Methodology and Model Implementation

Linglu Li

kNN-Based Missing Value Imputation

As noted in the Dataset Overview, EDUC (201 values; ≈ 46%), SES (220 values; ≈ 50%), MMSE (201 values; ≈ 46%), and CDR (201 values; ≈ 46%) were missings.

First, the dementia outcome variable (Group2) was defined using CDR scores, where CDR = 0 indicates a nondemented individual; rows with missing CDR values were removed because the outcome could not be determined. These same rows also lacked Educ and MMSE, so removing them eliminated missingness in those variables as well. Among the remaining observations, only 19 SES values were missing; these were imputed using the kNN() function from the VIM package with $k = 5$, which predicts each missing entry based on the five most similar observations across the other variables. After imputation, the auxiliary _imp columns created by kNN() were removed, and the cleaned dataset was used for further analysis. The full R implementation of this imputation procedure is shown below.

Non-predictive identifiers (ID, Hand, Visits, Delay) were removed prior to model training, and variables highly correlated with the target outcome (CDR) were also excluded to reduce information leakage and prevent overfitting.

```
cross2 <- cross %>%
  mutate(
    Group2 = ifelse(CDR == 0, "Nondemented", "Demented"),
    Group2 = factor(Group2, levels = c("Nondemented", "Demented"))
  ) %>%
  filter(!is.na(Group2)) %>%
  # Remove ID + other columns don't want as predictors
  select(M.F, Age, Educ, SES, MMSE, eTIV, nWBV, ASF, Group2)

# Impute SES using KNN
set.seed(1)
cross2_imp <- kNN(
  cross2,
  variable = "SES",
  k = 5
)

# VIM adds *_imp indicator columns
cross2_imp <- cross2_imp %>%
  select(-ends_with("_imp"))
```

Data Splitting Procedure

An initial 80/20 train–test split was used to evaluate model performance, but this approach produced accuracy values roughly 5% lower than those obtained with a more robust method. To achieve a more reliable estimate, 10-fold cross-validation was was implemented: the training data were partitioned into ten folds, and models were repeatedly trained and validated across all folds using a trainControl specification with class probabilities enabled.

```
set.seed(1)
K <-10

# One 80/20 split
train2 = sample(nrow(cross2_imp), 0.8*nrow(cross2_imp))
train_dat = cross2_imp[train2, ]
y_train <- train_dat$Group2

# TRUE labels
test_dat = cross2_imp[-train2,]
y_test <- test_dat$Group2

# Set num of folds
train_control <- trainControl(
  method = "cv",
  number = 10,
  verboseIter = TRUE,
  classProbs = TRUE
)
```

Classifier Models

This study evaluates five classifiers—Decision Tree, Random Forest, SVM, XGBoost, and Logistic Regression—to compare their effectiveness in predicting dementia status. Each model was trained using the train() function from the *caret* package, and the full R implementation is provided below.

Decision Tree (DT)

```
set.seed(1)

tree_grid <- expand.grid(cp = seq(0.0001, 0.05, length.out = 30))
tree_model <- train(
  Group2 ~ .,
  data = train_dat,
  method = "rpart",
  trControl = train_control,
  tuneGrid  = tree_grid
)
```

Random Forest (RF)

```
set.seed(1)

rf_model <- train(
  Group2 ~ . ,
  data      = train_dat,
  method    = "rf",
  trControl = train_control,
  tuneGrid   = data.frame(mtry = c(1, 2, 3, 4, 5)),
  importance = TRUE)
```

Support Vector Machine (SVM)

```
set.seed(1)

svm_grid <- expand.grid(
  C     = c(0.1, 0.5, 1, 2, 5, 10),
  sigma = c(0.001, 0.005, 0.01, 0.02, 0.05))
svm_model <- train(
  Group2 ~ .,
  data       = train_dat,
  method     = "svmRadial",
  trControl  = train_control,
  preProcess = c("center", "scale"),
  tuneGrid   = svm_grid,
  metric     = "Accuracy")
```

XGBoost

```
set.seed(1)

grid_tune <- expand.grid(
  nrounds = c(500, 1000, 1500), # number of trees
  max_depth = c(2,4,6),
  eta = 0.3, #learning rate
  gamma = 0, # pruning draft
  colsample_bytree = 1, #subsample ratio of col for tree
  min_child_weight = 1, # the larger the more conservative the model; can be used as a stop
  subsample = 1 # used to prevent overfitting by sampling x% train
)
xgb_model <- train(
  Group2 ~ .,
  data      = train_dat,
  method    = "xgbTree",
  trControl = train_control,
  tuneGrid  = grid_tune,
  verbose   = TRUE)
```

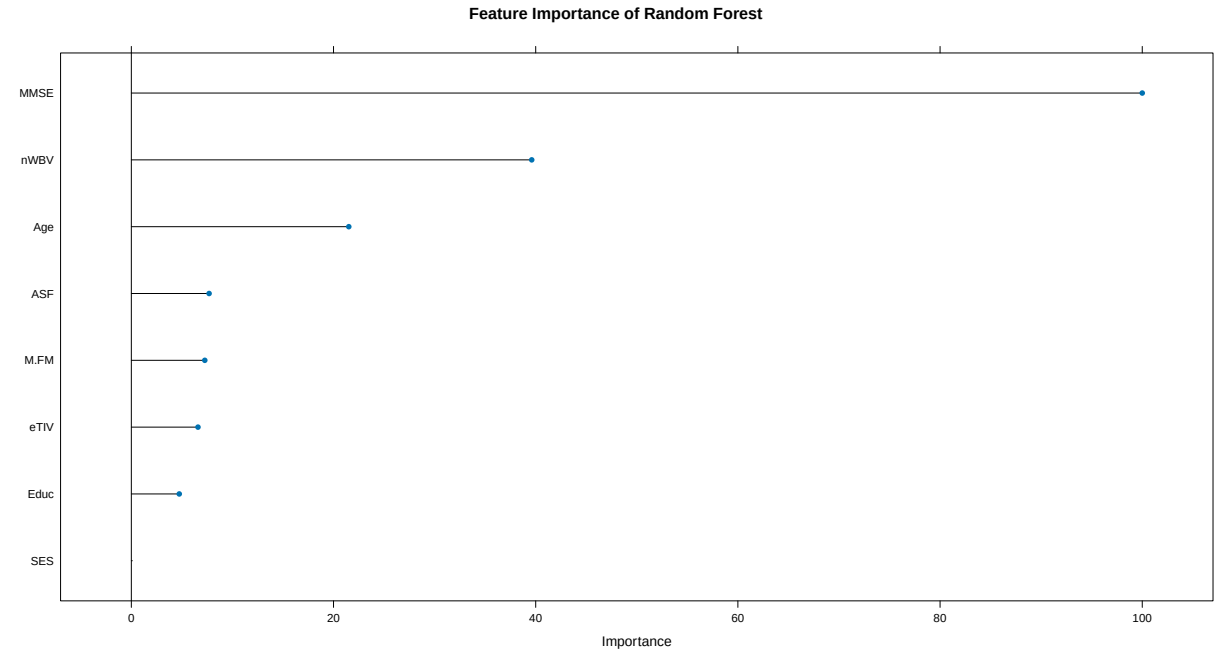
Logistic Regression

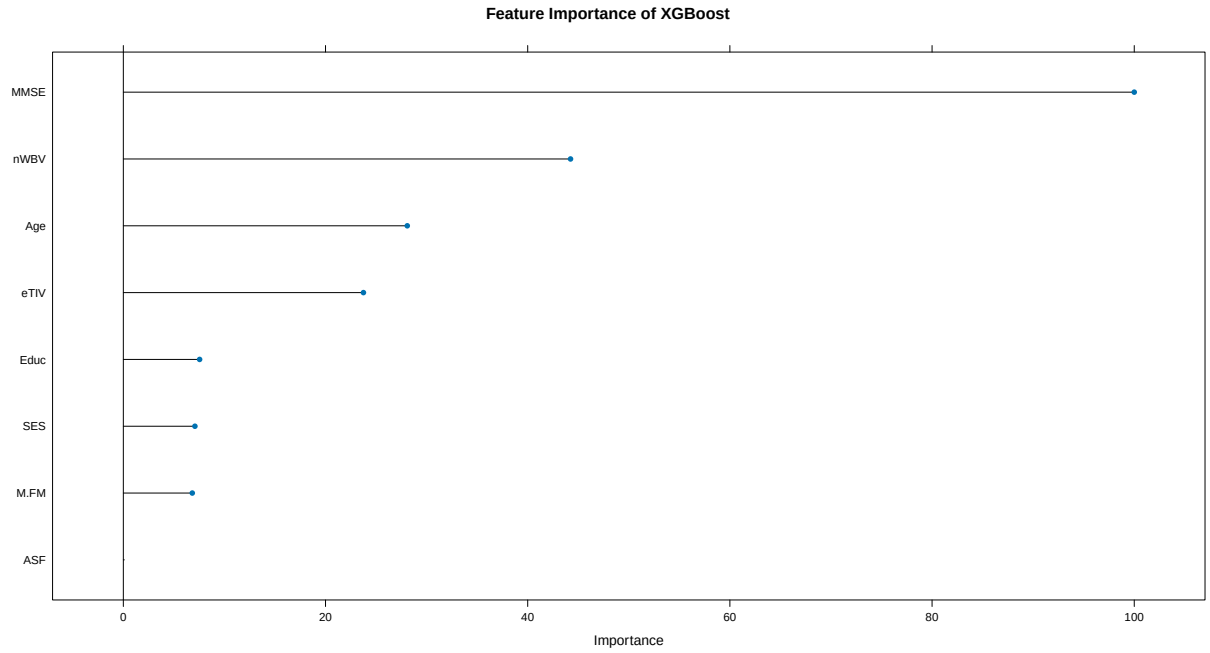
```
set.seed(1)

logit_model <- train(
  Group2 ~ .,
  data      = train_dat,
  method    = "glm",
  family    = "binomial",
  trControl = train_control)
```

Feature Importance Insights

As shown in figure below, MMSE receives the highest importance in both Random Forest and XGBoost, which is expected because it directly measures cognitive impairment and declines substantially in dementia. Although eTIV shows slight variation in importance between the two models, the top three predictors remain the same in both models: MMSE, nWBV, and Age, consistent with clinical expectations.




Feature Importance of XGBoost
MMSE
nWBV
Age
eTIV
Educ
SES
M.FM
ASF
0
20
40
60
80
100
Importance