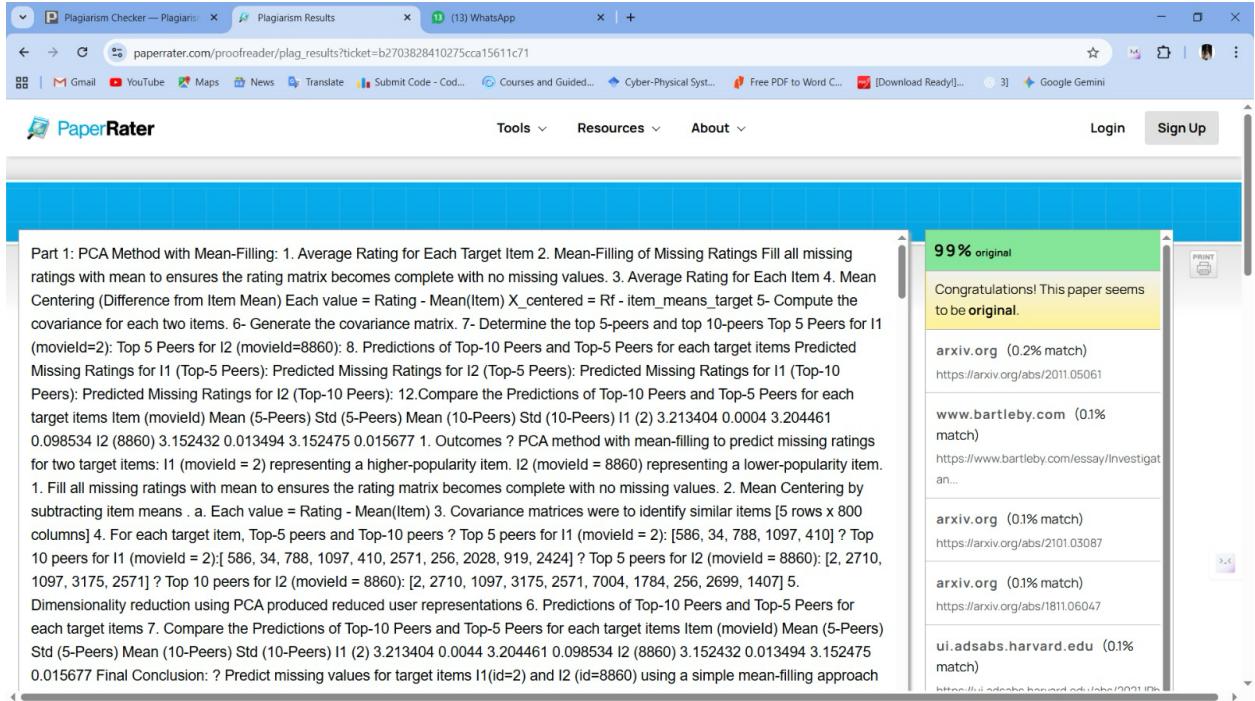
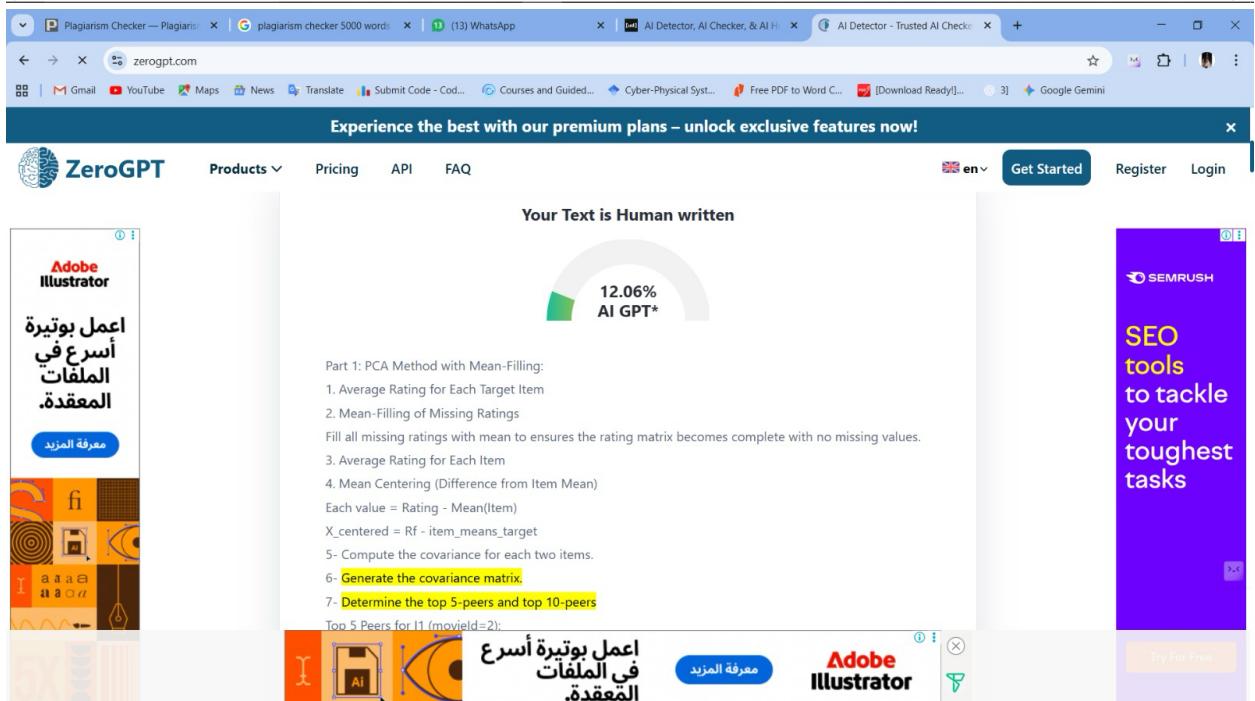


Plagiarism Report

1) Section 1:



Part 1: PCA Method with Mean-Filling: 1. Average Rating for Each Target Item 2. Mean-Filling of Missing Ratings Fill all missing ratings with mean to ensures the rating matrix becomes complete with no missing values. 3. Average Rating for Each Item 4. Mean Centering (Difference from Item Mean) Each value = Rating - Mean(item) $X_{centered} = Rf - item_means_target$ 5- Compute the covariance for each two items. 6- Generate the covariance matrix. 7- Determine the top 5-peers and top 10-peers Top 5 Peers for I1 (movield=2): Top 5 Peers for I2 (movield=8860): 8. Predictions of Top-10 Peers and Top-5 Peers for each target items Predicted Missing Ratings for I1 (Top-5 Peers): Predicted Missing Ratings for I2 (Top-5 Peers): Predicted Missing Ratings for I1 (Top-10 Peers): Predicted Missing Ratings for I2 (Top-10 Peers): 12. Compare the Predictions of Top-10 Peers and Top-5 Peers for each target items Item (movield) Mean (5-Peers) Std (5-Peers) Mean (10-Peers) Std (10-Peers) I1 (2) 3.213404 0.0004 3.204461 0.098534 I2 (8860) 3.152432 0.013494 3.152475 0.015677 1. Outcomes ? PCA method with mean-filling to predict missing ratings for two target items: I1 (movield = 2) representing a higher-popularity item. I2 (movield = 8860) representing a lower-popularity item. 1. Fill all missing ratings with mean to ensures the rating matrix becomes complete with no missing values. 2. Mean Centering by subtracting item means . a. Each value = Rating - Mean(item) 3. Covariance matrices were to identify similar items [5 rows x 800 columns] 4. For each target item, Top-5 peers and Top-10 peers ? Top 5 peers for I1 (movield = 2): [586, 34, 788, 1097, 410] ? Top 10 peers for I1 (movield = 2): [586, 34, 788, 1097, 410, 2571, 256, 2028, 919, 2424] ? Top 5 peers for I2 (movield = 8860): [2, 2710, 1097, 3175, 2571] ? Top 10 peers for I2 (movield = 8860): [2, 2710, 1097, 3175, 2571, 7004, 1784, 256, 2699, 1407] 5. Dimensionality reduction using PCA produced reduced user representations 6. Predictions of Top-10 Peers and Top-5 Peers for each target items 7. Compare the Predictions of Top-10 Peers and Top-5 Peers for each target items Item (movield) Mean (5-Peers) Std (5-Peers) Mean (10-Peers) Std (10-Peers) I1 (2) 3.213404 0.0044 3.204461 0.098534 I2 (8860) 3.152432 0.013494 3.152475 0.015677 Final Conclusion: ? Predict missing values for target items I1(id=2) and I2 (id=8860) using a simple mean-filling approach



Your Text is Human written

12.06% AI GPT*

Part 1: PCA Method with Mean-Filling:

1. Average Rating for Each Target Item
2. Mean-Filling of Missing Ratings

Fill all missing ratings with mean to ensures the rating matrix becomes complete with no missing values.

3. Average Rating for Each Item
4. Mean Centering (Difference from Item Mean)

Each value = Rating - Mean(item)

$$X_{centered} = Rf - item_means_target$$

- 5- Compute the covariance for each two items.
- 6- Generate the covariance matrix.
- 7- Determine the top 5-peers and top 10-peers

Top 5 Peers for I1 (movield=2):

Adobe Illustrator

اعمل بوتيرة أسرع في الملفات المعقدة.

معرفة المزيد

SEM RUSH

SEO tools to tackle your toughest tasks

Try For Free

2) Section 2:

 PaperRater

Tools ▾ Resources ▾ About ▾

Login Sign Up

99% original

Congratulations! This paper seems to be **original**.

PRINT

Section 2: Podcast Recommendation Engine with Transcript Analysis 1) Introduction This system implements a podcast recommendation system with transcript analysis where it aims to recommend podcasts to users according to different techniques that will recommend based on user behavior and podcast features. Our system does not depend on only one technique, instead it combines content based with collaborative filtering to reach a higher performance. Our system integrates the features such as podcast description, reviews, ratings, categories, popularity metrics, and others to enhance our recommendation system. We want to recommend podcasts to different types of users including new listeners and active listeners. Several challenges were faced in this domain since podcasts in general has a big sparsity problem where most users rarely rate the podcasts and even if they do they mostly gave a rating of 5. This also contributes to the cold start problem since some users have limited interactions with podcasts so we had to find a proper balance between choosing to recommend based on popularity or the user behavior and item features. 2) Data and Methodology 2.1) Dataset Description: The dataset utilized is a combination of two main datasets on kaggle. The two datasets are of the same author but they are different versions. Three main files were merged sql file for user reviews, sql file for podcast information, and json file for podcast descriptions. This was done to include the reviews of the users to the recommendation system. The dataset includes features such as podcast ID, itunes ID, title, rating, description, review, podcast category, and others. After merging according to the itunes ID, they had 1,475,285 unique users, 111,544 unique podcasts, and 4,553,715 total interactions. Due to computational resources, data was sampled to 25,000 users, 7206 podcasts, 70,886 interactions. The ratings are on a scale of 1 to 5 and the dataset has a huge sparsity problem reaching 99.9% of sparsity. 2.2) Dataset Processing & Statistics: Due to merging different versions of the datasets, several podcasts had missing description reaching 2,572,621 missing values. Additionally, other features had 24,232 missing values as shown below: We decided to solve this problem by dropping the missing values since the rows did not have critical features that were needed. Additionally, we only kept rows that include descriptions with at least 50 letters and are written in english to ensure only meaningful descriptions remain. Afterwards, we showed the data statistics

www.studymode.com (0.2% match)
https://www.studymode.com/essays/LeaderIn-...

www.termpaperwarehouse.com (0.2% match)
https://www.termpaperwarehouse.com/essays/The...

link.springer.com (0.2% match)
http://link.springer.com/article/10.1186/s125...

www.biomedcentral.com (0.2% match)
http://www.biomedcentral.com/1472-6963/7/2

NOTE: redundant sources may not be shown

 ZeroGPT

Products ▾ Pricing API FAQ

en ▾ Get Started Register Login

Your Text is Human written

3.08%
AI GPT*

Section 2: Podcast Recommendation Engine with Transcript Analysis

1) Introduction

This system implements a podcast recommendation system with transcript analysis where it aims to recommend podcasts to users according to different techniques that will recommend based on user behavior and podcast features. Our system does not depend on only one technique, instead it combines content based with collaborative filtering to reach a higher performance.

Our system integrates the features such as podcast description, reviews, ratings, categories, popularity metrics, and others to enhance our recommendation system. We want to recommend podcasts to different types of users including new listeners and active listeners.

Several challenges were faced in this domain since podcasts in general has a big sparsity problem where most users rarely rate the podcasts and even if they do they mostly gave a rating of 5. This also contributes to the cold start problem since some users have limited interactions with podcasts so we had to find a proper balance between choosing to recommend based on popularity or the user behavior and item features.

2) Data and Methodology

2.1) Dataset Description:

The dataset utilized is a combination of two main datasets on kaggle. The two datasets are of the same