Movie Recommendation with Sentiment Analysis

CSE3013 - EPJ - Artificial Intelligence

Review - 3 Report

Slot: C2+TC2

Team Members

Kulvir Singh (19BCE2074)
Hritam Dutta (19BCT0121)
Vansh Kumar Kaushik (19BCT0115)
Aditya Pradhan (19BCT0155)
Anumula Vignaya Virajith (19BDS0116)

Submitted To

Prof. Sunil Kumar P.V.
Asst. Professor Sr. Grade 1
School of Computer Science (SCOPE)
VIT University, Vellore

Acknowledgement

We would like to express our gratitude to our mentor cum faculty, **Dr Sunil Kumar P.V.**, who gave us the golden opportunity to do this interestingly unique project on the topic **Movie Recommendation along with Sentiment Analysis.**

We did a substantial amount of research and were thereby educated about many new things in different fields. We are really thankful to him for his valuable inputs and expert guidance. We would also like to thank each of our teammates who have constantly supported each other during the entire course of the project, especially during such times the whole world is suffering from COVID-19, which led to the timely completion of our product.

Table Of Contents

1. Abstract

In today's time, there is an increasing number of people viewing movies on a web-based platform and the platforms include content from various demographics. This project aims to make the online content-consuming experience catered towards each user by simplifying their movie selection process. Initially, the system can take the types/genres preference & other parameters as input from the user and progressively go on to take user ratings as an important attribute for recommending curated content. A recommendation algorithm will be made entertaining the same.

Furthermore, a critique system will be made to perform sentiment analysis for the user to get a better understanding of the recommendations of the movie. The progressive personalisation of the user experience by the system can be a good business tool for organisations aiming to develop a lasting customer base across the world. The customers/users benefit from avoiding a search through thousands of movies to find something they might like.

2. Aim

The aim of this project work is to build a movie recommendation system using various machine learning and artificial intelligence concepts. Moreover, the aim is to focus on the high accuracy of the recommender. Apart from this, the brand new feature of this recommender system would be a Critique System that classifies the comments and reviews of the movie as a positive or negative review using neural networks.

3. Objectives

- To enable the user to get recommended movies and showcase personalised content and to prevent the user from watching a movie that does not meet its expectations.
- To enable the user to understand the positive aspects and negative aspects of the movie in one go.
- It is a platform that is easily accessible and simple and gives authenticated ratings, displays top-rated movies.
- To have a higher accuracy at predicting the movie recommendations as compared to other movie recommenders.
- To maintain a highly accurate comment classifier that enables the user to view the positive and negative comments.

4. Scope

The Movie Recommendation system aims to create a website where you can find movies filtered out from a large database, based on reviews they have received while keeping in mind the input preferences provided by the user. It is designed to enable users to give reviews/ratings for the movie and analyses users' film preferences and suggestions. This system is designed to be accessible and easy to use for all age groups and all the way from casual movie-goers to die-hard movie enthusiasts.

5. Introduction

The Movie Recommendation along with the Sentiment Analysis project work paves way for an intelligent system that recommends the movies based on the input given to the function. This feature is also supported by the sentiment analysis of the comments and reviews of the same movie that is searched by a user and classifies them as positive or negative [8]. This makes it easy for the system to provide an overview of the movie given as input to the system and increases the accuracy of the intelligent models. Our project is supported by a beautifully designed web application to support the AI and Machine Learning code. This 'flask' web app will act as the interface for the user making the client side of the project easily accessible, memorable and fun to use while the server-side will handle the responses, training and retraining of the database and datasets.

6. Literature Review

6.1 Movie Recommendation Framework based on Neural Networks

It is vital to blend display information with implicit information and to dig out prospective information in order to deliver more accurate and steady recommendations. Existing techniques only examine explicit or implicit feedback information unilaterally, ignoring the potential information of explicit and implicit feedback information, which is also important for the accuracy of the recommendation system. The authors present a new approach [1] to address the potential information acquisition problem from various feedback, in which they consider explicit and implicit feedback information to predict the prospective preferences of consumers and the possible features of the product. Matrix factorization (MF) and a deep neural network (DNN) is then used to develop separate feature embeddings that completely account for the user-linear objects and non-linear features. The model was put to the test on a variety of real-world data sets, and it dramatically improved the hit ratio.

6.2 Deep Learning using Rectified Linear Units (ReLU)

The use of rectified linear units (ReLU) as the classification function in a deep neural network (DNN) is introduced in this paper [2]. In a deep neural network, ReLU is commonly employed as an activation function for the hidden layers. This is done by multiplying the activation of the penultimate layer in a neural network by weight parameters (θ) to obtain the raw scores (RS). The raw scores are then thresholded by 0, i.e. f(o) = max(0, RS), where f(o) is the ReLU function. Simply put, it outputs 0 when x is zero and a linear function when x is greater than zero (refer to Figure 1 for visual representation).

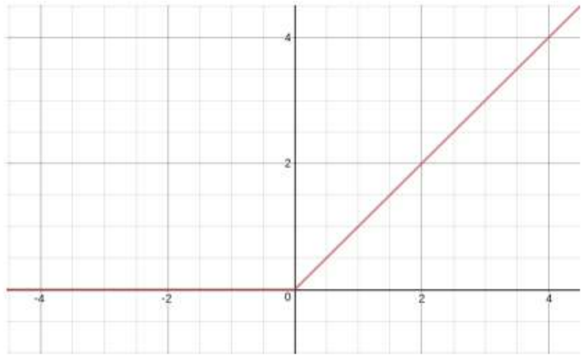


Figure 1: The Rectified Linear Unit (ReLU) activation function produces 0 as an output when $x < 0$, and then produces a linear with slope of 1 when $x > 0$.

6.3 Probabilistic Matrix Factorization for Automated Machine Learning

Modern machine learning algorithms necessitate extensive data pre-processing and hyperparameter optimization in order to attain cutting-edge performance. Furthermore, with an ever-increasing volume of machine learning models being created, model selection is becoming extremely critical. In this study [3], the authors have suggested merging principles from collaborative filtering and Bayesian optimization to address this meta-learning issue. To be more specific, the authors have employed a probabilistic matrix factorization model to transfer knowledge between trials performed on hundreds of distinct datasets, as well as an acquisition function to drive the exploration of the space of feasible pipelines. In their studies, the authors demonstrate that their approach rapidly discovers high-performing pipelines across a broad range of datasets, exceeding the existing state-of-the-art. The authors compared their method against the state-of-the-art using a wide variety of OpenML datasets with varying sample sizes, number of classes, and number of features. Overall, the findings reveal that the suggested method beats both the state-of-the-art and a set of strong baselines.

6.4 Learning User Profiles for Content-Based Filtering in e-Commerce

This paper [4] describes a personalization component that employs supervised machine learning to train a classifier capable of distinguishing between appealing and unappealing objects for the user. The prototype system utilizes a content-based technique that employs textual annotations which typically describe the products sold by e-commerce sites. The experimental results illustrate the method's efficacy and urge the prototype's inclusion in the COGITO project's personalization module, which seeks to improve consumer-supplier relationships in forthcoming e-commerce through the use of modern technologies. This paper examined a basic technique for building user profiles for a content-based book recommender system that focuses on the Naive Bayes machine learning model. The authors demonstrated Item Recommender, a prototype system capable of strengthening user profiles by appending a list of terms to each book category selected by a specific user. The goal was to incorporate the prototype into an already functioning customization system, the Profile Extractor, which uses machine learning techniques to deduce a user's chosen book categories and store them in a user profile.

6.5 Using Content-Based Filtering for Recommendation

Information filtering deals with the delivery of information chosen from a huge collection that the user is likely to find interesting or useful. Recommender systems are a special type of information filtering system. In this paper [5], a user model is induced based on training data, allowing the filtering system to classify unseen things into a positive class c (relevant to the user) or a negative class c (not relevant to the user) (irrelevant to the user). The training set is made up of the items that piqued the user's interest.

In contrast to a collaborative filtering system, which selects items based on the correlation between persons with similar preferences, a content-based filtering system selects items based on the correlation between the content of the items and the user's priorities. PRES is a content-based filtering system. It produces recommendations by comparing the content of each document in the collection to a user profile. A set of phrases can be used to represent the content of a document. Term extraction from documents is accomplished through a series of parsing procedures.

6.6 Semantic Cosine Similarity

In information retrieval cosine similarity is a commonly used metric. This metric models a text as a vector of terms, with the cosine value between the term vectors determining the similarity between two texts. This paper [6] deals with the improvement of cosine similarity between two-term vectors. This modification takes into account not only ordinary vector operations but also the semantic relationship between the dimensions of the vector. It includes semantic checking between dimensions of two-term vectors. This checking uses one of the semantic word similarity methods. The goal of this method is to improve the similarity between two-term vectors. It contains semantic relation between their dimensions against different syntax. This research aims to make the result value be more reasonable than typical cosine similarity measurement, based on human judgment.

6.7 Explaining Recurrent Neural Network Predictions in Sentiment Analysis

In this paper [7], the authors extend the usage of LRP to recurrent neural networks. This work provides a unique propagation rule applicable to multiplicative connections as they arise in recurrent network architectures like LSTMs and GRUs. Further, the authors use the extended LRP approach to a bidirectional LSTM trained on a five-class sentiment prediction task. It allows us to produce trustworthy explanations of which words are responsible for attributing sentiment in individual texts, compared to the explanations derived using a gradient-based approach. This method is deterministic and can be computed in one pass through the network, unlike other non-gradient based explanation methods that rely on random sampling or iterative representation occlusion. Furthermore, this strategy is self-contained in the sense that it does not require the training of an external classifier to supply the explanations, instead, the explanations are gained directly through the original classifier.

6.8 Sentiment Analysis Algorithms and Applications: A Survey

This survey [8] investigates and briefly presents a number of recently proposed algorithm upgrades as well as numerous SA applications. These articles are categorized according to their contributions to the various SA techniques. The related fields to SA (transfer learning, emotion detection, and building resources) that attracted researchers recently are discussed. The major goal of this survey is to provide a near-complete picture of SA techniques and related topics, along with some brief information. The advanced categorizations of a large number of recent publications and the explanation of the contemporary trend of research in sentiment analysis and related areas are the key contributions of this study.

6.9 Movie Recommendation System using Clustering & Pattern Recognition Network

This paper [9] deals with a particular type of movie recommender system which involves the use of a K-means clustering algorithm supported by a pattern recognition technique. This also throws light on the different ways and applications of movie recommendations. We got an overview of the techniques and their accuracy. This paper gave us an insight into recommendation systems. However the gap analysis of this paper brings out the complex technique which does not yield optimal results, hence we took inspiration from the paper to develop a better technique.

6.10 Movie Recommendations from User Ratings

Based on the hetrec2011-movielens-2k dataset, this study [10] shows how to use k-means clustering and softmax regression classification for movie recommendation. The Root-Means-Square Error (RMSE) is used to assess the accuracy of the results. The ability to perform reasonably has been demonstrated. The paper discusses a similar technique of K-means clustering for movie recommendation. However, this technique did not yield optimal results as compared to other methods. Hence analysing the gaps of this technique we decide to use direction cosine over the k-means clustering for better accuracy.

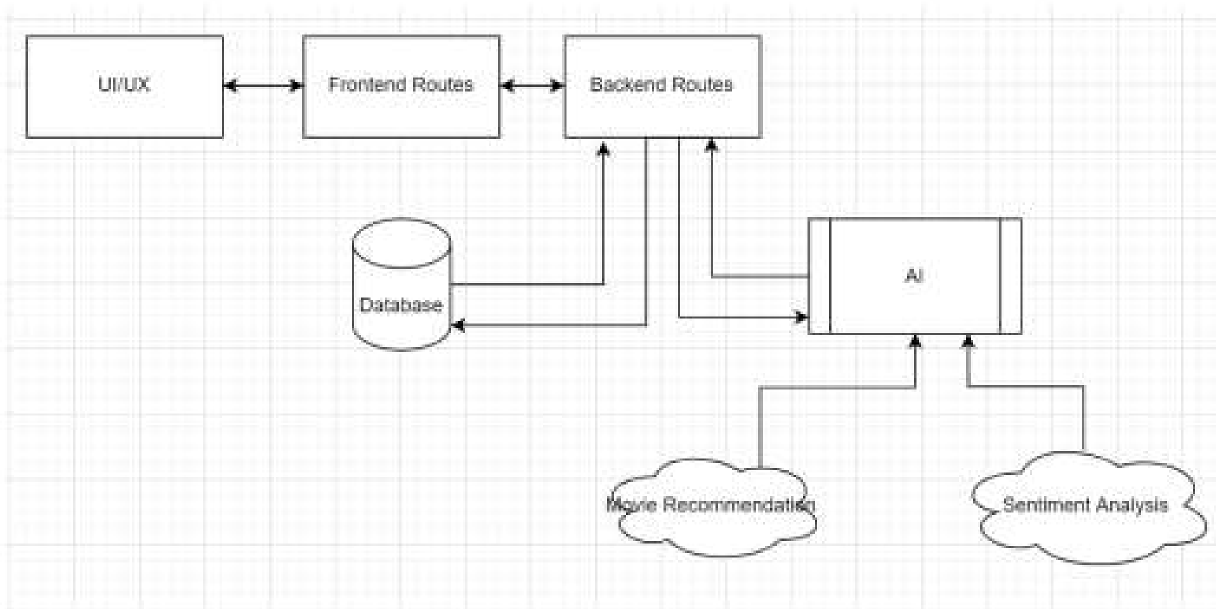
7. Description of Novel AI ideas Used

The current recommendation systems available in the market have relatively less accuracy as compared to what our project will deliver. The use of cosine similarity makes sure that the accuracy of recommendations increases. The use of sentiment analysis to classify the review comments of various users gives this project a unique identity.

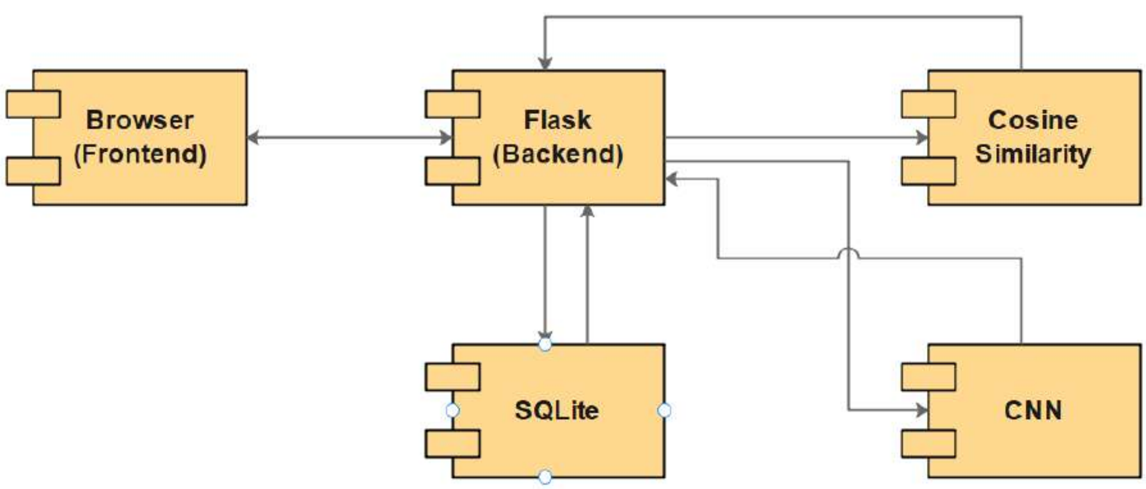
- Sentiment analyzers are not present in many Recommendation Systems.
- The majority of such systems have lesser accuracy.
- Users have no classification of review.
- Using neural networks that are capable of retraining themselves.
- A Validation dataset that we have incorporated will help in retraining the module using back-propagation.
- Cosine similarity.
- Probabilistic function which will classify according to the percentage if it's positive or negative.

8. Implementation

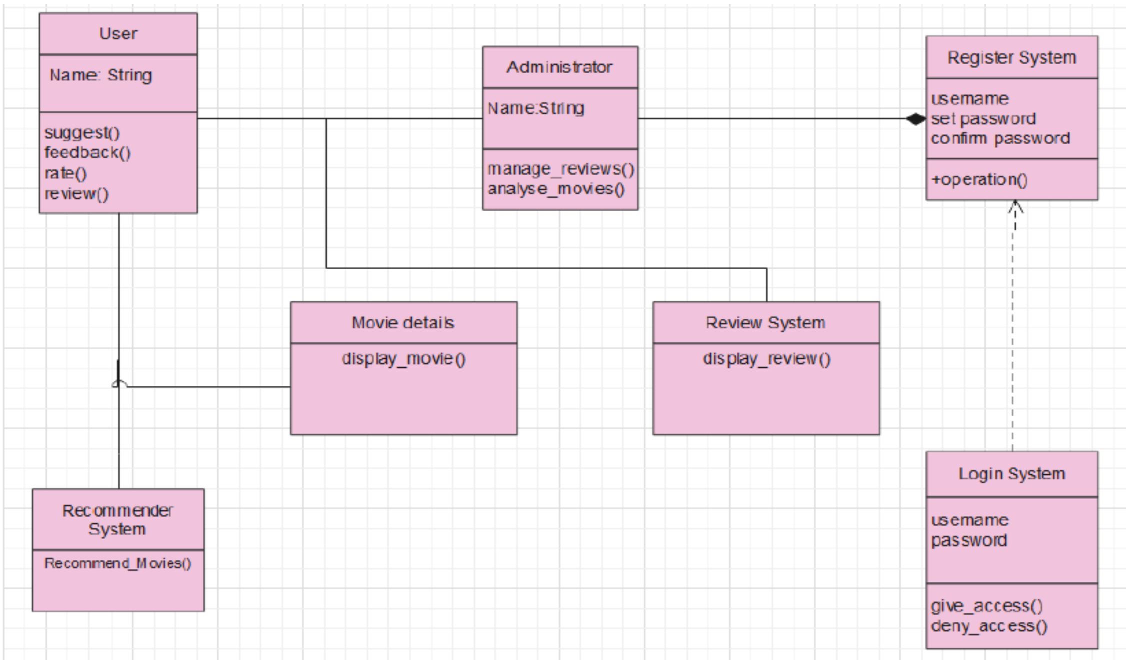
8.1 Workflow Diagram



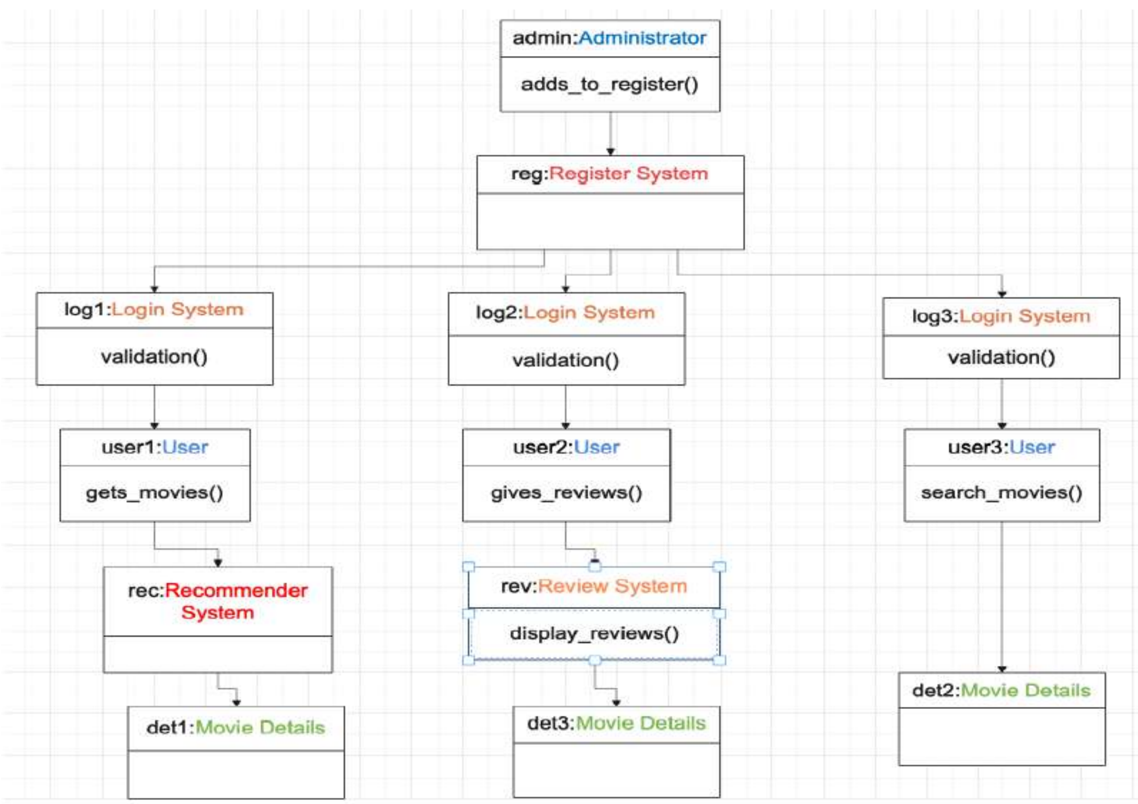
8.1 Component Diagram



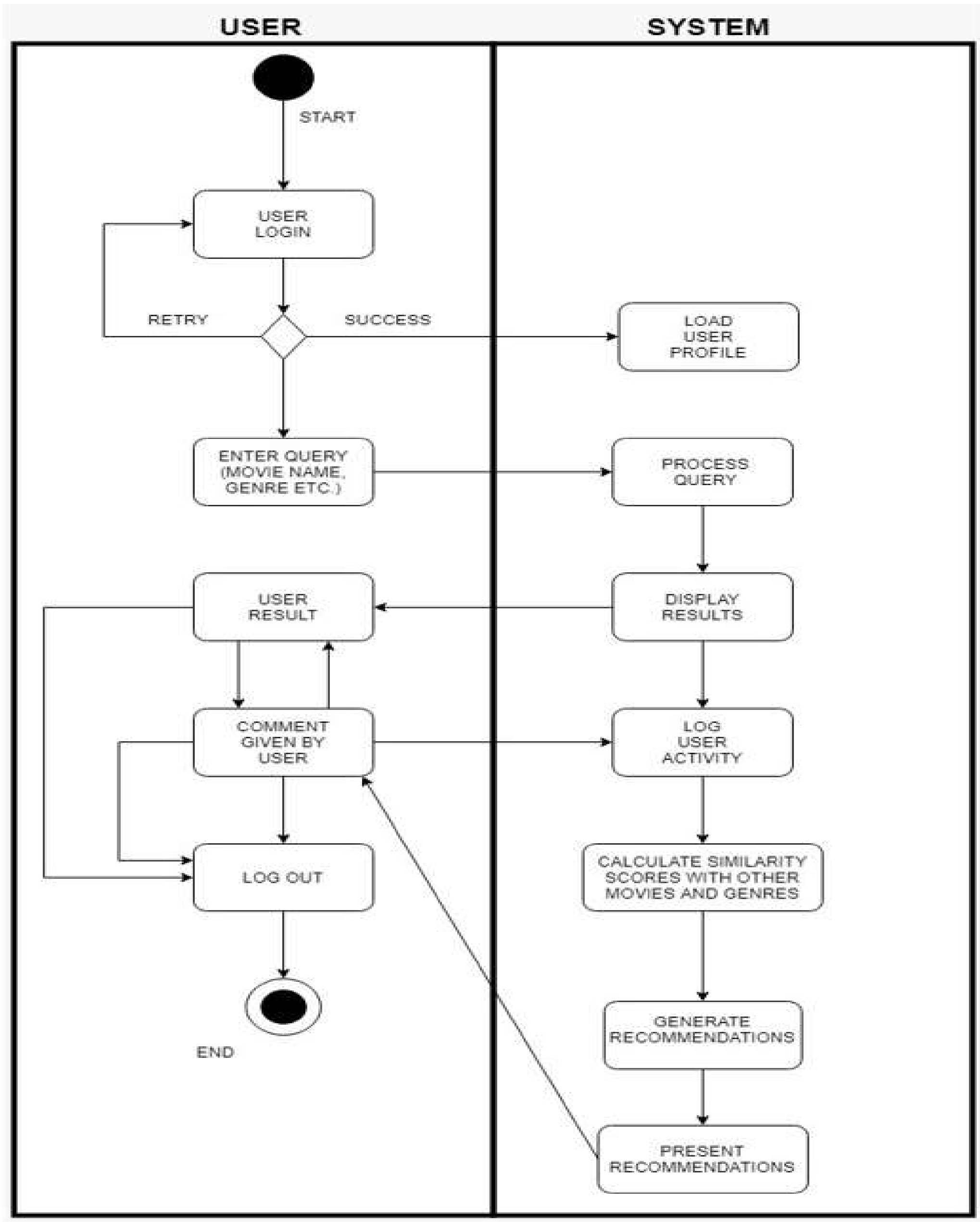
8.3 Class Diagram



8.4 Object Diagram



8.4 Sequence Diagram



9. Modules & their Description

Database

The system will have a database that will feature as a part of the backend of the project. This will deal with the creation of a dynamic schema that will be used for the registration of users. The same data will be required for authentication hence a proper key association would be required. The entire database would be normalized and configured for the project.

Frontend

AI projects generally do not pay much attention to the user interface. This is the reason why we are giving a separate module dedicated to making our project user friendly. A proper frontend design for the system will be made. The pages will be designed using proper styling and will help make the session of the user interactive [5]. This module will give special importance to the ease of accessibility of our tool. This module is important as it makes our project stand out with respect to other recommendation projects in the market.

Backend

This module deals with the structure and flow control of the application. The web application will contain various routes that will be handled in this section. This module will have the control to create, read, update and delete from the database. It will also shift control from the frontend to the AI modules. Apart from this, it will fetch the required output that our AI modules will produce. Hence, it has a major functionality and an efficient low-cost system is required as the backend acts as a gateway between the frontend and AI modules.

Recommendation System

The recommendation system module [9, 10] is the first major deliverable of the project. This module will contain the code that will produce highly accurate recommendations based on user input. This module will be connected to the backend module. The algorithm we will be using will be a relatively new feature of the movie recommendation system. The module consists of an in-depth content filtering mechanism [4] supported by the cosine similarity featuring count vectorization [6].

```python
def get_recommendations(title):
    cosine_sim = cosine_similarity(count_matrix, count_matrix)
    idx = indices[title]
    sim_scores = list(enumerate(cosine_sim[idx]))
    sim_scores = sorted(sim_scores, key=lambda x: x[1], reverse=True)
    sim_scores = sim_scores[1:11]
    movie_indices = [i[0] for i in sim_scores]
    tit = df2['title'].iloc[movie_indices]
    mov_id = df2['id'].iloc[movie_indices]
    return_df = pd.DataFrame(columns=['Title', 'Movie ID'])
    return_df['Title'] = tit
    return_df['Movie ID'] = mov_id
    return return_df
```

Recommendation System Code

Sentiment Analyzer

This module deals with the implementation of classification of users' sentiment based on their review comments about the movie. A 20-dimensional neural network will be implemented using

TensorFlow Keras for creating the sentiment analyzer. The neural network [1, 7] will train the data set in the form of layers. The ReLu function [2] will train the intermediate layer and the sigmoid activation function [1] will be used for the final output layer. The final output layer will hence become a probabilistic function [3] which ensures maximum accuracy for classification.[An additional operation of tuning the parameters for artificial neural networks will also be employed to get the best possible results.

```
tfds.load(name='imdb_reviews')#load dataset
train_data, validation_data , test_data = tfds.load(name='imdb_reviews',split=['train[:60%]','train[
train_examples_data, train_labels_data = next(iter(train_data.batch(10)))
vector_model = "https://tfhub.dev/google/tf2-preview/gnews-swivel-20dim/1" #Importing Pretrained mod
hub_layer = hub.KerasLayer(vector_model, input_shape=[],dtype=tf.string, trainable=True) #Creating a
ann_model = tf.keras.Sequential();
ann_model.add(hub_layer)#20 dim vector hence 20 neurons
ann_model.add(tf.keras.layers.Dense(16,activation = 'relu'))
ann_model.add(tf.keras.layers.Dense(1,activation = 'sigmoid'))
ann_model.compile(optimizer="adam",loss="binary_crossentropy",metrics=["accuracy"])
ann_model.fit(train_data.shuffle(10000).batch(256),epochs=10,validation_data = validation_data.batch
test_examples_data, test_labels_data = next(iter(test_data.batch(2000))) #Creating Batches of our Te
y_pred = ann_model.predict(test_examples_data) #Predicting the on the Test Data
y_pred = (y_pred > 0.5) #Converting y_pred into 0's and 1's depending on whether it is more than or
y_test = np.reshape(test_labels_data,(-1,1)) #Reshaping the Test Label Data according to our y_pred
print(y_pred)
print(y_test)
```

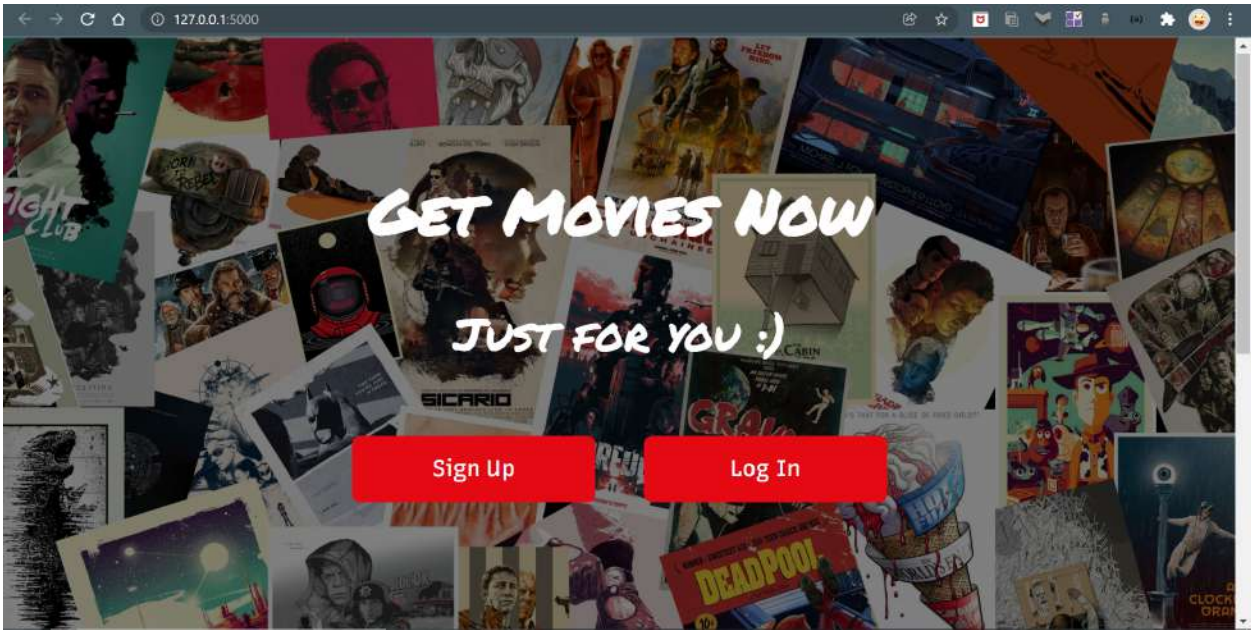
Sentiment Analyser Code

10. Results and Discussion

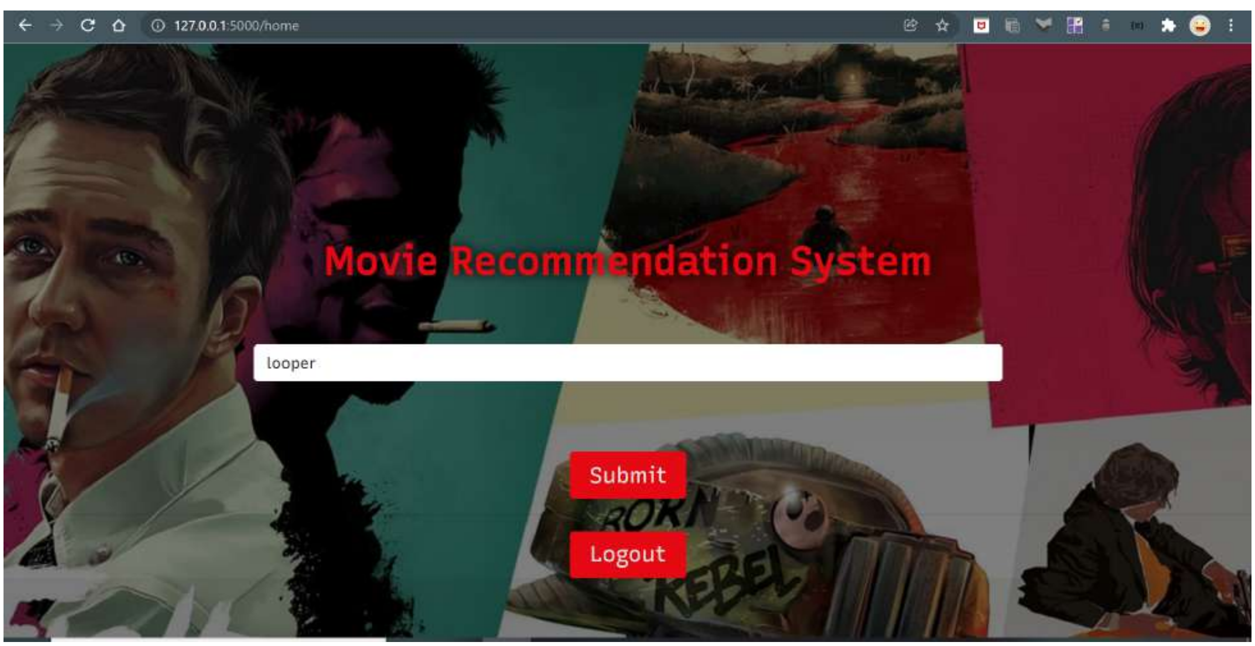
```
43/59 [====================>.........] - ETA: 0s - loss: 0.2559 - accuracy: 0.9032
45/59 [=====================>........] - ETA: 0s - loss: 0.2570 - accuracy: 0.9025
47/59 [======================>.......] - ETA: 0s - loss: 0.2581 - accuracy: 0.9016
49/59 [=======================>......] - ETA: 0s - loss: 0.2579 - accuracy: 0.9014
51/59 [========================>.....] - ETA: 0s - loss: 0.2588 - accuracy: 0.9010
53/59 [=========================>....] - ETA: 0s - loss: 0.2572 - accuracy: 0.9016
56/59 [===========================>..] - ETA: 0s - loss: 0.2571 - accuracy: 0.9016
58/59 [============================>.] - ETA: 0s - loss: 0.2572 - accuracy: 0.9016
59/59 [==============================] - 3s 51ms/step - loss: 0.2569 - accuracy: 0.9016 - val_loss: 0.3265 - val_accuracy: 0.8605
Epoch 10/10

 1/59 [..............................] - ETA: 6s - loss: 0.2573 - accuracy: 0.8984
 3/59 [>.............................] - ETA: 2s - loss: 0.2572 - accuracy: 0.9010
 5/59 [=>............................] - ETA: 2s - loss: 0.2460 - accuracy: 0.9070
 7/59 [==>...........................] - ETA: 2s - loss: 0.2380 - accuracy: 0.9118
 9/59 [===>..........................] - ETA: 2s - loss: 0.2337 - accuracy: 0.9175
11/59 [====>.........................] - ETA: 1s - loss: 0.2370 - accuracy: 0.9126
13/59 [=====>........................] - ETA: 1s - loss: 0.2352 - accuracy: 0.9126
15/59 [======>.......................] - ETA: 1s - loss: 0.2405 - accuracy: 0.9094
17/59 [=======>......................] - ETA: 1s - loss: 0.2423 - accuracy: 0.9097
19/59 [========>.....................] - ETA: 1s - loss: 0.2421 - accuracy: 0.9095
21/59 [=========>....................] - ETA: 1s - loss: 0.2412 - accuracy: 0.9090
23/59 [==========>...................] - ETA: 1s - loss: 0.2377 - accuracy: 0.9103
25/59 [===========>..................] - ETA: 1s - loss: 0.2388 - accuracy: 0.9092
27/59 [============>.................] - ETA: 1s - loss: 0.2373 - accuracy: 0.9106
29/59 [=============>................] - ETA: 1s - loss: 0.2380 - accuracy: 0.9111
31/59 [==============>...............] - ETA: 1s - loss: 0.2389 - accuracy: 0.9102
33/59 [===============>..............] - ETA: 0s - loss: 0.2398 - accuracy: 0.9096
35/59 [================>.............] - ETA: 0s - loss: 0.2389 - accuracy: 0.9096
37/59 [=================>............] - ETA: 0s - loss: 0.2395 - accuracy: 0.9097
39/59 [==================>...........] - ETA: 0s - loss: 0.2391 - accuracy: 0.9094
41/59 [===================>..........] - ETA: 0s - loss: 0.2392 - accuracy: 0.9097
43/59 [====================>.........] - ETA: 0s - loss: 0.2389 - accuracy: 0.9106
45/59 [=====================>........] - ETA: 0s - loss: 0.2383 - accuracy: 0.9104
46/59 [======================>.......] - ETA: 0s - loss: 0.2373 - accuracy: 0.9109
48/59 [=======================>......] - ETA: 0s - loss: 0.2369 - accuracy: 0.9115
50/59 [========================>.....] - ETA: 0s - loss: 0.2362 - accuracy: 0.9118
52/59 [=========================>....] - ETA: 0s - loss: 0.2365 - accuracy: 0.9118
54/59 [==========================>...] - ETA: 0s - loss: 0.2363 - accuracy: 0.9116
56/59 [===========================>..] - ETA: 0s - loss: 0.2344 - accuracy: 0.9127
58/59 [============================>.] - ETA: 0s - loss: 0.2337 - accuracy: 0.9129
59/59 [==============================] - 3s 52ms/step - loss: 0.2336 - accuracy: 0.9129 - val_loss: 0.3196 - val_accuracy: 0.8640
2021-12-06 00:02:25.993783: W tensorflow/core/kernels/data/cache_dataset_ops.cc:768] The calling iterator did not fully read the dataset being cached. In order to avoid une
xpected truncation of the dataset, the partially cached contents of the dataset  will be discarded. This can happen if you have an input pipeline similar to `dataset.cache(
).take(k).repeat()`. You should use `dataset.take(k).cache().repeat()` instead.
WARNING:werkzeug: * Debugger is active!
INFO:werkzeug: * Debugger PIN: 956-558-510
INFO:werkzeug: * Running on http://127.0.0.1:5000/ (Press CTRL+C to quit)
```

Neural Network trained with 10 epochs

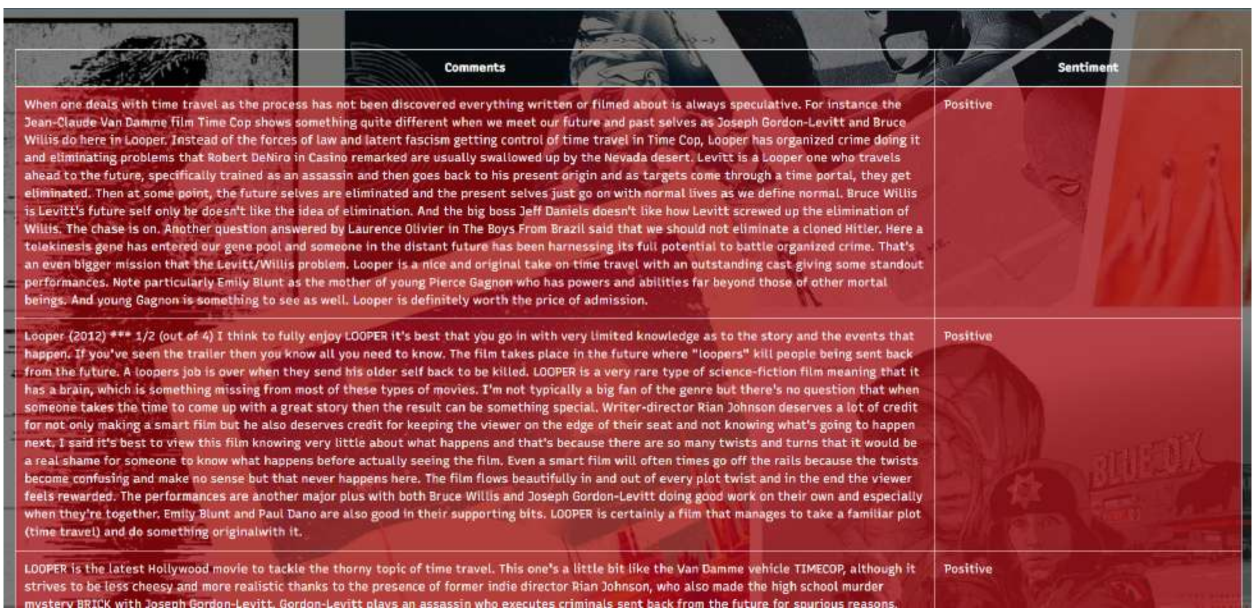


UI/UX Login & Sign Up Page



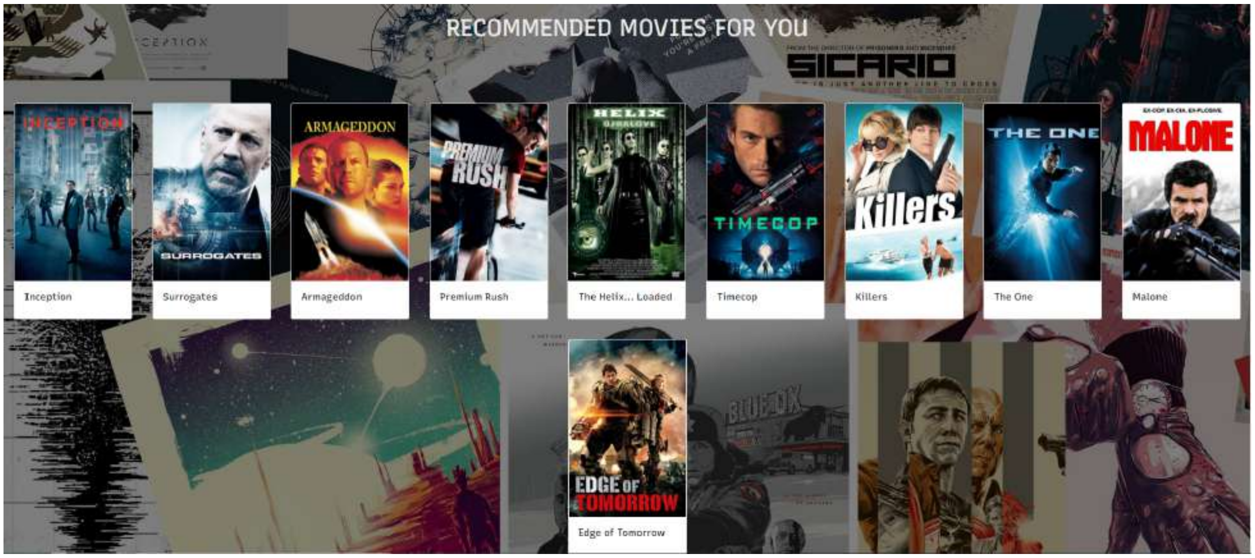
Home Page to enter the Movie Name for Sentiment Analysis & Get Recommendations

Details of the Movie entered by the User



Comments	Sentiment
When one deals with time travel as the process has not been discovered everything written or filmed about is always speculative. For instance the Jean-Claude Van Damme film Time Cop shows something quite different when we meet our future and past selves as Joseph Gordon-Levitt and Bruce Willis do here in Looper. Instead of the forces of law and latent fascism getting control of time travel in Time Cop, Looper has organized crime doing it and eliminating problems that Robert DeNiro in Casino remarked are usually swallowed up by the Nevada desert. Levitt is a Looper one who travels ahead to the future, specifically trained as an assassin and then goes back to his present origin and as targets come through a time portal, they get eliminated. Then at some point, the future selves are eliminated and the present selves just go on with normal lives as we define normal. Bruce Willis is Levitt's future self only he doesn't like the idea of elimination. And the big boss Jeff Daniels doesn't like how Levitt screwed up the elimination of Willis. The chase is on. Another question answered by Laurence Olivier in The Boys From Brazil said that we should not eliminate a cloned Hitler. Here a telekinesis gene has entered our gene pool and someone in the distant future has been harnessing its full potential to battle organized crime. That's an even bigger mission that the Levitt/Willis problem. Looper is a nice and original take on time travel with an outstanding cast giving some standout performances. Note particularly Emily Blunt as the mother of young Pierce Gagnon who has powers and abilities far beyond those of other mortal beings. And young Gagnon is something to see as well. Looper is definitely worth the price of admission.	Positive
Looper (2012) *** 1/2 (out of 4) I think to fully enjoy LOOPER it's best that you go in with very limited knowledge as to the story and the events that happen. If you've seen the trailer then you know all you need to know. The film takes place in the future where "loopers" kill people being sent back from the future. A loopers job is over when they send his older self back to be killed. LOOPER is a very rare type of science-fiction film meaning that it has a brain, which is something missing from most of these types of movies. I'm not typically a big fan of the genre but there's no question that when someone takes the time to come up with a great story then the result can be something special. Writer-director Rian Johnson deserves a lot of credit for not only making a smart film but he also deserves credit for keeping the viewer on the edge of their seat and not knowing what's going to happen next. I said it's best to view this film knowing very little about what happens and that's because there are so many twists and turns that it would be a real shame for someone to know what happens before actually seeing the film. Even a smart film will often times go off the rails because the twists become confusing and make no sense but that never happens here. The film flows beautifully in and out of every plot twist and in the end the viewer feels rewarded. The performances are another major plus with both Bruce Willis and Joseph Gordon-Levitt doing good work on their own and especially when they're together. Emily Blunt and Paul Dano are also good in their supporting bits. LOOPER is certainly a film that manages to take a familiar plot (time travel) and do something originalwith it.	Positive
LOOPER is the latest Hollywood movie to tackle the thorny topic of time travel. This one's a little bit like the Van Damme vehicle TIMECOP, although it strives to be less cheesy and more realistic thanks to the presence of former indie director Rian Johnson, who also made the high school murder mystery BRICK with Joseph Gordon-Levitt. Gordon-Levitt plays an assassin who executes criminals sent back from the future for spurious reasons.	Positive

Sentiment Analysis of the Comments & Reviews of other Users Classified as Positive or Negative



Recommendations Generated by the AI Program using Cosine Similarity

A movie critique system has been successfully created where a user can sign in and see if a movie is a hit or a flop, get recommendations based on its preferences, budget, gross income, rating and gamification. Movies are proportioned according to their titles and languages and actors. Recommendations are made after collecting (explicitly or implicitly), processing and analyzing user or item data. However, developing a Recommendation System is a manual and laborious task for programmers, as there is a clear variation of the data available to process. One method to facilitate procurement of needs, and the resulting Recommendation System development, is the definition of a general user and item model that is compatible with any domain. This study describes a systematic review that aims to identify user and item information used in an advanced and implemented Recommendation System. The number of Recommendation Systems with a collaborative filter is low, indicating that user profiles have not been compared or that the user's historical information has been adequately researched. Privacy concerns over user data may explain this result. The ease of access to real data is an important factor in this result, as it has access to great databases such as TMDb and IMDb. We are able to give them their best use and recommendations in visualization. The results presented in this work can be extended to include more information about new case studies with different methods. One major change in computer science, in this case, is the emergence of Big Data, which is made up of very large datasets that have multiple formats in their data and are constantly changing. This can change how user and item data is modelled and, in particular, how information is collected from users and objects in need of engineering.

11. Conclusion

A movie critique system has been successfully created where a user can sign in and see if a movie is a hit or a flop, get recommendations based on its preferences, budget, gross income, rating and gamification. Movies are proportioned according to their titles and languages and actors. Recommendations are made after collecting (explicitly or implicitly), processing and analyzing user or item data. However, developing a Recommendation System is a manual and laborious task for programmers, as there is a clear variation of the data available to process. One method to facilitate procurement of needs, and the resulting Recommendation System development, is the definition of a general user and item model that is compatible with any domain. This study describes a systematic review that aims to identify user and item information used in an advanced and implemented Recommendation System.

The number of Recommendation Systems with a collaborative filter is low, indicating that user profiles have not been compared or that the user's historical information has been adequately researched. Privacy concerns over user data may explain this result. The ease of access to real data is an important factor in this result, as it has access to great databases such as TMDb and IMDB. We are able to give them their best use and recommendations.

12. References

[1]Gong, J., Zhang, X., Li, Q., Wang, C., Song, Y., Zhao, Z., & Wang, S. (2021). A Top-N Movie Recommendation Framework Based on Deep Neural Network with Heterogeneous Modeling. Applied Sciences, 11(16), 7418.

[2] Agarap, A. F. (2018). Deep learning using rectified linear units (relu). arXiv preprint arXiv:1803.08375.

[3]Fusi, N., Sheth, R., & Elibol, M. (2018). Probabilistic matrix factorization for automated machine learning. Advances in neural information processing systems, 31, 3348-3357.

[4] Abbattista, F., Degemmis, M., Fanizzi, N., Licchelli, O., Lops, P., Semeraro, G., & Zambetta, F. (2002, September). Learning user profiles for content-based filtering in e-commerce. In Proceedings AI* AI Workshop su Apprendimento Automatico: Metodi e Applicazioni.

[5] Van Meteren, R., & Van Someren, M. (2000, May). Using content-based filtering for recommendation. In Proceedings of the machine learning in the new information age: MLnet/ECML2000 workshop (Vol. 30, pp. 47-56).

[6] Rahutomo, F., Kitasuka, T., & Aritsugi, M. (2012, October). Semantic cosine similarity. In The 7th International Student Conference on Advanced Science and Technology ICAST (Vol. 4, No. 1, p. 1).

[7] Arras, L., Montavon, G., Müller, K. R., & Samek, W. (2017). Explaining recurrent neural network predictions in sentiment analysis. arXiv preprint arXiv:1706.07206.

[8] Medhat, W., Hassan, A., & Korashy, H. (2014). Sentiment analysis algorithms and applications: A survey. Ain Shams engineering journal, 5(4), 1093-1113.

[9] Ahmed, M., Imtiaz, M. T., & Khan, R. (2018, January). Movie recommendation system using clustering and pattern recognition network. In 2018 IEEE 8th Annual Computing and Communication Workshop and Conference (CCWC) (pp. 143-147). IEEE.

[10] Byström, H. (2013). Movie Recommendations from User Ratings.