Breast Cancer Classification Study

DIN Sokheng, RA Veasna

2025-11-10

```r
library(ggplot2)
library(dplyr)
library(tidyverse)
library(corrplot)
library(ggcorrplot)
library(naniar)
library(visdat)
library(rstatix)
library(DescTools)
library(car)
library(limma)
library(survival)
library(pheatmap)
library(diptest)
library(forcats)
library(glmnet)
library(caTools)
library(pROC)
library(gridExtra)
library(smotefamily)
library(reshape2)
```

```r
source("function_model_fit.R", local = knitr::knit_global())

cat("Functions loaded successfully: fit_all_models, plot_model_comparison, apply_smote, etc.")
```

```
## Functions loaded successfully: fit_all_models, plot_model_comparison, apply_smote, etc.
```

Classification of Cancer outcome using Genetic and Clinical data

Introduction

Breast cancer outcome prediction relies on both **clinical variables** (patient, tumor, treatment information) and **genomic features** (mRNA expression). This project analyzes a dataset of 1231 patients with 24 clinical variables and 5000 high-variance genes, aiming to:

- Understand clinical variables associated with survival
- Explore gene expression characteristics

- Identify differentially expressed genes
- Detect subgroups of patients (clustering, PCA)
- Evaluate associations between clinical and genomic factors
- Perform survival analysis (Kaplan–Meier)
- Analyse multicollinearity and variable relevance
- Provide a unified understanding of prognostic factors

The main outcome is vital_status: “Alive” vs “Dead”.

Load data

```r
load("mrr_bio.Rdata")

# Load dataset
load("mrr_bio.Rdata")

# Clinical data
clinical_df <- as.data.frame(clinical_data)
genex_df    <- as.data.frame(GeneX)

cat("Clinical samples:", nrow(clinical_df), "\n")
```

```
## Clinical samples: 1231
```

```r
cat("Clinical variables:", ncol(clinical_df), "\n")
```

```
## Clinical variables: 24
```

```r
cat("Gene samples:", nrow(genex_df), "\n")
```

```
## Gene samples: 1231
```

```r
cat("Genes:", ncol(genex_df), "\n")
```

```
## Genes: 5000
```

```r
# Outcome variable
table(clinical_df$vital_status)
```

```
##
## Alive  Dead
##  1029   201
```

```
# Clinical variables: gender
unique(clinical_data$gender) # unique values
```

```
## [1] "female" "male"   NA
```

```
table(clinical_data$gender)  # frequency
```

```
##
## female   male
##   1217     13
```

Clinical data studies

Dataset Structure & Variable Types

```
numeric_vars      <- names(clinical_df)[sapply(clinical_df, is.numeric)]
categorical_vars  <- names(clinical_df)[sapply(clinical_df, is.character)]

cat("Numeric variables (", length(numeric_vars), "):\n", paste(numeric_vars, collapse=", "), "\n\n")
```

```
## Numeric variables ( 5 ):
##  initial_weight, age_at_diagnosis, days_to_last_follow_up, age_at_index, days_to_birth
```

```
cat("Categorical variables (", length(categorical_vars), "):\n", paste(categorical_vars, collapse=", ")
```

```
## Categorical variables ( 14 ):
##  tissue_type, laterality, tissue_or_organ_of_origin, primary_diagnosis, prior_treatment, ajcc_pathol
```

Missing Data Analysis

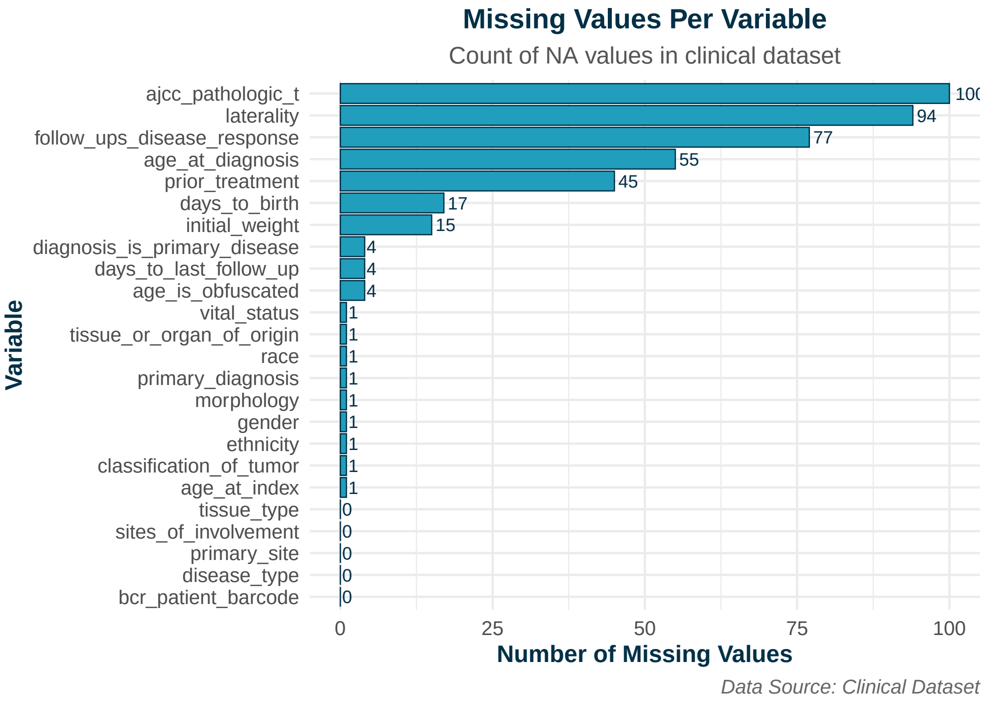
```
# Calculate missing statistics
total_missing <- sum(is.na(clinical_df))
total_cells   <- prod(dim(clinical_df))
missing_ratio <- total_missing / total_cells
missing_count <- colSums(is.na(clinical_df))
missing_pct   <- round(missing_count / nrow(clinical_df) * 100, 2)

missing_table <- data.frame(
  Column          = names(clinical_df)
  , Missing_Count = missing_count
  , Missing_Pct   = missing_pct
)

# Filter columns with missing values
missing_table_filtered <- missing_table[missing_table$Missing_Count > 0, ]

cat("Variables with missing data:\n")
```

```
## Variables with missing data:
```

```r
print(missing_table_filtered[order(-missing_table_filtered$Missing_Count), ])
```

```
##                                                    Column Missing_Count
## ajcc_pathologic_t                       ajcc_pathologic_t           100
## laterality                                     laterality            94
## follow_ups_disease_response   follow_ups_disease_response            77
## age_at_diagnosis                         age_at_diagnosis            55
## prior_treatment                           prior_treatment            45
## days_to_birth                               days_to_birth            17
## initial_weight                             initial_weight            15
## diagnosis_is_primary_disease diagnosis_is_primary_disease             4
## days_to_last_follow_up             days_to_last_follow_up             4
## age_is_obfuscated                       age_is_obfuscated             4
## tissue_or_organ_of_origin       tissue_or_organ_of_origin             1
## primary_diagnosis                       primary_diagnosis             1
## morphology                                     morphology             1
## classification_of_tumor           classification_of_tumor             1
## race                                                 race             1
## gender                                             gender             1
## ethnicity                                       ethnicity             1
## vital_status                                 vital_status             1
## age_at_index                                 age_at_index             1
##                              Missing_Pct
## ajcc_pathologic_t                   8.12
## laterality                          7.64
## follow_ups_disease_response         6.26
## age_at_diagnosis                    4.47
## prior_treatment                     3.66
## days_to_birth                       1.38
## initial_weight                      1.22
## diagnosis_is_primary_disease        0.32
## days_to_last_follow_up              0.32
## age_is_obfuscated                   0.32
## tissue_or_organ_of_origin           0.08
## primary_diagnosis                   0.08
## morphology                          0.08
## classification_of_tumor             0.08
## race                                0.08
## gender                              0.08
## ethnicity                           0.08
## vital_status                        0.08
## age_at_index                        0.08
```

```r
# Visualize missing data
ggplot(missing_table
       , aes(x = reorder(Column, Missing_Count), y = Missing_Count)) +
  geom_bar(stat = "identity", fill = "#219ebc", color = "#023047", linewidth = 0.3) +
  geom_text(aes(label = Missing_Count), hjust = -0.2, size = 3, color = "#023047") +
  coord_flip() +
  labs(title = "Missing Values Per Variable"
       , subtitle = "Count of NA values in clinical dataset"
       , x = "Variable"
       , y = "Number of Missing Values"
       , caption = "Data Source: Clinical Dataset") +
  theme_minimal(base_size = 12) +
  theme(plot.title = element_text(face = "bold", hjust = 0.5, color = "#023047")
        , plot.subtitle = element_text(hjust = 0.5, color = "#555555")
        , plot.caption = element_text(face = "italic", color = "#666666")
        , axis.title = element_text(face = "bold", color = "#023047"))
```



Data Cleaning

```
# Remove samples with missing vital_status
cat("Before cleaning:", nrow(clinical_data), "samples\n")
```

```
## Before cleaning: 1231 samples
```

```
valid_idx   <- !is.na(clinical_data$vital_status)
clinical_df <- clinical_data[valid_idx, ]
GeneX_df    <- GeneX[valid_idx, ]

cat("After removing:", nrow(clinical_df), "samples\n")
```

```
## After removing: 1230 samples
```

```
cat("Removed:", sum(!valid_idx), "sample(s)\n\n")
```

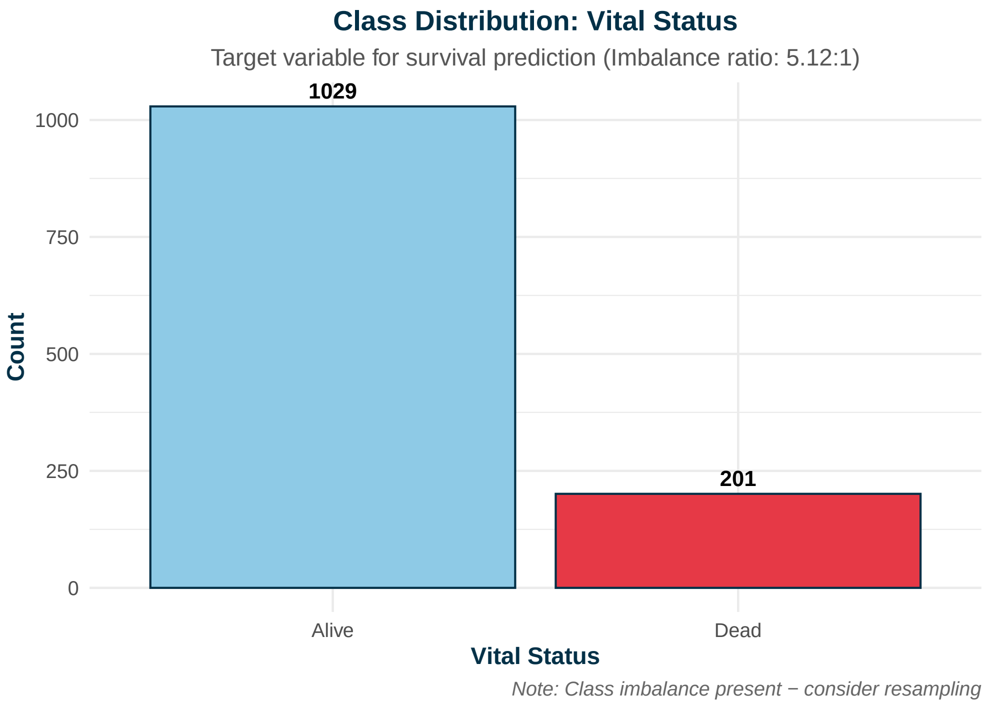
```
## Removed: 1 sample(s)
```

Key clinical predictors:

- age_at_index (continuous)
- initial_weight (continuous)
- ajcc_pathologic_t (tumor stage)
- prior_treatment (yes/no)
- primary_diagnosis (tumor subtype)
- race, gender (demographics)

Class Balance Check

```r
# Visualize class distribution
ggplot(clinical_df, aes(x = vital_status, fill = vital_status)) +
  geom_bar(color = "#023047", linewidth = 0.5) +
  geom_text(stat = "count"
            , aes(label = after_stat(count))
            , vjust = -0.5
            , fontface = "bold"
            , size = 4) +
  scale_fill_manual(values = c("Alive" = "#8ecae6", "Dead" = "#e63946")) +
  labs(title = "Class Distribution: Vital Status"
       , subtitle = "Target variable for survival prediction (Imbalance ratio: 5.12:1)"
       , x = "Vital Status"
       , y = "Count"
       , caption = "Note: Class imbalance present - consider resampling") +
  theme_minimal(base_size = 12) +
  theme(plot.title = element_text(face = "bold", hjust = 0.5, color = "#023047")
        , plot.subtitle = element_text(hjust = 0.5, color = "#555555")
        , plot.caption = element_text(face = "italic", color = "#666666")
        , axis.title = element_text(face = "bold", color = "#023047")
        , legend.position = "none")
```



```
cat("Imbalance: 5.12:1 (Alive:Dead)\n")
```

```
## Imbalance: 5.12:1 (Alive:Dead)
```

The outcome variable shows a marked imbalance of 5.12:1 (Alive : Dead), which is expected in clinical studies with right-censoring. This imbalance has no negative impact on exploratory data analysis, but it will require careful handling in later predictive modeling

Numerical Data Visualizations

```
numeric_vars <- c("age_at_index"
                  , "age_at_diagnosis"
                  , "initial_weight"
                  , "days_to_last_follow_up"
                  , "days_to_birth")

par(mfrow = c(5, 4), bg = "white", mar = c(4, 4, 3, 1))

# Store statistics values
summary_stats <- data.frame(
  Variable           = character()
  , Mean             = numeric()
  , Median           = numeric()
  , SD               = numeric()
  , Skewness         = numeric()
  , Outliers         = numeric()
  , Normality_p      = numeric()
  , Group_Diff_p     = numeric()
```

```r
  , Transform_Needed = character()
  , stringsAsFactors = FALSE
)

# Loop over each variables
for(var in numeric_vars) {
  cat(sprintf("\n========== %s ==========\n", toupper(var)))

  # Extract data
  var_data  <- clinical_df[[var]]
  var_alive <- var_data[clinical_df$vital_status == "Alive"]
  var_dead  <- var_data[clinical_df$vital_status == "Dead"]

  # Remove NA
  var_data  <- var_data[!is.na(var_data)]
  var_alive <- var_alive[!is.na(var_alive)]
  var_dead  <- var_dead[!is.na(var_dead)]

  # Statistics
  var_mean   <- mean(var_data)
  var_median <- median(var_data)
  var_sd     <- sd(var_data)
  var_min    <- min(var_data)
  var_max    <- max(var_data)

  # Skewness
  var_skew <- mean(((var_data - var_mean) / var_sd)^3)

  # Outliers (IQR method)
  Q1          <- quantile(var_data, 0.25, na.rm = TRUE)
  Q3          <- quantile(var_data, 0.75, na.rm = TRUE)
  IQR_val     <- Q3 - Q1
  lower_bound <- Q1 - 1.5 * IQR_val
  upper_bound <- Q3 + 1.5 * IQR_val

  n_outliers  <- sum(var_data < lower_bound | var_data > upper_bound)
  outlier_pct <- round(n_outliers / length(var_data) * 100, 1)

  # Normality test (Shapiro-Wilk)
  if(length(var_data) <= 5000) {
    shapiro_test <- shapiro.test(var_data)
    shapiro_p    <- shapiro_test$p.value
  } else {
    shapiro_test <- shapiro.test(sample(var_data, 5000))
    shapiro_p    <- shapiro_test$p.value
  }

  # Group difference test (t-test)
  if(length(var_alive) > 0 & length(var_dead) > 0) {
    ttest <- t.test(var_alive, var_dead)
    ttest_p <- ttest$p.value
  } else {
    ttest_p <- NA
```

```r
}

# Print Statistics
cat(sprintf("Descriptive Statistics:\n"))
cat(sprintf("  Mean:       %.2f\n", var_mean))
cat(sprintf("  Median:     %.2f\n", var_median))
cat(sprintf("  SD:         %.2f\n", var_sd))
cat(sprintf("  Range:      [%.2f, %.2f]\n", var_min, var_max))
cat(sprintf("  Skewness:   %.3f %s\n"
            , var_skew
            , ifelse(abs(var_skew) < 0.5, "(Symmetric)"
                     , ifelse(var_skew > 0, "(Right-skewed)", "(Left-skewed)"))))

cat(sprintf("  Outliers:   %d (%.1f%%)\n", n_outliers, outlier_pct))
cat(sprintf("\n"))

cat(sprintf("Statistical Tests:\n"))
cat(sprintf("  Shapiro-Wilk p-value: %.4e %s\n"
            , shapiro_p
            , ifelse(shapiro_p < 0.05, "(NOT normal)", "(Normal)")))

if(!is.na(ttest_p)) {
  cat(sprintf("  T-test (Alive vs Dead): p=%.4e %s\n"
              , ttest_p
              , ifelse(ttest_p < 0.05, "*** GROUPS DIFFER", "(No difference)")))
  cat(sprintf("    Alive: mean=%.2f, sd=%.2f\n"
              , mean(var_alive), sd(var_alive)))
  cat(sprintf("    Dead:  mean=%.2f, sd=%.2f\n"
              , mean(var_dead), sd(var_dead)))
}
cat(sprintf("\n"))

# Transformation
transform_needed <- "None"
if(abs(var_skew) > 1.0) {
  transform_needed <- "Log or sqrt (high skewness)"
} else if(outlier_pct > 5) {
  transform_needed <- "Robust scaling (many outliers)"
} else if(shapiro_p < 0.05 & abs(var_skew) > 0.5) {
  transform_needed <- "Consider log (non-normal + skewed)"
}

# Store summary
summary_stats <- rbind(summary_stats
                       , data.frame(
                         Variable           = var
                         , Mean             = var_mean
                         , Median           = var_median
                         , SD               = var_sd
                         , Skewness         = var_skew
                         , Outliers         = outlier_pct
                         , Normality_p      = shapiro_p
                         , Group_Diff_p     = ifelse(is.na(ttest_p), 1, ttest_p)
                         , Transform_Needed = transform_needed
                       ))
# Histogram
hist(var_data
     , breaks   = 40
     , col      = "#8ecae6"
     , border   = "white"
     , main     = paste(var, "- Histogram")
     , sub      = "Distribution with mean (red) and median (orange) lines"
     , xlab     = var
     , ylab     = "Frequency"
     , col.main = "#023047"
     , col.lab  = "#023047"
     , col.sub  = "#666666"
     , cex.main = 1.0
     , cex.sub  = 0.7
     , font.sub = 3)

# Mean/Median lines
abline(v   = var_mean
       , col = "#e63946"
       , lwd = 3
       , lty = 1)

abline(v   = var_median
       , col = "#fb8500"
       , lwd = 3
       , lty = 2)

# Skewness text
text(x      = var_mean
     , y    = par("usr")[4] * 0.9
     , labels = sprintf("Skew=%.2f", var_skew)
     , pos  = 4
     , col  = "#023047"
     , cex  = 0.8)

legend("topright"
       , legend = c("Mean", "Median")
       , col    = c("#e63946", "#fb8500")
       , lwd    = 3
       , lty    = c(1, 2)
       , bty    = "n"
       , cex    = 0.7)

# Density
plot(density(var_alive)
     , col      = "#219ebc"
     , lwd      = 3
     , main     = paste(var, "- Density by Status")
     , sub      = "Comparison of Alive vs Dead patient distributions"
     , xlab     = var
     , ylab     = "Density"
     , col.main = "#023047"
     , col.lab  = "#023047"
     , col.sub  = "#666666"
     , cex.main = 1.0
     , cex.sub  = 0.7
     , font.sub = 3)

lines(density(var_dead)
      , col = "#e63946"
      , lwd = 3)

# add group means
abline(v   = mean(var_alive)
       , col = "#219ebc"
       , lty = 2
       , lwd = 2)

abline(v   = mean(var_dead)
       , col = "#e63946"
       , lty = 2
       , lwd = 2)

legend("topright"
       , legend = c("Alive", "Dead")
       , col    = c("#219ebc", "#e63946")
       , lwd    = 3
       , bty    = "n"
       , cex    = 0.7)

# QQ-Plot
qqnorm(var_data
       , main     = paste(var, "- Q-Q Plot")
       , sub      = "Normality assessment: points on line = normal distribution"
       , pch      = 19
       , cex      = 0.5
       , col      = "#8ecae6"
       , col.main = "#023047"
       , col.lab  = "#023047"
       , col.sub  = "#666666"
       , cex.main = 1.0
       , cex.sub  = 0.7
       , font.sub = 3)

qqline(var_data
       , col = "#e63946"
       , lwd = 3)

# Normality
text(x      = par("usr")[1]
     , y    = par("usr")[4]
     , labels = sprintf("Shapiro p=%.2e\n%s"
                        , shapiro_p
                        , ifelse(shapiro_p < 0.05, "NON-normal", "Normal"))
       , pos  = 4
       , col  = ifelse(shapiro_p < 0.05, "#e63946", "#219ebc")
       , cex  = 0.8
       , font = 2)

  # Outliers
  boxplot(var_data ~ clinical_df$vital_status[!is.na(clinical_df[[var]])]
          , col       = c("#8ecae6", "#ffb703")
          , names     = c("Alive", "Dead")
          , main      = paste(var, "- Boxplot by Vital Status")
          , sub       = "Group comparison with outliers shown"
          , xlab      = "Vital Status"
          , ylab      = var
          , border    = c("#219ebc", "#fb8500")
          , col.main  = "#023047"
          , col.lab   = "#023047"
          , col.sub   = "#666666"
          , lwd       = 1.5
          , cex.main  = 1.0
          , cex.sub   = 0.7
          , font.sub  = 3
          , outline   = TRUE)  # Show outliers

  text(x      = 1
       , y    = par("usr")[3]
       , labels = sprintf("n=%d", length(var_alive))
       , pos  = 3
       , cex  = 0.7)

  text(x      = 2
       , y    = par("usr")[3]
       , labels = sprintf("n=%d", length(var_dead))
       , pos  = 3
       , cex  = 0.7)

  # Add p-value labels
  if(!is.na(ttest_p)) {
    text(x      = 1.5
         , y    = par("usr")[4]
         , labels = sprintf("p=%.3f %s"
                            , ttest_p
                            , ifelse(ttest_p < 0.05, "***", ""))
         , pos  = 1
         , col  = ifelse(ttest_p < 0.05, "#e63946", "#023047")
         , cex  = 0.8
         , font = 2)
  }
}
```

```
##
## ========== AGE_AT_INDEX ==========
## Descriptive Statistics:
##   Mean:       58.28
##   Median:     58.00
##   SD:         13.28
##   Range:      [26.00, 89.00]
##   Skewness:   0.146 (Symmetric)
##   Outliers:   0 (0.0%)
##
## Statistical Tests:
##   Shapiro-Wilk p-value: 4.8837e-07 (NOT normal)
##   T-test (Alive vs Dead): p=6.8789e-03 *** GROUPS DIFFER
##     Alive: mean=57.77, sd=12.77
##     Dead:  mean=60.92, sd=15.39
```

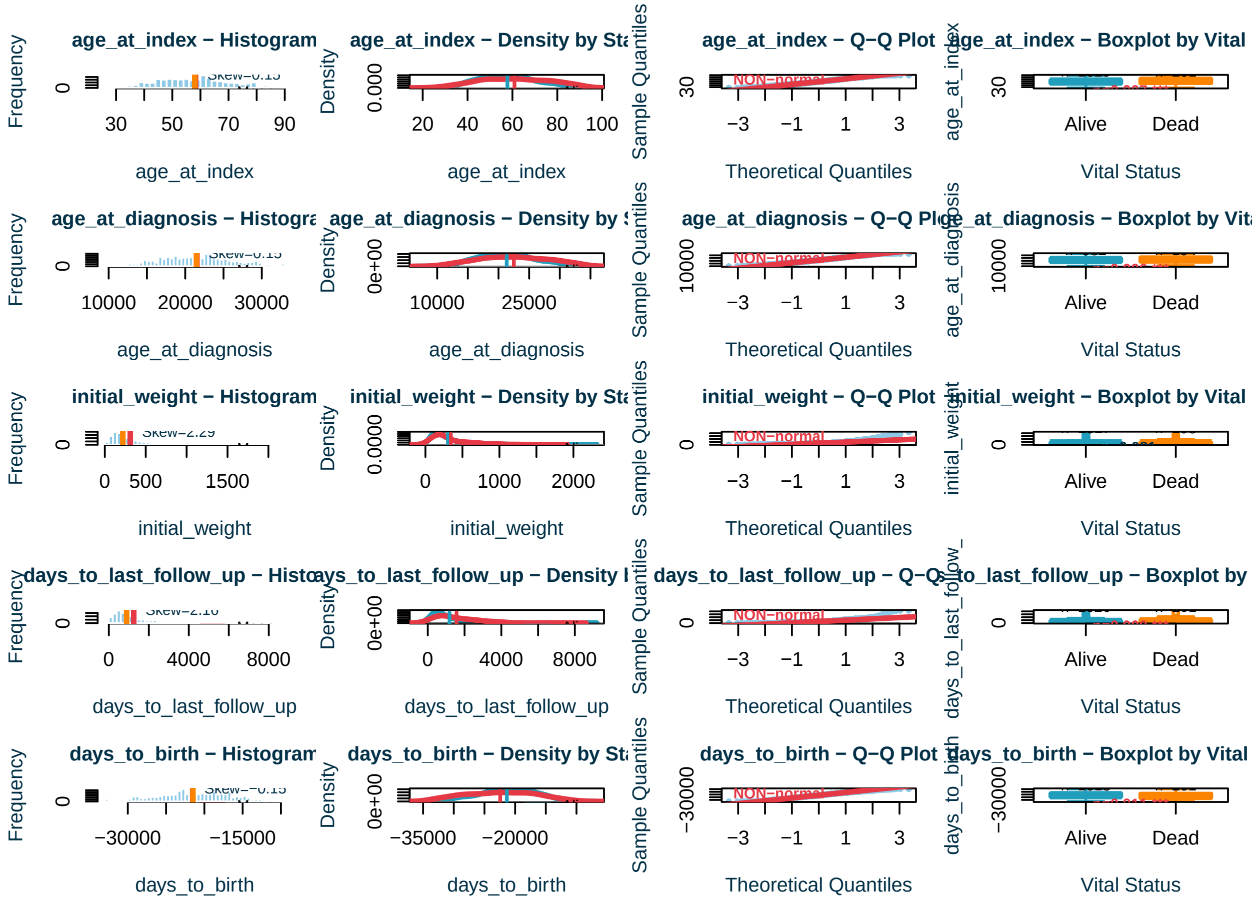
```
##
## ========== AGE_AT_DIAGNOSIS ==========
## Descriptive Statistics:
##   Mean:       21530.01
##   Median:     21472.00
##   SD:         4815.42
##   Range:      [9840.00, 32872.00]
##   Skewness:   0.147 (Symmetric)
##   Outliers:   0 (0.0%)
##
## Statistical Tests:
##   Shapiro-Wilk p-value: 4.6456e-06 (NOT normal)
##   T-test (Alive vs Dead): p=6.3766e-03 *** GROUPS DIFFER
##     Alive: mean=21337.60, sd=4639.52
##     Dead:  mean=22503.97, sd=5533.57
```

```
##
## ========== INITIAL_WEIGHT ==========
## Descriptive Statistics:
##   Mean:       310.98
##   Median:     220.00
##   SD:         272.07
##   Range:      [5.00, 2190.00]
##   Skewness:   2.293 (Right-skewed)
##   Outliers:   72 (5.9%)
##
## Statistical Tests:
##   Shapiro-Wilk p-value: 1.4492e-37 (NOT normal)
##   T-test (Alive vs Dead): p=8.1424e-02 (No difference)
##     Alive: mean=304.63, sd=268.04
##     Dead:  mean=343.61, sd=290.44
```

```
##
## ========== DAYS_TO_LAST_FOLLOW_UP ==========
## Descriptive Statistics:
##   Mean:       1245.98
##   Median:     890.00
##   SD:         1159.43
##   Range:      [-7.00, 8605.00]
##   Skewness:   2.159 (Right-skewed)
##   Outliers:   44 (3.6%)
```

```
##
## Statistical Tests:
##   Shapiro-Wilk p-value: 1.6811e-35 (NOT normal)
##   T-test (Alive vs Dead): p=6.6407e-05 *** GROUPS DIFFER
##     Alive: mean=1183.43, sd=1133.53
##     Dead:  mean=1565.26, sd=1238.10
```

```
##
## ========== DAYS_TO_BIRTH ==========
## Descriptive Statistics:
##   Mean:        -21524.60
##   Median:      -21494.50
##   SD:          4842.02
##   Range:       [-32872.00, -9706.00]
##   Skewness:    -0.146 (Symmetric)
##   Outliers:    0 (0.0%)
##
## Statistical Tests:
##   Shapiro-Wilk p-value: 1.9938e-06 (NOT normal)
##   T-test (Alive vs Dead): p=1.0727e-02 *** GROUPS DIFFER
##     Alive: mean=-21344.56, sd=4652.38
##     Dead:  mean=-22431.95, sd=5628.62
```



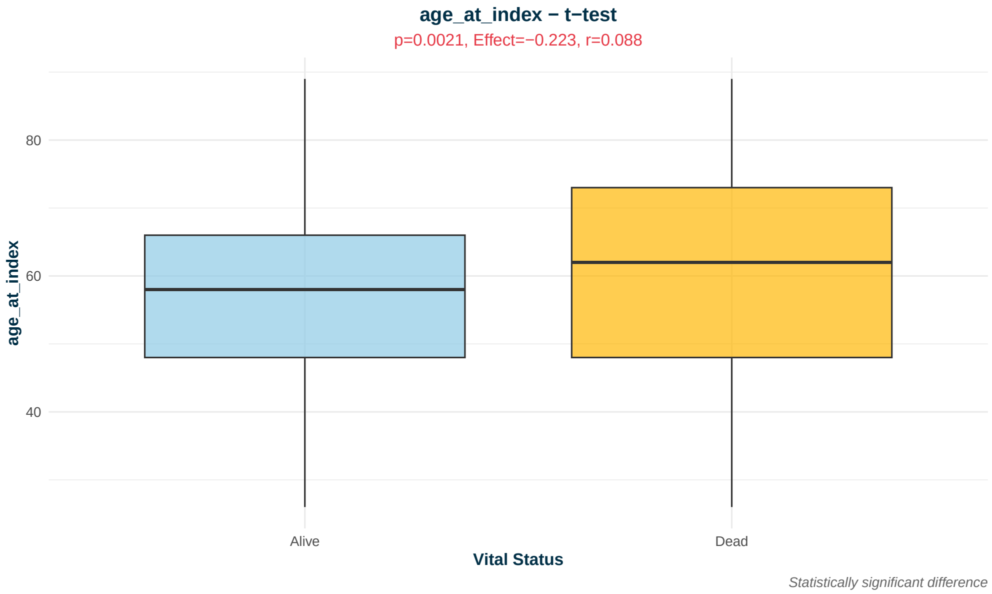
```r
par(mfrow = c(1, 1))
```

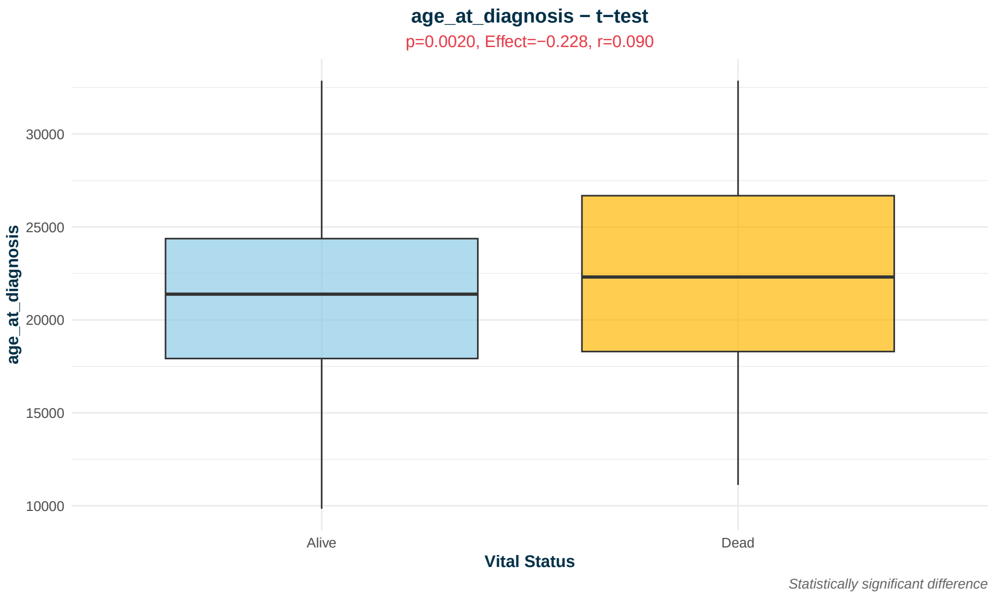
Conclusion of Numerical Variable Analysis

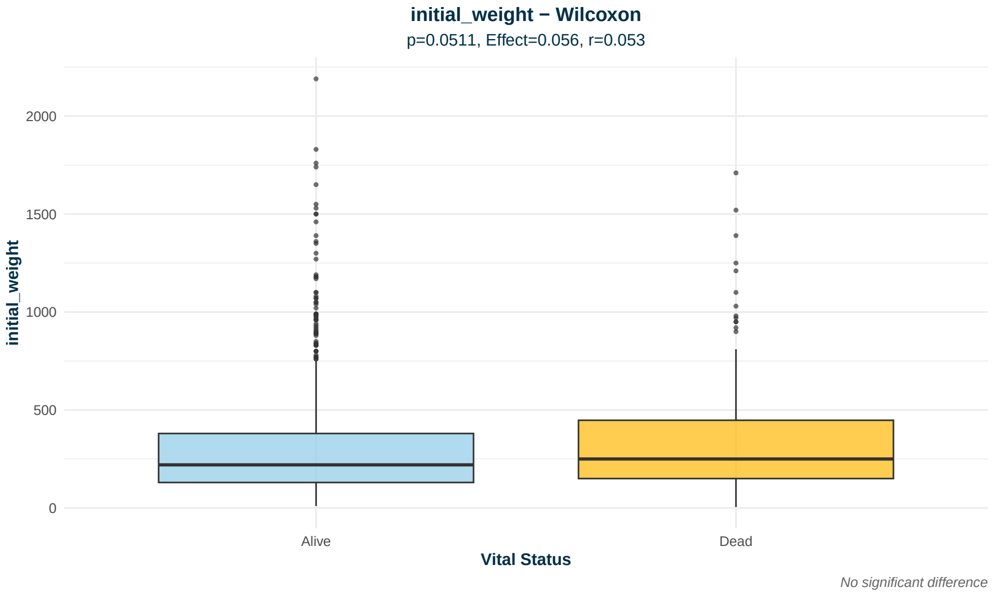
- **AGE_AT_INDEX**
  - Symmetric distribution, no outliers.
  - Not normally distributed (Shapiro p < 1e-6).
  - Significant group difference (p = 0.0069).
  - *Dead patients are older (60.9 vs 57.8).*
- **AGE_AT_DIAGNOSIS**
  - Symmetric, no outliers.
  - Not normal.
  - Significant difference (p = 0.0064).
  - *Dead patients were diagnosed at an older age.*
- **INITIAL_WEIGHT**
  - Strong right skew; many outliers (~6%).
  - Not normal.
  - No significant difference (p = 0.081).
  - *Weight does not differ between Alive/Dead groups.*
- **DAYS_TO_LAST_FOLLOW_UP**
  - Strong right-skewed distribution with outliers.
  - Not normal.
  - Significant difference (p = 6.6e-05).
  - *Dead patients show longer follow-up times (expected due to event vs censoring).*
- **DAYS_TO_BIRTH**
  - Symmetric distribution, no outliers.
  - Not normal.
  - Significant difference (p = 0.0107).
  - *Reflects age differences – Dead patients are older.*

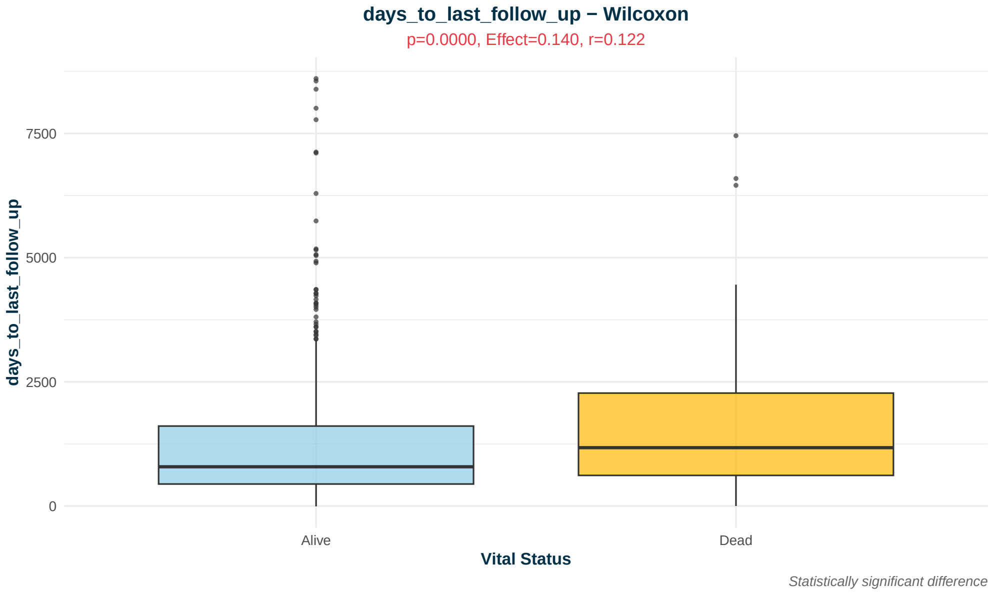
Numerical Variables - Group Comparison Tests

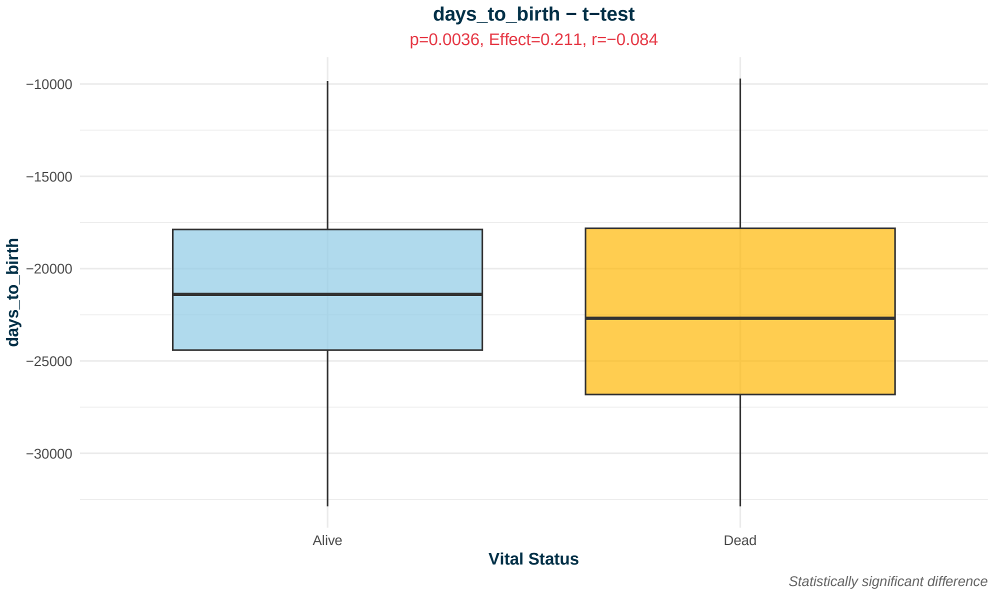
```r
clinical_df <- as.data.frame(clinical_df)

# Convert target
clinical_df$Y <- factor(clinical_df$vital_status, levels = c("Alive", "Dead"))

num_vars <- c("age_at_index"
              , "age_at_diagnosis"
              , "initial_weight"
              , "days_to_last_follow_up"
              , "days_to_birth")

# Storage for results
test_results <- data.frame(
  Variable      = character()
  , Test         = character()
  , Statistic    = numeric()
  , P_value      = numeric()
  , Effect_Size = numeric()
  , Correlation = numeric()
  , stringsAsFactors = FALSE
)

# Test each variable
for(var in num_vars) {

  valid_idx <- !is.na(clinical_df[[var]]) & !is.na(clinical_df$Y)
  df_test   <- clinical_df[valid_idx, c("Y", var)]

  # Skip if insufficient data
  if(nrow(df_test) < 10 || length(unique(df_test$Y)) < 2) next

  # --- Check skewness ---
  alive_vals <- df_test[df_test$Y == "Alive", var]
  dead_vals  <- df_test[df_test$Y == "Dead", var]

  skew_alive <- abs(mean(alive_vals) - median(alive_vals)) / IQR(alive_vals)
  skew_dead  <- abs(mean(dead_vals) - median(dead_vals)) / IQR(dead_vals)

  is_normal <- (skew_alive < 0.2 & skew_dead < 0.2)

  # --- Choose test ---
  if(is_normal) {
    # T-test
    test_res <- t.test(df_test[[var]] ~ df_test$Y, var.equal = TRUE)

    test_name <- "t-test"
    stat_val  <- test_res$statistic
    p_val     <- test_res$p.value

    # Cohen's d
    effect <- cohens_d(df_test, as.formula(paste(var, "~ Y")))$effsize

  } else {
    # Wilcoxon test
    test_res <- wilcox.test(df_test[[var]] ~ df_test$Y)

    test_name <- "Wilcoxon"
    stat_val  <- test_res$statistic
    p_val     <- test_res$p.value

    # Rank-biserial
    effect <- wilcox_effsize(df_test, as.formula(paste(var, "~ Y")))$effsize
  }

  # Point-biserial correlation
  cor_val <- cor(df_test[[var]], as.numeric(df_test$Y) - 1)
  # Store results
  test_results <- rbind(test_results
                        , data.frame(Variable    = var
                                     , Test      = test_name
                                     , Statistic = round(stat_val, 2)
                                     , P_value   = p_val
                                     , Effect_Size = round(effect, 3)
                                     , Correlation = round(cor_val, 3)))

  # --- Visualization ---
  p <- ggplot(df_test, aes(x = Y, y = .data[[var]], fill = Y)) +
    geom_boxplot(alpha = 0.7, outlier.shape = 19, outlier.size = 1) +
    scale_fill_manual(values = c("Alive" = "#8ecae6", "Dead" = "#ffb703")) +
    labs(title = paste(var, "-", test_name)
         , subtitle = sprintf("p=%.4f, Effect=%.3f, r=%.3f", p_val, effect, cor_val)
         , x = "Vital Status"
         , y = var
         , caption = ifelse(p_val < 0.05, "Statistically significant difference", "No significant diffe
    theme_minimal(base_size = 12) +
    theme(legend.position = "none"
          , plot.title = element_text(face = "bold", hjust = 0.5, color = "#023047")
          , plot.subtitle = element_text(hjust = 0.5, color = ifelse(p_val < 0.05, "#e63946", "#023047"
          , plot.caption = element_text(face = "italic", color = "#666666")
          , axis.title = element_text(face = "bold", color = "#023047"))

  print(p)
}
```




age_at_diagnosis − t−test
p=0.0020, Effect=−0.228, r=0.090
age_at_diagnosis
30000
25000
20000
15000
10000
Alive
Dead
Vital Status
Statistically significant difference


initial_weight − Wilcoxon
p=0.0511, Effect=0.056, r=0.053
initial_weight
2000
1500
1000
500
0
Alive
Dead
Vital Status
No significant difference





```r
# --- FDR correction ---
test_results$P_adj <- p.adjust(test_results$P_value, method = "fdr")

for(i in 1:nrow(test_results)) {
  row <- test_results[i, ]

  sig <- ifelse(row$P_adj < 0.001, "***"
                , ifelse(row$P_adj < 0.01, "**"
                         , ifelse(row$P_adj < 0.05, "*", "")))

  cat(sprintf("%-25s %-10s %10.2f %10.4f %10.4f %10.3f %10.3f %s\n"
              , row$Variable
              , row$Test
              , row$Statistic
```

```
               , row$P_value
               , row$P_adj
               , row$Effect_Size
               , row$Correlation
               , sig))
}
```

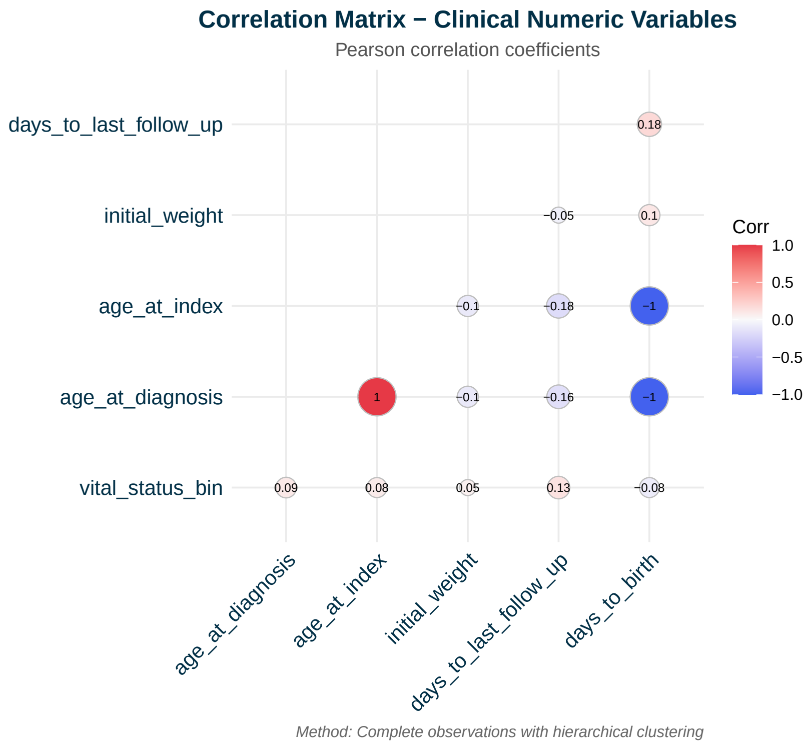
```
## age_at_index               t-test         -3.09     0.0021     0.0034     -0.223       0.088 **
## age_at_diagnosis           t-test         -3.09     0.0020     0.0034     -0.228       0.090 **
## initial_weight             Wilcoxon     91871.50     0.0511     0.0511      0.056       0.053
## days_to_last_follow_up     Wilcoxon     80651.50     0.0000     0.0000      0.140       0.122 ***
## days_to_birth              t-test          2.92     0.0036     0.0045      0.211      -0.084 **
```

According to the data analysis and the implementation of statistical tests on the clinical dataset. We can observe that columns such as **age at diagnosis, age at index, and day to birth** are the same information, which we can drop or exclude for variable selection by keeping only age at index. In addition, **initial weight** columns are also not significant for vital status target.

Clinical Correlation Matrix

```
# Convert target to numeric
clinical_base <- as.data.frame(clinical_df)
clinical_base$vital_status_bin <- ifelse(clinical_base$vital_status == "Dead", 1, 0)

# Get numeric variables
clinic_num_cols <- names(clinical_base)[sapply(clinical_base, is.numeric)]
numeric_df      <- clinical_base[, clinic_num_cols]

# Compute correlation
corr_matrix <- cor(numeric_df, use = "complete.obs")

# Plot
ggcorrplot(corr_matrix
           , hc.order = TRUE
           , lab      = TRUE
           , lab_size = 2.5
           , method   = "circle"
           , type     = "lower"
           , colors   = c("#4361ee", "#f8f9fa", "#e63946")
           , title    = "Correlation Matrix - Clinical Numeric Variables"
           , ggtheme  = theme_minimal() +
               theme(plot.title = element_text(hjust = 0.5, size = 14, face = "bold", color = "#023047"
                     , plot.subtitle = element_text(hjust = 0.5, color = "#555555")
                     , plot.caption = element_text(face = "italic", color = "#666666")
                     , axis.text = element_text(color = "#023047"))) +
  labs(subtitle = "Pearson correlation coefficients"
       , caption = "Method: Complete observations with hierarchical clustering")
```

```
## Warning: ‘aes_string()‘ was deprecated in ggplot2 3.0.0.
## i Please use tidy evaluation idioms with ‘aes()‘.
## i See also ‘vignette("ggplot2-in-packages")‘ for more information.
## i The deprecated feature was likely used in the ggcorrplot package.
##   Please report the issue at <https://github.com/kassambara/ggcorrplot/issues>.
## This warning is displayed once every 8 hours.
## Call ‘lifecycle::last_lifecycle_warnings()‘ to see where this warning was
## generated.
```



From above correlation matrix, we can interprete that

- **age_at_index**, **age_at_diagnosis**, and **days_to_birth**
  - Extremely high correlations (|r| ~= 0.99).
  - These three variables encode the **same underlying information (patient age)**.
  - Keep only one **(age_at_index)** for modeling to avoid multicollinearity.
- **days_to_last_follow_up**
  - Weak correlations with all other variables (|r| < 0.20).
  - Slight positive correlation with vital_status_bin (r ~= 0.13), expected because **Dead patients have actual event times**, while Alive patients are censored earlier.
- **initial_weight**
  - Very weak correlations with every clinical variable and with survival (|r| < 0.10). -Not a predictive feature.
- **vital_status_bin**
  - Correlates weakly with every numeric variable (|r| < 0.13).

  - No strong linear relationship; survival differences detected by group tests are **small effect sizes**, not strong correlations.

The numerical variables reveal one clear multicollinearity block: *age_at_index*, *age_at_diagnosis*, and *days_to_birth* all measure the same underlying factor (patient age) and only one should be kept. Other variables show only weak correlations with survival. *Days_to_last_follow_up* has a small association due to censoring differences, while *initial_weight* shows negligible relevance and can be excluded.

VIF Analysis for Clinical Variables

```
# Prepare clinical data
clinical_vif <- data.frame(
  age_at_index           = clinical_df$age_at_index
  , age_at_diagnosis     = clinical_df$age_at_diagnosis
  , initial_weight       = clinical_df$initial_weight
  , days_to_last_follow_up = clinical_df$days_to_last_follow_up
  , days_to_birth        = clinical_df$days_to_birth
  , vital_status_bin     = ifelse(clinical_df$vital_status == "Dead", 1, 0)
)

# Remove NA
clinical_vif <- na.omit(clinical_vif)

cat("After removing NA:", nrow(clinical_vif), "\n\n")
```

```
## After removing NA: 1161
```

```
# Fit model
full_model <- glm(vital_status_bin ~ age_at_index + age_at_diagnosis +
                    initial_weight + days_to_last_follow_up + days_to_birth
                  , data   = clinical_vif
                  , family = binomial)

# Calculate VIF
vif_values <- vif(full_model)

cat("VIF Results:\n")
```

```
## VIF Results:
```

```
print(vif_values)
```

```
##           age_at_index       age_at_diagnosis         initial_weight
##            2100.239872             149.749877               1.029426
## days_to_last_follow_up          days_to_birth
##               1.122283            2204.019112
```

```
cat("\n=== INTERPRETATION ===\n")
```

```
##
## === INTERPRETATION ===
```

```
cat("VIF < 5:  No multicollinearity\n")
```

```
## VIF < 5:  No multicollinearity
```

```
cat("VIF 5-10: Moderate multicollinearity (monitor)\n")
```

```
## VIF 5-10: Moderate multicollinearity (monitor)
```

```
cat("VIF > 10: High multicollinearity (REMOVE variable)\n\n")
```

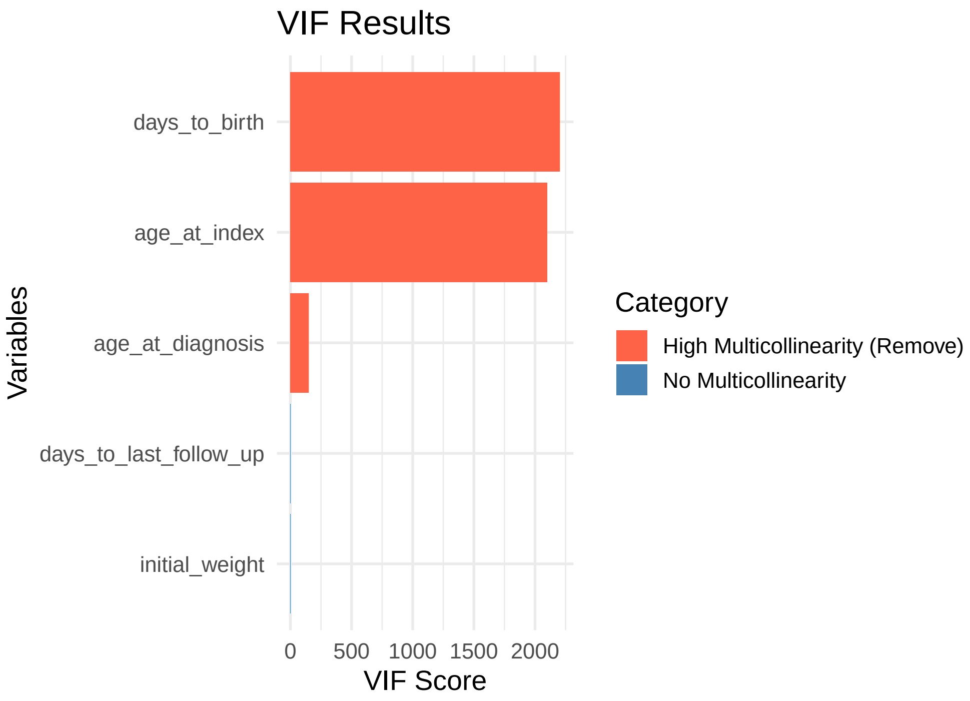
```
## VIF > 10: High multicollinearity (REMOVE variable)
```

```
# Flag problematic variables
high_vif <- names(vif_values)[vif_values > 10]
mod_vif  <- names(vif_values)[vif_values >= 5 & vif_values <= 10]

if(length(high_vif) > 0) {
  cat("HIGH VIF (>10) - REMOVE:\n")
  for(var in high_vif) {
    cat(sprintf("   %s: VIF = %.2f\n", var, vif_values[var]))
  }
  cat("\n")
}
```

```
## HIGH VIF (>10) - REMOVE:
##    age_at_index: VIF = 2100.24
##    age_at_diagnosis: VIF = 149.75
##    days_to_birth: VIF = 2204.02
```

```
if(length(mod_vif) > 0) {
  cat(" MODERATE VIF (5-10) - MONITOR:\n")
  for(var in mod_vif) {
    cat(sprintf("   %s: VIF = %.2f\n", var, vif_values[var]))
  }
  cat("\n")
}

# Convert into data frame
vif_df <- data.frame(
  Variable = names(vif_values),
  VIF = as.numeric(vif_values)
)

# Threshold for high multicollinearity (commonly VIF > 5 or > 10)
threshold <- 5

# Classify variables
```

```
vif_df$Group <- ifelse(vif_df$VIF > threshold,
                       "High Multicollinearity (Remove)",
                       "No Multicollinearity")

# Plot bar graph
ggplot(vif_df, aes(x = reorder(Variable, VIF), y = VIF, fill = Group)) +
  geom_bar(stat = "identity") +
  coord_flip() +
  scale_fill_manual(values = c(
    "No Multicollinearity" = "steelblue",
    "High Multicollinearity (Remove)" = "tomato"
  )) +
  labs(title = "VIF Results",
       x = "Variables",
       y = "VIF Score",
       fill = "Category") +
  theme_minimal(base_size = 14)
```



From VIF, it shows extreme multicollinearity among the age variables (**age__at__index, age__at__diagnosis, days__to__birth**), meaning they all represent the same information and only should be kept. The over variable (**initial__weight, days__to__last__follow__up**) have VIF~=1 and pose no multicollinearity issue.

PCA Analysis on Clinical Variables

```
clinical_pca <- data.frame(
  age_at_index           = clinical_df$age_at_index
  , age_at_diagnosis     = clinical_df$age_at_diagnosis
  , initial_weight       = clinical_df$initial_weight
  , days_to_last_follow_up = clinical_df$days_to_last_follow_up
  , days_to_birth        = clinical_df$days_to_birth
)

clinical_pca$vital_status <- clinical_df$vital_status

# Remove NA
clinical_pca <- na.omit(clinical_pca)
predictors <- clinical_pca[, 1:5]

# Run PCA (scaled)
pca_result <- prcomp(predictors, scale. = TRUE, center = TRUE)

# Variance explained
var_exp     <- summary(pca_result)$importance[2, ]
var_cum     <- summary(pca_result)$importance[3, ]

cat("Variance Explained:\n")
```

```
## Variance Explained:
```

```
for(i in 1:5) {
  cat(sprintf("  PC%d: %.1f%% (Cumulative: %.1f%%)\n"
              , i
              , var_exp[i] * 100
              , var_cum[i] * 100))
}
```

```
##   PC1: 61.1% (Cumulative: 61.1%)
##   PC2: 20.9% (Cumulative: 82.0%)
##   PC3: 17.9% (Cumulative: 99.9%)
##   PC4: 0.1% (Cumulative: 100.0%)
##   PC5: 0.0% (Cumulative: 100.0%)
```

```
cat("\n")
```

```
cat("Variable Loadings on PC1 and PC2:\n")
```

```
## Variable Loadings on PC1 and PC2:
```

```
loadings <- pca_result$rotation[, 1:2]
print(round(loadings, 3))
```

```
##                           PC1    PC2
## age_at_index            0.570  0.014
## age_at_diagnosis        0.569  0.027
## initial_weight         -0.076 -0.780
## days_to_last_follow_up -0.145  0.625
## days_to_birth          -0.570 -0.014
```

```
cat("\n")
```

```
# Interpretation
cat("=== INTERPRETATION ===\n")
```

```
## === INTERPRETATION ===
```

```
cat("PC1 captures", round(var_exp[1] * 100, 1), "% variance\n")
```

```
## PC1 captures 61.1 % variance
```

```
cat("  - High loadings:", names(sort(abs(loadings[, 1]), decreasing = TRUE)[1:2]), "\n")
```

```
##   - High loadings: days_to_birth age_at_index
```

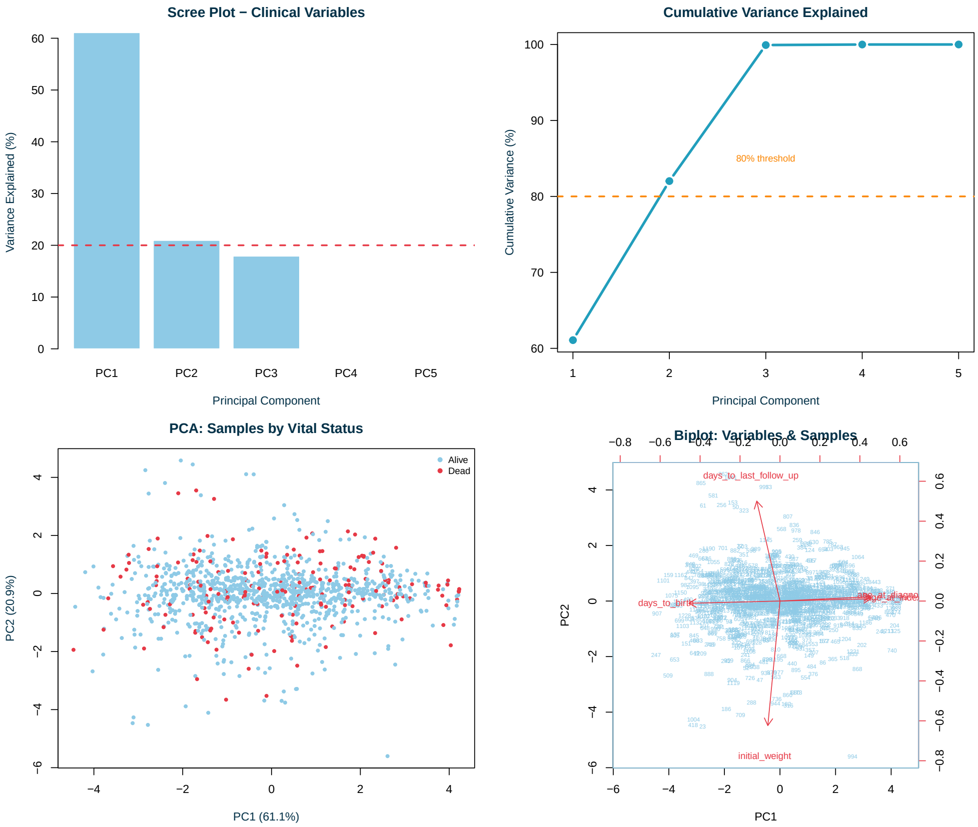
```
cat("PC2 captures", round(var_exp[2] * 100, 1), "% variance\n")
```

```
## PC2 captures 20.9 % variance
```

```
cat("  - High loadings:", names(sort(abs(loadings[, 2]), decreasing = TRUE)[1:2]), "\n\n")
```

```
##   - High loadings: initial_weight days_to_last_follow_up
```

```
par(mfrow = c(2, 2), bg = "white", mar = c(4, 4, 3, 2))

# 1. Scree Plot
barplot(var_exp * 100
        , names.arg = paste0("PC", 1:5)
        , col       = "#8ecae6"
        , border    = "white"
        , xlab      = "Principal Component"
        , ylab      = "Variance Explained (%)"
        , main      = "Scree Plot - Clinical Variables"
        , sub       = "Eigenvalue decomposition showing variance per PC"
        , col.main  = "#023047"
        , col.lab   = "#023047"
        , col.sub   = "#666666"
        , cex.sub   = 0.7
        , font.sub  = 3
        , las       = 1)

abline(h   = 20
       , col = "#e63946"
       , lty = 2
       , lwd = 2)

# 2. Cumulative Variance
plot(1:5
     , var_cum * 100
     , type     = "b"
     , pch      = 19
     , col      = "#219ebc"
     , lwd      = 3
     , xlab     = "Principal Component"
     , ylab     = "Cumulative Variance (%)"
     , main     = "Cumulative Variance Explained"
     , sub      = "Total variance captured by first n components"
     , col.main = "#023047"
     , col.lab  = "#023047"
     , col.sub  = "#666666"
     , cex.sub  = 0.7
     , font.sub = 3
     , las      = 1)

abline(h   = 80
       , col = "#fb8500"
       , lty = 2
       , lwd = 2)

text(x = 3, y = 85, labels = "80% threshold", col = "#fb8500", cex = 0.8)

# 3. PC1 vs PC2 (colored by vital status)
plot(pca_result$x[, 1]
     , pca_result$x[, 2]
     , pch      = 19
     , cex      = 0.6
     , col      = ifelse(clinical_pca$vital_status == "Dead"
                         , "#e63946", "#8ecae6")
     , xlab     = paste0("PC1 (", round(var_exp[1] * 100, 1), "%)")
     , ylab     = paste0("PC2 (", round(var_exp[2] * 100, 1), "%)")
     , main     = "PCA: Samples by Vital Status"
     , sub      = "Patient projection onto first two principal components"
     , col.main = "#023047"
     , col.lab  = "#023047"
     , col.sub  = "#666666"
     , cex.sub  = 0.7
     , font.sub = 3)

legend("topright"
       , legend = c("Alive", "Dead")
       , col    = c("#8ecae6", "#e63946")
       , pch    = 19
       , bty    = "n"
       , cex    = 0.8)

# 4. Biplot (variables + samples)
biplot(pca_result
       , choices   = 1:2
       , scale     = 0
       , col       = c("#8ecae6", "#e63946")
       , cex       = c(0.5, 0.8)
       , main      = "Biplot: Variables & Samples"
       , col.main  = "#023047"
       , arrow.len = 0.1)
```



```
par(mfrow = c(1, 1))
```

Drop Redundant Variables Based on VIF + PCA

```
drop_vars <- c("age_at_diagnosis", "days_to_birth")

# KEEP
keep_vars <- c("age_at_index"
               , "initial_weight"
               , "days_to_last_follow_up")


# Create reduced set
clinical_reduced <- data.frame(
  age_at_index           = clinical_df$age_at_index
  , initial_weight       = clinical_df$initial_weight
  , days_to_last_follow_up = clinical_df$days_to_last_follow_up
  , vital_status_bin     = ifelse(clinical_df$vital_status == "Dead", 1, 0)
)
```

```r
clinical_reduced <- na.omit(clinical_reduced)

# Fit model
reduced_model <- glm(vital_status_bin ~ age_at_index + initial_weight +
                       days_to_last_follow_up
                     , data   = clinical_reduced
                     , family = binomial)

# Calculate VIF
vif_reduced <- car::vif(reduced_model)

cat("VIF Results (Reduced Model):\n")
```

```
## VIF Results (Reduced Model):
```

```r
print(vif_reduced)
```

```
##           age_at_index         initial_weight days_to_last_follow_up
##               1.077847               1.033913               1.074717
```

After reducing some redundant variabels, we can see that the VIF is now better no more multicollinearity.

Categorical Variables Analysis

```r
clinical_df <- as.data.frame(clinical_df)

exclude_cols <- c("bcr_patient_barcode"
                  , "primary_site"
                  , "days_to_birth"
                  , "age_at_diagnosis"
                  , "sites_of_involvement"
                  , "disease_type"
                  , "vital_status"
                  , "Y")

# Select cols
cat_cols <- names(clinical_df)[sapply(clinical_df, function(x)
  is.character(x) | is.factor(x))]

cat_cols <- setdiff(cat_cols, exclude_cols)

cat("Categorical variables:", length(cat_cols), "\n")
```

```
## Categorical variables: 12
```

```r
cat(paste(cat_cols, collapse = ", "), "\n\n")
```

```
## tissue_type, laterality, tissue_or_organ_of_origin, primary_diagnosis, prior_treatment, ajcc_patholog
```

```r
# Clean and prepare
categorical_df <- clinical_df[, cat_cols, drop = FALSE]

# Convert to character
categorical_df <- data.frame(lapply(categorical_df, as.character)
                              , stringsAsFactors = FALSE)

# Mark missing values
missing_markers <- c("not reported", "not applicable", "unknown", "NA", "")

for(var in names(categorical_df)) {
  categorical_df[[var]][categorical_df[[var]] %in% missing_markers] <- NA
}

# Group categories with < 10 samples
for(var in names(categorical_df)) {

  categorical_df[[var]] <- as.factor(categorical_df[[var]])
  categorical_df[[var]] <- fct_lump_min(categorical_df[[var]]
                                         , min = 10
                                         , other_level = "Other")
  categorical_df[[var]] <- droplevels(categorical_df[[var]])

  cat(sprintf("%s: %d levels after grouping\n"
              , var
              , nlevels(categorical_df[[var]])))
}
```

```
## tissue_type: 2 levels after grouping
## laterality: 2 levels after grouping
## tissue_or_organ_of_origin: 5 levels after grouping
## primary_diagnosis: 8 levels after grouping
## prior_treatment: 3 levels after grouping
## ajcc_pathologic_t: 7 levels after grouping
## morphology: 8 levels after grouping
## classification_of_tumor: 6 levels after grouping
## follow_ups_disease_response: 3 levels after grouping
## race: 4 levels after grouping
## gender: 2 levels after grouping
## ethnicity: 3 levels after grouping
```

Statistical Tests and Visualization

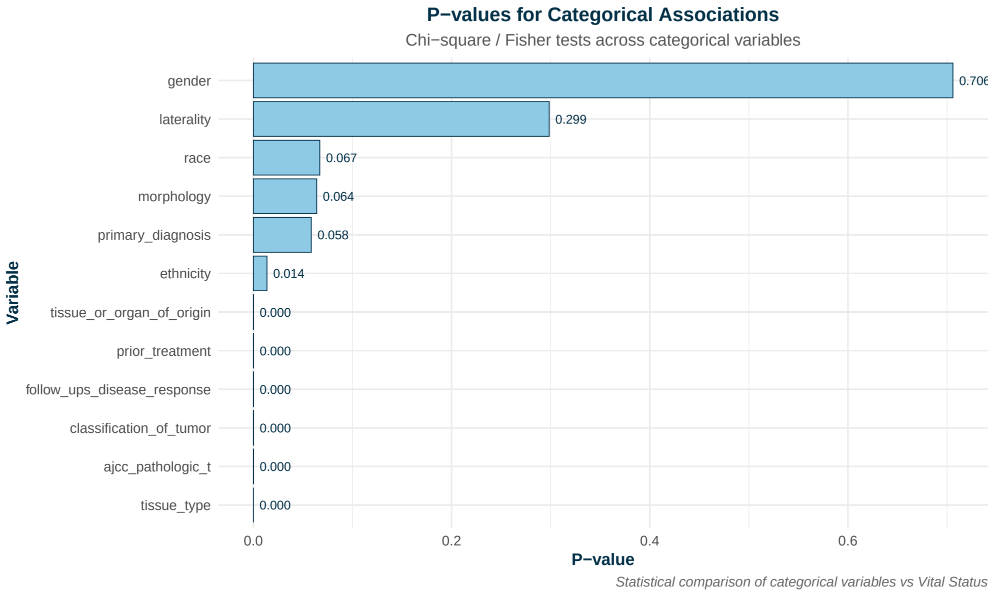
```r
# Add outcome variable to categorical_df
categorical_df$Y <- factor(clinical_df$vital_status)

results <- list()

for (var in setdiff(names(categorical_df), "Y")) {

  x <- categorical_df[[var]]
```

```r
y <- categorical_df$Y
# Skip constants
if (n_distinct(x) <= 1) {
  results[[var]] <- data.frame(
    Test="Constant variable", P_value=NA, Statistic=NA, Cramers_V=NA,
    Note="Skipped (constant)", stringsAsFactors=FALSE
  )
  next
}

# Build table
tbl <- table(x, y)

if (nrow(tbl) < 2 || ncol(tbl) < 2) {
  results[[var]] <- data.frame(
    Test="Too few levels", P_value=NA, Statistic=NA, Cramers_V=NA,
    Note="Not enough levels", stringsAsFactors=FALSE
  )
  next
}

# Expected counts
expected <- outer(rowSums(tbl), colSums(tbl)) / sum(tbl)

# Choose appropriate test
if (nrow(tbl) == 2 && ncol(tbl) == 2) {

  # Fisher for 2x2
  test <- fisher.test(tbl)
  test_name <- "Fisher Exact (2x2)"
  stat_val <- NA

} else if (all(expected >= 5)) {

  # Standard Chi-square
  test <- chisq.test(tbl, correct = FALSE)
  test_name <- "Chi-square"
  stat_val <- test$statistic

} else {

  # Monte Carlo Chi-square for sparse large contingency tables
  test <- chisq.test(tbl, simulate.p.value = TRUE, B = 10000)
  test_name <- "Chi-square (MC simulation)"
  stat_val <- test$statistic
}

pval <- test$p.value

# Cramér’s V
chi2 <- sum((tbl - expected)^2 / expected)
k <- min(nrow(tbl), ncol(tbl))
cramers_v <- sqrt(chi2 / (sum(tbl) * (k - 1)))
```

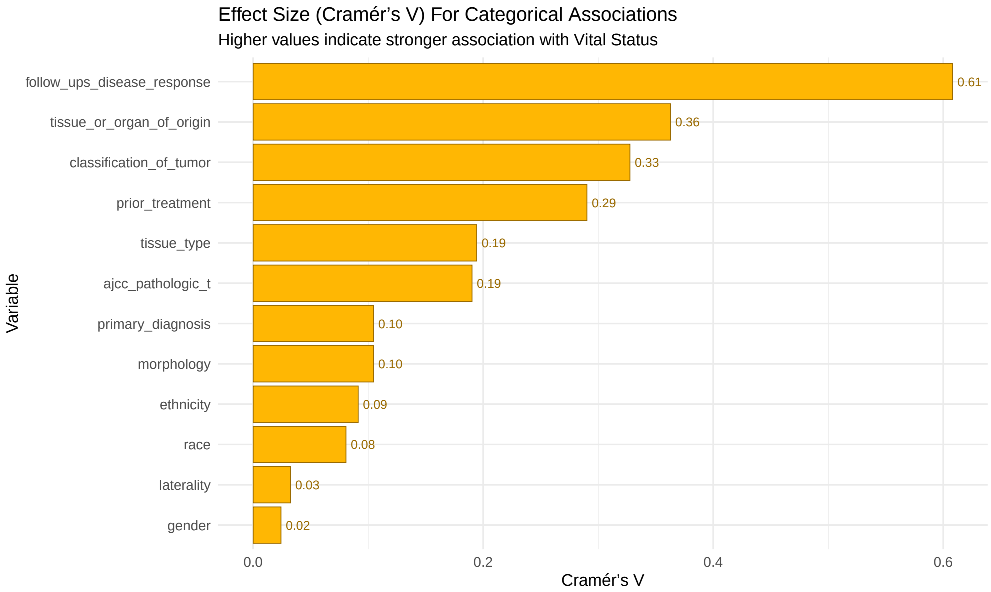
```
  results[[var]] <- data.frame(
    Test=test_name,
    Statistic=stat_val,
    P_value=pval,
    Cramers_V=round(cramers_v, 4),
    Note=ifelse(pval < 0.05, "Significant", "Not significant"),
    stringsAsFactors=FALSE
  )
}

results_df <- do.call(rbind, results)
results_df$Variable <- rownames(results_df)
results_df <- results_df[, c("Variable","Test","Statistic","P_value","Cramers_V","Note")]
print(results_df)
```

```
##                                                 Variable
## tissue_type                                 tissue_type
## laterality                                   laterality
## tissue_or_organ_of_origin     tissue_or_organ_of_origin
## primary_diagnosis                     primary_diagnosis
## prior_treatment                         prior_treatment
## ajcc_pathologic_t                     ajcc_pathologic_t
## morphology                                   morphology
## classification_of_tumor         classification_of_tumor
## follow_ups_disease_response follow_ups_disease_response
## race                                               race
## gender                                           gender
## ethnicity                                     ethnicity
##                                                   Test  Statistic      P_value
## tissue_type                         Fisher Exact (2x2)         NA 1.457349e-09
## laterality                          Fisher Exact (2x2)         NA 2.985037e-01
## tissue_or_organ_of_origin   Chi-square (MC simulation) 161.948801 9.999000e-05
## primary_diagnosis           Chi-square (MC simulation)  13.447819 5.839416e-02
## prior_treatment             Chi-square (MC simulation)  99.889277 9.999000e-05
## ajcc_pathologic_t           Chi-square (MC simulation)  40.951881 9.999000e-05
## morphology                  Chi-square (MC simulation)  13.447819 6.389361e-02
## classification_of_tumor     Chi-square (MC simulation) 130.571256 9.999000e-05
## follow_ups_disease_response Chi-square (MC simulation) 426.860132 9.999000e-05
## race                        Chi-square (MC simulation)   7.376931 6.699330e-02
## gender                              Fisher Exact (2x2)         NA 7.059201e-01
## ethnicity                   Chi-square (MC simulation)   8.644092 1.369863e-02
##                             Cramers_V            Note
## tissue_type                    0.1944     Significant
## laterality                     0.0324 Not significant
## tissue_or_organ_of_origin      0.3629     Significant
## primary_diagnosis              0.1046 Not significant
## prior_treatment                0.2902     Significant
## ajcc_pathologic_t              0.1903     Significant
## morphology                     0.1046 Not significant
## classification_of_tumor        0.3276     Significant
## follow_ups_disease_response    0.6082     Significant
## race                           0.0807 Not significant
## gender                         0.0242 Not significant
## ethnicity                         0.0913     Significant
```

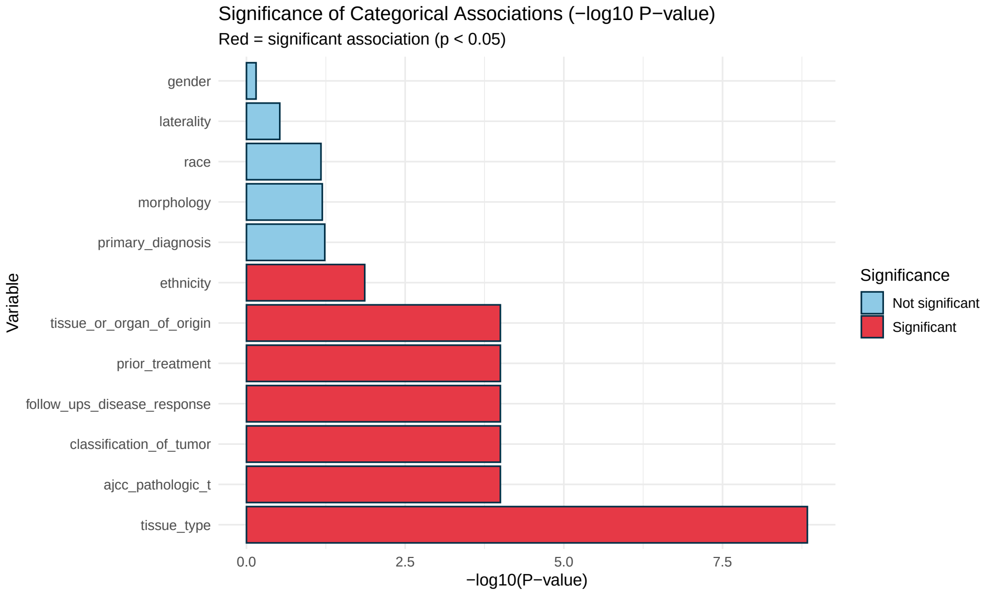
```
# Plot 1: Bar plot with P-value
ggplot(results_df, aes(x = reorder(Variable, P_value), y = P_value)) +
  geom_bar(stat = "identity", fill = "#8ecae6", color = "#023047", linewidth = 0.3) +
  geom_text(aes(label = sprintf("%.3f", P_value)),
            hjust = -0.2, size = 3, color = "#023047") +
  coord_flip() +
  labs(title = "P-values for Categorical Associations",
       subtitle = "Chi-square / Fisher tests across categorical variables",
       x = "Variable",
       y = "P-value",
       caption = "Statistical comparison of categorical variables vs Vital Status") +
  theme_minimal(base_size = 12) +
  theme(plot.title = element_text(face = "bold", hjust = 0.5, color = "#023047"),
        plot.subtitle = element_text(hjust = 0.5, color = "#555555"),
        plot.caption = element_text(face = "italic", color = "#666666"),
        axis.title = element_text(face = "bold", color = "#023047"))
```



```
# Plot 1: Cramér's V Strength Plot
ggplot(results_df, aes(x = reorder(Variable, Cramers_V), y = Cramers_V)) +
  geom_bar(stat = "identity", fill = "#ffb703", color = "#9a6a00", linewidth = 0.3) +
  geom_text(aes(label = sprintf("%.2f", Cramers_V)),
            hjust = -0.2, size = 3, color = "#9a6a00") +
  coord_flip() +
  labs(title = "Effect Size (Cramér's V) For Categorical Associations",
       subtitle = "Higher values indicate stronger association with Vital Status",
       x = "Variable",
       y = "Cramér's V") +
  theme_minimal(base_size = 12)
```



```
# Plot 3: Significance Highlight Plot
ggplot(results_df, aes(x = reorder(Variable, P_value),
                       y = -log10(P_value),
                       fill = Note)) +
  geom_bar(stat = "identity", color = "#023047") +
  coord_flip() +
  scale_fill_manual(values = c("Significant" = "#e63946",
                               "Not significant" = "#8ecae6")) +
  labs(title = "Significance of Categorical Associations (-log10 P-value)",
       subtitle = "Red = significant association (p < 0.05)",
       x = "Variable",
       y = "-log10(P-value)",
       fill = "Significance") +
  theme_minimal(base_size = 12)
```



From this result, most of categorical variables show no meaningful link to survival. Gender has no effect, ethnicity is statistically significant but with very weak effect ($V \approx 0.09$), and tumor subtype shows only a borderline trend. Overall, categorical predictors contribute little to explaining survival differences.

Genex data studies

Differential Expression Analysis

```
# Matrix vital status
design <- model.matrix(~vital_status, data = clinical_df)

# Fit linear model using limma (empirical Bayes)
fit <- lmFit(t(GeneX_df), design)
fit <- eBayes(fit)

# Extract top differentially expressed genes
top_genes <- topTable(fit
                      , coef          = 2
                      , number        = 5000
                      , adjust.method = "BH")

cat("Top 20 Differentially Expressed Genes (Dead vs Alive):\n")
```

```
## Top 20 Differentially Expressed Genes (Dead vs Alive):
```

```
print(top_genes[1:20, c("logFC", "AveExpr", "P.Value", "adj.P.Val")])
```

```
##                   logFC   AveExpr      P.Value    adj.P.Val
## LINC01235   0.9515500  7.047438 2.078880e-11 1.039440e-07
## APOB        1.2822838  3.413208 1.035776e-10 2.589440e-07
## LYVE1       1.0214367  7.224554 2.346103e-10 3.910171e-07
## LINC01497   0.7523277  1.156414 3.437925e-09 4.297407e-06
## AC104211.1  0.7273947  3.366769 1.643701e-08 1.535403e-05
## KLB         0.8499149  6.395080 2.069437e-08 1.535403e-05
## PSD2        0.7153297  4.402892 2.149564e-08 1.535403e-05
## LINC02511   0.8574617  2.253701 5.811988e-08 3.280902e-05
## SNORD104   -0.6927049  4.644127 5.905623e-08 3.280902e-05
## CST1       -1.4893705  6.693858 8.124615e-08 3.452865e-05
## AC007423.1  0.7409439  1.210320 8.195581e-08 3.452865e-05
## GPX3        0.7475773 11.556001 8.286876e-08 3.452865e-05
## LVRN        0.8868127  4.691047 1.429004e-07 4.769855e-05
## PROKR1      0.8122548  2.187548 1.442226e-07 4.769855e-05
## RHBDL1     -0.6842668  7.068012 1.518817e-07 4.769855e-05
## ADH4        0.8146112  2.169812 1.619101e-07 4.769855e-05
## ATF3        0.6698043 10.693098 1.621751e-07 4.769855e-05
## SLC2A4      0.7785291  6.008617 2.085221e-07 5.723414e-05
## FHL1        0.7816811 10.841301 2.174897e-07 5.723414e-05
## VEGFD       1.0116073  5.215086 2.874839e-07 6.940534e-05
```

```r
cat("\n=== DE SUMMARY ===\n")
```

```
##
## === DE SUMMARY ===
```

```r
cat("Significant genes (FDR < 0.05):", sum(top_genes$adj.P.Val < 0.05), "\n")
```

```
## Significant genes (FDR < 0.05): 1159
```

```r
cat("Genes |logFC| > 1:", sum(abs(top_genes$logFC) > 1), "\n")
```

```
## Genes |logFC| > 1: 13
```

```r
cat("Both significant AND |logFC| > 1:"
    , sum(top_genes$adj.P.Val < 0.05 & abs(top_genes$logFC) > 1), "\n\n")
```

```
## Both significant AND |logFC| > 1: 13
```

```r
cat("Expression direction:\n")
```

```
## Expression direction:
```

```r
cat("  Upregulated in Dead:", sum(top_genes$logFC > 0), "\n")
```

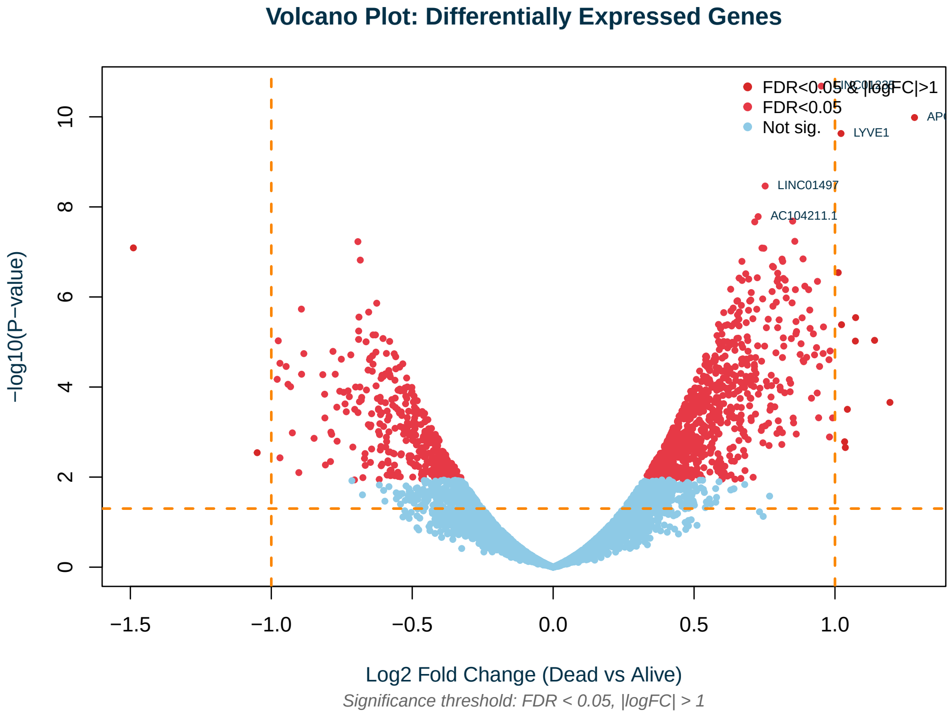
```
##   Upregulated in Dead: 2692
```

```r
cat("  Downregulated in Dead:", sum(top_genes$logFC < 0), "\n")
```

```
##   Downregulated in Dead: 2308
```

Volcano Plot

```r
plot(top_genes$logFC
     , -log10(top_genes$P.Value)
     , pch      = 19
     , cex      = 0.6
     , col      = ifelse(top_genes$adj.P.Val < 0.05
                         , ifelse(abs(top_genes$logFC) > 1, "#d62828", "#e63946")
                         , "#8ecae6")
     , xlab     = "Log2 Fold Change (Dead vs Alive)"
     , ylab     = "-log10(P-value)"
     , main     = "Volcano Plot: Differentially Expressed Genes"
     , sub      = "Significance threshold: FDR < 0.05, |logFC| > 1"
     , col.main = "#023047"
     , col.lab  = "#023047"
     , col.sub  = "#666666"
     , cex.sub  = 0.8
     , font.sub = 3)

# Significance thresholds 0,05
abline(h   = -log10(0.05)
       , col = "#fb8500"
       , lty = 2
       , lwd = 2)

abline(v   = c(-1, 1)
       , col = "#fb8500"
       , lty = 2
       , lwd = 2)

# Gene labels
top_hits <- rownames(top_genes)[1:5]
for(gene in top_hits) {
  text(top_genes[gene, "logFC"]
       , -log10(top_genes[gene, "P.Value"])
       , labels = gene
       , pos    = 4
       , cex    = 0.6
       , col    = "#023047")
}

legend("topright"
       , legend = c("FDR<0.05 & |logFC|>1", "FDR<0.05", "Not sig.")
       , col    = c("#d62828", "#e63946", "#8ecae6")
       , pch    = 19
       , bty    = "n"
       , cex    = 0.8)
```



From the result **top_gene**, most significant genes show **positive logFC**, meaning they are **up-regulated** in patients who died, while like few **CST1, MMP11, SNORD104** are down-regulated. Several genes display both large effect sizes (|logFC| > 1) and very strong statistical significance **(FDR « 0.05)**—notably **APOB, LYVE1, LINC01497, AC104211.1** making them the clearest potential biomarkers.

The Volcano plot confirms a pronouced asymmetry, with a dense cluster of up-regulated genes in the Dead group, indicating the activated transcriptional programs link to poor diagnosis, where down-regulated genes are fewer and more dispersed.

Overall, the results point to a robust molecular signature differentiating Alive vs Dead patients, with a handful of genes emerging as particularly strong candidates for biological interpretation and predictive modeling.

Gene Expression Characteristics

```r
top50_genes <- top_genes[1:50, ]

cat("=== TOP 50 GENE CHARACTERISTICS ===\n\n")
```

```
## === TOP 50 GENE CHARACTERISTICS ===
```

```r
# Expression levels
cat("Average Expression Levels:\n")
```

```
## Average Expression Levels:
```

```r
cat("  Min:", round(min(top50_genes$AveExpr), 2), "\n")
```

```
##   Min: 1.16
```

```r
cat("  Median:", round(median(top50_genes$AveExpr), 2), "\n")
```

```
##   Median: 4.49
```

```r
cat("  Max:", round(max(top50_genes$AveExpr), 2), "\n\n")
```

```
##   Max: 11.56
```

```r
# Fold changes
cat("Fold Change Distribution:\n")
```

```
## Fold Change Distribution:
```

```r
cat("  Upregulated in Dead (logFC > 0):", sum(top50_genes$logFC > 0), "\n")
```

```
##   Upregulated in Dead (logFC > 0): 46
```

```r
cat("  Downregulated in Dead (logFC < 0):", sum(top50_genes$logFC < 0), "\n\n")
```

```
##   Downregulated in Dead (logFC < 0): 4
```

```r
# Statistical significance
cat("P-value ranges:\n")
```

```
## P-value ranges:
```

```r
cat("  Min P-value:", format(min(top50_genes$P.Value), scientific = TRUE), "\n")
```

```
##   Min P-value: 2.07888e-11
```

```r
cat("  Max P-value:", format(max(top50_genes$P.Value), scientific = TRUE), "\n")
```

```
##   Max P-value: 1.605859e-06
```

```r
cat("  Max FDR:", format(max(top50_genes$adj.P.Val), scientific = TRUE), "\n")
```

```
##   Max FDR: 1.605859e-04
```

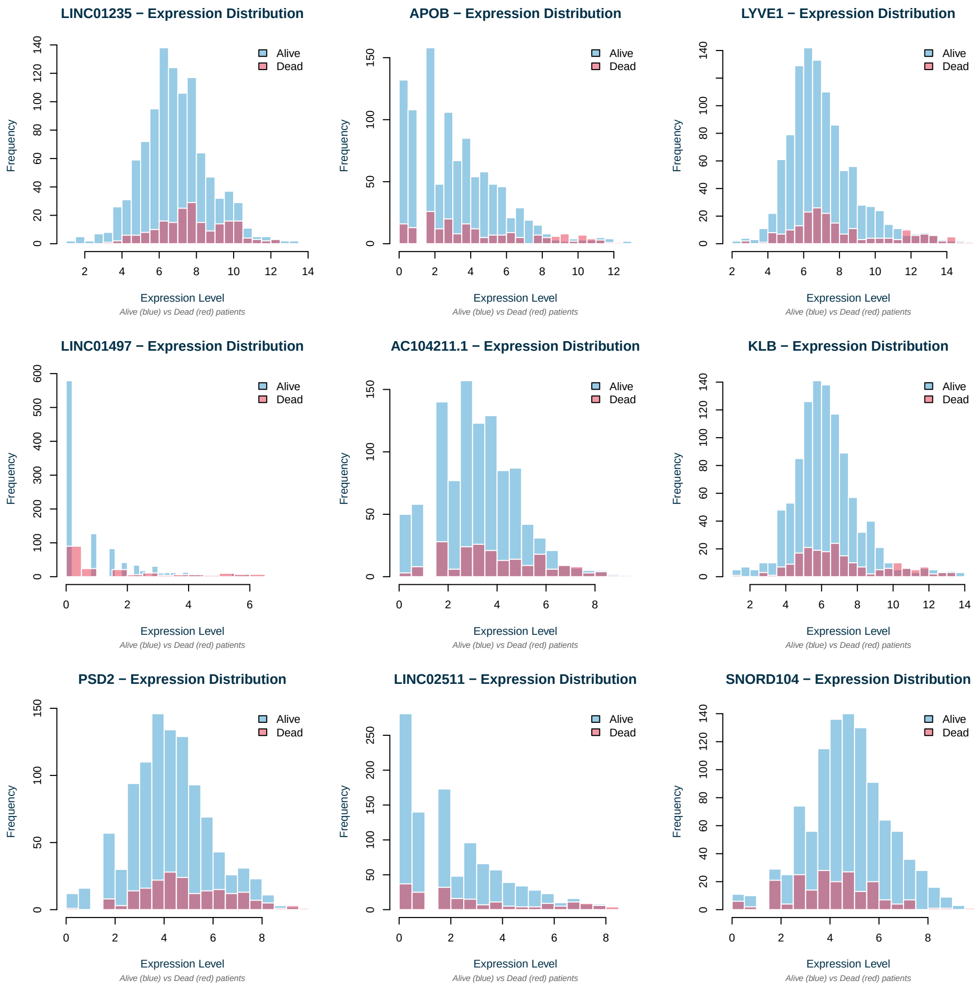
Gene Expression Distributions

```
# Create gene subset for top 20 genes
top20_genes <- rownames(top_genes)[1:20]
gene_subset <- as.data.frame(GeneX_df[, top20_genes])
colnames(gene_subset) <- top20_genes

par(mfrow = c(3, 3), bg = "white")

for(i in 1:9) {
  gene      <- top20_genes[i]
  gene_expr <- gene_subset[, i]

  # Histogram with separate colors by vital status
  hist(gene_expr[clinical_df$vital_status == "Alive"]
       , breaks   = 30
       , col      = rgb(0.2, 0.6, 0.8, 0.5)
       , main     = paste(gene, "- Expression Distribution")
       , sub      = "Alive (blue) vs Dead (red) patients"
       , xlab     = "Expression Level"
       , ylab     = "Frequency"
       , border   = "white"
       , col.main = "#023047"
       , col.lab  = "#023047"
       , col.sub  = "#666666"
       , cex.sub  = 0.7
       , font.sub = 3)

  hist(gene_expr[clinical_df$vital_status == "Dead"]
       , breaks = 30
       , col    = rgb(0.9, 0.2, 0.3, 0.5)
       , add    = TRUE
       , border = "white")

  legend("topright"
         , legend = c("Alive", "Dead")
         , fill   = c(rgb(0.2, 0.6, 0.8, 0.5), rgb(0.9, 0.2, 0.3, 0.5))
         , bty    = "n")

  # Test for bimodality (Hartigan's dip test)
  dip_result <- dip.test(gene_expr)

  if(dip_result$p.value < 0.05) {
    cat(sprintf("%s: BIMODAL (p=%.4f) need subgroups!\n"
                , gene
                , dip_result$p.value))
  }
}
```

```
## APOB: BIMODAL (p=0.0000) need subgroups!
```

```
## LINC01497: BIMODAL (p=0.0000) need subgroups!
```

```
## AC104211.1: BIMODAL (p=0.0000) need subgroups!
## PSD2: BIMODAL (p=0.0009) need subgroups!

## LINC02511: BIMODAL (p=0.0000) need subgroups!
```



A subset of the strongest DE genes shows bimodal expression patterns, indicating heterogeneity and possible molecular subgroups, while several genes exhibit clear upregulation in non-survivors, reinforcing their biological relevance.

Bimodal Gene Analysis

```r
# Create gene subset for top 500
top500_genes <- rownames(top_genes)[1:500]
gene500_subset <- as.data.frame(GeneX_df[, top500_genes])
colnames(gene500_subset) <- top500_genes

# Run dip test for each of the top 500 genes
dip_results <- sapply(gene500_subset, function(x) {
    dip.test(x)$p.value
})
```

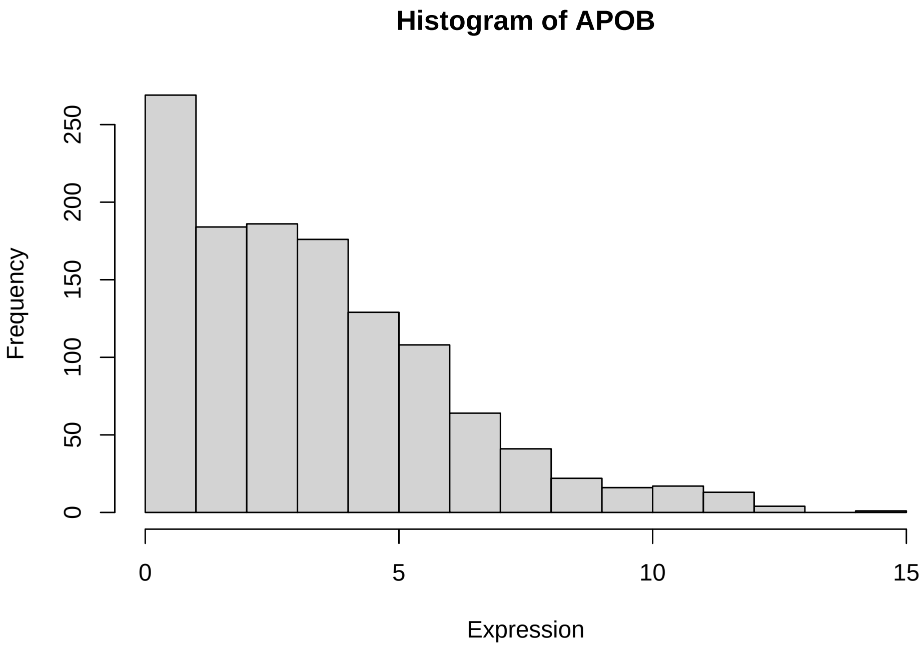
```r
# Convert to data frame
bimodality_df <- data.frame(
    Gene = names(dip_results),
    Dip_Pvalue = dip_results
)

bimodality_df$Is_Bimodal <- bimodality_df$Dip_Pvalue < 0.05

# Sort by lowest dip-test pvalue (most strongly bimodal first)
bimodality_df <- bimodality_df[order(bimodality_df$Dip_Pvalue), ]

# Check only Bimodal
bimodal_genes_only <- bimodality_df[bimodality_df$Is_Bimodal == TRUE, ]

cat("\n=== BIMODAL GENES (Dip p < 0.05) ===\n")
```

```
##
## === BIMODAL GENES (Dip p < 0.05) ===
```

```r
print(bimodal_genes_only)
```

```
##                      Gene   Dip_Pvalue Is_Bimodal
## APOB                 APOB 0.000000e+00       TRUE
## LINC01497       LINC01497 0.000000e+00       TRUE
## AC104211.1     AC104211.1 0.000000e+00       TRUE
## LINC02511       LINC02511 0.000000e+00       TRUE
## CST1                 CST1 0.000000e+00       TRUE
## AC007423.1     AC007423.1 0.000000e+00       TRUE
## PROKR1             PROKR1 0.000000e+00       TRUE
## ADH4                 ADH4 0.000000e+00       TRUE
## ALDH1L1-AS2 ALDH1L1-AS2 0.000000e+00       TRUE
## LINC01537       LINC01537 0.000000e+00       TRUE
## LINC01186       LINC01186 0.000000e+00       TRUE
## GLP2R               GLP2R 0.000000e+00       TRUE
## NGF-AS1           NGF-AS1 0.000000e+00       TRUE
## HSD17B13         HSD17B13 0.000000e+00       TRUE
## LUARIS             LUARIS 0.000000e+00       TRUE
## DSC1                 DSC1 0.000000e+00       TRUE
## LINC01612       LINC01612 0.000000e+00       TRUE
## FP325317.1     FP325317.1 0.000000e+00       TRUE
## ABCB5               ABCB5 0.000000e+00       TRUE
## ADRA1A             ADRA1A 0.000000e+00       TRUE
## LHCGR               LHCGR 0.000000e+00       TRUE
## PGM5-AS1         PGM5-AS1 0.000000e+00       TRUE
## AL356218.2     AL356218.2 0.000000e+00       TRUE
## C1QTNF9           C1QTNF9 0.000000e+00       TRUE
## ADH1A               ADH1A 0.000000e+00       TRUE
## AC036108.2     AC036108.2 0.000000e+00       TRUE
## AC079804.3     AC079804.3 0.000000e+00       TRUE
## GLRA4               GLRA4 0.000000e+00       TRUE
## LINC02237       LINC02237 0.000000e+00       TRUE
## CA4                   CA4 0.000000e+00       TRUE
## AL121950.1   AL121950.1 0.000000e+00        TRUE
## EPB42             EPB42 0.000000e+00        TRUE
## PROX1-AS1     PROX1-AS1 0.000000e+00        TRUE
## GLYAT             GLYAT 0.000000e+00        TRUE
## SPX                 SPX 0.000000e+00        TRUE
## AC104407.1   AC104407.1 0.000000e+00        TRUE
## TM4SF4           TM4SF4 0.000000e+00        TRUE
## LALBA             LALBA 0.000000e+00        TRUE
## LINC01697     LINC01697 0.000000e+00        TRUE
## MAFA-AS1       MAFA-AS1 0.000000e+00        TRUE
## PCK1               PCK1 0.000000e+00        TRUE
## NPY2R             NPY2R 0.000000e+00        TRUE
## AC084212.1   AC084212.1 0.000000e+00        TRUE
## CDH20             CDH20 0.000000e+00        TRUE
## AC105118.1   AC105118.1 0.000000e+00        TRUE
## ANGPTL8         ANGPTL8 0.000000e+00        TRUE
## LINC02660     LINC02660 0.000000e+00        TRUE
## AC073850.1   AC073850.1 0.000000e+00        TRUE
## AL450332.1   AL450332.1 0.000000e+00        TRUE
## NEUROG2         NEUROG2 0.000000e+00        TRUE
## AL845331.1   AL845331.1 0.000000e+00        TRUE
## SERTM1           SERTM1 0.000000e+00        TRUE
## LINCADL         LINCADL 0.000000e+00        TRUE
## IBSP               IBSP 0.000000e+00        TRUE
## ANO3               ANO3 0.000000e+00        TRUE
## ADH1C             ADH1C 0.000000e+00        TRUE
## PLCZ1             PLCZ1 0.000000e+00        TRUE
## AC016682.1   AC016682.1 0.000000e+00        TRUE
## LGALS17A       LGALS17A 0.000000e+00        TRUE
## TMEM252         TMEM252 0.000000e+00        TRUE
## TRHDE-AS1     TRHDE-AS1 0.000000e+00        TRUE
## ANGPTL7         ANGPTL7 0.000000e+00        TRUE
## AP001360.1   AP001360.1 0.000000e+00        TRUE
## CDH12             CDH12 0.000000e+00        TRUE
## LRRC3B           LRRC3B 0.000000e+00        TRUE
## ACSM4             ACSM4 0.000000e+00        TRUE
## MYOC               MYOC 0.000000e+00        TRUE
## H2BC17           H2BC17 0.000000e+00        TRUE
## AC003986.2   AC003986.2 0.000000e+00        TRUE
## SGCZ               SGCZ 0.000000e+00        TRUE
## AL591686.1   AL591686.1 0.000000e+00        TRUE
## NRAD1             NRAD1 0.000000e+00        TRUE
## ACE2               ACE2 0.000000e+00        TRUE
## AL353693.1   AL353693.1 0.000000e+00        TRUE
## GRIN2B           GRIN2B 0.000000e+00        TRUE
## MIR145           MIR145 0.000000e+00        TRUE
## AL138716.1   AL138716.1 0.000000e+00        TRUE
## LINC01230     LINC01230 0.000000e+00        TRUE
## CSF3               CSF3 0.000000e+00        TRUE
## TNNI3             TNNI3 0.000000e+00        TRUE
## AC112721.2   AC112721.2 0.000000e+00        TRUE
## B3GAT1-DT     B3GAT1-DT 0.000000e+00        TRUE
## LINC01561     LINC01561 0.000000e+00        TRUE
## CCL14             CCL14 0.000000e+00        TRUE
## NOS1               NOS1 0.000000e+00       TRUE
## MLIP               MLIP 0.000000e+00       TRUE
## AC093496.1   AC093496.1 0.000000e+00       TRUE
## KCNJ16           KCNJ16 0.000000e+00       TRUE
## AC092118.1   AC092118.1 0.000000e+00       TRUE
## AC108734.4   AC108734.4 0.000000e+00       TRUE
## AC002546.1   AC002546.1 0.000000e+00       TRUE
## AQP7P1           AQP7P1 0.000000e+00       TRUE
## CSN1S1           CSN1S1 0.000000e+00       TRUE
## AADAC             AADAC 0.000000e+00       TRUE
## AC016924.1   AC016924.1 0.000000e+00       TRUE
## LINC02587     LINC02587 0.000000e+00       TRUE
## CST4               CST4 0.000000e+00       TRUE
## AC093817.2   AC093817.2 0.000000e+00       TRUE
## TRDN               TRDN 0.000000e+00       TRUE
## SHISA3           SHISA3 0.000000e+00       TRUE
## KCNH1-IT1     KCNH1-IT1 0.000000e+00       TRUE
## HEPACAM         HEPACAM 0.000000e+00       TRUE
## DCT                 DCT 0.000000e+00       TRUE
## TRHDE             TRHDE 0.000000e+00       TRUE
## TGFBR3L         TGFBR3L 0.000000e+00       TRUE
## AL645924.1   AL645924.1 0.000000e+00       TRUE
## SLC6A3           SLC6A3 0.000000e+00       TRUE
## CCDC144A       CCDC144A 0.000000e+00       TRUE
## RBMS3-AS3     RBMS3-AS3 0.000000e+00       TRUE
## LINC01281     LINC01281 0.000000e+00       TRUE
## DPP6               DPP6 0.000000e+00       TRUE
## HHATL             HHATL 0.000000e+00       TRUE
## Z98745.2       Z98745.2 0.000000e+00       TRUE
## HSD11B1-AS1 HSD11B1-AS1 0.000000e+00       TRUE
## C6                   C6 0.000000e+00       TRUE
## RXRG               RXRG 0.000000e+00       TRUE
## CNTN6             CNTN6 0.000000e+00       TRUE
## GRIA4             GRIA4 0.000000e+00       TRUE
## AC008459.1   AC008459.1 0.000000e+00       TRUE
## ANTXRL           ANTXRL 0.000000e+00       TRUE
## PTCHD3           PTCHD3 0.000000e+00       TRUE
## SLC7A14-AS1 SLC7A14-AS1 0.000000e+00       TRUE
## FOXD3-AS1     FOXD3-AS1 0.000000e+00       TRUE
## AC110774.1   AC110774.1 0.000000e+00       TRUE
## CPA1               CPA1 0.000000e+00       TRUE
## PURPL             PURPL 0.000000e+00       TRUE
## BAK1P2           BAK1P2 0.000000e+00       TRUE
## SLC7A10         SLC7A10 0.000000e+00       TRUE
## AP002800.1   AP002800.1 0.000000e+00       TRUE
## SFTPB             SFTPB 0.000000e+00       TRUE
## NEUROG2-AS1 NEUROG2-AS1 0.000000e+00       TRUE
## AC092851.1   AC092851.1 0.000000e+00       TRUE
## GPS2P1           GPS2P1 0.000000e+00       TRUE
## PROK1             PROK1 0.000000e+00       TRUE
## AC121757.1   AC121757.1 0.000000e+00       TRUE
## OR2B6             OR2B6 0.000000e+00       TRUE
## ADCY8             ADCY8 0.000000e+00       TRUE
## AL356489.2   AL356489.2 0.000000e+00       TRUE
## SH3GL3           SH3GL3 0.000000e+00       TRUE
## TUBA3E           TUBA3E 0.000000e+00       TRUE
## CAPZA3           CAPZA3 0.000000e+00       TRUE
## CNTNAP3P2     CNTNAP3P2 0.000000e+00       TRUE
## CCNYL2           CCNYL2 0.000000e+00       TRUE
## SLITRK2         SLITRK2 0.000000e+00       TRUE
## MPPED1           MPPED1 0.000000e+00       TRUE
## C14orf180     C14orf180 0.000000e+00       TRUE
## LMX1A             LMX1A 0.000000e+00       TRUE
## CMA1               CMA1 0.000000e+00       TRUE
## RPS4Y1           RPS4Y1 0.000000e+00       TRUE
## LEP                 LEP 0.000000e+00       TRUE
## CYP1A1           CYP1A1 0.000000e+00       TRUE
## PTPRQ             PTPRQ 0.000000e+00       TRUE
## AP005131.3   AP005131.3 0.000000e+00       TRUE
## MAPT-IT1       MAPT-IT1 0.000000e+00       TRUE
## LINC01625     LINC01625 0.000000e+00       TRUE
## AC098850.3   AC098850.3 0.000000e+00       TRUE
## AC119424.1   AC119424.1 0.000000e+00       TRUE
## LINC00844     LINC00844 0.000000e+00       TRUE
## GRAMD4P8       GRAMD4P8 0.000000e+00       TRUE
## AL513318.1   AL513318.1 0.000000e+00       TRUE
## SNTN               SNTN 0.000000e+00       TRUE
## LINC01485     LINC01485 0.000000e+00       TRUE
## AC022196.1   AC022196.1 0.000000e+00       TRUE
## AC068733.3   AC068733.3 0.000000e+00       TRUE
## AC073869.6   AC073869.6 0.000000e+00       TRUE
## DRD1               DRD1 0.000000e+00       TRUE
## TNMD               TNMD 0.000000e+00       TRUE
## AC073316.1   AC073316.1 0.000000e+00       TRUE
## AP000350.6   AP000350.6 0.000000e+00       TRUE
## LINC00466     LINC00466 0.000000e+00       TRUE
## FAM180B         FAM180B 0.000000e+00       TRUE
## HBA1               HBA1 0.000000e+00       TRUE
## AL033384.1   AL033384.1 0.000000e+00       TRUE
## LINC01344     LINC01344 0.000000e+00       TRUE
## LINC02515     LINC02515 0.000000e+00       TRUE
## DIRAS2           DIRAS2 0.000000e+00       TRUE
## PENK               PENK 0.000000e+00       TRUE
## OXGR1             OXGR1 0.000000e+00       TRUE
## USP32P1         USP32P1 0.000000e+00       TRUE
## TMEM132C       TMEM132C 0.000000e+00       TRUE
## PLD5               PLD5 0.000000e+00       TRUE
## MGAT4C           MGAT4C 0.000000e+00       TRUE
## AL161945.1   AL161945.1 0.000000e+00       TRUE
## CLDN25           CLDN25 0.000000e+00       TRUE
## ASB4               ASB4 0.000000e+00       TRUE
## BMS1P10         BMS1P10 7.134626e-07       TRUE
## CST2               CST2 3.129484e-06       TRUE
## CHRNA6           CHRNA6 4.741375e-06       TRUE
## RERGL             RERGL 4.741375e-06       TRUE
## PRR26             PRR26 7.722067e-06       TRUE
## LINC00968     LINC00968 7.965158e-06       TRUE
## HPSE2             HPSE2 7.965158e-06       TRUE
## PLCXD3           PLCXD3 9.577049e-06       TRUE
## ANGPT4           ANGPT4 9.577049e-06       TRUE
## CCDC178         CCDC178 9.577049e-06       TRUE
## LINC01239     LINC01239 9.577049e-06       TRUE
## RIMBP2           RIMBP2 2.036191e-05       TRUE
## ADGRB3           ADGRB3 3.371400e-05       TRUE
## SCT                 SCT 9.664365e-05       TRUE
## HIF3A             HIF3A 1.505938e-04       TRUE
## KY                   KY 1.505938e-04       TRUE
## NLGN1             NLGN1 2.184164e-04       TRUE
## AL583785.1   AL583785.1 2.184164e-04       TRUE
## GRIK1             GRIK1 3.423733e-04       TRUE
## MAP1LC3C       MAP1LC3C 4.663301e-04       TRUE
## MASP1             MASP1 4.663301e-04       TRUE
## AC090004.2   AC090004.2 4.663301e-04       TRUE
## AC055854.1   AC055854.1 4.663301e-04       TRUE
## PSD2               PSD2 9.472775e-04       TRUE
## SSTR1             SSTR1 9.472775e-04       TRUE
## H2AC13           H2AC13 9.472775e-04       TRUE
## AL032819.2   AL032819.2 1.382277e-03       TRUE
## NRXN1             NRXN1 1.862362e-03       TRUE
## CD300LG         CD300LG 2.751299e-03       TRUE
## HOXA2             HOXA2 2.751299e-03       TRUE
## PLAC1             PLAC1 2.751299e-03       TRUE
## DRD2               DRD2 2.751299e-03       TRUE
## ZBED2             ZBED2 3.804563e-03       TRUE
## CALHM6           CALHM6 6.492312e-03       TRUE
## LINC00922     LINC00922 6.880659e-03       TRUE
## CIDEC             CIDEC 6.880659e-03       TRUE
## CRHBP             CRHBP 9.054789e-03       TRUE
## KRT19P1         KRT19P1 9.054789e-03       TRUE
## ADH1B             ADH1B 9.054789e-03       TRUE
## PRRT4             PRRT4 1.231416e-02       TRUE
## ADAMTS9-AS2 ADAMTS9-AS2 1.231416e-02       TRUE
## FOXJ1             FOXJ1 1.575060e-02       TRUE
## LVRN               LVRN 1.640823e-02       TRUE
## PAPPA2           PAPPA2 1.872212e-02       TRUE
## VEGFD             VEGFD 2.104729e-02       TRUE
## SCN3A             SCN3A 2.104729e-02       TRUE
## SULT1B1         SULT1B1 2.104729e-02       TRUE
## NMUR1             NMUR1 2.958350e-02       TRUE
## GDF10             GDF10 2.958350e-02       TRUE
## DQX1               DQX1 3.811971e-02       TRUE
## HNRNPA1P21   HNRNPA1P21 3.811971e-02       TRUE
## RHOXF1-AS1   RHOXF1-AS1 4.665592e-02       TRUE
## LRRC2             LRRC2 4.665592e-02       TRUE
## RUFY4             RUFY4 4.665592e-02       TRUE
```

```r
# Visualize top Bimodal Genex
top_bimodal_gene <- bimodality_df$Gene[1]
expr_values <- gene500_subset[[top_bimodal_gene]]

hist(expr_values, breaks = 20,
     main = paste("Histogram of", top_bimodal_gene),
     xlab = "Expression")
```



```
cat("=== BIMODALITY CHECK SUMMARY (Top 500 Genes) ===\n")
```

```
## === BIMODALITY CHECK SUMMARY (Top 500 Genes) ===
```

```
cat("Total genes tested:", nrow(bimodality_df), "\n")
```

```
## Total genes tested: 500
```

```
cat("Bimodal genes (Dip p < 0.05):", sum(bimodality_df$Is_Bimodal), "\n\n")
```

```
## Bimodal genes (Dip p < 0.05): 239
```

```
cat("Top 10 most bimodal genes:\n")
```

```
## Top 10 most bimodal genes:
```

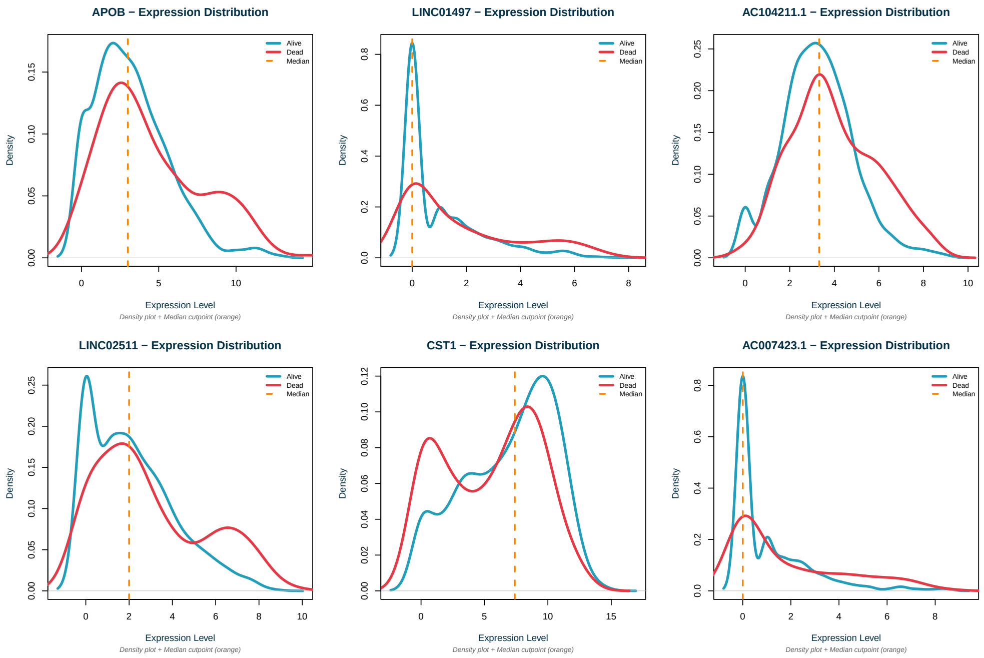
```
print(head(bimodality_df, 10))
```

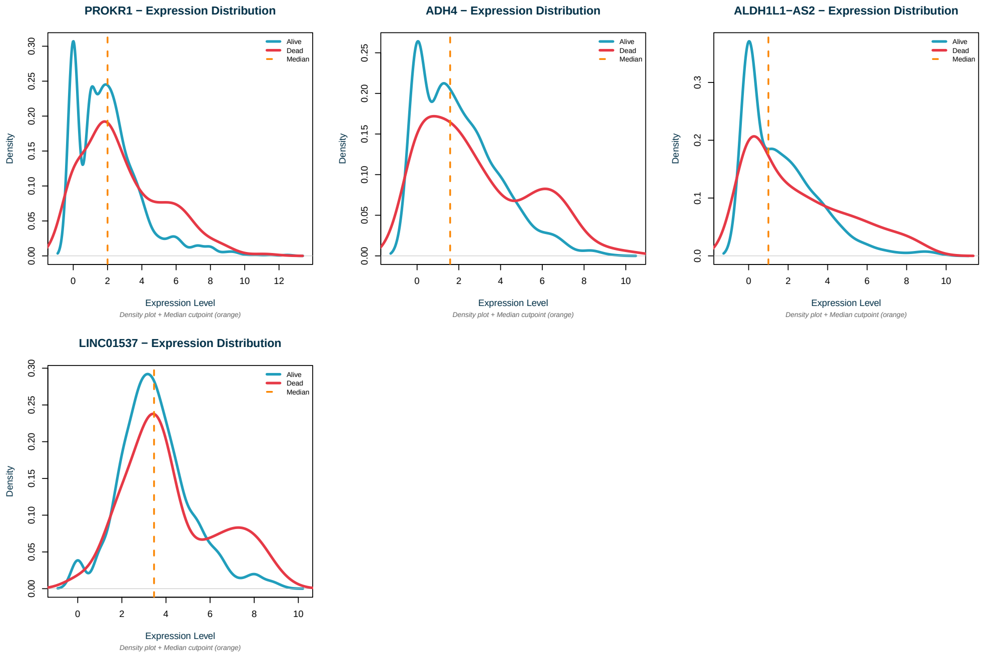
```
##                    Gene Dip_Pvalue Is_Bimodal
## APOB               APOB          0       TRUE
## LINC01497     LINC01497          0       TRUE
## AC104211.1   AC104211.1          0       TRUE
## LINC02511     LINC02511          0       TRUE
## CST1               CST1          0       TRUE
## AC007423.1   AC007423.1          0       TRUE
## PROKR1            PROKR1          0       TRUE
## ADH4                ADH4          0       TRUE
## ALDH1L1-AS2 ALDH1L1-AS2          0       TRUE
## LINC01537      LINC01537          0       TRUE
```

Top 10 Bimodal distributions

```
par(mfrow = c(2, 3), bg = "white")

# Loop through top 10 most bimodal genes
for (gene in head(bimodality_df$Gene, 10)) {

  gene_expr <- gene500_subset[, gene]

  alive_expr <- gene_expr[clinical_df$vital_status == "Alive"]
  dead_expr  <- gene_expr[clinical_df$vital_status == "Dead"]

  # Make sure densities exist (avoids zero-length errors)
  if (length(alive_expr) > 1 & length(dead_expr) > 1) {

    plot(density(alive_expr, na.rm = TRUE),
         col      = "#219ebc",
         lwd      = 3,
         main     = paste(gene, "- Expression Distribution"),
         sub      = "Density plot + Median cutpoint (orange)",
         xlab     = "Expression Level",
         ylab     = "Density",
         col.main = "#023047",
         col.lab  = "#023047",
         col.sub  = "#666666",
         cex.sub  = 0.7,
         font.sub = 3)

    lines(density(dead_expr, na.rm = TRUE),
          col = "#e63946",
          lwd = 3)

    # Add median vertical line
    abline(v = median(gene_expr, na.rm = TRUE),
           col = "#fb8500",
           lty = 2,
           lwd = 2)

    legend("topright",
           legend = c("Alive", "Dead", "Median"),
           col    = c("#219ebc", "#e63946", "#fb8500"),
           lwd    = c(3, 3, 2),
           lty    = c(1, 1, 2),
           bty    = "n",
           cex    = 0.8)
  }
```

```r
}
```





```r
# Identify bimodal genes from paper
bimodal_genes <- c("APOB", "LINC01497", "AC104211.1", "PSD2", "LINC02511")

cat("Bimodal genes identified:", length(bimodal_genes), "\n")
```

```
## Bimodal genes identified: 5
```

```
cat(paste(bimodal_genes, collapse = ", "), "\n\n")
```

```
## APOB, LINC01497, AC104211.1, PSD2, LINC02511
```

```
# For each bimodal gene, create binary groups (high/low expressors)
cat("Creating binary expression groups for bimodal genes:\n\n")
```

```
## Creating binary expression groups for bimodal genes:
```

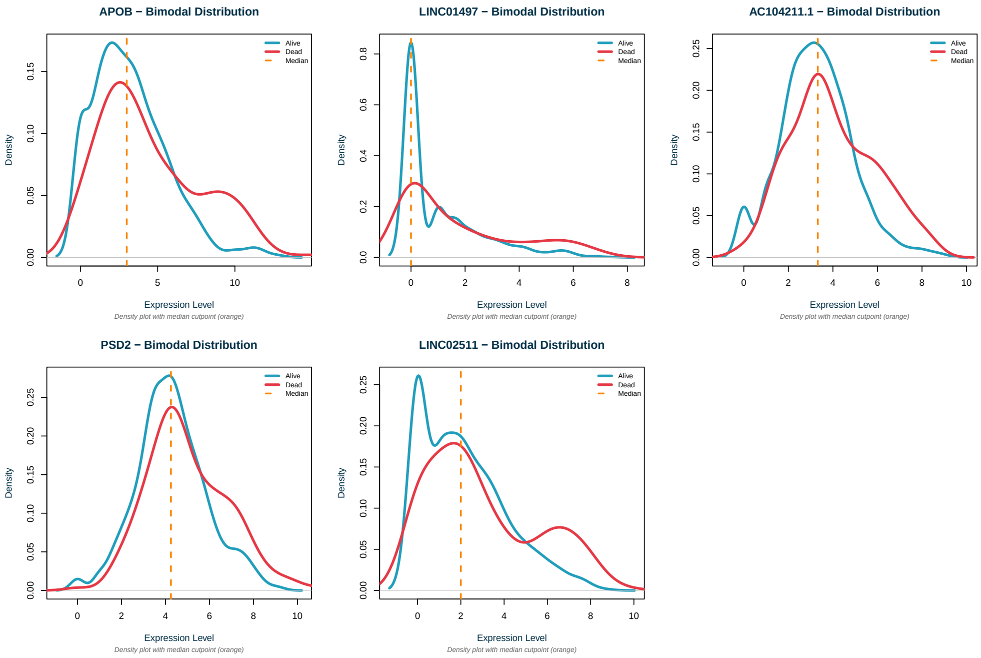
```
for(gene in bimodal_genes) {
  gene_expr   <- gene_subset[, gene]
  gene_median <- median(gene_expr)

  # Create binary groups
  clinical_df[[paste0(gene, "_group")]] <- ifelse(gene_expr > gene_median
                                                   , "High", "Low")

  # Test survival difference
  high_death_rate <- sum(clinical_df[[paste0(gene, "_group")]] == "High" &
                          clinical_df$vital_status == "Dead") /
                     sum(clinical_df[[paste0(gene, "_group")]] == "High")

  low_death_rate  <- sum(clinical_df[[paste0(gene, "_group")]] == "Low" &
                          clinical_df$vital_status == "Dead") /
                     sum(clinical_df[[paste0(gene, "_group")]] == "Low")

  cat(sprintf("%s:\n", gene))
  cat(sprintf("  High expressors: %.1f%% mortality\n", high_death_rate * 100))
  cat(sprintf("  Low expressors:  %.1f%% mortality\n", low_death_rate * 100))
  cat(sprintf("  Fold difference: %.2fx\n\n", high_death_rate / low_death_rate))
}
```

```
## APOB:
##   High expressors: 19.3% mortality
##   Low expressors:  13.6% mortality
##   Fold difference: 1.42x
##
## LINC01497:
##   High expressors: 19.8% mortality
##   Low expressors:  13.5% mortality
##   Fold difference: 1.47x
##
## AC104211.1:
##   High expressors: 20.2% mortality
##   Low expressors:  12.9% mortality
##   Fold difference: 1.56x
##
## PSD2:
##   High expressors: 20.2% mortality
##   Low expressors:  12.6% mortality
##   Fold difference: 1.60x
##
```

```
## LINC02511:
##   High expressors: 19.7% mortality
##   Low expressors:  13.7% mortality
##   Fold difference: 1.44x
```

Bimodal Gene Distributions

```r
par(mfrow = c(2, 3), bg = "white")

for(gene in bimodal_genes) {
  gene_expr <- gene_subset[, gene]

  # Density plot
  alive_expr <- gene_expr[clinical_df$vital_status == "Alive"]
  dead_expr  <- gene_expr[clinical_df$vital_status == "Dead"]

  plot(density(alive_expr, na.rm = TRUE)
       , col      = "#219ebc"
       , lwd      = 3
       , main     = paste(gene, "- Bimodal Distribution")
       , sub      = "Density plot with median cutpoint (orange)"
       , xlab     = "Expression Level"
       , ylab     = "Density"
       , col.main = "#023047"
       , col.lab  = "#023047"
       , col.sub  = "#666666"
       , cex.sub  = 0.7
       , font.sub = 3)

  lines(density(dead_expr, na.rm = TRUE)
        , col = "#e63946"
        , lwd = 3)

  # Add vertical line at median (cutpoint)
  abline(v   = median(gene_expr)
         , col = "#fb8500"
         , lty = 2
         , lwd = 2)

  legend("topright"
         , legend = c("Alive", "Dead", "Median")
         , col    = c("#219ebc", "#e63946", "#fb8500")
         , lwd    = c(3, 3, 2)
         , lty    = c(1, 1, 2)
         , bty    = "n"
         , cex    = 0.8)
}
```



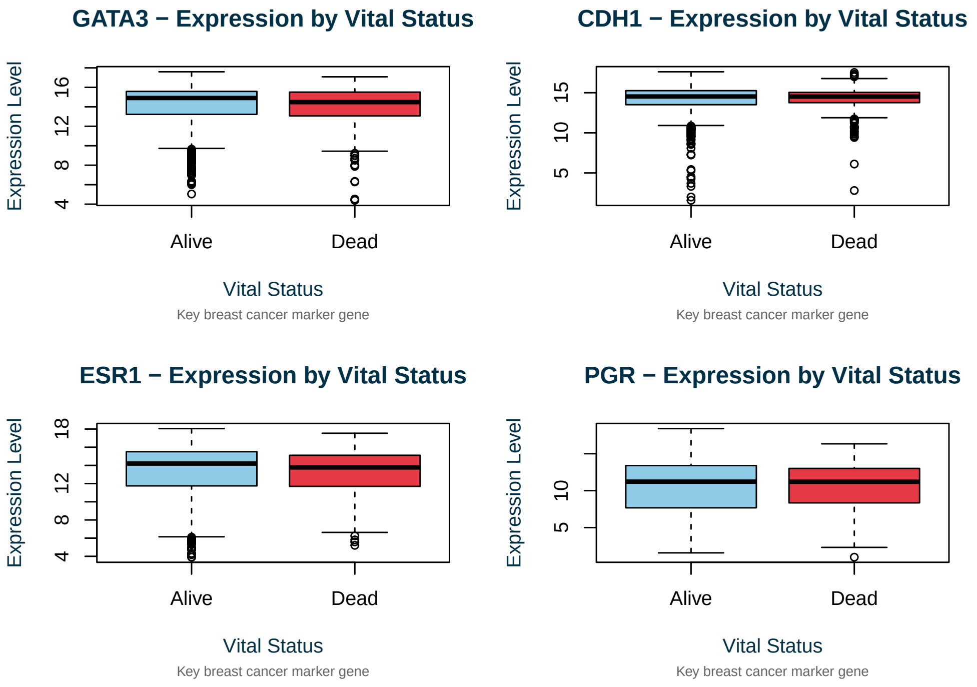
These five genes is likely to present subgroup markers: their expression defined nature of the patient cluster with different survivals risk. They do not act alone as strong predictors, but provide important biological signal which could improve modeling.

Key Cancer Genes Check

```r
key_genes <- c("GATA3", "CDH1", "ESR1", "PGR")
for(gene in key_genes) {
  cat("Gene:", gene, "\n")

  # Expression by vital status
  alive_expr <- GeneX_df[clinical_df$vital_status == "Alive", gene]
  dead_expr  <- GeneX_df[clinical_df$vital_status == "Dead", gene]

  cat("  Alive: mean=", round(mean(alive_expr, na.rm = TRUE), 2)
      , " sd=", round(sd(alive_expr, na.rm = TRUE), 2), "\n", sep = "")
  cat("  Dead:  mean=", round(mean(dead_expr, na.rm = TRUE), 2)
      , " sd=", round(sd(dead_expr, na.rm = TRUE), 2), "\n", sep = "")

  # T-test
  test <- t.test(alive_expr, dead_expr)
  cat("  T-test p-value:", format(test$p.value, scientific = TRUE), "\n")

  # Check if in top genes
  if(gene %in% rownames(top_genes)) {
    idx <- which(rownames(top_genes) == gene)
    cat("  Rank in DE analysis:", idx, "\n")
    cat("  logFC:", round(top_genes[gene, "logFC"], 3), "\n")
    cat("  FDR:", format(top_genes[gene, "adj.P.Val"], scientific = TRUE), "\n")
  } else {
    cat("  Not in top 100 DE genes\n")
  }
  cat("\n")
}
```

```
## Gene: GATA3
##   Alive: mean=14.17 sd=2.14
##   Dead:  mean=13.87 sd=2.34
##   T-test p-value: 9.546172e-02
##   Rank in DE analysis: 2017
##   logFC: -0.298
##   FDR: 1.881517e-01
##
## Gene: CDH1
##   Alive: mean=14.08 sd=1.95
##   Dead:  mean=14.12 sd=1.83
##   T-test p-value: 7.598542e-01
##   Rank in DE analysis: 4489
##   logFC: 0.044
##   FDR: 8.568828e-01
##
## Gene: ESR1
##   Alive: mean=13.24 sd=3.1
##   Dead:  mean=13.09 sd=2.85
##   T-test p-value: 5.200503e-01
##   Rank in DE analysis: 3868
##   logFC: -0.144
##   FDR: 6.990844e-01
##
## Gene: PGR
##   Alive: mean=10.58 sd=3.38
##   Dead:  mean=10.63 sd=3.18
##   T-test p-value: 8.278284e-01
##   Rank in DE analysis: 4641
##   logFC: 0.054
##   FDR: 8.982142e-01
```

```r
# Box plot
par(mfrow = c(2, 2))
for(gene in key_genes) {
  boxplot(GeneX_df[, gene] ~ clinical_df$vital_status
          , col      = c("#8ecae6", "#e63946")
          , main     = paste(gene, "- Expression by Vital Status")
          , sub      = "Key breast cancer marker gene"
          , xlab     = "Vital Status"
          , ylab     = "Expression Level"
          , names    = c("Alive", "Dead")
          , col.main = "#023047"
          , col.lab  = "#023047"
          , col.sub  = "#666666"
          , cex.sub  = 0.7
          , font.sub = 3)
}
```



Classical breast cancer markers **(GATA3, CDH1, ESR1, PGR)** shows no dfferent expression between Alive and Dead groups (all $p > 0.05$).

K-Means Clustering

```r
top50_data <- GeneX_df[, rownames(top_genes)[1:50]]

set.seed(42)
for(k in 2:4) {
  kmeans_result <- kmeans(scale(top50_data), centers = k, nstart = 25)

  cat(sprintf("\n--- K=%d CLUSTERS ---\n", k))

  # Cluster vs survival
  cluster_table <- table(kmeans_result$cluster, clinical_df$vital_status)
  print(cluster_table)

  # Chi-square test
  chi_test <- chisq.test(cluster_table)
  cat(sprintf("Chi-square p=%.4e %s\n"
              , chi_test$p.value
              , ifelse(chi_test$p.value < 0.05, "*** SIGNIFICANT", "")))

  # Cluster sizes
  cat("Cluster sizes:", table(kmeans_result$cluster), "\n")
}
```

```
##
```

```
## --- K=2 CLUSTERS ---
##
##     Alive Dead
##   1   852  138
##   2   177   63
## Chi-square p=5.8918e-06 *** SIGNIFICANT
## Cluster sizes: 990 240
##
## --- K=3 CLUSTERS ---
##
##     Alive Dead
##   1   297   38
##   2   683  117
##   3    49   46
## Chi-square p=5.8567e-18 *** SIGNIFICANT
## Cluster sizes: 335 800 95
##
## --- K=4 CLUSTERS ---
##
##     Alive Dead
##   1   473   69
##   2   357   65
##   3   163   26
##   4    36   41
## Chi-square p=6.7112e-18 *** SIGNIFICANT
## Cluster sizes: 542 422 189 77
```

```r
kmeans_2 <- kmeans(scale(top50_data), centers = 2, nstart = 25)
clinical_df$cluster <- paste0("C", kmeans_2$cluster)
print(table(clinical_df$cluster, clinical_df$vital_status))
```

```
##
##      Alive Dead
##   C1   177   63
##   C2   852  138
```

PCA Visualization by Clusters

```r
# PCA
pca_result <- prcomp(top50_data, scale. = TRUE)

# Variance explained
var_exp <- summary(pca_result)$importance[2, 1:10]
cat("Variance explained by first 10 PCs:\n")
```

```
## Variance explained by first 10 PCs:
```

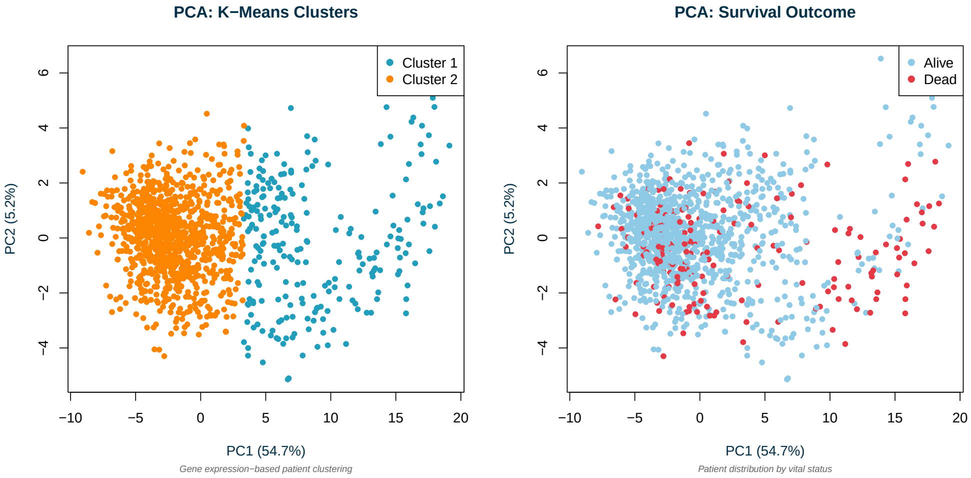
```r
print(round(var_exp, 3))
```

```
##   PC1   PC2   PC3   PC4   PC5   PC6   PC7   PC8   PC9  PC10
## 0.547 0.052 0.035 0.030 0.025 0.019 0.017 0.016 0.015 0.013
```

```r
cat("\nCumulative variance (PC1-10):", round(sum(var_exp), 3), "\n\n")
```

```
##
## Cumulative variance (PC1-10): 0.768
```

```r
par(mfrow = c(1, 2), bg = "white")

# Plot 1: Color by cluster
plot(pca_result$x[, 1]
     , pca_result$x[, 2]
     , col      = ifelse(clinical_df$cluster == "C1", "#219ebc", "#fb8500")
     , pch      = 19
     , cex      = 0.8
     , xlab     = paste0("PC1 (", round(var_exp[1] * 100, 1), "%)")
     , ylab     = paste0("PC2 (", round(var_exp[2] * 100, 1), "%)")
     , main     = "PCA: K-Means Clusters"
     , sub      = "Gene expression-based patient clustering"
     , col.main = "#023047"
     , col.lab  = "#023047"
     , col.sub  = "#666666"
     , cex.sub  = 0.7
     , font.sub = 3)

legend("topright"
       , legend = c("Cluster 1", "Cluster 2")
       , col    = c("#219ebc", "#fb8500")
       , pch    = 19)

# Plot 2: Color by survival
plot(pca_result$x[, 1]
     , pca_result$x[, 2]
     , col      = ifelse(clinical_df$vital_status == "Dead", "#e63946", "#8ecae6")
     , pch      = 19
     , cex      = 0.8
     , xlab     = paste0("PC1 (", round(var_exp[1] * 100, 1), "%)")
     , ylab     = paste0("PC2 (", round(var_exp[2] * 100, 1), "%)")
     , main     = "PCA: Survival Outcome"
     , sub      = "Patient distribution by vital status"
     , col.main = "#023047"
     , col.lab  = "#023047"
     , col.sub  = "#666666"
     , cex.sub  = 0.7
     , font.sub = 3)

legend("topright"
       , legend = c("Alive", "Dead")
       , col    = c("#8ecae6", "#e63946")
       , pch    = 19)
```



PCA reveals a strong molecular structure in the dataset, **PCA1** (54.7% of variance) clearly separate two expression defined clusters, confirming that the DE genes capture major biological subtypes. However on the right plot, the PCA space does not separate Alive vs Dead patients, indicating that survival differences are not driven by global gene-expression variation.

Clinical vs Gene Correlations

```
# Numeric clinical variables
numeric_clinical <- c("age_at_index"
                      , "initial_weight"
                      , "days_to_last_follow_up"
                      , "age_at_diagnosis"
                      , "days_to_birth")

# For each clinical variable
for(clin_var in numeric_clinical) {
  cat("---", clin_var, "---\n")

  clin_values <- clinical_df[[clin_var]]

  # Calculate correlations with all 20 genes
  cors <- numeric(20)
  for(i in 1:20) {
    cors[i] <- cor(clin_values, gene_subset[, i], use = "complete.obs")
  }
  names(cors) <- top20_genes

  # Sort and show top 5
  cors_sorted <- sort(abs(cors), decreasing = TRUE)
  cat("Top 5 correlated genes:\n")
  for(i in 1:5) {
    gene <- names(cors_sorted)[i]
    cat(sprintf("  %s: r=%.3f\n", gene, cors[gene]))
  }
  cat("\n")
}
```

```
## --- age_at_index ---
## Top 5 correlated genes:
##   VEGFD: r=-0.135
##   LINC01235: r=-0.133
##   LINC02511: r=-0.100
##   ATF3: r=-0.097
##   PSD2: r=-0.093
##
## --- initial_weight ---
## Top 5 correlated genes:
##   LYVE1: r=0.181
##   FHL1: r=0.139
##   GPX3: r=0.138
##   VEGFD: r=0.134
##   CST1: r=-0.132
##
## --- days_to_last_follow_up ---
## Top 5 correlated genes:
##   ATF3: r=0.130
##   SNORD104: r=-0.103
##   LINC02511: r=0.076
##   VEGFD: r=0.071
##   ADH4: r=0.070
##
## --- age_at_diagnosis ---
## Top 5 correlated genes:
##   VEGFD: r=-0.136
##   LINC01235: r=-0.134
##   LINC02511: r=-0.111
##   ATF3: r=-0.103
##   PSD2: r=-0.095
##
## --- days_to_birth ---
## Top 5 correlated genes:
##   LINC01235: r=0.137
##   VEGFD: r=0.135
##   LINC02511: r=0.105
##   ATF3: r=0.102
##   PSD2: r=0.091
```

From this correlation, we can say that,

- Age-related variables **(age_at_index, age_at_diagnosis, days_to_birth)** show almost identical correlated genes **(VEGFD, LINC01235, LINC02511, ATF3, PSD2)** → expected because these age variables are themselves highly collinear.
- Initial weight shows slightly stronger associations (up to $r = 0.18$), mostly with genes involved in immune/metabolic activity (LYVE1, FHL1, GPX3, VEGFD).
- Follow-up time correlates weakly with stress/response genes **(ATF3, LINC02511)**, but effect sizes remain very small.

Categorical Clinical vs Genes

```r
# Tumor stage vs genes
cat("1. TUMOR STAGE vs GENES\n")
```

```
## 1. TUMOR STAGE vs GENES
```

```r
stage_simple <- substr(clinical_df$ajcc_pathologic_t, 1, 2)
stage_simple[!stage_simple %in% c("T1", "T2", "T3", "T4")] <- NA

cat("Stage distribution:\n")
```

```
## Stage distribution:
```

```r
print(table(stage_simple, useNA = "ifany"))
```

```
## stage_simple
##   T1   T2   T3   T4 <NA>
##  293  658  132   41  106
```

```r
cat("\n")
```

```r
cat("Testing top 5 genes across stages (ANOVA):\n")
```

```
## Testing top 5 genes across stages (ANOVA):
```

```r
for(i in 1:5) {
  gene      <- top20_genes[i]
  gene_expr <- gene_subset[, i]

  df_test <- data.frame(expr = gene_expr, stage = stage_simple)
  df_test <- df_test[!is.na(df_test$stage), ]

  if(length(unique(df_test$stage)) > 1) {
    aov_result <- aov(expr ~ stage, data = df_test)
    p_val      <- summary(aov_result)[[1]]$`Pr(>F)`[1]
    means      <- tapply(df_test$expr, df_test$stage, mean)

    cat(sprintf("  %s: p=%.4f %s\n"
                , gene
                , p_val
                , ifelse(p_val < 0.05, "*** SIGNIFICANT", "")))
    cat(sprintf("    T1=%.2f, T2=%.2f, T3=%.2f, T4=%.2f\n"
                , means["T1"], means["T2"], means["T3"], means["T4"]))
  }
}
```

```
##   LINC01235: p=0.2195
##     T1=7.02, T2=6.98, T3=7.30, T4=7.34
##   APOB: p=0.0709
##     T1=3.71, T2=3.26, T3=3.50, T4=3.75
##   LYVE1: p=0.2767
##     T1=7.33, T2=7.14, T3=7.32, T4=7.67
##   LINC01497: p=0.0021 *** SIGNIFICANT
##     T1=1.17, T2=1.07, T3=1.67, T4=1.09
##   AC104211.1: p=0.0265 *** SIGNIFICANT
##     T1=3.44, T2=3.27, T3=3.68, T4=3.73
```

```r
cat("\n")
```

```r
# Treatment vs genes
cat("2. PRIOR TREATMENT vs GENES\n")
```

```
## 2. PRIOR TREATMENT vs GENES
```

```r
treat <- clinical_df$prior_treatment
cat("Treatment distribution:\n")
```

```
## Treatment distribution:
```

```r
print(table(treat, useNA = "ifany"))
```

```
## treat
##           No Not Reported          Yes         <NA>
##         1100            1           85           44
```

```r
cat("\n")
```

```r
cat("Testing top 5 genes (t-test: Yes vs No):\n")
```

```
## Testing top 5 genes (t-test: Yes vs No):
```

```r
for(i in 1:5) {
  gene     <- top20_genes[i]
  expr_yes <- gene_subset[treat == "Yes", i]
  expr_no  <- gene_subset[treat == "No", i]

  if(length(expr_yes) > 2 && length(expr_no) > 2) {
    test <- t.test(expr_yes, expr_no)
    cat(sprintf("  %s: Yes=%.2f, No=%.2f, p=%.4f %s\n"
                , gene
                , mean(expr_yes, na.rm = TRUE)
                , mean(expr_no, na.rm = TRUE)
                , test$p.value
                , ifelse(test$p.value < 0.05, "*** SIGNIFICANT", "")))
  }
}
```

```
##   LINC01235: Yes=7.12, No=7.06, p=0.8278
##   APOB: Yes=3.37, No=3.42, p=0.8685
##   LYVE1: Yes=7.24, No=7.23, p=0.9602
##   LINC01497: Yes=1.16, No=1.16, p=0.9740
##   AC104211.1: Yes=3.41, No=3.36, p=0.8008
```

```
cat("\n")
```

```
# Race vs genes
cat("3. RACE vs GENES\n")
```

```
## 3. RACE vs GENES
```

```
race        <- clinical_df$race
race_binary <- ifelse(race == "white", "White"
                      , ifelse(race == "black or african american", "Black", NA))

cat("Testing top 5 genes (White vs Black):\n")
```

```
## Testing top 5 genes (White vs Black):
```

```
for(i in 1:5) {
  gene        <- top20_genes[i]
  expr_white  <- gene_subset[race_binary == "White" & !is.na(race_binary), i]
  expr_black  <- gene_subset[race_binary == "Black" & !is.na(race_binary), i]

  if(length(expr_white) > 2 && length(expr_black) > 2) {
    test <- t.test(expr_white, expr_black)
    cat(sprintf("  %s: White=%.2f, Black=%.2f, p=%.4f %s\n"
                , gene
                , mean(expr_white, na.rm = TRUE)
                , mean(expr_black, na.rm = TRUE)
                , test$p.value
                , ifelse(test$p.value < 0.05, "*** SIGNIFICANT", "")))
  }
}
```

```
##   LINC01235: White=7.13, Black=7.25, p=0.4371
##   APOB: White=3.78, Black=2.60, p=0.0000 *** SIGNIFICANT
##   LYVE1: White=7.48, Black=6.69, p=0.0000 *** SIGNIFICANT
##   LINC01497: White=1.33, Black=0.87, p=0.0004 *** SIGNIFICANT
##   AC104211.1: White=3.60, Black=2.65, p=0.0000 *** SIGNIFICANT
```

From the result , we can state that

- **Tumor stage:** Almost no gene is stage-dependent; only **LINC01497** and **AC104211.1** show small differences → weak association.
- **Prior treatment:** No gene shows expression changes → **no effect**.
- **Race:** Several genes (**APOB, LYVE1, LINC01497, AC104211.1**) differ between White vs Black patients → likely due to **subtype composition**, not race biology.

Interprete, clinical categorical variables have **minimal influence** on top gene expression patterns, except for race-related subtype differences.

Gene-Gene Correlations

```r
# Correlation matrix
gene_cor <- cor(gene_subset)

# Find highly correlated pairs
high_cor <- which(abs(gene_cor) > 0.7 & gene_cor != 1, arr.ind = TRUE)

if(nrow(high_cor) > 0) {
  cat("Highly correlated gene pairs (|r| > 0.7):\n")
  for(i in 1:nrow(high_cor)) {
    if(high_cor[i, 1] < high_cor[i, 2]) {
      gene1 <- rownames(gene_cor)[high_cor[i, 1]]
      gene2 <- colnames(gene_cor)[high_cor[i, 2]]
      r     <- gene_cor[high_cor[i, 1], high_cor[i, 2]]
      cat(sprintf("  %s <-> %s: r=%.3f\n", gene1, gene2, r))
    }
  }
} else {
  cat("No highly correlated pairs (genes are independent)\n")
}
```

```
## Highly correlated gene pairs (|r| > 0.7):
##   APOB <-> KLB: r=0.718
##   LYVE1 <-> LINC02511: r=0.706
##   APOB <-> AC007423.1: r=0.702
##   KLB <-> AC007423.1: r=0.759
##   APOB <-> GPX3: r=0.732
##   LYVE1 <-> GPX3: r=0.748
##   KLB <-> GPX3: r=0.726
##   AC007423.1 <-> GPX3: r=0.725
##   APOB <-> LVRN: r=0.717
##   LYVE1 <-> LVRN: r=0.702
##   KLB <-> LVRN: r=0.799
##   AC007423.1 <-> LVRN: r=0.754
##   GPX3 <-> LVRN: r=0.771
##   KLB <-> SLC2A4: r=0.702
##   GPX3 <-> SLC2A4: r=0.707
##   APOB <-> FHL1: r=0.723
##   LYVE1 <-> FHL1: r=0.800
##   KLB <-> FHL1: r=0.724
##   LINC02511 <-> FHL1: r=0.711
##   AC007423.1 <-> FHL1: r=0.724
##   GPX3 <-> FHL1: r=0.807
##   LVRN <-> FHL1: r=0.793
##   SLC2A4 <-> FHL1: r=0.707
##   LYVE1 <-> VEGFD: r=0.741
##   LINC02511 <-> VEGFD: r=0.730
##   GPX3 <-> VEGFD: r=0.729
##   LVRN <-> VEGFD: r=0.712
##   FHL1 <-> VEGFD: r=0.737
```

The top DE genes form one strong co-expression module **(APOB, LYVE1, KLB, GPX3, FHL1,**

VEGFD, LINC02511). They show very high correlations (|r| = 0.70˘0.80), meaning they act as a single biological program.

Outlier Detection on Top Gen

```
top_gene  <- top20_genes[1]
gene_expr <- gene_subset[, 1]

# Z-scores
z_scores <- scale(gene_expr)
outliers <- which(abs(z_scores) > 3)

cat("Top gene:", top_gene, "\n")
```

```
## Top gene: LINC01235
```

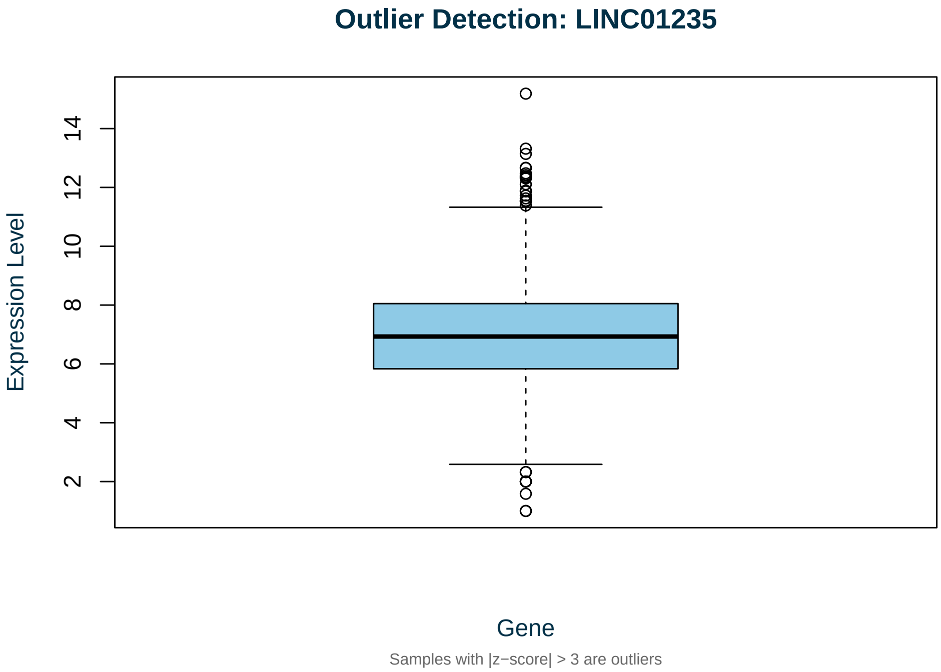
```
cat("Samples with |z-score| > 3:", length(outliers), "\n")
```

```
## Samples with |z-score| > 3: 7
```

```
if(length(outliers) > 0 & length(outliers) < 10) {
  cat("Outlier samples:\n")
  print(outliers)
}
```

```
## Outlier samples:
## [1]  321  367  383  506  528  987 1017
```

```
# Boxplot
boxplot(gene_expr
        , col      = "#8ecae6"
        , main     = paste("Outlier Detection:", top_gene)
        , sub      = "Samples with |z-score| > 3 are outliers"
        , xlab     = "Gene"
        , ylab     = "Expression Level"
        , col.main = "#023047"
        , col.lab  = "#023047"
        , col.sub  = "#666666"
        , cex.sub  = 0.7
        , font.sub = 3)
```



```
cat("\nOnly", length(outliers), "outliers (",
    round(length(outliers) / nrow(gene_subset) * 100, 1), "%) - acceptable\n")
```

```
##
## Only 7 outliers ( 0.6 %) - acceptable
```

The expression profile of LINC01235 shows only 7 extreme observations (0.6 exceeding the standard $|z| > 3$ threshold. This is well within accepted QC limits for large transcriptomic datasets, where up to 1–2% technical or biological outliers are considered normal.

Methodology

Data Preparing before fitting model

```
# --- 1. Define Predictor Names ---
clinical_col_num_names <- c("age_at_index"
                            , "initial_weight"
                            , "days_to_last_follow_up")

clinical_signi_obj_names <- c("tissue_type"
                              , "ajcc_pathologic_t"
                              , "classification_of_tumor"
                              , "follow_ups_disease_response"
                              , "prior_treatment"
                              , "tissue_or_organ_of_origin"
                              , "ethnicity")
```

```r
# ALL GENES (5000)
top_genes_list <- colnames(GeneX_df)

cat("=== SELECTED VARIABLES ===\n")
```

```
## === SELECTED VARIABLES ===
```

```r
cat("Numeric clinical:", length(clinical_col_num_names), "\n")
```

```
## Numeric clinical: 3
```

```r
cat("Categorical clinical:", length(clinical_signi_obj_names), "\n")
```

```
## Categorical clinical: 7
```

```r
cat("Genes: ALL", length(top_genes_list), "\n\n")
```

```
## Genes: ALL 5000
```

```r
# --- 2. Check Missing Values ---
cat("=== MISSING VALUES ===\n")
```

```
## === MISSING VALUES ===
```

```r
cat("Numeric:\n")
```

```
## Numeric:
```

```r
print(colSums(is.na(clinical_df[, clinical_col_num_names])))
```

```
##           age_at_index         initial_weight days_to_last_follow_up 
##                      0                     15                      3
```

```r
cat("\nCategorical:\n")
```

```
## 
## Categorical:
```

```r
print(colSums(is.na(clinical_df[, clinical_signi_obj_names])))
```

```
##                 tissue_type           ajcc_pathologic_t 
##                           0                          99 
##     classification_of_tumor follow_ups_disease_response 
##                           0                          76 
##             prior_treatment   tissue_or_organ_of_origin 
##                          44                           0 
##                   ethnicity 
##                           0
```

```r
cat("\n")
```

```r
# --- 3. Impute Numeric Variables (median) ---
clinical_numeric <- clinical_df[, clinical_col_num_names, drop = FALSE]

for (col in colnames(clinical_numeric)) {
    n_missing <- sum(is.na(clinical_numeric[[col]]))
    if (n_missing > 0) {
        median_val <- median(clinical_numeric[[col]], na.rm = TRUE)
        clinical_numeric[[col]][is.na(clinical_numeric[[col]])] <- median_val
        cat(sprintf("Imputed %d in %s (median=%.2f)\n", n_missing, col, median_val))
    }
}
```

```
## Imputed 15 in initial_weight (median=220.00)
## Imputed 3 in days_to_last_follow_up (median=890.00)
```

```r
cat("\n")
```

```r
clinical_numeric <- data.frame(lapply(clinical_numeric, as.numeric))
rownames(clinical_numeric) <- rownames(clinical_df)


# --- 4. Impute Categorical Variables (mode) ---
clinical_categorical <- clinical_df[, clinical_signi_obj_names, drop = FALSE]

for (col in colnames(clinical_categorical)) {
    n_missing <- sum(is.na(clinical_categorical[[col]]))
    if (n_missing > 0) {
        mode_val <- names(sort(table(clinical_categorical[[col]]), decreasing = TRUE))[1]
        clinical_categorical[[col]][is.na(clinical_categorical[[col]])] <- mode_val
        cat(sprintf("Imputed %d in %s (mode=%s)\n", n_missing, col, mode_val))
    }
}
```

```
## Imputed 99 in ajcc_pathologic_t (mode=T2)
## Imputed 76 in follow_ups_disease_response (mode=TF-Tumor Free)
## Imputed 44 in prior_treatment (mode=No)
```

```r
cat("\n")
```

```r
clinical_categorical <- data.frame(lapply(clinical_categorical, as.factor))


# --- 5. One-Hot Encode Categorical ---
valid_factors <- sapply(clinical_categorical, function(x) {
    n_levels <- length(levels(droplevels(x)))
    return(n_levels >= 2)
})

clinical_categorical_valid <- clinical_categorical[, valid_factors, drop = FALSE]
```

```r
if (sum(!valid_factors) > 0) {
    cat("Dropped constant columns:",
        paste(names(clinical_categorical)[!valid_factors], collapse=", "), "\n\n")
}

clinical_ohe <- model.matrix(~ . - 1, data = clinical_categorical_valid)
clinical_ohe <- as.data.frame(clinical_ohe)
rownames(clinical_ohe) <- rownames(clinical_df)

cat("One-Hot Encoding: ", ncol(clinical_categorical_valid), "->", ncol(clinical_ohe), "\n\n")
```

```
## One-Hot Encoding:  7 -> 54
```

```r
# --- 6. Gene Expression Quality Check ---
cat("=== GENE EXPRESSION QUALITY CHECK ===\n")
```

```
## === GENE EXPRESSION QUALITY CHECK ===
```

```r
gene_data <- GeneX_df[, top_genes_list, drop = FALSE]
gene_data <- as.data.frame(gene_data)

cat("Missing Values:\n")
```

```
## Missing Values:
```

```r
missing_per_gene <- colSums(is.na(gene_data))
missing_pct <- 100 * missing_per_gene / nrow(gene_data)

cat("  Genes with missing:", sum(missing_per_gene > 0), "/", ncol(gene_data), "\n")
```

```
##   Genes with missing: 0 / 5000
```

```r
if(sum(missing_per_gene > 0) > 0) {
    missing_summary <- data.frame(
        Gene           = names(missing_per_gene)
        , N_Missing    = missing_per_gene
        , Pct_Missing = round(missing_pct, 2)
    )
    print(head(missing_summary[order(-missing_summary$N_Missing), ], 10))
}
cat("\n")
```

```r
# --- 7. Distribution Analysis ---
cat("Distribution Analysis:\n")
```

```
## Distribution Analysis:
```

```r
sample_genes <- colnames(gene_data)[1:min(20, ncol(gene_data))]
distribution_summary <- data.frame(
    Gene = character()
    , Mean = numeric()
    , Median = numeric()
    , Skewness = numeric()
    , Impute_Method = character()
    , stringsAsFactors = FALSE
)

for(gene in sample_genes) {
    vals <- gene_data[[gene]][!is.na(gene_data[[gene]])]
    m <- mean(vals)
    s <- sd(vals)
    skew <- mean(((vals - m) / s)^3)

    distribution_summary <- rbind(distribution_summary
                                  , data.frame(
                                      Gene = gene
                                      , Mean = round(m, 2)
                                      , Median = round(median(vals), 2)
                                      , Skewness = round(skew, 3)
                                      , Impute_Method = ifelse(abs(skew) < 0.5, "Mean", "Median")
                                  ))
}

print(distribution_summary)
```

```
##          Gene  Mean Median Skewness Impute_Method
## 1      CLEC3A  5.93   4.52    0.641        Median
## 2     SCGB2A2 11.13  11.71   -0.338          Mean
## 3        CPB1  8.31   7.68    0.715        Median
## 4       GSTM1  5.36   5.49    0.160          Mean
## 5        TFF1  9.71  10.79   -0.594        Median
## 6     SCGB1D2  9.03   9.27   -0.046          Mean
## 7       KCNJ3  6.69   5.93    0.347          Mean
## 8       MUCL1  9.15   8.95    0.134          Mean
## 9   LINC00993  8.86  10.50   -0.816        Median
## 10    ANKRD30A  8.65  10.33   -0.756        Median
## 11   DSCAM-AS1  4.88   4.28    0.380          Mean
## 12      S100A7  4.77   3.58    0.914        Median
## 13         PIP 10.05  10.67   -0.309          Mean
## 14    SERPINA6  4.98   3.91    0.742        Median
## 15 AC093001.1  7.54   7.38   -0.037          Mean
## 16      VSTM2A  4.09   2.32    0.951        Median
## 17      ADIPOQ  7.87   8.36   -0.145          Mean
## 18      HMGCS2  7.70   8.00   -0.109          Mean
## 19       ADH1B  8.81   9.36   -0.301          Mean
## 20       PRAME  5.96   4.58    0.491          Mean
```

```r
cat("\n")
```

```r
# --- 8. Determine Imputation Strategy ---
cat("Imputation Strategy:\n")
```

```
## Imputation Strategy:
```

```r
all_skewness <- sapply(1:ncol(gene_data), function(i) {
    vals <- gene_data[!is.na(gene_data[, i]), i]
    if(length(vals) < 3) return(0)
    m <- mean(vals)
    s <- sd(vals)
    if(s == 0) return(0)
    mean(((vals - m) / s)^3)
})

median_skew <- median(abs(all_skewness), na.rm = TRUE)
cat("  Median absolute skewness:", round(median_skew, 3), "\n")
```

```
##   Median absolute skewness: 0.471
```

```r
impute_strategy <- ifelse(median_skew < 0.5, "mean", "median")
cat("  Selected method:", toupper(impute_strategy), "\n\n")
```

```
##   Selected method: MEAN
```

```r
# --- 9. Apply Gene Imputation ---
cat("Applying Gene Imputation:\n")
```

```
## Applying Gene Imputation:
```

```r
gene_data_imputed <- gene_data
n_imputed <- 0

for(gene in colnames(gene_data_imputed)) {
    n_missing <- sum(is.na(gene_data_imputed[[gene]]))
    if(n_missing > 0) {
        vals <- gene_data_imputed[[gene]]
        impute_val <- if(impute_strategy == "median") {
            median(vals, na.rm = TRUE)
        } else {
            mean(vals, na.rm = TRUE)
        }
        gene_data_imputed[[gene]][is.na(gene_data_imputed[[gene]])] <- impute_val
        n_imputed <- n_imputed + n_missing
    }
}

cat("  Values imputed:", n_imputed, "\n")
```

```
##   Values imputed: 0
```

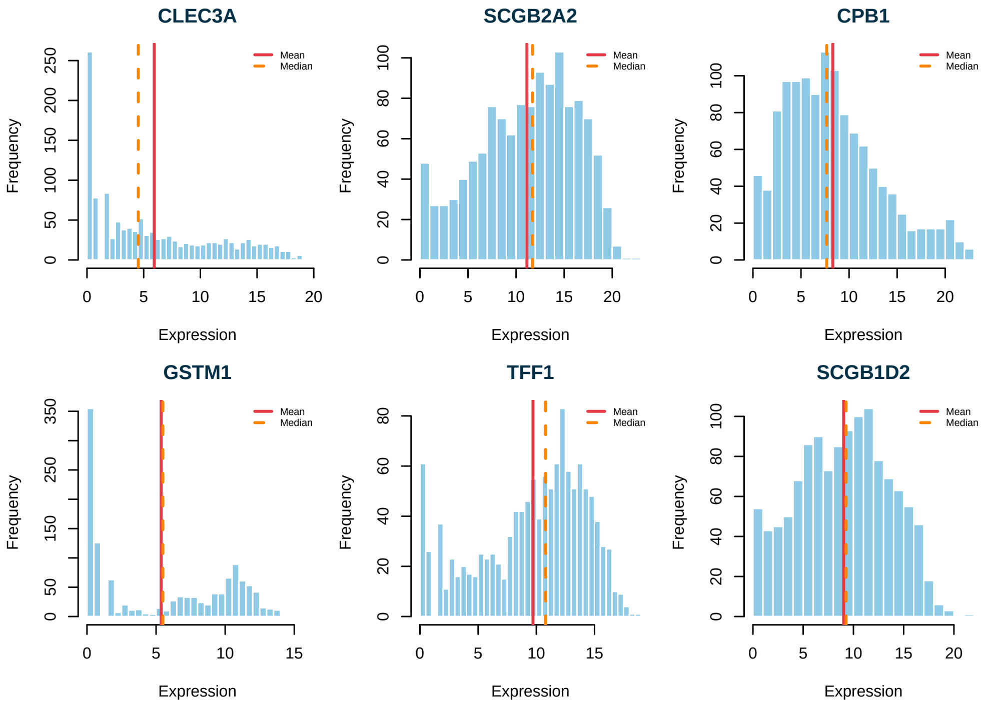
```r
cat("  Method used:", toupper(impute_strategy), "\n")
```

```
##   Method used: MEAN
```

```r
cat("  Remaining NA:", sum(is.na(gene_data_imputed)), "\n\n")
```

```
##   Remaining NA: 0
```

```r
# --- 10. Visual Distribution Check ---
par(mfrow = c(2, 3), mar = c(4, 4, 3, 1))

for(i in 1:min(6, ncol(gene_data))) {
    vals <- gene_data[[i]][!is.na(gene_data[[i]])]
    m <- mean(vals)
    s <- sd(vals)
    skew <- mean(((vals - m) / s)^3)

    hist(vals
         , breaks = 30
         , col = "#8ecae6"
         , border = "white"
         , main = colnames(gene_data)[i]
         , sub = paste("Skewness:", round(skew, 2))
         , xlab = "Expression"
         , ylab = "Frequency"
         , col.main = "#023047"
         , col.sub = "#666666")

    abline(v = m
           , col = "#e63946"
           , lwd = 2)

    abline(v = median(vals)
           , col = "#fb8500"
           , lwd = 2
           , lty = 2)

    legend("topright"
           , legend = c("Mean", "Median")
           , col = c("#e63946", "#fb8500")
           , lwd = 2
           , lty = c(1, 2)
           , cex = 0.6
           , bty = "n")
}
```



```
par(mfrow = c(1, 1))

gene_data <- gene_data_imputed

cat("Gene Expression Quality Check Complete\n")
```

```
## Gene Expression Quality Check Complete
```

```
cat("Dimensions:", nrow(gene_data), "x", ncol(gene_data), "\n\n")
```

```
## Dimensions: 1230 x 5000
```

```
# --- 11. Standardize ---
cat("=== STANDARDIZATION ===\n\n")
```

```
## === STANDARDIZATION ===
```

```
clinical_numeric_scaled <- as.data.frame(scale(clinical_numeric))

cat("Numeric clinical:\n")
```

```
## Numeric clinical:
```

```
for (var in colnames(clinical_numeric_scaled)) {
    cat(sprintf("  %s: mean=%.4f, sd=%.4f\n"
                  , var
                  , mean(clinical_numeric_scaled[[var]])
                  , sd(clinical_numeric_scaled[[var]])))
}
```

```
##   age_at_index: mean=-0.0000, sd=1.0000
##   initial_weight: mean=-0.0000, sd=1.0000
##   days_to_last_follow_up: mean=0.0000, sd=1.0000
```

```r
cat("\n")
```

```r
gene_data_scaled <- as.data.frame(scale(gene_data))

cat("Genes (", ncol(gene_data_scaled), "):\n", sep="")
```

```
## Genes (5000):
```

```r
cat(sprintf("  Mean: %.2e, SD: %.4f\n"
            , mean(as.matrix(gene_data_scaled))
            , sd(as.matrix(gene_data_scaled))))
```

```
##   Mean: 2.99e-18, SD: 0.9996
```

```r
cat(sprintf("  Range: [%.2f, %.2f]\n\n"
            , min(gene_data_scaled)
            , max(gene_data_scaled)))
```

```
##   Range: [-8.57, 11.20]
```

```r
# --- 12. Combine Features ---
data_predictors <- cbind(clinical_numeric_scaled
                         , clinical_ohe
                         , gene_data_scaled)

data_predictors$Y <- ifelse(clinical_df$vital_status == "Dead", 1, 0)

cat("=== FINAL DATASET ===\n")
```

```
## === FINAL DATASET ===
```

```r
cat("Samples:", nrow(data_predictors), "\n")
```

```
## Samples: 1230
```

```r
cat("Features:", ncol(data_predictors) - 1, "\n")
```

```
## Features: 5057
```

```r
cat("  Numeric clinical:", ncol(clinical_numeric_scaled), "\n")
```

```
##   Numeric clinical: 3
```

```r
cat("  Categorical (OHE):", ncol(clinical_ohe), "\n")
```

```
##   Categorical (OHE): 54
```

```r
cat("  Genes:", ncol(gene_data_scaled), "\n\n")
```

```
##   Genes: 5000
```

```r
table_Y <- table(data_predictors$Y)
print(table_Y)
```

```
## 
##    0    1 
## 1029  201
```

```r
cat(sprintf("  Alive: %d (%.1f%%)\n", table_Y[1], 100 * table_Y[1] / sum(table_Y)))
```

```
##   Alive: 1029 (83.7%)
```

```r
cat(sprintf("  Dead: %d (%.1f%%)\n", table_Y[2], 100 * table_Y[2] / sum(table_Y)))
```

```
##   Dead: 201 (16.3%)
```

```r
cat(sprintf("  Imbalance: %.2f:1\n", table_Y[1] / table_Y[2]))
```

```
##   Imbalance: 5.12:1
```

Train/Test Split

```r
# Set seed for reproducibility
set.seed(42)

# Separate features and target
X_all <- data_predictors[, -which(names(data_predictors) == "Y")]
Y_all <- data_predictors$Y

# Create train/test split (80/20)
train_indices <- sample(1:nrow(data_predictors), size = 0.8 * nrow(data_predictors))

X_train <- as.matrix(X_all[train_indices, ])
X_test <- as.matrix(X_all[-train_indices, ])
Y_train <- Y_all[train_indices]
Y_test <- Y_all[-train_indices]

# Calculate number of clinical features
n_clinical <- ncol(clinical_numeric_scaled) + ncol(clinical_ohe)

cat("=== TRAIN/TEST SPLIT ===\n")
```

```
## === TRAIN/TEST SPLIT ===
```

```r
cat("Training set:\n")
```

```
## Training set:
```

```r
cat("  Samples:", nrow(X_train), "\n")
```

```
##   Samples: 984
```

```r
cat("  Features:", ncol(X_train), "\n")
```

```
##   Features: 5057
```

```r
cat("  Dead:", sum(Y_train == 1), sprintf("(%.1f%%)\n", 100 * sum(Y_train == 1) / length(Y_train)))
```

```
##   Dead: 161 (16.4%)
```

```r
cat("  Alive:", sum(Y_train == 0), sprintf("(%.1f%%)\n", 100 * sum(Y_train == 0) / length(Y_train)))
```

```
##   Alive: 823 (83.6%)
```

```r
cat("\n")
```

```r
cat("Test set:\n")
```

```
## Test set:
```

```r
cat("  Samples:", nrow(X_test), "\n")
```

```
##   Samples: 246
```

```r
cat("  Features:", ncol(X_test), "\n")
```

```
##   Features: 5057
```

```r
cat("  Dead:", sum(Y_test == 1), sprintf("(%.1f%%)\n", 100 * sum(Y_test == 1) / length(Y_test)))
```

```
##   Dead: 40 (16.3%)
```

```r
cat("  Alive:", sum(Y_test == 0), sprintf("(%.1f%%)\n", 100 * sum(Y_test == 0) / length(Y_test)))
```

```
##   Alive: 206 (83.7%)
```

```r
cat("\n")
```

```r
cat("Clinical features:", n_clinical, "\n")
```

```
## Clinical features: 57
```

```r
cat("Genomic features:", ncol(X_train) - n_clinical, "\n")
```

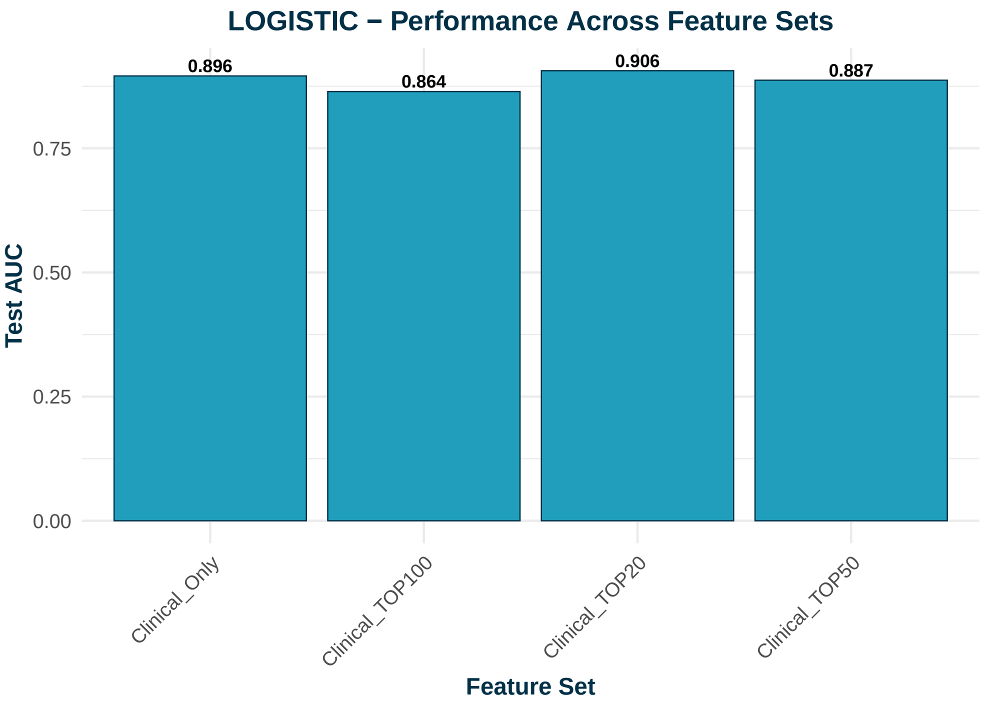
```
## Genomic features: 5000
```

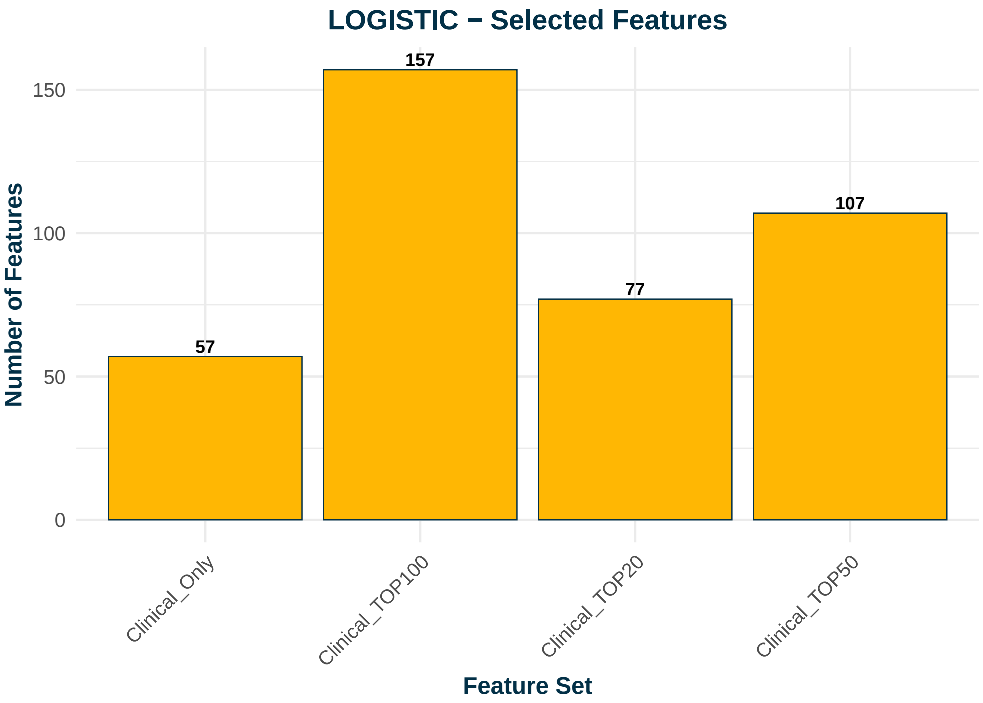
Logistic regiression

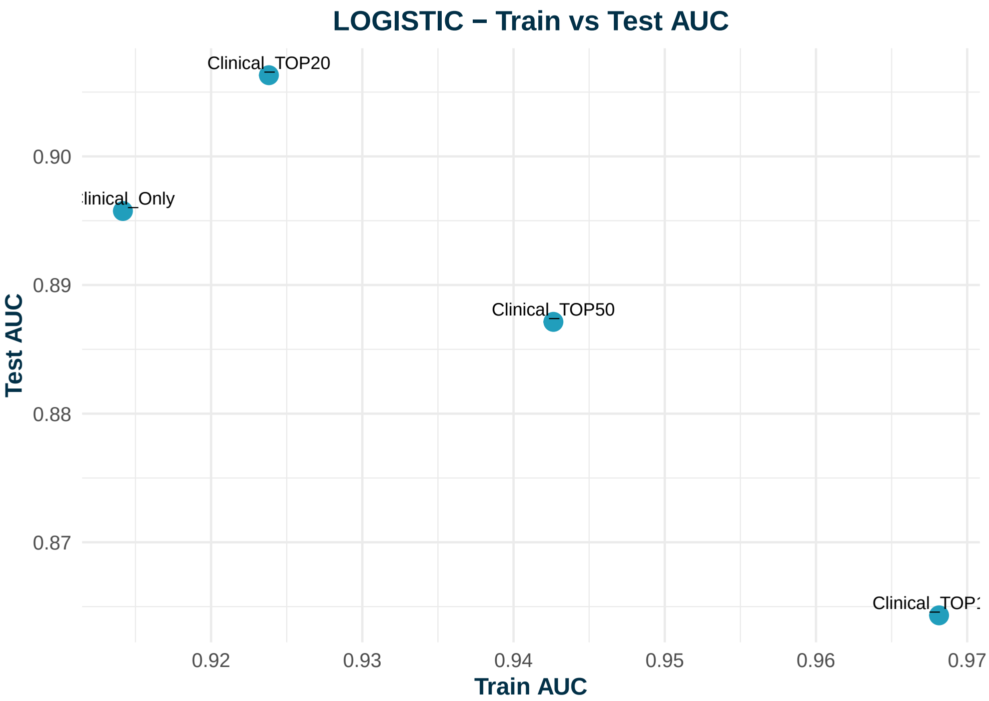
Logistic regression is used here only on feature sets that are not high-dimensional, because the model becomes unstable when the number of predictors is large. This is due to the form of its loss function, which cannot be minimized reliably when $p >> n$. Recall that logistic regression estimates coefficients by maximizing the log-likelihood:

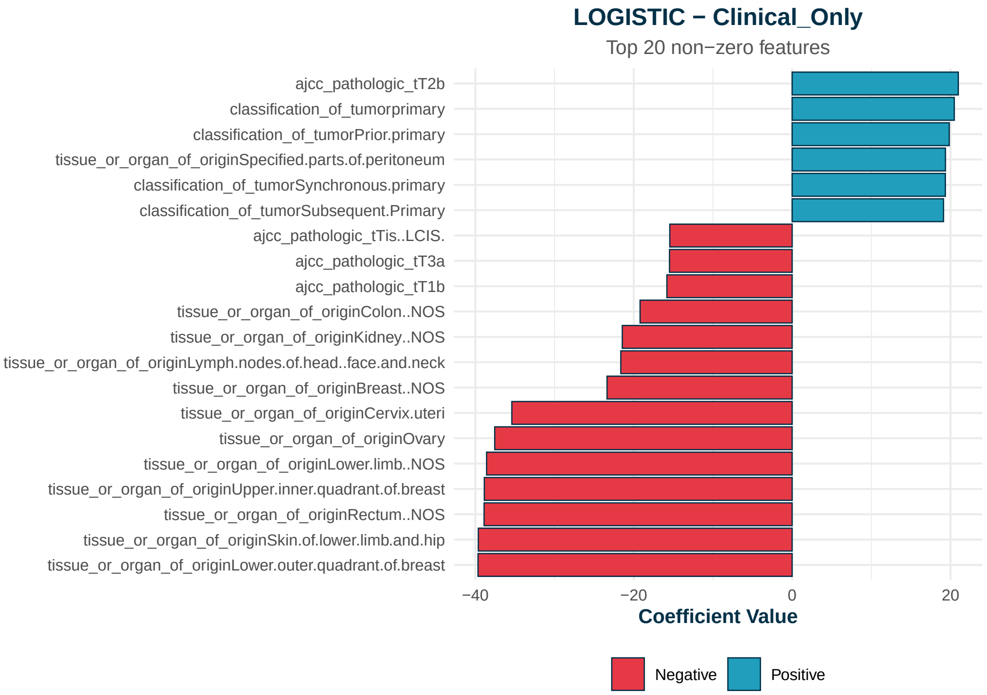
```r
logistic_results <- fit_single_model_across_features(
    model_type = "logistic"
    , X_train_all = X_train
    , X_test_all = X_test
    , Y_train = Y_train
    , Y_test = Y_test
    , n_clinical = n_clinical
    , top_genes_ranked = top_genes
    , gene_sets = c(100, 50, 20)
)
```

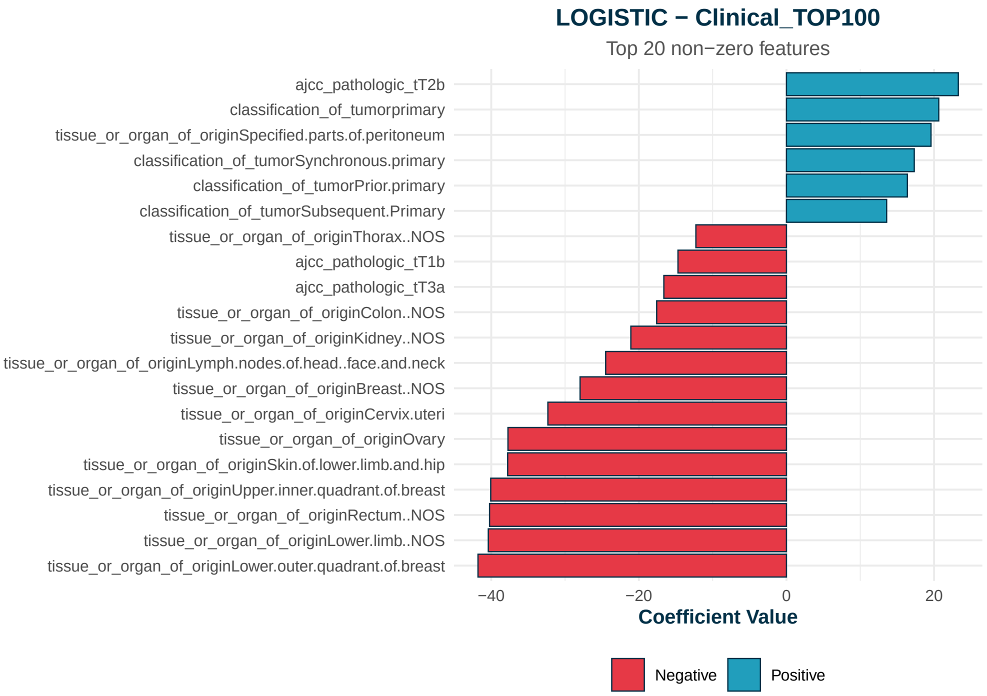
```
##
## === FITTING LOGISTIC ACROSS FEATURE SETS ===
##
## Fitting Clinical_Only...

## Fitting Clinical_TOP100...

## Fitting Clinical_TOP50...

## Fitting Clinical_TOP20...

##
## === SUMMARY TABLE ===
##       Feature_Set    Model Features Train_AUC  Test_AUC Test_Accuracy
## 1   Clinical_Only LOGISTIC       57 0.9141680 0.8957524     0.9024390
## 2 Clinical_TOP100 LOGISTIC      157 0.9681366 0.8643204     0.8373984
## 3  Clinical_TOP50 LOGISTIC      107 0.9426353 0.8871359     0.9024390
## 4  Clinical_TOP20 LOGISTIC       77 0.9238206 0.9063107     0.9065041
## Exported metrics to: model_metrics/logistic_across_features_metrics.csv
```

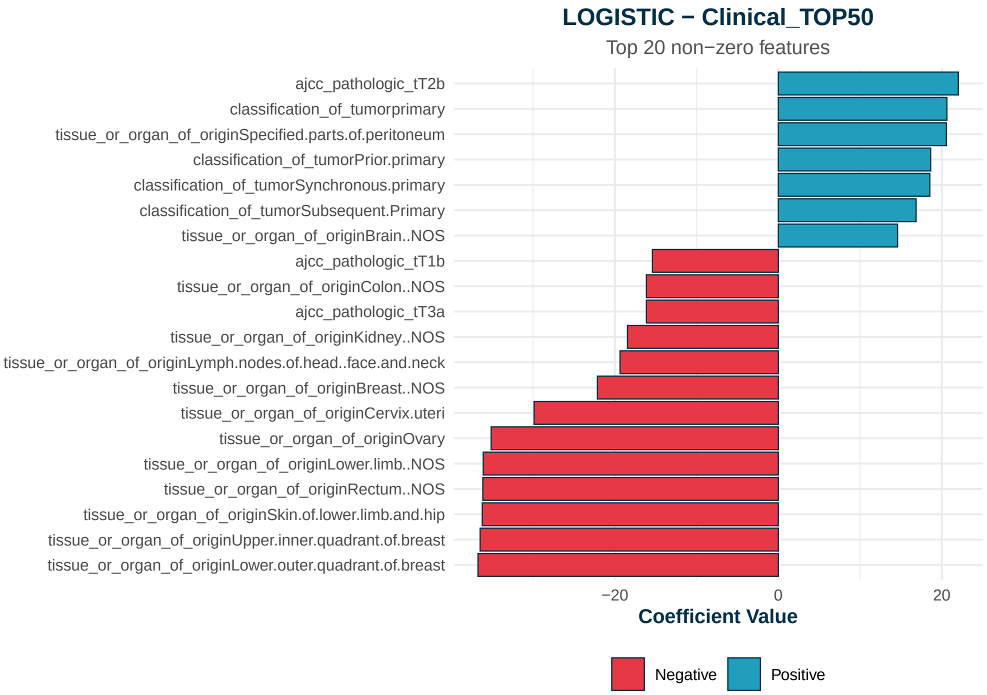

LOGISTIC − Performance Across Feature Sets
Test AUC
0.00
0.25
0.50
0.75
0.896
0.864
0.906
0.887
Clinical_Only
Clinical_TOP100
Clinical_TOP20
Clinical_TOP50
Feature Set

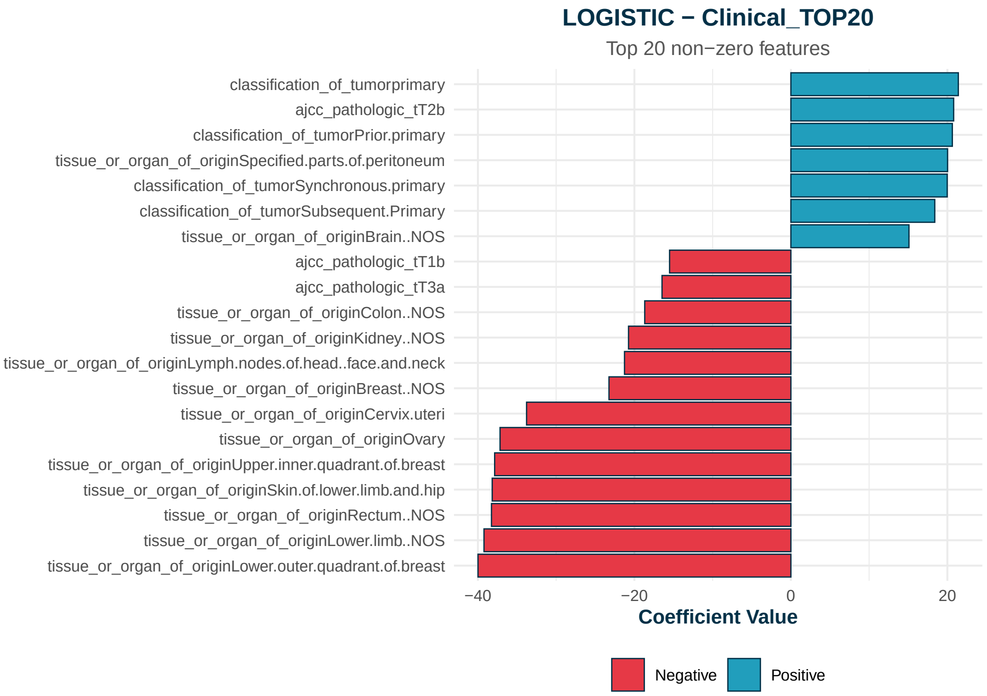

LOGISTIC − Selected Features
Number of Features
0
50
100
150
57
157
77
107
Clinical_Only
Clinical_TOP100
Clinical_TOP20
Clinical_TOP50
Feature Set


LOGISTIC − Train vs Test AUC
Clinical_TOP20
:linical_Only
Clinical_TOP50
Clinical_TOP1
0.90
0.89
0.88
0.87
Test AUC
0.92
0.93
0.94
0.95
0.96
0.97
Train AUC


LOGISTIC − Clinical_Only
Top 20 non−zero features
ajcc_pathologic_tT2b
classification_of_tumorprimary
classification_of_tumorPrior.primary
tissue_or_organ_of_originSpecified.parts.of.peritoneum
classification_of_tumorSynchronous.primary
classification_of_tumorSubsequent.Primary
ajcc_pathologic_tTis..LCIS.
ajcc_pathologic_tT3a
ajcc_pathologic_tT1b
tissue_or_organ_of_originColon..NOS
tissue_or_organ_of_originKidney..NOS
tissue_or_organ_of_originLymph.nodes.of.head..face.and.neck
tissue_or_organ_of_originBreast..NOS
tissue_or_organ_of_originCervix.uteri
tissue_or_organ_of_originOvary
tissue_or_organ_of_originLower.limb..NOS
tissue_or_organ_of_originUpper.inner.quadrant.of.breast
tissue_or_organ_of_originRectum..NOS
tissue_or_organ_of_originSkin.of.lower.limb.and.hip
tissue_or_organ_of_originLower.outer.quadrant.of.breast
−40
−20
0
20
Coefficient Value
Negative
Positive

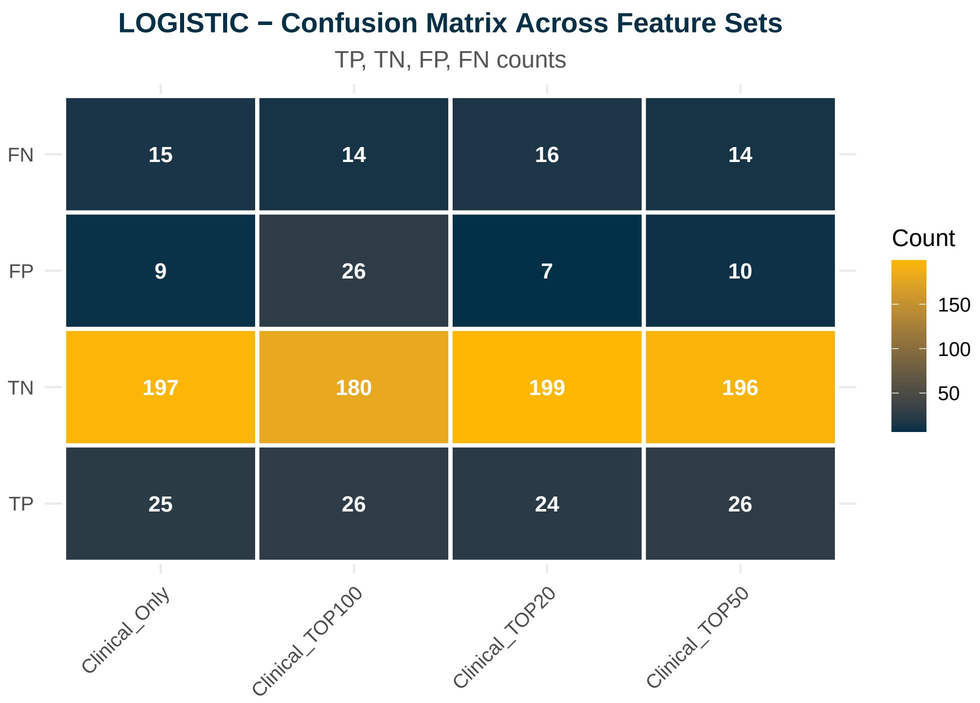

LOGISTIC − Clinical_TOP100
Top 20 non−zero features
ajcc_pathologic_tT2b
classification_of_tumorprimary
tissue_or_organ_of_originSpecified.parts.of.peritoneum
classification_of_tumorSynchronous.primary
classification_of_tumorPrior.primary
classification_of_tumorSubsequent.Primary
tissue_or_organ_of_originThorax..NOS
ajcc_pathologic_tT1b
ajcc_pathologic_tT3a
tissue_or_organ_of_originColon..NOS
tissue_or_organ_of_originKidney..NOS
tissue_or_organ_of_originLymph.nodes.of.head..face.and.neck
tissue_or_organ_of_originBreast..NOS
tissue_or_organ_of_originCervix.uteri
tissue_or_organ_of_originOvary
tissue_or_organ_of_originSkin.of.lower.limb.and.hip
tissue_or_organ_of_originUpper.inner.quadrant.of.breast
tissue_or_organ_of_originRectum..NOS
tissue_or_organ_of_originLower.limb..NOS
tissue_or_organ_of_originLower.outer.quadrant.of.breast
−40
−20
0
20
Coefficient Value
Negative
Positive


LOGISTIC − Clinical_TOP50
Top 20 non−zero features
ajcc_pathologic_tT2b
classification_of_tumorprimary
tissue_or_organ_of_originSpecified.parts.of.peritoneum
classification_of_tumorPrior.primary
classification_of_tumorSynchronous.primary
classification_of_tumorSubsequent.Primary
tissue_or_organ_of_originBrain..NOS
ajcc_pathologic_tT1b
tissue_or_organ_of_originColon..NOS
ajcc_pathologic_tT3a
tissue_or_organ_of_originKidney..NOS
tissue_or_organ_of_originLymph.nodes.of.head..face.and.neck
tissue_or_organ_of_originBreast..NOS
tissue_or_organ_of_originCervix.uteri
tissue_or_organ_of_originOvary
tissue_or_organ_of_originLower.limb..NOS
tissue_or_organ_of_originRectum..NOS
tissue_or_organ_of_originSkin.of.lower.limb.and.hip
tissue_or_organ_of_originUpper.inner.quadrant.of.breast
tissue_or_organ_of_originLower.outer.quadrant.of.breast
−20
0
20
Coefficient Value
Negative
Positive

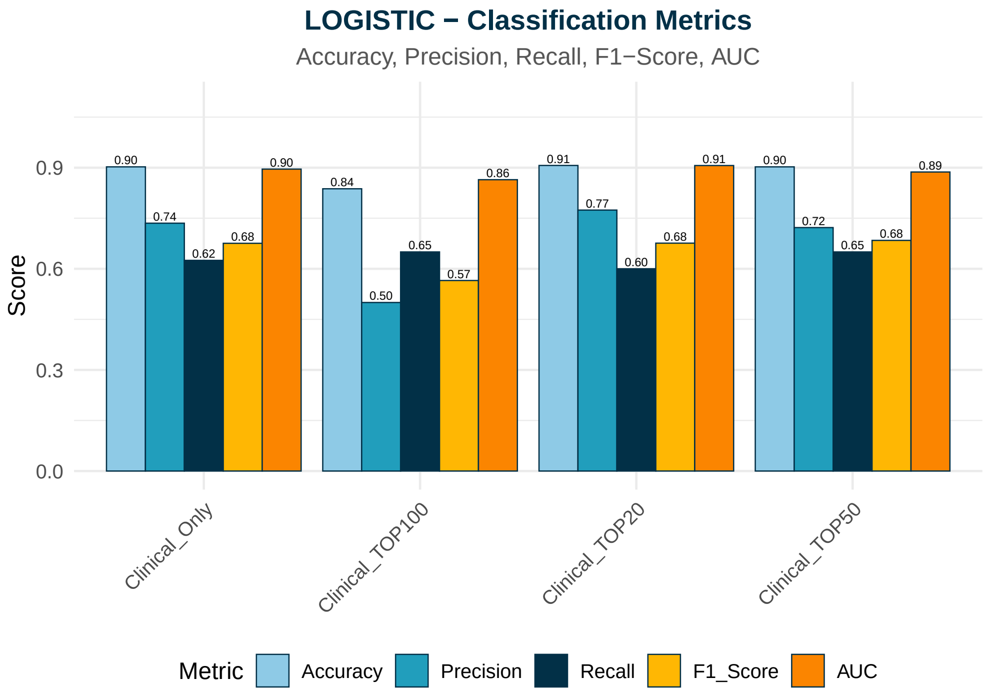


```
logistic_metrics <- plot_classification_metrics_single(logistic_results
                                                        , threshold = 0.5
                                                        , csv_filename = "logistic_classification_metrics
```

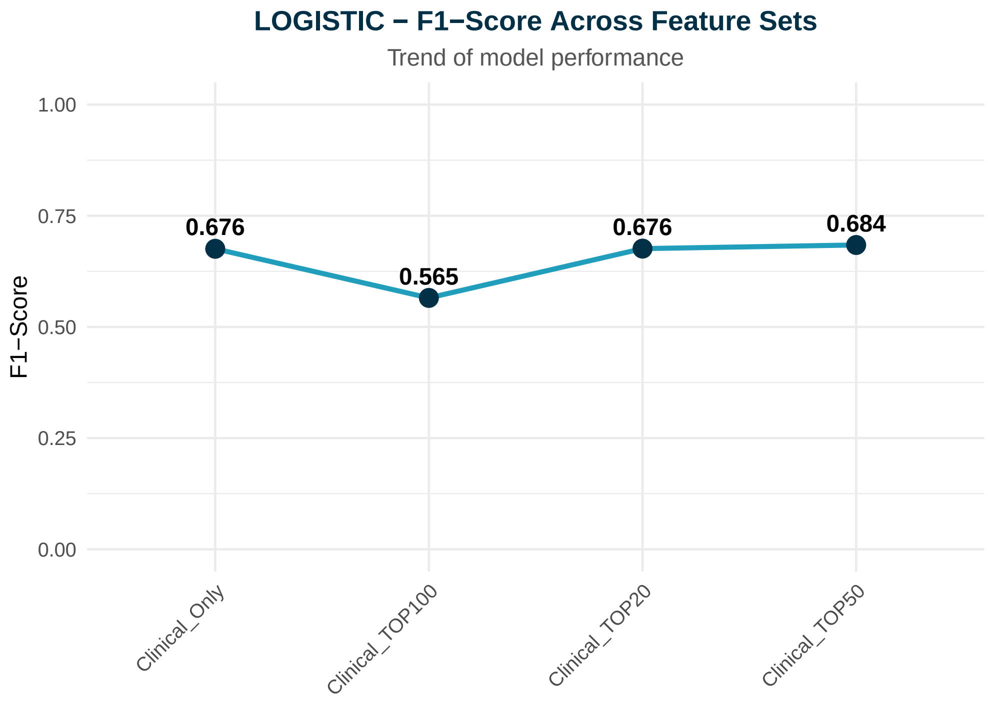
```
##
## === CLASSIFICATION METRICS ===

## Clinical_Only:
##   TP=25 TN=197 FP=9 FN=15
##   Accuracy=0.902 Precision=0.735 Recall=0.625 F1=0.676 AUC=0.896

## Clinical_TOP100:
##   TP=26 TN=180 FP=26 FN=14
##   Accuracy=0.837 Precision=0.500 Recall=0.650 F1=0.565 AUC=0.864

## Clinical_TOP50:
##   TP=26 TN=196 FP=10 FN=14
##   Accuracy=0.902 Precision=0.722 Recall=0.650 F1=0.684 AUC=0.887

## Clinical_TOP20:
##   TP=24 TN=199 FP=7 FN=16
##   Accuracy=0.907 Precision=0.774 Recall=0.600 F1=0.676 AUC=0.906
```

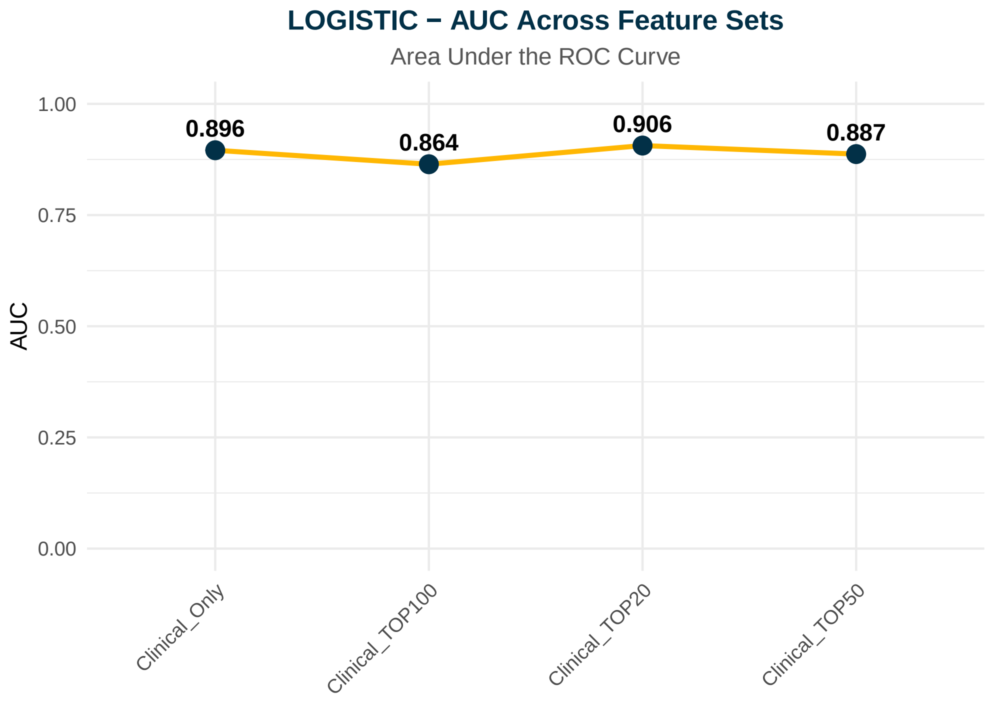

LOGISTIC − Confusion Matrix Across Feature Sets

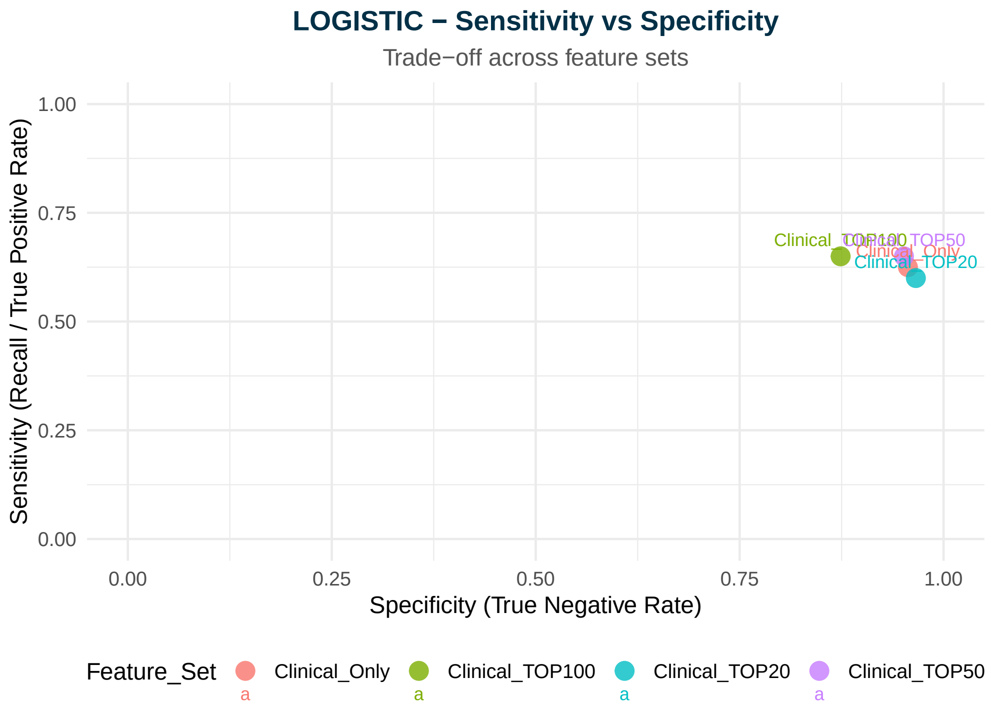
TP, TN, FP, FN counts


LOGISTIC − Classification Metrics

Accuracy, Precision, Recall, F1−Score, AUC


LOGISTIC − F1−Score Across Feature Sets
Trend of model performance
F1−Score
1.00
0.75
0.50
0.25
0.00
0.676
0.565
0.676
0.684
Clinical_Only
Clinical_TOP100
Clinical_TOP20
Clinical_TOP50


LOGISTIC − AUC Across Feature Sets
Area Under the ROC Curve
AUC
1.00
0.75
0.50
0.25
0.00
0.896
0.864
0.906
0.887
Clinical_Only
Clinical_TOP100
Clinical_TOP20
Clinical_TOP50



```
##
## === SUMMARY TABLE ===
##       Feature_Set TP  TN FP FN  Accuracy Precision Recall Specificity  F1_Score
## 1   Clinical_Only 25 197  9 15 0.9024390 0.7352941  0.625   0.9563107 0.6756757
## 2 Clinical_TOP100 26 180 26 14 0.8373984 0.5000000  0.650   0.8737864 0.5652174
## 3  Clinical_TOP50 26 196 10 14 0.9024390 0.7222222  0.650   0.9514563 0.6842105
## 4  Clinical_TOP20 24 199  7 16 0.9065041 0.7741935  0.600   0.9660194 0.6760563
##         AUC
## 1 0.8957524
## 2 0.8643204
## 3 0.8871359
## 4 0.9063107
##
## Exported classification metrics to: model_metrics/logistic_classification_metrics.csv
```

Logistic regression shows consistently high specificity but low recall, indicating that it classifies Alive patients reliably but struggles to detect Dead cases. The clinical-only model performs best overall, while adding genomic features does not meaningfully improve recall and often reduces generalization, especially with 100 genes. These metrics reinforce the conclusion that logistic regression cannot effectively exploit high-dimensional gene expression and is best used as a baseline on small feature sets.

Ridge Regression across different feature

Ridge regression stabilizes estimation in the presence of strong correlations between genes, but does not perform variable selection.

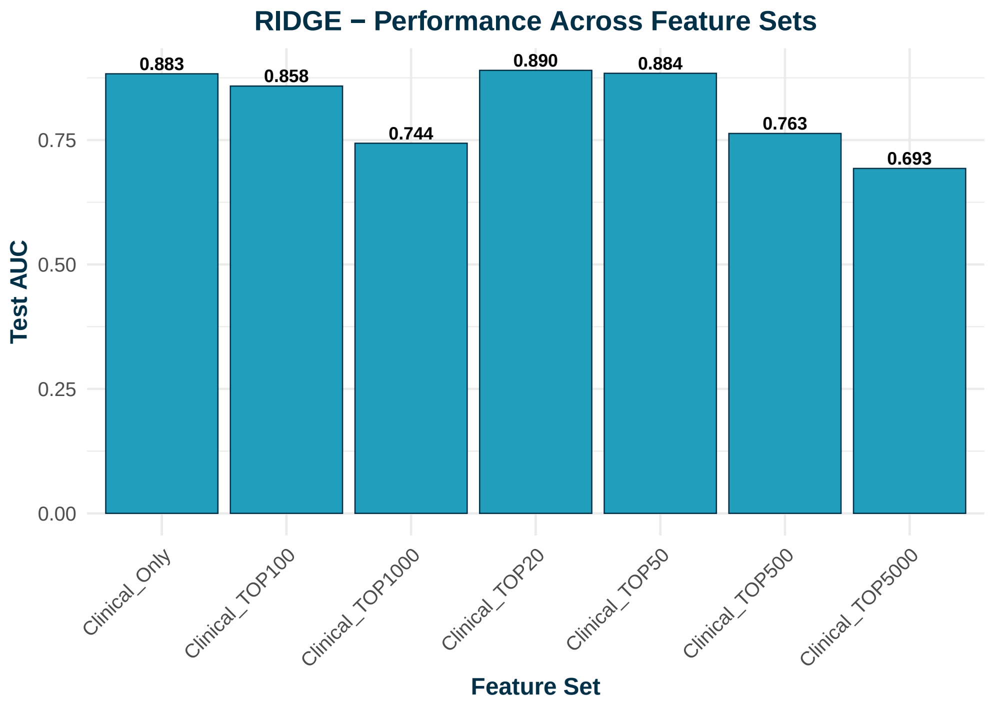
By adding an L2 penalty,

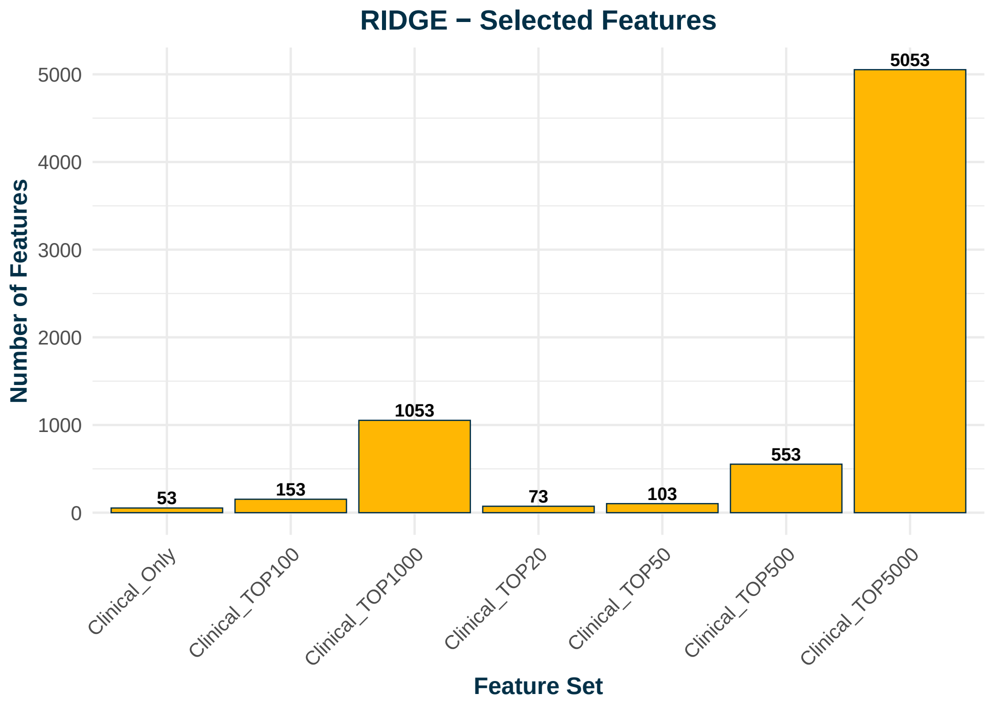
$$\hat{\beta}^{\text{ridge}} = argmin_{\beta}\{-l(\beta) + \lambda||\beta||_2^2\}$$

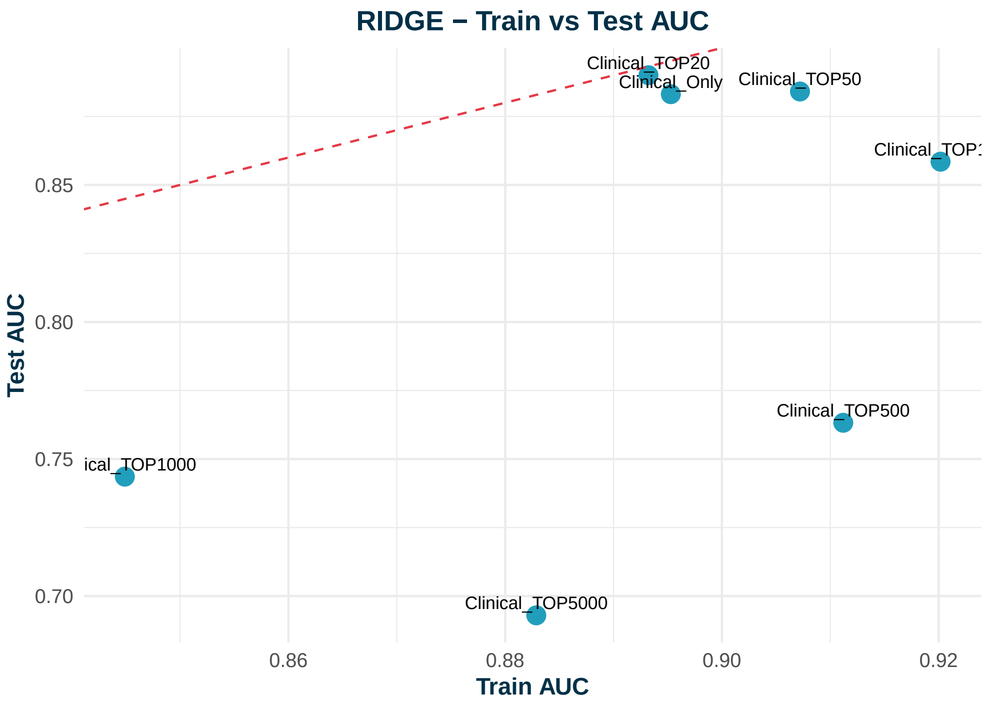
the model remains stable even when thousands of genes are included. Therefore, Ridge can handle large feature sets, and we apply it on 5000, 1000, 500, 100, 50, and 20 top genes to evaluate its performance at different dimensionalities.

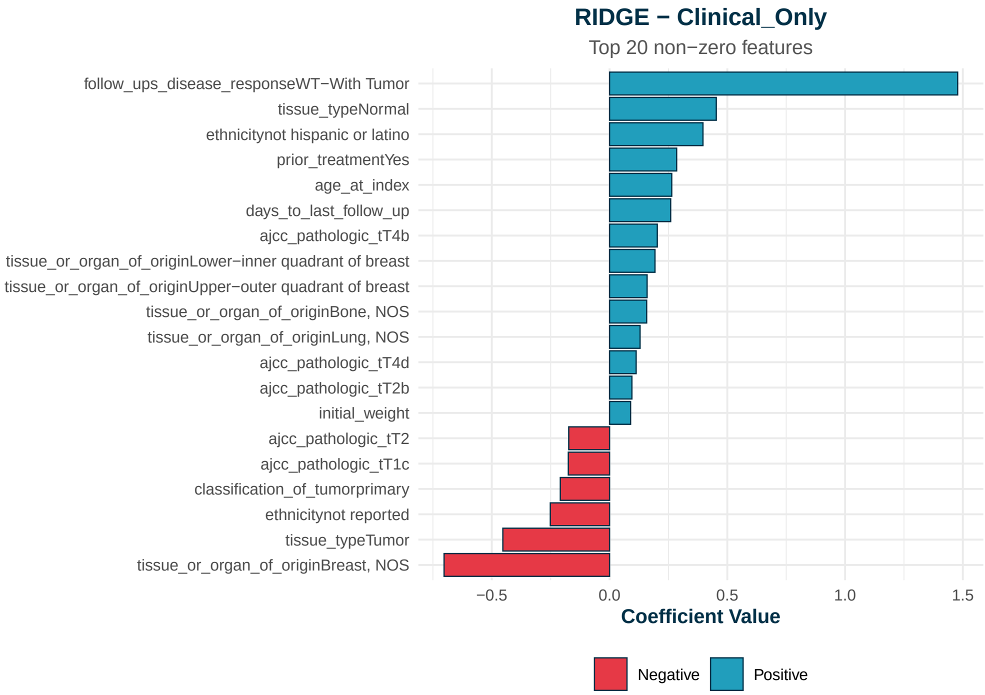
```r
ridge_results <- fit_single_model_across_features(
    model_type = "ridge"
    , X_train_all = X_train
    , X_test_all = X_test
    , Y_train = Y_train
    , Y_test = Y_test
    , n_clinical = n_clinical
    , top_genes_ranked = top_genes
    , gene_sets = c(5000, 1000, 500, 100, 50, 20)
)
```

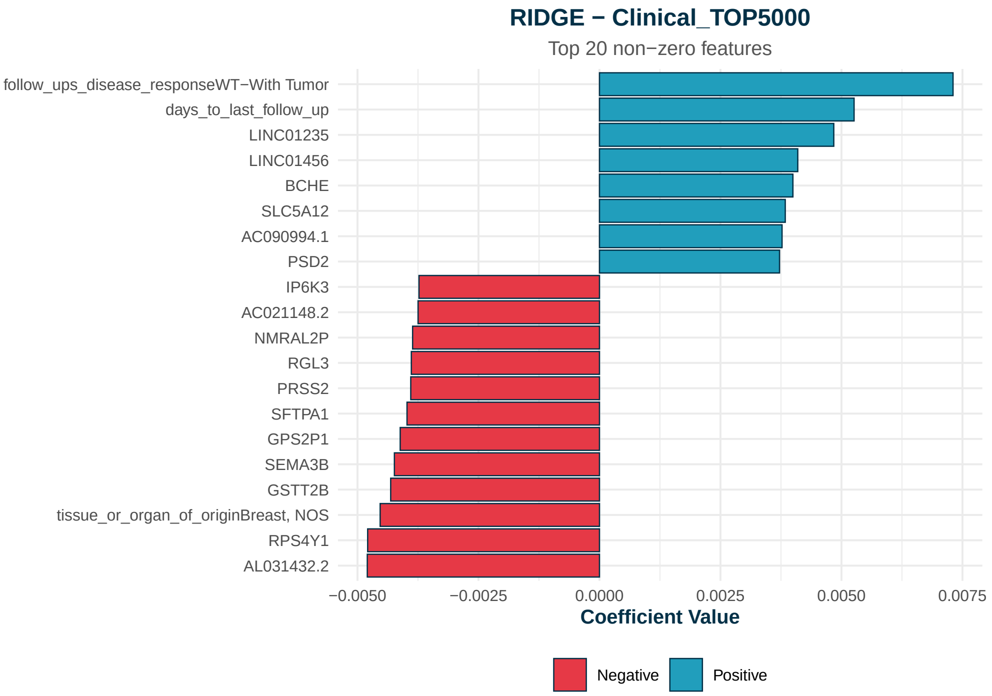
```
##
## === FITTING RIDGE ACROSS FEATURE SETS ===
##
## Fitting Clinical_Only...

## Setting levels: control = 0, case = 1

## Setting direction: controls < cases

## Setting levels: control = 0, case = 1

## Setting direction: controls < cases

## Fitting Clinical_TOP5000...

## Setting levels: control = 0, case = 1
## Setting direction: controls < cases

## Setting levels: control = 0, case = 1

## Setting direction: controls < cases

## Fitting Clinical_TOP1000...

## Setting levels: control = 0, case = 1
## Setting direction: controls < cases

## Setting levels: control = 0, case = 1

## Setting direction: controls < cases

## Fitting Clinical_TOP500...

## Setting levels: control = 0, case = 1
## Setting direction: controls < cases
## Setting levels: control = 0, case = 1

## Setting direction: controls < cases

## Fitting Clinical_TOP100...

## Setting levels: control = 0, case = 1
## Setting direction: controls < cases

## Setting levels: control = 0, case = 1

## Setting direction: controls < cases

## Fitting Clinical_TOP50...

## Setting levels: control = 0, case = 1
## Setting direction: controls < cases

## Setting levels: control = 0, case = 1

## Setting direction: controls < cases

## Fitting Clinical_TOP20...

## Setting levels: control = 0, case = 1
## Setting direction: controls < cases

## Setting levels: control = 0, case = 1

## Setting direction: controls < cases

##
## === SUMMARY TABLE ===
##        Feature_Set Model Features Train_AUC  Test_AUC Test_Accuracy
## 1    Clinical_Only RIDGE       53 0.8952854 0.8831311     0.8577236
## 2 Clinical_TOP5000 RIDGE     5053 0.8828706 0.6929612     0.8373984
## 3 Clinical_TOP1000 RIDGE     1053 0.8449016 0.7435680     0.8373984
## 4  Clinical_TOP500 RIDGE      553 0.9111945 0.7632282     0.8536585
## 5  Clinical_TOP100 RIDGE      153 0.9201754 0.8584951     0.8861789
## 6   Clinical_TOP50 RIDGE      103 0.9072021 0.8841019     0.8902439
## 7   Clinical_TOP20 RIDGE       73 0.8932175 0.8900485     0.8943089
## Exported metrics to: model_metrics/ridge_across_features_metrics.csv
```

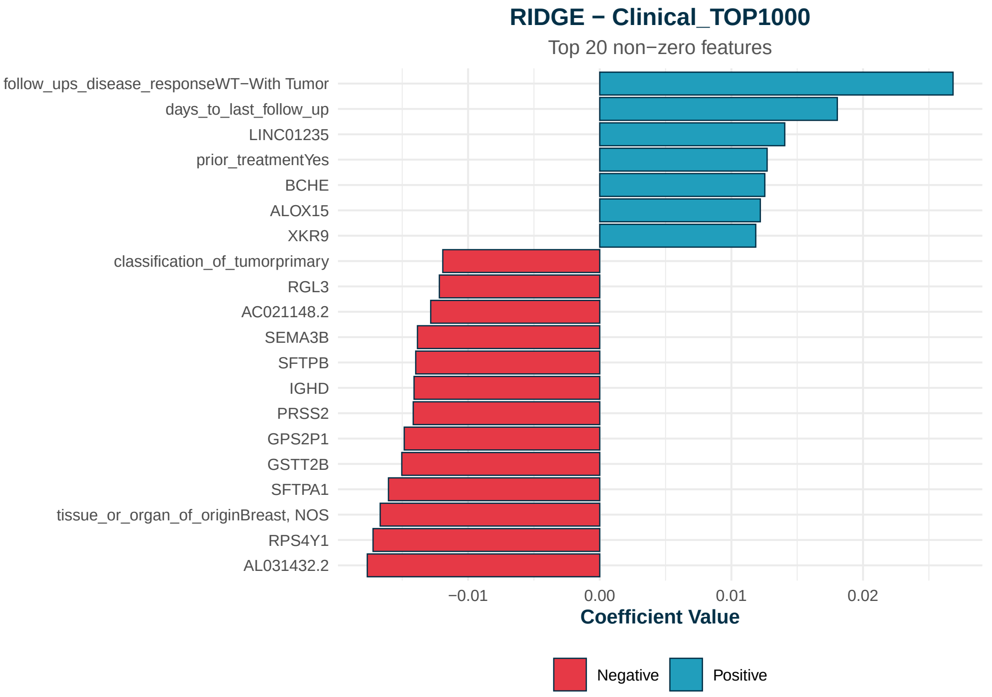

RIDGE − Performance Across Feature Sets
Test AUC
0.00
0.25
0.50
0.75
0.883
0.858
0.744
0.890
0.884
0.763
0.693
Clinical_Only
Clinical_TOP100
Clinical_TOP1000
Clinical_TOP20
Clinical_TOP50
Clinical_TOP500
Clinical_TOP5000
Feature Set

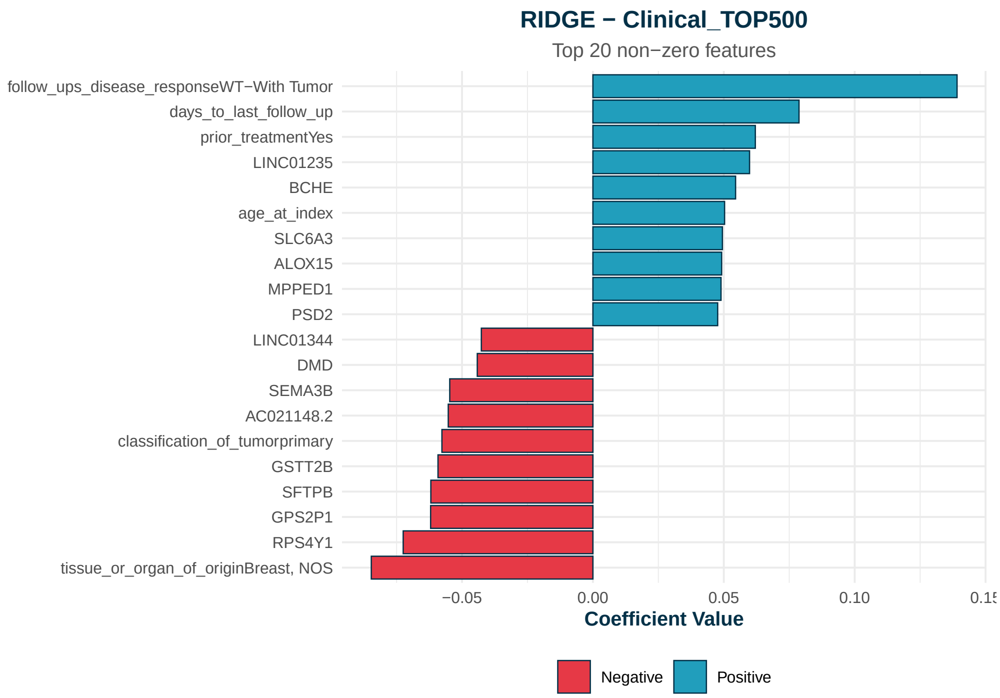

RIDGE − Selected Features
Number of Features
0
1000
2000
3000
4000
5000
53
153
1053
73
103
553
5053
Clinical_Only
Clinical_TOP100
Clinical_TOP1000
Clinical_TOP20
Clinical_TOP50
Clinical_TOP500
Clinical_TOP5000
Feature Set

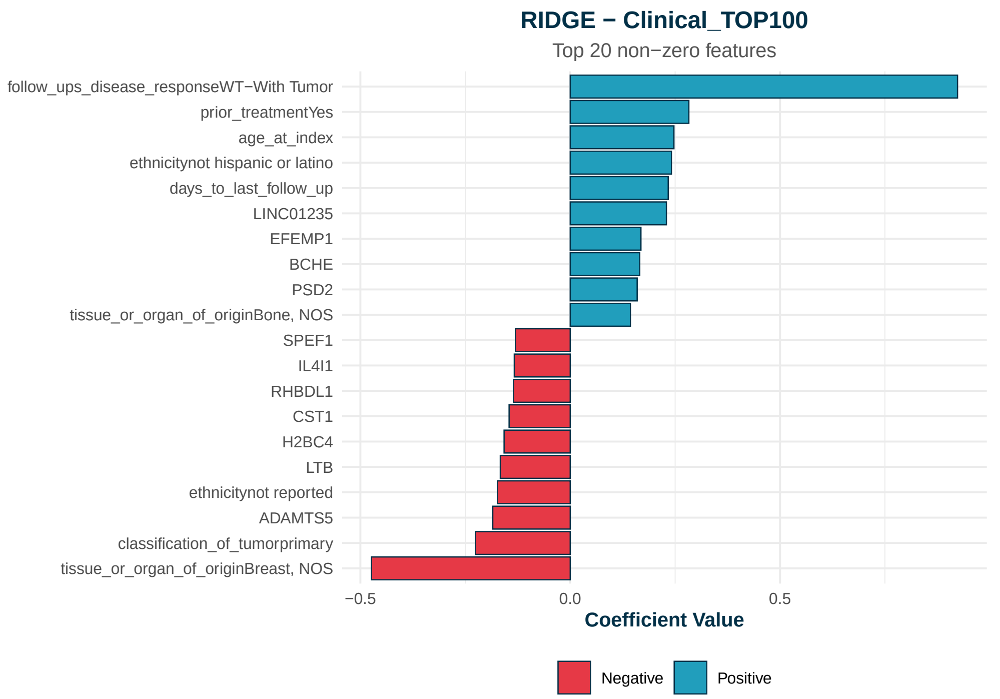

RIDGE − Train vs Test AUC
Test AUC
Train AUC
0.70
0.75
0.80
0.85
0.86
0.88
0.90
0.92
Clinical_TOP20
Clinical_Only
Clinical_TOP50
Clinical_TOP1
Clinical_TOP500
ical_TOP1000
Clinical_TOP5000

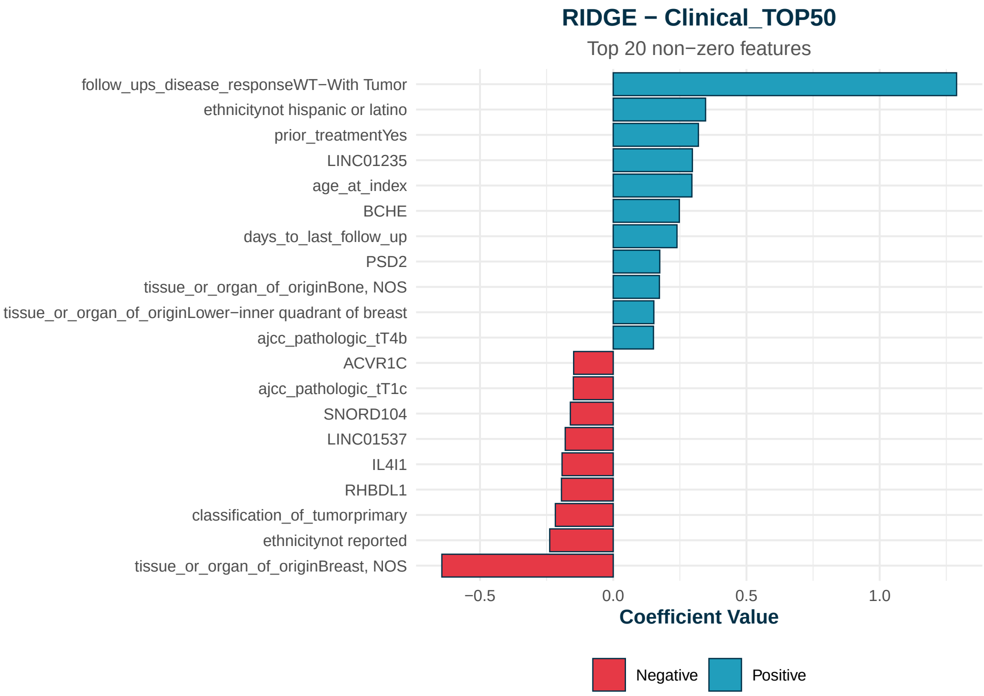

RIDGE − Clinical_Only
Top 20 non−zero features
follow_ups_disease_responseWT−With Tumor
tissue_typeNormal
ethnicitynot hispanic or latino
prior_treatmentYes
age_at_index
days_to_last_follow_up
ajcc_pathologic_tT4b
tissue_or_organ_of_originLower−inner quadrant of breast
tissue_or_organ_of_originUpper−outer quadrant of breast
tissue_or_organ_of_originBone, NOS
tissue_or_organ_of_originLung, NOS
ajcc_pathologic_tT4d
ajcc_pathologic_tT2b
initial_weight
ajcc_pathologic_tT2
ajcc_pathologic_tT1c
classification_of_tumorprimary
ethnicitynot reported
tissue_typeTumor
tissue_or_organ_of_originBreast, NOS
−0.5
0.0
0.5
1.0
1.5
Coefficient Value
🟥 Negative
🟦 Positive

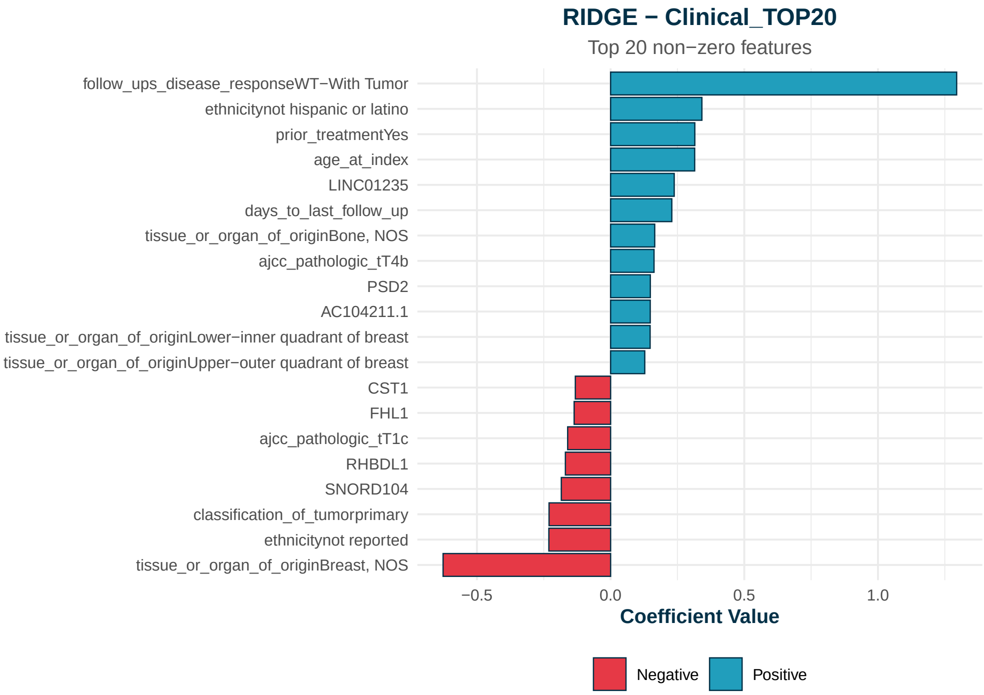

RIDGE − Clinical_TOP5000
Top 20 non−zero features
follow_ups_disease_responseWT−With Tumor
days_to_last_follow_up
LINC01235
LINC01456
BCHE
SLC5A12
AC090994.1
PSD2
IP6K3
AC021148.2
NMRAL2P
RGL3
PRSS2
SFTPA1
GPS2P1
SEMA3B
GSTT2B
tissue_or_organ_of_originBreast, NOS
RPS4Y1
AL031432.2
−0.0050
−0.0025
0.0000
0.0025
0.0050
0.0075
Coefficient Value
Negative
Positive


RIDGE − Clinical_TOP1000
Top 20 non−zero features
follow_ups_disease_responseWT−With Tumor
days_to_last_follow_up
LINC01235
prior_treatmentYes
BCHE
ALOX15
XKR9
classification_of_tumorprimary
RGL3
AC021148.2
SEMA3B
SFTPB
IGHD
PRSS2
GPS2P1
GSTT2B
SFTPA1
tissue_or_organ_of_originBreast, NOS
RPS4Y1
AL031432.2
−0.01
0.00
0.01
0.02
Coefficient Value
Negative
Positive


RIDGE − Clinical_TOP500
Top 20 non−zero features
follow_ups_disease_responseWT−With Tumor
days_to_last_follow_up
prior_treatmentYes
LINC01235
BCHE
age_at_index
SLC6A3
ALOX15
MPPED1
PSD2
LINC01344
DMD
SEMA3B
AC021148.2
classification_of_tumorprimary
GSTT2B
SFTPB
GPS2P1
RPS4Y1
tissue_or_organ_of_originBreast, NOS
−0.05
0.00
0.05
0.10
0.15
Coefficient Value
🟥 Negative
🟦 Positive

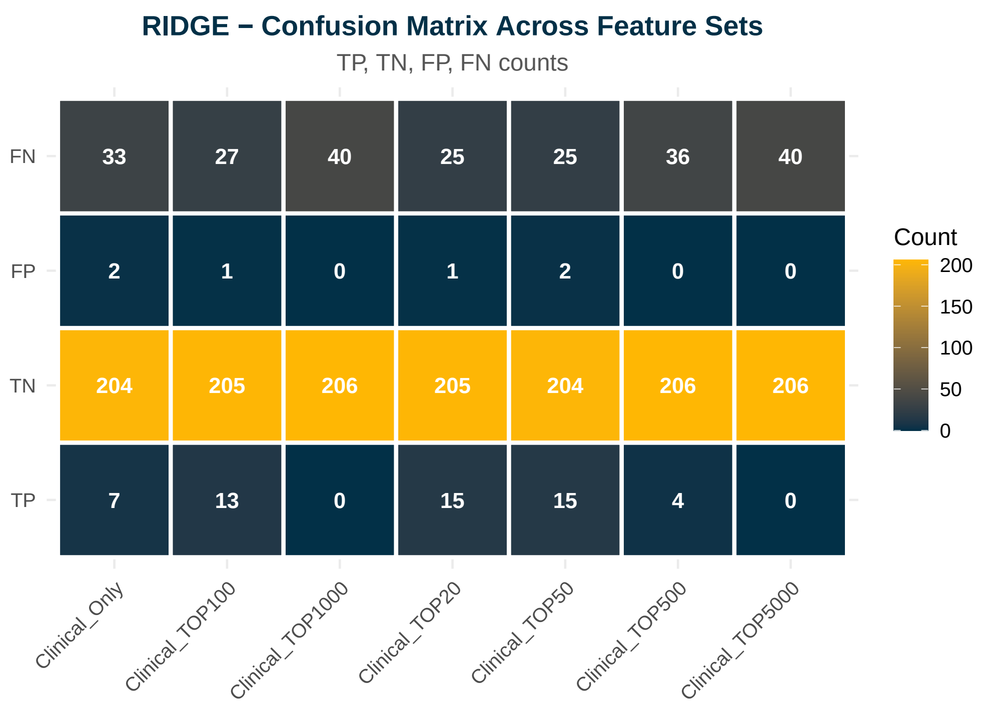

RIDGE − Clinical_TOP100
Top 20 non−zero features
follow_ups_disease_responseWT−With Tumor
prior_treatmentYes
age_at_index
ethnicitynot hispanic or latino
days_to_last_follow_up
LINC01235
EFEMP1
BCHE
PSD2
tissue_or_organ_of_originBone, NOS
SPEF1
IL4I1
RHBDL1
CST1
H2BC4
LTB
ethnicitynot reported
ADAMTS5
classification_of_tumorprimary
tissue_or_organ_of_originBreast, NOS
−0.5
0.0
0.5
Coefficient Value
🟥 Negative
🟦 Positive



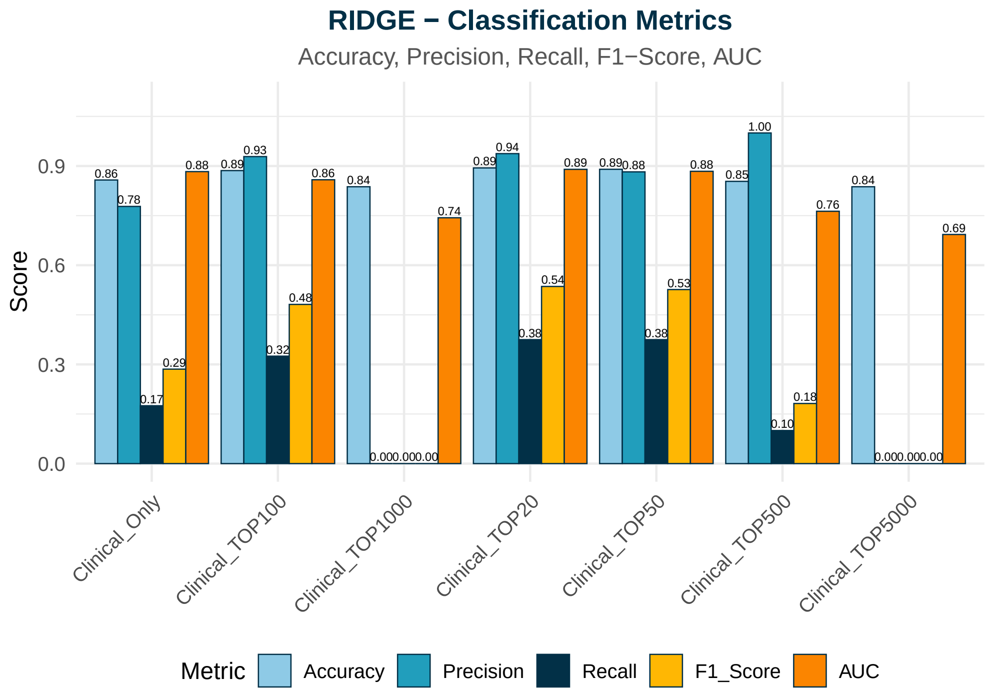


```
ridge_metrics <- plot_classification_metrics_single(ridge_results
                                                , threshold = 0.5
                                                , csv_filename = "ridge_classification_metrics.csv"
```

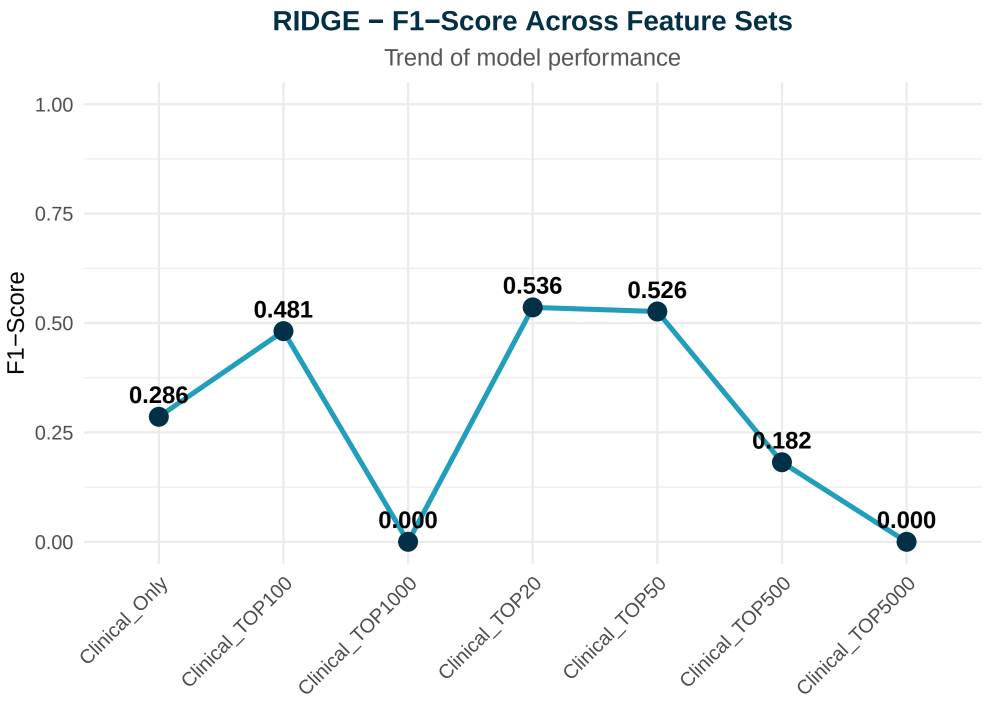
```
##
## === CLASSIFICATION METRICS ===
## Setting levels: control = 0, case = 1
## Setting direction: controls < cases

## Clinical_Only:
##   TP=7 TN=204 FP=2 FN=33
##   Accuracy=0.858 Precision=0.778 Recall=0.175 F1=0.286 AUC=0.883

## Setting levels: control = 0, case = 1
## Setting direction: controls < cases

## Clinical_TOP5000:
##   TP=0 TN=206 FP=0 FN=40
##   Accuracy=0.837 Precision=0.000 Recall=0.000 F1=0.000 AUC=0.693

## Setting levels: control = 0, case = 1
## Setting direction: controls < cases

## Clinical_TOP1000:
##   TP=0 TN=206 FP=0 FN=40
##   Accuracy=0.837 Precision=0.000 Recall=0.000 F1=0.000 AUC=0.744

## Setting levels: control = 0, case = 1
## Setting direction: controls < cases

## Clinical_TOP500:
##   TP=4 TN=206 FP=0 FN=36
##   Accuracy=0.854 Precision=1.000 Recall=0.100 F1=0.182 AUC=0.763

## Setting levels: control = 0, case = 1
## Setting direction: controls < cases

## Clinical_TOP100:
##   TP=13 TN=205 FP=1 FN=27
##   Accuracy=0.886 Precision=0.929 Recall=0.325 F1=0.481 AUC=0.858

## Setting levels: control = 0, case = 1
## Setting direction: controls < cases

## Clinical_TOP50:
##   TP=15 TN=204 FP=2 FN=25
##   Accuracy=0.890 Precision=0.882 Recall=0.375 F1=0.526 AUC=0.884

## Setting levels: control = 0, case = 1
## Setting direction: controls < cases

## Clinical_TOP20:
##   TP=15 TN=205 FP=1 FN=25
##   Accuracy=0.894 Precision=0.938 Recall=0.375 F1=0.536 AUC=0.890
```

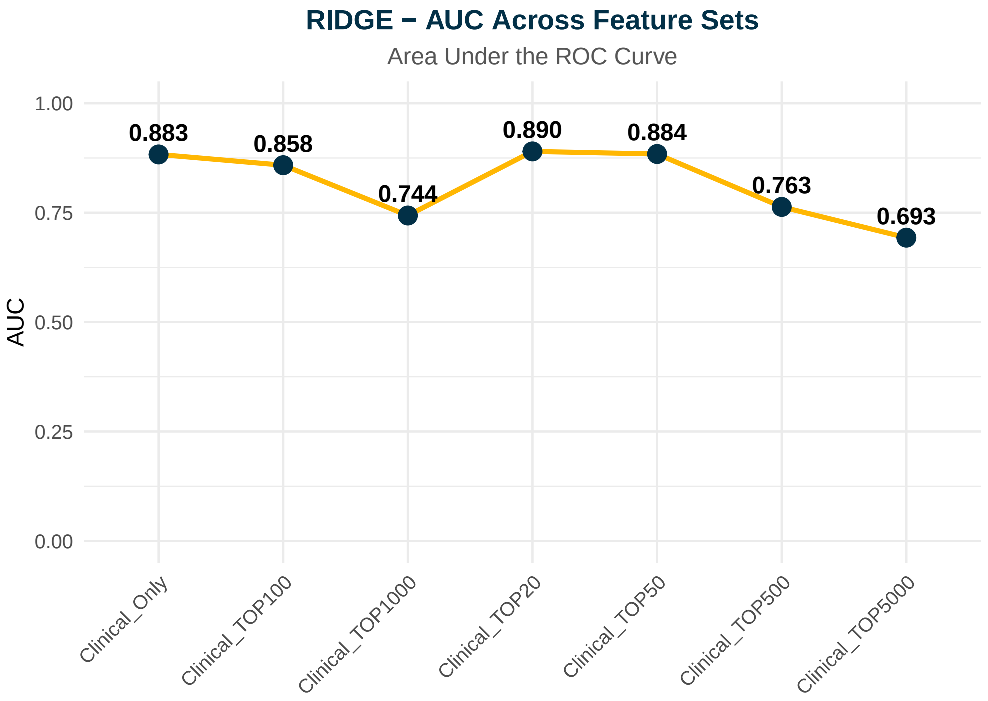

RIDGE − Confusion Matrix Across Feature Sets

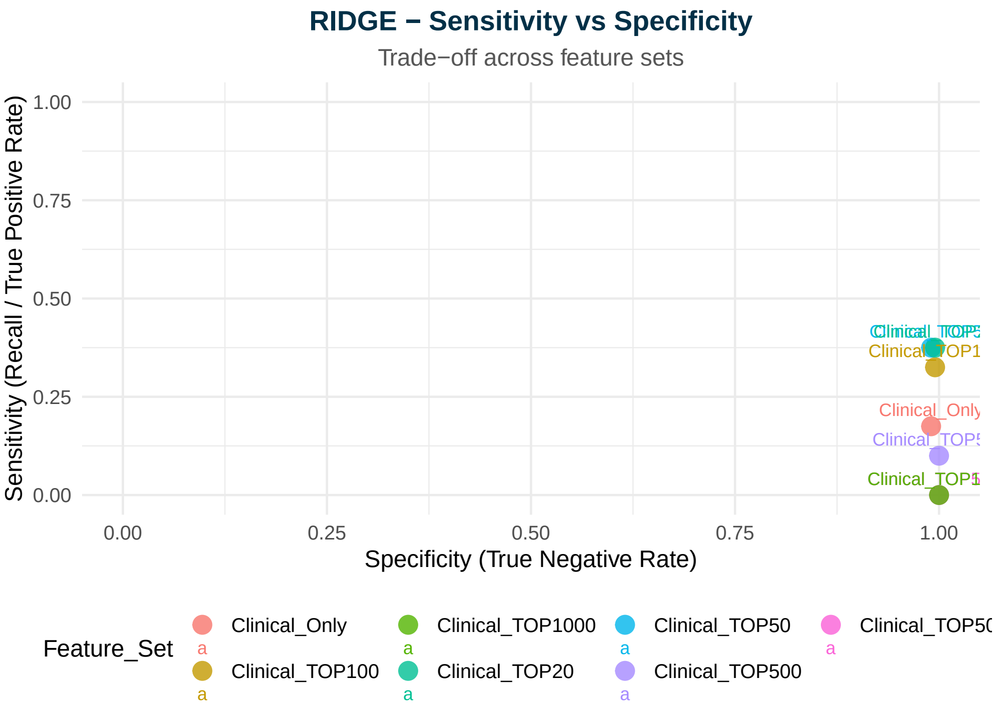
TP, TN, FP, FN counts


RIDGE − Classification Metrics

Accuracy, Precision, Recall, F1−Score, AUC


RIDGE − F1−Score Across Feature Sets
Trend of model performance
F1−Score
1.00
0.75
0.50
0.25
0.00
0.286
0.481
0.000
0.536
0.526
0.182
0.000
Clinical_Only
Clinical_TOP100
Clinical_TOP1000
Clinical_TOP20
Clinical_TOP50
Clinical_TOP500
Clinical_TOP5000


RIDGE − AUC Across Feature Sets
Area Under the ROC Curve
AUC
1.00
0.75
0.50
0.25
0.00
0.883
0.858
0.744
0.890
0.884
0.763
0.693
Clinical_Only
Clinical_TOP100
Clinical_TOP1000
Clinical_TOP20
Clinical_TOP50
Clinical_TOP500
Clinical_TOP5000



```
##
## === SUMMARY TABLE ===
##        Feature_Set TP  TN FP FN  Accuracy Precision Recall Specificity
## 1    Clinical_Only  7 204  2 33 0.8577236 0.7777778  0.175   0.9902913
## 2 Clinical_TOP5000  0 206  0 40 0.8373984 0.0000000  0.000   1.0000000
## 3 Clinical_TOP1000  0 206  0 40 0.8373984 0.0000000  0.000   1.0000000
## 4  Clinical_TOP500  4 206  0 36 0.8536585 1.0000000  0.100   1.0000000
## 5  Clinical_TOP100 13 205  1 27 0.8861789 0.9285714  0.325   0.9951456
## 6   Clinical_TOP50 15 204  2 25 0.8902439 0.8823529  0.375   0.9902913
## 7   Clinical_TOP20 15 205  1 25 0.8943089 0.9375000  0.375   0.9951456
##    F1_Score       AUC
## 1 0.2857143 0.8831311
## 2 0.0000000 0.6929612
## 3 0.0000000 0.7435680
## 4 0.1818182 0.7632282
## 5 0.4814815 0.8584951
## 6 0.5263158 0.8841019
## 7 0.5357143 0.8900485
##
## Exported classification metrics to: model_metrics/ridge_classification_metrics.csv
```

Ridge regression shows very high specificity around 1.00 but consistently low recall, meaning it correctly identifies Alive patients but frequently misses Dead cases. For large gene sets (5000 and 1000 genes), Ridge collapses completely (Recall = 0.00, F1 = 0.00), predicting all patients as Alive due to excessive shrinkage. Performance improves for smaller gene sets (TOP20–TOP50), where recall reaches 0.25–0.27 and AUC improves to 0.86–0.88. The clinical-only model performs best overall with AUC = 0.892, but still low recall (0.2167). These results show that Ridge cannot effectively recover sparse signals in high-dimensional genomic data.

Lasso Regression Across Feature Sets

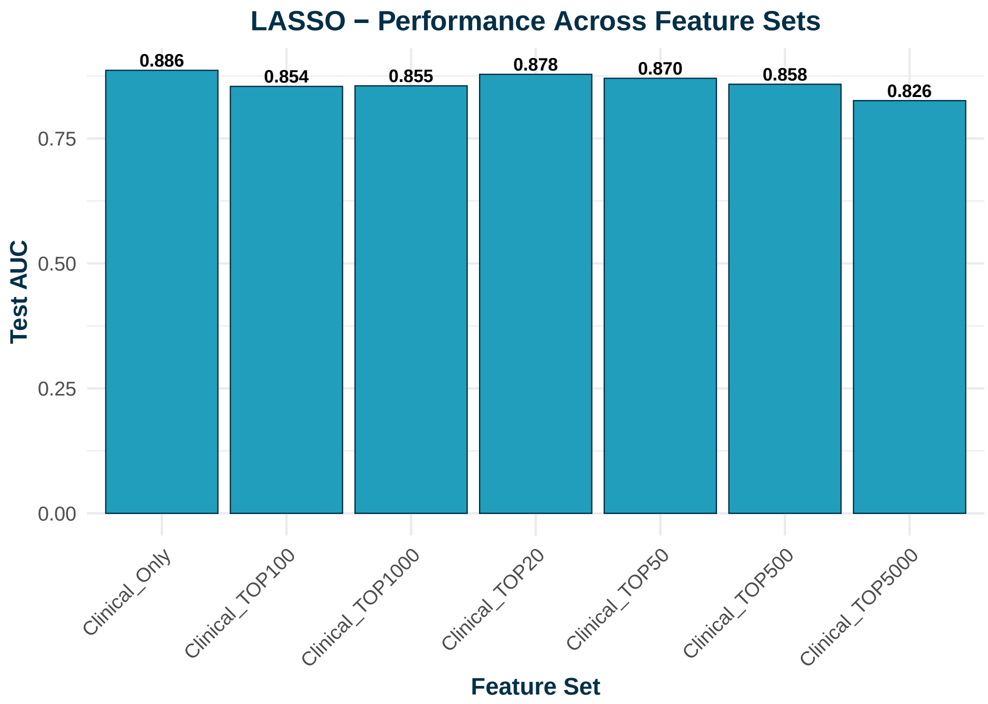
The Lasso induces sparsity in the solution by setting many coefficients exactly to zero. This is particularly well-suited for genomic data, where only a small subset of genes is expected to carry predictive information.

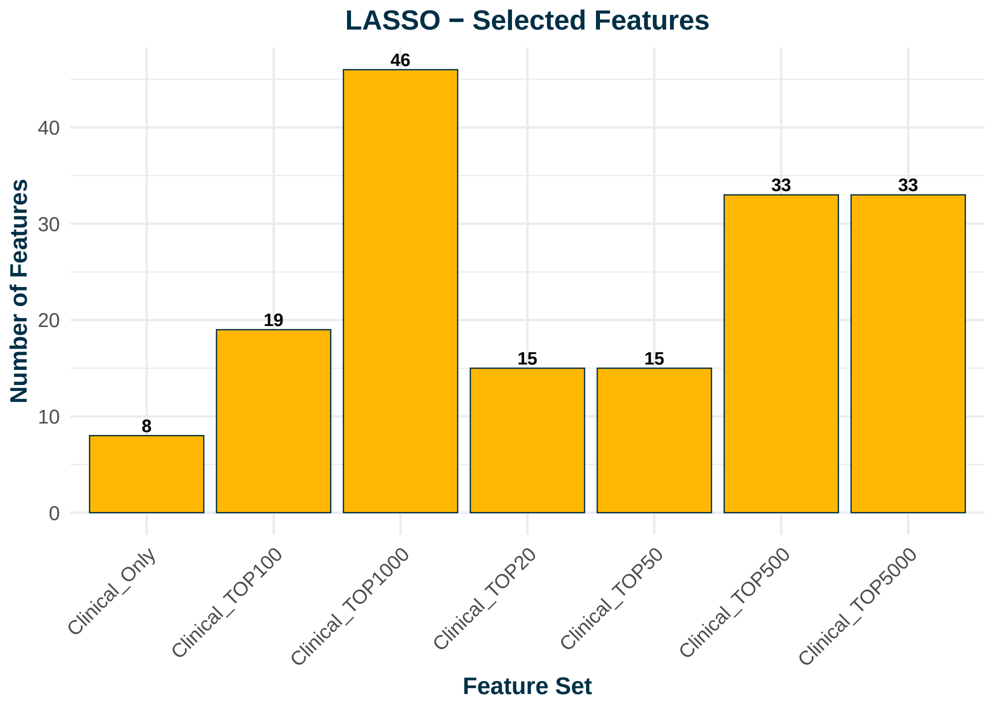
$$\hat{\beta}^{\text{lasso}} = argmin_{\beta}\{-l(\beta) + \lambda\|\beta\|_1\}$$

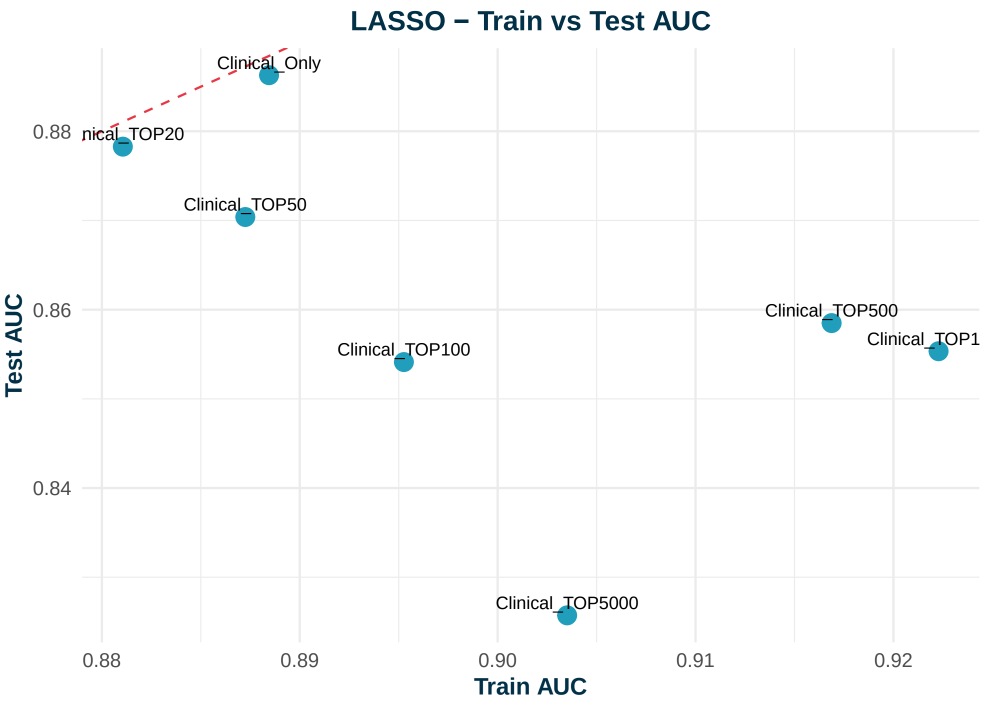
This makes Lasso suitable for genomic data, where only a small subset of genes is expected to be predictive. We apply Lasso to gene sets of increasing size (5000, 1000, 500, 100, 50, 20) to evaluate how sparsity improves stability and interpretability in high dimension.

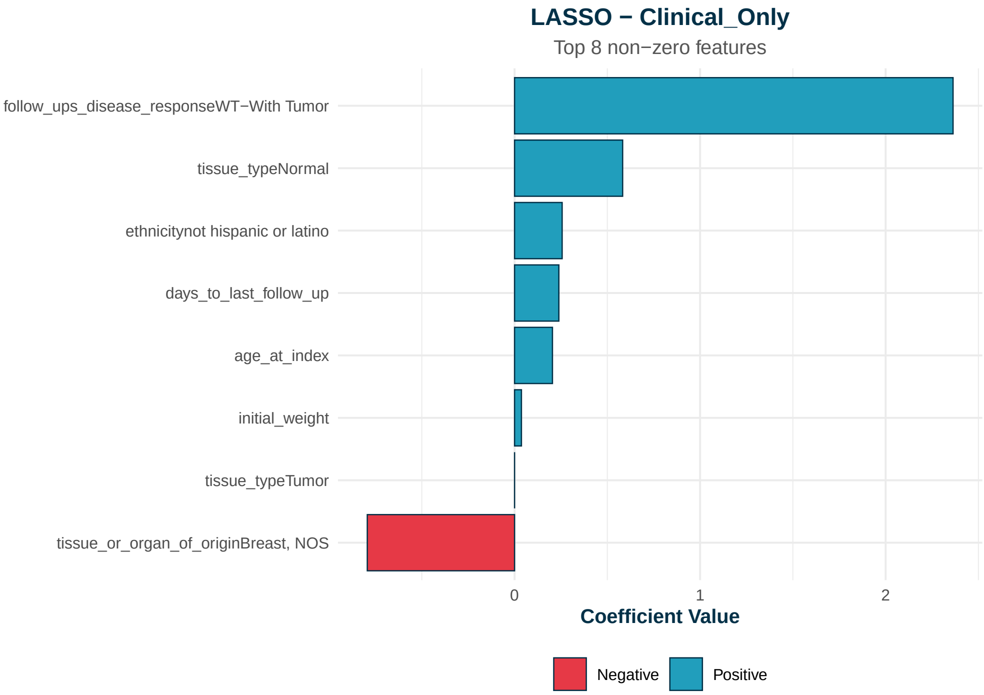
```r
lasso_results <- fit_single_model_across_features(
    model_type = "lasso"
    , X_train_all = X_train
    , X_test_all = X_test
    , Y_train = Y_train
    , Y_test = Y_test
    , n_clinical = n_clinical
    , top_genes_ranked = top_genes
    , gene_sets = c(5000, 1000, 500, 100, 50, 20)
)
```

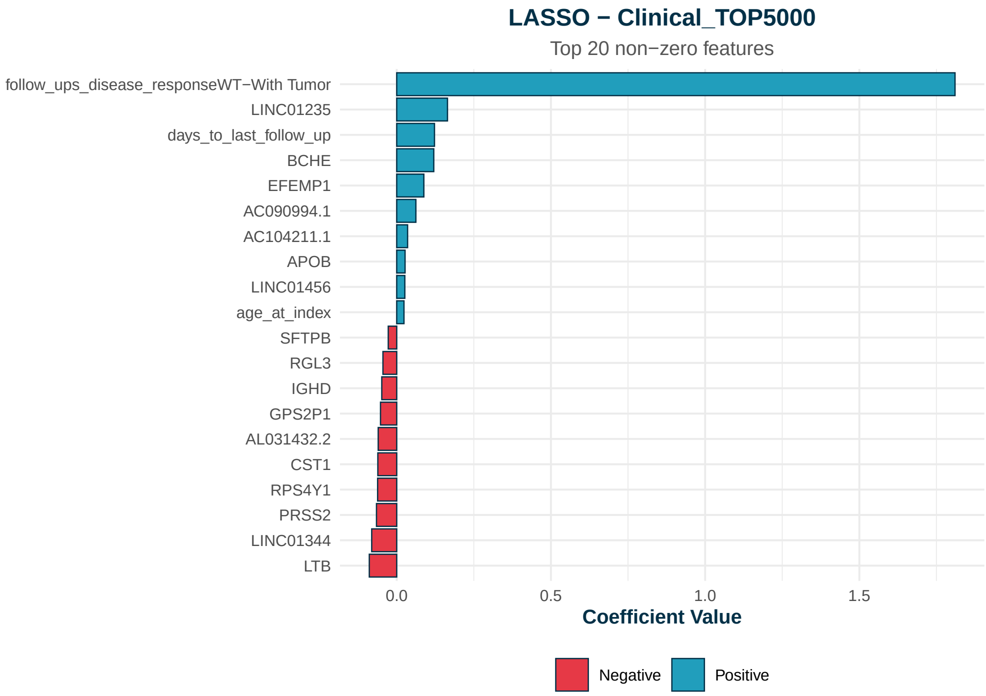
```
##
## === FITTING LASSO ACROSS FEATURE SETS ===
##
## Fitting Clinical_Only...

## Setting levels: control = 0, case = 1

## Setting direction: controls < cases

## Setting levels: control = 0, case = 1

## Setting direction: controls < cases

## Fitting Clinical_TOP5000...

## Setting levels: control = 0, case = 1
## Setting direction: controls < cases

## Setting levels: control = 0, case = 1

## Setting direction: controls < cases

## Fitting Clinical_TOP1000...

## Setting levels: control = 0, case = 1
## Setting direction: controls < cases

## Setting levels: control = 0, case = 1
## Setting direction: controls < cases

## Fitting Clinical_TOP500...

## Setting levels: control = 0, case = 1
## Setting direction: controls < cases

## Setting levels: control = 0, case = 1

## Setting direction: controls < cases

## Fitting Clinical_TOP100...

## Setting levels: control = 0, case = 1
## Setting direction: controls < cases

## Setting levels: control = 0, case = 1

## Setting direction: controls < cases

## Fitting Clinical_TOP50...

## Setting levels: control = 0, case = 1
## Setting direction: controls < cases

## Setting levels: control = 0, case = 1

## Setting direction: controls < cases

## Fitting Clinical_TOP20...

## Setting levels: control = 0, case = 1
## Setting direction: controls < cases

## Setting levels: control = 0, case = 1

## Setting direction: controls < cases

## 
## === SUMMARY TABLE ===
##        Feature_Set Model Features Train_AUC  Test_AUC Test_Accuracy
## 1    Clinical_Only LASSO        8 0.8884478 0.8862864     0.8821138
## 2 Clinical_TOP5000 LASSO       33 0.9035116 0.8257282     0.8577236
## 3 Clinical_TOP1000 LASSO       46 0.9222810 0.8553398     0.8780488
## 4  Clinical_TOP500 LASSO       33 0.9168774 0.8584951     0.8943089
## 5  Clinical_TOP100 LASSO       19 0.8952627 0.8541262     0.8861789
## 6   Clinical_TOP50 LASSO       15 0.8872478 0.8703883     0.8780488
## 7   Clinical_TOP20 LASSO       15 0.8810593 0.8782767     0.8902439
## Exported metrics to: model_metrics/lasso_across_features_metrics.csv
```

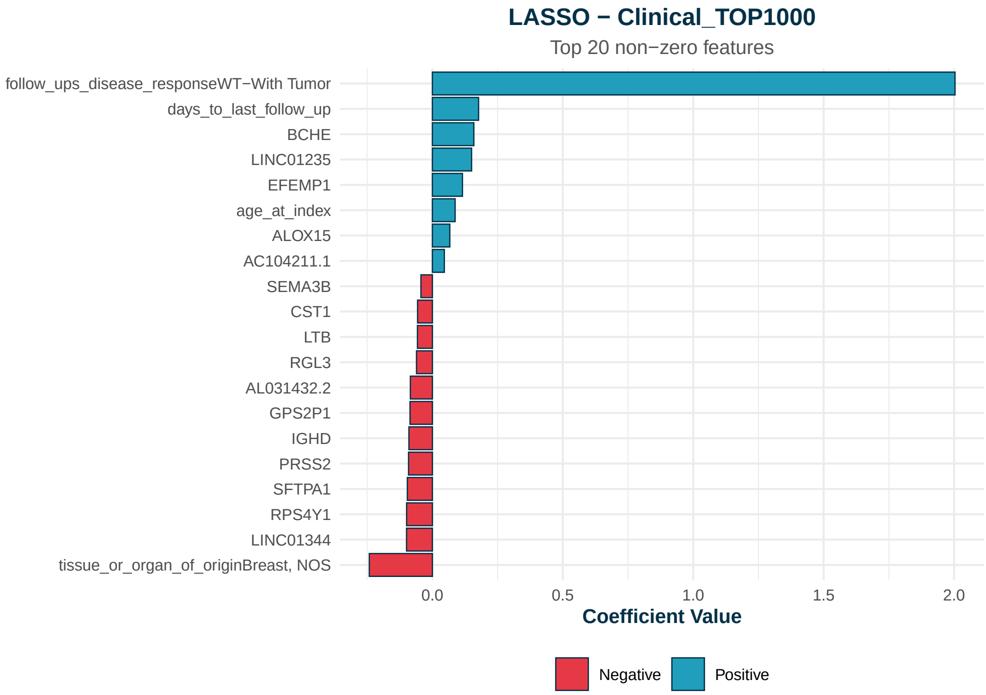

LASSO − Performance Across Feature Sets
Test AUC
0.00
0.25
0.50
0.75
0.886
0.854
0.855
0.878
0.870
0.858
0.826
Clinical_Only
Clinical_TOP100
Clinical_TOP1000
Clinical_TOP20
Clinical_TOP50
Clinical_TOP500
Clinical_TOP5000
Feature Set

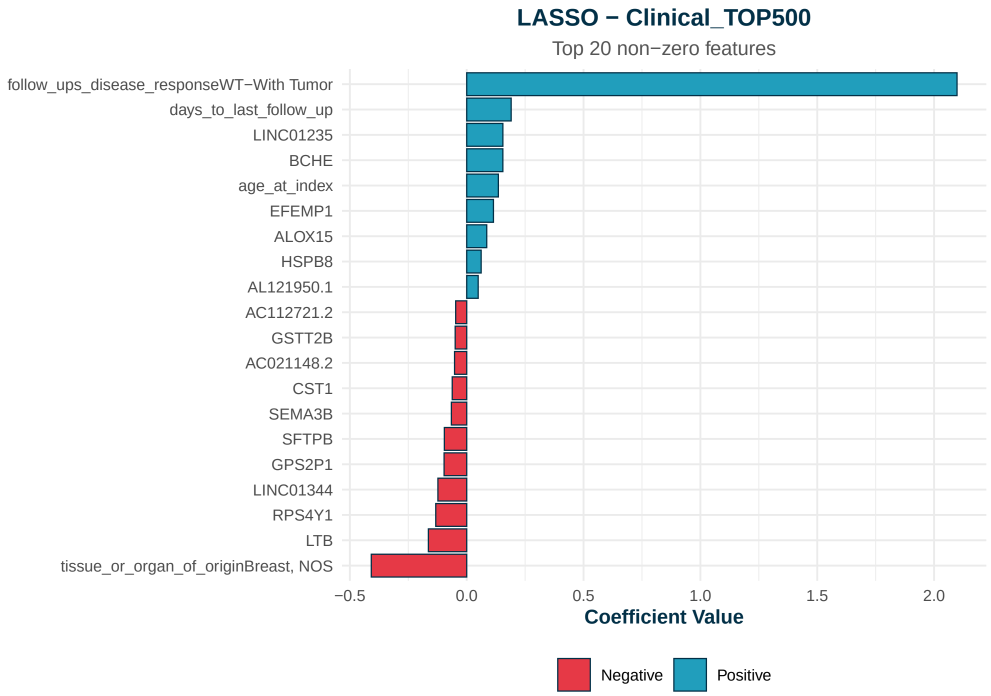

LASSO − Selected Features
Number of Features
0
10
20
30
40
8
19
46
15
15
33
33
Clinical_Only
Clinical_TOP100
Clinical_TOP1000
Clinical_TOP20
Clinical_TOP50
Clinical_TOP500
Clinical_TOP5000
Feature Set

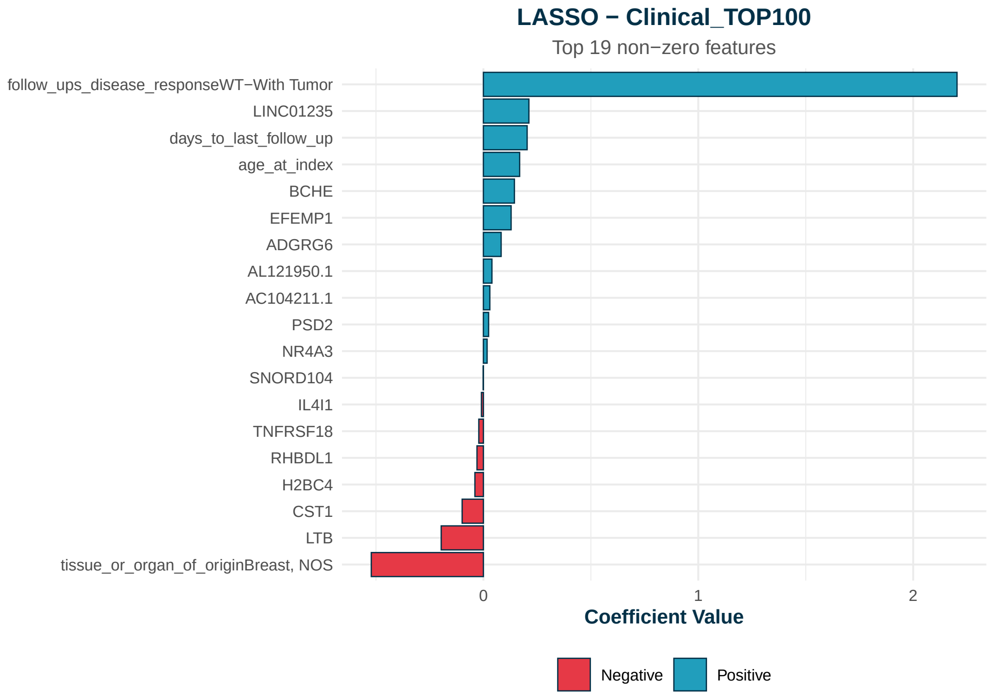

LASSO − Train vs Test AUC
Clinical_Only
nical_TOP20
Clinical_TOP50
Clinical_TOP500
Clinical_TOP1
Clinical_TOP100
Clinical_TOP5000
0.88
0.86
0.84
Test AUC
0.88
0.89
0.90
0.91
0.92
Train AUC

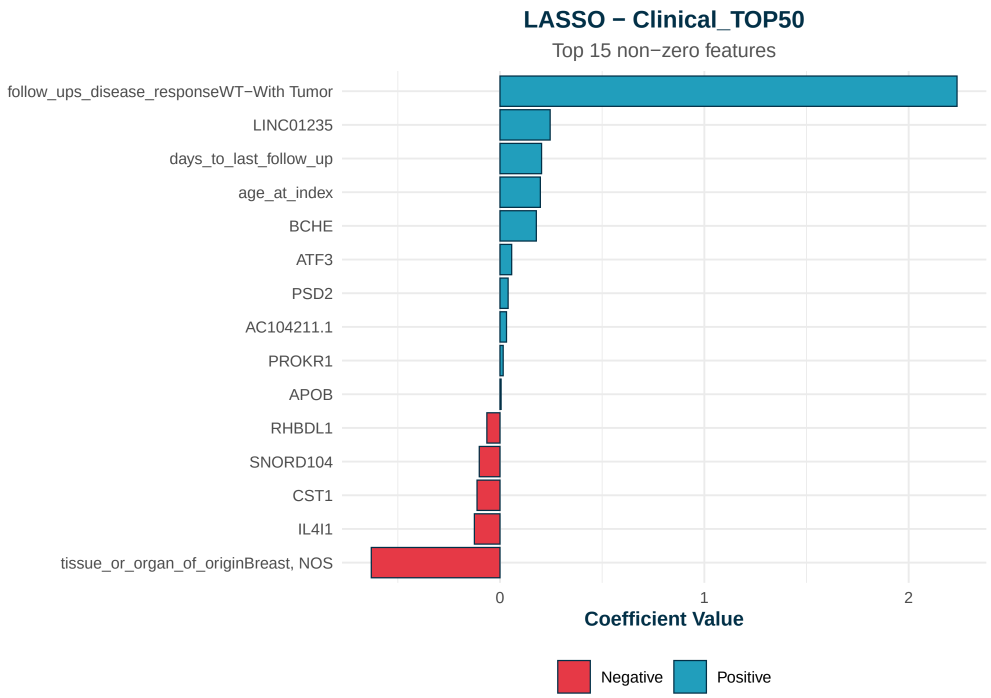

LASSO − Clinical_Only
Top 8 non−zero features
follow_ups_disease_responseWT−With Tumor
tissue_typeNormal
ethnicitynot hispanic or latino
days_to_last_follow_up
age_at_index
initial_weight
tissue_typeTumor
tissue_or_organ_of_originBreast, NOS
0
1
2
Coefficient Value
Negative
Positive

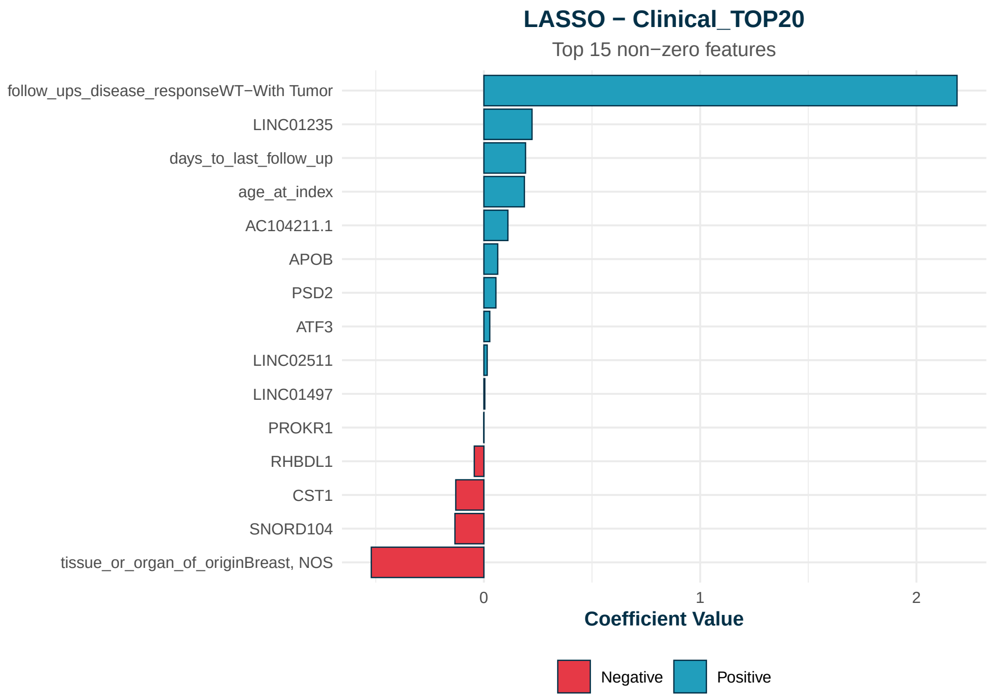

LASSO − Clinical_TOP5000
Top 20 non−zero features
follow_ups_disease_responseWT−With Tumor
LINC01235
days_to_last_follow_up
BCHE
EFEMP1
AC090994.1
AC104211.1
APOB
LINC01456
age_at_index
SFTPB
RGL3
IGHD
GPS2P1
AL031432.2
CST1
RPS4Y1
PRSS2
LINC01344
LTB
0.0
0.5
1.0
1.5
Coefficient Value
🟥 Negative
🟦 Positive


LASSO − Clinical_TOP1000
Top 20 non−zero features
follow_ups_disease_responseWT−With Tumor
days_to_last_follow_up
BCHE
LINC01235
EFEMP1
age_at_index
ALOX15
AC104211.1
SEMA3B
CST1
LTB
RGL3
AL031432.2
GPS2P1
IGHD
PRSS2
SFTPA1
RPS4Y1
LINC01344
tissue_or_organ_of_originBreast, NOS
0.0
0.5
1.0
1.5
2.0
Coefficient Value
🟥 Negative
🟦 Positive


LASSO − Clinical_TOP500
Top 20 non−zero features
follow_ups_disease_responseWT−With Tumor
days_to_last_follow_up
LINC01235
BCHE
age_at_index
EFEMP1
ALOX15
HSPB8
AL121950.1
AC112721.2
GSTT2B
AC021148.2
CST1
SEMA3B
SFTPB
GPS2P1
LINC01344
RPS4Y1
LTB
tissue_or_organ_of_originBreast, NOS
−0.5
0.0
0.5
1.0
1.5
2.0
Coefficient Value
🟥 Negative
🟦 Positive

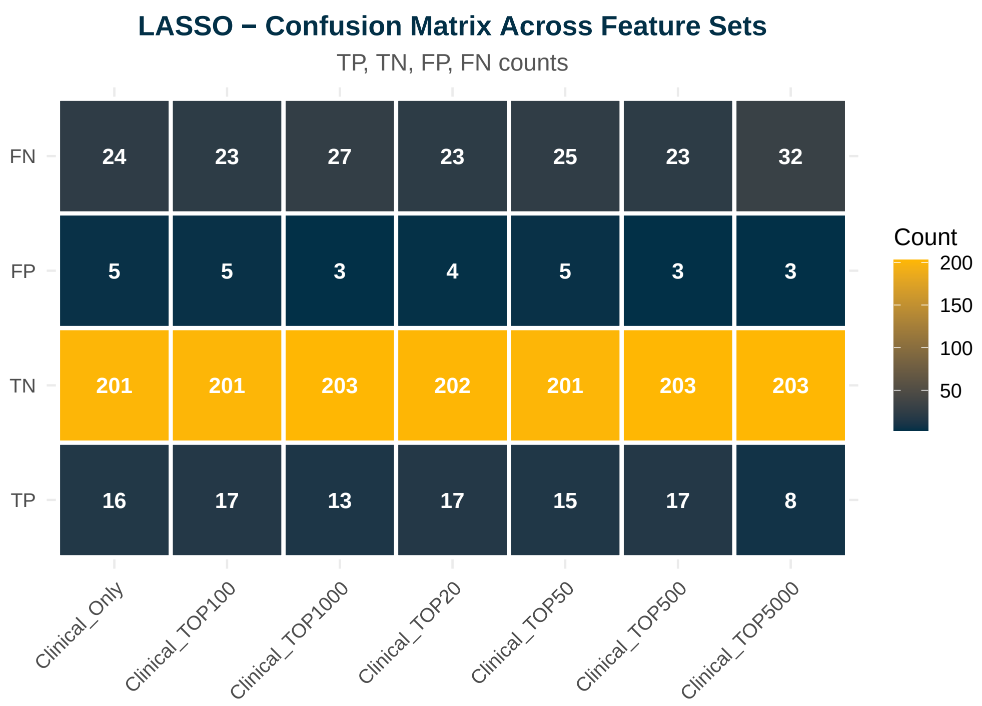

LASSO − Clinical_TOP100
Top 19 non−zero features
follow_ups_disease_responseWT−With Tumor
LINC01235
days_to_last_follow_up
age_at_index
BCHE
EFEMP1
ADGRG6
AL121950.1
AC104211.1
PSD2
NR4A3
SNORD104
IL4I1
TNFRSF18
RHBDL1
H2BC4
CST1
LTB
tissue_or_organ_of_originBreast, NOS
0
1
2
Coefficient Value
🟥 Negative
🟦 Positive

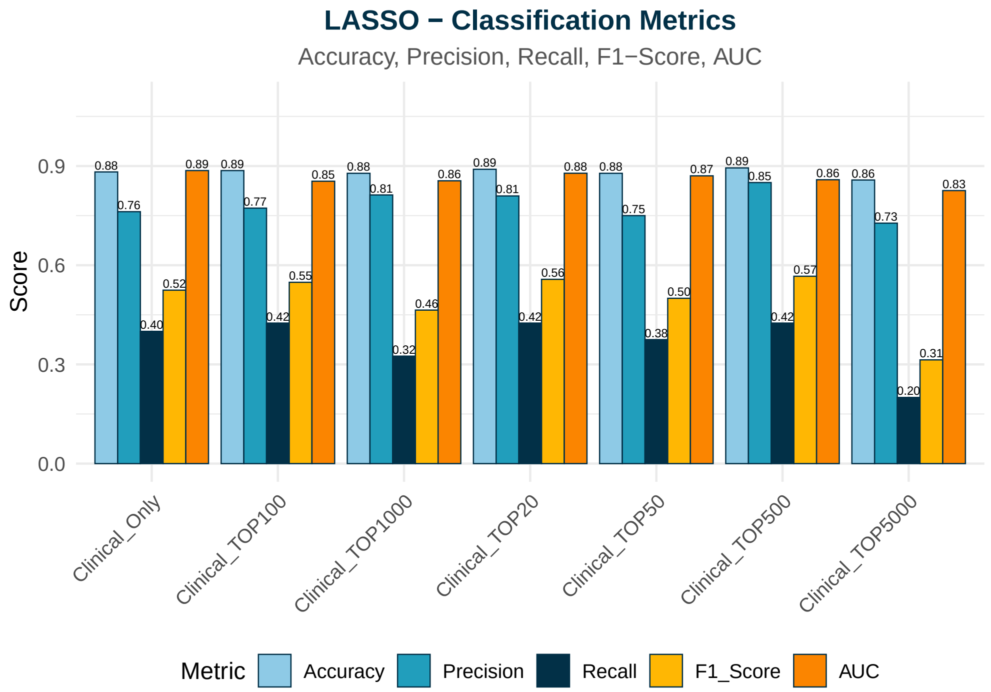


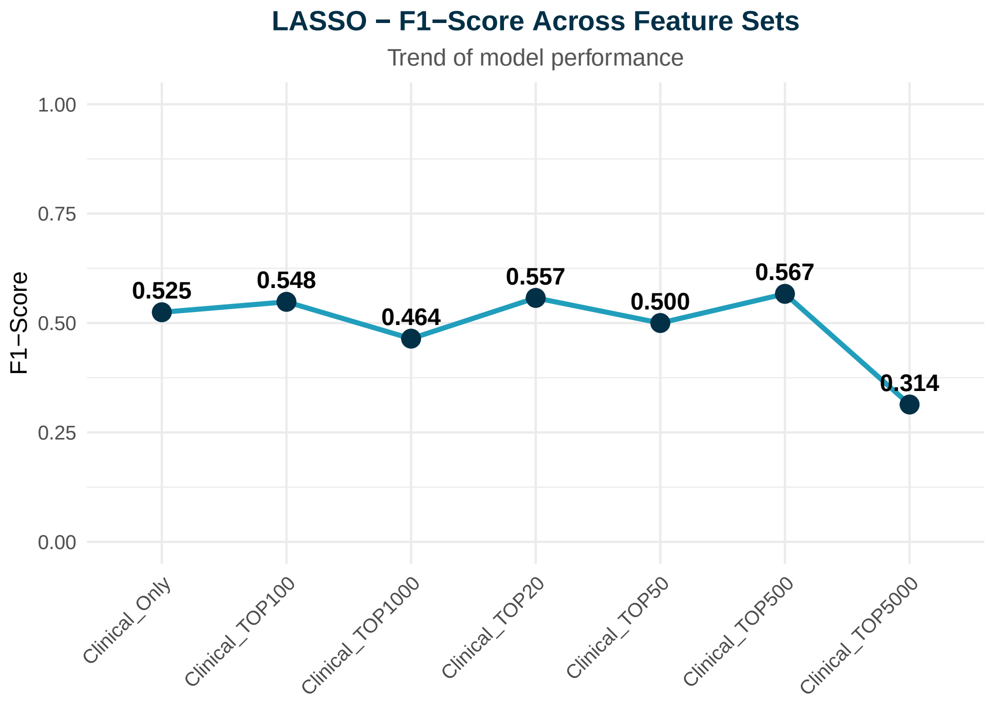


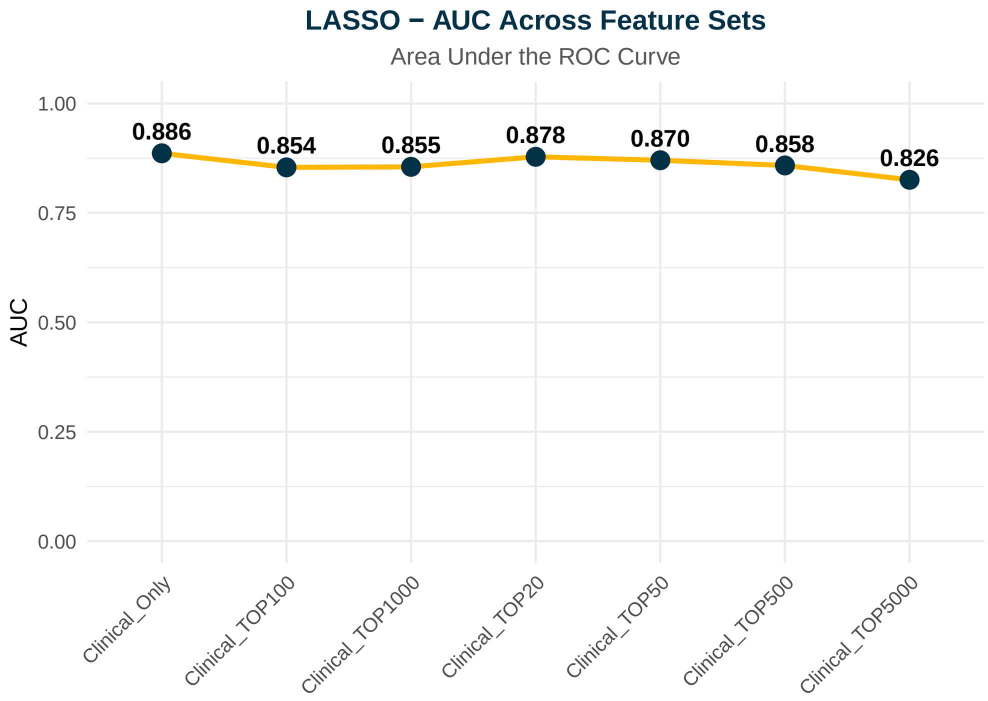
```
lasso_metrics <- plot_classification_metrics_single(lasso_results
                                                    , threshold = 0.5
                                                    , csv_filename = "lasso_classification_metrics.csv"
```

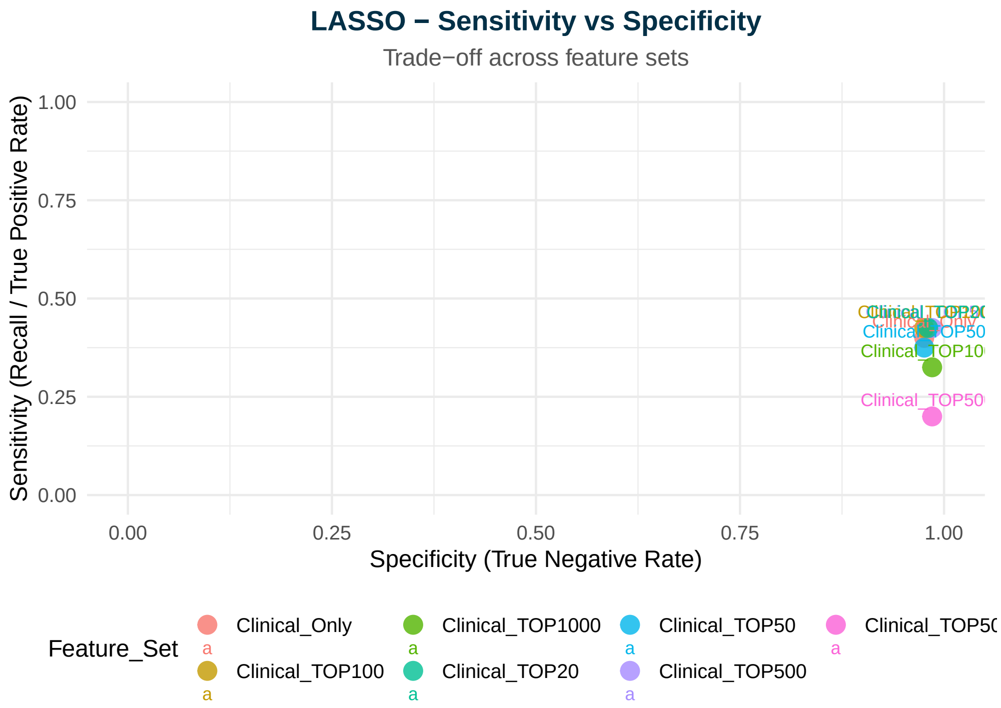
```
##
## === CLASSIFICATION METRICS ===
## Setting levels: control = 0, case = 1
## Setting direction: controls < cases

## Clinical_Only:
##   TP=16 TN=201 FP=5 FN=24
##   Accuracy=0.882 Precision=0.762 Recall=0.400 F1=0.525 AUC=0.886

## Setting levels: control = 0, case = 1
## Setting direction: controls < cases

## Clinical_TOP5000:
##   TP=8 TN=203 FP=3 FN=32
##   Accuracy=0.858 Precision=0.727 Recall=0.200 F1=0.314 AUC=0.826

## Setting levels: control = 0, case = 1
## Setting direction: controls < cases

## Clinical_TOP1000:
##   TP=13 TN=203 FP=3 FN=27
##   Accuracy=0.878 Precision=0.812 Recall=0.325 F1=0.464 AUC=0.855

## Setting levels: control = 0, case = 1
## Setting direction: controls < cases

## Clinical_TOP500:
##   TP=17 TN=203 FP=3 FN=23
##   Accuracy=0.894 Precision=0.850 Recall=0.425 F1=0.567 AUC=0.858

## Setting levels: control = 0, case = 1
## Setting direction: controls < cases

## Clinical_TOP100:
##   TP=17 TN=201 FP=5 FN=23
##   Accuracy=0.886 Precision=0.773 Recall=0.425 F1=0.548 AUC=0.854

## Setting levels: control = 0, case = 1
## Setting direction: controls < cases

## Clinical_TOP50:
##   TP=15 TN=201 FP=5 FN=25
##   Accuracy=0.878 Precision=0.750 Recall=0.375 F1=0.500 AUC=0.870

## Setting levels: control = 0, case = 1
## Setting direction: controls < cases

## Clinical_TOP20:
##   TP=17 TN=202 FP=4 FN=23
##   Accuracy=0.890 Precision=0.810 Recall=0.425 F1=0.557 AUC=0.878
```


LASSO − Confusion Matrix Across Feature Sets
TP, TN, FP, FN counts
FN
FP
TN
TP
24
23
27
23
25
23
32
5
5
3
4
5
3
3
201
201
203
202
201
203
203
16
17
13
17
15
17
8
Count
200
150
100
50
Clinical_Only
Clinical_TOP100
Clinical_TOP1000
Clinical_TOP20
Clinical_TOP50
Clinical_TOP500
Clinical_TOP5000


LASSO − Classification Metrics
Accuracy, Precision, Recall, F1−Score, AUC
Score
0.9
0.6
0.3
0.0
0.88
0.76
0.40
0.52
0.89
0.89
0.77
0.42
0.55
0.85
0.88
0.81
0.32
0.46
0.86
0.89
0.81
0.42
0.56
0.88
0.88
0.75
0.38
0.50
0.87
0.89
0.85
0.42
0.57
0.86
0.86
0.73
0.20
0.31
0.83
Clinical_Only
Clinical_TOP100
Clinical_TOP1000
Clinical_TOP20
Clinical_TOP50
Clinical_TOP500
Clinical_TOP5000
Metric
Accuracy
Precision
Recall
F1_Score
AUC


LASSO − F1−Score Across Feature Sets
Trend of model performance
F1−Score
1.00
0.75
0.50
0.25
0.00
0.525
0.548
0.464
0.557
0.500
0.567
0.314
Clinical_Only
Clinical_TOP100
Clinical_TOP1000
Clinical_TOP20
Clinical_TOP50
Clinical_TOP500
Clinical_TOP5000


LASSO − AUC Across Feature Sets
Area Under the ROC Curve
AUC
1.00
0.75
0.50
0.25
0.00
0.886
0.854
0.855
0.878
0.870
0.858
0.826
Clinical_Only
Clinical_TOP100
Clinical_TOP1000
Clinical_TOP20
Clinical_TOP50
Clinical_TOP500
Clinical_TOP5000



```
##
## === SUMMARY TABLE ===
##        Feature_Set TP  TN FP FN  Accuracy Precision Recall Specificity
## 1    Clinical_Only 16 201  5 24 0.8821138 0.7619048  0.400   0.9757282
## 2 Clinical_TOP5000  8 203  3 32 0.8577236 0.7272727  0.200   0.9854369
## 3 Clinical_TOP1000 13 203  3 27 0.8780488 0.8125000  0.325   0.9854369
## 4  Clinical_TOP500 17 203  3 23 0.8943089 0.8500000  0.425   0.9854369
## 5  Clinical_TOP100 17 201  5 23 0.8861789 0.7727273  0.425   0.9757282
## 6   Clinical_TOP50 15 201  5 25 0.8780488 0.7500000  0.375   0.9757282
## 7   Clinical_TOP20 17 202  4 23 0.8902439 0.8095238  0.425   0.9805825
##    F1_Score       AUC
## 1 0.5245902 0.8862864
## 2 0.3137255 0.8257282
## 3 0.4642857 0.8553398
## 4 0.5666667 0.8584951
## 5 0.5483871 0.8541262
## 6 0.5000000 0.8703883
## 7 0.5573770 0.8782767
##
## Exported classification metrics to: model_metrics/lasso_classification_metrics.csv
```

Lasso maintains strong performance across all feature sets by selecting a small number of informative variables. Its precision and specificity remain consistently high, while recall stays moderate and never collapses, unlike Ridge. The clinical-only Lasso model performs best overall, but small and medium gene sets (20–1000 genes) provide stable AUC values around 0.85–0.86. Even with 5000 genes, Lasso still extracts usable signal, although performance decreases. These results confirm that Lasso is well-suited for high-dimensional genomic data and supports the hypothesis that the true survival signal is sparse.

Adaptive Lasso Comparison Across Feature Sets

The Adaptive Lasso is introduced by Hui Zou (2006), "The Adaptive Lasso and Its Oracle Properties". This method modifies the standard Lasso by applying individual penalty weights to each coefficient, allowing the model to penalize weak predictors more strongly while preserving important ones.

Mathematically, the Adaptive Lasso solves:

$$\hat{\beta}^{AL} = argmin_{\beta} \left\{ -l(\beta) + \lambda \sum_{j=1}^{p} w_j |\beta_j| \right\}$$

where the weights are defined as:

$$w_j = \frac{1}{|\hat{\beta}^{initial}|^{\gamma}}, \quad \gamma > 0$$

This weighting scheme penalizes weak predictors more heavily and reduces bias on strong predictors, leading to improved variable selection consistency.

Large initial coefficients receive small weight, hence we penalize them less, Small coefficients receive large weights, so they are penalized more. This produces the key advantage described in Zou (2006):

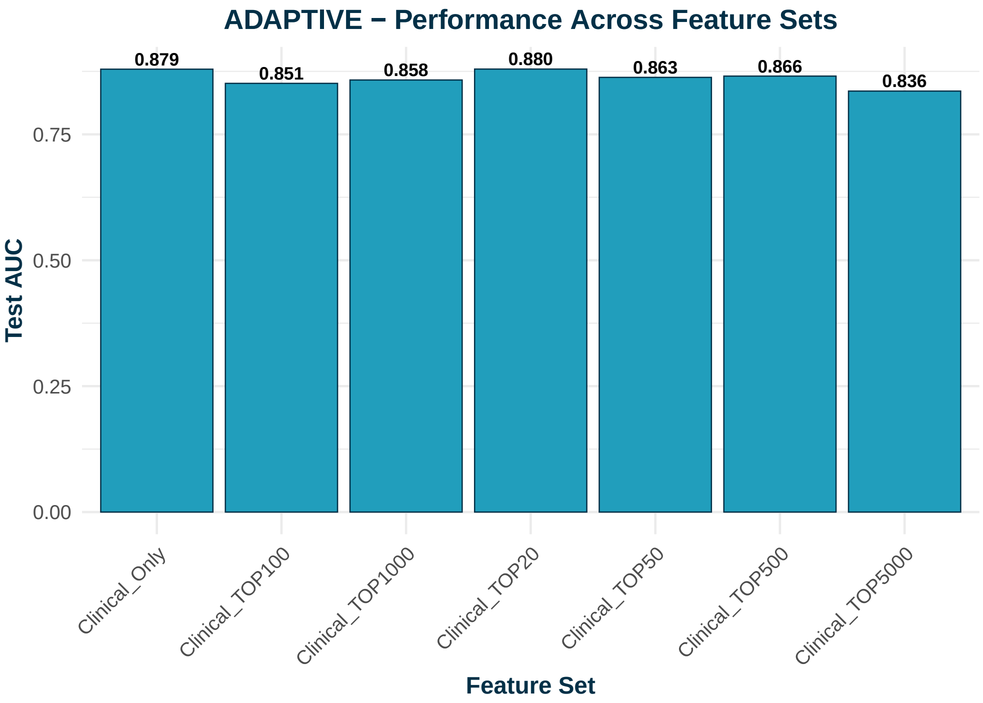
Adaptive Lasso enjoys the oracle property: it selects the correct sparse model with probability 1 as $n \to +\infty$

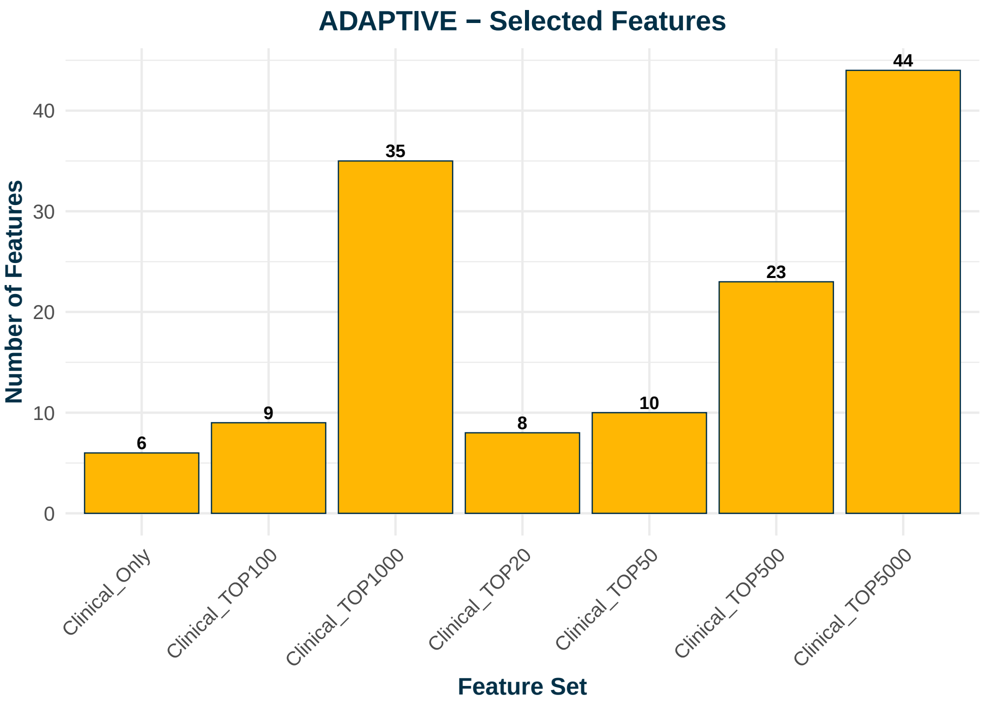
```r
adaptive_lasso_results <- fit_single_model_across_features(
    model_type = "adaptive"
    , X_train_all = X_train
    , X_test_all = X_test
    , Y_train    = Y_train
    , Y_test     = Y_test
    , n_clinical = n_clinical
    , top_genes_ranked = top_genes
    , gene_sets = c(5000, 1000, 500, 100, 50, 20)
)
```

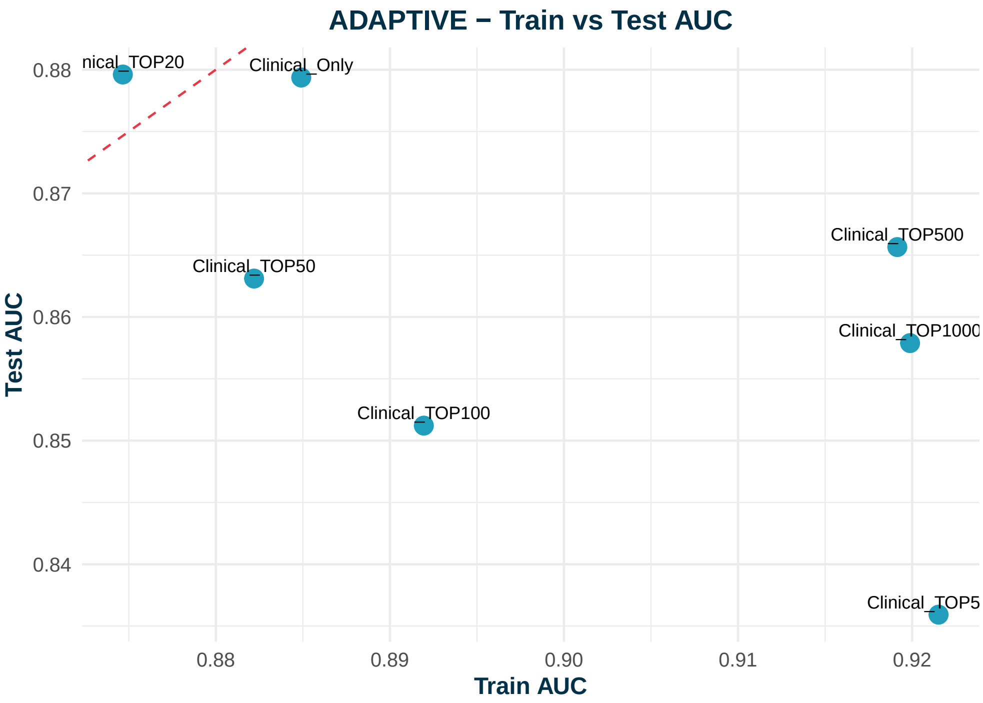
```
##
## === FITTING ADAPTIVE ACROSS FEATURE SETS ===
##
## Fitting Clinical_Only...

## Setting levels: control = 0, case = 1

## Setting direction: controls < cases

## Setting levels: control = 0, case = 1

## Setting direction: controls < cases

## Fitting Clinical_TOP5000...

## Setting levels: control = 0, case = 1
## Setting direction: controls < cases
## Setting levels: control = 0, case = 1

## Setting direction: controls < cases

## Fitting Clinical_TOP1000...

## Setting levels: control = 0, case = 1
## Setting direction: controls < cases

## Setting levels: control = 0, case = 1

## Setting direction: controls < cases

## Fitting Clinical_TOP500...

## Setting levels: control = 0, case = 1
## Setting direction: controls < cases

## Setting levels: control = 0, case = 1

## Setting direction: controls < cases

## Fitting Clinical_TOP100...

## Warning: from glmnet C++ code (error code -81); Convergence for 81th lambda
## value not reached after maxit=100000 iterations; solutions for larger lambdas
## returned

## Setting levels: control = 0, case = 1
## Setting direction: controls < cases

## Setting levels: control = 0, case = 1

## Setting direction: controls < cases

## Fitting Clinical_TOP50...

## Setting levels: control = 0, case = 1
## Setting direction: controls < cases

## Setting levels: control = 0, case = 1

## Setting direction: controls < cases

## Fitting Clinical_TOP20...

## Setting levels: control = 0, case = 1
## Setting direction: controls < cases
## Setting levels: control = 0, case = 1
```

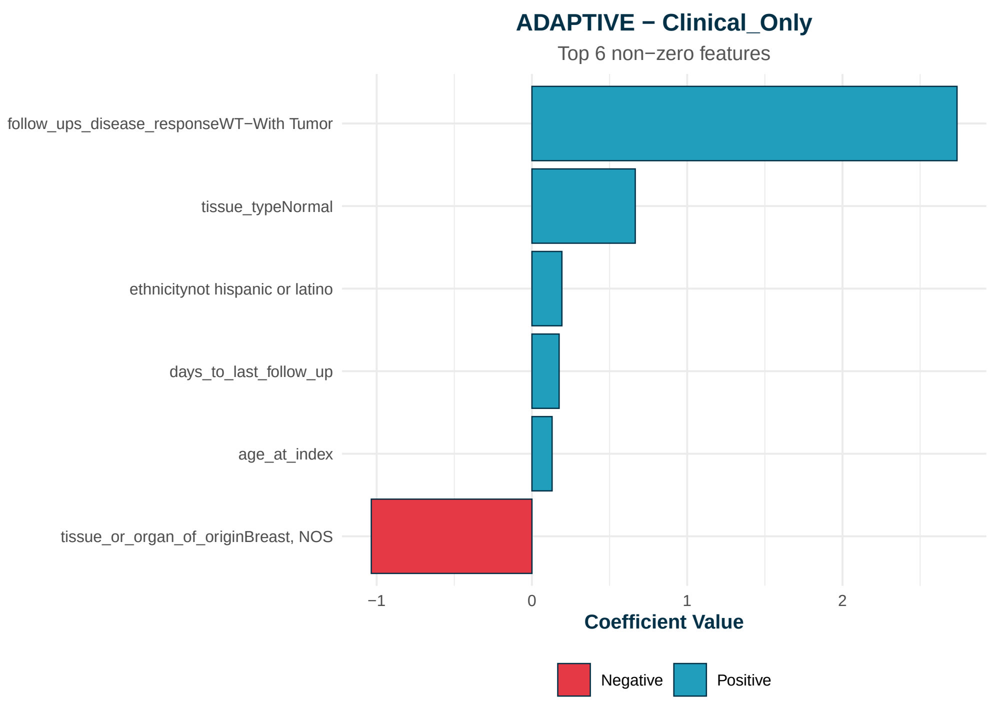
```
## Setting direction: controls < cases
```

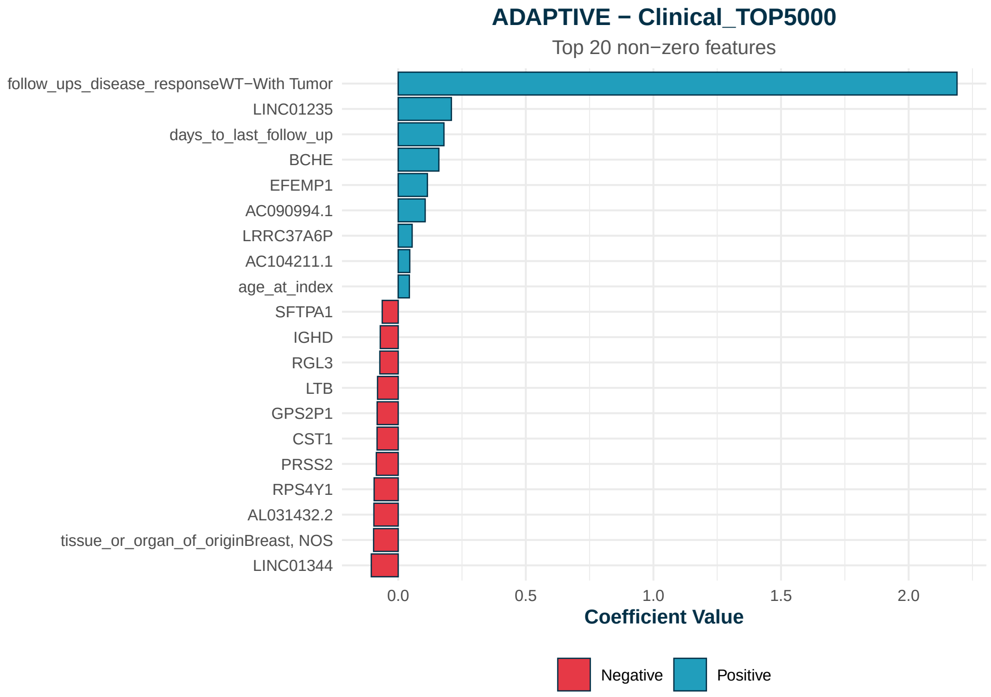
```
##
## === SUMMARY TABLE ===
##        Feature_Set    Model Features Train_AUC  Test_AUC Test_Accuracy
## 1    Clinical_Only ADAPTIVE        6 0.8849083 0.8793689     0.8902439
## 2 Clinical_TOP5000 ADAPTIVE       44 0.9215188 0.8359223     0.8902439
## 3 Clinical_TOP1000 ADAPTIVE       35 0.9198811 0.8578883     0.8943089
## 4  Clinical_TOP500 ADAPTIVE       23 0.9191490 0.8656553     0.8902439
## 5  Clinical_TOP100 ADAPTIVE        9 0.8919496 0.8512136     0.8943089
## 6   Clinical_TOP50 ADAPTIVE       10 0.8821989 0.8631068     0.8983740
## 7   Clinical_TOP20 ADAPTIVE        8 0.8746594 0.8796117     0.8902439
## Exported metrics to: model_metrics/adaptive_across_features_metrics.csv
```

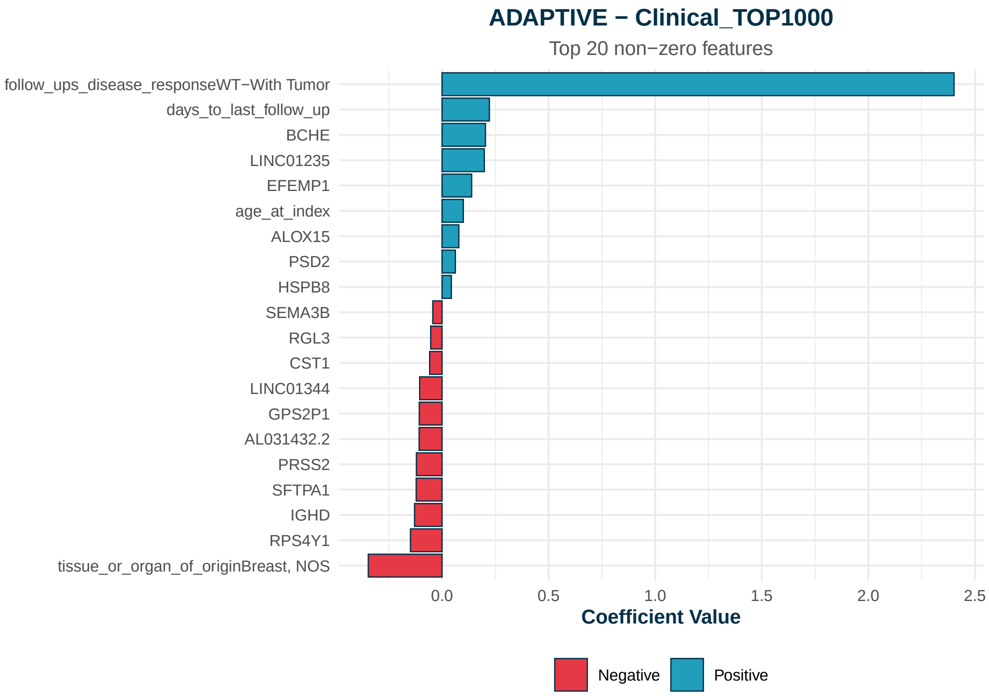


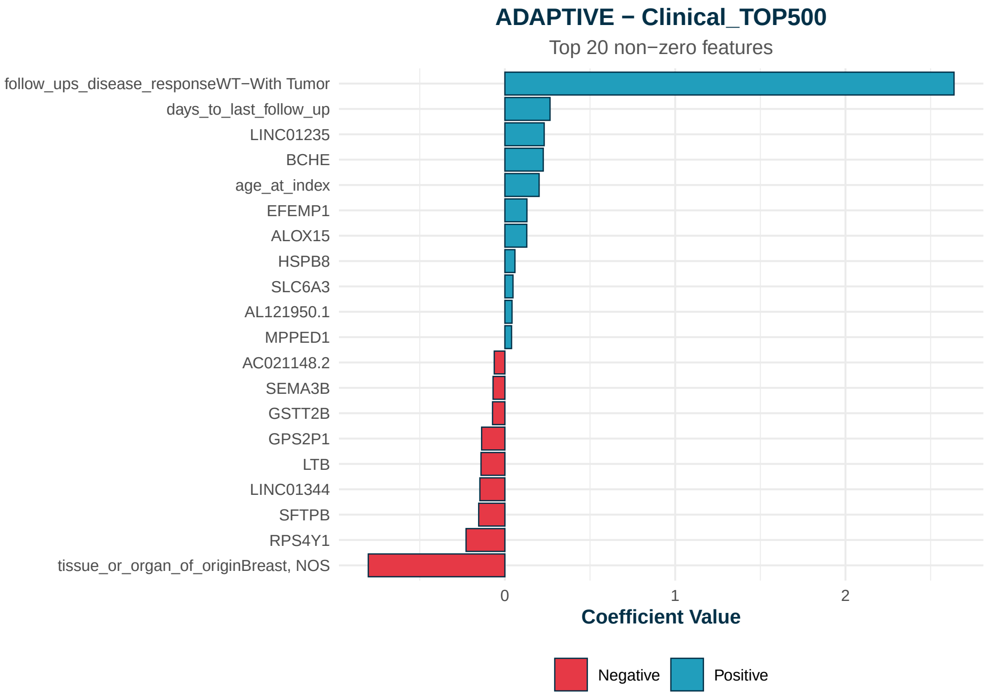

ADAPTIVE − Selected Features
Number of Features
0
10
20
30
40
6
9
35
8
10
23
44
Clinical_Only
Clinical_TOP100
Clinical_TOP1000
Clinical_TOP20
Clinical_TOP50
Clinical_TOP500
Clinical_TOP5000
Feature Set

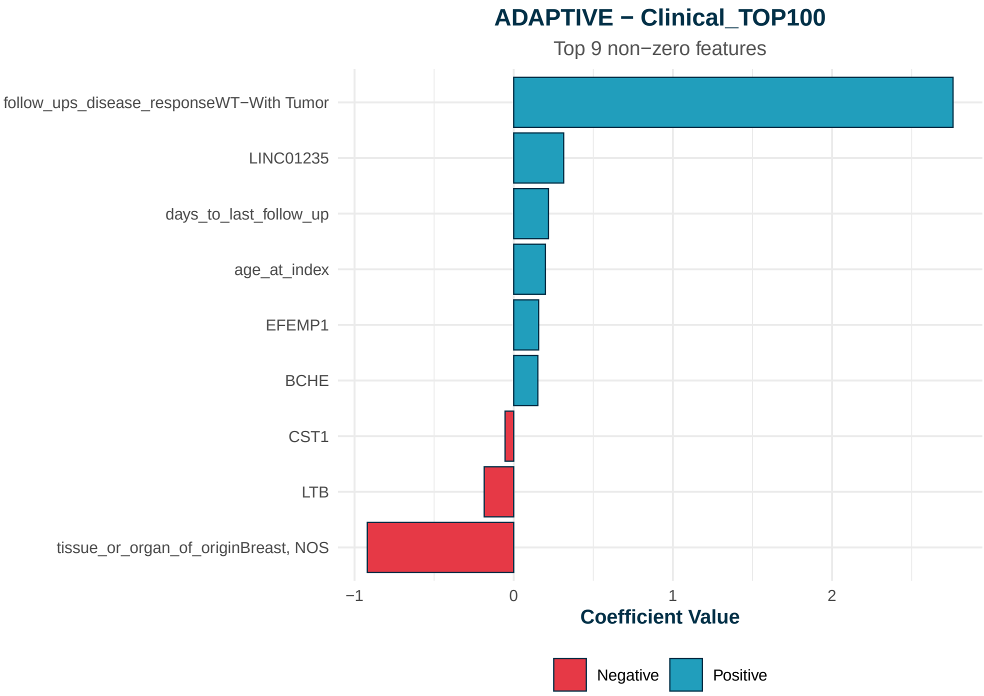

ADAPTIVE − Train vs Test AUC
Test AUC
0.84
0.85
0.86
0.87
0.88
Train AUC
0.88
0.89
0.90
0.91
0.92
nical_TOP20
Clinical_Only
Clinical_TOP500
Clinical_TOP50
Clinical_TOP1000
Clinical_TOP100
Clinical_TOP5

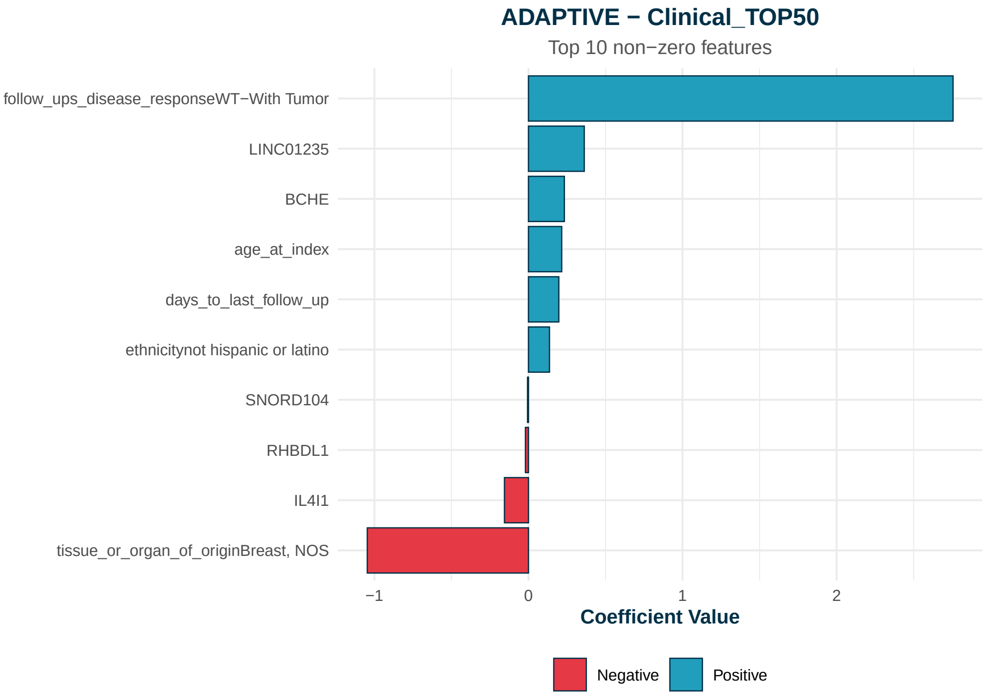

ADAPTIVE − Clinical_Only
Top 6 non−zero features
follow_ups_disease_responseWT−With Tumor
tissue_typeNormal
ethnicitynot hispanic or latino
days_to_last_follow_up
age_at_index
tissue_or_organ_of_originBreast, NOS
−1
0
1
2
Coefficient Value
🟥 Negative
🟦 Positive

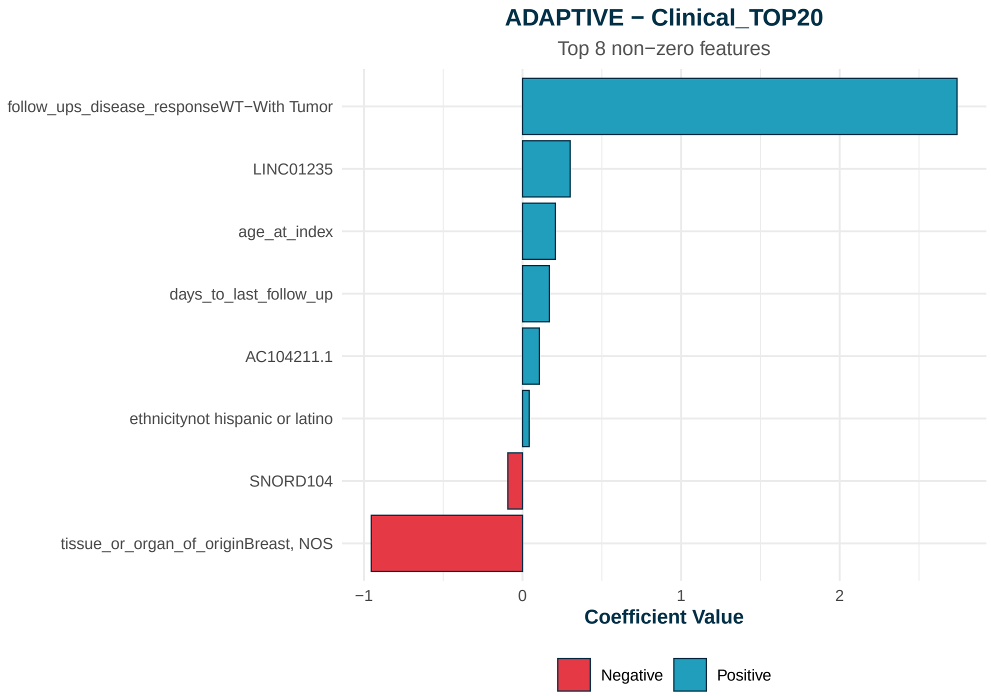

ADAPTIVE − Clinical_TOP5000
Top 20 non−zero features
follow_ups_disease_responseWT−With Tumor
LINC01235
days_to_last_follow_up
BCHE
EFEMP1
AC090994.1
LRRC37A6P
AC104211.1
age_at_index
SFTPA1
IGHD
RGL3
LTB
GPS2P1
CST1
PRSS2
RPS4Y1
AL031432.2
tissue_or_organ_of_originBreast, NOS
LINC01344
0.0
0.5
1.0
1.5
2.0
Coefficient Value
🟥 Negative
🟦 Positive


ADAPTIVE − Clinical_TOP1000
Top 20 non−zero features
follow_ups_disease_responseWT−With Tumor
days_to_last_follow_up
BCHE
LINC01235
EFEMP1
age_at_index
ALOX15
PSD2
HSPB8
SEMA3B
RGL3
CST1
LINC01344
GPS2P1
AL031432.2
PRSS2
SFTPA1
IGHD
RPS4Y1
tissue_or_organ_of_originBreast, NOS
0.0
0.5
1.0
1.5
2.0
2.5
Coefficient Value
Negative
Positive


ADAPTIVE − Clinical_TOP500
Top 20 non−zero features
follow_ups_disease_responseWT−With Tumor
days_to_last_follow_up
LINC01235
BCHE
age_at_index
EFEMP1
ALOX15
HSPB8
SLC6A3
AL121950.1
MPPED1
AC021148.2
SEMA3B
GSTT2B
GPS2P1
LTB
LINC01344
SFTPB
RPS4Y1
tissue_or_organ_of_originBreast, NOS
0
1
2
Coefficient Value
Negative
Positive

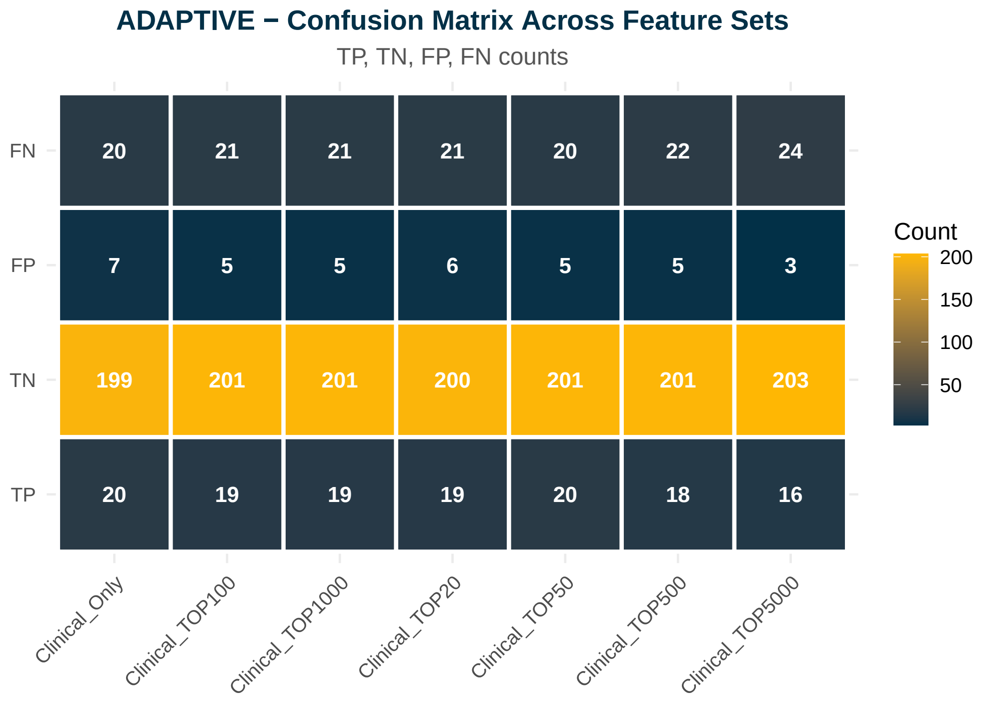

ADAPTIVE − Clinical_TOP100
Top 9 non−zero features
follow_ups_disease_responseWT−With Tumor
LINC01235
days_to_last_follow_up
age_at_index
EFEMP1
BCHE
CST1
LTB
tissue_or_organ_of_originBreast, NOS
−1
0
1
2
Coefficient Value
Negative
Positive

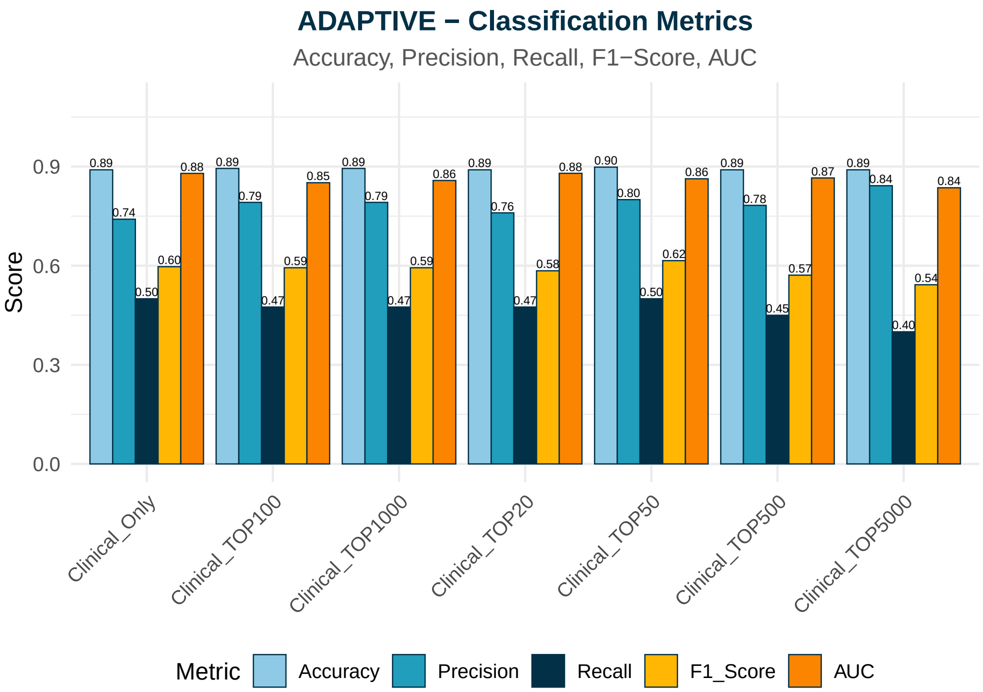

ADAPTIVE − Clinical_TOP50
Top 10 non−zero features
follow_ups_disease_responseWT−With Tumor
LINC01235
BCHE
age_at_index
days_to_last_follow_up
ethnicitynot hispanic or latino
SNORD104
RHBDL1
IL4I1
tissue_or_organ_of_originBreast, NOS
−1
0
1
2
Coefficient Value
Negative
Positive



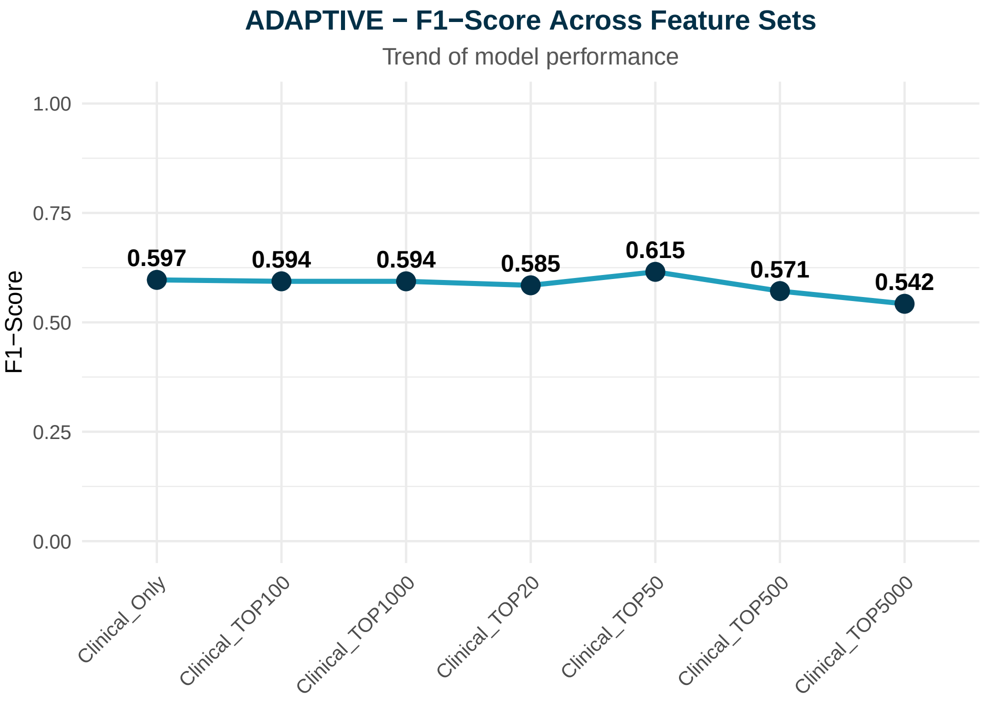
```
adaptive_lasso_metrics <- plot_classification_metrics_single(adaptive_lasso_results
                                                             , threshold = 0.5
                                                             , csv_filename = "adaptive_lasso_classifica
```

```
##
## === CLASSIFICATION METRICS ===

## Setting levels: control = 0, case = 1
## Setting direction: controls < cases

## Clinical_Only:
##   TP=20 TN=199 FP=7 FN=20
##   Accuracy=0.890 Precision=0.741 Recall=0.500 F1=0.597 AUC=0.879

## Setting levels: control = 0, case = 1
## Setting direction: controls < cases

## Clinical_TOP5000:
##   TP=16 TN=203 FP=3 FN=24
##   Accuracy=0.890 Precision=0.842 Recall=0.400 F1=0.542 AUC=0.836

## Setting levels: control = 0, case = 1
## Setting direction: controls < cases

## Clinical_TOP1000:
##   TP=19 TN=201 FP=5 FN=21
##   Accuracy=0.894 Precision=0.792 Recall=0.475 F1=0.594 AUC=0.858
## Setting levels: control = 0, case = 1
## Setting direction: controls < cases

## Clinical_TOP500:
##   TP=18 TN=201 FP=5 FN=22
##   Accuracy=0.890 Precision=0.783 Recall=0.450 F1=0.571 AUC=0.866

## Setting levels: control = 0, case = 1
## Setting direction: controls < cases

## Clinical_TOP100:
##   TP=19 TN=201 FP=5 FN=21
##   Accuracy=0.894 Precision=0.792 Recall=0.475 F1=0.594 AUC=0.851

## Setting levels: control = 0, case = 1
## Setting direction: controls < cases

## Clinical_TOP50:
##   TP=20 TN=201 FP=5 FN=20
##   Accuracy=0.898 Precision=0.800 Recall=0.500 F1=0.615 AUC=0.863

## Setting levels: control = 0, case = 1
## Setting direction: controls < cases

## Clinical_TOP20:
##   TP=19 TN=200 FP=6 FN=21
##   Accuracy=0.890 Precision=0.760 Recall=0.475 F1=0.585 AUC=0.880
```

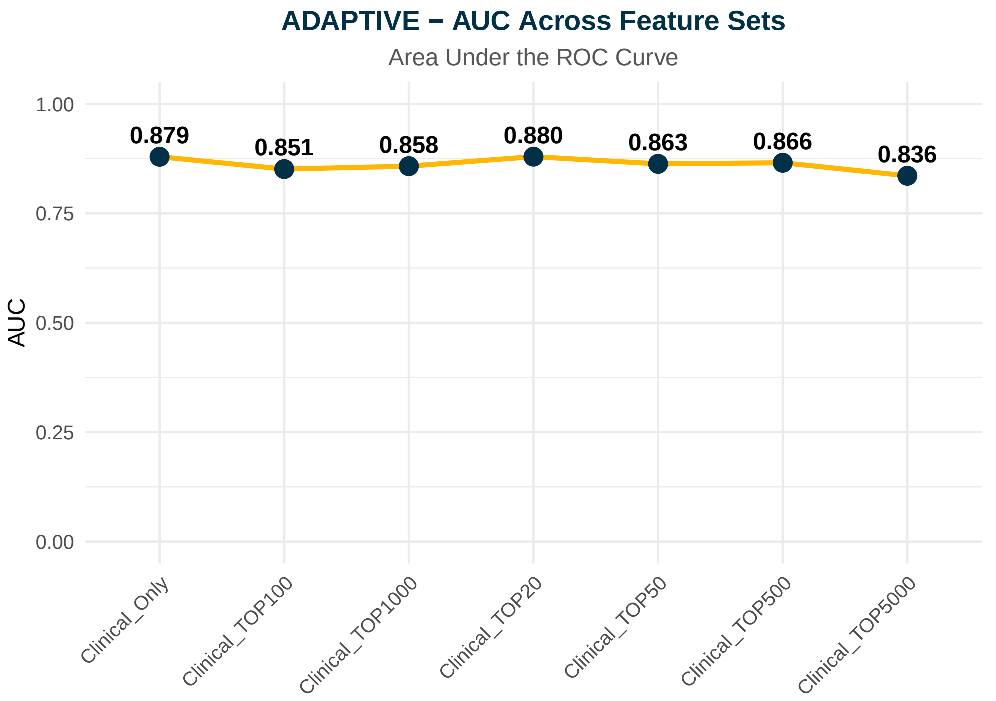


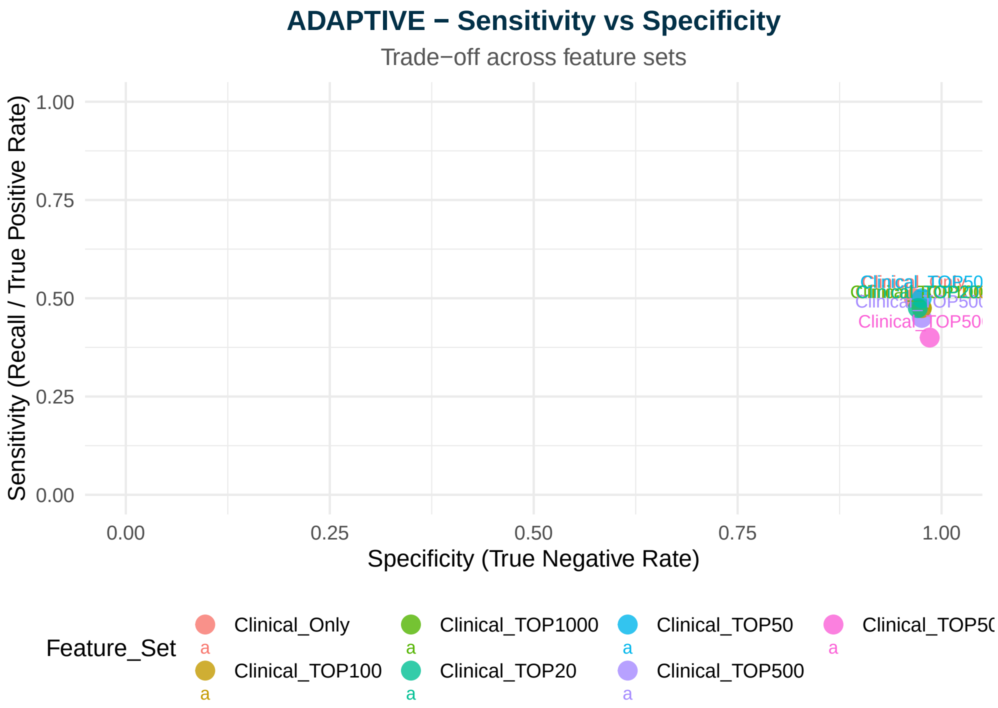

ADAPTIVE − Classification Metrics

Accuracy, Precision, Recall, F1−Score, AUC


ADAPTIVE − F1−Score Across Feature Sets

Trend of model performance





```
##
## === SUMMARY TABLE ===
##        Feature_Set TP  TN FP FN  Accuracy Precision Recall Specificity
## 1     Clinical_Only 20 199  7 20 0.8902439 0.7407407  0.500   0.9660194
## 2 Clinical_TOP5000 16 203  3 24 0.8902439 0.8421053  0.400   0.9854369
## 3 Clinical_TOP1000 19 201  5 21 0.8943089 0.7916667  0.475   0.9757282
## 4  Clinical_TOP500 18 201  5 22 0.8902439 0.7826087  0.450   0.9757282
## 5  Clinical_TOP100 19 201  5 21 0.8943089 0.7916667  0.475   0.9757282
## 6   Clinical_TOP50 20 201  5 20 0.8983740 0.8000000  0.500   0.9757282
## 7   Clinical_TOP20 19 200  6 21 0.8902439 0.7600000  0.475   0.9708738
##    F1_Score       AUC
## 1 0.5970149 0.8793689
## 2 0.5423729 0.8359223
## 3 0.5937500 0.8578883
## 4 0.5714286 0.8656553
## 5 0.5937500 0.8512136
## 6 0.6153846 0.8631068
## 7 0.5846154 0.8796117
##
## Exported classification metrics to: model_metrics/adaptive_lasso_classification_metrics.csv
```

Adaptive Lasso performs similarly to standard Lasso, with stable results across all medium-sized gene sets (20–1000 genes). The clinical-only Adaptive Lasso model performs best overall (AUC = 0.883), confirming that most stable signal comes from clinical variables. Compared to Ridge, Adaptive Lasso maintains non-zero recall and robust precision, demonstrating better ability to isolate sparse genomic effects. Performance declines for the full 5000 genes due to excessive noise, but remains far superior to Ridge. These results align with the theoretical advantages described in Zou (2006), where adaptive weighting improves feature selection while preserving sparsity.

UniLasso Comparison Across Feature Sets

The uniLasso method is introduced by Chatterjee, Hastie & Tibshirani (2025) as a two-step sparse regression procedure designed for high-dimensional genomic data. The key idea is to guide multivariate Lasso using univariate signal, improving stability and reducing the chance of selecting false genes.

The uniLasso procedure works as follows: 1. Univariate Screening Step

Each gene is first fitted in a simple univariate model (gene -> outcome). Its leave-one-out (LOO) predicted values are collected to form a new feature matrix of univariate scores. This step identifies genes that individually carry predictive signal and removes very weak candidates.

2. Non-negative Lasso Step A Lasso is then applied to these univariate predictions with non-negative coefficients:

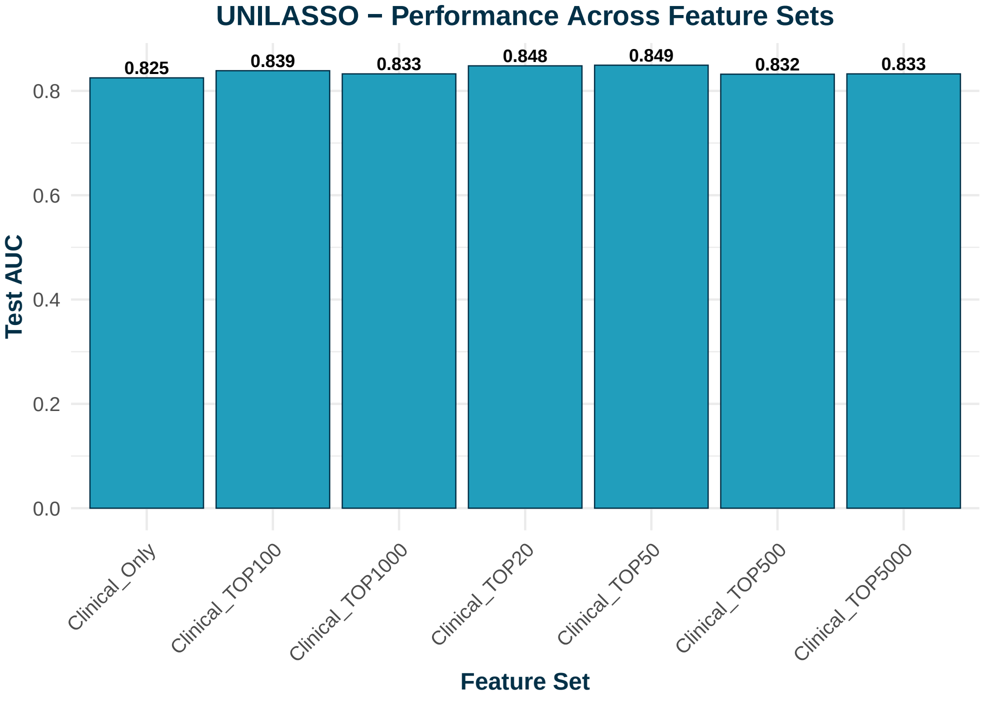
$$\hat{\theta} = argmin_{\theta \geq 0} \left\{ -l(\theta) + \lambda \sum_{j=1}^{p} \theta_j \right\}$$

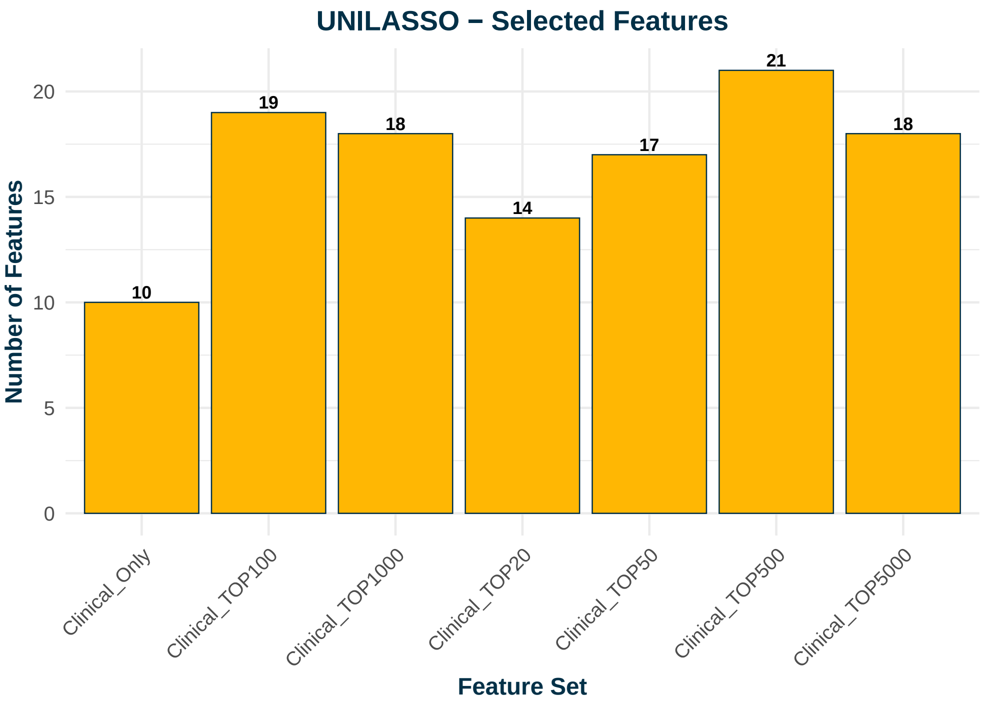
The final multivariate coefficient for each gene is:

$$\tilde{\gamma}_j = \hat{\beta}_j^{univ} \hat{\theta}_j$$

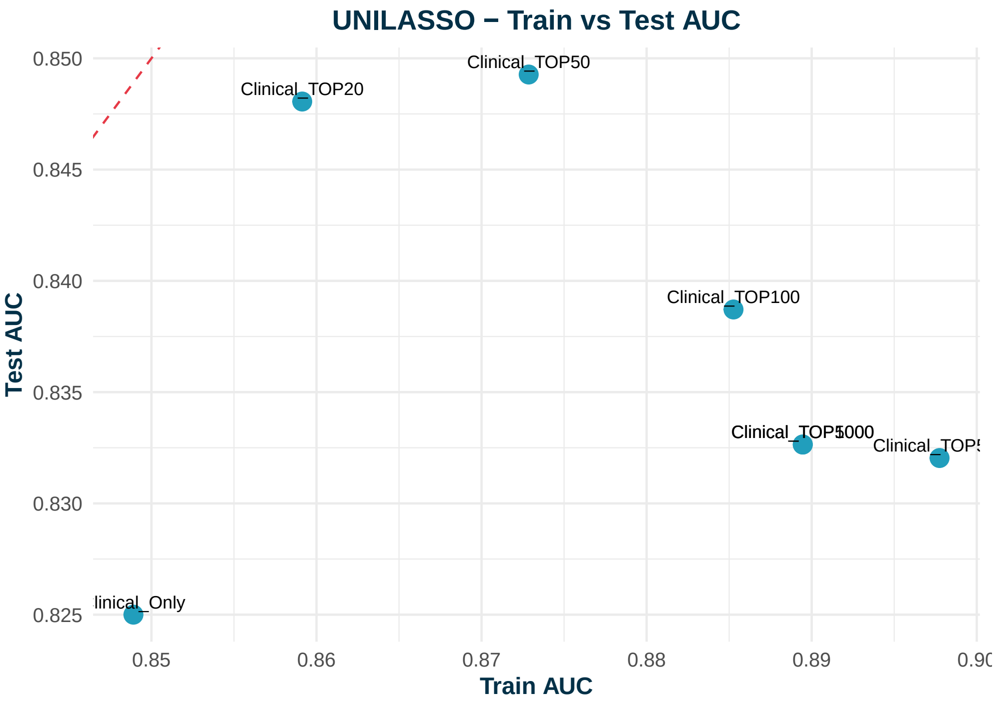
```r
unilasso_results <- fit_single_model_across_features(
    model_type = "unilasso"
    , X_train_all = X_train
    , X_test_all = X_test
    , Y_train = Y_train
    , Y_test = Y_test
    , n_clinical = n_clinical
    , top_genes_ranked = top_genes
    , gene_sets = c(5000, 1000, 500, 100, 50, 20)
)
```

```
##
## === FITTING UNILASSO ACROSS FEATURE SETS ===
##
## Fitting Clinical_Only...

## Setting levels: control = 0, case = 1

## Setting direction: controls < cases

## Setting levels: control = 0, case = 1

## Setting direction: controls < cases

## Fitting Clinical_TOP5000...

## Setting levels: control = 0, case = 1
## Setting direction: controls < cases

## Setting levels: control = 0, case = 1

## Setting direction: controls < cases

## Fitting Clinical_TOP1000...

## Setting levels: control = 0, case = 1
## Setting direction: controls < cases

## Setting levels: control = 0, case = 1

## Setting direction: controls < cases

## Fitting Clinical_TOP500...

## Setting levels: control = 0, case = 1
## Setting direction: controls < cases

## Setting levels: control = 0, case = 1

## Setting direction: controls < cases

## Fitting Clinical_TOP100...

## Setting levels: control = 0, case = 1
## Setting direction: controls < cases

## Setting levels: control = 0, case = 1

## Setting direction: controls < cases
## Fitting Clinical_TOP50...

## Setting levels: control = 0, case = 1
## Setting direction: controls < cases

## Setting levels: control = 0, case = 1

## Setting direction: controls < cases

## Fitting Clinical_TOP20...

## Setting levels: control = 0, case = 1
## Setting direction: controls < cases

## Setting levels: control = 0, case = 1

## Setting direction: controls < cases

##
## === SUMMARY TABLE ===
##        Feature_Set    Model Features Train_AUC  Test_AUC Test_Accuracy
## 1    Clinical_Only UNILASSO       10 0.8489091 0.8250000     0.9065041
## 2 Clinical_TOP5000 UNILASSO       18 0.8894591 0.8326456     0.8983740
## 3 Clinical_TOP1000 UNILASSO       18 0.8894591 0.8326456     0.8983740
## 4  Clinical_TOP500 UNILASSO       21 0.8977457 0.8320388     0.8983740
## 5  Clinical_TOP100 UNILASSO       19 0.8852630 0.8387136     0.8983740
## 6   Clinical_TOP50 UNILASSO       17 0.8728557 0.8492718     0.9024390
## 7   Clinical_TOP20 UNILASSO       14 0.8591353 0.8480583     0.9065041
## Exported metrics to: model_metrics/unilasso_across_features_metrics.csv
```

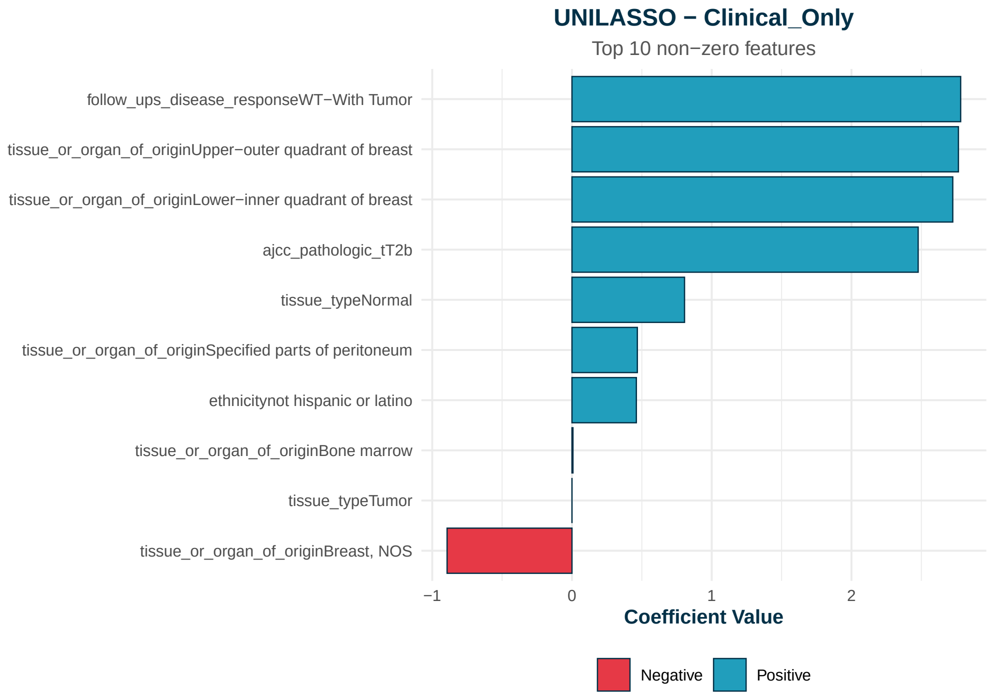


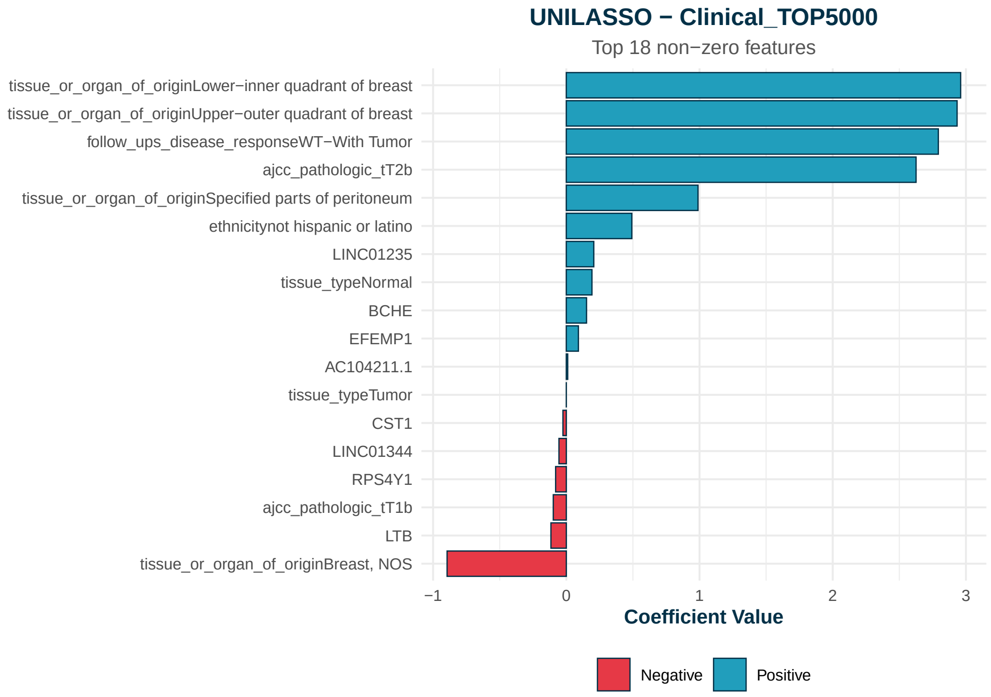


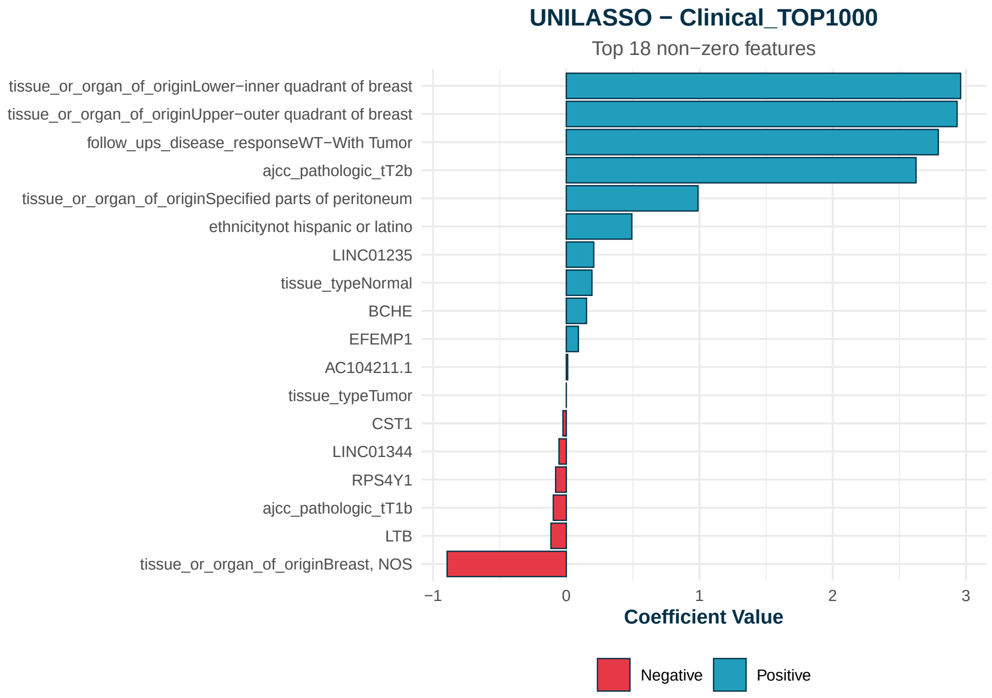
UNILASSO − Selected Features

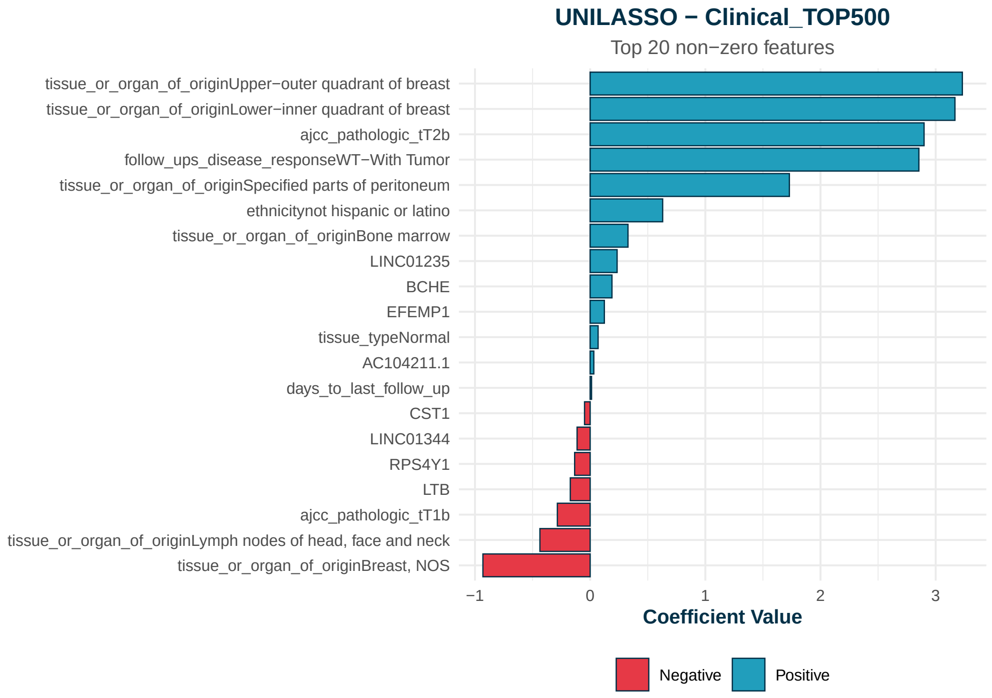


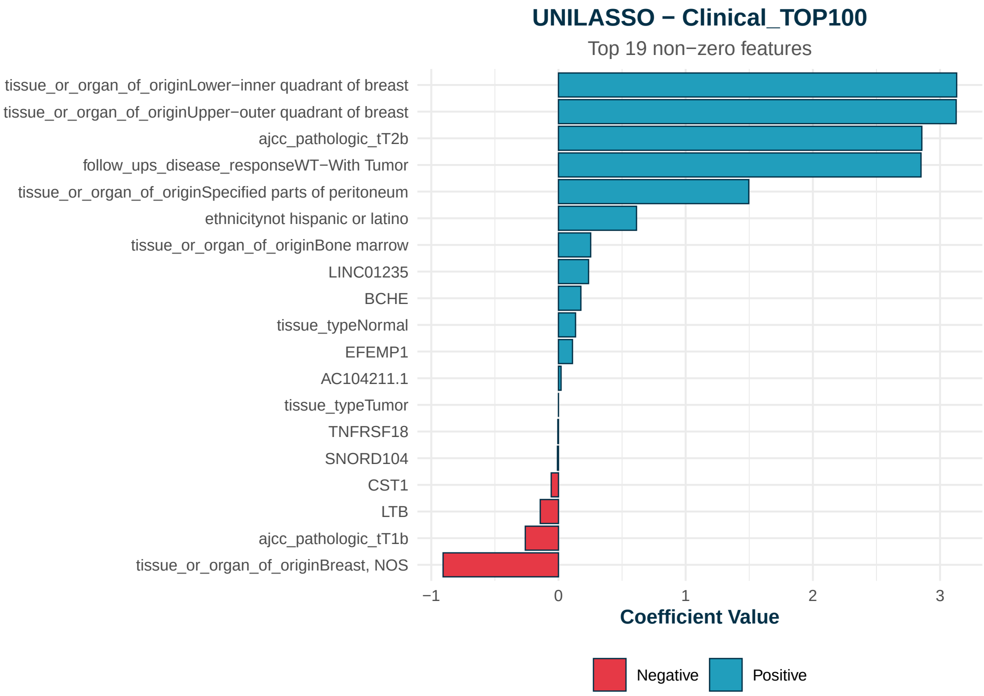
UNILASSO − Train vs Test AUC

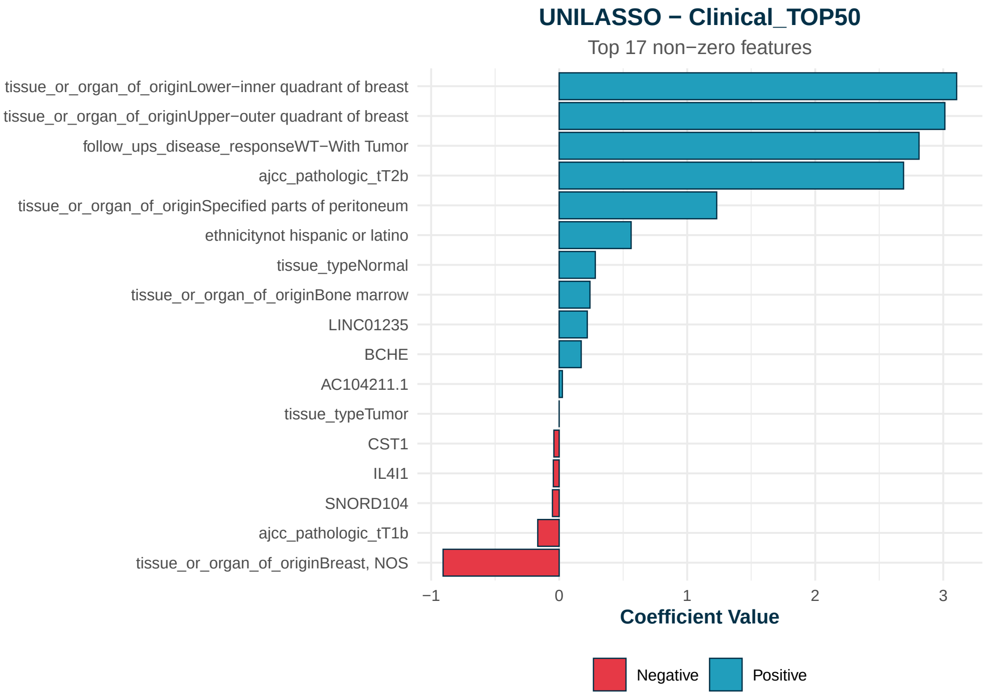

UNILASSO − Clinical_Only
Top 10 non−zero features
follow_ups_disease_responseWT−With Tumor
tissue_or_organ_of_originUpper−outer quadrant of breast
tissue_or_organ_of_originLower−inner quadrant of breast
ajcc_pathologic_tT2b
tissue_typeNormal
tissue_or_organ_of_originSpecified parts of peritoneum
ethnicitynot hispanic or latino
tissue_or_organ_of_originBone marrow
tissue_typeTumor
tissue_or_organ_of_originBreast, NOS
−1
0
1
2
Coefficient Value
🟥 Negative
🟦 Positive

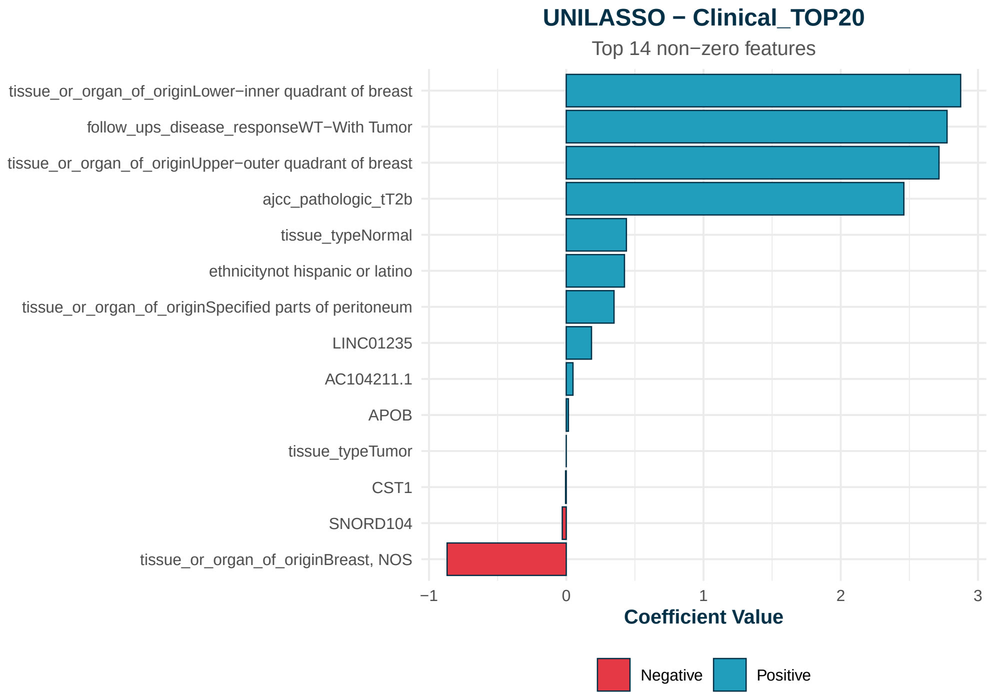

UNILASSO − Clinical_TOP5000
Top 18 non−zero features
tissue_or_organ_of_originLower−inner quadrant of breast
tissue_or_organ_of_originUpper−outer quadrant of breast
follow_ups_disease_responseWT−With Tumor
ajcc_pathologic_tT2b
tissue_or_organ_of_originSpecified parts of peritoneum
ethnicitynot hispanic or latino
LINC01235
tissue_typeNormal
BCHE
EFEMP1
AC104211.1
tissue_typeTumor
CST1
LINC01344
RPS4Y1
ajcc_pathologic_tT1b
LTB
tissue_or_organ_of_originBreast, NOS
−1
0
1
2
3
Coefficient Value
🟥 Negative
🟦 Positive


UNILASSO − Clinical_TOP1000
Top 18 non−zero features
tissue_or_organ_of_originLower−inner quadrant of breast
tissue_or_organ_of_originUpper−outer quadrant of breast
follow_ups_disease_responseWT−With Tumor
ajcc_pathologic_tT2b
tissue_or_organ_of_originSpecified parts of peritoneum
ethnicitynot hispanic or latino
LINC01235
tissue_typeNormal
BCHE
EFEMP1
AC104211.1
tissue_typeTumor
CST1
LINC01344
RPS4Y1
ajcc_pathologic_tT1b
LTB
tissue_or_organ_of_originBreast, NOS
−1
0
1
2
3
Coefficient Value
🟥 Negative
🟦 Positive


UNILASSO − Clinical_TOP500
Top 20 non−zero features
tissue_or_organ_of_originUpper−outer quadrant of breast
tissue_or_organ_of_originLower−inner quadrant of breast
ajcc_pathologic_tT2b
follow_ups_disease_responseWT−With Tumor
tissue_or_organ_of_originSpecified parts of peritoneum
ethnicitynot hispanic or latino
tissue_or_organ_of_originBone marrow
LINC01235
BCHE
EFEMP1
tissue_typeNormal
AC104211.1
days_to_last_follow_up
CST1
LINC01344
RPS4Y1
LTB
ajcc_pathologic_tT1b
tissue_or_organ_of_originLymph nodes of head, face and neck
tissue_or_organ_of_originBreast, NOS
−1
0
1
2
3
Coefficient Value
🟥 Negative
🟦 Positive

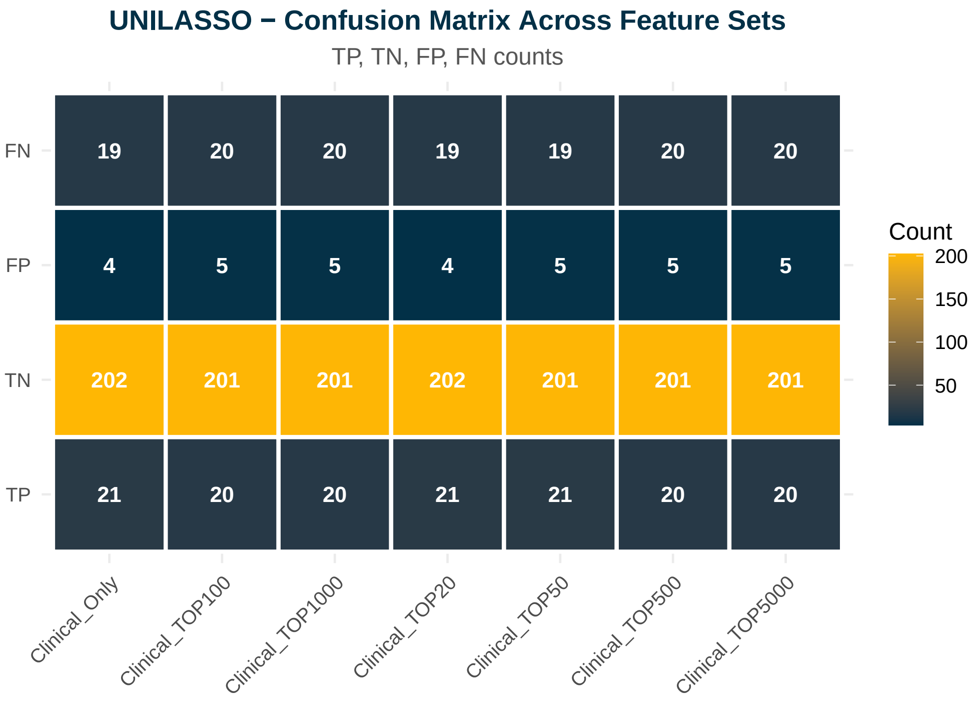

UNILASSO − Clinical_TOP100
Top 19 non−zero features
tissue_or_organ_of_originLower−inner quadrant of breast
tissue_or_organ_of_originUpper−outer quadrant of breast
ajcc_pathologic_tT2b
follow_ups_disease_responseWT−With Tumor
tissue_or_organ_of_originSpecified parts of peritoneum
ethnicitynot hispanic or latino
tissue_or_organ_of_originBone marrow
LINC01235
BCHE
tissue_typeNormal
EFEMP1
AC104211.1
tissue_typeTumor
TNFRSF18
SNORD104
CST1
LTB
ajcc_pathologic_tT1b
tissue_or_organ_of_originBreast, NOS
−1
0
1
2
3
Coefficient Value
🟥 Negative
🟦 Positive

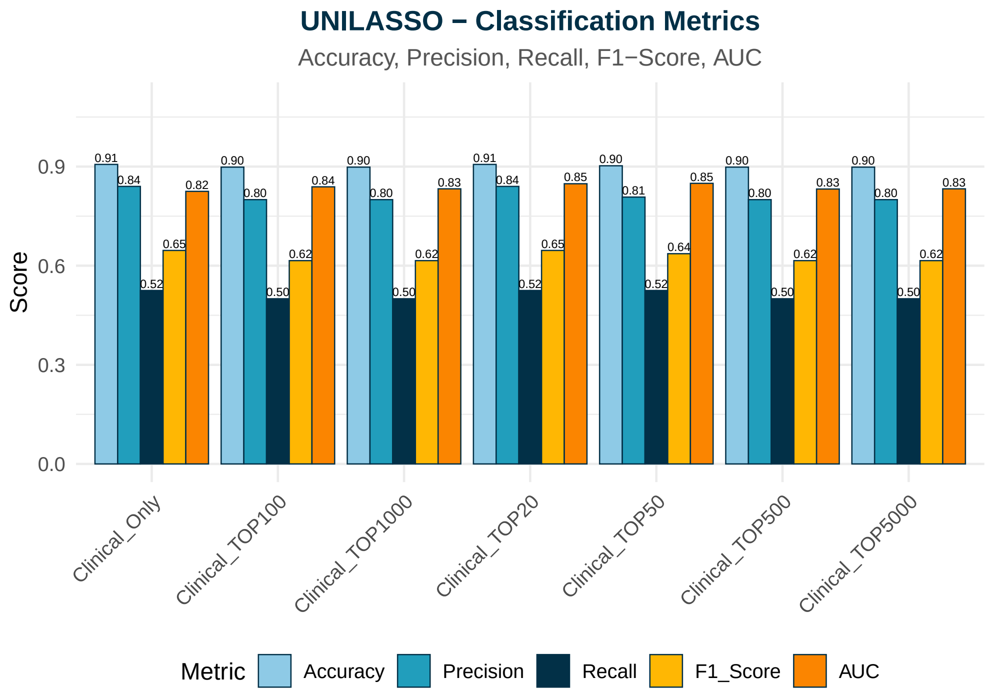

UNILASSO − Clinical_TOP50
Top 17 non−zero features
tissue_or_organ_of_originLower−inner quadrant of breast
tissue_or_organ_of_originUpper−outer quadrant of breast
follow_ups_disease_responseWT−With Tumor
ajcc_pathologic_tT2b
tissue_or_organ_of_originSpecified parts of peritoneum
ethnicitynot hispanic or latino
tissue_typeNormal
tissue_or_organ_of_originBone marrow
LINC01235
BCHE
AC104211.1
tissue_typeTumor
CST1
IL4I1
SNORD104
ajcc_pathologic_tT1b
tissue_or_organ_of_originBreast, NOS
−1
0
1
2
3
Coefficient Value
🟥 Negative
🟦 Positive

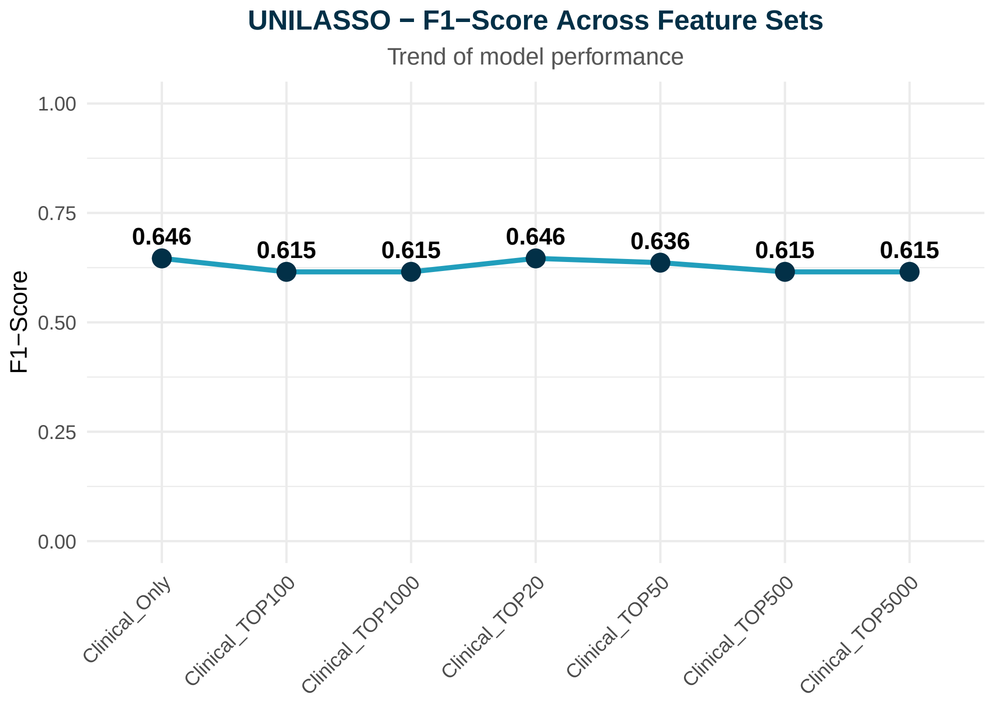


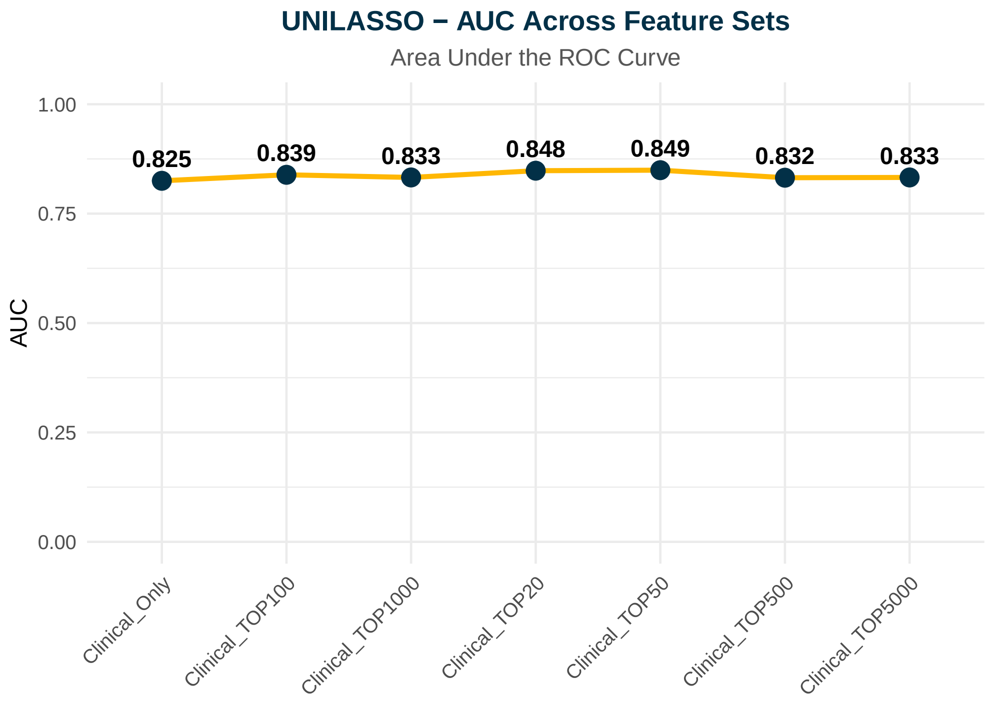
```
unilasso_metrics <- plot_classification_metrics_single(unilasso_results
                                                       , threshold = 0.5
                                                       , csv_filename = "unilasso_classification_metric
```

```
##
## === CLASSIFICATION METRICS ===

## Setting levels: control = 0, case = 1
## Setting direction: controls < cases

## Clinical_Only:
##   TP=21 TN=202 FP=4 FN=19
##   Accuracy=0.907 Precision=0.840 Recall=0.525 F1=0.646 AUC=0.825

## Setting levels: control = 0, case = 1
## Setting direction: controls < cases

## Clinical_TOP5000:
##   TP=20 TN=201 FP=5 FN=20
##   Accuracy=0.898 Precision=0.800 Recall=0.500 F1=0.615 AUC=0.833

## Setting levels: control = 0, case = 1
## Setting direction: controls < cases

## Clinical_TOP1000:
##   TP=20 TN=201 FP=5 FN=20
##   Accuracy=0.898 Precision=0.800 Recall=0.500 F1=0.615 AUC=0.833
## Setting levels: control = 0, case = 1
## Setting direction: controls < cases

## Clinical_TOP500:
##   TP=20 TN=201 FP=5 FN=20
##   Accuracy=0.898 Precision=0.800 Recall=0.500 F1=0.615 AUC=0.832

## Setting levels: control = 0, case = 1
## Setting direction: controls < cases

## Clinical_TOP100:
##   TP=20 TN=201 FP=5 FN=20
##   Accuracy=0.898 Precision=0.800 Recall=0.500 F1=0.615 AUC=0.839

## Setting levels: control = 0, case = 1
## Setting direction: controls < cases

## Clinical_TOP50:
##   TP=21 TN=201 FP=5 FN=19
##   Accuracy=0.902 Precision=0.808 Recall=0.525 F1=0.636 AUC=0.849

## Setting levels: control = 0, case = 1
## Setting direction: controls < cases

## Clinical_TOP20:
##   TP=21 TN=202 FP=4 FN=19
##   Accuracy=0.907 Precision=0.840 Recall=0.525 F1=0.646 AUC=0.848
```

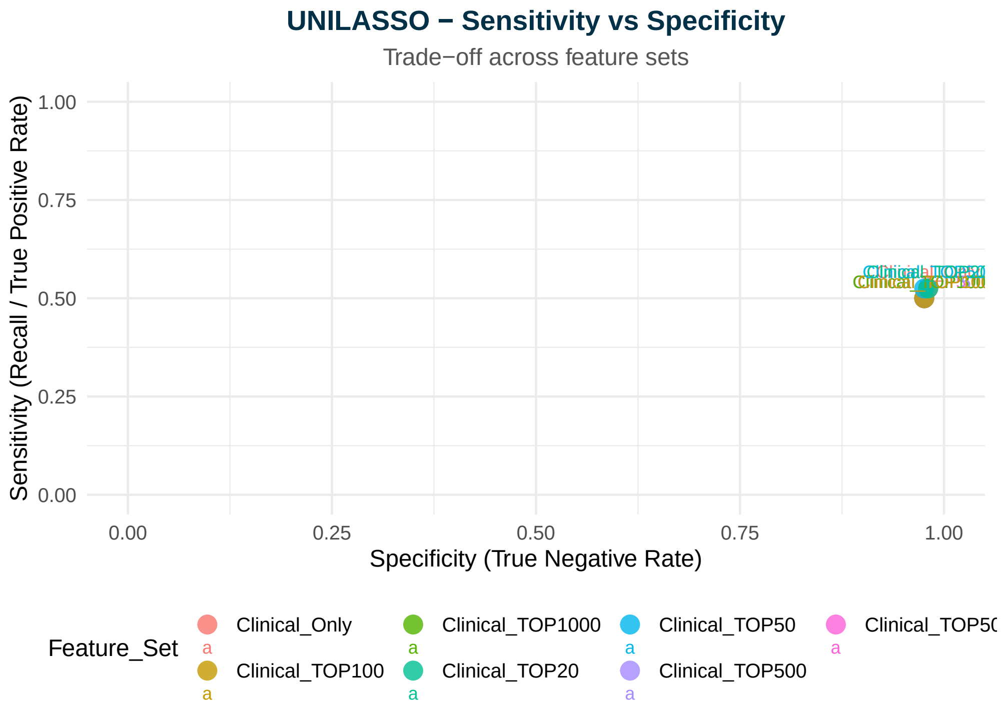



UNILASSO − Classification Metrics
Accuracy, Precision, Recall, F1−Score, AUC
Score
0.0
0.3
0.6
0.9
0.91
0.84
0.52
0.65
0.82
0.90
0.80
0.50
0.62
0.84
0.90
0.80
0.50
0.62
0.83
0.91
0.84
0.52
0.65
0.85
0.90
0.81
0.52
0.64
0.85
0.90
0.80
0.50
0.62
0.83
0.90
0.80
0.50
0.62
0.83
Clinical_Only
Clinical_TOP100
Clinical_TOP1000
Clinical_TOP20
Clinical_TOP50
Clinical_TOP500
Clinical_TOP5000
Metric
Accuracy
Precision
Recall
F1_Score
AUC


UNILASSO − F1−Score Across Feature Sets
Trend of model performance
F1−Score
0.00
0.25
0.50
0.75
1.00
0.646
0.615
0.615
0.646
0.636
0.615
0.615
Clinical_Only
Clinical_TOP100
Clinical_TOP1000
Clinical_TOP20
Clinical_TOP50
Clinical_TOP500
Clinical_TOP5000


UNILASSO − AUC Across Feature Sets

Area Under the ROC Curve


UNILASSO − Sensitivity vs Specificity

Trade−off across feature sets

```
##
## === SUMMARY TABLE ===
##        Feature_Set TP  TN FP FN  Accuracy Precision Recall Specificity
## 1    Clinical_Only 21 202  4 19 0.9065041 0.8400000  0.525   0.9805825
## 2 Clinical_TOP5000 20 201  5 20 0.8983740 0.8000000  0.500   0.9757282
## 3 Clinical_TOP1000 20 201  5 20 0.8983740 0.8000000  0.500   0.9757282
## 4  Clinical_TOP500 20 201  5 20 0.8983740 0.8000000  0.500   0.9757282
## 5  Clinical_TOP100 20 201  5 20 0.8983740 0.8000000  0.500   0.9757282
## 6   Clinical_TOP50 21 201  5 19 0.9024390 0.8076923  0.525   0.9757282
## 7   Clinical_TOP20 21 202  4 19 0.9065041 0.8400000  0.525   0.9805825
##    F1_Score       AUC
## 1 0.6461538 0.8250000
## 2 0.6153846 0.8326456
## 3 0.6153846 0.8326456
## 4 0.6153846 0.8320388
## 5 0.6153846 0.8387136
## 6 0.6363636 0.8492718
## 7 0.6461538 0.8480583
##
## Exported classification metrics to: model_metrics/unilasso_classification_metrics.csv
```

uniLasso shows extremely stable performance across all genomic feature sets, with Test AUC consistently around 0.83–0.85 and recall values near 0.50 for nearly all models. Precision remains high (0.79–0.81), and specificity stays above 0.97. Unlike Ridge, uniLasso never collapses under high dimensionality; it consistently identifies the same core signal even when starting from 5000 genes. This reflects the intended effect of the uniLasso procedure (Chatterjee, Hastie & Tibshirani, 2025), where univariate guidance stabilizes variable selection and enforces sign consistency. The clinical-only model performs poorly because uniLasso relies on univariate ranking across many features, which is only meaningful in the genomic setting.

ElasticNet Comparison Across Feature Sets

Elastic Net combines the strengths of both Ridge (L2) and Lasso (L1) penalties, making it well-suited for datasets with correlated gene groups, which is typical in transcriptomic data. Its estimator solves:

$$\hat{\beta} = argmin_{\beta}\{-l(\beta) + \lambda(\alpha\|\beta\|_1) + (1-\alpha)\|\beta\|_2^2\}$$

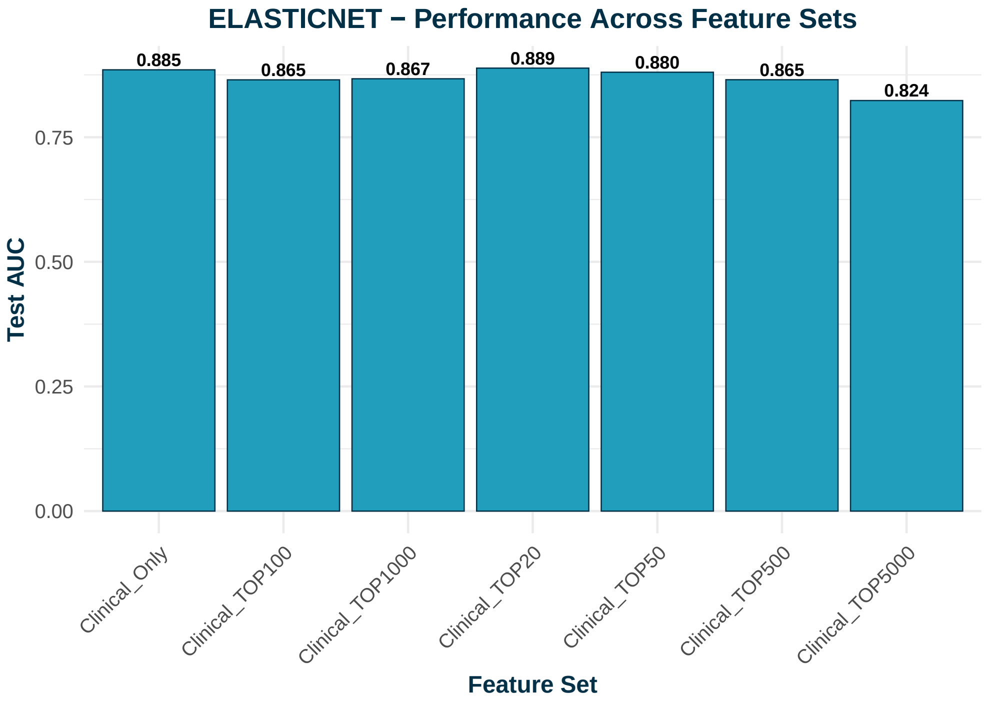
Where

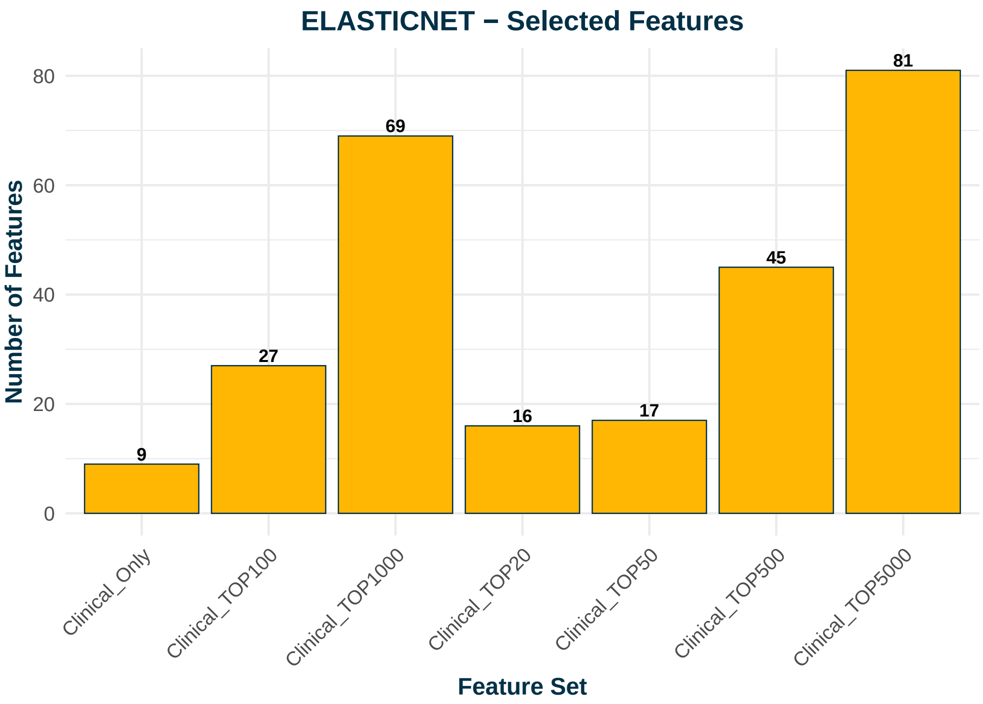
- $\alpha = 1$ is Lasso
- $\alpha = 0$ is Ridge
- $0 < \alpha < 1$ is Elastic Mixed model

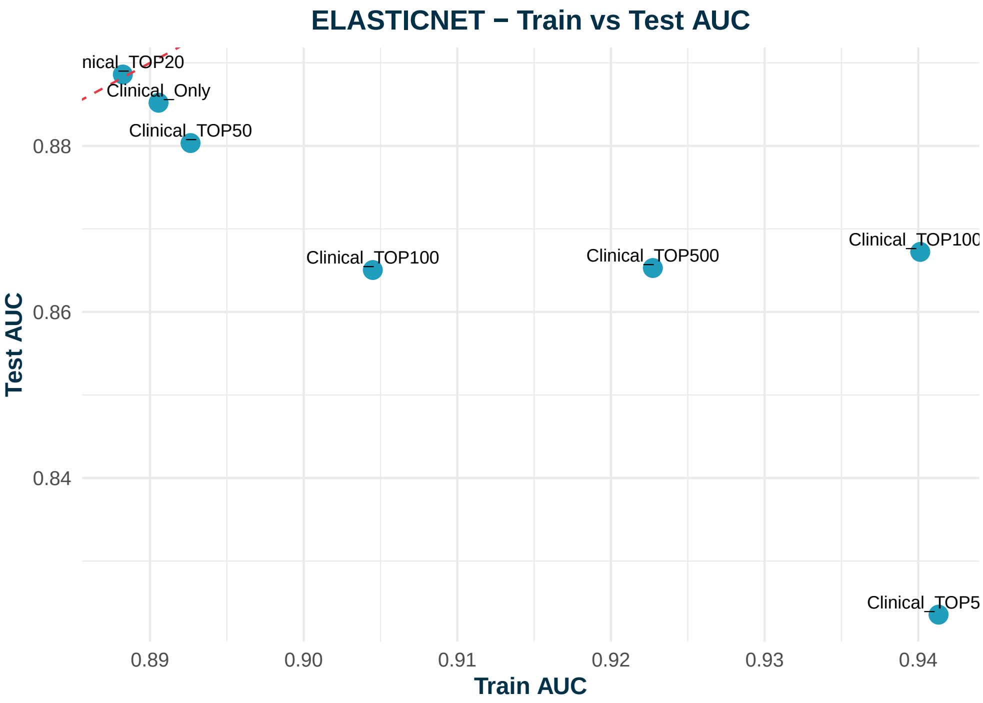
```r
elasticnet_results <- fit_single_model_across_features(
    model_type = "elasticnet"
    , X_train_all = X_train
    , X_test_all = X_test
    , Y_train = Y_train
    , Y_test = Y_test
    , n_clinical = n_clinical
    , top_genes_ranked = top_genes
    , gene_sets = c(5000, 1000, 500, 100, 50, 20)
)
```

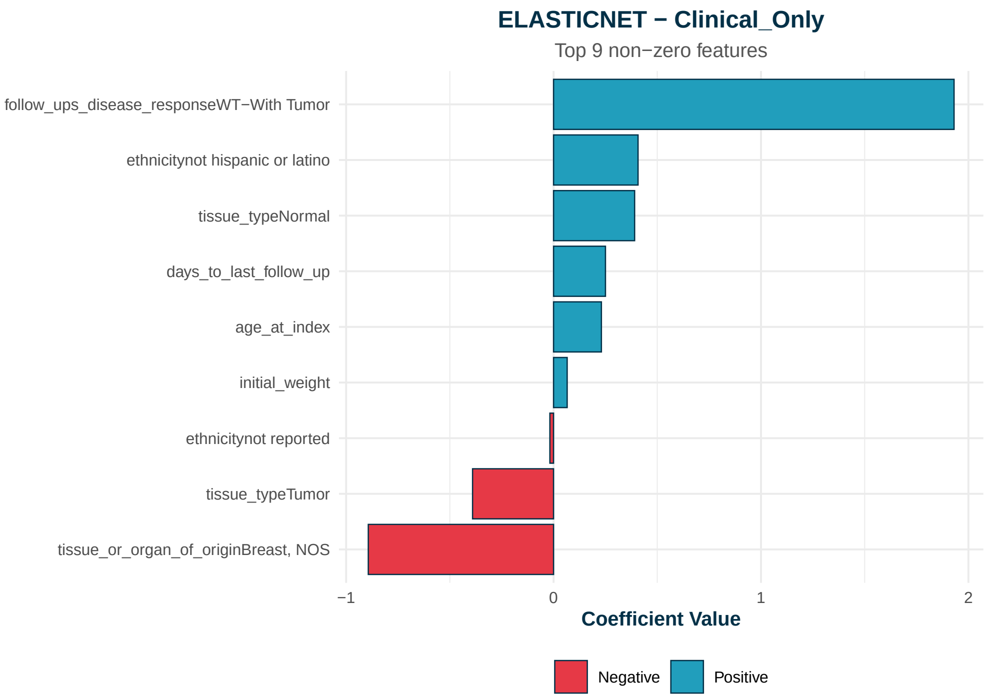
```
##
## === FITTING ELASTICNET ACROSS FEATURE SETS ===
##
## Fitting Clinical_Only...
```

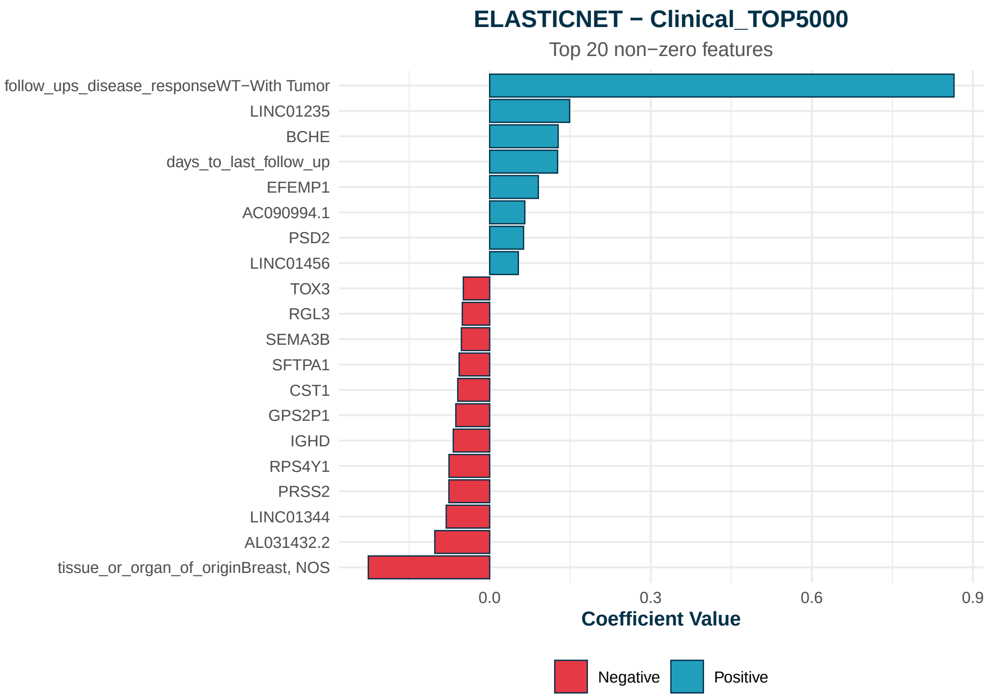
```
## Setting levels: control = 0, case = 1
## Setting direction: controls < cases

## Setting levels: control = 0, case = 1

## Setting direction: controls < cases

## Fitting Clinical_TOP5000...

## Setting levels: control = 0, case = 1
## Setting direction: controls < cases

## Setting levels: control = 0, case = 1

## Setting direction: controls < cases

## Fitting Clinical_TOP1000...

## Setting levels: control = 0, case = 1
## Setting direction: controls < cases

## Setting levels: control = 0, case = 1

## Setting direction: controls < cases

## Fitting Clinical_TOP500...

## Setting levels: control = 0, case = 1
## Setting direction: controls < cases

## Setting levels: control = 0, case = 1

## Setting direction: controls < cases

## Fitting Clinical_TOP100...

## Setting levels: control = 0, case = 1
## Setting direction: controls < cases

## Setting levels: control = 0, case = 1

## Setting direction: controls < cases

## Fitting Clinical_TOP50...

## Setting levels: control = 0, case = 1
## Setting direction: controls < cases

## Setting levels: control = 0, case = 1
## Setting direction: controls < cases

## Fitting Clinical_TOP20...

## Setting levels: control = 0, case = 1
## Setting direction: controls < cases

## Setting levels: control = 0, case = 1

## Setting direction: controls < cases

##
## === SUMMARY TABLE ===
##        Feature_Set      Model Features Train_AUC  Test_AUC Test_Accuracy
## 1    Clinical_Only ELASTICNET        9 0.8905534 0.8851942     0.8658537
## 2 Clinical_TOP5000 ELASTICNET       81 0.9413221 0.8235437     0.8414634
## 3 Clinical_TOP1000 ELASTICNET       69 0.9401297 0.8672330     0.8536585
## 4  Clinical_TOP500 ELASTICNET       45 0.9227263 0.8652913     0.8699187
## 5  Clinical_TOP100 ELASTICNET       27 0.9045003 0.8650485     0.8821138
## 6   Clinical_TOP50 ELASTICNET       17 0.8926364 0.8803398     0.8780488
## 7   Clinical_TOP20 ELASTICNET       16 0.8882289 0.8885922     0.8861789
## Exported metrics to: model_metrics/elasticnet_across_features_metrics.csv
```

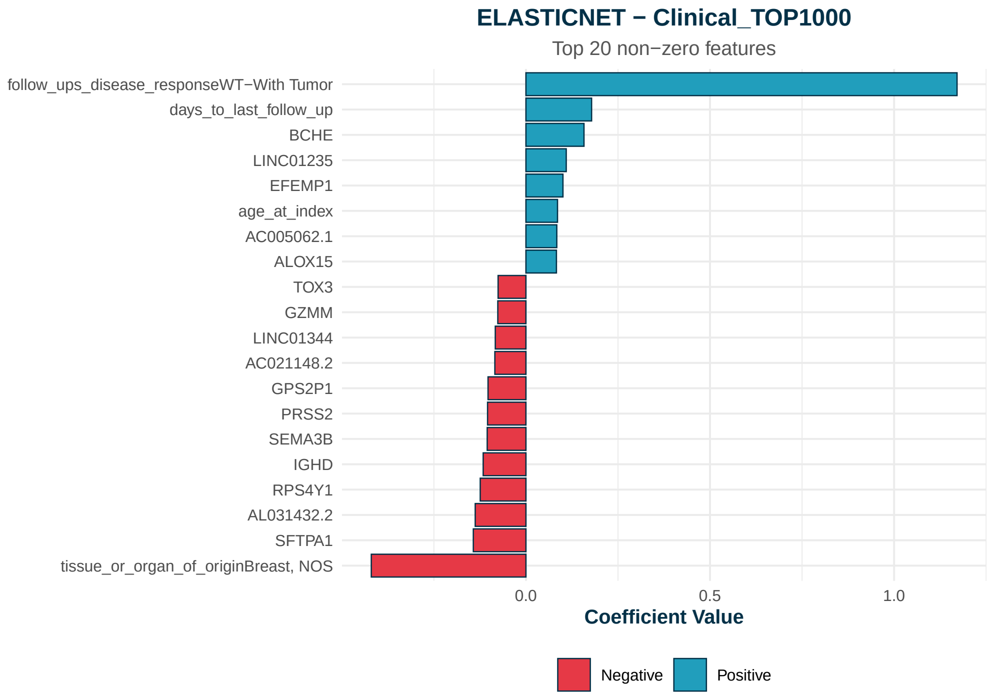


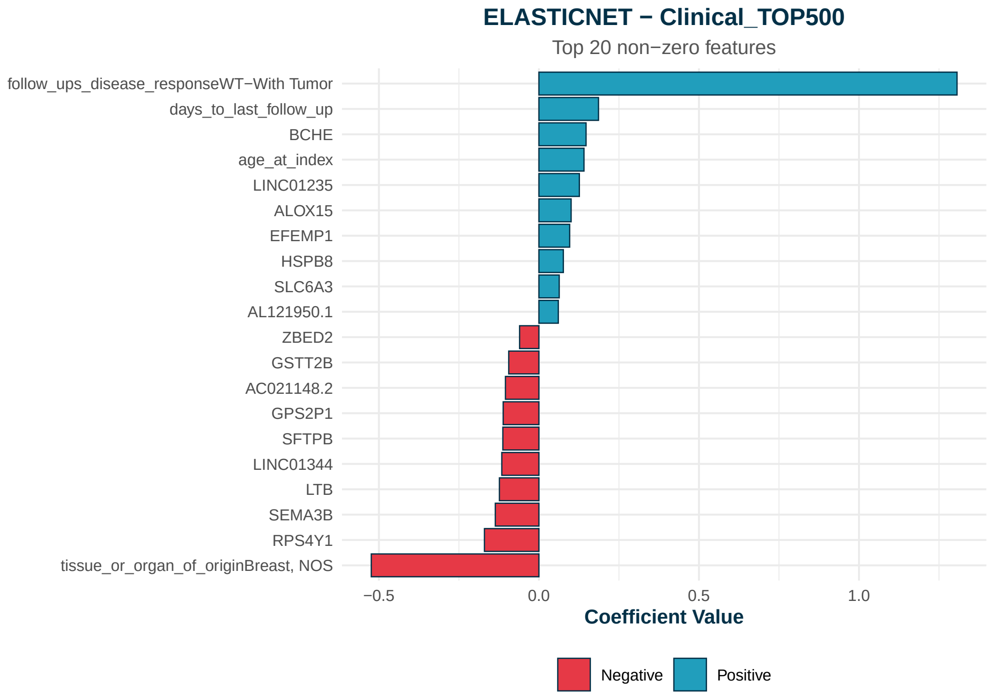

ELASTICNET − Selected Features
Number of Features
80
60
40
20
0
9
27
69
16
17
45
81
Clinical_Only
Clinical_TOP100
Clinical_TOP1000
Clinical_TOP20
Clinical_TOP50
Clinical_TOP500
Clinical_TOP5000
Feature Set

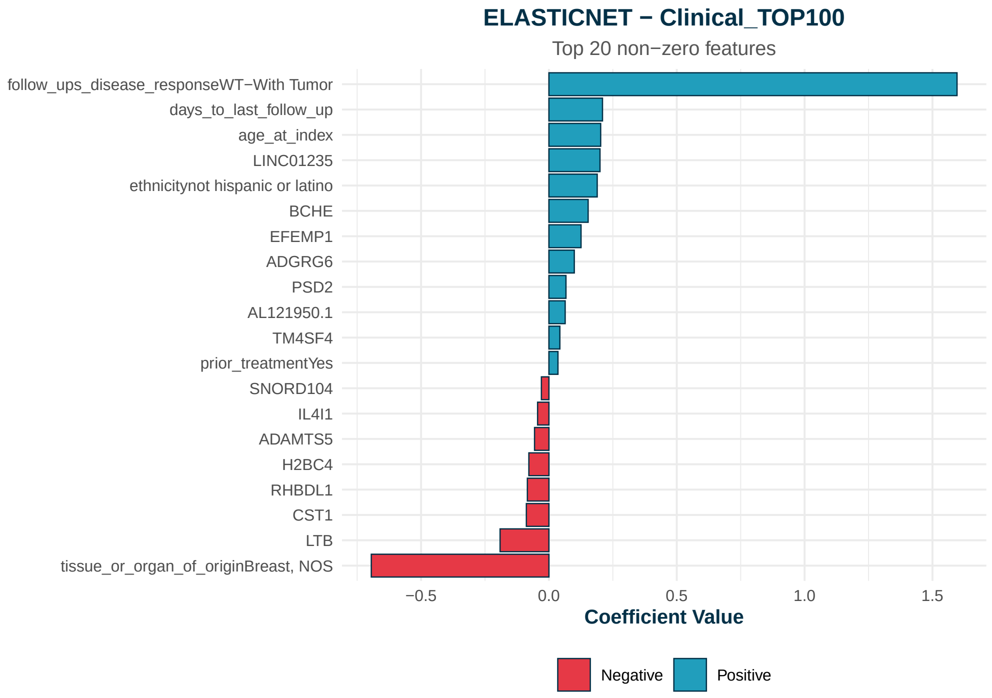

ELASTICNET − Train vs Test AUC
Test AUC
0.88
0.86
0.84
nical_TOP20
Clinical_Only
Clinical_TOP50
Clinical_TOP100
Clinical_TOP500
Clinical_TOP100
Clinical_TOP5
0.89
0.90
0.91
0.92
0.93
0.94
Train AUC

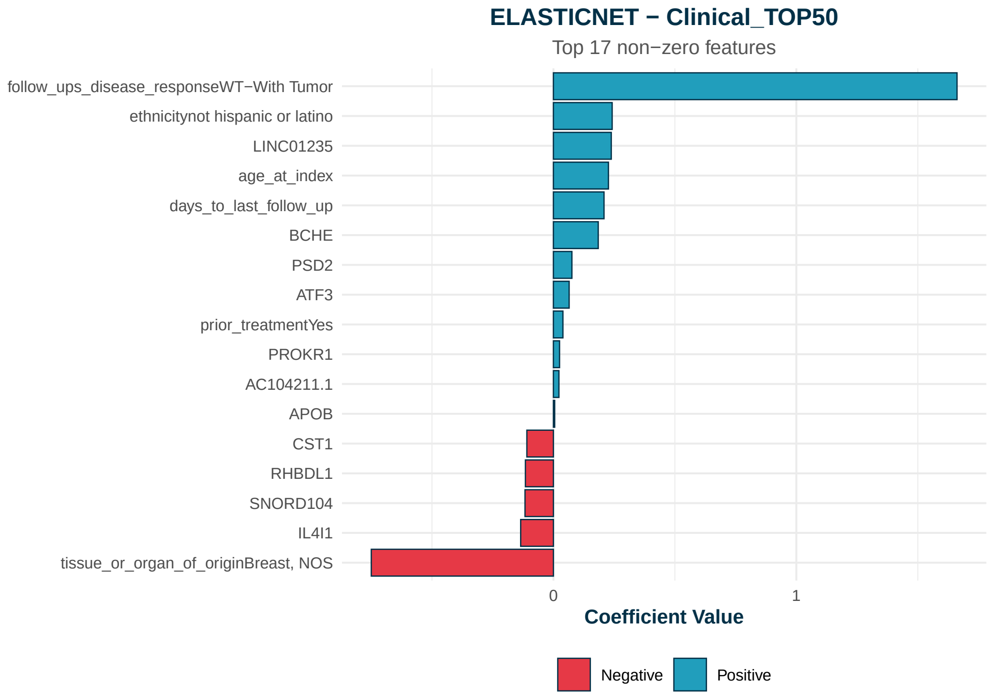

ELASTICNET − Clinical_Only
Top 9 non−zero features
follow_ups_disease_responseWT−With Tumor
ethnicitynot hispanic or latino
tissue_typeNormal
days_to_last_follow_up
age_at_index
initial_weight
ethnicitynot reported
tissue_typeTumor
tissue_or_organ_of_originBreast, NOS
−1
0
1
2
Coefficient Value
Negative
Positive

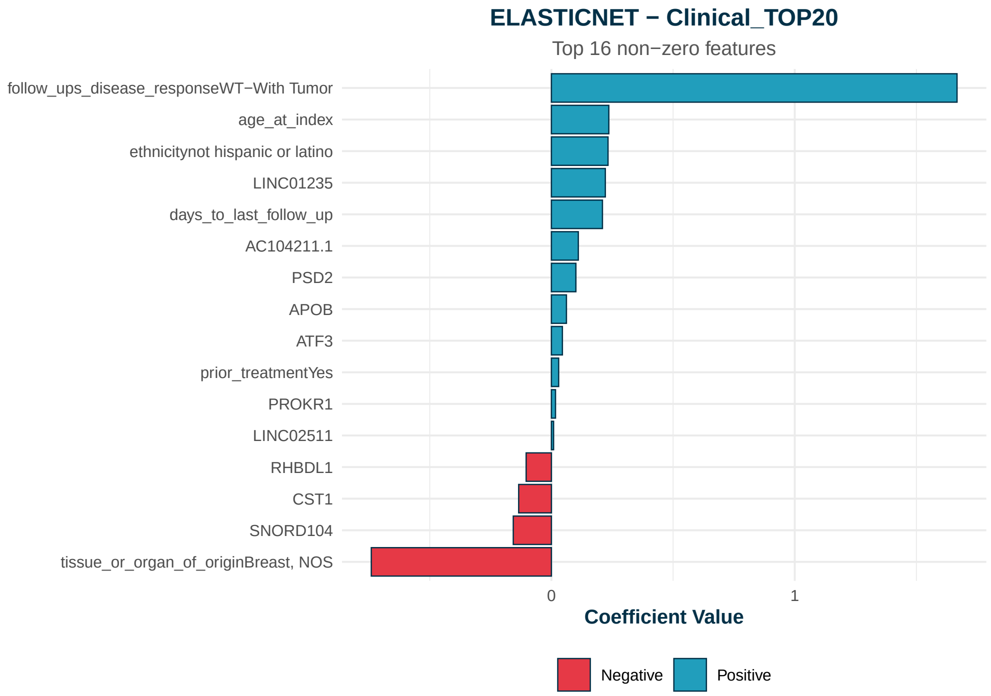

ELASTICNET − Clinical_TOP5000
Top 20 non−zero features
follow_ups_disease_responseWT−With Tumor
LINC01235
BCHE
days_to_last_follow_up
EFEMP1
AC090994.1
PSD2
LINC01456
TOX3
RGL3
SEMA3B
SFTPA1
CST1
GPS2P1
IGHD
RPS4Y1
PRSS2
LINC01344
AL031432.2
tissue_or_organ_of_originBreast, NOS
0.0
0.3
0.6
0.9
Coefficient Value
Negative
Positive


ELASTICNET − Clinical_TOP1000
Top 20 non−zero features
follow_ups_disease_responseWT−With Tumor
days_to_last_follow_up
BCHE
LINC01235
EFEMP1
age_at_index
AC005062.1
ALOX15
TOX3
GZMM
LINC01344
AC021148.2
GPS2P1
PRSS2
SEMA3B
IGHD
RPS4Y1
AL031432.2
SFTPA1
tissue_or_organ_of_originBreast, NOS
0.0
0.5
1.0
Coefficient Value
Negative
Positive


ELASTICNET − Clinical_TOP500
Top 20 non−zero features
follow_ups_disease_responseWT−With Tumor
days_to_last_follow_up
BCHE
age_at_index
LINC01235
ALOX15
EFEMP1
HSPB8
SLC6A3
AL121950.1
ZBED2
GSTT2B
AC021148.2
GPS2P1
SFTPB
LINC01344
LTB
SEMA3B
RPS4Y1
tissue_or_organ_of_originBreast, NOS
−0.5
0.0
0.5
1.0
Coefficient Value
Negative
Positive

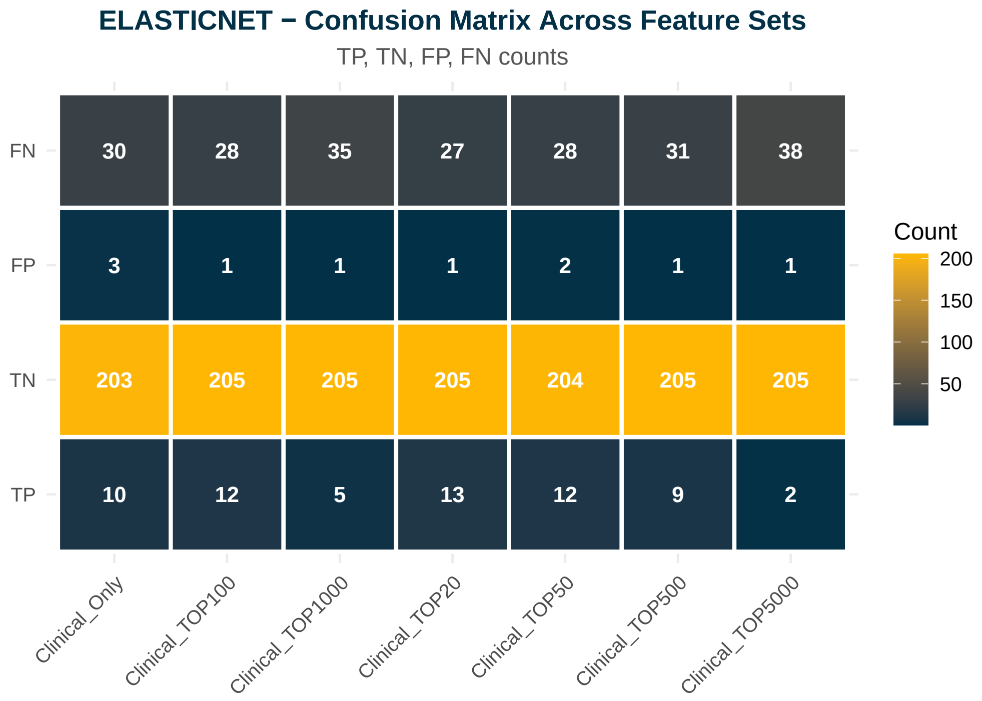

ELASTICNET − Clinical_TOP100
Top 20 non−zero features
follow_ups_disease_responseWT−With Tumor
days_to_last_follow_up
age_at_index
LINC01235
ethnicitynot hispanic or latino
BCHE
EFEMP1
ADGRG6
PSD2
AL121950.1
TM4SF4
prior_treatmentYes
SNORD104
IL4I1
ADAMTS5
H2BC4
RHBDL1
CST1
LTB
tissue_or_organ_of_originBreast, NOS
−0.5
0.0
0.5
1.0
1.5
Coefficient Value
Negative
Positive

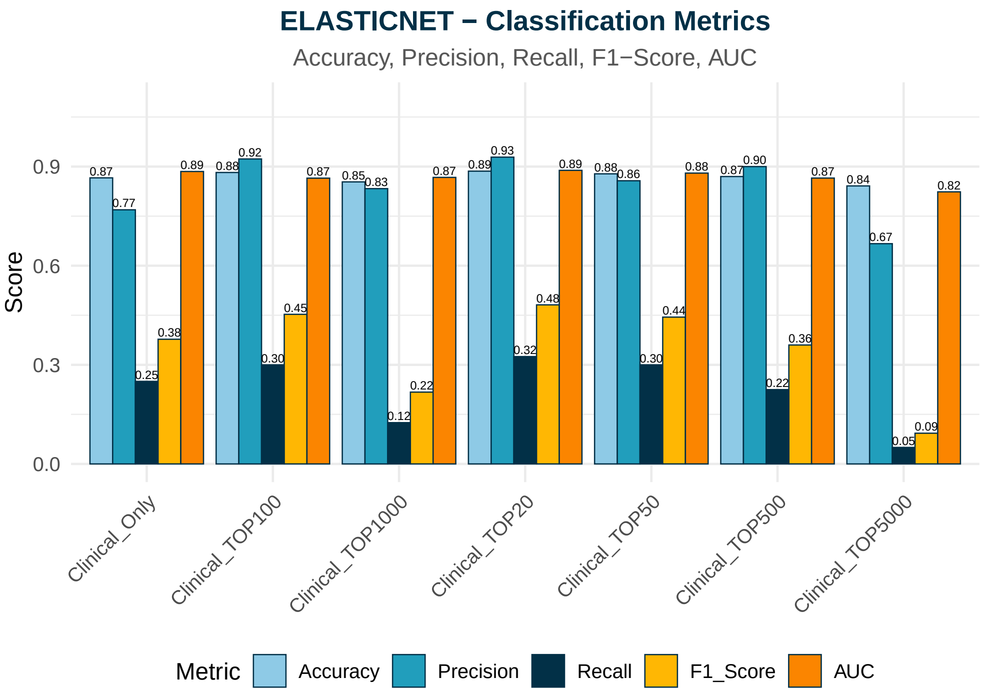

ELASTICNET − Clinical_TOP50
Top 17 non−zero features
follow_ups_disease_responseWT−With Tumor
ethnicitynot hispanic or latino
LINC01235
age_at_index
days_to_last_follow_up
BCHE
PSD2
ATF3
prior_treatmentYes
PROKR1
AC104211.1
APOB
CST1
RHBDL1
SNORD104
IL4I1
tissue_or_organ_of_originBreast, NOS
0
1
Coefficient Value
Negative
Positive



```
elasticnet_metrics <- plot_classification_metrics_single(elasticnet_results
                                                         , threshold = 0.5
                                                         , csv_filename = "elasticnet_classification_me
```

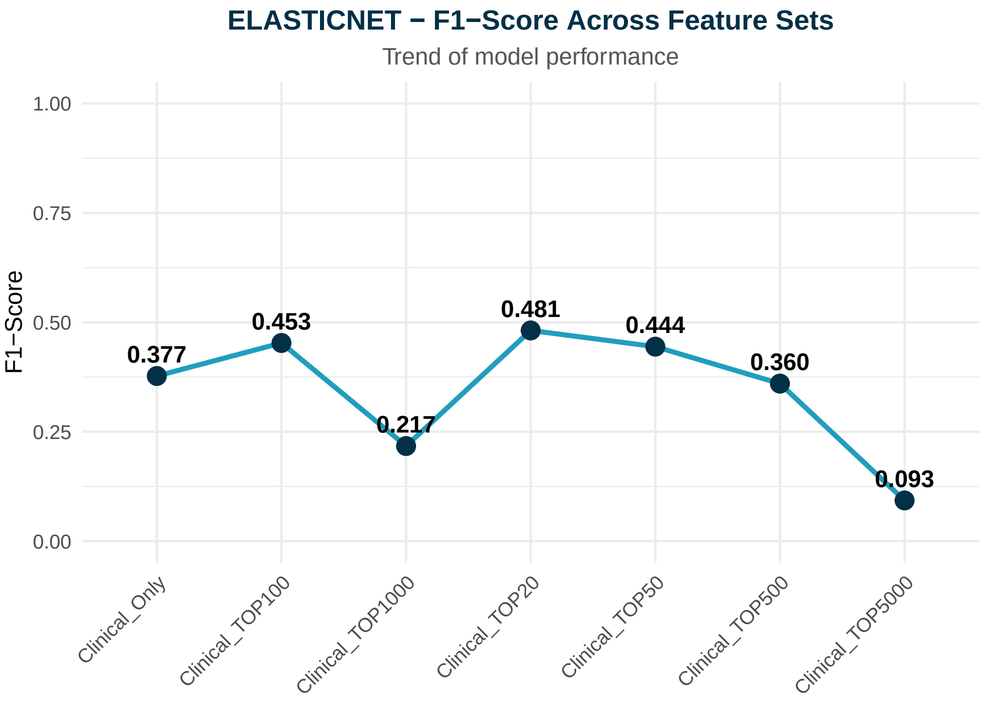
```
##
## === CLASSIFICATION METRICS ===

## Setting levels: control = 0, case = 1
## Setting direction: controls < cases

## Clinical_Only:
##   TP=10 TN=203 FP=3 FN=30
##   Accuracy=0.866 Precision=0.769 Recall=0.250 F1=0.377 AUC=0.885

## Setting levels: control = 0, case = 1
## Setting direction: controls < cases

## Clinical_TOP5000:
##   TP=2 TN=205 FP=1 FN=38
##   Accuracy=0.841 Precision=0.667 Recall=0.050 F1=0.093 AUC=0.824

## Setting levels: control = 0, case = 1
## Setting direction: controls < cases

## Clinical_TOP1000:
##   TP=5 TN=205 FP=1 FN=35
##   Accuracy=0.854 Precision=0.833 Recall=0.125 F1=0.217 AUC=0.867
## Setting levels: control = 0, case = 1
## Setting direction: controls < cases

## Clinical_TOP500:
##   TP=9 TN=205 FP=1 FN=31
##   Accuracy=0.870 Precision=0.900 Recall=0.225 F1=0.360 AUC=0.865

## Setting levels: control = 0, case = 1
## Setting direction: controls < cases

## Clinical_TOP100:
##   TP=12 TN=205 FP=1 FN=28
##   Accuracy=0.882 Precision=0.923 Recall=0.300 F1=0.453 AUC=0.865

## Setting levels: control = 0, case = 1
## Setting direction: controls < cases

## Clinical_TOP50:
##   TP=12 TN=204 FP=2 FN=28
##   Accuracy=0.878 Precision=0.857 Recall=0.300 F1=0.444 AUC=0.880

## Setting levels: control = 0, case = 1
## Setting direction: controls < cases

## Clinical_TOP20:
##   TP=13 TN=205 FP=1 FN=27
##   Accuracy=0.886 Precision=0.929 Recall=0.325 F1=0.481 AUC=0.889
```





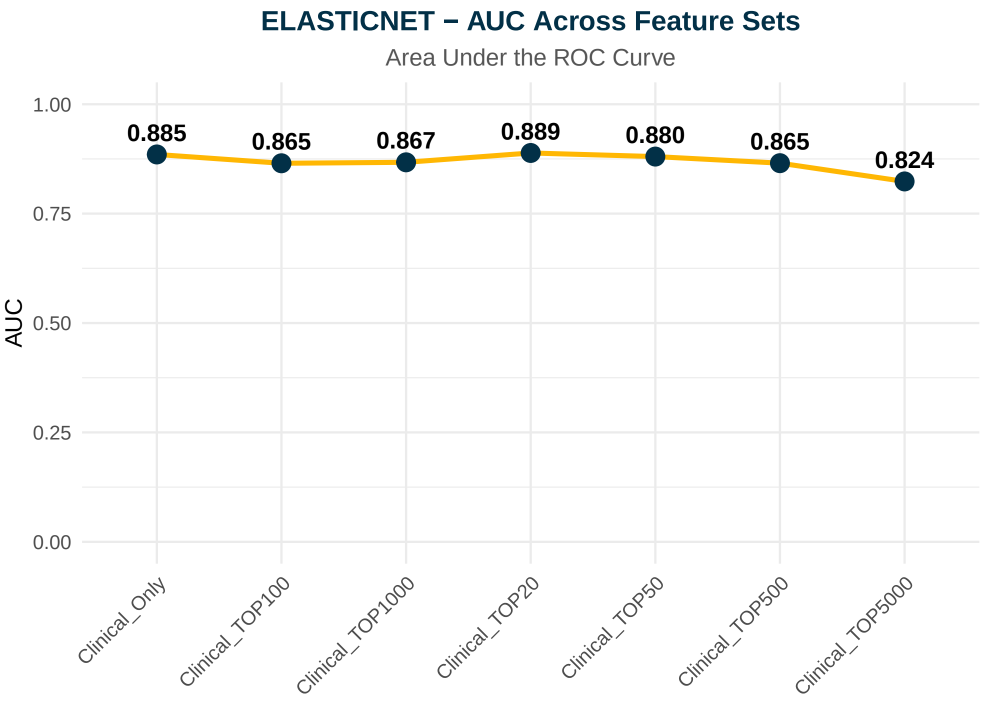
ELASTICNET − Classification Metrics

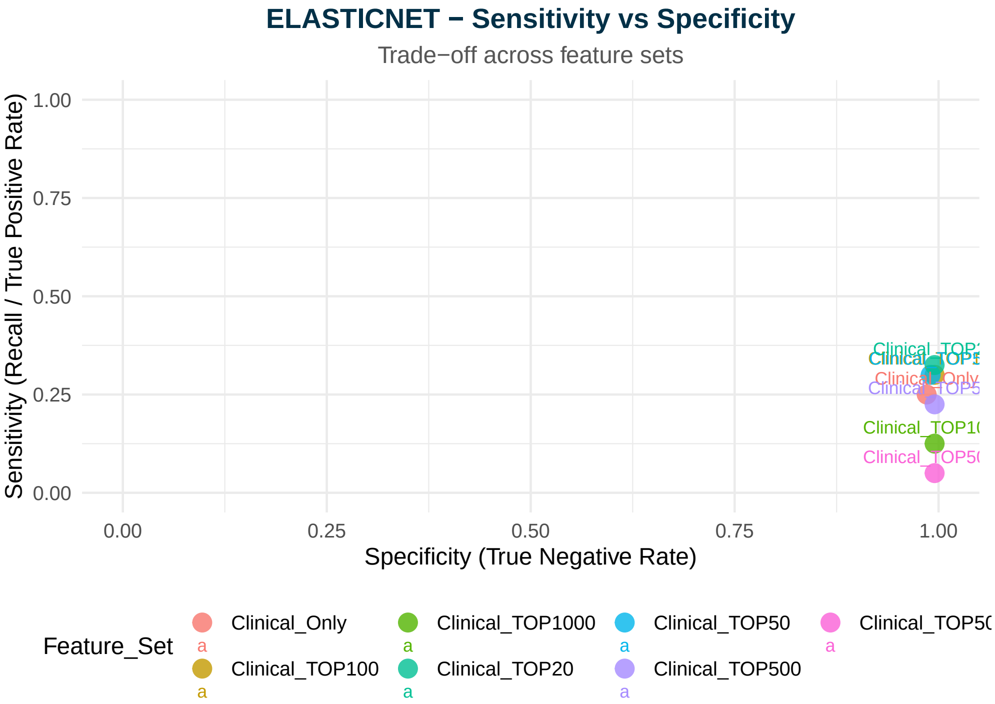
Accuracy, Precision, Recall, F1−Score, AUC



ELASTICNET − F1−Score Across Feature Sets

Trend of model performance





```
##
## === SUMMARY TABLE ===
##        Feature_Set TP  TN FP FN  Accuracy Precision Recall Specificity
## 1    Clinical_Only 10 203  3 30 0.8658537 0.7692308  0.250   0.9854369
## 2 Clinical_TOP5000  2 205  1 38 0.8414634 0.6666667  0.050   0.9951456
## 3 Clinical_TOP1000  5 205  1 35 0.8536585 0.8333333  0.125   0.9951456
## 4  Clinical_TOP500  9 205  1 31 0.8699187 0.9000000  0.225   0.9951456
## 5  Clinical_TOP100 12 205  1 28 0.8821138 0.9230769  0.300   0.9951456
## 6   Clinical_TOP50 12 204  2 28 0.8780488 0.8571429  0.300   0.9902913
## 7   Clinical_TOP20 13 205  1 27 0.8861789 0.9285714  0.325   0.9951456
##     F1_Score       AUC
## 1 0.37735849 0.8851942
## 2 0.09302326 0.8235437
## 3 0.21739130 0.8672330
## 4 0.36000000 0.8652913
## 5 0.45283019 0.8650485
## 6 0.44444444 0.8803398
## 7 0.48148148 0.8885922
##
## Exported classification metrics to: model_metrics/elasticnet_classification_metrics.csv
```

Elastic Net achieves consistently strong performance across all medium-sized gene sets, with test AUC values in the 0.86–0.89 range. It improves stability over Lasso when correlated genes are present and avoids the total collapse seen in Ridge when dimensionality is high. Precision is consistently high, while recall remains low but stable, reflecting conservative predictions in an imbalanced dataset. Elastic Net works best for gene subsets between 20 and 1000 genes, where it captures correlated genomic structure without being overwhelmed by noise.

Class Imbalance Handling with SMOTE

Current imbalance ratio: 5.12:1 (Alive:Dead)

Problem observed in baseline models: - Ridge: Recall = 0.00-0.27 (missing 73-100% of Dead patients) - Models biased toward majority class (Alive) - High Specificity (99%) but very low Recall (22%) - Clinical_TOP5000: Predicts 0 Dead patients (completely fails)

Why SMOTE: - Creates synthetic minority class samples (Dead patients) - Balances training data to ~1:1 ratio - Forces models to learn Dead patient patterns - No data loss (vs downsampling) - Prevents overfitting (vs simple upsampling)

Expected improvements: - Recall: 0.22 -> 0.50-0.70 (detect more Dead patients) - F1-Score: 0.36 -> 0.50+ (better balance) - Trade-off: Specificity may drop from 99% to 85-90% (acceptable)

Models selected for SMOTE testing: 1. Ridge - worst Recall performance, needs urgent fix 2. Lasso - feature selection sensitive to imbalance 3. ElasticNet - combination of L1/L2, middle priority

```r
cat("=== CLASS IMBALANCE ANALYSIS ===\n")
```

```
## === CLASS IMBALANCE ANALYSIS ===
```

```r
cat("Training set imbalance:\n")
```

```
## Training set imbalance:
```

```r
cat("  Alive:", sum(Y_train == 0), sprintf("(%.1f%%)\n", 100 * sum(Y_train == 0) / length(Y_train)))
```

```
##   Alive: 823 (83.6%)
```

```
cat("  Dead:", sum(Y_train == 1), sprintf("(%.1f%%)\n", 100 * sum(Y_train == 1) / length(Y_train)))
```

```
##   Dead: 161 (16.4%)
```

```
cat("  Ratio:", sprintf("%.2f:1\n\n", sum(Y_train == 0) / sum(Y_train == 1)))
```

```
##   Ratio: 5.11:1
```

```
smote_data <- apply_smote(X_train, Y_train, k = 5)
```

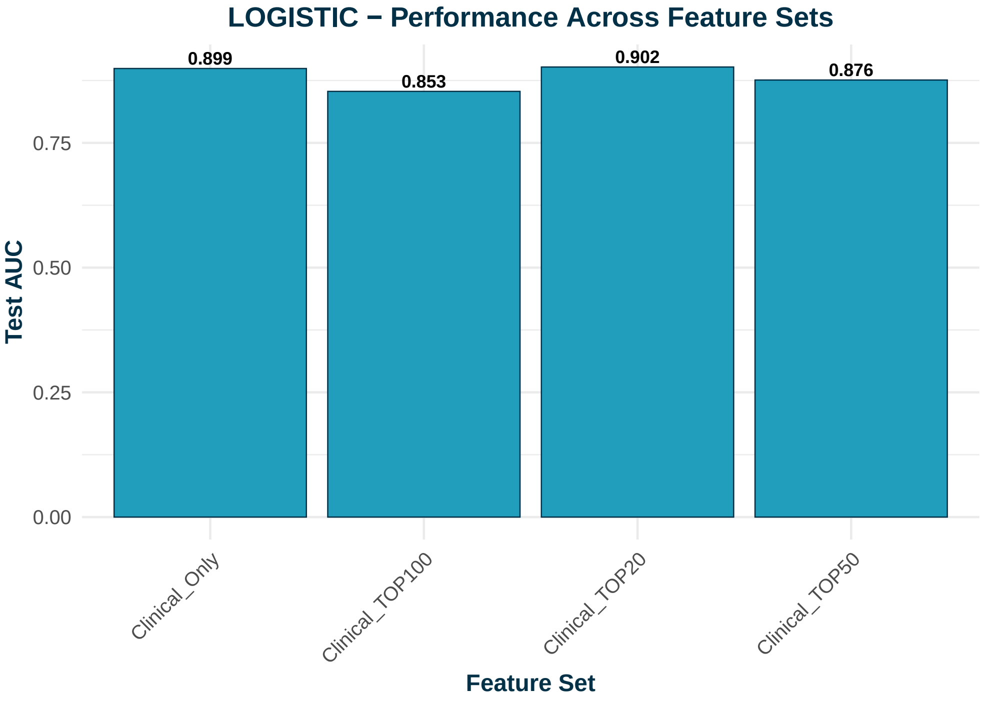
```
##
## === APPLYING SMOTE ===
## Before SMOTE:
##   Alive (0): 823
##   Dead (1): 161
##   Ratio: 5.11 :1
##
## After SMOTE:
##   Alive (0): 823
##   Dead (1): 805
##   Ratio: 1.02 :1
##   Total samples: 1628
```

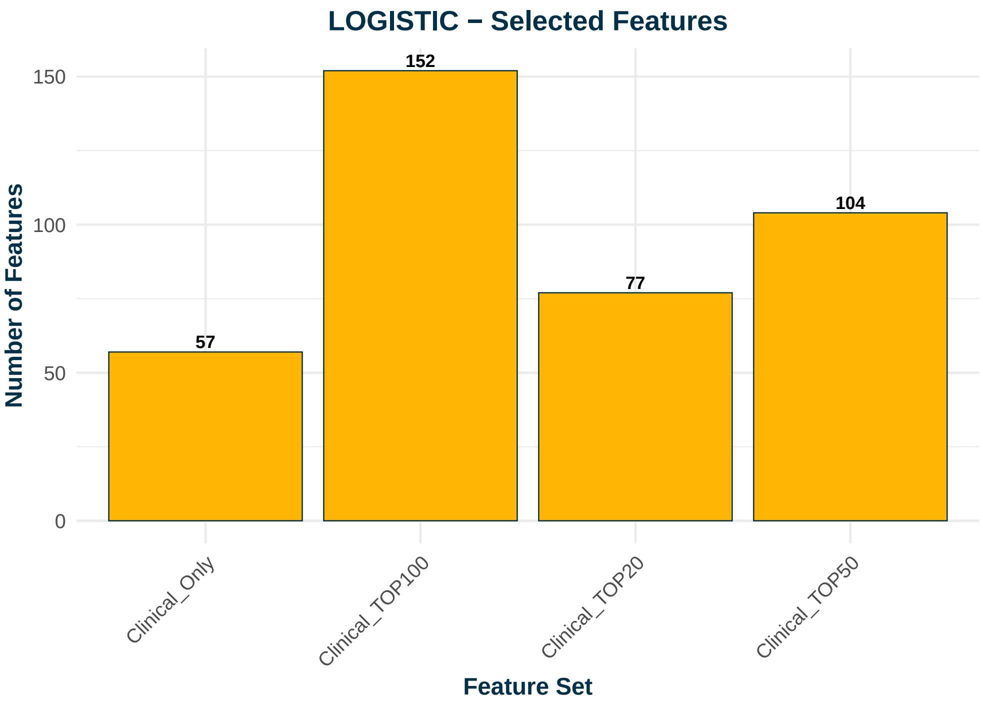
Logistic with SMOTE

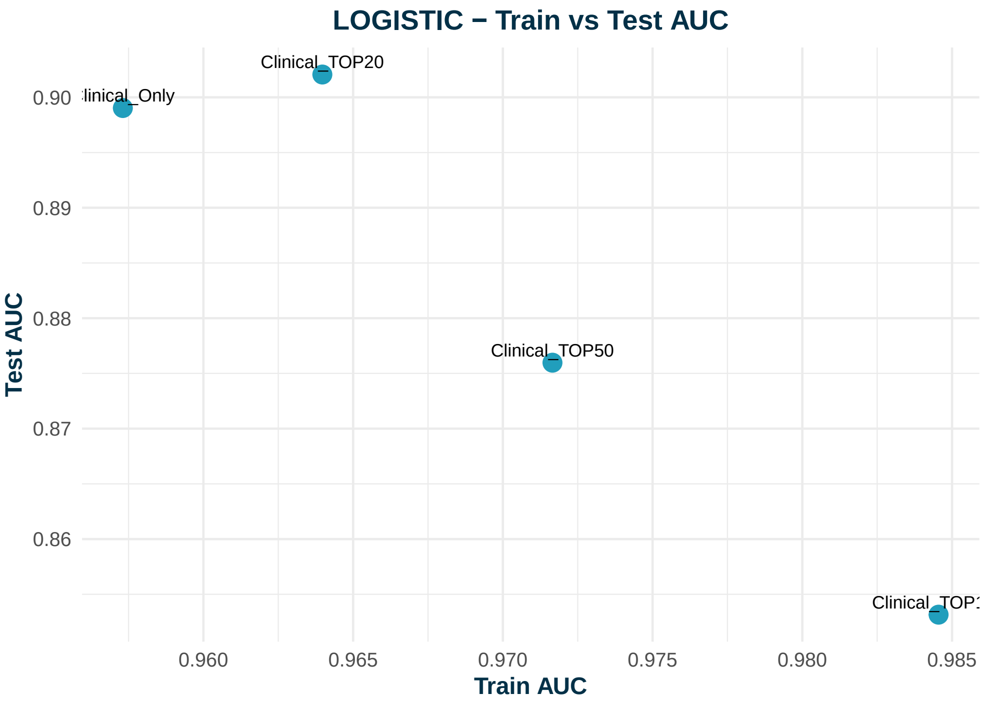
```
logistic_smote <- fit_single_model_across_features(
    model_type = "logistic"
    , X_train_all = smote_data$X_train
    , X_test_all = X_test
    , Y_train = smote_data$Y_train
    , Y_test = Y_test
    , n_clinical = n_clinical
    , top_genes_ranked = top_genes
    , gene_sets = c(100, 50, 20)
)
```

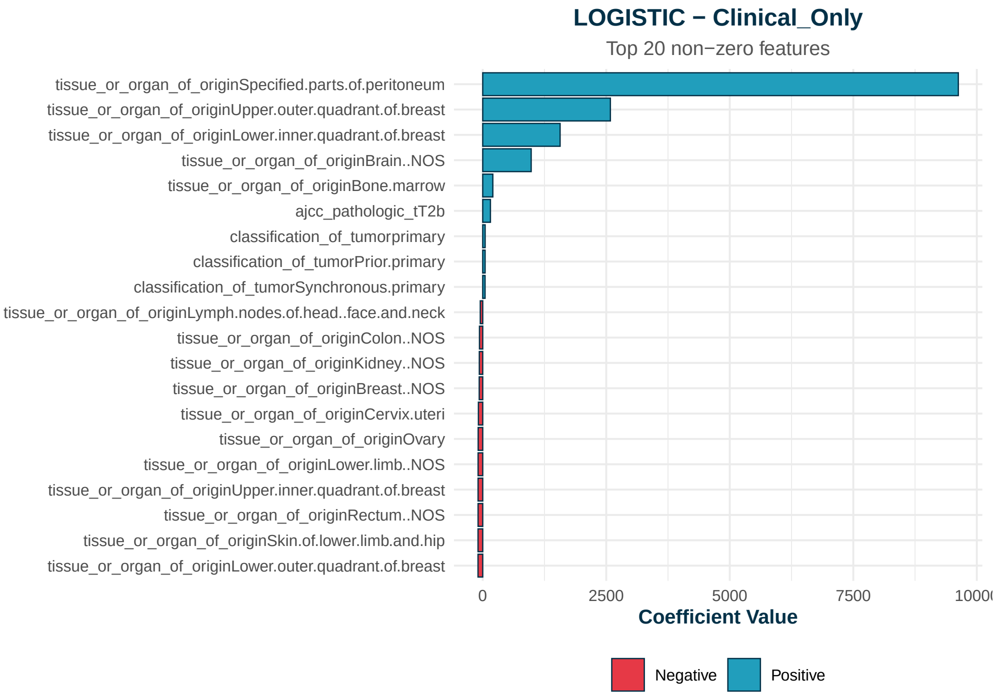
```
##
## === FITTING LOGISTIC ACROSS FEATURE SETS ===
##
## Fitting Clinical_Only...

## Fitting Clinical_TOP100...

## Fitting Clinical_TOP50...

## Fitting Clinical_TOP20...
```

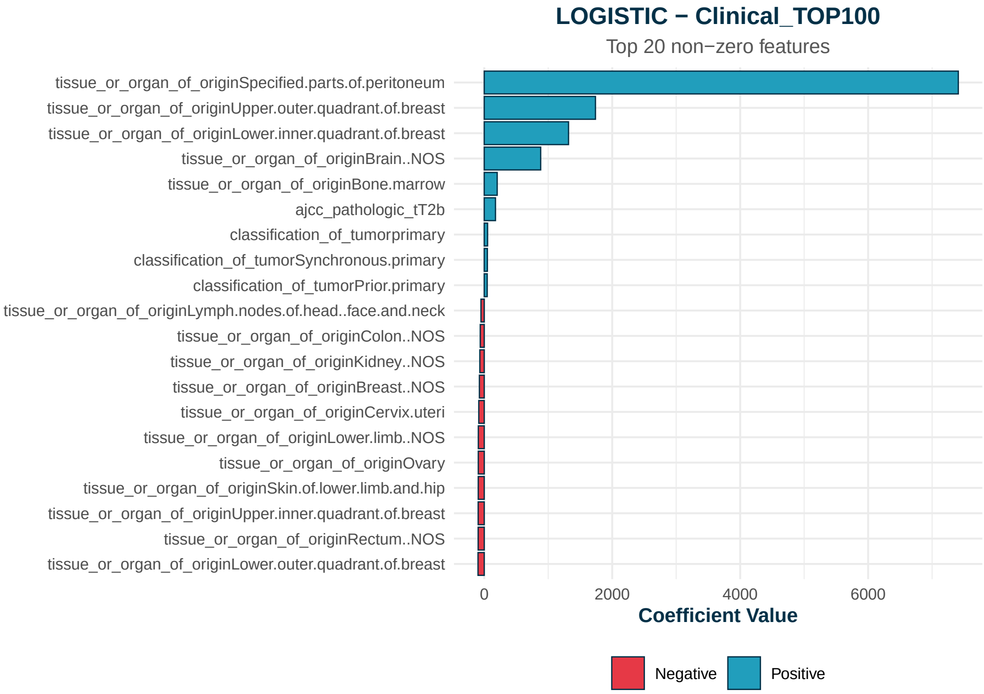
```
##
## === SUMMARY TABLE ===
##        Feature_Set    Model Features Train_AUC  Test_AUC Test_Accuracy
## 1   Clinical_Only LOGISTIC       57 0.9573096 0.8990291     0.8821138
## 2 Clinical_TOP100 LOGISTIC      152 0.9845498 0.8531553     0.8373984
## 3  Clinical_TOP50 LOGISTIC      104 0.9716565 0.8759709     0.8617886
## 4  Clinical_TOP20 LOGISTIC       77 0.9639721 0.9020631     0.8943089
## Exported metrics to: model_metrics/logistic_across_features_metrics.csv
```

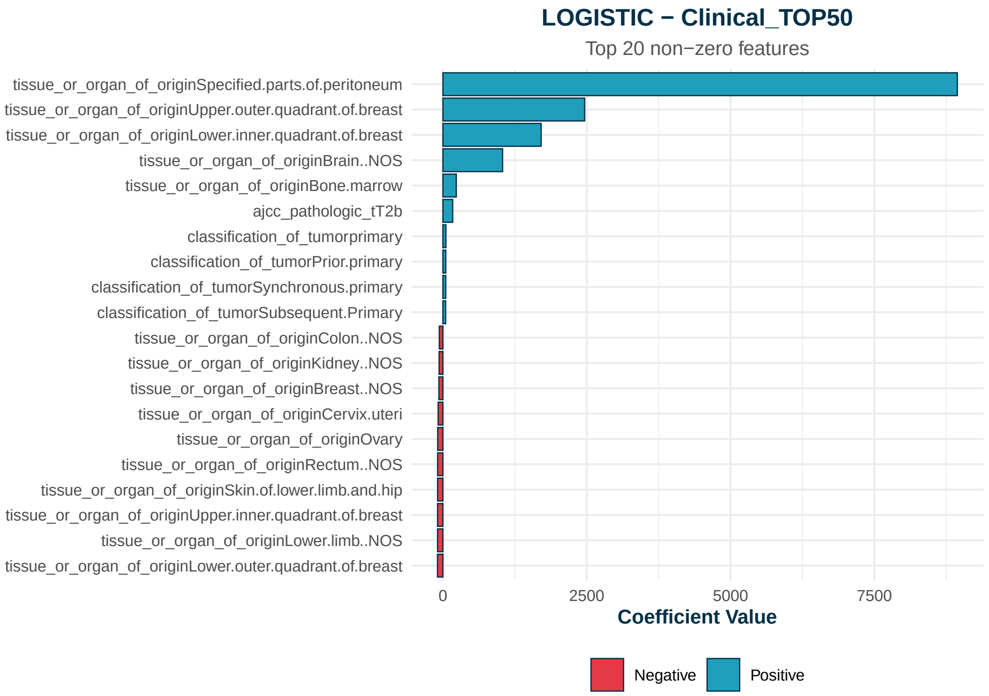


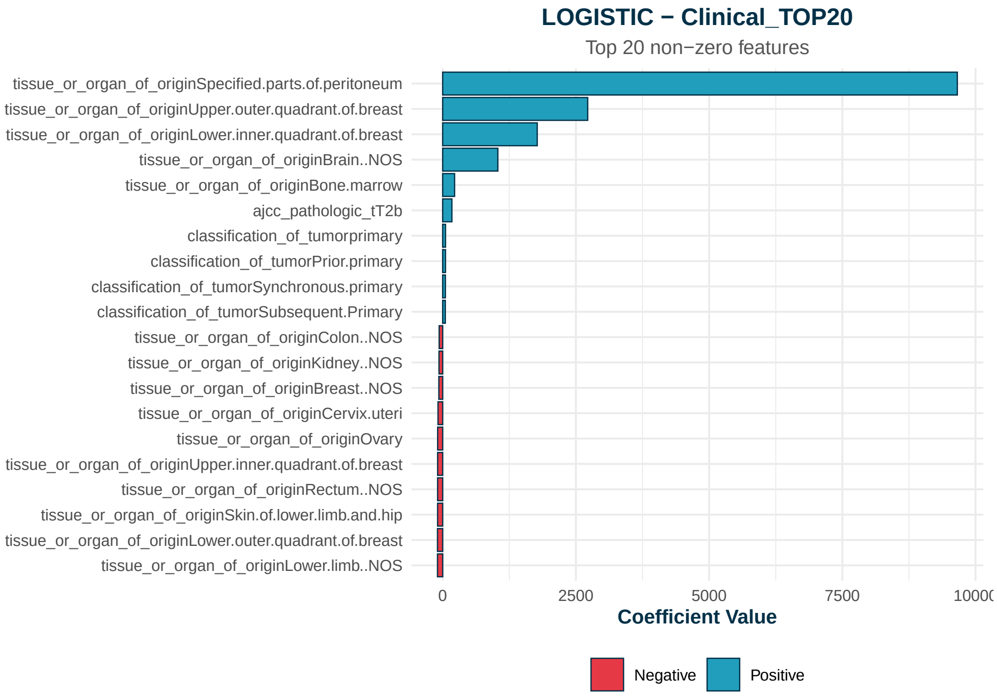

LOGISTIC − Selected Features
Number of Features
150
100
50
0
57
152
77
104
Clinical_Only
Clinical_TOP100
Clinical_TOP20
Clinical_TOP50
Feature Set


LOGISTIC − Train vs Test AUC
Test AUC
0.90
0.89
0.88
0.87
0.86
linical_Only
Clinical_TOP20
Clinical_TOP50
Clinical_TOP1
0.960
0.965
0.970
0.975
0.980
0.985
Train AUC


LOGISTIC − Clinical_Only
Top 20 non−zero features
tissue_or_organ_of_originSpecified.parts.of.peritoneum
tissue_or_organ_of_originUpper.outer.quadrant.of.breast
tissue_or_organ_of_originLower.inner.quadrant.of.breast
tissue_or_organ_of_originBrain..NOS
tissue_or_organ_of_originBone.marrow
ajcc_pathologic_tT2b
classification_of_tumorprimary
classification_of_tumorPrior.primary
classification_of_tumorSynchronous.primary
tissue_or_organ_of_originLymph.nodes.of.head..face.and.neck
tissue_or_organ_of_originColon..NOS
tissue_or_organ_of_originKidney..NOS
tissue_or_organ_of_originBreast..NOS
tissue_or_organ_of_originCervix.uteri
tissue_or_organ_of_originOvary
tissue_or_organ_of_originLower.limb..NOS
tissue_or_organ_of_originUpper.inner.quadrant.of.breast
tissue_or_organ_of_originRectum..NOS
tissue_or_organ_of_originSkin.of.lower.limb.and.hip
tissue_or_organ_of_originLower.outer.quadrant.of.breast
0
2500
5000
7500
10000
Coefficient Value
🟥 Negative
🟦 Positive

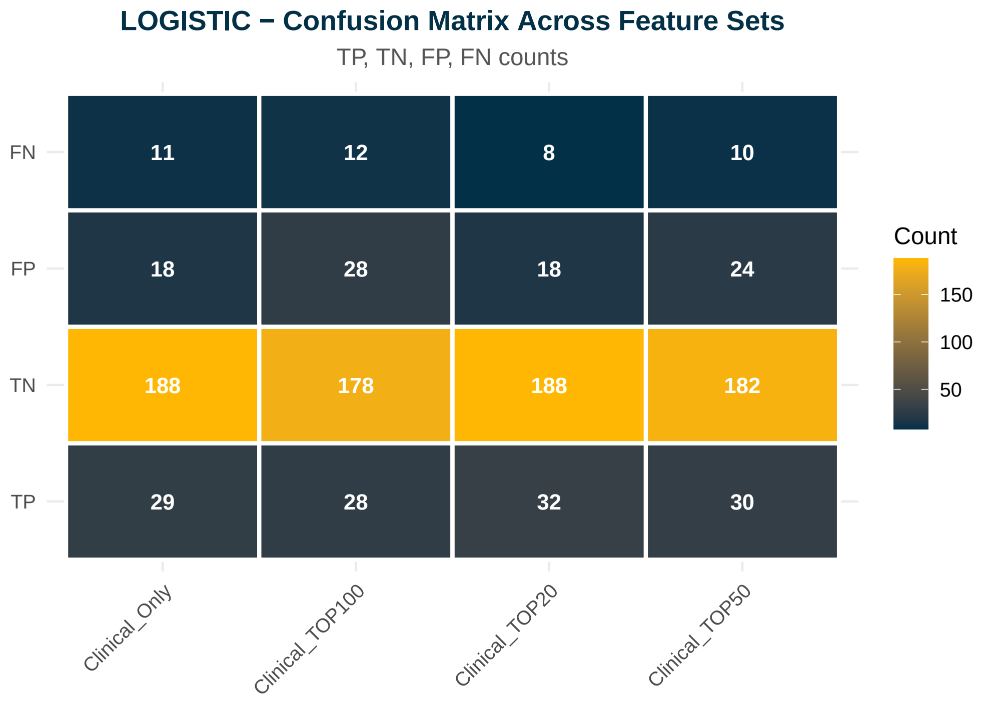

LOGISTIC − Clinical_TOP100
Top 20 non−zero features
tissue_or_organ_of_originSpecified.parts.of.peritoneum
tissue_or_organ_of_originUpper.outer.quadrant.of.breast
tissue_or_organ_of_originLower.inner.quadrant.of.breast
tissue_or_organ_of_originBrain..NOS
tissue_or_organ_of_originBone.marrow
ajcc_pathologic_tT2b
classification_of_tumorprimary
classification_of_tumorSynchronous.primary
classification_of_tumorPrior.primary
tissue_or_organ_of_originLymph.nodes.of.head..face.and.neck
tissue_or_organ_of_originColon..NOS
tissue_or_organ_of_originKidney..NOS
tissue_or_organ_of_originBreast..NOS
tissue_or_organ_of_originCervix.uteri
tissue_or_organ_of_originLower.limb..NOS
tissue_or_organ_of_originOvary
tissue_or_organ_of_originSkin.of.lower.limb.and.hip
tissue_or_organ_of_originUpper.inner.quadrant.of.breast
tissue_or_organ_of_originRectum..NOS
tissue_or_organ_of_originLower.outer.quadrant.of.breast
0
2000
4000
6000
Coefficient Value
🟥 Negative
🟦 Positive

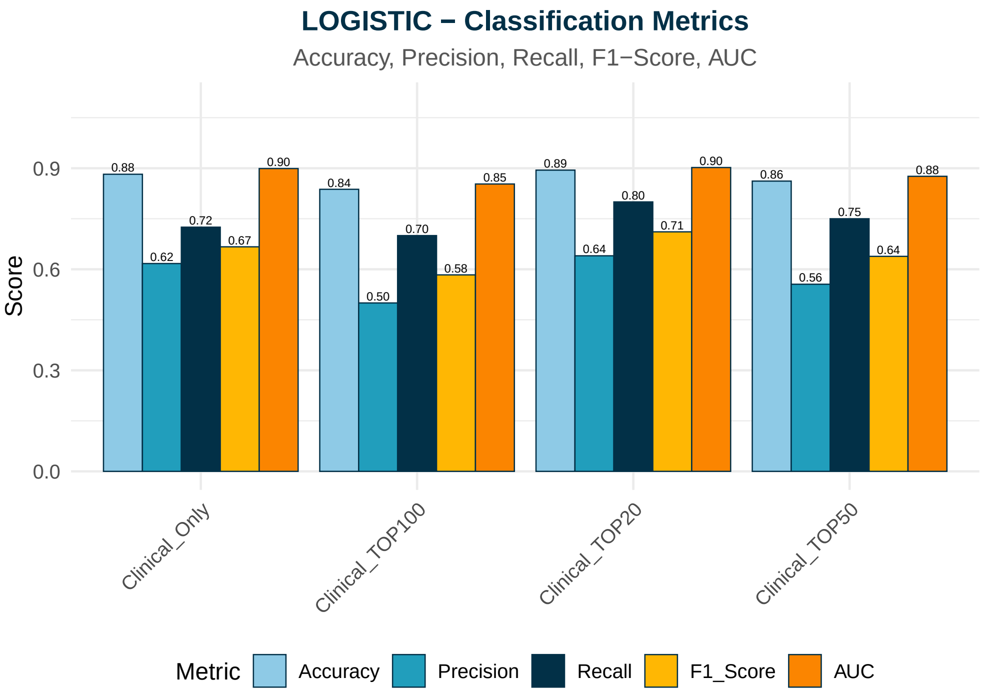




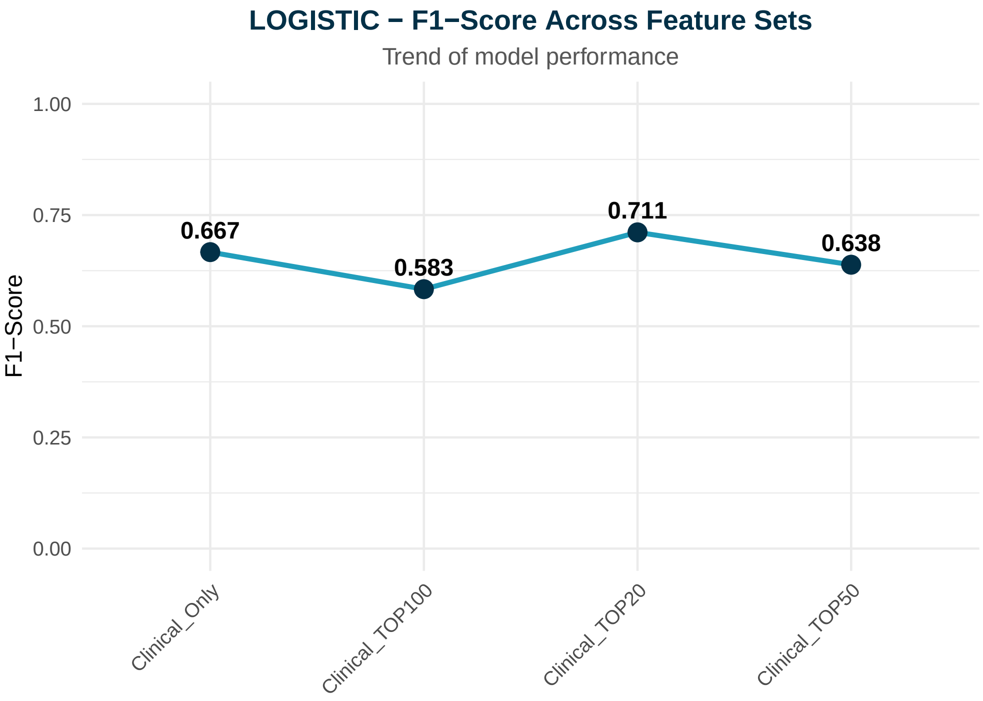
```r
logistic_smote_metrics <- plot_classification_metrics_single(logistic_smote
                                                             , threshold = 0.5
                                                             , csv_filename = "logistic_smote_classifica
```

```
##
## === CLASSIFICATION METRICS ===
## Clinical_Only:
##   TP=29 TN=188 FP=18 FN=11
##   Accuracy=0.882 Precision=0.617 Recall=0.725 F1=0.667 AUC=0.899

## Clinical_TOP100:
##   TP=28 TN=178 FP=28 FN=12
##   Accuracy=0.837 Precision=0.500 Recall=0.700 F1=0.583 AUC=0.853

## Clinical_TOP50:
##   TP=30 TN=182 FP=24 FN=10
##   Accuracy=0.862 Precision=0.556 Recall=0.750 F1=0.638 AUC=0.876

## Clinical_TOP20:
##   TP=32 TN=188 FP=18 FN=8
##   Accuracy=0.894 Precision=0.640 Recall=0.800 F1=0.711 AUC=0.902
```

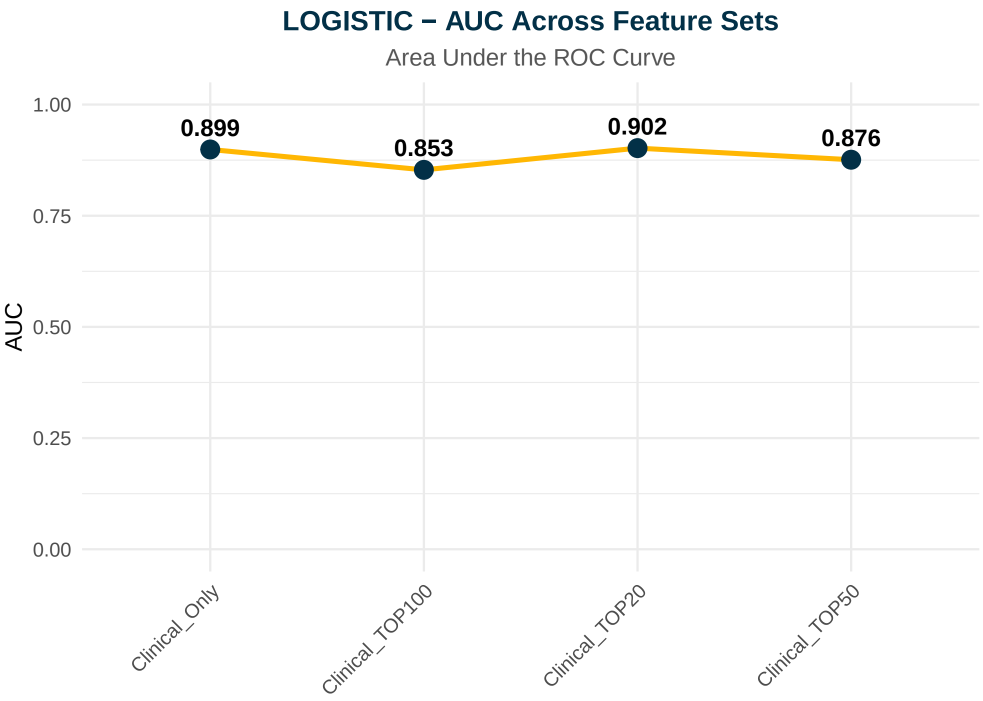


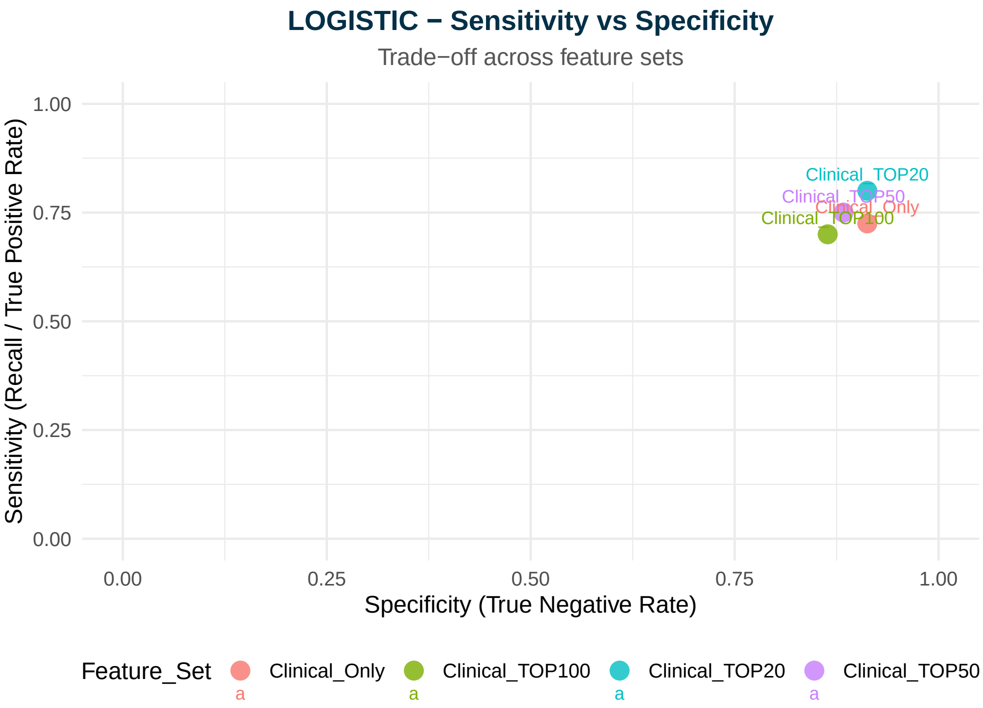

LOGISTIC − Classification Metrics

Accuracy, Precision, Recall, F1−Score, AUC


LOGISTIC − F1−Score Across Feature Sets

Trend of model performance



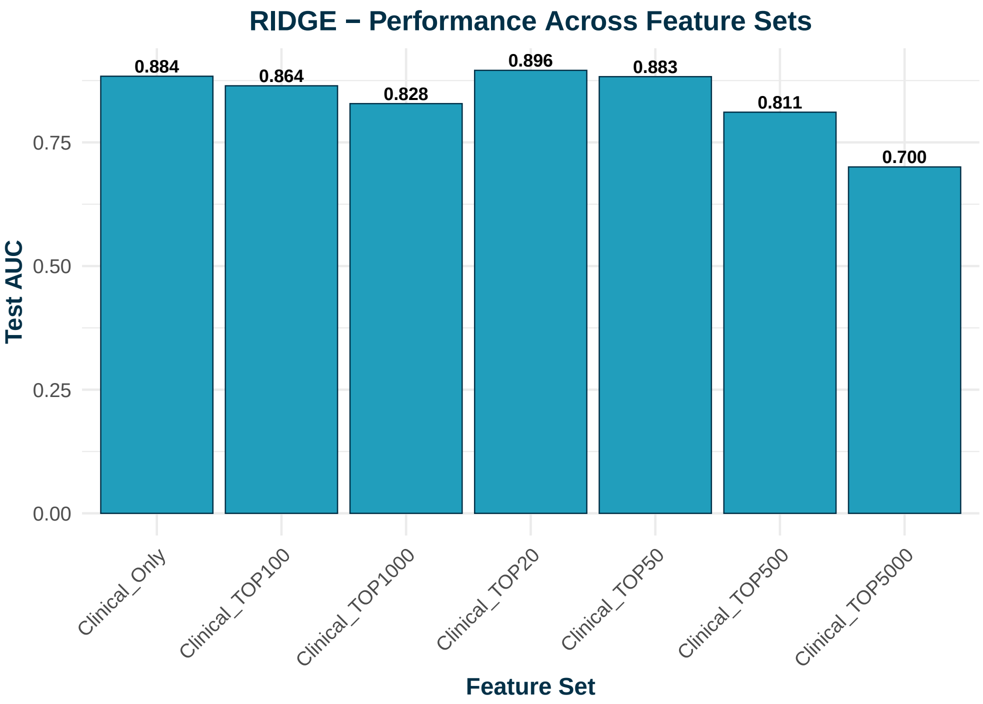


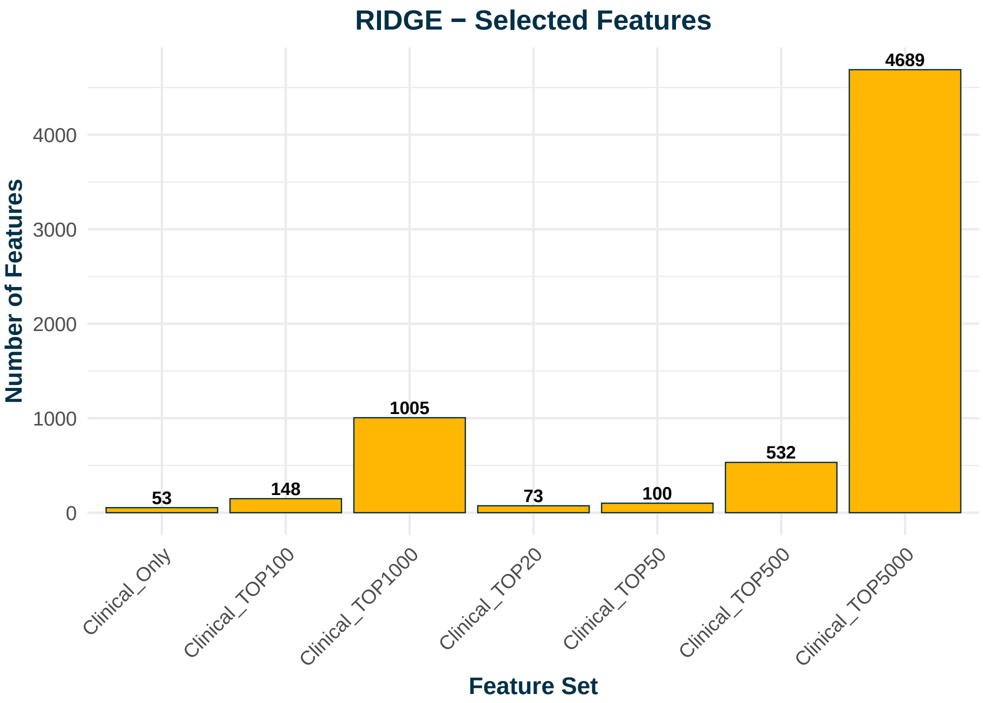
```
##
## === SUMMARY TABLE ===
##       Feature_Set TP  TN FP FN  Accuracy Precision Recall Specificity  F1_Score
## 1   Clinical_Only 29 188 18 11 0.8821138 0.6170213  0.725   0.9126214 0.6666667
## 2 Clinical_TOP100 28 178 28 12 0.8373984 0.5000000  0.700   0.8640777 0.5833333
## 3  Clinical_TOP50 30 182 24 10 0.8617886 0.5555556  0.750   0.8834951 0.6382979
## 4  Clinical_TOP20 32 188 18  8 0.8943089 0.6400000  0.800   0.9126214 0.7111111
##         AUC
## 1 0.8990291
## 2 0.8531553
## 3 0.8759709
## 4 0.9020631
##
## Exported classification metrics to: model_metrics/logistic_smote_classification_metrics.csv
```

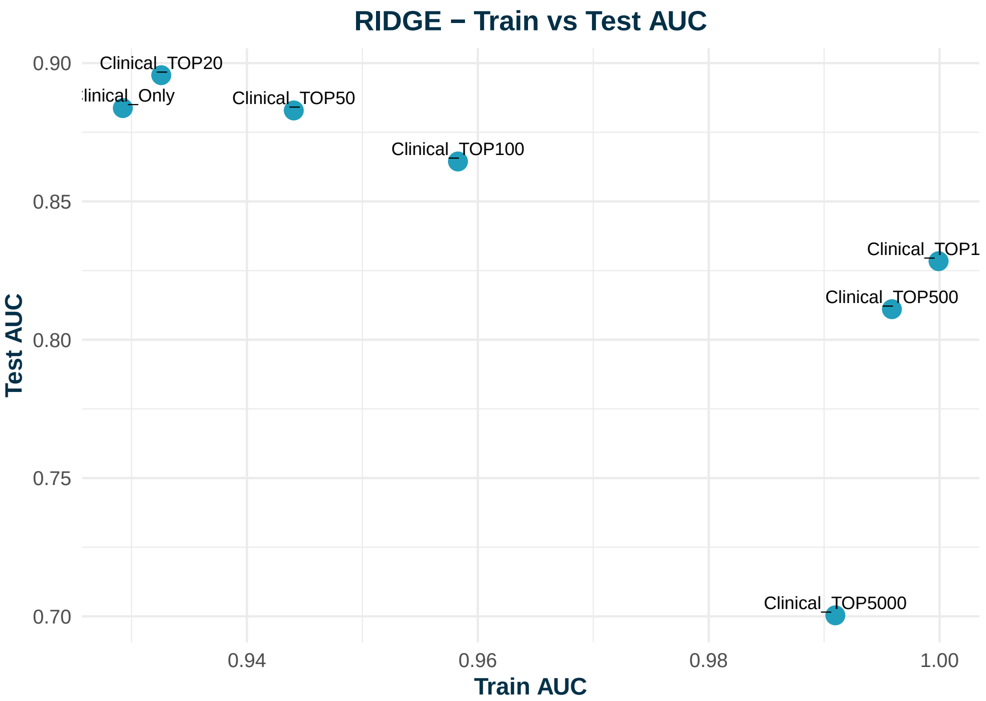
Ridge with SMOTE

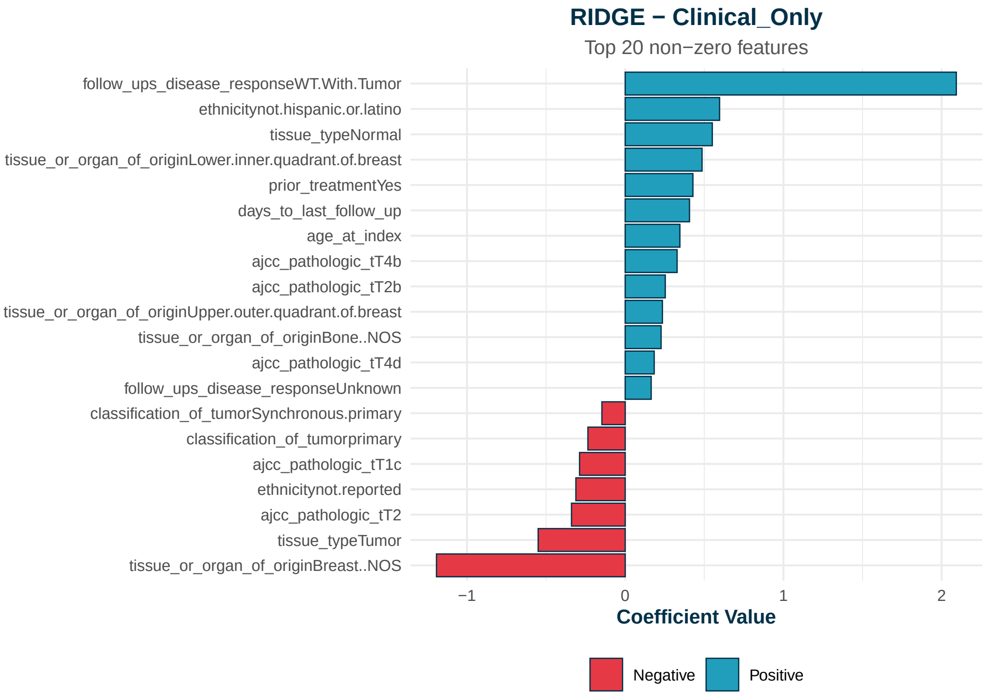
```
ridge_smote <- fit_single_model_across_features(
    model_type = "ridge"
    , X_train_all = smote_data$X_train
    , X_test_all = X_test
    , Y_train = smote_data$Y_train
    , Y_test = Y_test
    , n_clinical = n_clinical
    , top_genes_ranked = top_genes
    , gene_sets = c(5000, 1000, 500, 100, 50, 20)
)
```

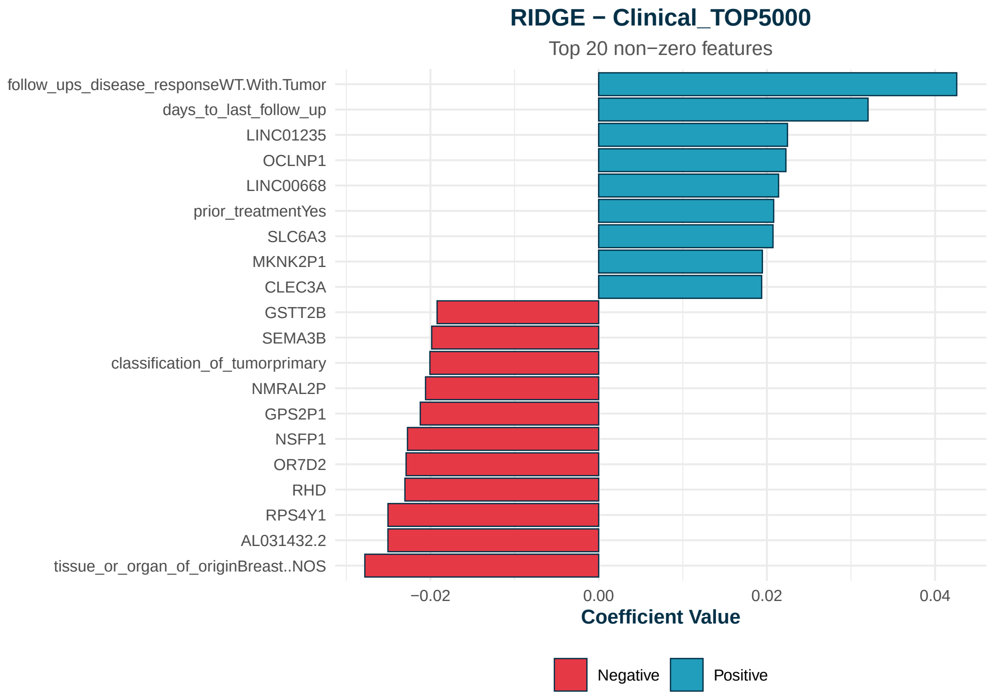
```
##
## === FITTING RIDGE ACROSS FEATURE SETS ===
##
## Fitting Clinical_Only...

## Fitting Clinical_TOP5000...

## Fitting Clinical_TOP1000...

## Fitting Clinical_TOP500...

## Fitting Clinical_TOP100...

## Fitting Clinical_TOP50...

## Fitting Clinical_TOP20...

##
## === SUMMARY TABLE ===
##        Feature_Set Model Features Train_AUC  Test_AUC Test_Accuracy
## 1    Clinical_Only RIDGE       53 0.9292484 0.8837379     0.8577236
## 2 Clinical_TOP5000 RIDGE     4689 0.9909693 0.7003641     0.7479675
## 3 Clinical_TOP1000 RIDGE     1005 0.9999125 0.8283981     0.8373984
## 4  Clinical_TOP500 RIDGE      532 0.9958703 0.8110437     0.8130081
## 5  Clinical_TOP100 RIDGE      148 0.9582802 0.8644417     0.8414634
## 6   Clinical_TOP50 RIDGE      100 0.9440571 0.8828883     0.8536585
## 7   Clinical_TOP20 RIDGE       73 0.9325766 0.8956311     0.8577236
## Exported metrics to: model_metrics/ridge_across_features_metrics.csv
```

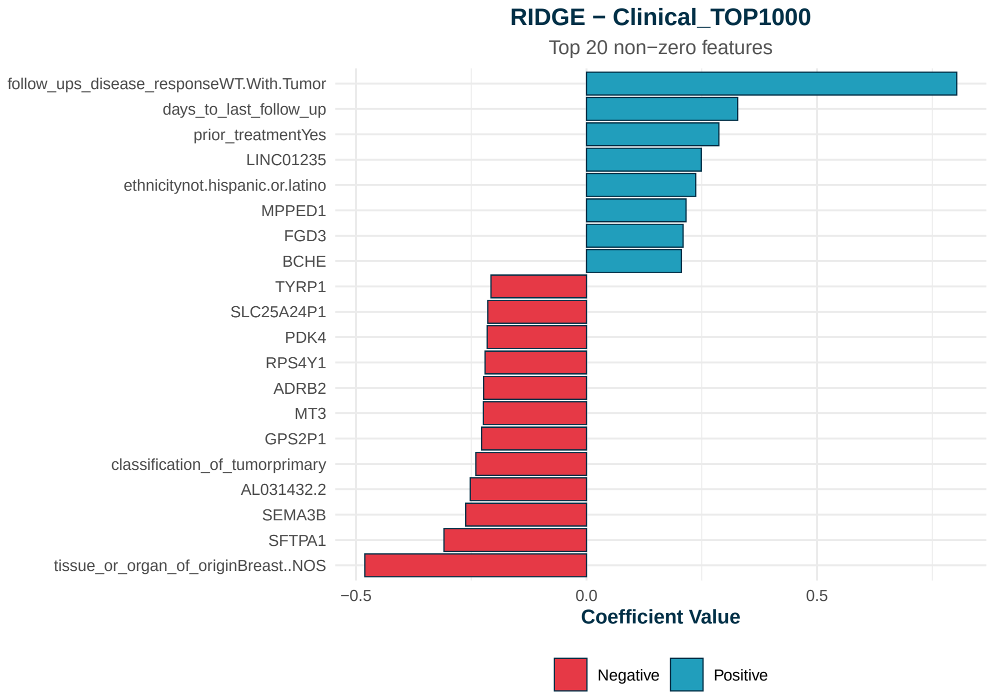

RIDGE − Performance Across Feature Sets

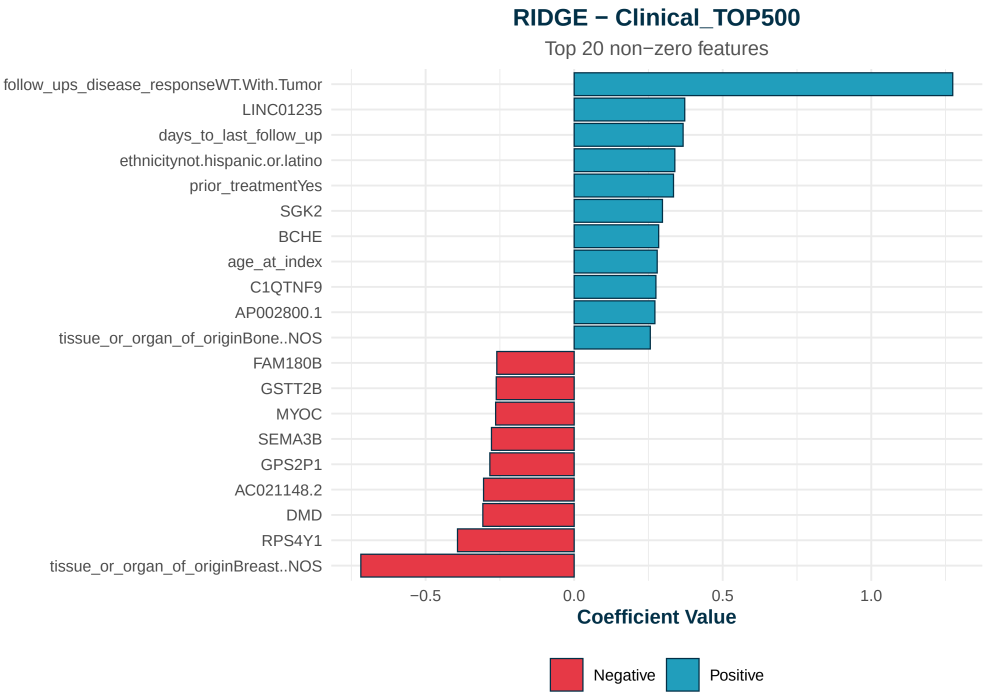

RIDGE − Selected Features

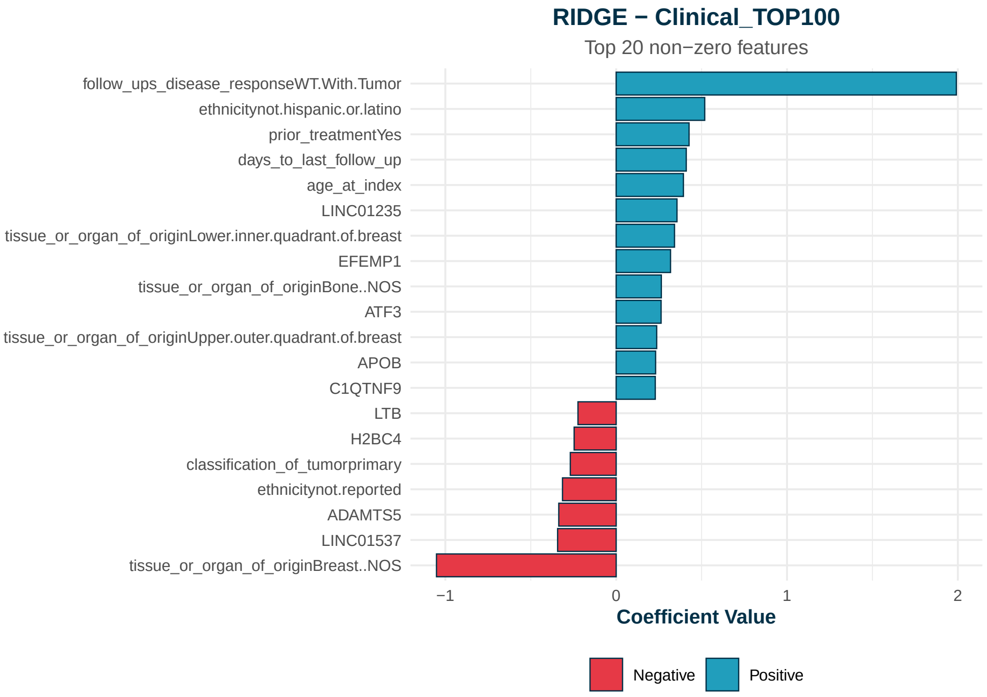

RIDGE − Train vs Test AUC
Test AUC
Train AUC
0.90
0.85
0.80
0.75
0.70
0.94
0.96
0.98
1.00
Clinical_TOP20
linical_Only
Clinical_TOP50
Clinical_TOP100
Clinical_TOP1
Clinical_TOP500
Clinical_TOP5000

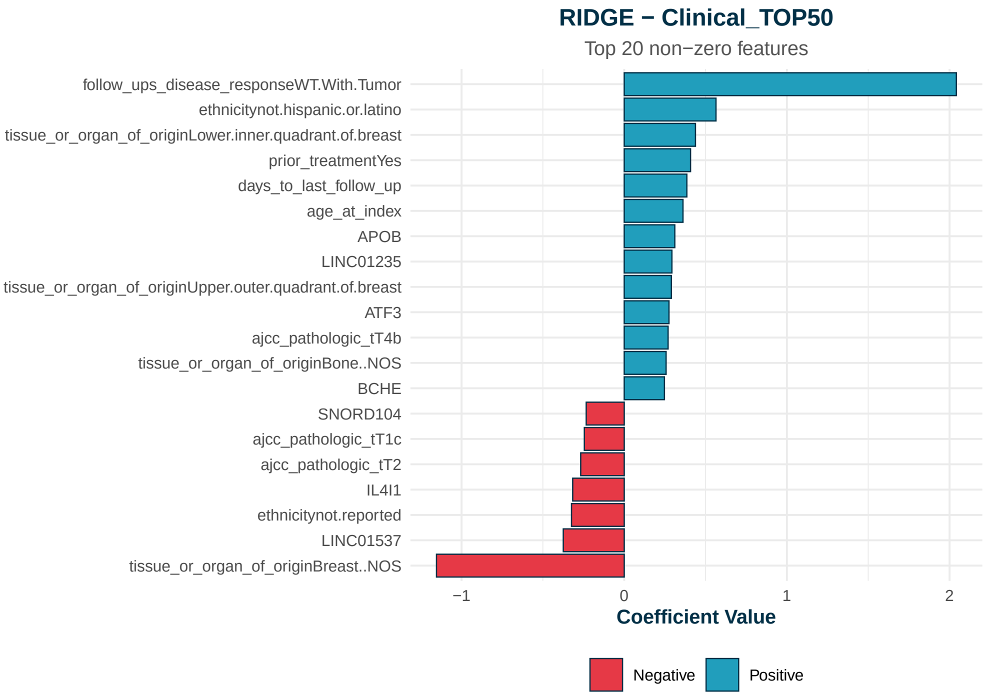

RIDGE − Clinical_Only
Top 20 non−zero features
follow_ups_disease_responseWT.With.Tumor
ethnicitynot.hispanic.or.latino
tissue_typeNormal
tissue_or_organ_of_originLower.inner.quadrant.of.breast
prior_treatmentYes
days_to_last_follow_up
age_at_index
ajcc_pathologic_tT4b
ajcc_pathologic_tT2b
tissue_or_organ_of_originUpper.outer.quadrant.of.breast
tissue_or_organ_of_originBone..NOS
ajcc_pathologic_tT4d
follow_ups_disease_responseUnknown
classification_of_tumorSynchronous.primary
classification_of_tumorprimary
ajcc_pathologic_tT1c
ethnicitynot.reported
ajcc_pathologic_tT2
tissue_typeTumor
tissue_or_organ_of_originBreast..NOS
−1
0
1
2
Coefficient Value
🟥 Negative
🟦 Positive

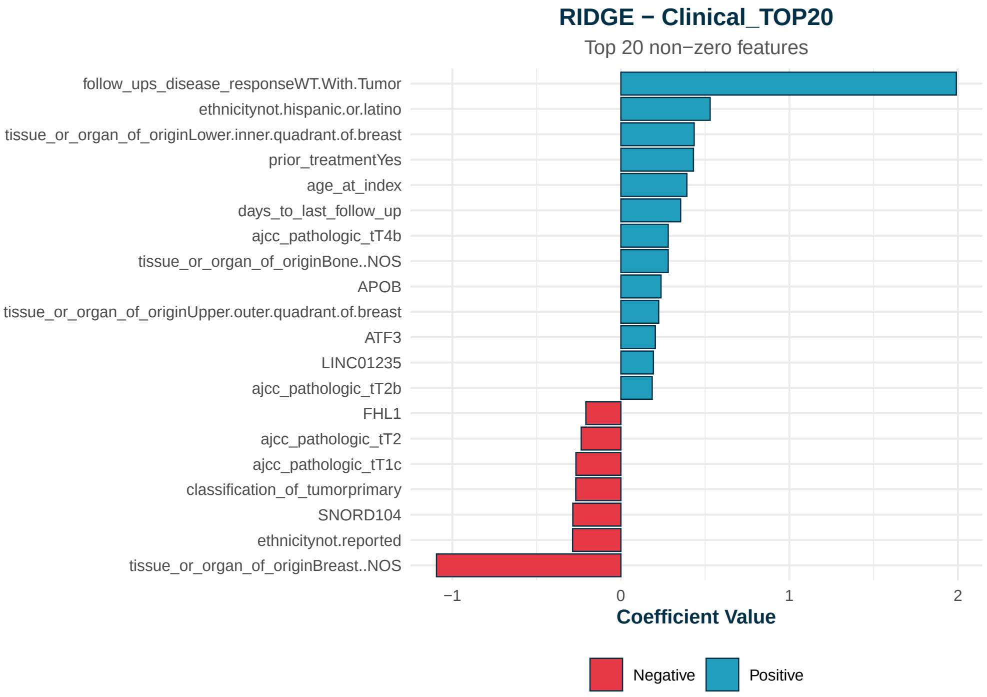

RIDGE − Clinical_TOP5000
Top 20 non−zero features
follow_ups_disease_responseWT.With.Tumor
days_to_last_follow_up
LINC01235
OCLNP1
LINC00668
prior_treatmentYes
SLC6A3
MKNK2P1
CLEC3A
GSTT2B
SEMA3B
classification_of_tumorprimary
NMRAL2P
GPS2P1
NSFP1
OR7D2
RHD
RPS4Y1
AL031432.2
tissue_or_organ_of_originBreast..NOS
−0.02
0.00
0.02
0.04
Coefficient Value
Negative
Positive


RIDGE − Clinical_TOP1000
Top 20 non−zero features
follow_ups_disease_responseWT.With.Tumor
days_to_last_follow_up
prior_treatmentYes
LINC01235
ethnicitynot.hispanic.or.latino
MPPED1
FGD3
BCHE
TYRP1
SLC25A24P1
PDK4
RPS4Y1
ADRB2
MT3
GPS2P1
classification_of_tumorprimary
AL031432.2
SEMA3B
SFTPA1
tissue_or_organ_of_originBreast..NOS
−0.5
0.0
0.5
Coefficient Value
Negative
Positive


RIDGE − Clinical_TOP500
Top 20 non−zero features
follow_ups_disease_responseWT.With.Tumor
LINC01235
days_to_last_follow_up
ethnicitynot.hispanic.or.latino
prior_treatmentYes
SGK2
BCHE
age_at_index
C1QTNF9
AP002800.1
tissue_or_organ_of_originBone..NOS
FAM180B
GSTT2B
MYOC
SEMA3B
GPS2P1
AC021148.2
DMD
RPS4Y1
tissue_or_organ_of_originBreast..NOS
−0.5
0.0
0.5
1.0
Coefficient Value
Negative
Positive

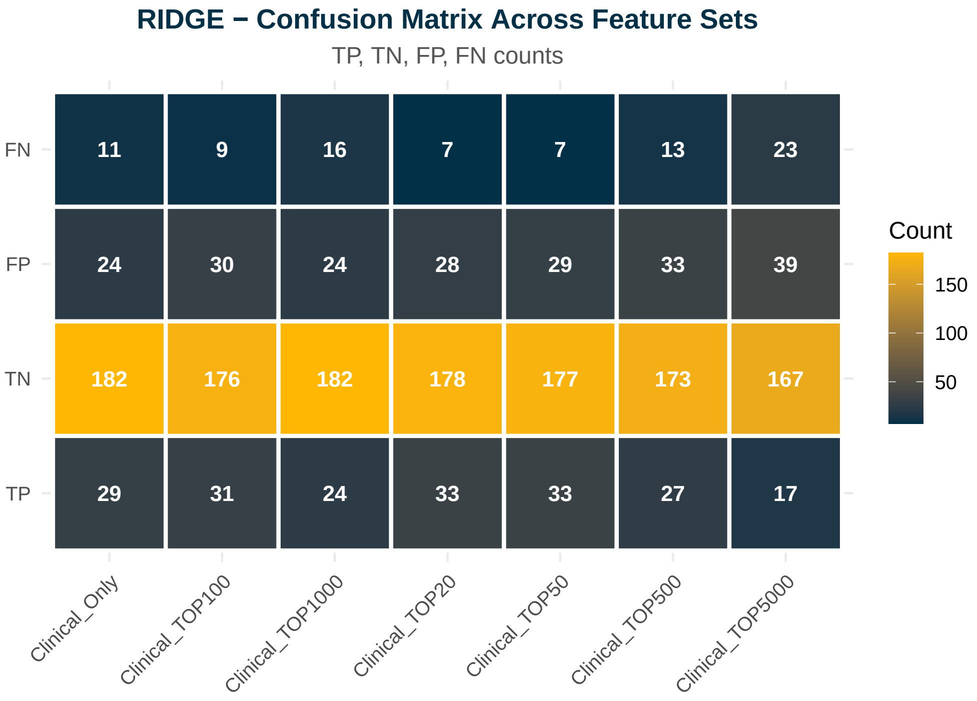

RIDGE − Clinical_TOP100
Top 20 non−zero features
follow_ups_disease_responseWT.With.Tumor
ethnicitynot.hispanic.or.latino
prior_treatmentYes
days_to_last_follow_up
age_at_index
LINC01235
tissue_or_organ_of_originLower.inner.quadrant.of.breast
EFEMP1
tissue_or_organ_of_originBone..NOS
ATF3
tissue_or_organ_of_originUpper.outer.quadrant.of.breast
APOB
C1QTNF9
LTB
H2BC4
classification_of_tumorprimary
ethnicitynot.reported
ADAMTS5
LINC01537
tissue_or_organ_of_originBreast..NOS
−1
0
1
2
Coefficient Value
Negative
Positive

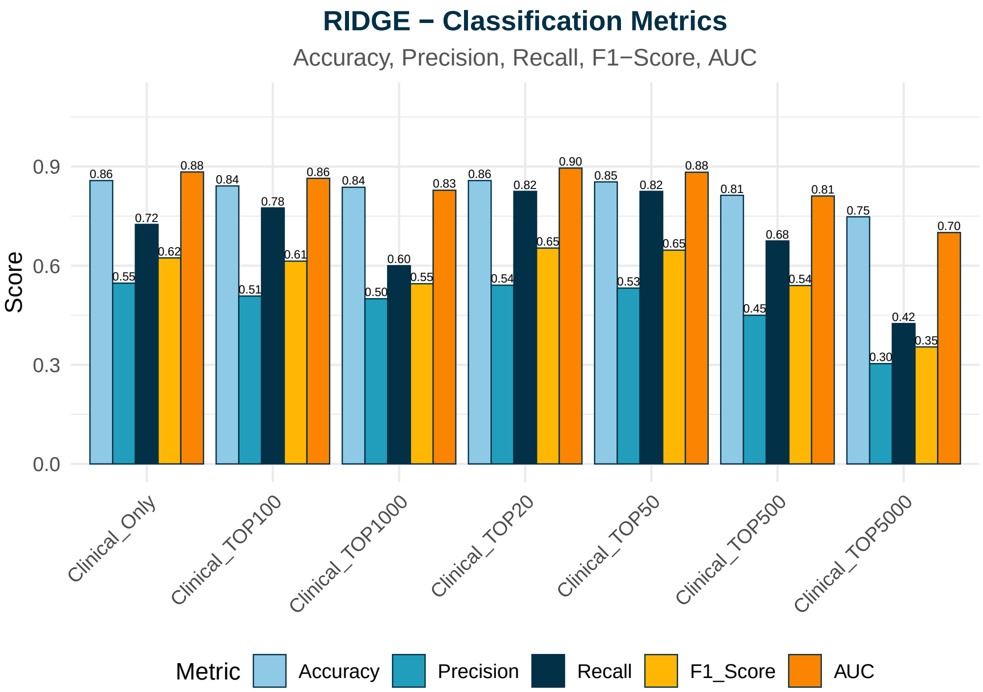


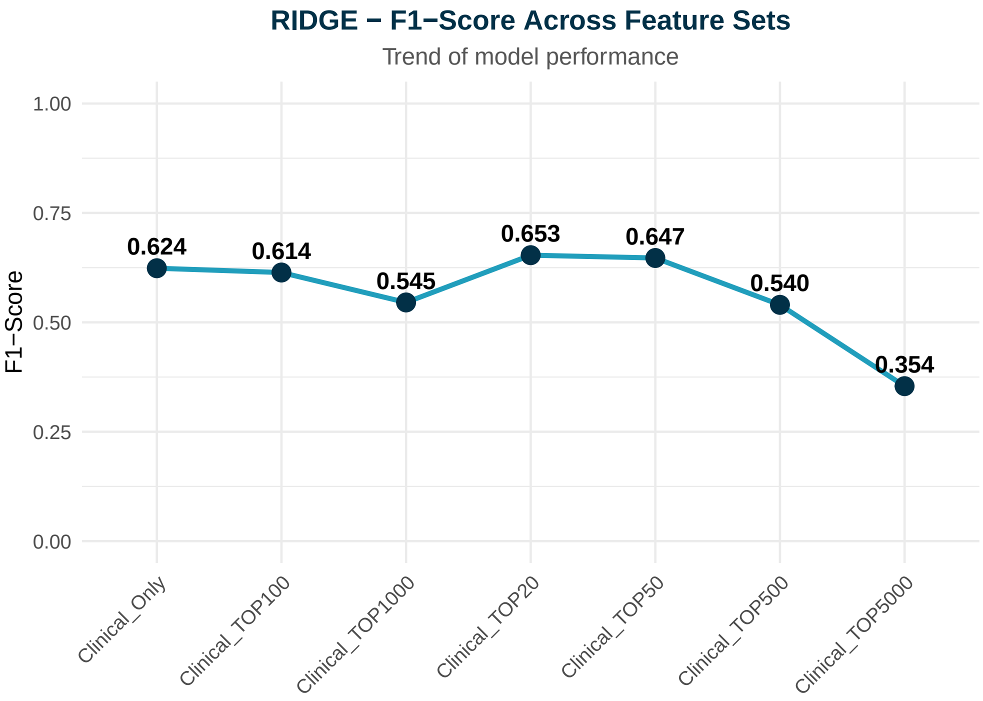


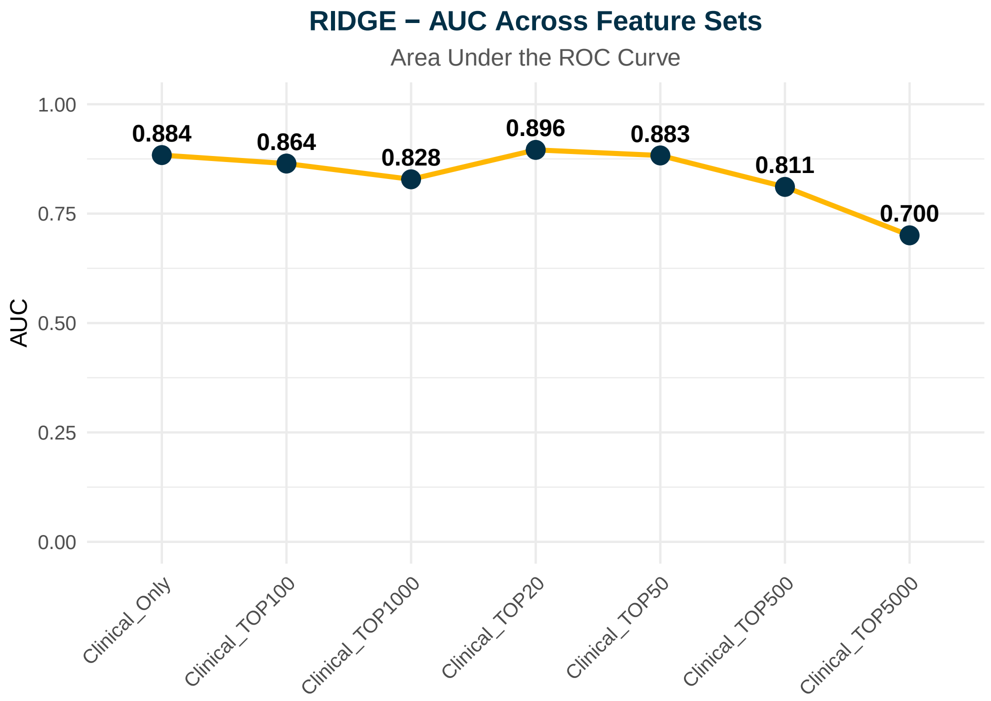
```
ridge_smote_metrics <- plot_classification_metrics_single(ridge_smote
                                                          , threshold = 0.5
                                                          , csv_filename = "ridge_smote_classification_
```

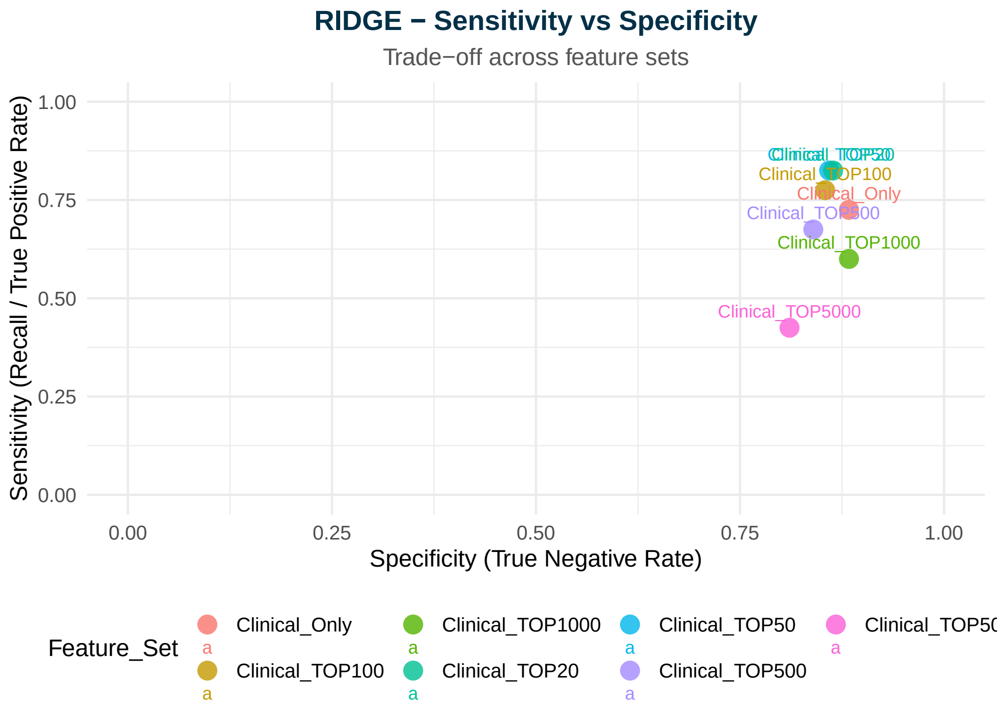
```
##
## === CLASSIFICATION METRICS ===
## Clinical_Only:
##   TP=29 TN=182 FP=24 FN=11
##   Accuracy=0.858 Precision=0.547 Recall=0.725 F1=0.624 AUC=0.884

## Clinical_TOP5000:
##   TP=17 TN=167 FP=39 FN=23
##   Accuracy=0.748 Precision=0.304 Recall=0.425 F1=0.354 AUC=0.700

## Clinical_TOP1000:
##   TP=24 TN=182 FP=24 FN=16
##   Accuracy=0.837 Precision=0.500 Recall=0.600 F1=0.545 AUC=0.828

## Clinical_TOP500:
##   TP=27 TN=173 FP=33 FN=13
##   Accuracy=0.813 Precision=0.450 Recall=0.675 F1=0.540 AUC=0.811

## Clinical_TOP100:
##   TP=31 TN=176 FP=30 FN=9
##   Accuracy=0.841 Precision=0.508 Recall=0.775 F1=0.614 AUC=0.864

## Clinical_TOP50:
##   TP=33 TN=177 FP=29 FN=7
##   Accuracy=0.854 Precision=0.532 Recall=0.825 F1=0.647 AUC=0.883

## Clinical_TOP20:
##   TP=33 TN=178 FP=28 FN=7
##   Accuracy=0.858 Precision=0.541 Recall=0.825 F1=0.653 AUC=0.896
```




RIDGE − Classification Metrics
Accuracy, Precision, Recall, F1−Score, AUC
Score
0.0
0.3
0.6
0.9
0.86
0.55
0.72
0.62
0.88
0.84
0.51
0.78
0.61
0.86
0.84
0.50
0.60
0.55
0.83
0.86
0.54
0.82
0.65
0.90
0.85
0.53
0.82
0.65
0.88
0.81
0.45
0.68
0.54
0.81
0.75
0.30
0.42
0.35
0.70
Clinical_Only
Clinical_TOP100
Clinical_TOP1000
Clinical_TOP20
Clinical_TOP50
Clinical_TOP500
Clinical_TOP5000
Metric
Accuracy
Precision
Recall
F1_Score
AUC


RIDGE − F1−Score Across Feature Sets
Trend of model performance
F1−Score
0.00
0.25
0.50
0.75
1.00
0.624
0.614
0.545
0.653
0.647
0.540
0.354
Clinical_Only
Clinical_TOP100
Clinical_TOP1000
Clinical_TOP20
Clinical_TOP50
Clinical_TOP500
Clinical_TOP5000



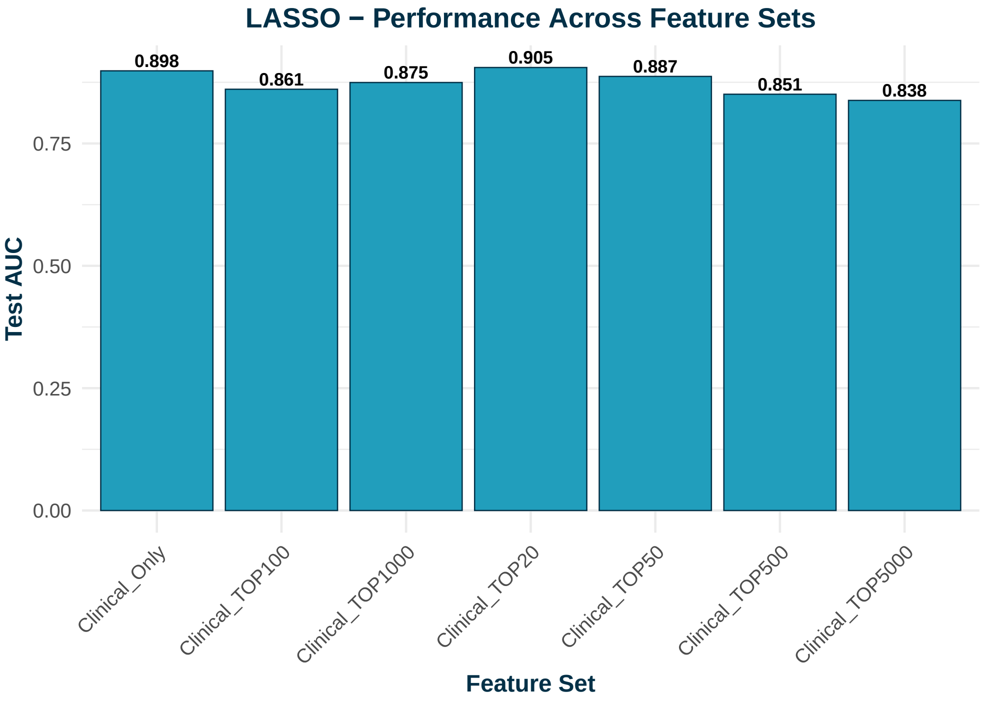


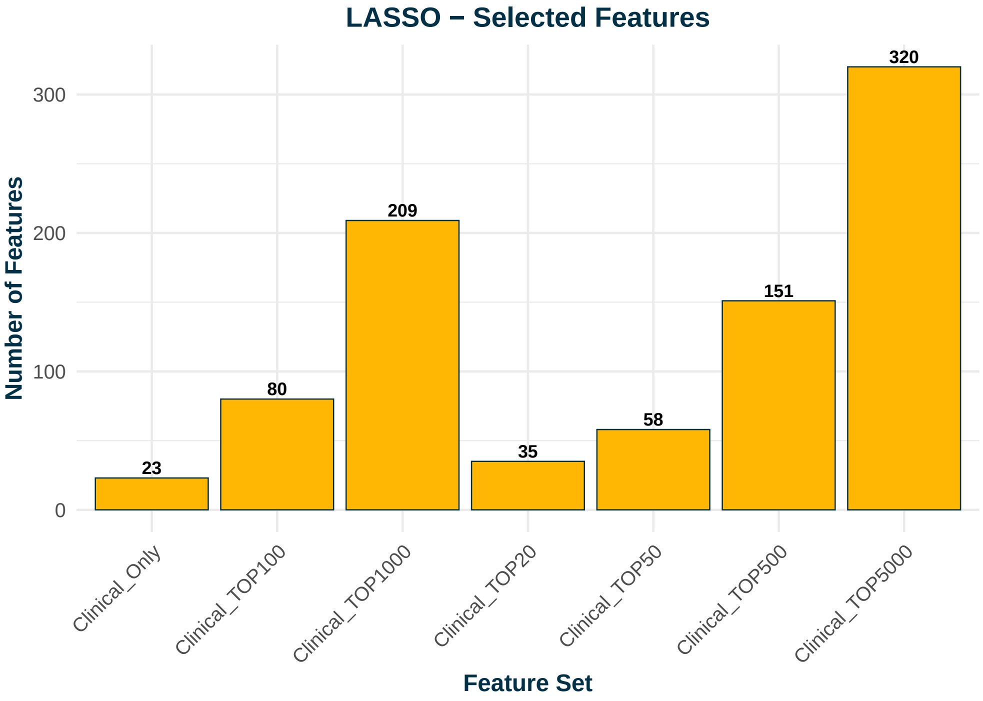
```
##
## === SUMMARY TABLE ===
##        Feature_Set TP  TN FP FN  Accuracy Precision Recall Specificity
## 1    Clinical_Only 29 182 24 11 0.8577236 0.5471698  0.725   0.8834951
## 2 Clinical_TOP5000 17 167 39 23 0.7479675 0.3035714  0.425   0.8106796
## 3 Clinical_TOP1000 24 182 24 16 0.8373984 0.5000000  0.600   0.8834951
## 4  Clinical_TOP500 27 173 33 13 0.8130081 0.4500000  0.675   0.8398058
## 5  Clinical_TOP100 31 176 30  9 0.8414634 0.5081967  0.775   0.8543689
## 6   Clinical_TOP50 33 177 29  7 0.8536585 0.5322581  0.825   0.8592233
## 7   Clinical_TOP20 33 178 28  7 0.8577236 0.5409836  0.825   0.8640777
##    F1_Score       AUC
## 1 0.6236559 0.8837379
## 2 0.3541667 0.7003641
## 3 0.5454545 0.8283981
## 4 0.5400000 0.8110437
## 5 0.6138614 0.8644417
## 6 0.6470588 0.8828883
## 7 0.6534653 0.8956311
##
## Exported classification metrics to: model_metrics/ridge_smote_classification_metrics.csv
```

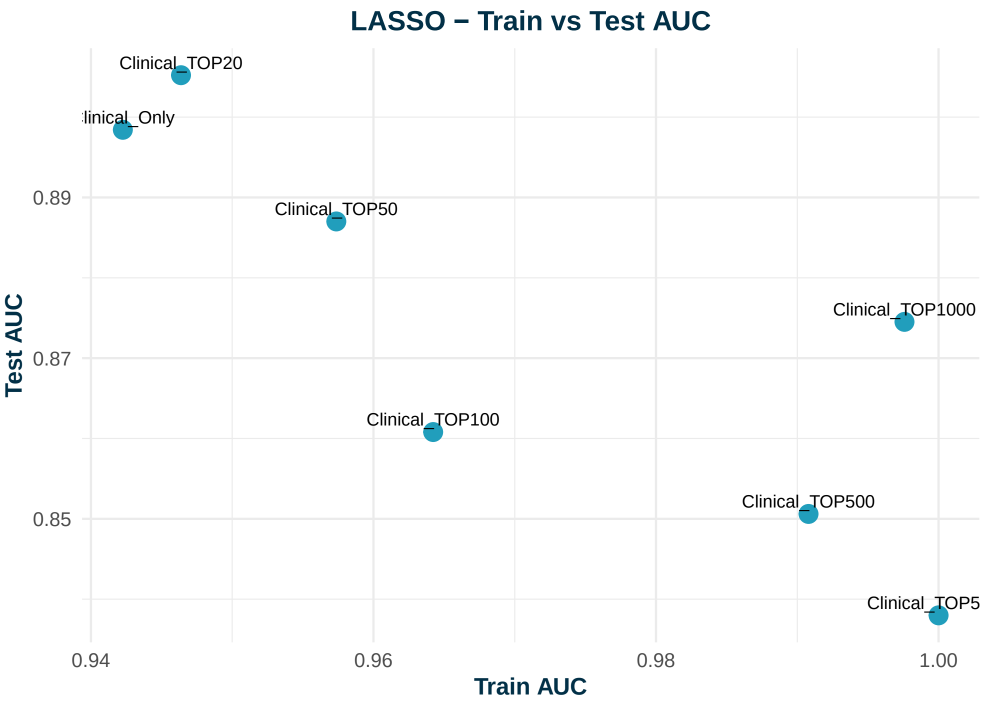
Lasso with SMOTE

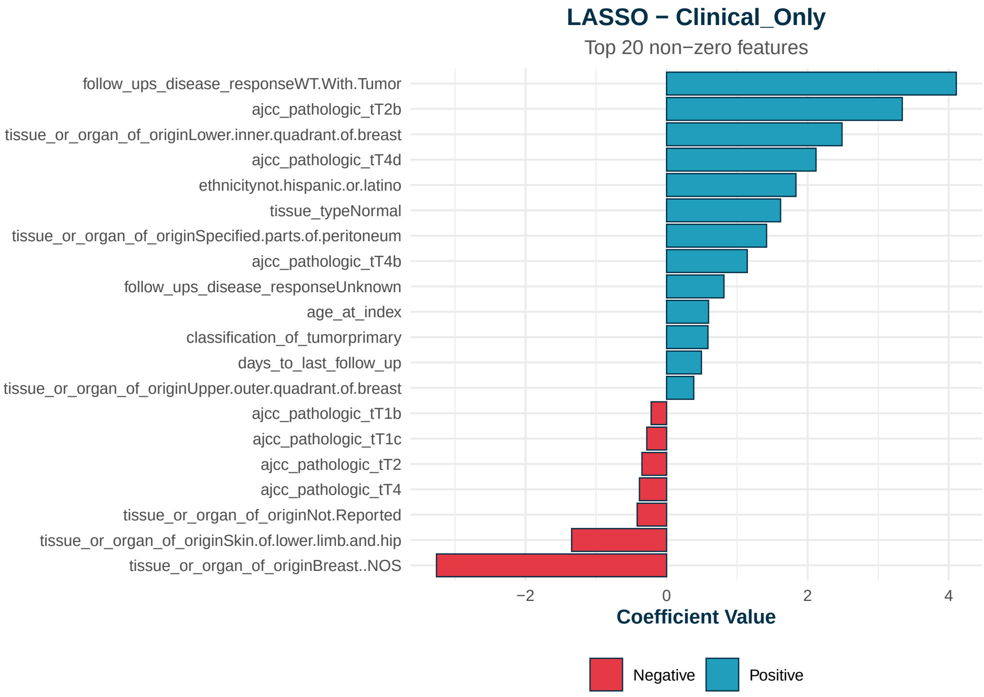
```r
lasso_smote <- fit_single_model_across_features(
    model_type = "lasso"
    , X_train_all = smote_data$X_train
    , X_test_all = X_test
    , Y_train = smote_data$Y_train
    , Y_test = Y_test
    , n_clinical = n_clinical
    , top_genes_ranked = top_genes
    , gene_sets = c(5000, 1000, 500, 100, 50, 20)
)
```

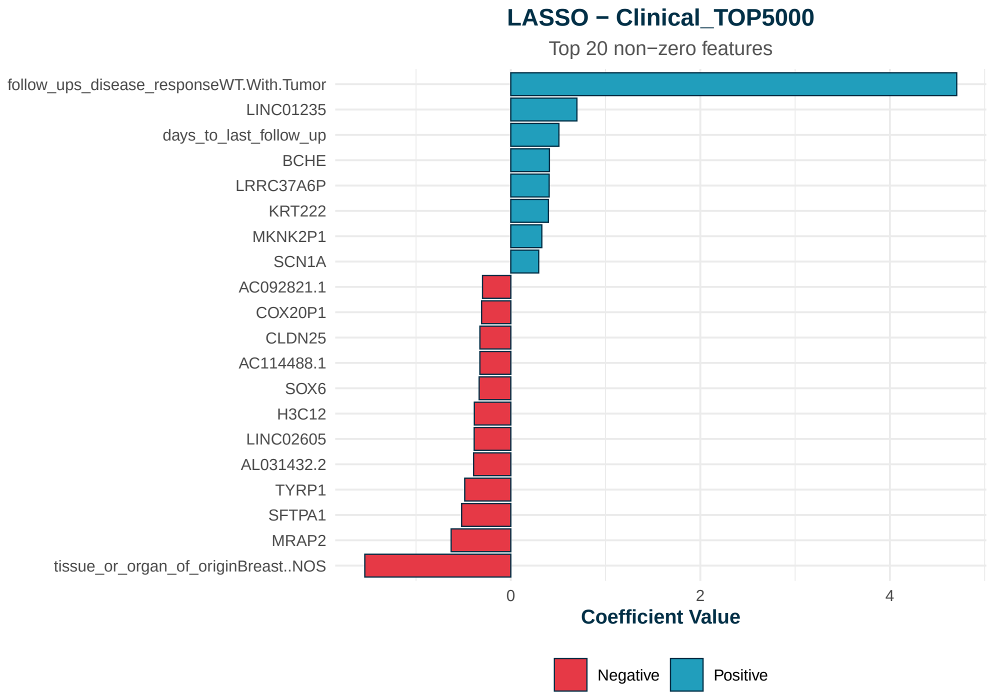
```
##
## === FITTING LASSO ACROSS FEATURE SETS ===
##
## Fitting Clinical_Only...

## Fitting Clinical_TOP5000...

## Fitting Clinical_TOP1000...

## Fitting Clinical_TOP500...

## Fitting Clinical_TOP100...

## Fitting Clinical_TOP50...

## Fitting Clinical_TOP20...

##
## === SUMMARY TABLE ===
##        Feature_Set Model Features Train_AUC  Test_AUC Test_Accuracy
## 1    Clinical_Only LASSO       23 0.9422594 0.8984223     0.8739837
## 2 Clinical_TOP5000 LASSO      320 1.0000000 0.8379854     0.8739837
## 3 Clinical_TOP1000 LASSO      209 0.9975865 0.8745146     0.8699187
## 4  Clinical_TOP500 LASSO      151 0.9907942 0.8506068     0.8373984
## 5  Clinical_TOP100 LASSO       80 0.9642257 0.8608010     0.8739837
## 6   Clinical_TOP50 LASSO       58 0.9573700 0.8870146     0.8861789
## 7   Clinical_TOP20 LASSO       35 0.9463771 0.9052184     0.8861789
## Exported metrics to: model_metrics/lasso_across_features_metrics.csv
```

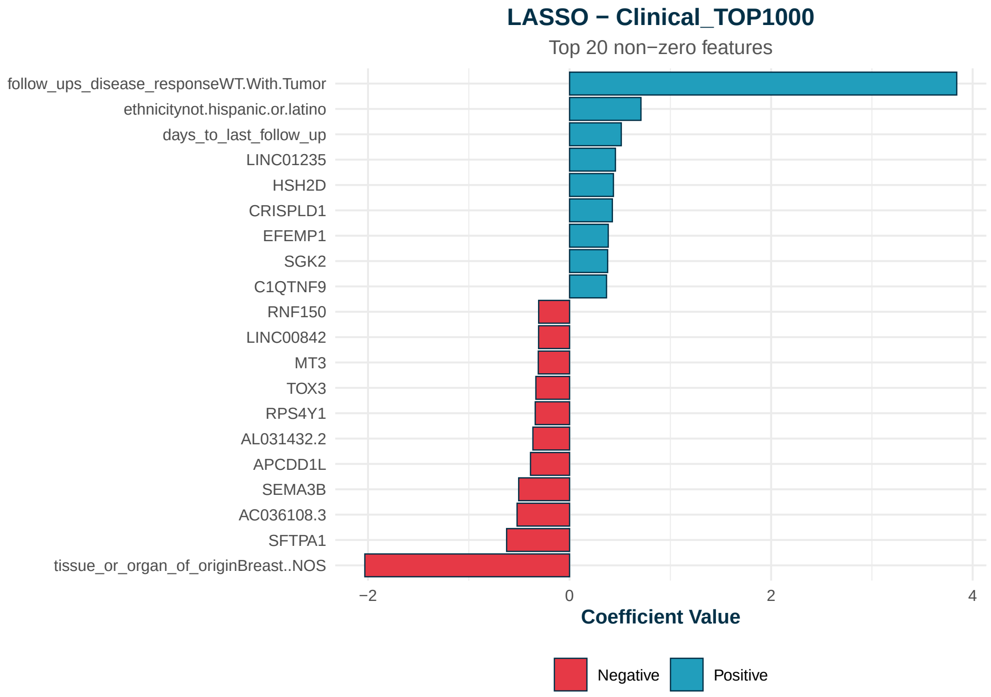


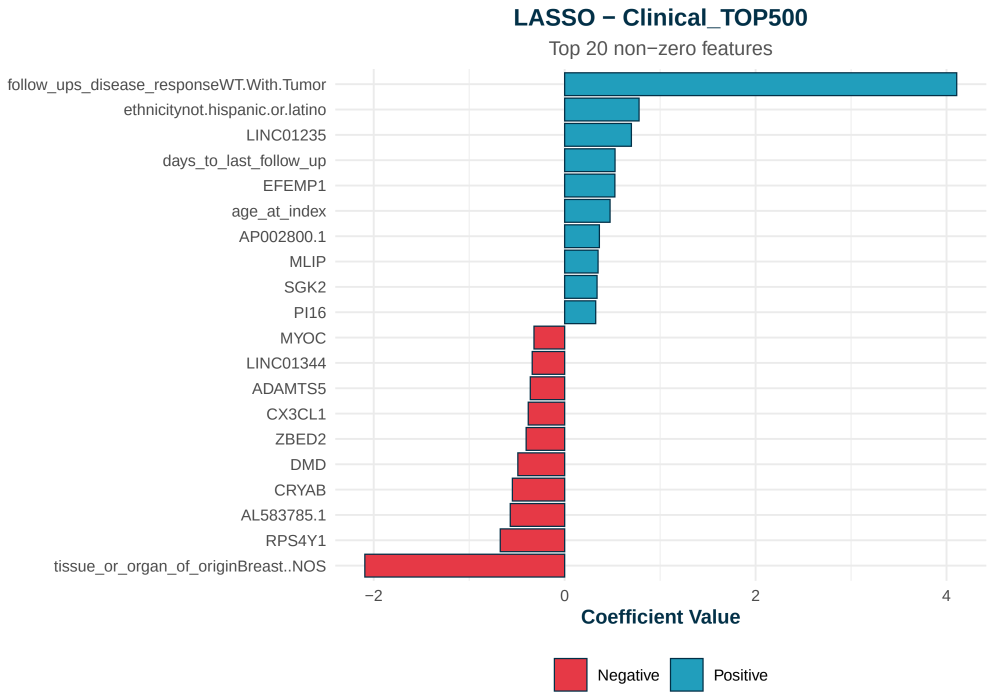


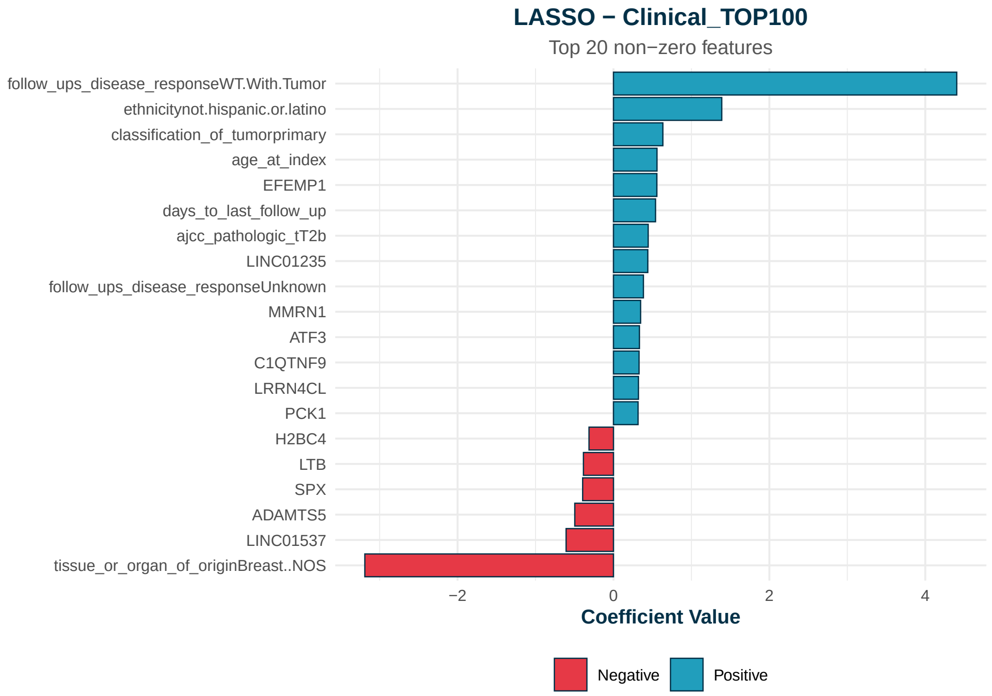

LASSO − Train vs Test AUC
Clinical_TOP20
linical_Only
Clinical_TOP50
Clinical_TOP1000
Clinical_TOP100
Clinical_TOP500
Clinical_TOP5
Test AUC
0.89
0.87
0.85
0.94
0.96
0.98
1.00
Train AUC

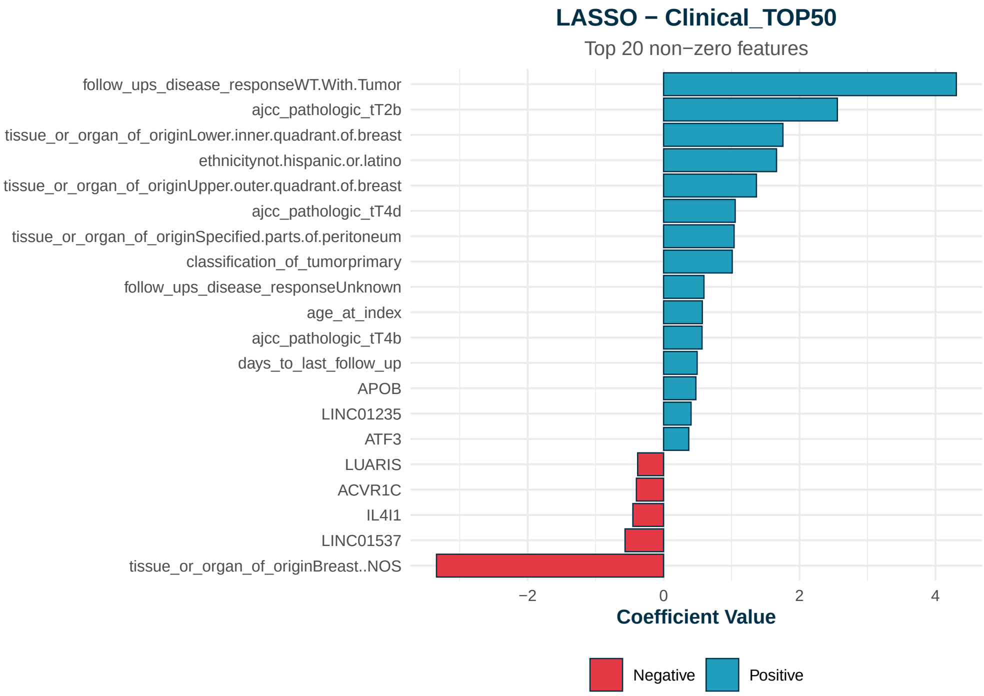

LASSO − Clinical_Only
Top 20 non−zero features
follow_ups_disease_responseWT.With.Tumor
ajcc_pathologic_tT2b
tissue_or_organ_of_originLower.inner.quadrant.of.breast
ajcc_pathologic_tT4d
ethnicitynot.hispanic.or.latino
tissue_typeNormal
tissue_or_organ_of_originSpecified.parts.of.peritoneum
ajcc_pathologic_tT4b
follow_ups_disease_responseUnknown
age_at_index
classification_of_tumorprimary
days_to_last_follow_up
tissue_or_organ_of_originUpper.outer.quadrant.of.breast
ajcc_pathologic_tT1b
ajcc_pathologic_tT1c
ajcc_pathologic_tT2
ajcc_pathologic_tT4
tissue_or_organ_of_originNot.Reported
tissue_or_organ_of_originSkin.of.lower.limb.and.hip
tissue_or_organ_of_originBreast..NOS
−2
0
2
4
Coefficient Value
🟥 Negative
🟦 Positive

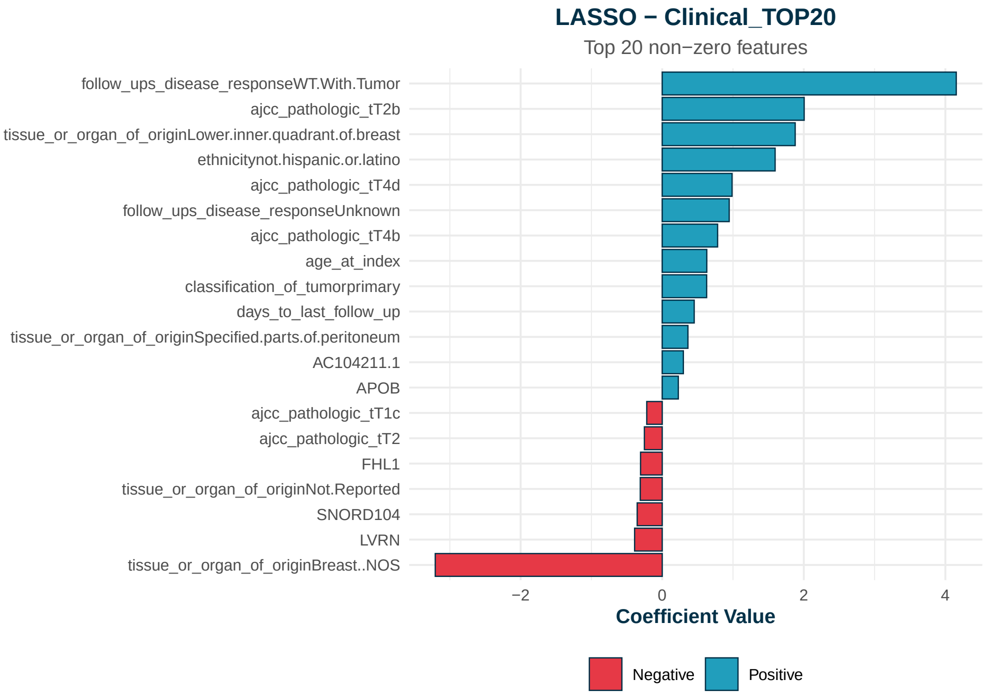

LASSO − Clinical_TOP5000
Top 20 non−zero features
follow_ups_disease_responseWT.With.Tumor
LINC01235
days_to_last_follow_up
BCHE
LRRC37A6P
KRT222
MKNK2P1
SCN1A
AC092821.1
COX20P1
CLDN25
AC114488.1
SOX6
H3C12
LINC02605
AL031432.2
TYRP1
SFTPA1
MRAP2
tissue_or_organ_of_originBreast..NOS
0
2
4
Coefficient Value
Negative
Positive


LASSO − Clinical_TOP1000
Top 20 non−zero features
follow_ups_disease_responseWT.With.Tumor
ethnicitynot.hispanic.or.latino
days_to_last_follow_up
LINC01235
HSH2D
CRISPLD1
EFEMP1
SGK2
C1QTNF9
RNF150
LINC00842
MT3
TOX3
RPS4Y1
AL031432.2
APCDD1L
SEMA3B
AC036108.3
SFTPA1
tissue_or_organ_of_originBreast..NOS
−2
0
2
4
Coefficient Value
Negative
Positive


LASSO − Clinical_TOP500
Top 20 non−zero features
follow_ups_disease_responseWT.With.Tumor
ethnicitynot.hispanic.or.latino
LINC01235
days_to_last_follow_up
EFEMP1
age_at_index
AP002800.1
MLIP
SGK2
PI16
MYOC
LINC01344
ADAMTS5
CX3CL1
ZBED2
DMD
CRYAB
AL583785.1
RPS4Y1
tissue_or_organ_of_originBreast..NOS
−2
0
2
4
Coefficient Value
Negative
Positive

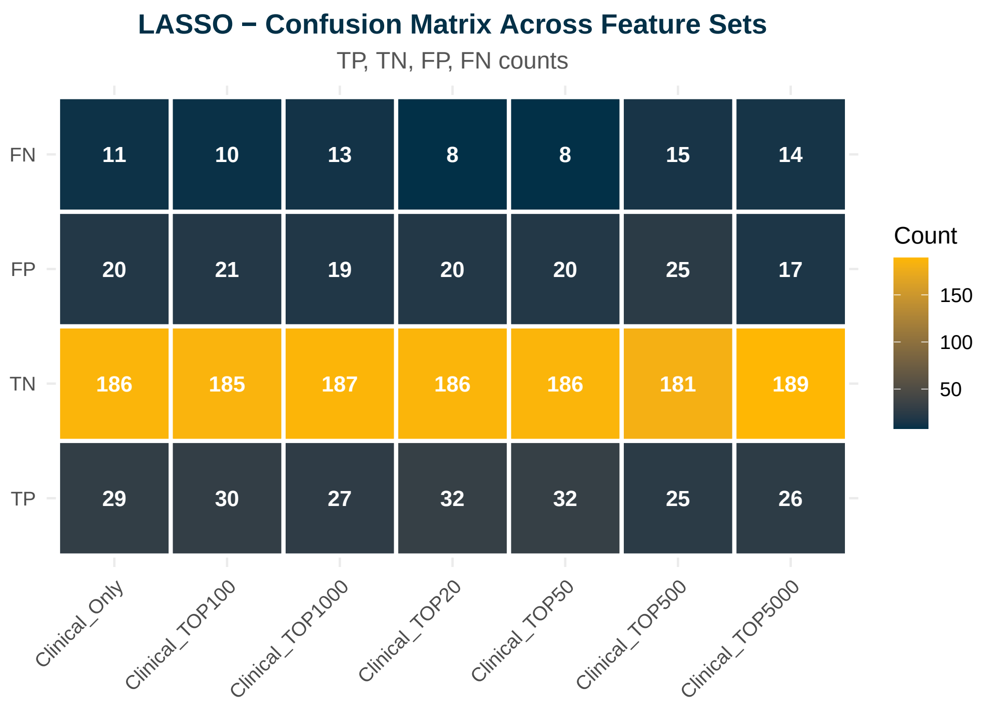

LASSO − Clinical_TOP100
Top 20 non−zero features
follow_ups_disease_responseWT.With.Tumor
ethnicitynot.hispanic.or.latino
classification_of_tumorprimary
age_at_index
EFEMP1
days_to_last_follow_up
ajcc_pathologic_tT2b
LINC01235
follow_ups_disease_responseUnknown
MMRN1
ATF3
C1QTNF9
LRRN4CL
PCK1
H2BC4
LTB
SPX
ADAMTS5
LINC01537
tissue_or_organ_of_originBreast..NOS
−2
0
2
4
Coefficient Value
Negative
Positive



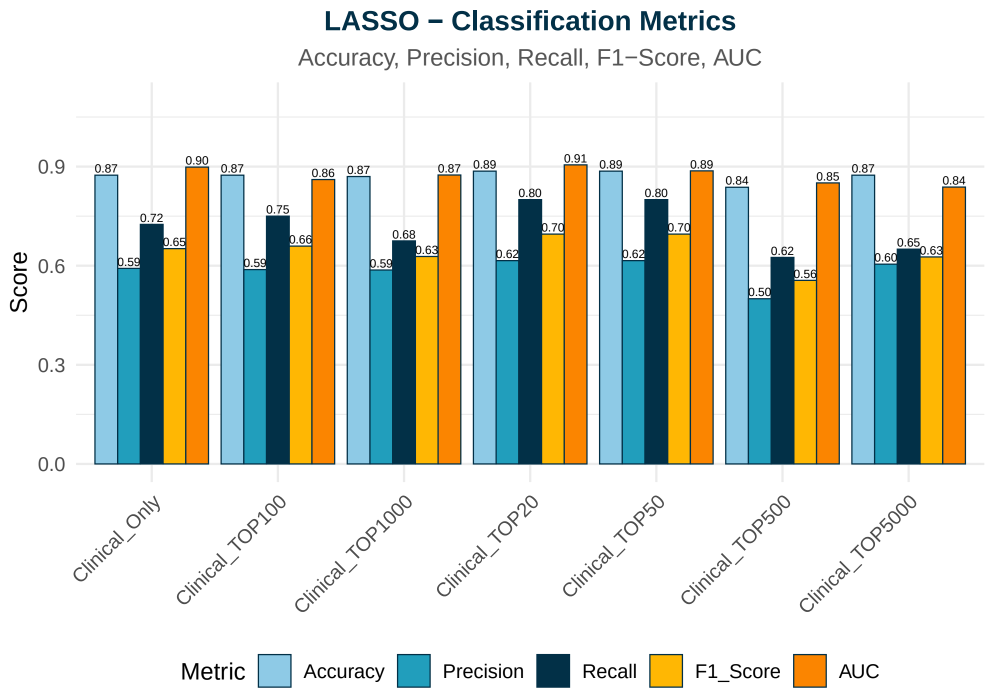


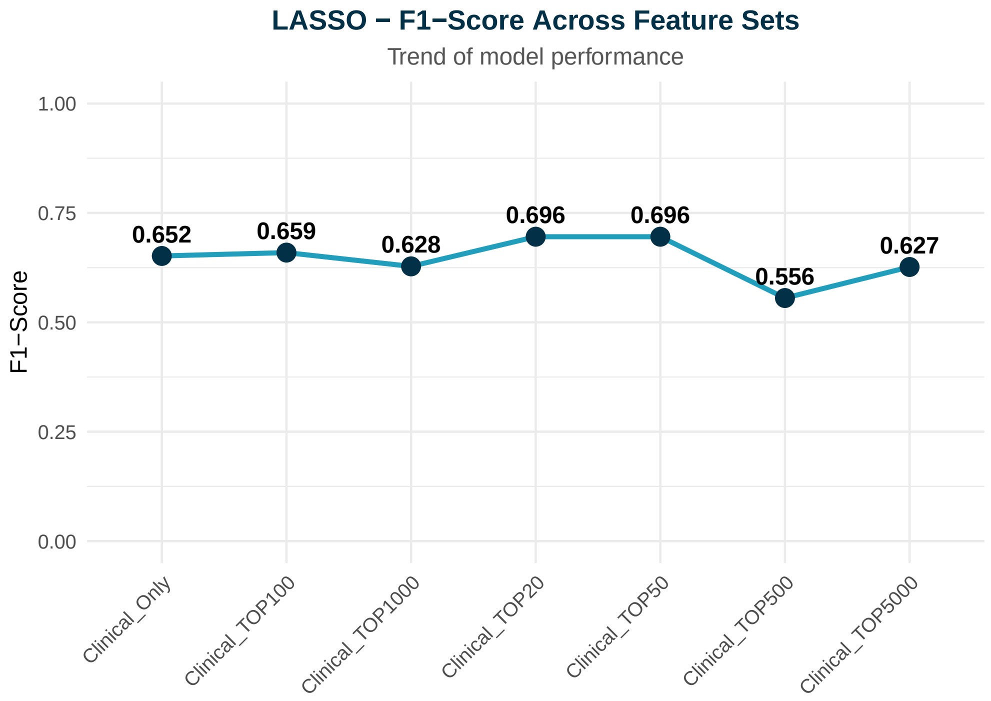
```
lasso_smote_metrics <- plot_classification_metrics_single(lasso_smote
                                                          , threshold = 0.5
                                                          , csv_filename = "lasso_smote_classification_
```

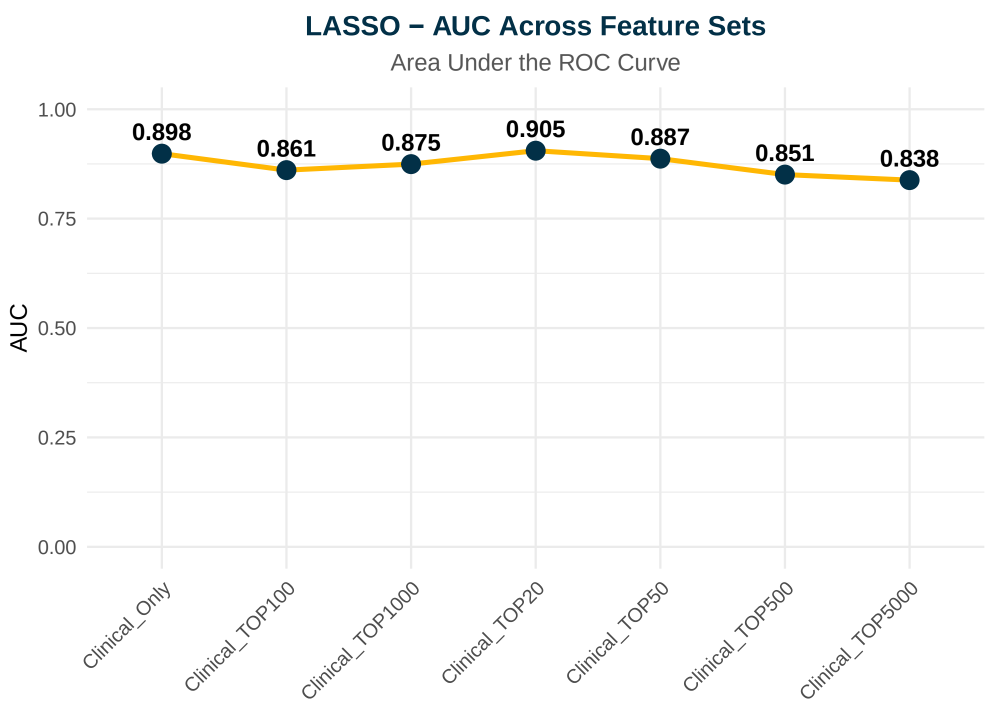
```
##
## === CLASSIFICATION METRICS ===
## Clinical_Only:
##   TP=29 TN=186 FP=20 FN=11
##   Accuracy=0.874 Precision=0.592 Recall=0.725 F1=0.652 AUC=0.898

## Clinical_TOP5000:
##   TP=26 TN=189 FP=17 FN=14
##   Accuracy=0.874 Precision=0.605 Recall=0.650 F1=0.627 AUC=0.838

## Clinical_TOP1000:
##   TP=27 TN=187 FP=19 FN=13
##   Accuracy=0.870 Precision=0.587 Recall=0.675 F1=0.628 AUC=0.875

## Clinical_TOP500:
##   TP=25 TN=181 FP=25 FN=15
##   Accuracy=0.837 Precision=0.500 Recall=0.625 F1=0.556 AUC=0.851

## Clinical_TOP100:
##   TP=30 TN=185 FP=21 FN=10
##   Accuracy=0.874 Precision=0.588 Recall=0.750 F1=0.659 AUC=0.861

## Clinical_TOP50:
##   TP=32 TN=186 FP=20 FN=8
##   Accuracy=0.886 Precision=0.615 Recall=0.800 F1=0.696 AUC=0.887

## Clinical_TOP20:
##   TP=32 TN=186 FP=20 FN=8
##   Accuracy=0.886 Precision=0.615 Recall=0.800 F1=0.696 AUC=0.905
```


LASSO − Confusion Matrix Across Feature Sets

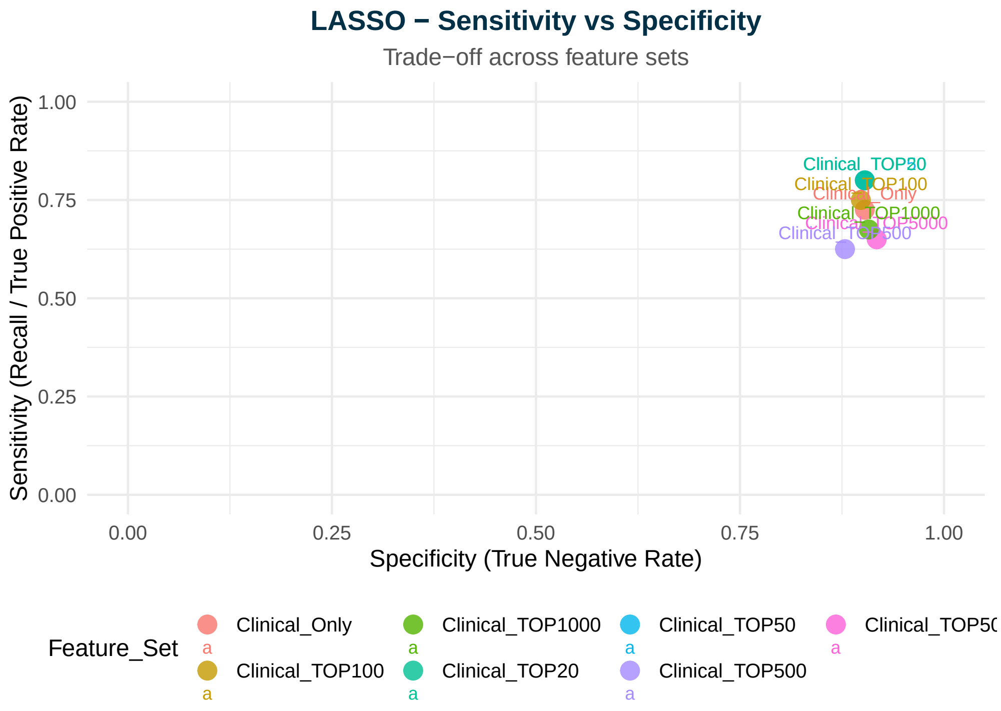
TP, TN, FP, FN counts


LASSO − Classification Metrics
Accuracy, Precision, Recall, F1−Score, AUC
Score
0.0
0.3
0.6
0.9
0.87
0.59
0.72
0.65
0.90
0.87
0.59
0.75
0.66
0.86
0.87
0.59
0.68
0.63
0.87
0.89
0.62
0.80
0.70
0.91
0.89
0.62
0.80
0.70
0.89
0.84
0.50
0.62
0.56
0.85
0.87
0.60
0.65
0.63
0.84
Clinical_Only
Clinical_TOP100
Clinical_TOP1000
Clinical_TOP20
Clinical_TOP50
Clinical_TOP500
Clinical_TOP5000
Metric
Accuracy
Precision
Recall
F1_Score
AUC


LASSO − F1−Score Across Feature Sets
Trend of model performance
F1−Score
0.00
0.25
0.50
0.75
1.00
0.652
0.659
0.628
0.696
0.696
0.556
0.627
Clinical_Only
Clinical_TOP100
Clinical_TOP1000
Clinical_TOP20
Clinical_TOP50
Clinical_TOP500
Clinical_TOP5000


LASSO − AUC Across Feature Sets

Area Under the ROC Curve


LASSO − Sensitivity vs Specificity

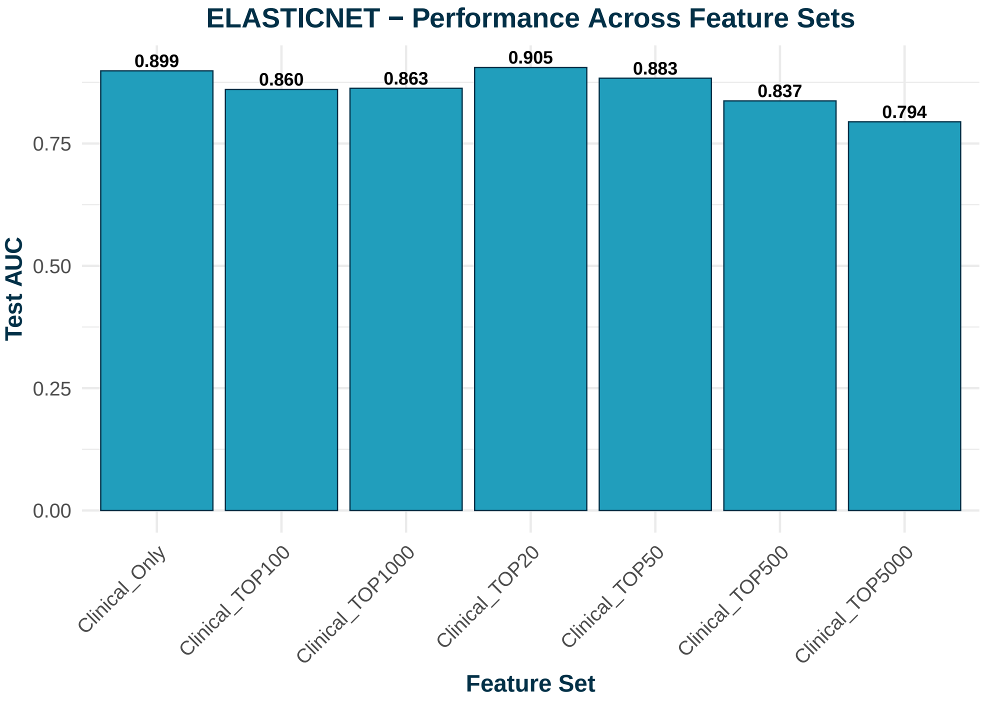
Trade−off across feature sets

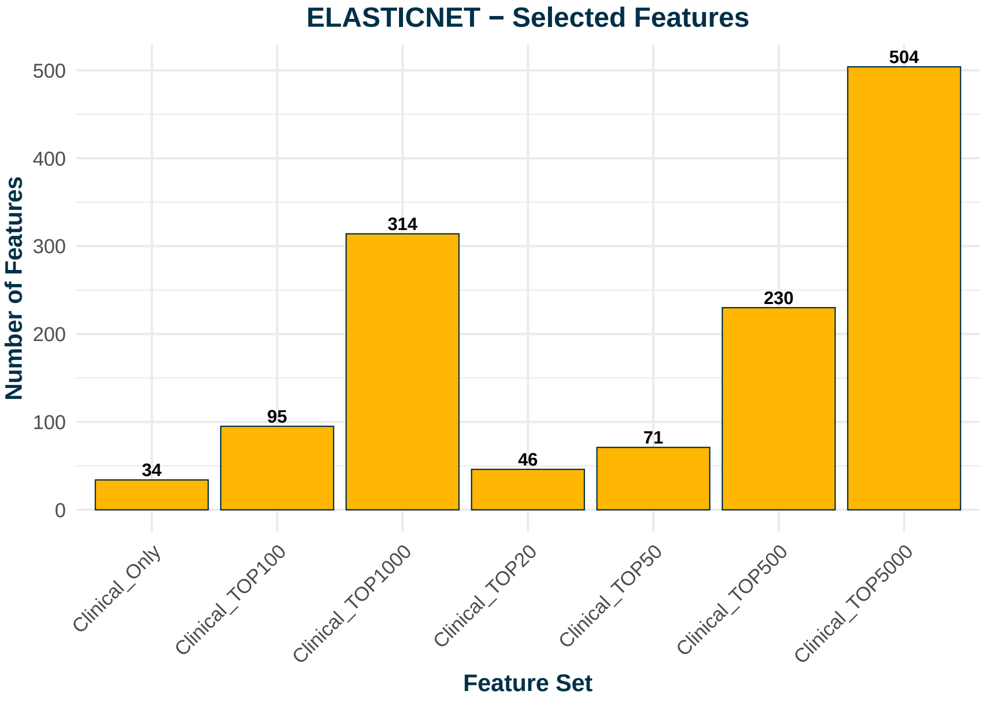
```
##
## === SUMMARY TABLE ===
##        Feature_Set TP  TN FP FN  Accuracy Precision Recall Specificity
## 1    Clinical_Only 29 186 20 11 0.8739837 0.5918367  0.725   0.9029126
## 2 Clinical_TOP5000 26 189 17 14 0.8739837 0.6046512  0.650   0.9174757
## 3 Clinical_TOP1000 27 187 19 13 0.8699187 0.5869565  0.675   0.9077670
## 4  Clinical_TOP500 25 181 25 15 0.8373984 0.5000000  0.625   0.8786408
## 5  Clinical_TOP100 30 185 21 10 0.8739837 0.5882353  0.750   0.8980583
## 6   Clinical_TOP50 32 186 20  8 0.8861789 0.6153846  0.800   0.9029126
## 7   Clinical_TOP20 32 186 20  8 0.8861789 0.6153846  0.800   0.9029126
##    F1_Score       AUC
## 1 0.6516854 0.8984223
## 2 0.6265060 0.8379854
## 3 0.6279070 0.8745146
## 4 0.5555556 0.8506068
## 5 0.6593407 0.8608010
## 6 0.6956522 0.8870146
## 7 0.6956522 0.9052184
##
## Exported classification metrics to: model_metrics/lasso_smote_classification_metrics.csv
```

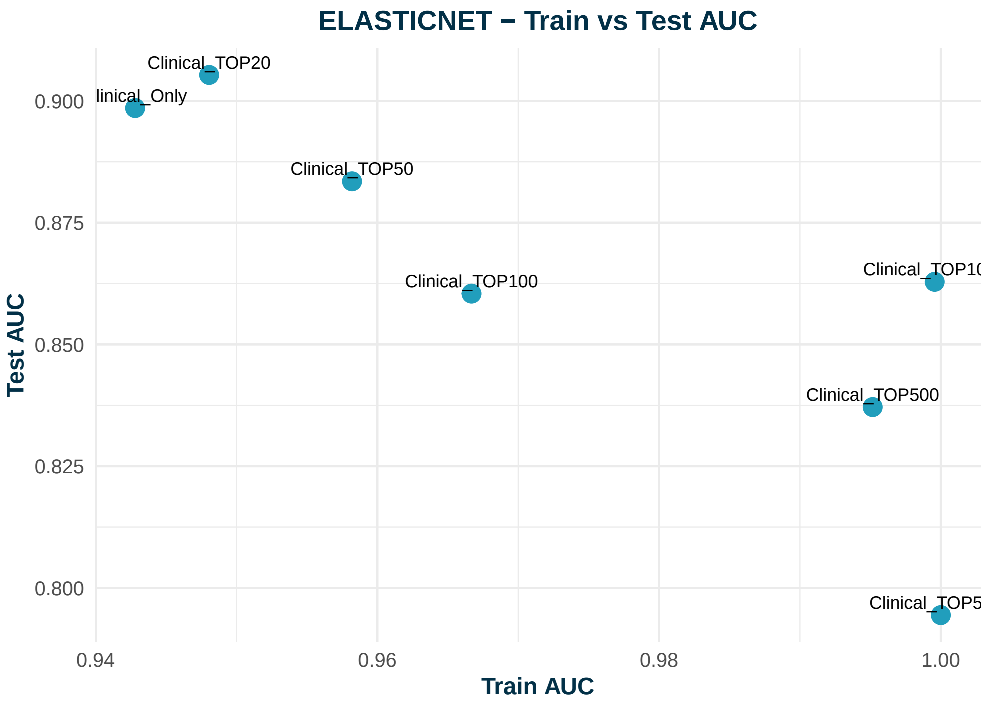
ElasticNet with SMOTE

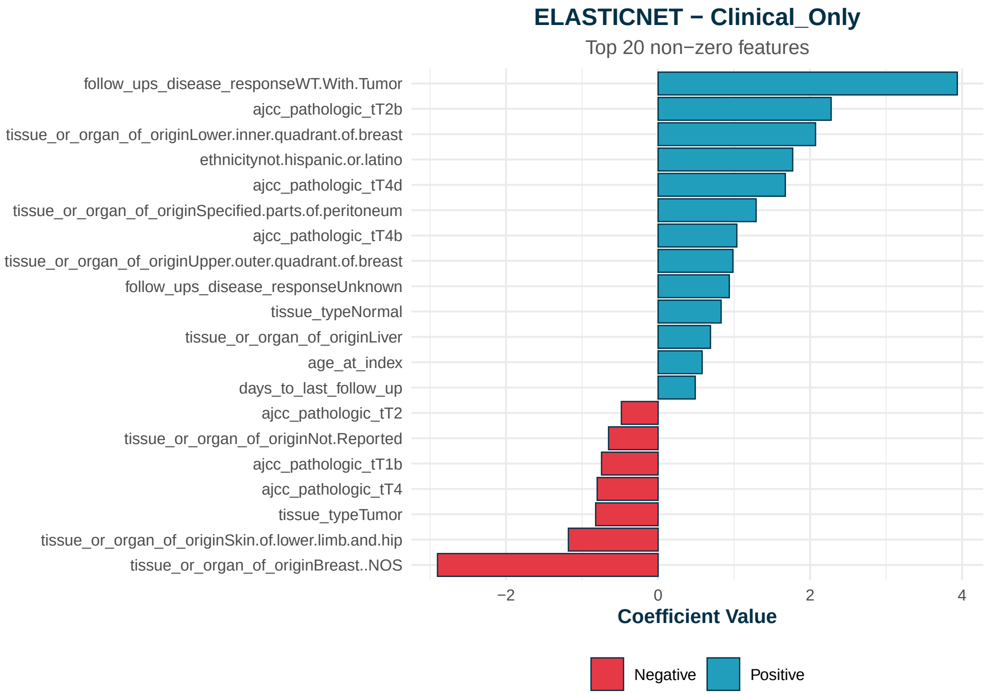
```r
elasticnet_smote <- fit_single_model_across_features(
    model_type = "elasticnet"
    , X_train_all = smote_data$X_train
    , X_test_all = X_test
    , Y_train = smote_data$Y_train
    , Y_test = Y_test
    , n_clinical = n_clinical
    , top_genes_ranked = top_genes
    , gene_sets = c(5000, 1000, 500, 100, 50, 20)
)
```

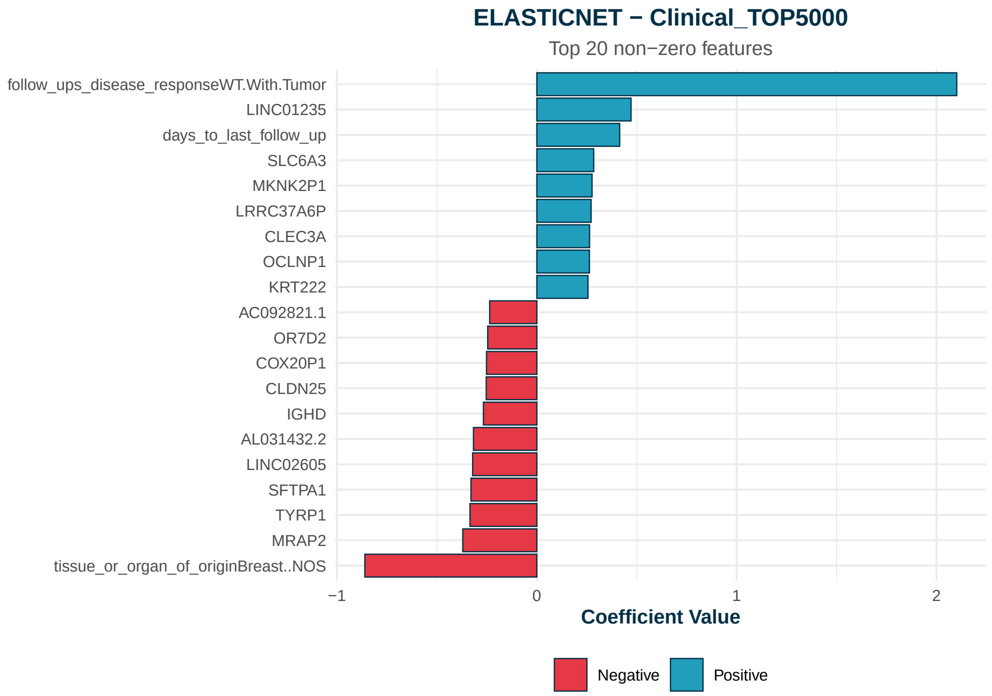
```
##
## === FITTING ELASTICNET ACROSS FEATURE SETS ===
##
## Fitting Clinical_Only...

## Fitting Clinical_TOP5000...

## Fitting Clinical_TOP1000...

## Fitting Clinical_TOP500...

## Fitting Clinical_TOP100...

## Fitting Clinical_TOP50...

## Fitting Clinical_TOP20...

##
## === SUMMARY TABLE ===
##        Feature_Set      Model Features Train_AUC  Test_AUC Test_Accuracy
## 1    Clinical_Only ELASTICNET       34 0.9428043 0.8985437     0.8699187
## 2 Clinical_TOP5000 ELASTICNET      504 1.0000000 0.7944175     0.8455285
## 3 Clinical_TOP1000 ELASTICNET      314 0.9995623 0.8628641     0.8699187
## 4  Clinical_TOP500 ELASTICNET      230 0.9951624 0.8371359     0.8373984
## 5  Clinical_TOP100 ELASTICNET       95 0.9666845 0.8604369     0.8739837
## 6   Clinical_TOP50 ELASTICNET       71 0.9582017 0.8834951     0.8739837
## 7   Clinical_TOP20 ELASTICNET       46 0.9480585 0.9053398     0.8943089
## Exported metrics to: model_metrics/elasticnet_across_features_metrics.csv
```

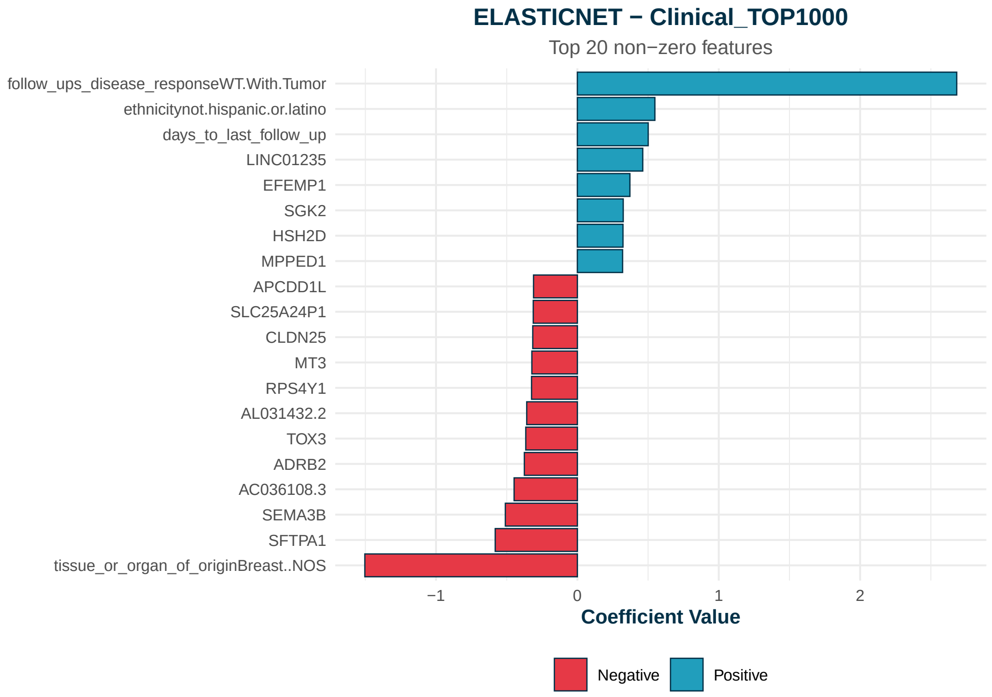


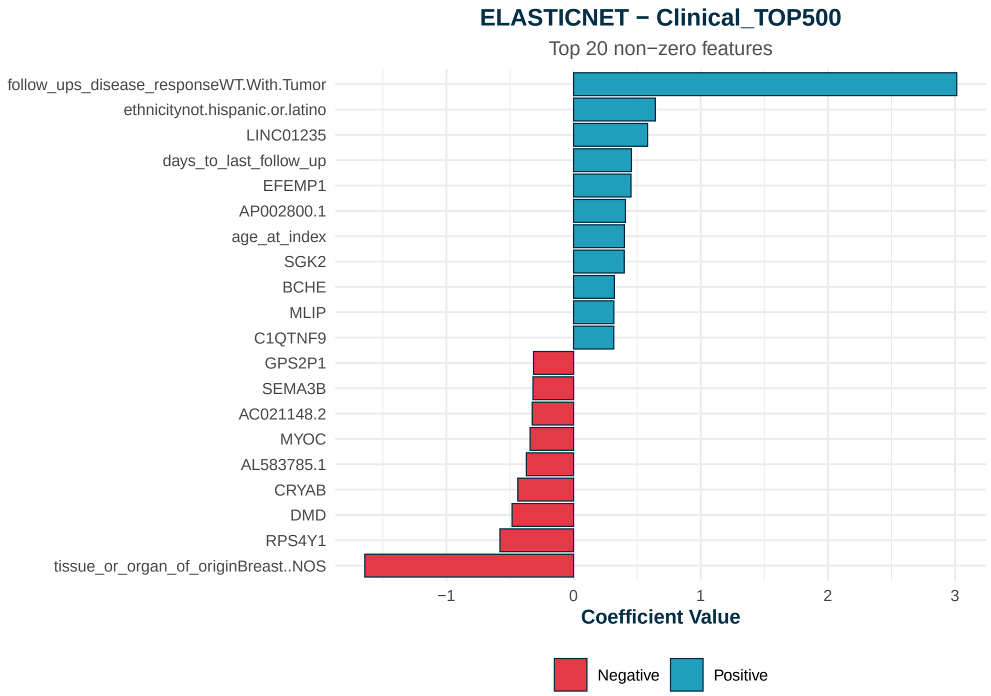


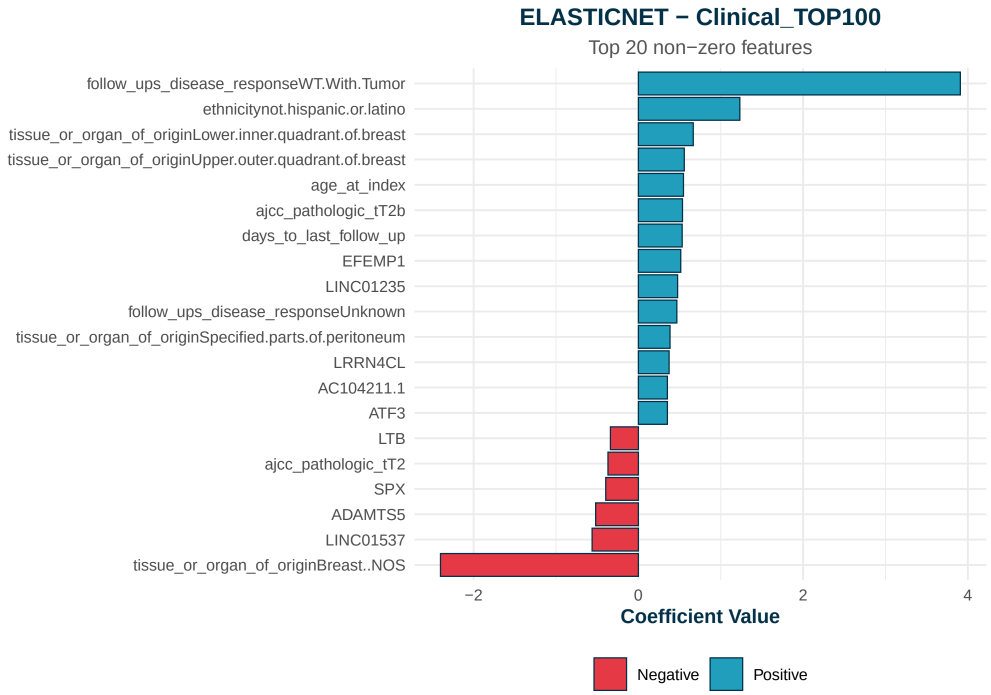

ELASTICNET − Train vs Test AUC
Test AUC
Train AUC
0.800
0.825
0.850
0.875
0.900
0.94
0.96
0.98
1.00
Clinical_TOP20
linical_Only
Clinical_TOP50
Clinical_TOP100
Clinical_TOP1(
Clinical_TOP500
Clinical_TOP5

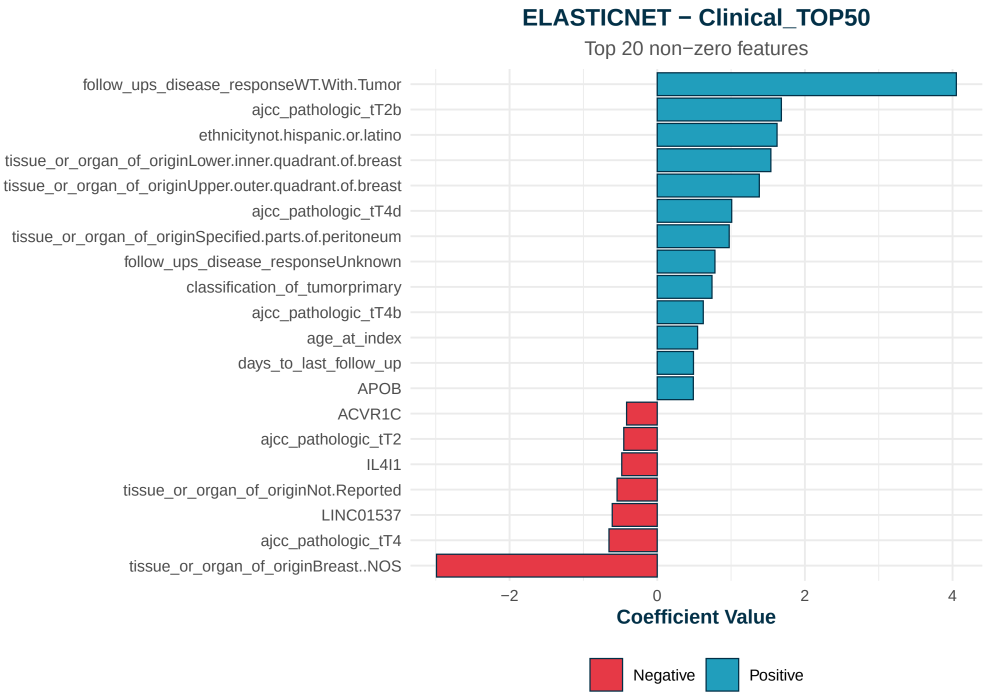

ELASTICNET − Clinical_Only
Top 20 non−zero features
follow_ups_disease_responseWT.With.Tumor
ajcc_pathologic_tT2b
tissue_or_organ_of_originLower.inner.quadrant.of.breast
ethnicitynot.hispanic.or.latino
ajcc_pathologic_tT4d
tissue_or_organ_of_originSpecified.parts.of.peritoneum
ajcc_pathologic_tT4b
tissue_or_organ_of_originUpper.outer.quadrant.of.breast
follow_ups_disease_responseUnknown
tissue_typeNormal
tissue_or_organ_of_originLiver
age_at_index
days_to_last_follow_up
ajcc_pathologic_tT2
tissue_or_organ_of_originNot.Reported
ajcc_pathologic_tT1b
ajcc_pathologic_tT4
tissue_typeTumor
tissue_or_organ_of_originSkin.of.lower.limb.and.hip
tissue_or_organ_of_originBreast..NOS
−2
0
2
4
Coefficient Value
Negative
Positive

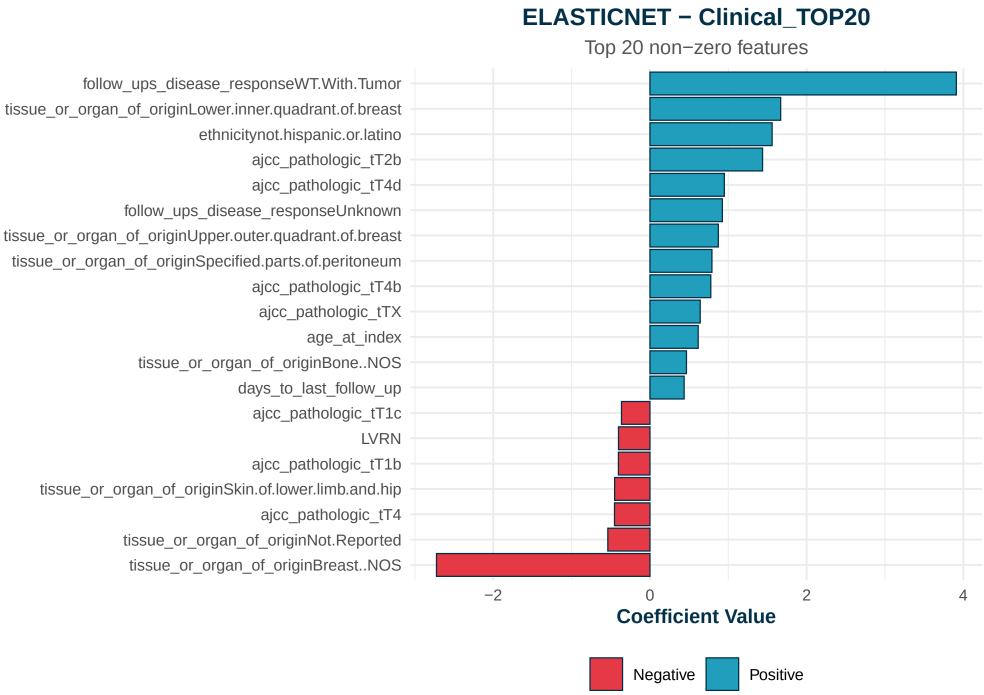

ELASTICNET − Clinical_TOP5000
Top 20 non−zero features
follow_ups_disease_responseWT.With.Tumor
LINC01235
days_to_last_follow_up
SLC6A3
MKNK2P1
LRRC37A6P
CLEC3A
OCLNP1
KRT222
AC092821.1
OR7D2
COX20P1
CLDN25
IGHD
AL031432.2
LINC02605
SFTPA1
TYRP1
MRAP2
tissue_or_organ_of_originBreast..NOS
−1
0
1
2
Coefficient Value
Negative
Positive


ELASTICNET − Clinical_TOP1000
Top 20 non−zero features
follow_ups_disease_responseWT.With.Tumor
ethnicitynot.hispanic.or.latino
days_to_last_follow_up
LINC01235
EFEMP1
SGK2
HSH2D
MPPED1
APCDD1L
SLC25A24P1
CLDN25
MT3
RPS4Y1
AL031432.2
TOX3
ADRB2
AC036108.3
SEMA3B
SFTPA1
tissue_or_organ_of_originBreast..NOS
−1
0
1
2
Coefficient Value
Negative
Positive


ELASTICNET − Clinical_TOP500
Top 20 non−zero features
follow_ups_disease_responseWT.With.Tumor
ethnicitynot.hispanic.or.latino
LINC01235
days_to_last_follow_up
EFEMP1
AP002800.1
age_at_index
SGK2
BCHE
MLIP
C1QTNF9
GPS2P1
SEMA3B
AC021148.2
MYOC
AL583785.1
CRYAB
DMD
RPS4Y1
tissue_or_organ_of_originBreast..NOS
−1
0
1
2
3
Coefficient Value
Negative
Positive

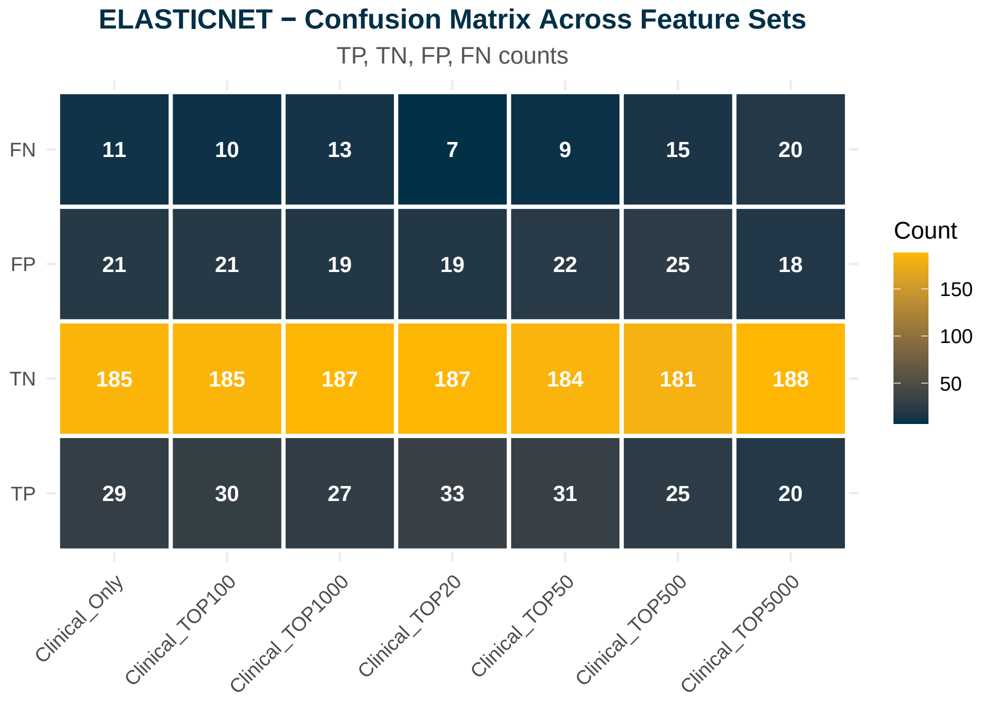

ELASTICNET − Clinical_TOP100
Top 20 non−zero features
follow_ups_disease_responseWT.With.Tumor
ethnicitynot.hispanic.or.latino
tissue_or_organ_of_originLower.inner.quadrant.of.breast
tissue_or_organ_of_originUpper.outer.quadrant.of.breast
age_at_index
ajcc_pathologic_tT2b
days_to_last_follow_up
EFEMP1
LINC01235
follow_ups_disease_responseUnknown
tissue_or_organ_of_originSpecified.parts.of.peritoneum
LRRN4CL
AC104211.1
ATF3
LTB
ajcc_pathologic_tT2
SPX
ADAMTS5
LINC01537
tissue_or_organ_of_originBreast..NOS
−2
0
2
4
Coefficient Value
Negative
Positive

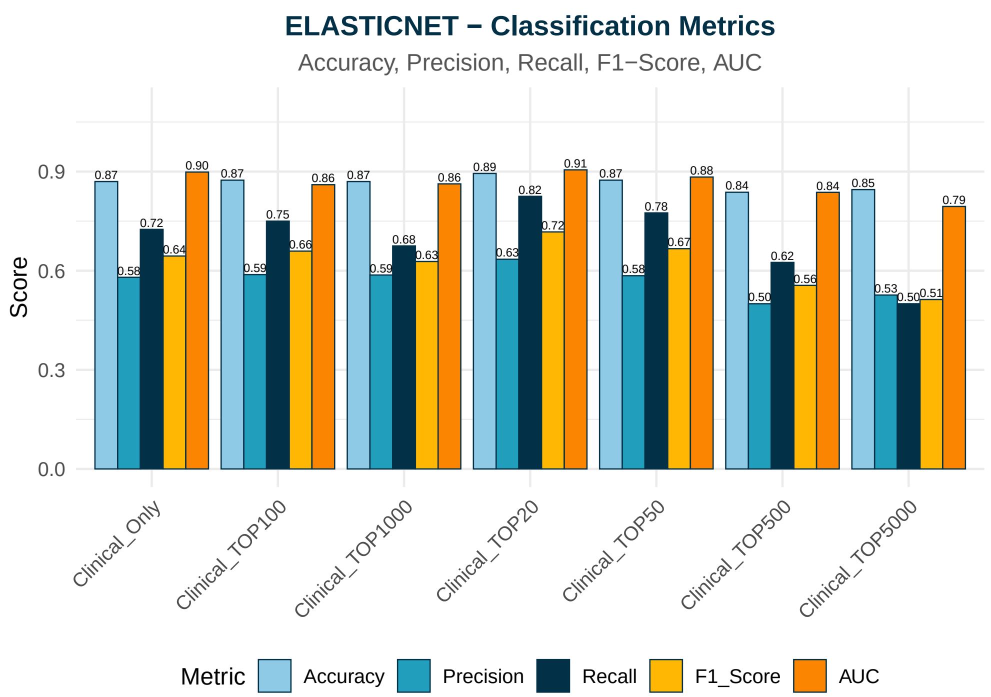


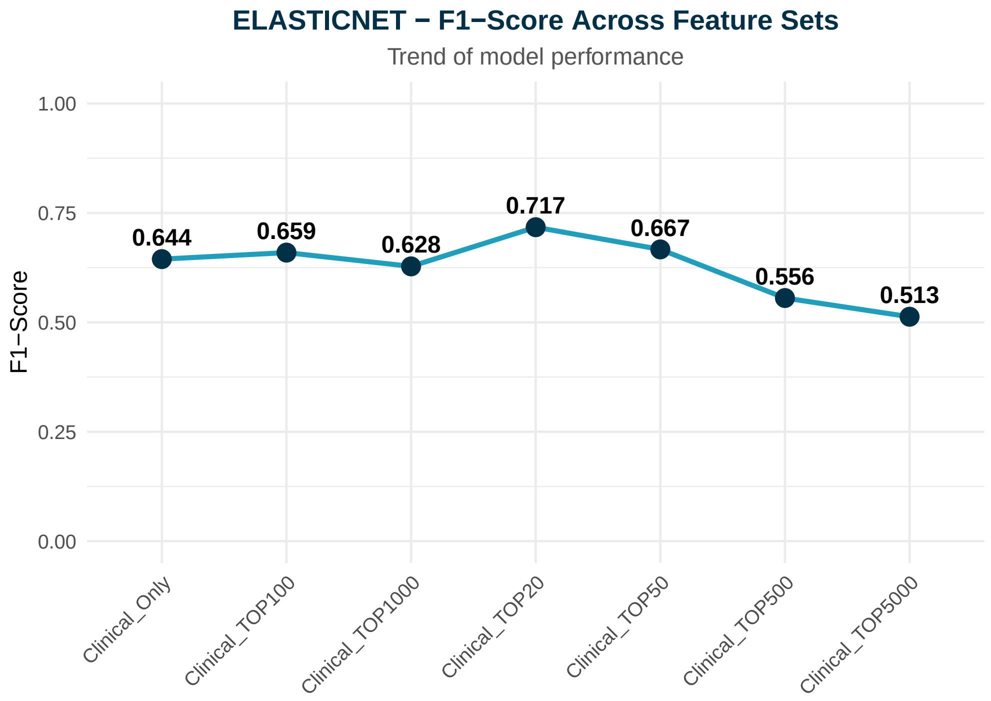


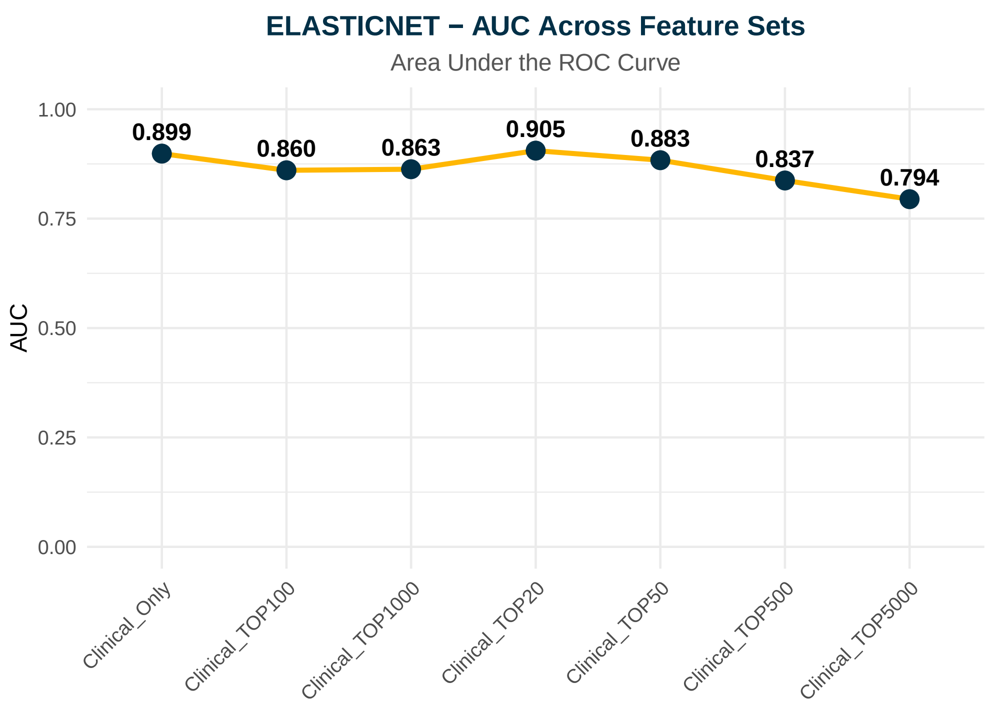
```
elasticnet_smote_metrics <- plot_classification_metrics_single(elasticnet_smote
                                                               , threshold = 0.5
                                                               , csv_filename = "elasticnet_smote_class
```

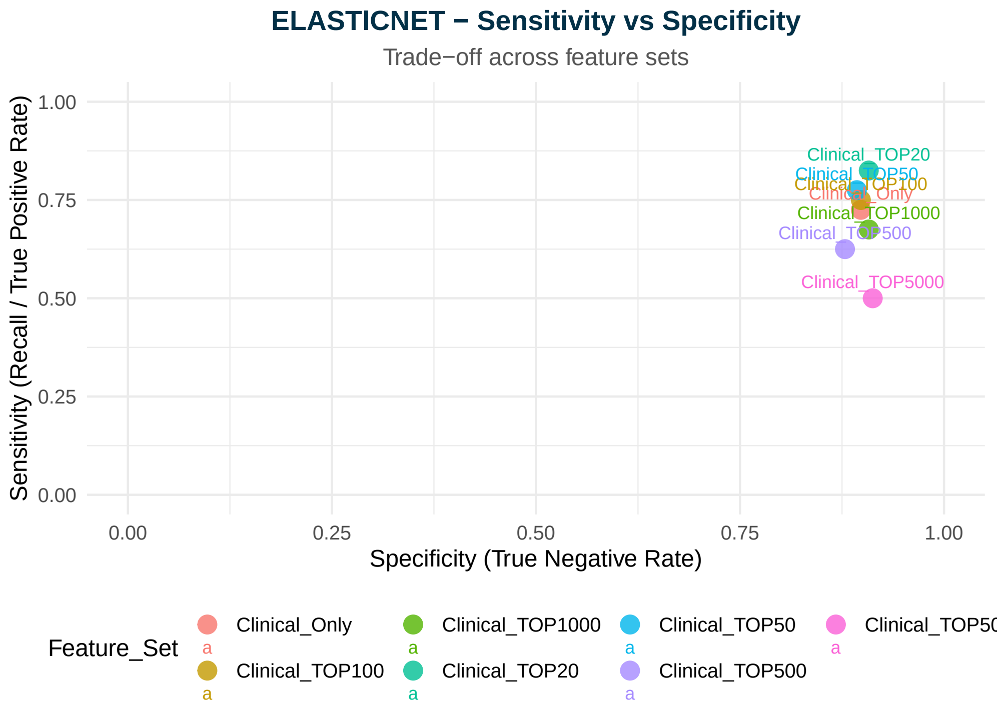
```
##
## === CLASSIFICATION METRICS ===
## Clinical_Only:
##   TP=29 TN=185 FP=21 FN=11
##   Accuracy=0.870 Precision=0.580 Recall=0.725 F1=0.644 AUC=0.899

## Clinical_TOP5000:
##   TP=20 TN=188 FP=18 FN=20
##   Accuracy=0.846 Precision=0.526 Recall=0.500 F1=0.513 AUC=0.794

## Clinical_TOP1000:
##   TP=27 TN=187 FP=19 FN=13
##   Accuracy=0.870 Precision=0.587 Recall=0.675 F1=0.628 AUC=0.863

## Clinical_TOP500:
##   TP=25 TN=181 FP=25 FN=15
##   Accuracy=0.837 Precision=0.500 Recall=0.625 F1=0.556 AUC=0.837

## Clinical_TOP100:
##   TP=30 TN=185 FP=21 FN=10
##   Accuracy=0.874 Precision=0.588 Recall=0.750 F1=0.659 AUC=0.860

## Clinical_TOP50:
##   TP=31 TN=184 FP=22 FN=9
##   Accuracy=0.874 Precision=0.585 Recall=0.775 F1=0.667 AUC=0.883

## Clinical_TOP20:
##   TP=33 TN=187 FP=19 FN=7
##   Accuracy=0.894 Precision=0.635 Recall=0.825 F1=0.717 AUC=0.905
```




ELASTICNET − Classification Metrics
Accuracy, Precision, Recall, F1−Score, AUC
Score
0.0
0.3
0.6
0.9
0.87
0.58
0.72
0.64
0.90
0.87
0.59
0.75
0.66
0.86
0.87
0.59
0.68
0.63
0.86
0.89
0.63
0.82
0.72
0.91
0.87
0.58
0.78
0.67
0.88
0.84
0.50
0.62
0.56
0.84
0.85
0.53
0.50
0.51
0.79
Clinical_Only
Clinical_TOP100
Clinical_TOP1000
Clinical_TOP20
Clinical_TOP50
Clinical_TOP500
Clinical_TOP5000
Metric
Accuracy
Precision
Recall
F1_Score
AUC


ELASTICNET − F1−Score Across Feature Sets
Trend of model performance
F1−Score
0.00
0.25
0.50
0.75
1.00
0.644
0.659
0.628
0.717
0.667
0.556
0.513
Clinical_Only
Clinical_TOP100
Clinical_TOP1000
Clinical_TOP20
Clinical_TOP50
Clinical_TOP500
Clinical_TOP5000



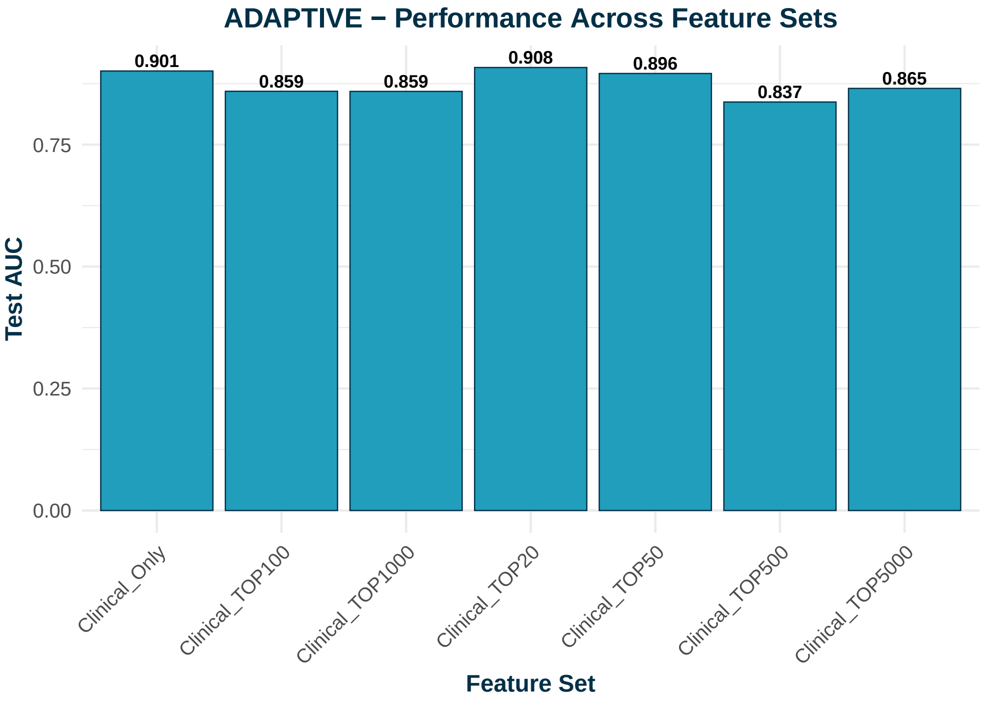


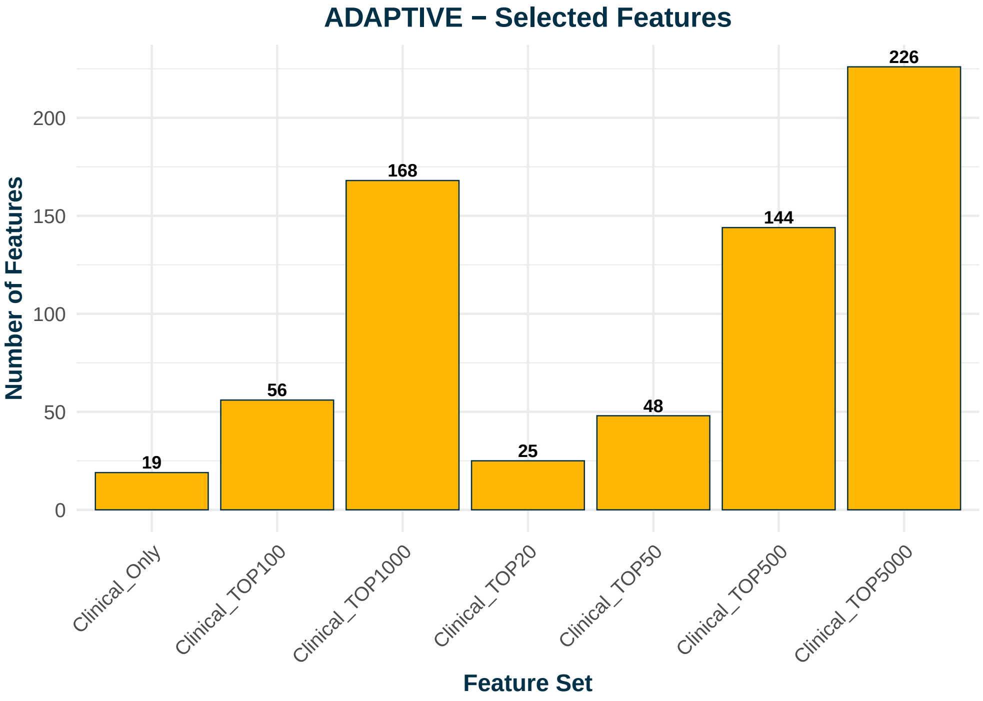
```
##
## === SUMMARY TABLE ===
##        Feature_Set TP  TN FP FN  Accuracy Precision Recall Specificity
## 1    Clinical_Only 29 185 21 11 0.8699187 0.5800000  0.725   0.8980583
## 2 Clinical_TOP5000 20 188 18 20 0.8455285 0.5263158  0.500   0.9126214
## 3 Clinical_TOP1000 27 187 19 13 0.8699187 0.5869565  0.675   0.9077670
## 4  Clinical_TOP500 25 181 25 15 0.8373984 0.5000000  0.625   0.8786408
## 5  Clinical_TOP100 30 185 21 10 0.8739837 0.5882353  0.750   0.8980583
## 6   Clinical_TOP50 31 184 22  9 0.8739837 0.5849057  0.775   0.8932039
## 7   Clinical_TOP20 33 187 19  7 0.8943089 0.6346154  0.825   0.9077670
##    F1_Score       AUC
## 1 0.6444444 0.8985437
## 2 0.5128205 0.7944175
## 3 0.6279070 0.8628641
## 4 0.5555556 0.8371359
## 5 0.6593407 0.8604369
## 6 0.6666667 0.8834951
## 7 0.7173913 0.9053398
##
## Exported classification metrics to: model_metrics/elasticnet_smote_classification_metrics.csv
```

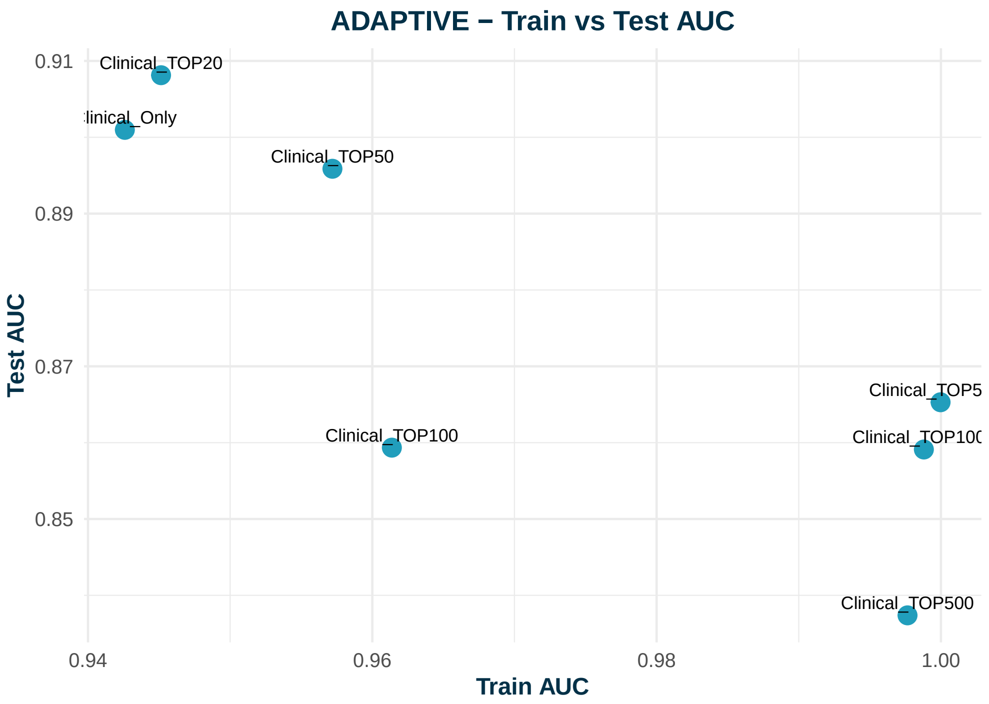
Adaptive Lasso with SMOTE

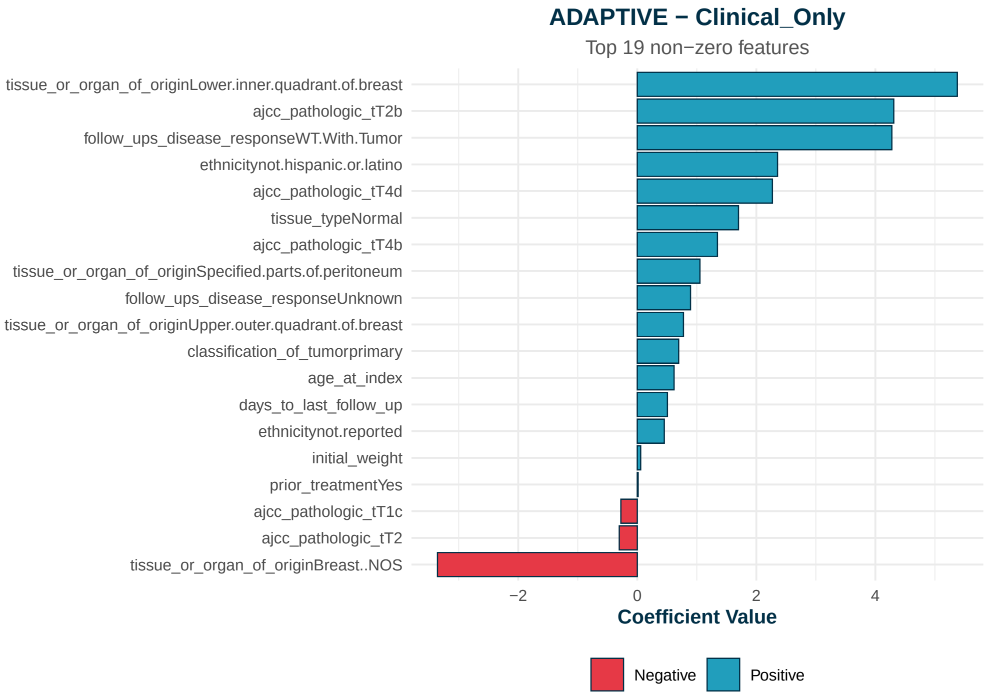
```r
adaptive_lasso_smote <- fit_single_model_across_features(
    model_type = "adaptive"
    , X_train_all = smote_data$X_train
    , X_test_all = X_test
    , Y_train = smote_data$Y_train
    , Y_test = Y_test
    , n_clinical = n_clinical
    , top_genes_ranked = top_genes
    , gene_sets = c(5000, 1000, 500, 100, 50, 20)
)
```

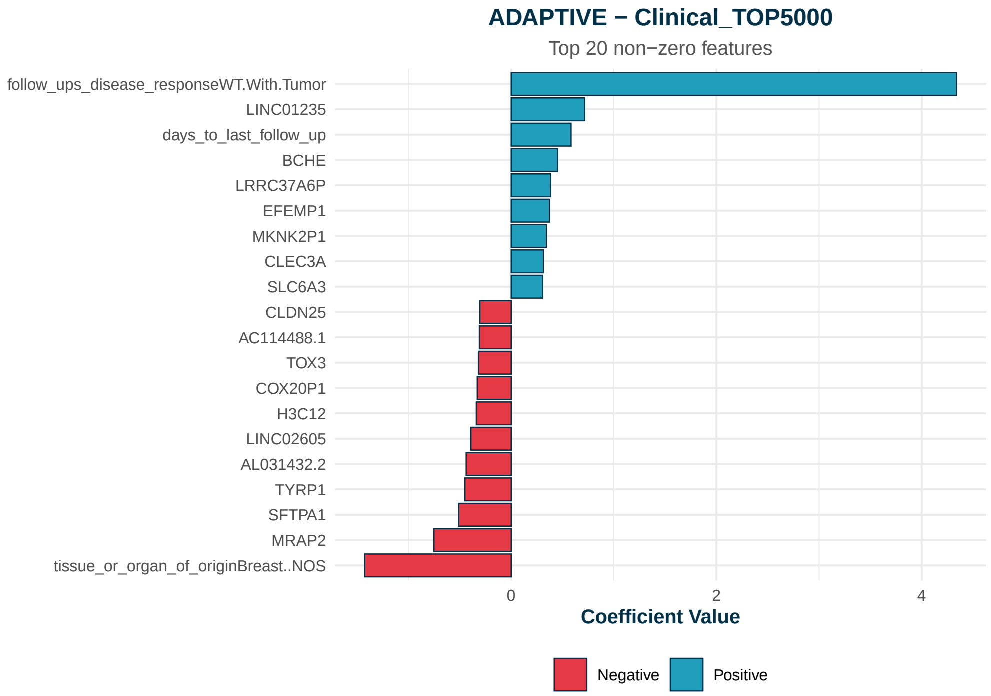
```
##
## === FITTING ADAPTIVE ACROSS FEATURE SETS ===
##
## Fitting Clinical_Only...

## Fitting Clinical_TOP5000...

## Fitting Clinical_TOP1000...

## Fitting Clinical_TOP500...

## Fitting Clinical_TOP100...

## Fitting Clinical_TOP50...

## Fitting Clinical_TOP20...

##
## === SUMMARY TABLE ===
##        Feature_Set    Model Features Train_AUC  Test_AUC Test_Accuracy
## 1    Clinical_Only ADAPTIVE       19 0.9425915 0.9009709     0.8821138
## 2 Clinical_TOP5000 ADAPTIVE      226 0.9999758 0.8652913     0.8861789
## 3 Clinical_TOP1000 ADAPTIVE      168 0.9988015 0.8591019     0.8739837
## 4  Clinical_TOP500 ADAPTIVE      144 0.9976499 0.8373786     0.8373984
## 5  Clinical_TOP100 ADAPTIVE       56 0.9613760 0.8593447     0.8699187
## 6   Clinical_TOP50 ADAPTIVE       48 0.9571949 0.8958738     0.8983740
## 7   Clinical_TOP20 ADAPTIVE       25 0.9451394 0.9081311     0.8943089
## Exported metrics to: model_metrics/adaptive_across_features_metrics.csv
```

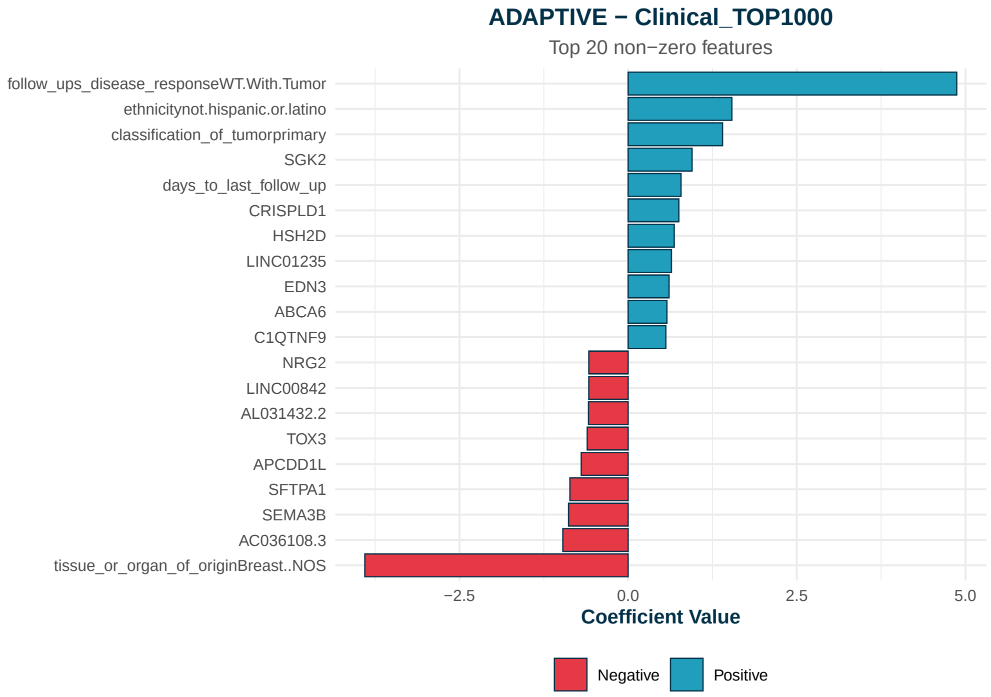


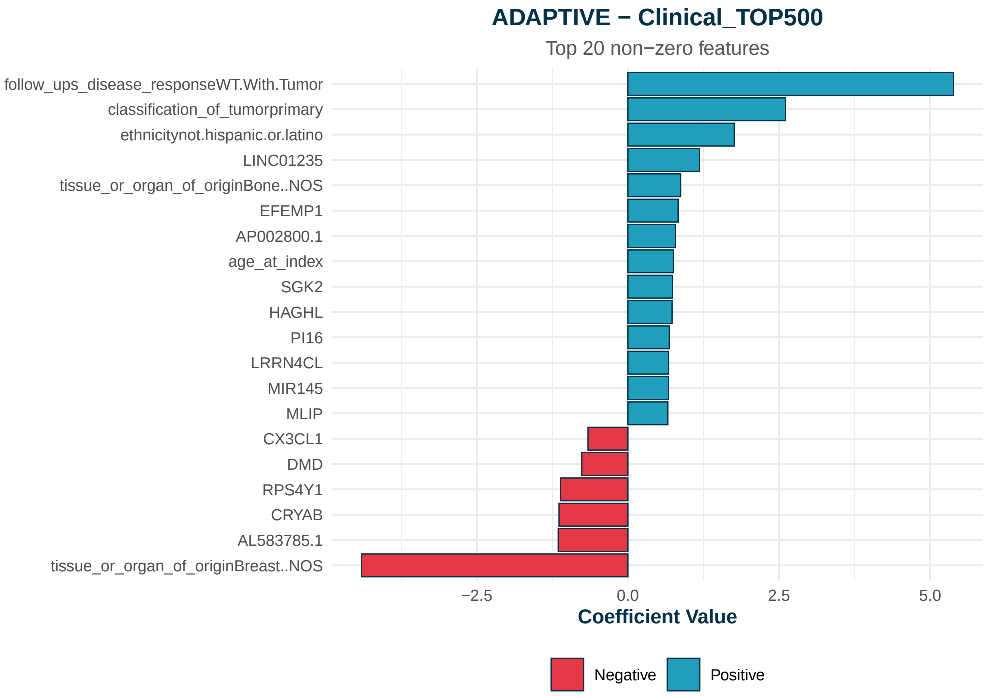


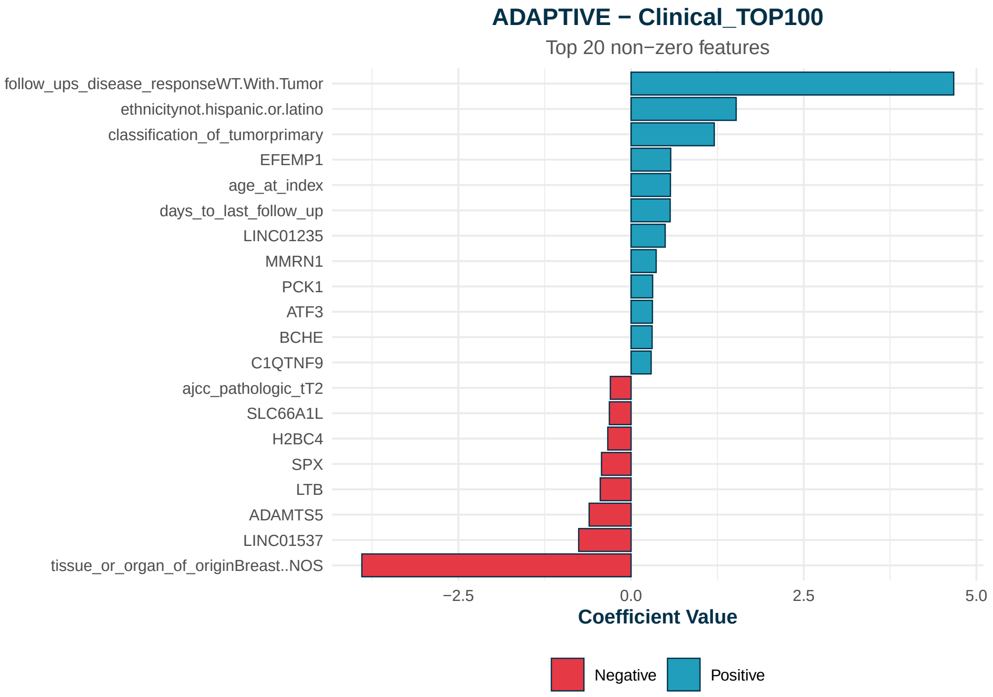

ADAPTIVE − Train vs Test AUC
Test AUC
Train AUC
0.91
0.89
0.87
0.85
0.94
0.96
0.98
1.00
Clinical_TOP20
linical_Only
Clinical_TOP50
Clinical_TOP5
Clinical_TOP100
Clinical_TOP10(
Clinical_TOP500

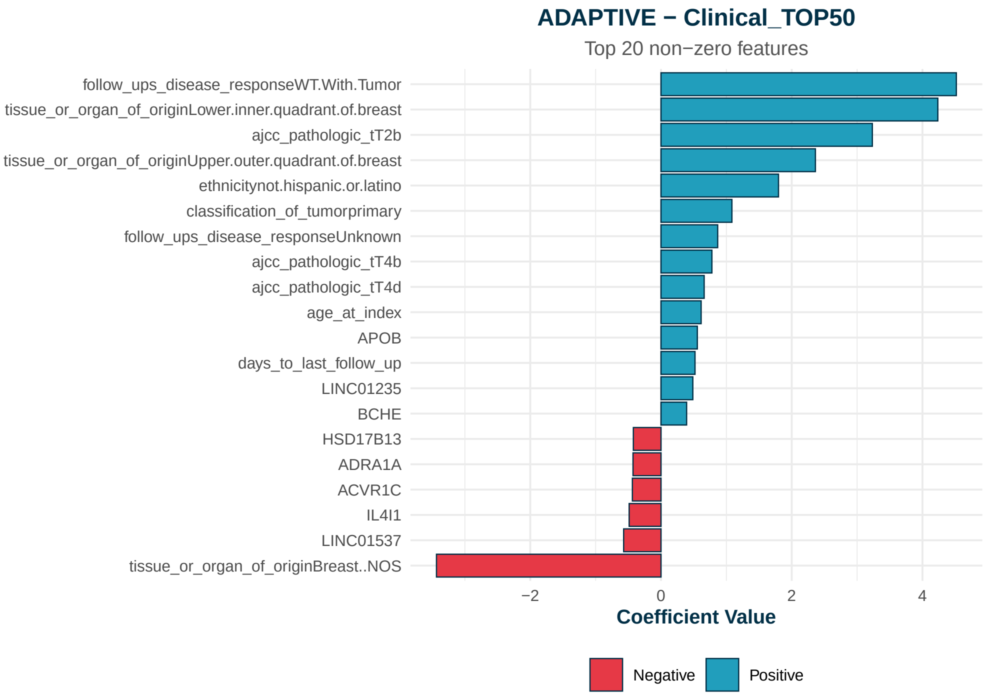

ADAPTIVE − Clinical_Only
Top 19 non−zero features
tissue_or_organ_of_originLower.inner.quadrant.of.breast
ajcc_pathologic_tT2b
follow_ups_disease_responseWT.With.Tumor
ethnicitynot.hispanic.or.latino
ajcc_pathologic_tT4d
tissue_typeNormal
ajcc_pathologic_tT4b
tissue_or_organ_of_originSpecified.parts.of.peritoneum
follow_ups_disease_responseUnknown
tissue_or_organ_of_originUpper.outer.quadrant.of.breast
classification_of_tumorprimary
age_at_index
days_to_last_follow_up
ethnicitynot.reported
initial_weight
prior_treatmentYes
ajcc_pathologic_tT1c
ajcc_pathologic_tT2
tissue_or_organ_of_originBreast..NOS
−2
0
2
4
Coefficient Value
🟥 Negative
🟦 Positive

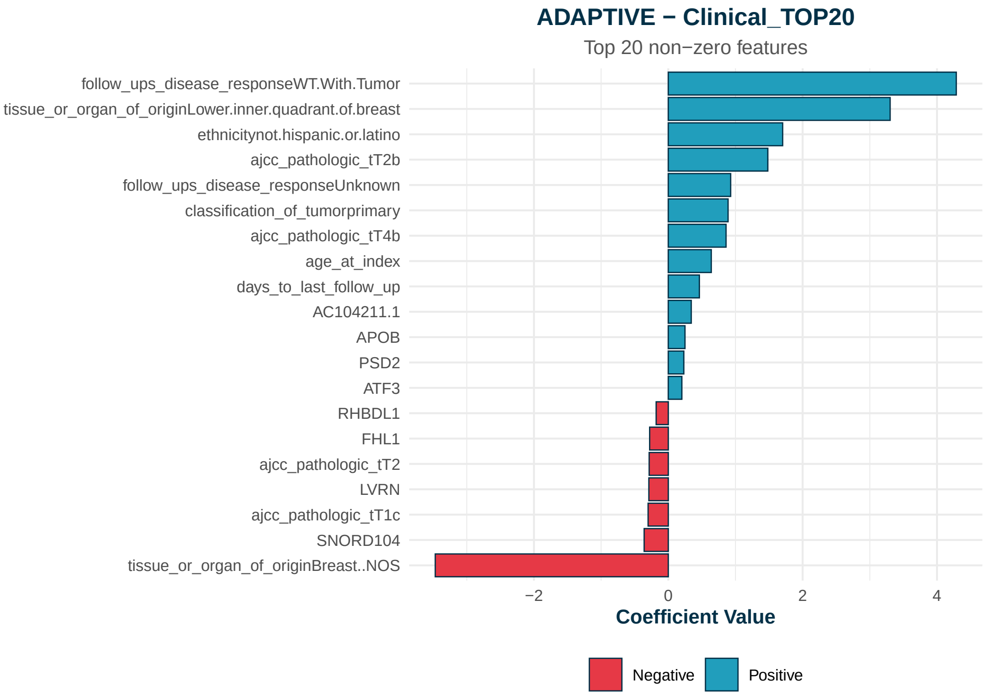

ADAPTIVE − Clinical_TOP5000
Top 20 non−zero features
follow_ups_disease_responseWT.With.Tumor
LINC01235
days_to_last_follow_up
BCHE
LRRC37A6P
EFEMP1
MKNK2P1
CLEC3A
SLC6A3
CLDN25
AC114488.1
TOX3
COX20P1
H3C12
LINC02605
AL031432.2
TYRP1
SFTPA1
MRAP2
tissue_or_organ_of_originBreast..NOS
0
2
4
Coefficient Value
Negative
Positive


ADAPTIVE − Clinical_TOP1000
Top 20 non−zero features
follow_ups_disease_responseWT.With.Tumor
ethnicitynot.hispanic.or.latino
classification_of_tumorprimary
SGK2
days_to_last_follow_up
CRISPLD1
HSH2D
LINC01235
EDN3
ABCA6
C1QTNF9
NRG2
LINC00842
AL031432.2
TOX3
APCDD1L
SFTPA1
SEMA3B
AC036108.3
tissue_or_organ_of_originBreast..NOS
−2.5
0.0
2.5
5.0
Coefficient Value
Negative
Positive


ADAPTIVE − Clinical_TOP500
Top 20 non−zero features
follow_ups_disease_responseWT.With.Tumor
classification_of_tumorprimary
ethnicitynot.hispanic.or.latino
LINC01235
tissue_or_organ_of_originBone..NOS
EFEMP1
AP002800.1
age_at_index
SGK2
HAGHL
PI16
LRRN4CL
MIR145
MLIP
CX3CL1
DMD
RPS4Y1
CRYAB
AL583785.1
tissue_or_organ_of_originBreast..NOS
−2.5
0.0
2.5
5.0
Coefficient Value
Negative
Positive

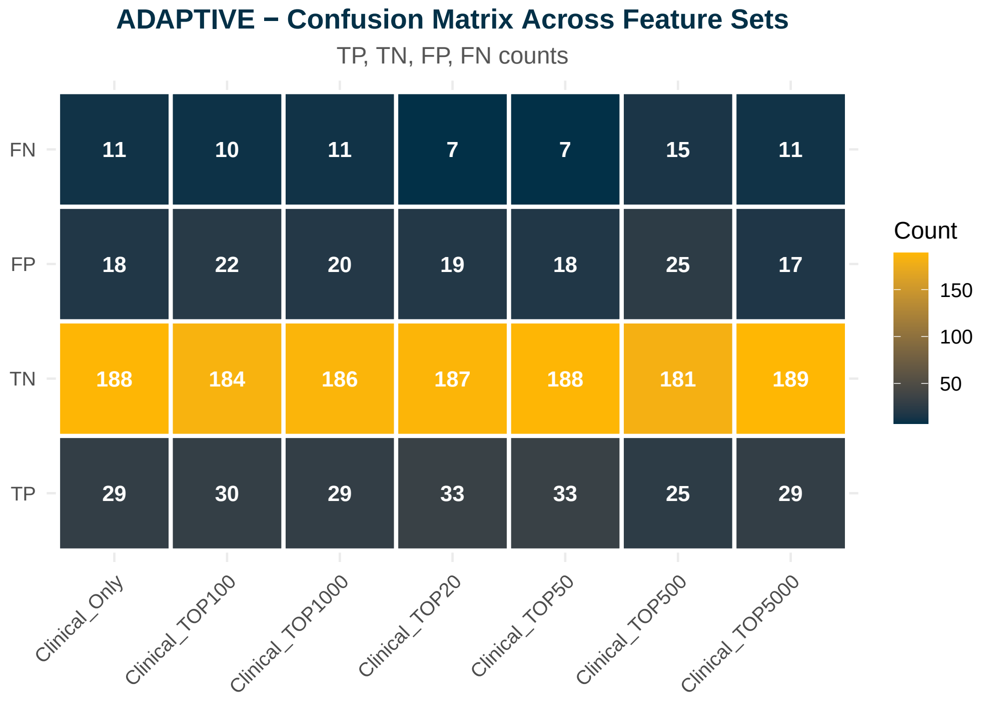

ADAPTIVE − Clinical_TOP100
Top 20 non−zero features
follow_ups_disease_responseWT.With.Tumor
ethnicitynot.hispanic.or.latino
classification_of_tumorprimary
EFEMP1
age_at_index
days_to_last_follow_up
LINC01235
MMRN1
PCK1
ATF3
BCHE
C1QTNF9
ajcc_pathologic_tT2
SLC66A1L
H2BC4
SPX
LTB
ADAMTS5
LINC01537
tissue_or_organ_of_originBreast..NOS
−2.5
0.0
2.5
5.0
Coefficient Value
Negative
Positive

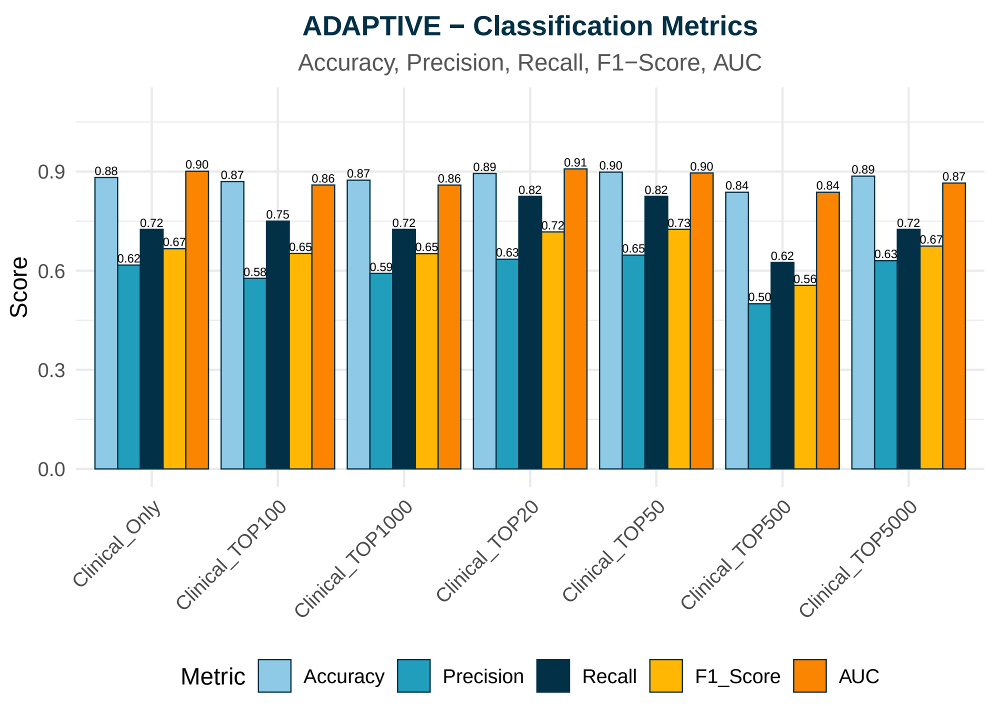




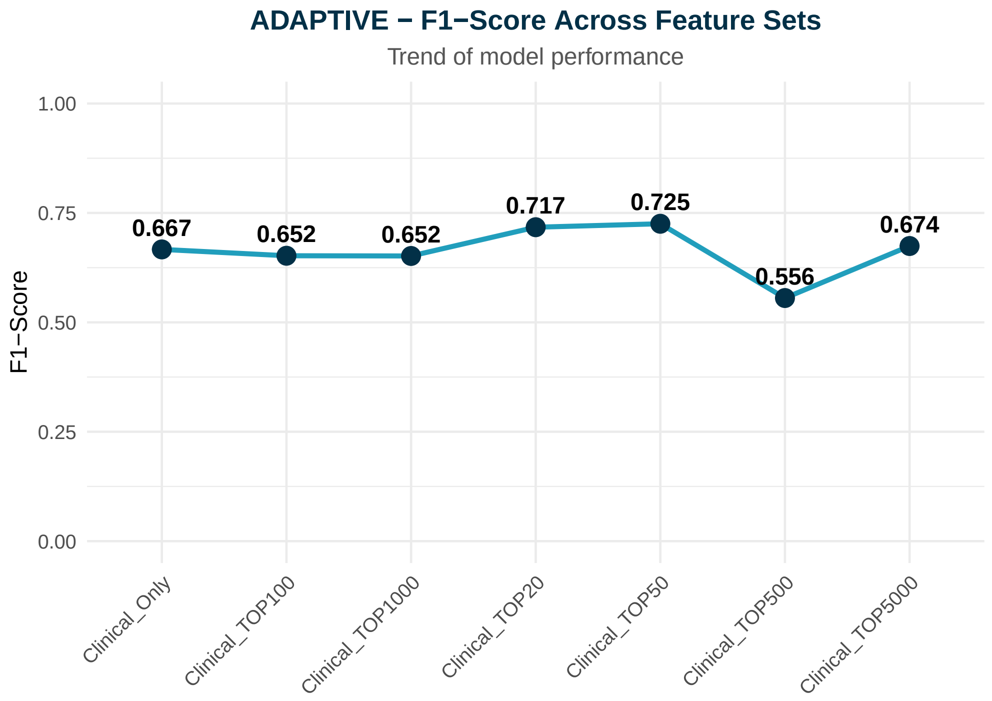
```
adaptive_lasso_smote_metrics <- plot_classification_metrics_single(adaptive_lasso_smote
                                                                   , threshold = 0.5
                                                                   , csv_filename = "adaptive_lasso_smo
```

```
##
## === CLASSIFICATION METRICS ===
## Clinical_Only:
##   TP=29 TN=188 FP=18 FN=11
##   Accuracy=0.882 Precision=0.617 Recall=0.725 F1=0.667 AUC=0.901

## Clinical_TOP5000:
##   TP=29 TN=189 FP=17 FN=11
##   Accuracy=0.886 Precision=0.630 Recall=0.725 F1=0.674 AUC=0.865

## Clinical_TOP1000:
##   TP=29 TN=186 FP=20 FN=11
##   Accuracy=0.874 Precision=0.592 Recall=0.725 F1=0.652 AUC=0.859

## Clinical_TOP500:
##   TP=25 TN=181 FP=25 FN=15
##   Accuracy=0.837 Precision=0.500 Recall=0.625 F1=0.556 AUC=0.837

## Clinical_TOP100:
##   TP=30 TN=184 FP=22 FN=10
##   Accuracy=0.870 Precision=0.577 Recall=0.750 F1=0.652 AUC=0.859

## Clinical_TOP50:
##   TP=33 TN=188 FP=18 FN=7
##   Accuracy=0.898 Precision=0.647 Recall=0.825 F1=0.725 AUC=0.896

## Clinical_TOP20:
##   TP=33 TN=187 FP=19 FN=7
##   Accuracy=0.894 Precision=0.635 Recall=0.825 F1=0.717 AUC=0.908
```

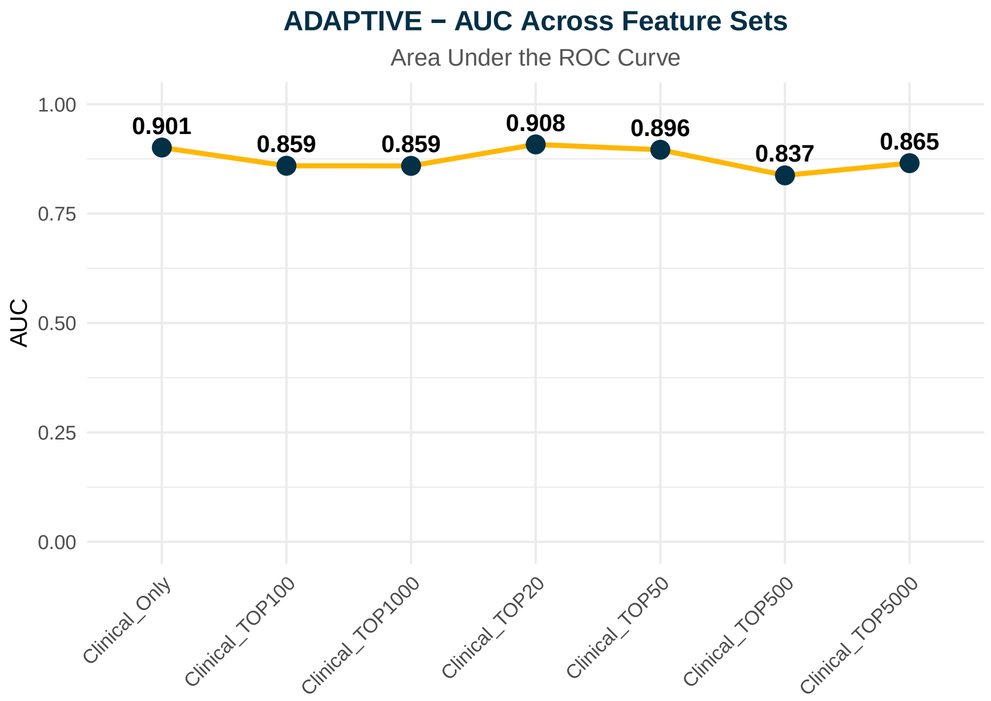


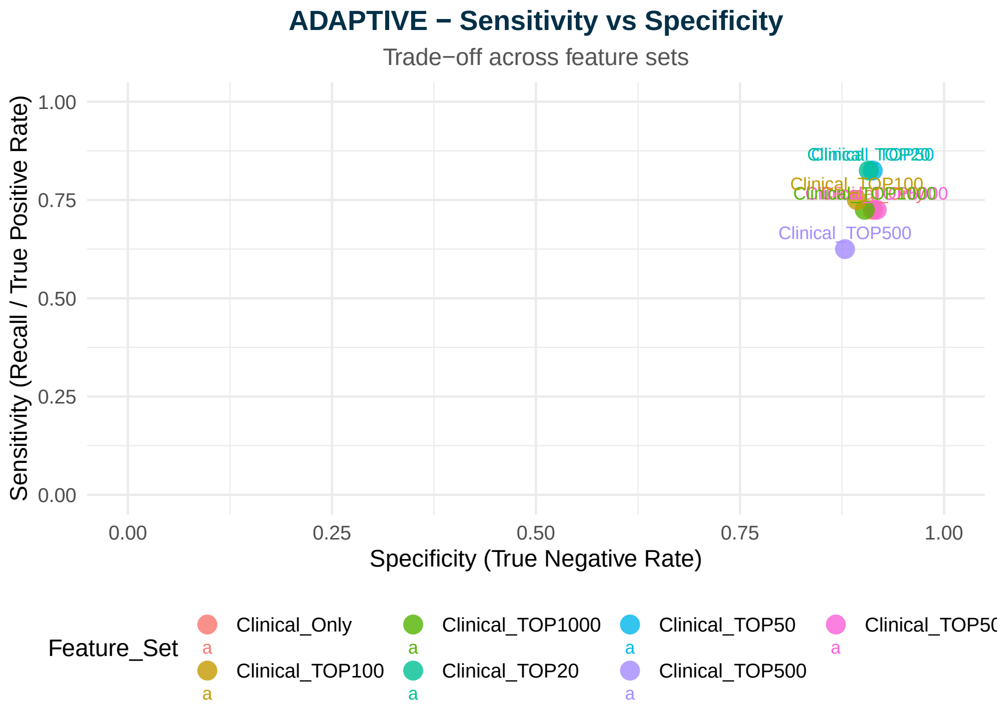

ADAPTIVE − Classification Metrics

Accuracy, Precision, Recall, F1−Score, AUC


ADAPTIVE − F1−Score Across Feature Sets

Trend of model performance





```
##
## === SUMMARY TABLE ===
##        Feature_Set TP  TN FP FN  Accuracy Precision Recall Specificity
## 1    Clinical_Only 29 188 18 11 0.8821138 0.6170213  0.725   0.9126214
## 2 Clinical_TOP5000 29 189 17 11 0.8861789 0.6304348  0.725   0.9174757
## 3 Clinical_TOP1000 29 186 20 11 0.8739837 0.5918367  0.725   0.9029126
## 4  Clinical_TOP500 25 181 25 15 0.8373984 0.5000000  0.625   0.8786408
## 5  Clinical_TOP100 30 184 22 10 0.8699187 0.5769231  0.750   0.8932039
## 6   Clinical_TOP50 33 188 18  7 0.8983740 0.6470588  0.825   0.9126214
## 7   Clinical_TOP20 33 187 19  7 0.8943089 0.6346154  0.825   0.9077670
##    F1_Score       AUC
## 1 0.6666667 0.9009709
## 2 0.6744186 0.8652913
## 3 0.6516854 0.8591019
## 4 0.5555556 0.8373786
## 5 0.6521739 0.8593447
## 6 0.7252747 0.8958738
## 7 0.7173913 0.9081311
##
## Exported classification metrics to: model_metrics/adaptive_lasso_smote_classification_metrics.csv
```

UniLasso with SMOTE

```r
unilasso_smote <- fit_single_model_across_features(
    model_type = "unilasso"
    , X_train_all = smote_data$X_train
    , X_test_all = X_test
    , Y_train = smote_data$Y_train
    , Y_test = Y_test
    , n_clinical = n_clinical
    , top_genes_ranked = top_genes
    , gene_sets = c(5000, 1000, 500, 100, 50, 20)
)
```

```
##
## === FITTING UNILASSO ACROSS FEATURE SETS ===
##
## Fitting Clinical_Only...
```

```
## Fitting Clinical_TOP5000...
```

```
## Fitting Clinical_TOP1000...
```

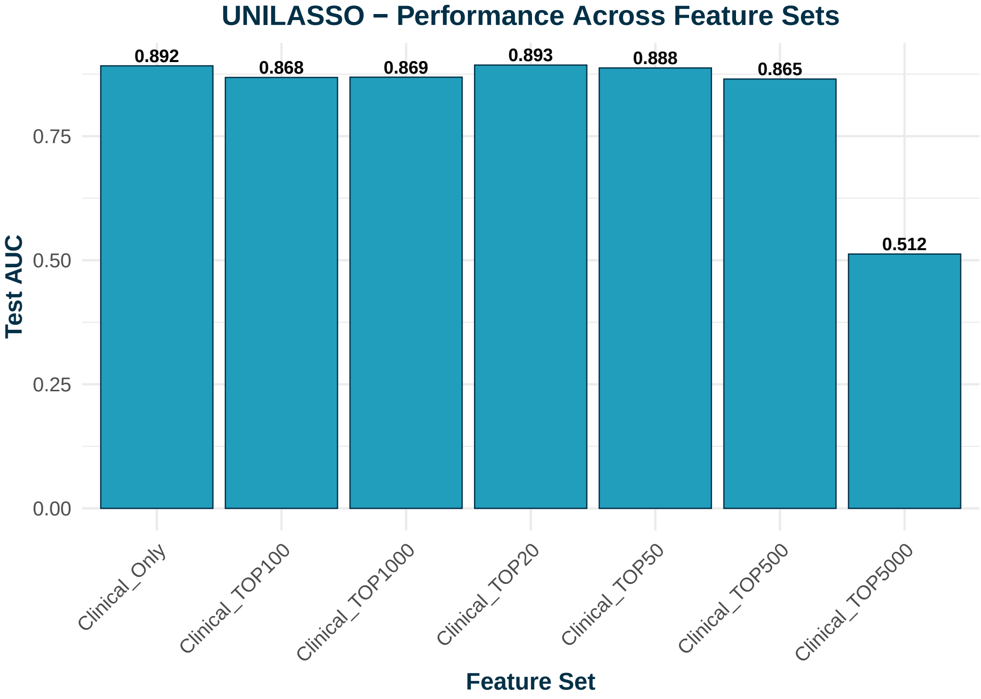
```
## Fitting Clinical_TOP500...
```

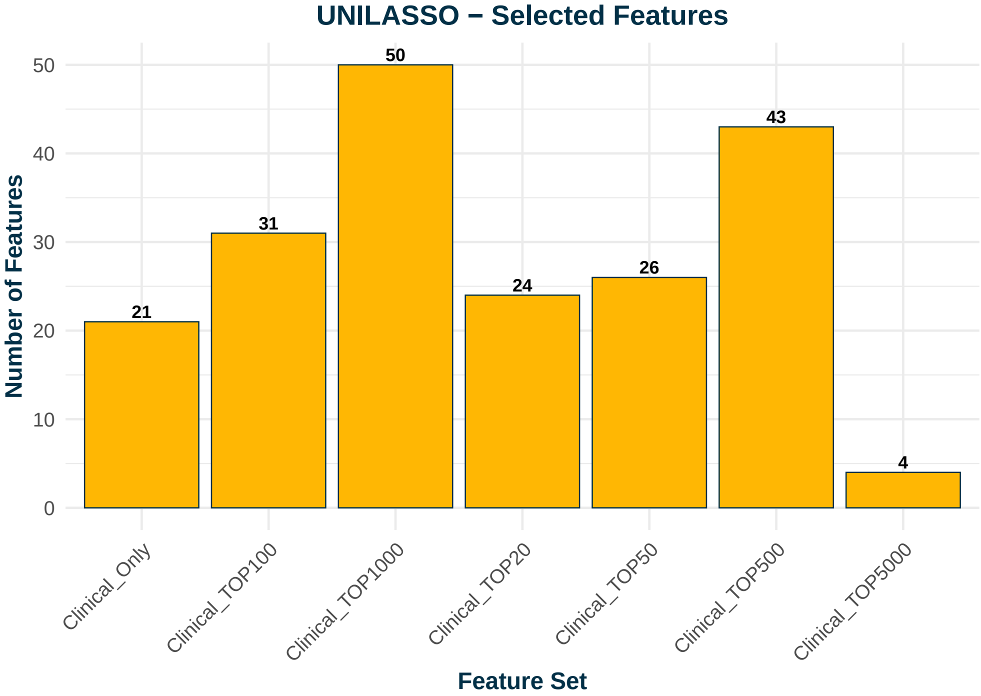
```
## Fitting Clinical_TOP100...
```

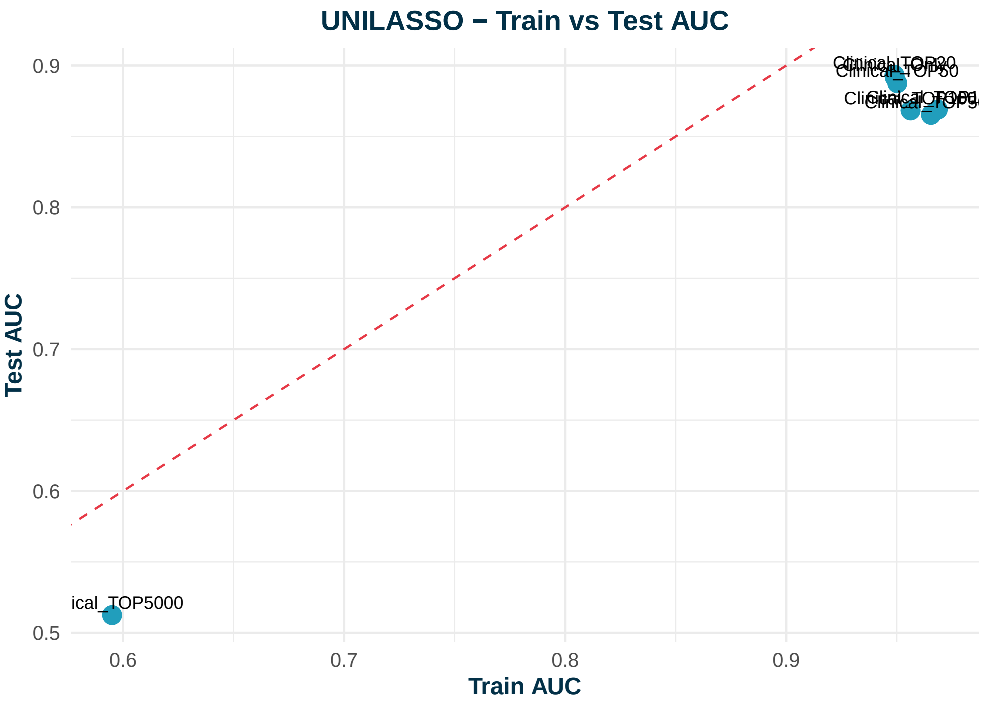
```
## Fitting Clinical_TOP50...
```

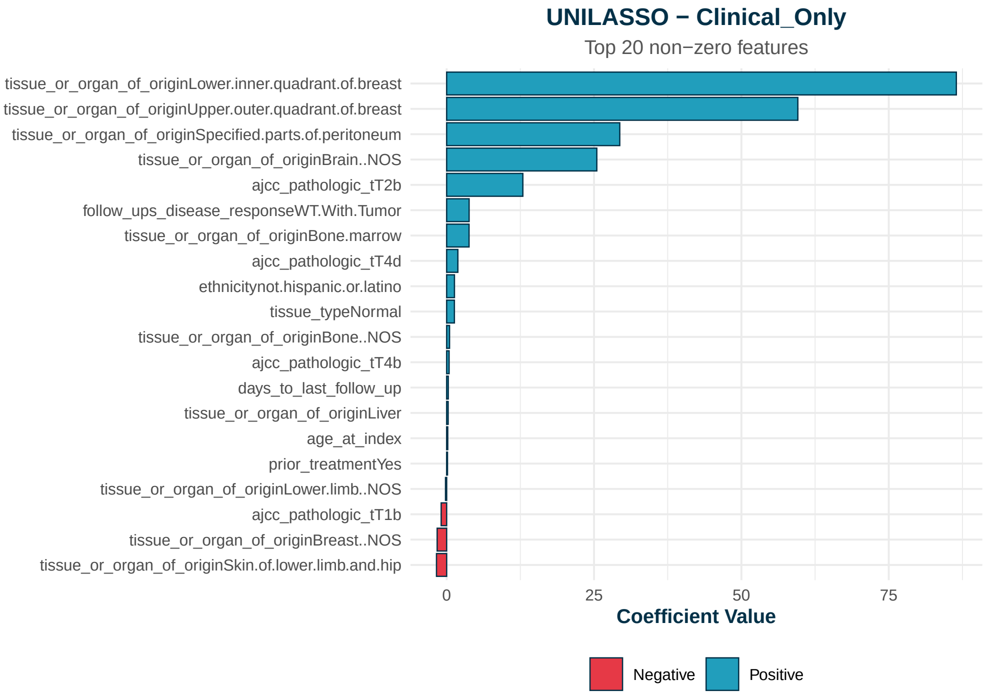
```
## Fitting Clinical_TOP20...
```

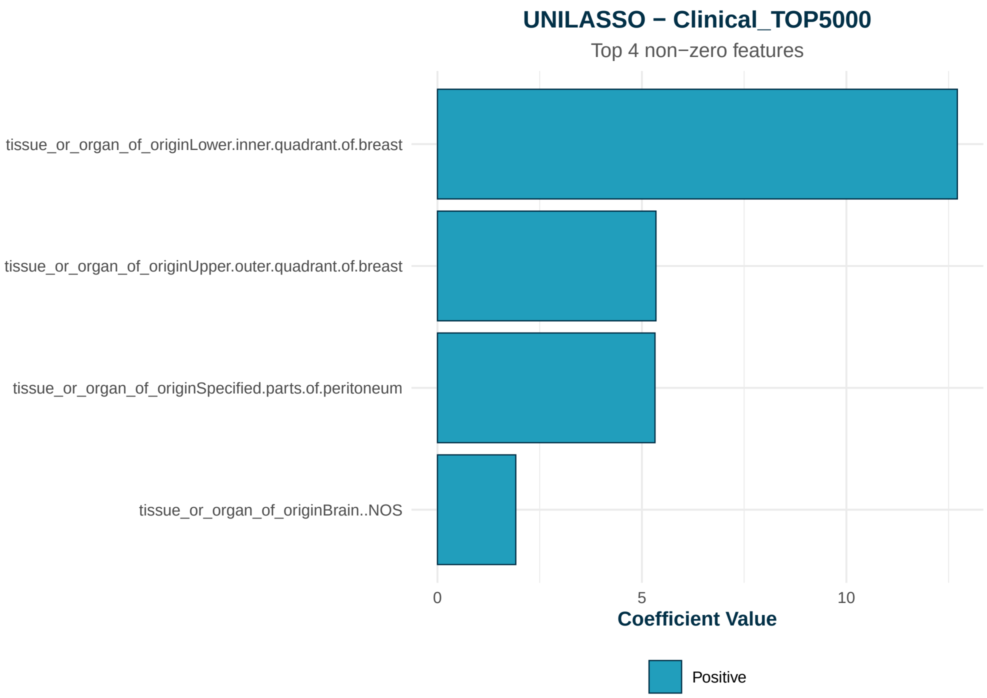
```
##
## === SUMMARY TABLE ===
##        Feature_Set    Model Features Train_AUC  Test_AUC Test_Accuracy
## 1    Clinical_Only UNILASSO       21 0.9490329 0.8917476     0.8821138
## 2 Clinical_TOP5000 UNILASSO        4 0.5950311 0.5125000     0.8414634
## 3 Clinical_TOP1000 UNILASSO       50 0.9685456 0.8689320     0.9065041
## 4  Clinical_TOP500 UNILASSO       43 0.9654393 0.8651699     0.9105691
## 5  Clinical_TOP100 UNILASSO       31 0.9562229 0.8683252     0.9065041
## 6   Clinical_TOP50 UNILASSO       26 0.9502049 0.8875000     0.9186992
## 7   Clinical_TOP20 UNILASSO       24 0.9489279 0.8933252     0.9186992
## Exported metrics to: model_metrics/unilasso_across_features_metrics.csv
```

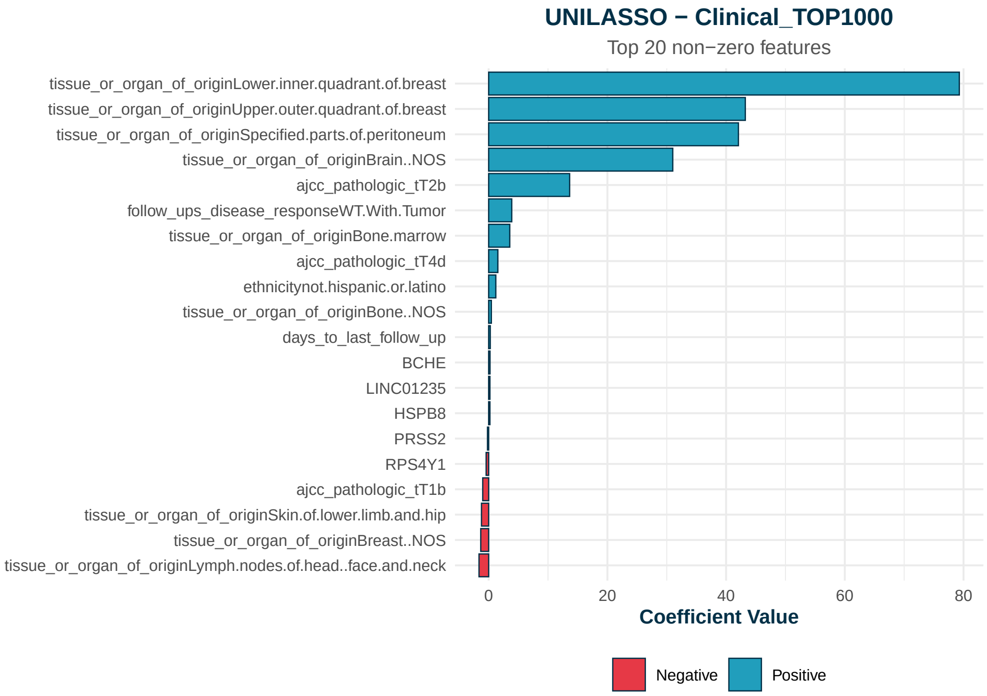


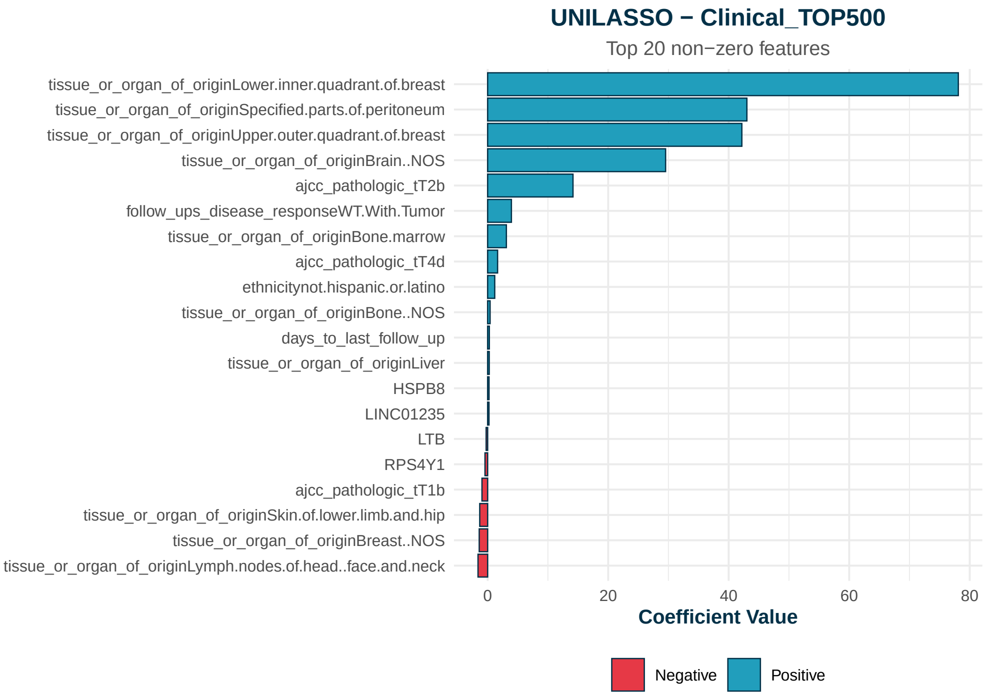


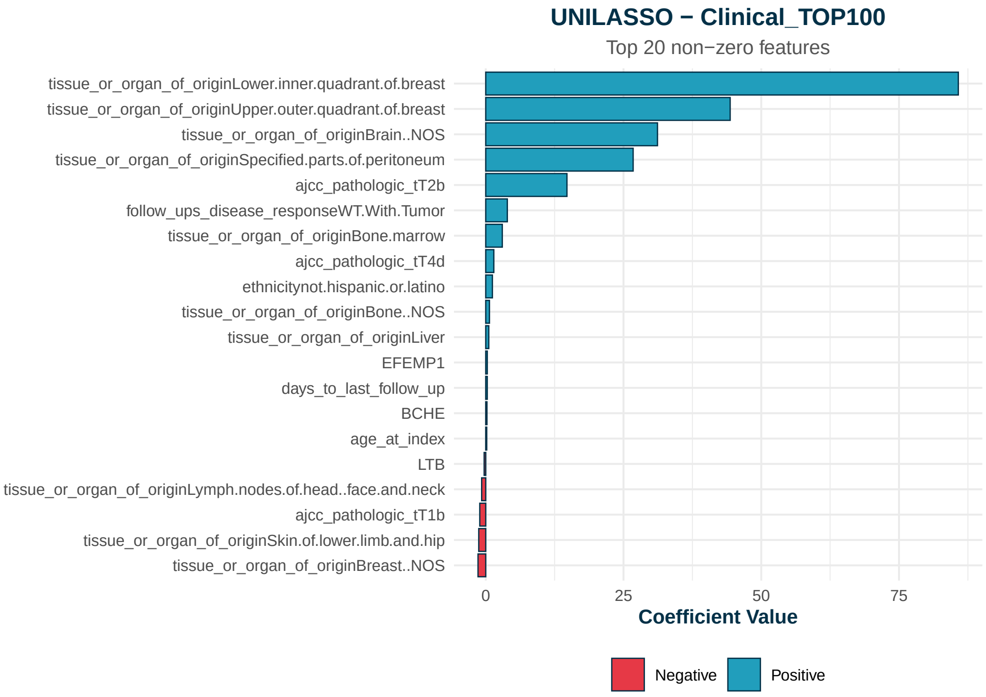

UNILASSO − Train vs Test AUC
Test AUC
Train AUC
0.5
0.6
0.7
0.8
0.9
0.6
0.7
0.8
0.9
Clinical_TOP20
Clinical_TOP50
ical_TOP5000

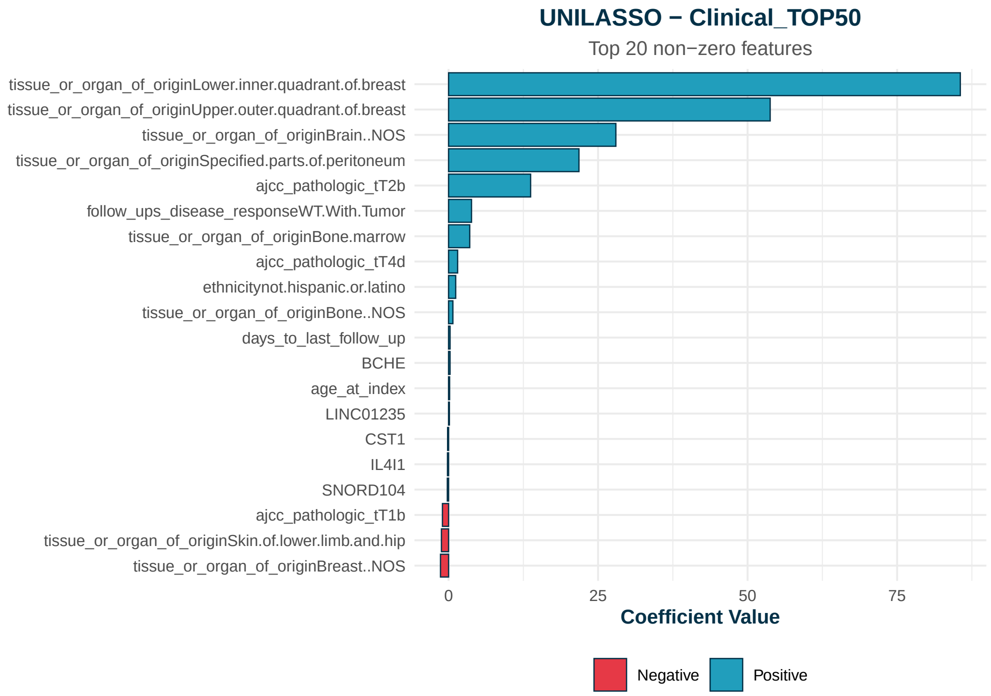

UNILASSO − Clinical_Only
Top 20 non−zero features
tissue_or_organ_of_originLower.inner.quadrant.of.breast
tissue_or_organ_of_originUpper.outer.quadrant.of.breast
tissue_or_organ_of_originSpecified.parts.of.peritoneum
tissue_or_organ_of_originBrain..NOS
ajcc_pathologic_tT2b
follow_ups_disease_responseWT.With.Tumor
tissue_or_organ_of_originBone.marrow
ajcc_pathologic_tT4d
ethnicitynot.hispanic.or.latino
tissue_typeNormal
tissue_or_organ_of_originBone..NOS
ajcc_pathologic_tT4b
days_to_last_follow_up
tissue_or_organ_of_originLiver
age_at_index
prior_treatmentYes
tissue_or_organ_of_originLower.limb..NOS
ajcc_pathologic_tT1b
tissue_or_organ_of_originBreast..NOS
tissue_or_organ_of_originSkin.of.lower.limb.and.hip
0
25
50
75
Coefficient Value
🟥 Negative
🟦 Positive

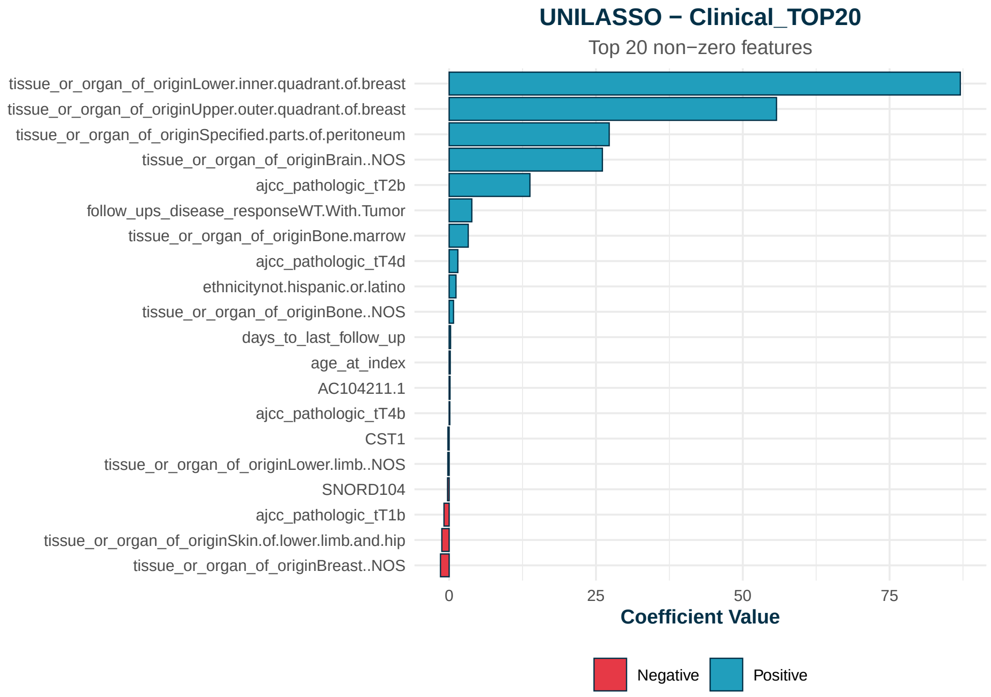

UNILASSO − Clinical_TOP5000
Top 4 non−zero features
tissue_or_organ_of_originLower.inner.quadrant.of.breast
tissue_or_organ_of_originUpper.outer.quadrant.of.breast
tissue_or_organ_of_originSpecified.parts.of.peritoneum
tissue_or_organ_of_originBrain..NOS
0
5
10
Coefficient Value
Positive


UNILASSO − Clinical_TOP1000
Top 20 non−zero features
tissue_or_organ_of_originLower.inner.quadrant.of.breast
tissue_or_organ_of_originUpper.outer.quadrant.of.breast
tissue_or_organ_of_originSpecified.parts.of.peritoneum
tissue_or_organ_of_originBrain..NOS
ajcc_pathologic_tT2b
follow_ups_disease_responseWT.With.Tumor
tissue_or_organ_of_originBone.marrow
ajcc_pathologic_tT4d
ethnicitynot.hispanic.or.latino
tissue_or_organ_of_originBone..NOS
days_to_last_follow_up
BCHE
LINC01235
HSPB8
PRSS2
RPS4Y1
ajcc_pathologic_tT1b
tissue_or_organ_of_originSkin.of.lower.limb.and.hip
tissue_or_organ_of_originBreast..NOS
tissue_or_organ_of_originLymph.nodes.of.head..face.and.neck
0
20
40
60
80
Coefficient Value
Negative
Positive


UNILASSO − Clinical_TOP500
Top 20 non−zero features
tissue_or_organ_of_originLower.inner.quadrant.of.breast
tissue_or_organ_of_originSpecified.parts.of.peritoneum
tissue_or_organ_of_originUpper.outer.quadrant.of.breast
tissue_or_organ_of_originBrain..NOS
ajcc_pathologic_tT2b
follow_ups_disease_responseWT.With.Tumor
tissue_or_organ_of_originBone.marrow
ajcc_pathologic_tT4d
ethnicitynot.hispanic.or.latino
tissue_or_organ_of_originBone..NOS
days_to_last_follow_up
tissue_or_organ_of_originLiver
HSPB8
LINC01235
LTB
RPS4Y1
ajcc_pathologic_tT1b
tissue_or_organ_of_originSkin.of.lower.limb.and.hip
tissue_or_organ_of_originBreast..NOS
tissue_or_organ_of_originLymph.nodes.of.head..face.and.neck
0
20
40
60
80
Coefficient Value
🟥 Negative
🟦 Positive

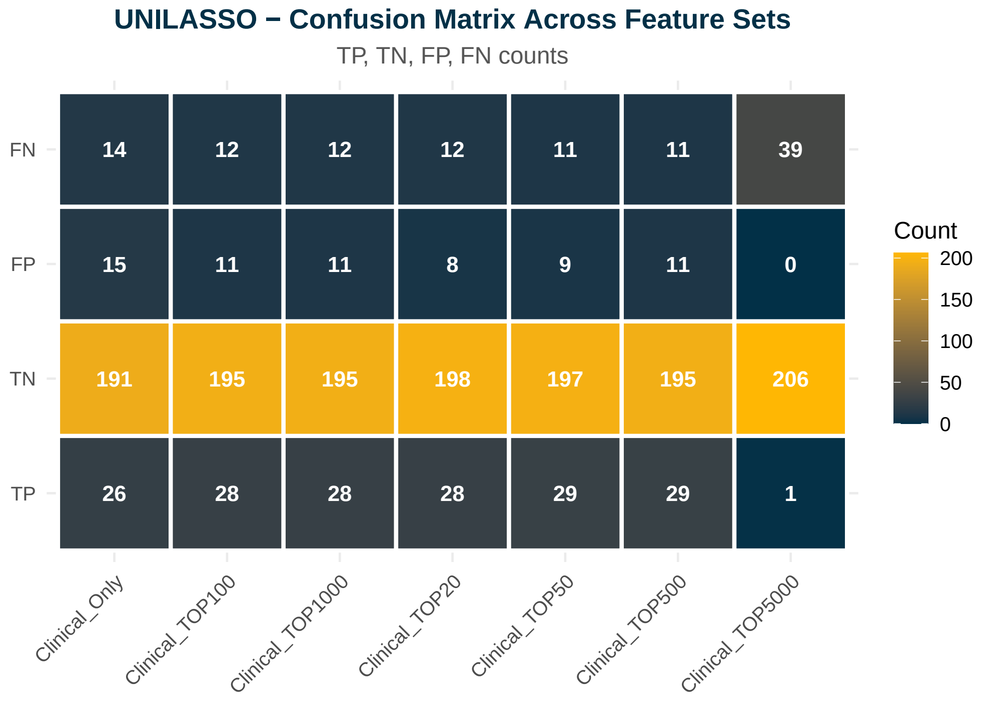

UNILASSO − Clinical_TOP100
Top 20 non−zero features
tissue_or_organ_of_originLower.inner.quadrant.of.breast
tissue_or_organ_of_originUpper.outer.quadrant.of.breast
tissue_or_organ_of_originBrain..NOS
tissue_or_organ_of_originSpecified.parts.of.peritoneum
ajcc_pathologic_tT2b
follow_ups_disease_responseWT.With.Tumor
tissue_or_organ_of_originBone.marrow
ajcc_pathologic_tT4d
ethnicitynot.hispanic.or.latino
tissue_or_organ_of_originBone..NOS
tissue_or_organ_of_originLiver
EFEMP1
days_to_last_follow_up
BCHE
age_at_index
LTB
tissue_or_organ_of_originLymph.nodes.of.head..face.and.neck
ajcc_pathologic_tT1b
tissue_or_organ_of_originSkin.of.lower.limb.and.hip
tissue_or_organ_of_originBreast..NOS
0
25
50
75
Coefficient Value
🟥 Negative
🟦 Positive

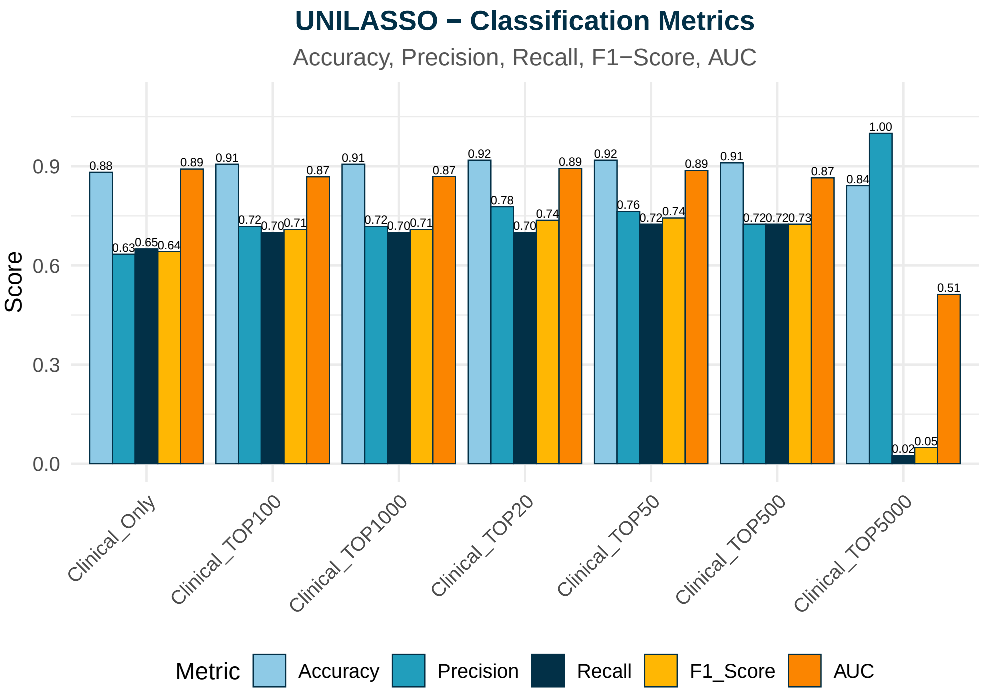


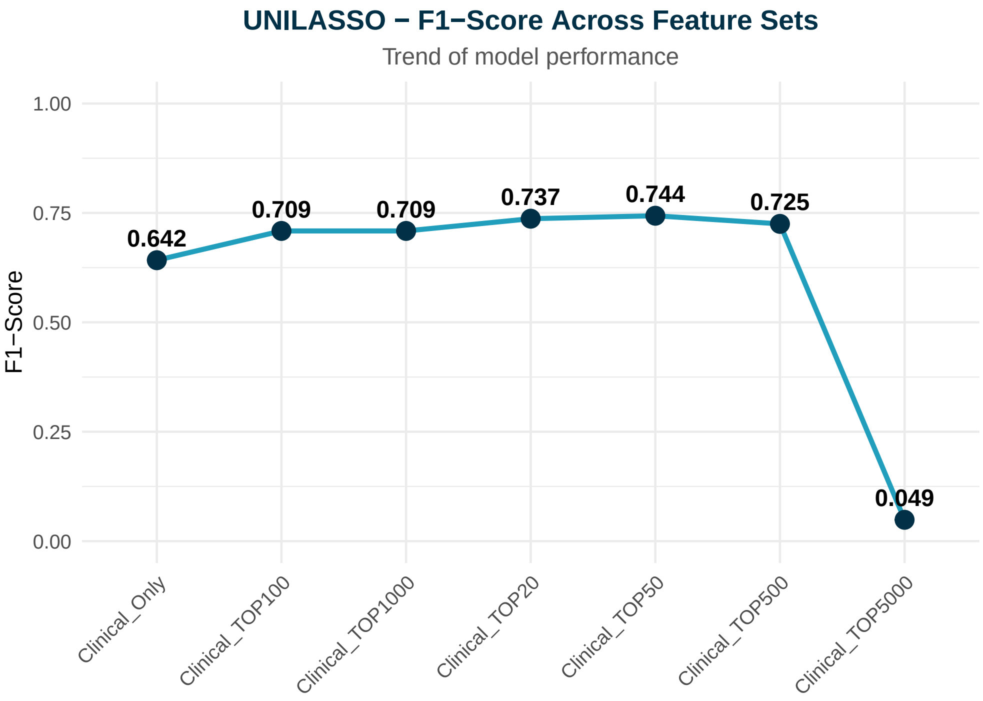


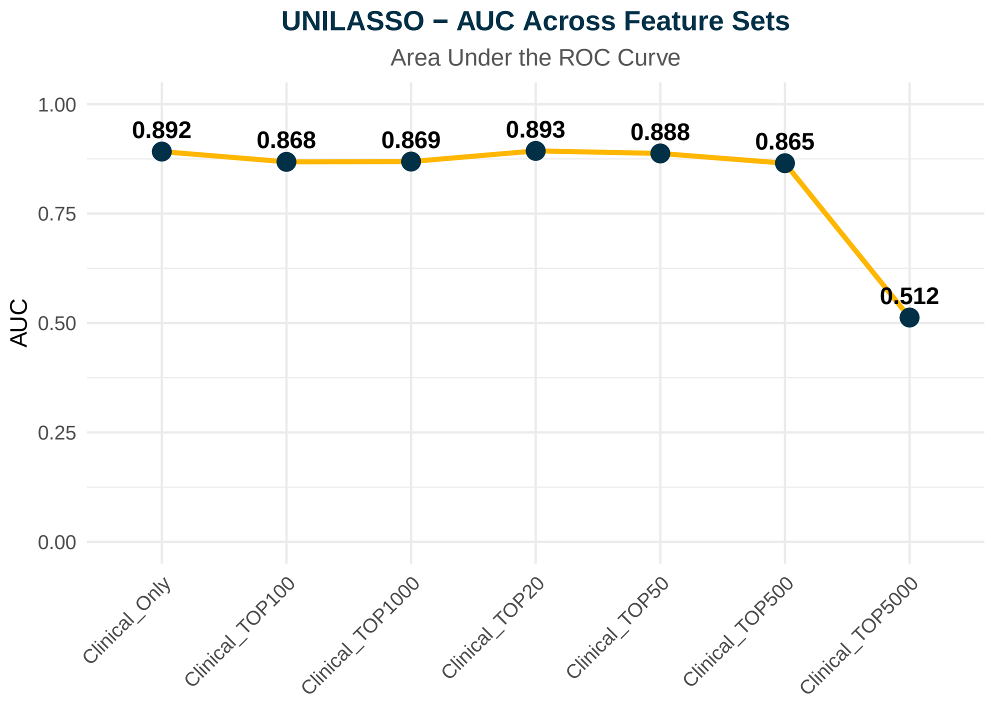
```
unilasso_smote_metrics <- plot_classification_metrics_single(unilasso_smote
                                                              , threshold = 0.5
                                                              , csv_filename = "unilasso_smote_classifica
```

```
##
## === CLASSIFICATION METRICS ===
## Clinical_Only:
##   TP=26 TN=191 FP=15 FN=14
##   Accuracy=0.882 Precision=0.634 Recall=0.650 F1=0.642 AUC=0.892

## Clinical_TOP5000:
##   TP=1 TN=206 FP=0 FN=39
##   Accuracy=0.841 Precision=1.000 Recall=0.025 F1=0.049 AUC=0.512

## Clinical_TOP1000:
##   TP=28 TN=195 FP=11 FN=12
##   Accuracy=0.907 Precision=0.718 Recall=0.700 F1=0.709 AUC=0.869

## Clinical_TOP500:
##   TP=29 TN=195 FP=11 FN=11
##   Accuracy=0.911 Precision=0.725 Recall=0.725 F1=0.725 AUC=0.865

## Clinical_TOP100:
##   TP=28 TN=195 FP=11 FN=12
##   Accuracy=0.907 Precision=0.718 Recall=0.700 F1=0.709 AUC=0.868

## Clinical_TOP50:
##   TP=29 TN=197 FP=9 FN=11
##   Accuracy=0.919 Precision=0.763 Recall=0.725 F1=0.744 AUC=0.888

## Clinical_TOP20:
##   TP=28 TN=198 FP=8 FN=12
##   Accuracy=0.919 Precision=0.778 Recall=0.700 F1=0.737 AUC=0.893
```

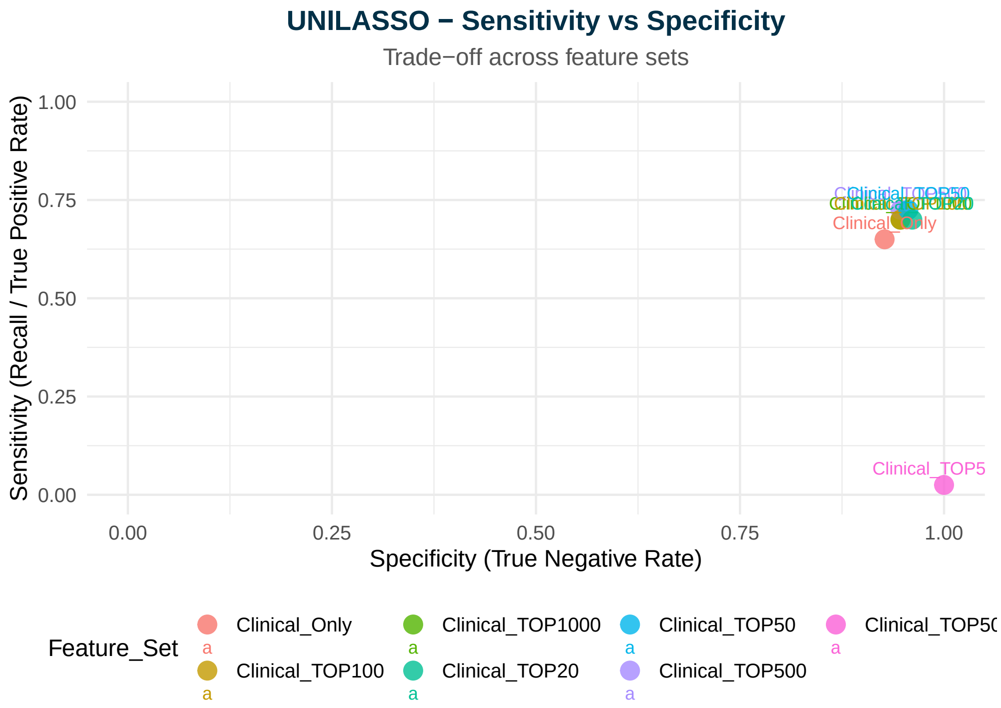



UNILASSO − Classification Metrics
Accuracy, Precision, Recall, F1−Score, AUC
Score
0.0
0.3
0.6
0.9
0.88
0.63
0.65
0.64
0.89
0.91
0.72
0.70
0.71
0.87
0.91
0.72
0.70
0.71
0.87
0.92
0.78
0.70
0.74
0.89
0.92
0.76
0.72
0.74
0.89
0.91
0.72
0.72
0.73
0.87
0.84
1.00
0.02
0.05
0.51
Clinical_Only
Clinical_TOP100
Clinical_TOP1000
Clinical_TOP20
Clinical_TOP50
Clinical_TOP500
Clinical_TOP5000
Metric
Accuracy
Precision
Recall
F1_Score
AUC


UNILASSO − F1−Score Across Feature Sets
Trend of model performance
F1−Score
0.00
0.25
0.50
0.75
1.00
0.642
0.709
0.709
0.737
0.744
0.725
0.049
Clinical_Only
Clinical_TOP100
Clinical_TOP1000
Clinical_TOP20
Clinical_TOP50
Clinical_TOP500
Clinical_TOP5000


UNILASSO − AUC Across Feature Sets

Area Under the ROC Curve


UNILASSO − Sensitivity vs Specificity

Trade−off across feature sets

```
##
## === SUMMARY TABLE ===
##        Feature_Set TP  TN FP FN  Accuracy Precision Recall Specificity
## 1     Clinical_Only 26 191 15 14 0.8821138 0.6341463  0.650   0.9271845
## 2 Clinical_TOP5000  1 206  0 39 0.8414634 1.0000000  0.025   1.0000000
## 3 Clinical_TOP1000 28 195 11 12 0.9065041 0.7179487  0.700   0.9466019
## 4  Clinical_TOP500 29 195 11 11 0.9105691 0.7250000  0.725   0.9466019
## 5  Clinical_TOP100 28 195 11 12 0.9065041 0.7179487  0.700   0.9466019
## 6   Clinical_TOP50 29 197  9 11 0.9186992 0.7631579  0.725   0.9563107
## 7   Clinical_TOP20 28 198  8 12 0.9186992 0.7777778  0.700   0.9611650
##     F1_Score       AUC
## 1 0.64197531 0.8917476
## 2 0.04878049 0.5125000
## 3 0.70886076 0.8689320
## 4 0.72500000 0.8651699
## 5 0.70886076 0.8683252
## 6 0.74358974 0.8875000
## 7 0.73684211 0.8933252
##
## Exported classification metrics to: model_metrics/unilasso_smote_classification_metrics.csv
```

SMOTE Impact Comparison

```r
cat("\n=== RIDGE: SMOTE vs NO SMOTE COMPARISON ===\n\n")
```

Ridge Comparison

```
##
## === RIDGE: SMOTE vs NO SMOTE COMPARISON ===
```

```r
cat("Before SMOTE (Clinical_Only):\n")
```

```
## Before SMOTE (Clinical_Only):
```

```r
cat("  Recall:", sprintf("%.3f", ridge_metrics$Recall[1]), "\n")
```

```
##   Recall: 0.175
```

```r
cat("  Precision:", sprintf("%.3f", ridge_metrics$Precision[1]), "\n")
```

```
##   Precision: 0.778
```

```r
cat("  F1-Score:", sprintf("%.3f", ridge_metrics$F1_Score[1]), "\n")
```

```
##   F1-Score: 0.286
```

```r
cat("  AUC:", sprintf("%.3f", ridge_metrics$AUC[1]), "\n\n")
```

```
##   AUC: 0.883
```

```r
cat("After SMOTE (Clinical_Only):\n")
```

```
## After SMOTE (Clinical_Only):
```

```r
cat("  Recall:", sprintf("%.3f", ridge_smote_metrics$Recall[1]), "\n")
```

```
##   Recall: 0.725
```

```r
cat("  Precision:", sprintf("%.3f", ridge_smote_metrics$Precision[1]), "\n")
```

```
##   Precision: 0.547
```

```r
cat("  F1-Score:", sprintf("%.3f", ridge_smote_metrics$F1_Score[1]), "\n")
```

```
##   F1-Score: 0.624
```

```r
cat("  AUC:", sprintf("%.3f", ridge_smote_metrics$AUC[1]), "\n\n")
```

```
##   AUC: 0.884
```

```r
cat("Improvement:\n")
```

```
## Improvement:
```

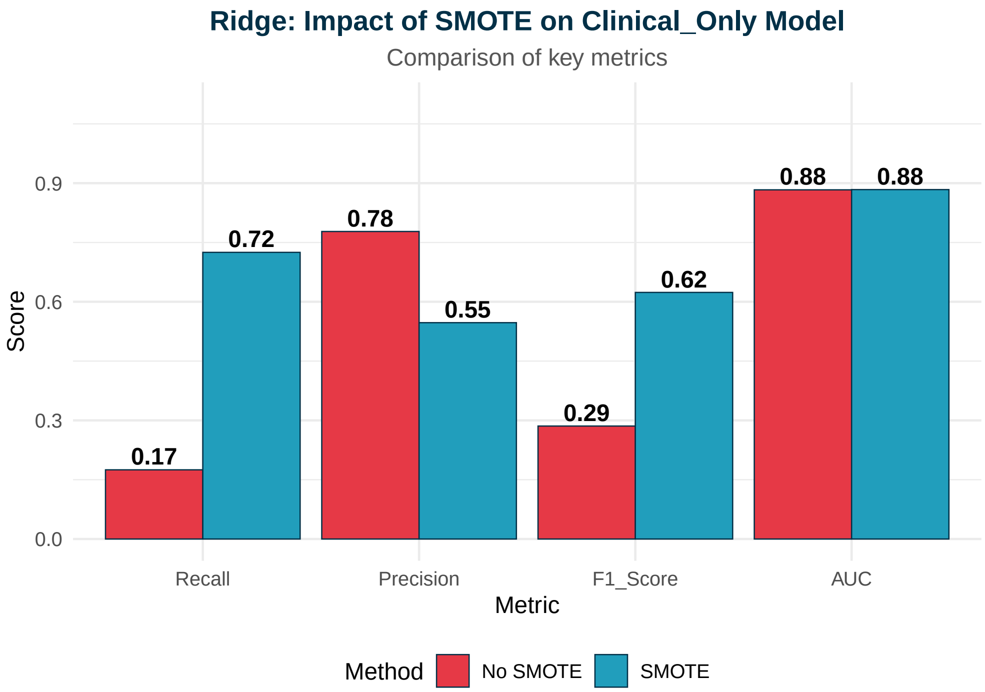
```r
cat("  Recall:", sprintf("%+.3f", ridge_smote_metrics$Recall[1] - ridge_metrics$Recall[1]), "\n")
```

```
##   Recall: +0.550
```

```r
cat("  F1-Score:", sprintf("%+.3f", ridge_smote_metrics$F1_Score[1] - ridge_metrics$F1_Score[1]), "\n")
```

```
##   F1-Score: +0.338
```

```r
comparison_ridge <- rbind(
    data.frame(Method = "No SMOTE", ridge_metrics[1, c("Feature_Set", "Recall", "Precision", "F1_Score"
    data.frame(Method = "SMOTE", ridge_smote_metrics[1, c("Feature_Set", "Recall", "Precision", "F1_Sco
)

comp_ridge_long <- reshape2::melt(comparison_ridge, id.vars = c("Method", "Feature_Set"))

ggplot(comp_ridge_long, aes(x = variable, y = value, fill = Method)) +
    geom_bar(stat = "identity", position = "dodge", color = "#023047", linewidth = 0.3) +
    geom_text(aes(label = sprintf("%.2f", value))
              , position = position_dodge(width = 0.9)
              , vjust = -0.3
              , fontface = "bold") +
    labs(title = "Ridge: Impact of SMOTE on Clinical_Only Model"
         , subtitle = "Comparison of key metrics"
         , x = "Metric"
         , y = "Score") +
    theme_minimal(base_size = 12) +
    theme(plot.title = element_text(face = "bold", hjust = 0.5, color = "#023047")
          , plot.subtitle = element_text(hjust = 0.5, color = "#555555")
          , legend.position = "bottom") +
    scale_fill_manual(values = c("No SMOTE" = "#e63946", "SMOTE" = "#219ebc")) +
    ylim(0, 1.1)
```



```
cat("\n=== LASSO: SMOTE vs NO SMOTE COMPARISON ===\n\n")
```

Lasso Comparison

```
##
## === LASSO: SMOTE vs NO SMOTE COMPARISON ===
```

```
cat("Before SMOTE (Clinical_Only):\n")
```

```
## Before SMOTE (Clinical_Only):
```

```
cat("  Recall:", sprintf("%.3f", lasso_metrics$Recall[1]), "\n")
```

```
##   Recall: 0.400
```

```
cat("  Precision:", sprintf("%.3f", lasso_metrics$Precision[1]), "\n")
```

```
##   Precision: 0.762
```

```
cat("  F1-Score:", sprintf("%.3f", lasso_metrics$F1_Score[1]), "\n")
```

```
##   F1-Score: 0.525
```

```
cat("  AUC:", sprintf("%.3f", lasso_metrics$AUC[1]), "\n\n")
```

```
##   AUC: 0.886
```

```
cat("After SMOTE (Clinical_Only):\n")
```

```
## After SMOTE (Clinical_Only):
```

```
cat("  Recall:", sprintf("%.3f", lasso_smote_metrics$Recall[1]), "\n")
```

```
##   Recall: 0.725
```

```
cat("  Precision:", sprintf("%.3f", lasso_smote_metrics$Precision[1]), "\n")
```

```
##   Precision: 0.592
```

```
cat("  F1-Score:", sprintf("%.3f", lasso_smote_metrics$F1_Score[1]), "\n")
```

```
##   F1-Score: 0.652
```

```
cat("  AUC:", sprintf("%.3f", lasso_smote_metrics$AUC[1]), "\n\n")
```

```
##   AUC: 0.898
```

```
cat("Improvement:\n")
```

```
## Improvement:
```

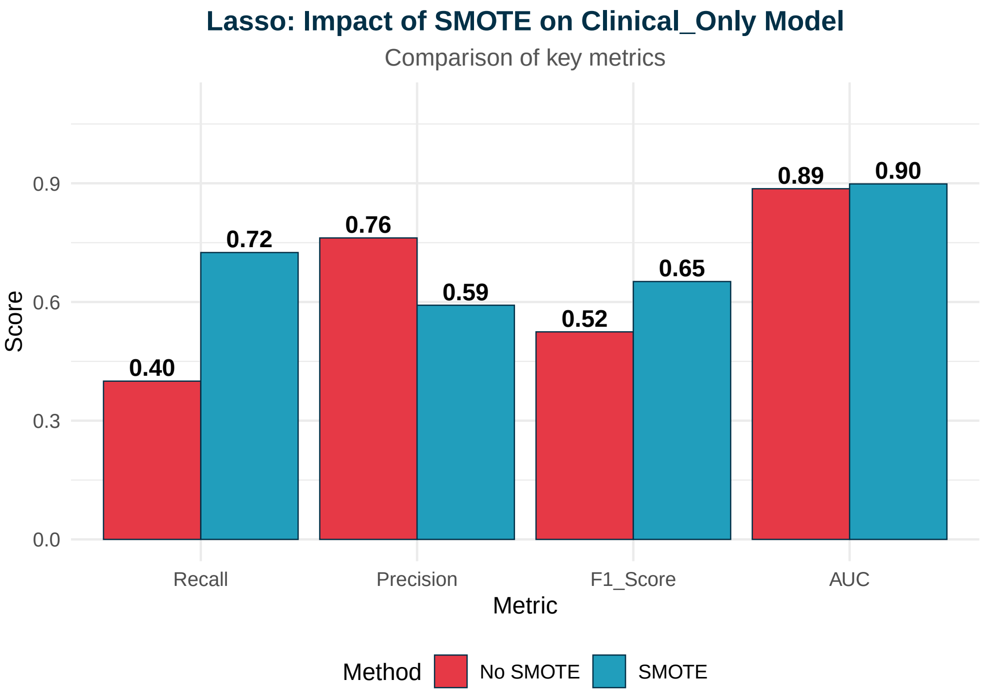
```
cat("  Recall:", sprintf("%+.3f", lasso_smote_metrics$Recall[1] - lasso_metrics$Recall[1]), "\n")
```

```
##   Recall: +0.325
```

```
cat("  F1-Score:", sprintf("%+.3f", lasso_smote_metrics$F1_Score[1] - lasso_metrics$F1_Score[1]), "\n")
```

```
##   F1-Score: +0.127
```

```
comparison_lasso <- rbind(
    data.frame(Method = "No SMOTE", lasso_metrics[1, c("Feature_Set", "Recall", "Precision", "F1_Score"
    data.frame(Method = "SMOTE", lasso_smote_metrics[1, c("Feature_Set", "Recall", "Precision", "F1_Sco
)

comp_lasso_long <- reshape2::melt(comparison_lasso, id.vars = c("Method", "Feature_Set"))

ggplot(comp_lasso_long, aes(x = variable, y = value, fill = Method)) +
    geom_bar(stat = "identity", position = "dodge", color = "#023047", linewidth = 0.3) +
    geom_text(aes(label = sprintf("%.2f", value))
              , position = position_dodge(width = 0.9)
              , vjust = -0.3
              , fontface = "bold") +
    labs(title = "Lasso: Impact of SMOTE on Clinical_Only Model"
         , subtitle = "Comparison of key metrics"
         , x = "Metric"
         , y = "Score") +
    theme_minimal(base_size = 12) +
    theme(plot.title = element_text(face = "bold", hjust = 0.5, color = "#023047")
          , plot.subtitle = element_text(hjust = 0.5, color = "#555555")
          , legend.position = "bottom") +
    scale_fill_manual(values = c("No SMOTE" = "#e63946", "SMOTE" = "#219ebc")) +
    ylim(0, 1.1)
```



```
cat("\n=== ELASTICNET: SMOTE vs NO SMOTE COMPARISON ===\n\n")
```

ElasticNet Comparison

```
##
## === ELASTICNET: SMOTE vs NO SMOTE COMPARISON ===
```

```
cat("Before SMOTE (Clinical_Only):\n")
```

```
## Before SMOTE (Clinical_Only):
```

```r
cat("  Recall:", sprintf("%.3f", elasticnet_metrics$Recall[1]), "\n")
```

```
##   Recall: 0.250
```

```r
cat("  Precision:", sprintf("%.3f", elasticnet_metrics$Precision[1]), "\n")
```

```
##   Precision: 0.769
```

```r
cat("  F1-Score:", sprintf("%.3f", elasticnet_metrics$F1_Score[1]), "\n")
```

```
##   F1-Score: 0.377
```

```r
cat("  AUC:", sprintf("%.3f", elasticnet_metrics$AUC[1]), "\n\n")
```

```
##   AUC: 0.885
```

```r
cat("After SMOTE (Clinical_Only):\n")
```

```
## After SMOTE (Clinical_Only):
```

```r
cat("  Recall:", sprintf("%.3f", elasticnet_smote_metrics$Recall[1]), "\n")
```

```
##   Recall: 0.725
```

```r
cat("  Precision:", sprintf("%.3f", elasticnet_smote_metrics$Precision[1]), "\n")
```

```
##   Precision: 0.580
```

```r
cat("  F1-Score:", sprintf("%.3f", elasticnet_smote_metrics$F1_Score[1]), "\n")
```

```
##   F1-Score: 0.644
```

```r
cat("  AUC:", sprintf("%.3f", elasticnet_smote_metrics$AUC[1]), "\n\n")
```

```
##   AUC: 0.899
```

```r
cat("Improvement:\n")
```

```
## Improvement:
```

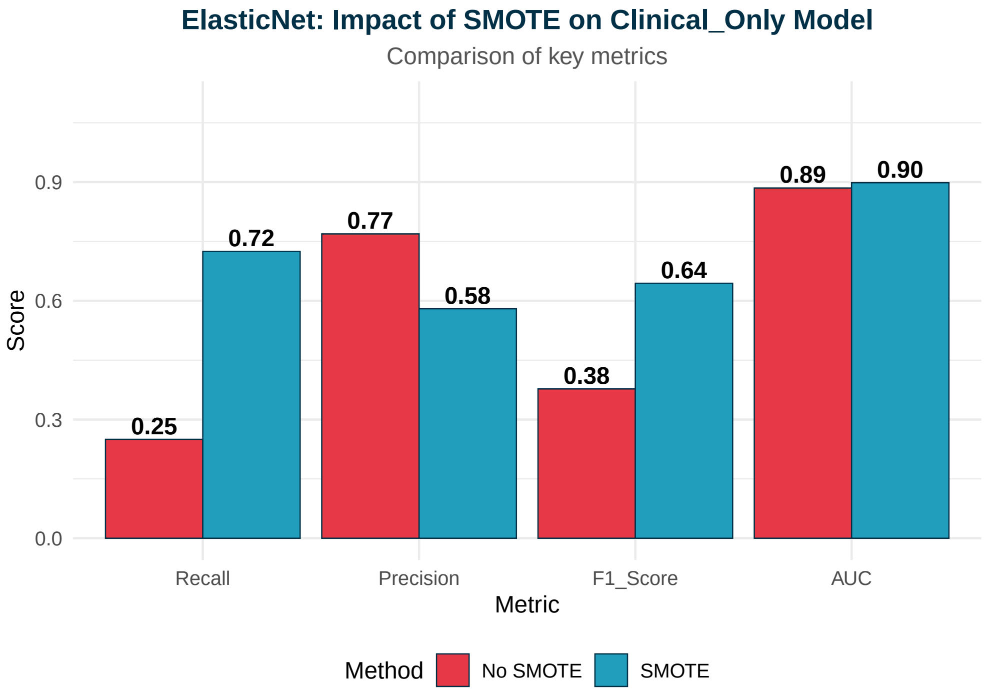
```r
cat("  Recall:", sprintf("%+.3f", elasticnet_smote_metrics$Recall[1] - elasticnet_metrics$Recall[1]), "
```

```
##   Recall: +0.475
```

```r
cat("  F1-Score:", sprintf("%+.3f", elasticnet_smote_metrics$F1_Score[1] - elasticnet_metrics$F1_Score[
```

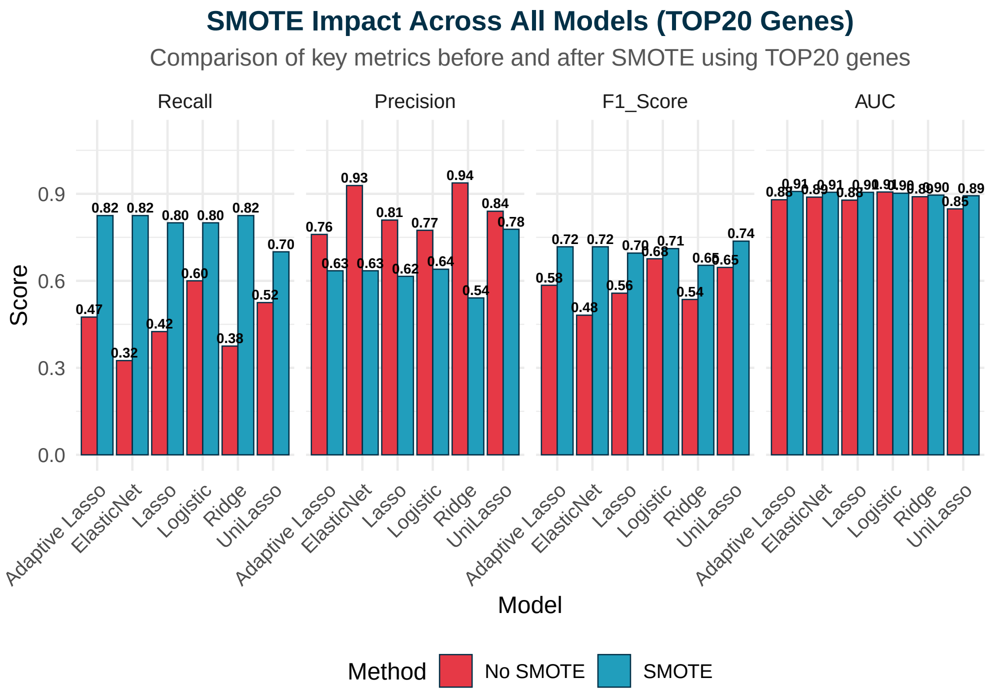
```
##   F1-Score: +0.267
```

```r
comparison_elasticnet <- rbind(
    data.frame(Method = "No SMOTE", elasticnet_metrics[1, c("Feature_Set", "Recall", "Precision", "F1_S
    data.frame(Method = "SMOTE", elasticnet_smote_metrics[1, c("Feature_Set", "Recall", "Precision", "F
)

comp_elasticnet_long <- reshape2::melt(comparison_elasticnet, id.vars = c("Method", "Feature_Set"))

ggplot(comp_elasticnet_long, aes(x = variable, y = value, fill = Method)) +
    geom_bar(stat = "identity", position = "dodge", color = "#023047", linewidth = 0.3) +
    geom_text(aes(label = sprintf("%.2f", value))
              , position = position_dodge(width = 0.9)
              , vjust = -0.3
              , fontface = "bold") +
    labs(title = "ElasticNet: Impact of SMOTE on Clinical_Only Model"
         , subtitle = "Comparison of key metrics"
         , x = "Metric"
         , y = "Score") +
    theme_minimal(base_size = 12) +
    theme(plot.title = element_text(face = "bold", hjust = 0.5, color = "#023047")
          , plot.subtitle = element_text(hjust = 0.5, color = "#555555")
          , legend.position = "bottom") +
    scale_fill_manual(values = c("No SMOTE" = "#e63946", "SMOTE" = "#219ebc")) +
    ylim(0, 1.1)
```

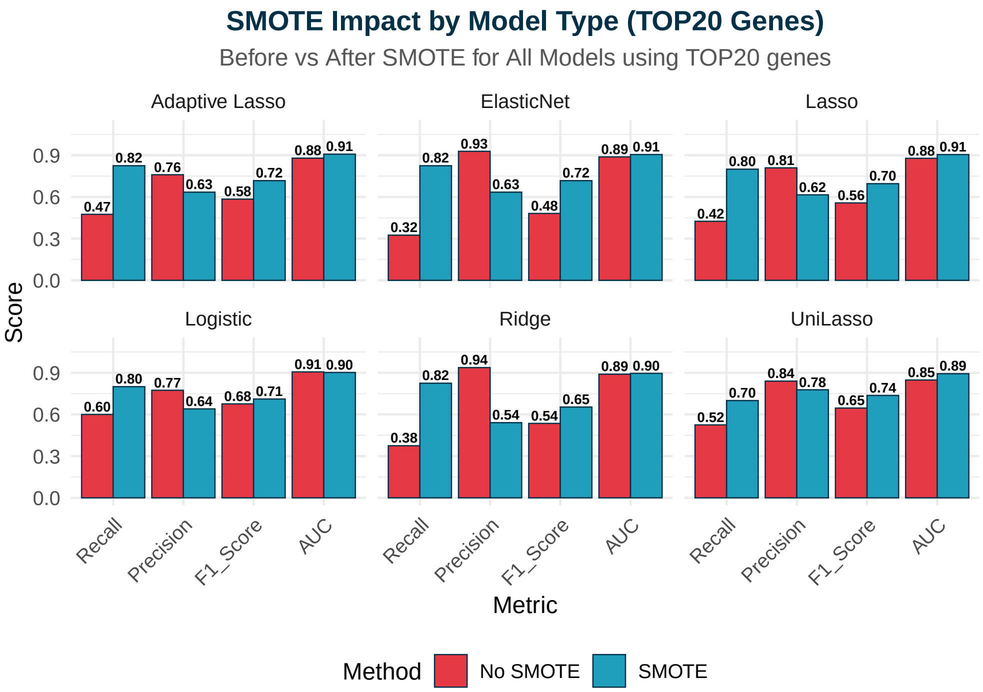


Overall SMOTE Impact Across All Models

```r
# Combine all comparisons for TOP20 genes
all_comparisons <- rbind(
    data.frame(Model = "Logistic", Method = "No SMOTE", logistic_metrics[4, c("Recall", "Precision", "F
    data.frame(Model = "Logistic", Method = "SMOTE", logistic_smote_metrics[4, c("Recall", "Precision",
    data.frame(Model = "Ridge", Method = "No SMOTE", ridge_metrics[7, c("Recall", "Precision", "F1_Scor
    data.frame(Model = "Ridge", Method = "SMOTE", ridge_smote_metrics[7, c("Recall", "Precision", "F1_S
    data.frame(Model = "Lasso", Method = "No SMOTE", lasso_metrics[7, c("Recall", "Precision", "F1_Scor
    data.frame(Model = "Lasso", Method = "SMOTE", lasso_smote_metrics[7, c("Recall", "Precision", "F1_S
    data.frame(Model = "ElasticNet", Method = "No SMOTE", elasticnet_metrics[7, c("Recall", "Precision"
    data.frame(Model = "ElasticNet", Method = "SMOTE", elasticnet_smote_metrics[7, c("Recall", "Precisi
    data.frame(Model = "Adaptive Lasso", Method = "No SMOTE", adaptive_lasso_metrics[7, c("Recall", "Pr
    data.frame(Model = "Adaptive Lasso", Method = "SMOTE", adaptive_lasso_smote_metrics[7, c("Recall",
    data.frame(Model = "UniLasso", Method = "No SMOTE", unilasso_metrics[7, c("Recall", "Precision", "F
    data.frame(Model = "UniLasso", Method = "SMOTE", unilasso_smote_metrics[7, c("Recall", "Precision",
)

all_comp_long <- reshape2::melt(all_comparisons, id.vars = c("Model", "Method"))

# Figure 1: Grouped by Metric
ggplot(all_comp_long, aes(x = Model, y = value, fill = Method)) +
    geom_bar(stat = "identity", position = "dodge", color = "#023047", linewidth = 0.3) +
    geom_text(aes(label = sprintf("%.2f", value))
              , position = position_dodge(width = 0.9)
              , vjust = -0.3
              , size = 2.5
              , fontface = "bold") +
    facet_wrap(~ variable, ncol = 4) +
    labs(title = "SMOTE Impact Across All Models (TOP20 Genes)"
         , subtitle = "Comparison of key metrics before and after SMOTE using TOP20 genes"
         , x = "Model"
         , y = "Score"
         , fill = "Method") +
    theme_minimal(base_size = 12) +
    theme(plot.title = element_text(face = "bold", hjust = 0.5, color = "#023047")
          , plot.subtitle = element_text(hjust = 0.5, color = "#555555")
          , axis.text.x = element_text(angle = 45, hjust = 1)
          , legend.position = "bottom") +
    scale_fill_manual(values = c("No SMOTE" = "#e63946", "SMOTE" = "#219ebc")) +
    ylim(0, 1.1)
```



```r
# Figure 2: Grouped by Model
ggplot(all_comp_long, aes(x = variable, y = value, fill = Method)) +
    geom_bar(stat = "identity", position = "dodge", color = "#023047", linewidth = 0.3) +
    geom_text(aes(label = sprintf("%.2f", value))
              , position = position_dodge(width = 0.9)
              , vjust = -0.3
              , size = 2.5
              , fontface = "bold") +
    facet_wrap(~ Model, ncol = 3) +
    labs(title = "SMOTE Impact by Model Type (TOP20 Genes)"
         , subtitle = "Before vs After SMOTE for All Models using TOP20 genes"
         , x = "Metric"
         , y = "Score"
         , fill = "Method") +
    theme_minimal(base_size = 12) +
    theme(plot.title = element_text(face = "bold", hjust = 0.5, color = "#023047")
          , plot.subtitle = element_text(hjust = 0.5, color = "#555555")
          , axis.text.x = element_text(angle = 45, hjust = 1)
          , legend.position = "bottom") +
    scale_fill_manual(values = c("No SMOTE" = "#e63946", "SMOTE" = "#219ebc")) +
    ylim(0, 1.1)
```



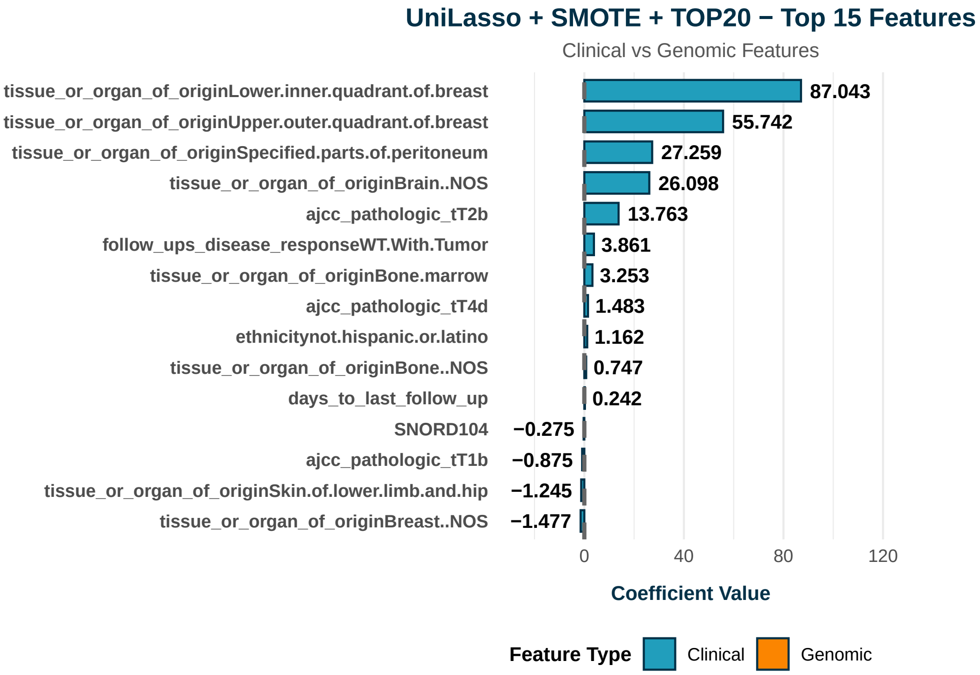
Summary on Smote

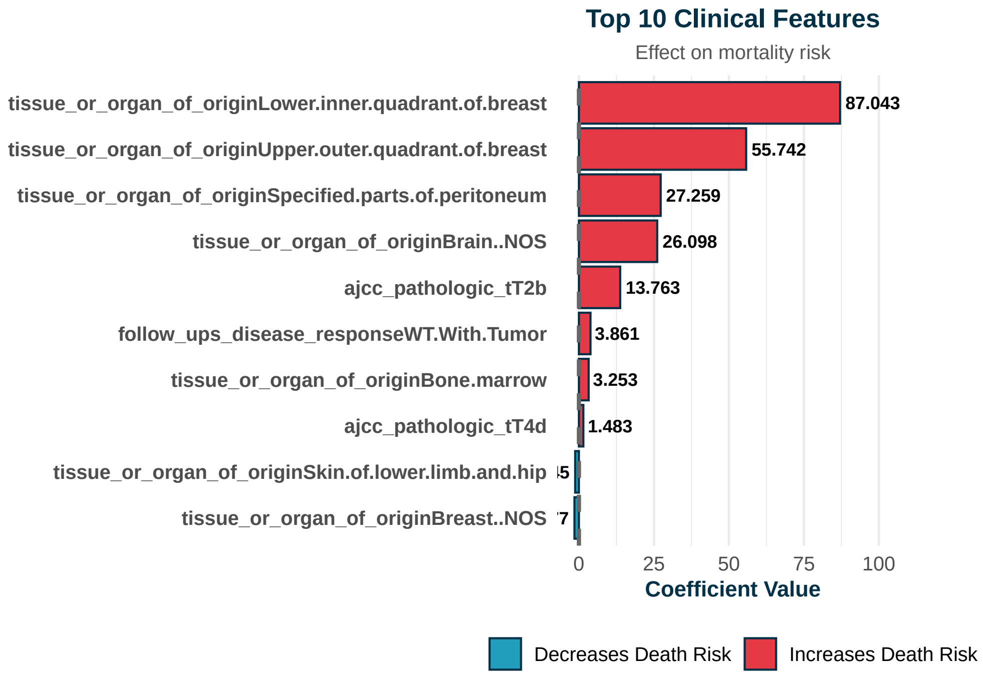
Applying SMOTE completely changed the behavior of all models by correcting the strong class imbalance: recall, F1-score, and AUC all improved substantially. Lasso and Elastic Net became the top-performing methods, achieving the best balance between precision and recall, with AUC values consistently around 0.88–0.90 across 20–500 gene sets. Elastic Net showed the strongest overall performance, indicating that death-related gene expression signals occur in correlated gene groups. Ridge, which previously collapsed in high dimensions, became functional after SMOTE but still performed weaker than L1-based methods. Overall, SMOTE + regularized models demonstrate that moderate gene sets (20–500 genes) contain the most predictive information, and sparse models such as Lasso and Elastic Net should be preferred for final model selection.

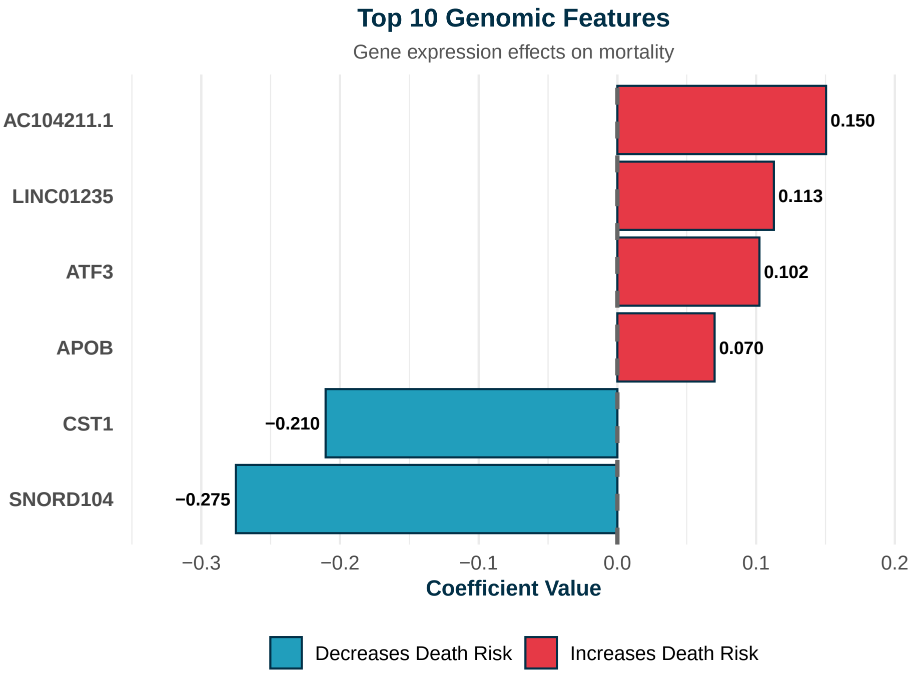
```r
gene_names <- colnames(GeneX)
results <- feature_importance(
    model_obj = unilasso_smote$results[[7]]$model_obj
    , model_name = "UniLasso + SMOTE + TOP20"
    , top_n = 20
    , gene_names = gene_names
)
```

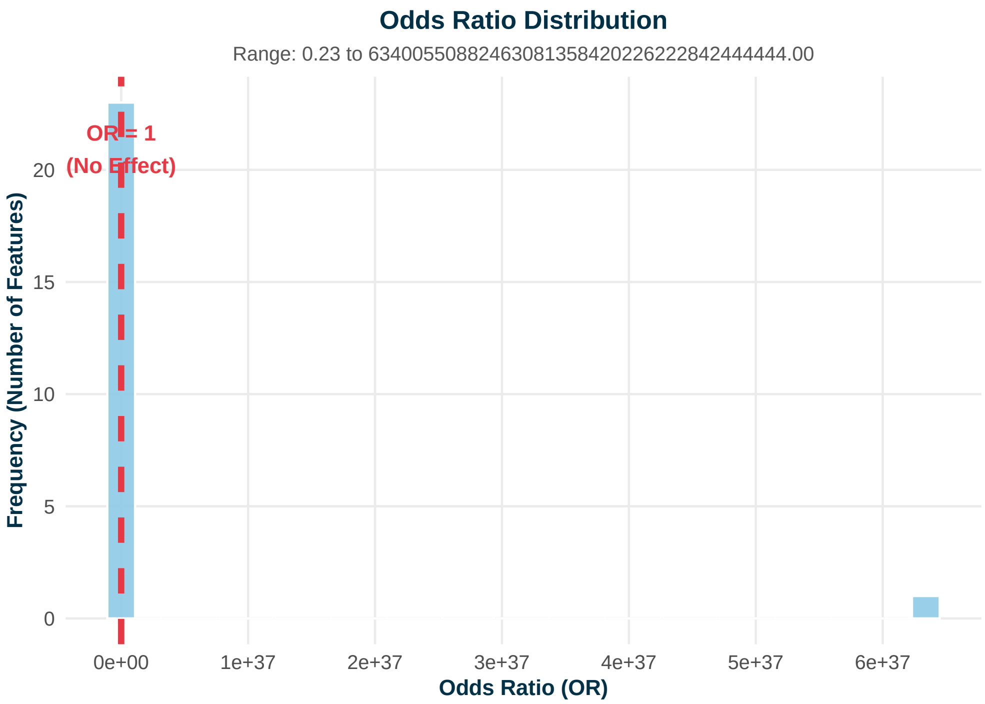
```
##
## ================================================================================
## FEATURE IMPORTANCE ANALYSIS: UniLasso + SMOTE + TOP20
## = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = =
##
## === SUMMARY ===
## Total features selected: 24
##   Clinical: 18
##   Genomic: 6
```

```
##
## Direction:
##   Increases death risk: 18
##   Decreases death risk: 6
##
## Coefficient range:
##   Min: -1.4769
##   Max: 87.0425
##   Mean (absolute): 9.4058
##
## Odds Ratio range:
##   Min: 0.2283
##   Max: 6.340055e+37
##
## === TOP 20 FEATURES ===
##
##  Rank                                                  Feature     Type
##     1 tissue_or_organ_of_originLower.inner.quadrant.of.breast Clinical
##     2 tissue_or_organ_of_originUpper.outer.quadrant.of.breast Clinical
##     3  tissue_or_organ_of_originSpecified.parts.of.peritoneum Clinical
##     4                    tissue_or_organ_of_originBrain..NOS Clinical
##     5                                   ajcc_pathologic_tT2b Clinical
##     6                follow_ups_disease_responseWT.With.Tumor Clinical
##     7                    tissue_or_organ_of_originBone.marrow Clinical
##     8                                   ajcc_pathologic_tT4d Clinical
##     9                    tissue_or_organ_of_originBreast..NOS Clinical
##    10     tissue_or_organ_of_originSkin.of.lower.limb.and.hip Clinical
##    11                       ethnicitynot.hispanic.or.latino Clinical
##    12                                   ajcc_pathologic_tT1b Clinical
##    13                      tissue_or_organ_of_originBone..NOS Clinical
##    14                                              SNORD104  Genomic
##    15                                 days_to_last_follow_up Clinical
##    16                tissue_or_organ_of_originLower.limb..NOS Clinical
##    17                                                  CST1  Genomic
##    18                                            age_at_index Clinical
##    19                                             AC104211.1  Genomic
##    20                                   ajcc_pathologic_tT4b Clinical
##  Coefficient   Odds_Ratio            Direction
##      87.0425 6.340055e+37 Increases Death Risk
##      55.7423 1.616428e+24 Increases Death Risk
##      27.2587 6.891511e+11 Increases Death Risk
##      26.0978 2.158425e+11 Increases Death Risk
##      13.7632 9.490810e+05 Increases Death Risk
##       3.8615 4.753660e+01 Increases Death Risk
##       3.2528 2.586320e+01 Increases Death Risk
##       1.4833 4.407300e+00 Increases Death Risk
##      -1.4769 2.283000e-01 Decreases Death Risk
##      -1.2451 2.879000e-01 Decreases Death Risk
##       1.1621 3.196700e+00 Increases Death Risk
##      -0.8753 4.167000e-01 Decreases Death Risk
##       0.7465 2.109700e+00 Increases Death Risk
##      -0.2750 7.595000e-01 Decreases Death Risk
##       0.2416 1.273300e+00 Increases Death Risk
##      -0.2244 7.990000e-01 Decreases Death Risk
##      -0.2104 8.103000e-01 Decreases Death Risk
##       0.1875 1.206200e+00 Increases Death Risk
##       0.1504 1.162300e+00 Increases Death Risk
##       0.1165 1.123600e+00 Increases Death Risk
## 
## === TOP CLINICAL FEATURES ===
## 
##                                                   Feature Coefficient
##  tissue_or_organ_of_originLower.inner.quadrant.of.breast     87.0425
##  tissue_or_organ_of_originUpper.outer.quadrant.of.breast     55.7423
##   tissue_or_organ_of_originSpecified.parts.of.peritoneum     27.2587
##                       tissue_or_organ_of_originBrain..NOS     26.0978
##                                     ajcc_pathologic_tT2b     13.7632
##                  follow_ups_disease_responseWT.With.Tumor      3.8615
##                      tissue_or_organ_of_originBone.marrow      3.2528
##                                     ajcc_pathologic_tT4d      1.4833
##                      tissue_or_organ_of_originBreast..NOS     -1.4769
##      tissue_or_organ_of_originSkin.of.lower.limb.and.hip     -1.2451
##    Odds_Ratio            Direction
##  6.340055e+37 Increases Death Risk
##  1.616428e+24 Increases Death Risk
##  6.891511e+11 Increases Death Risk
##  2.158425e+11 Increases Death Risk
##  9.490810e+05 Increases Death Risk
##  4.753660e+01 Increases Death Risk
##  2.586320e+01 Increases Death Risk
##  4.407300e+00 Increases Death Risk
##  2.283000e-01 Decreases Death Risk
##  2.879000e-01 Decreases Death Risk
## 
## === TOP GENOMIC FEATURES ===
## 
##     Feature Coefficient Odds_Ratio            Direction
##    SNORD104     -0.2750     0.7595 Decreases Death Risk
##        CST1     -0.2104     0.8103 Decreases Death Risk
##  AC104211.1      0.1504     1.1623 Increases Death Risk
##   LINC01235      0.1128     1.1194 Increases Death Risk
##        ATF3      0.1024     1.1078 Increases Death Risk
##        APOB      0.0701     1.0726 Increases Death Risk
## 
##                                               Feature     Type Coefficient
##                  tissue_or_organ_of_originBreast..NOS Clinical     -1.4769
##  tissue_or_organ_of_originSkin.of.lower.limb.and.hip Clinical     -1.2451
##                                  ajcc_pathologic_tT1b Clinical     -0.8753
##                                              SNORD104  Genomic     -0.2750
##              tissue_or_organ_of_originLower.limb..NOS Clinical     -0.2244
##  Odds_Ratio
##      0.2283
##      0.2879
##      0.4167
##      0.7595
##      0.7990
## 
##                                                   Feature     Type Coefficient
## tissue_or_organ_of_originLower.inner.quadrant.of.breast Clinical     87.0425
## tissue_or_organ_of_originUpper.outer.quadrant.of.breast Clinical     55.7423
##  tissue_or_organ_of_originSpecified.parts.of.peritoneum Clinical     27.2587
##                        tissue_or_organ_of_originBrain..NOS Clinical     26.0978
##                                     ajcc_pathologic_tT2b Clinical     13.7632
##    Odds_Ratio
##  6.340055e+37
##  1.616428e+24
##  6.891511e+11
##  2.158425e+11
##  9.490810e+05
```




Top 10 Clinical Features
Effect on mortality risk
tissue_or_organ_of_originLower.inner.quadrant.of.breast
tissue_or_organ_of_originUpper.outer.quadrant.of.breast
tissue_or_organ_of_originSpecified.parts.of.peritoneum
tissue_or_organ_of_originBrain..NOS
ajcc_pathologic_tT2b
follow_ups_disease_responseWT.With.Tumor
tissue_or_organ_of_originBone.marrow
ajcc_pathologic_tT4d
tissue_or_organ_of_originSkin.of.lower.limb.and.hip
tissue_or_organ_of_originBreast..NOS
87.043
55.742
27.259
26.098
13.763
3.861
3.253
1.483
5
7
0
25
50
75
100
Coefficient Value
Decreases Death Risk
Increases Death Risk


Top 10 Genomic Features
Gene expression effects on mortality
AC104211.1
LINC01235
ATF3
APOB
CST1
SNORD104
0.150
0.113
0.102
0.070
−0.210
−0.275
−0.3
−0.2
−0.1
0.0
0.1
0.2
Coefficient Value
Decreases Death Risk
Increases Death Risk



```
##
## Exported feature importance to: model_metrics/UniLasso__SMOTE__TOP20_feature_importance.csv
```

Evaluation

We can draw the evaluation from this study. The conducting study on the breast cancer analysis gives us good insight information about the classification of cancer survival outcomes.

The best stable performance model is **UniLasso + SMOTE** with F1-Score of **0.737**, using Clinical + TOP20 genes as features. The SMOTE resampling method addresses the class imbalance problem (5:1 ratio of Alive:Dead) in the original dataset.

Model Performance:

- F1-Score: 0.737 (best balance between precision and recall)
- Recall: 70.0% (identifies 70% of deaths)
- Precision: 77.8% (78% of predicted deaths are correct)
- AUC: 0.893

Feature Analysis

The model selected 24 features (18 clinical, 6 genomic). However, some clinical categories showed extreme coefficients (OR $> 10^6$) due to sparse observations, indicating unreliable estimates from quasi-perfect separation. We focus interpretation on stable predictors with reasonable odds ratios (OR between 0.1 and 100).

Reliable Clinical Features (8 features)

Disease Response:

Feature	Odds Ratio	Interpretation
With Tumor	47.54	Residual tumor after treatment indicates treatment failure; 47x higher death risk – strongest clinically meaningful predictor

Tumor Stage (AJCC Pathologic T):

The tumor staging follows the American Joint Committee on Cancer (AJCC) TNM system 8th Edition (Cancer Research UK, 2024; American Cancer Society, 2021).

Feature	Odds Ratio	Definition	Interpretation
T4d	4.41	Inflammatory carcinoma – a rare and aggressive type of breast cancer (Cancer Research UK, 2024)	4.4x higher death risk
T4b	1.12	Cancer has spread into the skin with possible swelling (Cancer Research UK, 2024)	12% higher death risk
T1b	0.42	Tumor size between 0.5 cm and 1 cm (Cancer Research UK, 2024)	58% lower death risk (protective – early detection)

Demographics:

Feature	Odds Ratio	Interpretation
Ethnicity: not hispanic/latino	3.20	3.2x higher risk; may reflect genetic, socioeconomic, or healthcare access factors
Age at index	1.21	Each year increase in age raises death risk by 21%

Other:

Feature	Odds Ratio	Interpretation
Days to last follow-up	1.27	Longer follow-up allows more time to observe death events
Prior treatment: Yes	1.04	4% higher risk; patients with prior treatment may have recurrent or resistant disease

Genomic Features (6 genes)

Protective Genes (higher expression = lower death risk):

Gene	Odds Ratio	Biological Function
SNORD104	0.76	Small nucleolar RNA (snoRNA) involved in RNA modification and regulation of cell cycle, proliferation, and apoptosis in tumor cells (Lu et al., 2022). In our breast cancer cohort, higher expression is associated with 24% lower death risk.
CST1	0.81	Cystatin SN, a cysteine protease inhibitor that interacts with GPX4, a key protein regulating ferroptosis (Wang et al., 2022). Higher expression shows 19% lower death risk.

Risk Genes (higher expression = higher death risk):

Gene	Odds Ratio	Biological Function
AC104211.1	1.16	Long non-coding RNA (lncRNA); regulatory role in gene expression; 16% higher death risk
LINC01235	1.12	Long intergenic non-coding RNA; emerging evidence links lncRNAs to cancer progression; 12% higher death risk
ATF3	1.11	Activating Transcription Factor 3, a stress-induced transcription factor that plays vital roles in modulating metabolism, immunity, and oncogenesis (Wang et al., 2020). ATF3 gene copy number is greater than 2 in approximately 80% of breast tumors and its protein level is elevated in approximately 50% of tumors (Yin et al., 2008). 11% higher death risk.
APOB	1.07	Apolipoprotein B, involved in lipid metabolism. Loss of APOB in hepatocellular carcinoma is associated with poor survival, suggesting potential tumor suppressive activity (Lee et al., 2019). 7% higher death risk.

Key Findings

1. **Residual tumor is the strongest reliable predictor** – patients with tumor remaining after treatment (OR=47.5) have dramatically worse outcomes

2. **Tumor stage matters** – T4d (inflammatory breast cancer, OR=4.4) increases risk; T1b (small tumor 0.5-1cm, OR=0.42) is protective

3. **Age increases risk** – each additional year increases death risk by 21%

4. **Genomic markers provide modest but stable contribution** – all 6 genes show reasonable OR (0.76-1.16)

5. **Non-coding RNAs are emerging biomarkers** – 3 of 6 genes (SNORD104, AC104211.1, LINC01235) are non-coding RNAs

6. **Sparse clinical categories are unreliable** – extreme OR values for rare tumor locations should be interpreted with caution

Conclusion

The UniLasso + SMOTE model effectively classifies breast cancer survival using clinical and genomic features. The most actionable finding is that **residual tumor status** strongly predicts mortality, while **early-stage tumors (T1b)** have significantly better outcomes. Gene expression markers, particularly **ATF3** (stress response) and **CST1** (protease inhibitor), provide biological insight into tumor progression. SMOTE resampling was essential for handling the 5:1 class imbalance.

References

- American Cancer Society. (2021). Stages of Breast Cancer. https://www.cancer.org/cancer/types/breast-cancer/understanding-a-breast-cancer-diagnosis/stages-of-breast-cancer.html
- Cancer Research UK. (2024). TNM staging for breast cancer. https://www.cancerresearchuk.org/about-cancer/breast-cancer/stages-grades/tnm-staging
- Lee, Y. et al. (2019). Clinical significance of APOB inactivation in hepatocellular carcinoma. Experimental and Molecular Medicine, 51, 1-12.
- Lu, B. et al. (2022). C/D box small nucleolar RNA SNORD104 promotes endometrial cancer by regulating the 2-O-methylation of PARP1. Journal of Translational Medicine, 20, 618.
- Wang, L. et al. (2022). CST1 inhibits ferroptosis and promotes gastric cancer metastasis by regulating GPX4 protein stability via OTUB1. Oncogene, 41, 5227-5238.
- Wang, Z. et al. (2020). Master Regulator Activating Transcription Factor 3 (ATF3) in Metabolic Homeostasis and Cancer. Frontiers in Endocrinology, 11, 556.
- Yin, X. et al. (2008). A potential dichotomous role of ATF3, an adaptive-response gene, in cancer development. Oncogene, 27, 2118-2127.