IMDb SENTIMENTAL ANALYSIS USING ML

INNOVATIVE PRODUCT DEVELOPMENT-I REPORT

Submitted by

U.DEDEEPYA	22RH1A67C4
V.AKSHARA	22RH1A67C7
Y.LASYA	22RH1A67C8
T.SARIKA	23RH5A6711

Under the Esteemed Guidance of

Bhimavarapu Revathi
Assistant Professor

in partial fulfillment of the Academic Requirements for the Degree of

BACHELOR OF TECHNOLOGY

CSE-DATA SCIENCE

MALLA REDDY ENGINEERING COLLEGE FOR WOMEN
(Autonomous Institution-UGC, Govt. of India)
Accredited by NBA & NAAC with 'A' Grade
National Ranking by NIRF – Rank band (151-300), MHRD, Govt. of India
Approved by AICTE, Affiliated to JNTUH, ISO 9001:2015 Certified Institution
Maisammaguda, Dhulapally, Kompally, Secunderabad, -500100.

2023 – 2024

MALLA REDDY ENGINEERING COLLEGE FOR WOMEN

(Autonomous Institution-UGC, Govt. of India)

Accredited by NBA & NAAC with 'A' Grade

National Ranking by NIRF – Rank band (151-300), MHRD, Govt. of India

Approved by AICTE, Affiliated to JNTUH, ISO 9001:2015 Certified Institution

Maisammaguda, Dhulapally, Kompally, Secunderabad, -500100.

DEPARTMENT OF CSE – DATA SCIENCE

CERTIFICATE

This is to certify that the Innovative Product Development work entitled **IMDb SENTIMENTAL ANALYSIS USING Ml** is carried out by U.Dedeepya(22RH1A67C4), V.Akshara(22RH1A67C4), Y.Lasya(22RH1A67C8), T.Sarika(23RH5A6711) in partial fulfillment for the award of degreeof BACHELOR OF TECHNOLOGY in CSE - DS, Jawaharlal Nehru Technological University, Hyderabad during the academic year 2023-2024.

Supervisor Signature

Bhimavarapu Revathi
Assistant Professor

Head of the Department

Dr.V.Pradeep
Assistant Professor

External Examiner

MALLA REDDY ENGINEERING COLLEGE FOR WOMEN
(Autonomous Institution-UGC, Govt. of India)
Accredited by NBA & NAAC with 'A' Grade
National Ranking by NIRF – Rank band (151-300), MHRD, Govt. of India
Approved by AICTE, Affiliated to JNTUH, ISO 9001:2015 Certified Institution
Maisammaguda, Dhulapally, Kompally, Secunderabad, -500100.

DEPARTMENT OF CSE – DATA SCIENCE

DECLARATION

We hereby declare that the Innovative Product Development entitled IMDb SENTIMENTAL ANALYSIS USING ML submitted to Malla Reddy Engineering College for Women affiliated to Jawaharlal Nehru Technological University, Hyderabad (JNTUH) for the award of the Degree of **Bachelor of Technology** in CSE-DS is a result of original research work done by us. It is further declared that the Innovative Product Development report or any part thereof has not been previously submitted to any University or Institute for the award of Degree.

U. Dedeepya (22RH1A67C4)

V. Akshara (22RH1A67C7)

Y. Lasya (22RH1A67C8)

T. Sarika (23RH5A6711)

ACKNOWLEDGEMENT

We feel ourselves honored and privileged to place our warm salutation to our college **Malla Reddy Engineering College for Women and Department of Computer Science and Engineering- Data Science** which gave us the opportunity to have expertise in engineering and profound technical knowledge.

We would like to deeply thank our Honorable Minister of Telangana State **Sri.Ch. Malla Reddy Garu**, founder chairman MRGI, the largest cluster of institutions in the state of Telangana for providing us with all the resources in the college to make our project success.

We wish to convey gratitude to our Principal **Dr. Y. Madhavee Latha**, for providing us with the environment and mean to enrich our skills and motivating us in our endeavor and helping us to realize our full potential.

We would like to thank Assistant prof **A. Radha Rani**, Director of Computer Science and Engineering & Information Technology for encouraging us to take up a project on this subject and motivating us towards the Project Work.

We express our sincere gratitude to **Dr. V. Pradeep**, Head of the Department of CSE – DATA SCIENCE for inspiring us to take up a project on this subject and successfully guiding us towards its completion.

We would like to thank our guide **Bhimavarapu Revathi**, Professor and all the Faculty members for their valuable guidance and encouragement towards the completion of our project work.

With Regards and Gratitude

U. Dedeepya (22RH1A67C4)
V. Akshara (22RH1A67C7)
Y. Lasya (22RH1A67C8)
T . Sarika (23RH5A6711)

ABSTRACT

Text is the largest repository of human knowledge acquired over thousands of years. This knowledge will impart even more meaning if mined for deeper insights. Sentiment Analysis (SA) provides a traditional machine learning (ML) solution to this problem. In this, we have performed SA on the IMDb movie reviews dataset to demonstrate how valuable insights can be drawn from a bulk of textual data collected from the internet. We derive these insights by applying two traditional ML algorithms namely, Naive Bayes (NB) and Logistic Regression (LR). The model used in this project is Scikit-learn.

Sentiment analysis is the study, to classify the text based on customer reviews which can provide valuable information to improve business. Previously the analysis was carried out based on the information provided by the customers using natural language processing and machine learning. In this project, sentiment analysis on IMDb movie reviews dataset is implemented using Machine Learning (ML) approaches to measure the accuracy of the model. ML algorithms are the traditional algorithms that work in a single layer gives better output.

INDEX

CHAPTER 1

INTRODUCTION

PROBLEM STATEMENT:

In places where the number of tweets increases and analysing documents are large, it becomes impossible for a single person to evaluate the sentiment of the texts. Thus, we require the ability of machines to process the data in a short period of time and generate the results. With the ability to learn and process things at a higher rate, the machines never fail to produce the best result.

PROBLEM OVERVIEW:

The knowledge of the sentiments of the people is very important for business purposes as it serves as a base to the needs and requirements of the customer upon which the business can produce good quality goods. The feedback of the customers is equally important as it provides valuable insights governing the tendency to like or dislike a product. This way the demand of the project will be met by the organisations. It will also help the business to know the performance of their project in the market and check if the customers are satisfied with the quality and pricing plans.

CHAPTER 2

LITERATURE SURVEY

ADVANTAGES

- **Efficiency:** ML algorithms process vast amounts of data quickly, providing efficient sentiment analysis of numerous IMDb reviews.

- **Accuracy:** ML models can learn to identify subtle nuances in language, improving the accuracy of sentiment classification compared to rule-based systems.

- **Scalability:** The approach is scalable, allowing for the analysis of a large volume of reviews, providing a comprehensive understanding of audience sentiment.

- **Adaptability:** ML models can adapt to changes in language use and user behavior, ensuring the sentiment analysis remains relevant over time.

- **Insight Generation**: By analyzing sentiments, valuable insights can be derived, helping filmmakers, studios, and advertisers understand audience preferences and improve content.

- **Decision Support:** Studios and creators can use sentiment analysis to make informed decisions about marketing strategies, content improvements, or future projects based on audience feedback.

TECHNIQUES

- **Text Pre-processing:**

  Tokenization: Break the text into individual words or tokens.

  Lower casing: Convert all words to lowercase to ensure uniformity.

  Removing Stopwords: Eliminate common words (e.g., "the," "and," "is") that do not contribute much to sentiment.

- **Feature Extraction**:

  Bag of Words (BoW): Represent each review as a vector of word frequencies.

  TF-IDF (Term Frequency-Inverse Document Frequency): Weigh words based on their importance in the entire dataset.

- **Word Embeddings**:

  Word2Vec, GloVe, FastText: Represent words as dense vectors to capture semantic relationships.

  Embedding Layer in Neural Networks: Use pre-trained word embeddings or train them on IMDb reviews.

- **Machine Learning Models:**

  Naive Bayes: Simple probabilistic model based on Bayes' theorem.

  Logistic Regression: Uses for finding relationship between the variables depending upon the factors given.It gives outcome like YES or NO

- **Sentiment Lexicons:**

  Use predefined lists of words associated with positive or negative sentiment. Assign sentiment scores based on the presence of these words in reviews.

- **Evaluation Metrics:**

  Use metrics such as accuracy, precision, recall, and F1 score to evaluate the performance of the model.

- **Handling Imbalanced Data:**

  IMDb reviews may be imbalanced, with more positive or negative reviews. Use techniques like oversampling, under sampling, or class weights to address this imbalance.

- **Hyper-parameter Tuning:**

  Optimize model hyper-parameters through techniques like grid search or random search.

- **Model Interpretability**:

  Use techniques like LIME (Local Interpretable Model-agnostic Explanations) to interpret and explain the predictions of complex models.

CHAPTER 3

METHODOLOGY

METHODOLOGY FOR LOGISTIC REGRESSION

Data Collection:

Collect a dataset of IMDb reviews with sentiment labels (positive or negative).

Data Pre-processing:

Clean and pre-process the text data by removing HTML tags, special characters, and stopwords.

Perform tokenization and lemmatization.

Labeling:

Assign sentiment labels to each review.

Data Splitting:

Split the dataset into training and testing sets.

Text Representation:

Convert text data into numerical features using TF-IDF or Bag-of-Words.

Logistic Regression Model:

Train a logistic regression model on the training set.

Hyper-parameter Tuning:

Optimize hyper-parameters (e.g., regularization strength) using cross-validation.

Evaluation:

Evaluate the model on the testing set using metrics like accuracy, precision, recall, and F1 score.

Error Analysis:

Analyze misclassifications to identify patterns and areas for improvement.

METHODOLOGY FOR NAIVE BAYES

Text Representation:

Use the same TF-IDF or Bag-of-Words representation as in Logistic Regression.

Naive Bayes Model:

Implement a Naive Bayes classifier (e.g., Multinomial Naive Bayes) using the training set.

Smoothing:

Apply Laplace smoothing to handle unseen words in the test data.

Evaluation:

Evaluate the Naive Bayes model on the testing set using metrics like accuracy, precision, recall, and F1 score.

Comparison:

Compare the performance of Logistic Regression and Naive Bayes models.

Ensemble:

Consider creating an ensemble of both models for potentially improved performance.

CHAPTER 4

SOURCE CODE

Data Description

In [21]:
```
import pandas as pd
import seaborn as sns
# Load dataset
df = pd.read_csv("IMDB_Dataset.csv")
```

In [22]:
```
df
```

Out[22]:

	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
...	...	...
49995	I thought this movie did a down right good job...	positive
49996	Bad plot, bad dialogue, bad acting, idiotic di...	negative
49997	I am a Catholic taught in parochial elementary...	negative
49998	I'm going to have to disagree with the previou...	negative
49999	No one expects the Star Trek movies to be high...	negative

50000 rows × 2 columns

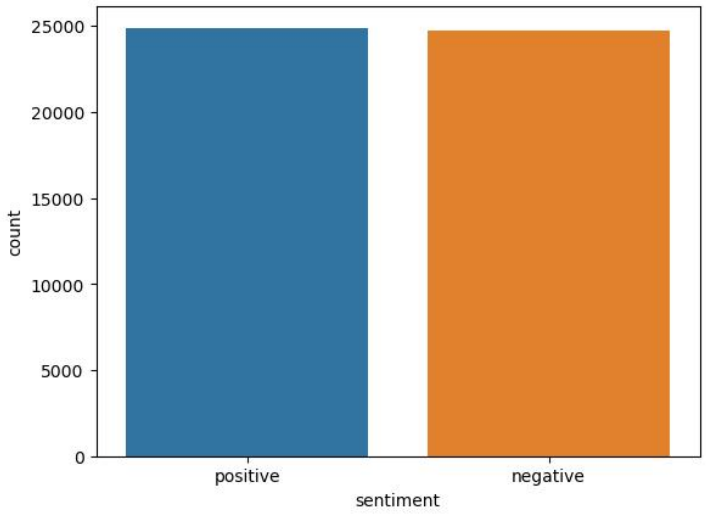
In [3]:
```
# Dropping the duplicate rows
duplicate_rows = df[df.duplicated()]
print("number of duplicate rows : ",duplicate_rows.shape)

df.drop_duplicates(inplace=True)
print("number of duplicate rows : ",df[df.duplicated()].shape)
```

```
number of duplicate rows :  (418, 2)
number of duplicate rows :  (0, 2)
```

In [4]:
```
# data visualization
sns.countplot(x=df['sentiment'])
```

Out[4]: <Axes: xlabel='sentiment', ylabel='count'>



In [5]:
```
# Mapping of label values
df_copy = df.copy()
print(df.head(10))

print("_____________________________________________________________")
df_copy["sentiment"] = df_copy["sentiment"].map({"negative":0,"positive":1})
print("Sentiment :\n",df_copy.head(20))
```

```
                                              review sentiment
0  One of the other reviewers has mentioned that ...  positive
1  A wonderful little production. <br /><br />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
5  Probably my all-time favorite movie, a story o...  positive
6  I sure would like to see a resurrection of a u...  positive
7  This show was an amazing, fresh & innovative i...  negative
8  Encouraged by the positive comments about this...  negative
9  If you like original gut wrenching laughter yo...  positive
_____________________________________________________________
Sentiment :
                                              review  sentiment
0   One of the other reviewers has mentioned that ...          1
1   A wonderful little production. <br /><br />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
5   Probably my all-time favorite movie, a story o...          1
6   I sure would like to see a resurrection of a u...          1
7   This show was an amazing, fresh & innovative i...          0
8   Encouraged by the positive comments about this...          0
9   If you like original gut wrenching laughter yo...          1
10  Phil the Alien is one of those quirky films wh...          0
11  I saw this movie when I was about 12 when it c...          0
12  So im not a big fan of Boll's work but then ag...          0
13  The cast played Shakespeare.<br /><br />Shakes...          0
14  This a fantastic movie of three prisoners who ...          1
15  Kind of drawn in by the erotic scenes, only to...          0
```

Text preprocessing

Remove panctuation

In [6]:
```
import string
string.punctuation
```

Out[6]:
```
'!"#$%&\'()*+,-./:;<=>?@[\\]^_`{|}~'
```

In [7]:
```
print("data before ... ")
print(df_copy.head(10))
print("_____________________________________________________________")

def remove_punctuation(text):
    text_nopunc = "".join([char for char in text if char not in string.punctuation]) # delete all punctuation
    return text_nopunc
df_copy['review_pun'] = df_copy['review'].apply(lambda x: remove_punctuation(x))
# Display first 10 records after remove punctuation

print("data after ... ")
print(df_copy.head(10))
print("_____________________________________________________________")
```

Tokenization

In [8]:
```
import re

# Function to Tokenize words
def tokenize(text):
    tokens = re.split('\W+', text) #W+ means that either a word character (A-Za-z0-9_) or a dash (-) can go there.
    return tokens

df_copy['review_tokenized'] = df_copy['review_pun'].apply(lambda x: tokenize(x.lower()))
#We convert to lower as Python is case-sensitive.

df_copy.head()
```

Out[8]:

	review	sentiment	review_pun	review_tokenized
0	One of the other reviewers has mentioned that ...	1	One of the other reviewers has mentioned that ...	[one, of, the, other, reviewers, has, mentione...
1	A wonderful little production. The...	1	A wonderful little production br br The filmin...	[a, wonderful, little, production, br, br, the...
2	I thought this was a wonderful way to spend ti...	1	I thought this was a wonderful way to spend ti...	[i, thought, this, was, a, wonderful, way, to,...
3	Basically there's a family where a little boy ...	0	Basically theres a family where a little boy J...	[basically, theres, a, family, where, a, littl...
4	Petter Mattei's "Love in the Time of Money" is...	1	Petter Matteis Love in the Time of Money is a ...	[petter, matteis, love, in, the, time, of, mon...

Remove stopwords

In [9]:
```
import nltk
nltk.download('stopwords')
from nltk.corpus import stopwords
stop_words = stopwords.words('english')
print(stop_words)
```

```
['i', 'me', 'my', 'myself', 'we', 'our', 'ours', 'ourselves', 'you', "you're", "you've", "you'll", "you'd", 'your', 'yours', 'y
ourself', 'yourselves', 'he', 'him', 'his', 'himself', 'she', "she's", 'her', 'hers', 'herself', 'it', "it's", 'its', 'itself',
'they', 'them', 'their', 'theirs', 'themselves', 'what', 'which', 'who', 'whom', 'this', 'that', "that'll", 'these', 'those',
'am', 'is', 'are', 'was', 'were', 'be', 'been', 'being', 'have', 'has', 'had', 'having', 'do', 'does', 'did', 'doing', 'a', 'a
n', 'the', 'and', 'but', 'if', 'or', 'because', 'as', 'until', 'while', 'of', 'at', 'by', 'for', 'with', 'about', 'against', 'b
etween', 'into', 'through', 'during', 'before', 'after', 'above', 'below', 'to', 'from', 'up', 'down', 'in', 'out', 'on', 'of
f', 'over', 'under', 'again', 'further', 'then', 'once', 'here', 'there', 'when', 'where', 'why', 'how', 'all', 'any', 'both',
'each', 'few', 'more', 'most', 'other', 'some', 'such', 'no', 'nor', 'not', 'only', 'own', 'same', 'so', 'than', 'too', 'very',
's', 't', 'can', 'will', 'just', 'don', "don't", 'should', "should've", 'now', 'd', 'll', 'm', 'o', 're', 've', 'y', 'ain', 'ar
en', "aren't", 'couldn', "couldn't", 'didn', "didn't", 'doesn', "doesn't", 'hadn', "hadn't", 'hasn', "hasn't", 'haven', "have
n't", 'isn', "isn't", 'ma', 'mightn', "mightn't", 'mustn', "mustn't", 'needn', "needn't", 'shan', "shan't", 'shouldn', "should
n't", 'wasn', "wasn't", 'weren', "weren't", 'won', "won't", 'wouldn', "wouldn't"]
```

```
[nltk_data] Downloading package stopwords to
[nltk_data]     C:\Users\Dell\AppData\Roaming\nltk_data...
[nltk_data]   Package stopwords is already up-to-date!
```

In [10]:
```
print("data before ... ")
print(df_copy.head(10))
print("_____________________________________________________________________")

def remove_stopwords(text):
    text = [word for word in text if word not in stop_words]
    return text

df_copy['review_stopword'] = df_copy['review_tokenized'].apply(lambda x: remove_stopwords(x))

print("data after ... ")
print(df_copy.head(10))
print("_____________________________________________________________________")
```

Lemetizing

In [*]:
```
nltk.download('wordnet')
print("data before ... ")
print(df_copy.head(10))
print("_____________________________________________________________________")
wn = nltk.WordNetLemmatizer()

def lemetizing(text):
    text = [wn.lemmatize(word) for word in text]
    return text
df_copy['review_lem'] = df_copy['review_stopword'].apply(lambda x: lemetizing(x))

print("data after ... ")
print(df_copy.head(10))
print("_____________________________________________________________________")
```

In [*]:
```
# Reconstruct the text
print("data before ... ")
print(df_copy.head(10))
print("_____________________________________________________________________")

def construct(text):
  text_clean = ""
  for word in text:
    text_clean += word +" "
  return text_clean

df_copy['review_cleaned'] = df_copy['review_lem'].apply(lambda x: construct(x))

print("data after ... ")
print(df_copy.head(10))
print("_____________________________________________________________________")
```

Models Implementation

In [13]:
```
# import libraries
from sklearn.feature_extraction.text import TfidfVectorizer
from sklearn.metrics import accuracy_score,confusion_matrix,ConfusionMatrixDisplay
from sklearn.model_selection import train_test_split
from sklearn.naive_bayes import MultinomialNB
from sklearn import metrics
```

In [14]:
```
y = df_copy["sentiment"].values
# split the dataset into training data and test data
X_train, X_test, y_train, y_test = train_test_split(df_copy, y,
                        test_size= 0.20, random_state=100, stratify=y)

vec = TfidfVectorizer()
train_vectors = vec.fit_transform(X_train['review'])
test_vectors = vec.transform(X_test['review'])

print("train data : ", train_vectors.shape)
print("test data : ", test_vectors.shape)
```

```
train data :  (39665, 93100)
test data :  (9917, 93100)
```

In [15]:
```
# Classify using Naive Bayes 0.86

MNB_clf = MultinomialNB()

# fit model no training data
MNB_clf.fit(train_vectors, y_train)

# making predictions on the testing set
predicted = MNB_clf.predict(test_vectors)

# Classification report
print("Classification report : \n", MNB_clf,"\n",
      metrics.classification_report(y_test, predicted))
```

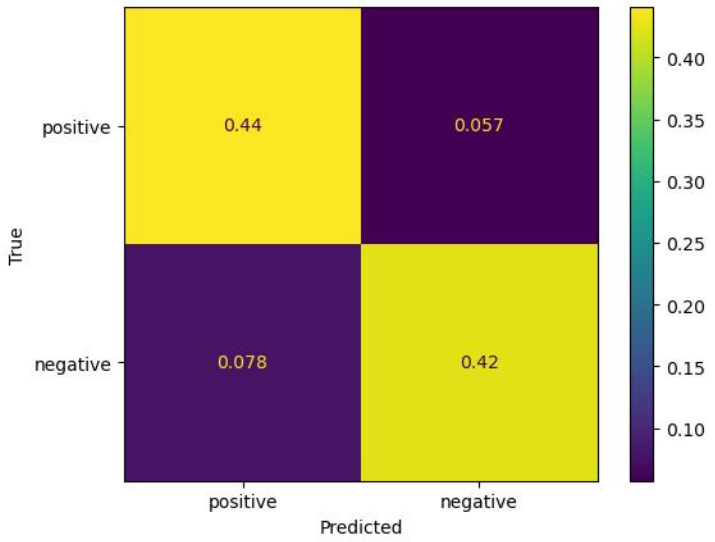
```
Classification report :
 MultinomialNB()
              precision    recall  f1-score   support

           0       0.85      0.88      0.87      4940
           1       0.88      0.84      0.86      4977

    accuracy                           0.86      9917
   macro avg       0.86      0.86      0.86      9917
weighted avg       0.87      0.86      0.86      9917
```

In [16]:
```
cm = confusion_matrix(y_test, predicted, normalize='all')
cmd = ConfusionMatrixDisplay(cm, display_labels=['positive','negative'])
cmd.plot()
cmd.ax_.set(xlabel='Predicted', ylabel='True')
```

Out[16]:
```
[Text(0.5, 0, 'Predicted'), Text(0, 0.5, 'True')]
```



In [17]:
```
from sklearn.linear_model import LogisticRegression

# Create the classifier
# multi_class: default = "auto"
logreg = LogisticRegression(random_state=42)

# Train the classifier
logreg.fit(train_vectors, y_train)

# Predict the value of X_test
predicted = logreg.predict(test_vectors)

# Classification report
print("Classifier : ", logreg)
print("Classification report for classifier : \n", metrics.classification_report(y_test, predicted))
```

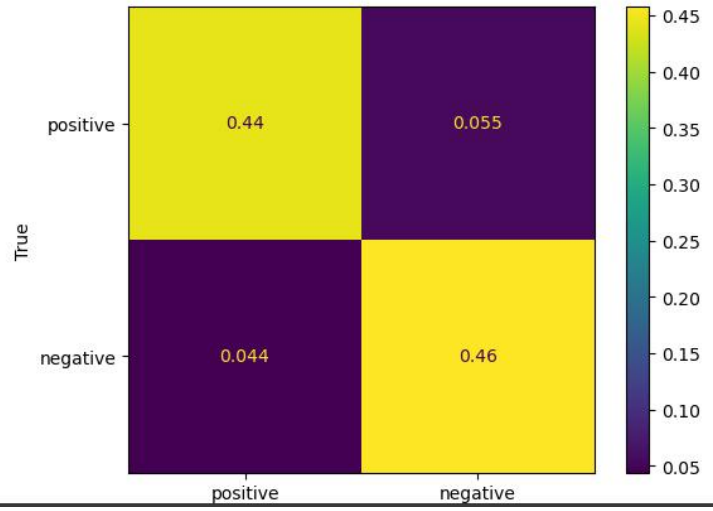
```
Classifier :  LogisticRegression(random_state=42)
Classification report for classifier :
              precision    recall  f1-score   support

           0       0.91      0.89      0.90      4940
           1       0.89      0.91      0.90      4977

    accuracy                           0.90      9917
   macro avg       0.90      0.90      0.90      9917
weighted avg       0.90      0.90      0.90      9917
```

In [18]:
```
cm = confusion_matrix(y_test, predicted, normalize='all')
cmd = ConfusionMatrixDisplay(cm, display_labels=['positive','negative'])
cmd.plot()
cmd.ax_.set(xlabel='Predicted', ylabel='True')
```

Out[18]: [Text(0.5, 0, 'Predicted'), Text(0, 0.5, 'True')]



RESULT ANALYSIS

MULTINOMIAL NAIVE BAYES

```
Classification report :
 MultinomialNB()
              precision    recall  f1-score   support

           0       0.85      0.88      0.87      4940
           1       0.88      0.84      0.86      4977

    accuracy                           0.86      9917
   macro avg       0.86      0.86      0.86      9917
weighted avg       0.87      0.86      0.86      9917
```

LOGISTIC REGRESSION

```
Classifier :  LogisticRegression(random_state=42)
Classification report for classifier :
               precision    recall  f1-score   support

           0       0.91      0.89      0.90      4940
           1       0.89      0.91      0.90      4977

    accuracy                           0.90      9917
   macro avg       0.90      0.90      0.90      9917
weighted avg       0.90      0.90      0.90      9917
```

CHAPTER 5

FUTURE SCOPE

The future scope of this project is as follows

1.Fine-grained Analysis: Advances in natural language processing (NLP) may lead to more nuanced sentiment analysis, enabling systems to understand and interpret subtle emotions
and context.

2.Multi modal Sentiment Analysis: Integration of text, images, and videos for a more comprehensive understanding of sentiment,allowing systems to analyze emotions conveyed through various mediums.

3.Cross-ingual Sentiment Analysis: Improved capabilities to analyze sentiment in multiple languages, making sentiment analysis more globally applicable and effective.

4.Contextual Understanding: Enhancements in machine learning models to better grasp context, sarcasm, and cultural nuances, leading to more accurate sentiment interpretation.

5.Real-time Sentiment Analysis: Faster and more efficient algorithms enabling real-time sentiment analysis for quick responses in applications such as customer service, social media monitoring, and financial trading.

6.Personalized Sentiment Analysis: Customized sentiment analysis models tailored to individual users, providing more accurate insights based on personal preferences and communication styles.

7.Ethical and Bias Considerations: A growing focus on addressing bias in sentiment analysis algorithms and ensuring ethical use, promoting fairness and accountability in automated decision-making based on sentiment.

8.Emotion Detection: Advancements in identifying specific emotions rather than just positive, negative, or neutral sentiments,leading to more detailed emotional insights.As technology continues to evolve, the future of sentiment analysis is likely to witness a convergence of advancements in Al,. NLP. And data processing, offering more sophisticated and context-aware sentiment analysis capabilities.

CHAPTER 6

CONCLUSION

In the context of our project we have presented an approach for the classification of the sentiments using the IMDb dataset. Pre-processing of the dataset was done to make it suitable to be fed to the classifier model. Bag of Words approach was chosen for text representation in the work. Finally, two different traditional machine learning algorithms were deployed for getting results. These results were then compared on the basis of different evaluation metrics. Using Logistic Regression and Multinomial Naive Bayes gave the best validation for our task.

REFERENCES

- https://ieeexplore.ieee.org/document/10080860
- https://www.researchgate.net/publication/341904376_Sentiment_Analysis_of_Movie_Review_using_Machine_Learning_Approach
- https://www.analyticsvidhya.com/blog/2022/02/sentiment-analysis-of-imdb-reviews-with-nlp/
- https://www.kaggle.com/c/sentiment-analysis-on-movie-reviews
- https://www.diva-portal.org/smash/get/diva2:1779708/FULLTEXT02
- https://paperswithcode.com/dataset/imdb-movie-reviews