# 加载必要的包

library(readxl) # 用于读取Excel文件

library(caret) # 用于模型训练和评估

library(randomForest) # 随机森林模型

library(xgboost) # XGBoost模型

library(writexl) # 用于导出Excel文件

# 读取数据

trainData <- read\_excel("C:/Users/Liu/Desktop/14ta.xlsx")

testData <- read\_excel("C:/Users/Liu/Desktop/14te.xlsx")

# 设定种子以确保可重复性

set.seed(3) # 设置一个随机种子

# 检查数据结构

str(trainData)

str(testData)

# 检查训练数据中的缺失值

missing\_train <- sum(is.na(trainData))

cat("训练数据集中的缺失值总数：", missing\_train, "\n")

# 检查测试数据中的缺失值

missing\_test <- sum(is.na(testData))

cat("测试数据集中的缺失值总数：", missing\_test, "\n")

# 处理缺失值（如有必要）

# 确保测试数据中的特征列与训练数据一致

testData <- testData[ , colnames(trainData)]

# 设置训练控制

controlD <- trainControl(method = "cv", number = 10, search = "grid")

# 设置 XGBoost 参数，包括 'rate\_drop' 和 'skip\_drop'

xgbGrid <- expand.grid(nrounds = 38, # 迭代次数

max\_depth = 2, # 树的最大深度

eta = 0.3, # 学习率

gamma = 0, # 节点分裂所需的最小损失函数下降值

colsample\_bytree = 1, # 每棵树使用的特征占比

min\_child\_weight = 1, # 叶节点最小样本权重和

subsample = 1, # 每棵树训练所用的样本占比

rate\_drop = 0.3, # 丢弃的权重比例

skip\_drop = 0.5) # 丢弃的树的比例

# 训练 XGBoost 模型

modelXGB <- train(AGB ~ ., data = trainData, method = 'xgbDART',

trControl = controlD, tuneGrid = xgbGrid,

verbose = FALSE)

# 查看模型

print(modelXGB)

# 进行预测

predictions <- predict(modelXGB, newdata = testData)

# 将预测结果与实际标签（如果有）导出到 Excel 文件

# 如果测试数据集中有真实标签，请确保它们的列名为 'AGB'

if ("AGB" %in% colnames(testData)) {

results <- data.frame(Actual = testData$AGB, Predicted = predictions)

} else {

results <- data.frame(Predicted = predictions)

}

# 导出结果到 Excel 文件

write\_xlsx(results, "C:/Users/asus/Desktop/15XGB.xlsx")

cat("预测结果已导出到 'C:/Users/asus/Desktop/15XGB.xlsx'\n")