# Recommendation Systems

Aniket K Singh

April 22, 2022

## Contents

# Abstract

Recommendation Systems have become a part of our life, we see them everywhere without realizing that a recommendation algorithm is being used behind them. We get recommendations while watching videos, buying any product, reading news on websites, and booking flights/hotels. All of these are possible with Recommendation Systems. We are using the Matrix Factorization method to implement our recommendations system. We used both user and item-based recommendation systems.

# Introduction

Recommendation Systems are becoming more and more popular. Using a good recommendation system has become an essential part of most businesses. More and more companies are using their own version of recommendation systems. Big companies like Netflix, Youtube, Amazon, Airbnb, Facebook all have their recommendation systems. People are becoming more platform dependent, we want the application to tell us what we want. A Netflix user is satisfied because the recommendation they get for movies and shows are more personalized for their need. Amazon sells more product when they show products that a person would like to buy. Ads are more likely to generate revenue when the ads are relevant. [1]
In this study, we would learn different recommendation systems and how they function. We aim to develop our own recommendation systems using Deep Learning and Machine Learning methods.

# Types of Recommendation Systems

## Popularity Based Recommendation Systems

This recommendation system works on the basis of trends and popularity. The goal of this recommendation system is to recommend what is trending among the targetted group of users. This recommendation system does not need the historical data from the users. This recommendation system has its own advantages because there's no need for the historical dataset. The recommendation system doesn't suffer from a well known problem of most recommendation systems, known as "Cold Start Problem."
We can use this recommendation system when we do not have any data from the user, and also have this as a "Popularity Section" for every user so they can easily find what's popular in their preferred genre.[2]

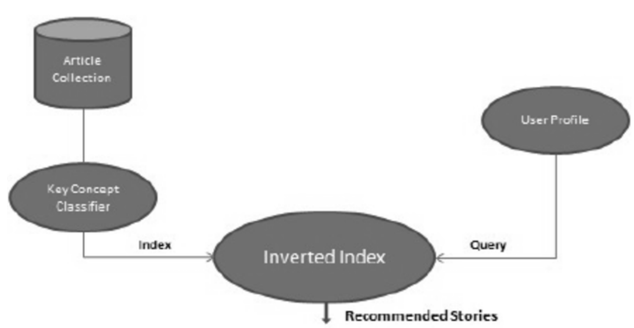



Figure 1: Architecture of the Popularity-based Recommender System [3]

## Classification Model

This recommendation system uses features of both the product and the user to predict whether the recommendation will be liked/disliked by the given user. This model employees a classifer based on certain rules to perform the prediction. [2]

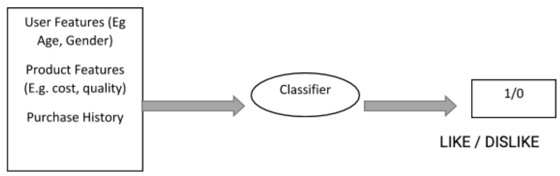



Figure 2: Classification Model Recommender System [2]

## Conent-Based Recommendation Systems

This recommendation system also uses content/product features to determine the recommendations. This recommendation system does not employ the historical data of the user. Multiple methods are used to determine the recommendations. We usually use Euclidean distance between the contents to find the nearest recommendations that a user may like. Other methods can be used such as using a Pearson correlation coefficient, and recommending the content/product if they're closely correlated. We would get a more in depth overview of these recommendation systems in this study. [2]

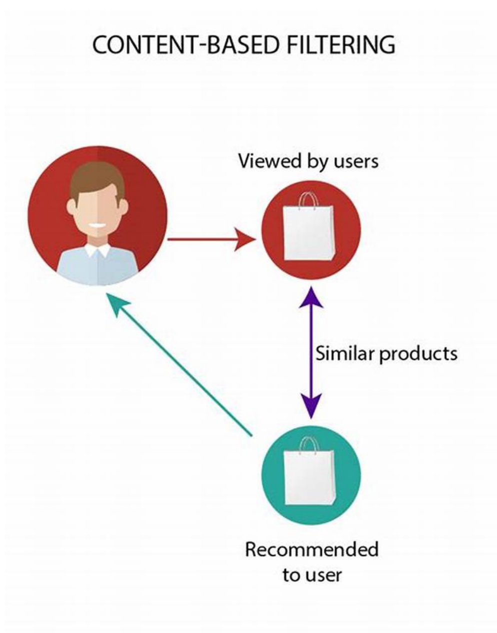



Figure 3: Content-Based Recommendations System [4]

## Collaborative Filtering Recommendation Systems

This type of recommendations systems are widely used and considered to be the intelligent recommendation system. Collaborative filtering uses historical data of the users. Collaborative filtering finds similarties between users and makes prediction for better recommendations. Collaborative filtering requires data, and having more data makes the prediction better, which results in better recommendations. We have many methods for collaborative filtering that we will be doing futher study in this paper. [2]

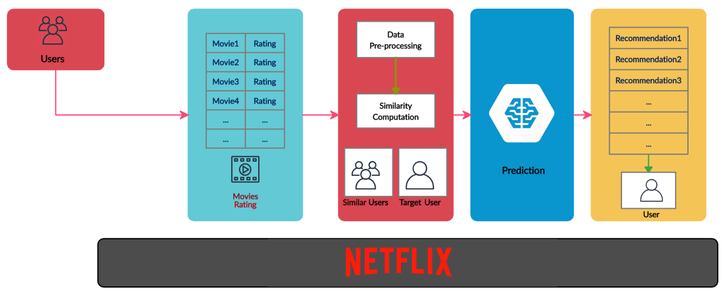



Figure 4: Collaborative Filtering Recommendation System [5]

## Hybrid Recommendation Systems

This type of recommendations systems uses both Content-based filtering and Collaborative filtering algorithms. These recommendation systems can produce better recommendations as they have both historical data and also using the contents to provide users with the recommendations. We can expect more accurate recommendations even with less data and we may be able to overcome cold start problems. A hybrid recommendation systems may also use other recommendation systems algorithms to provide us with more accurate results. We will be doing more in-depth study of hybrid recommendations system in this paper. We will be also comparing performance of a conventional recommendations system with some hybrid recommendations systems. [6]

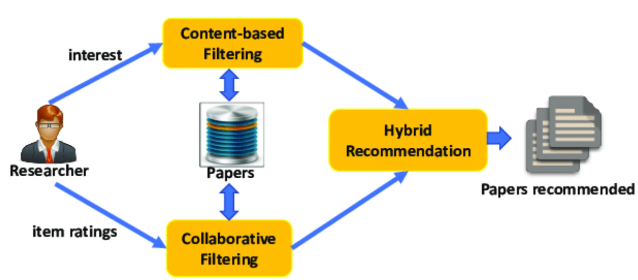



Figure 5: Hybrid Recommendation System [6]

# Problems with Recommendation Systems

## Cold Start Problem

One of the major problem that we encounter while working with recommendation systems are the Cold Start Problem. In this study, we have mentioned "Cold Start Problem" several times, but what is a cold start problem? Let's understand why it is a problem and how we resolve these issues.

Cold Start Problems occur when there is no/less data available realted to user or the product. As we know most recommendation systems are dependent on hisotrical data about the user or product and sometimes both. So not having data/information available, would be an issue and we can't make expect our recommendation system to produce meaningful recommendations. There are multiple mitigation techniques that can be used to resolve this issue and expect better recommendations in such cases. We will be using those techniques to resolve any cold start problems. To learn more about Cold Star Problem and it's mitigation techniques, see [7]

# Models for Recommendation Systems

## Matrix Factorization Model

The matrix factorization model factorizes the rating matrix into the product of two lower-rank matrix. Matrix facotization is one of the most reiable model to build a recommendation system.

Let $\mathbb{R} \in \mathbb{R}^{p \times q}$ denote the interaction matrix with 'p' number of users and 'q' number of products. Then the user-item matrix will be factorized into a user latent matrix $\mathbf{A} \in \mathbb{R}^{p \times k}$ and a product latent matrix $\mathbf{B} \in \mathbb{R}^{q \times k}$ where k < p, k < q.

Here, **k** denotes the latent factor size. Let $a_u$ denote the u$^{\text{th}}$ row of $\mathbf{A}$ and $b_i$ denote the i$^{\text{th}}$ row of $\mathbf{B}$.

Now, we estimate the predicted ratings by

$$\hat{\mathbb{R}} = AB^T$$

Where $\hat{\mathbb{R}} \in \mathbb{R}^{p \times q}$ is the matrix that represents predicted ratings. We can calculate the predicted ratings of a product given by user u by a product.

$$\hat{\mathbb{R}} = a_u {b_i}^T + c_u + c_i$$

Now, we train our model by minimizing the objective fucntion given below.

$$\underset{A,B,C}{\arg\max} \sum_{(u,i)\in k} \|\mathbb{R}_{ui} - \hat{\mathbb{R}}_{ui}\| + \beta({\|A\|_F}^2 + {\|B\|_F}^2 + {C_u}^2 + {C_i}^2)$$

Where $\beta$ is the rate of regularization. $\beta(\|A\|_F{}^2 + \|B\|_F{}^2 + C_u{}^2 + C_i{}^2)$ is term that avoids overfitting of data.
We use Root Mean Square Error (RMSE) to evaluate the matrix factorization model.
RMSE is calculated by

$$RMSE = \sqrt{\frac{1}{|T|}\sum_{ui \in T}(\mathbb{R}_{ui} - \hat{\mathbb{R}}_{ui})^2}$$

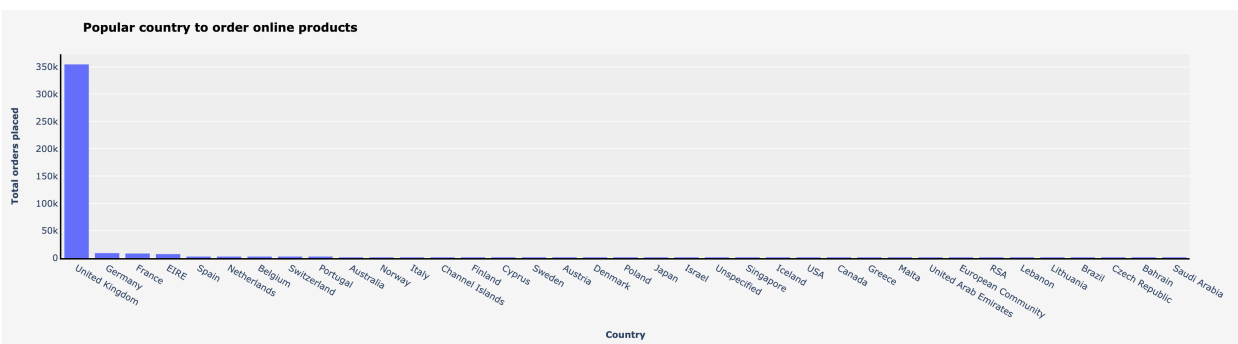
Here, T is the set of pairs of users and products evaluated. [8]

# Product Recommendation System

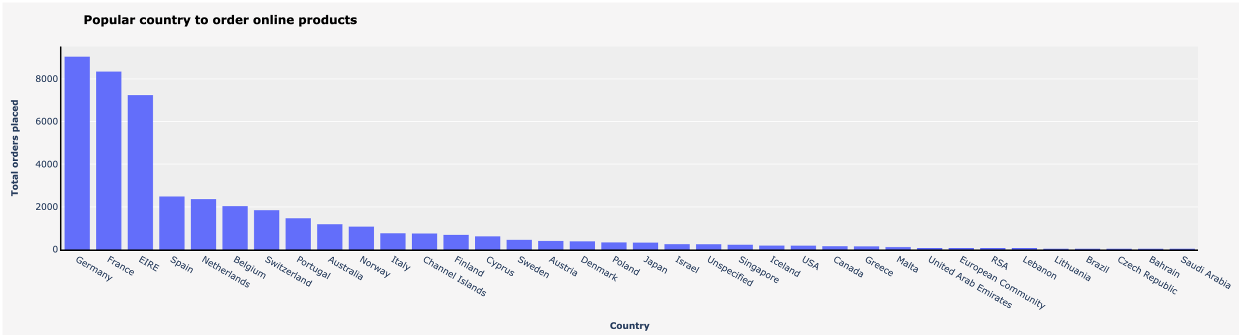
For this study, we implemented a product recommendation system using Collaborative Filtering, Matrix Factorization. We have a dataset of Online Retail with 531285 rows and 8 columns. Our features include Invoice number, Stock Code, Description, Quanity, Date, Unit Price, CustomerID, and Country.

## Data Preparation

In our dataset, we first converted Date to time series datatime format. After conversion, we created new variable of datetime into Day, Month, and Year to perform better analysis. As our dataset had most sales from UK.




We then created two seperate datasets with UK-only and Non-Uk sales. We can see that now we have a better insight of sales from other countries in the plot below.




As the dataset didn't have any missing values, we didn't need to do anything about them.

We then created matrix for dataset to use Matrix Facotrization. As we used Python for this implementation, python allows us to create matrix using pivot table function. We first created a customer item matrix. Also, we created user-to-user similarity matrix using Python's cosine similarity function and pandas.

## Model Results

We first used item based Collaborative Filtering using Sklearn. We used items bought by user A and items bought by user B, to find out what new items should be recommended to user B. Using Matrix Factorization, we succesfully created new list of items that would be recommended to user B based on items bought by user A and B. We also used User based Collaborative Filtering using Sklearn. We used similar approach but with users A and user B. In this case we used user's information to create new recommendations for different user. This also used a similar approach to produce new recommendations. Here are new recommendations we got from our result.

| | Description |
|---|---|
| StockCode | |
| 21832 | CHOCOLATE CALCULATOR |
| 21915 | RED HARMONICA IN BOX |
| 22620 | 4 TRADITIONAL SPINNING TOPS |
| 79066K | RETRO MOD TRAY |
| 21864 | UNION JACK FLAG PASSPORT COVER |
| 79191C | RETRO PLASTIC ELEPHANT TRAY |
| 21908 | CHOCOLATE THIS WAY METAL SIGN |
| 20615 | BLUE POLKADOT PASSPORT COVER |
| 20652 | BLUE POLKADOT LUGGAGE TAG |
| 22348 | TEA BAG PLATE RED RETROSPOT |
| 22412 | METAL SIGN NEIGHBOURHOOD WITCH |
| 21171 | BATHROOM METAL SIGN |
| 84086C | PINK/PURPLE RETRO RADIO |

## Model Strength

Our model uses both items and users both to create new recommendations for a new user. The model uses collaborative filtering method. Our model succesfully predicts recommendations using only user and items.

## Model Weakness

Model doesn't include ratings because of lack of ratings in the dataset.

## Future Work

Using a better dataset which includes user ratings would perform better with Matrix Factorization. Using newer deep learning algorithms such as AutoRec, Personalized Ranking Recommendation Systems, Neural Collaborative Fiktering Recommendation systems.

# Bibliography


[1] Rocca, B. *Introduction to recommender systems*, 2021.

[2] *What Are Recommendation Systems in Machine Learning?*

[3] Nirmal Jonnalagedda and Susan Gauch. Personalized news recommendation using twitter. pages 21–25, 11 2013.

[4] Arif Zainurrohman. Content-based recommender system using nlp. [Online; accessed January 20, 2022].

[5] Saniya Parveez, Roberto Iriondo. Recommendation system tutorial with python using collaborative filtering. [Online; accessed January 21, 2022].

[6] Yugesh Verma. A guide to building hybrid recommendation systems for beginners. [Online; accessed January 21, 2022].

[7] VIJAYSINH LENDAVE. Cold-start problem in recommender systems and its mitigation techniques. [Online; accessed January 23, 2022].

[8] Aston Zhang, Zachary C. Lipton, Mu Li, and Alexander J. Smola. Dive into deep learning. [Online; accessed April 21, 2022].