Kenneth Hua

Module 5: Natural Language Processing and Unsupervised Learning

Abstract

Recommendation systems are useful tools that businesses are employing to help match customers with products they are likely to engage with. These systems, when developed properly, are extremely powerful and directly improve a company's ability to engage users. For my project, I will build a book recommender system using Amazon review data. The recommender system will be a collaborative filtering model that uses users sentiments to recommend books to other users with similar sentiments. Through this model, we will be able to predict whether a user will like a book that they have not read, and be able to recommend them books that they are likely to enjoy. The sentiments will be measured using sentiment analysis and natural language processing techniques.

Design and Results

The design of the project consists of 3 main steps:

1) Data Loading/Preprocessing
2) Sentiment Analysis of Reviews
3) Collaborative Filter Modeling

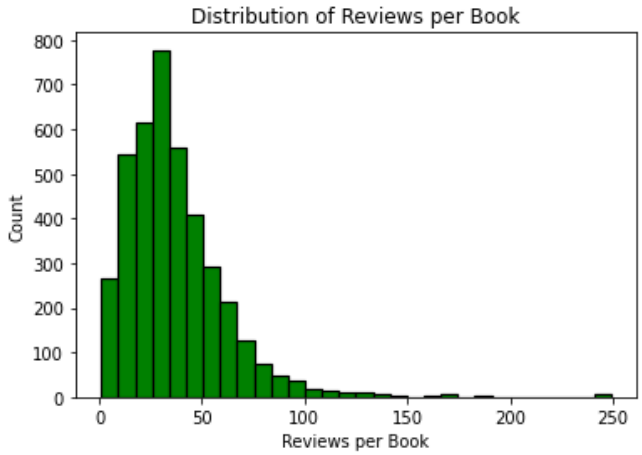
Data Loading/Preprocessing

For my project, I am beginning with Amazon review data, which can be downloaded from the link here. This is an extremely large dataset, so I am choosing only the "Books category" for my analysis. This is still a large dataset, so I will need to load the reviews in with chunks.

Since we are building a recommendation system, the most important columns that need to be extracted are 'reviewerID' (User-ID), 'asin' (Book-ID), and reviewerText (Comment Review). Other columns were explored during EDA, but not used downstream for model building.

The dataset was filtered down to Books that had 100 or reviews in the original dataset and Users who had 100 or more reviews in the original dataset. In the end, there were 5893 unique users and 4049 unique books that was made available for modelling. From this table, there were 149,580 unique user-book combinations with reviews that were made available for training and validation.

Here are some summary statistics of the filtered Books and Users in our study:
Books:

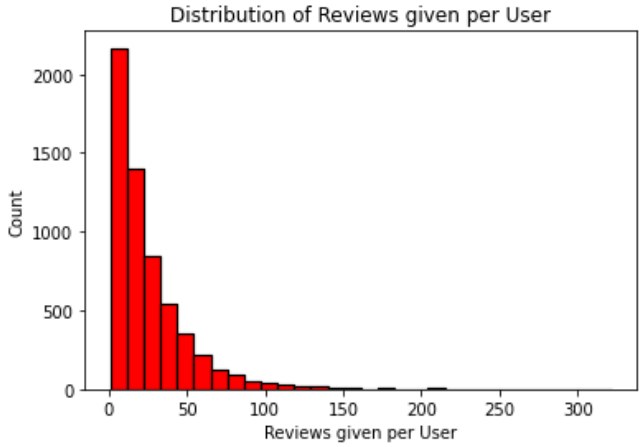


Count	4049
Mean	36.94
Median	32
St.Dev	25.35

Most Reviewed Books:

:viewCou	title
249	Divergent
248	Hobbit
248	Hobbit and Lord of the Rings Trilogy - Boxed Set of 4 Books
248	THE LORD OF THE RINGS Trilogy - (The Fellow…
248	Fellowship of the Ring (Lord of the Rings Part 1)
245	The Hobbit
241	Gone Girl
213	Where the Red Fern Grows
199	The Girl on the Train
185	Harry Potter and the Deathly Hallows, Book 7
185	Harry Potter and the Chamber of Secrets, Book 2
185	The Martian
177	The Girl with the Dragon Tattoo
173	A Tale of Two Cities (Collins Classics)
173	Oliver Twist (Penguin Clothbound Classics)

<u>Users:</u>



Count	5942
Mean	25.17
Median	17
St.Dev	26.29

Metadata on all books was also loaded in order to obtain the title names of each book.

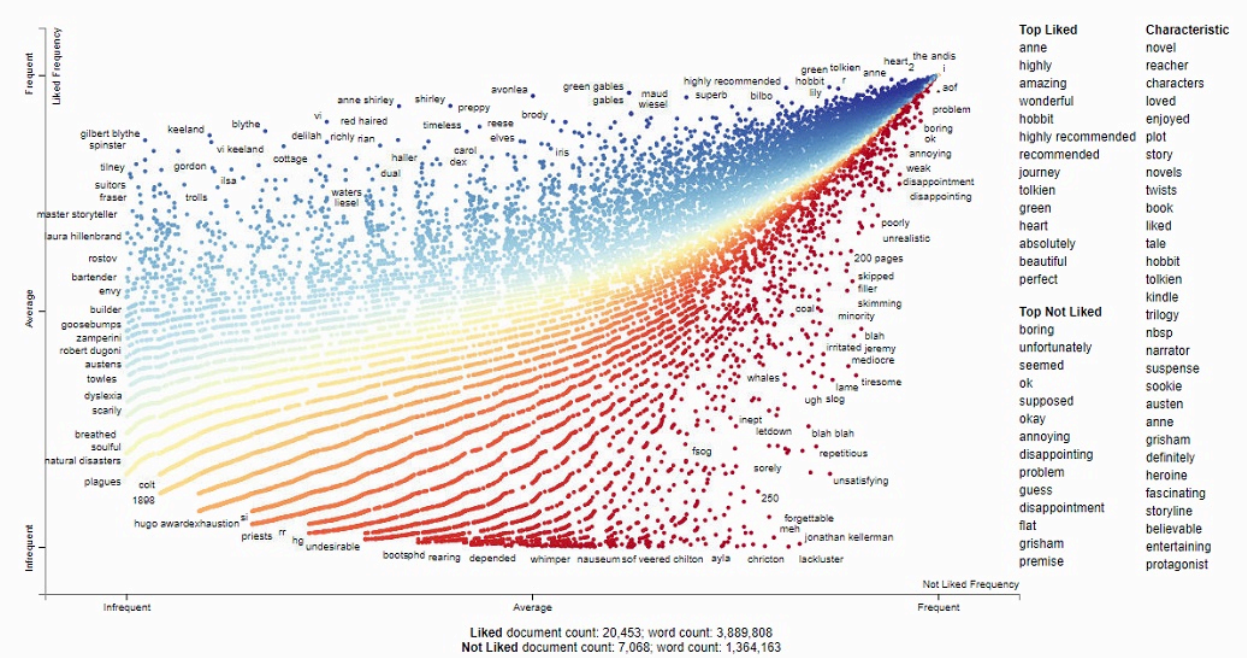
Sentiment Analysis of Reviews

After the data has been properly loaded and processed, the next step is to sentiment analysis of reviews. We can also make a ScatterText plot of words in positive sentiment comments vs negative sentiment comments to visualize if our sentiment analysis makes sense to us.

For sentiment analysis, we tested three different sentiment analysis models: Vader, TextBlob and Flair. Vader and TextBlob are lexicon and rule based sentiment analysis tools that are effective for social media platforms. However, for our data, these tools were not the most effective, which could be attributed to the fact that many reviews are of varying lengths. In many cases, the neutral scores dominate.

Flair, which uses a pre-trained neural network model to classify sentiment, worked much better, so that is the Sentiment Analysis Model we went with. The classification probabilities were transformed using QuantileTransformation, to be appropriately scaled for Collaborative Filtering.

To check to see if Flair worked effectively, it's good to use ScatterText to split the reviews into "Liked Reviews" and "Disliked Reviews". The Scattertext summary makes sense, with negative words being associated with the "Disliked Reviews" and generally positive terms with the "Liked Reviews".



Collaborative Filtering

The last step to building a recommendation system using Amazon review data is to build out the model. For this project, we built three models of increasing complexity.

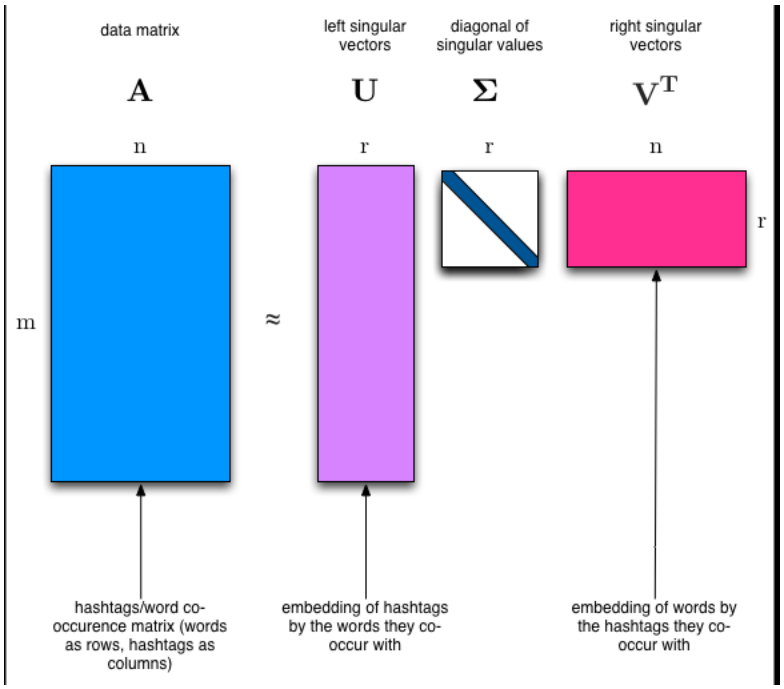
1) kNN
2) Matrix Factorization
3) Trained Neural Network

The kNN approach is an item-based approach that finds the closest neighbors to a particular book. This approach is okay, but can suffer from some dimensionality

An example from this is from below:

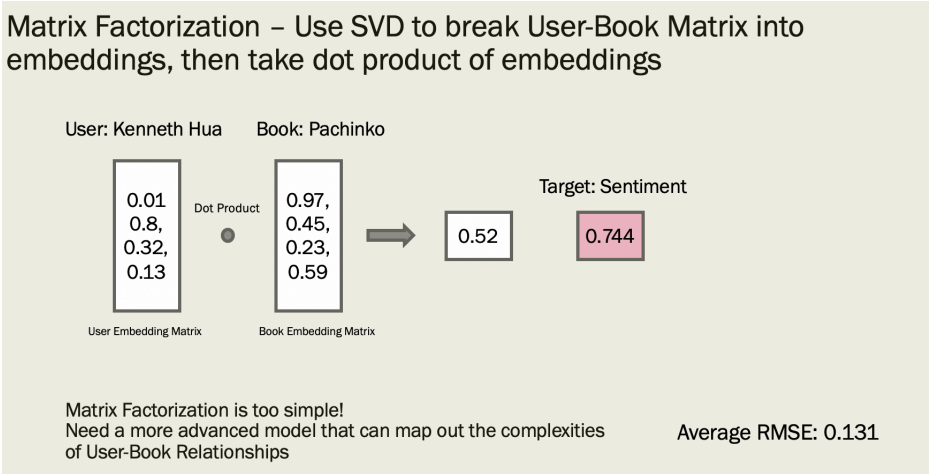
The matrix factorization approach involves dimensionality reduction of the Sparse User-Book-Matrix into SVD embedding vectors. So each User in the matrix will have a 16 value vector and each Book will have a 16 value vector.

The SVD breaks the User-Book-Matrix into the Following format. This reduces the dimensionality of the problem and captures the variation of the sparse matrix in encoded embeddings.



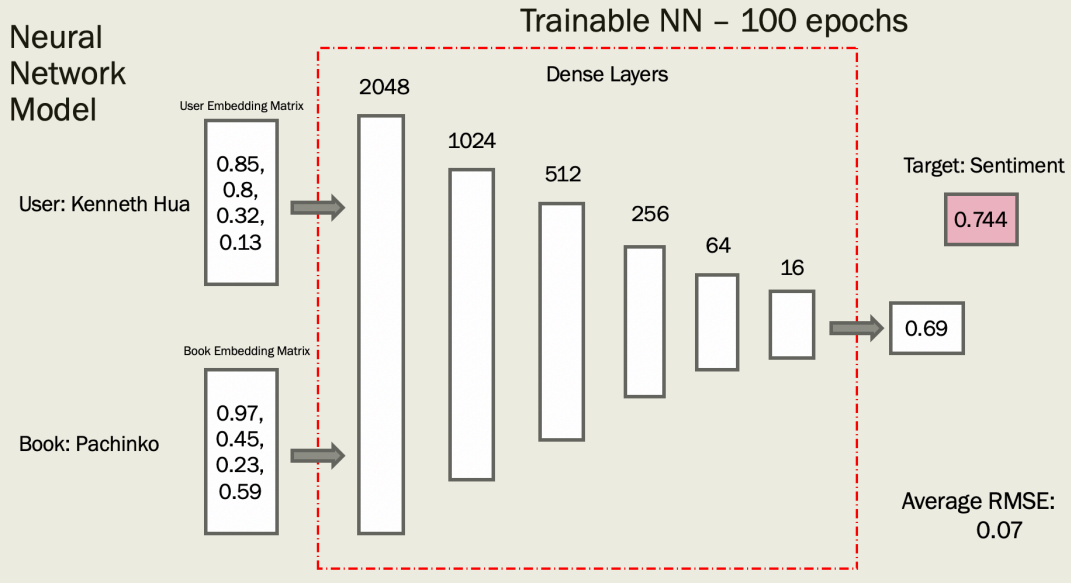
By multiplying the reduced vectors together (alongwide the diagonal of singular values), we can obtain the prediction of each user's predicted sentiment of each book.

A diagram of Matrix Factorization for this dataset is demonstrated in the graphic below:



While, this is straightforward to understand, the loss of this method is quite high. The prediction doesn't work the best on unknown user-book pairs. In order to improve this, it's better to use a trainable neural network that can be more flexible in adapting the embeddings to the target value.

Our baseline model will have the following model architecture:



A few different neural network models were tried out, as described below:

1. Baseline Model: Dense Layers of 2048, 1024, 512, 256, 64, 16, 1
2. Short Model: Dense Layers of 256, 64, 16, 1
3. Wide, deep Model: Dense Layers of 8192, 4096, 2048, 1024, 512, 64, 16, 1

The batch_size used is, and the epochs is 100. The validation MSE is shown below, with comparison to Matrix Factorization Dot Product.

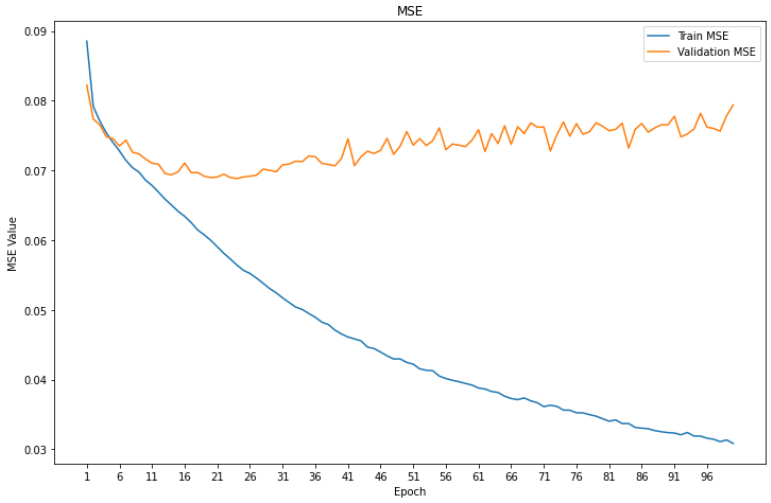
Model Comparison:

Model	Mean Squared Error
Matrix Factorization Dot Product	0.13105
NN: Baseline Model	0.070343
NN: Short Model	0.070548
NN: Wide, Deep Model	0.071473

Seems like the Baseline model performs the best by a slim margin, so we choose this to work out our examples.

The Training and Validation Loss of our Baseline Model can be seen as below:

CNN Performance



From this figure, we can see a few things:

- Validation MSE is optimal around epoch 25
- Training loss continues to decease, some overfitting occurs.
- Model can be improved with more data, hyperparameter tuning

Now, let's move on to making somoe predictions using some example users.

Example:

Example 1:

User-ID: A2CWQW5MCYL10J
User-Name: Craig Wood
Books read in the dataset: 53

Top 5 Rated Books:	Review Sentiment
MiddleSex: A Novel	1.0
The Things They Carried (Contemperary American Fiction)	0.94
Bury My Heart At Wounded Knee: An Indian History of the American West	0.94
How the Light Gets In: A Chief Inspector Gamache Novel	0.88
Fire and Fury: Inside the Trump White House	0.88

Top 5 Recommendations:	Predicted Sentiment
The Known World	0.97
Interpreter of Maladies	0.96

The PoisonWood Bible	0.94
Gates of Fire	0.93
Daniel Silva Thriller 6	0.92

This User Craig Wood, really enjoyed books related to History, War, Politics, and long winding historical novels. The recommendations are books related to missionaries, colonisms, culture, and history. The recommendation seems to make sense, but whether Craig will like them is up to himself! To evaluate how the model performs, we will need to evaluate sales over time while controlling for other factors.

Example 2:

User-ID: A2CWQW5MCYL103
User-Name: nobinzinfla
Books read in the dataset: 41

Top 5 Rated Books:	Review Sentiment
The Kill Artist	1.0
Mystic River	1.0
Timeline	1.0
The Closers	1.0
The Day of the Jackal	0.94

Top 5 Recommendations:	Predicted Sentiment
In Cold Blood	1.04
The Grapes of Wrath	1.00
Echo Park	1.00
The Brass Verdict: A Novel	0.98
Empire Falls	0.97

For this user nobbinzinfla, they really like Mystery, Detective and Thriller books. So the topo 5 recommendations returned similar type mystery books. Including two additional books by author Michael Connelly. They also returned The Grapes of Wrath and Empire Falls, which are two books that are more about Small Town America than thriller, but perhaps the user will enjoy branching out to that genre!

Conclusion and Next Steps

Conclusion:

We have successfully built an end to end recommendation system that is capable of handling large datasets. We have also demonstrated that it can make useful predictions that have the

potential to improve user sales once implemented in the field. This was done using a pipeline of:

- Data Loading/Processing
- Sentiment Analysis
- Collaborative Filtering with Neural Networks

Next Steps:

- The next steps are to implement this model in production. This is the true to see if this model can recommend books to users in a way that leads to an increase in sales, which is the ultimate goal of the project.
- Once it's implemented, ongoing evaluation is needed to understand the performance of the model
- If the model is successful, we then want to continuously train the model on new data, so we need to adapt the model to be able to accommodate new data in a controlled way
- If the model does not meet our expectation, we can then go back to redesign the Neural Network Architecture and adjust the process for data cleaning and preparation.

Tools

Pandas: data loading, cleaning and processing
Sklearn: Countvectorization, Matrix factorization, kNN, Data processing, Clustering
Nltk/Vader/Flair: Sentiment Analysis and Tokenization
Scattertext: Natural Language Visualization
Tensorflow: NN/Algorithm for Collaborative Filtering

Communication

For additional information, please contact kenhua15@gmail.com. The project will also be posted on my github found here: https://github.com/kenhua15/Metis-Projects