Machine Learning Prediction Project

August 16, 2017

Summary

We have investigated two models or methods, classification tree and random forest tree, to find out that which method has a better accuracy to predict outcomes. To demonstrate this, both modes were tested on 30 % data of training set for cross validation of predictions. It has been observed that random forest tree provides bettre accuracy 99 % (0.99) than classification tree method with 49 % (0.49) accuracy. Random forest tree model was used to prdict the outcomes of test data set.

Loading libraries and data sets

```
library(caret);library(rpart);library(randomForest)
training <- read.csv("pml-training.csv", na.strings = c("NA", ""))
testing <- read.csv("pml-testing.csv", na.strings = c("NA", ""))
```

Data cleaning

In this step, columns with missing values have been removed from both the test and train data sets, and also variables those are not important or required for the model calculations have been delted form the data sets. This process reduces the calculation time for model and make it computationally efficent.

```
training <- training[ ,-c(1:7)]
new.training <- training[, colSums(is.na(training)) == 0]

testing <- testing[ ,-c(1:7)]
testing <- testing[, colSums(is.na(testing)) == 0]
new.testing <- testing2[ ,-53]
```

Data partitioned

After cleaning the data, the training data set is divided into two data sets, (70% training.data and 30% valid.training.data) for training and prediction with models used in this data analysis.

```
inTrain <- createDataPartition(new.training$classe, p = 0.7, list =FALSE)
training.data <- new.training[inTrain,]
Valid.training.data <- new.training[-inTrain,]
```

1. Prediction with clssification tress using validation data set

We used trainControl functon with K=5 fold and rpart method for cross validation of predictions

```
Control<- trainControl(method = "cv", number = 5)
Fit.Model <- train(classe ~.,method = "rpart", data =training.data,trControl = Control)
print(Fit.Model, digit=4)

CART
```

```
13737 samples
   52 predictor
    5 classes: 'A', 'B', 'C', 'D', 'E'

No pre-processing
Resampling: Cross-Validated (5 fold)
Summary of sample sizes: 10989, 10989, 10991, 10989, 10990
Resampling results across tuning parameters:

  cp       Accuracy  Kappa
  0.03428  0.5097    0.3601
  0.06076  0.4155    0.2077
  0.11647  0.3328    0.0738

Accuracy was used to select
 the optimal model using  the
 largest value.
The final value used for the model
 was cp = 0.03428.
```

```
prediction <- predict(Fit.Model, Valid.training.data)
classification <- confusionMatrix(prediction, Valid.training.data$classe)
classification
```

```
Confusion Matrix and Statistics
          Reference
Prediction    A    B    C    D    E
         A 1509  454  460  448  155
         B   41  383   32  163  147
         C  118  302  534  353  302
         D    0    0    0    0    0
         E    6    0    0    0  478

Overall Statistics

               Accuracy : 0.4935
                 95% CI : (0.4806, 0.5063)
    No Information Rate : 0.2845
    P-Value [Acc > NIR] : < 2.2e-16

                  Kappa : 0.3385
 Mcnemar's Test P-Value : NA
Statistics by Class:

                     Class:A  Class:B  Class:C  Class:D  Class:E
Sensitivity           0.9014  0.33626  0.52047   0.0000  0.44177
Specificity           0.6398  0.91930  0.77876   1.0000  0.99875
Pos Pred Value        0.4987  0.50000  0.33188      NaN  0.98760
Neg Pred Value        0.9423  0.85231  0.88494   0.8362  0.88817
Prevalence            0.2845  0.19354  0.17434   0.1638  0.18386
Detection Rate        0.2564  0.06508  0.09074   0.0000  0.08122
Detection Prevalence  0.5142  0.13016  0.27341   0.0000  0.08224
Balanced Accuracy     0.7706  0.62778  0.64961   0.5000  0.72026
```

This method has a 50 % acuracy and out of sample error is 0.50 (50 %). This classification tree doesn't do good job in prediction of outcomes. Next, we will experiment with random forest tree to obtain better accuracy in prediction than calssification tree method.

2. Prediction with random forest tree using validation data set

```
Fit.Model2 <- train(classe ~.,method = "rf", data =training.data,trControl = Control)
prediction2 <- predict(Fit.Model2, Valid.training.data)
print(Fit.Model2, digit=4)
```

```
Random Forest

13737 samples
   52 predictor
    5 classes: 'A', 'B', 'C', 'D', 'E'

No pre-processing
Resampling: Cross-Validated (5 fold)
Summary of sample sizes: 10990, 10990, 10989, 10989, 10990
Resampling results across tuning parameters:

  mtry  Accuracy  Kappa
   2    0.9903    0.9878
  27    0.9909    0.9885
  52    0.9815    0.9766

Accuracy was used to select
 the optimal model using  the
 largest value.
The final value used for the model
 was mtry = 27.
```

```
classification2 <- confusionMatrix(prediction2, Valid.training.data$classe)
classification2
```

```
Confusion Matrix and Statistics

          Reference
Prediction    A    B    C    D    E
         A 1672    1    0    0    0
         B    2 1135    3    0    0
         C    0    3 1021   13    1
         D    0    0    2  950    6
         E    0    0    0    1 1075

Overall Statistics

               Accuracy : 0.9946
                 95% CI : (0.9923, 0.9963)
    No Information Rate : 0.2845
    P-Value [Acc > NIR] : < 2.2e-16

                  Kappa : 0.9931
```

```
 Mcnemar's Test P-Value : NA

Statistics by Class:

                     Class: A Class: B Class: C Class: D Class: E
Sensitivity            0.9988   0.9965   0.9951   0.9855   0.9935
Specificity            0.9998   0.9989   0.9965   0.9984   0.9998
Pos Pred Value         0.9994   0.9956   0.9836   0.9916   0.9991
Neg Pred Value         0.9995   0.9992   0.9990   0.9972   0.9985
Prevalence             0.2845   0.1935   0.1743   0.1638   0.1839
Detection Rate         0.2841   0.1929   0.1735   0.1614   0.1827
Detection Prevalence   0.2843   0.1937   0.1764   0.1628   0.1828
Balanced Accuracy      0.9993   0.9977   0.9958   0.9919   0.9967
```

Random forest tree model has a accuracy of 0.99 and out of sampleor err 0.01 that is far better than classification tree method. We will use this model or method to predict outcomes of 20 test data sets.

3. Prediction with random forest tree using test data set

```
prediction3<- predict(Fit.Model2, new.testing)
prediction3
```