International Research Journal of Engineering and Technology (IRJET) e-ISSN: 2395-0056
Volume: 09 Issue: 06 | June 2022 www.irjet.net p-ISSN: 2395-0072

Text Document Classification System

Sarosh Dandoti[1]

Abstract - *Document classification needed in day to day activities while arranging loads of text documents containing various kinds of articles on different topics. This text Document Classification is essentially the process of assigning each text document a category. Text Classification focuses on a wide range of applications from detecting emotion from a sentence to finding the general context of a summary of an article. In this paper, however, we have focused on the Classification of different newspaper articles to arrange them into different sections. The goal of this research is to design a multi-label classification model with parameter tuning to improve performance and predictions. Text and Document Classification has become an important part of today's social internet media. Tweets, messages, and posts must be monitored to find out the existence of hateful speeches and cyberbullying.*

One can use these classifiers in these areas where the model makes sure no content is posted which violates the social platforms laws.Social listening and opinion classification Businesses are interested in hearing what their consumers have to say about them. One of the most efficient methods is to use sentiment analysis to categorize social media comments and reviews based on their emotional nature.

Sentiment analysis is a subset of NLP-based systems that focuses on deciphering the emotion, viewpoint, or attitude indicated in a text. They can distinguish between words with positive and negative implications. This is how we can automatically assess customer feedback or reactions to your products or services. For example, a business that designs airports uses sentiment analysis to categorize criticism left on social media by tourists. Managers can use opinion mining to make better decisions, win contracts, and deliver better services.

Keywords: Text, Classification, Document, Categorizing, Python, Model Tuning, Supervised Learning.

1. INTRODUCTION

Text Document Classification using Supervised Machine Learning Algorithms. In the Document Classification, we have used multiple ML models such as KNNs, Naive Bayes, SVMs, Random Forest, and a comparative analysis of each model. Machine learning-based text classification is considered to be more useful for applications that have the classification of text documents in a soft format. These applications and their importance can be identified from the use of spam filtering in email, web services, fake news detection, and opinion analysis mining. As online articles and blogging has taken popular in online services using the web, text and article classification plays an important role in this field.

1.1 Document Classification

Document classification is the process of labeling documents using categories, depending on their content. Document classification can be manual or automated and is used to easily sort and manage texts, images, or videos. There are advantages and disadvantages to both processes. Classifying documents manually gives humans greater control over the process of classification, and they can make decisions as to which categories to use. However, when handling large volumes of documents, this process can be slow and repetitive. Hence, it is much faster, more cost-efficient, and more accurate, to carry out document classification using machine learning. In order to carry out this task, we need to create a solution workflow, and the first step is to find out about the data and its characteristics. Document classification can also be done using OCR where ML models recognize the structure of the document so it can be analyzed and segregated into different sectors. Here we are performing classification based on the text of the document and not the way it is structured. On a general level, we analyze the frequently occurring words in the documents of each category, and then we train the model accordingly. When a new document comes in, we check the words in that document with the words in the training classes and then predict its label accordingly.

1.2 Text and Document Classification

Even though these terms sound very similar there is one major point standing between them. For example, in document classification, we analyze the entire document and get a broad understanding of what the article is talking about.

But we can go deeper into the document, divide it into bits of text, and get a granular understanding of the documents with text bits and the context and emotion they represent.

A document may talk about the pros and cons of a certain topic, it depends on us to determine the level of detail we want to dwell into. This was a very simple explanation. When we are working with more complex text classification problems, we require natural language processing or NLP.

NLP is made up of many different disciplines like computer science , grammar, statistics, and structural maintenance of a text.NLP is a machine learning technology, which means it requires a large amount of data to train a model. However, it can help with more complex text categorization problems like evaluating comments, articles, reviews, and other media materials. ML models are trained to recognize categories automatically, as opposed to the human-crafted if-then logic we write for rule-based systems. To learn statistical relationships between words and phrases, ML training presupposes that we feed historical data to the model with specified categories and a set of characteristics.

2. Dataset

Our dataset needs to contain enough samples of documents of each category so that the model can train on it.Having inconsistent data led to problems such as vanishing of entire categories since not enough samples were present on it to train on. We have used the BBC News Classification dataset for our model. It has a total of 1490 unique articles in 5 categories namely, business, entertainment, politics, sports, and tech. Making sure the quality of the dataset is the most important part in any ML application. If the data labeling itself is wrong, one cannot blame the ML model for underperforming. Data, or papers in this context, are preprocessed to make them viable and predictable. This entails a variety of actions, such as lower-casing all of the words, converting them to a root form (walking – walk), or simply leaving the root. The material may then be cleansed of stop words (and, the, is, are) and normalized, bringing together diverse spellings of the same terms (okay, ok, kk).

3. Modelling

Before Traning any text data is converted into numerical values using many different methods. One such method used in this implementation is TfidfVectorizer.

In TfidfVectorizer we consider the overall document weightage of a word. It helps us in finding the most frequent words. TfidfVectorizer finds the word counts by a measure of their frequency in the documents. The formula for this method would be

N - Number of documents

d - individual document

D - a collection of all documents

w - given word in a document

The first step would be to calculate the term frequency

$tf(w,d) = log(1+ f(w,d))$

f(w,d) is the frequency of the word in the document.

Now to Calculate the Inverse term frequency.

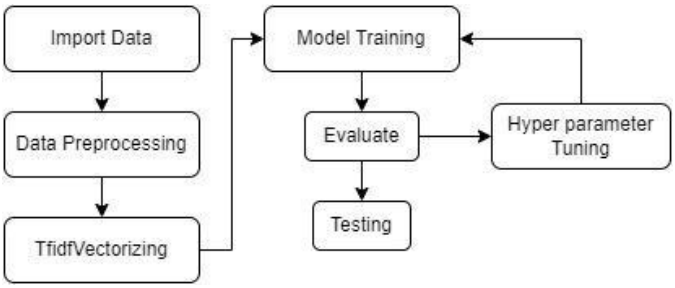
$idf(w,D) = log(N / f(w,D))$

Lastly, we need to find the Term Frequency — Inverse Document Frequency.

$tfidf(w,d,D) = tf(w,d) * idf(w,D)$

Traning is performed on the dataset using many different models to analyze the results. There are many complex algorithms one can use to create a classification model. We have used ML models such as KNNs, Naive Bayes, SVMs, and Random Forest and performed hyperparameter tuning on each model.

Here is the overview architecture of the entire process of the project.



For testing and improving performance. A base model is created with no tuning at all so that it can be compared to other tuned models.

3.1. Hyperparameter Tuning

Hyperparameter tuning is an important process of optimizing the behavior of a machine learning model. If we don't correctly tune our hyperparameters, our estimated model parameters produce suboptimal results, as they don't minimize the loss function. This means our model makes more errors.

We have performed tuning on the mentioned models with KFold Cross-Validation and GridSearchCV. KFold ensures the appearance of every data point from the dataset in the training and the evaluation set. It is a process to estimate the performance of the same model on new and unseen data. The general procedure in KFold is shuffling the dataset randomly. Then splitting the entire dataset into K small sets.

For each group, consider it a test set. Now select the rest of the groups as training data for the model. Train the model and evaluate it on the test set. The value for k is very essential in modeling. Generally, a value of between 5 to 10 is used for KFold.

Parameters used in tuning for a few models are mentioned below.

KNN:

params = {'n_neighbors': [3, 4, 5, 6], 'weights': ['uniform', 'distance'], 'metric': ['euclidean', 'manhattan']}

SVM:

params = {'C': [0.1, 1, 10, 100, 1000], 'gamma': [1, 0.1, 0.01, 0.001, 0.0001], 'kernel': ['rbf']}

RFC :

params = {'n_estimators': [100, 200, 400], 'criterion': ['gini', 'entropy'], 'max_features': ['auto', 'sqrt', 'log2']}

MNB:

params = {'alpha' = 1, 'fit_prior' : bool}

4. Results

Multinomial Naive Bayes is the best performing model with hyperparameter tuning. It is also called a probabilistic classifier. It is named Naive because it considers that the occurrence of one feature has no relation to the occurrence of other features.

The Naive Bayes method is based on Bayes' Theorem, which allows us to calculate the conditional probabilities of two occurrences based on the probabilities of each individual event. So, for a given text, we calculate the probability of each tag and then output the tag with the highest probability.

Formula :

$$P(A \mid B) = P(B \mid A) \times P(A) \;/\; P(B)$$

The chance of A being true if B is true is equal to the probability of B being true if A is true,divided by the probability of B being true.

This means that any vector representing a text must include information on the probabilities of particular words appearing in texts belonging to a given category, so that the algorithm may calculate the likelihood of that text belonging to that category.

It results in an accuracy of 0.95 which is quite good for a large text classification model.

Here is the multilabel classification matrix for the best model.

A/P	Business	Fun	Politics	Sport	Tech
Business	158	1	12	0	1
Fun	0	106	2	0	5
Politics	3	0	111	0	1
Sport	1	0	0	139	0
Tech	2	2	0	2	122

To find the Positives and Negatives of the category Business, we use some tricks.

True Positive: **158**

False Negative: 1 + 12 +0 + 1 = **14**

False Positive : 0 + 3 + 1 + 2 = **6**

True Negative: 106 + 2 + 0 + 5 +
0 + 111 + 0 + 1 +
0 + 0 + 139 + 0 +
2 + 0 + 2 + 122 = **490**

Using these values one can calculate evaluation metrics like Precision , F1 Score , Recall.

5. Scalability

These models can be fit into large corporations for classifying thousands of text documents into their required needs. companies can use such documents to classify resumes. Other media companies can use it to target hate speech and block certain content on the internet. The applications and scalability of this implementation are vast and its usage in the industry is rapidly growing.

Any text format can be taken and trained according to the provided labels. A text can be classified into different topics, into different sentiments. This model can be used to detect if there is hateful speech and illegal content in the given text. A good implementation would be to identify emotions in chats so a given AI can reply accordingly based on the predicted emotion.

For implementation, one can also try text classification without supervision. The model can scan an existing dataset and detect similarities between documents using NLP to comprehend the context of words. The set of comparable papers is then divided into clusters. This cluster will comprise entries that have content that is theoretically related.

As mentioned above one could try image categorization without supervision. Unsupervised learning methods can also be used to find repeated patterns in photos and cluster them according to comparable attributes. It's possible that this is an instance of object classification: The model is trained to recognize items in images and determine which class they belong to. Unsupervised models, like clustering in text classification, will need a lot of data to understand how to distinguish between object kinds and which items exist.

6. Conclusion

This project is successfully implemented and works and can be implemented as plugins into different large software. This project can also be implemented as a google chrome extension for individual users to help them classify their documents. Naive Bayes is the best model for Text Classification and text-related problems.

REFERENCES

[1] Comparative Study of Long Document Classification doi.org/10.48550/arXiv.2111.00702
1 Nov 2021

[2] Text Document Classification using Convolutional Neural Networks. ISSN:2349-5162, Vol.7, Issue 6 jetir.org/papers/JETIR2006387.pdf
June-2020

[3] Text Document Classification: An Approach Based on Indexing. DOI:10.5121/ijdkp.2012.2104
January 2012