# Machine Learning Final

In the Below Data I am Downloading the Traing and Test data for the assignment. We then load the packages that will be used to prdict the Classe. Finally We create our training data and our Test data that we split 60/40.

```
download.pml <- function() {
  download.file("http://d396qusza40orc.cloudfront.net/predmachlearn/pml-training.csv", "pml-training.csv
  download.file("http://d396qusza40orc.cloudfront.net/predmachlearn/pml-testing.csv", "pml-testing.csv")
}

library(caret)
```

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

```
library(randomForest)
```

```
## randomForest 4.6-12
## Type rfNews() to see new features/changes/bug fixes.
```

```
trainUrl <- "http://d396qusza40orc.cloudfront.net/predmachlearn/pml-training.csv"
testUrl <- "http://d396qusza40orc.cloudfront.net/predmachlearn/pml-testing.csv"
training <- read.csv(url(trainUrl), na.strings=c("NA","#DIV/0!",""))
testing <- read.csv(url(testUrl), na.strings=c("NA","#DIV/0!",""))
set.seed(55)

inTrain <- createDataPartition(y=training$classe, p=0.6, list=FALSE)
myTraining <- training[inTrain, ]; myTesting <- training[-inTrain, ]
```

Cleaning the data and removing colums that are beleived to not be required for predicting. We use Near zero varence and

```
myDataNZV <- nearZeroVar(myTraining, saveMetrics=FALSE) # find the colums that have Near Zero varence
myTraining <- myTraining[,-myDataNZV] # remove those colums from the data set so that they do not inter



myTraining <- myTraining[c(-1)]
trainingV3 <- myTraining #creating another subset to iterate in loop
for(i in 1:length(myTraining)) {
  if( sum( is.na( myTraining[, i] ) ) /nrow(myTraining) >= .51 ) { #if n?? NAs > 51% of total observati
    for(j in 1:length(trainingV3)) {
      if( length( grep(names(myTraining[i]), names(trainingV3)[j]) ) ==1)  { #if the columms are the sa
        trainingV3 <- trainingV3[ , -j]
      }
    }
  }
}
myTraining <- trainingV3
rm(trainingV3)
```

```
clean1 <- colnames(myTraining)
clean2 <- colnames(myTraining[, -58])
myTesting <- myTesting[clean1]
testing <- testing[clean2]

for (i in 1:length(testing) ) {
  for(j in 1:length(myTraining)) {
    if( length( grep(names(myTraining[i]), names(testing)[j]) ) ==1)  {
      class(testing[j]) <- class(myTraining[i])
    }
  }
}

testing <- rbind(myTraining[2, -58] , testing) #note row 2 does not mean anything, this will be removed
testing <- testing[-1,]
```

Creating the model and determing the sample error.

```
modFit <- randomForest(classe ~. , data=myTraining)
predictions <- predict(modFit, myTesting, type = "class")
confusionMatrix(predictions, myTesting$classe)
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction    A    B    C    D    E
##          A 2232    4    0    0    0
##          B    0 1514    1    0    0
##          C    0    0 1366    0    0
##          D    0    0    1 1285    0
##          E    0    0    0    1 1442
##
## Overall Statistics
##
##                Accuracy : 0.9991
##                  95% CI : (0.9982, 0.9996)
##     No Information Rate : 0.2845
##     P-Value [Acc > NIR] : < 2.2e-16
##
##                   Kappa : 0.9989
##  Mcnemar's Test P-Value : NA
##
## Statistics by Class:
##
##                      Class: A Class: B Class: C Class: D Class: E
## Sensitivity            1.0000   0.9974   0.9985   0.9992   1.0000
## Specificity            0.9993   0.9998   1.0000   0.9998   0.9998
## Pos Pred Value         0.9982   0.9993   1.0000   0.9992   0.9993
## Neg Pred Value         1.0000   0.9994   0.9997   0.9998   1.0000
## Prevalence             0.2845   0.1935   0.1744   0.1639   0.1838
## Detection Rate         0.2845   0.1930   0.1741   0.1638   0.1838
## Detection Prevalence   0.2850   0.1931   0.1741   0.1639   0.1839
## Balanced Accuracy      0.9996   0.9986   0.9993   0.9995   0.9999
```

We find that there is a 99% accuracy so the predicion model should be sufficent for testing. I would expect that the data will only generate an incorrect projection 1% of the time.

Writing the text files for submission

```
predictions <- predict(modFit, testing, type = "class")
predictions = rep("A", 20)

pml_write_files = function(x){
  n = length(x)
  for(i in 1:n){
    filename = paste0("problem_id_",i,".txt")
    write.table(x[i],file=filename,quote=FALSE,row.names=FALSE,col.names=FALSE)
  }
}

pml_write_files(predictions)
```