Project #2

J.Bonacci, Stefan Jonsson, James Adler, Nikki Barden

5/15/2018

remove(list=ls())

## R Markdown

## Nikki’s method consistently performed the best so we chose to go with hers. We tried to cut our predictors by making the non zero variance function stricter but it actually hurt our prediction out of sample. We chose not to validate the data because it did not improve the random forest model much or at all. Nikki\_finalpred is the final output for the out of sample test set.

## Libraries

library('ddalpha')

## Warning: package 'ddalpha' was built under R version 3.4.4

## Loading required package: MASS

## Loading required package: class

## Loading required package: robustbase

## Loading required package: sfsmisc

## Warning: package 'sfsmisc' was built under R version 3.4.4

## Loading required package: geometry

## Loading required package: magic

## Loading required package: abind

library('kernlab')  
library('caret')

## Warning: package 'caret' was built under R version 3.4.4

## Loading required package: lattice

## Loading required package: ggplot2

##   
## Attaching package: 'ggplot2'

## The following object is masked from 'package:kernlab':  
##   
## alpha

library('MASS')  
library('randomForest')

## Warning: package 'randomForest' was built under R version 3.4.4

## randomForest 4.6-14

## Type rfNews() to see new features/changes/bug fixes.

##   
## Attaching package: 'randomForest'

## The following object is masked from 'package:ggplot2':  
##   
## margin

library("dplyr")

##   
## Attaching package: 'dplyr'

## The following object is masked from 'package:randomForest':  
##   
## combine

## The following object is masked from 'package:sfsmisc':  
##   
## last

## The following object is masked from 'package:MASS':  
##   
## select

## The following objects are masked from 'package:stats':  
##   
## filter, lag

## The following objects are masked from 'package:base':  
##   
## intersect, setdiff, setequal, union

library("e1071")  
library('rpart')  
library('rpart.plot')  
library('neuralnet')

##   
## Attaching package: 'neuralnet'

## The following object is masked from 'package:dplyr':  
##   
## compute

library('nnet')  
library('ggvis')

##   
## Attaching package: 'ggvis'

## The following object is masked from 'package:ggplot2':  
##   
## resolution

library('class')  
library('gmodels')

# Reading in the datasets

train\_data <- read.csv('https://raw.githubusercontent.com/slevkoff/ECON386REPO/master/Prediction%20Project/training.csv')  
test\_data <- read.csv('https://raw.githubusercontent.com/slevkoff/ECON386REPO/master/Prediction%20Project/testing.csv')

## Count NA values

sapply(train\_data, function(x) sum(is.na(x)))

## X user\_name raw\_timestamp\_part\_1   
## 0 0 0   
## raw\_timestamp\_part\_2 cvtd\_timestamp new\_window   
## 0 0 0   
## num\_window roll\_belt pitch\_belt   
## 0 0 0   
## yaw\_belt total\_accel\_belt kurtosis\_roll\_belt   
## 0 0 0   
## kurtosis\_picth\_belt kurtosis\_yaw\_belt skewness\_roll\_belt   
## 0 0 0   
## skewness\_roll\_belt.1 skewness\_yaw\_belt max\_roll\_belt   
## 0 0 19216   
## max\_picth\_belt max\_yaw\_belt min\_roll\_belt   
## 19216 0 19216   
## min\_pitch\_belt min\_yaw\_belt amplitude\_roll\_belt   
## 19216 0 19216   
## amplitude\_pitch\_belt amplitude\_yaw\_belt var\_total\_accel\_belt   
## 19216 0 19216   
## avg\_roll\_belt stddev\_roll\_belt var\_roll\_belt   
## 19216 19216 19216   
## avg\_pitch\_belt stddev\_pitch\_belt var\_pitch\_belt   
## 19216 19216 19216   
## avg\_yaw\_belt stddev\_yaw\_belt var\_yaw\_belt   
## 19216 19216 19216   
## gyros\_belt\_x gyros\_belt\_y gyros\_belt\_z   
## 0 0 0   
## accel\_belt\_x accel\_belt\_y accel\_belt\_z   
## 0 0 0   
## magnet\_belt\_x magnet\_belt\_y magnet\_belt\_z   
## 0 0 0   
## roll\_arm pitch\_arm yaw\_arm   
## 0 0 0   
## total\_accel\_arm var\_accel\_arm avg\_roll\_arm   
## 0 19216 19216   
## stddev\_roll\_arm var\_roll\_arm avg\_pitch\_arm   
## 19216 19216 19216   
## stddev\_pitch\_arm var\_pitch\_arm avg\_yaw\_arm   
## 19216 19216 19216   
## stddev\_yaw\_arm var\_yaw\_arm gyros\_arm\_x   
## 19216 19216 0   
## gyros\_arm\_y gyros\_arm\_z accel\_arm\_x   
## 0 0 0   
## accel\_arm\_y accel\_arm\_z magnet\_arm\_x   
## 0 0 0   
## magnet\_arm\_y magnet\_arm\_z kurtosis\_roll\_arm   
## 0 0 0   
## kurtosis\_picth\_arm kurtosis\_yaw\_arm skewness\_roll\_arm   
## 0 0 0   
## skewness\_pitch\_arm skewness\_yaw\_arm max\_roll\_arm   
## 0 0 19216   
## max\_picth\_arm max\_yaw\_arm min\_roll\_arm   
## 19216 19216 19216   
## min\_pitch\_arm min\_yaw\_arm amplitude\_roll\_arm   
## 19216 19216 19216   
## amplitude\_pitch\_arm amplitude\_yaw\_arm roll\_dumbbell   
## 19216 19216 0   
## pitch\_dumbbell yaw\_dumbbell kurtosis\_roll\_dumbbell   
## 0 0 0   
## kurtosis\_picth\_dumbbell kurtosis\_yaw\_dumbbell skewness\_roll\_dumbbell   
## 0 0 0   
## skewness\_pitch\_dumbbell skewness\_yaw\_dumbbell max\_roll\_dumbbell   
## 0 0 19216   
## max\_picth\_dumbbell max\_yaw\_dumbbell min\_roll\_dumbbell   
## 19216 0 19216   
## min\_pitch\_dumbbell min\_yaw\_dumbbell amplitude\_roll\_dumbbell   
## 19216 0 19216   
## amplitude\_pitch\_dumbbell amplitude\_yaw\_dumbbell total\_accel\_dumbbell   
## 19216 0 0   
## var\_accel\_dumbbell avg\_roll\_dumbbell stddev\_roll\_dumbbell   
## 19216 19216 19216   
## var\_roll\_dumbbell avg\_pitch\_dumbbell stddev\_pitch\_dumbbell   
## 19216 19216 19216   
## var\_pitch\_dumbbell avg\_yaw\_dumbbell stddev\_yaw\_dumbbell   
## 19216 19216 19216   
## var\_yaw\_dumbbell gyros\_dumbbell\_x gyros\_dumbbell\_y   
## 19216 0 0   
## gyros\_dumbbell\_z accel\_dumbbell\_x accel\_dumbbell\_y   
## 0 0 0   
## accel\_dumbbell\_z magnet\_dumbbell\_x magnet\_dumbbell\_y   
## 0 0 0   
## magnet\_dumbbell\_z roll\_forearm pitch\_forearm   
## 0 0 0   
## yaw\_forearm kurtosis\_roll\_forearm kurtosis\_picth\_forearm   
## 0 0 0   
## kurtosis\_yaw\_forearm skewness\_roll\_forearm skewness\_pitch\_forearm   
## 0 0 0   
## skewness\_yaw\_forearm max\_roll\_forearm max\_picth\_forearm   
## 0 19216 19216   
## max\_yaw\_forearm min\_roll\_forearm min\_pitch\_forearm   
## 0 19216 19216   
## min\_yaw\_forearm amplitude\_roll\_forearm amplitude\_pitch\_forearm   
## 0 19216 19216   
## amplitude\_yaw\_forearm total\_accel\_forearm var\_accel\_forearm   
## 0 0 19216   
## avg\_roll\_forearm stddev\_roll\_forearm var\_roll\_forearm   
## 19216 19216 19216   
## avg\_pitch\_forearm stddev\_pitch\_forearm var\_pitch\_forearm   
## 19216 19216 19216   
## avg\_yaw\_forearm stddev\_yaw\_forearm var\_yaw\_forearm   
## 19216 19216 19216   
## gyros\_forearm\_x gyros\_forearm\_y gyros\_forearm\_z   
## 0 0 0   
## accel\_forearm\_x accel\_forearm\_y accel\_forearm\_z   
## 0 0 0   
## magnet\_forearm\_x magnet\_forearm\_y magnet\_forearm\_z   
## 0 0 0   
## classe   
## 0

sapply(test\_data, function(x) sum(is.na(x)))

## X user\_name raw\_timestamp\_part\_1   
## 0 0 0   
## raw\_timestamp\_part\_2 cvtd\_timestamp new\_window   
## 0 0 0   
## num\_window roll\_belt pitch\_belt   
## 0 0 0   
## yaw\_belt total\_accel\_belt kurtosis\_roll\_belt   
## 0 0 20   
## kurtosis\_picth\_belt kurtosis\_yaw\_belt skewness\_roll\_belt   
## 20 20 20   
## skewness\_roll\_belt.1 skewness\_yaw\_belt max\_roll\_belt   
## 20 20 20   
## max\_picth\_belt max\_yaw\_belt min\_roll\_belt   
## 20 20 20   
## min\_pitch\_belt min\_yaw\_belt amplitude\_roll\_belt   
## 20 20 20   
## amplitude\_pitch\_belt amplitude\_yaw\_belt var\_total\_accel\_belt   
## 20 20 20   
## avg\_roll\_belt stddev\_roll\_belt var\_roll\_belt   
## 20 20 20   
## avg\_pitch\_belt stddev\_pitch\_belt var\_pitch\_belt   
## 20 20 20   
## avg\_yaw\_belt stddev\_yaw\_belt var\_yaw\_belt   
## 20 20 20   
## gyros\_belt\_x gyros\_belt\_y gyros\_belt\_z   
## 0 0 0   
## accel\_belt\_x accel\_belt\_y accel\_belt\_z   
## 0 0 0   
## magnet\_belt\_x magnet\_belt\_y magnet\_belt\_z   
## 0 0 0   
## roll\_arm pitch\_arm yaw\_arm   
## 0 0 0   
## total\_accel\_arm var\_accel\_arm avg\_roll\_arm   
## 0 20 20   
## stddev\_roll\_arm var\_roll\_arm avg\_pitch\_arm   
## 20 20 20   
## stddev\_pitch\_arm var\_pitch\_arm avg\_yaw\_arm   
## 20 20 20   
## stddev\_yaw\_arm var\_yaw\_arm gyros\_arm\_x   
## 20 20 0   
## gyros\_arm\_y gyros\_arm\_z accel\_arm\_x   
## 0 0 0   
## accel\_arm\_y accel\_arm\_z magnet\_arm\_x   
## 0 0 0   
## magnet\_arm\_y magnet\_arm\_z kurtosis\_roll\_arm   
## 0 0 20   
## kurtosis\_picth\_arm kurtosis\_yaw\_arm skewness\_roll\_arm   
## 20 20 20   
## skewness\_pitch\_arm skewness\_yaw\_arm max\_roll\_arm   
## 20 20 20   
## max\_picth\_arm max\_yaw\_arm min\_roll\_arm   
## 20 20 20   
## min\_pitch\_arm min\_yaw\_arm amplitude\_roll\_arm   
## 20 20 20   
## amplitude\_pitch\_arm amplitude\_yaw\_arm roll\_dumbbell   
## 20 20 0   
## pitch\_dumbbell yaw\_dumbbell kurtosis\_roll\_dumbbell   
## 0 0 20   
## kurtosis\_picth\_dumbbell kurtosis\_yaw\_dumbbell skewness\_roll\_dumbbell   
## 20 20 20   
## skewness\_pitch\_dumbbell skewness\_yaw\_dumbbell max\_roll\_dumbbell   
## 20 20 20   
## max\_picth\_dumbbell max\_yaw\_dumbbell min\_roll\_dumbbell   
## 20 20 20   
## min\_pitch\_dumbbell min\_yaw\_dumbbell amplitude\_roll\_dumbbell   
## 20 20 20   
## amplitude\_pitch\_dumbbell amplitude\_yaw\_dumbbell total\_accel\_dumbbell   
## 20 20 0   
## var\_accel\_dumbbell avg\_roll\_dumbbell stddev\_roll\_dumbbell   
## 20 20 20   
## var\_roll\_dumbbell avg\_pitch\_dumbbell stddev\_pitch\_dumbbell   
## 20 20 20   
## var\_pitch\_dumbbell avg\_yaw\_dumbbell stddev\_yaw\_dumbbell   
## 20 20 20   
## var\_yaw\_dumbbell gyros\_dumbbell\_x gyros\_dumbbell\_y   
## 20 0 0   
## gyros\_dumbbell\_z accel\_dumbbell\_x accel\_dumbbell\_y   
## 0 0 0   
## accel\_dumbbell\_z magnet\_dumbbell\_x magnet\_dumbbell\_y   
## 0 0 0   
## magnet\_dumbbell\_z roll\_forearm pitch\_forearm   
## 0 0 0   
## yaw\_forearm kurtosis\_roll\_forearm kurtosis\_picth\_forearm   
## 0 20 20   
## kurtosis\_yaw\_forearm skewness\_roll\_forearm skewness\_pitch\_forearm   
## 20 20 20   
## skewness\_yaw\_forearm max\_roll\_forearm max\_picth\_forearm   
## 20 20 20   
## max\_yaw\_forearm min\_roll\_forearm min\_pitch\_forearm   
## 20 20 20   
## min\_yaw\_forearm amplitude\_roll\_forearm amplitude\_pitch\_forearm   
## 20 20 20   
## amplitude\_yaw\_forearm total\_accel\_forearm var\_accel\_forearm   
## 20 0 20   
## avg\_roll\_forearm stddev\_roll\_forearm var\_roll\_forearm   
## 20 20 20   
## avg\_pitch\_forearm stddev\_pitch\_forearm var\_pitch\_forearm   
## 20 20 20   
## avg\_yaw\_forearm stddev\_yaw\_forearm var\_yaw\_forearm   
## 20 20 20   
## gyros\_forearm\_x gyros\_forearm\_y gyros\_forearm\_z   
## 0 0 0   
## accel\_forearm\_x accel\_forearm\_y accel\_forearm\_z   
## 0 0 0   
## magnet\_forearm\_x magnet\_forearm\_y magnet\_forearm\_z   
## 0 0 0   
## problem\_id   
## 0

# Remove all columns containing at least one NA

train\_data2 <- train\_data[ , apply(train\_data, 2, function(x) !any(is.na(x)))]  
test\_data2 <- test\_data[ , apply(test\_data, 2, function(x) !any(is.na(x)))]

# input NAs into all blank observations

train\_data2[train\_data2==""] <- NA  
test\_data2[test\_data2==""] <- NA

# Count NA values again to check

sapply(train\_data2, function(x) sum(is.na(x)))

## X user\_name raw\_timestamp\_part\_1   
## 0 0 0   
## raw\_timestamp\_part\_2 cvtd\_timestamp new\_window   
## 0 0 0   
## num\_window roll\_belt pitch\_belt   
## 0 0 0   
## yaw\_belt total\_accel\_belt kurtosis\_roll\_belt   
## 0 0 19216   
## kurtosis\_picth\_belt kurtosis\_yaw\_belt skewness\_roll\_belt   
## 19216 19216 19216   
## skewness\_roll\_belt.1 skewness\_yaw\_belt max\_yaw\_belt   
## 19216 19216 19216   
## min\_yaw\_belt amplitude\_yaw\_belt gyros\_belt\_x   
## 19216 19216 0   
## gyros\_belt\_y gyros\_belt\_z accel\_belt\_x   
## 0 0 0   
## accel\_belt\_y accel\_belt\_z magnet\_belt\_x   
## 0 0 0   
## magnet\_belt\_y magnet\_belt\_z roll\_arm   
## 0 0 0   
## pitch\_arm yaw\_arm total\_accel\_arm   
## 0 0 0   
## gyros\_arm\_x gyros\_arm\_y gyros\_arm\_z   
## 0 0 0   
## accel\_arm\_x accel\_arm\_y accel\_arm\_z   
## 0 0 0   
## magnet\_arm\_x magnet\_arm\_y magnet\_arm\_z   
## 0 0 0   
## kurtosis\_roll\_arm kurtosis\_picth\_arm kurtosis\_yaw\_arm   
## 19216 19216 19216   
## skewness\_roll\_arm skewness\_pitch\_arm skewness\_yaw\_arm   
## 19216 19216 19216   
## roll\_dumbbell pitch\_dumbbell yaw\_dumbbell   
## 0 0 0   
## kurtosis\_roll\_dumbbell kurtosis\_picth\_dumbbell kurtosis\_yaw\_dumbbell   
## 19216 19216 19216   
## skewness\_roll\_dumbbell skewness\_pitch\_dumbbell skewness\_yaw\_dumbbell   
## 19216 19216 19216   
## max\_yaw\_dumbbell min\_yaw\_dumbbell amplitude\_yaw\_dumbbell   
## 19216 19216 19216   
## total\_accel\_dumbbell gyros\_dumbbell\_x gyros\_dumbbell\_y   
## 0 0 0   
## gyros\_dumbbell\_z accel\_dumbbell\_x accel\_dumbbell\_y   
## 0 0 0   
## accel\_dumbbell\_z magnet\_dumbbell\_x magnet\_dumbbell\_y   
## 0 0 0   
## magnet\_dumbbell\_z roll\_forearm pitch\_forearm   
## 0 0 0   
## yaw\_forearm kurtosis\_roll\_forearm kurtosis\_picth\_forearm   
## 0 19216 19216   
## kurtosis\_yaw\_forearm skewness\_roll\_forearm skewness\_pitch\_forearm   
## 19216 19216 19216   
## skewness\_yaw\_forearm max\_yaw\_forearm min\_yaw\_forearm   
## 19216 19216 19216   
## amplitude\_yaw\_forearm total\_accel\_forearm gyros\_forearm\_x   
## 19216 0 0   
## gyros\_forearm\_y gyros\_forearm\_z accel\_forearm\_x   
## 0 0 0   
## accel\_forearm\_y accel\_forearm\_z magnet\_forearm\_x   
## 0 0 0   
## magnet\_forearm\_y magnet\_forearm\_z classe   
## 0 0 0

sapply(test\_data2, function(x) sum(is.na(x)))

## X user\_name raw\_timestamp\_part\_1   
## 0 0 0   
## raw\_timestamp\_part\_2 cvtd\_timestamp new\_window   
## 0 0 0   
## num\_window roll\_belt pitch\_belt   
## 0 0 0   
## yaw\_belt total\_accel\_belt gyros\_belt\_x   
## 0 0 0   
## gyros\_belt\_y gyros\_belt\_z accel\_belt\_x   
## 0 0 0   
## accel\_belt\_y accel\_belt\_z magnet\_belt\_x   
## 0 0 0   
## magnet\_belt\_y magnet\_belt\_z roll\_arm   
## 0 0 0   
## pitch\_arm yaw\_arm total\_accel\_arm   
## 0 0 0   
## gyros\_arm\_x gyros\_arm\_y gyros\_arm\_z   
## 0 0 0   
## accel\_arm\_x accel\_arm\_y accel\_arm\_z   
## 0 0 0   
## magnet\_arm\_x magnet\_arm\_y magnet\_arm\_z   
## 0 0 0   
## roll\_dumbbell pitch\_dumbbell yaw\_dumbbell   
## 0 0 0   
## total\_accel\_dumbbell gyros\_dumbbell\_x gyros\_dumbbell\_y   
## 0 0 0   
## gyros\_dumbbell\_z accel\_dumbbell\_x accel\_dumbbell\_y   
## 0 0 0   
## accel\_dumbbell\_z magnet\_dumbbell\_x magnet\_dumbbell\_y   
## 0 0 0   
## magnet\_dumbbell\_z roll\_forearm pitch\_forearm   
## 0 0 0   
## yaw\_forearm total\_accel\_forearm gyros\_forearm\_x   
## 0 0 0   
## gyros\_forearm\_y gyros\_forearm\_z accel\_forearm\_x   
## 0 0 0   
## accel\_forearm\_y accel\_forearm\_z magnet\_forearm\_x   
## 0 0 0   
## magnet\_forearm\_y magnet\_forearm\_z problem\_id   
## 0 0 0

# input NAs into all blank observations

train\_data3<- train\_data2[ , apply(train\_data2, 2, function(x) !any(is.na(x)))]  
test\_data3<- test\_data2[ , apply(test\_data2, 2, function(x) !any(is.na(x)))]

# omits zero variance predictors

##freq cut and unique cut arguments can be ommitted if it fits better with out them (leaving arguments in cuts more predictors)  
  
remove\_cols <- nearZeroVar(train\_data3,names=TRUE)  
all\_cols<-names(train\_data3)  
train\_data4<-train\_data3[ , setdiff(all\_cols,remove\_cols)]  
  
  
  
remove\_cols2<-nearZeroVar(test\_data3,names=TRUE)  
all\_cols2<-names(test\_data3)  
test\_data4<-test\_data3[ , setdiff(all\_cols2,remove\_cols2)]

# rename datasets

train <- train\_data4  
test <- test\_data4

# removing timestamps and factor variables

train\_final<- train[c(7:59)]  
test\_final<- test[c(7:59)]

# partitioning data

#partitions 70% of data into training set  
trainingRowIndex<-sample(1:nrow(train\_final), size = .7\*nrow(train\_final))  
part\_training<-train\_final[trainingRowIndex, ]  
  
#leaves 30% for testing and validating   
part\_test <-train\_final[-trainingRowIndex, ]

# removing old datasets

remove(train\_data, train\_data2, train\_data3,train\_data4, train, test\_data, test\_data2, test\_data3, test,test\_data4,train\_final)

# random forest with default number of variables at each node (Jack Bonacci)

set.seed(1234)  
#creates the random forest  
part\_training$classe<- as.factor(part\_training$classe)  
jack\_rf<- randomForest(classe~., part\_training)  
jack\_pred<- predict(jack\_rf, part\_test)  
confusionMatrix(jack\_pred, part\_test$classe, dnn = c("Prediction", "Reference"))

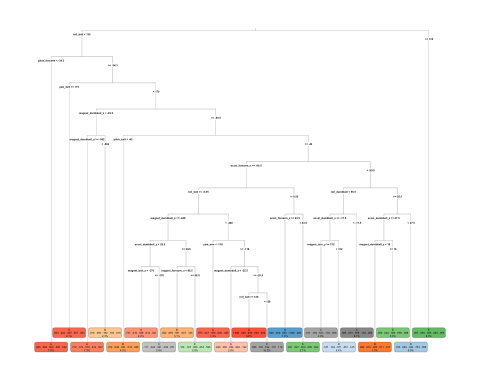
## Confusion Matrix and Statistics  
##   
## Reference  
## Prediction A B C D E  
## A 1687 4 0 0 0  
## B 0 1115 9 0 0  
## C 0 1 1009 13 2  
## D 0 0 2 952 0  
## E 0 0 0 2 1091  
##   
## Overall Statistics  
##   
## Accuracy : 0.9944   
## 95% CI : (0.9921, 0.9961)  
## No Information Rate : 0.2866   
## P-Value [Acc > NIR] : < 2.2e-16   
##   
## Kappa : 0.9929   
## Mcnemar's Test P-Value : NA   
##   
## Statistics by Class:  
##   
## Class: A Class: B Class: C Class: D Class: E  
## Sensitivity 1.0000 0.9955 0.9892 0.9845 0.9982  
## Specificity 0.9990 0.9981 0.9967 0.9996 0.9996  
## Pos Pred Value 0.9976 0.9920 0.9844 0.9979 0.9982  
## Neg Pred Value 1.0000 0.9990 0.9977 0.9970 0.9996  
## Prevalence 0.2866 0.1902 0.1733 0.1643 0.1857  
## Detection Rate 0.2866 0.1894 0.1714 0.1617 0.1853  
## Detection Prevalence 0.2872 0.1909 0.1741 0.1621 0.1857  
## Balanced Accuracy 0.9995 0.9968 0.9930 0.9920 0.9989

#Running algorithm on test data  
jack\_final\_prediction<- predict(jack\_rf, test\_final)  
jack\_final\_prediction

## 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

# cart (James Adler)

#creating cart model   
set.seed(117)  
james\_tree <- rpart(classe ~. ,data = part\_training, method = "class")  
#gets the optimal cp (cp with minimum erro)  
optimal\_cp<-james\_tree$cptable[which.min(james\_tree$cptable[,"xerror"]),"CP"]  
#uses optimal cp to create new tree  
james\_ptree<-prune(james\_tree,optimal\_cp)  
#viewing new tree  
rpart.plot(james\_ptree,type = 3,digits = 3, fallen.leaves = TRUE)



#runs new pruned tree model with partitioned test data  
james\_pred<-predict(james\_ptree, part\_test, type="class")  
confusionMatrix(james\_pred,part\_test$classe)

## Confusion Matrix and Statistics  
##   
## Reference  
## Prediction A B C D E  
## A 1418 201 24 15 32  
## B 20 554 60 66 62  
## C 104 222 877 241 160  
## D 111 108 59 636 64  
## E 34 35 0 9 775  
##   
## Overall Statistics  
##   
## Accuracy : 0.7236   
## 95% CI : (0.712, 0.735)  
## No Information Rate : 0.2866   
## P-Value [Acc > NIR] : < 2.2e-16   
##   
## Kappa : 0.6509   
## Mcnemar's Test P-Value : < 2.2e-16   
##   
## Statistics by Class:  
##   
## Class: A Class: B Class: C Class: D Class: E  
## Sensitivity 0.8405 0.49464 0.8598 0.6577 0.7091  
## Specificity 0.9352 0.95637 0.8506 0.9305 0.9837  
## Pos Pred Value 0.8391 0.72703 0.5468 0.6503 0.9086  
## Neg Pred Value 0.9359 0.88956 0.9666 0.9326 0.9368  
## Prevalence 0.2866 0.19025 0.1733 0.1643 0.1857  
## Detection Rate 0.2409 0.09411 0.1490 0.1080 0.1316  
## Detection Prevalence 0.2871 0.12944 0.2725 0.1661 0.1449  
## Balanced Accuracy 0.8879 0.72550 0.8552 0.7941 0.8464

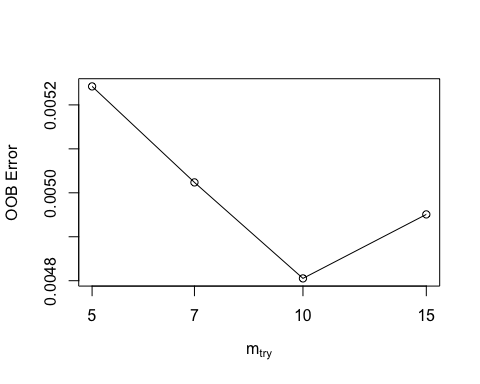
#tests our algorithm on the 20 outputless observations  
james\_final\_prediction<-predict(james\_ptree,test\_final)  
james\_final\_prediction

## A B C D E  
## 1 0.10071942 0.194244604 0.54316547 0.10431655 0.05755396  
## 2 0.48958333 0.302083333 0.04427083 0.04166667 0.12239583  
## 3 0.05553068 0.158978952 0.53246753 0.13703538 0.11598746  
## 4 0.97311828 0.026881720 0.00000000 0.00000000 0.00000000  
## 5 0.72743056 0.210069444 0.02777778 0.01388889 0.02083333  
## 6 0.03050847 0.027118644 0.00000000 0.00000000 0.94237288  
## 7 0.10530191 0.092047128 0.04197349 0.70250368 0.05817378  
## 8 0.10530191 0.092047128 0.04197349 0.70250368 0.05817378  
## 9 0.99631336 0.003686636 0.00000000 0.00000000 0.00000000  
## 10 0.77222777 0.178821179 0.00999001 0.01198801 0.02697303  
## 11 0.05553068 0.158978952 0.53246753 0.13703538 0.11598746  
## 12 0.05553068 0.158978952 0.53246753 0.13703538 0.11598746  
## 13 0.72743056 0.210069444 0.02777778 0.01388889 0.02083333  
## 14 0.99631336 0.003686636 0.00000000 0.00000000 0.00000000  
## 15 0.05553068 0.158978952 0.53246753 0.13703538 0.11598746  
## 16 0.15384615 0.197115385 0.00000000 0.05288462 0.59615385  
## 17 0.72743056 0.210069444 0.02777778 0.01388889 0.02083333  
## 18 0.10530191 0.092047128 0.04197349 0.70250368 0.05817378  
## 19 0.10071942 0.194244604 0.54316547 0.10431655 0.05755396  
## 20 0.05187320 0.694524496 0.09077810 0.03746398 0.12536023

# Random forest variation with regularization of data (Nikki)

set.seed(1234)  
x <- subset(part\_training, select=-classe)  
y <- part\_training$classe  
bestmtry <- tuneRF(x, y, stepFactor = 1.5, improve=1e-5, ntree=500)

## mtry = 7 OOB error = 0.5%   
## Searching left ...  
## mtry = 5 OOB error = 0.52%   
## -0.04347826 1e-05   
## Searching right ...  
## mtry = 10 OOB error = 0.48%   
## 0.04347826 1e-05   
## mtry = 15 OOB error = 0.5%   
## -0.03030303 1e-05



print(bestmtry)

## mtry OOBError  
## 5.OOB 5 0.005242082  
## 7.OOB 7 0.005023662  
## 10.OOB 10 0.004805242  
## 15.OOB 15 0.004950855

nikki\_rf <- randomForest(classe~., data=part\_training, ntree=300, mtry=10, importance=TRUE)  
nikki\_pred <- predict(nikki\_rf, part\_test)  
confusionMatrix(nikki\_pred, part\_test$classe, dnn=c("prediction","reference"))

## Confusion Matrix and Statistics  
##   
## reference  
## prediction A B C D E  
## A 1685 3 0 0 0  
## B 2 1116 7 0 0  
## C 0 1 1011 12 2  
## D 0 0 2 954 0  
## E 0 0 0 1 1091  
##   
## Overall Statistics  
##   
## Accuracy : 0.9949   
## 95% CI : (0.9927, 0.9966)  
## No Information Rate : 0.2866   
## P-Value [Acc > NIR] : < 2.2e-16   
##   
## Kappa : 0.9936   
## Mcnemar's Test P-Value : NA   
##   
## Statistics by Class:  
##   
## Class: A Class: B Class: C Class: D Class: E  
## Sensitivity 0.9988 0.9964 0.9912 0.9866 0.9982  
## Specificity 0.9993 0.9981 0.9969 0.9996 0.9998  
## Pos Pred Value 0.9982 0.9920 0.9854 0.9979 0.9991  
## Neg Pred Value 0.9995 0.9992 0.9981 0.9974 0.9996  
## Prevalence 0.2866 0.1902 0.1733 0.1643 0.1857  
## Detection Rate 0.2862 0.1896 0.1717 0.1621 0.1853  
## Detection Prevalence 0.2867 0.1911 0.1743 0.1624 0.1855  
## Balanced Accuracy 0.9991 0.9973 0.9940 0.9931 0.9990

# Run final prediction  
nikki\_finalpred <- predict(nikki\_rf, test\_final)  
nikki\_finalpred

## 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

# Support Vector Machine by Stefan (we attempted to tune it but the accuracy only minorly improved and it was quite time consuming)

stefan\_svm <- svm(classe ~. ,data = part\_training)  
stefan\_pred<- predict(stefan\_svm, part\_test)  
confusionMatrix(stefan\_pred,part\_test$classe, dnn=c("prediction","reference"))

## Confusion Matrix and Statistics  
##   
## reference  
## prediction A B C D E  
## A 1678 65 0 3 0  
## B 0 1033 23 0 4  
## C 7 22 986 82 30  
## D 0 0 8 878 17  
## E 2 0 3 4 1042  
##   
## Overall Statistics  
##   
## Accuracy : 0.9541   
## 95% CI : (0.9485, 0.9593)  
## No Information Rate : 0.2866   
## P-Value [Acc > NIR] : < 2.2e-16   
##   
## Kappa : 0.9419   
## Mcnemar's Test P-Value : NA   
##   
## Statistics by Class:  
##   
## Class: A Class: B Class: C Class: D Class: E  
## Sensitivity 0.9947 0.9223 0.9667 0.9080 0.9533  
## Specificity 0.9838 0.9943 0.9710 0.9949 0.9981  
## Pos Pred Value 0.9611 0.9745 0.8749 0.9723 0.9914  
## Neg Pred Value 0.9978 0.9820 0.9929 0.9821 0.9895  
## Prevalence 0.2866 0.1902 0.1733 0.1643 0.1857  
## Detection Rate 0.2850 0.1755 0.1675 0.1491 0.1770  
## Detection Prevalence 0.2966 0.1801 0.1914 0.1534 0.1785  
## Balanced Accuracy 0.9892 0.9583 0.9688 0.9514 0.9757

# This is Stefan’s neural network, it wouldn’t run so we left it commented out

scaledata<-scale(train\_final[,1:52]) #scale normalization

normalize <- function(x) { return ((x - min(x)) / (max(x) - min(x))) }#max-min normalization

maxmindf<-as.data.frame(lapply(scaledata, normalize))#normilization takes too much computing power

test\_final\_2<-cbind(maxmindf,class.ind(train\_final$classe))

# partitions 70% of data into training set

trainingRowIndex<-sample(1:nrow(train\_final2), size = .7\*nrow(train\_final2)) trainset<-train\_final2[trainingRowIndex, ]

# leaves 30% for testing and validating

testset<-train\_final2[-trainingRowIndex, ]

library(neuralnet) #Neural Network

nn <- neuralnet(A + B + C + D + E ~ roll\_belt+pitch\_belt+yaw\_belt+total\_accel\_belt+gyros\_belt\_x+gyros\_belt\_y +gyros\_belt\_z+accel\_belt\_x+accel\_belt\_y+accel\_belt\_z+magnet\_belt\_x+magnet\_belt\_y +magnet\_belt\_z+roll\_arm+pitch\_arm+yaw\_arm+total\_accel\_arm+gyros\_arm\_x+gyros\_arm\_y +gyros\_arm\_z+accel\_arm\_x+accel\_arm\_y+accel\_arm\_z+magnet\_arm\_x+magnet\_arm\_y+magnet\_arm\_z +roll\_dumbbell+pitch\_dumbbell+yaw\_dumbbell+total\_accel\_dumbbell+gyros\_dumbbell\_x +gyros\_dumbbell\_y+gyros\_dumbbell\_z+accel\_dumbbell\_x+accel\_dumbbell\_y+accel\_dumbbell\_z +magnet\_dumbbell\_x+magnet\_dumbbell\_y+magnet\_dumbbell\_z+roll\_forearm+pitch\_forearm +yaw\_forearm+total\_accel\_forearm+gyros\_forearm\_x+gyros\_forearm\_y+gyros\_forearm\_z +accel\_forearm\_x+accel\_forearm\_y+accel\_forearm\_z+magnet\_forearm\_x+magnet\_forearm\_y +magnet\_forearm\_z, data=test\_final\_2, hidden=c(10,5),linear.output=FALSE, threshold=0.01) nn$result.matrix plot(nn)

# Test the resulting output

temp\_test <- subset(testset, select = c(“roll\_belt”,“pitch\_belt”, “yaw\_belt”,“total\_accel\_belt”,“gyros\_belt\_x”,“gyros\_belt\_y”,“gyros\_belt\_z”,“accel\_belt\_x”,“accel\_belt\_y”,“accel\_belt\_z”,“magnet\_belt\_x”,“magnet\_belt\_y”,“magnet\_belt\_z”,“roll\_arm”,“pitch\_arm”,“yaw\_arm”,“total\_accel\_arm”,“gyros\_arm\_x”,“gyros\_arm\_y”,“gyros\_arm\_z”,“accel\_arm\_x”,“accel\_arm\_y”,“accel\_arm\_z”,“magnet\_arm\_x”,“magnet\_arm\_y”,“magnet\_arm\_z”,“roll\_dumbbell”,“pitch\_dumbbell”,“yaw\_dumbbell”,“total\_accel\_dumbbell”,“gyros\_dumbbell\_x”,“gyros\_dumbbell\_y”,“gyros\_dumbbell\_z”,“accel\_dumbbell\_x”,“accel\_dumbbell\_y”,“accel\_dumbbell\_z”,“magnet\_dumbbell\_x”,“magnet\_dumbbell\_y”,“magnet\_dumbbell\_z”,“roll\_forearm”,“pitch\_forearm”,“yaw\_forearm”,“total\_accel\_forearm”,“gyros\_forearm\_x”,“gyros\_forearm\_y”,“gyros\_forearm\_z”,“accel\_forearm\_x”,“accel\_forearm\_y”,“accel\_forearm\_z”,“magnet\_forearm\_x”,“magnet\_forearm\_y”,“magnet\_forearm\_z”))

head(temp\_test)

nn.results <- compute(nn, temp\_test)

# Accuracy

results <- data.frame(actual = testset(“A”,“B”,“C”,“D”,“E”), prediction = nn.results$net.result)

results roundedresults<-sapply(results,round,digits=0) roundedresultsdf=data.frame(roundedresults) attach(roundedresultsdf) table(actual,prediction)