**CCT College Dublin**

|  |  |
| --- | --- |
| **Module Title:** | Problem Solving for Industry |
| **Assessment Title:** | RECOMENDATION SYSTEM, Using Machine Learning |
| **Lecturers Name:** | Ken Healy MSc. HDip. Ed.  Dr. Muhammad Iqbal |
| **Students Full Name:** | Carlos Santana Bobadilla  Nisha Raj  Ricardo Alves de Souza  Rosemol Joy |
| **Student Numbers:** | 2017227  2018493  2018499  2018492 |
| **Assessment Due Date:** | 16/05/2022 |
| **Date of Submission:** | 16/05/2022 |

**Declaration**

|  |
| --- |
| By submitting this assessment, I confirm that I have read the CCT policy on Academic Misconduct and understand the implications of submitting work that is not my own or does not appropriately reference material taken from a third party or other source. I declare it to be my own work and that all material from third parties has been appropriately referenced. I further confirm that this work has not previously been submitted for assessment by myself or someone else in CCT College Dublin or any other higher education institution. |

# Abstract

**Recommender System using Machine Learning.**

Understand what a recommender system is and use it to generate recommendations in a dataset within the paradigms of CRISP-DM (CRoss Industry Standard Process for Data Mining).

Recommender systems are massively used by the current industry in various sectors such as products, books, movies, among others. This has generated in recent times a growing volume of research and development of techniques to facilitate and extract better results.

In this case study, we used MovieLens movie ratings as a data source for mining to generate a recommendation system for users.

We implement a variation of the k-NN algorithm, which stands for k-nearest neighbors, used for classification and regression that does not need to have a trained model. This being a famous algorithm, created in 1951 by Evelyn Fix and Joseph Hodges.

Table of Contents

[Abstract 3](#_Toc103610019)

[Roles and Responsibilities 6](#_Toc103610020)

[Chapter 1 - Development Methodology and Project Plan 7](#_Toc103610021)

[( Crisp-DM framework ) 7](#_Toc103610022)

[1.1 Business Understanding 7](#_Toc103610023)

[**1.1.1** **Some benefits of recommendation system:** 7](#_Toc103610024)

[**1.1.2** **Some famous Recommendation Systems** 9](#_Toc103610025)

[**1.1.2.1** **Google Recommendation System** 9](#_Toc103610026)

[**1.1.2.2** **Netflix Recommendation System** 9](#_Toc103610027)

[**1.1.3** **How to Boost the sales?** 10](#_Toc103610028)

[**1.1.4** **How it works** 10](#_Toc103610029)

[1.2 Data Understanding 12](#_Toc103610030)

[**1.2.1** **Errors which can reduce the performance of the model:** 12](#_Toc103610031)

[1.3 Data Preparation 13](#_Toc103610032)

[**1.3.1** **Movie Recommendation System:** 14](#_Toc103610033)

[**1.3.2** **Exploratory data Analysis on Movies dataset:** 14](#_Toc103610034)

[**1.3.3** **Nearest Neighbors:** 15](#_Toc103610035)

[**1.3.4 Product Recommendation System:** 15](#_Toc103610036)

[**1.3.6** **Results of Product Recommendation Engine:** 16](#_Toc103610037)

[1.4 Modelling 17](#_Toc103610038)

[**1.4.1 Types of Recommendation System** 18](#_Toc103610039)

[**1.4.2** **K-Nearest Neighbors** 20](#_Toc103610040)

[1.5 Evaluation 22](#_Toc103610041)

[**1.5.1** **Evaluation Metrics for Recommendation Systems** 22](#_Toc103610042)

[**1.5.2** **Classification Accuracy Metrics** 23](#_Toc103610043)

[**1.5.3** **Rank Accuracy Metrics:** 24](#_Toc103610044)

[1.6 Deployment 25](#_Toc103610045)

[**1.6.1** **Best Approaches to Deploy a Recommendation System:** 25](#_Toc103610046)

[**1.6.2** **Platforms To Deploy Recommendation Systems** 25](#_Toc103610047)

[**1.6.3** **Why To Choose Heroku for Testing Recommendation Systems** 25](#_Toc103610048)

[**1.6.4** **Heroku using Flask** 26](#_Toc103610049)

[**1.6.5** **Steps for Recommendation System deployment on Heroku using Flask:** 26](#_Toc103610050)

[Conclusion 30](#_Toc103610051)

[References List 31](#_Toc103610052)

# Roles and Responsibilities

Today in the Modern world many companies are using recommendation engines to recommend movies, music, television programs, books, documents, websites, conferences, tourism scenic spots, and learning materials, and involve the areas of e-commerce, e-learning, e-library, e-government, and e-business services.

Despite its great progress so far, artificial intelligence (AI) is facing a serious challenge in the availability of high-quality Big Data. In many practical applications, data are in the form of isolated islands. Efforts to integrate the data are increasingly difficult partly due to serious concerns over user privacy and data security.

The problem is exacerbated by strict government regulations such as Europe's General Data Privacy Regulations (GDPR). In this talk, I will review these challenges and describe efforts to address them in recommendation systems area. In particular, I will give an overview of recent advances in federated learning and then focus on developments of “federated recommendation systems”, which aims to build high-performance recommendation systems by bridging data repositories without compromising data security and privacy. ( 2019, Y. Qiang )

Y. Qiang, "Federated Recommendation Systems," *2019 IEEE International Conference on Big Data (Big Data)*, 2019, pp. 1-1, doi: 10.1109/BigData47090.2019.9005952.

# Chapter 1 - Development Methodology and Project Plan

# ( Crisp-DM framework )

## Business Understanding

A recommendation system is used to predict whether a customer will buy any specific product or not based on previous shopping history. We can use recommendations to bring any kind of business to another level. On top of strengthening relationships with your customers, the recommendation engines can provide higher returns to your business as they can help boost engagement opportunities with your products and offer a greater influx of cross-selling opportunities.

Netflix, YouTube, Tinder, and Amazon are all examples of recommender systems in use. The systems entice users with relevant suggestions based on the choices they make. Recommender systems can also enhance experiences for News Websites.

### **Some benefits of recommendation system:**

#### **Engage Shoppers**

Shoppers become more engaged when personalized product recommendations are made to them across the customer journey. Through individualized product recs, customers can delve more deeply into your product line without having to dive into (and very likely get lost in) an e-commerce rabbit hole. Kibo Research shows that 52% of retailers are leveraging AI-driven personalization to deliver personalized product recommendations to their customers.

#### **Drive Traffic**

Through personalized email messages and targeted blasts, a recommendation engine can encourage elevated amounts of traffic to your site, thus increasing the opportunity to scoop up more data to further enrich a customer profile.

#### **Increase Average Order Value**

Average order values typically go up when an engine is leveraged to display personalized options as shoppers are more willing to spend generously on items they thoroughly covet.

#### **Increase the Number of Items per Order**

In addition to the average order value rising, the number of items per order also typically rises when an engine is employed. When the customer is shown options that meet their interest, they are far more likely to add items to their active purchase cart.

#### **Convert Shoppers to Customers**

Converting shoppers into customers takes a special touch. Personalized interactions from a recommendation engine show your customer that he or she is valued as an individual, in turn, engendering long-term loyalty.

#### **A Recommendation Engine Provides Reports**

Detailed reports are an integral part of a personalization system. Accurate and up-to-the-minute reporting will allow you to make informed decisions about the direction of a campaign or the structure of a product page.

#### **Offer Advice and Direction**

An experienced recommendation provider like Kibo can offer advice on how to use the data collected from your recommendation engine. Acting as a partner and a consultant, the provider should have the industry know-how needed to help guide you and your eCommerce site to a prosperous future.

### **Some famous Recommendation Systems**

### **Google Recommendation System**

When you use Google Shopping, you're browsing products from advertisers and sellers who have chosen to feature their products on Google Shopping. Unless otherwise indicated, offers on Google Shopping are ranked based on relevance, including your search terms and other Google activity.

### **Netflix Recommendation System**

Netflix uses machine learning, a subset of artificial intelligence, to help their algorithms “learn” without human assistance. Machine learning gives the platform the ability to automate millions of decisions based on user activities. Netflix takes feedback from every visit to the website service and continually re-trains our algorithms with those signals to improve the accuracy of their prediction of what we're most likely to watch.

### **How to Boost the sales?**

One powerful way to boost your sales, customer satisfaction, and customer loyalty is to use a recommender system. Such a system enables you to draw up user profiles based on their behavior and to predict which products are most relevant to such profiles. A user profile specifies the properties of how much interest a customer has in a particular product. The Product recommendation system is used to recommend popular products to potential customers.

Recommender systems are not new. In 2006, Netflix started a one million dollar competition to improve its recommendation algorithms. In 2019, 75% of their viewed content is attributed to personalized recommendations.

### **How it works**

The basic settings for recommendation services are similar to those for search services. Similar to search services, the primary purpose of a recommender system is to provide a customer with a ranked list of recommended items.

The vast majority of recommendation methods assume that customer ratings are available for catalog items. The ratings can be explicitly provided by customers or derived from behavioral data, such as purchases and online browsing histories.

Certain recommendation methods rely on content and catalog data to calculate similarities between the items based on their attributes.

Some recommender systems can take advantage of additional user data, such as online order histories or store transactions.

Both recommendation requests and customer ratings can be complemented with contextual information, such as time, location, or marketing channel. A recommender system can use contextual data to improve the relevance of recommendations.

Many businesses take up artificial intelligence (AI) technology to try to reduce operational costs, increase efficiency, grow revenue, and improve customer experience. For the greatest benefits, businesses should look at putting the full range of smart technologies - including machine learning, natural language processing, and more - into their processes and products. However, even businesses that are new to AI can reap major

![Machine%20learning.png](data:image/png;base64;base64,)

## Data Understanding

The advent of the e-commerce markets facilitates the process of shopping without the need for physical interactions with products. However, an appealing aspect of physical retail stores is that customers who are undecided on the products they desire to purchase have the ability to browse and receive recommendations from shelf displays and salespeople. The e-commerce industry utilizes recommendation models to satisfy this objective.

A personalized recommendation model aims to identify products that are of most relevance to a customer based on his or her past interactions. This enhances a user’s intention to browse more products and makes them more likely to buy these products, effectively increasing e-commerce revenue . Thus, the evaluation of recommendation algorithms for a range of properties is essential in order to select the best algorithm from a set of candidates]. Performance is either measured by offline evaluations, by conducting user studies, or by using online evaluations when a recommendation model is live on an online platform. Online evaluations are unique in that they allow direct measurement of overall system goals, such as long-term profit or user retention.

### **Errors which can reduce the performance of the model:**

* **Data entry errors:**

Data entries recorded by humans from speech or written context, where most often there are fields of data stored that might not always have a value present; hence an improper default value may be assigned to this field without much consideration to whether or not the default value is a possible outcome of real data.

* **Distillation errors:**

Data that should be pre processed before storing them onto a database to reduce complexity and noise of raw data, which if not considered, may impede final analysis. For example, incremental revenue or the conversion metrics calculated may be invalid if the prices utilized for each product varies in the currency they are stored against. It can then be beneficial to assert this when recording data for e-commerce businesses who have a global dominance, to standardize the currency used against one system (such as the USD) for all customer interactions. This notion serves as the background to Section Uniform Data Management.

* **Data Integration errors:**

When the data collected stems from various servers or sources where the merging of records into a single database may cause inconsistencies. For example, it is crucial that the timestamps of each user interaction on the database are stored against a standard time zone, such as using Coordinated Universal Time (UTC), and not against the time zone of the IP registering the interaction.

## Data Preparation

Let’s build two recommendations with help of Nearest Neighbour. I will use the collaborative filtering technique. I have two datasets one is about movies rating and the other one is about amazon product reviews.

**Movie’s rating dataset features:**

1. UserId
2. MovieId
3. Rating (Target column)
4. Title

### **Movie Recommendation System:**

I have a dataset of movies reviews. This dataset contains features. The dataset is pretty so I don't have to do much work for cleaning.

![recommendation-system-project.png](data:image/png;base64;base64,)

### **Exploratory data Analysis on Movies dataset:**

According to the dataset these are the best movies and top rated movies.

![Top%2010%20movies.png](data:image/png;base64;base64,)

### **Nearest Neighbors:**

k-NN is a machine learning algorithm to find clusters of similar users based on common book ratings, and make predictions using the average rating of top-k nearest neighbors.

![Cosine%20Similarity.png](data:image/png;base64;base64,)

We are going to use the Nearest neighbors to do collaborative filtering. I will use cosine similarity to calculate the distance between the points. Cosine similarity is a metric used to measure how similar two items are. Mathematically, it measures the cosine of the angle between two vectors projected in a multi-dimensional space. The output value ranges from 0–to 1. 0 means no similarity, whereas 1 means that both the items are 100% similar.

### **1.3.4 Product Recommendation System:**

For product Recommendation I am going to use the amazon reviews dataset. the dataset contains 4 features but it does not contain the product title feature. we only have got product id. so our recommendation system will only return product id instead of title.

**1.3.5 Products rating dataset features:**

1) UserId  
2) ProductId  
3) Rating (Target column)  
4) Timestamp

![Product%20dataset.png](data:image/png;base64;base64,)

### **Results of Product Recommendation Engine:**

![Top%2010%20products.png](data:image/png;base64;base64,)

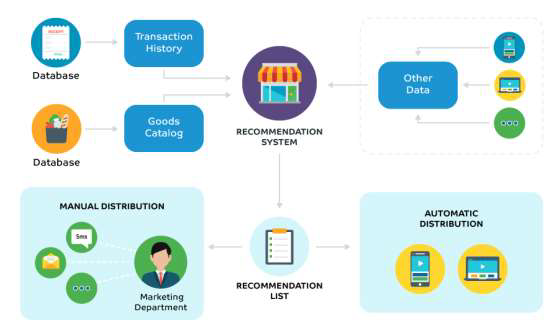
**Product:**

* Eyeliner Pen.

**Suggestions:**

* Kms California Hair Conditioner.
* Anti Aging face cream.
* Women’s Perfume.
* Peanuts: A Charlie Brown Christmas.

## Modelling



Recommender systems are the systems that are designed to recommend things to the user based on many different factors. These systems predict the most likely product that the users are most likely to purchase and are of interest to. Companies like Netflix, Amazon, etc. use recommender systems to help their users to identify the correct product or movies for them.

The recommender system deals with a large volume of information present by filtering the most important information based on the data provided by a user and other factors that take care of the user’s preference and interest. It finds out the match between user and item and imputes the similarities between users and items for recommendation. Both the users and the services provided have benefited from these kinds of systems.

The quality and decision-making process has also improved through these kinds of systems.

**What can be Recommended?**

There are many different things that can be recommended by the system like movies, books, news, articles, jobs, advertisements, etc.

Netflix uses a recommender system to recommend movies and web-series to its users. Similarly, YouTube recommends different videos.

There are many examples of recommender systems that are widely used today.

### **1.4.1 Types of Recommendation System**

1. **Popularity based recommendation System:**

It is a type of recommendation system which works on the principle of popularity and or anything which is in trend. These systems check about the product or movie which are in trend or are most popular among the users and directly recommend those. For example, if a product is often purchased by most people then the system will get to know that that product is most popular so for every new user who just signed it, the system will recommend that product to that user also and chances becomes high that the new user will also purchase that.

1. **Content-Based Recommendation System**

It is another type of recommendation system which works on the principle of similar content. If a user is watching a movie, then the system will check about other movies of similar content or the same genre of the movie the user is watching. There are various fundamentals attributes that are used to compute the similarity while checking about similar content.

1. **Collaborative Filtering**

It is considered to be one of the very smart recommender systems that work on the similarity between different users and also items that are widely used as an e-commerce website and also online movie websites. It checks about the taste of similar users and does recommendations. The similarity is not restricted to the taste of the user moreover there can be consideration of similarity between different items also. The system will give more efficient recommendations if we have a large volume of information about users and items.

1. **Hybrid Model**

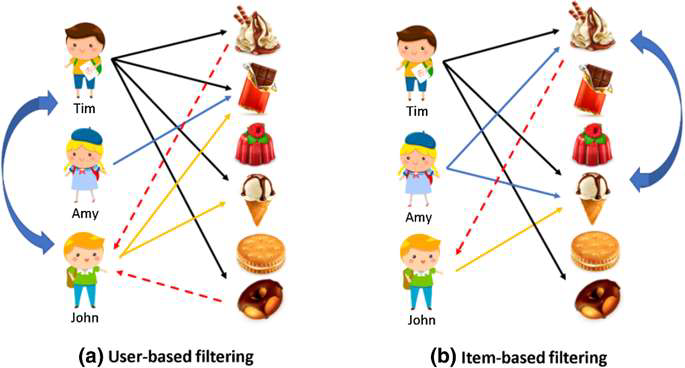
Hybrid filters combine several passive or active filters, their structure may be of series or parallel topology or a combination of the two. this type of filtering is used for advanced level recommendation systems Advantages of Collaborative filtering:

• We don't need domain knowledge because the embeddings are automatically learned.

• The model can help users discover new interests. In isolation, the ML system may not know the user is interested in a given item, but the model might still recommend it because similar users are interested in that item.

• To some extent, the system needs only the feedback matrix to train a matrix factorization model. In particular, the system doesn't need contextual features. In practice, this can be used as one of multiple candidate generators.

### **K-Nearest Neighbors**



We will also predict the rating of the given movie based on its neighbors and compare it with the actual rating. Item-based nearest-neighbour collaborative filtering:

Figure b shows user X, Y, and Z respectively.

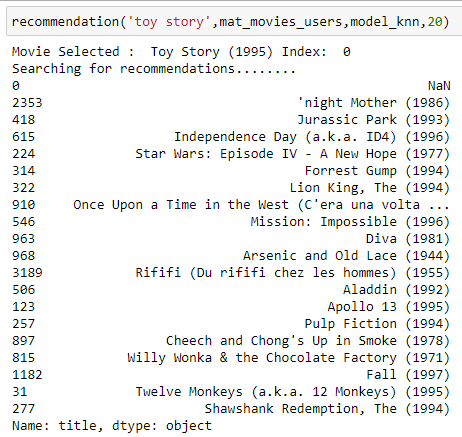
The system checks the items that are similar to the items the user bought. The similarity between different items is computed based on the items and not the users for the prediction. Users X and Y both purchased items A and B so they are found to have similar tastes.

Limitations

• Enough users required to find a match. To overcome such cold start problems, often hybrid approaches are made use of between CF and Content-based matching.

• Even if there are many users and many items that are to be recommended often, problems can arise of user and rating matrix to be sparse and will become challenging to find out about the users who have rated the same item.

• The problem in recommending items to the user due to sparsity problems. Results of Movie recommendation system: I have created a function that will take 5 parameters title of the movie, model, number of recommendations, and dataset. I have chosen the title of the toy story. Now you can see the recommendation engine is giving 20 suggestions which closer to the toy story.



## Evaluation

### **Evaluation Metrics for Recommendation Systems**

Evaluation Metrics Helps to find the Accuracy or Success of Recommender Systems. Predictive accuracy metrics, classification accuracy metrics, rank accuracy metrics, and non-accuracy measurements are the four major types of evaluation metrics for recommender systems.

**Predictive Accuracy Metrics**

![20190714113817886.png](data:image/png;base64;base64,)

Predictive accuracy or rating prediction measures address the subject of how near a recommender’s estimated ratings are to genuine user ratings.

This sort of measure is widely used for evaluating non-binary ratings. It is best suited for usage scenarios in which accurate prediction of ratings for all products is critical.

* Mean Absolute Error (MAE)
* Mean Squared Error (MSE)
* Root Mean Squared Error (RMSE)
* Normalized Mean Absolute Error (NMAE)

Are the most important measures for this purpose. In comparison to the MAE metric, MSE and RMSE employ squared deviations and consequently emphasize bigger errors.

The error is described by MAE and RMSE in the same units as the obtained data, whereas MSE produces squared units.

To make results comparable among recommenders with different rating scales, NMAE normalizes the MAE measure to the range of the appropriate rating scale.

In the Netflix competition, the RMSE measure was utilized to determine the improvement in comparison to the Cinematch algorithm, as well as the prize winner.

### **Classification Accuracy Metrics**

![1_NhPwqJdAyHWllpeHAqrL_g.png](data:image/png;base64;base64,)

Classification accuracy measures attempt to evaluate a recommendation algorithm’s successful decision-making capacity (SDMC). They are useful for user tasks such as identifying nice products since they assess the number of right and wrong classifications as relevant or irrelevant things generated by the recommender system.

The exact rating or ranking of objects is ignored by SDMC measures, which simply quantify correct or erroneous classification. This type of measure is particularly well suited to e-commerce systems that attempt to persuade users to take certain actions, such as purchasing products or services.

### **Rank Accuracy Metrics:**

![1_3vI82IYrTiN7fX0ht6mscw.png](data:image/png;base64;base64,)

In statistics, a rank accuracy or ranking prediction metric assesses a recommender’s ability to estimate the correct order of items based on the user’s preferences, which is known as rank correlation measurement. As a result, if the user is given a long, sorted list of goods that are recommended to him, this type of measure is most appropriate.

The relative ordering of preference values is used in a rank prediction metric, which is independent of the exact values assessed by a recommender. A recommender that consistently overestimates item ratings to be lower than genuine user preferences, for example, might still get a perfect score as long as the ranking is correct.

## Deployment

### **Best Approaches to Deploy a Recommendation System:**

The deployment pattern for any recommender is specific to the volumes and volatility of the data on which they are based. The specific business scenarios enabled with the recommenders also affect how you should implement the recommendation platform. There are Five best Practice to Deploy a Right Model:

* Data Assessment
* Evaluation of the right tech stack
* Robust Deployment approach
* Post deployment support & testing
* Change management & communication

### **Platforms To Deploy Recommendation Systems**

* Amazon SageMaker
* Azure Machine Learning
* Google Cloud AI Platform
* IBM Watson Studio
* Heroku
* Docker
* Etc.

### **Why To Choose Heroku for Testing Recommendation Systems**

Deploying a machine learning model as a service can solve most of these problems, and predictions will be real-time. But there will be issues like scalability, monitoring, and down-time of service. Though there are many cloud providers to resolve these issues and provide 24\*7 support.

Still, if you are a small company or just starting in AI/ML and don’t want to spend more time to handle cloud deployments or DevOps tasks and want a quick deployment option, then deploying your machine learning model on Heroku using Flask will address all your issues. and it is free Cloud Source Platform and very easy to use.

### **Heroku using Flask**

Heroku is a Platform-as-a-Service tool by Salesforce. Heroku is backed by AWS and all Heroku applications/services are hosted on AWS. AWS provides the infrastructure and handles all the load-balancing, resource utilization, networking, logging, monitoring and Heroku acts as a middle-man to provide a scalable, automated rapid deployment platform with all cloud capabilities. Using Flask will provide UI to test and it can be integrated with enterprise-level applications.

### **Steps for Recommendation System deployment on Heroku using Flask:**

Deployment on Heroku using Flask has 7 steps from creating a machine learning model to deployment. These steps are the same for all machine learning models and you can deploy any ML model on Heroku using these steps.

**STEP 1:**

Create Recommendation ML Model and save (pickle) it

![save-machine-learning-model-python.png](data:image/png;base64;base64,)

**STEP 2:**

![1_AqA4VimNlY0ZUFTCyXFfsA.png](data:image/png;base64;base64,)

Create Flask files for UI and python main file (app.py) or with the name of any other .py file

that can unpickle the machine learning model from step 1 and do predictions. we will create flask files — index.html and app.py. index.html, is a flask UI file for providing inputs (or features) to the model. app.py, is a python main file that unpickles the gradient boosting model from Step 1, renders the flask UI page index.html and makes predictions based on input from UI.

**STEP 3:**

Create requirements.txt to setup Flask web app with all python dependencies. In Step 3, we will create requirements.txt to add all of the dependencies for the flask app.

**STEP 4:**

Create Profile to initiate Flask app command. In Step 4, we will create Proc file to specify the commands that are executed by Heroku app on startup.

**STEP 5:**

Commit files from Step 1, 2, 3 & 4 in the GitHub repo

**STEP 6:**

Create account/Login on Heroku, create an app, connect with GitHub repo, and select branch.

![heroku_create_app.png](data:image/png;base64;base64,)

In Step 6, we will log in on Heroku and create a new app. Next, we will connect the GitHub repo created in step 5 to the Heroku app and select a branch.

**STEP 7:**

Select manual deploy (or enable Automatic deploys) on Heroku.

![download.png](data:image/png;base64;base64,)

Finally in Step 7, select manual (or automatic) deploy and you can see the build logs scrolling. Once the application is deployed, you will get the application URL in logs and it will show the success message.

# 

# Conclusion

After of this research and project we changed the way we saw the recommenders.

We did not imagine the complexity of research and calcs involved by a recommendation. We were also able to realize that a bad approach can bring non-realist results and that also all this can be measured by calcs.

And the most important thing this engineering is able to transform results on business are applied.

# 

# References List

Adomavicius, G., and Tuzhilin, A. 2005. Toward the Next Generation of Recommender Systems: A Survey of the State-of-the-Art and Possible Extensions.

IEEE Transactions on Knowledge and Data Engineering 17(6): 734–749.

Bell, R. M., and Koren, Y. 2007. Scalable Collaborative Filtering with Jointly Derived Neighborhood Interpolation Weights. In Proceedings of the 2007 Seventh.

IEEE International Conference on Data Mining, 43–52. Los Alamitos, CA: IEEE Computer Society.

Bell, R. M.; Koren, Y.; and Volinsky, C. 2007. The BellKor Solution to the Netflix Prize. Technical Report, AT&T Labs Research, Forham Park, NJ.

Bennett, J., and Lanning, S. 2007. The Netflix Prize. Paper presented at the KDD Cup Workshop, San Jose, CA, 12 August.

Burke, R. 2002. Hybrid Recommender Systems: Survey and Experiments. User Modeling and User-Adapted Interaction 12(4): 331–370.

Burke, R., and Ramezani, M. 2011. Matching Recommendation Technologies and Domains. In Recommender Systems Handbook, ed. F. Ricci, L. Rokach, B. Shapira, and P. B. Kantor, 367–386. Berlin: Springer

Felfernig, A., and Burke, R. 2008. Constraint-Based Recommender Systems: Technologies and Research Issues. In Proceedings of the 10th International Conference on Electronic Commerce (ICEC ’08). New York: Association for Computing Machinery.\

McGinty, L., and Reilly, J. 2011. On the Evolution of Critiquing Recommenders. In Recommender Systems Handbook, ed. F. Ricci, L. Rokach, B. Shapira, and P. B. Kantor, 419–453.

Berlin: Springer. Resnick, P., and Varian, H. R. 1997. Recommender Systems. Communications of the ACM 40(3): 56–58. Rich, E. 1979.

User Modeling via Stereotypes. Cognitive Science 3(4): 329–354. Rocchio, J. J. 1971.

Relevance Feedback in Information Retrieval. In The SMART Retrieval System: Experiments in Automatic Document Processing, ed. G. Salton, 313–323.

Englewood Cliffs, NJ: Prentice-Hall. Schafer, J.; Frankowski, D.; Herlocker, J.; and Sen, S. 2007. Collaborative Filtering Recommender Systems. In The Adaptive Web, ed. P.\

Brusilovsky, A. Kobsa, and W. Nejdl, 291–324. Lecture Notes in Computer Science 4321. Berlin: Springer.

<https://scholar.harvard.edu/files/sendhil/files/recommenders55.pdf>

<https://ui.adsabs.harvard.edu/abs/2020arXiv201003240C/abstract>

<https://ui.adsabs.harvard.edu/abs/2020arXiv201003240C/abstract>