Practical Machine Learning Prediction Assignment

```
pml.testing <- read.csv("pml-testing.csv")
pml.training <- read.csv("pml-training.csv")
```

Data allocation strategy:

- Use 80% of the data for training
- Use 10% of the data for cross-validation
- Use 10% of the data for testing –This set will only be used once

There are several alrogithms to choose from when it comes to classification problems. For this assignment, I have considered following two models:

- Linear discriminant analysis (LDA)
- Quadratic discriminant analysis (QDA)

To decide which algorithm performs better on assignment data, I performed exploratory analysis, which can be summarized as follows:

- Start with all the features that are available for all the training subjects
- Train one predictor with LDA and one with QDA
- Pick the model that outperforms, if both models are equally good pick LDA as it is simpler than QDA
- Perform feature selection on the model that I picked

Now that I know which analysis algorithm I will pursue, I will perform feature selection to reduce the number of features without compromising the prediction accuracy to obtain simpler model for better interpretability. Note that there are 159 features in the training data.

```
library(caret);
```

```
## Loading required package: lattice
## Loading required package: ggplot2
```

```
library(kernlab);
library(MASS);
```

```
pml.testing <- read.csv("pml-testing.csv")
pml.training <- read.csv("pml-training.csv")

inTraincv=createDataPartition(y=pml.training$classe,p=0.90,list=FALSE)
training_and_cv=pml.training[inTraincv,]
inTrain=createDataPartition(y=training_and_cv$classe,p=80/90.0,list=FALSE)
training =training_and_cv[inTrain,]
crossval = training_and_cv[-inTrain,]

testing=pml.training[-inTraincv,]
#fit=qda(classe ~ num_window + <others >, data=pml.training, subset=inTrain)
```

- LDA cross validation accuracy: 0.7046
- QDA cross validation accuracy: 0.9051

QDA outperformed LDA, hence, it is better fit than LDA for this data set.

```
qda_lean_pred=predict(ft,crossval)
qda_accuracy_after_fs=mean(qda_lean_pred$class==crossval$classe,na.rm=TRUE)
```

Pruning the model: * Remove a feature * calculate the accurancy on cross validation * If accuracy does not decrease, remove the feature permanently, otherwise put the feature back in to model. * Repeat

After feature selection number of parameters went down from 55 to 46. See appendix initial model and final model. The accuracy is: 0.8985 The error rate is 0.1015, and the expected error rate on test data will be close this value.

```
qda_test_prediction = predict(ft,testing)
qda_accuracy_on_test=mean(qda_test_prediction$class==testing$classe,na.rm=TRUE)
```

Finally, after feature selection is done, classifier accurancy is tested with previously unseen data (test data) * Test Accuracy: 0.8944 * Error: 0.1056

As we can see from error rate from cross-validation set and test set, they are very close to each other.

The confusion matrix is as follows:

```
table(ft.pred$class,testing$classe)
```

```
##
##       A   B   C   D   E
##   A 522  14   0   0   0
##   B  19 320  12   2   9
##   C  10  41 328  45  11
##   D   7   0   2 270   6
##   E   0   4   0   4 334
```