

Cisse, Yelene

Dr. Spencer W. Luo

COMS 4995W32

04 November 2025

Predicting Term Deposit Subscriptions: Comparative Analysis of Boosting Techniques

Introduction

This report analyzes the Bank Marketing dataset from the UCI Machine Learning Repository, consisting of 41,188 observations across 20 features, to predict whether customers will subscribe to a term deposit based on demographic and marketing campaign data.

The primary learning objective is to compare boosting techniques—specifically Gradient Boosting Machines (GBM) and XGBoost—against baseline Decision Tree and Random Forest models to understand how these ensemble methods improve predictive performance through bias-variance trade-offs and sequential error correction.

The analysis follows a structured approach:

1. Data preprocessing and Exploratory Data Analysis (EDA)
2. Baseline model development using Decision Trees
3. Boosting Machines exploration
 - a) Implementation of Random Forest as an improved baseline
 - b) Gradient Boosting Machine (GBM)
 - c) XGBoost (XGB)

Given the imbalanced nature of the target variable (11.3% positive class), evaluation metrics focus on F1-score and Area Under the Precision-Recall Curve (AUCPR) rather than accuracy alone.

1. Data preprocessing and EDA

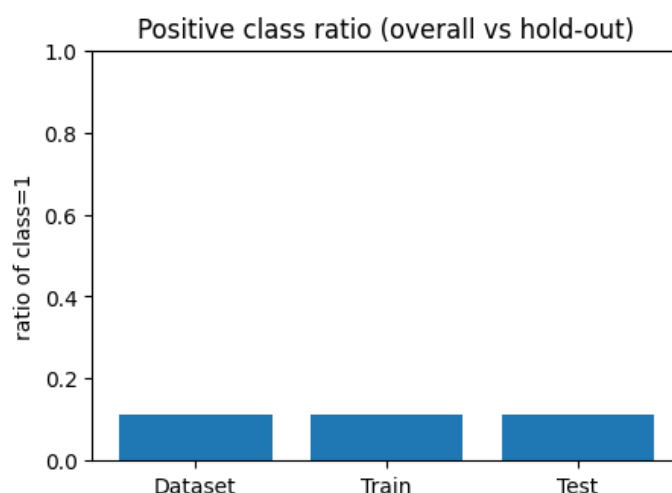
The dataset contains both categorical and continuous features describing bank clients, campaign details, and economic indicators. Initial exploratory analysis revealed several preprocessing requirements. Categorical variables including job, marital status, education, default, housing, loan, and personal loan contained "unknown" levels representing missing values. These were retained as separate categories during one-hot encoding rather than imputed, preserving information about data quality.

Among continuous variables, the "pdays" feature showed 96% of values coded as 999 (indicating no previous contact), suggesting limited informational value. High correlations were observed among economic indicators: euribor3m, emp.var.rate, and nr.employed (>94%), while euribor3m

and cons.price.idx showed 77.5% correlation. These relationships were noted for potential feature engineering in future iterations.

Following UCI dataset documentation recommendations, the "duration" variable was excluded from modeling despite its predictive power, as call duration is unknown before contact and only observable after the outcome is determined. Including it would create unrealistic predictions unsuitable for real-world deployment.

Data splitting preceded all transformations to prevent data leakage. An 80-20 train-test split with stratification maintained the target variable's distribution across both sets (11.27% positive class in training, 11.26% in test).



One-hot encoding transformed categorical variables into binary indicators, while continuous variables underwent standardization. The final feature space comprised 62 variables for model training.

2. Decision Tree

Decision Trees partition feature space through recursive binary splits, selecting split points that maximize information gain or minimize impurity at each node. This interpretable approach serves as our initial baseline for understanding model performance before applying ensemble methods.

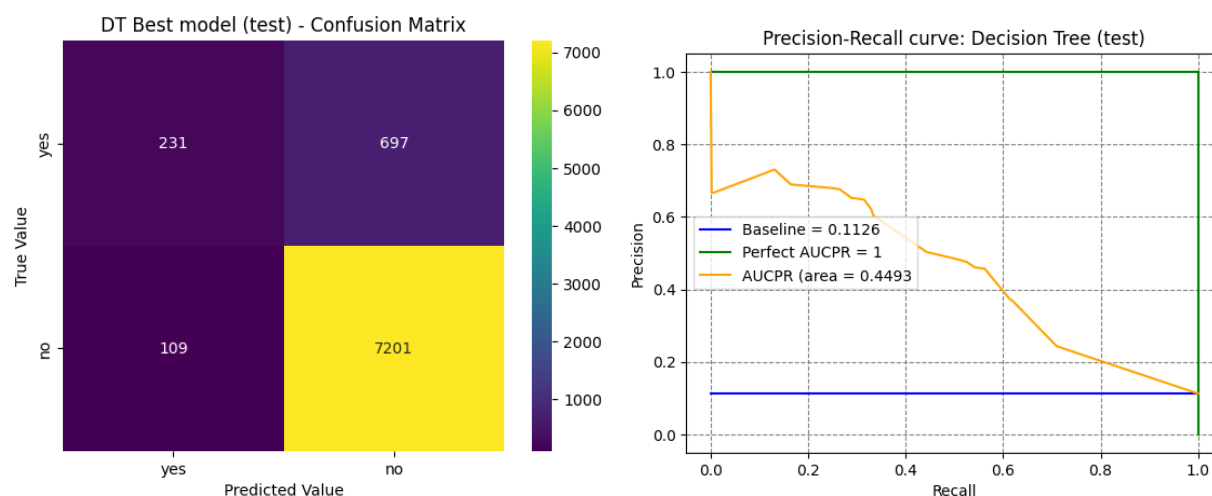
Hyperparameter tuning via GridSearchCV with 5-fold cross-validation explored combinations of max_depth (2, 5, 10, 15, 20), min_samples_split (2, 5, 10, 15, 20), and ccp_alpha (0, 0.0001, 0.001, 0.01, 0.05), using negative log loss as the optimization metric. The optimal configuration achieved max_depth=5, min_samples_split=10, and ccp_alpha=0.0001, yielding an average cross-validation score of -0.2795.

Analysis of individual hyperparameters revealed clear bias-variance patterns. Maximum depth of 2 produced shallow trees with high bias and underfitting, while depths of 15 and 20 led to overfitting evidenced by widening gaps between training and validation performance.

The validation curve for `max_depth` showed training negative log loss continuously improving with depth, but validation performance degrading sharply beyond depth 5, indicating high variance. The optimal depth of 5 balanced model complexity with generalization.

The regularization parameter `ccp_alpha` demonstrated that both extremes—no regularization (0) and excessive pruning (0.05)—yielded suboptimal performance. Moderate regularization (0.0001-0.001) prevented overfitting without oversimplifying the model. The `min_samples_split` parameter showed less pronounced effects when combined with appropriate `max_depth` and `ccp_alpha` values, as these parameters already controlled tree complexity effectively.

Test set evaluation revealed accuracy of 90.2%, F1-score of 36.4%, and AUCPR of 44.9%. The high accuracy masked moderate challenges with the minority class, though the F1-score of 36.4% demonstrated reasonable precision-recall balance for the imbalanced problem. The confusion matrix showed 697 of 928 actual positives (75%) misclassified as negatives, reflecting high precision but low recall. The precision-recall curve confirmed this pattern, with the model favoring specificity over sensitivity.



Feature importance analysis identified severe concentration: `nr.employed` dominated with 65.2% importance, followed distantly by `pdays` (13.7%), `cons.conf.idx` (6.2%), and `euribor3m` (5.2%). This heavy reliance on a single economic indicator suggested overfitting and high variance, motivating the exploration of ensemble methods to diversify feature utilization and improve generalization.

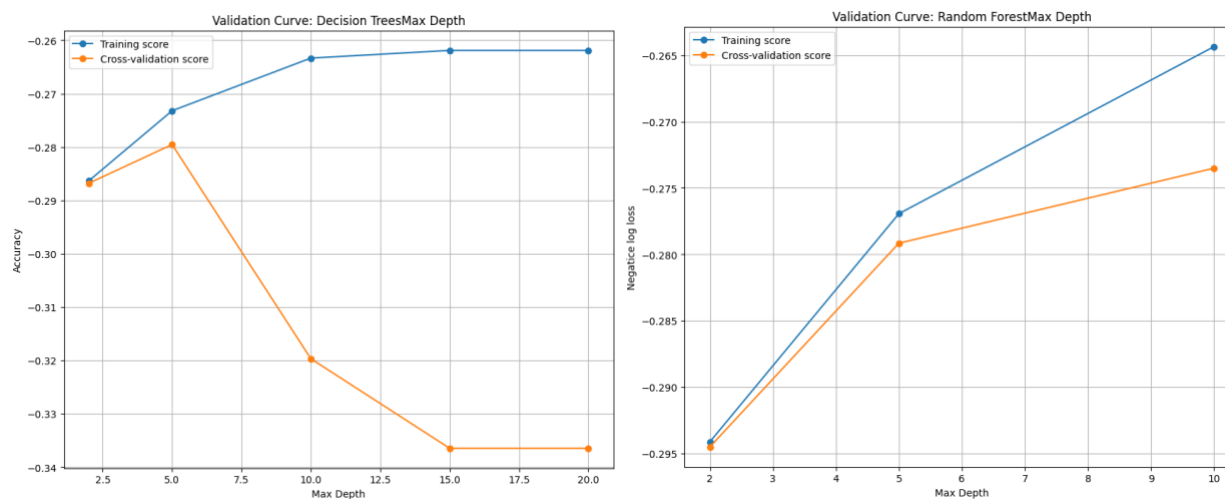
3. Boosting Machines

a) Random Forest

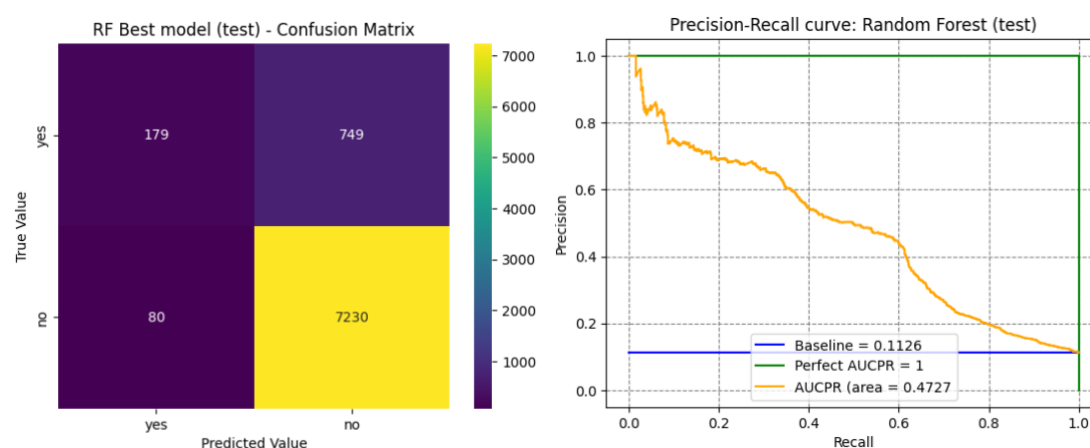
Random Forest constructs multiple decision trees on bootstrapped samples with random feature subsets at each split, then aggregates predictions through majority voting. This bagging approach reduces variance by averaging predictions across decorrelated trees.

GridSearchCV explored `max_depth` (2, 5, 10), `min_samples_split` (2, 5, 10, 15), and `ccp_alpha` (0.0001, 0.001, 0.01) with 5-fold cross-validation. The parameter space was reduced compared to the single Decision Tree to manage computational cost. The optimal model achieved `max_depth=10`, `min_samples_split=10`, `ccp_alpha=0.0001`, with cross-validation score of -0.2735, representing marginal improvement over the single tree (-0.2795).

The validation curve for `max_depth` showed smoother convergence between training and validation scores compared to the Decision Tree, though some divergence remained at depth 10. This demonstrated Random Forest's variance reduction through ensemble averaging.



Test performance yielded accuracy of 89.9%, F1-score of 30.2%, and AUCPR of 47.3%. While AUCPR improved by approximately 2.4 percentage points over Decision Tree (47.3% vs 44.9%), the F1-score decreased from 36.4% to 30.2%, and the confusion matrix showed 749 of 928 positives (81%) misclassified—worse than the single Decision Tree (75%).



Despite the lower F1-score (30.2% vs 36.4%), Random Forest demonstrated clear improvements over Decision Tree in other key areas. Feature importance showed substantially improved distribution: `nr.employed` (17.1%), `euribor3m` (16.8%), `pdays` (12.5%), `poutcome=success`

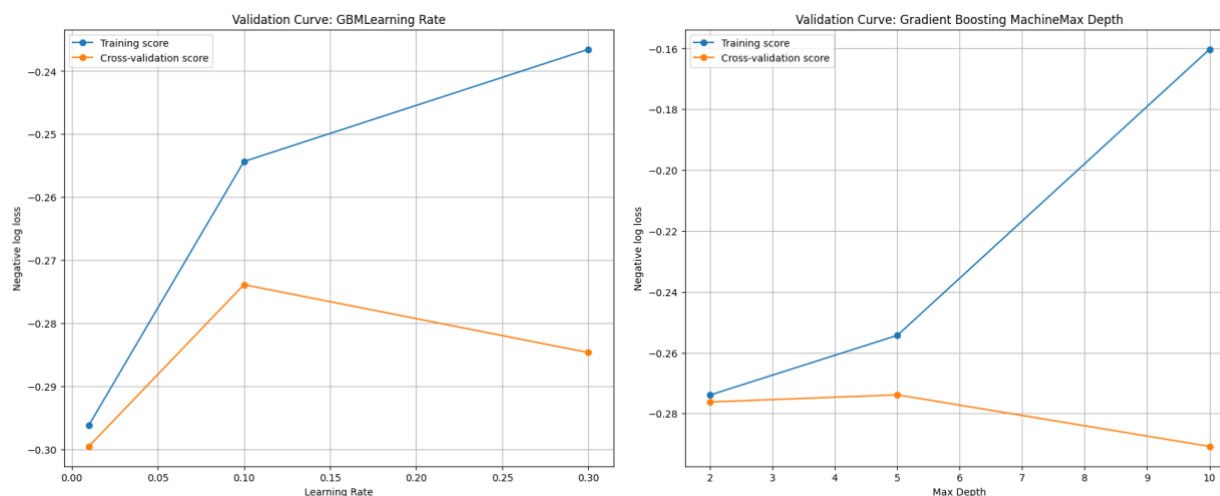
(9.0%), and cons.conf.idx (7.5%). This broader distribution across features—contrasted with Decision Tree's 65.2% concentration on nr.employed—indicated reduced variance and less overfitting to a single predictor. The validation curves confirmed reduced overfitting, with smaller gaps between training and validation performance across max_depth values. While the model still struggled with the minority class, the improved AUCPR (47.3% vs. 44.9%) and more balanced feature utilization demonstrated that Random Forest successfully addressed the high variance problem observed in the single Decision Tree. These improvements motivated the transition to boosting methods that could further enhance performance through iterative error correction.

b) Gradient Boosting Machine

Gradient Boosting builds trees sequentially, with each new tree fitting the residual errors from the ensemble of previous trees. By iteratively correcting mistakes, GBM reduces bias while controlling variance through regularization and learning rate parameters.

The hyperparameter grid included n_estimators (20, 50, 100), max_depth (2, 5, 10), learning_rate (0.01, 0.1, 0.3), and subsample (0.25, 0.5, 0.75). The optimal configuration selected n_estimators=50, max_depth=5, learning_rate=0.1, and subsample=0.75, achieving cross-validation score of -0.2739.

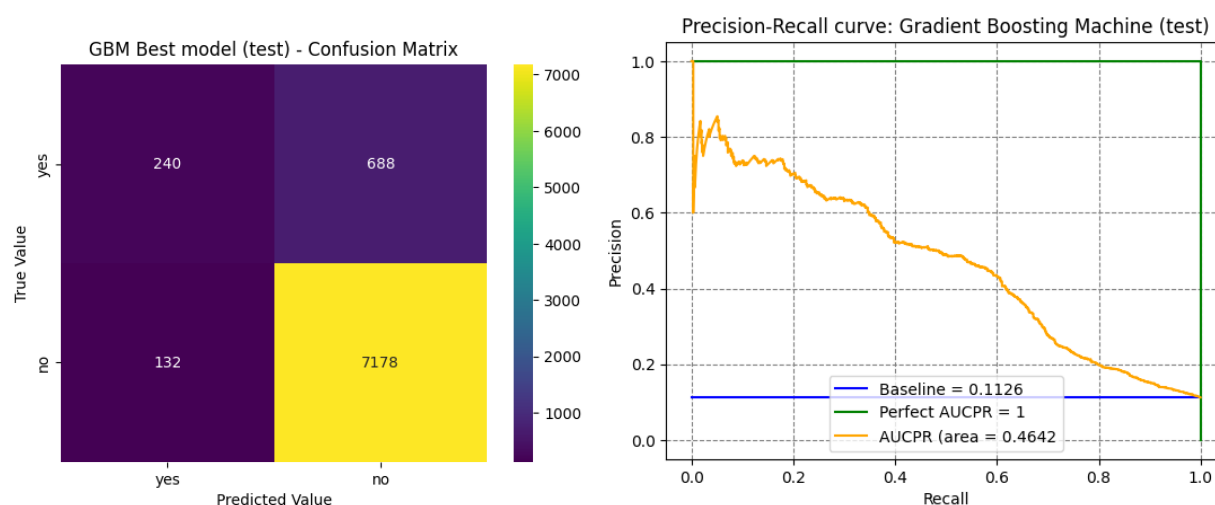
Hyperparameter analysis revealed intuitive patterns. Subsample of 75% outperformed 50% and 25%, as higher sampling rates provided sufficient data for each boosting round while maintaining some randomness for variance reduction. Learning rate of 0.1 balanced convergence speed with stability—0.01 required more iterations to reduce error effectively, while 0.3 risked overfitting by giving excessive weight to error corrections. The validation curve for learning rate demonstrated this clearly: training performance improved monotonically with higher rates, but validation performance peaked at 0.1 before degrading.



For n_estimators, 50 and 100 trees performed best when paired with learning_rate=0.1. More estimators allowed gradual error reduction, but the model avoided overfitting by stopping at appropriate depth. Max_depth=5 emerged optimal, with depth 2 showing underfitting and depth

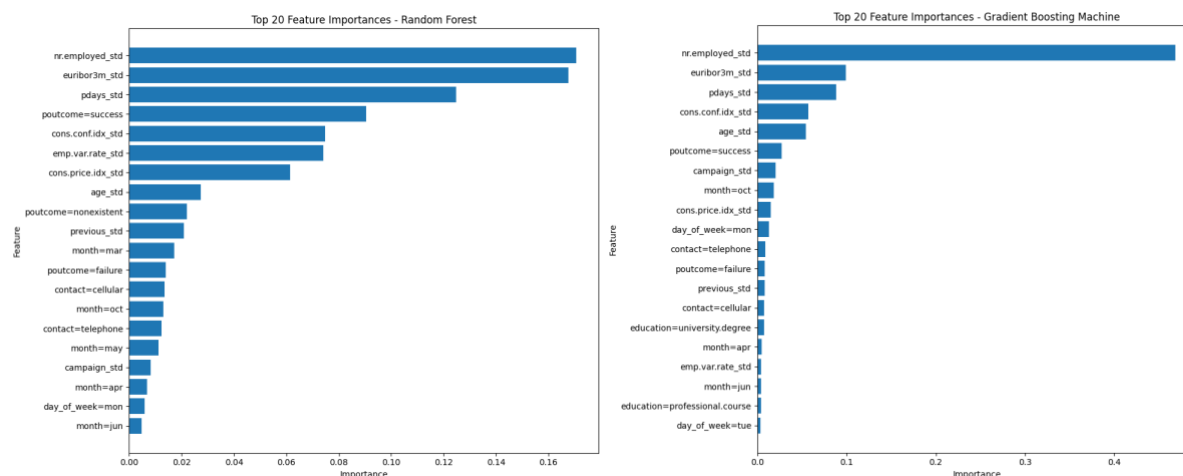
10 showing widening train-validation gaps indicative of high variance. The validation curve confirmed that depth 5 achieved the best bias-variance trade-off, with validation scores degrading beyond this point while training scores continued improving.

Test evaluation showed significant improvement over Random Forest: accuracy 90.0%, F1-score 36.9%, and AUCPR 46.4%. Most notably, the F1-score increased dramatically from Random Forest's 30.2% to 36.9%, a 6.7 percentage point improvement. This F1-score of 36.9% also narrowly surpassed the Decision Tree baseline (36.4%), establishing GBM as the best performer on this primary evaluation metric. The confusion matrix showed 688 of 928 positives (74.1%) misclassified, a substantial improvement from Random Forest's 81% misclassification rate.



GBM outperformed Random Forest through its sequential error-correction mechanism. While Random Forest reduced variance through parallel ensemble averaging and achieved superior AUCPR (47.3% vs 46.4%), GBM's sequential approach of fitting each tree to residual errors enabled superior precision-recall optimization, resulting in the highest F1-score (36.9%) among all models tested. This demonstrated that for imbalanced classification tasks, sequential boosting's targeted error correction can outperform parallel bagging's variance reduction when F1-score is the priority.

Feature importance revealed more balanced distribution than Decision Tree but slightly more concentrated than Random Forest: nr.employed (46.9%), euribor3m (9.9%), pdays (8.9%), cons.conf.idx (5.7%), age (5.4%), and poutcome=success (4.5%). This pattern indicated GBM's ability to identify the most predictive features while still incorporating diverse secondary features for improved generalization. Compared to Random Forest's more uniform distribution, GBM's focused yet diversified feature utilization suggested better signal extraction from the available predictors.

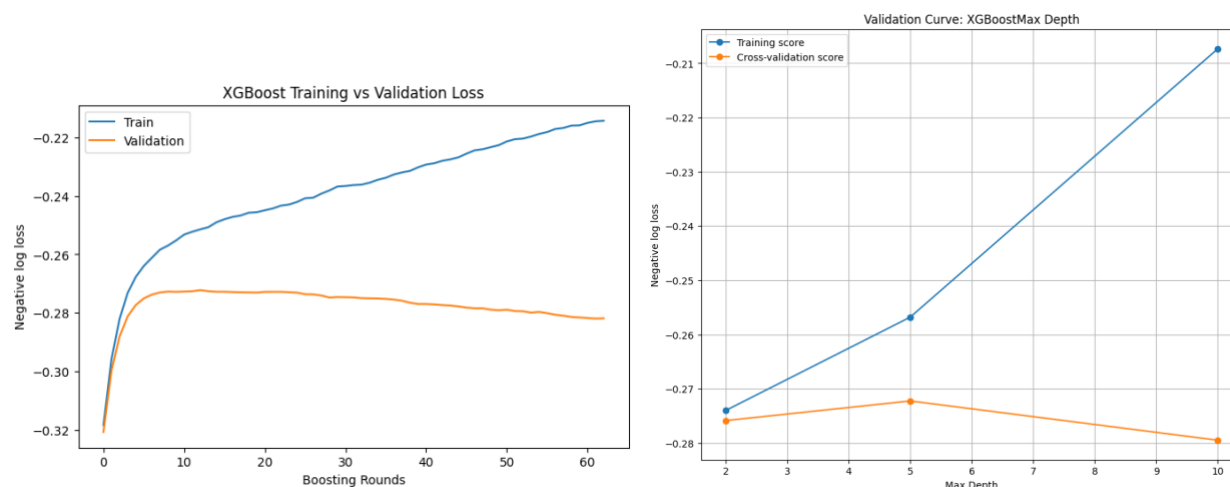


The validation curves demonstrated GBM's superior bias-variance trade-off. Unlike Decision Tree's severe divergence at higher depths or Random Forest's struggle with minority class recall, GBM maintained tight convergence between training and validation performance at optimal hyperparameters. This characteristic, combined with the substantially improved F1-score, established GBM as a clear advancement over both baseline approaches.

c) XGBoost

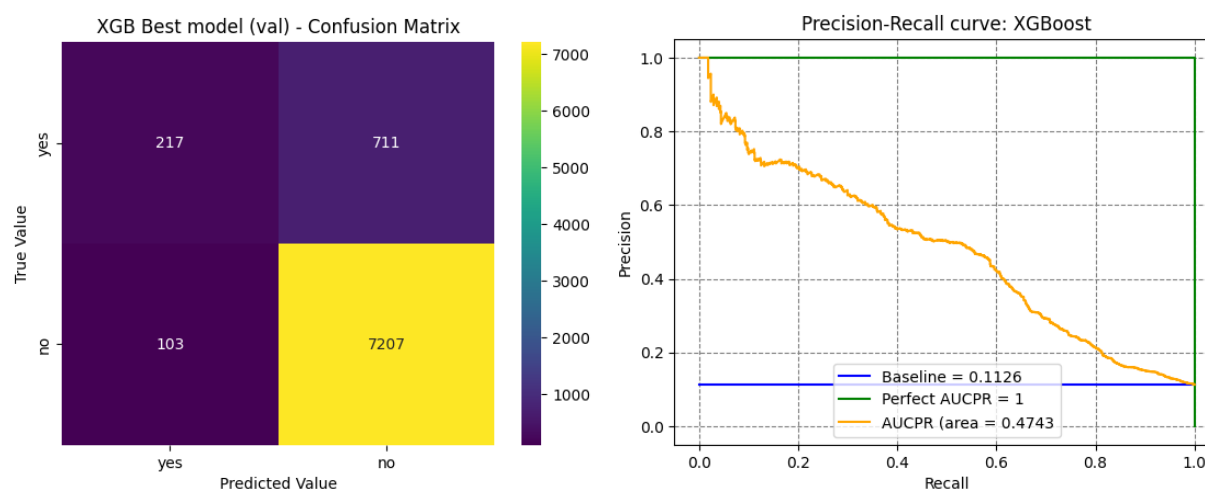
XGBoost implements gradient boosting with additional optimizations including regularization terms in the objective function, parallel tree construction, and built-in handling of missing values. These enhancements often improve computational efficiency and predictive performance.

Using the same parameter grid as GBM, the optimal XGBoost model selected $n_estimators=50$, $max_depth=5$, $learning_rate=0.1$, and $subsample=0.5$, achieving cross-validation score of -0.2722—the best among all models. An additional analysis with $early_stopping_rounds=50$ on a validation split from the training data revealed that validation loss plateaued around 12 boosting rounds before slowly degrading, while training loss continued decreasing. This visualization confirmed the model's ability to prevent overfitting through early stopping mechanisms.



Validation curves for max_depth and learning_rate mirrored GBM patterns: depth 5 provided optimal complexity, and learning_rate=0.1 balanced bias and variance. The training-validation gap remained minimal for these settings, indicating good generalization. Hyperparameter effects aligned with GBM: higher estimators with moderate learning rates, optimal depth preventing both underfitting and overfitting, and sufficient subsampling for robust training.

Test performance showed accuracy 90.1%, F1-score 34.8%, and AUCPR 47.4%. While XGBoost achieved the highest AUCPR among all models (47.4%), its F1-score of 34.8% fell short of GBM's leading 36.9%, representing a 2.1 percentage point gap. This placed XGBoost second in F1-score performance, behind GBM (36.9%).



XGBoost's performance relative to GBM and Random Forest reveals interesting trade-offs. Compared to GBM, XGBoost showed lower F1-score (34.8% vs 36.9%), a meaningful 2.1 percentage point difference that positioned it behind the leader. However, XGBoost achieved superior cross-validation performance and the highest AUCPR (47.4% vs 46.4%), suggesting that while GBM better optimized the precision-recall trade-off at the decision threshold, XGBoost provided superior probability calibration across all thresholds.

However, XGBoost still substantially outperformed Random Forest on the primary F1-score metric. The F1-score of 34.8%, while lower than GBM's 36.9%, remained significantly higher than Random Forest's 30.2%, representing a 4.6 percentage point improvement. XGBoost also achieved marginally higher AUCPR (47.4% vs 47.3%), demonstrating the benefits of sequential boosting over parallel bagging for this task.

Feature importance demonstrated patterns similar to GBM: nr.employed (46.8%), emp.var.rate (5.1%), poutcome=success (4.8%), month=apr (3.0%), cons.conf.idx (2.7%), and pdays (2.6%). The distribution across economic, temporal, and campaign features suggested XGBoost effectively identified relevant predictors while managing multicollinearity among correlated economic indicators. This balanced utilization, combined with the strong cross-validation performance, indicated that XGBoost's regularization framework successfully controlled overfitting even if it slightly underperformed GBM in maximizing F1-score.

The similar hyperparameter selections across both boosting methods ($n_estimators=50$, $max_depth=5$, $learning_rate=0.1$) but different subsample choices (GBM: 0.75, XGBoost: 0.5) highlighted subtle algorithmic differences. XGBoost's more aggressive subsampling paired with built-in regularization terms may have contributed to its superior log loss while slightly limiting its ability to capture minority class patterns as effectively as GBM's configuration.

Conclusion

This comparative analysis demonstrates that boosting techniques substantially improve upon baseline Decision Tree and Random Forest models for predicting term deposit subscriptions, with GBM emerging as the best overall model. Using F1-score as the primary evaluation metric for this imbalanced classification problem, GBM achieved the highest F1-score at 36.9%, outperforming XGBoost (34.8%) and Random Forest (30.2%). GBM's sequential error-correction approach proved most effective for optimizing precision-recall balance, achieving 90.0% accuracy and 46.4% AUCPR.

The progression from Decision Tree (65% reliance on one feature) through Random Forest (improved feature distribution, reduced variance) to GBM and XGBoost (bias-variance optimization with iterative error correction) illustrates fundamental machine learning principles. Random Forest successfully addressed Decision Tree's high variance through bagging but struggled with minority class recall. Boosting methods enhanced performance by specifically targeting misclassified instances across iterations, with GBM's configuration achieving the best balance for this dataset.

Key predictors across successful models included `nr.employed` (number of employees—a quarterly economic indicator), `euribor3m` (3-month interest rate), `pdays` (days since previous contact), and `poutcome=success` (previous campaign success). These features suggest that economic conditions and prior engagement history strongly influence subscription likelihood.

Several limitations warrant acknowledgment. The exclusive focus on hyperparameter tuning to compare boosting techniques meant that potential improvements through feature engineering were not explored. The high correlation among economic indicators (>94% between `euribor3m`, `emp.var.rate`, and `nr.employed`) suggests that dimensionality reduction or selective feature exclusion could improve generalization. The imbalanced target variable (11.3% positive class) indicates that threshold optimization above the default 0.5 could enhance sensitivity without sacrificing precision.

Future work should address these limitations through systematic feature engineering, correlation-based feature selection, and threshold tuning specific to the business context. The practical implementation of boosting methods requires consideration beyond predictive performance. These models demand greater computational resources for training, produce predictions more slowly than simpler alternatives, and present challenges in explaining their decision-making process to non-technical stakeholders—all factors that influence whether they are suitable for real-world deployment.

Despite these considerations, this analysis successfully demonstrates the value of boosting techniques in handling complex, imbalanced classification problems, with GBM representing the optimal choice for this term deposit prediction task.

AI Tool Usage Disclosure

This analysis utilized several AI tools to support development and documentation. ChatGPT provided assistance with code debugging, suggested typical hyperparameter ranges for model training, offered guidance on code modifications for improved computational efficiency (such as setting `n_jobs=-1` for parallel processing), provided feedback on manually written code for accuracy verification, generated plotting functions that were subsequently adapted for specific use cases, and supported general troubleshooting throughout the analysis workflow.

Microsoft Word's dictate feature was employed alongside handwritten analytical notes to draft initial report sections and summarize model-specific observations. Claude AI generated the polished final report based on the provided outline, written draft, and annotated code PDF containing all self-documented comments. The final report underwent manual editing for technical accuracy, clarity, and alignment with observations, with all graphs, and quantitative results inserted and verified.

All substantive analytical conclusions, model interpretations, hyperparameter analyses, and comparative evaluations represent original work based on direct examination of model outputs, validation curves, confusion matrices, and feature importance distributions.

Works Cited (Includes code references)

Moro, S., P. Cortez, and P. Rita. "Bank Marketing." *UCI Machine Learning Repository*, 2014, doi.org/10.24432/C5K306.

Columbia University. "AML Course: Decision Trees and Ensemble Methods." *Applied Machine Learning Course Materials*, 2025, colab.research.google.com/drive/1Ajm0FAz-dm3-0jDawe4AiS1rH9srbYM8.