



MANIPAL INSTITUTE OF TECHNOLOGY
BENGALURU
(A constituent unit of MAHE, Manipal)

MOVIE REVIEW SENTIMENT ANALYSIS

ARTIFICIAL INTELLIGENCE LAB – MINI PROJECT PROPOSAL

Under guidance of

Dr. Mangalampalli Surya Sudheer

B.Tech, M.Tech, Ph.D.

Associate Professor, School of Computer Engineering

Manipal Institute of Technology – Bengaluru

MANIPAL ACADEMY OF HIGHER EDUCATION

By

Raghav Mishra

(235890240)

CSE (AI - B)

Shankaraditya Kompella

(235890106)

CSE (AI - B)



School of Computer Engineering
Artificial Intelligence Lab (CSE_3182)
Mini Project Proposal

Movie Review Sentiment Analysis

Raghav Mishra
(235890240)
CSE (AI - B)

Shankaraditya Kompella
(235890106)
CSE (AI - B)

ABSTRACT - Movie reviews play a crucial role in shaping audience opinions and box office success. However, manually analyzing a large number of reviews is time consuming and subjective. This project aims to develop a web-based sentiment analysis system that classifies movie reviews as positive or negative using Natural Language Processing (NLP) techniques. The application integrates a simple web interface, a Flask backend, and an NLP based model (VADER or pre trained dataset) to provide real time sentiment predictions. The project demonstrates how Artificial Intelligence can automate opinion mining and enhance user experience in understanding public sentiment.

Movie reviews strongly influence audience decisions, but reading through many reviews is time consuming and subjective. Automated sentiment analysis using NLP can provide quick insights into public opinion and make the process more efficient.

B. Objectives:

- To build a web based tool that classifies movie reviews as *positive* or *negative*.
- To apply Natural Language Processing (NLP) techniques for sentiment detection.
- To demonstrate integration of frontend, backend, and database in a simple AI project.

1. Introduction

With the growth of online platforms like IMDb, Rotten Tomatoes, and social media, users share thousands of movie reviews daily. Reading through these manually is inefficient. Sentiment Analysis, an application of NLP helps automatically identify whether a review expresses a positive or negative opinion.

This project leverages Python's Flask framework to create a web based interface that accepts a movie review as input, processes it through a sentiment analysis model, and displays the sentiment result instantly. The project also demonstrates how AI, NLP, and web development can work together in a practical mini application.

A. Background / Motivation:

2. Literature Survey

Numerous studies have focused on sentiment analysis using machine learning and deep learning techniques.

- **B. Pang and L. Lee (2008)** explored sentiment classification using Naive Bayes and SVM on movie review datasets.
- **Hutto & Gilbert (2014)** introduced the **VADER (Valence Aware Dictionary and sentiment Reasoner)** model, optimized for social media and review text.
- **Recent works** use pre trained transformer models like **BERT**, which achieve high accuracy but require significant computational power.

In contrast, this project focuses on simplicity and interpretability, using lightweight NLP tools suitable for small scale deployment.

3. Methodology

The proposed system consists of three main components: **A.**

Frontend:

- Developed using **HTML, CSS, and JavaScript**.
- Provides a simple interface for users to input movie reviews and view results. **B. Backend:**
- Built using **Flask (Python)**.
- Receives user input and processes it through an NLP model.

C. Sentiment Analysis Model:

- Utilizes **NLTK's VADER Sentiment Analyzer** or a small pre trained model.
- Calculates sentiment scores and classifies the review as *positive* or *negative*. **D. Database (Optional):**
- **SQLite/MySQL** used to store user reviews and their corresponding sentiment results for analysis.

→ Flow of Operation:

User → Enters Review → Flask Backend → NLP Sentiment Model → Result Displayed → (Optional) Stored in Database

4. Results

- The system successfully classifies input reviews into *positive* or *negative* sentiments.
- It provides near instant results with a user friendly interface.
- Accuracy (using VADER) ranges between **80–85%** for typical movie review data.
- The project demonstrates complete integration of frontend, backend, and AI components.

Add some insights we noticed:

- Short reviews like “OK nice” are often correctly classified as positive.

- Reviews with negations like “It’s not a nice movie” need preprocessing (handling “not”) for accurate classification.
- Extreme words like “disaster” or “wonderful” strongly influence sentiment prediction.
-

Project output after running the code :

Review (Excerpt)	Sentiment
One of the other reviewers has mentioned that this is a wonderful film...	Positive
A wonderful little production. The story and acting were both excellent.	Positive
I thought this was a wonderful way to spend time; the performances were great.	Positive
Basically there's a family where a little boy sees dead people...	Negative
Loved the cinematography and background music.	Positive

Dataset after output encoding :

Dataset after output encoding :	Sentiment
One of the other reviewers has mentioned that this is a wonderful film...	1
A wonderful little production. The story and acting were both excellent.	1
I thought this was a wonderful way to spend time; the performances were great.	1
Basically there's a family where a little boy sees dead people...	0
Loved the cinematography and background music.	1
A good attempt, but the storyline was weak.	0
It was a complete waste of time and money	0

Example of Text Preprocessing:

Stage	Sample Review
Original Review	I thought this was a wonderful way to spend time; the performances were outstanding and the direction was superb.
After Stopword Removal	thought wonderful way spend time performances outstanding direction superb
After Stemming	thought', 'wonder', 'way', 'spend', 'time', 'perform', 'outstand', 'direct', 'superb'.

Sentiment Classification:

```
reviews_list=[ "It's a nice movie","It's not a nice movie","the Movie is more violent", "In the movie, the hreo doesn't performed well","OK, the movie is well","OK nice","it's a disaster" ]

for review in reviews_list:
    words = word_tokenize(review)
    words = create_word_features(words)
    print(review,' ('+classifier.classify(words)+')')
```

Result from the code:

```
It's a nice movie      (positive)
It's not a nice movie (positive)
the Movie is more violent (positive)
In the movie, the hreo doesn't performed well (positive)
OK, the movie is well (positive)
OK nice (positive)
it's a disaster (negative)
```

Models accuracy:

```
Gaussian accuracy    = 74.0 %  
Multinomial accuracy = 84.0 %  
Bernoulli accuracy   = 85.0 %
```

5. Conclusion

This project successfully demonstrates how NLP can automate the process of analysing public opinion on movies. The implementation of a Flask based web app with integrated sentiment analysis offers a practical example of AI deployment. Future improvements could include using more advanced models (like BERT) and adding neutral sentiment detection or visualization dashboards for trend analysis.

6. References

- [1]. B. Pang and L. Lee, *Opinion Mining and Sentiment Analysis*, Foundations and Trends in Information Retrieval, 2008.
- [2]. C.J. Hutto and E. Gilbert, *VADER: A Parsimonious Rule Based Model for Sentiment Analysis of Social Media Text*, ICWSM, 2014.
- [3]. NLTK Documentation: <https://www.nltk.org>
- [4]. Flask Documentation: <https://flask.palletsprojects.com>