# DATA ANALYTICS WITH R, EXCEL AND TABLAEU

## ASSIGNMENT Time series forecasting 20.1

## ANSWERS

**By ASHISH S SHANBHAG**

**ashishshanbhag108@gmail.com**

## Question no:

## 5)

## 2. Perform the below given activities:

## a. Create classification model using different random forest models
## Ans

```
# parallel processing
registerDoMC(cores = getDoParWorkers())
# 4 fold cross-validation
ctrl <- trainControl(allowParallel=T, method="cv", number=4)

# train the model
model_rf <- train(classe ~ ., data=training_reduced, model="rf", trControl=ctrl)
# make predictions on the validation set
pred_rf <- predict(model_rf, validation_reduced[,-35])
# confusion matrix
cm_rf <- confusionMatrix(validation_reduced$classe, pred_rf)
cm_rf
## Confusion Matrix and Statistics
##
##           Reference
## Prediction    A    B    C    D    E
##          A 1391    2    2    0    0
##          B    4  942    3    0    0
##          C    0    2  846    7    0
##          D    1    0   11  791    1
##          E    0    0    0    2  899
##
## Overall Statistics
##
##                Accuracy : 0.9929
##                  95% CI : (0.9901, 0.995)
##     No Information Rate : 0.2847
##     P-Value [Acc > NIR] : < 2.2e-16
##
##                   Kappa : 0.991
```

```
##  Mcnemar's Test P-Value : NA
##
## Statistics by Class:
##
##                      Class: A Class: B Class: C Class: D Class: E
## Sensitivity            0.9964   0.9958   0.9814   0.9888   0.9989
## Specificity            0.9989   0.9982   0.9978   0.9968   0.9995
## Pos Pred Value         0.9971   0.9926   0.9895   0.9838   0.9978
## Neg Pred Value         0.9986   0.9990   0.9960   0.9978   0.9998
## Prevalence             0.2847   0.1929   0.1758   0.1631   0.1835
## Detection Rate         0.2836   0.1921   0.1725   0.1613   0.1833
## Detection Prevalence   0.2845   0.1935   0.1743   0.1639   0.1837
## Balanced Accuracy      0.9976   0.9970   0.9896   0.9928   0.9992
# accuracy
cm_rf$overall['Accuracy']
## Accuracy
## 0.992863
# make predictions on the testing dataset
pred_rf_testing <- predict(model_rf, test)
pred_rf_testing
##  [1] B A B A A E D B A A B C B A E E A B B B
## Levels: A B C D E
```

## b. Verify model goodness of fit
## Ans

Load all the required libraries

```
library(caret)
library(corrplot)
library(rattle)
library(rpart.plot)
library(doMC)
library(randomForest)
```

Loading the data

We first download and load the datasets into our working directory in R, assigning missing values to entries that are currently ‘NA’, blank and “#DIV/0!”

```
train <- read.csv("pml-training.csv", na.strings = c("NA", "#DIV/0!", ""))
test <- read.csv("pml-testing.csv", na.strings = c("NA", "#DIV/0!", ""))
dim(train)
## [1] 19622   160
dim(test)
## [1]  20 160
```

Basic pre-processing

We now discard columns which contain more than 90% NA values.

```
train <- train[ , colMeans(is.na(train)) <= .90]
dim(train)
```

```
## [1] 19622    60
```

We also discard variables that contain timestamp and date information

```
train<- subset(train, select = -c(1,2,3,4,5,6,7))
dim(train)
## [1] 19622    53
```

Partitioning the data

As a next step, we partition the training dataset into a training set (75%) and a validation set (25%).

```
set.seed(1)
inTrain = createDataPartition(y=train$classe, p=0.75, list=FALSE)
training = train[inTrain,]
validation =  train[-inTrain,]
dim(training)
## [1] 14718    53
dim(validation)
## [1] 4904   53
```

## c. Apply all the model validation techniques
## Ans

**Model 1 : Decision Trees**

```
# train the model
model_rpart <- train(classe~., data=training_reduced, method = "rpart")
# make predictions for the validation set
pred_rpart <- predict(model_rpart, validation_reduced[,-35])
# print the confusion matrix
cm_rpart <- confusionMatrix(validation_reduced$classe, pred_rpart)
cm_rpart
## Confusion Matrix and Statistics
##
##           Reference
## Prediction   A   B   C   D   E
##          A 846 270 207  72   0
##          B 150 599 177  23   0
##          C  21 157 637  39   1
##          D  45 278 204 221  56
##          E  17 376 161  35 312
##
## Overall Statistics
##
##                Accuracy : 0.5332
##                  95% CI : (0.5192, 0.5473)
##     No Information Rate : 0.3426
##     P-Value [Acc > NIR] : < 2.2e-16
##
##                   Kappa : 0.4129
##  Mcnemar's Test P-Value : < 2.2e-16
```

```
##
## Statistics by Class:
##
##                      Class: A Class: B Class: C Class: D Class: E
## Sensitivity            0.7841   0.3565   0.4596  0.56667  0.84553
## Specificity            0.8565   0.8914   0.9380  0.87085  0.87012
## Pos Pred Value         0.6065   0.6312   0.7450  0.27488  0.34628
## Neg Pred Value         0.9336   0.7267   0.8150  0.95878  0.98576
## Prevalence             0.2200   0.3426   0.2826  0.07953  0.07524
## Detection Rate         0.1725   0.1221   0.1299  0.04507  0.06362
## Detection Prevalence   0.2845   0.1935   0.1743  0.16395  0.18373
## Balanced Accuracy      0.8203   0.6240   0.6988  0.71876  0.85782
# accuracy
cm_rpart$overall['Accuracy']
##  Accuracy
## 0.5332382
```

**Model 2: Random Forests**

```
# parallel processing
registerDoMC(cores = getDoParWorkers())
# 4 fold cross-validation
ctrl <- trainControl(allowParallel=T, method="cv", number=4)

# train the model
model_rf <- train(classe ~ ., data=training_reduced, model="rf", trControl=ctrl)
# make predictions on the validation set
pred_rf <- predict(model_rf, validation_reduced[,-35])
# confusion matrix
cm_rf <- confusionMatrix(validation_reduced$classe, pred_rf)
cm_rf
## Confusion Matrix and Statistics
##
##           Reference
## Prediction    A    B    C    D    E
##          A 1391    2    2    0    0
##          B    4  942    3    0    0
##          C    0    2  846    7    0
##          D    1    0   11  791    1
##          E    0    0    0    2  899
##
## Overall Statistics
##
##                Accuracy : 0.9929
##                  95% CI : (0.9901, 0.995)
##     No Information Rate : 0.2847
##     P-Value [Acc > NIR] : < 2.2e-16
##
##                   Kappa : 0.991
##  Mcnemar's Test P-Value : NA
```

```
##
## Statistics by Class:
##
##                      Class: A Class: B Class: C Class: D Class: E
## Sensitivity            0.9964   0.9958   0.9814   0.9888   0.9989
## Specificity            0.9989   0.9982   0.9978   0.9968   0.9995
## Pos Pred Value         0.9971   0.9926   0.9895   0.9838   0.9978
## Neg Pred Value         0.9986   0.9990   0.9960   0.9978   0.9998
## Prevalence             0.2847   0.1929   0.1758   0.1631   0.1835
## Detection Rate         0.2836   0.1921   0.1725   0.1613   0.1833
## Detection Prevalence   0.2845   0.1935   0.1743   0.1639   0.1837
## Balanced Accuracy      0.9976   0.9970   0.9896   0.9928   0.9992
# accuracy
cm_rf$overall['Accuracy']
## Accuracy
## 0.992863
# make predictions on the testing dataset
pred_rf_testing <- predict(model_rf, test)
pred_rf_testing
##  [1] B A B A A E D B A A B C B A E E A B B B
## Levels: A B C D E
```

**Model 3: Gradient Boost Machine**

```
# train the model
model_gbm <- train(classe ~ ., data=training_reduced, model="gbm", trControl=ctrl)
pred_gbm <- predict(model_gbm, validation_reduced[,-35])
# confusion matrix
cm_gbm <-confusionMatrix(validation_reduced$classe, pred_gbm)
cm_gbm
## Confusion Matrix and Statistics
##
##           Reference
## Prediction    A    B    C    D    E
##          A 1389    3    2    0    1
##          B    4  941    4    0    0
##          C    0    4  844    7    0
##          D    1    0   11  791    1
##          E    0    0    0    2  899
##
## Overall Statistics
##
##                Accuracy : 0.9918
##                  95% CI : (0.9889, 0.9942)
##     No Information Rate : 0.2843
##     P-Value [Acc > NIR] : < 2.2e-16
##
##                   Kappa : 0.9897
##  Mcnemar's Test P-Value : NA
##
```

```
## Statistics by Class:
##
##                      Class: A Class: B Class: C Class: D Class: E
## Sensitivity            0.9964   0.9926   0.9803   0.9888   0.9978
## Specificity            0.9983   0.9980   0.9973   0.9968   0.9995
## Pos Pred Value         0.9957   0.9916   0.9871   0.9838   0.9978
## Neg Pred Value         0.9986   0.9982   0.9958   0.9978   0.9995
## Prevalence             0.2843   0.1933   0.1756   0.1631   0.1837
## Detection Rate         0.2832   0.1919   0.1721   0.1613   0.1833
## Detection Prevalence   0.2845   0.1935   0.1743   0.1639   0.1837
## Balanced Accuracy      0.9974   0.9953   0.9888   0.9928   0.9986
# accuracy
cm_gbm$overall['Accuracy']
##  Accuracy
## 0.9918434
# make predictions on the testing set
pred_gbm_testing <- predict( model_gbm, test)
pred_gbm_testing
##  [1] B A B A A E D B A A B C B A E E A B B B
## Levels: A B C D E
```

**Model 4 : Linear Discriminant Analysis**

```
model_lda <- train(classe ~ ., data=training_reduced, model="lda", trControl=ctrl)
pred_lda <- predict(model_lda, validation_reduced[,-35])
# confusion matrix
cm_lda <- confusionMatrix(validation_reduced$classe, pred_lda)
cm_lda
## Confusion Matrix and Statistics
##
##           Reference
## Prediction    A    B    C    D    E
##          A 1390    2    2    0    1
##          B    4  942    3    0    0
##          C    0    4  845    6    0
##          D    1    0   11  791    1
##          E    0    0    0    1  900
##
## Overall Statistics
##
##                Accuracy : 0.9927
##                  95% CI : (0.9899, 0.9949)
##     No Information Rate : 0.2845
##     P-Value [Acc > NIR] : < 2.2e-16
##
##                   Kappa : 0.9907
##  Mcnemar's Test P-Value : NA
##
## Statistics by Class:
##
```

```
##                      Class: A Class: B Class: C Class: D Class: E
## Sensitivity            0.9964   0.9937   0.9814   0.9912   0.9978
## Specificity            0.9986   0.9982   0.9975   0.9968   0.9998
## Pos Pred Value         0.9964   0.9926   0.9883   0.9838   0.9989
## Neg Pred Value         0.9986   0.9985   0.9960   0.9983   0.9995
## Prevalence             0.2845   0.1933   0.1756   0.1627   0.1839
## Detection Rate         0.2834   0.1921   0.1723   0.1613   0.1835
## Detection Prevalence   0.2845   0.1935   0.1743   0.1639   0.1837
## Balanced Accuracy      0.9975   0.9960   0.9895   0.9940   0.9988
# accuracy
cm_lda$overall['Accuracy']
##  Accuracy
## 0.9926591
pred_lda_testing <- predict( model_lda, test)
pred_lda_testing
##  [1] B A B A A E D B A A B C B A E E A B B B
## Levels: A B C D E
```

**Model 5 : Support Vector Machines**

```
system.time(model_svm <- train(classe ~ ., data=training_reduced, model="svm",
trControl=ctrl))
##    user  system elapsed
## 537.656   5.240 542.984
pred_svm <- predict(model_svm, validation_reduced[,-35])
# confusion matrix
cm_svm <- confusionMatrix(validation_reduced$classe, pred_svm)
cm_svm
## Confusion Matrix and Statistics
##
##           Reference
## Prediction    A    B    C    D    E
##          A 1389    3    2    0    1
##          B    4  942    3    0    0
##          C    0    2  848    5    0
##          D    2    0   10  791    1
##          E    0    0    0    2  899
##
## Overall Statistics
##
##                Accuracy : 0.9929
##                  95% CI : (0.9901, 0.995)
##     No Information Rate : 0.2845
##     P-Value [Acc > NIR] : < 2.2e-16
##
##                   Kappa : 0.991
##  Mcnemar's Test P-Value : NA
##
## Statistics by Class:
##
```

```
##                      Class: A Class: B Class: C Class: D Class: E
## Sensitivity            0.9957   0.9947   0.9826   0.9912   0.9978
## Specificity            0.9983   0.9982   0.9983   0.9968   0.9995
## Pos Pred Value         0.9957   0.9926   0.9918   0.9838   0.9978
## Neg Pred Value         0.9983   0.9987   0.9963   0.9983   0.9995
## Prevalence             0.2845   0.1931   0.1760   0.1627   0.1837
## Detection Rate         0.2832   0.1921   0.1729   0.1613   0.1833
## Detection Prevalence   0.2845   0.1935   0.1743   0.1639   0.1837
## Balanced Accuracy      0.9970   0.9965   0.9904   0.9940   0.9986
# accuracy
cm_svm$overall['Accuracy']
## Accuracy
## 0.992863
pred_svm_testing <- predict( model_svm, test)
pred_svm_testing
##  [1] B A B A A E D B A A B C B A E E A B B B
## Levels: A B C D E
```

## d. Make conclusions
## Ans

Using Decision trees results in poor performance with an accuracy of just about 50%. Random forests, linear discriminant analysis, gradient boosted machine and support vector machines fare very well with all of them yielding an out of sample accuracy of about 99% on the validation set. Hence the out of sample error rate with five fold cross-validation is about 1%.

We then use rf, lda, lda and svm models to make predictions on the testing dataset. All these four models correctly predict the class (A,B,C,D,E) for all the 20 test cases.

## e. Plot importance of variables
## Ans