

BUDT737: Enterprise Cloud Computing and Big Data

Project Report

Project Title: CinFind: Movie Recommendation Tool

Team Members:

- Balaji Udayakumar
- Siddharth Khare
- Arhab Hasan Khan

Group Number: 07

Table of Contents:

1. Executive Summary
2. Our Dataset
3. Sample Dataset
4. Research Questions
5. Methodology
6. Results and Findings
7. Conclusions

ORIGINAL WORK STATEMENT

We the undersigned certify that the actual composition of this proposal was done by us and is original work.

	Typed Name	Signature
	Balaji Udayakumar	Balaji Udayakumar
	Siddharth Khare	Siddharth Khare
	Arhab Hasan Khan	Arhab Hasan Khan

1) Executive Summary

Our research project aims to create a movie recommendation tool that enhances user satisfaction by providing personalized movie suggestions. In other words, our goal here is to create a system capable of recommending movies based on their unique features.

Key Findings

Efficient Tool: Our team successfully built an efficient recommendation engine that leverages key big data technologies like web scraping, relational databases, machine learning, data pre-processing, and web application deployment.

Seamless Web Interface: We deployed our tool on a web application via Streamlit, granting any user easy access. The interface provides a smooth user experience.

Novel Approach: Rather than generic suggestions, our tool considers specific movie features, such as genre, cast, and user preferences.

2) Our Dataset

- **Data source:**

The movie's dataset has been sourced from Rotten Tomatoes. Rotten Tomatoes is a review-aggregation website for film and television. The dataset was obtained through web scraping using BeautifulSoup and subsequently stored in a database. Our analysis involves storing, pre-processing, and training models on this data.

- The dataset (after it has been scraped and pre-processed) has been stored in the DB and contains the following columns which will be of use to us:
 - A. id (char): This column specified the unique identification given to each movie. It is in char format.
 - B. Title (char): The column stores the title of a movie. It is a variable of type char.
 - C. genre1 (char): The genre column specifies the first category or type of a movie (e.g., action, drama, sci-fi). It is a categorical variable.
 - D. genre2 (char): The genre column specifies the second category or type of a movie (e.g., action, drama, sci-fi). It is a categorical variable.

- E. actor1 (char): The actor column specifies the first protagonist of a movie. It is stored as char.
- F. actor2 (char): The actor column specifies the second protagonist of a movie. It is a char variable.
- G. rating (char): The rating column specifies the rating of a movie for appropriate audiences. Example: PG-1, PG-18 etc.
- H. Tomatometer (Numerical): This metric is the score which has been given to a movie by Rotten Tomatoes. It shows how well a film was received by professional critics.
- I. Audience Score (Numerical): This metric is the score which has been given to movies by the viewers and audience.

Subsequently, the data is fetched from the database, and the 'char' data type columns are converted into numerical form using StringIndexing and then converted into vectors using OneHotEncoding.

- A. New Genre1 (double): This column contains genres that have been processed and standardized after pre-processing the raw dataset. This new genre is selected by picking the first of the two genres of the total genre.
- B. New Genre2 (double): This column contains genres that have been processed and standardized after pre-processing the raw dataset. This new genre is selected by picking the next most suitable genre of the total genres.
- C. New Actor1 (double): This column contains actors that have been processed and standardized after pre-processing the raw dataset. This new protagonist is selected by picking the first two actors of the total genre.
- D. New Actor2 (double): This column contains actors that have been processed and standardized after pre-processing the raw dataset. This new protagonist is selected by picking the next most suitable actor of the total actors in that particular movie.

- E. New Rating (double): This column contains the new rating that has been processed and standardized after pre-processing the raw dataset. This new rating is in double format.
- F. One-Hot Encoded Genre1 (Vector): This column represents the first genre information in a one-hot encoded format. Each value corresponds to a binary value (0 or 1).
- G. One-Hot Encoded Genre2 (Vector): This column represents the second genre information in a one-hot encoded format. Each value corresponds to a binary value (0 or 1).
- H. One-Hot Encoded Actor1 (Vector): This column represents the first actor information in a one-hot encoded format. Each value corresponds to a binary value (0 or 1).
- I. One-Hot Encoded Actor2 (Vector): This column represents the second actor information in a one-hot encoded format. Each value corresponds to a binary value (0 or 1).
- J. One-Hot Encoded Rating (Vector): This column contains the one-hot encoded ratings. Again, each value corresponds to a binary value (0 or 1).
- K. Features: This column contains output values from the VectorAssembler, which is used for performing clustering.
- L. Prediction: This column is the prediction values which has been generated by our predictive model.

- **Final Columns (sample):**

title: "Inception"
tomatometer: 87
audience_score: 91
genre1: Sci_Fi,
genre2: Action
actor1: Leonardo DiCaprio
actor2: Cillian Murphy
Rating: PG-13
new_Genre1: 2.0
new_Genre2: 1.0
new_Actor1: 5.0
new_Actor2: 7.0
new_rating: 2.0
One_Hot_genre1: (7,[2],[2.0])
One_Hot_genre2: (7,[3],[1.0])
One_Hot_actor1: (5,[2],[5.0])
One_Hot_actor2: (5,[3],[7.0])
features: (9,[0,1,4],[83.0,90.0,1.0])
prediction: 3

In the above example, it is visible that the movie Inception has been given a rating of 87 by RottenTomatoes, and 91 by the audience. It's genre is science fiction & action (rightly-so) and the new genre is given as Sci_Fi. The prediction score will arrange movies fetched from similar genres or actors and make clusters. When a user searches a movie via the search box on our web application, they will get the three lists of movies (basically three clusters) that are most similar to the movie they have specified.

- **Why is this data of interest?**

Movie ratings and genres (especially Rotten Tomatoes) are of significant interest to both viewers and filmmakers. Understanding critical and audience reception helps guide movie choices for people looking to watch similar kinds of movies.

As part of our research project, we conducted web scraping to collect relevant movie data from Rotten Tomatoes. This process involved extracting information such as movie titles, ratings, and genres. We have filled a form on their website that allows students to use their datasets for academic research purposes. We intend to ensure that the data collection process respects Rotten Tomatoes' terms of use and copyright policies.

3) Research Questions

Some of the research questions that we aim to answer in this research project are as follows:

1. What Factors Influence Movie Ratings?

We want to understand the key factors that contribute to movie ratings. Are critical reviews (Tomatometer scores) more influential, or do audience preferences (audience scores) play a much more significant role?

2. Genre-Based Recommendations:

How do different genres affect movie ratings? Are certain genres consistently well-received by critics and audiences? We'll try to map out a correlation to see which genres have more high-rated movies in the dataset we have sourced.

3. Personalization and User Preferences:

Can we create personalized movie recommendations based on individual preferences? By considering a movie's features like genre, titles, actors, etc., we'll explore how to tailor recommendations to enhance user satisfaction.

4. Predictive Modeling:

Can we build a predictive model that estimates a movie's rating based on its features (such as genre, artists, and title)? Our goal is to create an accurate recommendation system that assists users in selecting movies aligned with their preferences.

5. Business Implications:

How can streaming platforms leverage this data to enhance user engagement? We'll discuss strategies for content targeting, personalized recommendations, and movie libraries. This will also help in understanding how cinema-aggregation websites can benefit from this to reach the maximum amount of customer viewership.

4) Methodology

Web Scraping (BeautifulSoup):

We used web scraping with BeautifulSoup to extract movie data from Rotten Tomatoes. Web scraping allowed us to collect a diverse dataset directly from the Rotten Tomatoes website. It provided real-world movie ratings, genres, and other relevant information.

SQLite Database:

We stored the scraped data in an SQLite database named “movies.db.”. SQLite is suitable for small-scale projects. It allowed us to organize and manage the data efficiently.

Exploratory Data Analysis (EDA):

Before preprocessing the data, we performed preliminary EDA to understand its characteristics, and distributions better. It also helped us in the next machine learning modeling decisions.

Data Preprocessing and feature engineering:

We cleaned and transformed the raw data. Preprocessing involved removing N/A values, converting numerical values to float, converting movie titles to lowercase, imputing with mean, clustering, and plotting data consistency, and handling missing values. Additionally, we performed string indexing and one-hot encoding on the movie genre column to better fit them into the training and testing datasets for our machine learning model. Through this feature engineering process, we aim to improve the model accuracy by capturing relevant information.

Clustering:

We applied clustering to group movies based on similar features (like: genre, ratings). Clustering will allow us to discover underlying structures into movie categorization.

Predictive Modeling (K-Means Clustering):

We have used the K-Means clustering algorithm as the predictive model. This model is useful for predicting numerical values, which is the “prediction” column in our case. After doing a 70-30 split on the train-test data, we have evaluated the accuracy using RMSE (Root Mean Squared-Error).

Web Application Deployment (Streamlit and localtunnel):

After our model was completed, we have broadcasted our recommender system on a web-application using streamlit and localtunnel. This will allow users to search for movie recommendations in a seamless and efficient way. We have talked more about this in the presentation.

5) Results and Findings

In our research project, we explored various research questions related to personalized movie recommendations. Here are some of the findings:

Factors Influencing Movie Ratings:

Both critical reviews (Tomatometer scores) and audience preferences (audience scores) significantly impact movie ratings. Movies with high Tomatometer scores tend to receive positive audience ratings, but exceptions exist.

Genre-Based Recommendations:

On average, genres like drama and sci-fi have high-rated movies in our dataset. However, this finding is inconclusive as this was based only on the data which we have scraped from the RottenTomatoes website.

Predictive Modeling:

Firstly, we have built a predictive model (Random Forest) to predict the movies based on the user input. Features like genre, cast, and movie rating can contribute to accurate predictions.

However, on further testing, we found that the model was not accurately predicting the movies. Therefore, we switched to most popular unsupervised ML model i.e. K-means clustering. By using K-means ML model, we are able to predict good results which will help users toward relevant content based on their preferences (like watching a similar kind of movie they've found interesting).

Business Implications:

Streaming platforms and movie-aggregation websites can use this web application for content targeting and personalized recommendations. By understanding which movies their audience like, they can take actions and steps that will lead to customer retention and engagement.

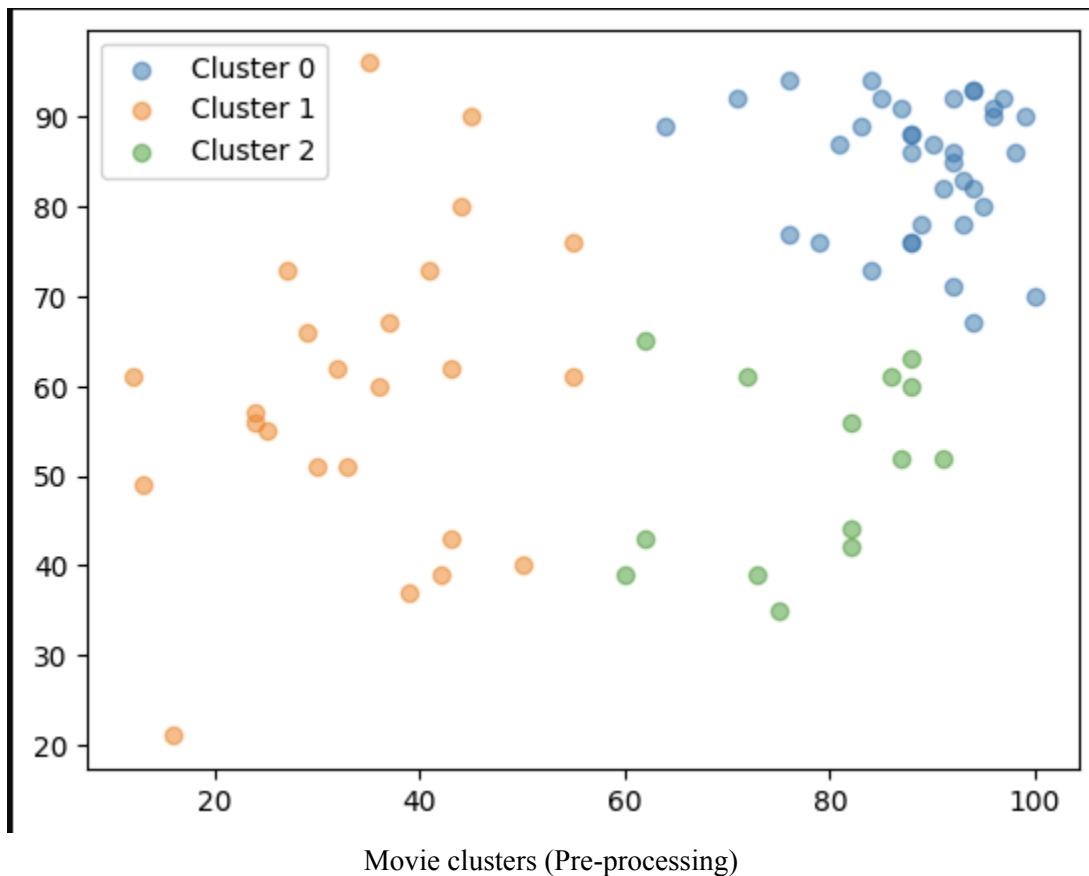
6) Conclusion

In this research project, we explored movie ratings, genres, and personalized recommendations. We created a predictive model using cloud computing technical stack including web scraping, relational databases, machine learning, data pre-processing, and web application deployment. We were able to see that features like genre, cast, and title contribute to robust predictions.

Our work sheds light on critical factors influencing movie reception, genre trends, and the usefulness of predictive modeling.

III. Appendix:

Visualizations:





Web Application snippet:

Web Application can be accessed by:

URL: <https://giant-queens-bow.loca.lt/>

Password: 35.203.136.187

NOTE: This URL is dynamic and changes whenever the submitted Python Jupyter notebook runs.

 RUNNING... Stop 

CinFind - Movie Recommeder

Enter a movie name:

Fetch Info

Initial Page of the Web Application

CinFind - Movie Recommender

Enter a movie name:

Dune

Fetch Info

Rotten Tomatoes Score for 'Dune': 93

Audience Rotten Tomatoes Score for 'Dune': 95

Actors for Dune : ['Timothée Chalamet', 'Zendaya', 'Rebecca Ferguson', 'Javier Bardem', 'Josh Brolin']

Genre for Dune : ['Sci-Fi', 'Adventure', 'Action', 'Fantasy', 'Drama']

Recommended Movies are :

	Set 1	Set 2	Set 3
0	synchronic	voyagers	the phantom
1	godzilla: city on the edge of battle	the mother	spirited away
2	asteroid city	daniel	big fish & begonia
3	the matrix	the bricklayer	guy ritchie's the covenant
4	ferrari	the last airbender	avatar

Search results for “Dune”

References:

1. Rottentomatoes-python Library · PyPI
<https://pypi.org/project/rottentomatoes-python/>
2. Localtunnel for Web Application Hosting
<https://github.com/localtunnel/localtunnel>
<https://localtunnel.github.io/www/>
3. “Machine Learning based Movie Recommendation System”
<https://hcis-journal.springeropen.com/articles/10.1186/s13673-018-0161-6>
4. “An Efficient movie recommendation algorithm based on improved k-clique methods”
https://www.academia.edu/51025889/A_Research_Paper_on_Machine_Learning_based_Movie_Recommendation_System
5. Other referenced links:
https://www.reddit.com/r/webdev/comments/4649rw/rotten_tomatoes_api/
https://www.rottentomatoes.com/browse/movies_at_home/genres

—end report—