Capstone Project: H&M Personalised Recommendations

Helen Gaskell, 10/04/2023

Problem Statement and Background

In order to maintain customer engagement and brand loyalty, many top retailers are using machine learning to recommend products, personalised to their customers' needs. With so much choice at your fingertips as a consumer, it's becoming increasingly important for companies to stand out and work on ways to encourage customers to buy from them.

H&M Group is a family of brands and businesses with 53 online markets and approximately 4,850 stores. The online store offers shoppers an extensive selection of products to browse through. H&M was predominantly an in-store based retailer, but with many e-commerce websites such as asos and prettylittlething dominating the online market-place, it's important for H&M to stay relevant and offer similar online experiences to their competitors, in order to keep their customers.

In this project, I will try various machine learning models to predict the next 8 items an H&M customer would likely buy. I will use methods such as baseline popularity models, Decision Tree and collaborative filtering/market basket analysis.

About the Data

Despite H&M selling products

The data consists of 3 tables:

- **Customers** - This table provides information such as age, membership status, if they're signed up to H&M newsletter, customer id
- **Transactions** - This table provides information such as transaction id, article id, price of article, date of transaction, number of purchases per article, online or in-store.
- **Articles** - This table provides a description of each article available for customers to buy at H&M

They are very large datasets, the transactions table consists of 31 million rows, there are 105,000 unique products in the articles dataset and 3 million unique customers.

Datasets of this size can be difficult to work with, so along the way I've had to manipulate it in certain ways to make it easier to handle. E.g. dropping certain columns in the Decision Tree process and only selecting customers who made over a certain number of transactions for collaborative filtering.

Data Cleaning and EDA

Transactions: On exploring transactions, I discovered that the transactions range from the dates September 2018 to September 2020. There were 3 million duplicates, which is usually something of concern. However, when exploring this further, I have deduced that they must be from customers buying more than 1 size of an article. When ordering something online, it's very common for someone to order more than 1 size of an item and return the item that didn't fit as well. For this reason I kept the duplicated rows. There were no null values and 104,000 unique articles bought in the 2 year period.

I discovered via a boxplot that there were a large range of outliers in the price of articles, this is likely due to limited boutique type ranges that H&M tend to do as collaborations. Unsurprisingly, there were over double the amount of transactions online than in-store. The data was also recorded over the pandemic which backs this up.
On plotting a bar chart of monthly transactions. It was clear to see that sales were much higher in the summer months.

Customers: In this table there were a high percentage of null values within columns, particularly in FN (whether a customer gets a fashion newsletter) and Active (If a customer is active for communication or not). I dropped FN column and converted NaN values in Active to 0. There were also some null values in age which I filled with the mean age.

Customers between the ages 22-28 made the most transactions in the 2 year period.

Articles: On exploring the articles dataset, I discovered there were 0.4% of product detailed descriptions missing. I decided to remove these rows as they count for a very small percentage of the dataset.

Garment upper body, lower body and full body were the most expensive items. Trousers were the most popular items, in particular Jade HW Skinny Denim TRS which was the most bought product in the 2 year period. Ladieswear was the department with the most transactions.

Modeling

Baseline Model 1: I decided to start the modeling process by recommending the most popular item from a customer's most purchased category. I did this by selecting those customers who had purchased over 50 items. This way, we can actually see a trend in which category they bought from most. E.g. H&M's top customer bought over 100 dresses, therefore I recommended the most popular bought dress to them.

Decision Tree Model 2: Here I used a classification model to make predictions based on a set of rules or conditions. I built the model based on customer's historical behaviour (past purchases), demographic information and item attributes, such as price and popularity. Each node in the tree represented a decision based on a feature and the branches represented the possible outcomes or decisions that can be made on that feature. Items were recommended to the customer based on the path through the tree that corresponds to the customer's characteristics and preferences.

Collaborative filtering/Market Basket Analysis: This was the most complex but effective way of recommendation system modeling. I used turicreate which is a library which specialises in handling large datasets and recommender systems. Again, I used a portion of the dataset to make the process more efficient. I used 3 different models which are available in turicreate library; popularity, cosine similarity and pearson similarity.

Popularity model:
The popularity model recommends the same 8 items to each customer based on their popularity. This makes it the least personalised model.

Cosine similarity:
The cosine similarity model creates vectors between items A and B and works out the cosine angle between the two. The smaller the cosine angle, the bigger the cosine similarity between the two items. The angle is calculated using below:

$$\text{similarity} = \cos(\theta) = \frac{\mathbf{A} \cdot \mathbf{B}}{\|\mathbf{A}\| \|\mathbf{B}\|} = \frac{\sum_{i=1}^{n} A_i B_i}{\sqrt{\sum_{i=1}^{n} A_i^2} \sqrt{\sum_{i=1}^{n} B_i^2}},$$

Pearson similarity:
The Pearson similarity is the pearson coefficient between the two item vectors. It measures the correlation between the purchase behaviours of two users for a set of items. Pearson similarity ranges from -1 to 1, where 1 indicates that two users have identical ratings for all items, -1 indicates that the users have opposite ratings, and 0 indicates that there is no correlation between the ratings.

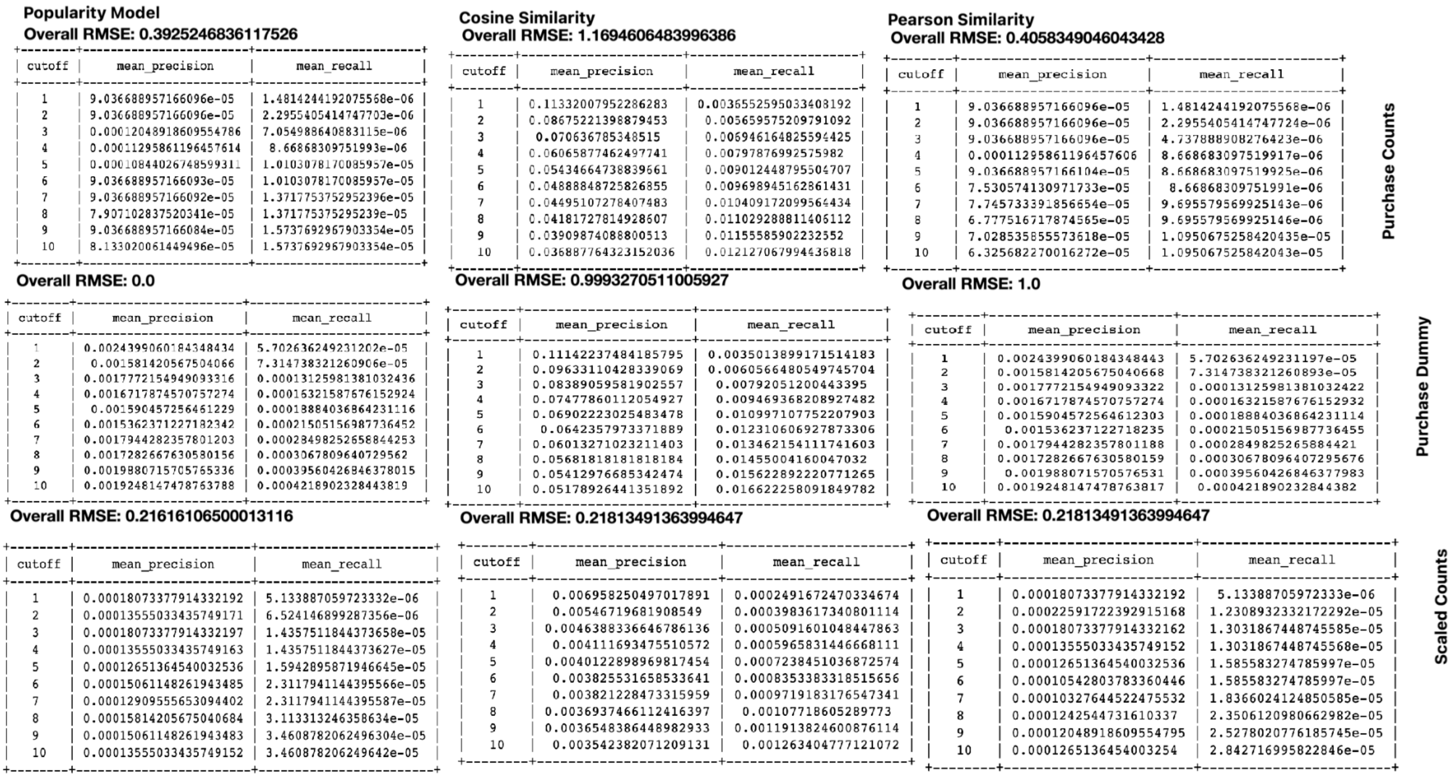
$$r = \frac{\sum_{i=1}^{n} (x_i - \bar{x})(y_i - \bar{y})}{\sqrt{\sum_{i=1}^{n} (x_i - \bar{x})^2 \sum_{i=1}^{n} (y_i - \bar{y})^2}}$$

Results and insights

Baseline Model 1: This model only recommends 1 item for each customer, based on which category they buy from the most. It isn't particularly accurate as doesn't take into account any similarity between other customers.

Decision Tree Model: The accuracy for this model on the train data was 0.6 but 0.0 on the test data meaning it's completely inaccurate. It makes predictions based on the customer demographic, e.g. age, how often they receive newsletters, whether they are a member. This shows us that customer demographic information is not correlated enough with the purchases that they buy.

Collaborative Filtering:



Popularity Model
Overall RMSE: 0.3925246836117526

```
+--------+--------------------------+--------------------------+
| cutoff |      mean_precision      |       mean_recall        |
+--------+--------------------------+--------------------------+
|   1    |  9.036688957166096e-05   |  1.4814244192075568e-06  |
|   2    |  9.036688957166096e-05   |  2.2955405414747703e-06  |
|   3    |  0.00012048918609554786  |  7.054988640883115e-06   |
|   4    |  0.00011295861196457614  |   8.66868309751993e-06   |
|   5    |  0.00010844026748599311  |  1.0103078170085957e-05  |
|   6    |  9.036688957166093e-05   |  1.0103078170085957e-05  |
|   7    |  9.036688957166092e-05   |  1.3717753752952396e-05  |
|   8    |  7.907102837520341e-05   |  1.371775375295239e-05   |
|   9    |  9.036688957166084e-05   |  1.5737692967903354e-05  |
|   10   |  8.133020061449496e-05   |  1.5737692967903354e-05  |
+--------+--------------------------+--------------------------+
```

Cosine Similarity
Overall RMSE: 1.1694606483996386

```
+--------+--------------------------+--------------------------+
| cutoff |      mean_precision      |       mean_recall        |
+--------+--------------------------+--------------------------+
|   1    |   0.11332007952286283    |  0.0036552595033408192   |
|   2    |   0.08675221398879453    |   0.005659575209791092   |
|   3    |    0.070636785348515     |   0.006946164825594425   |
|   4    |   0.06065877462497741    |   0.00797876992575982    |
|   5    |   0.05434664738839661    |   0.009012448795504707   |
|   6    |   0.04888848725826855    |   0.009698945162861431   |
|   7    |   0.04495107278407483    |   0.010409172099564434   |
|   8    |   0.04181727814928607    |   0.011029288811406112   |
|   9    |   0.03909874088800513    |   0.01155585902232552    |
|   10   |   0.036887764323152036   |   0.012127067994436818   |
+--------+--------------------------+--------------------------+
```

Pearson Similarity
Overall RMSE: 0.4058349046043428

```
+--------+--------------------------+--------------------------+
| cutoff |      mean_precision      |       mean_recall        |
+--------+--------------------------+--------------------------+
|   1    |  9.036688957166096e-05   |  1.4814244192075568e-06  |
|   2    |  9.036688957166096e-05   |  2.2955405414747724e-06  |
|   3    |  9.036688957166096e-05   |  4.737888908276423e-06   |
|   4    |  0.00011295861196457606  |  8.668683097519917e-06   |
|   5    |  9.036688957166104e-05   |  8.668683097519925e-06   |
|   6    |  7.530574130971733e-05   |   8.66868309751991e-06   |
|   7    |  7.745733391856654e-05   |  9.695579569925143e-06   |
|   8    |  6.777516717874565e-05   |  9.695579569925146e-06   |
|   9    |  7.028535855573618e-05   |  1.0950675258420435e-05  |
|   10   |  6.325682270016272e-05   |  1.095067525842043e-05   |
+--------+--------------------------+--------------------------+
```

Purchase Counts

Overall RMSE: 0.0

```
+--------+--------------------------+--------------------------+
| cutoff |      mean_precision      |       mean_recall        |
+--------+--------------------------+--------------------------+
|   1    |  0.0024399060184348434   |  5.702636249231202e-05   |
|   2    |   0.001581420567504066   |  7.314738321260906e-05   |
|   3    |  0.0017772154949093316   |  0.00013125981381032436  |
|   4    |  0.0016717874570757274   |  0.00016321587676152924  |
|   5    |   0.001590457256461229   |  0.00018884036864231116  |
|   6    |  0.0015362371227182342   |  0.00021505156987736452  |
|   7    |  0.0017944282357801203   |  0.00028498252658844253  |
|   8    |  0.0017282667630580156   |  0.0003067809640729562   |
|   9    |  0.0019880715705765336   |  0.00039560426846378015  |
|   10   |  0.0019248147478763788   |  0.0004218902328443819   |
+--------+--------------------------+--------------------------+
```

Overall RMSE: 0.9993270511005927

```
+--------+--------------------------+--------------------------+
| cutoff |      mean_precision      |       mean_recall        |
+--------+--------------------------+--------------------------+
|   1    |   0.11142237484185795    |  0.0035013899171514183   |
|   2    |   0.09633110428339069    |  0.0060566480549745704   |
|   3    |   0.08389059581902557    |   0.00792051200443395    |
|   4    |   0.07477860112054927    |   0.009469368208927482   |
|   5    |   0.06902223025483478    |   0.010997107752207903   |
|   6    |    0.0642357973371889    |   0.012310606927873306   |
|   7    |   0.06013271023211403    |   0.013462154111741603   |
|   8    |   0.05681818181818184    |   0.01455004160047032    |
|   9    |   0.05412976685342474    |   0.015622892220771265   |
|   10   |   0.05178926441351892    |   0.016622258091849782   |
+--------+--------------------------+--------------------------+
```

Overall RMSE: 1.0

```
+--------+--------------------------+--------------------------+
| cutoff |      mean_precision      |       mean_recall        |
+--------+--------------------------+--------------------------+
|   1    |  0.0024399060184348443   |  5.702636249231197e-05   |
|   2    |  0.0015814205675040668   |  7.314738321260893e-05   |
|   3    |  0.0017772154949093322   |  0.00013125981381032422  |
|   4    |  0.0016717874570757274   |  0.00016321587676152932  |
|   5    |  0.0015904572564612303   |  0.00018884036864231114  |
|   6    |   0.001536237122718235   |  0.00021505156987736455  |
|   7    |  0.0017944282357801188   |  0.0002849825265884421   |
|   8    |  0.0017282667630580159   |  0.00030678096407295676  |
|   9    |   0.001988071570576531   |  0.00039560426846377983  |
|   10   |  0.0019248147478763817   |   0.000421890232844382   |
+--------+--------------------------+--------------------------+
```

Purchase Dummy

Overall RMSE: 0.21616106500013116

```
+--------+--------------------------+--------------------------+
| cutoff |      mean_precision      |       mean_recall        |
+--------+--------------------------+--------------------------+
|   1    |  0.00018073377914332192  |  5.133887059723332e-06   |
|   2    |  0.00013555033435749171  |  6.524146899287356e-06   |
|   3    |  0.00018073377914332197  |  1.4357511844373658e-05  |
|   4    |  0.00013555033435749163  |  1.4357511844373627e-05  |
|   5    |  0.00012651364540032536  |  1.5942895871946645e-05  |
|   6    |  0.00015061148261943485  |  2.3117941144395566e-05  |
|   7    |  0.00012909555653094402  |  2.3117941144395587e-05  |
|   8    |  0.00015814205675040684  |  3.113313246358634e-05   |
|   9    |  0.00015061148261943483  |  3.4608782062496304e-05  |
|   10   |  0.00013555033435749152  |  3.460878206249642e-05   |
+--------+--------------------------+--------------------------+
```

Overall RMSE: 0.21813491363994647

```
+--------+--------------------------+--------------------------+
| cutoff |      mean_precision      |       mean_recall        |
+--------+--------------------------+--------------------------+
|   1    |   0.006958250497017891   |  0.0002491672470334674   |
|   2    |   0.00546719681908549    |  0.0003983617340801114   |
|   3    |  0.0046388336646786136   |  0.0005091601048447863   |
|   4    |   0.004111693475510572   |  0.0005965831446668111   |
|   5    |  0.0040122898969817454   |  0.0007238451036872574   |
|   6    |   0.003825531658533641   |  0.0008353383318515656   |
|   7    |   0.003821228473315959   |  0.0009719183176547341   |
|   8    |  0.0036937466112416397   |   0.00107718605289773    |
|   9    |  0.0036548386448982933   |  0.0011913824600876114   |
|   10   |   0.003542382071209131   |   0.001263404777121072   |
+--------+--------------------------+--------------------------+
```

Overall RMSE: 0.21813491363994647

```
+--------+--------------------------+--------------------------+
| cutoff |      mean_precision      |       mean_recall        |
+--------+--------------------------+--------------------------+
|   1    |  0.00018073377914332192  |   5.13388705972333e-06   |
|   2    |  0.00022591722392915168  |  1.2308932332172292e-05  |
|   3    |  0.00018073377914332162  |  1.3031867448745585e-05  |
|   4    |  0.00013555033435749152  |  1.3031867448745568e-05  |
|   5    |  0.00012651364540032536  |  1.585583274785997e-05   |
|   6    |  0.00010542803783360446  |  1.585583274785997e-05   |
|   7    |  0.00010327644522475532  |  1.8366024124850585e-05  |
|   8    |  0.0001242544731610337   |  2.3506120980662982e-05  |
|   9    |  0.00012048918609554795  |  2.5278020776185745e-05  |
|   10   |  0.0001265136454003254   |  2.842716995822846e-05   |
+--------+--------------------------+--------------------------+
```

Scaled Counts

The above shows the results for each model on each variation of the dataset. When looking at the RMSE, you'd think that the popularity model on the purchase dummy set was the most accurate as the RMSE is 0.0. RMSE measures the error of the predicted values, therefore usually the lower this is the more accurate the model. However you also need to take into account the recall and precision of the model. The recall shows us which percentage of products that a user buys were actually recommended. E.g. if a customer buys 5 products and the recommendation shows 3 of them then the recall would be 0.6. Precision shows us out of all the recommended items, how many the user actually liked. If 5 products were recommended to the customer out of which they buy 4 of them then the precision is 0.8. Both of these are important as if a customer was recommended every single product then recall would be 1 but that doesn't necessarily mean the model is accurate, it just means it covers all bounds. Precision is the most important as if the percentage is very small it would mean that a customer only buys a small percentage of the products recommended to them. This means we have to consider precision and recall, and aim for them to be as close to 1 as possible.

Looking at the above results, it's clear to see that the cosine similarity precision and recall scores are by far the closest to 1. Purchase dummy and purchase counts are closest to 1 out of the three. Then when looking at the RMSE score, Purchase dummy is the lowest therefore this is the model that I have chosen to use.