# Seminar/Case Study

**Final Report**

*Prioritized Movie Recommendation*

*Based on Content and Content Creators*

***By***

*Gali Ravi Praveen*

*190C2030015*

## Supervisor

*Dr. Kiran Khatter*

Associate Professor



**Department of Computer Science and Engineering School of Engineering and Technology**

*December ,2021*

# Acknowledgement

I would like to express my sincere gratitude to my supervisor Dr. Kiran Khatter who gave me the golden opportunity to do this wonderful project on the topic “*Prioritized Movie Recommendation Based on Content and Content Creators*”, which also helped me in doing a lot of research in the fields which I am new like Data Science, Knowledge Graphs and Natural Language Processing.

Thanking You.

Gali Ravi Praveen, 190C2030015

2019

12/12/2021



**BML Munjal University, Gurgaon, Haryana**

# CANDIDATE’S DECLARATION

I, Gali Ravi Praveen, hereby declare that the work done in my seminar/case study/project entitled *"Prioritized Movie Recommendation Based on Content and Content Creators*" in fulfillment of completion of 5thsemester of Bachelor of Technology (B.Tech) program in the Department of Computer Science and Engineering, BML Munjal University is an authentic record of our original work carried out under the guidance of *Dr. Kiran Khatter, Associate Professor.*

Due acknowledgements have been made in the text of the project to all other materials used.

This seminar/case study/project was done in the full compliance with the requirements and constraints of the prescribed curriculum.

Gali Ravi Praveen, 190C2030015, A picture containing text

Description automatically generated

Place:

Sri Kalahasthi, Andhra Pradesh

Date:

12/12/2021

## CERTIFICATE

This is to certify that the Seminar/Case Study/Project entitled “*Prioritized Movie Recommendation Based on Content and Content Creators*” to the best of my knowledge is a record of the bonafide work carried out by Mr. *Gali Ravi Praveen* under my guidance and/or supervision. The contents embodied in this report, to the best of my knowledge, have not been submitted anywhere else in any form for the award of any other degree or diploma. Indebtedness to other works/publications has been duly acknowledged at relevant places. The work was carried out during *July -December*

*2021* as part of their 5th semester coursework for Bachelor of Technology (B.Tech) program in the Department of Computer Science and Engineering, BML Munjal University*.*

Name and Designation of the Supervisor: Signature:

Date:

Place:

## Contents

1. Abstract
2. Problem Definition
3. Objectives
   1. Requirements
4. Challenges
5. Deliverables
   1. Image / Screenshots of WEB Interface
6. Description of the Dataset
   1. Image / Screenshots of Database
7. Proposed Methodology
8. Experimental Results
9. Conclusion
10. References
11. **Abstract**

The main aim of this project is to recommend the best movies that gives the same vibe as the movie the user likes. The movie that user likes will be taken as the input and the best movies will be shown to the user from the database which are similar to the input movie, this process happens using knowledge graphs and natural language processing. In this process the recommendation happens based on content and content creators i.e., genre of the movie, the main plot of the movie, directors, and writers of the movie.

1. **Problem Definition**

The problem that this project trying to solve is to recommend the accurate movies to the persons who are interested to watch the movies that are similar to the movies that they liked. So, based on these liked movies the suitable movies will be recommended to the users using an interface like web in which all the information about the movie is also shown.

1. **Objectives**

To make a project which has access to movies data and can perform various operations on the data to recommend the user best suited movies based on the one’s he liked based on content and content creators.

1. ***Requirements***
2. The dataset that contains all the movies along with some information about the movie like genre, ratings, directors, and writers.
3. Best database which uses knowledge graph as a data structure to retrieve data fast based on the relations between the data.
4. Selecting the suitable web framework which can communicate with graph database and with the web program.
5. Best natural language processing library which can generate contextual embeddings of a word.
6. A GPU to make the debugging process fast.
7. **Challenges**

The main challenge in this project is to get data of the movies that is sufficient to make accurate recommendations.

The ways to get this data are:

1. Scraping the web to get each movie and its corresponding data. I tested with Wikipedia site to scrape the data but most of the movie’s data is impossible to get because of inconsistence position and tags used in the site.
2. Using IMDB datasets. I downloaded the IMDb dataset which has all the data of the movies since 1911 to 2020. But the main problem in this method is that the data is inconsistent. Data cleaning is needed to this process. Also, all the necessary data about the content creators are not found in this dataset.
3. The other method is to use the IMDB library which is available to python to get the necessary data of the movie using movie’s IMDB id. This id can be found from the previous method.
4. The last and useful method that worked is that combining the methods 2 and 3. The ids from the second method can be used to retrieve the data from the IMDB library which is third method.

The other challenge is to insert this huge data to the Neo4j database, which uses knowledge graphs as it’s data structure, because it’s time consuming. But it is possible, if necessary, time is given. Similar challenges that take time in the knowledge graph are to create nodes for the cast, directors, and writers. Also, to create relations from movie to the director as “directed”, to writer as “written” and to the cast member as “acted-in”.

1. **Deliverables**

The main deliverables of this project are:

1. Dump file of the neo4j database so that the database can be created using this dump file any time.
2. The whole project file that contains necessary files to run this we application.
3. ***Image / Screenshots of WEB Interface***

* ***Home Page***

***Graphical user interface, website

Description automatically generated***

* ***Results Page***

***A picture containing text, monitor, indoor, screen

Description automatically generated***

* ***Movie Info Page***

***A screenshot of a video game

Description automatically generated***

1. **Description of the Dataset**

The dataset is generated using the IMDB ids from the IMDB dataset and the other necessary information in the dataset is generated using the IMDB library using python programing Language. This dataset is converted into a knowledge graph using the Neo4j database.

This database contains 1,288,566 nodes and 3,348,593 relations. The nodes contain Movie nodes, Director nodes, Writer nodes, Cast nodes, Year of Release nodes and Genre nodes. The relations contain directed relation, written relation, acted-in relation, genre-type relation, released-in relation.

Each movie node contains properties like year, rating, duration, plot, voting’s, genre, movie poster and some other data which is not used in the project. Similarly, director, writer, actor also has properties like name, biography, image of themselves.

1. ***Image / Screenshots of Database***

* ***Movie Nodes***

A screenshot of a computer

Description automatically generated with medium confidence

* ***Director Nodes***

A screenshot of a computer screen

Description automatically generated with medium confidence

* ***Writer Nodes***

A screenshot of a computer

Description automatically generated with medium confidence

* ***Relations***

A picture containing diagram

Description automatically generated

1. **Proposed Methodology**

The best way to get the movies that are similar to the input movie is using the genre and plot of the movie but to connect emotionally with the film the user must feel the way the movie created so I included the content creators while filtering the movies from the database. The content creators that I chose are directors and writers of the film.

First the directors and writers of the input movie is found and then the other creators with the same personality of the input movie creators are found. This can be achieved matching their personality using the percentage of the films that the director has directed in the genre. If the percentage of this genre trait is equal or close to the other director, we can say that the thinking process or knowledge on that genre of the director is same or similar. Based on this idea the same personality directors will be selected. Also same happens with writers as well.

After selecting the same personality creators, a list of movies is made directed by the same personality directors and written by same personality writers. In this list, movies are again filtered based on their ratings and the genre related to the input movie.

Finally, from this final list of movies, plot of each movie is taken and converted it into a vector using Bert library, which can convert a word or a sentence into a vector. Same done with the input movie’s plot as well. Using those vector cosine similarity is found for all the movies in the list with the input movie’s plot. If the value is closer to 1 then it is similar. So, this list is sorted based on this value in the descending order.

These final movies after sorting is shown to the user. Therefore, it is “**Prioritized Movie Recommendation Based on Content and Content Creators**”.

1. **Experimental Results**

The result of the project is currently shown using a web interface, which using TensorFlow in background to find cosine similarity at the last step of the process. This library uses GPU to calculate the vectors of each sentence, this is where I am facing an issue because of my GPU memory which is of 4GB. More is needed to calculate vectors for the entire final list of the movies to manage this problem I am reducing the number of similar personality directors and writers and taking only few movies in the list to find their corresponding cosine similarity values.

But it is possible to maximize the results but not in the web interface instead in text.

1. **Conclusion**

To recommend movies based on other movie as an input to qualify the content of the input movie in all the ways possible. To make this happen this project tried it’s best for the user to experience most from the data available the movies on the internet from 1911 to 2020.

1. **References**

For miner error in the libraries which I am using are solved by looking some posts in this website: <https://stackoverflow.com/>

To retrieve data from the IMDB library I used Google Colab to run the code also to run faster:

<https://research.google.com/colaboratory/>

To make the insertion and making relations between nodes in the database is only possible now because of using indexing in neo4j. I raised a query in the community for this and I got a reply to my query: <https://community.neo4j.com/>

I used Bert library to generate vectors from the sentences to calculate the cosine similarity values of the movie plots: <https://github.com/google-research/bert>

I used the ids of the movie from the dataset that I got from IMDB from here: <https://www.imdb.com/interfaces/>

**Plagiarism Report** *(please contact the Library for the credentials of Turnitin)*