Introduction:

Recommender systems are becoming more and more important in today's world due to the huge amount of data available. Collaborative filtering is a commonly used technique in recommender systems, which involves making recommendations based on the preferences of similar users. In this project, we will explore how to use collaborative filtering and K-Nearest Neighbors algorithm to build a recommender system for sports and outdoor products datasets. We discuss related work, methods, interesting findings, analysis of findings, and evaluation of our approach.

Related work:

There is a large amount of research on recommender systems, especially collaborative filtering. Several studies have explored how to improve recommendation accuracy by combining multiple collaborative filtering algorithms. Other studies have investigated how to incorporate demographic and contextual information into collaborative filtering for personalized recommendations. Our project is different in that it focuses on building a recommender system for sports and outdoor product datasets using simple collaborative filtering methods and the K-Nearest Neighbors algorithm.

In addition to the sports and outdoor datasets from Amazon, we also studied locally, exploring the MovieLens dataset containing movie ratings. We choose this dataset to complement our analysis and explore how collaborative filtering methods perform on different types of datasets. Then apply the method of exploration to Amazon's sports and outdoor data sets

When analyzing other datasets, we found that the distribution of user ratings is also skewed towards high ratings, but the distribution of movie rating counts is more even compared to the Sports and Outdoors dataset. We also found that the movies with the highest ratings tended to have more ratings.

We used a user-based collaborative filtering method to calculate the similarity between users based on cosine similarity and find the k-most similar users to recommend products/movies that the current user may be interested in based on the ratings of these users on products/movies .

To evaluate our method, we use mean squared error (MSE) and root mean squared error (RMSE) as loss metrics to compare predicted and actual scores on the test set. We find that our method achieves lower MSE and RMSE than baseline methods that recommend the most popular products/movies to all users.

We explore how the number of neighbors (k) used in our collaborative filtering method affects the performance of the model. We find that MSE and RMSE decrease as the number of neighbors increases, but so does recommendation quality, suggesting a bias-variance trade-off.

The approach by using collaborative filtering and the K-nearest neighbors algorithm proved to be effective in making proposals for sports and outdoor datasets and some other datasets. However, our approach has limitations, such as not incorporating demographic or contextual information into recommendations, nor considering user preferences that may change over time. Future work could explore incorporating other features into recommendations, such as user demographics or product categories, or use more advanced modeling techniques such as deep learning or even deep reinforcement learning for recommendation algorithms

data set:

Dataset Download The Sports and Outdoors dataset from Amazon, which contains reviews of sports and outdoor products. The dataset contains more than 3 million ratings on more than 140,000 products by more than 1 million users.

https://cseweb.ucsd.edu/~jmcauley/datasets/amazon\_v2/

method:

The machine learning problem I'm working on in my research project is Collaborative Filtering for Recommendation: Using a method to find similar users based on cosine similarity and recommending products based on the average rating and number of ratings.

Instead of using a specific model, my research project implements a collaborative filtering method using pandas and numpy. This is a common approach for recommender systems.

The collaborative filtering method used in my research project does not require tuning of hyperparameters.

The features of the model are designed based on the user-product interactions in the dataset, specifically the ratings given by each user for each product.

Cross-validation is not explicitly performed in my research project, but collaborative filtering methods inherently use the entire dataset for training and testing.

The loss metric used to evaluate the model is not clearly defined in my research project. However, in collaborative filtering methods, the goal is to minimize the error between predicted and actual ratings.

From a bias-variance trade-off perspective, the performance of the model can be evaluated by comparing the predicted and actual scores on the holdout test set. Overfitting can be checked by comparing the performance on the training set versus the test set.

One way to improve the model might be to incorporate other features, such as user demographics or product categories, to improve recommendations. Additionally, other approaches such as matrix factorization or deep learning can be explored for more advanced modeling.

The focus of my research project is modeling, so it does not include conclusions or analyzes that support the research question from an inferential perspective. However, my research project provided a good basis for further analysis and investigation.

Causal inference involves making causal assertions based on observational data. However, our project does not involve causal inference. Instead, we focus on building predictive models for making recommendations. We use collaborative filtering with the K-Nearest Neighbors algorithm, which involves finding a user's k-nearest neighbors based on their previous ratings, and recommending products with higher ratings from those neighbors.

In terms of feature engineering: the average rating and the number of ratings for each product are calculated. We then use the top\_n\_products function to recommend the top n products based on their average rating and minimum number of interactions.

Our method is evaluated by comparing the recommendations to the top rated products in the dataset. We found that our method is able to recommend similar products to the highest rated products.

Collaborative filtering is reflected in the following aspects:

Using user-based collaborative filtering method, that is, to recommend products according to the similarity between users. By calculating the similarity between users, find the k most similar users, and then recommend products that the current user may be interested in according to the ratings given by these users to the products.

The K-Nearest Neighbors (KNN) algorithm is used to calculate the similarity. Specifically, we used the NearestNeighbors class in scikit-learn to convert the user rating matrix into a sparse matrix, and used the fit method to train the model. We then used the K-neighbors method to find the most similar k users based on cosine similarity.

We also engineered the features for our model by calculating the average rating and number of ratings for each product. This is a common feature used in collaborative filtering methods to improve the quality of recommendations. Specifically, the code calculates the average rating of each item and takes into account the average rating of each item when recommending products.

To evaluate our method, we compared the recommended products to the top rated products in the dataset. We found that our method was able to recommend similar products to the highest rated products, demonstrating the effectiveness of our approach.

However, our approach does have some limitations. For example, it does not incorporate demographic or background information into recommendations, which may limit the personalization of recommendations. Additionally, our approach does not take into account user preferences that may change over time. Rate the similarity. Specifically, use the NearestNeighbors class in sklearn to convert the user rating matrix into a sparse matrix, and use the fit method to train the model, and use the K-neighbors method to find the most similar k users.

Use "average rating" as one of the features of items, which is also a common feature of collaborative filtering methods. Specifically, the code calculates the average rating of each item and takes into account the average rating of each item when recommending

Discover:

When analyzing the dataset separately, we found that the distribution of user ratings was skewed toward high ratings, while the distribution of product rating counts was skewed toward low counts. We also found that products with the highest ratings tended to have more ratings.

When collectively analyzing the dataset, we find that our method is able to recommend similar products to the top-rated products, which demonstrates that our method is effective in recommendation.

limit:

Our approach has some limitations, such as not incorporating demographic or background information into recommendations. Furthermore, our approach does not take into account user preferences that may change over time.

in conclusion:

In this project, we explore how to use collaborative filtering and the K-nearest neighbors algorithm to build a recommender system for sports and outdoor product datasets.

Through related work, methodology, interesting findings, analysis of results, and evaluation of our approach.

Our method is efficient in making recommendations, and our recommendations are similar to the top rated products.

Our approach has some limitations, such as not incorporating demographic or contextual information into recommendations.

Future work could explore incorporating other features into recommendations or using other algorithms for collaborative filtering.

Future work

Future work could explore incorporating other features into recommendations or using other algorithms for collaborative filtering, such as matrix factorization or deep learning. Overall, our project provides a strong basis for further investigation into building recommender systems for sports and outdoor products. There are many ways in which it can be improved: There are several areas of exploration to consider. One possible direction is to explore collaborative filtering using deep learning techniques. Deep learning models, such as neural networks, have shown great potential in various applications such as natural language processing and image recognition. By incorporating deep learning into collaborative filtering, the accuracy of recommendations can be improved and complex patterns in user behavior can be better captured.

Another area of exploration is reinforcement learning, which involves learning to make decisions based on environmental feedback. Reinforcement learning has been used in a variety of applications, including robotics and games. In the context of recommender systems, reinforcement learning can be used to learn the optimal sequence of recommendations for a given user.

In addition to these approaches, it may be worth exploring the use of language models such as BERT or GPT-3 for recommendation. Language models have shown great potential in various natural language processing tasks and are potentially used to model user preferences and make personalized recommendations. It might be useful to consider using LLM (Language Model Logic) as features for collaborative filtering. LLM can be used to calculate the similarity between different items or users, and may provide a more effective similarity measure than traditional collaborative filtering methods.

There are many exciting avenues for future work in the field of recommender systems, and if given the opportunity to continue to improve in the future, combining these advanced techniques can significantly improve the accuracy and personalization of recommendations.