## Predicting the probability of the malignant breast cancer

/\* Create your own library in SAS like here it is libref and mention the path \*/

libname libref “/home/aro1260/deep”;

/\* Importing Synthetic\_cancer\_data \*/

PROC IMPORT DATAFILE= “/home/aroragaurav1260/data/synthetic\_cancer\_data.csv”

DBMS=CSV Replace

OUT=libref.cancer;

GETNAMES=YES;

RUN;

/\* To check the contents of the data \*/

PROC CONTENTS DATA=libref.cancer;

RUN;

/\* Applying Proc freq to see the frequency of the data \*/

proc freq data = libref.cancer ;

tables Bare\_Nuclei Outcome Outcome \* Bare\_Nuclei;

run;

## /\*Dropping the Sample\_No from the data\*/

data libref.final\_cancer;

set libref.cancer(drop= Sample\_No);

RUN;

PROC CONTENTS DATA=libref.final\_cancer;

RUN;

## Model Building & Interpretation on Full Data

/\*Building Random Forests model on full data \*/

**Program2:**

Proc hpforest data = libref.final\_cancer maxtrees=1500 vars\_to\_try=3 ;

target Outcome/level=binary;

input Thickness\_of\_Clump Cell\_Size\_Uniformity Cell\_Shape\_Uniformity Marginal\_Adhesion Single\_Epithelial\_Cell\_Size Bare\_Nuclei Bland\_Chromatin

Normal\_Nucleoli Mitoses/level=nominal;

ods output fitstatistics = libref.fitstats\_out;

save file = "/home/aroragaurav1260/data/Random\_model\_fit.bin";

RUN;

/\* PLOTTING MISCLASSIFICATION RATE FOR TRAINING DATA \*/

proc sgplot data=libref.fitstats\_out;

title "Misclassification Rate for Training Data";

series x=Ntrees y=MiscALL;

yaxis label='OOB Misclassification Rate';

Run;

/\* Plot of OOB versus Training Misclassification Rate \*/

proc sgplot data=libref.fitstats\_out;

title "OOB vs Training";

series x=Ntrees y=MiscAll;

series x=Ntrees y=MiscOob/lineattrs=(pattern=shortdash thickness=2);

yaxis label='Misclassification Rate';

Run;

/\*splitting data set into train and valid dataset in 70:30\*/

proc surveyselect data= libref.final\_cancer method=srs seed=2 outall samprate=0.7 out=libref.cancer\_subset;

/\*Values of selected variable: 1 means for train set, 0 means test set\*/

data libref.train;

set libref.cancer\_subset;

if selected=1;

data libref.valid;

set libref.cancer\_subset;

if selected=0;

### Model Building & Interpretation on Training and Testing Data

data libref.train\_valid;

set libref.cancer\_subset;

Run;

**Program2.1**

/\*Building Random Forests model using train data\*/

proc hpforest data=libref.train\_valid

maxtrees=1500 vars\_to\_try=3;

target Outcome/level=binary;

input Thickness\_of\_Clump

Cell\_Size\_Uniformity Cell\_Shape\_Uniformity

Marginal\_Adhesion Single\_Epithelial\_Cell\_Size

Bare\_Nuclei Bland\_Chromatin

Normal\_Nucleoli Mitoses/level=nominal;

partition var= Selected (train = 1,valid = 0);

ods output VariableImportance= libref.loss\_reduction\_importance;

save file="/home/aroragaurav1260/data/Random\_forest\_fit.bin";

Run;

/\* Predicting the model using valid data\*/

**Program2.1**

proc hp4score data= libref.valid;

ods output VariableImportance=libref.rba\_importance\_valid;

performance threads=1;

importance file=

"/home/aroragaurav1260/data/Random\_forest\_fit.bin"

out=libref.scored

var=(Outcome Thickness\_of\_Clump Cell\_Size\_Uniformity Cell\_Shape\_Uniformity Marginal\_Adhesion Single\_Epithelial\_Cell\_Size Bare\_Nuclei Bland\_Chromatin

Normal\_Nucleoli Mitoses);

Run;

/\*Sorting the rba\_importance\_valid by Margin /\*

proc sort data = libref.rba\_importance\_valid;

by descending Margin;

Run;

proc print data= libref.rba\_importance\_valid;

Run;

/\*computing misclassification rate \*/

data libref.final\_score;

set libref.scored ;

if upcase(Outcome) ne upcase(I\_Outcome) then misclass=1;

else misclass=0;

run;

proc means data=libref.final\_score(where=(Outcome ne ''));

var misclass;

run;