BINARAY LOGISTIC REGRESSION MODEL CROSS VALIDATION

Contents

Cross Validation in Predictive Modeling

Cross Validation is a
process of evaluating the model on
'Out of Sample' data

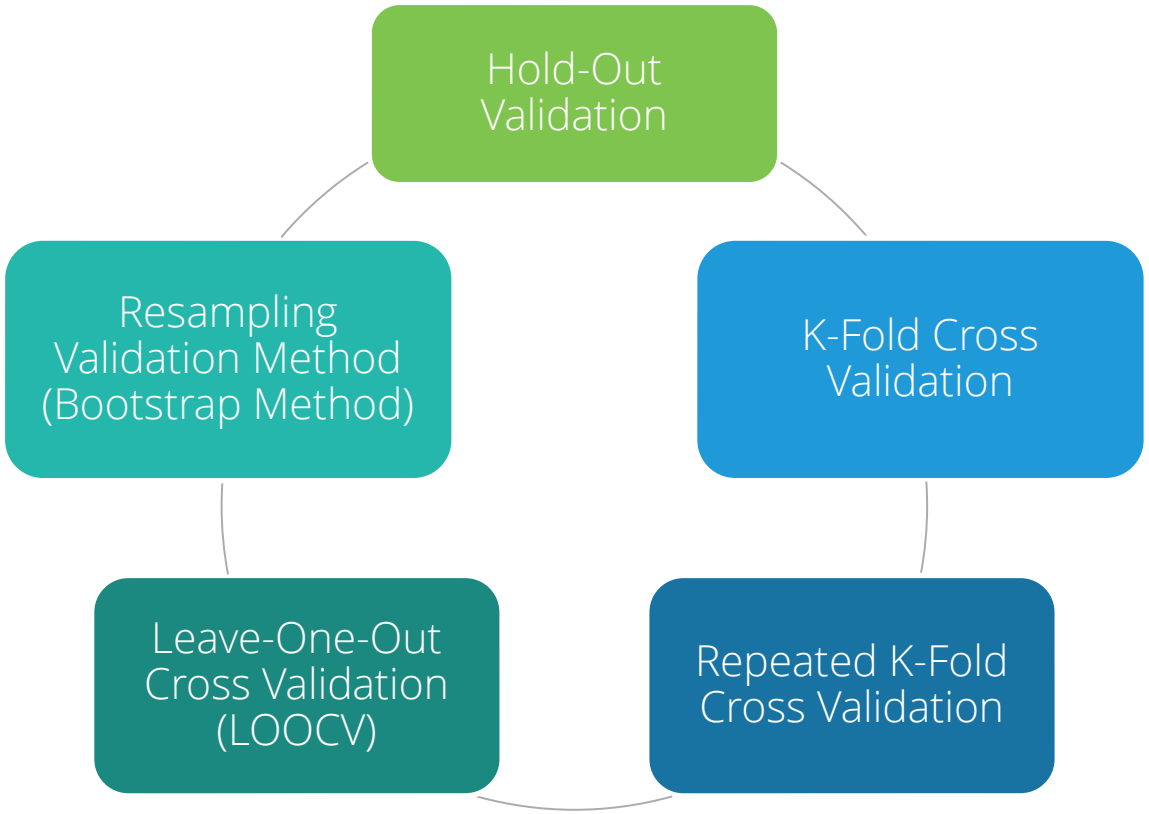
- **Model performance measures** for binary logistic regression such as Accuracy rate, Sensitivity, Specificity **tend to be optimistic on 'In Sample Data'**
- More realistic measures of model performance are calculated using "Out of Sample' data
- Cross-validation is a procedure for estimating the generalization performance in this context

Cross validation is important because although a model is built on historical data, ultimately it is to be used on future data. However good the model, if it fails on out of sample data then it defeats the purpose of predictive modeling

DATA SCIENCE
INSTITUTE

Cross Validation in Predictive Modeling

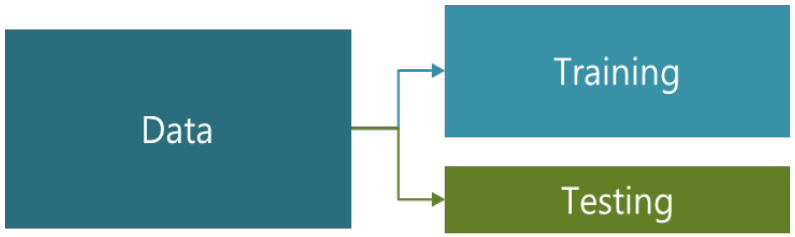
There are different approaches for cross validation. Five most significant of them are:



We will focus on **Hold Out** and **K-Fold** Cross validation methods.

DATA SCIENCE INSTITUTE

Hold-Out Validation



In Hold-Out validation method, available data is split into two non-overlapped parts: 'Training Data' and 'Testing Data'

- The model is
  - Developed using training data
  - Evaluated using testing data
- Training data should have more sample size. Typically 70%-80% data is used for model development

Here we continue to use previous data of bank loan for our further analysis.

Hold Out Validation in R

```
# Install and Load “caret” package
# Create 2 groups of the data: Training and Testing
```

```
install.packages("caret")
library(caret)

data <- read.csv("BANK LOAN.csv", header=TRUE)
index<-createDataPartition(data$DEFAULTER, p=0.7, list=FALSE)

dim(index)

traindata<-data[index,]
testdata<-data[-index,]
```

- **createDataPartition()** generates list of observation numbers to be included in training data.
- **p=** is the percentage of data that goes into training.
- **list=** specifies if results should be in a list format

DATA SCIENCE INSTITUTE

Hold Out Validation in R

```
# Check the dimensions of
dim(traindata)
[1] 490   8

dim(testdata)
[1] 210   8
```

The data of 700 observations are partitioned into 2 parts:
With 490 observations in training (model development) part and remaining 210 observations in testing data part (out of sample).

Hold Out Validation

- Model will be run on the training data and predicted probabilities will be generated.
- Same model will be applied to test data to get the predicted probabilities.
- Confusion matrix will be used to check the performance of the model in training and testing data.

Confusion Matrix

	Observed	
Predicted	Event	No Event
Event	A	B
No Event	C	D

- Sensitivity = A/(A + C)
- Specificity = D/(B + D)
- Prevalence = (A + C)/(A + B + C + D)
- Positive Predicted Value = A / (A + B)
- Negative Predicted Value = D / (C + D)
- Detection Rate = A/(A + B + C + D)
- Detection Prevalence = (A + B)/(A + B + C + D)
- Balanced Accuracy = (Sensitivity + Specificity)/2
- Precision = A/(A + B)
- Recall = A/(A + C)

Confusion Matrix in R

```
# Generate confusion matrix for training data
riskmodel<-glm(DEFAULTER~EMPLOY+ADDRESS+DEBTINC+CREDDEBT,
               family=binomial,data=traindata)

traindata$predprob<-predict(riskmodel,traindata,type='response')
traindata$predY<-ifelse(traindata$predprob>0.30,1,0)
traindata$predY<-factor(traindata$predY)
traindata$DEFAULTER<-factor(traindata$DEFAULTER)
confusionMatrix(traindata$predY,traindata$DEFAULTER,positive="1")
```

```
# Generate confusion matrix for test data
testdata$predprob<-predict(riskmodel,testdata,type='response')
testdata$predY<-ifelse(testdata$predprob>0.3,1,0)
testdata$predY<-factor(testdata$predY)
testdata$DEFAULTER<-factor(testdata$DEFAULTER)
confusionMatrix(testdata$predY,testdata$DEFAULTER,positive="1")
```

- **confusionMatrix()** creates cross-tabulation of observed and predicted classes with associated statistics. The function contains data, reference.
- **positive=** factor level that corresponds to a

DATA SCIENCE INSTITUTE

Confusion Matrix in R

Output:

For Training data

```
Confusion Matrix and Statistics

          Reference
Prediction   0   1
         0 286  31
         1  73 100

               Accuracy : 0.7878
                 95% CI : (0.7488, 0.8232)
    No Information Rate : 0.7327
    P-Value [Acc > NIR] : 0.002877

                  Kappa : 0.5083
 Mcnemar's Test P-Value : 5.81e-05

            Sensitivity : 0.7634
            Specificity : 0.7967
         Pos Pred Value : 0.5780
         Neg Pred Value : 0.9022
             Prevalence : 0.2673
         Detection Rate : 0.2041
   Detection Prevalence : 0.3531
      Balanced Accuracy : 0.7800

       'Positive' Class : 1
```

For Testing data

```
Confusion Matrix and Statistics

          Reference
Prediction   0   1
         0 127  14
         1  31  38

               Accuracy : 0.7857
                 95% CI : (0.724, 0.8392)
    No Information Rate : 0.7524
    P-Value [Acc > NIR] : 0.14903

                  Kappa : 0.4817
 Mcnemar's Test P-Value : 0.01707

            Sensitivity : 0.7308
            Specificity : 0.8038
         Pos Pred Value : 0.5507
         Neg Pred Value : 0.9007
             Prevalence : 0.2476
         Detection Rate : 0.1810
   Detection Prevalence : 0.3286
      Balanced Accuracy : 0.7673

       'Positive' Class : 1
```

Interpretation :

- Accuracy of both the data is almost same.
  Sensitivity is also similar of both the datasets.
  Model is performing well for test data.

* Note : Since the Train –test data is chosen randomly, output may vary slightly on different devices.

DATA SCIENCE INSTITUTE

K fold Cross Validation

- In k-fold cross-validation the data is first partitioned into k equally (or nearly equally) sized segments or folds.

- Then k iterations of training and testing are performed such that each time one fold is kept aside for testing and model is developed using k-1 folds.

DATA SCIENCE INSTITUTE

K-fold Validation in R

```
# Create k-folds
library(caret)
kfolds<-trainControl(method="cv",number=4)

riskmodel<-train(as.factor(DEFAULTER)~EMPLOY+ADDRESS+
                   DEBTINC+CREDDEBT,data=data,method="glm",
                 family=binomial,trControl=kfolds)
riskmodel
```

- **trainControl()** control the computational nuances of the train function.
- **method="cv"** tells R to use Cross Validation method.
- **number=** specifies the number of folds.
- **train ()** fits predictive models over different tuning parameters.
- It performs a number of classification and regression routines, fits each model and

K-fold Validation in R

```
# Output:

Generalized Linear Model

700 samples
  4 predictor
  2 classes: '0', '1'

No pre-processing
Resampling: Cross-Validated (4 fold)
Summary of sample sizes: 525, 524, 525, 526
Resampling results:

  Accuracy  Kappa
  0.810054  0.4595083
```

Interpretation : accuracy of 0.81 indicates the good model.

* Note : Since observations are assigned randomly in kfolds, output may vary slightly on different devices.

K-fold Validation in R

```
# Generate confusion matrix for k-fold validation
library(caret)
data$pred<-riskmodel$finalModel$fitted.values
data$predY<-ifelse(data$pred>0.3,1,0)

data$predY<-factor(data$predY)
data$DEFAULTER<-factor(data$DEFAULTER)

confusionMatrix(data$predY,data$DEFAULTER,positive="1")
```

- **riskmodel$finalModel$fitted.values:** Extract fitted model values from “riskmodel”.

K-fold Validation in R

```
# Output:
```

```
Confusion Matrix and Statistics

          Reference
Prediction   0   1
         0 415  45
         1 102 138

               Accuracy : 0.79
                 95% CI : (0.7579, 0.8196)
    No Information Rate : 0.7386
    P-Value [Acc > NIR] : 0.0009121

                  Kappa : 0.5059
 Mcnemar's Test P-Value : 3.86e-06

            Sensitivity : 0.7541
            Specificity : 0.8027
         Pos Pred Value : 0.5750
         Neg Pred Value : 0.9022
             Prevalence : 0.2614
         Detection Rate : 0.1971
   Detection Prevalence : 0.3429
      Balanced Accuracy : 0.7784

       'Positive' Class : 1
```

Interpretation : sensitivity and accuracy indicates stable model.

DATA SCIENCE INSTITUTE

Quick Recap

In this session, we studied about model validation of Binary Logistic :

Cross Validation	• Cross Validation is a process of evaluating the model on 'Out of Sample' data.
Hold out validation	• In Hold-Out validation method, available data is split into two non-overlapped parts: 'Training Data' and 'Testing Data'.
Confusion matrix	• It is used to check the performance of the model in training and testing data. • It has performance measures as Accuracy, sensitivity, specificity, etc..
K-fold validation	• In k-fold cross-validation the data is first partitioned into k equally (or nearly equally) sized segments or folds. • Then k iterations of training and testing are performed such that each time one fold is kept aside for testing and model is developed using k-1 folds.

THANK YOU!!

DATA SCIENCE
INSTITUTE