Capstone Project - 4

Unsupervised machine learning-
Product recommendation engine

Mohammed arifuddin atif

Sajal sinha

All about this presentation:

1. Defining problem statement.
2. Overview of data.
3. Performing exploratory data analysis.
4. Model preparation.
5. Model 1 - Popularity based recommender model
6. Model 2 - Collaborative Filtering using surprise library
7. Evaluation of all models.
8. Recommending 5 Products.

Problem statement

We are tasked with building a recommender engine that reviews customer ratings and purchase history to recommend items and improve sales.This engine looks at the customers previous purchases and ratings and provides recommendations of products similar to the products they like.

What is a recommender engine?

- A recommender engine, or a recommendation system, is a subclass of information filtering system that seeks to predict the “rating” or “preference” a user would give to an item. They are primarily used in commercial applications.

Overview of given data

We are given the following columns in our data:

Unique userId - Unique id used for customer Identification.

Product ASIN - Amazon's unique product identification code for each product.

Ratings - Ratings(1-5) given by customers based on their satisfaction.

Timestamp - Timestamp of the given rating in UNIX time.

Overview of given data

The Ratings column provides valuable information which tells the satisfaction of the customer from the product or the extent of their likeness towards a particular product.

The Unique userId column provides information on the identification of the customer.This id is unique to a customer.

Description of data

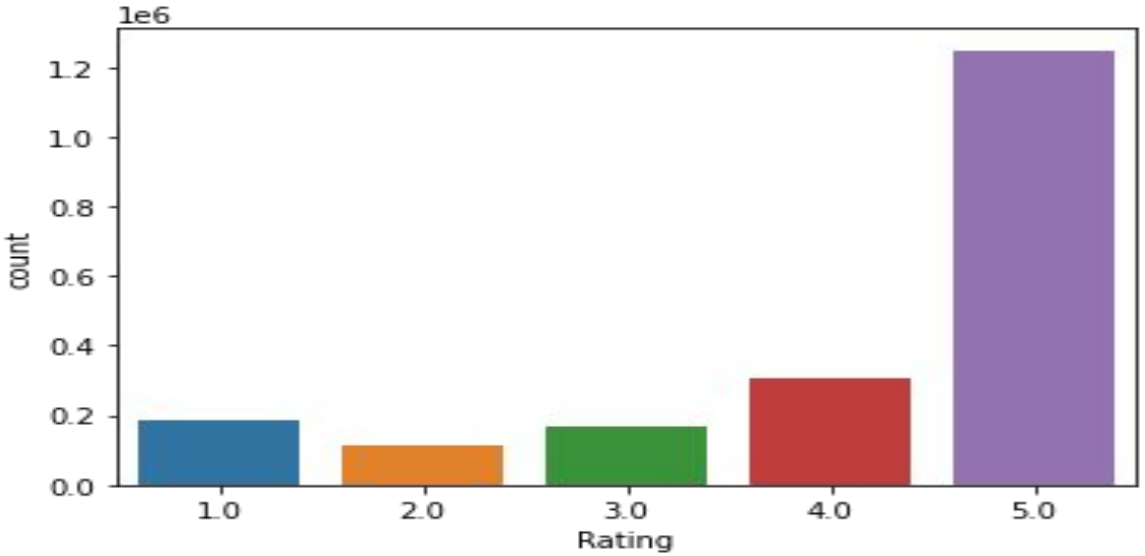
	UserId	ProductId	Rating	Timestamp
count	2023070	2023070	2.023070e+06	2.023070e+06
unique	1210271	249274	NaN	NaN
top	A3KEZLJ59C1JVH	B001MA0QY2	NaN	NaN
freq	389	7533	NaN	NaN
mean	NaN	NaN	4.149036e+00	1.360389e+09
std	NaN	NaN	1.311505e+00	4.611860e+07
min	NaN	NaN	1.000000e+00	9.087552e+08
25%	NaN	NaN	4.000000e+00	1.350259e+09
50%	NaN	NaN	5.000000e+00	1.372810e+09
75%	NaN	NaN	5.000000e+00	1.391472e+09
max	NaN	NaN	5.000000e+00	1.406074e+09

sample of data

	UserId	ProductId	Rating	Timestamp
0	A39HTATAQ9V7YF	0205616461	5.0	1369699200
1	A3JM6GV9MNOF9X	0558925278	3.0	1355443200
2	A1Z513UWSAAO0F	0558925278	5.0	1404691200
3	A1WMRR494NWEWV	0733001998	4.0	1382572800
4	A3IAAVS479H7M7	0737104473	1.0	1274227200

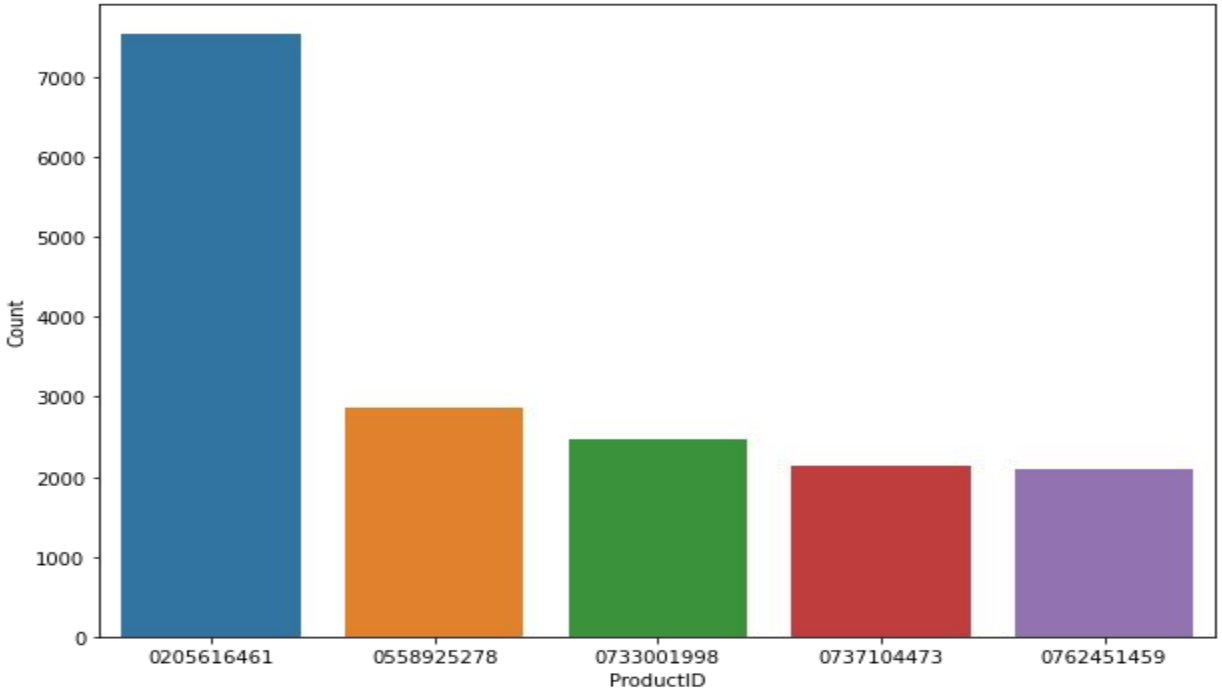
Exploratory Data Analysis

Count of ratings



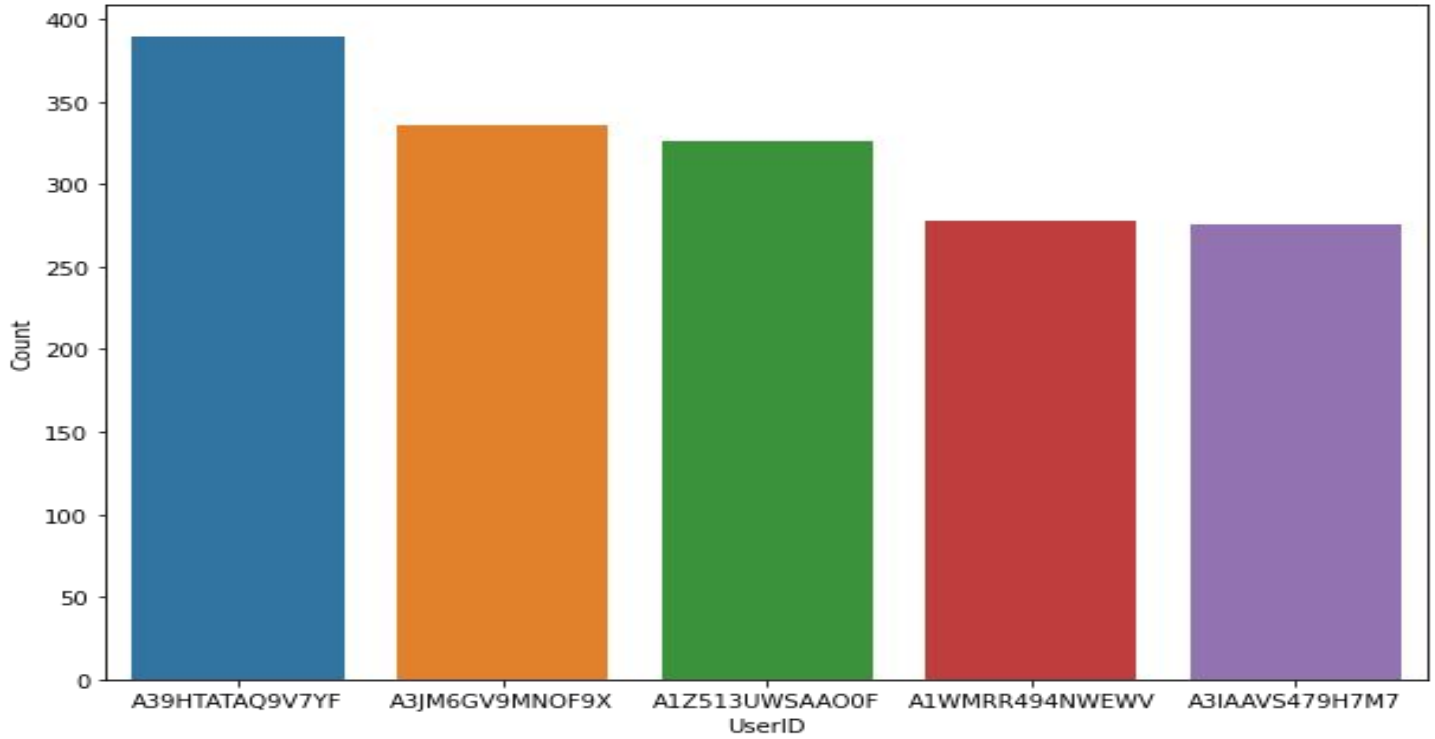
Above plot shows that most products are given the highest rating which is 5.

Most purchased product



It is clear from above that the product with product id **0205616461** has most purchases.

Top users who purchased products



Above plot shows that userid-A39HTATAQ9V7YF has made the most purchases.

Insights from the data

1. There are total of 1210271 unique users in the data.
2. The total number of unique products are 249374.
3. The products are rated from 1-5 by the users.
4. There are no null values in the dataset.
5. The number unique products which are rated high amount to 220746.

Model preparation

1. We had to drop Timestamp column as it does not provide much help in recommending a product to the customer.
2. After this we split our data into train and test sets.
3. As the data did not have any null values it was easy for us to directly move into preparing a recommender model.

Popularity based recommender model

1. It is a type of recommendation system which works on the principle of popularity and or anything which is in trend. These systems check about the product or movie which are in trend or are most popular among the users and directly recommend those.
2. For example, if a product is often purchased by most people then the system will get to know that that product is most popular so for every new user who just signed it, the system will recommend that product to that user also and chances become high that the new user will also purchase that.

Collaborative Filtering using surprise library

1. Collaborative does not need the features of the items to be given. Every user and item is described by a feature vector or embedding.It creates embedding for both users and items on its own. It embeds both users and items in the same embedding space.

2. It considers other users' reactions while recommending a particular user. It notes which items a particular user likes and also the items that the users with behavior and likings like him/her likes, to recommend items to that user.It collects user feedbacks on different items and uses them for recommendations.

Types of collaborative recommender systems

Memory-based collaborative filtering: Done mainly remembering the user-item interaction matrix, and how a user reacts to it, i.e, the rating that a user gives to an item. There is no dimensionality reduction or model fitting as such.

User-User filtering: In this kind, if a user A's characteristics are similar to some other user B then, the products that B liked are recommended to A. As a statement, we can say, "the users who like products similar to you also liked those products". So here we recommend using the similarities between two users.

Item-Item filtering: Here, if user A likes an item x, then, the items y and z which are similar to x in property, then y and z are recommended to the user. As a statement, it can be said, "Because you liked this, you may also like those".

Surprise library

Surprise is a Python library for building and analyzing rating prediction algorithms. It was designed to closely follow the scikit-learn API , which should be familiar to users acquainted with the Python machine learning ecosystem.

Surprise provides a collection of estimators (or prediction algorithms) for rating prediction. Among others, classical algorithms are implemented such as the main similarity-based algorithms , as well as algorithms based on matrix factorization like SVD or NMF. It also supports tools for model evaluation like cross-validation iterators and built-in metrics scikit-learn, as well as tools for model selection and automatic hyper-parameter search, namely grid search
and randomized search. Thanks to simple primitives and a light API, users can also implement their own recommendation technique with a minimal amount of code.

K nearest neighbours

AI

K-Nearest Neighbour is one of the simplest Machine Learning algorithms based on Supervised Learning technique.K-NN algorithm assumes the similarity between the new case/data and available cases and put the new case into the category that is most similar to the available categories.

K-NN is a **non-parametric algorithm**, which means it does not make any assumption on underlying data.It is also called a **lazy learner algorithm** because it does not learn from the training set immediately instead it stores the dataset and at the time of classification, it performs an action on the dataset.KNN algorithm at the training phase just stores the dataset and when it gets new data, then it classifies that data into a category that is much similar to the new data.

Singular value decomposition

The Singular Value Decomposition (SVD), a method from linear algebra that has been generally used as a dimensionality reduction technique in machine learning. SVD is a matrix factorisation technique, which reduces the number of features of a dataset by reducing the space dimension from N-dimension to K-dimension (where K<N). In the context of the recommender system, the SVD is used as a collaborative filtering technique. It uses a matrix structure where each row represents a user, and each column represents an item. The elements of this matrix are the ratings that are given to items by users.

Evaluation of models

1. The RMSE value for Popularity Recommender model resulted as 1.3078.
2. The RMSE value for collaborative filtering Recommender model resulted as 1.0561 (with KNN) and 1.01 (with SVD).
3. After hyper-parameter tuning we get rmse as 0.8750 (with SVD).

Snippet of recommendations

```
A25C2M3QF9G7OQ ['B009YSSLAU', 'B00HSNWZKU', 'B008VSYO5A', 'B00HSNWXVQ', 'B00AO4EMKQ']
AY3D7DG5L5WCK ['B00A0J09C0', 'B00GTBZHUW', 'B00AWLB9G6', 'B00AE07EGO', 'B00GTBZHCA']
A2DPYMNI2HCIOI ['B008RVYJS8', 'B00014D5O8', 'B008U2Y9BQ', 'B0002VQ0WO', 'B0014AX89K']
A2Y4S4CNGKF21S ['B00021B8L2', 'B003JTA1IU', 'B00069FJUG', 'B00392FQSM', 'B0007CXX82']
AYB4ELCS5AM8P ['B004GIM68G', 'B00A0J084Y', 'B008VSYO5A', 'B001ECQ4JE', 'B001ECQ55M']
A3GPAR4H5Y5OU8 ['B001ECQ4JE', 'B0030O3VRW', 'B000GUN98O', 'B003WN1ELQ', 'B001JQLNNC']
A3IOCPLIMYDBCD ['B000F37UNM', 'B002QXINAS', 'B004YZMKKU', 'B0043TUSME', 'B001LF4I8I']
ALNFHVS3SC4FV ['B004Z209HS', 'B004B9C8MO', 'B00GTC02LA', 'B001ECQ4YO', 'B009YSSLAU']
AQY5XBYSENNZQ ['B001U0ONNY', 'B00GTBZHUW', 'B006L6A06Q', 'B002MO780W', 'B00AE0790U']
A3091RP0SPZLMN ['B00AE07FUE', 'B00GTC02LA', 'B00AE07CTI', 'B001LF4G44', 'B00ALV8EJM']
A3AZI828WJN1CD ['B007RTR8UC', 'B00A0IWRRG', 'B006L6A06Q', 'B00AO4EBOI', 'B007WFZ0XI']
A2XKJ1KX6XUHYP ['B004CDQ73K', 'B000I5KCIW', 'B002CML1XE', 'B0037AO682', 'B006OC431K']
A3FBM0RMCMAABV ['B00AWLB9G6', 'B00AWLBACE', 'B007JT7AGC', 'B00AE0790U', 'B001JQLNJ6']
A3IQXED40ZZ2T2 ['B00GCQMSVA', 'B00IDWP4IA', 'B00IP42FBA', 'B00KD73PBQ', 'B004B9C8MO']
A16YU3GRGCD95S ['B00ALV8EL0', 'B00AE07FUE', 'B00AO4EMKQ', 'B00GTBZWPW', 'B00ALV8EJM']
A33AQPJYH7UUXR ['B0006ZHCK0', 'B004Y5NA1I', 'B0018V1DWE', 'B001M7SIFY', 'B001DKRMB6']
ACN76NFHYL6JK ['B0016B5LAQ', 'B005Z446JC', 'B002DPUY10', 'B000Q4B108']
```

The above product ids are 5 recommendations for a particular user.

Challenges faced

1. The given data contains approximately 20 lakh data points which made it difficult to work on the data.So we had to filter the data accordingly.
2. Tuning the model appropriately was one of the challenges we faced as the metrics of our model before tuning were not acceptable.

Conclusion

We are finally at the conclusion of our project!

Coming from the beginning we did EDA on the dataset and also cleaned the data according to our needs.After that we were able to draw relevant conclusions from the given data and then we trained our model on recommender systems.

For our model we were able to get an RMSE value of 0.875 after hyper parameter tuning.

Given the size of data and the amount of irrelevance in the data , the above score is good.