Prediction in R

Zhihan Fang

Install package

```
#install new package
install.packages('randomForest',dependencies = T)
install.packages('rpart',dependencies = T)
install.packages('MASS',dependencies = T)

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

#check package information
help(rpart)
help(randomForest)
help(lda)
```

#package information

rpart: recursive partitioning and regression tree

randomForest: classification and regression with random forest.

lda: linear discriminant analysis

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")
```

Train the model

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.

```
#train data model
lou.tree <- rpart(INCIDENT ~ PENALTY + VICTIM + LOCATION +COUNTRY + RESULT + GOALS +
YEAR,data=lou_training)

lou.forest <- randomForest(INCIDENT ~ PENALTY + VICTIM + LOCATION +COUNTRY + RESULT
+ GOALS + YEAR,data=lou_training)

lou.lda <- lda(INCIDENT ~ PENALTY + VICTIM + LOCATION +COUNTRY + RESULT + GOALS +
YEAR,data=lou_training)
```

Prediction

We train data by training data set, in which the incident (bite or not bite) is given. Then we predict the incident type (if Louis bites or not in a game) in the testing data.

```
# Now use these *models* to predict the labels in the testing data (i.e. predict whether or not a bite incident)

pred.tree <- predict(lou.tree,newdata=lou_testing_nolabel,type="class")
pred.forest <- predict(lou.forest,newdata=lou_testing_nolabel,type="class")
pred.lda <- predict(lou.lda,newdata=lou_testing_nolabel)$class
```

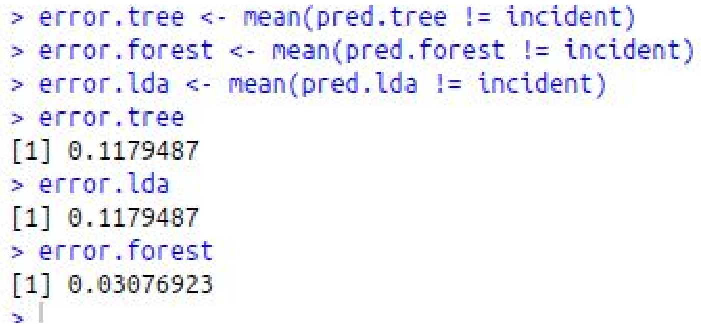
Calculate the error

When prediction is done, the error can be calculated by the standard solution(which is not given when you do the project).

```
lou_testing <- read.csv("~/workspace/data_literacy/week10/lou_testing.csv")
incident <- lou_testing$INCIDENT

#calculate the ratio of wrong prediction
error.tree <- mean(pred.tree != incident)
error.forest <- mean(pred.forest != incident)
error.lda <- mean(pred.lda != incident)
```

Error rate



```
> error.tree <- mean(pred.tree != incident)
> error.forest <- mean(pred.forest != incident)
> error.lda <- mean(pred.lda != incident)
> error.tree
[1] 0.1179487
> error.lda
[1] 0.1179487
> error.forest
[1] 0.03076923
>
```

Here, the random forest model has the lowest error rate.

In your project, you can use 20% of training data to do test and calculate the error rate when the test solution is not given to you.