Blended Intensive Program (BIP)

Machine Learning: Mathematical aspects,

techniques, and applications

2nd Edition

Bank Marketing Dataset

Classification Project Report

Authors: Jakub Korous, Eliška Zajacová

Generated: February 24, 2026

Table of Contents

1. Introduction

This report presents a comprehensive machine learning project focused on predicting bank term deposit subscriptions using the Bank Marketing dataset. The project follows a structured approach including exploratory data analysis, data preprocessing, model training, and evaluation.

The primary objective is to develop classification models capable of accurately predicting whether a client will subscribe to a term deposit (variable 'y'). This prediction can help banks optimize their marketing campaigns by targeting clients with higher likelihood of subscription.

The project implements three different machine learning algorithms:
• Logistic Regression
• Decision Tree
• Random Forest

Each model is trained with both default and tuned hyperparameters to compare performance and identify the best approach for this classification task.

2. Dataset Description

Dataset: Bank Marketing Dataset
Source: UCI Machine Learning Repository
Total Samples: 4,521
Training Samples: 3,616
Test Samples: 905
Original Features: 16
Features after Encoding: 42
Target Distribution: 4000/521

The dataset contains information about direct marketing campaigns of a Portuguese banking institution. The campaigns were based on phone calls, and the goal is to predict if the client will subscribe to a term deposit.

Features include:
• Client data: age, job, marital status, education, default, balance, housing, loan
• Campaign data: contact type, day, month, duration, campaign, pdays, previous, poutcome
• Target: y (yes/no - subscription to term deposit)

3. Exploratory Data Analysis Findings

Key findings from the exploratory data analysis:

1. Dataset Structure:
   • No missing values detected
   • Mix of numerical and categorical features
   • Significant class imbalance (majority class: 'no')

2. Target Variable:
   • Binary classification problem
   • Imbalanced dataset requires special handling

3. Feature Distributions:
   • Numerical features show various distributions
   • Categorical features have multiple categories
   • Some features show correlation with target variable

4. Insights:
   • Duration of contact appears to be an important predictor
   • Previous campaign outcomes influence subscription rates
   • Client demographics (age, job, education) show patterns

All EDA visualizations are saved in the figures/ directory and included in this report.

4. Data Preprocessing

The following preprocessing steps were applied:

1. Target Encoding:
   • Converted 'yes'/'no' to binary (1/0)

2. Categorical Encoding:
   • Applied OneHot Encoding to all categorical variables
   • Used 'drop_first' to avoid multicollinearity
   • Resulted in 42 features after encoding

3. Train-Test Split:
   • 80% training, 20% testing
   • Stratified split to maintain class distribution
   • Random state: 42 for reproducibility

4. Feature Scaling (applied only where needed):
   • StandardScaler (mean=0, std=1) was applied for Logistic Regression only.
   • Reason: Logistic Regression is sensitive to feature magnitudes because it uses a linear combination z = w^T x + b; coefficients w are learned via gradient-based optimization, and features on different scales would dominate the objective and slow convergence or bias the solution.
   • Tree-based models (Decision Tree, Random Forest) are scale-invariant: splits compare values within a single feature, so multiplying a feature by a constant does not change the split structure. Therefore scaling was not applied for these models.

5. Class Imbalance Handling:
   • Computed class weights for balanced learning
   • Applied in tuned models using class_weight parameter

5. Methodology

Three classification algorithms were implemented:

1. Logistic Regression:
   • Default: Standard parameters
   • Tuned: class_weight='balanced', C=0.1, solver='liblinear'
   • Requires feature scaling

2. Decision Tree:
   • Default: Standard parameters
   • Tuned: max_depth=10, min_samples_split=20, min_samples_leaf=10, class_weight='balanced'
   • Handles non-linear relationships
   • Provides feature importance

3. Random Forest:
   • Default: Standard parameters (100 trees)
   • Tuned: n_estimators=200, max_depth=15, min_samples_split=10, min_samples_leaf=5, class_weight='balanced'
   • Ensemble method for improved performance
   • Provides feature importance

Evaluation Metrics:
• Accuracy: Overall correctness
• Precision: True positives / (True positives + False positives)
• Recall: True positives / (True positives + False negatives)
• F1-Score: Harmonic mean of precision and recall
• Confusion Matrix: Detailed classification breakdown

6. Mathematical Foundations of the Models

Logistic Regression

- Logistic (sigmoid) function: $\sigma(z) = 1 / (1 + e^{-z})$ maps z to (0, 1).
- Linear model: $z = w^T x + b$ (weight vector w, bias b). Probability $P(y=1|x) = \sigma(z)$.
- Binary cross-entropy loss (for one sample): $L = -[y \log(p) + (1-y) \log(1-p)]$, with $p = \sigma(z)$. Minimizing this over the training set gives maximum likelihood estimates for w, b.
- Regularization: L2 penalty adds $(1/(2C)) ||w||^2$ to the loss. Larger C means weaker regularization (less constraint on w); smaller C shrinks coefficients and increases bias.
- Hyperparameter C: inverse regularization strength. Small C => stronger regularization, higher bias.

Decision Tree

- Entropy of a set S (with class proportions p_i): $H(S) = - \sum_i p_i \log_2(p_i)$. Maximum when classes are equiprobable; zero when one class only.
- Gini impurity: $Gini(S) = 1 - \sum_i p_i^2$. Also measures class mixing; minimized when pure.
- Splitting: at each node the algorithm chooses the feature and threshold that maximize information gain (reduction in entropy or Gini) or equivalently minimize impurity in children.
- Effect of max_depth: limiting depth reduces model complexity. Deep trees fit training data closely (low bias, high variance); shallow trees underfit (higher bias, lower variance).

6. Mathematical Foundations (continued)

Random Forest

- Bagging (Bootstrap Aggregating): train many base learners (here, decision trees) on bootstrap samples of the training set (random draws with replacement, same size as train). Final prediction: majority vote (classification) or average (regression).
- Feature randomness: at each split, only a random subset of features is considered (e.g. sqrt(n_features)). This decorrelates trees and reduces variance of the ensemble.
- Variance reduction: if base learners have high variance and low bias (e.g. deep trees), averaging reduces variance while keeping bias roughly unchanged. Hence Random Forest typically has lower variance than a single decision tree.
- Why Random Forest reduces overfitting: (1) averaging many trees smooths out individual tree overfitting; (2) bootstrap and feature subsampling make each tree see different data/features, so errors are less correlated; (3) overall model complexity is controlled by number of trees and tree depth, trading bias for variance.

7. Bias–Variance Tradeoff

- Logistic Regression has relatively high bias (linear decision boundary) and low variance (stable under different samples). In our results, it gives moderate F1; tuning (e.g. C, class_weight) mainly shifts the precision–recall tradeoff rather than drastically changing accuracy, consistent with a stable (low-variance) model.

- Decision Tree (default) tends to low bias and high variance: it can fit complex boundaries and overfit. Our default tree shows moderate recall and F1; the tuned tree (with max_depth, min_samples_leaf, etc.) restricts complexity, increasing bias but reducing variance and often improving generalization (e.g. different recall/F1 balance on test set).

- Random Forest reduces variance compared to a single tree by averaging many trees while keeping bias similar. In our experiment, Random Forest (Tuned) achieves the best F1, reflecting lower variance (more stable predictions) and a good bias–variance tradeoff for this dataset. The gap between default and tuned Random Forest illustrates how hyperparameters (n_estimators, max_depth, min_samples_leaf) further control variance.

8. Model Results

Model	Accuracy	Precision	Recall	F1-Score
Logistic Regression (Default)	0.8917	0.5536	0.2981	0.3875
Logistic Regression (Tuned)	0.8243	0.3744	0.7885	0.5077
Decision Tree (Default)	0.8486	0.3701	0.4519	0.4069
Decision Tree (Tuned)	0.7580	0.2968	0.8077	0.4341
Random Forest (Default)	0.8862	0.5102	0.2404	0.3268
Random Forest (Tuned)	0.8773	0.4762	0.6731	0.5578

9. Results Analysis and Interpretation

Interpretation of metrics in context:

- Recall is critical in marketing campaigns: it measures the fraction of actual subscribers (positive class) that the model identifies. Missing a subscriber (false negative) means a lost opportunity; missing a non-subscriber (false positive) means an unnecessary contact. Hence high recall is often preferred when the cost of missing positives is high. Our tuned Logistic Regression and tuned Decision Tree achieve high recall (~0.79–0.81), suitable for "do not miss subscribers" strategies; Random Forest (Tuned) offers a more balanced recall (~0.67) with better precision.

- F1-score is more relevant than accuracy under class imbalance: with ~88% "no" and ~12% "yes", a classifier that always predicts "no" would have ~88% accuracy but zero recall for the positive class. F1 combines precision and recall and is not dominated by the majority class; thus we use F1 (and recall) to compare models. Random Forest (Tuned) achieves the best F1 (0.56) in our results, indicating the best compromise for this imbalanced problem.

- Class imbalance impact: without balancing (e.g. class_weight or SMOTE), models tend to favor the majority class, reducing recall. The tuned models explicitly account for imbalance; the tradeoff is often lower precision (more false positives) for higher recall. For the bank, this can mean more contacts with a higher chance of conversion among contacted clients.

- Practical implications: a strategy that maximizes recall may be used for initial screening (minimize missed subscribers); one that maximizes F1 balances outreach cost and conversion. Choosing a threshold on the probability output (e.g. ROC curve analysis) allows the bank to shift the operating point along the precision–recall tradeoff without retraining.

10. Model Comparison

Key observations from the results table:

• Random Forest (Tuned): best F1-Score; best balance between precision and recall and robustness to class imbalance. Recommended when a single operating point is needed without threshold tuning.

• Logistic Regression (Tuned): highest recall; useful when maximizing coverage of potential subscribers is the priority. Lower precision implies more contacts per conversion.

• Decision Tree (Tuned): high recall, lower accuracy; tuning reduced overfitting compared to the default tree. Feature importance (e.g. duration, poutcome) aligns with domain expectations.

• Feature importance (tree-based models): duration of contact and previous campaign outcome are among the strongest predictors; client balance and age also contribute. This supports targeting and contact-policy design.

11. Conclusion

This project implemented and compared Logistic Regression, Decision Tree, and Random Forest on the Bank Marketing dataset. Random Forest (Tuned) achieved the best F1-score, offering a favorable bias–variance tradeoff; Logistic Regression (Tuned) and Decision Tree (Tuned) provided high recall for strategies that prioritize identifying potential subscribers.

Model tradeoffs: Logistic Regression is interpretable and stable but linear; Decision Trees are flexible but prone to overfitting without tuning; Random Forest reduces variance at the cost of interpretability. The choice depends on whether the bank prioritizes interpretability, recall, or balanced F1.

Limitations: the dataset is a single 10% sample (bank.csv), temporal effects and external validity are not assessed, and the positive class remains small, so metrics are subject to sampling variation. Duration is a strong predictor but may not be known before the call, limiting its use in pre-campaign targeting.

Possible improvements: (1) k-fold cross-validation for more reliable metric estimates and hyperparameter selection; (2) extended GridSearchCV or RandomizedSearchCV over broader parameter ranges; (3) ROC-AUC and precision–recall curves to choose classification thresholds; (4) calibration of probability outputs for better decision support. The pipeline remains reproducible and academically grounded for further extension.