Ruben Valenzuela

Miguel Mora  
Professor Anna Devarakonda

Artificial Intel Applications

28 July 2025

**Reflection Journal – NewsBot Intelligent Classifier**

**Project Overview**

For this midterm project, We developed NewsBot, an intelligent news classification system powered by Python and Natural Language Processing (NLP). The core goal was to classify BBC news articles into predefined categories (such as politics, entertainment, tech, etc.) using classical machine learning techniques. The implementation was entirely done in a Google Colab notebook and the entire project was versioned and documented via GitHub.

**Phase 1: Data Preprocessing and Cleaning**

In the initial stage, we loaded the BBC News dataset and began by inspecting its structure. I removed any missing values and renamed the columns to id, content, and category for clarity. Then, I performed an analysis of the category distribution to ensure data was balanced enough for model training. Saving the cleaned dataset marked the successful end of this phase.

**Phase 2: Text Processing and Feature Engineering**

This was one of the most important phases. I used TfidfVectorizer to convert text into numerical vectors. we experimented with unigrams and bigrams, removing stopwords and punctuation for noise reduction. This vectorization step turned raw article content into structured input for machine learning.

**Phase 3: Model Building and Evaluation**

For classification, we used the Multinomial Naive Bayes model from scikit-learn, which is especially suited for text-based categorization tasks. we split the dataset into training and testing sets, trained the model, and evaluated it using accuracy, precision, recall, and F1-score metrics. The model achieved an accuracy of 97%, which was consistent with the performance reported in similar academic projects.

**Phase 4: Interface and User Interaction**

To simulate a user interface, I implemented a basic Python function (newsbot()) that prompts the user to enter a news headline or short article and then returns the predicted category. Although simple, this interface provides a glimpse into how real-world newsbots can offer real-time classification.

**Technologies Used**

- Python for scripting and development  
- pandas, NumPy, matplotlib for data handling and visualization  
- scikit-learn for vectorization and machine learning  
- Google Colab as the primary development environment  
- GitHub for version control and submission

**Key Takeaways**

- We deepened my understanding of NLP pipelines, especially the power of TF-IDF in representing text data effectively.  
- We learned how essential preprocessing is in determining model success, especially for unstructured data like text.  
- This project gave us hands-on experience with classification metrics and model evaluation beyond simple accuracy.  
- We also improved my documentation and version control skills by maintaining a clean GitHub repository, complete with a detailed README.

**Challenges and Solutions**

- Challenge: Some categories (like 'entertainment') had a much smaller number of articles.  
 Solution: We ensured stratified sampling during train-test split to maintain category proportions.

- Challenge: Balancing the vectorization settings (e.g., n-grams, min\_df).  
 Solution: we experimented with different configurations and chose the ones that led to the highest F1-score.