Amazon Recommendation System for Grocery and Gourmet Food

Yifeng He, Wang Xiang

UMSI

Introduction

In America, the biggest online shopping website is Amazon. In addition to having a variety of products, Amazon's recommendation system is also widely praised. A powerful recommendation system can attract users to make unplanned purchases decisions and significantly increases the conversion rate of the website. Users will also feel that the website understands them well, thereby increasing user stickiness.

Objective

1. Perform sentiment analysis based on classification algorithms such as logistic regression (LR), and Naïve Bayes.

2. Design a recommendation system based on collaborative filtering and content-based algorithms to recommend items

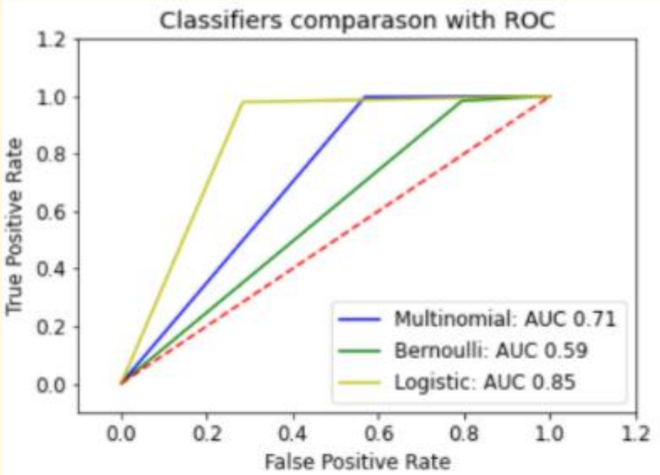
Sentiment Analysis

Use classification algorithm to determine whether the text review is positive or negative

	feature	coefficient
846227	great	15.532734
217286	best	14.568253
1378360	not bad	13.326454
599783	excellent	11.679535
1379656	not bitter	11.135134
1548451	perfect	10.307389
2326992	wonderful	10.208899
166549	awesome	10.104455
1399006	not too	9.843386
498511	delicious	9.301952

	feature	coefficient
519780	didn	-9.290396
2375661	yuck	-9.365434
2103289	three stars	-10.166446
1945645	terrible	-10.342181
970384	horrible	-10.925348
170620	awful	-11.006077
1472563	one star	-11.091099
2179306	two stars	-11.751694
2338542	worst	-13.014682
1376719	not	-14.703824

Coefficient of Word in the LR model



Hybrid Recommendation System

Collaborative Filtering System:
user id, item id

Content Based System:
title id

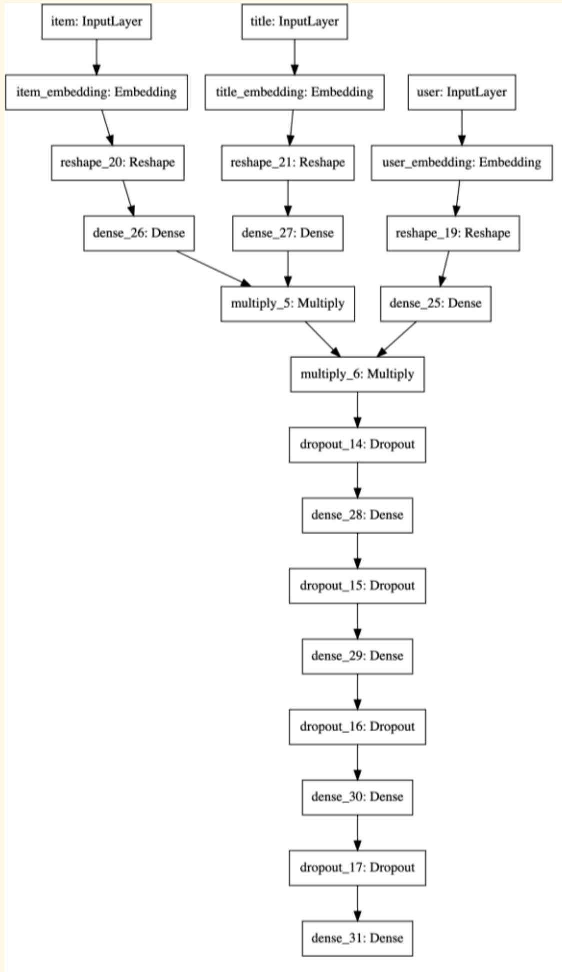
User id: 45302, give 5 ratings product

1. Sweet Leaf Drops Liquid Stevia Sweetener
2. Gluten Free Pancake and Baking Mix
3. Grove Square Cappuccino, French Vanilla
4. Grove Square CARAMEL HOT APPLE CIDER
5. GROVE SQUARE SPICED HOT APPLE CIDER
6. Barilla Gluten Free Pasta
7. Top Brand Coffee, Tea, Cider, Hot Cocoa and Cappuccino Variety Sampler Pack

Top 9 Recommend Items

1. Crunch Bars Peanut Butter Crème
2. Crunch High Protein Energy Snack
3. KIND Bars, Caramel Almond and Sea Salt, Gluten Free
4. KIND Bars, Dark Chocolate Nuts, Sea Salt, Gluten Free
5. KIND Bars, Madagascar Vanilla Almond, Gluten Free
6. Jalapeno flavor Potato Chips
7. KIND Bars, Cranberry Almond, Gluten Free, Low Sugar
8. Peanut Butter Candy
9. Organic Cacao Powder

Structure of the Neural Network



Amazon Recommendation System for Grocery and Gourmet Food

Yifeng He
Industrial & Operations Engineering
University of Michigan
Ann Arbor, MI 48105

Wang Xiang
Industrial & Operations Engineering
University of Michigan
Ann Arbor, MI 48105

1 Introduction

Currently, online shopping is becoming increasingly popular because of its convenience and accessibility. Especially during the pandemic, people greatly rely on online shopping for daily needs. In America, the biggest online shopping website is Amazon. In addition to having a variety of products, Amazon's recommendation system is also widely praised. A powerful recommendation system can attract users to make unplanned purchases decisions and significantly increases the conversion rate of the website. Users will also feel that the website understands them well, thereby increasing user stickiness. Therefore, we decided to design a recommendation system to recommend products on Amazon. The reviews and ratings show users' implicit and explicit feedback on the products, so, by analyzing the reviews and ratings, we can summarize the habits and tendencies of users when shopping online. The whole project is divided into three parts:

1. Perform sentiment analysis based on classification algorithms such as logistic regression, svm, and Naïve Bayes.
2. Design an item-based collaborative filtering model based on K-Nearest Neighbors to find the 2 most similar items. Further, this model can also predict overall ratings based on users' reviews. The prediction result is measured by accuracy, and mean squared error(MSE).
3. Design a recommendation system based on collaborative filtering and content-based algorithms to recommend items. In the collaborative filtering system, we used rating scores given by users as the main feature, and for the content based system, we used item's title as the main feature so the system could recommend products similar to those user rated high. The collaborative filtering system is implemented by singular value decomposition and matrix factorization. Further, deep neural networks are used to realize the hybrid recommendation system, which combines collaborative and content based features.

2 Methodology

In this part, we will talk about how we manipulate data, and implement each algorithm.

2.1 Data Preprocessing

2.1.1 Dataset Overview

The Grocery and Gourmet Food dataset is from the public amazon review dataset in a period between May,1994 and Oct,2018. (https://nijianmo.github.io/amazon/index.html#subsets) Since the original dataset contains a lot of reviews, it is hard to load and manipulate. We used the 5-review dataset, meaning that each of the users and items have 5 reviews each. The dataset we used has 1,143,860 reviews in total, and has 12 features, which are introduced below.

- *Asin:* ID of the product
- *ReviewerName:* name of the reviewer
- *ReviewText:* text of the review
- *Summary:* summary of the review
- *UnixReviewTime:* time of the review(raw)
- *Vote:* helpful votes of the review
- *Style:* Style of the product
- *Image:* Images that users post after they have received the product

2.1.2 Data Preprocessing

	overall	verified	reviewTime	reviewerID	asin	reviewerName	reviewText	summary	unixReviewTime	vote	style	image
0	5	True	11 19, 2014	A1QVBUH9E1V6I8	4639725183	Jamshed Mathur	No adverse comment.	Five Stars	1416355200	NaN	NaN	NaN
1	5	True	10 13, 2016	A3GEOILWLK86XM	4639725183	itsjustme	Gift for college student.	Great product.	1476316800	NaN	NaN	NaN
2	5	True	11 21, 2015	A32RD6L701BIGP	4639725183	Krystal Clifton	If you like strong tea, this is for you. It mi...	Strong	1448064000	NaN	NaN	NaN
3	5	True	08 12, 2015	A2UY1O1FBGKIE6	4639725183	U. Kane	Love the tea. The flavor is way better than th...	Great tea	1439337600	NaN	NaN	NaN
4	5	True	05 28, 2015	A3QHVBQYDV7Z6U	4639725183	The Nana	I have searched everywhere until I browsed Ama...	This is the tea I remembered!	1432771200	NaN	NaN	NaN

Figure 1

Figure 1 shows the top five rows of the dataset. We can notice that there are many missing values in vote, style and image features, so we checked the percentages of missing values for each column, which is shown below.

	column_name	percent_missing
image	image	99.168605
vote	vote	86.169461
style	style	48.237896
reviewText	reviewText	0.034095
summary	summary	0.019146
reviewerName	reviewerName	0.012064
overall	overall	0.000000
verified	verified	0.000000
reviewTime	reviewTime	0.000000
reviewerID	reviewerID	0.000000
asin	asin	0.000000
unixReviewTime	unixReviewTime	0.000000

Figure 2

Since image and style have high percentages of missing values, we dropped those two features directly, but for vote, we think it is an important feature for our analysis, so we filled missing vote values as -1. In terms of reviewText, summary, and reviewerName, we dropped records that have missing values on these three attributes.

2.2 Sentiment analysis

Since sentiment analysis is a classification problem, we decided to use some classification algorithms like logistic regression, svm, and naiveBayes to judge positive and negative reviews.

2.2.1 Logistic Regression & SVM

1 Data Manipulation

First, we added another column called "upvote", and it has several sections based on the number of votes, 'Empty'(-1), '0-50', '50-400', '400-700', and '700-1000'. The distribution of data is shown below. We can find that many votes are in the range 0-50 and most votes are empty (missing values). Another finding is that most products have overall score 5 as shown on the bottom right corner.

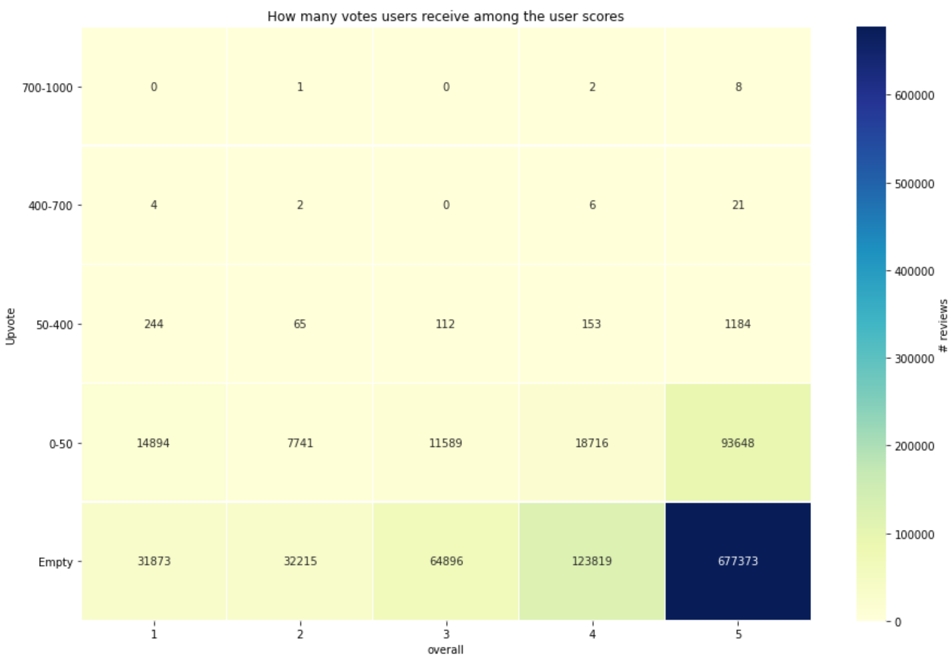


Figure 3

2 Use review text to predict the ratings

In this part, we firstly removed the overall rating of 3, and converted the rating to the binary values: '4' and '5' to positive (1), '1' and '2' to negative (0).

To build the model, we chose Logistic Regression as our main method. Logistic Regression is a statistical model that in its basic form uses a logistic function to model a binary dependent variable.

Now we could apply Logistic Regression on word count and get the result, which shows the coefficients of the words that have obvious positive or negative sentiment orientation.

While we only use Logistic Regression, the result showed that some words with high positive and negative coefficients are not very reasonable (such as pizzas and bridgeford), so we need to optimize the model. One way is to add a TF-IDF vectorizer to Logistic Regression.

The reason we used TF-IDF score is that compared to the simple word count strategy, the TF*IDF algorithms can weigh a keyword in any content and assign the importance to that keyword not only based on the term frequency but also based on the inverse document frequency. The equation of the TF-IDF score is

$$\mathbf{w(t,d) = TF(t,d)*log(} \frac{N}{DF(t)} \mathbf{)}$$

Where:

- TF(t,d) is the number of occurrences of t in document d.
- DF(t) is the number of documents containing the term t.
- N is the total number of documents in the corpus.

The result of the new method showed that some words that had no obvious sentiment orientation were replaced. Then, we tried to further optimize the model: the combination of TF-IDF and n-grams algorithms are applied to Logistic Regression.

An n-gram is a contiguous sequence of n items from a given sample of text or speech. With larger n, a model can store more context with a well-understood space–time tradeoff, enabling small experiments to scale up efficiently. After grid searching, we found that the range of n-gram being (1, 2) could make the result better. As for Logistic Regression, the default hyperparameters are unnecessary to be changed. The result is shown below.

```
# features: 4123432
# train records: 751476
# test records: 250493
Model Accuracy: 0.9558271089411681
```

-Top 20 positive-

Word	Coefficient
delicious	17.415903
great	16.428972
perfect	14.399175
love	14.023578
best	13.928612
excellent	12.969308
nice	12.076147
wonderful	10.877582
yummy	10.613981
awesome	9.854771
amazing	9.848795
loves	9.196059
favorite	8.431807
works	8.384333
fantastic	8.319890
easy	8.063212
yum	7.965631
pleased	7.408866
won disappointed	7.342433
just right	7.326111

-Top 20 negative-

Word	Coefficient
poor	-8.489279
refund	-8.544791
unfortunately	-8.909447
sorry	-8.912128
weak	-9.189067
threw	-9.284270
gross	-9.523193
stale	-9.867208
bland	-9.970491
return	-10.123019
disgusting	-10.260065
yuck	-10.404012
disappointment	-10.633610
tasteless	-10.710083
terrible	-12.254870
horrible	-12.421867
disappointed	-13.149658
awful	-13.246663
disappointing	-13.607846
worst	-14.006600

Figure 4

we could see that the result seemed more reasonable, since we can find that the words which have high coefficients are words with obvious sentiment orientation, and the high coefficients indicate that they have high weights in the model.

3 Use review text to predict the votes

In this part, we set the 'vote' in '0-50' as '0' (negative), and 'vote' > 50 as '1' (positive). From figure 3, we noticed that the data seemed skewed towards the negative side. So, we counted the number of each side when 'overall' equals to 5 to verify this observation. The result shows that the label '0' has 93648 samples, but label '1' only has 1213 samples, so we could find that the data was actually unbalanced. To avoid it, resampling the data had to be performed. We randomly

chose the number of samples with label 1 from the samples with label 0 and got the resampled data: there are 2,426 samples in total, of which label 1 and 0 are half each.
Then we used the similar operations as above to measure accuracy of the model on the resampled data. When we only use Logistic Regression, we could see that the accuracy was not very good (0.718), so we used TF-IDF+n-grams to optimize the model:

```
# features: 153461
# train records: 1819
# test records: 607
Model Accuracy: 0.7248764415156508
```

-Top 20 positive-

Word	Coefficient
coconut	2.088681
oil	1.890879
ve	1.414377
water	1.357736
sugar	1.317366
seeds	1.301365
honey	1.291716
don	1.258669
matcha	1.233196
milk	1.146265
cups	1.125112
let	1.094681
powder	1.064751
eating	1.044805
fat	0.980285
know	0.970051
product	0.944226
high	0.942694
day	0.918764
way	0.910231

-Top 20 negative-

Word	Coefficient
far best	-0.397343
great product	-0.417041
size	-0.433342
ginger	-0.435444
quality	-0.452281
flavorful	-0.492134
flavors	-0.517042
excellent	-0.526972
yummy	-0.534509
crackers	-0.536687
good quality	-0.540328
tasty	-0.546389
nice	-0.622586
order	-0.633933
flavor	-0.799563
perfect	-0.844743
delicious	-0.862145
good	-0.980195
love	-1.508039
great	-1.719112

Figure 5

Figure 5 shows that when we only use 2,426 records, the accuracy was not that good as we use the whole dataset, and the results were not very accurate and reasonable as well. We thought that the reason may be lots of people do not have the habit of voting the reviews, so the quantity of reviews with low votes is quite large. In this dataset, the positive side has too little data, so that our resampled data is not enough to train an accurate model to get a good result.

4 Effect of non-contextual features

We also considered whether the non-contextual features, such as punctuation and question mark, in reviews would have effects on the number of votes. So, we extracted these kinds of features from the reviews, the results are shown below:

Upvote	0	1
word_count	83.653751	243.001649
capital_count	13.780709	41.081616
question_mark	0.111294	0.384171
exclamation_mark	0.840066	1.350371
punctuation	16.164880	50.895301

Figure 6

It seems that for positive reviews (upvote=1 or vote >50), they have more values in these five features. To verify it, we used Logistic Regression and SVM to build the simple default-hyperparameter models and predict the upvotes. The accuracies of the 2 models are: 0.7282 for Logistic Regression and 0.7381 for SVM.

The accuracies of these 2 models were acceptable, which means our assumption is valid. In other words, reviews with more detailed descriptions and details are easier to get votes.

2.2.2 Naïve Bayes & Logistic Regression

1 Data Manipulation

In this part, we created a new feature called sentiment. It has value “positive” if overall score is greater than 3 otherwise, its value is “negative”. In addition, we also created a new feature called usefulScore. It has value useful if it has non-zero vote otherwise its value is useless.

2 Feature Extraction from summary

In this part, we first used regular expressions to get the English reviews for the summary feature. Then, we counted the frequency of each word and a TF-IDF algorithm was employed to get the TF-IDF score for each word, which will be used to predict the “sentiment” value, which are “positive” and “negative”.

Before applying ML models to make predictions, we removed the stopwords and used the wordcloud package to extract some feature words for different ratings.

Figure 7

Figure 8

Figure 7 shows the featured words for products with rating 1, and some representative words are poor, old, disappointed.

Figure 8 shows the featured words for products with rating 5, and some representative words are strong, remembered and great.

3 Predict Sentiment Values

After having tf-idf score for each word as x values, and sentiment results (positive and negative) as y values, we trained Multinomial Naïve Bayes, Bernoulli Naïve Bayes and Logistic Regression models and used models to predict whether the summary of products is positive or negative. The auc-roc results of these three models are shown below.

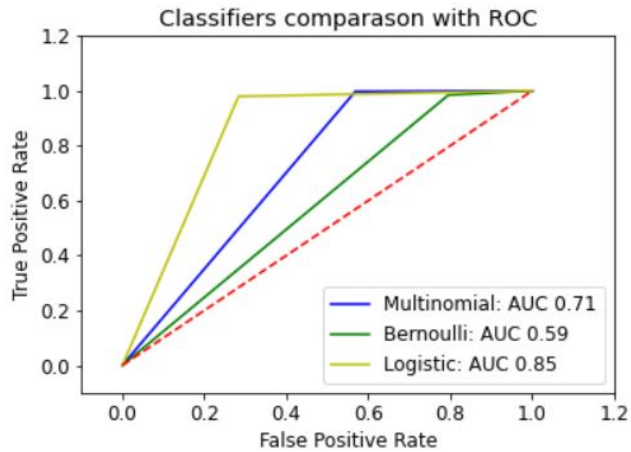


Figure 9

Figure 9 shows that logistic regression has the best prediction ability, and it has the highest auc score 0.85. One possible reason that Naïve Bayes based models perform worse here is that the distribution of word tf-idf scores don't follow either Multinomial or Bernoulli distributions. Afterwards, we also made confusion matrices to show each model's result.

Logistic Regression

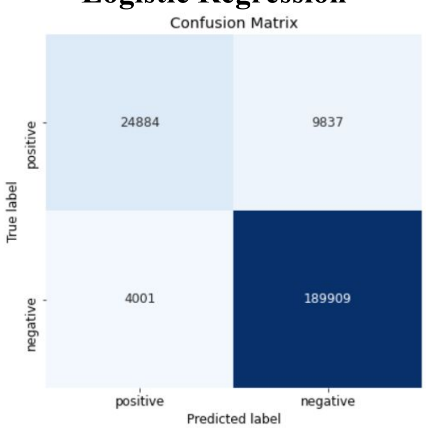


Figure 10

Bernoulli Naïve Bayes

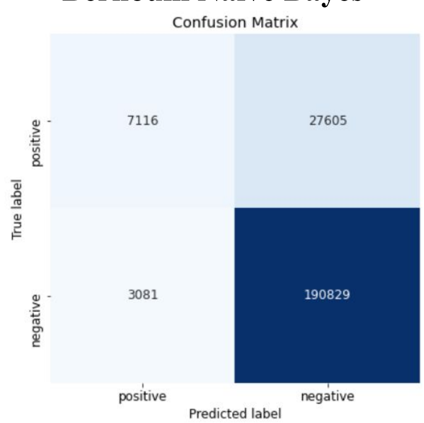


Figure 11

Multinomial Naïve Bayes

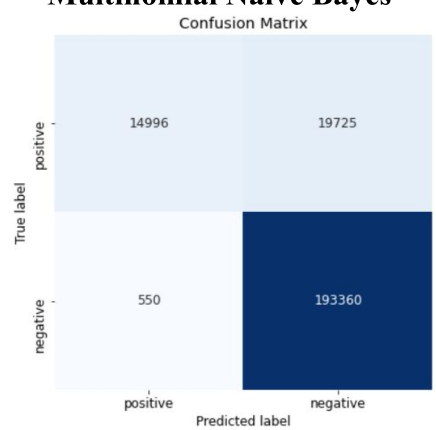


Figure 12

Figure 10, 11 and 12 also show that logistic regression produces the best result since it has the greatest number of true-positive and true-negative predictions.
Further, another advantage of logistic regression is that we could see the importance of word for determination based on its coefficients. The positive coefficient means this word has positive orientation, and negative coefficient means this word has negative orientation. The results are shown below.
Figure 13 displays top 10 words that have the highest coefficient values, and figure 14 displays top 10 words that have the lowest coefficient values. Figure 13 shows that some positive orientation words are "great", "best" and "not bad". Figure 14 shows that some negative orientation words are "not", "worst" and "two stars".
In a short, logistic regression is the best model to do sentiment analysis. One possible reason is that the distribution of words' tf-idf scores are not in either Multinomial or Bernoulli distributions.

	feature	coefficient
846227	great	15.532734
217286	best	14.568253
1378360	not bad	13.326454
599783	excellent	11.679535
1379656	not bitter	11.135134
1548451	perfect	10.307389
2326992	wonderful	10.208899
166549	awesome	10.104455
1399006	not too	9.843386
498511	delicious	9.301952

Figure 13

	feature	coefficient
519780	didn	-9.290396
2375661	yuck	-9.365434
2103289	three stars	-10.166446
1945645	terrible	-10.342181
970384	horrible	-10.925348
170620	awful	-11.006077
1472563	one star	-11.091099
2179306	two stars	-11.751694
2338542	worst	-13.014682
1376719	not	-14.703824

Figure 14

2.3 Item-based Collaborative Filtering Recommendation System

In this part, we built an item-based collaborative filtering recommendation system based on the K-Nearest Neighbors algorithm. One reason that we use KNN is that it can be used both for regression problem and classification problem. Another reason is that the conclusion is easy to understand since the algorithm predicts scores based on the review similarities.

1 Data Manipulation

First, we calculated the mean score for each item, and added all summaries in a list for each product as shown below.

	asin	summaryReview	overall	unixReviewTime
0	B00008RCN8	[I chew too much, Orbit's the Best!, Favorite ...	4.466292	1.435446e+09
1	B0000CFPI2	[Five Stars, This flavor rocks, Five Stars, EL...	4.560386	1.430755e+09
2	B0000D9169	[My local Publix Supermarket has better cookie...	4.477528	1.486752e+09
3	B0000D916Y	[Five Stars, Five Stars, Pretty good, Excellen...	4.477528	1.486752e+09
4	B0000DHXGL	[These "First to Live ALMONDS." are "Top Quali...	4.352349	1.446733e+09

Figure 15

Then, we filtered reviews to keep english reviews only by using regular expressions. Before counting the frequency of each word for each product, we also removed stop words. The result after counting the word frequency is shown below.

	absolutely	add	added	addictive	aftertaste	almond	almonds	alternative	amazing	amazon	apple	arrived	awesome	bad	bag	bags	baking	bar	bars	beans	best
0	0	1	0	0	0	0	0	0	0	1	0	0	1	1	0	0	0	1	0	0	9
1	1	0	0	1	0	0	0	0	2	0	0	0	0	0	0	0	0	0	0	0	13
2	1	0	0	0	0	0	0	0	0	0	0	3	1	0	0	0	0	0	0	0	11
3	1	0	0	0	0	12	54	0	1	0	0	0	1	4	2	0	0	0	0	0	12
4	0	1	0	0	1	0	0	0	1	0	0	0	1	1	1	1	0	0	0	0	11
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
1621	0	0	2	1	1	20	2	6	1	0	0	0	5	2	0	0	3	0	0	0	4
1622	0	0	0	0	0	4	42	0	0	3	0	3	3	0	3	0	0	0	0	0	7
1623	1	0	1	1	0	0	0	0	1	0	0	0	4	1	0	0	0	0	0	0	6
1624	0	0	0	0	0	0	0	2	0	0	0	1	2	8	1	0	1	0	0	0	10
1625	0	0	0	0	0	5	53	0	4	0	0	1	3	0	0	0	0	0	0	0	15

1626 rows × 300 columns

Figure 16

Each row can be seen as a vector representation for that product.

2 Model Training & Results

After getting the vector representation for each product, we trained a KNN model to find the top 2 similar products given a product. Some results are shown below.

```
Based on product reviews, for  B00GRNUJYQ  average rating is  4.066921606118547
The first similar product is  B007PA34CE  average rating is  4.619658119658119
The second similar product is  B0029XLH4Y  average rating is  4.548654244306419
================================================================
Based on product reviews, for  B00GRY2Z5Q  average rating is  4.296551724137931
The first similar product is  B007BIUB62  average rating is  4.1946902654867255
The second similar product is  B0019LTH3U  average rating is  4.590551181102362
```

Figure 17

Figure 17 shows that similar products have similar ratings because we used review summaries as x values. Further, we also applied KNN to predict product scores. We tried three different algorithms, 'ball_tree', 'kd_tree', and 'brute force search' and two different k values. In addition, these three algorithms are different strategies to find similar products. The results are shown below.

Parameter for KNN	Accuracy	Mean Squared Error
K=3, ball tree	0.94	0.06
K=5, ball tree	0.93	0.07
k=3, brute force search	0.94	0.06
k=3, KD tree	0.94	0.06

Table 1

Table 1 shows that the results are almost the same for these three algorithms, and using the nearest 3 neighbors is better than choosing the nearest 5 neighbors to predict scores. However, although we could predict the mean score of products at this time, the system can only recommend products that are similar to the products he/she likes. In other words, for those products he may like but are not similar, he will never be recommended. This problem will be solved in the third part.

2.4 Recommendation System based on rating score prediction

2.4.0 The baseline method

The baseline method used the average score of the whole dataset as the prediction scores, and the mean squared error between the baseline prediction scores and the rating scores in the test dataset is 1.13. We measured our method by comparing the mse value of our method and the mse value of the baseline method.

2.4.1 Singular Value Decomposition

Singular Value Decomposition is a latent factor model, which can extract features and correlation form the user-item matrix. [1] Given the sparse matrix of user-item ratings, after applying SVD, we could predict ratings for products that users have not rated before.

1 Data Manipulation

Since the given dataset only contains user rating scores for some products, we constructed a sparse matrix that contains this part of the ratings, and set the user-item ratings that are not given as 0. Part of the matrix is shown below.

productId	9742356831	B000052X2S	B000052Y74	B00005BPQ9	B00006BN4U	B00006FMLY	B00006IDJU	B00006IDK9	B00008RCN8
userId									
A100WO06OQR8BQ	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
A1047EDJ84IMAS	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
A10AFVU66A79Y1	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
A11ED8O95W2103	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
A11OTLEDSW8ZXD	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
A11TT460OXJPVA	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
A11WNQ3PPU73Y1	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
A12CBDHX1MJLGZ	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
A12FLMSWRKK2IK	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	5.0
A12JTIKL4N0H3V	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0

Figure 18

Figure 18 shows that since each user only rated a limited number of products, most of values in this matrix is 0.

2 Model Training & Results

After having the matrix, we could apply SVD algorithms to predict users' rating scores. Firstly, we used the svds() package from scipy sparse linalg package and set the latent factor to be 10 to get three matrices, which are user-latent factor, latent factor- latent factor, and latent factor - item matrices. Then, we multiplied these three matrices together to get a large matrix, which contains user ratings for all items. The result is shown below.

productId	9742356831	B000052X2S	B000052Y74	B00005BPQ9	B00006BN4U	B00006FMLY	B00006IDJU	B00006IDK9	B00008RCN8	B0000A0BS3
0	0.016738	0.003311	0.001548	0.013460	0.013828	0.042272	0.002261	0.012844	0.028748	0.012507
1	0.012544	-0.007123	-0.002440	0.047498	-0.005212	0.027515	0.000242	-0.001092	-0.030589	-0.010970
2	0.004250	0.012231	-0.002476	0.049063	0.001796	0.036033	-0.000204	0.005759	0.001665	-0.009531
3	0.006533	0.002814	0.000125	0.004718	0.002629	-0.006892	0.000447	0.002209	0.006263	0.000172
4	0.001457	0.026123	-0.001024	0.007957	0.024658	-0.059820	-0.001318	0.039607	0.082744	-0.001839

Figure 19

Figure 19 shows that the predicted scores are very low overall, which is inconsistent with the actual situation. Further, based on these prediction scores, we could infer that the mse of this method must be larger than the mse of the baseline. One possible reason is that the given number of ratings is 57,061, but the possible number of ratings in the sparse matrix is 16,108,488. So, the density of this matrix is only 0.35%, which is not enough to predict users' ratings. Another reason is that we may also include user bias, movie bias, and global bias in the prediction function.

In a short, this method does not work very well, and may be improved in the future.

2.4.2 Matrix Factorization

SVD is one of matrix factorization algorithms. The difference between this part and the previous part is that instead of using svds() function, we performed a dot product between the respective user and item embeddings, and using deep neural network frame to predict scores. One advantage of this method is that at this time we don't need to construct a sparse matrix to train the model, which may make our prediction results better.

1 Data Manipulation

Since the original user id has many characters, if we use them directly, it will increase the amount of calculation, so we add a new column to the dataset, which records the new user id for each user. The new user id starts from 0, and ends at 127,474. Similarly, the original item id also has many characters, so we add a new column to record the new item id, which starts from 0, and ends at 41319. The new dataset is shown below.

	asin	summary	reviewText	score	user_id	user_name	reviewTime	old_item_id	item_id	old_user_id
0	4639725183	Five Stars	No adverse comment.	5	0	Jamshed Mathur	11 19, 2014	4639725183	0	A1QVBUH9E1V6I8
1	4639725183	Great product.	Gift for college student.	5	1	itsjustme	10 13, 2016	4639725183	0	A3GEOILWLK86XM
2	4639725183	Strong	If you like strong tea, this is for you. It mi...	5	2	Krystal Clifton	11 21, 2015	4639725183	0	A32RD6L701BIGP

Figure 20

2 Model Training & Evaluation

User and item embeddings are the 'user_id' column and 'item_id' columns respectively in figure 20. Then, we used these two embeddings as inputs, and the dot product of them as output to train a deep neural network. The structure of the neural network is shown below.

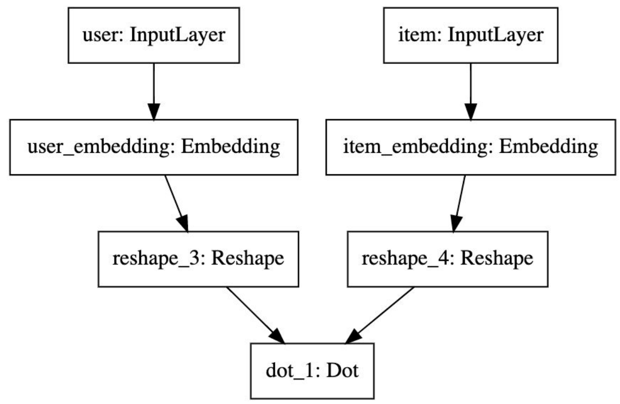


Figure 21

However, the mean square error between the prediction scores and the actual scores is 15.02, which has much larger mse value then the mse value of the baseline method, so this method should not be used for our recommendation algorithm.

2.4.3 Recommendation System based on hybrid recommendation system

In this part, we decided to create a hybrid architecture that combines collaborative filtering and content-based approaches. In the collaborative part, the system will learn item similarities from user's interaction and recommend to the user items which he is likely to rate high according to learnt item & user embeddings, and in the content based part, the system will learn item similarities from item's metadata attributes (such as price and title), and recommend to the user contents similar to those he rated high. The system is also built by using deep neural networks.

1 Data Manipulation

First, we used the same way previously to get the user embedding and the item embedding. In addition, at this time, we also imported another dataset from Amazon, which includes the “title” and the “price” features. Then, we merge these two datasets based on the same attribute “asin”. However, the percentage of missing values for “price” has 54%, so we decided to use the “title” feature only for the content based part. Similarly, since the deep neural network only accepts numeric features, we add a new column called “title_id” to give each title an id. This feature is the title embeddings we are going to use in the deep neural network.

2 Model Training & Evaluation

In this part, we use id, item id and title id as inputs to the deep neural network, and the output is the prediction score. For the hyperparameter tuning, we tried a different number of layers, number of embedding sizes, drop out and number of epochs to find the best model structure, and the results are shown below.

Layers	Embedding Size	Drop Out	Epoches	MSE
3	10	Once	10	1.35
3	15	Three Times	3	0.99
3	50	Three Times	3	1.21
3	15	Three Times	2	0.95

Table 2

Table 21 shows that the best deep learning model should have 3 layers, 15 embedding sizes for each feature, dropout layer after each layer and 2 epochs to avoid overfitting. The structure of our neural network is shown below.

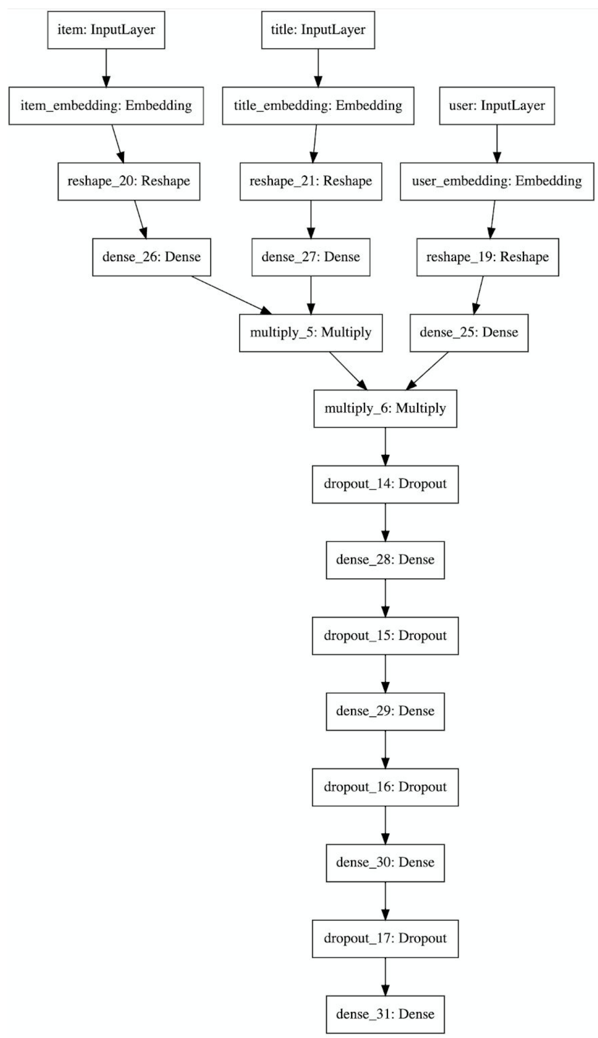


Figure 22

Further, since the mse of this model is 0.95, which is smaller than the baseline model, we could use it for recommendations. To save the calculation time, we sampled 100 user ids, and 100 items, and did a cross join of these two datasets to get 10,000 records in total. Then, use this dataset as input for the trained neural network model to predict scores. Based on the predicted scores, recommend items to users. One example is shown below.

```
SweetLeaf Sweet Drops Liquid Stevia Sweetener, English Toffee, 2 Ounce
Betty Crocker Bisquick Baking Mix, Gluten Free Pancake and Baking Mix, 16 Oz Box (Pack of 3)
Grove Square Cappuccino, French Vanilla, 24 Count Single Serve Cups
Grove Square CARAMEL HOT APPLE CIDER - 12 Single serve cups
Grove Square SPICED HOT APPLE CIDER - 12 cups
Barilla Gluten Free Pasta, Elbows, 12 Ounce (Pack of 12)
SweetLeaf Sweet Drops Liquid Stevia Sweetener, Chocolate, 2 Ounce
30-count Top Brand Coffee, Tea, Cider, Hot Cocoa and Cappuccino Variety Sampler Pack, Single-Serve Cups
```

Figure 23

```
Bionutritional Power Crunch Bars Peanut Butter Creme,  1.4 oz., 12 Bars
Power Crunch High Protein Energy Snack, Cookies &amp; Creme, 1.4-Ounce Bars (Pack of 12)
KIND Bars, Caramel Almond and Sea Salt, Gluten Free, 1.4 Ounce Bars, 12 Count
KIND Bars, Dark Chocolate Nuts &amp; Sea Salt, Gluten Free, 1.4 Ounce Bars, 12 Count
KIND Bars, Dark Chocolate Nuts &amp; Sea Salt, Gluten Free, 1.4 Ounce Bars, 12 Count
Kind Bars, Madagascar Vanilla Almond, Gluten Free, Low Sugar, 1.4oz
Kettle Brand Potato Chips, Jalapeno, Single-Serve 1 Ounce Bags (Pack of 72)
KIND Bars, Cranberry Almond + Antioxidants with Macadamia Nuts, Gluten Free, Low Sugar, 1.4oz, 12 Count
REESE'S Pieces Peanut Butter Candy (Pack of 18)
Viva Naturals #1 Best Selling Certified Organic Cacao Powder from Superior Criollo Beans, 1 LB Bag
```

Figure 24

Figure 24 shows products that user “45302” give 5 scores, which is the highest score a user can give, and figure 24 shows top 10 products we recommended. Although, our recommended products are not from the same category he/she likes, there is a high possibility that the user will like them since these two groups of food are very sweet, and some foods are also gluten free.

3 Discussion & Conclusion

After finishing all the above work, we have achieved the following results:

1. A sentiment analysis model based on the logistic regression, which can analyze the overall emotional tendency of the product reviews based on the user's comments on the product, and prepare for the next part of the recommendation system。
2. Recommendation systems, including item-based collaborative filtering system, and hybrid system can find the similarity between products effectively and provide users convincing personalized recommendations.

Through the methods of model evaluation, the overall effect of the system is at a trustworthy level. But, if the amount of data can be more sufficient, the system can be more convincing and practical. For instance, since the dataset only provides five reviews for each customer and five reviews for each product, the customer-item matrix is sparse, which only has 0.35% non-zero values. If more ratings are provided, the recommendation system based on the SVD algorithm could provide more convincing results. Further, in this project, the SVD algorithm does not provide convincing results. In the future, maybe we could include user bias, item bias, and global bias in the prediction function, and try different numbers of latent factors to make it better.

4 Reference

1. Chen, D. (2020, August 06). Recommender System - singular value decomposition (SVD) & truncated SVD. Retrieved December 08, 2020, from https://towardsdatascience.com/recommender-system-singular-value-decomposition-svd-truncated-svd-97096338f361

5 Amount of Effort

Yifeng He:Naive Bayes & Logistic Regression, Hybrid Recommendation System(Deep Neural Network),Item-based Collaborative Filtering Recommendation System (part of), Poster

Wang Xiang: Logistic Regression & SVM Sentiment Analysis, Recommendation System based on rating score prediction, Item-based Collaborative Filtering Recommendation System (part of)

6 Google Docs Link(Including code & Data File):

https://drive.google.com/drive/folders/1x9ZcLqX6GNsSLIVQqRxlxJ-_zniMzghk?usp=sharing