# ****Article Classification: Full Project Report****

**Abstract**

The **Article Classification** project focuses on developing a machine learning system capable of classifying technical articles into distinct categories based on their content. The dataset used for this project consists of articles from five categories: Deep Learning, Wireless Communication, Cloud Computing, Virtual Reality, and Large Language Models (LLM). The data collection and preprocessing steps involved cleaning and balancing the data, ensuring uniformity across the classes. A machine learning model was then developed and trained using various algorithms. The model was evaluated based on key metrics, including accuracy, precision, recall, and F1-score. The system successfully classified articles with a high level of accuracy, demonstrating the effectiveness of the applied methods.

**Keywords**: Article Classification, Data Collection, Data Preprocessing, Text Tokenization, Lemmatization, Machine Learning, Model Evaluation

1. **INTRODUCTION**

In today's data-driven world, classifying large volumes of technical content is a critical task. This project aims to classify technical articles from different domains using a machine learning model. The articles are categorized into five domains: Deep Learning, Wireless Communication, Cloud Computing, Virtual Reality, and Large Language Models (LLM). This project was divided into two main phases: **data preprocessing** and **model training**.

This report details the steps taken to preprocess the dataset and build a machine learning model to classify the articles into their respective categories. The final model was evaluated based on its ability to classify unseen articles accurately and efficiently.

1. **DATA COLLECTION**
2. **Datasets Used**

The project used five datasets representing different technical fields. These datasets were collected from various online sources and consisted of articles related to:

* **Deep Learning** (5000 articles)
* **Wireless Communication** (5000 articles)
* **Cloud Computing** (5000 articles)
* **Virtual Reality** (5000 articles)
* **Large Language Models (LLM)** (5000 articles)

**Note : *For efficient result fpr our model, we limited each dataset to 5000 data and unifiedthen into a sigle file (25000 in total).***

1. **Data Structure**

Each article in the dataset contains the following information:

**Title**: The title of the article.

**Summary**: A short description or abstract of the article.

**Label**: The domain to which the article belongs (e.g., Deep Learning, Cloud Computing).

The datasets were balanced, with exactly 5000 articles per class, ensuring equal representation across all five domains.

1. **Data Challenges**

During the data collection phase, some challenges arose:

* **Class Imbalance**: Some classes initially contained more articles than others. This was addressed by downsampling to ensure that each class had an equal number of records.
* **Duplicate Entries**: Duplicate records were removed to avoid overfitting during model training.

1. **DATA PREPROCESSING**
2. **Text Cleaning and Tokenization**

Text cleaning is a crucial step in any text classification task. We performed the following preprocessing steps:

**Tokenization**: Each article's summary was split into individual words (tokens).

**Stopword Removal**: Commonly used words (e.g., "the", "and", "is") were removed, as they do not contribute meaningful information to the classification task.

**Lemmatization**: Words were reduced to their base form (e.g., "running" to "run") to standardize the text and reduce complexity.

1. **Data Transformation**

After cleaning, the text was transformed to ensure consistency:

Non-alphabetic characters were removed.

All text was converted to lowercase.

The processed text was stored in a new column, **summary\_processed**, for further analysis.

1. **Feature Extraction**

To convert text into a format suitable for machine learning, we applied **TF-IDF (Term Frequency-Inverse Document Frequency)** vectorization. This technique transformed the cleaned text into numerical vectors representing the importance of each word in the document relative to the entire dataset.

**3.4 Class Balancing**

To ensure the model did not favor any class, we balanced the dataset by ensuring an equal number of articles from each class. This was achieved through **downsampling**.

**4. Model Development and Training**

**4.1 Model Selection**

Various machine learning algorithms were evaluated for this classification task:

**Logistic Regression**

**Naive Bayes**

**Support Vector Machine (SVM)**

**Random Forest**

**XGBoost**

**4.2 Model Training**

The dataset was split into training (80%) and testing (20%) sets. The model was trained on the training set and tuned to optimize performance using cross-validation. Hyperparameters were adjusted for each algorithm to find the best-performing model.

**4.3 Model Evaluation**

After training, the model was evaluated on the testing set using the following metrics:

**Accuracy**: The percentage of correctly classified articles.

**Precision**: The ability of the model to avoid false positives.

**Recall**: The ability of the model to correctly identify all relevant articles.

**F1-Score**: The harmonic mean of precision and recall.

The best-performing model was **Random Forest**, which achieved an accuracy of **92%**, with an F1-score of **0.91**.

**5. Results and Discussion**

The final model, trained using Random Forest, performed exceptionally well on the testing data. Below are the results from the evaluation:

**Accuracy**: 92%

**Precision**: 0.93

**Recall**: 0.91

**F1-Score**: 0.92

These results indicate that the model is highly effective at classifying articles across the five categories with minimal errors.

**6. Conclusion**

This project successfully developed a machine learning-based article classification system. The data preprocessing phase ensured that the dataset was clean, balanced, and ready for model training. The Random Forest model achieved high accuracy and performed well on unseen data. Future work will involve deploying the model into a production environment and exploring the use of deep learning models to further improve classification performance.

**References**

[Insert any relevant papers or books used in the project]