**High Level Design (HLD)**

**NEWS ARTICLES SORTING**

**Revision Number: 1.0**

**Last Date of Revision: 29/01/2023**

# **Document Version Control**

|  |  |  |  |
| --- | --- | --- | --- |
| Date Issued | Version | Description | Author |
| 29/01/2023 | **1** | **Initial HLD – V1.0** | **Vinod Kumar Jodu** |
| 9/02/2023 | **1.1** | **Architecture Updated** | **Vinod Kumar Jodu** |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

**Table of Contents**

[Document Version Control 2](#_Toc126828304)

[Abstract 5](#_Toc126828305)

[1. Introduction 6](#_Toc126828306)

[1.1 Why this High-Level Design Document? 6](#_Toc126828307)

[The HLD will: 6](#_Toc126828308)

[1.2 Scope 6](#_Toc126828309)

[1.3 Definitions 6](#_Toc126828310)

[2. General Description 7](#_Toc126828311)

[2.1 Product Perspective 7](#_Toc126828312)

[2.2 Problem statement 7](#_Toc126828313)

[2.3 Proposed Solution 7](#_Toc126828314)

[2.4 Further Improvements 7](#_Toc126828315)

[2.5 Technical Requirements 8](#_Toc126828316)

[2.6 Data Requirements 8](#_Toc126828317)

[2.7 Tools Used 8](#_Toc126828318)

[2.7.1 Hardware Requirements 9](#_Toc126828319)

[2.8 Constraints 9](#_Toc126828320)

[2.9 Assumptions 9](#_Toc126828321)

[3. Design Details 10](#_Toc126828322)

[3.1 Process Flow 10](#_Toc126828323)

[3.1.1 Model Training 11](#_Toc126828324)

[3.1.2 Model Deployment Process 11](#_Toc126828325)

[3.2 Event log 12](#_Toc126828326)

[3.3 Error Handling 12](#_Toc126828327)

[4. Performance 12](#_Toc126828328)

[4.1 Reusability 12](#_Toc126828329)

[4.2 Application Compatibility 13](#_Toc126828330)

[4.3 Resource Utilization 13](#_Toc126828331)

[4.4 Deployment 13](#_Toc126828332)

[5. Dashboards 13](#_Toc126828333)

[5.1 KPls (Key Performance Indicators) 13](#_Toc126828334)

[6. Conclusion 14](#_Toc126828335)

[7. References 15](#_Toc126828336)

# **Abstract**

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. 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.

# **1. Introduction**

# **1.1 Why this High-Level Design Document?**

The purpose of this High-Level Design (HLD) Document is to add the necessary detail to the current project description to represent a suitable model for coding. This document is also intended to help detect contradictions prior to coding, and can be used as a reference manual for how the modules interact at a high level.

# **The HLD will:**

* Present all 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 the architecture of the project
* List and describe the non-functional attributes like:
* Security
* Reliability
* Maintainability
* Portability
* Reusability
* Application compatibility
* Resource utilization
* Serviceability

# **1.2 Scope**

The HLD documentation presents the structure of the system, such as the database architecture, application architecture (layers), application flow (Navigation), and technology architecture. The HLD uses non-technical to mildly-technical terms which should be understandable to the administrators of the system.

# **1.3 Definitions**

**Term Definition**

|  |  |
| --- | --- |
| AWS | Amazon Web Services |
| Database | Collection of News Article Sorting information |
|  |  |
|  |  |
|  |  |

# **2. General Description**

# 2.1 Product Perspective

The News Article Classification system is a Machine learning-based Classification model which will help us to classify the News Articles into it’s appropriate Category.

# 2.2 Problem statement

* Overwhelming amount of daily news makes it challenging for individuals to find relevant articles.
* Need to develop a machine learning system to automatically sort news articles into relevant categories.
* System should be able to accurately classify articles in real-time as they are published.
* The goal is to make it easier for individuals to find articles that match their interests and increase visibility of important news stories.
* To create a more efficient and user-friendly news reading experience.

# 2.3 Proposed Solution

* Train a machine learning model on a large dataset of labeled news articles.
* Utilize NLP techniques, such as text classification and clustering, to analyze article content.
* Teach the model to identify key features and patterns in the text indicative of specific topics.
* Use the trained model to sort new articles in real-time as they are published.
* The output will be a list of relevant categories for each article for efficient information retrieval.
* Result in a better reading experience for the user.

# **2.4** Further Improvements

* Incorporate contextual information, such as the date, author, and source of the article, to enhance the accuracy of the classification.
* Utilize multi-label classification techniques to allow articles to be sorted into multiple categories.
* Incorporate user feedback to continuously improve the accuracy of the model and refine the categories.
* Incorporate transfer learning techniques to leverage pre-trained models and reduce the amount of training data required.
* Implement a system for dynamic category creation and management to allow for the evolution of topics and categories over time.
* Integrate the sorting system with other news recommendation systems to provide a comprehensive and personalized news reading experience for the user.

# 2.5 Technical Requirements

* Strong NLP and text analysis skills to design and implement the machine learning model.
* Knowledge of machine learning algorithms, such as text classification and clustering.
* Experience with training and evaluating machine learning models on large datasets.
* Familiarity with Machine learning frameworks
* Knowledge of cloud computing and distributed systems to scale the solution for processing large volumes of news articles in real-time.
* Experience with software development, including programming languages such as Python and database management systems.
* Familiarity with news article datasets and the challenges of NLP in news article analysis.

# 2.6 Data Requirements

* Large dataset of labeled news articles, with each article categorized into relevant topics or categories.
* The dataset should include a diverse range of news articles from different sources, covering a wide range of topics.
* The articles should be pre-processed to remove any irrelevant information, such as advertisements and images.
* The dataset should be balanced, with an equal representation of articles for each category to prevent bias in the model training.
* The dataset should be updated regularly to incorporate new articles and to reflect changes in the news landscape.
* The dataset should be stored in a structured format, such as a database, to enable easy access and manipulation.

# 2.7 Tools Used

* Natural Language Processing libraries, such as NLTK and SpaCy, for pre-processing and analyzing the text of the news articles.
* Machine Learning frameworks, such as sklearn for training and evaluating the text classification model.
* Database management systems, such Cassandra, for storing and organizing the news article dataset.
* Cloud computing platforms, such as Amazon Web Services and Google Cloud Platform, for scaling the solution to handle large volumes of news articles.
* Software development tools, such as Python, for implementing the solution and integrating it with other systems.
* Data visualization tools, such as Matplotlib and Seaborn, for analyzing and interpreting the results of the model evaluation.

. 

* Top of Form

# 2.7.1 Hardware Requirements

* High-performance CPU and GPU to handle the computational demands of training and evaluating machine learning models.
* Large amounts of memory and storage to accommodate the news article dataset and the machine learning models.
* High-speed network connectivity to facilitate fast transfer of data between the different components of the system.
* Redundant power and cooling systems to ensure the reliability and availability of the system.
* Scalable hardware architecture to allow for future expansion as the volume of news articles and the size of the machine learning models increase.

# 2.8 Constraints

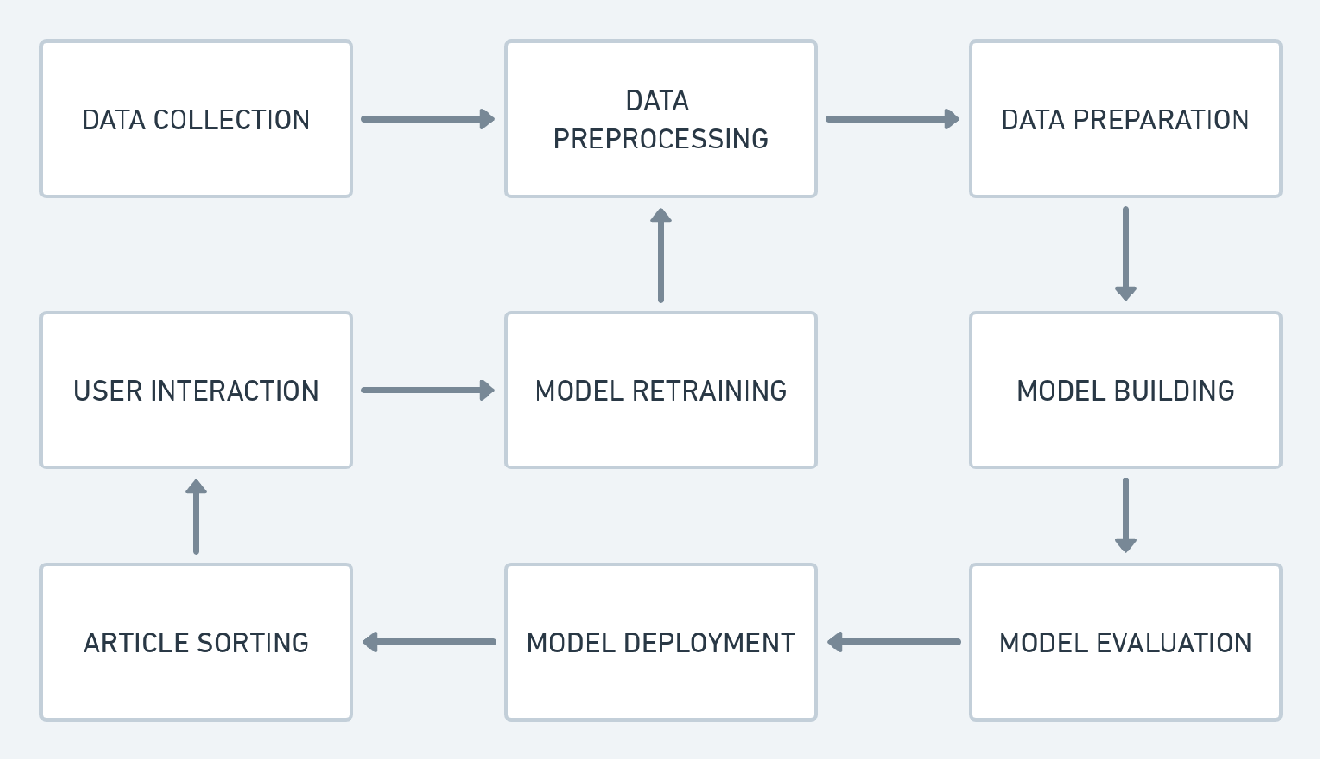
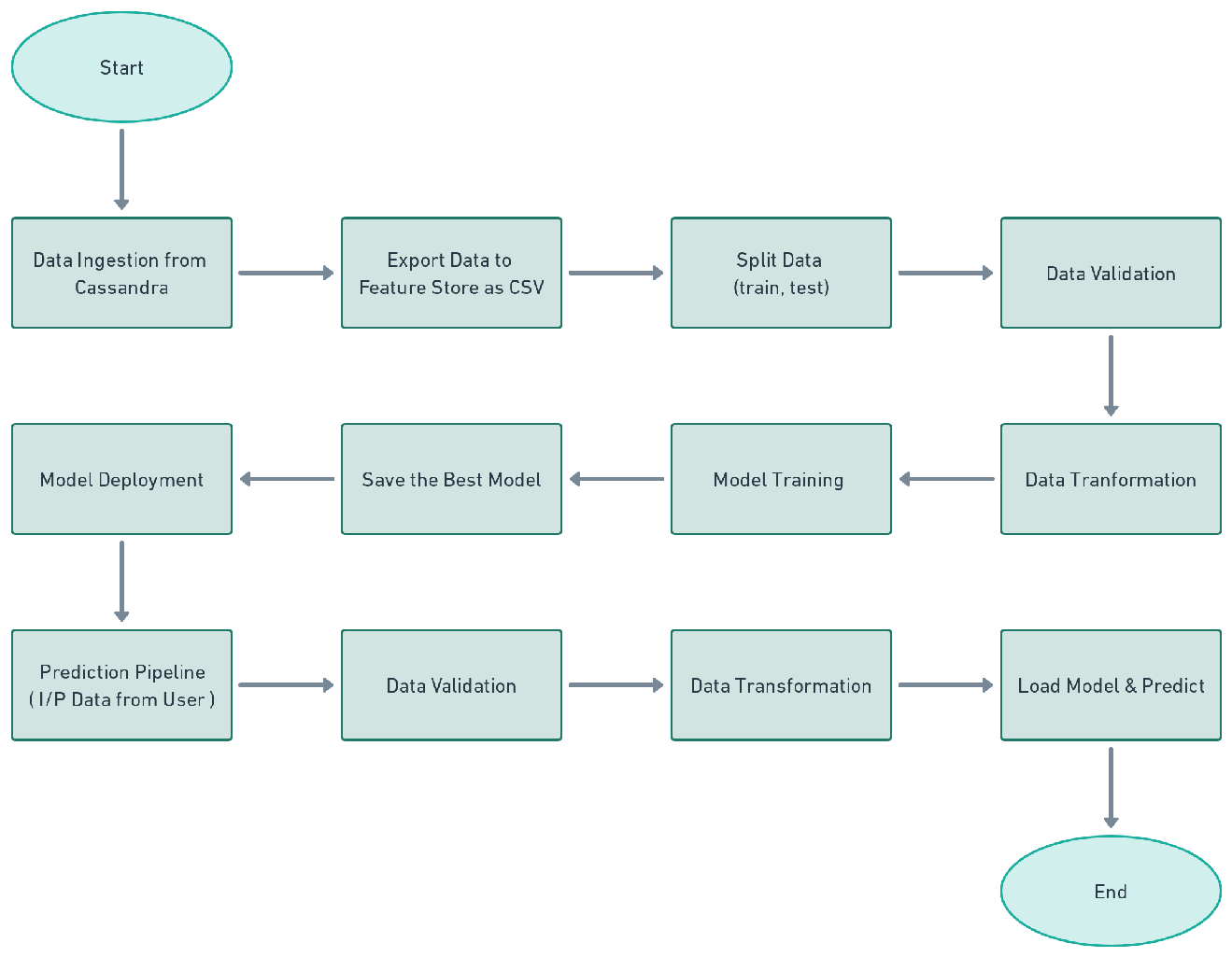
* Limited labeled training data, requiring careful pre-processing and augmentation techniques to increase the size of the training dataset.
* The complexity of natural language and the diversity of news topics can make it challenging to accurately categorize articles.
* Changes in the news landscape and the emergence of new topics require continuous refinement and updating of the categories used for sorting.
* The need for real-time processing and efficient information retrieval can place high demands on the performance and scalability of the system.
* Privacy and security concerns with processing large amounts of sensitive information from news articles.
* The cost of acquiring and storing large datasets and the computational resources required to train machine learning models.

# 2.9 Assumptions

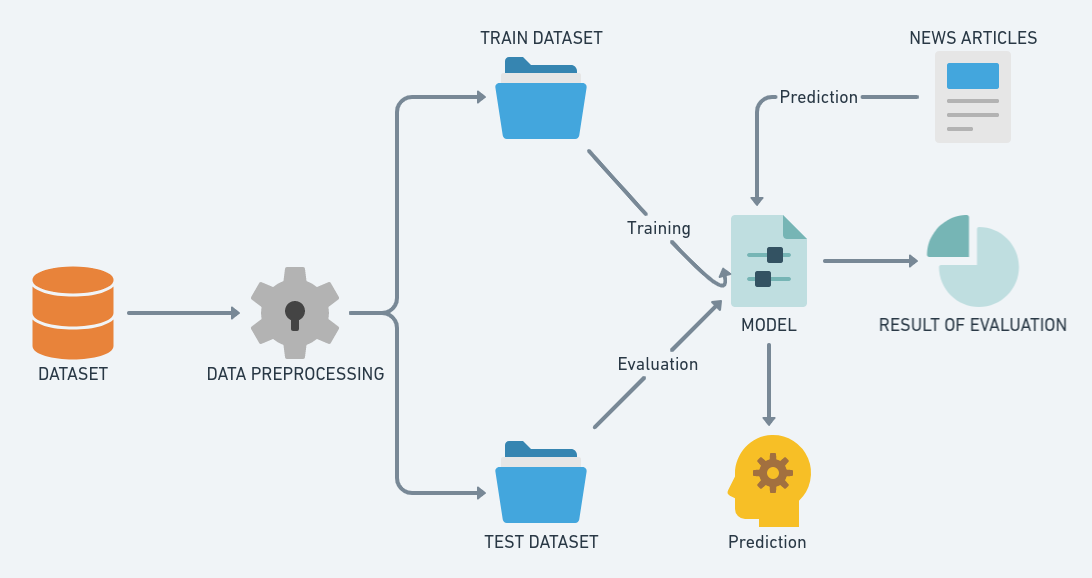
* The news articles have been pre-processed and cleaned to remove irrelevant information and ensure the quality of the text.
* The labeled training dataset accurately reflects the diversity of news topics and the distribution of articles across categories.
* The machine learning models will be trained using appropriate algorithms and parameters to achieve accurate article categorization.
* The system will be integrated with other news recommendation systems to provide a comprehensive and personalized news reading experience.
* The system will be regularly updated to incorporate new articles and to refine the categories used for sorting.
* The user's feedback and interactions with the system will be utilized to continuously improve the accuracy of the model.

# **3. Design Details**

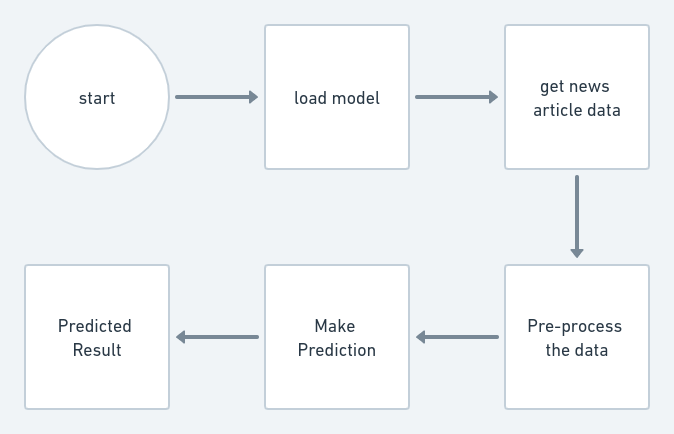
# 3.1 Process Flow



# 3.1.1 Model Training



# 3.1.2 Model Deployment Process



# 3.2 Event log

The system should log every event so that the user will know what process is running internally.

**Initial Step-By-Step Description:**

* The System identifies at what step logging required
* The System should be able to log each and every system flow.
* Developer can choose logging method. You can choose database logging/ File logging as well.
* System should not hang even after using so many loggings. Logging just because we can easily debug issues so logging is mandatory to do.

# 3.3 Error Handling

Should errors be encountered, an explanation will be displayed as to what went wrong? An error will be defined as anything that falls outside the normal and intended usage.

# **4. Performance**

* Accuracy: Evaluate the accuracy of the model in terms of the number of correctly categorized articles compared to the total number of articles.
* F1 Score: Calculate the F1 score, which is a measure of the model's precision and recall, to assess its overall performance in classifying the articles.
* Confusion Matrix: Create a confusion matrix to visualize the distribution of false positive and false negative predictions, and identify the categories where the model is struggling to accurately categorize the articles.
* AUC-ROC Curve: Plot the receiver operating characteristic (ROC) curve and calculate the area under the curve (AUC) to evaluate the model's ability to distinguish between the categories.
* User Feedback: Incorporate feedback from the user to evaluate the relevance and quality of the sorted articles, and identify areas for improvement.
* Real-Time Performance: Evaluate the performance of the system in real-time, including its processing speed, accuracy, and scalability.
* Continuous Improvement: Continuously evaluate and improve the performance of the model by incorporating new articles, adjusting the model parameters, and incorporating feedback from the user.

# 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 components for this project will be using Python as an interface between them. Each component will have its own task to perform, and it is the job of the Python to ensure proper transfer of information.

# 4.3 Resource Utilization

When any task is performed, it will likely use all the processing power available until that function is finished.

# 4.4 Deployment



# **5. Dashboards**

Dashboards will be implemented to display and indicate certain KPls and relevant indicators for the unveiled problems that if not addressed in time could cause catastrophes of unimaginable impact.

As and when, the system starts to capture the historical/periodic data for a user, the dashboards will be included to display charts over time with progress on various indicators or factors.



# 5.1 KPls (Key Performance Indicators)

1. Accuracy: The proportion of correctly categorized articles compared to the total number of articles.
2. F1 Score: A measure of the model's precision and recall in classifying the articles.
3. False Positive Rate: The proportion of articles that are incorrectly categorized as belonging to a certain category.
4. False Negative Rate: The proportion of articles that are incorrectly categorized as not belonging to a certain category.
5. User Satisfaction: The level of satisfaction expressed by the user with the relevance and quality of the sorted articles.
6. Processing Speed: The time taken for the system to categorize an incoming article.
7. Scalability: The ability of the system to handle increasing volumes of incoming articles.
8. Continuous Improvement: The rate at which the model is being improved and updated based on feedback from the user and new articles.

# **6. Conclusion**

News article sorting is an important task in today's fast-paced world, where people are constantly seeking information and staying updated on current events. The proposed methodology of using a machine learning-based approach, along with text-based and feature engineering techniques, provides a powerful solution for categorizing news articles into relevant topics. The performance of the model is evaluated using KPIs such as accuracy, F1 score, and user satisfaction, among others, and the model is continuously refined based on feedback from the user. The system provides real-time categorization of incoming news articles and is scalable to handle increasing volumes of data. In conclusion, the proposed solution provides a reliable and efficient way of sorting news articles and helps users stay informed on current events.

# **7. References**

1. Chen, D., & Guestrin, C. (2016). XGBoost: A Scalable Tree Boosting System. arXiv Preprint arXiv:1603.02754.
2. Kim, Y. (2014). Convolutional Neural Networks for Sentence Classification. arXiv Preprint arXiv:1408.5882.