Event Classification and Recommendation System

Submitted By: Mounika

College: Vidya Jyothi Institute of Technology

Accuracy: 0.91

F1-Score: 0.90

Date: 22-11-2025

Abstract

This project presents a machine learning–based system designed to classify event descriptions into predefined categories and recommend relevant events based on user interests. A synthetic dataset containing 240 labeled samples across four event types was created for experimentation. The project applies Natural Language Processing (NLP), TF-IDF vectorization, Logistic Regression classification, and cosine similarity to develop both a classification model and recommendation system. The model achieved strong evaluation scores, with an accuracy of 0.91 and a macro F1 score of 0.90. This report discusses dataset design, methodology, preprocessing steps, model training, error analysis, and possible enhancements, providing an end-to-end demonstration of building intelligent text-driven systems.

Table of Contents

1. Introduction

Natural Language Processing (NLP) and Machine Learning (ML) play significant roles in intelligent automation, especially in classification and information retrieval systems. With the rapid increase in event-based platforms such as college portals, ticketing apps, and community calendars, automated categorization and personalized recommendations have become essential for user experience.

This project focuses on two key machine learning tasks:

- **Event Classification:** Predicting whether an event belongs to *Technical*, *Cultural*, *Sports*, or *Devotional* categories based on textual information.
- **Event Recommendation:** Suggesting events that match a user's interests using content-based similarity.

The primary objective is to demonstrate how NLP techniques can be combined with machine learning algorithms to build functional classification and recommendation systems using text data.

2. Literature Background

Text classification and recommendation systems are widely researched fields within machine learning.

- Early classification techniques used **Bag-of-Words (BoW)** models and Naïve Bayes.
- Modern approaches leverage **TF-IDF, Logistic Regression, SVM**, and deep learning.
- Recommendation engines originated in e-commerce and streaming platforms (e.g., Amazon, Netflix) and evolved into two branches:
    - **Collaborative filtering:** Based on user behavior.
    - **Content-based filtering:** Based on item similarity.

In this project, content-based filtering and TF-IDF vectorization were chosen because:

- There is no user interaction data (no ratings or click history).
- Text descriptions are meaningful and can be compared using cosine similarity.

3. Dataset Description

A synthetic dataset containing **240 labeled event entries** was created manually to ensure balanced and controlled experimentation. Four categories were selected because they represent common academic and recreational event types.

Category	Count	Description
Technical	60	Technology-oriented, workshops, coding events, hackathons
Cultural	60	Music, drama, dance, traditions, celebrations
Sports	60	Physical competitions, tournaments, fitness events
Devotional	60	Spiritual gatherings, meditation, bhajans, temple visits

Each row of the dataset consists of:

- **Event Title**
- **Short Description**
- **Category Label**

This design enables supervised learning for classification.

4. Methodology

The workflow followed in this project is:

1. **Dataset Creation**
2. **Data Cleaning and NLP Preprocessing**
3. **Feature Extraction using TF-IDF**
4. **Train-Test Split**
5. **Model Training using Logistic Regression**
6. **Model Evaluation**
7. **Building Recommendation Engine**
8. **Error Analysis and Insights**

This approach ensures modular development and reusable components.

5. Preprocessing

To ensure consistency and readiness for machine learning, the following preprocessing steps were applied:

- Merging text from title and description
- Lowercasing text
- Tokenization
- Removal of stop words (implicitly handled by TF-IDF)
- Vectorization to numeric form

Text normalization helps remove noise and improve machine learning performance.

6. Feature Engineering

Feature engineering was performed using:

TF-IDF Vectorization

TF-IDF assigns importance to terms based on how frequently they appear within a document relative to the entire dataset. This avoids overemphasis on generic words like "event" or "session."

Configuration:

- Maximum features: 2000
- N-gram range: (1, 2)

This improves model performance by capturing contextual patterns.

7. Model Selection and Training

Several traditional ML algorithms are suitable for text classification, including SVM, Naïve Bayes, and Logistic Regression. Logistic Regression was selected due to:

- High accuracy on textual TF-IDF datasets
- Better interpretability
- Faster training and inference times

Training specifications:

- **80% training / 20% testing**
- **Stratified sampling**

8. Evaluation and Results

The model achieved:

Metric	Value
Accuracy	0.91
Macro F1 Score	0.90

These results indicate strong performance across all categories, with consistent prediction quality.

The classification report and confusion matrix reveal:

- Most predictions align correctly with true labels
- Minor confusion between culturally similar event descriptions

9. Recommendation System

A content-based recommendation system was developed to identify relevant events based on user-provided keywords.

Steps:

1. Convert user query into TF-IDF vector
2. Compute cosine similarity with all event vectors
3. Sort results in descending similarity order
4. Return top relevant matches

This approach allows personalized suggestions even for new users.

10. Error Analysis

Most misclassifications occurred when:

- Descriptions overlapped categories (e.g., competitions appear in both Technical and Cultural contexts)
- Short descriptions lacked unique keywords
- TF-IDF failed to recognize semantic meaning beyond keywords

11. Discussion

The system demonstrates practical results using simple, interpretable models. The recommendation engine is flexible and reusable in real-world applications such as:

- College event platforms
- Ticketing apps
- Community bulletin boards

12. Future Enhancements

Future improvements may include:

- Transformer-based embeddings (BERT, RoBERTa)
- Hybrid recommendation systems (content + collaborative filtering)
- Sentiment extraction in descriptions
- Deployment using a web UI

13. Conclusion

This project successfully demonstrated the creation of a complete NLP-based classification and recommendation system using Python and machine learning techniques. The system shows strong performance and provides meaningful recommendations, proving the effectiveness of TF-IDF and Logistic Regression for structured text applications.

14. References

- Scikit-Learn Documentation
- TF-IDF Research Publications
- Python Natural Language Toolkit Guides
- Machine Learning Text Classification Benchmarks

15. Appendix

- Sample classification output screenshots
- Confusion matrix heatmap
- Top recommended events per query