Movie Recommendation Application

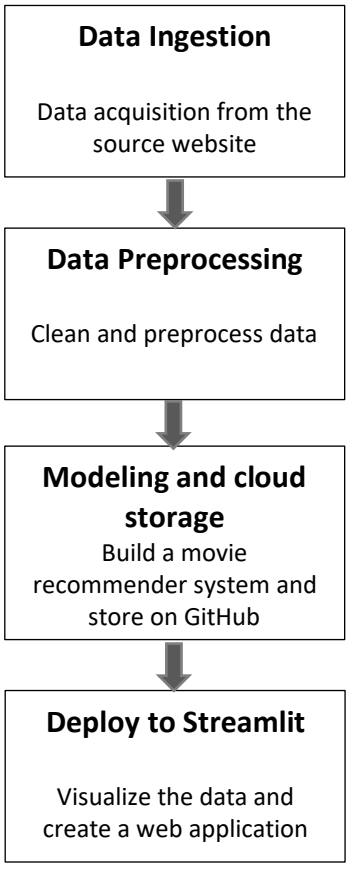
Project Write-up by: Himani Kaushik

Abstract

A client wants to launch a new web-based application for recommending movies to its subscribers. The purpose of the model is to recommend similar movies based on the content. This model will apply the text transformation and modeling techniques to create a movie recommender system and then deploy the entire system using a web application. The data was obtained from the Kaggle website. The dataset has details for 45,000 movies, but only a sample was used. The data was cleaned and preprocessed. TfidfVectorizer was used to transform text to feature vectors. Cosine similarity matrix was created using these vectors to measure similarity between the vectors. This matrix was used to recommend movies based on their title, genre and director. The movie recommender was deployed as a web-based application using streamlit. The output of the system gives five movie recommendations based on the input provided by the user.

Design

The project is designed around the four components as illustrated in the following data pipeline. The data is extracted from the website by downloading the required files and later stored in the pandas dataframe. The data is cleaned and prepared for the modeling. The movie recommender system is created based on the title, genre or director inputs provided by the user. The concept of feature vectors and cosine similarity is leveraged for the recommender system. The streamlit app is built and deployed through GitHub.



Data

The data was obtained from the kaggle website - https://www.kaggle.com/datasets/rounakbanik/the-movies-dataset . The dataset has details for 45,000 movies, that includes cast, crew, plot keywords, budget, revenue, posters, release dates, languages, production companies, countries, vote counts and vote averages. A subsample of the data was used to ease uploading on the GitHub. The movies that had more than 600 votes were used in the sample.

The data was cleaned and preprocessed. I used TfidfVectorizer to transform text to feature vectors. Cosine similarity matrix was created using these vectors to measure similarity between the vectors. This matrix was used to recommend movies based on their title, genre and director.

Algorithms

- Data preprocessing:
    - Cleaning
    - Custom fine-tuning
    - Vectorization (TfidfVectorizer)
    - Cosine similarity matrix
- Recommender Model:
    - Custom functions using cosine similarity matrix to return 5 similar movies
    - Choice of three inputs provided to the user
- GitHub repository to store data and outputs
- Data Visualization:
    - Streamlit app deployed through GitHub

Tools

- Pandas and Numpy: For cleaning data and preprocessing.
- NLTK, scikit-learn: For preprocessing and modeling data.
- Streamlit: For creating the web application.
- GitHub: For storing data and deploying streamlit app.

Communication

The following examples are from the streamlit app:

Movies Recommendation System

Select Criteria for Recommendation

Title

Select

Toy Story

Recommend

Top Five Recommended Movies:

Toy Story 2

Toy Story 3

A Bug's Life

Cars

Cars 2

Movies Recommendation System

Select Criteria for Recommendation

Genre

Select

Drama Crime

Recommend

Top Five Recommended Movies:

GoodFellas

The Departed

Raging Bull

Shutter Island

Taxi Driver

Movies Recommendation System

Select Criteria for Recommendation

Director

Select

ChrisNoonan

Recommend

Top Five Recommended Movies:

The Simpsons Movie

Happy Feet

Mr. Deeds

The Angry Birds Movie

The Lion King 1¬Ω