# Weight Lifting Exercise Analysis

# Import Data Set

data\_set <- read.csv("E:/Acadgild/Class9/Assignments/Example\_WearableComputing\_weight\_lifting\_exercises\_biceps\_curl\_variations.csv")

View(data\_set)

# remove irrelevant collumns viz. name, cvtd\_timestamp, new\_window

data <- data\_set[,-c(1,4,5)]

View(data)

str(data)

# 2. Perform the below given activities:

# a. Create classification model using logistic regression model

# the target variable variable is multiple level

sum(is.na(data)) # there are no missing values

# spliting the data set for train and test

library(caTools)

set.seed(123)

split = sample.split(data$classe, SplitRatio = 0.7)

train = subset(data, split == TRUE) # train data

test = subset(data, split == FALSE) # test data

library(nnet) ; library(MASS)

model <- multinom(classe ~., data = train)

summary(model)

# stepAIC(model, direction = "backward")

final <- multinom(classe ~ raw\_timestamp\_part\_1 + num\_window + 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\_y +

magnet\_arm\_z + roll\_dumbbell + pitch\_dumbbell + yaw\_dumbbell +

gyros\_dumbbell\_x + 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 = train)

final

summary(final)

# Predictions

predicted <- predict(final, newdata= test)

# ---------------------------------------------------------------------

# b. Goodness of Fit

library(car)

chisq.test(table(test$classe), prop.table(table(predicted)))

# --------------------------------------------------------------------------

# c. Report the accuracy measures

# Accuracy

conf <- table(test$classe, predicted)

OAA <- (conf[1,1]+conf[2,2]+conf[3,3]+conf[4,4]+conf[5,5]) / sum(conf)

OAA

# --------------------------------------------------------------------------

# d. Report the variable importance

coef(final)

library(caret)

varImp(final)

# --------------------------------------------------------------------------

# e. Report the unimportant variables

install\_github("riv","tomasgreif")

install\_github("woe","tomasgreif")

library(devtools); library(woe); library(riv)

iv\_df <- iv.mult(train, y = "classe", summary= FALSE, verbose = TRUE)

variables <- c(colnames(train[,-56]))

imp\_variables <- names(as.data.frame(coef(final)))

unimportant\_variables <- setdiff(variables, imp\_variables)

unimportant\_variables

# --------------------------------------------------------------------------

# f. Interpret the results

# 1. Model execution output shows some iteration history and includes the final negative log-likelihood 49.43.

# This value is multiplied by two as shown in the model summary as the Residual Deviance 98.86

# 2. The summary output has a block of coefficients and another block of standard errors.

# Each blocks has row of values corresponding to each category of DV

# and each column represents the predictor

# and the values show the coefficients and standard errors

library(pROC)

m <- multiclass.roc(as.numeric(classe) ~ as.numeric(predicted) , data = test)

# AUC = 0.9977

# --------------------------------------------------------------------------

# g. Visualize the results

plot <- plot(conf, col = topo.colors(6))

library(ggplot2)

ggplot(data = as.data.frame(conf), mapping = aes(x = predicted,y = Var1)) +

geom\_tile(aes(fill = Freq)) +

geom\_text(aes(label = sprintf("%1.0f", Freq)), vjust = 1) +

scale\_fill\_gradient(low = "blue", high = "red", trans = "log")