phase2_classification.R

abdulbasir

2025-12-09

```r
# phase 2: therapeutic area classification
#
# implements multi-stage classification of drugs into oncology vs other
# therapeutic areas using keyword matching and supplemental methods

# sourcing configuration and utilities
source("config.R")
source("utils.R")

# loading required libraries
suppressPackageStartupMessages({
  library(dplyr)
  library(readr)
  library(stringr)
})

print_section_header("Phase 2: Therapeutic Area Classification")
```

```
##
## ======================================================================
## Phase 2: Therapeutic Area Classification
## ======================================================================
```

```r
# 1. defining oncology keywords
cat(paste("Defined", length(ONCOLOGY_KEYWORDS), "oncology keywords for classification\n"))
```

```
## Defined 53 oncology keywords for classification
```

```r
cat("Sample keywords:", paste(ONCOLOGY_KEYWORDS[1:10], collapse = ", "), "\n")
```

```
## Sample keywords: cancer, tumor, tumour, carcinoma, leukemia, leukaemia, lymphoma, melanoma, myeloma,
```

```r
# 2. loading cleaned data from phase 1
clean_fda_data = load_csv(CLEAN_DATA_FILE)

# 3. inspecting orphan drug designation column
orphan_unique = clean_fda_data$`Orphan Drug Designation` %>% unique() %>% length()
orphan_nulls = sum(is.na(clean_fda_data$`Orphan Drug Designation`))

cat("Orphan drug designation column analysis:\n")
```

```
## Orphan drug designation column analysis:
```

```r
cat(paste("  Unique values:", orphan_unique, "\n"))
```

```
##   Unique values: 5
```

```r
cat(paste("  Null values:", orphan_nulls, "\n"))
```

```
##   Null values: 0
```

```r
cat("Top 10 values:\n")
```

```
## Top 10 values:
```

```r
print(head(table(clean_fda_data$`Orphan Drug Designation`), 10))
```

```
##
##                                No                               Yes         Yes (indication [A] only)
##                               889                               438                                 3
## Yes (indication [B] and [C] only)         Yes (indication [B] only)
##                                 1                                 4
```

```r
# 4. performing primary classification based on abbreviated indication

classify_by_indication = function(indication_text, keywords) {
  if (is.na(indication_text) || indication_text == "") {
    return(NA_character_)
  }

  indication_lower = tolower(as.character(indication_text))

  for (keyword in keywords) {
    if (grepl(keyword, indication_lower, fixed = TRUE)) {
      return("Oncology")
    }
  }

  return("Other")
}

primary_classification = sapply(
  clean_fda_data$`Abbreviated Indication(s)`,
  function(x) classify_by_indication(x, ONCOLOGY_KEYWORDS)
)

oncology_count = sum(primary_classification == "Oncology", na.rm = TRUE)
other_count = sum(primary_classification == "Other", na.rm = TRUE)
missing_count = sum(is.na(primary_classification))

cat("Primary classification (indication matching) complete:\n")
```

```
## Primary classification (indication matching) complete:
```

```r
cat(paste("  Oncology:", oncology_count, "drugs\n"))
```

```
##   Oncology: 98 drugs
```

```r
cat(paste("  Other:", other_count, "drugs\n"))
```

```
##   Other: 561 drugs
```

```r
cat(paste("  Missing indication text:", missing_count, "drugs\n"))
```

```
##   Missing indication text: 676 drugs
```

```r
# 5. performing supplemental classification based on approved use(s)

classify_by_approved_use = function(approved_use_text, keywords) {
  if (is.na(approved_use_text) || approved_use_text == "") {
    return(NA_character_)
  }

  approved_use_lower = tolower(as.character(approved_use_text))

  for (keyword in keywords) {
    if (grepl(keyword, approved_use_lower, fixed = TRUE)) {
      return("Oncology")
    }
  }

  return(NA_character_)
}

approved_use_classification = sapply(
  clean_fda_data$`Approved Use(s)`,
  function(x) classify_by_approved_use(x, ONCOLOGY_KEYWORDS)
)

approved_use_oncology = sum(approved_use_classification == "Oncology", na.rm = TRUE)

cat("Approved use(s) supplemental classification:\n")
```

```
## Approved use(s) supplemental classification:
```

```r
cat(paste("  Oncology matches in approved use(s):", approved_use_oncology, "total\n"))
```

```
##   Oncology matches in approved use(s): 201 total
```

```r
# 6. performing tertiary classification based on orphan drug designation
# if orphan designation is "Yes", search approved use(s) for oncology keywords

classify_by_orphan = function(orphan_status, approved_use_text, indication_text, keywords) {
  # only proceed if orphan designation is "Yes" or "yes"
  if (is.na(orphan_status) || !(orphan_status %in% c("Yes", "yes"))) {
    return(NA_character_)
  }

  # for orphan drugs, search approved use(s) first
  if (!is.na(approved_use_text) && approved_use_text != "") {
    approved_lower = tolower(as.character(approved_use_text))
    for (keyword in keywords) {
      if (grepl(keyword, approved_lower, fixed = TRUE)) {
        return("Oncology")
      }
    }
  }

  # if no match in approved uses, try indication text
  if (!is.na(indication_text) && indication_text != "") {
    indication_lower = tolower(as.character(indication_text))
    for (keyword in keywords) {
      if (grepl(keyword, indication_lower, fixed = TRUE)) {
        return("Oncology")
      }
    }
  }

  # orphan drug but no oncology keywords found - classify as Other
  return("Other")
}

orphan_classification = mapply(
  function(orphan, approved, indication) {
    classify_by_orphan(orphan, approved, indication, ONCOLOGY_KEYWORDS)
  },
  clean_fda_data$`Orphan Drug Designation`,
  clean_fda_data$`Approved Use(s)`,
  clean_fda_data$`Abbreviated Indication(s)`,
  SIMPLIFY = TRUE
)

orphan_oncology = sum(orphan_classification == "Oncology", na.rm = TRUE)
orphan_other = sum(orphan_classification == "Other", na.rm = TRUE)

cat("Orphan drug designation tertiary classification:\n")
```

```
## Orphan drug designation tertiary classification:
```

```r
cat(paste("  Oncology matches in orphan drugs:", orphan_oncology, "total\n"))
```

```
##   Oncology matches in orphan drugs: 170 total
```

```r
cat(paste("  Other matches in orphan drugs:", orphan_other, "total\n"))
```

```
##   Other matches in orphan drugs: 268 total
```

```r
# 7. merging classifications with three-stage hierarchy

final_therapeutic_area = character(nrow(clean_fda_data))
classification_confidence = character(nrow(clean_fda_data))
classification_method = character(nrow(clean_fda_data))

for (idx in 1:nrow(clean_fda_data)) {
  primary = primary_classification[idx]

  if (!is.na(primary)) {
    # stage 1: primary method succeeded (indication text)
    final_therapeutic_area[idx] = primary
    classification_confidence[idx] = "high"
    classification_method[idx] = "indication_keyword_match"
  } else {
    # stage 2: checking approved use(s)
    approved_use = approved_use_classification[idx]

    if (!is.na(approved_use)) {
      # resolved via approved use(s)
      final_therapeutic_area[idx] = approved_use
      classification_confidence[idx] = "medium"
      classification_method[idx] = "approved_use_keyword_match"
    } else {
      # stage 3: checking orphan drug designation
      orphan = orphan_classification[idx]

      if (!is.na(orphan)) {
        # resolved via orphan drug designation
        final_therapeutic_area[idx] = orphan
        classification_confidence[idx] = "medium"
        classification_method[idx] = "orphan_drug_keyword_match"
      } else {
        # stage 4: flagging as uncertain - no default to catch-all category
        final_therapeutic_area[idx] = "Uncertain"
        classification_confidence[idx] = "low"
        classification_method[idx] = "missing_indication_text"
      }
    }
  }
}

# 8. adding classification columns to dataset
classified_data = clean_fda_data %>%
  mutate(
    therapeutic_area = final_therapeutic_area,
    classification_confidence = classification_confidence,
    classification_method = classification_method
  )
```

```r
# 9. finalizing classification
final_oncology_total = sum(classified_data$therapeutic_area == "Oncology")
final_other_total = sum(classified_data$therapeutic_area == "Other")
final_uncertain_total = sum(classified_data$therapeutic_area == "Uncertain")

cat("Final multi-stage classification complete:\n")
```

```
## Final multi-stage classification complete:
```

```r
cat(paste("  Oncology:", final_oncology_total, "drugs\n"))
```

```
##   Oncology: 299 drugs
```

```r
cat(paste("  Other:", final_other_total, "drugs\n"))
```

```
##   Other: 739 drugs
```

```r
cat(paste("  Uncertain:", final_uncertain_total, "drugs\n"))
```

```
##   Uncertain: 297 drugs
```

```r
cat("Confidence distribution:\n")
```

```
## Confidence distribution:
```

```r
print(table(classified_data$classification_confidence))
```

```
## 
##   high    low medium 
##    659    297    379
```

```r
# 10. generating classification summary report
cat("\nFinal classification results\n")
```

```
## 
## Final classification results
```

```r
cat(paste("   Total drugs classified:", nrow(classified_data), "\n"))
```

```
##    Total drugs classified: 1335
```

```r
cat(sprintf("   Oncology: %d (%.1f%%)\n", final_oncology_total,
            final_oncology_total/nrow(classified_data)*100))
```

```
##    Oncology: 299 (22.4%)
```

```
cat(sprintf("   Other: %d (%.1f%%)\n", final_other_total,
            final_other_total/nrow(classified_data)*100))
```

```
##    Other: 739 (55.4%)
```

```
cat(sprintf("   Uncertain: %d (%.1f%%)\n", final_uncertain_total,
            final_uncertain_total/nrow(classified_data)*100))
```

```
##    Uncertain: 297 (22.2%)
```

```
cat("\nClassification confidence levels\n")
```

```
##
## Classification confidence levels
```

```
confidence_counts = table(classified_data$classification_confidence)
for (conf_level in c("high", "medium", "low")) {
  if (conf_level %in% names(confidence_counts)) {
    count_val = confidence_counts[conf_level]
    cat(sprintf("   %s: %d (%.1f%%)\n",
                tools::toTitleCase(conf_level),
                count_val,
                count_val/nrow(classified_data)*100))
  }
}
```

```
##    High: 659 (49.4%)
##    Medium: 379 (28.4%)
##    Low: 297 (22.2%)
```

```
cat("\nClassification methods used\n")
```

```
##
## Classification methods used
```

```
method_counts = table(classified_data$classification_method)
for (method_name in names(method_counts)) {
  cat(paste("  ", method_name, ":", method_counts[method_name], "drugs\n"))
}
```

```
##    approved_use_keyword_match : 201 drugs
##    indication_keyword_match : 659 drugs
##    missing_indication_text : 297 drugs
##    orphan_drug_keyword_match : 178 drugs
```

```
cat("\nClassification complete - no silent defaulting to catch-all categories\n")
```

```
##
## Classification complete - no silent defaulting to catch-all categories
```

```r
# 11. saving classified data
output_file = file.path(RESULTS_DIR, "fda_analysis_clean_classified.csv")
save_csv(classified_data, output_file)
```

```
## saving results to: /Users/abdulbasir/Downloads/Experimental AI/fda-oncology-approval-analysis/results
```

```r
cat("\nPhase 2 complete\n")
```

```
##
## Phase 2 complete
```