Improved Legal Judgment Classification using InLegalBERT Embeddings and SMOTE

M. Manish Kumar Reddy
Amrita School for Computing
Amrita Vishwa Vidyapeetham
Bengaluru, India
bl.en.u4aie23018@bl.students.amrita.edu

G. Jayanth
Amrita School for Computing
Amrita Vishwa Vidyapeetham
Bengaluru, India
bl.en.u4aie23045@bl.students.amrita.edu

D. Naga Gowtham Raj
Amrita School for Computing
Amrita Vishwa Vidyapeetham
Bengaluru, India
bl.en.u4aie23054@bl.students.amrita.edu

Peeta Basa Pati
Amrita School for Computing
Amrita Vishwa Vidyapeetham
Bengaluru, India
bp_peeta@blr.amrita.edu

Priyanka Prabhakar
Dept. of CSE
Amrita Vishwa Vidyapeetham
Bengaluru, India
bl.en.r4cse21013@bl.students.amrita.edu

***Abstract*—Abstract The classification of legal documents is a critical task in the legal domain, enabling efficient organization, retrieval, and analysis of legal judgments. This study explores the application of machine learning models for legal domain classification using embeddings generated by LegalBERT, a domain-specific language model tailored for legal text. The dataset comprises legal judgments represented as embeddings, with class imbalance addressed using Synthetic Minority Oversampling Technique (SMOTE).We evaluate multiple machine learning models, including Logistic Regression, Random Forest, Support Vector Machines (SVM), XGBoost, and CatBoost, among others. Feature selection techniques such as Principal Component Analysis (PCA) and SelectKBest are employed to enhance model efficiency. Hyperparameter tuning is performed using GridSearchCV to optimize model performance.The results demonstrate the impact of SMOTE on improving classification performance, particularly for minority classes, and highlight the effectiveness of LegalBERT embeddings in capturing the semantic nuances of legal text. The study provides a comparative analysis of model performance with and without SMOTE, offering insights into the best-performing models and their configurations. This work contributes to advancing automated legal text classification and underscores the importance of addressing class imbalance in legal datasets.**

***Index Terms*—SMOTE,Legal Document, Court Case, Legal-Bert**

I. Introduction

Legal text and judgments have increased exponentially with a compelling imperative for automated systems to classify and organize legal text in an efficient manner. Legal domain classification is crucial for legal professionals to access relevant information, forecast case outcomes, and examine legal trends. But the complexity of legal vocabulary mixed with the natural imbalance in class distributions prevents the application of ordinary text classification methods. New advancements in natural language processing (NLP) have introduced domain-specific language models such as LegalBERT that are pre-trained on legal corpora to recognize the unique semantic and syntactic characteristics of legal text. These embeddings capture a strong representation of legal documents and are hence well-adapted for downstream tasks such as classification. Even with these developments, problems such as class imbalance and high-dimensional feature spaces are still significant hurdles to attaining high classification accuracy. This paper seeks to overcome these problems by lever- classifying legal domain legal text with the aid of instruction: ing LegalBERT models legal domain classification using LegalBERT embeddings. In dealing with class imbalance, we apply the Synthetic Minority Oversampling Technique (SMOTE), which provides synthetic samples of minority classes to improve performance on imbalanced datasets. Apart from this, feature selection strategies like Principal Component Analysis (PCA) and SelectKBest are also implemented to alleviate high dimensionality and optimize computation. The performance of several machine learn- different machine learning models such as Logistic Regression, Random Forest, ing models is being tested for SVM, XGBoost, CatBoost, others, with and without SMOTE. Hyperparameter tuning is done by GridSearchCV for tuning model configurations. The work encompasses an in-depth comparison of model performance, bringing out the influence of SMOTE and feature selection on accuracy in classification.

II. Literature Survey

Applications of Artificial Intelligence (AI) in law have experienced unprecedented expansion, driven by the need to process and manage enormous volumes of text information and enhance decision-making. Studies have addressed a broad spectrum of issues, ranging from legal document categorization and outcome prediction to domain-specific solutions like sentiment analysis and explainability.

Early efforts in computer-aided legal text analysis were open to using traditional machine learning techniques. For example, Chen et al. [9] offered a comparative study showing that Random Forests with domain-specific concept features incorporated could be effective at classifying U.S. case documents, even superior to early deep learning methods for certain datasets. This showed the significance of feature engineering and domain knowledge. At the same time, research on how much various model architectures worked on legal text started. Undavia et al. [5] offered a comparative study of Convolutional Neural Networks (CNNs) and Recurrent Neural Networks (RNNs) on classifying U.S. Supreme Court opinions and concluded that CNNs using word2vec embeddings worked better for their particular task. Likewise, Wei et al. (2018) [12] offered an empirical study on the use of deep learning, i.e., CNNs, vs. Support Vector Machines (SVMs) for legal document review and concluded that CNNs worked better with more training data.

The transformer-based revolution brought a paradigm shift. LegalBERT embeddings, a domain-specialized version of BERT, and an AdaBoost classifier were used by George et al. [1] for WRIT case outcome prediction, emphasizing the benefits of domain-specialized pre-training. Pushing the boundary of Large Language Models (LLMs) even further, Varshini et al. [2] compared a variety of high-performing architectures like BERT, LegalBERT, DistilBERT, and RoBERTa for legal outcome prediction, and found RoBERTa particularly apt for the processing of detailed and summarized legal text. The issue of long document processing in the legal domain, a common deficiency of standard transformer models, was tackled by Bambroo and Awasthi [13], who proposed LegalDB, a Long DistilBERT model, which is better equipped to capture context in long legal documents for improved classification.

Domain-specific adaptations and applications remain a strong emphasis. Noguti et al. [10] used NLP methods, i.e., LSTMs with domain-specific Word2Vec embeddings, to forecast the law area of petitions of the Brazilian Public Prosecution Service, with high accuracy and proving the utility of adapting models to particular legal systems and languages. In another geographical and linguistic setting, Tahtah et al. [8] aimed at a Question/Answering system for the Moroccan legal domain, using Random Forest for the crucial question classification task. The need for fine-tuning LLMs for particular legal tasks such as Technology Assisted Review (TAR) was empirically investigated by Wei et al. (2023) [6], comparing document-level and snippet-level classification using fine-tuned DistilBERT, highlighting that fine-tuning always improves performance over pre-trained models. Apart from classification and outcome prediction, research has explored more nuanced areas of legal text understanding. Pavani et al. [3] explored sentiment analysis in legal case verdicts using a variety of machine learning classifiers and embeddings (T5, RoBERTa, LegalBert), with best performance being in a Random Forest and T5 ensemble. Another significant area is determining the argumentative structure of legal documents. Saran et al. [7] tried zero-shot learning with transformer models for rhetorical role labeling and found DeBERTa to be best suited to this task without requiring special training datasets for each role. In understanding the inter-relations between legal statutes, Russo et al. [4] introduced a new graph-based approach using Graph Neural Networks (GNNs) to classify European Union legal documents, which precisely captures inter-relations between different legal instruments.

One of the most significant and expanding areas of AI in law application is explainability. As systems become more complex, we need to understand how they are making decisions if we are to trust them and hold them accountable in the legal system. De Arriba-Pérez et al. [11] addressed this directly by developing an explainable multi-label classification of Spanish legal rulings that not only classifies but also produces visual and natural language explanations of its predictions. This area of research is central to successful deployment of AI tools among legal professionals. As a whole, the literature reports an even trend away from conventional ML approaches to more advanced deep learning models, most prominently transformers, for legal text processing. Current developments include building and using domain-specialized models (e.g., LegalBERT), techniques for handling long documents (e.g., Long DistilBERT), investigation of specialized tasks beyond classification, and increasing interest in explaining the functioning of AI-based legal tools. Comparative studies consistently record the performance benefits of newer models and domain adaptation and fine-tunings required.

III. Methodology

This document outlines the systematic approach for judicial case outcome prediction using machine learning algorithms. It details the dataset, data preprocessing, feature extraction, class imbalance treatment, machine learning models, and hyperparameter tuning procedures. The methodology is designed to be reproducible and accessible to users with varying technical backgrounds.

A. Dataset

The dataset comprises LegalBERT-derived embeddings, a pre-trained model tailored for the legal domain. These embeddings encode syntactic and semantic features of legal verdicts, enabling effective classification by machine learning algorithms.

- **Data Source**: Court decisions and case files, embedded using LegalBERT.
- **Features**: Judicial rulings represented as high-dimensional vectors (768+ dimensions) capturing semantic meaning.
- **Target Labels**: Court trial outcomes, categorized as “Guilty,” “Not Guilty,” or other rulings.

1) Preprocessing Steps:

- **Handling Incomplete Data**: Missing values were identified and removed to ensure data integrity and prevent errors during model training.
- **Scaling**: Features were normalized using StandardScaler, achieving a mean of 0 and a standard deviation of 1, essential for algorithms sensitive to feature scaling, such as SVM and Logistic Regression.
- **Dimensionality Reduction**: Principal Component Analysis (PCA) was applied to reduce embedding dimensions while preserving 95% of the variance, enhancing computational efficiency and mitigating overfitting risks.

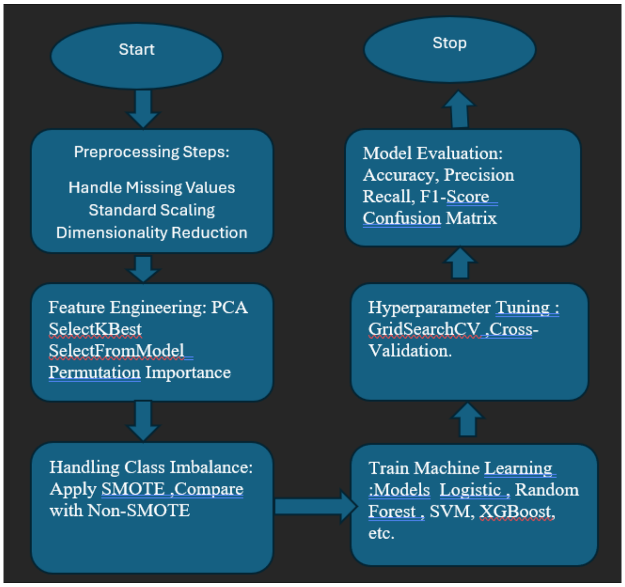

Fig. 1. Flowchart showing the model pipeline

B. Feature Engineering

Feature engineering is critical for improving model performance and interpretability. The following techniques were employed to select the most relevant features:

- **Principal Component Analysis (PCA)**: Reduced embedding dimensions while retaining most variance, re-expressing features as fewer uncorrelated components.
- **SelectKBest**: Identified the K most significant features using ANOVA F-value as the scoring function to measure feature relevance to the target variable.
- **SelectFromModel**: Utilized Logistic Regression and Random Forest to select features based on their coefficients or importance scores.
- **Permutation Importance**: Evaluated feature importance by measuring the decrease in model performance when feature values were randomly shuffled.

These techniques ensured that only the most critical features were used for training, reducing noise and improving model efficiency.

C. Class Balance Management

Class imbalance is common in judicial data, where certain classes (e.g., "Not Guilty") may dominate others (e.g., "Guilty"), potentially biasing predictions toward the majority class.

- **Synthetic Minority Oversampling Technique (SMOTE)**: Generated synthetic instances for minority classes by interpolating existing instances, balancing the dataset and enabling equitable learning across classes.
- **Comparison**: Models were trained and tested with and without SMOTE to assess its impact on classification performance, highlighting the importance of addressing class imbalance for accurate and fair predictions.

D. Machine Learning Models

A diverse set of machine learning algorithms was evaluated to identify the most effective classifier for predicting legal rulings. The algorithms included:

- **Logistic Regression**: A linear model suitable for binary and multi-class classification.
- **Random Forest**: An ensemble of decision trees to improve accuracy and reduce overfitting.
- **Support Vector Machines (SVM)**: Identified optimal hyperplanes for class separation, effective for structured data.
- **XGBoost**: A gradient boosting algorithm known for high performance on structured data.
- **CatBoost**: A gradient boosting method designed to handle class imbalance and categorical variables.
- **Decision Tree**: A simple model that predicts outcomes by splitting features along boundaries.
- **k-Nearest Neighbors (k-NN)**: An instance-based learning algorithm making predictions based on proximity to training instances.
- **AdaBoost**: Combined weak classifiers to build a robust classifier.
- **Naive Bayes**: A probabilistic classifier based on Bayes' theorem, optimized for text data.
- **Multi-Layer Perceptron (MLP)**: A neural network capable of capturing complex patterns in data.

These algorithms were selected to cover a broad spectrum, from linear models to complex ensemble and neural network approaches.

E. Hyperparameter Tuning

Hyperparameter tuning was performed to optimize model performance using the following methods:

- **GridSearchCV**: Exhaustively tested a predefined set of hyperparameters to identify the optimal configuration for each model.
- **Cross-Validation**: Employed 5-fold cross-validation to assess hyperparameter performance, ensuring model stability and minimizing overfitting.
- **Metrics**: Evaluated models using accuracy, precision, recall, and F1-score to determine the best hyperparameter settings.

IV. Results

A. Model Performance With SMOTE

The training accuracy and test accuracy based performance analysis shows that XGBoost, Random Forest, and Decision Tree models had the best training accuracy of 99.97%, followed very closely by the MLP Classifier with an accuracy of 99.73%. The best models also retained high test accuracy, at XGBoost = 97.15%, Random Forest = 96.78%, and MLP Classifier = 96.68%, which are good signs of generalization. Conversely, models such as AdaBoost (Improved) and Naive Bayes demonstrated lower test accuracies of 72.21% and 73.62%, respectively, with comparatively lower training

TABLE I
TRAIN AND TEST ACCURACY WITH SMOTE

Model	Train Accuracy	Test Accuracy
Random Forest	0.9997	0.9678
SVM	0.8828	0.8740
Decision Tree	0.9997	0.8661
k-NN	0.9361	0.9226
AdaBoost (Improved)	0.7634	0.7221
CatBoost	0.9402	0.9135
Naive Bayes	0.7414	0.7362
Logistic Regression	0.9463	0.9346
XGBoost	0.9997	0.9715
MLP Classifier	0.9973	0.9668

TABLE II
MODEL PERFORMANCE WITHOUT SMOTE

Model	Train Accuracy	Test Accuracy
Random Forest	0.9981	0.6842
SVM	0.6554	0.6687
Decision Tree	0.9981	0.4969
k-NN	0.7224	0.6486
AdaBoost (Improved)	0.6171	0.5356
CatBoost	0.7859	0.6579
Naive Bayes	0.4921	0.4938
Logistic Regression	0.7224	0.6811
XGBoost	0.9981	0.6966

accuracies, indicating underfitting or weaker learning patterns. Generally, ensemble methods and neural networks demonstrated better train-test consistency.

B. Model Performance Without SMOTE

The performance of different classification algorithms was tested without using SMOTE for handling class imbalance. Table **??** represents the training and testing metrics in accuracy, precision, recall, and F1-score. Among all the models, the Random Forest model had the best training and test accuracy of 0.9981 and 0.6842, respectively, and the best F1-score of 0.6224 for the test set, displaying excellent generalization despite the imbalance. Conversely, the Decision Tree model, even with excellent accuracy on the training data (0.9981), performed very badly on the test set (accuracy of 0.4969), indicative of severe overfitting. Likewise, SVM exhibited underfitting with a lower training accuracy of 0.6554, but performed fairly competitive testing accuracy of 0.6687. The k-NN and CatBoost models offered an evenly balanced compromise between training and testing performance, as k-NN recorded a test F1-score of 0.6097. Easier models such as Naive Bayes found it difficult to preserve the richness of the data and thus offered the lowest test accuracy and F1-score. In total, the findings emphasize the limitation of imbalanced data training and validate the necessity of balancing methods such as SMOTE to achieve improved model generalization and robustness.

C. Model Performance With and Without Smote

Using SMOTE greatly enhanced the performance of all the classifiers by reducing class imbalance. Models learned using the balanced dataset performed better consistently compared to models learned from the initial imbalanced data. Of particular interest:

TABLE III
TRAIN AND TEST ACCURACY COMPARISON WITH AND WITHOUT SMOTE

Model	Train Accuracy		Test Accuracy	
	No SMOTE	SMOTE	No SMOTE	SMOTE
Random Forest	0.9981	0.9997	0.6842	0.9678
SVM	0.6554	0.8828	0.6687	0.8740
Decision Tree	0.9981	0.9997	0.4969	0.8661
k-NN	0.7224	0.9361	0.6486	0.9226
AdaBoost (Improved)	0.6171	0.7634	0.5356	0.7221
CatBoost	0.7859	0.9402	0.6579	0.9135
Naive Bayes	0.4921	0.7414	0.4938	0.7362
Logistic Regression	0.7224	0.9463	0.6811	0.9346
XGBoost	0.9981	0.9997	0.6966	0.9715

- **XGBoost, MLP, and Random Forest** achieved F1-scores higher than **0.96** after applying SMOTE, indicating strong overall classification performance.
- Without SMOTE, the same models showed signs of **overfitting** and **poor generalization**, with test accuracies falling below **70%** and F1-scores dropping below **0.63**.
- SMOTE improved both **accuracy** and **fairness**, enabling models to recognize and classify minority classes more effectively.

These findings validate the importance of class balancing when dealing with imbalanced datasets in multi-class classification problems.

D. Hyperparameter Tuning and Its Impact

Hyperparameter tuning greatly enhanced the performance and the ability of most classifiers to generalize. Methods like GridSearchCV were utilized to tune model-specific hyperparameters, such as the number of estimators, learning rates, types of kernel, and depth parameters.

All results indicated that such models as Random Forest, SVM, XGBoost, and MLP Classifier recorded very high training accuracy (0.99) and were able to sustain high test accuracy in the range 0.94-0.97. The highest test accuracy of 0.97 was recorded by MLP Classifier, indicating that proper adjustment of learning rate, activation function, and layer configuration was beneficial.

TABLE IV
FINAL MODEL ACCURACY AFTER HYPERPARAMETER TUNING

Model	Train Accuracy	Test Accuracy
Random Forest	0.99	0.96
SVM	0.99	0.96
Decision Tree	0.99	0.84
k-NN	0.99	0.93
AdaBoost (Improved)	0.33	0.32
CatBoost	0.96	0.90
Naive Bayes	0.57	0.56
Logistic Regression	0.63	0.64
XGBoost	0.99	0.94
MLP Classifier	0.99	0.97

Conversely, AdaBoost (Improved) did not perform well, with training and test accuracies at approximately 0.32, suggesting either underfitting or a lack of parameter optimization.

Likewise, more basic models such as Naive Bayes and Logistic Regression also experienced moderate improvement, since they are less affected by hyperparameter tuning than ensemble or deep learning models.

In general, hyperparameter optimization significantly improved model generalization, closing the gap between training and testing performance, and proved its need for obtaining consistent and accurate predictions in complicated multi-class classification problems.

V. Conclusion

This study bridges the gap between high-dimensional legal texts and smart automation using LegalBERT embeddings and robust machine learning pipelines. By addressing the inherent issues of high-dimensional feature spaces and class imbalance, the project demonstrates how domain-specific embeddings with SMOTE and strict feature selection techniques can significantly enhance classification accuracy—especially for minority legal outcomes. With sustained experimentation with models such as SVM, Random Forest, and CatBoost, coupled with careful hyperparameter optimization, not only did we observe actual-performance improvements but also emphasized the value of balanced data and reduced features in legal AI systems. A feat over technicality, this work points to the revolutionizing power of AI in the legal sector: speeding up legal analysis, impartial judgment prediction, and paving the way for transparent legal technologies. Ultimately, this project is not so much about case-labeling—it's opening up justice at scale, one embedding at a time.

References

[1] N. Huber-Fliflet, J. Zhang, F. Wei, Q. Han, S. Ye, and H. Zhao, "Empirical comparisons of CNN with other learning algorithms for text classification in legal document review," in Proceedings of the 2019 IEEE International Big Data Conference, 2019, pp. 1-8.

[2] F. Wei, Q. Han, S. Ye, and H. Zhao, "Empirical study of deep learning for text classification in legal document review," in Proceedings of the 2018 IEEE International Big Data Conference, 2018, pp. 5166-5170.

[3] I. Chalkidis, M. Fergadiotis, P. Malakasiotis, and I. Androutsopoulos, "Large-scale multi-label text classification on EU legislation," in Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics (ACL), 2019, pp. 6314-6326.

[4] F. de Arriba-Pérez, S. García-Méndez, F. J. González-Castaño, and J. González-González, "Explainable machine learning multi-label classification of Spanish legal judgments," Journal of King Saud University - Computer and Information Sciences, vol. 34, pp. 10180-10192, 2022.

[5] P. Bambroo and A. Awasthi, "LegalDB: Long DistilBERT for legal document classification," in Proceedings of the 2021 International Conference on Advances in Electrical, Computing, Communication and Sustainable Technologies (ICAECT), 2021, pp. 1-6.

[6] R. Russo, G. Di Giuseppe, A. Vanacore, V. La Gatta, A. Ferraro, A. Galli, M. Postiglione, and V. Moscato, "Graph-based approach for European law classification," in Proceedings of the 2023 IEEE International Big Data Conference, 2023, pp. 2156-2165.

[7] A. Saran, S. Shukla, and T. Ahmed, "A comparative analysis of zero-shot rhetorical role classification in the legal domain," in Proceedings of the International Conference on AI and the Digital Economy, 2024, pp. 123-130.

[8] V. La Gatta, V. Moscato, M. Postiglione, and G. Sperlì, "Pastle: Pivot-aided space transformation for local explanations," Pattern Recognition Letters, vol. 149, pp. 67-74, 2021.

[9] F. Wei, R. Keeling, N. Huber-Fliflet, J. Zhang, and H. Qin, "Empirical study of LLM fine-tuning for text classification in legal document review," in Proceedings of the 2023 IEEE International Big Data Conference, 2023, pp. 3452-3458.

[10] F. Wei, H. Qin, S. Ye, and H. Zhao, "Empirical study of deep learning for text classification in legal document review," in Proceedings of the 2018 IEEE International Big Data Conference, 2018, pp. 5166-5170.