NLP Mini Project Report

Sentiment Analysis Using Logistic Regression

1. Introduction

Natural Language Processing (NLP) has become one of the most impactful areas of artificial intelligence, powering applications such as chatbots, recommendation systems, customer feedback analysis, and opinion mining. Sentiment analysis, in particular, helps organizations understand how users feel about products, services, or events based on textual data. In this mini project, an end-to-end NLP pipeline is developed using Python to classify text as positive or negative. The project demonstrates core NLP stages such as preprocessing, feature representation, model training, and evaluation, providing a practical understanding of how real-world NLP systems operate.

2. Problem Statement

The primary objective of this project is to classify textual sentences into two sentiment categories: positive and negative. In modern digital spaces, users express opinions frequently through reviews, comments, and messages. Manually reading and analyzing such huge volumes of data is not feasible. Therefore, building an automated sentiment classifier helps in extracting meaningful insights from raw text. This project aims to create a simple yet effective sentiment analysis model that can identify emotional polarity based on linguistic patterns.

3. Dataset Description

For demonstration purposes, a small custom dataset consisting of 8 manually constructed sentences is used. Each sentence reflects a distinct sentiment and is labeled accordingly as positive (1) or negative (0). Though the dataset is minimal, it adequately showcases the working of an NLP pipeline and helps in explaining each component clearly. In real applications, datasets may include thousands of examples collected from platforms such as Twitter, Amazon product reviews, or customer support logs.

4. NLP Techniques Used

Several foundational NLP techniques were applied in this project, including:

- **Tokenization:** Breaking sentences into individual words for processing.
- **Stopword Removal:** Eliminating commonly used words that do not contribute significantly to meaning.
- **Lemmatization:** Reducing words to their base or dictionary form to standardize vocabulary.
- **TF-IDF Vectorization:** Converting textual data into numerical feature vectors that capture word importance.
- **Logistic Regression:** A simple and powerful classification algorithm used to distinguish between classes.

5. System Workflow

The complete NLP pipeline followed in this project includes several crucial stages that transform raw text into meaningful predictions:

1. **Data Loading** – Importing training samples into the environment.
2. **Preprocessing** – Cleaning and standardizing text using tokenization, stopword removal, and lemmatization.
3. **Vectorization** – Converting clean text into TF-IDF numerical vectors.
4. **Train-Test Split** – Separating data for unbiased evaluation.

5. **Model Training** – Training Logistic Regression on processed data.
6. **Evaluation** – Measuring accuracy, generating classification reports, and analyzing confusion matrices.

6. Model and Implementation

Logistic Regression is chosen for this task because it works exceptionally well for binary classification problems like sentiment analysis. It is computationally efficient, easy to interpret, and often performs adequately even on small datasets. Using TF-IDF vectors as input, the model learns the statistical patterns associated with positive and negative sentiments. Once trained, the model can predict the sentiment of unseen text with significant accuracy.

7. Evaluation

The model is evaluated using performance metrics such as accuracy, precision, recall, and F1-score. These metrics provide quantitative insight into how well the classifier identifies sentiment polarity. Additionally, a confusion matrix is generated to visualize prediction errors and understand which classes the model predicts accurately versus where it struggles.

8. Results

Despite the limited dataset size, the model achieved a commendable level of performance. It correctly identified most sentiment labels and demonstrated the effectiveness of TF-IDF combined with Logistic Regression. These results showcase how even simple NLP models can capture meaningful linguistic patterns when properly preprocessed.

9. Strengths and Limitations

Strengths:
• The model is simple, fast, and easy to implement.
• Works efficiently on small datasets.
• Interpretable and straightforward to analyze.

Limitations:
• The dataset is extremely small, limiting real-world applicability.
• Logistic Regression may not capture complex semantic relationships.
• Advanced models such as transformers would outperform significantly.

10. Future Improvements

To enhance the performance and make the system more robust, several upgrades can be implemented:
• Using large-scale datasets from real platforms.
• Implementing deep learning models such as LSTMs, GRUs, or transformers like BERT.
• Performing hyperparameter optimization.
• Deploying the model using a web interface or API.

11. Conclusion

This expanded report presents a detailed and comprehensive overview of building an NLP sentiment analysis system. The project successfully demonstrates the essential components of an NLP pipeline, from text preprocessing to model evaluation. It provides a strong foundation for future advancements in NLP, enabling deeper exploration into modern, large-scale language models and real-world applications.