Amazon Reviews Analysis

Thushani Nipunika Kalamullage Kalamullage
Applied Machine Learning
COMP-1804

***Abstract*—With a fast-growing online community of online shopping, product reviews and ratings have been popular during the past decades. Merchants even request and encourage to post consumers' opinions on online websites. Also, with the covid19 pandemic, people are shopping online more. As an outcome, there was a huge amount of reviews published on the Amazon website. that makes new consumers to spent time on reviews to identify the most related reviews of the product. Reviews with mismatched ratings may lead to a negative customer experience. Because of that, this project focused on classifying reviews and making rate prediction based on the reviews to overcome this problem, and this can be used to avoid the review analyzing part for users. Both Convolutional Neural Network and Multinomial Naive Bays models were used in the reviews classification and rates prediction in the project. Video games and music instrument reviews on Amazon were used as the data set. This project illustrates accuracy, precision, recall, and f1 score variation among the learning rate, epochs in Neural Network and alpha value in Multinomial Naive Bays. The final results have been compared and Multinomial Naive bays perform well in this case. This project will support e-commerce industries like Amazon to manipulate customer reviews to understand customer experiences.**

I. Introduction and related work

The huge volume of review manipulation is a challenging problem for platforms such as Amazon. This project aims to predict the ratings based on the Amazon text reviews, and categorize the reviews into two categories such as video games and musical instruments. The online reviews and rating system was introduced by Lakermair, Kailer, and Kanmaz(2013) [4]. This rating system is helpful for online shopping customers to make their decision beforehand. When it comes to online shopping, users can't check the actual product, to get the overall idea about the products in the market with different brands. Then customers have to go through the reviews of each product. It is not efficient at all. Using this system, it is easy to select a product based on the star rating system that improves the customer experience. Using a review category easy to identify the inappropriate reviews under the product as well. It helps the customers to focus on the product.

I have come to cross the BERT, fastText, and the catBoost classification models in the article which is written by Nicholas Chu (Chu, 2021). It is titled "Amazon Review Rating Prediction with NLP". Few regression models were also used in the implementation such as LightGBM, CatBoost, ReLU with the neural network, 1D Convolution Layer, LSTM/GRU model, and BERT [1]. Apart from that, the "Predicting the ratings of reviews of a hotel using Machine Learning" blog describes the implementation of a Neural network to predict the ratings which are written by B. Shiv Kumar (Kumar, 2021) [3].

Amazon reviews data contain 32918 reviews with unwarranted data. To clean the data, I have followed pre-processing steps such as removing stop words, punctuations, Html related tags, web URLs, special characters, numbers, and emails. Moreover, I have used text formatting, tokenization, and lemmatization as well. As a next step, I split the dataset into three sets and sampled the test and validation sets. I have chosen classification algorithms to classify both product catalogue and review scores. Neural Networks are better for this type of classification. I have selected this Neural Networks model because of its efficiency and easiness [2]. Apart from that, this model gives quality and accurate results. The Neural Network can build a model using complex relationships. Moreover, this model is useful for pattern recognition, and it has simple gradient-based learning. Another model that I have used for classification is Multinomial Naive bays. This model is mostly used for text classification. I have chosen the Multinomial Naive Bays model because of its ability to handle large data sets and scalability [6]. I have used the learning rate (lr) and the number of epochs to experiment with the Neural Network model. For the Multinomial Naive Bays, I have variated the alpha parameter to test the model.

II. Ethical discussion

Considering the growing online market, most people tend be shopping through the website rather than going to shopping malls. In that case, online reviews, and the ratings on the product play an important role in online shopping. Amazon is a prime source of shopping in Europe and America. Reviews and ratings are certainly needed to determine the quality of the product as a state on amazon. However, these aren't accurate all the time. Users can add any review on the product, some may experience the best side of it, and some may not see it as others. Most of the time reviews are based on the personal preferences of the consumers. Consumers can add unrelated reviews as well. That can happen intentionally or the un intensionally. Since the majority of the incoming consumers depend on the reviews and ratings on the product, ML can help to determine the unbiased rating based on the reviews of the product. It can help to avoid the most related reviews with the product.

Since Amazon has a wide variety of product categorizations, this research project only focused on two categories such as video games and musical instruments. This narrow-down scope is determined based on the complexity and the timeline of the project. Even though the algorithm is capable of handling a wide variety of categories.

This algorithm is allowing consumers to focus on shopping instead of following the reviews on the product. This algorithm is capable of giving unbiased output on the product and reducing the time wastage of the users. The training data set is balanced by applying oversampling, which is likely to decrease any existing biases. There might still be some imbalanced data left in the set.

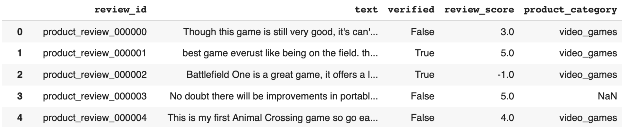
Considering the final output of the implemented algorithm every product should get a rating between 1 and 5 according to the related reviews to the product.

III. Dataset preparation

Amazon reviews data taken from google drive that is located below the URL. https://drive.google.com/file/d/10oVcDAopVEIHzcwU90bRXfVwBzSxLdSX/view?usp=sharing. This data contains 32918 reviews with 5 attributes. Each Amazon product review contains the following attributes such as,

- review_id: unique id of the review
- text: The text review
- verified: Status of verification of the review
- review_score: The rating value for the review
- product_category: The category of the product reviews

The first five records in the dataset are shown in 1.



	review_id	text	verified	review_score	product_category
0	product_review_000000	Though this game is still very good, it's can'...	False	3.0	video_games
1	product_review_000001	best game everust like being on the field. th...	True	5.0	video_games
2	product_review_000002	Battlefield One is a great game, it offers a l...	True	-1.0	video_games
3	product_review_000003	No doubt there will be improvements in portabl...	False	5.0	NaN
4	product_review_000004	This is my first Animal Crossing game so go ea...	False	4.0	video_games

Fig. 1. First five records

The Amazon reviews data have many unnecessary data. To make it meaningful, data need to be clean. As a first step, I observed the dataset and identified the mismatch and unwarranted data. I have checked column names to check what are the attributes that I needed for this project as input and output. I have chosen 'text' attributes as input and the 'review_score', and 'product_category' as targets. I have also checked the attribute's data types to check whether they are matching with the value, and I fixed the "review_score" attribute datatype from float to int. Furthermore, I have got the statistical data such as mean, median. to get an idea about how the 'review_score' is distributed. And I have got the count of each categorical data to recognize inappropriate category types. In the 'review_score', I have discovered the '-1' category but it is an invalid data category. I have removed those rows in the data set. Furthermore, I checked whether is there any duplicates, but I didn't catch anyone. As the input of our model, text reviews have to be cleaned as well. I have examined the text data in deep to identify the pattern and its behaviour. Figure 2 shows the bigrams for the text data for each product category.

The word cloud represents the most frequent words in the dataset. Figures 3 and 4 show the word cloud graph for each product category.

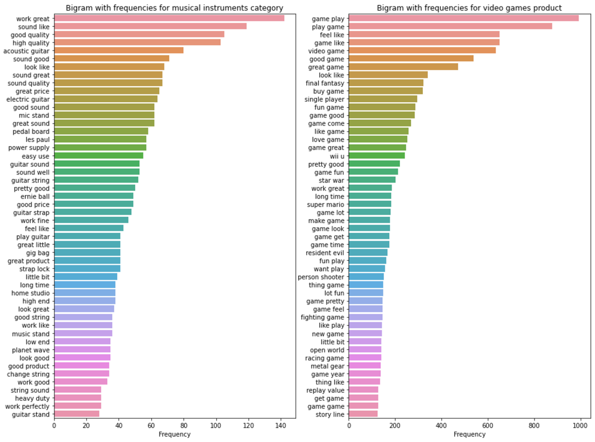


Fig. 2. The word frequencies

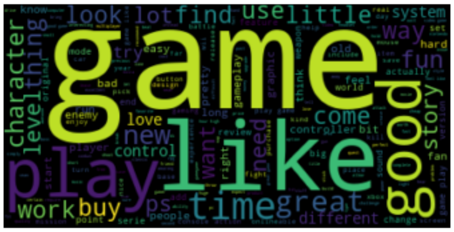


Fig. 3. The word cloud graph for the video game product category

And the histogram shows how the frequency is variate among the review scores. Figure 5 shows the histogram of each product category. To avoid the case sensitivity, I have converted the text into lowercase. There are some short-form words in the reviews such as "I'm", "can't". I have converted them into meaningful words such as "I am", "cannot". using regex. After that, I tokenized the text data and removed stop words and punctuations using Spacy because those words are not added much information to the reviews. Then I used lemmatization to get the word into root format using Spacy which makes the text data normalization. Then vocabulary size can be reduced. After lemmatization, I have joined all tokens as a text. Then I removed Html related tags, web URLs, special characters, numbers, and emails that do not give sense to the document. After cleaning the dataset, I handled the missing value to ensure is there any empty values in the dataset. Then I split the dataset into three sets such as train, validation, and

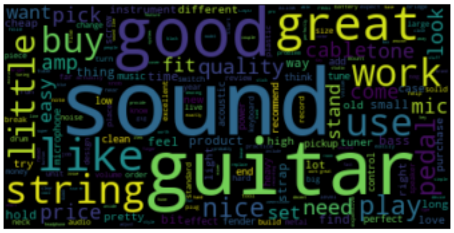


Fig. 4. The word cloud graph for the musical instrument product category

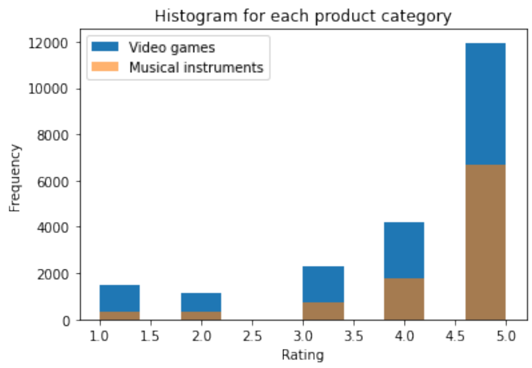


Fig. 5. The histogram of each product category

test data. For the train and validation sets have imbalanced target values. For both ”review_score” and ”product_category” features, I have upsampled using RandonOverSampler and SMOTE in the ”imblearn” library. And also, I have encoded the ”review_score” and ”product_category” using Label Encoder to make all target attributes numerical. Finally, the data is cleaned and ready to feed the ML model. Figure 6 is shown the pre-processing step that I have used in this project.

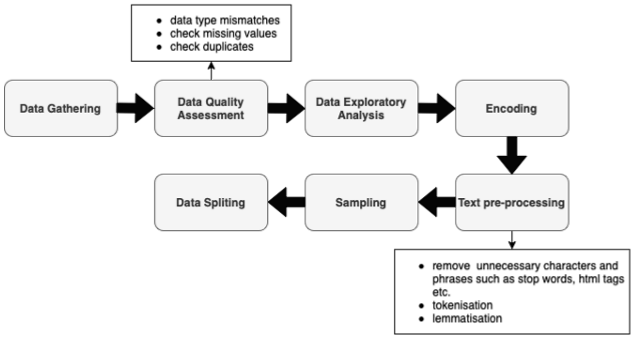


Fig. 6. Pre-processing steps

IV. Methods

To predict the product catalogue and review score I have used Neural Network (NN) and Multinomial Naive Bays (MNB) models. The Neural Network model I have used with batch size 258 and that number of samples passed to neural network at a time. The larger batch size makes the process faster. Before feeding texts into the model, have to convert texts into integer values.. To do so, I have used TextVectorization in the TensorFlow library. It truncated the words to the given output sequence length and then I have retrieved the vocabulary. After creating the dictionary mapping, I created an Embedding layer for word embedding using Spacy and the TensorFlow. Then I have created 4 layered Conv1D and the output is flattened using the GlobalMaxPooling1D layer at the end. These Conv1d layers have the capability to analyse a bunch of words at once and it improves the context. And also, I have used Maxpooling1D to focus on important features. Then the result is fed to the Dense layer with 5 units for review score prediction and 2 units for product catalogue prediction. As the activation function, I have used the ”softmax” activation function because it is suitable for multi-class classification tasks. ”sparse_categorical_crossentropy” uses as loss because the activation function and label are encoded as integer values. The equation of the Neural network is shown in equation (1).

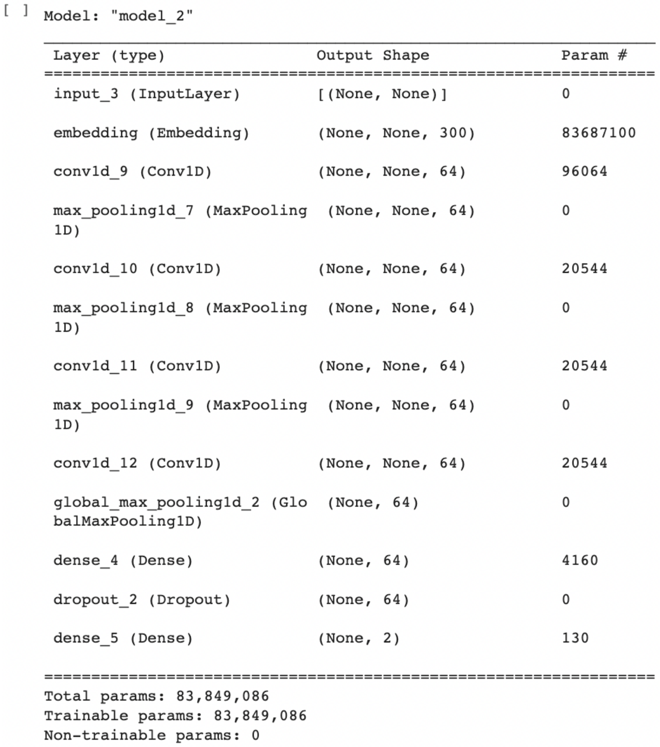
$$\boldsymbol{y \;= f(X_n W_n + b_n)} \tag{1}$$

The X is the input matrix, W represents the weight matrix and y represents the output. Equation (2) represents the softmax function equation.

$$\boldsymbol{softmax(z_k) = \frac{exp(z_k)}{\sum_{c=1}^{c} exp(z_c)}} \tag{2}$$

Figure 7 shows the Neural Network model that I have used for the product category prediction. Because of the accurate and quality results I have chosen this model. To implement the Neural Network, I have used the TensorFlow library.

Multinomial Naive Bays are also popular in text classification.



```
[ ]  Model: "model_2"
     _________________________________________________________________
      Layer (type)                Output Shape              Param #
     =================================================================
      input_3 (InputLayer)        [(None, None)]            0

      embedding (Embedding)       (None, None, 300)         83687100

      conv1d_9 (Conv1D)           (None, None, 64)          96064

      max_pooling1d_7 (MaxPooling  (None, None, 64)         0
      1D)

      conv1d_10 (Conv1D)          (None, None, 64)          20544

      max_pooling1d_8 (MaxPooling  (None, None, 64)         0
      1D)

      conv1d_11 (Conv1D)          (None, None, 64)          20544

      max_pooling1d_9 (MaxPooling  (None, None, 64)         0
      1D)

      conv1d_12 (Conv1D)          (None, None, 64)          20544

      global_max_pooling1d_2 (Glo  (None, 64)               0
      balMaxPooling1D)

      dense_4 (Dense)             (None, 64)                4160

      dropout_2 (Dropout)         (None, 64)                0

      dense_5 (Dense)             (None, 2)                 130

     =================================================================
     Total params: 83,849,086
     Trainable params: 83,849,086
     Non-trainable params: 0
```

Fig. 7. Neural Network model

Considering the frequency of each word, this model does a better job of predicting the text analysis. In the Multinomial Naive bays, I have used a TF-IDF vectorizer with unigram and bigram. It illustrates how relevant each word is to the review and it makes the frequencies of each unique word. For the text reviews, I applied TF-IDF vectorization to the text review and prepared the input of the model. The product catalogue and review score are targets. I have predicted the

product catalogue and review score separately.
Equation of the Multinomial Naive Bays is shown in equation (3)

$$P(c|X) = \frac{P(x|c)P(c)}{P(x)} \tag{3}$$

Multinomial Naive Bayes is easy to implement and consumes a less amount of time to predict, compared to most other classifications. It gives better performance for low volume data as well as high volume data. I have proposed this technique to be used here, because of the better performance. To create the Multinomial Naive Bayes model, I have used the MultinomialNB module in sklearn.naive_bayes library.

V. Experiments and Evaluation

As for the evaluation metrics, I have chosen precision, recall, fi score and accuracy. The confusion metrics illustrate the performance of the classification model. Accuracy is the most used metric for performance evaluation. But considering the accuracy may lead to the wrong decision because the accuracy metric is not good for evaluating the imbalanced data. The precision metric shows how precise our model is for detecting positive class and how useful. The recall shows the percentage of given reviews is predicted correctly according to the total number of reviews. The f1 score is a combination of precision and recall and represents their harmonic mean. And it measures incorrectly classified classes much better [5]. Table I illustrates the parameters that I have used in both Neural Network and Multinomial Naive Bays.
Using the mentioned parameters in table I, I have got below results for the Neural Network model as shown in figure 8, figure 9, figure 10, and figure 11 .

TABLE I
Hyper-parameter list of each machine learning model

Machine Learning models	Parameter Name	Value
Neural Network	Batch size	258
	Layers	12
	Epochs	10
	lr	0.05
Multinomial Naive Bays	alpha	0.6

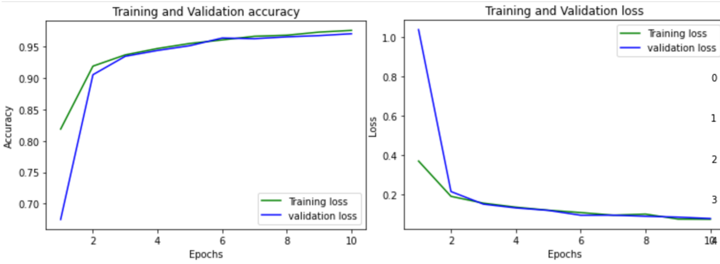


Fig. 8. The accuracy and loss behaviour with respect to the epochs in product catalogue prediction using Neural Network

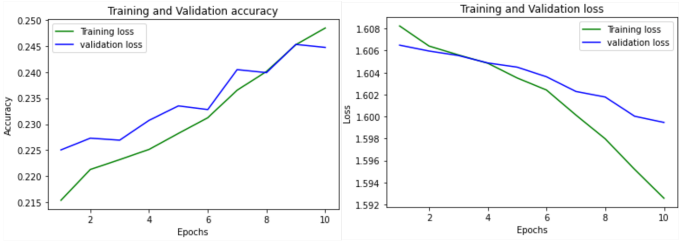


Fig. 9. The accuracy and loss behaviour with respect to the epochs in review score prediction using Neural Network

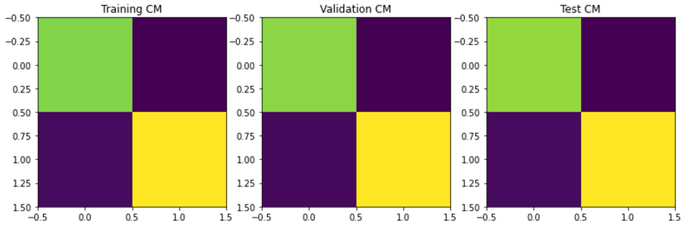


Fig. 10. The confusion matrix of the review score prediction using Neural Network

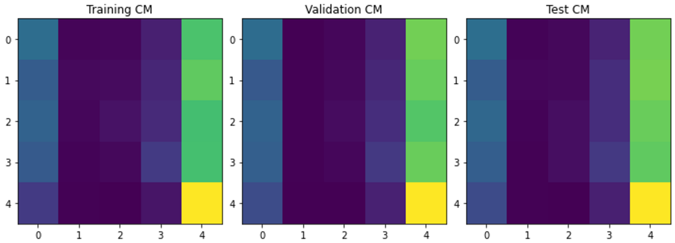


Fig. 11. The confusion matrix of the product catalogue prediction using Neural Network

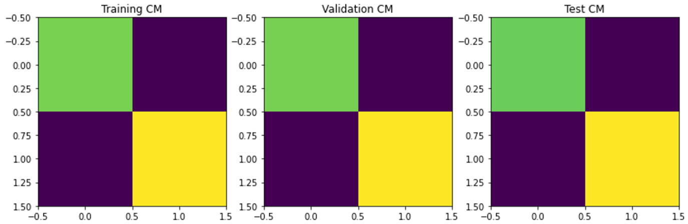


Fig. 12. The confusion matrix of review score prediction using Multinomial Naive Bays

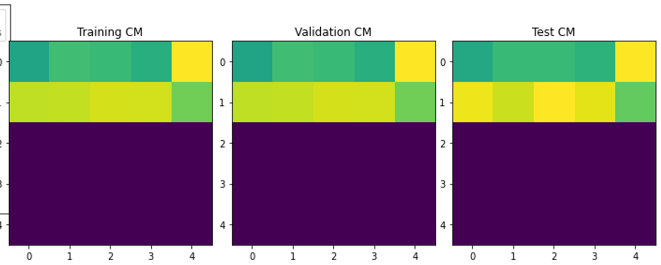


Fig. 13. The confusion matrix of the product catalogue prediction using Multinomial Naive Bay

For the Multinomial Naive Bays, I have got results that are shown in figure 12 and figure 13.

In the Neural Network, I have changed the learning rate and epochs to fine-tune the model. Below, Table II shows how precision, recall, accuracy and f1 score are variate with respect to parameters. The table II shows the exact parameter that I have changed during each experiment by taking the base parameters from table I.

TABLE II
RESULTS OF TEST DATASET FOR EACH MACHINE LEARNING MODELS

Model	Param	Value	Accuracy	Precision	Recall	F1 score
NN for category	Epoch	10	0.9659	0.9648	0.9671	0.9657
	Epochs	20	0.9737	0.9726	0.9744	0.9734
	Epochs	50	0.9780	0.9768	0.9790	0.9778
	Lr	0.01	0.9394	0.9376	0.9432	0.9390
	Lr	0.05	0.9659	0.9648	0.9671	0.9657
	Lr	0.1	0.9737	0.9726	0.9744	0.9734
NN for review score	Epochs	10	0.2430	0.2469	0.2395	0.1844
	Epochs	20	0.2510	0.2701	0.2517	0.2336
	Epochs	50	0.2550	0.2608	0.2552	0.2480
	Lr	0.01	0.2228	0.2205	0.2214	0.1760
	Lr	0.05	0.2430	0.2469	0.2395	0.1844
	Lr	0.1	0.2247	0.2043	0.2248	0.1638
MNB for category	alpha	100	0.7102	0.8215	0.6793	0.6603
	alpha	10	0.9412	0.9487	0.9359	0.9400
	alpha	1	0.9837	0.9837	0.9833	0.9835
MNB for review score	alpha	100	0.0547	0.1732	0.0204	0.0363
	alpha	10	0.1467	0.1857	0.0574	0.0877
	alpha	1	0.1673	0.1846	0.0660	0.0973

In the Multinomial Naive bays, the hyper-parameter is alpha is used for Laplace smoothing and it avoids the zero probability. When alpha increases it towards the uniform distribution and higher alpha values are not recommended because it was more biased toward the class that has more records. This may lead to underfitting. The Multinomial Naive bays models give accurate results with smaller alpha values. Hence the smaller alpha value is better to achieve accurate results. Moreover, the Multinomial Naive Bays model gives results faster compared to the Convolutional Neural Network. According to the results, alpha with 1 is more suitable for Multinomial Naive Bays for this project.

Table II shows a few examples that how evaluation metrics changes while changing the learning rate and epochs. The learning rate handles how quickly achieve the results. The small learning rates need more epochs because weights are updated with smaller changes. As well as larger learning rates require few epochs. Figure 14 shows a variation of the loss with respect to the learning rate. As per the figure 14, when the learning rate is too low, the loss does not show significant changes. And the learning rate is too high it is beginning to diverge. So, the best learning rate range is marked in the dotted line in the figure 14.

Same as when increasing the epochs, the model goes from underfitting to optimal stage and then converges to overfitting. As per the results in Table II, the best learning rate is 0.1 when the epoch is equal to 10 and the best number of epochs is 50 When the learning rate is equal to 0.05 for predicting the category. In the review score prediction, the best values for learning rate and epoch are 0.05 and 50.

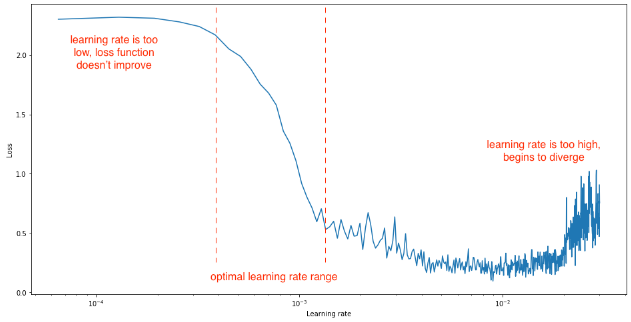


Fig. 14. The loss variation with respect to the learning rate. [7]

VI. DISCUSSION AND FUTURE WORK

According to the results in table II, Multinomial naive bays is the best model due to their high accuracy and F1 score. Both Neural and Naive Bays work well in the classification of the product category. But those models did not give good performance on the review score classification. In review score classification has to classify reviews into 5 groups. Due to lack of data for each category may lead to bad performance. This can be avoided by feeding more data to the model. According to the performance and time consumption, Multinomial Naive Bays is the best model compared to the Neural Network for Amazon review manipulation. This project target only two product categories, and this can be increased to classify more categories in future. As well as the project targets the Amazon reviews, but this project can be extended to make a classification based on other platforms. In this research, I have used only two machine learning modules and in future research, I will explore more efficient machine learning models to build the best ML model.

VII. CONCLUSIONS

In this research, I propose machine learning (ML) techniques to classify Amazon reviews by their score and product category. Reviews are collected from Amazon, and I have preprocessed data to remove unnecessary parts. To preprocess I have used tokenization, lemmatization, stop word, HTML tags, URL, and unnecessary character removal techniques. Preprocessed data was fed to the ML models and trained using a Convolution neural network and Multinomial Naive Bays. Then fine-tune the model using validation data sets and make the model more accurate. According to the performance of the test data, Multinomial Naive bays gives more accurate and precise results. In a summary, I have demonstrated how to build machine learning models with reviews and product category information. This research can be useful to solve the problem of identifying the review rating mismatch submitted by the customer.

REFERENCES

[1] Chu, N. (2021). Amazon Review Rating Prediction with NLP. [online] Data Science Lab Spring 2021. Available at: https://medium.com/data-science-lab-spring-2021/amazon-review-rating-prediction-with-nlp-28a4acdd4352 [Accessed 21 Apr. 2022].

[2] Imran, M. (2020). Advantages of Neural Networks - Benefits of AI and Deep Learning. [online] Folio3 AI. Available at: https://www.folio3.ai/blog/advantages-of-neural-networks/ [Accessed 21 Apr. 2022].

[3] Kumar, B.S. (2021). Predicting the ratings of reviews of a hotel using Machine Learning. [online] Analytics Vidhya. Available at: https://medium.com/analytics-vidhya/predicting-the-ratings-of-reviews-of-a-hotel-using-machine-learning-bd756e6a9b9b [Accessed 21 Apr. 2022].

[4] Georg Lackermair, Daniel Kailer, Kenan Kanmaz, ”Importance of Online Product Reviews from a Consumer’s Perspective,” Advances in Economics and Business, Vol. 1, No. 1, pp. 1 - 5, 2013. DOI: 10.13189/aeb.2013.010101.

[5] Oliveira, M. (2021). Evaluating classification models with Accuracy, Precision and Recall. [online] Medium. Available at: https://medium.com/@marciojmo/evaluating-classification-models-with-accuracy-precision-and-recall-66f552380028 [Accessed 22 Apr. 2022].

[6] Shriram (2021). Multinomial Naive Bayes Explained: Function, Advantages Disadvantages, Applications in 2021. [online] upGrad blog. Available at: https://www.upgrad.com/blog/multinomial-naive-bayes-explained/ [Accessed 21 Apr. 2022].

[7] Jordan, J. (2018). Setting the learning rate of your neural network. [online] Jeremy Jordan. Available at: https://www.jeremyjordan.me/nn-learning-rate/ [Accessed 24 Apr. 2022].