Movie Recommendation System

Internship Project Report

---

1. Introduction

In the current digital age, users are overwhelmed by the vast array of choices available across streaming platforms. A movie recommendation system enhances the user experience by suggesting films that match individual preferences. This project focuses on building a content-based recommendation engine that suggests movies based on similarity in genre and plot description.

---

2. Abstract

This project implements a content-based filtering approach using textual metadata of movies such as genres and plot overviews. The core idea is to analyze the descriptive attributes of a film and compare them with others using Natural Language Processing (NLP) techniques. Using vectorization and cosine similarity, the system ranks movies based on how similar they are to a selected title. The project is developed using Python and leverages libraries such as Scikit-learn and Pandas. Final deliverables include a recommendation function and serialized data models for scalability.

---

3. Tools and Technologies Used

Tool / Library	Purpose
Python	Core programming language
Pandas	Data handling and manipulation
Scikit-learn	Text vectorization and similarity computation
CountVectorizer	Text feature extraction (Bag-of-Words model)
Cosine Similarity	Calculating pairwise similarity scores
Pickle	Model serialization and storage
Jupyter Notebook	Development and interactive analysis environment

---

4. Steps Involved in Building the Project

4.1 Data Loading and Preprocessing

- The dataset `New Movies Dataset.csv` was loaded using Pandas.
- Missing values were checked and irrelevant columns were removed.
- Key columns (`title`, `overview`, `genre`) were selected for feature engineering.

4.2 Feature Engineering

- Combined `overview` and `genre` into a single text feature called `tags`.
- This new feature provides a richer representation of movie content.

4.3 Text Vectorization

- The `tags` column was converted into numeric form using `CountVectorizer`.
- A vocabulary of the top 10,000 terms was created excluding English stopwords.

4.4 Similarity Computation

- Cosine similarity was used to measure how close movies are in the vector space.
- This enabled the identification of the most similar movies to a given title.

4.5 Recommendation Logic

- A function `recommand(movie_name)` was developed.
- For a given movie, it returns the top 5 most similar movie recommendations.

4.6 Model Serialization

- Processed data and similarity matrices were saved using Pickle for future reuse.

---

5. Conclusion

The content-based movie recommendation system successfully suggests relevant films based on the user's input. By leveraging text data and similarity metrics, it enables efficient discovery of similar content without requiring explicit user ratings. This foundational system can be extended with collaborative filtering, sentiment analysis, or deployment as a web application using Streamlit or Flask.
