Dataset_D_Model

Group_K_

2025-10-03

1) Load & Prepare Dataset

```r
set.seed(123)

library(dplyr)
library(ggplot2)
library(caret)
library(rpart)
library(rpart.plot)
library(randomForest)
library(tidyr)
library(broom)

# Option A (recommended): forward slashes
csv_path <- "C:/Users/601277/OneDrive - belgiumcampus.ac.za/Desktop/BIN381-Project/merged datasets/Grou

# Option B: escaped backslashes (either A or B, not both)
# csv_path <- "C:\\Users\\601277\\OneDrive - belgiumcampus.ac.za\\Desktop\\BIN381-Project\\merged datas

# Load Dataset D only
df <- read.csv(csv_path, stringsAsFactors = FALSE)

# Minimal, targeted cleaning and typing
df <- df %>%
  mutate(
    CharacteristicCategory = as.factor(CharacteristicCategory),
    CharacteristicLabel    = as.factor(CharacteristicLabel),
    IndicatorId            = as.factor(IndicatorId),
    SurveyYear             = as.integer(SurveyYear)
  ) %>%
  filter(!is.na(Value))

# Quick outcome overview
summary(df$Value)  # <- R uses # for comments
```

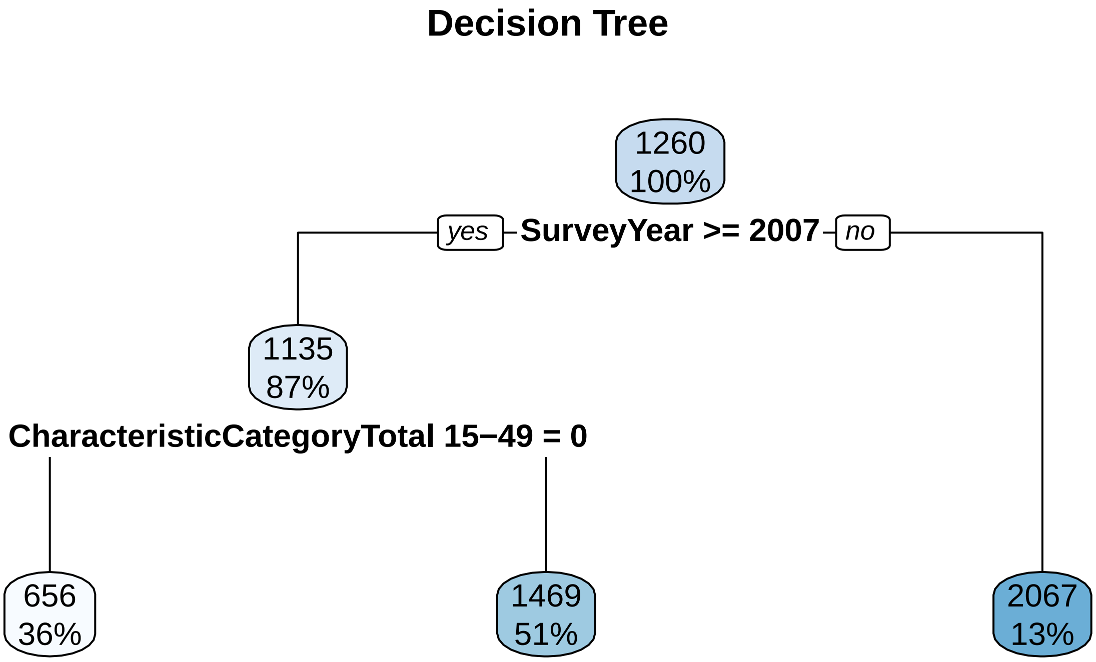
```
##     Min.  1st Qu.   Median     Mean  3rd Qu.     Max.
##     0.00    38.12   250.00  1348.20  1952.50 11805.00
```

2) Train/Test Split

```
set.seed(123)
idx   <- createDataPartition(df$Value, p = 0.7, list = FALSE)
train <- df[idx, ]
test  <- df[-idx, ]
```

3) Model 1 — Decision Tree (Regression)

```
# 10-fold CV
ctrl <- trainControl(method = "cv", number = 10)

tree_fit <- train(
  Value ~ CharacteristicCategory + CharacteristicLabel + IndicatorId + SurveyYear,
  data = train,
  method = "rpart",
  trControl = ctrl,
  tuneLength = 10,
  metric = "RMSE"
)

# Plot the final decision tree
rpart.plot(tree_fit$finalModel, main = "Decision Tree")
```



```r
# Predict & evaluate
tree_pred    <- predict(tree_fit, newdata = test)
tree_metrics <- postResample(tree_pred, test$Value)
tree_metrics
```

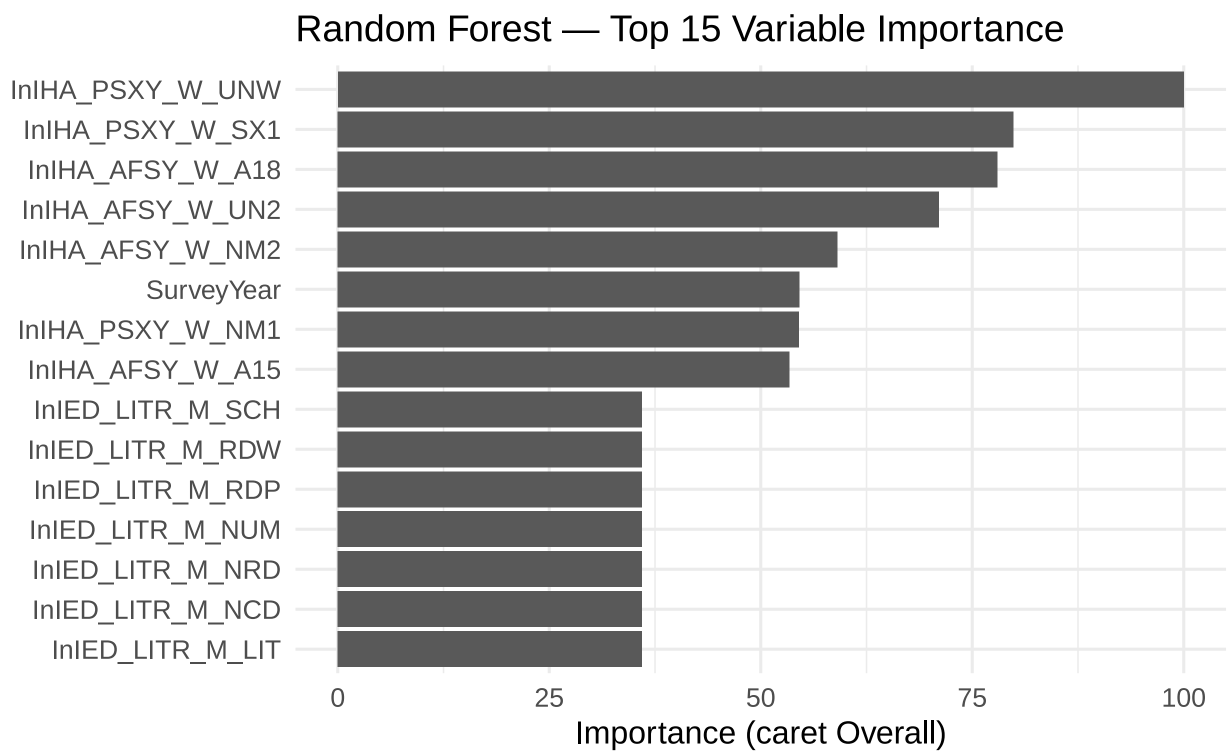
```
##         RMSE     Rsquared          MAE
## 2.579979e+03 2.881919e-02 1.649724e+03
```

4) Model 2 — Random Forest (Regression)

```r
# 4) Model 2 - Random Forest (Regression)
ctrl <- trainControl(method = "cv", number = 10)

# Train Random Forest
rf_fit <- train(
  Value ~ CharacteristicCategory + CharacteristicLabel + IndicatorId + SurveyYear,
  data = train,
  method = "rf",
  trControl = ctrl,
  tuneLength = 5,
  ntree = 500,
  importance = TRUE,
  metric = "RMSE"
)

# Predict & evaluate
rf_pred    <- predict(rf_fit, newdata = test)
rf_metrics <- postResample(rf_pred, test$Value)
rf_metrics
```

```
##         RMSE     Rsquared          MAE
## 2.597942e+03 7.944813e-02 1.631028e+03
```

```r
# ---- Clean variable-importance plot (top 15) ----
imp <- caret::varImp(rf_fit)$importance
imp$Variable <- rownames(imp); rownames(imp) <- NULL

imp %>%
  arrange(desc(Overall)) %>%
  slice_head(n = 15) %>%
  mutate(Variable = abbreviate(Variable, minlength = 16)) %>%
  ggplot(aes(x = reorder(Variable, Overall), y = Overall)) +
  geom_col() +
  coord_flip() +
  labs(title = "Random Forest - Top 15 Variable Importance",
       x = NULL, y = "Importance (caret Overall)") +
  theme_minimal(base_size = 12)
```



5) Compare Models

```r
results <- data.frame(
  Model = c("Decision Tree", "Random Forest"),
  RMSE  = c(tree_metrics["RMSE"],     rf_metrics["RMSE"]),
  R2    = c(tree_metrics["Rsquared"], rf_metrics["Rsquared"])
)

knitr::kable(results, caption = "Model Performance on Test Set")
```

Table 1: Model Performance on Test Set

Model	RMSE	R2
Decision Tree	2579.979	0.0288192
Random Forest	2597.942	0.0794481

6) Model 3 — Logistic Regression (Literacy vs Condom Use)

```r
# Select literacy-related rows (IndicatorId contains "lit" or "read")
literacy_rows <- df %>%
  filter(grepl("lit|read", IndicatorId, ignore.case = TRUE)) %>%
  transmute(SurveyYear, CharacteristicCategory, CharacteristicLabel, LiteracyValue = Value)
```

```r
# Select condom-related rows (IndicatorId contains "condom" or "cond")
condom_rows <- df %>%
  filter(grepl("condom|cond", IndicatorId, ignore.case = TRUE)) %>%
  transmute(SurveyYear, CharacteristicCategory, CharacteristicLabel, CondomValue = Value)

# Join literacy and condom subsets
joined <- inner_join(
  literacy_rows, condom_rows,
  by = c("SurveyYear","CharacteristicCategory","CharacteristicLabel")
) %>% tidyr::drop_na()

cat("Matched rows (both literacy & condom present):", nrow(joined), "\n")
```

```
## Matched rows (both literacy & condom present): 0
```

```r
if (nrow(joined) >= 20) {
  # Binarize Condom Use by median threshold
  thr <- median(joined$CondomValue, na.rm = TRUE)
  joined <- joined %>%
    mutate(CondomUse = factor(ifelse(CondomValue >= thr, 1, 0)))

  # Fit logistic regression
  logit_model <- glm(CondomUse ~ LiteracyValue + SurveyYear,
                     data = joined, family = binomial)

  # Summary
  summary(logit_model)

  # Odds Ratios with 95% CI
  exp(cbind(OR = coef(logit_model), confint(logit_model)))
} else {
  cat("Not enough matched rows to run logistic regression.\n")
}
```

```
## Not enough matched rows to run logistic regression.
```

7) Reproducibility

```r
sessionInfo()
```

```
## R version 4.4.2 (2024-10-31 ucrt)
## Platform: x86_64-w64-mingw32/x64
## Running under: Windows Server 2022 x64 (build 26100)
##
## Matrix products: default
##
##
## locale:
## [1] LC_COLLATE=English_United States.utf8
## [2] LC_CTYPE=English_United States.utf8
## [3] LC_MONETARY=English_United States.utf8
## [4] LC_NUMERIC=C
## [5] LC_TIME=English_United States.utf8
##
## time zone: Africa/Johannesburg
## tzcode source: internal
##
## attached base packages:
## [1] stats     graphics  grDevices utils     datasets  methods   base
##
## other attached packages:
## [1] broom_1.0.10         tidyr_1.3.1          randomForest_4.7-1.2
## [4] rpart.plot_3.1.3     rpart_4.1.24         caret_7.0-1
## [7] lattice_0.22-6       ggplot2_4.0.0        dplyr_1.1.4
##
## loaded via a namespace (and not attached):
##  [1] gtable_0.3.6         xfun_0.53            recipes_1.3.1
##  [4] vctrs_0.6.5          tools_4.4.2          generics_0.1.4
##  [7] stats4_4.4.2         parallel_4.4.2       tibble_3.3.0
## [10] pkgconfig_2.0.3      ModelMetrics_1.2.2.2 Matrix_1.7-1
## [13] data.table_1.17.8    RColorBrewer_1.1-3   S7_0.2.0
## [16] lifecycle_1.0.4      compiler_4.4.2       farver_2.1.2
## [19] stringr_1.5.2        codetools_0.2-20     htmltools_0.5.8.1
## [22] class_7.3-22         yaml_2.3.10          prodlim_2025.04.28
## [25] pillar_1.11.1        MASS_7.3-61          gower_1.0.2
## [28] iterators_1.0.14     foreach_1.5.2        nlme_3.1-166
## [31] parallelly_1.45.1    lava_1.8.1           tidyselect_1.2.1
## [34] digest_0.6.37        stringi_1.8.7        future_1.67.0
## [37] reshape2_1.4.4       purrr_1.1.0          listenv_0.9.1
## [40] labeling_0.4.3       splines_4.4.2        fastmap_1.2.0
## [43] grid_4.4.2           cli_3.6.5            magrittr_2.0.4
## [46] survival_3.7-0       future.apply_1.20.0  withr_3.0.2
## [49] backports_1.5.0      scales_1.4.0         lubridate_1.9.4
## [52] timechange_0.3.0     rmarkdown_2.30       globals_0.18.0
## [55] nnet_7.3-19          timeDate_4041.110    evaluate_1.0.5
## [58] knitr_1.50           hardhat_1.4.2        rlang_1.1.6
## [61] Rcpp_1.1.0           glue_1.8.0           pROC_1.19.0.1
## [64] ipred_0.9-15         rstudioapi_0.17.1    R6_2.6.1
## [67] plyr_1.8.9
```