

---

# CAS2105 Homework 6: Mini AI Pipeline Project 🙌

A/P Satheskumar Sarvina (2024148001)

---

## 1 Introduction

This project explores a small AI pipeline for classifying news headlines. The chosen task is **news headline classification**, where each headline is assigned to one of four categories: **World**, **Sports**, **Business**, or **Sci/Tech**.

This task is interesting because accurate headline classification is widely applicable: it supports news aggregators, recommendation engines, and automated content moderation. It also illustrates the difference between simple heuristics and modern embedding-based approaches.

The project uses the **AG News dataset** due to its public availability, structured format, and suitability for fast experimentation on a manageable subset. We compare a naïve baseline (keyword-based) with an **AI pipeline** (MiniLM embeddings + Logistic Regression), highlighting the benefits of semantic embeddings.

Overall, this project emphasizes the AI workflow: **defining the problem, designing a baseline, building an improved pipeline, evaluating results, and reflecting on outcomes**. The pipeline is lightweight, reproducible, and interpretable, capable of running efficiently on a single GPU or CPU.

## 2 Task Definition

- **Task:** Classify news headlines into four categories: **World**, **Sports**, **Business**, and **Sci/Tech** using the AG News dataset.
- **Motivation:** Quick and reliable headline classification is useful for news aggregation, topic routing, personalized feeds, and automated content moderation. This project emphasizes the AI pipeline process: problem definition, baseline, model pipeline, evaluation, and reflection. It demonstrates the workflow of developing and comparing simple AI methods on a real dataset.
- **Input:** Single short news headline (text).
- **Output:** One of four labels {**World**, **Sports**, **Business**, **Sci/Tech**}.
- **Success criteria:** High classification accuracy and balanced macro F1 across classes on held-out test data. The system should be reproducible and interpretable for quick experimentation.

## 3 Dataset

- **Source:** AG News via Hugging Face datasets.
- **Subset used:** 250 training examples per class (1000 total) and 100 test examples per class (400 total).
- **Splits:** Train and test subsets, sampled to ensure balanced class representation.

- **Preprocessing:** Minimal. Headlines are lowercased for keyword baseline. AI pipeline uses raw text; tokenization handled internally by embedding model.
- **Rationale:** Using a small balanced subset allows fast experimentation while still demonstrating differences between baseline and AI pipeline performance.

## 4 Methods

### 4.1 Naïve Baseline

- **Method description:** Keyword-based classifier — each class has a curated list of keywords. Headlines are scored by keyword occurrences and assigned to the highest-scoring class. Ties or zero matches are resolved deterministically.
- **Why naïve:** It ignores semantics, word order, and polysemy. Cannot handle synonyms or paraphrased expressions.
- **Likely failure modes:**
  - Fails on ambiguous headlines or headlines without keywords.
  - Sensitive to word choice; minor phrasing changes can drastically alter predictions.
  - Example: “NASA launches new Mars rover” → baseline predicts **World** instead of **Sci/Tech**.

### 4.2 AI Pipeline

- **Models used:** `sentence-transformers/all-MiniLM-L6-v2` to compute sentence embeddings; `sklearn.LogisticRegression` as classifier.
- **Pipeline stages:**
  1. Preprocessing: raw headlines (tokenization handled by the embedding model)
  2. Embedding: encode headlines into 384-dimensional vectors (MiniLM)
  3. Classifier: logistic regression trained on embeddings
  4. Post-processing: predicted label mapping back to text
- **Design choices and justification:** Embeddings + linear classifier is inference-light, reproducible, and effective for short texts without fine-tuning. Meets CPU-friendly constraints.

## 5 Experiments & Results

### 5.1 Metrics

- Accuracy
- Precision (macro)
- Recall (macro)
- F1 (macro)

Method	Accuracy	Precision	Recall	F1	Notes
Baseline (keywords)	0.47	0.73	0.47	0.45	High precision but low recall; fails on paraphrased headlines
AI Pipeline (MiniLM+LR)	0.88	0.88	0.88	0.88	Consistent performance; captures semantic meaning

## 5.2 Qualitative Examples

Headline	Gold Label	Baseline Pred	AI Pred
“NASA launches new Mars rover”	Sci/Tech	World	Sci/Tech
“Stock market surges after tech earnings”	Business	World	Business
“Local team wins championship final”	Sports	Business	Sports

## 6 Reflection & Limitations

- **Successes:** Embedding + logistic regression pipeline outperforms baseline, handling paraphrased headlines and synonyms effectively.
- **Challenges:** Baseline fails on short, ambiguous, or synonym-rich headlines. Some classes overlap conceptually (e.g., World vs Business), causing occasional misclassification.
- **Metric suitability:** Accuracy and macro F1 appropriate for this balanced subset; micro-averaged metrics may be better for imbalanced datasets.
- **Future work:** Fine-tune small transformer on full AG News, expand baseline keywords, explore ensemble methods, or deploy lightweight API.

## 7 Reproducibility & Use

- **Notebook:** `notebooks/pipeline_demo.ipynb` reproduces all steps.
- **Requirements:** `requirements.txt` includes necessary dependencies.
- **Artifacts:** Trained model and embeddings can be saved under `artifacts/` if saving steps are run.

## 8 References

- AG News dataset (Hugging Face)
- `sentence-transformers/all-MiniLM-L6-v2`
- scikit-learn documentation