Classwork 3: Model Evaluation and Business Impact Analysis

Introduction to Predictive Analytics in R

Prof. Asc. Endri Raco, PhD

Polytechnic University of Tirana

November 2025

Section 1

Classwork Overview

Today's Mission: From Models to Money

Evaluate Models Using Business Metrics

Scenario: The bank's marketing director wants answers to:

1. "If we contact the top 30% of customers, how many will subscribe?"
2. "What's the expected profit from this campaign?"
3. "How much better is your model than random selection?"

Your Task: Use cumulative gains and lift curves to provide data-driven answers.

Learning Objectives

By completing this classwork, you will:

1. Create cumulative gains curves from scratch
2. Generate and interpret lift curves
3. Calculate expected profit at different contact levels
4. Identify optimal campaign strategy
5. Compare multiple models visually
6. Prepare executive-ready recommendations

Dataset: Bank Marketing Campaign

Target Variable: `subscribed` (1 = yes, 0 = no)

Your Predictions: Already generated from Classwork 2

Campaign Economics:

- Revenue per subscription: €100
- Cost per contact: €5
- Database size: 5,000 customers
- Baseline subscription rate: 11.7%

Deliverables

What You'll Submit

1. R script with all code (`classwork3.R`) 2. Report answering 8 key questions 3. Four required visualizations:
   - Cumulative gains curve
   - Lift curve
   - Profit optimization curve
   - Model comparison plot
4. Executive summary with recommendations

Time Allocation: 40 minutes

Getting Started

Setup

Create `classwork3.R` and load necessary libraries

```r
# Load libraries
library(tidyverse)
library(pROC)

# Set seed
set.seed(789)

# Load your data from Classwork 2
# (assumes you have test set with predictions)
load("classwork2_results.RData")

# Or recreate if needed
# test_data should have:
# - actual values (subscribed)
# - predicted probabilities (pred_prob)
```

Section 2

Part 1: Cumulative Gains Curve (12 minutes)

Task 1.1: Understanding Your Data

```r
# Examine your test data
glimpse(test_data)

# Check we have what we need
cat("Sample size:", nrow(test_data), "\n")
cat("Actual subscription rate:",
    round(mean(test_data$subscribed) * 100, 1),
    "%\n")

# View distribution of predictions
summary(test_data$pred_prob)
```

Question 1: What is the actual subscription rate in your test set?

Task 1.2: Create Gains Calculation Function

```r
# Function to calculate gains curve data
calculate_gains <- function(actual, predicted) {

  # Create dataframe and sort by prediction
  df <- data.frame(
    actual = actual,
    predicted = predicted
  ) %>%
    arrange(desc(predicted))

  # Calculate cumulative percentages
  df <- df %>%
    mutate(
      perc_sample = row_number() / n(),
      cum_targets = cumsum(actual),
      total_targets = sum(actual),
      perc_targets = cum_targets / total_targets
    )

  return(df)
```

Task 1.3: Generate Gains Data

```r
# Calculate gains curve
gains_data <- calculate_gains(
  actual = test_data$subscribed,
  predicted = test_data$pred_prob
)

# View first few rows
head(gains_data)

# Check key points
gains_at_30 <- gains_data %>%
  filter(perc_sample >= 0.30) %>%
  slice_head(n = 1) %>%
  pull(perc_targets)

cat("At 30% of sample, we capture",
    round(gains_at_30 * 100, 1),
    "% of targets\n")
```

Task 1.4: Plot Cumulative Gains Curve

```r
# Create the plot
gains_plot <- ggplot(gains_data,
                     aes(x = perc_sample,
                         y = perc_targets)) +
  geom_line(color = "darkgreen",
            size = 1.5) +
  geom_abline(slope = 1,
              intercept = 0,
              linetype = "dashed",
              color = "black") +
  labs(
    title = "Cumulative Gains Curve",
    x = "Percentage of Sample Contacted",
    y = "Percentage of Targets Reached"
  ) +
  scale_x_continuous(labels = scales::percent) +
  scale_y_continuous(labels = scales::percent) +
  theme_minimal(base_size = 12)

print(gains_plot)
```

Task 1.5: Interpret Key Points

```r
# Calculate gains at key percentiles
key_points <- c(0.10, 0.20, 0.30, 0.40, 0.50)

gains_summary <- data.frame(
  contact_perc = key_points,
  targets_reached = NA
)

for (i in 1:length(key_points)) {
  gains_summary$targets_reached[i] <-
    gains_data %>%
    filter(perc_sample >= key_points[i]) %>%
    slice_head(n = 1) %>%
    pull(perc_targets)
}

print(gains_summary)
```

Question 2: If you contact the top 20% of customers, what % of potential

Task 1.6: Add Annotations to Plot

```r
# Enhanced plot with annotation
gains_plot_annotated <- gains_plot +
  geom_point(data = gains_data %>%
               filter(perc_sample >= 0.30) %>%
               slice_head(n = 1),
             aes(x = perc_sample, y = perc_targets),
             color = "red", size = 4) +
  annotate("text",
           x = 0.35,
           y = gains_at_30 - 0.05,
           label = paste0("30% contacts\n",
                          round(gains_at_30*100, 1),
                          "% targets"),
           size = 3.5)

print(gains_plot_annotated)
```

Section 3

Part 2: Lift Curve Analysis (10 minutes)

Task 2.1: Create Lift Calculation Function

```r
# Function to calculate lift
calculate_lift <- function(actual, predicted) {

  # Baseline rate
  baseline_rate <- mean(actual)

  # Sort by prediction
  df <- data.frame(
    actual = actual,
    predicted = predicted
  ) %>%
    arrange(desc(predicted))

  # Calculate lift at each point
  df <- df %>%
    mutate(
      perc_sample = row_number() / n(),
      cum_targets = cumsum(actual),
      cum_sample = row_number(),
      lift = (cum_targets / cum_sample) / baseline_rate
```

Task 2.2: Generate Lift Data

```r
# Calculate lift curve
lift_data <- calculate_lift(
  actual = test_data$subscribed,
  predicted = test_data$pred_prob
)

# Maximum lift (at beginning)
max_lift <- max(lift_data$lift)
cat("Maximum lift:", round(max_lift, 2), "\n")

# Lift at key percentiles
lift_at_20 <- lift_data %>%
  filter(perc_sample >= 0.20) %>%
  slice_head(n = 1) %>%
  pull(lift)

cat("Lift at 20%:", round(lift_at_20, 2), "\n")
```

Question 3: What is the maximum lift value? What does this mean?

Task 2.3: Plot Lift Curve

```r
# Create lift plot
lift_plot <- ggplot(lift_data,
                    aes(x = perc_sample,
                        y = lift)) +
  geom_line(color = "darkblue",
            size = 1.5) +
  geom_hline(yintercept = 1,
             linetype = "dashed",
             color = "black") +
  labs(
    title = "Lift Curve",
    subtitle = "How much better than random?",
    x = "Percentage of Sample Contacted",
    y = "Lift"
  ) +
  scale_x_continuous(labels = scales::percent) +
  theme_minimal(base_size = 12)

print(lift_plot)
```

Task 2.4: Interpret Lift Values

```r
# Create lift summary table
lift_summary <- lift_data %>%
  filter(perc_sample %in%
           c(0.1, 0.2, 0.3, 0.4, 0.5)) %>%
  select(perc_sample, lift) %>%
  mutate(
    interpretation = case_when(
      lift >= 3 ~ "Excellent targeting",
      lift >= 2 ~ "Good targeting",
      lift >= 1.5 ~ "Moderate targeting",
      TRUE ~ "Weak targeting"
    )
  )

print(lift_summary)
```

Question 4: At what contact percentage does lift drop below 2?

Section 4

Part 3: Profit Analysis (12 minutes)

Task 3.1: Define Profit Function

```r
# Campaign economics
REVENUE_PER_SUBSCRIPTION <- 100  # euros
COST_PER_CONTACT <- 5            # euros
DATABASE_SIZE <- nrow(test_data)
BASELINE_RATE <- mean(test_data$subscribed)

# Profit calculation function
calculate_profit <- function(perc_contacted,
                             lift_value,
                             revenue = REVENUE_PER_SUBSCRIPTION,
                             cost = COST_PER_CONTACT,
                             pop_size = DATABASE_SIZE,
                             baseline = BASELINE_RATE) {

  n_contacted <- pop_size * perc_contacted
  expected_rate <- baseline * lift_value
  n_subscribers <- n_contacted * expected_rate

  total_revenue <- n_subscribers * revenue
  total_cost <- n_contacted * cost
```

Task 3.2: Test Profit Calculations

```r
# Scenario 1: Contact top 20% with your model
lift_20 <- lift_data %>%
  filter(perc_sample >= 0.20) %>%
  slice_head(n = 1) %>%
  pull(lift)

profit_20 <- calculate_profit(
  perc_contacted = 0.20,
  lift_value = lift_20
)

cat("Profit (top 20%):",
    scales::dollar(profit_20, prefix = "€"), "\n")

# Scenario 2: Contact everyone randomly
profit_all <- calculate_profit(
  perc_contacted = 1.0,
  lift_value = 1.0
)
```

Task 3.3: Calculate Profit at All Levels

```r
# For each percentage, calculate profit
profit_analysis <- lift_data %>%
  select(perc_sample, lift) %>%
  filter(perc_sample >= 0.05) %>%
  mutate(
    profit = map2_dbl(perc_sample, lift,
                      ~calculate_profit(.x, .y))
  )

# Find optimal contact percentage
optimal <- profit_analysis %>%
  filter(profit == max(profit))

cat("Optimal contact %:",
    round(optimal$perc_sample * 100, 1), "%\n")
cat("Maximum profit:",
    scales::dollar(optimal$profit, prefix = "€"),
    "\n")
```

Question 5: What is the optimal contact percentage?

Task 3.4: Visualize Profit Curve

```
# Create profit optimization plot
profit_plot <- ggplot(profit_analysis,
                      aes(x = perc_sample,
                          y = profit)) +
  geom_line(color = "darkgreen",
            size = 1.5) +
  geom_point(data = optimal,
             color = "red",
             size = 5) +
  geom_hline(yintercept = 0,
             linetype = "dashed",
             color = "black") +
  labs(
    title = "Profit Optimization",
    subtitle = "Finding the sweet spot",
    x = "Percentage of Database Contacted",
    y = "Expected Profit (€)"
  ) +
  scale_x_continuous(labels = scales::percent) +
  scale_y_continuous(labels = scales::dollar_format(
```

Task 3.5: Create Profit Comparison Table

```r
# Compare different strategies
strategies <- data.frame(
  strategy = c("Random (contact all)",
               "Model (top 20%)",
               "Model (top 30%)",
               "Model (optimal)"),
  contact_perc = c(1.0, 0.20, 0.30,
                   optimal$perc_sample),
  lift = c(1.0, lift_20, NA, optimal$lift)
)

# Fill in missing lift
strategies$lift[3] <- lift_data %>%
  filter(perc_sample >= 0.30) %>%
  slice_head(n = 1) %>%
  pull(lift)

# Calculate profits
strategies$profit <- map2_dbl(
  strategies$contact_perc,
```

Task 3.6: Analyze Break-Even Point

```r
# Find break-even contact percentage
# (where profit = 0)
breakeven <- profit_analysis %>%
  filter(profit >= 0) %>%
  slice_tail(n = 1)

cat("Break-even at:",
    round(breakeven$perc_sample * 100, 1),
    "% contacts\n")

# Add to profit plot
profit_plot_enhanced <- profit_plot +
  geom_vline(xintercept = breakeven$perc_sample,
             linetype = "dotted",
             color = "orange") +
  annotate("text",
           x = breakeven$perc_sample + 0.05,
           y = max(profit_analysis$profit) / 2,
           label = "Break-even",
           angle = 90,
```

Section 5

Part 4: Model Comparison (6 minutes)

Task 4.1: Compare with Baseline Model

```r
# Create a simple baseline model (just use age)
baseline_model <- glm(
  subscribed ~ age,
  data = train_data,
  family = binomial
)

# Get predictions on test set
baseline_preds <- predict(baseline_model,
                          newdata = test_data,
                          type = "response")

# Calculate gains for baseline
gains_baseline <- calculate_gains(
  actual = test_data$subscribed,
  predicted = baseline_preds
)

# Add model identifier
gains_data$model <- "Full Model"
```

Task 4.2: Combined Gains Plot

```r
# Combine both models
gains_combined <- rbind(
  gains_data %>%
    select(perc_sample, perc_targets, model),
  gains_baseline %>%
    select(perc_sample, perc_targets, model)
)

# Plot comparison
comparison_plot <- ggplot(gains_combined,
                          aes(x = perc_sample,
                              y = perc_targets,
                              color = model)) +
  geom_line(size = 1.2) +
  geom_abline(slope = 1, intercept = 0,
              linetype = "dashed",
              color = "gray") +
  scale_color_manual(
    values = c("Full Model" = "darkgreen",
               "Baseline (age only)" = "blue")
  )
```

Task 4.3: Compare AUC vs Lift

```r
# Calculate AUC for both models
library(pROC)

auc_full <- auc(roc(test_data$subscribed,
                    test_data$pred_prob))
auc_baseline <- auc(roc(test_data$subscribed,
                        baseline_preds))

# Calculate lift at 20% for both
lift_full_20 <- lift_data %>%
  filter(perc_sample >= 0.20) %>%
  slice_head(n = 1) %>%
  pull(lift)

lift_baseline <- calculate_lift(
  test_data$subscribed,
  baseline_preds
)

lift_baseline_20 <- lift_baseline %>%
```

Task 4.4: Model Performance Summary

```r
# Create comparison table
model_comparison <- data.frame(
  Model = c("Full Model", "Baseline (age)"),
  AUC = c(auc_full, auc_baseline),
  Lift_at_20pct = c(lift_full_20,
                    lift_baseline_20),
  Profit_at_20pct = c(
    calculate_profit(0.20, lift_full_20),
    calculate_profit(0.20, lift_baseline_20)
  )
) %>%
  mutate(
    AUC = round(AUC, 3),
    Lift_at_20pct = round(Lift_at_20pct, 2),
    Profit_at_20pct = scales::dollar(
      Profit_at_20pct, prefix = "€")
  )

print(model_comparison)
```

Section 6

Part 5: Executive Summary (5 minutes)

Task 5.1: Generate Campaign Recommendation

```
# Create executive summary function
create_executive_summary <- function(optimal_data,
                                     strategies_data,
                                     model_comparison) {

  cat("="
      %>% rep(50) %>% paste(collapse = ""), "\n")
  cat("BANK MARKETING CAMPAIGN RECOMMENDATION\n")
  cat("By: Prof. Asc. Endri Raco, PhD\n")
  cat("="
      %>% rep(50) %>% paste(collapse = ""), "\n\n")

  cat("OPTIMAL STRATEGY:\n")
  cat("- Contact:",
      round(optimal_data$perc_sample * 100, 1),
      "% of database\n")

  # Continue on next slide...
}
```

Task 5.2: Complete Executive Summary

```
create_executive_summary <- function(optimal_data,
                                     strategies_data,
                                     model_comparison) {
  # ... header code ...

  cat("- Expected subscribers:",
      round(optimal_data$perc_sample *
            DATABASE_SIZE *
            optimal_data$lift *
            BASELINE_RATE), "\n")
  cat("- Expected profit:",
      scales::dollar(optimal_data$profit,
                     prefix = "€"), "\n\n")

  cat("COMPARISON TO ALTERNATIVES:\n")
  random_profit <- strategies_data %>%
    filter(strategy == "Random (contact all)") %>%
    pull(profit)

  improvement <- optimal_data$profit - random_profit
```

Task 5.3: Executive Summary (Final)

```r
create_executive_summary <- function(...) {
  # ... previous code ...

  cat("- Random selection profit:",
      scales::dollar(random_profit, prefix = "€"),
      "\n")
  cat("- Additional profit from model:",
      scales::dollar(improvement, prefix = "€"),
      "\n")
  cat("- ROI improvement:",
      round((improvement/abs(random_profit))*100, 1),
      "%\n\n")

  cat("KEY INSIGHTS:\n")
  cat("1. Model provides",
      round(optimal_data$lift, 1),
      "x lift at optimal point\n")
  cat("2. Contact",
      round((1-optimal_data$perc_sample)*100, 1),
      "% fewer customers\n")
```

Task 5.4: Create Visual Summary Dashboard

```r
library(gridExtra)

# Combine key plots
dashboard <- grid.arrange(
  gains_plot_annotated,
  lift_plot,
  profit_plot_enhanced,
  comparison_plot,
  ncol = 2,
  top = "Campaign Analysis Dashboard"
)

# Save dashboard
ggsave("campaign_dashboard.pdf",
       dashboard,
       width = 12, height = 10)
```

Task 5.5: Document Your Findings

```r
# Create findings document
sink("campaign_findings.txt")

cat("CAMPAIGN ANALYSIS FINDINGS\n")
cat("Date:", format(Sys.Date(), "%B %d, %Y"), "\n")
cat("Analyst: Prof. Asc. Endri Raco, PhD\n\n")

cat("QUESTION 1: Actual subscription rate\n")
cat("Answer:",
    round(mean(test_data$subscribed)*100, 2),
    "%\n\n")

# Continue with all 8 questions...

sink()  # Close file
```

Question 8: Based on all analysis, would you recommend using the model? Why?

Section 7

Bonus Challenges

Bonus Challenge 1: Decile Analysis

```r
# Divide into 10 equal groups (deciles)
decile_analysis <- test_data %>%
  mutate(
    decile = ntile(desc(pred_prob), 10)
  ) %>%
  group_by(decile) %>%
  summarise(
    n_customers = n(),
    n_subscribers = sum(subscribed),
    subscription_rate = mean(subscribed),
    lift = subscription_rate / BASELINE_RATE
  )

# Visualize decile performance
ggplot(decile_analysis,
       aes(x = factor(decile), y = lift)) +
  geom_col(fill = "steelblue") +
  geom_hline(yintercept = 1,
             linetype = "dashed") +
  labs(title = "Lift by Decile",
```

Bonus Challenge 2: Sensitivity Analysis

```r
# How does profit change with different costs?
cost_scenarios <- c(2, 5, 10, 15, 20)

sensitivity_results <- map_df(cost_scenarios,
  function(cost) {
    profit_analysis %>%
      mutate(
        cost_per_contact = cost,
        profit = map2_dbl(
          perc_sample, lift,
          ~calculate_profit(.x, .y, cost = cost)
        )
      )
  })

# Plot sensitivity
ggplot(sensitivity_results,
       aes(x = perc_sample, y = profit,
           color = factor(cost_per_contact))) +
  geom_line(size = 1) +
```

Bonus Challenge 3: Multi-Threshold Analysis

```r
# Instead of % contacted, use probability threshold
thresholds <- seq(0.1, 0.9, by = 0.1)

threshold_analysis <- map_df(thresholds,
  function(thresh) {
    selected <- test_data %>%
      filter(pred_prob >= thresh)

    if(nrow(selected) == 0) return(NULL)

    data.frame(
      threshold = thresh,
      n_contacted = nrow(selected),
      perc_contacted = nrow(selected) /
                       nrow(test_data),
      n_subscribers = sum(selected$subscribed),
      conversion_rate = mean(selected$subscribed)
    )
  })
```

Section 8

Submission Guidelines

What to Submit

Required Files

1. `classwork3.R` - Complete R script 2. `classwork3_report.pdf` - Written report 3. `campaign_dashboard.pdf` - Visual summary 4. `campaign_findings.txt` - Text summary

Report Structure:

- Executive Summary (1 page)
- Answers to 8 questions
- Analysis and insights
- Recommendation with justification

Answer Key Template

Question 1: Actual subscription rate = _____%

Question 2: At top 20%, reach _____% of subscribers

Question 3: Maximum lift = _____, meaning _____

Question 4: Lift drops below 2 at _____%

Question 5: Optimal contact % = _____

Question 6: Campaign unprofitable after _____%

Question 7: Better model: _____, because _____

Question 8: Recommendation: _____, justification: _____

Grading Rubric

Component	**Points**
Code completeness and correctness	30
All visualizations created	25
All 8 questions answered correctly	25
Executive summary quality	10
Code documentation and style	5
Written analysis depth	5
Bonus challenges	+5 each
Total	**100**

Key Formulas Reference

Cumulative Gains:

$$\text{Gain at } X\% = \frac{\text{Targets in top } X\%}{\text{Total targets}}$$

Lift:

$$\text{Lift} = \frac{\text{Response rate in selection}}{\text{Overall response rate}}$$

Profit:

$$\text{Profit} = (n_{subscribers} \times \text{revenue}) - (n_{contacted} \times \text{cost})$$

Expected Subscribers:

$$n_{subscribers} = n_{contacted} \times \text{baseline rate} \times \text{lift}$$

Common Mistakes to Avoid

Mistake 1: Not sorting by predicted probability

- Gains curve requires descending sort
- Always verify sort order

Mistake 2: Confusing lift with probability

- Lift is a multiplier, not a probability
- Lift of 2 means 2x more likely

Mistake 3: Using training data

- Always use test/holdout set
- Training data gives overoptimistic curves

Tips for Success

Technical Tips

- Test each function with small examples first - Use `head()` to verify data at each step - Save intermediate results - Comment your code thoroughly

Analysis Tips

- Look at the curves holistically - Don't just report numbers, interpret them - Think like a business decision-maker - Connect findings to profitability

Time Management

Suggested Timeline:

- **Minutes 0-12:** Part 1 (Gains curve)
- **Minutes 12-22:** Part 2 (Lift curve)
- **Minutes 22-34:** Part 3 (Profit analysis)
- **Minutes 34-40:** Part 4 (Comparison)
- **Minutes 40-45:** Part 5 (Summary)

If running behind: Focus on Parts 1-3, complete 4-5 later

Getting Help

During Classwork:

- Check the lecture slides (Chapter 3)
- Review previous classwork code
- Ask teaching assistant
- Discuss concepts with peers (but write own code)

Resources:

- `?arrange` - Sorting data
- `?ntile` - Creating deciles
- `?scales::dollar` - Currency formatting
- Lecture code examples

Data Validation Checklist

Before proceeding, verify:

- ✓ Test set has actual values (subscribed)
- ✓ Test set has predicted probabilities
- ✓ Predictions are between 0 and 1
- ✓ No missing values in key columns
- ✓ Sample size is reasonable (> 100)
- ✓ Baseline rate matches expectations

```r
# Quick validation
stopifnot(
  all(test_data$pred_prob >= 0 &
      test_data$pred_prob <= 1),
  !any(is.na(test_data$subscribed)),
  nrow(test_data) > 100
)
```

Expected Outputs Preview

Your final outputs should include:

1. **Gains plot**: Smooth curve from (0,0) to (1,1)
2. **Lift plot**: Decreasing curve starting high
3. **Profit plot**: Curve with clear maximum
4. **Comparison**: Two gains curves on same plot
5. **Summary**: Clear recommendation with numbers

All plots should be properly labeled with titles, axis labels, and legends.

Business Context Reminder

Why This Matters:

- Banks have limited marketing budgets
- Random campaigns waste money
- Targeted campaigns maximize ROI
- Your analysis guides real decisions

Real Impact:

- €10K budget $\rightarrow$ contact 2,000 people
- With your model $\rightarrow$ 300+ subscriptions
- Without model $\rightarrow$ 120 subscriptions
- **Value created: €18,000 additional profit**

Academic Integrity

Collaboration Policy

- Allowed: Discussing concepts and approaches - Allowed: Helping debug code errors - Not Allowed: Sharing complete code solutions - Not Allowed: Copying answers directly

Your submission must be your own work.

Professor Raco uses plagiarism detection tools.

Extended Deadline Option

Standard Deadline: Before next lecture

Extension Available: +2 days with 10% penalty

To Request Extension:

- Email Prof. Raco before original deadline
- Provide brief reason
- Commit to extended timeline

No extensions for: Bonus challenges

After Submission

What Happens Next:

1. Automated tests run on your code
2. Manual review of visualizations and report
3. Grading completed within 1 week
4. Feedback provided via email
5. Model solutions posted after grading

Top submissions may be shared (anonymously) as exemplars.

Learning Outcomes Checklist

After completing this classwork, you should be able to:

- ✓ Explain cumulative gains to a business stakeholder
- ✓ Calculate and interpret lift values
- ✓ Estimate campaign profitability
- ✓ Identify optimal campaign strategy
- ✓ Compare models using business metrics
- ✓ Make data-driven recommendations

Final Checklist Before Submission

Item	Done?
All 8 questions answered	☐
Four plots created and saved	☐
Executive summary written	☐
Code runs without errors	☐
Code is well-commented	☐
Files named correctly	☐
Report is spell-checked	☐
All files submitted	☐

Ready to Begin!

Good Luck!

You have 45 minutes

Remember: Focus on understanding,
not just getting the right numbers

Begin when ready!

Need Help?

Prof. Asc. Endri Raco, PhD

- Office: Data Science Lab, Room 305
- Email: endri.raco@upt.al
- Office Hours: Tuesday 14:00-16:00

Teaching Assistant

- Lab: Computer Lab B
- Drop-in Hours: Monday 10:00-12:00

Bonus: Real-World Application

Industry Example

A major European bank implemented similar analysis:
- Database: 500,000 customers - Contacted top 15% (75,000 customers) - Achieved lift of 4.2 - Generated €2.3M additional profit - Reduced contact costs by 85%

Your skills are valuable!

This type of analysis is in high demand in: banking, e-commerce, telecommunications, healthcare, and marketing.