Contrastive Explanations for Recommendation Systems

Meir Nizry (meir.nizri@msmail.ariel.ac.il)
Computer Science Dept., Ariel University, Israel

Amos Azaria (amos.azaria@ariel.ac.il)
Computer Science Dept., Ariel University, Israel

Noam Hazon (noamh@ariel.ac.il)
Computer Science Dept., Ariel University, Israel

Abstract

Recommendation systems are widely used and are present in many applications, such as movie recommendation, product sales, and content providers. However, current recommendation systems are usually stern and lack the ability to explain their decisions or allow the users to question them.

In this paper, we develop an automatic method that generates contrastive explanations for recommendation systems based on items features and users preferences. That is, once receiving a recommendation (e.g., to buy a Samsung S22), the users will have the option to ask the system why it did not recommend a specific different item (e.g., Xiaomi 12). With our method the system have the ability to reply with a meaningful and convincing personalized explanation (e.g., it might seem that a good camera is very important to a specific user, and the Samsung S22 includes a better camera than the Xiaomi 12).

We run an experiment with 100 human participants and show that when applying our method, humans are more convinced that the recommended item is better for them, than when using contrastive explanations generated by other methods.

1 Introduction

Recommendation systems are becoming significant in extensive aspects of our daily lives. In some areas, such as medicine (Suphavilai, Bertrand, & Nagarajan, 2018) and finance (Zibriczky, 2016), recommendation systems are used as tools that help making decisions at high stakes. In these settings, the system is expected to justify its recommendations, since the cost of a wrong decision may be high. However, state-of-art techniques used in recommendation systems are becoming more and more inscrutable due to the complexity of models and high dimension of data on which the models are trained. Arguably, the most significant means to increase trust and transparency in recommendation systems is by providing explanations for the system's decisions that are concise and understandable to humans (Tintarev & Masthoff, 2015).

In order to explain the outputs of a recommendation system, two strategies that are common in the field of explainable AI can be applied. The first is to adopt interpretable models, whose decision mechanism is transparent, and thus, we can naturally provide explanations for the model decisions (Abdollahi & Nasraoui, 2017; Bauman, Liu, & Tuzhilin, 2017). Explanations of this type are called ante-hoc, and they seek to understand the inner workings of a model while the model is in the process of making decisions. However, they often suffer from lower prediction accuracy, which is known as the trade-off between complexity and accuracy (Doshi-Velez & Kim, 2017). In addition, when recommendations are provided as an external service, the inner workings of the system are the intellectual property of the service provider and cannot be revealed. Therefore, in this work we use the second strategy, which is to provide post-hoc explanations, i.e., after the recommendation has been given. Post-hoc explanations do not precisely reflect the underlying recommendation model. Instead, they present rationale, plausible, and valuable information to the user. Consequently, post-hoc explanations are independent of the main recommendation algorithm, and therefore they have the potential to be used across a greater diversity of recommendation systems.

Indeed, what constitutes good explanations has been extensively studied in social science, cognitive science, and psychology (Nozick, 1983; Lombrozo, 2006). In particular, P. Lipton (1990) investigated different types of explanations, and concluded that they should be contrastive. That is, instead of explaining why an event happened, explanations are more convincing if the person receiving the explanation can ask why one event happened rather than another specific event. We follow this idea, and develop a method that generates contrastive explanations for recommendation systems. A contrastive explanation in recommendation system allow the user to pose questions of the form "why did you recommend this item rather than another specific item?". With our method the system can reply with a meaningful and convincing personalized explanation which is based on the features of the recommended item, the contrasting item or the user's preferences. To the best of our knowledge, currently, no recommendation system allows users to pose contrastive questions.

Given a user who was recommended an item p, the user may ask the system why it did not recommend another item q. The essence of our explanation is to select a set of features and show the user the differences between p and q in the selected features. For example, suppose that the item p is the Galaxy S22 cell phone and the item q is the iPhone 13 cell phone. If the selected set of features is price and main camera resolution, then the generated explanation should be "The recommended cell phone Galaxy S22 costs \$528, compared to iPhone 13, which costs \$699; The recommended cell phone Galaxy S22 main camera's resolution is 50 MP, compared to a resolution of 12 MP in the iPhone 13".

In order to select meaningful and convincing set of features, we develop an algorithm, which we termed *CX-RS*. CX-RS measures the influence of each item feature by sampling hypothetical items that are close (in terms of their feature values) to p, q, and in the area between them. The new items are used to train an interpretable model (e.g., linear regression, decision tree) from which we derive weights that represent the importance of each feature in the area between p and q. We then iterate over the features, and select only the first set of features such that if we replace their values in q to their values in p, the recommendation system will rank q as equal or higher than p.

In order to evaluate the performance of CX-RS, we conducted a survey with human participants. The data we used in the survey is a cell phone dataset. Each participant in the survey was presented with 10 randomly selected cell phones with all the necessary features to evaluate them. The participant was asked to rate each cell phones and provide demographic information. Then, two cell phones were presented. The first is the most recommended cell phone for the participant, selected by a state-of-the-art recommendation system. The participant was told that the second cell phone was preferred by a user with preferences similar to his. Each participant was then presented with four explanations why the recommended cell phones is better for him. One of the explanations was generated by CX-RS and the rest served as baselines. Finally, the participant was asked to rate each explanations by indicating to what extent the explanation convinces him that the recommended cell phone is indeed better for him. Overall, 100 different people participated in the survey. The explanations that were generated by CX-RS achieved higher average rating compared to the other explanations.

To summarize, the main contribution of this paper is that it provides a method that can generate more human-like explanation to recommendation systems and can be applied to any feature-based recommendation system models. More specifically, we propose a method that generates post-hoc contrastive explanation for recommendation systems.

2 Related work

Our work is related to the field of Explainable AI (XAI) (Core et al., 2006; Gunning et al., 2019). In a typical XAI setting, the goal is to explain the output of an AI system to a human. This explanation is important for allowing the human to trust the system, better understand, and to allow transparency of the system's output (Adadi & Berrada, 2018). Other XAI systems are designed to provide explanations, comprehensible by humans, for legal or ethical reasons (Doran, Schulz, & Besold, 2017). For example, an AI system for the medical domain might be required to explain its choice for recommending the prescription of a specific drug (Holzinger, Biemann, Pattichis, & Kell, 2017).

Explanation methods in AI can be classified into local explanation methods and global explanation methods (Z. C. Lipton, 2018). Global explanation methods can explain the entire model behavior, how each feature can influence the results of the model. On the contrary, local explanation methods zoom in on a single instance to examine how the output has been generated. Local explanation offering more personalised explanations, and therefore we choose this method in our work.

Based on the model adopted in the recommendation system, explanations can include different types of information to explain the output of the recommendation system. Similar user/item style explanations (Herlocker, Konstan, & Riedl, 2000), feature-based explanations (Ferwerda, Swelsen, & Yang, 2018), social explanations (Quijano-Sanchez, Sauer, Recio-Garcia, & Diaz-Agudo, 2017), and context-aware explanations (Li, Chen, & Dong, 2021). In our explanation, we focus on items features since they can show the difference between two items in the most simplest and yet meaningful way.

There are several types of post-hoc explanations in recommendation systems. One common type provides explanations that answer why a specific item was recommended. Specifically, Peake and Wang (2018) presented a post-hoc approach that extracts explanations from latent factor-based recommendation systems, by training association rules on the output of a matrix factorization black-box model. Nóbrega and Marinho (2019) introduced LIME-RS, which was later improved by Chanson et al. (2021). LIME-RS is an adaptation of LIME to recommendation systems. LIME is a popular approach for explaining a machine learning model's output by identifying the top-n features that have the greatest impact on the model's output (Ribeiro, Singh, & Guestrin, 2016). SHAP is another common approach for explaining a machine learning model's output; it relies on computing the average, over all permutations, of the marginal contributions of each feature (Lundberg & Lee, 2017). Indeed, Guo et al. (2021) adapt SHAP to derive post-hoc explanations for recommendation systems. Shmaryahu et al. (2020) use a set of simple, easily explainable recommendation algorithms to provide post-hoc explanations for a complex recommendation system. Their method attempts to find a simple explainable recommendation system that agrees with the complex model on a recommended item and applies the explanation of the simple model.

Another type of explanation is counterfactual explanations, which consider changes to features and events that alter the outputs of the recommendation system (Wachter, Mittelstadt, & Russell, 2017). A typical counterfactual explanation describes a causal situation: "If the specific event did not occur, a different item would have been recommended". Unlike methods that try to approximate the original models, counterfactual explanations examine changes in the output of the original model and therefore have high fidelity (Kaffes, Sacharidis, & Giannopoulos, 2021). In the recommendation system domain, there exists some works that aim to provide counterfactual explanations to explain recommendations. Ghazimatin et al. (2020) introduce PRINCE, which provides an explanation by finding a set of minimal actions performed by the user that, if removed, changes the recommendation to a different item. Given a recommendation, PRINCE uses a polynomial-time optimal algorithm for finding this minimal set of a user's actions from an exponential search space, based on random walks over dynamic graphs. This approach was later improved by Kaffes et al. (2021), who used normalized length and the importance of a candidate to guide the search. Zhong and Negre (2022) provides counterfactual explanation by sorting all features according to their SHAP values, and selects the minimum of the first features that, if we change their value to another random value, the recommendation system will change its recommendation.

Our proposed work should not be confused with interactive recommendation systems (He, Parra, & Verbert, 2016; Gao, Lei, He, de Rijke, & Chua, 2021), a field that focuses on updating the system-model based on interaction with the user and does not consider interactive explanations as we do.

3 CX-RS

Given a user, we assume that the goal of a recommendation system is to recommend an item that best fits the user preferences. Thus, the recommendation system rates each item and the item with the highest rating is recommended to the user. In our setting, the recommendation system uses a set of item features F, for example the item's brand or price; the features can be numeric or categorical. The recommendation system is not limited to using the item features and may also use features of the user (e.g., demographic information) and additional data (e.g., collaborative-based).

We developed CX-RS, an algorithm that generate post-hoc contrastive explanations that be applied to any feature-based recommendation system models. Given a user who was recommended an item p, the user may ask the system why it did not recommend another item q, and the system must be able to reply with a convincing explanation that is suited to the user's preferences. We propose to select a set of features in order to generate an explanation. The essence of the explanation is to show the user the differences between p and q in the selected features.

In CX-RS algorithm we propose a sampling-based approach, which measures the influence of each feature by sampling hypothetical items that are close (in terms of their feature values) to p, q, and in the area between them. We first generate new items. For each new item, we independently sample each of its features from a uniform distribution between p's value for that feature and q's value for it. That is, for every new item x, each feature $f \in F$ of x, denoted by x_f, is independently sampled from $U\{\min\{p_f, q_f\}, \max\{p_f, q_f\}\}$. In addition, the rating of the recommendation system for each new item is computed. The new items are used to train an interpretable model (e.g., linear regression, decision tree). The model is used to derive weights, which their absolute values represent the importance of a unit of each feature in the area between p and q. We multiply these weights, for each feature f, by the distance $p_f - q_f$, which will provide the explanatory power of each feature.

We then iterate over the features, where they are ordered by the explanatory powers (in descending order). We select only the first set of features such that if we replace their values in q to their values in p, the recommendation system will rank q as equal or higher than p (see Algorithm 1). Intuitively, this approach finds the disadvantages of q such that if they are addressed (using the values of the features in p), q will be ranked as equal or higher than p. We consider the opposite approach, i.e., replacing the values of features in p with their values in q. Intuitively, this approach finds the advantages of p. We also consider limiting the number of selected features by some constant, k. As mentioned before, the explanation is generated from this set of features.

Algorithm 1: Sampling-based approach

```
Input : Recommendation system $R$; Set of item
        features $F$; User $u$; Recommended item $p$;
        Contrastive item $q$; Number of samples $n$.
Output: A set of explanatory features.
$X, E \leftarrow \emptyset$
for $i \leftarrow 1$ to $n$ do
    for $f \in F$ do
        $x_f \leftarrow$ sample from
            $U\{\min\{p_f, q_f\}, \max\{p_f, q_f\}\}$
    $X \leftarrow X \cup x$
$Y \leftarrow R(X, u)$
train an interpretable RS, $T$, on $(X, Y)$
$W \leftarrow T$'s weights multiplied by $(p - q)$
for $f \in F$ sorted in descending order according to $W$
  do
    $E \leftarrow E \cup f$
    $q_f \leftarrow p_f$
    if $R(q, u) \geq R(p, u)$ then
        break
return $E$
```

This approach output contrastive explanations since it answer the question "why p and not q"?. This explanation is also selected as only small set of features values presented. Furthermore, the explanations are expected to have high fidelity, since they do not try to approximate the original models but examine changes in its output.

4 Experimental design

We evaluated the explanation generated by CX-RS through a survey on Mechanical Turk (Paolacci, Chandler, & Ipeirotis, 2010). This survey requires an item feature-based dataset with user ratings. Unfortunately, many of the available rating datasets focus on content recommendations (e.g., movies and books) and thus they have few numerical features. Therefore, these datasets are less suitable for our setting.

Therefore, we composed our own dataset. To that end, we first collected data on the most popular cell phones in the US in 2022[1]. The data for each cell phone consists of the most important features such as performance rating (AnTuTu), memory size, camera's resolution, battery size, screen size, release date, etc. We collected the price of each cell phone from Amazon and Best-Buy (in Aug 22). In order to elicit the ratings, we conducted a survey on Mechanical Turk. Each participant was presented with 10 random cell phones, and she was asked to indicate how likely she is to purchase each of the cell phones at the given price, on a scale from 1 (very unlikely) to 10 (very likely). We also asked each participant to add personal information: age, gender and occupation. Overall, in our dataset there are 34 cell phones with 13 features.

In our main survey, we evaluated the explanation generated by CX-RS. We ran a similar survey on Mechanical Turk, in which the participants were asked to rate 10 cell phones and provide demographic information. Then, two cell phones were presented. The first is the most recommended cell phone for the participant, selected by a state-of-the-art recommendation system. The participant was told that the second cell phone was preferred by users with preferences similar to his. We output the recommended cell phone using two state-of-the-art recommendation system models.

- *MeLU* (Lee, Im, Jang, Cho, & Chung, 2019), a meta-learning-based recommendation system that designed to alleviate the cold-start problem, which is the problem of recommending items to users who consumed only a few items. From meta-learning (learning to learn), which can rapidly adopt new task with a few examples, MeLU can estimate the user preferences using only a few items.

- *Wide & Deep* (Cheng et al., 2016), which is a recommendation system that combines the benefits of a linear (”wide”) model and a deep learning model. A linear model with a wide set of features can memorize the interaction between the features, while a deep neural network can generalize better to unseen feature combinations through low-dimensional dense embeddings learned for the sparse features. Wide & Deep learning jointly trains wide linear models and deep neural networks to combine the benefits of memorization and generalization for recommendation systems.

We compare our explanation with two baselines. The first baseline randomly select features whose values in p and q are not equal. In the second baseline a linear regression model is trained on all the items that a user rated, and the features with the highest absolute weights will be selected. The features are ordered in descending order according to the weights of the linear model. In both baselines, the number of features selected will be the same as the number of features our explanation select to present to the user.

For the first survey, in which we collected ratings on cellphones, we set the reward to $0.5. We recruited 100 participants, of them, 51 were males, 46 females, and 3 participants who chose not to specify their gender.The average age of the participants was 36. In the main survey, which compared CX-RS with the two other baselines, the reward for each participant was $0.7. Here we also recruited 100 participants, half received a recommendation according to Melu and the other half according to Wide & Deep. Of the 100 participants 47 were males and 53 females. The average age of the participants was 38. In both surveys we set a requirement on Mechanical Turk that the approval rate of the workers must be at least 99% and did not require the Turkers to be masters.

5 Results

Figures 1,2 compares the performance, in terms of average ratings, of CX-RS with that of the baselines in both Melu and Wide & Deep recommendation systems. As depicted by the figures, the explanations that were generated by CX-RS outperformed all other explanations. These differences are statistically significant ($p < 0.01$; using a student t-test). That is, the human participants, when presented with explanations according to our method, are more convinced that the recommended item is better for them, than when using contrastive explanations generated by other methods.

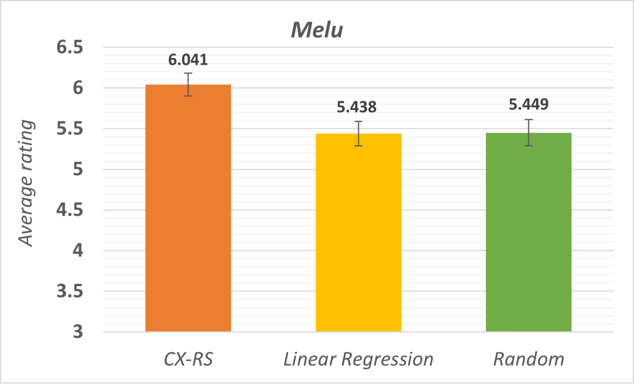

Figure 1: Average rating for Melu for every type of explanation. Error bars present the standard error.

6 Conclusion and Future Work

To the best of our knowledge, we propose the first method that allows users to pose contrastive questions to recommendation systems. our algorithms, CX-RS, can be embedded in any feature-based recommendation system models. This will enable users to pose contrastive questions to a recommendation system, a significant feature missing in current recommendation systems. Our method make the systems more attentive and not only provide a recommendation but also allow the user to interact with them and question their decisions.

We showed that when applying our method, humans are more convinced that the recommended item is better for them,

[1] https://www.theverge.com/22163811/best-phone. https://www.techadvisor.com/article/724318/best-smartphone.html. https://www.tomsguide.com/best-picks/best-phones.

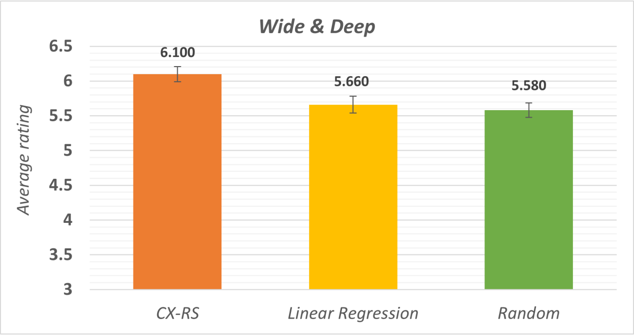

Figure 2: Average rating for Wide & Deep for every type of explanation. Error bars present the standard error.

than when using contrastive explanations generated by other methods.

In future work, we intend to allow the user to further ask why the system did not recommend another, different item, q'. A naive approach, which can be used as a baseline, is to use our approach from the first phase and find features that explain the advantages of p over q'. However, we hypothesize that an algorithm that considers also the previously queried item q along with the previously provided explanation, will result in better explanations for the user. Clearly, the interaction between the user and the system may include several contrastive queries, and we intend to develop algorithms that consider *all* previously contrastive queries and previously provided explanations to generate a new explanation.

Another setting that we intend to tackle in the future is to allow the user to post multiple contrastive queries simultaneously. Our algorithm should provide a more holistic explanation, accounting for all the contrastive queried items together.

Acknowledgment

This research was supported in part by the Ministry of Science, Technology & Space, Israel.

References

Abdollahi, B., & Nasraoui, O. (2017). Using explainability for constrained matrix factorization. In *Proceedings of the eleventh acm conference on recommender systems* (p. 79-–83).

Adadi, A., & Berrada, M. (2018). Peeking inside the black-box: A survey on explainable artificial intelligence (XAI). *IEEE Access*, *6*, 52138-52160.

Bauman, K., Liu, B., & Tuzhilin, A. (2017). Aspect based recommendations: Recommending items with the most valuable aspects based on user reviews. In *Proceedings of the 23rd acm sigkdd international conference on knowledge discovery and data mining* (p. 717-–725).

Chanson, A., Labroche, N., & Verdeaux, W. (2021). Towards local post-hoc recommender systems explanations. In *Proceedings of the 23rd international workshop on design, optimization, languages and analytical processing of big data (dolap).*

Cheng, H.-T., Koc, L., Harmsen, J., Shaked, T., Chandra, T., Aradhye, H., … Shah, H. (2016). Wide & deep learning for recommender systems. In *Proceedings of the 1st workshop on deep learning for recommender systems* (p. 7—10).

Core, M. G., Lane, H. C., Van Lent, M., Gomboc, D., Solomon, S., & Rosenberg, M. (2006). Building explainable artificial intelligence systems. In *Aaai* (pp. 1766–1773).

Doran, D., Schulz, S., & Besold, T. R. (2017). What does explainable AI really mean? a new conceptualization of perspectives. *arXiv preprint arXiv:1710.00794*.

Doshi-Velez, F., & Kim, B. (2017). Towards a rigorous science of interpretable machine learning. *arXiv: Machine Learning*.

Ferwerda, B., Swelsen, K., & Yang, E. (2018). Explaining content-based recommendations. *New York*, 1–24.

Gao, C., Lei, W., He, X., de Rijke, M., & Chua, T.-S. (2021). Advances and challenges in conversational recommender systems: A survey. *AI Open*, *2*, 100-126.

Ghazimatin, A., Balalau, O., Saha Roy, R., & Weikum, G. (2020). Prince: Provider-side interpretability with counterfactual explanations in recommender systems. In *Proceedings of the 13th international conference on web search and data mining* (pp. 196–204).

Gunning, D., Stefik, M., Choi, J., Miller, T., Stumpf, S., & Yang, G.-Z. (2019). Xai—explainable artificial intelligence. *Science robotics*, *4*(37).

Guo, M., Yan, N., Cui, X., Hughes, S., & Al Jadda, K. (2021). Online product feature recommendations with interpretable machine learning. *arXiv preprint arXiv:2105.00867*.

He, C., Parra, D., & Verbert, K. (2016). Interactive recommender systems: A survey of the state of the art and future research challenges and opportunities. *Expert Systems with Applications*, *56*, 9–27.

Herlocker, J. L., Konstan, J. A., & Riedl, J. (2000). Explaining collaborative filtering recommendations. In *Proceedings of the 2000 acm conference on computer supported cooperative work* (pp. 241–250).

Holzinger, A., Biemann, C., Pattichis, C. S., & Kell, D. B. (2017). What do we need to build explainable AI systems for the medical domain? *arXiv preprint arXiv:1712.09923*.

Kaffes, V., Sacharidis, D., & Giannopoulos, G. (2021). Model-agnostic counterfactual explanations of recommendations. In *Proceedings of the 29th acm conference on user modeling, adaptation and personalization* (pp. 280–285).

Lee, H., Im, J., Jang, S., Cho, H., & Chung, S. (2019). Melu: Meta-learned user preference estimator for cold-start recommendation. *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*.

Li, L., Chen, L., & Dong, R. (2021). Caesar: context-aware explanation based on supervised attention for service recommendations. *Journal of Intelligent Information Systems*, *57*, 147-170.

Lipton, P. (1990). Contrastive explanation. *Royal Institute of Philosophy Supplement*, *27*, 247—266.

Lipton, Z. C. (2018). The mythos of model interpretability: In machine learning, the concept of interpretability is both important and slippery. *Queue*, *16*(3), 31–57.

Lombrozo, T. (2006). The structure and function of explanations. *Trends in Cognitive Sciences*, *10*, 464–470.

Lundberg, S. M., & Lee, S.-I. (2017). A unified approach to interpreting model predictions. In *Proceedings of the 31st international conference on neural information processing systems* (p. 4768-–4777).

Nóbrega, C., & Marinho, L. (2019). Towards explaining recommendations through local surrogate models. In *Proceedings of the 34th acm/sigapp symposium on applied computing* (pp. 1671–1678).

Nozick, R. (1983). *Philosophical explanations*. Harvard University Press.

Paolacci, G., Chandler, J., & Ipeirotis, P. G. (2010). Running experiments on amazon mechanical turk. *Judgment and Decision making*, *5*(5), 411–419.

Peake, G., & Wang, J. (2018). Explanation mining: Post hoc interpretability of latent factor models for recommendation systems. In *Proceedings of the 24th acm sigkdd international conference on knowledge discovery & data mining* (pp. 2060–2069).

Quijano-Sanchez, L., Sauer, C., Recio-Garcia, J. A., & Diaz-Agudo, B. (2017). Make it personal: a social explanation system applied to group recommendations. *Expert Systems with Applications*, *76*, 36–48.

Ribeiro, M. T., Singh, S., & Guestrin, C. (2016). "why should i trust you?": Explaining the predictions of any classifier. In *Proceedings of the 22nd acm sigkdd international conference on knowledge discovery and data mining* (p. 1135—1144).

Shmaryahu, D., Shani, G., & Shapira, B. (2020). Post-hoc explanations for complex model recommendations using simple methods. In *Proceedings of the 7th joint workshop* *on interfaces and human decision making for recommender systems* (pp. 26–36).

Suphavilai, C., Bertrand, D., & Nagarajan, N. (2018). Predicting cancer drug response using a recommender system. *Bioinformatics*, *34*, 3907–3914.

Tintarev, N., & Masthoff, J. (2015). Explaining recommendations: Design and evaluation. In *Recommender systems handbook* (pp. 353–382). Springer.

Wachter, S., Mittelstadt, B., & Russell, C. (2017). Counterfactual explanations without opening the black box: Automated decisions and the gdpr. *Harvard journal of law & technology*, *31*, 841–887.

Zhong, J., & Negre, E. (2022). Shap-enhanced counterfactual explanations for recommendations. In *Proceedings of the 37th acm/sigapp symposium on applied computing* (p. 1365—1372). Association for Computing Machinery.

Zibriczky, D. (2016). Recommender systems meet finance: a literature review. In *Finrec*.