Report on Predicting Exercise Performance with Wearable Device Data

Raghda

2024-06-24

Background

The advent of wearable devices such as the Jawbone Up, Nike FuelBand, and Fitbit has enabled the collection of extensive personal activity data at a relatively low cost. These devices contribute to the quantified self movement, where enthusiasts regularly measure various aspects of their activities to improve health, identify behavior patterns, or simply due to their interest in technology. While people often quantify the amount of a particular activity, they seldom quantify how well they perform it. This project aims to bridge this gap by using data from accelerometers on the belt, forearm, arm, and dumbbell of six participants who performed barbell lifts correctly and incorrectly in five different ways. The goal is to predict the manner of exercise, classified by the "classe" variable in the training dataset.

Load necessary packages

```
library(caret)
library(randomForest)
library(rpart)
library(RColorBrewer)
library(rattle)
```

Data Preprocessing

Loading the Data:

The training and testing datasets were loaded into R, with missing values represented by "NA", "#DIV/0!", and empty strings handled appropriately.

```
training <- read.csv(url(trUrl), na.strings=c("NA","#DIV/0!",""))
testing <- read.csv(url(teUrl), na.strings=c("NA","#DIV/0!",""))
```

Cleaning the Data:

Columns with missing values and irrelevant columns (such as ID columns) were removed from both datasets. This resulted in a clean training and testing dataset.

```
training <- training[,colSums(is.na(training)) == 0]
testing <- testing[,colSums(is.na(testing)) == 0]
trainclean <- training[,-c(1:7)]
testclean <- testing[,-c(1:7)]
```

Including Plots

You can also embed plots, for example:

Splitting the Data:

The clean training data was split into 70% training and 30% validation sets to enable model evaluation.

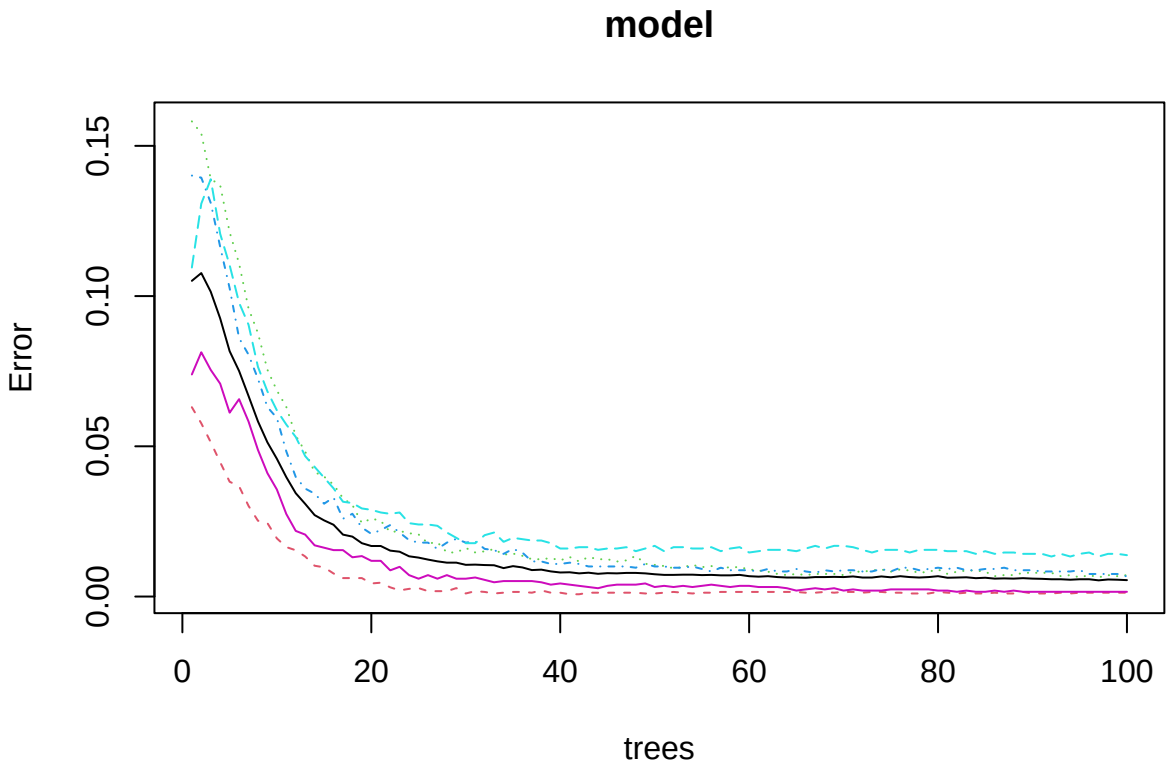
```
set.seed(1234)
inTrain <- createDataPartition(y = trainclean$classe, p = 0.70, list = FALSE)
mytrain <- trainclean[inTrain, ]
valset <- trainclean[-inTrain, ]
```

Model Building

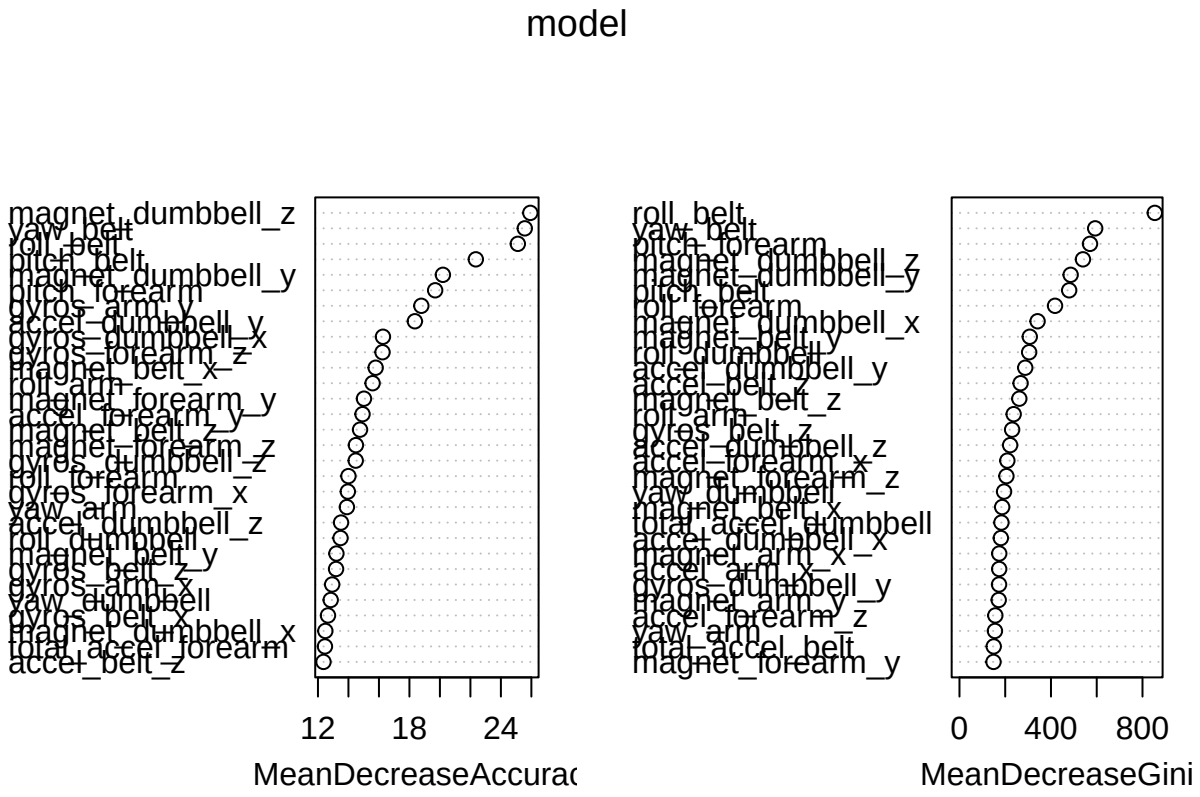
Random Forest Model:

A Random Forest model was trained on the training set using 100 trees. The importance of each variable was also analyzed.

```
model <- randomForest(classe ~ ., data = mytrain, ntree = 100, importance = TRUE)
plot(model)
```




```
varImpPlot(model)
```



Model Evaluation

Predictions and Confusion Matrix:

Predictions were made on the validation set, and a confusion matrix was calculated to evaluate the model's performance.

```
predictions <- predict(model, valset)
predictions <- factor(predictions, levels = levels(mytrain$classe))
valset$classe <- factor(valset$classe, levels = levels(mytrain$classe))
```

```
confMatrix <- confusionMatrix(predictions, valset$classe)
print(confMatrix)
```

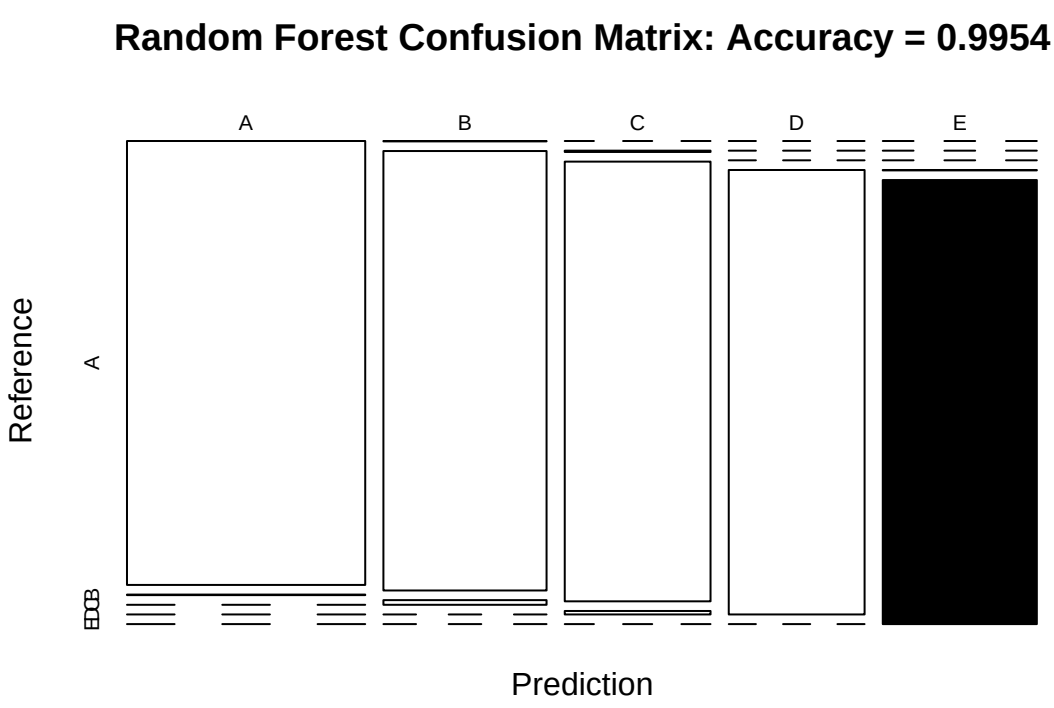
```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction    A    B    C    D    E
##          A 1673    2    0    0    0
##          B    1 1134   12    0    0
##          C    0    3 1014    8    0
##          D    0    0    0  955    0
##          E    0    0    0    1 1082
##
```

```
## Overall Statistics
##
##                Accuracy : 0.9954
##                  95% CI : (0.9933, 0.997)
##     No Information Rate : 0.2845
##     P-Value [Acc > NIR] : < 2.2e-16
##
##                   Kappa : 0.9942
##
##  Mcnemar's Test P-Value : NA
##
## Statistics by Class:
##
##                      Class: A Class: B Class: C Class: D Class: E
## Sensitivity            0.9994   0.9956   0.9883   0.9907   1.0000
## Specificity            0.9995   0.9973   0.9977   1.0000   0.9998
## Pos Pred Value         0.9988   0.9887   0.9893   1.0000   0.9991
## Neg Pred Value         0.9998   0.9989   0.9975   0.9982   1.0000
## Prevalence             0.2845   0.1935   0.1743   0.1638   0.1839
## Detection Rate         0.2843   0.1927   0.1723   0.1623   0.1839
## Detection Prevalence   0.2846   0.1949   0.1742   0.1623   0.1840
## Balanced Accuracy      0.9995   0.9964   0.9930   0.9953   0.9999
```

Accuracy and Visualization:

The model's accuracy was plotted, showing the confusion matrix with class-wise accuracy.

```
plot(confMatrix$table, col = confMatrix$byClass, main = paste("Random Forest Confusion Matrix: Accuracy
```



Final Prediction

The trained Random Forest model was used to make predictions on the cleaned test dataset.

```
finalpredict <- predict(model, testclean)
print(finalpredict)
```

```
##  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
```

Results

The Random Forest model demonstrated high accuracy in predicting the manner in which the exercises were performed, as indicated by the confusion matrix on the validation set. The final predictions on the test set provided the manner of exercise for 20 different test cases.

Conclusion

This project successfully applied machine learning techniques to predict the quality of barbell lifts using data from wearable devices. The Random Forest model proved to be an effective method, with its robustness and ability to handle numerous predictor variables. Future work could explore the integration of additional data sources and the application of other machine learning algorithms to further improve prediction accuracy.