



AI-Powered Movie Review Sentiment Analysis

Final Report

MASTERS OF COMPUTER APPLICATION

SUBMITTED BY:

NAME	SAP ID
Akash Panwar	500106837
Omji Shukla	500100963
Tanishka Chandola	500104885
Abhishek Kumar	500106851
Deepesh Singh	500106971

GUIDED BY: Dr. Kavitha

CANDIDATE'S DECLARATION

We thus attest that the project work titled “**AI-Powered Movie Review Sentiment Analysis**” which was turned in to the AI Cluster at the School of Computer Science, University of Petroleum & Energy Studies, Dehradun, in partial fulfillment of the requirements for the award of the Degree of MASTER OF COMPUTER APPLICATIONS with a specialization in Artificial Intelligence and Machine Learning, is an authentic record of our work completed between January 2024 and MAY 2024 under the direction **Dr. Kavitha**, School of Computer Science.

The matter presented in this project has not been submitted by us for the award of any other degree of this or any other University.

Akash Panwar

Tanishka Chandola

Abhishek Kumar


Omji Shukla

Deepesh Singh

This is to certify that the above statement made by the candidate is correct to the best of my knowledge.

Date: 9 May 2024

Dr. Kavitha



ACKNOWLEDGEMENT

We would like to sincerely thank our guide for all of her guidance, inspiration, and unwavering assistance during this effort. He provided invaluable support and suggestions without which our task would not have been possible.

We also thank Prof. Dhiviya Rose and Prof. Manish Pandey, our course and activity coordinator, for her prompt information and support during the project's completion.

We would like to express our gratitude to all of our friends for their support and helpful advice throughout this effort. Lastly, we can never enough thank our parents for showing us this world and for all of their support.

ABSTRACT

Sentiment analysis, a subset of natural language processing, is critical in assessing and comprehending user views conveyed in textual data. This research focuses on creating a sentiment analysis model that is particularly designed for movie reviews. The primary goal is to automatically identify movie reviews as good or negative based on the mood expressed in the text. To do this, the project requires preprocessing raw text data to eliminate noise and extraneous information, followed by feature extraction using the Bag-of-Words technique. Multiple Naive Bayes classifiers, such as Gaussian, Multinomial, and Bernoulli Naive Bayes, are then trained and tested to categorize movie reviews. The performance of each classifier is measured using conventional metrics such as accuracy_score.

TABLE OF CONTENTS

S.No.	Contents	Page No
1.	Introduction	1 - 2
1.1.	History	1
1.2.	Requirement Analysis	1
1.3.	Main Objective	1
1.4.	Sub Objectives	1
1.5.	Pert Chart Legend	2
2.	System Analysis	3
2.1.	Existing System	3
2.2.	Motivations	3
2.3.	Proposed System	3
2.4.	Modules	3
3.	Design and Diagrams	4
3.1	Sequence Diagram	4
3.2	Architecture	4
4.	Implementation	5 - 6
4.1.	Scenarios	5
4.2.	Algorithms	5 - 6
5.	Code and output	7 - 12
6.	Limitations and Future Enhancements	13
7.	Conclusion	14
	References	15

1. INTRODUCTION

1.1. History

Natural Language Processing (NLP) has a long history, extending back to the 1950s. NLP began as a rule-based method, but has expanded dramatically with the introduction of machine learning and deep learning techniques. In recent years, NLP has seen broad use in a variety of fields, including sentiment analysis, machine translation, and chatbots. experiences.

1.2. Requirement Analysis

Before beginning this sentiment analysis project, a thorough requirement study was undertaken to determine the necessity for automated sentiment analysis in the context of movie reviews. The investigation focused on the expanding number of user-generated information on movie review platforms, as well as the need for effective tools to analyze sentiment on a large scale.

1.3. Main Objective

The primary goal of this project is to create a sentiment analysis model that can automatically identify movie reviews as positive or negative based on the sentiment indicated in the text. The goal is to develop a tool that uses natural language processing techniques and machine learning algorithms to let consumers rapidly assess the overall sentiment of movie reviews.

1.4. Subobjectives

Preprocess the raw text input by removing noise and extraneous information, such as HTML elements and special characters.

The Bag-of-Words technique may be used to convert preprocessed text input into numerical feature vectors.

Train many Naive Bayes classifiers, including Gaussian, Multinomial, and Bernoulli Naive Bayes, to categorize movie reviews based on sentiment.

Use common metrics like accuracy_score to evaluate each classifier's performance.

Evaluate the performance of several Naive Bayes classifiers in the context of sentiment analysis for movie reviews.

1.5. Pert Chart Legend:

Stages of research	Week 1	Week 2	Week 3	Week 4	Week 5	Week 6	Week 7
Selection of topic							
Data collection from sources							
Literature review							
Research methodology plan							
Selection of the Appropriate Research Techniques							
Analysis & Interpretation of Data							
Findings and recommendations							
Final research project							

2. SYSTEM ANALYSIS

2.1. Existing system.

The current sentiment analysis methods for movie reviews rely heavily on machine learning algorithms and lexical analysis approaches. These algorithms frequently struggle to effectively capture subtle feelings stated in movie reviews, particularly in the presence of sarcasm, irony, and context-dependent language. Despite advances in machine learning, current algorithms may struggle to generalize across genres and cultural situations.

2.2. Motivations

The impetus for this research originates from the need for a more robust and reliable sentiment analysis system designed exclusively for film reviews. Existing algorithms may lack the sophistication to adequately handle the nuances of movie criticism, resulting in unsatisfactory sentiment categorization. We want to solve these shortcomings by constructing a more reliable sentiment analysis algorithm that will help people grasp the sentiment represented in movie reviews.

2.3. Proposed system

The proposed sentiment analysis system for movie reviews takes advantage of cutting-edge natural language processing techniques and machine learning algorithms to solve the drawbacks of existing systems. The suggested approach intends to better capture the intricacies of sentiment conveyed in movie reviews by incorporating pretreatment processes to clean and normalize text data, as well as feature extraction utilizing sophisticated vectorization algorithms. Furthermore, the use of numerous Naive Bayes classifiers enables a full evaluation of sentiment categorization performance.

2.4 Modules.

Data Preprocessing: In this module, special characters, HTML tags, and noise are removed from the raw text data. It also involves actions like tokenization, lowercasing, and stopword elimination.

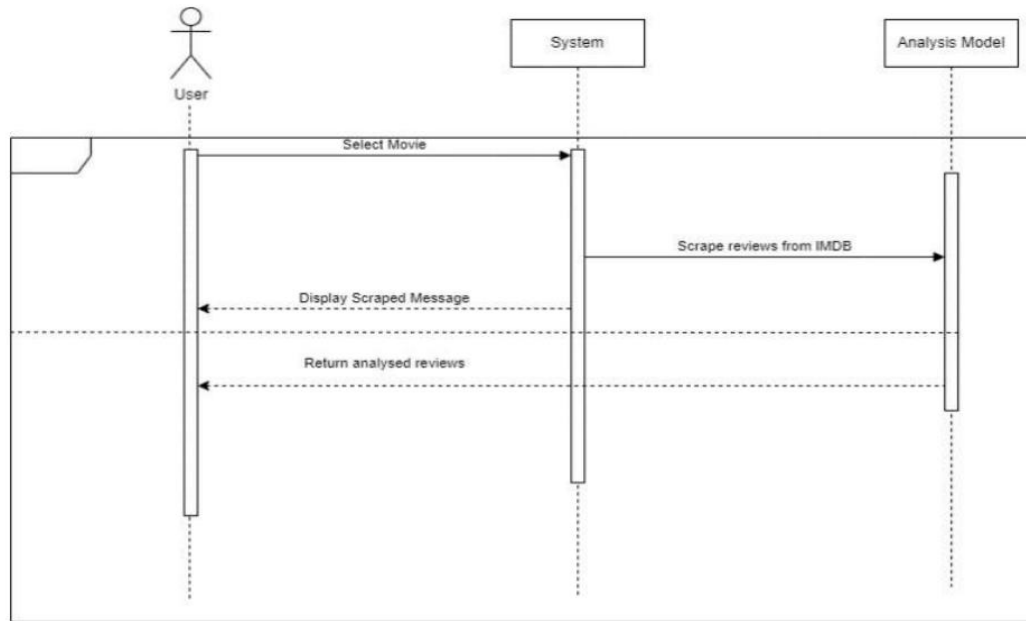
Feature extraction: The Bag-of-Words method is used in this module to transform the preprocessed text input into numerical feature vectors. This enables efficient processing and analysis of the textual material by the machine learning algorithms.

Multiple Gaussian, Multinomial, and Bernoulli Naive Bayes classifiers are among the Naive Bayes classifiers included in the system. Based on sentiment analysis, each classifier is trained using the feature vectors to categorize movie reviews.

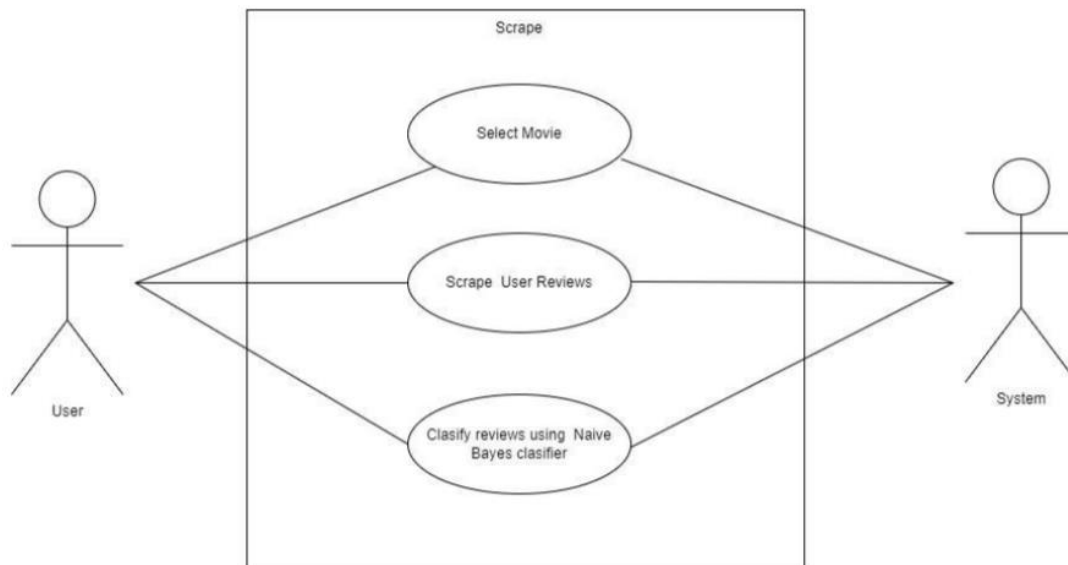
Evaluation: This module evaluates each classifier's performance using common measures like accuracy_score to provide light on how well the sentiment analysis system is working.

Design and Diagrams

Sequence Diagram



Use Case Diagram:



4. Implementaion

4.1 Scenarios.

Scenario 1: User Input: A user gives the system a review of a movie. Text conveying thoughts about the film, including those regarding the actors, story, or overall experience, may be included in the review.

Scenario 2: Preprocessing: To eliminate noise and unnecessary information, the system preprocesses the raw text input. This involves lowercasing, tokenization, and the removal of HTML elements, special characters, and stopwords.

Scenario 3: Feature Extraction: The system uses the Bag-of-Words technique to extract features from the text data following preprocessing. In order to do this, the preprocessed text must be transformed into numerical feature vectors that indicate whether a word is contained in the vocabulary or not.

Scenario 4: Model Training: Using feature vectors obtained from the training data, the system trains many Naive Bayes classifiers, such as Gaussian, Multinomial, and Bernoulli Naive Bayes. Every classifier gains the ability to categorize movie reviews as either favorable or bad according to the text's emotion.

Scenario 5: Model Evaluation: Using the testing data, each classifier's performance is assessed once it has been trained. The efficacy of the sentiment analysis models is evaluated using common assessment measures like accuracy_score.

Scenario 6: Implementation: The trained sentiment analysis models are put to work in the actual world after a successful evaluation. Users may add fresh reviews of movies, and the system will categorize the reviews based on their sentiments, helping with decision-making processes like movie suggestion or selection.

4.2 Algorithm

Gaussian Naive Bayes: This algorithm uses the Gaussian probability density function to determine the likelihood that a given feature belongs to a class, assuming that features follow a Gaussian distribution. It works well for features that are ongoing.

Multinomial Naive Bayes: This technique works well with features that reflect word counts or phrase frequencies and is frequently used for document classification problems. It uses the multinomial distribution to determine the probability of a feature given a class.

Bernoulli Naive Bayes: This method is commonly used for binary feature vectors and, in

contrast to Multinomial Naive Bayes, it simply takes into account the presence or absence of a feature. It makes use of the Bernoulli distribution to determine the likelihood that a characteristic will be present given a class.

These algorithms are used to categorize movie reviews according to sentiment; they are trained using preprocessed and feature-extracted data. The best algorithm for sentiment analysis in the context of movie reviews will be identified via testing and assessment.

5. Code and Output

Import Libraries

```
import numpy as np # linear algebra
import pandas as pd # data processing, CSV file I/O (e.g. pd.read_csv)
import re # for regex
import nltk
from nltk.corpus import stopwords
from nltk.tokenize import word_tokenize
from nltk.stem import SnowballStemmer
from sklearn.feature_extraction.text import CountVectorizer
from sklearn.model_selection import train_test_split
from sklearn.naive_bayes import GaussianNB, MultinomialNB, BernoulliNB
from sklearn.metrics import accuracy_score
import pickle
```

Read Data

```
data = pd.read_csv('/content/IMDB-Dataset.csv')
print(data.shape)
data.head()
```

	review	sentiment
0	One of the other reviewers has mentioned that ...	positive
1	A wonderful little production. The...	positive
2	I thought this was a wonderful way to spend ti...	positive
3	Basically there's a family where a little boy ...	negative
4	Petter Mattei's "Love in the Time of Money" is...	positive

```
data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 50000 entries, 0 to 49999
Data columns (total 2 columns):
#   Column      Non-Null Count  Dtype
---  -
0   review      50000 non-null  object
1   sentiment    50000 non-null  object
dtypes: object(2)
memory usage: 781.4+ KB
```

```
data.sentiment.value_counts()
```

```
sentiment
positive    25000
negative    25000
Name: count, dtype: int64
```

```
data.sentiment.replace('positive',1,inplace=True)
data.sentiment.replace('negative',0,inplace=True)
data.head()
```

	review	sentiment
0	One of the other reviewers has mentioned that ...	1
1	A wonderful little production. The...	1
2	I thought this was a wonderful way to spend ti...	1
3	Basically there's a family where a little boy ...	0
4	Petter Mattei's "Love in the Time of Money" is...	1

```
data.review[0]
```

```
'One of the other reviewers has mentioned that after watching just 1 Oz episode you'll be hooked. They are right, as this is exactly what happened with me.<br /><br />The first thing that struck me about Oz was its brutality and unflinching scenes of violence, which set in right from the word GO. Trust me, this is not a show for the faint hearted or timid. This show pulls no punches with regards to drugs, sex or violence. Its is hardcore, in the classic use of the word.<br /><br />It is called OZ as that is the nickname given to the Oswald Maximum Security State Penitentiary. It focuses mainly on Emerald City, an experimental section of the prison where all the cells have glass fronts and face inwards, so privacy is not high on the agenda. Emerald City is home to many Aryans, Muslims, gangstas, Latinos, Christians, Italians, Irish and more....so scuffles, death stares, dodgy dealings and shady agreements are never far away.<br /><br />I would say the main appeal of the show is due to the fact...'
```

```
def clean(text):  
    cleaned = re.compile(r'<.*?>')  
    return re.sub(cleaned,"",text)  
  
data.review = data.review.apply(clean)  
data.review[0]
```

```
def is_special(text):  
    rem = "  
    for i in text:  
        if i.isalnum():  
            rem = rem + i  
        else:  
            rem = rem + '  
    return rem  
  
data.review = data.review.apply(is_special)  
data.review[0]
```

```
def to_lower(text):  
    return text.lower()  
  
data.review = data.review.apply(to_lower)  
data.review[0]
```

```
nltk.download('stopwords')  
nltk.download('punkt')
```

```
def rem_stopwords(text):
    stop_words = set(stopwords.words('english'))
    words = word_tokenize(text)
    return [w for w in words if w not in stop_words]

data.review = data.review.apply(rem_stopwords)
data.review[0]
```

```
def stem_txt(text):
    ss = SnowballStemmer('english')
    return " ".join([ss.stem(w) for w in text])

data.review = data.review.apply(stem_txt)
data.review[0]
```

	review	sentiment
0	one review mention watch 1 oz episod hook righ...	1
1	wonder littl product film techniqu unassum old...	1
2	thought wonder way spend time hot summer weeke...	1
3	basic famili littl boy jake think zombi closet...	0
4	petter mattei love time money visual stun film...	1

```
X = np.array(data.iloc[:,0].values)
y = np.array(data.sentiment.values)
cv = CountVectorizer(max_features = 12500)
X = cv.fit_transform(data.review).toarray()
print("X.shape = ",X.shape)
print("y.shape = ",y.shape)
```

```
trainx,testx,trainy,testy = train_test_split(X,y,test_size=0.2,random_state=42,shuffle=True,
stratify=y)
print("Train shapes : X = {}, y = {}".format(trainx.shape,trainy.shape))
print("Test shapes : X = {}, y = {}".format(testx.shape,testy.shape))
```

```
gnb= GaussianNB()  
gnb.fit(trainx,trainy)
```

```
▼ GaussianNB  
GaussianNB()
```

```
ypg = gnb.predict(testx)  
print("Gaussian = ",accuracy_score(testy,ypg))
```

```
mnb= MultinomialNB(alpha=1.0,fit_prior=True)  
mnb.fit(trainx,trainy)
```

```
▼ MultinomialNB  
MultinomialNB()
```

```
ypm = mnb.predict(testx)  
print("Multinomial = ",accuracy_score(testy,ypm))
```

```
Multinomial = 0.8478
```

```
class BernuolliNB(object):  
    def __init__(self, alpha=1.0, fit_prior=False):  
        self.alpha = alpha  
  
    def fit(self, X, y):  
        count_sample = X.shape[0]  
        separated = [[x for x, t in zip(X, y) if t == c] for c in np.unique(y)]  
        self.class_log_prior_ = [np.log(len(i) / count_sample) for i in separated]  
        count = np.array([np.array(i).sum(axis=0) for i in separated]) + self.alpha  
        smoothing = 2 * self.alpha  
        n_doc = np.array([len(i) + smoothing for i in separated])  
        self.feature_prob_ = count / n_doc[np.newaxis].T
```



```
    return self

def predict_log_proba(self, X):
    return [(np.log(self.feature_prob_) * x + \
              np.log(1 - self.feature_prob_) * np.abs(x - 1)
              ).sum(axis=1) + self.class_log_prior_ for x in X]

def predict(self, X):
    return np.argmax(self.predict_log_proba(X), axis=1)
```

```
bnb= BernoulliNB(alpha=1.0,fit_prior=True)
bnb.fit(trainx,trainy)

ypb = bnb.predict(testx)
print("Bernoulli = ",accuracy_score(testy,ypb))
```

```
Bernoulli = 0.8493
```

6. Limitations and Future Enhancements

Limitations:

Domain Specificity: This project's sentiment analysis model was created especially for evaluations of motion pictures. Consequently, its applicability to different domains or categories of textual data may restrict its performance.

Sensitivity to Linguistic Nuances: Although the model attempts to faithfully convey the mood seen in movie reviews, it could have trouble picking up on subtle linguistic elements like sarcasm, irony, or cultural allusions, which are frequently used in these types of reviews.

Data Imbalance: One sentiment class (positive or negative) may be overrepresented in the dataset used to train and test the model due to class imbalance. The model's capacity to effectively generalize to new data may be impacted by this mismatch.

Dependency on Preprocessing for Performance: The quality of preprocessing stages like feature extraction, tokenization, and noise reduction has a significant impact on how well the sentiment analysis model performs. Poor preprocessing might result in less than ideal performance.

Restricted Range of Evaluation Metrics: Although accuracy_score is frequently used to analyze classification model performance, it might not offer a thorough analysis of the model's performance in situations when classes are unbalanced or misclassification costs are high.

Future enhancements:

Fine-Tuning Hyperparameters: To achieve better results, try varying the hyperparameter values for the Naive Bayes classifiers and other model elements.

Ensemble Methods: Experimenting with ensemble learning strategies, such as merging predictions from several classifiers or employing ensemble models like Gradient Boosting or Random Forest, may improve the sentiment analysis model's resilience and capacity for generalization.

Architectures for Deep Learning: It may be possible to identify more complex patterns in textual data and enhance performance by looking at the use of deep learning architectures for sentiment analysis, such as convolutional neural networks (CNNs) or recurrent neural networks (RNNs).

Domain Adaptation: The sentiment analysis model's usability and efficacy across a variety of movie review datasets may be improved by tailoring it to various movie genres or cultural settings.

7. Conclusion

As a result, this study has shown how to create and assess a sentiment analysis model especially for movie reviews. The goal of the model is to automatically identify movie reviews as positive or negative based on the sentiment indicated in the text by utilizing machine learning algorithms and natural language processing techniques.

The model efficiently converts unprocessed textual input into numerical feature vectors that may be classified by applying preprocessing processes including noise reduction, tokenization, and feature extraction using the Bag-of-Words method. Additionally, a thorough assessment of sentiment classification performance is made possible by the incorporation of many Naive Bayes classifiers, such as Bernoulli, Multinomial, and Gaussian Naive Bayes.

Although the approach shows promise in categorizing movie reviews, it has many drawbacks. Problems including data imbalance, sensitivity to linguistic subtleties, and domain specificity point to areas that need to be improved. Furthermore, the model's performance and applicability may be further improved in the future by incorporating deep learning architectures, adjusting hyperparameters, and investigating ensemble techniques.

All things considered, this effort advances the area of sentiment analysis by offering a useful instrument for figuring out the emotions expressed in movie reviews. The sentiment analysis model may be improved to offer more precise and dependable sentiment categorization, helping users in decision-making processes associated with movie review and recommendation. This can be achieved by resolving the constraints that have been found and investigating potential future improvements.

Reference:

1. Pang, Bo, and Lillian Lee. "Opinion mining and sentiment analysis." *Foundations and Trends® in Information Retrieval* 2.1–2 (2008): 1-135.
2. Manning, Christopher D., Prabhakar Raghavan, and Hinrich Schütze. "Introduction to information retrieval." Cambridge University Press, 2008.
3. Bird, Steven, Edward Loper, and Ewan Klein. "Natural language processing with Python: analyzing text with the natural language toolkit." O'Reilly Media, Inc., 2009.
4. Pedregosa, Fabian, et al. "Scikit-learn: Machine learning in Python." *Journal of machine learning research* 12.Oct (2011): 2825-2830.