How to evaluate your prediction

Zhihan Fang

Load your data

Training data: data that is used for training the model, all decision tree, random forest and lda model

Testing without label: data without incidents, you need to predict the incidents(bite or not bite).

Testing data: sample solution including the right answer to calculate the error rate of the result.

```
#import training and testing data
lou_training <-  read.csv("~/workspace/data_literacy/week10/lou_training.csv")
lou_testing_nolabel <- read.csv("~/workspace/data_literacy/week10/lou_testing_nolabel.csv")
```

Cross Validation

rpart(): train the relation between INCIDENT and other attributes using decision tree model.

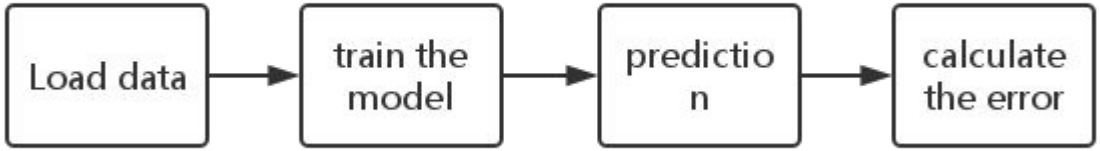
randomForest(): train the relation between INCIDENT and other attributes using random forest model.

lda(): train the relation between INCIDENT and other attributes using linear discriminant analysis model.

Decision Tree ? Linear Discriminant Analysis? Or random forest? Which one is best for the problem?

How to determine which attributes can be selected to do prediction?

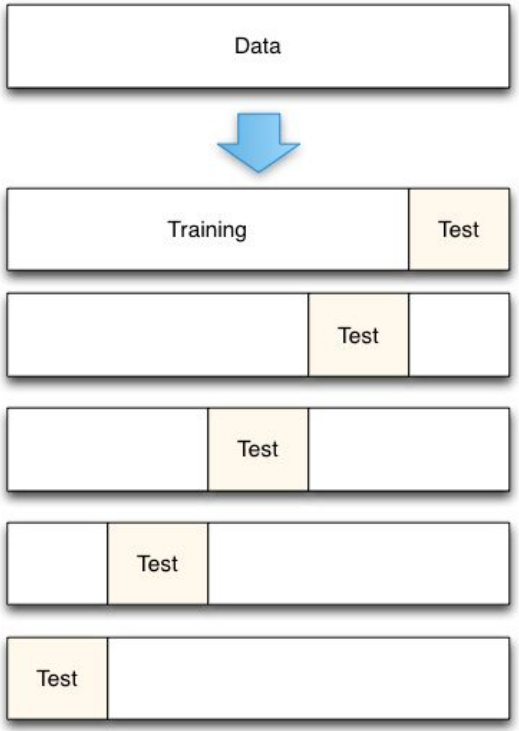
Last challenge?



Find the model with the smallest error.

Problem: we do not have the solution to calculate the error. But we have the solution in the training data set.

Cross Validation

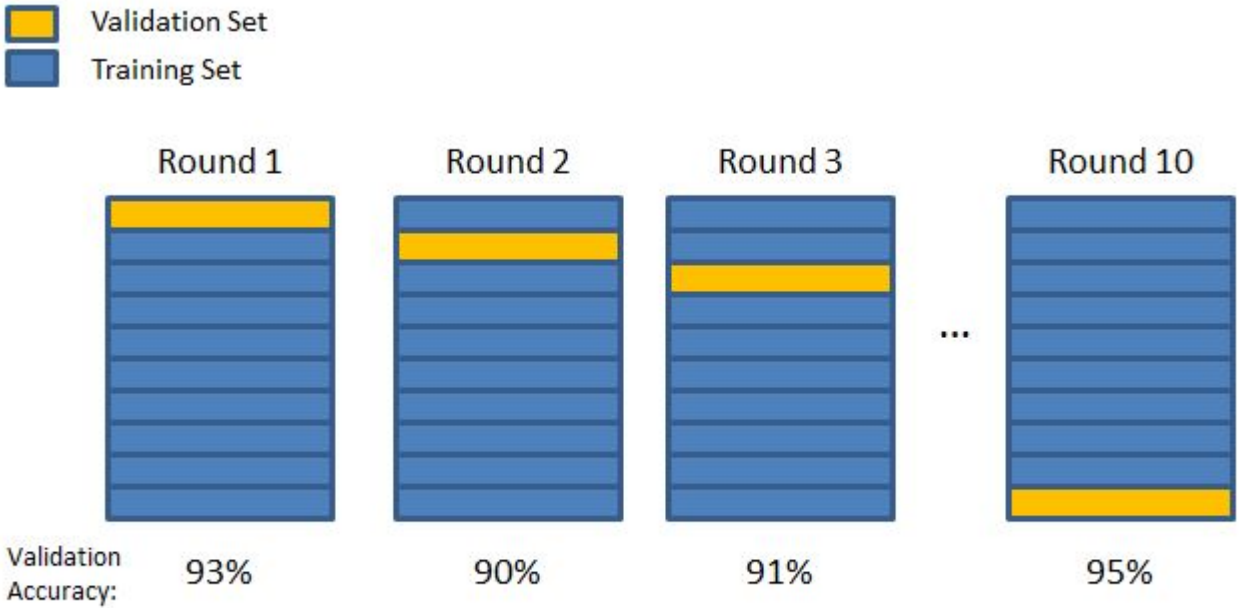


Data is your training data, separate your training data into two data sets.

1. One big data set for training
2. One small data set for testing

There is no overlap between new training and testing

Cross Validation



Final Accuracy = Average(Round 1, Round 2, ...)

Validation

```
#import your package before using it
library(randomForest)
library(MASS)
library(rpart)

#import training data
lou_training <- read.csv("~/workspace/data_literacy/week11/lou_training.csv")

#set fold number
fold_number = 10
n_row <- nrow(lou_training)
all_index <- 1:n_row
#how many rows in each fold
row_number_each_fold <-n_row/fold_number
```

Validation

```
#vector that contains all results from validation
error.tree <- c()
error.lda <- c()
error.forest <- c()
```

Validation

```
# first time, set first fold as testing data; second loop, set second fold as test data, and so on.
for (i in 1:fold_number){
        # set start and end index of testing
        start_point <- (i-1) * row_number_each_fold +1
        end_point <- i * row_number_each_fold
        #get test index
        test_index <- start_point:end_point
        #train index = remove test index from all index
        train_index <- all_index[-test_index]
        #get data based on index
        new_testing <- lou_training[test_index,]
        new_training <- lou_training[train_index,]
        #decision tree model
        lou.tree <- rpart(INCIDENT ~ PENALTY + VICTIM + LOCATION +COUNTRY + RESULT + GOALS + YEAR,data=new_training)
        pred.tree <- predict(lou.tree,newdata=new_testing,type="class")
        error.tree[i] <- mean(pred.tree != new_testing$INCIDENT)
        #random forest model
        lou.forest <- randomForest(INCIDENT ~ PENALTY + VICTIM + LOCATION +COUNTRY + RESULT + GOALS + YEAR,data=new_training)
        pred.forest <- predict(lou.forest,newdata=new_testing,type="class")
        error.forest[i] <- mean(pred.forest != new_testing$INCIDENT)
        #lda model
        lou.lda <- lda(INCIDENT ~ PENALTY + VICTIM + LOCATION +COUNTRY + RESULT + GOALS + YEAR,data=new_training)
        pred.lda <- predict(lou.lda,newdata=new_testing)$class
        error.lda[i] <- mean(pred.lda != new_testing$INCIDENT)}
```

Error for numerical value

$$\mathrm{MSE} = \frac{1}{n}\sum_{i=1}^{n}(\hat{Y}_i - Y_i)^2$$

$\hat{Y}$ Is your prediction value and Y is the real results.