Task 4: Feature Engineering & Data Preparation

Juan Oosthuizen

2025-09-19

Table of Contents

# 1 Load and Combine Datasets

# Load the 7 selected cleaned datasets  
selected\_datasets <- c(  
 "access-to-health-care\_national\_zaf", "immunization\_national\_zaf", "hiv-behavior\_national\_zaf",  
 "water\_national\_zaf", "dhs-quickstats\_national\_zaf", "toilet-facilities\_national\_zaf", "child-mortality-rates\_national\_zaf"  
)  
  
cleaned\_csv\_files <- paste0(file.path(cleaned\_data\_path, selected\_datasets), "\_final.csv")  
if (length(cleaned\_csv\_files[file.exists(cleaned\_csv\_files)]) == 0) {  
 cleaned\_csv\_files <- paste0(file.path(cleaned\_data\_path, selected\_datasets), ".csv")  
}  
  
# Load and combine datasets  
datasets <- lapply(cleaned\_csv\_files[file.exists(cleaned\_csv\_files)], function(file\_path) {  
 df <- read\_csv(file\_path, show\_col\_types = FALSE)  
 df$dataset\_source <- tools::file\_path\_sans\_ext(basename(file\_path))  
 return(df)  
})  
  
combined\_df <- bind\_rows(datasets)  
cat("Combined", length(datasets), "datasets:", nrow(combined\_df), "records,", ncol(combined\_df), "fields\n")

## Combined 7 datasets: 747 records, 12 fields

# 2 Feature Engineering

df\_features <- combined\_df  
  
# Categorical encoding  
df\_features$indicator\_encoded <- as.numeric(as.factor(df\_features$indicator))  
df\_features$survey\_cohort <- as.numeric(as.factor(df\_features$denominator\_unweighted))  
df\_features$dataset\_source\_encoded <- as.numeric(as.factor(df\_features$dataset\_source))  
  
# Group rare categories in by\_variable\_id  
by\_var\_counts <- table(df\_features$by\_variable\_id)  
rare\_threshold <- RARE\_CATEGORY\_THRESHOLD \* nrow(df\_features)  
rare\_categories <- names(by\_var\_counts[by\_var\_counts < rare\_threshold])  
df\_features$by\_variable\_id\_grouped <- ifelse(  
 df\_features$by\_variable\_id %in% rare\_categories, "Other", df\_features$by\_variable\_id  
)  
  
# Create dummy variables  
by\_var\_dummies <- model.matrix(~ by\_variable\_id\_grouped - 1, data = df\_features)  
type\_dummies <- model.matrix(~ indicator\_type - 1, data = df\_features)  
char\_dummies <- model.matrix(~ characteristic\_category - 1, data = df\_features)  
colnames(by\_var\_dummies) <- paste0("by\_var\_", gsub("by\_variable\_id\_grouped", "", colnames(by\_var\_dummies)))  
colnames(type\_dummies) <- paste0("type\_", gsub("indicator\_type", "", colnames(type\_dummies)))  
colnames(char\_dummies) <- paste0("char\_", gsub("characteristic\_category", "", colnames(char\_dummies)))  
  
cat("Encoded categorical variables:", ncol(by\_var\_dummies) + ncol(type\_dummies) + ncol(char\_dummies), "dummy variables created\n")

## Encoded categorical variables: 14 dummy variables created

# 3 Numeric Feature Engineering and Scaling

# Create engineered numeric features  
df\_features$high\_precision <- as.numeric(df\_features$precision <= 1)  
df\_features$char\_order\_quintile <- if(max(df\_features$characteristic\_order, na.rm = TRUE) > 10) {  
 ntile(df\_features$characteristic\_order, 5)  
} else {  
 df\_features$characteristic\_order  
}  
  
df\_features$indicator\_importance <- case\_when(  
 df\_features$indicator\_order <= 3 ~ "High",  
 df\_features$indicator\_order <= 6 ~ "Medium",  
 TRUE ~ "Low"  
)  
  
# Target variable processing  
value\_skewness <- skewness(df\_features$value, na.rm = TRUE)  
df\_features$value\_log <- if(abs(value\_skewness) > 2) log1p(abs(df\_features$value)) else df\_features$value  
df\_features$value\_category <- cut(df\_features$value, breaks = quantile(df\_features$value, c(0, 0.33, 0.67, 1), na.rm = TRUE),  
 labels = c("Low", "Medium", "High"), include.lowest = TRUE)  
  
# Additional engineered features  
df\_features$sample\_size\_tier <- case\_when(  
 df\_features$denominator\_unweighted < SAMPLE\_SIZE\_SMALL ~ "Small",  
 df\_features$denominator\_unweighted < SAMPLE\_SIZE\_LARGE ~ "Medium",  
 TRUE ~ "Large"  
)  
df\_features$data\_quality\_score <- (df\_features$high\_precision \* 0.6) + (df\_features$is\_preferred \* 0.4)  
  
# Scale numeric variables  
numeric\_vars <- c("value\_log", "precision", "characteristic\_order", "indicator\_order", "data\_quality\_score")  
scaled\_data <- df\_features[numeric\_vars] %>% mutate(across(everything(), ~ as.numeric(scale(.))))  
names(scaled\_data) <- paste0(names(scaled\_data), "\_scaled")  
  
cat("Created", ncol(scaled\_data), "scaled numeric features\n")

## Created 5 scaled numeric features

cat("Target variable skewness:", round(value\_skewness, 3),  
 if(abs(value\_skewness) > 2) " (log transformed)" else " (no transform)", "\n")

## Target variable skewness: 8.313 (log transformed)

# 4 Create Final Dataset

# Combine all features into final dataset  
all\_dummies <- cbind(by\_var\_dummies, type\_dummies, char\_dummies)  
  
final\_features <- bind\_cols(  
 df\_features %>% select(data\_id, value, value\_log, value\_category),  
 scaled\_data,  
 df\_features %>% select(is\_preferred, high\_precision, indicator\_encoded, survey\_cohort, dataset\_source\_encoded),  
 df\_features %>% select(char\_order\_quintile, indicator\_importance, sample\_size\_tier),  
 as.data.frame(all\_dummies)  
)  
  
# Create modeling-ready dataset (features only)  
modeling\_features <- final\_features %>% select(-data\_id, -value, -value\_category)  
  
# Export datasets  
write\_csv(final\_features, file.path(outputs\_path, "final\_features\_comprehensive.csv"))  
write\_csv(modeling\_features, file.path(outputs\_path, "modeling\_features.csv"))  
  
# Summary  
cat("Final dataset:", nrow(final\_features), "records,", ncol(modeling\_features), "features\n")

## Final dataset: 747 records, 28 features

cat("- Scaled numeric:", ncol(scaled\_data), "\n")

## - Scaled numeric: 5

cat("- Categorical encoded:", ncol(all\_dummies) + 6, "\n")

## - Categorical encoded: 20

cat("- Exported: modeling\_features.csv (ready for ML)\n")

## - Exported: modeling\_features.csv (ready for ML)

# 5 Dataset Preview

cat("Final training file: modeling\_features.csv\n")

## Final training file: modeling\_features.csv

cat("Records:", nrow(modeling\_features), "| Features:", ncol(modeling\_features), "\n\n")

## Records: 747 | Features: 28

cat("Sample data (first 5 rows, first 8 columns):\n")

## Sample data (first 5 rows, first 8 columns):

head(modeling\_features[1:8], 5)

## # A tibble: 5 × 8  
## value\_log value\_log\_scaled precision\_scaled characteristic\_order\_scaled  
## <dbl> <dbl> <dbl> <dbl>  
## 1 3.38 -0.303 0.660 -0.323  
## 2 3.43 -0.282 0.660 -0.323  
## 3 3.34 -0.320 0.660 -0.323  
## 4 4.21 0.0386 0.660 -0.323  
## 5 4.19 0.0288 0.660 -0.323  
## # ℹ 4 more variables: indicator\_order\_scaled <dbl>,  
## # data\_quality\_score\_scaled <dbl>, is\_preferred <dbl>, high\_precision <dbl>