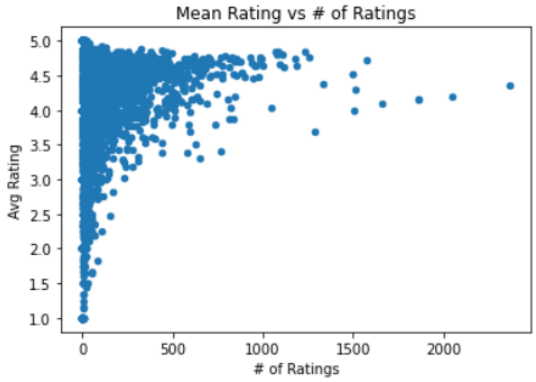
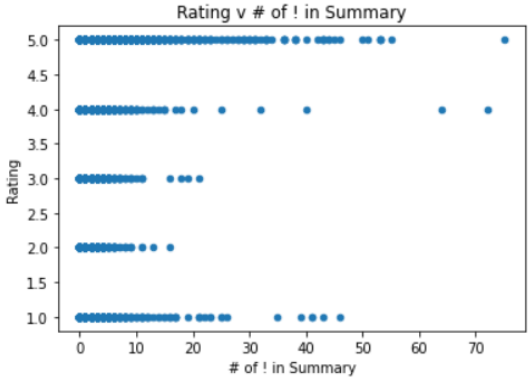
CSE 158 Assignment 2

1. Our project's dataset consists of Amazon product reviews collected in 2018. Specifically, products in the “Movies and TV” category of Amazon. We will be using the first 500,000 rows of the data set. The dataset's features include the overall score (rating which the user gave between 1.0 to 5.0), whether the user is verified, the time the review was posted, the reviewer id, the item id, the type of product, the reviewer name, the text of the review, the summary of the review, the review time in Unix form, and the number of upvotes for the review. We first experimented with popularity, determining if there was a correlation between popular items and a high average rating for an item (avg rating is calculated by adding up all the ratings which users gave for a specific item and dividing that by the number of reviews for that specific item).

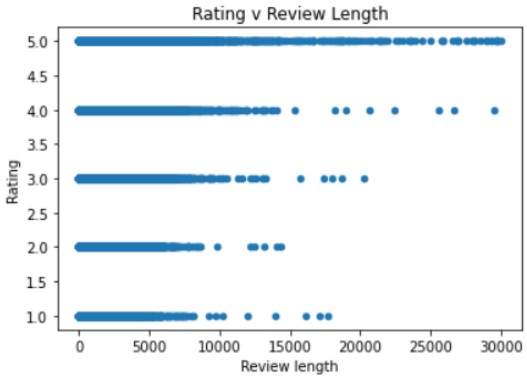


Looking at the scatterplot, we did not see a strong correlation between the two variables. However, we identified that almost all items with 1000+ reviews had at least an average rating of 3.0 or above.

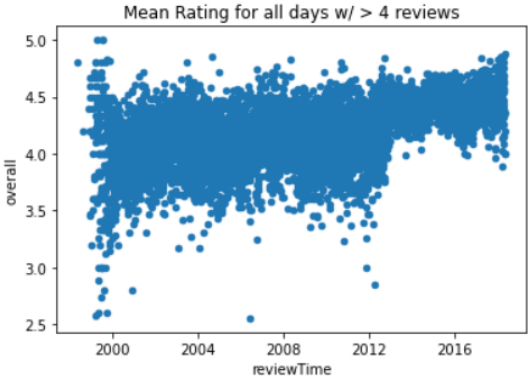
We also experimented with the number of exclamation marks, trying to identify a correlation between increased exclamation marks and ratings given.



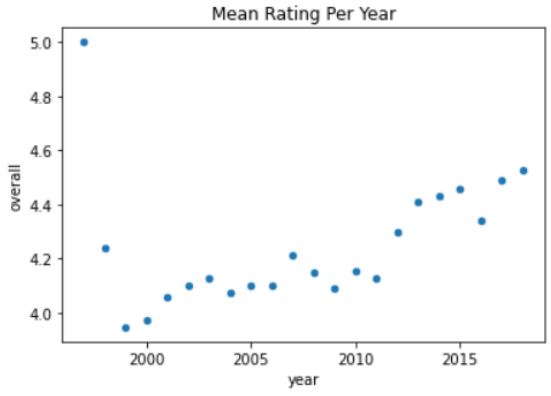
Another variable we looked into was the length of the review text. We made a scatter plot plotting the relationship between the review text's length and the rating given.

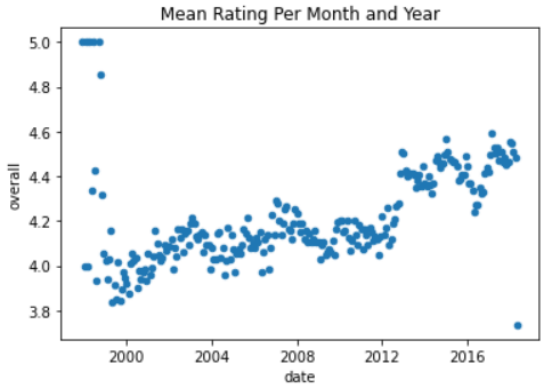


In the scatterplot graph displaying the correlation, we saw a very bimodal distribution where many exclamation marks were correlated with either a very low or a very high rating. We also grouped months and years and looked at the relationship between those variables and average ratings. We saw that reviews with over 15000 characters tended to be rated more highly. In contrast, reviews under 15000 characters had ratings ranging from 1.0 to 5.0.



We looked at ratings for items that had greater than or equal to 4 reviews because items with less than four reviews tended to all have the same rating. Thus on the scatter plot, it was clustering around a singular rating and was not very useful data. Looking at the scatter plot, however, we see a slightly sharp increase in ratings around 2012, which has been increasing since. To further investigate this trend and to see if it appears in other temporal variables, we plotted the mean rating against the year, month, day of the week, and day of the month.





The scatter plot describes the relationship between the average rating grouped by year and the period of years from 2000 to 2020. We see a sharp increase in the average rating around 2012/2013, which continues to increase after those years. This trend is also present when graphing the average rating against the month and year. Based on this data, these temporal dynamics and trends are something we will consider when developing our model.

We also looked into the number of capitals in a review summary and the review's rating.

There seems to be no significant correlation or difference between the number of capitals in a review summary for high-rated items and low-rated items.

2. Our predictive task will predict the user's rating for a specific product. We will evaluate our model using Mean Squared Error. The models can be compared by looking at the MSE for a validation subset of the data. For our very first and most basic baseline model, we will simply just predict the mean rating of all the items in the training set. This will be the best model we can have without any information about the review. Now that we have the basic baseline, we will be able to determine the performance of our more complex models we develop relative to this baseline. Surely, we should be able to develop a model that performs better than the baseline by actually taking advantage of the information we are given about the user and the item. To assess the validity of our models' predictions, we will use the same training, validation, and test datasets to ensure that the models work with the same data. This way, we can compare the MSE for the validation dataset between models and determine which model performs the best.

As for the features we will try to use for our models, we will use the userID, productID, the review text of the review we try to determine the rating of, and the timestamp of the review. From our exploratory analysis in part one, we found that there was a relationship between the time of a review and the rating of that review. As for the userID and productID, we can take advantage of information from our training set if the user and/or product has been seen before, since then we can use the previous data to have the model make a more informed decision on its prediction on the user/item. Lastly, for the review text of the review, we figured that with some sentiment analysis, we would be able to derive some sort of information on what the user thought of the item in the review, and with that we would likely be able to make a more accurate prediction on the rating. For data processing, the timestamp will be converted into a feature using FPMC to see the interactions between user and the previous item as well as the previous item and current item. Additionally, to get/process the data, we simply read through the training data and stored the information of the users, id, and timestamp so that they could be used by our models.

3. Our proposed model is a FPMC model to determine the review rating. This is due to the fact that FPMC includes previous reviews as well as interactions between users and multiple items. We thought that this model architecture would account for the slight correlation in time for each year. When we first trained it, the model seemed to take too long to converge to a good point so we increased the learning rate. We also increased the number of training epochs so that the model could converge.

We ran into issues with cold start items or users because their weights were not adjusted for in the model. We remedied this by using the global average as a prediction and implemented this method in all models. This prediction also

affected our loss but has the same effect for all models.

We ran another basic model that just has user and item bias terms that seemed to get better MSE. We also looked at a latent factor model that had similar issues to our FPMC model likely due to underfitting and too low of a learning rate. The motivation for the latent factor model was to see if previous items had a large effect on the ratings.

We also tried a linear regression model with one hot encoded year, month, and day of the week due to the trends that we saw on our graph. This model would take into account the time of rating instead of the previously rated item that might not have anything to do with the rating. We decided to test this model out as a comparison to see if just taking account of the time of review is more important than knowing the previous review.

We then tried to see if a bag of words approach would help us give a more accurate rating prediction. To do this, we defined a set sized dictionary to keep track of the 1000 most frequent terms. Then for each review, we would create a feature vector of the counts of these frequent terms, which would be plugged into the model to predict the rating. Doing this, our model was slightly worse than our more basic linear regression mode, giving an MSE of around 1.069. We attempted to improve our model by increasing the dictionary size, since it is likely that with more information on the most popular words in the review text, the more accurate a prediction we can make. In the end, our best version of this model was with a dictionary of size 5000, giving us an MSE of around 1.009, which was around .06 better than our first version of the bag of words model. However, this is still slightly worse than our basic linear regression model.

The FPMC model does quite well when previous item and current item pairs are complementary or correlated. However, if the two items are independent of each other then the model will just predict very similar to the latent factor model.

The strength of the latent factor model is that we can take into account the interactions between user and item but it doesn't factor in the review time. The linear regression models didn't take into account the interactions between the item and the user.

For the basic models, we will try to optimize them with 5-fold cross validation and a random grid search of the hyperparameters. Similarly, for the latent factor model and the FPMC model, we will use a random grid search of the hyperparameters to find the hyperparameters that give the largest change in MSE.

After adjusting the learning rate and adding more training epochs, we found that all of our models began to overfit to the training data. Our validation MSE would be around 0.9 or 1.0, except our training MSE would be around 0.7 for the linear model and about 0.4 for the latent factor model and 0.35 for the FPMC

model. We tried to address this issue by increasing the regularization constant, but for the latent factor models this change did not help the overfitting.

4. We received our data set from a list of data sets provided by Julian McAuley. Those involved with the creation and scraping include Jianmo Ni, Jiacheng Li, and Julian McAuley. There have been other existing data sets regarding products and reviews on other shopping websites and other data sets regarding other Amazon products besides products in the "Movies and TV" category. These data sets may have been used to predict which items are recommended to which users and at what times. They may also have been used to looking at specific trends to find their worst-performing and best-performing items. The current state-of-the-art methods employed to study item product reviews may involve creating an annotated data set and a pipeline involving various models. Various factors, including the type of product, the history of the user's purchases, text analysis, sentiment analysis of the reviews, and the specific time may be accounted for in a modern model.
As for state of the art methods, on May 19th, 2021, Commerce.AI, a company that utilizes AI to help understand and analyze commerce data, published an article detailing their Amazon product review analysis. Approach. In its introduction, the article establishes how manually analyzing Amazon reviews is complex due to the endless number of reviews made daily and its inability to track trends over time. With AI, however, Commerce.AI can more easily deal with the dynamically changing Amazon review data and produce a more detailed analysis of things such as tone, language, keywords, and trends over time. For Amazon product review analysis, the focus of Commerce.AI is on sentiment analysis, which tries to gather data and ideas from the review text. In the article, Commerce.AI states that they use three main types of sentiment analysis. The first is predictive sentiment analysis, which predicts how customers will react or feel about a topic or product in the future. This is very relevant to this assignment because our goal with our model is to use predictive analysis to predict what rating a specific user would give to an item in the future. The implications of this data would have a significant impact on commerce. If we can predict users' ratings accurately, we could recommend the customer items they would like and vice versa, most likely leading to higher customer satisfaction. The second primary type of sentiment analysis that Commerce.AI applies is Diagnostic Sentiment Analysis. In this case, instead of trying to predict the sentiment of a specific product or idea, diagnostic sentiment analysis would try to analyze existing trends within the data, for example, detecting if specific news stories are 'fake news.' The third and final type of sentiment analysis that Commerce.AI discusses is sentiment classification. In this case, analysis is used to derive meaning from text, i.e.,

an Amazon product review. Commerce.AI mentions many approaches to sentiment classification that we have gone over in class, such as a bag of words and TF-IDF. Sentiment classification is also relevant to our rating predictions model in part two. If we can determine the sentiment of a specific review, we will more likely be able to accurately predict that review's rating. Therefore we applied these concepts in the construction of our predictive model.

Although our sentiment analysis/bag of words model was not the most accurate model that we were able to develop, it was still relatively a solid model as it did outperform the very most basic baseline model we defined by just predicting the mean. We would therefore say that our conclusions from the Commerce.AI article is similar to our own findings, because from the model we defined, clearly analyzing the review text/using bag of words was a good feature to use within our model, which shows the importance of sentiment analysis and the bag of words approach. Perhaps the most optimal model somehow incorporates sentiment analysis with one of the other models we had created, instead of it being used on its own like the model we designed.

5.

Model	Validation MSE
Global Average	1.1932686128249765
Linear	0.9375539798983413
Linear + Time	0.9298113585516243
Latent Factor	0.9738091841086444
FPMC	1.0749071259060803
Bag of Words	1.0098552483236374

Based on the validation MSE, we found that the linear model with the time features performed the best. After running this model on our test dataset, we achieve a test MSE of 0.9325879192036294.

Our proposed model of FPMC actually does the worst of the model except the naive global average model. At first, we were very surprised to see that with more parameters, we did not have a lower loss but in fact a much higher one. We believe that with the small amount of data that we are using, models with many parameters like the latent factor model and FPMC will not have enough different data to actually create good embeddings for either item, user, or their interactions. With only a dataset of 500,000, we have 166456 unique user IDs and 7189 unique items. Previous and current item pairs likely have 4-5 sets in the training data which makes the model easily overfit to these reviews. The users are likely to have only 4-5 reviews as well which leads to poor parameter representation and therefore, poor rating predictions.

The most basic feature representations for items and users as well as time did very well because the model was not overfitting to the

training data as much. The time parameter gave the model a slight edge likely due to the trend that we saw in our exploratory data analysis.

The bag of words model with a dictionary size of 5000 did much better than our FPMC model but still does not do as well as just making parameters for users and items as well as accounting for time. There is likely a better model if we combine the bag of words model with our linear + time model. However, it leads to a large increase in parameters to train so we might face an overfitting problem again with a dataset of our size. Another possible model to experiment with is a tf-idf model which will likely do better than the bag of words model but maybe not capture enough information to reach the effect of the basic linear model.

If we were to do this experiment again, the dataset would need to be much larger than 500,000 for the latent factor models to be able to stop overfitting to the training set. These deeper learning techniques and models are very data hungry and usually a simple linear model will capture most of the information for the prediction task especially at the scale of data that we are using. In all, this experiment has shown that the most complicated models that model several different interactions are starving for data while many simpler models work better at this scale.

Citations

Jianmo Ni, Jiacheng Li, Julian McAuley
Empirical Methods in Natural Language Processing (EMNLP), 2019

Bussler, Frederick. “Amazon Product Review Analysis: The Ultimate Guide (2021).” *Commerce.AI - AI for Commerce*, https://www.commerce.ai/blog/amazon-product-review-analysis-the-ultimate-guide.