Prediction Assignment Writeup
Xiaoqin Jin

1. ## Libraries

```
library(“caret”)
library(“rpart”)
library(“tree”)
library(“randomForest”)
```

2. ## Load Data

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

3. ## Tidy the Data

Remove variables which have an excess number of NA values.

```
> nacounts <- colSums(is.na(training))
> table(nacounts)
nacounts
    0 19216 19217 19218 19220 19221 19225 19226 19227 19248 19293 19294
   60    67     1     1     1     4     1     4     2     2     1
1
19296 19299 19300 19301 19622
    2     1     4     2     6
> training1<- training[nacounts == 0]
> dim(training1)
[1] 19622    60
>     extraneous<-c('X',     'user_name',     'raw_timestamp_part_1',
'raw_timestamp_part_2', 'cvtd_timestamp', 'new_window', 'num_window')
> training1_ext <- training1[, -which(names(training1) %in% extraneous)]
> dim(training1_ext)
[1] 19622    53
```

4. ## Explore and Preprocess Data
Check low variance variables.

```
>       near_ZV=       nearZeroVar(training1_ext[sapply(training1_ext,
is.numeric)], saveMetrics = TRUE)
> train_nzv = training1_ext[,near_ZV[, 'nzv']==0]
> dim(train_nzv)
[1] 19622    53
##  Remove variables with high correlation
> cor_matrix <- cor(na.omit(train_nzv[sapply(train_nzv, is.numeric)]))
> remove_corr<-findCorrelation(cor_matrix, cutoff = .90, verbose = TRUE)
Compare row 10  and column  1 with corr  0.992
  Means:  0.27 vs 0.168 so flagging column 10
Compare row 1  and column  9 with corr  0.925
  Means:  0.25 vs 0.164 so flagging column 1
Compare row 9  and column  4 with corr  0.928
  Means:  0.233 vs 0.161 so flagging column 9
Compare row 8  and column  2 with corr  0.966
  Means:  0.245 vs 0.157 so flagging column 8
Compare row 19  and column  18 with corr  0.918
  Means:  0.091 vs 0.158 so flagging column 18
Compare row 46  and column  31 with corr  0.914
  Means:  0.101 vs 0.161 so flagging column 31
Compare row 46  and column  33 with corr  0.933
  Means:  0.083 vs 0.164 so flagging column 33
All correlations <= 0.9
> training_corr<-train_nzv[,-remove_corr]
> dim(training_corr)
[1] 19622    46
```

5. ## Cross Validation

```
>  training_dataPart  <-  createDataPartition(y=training_corr$classe,
p=0.7, list=FALSE)
>  training 1 <-  training_corr[training_dataPart,];  testing 1   <-
training_corr[-training_dataPart,]
> dim(training 1 )
[1] 13737    46
> dim(testing 1 )
[1] 5885   46
> set.seed(5555555)
> tree_training = tree(classe~.,data=training 1 )
> tree_prune <- train(classe ~ .,method="rpart",data=training 1 )
> tree_predict = predict(tree_prune,testing 1 )
> pred_matrix = with(testing 1 ,table(tree_predict,classe))
> sum(diag(pred_matrix))/sum(as.vector(pred_matrix))
[1] 0.4927782
```

```
> set.seed(555555)
> training_ranForest = randomForest(classe~.,data=training1,ntree=100,
importance=TRUE)
> training_ranForest
Call:
 randomForest(formula = classe ~ ., data = training1, ntree = 100,
importance = TRUE)
               Type of random forest: classification
                     Number of trees: 100
No. of variables tried at each split: 6
        OOB estimate of  error rate: 0.6%
Confusion matrix:
     A    B    C    D    E class.error
A 3901    4    0    0    1 0.001280082
B   14 2637    7    0    0 0.007900677
C    1   13 2379    3    0 0.007095159
D    0    0   31 2219    2 0.014653641
E    0    0    1    6 2518 0.002772277
> tree_predict = predict(training_ranForest,testing1,type="class")
> pred_matrix = with(testing1,table(tree_predict,classe))
> sum(diag(pred_matrix))/sum(as.vector(pred_matrix))
[1] 0.9937128
```

6. ## Conclusion

```
> predict <- predict(training_ranForest, test)
> predict
 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
```

Inserting this data in the Quiz, it provides a 100% scoring