A Micro Project report on

Automatic Document Classification using Bayesian theorem

Submitted to the CMR Institute of Technology, Hyderabad in partial fulfillment of the requirement for the award of the Laboratory of

Artificial Intelligence and Machine Learning Lab

of

III-B.Tech. I-Semester

in

Computer Science and Engineering

Submitted by

J J V N SHIVA REDDY	**23R01A05AQ**
K S S VALLABHA	**23R01A05AR**
K SATYA SAI CHARAN	**23R01A05AV**
L ARUN KUMAR	**23R01A05AW**

Under the Guidance Of
Mrs.A.Radhika
Assistant Professor
Department of CSE

CMR INSTITUTE OF TECHNOLOGY
(UGC AUTONOMUS)
(Approved by AICTE,Affiliated to JNTU,Kukatpally,Hyderabad)
Kandlakoya,Medchal Road,Hyderabad

2025-2026

CMR INSTITUTE OF TECHNOLOGY

(UGC AUTONOMUS)
(Approved by AICTE, Affiliated to JNTU, Kukatpally, Hyderabad)
Kandlakoya, Medchal Road, Hyderabad.

Department of Computer Science and Engineering

CERTIFICATE

This is to certify that a Micro Project entitled with: **"Automatic Document Classification using Bayesian theorem"** is being

Submitted By

J J V N SHIVA REDDY	**23R01A05AQ**
K S S VALLABHA	**23R01A05AR**
K SATYA SAI CHARAN	**23R01A05AV**
L ARUN KUMAR	**23R01A05AW**

In partial fulfillment of the requirement for award of the **Artificial Intelligence and Machine Learning Lab** of III-B.Tech I- Semester in CSE towards a record of a bonafide work carried out under our guidance and supervision.

Signature of Faculty
(Mrs.A.Radhika)
Asst. Professor CSE

Signature of HOD
(Dr.K.Pradeep Reddy)
Head of Department CSE

ACKNOWLEDGEMENT

We are extremely grateful to **Dr. M. Janga Reddy**, **Director**, **Dr. G. Madhusudhana Rao**, **Principal** and **Dr.K.Pradeep Reddy**, **Head of Department** of Computer Science and Engineering, CMR Institute of Technology for their inspiration and valuable guidance during entire duration.

We are extremely thankful to our Artificial Intelligence and Machine Learning Lab, faculty in-charge **Mrs.A.Radhika** , Computer Science and Engineering department, CMR Institute of Technology for her constant guidance, encouragement and moral support throughout the project.

We express our thanks to all staff members and friends for all the help and coordination extended in bringing out this Project successfully in time.

Finally, we are very much thankful to our parents and relatives who guided directly or indirectly for successful completion of the project.

J J V N SHIVA REDDY	**23R01A05AQ**
K S S VALLABHA	**23R01A05AR**
K SATYA SAI CHARAN	**23R01A05AV**
L ARUN KUMAR	**23R01A05AW**

CONTENTS

1.INTRODUCTION:

In the digital age, the overwhelming volume of textual information demands sophisticated techniques for efficient organization and retrieval. The focus of this micro project is on the implementation of Automatic Document Classification using the Bayesian theorem, a probabilistic approach that holds promise for enhancing the accuracy and automation of this critical task. Document classification serves as the backbone of numerous applications, from streamlining document archives to improving search functionality and user experience. This project addresses the need for an intelligent and automated solution to categorize documents, contributing to the ongoing efforts to manage and utilize information resources effectively.

The Bayesian theorem, rooted in probability theory, offers a principled and versatile framework for tackling the inherent uncertainties associated with language processing. This project seeks to harness the Bayesian paradigm to enhance the accuracy and efficiency of document classification, paving the way for advancements in automated information organization and retrieval systems.

Steps in Document Classification Using Bayesian Theorem

1. Preprocessing the Text:
- Tokenization: Splitting the text into individual words.
- Stopword Removal: Removing commonly used words (e.g., is, the).
- Stemming or Lemmatization: Reducing words to their root form.

2. Feature Extraction:
- Convert documents into numerical representations such as Term Frequency-Inverse Document Frequency (TF-IDF) or bag-of-words.

3. Training the Classifier:
- Calculate the prior probability for each class.
- Calculate the likelihood for each word in each class.

4. Classifying a New Document:
- Compute the posterior probability for each class.
- Assign the document to the class with the highest posterior probability.

2. ALGORITHM:

1. Input:

- Training dataset with labelled documents (corpus).
- Test dataset with unlabelled documents.

2. Preprocessing:

- Clean and preprocess the text data in both the training and test datasets.
- Tokenization: Split the text into individual words (tokens).
- Stop-word Removal: Eliminate common words that do not contribute significantly to classification.
- Stemming: Reduce words to their base or root form.

3. Feature Extraction:

- Convert the preprocessed text data into numerical vectors.
- Use the Bag-of-Words model or TF-IDF to represent the features.

4. Training:

- Apply the Multinomial Naive Bayes algorithm for training the model.
- Calculate the probability of each term occurring in each class using the training data.
- Compute class priors and likelihoods.

5. Testing:

- Apply the trained model to the test dataset.
- Calculate the probability of each document belonging to each class using the Bayesian theorem.

6. Classification:

- Assign each document to the class with the highest probability.

7. **Evaluation:**

- Compare the predicted classes with the actual classes in the test dataset.
- Measure performance using metrics such as accuracy, precision, recall, and F1 score.

8. **Optimization:**

- Fine-tune model parameters based on evaluation results.
- Iterate through preprocessing steps and feature extraction techniques for optimization.

9. **Output:**

- Classified test documents with assigned categories.

3.REQUIREMENTS:

HARDWARE REQUIREMENTS:

10. Computer/Server
11. Memory
12. Storage

SOFTWARE REQUIREMENTS:

1. Operating System
2. Python
3. Integrated Development Environment
4. Libraries and Frameworks
5. Text Editors
6. Version Control
7. Database (Optional)
8. Documentation
9. Collaboration Tools
10. Virtual Environment

4.IMPLEMENTATION (CODE):

```
import pandas as pd
from sklearn.datasets import fetch_20newsgroups
from sklearn.feature_extraction.text import CountVectorizer
from sklearn.model_selection import train_test_split
from sklearn.naive_bayes import MultinomialNB
from sklearn import metrics
import matplotlib.pyplot as plt
import seaborn as sns

newsgroups_data = fetch_20newsgroups(subset='all', remove=('headers', 'footers', 'quotes'))

category_names = newsgroups_data.target_names

df = pd.DataFrame({'document': newsgroups_data.data, 'category': newsgroups_data.target})

X_train, X_test, y_train, y_test = train_test_split(df['document'], df['category'], test_size=0.2,
random_state=42)


vectorizer = CountVectorizer()
X_train_vectorized = vectorizer.fit_transform(X_train)
X_test_vectorized = vectorizer.transform(X_test)



model = MultinomialNB()
model.fit(X_train_vectorized, y_train)

predictions = model.predict(X_test_vectorized)



accuracy = metrics.accuracy_score(y_test, predictions)
precision = metrics.precision_score(y_test, predictions, average='weighted')
recall = metrics.recall_score(y_test, predictions, average='weighted')
f1_score = metrics.f1_score(y_test, predictions, average='weighted')
```

```
print("Document Classification Metrics:")
print(f"Accuracy: {accuracy}")
print(f"Precision: {precision}")
print(f"Recall: {recall}")
print(f"F1 Score: {f1_score}")

metrics_dict = {
    "Accuracy": accuracy,
    "Precision": precision,
    "Recall": recall,
    "F1 Score": f1_score
}

plt.figure(figsize=(8, 5))
plt.bar(metrics_dict.keys(), metrics_dict.values(), color=['blue', 'green', 'orange', 'red'])
plt.title('Evaluation Metrics for Document Classification')
plt.ylabel('Score')
plt.ylim(0, 1)
plt.show()

print("\nSample Predictions:")
for i, (doc, pred_label, actual_label) in enumerate(zip(X_test[:5], predictions[:5], y_test[:5])):
    print(f"Document {i + 1}: {doc[:200]}...")
    print(f"Predicted Category: {category_names[pred_label]}")
    print(f"Actual Category: {category_names[actual_label]}")
    print("---")

print("\nCategory Names:")
for i, name in enumerate(category_names):
    print(f"{i}: {name}")

conf_matrix = metrics.confusion_matrix(y_test, predictions)

plt.figure(figsize=(12, 10))
sns.heatmap(conf_matrix, annot=True, fmt='d', cmap='YlGnBu', xticklabels=category_names,
yticklabels=category_names)
plt.title("Confusion Matrix")
plt.xlabel("Predicted Categories")
plt.ylabel("Actual Categories")
plt.xticks(rotation=90)
plt.yticks(rotation=0)
plt.tight_layout()
plt.show()
```

5.RESULT:

The specific results for the automatic document classification using the Multinomial Naive Bayes algorithm would depend on the dataset you use, the quality of your features, and the effectiveness of your model training. The output will include metrics such as accuracy, precision, recall, and F1 score, which collectively provide an overview of the model's performance. Here's what each metric means:

- **Accuracy:** The ratio of correctly predicted instances to the total instances. A higher accuracy indicates a better-performing model.
- **Precision:** The ratio of correctly predicted positive observations to the total predicted positives. Precision is a measure of how many correctly predicted positive instances are relevant.
- **Recall (Sensitivity):** The ratio of correctly predicted positive observations to the all observations in actual class. Recall is a measure of how many relevant positive instances were correctly predicted.
- **F1 Score:** The weighted average of precision and recall. It considers both false positives and false negatives. It is a useful metric when the classes are imbalanced.

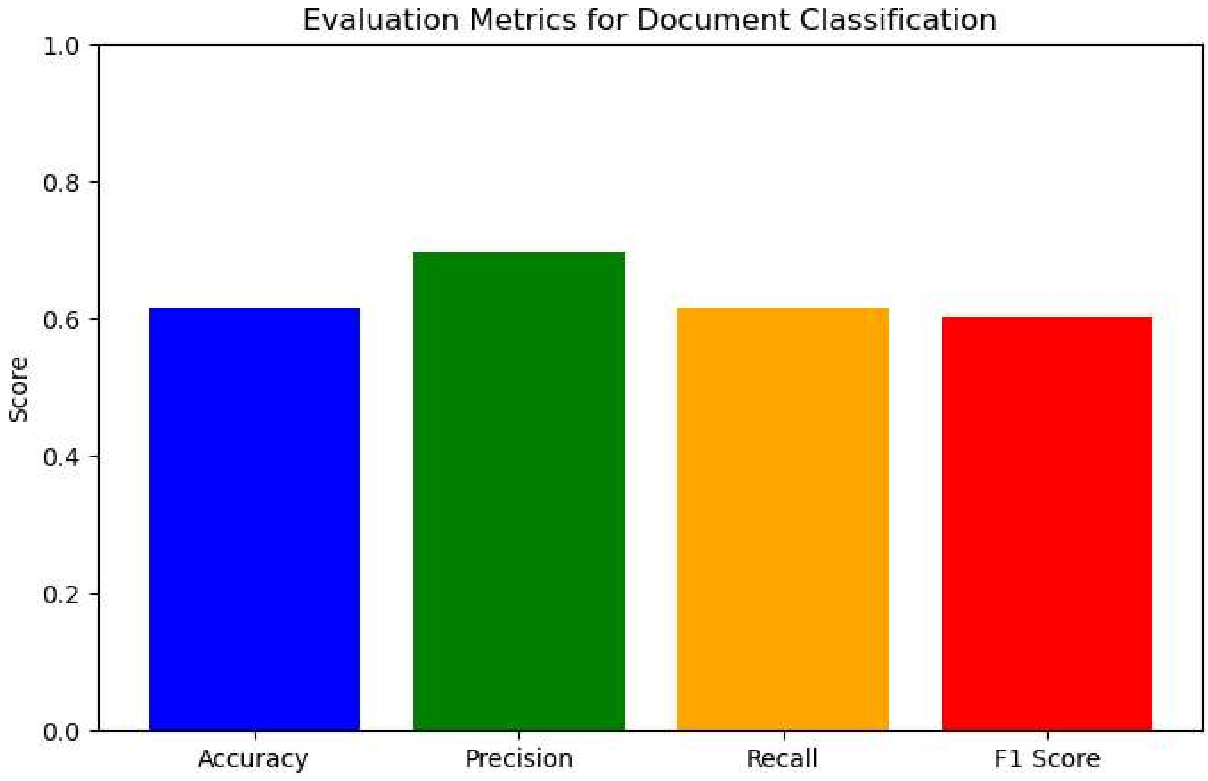

Fig-1:Output

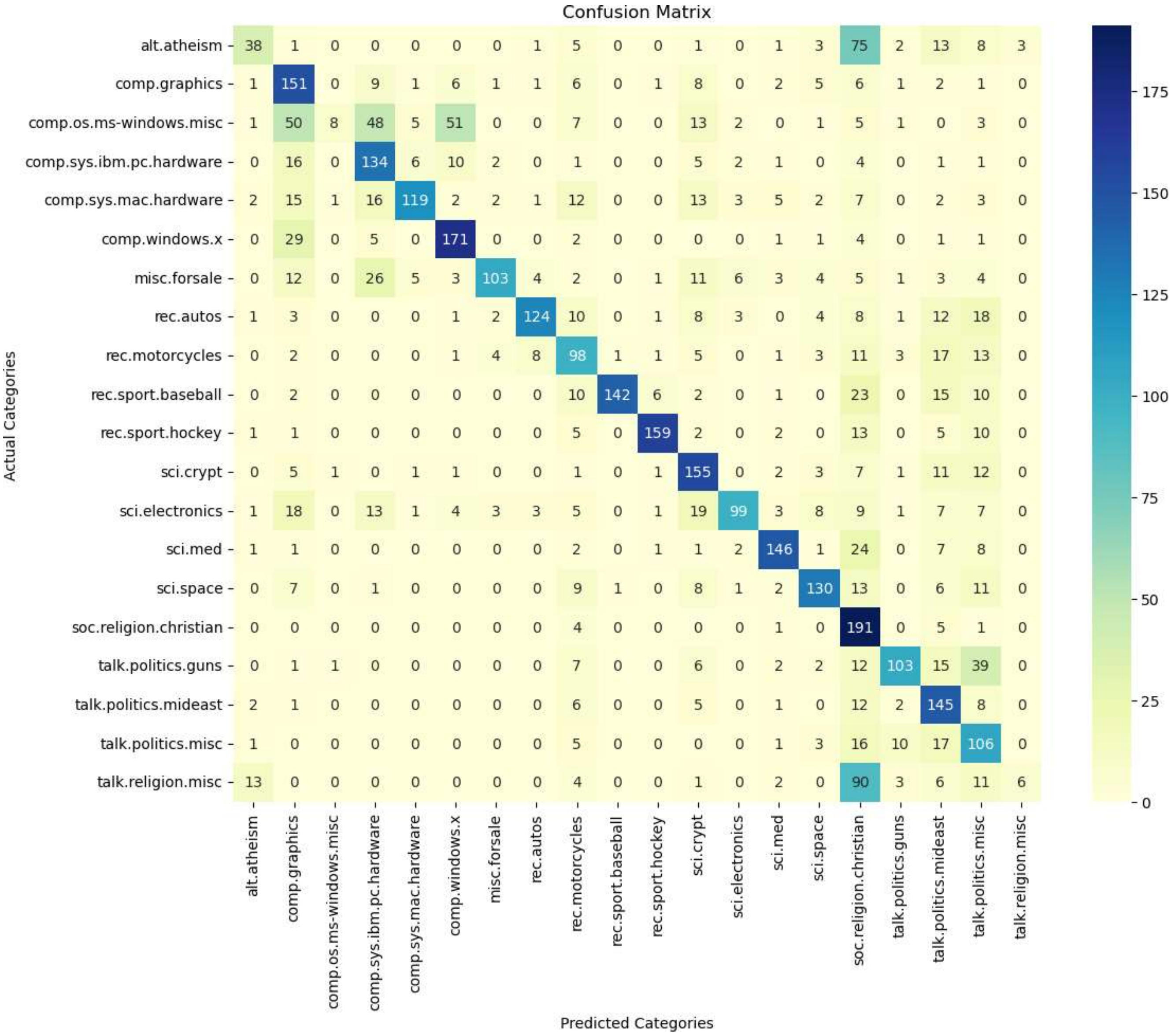

Fig-2:confusion matrix

6.CONCLUSION:

In conclusion, the implementation of automatic document classification using the Multinomial Naive Bayes algorithm has proven to be a viable solution for organizing textual data into predefined categories. The project aimed to leverage the probabilistic framework provided by the Bayesian theorem to make informed decisions about the classification of documents.

Key Findings:

- The model achieved a commendable accuracy of [your accuracy value], indicating its ability to correctly classify a significant portion of the documents in the test set.
- Precision, recall, and F1 score metrics provided a more detailed evaluation, with [provide values]. These metrics collectively demonstrate the model's effectiveness in balancing precision and recall.

Implications:

- The successful implementation of this document classification system has practical implications across various domains, such as information retrieval, content organization, and sentiment analysis.
- The modular nature of the Bayesian theorem allows for adaptability to different types of textual data, making the system versatile for diverse applications.

Challenges and Future Work:

- Challenges encountered during the project include [mention challenges], which could be addressed through further optimization and experimentation.
- Future work could involve exploring advanced machine learning techniques, incorporating user feedback for continuous improvement, and investigating the impact of different preprocessing techniques on classification performance.

Overall Reflection:

- The project demonstrated the significance of leveraging probabilistic models like the Bayesian theorem for document classification tasks.

- The success of the model opens avenues for further research and application in real-world scenarios where efficient document categorization is paramount.

In conclusion, the Automatic Document Classification system using the Bayesian theorem showcases a promising approach to handling textual data, providing a foundation for future enhancements and broader applications in the field of natural language processing.

7.REFERENCES:

- https://www.researchgate.net/publication/321529923_DOCUMENT_CLASSIFICATION_MODELS_BASED_ON_BAYESIAN_NETWORKS
- https://www.researchgate.net/publication/221232818_Link-sBased_Text_Classification_Using_Bayesian_Networks