Classwork 4: Predictor Insight Graphs and Model Interpretation

Introduction to Predictive Analytics in R

Prof. Asc. Endri Raco, PhD

Polytechnic University of Tirana

November 2025

Section 1

Classwork Overview

Today's Mission: Validate Model Interpretability

Create Predictor Insight Graphs

Scenario: The marketing director asks:

"I see your model has good AUC and lift, but can you explain HOW each variable affects subscription probability? Which customer characteristics really matter?"

Your Task: Create predictor insight graphs that visually explain each variable's relationship with the target.

Learning Objectives

By completing this classwork, you will:

1. Identify which variables need discretization
2. Apply quantile-based and custom binning strategies
3. Create predictor insight graph (PIG) tables
4. Generate dual-axis PIG plots
5. Interpret variable relationships
6. Identify data quality issues from PIG patterns
7. Communicate findings to stakeholders

Dataset: Bank Marketing (Continued)

Using the same dataset from previous classworks

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

Model Variables from Classwork 2:

- `age` (continuous)
- `balance` (continuous)
- `duration` (continuous)
- `campaign` (discrete)
- `education` (categorical)
- `marital` (categorical)
- `housing` (categorical)

Deliverables

What You'll Submit

1. R script with all code (`classwork4.R`) 2. Report answering 10 key questions 3. Six predictor insight graphs (one per variable) 4. Summary table comparing all variables 5. Executive interpretation document

Time Allocation: 45 minutes

Getting Started

Setup

Create `classwork4.R` and load necessary libraries

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

# Set seed for reproducibility
set.seed(456)

# Load your data from previous classwork
# Assumes you have basetable with model results
load("classwork2_results.RData")

# Verify data structure
glimpse(basetable)
str(basetable)
```

Section 2

Part 1: Variable Assessment (8 minutes)

Task 1.1: Identify Variable Types

```r
# Model variables
model_variables <- c("age", "balance", "duration",
                     "campaign", "education",
                     "marital", "housing")

# Function to check if discretization needed
should_discretize <- function(data, variable,
                              threshold = 5) {
  n_unique <- length(unique(data[[variable]]))
  return(n_unique > threshold)
}

# Check each variable
for (var in model_variables) {
  needs_disc <- should_discretize(basetable, var)
  n_unique <- length(unique(basetable[[var]]))
  cat(var, ": ", n_unique, " unique values - ",
      ifelse(needs_disc, "DISCRETIZE", "USE AS-IS"),
      "\n")
}
```

Task 1.2: Examine Variable Distributions

```r
# Summary statistics for continuous variables
continuous_vars <- c("age", "balance", "duration")

for (var in continuous_vars) {
  cat("\n=== ", var, " ===\n")
  print(summary(basetable[[var]]))

  # Check for outliers
  q1 <- quantile(basetable[[var]], 0.25, na.rm = TRUE)
  q3 <- quantile(basetable[[var]], 0.75, na.rm = TRUE)
  iqr <- q3 - q1

  n_outliers <- sum(basetable[[var]] < (q1 - 1.5*iqr) |
                    basetable[[var]] > (q3 + 1.5*iqr),
                    na.rm = TRUE)

  cat("Outliers: ", n_outliers, "\n")
}
```

Question 1: Which variable has the most outliers?

Task 1.3: Check for Missing Values

```r
# Missing value analysis
check_missing <- function(data, variables) {
  missing_report <- data.frame(
    Variable = character(),
    N_Missing = numeric(),
    Pct_Missing = numeric(),
    stringsAsFactors = FALSE
  )

  for (var in variables) {
    n_miss <- sum(is.na(data[[var]]))
    pct_miss <- round(n_miss / nrow(data) * 100, 2)

    missing_report <- rbind(missing_report,
      data.frame(Variable = var,
                 N_Missing = n_miss,
                 Pct_Missing = pct_miss))
  }

  return(missing_report)
}
```

Section 3

Part 2: Discretization (10 minutes)

Task 2.1: Quantile-Based Discretization

```r
# Discretize continuous variables into 5 bins
continuous_vars <- c("age", "balance", "duration")
n_bins <- 5

for (var in continuous_vars) {
  # Create discretized variable name
  disc_var <- paste0(var, "_disc")

  # Apply quantile-based discretization
  basetable[[disc_var]] <- cut(
    basetable[[var]],
    breaks = quantile(basetable[[var]],
                      probs = seq(0, 1, 1/n_bins),
                      na.rm = TRUE),
    include.lowest = TRUE,
    dig.lab = 4
  )

  # Show the bins created
  cat("\n=== Bins for", var, "===\n")
```

Task 2.2: Examine Bin Boundaries

```r
# Look at the bin labels
for (var in continuous_vars) {
  disc_var <- paste0(var, "_disc")
  cat("\n=== Bin labels for", var, "===\n")
  print(levels(basetable[[disc_var]]))
}
```

Question 2: Are the bin boundaries easy to interpret? Which variable has the messiest labels?

Task 2.3: Create Clean Bins for Age

```r
# Problem: Age bins like [18, 34.6] are hard to explain
# Solution: Use custom breaks with round numbers

age_breaks <- c(18, 30, 40, 50, 60, 95)

basetable$age_disc_clean <- cut(
  basetable$age,
  breaks = age_breaks,
  include.lowest = TRUE,
  right = TRUE
)

# Compare original vs clean
cat("\nOriginal age bins:\n")
print(table(basetable$age_disc))

cat("\nClean age bins:\n")
print(table(basetable$age_disc_clean))
```

Task 2.4: Apply Clean Binning to All Variables

```r
# Custom breaks for balance
# (in euros, rounded to thousands)
balance_breaks <- c(-8000, 0, 500, 1500,
                    5000, 102000)

basetable$balance_disc_clean <- cut(
  basetable$balance,
  breaks = balance_breaks,
  include.lowest = TRUE,
  right = TRUE
)

# Custom breaks for duration
# (in seconds, rounded to minutes)
duration_breaks <- c(0, 120, 240, 360,
                     600, 5000)

basetable$duration_disc_clean <- cut(
  basetable$duration,
  breaks = duration_breaks,
```

Task 2.5: Compare Binning Strategies

```r
# Compare quantile vs custom binning for age
comparison <- data.frame(
  Method = c("Quantile", "Custom"),
  N_Bins = c(
    length(unique(basetable$age_disc)),
    length(unique(basetable$age_disc_clean))
  ),
  Min_Size = c(
    min(table(basetable$age_disc)),
    min(table(basetable$age_disc_clean))
  ),
  Max_Size = c(
    max(table(basetable$age_disc)),
    max(table(basetable$age_disc_clean))
  )
)

print(comparison)
```

Section 4

Part 3: Creating PIG Tables (10 minutes)

Task 3.1: PIG Table Function

```r
# Function to create predictor insight graph table
create_pig_table <- function(data, target, variable) {

  # Remove NA values
  data_clean <- data[!is.na(data[[variable]]), ]

  # Group and calculate
  pig_table <- data_clean %>%
    group_by(!!sym(variable)) %>%
    summarise(
      Size = n(),
      N_Targets = sum(!!sym(target), na.rm = TRUE),
      Incidence = mean(!!sym(target), na.rm = TRUE),
      .groups = 'drop'
    )

  # Rename first column
  names(pig_table)[1] <- "Category"

  # Add percentage of total
```

Task 3.2: Create PIG Tables for All Variables

```r
# Variables to analyze (use clean versions)
pig_variables <- c("age_disc_clean",
                   "balance_disc_clean",
                   "duration_disc_clean",
                   "campaign",
                   "education",
                   "marital",
                   "housing")

# Create all PIG tables
pig_tables <- list()

for (var in pig_variables) {
  pig_tables[[var]] <- create_pig_table(
    data = basetable,
    target = "subscribed",
    variable = var
  )
}
```

Task 3.3: Examine Age PIG Table

```r
# Detailed look at age table
age_pig <- pig_tables[["age_disc_clean"]]

cat("Age Group Analysis:\n")
print(age_pig)

cat("\nKey Statistics:\n")
cat("Total sample:", sum(age_pig$Size), "\n")
cat("Overall incidence:",
    weighted.mean(age_pig$Incidence, age_pig$Size), "\n")
cat("Lowest incidence group:",
    age_pig$Category[which.min(age_pig$Incidence)], "\n")
cat("Highest incidence group:",
    age_pig$Category[which.max(age_pig$Incidence)], "\n")
```

Question 4: What age group has the highest subscription rate?

Task 3.4: Calculate Incidence Range

```r
# Function to calculate incidence range
calculate_range <- function(pig_table) {
  max(pig_table$Incidence) - min(pig_table$Incidence)
}

# Calculate for all variables
incidence_ranges <- sapply(pig_tables, calculate_range)

# Sort by range (predictive power indicator)
sorted_ranges <- sort(incidence_ranges,
                      decreasing = TRUE)

cat("Variables by Incidence Range:\n")
print(round(sorted_ranges, 4))
```

Question 5: Which variable shows the largest incidence range?

Task 3.5: Identify Small Sample Bins

```
# Check for bins with insufficient sample
min_sample_threshold <- 100

for (var_name in names(pig_tables)) {
  pig_table <- pig_tables[[var_name]]

  small_bins <- pig_table %>%
    filter(Size < min_sample_threshold)

  if (nrow(small_bins) > 0) {
    cat("\nWARNING:", var_name,
        "has small sample bins:\n")
    print(small_bins)
  }
}
```

Question 6: Do any variables have bins with fewer than 100 observations?

Section 5

Part 4: Creating PIG Plots (12 minutes)

Task 4.1: Basic PIG Plot Function

```r
library(ggplot2)

plot_pig <- function(pig_table, var_name) {

  # Calculate scaling factor for dual axis
  max_size <- max(pig_table$Size)
  max_incidence <- max(pig_table$Incidence)
  scale_factor <- max_size / max_incidence

  # Create base plot
  p <- ggplot(pig_table,
              aes(x = Category)) +
    # Bars for size
    geom_col(aes(y = Size),
             fill = "lightgray",
             alpha = 0.7,
             width = 0.6) +
    # Line for incidence (scaled)
    geom_line(aes(y = Incidence * scale_factor,
                  group = 1),
```

Task 4.2: Complete PIG Plot Function

```
plot_pig <- function(pig_table, var_name) {
  # ... previous code ...

  p <- p +
    # Points on line
    geom_point(aes(y = Incidence * scale_factor),
               color = "darkgreen",
               size = 3) +
    # Dual y-axis
    scale_y_continuous(
      name = "Sample Size",
      labels = comma,
      sec.axis = sec_axis(
        ~./scale_factor,
        name = "Subscription Rate",
        labels = percent_format(accuracy = 0.1)
      )
    ) +
    labs(title = paste("Predictor Insight:", var_name),
         x = var_name) +
```

Task 4.3: Generate All PIG Plots

```r
# Create plots for all variables
pig_plots <- list()

for (var_name in names(pig_tables)) {
  pig_plots[[var_name]] <- plot_pig(
    pig_table = pig_tables[[var_name]],
    var_name = var_name
  )
}

# Display age plot
print(pig_plots[["age_disc_clean"]])

# Save age plot
ggsave("pig_age.pdf",
       pig_plots[["age_disc_clean"]],
       width = 8,
       height = 6)
```

Task 4.4: Create Multi-Panel Display

```r
library(gridExtra)

# Display top 4 variables by incidence range
top_vars <- names(sorted_ranges)[1:4]

grid.arrange(
  pig_plots[[top_vars[1]]],
  pig_plots[[top_vars[2]]],
  pig_plots[[top_vars[3]]],
  pig_plots[[top_vars[4]]],
  ncol = 2,
  top = "Top 4 Predictive Variables"
)

# Save multi-panel plot
ggsave("pig_top4.pdf",
       width = 12,
       height = 10)
```

Task 4.5: Add Value Labels

```r
# Enhanced plot with incidence labels
plot_pig_labeled <- function(pig_table, var_name) {

  scale_factor <- max(pig_table$Size) /
                  max(pig_table$Incidence)

  ggplot(pig_table, aes(x = Category)) +
    geom_col(aes(y = Size),
             fill = "lightgray",
             alpha = 0.7,
             width = 0.6) +
    geom_line(aes(y = Incidence * scale_factor,
                  group = 1),
              color = "darkgreen",
              size = 1.5) +
    geom_point(aes(y = Incidence * scale_factor),
               color = "darkgreen",
               size = 3) +
    # Add labels
    geom_text(aes(y = Incidence * scale_factor,
```

Section 6

Part 5: Interpretation (10 minutes)

Task 5.1: Identify Relationship Patterns

```r
# Classify each variable's relationship
classify_relationship <- function(pig_table) {

  incidences <- pig_table$Incidence

  # Check for monotonic increase
  is_increasing <- all(diff(incidences) >= 0)

  # Check for monotonic decrease
  is_decreasing <- all(diff(incidences) <= 0)

  # Calculate correlation with ordinal position
  positions <- 1:nrow(pig_table)
  correlation <- cor(positions, incidences)

  if (is_increasing) {
    return("Positive (monotonic)")
  } else if (is_decreasing) {
    return("Negative (monotonic)")
  } else if (abs(correlation) > 0.5) {
```

Task 5.2: Check Data Quality Flags

```r
# Function to identify potential data issues
check_data_quality <- function(pig_table, var_name) {

  issues <- c()

  # Check 1: Very small bins
  small_bins <- sum(pig_table$Size < 50)
  if (small_bins > 0) {
    issues <- c(issues,
      paste(small_bins, "bins with < 50 obs"))
  }

  # Check 2: Extreme incidence values
  if (max(pig_table$Incidence) > 0.5) {
    issues <- c(issues, "Very high incidence")
  }

  # Check 3: Erratic pattern
  if (sd(diff(pig_table$Incidence)) > 0.05) {
    issues <- c(issues, "Erratic pattern")
```

Task 5.3: Create Summary Comparison Table

```r
# Create comprehensive summary
summary_table <- data.frame(
  Variable = names(pig_tables),
  N_Categories = sapply(pig_tables, nrow),
  Min_Incidence = sapply(pig_tables,
    function(x) min(x$Incidence)),
  Max_Incidence = sapply(pig_tables,
    function(x) max(x$Incidence)),
  Incidence_Range = incidence_ranges,
  Min_Sample_Size = sapply(pig_tables,
    function(x) min(x$Size))
)

# Sort by incidence range
summary_table <- summary_table %>%
  arrange(desc(Incidence_Range))

# Format for display
summary_table$Min_Incidence <-
  percent(summary_table$Min_Incidence, accuracy = 0.1)
```

Task 5.4: Interpret Duration Variable

```r
# Detailed analysis of duration
duration_pig <- pig_tables[["duration_disc_clean"]]

cat("=== Duration Analysis ===\n\n")

cat("Categories and Rates:\n")
print(duration_pig[, c("Category", "Incidence", "Size")])

# Calculate lift for each category
baseline_rate <- weighted.mean(duration_pig$Incidence,
                               duration_pig$Size)

duration_pig$Lift <- duration_pig$Incidence / baseline_rate

cat("\nLift by Duration:\n")
print(duration_pig[, c("Category", "Lift")])

cat("\nInterpretation:\n")
cat("Baseline rate:", percent(baseline_rate, 0.1), "\n")
cat("Longest duration category has",
```

Task 5.5: Compare Categorical Variables

```r
# Focus on categorical variables
categorical_vars <- c("education", "marital", "housing")

cat("=== Categorical Variable Comparison ===\n\n")

for (var in categorical_vars) {
  pig <- pig_tables[[var]]

  cat("\n", var, ":\n")
  cat("Number of categories:", nrow(pig), "\n")
  cat("Most predictive category:",
      pig$Category[which.max(pig$Incidence)],
      "(",
      percent(max(pig$Incidence), 0.1),
      ")\n")
  cat("Least predictive category:",
      pig$Category[which.min(pig$Incidence)],
      "(",
      percent(min(pig$Incidence), 0.1),
      ")\n")
}
```

Section 7

Part 6: Executive Summary (5 minutes)

Task 6.1: Generate Key Findings

```r
# Generate executive summary
generate_executive_summary <- function(pig_tables,
                                       summary_table) {

  cat("=" %>% rep(60) %>% paste(collapse = ""), "\n")
  cat("PREDICTOR INSIGHT GRAPH ANALYSIS\n")
  cat("Bank Marketing Campaign Model\n")
  cat("Analyst: Prof. Asc. Endri Raco, PhD\n")
  cat("=" %>% rep(60) %>% paste(collapse = ""), "\n\n")

  cat("KEY FINDINGS:\n\n")

  # Most predictive variable
  top_var <- summary_table$Variable[1]
  top_range <- summary_table$Incidence_Range[1]

  cat("1. MOST PREDICTIVE VARIABLE:", top_var, "\n")
  cat("   - Incidence range:", round(top_range, 4), "\n")

  # Continue on next slide...
```

Task 6.2: Complete Executive Summary

```r
generate_executive_summary <- function(pig_tables,
                                       summary_table) {
  # ... previous code ...

  # Weakest variable
  weak_var <- summary_table$Variable[nrow(summary_table)]
  weak_range <- summary_table$Incidence_Range[
    nrow(summary_table)]

  cat("\n2. WEAKEST PREDICTOR:", weak_var, "\n")
  cat("   - Incidence range:", round(weak_range, 4), "\n")

  # Data quality issues
  cat("\n3. DATA QUALITY:\n")
  total_vars <- nrow(summary_table)
  small_sample_vars <- sum(summary_table$Min_Sample_Size < 100)

  cat("   - Variables analyzed:", total_vars, "\n")
  cat("   - Variables with small samples:",
      small_sample_vars, "\n")
```

Task 6.3: Create Variable Ranking

```r
# Rank variables by predictive power
variable_ranking <- summary_table %>%
  select(Variable, Incidence_Range, N_Categories) %>%
  mutate(
    Rank = row_number(),
    Predictive_Power = case_when(
      Incidence_Range > 0.15 ~ "Strong",
      Incidence_Range > 0.05 ~ "Moderate",
      TRUE ~ "Weak"
    )
  )

print(variable_ranking)

# Save ranking
write.csv(variable_ranking,
          "variable_ranking.csv",
          row.names = FALSE)
```

Question 9: How many variables show "Strong" predictive power?

Task 6.4: Document Unexpected Patterns

```r
# Document surprises or concerns
document_patterns <- function(pig_tables) {

  cat("PATTERN ANALYSIS NOTES:\n\n")

  for (var_name in names(pig_tables)) {
    pig <- pig_tables[[var_name]]

    # Check for U-shaped or inverted-U patterns
    if (nrow(pig) >= 3) {
      mid_idx <- ceiling(nrow(pig) / 2)
      mid_inc <- pig$Incidence[mid_idx]
      edge_avg <- mean(c(pig$Incidence[1],
                         pig$Incidence[nrow(pig)]))

      if (abs(mid_inc - edge_avg) > 0.05) {
        cat("\n", var_name, ": Non-monotonic pattern\n")
        cat("  Middle:", percent(mid_inc, 0.1), "\n")
        cat("  Edges avg:", percent(edge_avg, 0.1), "\n")
      }
```

Task 6.5: Prepare Stakeholder Presentation

```r
# Create presentation-ready statements
create_insights <- function(pig_tables) {

  cat("STAKEHOLDER TALKING POINTS:\n\n")

  # Duration insight
  dur_pig <- pig_tables[["duration_disc_clean"]]
  longest_inc <- max(dur_pig$Incidence)
  shortest_inc <- min(dur_pig$Incidence)

  cat("1. Call Duration:\n")
  cat("   'Customers who speak with us longer are",
      round(longest_inc/shortest_inc, 1),
      "times more likely to subscribe'\n\n")

  # Age insight
  age_pig <- pig_tables[["age_disc_clean"]]
  cat("2. Age:\n")
  cat("   'Subscription rates range from",
      percent(min(age_pig$Incidence), 0.1),
```

Section 8

Submission Guidelines

What to Submit

Required Files

1. `classwork4.R` - Complete R script with comments 2. `classwork4_report.pdf` - Written report 3. `pig_plots/` - Folder with all 6-7 PIG plots 4. `variable_ranking.csv` - Summary table 5. `executive_summary.txt` - Key findings

Report Structure:

- Executive summary (1 page)
- Answers to 10 questions
- Interpretation of each variable
- Data quality concerns
- Recommendations

Answer Key Template

Question 1: Variable with most outliers: _____

Question 2: Variable with messiest bin labels: _____

Question 3: Binning method with more balanced groups: _____

Question 4: Age group with highest rate: _____

Question 5: Variable with largest incidence range: _____

Question 6: Variables with bins < 100 obs: _____

Question 7: Duration pattern interpretation: _____

Question 8: Most variable categorical predictor: _____

Question 9: Number of "Strong" predictors: _____

Question 10: Balance variable insight: _____

Grading Rubric

Component	Points
Discretization correct	20
PIG tables created properly	20
All plots generated and saved	20
All 10 questions answered	20
Interpretation quality	10
Executive summary	5
Code quality and documentation	5
Total	**100**

Bonus: +5 points for identifying and properly handling a data quality issue

Common Mistakes to Avoid

Mistake 1: Using original variables instead of discretized versions

- Always use `*_disc_clean` versions for continuous variables

Mistake 2: Forgetting to handle missing values

- Use `na.rm = TRUE` in calculations

Mistake 3: Not labeling axes properly

- Both axes need clear labels and appropriate scales

Mistake 4: Interpreting correlation as causation

- PIG shows association, not causation

Tips for Success

Technical Tips

- Test functions on one variable before looping - Save plots as you create them - Use descriptive variable names - Comment your code thoroughly

Interpretation Tips

- Compare patterns across variables - Look for business logic in patterns - Flag unexpected results for investigation - Think about actionability

Communication Tips

- Use percentages in stakeholder messages - Compare categories to baseline - Avoid jargon - Focus on business impact

Time Management

Suggested Timeline:

- **Minutes 0-8:** Part 1 (Variable assessment)
- **Minutes 8-18:** Part 2 (Discretization)
- **Minutes 18-28:** Part 3 (PIG tables)
- **Minutes 28-40:** Part 4 (Plotting)
- **Minutes 40-45:** Part 5-6 (Interpretation & summary)

If running behind: Complete Parts 1-4, then do executive summary. Return to detailed interpretation later.

Expected Outputs Preview

Your final submission should include:

1. **6-7 PIG plots**: One per variable, properly formatted
2. **Summary table**: Ranking all variables
3. **Executive summary**: 1-page key findings
4. **Detailed report**: Answers to all questions
5. **Clean code**: Well-commented R script

All plots should have: - Clear titles - Labeled axes - Dual y-axis (size and incidence) - Professional appearance

Integration with Previous Work

Connection to Classwork 2 and 3:

- Classwork 2: You built the model
- Classwork 3: You evaluated business value (lift/profit)
- Classwork 4: You validate interpretability

Together, these three form complete model validation:

1. Technical performance (AUC)
2. Business value (profit)
3. Interpretability (PIG)

Validation Checklist

Before submitting, verify:

Item	**Done?**
All variables discretized correctly	☐
All 6-7 PIG tables created	☐
All plots generated and saved	☐
All 10 questions answered	☐
Executive summary written	☐
Code runs without errors	☐
All files named correctly	☐
Report is spell-checked	☐

Getting Help

During Classwork:

- Review Chapter 4 lecture slides
- Check your Classwork 2 code for data structure
- Consult with teaching assistant
- Ask peers about concepts (not code)

Resources:

- `?cut` - Discretization help
- `?sec_axis` - Dual axis plots
- ggplot2 documentation
- Previous classwork solutions

Real-World Application

Industry Context

PIG graphs are commonly used in:
- Credit scoring models (banking) - Customer churn prediction (telecom) - Marketing response models (retail) - Fraud detection (insurance) - Medical diagnosis (healthcare)

Why? Because stakeholders need to understand:

- Which factors drive predictions
- Whether patterns make business sense
- How to act on model insights

Learning Outcomes

After completing this classwork, you will be able to:

- ✓ Discretize continuous variables appropriately
- ✓ Create PIG tables with size and incidence
- ✓ Generate professional dual-axis visualizations
- ✓ Interpret variable relationships
- ✓ Identify data quality issues
- ✓ Communicate findings to stakeholders
- ✓ Rank variables by predictive importance

Advanced Challenge (Optional)

For students who finish early:

1. Create an automated PIG report generator
2. Compare different binning strategies (3, 5, 7 bins)
3. Calculate Weight of Evidence (WoE) for each bin
4. Create an interactive PIG dashboard using Shiny
5. Analyze interaction effects between variables

Bonus: +10 points for completing one advanced challenge

Academic Integrity Reminder

Collaboration Policy

- Allowed: Discussing interpretation approaches - Allowed: Helping debug specific errors - Not Allowed: Sharing complete code - Not Allowed: Copying plot outputs

Your work must be your own.

Prof. Raco uses similarity detection tools.

After Submission

What Happens Next:

1. Code reviewed for correctness and style
2. Plots evaluated for quality and completeness
3. Interpretations assessed for accuracy
4. Grading completed within 1 week
5. Feedback provided via email
6. Solutions posted after deadline

Outstanding work may be featured (anonymously) as exemplars in future courses.

Course Progress

You've Completed the Core Curriculum!

- ✓ Basetable construction
- ✓ Logistic regression modeling
- ✓ Variable selection
- ✓ AUC evaluation
- ✓ Business metrics (gains/lift)
- ✓ Model interpretation (PIG)

Next: Advanced topics and final project

Final Reminders

Critical Points

- Use `_disc_clean` versions for continuous variables - Calculate scaling factor correctly for dual-axis plots - Interpret patterns in business context - Document any data quality concerns - Save all plots before closing R

Time: 45 minutes

Due: Before next lecture

Ready to Begin!

Good Luck!

You have 45 minutes

Remember: Focus on understanding the patterns, not just creating pretty pictures

Begin when ready!

Support Available

Need Help?

Prof. Asc. Endri Raco, PhD

Practical Industry Example

Case Study: Retail Bank

A major bank used PIG analysis to understand credit card uptake:
- Duration variable showed 5x difference - Age pattern revealed optimal target: 30-50 year-olds - Balance variable had U-shaped pattern (investigation needed) - Education showed minimal effect (removed from model)
Result: 40% improvement in campaign efficiency

Your skills are valuable in the real world!