

Delivery Report

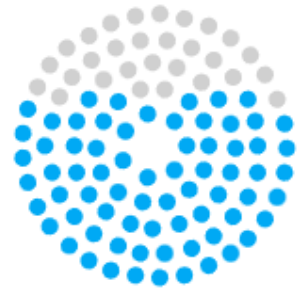
Customer Recommendation System

Domain Impact

The target stakeholder for this project are online web shops, who want to improve user experience in combination with possible increase in revenue. For this challenge done during this project the stakeholder was the e-commerce company who supplied the data for this challenge, sadly I could not get into direct contact with them.

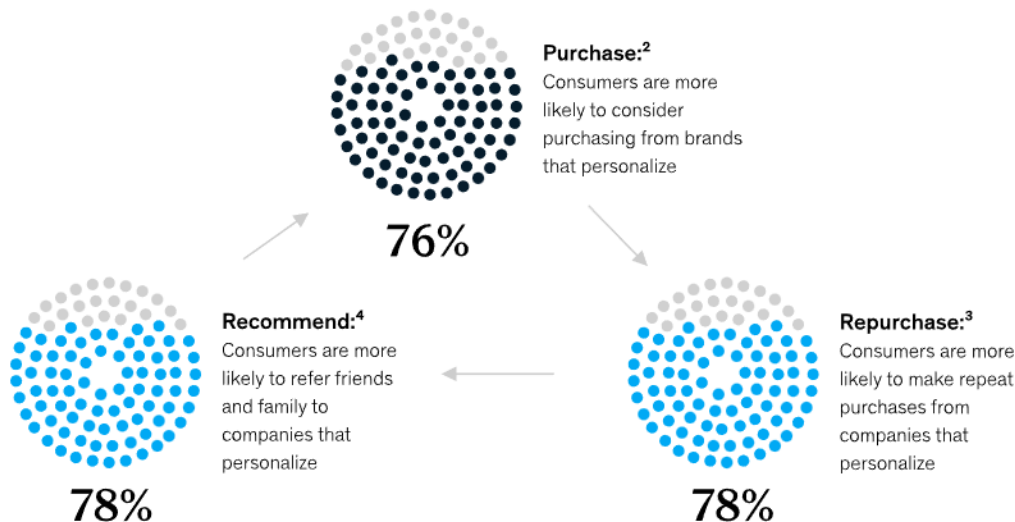
From a research post doing surveys on online consumers (taken by ~2000 people) during and after the COVID-19 pandemic, it was found that consumers in the modern decade are not just liking recommendation systems and personalization on their websites but are expecting it.

Because of the importance the online consumer market puts on personalization, and the shown viability of recommendation systems by websites such as YouTube and Amazon, it becomes clear that forms of recommendation systems are here to stay. For this reason, it will be worth while for online retailers to invest their time into learning these technologies and implementing them in their own services.



71%

of consumers expect personalization²



Introduction

The project's goal is to show the viability and gain a greater understanding into recommendation systems, to help online web shops create the most efficient and user-friendly service possible.

This project is focused on creating an unsupervised AI model that can recommend products to users, based on their purchasing habits and that of similar users, this will increase the user experience of finding products the user might be looking for, as well as helping the web store create more sales by showing users products, they might be interested in.

Findings

There were many different discoveries made during this project, such as the analytic data found for the current day appeal of recommendation systems as shown in the [Domain Impact](#).

Beyond this it was discovered that the sales data that was used for this project, had some work on it that had to be done before it could be used in training any models.

The problems discovered in the data were:

- The overlap of data periods between data sheets causing duplicated records.
- Products being bought multiple times in a single invoice instead of increasing the product quantity.
- Lastly, Manual records being added with none sold products in the product field such as postage, "manual", etc.

These issues had to be solved before the data could be used in any way, the duplicated records would cause the model to be skewed towards the duplicated data and the nonproducts in the records would cause these nonproducts to be recommended as well (such as postage).

Furthermore, because of the lack of easy way to evaluate unsupervised recommendation systems, special methods are used to identify the performance of the system.

The most common method of evaluating is by using a review score for products and having the model predict what score the user will give the product, this also helps create an order of how aggressively to order certain products.

Sadly, the data used during this project did not contain any kind of review score metric, instead the quantity being sold was scaled to a 0-10 scale and used instead.

Lastly, besides evaluating using some kind of evaluation metric, it was found that using real world examples such as deploying the recommendation system on a small scale and getting user feedback, is a much more useful metric for performance of the model.

Showcase

The model used in this project has been trained using the cleaned data originating from an e-commerce company, the model has been cross validated to avoid possible data split biases and has been hyper parameter tuned using grid searching.

The model trained to be used as the deliverable is a KNN model which implementation has been supplied by [scikit surprise](#).

Using the evaluation method described during the [Findings](#), the model ended up with an MSE of <0.5 out of the quantity 0-10 range, this means that the model makes an average error of 0.5 from the quantity range supplied.

This result is very good and can be used as a fairly accurate prediction model to suggest what products the user will be most likely to be interested in / buy.

After training the model, any user can be supplied into the model to return an ordered list of products to recommend to the target user.

For example, here the top 10 recommended products for a target user has been returned. The first product in this list and thus the first product to recommend to this user is a 72 pack of retro spot cake cases.

```
['21212', '22086', '85099B', '84879', '21080', '22555',  
'21977', '82494L', '84991', '21121']
```

StockCode	Description
21212	PACK OF 72 RETRO SPOT CAKE CASES



Recommendation

The model trained ended up being a very good model to use for a web shop's recommendation system and is very easy to use and implement into such a system.

However, for a future iteration I would suggest keeping a couple things in mind, firstly when tuning the model for the best results it was found that min_k (minimum amount of neighbors to use) had an optimal value of 5, while the optimal value for k (amount of neighbors to consider) was the highest possible value, however increasing the k value does also come with risks such as overfitting the model, for this reason I suggest keeping the k to a lower value such as 20-40.

Furthermore, as explained during the [Findings](#), a scaled version of the quantity feature was used to order the products recommendation list and to be used for evaluating the model.

I think that in future iterations of the model, a different indicator can be used for this. The best indicator to use would be to introduce product review score as done by most unsupervised recommendation systems. However, if a review score can not be used, I would recommend using a form of potential profit instead such as quantity*price, seen as both these features already exist in the current data.

Lastly, I would recommend splitting the data into product purchases and other costs, this way postage costs, manual purchases and adjustments can be done without worsening the dataset for a recommendation system.