RECOMMENDATION SYSTEM USING NLP AND NN ON AMAZON PRODUCT REVIEWS

Credit: eDesk

When we see a list of movies, we like, suggested on Netflix, when our favorite apps show us all the products we like and would love to purchase, when YouTube shows you a list of videos to watch next, you wonder is it magic that these apps or websites read you so well. Well, the answer is that they actually learn your likes and dislikes. They learn people like you, people with similar taste. They then use this knowledge to suggest you with bunch of things you would like to listen to, shop or watch. These are the Recommendation systems. All the magic happening behind the scenes is done by machine learning, or more specifically, recommendation systems that use algorithms, to find similar items and similar customers, based on their behavior, and recommend items which the specific customer should like. There are different aspects the algorithm can be trained to use to build a better recommendation. The ratings, similar interest in 2 or more individuals or the review written on a purchased product. Ratings is commonly used, but I have used review given by a customer for the product and use it to get sentiment

1) DATA SOURCE

The data is collected from the following sources:

- Amazon Review Data (2018) https://nijianmo.github.io/amazon/index.html#sample-review

- Since the data is large, I will be planning to use review data, where a product has 5 or more reviews ("Small" subsets for experimentation – 5-core).
- There are two datasets to consider, one is the review and ratings information for a product by a customer, with primary key as the product ID.
- The product details is present in a meta dataset with the primary key of product ID and other product details. We will be merging these two datasets for the project.

2) DATA WRANGLING

The features which have non-null values include:

* overall: rating of the product
* Verified
* reviewTime: time of the review (raw)
* reviewerID: ID of the reviewer
* asin: ID of the product
* unixReviewTime: time of the review (Unix time)
* reviewerName: name of the reviewer
* summary: summary of the review
* reviewerText: text of the review

The features that have less than 100 null values include:

* vote: helpful votes of the review
* image: images that users post after they have received the product

The features which have considerable number of null values include:

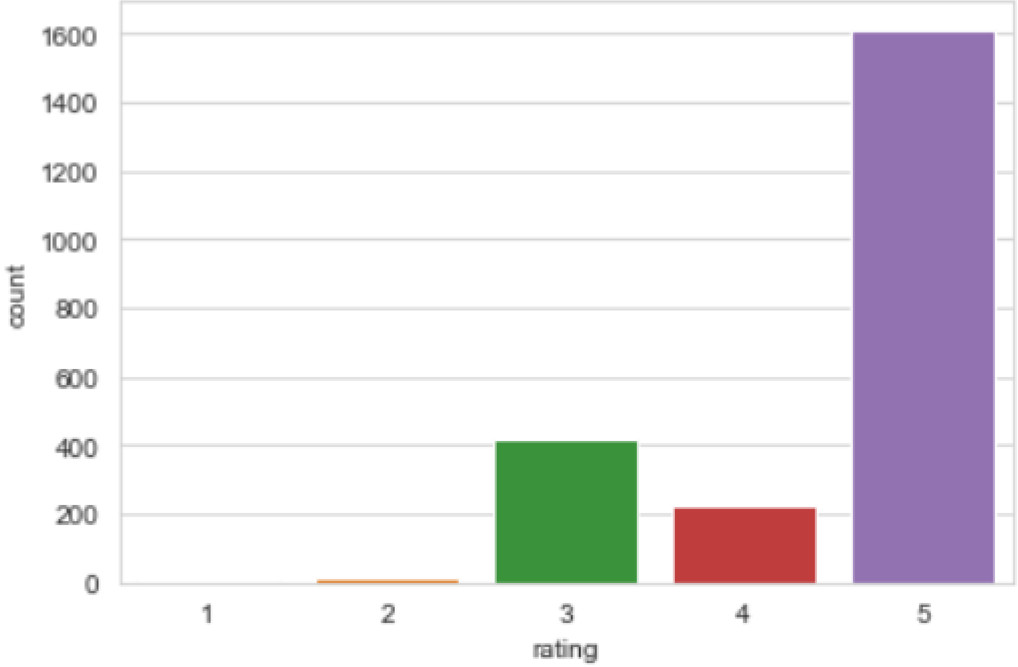
* style: a dictionary of the product metadata

We will need overall, reviewerID, asin, reviewerText further to work on our model. We can select other features like reviewerName, but we already have reviewerID, so this feature is not necessary.

- The duplicated values for a productid and customerID combination was dropped. The column values were renamed appropriately and unwanted columns were dropped.
- The meta dataset was included to have only productid, product title, brand, image URL columns. The review dataset was included to have only productid, reviewerID, reviewer name, review text, summary and rating.

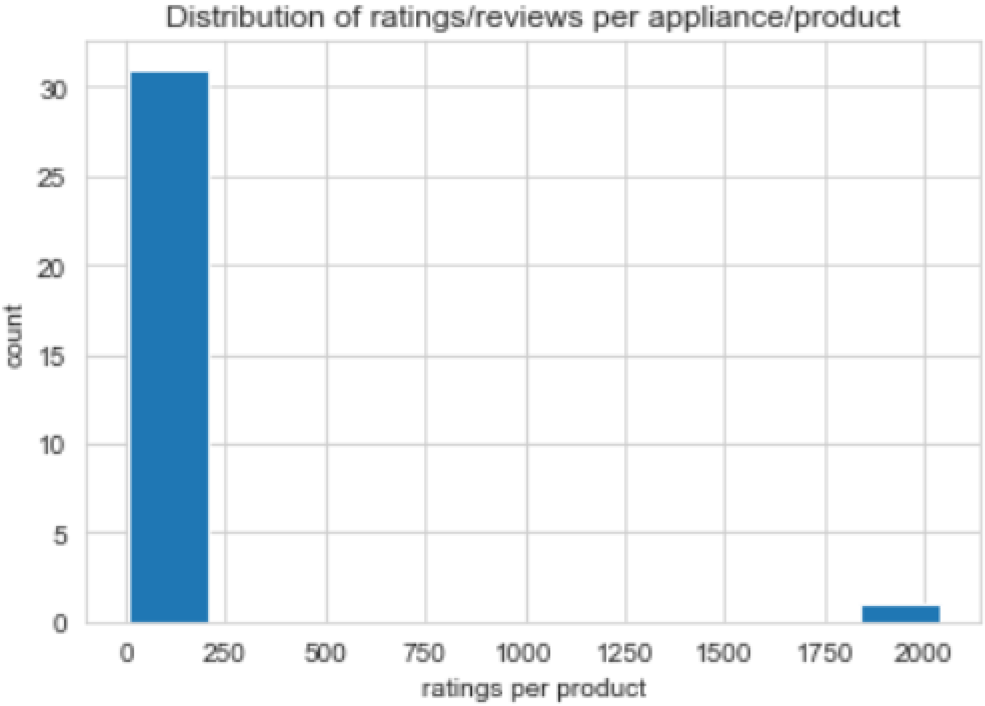
3) EXPLORATORY DATA ANALYSIS

- **RATINGS COUNT PLOT**

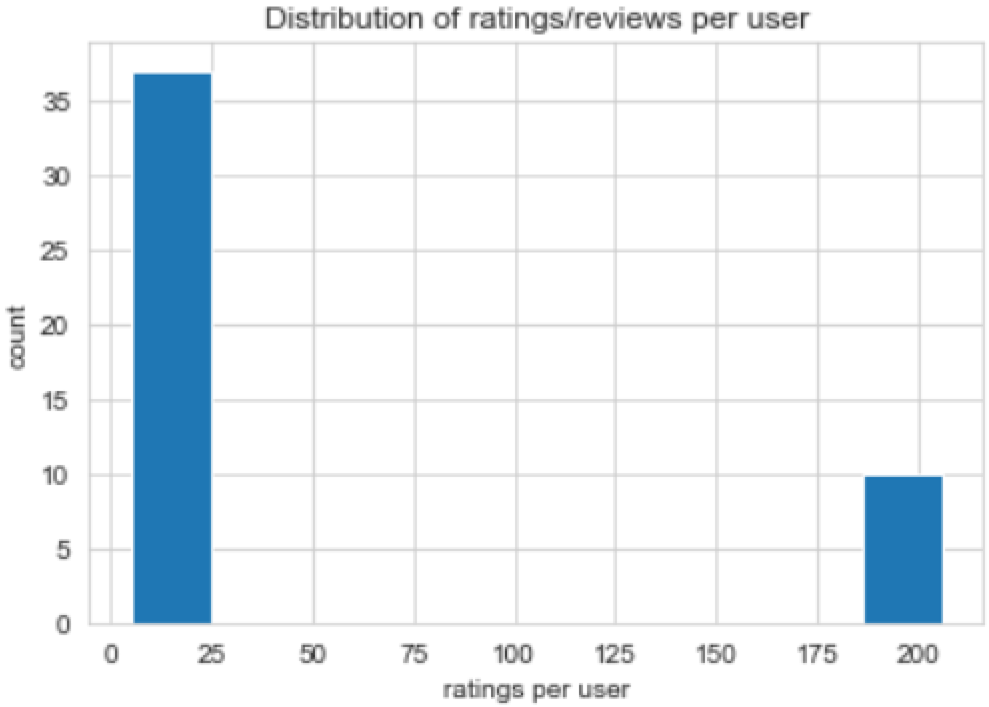


There are hardly 1- and 2-star ratings and 5 start rating make more than half the number of ratings given by the customer for amazon appliances.

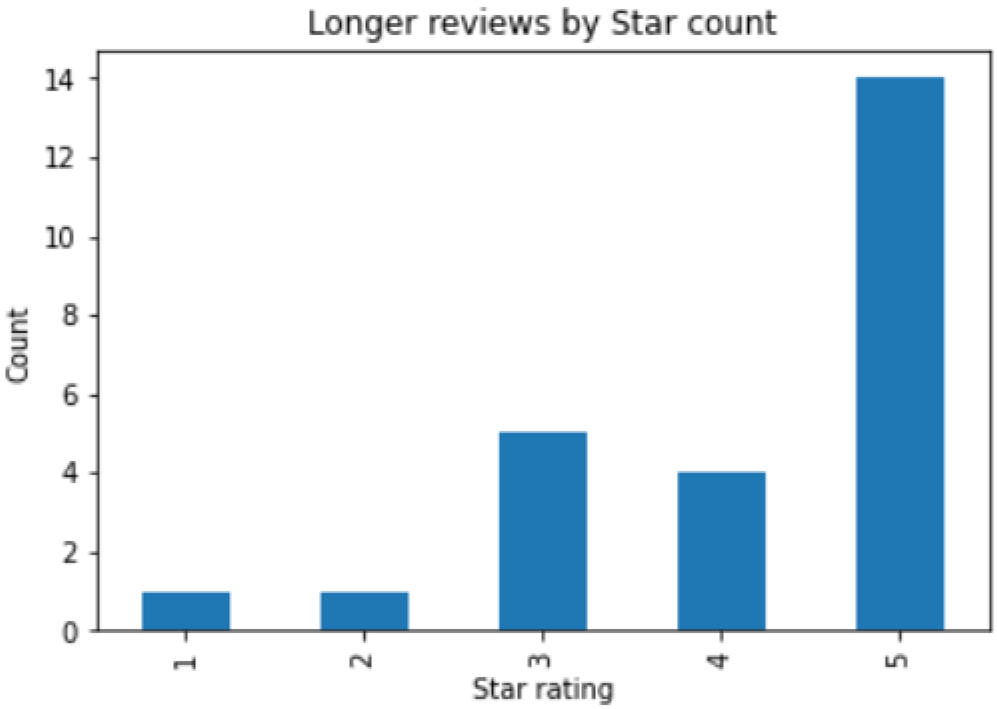
- **Distribution of reviews/ ratings**

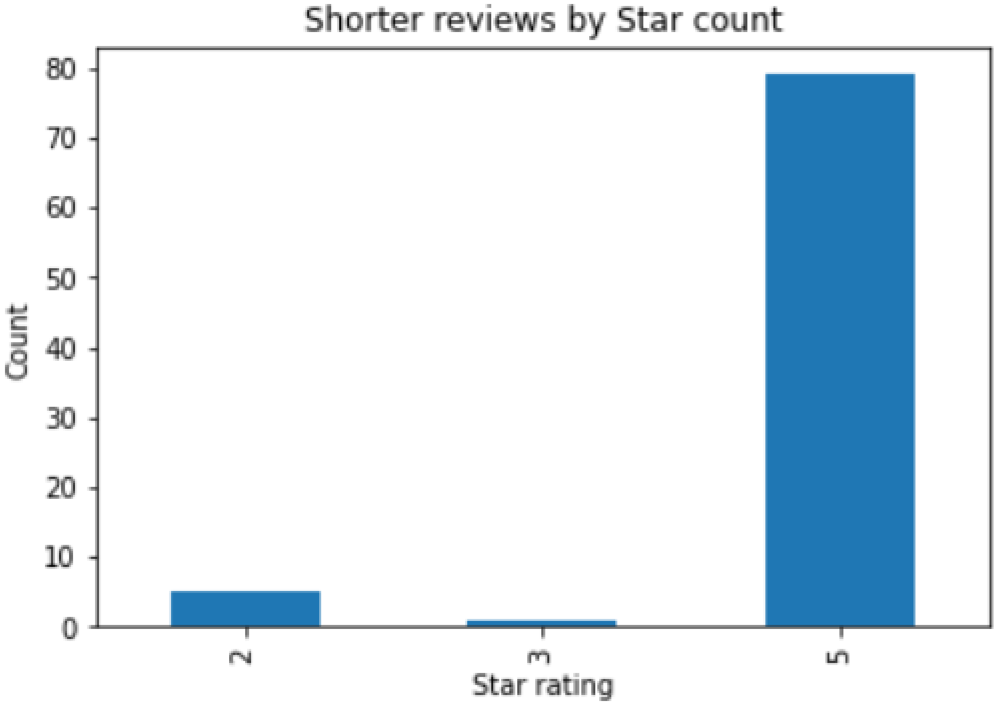


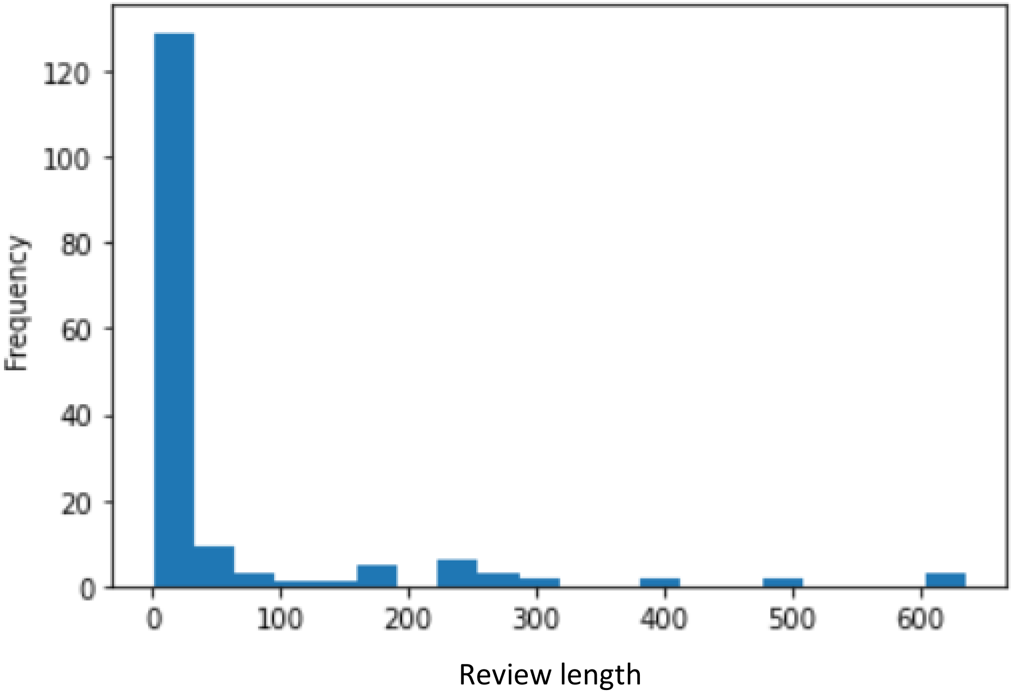
Looks like only one product has got more than 1750 number of ratings and other products have got between 0-250 number of ratings.



Around 10 users have given more than 185 number of ratings. The rest of the users have given between 0-25 ratings.
We cannot consider the skewing as outliers.

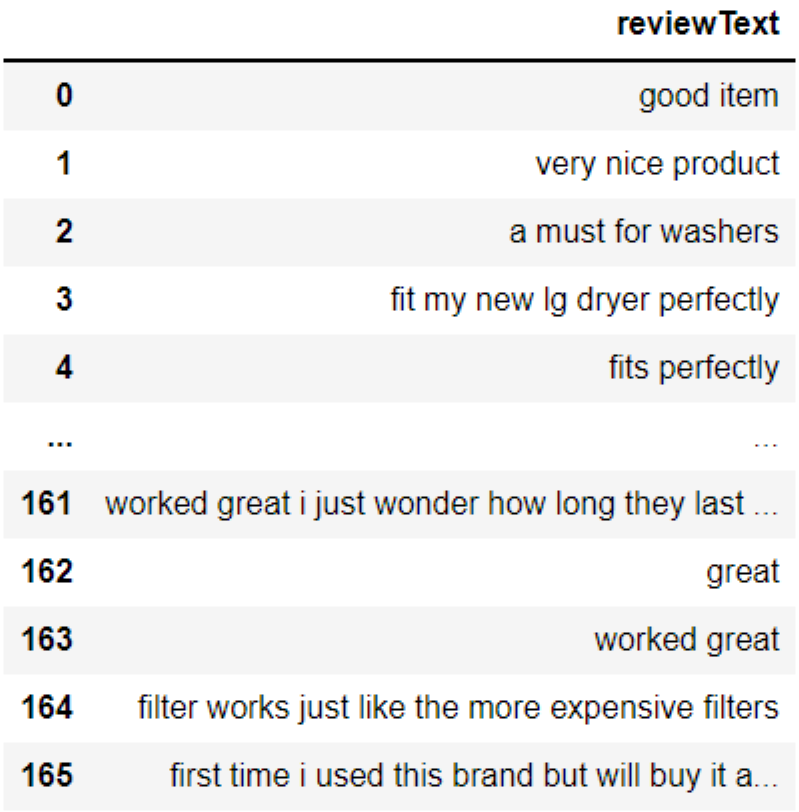

Longer reviews by Star count
Count
0
2
4
6
8
10
12
14
1
2
3
4
5
Star rating


Shorter reviews by Star count
Count
0
10
20
30
40
50
60
70
80
2
3
5
Star rating



The above graph shows that the most of the reviews are at < 50-word count. The 5-star rating being highest in both shorter and longer reviews, has a large count with less words in the review text.

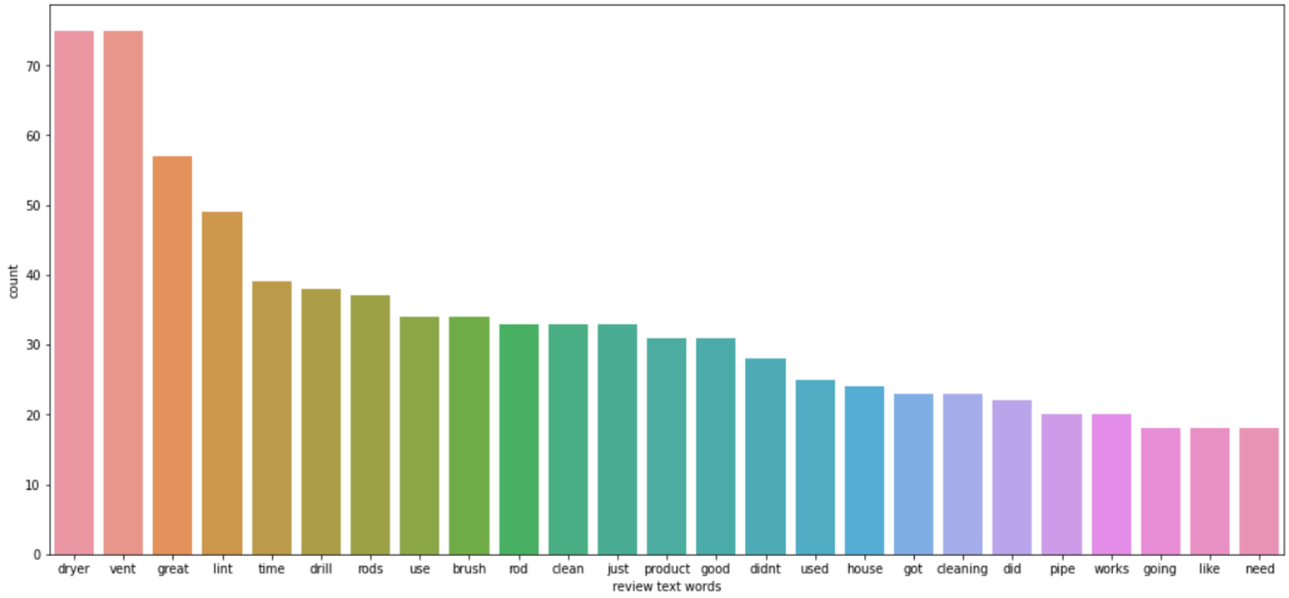
- The review text is used in sentiment analysis and hence requires cleaning. I converted text to lowercase, removed text in square brackets, removed punctuation and removed words containing numbers and new line character in text.



	reviewText
0	good item
1	very nice product
2	a must for washers
3	fit my new lg dryer perfectly
4	fits perfectly
...	...
161	worked great i just wonder how long they last ...
162	great
163	worked great
164	filter works just like the more expensive filters
165	first time i used this brand but will buy it a...

This is the review text column after cleaning.

- The words appearing in the review text with frequency of occurrence is given below



4) MACHINE LEARNING MODEL TRAINING AND PREDICTION

SENTIMENT ANALYSIS ON REVIEW TEXT COLUMN

Sentiment analysis (or opinion mining) is a natural language processing (NLP) technique used to determine whether data is positive, negative or neutral. Sentiment analysis is often performed on textual data to help businesses monitor brand and product sentiment in customer feedback, and understand customer needs.

- Since we have small amount of data, we will use all the review text data, larger length and smaller length review.

- We will analyze the review texts using Sentiment analysis. We will be using 3 different NLP sentiment analysis packages:

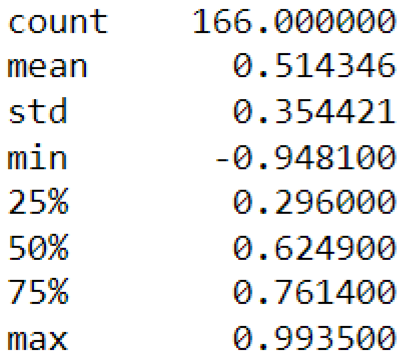
  * VADER: It uses a list of lexical features (e.g., word) which are labeled as positive or negative according to their semantic orientation to calculate the text sentiment. Vader sentiment returns the probability of a given input sentence to be: Positive, negative, and neutral.

  * TextBlob: TextBlob sentiment analyzer returns two properties for a given input sentence:
  Polarity is a float that lies between [-1,1], -1 indicates negative sentiment and +1 indicates positive sentiments.
  Subjectivity is also a float which lies in the range of [0,1]. Subjective sentences generally refer to personal opinion, emotion, or judgment.

  * Flair: Flair is a simple natural language processing (NLP) library developed and open-sourced by Zalando Research. Flair's framework builds directly on PyTorch, one of the best deep learning frameworks out there. The Zalando Research team has also released several pre-trained models for the following NLP tasks:

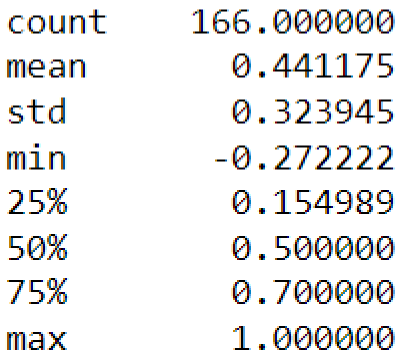
  Name-Entity Recognition (NER): It can recognize whether a word represents a person, location or names in the text.
  Parts-of-Speech Tagging (PoS): Tags all the words in the given text as to which "part of speech" they belong to.
  Text Classification: Classifying text based on the criteria (labels)
  Training Custom Models: Making our own custom models.

VADER:



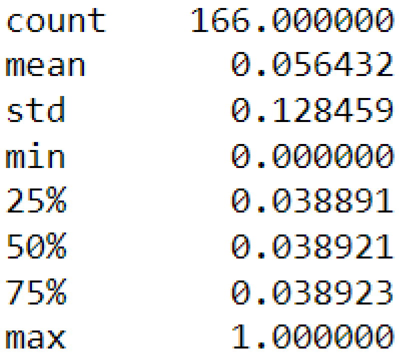
```
count    166.000000
mean       0.514346
std        0.354421
min       -0.948100
25%        0.296000
50%        0.624900
75%        0.761400
max        0.993500
```

TEXTBLOB:



```
count    166.000000
mean       0.441175
std        0.323945
min       -0.272222
25%        0.154989
50%        0.500000
75%        0.700000
max        1.000000
```

FLAIR:



```
count    166.000000
mean       0.056432
std        0.128459
min        0.000000
25%        0.038891
50%        0.038921
75%        0.038923
max        1.000000
```

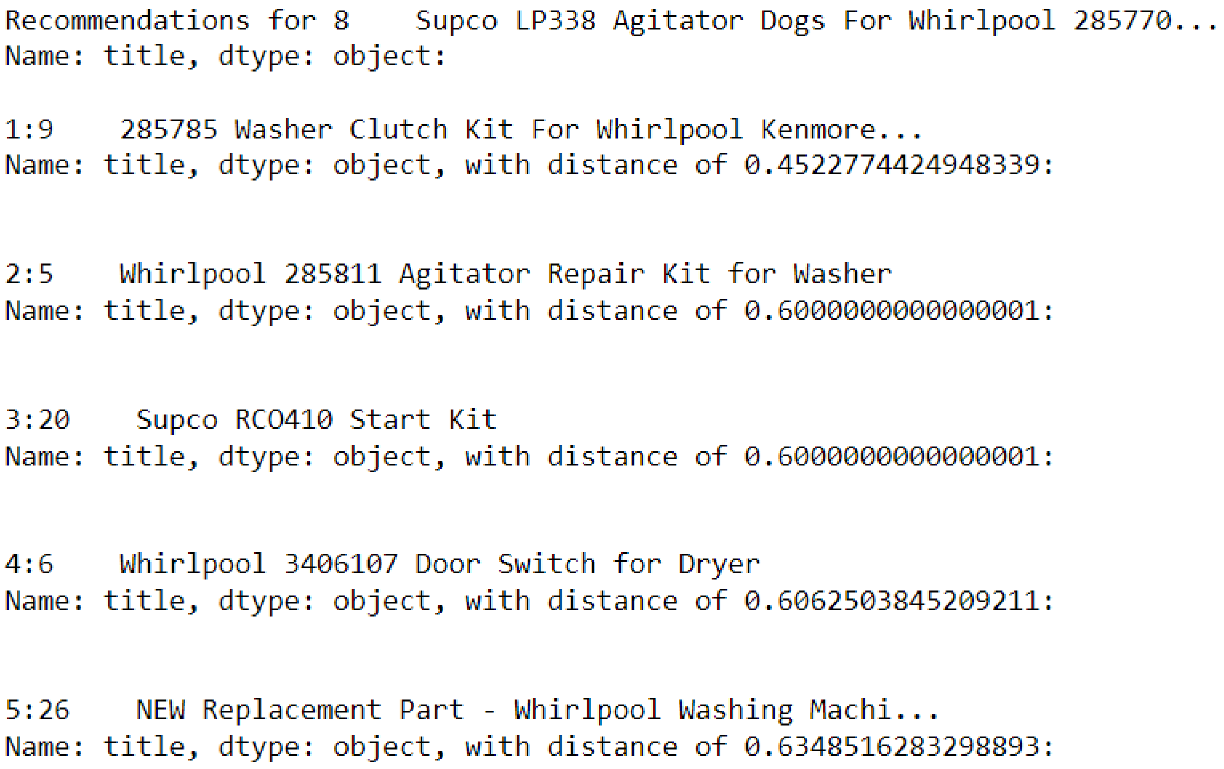
- The Vader, TextBlob and Flair scores when compared with the ratings, the Vader and TextBlob sentiment analysis looks pretty decent compared to flair sentiment analysis. The range of sentiment score for flair is large. The former 2 packages look more promising.

RECOMMENDATION USING KNN, MATRIX FACTORIZATION AND NEURAL NETWORK:

- The recommendation system usually uses the product X customer rating values to train and give out recommendations.
- I have used two scenarios, the rating given by the user and the sentiment analysis score (Using the TextBlob polarity score) to see how my recommendation system works.

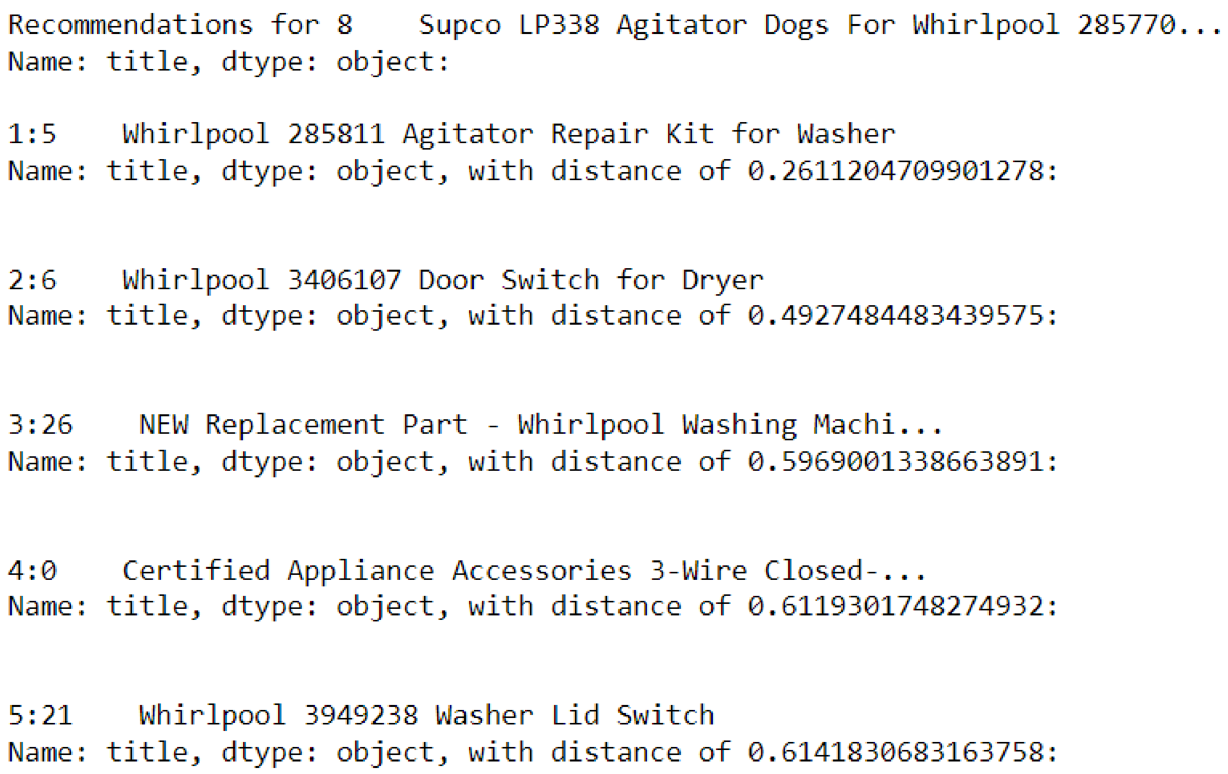
KNN using COSINE SIMILARITY: The KNN algorithm assumes that similar things exist in close proximity. In other words, similar things are near to each other. Cosine similarity measures the similarity between two vectors of an inner product space. It is measured by the cosine of the angle between two vectors and determines whether two vectors are pointing in roughly the same direction. It is often used to measure document similarity in text analysis.

Using customer ratings:



```
Recommendations for 8    Supco LP338 Agitator Dogs For Whirlpool 285770...
Name: title, dtype: object:

1:9    285785 Washer Clutch Kit For Whirlpool Kenmore...
Name: title, dtype: object, with distance of 0.4522774424948339:


2:5    Whirlpool 285811 Agitator Repair Kit for Washer
Name: title, dtype: object, with distance of 0.6000000000000001:


3:20    Supco RCO410 Start Kit
Name: title, dtype: object, with distance of 0.6000000000000001:


4:6    Whirlpool 3406107 Door Switch for Dryer
Name: title, dtype: object, with distance of 0.6062503845209211:


5:26    NEW Replacement Part - Whirlpool Washing Machi...
Name: title, dtype: object, with distance of 0.6348516283298893:
```

Using TextBlob polarity score:



```
Recommendations for 8    Supco LP338 Agitator Dogs For Whirlpool 285770...
Name: title, dtype: object:

1:5    Whirlpool 285811 Agitator Repair Kit for Washer
Name: title, dtype: object, with distance of 0.2611204709901278:


2:6    Whirlpool 3406107 Door Switch for Dryer
Name: title, dtype: object, with distance of 0.4927484483439575:


3:26    NEW Replacement Part - Whirlpool Washing Machi...
Name: title, dtype: object, with distance of 0.5969001338663891:


4:0    Certified Appliance Accessories 3-Wire Closed-...
Name: title, dtype: object, with distance of 0.6119301748274932:


5:21    Whirlpool 3949238 Washer Lid Switch
Name: title, dtype: object, with distance of 0.6141830683163758:
```

- The recommendations using the KNN looks good. We can see that the product recommendations have been given from the company mostly (Whirlpool). Let us also try Matrix factorization and see how the model works.

MATRIX FACTORIZATION AND NEURAL NETWORK: Matrix factorization is a simple embedding model. Given the feedback matrix $A \in R^{m \times n}$, where m is the number of users (or queries) and n is the number of items, the model learns:

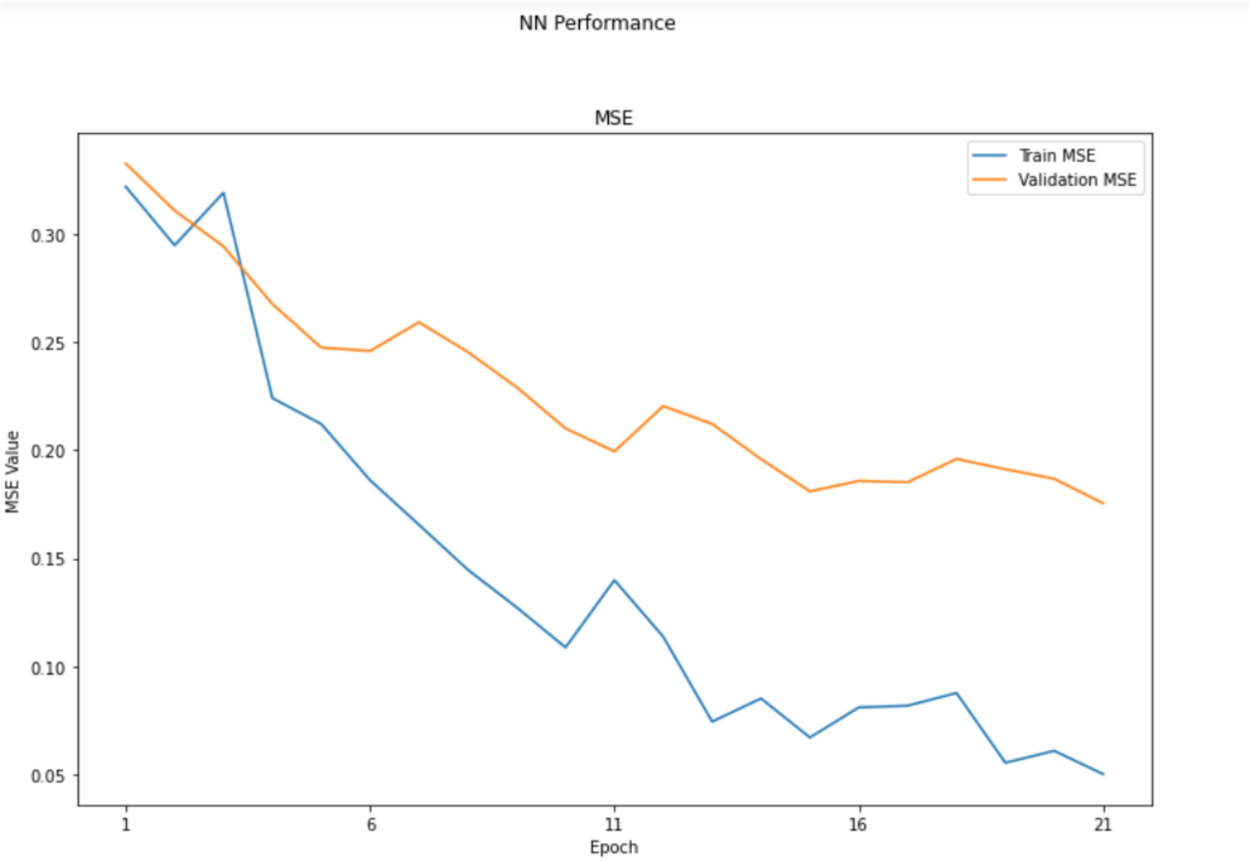
A user embedding matrix $U \in \mathbb{R}^{m \times d}$, where row i is the embedding for user i.

An item embedding matrix $V \in \mathbb{R}^{n \times d}$, where row j is the embedding for item j.

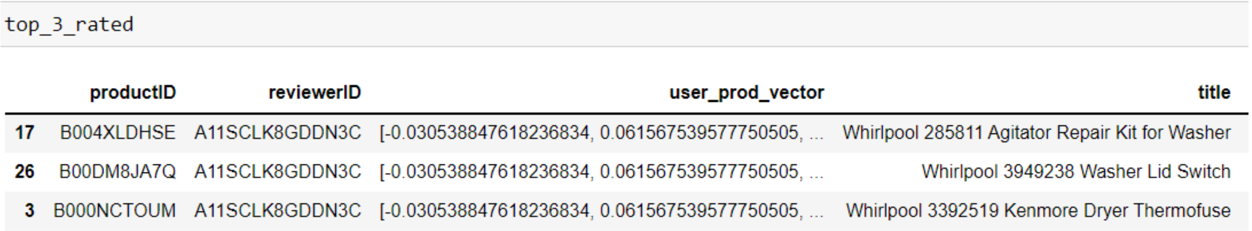
The embeddings are learned such that the product UV^T is a good approximation of the feedback matrix A. Observe that the (i,j) entry of $U.V^T$ is simply the dot product $\langle U_i, V_j \rangle$ of the embeddings of user i and item j, which you want to be close to $A_{i,j}$.

Artificial neural networks, usually simply called neural networks, are computing systems inspired by the biological neural networks that constitute animal brains. An ANN is based on a collection of connected units or nodes called artificial neurons, which loosely model the neurons in a biological brain.

MODEL	**MSE for Vader polarity and predicted rating**	**MSE for TextBlob compound and predicted rating**
MATRIX FACTORIZATION	0.33886	0.28554
NEURAL NETWORK		0.08104

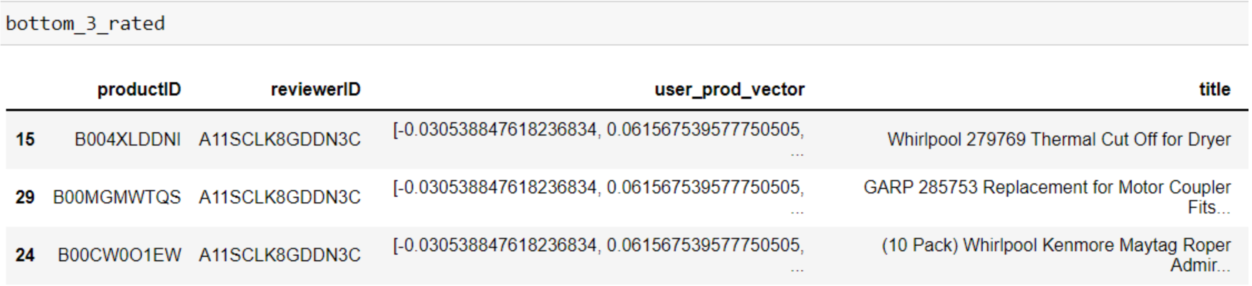


- The short model with 3 layers, gives us a better MSE compared to baseline model with 5 dense layers for neural network model. This can also be seen in the above NN performance graph. The base line model seemed to overfit even when dropout layers were added. Also, when compared to Matrix factorization, the MSE for Neural network model is small.
- KNN and NN model can be used to predict the recommendation of products.
- The predictions with the neural network model are shown below:



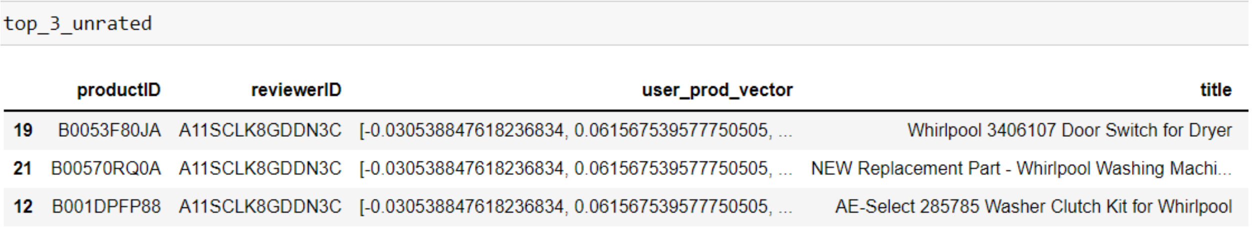
```
top_3_rated
```

	productID	reviewerID	user_prod_vector	title
17	B004XLDHSE	A11SCLK8GDDN3C	[-0.030538847618236834, 0.061567539577750505, ...	Whirlpool 285811 Agitator Repair Kit for Washer
26	B00DM8JA7Q	A11SCLK8GDDN3C	[-0.030538847618236834, 0.061567539577750505, ...	Whirlpool 3949238 Washer Lid Switch
3	B000NCTOUM	A11SCLK8GDDN3C	[-0.030538847618236834, 0.061567539577750505, ...	Whirlpool 3392519 Kenmore Dryer Thermofuse



```
bottom_3_rated
```

	productID	reviewerID	user_prod_vector	title
15	B004XLDDNI	A11SCLK8GDDN3C	[-0.030538847618236834, 0.061567539577750505, ...	Whirlpool 279769 Thermal Cut Off for Dryer
29	B00MGMWTQS	A11SCLK8GDDN3C	[-0.030538847618236834, 0.061567539577750505, ...	GARP 285753 Replacement for Motor Coupler Fits...
24	B00CW0O1EW	A11SCLK8GDDN3C	[-0.030538847618236834, 0.061567539577750505, ...	(10 Pack) Whirlpool Kenmore Maytag Roper Admir...



```
top_3_unrated
```

	productID	reviewerID	user_prod_vector	title
19	B0053F80JA	A11SCLK8GDDN3C	[-0.030538847618236834, 0.061567539577750505, ...	Whirlpool 3406107 Door Switch for Dryer
21	B00570RQ0A	A11SCLK8GDDN3C	[-0.030538847618236834, 0.061567539577750505, ...	NEW Replacement Part - Whirlpool Washing Machi...
12	B001DPFP88	A11SCLK8GDDN3C	[-0.030538847618236834, 0.061567539577750505, ...	AE-Select 285785 Washer Clutch Kit for Whirlpool

- The top 3 rated gives the highest 3 ratings given by the reviewer “**mrclobhead**”, bottom 3 rated is the lowest 3 ratings given and the top 3 unrated are the recommendations.
- Looks like the model has done pretty well with the predictions.

5) LEARNINGS AND FUTURE WORK

- The neural network model would work better with big data. Even adding a dropout layer, did not let us recommend products perfectly. I would like to work with a bigger amazon review dataset.
- The review text is a combination of satisfaction with the product, deliver, condition of the product. I would like to see these impact on the ratings given for the product.