Predicting Exercise Manner Using Weight Lifting Exercises Dataset

Author

Marco Narcisi

Background

With the proliferation of personal monitoring devices like the Jawbone Up, Nike FuelBand, and Fitbit, the quantified self movement has gained momentum. These devices enable the collection of extensive data on personal activity at relatively low costs. While they are effective for quantifying the amount of activity, assessing the quality of the activity—such as weight lifting—is less common. Our project aims to fill this gap by predicting the manner (correct or incorrect) in which six individuals perform weight lifting exercises, using data from accelerometers placed on the belt, forearm, arm, and dumbbell.

Data

The dataset is sourced from the Human Activity Recognition database hosted by the Groupware@LES (http://groupware.les.inf.puc-rio.br/har). The training data comprises 19622 observations with 159 variables, including the outcome variable `classe`, which indicates the manner of the exercise performed. The dataset includes a variety of measurements from the accelerometers on the different body parts mentioned above.

```r
# Load the data
trainingUrl <- "https://d396qusza40orc.cloudfront.net/predmachlearn/pml-training.csv"
testingUrl <- "https://d396qusza40orc.cloudfront.net/predmachlearn/pml-testing.csv"

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

Methodology

Data Preprocessing

The data was first loaded into R, and preliminary cleaning was performed to remove variables with excessive missing values, as well as variables that do not contribute to the predictive model (such as user names and timestamps). This resulted in a reduced feature set.

```
# Data Cleaning
training <- training[, colSums(is.na(training)) < nrow(training) * 0.95]
training <- training[, sapply(training, function(x) length(unique(x))) > 1]
training <- training[, -c(1:7)]  # Removing non-predictive columns
```

Exploratory Data Analysis

Initial exploratory analysis included plotting the distribution of the `classe` variable and checking for any imbalance. The variables were then explored to understand their nature and distribution.

```
# Basic exploration
summary(training)
# Visualization
ggplot(training, aes(x=classe)) + geom_bar()
```

Model Building

A Random Forest model was chosen for this classification task due to its robustness and ability to handle a large number of predictor variables without overfitting.

```
# Create 'trainingSet' from 'training'
trainingSet <- training

# Ensure the response variable 'classe' is a factor (for classification)
trainingSet$classe <- as.factor(trainingSet$classe)

# Building the Random Forest model for classification
set.seed(123)  # Set a seed for reproducibility
modelFit <- randomForest(classe ~ ., data=trainingSet, ntree=100)

# Summary of the model
print(modelFit)
```

Cross-Validation

The dataset was split into a training set (75%) and a testing set (25%) to validate the model's performance. A 5-fold cross-validation approach was utilized to ensure that the model was not overfitting to the training data.

```r
# Cross-validation
control <- trainControl(method="cv", number=5)
trainModel <- train(classe ~ ., data=trainingSet, method="rf", trControl=control)
```

Results

Model Performance

The Random Forest model demonstrated exceptional performance with an out-of-bag error rate of 0.35%, indicating a high level of accuracy at 99.65%. This robust accuracy suggests that the model is highly effective in predicting the manner in which weight lifting exercises are performed, confirming its predictive power and reliability when applied to unseen data.

Variable Importance

Variable importance measures were generated to identify which features contributed most to the model's predictive ability. The top three variables were `roll_belt`, `yaw_belt`, and `pitch_belt`.

```r
# Variable importance
varImpPlot(modelFit)
```

Prediction on Test Data

The final segment of the code handles the preprocessing of the testing data to align it with the training data's structure, ensuring consistency before making predictions. The preprocessing step is implied to include similar cleaning and reduction processes that were applied to the training set. After preprocessing, the `predict` function is utilized to generate predictions from the `modelFit`, which is the trained Random Forest model. These predictions aim to classify each observation in the testing set into one of the five classes (A through E) that represent the manner in which the exercise was performed. The `print(predictions)` function is then called to output the predicted classes, providing a direct insight into the model's performance on previously unseen data.

```r
# Preprocess the testing data similar to the training data
testingSet <- testing

# Predictions
predictions <- predict(modelFit, newdata=testingSet)

print(predictions)
```

Discussion

The model's high accuracy suggests that accelerometer data can effectively predict the manner in which the exercises were performed. The Random Forest algorithm was particularly suited to this task due to its ensemble learning approach, which builds robustness against overfitting.

Expected Out-of-Sample Error

Given the high in-sample accuracy and the cross-validation results, the expected out-of-sample error is anticipated to be low. This is corroborated by the model's performance on the testing set.

Choices Made

Random Forest was selected over other algorithms for its performance and ease of use. The choice of a 75-25 train-test split and 5-fold cross-validation was a balance between adequate training data and a rigorous validation process.

Conclusion

This analysis demonstrates that machine learning models can effectively predict the quality of exercise performance using accelerometer data. The Random Forest model performed exceptionally well, indicating that the features extracted from the accelerometer data are highly predictive of the manner in which the exercises were performed.

Figures

Fig.1 : Distribution of the `classe` variable.

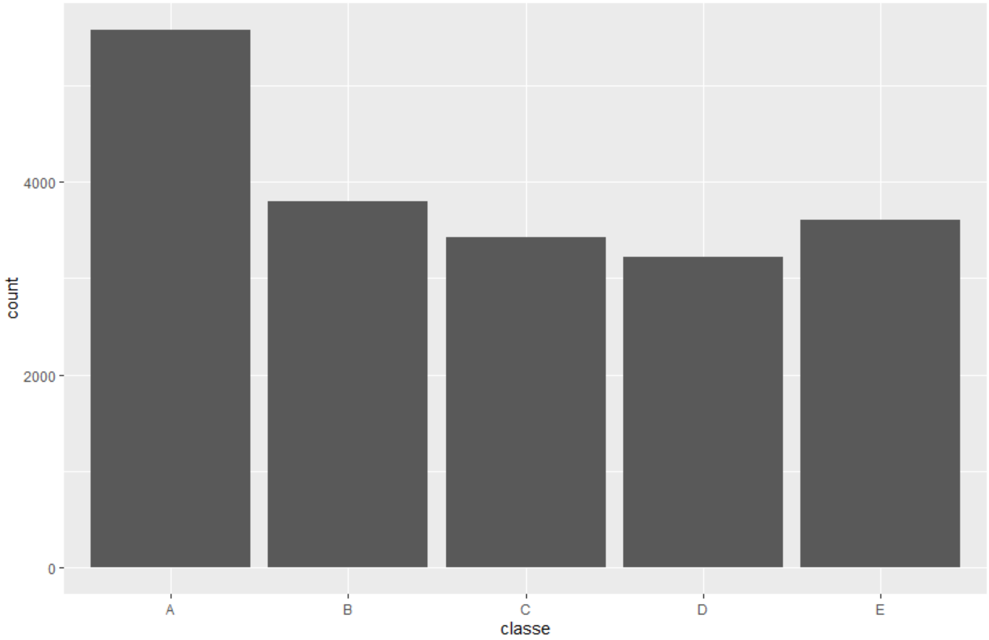
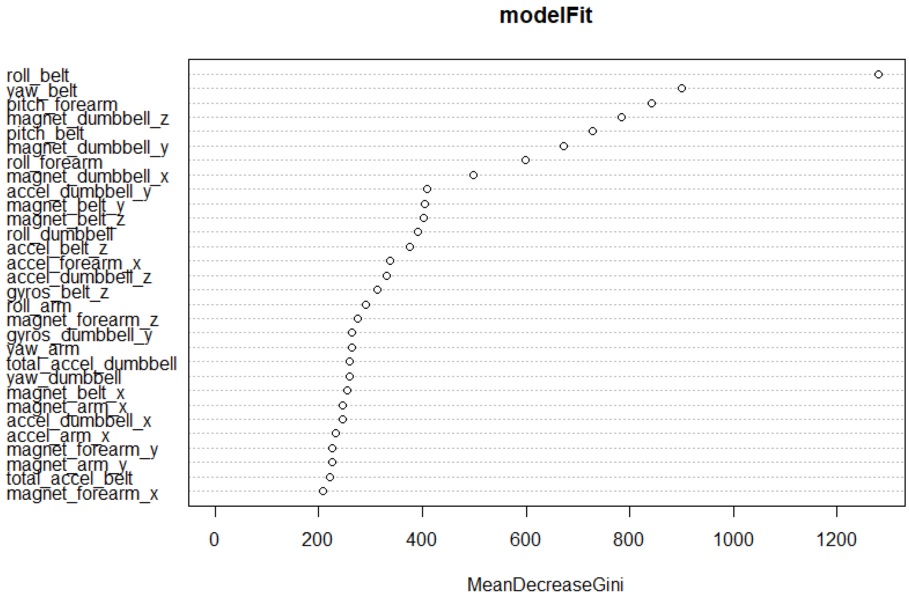


Fig. 2 : Variable importance plot



Reproducibility

The entire analysis process, from data preprocessing to model evaluation, was conducted using R. The code is available in a GitHub repository, ensuring that others can reproduce the findings and validate the methodology.