PREDICTING MACHINE LEARNING PROJECT

Sadza Raisya Salsabila

6/24/2025

Overview

This project aims to forecast the "classe" variable in the training dataset. A Random Forest model will be developed, incorporating cross-validation for robustness. The model's performance will be assessed based on out-of-sample error, and it will subsequently be applied to predict outcomes for 20 distinct test cases.

```
library(dplyr)
```

```
##
## Attaching package: 'dplyr'

## The following objects are masked from 'package:stats':
##
##     filter, lag

## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
```

```
library(ggplot2)
library(caret)
```

```
## Warning: package 'caret' was built under R version 4.0.3

## Loading required package: lattice
```

```
set.seed(42)
```

Loading the Data

```
train_csv <- read.csv("pml-training.csv")
test_csv <- read.csv("pml-testing.csv")
```

Cleaning the Data

Columns exhibiting near-zero variability, predominantly containing NA values or irrelevant metadata, will be removed to refine the dataset.

```
nzv<-nearZeroVar(train_csv)

clean_train <- train_csv %>%
    select(-nzv)%>% #drop near zero variance columns
    select(-c(1:5))%>% #irrelevant metadata
    select_if(colMeans(is.na(.)) < .9)
```

```
## Note: Using an external vector in selections is ambiguous.
## i Use `all_of(nzv)` instead of `nzv` to silence this message.
## i See <https://tidyselect.r-lib.org/reference/faq-external-vector.html>.
## This message is displayed once per session.
```

```
clean_test <- test_csv%>%
    select(colnames(select(clean_train, -"classe")),"problem_id")

table(clean_train$classe)
```

```
##
##    A    B    C    D    E
## 5580 3797 3422 3216 3607
```

Model development:

Data partition

The training set will be divided into a validation subset and a smaller training subset. The original testing set ("clean_test") will remain untouched and reserved for final evaluation.

```
partition <- createDataPartition(clean_train$classe, p=0.70, list=FALSE)
train_data <- clean_train[partition, ]
test_data <- clean_train[-partition, ]
```

Creating and Testing the Models

A Random Forest algorithm with 5-fold cross-validation will be implemented. This choice is justified by the algorithm's resilience to outliers and its ability to handle correlated predictor variables effectively.

```
controlRf <- trainControl(method="cv", 5)
modelRf <- train(classe ~ ., data=train_data, method="rf", trControl=controlRf, ntree=250)
modelRf
```

```
## 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: 10990, 10990, 10990, 10989, 10989
## Resampling results across tuning parameters:
##
##   mtry  Accuracy   Kappa
##    2    0.9928662  0.9909758
##   27    0.9972338  0.9965011
##   53    0.9943218  0.9928172
##
## Accuracy was used to select the optimal model using the largest value.
## The final value used for the model was mtry = 27.
```

```
predictRf <- predict(modelRf, test_data)
```

The confusion matrix will be used to visually evaluate the model's predictive accuracy and classify its performance across different outcome categories.

```
confusionMatrix(as.factor(test_data$classe), predictRf)
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction    A    B    C    D    E
##          A 1674    0    0    0    0
##          B    1 1136    2    0    0
##          C    0    3 1023    0    0
##          D    0    0    3  960    1
##          E    0    0    0    4 1078
##
## Overall Statistics
##
##                Accuracy : 0.9976
##                  95% CI : (0.996, 0.9987)
##     No Information Rate : 0.2846
##     P-Value [Acc > NIR] : < 2.2e-16
##
##                   Kappa : 0.997
##
##  Mcnemar's Test P-Value : NA
##
## Statistics by Class:
##
##                      Class: A Class: B Class: C Class: D Class: E
## Sensitivity            0.9994   0.9974   0.9951   0.9959   0.9991
## Specificity            1.0000   0.9994   0.9994   0.9992   0.9992
## Pos Pred Value         1.0000   0.9974   0.9971   0.9959   0.9963
## Neg Pred Value         0.9998   0.9994   0.9990   0.9992   0.9998
## Prevalence             0.2846   0.1935   0.1747   0.1638   0.1833
## Detection Rate         0.2845   0.1930   0.1738   0.1631   0.1832
## Detection Prevalence   0.2845   0.1935   0.1743   0.1638   0.1839
## Balanced Accuracy      0.9997   0.9984   0.9973   0.9975   0.9991
```

Results (Accuracy & Out of Sample Error)

The model's reliability will be determined by analyzing its accuracy metrics, ensuring the predictions are both precise and consistent.

```
accuracy <- as.numeric(confusionMatrix(as.factor(test_data$classe), predictRf)$overall[1])
accuracy
```

```
## [1] 0.9976211
```

So, the estimated accuracy of the model is 99.75% and the estimated out-of-sample error is 0.25%.

Predicting for Clean Test Data Set

After thorough development and validation, the model will be deployed on the original test dataset sourced from the provided repository to generate the required predictions

```
result_test <- predict(modelRf, clean_test)
result_test
```

```
##  [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
```