

Movie Recommendation System

Welcome to this presentation on building a movie recommendation system using Python. Discover how to gather, clean, analyze, and build accurate movie recommendations.

By :

Rohan Katara (IU2141050069)

Vedant Domadiya (IU2141050039)



Introduction

Why a Movie Recommendation System?

- Welcome to the world of Movie Recommendation Systems!
- Why a Movie Recommendation System?
 - The growing abundance of movies
 - Overwhelming choices for viewers
 - The need for personalized recommendations
- Our journey to build a Python-based movie recommendation system.

Data Collection and Preparation

- The foundation of any recommendation system.
- Data sources for movie information.
- Data preprocessing:
 - Data cleaning
 - Data transformation

Exploratory Data Analysis



Analyzing Movie Genres and Ratings

- Understanding the dataset.
- Analyzing Movie Genres and Ratings:
 - Popular genres
 - Distribution of ratings
- Visualizing Movie Data:
 - Histograms, bar plots, and scatter plots
 - Insights into the data.



Building the Recommendation System

The heart of the presentation.

Collaborative Filtering

- User-based and item-based approaches
- The concept of similarity
- Pros and cons

Content-based Filtering

- Leveraging movie metadata
- Feature extraction
- Pros and cons

Hybrid Models:

- Combining collaborative and content-based filtering
- Strengthening recommendation accuracy

Evaluating and Improving the System

Measuring the system's effectiveness:

Accuracy Metrics

- RMSE (Root Mean Square Error)
- MAE (Mean Absolute Error)

Cross-Validation

- k-fold cross-validation
- Avoiding overfitting



Conclusion

- Summarizing our journey:
 - Data Collection and Preparation
 - Exploratory Data Analysis
 - Building the Recommendation System
 - Evaluating and Improving the System
- The importance of personalization in modern content recommendation.
- The continuous evolution of recommendation systems.
- Thank you for joining us on this journey!

References

Libraries:

- `numpy`
- `pandas`
- `difflib`
- `sklearn.feature_extraction.text.TfidfVectorizer`
- google colab