Adult Income Prediction: Impact of Preprocessing and Tuning on Classical ML Models

Ferdynand Sikora Krystyna Supińska

- 123456@student.pwr.edu.pl

Politechnika Wrocławska

Introduction

This study addresses the binary classification task of predicting whether an individual's annual income exceeds $50,000 based on demographic and employment attributes from the UCI Adult Census Income dataset. The dataset serves as a classic benchmark for evaluating tabular machine learning methods and illustrates the effects of preprocessing choices on model performance.

Research Question: *How much do preprocessing choices and hyperparameter tuning influence the predictive performance of classical machine-learning models on the Adult income prediction task?*

Dataset Overview

Figure 1. Target class distribution showing significant imbalance.

Dataset Statistics:

- **N = 48,842** samples (combined train + test)
- **14 features**: 6 numerical, 8 categorical
- **Class imbalance**: 76% (≤50K) vs 24% (>50K)
- **Missing values**: 6,465 total (workclass, native_country, fnlwgt)

Objectives

- **Compare classical ML models** under a consistent preprocessing pipeline
- **Quantify preprocessing effects**, particularly missing value handling and feature selection strategies
- **Rigorous evaluation** using nested stratified cross-validation with multiple metrics and statistical significance testing

Preprocessing Pipeline & Architecture

Preprocessing: Missing values → "Missing" category; One-hot encoding; StandardScaler; Mutual Information feature selection (non-tree models only).

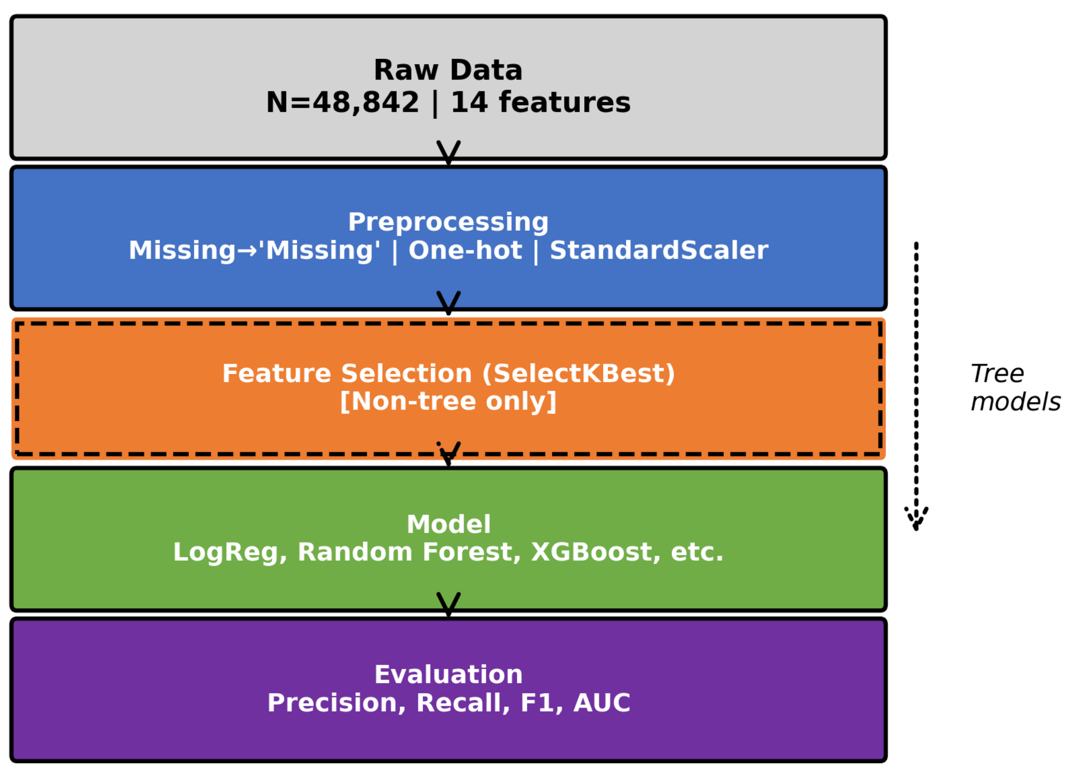

Figure 2. Preprocessing and modeling pipeline. Dashed border = conditional step (non-tree models only). Dashed arrow = tree models bypass feature selection.

Models Compared

We evaluate five classical machine learning algorithms:

- **Logistic Regression** — Linear baseline with L2 regularization
- **Naive Bayes (Gaussian)** — Probabilistic generative baseline
- **k-Nearest Neighbors** — Instance-based method, sensitive to scaling
- **Random Forest** — Ensemble of decision trees with bagging
- **XGBoost** — Gradient boosting with regularization

Study Design

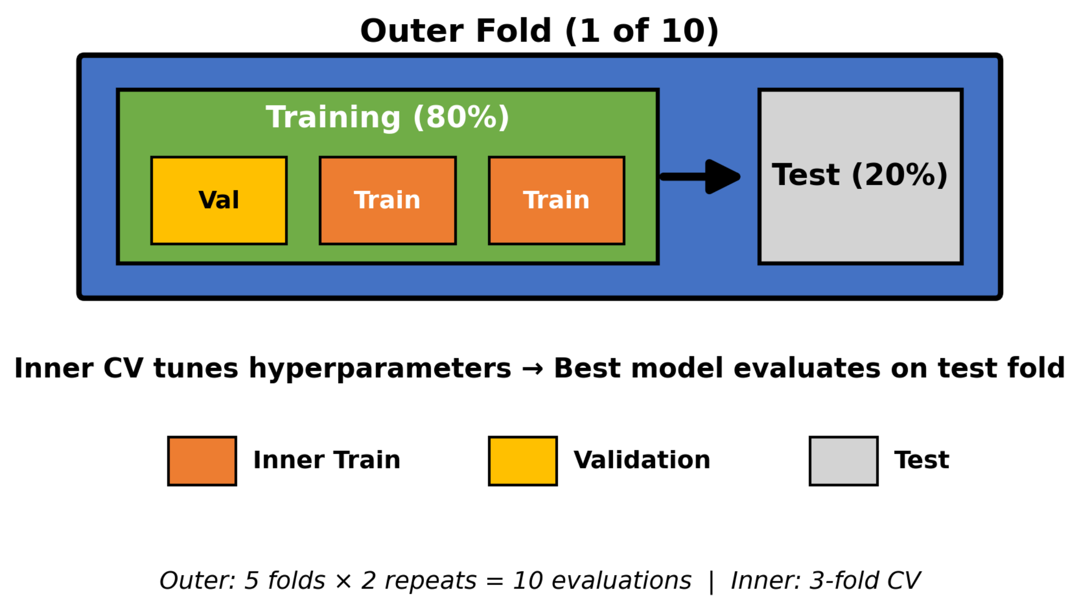

Figure 3. Nested cross-validation design. Outer loop (5 folds × 2 repeats = 10 evaluations) provides unbiased performance estimates. Inner loop (3-fold CV) tunes hyperparameters on the training set only.

Hyperparameter Search Space

Model	Tuned Parameters
Logistic Regression	C: {0.01, 0.1, 1, 10}
k-NN	n_neighbors: {3, 5, 7}
Random Forest	n_estimators: {100, 200}
XGBoost	max_depth: {3, 5}, learning_rate: {0.05, 0.1}, n_estimators: {100, 200}, subsample: {0.8, 1.0}

Table 1. Key hyperparameters explored via grid search. Feature selection (k) tuned for non-tree models.

Statistical Comparison (Wilcoxon)

Goal: Assess whether performance differences are statistically significant across paired outer-fold ROC-AUC scores.

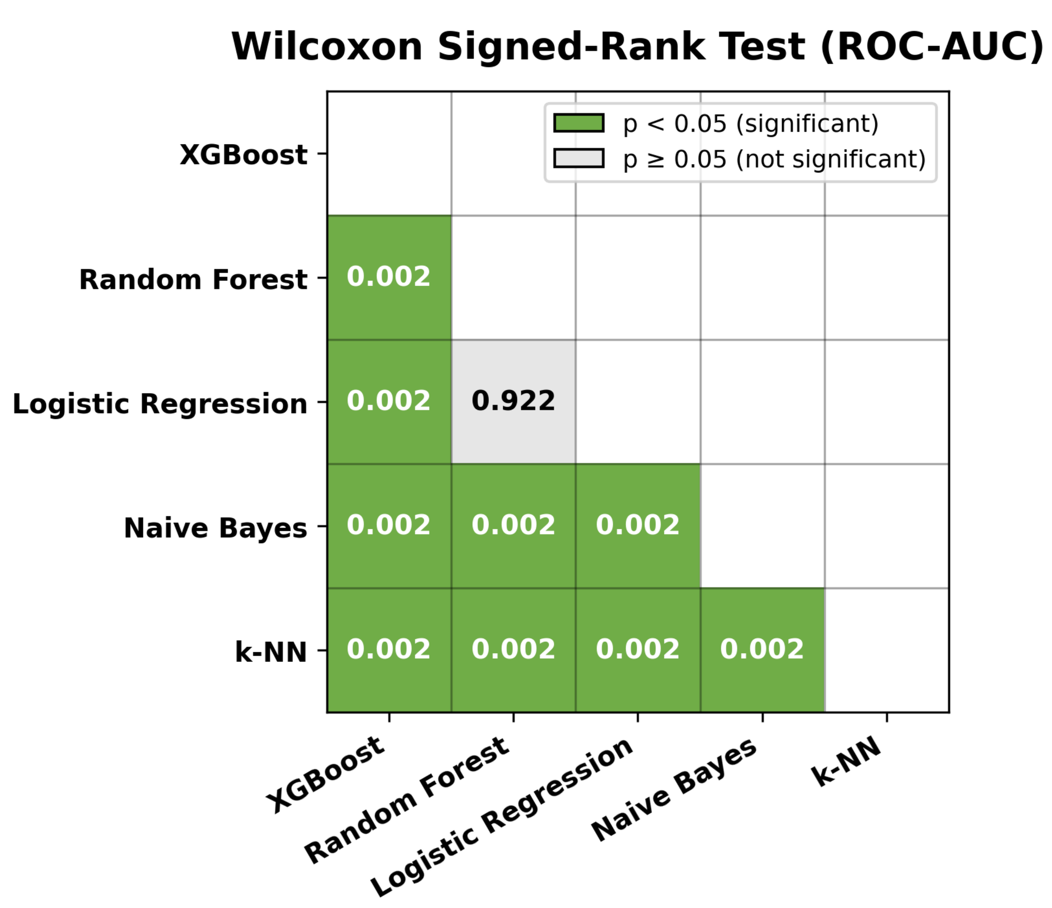

Figure 4. Pairwise Wilcoxon signed-rank test p-values (ROC-AUC). Green indicates significant differences ($p < 0.05$).

Key Findings

- **Best performers:** XGBoost and Random Forest achieved highest mean ROC-AUC across outer folds
- **Preprocessing impact:** "Missing" category encoding improved model stability compared to row dropping
- **Feature selection:** MI-based selection provided marginal benefit for linear models but was unnecessary for tree-based methods

Model Performance Comparison

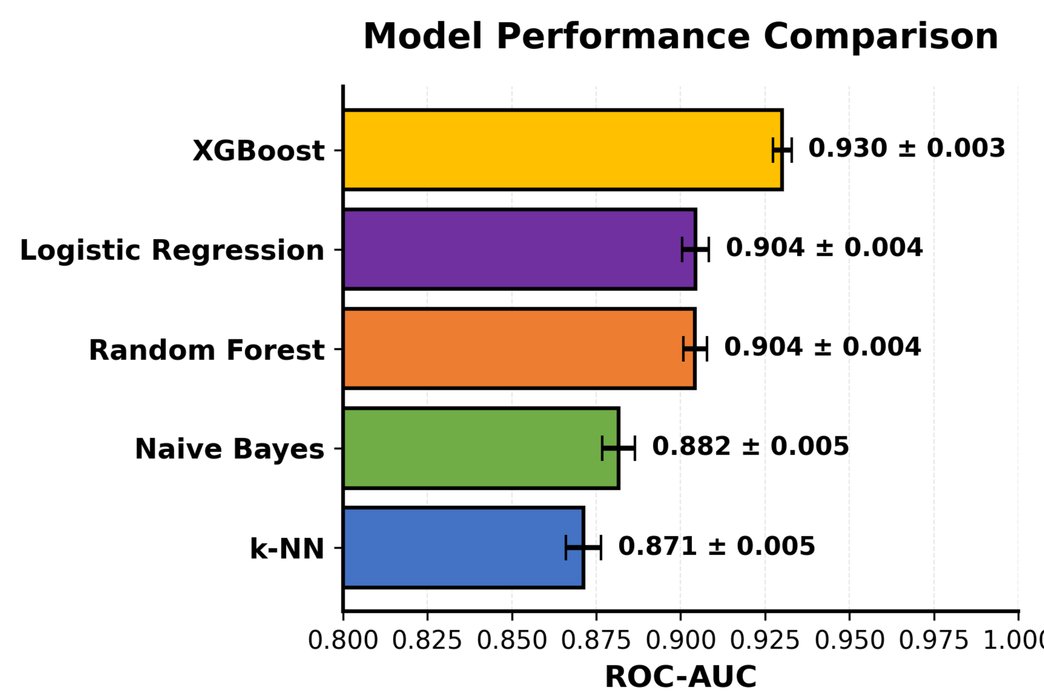

Figure 5. Performance comparison across models. Error bars show standard deviation across 10 outer folds.

Performance Metrics Comparison

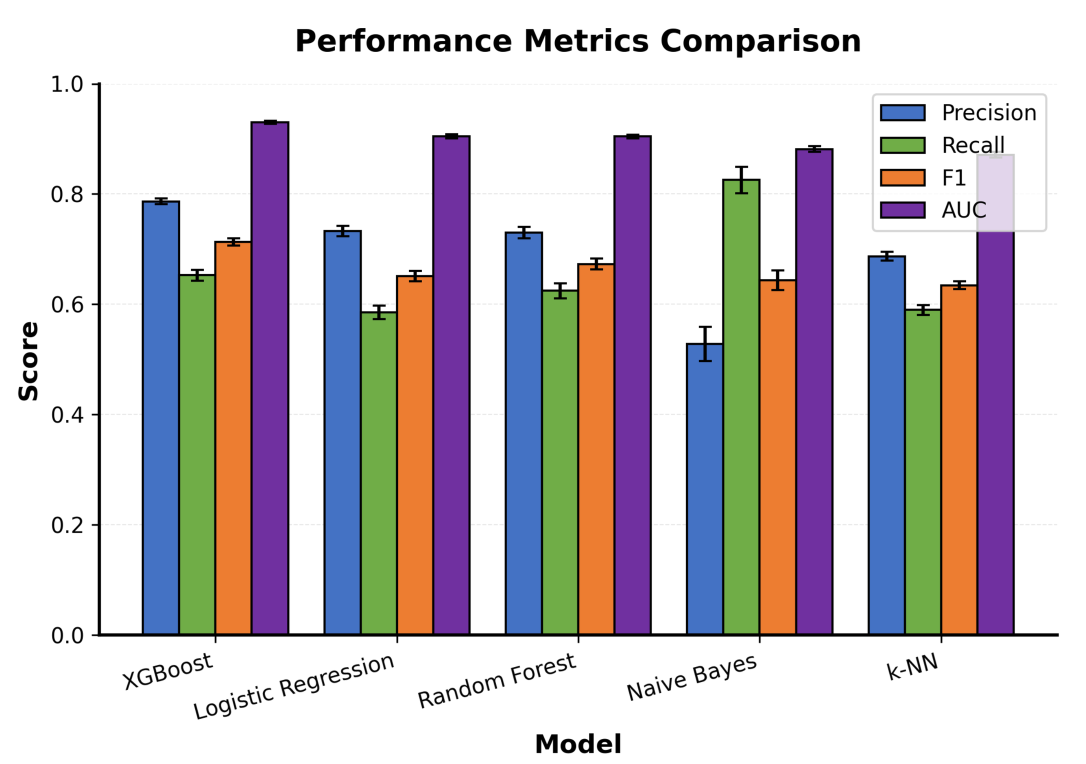

Figure 6. Comparison of Precision, Recall, F1-Score, and ROC-AUC across all models. Error bars show standard deviation across 10 outer folds.

Performance Summary

Model	Prec.	Rec.	F1	AUC
XGBoost	$0.787_{\pm 0.005}$	$0.653_{\pm 0.010}$	$0.713_{\pm 0.007}$	$0.930_{\pm 0.003}$
RF	$0.730_{\pm 0.011}$	$0.624_{\pm 0.013}$	$0.673_{\pm 0.010}$	$0.904_{\pm 0.004}$
Log.Reg.	$0.733_{\pm 0.009}$	$0.586_{\pm 0.012}$	$0.651_{\pm 0.009}$	$0.904_{\pm 0.004}$

Table 2. Top 3 models (mean ± std across 10 outer folds).

Discussion & Limitations

- **Dataset age:** 1994 Census data may not reflect current income patterns
- **High dimensionality:** One-hot encoding expands feature space significantly

Conclusion

XGBoost achieved the highest ROC-AUC (0.930), significantly outperforming all other models according to Wilcoxon signed-rank tests. Encoding missing values as a dedicated category proved effective, and Mutual Information feature selection offered marginal gains for non-tree models. For practitioners, gradient boosting with careful preprocessing provides the best accuracy–complexity tradeoff on tabular census data.