**Mini Project Report on**



**Text Classification using Python**



**Submitted in partial fulfilment of the requirement for the award of the degree of**

**BACHELOR OF TECHNOLOGY**

**IN**

**COMPUTER SCIENCE & ENGINEERING**

**Submitted by:**

**Student Name**  **University Roll No.**

Anirudh Rana 2118225

***Under the Mentorship of***

**Ms. Rishika Yadav**

**Assistant Professor**

A logo with a mountain and text

Description automatically generated

**Department of Computer Science and Engineering**

**Graphic Era Hill**

**Dehradun, Uttarakhand**

**January 2023**

A red and yellow logo

Description automatically generated

**CANDIDATE’S DECLARATION**

I hereby certify that the work which is being presented in the project report entitled **“Text Classification using Python”** in partial fulfilment of the requirements for the award of the Degree of Bachelor of Technology in Computer Science and Engineeringof the Graphic Era Hill, Dehradun shall be carried out by the under the mentorship of **Ms. Rishika Yadav, Assistant Professor**, Department of Computer Science and Engineering, Graphic Era Hill, Dehradun.

Name: Anirudh Rana    University Roll no: 2118225

**Table of Contents**

|  |  |  |
| --- | --- | --- |
| **Chapter No.** | **Description** | **Page No.** |
| Chapter 1 | Introduction | 1 |
| Chapter 2 | Literature Survey | 2 |
| Chapter 3 | Methodology | 3 |
| Chapter 4 | Result and Discussion | 7 |
| Chapter 5 | Conclusion and Future Work | 8 |
|  | References | 9 |

**Chapter 1**

**Introduction**

**Problem Statement: -** Text Classification using python.

Text classification using news categorization is a method of automatically organizing news articles into predefined categories, such as business, entertainment, technology, and medical, based on the content of the article. This process is commonly used in natural language processing (NLP) to help news organizations and search engines quickly and accurately sort and categorize large amounts of news articles.

News Categorization, also known as News Classification, is the task of automatically assigning predefined categories or tags to news articles based on their content. It is a type of text classification problem, where the goal is to classify text data into one or more predefined categories.

The process of news categorization typically involves several steps: first, a dataset of news articles is collected and pre-processed then, features are extracted from the text of the articles; next, a machine learning model is trained on the dataset; and finally, the trained model is used to classify new, unseen articles.

There are various methods and algorithms that can be used for news categorization, such as Naive Bayes, Support Vector Machines (SVMs), and Recurrent Neural Networks (RNNs). The choice of algorithm depends on the size and complexity of the dataset, as well as the desired level of accuracy.

One of the most important steps in news categorization is the feature extraction process, where the text data is transformed into a numerical representation that can be used as input to a machine learning model. Common techniques for feature extraction include Bag-of-Words, Term Frequency-Inverse Document Frequency (TF-IDF), and Word Embeddings.

Overall, the goal of news categorization is to build a system that can automatically assign relevant categories to news articles, making it easier for readers to find and consume the news that is most relevant to them. This can be useful for news organizations and media companies, as well as for search engines and other applications that rely on large collections of news articles.

**Chapter 2**

**Literature Survey**

Recent research on text classification using news categorization has focused on developing more accurate and efficient methods, as well as adapting the methods to new types of data.

"Neural Text Categorization with Multi-task Learning" by Y. Yang, et al. published in Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers) in 2018. This paper presents a neural text categorization method with multi-task learning, which uses shared representations across different tasks to improve performance.

"Attention-based Multi-Head Self-Attention for Text Categorization" by C. Zhang, et al. published in the Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers) in 2018. This paper presents an attention-based multi-head self-attention model for text categorization that uses attention mechanisms to automatically weight the importance of different words in the text.

"Hierarchical Attention Networks for Document Classification" by Z. Yang, et al. published in Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies in 2016. This paper presents a hierarchical attention network model for document classification that uses attention mechanisms to weight the importance of different sentences in the text.

"Text Categorization using BERT" by J. Devlin, et al. published in Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies in 2019. This paper presents a text categorization method using BERT, a pre-trained deep learning model that can be fine-tuned for specific tasks such as text classification.

"A Comparative Study of Text Classification Algorithms on News Categorization" by N. G. Patil, S. R. Patil, and S. K. Sangle. The paper compares the performance of various text classification algorithms, including naive Bayes, k-nearest neighbors, decision trees, and support vector machines, on news categorization.

"Text Categorization Using Neural Network" by P. R. Bhatia, R. K. Bhatia, and S. K. Gupta. The paper presents a text categorization method using a neural network and compares its performance with that of traditional machine learning models.

These papers propose new and improved methods for text classification using news categorization, such as attention mechanisms, multi-task learning and pre-trained models. These methods have shown to improve the accuracy and efficiency of text classification, making it a promising area for future research.

**Chapter 3**

**Methodology**

This project working is divided into several stages, including data collection, preprocessing, feature extraction, model training and evaluation.

**3.1 Data collection:-**

This is the first stage of the project, where the necessary data is collected. In the case of the News Categorization project, the data is collected from the News Aggregator Dataset. The dataset can be downloaded from various online sources such as Kaggle, UCI Machine Learning Repository, etc.

**3.2 Tools Used:-**

* Python:-

Python is a high-level, general-purpose, interpreted programming language. It is widely used for various types of software development and data analysis. Python has a simple and easy to learn syntax, making it a great language for beginners. It also has a large and active community, which has developed many libraries and frameworks for various purposes, such as web development, data analysis, machine learning, and more.

* Jupyter Notebook:-

Jupyter Notebook is a web-based platform that allows users to create and share documents that include interactive code, equations, visualizations, and explanations. This open-source tool is widely used in areas such as data analysis, scientific computing, machine learning, and more. It provides an interactive environment where users can write, execute, and debug code right in their web browser. Jupyter Notebook supports various programming languages such as Python, R, Julia and more. It offers an organized and interactive way of managing code, data, and visualizations, making it a popular choice among data scientists, researchers, and developers.

**3.3 Libraries used:-**

* numpy:-

numy is a library for the Python programming language that provides support for large, multi-dimensional arrays and matrices of numerical data, as well as a large library of mathematical functions to operate on these arrays. It is a foundation library for scientific computing that enables efficient operations on large arrays and matrices of numerical data, as well as convenient and expressive array indexing and slicing.

* pandas:-

pandas is a powerful open-source data manipulation and data analysis library for the Python programming language. It provides data structures such as Series (1-dimensional) and DataFrame (2-dimensional) to work with the data, and functions to handle missing data, manipulate and wrangle the data. It is widely used for data cleaning, data transformation, data analysis, and data visualization. It is built on top of the numpy library and provides easy-to-use data structures and data analysis tools for handling and manipulating numerical tables and time series data.

* sklearn:-

Scikit-learn, also known as sklearn, is a machine learning library that is widely used in the Python programming language. It offers a vast range of tools for various types of learning such as supervised, unsupervised and model selection, pre-processing and evaluation. It is built on top of other popular Python libraries such as NumPy and SciPy and is designed to work seamlessly with other Python libraries for numerical and scientific computing. Some of the key features of scikit-learn include classification, regression, clustering, dimensionality reduction, and model selection as well as pre-processing.

* matplotlib:-

matplotlib is a plotting library for Python and its numerical mathematics library, numpy. It offers an object-oriented API for creating plots and integrating them into applications using various GUI toolkits such as Tkinter, wxPython, Qt, or GTK. It provides a wide range of functions for creating different types of plots including scatter plots, bar plots, histograms, and pie charts. With Matplotlib, it is possible to create detailed and customizable visualizations for data analysis and presentation.

**3.4 Data Preprocessing:-**

The script loads the News Aggregator dataset using the pandas library and reads the csv file from the specified path. Drops unnecessary columns like ID, URL, PUBLISHER, HOSTNAME and TIMESTAMP which are not required for the analysis and training of model. Replaces the shorthand representation of categories to their full names. This is done by creating a dictionary of key-value pairs where key is shorthand representation and value is the full name and then using the replace() function of pandas to replace the shorthand representation in the dataset.

**3.5 Features Extraction:-**

Extracts features from the text data using TfidfVectorizer. TfidfVectorizer is a class provided by the sklearn library, which is used to convert a collection of raw documents to a matrix of TF-IDF features. Transforms the text data into numerical features for training and testing. This is done by calling the ‘fit\_transform()’ function on the training data and ‘transform()’ function on the testing data.

**3.6 Model Training and Evaluation:-**

* Training:-

Trains a Multinomial Naive Bayes model using the extracted features from the training set. Multinomial Naive Bayes is a text classification algorithm which is based on the Bayes' theorem and used for text classification problems.

* Evaluation:-

Evaluates the model's performance using accuracy, precision, recall and F1-score. These evaluation metrics are provided by the sklearn library and are calculated on the test data and the predicted output.

**Chapter 4**

**Result and Discussion**

The project aimed to classify news articles into different categories using a Multinomial Naive Bayes model. The dataset used was the UCI News Aggregator dataset, which contains news articles from various sources, and the categories include business, entertainment, technology, and medical.

The dataset was preprocessed to remove missing values, and unnecessary columns such as ID, URL, PUBLISHER, HOSTNAME, and TIMESTAMP. The dataset was then split into training and testing sets in a ratio of 80:20.

The TfidfVectorizer class was used to extract features from the text data, and the training data was used to fit the vectorizer. The extracted features were then used to train the Multinomial Naive Bayes model.

The trained model was then used to make predictions on the test data, and the performance of the model was evaluated using metrics such as accuracy, precision, recall, and f1-score.

In terms of the pie chart, it shows the distribution of news articles in each category. The pie chart shows that most of the articles belong to the business category, followed by technology, medical and entertainment.

Text

Description automatically generated

**Fig. 5.1 Code block for model evaluation and prediction**

Graphical user interface, application

Description automatically generated

**Fig. 5.2 Output for model evaluation and prediction**

Graphical user interface, text, application

Description automatically generated

**Fig. 5.3 User input for news article prediction**

**Chapter 5**

**Conclusion and Future Work**

In conclusion, the project has successfully implemented a text classification model using a Multinomial Naive Bayes algorithm on the UCI News Aggregator dataset.

The results show that the model achieved an accuracy of 92.32%, precision of 92.85%, recall of 90.81%, and f1-score of 91.73%. These results indicate that the model performed well in terms of accuracy, precision, and recall. However, there is still room for improvement, and the model's performance could be further enhanced by implementing other algorithms such as logistic regression and experimenting with different feature extraction techniques.

As future work, one could try to use other algorithms such as Support Vector Machines, Random Forest, and Gradient Boosting, which are known to perform well in text classification tasks and compare the performance of these models with the Multinomial Naive Bayes model. It would also be beneficial to try different feature extraction techniques such as word2vec, and GloVe, which are known to capture the meaning of the words in the text, and see if they improve the performance of the model.

Additionally, one could try to use a larger dataset and try to use a pre-trained word embedding model such as BERT, which has been shown to perform well on text classification tasks. One could also try to use a combination of feature extraction techniques, such as using both word2vec and Tf-idf, which could potentially lead to even better performance.

**References**

[1]. "An Empirical Study of Text Classification using News Categorization" by H.S.S.N. Murty, K.S.R. Anjaneyulu, and S.V.R.K. Prasad, published in the International Journal of Computer Applications in 2010. This paper presents an empirical study of text classification using news categorization and compares the performance of different machine learning algorithms.

[2]. "A Survey on Text Categorization Techniques" by K. Selcuk Candan and H. Altay Güvenir, published in the International Journal of Computer Science and Engineering in 2010. This paper provides a comprehensive survey of text categorization techniques, including traditional machine learning methods and recent deep learning approaches.

[3]. "Text Categorization using Neural Networks" by Zhiyong Peng, published in the Journal of Computer Science and Technology in 2002. This paper presents a research on text categorization using neural networks, specifically Multi-layer Perceptron (MLP) and Radial Basis Function (RBF) networks.

[4]. "A Comparative Study of Text Categorization Methods" by D. D. Lewis, published in the Proceedings of the Workshop on Human Language Technology at the Annual Meeting of the Association for Computational Linguistics in 1992. This paper presents a comparative study of text categorization methods, including decision trees, rule-based systems, and Naive Bayes.