Table of Contents

4. Methodology

4.1 Custom SMOTE Implementation

SMOTE addresses class imbalance by synthesizing new minority class samples through linear interpolation between existing ones. For each minority instance, SMOTE selects k nearest neighbors, then generates points along the line segments connecting them. This diversifies the minority class while avoiding mere duplication, as seen in random oversampling. SMOTE is widely adopted due to its simplicity and effectiveness, but it has limitations: it may introduce noise by interpolating outliers, and it ignores majority class distribution, potentially creating overlapping regions. Instead of linearly interpolating new synthetic points; other interpolation methods such as cubic spline interpolation can be used here.

4.2 Distance-Weighted ENN

Presents extension of ENN with various distance-based weighting schemes (inverse, Gaussian, exponential, rank, adaptive power). Includes theoretical rationale and practical advantages. Distance-Weighted Edited Nearest Neighbors (DW-ENN) is an advanced variant of the classic ENN algorithm that enhances noise filtering by incorporating distance-based weighting schemes. Unlike traditional ENN, which removes majority class samples misclassified by their k-nearest neighbors (KNN) without considering proximity, DW-ENN assigns higher importance to closer neighbors during voting. Common weighting strategies include inverse distance (where influence decays linearly with distance), Gaussian kernels (smoother decay based on exponential squared distance), and adaptive power weighting (dynamic adjustment using a tunable exponent). These schemes reduce bias in noise removal, particularly near class boundaries, where misclassified samples may still be informative. By preserving critical majority samples while eliminating outliers, DW-ENN improves classifier generalization, especially in imbalanced datasets where majority class dominance can skew performance

4.3 SMOTEENN with Weighted Voting

Introduces hybrid pipeline combining SMOTE and distance-weighted ENN. Highlights how the two-stage process improves both balance and noise reduction. SMOTEENN with Weighted Voting is a hybrid pipeline that combines synthetic oversampling (SMOTE) and distance-weighted undersampling (DW-ENN) to address both class imbalance and noise. In the first stage, SMOTE generates synthetic minority samples to balance class distribution. The second stage applies DW-ENN to clean the augmented dataset, but instead of majority voting, it employs distance-weighted voting (e.g., Gaussian-weighted influence) to decide whether to retain or discard samples. This ensures that borderline majority samples—often misclassified due to their proximity to minority clusters—are preserved if their nearby neighbors strongly support their class label. The result is a refined dataset that improves minority class representation while mitigating noise, leading to more robust classifier performance.

4.4 SVM-SMOTE

Details how one utilizes support vector machine decision boundaries to generate synthetic minority samples, placing them in difficult or sparse regions for better learning. SVM-SMOTE enhances traditional SMOTE by leveraging Support Vector Machine (SVM) decision boundaries to guide synthetic sample generation. Unlike SMOTE, which interpolates between random minority samples, SVM-SMOTE identifies difficult-to-classify regions (e.g., near the margin or in sparse minority areas) using SVM support vectors. Synthetic samples are then strategically placed in these regions to strengthen the minority class's representation where it matters most. This approach improves classifier decision boundaries and is particularly useful for datasets with complex separability or high variance.

4.5 SMOTified-GAN and Hybrid Variants of SMOTE with other Adversial Networks (WGAN,CGAN)

Discusses use of GANs for generating minority class samples. Covers: - GAN architecture. - Conditioning strategies. - Comparison of synthetic sample quality vs. classical oversampling. - Evaluation of its impact on classification performance. Adopts Generative Adversarial Networks (GANs) to generate synthetic minority samples, offering an alternative to classical oversampling. The GAN framework consists of a generator (which creates synthetic samples) and a discriminator (which distinguishes real from synthetic data). By conditioning the generator on minority class features, SMOTified-GAN produces more realistic and diverse samples compared to SMOTE's linear interpolation. Key advantages include better handling of non-linear data distributions and reduced risk of overfitting.

4.6 Ideas

To further enhance default prediction in imbalanced datasets, optimization-based approaches can be integrated with advanced oversampling techniques.

- **Evolutionary algorithms** (e.g., Genetic Algorithms, Particle Swarm Optimization) can optimize SMOTE's hyperparameters (e.g., k-neighbors, sampling ratios) to maximize F1-score or G-mean, ensuring synthetic samples improve classifier decision boundaries without overfitting.
- **Cost-sensitive learning** can be combined with ensemble methods (e.g., Optimized XGBoost with class-weighted gradients) to penalize misclassifications of minority defaults more heavily.

For oversampling:

- **Kernel-based SMOTE** (using RBF or polynomial kernels) can generate non-linear synthetic samples, better capturing complex feature interactions in financial data.
- **Contrastive data augmentation** can create informative minority samples by perturbing real defaults in latent space.

- **Active learning** can iteratively query the most uncertain majority samples for undersampling, reducing noise.
- **Hybrid frameworks** like SMOTE + Optimal Transport can align synthetic and real data distributions more precisely, minimizing domain shift.

Finally, **Bayesian optimization** can dynamically tune resampling and classifier parameters in a unified pipeline, balancing precision-recall trade-offs for default prediction.

5. Experimental Setup and Evaluation

This chapter explains how the experiments were designed and evaluated: - Data split strategy (train/validation/test). - Evaluation metrics (recall, precision, AUC, balanced accuracy, F1-score). - Baseline models (e.g., LightGBM). - Experimental protocol for each method. - Hardware/software used (mention GPU usage for GANs, frameworks like Scikit-learn, PyTorch).

6. Results and Discussion

Presents a comparative analysis of all methods. Includes: - Quantitative results across models and sampling methods. - Visualizations: confusion matrices, ROC curves, distribution shifts. - Effectiveness of each method in handling class imbalance. - Discussion on trade-offs (e.g., increased recall vs. precision loss). - Case-specific insights and interpretation of why certain methods outperformed others.

7. Conclusion and Future Work

Summarizes the findings, highlighting the contributions of custom implementations of the algorithms. It also reflects on limitations (e.g., computational cost of GANs, potential overfitting).

References

A list of all cited works including papers on SMOTE, ENN, GANs, and evaluation metrics.

Appendices

- Code snippets or pseudocode of implementations. - Extended tables of results. - Additional figures (e.g., sample visualizations from SMOTified-GAN). - Hyperparameter grids or search spaces used for tuning.