Amazon Home & Kitchen Recommendation System

Capstone 3 Project

Lanxi Liu

1. Introduction

During the pandemic time starting in 2020, people spend a tremendous amount of time at home. It is really the time that people start to plan about home improvement. According to Max Aderson, chief economist at Porch.com, it has reached all-time high in terms of like measured history in the United States, this is the, the highest levels of, of home improvement spending we've ever seen. The real estate marking also booming as the interest rate fell dramatically in 2020 and reached a record low on 2021. Selling and buying a home also encourage the purchase under home and kitchen category.

Imagine people start browsing the Amazon website to get some inspiration for their projects. It will be good to have a well-developed recommendation engine that reads customers' minds and prompts more sales.

On the other hand, as a customer looking for stuff for home improvement, I would like to be recommended related items whether they are related to what I am browsing or just popular and trending items people are buying. It would save a lot of time brainstorming about what I want.

2. Data Source

The data used for the prediction model is the Amazon product data from lab lead by professor Julian McAuley at UCSD. Home and Kitchen category was picked for this particular capstone project. The source files contain a reviews file and a metadata file. The review data shows information like the reviewer's name, review context, review time and whether other users rated it as helpful review. The metadata includes descriptions, price, sales-rank, brand information, and co-purchasing information of products.

See the following for the links:

- http://jmcauley.ucsd.edu/data/amazon/links.html

3. Data Wrangling

There are two sets of data as mentioned above. The reviews dataset has 4,253,926 entries. The meta dataset has 436,988 entries. The review data has much more entries because there are multiple reviews for products.

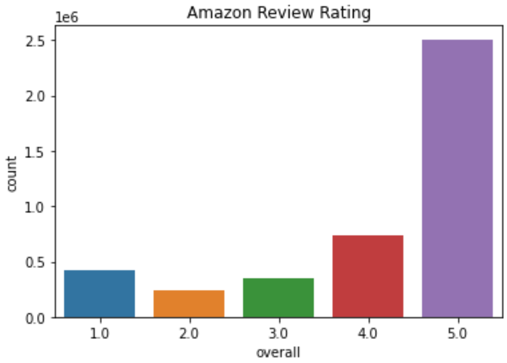
The source files are parsed from JSON format. Some of the columns still remain in JSON format such as the "related" column. There are many missing values in both datasets, however, there is no missing value for the reviewer ID number and the product unique number.

The idea for data wrangling is to use current features and aggregate new features for further analysis. The recommendation system does not need all that much information about the products and reviews. Essentially, three columns (reviewer, product, and rating) are required to feed the algorithms in the machine learning section. But, indeed, other features that have some influence on rating could improve the model. To achieve that, some redundant columns were dropped. The date related information was converted to datetime object. The reviews and meta datasets were merged to based on product ID.

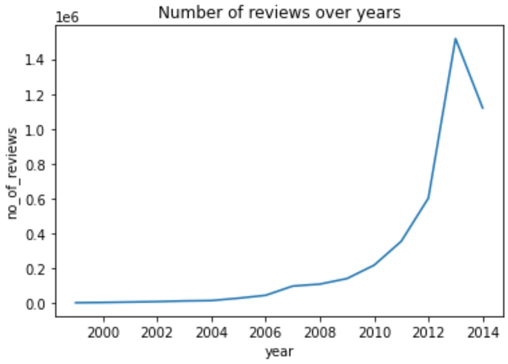
More detailed clean process could be found in the IPython notebook.

4. Exploratory Data Analysis (EDA)

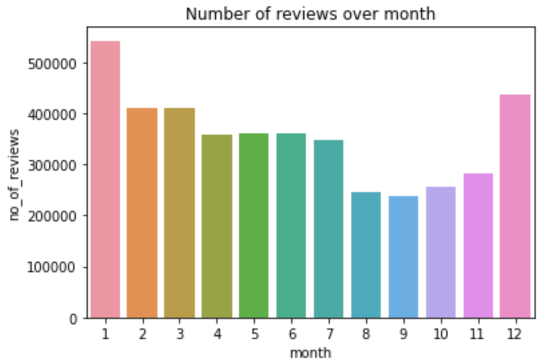
Looking at the rating data, which is the most crucial item in the recommendation system, we can see that it is highly skewed. The rating ranges from 1 to 5 and only contains integer values. Since the rating data is biased, it would be harder to predict lower values because the number of data is much less. For training the machine learning model, one way to do is to under select 5-star ratings. However, the dataset is huge, biased may not be cause too much trouble to the final model.



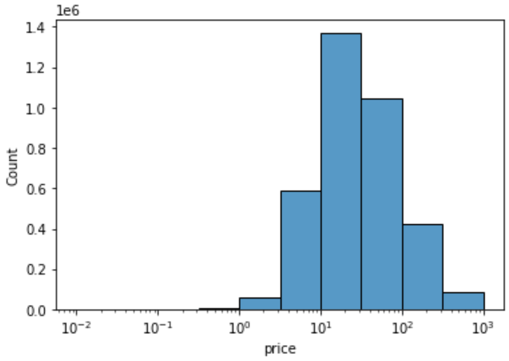
There are many interesting trends based on time.



The number of reviews increased exponentially from 1999 to 2013. Then it dropped between 2013 to 2014. The peak on 2013 might be related to a promotion which boost the number of reviewers. One thing we can say it there are more and more users giving out reviews and potentially more and more users using Amazon.

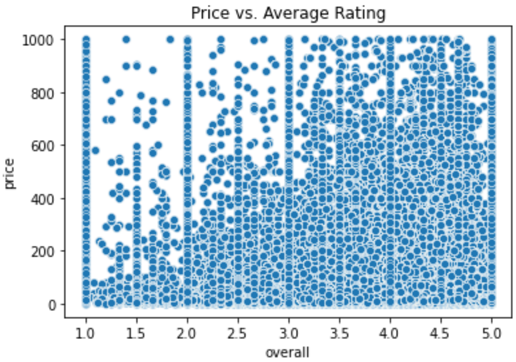


From the distribution of reviews over month, we can see that most of people left reviews on December and January. It might be related to the holiday season. The reviews are relatively low from August to November. People might just be waiting for the holiday season to come.

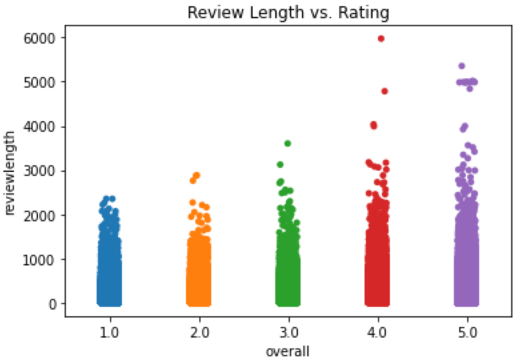


Most of the products were listed from 13 to 60 dollars. It is reasonable because products under Kitchen and Dining category usually are small to medium items that would not cost a whole lot of money. People are less interested on expansive items. This might because the demand is low compare to cheap items and people don't feel safe buying pricy items online.

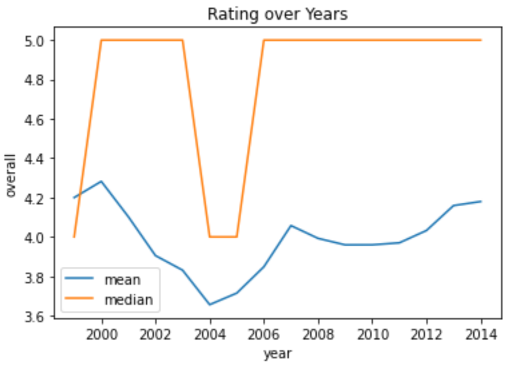
The average length of reviews is only 7 words, while the longest review has more than 6000 works. Most of the reviews are concise because most people didn't spend effort writing a review. People tended to leave a few sentences expressing whether they had a good or bad experience but would not go into details.



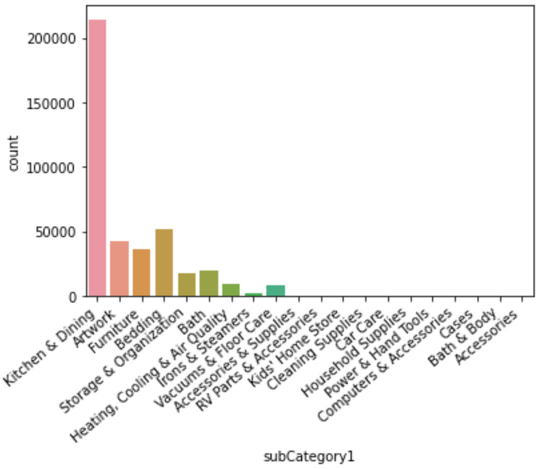
Looking at the scatter plot above, if we ignore the points on the exact numbers, we can see some relationships between price and rating. The items with lower price turn to get various rating. The higher price items are more likely to get high ratings. This might because higher price items tend to have better quality, which worth a higher score.



While most reviewers gave concise reviews, we can still see relationship between review length and ratings. For people who wrote very long reviews, they are mostly likely be very satisfied with the items and cannot wait to share their love to other customers. People who didn't get a good experience might already return the items back and didn't leave any reviews.



We already know that most people gave 5-star review for items they bought. As we can see in above, the median of the rating is mostly 5 over years. However, the median was 4 in 1999 and 2004. If we look at the average rating, we can get a different picture. The average rating dropped from 2000 and reached the lowest point at 2004. Then it went back up but still lower than the pick average in 2000.



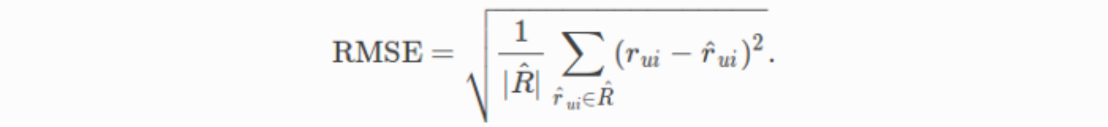
After subdividing the Home and Kitchen main category, we see that Kitchen & Dining is the most demanding subcategory. Here I decided to save this category as a slice of the main category for future analysis. There are 2,575,376 entries in this subcategory. Because the recommendation system works better if we have similarities between users and items. It would be less useful to include users who are new to the platform and gave less than 10 reviews. For new listed items that have less than 10 reviews are also less useful to the system. Therefore, in order to accelerate the modeling process, I decided to filter out less important users and items. The trimmed dataset finally includes 81,948 entries.

More details about the procedure of exploratory data analysis could be found in the IPython notebook.

5. Modeling

The trimmed Kitchen and Dining dataset was randomly divided to two parts. 80% goes to the training set and the rest 20% goes to the test set. For the ease of training, I implemented some feature engineering, which I manually create some numerical statistic values like average rating of user, average rating of item, number of users rate this item and number of items this user rates. So, for each entry in this dataset, the model can learn the relationship to others. There is also a column in the original dataset call "related". The source data summarized the items people also viewed, item people also bought or bought together and items users bought after viewing. Here in my analysis, I only include items also bought as a similarity metric. The average rating of related items bought was computed for every entry in the dataset. If there is not any information about related items bought, the average rating of the item will be used instead.

RMSE was used as the accuracy metric over mean absolute error because the errors are squared before being averaged. When there are large errors RMSE gives more weight to them and penalize them. In the recommendation system, we don't want to suggest an item that user don't like. Therefore, outliers are definitely not favorable.



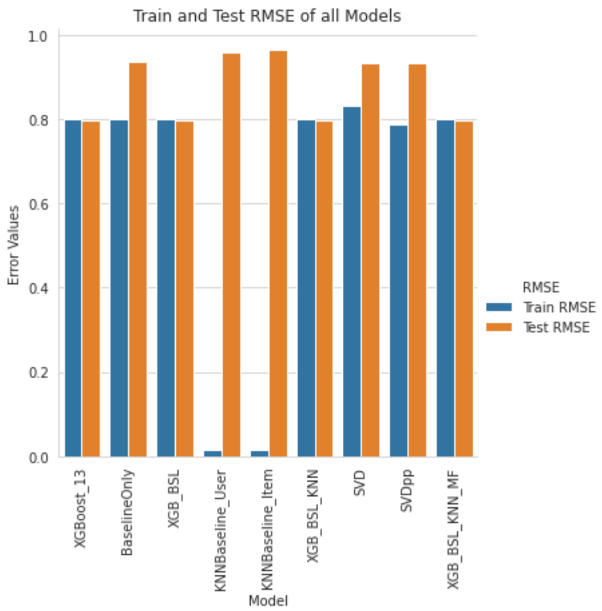
$$\text{RMSE} = \sqrt{\frac{1}{|\hat{R}|} \sum_{\hat{r}_{ui} \in \hat{R}} (r_{ui} - \hat{r}_{ui})^2}.$$

$$\mathrm{MAE} = \frac{1}{|\hat{R}|} \sum_{\hat{r}_{ui} \in \hat{R}} |r_{ui} - \hat{r}_{ui}|$$

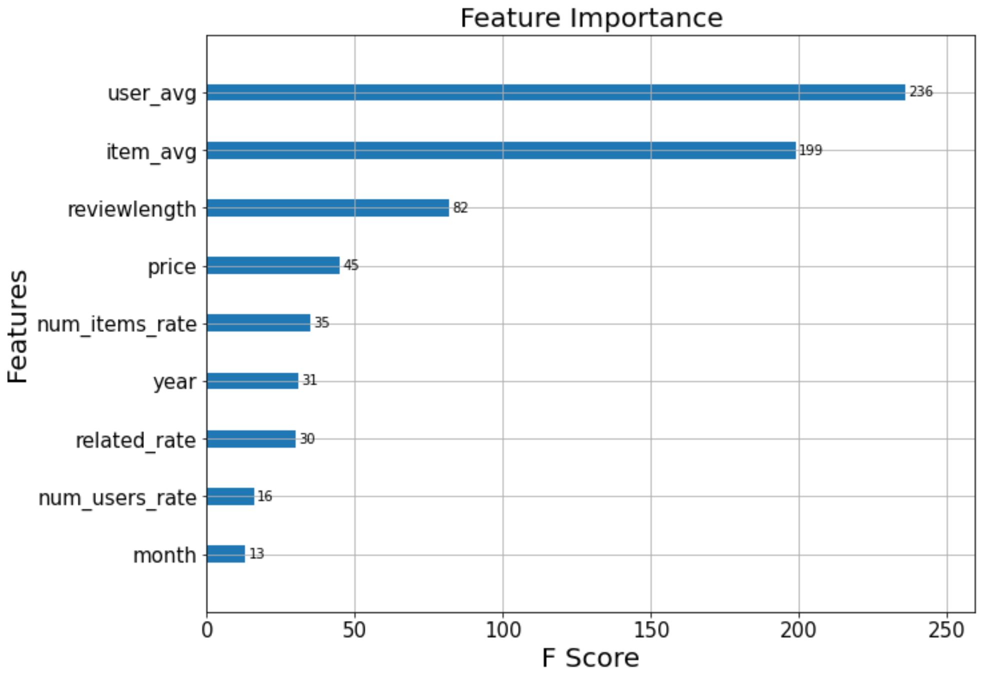
There are four base models that I created before using any implementation ready libraries. The first model predicts rating as 5 star no matter what, and receives an RMSE of 1.168. The second model was a simplest content-filtering based recommendation system, which predicts based on the mean rating of the user. This model gave us a RMSE of 1.016. The third model is the opposite, which is the collaborative-filtering based recommendation system. It predicts based on the mean rating of the item. The RMSE for the last model is 0.98. The last model is based on average rating of items people bought together. It is also a collaborative-filtering based model. The RMSE is 0.972, which is the smallest error we got so far.

We know that in reality, the user and product interaction is more complicate than just taking the average. Therefore, I chose to work with the Python surprise library. I tested several algorithms provided including BaselineOnly, KNN-Baseline, SVD, SVD++. In addition, I also used the XGBoost (Extreme Gradient Boosting) to train and predict the ratings. XGBoost was used because it has faster learning time and higher model performance. It also reduces the chance of overfitting which is a big concern of gradient boosting.

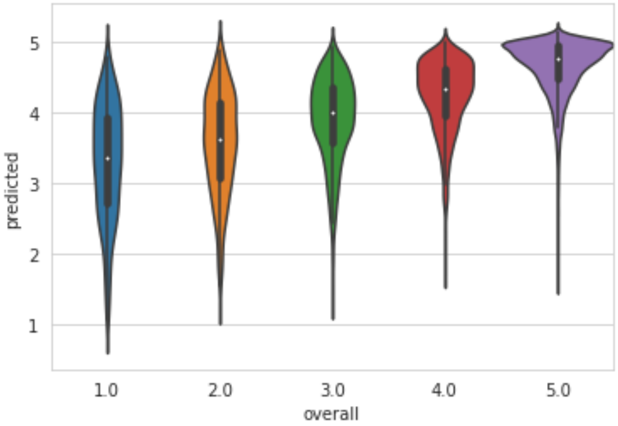
There are also three models based on combination of different algorisms. The predicted values were added to the train and test set. After combining all different models, we may obtain a better model.



The XGBoost regression model has the minimum RMSE of 0.795. However, the RMSE are almost the same from all the combination models such us XGBoost with Baseline model, XGBoost with Baseline plus KNN Baseline models and XGBoost with Baseline plus KNN Baseline plus SVD & SVD++ models. In the experiment, one of above will standout with a difference of 0.0004 for RMSE. We can see that KNNbaseline models did great on the training data but performed the worse on the testing data because of overfitting. But even the worst model has a smaller RMSE than the base models I performed manually.



The average rating of users and average rating of items are no doubt the most important features. The review length is also related to the rating as what we have seen in the exploratory data analysis section. The rest of the top five features are price and number of items user rated. We examined the relationship between price and rating already. The rankings of the top four features never changed among the XGBoost based combine models.

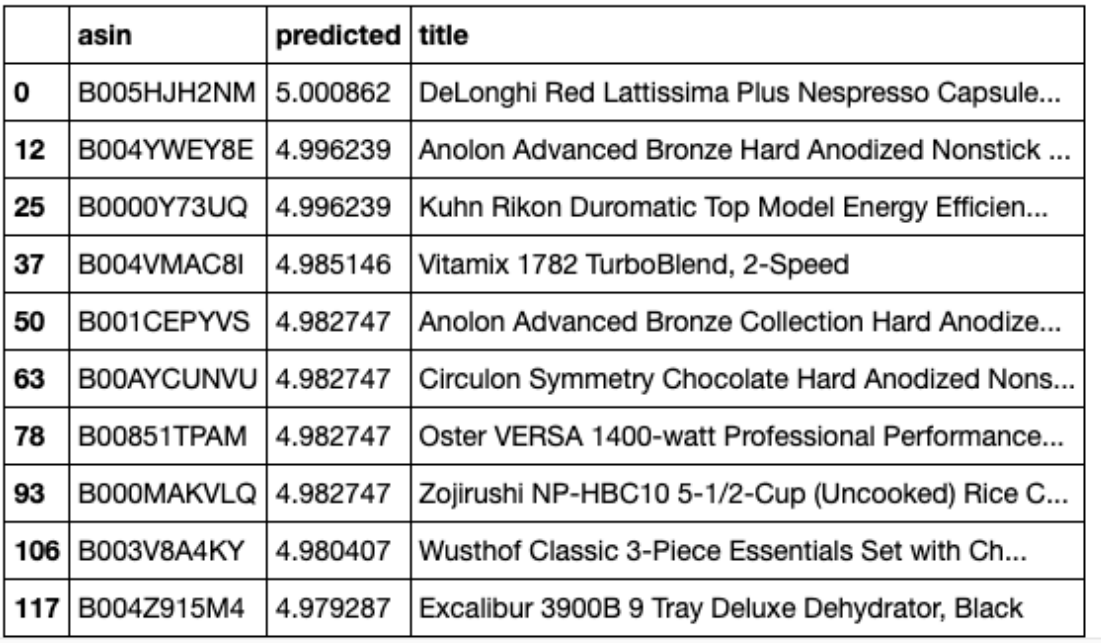


The predictions for 1-4 are all skewed to the higher ratings, because there are so many 5-star rating in the original data that mislead the model to give higher rating. The distribution of predicted range gets wider when the original rating gets lower. For the recommendation system, we want to recommend users items they potential like and prevent showing items they hate. So, prediction for 5-star rating becomes most important. As we can see here, most of the 5 star values were predicted pretty closed. There are chances for rating 1-4 to be predicted as 5, but the chance seems to be relatively low. The mean absolute error is 0.537435, which might lead to one rating lower or higher.

More details about the modeling procedure could be found in this IPython notebook.

6. Prediction

To mimic the working scenario for the recommendation engine, I tried to give out top 10 items they might like using the best model from previous section.



	asin	predicted	title
0	B005HJH2NM	5.000862	DeLonghi Red Lattissima Plus Nespresso Capsule...
12	B004YWEY8E	4.996239	Anolon Advanced Bronze Hard Anodized Nonstick ...
25	B0000Y73UQ	4.996239	Kuhn Rikon Duromatic Top Model Energy Efficien...
37	B004VMAC8I	4.985146	Vitamix 1782 TurboBlend, 2-Speed
50	B001CEPYVS	4.982747	Anolon Advanced Bronze Collection Hard Anodize...
63	B00AYCUNVU	4.982747	Circulon Symmetry Chocolate Hard Anodized Nons...
78	B00851TPAM	4.982747	Oster VERSA 1400-watt Professional Performance...
93	B000MAKVLQ	4.982747	Zojirushi NP-HBC10 5-1/2-Cup (Uncooked) Rice C...
106	B003V8A4KY	4.980407	Wusthof Classic 3-Piece Essentials Set with Ch...
117	B004Z915M4	4.979287	Excalibur 3900B 9 Tray Deluxe Dehydrator, Black

The top 10 items were based on the rank of predicted rating from the model. There are some values needs preprocessing like review length and review time. Because the review data we feed the model are imaginary data, those value can only be the average of user's historic values.

More information about the prediction procedure could be found in the IPython notebook.

7. Conclusions

The model is decent enough to give some recommendation to users based on the XGBoost regression model. However, if the item is new to the system or user is new to the platform, it will be hard to ensure the accuracy.

There are still rooms to improve this model.

1. We could use more recent data to train the model. The dataset for this study only covers data from 1999 to 2014
2. Due to the limitation of time, the model was trained and tested with the trimmed dataset. It will be good to see if the results change or not once the dataset grows bigger.
3. Like mentioned above, the current model didn't solve new user problem. One way to solve this problem could be recommending top-rated times to new users like what Amazon is currently using.
4. There are still ways to develop new features. We can collect more user and item information. For the review content that we already have, we can try to study users' preference from the words they left.