Predicción de la Calidad del Ejercicio

Victor

2026-02-15

Preparación de Datos y Limpieza

Dado que el dataset tiene muchas columnas con valores NA o vacios, la limpieza es vital para que el modelo no se "pasme".:

```
library(caret); library(randomForest)
```

```
## Cargando paquete requerido: ggplot2
```

```
## Cargando paquete requerido: lattice
```

```
## randomForest 4.7-1.2
```

```
## Type rfNews() to see new features/changes/bug fixes.
```

```
##
## Adjuntando el paquete: 'randomForest'
```

```
## The following object is masked from 'package:ggplot2':
##
##     margin
```

```
# Carga de datos
train_url <- "https://d396qusza40orc.cloudfront.net/predmachlearn/pml-training.csv"
training_raw <- read.csv(url(train_url), na.strings=c("NA","#DIV/0!",""))

# 1. Eliminar columnas con NAs (más del 95%)
training_clean <- training_raw[, colSums(is.na(training_raw)) < (nrow(training_raw) * 0.95)]

# 2. Eliminar variables de identificación (ID, timestamps, nombres)
training_clean <- training_clean[, -(1:7)]
```

Estrategia de Validación Cruzada

Dividiremos el dataset training en un 75% para entrenamiento y un 25% para validación interna. Esto nos permitirá calcular el Out-of-Sample Error antes de tocar los 20 casos de prueba finales:

```r
set.seed(12345)
inTrain <- createDataPartition(training_clean$classe, p=0.75, list=FALSE)
train_set <- training_clean[inTrain, ]
val_set <- training_clean[-inTrain, ]
```

Construcción del Modelo: Random Forest

You can also embed plots, for example:

```r
# Entrenamiento con 5-fold Cross Validation para optimizar velocidad
control_rf <- trainControl(method="cv", number=5, verboseIter=FALSE)
model_rf <- train(classe ~ ., data=train_set, method="rf", trControl=control_rf)
```

Evaluación del Error Fuera de Muestra

```r
pred_rf <- predict(model_rf, val_set)
conf_matrix <- confusionMatrix(pred_rf, factor(val_set$classe))
# La precisión esperada suele ser > 99%
```

Predicciones para el Quiz (20 casos)

You can also embed plots, for example:

```r
# 1. Cargar datos de prueba (URL limpia)
test_url <- "https://d396qusza40orc.cloudfront.net/predmachlearn/pml-testing.csv"
testing_final <- read.csv(test_url, na.strings=c("NA","#DIV/0!",""))

# 2. Asegurar que las columnas coincidan con el entrenamiento
# Buscamos solo las variables predictoras usadas en el modelo
columnas_modelo <- names(train_set)[names(train_set) != "classe"]
testing_final_subset <- testing_final[, columnas_modelo]

# 3. Predicción final
quiz_predictions <- predict(model_rf, testing_final_subset)
quiz_predictions
```

```
##  [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
```

Inferencia: Si la precisión es del 99.2%, el Expected Out-of-Sample Error es de aproximadamente 0.8%.

Note Random Forest: Es excelente para capturar interacciones complejas entre los acelerómetros del cinturón, brazo y mancuerna sin necesidad de escalar los datos.

Limpieza de NAs: Reducir de 160 variables a unas 53 variables útiles acelera el entrenamiento drásticamente.

Cross-Validation (k=5): Proporciona un equilibrio entre sesgo y varianza sin sobrecargar la RAM de tu equipo.