```
library(randomForest)

#####################################################################################################
#####################################################################################################
#####################################################################################################

build_model=function(X,y){
clf = randomForest(x=X,y=y,mtry=6,importance=TRUE)
return(clf)
}


test_model <- function(model,X){
return(predict(model,X))
}

#####################################################################################################
#####################################################################################################
#####################################################################################################

## set the seed to make your partition reproducible
## do not change the seed, this would make a lot of things a lot more difficult than it needs to  be
random_seed = 42
set.seed(random_seed)

# do not change this code
# the snippet generates the datasets for modelling

raw_dat = read.csv('DSDataLastThreeMonths.csv')
hm_temp=raw_dat$HM_TEMP
hm_temp[is.na(hm_temp)]=mean(hm_temp,na.rm=T)
sum(is.na(hm_temp))
[1] 0
head((raw_dat1=cbind(raw_dat,hm_temp)))


## 2/3rd of the sample size
smp_size <- floor(0.67 * nrow(raw_dat1))

train_ind <- sample(seq_len(nrow(raw_dat1)), size = smp_size)

train <- raw_dat1[train_ind, ]
test <- raw_dat1[-train_ind, ]
X_names = c('HM_WT', 'AIM_S', 'HM_S', 'HM_C', 'HM_SI', 'HM_TI','HM_MN', 'CAC2', 'MG', 'hm_temp', 'CA
y_name = 'DS_S'
form = paste(y_name,'~',paste(X_names,collapse ='+'))

X_train = train[,X_names]
X_test = test[,X_names]

y_train = train[,y_name]
y_test = test[,y_name]


model = build_model(X_train,y_train)


pred_test = test_model(model,X_test)
pred_train = test_model(model,X_train)


#tolerance range
check = 0.003

# finding the error on the predictions
err_test = pred_test - y_test
err_train = pred_train - y_train

# finding the strike rates on the datasets
strike_rate_test = 100*(sum((err_test<=check) & (err_test >= -check)))/length(err_test)
```

```
strike_rate_train = 100*(sum((err_train<=check) & (err_train >= -check)))/length(err_train)

# printint the results
print(paste("Test strike rate :",strike_rate_test,"Train strike rate :",strike_rate_train))
[1] "Test strike rate : 87.2908186341022 Train strike rate : 99.2197949175212"   ###Result from R co


#Finding the mean square error (MSE) value
MSE=mean((pred_test-y_test)^2)
MSE
[1] 4.480502e-06   ###### Result from R console #####
# Finding the important variables used

importance(model)
             %IncMSE IncNodePurity         #####Result from R console####
HM_WT          2.623929  0.0018574964
AIM_S         23.532338  0.0010039644
HM_S          50.389180  0.0041117669
HM_C           5.042195  0.0017198046
HM_SI         28.454821  0.0036075003
HM_TI         75.469090  0.0091474565
HM_MN         14.290039  0.0019282223
CAC2          30.682715  0.0024710430
MG            50.633741  0.0028189891
hm_temp       15.563958  0.0027144721
CAC2_INJ_TIME 30.168980  0.0012265768
MG_INJ_TIME   23.051144  0.0009825158

#Plotting the important variables

varImpPlot(model)
Plot attached in the mail

save.image('team_The Bug Slayer-Final(1).RData')
```