International Journal of Trend in Scientific Research and Development (IJTSRD)
Special Issue on Smart Innovations in Computer Science and Applications
Available Online: www.ijtsrd.com e-ISSN: 2456 – 6470

Movie Recommendation System Using Python, SQL, and Statistics

Amisha Dixit
Department of Science and Technology, G.H.Raisoni Skill Tech University, Nagpur, India

Abstract
Recommendation systems have become an essential component in various industries such as e-commerce and OTT-platforms. These systems use various algorithms to recommend the most relevant data to the user. Movie recommendation systems, in particular, suggest movies based on the user's interests, thus saving time and effort for the user in searching through a large list of movies to watch. The aim of this project was to develop a movie recommendation system using cosine similarity algorithm. The system is designed to provide personalized movie recommendations based on the user's movie preferences. The project began with data collection from various sources, including movie reviews, ratings, and user preferences. The collected data was preprocessed and transformed into a structured format suitable for analysis. The development of a cosine similarity algorithm comes next. This algorithm is used to compare two sets of vectors. The technique was used to assess how well the films in the dataset fit the user's preferences. The method for proposing films was built using a web-based interface. The interface allows users to enter their film tastes and receive suggestions based on what they say. The suggestions are presented in descending order of similarity, with the most comparable films at the top. Even films that the user has never heard of could be suggested by the system. Giving users a wide range of recommendations that are tailored to their particular preferences is made possible thanks to this capability. In conclusion, the project's goal of developing a cosine similarity-based movie recommendation system was accomplished. Users could easily access and interact with the system because of its web-based interface, and it was quite accurate at providing individualized recommendations.

1. Introduction

Movie recommenders sit at the intersection of **information retrieval**, **statistical learning**, and **product experimentation**. The core empirical problem is to infer a user's preference over a large catalog from sparse signals (ratings, clicks, watch time), then rank items under constraints (latency, diversity, fairness, and policy requirements). The canonical public benchmark for explicit ratings is MovieLens, whose released variants include timestamped user–movie ratings, tags, and (for some variants) coarse demographics.

A rigorous research paper can be organized around a unified, SQL-centered analytics layer (Postgres) fed by reproducible ETL (pandas + SQLAlchemy). This architecture allows you to compare (i) explicit-feedback rating prediction models (RMSE/MAE) that were central in the Netflix Prize, (ii) implicit-feedback ranking models

(Precision@K/Recall@K/NDCG/MAP/AUC), and (iii) sequence models that predict "next watch" from recent sessions.

The literature provides a clear methodological arc: the Netflix Prize popularized **regularized matrix factorization** and time/implicit extensions; the "deep era" introduced neural CF and two-stage candidate-generation/ranking pipelines; and the modern era emphasizes **bias-aware offline evaluation** (selection and exposure bias), fairness/multi-stakeholder metrics, and proper online experimentation.

Open datasets each trade off realism vs. accessibility. Public sources with **per-user viewing logs** and demographics are scarce due to privacy (and famously, de-anonymization concerns in the Netflix Prize). A defensible strategy is to treat explicit ratings (MovieLens / Netflix Prize) as your "ground truth" for model comparison, and then design a production-grade schema that *could* store impressions/clicks/watch-time logs for causal evaluation when such data are available (e.g., internal or partner data).

What follows is a research-paper-ready blueprint: objectives and hypotheses; literature synthesis; dataset recommendations with download links, formats, and licensing constraints; a Postgres schema with ETL patterns; a full modeling + evaluation plan including causal inference and statistical testing; and a 12-week milestone schedule starting **2026-02-20**.

2. Related work

In this part, we are going to look at some of the research that has been done in the field of Deepfake detection & production. Deepfake films like these are becoming more popular on social media networks, prompting the international community to take the threat seriously and, as a result, encouraging academics throughout the globe to build sophisticated deepfake detection tools. It is possible to find a variety of ways in the recent literature.

Unsupervised contrastive learning is used to build a novel deepfake detection algorithm in this study [17]. We first make 2 different versions of an image and feed them into 2 separate networks, one of which is an encoder and the other is a projection head. Maximizing the projection head's outputs' degree of correspondence is how the unsupervised training is accomplished. To test the detection efficiency of our unsupervised technique, we train an efficient linear classification network using the unsupervised features. Unsupervised learning may achieve equivalent recognition efficiency to current advanced supervised algorithms in both inter & Intra database contexts, according to several experiments. In addition, they carry out ablation tests to test the efficacy of our approach.

In this article, [18] CNN facial recognition models, such as Alex Net & Shuffle Net, are applied to distinguish between authentic and fraudulent images of people. Fake/real face recognition collection from Yonsei University's Computational Intelligence Photography Lab is used to evaluate the performance & operation of all separate algorithms After normalizing the images, Error Level Analysis (ELA) is performed before the images are used in a variety of CNN models, which are all unique. Next, the K-NN (K-Nearest Neighbour) & SVM (Support Vector Machine) techniques are used to extract the detailed features from the CNN models an accuracy of 88.2% was found for Shuffle Net's KNN, compared to an accuracy of 86.8% for Alex Net's vector.

In this article [19], multilayer hybrid recurrent DL models for deepfake video detection are presented. Noise- based temporal face convolutional features & temporal learning of hybrid recurrent DL models are used to create the models proposed in this article. These models' performance against stacked recurrent DL models has been shown via experiments. A deep ensemble learning method called DeepfakeStack has been proposed thru [20] to address the issues given by Deepfake multimedia. The proposed method generates a better composite classifier by combining several state-of- the-art classification models based on deep learning. With an accuracy of 99.65 percent and an AUROC score of 1.0, our tests reveal that DeepfakeStack beats the competition in identifying Deepfake. A Real-time Deepfake detector might be built using our technique.

This work [21] provides a method for exposing such fake videos, which makes usage of CNN architecture & Transfer Learning approach. Every video frame is fed into a CNN, which then utilizes the extracted feature information to train an effective binary classifier that can tell the difference between genuine and altered videos. A comprehensive collection of deepfake videos collected from a variety of datasets is used to test the method's accuracy. All of the models had good training accuracy, with each model achieving more than 98%. It was the inceptionV3- based model that had the best accuracy.

In this study [22], the proposed strategy is depending upon utilizing residual noise, which is the difference between the original picture as well as its denoised form. Research of residual noise has shown that it is efficient in deepfake detection due to the unique and discriminative properties that it has, which can be captured successfully by CNNs with TL (Transfer Learning). To test the effectiveness of our technique, we used low-resolution video clips from FaceForensics++ & high-resolution video clips from Kaggle DFDC (Deepfake Detection challenge). When compared to other competing methodologies, the acquired findings demonstrate a high degree of accuracy.

[16] examine deepfake detection algorithms Xception &MobileNet as 2 techniques for classification tasks to repeatedly identify deepfake videos in this research. Four fake video creation techniques & 2 advanced NNs were used to train, test, and compare a total of 8 deepfake video classification models, which were then associated & reviewed. Each model demonstrated adequate classification performance when applied to the corresponding dataset that was utilized in its development. This article makes use of four datasets created using four distinct deepfake technologies that were used in the development of FaceForensics++ to train and evaluate the algorithm. The accuracy of the findings is high across all datasets, with accuracy ranging between 91 and 98 percent depending upon deepfake technologies used. In addition, we built a voting process that can identify fake videos by aggregating results of all 4 approaches, rather than just one.

3. Research Methodology

1. **Quantify interaction structure**: characterize sparsity, temporal drift, and long-tail behavior in movie interactions (ratings or watch events), and how these properties affect model performance and evaluation stability.
2. **Benchmark model families** on a common schema: neighborhood baselines, regularized matrix factorization (MF), SVD++/implicit variants, content-based models, hybrid models (e.g., factorization machines / wide&deep), and sequential recommenders (RNN/Transformer).
3. **Establish evaluation validity**: implement time-aware splits and bias-aware offline evaluation (addressing MNAR/selection bias), then compare conclusions to those from naïve random splits.
4. **Connect offline to online**: specify an experimentation and causal-inference framework—A/B tests as the gold standard, with variance reduction (CUPED), multiple-testing control, and (where applicable) Did/event-study style analyses for algorithm rollouts.
5. **Audit fairness and bias**: quantify popularity bias, calibration (genre distribution match), and group disparities (where demographic features exist), and evaluate mitigation strategies and their trade-offs with accuracy.

Testable hypotheses (examples suitable for formal evaluation)

- **H1 (MF > neighbors on sparse ratings):** Regularized matrix factorization materially improves rating prediction accuracy over neighborhood baselines on large-scale sparse ratings benchmarks.
- **H2 (Implicit features help):** Adding implicit feedback signals (e.g., “items interacted with” even without explicit ratings) improves ranking metrics versus models trained on explicit ratings alone.
- **H3 (Time-aware evaluation changes ranking):** Chronological (time-aware) splits reorder model rankings compared to random splits, indicating temporal leakage under random splitting.

- **H4 (Sequence models win for next-item):** For next-watch / session-based prediction tasks, RNN/Transformer sequence recommenders outperform static MF models when user histories are short and recent context dominates.
- **H5 (Offline metrics are biased without correction):** Offline evaluation on logged interaction data without exposure/selection-bias correction overestimates real-world performance, and IPS-style corrections materially change estimated performance.
- **H6 (Popularity bias–accuracy trade-off):** Interventions that reduce popularity bias or improve calibration (e.g., re-ranking for long-tail or genre calibration) can reduce short-horizon accuracy metrics, implying a Pareto frontier rather than a single optimum.

H7 (Demographic disparity exists where observable): When demographic attributes are available (e.g., in MovieLens 1M/100K), recommendation quality differs across groups unless explicitly constrained/regularized for fairness. **From Netflix Prize to latent factor dominance**

The Netflix Prize introduced a large, timestamped rating dataset and used **RMSE** as the defining leaderboard metric, catalyzing advances in collaborative filtering and ensembling. The "matrix factorization techniques" synthesis describes why latent factor models became central: they capture user and item structure in low-dimensional embeddings, accommodate biases, and can integrate additional signals. The multifaceted Netflix-era models (including neighborhood + factor blends like SVD++) explicitly incorporate implicit feedback and yield significant gains, but also highlight that metric choice (RMSE vs top-K ranking) changes what "good" means.

Deep recommenders and industrial-scale pipelines

Neural collaborative filtering generalizes matrix factorization by replacing inner products with neural interaction functions and was validated on implicit-feedback settings. Industrial systems often use two-stage pipelines: candidate generation over huge corpora followed by ranking; a widely cited example details this pattern and practical lessons in a large-scale video platform recommender. In hybrid modeling, factorization machines provide a principled way to combine sparse features with latent interactions, and wide&deep models formalize the "memorization + generalization" split typical of production systems.

Sequential recommendation

When the target shifts from "predict rating" to "predict next watch," sequence modeling becomes central. Session-based RNN recommenders introduced ranking-optimized losses for short sessions, and Transformer-based approaches (self-attention and bidirectional masking) largely define the modern sequential benchmark space.

Evaluation science: beyond-accuracy and bias-aware offline evaluation

Evaluation is not a footnote: the evaluation decisions (task definition, splitting, metrics, candidate generation) can dominate the conclusion, and evaluation practices can be incomparable across papers if not standardized. Offline evaluation with logged data is often biased because observations are **not missing at random**: users self-select items, and (in online systems) recommenders affect what is even seen. The "recommendations as treatments" framing imports causal-inference tools (propensity scoring / IPS estimators) to debias learning and evaluation. A modern synthesis argues that offline evaluation faces structural challenges, reinforcing that online experiments remain the ultimate arbiter when feasible.

Fairness, calibration, and regulation

Fairness and multi-stakeholder concerns are now core: recommenders impact exposure distribution, long-tail discovery, and group outcomes. Survey work formalizes taxonomies of fairness in recommenders and multi-stakeholder objectives, while additional strands emphasize popularity bias management and calibrated recommendation lists. Policy context increasingly matters: the EU Digital

Services Act introduces transparency obligations regarding recommender system parameters for online platforms, shaping what a "responsible" recommender paper should consider in discussion/limitations. Broader AI governance frameworks emphasize risk management, transparency, and accountability—useful scaffolding for the ethics section of a paper.

4. Research Methodology

A research-grade movie recommender paper usually needs *two* types of data:

6. **User–item interaction data** (ratings or implicit interactions with timestamps) for model training and evaluation.
7. **Item metadata** (genres, cast/crew, text, language, year) for cold-start modeling, content-based baselines, and interpretability/fairness analyses.

Public datasets vary widely in licensing and whether they provide user demographics or true viewing logs. MovieLens variants are the most straightforward for research, while IMDb provides rich metadata but only aggregated ratings (not per-user).

Dataset comparison table

Raw URLs are provided in code formatting for reproducibility.

Dataset	What it contains (relevant fields)	Strengths for research	Formats	Download link(s)	License / constraints (high-level)
Movie Lens 100K	100K user–movie ratings + timestamps; per-user demographics (age, gender, occupation, zip); item file includes genres and IMDb URL; includes predefined train/test split files for 5-fold CV.	Best "starter" benchmark; includes demographics + official CV splits; fast iteration.	TSV-like files; multiple split files.	https://grouplens.org/datasets/movielens/100k/	Research use allowed; no redistribution; no commercial use without permission.
Movie Lens 1M	~1M ratings with	Classic benchmark used in	::-delimited .dat	https://grouplens.org/datasets/movielens/1m/	Research use; no redistribution;

	timestamps; demographics in users.dat (Gender, Age bucket, Occupation, Zip); movies and genres.	many studies; demographics enable fairness/differential-performance audits.	files.		commercial use requires permission.
Movie Lens 20M	20M ratings + tags; **no demographics**; includes links.csv mapping to IMDb and TMDB IDs; includes tag genome.	Scales MF and ranking models; strong for metadata joins through links file.	CSV.	https://grouplens.org/datasets/movielens/20m/	Research use; no redistribution; commercial permission required.
Movie Lens 25M	25M ratings + tags; updated larger catalog; includes tag genome; license similar to other MovieLens packages.	Better modern scale; large enough to stress-test retrieval + ranking pipelines.	CSV.	https://grouplens.org/datasets/movielens/25m/	Research use; no redistribution; commercial permission required.
IMDb Non-Commercial Datasets	Rich canonical metadata (title.basics, title.crew, title.principals,	Best public source for high-quality metadata (cast/crew, genres, runtime); supports	gzippd TSV (UTF-8).	https://datasets.imdbws.com/	Personal, non-commercial use; no scraping; permission can be withdrawn; strong restrictions on

	etc.) + **aggregated** ratings (title.ratings averageRating, numVots). TSV gz, refreshed daily.	content-based features and joins via IMDb IDs.			republishing/repurposing into databases.
TMDB developer API / data	Movie metadata served via API; attribution requirements (logo + required notice). Free for non-commercial use; commercial licensing required otherwise.	Useful metadata enrichment (posters, keywords, collections), but API terms and attribution obligations must be followed.	JSON via HTTP API.	https://developer.themoviedb.org/docs/getting-started	Free for non-commercial with attribution; commercial licensing needed; explicit branding/notice requirements.
Netflix Prize ratings dataset	~100M anonymized movie ratings with dates; designed for RMSE prediction competition.	Historically central benchmark; very large explicit-rating dataset; good for MF/SVD++ scaling.	Custom text format.	https://www.kaggle.com/datasets/netflix-inc/netflix-prize-data	Privacy sensitivity: de-anonymization risk documented; original distribution constraints vary; treat as research-only and follow dataset terms.
Amazon Review Data (Movies/TV	User reviews with star ratings, text, helpfulne	Provides text + implicit signals proxies; enables hybrid	JSON / JSONL.	https://cseweb.ucsd.edu/~jmcauley/datasets/amazon_v2/	Research dataset; follow site's intended academic use and citation guidance.

category)	ss votes; product metadata; also-viewed/bought graphs (depending on release).	recommenders and explanations.			
YouTube-8M	Millions of video IDs with machine-generated labels + audio/visual features; licensed under CC BY 4.0 (dataset).	Not movies/ratings, but useful as an auxiliary "video content" dataset; helps test large-scale retrieval and embedding visualization workflows.	TFRecords; other mirrors include parquet conversions (distribution-dependent).	https://research.google.com/youtube8m/download.html	CC BY 4.0 for dataset as stated on download page; code licensing can differ from dataset.
LFM-1b	Large-scale *music listening* events with timestamps + some demographics in the original paper; **dataset not available for download anymore due to license issues**.	Included here because it is common in recommender research; can inform sequence/log modeling methodology even if not usable now.	N/A (official download unavailable).	https://www.cp.jku.at/datasets/lfm-1b/	Not downloadable; license constraints prevent distribution.

Recommended "backbone configuration" for this paper

- Main benchmark: MovieLens 1M (for demographics + fairness analyses) and MovieLens 20M or 25M (for scale and metadata joins).
- Metadata enrichment: IMDb TSVs (non-commercial) joined through MovieLens links where available.
- Optional large-scale historical baseline: Netflix Prize for MF/SVD++ comparisons and replication of classic results (with strict attention to privacy/terms).

Optional hybrid text-based modeling: Amazon review data (Movies/TV subset), if your paper includes explanations or review-text modeling.

5. Conclusion and Future work

Netflix has given permission to use the dataset for other non commercial research purposes. In particular, the dataset is being used in another data mining contest, the KDD Cup 2007. In this contest participants may address two tasks. The first task is to predict which movies certain individuals would rate in 2006, the year after the training dataset was based. The second task is to predict the total number of ratings certain movies received in 2006 from the Netflix Prize user base. Results from the KDD Cup 2007 competition will be available in late August, 2007.

As part of this study, a unique approach has been developed to reveal AI-generated deepfake video together with powerful feature extraction & classification utilizing customized CNNs. The proposed method exhibits testing accuracy of 91.47%, loss of 0.342, and AUC score of 0.92 despite of training on a small subset of data. The comparative analysis found that the proposed customized CNN outperform two existing methods like CNN and MLP- CNN.

References

1. Aslam, J. A., Pavlu, V., and Yilmaz, E. A statistical method for system evaluation using incomplete judgments. In SIGIR, pp. 541–548, 2006.

2. Bell, R., Koren, Y., and Volinsky, C. Chasing $1,000,000. how we won the netflix progress prize. Statistical Com puting and Graphics, 18(2):4–12, 2007.

3. Hu, Y., Koren, Y., and Volinsky, C. Collaborative filtering for implicit feedback datasets. In ICDM, pp. 263–272, 2008

4. Bengio, Y. Learning deep architectures for ai. Foundations and Trends in Machine Learning, 2(1):1–127, 2009.

5. Cortes, C., Mansour, Y., and Mohri, M. Learning bounds for importance weighting. In NIPS, pp. 442–450, 2010.

6. Dud´ ık, M., Langford, J., and Li, L. Doubly robust policy evaluation and learning. In ICML, pp. 1097–1104, 2011.

7. Lim, D., McAuley, J., and Lanckriet, G. Top-n recommen dation with missing implicit feedback. In RecSys, pp. 309–312, 2015.

8. Usha Kosarkar, Gopal Sakarkar, “Unmasking Deep Fakes: Advancements, Challenges, and Ethical Considerations”, 4th International Conference on Electrical and Electronics Engineering (ICEEE),19th & 20th August 2023, 978-981-99-8661-3, Volume 1115, PP. 249-262, https://doi.org/10.1007/978-981-99-8661-3_19

9. Usha Kosarkar, Gopal Sakarkar, Shilpa Gedam, "Deepfakes, a threat to society", International Journal of Scientific Research in Science and Technology (IJSRST), 13th October 2021, 2395-602X, Volume 9, Issue 6, PP. 1132-1140, https://ijsrst.com/IJSRST219682