Project Report: NLP-Based Classification Systems

Student Name: **Gungun Sharma**

Roll No.: **2301201088**

Course: **BCA [AI & DS]**

Submitted to: **Sahil Sir.**

1. Introduction

This report documents two Natural Language Processing (NLP) projects developed as part of my academic curriculum. The projects demonstrate practical applications of machine learning for text classification tasks.

1.1 Project Overview

- Project 1: BBC News Article Classification
- Project 2: Movie Review Sentiment Analysis
- Technology Stack: Python, Scikit-learn, NLTK, Pandas, Matplotlib

1.2 Objectives

- Implement text preprocessing techniques
- Apply machine learning algorithms for classification
- Evaluate model performance using standard metrics
- Gain hands-on experience with NLP pipelines

2. Project 1: BBC News Classification

2.1 Problem Statement

Classify BBC news articles into predefined categories (Sports, Politics, Technology, Business, Entertainment) based on their content.

2.2 Dataset

- Source: BBC News Dataset
- Size: 2,225 articles
- Categories: 5 distinct classes
- Format: Text documents with category labels

2.3 Methodology

2.3.1 Data Preprocessing

- Text cleaning and normalization
- Stop word removal using NLTK
- Lemmatization for word standardization
- Tokenization of text into features

2.3.2 Feature Extraction

- Bag of Words (BoW): Word frequency vectors
- TF-IDF: Term frequency-inverse document frequency
- Feature size limited to 5,000 most frequent words

2.3.3 Models Implemented

1. Logistic Regression
2. Support Vector Machine (SVM)
3. Performance comparison between BoW and TF-IDF

2.4 Results and Analysis

Model	Vectorizer	Accuracy	Precision	Recall
Logistic Regression	TF-IDF	0.92	0.91	0.90
SVM	TF-IDF	0.89	0.88	0.87
Logistic Regression	BoW	0.85	0.84	0.83

Key Findings:

- TF-IDF outperformed Bag of Words approach
- Logistic Regression achieved highest accuracy (92%)
- Sports category showed highest classification accuracy
- Politics and Business categories had some misclassification

3. Project 2: Movie Review Sentiment Analysis

3.1 Problem Statement

Classify IMDb movie reviews as positive or negative sentiment using text analysis techniques.

3.2 Dataset

- Source: IMDb Movie Reviews Dataset
- Size: 50,000 reviews (25k positive, 25k negative)
- Balance: Equal distribution of sentiments
- Format: Review text with sentiment labels

3.3 Methodology

3.3.1 Text Preprocessing

- HTML tag removal
- Special character cleaning
- Lowercase conversion
- Stop word elimination
- Lemmatization using WordNet

3.3.2 Feature Engineering

- TF-IDF Vectorization
- N-gram range: (1,2) for unigrams and bigrams
- Maximum features: 5,000

- Minimum document frequency: 5

3.3.3 Classification Algorithms

1. Multinomial Naive Bayes
2. Logistic Regression
3. Performance evaluation using multiple metrics

3.4 Results and Discussion

Model	Accuracy	Precision	Recall	F1-Score
Logistic Regression	0.88	0.87	0.88	0.87
Naive Bayes	0.85	0.84	0.85	0.84

Analysis:

- Logistic Regression performed better than Naive Bayes
- Model achieved 88% accuracy on test data
- Precision and recall balanced across both classes
- Confusion matrix showed good separation between sentiments

4. Implementation Challenges

4.1 Technical Challenges

1. Memory Management: Large datasets required efficient processing
2. Feature Selection: Choosing optimal number of features
3. Hyperparameter Tuning: Finding best model parameters
4. Text Cleaning: Handling special characters and HTML tags

4.2 Solutions Implemented

- Used efficient data structures (sparse matrices)
- Implemented incremental processing for large files
- Applied grid search for parameter optimization
- Developed custom text cleaning functions

5. Learning Outcomes

5.1 Technical Skills Gained

- Hands-on experience with NLP libraries (NLTK, Scikit-learn)
- Understanding of text preprocessing pipelines
- Experience with machine learning model evaluation
- Proficiency in data visualization for results presentation

5.2 Conceptual Understanding

- Difference between BoW and TF-IDF approaches
- Importance of data preprocessing in NLP
- Model selection criteria for text classification
- Interpretation of classification metrics

6. Conclusion and Future Work

6.1 Project Summary

Both projects successfully demonstrated practical applications of NLP techniques. The BBC news classifier achieved 92% accuracy, while the sentiment analyzer reached 88% accuracy, both meeting acceptable performance standards for academic projects.

6.2 Future Enhancements

1. Advanced Models: Experiment with neural networks and transformers
2. Feature Engineering: Incorporate word embeddings (Word2Vec, GloVe)
3. Real-time Deployment: Create web interfaces for practical use
4. Multi-language Support: Extend to other languages

6.3 Personal Reflection

These projects provided valuable insights into the complete machine learning pipeline from data collection to model deployment. The hands-on experience strengthened my understanding of both theoretical concepts and practical implementation challenges in NLP.

References:

1. Pedregosa et al., "Scikit-learn: Machine Learning in Python", JMLR 12, pp. 2825-2830, 2011

2. Bird, Steven, "Natural Language Processing with Python", O'Reilly Media, 2009

3. BBC News Dataset Documentation

4. IMDb Dataset Reference Materials

Appendix:

- GitHub Repository: https://github.com/gungun2005/NLP-Project
- Code Files: Two Jupyter notebooks (.ipynb)
- Documentation: README.md with setup instructions