A Study of Methods for Balancing Machine Learning Training Data

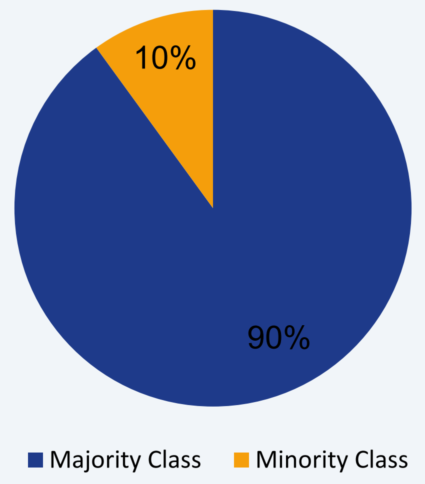
Comparative Analysis & Performance Evaluation

The Class Imbalance Problem

Class imbalance occurs when one class significantly outnumbers others in training data.

Common Scenarios

- Fraud detection (fraudulent transactions are rare)
- Medical diagnosis (disease cases are minority)
- Spam filtering (legitimate emails dominate)
- Manufacturing defects (defective products are uncommon)



Impact on Model Performance

Accuracy Paradox

Models achieve high accuracy by predicting only the majority class, missing critical minority cases.

Bias Toward Majority

Classifiers become biased toward the majority class, resulting in poor minority class predictions.

Poor Generalization

Models fail to generalize well to real-world scenarios where minority class detection is crucial.

Misleading Metrics

Standard accuracy metrics become unreliable, requiring alternative evaluation approaches.

Data Balancing Methods

Undersampling

Reduce majority class samples

Oversampling

Increase minority class samples

Hybrid

Combine both approaches

Undersampling Techniques

Random Undersampling

Randomly removes majority class samples until balance is achieved.

NearMiss Algorithm

Selectively removes majority samples based on distance to minority class.

Tomek Links

Removes majority class samples forming Tomek links with minority samples.

Undersampling Techniques

Handling Imbalanced Datasets in Machine Learning

Methods to Balance Class Distribution

The Class Imbalance Problem

What is Class Imbalance?

When one class significantly outnumbers the other in a dataset, models tend to favor the majority class.

Common Examples

- Fraud detection (rare fraudulent transactions)
- Medical diagnosis (rare diseases)
- Anomaly detection (rare system failures)

The Challenge

Before

Majority: 95%
Minority: 5%

After Undersampling

Majority: 50%
Minority: 50%

Random Undersampling

How It Works

Randomly removes majority class samples until balance is achieved.

Advantages

- Simple and fast
- Computationally efficient

Disadvantages

- May discard useful information
- No intelligent selection

Visualization

NearMiss Algorithm

How It Works

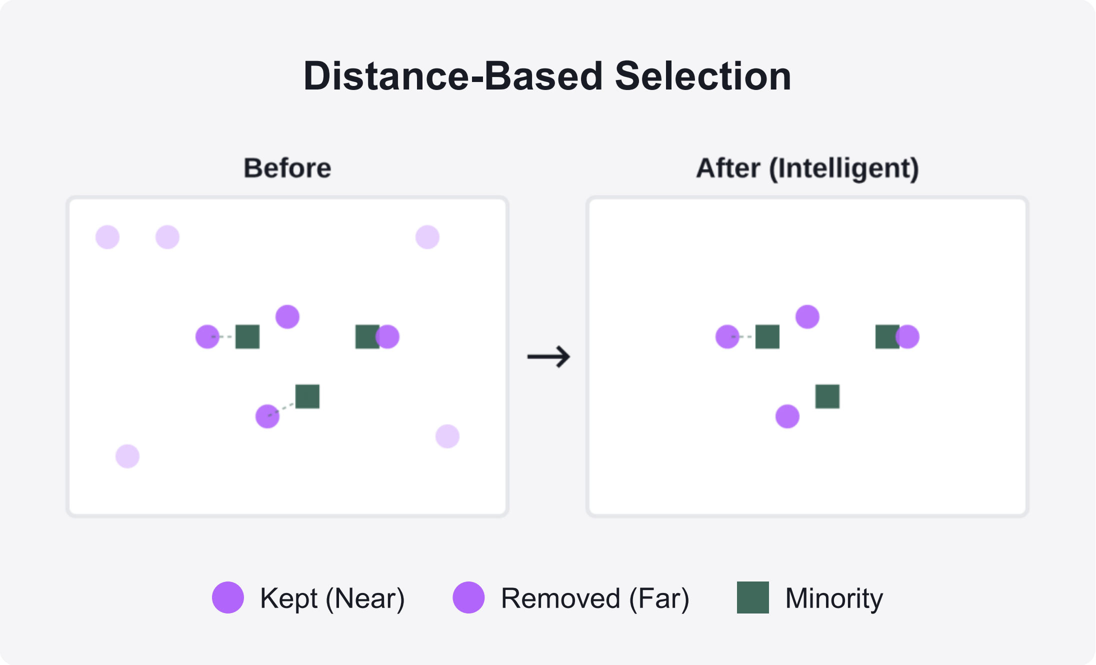
Selectively removes majority samples based on distance to minority class.

Three Variants

- •NearMiss-1: Closest to minority
- •NearMiss-2: Farthest from minority

Key Benefits

- •Intelligent sample selection
- •Preserves decision boundaries



Tomek Links

How It Works

Removes majority class samples forming Tomek links with minority samples.

What is a Tomek Link?

Two samples from different classes are each other's nearest neighbors.

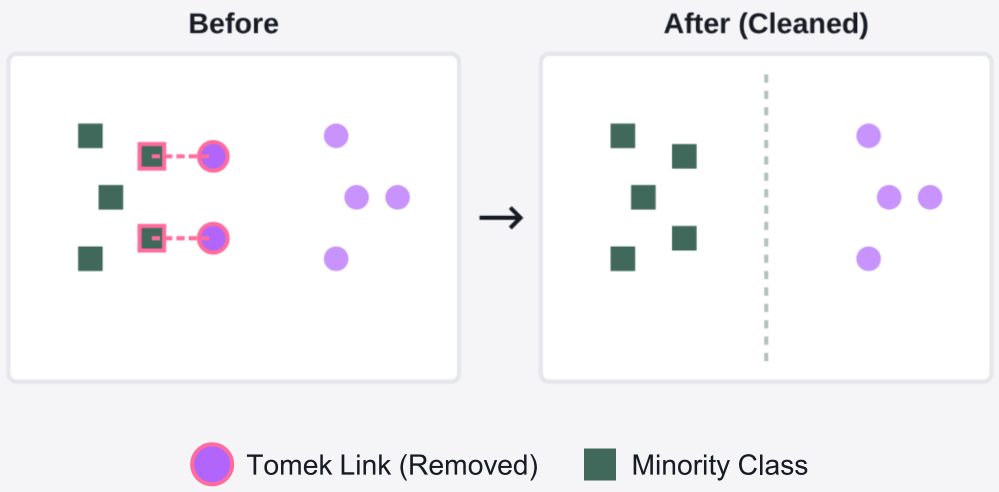
Advantages

- Cleans class boundaries
- Removes ambiguous samples

Limitations

- Only removes boundary samples

Boundary Cleaning Process



Comparison of Techniques

Random

Strategy

Random removal of majority samples

Speed

Fast ⚡ ⚡ ⚡

Intelligence

Low

Best For

Quick prototypes, large datasets

NearMiss

Strategy

Distance-based selection

Speed

Medium ⚡ ⚡

Intelligence

High

Best For

Preserving decision boundaries

Tomek Links

Strategy

Boundary cleaning

Speed

Medium ⚡ ⚡

Intelligence

High

Best For

Cleaning noisy boundaries

Key Takeaway

Choose based on your priorities: speed, intelligence, or boundary quality.

Best Practices & Recommendations

When to Use Undersampling

- Large majority class datasets
- Computational constraints
- When removing samples is acceptable

Implementation Tips

- Use cross-validation to assess impact
- Try multiple techniques and compare
- Start with Random for baseline
- Use imbalanced-learn library

Important Considerations

- May lose valuable information
- Always evaluate on holdout set
- Consider combining with oversampling
- Monitor for underfitting

Quick Decision Guide

Need speed? → Random

Need precision? → NearMiss

Need clean boundaries? → Tomek Links

Remember: The best technique depends on your specific dataset and problem!

Oversampling Techniques

Random Oversampling

Duplicates minority class samples randomly until balance is achieved.

SMOTE (Synthetic Minority Over-sampling)

Creates synthetic samples by interpolating between minority class neighbors.

ADASYN (Adaptive Synthetic Sampling)

Adaptively generates more samples in harder-to-learn regions.

Oversampling Techniques

Handling Imbalanced Datasets in Machine Learning

Methods to Enhance Minority Class Representation

What is Oversampling?

The Approach

Oversampling increases the minority class by adding samples until balance is achieved.

Key Advantages

- No information loss
- Better for small datasets
- Improves model learning

Unlike undersampling, oversampling preserves all original data.

The Transformation

Before

Majority: 950 samples
Minority: 50 samples

After Oversampling

Majority: 950 samples
Minority: 950 samples

Random Oversampling

How It Works

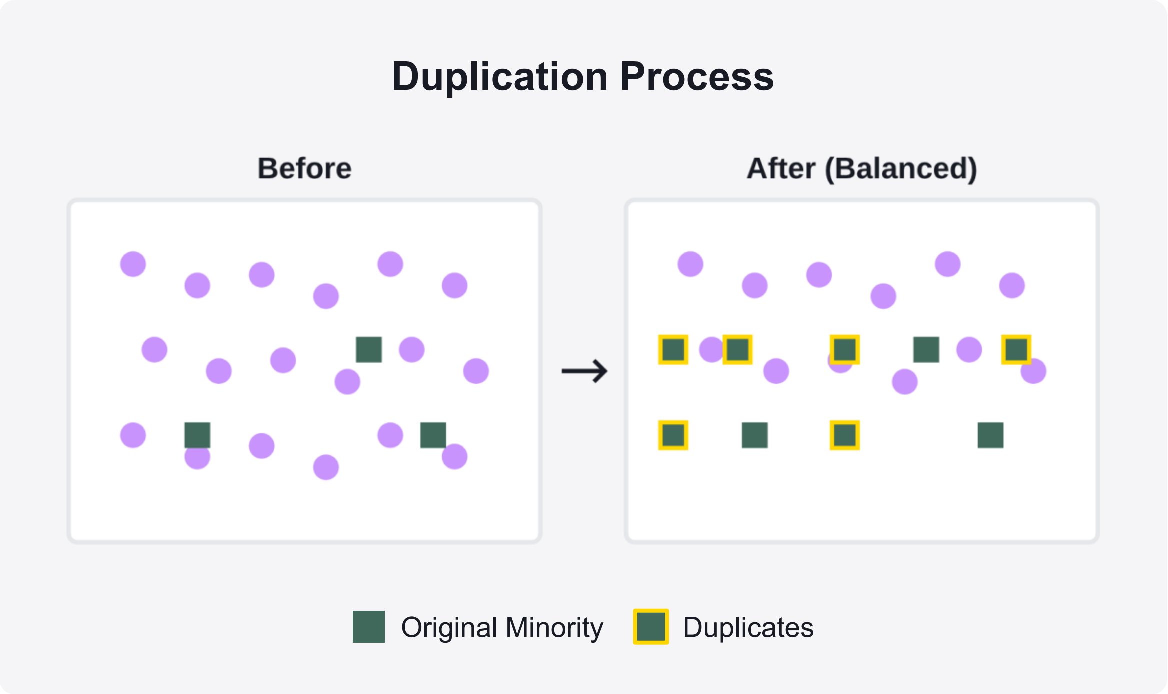
Randomly duplicates minority class samples until balance is achieved.

Advantages

- Simple and fast
- No data loss

Disadvantages

- Risk of overfitting
- Exact duplicates



SMOTE Algorithm

Synthetic Minority Over-sampling Technique

How It Works

Creates synthetic samples by interpolating between minority class neighbors.

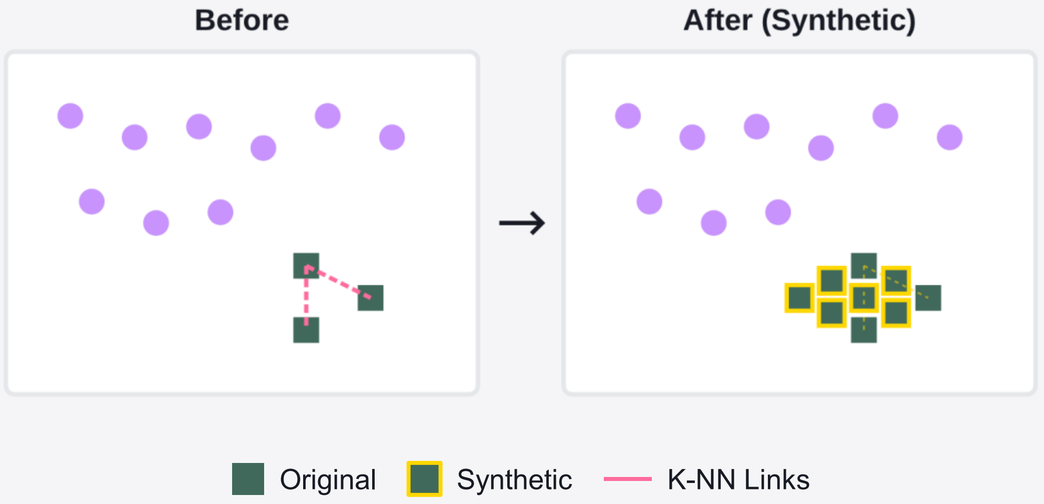
Process Steps

- Find k-nearest neighbors
- Select random neighbor
- Interpolate new sample

Key Benefits

- No exact duplicates
- Reduces overfitting

Interpolation Between Neighbors



ADASYN Algorithm

Adaptive Synthetic Sampling

How It Works

Adaptively generates more samples in harder-to-learn regions.

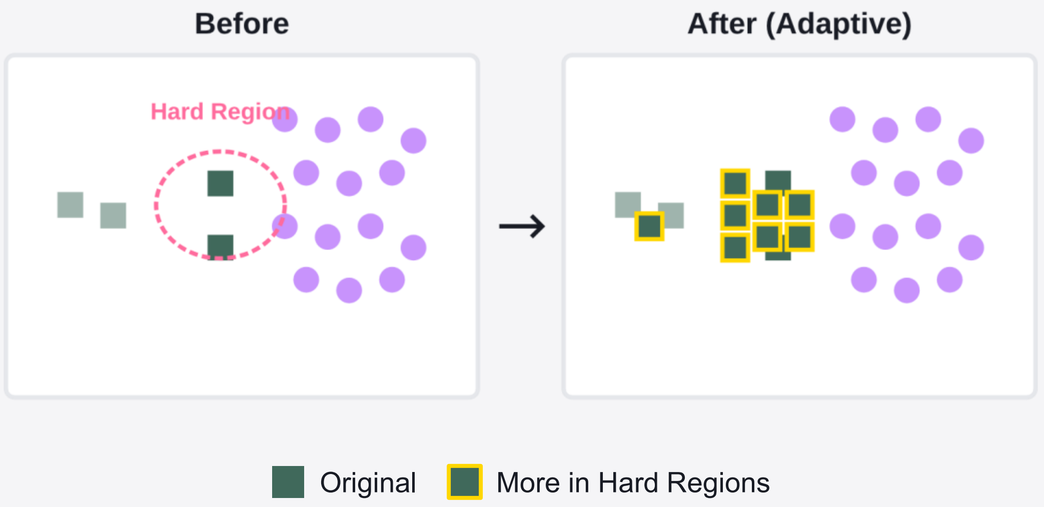
Key Concept

Focuses on difficult boundary regions where minority samples are surrounded by majority.

Advantages

- Intelligent distribution
- Improves decision boundaries

Adaptive Generation in Hard Regions



Comparison of Techniques

	Random	SMOTE	ADASYN
Method	Duplicates existing samples	Creates synthetic samples	Adaptive synthesis
Speed	Very Fast ⚡ ⚡ ⚡	Medium ⚡ ⚡	Medium ⚡ ⚡
Overfitting Risk	High	Lower	Lowest
Best For	Quick baseline	General purpose	Complex boundaries

Key Insight

SMOTE and ADASYN create new samples, avoiding exact duplicates.

When to Use

- Small datasets
- Severe imbalance
- Information preservation critical

Implementation Tips

- Use imbalanced-learn library
- Combine with cross-validation
- Consider hybrid approaches

Best Practices & Implementation

Critical Warnings

- Apply only to training set
- Never oversample test data
- Monitor for overfitting

Quick Guide

Simple task? → Random

General use? → SMOTE

Complex boundaries? → ADASYN

Combine oversampling with undersampling for optimal results!

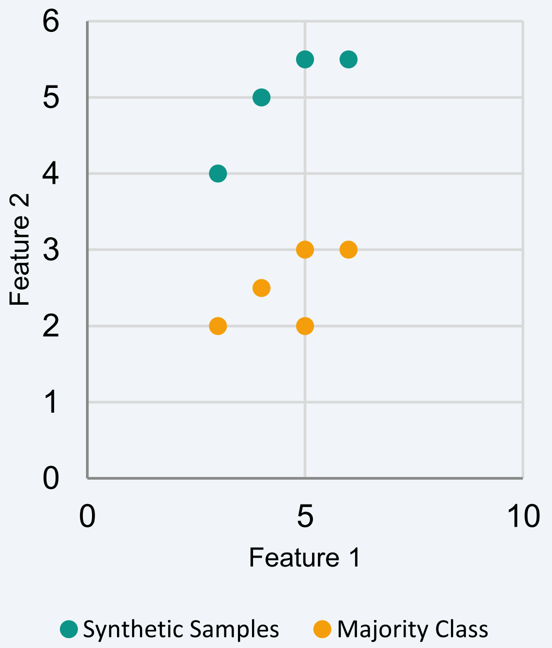
SMOTE Algorithm

How It Works

- Select a minority class sample
- Find k nearest neighbors (typically k=5)
- Choose a random neighbor
- Generate synthetic sample along the line between the two points
- Repeat until desired balance achieved

Key Advantage

Creates new samples in feature space rather than duplicating existing ones, reducing overfitting.



Hybrid Approaches

SMOTE + ENN

Combines SMOTE oversampling with Edited Nearest Neighbors cleaning.

- Generates synthetic samples then removes noise

SMOTE + Tomek Links

Combines SMOTE with Tomek link removal for boundary refinement.

- Oversamples then cleans class boundaries

Hybrid Approaches

Combining Oversampling and Undersampling

Best of Both Worlds for Imbalanced Data

Why Hybrid Approaches?

The Strategy

Hybrid methods combine oversampling and undersampling to leverage the strengths of both approaches.

Key Advantages

- Reduces overfitting from oversampling
- Minimizes information loss from undersampling
- Cleans noisy or overlapping samples

Hybrid methods often outperform single techniques.

Two-Step Process

Step 1: Oversample

Create synthetic minority samples to balance classes

Step 2: Clean

Remove noisy or ambiguous samples from boundaries

SMOTE + ENN

SMOTE with Edited Nearest Neighbors

How It Works

Step 1: SMOTE generates synthetic samples. Step 2: ENN removes misclassified samples.

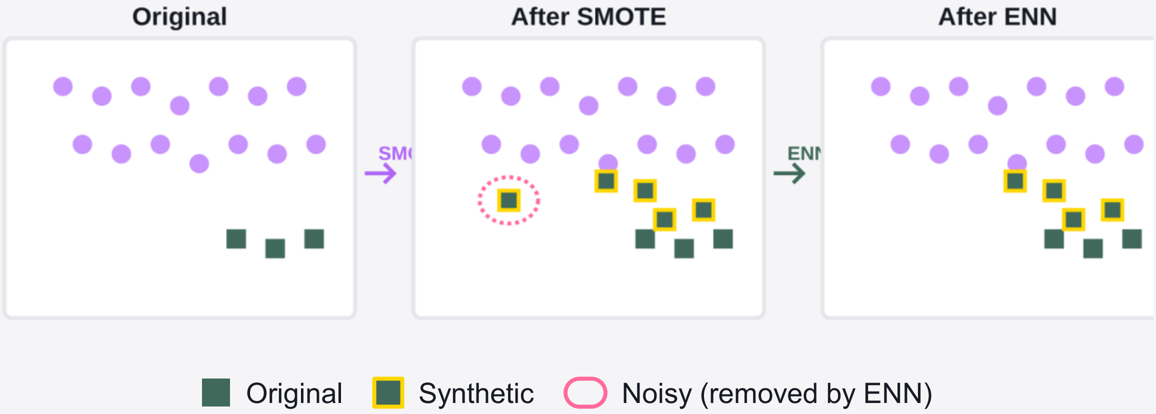
ENN Rule

Remove sample if majority of k-nearest neighbors are from different class.

Benefits

- Removes noisy samples
- Cleans overlapping regions

Two-Step Process



Edited Nearest Neighbors (ENN)

The ENN Rule

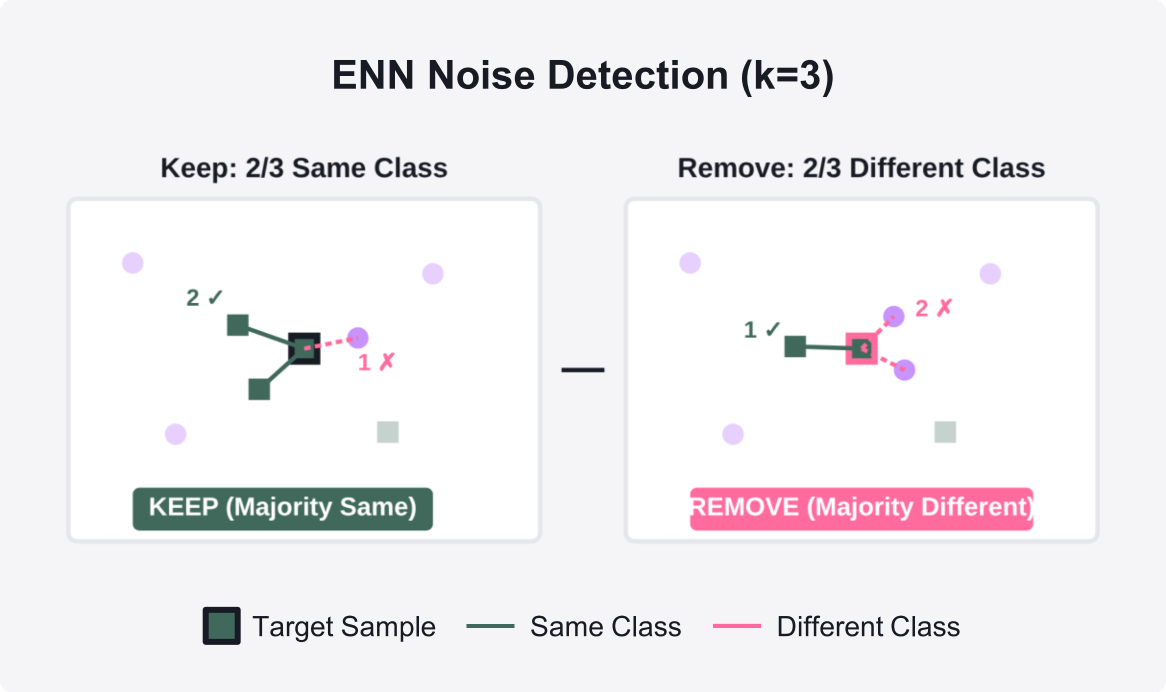
For each sample, find k-nearest neighbors.
Remove if majority are from different class.

What Gets Removed?

- •Misclassified samples
- •Noisy data points
- •Overlapping regions

Result

Cleaner decision boundaries with reduced noise.



SMOTE + Tomek Links

SMOTE with Boundary Refinement

How It Works

Step 1: SMOTE creates synthetic samples. Step 2: Tomek Links removes boundary pairs.

Tomek Link Rule

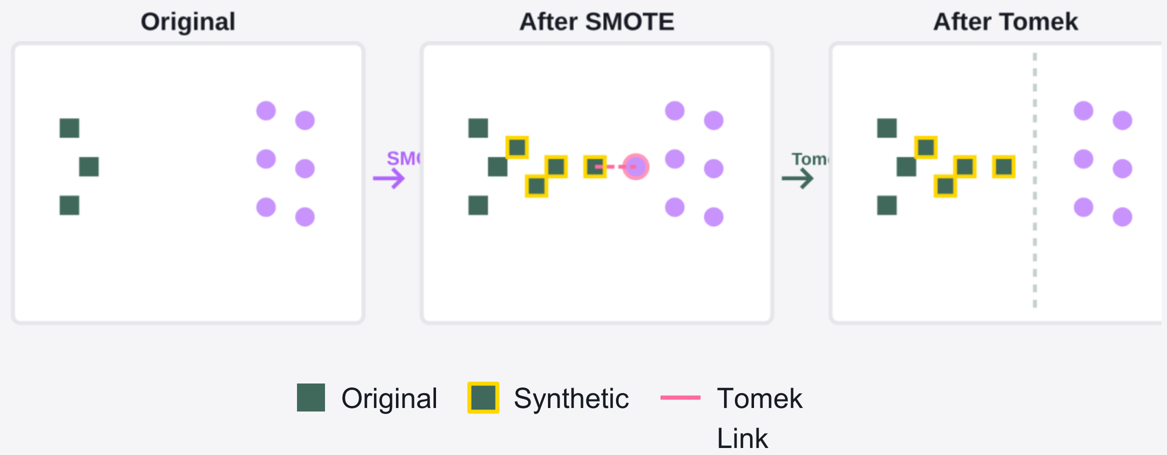
Remove majority sample if it forms Tomek link with minority.

Benefits

•Clears decision boundary

•Reduces class overlap

Boundary Cleaning Process



SMOTE + ENN

Cleaning Method

Removes misclassified samples

What It Removes

Noisy and overlapping samples from both classes

Aggressiveness

More Aggressive

Best For

Very noisy datasets

SMOTE + Tomek

Cleaning Method

Removes boundary pairs

What It Removes

Only majority samples in Tomek links

Aggressiveness

More Conservative

Best For

Boundary refinement

Comparing Hybrid Methods

Key Difference

ENN removes more samples for thorough cleaning. Tomek focuses only on decision boundary.

Both approaches:

Start with SMOTE oversampling

Both improve:

Model performance over SMOTE alone

Best Practices & Implementation

When to Use Hybrid

- Dataset has noise or outliers
- Class boundaries overlap
- SMOTE alone causes overfitting

Critical Warnings

- Apply only to training data
- ENN can be computationally expensive
- May reduce class balance achieved

Implementation Tips

- Use imbalanced-learn pipeline
- Tune k parameter for ENN
- Compare both hybrid methods

Decision Guide

Very noisy data? → SMOTE + ENN

Clean boundaries? → SMOTE + Tomek

Not sure? → Try both!

Hybrid methods consistently outperform single techniques in research!

Evaluation Metrics

Precision

Correct positive predictions

Recall

Actual positives found

F1-Score

Harmonic mean of precision and recall

AUC-ROC

Area under ROC curve

G-Mean

Geometric mean - ideal for imbalanced data

Study Methodology

Datasets

Seven benchmark datasets with imbalance ratios from 1:10 to 1:100

Classifiers

Decision Trees, Random Forest, SVM, Logistic Regression, Neural Networks

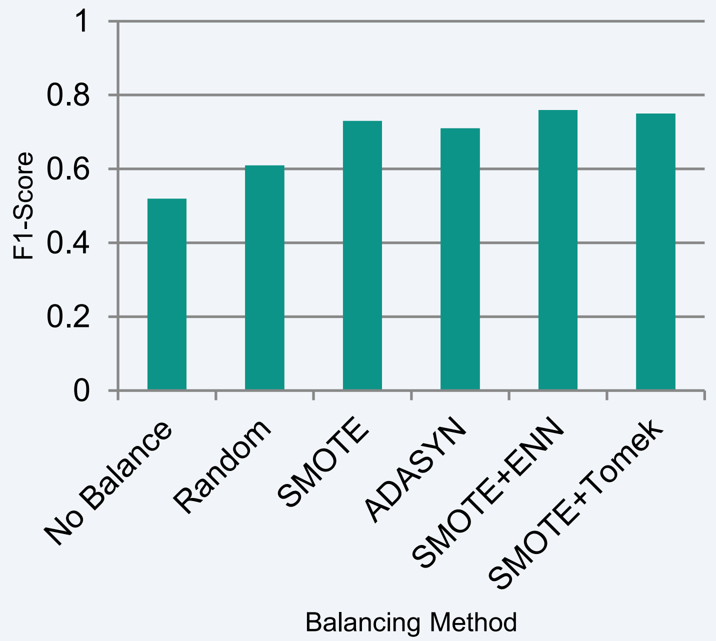
Methods Compared

Baseline, Random sampling, SMOTE, ADASYN, SMOTE+ENN, SMOTE+Tomek

Performance Comparison

Average F1-Score Across Datasets

- SMOTE consistently outperformed random sampling
- Hybrid methods showed best results on highly imbalanced data
- ADASYN performed well on complex boundary cases
- No single method dominated across all datasets



Key Findings

SMOTE variants showed 15-25% improvement in F1-score over baseline

Imbalance Ratio Matters

Hybrid methods excel with extreme imbalance ratios above 1:50

Classifier Dependency

Random Forest and SVM benefited most from balancing methods

Conclusions & Recommendations

General Recommendations

- Start with SMOTE for moderate imbalance (1:10 to 1:30)
- Use hybrid methods for severe imbalance (>1:50)
- Evaluate multiple metrics, not just accuracy

Future Directions

- Deep learning integration with balancing methods
- Adaptive strategies based on dataset characteristics

Select methods based on dataset needs and requirements