# Automated Crawling, Categorization and Sentiment Analysis of Digital News with Incorporated Feedback System

## Description:

This project aims to build a software which automatically crawls various news sources across the internet and then categorises the articles according to the respective government departments which are responsible for the involved domain. The news shall also be labelled as favourable (positive), neutral or not favourable (negative) to the government departments. Moreover, immediate notification of negative news pertaining to a particular department will be given to them via email. News sources from English, Hindi as well as regional language outlets are crawled and analysed by this system. Overall, a web application has been built where the user will be provided news articles along with their headings and descriptions. Also, the name of the government ministry to which it belongs will also be highlighted with the sentiment scores of the articles as its positivity, neutrality and negativity percentage. The user is also given the feature to filter news articles according to preferred ministries and languages. An option to refresh the news manually and an automatic refresh timer have also been provided.

First of all the application crawls the world wide web for news sources including text articles, images, e-newspapers as well as videos. This is done using a python program. It involves Django, a high-level web framework in Python to implement all the server side functionality of the code. For crawling the web, Beautiful Soup library is used. This library is used to scrap static pages or the pages which are directly generated by the server without any form of user input. It can parse components from both HTML and XML documents and text, images, audio and video can be easily extracted through this. For automated crawling of hidden pages or dynamic pages, Selenium library is used. This takes in input from the client and according to that searches the web for appropriate results. This way news articles separated into their headings and descriptions can be fetched and stored in a Comma Separated Values file. For video news, JSON files were also used to store

network logs extracted during headless mode of crawling videos from the automated browser. Audio from these video news was then extracted and the speech recognition library of python was used to obtain text from it. For articles in Hindi and regional languages, Google Translate API was used to translate the content of the news into English on which the machine learning techniques could be applied for classification and analysis. Finally the news dataset is obtained after crawling parsed into Heading and Description columns. No external APIs like NewsAPI were used to fetch the news because they do not provide support for regional languages. They can only be used on local devices for free and need subscription for deploying the application. Moreover, these third party sources only provide a limited number of news articles and cannot be used to train the model and get accurate predictions.

Machine Learning techniques are then applied on the obtained dataset. All the programs for this are written in Python using Jupyter Notebook files. As news categories are initially unlabelled, it was required to classify them into similar groups such that supervised learning techniques could be applied on them. On the raw news articles, preprocessing techniques were first implemented. These included visual analysis of the text, use of Regex and Spacy libraries to remove the stopwords and convert the remaining words into their root forms. Following this to label similar news as belonging to the same category, various clustering techniques were applied. These included K-means clustering, DBSCAN and HDBSCAN. First of all, embeddings were made of the text to get significant features as columns. Then dimensionality reduction was used using the UMAP library. Finally, only 2 components with the highest weightage in comparing similarity were kept and the rest were discarded. K-means analysis gave the best results on clustering and hence, was selected for the task. The clusters were then labelled according to the ministries under whose jurisdiction its news articles fell. This was done using analysis of important keywords featured in every cluster. Thus, this way labelled dataset of news according to respective ministries was created which could be further used to train the classification model.

After this, there was the need to train a machine learning model which could classify the news articles and provide the related government ministry as output. First of all classic sklearn models such Random Forest, Multinomial Naive Bayes,

Support Vector Classifier, Decision Tree Classifier, K Nearest Neighbours and Gaussian Naive Bayes were implemented. Among these, the Random Forest model gave the highest accuracy score of around 75%. Next up, a pretrained model named AutoKeras was used which resulted in an accuracy of 73%. The Roberta model was also implemented with similar scores. Finally, the DistilBERT model was tried, which gave a comparatively better accuracy of around 80%. Its parameters and hyperparameters were then fine-tuned eventually resulting in an accuracy score of 83%. Some other models and techniques were also tried and tested but the DistilBERT model gave the best predictions. Hence, it was selected as the final model. The model was saved as a .h5 file which could be loaded into the Django backend program.

Following this, the task of sentiment analysis of the classified news articles was achieved. For this, libraries such as Text-Blob were first tested. These provided promising results but failed on more complex input data. Then the pretrained Roberta model was used which gave excellent analysis. It reduced the development time and gave better results than most of the neural network models for text analysis. The model gave scores of positivity, neutrality and negativity of each news article according to which it could be classified as favourable or not favourable for the associated government agencies. If the news article came out to be majorly negative, it was required to send immediate notification to the respective department. This was made possible using Gmail SMTP feature and Nodemailer library. As soon as a news article is fetched and its classification and sentiment analysis is done, it is sent to the government ministry to which it belongs in a well specified format. This way appropriate actions could be taken at the earliest to resolve the issues.

All the news articles crawled and then analysed are finally displayed on an attractive and easy to navigate user interface. The frontend was created using Next.js framework in JavaScript and its integrated modules. TailwindCSS was used to style the website and make it visually appealing to the users. An option to refresh the news is provided to the users through which they could fetch the latest news articles from the web along with their classification and sentiment analysis report. If not refreshed manually, the news articles get updated after every passing hour. When the articles are loaded, they are displayed in the form of cards with

each card highlighting the government ministry to which the news belongs, the heading of the news and a brief introduction to its description. Following this, the sentiment analysis of the news is displayed with the positive, neutral and negative percentage scores of the article being shown to the user. Below this, there is the read more button, clicking on which redirects the user to the original article on the news outlet's website. This way users can go on and read their preferred news articles in detail. Moreover, on the navigation bar, there is the option to filter articles according to the respective government ministries under whose jurisdiction the news comes under. Also an option to filter news articles according to their languages has also been provided. This way users can select news according to their domains of interest as well as their language of choice. The frontend has been designed to provide an extremely smooth and seamless user experience to all the website visitors.

The integration of the machine learning models with the Django backend was done using .h5 files. APIs were implemented which called the predict functions of the models on the news articles crawled by the server code. The results of the crawled news and their analysis were then passed to the Next.js frontend using Fetch API calls. This way, the frontend, backend and models were linked together in a consistent manner. Hence, this way, all the programs created are combined together to create a fully functional web application with fast and reliable client-server communication, efficient web crawling for fetching news articles and accurate analysis of the results.

Overall, the project caters to the needs of all its major stakeholders, that is, the government departments, the common citizens of the country as well as the various news agencies. The citizens get better analysis of the news that they consume and can become aware about the varied perspectives and biases of different sources. This way, they can form a better and more informed opinion of events around them and not fall prey to yellow journalism and sensationalization. The government departments can get detailed reports about the news that are favourable as well as not favourable towards them. This will help them know their shortcomings and help in better decision making and public policies. Finally, the different news agencies will get to know about the sentiment scores of their published articles and videos. They will also get an analysis of public preference on their work. This will

help them improve their news reporting and make it more unbiased and informative such that it stays inline with the overall benefit of the society.

Hence, this project possesses the capacity to bring about a revolutionary change in the way in which digital news is distributed and consumed. It promises to reduce sensationalization in the media to a great extent and make the news landscape more unbiased, truthful and people centric in nature. This would lead to more informed opinions by the citizens, better government policy making and more responsible news publication and distribution. This would lead to higher development of the country, better quality of life and harmonious coexistence in the greater society.