SMART DOCUMENT CLASSIFICATION

Tanmay Magadum[1], Priyanka Kumbhar[2], Shruti Zanzane[3], Dr. Prasanna Shete[4]

[1]Department of E&TC, PVG's COET, Pune
[2]Department of E&TC, PVG's COET, Pune
[3]Department of E&TC, PVG's COET, Pune
[4]Department of E&TC, PVG's COET, Pune

---***---

Abstract - Smart document classification using Naive Bayes is a method used for organizing large volumes of unstructured data into categories or classes. This method uses the Naive Bayes algorithm's probabilistic approach to determine the likelihood that a text falls into a specific category based on the frequency of its words. A machine learning approach called Naive Bayes is used to forecast the likelihood of a particular category depending on the characteristics of the document. For text classification applications including spam filtering, sentiment analysis, and topic modeling, this method has been frequently employed. In this method, the algorithm is trained on a labeled dataset, which contains a set of documents and their corresponding categories The algorithm can be used to categorize fresh documents into the appropriate categories after it has been trained. The standard and applicability of the features used to represent the documents will determine how accurate the classification is. The Naive Bayes classifier is a popular option for smart document categorization since it is simple to use and needs little training data to attain high accuracy. Overall, automatic document classification using naive Bayes is an effective and efficient approach for organizing and managing large volumes of textual data.

Key Words: classification, Naïve Bayes, machine learning, textual data

1. INTRODUCTION

Smart document classification is the process of assigning pre-defined categories to documents based on their content. A well-liked machine-learning approach called Naive Bayes can automatically classify documents.

The Naive Bayes method is based on the Bayes' theorem, a concept in probability theory that enables us to estimate an event's likelihood given our current knowledge. The Naive Bayes algorithm determines the likelihood that a document belongs to a specific category in the context of automatic document classification based on the likelihood that the words in the document belong to that category.

The Naive Bayes algorithm's "naive" presumption is that the existence of one word in a document does not depend on the existence of any other words. Although not fully accurate, this assumption makes the calculation simpler and enables the technique to scale effectively even when dealing with massive datasets.

To use the Naive Bayes algorithm for smart document classification, we first need to train the algorithm using a labeled dataset. This dataset should contain documents that are already classified into pre-defined categories. This training dataset is used by the Naive Bayes algorithm to determine the likelihood of each word falling into each category for a brand-new document based on the frequency of the terms in the document. The document is then placed in the category with the highest likelihood.

To classify a new, unlabeled document, the algorithm calculates the probability of the document belonging to each category based on the likelihood of each word or feature in the document given each category and the prior probability of each category. The algorithm then assigns the document to the category with the highest probability.

In summary, smart document classification using Naive Bayes involves the following steps:

- Collect a labeled dataset of documents.
- By calculating the probability of each word falling into each category, you may train the Naive Bayes algorithm on the dataset.
- Use the trained algorithm to classify new documents by calculating the probability of each category based on the frequency of the words in the document.
- Assign the category with the highest probability to the document.

2. LITERATURE REVIEW

Researchers in the fields of machine learning and natural language processing have been putting a lot of effort into the field of smart document classification. Recent years have seen a large number of studies and research publications exploring the many methods, algorithms, and uses of smart document classification. Here is a quick summary of some of the most important studies:

"A Survey of Document Classification Techniques" (2017): An overview of various document classification methods, including rule-based, statistical, and machine learning-based methods, is given in this paper. The report lists the most widely used algorithms in smart document categorization and emphasizes the benefits and drawbacks of each technique.

"Automated Document Classification Using Machine Learning" (2018): This work investigates automated document classification using machine learning algorithms. The most efficient method for classifying documents is determined by comparing the performance of various algorithms, including Naive Bayes, Decision Trees, and Support Vector Machines.

"Smart Document Classification for Financial Institutions" (2020): In this work, the use of intelligent document classification in financial organizations is investigated. The paper emphasizes how employing smart document classification may automate document processing, lower operating costs, and boost compliance.

In a study by Chang and Chen (2018), Naive Bayes was used to classify a large dataset of news articles. The study found that Naive Bayes achieved high accuracy in categorizing the news articles and outperformed other machine learning algorithms such as Support Vector Machines (SVMs) and Decision Trees.

Similarly, in a study by Liu et al. (2019), Naive Bayes was used for smart document classification in the field of digital libraries. The study found that Naive Bayes achieved high accuracy in categorizing research articles and outperformed other machine learning algorithms such as k-Nearest Neighbors (KNN) and SVMs.

These studies show that smart document classification has enormous promise for automating document processing, boosting productivity, and improving decision-making across a range of businesses and organizations. The findings also emphasize the need for additional study to create more sophisticated methods and algorithms to boost the precision and efficiency of smart document classification.

3. ALGORITHM EXPLANATION

The algorithm that has been used in the project is Naïve Bayes Algorithm.

A probabilistic approach called Naive Bayes is utilized in machine learning for the categorization of tasks. It is based on the Bayes theorem, which states that the likelihood of a hypothesis—in this case, a class label—given some observed evidence is proportional to the likelihood of the evidence—in this case, the characteristics of an instance—given the hypothesis, multiplied by the prior probability of the hypothesis.

The "naive" in the name refers to the assumption that the features are conditionally independent given the class label. This means that the presence or absence of one feature does not affect the probability of another feature being present or absent, given the class label. This supposition makes the method simpler and the probabilities computation easier to calculate.

The Naive Bayes algorithm works by first training a model on a labeled dataset, where the class labels are known for each instance and the corresponding features. The model learns the prior probability of each class label and the conditional probability of each feature given each class label. This is done by calculating the frequency of each class label and the frequency of each feature given each class label.

Once the model is trained, it can be used to classify new instances. Given a set of features for an instance, the algorithm calculates the posterior probability of each class label using Bayes' theorem and chooses the class label with the highest probability as the predicted class label for that instance.

Naive Bayes is a simple and fast algorithm that can work well for classification tasks, especially when the number of features is large compared to the size of the training set. However, the assumption of feature independence may not hold in some cases, which can lead to suboptimal performance.

Multinomial Naive Bayes is a variant of the Naive Bayes algorithm that is specifically designed for text classification tasks. It assumes that the features are generated from a multinomial distribution, often with text data.

The key difference between Multinomial Naive Bayes and the standard Naive Bayes algorithm is that it uses a multinomial distribution to model the likelihood of a feature given a class, rather than a Gaussian distribution.

In Multinomial Naive Bayes, we represent each document as a bag of words, which is a vector of word counts. The algorithm then calculates the probability of a document belonging to a particular class based on the frequency of words in the document.

Here are the steps involved in building a Multinomial Naive Bayes classifier:

1. Data preparation: The first step is to prepare the text data for training the classifier. This involves tokenizing the text, removing stop words, and converting the text into a numerical format, such as a bag of words.

2. Calculate class priors: The next step is to calculate the prior probability of each class. This is the probability of a document belonging to a particular class, regardless of the words in the document.

3. Calculate likelihood probabilities: For each word in the vocabulary, we calculate the probability of the word given the class. This is known as the likelihood probability.

4. Calculate posterior probabilities: Using the Bayes' theorem, we calculate the posterior probability of each class given the words in the document.

5. Prediction: Finally, we predict the class of a new document by selecting the class with the highest posterior probability.

Multinomial Naive Bayes is a widely used algorithm in text classification tasks such as sentiment analysis and topic classification. It is fast and requires a small amount of training data, making it ideal for large datasets. However, it assumes that the features are independent of each other, which may not always hold true in real-world scenarios.

4. EXPERIMENTAL DESIGN

Experimental design details about the steps carried out for performing the experiment or project. It provides information about the project's blueprint or plan in order to achieve the desired results. It provides an explanation of the experiment's specifics, outlines each step of the experiment, and lists all the supporting tasks that were completed during the project's execution. The project's aim must be clearly defined, and planning is essential to achieving that goal within the allotted time frame. Additionally, the experimental design provides a general roadmap for when the project can be completed entirely if it is planned effectively and economically.

It is essential to set up an analysis properly in order to ensure accurate results.

A. Removal of header/footer

Since all the documents of the dataset contain headers/footers such as From, Subject, etc. Elimination of these from the actual content of the document can lead to better classification.

B. Removal of stop words

Almost all the documents across all categories contain words like 'The,' 'A,' 'From,' 'To,' etc. These words may interfere with the classification process and distort the findings. The removal of these words may give more accurate classification results.

C. Stemming of words

Some of the words originate from another word. In such cases, it would be better if we consider all forms of root words as the root word itself. Let us say we have the following words with frequency in a given document; walk: 3, walked: 4, walking: 6. Thus instead of considering all the three forms separately, we can consider the root word walk with the frequency of 13 since all the three words signify the same meaning in a different tense. Stemming of words before feature representation can lead to better classification accuracy.

D. Feature representation using TF/IDF

This is one of the most important experiments of the project work and the core of the hypothesis. For feature representation of documents, the count vectorizer takes into account the frequency of words occurring in a document regardless of the distinguishability of words in a text. Instead, the documents could be better represented if we consider the term frequency along with how much distinguishing the term is. Such representation should also help improve the overall accuracy of the model.

E. Naïve Bayes Classifier Training

After getting the feature representation of all documents, the classifier is trained on the training set of documents (feature vectors of training documents).

F. Testing

Following the construction of the Classification model, the classifier is evaluated using the 20% test set of documents.

5. HIGH LEVEL DESIGN

We have a detailed architecture that describes and outlines the general structure and functionality of the project modules and all the stages it comprises here in the high-level design with regard to the project. The project's high-level design explains how the project was created with the user in mind. To determine where everything can be completed at once, view the project from a higher end. Figure 1 provides information on the project where text documents are used as the source of data and data pre-processing is used to convert them into the necessary format for the project. Features are extracted from the data after analysis that aid in classification. To create an ML model (Machine Learning model), the desired algorithm is selected and used. The built model is saved and used for future purposes for testing new data as well.

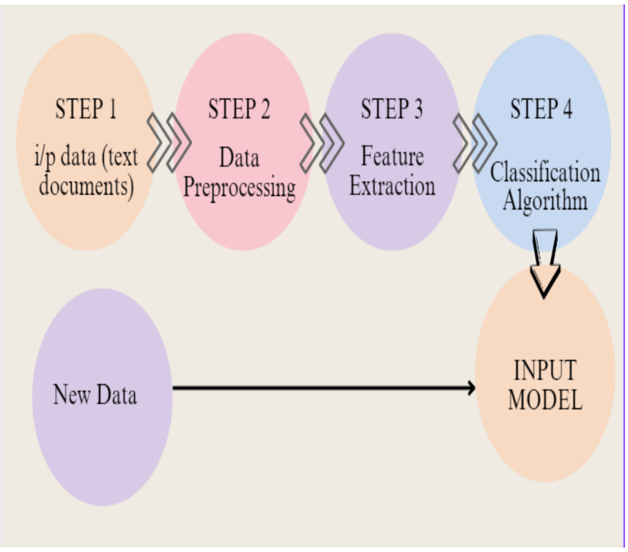

Figure 1. High-Level Design of Document Classification using Machine Learning

6. RESULT AND DISCUSSION

The documents have been classified by the system as per the predefined categories namely, sports, business, entertainment, politics, and tech.

A group of categorized documents that have been automatically categorized based on their content is the output of a smart document classifier. Each document can be put into a pre-established category by the classifier, or new categories can be made as needed.

To gain insights, guide decision-making, and enhance business procedures, more processing and analysis can be done on the smart document classifier's output. The following are some of the main advantages of smart document classification:

- Efficiency is increased since users can discover the papers they need more quickly and simply thanks to automatically categorized documents, saving time and energy compared to manual sorting and searching.
- Enhanced accuracy: AI systems can accurately analyze huge numbers of documents, lowering the chance of human error.
- Better decision-making: With documents categorized and organized, decision-makers can quickly access the information they need to make informed decisions.
- Increased compliance: For organizations that must comply with regulations or legal requirements, smart document classification can ensure that documents are

appropriately classified and stored, reducing the risk of non-compliance.

- Better organization: By classifying documents intelligently, a repository of documents can be built that is more structured and organized, which makes it simpler to manage and retrieve data.
- Better decision-making: Decision-makers can easily obtain the data they need to make educated judgments thanks to papers that have been categorized and organized.

7. CONCLUSION AND FUTURE ENHANCEMENT

The project Smart Document Classification Using Machine Learning is an innovative project that can be viewed as a demo project to comprehend the fundamentals of Data collection, Data analysis, Data preparation, and Data pre-processing by applying various types of cleaning techniques. It also gives researchers who are seriously interested in the field of data science an idea on the Machine learning models, feature extraction process, and Machine learning algorithm. Researchers can use this project as an introduction or example project to see how features are classified and how the categorization of documents works, as well as the fundamental stages that must be taken into account. Additionally, this project assists students in learning the fundamentals of machine learning and data-cleaning methods. It serves as a major documentation project or a demo project. The fundamentals of intelligent document classification and how machine learning contributes to smarter document categorization are simple to comprehend.

Since the main goal of our job work was to separate the many types of documents available and give the desired category to which the document belongs, only one algorithmic technique—Multinomial Naïve Bayes Algorithm—has been applied in this instance. One may also want to move the project forward in order to test various component portrayal strategies and try the model with additional AI calculations like SVM, Neural Network, Expedition amplification, Decision trees, and so forth.

Smart document classification is a rapidly developing technology that is always getting better because of developments in machine learning and natural language processing, so the field's future potential is quite bright. Future advancements and applications of smart document classification are anticipated to occur mostly in the following major areas:

1. Integration with other technologies: To improve its accuracy and capabilities, smart document categorization can be combined with other technologies including optical character recognition (OCR), natural language processing (NLP), and machine learning techniques.

2. Support for numerous languages: Due to the growing globalization of business, smart document categorization is likely to support a variety of languages in order to meet the demands of organizations with global operations.

3. More sophisticated classification: The classification of documents can be improved to include more sophisticated elements like sentiment analysis, entity recognition, and topic modeling.

4. Greater accuracy: With the development of more sophisticated machine learning algorithms and the accessibility of larger and more varied datasets, it is anticipated that the accuracy of smart document classification will continue to rise.

5. Improved user interface: Users will likely find it simpler to engage with and manage the categorized documents as a result of an improved user experience for smart document classification.

6. Use in new industries: The application of smart document classification to new fields like education, media, and research creates new prospects for automation and efficiency.

REFERENCES

1. Ankit Basarkar “Document classification using Machine Learning [1]”, Springer International Conference 5-25-2017 Vol 531 © 2018.
2. Sindhu Rashmi. H. R, Prof. Anisha. B. S, Dr. Ramakanth Kumar. P “Smart Document Analysis Using AI-ML”, (IJIRCST) ISSN: 2347-5552, Volume-7, Issue-3, May-2019 DOI: 10.21276/ijircst.2019.7.3.6
3. Suresh Yaram “Machine Learning Algorithms for Document clustering and Fraud Detection” 2016 IEEE International Conference on Data Science and Engineering (ICDSE) 978-1-5090-1281-7/16/$31.00 ©2016 IEEE
4. Fen-Jeng Yan, “An Implementation of Naive Bayes Classifier”, 2018 International Conference on Computational Science and Computational Intelligence (CSCI)