

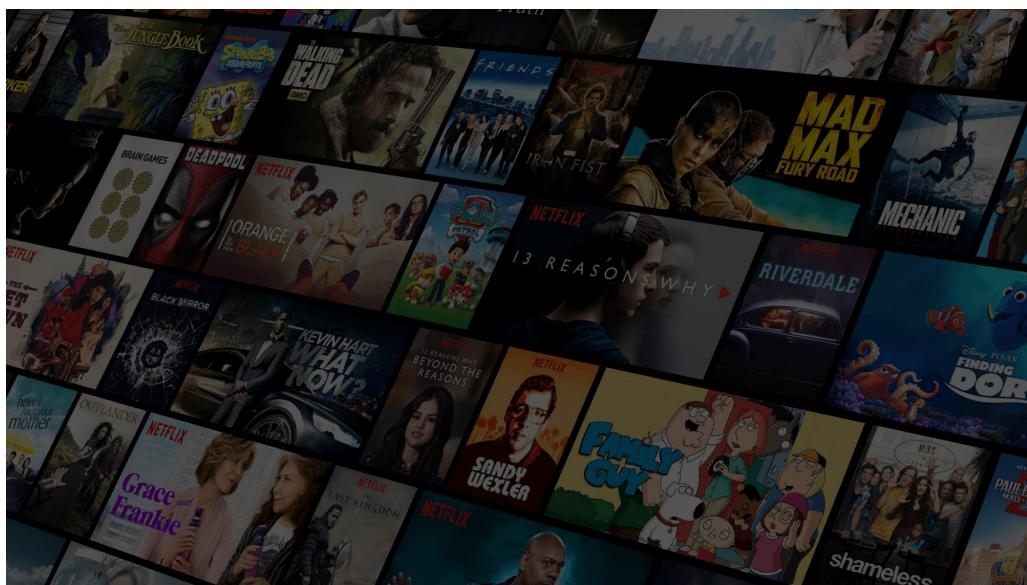
Machine Learning Final Project

Due date: July 1st

The following documentation explains your Machine Learning Final Project which focuses on deploying a machine learning model on **Hugging Face Spaces** for a **movie recommendation task**. Students are required to complete the project **individually or in teams of two**, with no preference given to either option.

Movie Recommender Systems

Movie recommendation systems are an integral part of the entertainment industry, aiming to provide personalized movie suggestions to the users based on their preferences. The project entails analyzing a large dataset of movies, incorporating user preferences, and implementing algorithms to generate relevant movie recommendations.



Dataset

The [Movies Dataset](#) is a collection of data on 45,000 movies and contains information such as budget, revenue, release dates, languages, production countries, and companies for each movie. The dataset also includes ratings of 270,000 users for these movies. The data was obtained from The Movie Database (TMDb) API.

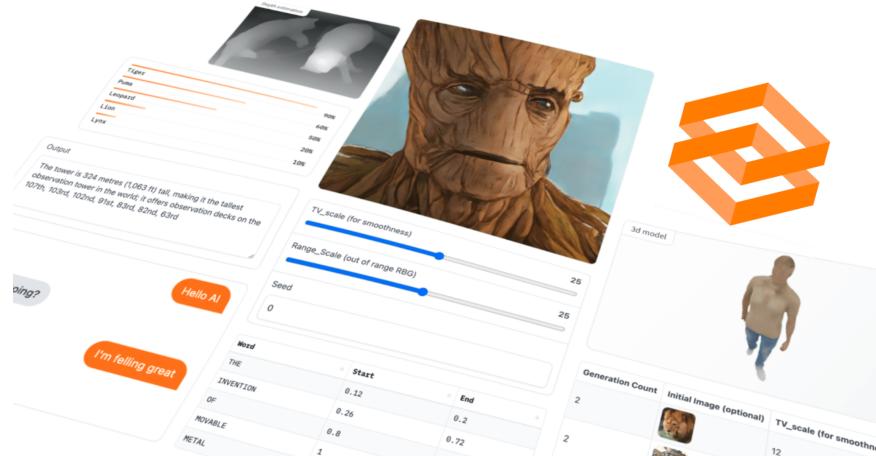
Method

You can employ a wide variety of methods and implement creative ideas you have to fulfill this objective. First, you need to get at least three movies from the user (along with other information

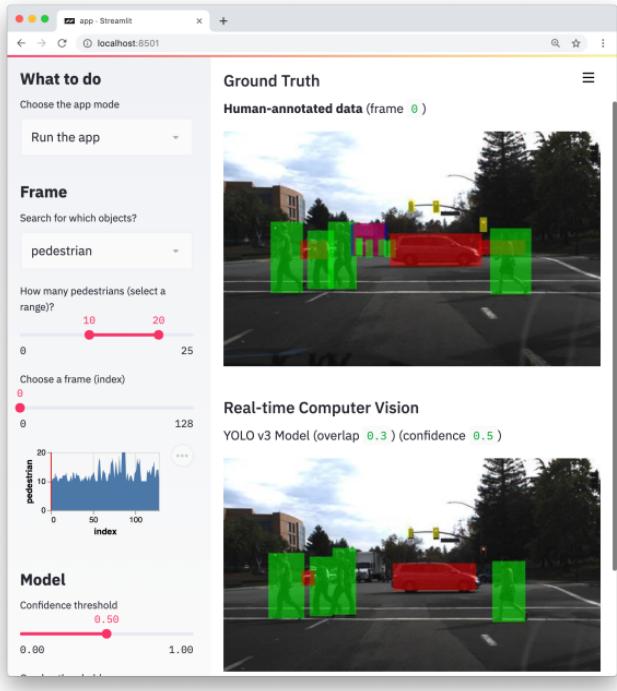
based on your method), then you can employ one of the following approaches to implement your recommender system:

1. **Cluster-Based:** these methods are a popular approach in recommender systems that aim to group users or items into clusters based on their similarities. These clusters can then be used to make recommendations by leveraging the preferences or characteristics of other users or items within the same cluster.
Recommendations can be generated by considering the clusters to which the input movies belong. Alternatively, movies from different clusters can be recommended to introduce new content to the user. Another option is to recommend movies that are most similar to the input movies or those located at the center of the clusters. Additionally, creative approaches can be used to generate recommendations. Extra scores may be granted for creative approaches.
2. **Recommendation-Specific Methods (Extra Point):** it includes collaborative filtering, content-based filtering, and hybrid methods. Collaborative filtering involves recommending items to users based on their similarity to other users' preferences. Content-based filtering, on the other hand, recommends items based on their similarity to the user's previously liked items. Hybrid methods combine both collaborative and content-based filtering approaches to provide more accurate and diverse recommendations. These methods can be implemented as part of your recommender system to enhance its performance and effectiveness. By leveraging these techniques, you can provide personalized and relevant recommendations to users based on their preferences and characteristics. References for these methods are provided in the last part of the documentation for further reading.

User Interface Design



[Gradio](#) and [Streamlit](#) are two popular Python libraries used for designing user interfaces in machine learning projects. These libraries provide intuitive and customizable options for creating interactive interfaces that allow users to interact with machine learning models and visualize results. Both libraries seamlessly integrate with popular machine learning frameworks and offer versatile options for showcasing models and gathering user feedback. You need to employ one of these libraries to implement a user-friendly interface for your system. References for these libraries are provided in the last part of the documentation for further reading.



Model Deployment on HuggingFace

Machine learning model deployment on **HuggingFace Spaces** is a powerful way to make trained models readily available to users. HuggingFace Spaces provides a platform where developers and researchers can showcase their models, allowing others to easily interact with and benefit from their work. You need to deploy your final model and interface into HuggingFace spaces. [References for this section are provided in the last part of the documentation for further reading.](#)



Hugging Face

Report

You are also required to write a detailed report on your implementation and the methods you have used. Explain the method that you have used, the recommender procedure and the logic behind it, and the similarity measures. Include pre-processing steps performed on the data and the explanation of the workflow for designing the interface and deploying to the HuggingFace spaces. At the end of the report, please include a list of all the additional tasks you have completed that were not mandatory but carried extra weight in terms of evaluation.

Extra points

- Extra scores may be granted for more attractive interfaces.
- To enhance the user experience with recommended movies, you have the option to utilize the following APIs that provide additional information about movies, including movie posters. These APIs are compatible with the provided dataset.
 - [OMDb](#)
 - [TMDB](#)
- Using different neural network architectures from plain MLP to more advanced ones like transformers to implement your model. You can also use graph-based recommendation systems, or reinforcement learning approaches in your method.
- Evaluate the proposed method's performance using specific metrics for recommendation systems.

Resources

Recommender systems methods:

<https://medium.com/@eskandar.sahel/recommender-systems-best-practices-a5962ae1b560>

<https://towardsdatascience.com/introduction-to-recommender-systems-6c66cf15ada>

<https://medium.com/recombee-blog/machine-learning-for-recommender-systems-part-1-algorithms-evaluation-and-cold-start-6f696683d0ed>

<https://medium.com/recombee-blog/machine-learning-for-recommender-systems-part-2-deep-recommendation-sequence-prediction-automl-f134bc79d66b>

<https://medium.com/berlin-tech-blog/deep-learning-for-recommender-systems-48c786a20e1a>

<https://heartbeat.comet.ml/recommender-systems-with-python-part-i-content-based-filtering-5df4940bd831>

<https://heartbeat.comet.ml/recommender-systems-with-python-part-ii-collaborative-filtering-k-nearest-neighbors-algorithm-c8dc5fd89b2>

<https://heartbeat.comet.ml/recommender-systems-with-python-part-iii-collaborative-filtering-singular-value-decomposition-5b5dc3f242b>

- Course on Recommender systems

<https://developers.google.com/machine-learning/recommendation>

- Evaluation metrics

<https://towardsdatascience.com/evaluation-metrics-for-recommender-systems-df56c6611093>

- **User Interface Design**

- **Gradio:**

- <https://ismailouahbi.medium.com/gradio-build-deploy-and-share-your-machine-learning-models-7b38babab659c>

- <https://www.youtube.com/watch?v=RiCQzBluTxU>

- <https://www.machinelearningnuggets.com/gradio-tutorial/>

- **Streamlit:**

- <https://www.jstintodata.com/streamlit-python-tutorial/>

- <https://www.youtube.com/watch?v=43RJ3JByygE>

- https://www.youtube.com/watch?v=_9WiB2PDO7k

- **Deployment on HuggingFace**

<https://towardsdatascience.com/host-ml-apps-with-huggingface-spaces-7d5eef00e781>

<https://learnopencv.com/deploy-deep-learning-model-huggingface-spaces/>