Weight Lifting Movement Analysis

Pratham Sheel

5/16/2022

Libraries

First we will load the required libraries for our analysis:-

```
library(caret)
```

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

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

```
library(rpart)
library(class)
```

Data Loading

Now,we will load our training and testing datasets.

```
training<-read.csv("H:/Course Data/coursera/Course 8/pml-training.csv")

testing<-read.csv("H:/Course Data/coursera/Course 8/pml-testing.csv")
```

Data Pre-Processing

To eliminate the variables which are meaningless in our dataset. for the elimination of variables we will use the Near Zero Variance function. It will give us a dataframe of all variables with their frequency ratio, percentage of unique values. Lets see the first 10 observations of Near Zero Variance DF:-

```
NZV <- nearZeroVar(training, saveMetrics = TRUE)
head(NZV,10)
```

```
##                        freqRatio percentUnique zeroVar   nzv
## X                       1.000000  100.00000000   FALSE FALSE
## user_name               1.100679    0.03057792   FALSE FALSE
## raw_timestamp_part_1    1.000000    4.26562022   FALSE FALSE
## raw_timestamp_part_2    1.000000   85.53154622   FALSE FALSE
## cvtd_timestamp          1.000668    0.10192641   FALSE FALSE
## new_window             47.330049    0.01019264   FALSE  TRUE
```

```
## num_window             1.000000    4.37264295   FALSE FALSE
## roll_belt              1.101904    6.77810621   FALSE FALSE
## pitch_belt             1.036082    9.37722964   FALSE FALSE
## yaw_belt               1.058480    9.97349913   FALSE FALSE
```

Now, we will remove the columns which has near zero variance or meaningless and save it in a new data frame called train01.

```
train01<-training[,!NZV$nzv]
test01<-testing[,!NZV$nzv]
```

Removing some of the columns which are not much relevant for the data modeling. These columns are the X variable, user_name, timestamp variables and new_window variable.

```
train02<-train01[,-c(1:5)]
test02<-test01[,-c(1:5)]
```

Remove all the remaining columns that contain "NA's"

```
cond <- (colSums(is.na(train02)) == 0)
train03 <- train02[, cond]
test03 <- test02[, cond]
test03<-test03[,-54]
```

The dimensions of our processed dataframe are **19622, 54**.

Removing all the objects which are not required.

```
rm(train01)
rm(train02)
rm(test01)
rm(test02)
rm(training)
rm(testing)
```

Data Partion

Now we will create **Validation set** to check the accuracy of our model.

```
set.seed(12345)
inTrain <- createDataPartition(train03$classe, p = 0.70, list = FALSE)
validation <- train03[-inTrain, ]
train03 <- train03[inTrain, ]
```

Data Modeling

Now we will use Machine Learning model for prediction. We will use two models for predictions. Then we will select one model which will give better accuracy.

First we will use **KNN (K Nearest Neighbor)** Model with **5 as K value** as default value.

```
model1<-train(classe~.,data=train03,method="knn")
model1
```

```
## k-Nearest Neighbors
##
## 13737 samples
##    53 predictor
##     5 classes: 'A', 'B', 'C', 'D', 'E'
##
## No pre-processing
## Resampling: Bootstrapped (25 reps)
## Summary of sample sizes: 13737, 13737, 13737, 13737, 13737, 13737, ...
## Resampling results across tuning parameters:
##
##   k  Accuracy   Kappa
##   5  0.8814217  0.8499958
##   7  0.8668359  0.8315475
##   9  0.8530358  0.8140678
##
## Accuracy was used to select the optimal model using the largest value.
## The final value used for the model was k = 5.
```

```
pred1<-predict(model1,validation)
acc1=mean(pred1==validation$classe)
```

So, the accuracy of our KNN model on validation set is **91.9456245%**

The second Machine Learning Model is **Random Forest algorithm** because it automatically selects important variables and is robust to correlated covariates & outliers in general.
We will use 5-fold cross validation when applying the algorithm.

```
model2 <- train(classe ~ ., data = train03, method = "rf", trControl = trainControl(method = "cv", 5), n
model2
```

```
## Random Forest
##
## 13737 samples
##    53 predictor
##     5 classes: 'A', 'B', 'C', 'D', 'E'
##
## No pre-processing
## Resampling: Cross-Validated (5 fold)
## Summary of sample sizes: 10989, 10989, 10989, 10991, 10990
## Resampling results across tuning parameters:
##
##   mtry  Accuracy   Kappa
##    2    0.9936669  0.9919882
##   27    0.9965059  0.9955801
##   53    0.9940305  0.9924486
##
## Accuracy was used to select the optimal model using the largest value.
## The final value used for the model was mtry = 27.
```

```
pred2<-predict(model2,validation)
acc2=mean(pred2==validation$classe)
```

So, the accuracy of our Random Forest model on validation set is **99.8810535%**

So, as Random forest is giving better accuracy than KNN. We will use Random Forest Model for predictions.

```
predict(model2,test03)
```

```
##  [1] 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
```

File Generation

Function to generate files with predictions to submit for assignment:-

```
pml_write_files = function(x){
n = length(x)
  for(i in 1:n){
    filename = paste0("C:/Users/user/R_codes/Course8/Assignment_Solutions/problem_id_",i,".txt")
    write.table(x[i], file = filename, quote = FALSE, row.names = FALSE, col.names = FALSE)
  }
}
```

For Generating files:-

```
pml_write_files(predict(model2,test03))
```