SYNOPSIS

Report on

<< Sentiment Analysis of Movie Reviews>>

by

Aman Nayak - 202410116100021

Alok Kumar Singh – 202410116100019

Ansh Raj - 202410116100031

Session:2024-2025 (II Semester)

Under the Supervision of

Mr. Apoorv Jain

Assistant Professor

KIET Group of Institutions, Delhi-NCR, Ghaziabad

Department Of Computer Applications

KIET GROUP OF INSTITUTIONS, DELHI-NCR, GHAZIABAD-201206

ABSTRACT

In the digital era, online reviews play a crucial role in shaping consumer decisions, especially in the entertainment industry. The proliferation of user-generated movie reviews on platforms such as IMDb, Rotten Tomatoes, and social media has made sentiment analysis an essential tool for understanding public opinion. This project, titled *"Sentiment Analysis of Movie Reviews,"* leverages Natural Language Processing (NLP) and Machine Learning (ML) techniques to automatically classify movie reviews as either *positive* or *negative*, thus providing a streamlined approach to gauging audience sentiment.

The project is implemented using Python with the Flask web framework to offer an intuitive user interface. Users can submit a movie review through a web form, which is then processed using a pre-trained machine learning model. The core components of the backend include a CountVectorizer for converting text into numerical features and a Multinomial Naive Bayes classifier trained on a dataset of labeled movie reviews. These components are serialized using Python's pickle module to ensure efficient loading during prediction.

The system architecture follows a clean separation of concerns: data preprocessing and model training are handled offline, while the Flask application is responsible for serving predictions in real-time. Upon receiving a user input, the review text is vectorized and passed to the classifier, which returns the sentiment label. The output is then displayed back to the user in a user-friendly format.

This application showcases the power of machine learning in text classification and can be further enhanced with advanced NLP techniques such as TF-IDF, word embeddings, or deep learning models. Overall, the project demonstrates a practical and scalable approach to sentiment analysis, with real-world applicability in review aggregation, feedback analysis, and recommendation systems.

TABLE OF CONTENTS

Title: Sentiment Analysis of Movie Reviews

1. Introduction

With the rapid growth of the internet and social media, people now have numerous platforms to express their opinions about movies, products, and services. Movie reviews, in particular, have become a significant source of information for potential viewers. However, manually reading through thousands of reviews to understand general public sentiment can be time-consuming and inefficient. This challenge has led to the emergence of **Sentiment Analysis**, a subfield of Natural Language Processing (NLP), which aims to automatically identify and extract opinions or emotions from text.

This project focuses on building a **Sentiment Analysis system for movie reviews** that can classify a given review as either *positive* or *negative*. It uses machine learning techniques to train a model on a labeled dataset of movie reviews. The primary objective is to provide a tool that helps users, platforms, or analysts to quickly determine the sentiment behind a review without human intervention.

The project is developed using Python and leverages libraries such as **scikit-learn** for model building and **Flask** for deploying the application as a web service. The system uses a **CountVectorizer** to convert textual data into numerical format, which is then fed into a **Multinomial Naive Bayes** classifier. Once trained, both the model and the vectorizer are saved using the pickle module to be reused during the prediction phase in the web application.

Through this project, users can input a movie review in a web interface, and the application will instantly display whether the sentiment expressed is positive or negative. The simplicity and efficiency of the system demonstrate the power of combining NLP with machine learning for real-world applications. This project not only serves as a foundational example of sentiment analysis but also paves the way for further enhancements using advanced NLP methods and deep learning models.

2. Literature Review

Sentiment analysis, also known as opinion mining, has become a significant area of research in Natural Language Processing (NLP) due to the growing importance of understanding public opinion from unstructured text data. Numerous studies have explored various approaches to sentiment classification, particularly in the context of movie reviews, which are rich sources of subjective expressions.

Pang et al. (2002) were among the pioneers in this field, introducing machine learning techniques such as Naive Bayes, Support Vector Machines (SVM), and Maximum Entropy for sentiment classification of movie reviews. Their work highlighted the effectiveness of supervised learning methods in achieving high accuracy in sentiment prediction. Later research has extended this by integrating feature selection, n-gram models, and ensemble techniques to improve performance.

The use of vectorization techniques like CountVectorizer and TF-IDF (Term Frequency-Inverse Document Frequency) has been common for converting text into numerical features suitable for machine learning models. In recent years, more advanced representations such as word embeddings (e.g., Word2Vec, GloVe) and contextual embeddings (e.g., BERT) have been introduced, offering improved understanding of context and semantics.

Several researchers have explored deep learning models like Recurrent Neural Networks (RNNs), Long Short-Term Memory (LSTM) networks, and Convolutional Neural Networks (CNNs) for capturing the sequential and hierarchical structure of language. However, traditional machine learning models like Multinomial Naive Bayes still remain popular due to their simplicity, speed, and surprisingly strong performance on text classification tasks.

Various datasets such as IMDb reviews, Rotten Tomatoes, and custom crawled datasets have been utilized to train and evaluate sentiment models. These studies underline the importance of preprocessing techniques like tokenization, stop-word removal, and stemming or lemmatization.

3. Project / Research Objective

The project titled *"Sentiment Analysis of Movie Reviews"* aims to classify user-submitted movie reviews into positive or negative categories using machine learning and natural language processing techniques. This review highlights the key methodologies, tools, and outcomes of the project while situating it within existing research practices.

The project begins with the collection and preprocessing of a labeled dataset containing movie reviews. Preprocessing includes tasks such as tokenization, removal of stop words, and conversion of text into numerical data using **CountVectorizer**. This technique transforms raw text into a matrix of token counts, enabling the machine learning algorithm to process textual data.

A **Multinomial Naive Bayes** classifier was chosen for its efficiency and effectiveness in text classification problems, especially for sentiment analysis. After training the model on the dataset, it is serialized using the **pickle** module for later use in real-time predictions. A user-friendly web interface was developed using the **Flask** framework, where users can input reviews and receive immediate sentiment classification results.

Compared to advanced models like LSTM or BERT, this project employs a simpler yet highly effective approach suitable for lightweight and quick deployment. While it may not capture deep contextual relationships in language like transformer-based models, its fast inference time and high accuracy on standard datasets make it ideal for small to medium-scale applications.

This project demonstrates the practical application of NLP and machine learning in analyzing public sentiment and highlights how even traditional models can yield valuable results when applied thoughtfully. It lays a strong foundation for future enhancements, such as integrating more complex models or expanding to multilingual sentiment analysis.

4. Hardware and Software Requirements

To successfully implement and run the *Sentiment Analysis of Movie Reviews* project, certain hardware and software configurations are necessary. This section outlines both the minimum and recommended specifications to ensure smooth development, model training, and deployment of the application.

Hardware Requirements:

Processor (CPU):

- **Minimum: Intel Core i3 (6th Generation) or equivalent**
- **Recommended: Intel Core i5 or i7 / AMD Ryzen 5 or above**
  A decent multi-core processor is required for training the machine learning model, especially when dealing with large datasets and vectorizing textual data.

2. RAM:

- **Minimum: 4 GB**
- **Recommended: 8 GB or higher**
  Higher RAM enables better handling of large datasets, model training, and running multiple services like Flask, Python scripts, and a browser simultaneously.

3. Storage:

- **Minimum: 10 GB free disk space**
- **Recommended: SSD with 20 GB or more**
  Storing datasets, trained models (pickle files), temporary files, and system dependencies requires adequate disk space.

4. Display:

- **Any display capable of supporting a standard resolution (1366x768 or higher) for web interface testing.**

5. Internet Connection:

- **Required for downloading packages, libraries, and datasets.**

Software Requirements:

1. Operating System:

- Preferred: Windows 10/11, Linux (Ubuntu 18.04 or higher), macOS
  The project is platform-independent, but Linux or macOS is generally more efficient for Python development environments.

2. Programming Language:

- Python 3.7 or higher
  Python is used for both backend development and model training due to its rich ecosystem of libraries for machine learning and web development.

3. Libraries and Frameworks:

- Flask: For building and deploying the web application interface
- scikit-learn: For machine learning model training and evaluation
- pandas: For data manipulation and preprocessing
- pickle: For model serialization and deserialization
- numpy: For numerical operations and handling arrays
- sklearn.feature_extraction.text.CountVectorizer: For text vectorization

Install these using pip:

bash

CopyEdit

pip install flask scikit-learn pandas numpy

4. Code Editor/IDE:

- VS Code / PyCharm / Jupyter Notebook

  A modern IDE or code editor is recommended for writing and debugging Python code effectively.

5. Browser:

- Chrome, Firefox, Edge, or any modern browser

  Used to access and test the Flask web interface locally or on a server.

5. Project Flow/ Research Methodology

The *Sentiment Analysis of Movie Reviews* project follows a systematic methodology, starting from data collection to deployment, to build an efficient and accurate sentiment classification system using machine learning and natural language processing (NLP).

The process begins with problem identification, where the goal is defined: to classify movie reviews as *positive* or *negative*. This is followed by data collection, where a dataset containing user reviews and corresponding sentiment labels is gathered from reliable sources such as IMDb or Kaggle.

Next, the data preprocessing phase is carried out. This includes cleaning the text data by converting it to lowercase, removing punctuation and stop words, and then tokenizing it into individual words. The cleaned text is then converted into a numerical format using the CountVectorizer, which transforms text into a bag-of-words representation suitable for machine learning.

In the model training phase, a Multinomial Naive Bayes classifier is used due to its simplicity and effectiveness in text-based tasks. The dataset is split into training and testing sets to evaluate model performance using metrics such as accuracy and F1-score.

Once the model is trained and evaluated, it is serialized using pickle along with the vectorizer to allow real-time usage without retraining. A Flask-based web application is then developed to allow users to input their reviews through a user-friendly interface. The backend loads the trained model and returns the predicted sentiment instantly.

Finally, the system undergoes testing and validation to ensure reliability and accuracy. The project is then ready for deployment on local or cloud servers. This structured approach ensures a functional and scalable sentiment analysis solution that can be extended in the future.

Project / Research Outcome

The *Sentiment Analysis of Movie Reviews* project successfully demonstrates the practical application of machine learning and natural language processing techniques to classify user-generated reviews as either *positive* or *negative*. The outcome of this project reflects both technical achievement and real-world usability.

The primary outcome is the development of a **functional and user-friendly web application** that allows users to input any movie review and receive immediate sentiment feedback. This is achieved using a trained **Multinomial Naive Bayes** model and **CountVectorizer**, which together provide efficient and accurate text classification. The model has shown high accuracy on the test dataset, indicating that even simple algorithms, when properly trained, can perform well on real-world NLP tasks.

The use of Python, Flask, and scikit-learn ensures that the system is lightweight and easily deployable, making it accessible on standard computing environments without the need for high-end hardware or advanced configurations. Additionally, the modular structure of the project—separating model training, data processing, and application logic—makes the system highly maintainable and extendable for future improvements.

From a research perspective, the project validates the effectiveness of traditional NLP techniques for sentiment analysis and sets a strong foundation for exploring more advanced methods such as TF-IDF, deep learning, or transformer-based models like BERT. It also opens up the possibility for expanding the model to include neutral sentiments, multilingual analysis, and integration with social media platforms.

Overall, the project has met its objective of building a reliable sentiment analysis tool and has provided valuable insights into the workflow of developing machine learning applications from data collection to deployment.

7. Proposed Time Duration

The development of the *Sentiment Analysis of Movie Reviews* project is planned to be completed over a span of **10 weeks**, divided into structured phases to ensure smooth progress, timely completion, and efficient resource utilization. Each phase focuses on a specific part of the project, from initial planning to final deployment.

Week 1 – 2: Project Planning and Research

- Understanding project objectives.
- Conducting background research on sentiment analysis and related technologies.
- Finalizing the dataset source and tools required (Python, Flask, scikit-learn, etc.).

Week 3 – 4: Data Collection and Preprocessing

- Acquiring the dataset (movie reviews and sentiment labels).
- Cleaning and preprocessing the data: removing noise, tokenization, stop word removal.
- Converting text to numerical format using CountVectorizer.

Week 5 – 6: Model Development and Evaluation

- Training the sentiment classification model using Multinomial Naive Bayes.
- Splitting data into training and testing sets.
- Evaluating model performance using accuracy, precision, recall, and F1-score.

Week 7: Model Saving and Integration

- Saving the trained model and vectorizer using the pickle module.
- Preparing them for real-time usage in the web application.

Week 8: Web Application Development

- Developing the front-end using HTML and Flask.

- Integrating the backend with the trained model for sentiment prediction.

Week 9: Testing and Debugging

- Performing unit testing and functional testing.
- Ensuring accurate predictions and a responsive user interface.

Week 10: Final Report and Deployment

- Documenting the project.
- Preparing the final report and presentation.
- Deploying the application locally or on a cloud platform.

This timeline provides a clear roadmap for systematic project execution while allowing flexibility for iteration and improvement at each stage.

References/ Bibliography.

- Pang, B., Lee, L., & Vaithyanathan, S. (2002). *Thumbs up? Sentiment classification using machine learning techniques*. Proceedings of the ACL-02 Conference on Empirical Methods in Natural Language Processing.
- Bird, S., Klein, E., & Loper, E. (2009). *Natural Language Processing with Python*. O'Reilly Media.
- Jurafsky, D., & Martin, J. H. (2020). *Speech and Language Processing* (3rd ed.). Pearson Education.
- Scikit-learn: Machine Learning in Python. Retrieved from: https://scikit-learn.org
- Flask Documentation – The Pallets Projects. Retrieved from: https://flask.palletsprojects.com
- IMDb Movie Review Dataset. Retrieved from: https://ai.stanford.edu/~amaas/data/sentiment/
- Aggarwal, C. C. (2018). *Machine Learning for Text*. Springer.
- Gensim and Text Processing Techniques. Retrieved from: https://radimrehurek.com/gensim/
- Kowsari, K., Meimandi, K. J., Heidarysafa, M., Mendu, S., Barnes, L. E., & Brown, D. E. (2019). *Text Classification Algorithms: A Survey*. Information (MDPI), 10(4), 150