Amazon product rating prediction

Divija Devarla
A59004004
ddevarla@ucsd.edu
Department of Computer Science
University of California San Diego

Subha Ramesh
A59002412
s2ramesh@ucsd.edu
Department of Computer Science
University of California San Diego

Vibha Satyanarayana
A59010723
vsatyanarayana@ucsd.edu
Department of Computer Science
University of California San Diego

***Abstract*—The advent of online shopping e-commerce platforms like Amazon, eBay, etc. enables a wide range of choices for end-users with respect to choosing products. Building a good recommender system to help users choose the product which matches the user's preferences helps both Amazon and users. In this paper we predict the user's rating which would help Amazon suggest products to users accordingly and increase the probability of the product being purchased. Various features we have considered to predict rating are review text, price, helpful and unhelpful votes, user to user, and item to item similarity. Considering these features and building models to predict rating is the goal of this paper.**

***Index Terms*—Amazon product review, rating, ridge regression**

I. Introduction

The purpose of this paper is to explore various models to predict the user rating from features like review text, price, reviewerID, asin, unixReviewTime. The dataset is based on Amazon reviews and focuses on the Electronics domain. The models tried in this paper use various combinations of features to predict the most probable rating and obtain the least Mean Squared Error. Some of the models tested include Similarity based models like Jaccard, Pearson correlation with and without temporal features, Unigram, bigram and mixture models. We compare various models tried in this paper and finally conclude the model which best performed.

II. Exploratory Analysis

A. Dataset

The dataset used in this paper is the Amazon Product Review dataset. The Electronics dataset consists of two parts: the product review data which contains the reviews each user writes for the product purchased, and the product metadata which includes the information of each product. Both parts are JSON files. Each review in the product review data consists of the following attributes:

Attribute	Description
reviewerID	ID of the reviewer, e.g. A2SUAM1J3GNN3B.
asin	ID of the product, e.g. 0000013714
reviewerName	name of the reviewer
helpful	number of helpful votes
unhelpful	number of unhelpful votes
reviewText	text of the review
overall	rating of the product (1 to 5)
summary	summary of the review
unixReviewTime	time of the review (unix time)
reviewTime	time of the review (raw)

Metadata present in the Amazon dataset consists of the following attributes:

Attribute	Description
asin	ID of the product, e.g. 0000031852
title	name of the product
feature	bullet-point format features of the product
description	description of the product
price	price in US dollars (at the time of crawl)
imageURL	url of the high resolution product image
related	products also bought, also viewed, bought together, buy after viewing
salesRank	sales rank information
brand	brand name
categories	list of categories the product belongs to
similar	similar products table

B. Dataset properties and interesting findings

A preliminary analysis of the dataset revealed the following properties:

- Total number of reviews - 1689188
- Average rating - 4.22
- Number of unique reviewer IDs - 192403
- Number of unique asins - 63001
- Average Helpfulness ratio - 0.1696553
- Item - SAN Disk Micro SD card with asin - B007WTAJTO has 4915 reviews which is the highest in the dataset
- The maximum number of reviews a user has is 431

- Number of asins present in the metadata set is 498196. Number of asins with reviews is 63001. This shows that many products do not have reviews
- Every user has a minimum of 4 reviews present in the dataset
- Each item has a minimum of 5 reviews present in the dataset

C. Basic Statistics

We first analyzed the overall distribution of rating across the dataset. It was observed that more than 50% of the electronic products have a rating of 5.

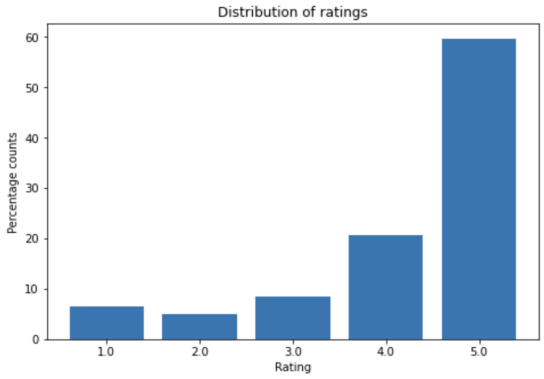

Fig. 1. Rating distribution

Fig. 2, describes the relationship between price and overall rating of products. The x-axis is the price of the product and the y-axis is the overall rating of the product. This was plotted for a subset of the data (10000 reviews).

Fig. 2. How rating varies with price

We also studied how the rating trends have changed through the years. From Fig. 3, we can observe that more users have started rating more products. The products receiving rating 5 have increased significantly.

We can also see the most common unigrams and bigrams in the review text. Fig. 4, is a word cloud of 10000 most common

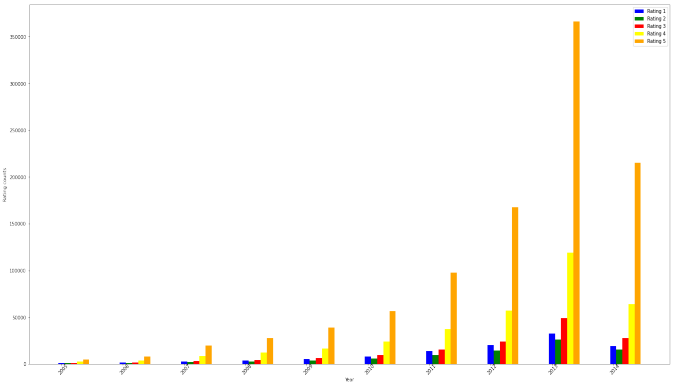

Fig. 3. Rating distribution across the years

unigrams and bigrams.

Fig. 4. Most common unigrams and bigrams

It is also interesting to see the relationship between sentiment and rating. Fig. 5, depicts the number of positive, negative and neutral reviews across all ratings. As expected we can see that the positivity levels increase with the increase in ratings and the number of neutral comments are always greater than positive and negative reviews.

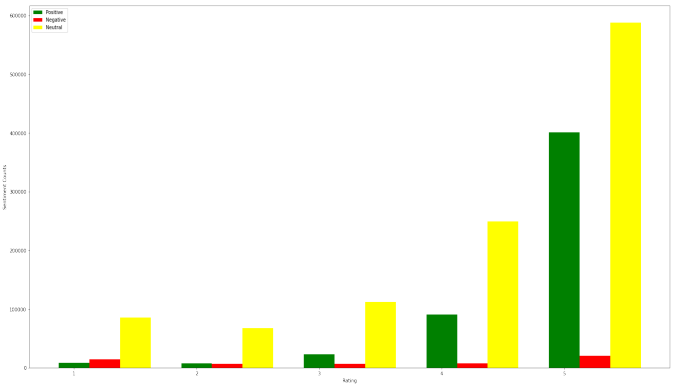

Fig. 5. How sentiment varies with rating

We observed that the number of helpful reviews also contributed to the overall rating. From Fig. 6, we can see that products with higher ratings tend to have more helpful reviews.

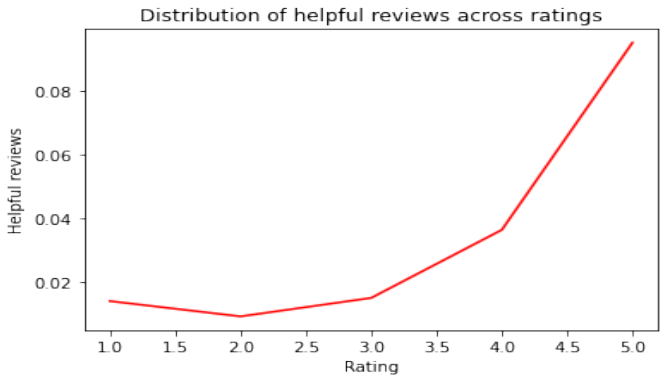

Fig. 6. How helpfulness ratio varies with ratings

Fig. 7 depicts the correlation between all the features under consideration. We can use this to determine if there is a relationship between any feature and rating.

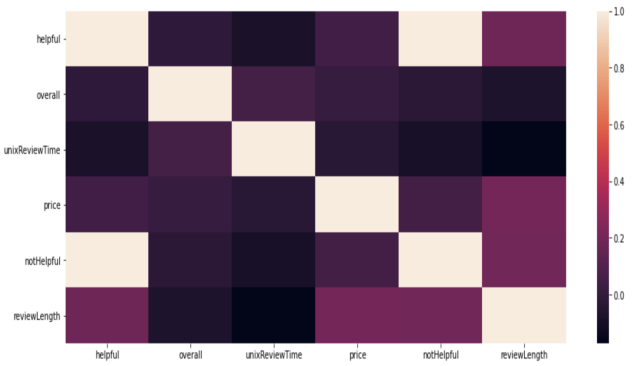

Fig. 7. Correlation of features

III. PREDICTIVE TASK

The goal is to predict the user rating received for an electronic item considering a combination of features like review text, price, reviewerID, asin, unixReviewTime, and helpfulness ratio. These features were considered after analysing the dataset and understanding how each of these features are correlated to the user rating. Once the features were selected, various models were tried with a combination of these features. Some models experimented with include Ridge Regression, Linear Regression, Singular Value Decomposition (SVD), Factorization Machines (FastFM), and Time Weighted collaborative filtering. We modeled the rating as a continuous variable and chose models accordingly that would allow us to predict real-valued ratings. We found that Ridge Regression performed the best when evaluated in terms of Mean Squared Error. More details will be provided in the next section.

A. Model evaluation

The final model was chosen based on how well the model performed compared to the actual rating present for that review. This measurement was calculated using the concept of Mean Squared Error (MSE):

$$MSE = \frac{1}{n}\sum_{1}^{n}(y_i - \hat{y}_i)^2$$

B. Baseline Model

A baseline model was established. Every model considered was compared to the baseline and evaluated based on MSE with respect to the baseline. The baseline model for this paper is based on the mean rating value across the entire dataset, as the predicted value for all the reviews. The MSE obtained for the baseline model is 1.4057.

C. Model Validation

We considered 12.5% of the dataset as the validation set, in order to tune the model selected. The validation set was constructed based on a shuffled subset from the original data.

D. Data Preprocessing

Before evaluating the data, the dataset was preprocessed to obtain meaningful insights and help us build a good recommender system. This dataset contains product reviews, and metadata from Amazon. We merged the product reviews and metadata into one dataset and removed all the rows which contained either a Nan value or missing data. As part of text preprocessing we removed stop words and punctuation.

E. Feature Selection

Various features present in the dataset were compared with the rating to help us decide which features can be considered in the model.

- Price - From the statistics obtained in section 2, we can see that as the price increases, the rating a product receives decreases. In most cases users prefer a product which is cheaper. Considering this analogy price is considered as one of the features
- Helpful Ratio - This is a ratio of number of helpful votes to total number of votes per review. From our analysis, the ratings increase with the ratio.
- Unix Review Time - This feature considers the relationship between time and rating, which are directly proportional.
- Product ID / Reviewer ID - asin, which is the product ID and reviewer ID are considered as one of the features. Considering these features helps us capture the relationship between product and user.

- Mixture Model of most Popular words - A Unigram set is a one word sequence of review text. A Bigram set is a set of two word sequences of review text. Mixture models include a combination of unigram set and bigram set.

- Sentiment score - This is a measure of how positive, negative or neutral the review text is. We used the VADER sentiment analysis from NLTK to determine this score.

IV. MODEL

A. Model chosen

The model that performed the best on the data was based on a Ridge Regressor. We chose Ridge Regression to minimize the problem of overfitting or underfitting that could occur with Linear Regression and achieve an increase in performance. With Ridge regression, we aim to optimize the prediction of the rating, thus reducing the MSE as per the following cost function:

$$J(W) = \frac{1}{2N}\sum_{1}^{N}((W_0+W_1X_1^{(i)}+...+W_pX_p^{(i)})-Y_i)^2+\frac{\lambda}{2N}\sum_{1}^{P}W_j^2$$

where,
W represents the weights,
X represents the observations,
J(W) represents Cost Function for Ridge Regression. The cost is the normalised sum of the individual loss functions.

Based on our correlation analysis of the dataset, most features present in the dataset contribute very little to the user rating, thus proving to be useless in the prediction. Intuitively, the products with positive reviews tend to produce higher ratings. It is understood that the reviewText influences the user rating to a large extent. Thus the feature vector predominantly is based on a mixture model of 10000 most frequent unigram and bigram words found in the training set. The performance achieved in this model also indicates that this is the case.

We also extracted the sentiment associated with the review using the VADER (Valence Aware Dictionary for Sentiment Reasoning) library which maps lexical features to emotion intensity. The compound score generated by normalizing the negative, neutral and positive scores, was then fed into the Ridge Regressor as an attribute.

Inclusion of the helpful ratio, the price of the product and the unixReviewTime (since the time has a good correlation with the rating) improved the performance of the model. The model was tuned using alpha (regularization strength) and solver for the computation routine. The inclusion of the new attributes and the tuning of the model reduced the MSE value from 1.05 to 0.98.

The training time for this model was greater compared to the Latent Factor models and the Factorization machines on the same dataset. This was due to the processing time associated with the Sentimental analysis on the entire dataset. Given the MSE achieved, this seems to be a reasonable trade-off. But the model is prone to scalability constraints with the increase in the size of the training set.

B. Other models tested

a) Singular Value Decomposition: The Singular Value Decomposition (SVD) model is based on the Matrix Factorization Algorithm, which uses Stochastic Gradient Descent for optimization. This model is primarily based on the reviewer and product factors and predicts the ratings based on the reviewer and item biases calculated during the course of the algorithm. We experimented with the SVD to achieve a good MSE value with quicker training times, while modelling the user-user and item-item relationship.

$$\hat{r_{ui}} = \mu + b_u + b_i + q_i^{\mathrm{T}}p_u$$

where,
r_{ui} represents the prediction
μ represents the rating mean
b_u represents user bias
b_i represents item bias
p_u represents user factor
q_i represents item factor

The input features to the SVD model were reviewerID, productID and the associated rating. The model was tuned on the regularization, number of epochs, learning rate parameters, using the GridSearchCV algorithm which performed an exhaustive search over the parameter space.

SVD does not provide support for incorporating the text based features in the dataset, which were found to have a high correlation with the user rating. Thus the MSE achieved with SVD is greater than that of the text-based model.

b) Time-Weighted Collaborative Filtering: This model is based on the addition of a temporal dynamic to the item-item collaborative filtering. This model was chosen to incorporate the timestamp, which has a good correlation to the rating of the product. The features associated with this model are reviewerID, productID, unixReviewTime and the rating of the product. The timestamp is used as a weight to the interactions between the users and products. Each product is compared based on its similarity with the other products in the dataset, which is used as an estimator for the rating prediction.

$$r(u,i) = \bar{R}_i + \frac{\sum_{j\in I_u\setminus j}(R_{u,j}-R_j)Sim(i,j)f(t_{u,j})}{\sum_{j\in I_u\setminus j}Sim(i,j)f(t_{u,j})}$$

The similarity metric used here is based on the Jaccard similarity model. The temporal weight is based on an exponential decay function that takes the timestamps associated with the two items under comparison.

$$J(A,B) = \frac{|A\cap B|}{|A\cup B|}$$

This model achieved a tolerable performance improvement over the usual collaborative filtering models (user-user and item-item). This model also had a Mean Squared Error close to the one associated with SVD.

C. Issues faced while testing various models

Scalability was a major factor affecting the training of models, given the dimensions of the dataset. The exploration of different solvers for Ridge regression was limited due to the size of the dataset, with each solver leading to an increase in the computational time and eventually leading to a Kernel crash.

We ran into issues with overfitting on Factorization machine models like FastFM. The MSE on the test set was much higher than

the train MSE. The dataset in itself had poor correlation among its attributes. This constrained the variety of the models that could be successfully trained against it.

The dataset had a lot of missing data. For instance, salesRank only had approximately 28000 valid entries among the entire dataset. This constrained the usage of salesRank as a feature.

Pre-processing times of the summary data increased exponentially with the length, due to the high dimensions associated with it. This slowed down the training times of the model, preventing us from considering this attribute.

V. LITERATURE REVIEW

The dataset [1] [2] [3] [4] we chose to consider is available on the Amazon product data page provided by Prof. Julian McAuley. The dataset was used for justifying recommendations using distantly labelled reviews and fine grained attributes.

In one of the papers [7], a machine learning based model which learns from graphs, developed based on features like helpfulness votes and total votes received by the user, is used on the Amazon Reviews dataset. Graph based models are further compared with non-graph based models, which consider review text and observed 95% prediction accuracy with graph based models.

In another paper [6], rating prediction is modelled as a multi-class classification problem, by predicting the class using the sentimental categorization, on the same dataset. The model achieved 61% accuracy on a SVM Classifier.

[8] used a Yelp dataset, combining linear regression with a bag of words model, to predict the star rating, using the review text. This model achieved a RMSE value of 0.6.

[9] used IMDB movies dataset to predict the user rating for a movie using a classifier trained on IMDB attributes.

Some of the state-of-the-art methods which are currently being employed include Convolutional Neural Network based models and word embedding representation on reviews [10] [11].

The existing works use models which are different from the models used in this paper. Given the numerous methods available to predict the user rating, varied datasets and the features considered, the results are different. The evaluation criteria used in this paper is also different from those of the other papers.

VI. RESULTS AND CONCLUSION

Table 1 shows the Mean Squared Error values obtained from various models which were evaluated. The best MSE was obtained with Ridge Regression. The Base Ridge Regressor is based on a unigram-bigram mixture model. Our experiments and exploration showed that the review text captured more information about the user rating compared to features like user and item similarity. This is because text based features are high dimensional and reflect how the user rates the product. Since the test set can contain heretofore unseen reviewer IDs and asins, the review text helps us predict the rating, thus resolving the cold start issue associated with the usage of Factorization models and SVD. The user's sentiment is a direct indication of how much they would rate the product. We also saw that the pricing information of the product as well as the time it was rated also contribute to the rating. Our model performed better when the helpfulness ratio was included in the feature set. This shows us that a product is more likely to be bought if it has helpful reviews.

The features which did not work well with the models considered are summaryText and salesRank. They reduced the performance of the model due to missing/ Nan values and extreme training times.

Ridge Regression performed better because it incorporated the review text, timestamp, helpfulness and price i.e it basically incorporates all the features which help us predict rating.

TABLE I
RESULTS OF SOME EXPERIMENTS CONDUCTED

Model	**Details**	**MSE**
Baseline	Mean rating	1.41
SVD	$\lambda = 1$	1.21
SVD	$\lambda = 0.3$	1.19
Jaccard based rating prediction		1.22
Temporal based rating prediction	decay = 0.00000001	1.22
Temporal based rating prediction	decay = 0.000000001	1.21
Fast FM	users and items	2.48
Base Ridge Regressor		1.06
Base Ridge Regressor + VADER sentiment + helpful ratio		1.00
Base Ridge Regressor + VADER sentiment + helpful ratio + price	$\alpha = 1, solver = auto$	1.00
Base Ridge Regressor + VADER sentiment + helpful ratio + price + time	$\alpha = 1, solver = auto$	0.98

The parameters used in Ridge Regression are alpha(regularization strength) and the solver which is used in computational routine. Ridge Regression uses the squared magnitude of alpha as the penalty term indicating the L2 regularization. We have chosen alpha as 1 and achieved the most optimal MSE while the solver is automatically chosen based on the type of data.

In conclusion, Ridge Regression worked optimally on this dataset providing an MSE of 0.9825.

REFERENCES

[1] https://cseweb.ucsd.edu/ jmcauley/datasets.htmlamazon_qa

[2] Ruining He and Julian McAuley. 2016. Ups and downs: Modeling the visual evolution of fashion trends with one-class collaborative filtering. In Proceedings of the WWW. 507–517

[3] Jianmo Ni, Jiacheng Li, and Julian J. McAuley. 2019. Justifying Recommendations using Distantly-Labeled Reviews and Fine-Grained Aspects. In EMNLP 2019

[4] McAuley, J., Targett, C., Shi, Q., and Van Den Hengel, A. Image-based recommendations on styles and substitutes. In SIGIR (2015).

[5] Pang, B., Lee, L. Vaithyanathan, S. (2002). Thumbs up? Sentiment Classification using Machine Learning Techniques. Proceedings of EMNLP, 271-278. doi: 10.3115/1118693.1118704 Pang, B. Lee, L. (2005).

[6] Seeing stars: Exploiting class relationships for sentiment categorization with respect to rating scales. Proceedings of the 43rd Meeting of the Association for Computational Linguistics, 115–124. doi : 10.3115/1219840.1219855

[7] http://snap.stanford.edu/class/cs224w-2012/projects/cs224w-019-final.v01.pdf

[8] Fan, M. Khademi, M. (2014). Predicting a Business' Star in Yelp from Its Reviews' Text Alone. ArXiv e-prints. doi : 1401.0864

[9] Hsu PY., Shen YH., Xie XA. (2014) Predicting Movies User Ratings with Imdb Attributes. In: Miao D., Pedrycz W., Ślzak D., Peters G., Hu Q., Wang R. (eds) Rough Sets and Knowledge Technology. RSKT 2014. Lecture Notes in Computer Science, vol 8818. Springer, Cham. https://doi.org/10.1007/978-3-319-11740-9_41

[10] Zahid Younas Khan, Zhendong Niu, CNN with depth wise separable convolutions and combined kernels for rating prediction,Expert Systems with Applications, Volume 170,2021,114528,ISSN 0957-4174,https://doi.org/10.1016/j.eswa.2020.114528.

[11] S Hasanzadeh, S M Fakhrahmad, M Taheri, Review-Based Recommender Systems: A Proposed Rating Prediction Scheme Using Word Embedding Representation of Reviews, The Computer Journal, 2020;, bxaa044, https://doi.org/10.1093/comjnl/bxaa044