Aqdas Hussain
Amberly Cavazos
Sarai Saenz
Jonathan McCanlas
Achyut Parmarthi

Bank Marketing Case Study Report

I. Executive Summary

This case study examines which customers are most likely to respond to a bank's term-deposit campaign, with a specific focus on how age and recency of contact (pdays: days since last contact) relate to response.

We analyzed a real-world dataset of ~41,000 customers. Because only ~11% responded "yes," we addressed class imbalance using ROSE to train fair, comparable models.

We built and compared two classification approaches: Logistic Regression (a standard business predictive model) and Linear Discriminant Analysis (LDA). Our optimization target was specificity—minimizing false positives—so outreach avoids contacting customers unlikely to say "yes," reducing wasted effort and cost.

Key insights

Age: Customers 26–35 show a higher likelihood of a positive response.

Recency: Customers contacted within the last 90 days are markedly more responsive.

Modeling: A tuned logistic regression achieved higher out-of-sample specificity at the selected threshold, with simpler implementation and monitoring. LDA performed competitively but required stricter preprocessing and assumptions.

Recommendation

Prioritize outreach to customers ages 26–35 and those contacted within 90 days. Use the logistic model with a specificity-oriented threshold to guide targeting; this concentrates effort on higher-yield segments and improves campaign efficiency. If space permits, report specificity, precision, and AUC for the chosen threshold to document expected impact.

II. The Problem

The task assigned for this case study is to develop predictive models to forecast customer responses to the bank's marketing campaigns aimed at encouraging subscriptions to longterm deposit accounts. To achieve this, we are required to build Logistic Regression and Linear Discriminant Analysis (LDA) models. These models will be compared based on their accuracy, sensitivity, and specificity to determine which model performs best in predicting the likelihood of customer subscription.

The goal of this study is to identify the most relevant customer characteristics and behavioral patterns that influence the likelihood of subscribing to the bank's deposit accounts. By accurately predicting customer responses, the bank can allocate its marketing resources more efficiently, targeting customers who are most likely to convert. This would not only help reduce marketing costs but also significantly improve conversion rates and the overall effectiveness of the campaigns.

The primary objectives of this study are to build and evaluate the Logistic Regression and LDA models, compare their performances in terms of accuracy, sensitivity, and specificity, and identify the most significant predictors of customer subscription. Additionally, this case study aims to recommend an optimal targeting strategy that will enhance the bank's marketing efforts and increase customer acquisition rates.

The remainder of the report will provide a brief overview of the relevant literature on predictive modeling techniques, a detailed discussion of the methodology used to process the dataset, and an analysis of the findings derived from the model comparisons.

III. Review of Related Literature

A. Carvalho, Miguel; Pinho, Armando J.; Brás, Susana — *"Resampling approaches to handle class imbalance: a review from a data perspective"* (Journal of Big Data, 2025)

https://journalofbigdata.springeropen.com/articles/10.1186/s40537-025-01119-4?utm_

- Important talk about ways to fix class imbalance, which is exactly what we had in our dataset since only about 11% of people said "yes." That's why we used ROSE in our code to make the model train more fairly

B. Moro, Sérgio; Laureano, Raul; Cortez, Paulo — *"A Data-Driven Approach to Predict the Success of Bank Telemarketing Calls for Selling Bank Term-Deposits"*, Decision Support Systems, 2014.

https://www.academia.edu/6412064/A_Data_Driven_Approach_to_Predict_the_Success_of_Bank_Telemarketing?utm_

- They showed that logistic regression works well and pointed out that things like pdays really affect the outcome. That connects directly to how we built our model and grouped variables in our code.

IV. Methodology

In this case study, we aim to develop and compare two predictive models: Logistic Regression and Linear Discriminant Analysis (LDA). The primary objective is to assess which model provides the best performance in predicting customer subscription likelihood for the bank's marketing campaign.

The data was first preprocessed by excluding the duration variable and performing feature engineering. New categorical variables such as age_group and pdays_group were created to improve model interpretability and segmentation. The dataset consists of both categorical and numerical variables, with categorical variables representing customer demographics and campaign history, while numerical variables capture interaction metrics and macroeconomic indicators. The dataset was then split into training (70%) and test (30%) sets using a random partitioning technique, ensuring the training data contains sufficient variability and representation of the target variable.

The Logistic Regression model was built to predict the binary outcome (yes or no) of customer subscription. This model operates under several assumptions, including a linear relationship between predictors and the log odds of the response variable, no multicollinearity, and normality of continuous predictors. These assumptions were tested and validated throughout the analysis. To optimize the threshold for classification, Youden’s index was employed to determine the best threshold that maximizes both sensitivity and specificity. Additionally, sensitivity-tuned thresholds were used to maximize the true positive rate, ensuring a balance between false positives and false negatives.

For the LDA model, which assumes normally distributed data for each class, we ensured that the predictors were properly dummied using dummy variables for categorical features. The LDA model was trained on the processed training set and evaluated on the test set. We used AUC (Area Under the Curve) as a performance metric, along with sensitivity and specificity, to assess the model's ability to correctly classify customers.

Given the class imbalance in the dataset, where the number of non-subscribers significantly outweighs the number of subscribers, the ROSE (Random Over-Sampling Examples) method was applied to balance the training set. This method helps in mitigating bias by generating synthetic samples for the minority class, ensuring that the models perform well on both classes.

For both models, the performance metrics of accuracy, sensitivity, specificity, and AUC were compared. The confusion matrix was used to analyze the classification performance of each model, and the ROC (Receiver Operating Characteristic) curve was generated to visualize the trade-off between sensitivity and specificity across different threshold values.

In addition to the core models, we performed additional analysis using interaction plots to identify significant variable interactions that could improve model predictions. Non-linear relationships were explored by creating smoothing splines using a general additive model (GAM) for continuous predictors, which helped in assessing the linearity of the data and identifying any complex relationships between the variables.

The assumptions of both models were tested, including checking for multicollinearity in Logistic Regression and ensuring the assumption of normally distributed predictors in LDA. Any violations of these assumptions were addressed by transforming variables or using more appropriate modeling techniques.

Lastly, the models were evaluated on a balanced test set created using the ROSE method to assess how well each model generalizes to unseen data. Performance on this test set was compared to that of the training set to evaluate overfitting and model robustness.

V. Data

Data cleaning

A. Dataset: UCI Bank Marketing “bank-additional-full.csv”
B. Established binary 1, 0 response variable for desired target (“y” – yes(1)/no(0))
C. Excluded “duration”, as this can only be known after the call, or if a call was answered, so it would overstate performance if kept

D. Confirmed dataset's mix of demographic, contact-history, and macroeconomic features (emp.var.rate, cons.price.idx, etc)

Data preprocessing

A. Feature Engineering: Created two case-specific predictors: age_group bins (17–25, 26–35, 36–45, 46–55, 56–65, 66+) & pdays_group (0–90, 91–180, 181–270, 271–360, 360+, Not_Contacted (pdays=999)

B. Train/Test Split: 70/30 stratified split (caret::createDataPartition()) to preserve "yes/no" output proportion in both subsets

C. Set categorical predictors as factors and numeric features as numeric in both "train" and "test" before resampling

D. Utilized "ROSE" library on training set to generate a synthetically balanced sample (train_bal <- ROSE::ROSE(form, data = train, seed = 42)$data) to learn minority pattern, "yes" in this dataset, without discarding majority pattern, "no" in this dataset. This ensured real-world base rates to start

E. To offset the sparse marketing categories, aligned factor levels across train/test with "forcats" to prevent "new level" errors

F. Expanded training levels to add "Other"

G. Mapped unseen test levels to "Other"

H. Seeded test-only levels back into training to ensure consistency across level sets for modeling

I. Logistic regression directly consumed factors

J. LDA used one-hot dummies (caret::dummyVars), removed near-zero variance and class-constant columns to meet LDA's assumptions

K. Selected operating points from the ROC curve using Youden's J (max Sensitivity + Specificity – 1)

Data limitations

A. Class balance: no: 36548 (88.73%); yes: 4640 (11.27%) – accuracy can be misleading without sensitivity/specificity and AUC

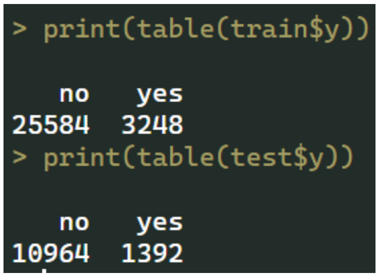
B. Unable to analyst response by gender, as there is no gender indicator included in the dataset

C. pdays: 999 = “Not contacted previously” – defined as categorical, but provided as numeric
D. “ROSE” library created artificial examples from the training distribution so results were determined to need validating with original test distribution with “balanced test” results utilized as sensitivity analyses
E. Excluded “duration” to avoid unrealistic performance indicators

VI. Findings (Results)

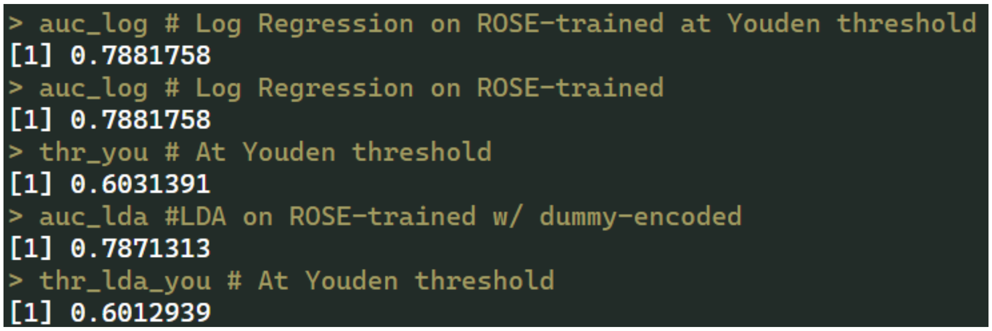
Results

A. Class balance for train and test



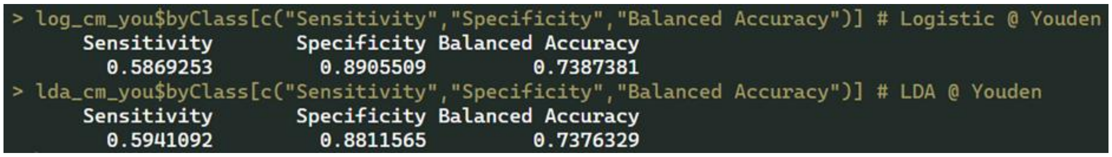
```
> print(table(train$y))

   no   yes
25584  3248
> print(table(test$y))

   no   yes
10964  1392
```

B. Model AUC



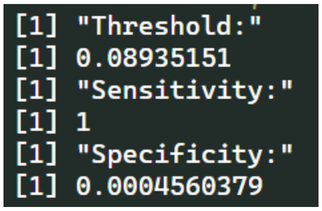
```
> auc_log # Log Regression on ROSE-trained at Youden threshold
[1] 0.7881758
> auc_log # Log Regression on ROSE-trained
[1] 0.7881758
> thr_you # At Youden threshold
[1] 0.6031391
> auc_lda #LDA on ROSE-trained w/ dummy-encoded
[1] 0.7871313
> thr_lda_you # At Youden threshold
[1] 0.6012939
```

C. Youden’s J



```
> log_cm_you$byClass[c("Sensitivity","Specificity","Balanced Accuracy")] # Logistic @ Youden
      Sensitivity       Specificity Balanced Accuracy
        0.5869253         0.8905509         0.7387381
> lda_cm_you$byClass[c("Sensitivity","Specificity","Balanced Accuracy")] # LDA @ Youden
      Sensitivity       Specificity Balanced Accuracy
        0.5941092         0.8811565         0.7376329
```

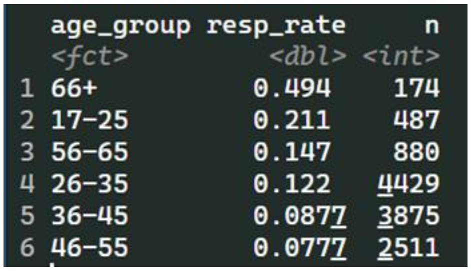
D. Sensitivity-maximizing policy variant



```
[1] "Threshold:"
[1] 0.08935151
[1] "Sensitivity:"
[1] 1
[1] "Specificity:"
[1] 0.0004560379
```

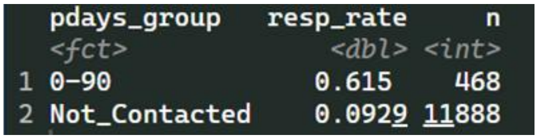
E. Descriptive statistic for who is more likely to respond yes

by Age



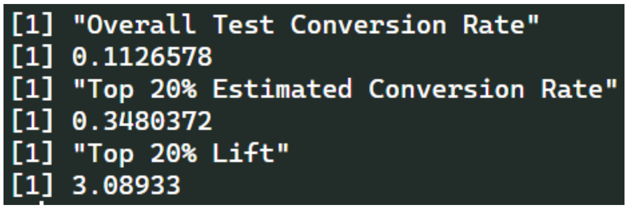
```
  age_group resp_rate     n
  <fct>         <dbl> <int>
1 66+          0.494    174
2 17-25        0.211    487
3 56-65        0.147    880
4 26-35        0.122   4429
5 36-45        0.0877  3875
6 46-55        0.0777  2511
```

By pdays



```
  pdays_group   resp_rate     n
  <fct>             <dbl> <int>
1 0-90             0.615    468
2 Not_Contacted    0.0929 11888
```

F. Lift and gains



```
[1] "Overall Test Conversion Rate"
[1] 0.1126578
[1] "Top 20% Estimated Conversion Rate"
[1] 0.3480372
[1] "Top 20% Lift"
[1] 3.08933
```

The class balance results confirm the dataset’s bias toward “no” responses, thus highlighting the need for resampling and threshold selection. The AUC and confusion matrix measures demonstrate both logistic regression and LDA to perform well above random classification and provide similar levels of discriminative ability. The descriptive statistics by age group and contact history also provides relative variation in response behavior, thus supporting deeper analysis of demographic and campaign history factors impact client decisions.

Results with respect to hypotheses/models

The models provided above support our initial hypothesis: Recent contact history and prior outcomes are strong predictors of client subscription. Logistic regression identified the group that had been contacted most recently (pdays_group = 0-90) as highly influential, as those clients held a response rate above 60% when compared to less than 10% from those who were not previously contacted. Age was also found to be a differentiating factor. Clients who were 66+ had the highest subscription rate at 49%, while those between 36-55 (36-45 and 46-55 bins) had subscription rates of only ~9%, which suggests that both demographics and campaign history are key drivers in predicting "yes" responses.

We found that logistic regression and LDA achieved almost identical AUC values - 0.788 and 0.787, respectively – indicating that responders and non-responders are distinguished the same way between the two models. At Youden's J threshold, both models also balanced sensitivity and specificity – 0.59 and 0.89, respectively. Logistic regression also demonstrated a level of flexibility in threshold tuning when optimizing strictly for sensitivity. It captured nearly all true responders (Ref Results section D – Sensitivity = 1.00) however yielding a low specificity rate (0.0004..). This highlighted the trade-off we've discussed in class between maximizing true positives (yes responses) and maintaining accuracy when rejecting negatives (no responses).

Factual information kept separate from interpretation, inference and evaluation

G. Training set includes 25,584 "no" responses and 3,248 "yes" responses ; test set contained 10,964 "no" responses" and 1,392 "yes" responses
H. AUC
    a. Logistic regression: 0.788
    b. LDA: 0.787
I. Youden's J threshold: ~ 0.60 for both Logistic regression and LDA models
J. Logistic regression at Youden threshold
    a. Sensitivity: 0.587
    b. Specificity: 0.891
    c. Balanced Accuracy: 0.738
K. LDA at Youden threshold

- a. Sensitivity: 0.594
- b. Specificity: 0.882
- c. Balanced Accuracy: 0.738

L. Logistic regression at sensitivity-maximizing threshold

- a. Sensitivity: 1.00
- b. Specificity: ~ 0.0005

M. Response Rate

- a. By age_group
  - i. 66+: 0.494
  - ii. 17-25: 0.211
  - iii. 56-65: 0.147
  - iv. 26-35: 0.122
  - v. 36-45: 0.088
  - vi. 46-55: 0.078
- b. By p_days group
  - i. 0-90 days: 0.615
  - ii. Not_Contacted (999): 0.093

N. Overall test conversion rate: 0.113

O. Top 20%

- a. Estimated conversion rate: 0.348
- b. Lift: 3.09x (relative to random selection)

VII. Conclusions and Recommendations

Alternative Methodologies

This study was mainly focused on logistic regression and LDA with ROSE balancing, but we could've used other approaches as well. Naive Bayes and K-Nearest Neighbors offer simple baselines that could provide useful comparison points. Decision Trees and Random Forests allow for more interpretable segmentation, while ensemble methods like

Boosting typically yield stronger predictive performance. Also, Support Vector Machines provide an alternative to linear classifier with the ability to model nonlinear boundaries through kernels. For handling imbalance, strategies such as SMOTE or cost-sensitive learning were also covered, and these methods provide different ways to address the skewed distribution of the target variable.

Recommendations

Based on the final results, we recommend using logistic regression as the baseline model for deployment due to its interpretability and solid AUC performance. In practice, the bank should prioritize contacting the top 25% of customers ranked by predicted probability just because this segment yields nearly three times the average conversion rate.

VIII. Appendix

---- Packages ----

```
library(tidyverse) library(caret) library(ROSE) library(MASS) library(pROC) library(dplyr)
library(forcats) library(broom)
```

```
set.seed(42)
```

---- Load ----

```
setwd("/Users/achyutparmarthi/Desktop/Data App/week 3/") bank <- read.csv("bank-
additional-full.csv", sep = ";") %>% dplyr::mutate(y = factor(y, levels = c("no","yes"))) %>%
dplyr::select(-duration) # duration excluded per assignment
```

----Data Understanding & Preparation ----

```
cat("Rows:", nrow(bank), " | Cols:", ncol(bank), "\n")
```

```
is_cat <- sapply(bank, is.factor) cats <- names(bank)[is_cat] nums <- names(bank)[!is_cat
& names(bank) != "y"] cat("\nCategorical variables (", length(cats), "):\n", paste(cats,
collapse = ", "), "\n", sep = "") cat("\nNumeric variables (", length(nums), "):\n",
paste(nums, collapse = ", "), "\n", sep = "")
```

```
class_tbl <- table(bank$y) class_prop <- prop.table(class_tbl) cat("\nClass balance:\n");
print(class_tbl); print(round(class_prop, 4))
```

Feature engineering

```
bank <- bank %>% dplyr::mutate( age_group = dplyr::case_when( age <= 25 ~ "17-25", age
<= 35 ~ "26-35", age <= 45 ~ "36-45", age <= 55 ~ "46-55", age <= 65 ~ "56-65", TRUE ~
"66+" ), pdays_group = dplyr::case_when( pdays == 999 ~ "Not_Contacted", pdays <= 90 ~
"0-90", pdays <= 180 ~ "91-180", pdays <= 270 ~ "181-270", pdays <= 360 ~ "271-360",
TRUE ~ "360+" ) )
```

```
cat("\nEngineered variables added: age_group, pdays_group; excluded: duration\n")
```

---- Split ----

```
idx <- caret::createDataPartition(bank$y, p = 0.7, list = FALSE) train <- bank[idx, ] test <-
bank[-idx, ]
```

---- which predictors are categorical vs numeric ----

```
cat_vars <- c("age_group","pdays_group",
"job","marital","education","default","housing","loan",
"contact","month","day_of_week","poutcome") num_vars <- c("campaign","previous",
"emp.var.rate","cons.price.idx","cons.conf.idx","euribor3m","nr.employed")
```

```
cat_vars <- intersect(cat_vars, names(train)) num_vars <- intersect(num_vars,
names(train))
```

```
to_factor <- function(df, cols) { for (c in cols) df[[c]] <- factor(as.character(df[[c]])) df }
to_numeric <- function(df, cols) { for (c in cols) df[[c]] <-
suppressWarnings(as.numeric(df[[c]])) df } train <- to_factor(train, cat_vars); test <-
to_factor(test, cat_vars) train <- to_numeric(train, num_vars); test <- to_numeric(test,
num_vars)
```

Aqdas Hussain
Amberly Cavazos
Sarai Saenz
Jonathan McCanlas
Achyut Parmarthi

---- Modeling ----

```
predictors <- c(cat_vars, num_vars) form <- as.formula(paste("y ~", paste(predictors,
collapse = " + ")))
```

```
train_bal <- ROSE::ROSE(form, data = train, seed = 42)$data
```

```
train_bal <- to_factor(train_bal, cat_vars) train_bal <- to_numeric(train_bal, num_vars)
```

```
align_levels <- function(train_df, test_df, cols) { for (col in cols) { # ensure "Other" is a
possible training level train_df[[col]] <- forcats::fct_expand(train_df[[col]], "Other")
keep_lvls <- levels(train_df[[col]])
```

```
# map test unseen -> "Other", coerce to same levels/order
test_df[[col]] <- forcats::fct_other(test_df[[col]], keep = keep_lvls, other_level = "Other")
test_df[[col]] <- factor(test_df[[col]], levels = keep_lvls)
```

```
} list(train = train_df, test = test_df) } tmp <- align_levels(train_bal, test, cat_vars) train_bal
<- tmp$train test <- tmp$test rm(tmp)
```

```
seed_missing_levels <- function(train_df, test_df, cols) { for (col in cols) { if
(!is.factor(train_df[[col]]) || !is.factor(test_df[[col]])) next # levels present in test, absent in
training counts test_lvls_used <- names(which(table(test_df[[col]]) > 0)) for (lv in
test_lvls_used) { if (!(lv %in% levels(train_df[[col]]))) next # skip truly unseen in both;
shouldn't happen # if level exists in levels but zero rows have it, seed one row if
(sum(train_df[[col]] == lv, na.rm = TRUE) == 0) { idx_any <- which(!is.na(train_df[[col]]))[1] if
(length(idx_any) == 1 && !is.na(idx_any)) { train_df[[col]][idx_any] <- lv } } } } train_df }
train_bal <- seed_missing_levels(train_bal, test, cat_vars)
```

---- Model 1: LOGISTIC ----

```
log_model <- glm(form, data = train_bal, family = binomial) log_probs <- predict(log_model,
newdata = test, type = "response")
```

```
ok_log <- !is.na(log_probs) & !is.na(test$y) log_probs_ok <- log_probs[ok_log] y_ok <-
test$y[ok_log]
```

```
roc_log <- pROC::roc(y_ok, log_probs_ok, levels = c("no","yes"), quiet = TRUE) auc_log <-
as.numeric(pROC::auc(roc_log)) thr_you <- as.numeric(pROC::coords(roc_log, "best",
best.method = "youden", ret = "threshold")) if (length(thr_you) > 1) thr_you <-
mean(thr_you)
```

```
log_pred_you <- factor(ifelse(log_probs_ok >= thr_you, "yes", "no"), levels = c("no","yes"))
log_cm_you <- caret::confusionMatrix(log_pred_you, y_ok, positive = "yes")
```

```
coords_all <- as.data.frame(pROC::coords( roc_log, x = "all", ret =
c("threshold","sensitivity","specificity"), transpose = FALSE ))
```

```
best_sens <- max(coords_all$sensitivity, na.rm = TRUE) cand <-
coords_all[coords_all$sensitivity == best_sens, , drop = FALSE] thr_sens <-
cand$threshold[which.max(cand$specificity)]
```

```
log_pred_sens <- factor(ifelse(log_probs_ok >= thr_sens, "yes", "no"), levels = c("no","yes"))
log_cm_sens <- caret::confusionMatrix(log_pred_sens, y_ok, positive = "yes")
```

```
cat("\nLogistic tuned for sensitivity (max sensitivity):\n") cat(sprintf("Chosen Thr: %.3f |
Sens: %.3f | Spec: %.3f\n", thr_sens, cand$sensitivity[which.max(cand$specificity)],
cand$specificity[which.max(cand$specificity)]))
print(log_cm_sens$byClass[c("Sensitivity","Specificity","Balanced Accuracy")])
```

---- Model 2: LDA ----

```
dv <- caret::dummyVars(~ . - y, data = train_bal) # dummy predictors only X_train <-
as.data.frame(predict(dv, newdata = train_bal)) X_test <- as.data.frame(predict(dv,
newdata = test))
```

```
nzv_idx <- caret::nearZeroVar(X_train) if (length(nzv_idx) > 0) { X_train <- X_train[ , -nzv_idx,
drop = FALSE] X_test <- X_test[ , -nzv_idx, drop = FALSE] }
```

```
const_within_class <- sapply(colnames(X_train), function(cn) { any(tapply(X_train[[cn]],
train_bal$y, function(v) length(unique(v)) <= 1)) }) if (any(const_within_class)) { X_train <-
X_train[ , !const_within_class, drop = FALSE] X_test <- X_test[ , !const_within_class, drop =
FALSE] }
```

```
row_ok_train <- stats::complete.cases(X_train) X_train2 <- X_train[row_ok_train, , drop = FALSE] y_train2 <- droplevels(train_bal$y[row_ok_train])
```

```
row_ok_test <- stats::complete.cases(X_test) & !is.na(test$y) X_test2 <- X_test[row_ok_test, , drop = FALSE] y_test2 <- droplevels(test$y[row_ok_test])
```

```
lda_model <- MASS::lda(x = X_train2, grouping = y_train2) lda_out <- predict(lda_model, newdata = X_test2) lda_probs <- lda_out$posterior[, "yes"]
```

```
roc_lda <- pROC::roc(y_test2, lda_probs, levels = c("no","yes"), quiet = TRUE) auc_lda <- as.numeric(pROC::auc(roc_lda)) thr_lda_you <- as.numeric(pROC::coords(roc_lda, "best", best.method = "youden", ret = "threshold")) if (length(thr_lda_you) > 1) thr_lda_you <- mean(thr_lda_you)
```

```
lda_pred_you <- factor(ifelse(lda_probs >= thr_lda_you, "yes", "no"), levels = c("no","yes")) lda_cm_you <- caret::confusionMatrix(lda_pred_you, y_test2, positive = "yes")
```

---- Comparison and Evaluate ----

```
cat(sprintf("Logistic AUC: %.4f | Youden Thr: %.3f\n", auc_log, thr_you)) print(log_cm_you$byClass[c("Sensitivity","Specificity","Balanced Accuracy")])
```

```
achieved_sens <- cand$sensitivity[which.max(cand$specificity)] achieved_spec <- cand$specificity[which.max(cand$specificity)]
```

```
cat("\nLogistic tuned for sensitivity (max sensitivity):\n") cat(sprintf("Chosen Thr: %.3f | Sens: %.3f | Spec: %.3f\n", thr_sens, achieved_sens, achieved_spec)) print(log_cm_sens$byClass[c("Sensitivity","Specificity","Balanced Accuracy")])
```

```
cat(sprintf("\nLDA AUC: %.4f | Youden Thr: %.3f\n", auc_lda, thr_lda_you)) print(lda_cm_you$byClass[c("Sensitivity","Specificity","Balanced Accuracy")])
```

```
test_bal <- ROSE::ROSE(y ~ ., data = test, seed = 42)$data
```

```
log_probs_bal <- predict(log_model, newdata = test_bal, type = "response") roc_log_bal <- pROC::roc(test_bal$y, log_probs_bal, levels = c("no","yes"), quiet = TRUE) auc_log_bal <- as.numeric(pROC::auc(roc_log_bal)) thr_bal <- as.numeric(pROC::coords(roc_log_bal,
"best", best.method = "youden", ret = "threshold")) if (length(thr_bal) > 1) thr_bal <- mean(thr_bal)

log_pred_bal <- factor(ifelse(log_probs_bal >= thr_bal, "yes", "no"), levels = c("no","yes")) log_cm_bal <- caret::confusionMatrix(log_pred_bal, test_bal$y, positive = "yes")

cat(sprintf("AUC: %.4f | Youden Thr: %.3f\n", auc_log_bal, thr_bal)) print(log_cm_bal$byClass[c("Sensitivity","Specificity","Balanced Accuracy")])

X_test_bal <- as.data.frame(predict(dv, newdata = test_bal))

keep_cols <- colnames(X_train2)

missing_cols <- setdiff(keep_cols, colnames(X_test_bal)) for (m in missing_cols) X_test_bal[[m]] <- 0 X_test_bal <- X_test_bal[, keep_cols, drop = FALSE]

extra_cols <- setdiff(colnames(X_test_bal), keep_cols) if (length(extra_cols) > 0) X_test_bal <- X_test_bal[, keep_cols, drop = FALSE]

row_ok_test_bal <- stats::complete.cases(X_test_bal) & !is.na(test_bal$y) X_test_bal2 <- X_test_bal[row_ok_test_bal, , drop = FALSE] y_test_bal2 <- droplevels(test_bal$y[row_ok_test_bal])

lda_probs_bal <- predict(lda_model, newdata = X_test_bal2)$posterior[, "yes"]

roc_lda_bal <- pROC::roc(y_test_bal2, lda_probs_bal, levels = c("no","yes"), quiet = TRUE) auc_lda_bal <- as.numeric(pROC::auc(roc_lda_bal)) thr_lda_bal <- as.numeric(pROC::coords(roc_lda_bal, "best", best.method = "youden", ret = "threshold")) if (length(thr_lda_bal) > 1) thr_lda_bal <- mean(thr_lda_bal)

lda_pred_bal <- factor(ifelse(lda_probs_bal >= thr_lda_bal, "yes", "no"), levels = c("no","yes")) lda_cm_bal <- caret::confusionMatrix(lda_pred_bal, y_test_bal2, positive = "yes")

cat(sprintf("AUC: %.4f | Youden Thr: %.3f\n", auc_lda_bal, thr_lda_bal)) print(lda_cm_bal$byClass[c("Sensitivity","Specificity","Balanced Accuracy")])
```

```
log_tidy <- broom::tidy(log_model) %>% dplyr::mutate(OR = exp(estimate), abs_z =
abs(statistic)) %>% dplyr::arrange(desc(abs_z))

print(log_tidy %>% dplyr::slice(1:10) %>% dplyr::select(term, estimate, OR, statistic,
p.value))

top_pos <- log_tidy %>% dplyr::filter(estimate > 0) %>% dplyr::slice_max(order_by =
estimate, n = 5) %>% dplyr::pull(term) cat("\nPersonas likely to respond (heuristic from
positive terms):\n"); print(top_pos) cat("\nNotes:\n- Positive estimate (OR>1) increases
odds of subscription; negative decreases.\n- Factor terms interpret relative to baseline
level.\n")

lift_df <- tibble(prob = log_probs_ok, actual = as.integer(y_ok == "yes")) %>%
dplyr::arrange(dplyr::desc(prob)) %>% dplyr::mutate(rank = row_number(), frac = rank / n(),
decile = ntile(-prob, 10)) # 1 = highest scores

overall_rate <- mean(lift_df$actual) cat(sprintf("Overall conversion rate in test: %.4f\n",
overall_rate))

top20 <- lift_df %>% dplyr::filter(frac <= 0.20) top20_rate <- mean(top20$actual) lift_top20
<- top20_rate / overall_rate cat(sprintf("If you contact top 20%%: expected conversion rate
= %.4f | Lift = %.2fx over random.\n", top20_rate, lift_top20))

decile_lift <- lift_df %>% dplyr::group_by(decile) %>% dplyr::summarise(n = dplyr::n(),
resp_rate = mean(actual), cum_n = cumsum(n), cum_resp =
cumsum(n*resp_rate), .groups = "drop") %>% dplyr::mutate(cum_frac = cum_n / sum(n),
lift = resp_rate / overall_rate, cum_resp_rate = cum_resp / cum_n)

cat("\nDecile lift table (top=1 is highest score):\n"); print(decile_lift)

cat("\nEthical considerations:\n") cat("- Audit for disparate impact across age, job,
education.\n") cat("- Avoid excluding protected groups based on historical bias.\n") cat("-
Consider fairness constraints or post-model reviews before deployment.\n")

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

Aqdas Hussain
Amberly Cavazos
Sarai Saenz
Jonathan McCanlas
Achyut Parmarthi

====Plots / Visuals ====

```
library(ggplot2)
```

```
roc_df_log <- data.frame( fpr = 1 - roc_log$specificities, tpr = roc_log$sensitivities, model = "Logistic" ) roc_df_lda <- data.frame( fpr = 1 - roc_lda$specificities, tpr = roc_lda$sensitivities, model = "LDA" ) roc_df_both <- rbind(roc_df_log, roc_df_lda)
```

```
ggplot(roc_df_both, aes(x = fpr, y = tpr, color = model)) + geom_line(linewidth = 1) + geom_abline(slope = 1, intercept = 0, linetype = "dotted") + labs(title = sprintf("ROC Curves (AUC: Logistic=%.3f, LDA=%.3f)", auc_log, auc_lda), x = "False Positive Rate (1 - Specificity)", y = "True Positive Rate (Sensitivity)") + theme_minimal()
```

```
coords_log <- as.data.frame(pROC::coords( roc_log, x = "all", ret = c("threshold","sensitivity","specificity"), transpose = FALSE )) ggplot(coords_log, aes(x = threshold)) + geom_line(aes(y = sensitivity, linetype = "Sensitivity"), linewidth = 1) + geom_line(aes(y = specificity, linetype = "Specificity"), linewidth = 1) + geom_vline(xintercept = thr_you, linetype = "dashed") + annotate("text", x = thr_you, y = 0.03, label = "Youden thr", angle = 90, vjust = -0.3, size = 3) + { if (exists("thr_target")) geom_vline(xintercept = thr_target, linetype = "dashed", alpha = 0.7) } + labs(title = "Sensitivity & Specificity vs Threshold (Logistic)", x = "Threshold", y = "Rate") + theme_minimal() + scale_linetype_manual(values = c("Sensitivity" = "solid", "Specificity" = "dotdash"))
```

```
coords_log$accuracy <- sapply(coords_log$threshold, function(t) { pred <- factor(ifelse(log_probs_ok >= t, "yes", "no"), levels = c("no","yes")) mean(pred == y_ok) }) ggplot(coords_log, aes(x = threshold, y = accuracy)) + geom_line(linewidth = 1) + geom_vline(xintercept = thr_you, linetype = "dashed") + labs(title = "Accuracy vs Threshold (Logistic)", x = "Threshold", y = "Accuracy") + theme_minimal()
```

```
cm_mat <- as.matrix(log_cm_you$table) # rows = Prediction, cols = Reference cm_df <- as.data.frame(as.table(cm_mat)) colnames(cm_df) <- c("Prediction", "Reference", "Freq") ggplot(cm_df, aes(x = Reference, y = Prediction, fill = Freq)) + geom_tile() + geom_text(aes(label = Freq), color = "white", fontface = "bold") + scale_fill_gradient(low = "#6baed6", high = "#08519c") + labs(title = "Confusion Matrix (Logistic @ Youden threshold)") + theme_minimal()
```

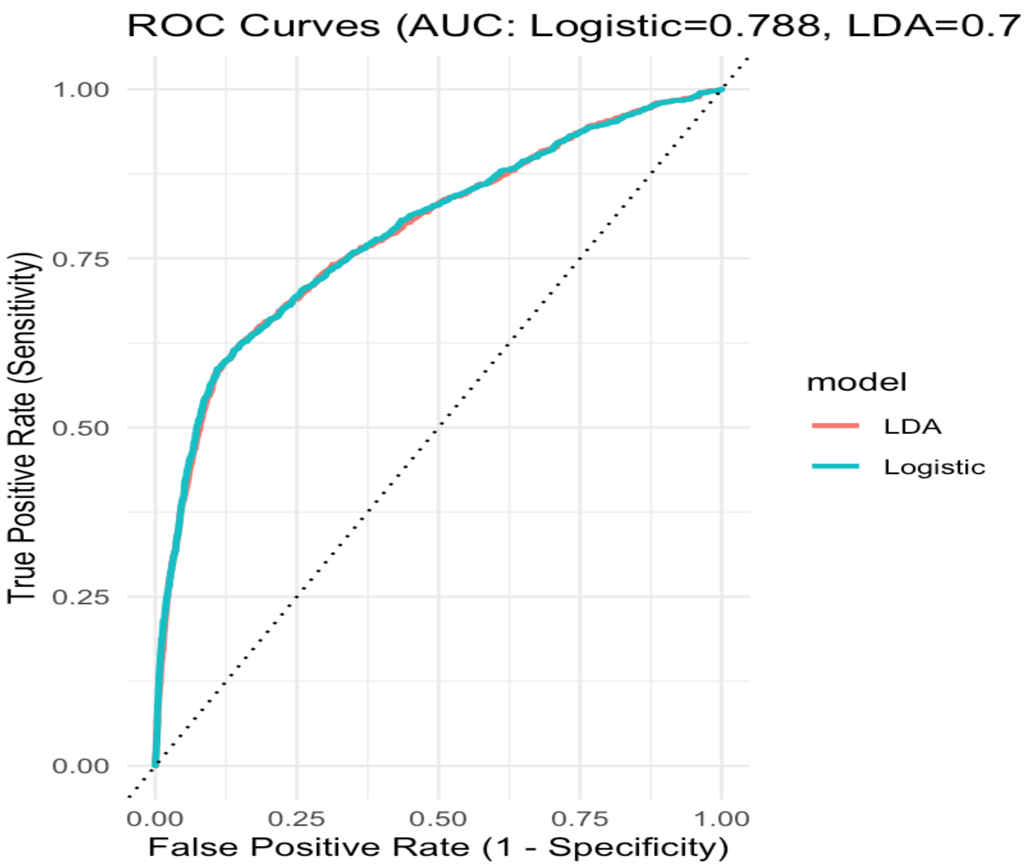
Aqdas Hussain
Amberly Cavazos
Sarai Saenz
Jonathan McCanlas
Achyut Parmarthi

```
ggplot(decile_lift, aes(x = factor(decile), y = resp_rate)) + geom_col() +
geom_hline(yintercept = overall_rate, linetype = "dashed") + labs(title = "Response Rate by
Score Decile (1 = highest scores)", x = "Decile", y = "Response Rate") + theme_minimal()
```

```
gains_df <- lift_df %>% arrange(desc(prob)) %>% mutate(cum_positives =
cumsum(actual), total_positives = sum(actual), cum_frac_pop = row_number()/n(),
cum_frac_pos = cum_positives/total_positives)
```

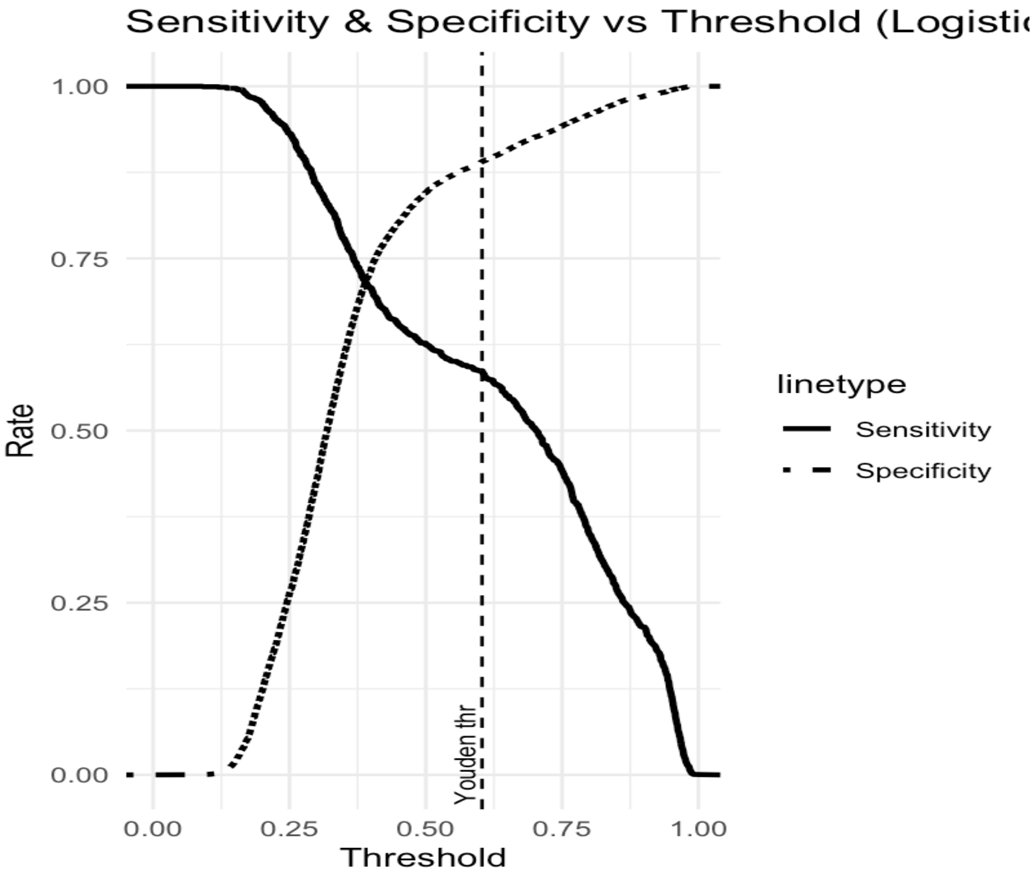
```
ggplot(gains_df, aes(x = cum_frac_pop, y = cum_frac_pos)) + geom_line(linewidth = 1) +
geom_abline(slope = 1, intercept = 0, linetype = "dotted") + labs(title = "Cumulative Gains
Curve (Logistic)", x = "Fraction of Population Contacted", y = "Fraction of Positive
Responses Captured") + theme_minimal()
```

Aqdas Hussain
Amberly Cavazos
Sarai Saenz
Jonathan McCanlas
Achyut Parmarthi

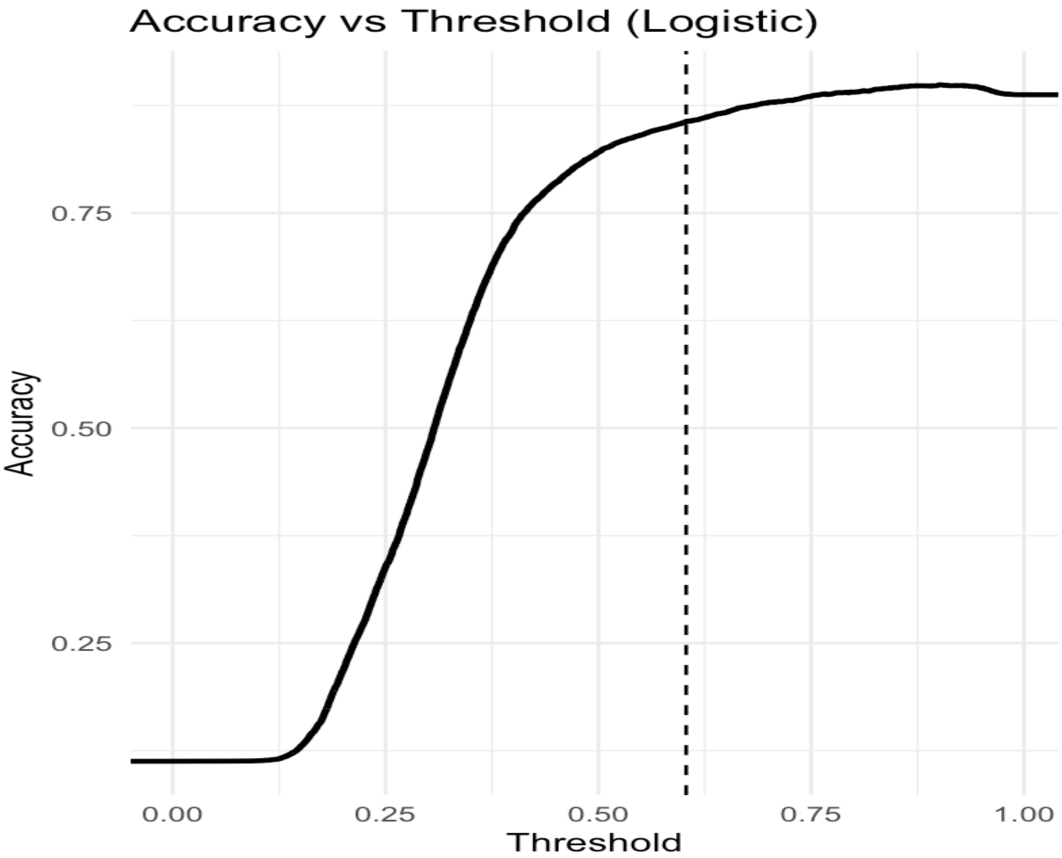


ROC Curves (AUC: Logistic=0.788, LDA=0.7

Aqdas Hussain
Amberly Cavazos
Sarai Saenz
Jonathan McCanlas
Achyut Parmarthi



Aqdas Hussain
Amberly Cavazos
Sarai Saenz
Jonathan McCanlas
Achyut Parmarthi


Accuracy vs Threshold (Logistic)
Accuracy
Threshold