Amazon Recommender System with user sentiment

Dixitha Kasturi
Venkata Siva Bhargav Konakanchi

Overview :

- The aim of the project is to build a recommendation system(using the user ratings) and also perform sentiment analysis , to understand the overall User sentiment(negative/positive). A total of over 278,677 ***Clothing,shoes and Jewelry*** reviews were analyzed from the 'Amazon Reviews' dataset, which had other categories as well.
- Platforms/Sources:
  - Google colab for code execution
  - Dataset Link
- Algorithms:
  - Recommendation System – ALS (Alternating least squares)
  - Sentiment Analysis - Logistic Regression

Data :

- The Amazon reviews dataset contains product reviews and metadata from Amazon, including 142.8 million reviews spanning May 1996 - July 2014. This dataset includes reviews (ratings, text, helpfulness votes, review time and so on). The file is in JSON format.
- A subcategory of 'Clothing, Shoes and Jewelry' is chosen. It has 39387 unique users gave reviews to 23033 distinct products.
- Overall there are 278,677 reviews and 9 attributes

```
+-------+-----------------+
|summary|          overall|
+-------+-----------------+
|  count|           278677|
|   mean|4.245133254628118|
| stddev|1.103747165196137|
|    min|              1.0|
|    max|              5.0|
+-------+-----------------+
```

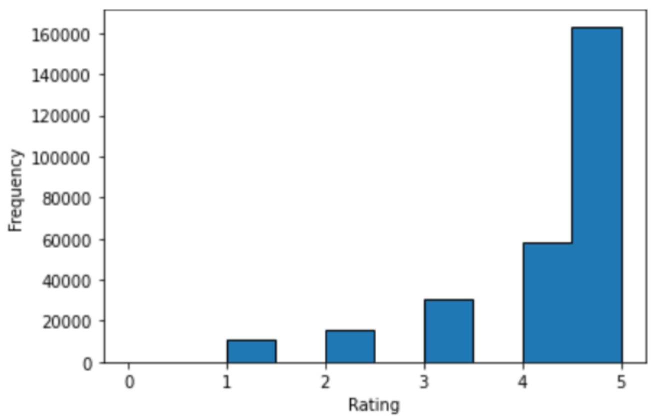
- Throughout the analysis, other columns were generated and added as required

asin	helpful	overall	reviewText	reviewTime	reviewerID	reviewerName	summary	unixReviewTime
0000031887	[0, 0]	5.0	This is a great t...	02 12, 2011	A1KLRMWW2FWPL4	Amazon Customer "...	Great tutu- not ...	1297468800
0000031887	[0, 0]	5.0	I bought this for...	01 19, 2013	A2G5TCU2WDFZ65	Amazon Customer	Very Cute!!	1358553600

Format of the reviews:

{
"reviewerID": "A2SUAM1J3GNN3B",
"asin": "0000013714",
"reviewerName": "J. McDonald",
"helpful": [2, 3],
"reviewText": "I bought this for my husband who plays the piano. He is having a wonderful time playing these old hymns. The music is at times hard to read because we think the book was published for singing from more than playing from. Great purchase though!",
"overall": 5.0,
"summary": "Heavenly Highway Hymns",
"unixReviewTime": 1252800000,
"reviewTime": "09 13, 2009"
}
where
reviewerID - ID of the reviewer, e.g. A2SUAM1J3GNN3B
asin - ID of the product, e.g. 0000013714
reviewerName - name of the reviewer
helpful - helpfulness rating of the review, e.g. 2/3
reviewText - text of the review
overall - rating of the product
summary - summary of the review
unixReviewTime - time of the review (unix time)
reviewTime - time of the review (raw)

Stats for Overall rating :



Overall, there were no ratings that were 0. Fewer ratings had 1,2 value. Majority of the ratings were 4,5

I. <u>Recommendation System</u>

Overview

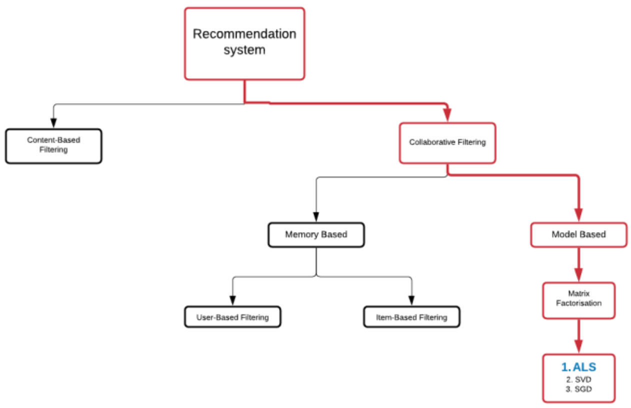
From all the available methods/techniques, Collaborative Filtering was used. g. It's called collaborative because it makes recommendations based on other people in effect, people collaborate (the algorithm does this) to come up with recommendations. This method aims to fill in the missing entries of a user-item association matrix. We will only be considering users and what items a user has interacted with (here interaction means which products the user has given a review/rating for). In real world, clicks/views besides what is bought previously and what ratings are given, are all used.

We are dealing with Explicit data(ratings) instead of implicit(views). For instance according to our data, with ratings we know that a 1 means the user did not like that item and a 5 that he/she really liked it. Using our interaction term(ratings) from other users and the considered user, we generate recommendations of products which he/she might like.

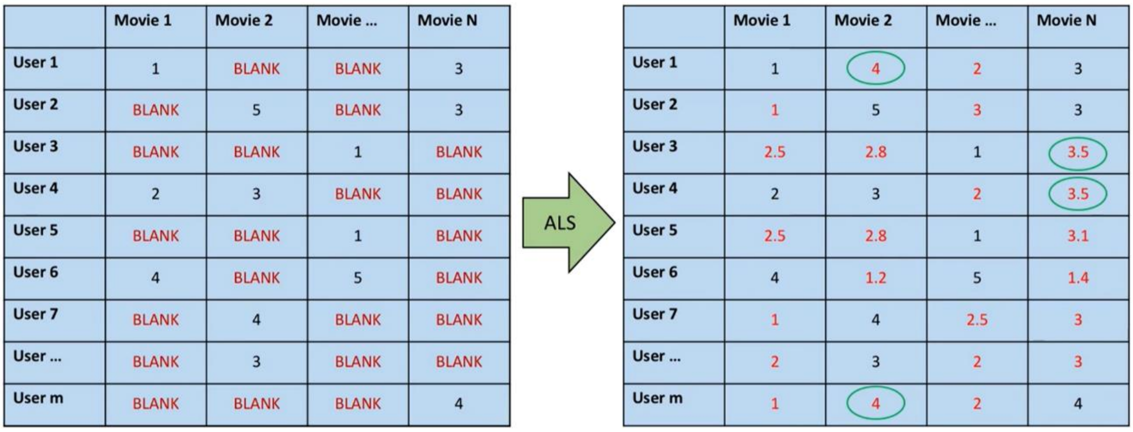
Approach used:

1. Checked the sparsity of user-item matrix
2. Converting all columns to Numeric for ALS
3. 70:30 Training and Testing split
4. ALS model generation with parameter tuning
5. Evaluating RMSE
6. Generating Recommendations

Under collaborative filtering, the one that is supported by spark is Matrix Factorization method known as ALS(Alternating least squares).



Alternating least Squares



	Movie 1	Movie 2	Movie ...	Movie N
User 1	1	BLANK	BLANK	3
User 2	BLANK	5	BLANK	3
User 3	BLANK	BLANK	1	BLANK
User 4	2	3	BLANK	BLANK
User 5	BLANK	BLANK	1	BLANK
User 6	4	BLANK	5	BLANK
User 7	BLANK	4	BLANK	BLANK
User ...	BLANK	3	BLANK	BLANK
User m	BLANK	BLANK	BLANK	4

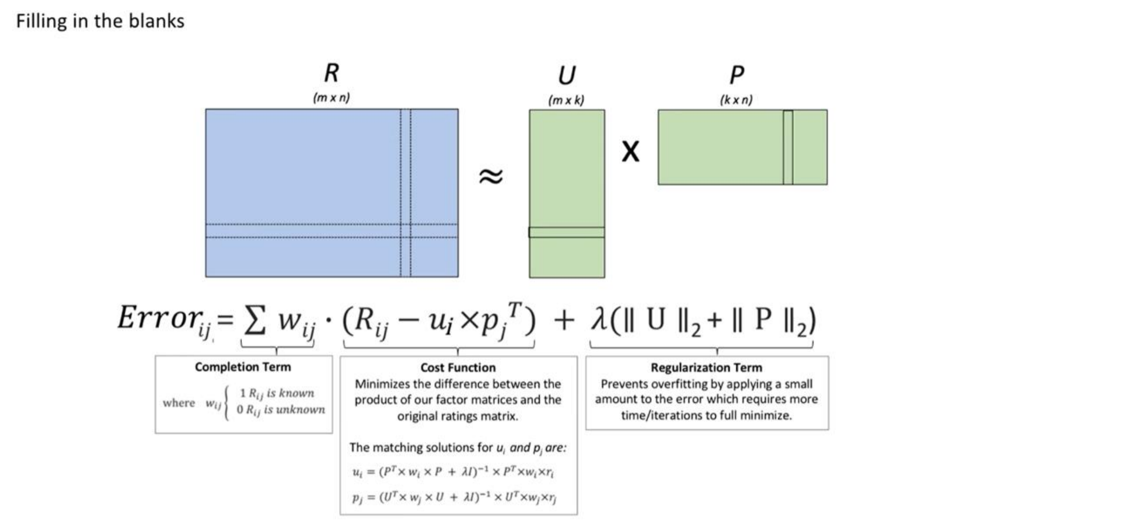
	Movie 1	Movie 2	Movie ...	Movie N
User 1	1	4	2	3
User 2	1	5	3	3
User 3	2.5	2.8	1	3.5
User 4	2	3	2	3.5
User 5	2.5	2.8	1	3.1
User 6	4	1.2	5	1.4
User 7	1	4	2.5	3
User ...	2	3	2	3
User m	1	4	2	4

The user-item utility matrix R where the values denote how item i has been rated by user u on a scale of 1–5. It is a sparse matrix. The goal is to generate values that are missing, highest values turn out to be recommendations for that particular user(marked in green).

- Latent factor model based collaborative filtering learns the user-item profiles(dimension K) through matrix factorization by minimizing the Root Mean Squared Error(RMSE) between the available ratings 'y' and their predicted values y^. Each item i is associated with a latent (feature) vector P, each user is associated with a latent (profile) vector U, and the rating y^(ui).
- ALS uses L2 regularization to reduce the errors by fitting the function appropriately on the given training set and avoid overfitting by reducing the complexity.The weights of features are handled by L2 regularization.L2 regularization forces weights towards zero but it does not make them exactly zero as it removes a small percentage of weights after each iteration. The parameter to tune is Lambda.

$$L_{ridge}(\hat{\beta}) = \sum_{i=1}^{n}(y_i - x_i'\hat{\beta})^2 + \lambda\sum_{j=1}^{m}\hat{\beta}_j^2 = ||y - X\hat{\beta}||^2 + \lambda||\hat{\beta}||^2.$$

- Finally the way ALS works is shown in the image below:



- The values of U and P are generated by alternating the multiplications. When finding/approximaitng values for one(U or P) the other(P or U) takes up random values and is fixed.
    - Fixing U to solve for P
    - Fixing U to solve for U
- Advantage of ALS: Don't need domain knowledge, the embeddings are automatically learnt.
- Disaddvantage of ALS : if an item is not seen during training, the system will not be able to create an embedding for it and query the model with this item. This issue is often called the ***cold-start problem*** .
- Used paramter tuning with cross validation to take the best model possible. Values chosen for Rank = [10,25] Values chosen for lambda(regularization parameter) = [0.05,0.1]
- Did 3-fold cross validation using the training split. Because the size of data is huge and comparing more models is time consuming , we took parameters less models for comparision.

- output recommendations :

asin	asin_index	reviewerID	reviewerID_index	overall	Average_rating_by_product	Average_rating_by_user
B00ECBEVLW	22759	A1NW1EFJD9WSKU	26581.0	4.0	4.8	2.4
B00EB72ND0	22758	A1XKQX71GJASJR	281.0	5.0	4.4	4.260869565217392
B008Z9E1MQ	21667	A1D2UNTX8IROLO	399.0	5.0	5.0	4.238095238095238
B008KFDD06	21512	A17RCXXL4169ZS	15673.0	5.0	5.0	5.0
B007S9UNRG	21102	A2Q6LTSB44MXKN	12548.0	5.0	4.8	3.5714285714285716
B007177F4E	20852	A3J16COE3SEIRM	2077.0	3.0	4.6	3.4615384615384617
B006QOILYQ	20765	AEB9TSD44W1P9	4974.0	5.0	5.0	4.7
B004NOMTP8	19930	A1N9V6ZXXJ0AE	26504.0	5.0	4.8	4.0
B0040HGRZQ	19619	A18YLTJKG5S7OK	10530.0	5.0	4.8	3.857142857142857
B00HXRUMDI	17872	A59BAUCCVXG3M	35882.0	4.0	4.833333333333333	4.0
B008HNJ5WQ	16893	A2788ZJSKB1CHM	8103.0	4.0	4.833333333333333	4.625
B004N2G8F2	15829	A2B14TU8SJPVZJ	18186.0	5.0	4.833333333333333	3.3333333333333335
B008H6GUBM	13773	A1I6C5L79Q8FAN	16317.0	5.0	4.857142857142857	5.0
B003U8TK5A	12757	A5B4E30CSOLU9	35890.0	3.0	4.571428571428571	3.8
B006LASEHE	11218	A1EL9AJBAWCAGD	3043.0	5.0	5.0	3.909090909090909
B004I5BUY8	10899	ADKIM2ECCR9CR	4971.0	5.0	4.875	4.2
B000F24I9M	10256	A1R8LRC0DE0OLP	26961.0	5.0	4.625	4.6
B0043RTPYS	9278	A2N9DD9X8HFRUO	30589.0	5.0	5.0	5.0
B008DJ4PG0	8426	A1BG13HMHSONHB	7367.0	5.0	5.0	5.0
B0059DZ3CI	8157	A1EEU4LS61USRA	16061.0	5.0	4.8	3.6666666666666665

II. User Sentiment Analysis

Overview of customer sentiment analysis:

Using the text review that was given by the user, sentiment analysis was performed using Logistic regression to understand the overall sentiment of the user.

Used all reviews given by the user to get their sentiment score. If their overall sentiment score is negative and the average rating given by them is <=2(we decided the threshold) or if the user overall sentiment score is positive and rating = 5, then further analysis should be peformed, where you look into the products reviewed to see if they are targetting a particular company's products.

The steps that were followed are:

1. For the input data:

   a) Tokenizing words(breaking down sentences into words)

   b) Removing stop words (commonly occuring filler words like articles, pronouns etc)

   c).Converting the words into features(The column features is a sparse vector representation of the words that appeared in the text

asin	reviewerID	reviewerName	reviewText	summary	sentiment	avg_sentiment	score	asin_index	reviewerID_index	words	features	tfidf
B000XXDJ70	A10IZ4YHFKDRY3	Mommyof4Hgirls	I bought this mag...	Very informative ...	1	1.0	1.0	97.0	1289.0	[i, bought, this,...	(47837,[1,2,3,4,5...	(47837,[1,2,3,4,5...
B00007AVYI	A11HD68D6QXXQ6	Fuzzy Dunlop	Love the magazine...	Great magazine fo...	1	1.0	1.0	41.0	11764.0	[love, the, magaz...	(47837,[0,1,2,3,5...	(47837,[0,1,2,3,5...
B00007M3M1	A13QQ6ILELMFTV	Noelle S	After subscribing...	A great publicati...	1	0.2	1.0	1850.0	12757.0	[after, subscribi...	(47837,[0,1,2,3,4...	(47837,[0,1,2,3,4...
B00007M3M1	A13QQ6ILELMFTV	Noelle S	After subscribing...	A great publicati...	-1	0.2	1.0	1850.0	12757.0	[after, subscribi...	(47837,[0,1,2,3,4...	(47837,[0,1,2,3,4...
B00HG1BOWO	A13SN2TIUFDZEI	David Loomis	The hat was cool	Four Stars	1	1.0	1.0	84.0	12777.0	[the, hat, was, c...	(47837,[0,25,615,...	(47837,[0,25,615,...

2. Get positive and negative words from Parquet file
3. For each review, generating the sentiment score for each review (1 for positive and 0 for negative). Positive implies average score > 0.
4. Using Tf-idf, we reduce the words by this numerical statistic that is intended to reflect how important a word is to a review.

   $$tf\text{-}idf_{ij}=f_{ij} * \log|D|+1/f_i +1$$

5. Using logistic regression to classify if the given review based on the tf-idf features, is positive(1) or negative(0).
6. Flag users based on their sentiment score, average rating
   - Negative/flagged if sentiment score = 0 and average rating < 2
   - overly positive if sentiment score = 1 and average rating = 5

Logistic Regression:

- The target variable here is to predict positive/negative sentiment. This is categorical. Binary logistic regression is used.
- It uses linear or non-linear sigmoid function as decision boundary.

$$y=1/1+\exp(x)$$

- To avoid overfitting Elastic net Regularization is used, which is a combination of both L1 and L2 regularization.

$$L_{\lambda,\alpha}\theta(p(X),Y)=-\left(\sum_i Y_i \log p_\theta(X_i)+(1-Y_i)\log(1-p_\theta(X_i))\right)+\lambda\left[(1-\alpha)\sum_{j>0}\theta_j^2+\alpha\sum_{j>0}|\theta_j|\right]$$

- HyperParamter tuning was performed, to select the model that gave the highest accuracy(meaning reduced cost). 60:30:10 training:validation:testing split was used. Based on the best accuracy generated, the final model was chosen to fit the testing set.

- We have 3 phases:
  - Using only logistic Regression (positive and negative words are overfit, weights are wrongly assigned),

	word	weight
34757	blurt	-34.666115
30548	30min	-30.935024
29894	quiltingarts	-30.377604
45255	corals	-29.680090
21432	everyones	-28.510804

	word	weight
18674	amsterdam	57.746247
20334	despaired	56.693760
23261	dilled	56.693760
25372	crabs	55.456693
29008	cleanliness	53.939322

  - Using Logistic regression with Elastic Net Regularization (weights are corrected)

	word	weight
28	issue	-0.484691
64	issues	-0.327007
270	disappointed	-0.325444
499	waste	-0.264010
1111	terrible	-0.226651
334	bad	-0.225091
231	hard	-0.208609
1192	horrible	-0.204461
1392	beware	-0.196518
460	wrong	-0.192581
652	boring	-0.191191
329	problem	-0.190610
517	miss	-0.186197
1105	worst	-0.185020
1125	shame	-0.183722

	word	weight
26	great	0.599625
39	love	0.438583
40	good	0.395814
29	like	0.297722
73	well	0.283044
104	best	0.245362
108	enjoy	0.240776
165	easy	0.233084
199	loves	0.228417
109	interesting	0.221498
210	excellent	0.221154
184	favorite	0.192389
206	nice	0.178763
180	recommend	0.175403
287	wonderful	0.172421

  - Parameter tuning with different lambda values to get the highest accuracy.

```
---------+-------------+------------------+------------------+------------------+---------------------+---------+-----------------+-----+----
     asin|   reviewerID|      reviewerName|        reviewText|           summary|Average_rating_by_user|sentiment|    avg_sentiment|score|flag|
---------+-------------+------------------+------------------+------------------+---------------------+---------+-----------------+-----+----
000031887|A1JR9KKF6UKUWW|     Queens Meadow|Bought this for m...|must have for a f...|                  4.0|        1|              1.0|  1.0|  NA|
000031887|A1KLRMWW2FWPL4|Amazon Customer "...|This is a great t...|Great tutu-  not ...|    4.285714285714286|        1|0.3333333333333333|  1.0|  NA|
000031887|A1KLRMWW2FWPL4|Amazon Customer "...|This is a great t...|Great tutu-  not ...|    4.285714285714286|       -1|0.3333333333333333|  1.0|  NA|
000031887|A26A4KKLAVTMCC|         Moonlight|My 3yr old loved ...|Came apart in 2we...|                  3.5|        1|              0.6|  1.0|  NA|
000031887|A26A4KKLAVTMCC|         Moonlight|My 3yr old loved ...|Came apart in 2we...|                  3.5|       -1|              0.6|  1.0|  NA|
---------+-------------+------------------+------------------+------------------+---------------------+---------+-----------------+-----+----
```

Future Scope/Extensions:

A more indepth analysis can be done by using the timestamp that is given in the dataset to understand if there are any targeted times for negative reviews and if meta data can be used, then interpreting the product ids and images to make it more understandable.

Other algorithms like clustering techniques for the recommender system and Naïve Bayes, Lstm, XGBoost could be used for Sentiment analysis for better understanding if a user behavior.

Because there are many records, training took a very longtime.