BugBERT: NLP for Bug Prioritization

Anant Maheshwari
anant200519@gmail.com
Atal Bihari Vajpayee Indian Institute of Information Technology and Management
Gwalior, Madhya Pradesh, India

Dr. Anjali
anjali@iiitm.ac.in
Atal Bihari Vajpayee Indian Institute of Information Technology and Management
Gwalior, Madhya Pradesh, India

Abstract

Software is ubiquitous and continues to grow in scale and complexity, making maintenance increasingly challenging and time-consuming. A critical aspect of managing software bugs involves prioritizing them based on severity, ensuring that high-priority issues are addressed promptly.

This work presents a pipeline for bug report prioritization using a combination of **DistilBERT**, feature engineering, and machine learning. DistilBERT embeddings are extracted to capture sentence-level context, and class imbalance is addressed with sample weights and **SMOTE-NC**. Feature engineering includes frequency and target encoding, **PCA** for dimensionality reduction, and cluster encoding for improved representation. Classification is performed using **XGBoost**, with hyperparameters optimized via **Optuna**. This approach effectively automates the task of bug report prioritization while achieving acceptable levels of accuracy

Keywords

DistilBERT, Sample weights, SMOTENC, Frequency encoding, Target encoding, PCA , Cluster encoding, XGBoost, Optuna

1 Introduction and Background

In the modern digital era, industries are increasingly automating and virtualizing their processes. However, as reliance on software grows, so does the impact of its failures. Bugs have become a significant bottleneck, with thousands of reports generated daily across large-scale software systems. There are so many bug reports coming in every day that it is simply not possible to attend to all of them promptly. This is why a crucial aspect of software maintenance is prioritizing bug reports based on their criticality to ensure that the most severe issues are addressed first. While experienced engineers are unmatched at this task, manual prioritization is often subjective, labor-intensive, and difficult to scale. With advancements in natural language processing and deep learning, ML models can help automate this process while achieving competitive accuracy and reducing the manual burden.

2 Approach

This work presents a hybrid ML-DL approach that combines transformer-based text embeddings with structured feature engineering to improve bug report prioritization. Figure 1 provides an overview of the proposed pipeline.

We fine-tune DistilBERT for sequence classification, leveraging CLS token embeddings of sequence features like Title and Description to capture the context of the entire input sequence. Additionally, categorical features—such as Status, Resolution, and Component—undergo frequency and target encoding to better capture the hidden relationships among them. Further, we use PCA to project the data onto a lower dimensional space while retaining maximum information. This enables us to extract structured information using clustering and apply SMOTE-NC to handle class imbalance. For inference, we employ XGBoost as the final classifier, further optimized using Optuna, an automated hyperparameter tuning framework, to maximize predictive performance.

3 Results and Conclusion

Class	Precision	Recall	F1-Score	Support
0	0.46	0.50	0.48	2236
1	0.36	0.34	0.35	2353
2	0.79	0.74	0.76	7789
3	0.21	0.33	0.26	572
4	0.20	0.25	0.22	299
Validation Accuracy				0.63
Validation Macro F1				0.41
Test Macro F1				0.42

Table 1: Model Performance Summary

Table 1 gives a comprehensive summary of the results obtained by the presented approach. By combining embeddings with structured data transformations, our approach effectively captures patterns within the data. The results highlight the potential of ML and DL in bug prioritization, enabling efficient software maintenance.

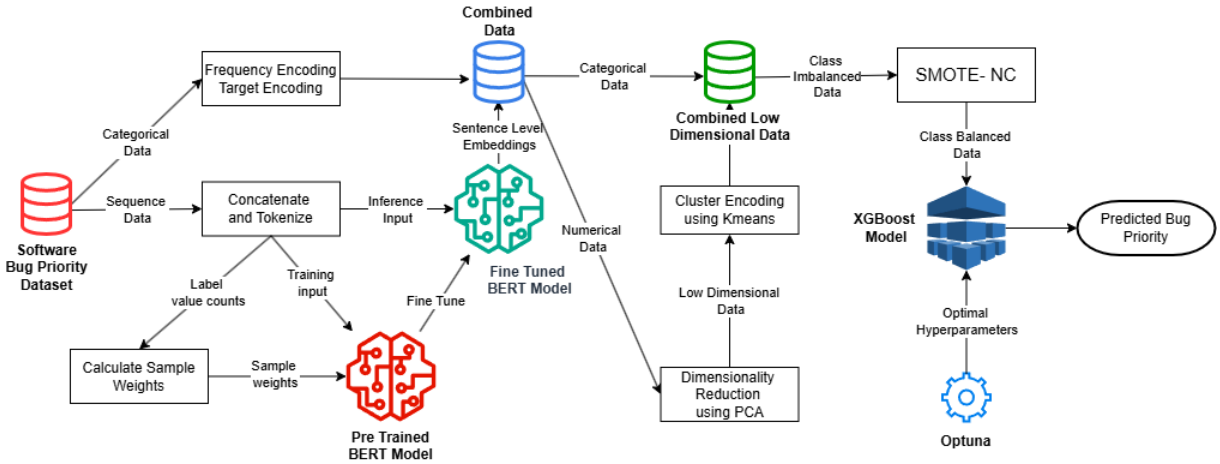

Figure 1: Bug Priority Classification Pipeline