Practical Machine Learning Project

Preparation

Load packages

```
library(lattice);
library(ggplot2);
```

```
## Warning: package 'ggplot2' was built under R version 3.1.3
```

```
library(caret);
```

```
## Warning: package 'caret' was built under R version 3.1.3
```

```
library(randomForest);
```

```
## Warning: package 'randomForest' was built under R version 3.1.1
```

```
## randomForest 4.6-10
## Type rfNews() to see new features/changes/bug fixes.
```

```
library(rpart);
library(rpart.plot);
```

```
## Warning: package 'rpart.plot' was built under R version 3.1.2
```

Load data

```
train <- read.csv("pml-training.csv", na.strings=c("NA","#DIV/0!",
""))
test <- read.csv("pml-testing.csv", na.strings=c("NA","#DIV/0!",
""))
```

Clean data

Remove columns with missing values and irrelevant columns

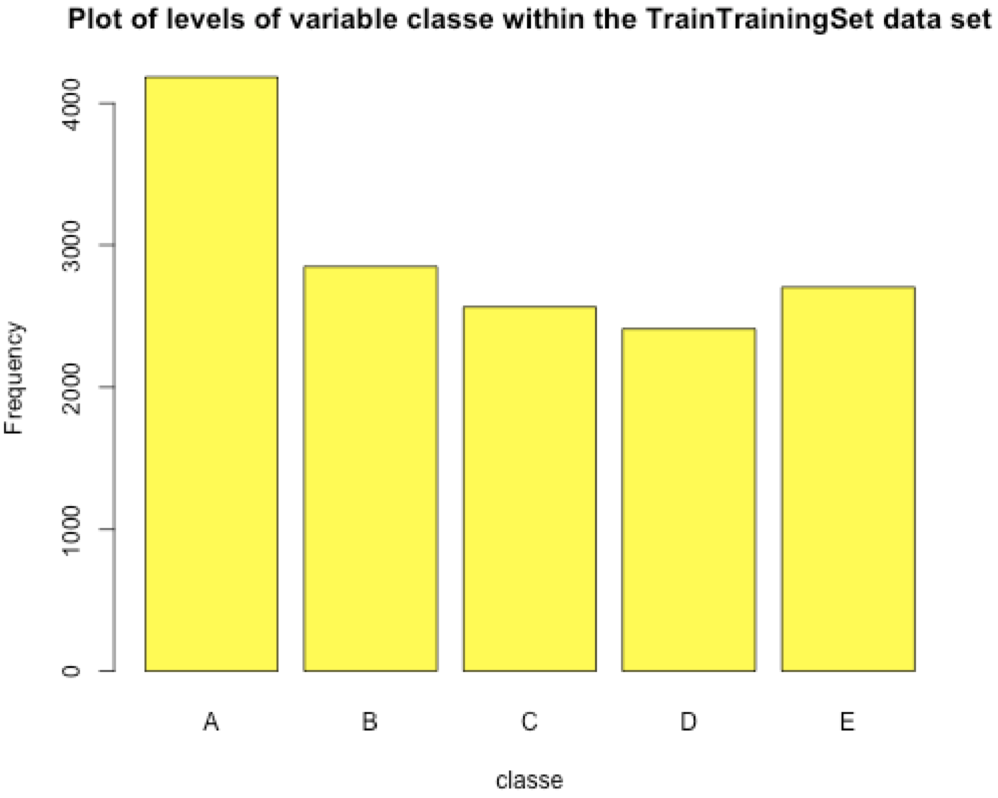
```
train <-train[,colSums(is.na(train)) == 0]
train<-train[,-c(1:7)]
test <-test[,colSums(is.na(test)) == 0]
test<-test[,-c(1:7)]
```

Partition the data into 75% training datset and 25% testing dataset

```
train.train<- createDataPartition(y=train$classe, p=0.75,
list=FALSE)
TrainTrainingSet <- train[train.train, ]
TestTrainingSet <- train[-train.train, ]
```

Exploratory analysis

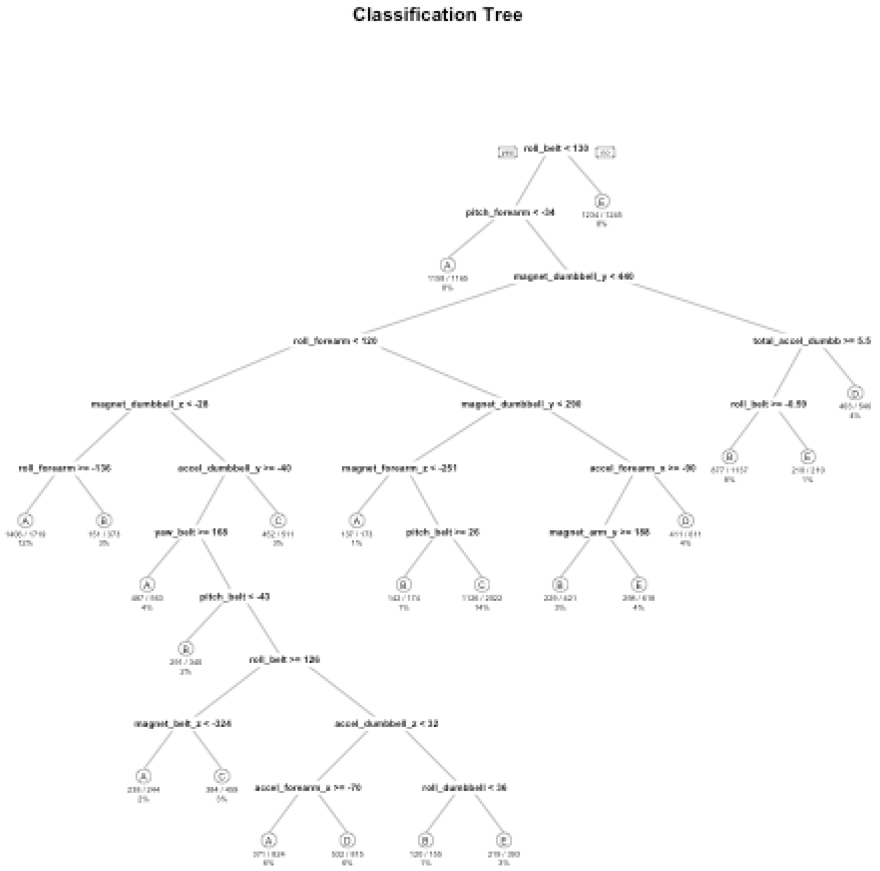
```
plot(TrainTrainingSet$classe, col="yellow", main="Plot of levels
of variable classe within the TrainTrainingSet data set",
xlab="classe", ylab="Frequency")
```



The plot shows that Level A is the most frequent while level D is the least frequent.

Model 1: Decision Tree

```
model1 <- rpart(classe ~ ., data=TrainTrainingSet, method="class")
prediction1 <- predict(model1, TestTrainingSet, type = "class")
rpart.plot(model1, main="Classification Tree", extra=102,
under=TRUE, faclen=0)
```



Test results

```
confusionMatrix(prediction1, TestTrainingSet$classe)
```

```
## Confusion Matrix and Statistics
## 
##           Reference
## Prediction    A    B    C    D    E
##          A 1254  172   74  101   28
##          B   46  600   67   67   81
##          C   36   68  632  133  107
##          D   34   65   60  439   46
##          E   25   44   22   64  639
## 
## Overall Statistics
## 
##                Accuracy : 0.7268          
##                  95% CI : (0.714, 0.7392)
##     No Information Rate : 0.2845          
##     P-Value [Acc > NIR] : < 2.2e-16       
## 
##                   Kappa : 0.6522          
##  Mcnemar's Test P-Value : < 2.2e-16       
## 
## Statistics by Class:
## 
##                      Class: A Class: B Class: C Class: D Class:
E
## Sensitivity            0.8989   0.6322   0.7392  0.54602
0.7092
## Specificity            0.8931   0.9340   0.9150  0.95000
0.9613
## Pos Pred Value         0.7698   0.6969   0.6475  0.68168
0.8048
## Neg Pred Value         0.9569   0.9137   0.9432  0.91432
0.9363
## Prevalence             0.2845   0.1935   0.1743  0.16395
0.1837
## Detection Rate         0.2557   0.1223   0.1289  0.08952
0.1303
## Detection Prevalence   0.3322   0.1756   0.1990  0.13132
0.1619
## Balanced Accuracy      0.8960   0.7831   0.8271  0.74801
0.8352
```

Model 2: Random Forest

```
model2 <- randomForest(classe ~. , data=TrainTrainingSet,
method="class")
```

Predicting:

```
prediction2 <- predict(model2, TestTrainingSet, type = "class")
```

Test results on TestTrainingSet data set:

```
confusionMatrix(prediction2, TestTrainingSet$classe)
```

```
## Confusion Matrix and Statistics
## 
##           Reference
## Prediction    A    B    C    D    E
##          A 1394    3    0    0    0
##          B    1  945    2    0    0
##          C    0    1  852    3    0
##          D    0    0    1  801    6
##          E    0    0    0    0  895
## 
## Overall Statistics
## 
##                Accuracy : 0.9965
##                  95% CI : (0.9945, 0.998)
##     No Information Rate : 0.2845
##     P-Value [Acc > NIR] : < 2.2e-16
## 
##                   Kappa : 0.9956
##  Mcnemar's Test P-Value : NA
## 
## Statistics by Class:
## 
##                      Class: A Class: B Class: C Class: D Class:
E
## Sensitivity            0.9993   0.9958   0.9965   0.9963
0.9933
## Specificity            0.9991   0.9992   0.9990   0.9983
1.0000
## Pos Pred Value         0.9979   0.9968   0.9953   0.9913
1.0000
## Neg Pred Value         0.9997   0.9990   0.9993   0.9993
0.9985
## Prevalence             0.2845   0.1935   0.1743   0.1639
0.1837
## Detection Rate         0.2843   0.1927   0.1737   0.1633
0.1825
## Detection Prevalence   0.2849   0.1933   0.1746   0.1648
0.1825
## Balanced Accuracy      0.9992   0.9975   0.9978   0.9973
0.9967
```

Conclusion

Random Forest is chosen because accuracy for Random Forest model was 0.995 which is higher than the 0.739 (95% CI: (0.727, 0.752)) of Decision Tree model. The expected out-of-sample error is estimated at 0.5%.

Submission

Final outcome of Random Forest Model applied to the testing data.

```
predict(model2, test, type="class")
```

```
##  1  2  3  4  5  6  7  8  9 10 11 12 13 14 15 16 17 18 19 20 
##  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
```