random_forest.R

Magilan

Mon Oct 08 17:27:38 2018

```
library(party)
```

```
## Loading required package: grid
```

```
## Loading required package: mvtnorm
```

```
## Loading required package: modeltools
```

```
## Loading required package: stats4
```

```
## Loading required package: strucchange
```

```
## Loading required package: zoo
```

```
##
## Attaching package: 'zoo'
```

```
## The following objects are masked from 'package:base':
##
##     as.Date, as.Date.numeric
```

```
## Loading required package: sandwich
```

```
library(randomForest)
```

```
## randomForest 4.6-14
```

```
## Type rfNews() to see new features/changes/bug fixes.
```

```
library(caret)
```

```
## Loading required package: lattice
```

```
## Loading required package: ggplot2
```

```
##
## Attaching package: 'ggplot2'
```

```
## The following object is masked from 'package:randomForest':
##
##     margin
```

```
# Data Input

data <- read.csv("C:/Users/Magilan/Desktop/ML_project/austin_weather.csv",header = TRUE)
data1=na.omit(data,invert=FALSE)
attach(data1)

# Data Partitioning

index <- createDataPartition(Rain, p = 0.7, list = FALSE)
train.df <- data1[index,-c(1,20,22)]
test.df <- data1[-index,-c(1,20,21,22)]
test.Y <- data1[-index,21]

# Random Forest

model.rf = randomForest(Rain ~ ., data= train.df)

pred <- predict(model.rf, test.df, type ="response")
head(pred)
```

```
##   1   6  11  18  19  22
## yes  no  no  no yes  no
## Levels: no yes
```

```
confusionMatrix(pred,test.Y)
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction  no yes
##        no  233  35
##        yes  24  98
##
##                Accuracy : 0.8487
##                  95% CI : (0.8092, 0.8828)
##     No Information Rate : 0.659
##     P-Value [Acc > NIR] : <2e-16
##
##                   Kappa : 0.6566
##  Mcnemar's Test P-Value : 0.193
##
##             Sensitivity : 0.9066
##             Specificity : 0.7368
##          Pos Pred Value : 0.8694
##          Neg Pred Value : 0.8033
##              Prevalence : 0.6590
##          Detection Rate : 0.5974
##    Detection Prevalence : 0.6872
##       Balanced Accuracy : 0.8217
##
##        'Positive' Class : no
##
```

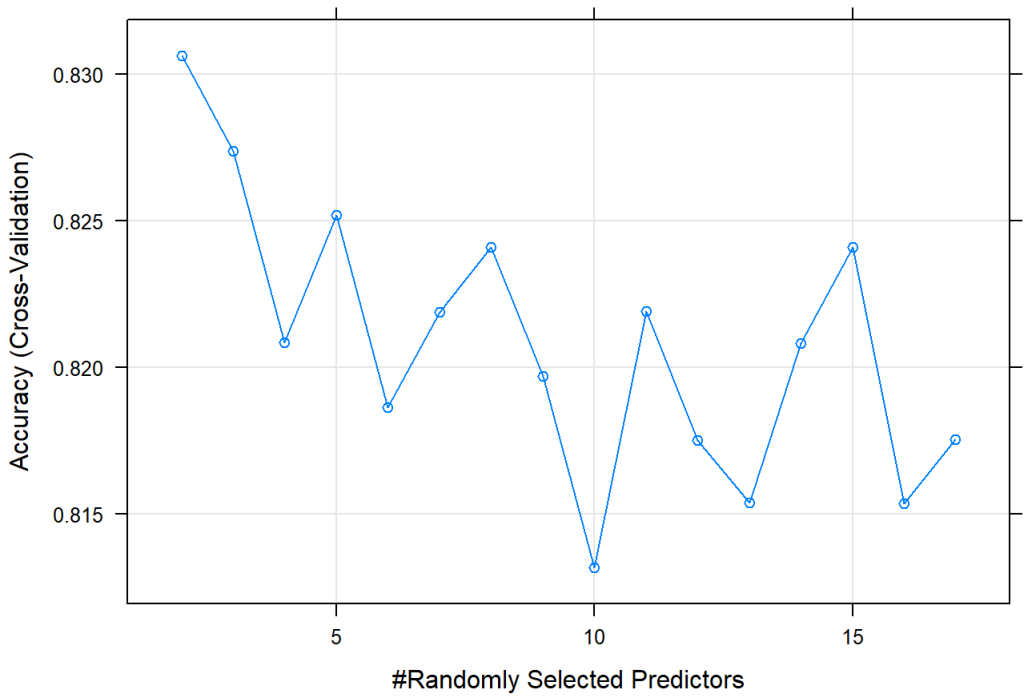
```
# Cross Validation

model.rf <- train(
  Rain ~., data = train.df[,-c(1,20,22)], method = "rf",
  trControl = trainControl("cv", number = 10),
  preProcess = c("center","scale"),
  tuneLength = 20
)
```

```
## note: only 16 unique complexity parameters in default grid. Truncating the grid to 16 .
```

```
model.rf
```

```
## Random Forest
##
## 915 samples
##  17 predictor
##   2 classes: 'no', 'yes'
##
## Pre-processing: centered (17), scaled (17)
## Resampling: Cross-Validated (10 fold)
## Summary of sample sizes: 823, 823, 823, 823, 823, 824, ...
## Resampling results across tuning parameters:
##
##   mtry  Accuracy   Kappa
##    2    0.8306259  0.6085034
##    3    0.8273650  0.6036578
##    4    0.8208313  0.5887250
##    5    0.8251911  0.5986349
##    6    0.8186335  0.5841256
##    7    0.8218705  0.5922942
##    8    0.8241042  0.5963846
##    9    0.8197086  0.5876869
##   10    0.8131629  0.5724501
##   11    0.8219183  0.5903565
##   12    0.8175227  0.5832722
##   13    0.8153727  0.5754351
##   14    0.8208194  0.5879672
##   15    0.8241042  0.5974383
##   16    0.8153488  0.5771795
##   17    0.8175466  0.5813555
##
## Accuracy was used to select the optimal model using the largest value.
## The final value used for the model was mtry = 2.
```

```
plot(model.rf)
```




```
k=model.rf$bestTune
k
```

```
##   mtry
## 1    2
```

```
pred.cv = predict(model.rf,test.df)
confusionMatrix(pred.cv,test.Y)
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction  no yes
##        no  233  42
##        yes  24  91
##
##                Accuracy : 0.8308
##                  95% CI : (0.7898, 0.8666)
##     No Information Rate : 0.659
##     P-Value [Acc > NIR] : 2.692e-14
##
##                   Kappa : 0.6108
##  Mcnemar's Test P-Value : 0.03639
##
##             Sensitivity : 0.9066
##             Specificity : 0.6842
##          Pos Pred Value : 0.8473
##          Neg Pred Value : 0.7913
##              Prevalence : 0.6590
##          Detection Rate : 0.5974
##    Detection Prevalence : 0.7051
##       Balanced Accuracy : 0.7954
##
##        'Positive' Class : no
##
```

```
model.rf1 = randomForest(Rain ~ ., data= train.df , mtry = 15)
pred1 <- predict(model.rf1, test.df, type ="response")
confusionMatrix(pred1,test.Y)
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction  no yes
##        no  232  34
##        yes  25  99
##
##                Accuracy : 0.8487
##                  95% CI : (0.8092, 0.8828)
##     No Information Rate : 0.659
##     P-Value [Acc > NIR] : <2e-16
##
##                   Kappa : 0.6578
##  Mcnemar's Test P-Value : 0.2976
##
##             Sensitivity : 0.9027
##             Specificity : 0.7444
##          Pos Pred Value : 0.8722
##          Neg Pred Value : 0.7984
##              Prevalence : 0.6590
##          Detection Rate : 0.5949
##    Detection Prevalence : 0.6821
##       Balanced Accuracy : 0.8235
##
##        'Positive' Class : no
##
```