Published by :
http://www.ijert.org

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Volume 13, Issue 02, February 2024

Using Machine Learning Algorithms with TF-IDF to Generate Legal Petitions

Poulami Basu, Dr. Dinabandhu Bhandari, Koustav Sengupta
Heritage Institute of Technology, Kolkata, India

***Abstract*—Legal documentation plays a crucial role in supporting lawyers in preparing petitions, yet the process of drafting such documents is often tedious and repetitive, typically handled by stenographers. While artificial intelligence (AI) has been employed to offer legal advice, the application of machine learning to predict the specific type of petition required by a lawyer remains relatively underexplored. In this research, we address this gap by proposing an innovative approach to automate the categorization of legal cases. Our solution leverages the Natural Language Toolkit (NLTK) and machine learning algorithms with Term Frequency-Inverse Document Frequency (TF-IDF) vectorization. This model ensures both reliability and precision in categorizing legal cases based on the specific requirements of a petition. To build a comprehensive and accurate database, we employ keywords and harness the capabilities of ChatGPT to generate commonly used sentences tailored to distinct petition categories. The proposed system aims to streamline and enhance the efficiency of the legal documentation process, providing lawyers with a tool that not only automates the categorization of cases but also learns and improves over time.**

***Index Terms*—legal documentation, machine learning, NLTK, TF-IDF, LLM, AI, ChatGPT**

I. INTRODUCTION

Legal documentation is an important part of the legal system which includes drafting petitions for a legal case. Legallanguage is often very complex and difficult to understand for normal people. It requires deep understanding of legal contexts to be interpreted correctly.

Using Artificial Intelligence(AI) in legal documentation speeds up the entire process, and increases efficiency as repetitive tasks like drafting the petition based on a format is easily done using AI. It can also analyse vast legal data, analyzing legal precedents and can provide valuable insights. It also reduces cost of employing stenographers for drafting the documents.

We have designed a website with the primary objectiveof streamlining the legal documentation process through the implementation of a machine learning TF-IDF model, which proficiently classifies the appropriate type of legal petition based on user-input data. The integration of this model within the website's architecture serves as an innovative approach to automating and enhancing the legal documentation experience, effectively discerning the specific category of legal petition suited to the user's input, thereby contributing to the overall efficiency and accuracy of the platform. [6]

II. THEORY

A. Naive Bayes Classifier

Naive Bayes Classifier is an improved version of the Bayes model with an assumption that the attributes are independent. It is widely used in text classification in machine learning by using conditional probability on the different attributes. It is simple and has high computational accuracy. [2] [1]

Let X be a data tuple or evidence. It has a set of n attributes. Let H be a hypothesis that X belongs to a specific class

C. $P\ (H|\ X)$ is the posterior probability or the conditional probability of H on X. Similarly $P\ (X\ |H)$ is the conditional probability of X on H.

$$P(H|X) = \frac{P(X|H) \cdot P(H)}{P(X)}$$

To find the correct class we need to find the maximum probability of the classes.

$$P(C_i|X) = \max\{P(C_1|X), P(C_2|X), \ldots, P(C_n|X)\}$$

B. Support Vector Machine

Support Vector Machines (SVMs) are a class of supervised learning algorithms designed for both classification and regression tasks. It can be used for both linear and non-linear data. SVM operates by finding the optimal hyperplane that separates different classes in the feature space. [3]

Let the dataset D be given as $(X_1, y_1), (X_2, y_2), \ldots, (X_j, y_j)$, where X_i is the set of training tuples with associated class labels y_i and $X_i \in \mathbb{R}^d$. Each y_i can take one of two values, either C_1 or -1 or +1.

The optimal hyperplane is defined as:

$$w^T x + b = 0$$

Here, w is the weight vector, x is the input feature vector, and b is the bias term.

The weight vector w and bias term b satisfy the following inequalities for all elements of the training set:

$$w^T x + b \geq +1 \text{ if } y_i = 1$$

$$w^T x + b \leq -1 \text{ if } y_i = -1$$

The primary goal of training an SVM model is to determine the values of w and b such that the hyperplane effectively separates the data while maximizing the margin, denoted by $\frac{1}{\|w\|_2}$.

Vectors x_i for which $y_i\,(w^T x_i + b) = 1$ will be referred to as support vectors.

C. Random Forest

Random forest is based on tree-based models. A tree-based model functions by iteratively segregating the given dataset into two distinct groups, guided by a particular criterion, until a predefined stopping condition is reached. These decision trees culminate in terminal points known as leaf nodes or leaves. These leaves represent the ultimate outcomes or predictions based on the features and conditions encountered throughout the recursive partitioning process. It is used for both classification and regression problems. [7]

D. Multi Layer Perceptron

Multi Layer Perceptron(MLP) is the most frequently used neural network. It's power lies in non-linear activation functions. A type of feedforward neural network, MLPs consist of multiple layers of interconnected nodes, each layer contributing to the transformation and extraction of intricate patterns from input data. With an input layer, one or more hidden layers, and an output layer, MLPs excel at learning complex, nonlinear relationships within datasets. The nodes within each layer, activated by intricate mathematical functions, allow the network to capture and comprehend intricate hierarchical features, enabling robust performance in tasks ranging from image recognition to natural language processing. [5]

E. TF-IDF Vectorizer

TF-IDF is used in information retrieval in natural language processing. It is used to find the importance of a word in a collection of documents.

Term Frequency(TF) is how frequently a word occurs in a document. The higher the value, the more important is that word.

$$\frac{\text{Number of times term } t \text{ appears in document } d}{\text{Total number of terms in document } d}$$

Inverse Document Frequency(IDF) measures the importance of a term across the entire corpus. It is calculated by taking the logarithm of the ratio of the total number of documents in the corpus to the number of documents containing the term. The goal is to assign higher weights to terms that are less common across the entire corpus, making them more discriminative.

$$\mathrm{IDF}(t, D) = \log \frac{\text{Number of documents containing term } t+1}{\text{Total number of documents in the corpus}}$$

The final TF-IDF score for a term t in a document d is obtained by multiplying the term frequency (TF) and the inverse document frequency (IDF):

$$\text{TF-IDF}(t, d, D) = \text{TF}(t, d) \times \text{IDF}(t, D)$$

Words with higher TF-IDF scores are considered more significant in representing the content of a document. [4]

III. METHODOLOGY

A. Dataset

In our experimental design, we formulated sentences articulating the reason behind each petition through the utilization of ChatGPT. Subsequently, we employed a csv file to represent these sentences, upon which our machine learning based TF-IDF vectorizer was applied to ascertain the respective petition types. This approach leverages both generative language modeling capabilities and a structured vectorization technique to enhance the accuracy and efficiency of petition classification within our research framework.

B. Procedure

The initial phase of our experimentation involves the loading of the dataset, a crucial step to access and prepare the corpus of data for subsequent analysis. Following dataset loading, we proceed to extract both features and labels from the dataset.

Features represent the independent variables, while labels denote the corresponding categories or classes associated with each data instance.

To facilitate effective machine learning, we undertake the conversion of textual data into TF-IDF (Term Frequency-Inverse Document Frequency) features. This transformation enables a numerical representation of the textual content, capturing the importance of words within the dataset. The next step involves the training of different machine learning models. This models is employed to discern patterns within the TF-IDF features and effectively categorize data instances into predefined classes or categories.

For real-world application, we incorporate a user input processing stage. The input is preprocessed to ensure compatibility with the trained model, and tokenization is applied to break down the input into constituent elements for analysis. Leveraging the trained model, we predict the category of the user input based on the extracted features and the learned patterns from the dataset.

To enhance the classifier’s adaptability and accuracy, we iteratively retrain the model by incorporating both the user input and the predicted category into the dataset. This iterative learning process ensures that the model continuously evolves and improves its performance over time.

In order to assess the performance and reliability of our developed model, a systematic evaluation procedure was employed. The dataset was strategically partitioned into two subsets, with 80% of the data designated for training purposes and the remaining 20% reserved for testing. This division ensures a robust evaluation by allowing the model to learn patterns and features from the training data, subsequently validating its efficacy on unseen instances within the test set.

The training phase involved the application of our model to the designated training dataset, enabling the algorithm to learn and adapt to the underlying patterns inherent in the legal document features. Subsequently, the model underwent testing on the reserved dataset, and performance metrics were meticulously computed to gauge its accuracy and effectiveness in real-world scenarios.

C. Performance Evaluation

Table I highlights the accuracies obtained for the different machine learning models, namely Naive Bayes Classifier, SVM, Random Forest and MLP with TF-IDF Vectorizer.

Fig 1. represents a comparison of the accuracies of thedifferent classifiers used.

Fig 2. represents a comparison of the confusion matrices of the different classifiers used.

Classifier	**Accuracy**
Naive Bayes	0.88
SVM	0.98
Random Forest	0.90
MLP	0.97

TABLE I
CLASSIFIER ACCURACIES

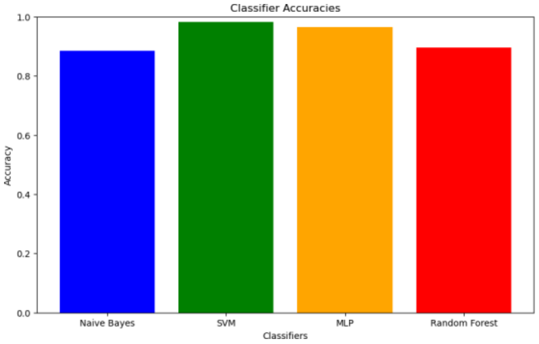

Fig. 1. Comparison of classifier accuracies

IV. CONCLUSION

In this research, we explored the application of machine learning models, specifically Naive Bayes, SVM, Random Forest, and MLP, for the classification of legal documents using TF-IDF features. The primary objective was to enhance the efficiency and accuracy of legal document categoriza- tion, providing a valuable tool for legal professionals andresearchers.

Our experiments yielded promising results, with SVM

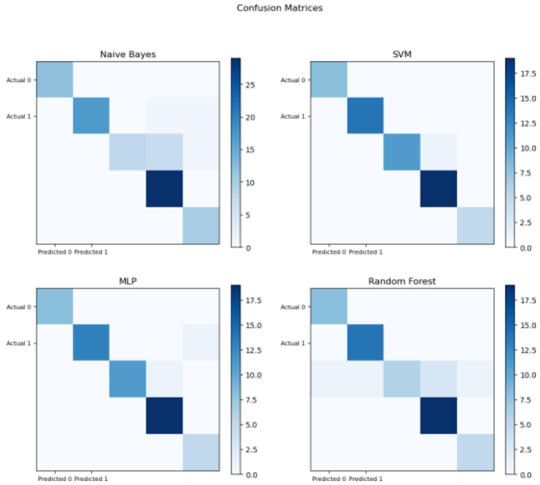

Fig. 2. Confusion matrices

achieving the highest accuracy of 98%, followed closely by MLP with an accuracy of 97%. Naive Bayes and Random Forest also demonstrated respectable accuracies of 88% and 90%, respectively. These findings underscore the effectiveness of employing TF-IDF features in conjunction with machine learning algorithms for the classification of legal documents.

The TF-IDF vectorization technique proved crucial in capturing the unique semantic signatures of legal texts, allowing our models to discern subtle nuances and patterns. The classifiers exhibited robust performance across various legal document types, showcasing the adaptability and generalizability of our approach.

This research contributes to the growing body of literature on automated legal document classification, emphasizing the significance of TF-IDF features and machine learning models in this domain. The developed models offer a valuable resource for legal practitioners, aiding in the swift categorization of diverse legal documents, thereby streamlining legal research and document management processes.

Future research endeavors could focus on expanding the dataset to encompass a more extensive range of legal doc- ument types and addressing potential biases that may arise from imbalanced class distributions. Additionally, exploring advanced natural language processing (NLP) techniques and deep learning architectures could further enhance the perfor- mance of legal document classification models.

REFERENCES

[1] M. BAYGIN. Classification of text documents based on naive bayes using n-gram features. In 2018 International Conference on Artificial Intelligence and Data Processing (IDAP), pages 1–5, 2018.

[2] B. Das and S. Chakraborty. An improved text sentiment classification model using tf-idf and next word negation. arXiv preprint arXiv:1806.06407, 2018.

[3] S. Huang, N. Cai, P. P. Pacheco, S. Narrandes, Y. Wang, and W. Xu. Applications of support vector machine (svm) learning in cancer genomics. Cancer genomics & proteomics, 15(1):41–51, 2018.

[4] S.-W. Kim and J.-M. Gil. Research paper classification systems based on tf-idf and lda schemes. Human-centric Computing and Information Sciences, 9:1–21, 2019.

[5] M.-C. Popescu, V. E. Balas, L. Perescu-Popescu, and N. Mastorakis. Multilayer perceptron and neural networks. WSEAS Transactions on Circuits and Systems, 8(7):579–588, 2009.

[6] R. Rodrigues. Legal and human rights issues of ai: Gaps, challenges and vulnerabilities. Journal of Responsible Technology, 4:100005, 2020.

[7] M. Schonlau and R. Y. Zou. The random forest algorithm for statistical learning. The Stata Journal, 20(1):3–29, 2020.