task4

zhexuan

2025-12-06

Task 4

In this task, we evaluated a classification model for the early detection of lung cancer using primary care data. The model outputs a predicted probability of lung cancer for each patient, and it was externally validated on a retrospective dataset of 1,000 matched patients with and without a lung cancer diagnosis. The main goals of the analysis were to assess the performance of the model, and then choose appropriate thresholds, taking into account the severe consequences of missing true lung cancer cases.

Data

The dataset contains 1,000 patients with observed outcome (0 for no lung cancer and 1 for lung cancer) and the model-predicted probability of lung cancer.

```
dat <- read.csv(″data_t4.csv″)

str(dat)
```

```
## ’data.frame’:    1000 obs. of  3 variables:
##  $ X         : int  1 2 3 4 5 6 7 8 9 10 ...
##  $ labels_obs: int  1 0 1 0 0 1 0 0 0 1 ...
##  $ prob_pred : num  0.586 0.348 0.729 0.655 0.432 ...
```

```
summary(dat)
```

```
##        X            labels_obs      prob_pred     
##  Min.   :   1.0   Min.   :0.000   Min.   :0.0588  
##  1st Qu.: 250.8   1st Qu.:0.000   1st Qu.:0.4770  
##  Median : 500.5   Median :1.000   Median :0.6207  
##  Mean   : 500.5   Mean   :0.507   Mean   :0.6121  
##  3rd Qu.: 750.2   3rd Qu.:1.000   3rd Qu.:0.7567  
##  Max.   :1000.0   Max.   :1.000   Max.   :1.0000  
```

Model Performance

To assess how well the model can distinguish between patients with and without lung cancer, we first computed a receiver operating characteristic (ROC) curve. The x-axis is actually specificity, which is plotted from 1 down to 0, so it looks reversed compared with the standard “FPR on the x-axis” ROC plots. The dashed diagonal line represents the performance of a classifier that randomly guesses class labels.

The resulting area under the ROC curve (AUC) was 0.8533. An AUC of 0.5 would indicate a model that performs no better than random guessing, while an AUC of 1.0 would correspond to perfect discrimination. An AUC around 0.85 therefore suggests that the model has good discriminative ability.

```
roc_obj <- roc(response = dat$labels_obs,
               predictor = dat$prob_pred)
```

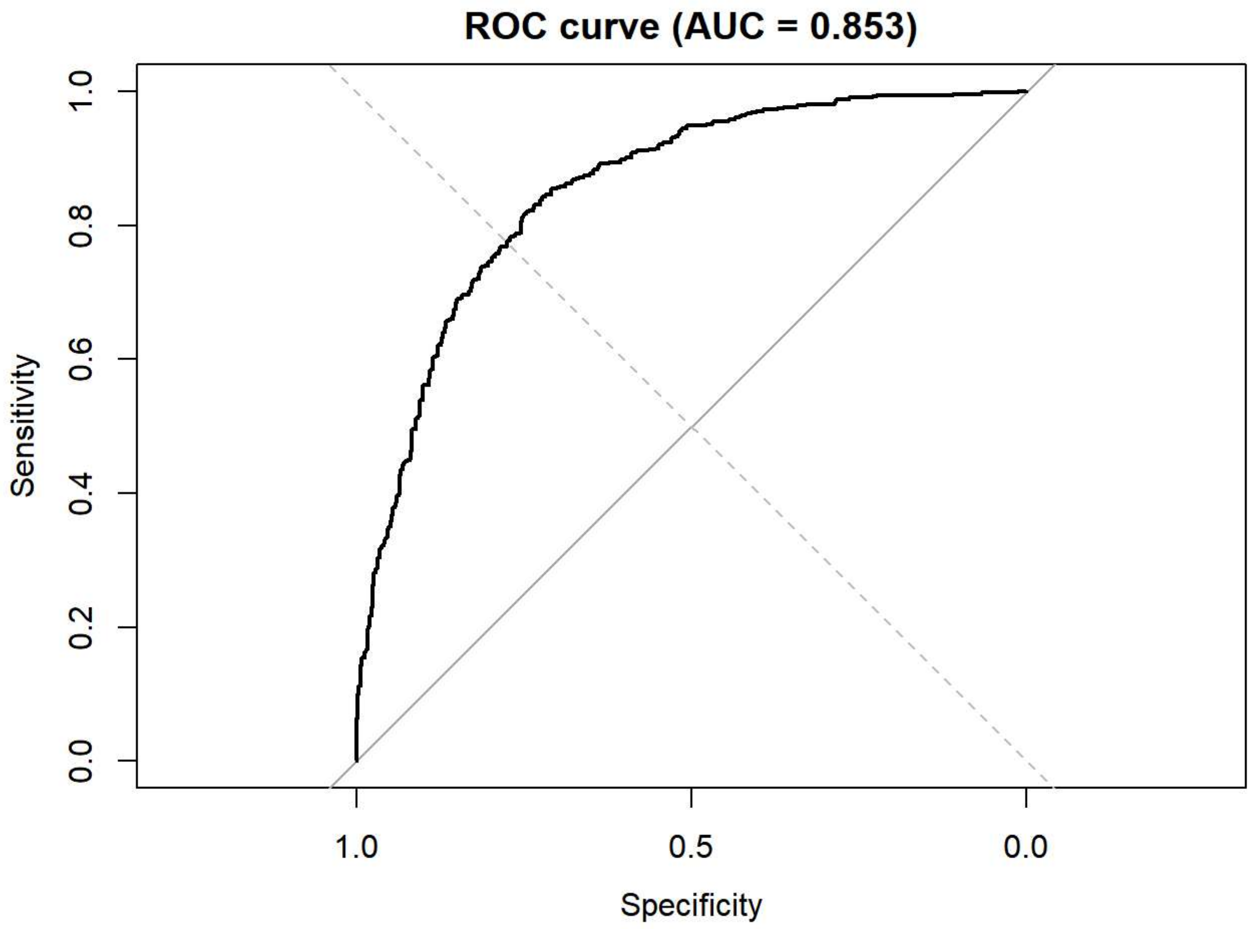
```
## Setting levels: control = 0, case = 1
```

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

```
# AUC
auc_value <- auc(roc_obj)
auc_value
```

```
## Area under the curve: 0.8533
```

```
# ROC curve
plot(roc_obj,
     main = paste0("ROC curve (AUC = ", round(auc_value, 3), ")"))
abline(a = 0, b = 1, lty = 2, col = "gray")
```



Precision–recall analysis could link recall (sensitivity) to the precision of the positive predictions. This is especially relevant in an early detection context, where the clinical priority is usually to maximize recall while keeping precision at a clinically acceptable level.

The area under the PR curve was 0.844, indicating that the model maintains a favorable trade-off between recall and precision across a range of thresholds.

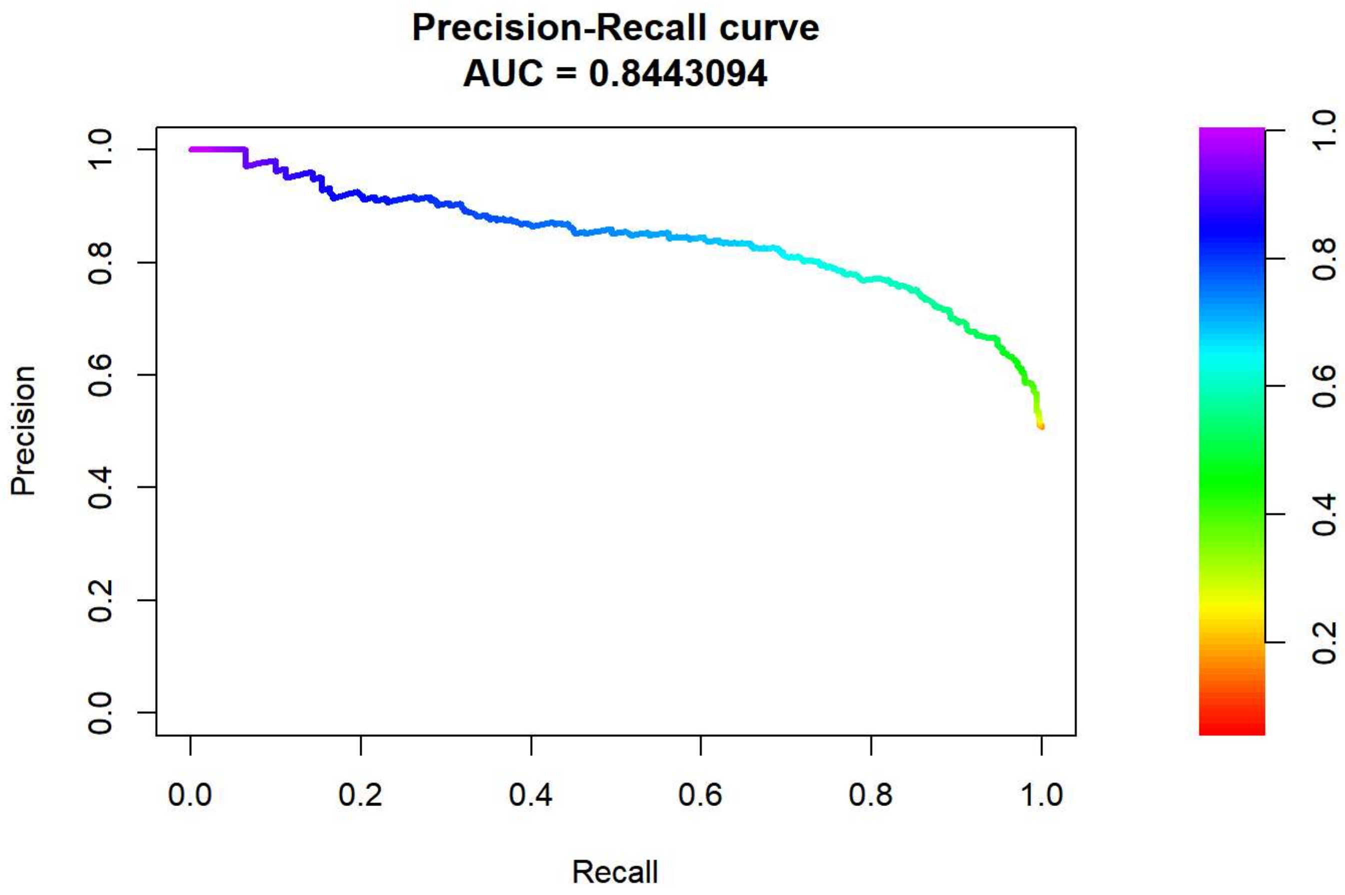
```
scores_pos <- dat$prob_pred[dat$labels_obs == 1]  # cancer
scores_neg <- dat$prob_pred[dat$labels_obs == 0]  # non-cancer

## PR curve
pr_obj <- pr.curve(scores.class0 = scores_pos,
                   scores.class1 = scores_neg,
                   curve = TRUE)

pr_obj$auc.integral  # PR AUC
```

```
## [1] 0.8443094
```

```
plot(pr_obj,
     main = paste0("Precision-Recall curve"))
```



For clinical decision-making, it is also important that the predicted probabilities themselves are well-calibrated. To assess calibration, we first computed the Brier score, which is the mean squared error between the predicted probabilities and the observed outcomes (coded as 0 or 1). A lower value indicates better probabilistic predictions, with 0 representing perfect predictions. A Brier score of around 0.18 suggests that the model’s probability estimates are reasonably good but not perfect, leaving some room for miscalibration.

To visualize calibration in more detail across the risk spectrum, we grouped the patients into ten bins based on the deciles of the predicted risk. Because the sample size is 1000 and the probabilities are fairly smoothly distributed, each bin ended up containing exactly 100 observations.

For each bin, we computed the average predicted probability and the observed proportion of lung cancer cases. And we plotted these as a calibration plot. In this plot, the dashed diagonal line corresponds to perfect calibration, where the predicted and observed probabilities are equal.

In our results, the points for the lower-risk bins lie below the diagonal, which means that in those groups the observed cancer rate is slightly lower than the predicted risk. In other words, for low predicted probabilities, the model tends to overestimate the absolute risk a bit. For higher predicted risk levels (around a mean predicted probability of 0.7 and above), the points move above the diagonal, which indicates that in those high-risk groups the observed cancer rate is actually higher than the predicted risk. This pattern suggests that the model becomes somewhat too conservative at the highest risk levels, underestimating risk in those patients who are truly at very high risk.

```
# Brier score: MSE
brier_score <- mean( (dat$prob_pred - dat$labels_obs)^2 )
brier_score
```

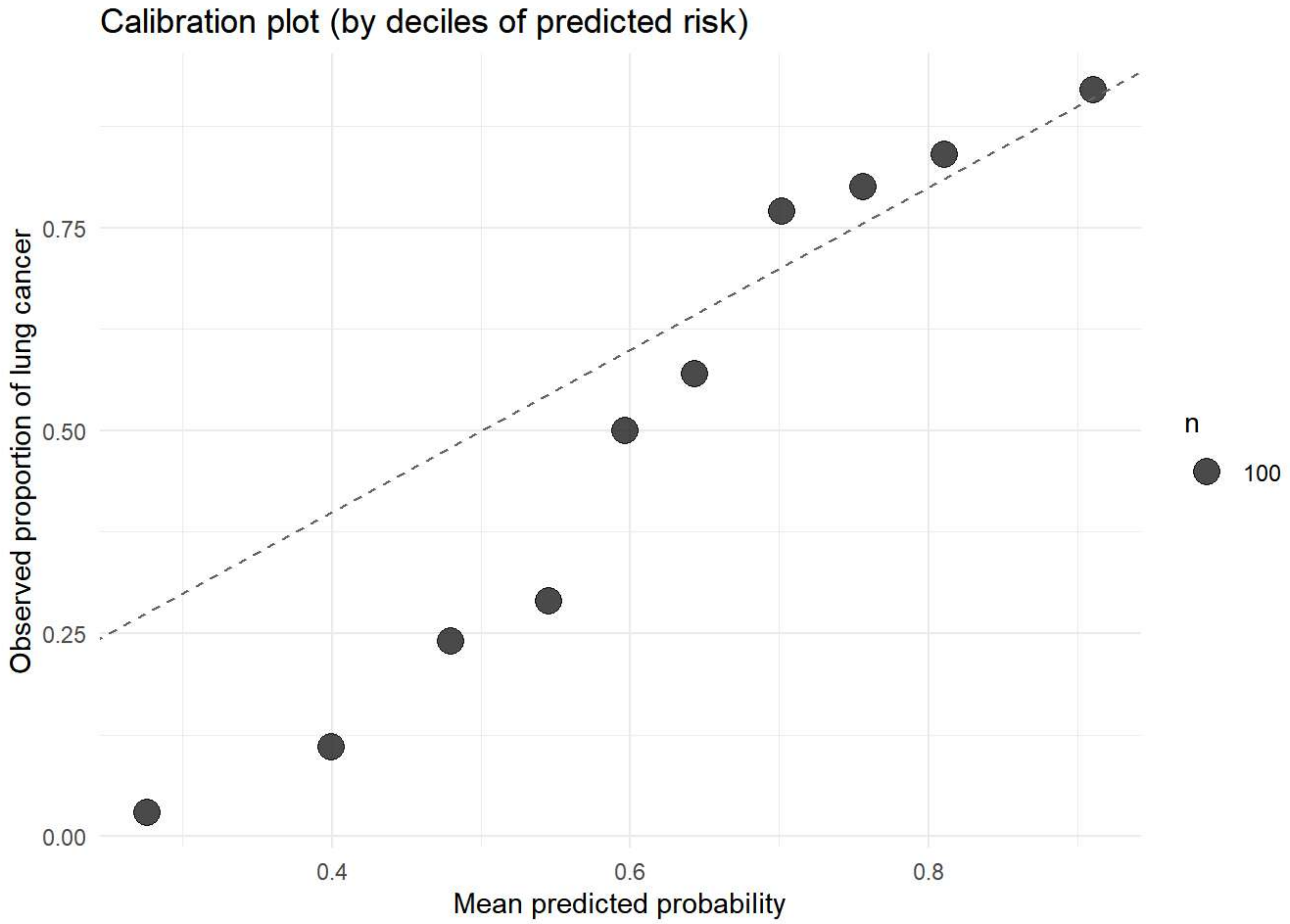
```
## [1] 0.1836193
```

```
# split prob_pred into 10 groups
dat_cal <- dat %>%
  mutate(bin = cut(prob_pred,
                   breaks = quantile(prob_pred, probs = seq(0, 1, by = 0.1)),
                   include.lowest = TRUE)) %>%
  group_by(bin) %>%
  summarise(
    mean_pred = mean(prob_pred),
    obs_rate  = mean(labels_obs),
    n = n()
  )

dat_cal
```

```
## # A tibble: 10 × 4
##    bin            mean_pred obs_rate     n
##    <fct>              <dbl>    <dbl> <int>
##  1 [0.0588,0.345]     0.276     0.03   100
##  2 (0.345,0.443]      0.400     0.11   100
##  3 (0.443,0.52]       0.480     0.24   100
##  4 (0.52,0.57]        0.546     0.29   100
##  5 (0.57,0.621]       0.597     0.5    100
##  6 (0.621,0.674]      0.643     0.57   100
##  7 (0.674,0.729]      0.702     0.77   100
##  8 (0.729,0.782]      0.756     0.8    100
##  9 (0.782,0.845]      0.811     0.84   100
## 10 (0.845,1]          0.911     0.92   100
```

```
# calibration plot
ggplot(dat_cal, aes(x = mean_pred, y = obs_rate, size = n)) +
  geom_point(alpha = 0.7) +
  geom_abline(intercept = 0, slope = 1, linetype = "dashed", color = "gray40") +
  labs(x = "Mean predicted probability",
       y = "Observed proportion of lung cancer",
       title = "Calibration plot (by deciles of predicted risk)") +
  theme_minimal()
```



Threshold Determination

The choice of threshold has direct clinical implications, especially in a setting such as lung cancer detection where the cost of missing a true cancer case is substantial.

We first used the ROC curve to identify the Youden-optimal threshold, which maximizes the quantity sensitivity + specificity − 1. This threshold represents the point on the ROC curve that achieves the best overall balance between correctly identifying cancer cases and correctly identifying non-cancer cases. Based on our data, the Youden index was maximized at a threshold of 0.6025, providing a statistically well-balanced operating point. However, this balance does not necessarily reflect clinical priorities, because it treats false negatives and false positives as equally costly.

```
tpr <- roc_obj$sensitivities
fpr <- 1 - roc_obj$specificities
thresholds <- roc_obj$thresholds

# Youden
youden <- tpr - fpr

# threshold for the maximum Youden
optimal_threshold <- thresholds[which.max(youden)]

optimal_threshold
```

```
## [1] 0.6025
```

To better understand how the model behaves across the full range of thresholds, we evaluated performance at every value from 0 to 1 in increments of 0.01. For each threshold we calculated sensitivity, specificity, PPV, NPV, and accuracy, producing a detailed map of how model performance shifts as the threshold changes. As a reference point, we also examined the commonly used default threshold of 0.5. At this threshold, the sensitivity was high (0.949), but specificity dropped to 0.507. It indicated that although most cancer cases were correctly detected, the model produced many false alarms in practice. At the threshold 0.6, where we derived maximum Youden index, the sensitivity was 0.82 and the specificity was 0.74, with a PPV of 0.77, an NPV of 0.80, and an overall accuracy of 0.78. These values indicate a well-balanced operating point from a purely statistical perspective, with both sensitivity and specificity at reasonably high levels.

The comparison between 0.50 and 0.60 illustrates an important trade-off: raising the threshold improves specificity and reduces false positives, but at the cost of missing more true cancer cases.

```r
# calculate all the metrics
metrics_at_threshold <- function(threshold, labels, probs) {
  pred_class <- ifelse(probs >= threshold, 1, 0)

  TP <- sum(pred_class == 1 & labels == 1)
  FP <- sum(pred_class == 1 & labels == 0)
  TN <- sum(pred_class == 0 & labels == 0)
  FN <- sum(pred_class == 0 & labels == 1)

  sensitivity <- ifelse((TP + FN) > 0, TP / (TP + FN), NA)
  specificity <- ifelse((TN + FP) > 0, TN / (TN + FP), NA)
  PPV <- ifelse((TP + FP) > 0, TP / (TP + FP), NA)
  NPV <- ifelse((TN + FN) > 0, TN / (TN + FN), NA)
  accuracy <- (TP + TN) / (TP + TN + FP + FN)

  ## Youden index (J)
  youden <- sensitivity + specificity - 1

  data.frame(
    threshold = threshold,
    TP = TP, FP = FP, TN = TN, FN = FN,
    sensitivity = sensitivity,
    specificity = specificity,
    PPV = PPV,
    NPV = NPV,
    accuracy = accuracy,
    youden = youden
  )
}

# list of thresholds
thresholds <- seq(0, 1, by = 0.01)

# calculate metrics of all the thresholds
metrics_all <- do.call(rbind,
                       lapply(thresholds, metrics_at_threshold,
                              labels = dat$labels_obs,
                              probs = dat$prob_pred))

head(metrics_all)
```

```
##   threshold  TP  FP TN FN sensitivity specificity   PPV NPV accuracy youden
## 1      0.00 507 493  0  0           1           0 0.507  NA    0.507      0
## 2      0.01 507 493  0  0           1           0 0.507  NA    0.507      0
## 3      0.02 507 493  0  0           1           0 0.507  NA    0.507      0
## 4      0.03 507 493  0  0           1           0 0.507  NA    0.507      0
## 5      0.04 507 493  0  0           1           0 0.507  NA    0.507      0
## 6      0.05 507 493  0  0           1           0 0.507  NA    0.507      0
```

```
# reference: threshold = 0.5
metrics_at_threshold(0.5, dat$labels_obs, dat$prob_pred)
```

```
##   threshold  TP  FP  TN FN sensitivity specificity       PPV       NPV accuracy
## 1       0.5 481 243 250 26   0.9487179   0.5070994 0.6643646 0.9057971    0.731
##      youden
## 1 0.4558173
```

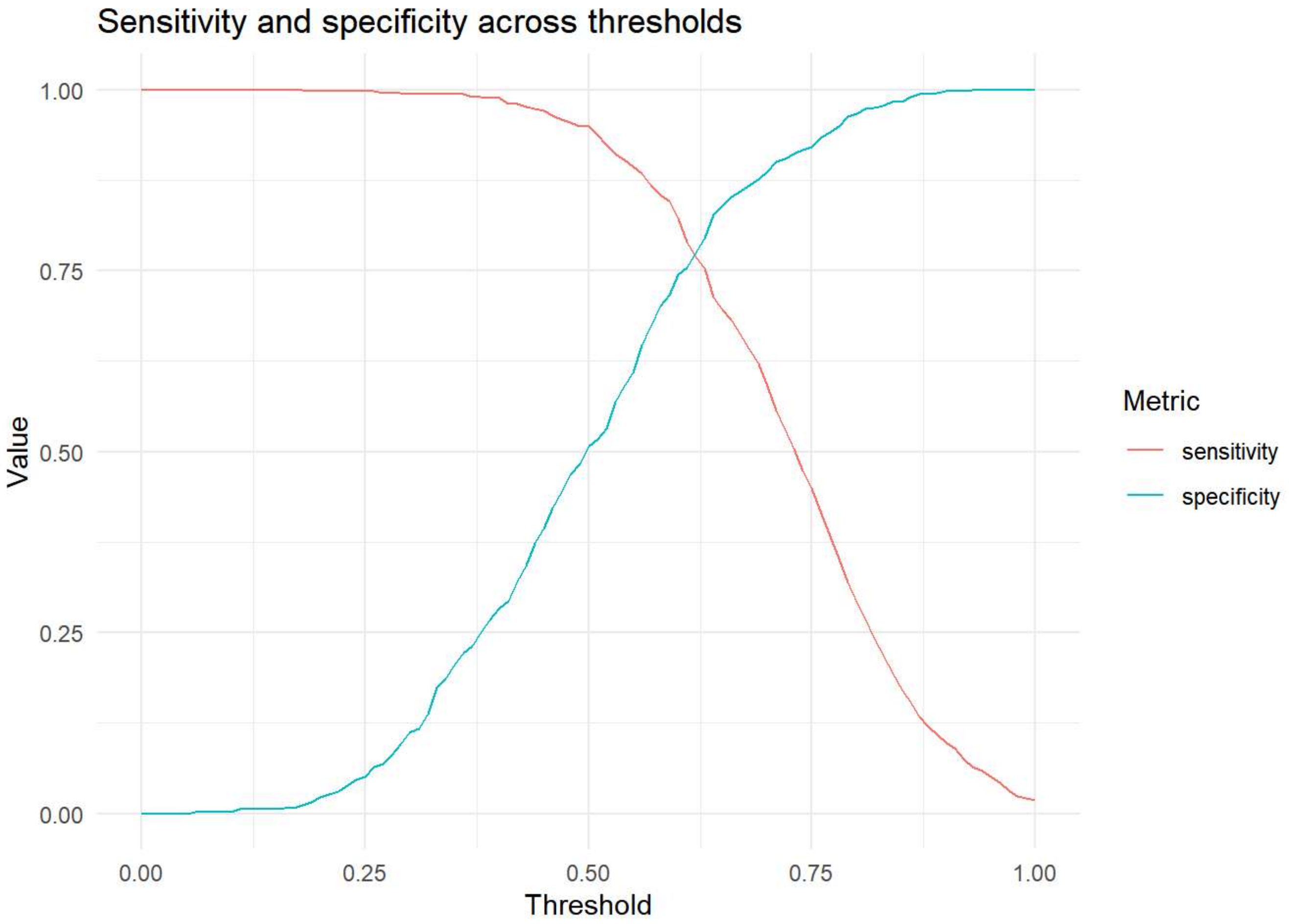
```
# threshold with maximum Youden
metrics_at_threshold(0.6, dat$labels_obs, dat$prob_pred)
```

```
##   threshold  TP  FP  TN FN sensitivity specificity       PPV       NPV accuracy
## 1       0.6 417 126 367 90   0.8224852   0.7444219 0.7679558 0.8030635    0.784
##      youden
## 1 0.5669071
```

We plotted sensitivity and specificity versus threshold. The two curves intersected at a threshold of approximately 0.62. We also plotted accuracy, PPV, and NPV across thresholds. Accuracy peaked close to 0.60, which is consistent with the Youden-optimal threshold and reflects the fact that accuracy is maximized when sensitivity and specificity are jointly high in a balanced dataset. PPV increased steadily with higher thresholds. NPV showed a gradual downward trend with a noticeable drop at lower thresholds.

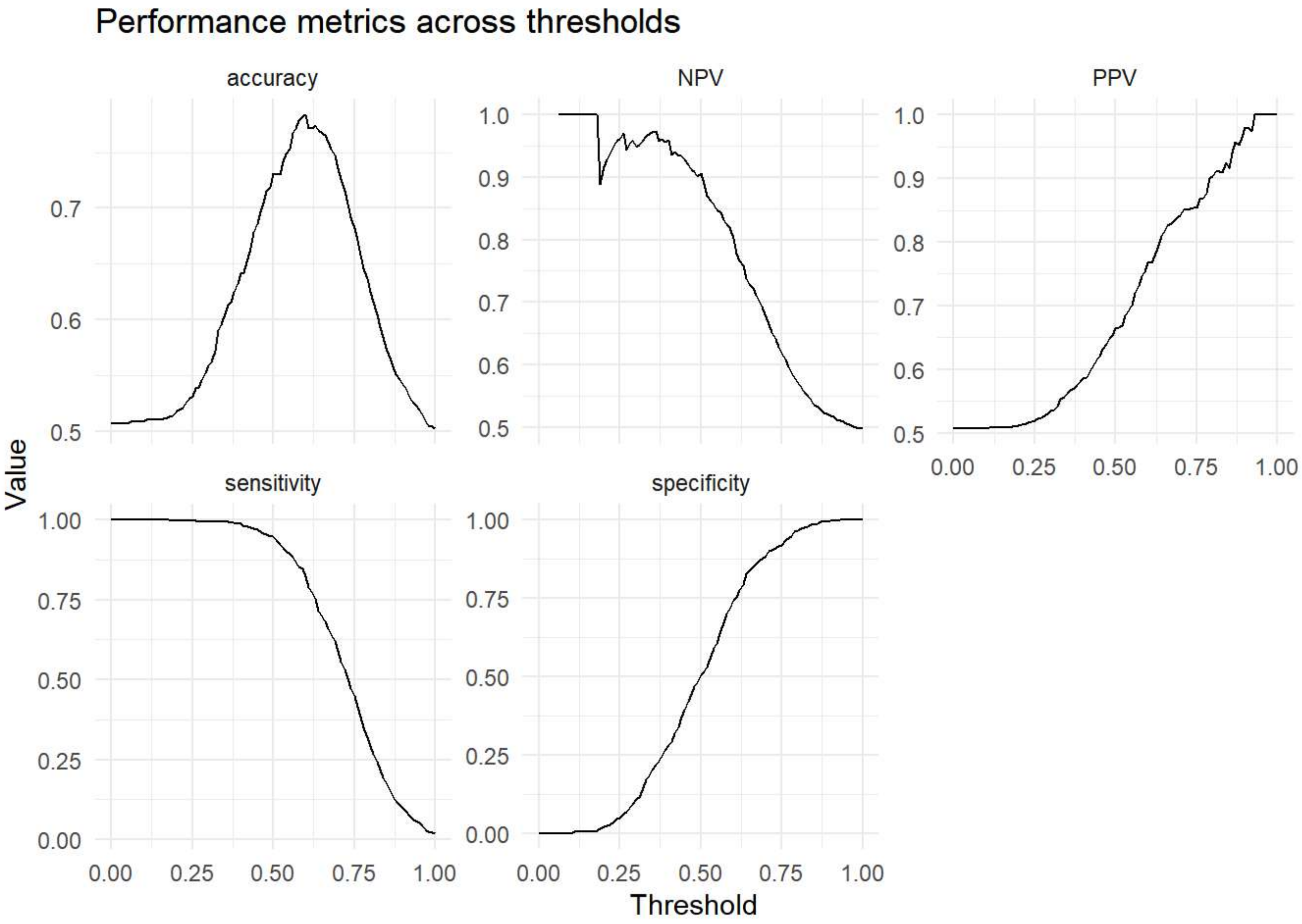
```
metrics_plot <- metrics_all %>%
  select(threshold, sensitivity, specificity) %>%
  pivot_longer(cols = c(sensitivity, specificity),
               names_to = "metric",
               values_to = "value")

ggplot(metrics_plot, aes(x = threshold, y = value, color = metric)) +
  geom_line() +
  labs(x = "Threshold",
       y = "Value",
       color = "Metric",
       title = "Sensitivity and specificity across thresholds") +
  theme_minimal()
```



```
metrics_plot_all <- metrics_all %>%
  select(threshold, sensitivity, specificity, PPV, NPV, accuracy) %>%
  pivot_longer(cols = -threshold,
               names_to = "metric",
               values_to = "value")

ggplot(metrics_plot_all, aes(x = threshold, y = value)) +
  geom_line() +
  facet_wrap(~ metric, scales = "free_y") +
  labs(x = "Threshold",
       y = "Value",
       title = "Performance metrics across thresholds") +
  theme_minimal()
```



Given the severe consequences of missing a lung cancer diagnosis, we then focused on thresholds that ensured high sensitivity (>= 0.90). Within this subset of thresholds, we selected the one with the highest specificity. It could minimize unnecessary false positives while still meeting the clinical requirement for high sensitivity. The threshold identified through this procedure was 0.54, with sensitivity 0.903 and specificity 0.590. Compared with the Youden-optimal threshold, this threshold sacrifices some specificity and overall accuracy but aligns more closely with the clinical objective of prioritizing patient safety and reducing the likelihood of missed cancer cases.

```
# thresholds with sensitivity >= 0.9
high_sens_candidates <- metrics_all %>%
  filter(sensitivity >= 0.9)

# threshold with maximum specificity
high_sens_best <- high_sens_candidates %>%
  arrange(desc(specificity)) %>%
  slice(1)

high_sens_best
```

```
##   threshold  TP  FP  TN FN sensitivity specificity       PPV       NPV accuracy
## 1      0.54 458 202 291 49   0.9033531   0.5902637 0.6939394 0.8558824    0.749
##      youden
## 1 0.4936167
```

We also explored an alternative strategy focused on the negative predictive value (NPV). In a screening context, a high NPV ensures that individuals classified as negative are indeed very unlikely to have the disease, reducing unnecessary follow-up investigations. We therefore identified all thresholds achieving NPV ≥ 0.90 and selected the one with the highest specificity within this set. And this procedure returned a threshold of

0.50, corresponding to sensitivity 0.95 and specificity 0.51. This result differs from the high-sensitivity threshold of 0.54 primarily because NPV depends not only on sensitivity but also on the underlying prevalence of the disease. In our balanced dataset, NPV remains relatively high over a broad range of thresholds, so the constraint NPV >= 0.90 does not push the threshold as low as in real-world low-prevalence screening settings. As a consequence, the NPV-based threshold resembles the default 0.50 threshold and places greater emphasis on avoiding false negatives while tolerating more false positives.

```
# thresholds with NPV >= 0.9
high_NPV_candidates <- metrics_all %>%
  filter(NPV >= 0.9)

# threshold with maximum specificity
high_NPV_best <- high_NPV_candidates %>%
  arrange(desc(specificity)) %>%
  slice(1)

high_NPV_best
```

```
##   threshold  TP  FP  TN FN sensitivity specificity       PPV       NPV accuracy
## 1       0.5 481 243 250 26   0.9487179   0.5070994 0.6643646 0.9057971    0.731
##      youden
## 1 0.4558173
```

Finally, we examined a cost-based approach to threshold selection, which explicitly encodes the idea that false negatives are more harmful than false positives in the context of lung cancer detection. We explored how the optimal threshold changes when different penalties are assigned to false negatives. As the relative cost of missing a cancer case increased, the cost-minimizing threshold shifted progressively downward, favoring thresholds that achieve higher sensitivity at the expense of specificity.

```
# set cost_FP = 1, try different cost_FN
cost_test <- function(cost_FN) {
  metrics_all %>%
    mutate(cost = cost_FN * FN + 1 * FP) %>%
    arrange(cost) %>%
    slice(1) %>%
    mutate(FN_cost = cost_FN)
}

cost_results <- bind_rows(
  cost_test(2),
  cost_test(4),
  cost_test(6),
  cost_test(10),
  cost_test(20)
)

cost_results
```

```
##   threshold  TP  FP  TN FN sensitivity specificity       PPV       NPV accuracy
## 1      0.56 448 174 319 59   0.8836292   0.6470588 0.7202572 0.8439153    0.767
## 2      0.50 481 243 250 26   0.9487179   0.5070994 0.6643646 0.9057971    0.731
## 3      0.44 494 309 184 13   0.9743590   0.3732252 0.6151930 0.9340102    0.678
## 4      0.40 501 353 140  6   0.9881657   0.2839757 0.5866511 0.9589041    0.641
## 5      0.36 504 384 109  3   0.9940828   0.2210953 0.5675676 0.9732143    0.613
##      youden cost FN_cost
## 1 0.5306880  292       2
## 2 0.4558173  347       4
## 3 0.3475841  387       6
## 4 0.2721413  413      10
## 5 0.2151782  444      20
```