iNeuron

HIGH-LEVEL DOCUMENT

News Article Sorting

Revision number – 1.2

Last date of revision - 11/9/2023

Authored by: Krishna Chandra Yadav

Document Version Control

Date	Version	Description	Author
02/9/2023	1.0	Abstract	Krishna Chandra Yadav
08/9/2023	1.1	Design Flow	Krishna Chandra Yadav
11/9/2023	1.2	Performance Evaluation Conclusion	Krishna Chandra Yadav

Contents:

Abstract:

This project focuses on improving the classification of news articles by leveraging the power of TF-IDF (Term Frequency-Inverse Document Frequency) for text vectorization and employing a range of machine learning algorithms for performance evaluation. In an era of information overload, efficient news article classification plays a pivotal role in helping user's access relevant content quickly and aiding advertisers in targeting their audience effectively.

The project employs TF-IDF, a widely used technique in natural language processing, to convert the textual content of news articles into numerical vectors. These vectors capture the importance of terms within articles, enabling the algorithms to understand the content better.

To assess the performance of the classification task, four powerful machine learning algorithms—Support Vector Machine (SVM), Random Forest (RF), Gradient Boosting Classifier, and AdaBoost are employed. Each algorithm brings its unique strengths, from SVM's ability to handle high-dimensional data to Random Forest's ensemble-based robustness.

The project evaluates these algorithms based on accuracy to determine their effectiveness in classifying news articles accurately. Through extensive experimentation and analysis, we aim to identify the algorithm that offers the best trade-off between precision and computational efficiency for news article classification.

Keywords: Natural Language Processing, machine learning, TF-IDF, content recommendation, news article classification, Support Vector Machine, Random Forest, Gradient Boosting Classifier, Ada-Boost, classification system.

1. Introduction

1.1 Why this HLD Document?

The main purpose of this HLD document is to feature the required details of the project and supply the outline of the Model Creation, Evaluation and Deployment. This additionally provides the careful description on however the complete project has been designed endto-end.

The HLD will:

- Present of the design aspects and define them in detail.
- Describe the user interface being implemented.
- Describe the hardware and software interfaces.
- Describe the performance requirements.
- Include design features and architectural design of the project.
- List and describe the non - functional attributes like:
    - Security
    - Reliability
    - Maintainability
    - Portability
    - Reusability
    - Resource Utilization

1.2 Scope

The HLD documentation presents the structure of the system, such as database design, architectural design, application flow, and technology architecture. The HLD uses non-technical terms to technical terms that can be understandable to the administrator of the system.

1.3 Definitions

Term	Description
FFP	News Article Sorting
Dataset	BBC News Article Data
Jupyter-Notebook	It is an interactive computational environment, in which you can combine code execution, rich text, mathematics, plots and rich media.
Visual Studio Code	Visual Studio Code is a code editor redefined and optimized for building and debugging modern web and cloud applications.
Amazon Web Services	AWS, is an online platform providing cost-effective, scalable cloud computing solutions.

2. General Description

2.1 Problem Perspective

In today's world, data is power. With News companies having terabytes of data stored in servers, everyone is in the quest to discover insights that add value to the organization. With various examples to quote in which analytics is being used to drive actions, one that stands out is news article classification.

2.2 Problem Statement

Nowadays on the Internet, there are a lot of sources that generate immense amounts of daily news. In addition, the demand for information by users has been growing continuously, so it is crucial that the news is classified to allow users to access the information of interest quickly and effectively. This way, the machine

learning model for automated news classification could be used to identify topics of untracked news and/or make individual suggestions based on the user's prior interests. Nowadays on the Internet there are a lot of sources that generate immense amounts of daily news. In addition, the demand for information by users has been growing continuously, so it is crucial that the news is classified to allow users to access the information of interest quickly and effectively. This way, the machine learning model for automated news classification could be used to identify topics of untracked news and/or make individual suggestions based on the user's prior interests.

The main objective here is -

1. **Enhance Information Accessibility:** The primary objective is to develop a robust news article classification system that improves information accessibility for users. This involves efficiently categorizing news articles into relevant topics or themes, making it easier for users to find and access the news that aligns with their interests.
2. **Personalization and Recommendation**: The project aims to implement machine learning techniques to enable personalization and recommendation features. This includes tailoring the news content based on users' preferences and prior interactions, enhancing user engagement, and delivering individualized news suggestions for a more tailored user experience.

2.3 Proposed Solution

We used 5-fold cross-validation on **SVM, Adaboost, Random Forest, Gradient Boosting Classifier** to know the best algorithm for the model. SVM algorithm gave the best results as compared with other models in 5-fold crossvalidation. SVM is used to train our model.

2.4 Further Improvements

1. Multi-Language Support: Extend the project to handle multiple languages effectively. This involves adapting the text preprocessing pipeline, selecting appropriate language-specific stop words, and potentially exploring multilingual models to accommodate a diverse user base.

2. Real-time Updates and Event Detection: Enhance the system's capability to provide real-time updates and event detection. Incorporate natural language processing techniques to identify emerging news topics and events as they happen, allowing users to stay informed about breaking news and trending stories in real time.

2.5 Technical Requirements

Hardware -

Processor - CPU or GPU

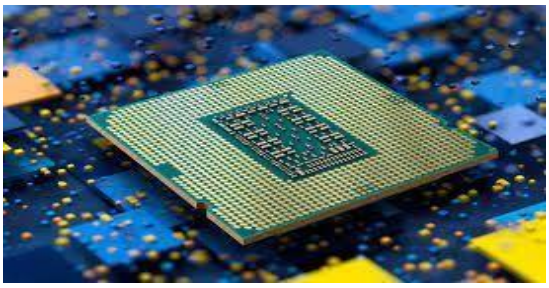
Memory - RAM

Storage - Hard Drive or SSD

Software -

Programming language - Python
ML libraries or Frameworks - Pandas, Numpy, Seaborn, Matplotlib, Sklearn, Nltk, Spacy, pickle.

Deployment – Amazon Web Services, flask



2.6 Data Requirements

Only Articles required which belong or are closed to Sports, Politics, Entertainment, and Technologies because our model is trained on the dataset which is having the above four categories.

2.7 Tools used

- Python 3.9 is employed because the programming language
- frameworks like Pandas, numpy, sklearn, Nltk, Spacy, pickle and alternative modules for building the model.
- Jupiter-Notebook is employed as an IDE.
- For Data visualizations, seaborn and components of matplotlib are getting used.
- Front end development is completed with Html
- Flask is employed for each information and backend readying.
- GitHub is employed for version management.
- Amazon Web Services is employed for deployment.
- Visual studio for backend.

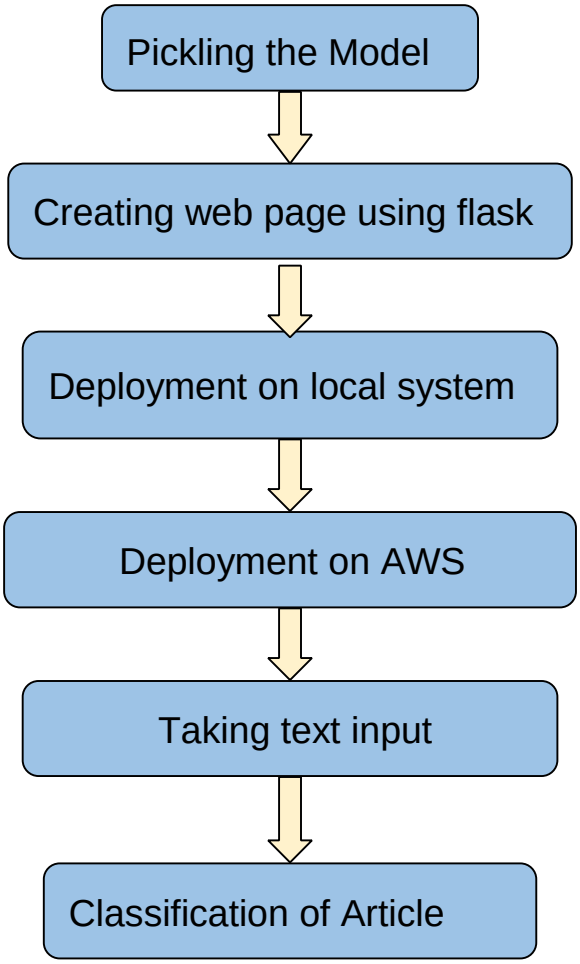
3. Design Flow

3.1 Model creation and evaluation

After preprocessing the data, we visualize our data to gain insights and then these insights are randomly spread and split into two parts, training and testing data. After splitting the data, we used 5-fold cross validation on **Svm, Random Forest, Gradient Boosting classification, and Adaboost** to know the best algorithm for the model. SVM algorithm gave the best results as compared with other models in 5-fold cross validation.

The required input Article in the form of text data is retrieved by using web pages.

3.2 Deployment Process



4. Performance Evaluation

4.1 Reusability

The code written and the components used should have the ability to be reused with no problems.

4. 2 Application Compatibility

The different parts of the system are communicating or using Python as an interface between them. All the components have its own tasks to perform and it is a job of a Python to ensure proper transfer of data.

4.3 Resource Utilization

When a task is performed, it'll doubtless use all the process power offered till the process is finished.

4.4 Deployment

Here model deployment is done using Amazon Web Services.

5. Conclusion

In the realm of news article classification, this project has successfully leveraged a robust data preprocessing pipeline and a variety of machine learning algorithms to enhance the organization and accessibility of news content. Through meticulous data preprocessing steps encompassing lowercasing, punctuation removal, stop word elimination, tokenization, and TF-IDF vectorization, the raw text data has been transformed into a format conducive to machine learning analysis.

The selection of machine learning algorithms, including Support Vector Machine (SVM), Random Forest (RF), Gradient Boosting, and AdaBoost, has allowed for a comprehensive exploration of classification techniques. Each algorithm has been

trained, tuned, and evaluated rigorously, with a focus on key performance metric which is accuracy score.

Through this diligent model-building process, we have identified the most effective algorithm for news article classification, meeting the project's objectives of delivering relevant content to readers and facilitating targeted advertising. The insights gained from this project not only improve content organization and personalization but also contribute to the broader field of natural language processing and text classification.

In summary, the synergy of advanced data preprocessing techniques, TF-IDF vectorization, and machine learning algorithms has empowered this project to achieve its goals of enhancing news article classification. The knowledge and methodologies gained here can be applied across various domains where efficient text classification is paramount, promising continued advancements in content delivery, user engagement, and data-driven decision-making.