

Group 7

2022/5/12

```
library(glmnet)
library(MASS)
library(class)
library(caret)
library(e1071)
library(mboost)
library(plyr)
library(import)
library(ipred)
library(LiblineaR)
library(naivebayes)
library(nnet)
library(randomForest)
library(gbm)
```

```
set.seed(1082)
data = read.csv("haralick_median blur.csv", header = T)
data$Label <- factor(data$Label)
```

```
trControl = trainControl(method = "cv", number = 5)
```

Penalized Logistic Regression

```
tuneGrid <- expand.grid(alpha = 1, lambda = seq(0, 5, by = 0.1))

model = train(Label ~ ., data = data, method = "glmnet", tuneGrid = tuneGrid,
              trControl = trControl)
model
```

```
## glmnet
##
## 1500 samples
## 52 predictor
## 6 classes: '0', '1', '2', '3', '4', '5'
##
## No pre-processing
## Resampling: Cross-Validated (5 fold)
## Summary of sample sizes: 1199, 1202, 1199, 1201, 1199
## Resampling results across tuning parameters:
##
##   lambda  Accuracy  Kappa
##   0.0      0.9460503  0.9320962
##   0.1      0.6486047  0.5374120
##   0.2      0.5786969  0.4375538
##   0.3      0.4980178  0.3233855
##   0.4      0.2633380  0.0000000
##   0.5      0.2633380  0.0000000
##   0.6      0.2633380  0.0000000
##   0.7      0.2633380  0.0000000
##   0.8      0.2633380  0.0000000
##   0.9      0.2633380  0.0000000
##   1.0      0.2633380  0.0000000
##   1.1      0.2633380  0.0000000
##   1.2      0.2633380  0.0000000
##   1.3      0.2633380  0.0000000
##   1.4      0.2633380  0.0000000
##   1.5      0.2633380  0.0000000
##   1.6      0.2633380  0.0000000
##   1.7      0.2633380  0.0000000
##   1.8      0.2633380  0.0000000
##   1.9      0.2633380  0.0000000
##   2.0      0.2633380  0.0000000
##   2.1      0.2633380  0.0000000
##   2.2      0.2633380  0.0000000
##   2.3      0.2633380  0.0000000
##   2.4      0.2633380  0.0000000
##   2.5      0.2633380  0.0000000
##   2.6      0.2633380  0.0000000
##   2.7      0.2633380  0.0000000
##   2.8      0.2633380  0.0000000
##   2.9      0.2633380  0.0000000
##   3.0      0.2633380  0.0000000
##   3.1      0.2633380  0.0000000
##   3.2      0.2633380  0.0000000
```

```
## 3.3      0.2633380  0.0000000
## 3.4      0.2633380  0.0000000
## 3.5      0.2633380  0.0000000
## 3.6      0.2633380  0.0000000
## 3.7      0.2633380  0.0000000
## 3.8      0.2633380  0.0000000
## 3.9      0.2633380  0.0000000
## 4.0      0.2633380  0.0000000
## 4.1      0.2633380  0.0000000
## 4.2      0.2633380  0.0000000
## 4.3      0.2633380  0.0000000
## 4.4      0.2633380  0.0000000
## 4.5      0.2633380  0.0000000
## 4.6      0.2633380  0.0000000
## 4.7      0.2633380  0.0000000
## 4.8      0.2633380  0.0000000
## 4.9      0.2633380  0.0000000
## 5.0      0.2633380  0.0000000
##
## Tuning parameter 'alpha' was held constant at a value of 1
## Accuracy was used to select the optimal model using the largest value.
## The final values used for the model were alpha = 1 and lambda = 0.
```

```
confusionMatrix(model, norm = "none")
```

```
## Cross-Validated (5 fold) Confusion Matrix
##
## (entries are un-normalized aggregated counts)
##
##           Reference
## Prediction  0   1   2   3   4   5
##           0 380   8   0  13   0   0
##           1   8 274   0   9   3   1
##           2   0   1  57   0   0   0
##           3   7   9   1 203   7   0
##           4   0   7   0   2 136   1
##           5   0   4   0   0   0 369
##
## Accuracy (average) : 0.946
```

KNN

```
knn.fit <- train(Label ~ ., method = "knn", tuneGrid = expand.grid(k = 5),  
  trControl = trControl, metric = "Accuracy", data = data)  
knn.fit
```

```
## k-Nearest Neighbors  
##  
## 1500 samples  
## 52 predictor  
## 6 classes: '0', '1', '2', '3', '4', '5'  
##  
## No pre-processing  
## Resampling: Cross-Validated (5 fold)  
## Summary of sample sizes: 1199, 1199, 1201, 1201, 1200  
## Resampling results:  
##  
## Accuracy Kappa  
## 0.7093552 0.6315787  
##  
## Tuning parameter 'k' was held constant at a value of 5
```

Random Forest

```
rf.fit <- train(Label ~ ., method = "rf", trControl = trControl,  
  metric = "Accuracy", data = data)
```

```
rf.fit
```

```
## Random Forest  
##  
## 1500 samples  
## 52 predictor  
## 6 classes: '0', '1', '2', '3', '4', '5'  
##  
## No pre-processing  
## Resampling: Cross-Validated (5 fold)  
## Summary of sample sizes: 1198, 1201, 1200, 1201, 1200  
## Resampling results across tuning parameters:  
##  
## mtry Accuracy Kappa  
## 2 0.9366703 0.9199542  
## 27 0.9366681 0.9200801  
## 52 0.9213323 0.9007005  
##  
## Accuracy was used to select the optimal model using the largest value.  
## The final value used for the model was mtry = 2.
```

```
confusionMatrix(rf.fit, norm = "none")
```

```
## Cross-Validated (5 fold) Confusion Matrix  
##  
## (entries are un-normalized aggregated counts)  
##  
##           Reference  
## Prediction  0  1  2  3  4  5  
##           0 387  9  4 21  0  0  
##           1  5 280  0 28  7  1  
##           2  0  0 54  0  0  0  
##           3  3  5  0 176  1  0  
##           4  0  0  0  2 138  0  
##           5  0  9  0  0  0 370  
##  
## Accuracy (average) : 0.9367
```

Boosting Tree

```
treeboost.fit <- train(Label ~ ., method = "gbm", verbose = FALSE,  
  trControl = trControl, metric = "Accuracy", data = data)  
treeboost.fit
```

```
## Stochastic Gradient Boosting  
##  
## 1500 samples  
## 52 predictor  
## 6 classes: '0', '1', '2', '3', '4', '5'  
##  
## No pre-processing  
## Resampling: Cross-Validated (5 fold)  
## Summary of sample sizes: 1199, 1200, 1200, 1201, 1200  
## Resampling results across tuning parameters:  
##  
## interaction.depth n.trees Accuracy Kappa  
## 1 50 0.8826351 0.8511830  
## 1 100 0.9126418 0.8895743  
## 1 150 0.9199730 0.8988969  
## 2 50 0.9239685 0.9038386  
## 2 100 0.9299774 0.9115167  
## 2 150 0.9393108 0.9233873  
## 3 50 0.9333041 0.9157462  
## 3 100 0.9453197 0.9309898  
## 3 150 0.9459886 0.9318364  
##  
## Tuning parameter 'shrinkage' was held constant at a value of 0.1  
##  
## Tuning parameter 'n.minobsinnode' was held constant at a value of 10  
## Accuracy was used to select the optimal model using the largest value.  
## The final values used for the model were n.trees = 150, interaction.depth =  
## 3, shrinkage = 0.1 and n.minobsinnode = 10.
```

```
confusionMatrix(treeboost.fit, norm = "none")
```

```
## Cross-Validated (5 fold) Confusion Matrix  
##  
## (entries are un-normalized aggregated counts)  
##  
##  
## Reference  
## Prediction 0 1 2 3 4 5  
## 0 386 11 4 18 0 1  
## 1 8 280 0 9 3 1  
## 2 0 1 53 0 0 0  
## 3 1 7 1 197 7 0  
## 4 0 1 0 3 136 2  
## 5 0 3 0 0 0 367  
##  
## Accuracy (average) : 0.946
```

SVM

```
svm.fit <- train(Label ~ ., method = "svmRadial", trControl = trControl,  
  metric = "Accuracy", data = data)  
svm.fit
```

```
## Support Vector Machines with Radial Basis Function Kernel  
##  
## 1500 samples  
## 52 predictor  
## 6 classes: '0', '1', '2', '3', '4', '5'  
##  
## No pre-processing  
## Resampling: Cross-Validated (5 fold)  
## Summary of sample sizes: 1202, 1198, 1199, 1201, 1200  
## Resampling results across tuning parameters:  
##  
## C Accuracy Kappa  
## 0.25 0.8453499 0.8015834  
## 0.50 0.8913376 0.8620260  
## 1.00 0.9153559 0.8927404  
##  
## Tuning parameter 'sigma' was held constant at a value of 0.01724361  
## Accuracy was used to select the optimal model using the largest value.  
## The final values used for the model were sigma = 0.01724361 and C = 1.
```

```
confusionMatrix(svm.fit, norm = "none")
```

```
## Cross-Validated (5 fold) Confusion Matrix  
##  
## (entries are un-normalized aggregated counts)  
##  
##           Reference  
## Prediction  0  1  2  3  4  5  
##           0 382 11  8 45  3  0  
##           1 10 279  0 20  7  3  
##           2  1  0 50  0  0  0  
##           3  2  6  0 160  1  0  
##           4  0  1  0  2 134  0  
##           5  0  6  0  0  1 368  
##  
## Accuracy (average) : 0.9153
```

Summary

Model	Predictor	Parameter	Accuracy
Penalized Logistic Regression	52	$\lambda = 0$	0.9460
KNN	52	K=5	0.7093
Random Forest	52	mtry = 2	0.9367
Boosting Tree	52	shrinkage = 0.1	0.9460
SVM	52	sigma = 0.0172 , C = 1	0.9153