Recommendation System

Amazon Beauty Products

Purpose

To build a recommender system for Amazon Beauty Products

Data

Variables

- 'customer_id',
- 'Helpful_votes',
- 'Marketplace',
- 'Product_category’,
- 'Product_id’,
- 'product_part’,
- 'Product_title',
- 'Review_body’,
- 'Review_date',
- ‘Review_headline',
- 'review_id',
- 'star_rating’,
- 'Total_votes’,
- 'verified_purchase'

We used only the variables

1. Customer_id
2. Product_id
3. Rating

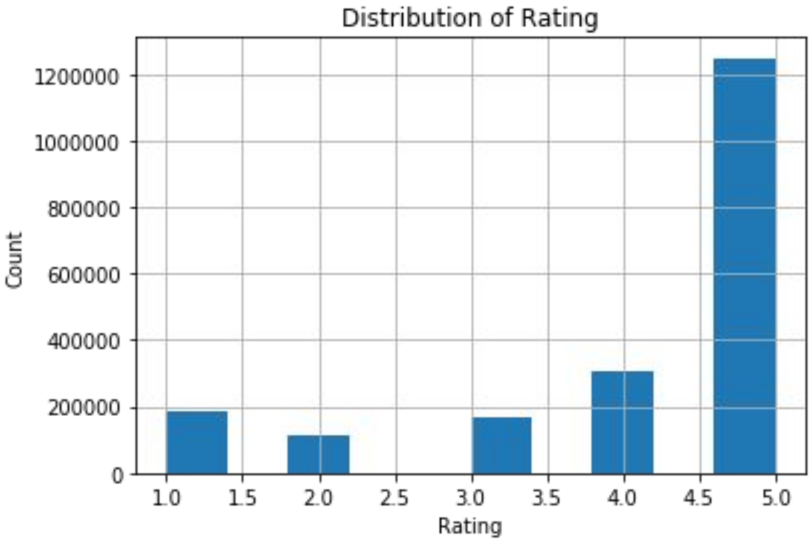
Method

There are three main types of recommenders used in practice today:

1. Content-based filter: Recommending future items to the user that have similar innate features with previously "liked" items.
2. Collaborative-based filter: Recommending products based on a similar user that has already rated the product.
3. Hybrid Method: Leverages both content & collaborative based filtering.

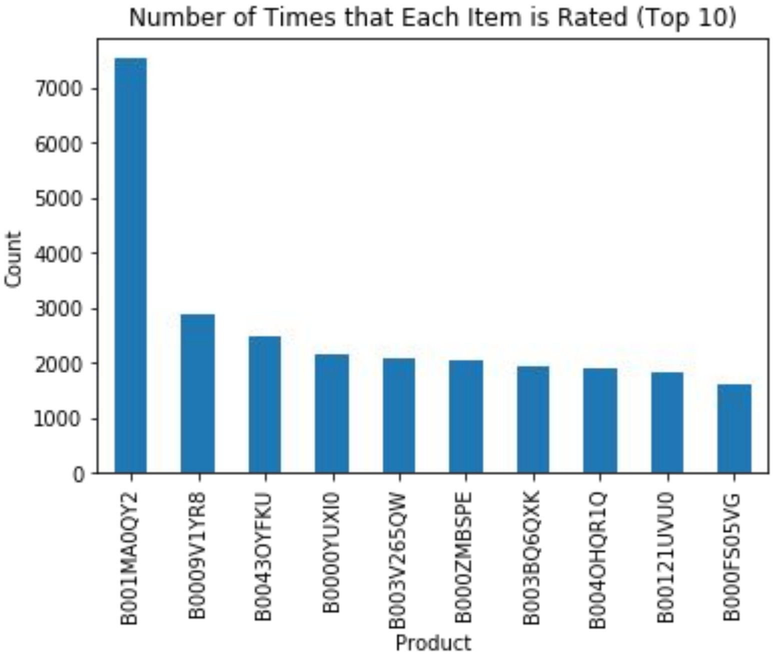
We use a user-based recommendation system that predicts the rating that a user would give to a certain product.

EDA



- Most of the rating values are above 4.
- There are almost 200,000 out of 2M items that received a rating of 1.
- And, almost 300,000 items rated as 4.
- The majority of the rating values are above 4.5

Most Popular Items



‘B001MA0QY2’

(‘Ceramic Tourmaline Ionic Flat Iron Hair Straightener’) is rated more than 7000 times with an average rating of 4.32

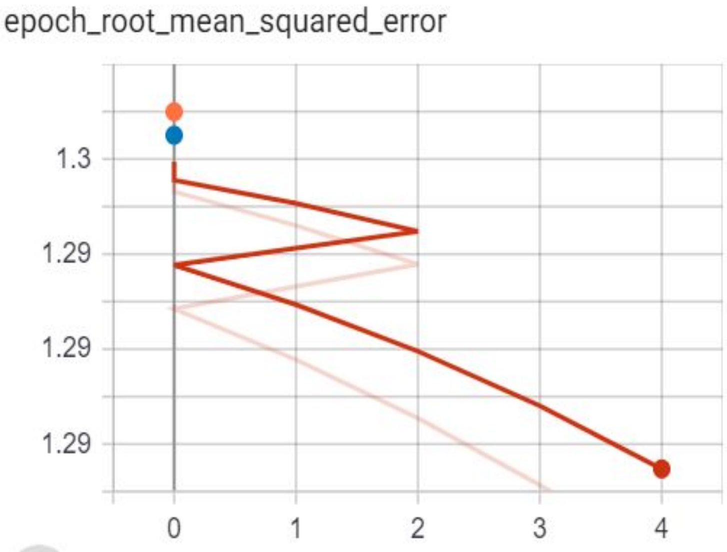
Algorithm and Machine Learning

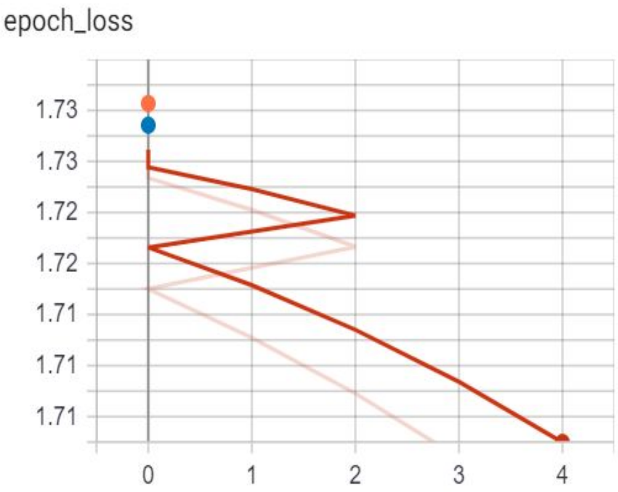
- Used Deep Learning Algorithm with Tensorflow

- Fit the model with 5 epochs

- We use the MeanSquaredError and RMSE Keras loss in order to predict the ratings.

Training Performance





As we can see, on the graphs above, as the model is trained the loss and RMSE decrease

Test Performance

root_mean_squared_error	1.296741
loss	1.672668
regularization_loss	0.000000
total_loss	1.672668

Performance Comparison - Test vs. Train

	Train	Test
Loss	1.697	1.672
RMSE	1.289	1.297

Based on the table above, the performance of the model on the training and the test data is quite similar.

Future Improvement

The model above gives us a decent start towards building a ranking system. Of course, making a practical ranking system requires much more effort. In most cases, a ranking model can be substantially improved by using more features rather than just user and candidate identifiers.