Recommendation System for online products using Collaborative filtering.

Sheshank Makkapati
Sjsu id:013824148
Computer Engineering Department, San Jose State University, CA

1.Project Overview:

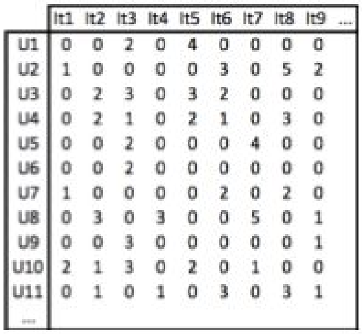
Recommendation systems play a vital role in today’s life. There is a recommendation behind every suggestion you get today on the web. One of the most famous recommendation approaches is Collaborative filtering. Collaborative filtering predicts items based on the ratings of the neighbourhood of the user.

2.Problem Statement:

This project deals with predicting the users’ rating to a certain item based on Amazon’s rating dataset. The dataset contains the actual ratings given by the amazon customers.The dataset consists of a set of users and a set of items to which the users are given ratings.The model in this project will mainly try to reduce the error between the predicted and the actual reviews.

3.Data Analysis:

The dataset used is formatted in JSON. Each entry to the dataset represents a user review to a certain item. The collaborative filtering uses item reviews, we need to build the table consisting of user reviews from json.



	It1	It2	It3	It4	It5	It6	It7	It8	It9	...
U1	0	0	2	0	4	0	0	0	0	
U2	1	0	0	0	0	3	0	5	2	
U3	0	2	3	0	3	2	0	0	0	
U4	0	2	1	0	2	1	0	3	0	
U5	0	0	2	0	0	0	4	0	0	
U6	0	0	2	0	0	0	0	0	0	
U7	1	0	0	0	0	2	0	2	0	
U8	0	3	0	3	0	0	5	0	1	
U9	0	0	3	0	0	0	0	0	1	
U10	2	1	3	0	2	0	1	0	0	
U11	0	1	0	1	0	3	0	3	1	
...										

Fig: User item ratings

The values in the table represent the rating of each item given by different users. 0 represents that the ratings are unknown.

Then the number of unique items, number of unique users and total number of ratings are considered. The next important term to be considered is the **sparsity.** The sparsity is defined as the ratio of the number of known ratings over the total number of possible ratings.

Exploratory Visualization: The distribution of ratings is analysed using the histogram and a pie chart.In our dataset, there are comparatively more ratings of 4 and 5 than lower ratings.

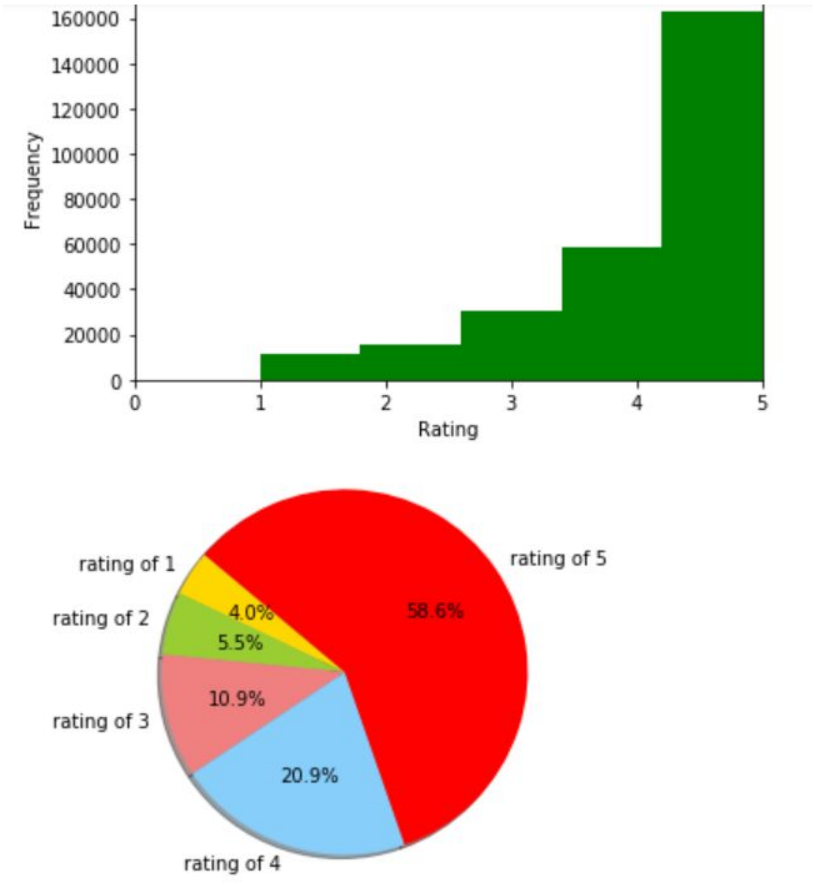


Fig: Ratings Distribution

4.Algorithms and techniques:

Singular Value Decomposition:SVD is a matrix factorization technique that is usually used to reduce the number of features of a dataset by reducing the matrix from N space to K space where K < N. For the recommendation system we just factorize the matrix keeping the dimensionality same. User-item rating matrix is used to do the matrix factorization.

Each item can be represented using a vector q . Similarly each user can be represented by a vector p such that the dot product of these 2 vectors is the expected rating. p and q can be found in such a way that the square error difference between their dot product and the known rating is minimum.

Baseline Estimate: Baseline is an algorithm in which the bias terms are added to the SVD. The Challenge in this algorithm is to obtain the terms which minimize the prediction error. So to avoid this difficulty the SVD parameters have to be adjusted.

Co-Clustering: The clustering algorithm normally puts each element in a dataset into separate clusters without duplication. An element can be only one cluster, no duplication is allowed. For example,suppose we have a dataset of fruits and we are doing co-clustering on it. We can see that beans appears in the "veggies" cluster and in the "edibles" cluster.

Co-Clustering for Collaborative filtering:

Here, we divide users into k clusters and items into m clusters. A specific user or item can be in the user cluster and the item cluster at a time.This part of the clustering is known as Co-clustering.

Prediction=(*user avg rating- avg rating of user u cluster)+(item avg rating-avg rating of item i cluster)+avg rating in the co-cluster*.

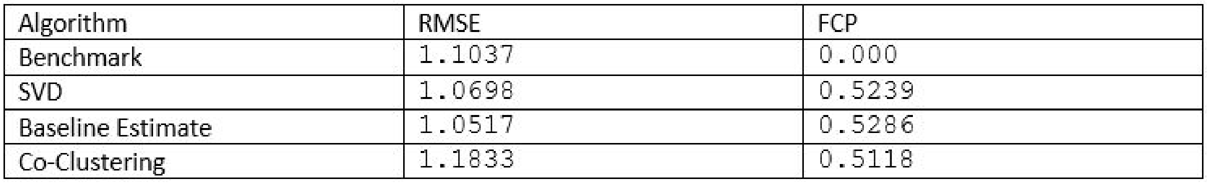
Then the clustering procedure is done by minimizing the loss function.

5.Methodology:

The first to be done is the Data preprocessing.The json file which was parsed on to the pandas dataframe is now processed to remove the unnecessary features of the collaborative filtering problem.

The next step is the implementation after the data preprocessing.The input data has to be prepared which is automatically done by surprise.Surprise automatically builds the user-item ratings matrix from a text file such as a csv, built of users, items and ratings entries. Thus it integrates easily with pandas. So to prepare the data for this library, all I had to do is to save the pandas data frame prepared in the data preprocessing phase as a csv file.

The Algorithm usage can be clearly understood using the figure below:



Algorithm	RMSE	FCP
Benchmark	1.1037	0.000
SVD	1.0698	0.5239
Baseline Estimate	1.0517	0.5286
Co-Clustering	1.1833	0.5118

The RMSE and the FCP values are computed for all the algorithms used and the following output is obtained.

Refinement: From the results in table above we can see that the best model in both RMSE and FCP measure is Baseline Estimate. On the other hand Co-Clustering is the worst where it does worse than the Benchmark. SVD results are very close to Baseline estimates, although they are little worse.Although it is counter intuitive that Baseline algorithms are doing better than other more complex algorithms, this can be explained by the fact that our user-item ratings matrix is very sparse.

After choosing the model, it has to be optimized.there are 3 factors to be varied for this algorithm which are the regularization factor, learning factor and the number of iterations.

Regularization factor effect: The RMSE and the FCP values are parallel and are similar. So as we increase the regularization factor, RMSE value increase and the FCP value reduces.

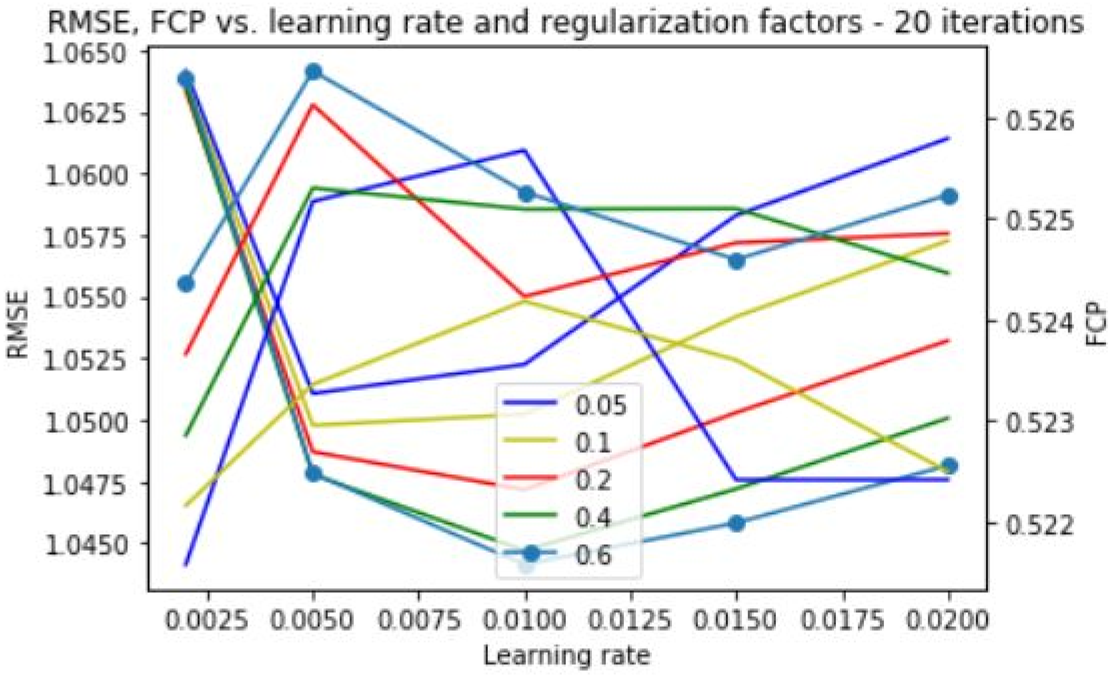
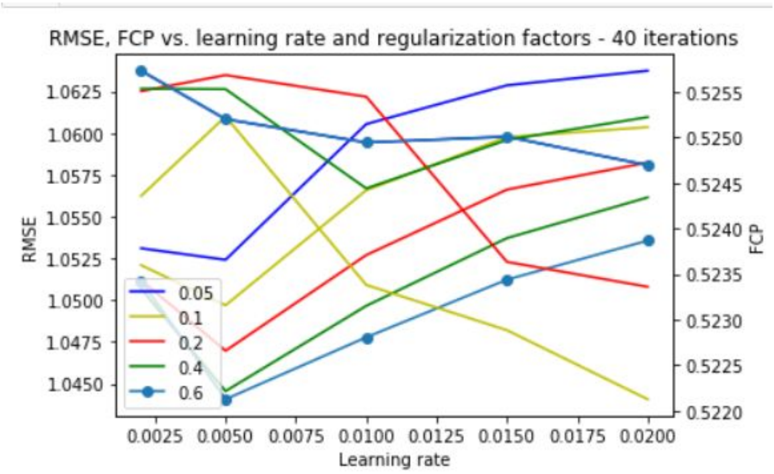


Fig:SVD parameter variations for 20 iterations.

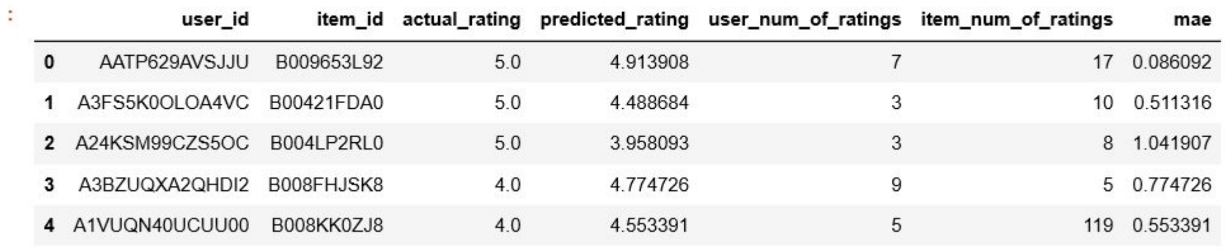
Leaning rate effect: By increasing the learning rate effect the values of RMSE and FCP improve drastically with a peak performance on FCP.

Number of iterations effect: Next we do the same measurements with 40 iterations for the stochastic gradient descent.



6.Results:

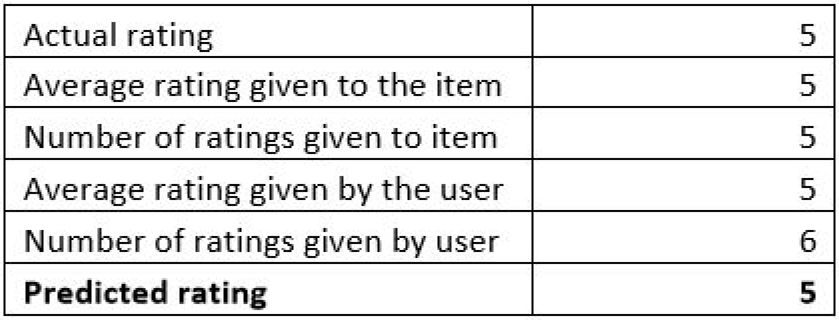
We calculate the Mean absolute error (MAE) for each entry along with the number of ratings the user gave and the number of ratings the item got in the training set. The values are obtained as follows.



	user_id	item_id	actual_rating	predicted_rating	user_num_of_ratings	item_num_of_ratings	mae
0	AATP629AVSJJU	B009653L92	5.0	4.913908	7	17	0.086092
1	A3FS5K0OLOA4VC	B00421FDA0	5.0	4.488684	3	10	0.511316
2	A24KSM99CZS5OC	B004LP2RL0	5.0	3.958093	3	8	1.041907
3	A3BZUQXA2QHDI2	B008FHJSK8	4.0	4.774726	9	5	0.774726
4	A1VUQN40UCUU00	B008KK0ZJ8	4.0	4.553391	5	119	0.553391

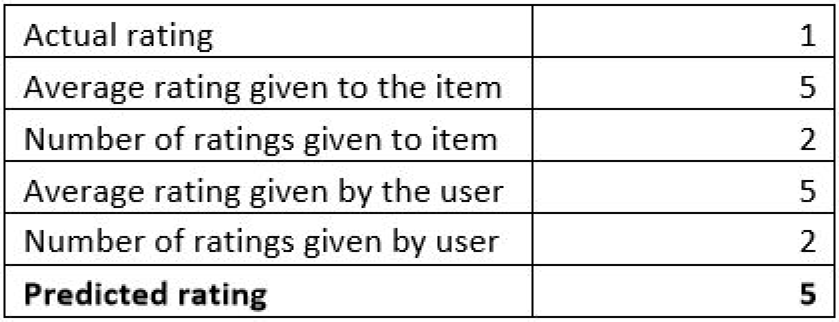
Best and worst Prediction Analysis:

The best and the worst prediction values are obtained as follows:



Actual rating	5
Average rating given to the item	5
Number of ratings given to item	5
Average rating given by the user	5
Number of ratings given by user	6
Predicted rating	**5**

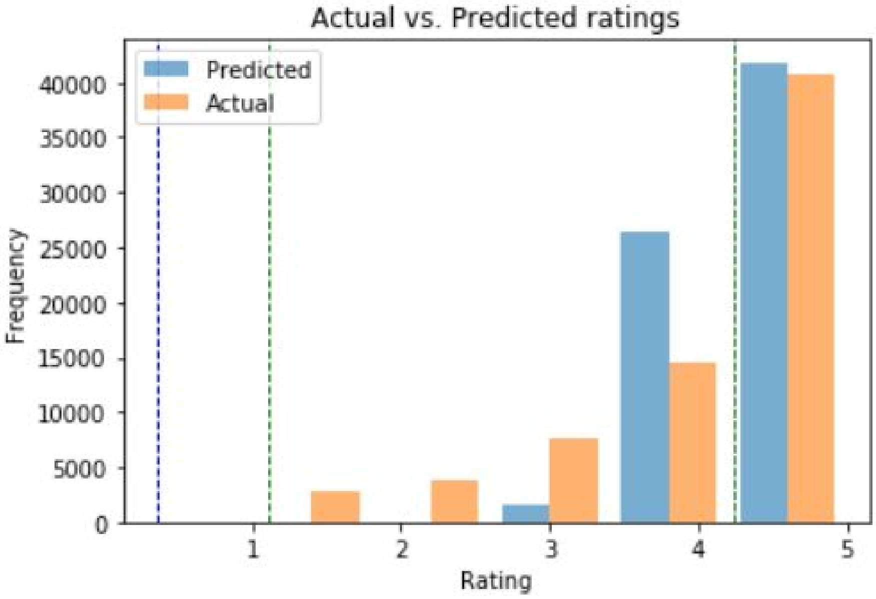
Fig: Best prediction



Actual rating	1
Average rating given to the item	5
Number of ratings given to item	2
Average rating given by the user	5
Number of ratings given by user	2
Predicted rating	**5**

Fig: Worst Prediction

Finally the plot between the actual and the predicted values is plotted as follows:


Actual vs. Predicted ratings
Predicted
Actual
Frequency
0
5000
10000
15000
20000
25000
30000
35000
40000
Rating
1
2
3
4
5