

Challenge Documentation

Customer Recommendation System

Thomas van der Molen
S4-AI41

Project Information	
Project members	Thomas van der Molen (4168003)
Project name	Challenge Documentation
Version	1.1

Table of Contents

Version History.....	2
Prerequisite.....	3
Challenge Introduction	3
Project Plan	4
Gathering Data.....	4
Observing / Cleaning.....	4
Modelling	4
Process	6
Data Features.....	6
Models	6
Goal	6
Deployment Recommendations	7

Version History

Version	Date	Change
1.0	22-06-2022	Created document
1.1	23-06-2022	Grammar fixes

Prerequisite

This documentation will go over the steps taken during the Challenge: “Customer Recommendation” this challenge will be referenced many times in this document and can be found [online](#) or if possible [locally](#).

Challenge Introduction

The goal of this challenge was to try and create an unsupervised product recommendation system, using data obtained from an e-commerce company. Being able to create a recommendation system for products, customers will be more likely to buy at your company and the user experience of your web shop will increase, allowing for more possible customers.

Because an unsupervised model, functions more like a blackbox with people not really knowing how or why the model gave the results it did, extra caution will have to be taken to evaluate the results properly.

Furthermore, because of the generally private company and personal data being used during the challenge, great care has to been taken into keeping in mind the privacy and intellectual property of the data being worked with.

The goal of this challenge will be to create a model that can recommend products to users based on their and other purchasing habits.

Recommendation systems have been used extensively in the real world, any large website such as YouTube, Netflix, Spotify or amazon will have a recommendation system to help users find things they are interested in and increase the user experience.

Because the viability of these systems has been proven on many occasions in the last decade, this challenge will be focused a lot on how and why these systems work.

Project Plan

To get to a functional recommender model a couple steps had to be taken, namely: gathering data, observing/cleaning the data and creating a model, these 3 steps will be discussed individually below.

Gathering Data

Obtaining sales data from an e-commerce business, is fairly difficult seen as this data can contain personal information, and is generally seen as high value intellectual property by companies.

While it is possible to obtain large amounts of sales records from businesses, most of these methods and data that can be found is legally questionable at best.

Thankfully Dr. Daqing Chen, who has worked with a wholesale e-commerce company from the UK to create multiple research papers on customer profitability, has donated the data used in his research on the [UCI archives](#).

Observing / Cleaning

This phase has been completely documented in the Challenge document referenced during the [Prerequisite](#).

The data used during this challenge contained ~1.000.000 sales records, split into two periods; 2009-2010 and 2010-2011. As you might have already noticed, the two data periods seem to overlap. This overlap in data turned out to indeed be the case, causing extra measures to have to be taken to not have duplicate records in the dataset.

The data contained information on:

- What invoice a product sale was part of
- The product being sold
- A product description
- Quantity being sold
- Date of transaction
- Price of a single unit of the product in GBP(£)
- Unique identifier for the user making the purchase
- Country of the user

Besides the overlap in data there seemed to be some other odd problems with the data that had to be cleaned.

It was found that ~230.000 records did not contain a valid identifier for the user, this is an important feature as we will need the user and the products bought to create a profile for this user to recommend products to.

Furthermore, it was found that some invoices contained the same item being purchased multiple times, instead of once with a higher quantity count.

Lastly, there seemed to be some records that were filled manually, these records did not follow any standard formatting and were generally missing information. In the same vein it was found that nonproducts such as postage costs were also present in these records, all these records would have to be filtered, cleaned and possibly removed.

Besides the inconsistencies found, the data was very good for this challenge, it contained the needed identifiers such as user ID and product ID, and the data itself was very evenly distributed, with a lot of products have plenty of sales records, multiple countries being sold to and customers on average having a large amount of datapoints to use for clustering.

Modelling

Creating unsupervised models works slightly differently than its supervised counterparts.

A recommendation system using collaborative filtering for example will try grouping users together dynamically based on buying habits the model discovers, a single user can be part of many different clusters, and the reasoning behind what cluster he is a part of is not always clear.

After the target user has been clustered with other users, the model will create a list of products similar users have bought and will recommend these based on some tiering variable (this can be user ratings, price, etc.).

Special measures must be used to evaluate the effectiveness of a model, seen as when a model recommends new items to buy, there is no direct way to see how well this actually helped the user. To help with evaluation an extra scoring measure is added, this is generally a review score for products. The model will try to predict what the user will give a product for review score and this can be compared to test data containing the score the user actually gave.

Sadly, the dataset used does not contain any product reviews, so instead the quantity purchased was used and scaled to a 0-10 range. If the model can predict how much a user will buy of a product, it is fair to say it knows to a fairly good degree how likely a user is to buy the product (especially since the general users in the dataset are small stores who are buying products as stock to be sold).

These evaluation methods were done in combination with more general evaluation techniques such as K-fold crossvalidation to get rid of possible data split biases and grid searching to find the most optimal hyper parameters.

During the challenge two models were used (supplied by [scikit surprise](#)). The first model used was NormalPredictor, this model was used as a baseline and uses relatively basic methods to create it's recommendations. The baseline model already performed fairly well, getting an error range of ~1 for values ranging between 0-10.

The second model used is called KNNBasic, as the name suggests it is a nearest neighbor like model, using a user based algorithm, this means it uses collaborative filtering as discussed earlier to recommend products. This second model performed within a ~0.5 error range, which is half of the baseline and I would say is very accurate.

The KNNBasic also went through hyper parameter tuning using a grid search for two parameters, these being k (amount of neighbors to consider) and min_k (minimum amount of neighbors to use), out of this tuning it was found that for min_k the optimal value is 5, and for k it was the highest possible, which makes sense as more neighbors will give more datapoints to reference, however due to the possibility of overfitting I would suggest keeping k to a lower value such as 20-40.

Process

During this process certain aspects were specifically prioritized or considered, some of these have been explicitly explained below.

Data Features

An unsupervised recommendation system, does not require many features to work.

However, the algorithms and evaluation methods are greatly improved by making use of extra data, for example because of the lack of evaluation features such as review score, during this challenge quantity had to be used (in the future I think using a form of profit such as quantity*price would be a good alternative).

Furthermore, other algorithms such as product based ones make use of recommending products similar to products the user already purchased, for this however there has to be extra data on products such as the category it belongs to be a viable usecase.

Models

As mentioned during [Modelling](#), extra measures have to be taken to try and evaluate the performance of a model, these methods however are nowhere near as good as testing the model in a real world scenario.

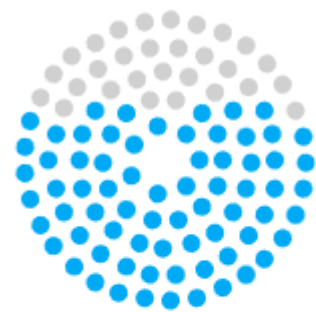
I think that any future attempts or further exploration of the challenge will have to be done in cooperation with a real world company that can apply the model on a small scale and get genuine user feedback on the products they are being recommended.

Goal

The goal of this challenge was to create a good understanding and implementation of a recommendation system. Recommendation systems have been used extensively and have shown their importance in the current day with 70% of the current consumers even expecting some sort of personalization.

I think this goal was achieved fairly well in the challenge, very extensive research has been done in the impact recommendation systems have had and the actual internal workings of them.

This has been proven by creating a recommendation model with an average error range of less than 0.5 out of 0-10 possible range.



71%

of consumers expect
personalization²

Deployment Recommendations

In the challenge a showcase has been given on how the recommendation system actually provides its recommendations. The system used is very straightforward, after a model has been trained any user can get recommendations which will be a list of the product ID's ordered based on what the model expects the user will buy the most of.

For deployment I would suggest looking into the scoring used to create the order of the list and evaluation metric, currently the quantity sold is being used, but possible a profit metric such as quantity*price or introducing review ratings would create a better experience for the user and possibly better profits for this company.

An extra recommendation for training the model in the future with new data would be to take care to look at the products in the dataset. With a business making many sales every day it is inevitable that manual changes will have to be made, such as creating a manual payment charge for a product not registered in the system, or postage charges being stored together with the products. These nonproducts should never be recommended by customer, and for this reason should be looked at extra carefully to not create such a situation.