Text Augmentation using LLMs

Improving the performance on text classification by generating augmented text using LLMs for under-represented categories

Sejal Agarwal
UMass, Amherst
sejalagarwal@umass.edu

Siddharth Jain
UMass, Amherst
siddharthjai@umass.edu

Abstract

Imbalanced datasets pose significant challenges in text classification, often leading to biased models that underperform on minority classes. This project explores the potential of Large Language Models (LLMs) like GPT-2 for text augmentation to address class imbalances in the AG News dataset. The dataset is deliberately imbalanced by reducing instances in the Science/Technology category, simulating real-world conditions. We employ 4 imbalance creating techniques, such as severe under-sampling, topic-specific sampling, clustered minority sampling, and progressive rarity imbalances. We use traditional methods like Synthetic Minority Oversampling Technique (SMOTE) as our baseline model to handle the imbalance.

LLM-based augmentation is implemented through context-aware prompts designed for each specific imbalance scenarios, generating synthetic text to enrich minority class data. Models such as Logistic Regression and Support Vector Machines (SVM) are trained and evaluated on both imbalanced and augmented datasets using metrics like accuracy, precision, recall, and F1-score.

The results demonstrate that LLM-based augmentation slightly improves classification performance for underrepresented categories compared to traditional methods, particularly in recall and F1-score for minority classes. Further work in this area will highlight the utility of LLMs in addressing data imbalance challenges, offering a scalable and contextually rich solution for enhancing text classification tasks in domains such as healthcare, finance, and natural language processing.

1. Introduction

1.1. Motivation

Text classification is a critical task in natural language processing (NLP) that entails categorizing textual inputs into predefined categories. This capability underpins a range of applications, such as spam detection, sentiment analysis, topic labeling, and more. Despite its foundational role in NLP, real-world datasets often suffer from **class imbalance**, where certain categories are underrepresented compared to others. This imbalance poses significant challenges:

- **Performance Degradation:** Classification models tend to bias predictions toward majority classes, resulting in poor generalization for minority categories.
- **Skewed Metrics:** Disparities in class representation can inflate metrics like accuracy, masking the true performance on underrepresented classes.
- **Real-world Implications:** Fields such as healthcare, finance, and autonomous systems require robust classification across all classes, including rare but critical cases.

To address this, techniques that enhance representation for minority classes are essential. Recent advancements in **large language models (LLMs)**, such as GPT-2 and BERT, present an opportunity to generate synthetic, contextually coherent data, potentially mitigating these challenges.

1.2. Objectives

The primary goal of this project is to investigate the efficacy of **LLM-based data augmentation** in improving classification performance on imbalanced datasets. Specific objectives include:

- **Simulating Real-world Imbalances:** Use the AG News dataset to create controlled class imbalances, emphasizing underrepresented categories. Analyze the effect of these imbalances on classification model performance, establishing a baseline.
- **Experimenting with Balancing Techniques:** Employ traditional methods like SMOTE to artificially balance datasets. Leverage LLMs to generate synthetic text for minority classes, ensuring contextual relevance and diversity.
- **Performance Evaluation:** Train models such as Logistic Regression and Support Vector Machines (SVM) on imbalanced, SMOTE-augmented, and LLM-augmented

datasets. Compare performance using metrics like **accuracy**, **precision**, **recall**, and **F1-score**, with a focus on improvements for minority classes.

1.3. Dataset

The project utilizes the **AG News dataset**, a benchmark corpus widely used in text classification tasks. Key details about the dataset are as follows:

- **Source:** The balanced dataset consists of news articles categorized into four high-level topics: World, Sports, Business, Science/Technology.
- **Structure:**
  - Training Set: 120,000 samples (30,000 per category)
  - Test Set: 7,600 samples (1,900 per category)
- **Characteristics:**
  - Each sample includes a headline and a short text body, making it suitable for both short-text and multi-sentence classification tasks.
  - Class distributions in the original dataset are balanced, ensuring an even representation of all categories.

Intentional class imbalances were introduced, with a focus on the Science/Technology category, which was underrepresented to simulate real-world scenarios. This deliberate modification allows the project to evaluate the effectiveness of various balancing techniques and their impact on minority class performance.

2. Related work

2.1. Literature Survey

[1] Cegin et al. (2024) explored when LLMs surpass traditional methods for text classification, revealing that while LLMs generate more semantically rich, diverse samples, they incur significantly higher computational costs. [2] Dai and colleagues (2023) introduced AugGPT, an augmentation tool leveraging ChatGPT for generating synthetic text data aimed at improving classification accuracy in NLP tasks in low-resource environments. [3] Wei and Zou (2019) proposed Easy Data Augmentation (EDA) techniques, which are simple, cost-effective, and enhance model robustness by introducing random synonym replacements, swaps, insertions, and deletions. Despite their success, EDA techniques lack semantic awareness, often introducing inconsistencies when word replacements change sentence meaning unintentionally. [4] Ubani et al. (2023) used ChatGPT in ZeroShotDataAug to generate training data without labeled examples, focusing on zero-shot classification. [5] Misra's (2022) News Category Dataset has become a key resource in testing text augmentation methods, offering annotated news articles across various categories to evaluate model robustness in diverse classification tasks. However, it is limited by its static nature, making it less effective for adapting models to emerging or novel topics in real time.

2.2. Comparison of Baseline Methods and Our Approach

Our study builds on existing work addressing class imbalance in text classification by comparing traditional methods like SMOTE with advanced LLM-based augmentation techniques. While SMOTE improved recall and F1-scores, particularly for underrepresented classes, it struggled with nuanced imbalances, consistent with findings by Cegin et al. (2024) and Wei and Zou (2019). In contrast, LLM-based augmentation using GPT-2 demonstrated superior performance, achieving a recall of 88.4% in progressive rarity scenarios, surpassing SMOTE. These results align with Dai et al. (2023)'s insights on the contextual richness of LLMs, highlighting their effectiveness in addressing complex imbalances. Our findings underscore the potential of LLMs to generate semantically diverse data and outperform traditional techniques in critical areas like minority class recognition.

2.3. Weakness

In our project, we aim to address the weaknesses of established baseline methodologies in text classification, particularly regarding class imbalance. While studies like those by [1] Cegin et al. (2024) and Dai et al. (2023) showcase the effectiveness of advanced models like BERT and transformers, they often overlook the performance drop for under-represented categories. Our approach will emphasize augmenting training data for these categories using large language models (LLMs) to enhance classification accuracy across all classes.

3. Methodology

3.1. Data Preparation

This section describes the key steps taken to prepare the AG News dataset for model training and evaluation, including handling class imbalance and preprocessing the text data. The AG News dataset was loaded using the Hugging Face datasets library. It consists of four categories: World, Sports, Business, and Science/Technology.

Text data underwent the following preprocessing steps: conversion to lowercase, removal of digits, punctuation, and extra spaces, and elimination of duplicate and missing values. The dataset was split into features (`X`) and labels (`y`) for both training and testing sets. **TF-IDF vectorization** was used to transform the text into numerical features, limiting to the top 5,000 features based on term frequency and inverse document frequency.

3.2. Creating Imbalance

In this project, we created an imbalanced dataset by intentionally manipulating the distribution of classes within the

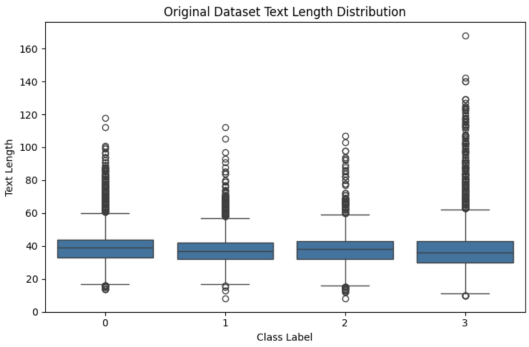

Figure 1. Text length distribution in original balanced dataset

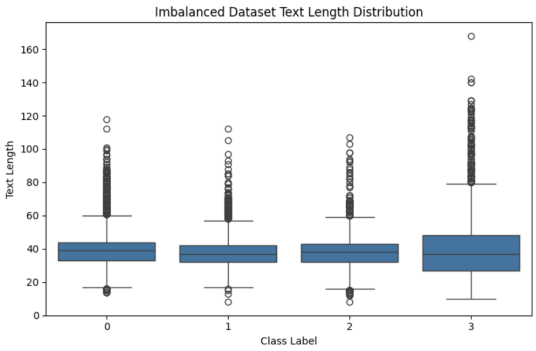

Figure 2. Text length distribution in dataset with clustered minority instances

AG News dataset. We employed the following four techniques to introduce and study the effects of imbalance:

3.2.1 Severe Under-sampling Across Multiple Classes

This technique significantly reduces instances from selected classes while maintaining full representation of others. In our case, the Science/Technology class was reduced to 60%, creating a pronounced imbalance. This approach assesses a model's adaptability to learn from fewer examples.

3.2.2 Clustered Minority Instances with Diverse Text Lengths

This method involves grouping minority class instances based on characteristics such as text length. A specified cluster size determines how many samples to keep from each group, ensuring diversity within the underrepresented class while limiting overall representation. It is useful for maintaining variability in the minority class. In our case, we have considered the Science/Technology class to be a minority class.

The broader range in the imbalanced dataset (figure 2) helps maintain variability within the minority class, which can improve the model's ability to handle unseen data from this class during evaluation. A larger IQR also suggests that the minority class retains a broad range of text lengths, which is crucial for ensuring that the model does not become biased toward a narrow set of samples.

3.2.3 Topic-Specific Under-sampling within a Class

In this technique, we focus on specific themes to selectively remove instances from a class based on predefined topic-related keywords. To determine these prominent topics, we employ Latent Dirichlet Allocation (LDA), which clusters thematic content within the dataset. For our study, we focused on the Science/Technology class, identifying 'internet' as a significant theme. Using LDA, we extracted topic-specific keywords such as *'web', 'internet', 'new', 'sites', 'online', 'site', 'mars', 'computer', 'says', 'one'*. We then systematically removed instances in the Science/Technology class containing a high frequency of these terms, resulting in an intentionally imbalanced dataset. This approach enables us to evaluate the model's performance on distinct thematic content and observe its adaptability to class imbalances centered on particular topics.

3.2.4 Progressive Rarity Imbalance (Simulating Long-Tail Distribution)

This approach selectively reduces instances in specified classes to create a long-tail distribution while preserving common classes. Only a defined fraction (e.g., 10%) of samples from target classes is retained, resulting in disproportionate representation. This technique evaluates how well a model can handle extreme class imbalances.

3.3. Applying SMOTE to balance the datasets

In our project, we implemented SMOTE as a standard baseline technique to handle class imbalance in the AG News dataset. This approach generates synthetic instances of the minority classes by interpolating between existing examples, effectively enriching the dataset. By doing so, we provide the model with more opportunities to learn from these under-represented categories.

Using SMOTE helps to create a more balanced representation of the data without the redundancy issues associated with simple oversampling. It enables the model to learn from a more diverse set of examples, which is crucial for improving performance on minority classes. The effectiveness of SMOTE is well-documented in the literature. [1] Cegin et al.(2024) analyze the computational costs and benefits of various augmentation methods, positioning SMOTE as a reliable approach for addressing imbalance. By comparing the results of models trained on datasets augmented with SMOTE against those augmented with LLM-generated examples, we aim to assess the relative performance and

benefits of these two strategies. This comparison will provide insights into the potential advantages of using LLMs for data augmentation in overcoming the limitations of traditional methods. The steps involved are as follows:

3.3.1 Vectorization of the Training Data

The raw text data was first transformed into numerical features using TF-IDF vectorization. This step was necessary to convert the text data into a format that can be processed by machine learning algorithms.

- **Why Vectorization?** Machine learning algorithms, including SMOTE, cannot operate directly on raw text data; thus, vectorization is required to represent the text in a structured, numerical format.
- The data was represented as a sparse matrix to ensure computational efficiency, as sparse matrices are memory-efficient and suitable for high-dimensional data.

3.3.2 Applying SMOTE to Sparse Data

SMOTE was applied directly to the sparse matrix, avoiding the need to convert it into a dense format, which would have incurred high memory costs. The following parameters were used:

- `random_state=42` for reproducibility.
- `k_neighbors=5`, the default value, defines the number of nearest neighbors used for generating synthetic samples.

SMOTE works by selecting the minority class samples, finding their nearest neighbors in feature space, and generating synthetic samples by interpolating between the original and neighbor samples. The mathematical formulation for generating a synthetic sample is as follows:
Given a minority class instance $x_i = (x_{i1}, x_{i2}, \ldots, x_{in})$, where x_i is the feature vector of the i^{th} sample, and its k nearest neighbors $\{x_1, x_2, \ldots, x_k\}$ are identified in the feature space, a synthetic sample x_{new} is generated as:

$$x_{\text{new}} = x_i + \lambda \cdot (x_{\text{nn}} - x_i)$$

where: - x_{new} is the synthetic sample, - x_i is the original sample, - x_{nn} is a randomly chosen nearest neighbor from the set of k neighbors, - λ is a random scalar factor drawn from the interval $[0, 1]$. The term $\lambda \cdot (x_{\text{nn}} - x_i)$ defines the interpolation between the original sample and its neighbor, and the factor λ ensures diversity in the synthetic samples. This interpolation generates a synthetic instance that is similar to the original sample but varies depending on the chosen nearest neighbor and the random scalar λ.

3.3.3 Converting Resampled Data Back to Sparse Format

After applying SMOTE, the resampled data was converted back into sparse matrix format to retain memory efficiency.

3.3.4 Model Training

The resampled, SMOTE-balanced dataset was used to train a Logistic Regression model. The vectorized data (`X_resampled_sparse`) and corresponding labels (`y_resampled`) were used to train the model.

3.3.5 Summary

In summary, this methodology aimed to improve the model's performance on under-represented categories by addressing the imbalance without introducing redundancy.

3.4. Text Augmentation using LLM

To address the imbalance in the AG News dataset, we augmented the minority class using synthetic text generation with a contextually prompted LLM pipeline. The approach enriched the dataset with diverse, semantically coherent examples, complementing the traditional SMOTE-based methods. Below is a detailed explanation of the steps and rationale behind the implementation:

3.4.1 Model Selection and Initialization

We used the **DistilGPT-2** model, a smaller and faster version of GPT-2, for text generation.

- **Rationale:** DistilGPT-2 balances efficiency and text generation quality, making it suitable for iterative experiments.
- **Hardware Optimization:**
  - Dynamically set the device to GPU or CPU
  - Used `torch.float16` for faster computation and reduced memory usage on GPUs.

The text generation pipeline was initialized using Hugging Face's `pipeline`, specifying the `text-generation` task.

3.4.2 Contextual Prompting

To guide the LLM in generating relevant synthetic data, we designed **context-specific prompts** tailored to the type of class imbalance:

- **Prompt Design:** The `generate_prompt()` function dynamically crafted prompts based on the imbalance type. For instance:
  - *Severe Under-sampling:* Focused on unique aspects of the input text.

  - *Clustered Minority Instances:* Encouraged clustering around key ideas.
  - *Topic-Specific Under-sampling:* Highlighted nuances within specific topics.
  - *Progressive Rarity Imbalance:* Created detailed text emphasizing rare aspects.
- **Example Prompt:** For a `Science/Technology` text:

```
Input text: "NASA's
Mars mission achieved a
breakthrough."
Task: Generate a similar text
that highlights unique aspects
of the topic.
```

3.4.3 Conversion to Hugging Face Dataset

The minority class data was extracted from the DataFrame and converted to a Hugging Face `Dataset`.

- **Why Hugging Face Dataset?**
  - Supports efficient **batch processing** and **mapping functions**, critical for large-scale text augmentation.
  - Enables direct integration with Hugging Face pipelines, simplifying workflow.

3.4.4 Map Function for Batch Text Generation

We used the `map()` function to process the minority class data in batches, ensuring scalability:

- **Prompt Creation:** Generated a prompt for each text instance in the batch using `generate_prompt()`.
- **Text Generation:** Passed the prompts to the `generator` pipeline with the following parameters:
  - `max_new_tokens=25`: Limited the output length to maintain coherence.
  - `num_return_sequences=n_augmentations`: Controlled the number of synthetic samples/input text.
  - `pad_token_id` and `eos_token_id`: Managed padding and end-of-sequence tokens for uniform outputs.
- **Output Handling:**
  - Extracted the generated texts and repeated the minority class label for each sample.
  - Processed batches of size `64` for memory efficiency and speed.

3.4.5 Augmented Dataset Construction

The augmented data was re-integrated into the original dataset by converting it to a DataFrame, concatenating with the original, and maintaining consistency in structure for the machine learning pipeline.

3.5. Model Training

Logistic Regression and Support Vector Machines (SVM) models were trained on the previous mentioned datasets.

3.5.1 Logistic Regression with TF-IDF for Text Classification

To implement Logistic Regression with TF-IDF features for text classification, the following steps were undertaken:

Text Data Preprocessing and Feature Extraction
The pre-processed text data was converted into numerical features using the TF-IDF vectorization technique. This approach enabled the model to capture the importance of words in each document relative to the entire dataset. By assigning higher weights to terms that were unique to specific categories, the model's ability to distinguish between different news topics was enhanced. The resulting TF-IDF features served as input for the Logistic Regression model, which was then trained to classify news articles based on their content.

Model Training and Evaluation
Once the Logistic Regression model was trained, its performance was evaluated by making predictions on a separate test set. Several key metrics were computed, including accuracy, F1 score, precision, and recall, to assess the model's effectiveness, particularly in the context of class imbalance. These metrics provided insights into how well the model performed across different categories, highlighting any challenges it faced with under-represented classes. By analyzing the results, areas for improvement were identified, and the performance of Logistic Regression was compared against other classification models, aiming to enhance the accuracy and robustness of the text classification efforts.

3.5.2 SVM Model with TF-IDF for Text Classification

To implement Support Vector Machine with TF-IDF features for text classification, the following steps were undertaken:

Text Data Preprocessing and Feature Extraction
The AG News dataset was prepared for training the Support Vector Machine (SVM) model to improve text classification accuracy. Initial preprocessing involved handling the imbalanced training set without augmentation. Synthetic Minority Over-sampling Techniques (SMOTE) and LLM-generated data were later employed to enrich the dataset with additional samples from minority classes, addressing class imbalance. The training data was transformed into a suitable feature representation, ensuring compatibility

with the SVM model. Various hyperparameters, such as kernel type (linear, polynomial, radial basis function) and regularization parameters, were experimented with to optimize the feature extraction process and subsequent model performance.

Model Training and Evaluation

The SVM model was trained on the processed dataset, learning decision boundaries to classify the four news categories effectively. After the initial training, the model was evaluated on the test set using metrics like accuracy, precision, recall, and F1-score, with a particular focus on underrepresented classes. Following the application of augmentation techniques, the SVM was retrained and reevaluated, allowing for a comparison of pre- and post-augmentation performance. The results were analyzed to document the impact of augmentation on the model's ability to address class imbalance, providing insights for future research on improving text classification systems.

3.5.3 Hyperparameter Tuning with GridSearchCV

GridSearchCV was employed to systematically explore combinations of hyperparameters. It ensured that the model was tuned to its optimal configuration, balancing regularization and bias-variance trade-offs, while also addressing class imbalance. The models were fine-tuned using GridSearchCV with a focus on optimizing key hyperparameters:

- **C**: Regularization strength values of [0.1, 1, 10].
- **penalty**: Regularization applied (`l2`).
- **solver**: Optimized solver (`saga`) for large datasets with sparse inputs.
- **class_weight**: Balanced to address the skewed class distribution.

3.5.4 Cross-Validation Strategy

A StratifiedKFold cross-validation strategy with 5 splits was implemented to:

- Preserve the class distribution in each fold, ensuring minority classes were adequately represented.
- Avoid overfitting by testing on multiple subsets of the training data.
- Provide robust estimates of model performance across all categories.

3.5.5 Model Evaluation Metrics

The models were trained on TF-IDF-transformed training data and evaluated on the test set using:

- **Accuracy**: To measure overall performance.
- **F1-score (macro)**: To account for class imbalance by considering precision and recall equally across all classes.

3.5.6 Model Saving

The best-performing Logistic Regression and SVM models were saved using the joblib library for future use.

4. Results & Analysis

This section presents the experimental results obtained from our classification models, highlighting the impact of class imbalance, baseline augmentation techniques, and LLM-based text augmentation on model performance. The findings are structured into subsections to provide a detailed evaluation of each stage of the experimentation.

4.1. Performance on the Original Balanced Dataset

On the originally balanced dataset, both models achieved the highest scores across all metrics. Particularly 92% for Precision with 92.6% for Logistic Regression and 91.7% for SVM. This baseline establishes a reference for the expected performance under ideal data conditions, providing a benchmark for subsequent experiments such as augmented datasets created using SMOTE and LLM with artificially created imbalanced datasets.

4.2. Impact of Artificially Created Imbalance

To evaluate the effect of class imbalance, we artificially reduced the samples for class label 3 using four different techniques. Under these scenarios, the classification performance for class label 3 dropped significantly. Imbalanced Logistic Regression under Clustered Minority Instances has a Recall of 0.302%. The metrics for other class labels also deteriorated. This substantial performance reduction highlights the sensitivity of both models to minority classes. It is important to note that, during this phase, no hyperparameter tuning or fine-tuning was applied. The performance degradation can be attributed to factors such as:

- **Bias toward the majority class** during imbalance.
- **Loss function misalignment,** which does not account for class imbalance.
- **Distortion of decision boundaries** due to the reduced minority class samples.
- **Overfitting to the minority class,** which can occur when the model learns from fewer samples in the imbalanced setting.

4.3. Hyperparameter Tuning and Fine-Tuning of Models

After we conducted hyperparameter tuning using GridSearchCV, the process identified optimal class weights for the minority class (class label 3) and loss functions that prioritize the minority class without excessively penalizing the majority class. After applying these optimized hyperparameters, we observed the following changes:

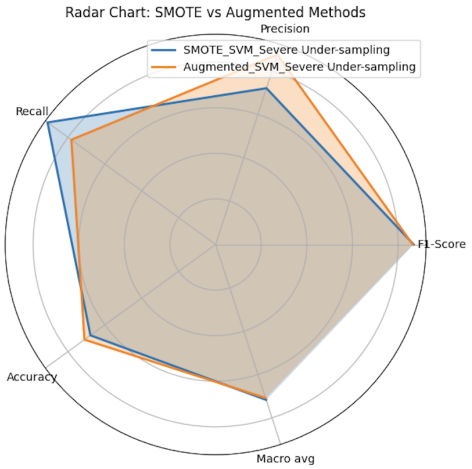

Figure 3. SMOTE vs Augmented Methods Radar Chart

- Metrics for the balanced class labels improved compared to the imbalanced scenario.
- However, metrics for class label 3 in the imbalanced datasets (all four techniques) remained consistently low.

4.4. Baseline Augmentation Using SMOTE

After augmenting the data using SMOTE, the performance metrics for class label 3 improved across all four techniques. The evaluation metrics for class label 3 increased overall, demonstrating the effectiveness of SMOTE in improving model performance. However, despite this improvement, the performance metrics for class label 3 were still far from those observed in the original balanced dataset. This result motivated the exploration of LLM-based augmentation as a potentially superior approach.

For instance:

- SMOTE with Logistic Regression under Severe Undersampling achieves an F1-Score of 0.805 but falls short in Recall (0.703) compared to LLM methods.
- SMOTE struggles in scenarios involving nuanced class imbalances, as seen in lower Macro and Weighted Averages.

4.5. Augmentation Using GPT-2 (LLM)

We implemented text augmentation using GPT-2 for the minority class (class label 3) across all four imbalance techniques. GPT-2 was chosen due to its computational feasibility and ability to generate contextually relevant text. LLMs consistently outperform their SMOTE counterparts in metrics like F1-Score and Recall. After augmenting the minority class with GPT-2, the models achieved significant improvements:

- SVM with LLM-Augmented Progressive Rarity Imbalance achieves the highest Recall 88.4% among augmented methods, indicating its effectiveness in addressing imbalanced datasets.
- The ability of LLMs to generate contextually relevant synthetic samples likely contributes to these higher scores.
- The performance metrics for the augmented datasets were now closer to those of the original balanced dataset, confirming the effectiveness of LLM-based augmentation.

4.6. Analysis

1. The original Logistic Regression and SVM models have the highest scores across all metrics.
2. Models trained on imbalanced datasets perform poorly across all metrics, especially Recall, with scores often below 0.5.
3. SMOTE struggles in scenarios involving nuanced class imbalances, as seen in lower Macro and Weighted Averages.
4. Logistic Regression models generally perform better than SVM models in augmented scenarios, particularly for Precision and Accuracy.
5. However, SVM models with LLM augmentation perform competitively, especially in Recall, demonstrating their ability to handle severe class imbalances.
6. Comparison of SMOTE and Augmentation Techniques in Addressing Class Imbalance
   - The Augmented SVM method appears to have a slight edge, particularly in precision and recall, implying that augmenting the dataset improves the model's ability to generalize better than SMOTE alone.
   - However, the differences are marginal, suggesting that both techniques yield comparable results.
   - The coherence and relevance of the generated data may not have been high because the prompts that we used to guide generation were not fine-tuned and the model struggled with underrepresented classes. Poor-quality synthetic samples could have failed to add significant value to the training data.

4.7. Key Insights

1. LLM Augmentation's Strengths
   - Superior at improving Recall, which is critical in scenarios where identifying minority classes is more important than Precision.
   - Produces contextually diverse synthetic samples, improving generalization.
2. SMOTE Limitations
   - Effective in less complex scenarios but fails to capture nuanced patterns in the data, leading to suboptimal Macro and Weighted Averages.
3. Original Models' Role
   - Serve as a robust baseline, especially for Precision and Accuracy, but lack the ability to address class imbal-

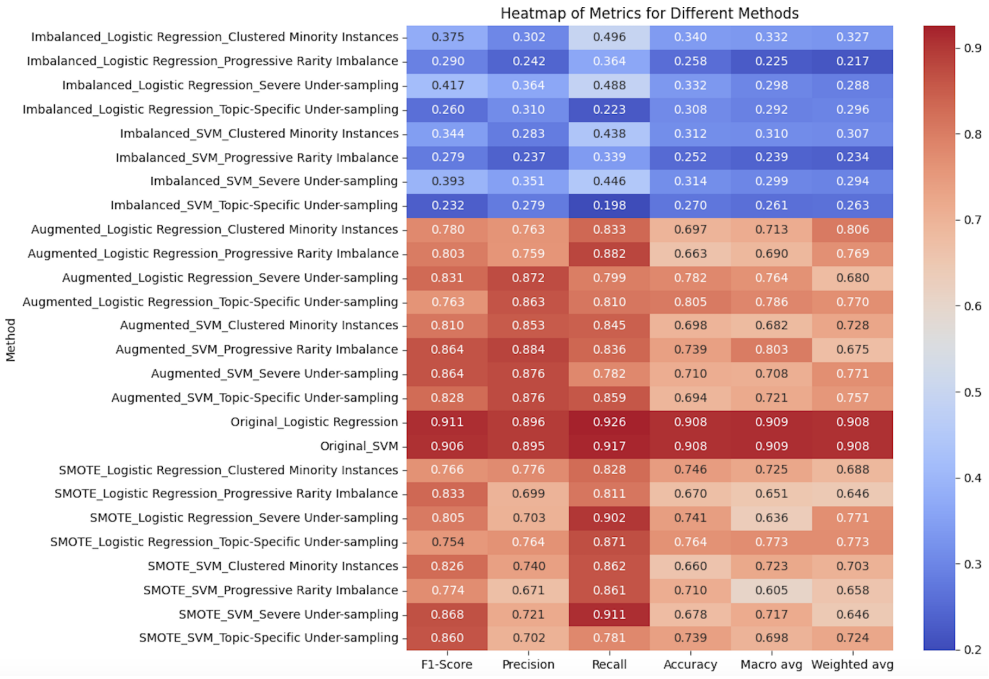

Figure 4. Heatmap of Metrics for Different Methods

ances effectively.

4.8. Errors and Challenges

1. Overfitting risks are evident in some LLM-augmented models with very high Precision but lower Recall, suggesting potential over-reliance on synthetic data.
2. Computational demands for LLM-based augmentation methods can limit scalability in resource-constrained settings.

5. Conclusion

5.1. Broader Implications

The findings of this study are highly applicable to fields like healthcare and finance, where class imbalances are common. In healthcare, LLM-based augmentation can improve rare disease detection by generating relevant synthetic cases, while in finance, it can enhance the identification of minority class transactions, such as fraud, where traditional methods often struggle. This approach offers improved accuracy and fairness in decision-making processes.

5.2. Limitations

While LLM-based augmentation demonstrated notable improvements, several limitations were identified:

- **Computational Resource Dependency:** The approach requires significant computational resources, making it less accessible for organizations with limited infra.
- **Risk of Overfitting:** Excessive or poorly diversified augmentation can lead to overfitting, particularly if synthetic data is too similar to the original samples.
- **Bias Amplification:** If the original dataset contains biases, LLMs might inadvertently propagate or amplify these biases in the augmented data.

5.3. Future Work

To address the limitations and further enhance the effectiveness of LLM-based augmentation, the following areas are suggested for future work:

- **Testing Transformer-Based Classifiers:** Investigate the performance of advanced transformer-based models (e.g., BERT, RoBERTa) trained on augmented datasets to explore improvements in classification accuracy.
- **Dynamic Prompting Strategies:** Develop adaptive prompting techniques to guide LLMs in generating more diverse and contextually relevant synthetic samples.
- **Real-World Validation:** Test the proposed methods on real-world datasets from critical domains, such as medical records and financial transactions, to validate their robustness and applicability.

The findings suggest that LLM-based augmentation offers notable strengths, such as superior recall and the generation of diverse, contextually relevant synthetic samples. However, traditional methods like SMOTE, while effective in simpler cases, struggle to capture more complex patterns, limiting their ability to address class imbalances.

References

[1] Cegin, J., Simko, J., and Brusilovsky, P. (2024). *LLMs vs Established Text Augmentation Techniques for Classification: When do the Benefits Outweigh the Costs? In Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing.* Retrieved from https://arxiv.org/abs/2408.16502 2, 3

[2] Dai, H., Liu, Z., Liao, W., Huang, X., Cao, Y., Wu, Z., Zhao, L., Xu, S., Liu, W., and Liu, N. (2023). *AugGPT: Leveraging ChatGPT for Text Data Augmentation. Preprint. arXiv:2302.13007.* Retrieved from https://arxiv.org/abs/2302.13007

[3] Wei, J., and Zou, K. (2019). *EDA: Easy Data Augmentation Techniques for Boosting Performance on Text Classification Tasks. In Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing (EMNLP-IJCNLP), pp. 6382–6388.*Retrieved from https://doi.org/10.18653/v1/D19-1670

[4] Ubani, S. O., Polat, S. O., and Nielsen, R. (2023). *ZeroShotDataAug: Generating and Augmenting Training Data with ChatGPT.* Preprint. arXiv:2304.14334. Retrieved from https://arxiv.org/abs/2304.14334