Rating Prediction on Amazon Book Review

Group 3_Amazon: Neoh Hai Liang, Sabreenah Saabirah Binti Shaik Dawood, Sabah Anwar Azmi, Muthuk Kumaran A/L Subramaniam

I. Structured Abstract

THIS research proposed a best performing prediction model on the data science book reviews in Amazon platform. The aim is to find the best performing prediction model from the selected algorithms: K-Nearest Neighbors, Decision Tree, Naïve Bayes, and Support Vector Machine (SVM). The dataset contains unstructured comments extracted as the text features in the prediction model. The data is split into 75:25 training-test ratio with the 5-fold cross validation for parameter tuning. Accuracy is used as the performance metrics. Upon examination of the result, Radial kernel SVM with the parameter C=1, sigma=1 yields the highest test accuracy at 68%. This result will allow more consideration of rating prediction on other domains and may contribute to future research of rating prediction using machine learning techniques.

Keywords: Rating Prediction, Amazon, SVM, Machine Learning

II. Introduction

Amazon Book Review is a platform to let users recommend their favorite books to other users. Amazon uses a recommender algorithm to rank the suggestions to users based on the volume of downloads and the number of book reviews. Besides, Amazon also pays authors for self-publishing their e-books via AmazonEncore, to further enlarge their library collection. By having more books on the platform, it will attract numerous users and traffics to the platform and hence generate more conversion. Based on Sellbrite.com (2017) [6], Amazon Books contributing about 9% of the total revenue. Moreover, according to Hall, M. (2020) [3], Kindle e-books were popular than traditional printed books in 2011 in the Amazon book platform. Amazon becoming so popular because it started to become an online bookseller in 1994 [6], hence it has a huge user database and generating a large portion of total income since then.

In traditional, human might need to manually tag a rating that whether the user like the product or not by understanding the comments, however, this step is very time-consuming and costly in term of financial. With this sentiment analysis, it is possible to train a machine learning model that can predict the rating automatically based on the unstructured text reviews commented on by the viewers.

However, this raised another question that in mind, which machine learning algorithm is the best performing in the book rating prediction? What is the optimal combination of hyper-parameter that produced the highest accuracy in the machine learning algorithm?

TABLE I
Features and Label In Raw Data

Feature	Type	Value/Statistics
stars	Discrete numerical	Range: 1 – 5 Mean: 4.3 Std: 1.2
comment	Text	Missing: 612 Total: 20,647
book_url	Text	Missing: 0 Unique: 836 Total: 20,647

Note: The target variable of the study is ***stars***, while ***book_url*** is an identity column

Hence, this project is aimed to find the best classifier within the selected algorithms which can predict the Amazon Book Reviews based on the viewers' comments. This study will help the Amazon platform to quickly identify the books that people are giving positive reviews and recommend to their users. This will not only benefit the author by having good exposure to their books but also help to increase the sales to the Amazon platform.

III. Background and Literature Review

Sentiment analysis is a machine learning approach to classify the unstructured text data into pre-defined classes. It is useful specially to analyze the emotions and product reviews. Sentiment analysis models primarily focus on the classification of polarity (good, bad, neutral) and emotions (happy, angry, sad). It is also used more prominently in the text domains such as reviews, tweets and feedbacks [8].

According to Bilal Saberi and Saidah Saad [5] sentiment analysis is an act of distinguishing and classifying the outlook that was presented. It starts with the collection of outlooks then processing it through examination of the outlooks. This is a very useful process to understand and gathers the outlooks of the communities although is it not a simple process as it requires analyzing slangs and misspelled wordings [4].

Cuizon, J. C. and Agravante (2020) [1] presents a sentiment analysis to use text reviews predict ratings. They compared the rating from Google Maps with rating rated manually by human based on the review texts. The performance metrics is using MAE. Based on their result, the rating prediction model achieved at 82% in overall.

IV. Methodology

This section describes the methodology that has been employed, which divided into several subsections as follows.:

A. Dataset Description

The dataset was based on 20,647 data science book reviews from the Amazon platform. This dataset was published on Kaggle by Vladimir under CC0: Public Domain license [7]. These reviews are scraped with *Scrapy* on Amazon.com.

The dataset contains 3 columns which are the ***stars***, ***comment***, and ***book_url***. The ***stars*** column is the rating given by the users. The ***comment*** columns contain the comments left by users. The ***book_url*** are the hyperlinks to the Amazon site for each review. The target column is identified as "***stars***". The target ranges from 1 to 5 with 1 being the lowest rating and 5 be the highest rating. The data type for ***stars*** is numerical whereas for ***comment*** and ***book_url*** are text.

The mean of ***stars*** is about 4.3 meaning the majority (at least 50%) of sample reviews are rating more than 4.3 stars. While the standard deviation of ***stars*** is about 1.2, meaning there are at least 75% of the rating lies between $4.3 \pm 2(1.2) = [1.9, 6.7] \approx [1.9, 5]$ stars given, based on the Chebyshev's Theorem. Besides, there are 612 ***comments*** that are blank, which indicates the situation where users provide the rating without leaving any comment. Overall, there are 836 unique books being reviewed in this dataset.

B. Data Treatment

This dataset was intended built for sentiment analysis combined with the Natural Language Processing technique. Hence, we only need to drop the identity column (***book_url***).

Besides, the ***stars*** column is in numeric data type, however the book rating from 1 to 5 is an ordinal scale of data, which the steps between each level are meaningless, hence we should treat the ***stars*** column as an ordered factor with 5 levels instead of a simple numeric data.

Moreover, we also dropped all rows with missing comments because these missing comments are not contributing to the rating prediction model.

Furthermore, the ***comment*** column is transformed into text features for the prediction model. We used corpus methodology to convert the comment sentences into a bag of words. To ensure the texts only represented in a single form so that the text features can contribute to the model with a higher accuracy, we performed a text cleaning procedure with the ***tm*** library. We first replace all characters in upper case to lower case. Then, we removed all the noise such as punctuations, numbers, stop-words, and extra whitespaces.

This will ensure the text features to be cleaned and more insightful. We also lemmatize the strings with ***textstem*** library to ensure the string representation only have one form. Finally, we reformatted the corpus into a Document-Term Matrix (DTM) with each document (***comments***) in each row, each formatted words in each column, the cells contained the Term-Frequency Inversed-Document-Frequency (TF-IDF) weights. The reason to use this TF-IDF weights will be discussed in Section IV C below.

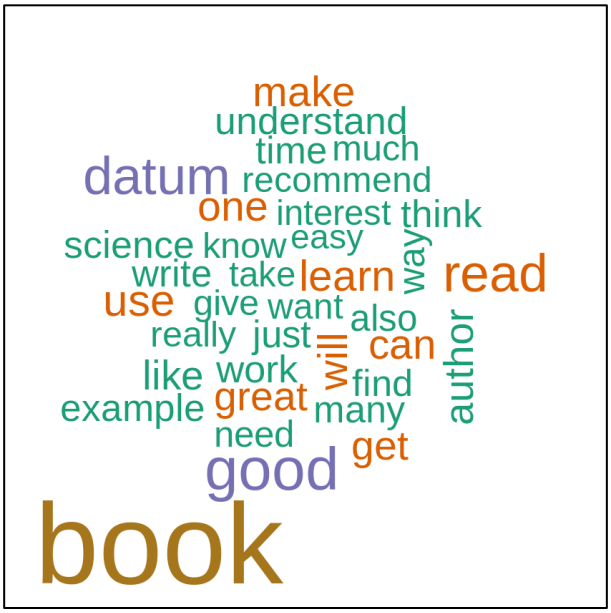
Due to the limited computational power, we reduced the size of DTM by indicating 0.9 of sparse before convert into matrix. By doing so, the DTM will only contain 37 unique words in the corpus, we dropped all the comments which does not contained any words in those 37 unique words. In result, only 94% (18893 of 20035) of the pre-processed documents are retained for the prediction purpose. After combined with the ***stars*** column in the original dataset, we renamed the target column to ***rating*** to avoid confusion, then the DTM is ready to make predictions based on the ***rating*** column.

C. Data Exploration

The distribution of the ***rating*** is plotted as below:

From the bar chart, the dataset is highly biased to positive ratings (5 out of 5 stars), which means most of the reviewers in this sample dataset are quite satisfied with the book they bought. This implies the distribution of rating is unbalanced; hence we should carefully choose a sampling method that will not yield a biased prediction model.

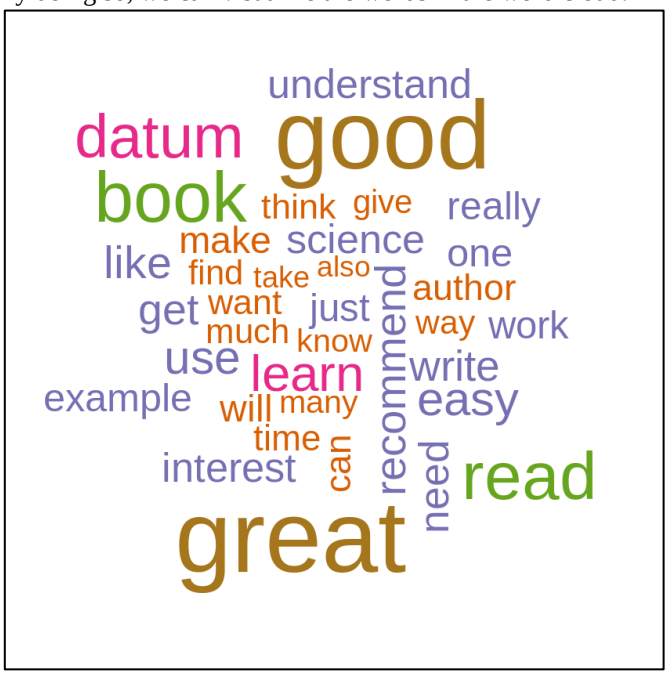
Besides, we also visualize the words in the processed corpus of ***comment***. The first approach is simply count on the word frequency. By doing so, we obtained a DTM with count of words in each document. Then, the following word cloud can be drawn as below:



The emphasized words in the word cloud are “book”, “datum”, “good”, because these words appear the most in the comments. However, the word cloud above is not insightful to the rating prediction model because the most frequent words are not necessary become the meaningful words (Fueyo, E., 2018) [2]. This situation can be observed in the stop-words.

Stop-words are the meaningless words that commonly appears during the daily conversation. Some examples of stop-words are the subject pronouns in English (I, he, she, they) and prepositions (with, despite, of, and). These words although is frequently used in daily conversation but does not contribute to our models.

Therefore, with the idea that the frequent words may not be meaningful, we used the TF-IDF weighting method. The weight is giving a higher score will implies a meaningful word. By doing so, we can visualize the words in the word cloud.



From the word cloud, we can find that the emphasized words that might contribute a positive rating such as “good” and “great” has higher TF-IDF scores. However, the word “book” which is not contribute much to the prediction model is having a lower TF-IDF score.

V. Models, Results and Discussion

A. *Data Modelling*

In order to make an unbiased Machine Learning algorithm, the data is split into 2 sets: training set and test set. The test set act as an unseen data that can be used to evaluate the Machine Learning algorithm. A split is done on the dataset with a training-to-test ratio of 75:25. The split is done on random to ensure the equal chance for every class to be collected in the test set.

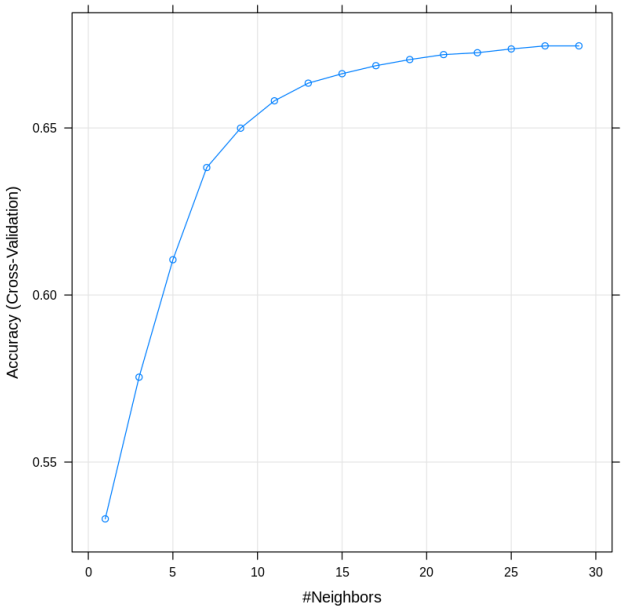
Due to imbalanced target class in ***rating*** column (rating 5 is much higher than other rating), we decide to train the prediction model in a 5-fold cross validation for hyper-parameter tuning. The training data is split into 5 parts, training set is a combination of 4 parts while validation set takes the leftover part. The training and validation set changes every iteration. By doing so, the model can be trained in a more unbiased sense because all documents in the training set contribute to the parameter tuning process.

Moreover, we used Accuracy as the performance metric to evaluate the model. This is because the aim for this prediction model is to generate the prediction label with maximum accuracy, we want the prediction as accurate as possible. Besides, there are no requirement on the minimizing the false positive or false negative in the predicted class label, hence the performance metrices Recall, Precision and F1 is not considered.

A grid search is implemented to fine-tune the hyper-parameters in the prediction models. For every model, we defined a search range of the parameters that tailored the prediction model based on our data. Every element in the search range passed into the prediction model in hoping to get the best performing model with the optimal hyper-parameter combination. The validation accuracies are compared within the grid search to find the best parameter combination. But the test accuracies are compared between different classification models.

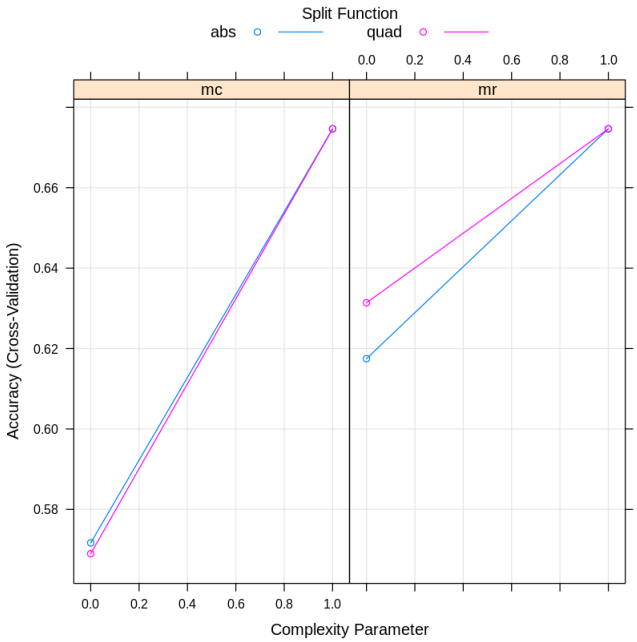
There are 4 classification algorithms selected to model the data: K-Nearest Neighbor (KNN), Decision Tree, Naïve Bayes and Support Vector Machine with Radial Basis Function Kernel (Radial SVM).

In the KNN experiment, a range of number of neighbors, ***k*** that ranging from 1 to 29 is used in the grid search. The validation accuracy will be different for every ***k*** because KNN use majority voting to predict the class of data point based on the class of closest neighbors in the surrounding. We only search for odd value of ***k*** because we want to minimize the chance of getting a tie in the majority voting. The ***k*** that produced the highest validation accuracy in the prediction model will be recorded.



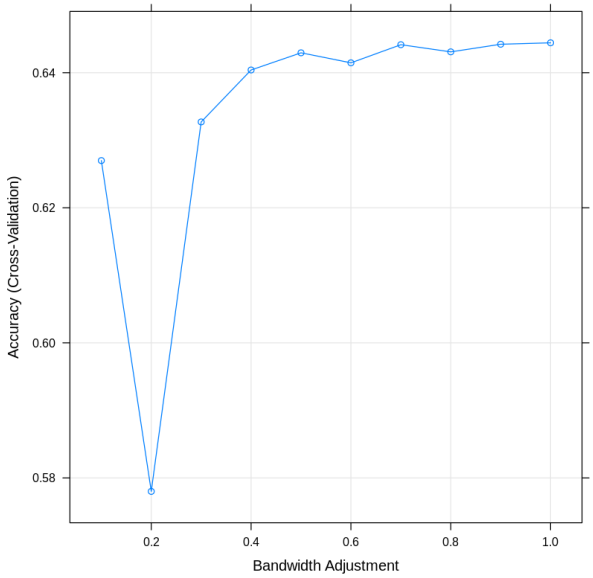
From the result, the validation accuracy converges to 67.5% as the number of neighbors, ***k*** increases. The highest validation accuracy occurred at ***k=29*** within the search range, hence the best performing model is ***KNN(k=29)*** with the validation accuracy at 67.46%.

Besides, in the Decision Tree experiment, the complexity variable, ***cp*** is searched in either 0 or 1. This complexity variable decides the growth of tree, if ***cp*** is positive, the tree will stop growing at each node. Besides, the ***rpartScore*** allow us to manipulate the splitting function, ***split*** to select the misclassification cost at the absolute (***abs***) or squared (***quad***) difference of the scores. This ***split*** will define the regulatory function to be L1 or L2 norm. Furthermore, we also can tune the prediction metric to prune the decision tree by searching the ***prune*** parameter between misclassification rate (***mr***) and misclassification cost (***mc***). By tuning in the ***cp***, ***split*** and ***prune*** hyper-parameters, we will get a best Decision Tree with the optimal combination of parameters that produce the highest validation accuracy.



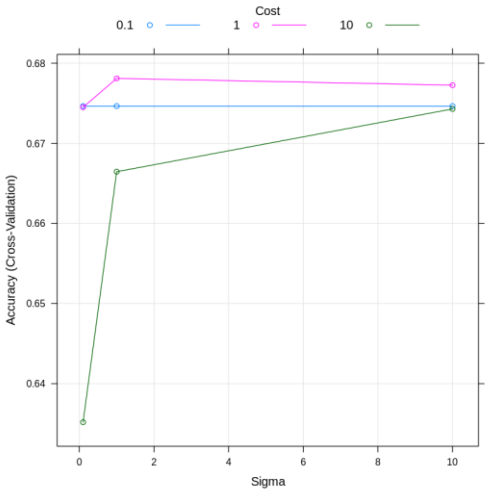
From the result, the highest accuracy happens at ***cp*** = 1, regardless the value of ***split*** and ***prune***. There are 4 Decision Trees having the highest accuracy, hence the grid search algorithm takes the first parameter set with the highest score as the final model. The selected model is ***DecisionTree(cp=1, split=abs, prune=mc)*** with the highest validation accuracy of 67.46%.

Moreover, in the Naïve Bayes experiment, we can tune the bandwidth adjustment, ***adjust*** in the parameter tuning process. This parameter controls the amount of spread in Kernel density estimation function, a higher bandwidth will be resulting a smoother but flatter estimation function. The adjustment between the bandwidth can be seen as the trade-off between bias and variance. We want to find an estimation that fit to the true density as close as possible. Hence, by tuning the ***adjust*** variable, the best prediction model with optimal hyper-parameter combination that produced the highest validation accuracy will be recorded.



From the result, the validation accuracy starting to stable at 64% after the bandwidth adjustment (***adjust***) reaches 0.4. The Naïve Bayes with ***adjust***=0.7 reaches the highest validation accuracy. at 64.63%. This implies that the best performing model is ***NaïveBayes(adjust=0.7)***.

It is well-known that SVM work well in text mining, hence we decide to take Radial SVM into consideration for rating prediction. There are two hyper-parameter that can be tuned: the cost parameter, ***C*** and spread parameter, ***sigma***. The cost parameters control the misclassification cost for the SVM. A larger ***C*** will allow misclassification example in the training data, but a lower ***C*** often cause high variance in the model, which resulting the overfit in the model because of tight margin in the decision boundary. Furthermore, the ***sigma*** controls the smoothness of the decision boundary. A lower ***sigma*** will be resulting a smaller range between the training data and the decision boundary that separates the target class. By implementing the grid search to find the best combination of hyper-parameters in SVM, we can obtain the best prediction SVM model.



From the result, the highest validation accuracy (67.81%) happened when ***C***=1 and ***sigma***=1. Hence, the best performing model is ***SVMRadial(C=1, sigma=1)***.

After retrieving the best performing models in each model category, the test accuracies of each model can be compared to find the ultimate classifier which can performed the prediction task in term of high accuracy.

From the result, all models are having an average accuracy of 66.96%. However, all models are well-fitted and able to generalize the prediction to unseen data, there is no overfitting happen because the difference between validation accuracy and test accuracy is small. Among all the selected algorithm, Radial SVM performed the best because it has the highest accuracy in the test set. This implies Radial SVM able to predict the rating at 68.06% accuracy based on the unseen data. Based on this result, we can say that the best performing model, ***SVMRadial(C=1, sigma=1)*** can correctly predict on 7 ***rating*** out of 10 ***comments*** given by the reviewers.

VI. Conclusion

In conclusion, the aim of this project is to determine the best classifier that enables to make predictions on books' ratings based on user comments. In this project, the dataset is a collection of review comments based on 20,647 data science book reviews on Amazon platform. We used the ***comment*** to extract the text features to predict the stars (***rating***). The data cleaning is performed to change the data type of the prediction target. Besides, we also dropped 612 rows with missing comments. Text cleaning is performed on the text features to reduce the noise, there are total of 37 unique words in the corpus, 94% of the preprocessed documents retained for the prediction purposed. The TF-IDF weighting method is applied to the DTM for prediction purpose.

In the data modelling part, in a 75:25 training-test split, the 5-fold cross validation method is used in the training set for parameter tuning process. We used accuracy as the performing metrics to evaluate the prediction model.

There are 4 classifiers selected to find the best prediction model: KNN, Decision Tree, Naïve Bayes and SVM Radial. In the parameter tuning process, the classifier is compared within the algorithm family, the model with highest validation accuracy is further compared between the algorithm family.

Based on the result, ***SVMRadial(C=1, sigma=1)*** is the best performing algorithm that achieve the highest test accuracy at 68%. As result, this model can correctly predict on 7 ***rating*** out of 10 ***comments*** given by the reviewers.

However, the prediction accuracy can be further improved by testing more hyper-parameter combinations, i.e.: change the searching range or tune other parameters which is not tested in this project. Besides, we can also include more classification models to find other possible best performing model, such as ensemble learning. Last but not least, we also can increase the sparse value to generate more unique words into the DTM, so add more text features to the prediction model.

References

[1] Cuizon, J. C., & Agravante, C. G. (2020, December). Sentiment Analysis for Review Rating Prediction in a Travel Journal. In *Proceedings of the 4th International Conference on Natural Language Processing and Information Retrieval* (pp. 70-74). https://dl.acm.org/doi/abs/10.1145/3443279.3443282

[2] Fueyo, E. (2018, August 2). WTF is TF-IDF? KDnuggets. https://www.kdnuggets.com/2018/08/wtf-tf-idf.html.

[3] Hall, M. (2020, April 9). Amazon.com. Encyclopedia Britannica. https://www.britannica.com/topic/Amazoncom

[4] M. S. Neethu and R. Rajasree, "Sentiment analysis in twitter using machine learning techniques," 2013 Fourth International Conference on Computing, Communications and Networking Technologies (ICCCNT), 2013, pp. 1-5, doi: 10.1109/ICCCNT.2013.6726818.

[5] Saad, S., & Saberi, B. (2017). Sentiment Analysis or Opinion Mining: A Review. International Journal on Advanced Science, Engineering and Information Technology, 7(5), 1660. https://doi.org/10.18517/ijaseit.7.4.2137

[6] Ugino, M. (2017, February 20). How Does Amazon Make Money? Sellbrite. https://www.sellbrite.com/blog/how-does-amazon-make-money/.

[7] Vladimir. (2020, August 26). Amazon Data Science Book Reviews. Kaggle. https://www.kaggle.com/vvorotnikov/amazon-data-science-book-reviews.

[8] Wang, K., & Wan, X. (2018). Sentiment Analysis of Peer Review Texts for Scholarly Papers. The 41st International ACM SIGIR Conference on Research & Development in Information Retrieval. Published. https://doi.org/10.1145/3209978.3210056

Appendix

Google Collaboratory Notebook:
https://colab.research.google.com/drive/1C3ydiYfmK3FrzvgQqSQf8k8F9VXf6Oxw

Spark Presentation:
https://spark.adobe.com/page/m52rLEP9m7zks/