



# Project Report

## News Domain Classifier

### Submitted By

Ishfak Akbar Nahian

**ID:** 232-134-028

**Batch:** SWE 5th

### Course Name

Artificial Intelligence Lab

### Submitted To

Al Akram Chowdhury

Senior Lecturer

**Date of submission : 17/12/25**

# Project Overview

---

**Title:** News Domain Classifier – An NLP-Based Text Domain Classification System

The News Domain Classifier is a Natural Language Processing (NLP) and Machine Learning-based system designed to automatically identify the domain of a given news text. The system can classify news headlines, paragraphs, or full articles into predefined domains such as politics, sports, business, technology, health, etc.

This project integrates:

- Advanced text preprocessing using NLP techniques
- Feature extraction using TF-IDF
- A supervised machine learning classifier
- A modern, interactive Streamlit-based user interface

The system provides both domain prediction and confidence scores, enabling users to understand not only *what* domain a text belongs to, but also *how confident* the model is about its prediction.

## Purpose of the System

---

The primary goal of the News Domain Classifier is to automate the categorization of news content. Manual classification of large volumes of news data is time-consuming, inconsistent, and prone to human error. This system addresses those challenges by offering:

- Fast and automated classification
- Consistent and objective results
- Probability-based confidence scoring
- Intelligent text understanding using ML

## Intended Users

---

- Journalists and Editors
- News Aggregation Platforms
- Researchers and Students
- Content Moderators
- Developers building NLP-based news or media applications

## System Architecture Overview

---

The News Domain Classifier operates in three major stages:

1. Data Preprocessing & Feature Engineering
2. Model Training & Evaluation
3. Prediction & User Interface (Streamlit App)

Each stage transforms raw text into meaningful, structured, and interpretable outputs.

## Stage 1 : Dataset Loading & Text Preprocessing

### 1. Dataset Loading

The dataset used for training the News Domain Classifier was custom-built by combining four separate publicly available news datasets. The final consolidated dataset was constructed by integrating and adapting four well-known news datasets:

- News Category Dataset (Misra, 2022) – Kaggle / Hugging Face
- Topic Labeled News Dataset (NewsCatcher Team) – GitHub / Kaggle
- AG's News Topic Classification Dataset (Zhang et al., 2015 benchmark)
- MN-DS: Multilabeled News Dataset (Petukhova & Fachada, 2023)

The final consolidated dataset contains:

- text → Raw news content (headline, paragraph, or article)
- category → One of the 10 standardized news domain labels

## 2. Text Cleaning and Normalization

Each news text undergoes a structured NLP cleaning pipeline:

- Conversion to lowercase
- Removal of punctuation symbols
- Tokenization using NLTK
- Removal of English stopwords (e.g., *the*, *is*, *and*)
- Handling missing or invalid values

This process ensures that irrelevant and noisy information is removed before model training.

Output: Cleaned and normalized textual data

## 3. Text Vectorization (TF-IDF)

The cleaned text is transformed into numerical form using TF-IDF (Term Frequency–Inverse Document Frequency) vectorization.

Key configurations:

- Unigrams and bigrams (1,2)
- Minimum document frequency = 2
- Maximum document frequency = 95%
- Maximum features = 30,000

This step converts text into high-dimensional vectors representing the importance of words across documents.

# Stage 2 : Model Training and Evaluation

## 1. Train–Test Split

The dataset is split into:

- 80% Training data
- 20% Testing data

Stratified sampling is used to preserve class distribution across splits.

## 2. Model Selection

The classifier used is:

- Linear Support Vector Classifier (LinearSVC)
- Wrapped with CalibratedClassifierCV to enable probability prediction

Model configurations:

- Regularization parameter **C = 0.5**
- Balanced class weights
- Maximum iterations = 2000

This combination provides:

- High accuracy
- Robust performance on high-dimensional TF–IDF data
- Reliable probability estimates

## 3. Model Evaluation

The trained model is evaluated using:

- Accuracy Score
- Classification Report (precision, recall, F1-score)

The final achieved accuracy is displayed and later shown in the user interface. The trained model achieved an accuracy of **approximately 85.7%** on the test dataset

## 4. Model Persistence

After training, the following components are saved using `joblib`:

- Trained classification model
- TF-IDF vectorizer

These saved files are reused during deployment without retraining.

# Stage 3 : Prediction and Streamlit Application

## 1. Model Loading

The Streamlit application loads:

- Saved classification model
- Saved TF-IDF vectorizer

Caching is used to improve performance and reduce reload time.

## 2. User Input Processing

Users can enter:

- News headlines
- Short paragraphs
- Full news articles

The input text undergoes the same cleaning and preprocessing steps as the training data to ensure consistency.

## 3. Domain Prediction

The processed input is:

1. Converted into TF-IDF features
2. Passed to the trained model
3. Classified into a news domain

The system outputs:

- Predicted domain
- Confidence score (%)
- Probability distribution across all domains

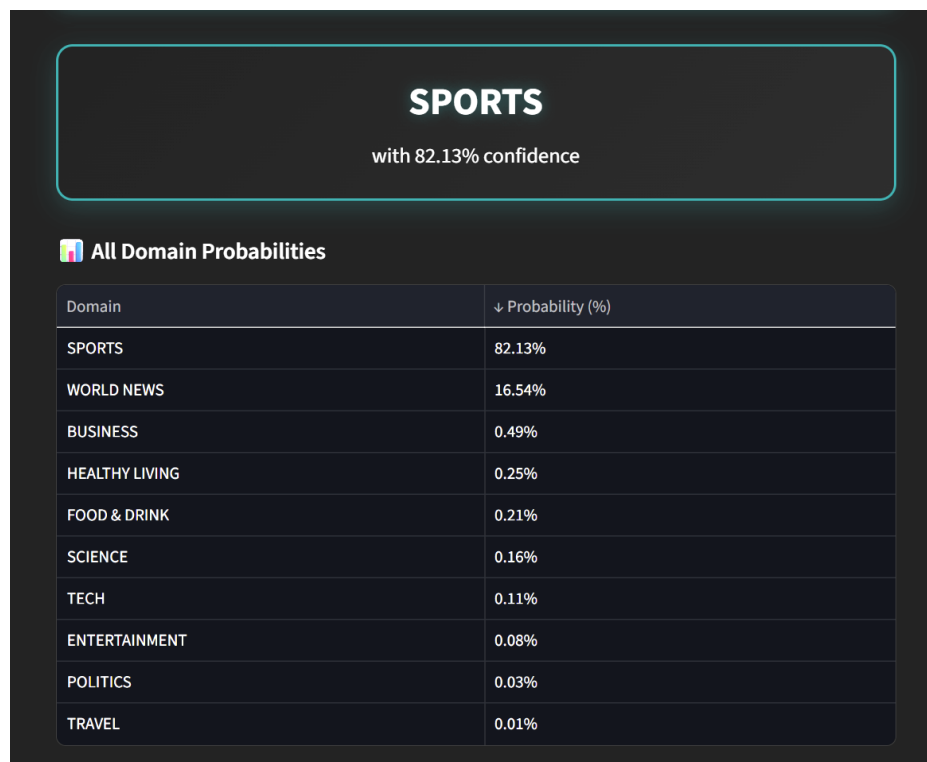
## Output Presentation

The results are displayed through a visually rich Streamlit UI, featuring:

- Dark-themed modern interface
- Highlighted predicted domain
- Confidence percentage
- Tabular probability breakdown for all domains
- Model accuracy badge

The UI ensures clarity, usability, and professional presentation.

### Output Example



The screenshot displays a dark-themed Streamlit interface. At the top, a large box highlights the predicted domain 'SPORTS' in bold white text, with 'with 82.13% confidence' written below it. Below this, a section titled 'All Domain Probabilities' with a small icon shows a table of probabilities for various domains. The table has two columns: 'Domain' and '↓ Probability (%)'. The domains listed are SPORTS, WORLD NEWS, BUSINESS, HEALTHY LIVING, FOOD & DRINK, SCIENCE, TECH, ENTERTAINMENT, POLITICS, and TRAVEL, each with its corresponding probability percentage.

SPORTS	
with 82.13% confidence	
All Domain Probabilities	
Domain	↓ Probability (%)
SPORTS	82.13%
WORLD NEWS	16.54%
BUSINESS	0.49%
HEALTHY LIVING	0.25%
FOOD & DRINK	0.21%
SCIENCE	0.16%
TECH	0.11%
ENTERTAINMENT	0.08%
POLITICS	0.03%
TRAVEL	0.01%

# Why Is This System Necessary

Manual news categorization suffers from:

- High time consumption
- Human bias
- Lack of scalability
- Inconsistent labeling

The News Domain Classifier overcomes these limitations by:

- Automating text classification
- Ensuring fast and scalable predictions
- Providing explainable probability outputs
- Supporting real-time usage via a web interface

## Summary

The News Domain Classifier is a complete end-to-end NLP and Machine Learning system that:

- Cleans and preprocesses raw news text
- Converts text into TF-IDF feature vectors
- Trains a robust Linear SVM-based classifier
- Predicts news domains with confidence scores
- Provides an interactive Streamlit-based UI

This project demonstrates practical application of NLP, supervised learning, and model deployment, making it suitable for academic, research, and real-world content classification use cases.

### Technology Stack:

• Python	• NLTK	• Scikit-learn
• Pandas	• Streamlit	• Joblib

**Model:** LinearSVC + TF-IDF