Machine Learning Final Project: Recommendation Systems on Recipes

Konstantinos Peridis
Department of Electrical and Computer Engineering
University of Thessaly
Volos, 383 34
kperidis@uth.gr

Nikoleta Tsavlidou
Department of Electrical and Computer Engineering
University of Thessaly
Volos, 383 34
ntsavlidou@uth.gr

Abstract

This project aims to develop a comprehensive recommender system for recipes using a combination of collaborative filtering and content-based filtering techniques. The dataset utilized comprises 522,517 recipes and 1,401,982 reviews from 271,907 different users sourced from Food.com. The motivation behind this work is to enhance the user experience on the platform by providing personalized recipe recommendations. The methodology involves implementing collaborative filtering with the scikit-surprise library, content-based filtering using TF-IDF count vectorizer and cosine similarity matrix, and a hybrid approach combining both methods. Additionally, a neural network-based collaborative filtering model using Keras embeddings is implemented. The report presents the exploration of the dataset, the creation of word clouds to represent prominent recipe names and categories, and the implementation of the recommender systems. The models are evaluated using metrics such as RMSE and MAE, providing insights into their performance.

1 Introduction

The problem addressed in this project is the creation of an effective recipe recommender system, motivated by the need to assist users in discovering recipes that align with their preferences. With a dataset comprising a vast collection of recipes and reviews, the goal is to leverage collaborative filtering, content-based filtering, and a hybrid approach to offer personalized recommendations. The input to the system is user reviews and metadata for each recipe, while the output is a list of recommended recipes. The report provides an in-depth exploration of the data, discusses the chosen methodologies, and presents the results of the implemented recommender systems.

The internet page food.com, where we got the dataset from, is a large and extremely easy-to-use collection of recipes with a myriad of recipe options. However, the choice of which one will reach the user's kitchen forces him to face a common dilemma, which leads to a sense of overload due to the abundance of information. In this case, an innovative solution emerges through the application of an algorithm designed not only to suggest exactly what the user wants to cook, but also to reveal options that users had not considered before. This innovative algorithm significantly improves the user experience, especially for people with specific dietary preferences. It tackles the challenge of sorting through a vast array of recipes, presenting customized results that align with the user's dietary restrictions. By discovering new recipes that cater to their unique preferences, users can explore recipes they might not have otherwise encountered. A positive feedback loop is established as users try to positively evaluate these personalized recommendations, contributing to the continuous improvement and enhancement of the Recommendation System in subsequent implementations. The machine learning technique we chose to use is a hybrid that includes both Content-Based Filtering and Collaborative Filtering, as they fully analyze the information in the dataset. Content-based filtering accepts as input a User Data Scoreboard and an Item Profile that can include the category each recipe belongs to, in text format. This information is processed by the algorithm alongside the numerical evaluations. The main problem it faces is the difficulty of making recommendations to new users with few or no reviews on recipes. However, the greatest value is that it excels in accuracy, especially compared to other techniques. At the same time, it perfectly meets the main goal that we have set for our recipe suggestions for each user based on positive evaluations and past impressions. We want to summarize the characteristics of the item and construct a profile for it, then utilize this item's profile to assess the user's inclination towards the attributes outlined in the aforementioned profile. In summary, using a hybrid between content-based filtering and collaborative filtering in a recipe recommendation system enhances personalization, addresses dietary preferences, reduces overload, and adapts to different dietary preferences, leading to more efficient and user-friendly experiences.

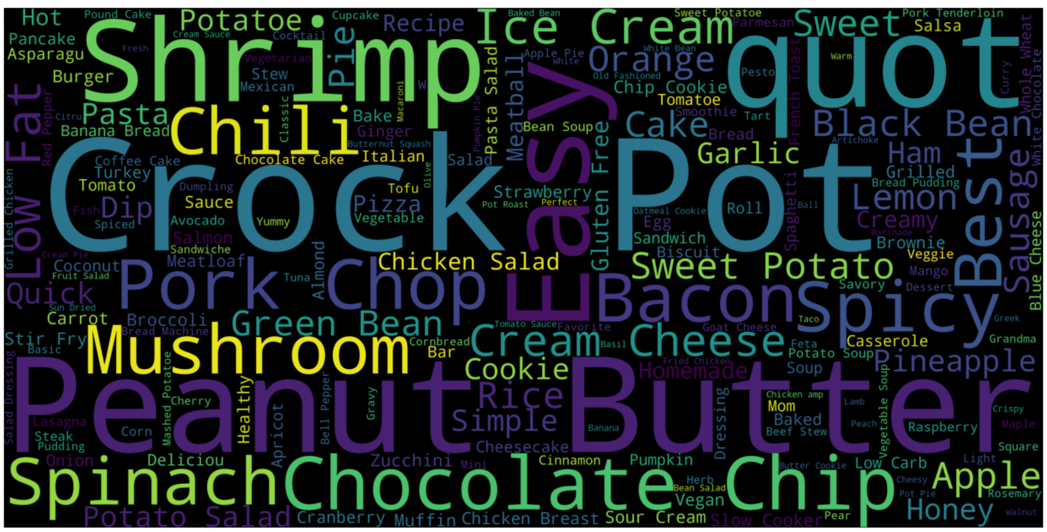


Figure 1: The most prominent recipes using "Wordcloud" representation.

2 Literature Review

The development of machine learning systems has garnered significant attention in recent years, as evidenced by a variety of research and application efforts. This literature review synthesizes knowledge from key sources in the field that we studied before starting to write the code.

"Recommender System Using Machine Learning" by Panarin and Sozinova (2024)

The authors delve into collaborative filtering, content-based filtering, and hybrid approaches, offering a comprehensive understanding of the fusion of these methodologies.

"Implementation of RS using collaborative, content-based and hybrid filtering" (2023)

This resource proves valuable for hands-on experience in collaborative, content-based, and hybrid filtering techniques.

“Movie Recommendation System Using Machine Learning” by Appaji et al. (2023)

Appaji, Patnaik, Kumar, and Kumar contribute to the discussion by discussing the application of machine learning to movie recommendation systems. This resource sheds light on collaborative and content-based filtering methods.

“Advanced Course in Machine Learning” by Google for Developers (2024)

This course was used to expand our knowledge of recommendation systems and helped explain different models used in recommendation, including matrix factorization and deep neural networks.

"Movielens - Content Based and Collaborative Filtering Models" by Khanhnamle

Khanhnamle’s GitHub repository provides practical implementations of content- and collaboration-based filtering models, making them easy to study and understand in recommender systems.

"Using Scikit-Surprise to Create a Simple Recipe Collaborative Filtering Recommender System" by Dor

Dor presents a perspective using the Scikit-Surprise library for a recipe recommendation system. This source contributes information about collective filtering in the area we have chosen to work on.

"KNN-based Extended Collaborative Filtering Adaptive Recommender Services" by Nguyen et al. (2023)

Nguyen et al. explore adaptive extended cooperative filtering based on KNN.

"What Are RMSE and MAE? - Towards Data Science" by Shwetha Acharya (2022)

Shwetha Acharya provides a clear description of Root Mean Squared Error (RMSE) and Mean Absolute Error (MAE), critical metrics for evaluating recommender system performance. This resource was shown for understanding the quantitative evaluation of recommendation models.

"3 Regression Metrics You Should Know: MAE, MSE, and RMSE | Proclus Academy" (2022)

The Proclus Academy further explores regression metrics, emphasizing the importance of MAE, mean squared error (MSE), and RMSE. This resource contributes to a deeper understanding of the metrics involved in evaluating the predictive capabilities of systems of systems.

"Content Based Recommender System Evaluation" by Gabriel Preda (2023)

Gabriel Preda’s Kaggle notebook focuses on evaluating content-based recommender systems. This resource provides information on methodologies and considerations when evaluating the effectiveness of content-based recommendations.

"Pandas" by The pandas development team (2024)

The pandas library, an essential tool in data analysis, is highlighted. As a flexible and widely used library, pandas plays a critical role in data preprocessing and manipulation, contributing to the development of the recommender system.

“Array Programming with NumPy” by Harris et al. (2020)

Harris et al. delve into the importance of NumPy, a library for programming arrays in Python. NumPy plays a key role in efficiently handling large ensembles, making it a key component in implementing machine learning models, including systems of systems.

"Matplotlib: A 2D Graphics Environment" by J. D. Hunter (2007)

J. D. Hunter's work on Matplotlib highlights the importance of visualization in data analysis and model evaluation. Matplotlib serves as a flexible tool for creating visualizations, helping to interpret the output of the recommender system.

"WordCloud: a Cytoscape plugin for creating a visual semantic network summary" by Oesper et al. (2011)

Oesper et al. enter WordCloud, a Cytoscape plugin, emphasizing the importance of visual semantic summaries in network analysis.

"Scikit-learn: Machine Learning in Python" by Pedregosa et al. (2011)

Pedregosa et al. provide insights into Scikit-learn, a powerful machine learning library. Scikit-learn is widely used for building and evaluating machine learning models, including those for recommender systems.

"Surprise: A Python library for recommender systems" by Hug (2020)

Hug introduces Surprise, a Python library specifically designed for recommender systems. This resource highlights the importance of specialized tools for building and evaluating recommendation models.

"TensorFlow: Large-Scale Machine Learning on Heterogeneous Systems" by Abadi et al. (2015)

The work of Abadi et al. in TensorFlow highlights the importance of large-scale machine learning frameworks. TensorFlow has become a cornerstone in the development and deployment of complex machine learning models, including those used in recommender systems.

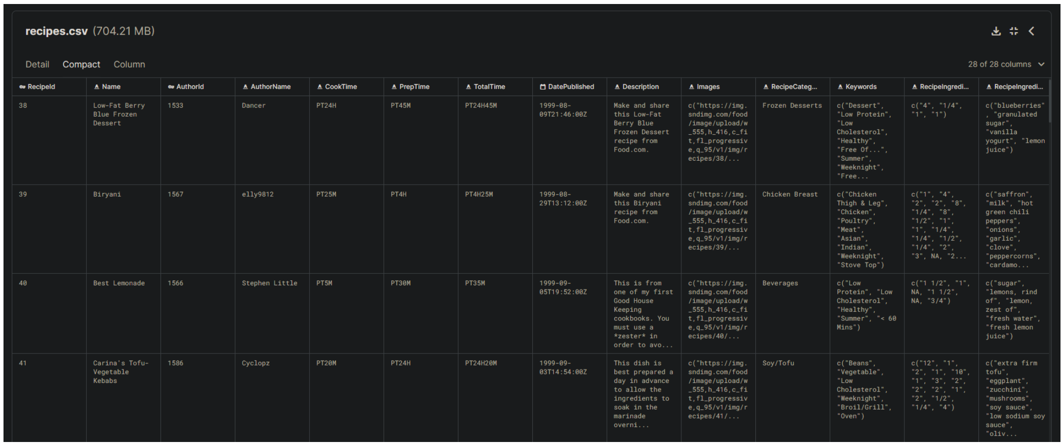
In conclusion, the literature review includes a wide range of resources, ranging from fundamental concepts and evaluation metrics to basic tools and libraries in the machine learning ecosystem, the theoretical and practical study background necessary for the development of our project.

3 Dataset and Features

The dataset comprises 522,517 recipes and 1,401,982 reviews from Food.com. Exploratory data analysis reveals the diversity of recipe categories, with a focus on popular genres such as Desserts, Lunch/Snacks, and One Dish Meal. The dataset contains information on recipe details, user reviews, and nutritional content. Preprocessing involves handling missing values and converting data types.

Each recipe is uniquely identified by a RecipeId and characterized by basic details such as recipe name, author details (AuthorId and AuthorName), CookTime, PrepTime, TotalTime, DatePublished and a short description. Visuals are integrated with Images representing the dish. Recipes are categorized for easy navigation and related keywords improve searchability. The database includes detailed information on RecipeIngredientQuantities and RecipeIngredientParts, ensuring accurate ingredient management. Comments displayed by users are recorded through an Aggregate Rating and Review Count. Nutritional information, including calories, fat content, saturated fat content, cholesterol content, sodium content, carbohydrate content, fiber content, sugar content and protein content, provides transparency about the health features of the recipe. However, in the context of the project we relied on id of the review, recipe and the author, the name of the author, the written review submitted by the user with the rating and the date that was submitted and modified.

The report emphasizes dataset characteristics, preprocessing steps, and feature engineering for model development. It provides a foundation for understanding the motivation, literature context, and technical aspects of the recommender system.



recipes.csv (704.21 MB)

Detail Compact Column 28 of 28 columns

RecipeId	Name	AuthorId	AuthorName	CookTime	PrepTime	TotalTime	DatePublished	Description	Images	RecipeCateg...	Keywords	RecipeIngredi...	RecipeIngredi...
38	Low-Fat Berry Blue Frozen Dessert	1533	Dancer	PT24H	PT45M	PT24H45M	1999-08-09T21:46:00Z	Make and share this Low-Fat Berry Blue Frozen Dessert recipe from Food.com.	c("https://img.sndimg.com/food/image/upload/w_555,h_416,c_fit,fl_progressive,q_95/v1/img/recipes/38/...	Frozen Desserts	c("Dessert", "Low Protein", "Low Cholesterol", "Healthy", "Free Of...", "Summer", "Weeknight", "Free...	c("4", "1/4", "1", "1")	c("blueberries", "granulated sugar", "vanilla yogurt", "lemon juice")
39	Biryani	1567	elly9812	PT25M	PT4H	PT4H25M	1999-08-29T13:12:00Z	Make and share this Biryani recipe from Food.com.	c("https://img.sndimg.com/food/image/upload/w_555,h_416,c_fit,fl_progressive,q_95/v1/img/recipes/39/...	Chicken Breast	c("Chicken Thigh & Leg", "Chicken", "Poultry", "Meat", "Asian", "Indian", "Weeknight", "Stove Top")	c("1", "4", "2", "2", "8", "1/4", "8", "1/2", "1", "1", "1/4", "1/4", "1/2", "1/4", "2", "3", NA, "2...	c("saffron", "milk", "hot green chili peppers", "onions", "garlic", "clove", "peppercorns", "cardamo...
40	Best Lemonade	1566	Stephen Little	PT5M	PT30M	PT35M	1999-09-05T19:52:00Z	This is from one of my first Good House Keeping cookbooks. You must use a *zester* in order to avo...	c("https://img.sndimg.com/food/image/upload/w_555,h_416,c_fit,fl_progressive,q_95/v1/img/recipes/40/...	Beverages	c("Low Protein", "Low Cholesterol", "Healthy", "Summer", "< 60 Mins")	c("1 1/2", "1", NA, "1 1/2", NA, "3/4")	c("sugar", "lemons, rind of", "lemon, zest of", "fresh water", "fresh lemon juice")
41	Carina's Tofu-Vegetable Kebabs	1586	Cyclopz	PT20M	PT24H	PT24H20M	1999-09-03T14:54:00Z	This dish is best prepared a day in advance to allow the ingredients to soak in the marinade overni...	c("https://img.sndimg.com/food/image/upload/w_555,h_416,c_fit,fl_progressive,q_95/v1/img/recipes/41/...	Soy/Tofu	c("Beans", "Vegetable", "Low Cholesterol", "Weeknight", "Broil/Grill", "Oven")	c("12", "1", "2", "1", "10", "1", "3", "2", "2", "2", "1", "2", "1/2", "1/4", "4")	c("extra firm tofu", "eggplant", "zucchini", "mushrooms", "soy sauce", "low sodium soy sauce", "oliv...

Figure 2: A sample of the dataset

4 Methods

About the content based filtering method, the concept of "Term Frequency – Inverse Document Frequency (TF-IDF)" is used as a content basted filtering mechanism in order to determine the relative importance of our recipes. TF is the frequency of a word in a document. IDF is the inverse of the document frequency. The equation to calculate the score is the following:

$$tfidf_{i,j} = tf_{i,j} \times \log \frac{N}{df_i}$$

After calculating TF-IDF scores, we determine which items are closer to each other using the Vector Space Model which computes the proximity based on the angle between the vectors. In this model, each item is stored as a vector of its attributes (which are also vectors) in an n-dimensional space and the angles between the vectors are calculated to determine the similarity between the vectors.

Next, the user profile vectors are also created based on his actions on previous attributes of items and the similarity between an item and a user is also determined in a similar way. We will be using the Cosine Similarity to calculate a numeric quantity that denotes the similarity between two movies. Since we have used the TF-IDF Vectorizer, calculating the Dot Product will directly give us the Cosine Similarity Score.

Regarding the collaborative filtering method, we use the k-Nearest Neighbors method through the scikit-surprise library. This algorithm finds clusters of similar users based on common recipe ratings and makes predictions using the average rating of the top-k nearest neighbors. The prediction of the rating for user u on a recipe I is defined by the following equation:

$$\hat{r}_{u,i} = \hat{r}_u + \frac{\sum_{v \in u_{ui}^K} sim(u,v)(r_{vi} - \bar{r}_v)}{\sum_{v \in u_{ui}^K} |sim\,(u,v)|}$$

where u_{ui}^K denotes the set of K nearest neighbors of user u who has rated item i, r_{vi} is the actual rating given by neighbor v for item i, and $\bar{r}_u$ and $\bar{r}_v$ are the average ratings of each user (u and neighbor v) calculated based on their rating history. The similarity sim(u,v) between users is calculated using a distance metric, which is the cosine similarity in our case. The value of K in a KNN-based collaborative filtering algorithm is usually fixed and determined prior to the recommendation process.

Regarding the content based filtering model the evaluation needs to be done manually. We will choose a few recipes and judge whether the recommendations based on these make sense in order to evaluate the model. In this part the scikit-surprise library is employed, a Python library specifically designed for building and analyzing recommender systems.

Incorporating content-based filtering further enriched the recommender system's capabilities. This technique analyzes item characteristics and user preferences, ensuring that recommendations align with the user's historical interactions and preferences. The inclusion of content-based filtering contributes to a more nuanced and personalized recommendation mechanism. Furthermore TF-IDF (Term Frequency-Inverse Document Frequency) technique in conjunction with cosine similarity vectorization utilized on recipe categories, creating a similarity matrix based on cosine similarity, measures the cosine of the angle between two non-zero vectors. TF-IDF is a numerical statistic that reflects how important a word is to a document in a collection or corpus. This approach facilitated the identification of patterns and similarities within the textual content, forming the basis for generating accurate recommendations to represent the significance of different recipe categories.

For collaborative filtering, the Surprise library was employed, specifically utilizing the KNNBasic collaborative filtering algorithm. The seamless integration of this library and algorithm spared the need for extensive manual intervention, providing reliable recommendations without the necessity for additional adjustments. Embeddings are a way to represent categorical variables as continuous vectors in neural networks. In this case, Keras embeddings are used to capture latent factors in user-item interactions. This involved creating dense representations of items and users, enhancing the model's ability to understand intricate patterns and relationships within the data and embeddings help the model learn hidden patterns or features in the collaborative filtering process. The integration of Keras embeddings was a pivotal step in refining the recommender system's performance. Beyond the specified requirements, certain optimizations were introduced.

In summary, collaborative filtering involves using a subset of the dataset with the scikit-surprise library. Content-based filtering involves representing recipe categories using TF-IDF vectors, creating a similarity matrix based on cosine similarity. The neural network model utilizes Keras embeddings to capture latent factors in user-item interactions for collaborative filtering. These enhancements encompassed refining the recommendation algorithms, incorporating user feedback loops for continuous improvement, and addressing potential scalability challenges to ensure the system's robustness in handling larger datasets.

5 Experiments/Results/Discussion

To assess the effectiveness of the recommender system, various evaluation metrics are employed. Common metrics include Mean Squared Error (MSE) for collaborative filtering models and precision, recall, and F1-score for content-based and neural network models. These metrics provide insights into the accuracy and relevance of the recommendations, guiding further improvements to the system.

Results from each model are presented and compared to evaluate their performance. The collaborative filtering model demonstrates its ability to make accurate predictions based on user-item interactions. Content-based filtering enhances diversity in recommendations, while the neural network model excels in capturing complex patterns. The trade-offs and strengths of each model are discussed, guiding potential enhancements in subsequent iterations.

The experiments carried out (which are also presented in the jupyter notebook) consist of testing the content based filtering mechanism separately from the collaborative filtering mechanism. The methodology of testing was to evaluate the RMSE, Root of the Mean of the Square of Errors, expressed mathematically as:

$$RMSE = \sqrt{\frac{\sum (y_i - y_p)^2}{n}}$$

and the MAE, Mean of Absolute value of Errors expressed as:

$$\frac{|(y_i \; - \; y_p)|}{n}$$

The evaluation yielded the following results:

RMSE	1.24
MAE	0.83

This means that if we were to call the metric of reviews a "star", the RMSE is 1.24 stars, while the MAE is 0.83 stars. The impact of this margin of error is debatable and depends on the way each and every one of the users ranks a recipe. For some users the review is important even up to the second decimal point. Some other users may not scrutinize their review that much and may just leave 5 star reviews for something they like and or 1-2 star reviews for something they did not like. For these people, this margin of error is bearable, so it does not have a negative impact on our predictions.

Regarding the content based filtering evaluation, we cannot evaluate it in the same way as the collaborative filtering model. For this reason, the evaluation is done manually. One of the examples presented also in the jupyter notebook is the one where we ask our recommender system to recommend a few recipes given the recipe "Chocolate Bourbon Pecan Pie". The recommendations made were the following:

Morton's Key Lime Pie
Simple Diabetic Pumpkin Pie
Yummy Vanilla Pie with Strawberries
Chocolate Chip Peanut Butter Pie
Upside Down Coconut Cream Pie
Miniature Mincemeat Pies
Sugar Pie
Strawberry Pie: Simple and Southern
Stupidly Simple Coconut Cream Pie
Pistachio Cream Cheese Whip
Lighter Pecan Pie
Look No Further! Apple Pie

We can safely conclude from the above results that our content based recommendation system works as expected. We gave as an input a recipe that is a dessert pie and it recommends other desserts in the pie category. We got a reasonable result, so our evaluation is complete and suggests that our model will successfully predict other recipes.

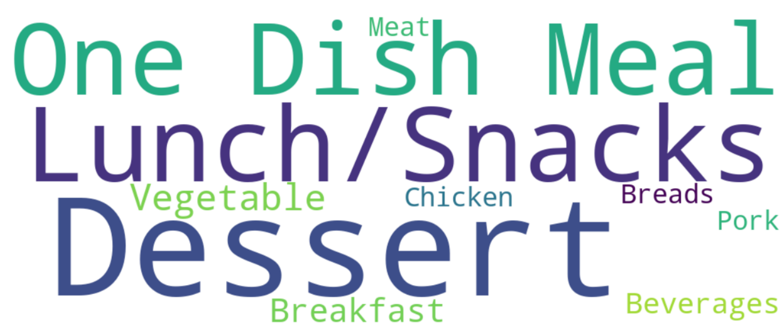


Figure 3: Most common recipe categories in "Wordcloud" representation.

6 Conclusion/Future Work

6.1 Future Work

Our forthcoming initiatives in advancing the project's development include: The integration of the remaining elements of the data set in order to maximize the personalization of the proposals to each user. Consider factors such as favorite ingredients, kitchen items and cooking difficulty levels, calories or nutritional value. Incorporation of a feature that suggests recipes based on seasonal availability of ingredients or the user's location. This brings numerous benefits, including contextual relevance, variety, connection to local culinary traditions, sustainability, increased user engagement, and enhanced practicality. This feature enriches the user experience by providing recommendations that align with the natural and cultural context of the user's environment. Development of a feature that recommends complementary recipes, suggesting side dishes, beverages, or desserts that pair well with the user's selected main course that complement the main course they have chosen. This aims to make the meal planning process more convenient and enjoyable for users.

6.2 Conclusion

In conclusion, the development of the recipe recommendation system has been a gratifying and enlightening journey. Through the integration of collaborative filtering, content-based filtering, and neural networks, the system has evolved to provide a comprehensive and personalized culinary experience for our users. The evaluation metrics provide insights into model performance, guiding future enhancements. We hope in the future to both delve into the field of machine learning and in particular the recommendation systems we studied alongside the development of our idea for a recipe protocol system.

Contributions

Both of us contributed to writing and proofing of the project's final report. The main parts of the report were discussed and were written by both of us, while the sections that we took up by ourselves were written by whoever did the work. The research topic and its dataset were found and acquired by Nikoleta, who also made sure to analyze and explore the data so it is usable (e.g. if there were missing values they should be handled accordingly). Kostas took up the task of implementing the collaborative and content based filtering systems. In addition, he evaluated these models using the methods described in the appropriate section of this report. Lastly, Nikoleta implemented the collaborative filtering using the Keras embeddings method. Overall, during the whole course of the development of the jupyter notebook and the written report, the team made sure to communicate and exchange ideas effectively, while discussing any concerns that arised.

References/Bibliography

[1] Panarin, Roman & Sozinova, Sofya (2024). Recommender System Using Machine Learning. https://maddevs.io/blog/recommender-system-using-machine-learning/.

[2] Unknown Author(s) (2023). Implementation of RS using collaborating, content based, and hybrid filtering. https://colab.research.google.com/drive/11P0Vk7TdPhVphLKTfLL72Kh-0j2RmtL6scrollTo=rT1K406ZuF2HprintMode=true.

[3] S. Vidya Sagar Appaji & M. Kusuma Patnaik & M. Dileep Nagendra Kumar & K. Manoj Kumar & K. Satish (2023). Movie recommendation system using machine learning AIP Conf. Proc. 2794, 020028 (2023)

[4] Unknown Author(s) (2024). Machine Learning | Google for Developers https://developers.google.com/machine-learning/recommendation

[5] Khanhnamle. "Movielens - Content Based and Collaborative Filtering Models" GitHub, github.com/khanhnamle1994/movielens/blob/master/Content_Based_and_Collaborative_Filtering_Models.ipynb.

[6] Dor. Using Scikit-Surprise to Create a Simple Recipe Collaborative Filtering Recommender System. – Data Science Portfolio. 8 Feb. 2022, www.alldatascience.com/recommender-systems/simple-recipe-recommender-system-with-scikit-surprise.

[7] Nguyen, Luong Vuong, et al. "Adaptive KNN-Based Extended Collaborative Filtering Recommendation Services." Big Data and Cognitive Computing, vol. 106, no. 2, 31 May 2023, doi:10.3390/bdcc7020106.

[8] Acharya, Shwetha. "What Are RMSE and MAE? - Towards Data Science." Medium, 6 Jan. 2022, towardsdatascience.com/what-are-rmse-and-mae-e405ce230383.

[9] 3 Regression Metrics You Must Know: MAE, MSE, and RMSE | Proclus Academy. 14 Oct. 2022, proclusacademy.com/blog/explainer/regression-metrics-you-must-know.

[10] Preda, Gabriel. "Content Based Recommender System Evaluation." Kaggle, 4 May 2023, www.kaggle.com/code/gpreda/content-based-recommender-system-evaluation.

[11] The pandas development team. Pandas-dev/pandas: Pandas. v2.2.0, Zenodo, 20 2024, doi:10.5281/zenodo.10537285.

[12] Harris, C.R., Millman, K.J., van der Walt, S.J. et al. Array programming with NumPy. Nature 585, 357–362 (2020). DOI: 10.1038/s41586-020-2649-2

[13] J. D. Hunter, "Matplotlib: A 2D Graphics Environment", Computing in Science Engineering, vol. 9, no. 3, pp. 90-95, 2007.

[14] Oesper, L., Merico, D., Isserlin, R., Bader, G. D. (2011). WordCloud: a Cytoscape plugin to create a visual semantic summary of networks. Source Code for Biology and Medicine, 6(1), 7.

[15] Scikit-learn: Machine Learning in Python, Pedregosa et al., JMLR 12, pp. 2825-2830, 2011.

[16] Hug, N., (2020). Surprise: A Python library for recommender systems. Journal of Open Source Software, 5(52), 2174, https://doi.org/10.21105/joss.02174

[17] Martín Abadi, Ashish Agarwal, Paul Barham, Eugene Brevdo, Zhifeng Chen, Craig Citro, Greg S. Corrado, Andy Davis, Jeffrey Dean, Matthieu Devin, Sanjay Ghemawat, Ian Goodfellow, Andrew Harp, Geoffrey Irving, Michael Isard, Rafal Jozefowicz, Yangqing Jia, Lukasz Kaiser, Manjunath Kudlur, Josh Levenberg, Dan Mané, Mike Schuster, Rajat Monga, Sherry Moore, Derek Murray, Chris Olah, Jonathon Shlens, Benoit Steiner, Ilya Sutskever, Kunal Talwar, Paul Tucker, Vincent Vanhoucke, Vijay Vasudevan, Fernanda Viégas, Oriol Vinyals, Pete Warden, Martin Wattenberg, Martin Wicke, Yuan Yu, and Xiaoqiang Zheng. TensorFlow: Large-scale machine learning on heterogeneous systems, 2015. Software available from tensorflow.org.