NA";
run;

Proc Freq data=data2;
tables NPA_status;
run;

*As the frequency distribution is very similar in both results we can ignore missing income values as it
will not create any bias in our training or validation dataset;

*Delete records with NA values";
Data data2;
Set data2;
if NumberofDependents ne "NA" and MonthlyIncome ne "NA";
run;


*Some more data preparation;
data logprep;
set data2;
rename revolvingutilizationofunsecuredl=credit_utilization;
rename Rented_OwnHouse=house;
rename NumberOfTime30_59DaysPastDueNotW=payment_history1;
rename NumberOfTimes90DaysLate=payment_history3;
rename NumberRealEstateLoansOrLines=No_of_RELoans;
rename NumberOfTime60_89DaysPastDueNotW=payment_history2;
rename NumberOfDependents=dependents;
rename NumberOfOpenCreditLinesAndLoans=Open_Credit_Lines;
run;


*Coverting character variables to numeric variables;
data Log_Prep;
set logprep;
Monthly_Income=Input(MonthlyIncome, best12.);
Dependent=Input(dependents,best3.);
drop MonthlyIncome dependents;
run;
```

```
*Creating dummy/indicator Variables;
Data Log_Prep;
set Log_Prep;

if house="Ownhouse" then house_1=1; else house_1=0;
if house="rented" then house_2=1; else house_2=0;

if occupation="Non-offi" then occupation_1=1; else occupation_1=0;
if occupation="Officer1" then occupation_2=1; else occupation_2=0;
if occupation="Officer2" then occupation_3=1; else occupation_3=0;
if occupation="Officer3" then occupation_4=1; else occupation_4=0;
if occupation="Self_Emp" then occupation_5=1; else occupation_5=0;

if Education="Graduate" then education_1=1; else education_1=0;
if Education="Professional" then education_2=1; else education_2=0;
if Education="PhD" then education_3=1; else education_3=0;
if Education="Post-Grad" then education_4=1; else education_4=0;
if Education="Matric" then education_5=1; else education_5=0;

if Gender="Male" then gender_1=1; else gender_1=0;
if Gender="Female" then gender_2=1; else gender_2=0;

if Region="West" then region_1=1; else region_1=0;
if Region="East" then region_2=1; else region_2=0;
if Region="North" then region_3=1; else region_3=0;
if Region="South" then region_4=1; else region_4=0;
if Region="Central" then region_5=1; else region_5=0;

Run;



*delete outliers in credit_untilization, debtratio, age and payment history, and number of loans;
data log_prep;
set log_prep;
if credit_utilization le 1;
if debtratio le 10;
if age>20;
if payment_history2 le 12;
if no_of_RELoans le 15;
if Open_Credit_Lines le 38;
run;


*creating development and validation dataset;
Data Development_Sample Validation_Sample;
Set log_prep;
if ranuni(100)<0.7 then output Development_Sample;
else output Validation_Sample;
run;


*First Logistic Iteration;
Proc logistic data=development_sample descending;
model NPA_Status= credit_utilization age gender_1 region_1 region_2  region_3 region_4 Monthly_Income education_1
education_2 education_3 education_4 house_1 occupation_1 occupation_2 occupation_3 occupation_4 payment_history1
payment_history2 payment_history3 DebtRatio Open_Credit_Lines No_of_RELoans dependent ;
run;

*Second Logistic Iteration;
Proc logistic data=development_sample descending outmodel=dmm outest=RegOut;
model NPA_Status= credit_utilization age gender_1 region_1 region_2  region_3 region_4 Monthly_Income education_1
education_2 education_3 education_4 house_1 occupation_1 occupation_3 payment_history1
payment_history2 payment_history3 DebtRatio Open_Credit_Lines No_of_RELoans dependent ;
score out=dmp;
run;

proc Score data=development_sample score=RegOut out=Outdata type=parms;
var credit_utilization age gender_1 region_1 region_2  region_3 region_4 Monthly_Income education_1
education_2 education_3 education_4 house_1 occupation_1 occupation_3 payment_history1
payment_history2 payment_history3 DebtRatio Open_Credit_Lines No_of_RELoans dependent ;
run;

data Outdata;
set Outdata;
P_1=exp(NPA_Status2)/(1+exp(NPA_Status2));
run;
```

```
**Regression on Validation Dataset;
Proc logistic data=validation_sample descending outmodel=dmm;
model NPA_Status= credit_utilization age gender_1 region_1 region_2  region_3 region_4 Monthly_Income education_1
education_2 education_3 education_4 house_1 occupation_1 occupation_3 payment_history1
payment_history2 payment_history3 DebtRatio Open_Credit_Lines No_of_RELoans dependent ;
score out=dmp;
run;


*Code for generating lift chart;
Proc Sort data=dmp;
by P_1;
run;

proc rank data=dmp groups=10 ties=mean out=Lift_Curve;
Var P_1;
Ranks decile;
run;

Proc export data=lift_curve
outfile='/home/pareshbits90/lift_curve.csv'
dbms=csv replace;
```