Predictive Optimization of Phone Call Marketing

Data Science and Decision Optimization in Banking & Financial Services

Zhaoyang Li, Jasper Ye, Ziyu Zhou, Chloe Pan

Duke University

May 29, 2025

Agenda

Motivation

- Real-world challenges
  - Many calls receive no answer
  - Only 2% of cold calls lead to a conversion
  - Time, effort, and money are wasted
- Key questions: Who should we call? Who should we call again?
- **Goal: Find people worth calling before we call them**
  - Save marketing costs 💵
  - Boost total revenue 🤑
  - Improve marketing efficiency under limited staff & time 🧑‍💼⏱️
  - Increase customer satisfaction by offering personalized deals & reduce unwanted calls 😃

Problem Statement

Based on data from direct marketing campaigns (phone calls) of a Portuguese banking institution, this project aims to achieve two objectives:

- **Customer Response Prediction**

  We apply statistical and machine learning methods to predict each customer's **likelihood of subscribing to a term deposit**.

  By analyzing patterns in **customer financial profiles** and **past campaign interactions**, the model helps identify and prioritize customers who are more likely to respond positively, enabling more targeted and effective outreach.

Problem Statement

- **Marketing Strategy Optimization**

  We aims to maximize the **total expected revenue** from term deposit subscriptions by **selecting customers to recall under the same offer**.

  Each customer is associated with a **predicted subscription probability** from the response model, an **estimated revenue** upon subscribing from another model trained on a separate dataset, and a **predicted contact cost**.

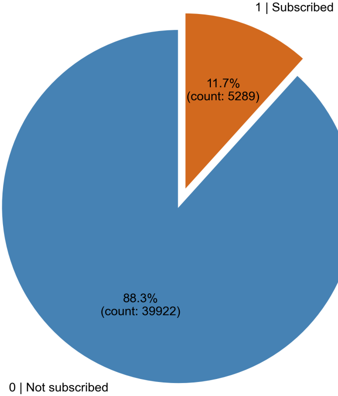
  Subject to a **total contact limit** within a period, the optimization model identifies a subset of customers who are more cost-effective to recall, thereby improving the overall efficiency and return of the campaign.

Data Source

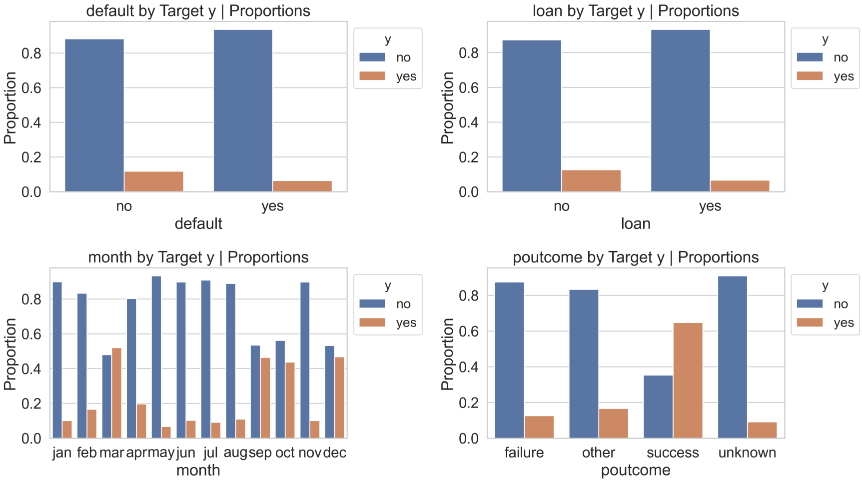
- This database is provided by UCI Machine Learning Repository. The data is about **direct marketing campaigns** by a Portuguese bank, which relied on **phone calls** to reach customers.
  https://archive.ics.uci.edu/dataset/222/bank+marketing
- Features:
  `age`, ~~`job`, `marital`, `education`~~, `default` (has credit in default), `balance` (average yearly balance), `housing` (has housing loan), `loan` (has personal loan), `day` & `month` (last contact date), `duration` (last contact duration), `campaign` (number of contacts in current campaign), `pdays` (days since last contact), `previous` (number of contacts before this campaign), `poutcome` (outcome of the previous campaign)
- Target: `y`, which indicates whether the client subscribed to a term deposit

Proportion of Subscription Outcomes

- **Class Imbalance:** Only 11.7% of clients subscribed, indicating a highly imbalanced target variable.

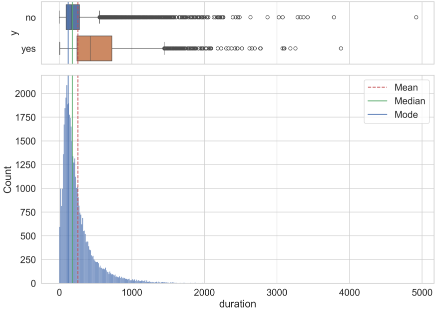


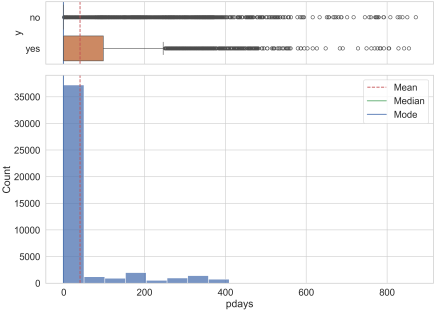
Categorical Variable Distribution by Subscription Status



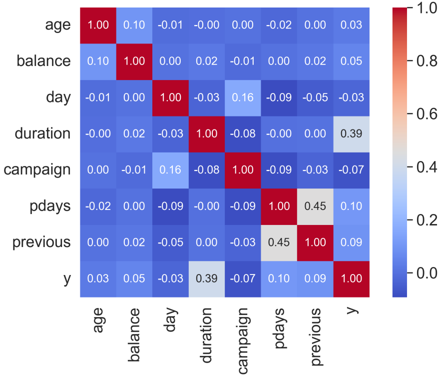
Distribution of Key Numerical Variables

- **Duration:** Longer calls are associated with higher subscription probability; highly right-skewed.
- **Pdays:** Many clients have no prior contact (`pdays` = -1); recent contact (low `pdays`) improves subscription chances.





Correlation Analysis and Preprocessing



- **Low Multicollinearity:** Most pairwise correlations are weak; majority near 0.
- **Top Correlation:** `pdays` vs. `previous` ($r = 0.45$), reflecting engagement history.
- **Target Relationship:** `duration` has moderate positive correlation with `y` ($r = 0.39$).
- **Preprocessing:**
  - Dropped `unknown` and `other` categories.
  - Standardized: `duration`, `balance`, `campaign`, `pdays`.

Logistic Regression

- Objective: Predict the **likelihood** of a customer subscribing to a term deposit based on customer financial profiles and past campaign interactions
- Outcome variable y is **binary** ("yes" or "no" for whether a client has subscribed to a term deposit)
- Use of **dummy variables**: Some of the input variables are categorical variables (e.g. month–last contact month of year "may", "jun", etc.)

Formula

- $$log\left(\frac{p}{1-p}\right) = \beta_0 + \beta_1 X_1 + \beta_2 X_2 + \cdots + \beta_k X_k$$

  Where:

  - p: Probability of the outcome (subscription = "yes"),
  - $log(\frac{p}{1-p})$ is logit of the probability
  - β_0: intercept
  - $\beta_1, \beta_2, \ldots, \beta_k$: represent the change in the log-odds of the outcome for a one-unit increase in the predictor
- **Odds Ratio**: Odds Ratio$=e^{\beta}$

Model Training and Evaluation

- **Train/Test Split:** 70/30 split (stratified);
- **Model:** logistic regression model (binary classifier) with robust standard errors
- **Evaluation Metrics** calculated at 0.5 probability threshold
- **In-sample Validation Result:**

In-Sample Confusion Matrix

	Predicted 0	**Predicted 1**	**Total**
Actual 0	27,255 (85.95%)	680 (2.14%)	27,935
Actual 1	2,524 (7.96%)	1,250 (3.94%)	3,774
Total	29,779	1,930	31,709

In-Sample Performance Metrics

Metric	**Value**
Accuracy	0.899
Recall (Sensitivity)	0.331
Specificity	0.976
Precision	0.648
F1 Score	0.438
AUC	0.897

Model Training and Evaluation

- **Out-of-sample Validation Results:**

Out-Sample Confusion Matrix

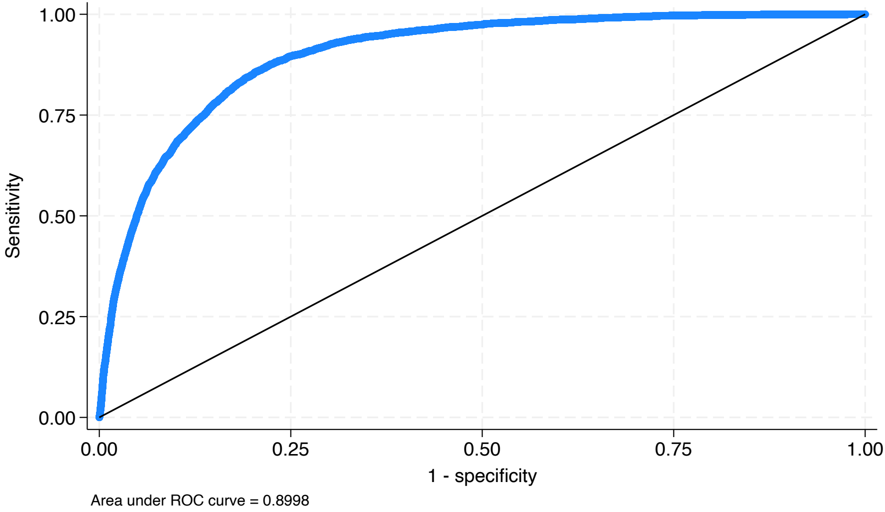
	Predicted 0	**Predicted 1**	**Total**
Actual 0	11,705 (86.69%)	282 (2.09%)	11,987
Actual 1	1,004 (7.44%)	511 (3.78%)	1,515
Total	12,709	793	13,502

Out-Sample Performance Metrics

	Accuracy	Precision	Recall	Specificity	F1-Score	AUC
Logit Model	.9047549	.6443884	.3372937	.9764745	.4428076	.9060501

Threshold = 0.5 Validation N = 13,502

ROC Curve


1.00
0.75
0.50
0.25
0.00
Sensitivity
0.00
0.25
0.50
0.75
1.00
1 - specificity
Area under ROC curve = 0.8998

Logistic Model

Variable	Coefficient	Variable	Coefficient
age	-0.007*** (0.000)	month_num=8	0.532*** (0.000)
balance	0.000*** (0.000)	month_num=9	1.990*** (0.000)
default_num	-0.105 (0.500)	month_num=10	2.034*** (0.000)
housing_num	-0.857*** (0.000)	month_num=11	0.305** (0.018)
loan_num	-0.495*** (0.000)	month_num=12	1.876*** (0.000)
month_num=2	0.986*** (0.000)	duration	0.004*** (0.000)
month_num=3	2.846*** (0.000)	campaign	-0.096*** (0.000)
month_num=4	1.207*** (0.000)	previous	0.010 (0.314)
month_num=5	0.199 (0.106)	success	1.862* (0.050)
month_num=6	0.536*** (0.000)	failure	-0.443 (0.641)
month_num=7	0.404*** (0.001)	constant	-3.428*** (0.000)
Day of month (1–31)	0.004 (0.134)	contacted_before=0	0.000 (.)
contacted_before=1	0.830 (0.382)		

p-values in parentheses. * $p < 0.10$, ** $p < 0.05$, *** $p < 0.01$

Odds Ratio

- The interpretation depends on whether the odds ratio is greater than, less than, or equal to 1 (e.g. $OR > 1:$ The predictor increases the odds of the outcome (subscription)).

Variable	**Odds Ratio (SE)**	**Variable**	**Odds Ratio (SE)**
age	0.993*** (0.002)	month_num=8	1.702*** (0.216)
balance	1.000*** (0.000)	month_num=9	7.313*** (1.188)
default_num	0.900 (0.141)	month_num=10	7.641*** (1.181)
housing_num	0.424*** (0.019)	month_num=11	1.357* (0.175)
loan_num	0.610*** (0.036)	month_num=12	6.529*** (1.480)
month_num=2	2.680*** (0.386)	duration	1.004*** (0.000)
month_num=3	17.224*** (2.822)	campaign	0.909*** (0.010)
month_num=4	3.342*** (0.427)	contacted_before=0	1.000 (.)
month_num=5	1.220 (0.150)	contacted_before=1	2.294 (2.179)
month_num=6	1.709*** (0.224)	previous	1.010 (0.010)
month_num=7	1.498** (0.186)	success	6.439 (6.126)
Day of month	1.004 (0.003)	failure	0.642 (0.610)

Exponentiated coefficients; Standard errors in parentheses.
* $p < 0.05$, ** $p < 0.01$, *** $p < 0.001$
Observations: 45,211

Interpretation

- **Strongest Positive Drivers** (Higher Subscription Odds):
  - **Seasoning Effects** (Set January as reference): March (OR=17.2), October (OR=7.6), September (OR=7.3); may align with tax return season, Post-Summer effects
  - **Previous marketing campaign success**: "Success" Poutcome (OR=6.4)
  - **Duration**: (OR=1.004)
- **Strongest Negative Drivers** (Lower Subscription Odds):
  - **Loans**: Loan holders have 39% lower odds to subscribe (OR=0.61)
  - **Campaign** (contacts performed during this campaign and for this client): Each additional contact reduces odds by 9.1% (**diminishing returns**) (OR=0.909)
  - **Housing Loan**: (OR=0.424)

Machine Learning Model XGBoost

- Captures **non-linear** interactions.
- Provides tools like `scale_pos_weight` to **handle class imbalance** effectively.
- Includes both **L1 (Lasso) and L2 (Ridge) regularization** terms in its objective function, helping control model complexity, prevents overfitting, and improves generalization.
- Boosting for **sequential** error correction, making it more powerful in reducing bias and improving predictive accuracy.

How XGBoost Works

- Objective function
  The overall objective to minimize is:

$$\mathcal{L}(\phi) = \sum_{i=1}^{n} \ell(y_i, \hat{y}_i) + \sum_{k=1}^{K} \Omega(f_k)$$

  Where:
  - $\ell(y_i, \hat{y}_i)$: Loss function (e.g., **log loss** for classification),
  - f_k: Regression trees added one at a time,
  - $\Omega(f) = \gamma T + \frac{1}{2}\lambda \sum w_j^2 + \alpha \sum |w_j|$: Regularization term.

How XGBoost Works

- Loss function
  For binary classification, XGBoost uses **logistic loss**:

$$\ell(y_i, \hat{y}_i) = -y_i \log(p_i) - (1 - y_i) \log(1 - p_i)$$

  Where

$$p_i = \sigma(\hat{y}_i) = \frac{1}{1 + e^{-\hat{y}_i}}$$

- Regularization
  To avoid overfitting:
  - **L2 regularization:** $\lambda \sum w_j^2$
  - **L1 regularization:** $\alpha \sum |w_j|$
  - γ**:** Penalty for each added leaf node (simplifies tree)

How XGBoost Optimizes the Model

- Additive Tree Model
  Boosting builds the model iteratively:

$$\hat{y}_i^{(t)} = \hat{y}_i^{(t-1)} + f_t(x_i)$$

At each round, a new tree f_t is added to minimize the **second-order Taylor approximation** of the objective:

$$\mathcal{L}^{(t)} \approx \sum_{i=1}^{n} \left[g_i f_t(x_i) + \frac{1}{2} h_i f_t(x_i)^2 \right] + \Omega(f_t)$$

Where:

- $g_i = \partial_{\hat{y}_i} \ell(y_i, \hat{y}_i)$ (first derivative),
- $h_i = \partial^2_{\hat{y}_i} \ell(y_i, \hat{y}_i)$ (second derivative).

How XGBoost Optimizes the Model

- Tree Building
  - The algorithm greedily splits nodes to maximize **gain** (improvement in the objective).
  - The best split maximizes:

$$\text{Gain} = \frac{1}{2}\left[\frac{G_L^2}{H_L+\lambda} + \frac{G_R^2}{H_R+\lambda} - \frac{(G_L+G_R)^2}{H_L+H_R+\lambda}\right] - \gamma$$

- Optimal Leaf Weight
  Each leaf's weight is calculated as:
  - **L2 only:** $w_j^* = -\frac{G_j}{H_j+\lambda}$
  - **L1 + L2:**

$$w_j^* = -\frac{\text{sign}(G_j)\cdot \max(|G_j|-\alpha, 0)}{H_j+\lambda}$$

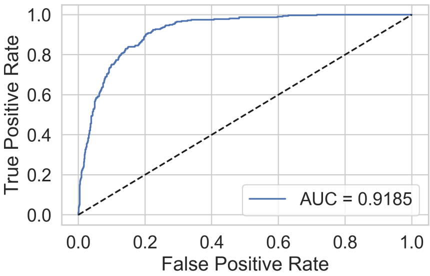
Model Training and Evaluation

- **Train/Test Split:** 80/20 split stratified on `y`; handled class imbalance with `scale_pos_weight` ≈ 3.01.
- **Model:** `XGBClassifier` with `hist` method and `logloss`; tuned via grid search with 5-fold CV.
- **Training Results:**
  - Accuracy: **0.8608**, AUC-ROC: **0.9276**
- **Test Set Results:**
  - Accuracy: **0.8516**, AUC-ROC: **0.9185**

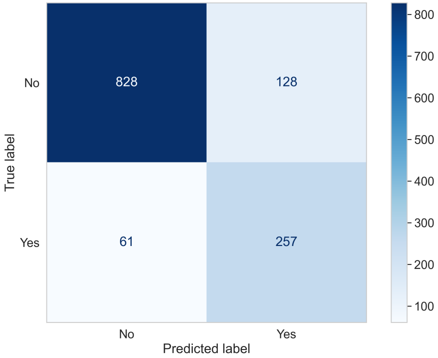
Class	Precision	Recall	F1-score	Support
0 (No subscription)	0.93	0.87	0.90	956
1 (Subscription)	0.67	0.81	0.73	318
Accuracy			**0.85**	1274
Macro avg	0.80	0.84	0.81	1274
Weighted avg	0.87	0.85	0.86	1274

XGBoost classification metrics on the test set

Performance Visualization



ROC curve on test set



Confusion matrix on test set

Model Interpretation with SHAP

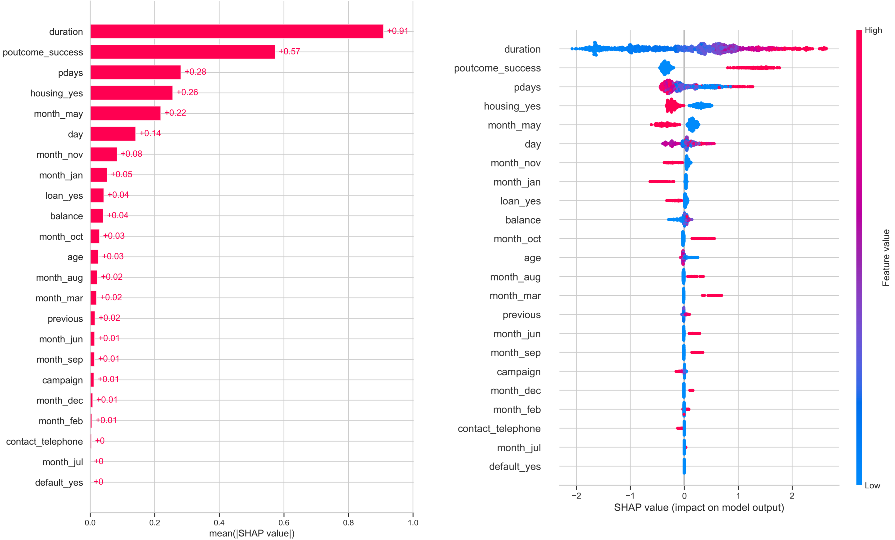
- SHAP(SHapley Additive exPlanations): aiming to explain the **prediction for a single instance** x by assigning **each feature a contribution** to the final prediction.
- Mathematical Formulation
  The prediction from model is explained as:

$$f(x) = \phi_0 + \sum_{j=1}^{d} \phi_j$$

  Where:
  - $f(x)$: Model prediction for input x,
  - ϕ_0: Baseline prediction (expected output across the training set),
  - ϕ_j: SHAP value for feature j,
  - d: Number of features.

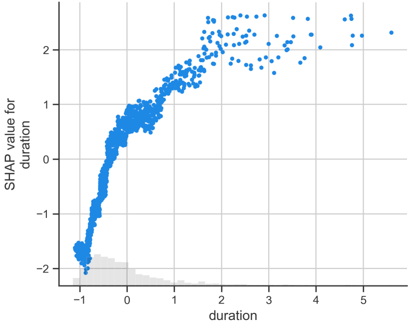
  Each ϕ_j tells **how much feature** j contributed to moving the prediction away from ϕ_0.

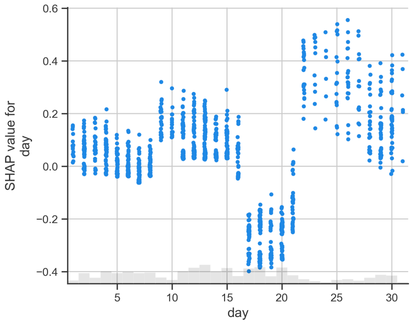
SHAP Interpretation of Global Importance



SHAP Dependence of Key Features

- **Duration:** Longer call durations sharply increase subscription likelihood, but gains flatten after a threshold (diminishing returns).
- **Day of Month:** Early (days 1–15) and late-month (days 21–31) contacts show higher responsiveness; mid-month contacts (days 16–20) lower responsiveness.





Key Findings from SHAP Analysis

- **Communication Quality:**
  - Longer call durations strongly increase subscription likelihood, reflecting higher client engagement.
- **Client History:**
  - Success in previous campaigns (`poutcome_success`) substantially boosts future subscription probability.
- **Recency of Contact:**
  - More recent contacts (lower `pdays`) correlate with higher subscription rates.
- **Financial Flexibility:**
  - Clients without a housing loan are more likely to subscribe, suggesting financial freedom plays a role.
- **Seasonal Timing:**
  - Lower subscription rates observed for May, November, and January, likely linked to public holidays, fiscal year-end, and post-holiday financial strain.
  - August, October, and March are more favorable months, aligned with leisure periods and financial stability windows.

Deep Neural Network (DNN)

- Inspired by the human brain, neural networks model complex, non-linear relationships
- DNNs contain multiple hidden layers, enabling hierarchical feature learning.
- Lower layers learn simple patterns, deeper layers learn abstract interactions
- Suitable for high-dimensional problems like customer subscription prediction

Why Deep Learning?

- Predicting subscriptions involves subtle and nonlinear patterns.
  - Interactions between age, balance, contact method, prior campaign success, and timing
- Deep learning models these interactions naturally and efficiently
- Offers significant performance gains over simpler models like logistic regression

Overcoming Data Challenges

- **Class Imbalance**: Only 11.7% subscribed
  - Applied class weighting in loss function
  - Performed oversampling of the minority class
- **Threshold Tuning**:
  - Optimized probability threshold (selected 0.48 instead of 0.5) to maximize F1-score
- Result: Better balance between precision and recall

DNN Training Process

- 80/20 train-test split (stratified)
- Preprocessing:
  - One-hot encoding (categorical), standard scaling (numerical)
- Trained with Adam optimizer (learning rate $= 0.001$) for 50 epochs
- Dropout for regularization, early stopping to prevent overfitting
- Post-training: threshold tuning based on F1-score

How Did DNN Perform?

Model	Accuracy	Precision	Recall	F1-Score	AUC
Logistic Regression	90.5%	64.4%	33.7%	44.3%	0.9061
XGBoost	85.2%	67%	81%	73%	0.9185
DNN	**92.2%**	**88%**	**98%**	**92.4%**	**0.9593**

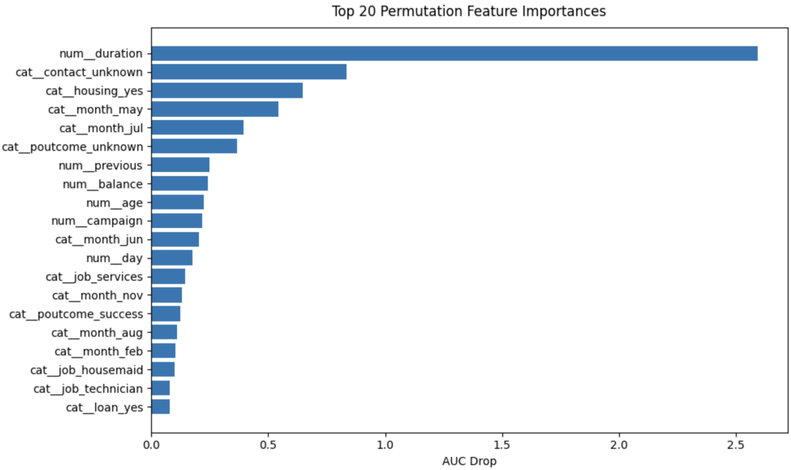
Key Assumptions in Applying DNN

- **Complexity of Relationships:** Customer subscription behavior involves non-linear and hierarchical feature interactions.
- **Sufficient Signal:** The dataset contains enough informative patterns for the DNN to learn meaningful relationships.
- **Data Representativeness:** Training data is assumed to reflect future real-world scenarios.
- **Preprocessing Validity:** One-hot encoding, scaling, and resampling maintain the integrity of the learning process.
- **Model Convergence:** The chosen architecture and training configuration (optimizer, dropout, early stopping) enable stable learning.
- **Threshold Optimization Matters:** Using a tuned threshold (0.48) improves performance over default 0.5.

DNN Concerns

- DNNs are powerful but lack interpretability.
- Hard to trace how input features influence predictions
- This is a concern in regulated industries (finance, healthcare).
- Solutions: Post hoc tools like SHAP can be used to explain predictions.

Model Interpretability via Feature Importance



- **duration** is by far the most influential feature in predicting term deposit subscription.

DNN Takeaways

- Deep learning significantly improved prediction performance.
- Tailored techniques (class weighting, resampling, threshold tuning) addressed dataset challenges.
- Tradeoff: accuracy vs. interpretability must be considered before deployment.
- Integrate explainability tools or explore interpretable deep models

Decision Optimization

- Objective: Determine which **previously contacted customers** deserve another call to maximize the **expected net revenue**

$$\text{Maximize} \quad \sum_{i=1}^{n} [(r_i - c_i) \cdot p_i] \cdot x_i$$

$$\text{subject to} \quad \begin{cases} r_i = f(\text{age}_i, \text{balance}_i, \text{personal loan}_i, \text{housing loan}_i) \\ c_i = g(\text{duration}_i, \text{poutcome}_i, \text{pdays}_i, \ldots) \\ \sum_{i=1}^{n} x_i \leq H \\ x_i \in \{0, 1\}, \quad \forall i = 1, 2, \ldots, n \end{cases}$$

Where:
r_i: Estimated revenue from customer i, based on their age, account balance, and personal and housing loan status,
p_i: Predicted probability that customer i will subscribe to the term deposit,
c_i: Estimated cost to recall customer i, as a function of last contact duration, previous campaign outcome, days since last contact, etc,
H: Human-resource constraint, which is the maximum number of customers that can be reached in one period.

Assumptions & Limitations

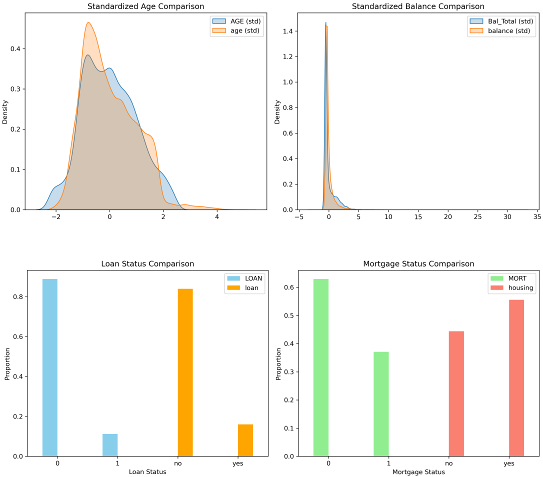
- Assume all revenue during the period is generated from term deposit subscriptions
- The effect of interest rate fluctuations on revenue is negligible
- Estimated contact cost: $c_i = g(\text{duration}_i, \text{poutcome}_i, \text{pdays}_i, \ldots) \approx p_i$
- Human resource constraint: average monthly contact limit (3768)
- Additional constraints to consider: behavioral risk, early withdrawal risk, liquidity risk, etc.

Revenue Estimation

- Introduce a new dataset:
  This dataset comes from the case study in *Building Better Models with JMP® Pro* (Grayson, Gardner & Stephens, SAS Press, 2015). It is based on real banking operations but has been anonymized and adapted for instructional demonstrations of model building.
  https://www.jmp.com/en/academic/case-study-library/bank-revenue
- Features:
  `Bal_Total` (sum of balances across all accounts), `AGE`, `LOAN` (has personal loan activity), `MORT` (has mortgage activity), ~~`CARD, INSUR, Check, AccountAge, ...`~~
- Target: `Rev_Total`, which indicates total revenue generated per customer over six months
- Pros: Moderate sample size (7,420), high quality, and no privacy concerns
- Cons: Rich features but limited overlap, processed for teaching, and lacks detailed scenario context

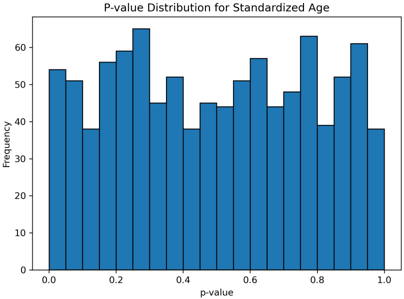
Population Consistency Test

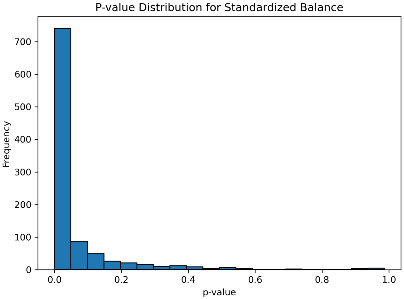
- Numerical features: `AGE` and `age`, `Bal_Total` and `balance`
- Binary features: `LOAN` and `loan`, `MORT` and `housing`



Population Consistency Test

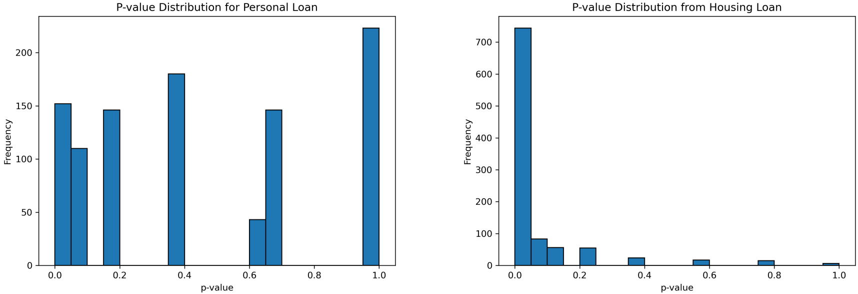
- Numerical features
  - Mann–Whitney U Test: a nonparametric test; significance level $\alpha = 0.05$
  - Sensitive to large sample sizes: mitigated by repeated subsampling (sample size $n = 50$, 1,000 trials)
  - Average p-value for standardized age: 0.4953 $>$ 0.05, indicating no significant difference
  - Average p-value for standardized balance: 0.0647 $>$ 0.05, indicating no significant difference





Population Consistency Test

- Binary features
  - Chi-Square Test: between two categorical variables; significance level $\alpha = 0.05$
  - Sensitive to large sample sizes: mitigated by repeated subsampling (sample size $n = 50$, 1,000 trials)
  - Average p-value for personal loan: 0.4555 $>$ 0.05, indicating no significant difference
  - Average p-value for housing loan: 0.0725 $>$ 0.05, indicating no significant difference

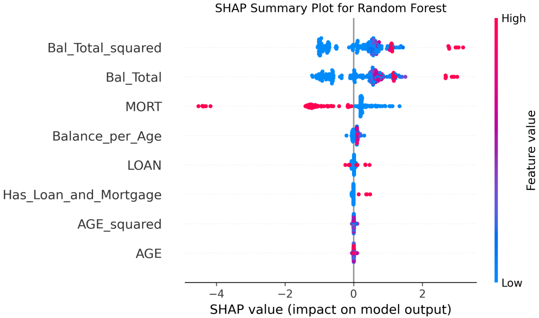


Random Forest Regression

- Why Random Forest over XGBoost?
  - Fewer hyperparameters, making it easier to tune
  - More robust to overfitting on moderate-sized data
  - Provided better generalization performance in our experiments
- Features (8 total): 4 raw predictors ($\texttt{AGE}$, $\texttt{Bal\_Total}$, $\texttt{LOAN}$, $\texttt{MORT}$) + 4 engineered features ($\texttt{AGE}^2$, $\texttt{Bal\_Total}^2$, $\texttt{Bal\_Total}/\texttt{AGE}$, $\texttt{LOAN} \times \texttt{MORT}$)
- 80/20 train-test split (stratified).
- Model: `RandomForestRegressor` with 300 trees and $\texttt{max\_depth} = 10$; hyperparameters tuned using grid search with 5-fold cross-validation

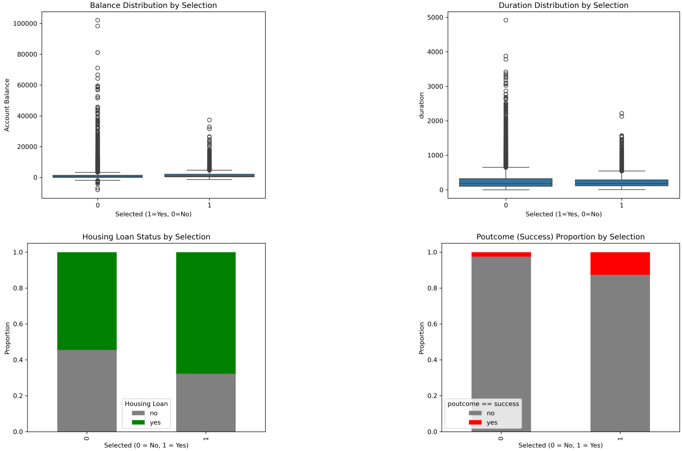
Random Forest Regression

- Test set results:
  - MSE: 6.2405
  - R^2: 0.2828
- Top 3 feature importances:
  - `Bal_Total_squared`: 0.406914
  - `Bal_Total`: 0.405778
  - `MORT`: 0.084732



Optimization Strategy

- Formulated as a **binary integer linear program**
- Optimization solved using **PuLP solver**
- To **prevent data leakage**, customers who have already subscribed are excluded.



Interpretation

- Customers who converted in previous campaigns are more likely to subscribe again.
- Recent contact increases subscription likelihood.
- Customers with loans are less likely to subscribe and less profitable, though this depends on other features.
- High balance customers are more profitable, but extremely high balances reduce subscription rates.
- Moderate call duration is optimal: longer calls raise both subscription likelihoods and costs.
- Time of contact matters: better responses occur at quarter-ends, during tax season (Apr–Jun, Portugal), and early or late in the month.

Implementation Guideline

- **Strategy Execution**
  - Embed the predictive model (DNN/XGBoost) into the bank's CRM system to score clients in real time
  - Run the optimization module periodically (e.g., weekly) to select high-value clients for outreach
  - Push selected clients to the marketing call system for follow-up
- **Ongoing Monitoring**
  - Track model performance using key metrics
  - Retrain the model monthly with recent campaign data
  - Apply SHAP and feature importance tools to ensure transparency and regulatory compliance
- **Special Considerations**
  - Monitor and address class imbalance to ensure fairness
  - Review thresholds quarterly to optimize outreach effectiveness
  - Ensure compliance with financial sector regulations

Conclusion

- **Logistic Regression:**
  - Interpretable and easy to implement
  - Suffers from low recall
- **XGBoost & Deep Neural Network:**
  - Strong performance on the test set
  - SHAP improves interpretability
  - Future improvements: Bayesian tuning, SMOTE, and ensemble methods
- **Decision Optimization:**
  - Considers profits, costs, and human resource constraints
  - Dependent on data quality and prediction accuracy
  - Limited in modeling complex feature interactions