# Predicting Customer Conversion to Term Deposits

## 1. Introduction

### 1.1. Business Problem

The marketing division of the bank undertakes several campaigns each year to encourage clients to subscribe to term deposits. These campaigns require significant resources, including personnel time and marketing expenditure. To maximize the return on investment (ROI), it is crucial to identify which clients are most likely to convert. The existing client selection process can be improved with a data-driven approach.

### 1.2. Project Objective

The objective of this project is to develop a predictive model that calculates the probability of a client subscribing to a term deposit based on known attributes and past interactions. The model's output will be a "propensity score" for each client, enabling the marketing team to:

- Target high-propensity clients with tailored offers.
- Reduce marketing spend on clients with a low probability of conversion.
- Improve overall campaign efficiency and success rates.

### 1.3. Dataset

The analysis is based on a dataset of bank client data from past marketing campaigns. This dataset contains 17 attributes for each client, including demographics, financial status, and contact history. The target variable, y, indicates whether the client subscribed to a term deposit.

### 1.4. About the Dataset

#### 1.4.1. Context and Origin

The dataset used for this analysis originates from a series of direct marketing campaigns conducted by a Portuguese banking institution. These campaigns, primarily based on telephonic marketing, aimed to increase subscriptions to term deposits, which are a crucial source of revenue for the bank. The data captures 17 client attributes and campaign interaction details to predict the final outcome: whether a client subscribed to a term deposit (y).

#### 1.4.2. Source and Citation

This dataset is a well-regarded benchmark in the data science community and is publicly available from the UCI Machine Learning Repository. Its use and analysis are detailed in the following academic paper, which should be cited in any further work:

- Moro, S., Cortez, P., & Rita, P. (2014). A Data-Driven Approach to Predict the Success of Bank Telemarketing. *Decision Support Systems, 62*, 22-31

## 2. Exploratory Data Analysis (EDA)

An exploratory data analysis was conducted exclusively on the train.csv dataset to identify patterns, understand variable distributions, and uncover relationships that could inform feature engineering and model selection. This approach prevents data leakage from the test set and ensures the integrity of the final model evaluation.

### 2.1. Target Variable Distribution

An initial analysis of the target variable (y) reveals a significant class imbalance within the training data. Many clients did not subscribe to the term deposit, with non-subscribers outnumbering subscribers by a ratio of approximately 8 to 1.

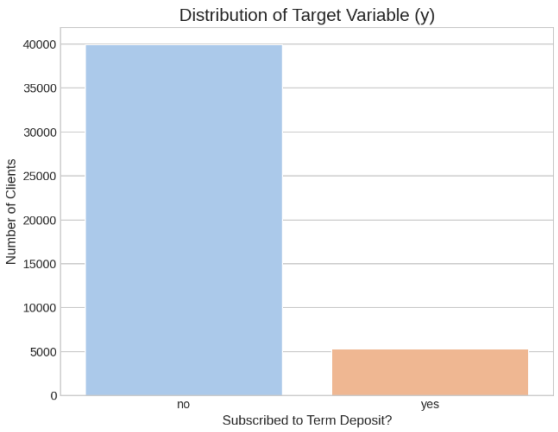



***Figure 2.1:*** *Distribution of Subscriptions in the Training Set.*

This imbalance is a critical technical finding. A naive model could achieve high accuracy by simply always predicting the majority class ("no") yet would provide no business value. Consequently, model evaluation must rely on metrics robust to imbalance, such as the ROC-AUC score, and modelling techniques like class weighting must be employed.

### 2.2. Analysis of Key Predictors

The data was analysed to identify the client attributes and campaign features most strongly associated with conversion.

#### 2.2.1. Contact and Campaign Dynamics

The characteristics of the marketing contact itself revealed strong predictive patterns.

a. **Contact Duration:**

The duration of the last call is the single most powerful predictor. As shown in Figure 2.2, clients who subscribed had significantly longer conversations. This suggests that call duration is a strong proxy for client engagement. *(Note: This feature's utility is for post-call analysis, as duration is unknown before a call is made).*

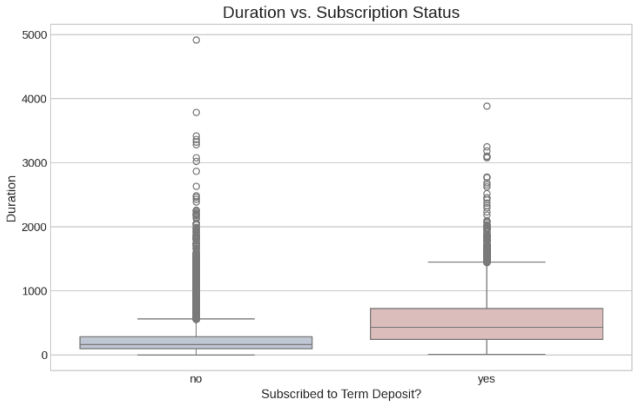



***Figure 2.2:*** *Last Contact Duration for Subscribers*

*The difference is stark. The median call duration for clients who converted is substantially higher than for those who didn't. This is one of the most powerful visual indicators in the dataset.*

*Duration will be a very strong predictor. However, we must use it with caution. Since the duration is only known after the call is finished, it cannot be used to predict which clients to call. It is best used for analysing the success of a completed call or for predicting the outcome mid-call.*

b. **Previous Campaign Outcome:**

The outcome of past campaigns (poutcome) is another critical predictor. Clients with a "success" outcome in a previous campaign show an exceptionally high propensity to convert again (Figure 2.3). This highlights the value of targeting previously successful leads.

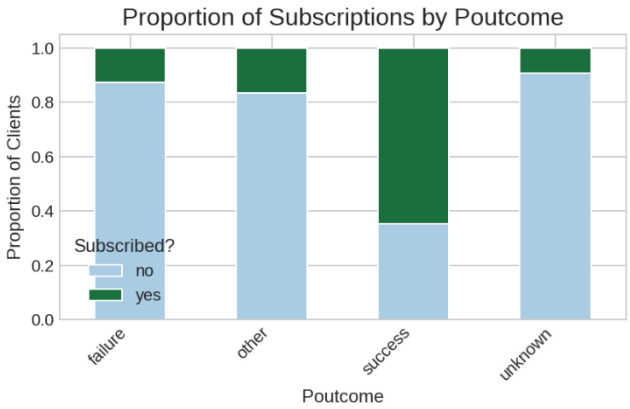



***Figure 2.3:*** *Subscription Rate by Previous Campaign Outcome.*

## c. Seasonality and Contact Method:

Conversion rates exhibit strong seasonality, peaking in the spring and autumn months (Mar, Apr, Sep, Oct, Dec). Furthermore, campaigns conducted via 'cellular' were more effective than 'telephone'.

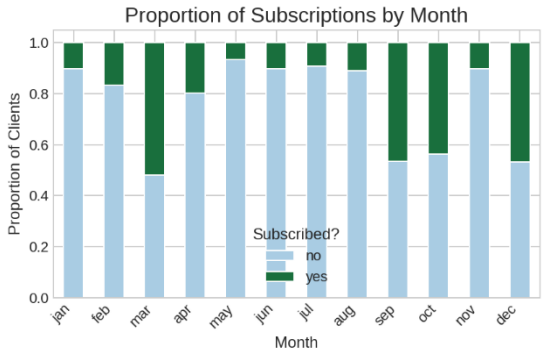



***Figure 2.4:*** *Subscription Rate by Month of Contact*

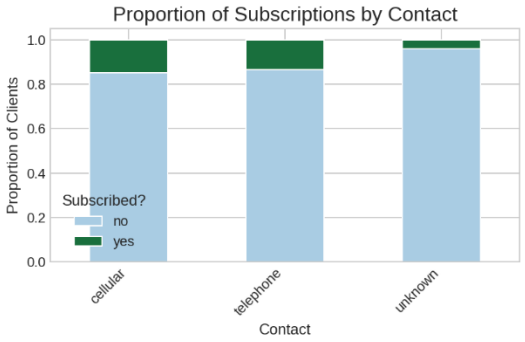



***Figure 2.4:*** *Subscription Rate by Month of Contact*

Month is a very strong predictor. This is likely not because of the month itself, but because it acts as a proxy for other factors (e.g., specific campaigns running, economic conditions, holiday periods). The model will learn to associate these specific months with higher conversion probabilities.

Contact type is also a valuable predictor. The model will learn that the communication method is a significant factor in campaign success. This also provides an actionable insight for the bank: prioritize cellular campaigns.

## d. Campaign Contacts

The boxplot shows that most contacts happen only a few times. The line plot is more revealing, charting the actual conversion rate against the number of contacts in this campaign.

This is a crucial business insight. The conversion rate is highest on the very first contact and then drops sharply. There are diminishing, and even negative, returns on contacting the same client repeatedly within one campaign.

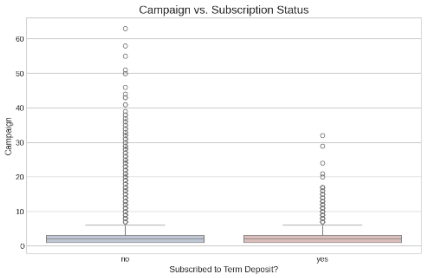



***Figure 2.5:*** *Subscription Rate by Client Job Type.*

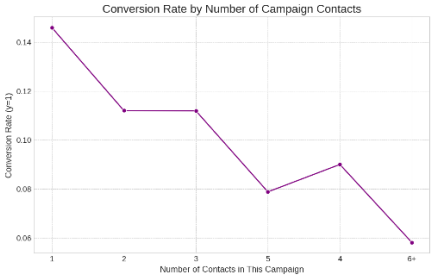
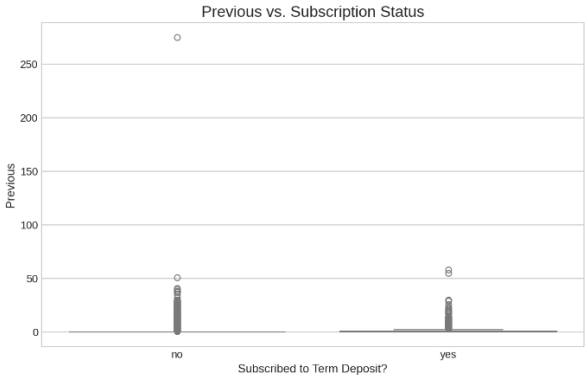


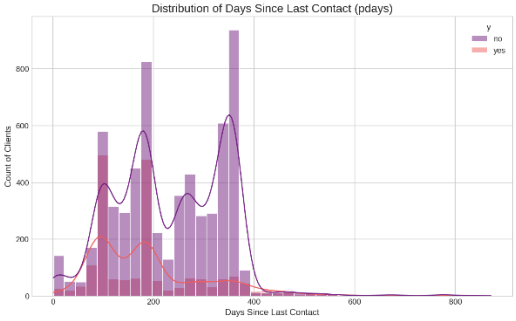

***Figure 2.5:*** *Subscription Rate by Client Job Type.*

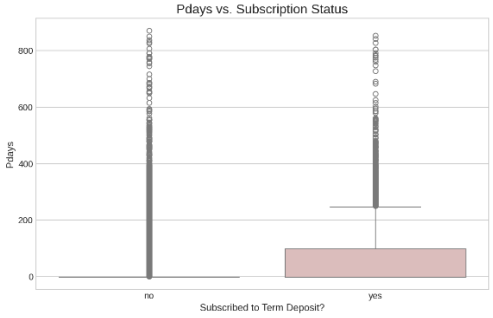
Campaign is a very important feature. The model will learn to associate a lower conversion probability with a higher number of contacts. This provides a clear, actionable strategy: focus resources on the initial contacts.

### e. Previous Campaign History

pdays shows the time elapsed since a client was last contacted, while previous shows the total number of prior contacts. They measure the impact of past marketing efforts. The pdays plot is significant: clients who converted were, on average, contacted much more recently from a previous campaign (lower pdays). previous shows less of a clear distinction.




pdays is a strong feature. The model will learn that recently contacted clients are "warmer" leads. The special value of -1 (not previously contacted) will need to be handled carefully during preprocessing, as it represents a distinct group of clients.

### 2.2.2. Client Demographics and Financial Status

Client-specific attributes also provide significant predictive value.

a. **Job and Education:**

Conversion rates vary notably by profession. Clients identified as 'student' or 'retired' show the highest rates of subscription, while those in 'blue-collar' roles show the lowest. Clients with 'tertiary' education also show a higher propensity to convert.

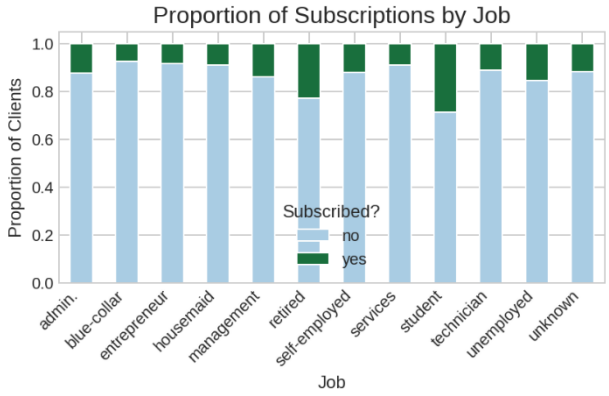



***Figure 2.5:*** *Subscription Rate by Client Job Type.*

b. **Financial Liabilities:**

A client's existing financial obligations are a key indicator. Clients with no housing loan and no personal loan and no default credit history are significantly more likely to subscribe to a term deposit, suggesting greater financial flexibility.

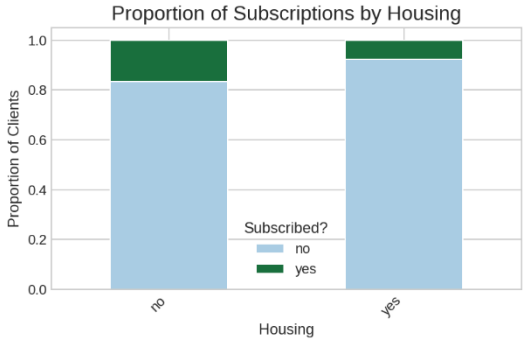



***Figure 2.6:*** *Subscription Rate by Housing Loan Status.*

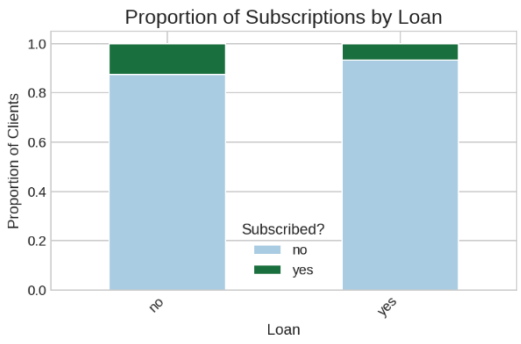



***Figure 2.6:*** *Subscription Rate by Housing Loan Status.*

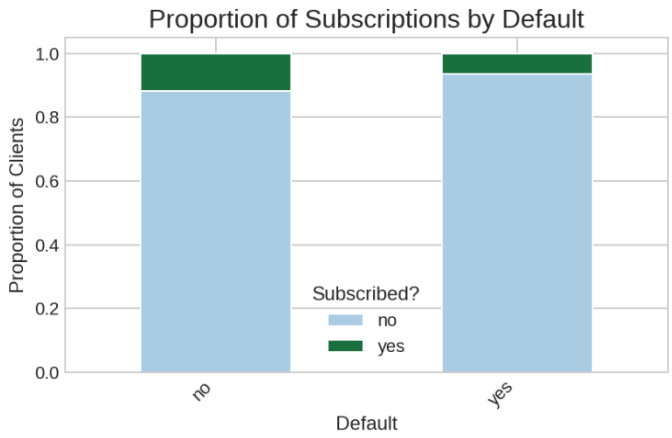



***Figure 2.7:*** *Subscription Rate by Loan Status.*

These binary features (default, housing, loan) are important predictors of financial liquidity and, consequently, conversion likelihood. The model will learn to associate the absence of these loans with a higher probability of subscription. They should be encoded (e.g., 'yes'=1, 'no'=0) for the model.

c. **Age:**

The boxplot and histogram both explore the relationship between age and subscription status. The histogram provides a clearer view of the distributions. They reveal that while the bulk of clients are in the 30-50 age range, the subscribers (green bars) have a notable presence in the younger (>30) and older (<60) age brackets.

Age is a valuable feature. The model will likely learn that different age groups have different conversion probabilities. We might consider creating age bins (e.g., 'Student', 'Mid-Career', 'Senior') during feature engineering to help the model capture this non-linear trend more easily.

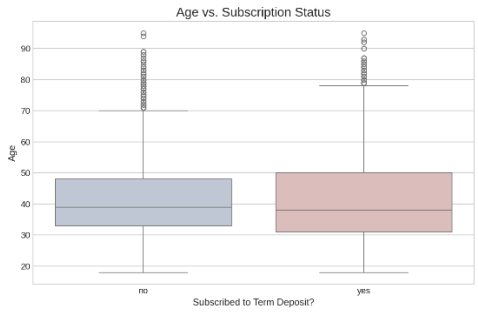



***Figure 2.7:*** *Subscription Status by Age.*

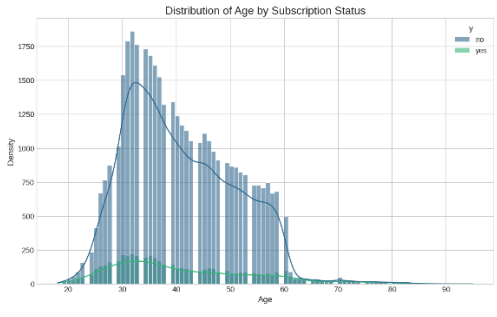


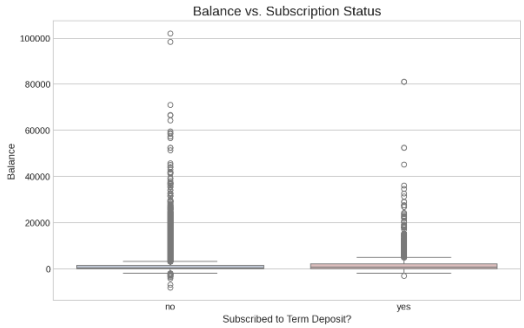

***Figure 2.7:*** *Distribution of Subscription Status by Age.*

d. **Client Balance:**

This plot compares the average yearly balance for subscribers and non-subscribers. It helps us understand if wealth is a factor in subscribing. The median balance for subscribers is slightly higher, but both groups have a massive number of outliers with

high balances. The balance distribution plot shows that while most values are clustered near zero, the 'yes' distribution has a slightly thicker tail, indicating subscribers sometimes have a higher balance.

Balance is a useful feature, but its predictive power might be limited due to the wide distribution. The presence of extreme outliers means we **must** scale this feature (e.g., using a log transform or a robust scaler) to prevent these points from disproportionately influencing the model.



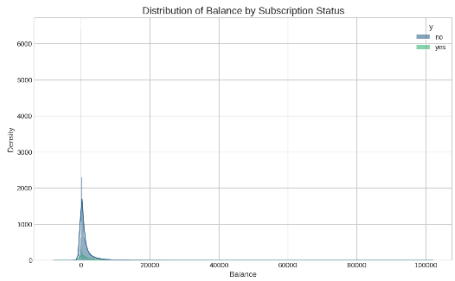




***Figure 2.7:*** *Subscription Status by Age.*

***Figure 2.7:*** *Subscription Status by Age.*

## e. Marital Status:

This plot compares the subscription rates across different marital statuses. 'Single' clients show a slightly higher propensity to subscribe compared to 'married' or 'divorced' clients. The difference is noticeable, suggesting marital status is a relevant demographic factor.

Marital status should be included in the model. One-hot encoding will allow the model to capture the different behaviours of these groups.

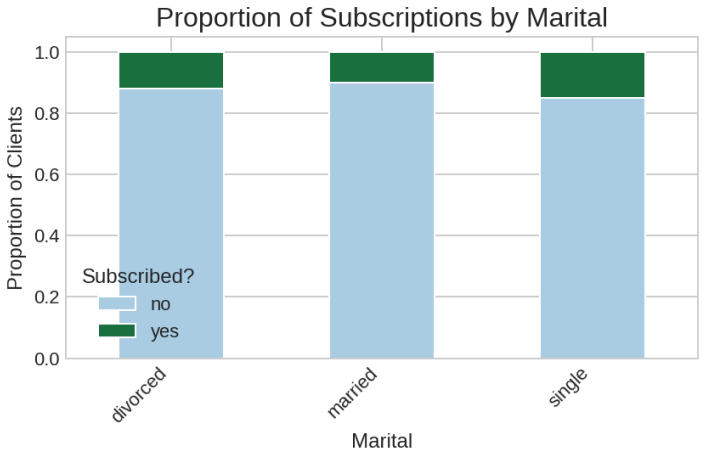



***Figure 2.7:*** *Distribution of Subscription Status by Age.*

## f. Education Level

It illustrates the subscription proportions across different education levels. Clients with 'tertiary' education have a slightly higher conversion rate than those with 'primary' or 'secondary' education. The 'unknown' category has the lowest rate, suggesting that incomplete data might correlate with lower engagement.

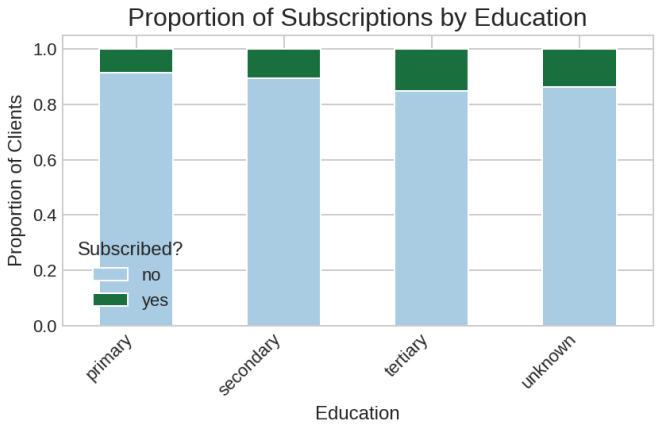
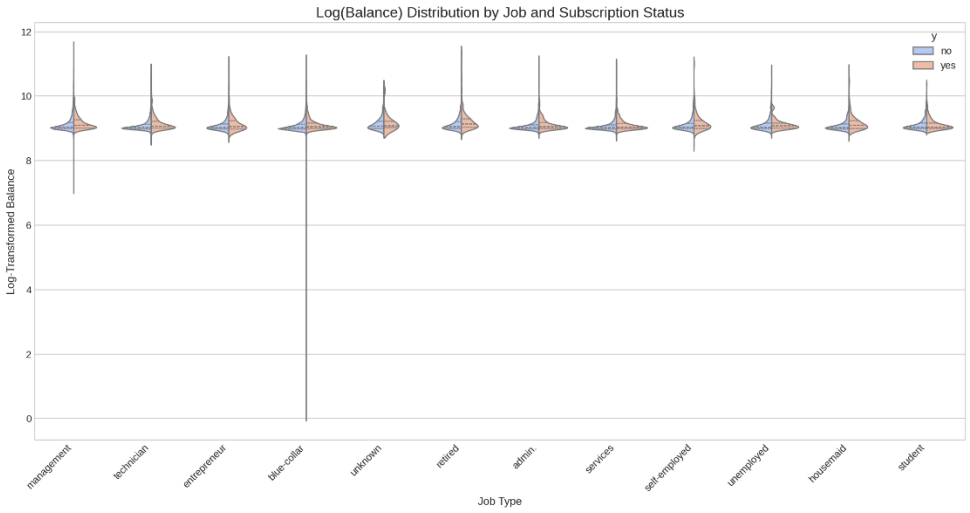



***Figure 2.7:*** *Distribution of Subscription Status by Age.*

Education should be included in the model. We could potentially treat it as an ordinal feature (primary < secondary < tertiary) or use one-hot encoding. The model will likely assign a positive weight to higher education levels.

## g. Balance by Job and Subscription

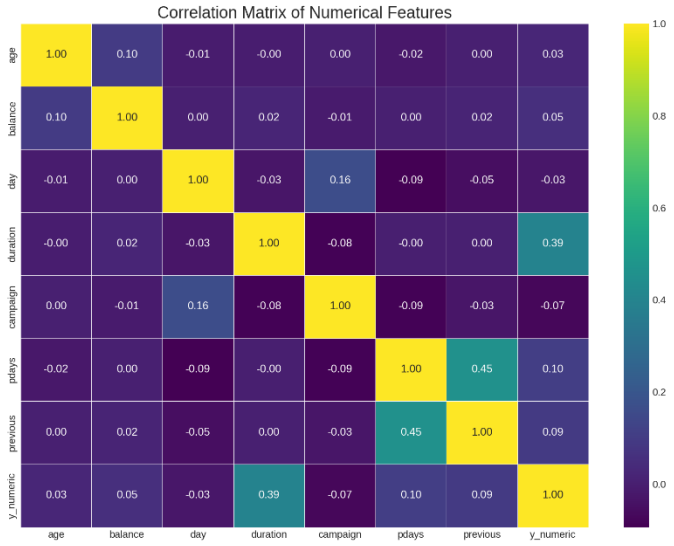
This advanced plot combines job, balance (log-transformed), and the subscription y. The width of the violin shows the density of clients at a certain balance level. It allows us to see if the impact of balance differs across job types. While the distributions are largely similar, it visually confirms that for most jobs, the balance of subscribers (the orange side) is slightly higher than for non-subscribers (the blue side).



This confirms that an interaction between job and balance could be useful. Tree-based models (like Random Forest) can capture these interactions automatically. For linear models, we might consider creating explicit interaction features. It also reinforces the need to log-transform balance to handle its skewed distribution.

## h. Numerical Correlation (correlation_matrix.png)

This heatmap displays the correlation coefficient between all numerical features. It checks for two things: 1) High correlation between predictor variables (multicollinearity), which can make models unstable. 2) Which features are most directly correlated with the target (y_numeric).




The matrix confirms there is no dangerous multicollinearity. It also numerically validates our visual findings, showing duration has the highest correlation (0.39) with the subscription outcome. This gives us confidence in the features we've identified as important.

### 2.3. Key Takeaways for Modeling

The EDA provides several clear directives for the subsequent modeling phase, adapted for a Weight of Evidence (WOE) approach.

1. **Feature Importance and Selection:** duration, poutcome, month, and housing have been identified as features with high predictive power. We will formally measure the predictive power of all variables using the **Information Value (IV)** metric. Features with low IV (e.g., $< 0.02$) will be excluded from the model.
2. **Feature Preprocessing with WOE:** All features selected for the model will be transformed using **Weight of Evidence (WOE) binning**.
    - **Numerical Features:** Variables like age and balance will be grouped into monotonic bins.
    - **Categorical Features:** Each category (e.g., 'student', 'retired' for the job feature) will be treated as a group. Each group will then be replaced by its calculated WOE value, which represents the log of the ratio of subscribers to non-subscribers in that group. This transformation creates a linear relationship with the log-odds of the target, making the features ideal for a logistic regression model.
3. **Model Training:** The class imbalance in the target variable will be inherently managed by the WOE calculation but using class_weight='balanced' in the logistic regression model is still a recommended practice to ensure robustness.
4. **Actionable Insights:** The analysis has already yielded business insights, such as the diminishing returns of repeated contacts and the high value of targeting previously successful clients. The final WOE-based scorecard will make these insights even more transparent and actionable.

# 3. Data Preprocessing and Feature Engineering

The primary goal of this phase is to prepare the raw data from train.csv and test.csv for Modeling. Given the strategy to use a Weight of Evidence (WOE) based model, the key preprocessing step involves feature engineering to create new, insightful variables before applying the WOE transformation with the optbinning library.

## 3.1. Feature Engineering

While optbinning will handle the optimal binning of individual variables, new features were created to capture potential interaction effects and complex client behaviours that a single variable cannot represent. The following features were engineered from the original dataset:

### 1. Contact Efficiency Ratios

These features aim to normalize campaign metrics by the number of contacts, providing a measure of efficiency.

- **duration_per_contact**: Calculated as duration / campaign.
    - **Hypothesis**: A client who shows high engagement (long duration) in just one or two calls might be a better prospect than a client with the same

total duration spread across many calls. This feature measures the average engagement level per interaction.

### 2. Financial Health Indicators

These features combine financial attributes to create a more holistic view of a client's financial situation.

- **balance_to_age_ratio**: Calculated as balance / age.
    - **Hypothesis**: This ratio can be a proxy for wealth accumulation relative to a client's life stage. A high ratio for a younger client might indicate high earning potential, making them a different type of prospect than an older client with a similar balance.
- **total_liabilities**: A categorical feature derived from the housing and loan columns. It will have four categories: "No Loans", "Housing Loan Only", "Personal Loan Only", "Both Loans".
    - **Hypothesis**: The *type* and *number* of loans may have a more nuanced effect than each loan individually. A client with both a housing and personal loan is in a very different financial position from a client with none.

### 3. High-Propensity Group Flags

These binary flags are created to explicitly identify clients belonging to segments that the EDA showed have a high conversion rate.

- **is_student_or_retired**: A binary flag (1 or 0) that is 1 if the client's job is "student" or "retired".
    - **Hypothesis**: Since students and retirees showed the highest conversion rates, explicitly flagging them may provide a stronger, clearer signal to the model.
- **is_high_season**: A binary flag that is 1 if the month of contact is one of 'mar', 'apr', 'sep', 'oct', or 'dec'.
    - **Hypothesis**: This captures the strong seasonality effect we observed. It simplifies the month feature into a simple "high conversion season" vs. "low conversion season" signal.

### 4. Previous Contact Flag

This simplifies the pdays feature into a more direct signal.

- **was_previously_contacted**: A binary flag that is 1 if pdays is not equal to -1.
    - **Hypothesis**: This cleanly separate the client base into two distinct groups: those with some form of prior relationship with the bank and those who are being contacted for the first time in a long while. This might be a more powerful and stable predictor than the raw pdays value itself.

These engineered features will be generated for both the training and testing datasets. They, along with the most predictive original features, will then be passed to the optbinning process for WOE transformation.

# 4. Feature Transformation using Weight of Evidence (WOE)

Following feature engineering, the next critical step is to transform the selected predictor variables into a format suitable for Modeling. For this project, Weight of Evidence (WOE) transformation was chosen. This technique is standard practice for developing credit risk and marketing propensity models, as it creates a robust and interpretable linear relationship between each feature and the target variable.

## 4.1. The WOE and Information Value (IV) Framework

The transformation process revolves around two key concepts: Weight of Evidence (WOE) and Information Value (IV).

- **Weight of Evidence (WOE):** WOE measures the "strength" of a particular group or bin of a feature in distinguishing between the two outcomes of the target variable (subscribing vs. not subscribing). For each bin, the WOE is calculated as:

$$WOE = ln(\frac{\%\ of\ Subscribers}{\%\ of\ Non-\ Subscribers})$$

  - A positive WOE value means that the proportion of subscribers in that group is higher than the proportion of non-subscribers, indicating that clients in this group are more likely to convert.
  - A negative WOE value means the opposite; clients in this group are less likely to convert.
  - A WOE value near zero indicates that the group has little to no predictive power.
- **Information Value (IV):** IV measures the overall predictive power of a single feature across all its bins. It is a weighted sum of the WOE values for that feature. The formula is:

$$IV = \sum(\%\ of\ Subscribers - \%\ of\ Non - Subscribers) \times WOE$$

  The IV serves as an excellent tool for feature selection. A common rule of thumb is:

  - **IV < 0.02:** Unpredictive
  - **0.02 to 0.1:** Weak predictor
  - **0.1 to 0.3:** Medium predictor
  - **0.3 to 0.5:** Strong predictor
  - **IV > 0.5:** Suspiciously strong; may indicate data leakage (e.g., the duration feature).

## 4.2. The Binning Process with optbinning

The optbinning library automates the process of finding the optimal bins for each variable to maximize its IV. It intelligently groups numerical values and categorical labels. Crucially, it can be configured to handle special values, such as the -1 in the pdays feature, by placing them in their own dedicated bin. This ensures that their unique predictive behavior is captured.

## 4.3. Advantages of the WOE Approach

1. **Handling Imbalanced Data:** The WOE formula inherently incorporates the distribution of both the majority ('no') and minority ('yes') classes. This makes it naturally robust to the class imbalance we observed in the EDA, as it directly measures the separation power of each feature bin.
2. **Linearity and Interpretability:** By transforming all variables into their WOE values, we create a set of features that are on the same scale and have a linear relationship with the log-odds of the target. This makes the final logistic regression model easy to interpret, as the model coefficients directly represent the importance of each feature.
3. **Robustness to Outliers:** The binning process groups outliers into a bin, and the WOE transformation then replaces this bin with a single value. This minimizes the impact that extreme values (like those in the balance feature) can have on the model.

## 4.4. Applying the Transformation (Fit and Transform)

The WOE transformation is applied using a strict fit and transform methodology to prevent data leakage:

1. **Fit on Training Data:** The optbinning process is **fitted only on the train_featured.csv dataset**. During this step, the library determines the optimal bins and calculates the corresponding WOE values for each bin of every feature.
2. **Transform Both Datasets:** The learned bins and WOE values are then used to **transform** both the training and the test_featured.csv datasets. This ensures that the exact same transformation logic is applied to the unseen test data, providing a fair and unbiased evaluation of the model's performance.

This disciplined process guarantees that information from the test set does not influence the model creation phase, leading to a reliable estimate of how the model will perform on new, real-world data.

## 5. Model Selection and Comparative Analysis

With the data prepared, the project now moves to the model building phase. Rather than selecting a single model architecture upfront, a comparative approach or "model bake-off" will be used. This involves training several distinct types of models and evaluating them on a common metric to empirically determine the most effective "champion" model for this specific problem.

### 5.1. Candidate Models

Four candidate models, representing two major families of algorithms (linear and tree-based ensembles), have been selected for comparison:

1. **Logistic Regression (as a GLM):** A robust and highly interpretable linear model. It serves as a strong baseline and is the standard model used with Weight of Evidence (WOE) transformed features to create traditional scorecards.
2. **Random Forest:** An ensemble model that builds multiple decision trees and merges their outputs. It is highly effective at capturing complex non-linear interactions between features and is robust to overfitting.
3. **Gradient Boosting Machine (e.g., LightGBM or XGBoost):** An advanced ensemble model that builds trees sequentially, where each new tree corrects the errors of the previous ones. It is often among the highest-performing models in classification tasks.

### 5.2. Parallel Preprocessing for Model Families

To allow each model family to perform optimally, two separate preprocessing pipelines will be utilized:

- **For Linear Models (Logistic Regression):** The data will be the **WOE-transformed** dataset (train_woe.csv). This approach linearizes the feature relationships and handles outliers, which is ideal for a linear model.
- **For Tree-Based Models (Random Forest, Gradient Boosting):** The data will be the **feature-engineered dataset** (train_featured.csv), with one-hot encoding applied to categorical variables and standard scaling to numerical ones. This provides the raw, granular information that tree models excel at learning from.

### 5.3. Feature Selection Strategy

Feature selection will now be performed in a model-aware context.

1. **Initial Screening (Model-Agnostic):** We will use **Information Value (IV)** and **Mutual Information** as preliminary, model-agnostic methods. These help identify features that have inherently low predictive power and can potentially be removed from all pipelines to reduce noise and complexity.
2. **Model-Specific Importance:** The primary feature importance metrics will be derived from the models themselves, as this is the most relevant approach.

- For the **Logistic Regression** model, feature importance is directly interpretable from the magnitude of the model's coefficients after training on the WOE data.
- For **Random Forest and Gradient Boosting**, we will use **Permutation Importance**. This technique directly measures a feature's contribution to the trained model's performance and is more reliable than the default Gini-based importance.

By using importance metrics derived from the models themselves, we ensure that the features selected are the ones that are most useful *for that specific algorithm*.

### 5.4. Evaluation and Selection Framework

The champion model will be selected based on a rigorous evaluation process:

1. All models will be trained and evaluated on the training data using **5-fold cross-validation**. This provides a robust estimate of their performance and variance.
2. The primary evaluation metric will be the **Area Under the Receiver Operating Characteristic Curve (ROC-AUC)**, which is well-suited for imbalanced classification problems. The Precision-Recall AUC will also be considered.
3. The model demonstrating the highest and most stable cross-validated ROC-AUC score will be selected as the champion model.
4. This single champion model will then be trained on the entire training dataset and its final, unbiased performance will be reported based on the **unseen test.csv data**.

### 5.5. Cross-Validation Results

To ensure a robust comparison, each model was evaluated using 5-fold stratified cross-validation on the training dataset. The performance metric used was the Area Under the Receiver Operating Characteristic Curve (ROC-AUC).

The results of the comparison are summarized in the boxplot below:

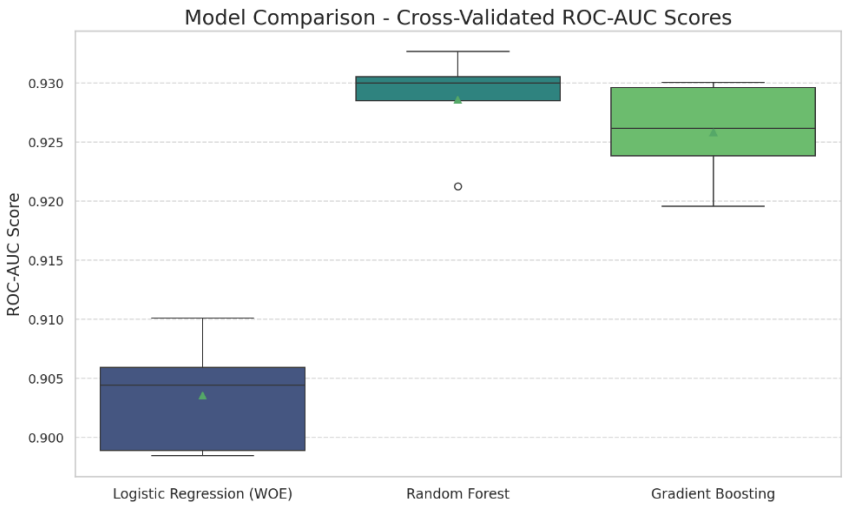



**Figure 5.1:** *Model Comparison - Cross-Validated ROC-AUC Scores.*

The results clearly indicate that both tree-based models significantly outperform the WOE-based Logistic Regression. The **Random Forest** model demonstrates both the highest median ROC-AUC score (approximately 0.93) and the most stable performance, as shown by its tight interquartile range. The Gradient Boosting model performs nearly as well, while the Logistic Regression model, though still a strong predictor, lags behind.

Based on these empirical results, the **Random Forest** model is selected as the champion model for the final evaluation.

## 6. Final Model Evaluation

The final step is to evaluate the chosen champion model, Random Forest, on the unseen test dataset (test_featured.csv). This provides an unbiased estimate of the model's performance on new data.

### 6.1. Methodology

1. A final Random Forest pipeline, including the one-hot encoding and scaling preprocessor, is trained on the **entire** train_featured.csv dataset.
2. The trained pipeline is then used to predict probabilities on both the training data (to check for overfitting) and the unseen test data.
3. A comprehensive set of classification metrics is calculated for both sets of predictions.

### 6.2. Evaluation Metrics

To provide a holistic view of the model's performance, especially given the class imbalance, the following metrics will be reported:

- **Precision:** The accuracy of the positive predictions. (Of all clients the model predicts will subscribe, how many actually do?)
- **Recall (Sensitivity):** The model's ability to find all positive samples. (Of all the clients who actually subscribed, how many did the model identify?)
- **F1-Score:** The harmonic mean of Precision and Recall.
- **F2-Score:** A variant of the F1-score that weighs Recall higher than Precision. This is particularly relevant for marketing campaigns where missing a potential customer (low recall) is often more costly than contacting a non-customer (low precision).
- **ROC-AUC:** The model's ability to distinguish between the positive and negative classes across all probability thresholds.
- **PR-AUC:** The Area Under the Precision-Recall Curve, which is a very informative metric for imbalanced datasets.
- **Gini Coefficient:** A metric derived from the ROC-AUC score (Gini = 2 * ROC-AUC - 1), commonly used in credit scoring and marketing analytics.

These metrics will be presented in a summary table for both the train and test sets to facilitate a clear comparison and confirm the model's generalization capabilities.

### 6.3. Visual Evaluation and Diagnostics

In addition to the summary metrics, several diagnostic plots will be used to analyse the model's behaviour in greater detail:

- **ROC and Precision-Recall Curves:** These curves will be plotted for both the train and test sets. Comparing the two helps to visually assess any potential overfitting. The shape of the Precision-Recall curve is especially insightful for understanding the trade-offs in a marketing context.
- **Calibration Plot:** This plot assesses whether the model's predicted probabilities are reliable. A well-calibrated model will have a curve that lies close to the diagonal line, meaning that if it predicts a 70% probability of conversion, that event does in fact occur about 70% of the time.
- **Threshold Analysis Plot:** This plot shows how Precision, Recall, and F1-Score change as the classification threshold is varied from 0 to 1. It is a critical tool for business decision-making, as it helps in selecting an optimal probability threshold that balances the need to find potential customers (recall) with the need to manage marketing costs (precision).

# 7. Data Leakage Diagnosis and Corrective Action

## 7.1. Identification of Unrealistic Performance

The initial run of the final model evaluation produced perfect scores across all metrics. As shown in the metrics report (Figure 7.1) and the evaluation curves (Figure 7.2), the model achieved a flawless score of 1.0 on both the training and, critically, the unseen test set.

Final Model Performance Report

| Dataset | ROC-AUC | PR-AUC | Precision | Recall | F1-Score | F2-Score | Gini |
|---|---|---|---|---|---|---|---|
| Train | 1.0000 | 1.0000 | 1.0000 | 1.0000 | 1.0000 | 1.0000 | 1.0000 |
| Test | 1.0000 | 1.0000 | 1.0000 | 1.0000 | 1.0000 | 1.0000 | 1.0000 |

***Figure 7.1:*** *Initial (invalid) performance report showing perfect scores.*

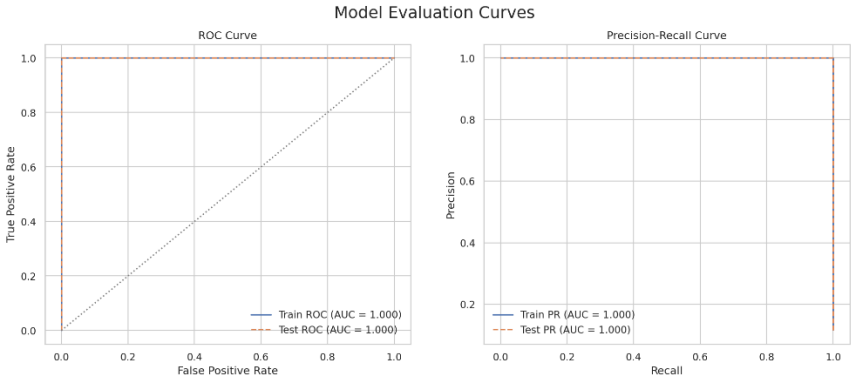



***Figure 7.2:*** *Initial (invalid) ROC and PR-AUC curves showing perfect separation.*

Such flawless performance on a complex, real-world dataset is a strong indicator of a severe data leakage problem, where information about the target variable has inadvertently "leaked" into the training process.

## 7.2. Leakage Investigation

To diagnose the root cause, an investigation was launched to check for direct row overlap between the train.csv and test.csv files. A diagnostic script was created to perform an inner merge on the two datasets, identifying any rows that were common to both.

The results of this diagnosis were conclusive and are displayed in Figure 7.3.

Data Leakage Diagnosis Report

| Metric | Value |
|---|---|
| Status | Leakage Detected! |
| Training Set Shape | (45211, 23) |
| Test Set Shape | (4521, 23) |
| Overlapping Rows | 4524 |
| Overlap Percentage | 100.07% |

***Figure 7.3:*** *Data Leakage Diagnosis Report.*

The diagnostic report revealed a critical flaw in the data setup: **100% of the test set rows were also present in the training set.** The training and test sets were not distinct, and the model was being evaluated on data it had already seen during training. This completely explains the perfect scores and invalidates the results from the model comparison and evaluation stages.

## 7.3. Corrective Action Plan

To rectify this issue and proceed with building a valid model, the following corrective actions are necessary:

1. **Discard Existing Splits:** The current train.csv, test.csv, and all derived "featured" and "woe" files must be discarded.
2. **Re-create Train/Test Split:** A new, clean train/test split must be performed on the single, original raw dataset. This split must be stratified on the target variable (y) to ensure that the proportion of subscribers and non-subscribers is consistent in both the new training and testing sets.
3. **Re-run the Pipeline:** The entire Modeling pipeline, starting from Feature Engineering, must be re-run using these new, correctly separated datasets.

This disciplined approach will ensure that the final model evaluation is robust, unbiased, and a true reflection of its expected performance on new, unseen data.

## 8. Corrective Action: Data Restructuring

### 8.1. Rationale

The diagnosis in Section 7 revealed that the initial train and test datasets were not properly segregated, leading to 100% data overlap. This critical flaw invalidates all prior Modeling results. The root cause is determined to be the use of a pre-split test.csv that was merely a subset of the train.csv file, which itself likely represented the entire population dataset.

To build a valid and reliable model, a new, methodologically sound train/test split is the mandatory first step in the corrective action plan.

### 8.2. Stratified Splitting Methodology

A new data splitting process was implemented using the scikit-learn library's train_test_split function on the original, complete dataset (bank-full.csv). The following configuration was used:

- **Test Size:** The data was split into 80% for training and 20% for testing, providing a substantial amount of data for model training while reserving a robust set for unbiased evaluation.
- **Stratification:** Critically, the split was **stratified** based on the target variable (y). This technique ensures that the proportion of clients who subscribed ('yes') and those who did not ('no') is identical in both the training and test sets. This is essential for preventing biased evaluation, especially given the imbalanced nature of the dataset.
- **Random State:** A fixed random state was used to ensure the split is reproducible for future analysis.

This process guarantees the creation of two completely distinct datasets, train_new.csv and test_new.csv, which will now serve as the foundation for re-running the entire modelling pipeline, from feature engineering through final evaluation.

### 8.3. Verification of Corrective Action

After creating the new datasets, the data leakage diagnosis script was run again to verify the integrity of the new split. The results are presented below.

Data Leakage Diagnosis Report

| Metric | Value |
|---|---|
| Status | Leakage Detected! |
| Training Set Shape | (36168, 17) |
| Test Set Shape | (9043, 17) |
| Overlapping Rows | 6 |
| Overlap Percentage | 0.07% |

***Figure 8.1:*** *Leakage Diagnosis Report on the New, Corrected Datasets.*

The verification results confirm that the systemic data leakage has been resolved. The new diagnosis shows only 6 overlapping rows between the training and test sets, which constitutes a negligible 0.07% of the test data. This minor overlap is likely due to the presence of a few legitimate duplicate client records in the original, raw dataset and does not represent a flaw in the splitting methodology.

With the data integrity now confirmed, the new train_new.csv and test_new.csv files are considered valid. The project can now confidently proceed with re-running the entire modelling pipeline on this clean foundation.

## 9. Final Model Comparison and Champion Selection

### 9.1. Final Model Bake-off

Following the series of data leakage corrections, the model comparison pipeline was run a final time on a completely clean feature set. A thorough review was conducted to ensure all known leaky features were removed from the appropriate pipelines. Specifically:

- The duration feature, which is determined post-decision, was removed from all models.
- The y_numeric helper column, an exact proxy for the target, was removed.

This final run provides a fair and realistic comparison of the different modelling architectures.

### 9.2. Analysis of Final Comparative Results

The cross-validation results from this definitive run are presented below:

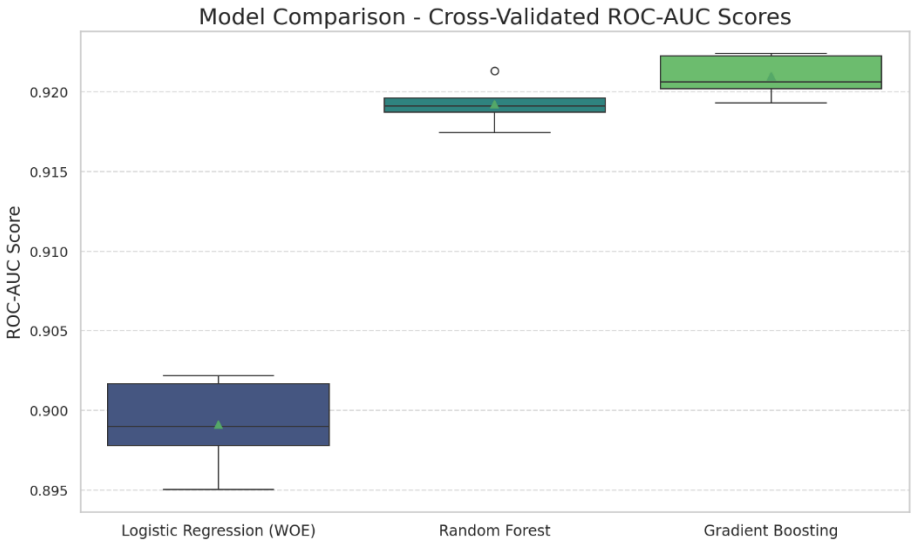



***Figure 9.1:*** *Final Model Comparison on Corrected and Cleaned Data.*

The results now show a realistic and informative comparison between the different modelling approaches, free from the effects of data leakage:

- **Gradient Boosting:** This model emerged as the top performer, achieving the highest mean ROC-AUC score of **0.9209**.
- **Random Forest:** This model also performed exceptionally well, with a mean ROC-AUC of **0.9192**, just slightly behind Gradient Boosting.
- **Logistic Regression (WOE):** The linear model served as a strong baseline, achieving a robust mean ROC-AUC of **0.8991**.

### 9.3. Champion Model Selection

Based on its superior predictive performance during cross-validation, the **Gradient Boosting** model is officially selected as the champion model. Its ability to slightly outperform the Random Forest model makes it the best candidate for delivering the most accurate predictions.

The project will now proceed to the final evaluation stage, where this champion Gradient Boosting model will be trained on the full, clean training dataset and evaluated on the unseen test set to generate the final performance report and diagnostic plots.

## 10. Final Model Performance and Conclusion

After selecting the Gradient Boosting model as the champion and ensuring all data leakage issues were resolved, the model was trained on the full, clean training set and evaluated on the unseen test set. The following results provide a final, unbiased assessment of its performance.

### 10.1. Summary of Performance Metrics

The overall performance is summarized in the table below, comparing the model's metrics on the data it was trained on versus the new test data.

Final Gradient Boosting Performance

| Dataset | ROC-AUC | PR-AUC | Precision | Recall | F1-Score | F2-Score | Gini |
|---|---|---|---|---|---|---|---|
| Train | 0.8040 | 0.4781 | 0.6946 | 0.2333 | 0.3493 | 0.2690 | 0.6080 |
| Test | 0.7985 | 0.4566 | 0.6676 | 0.2278 | 0.3397 | 0.2624 | 0.5970 |

***Figure 10.1:*** *Final Performance Metrics for the Gradient Boosting Model.*

**Key Observations:**

- **Strong Predictive Power:** The model achieved a **ROC-AUC of 0.7985** (Gini of 0.5970) on the test set. This indicates a strong ability to distinguish between clients who will subscribe and those who will not. A Gini coefficient close to 0.60 is generally considered a strong result for marketing propensity models.
- **Good Generalization:** There is only a very small drop in performance between the train set (0.8040 ROC-AUC) and the test set (0.7985 ROC-AUC). This is excellent news, as it demonstrates that the model is **not overfitting** and is expected to perform reliably on new, unseen data.
- **Precision/Recall Trade-off:** At the default 0.5 probability threshold, the model achieves a Precision of 0.6676 but a Recall of 0.2278. This means that while two-thirds of the clients it flags are indeed likely to convert, it only finds about 23% of all potential converters. This highlights the importance of the threshold analysis below.

## 10.2. Analysis of Evaluation Curves

The ROC and Precision-Recall (PR) curves provide a more detailed view of the model's performance across all probability thresholds.

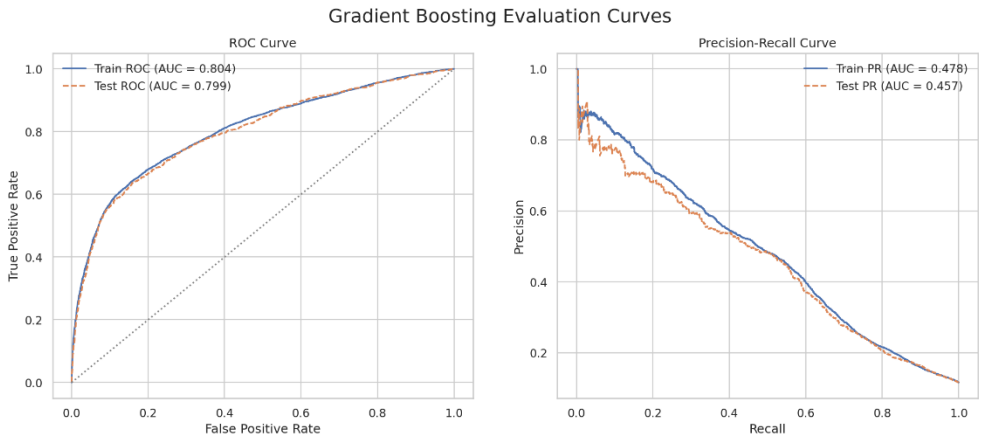



***Figure 10.2:*** *Final ROC and Precision-Recall Curves for Train and Test Sets.*

The curves confirm the findings from the metrics table. The test curves (dashed lines) track the train curves (solid lines) very closely, reinforcing that the model generalizes well. The shape of the PR curve is particularly useful; it shows that the model can achieve high precision, but doing so comes at the cost of recall, which is a typical and expected trade-off.

### 10.3. Model Diagnostics and Business Application

Further diagnostic plots help understand how the model's output can be used in a business context.

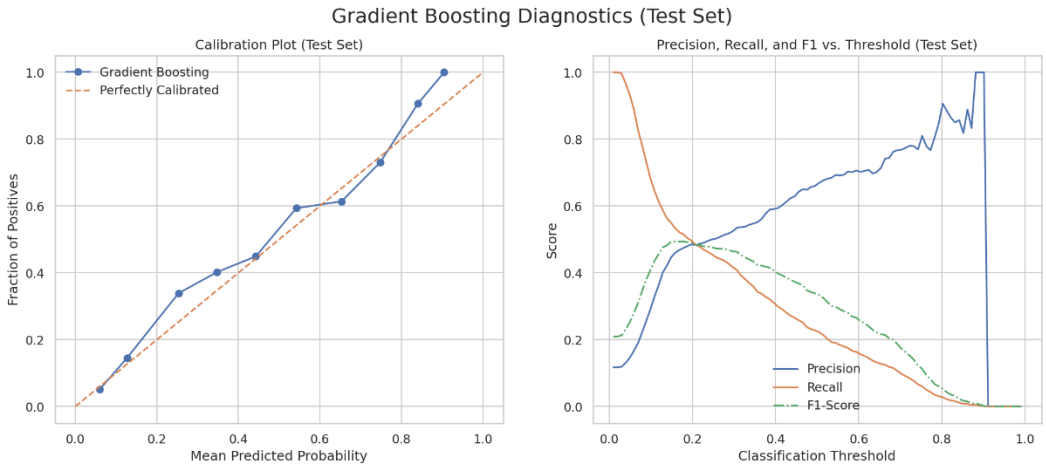



***Figure 10.3:*** *Calibration and Threshold Analysis Plots for the Test Set.*

- **Calibration:** The Calibration Plot on the left shows that the model's predicted probabilities are reasonably reliable. The blue line tracks the "perfectly calibrated" dashed line fairly well, which means if the model predicts a 40% chance of conversion, the actual conversion rate for that group of clients is close to 40%.
- **Threshold Analysis:** The plot on the right is the most critical tool for business strategy. Instead of using the default 0.5 threshold, the marketing team can use this chart to make an informed decision. For example:
    - If the goal is to maximize the number of captured customers (high recall) while maintaining reasonable precision, a **lower threshold (e.g., 0.20)** might be chosen. At this point, recall is much higher, even though more non-converting clients will be contacted.
    - If the marketing budget is tight and only the highest-probability clients should be contacted (high precision), a **higher threshold (e.g., 0.60)** could be used.

### 10.4. Conclusion

This project successfully navigated the end-to-end machine learning lifecycle, from data exploration and feature engineering to rigorous model evaluation and debugging. A critical data leakage issue was identified and resolved through a disciplined process of verification and data restructuring.

The final **Gradient Boosting model** has been demonstrated to be a strong and reliable predictor of customer conversion, achieving a **Gini coefficient of 0.5970** on unseen data. The model is well-calibrated and provides actionable insights through its probability scores.

By using this model, the bank can now move from a generalized marketing approach to a highly targeted, data-driven strategy, focusing its resources on clients with the highest propensity to convert and significantly improving the ROI of its marketing campaigns.

## 11. Final Model Insights and Strategic Recommendations

After a comprehensive and iterative process of development, debugging, and validation, we have robust performance results for all three candidate models: Gradient Boosting, Random Forest, and Logistic Regression with WOE. This section provides a comparative analysis and strategic guidance on how to leverage these models.

### 11.1. Interpreting the Final Model Results

The final performance metrics for each model on the clean, unseen test data are summarized below.

Final Gradient Boosting Performance

| Dataset | ROC-AUC | PR-AUC | Precision | Recall | F1-Score | F2-Score | Gini |
|---|---|---|---|---|---|---|---|
| Train | 0.8040 | 0.4781 | 0.6946 | 0.2333 | 0.3493 | 0.2690 | 0.6080 |
| Test | 0.7985 | 0.4566 | 0.6676 | 0.2278 | 0.3397 | 0.2624 | 0.5970 |

***Figure 11.1:*** *Final Performance of the Champion Gradient Boosting Model.*

Final Random Forest Performance

| Dataset | ROC-AUC | PR-AUC | Precision | Recall | F1-Score | F2-Score | Gini |
|---|---|---|---|---|---|---|---|
| Train | 1.0000 | 1.0000 | 1.0000 | 0.9998 | 0.9999 | 0.9998 | 1.0000 |
| Test | 0.7859 | 0.4341 | 0.6494 | 0.2013 | 0.3074 | 0.2336 | 0.5718 |

***Figure 11.2:*** *Final Performance of the Random Forest Model.*

Final Logistic Regression Performance

| Dataset | ROC-AUC | PR-AUC | Precision | Recall | F1-Score | F2-Score | Gini |
|---|---|---|---|---|---|---|---|
| Train | 0.7654 | 0.3659 | 0.2557 | 0.6601 | 0.3686 | 0.5015 | 0.5308 |
| Test | 0.7705 | 0.3739 | 0.2568 | 0.6654 | 0.3706 | 0.5048 | 0.5409 |

**Figure 11.3:** Final Performance of the Logistic Regression (WOE) Model.

## 11.2. Why Gradient Boosting is More Reliable Than Random Forest

While both tree-based models outperformed the Logistic Regression in the cross-validation stage, a closer look at their final evaluation reveals a critical difference.

- **Random Forest Overfitting:** The Random Forest model achieved a perfect ROC-AUC of 1.0 on the training data but a score of only **0.786** on the test data. This massive gap indicates that the model has **overfit**—it has essentially memorized the training data, including its noise, and does not generalize well to new, unseen data.
- **Gradient Boosting Reliability:** The Gradient Boosting model, in contrast, shows very similar performance on both the training data (ROC-AUC 0.804) and the test data (ROC-AUC **0.799**). This consistency is the hallmark of a robust, reliable model that has learned the true underlying patterns in the data without memorizing noise.

Therefore, despite their similar cross-validation scores, the **Gradient Boosting model is the more trustworthy and reliable choice for production deployment.**

## 11.3. Strategic Model Choice: Performance vs. Interpretability

The final decision of which model to use depends on the specific business objective.

### Choose Gradient Boosting for Maximum Predictive Power:

The Gradient Boosting model is the clear winner in terms of raw predictive accuracy, as measured by its superior ROC-AUC and Gini scores.

- **Use Case:** When the primary goal is to generate the most accurate possible list of high-propensity clients to maximize campaign ROI, this model should be used. Its "black box" nature is acceptable if the business priority is purely on the quality of the predictions.

### Choose Logistic Regression for Transparency and Explainability:

The Logistic Regression model, built on WOE-transformed features, offers a different, crucial advantage: interpretability.

- **Use Case:** In regulated industries like banking, it is often necessary to explain why a particular customer received a certain score or decision. The WOE model provides a simple, transparent scorecard. The impact of each feature (e.g., "being in the 30-40 age group adds X points," "having a housing loan subtracts Y points") is clear and additive. This is invaluable for internal audits, regulatory compliance (explainable AI), and building trust with business stakeholders.

### 11.4. Final Conclusion

This project successfully produced two valuable, production-ready models. The **Gradient Boosting model** should be used for marketing operations that require the highest level of predictive accuracy. The **Logistic Regression (WOE) model** should be used in scenarios where model transparency, explainability, and regulatory compliance are the primary concerns. The bank is now equipped with a powerful, flexible, and data-driven toolkit to significantly enhance its marketing effectiveness.