# Unbalanced Datasets

Matthew Rui

December 2024

# Outline

Introduction

Alternative Performance Metric

Undersampling

Oversampling

Focal Loss

Experiment

Conclusion

Appendix

# Unbalanced Datasets

- Many classification tasks assume classes are evenly represented.
- Real-world datasets often have imbalanced class distributions.
- Minority classes may be critically important (e.g., cancer detection).

# Limitations of Accuracy

- In unbalanced datasets, accuracy can be misleading.
- Baseline accuracy may be high by always predicting the majority class.
- Need metrics focusing on minority class performance.
  - **Precision**: True positives out of all predicted positives.
  - **Recall**: True positives out of all actual positives.

# Precision-Recall Curve (PRC)

- PRC demonstrates the trade-off between precision and recall for various **classification thresholds**.
- **AUPRC** (Area Under the PRC): provides a single performance metric.

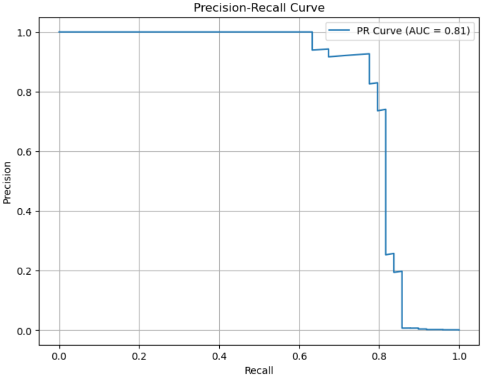


Figure: Sample Precision-Recall Curve (PRC)

# Undersampling Techniques

- Aim to balance the dataset by reducing majority class instances.
- **Random Undersampling**:
  - Randomly remove majority instances.
  - Risk of discarding important information.
- **Tomek Links**:
  - For majority instance, remove if nearest neighbor is a minority instance and if the closest neighbor of the minority neighbor is the majority instance.
  - Cleans the class boundary.
- **KNN**:
  - For majority instance, remove if more than $t$ of its $k$ nearest neighbors are minority instances.
  - Generalizes Tomek Links.

# Oversampling Techniques

- Balance the dataset by increasing minority class instances.
- **Random Oversampling**:
  - Duplicate minority instances.
  - Leads to overfitting.
- **SMOTE** (Synthetic Minority Over-sampling Technique):
  - Generate synthetic minority instances by interpolation.
  - Reduces overfitting compared to duplication.
  - For each minority instance $x$:
    - Find $k$ nearest minority neighbors.
    - Randomly select a neighbor $x_i$.
    - Generate new instance:

$$x_{\text{new}} = x + \lambda(x_i - x), \quad \lambda \sim U(0, 1)$$

# Focal Loss for Deep Learning

- Modify cross-entropy loss to focus on hard-to-classify instances.
- Definition:

$$FL = -\alpha_t (1 - p_t)^\gamma \log(p_t)$$

- Where:
  - $p_t$: Predicted probability for the true class.
  - $\gamma$: Modulating factor to down-weight easy examples.
  - $\alpha_t$: Balancing factor for class imbalance.

# Dataset and Preprocessing

- Credit card transactions dataset.
- 284,807 instances; 492 (0.17%) are fraudulent.
- Features:
  - 28 PCA (Principal Component Analysis) components.
  - Transaction amount.
  - Time elapsed.
- Data split:
  - Stratified train/test split.
  - Over/undersampling applied only on training set.
  - Data normalization.

# Model Training

- Neural network classifier with 3 hidden layers.
- Optimizer: Population-Based Training with Adam and weight decay.
- Evaluation metric: AUPRC on the test dataset.

# Results: Over/Undersampling

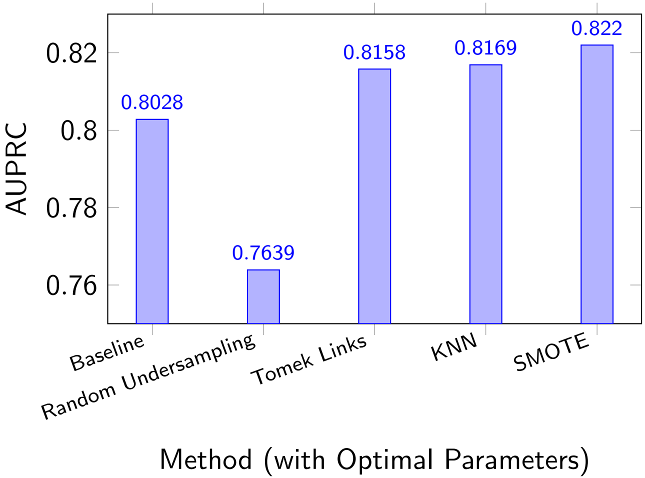


Figure: Over/undersampling Performance

# Results: SMOTE with Undersampling

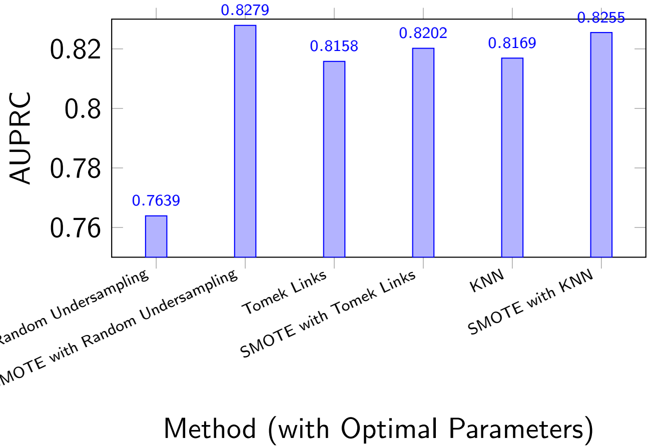


Figure: SMOTE with Undersampling Performance

# Results: Focal Loss

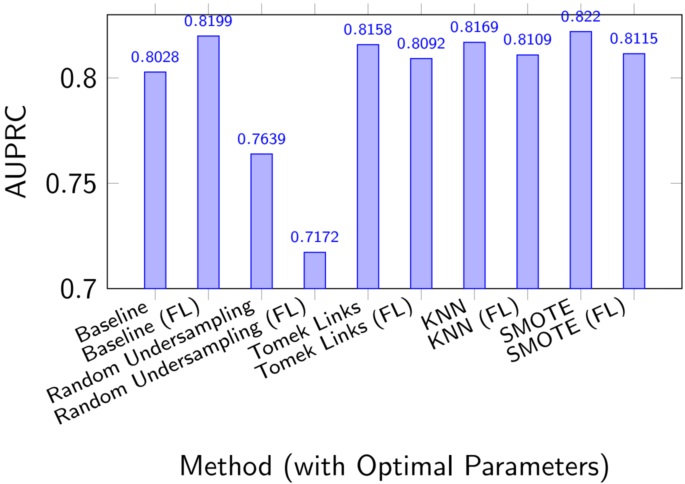


Figure: Focal Loss Performance

# Conclusion

- Oversampling (SMOTE) and undersampling (Random, Tomek Links, KNN) can improve performance.
- Focal loss provides an effective alternative by focusing on hard examples.
- Combining focal loss with sampling methods did not yield further improvements.
- For extreme imbalances, careful tuning is necessary.

# Under/Oversampling Performance

| **Method** | **Parameters** | **AUPRC** |
|---|---|---|
| Baseline | – | 0.8028 |
| Random Undersampling | $R = 1$ | 0.5551 |
| | $R = 2$ | 0.5921 |
| | $R = 3$ | 0.6470 |
| | $R = 4$ | 0.6943 |
| | $R = 5$ | 0.7188 |
| | $R = 6$ | 0.7110 |
| | $R = 7$ | 0.6973 |
| | $R = 8$ | 0.7639 |
| Tomek Links | – | 0.8158 |

# Under/Oversampling Performance

| **Method** | **Parameters** | **AUPRC** |
|---|---|---|
| KNN | $k = 50$ | 0.8165 |
| | $k = 100$ | 0.8169 |
| | $k = 150$ | 0.8007 |
| | $k = 200$ | 0.8124 |
| SMOTE | $N = 2$ | 0.8127 |
| | $N = 3$ | 0.8121 |
| | $N = 4$ | 0.8206 |
| | $N = 5$ | 0.8172 |
| | $N = 6$ | 0.8194 |
| | $N = 7$ | 0.8216 |
| | $N = 8$ | 0.8220 |
| | $N = 9$ | 0.8172 |
| | $N = 10$ | 0.8203 |

# SMOTE with Undersampling Performance

| **Method** | **Parameters** | **AUPRC** |
|---|---|---|
| SMOTE with Random Undersampling | $R = 5, N = 10$ | 0.8259 |
| | $R = 6, N = 10$ | 0.8279 |
| | $R = 7, N = 7$ | 0.8228 |
| | $R = 8, N = 8$ | 0.8181 |
| SMOTE with Tomek Links | $N = 6$ | 0.8202 |
| SMOTE with KNN | $K = 50, N = 7$ | 0.8254 |
| | $K = 100, N = 9$ | 0.8255 |
| | $K = 150, N = 10$ | 0.8190 |
| | $K = 200, N = 7$ | 0.8183 |

# Focal Loss Tuning

| Method | Parameters | AUPRC |
|---|---|---|
| Baseline | $\gamma = 0.1, \alpha = 0.75$ | 0.8116 |
| | $\gamma = 0.2, \alpha = 0.25$ | 0.8199 |
| | $\gamma = 0.5, \alpha = 0.25$ | 0.8194 |
| | $\gamma = 1.0, \alpha = 0.25$ | 0.8197 |
| | $\gamma = 2.0, \alpha = 0.5$ | 0.8070 |
| | $\gamma = 5.0, \alpha = 0.5$ | 0.7908 |

# Over/Undersampling with Focal Loss

| Method | Parameters | AUPRC |
|---|---|---|
| Random Undersampling (Focal Loss) | $R = 1$ | 0.6079 |
| | $R = 2$ | 0.6902 |
| | $R = 3$ | 0.6522 |
| | $R = 4$ | 0.7062 |
| | $R = 5$ | 0.7108 |
| | $R = 6$ | 0.7115 |
| | $R = 7$ | 0.7133 |
| | $R = 8$ | 0.7172 |
| Tomek Links (Focal Loss) | – | 0.8092 |

# Over/Undersampling with Focal Loss

| **Method** | **Parameters** | **AUPRC** |
|---|---|---|
| KNN (Focal Loss) | $K = 50$ | 0.8062 |
| | $K = 100$ | 0.8109 |
| | $K = 150$ | 0.8070 |
| | $K = 200$ | 0.8006 |
| SMOTE (Focal Loss) | $N = 2$ | 0.7980 |
| | $N = 3$ | 0.8054 |
| | $N = 4$ | 0.7961 |
| | $N = 5$ | 0.8115 |
| | $N = 6$ | 0.8097 |
| | $N = 7$ | 0.8026 |
| | $N = 8$ | 0.8048 |
| | $N = 9$ | 0.8127 |
| | $N = 10$ | 0.7920 |