Review-Rating Prediction for Crawled Amazon Comments

Amir Bahador Eizadkhah
MS. Applied Computer Science
40165791
a_eizadk@live.concordia.ca

Amin Nazarian Saralang
MS. Electrical and Computer Eng
40219812
am_naza@encs.concordia.ca

***Abstract*— Text mining and NLP have become one of the most exciting topics in the Machine Learning area, focusing on the interaction between Artificial Intelligence and Human Language.**

The primary purpose of the following project is to predict the rates of comments submitted by people who purchased the item on Amazon. Five different famous classifiers do the projects to test and measure the classifiers' accuracy when they face actual, live data. The models are designed to process the text as the matrix of features, and the targets are rates in the range of one to five.

Keywords—NLP, Machine Learning, Web Scraping, PyTorch, LSTM, Sk-Learn, Deep Learning, Rate Prediction

I. INTRODUCTION

In this project, we have a sequence of vocabulary as a sentence that shows the review or comment of a customer that has been written on the bottom of the good on the shopping website. Also, we get the data set for training our models from the GitHub website. The primary purpose of our project was to mine the main mean or emotions that the customers had when they gave their comments, and it is essential to analyze the value of the performance of an online shop. For instance, we used a data-set that deals with one kind of cellphone and accessories. When one person comments and says: "this cellphone did not work properly," the rate they give is, for example, "2". After our models' training, we expected that the models could predict this rate of the comment.

II. METHODOLOGY

In order to start solving the aforementioned problem, some vital steps need to be taken.

A. Dataset

Our Dataset is derived from the GitHub, which is about Cellphones and Accessories [1] which has .json suffix which converted to .csv in code. Generally, the dataset has exactly 9 columns such as Id, helpful, ReviewerName, summary, Score, Time, Text, etc. But the essential columns for the prediction were just Text and Score. In order to capture the two aforementioned columns, they were copied into a new Pandas Data Frame.

The initial dimensionality of the new data frame was 568,454 by 2. In details, the dataset has 568,454 unique Texts and Labels. The highest number of registered votes is five, and the lowest is one. In other words, the range of the labels is integer numbers from one to five. After that, all Null rows were eliminated.

	TEXT	SCORE
0	unlock phone use att tmobil phone differ chips...	5.0
1	well end go back old case reason fault case sa...	5.0
2	order case new dna debat sever differ case arr...	5.0
3	case bulki seem protect phone well sinc drop s...	5.0
4	love cover beauti blingi look great iphon incl...	5.0

Fig. 01
Example of the dataset

B. Web Scraping

Web Scraping was one of the primary parts of the project. To do so, Beautiful Soup 4 (BS4) was used which is one of the famous libraries in web crawling and scraping. In order to run the crawler, Browser Agent with the last version of all browsers is needed. With a little explore into the Amazon website, the tags of Reviews and Rates can be found. Reviews are under the ‘span’ tag and ‘data-hook’. Besides, Rates are under the ‘i’ tag in the ‘data-hook’. The context of the Reviews is in the ‘review-body’ and the amount of the Rates are in the ‘review-star-rating’. Initially, Rates are in the String data type which need to cats into Integer.

Then we can capture the values with the pseudocode written below.

```
SET X_tst TO []
SET y_tst TO []
SET FLAG TO True
SET checking TO 'https://www.amazon.'

WHILE FLAG:
```

```
SET link TO INPUT('Please put your Amazon link here: ')
IF checking IN link:
   SET FLAG TO False
ELSE:
      OUTPUT('Your website either is not Amazon or has not
        observed "HTTPS" legislation.')

with requests.Session() as session:
  FOR iterator IN range(400):
    FOR rates IN getRate(session, link + str(iterator)):
      y_tst.append(f'{rates.text}\n'[0:3])
    FOR reviews IN getReviews(session, link + str(iterator)):
      X_tst.append(f'{reviews.text}\n\n'[4:])

SET Data Frame TO {'TEXT': X_tst, 'SCORE': y_tst}
```

Having said that, the amazon product that we crawled is Apple iPhone XR with 4,990 reviews in total [2].

After crawling, the final shape of the crawled data frame was 4,990 by 2. In Comparison to the amount of the dataset, the amount of data that obtained by crawling is extremely low. In order to make a balance between new data frame with the base dataset, we just need to shatter the primary dataset with using the below formula.

$$Train\ Sample = \frac{len(Crawled\ data)\ \times\ 0.8}{0.2}$$

Train sample is composed of the specified number of rows from the primary dataset which is selected randomly. At the end, the amount of obtained data was appended at the end of the Train sample. The shape of the final dataset was 24,950 by 2.

C. Data Preprocessing

Data preprocessing (in this case, data is Text) is assumed to be the most important part of this project. Because it is exactly one step before classifying the data. Therefore, we tried to perform data preprocessing operations with great care to obtain acceptable results.

1) Data Cleaning

First, the text clearing operation was performed. In this section, all the letters were changed to lower case. Many people today use emojis to express their emotions. So, any emojis might have extra weight, so it can lead to a wrong prediction. So, all the emojis were eliminated. Then additional characters such as commas, slash, parentheses, etc., were removed from our sentences.

Then it was time for an unforeseen event to encounter non-English comments. Two basic strategies were considered, one of which was eventually chosen. Number one was to delete all of these comments, and in this case, approximately a quarter of the crawled comments from Amazon would have to be deleted.

The second was to translate sentences. Also, a lot of time would have to be spent on this part too. As far as all comments were considered necessary, the first strategy was omitted. It was decided that all comments should first be sent to a language detector and translated if the text was non-English.

In the next level, all of the English Stopwords were deleted. Stop words are a set of commonly used words in any language. For example, in English, “the”, “is” and “and”, would easily qualify as stop words. In NLP and text mining applications, stop words are used to eliminate unimportant words, allowing applications to focus on the important words instead.

At the end all of the words and verbs converted to their main root. For instance, the word of ‘translation’ converted to ‘translate’. Also, if the user writes a word mistakenly, it will be corrected. For example, ‘goood’ will be turned to ‘good’ which is the correct form.

2) Vectorization

While working with Text and NLP, Vectorization always should be considered as an essential part. In this project TF-IDF vectorizer is used.

Term frequency-inverse document frequency is a text vectorizer that transforms the text into a usable vector. It combines 2 concepts, Term Frequency (TF) and Document Frequency (DF). The term frequency is the number of occurrences of a specific term in a document. Term frequency indicates how important a specific term in a document. Term frequency represents every text from the data as a matrix whose rows are the number of documents and columns are the number of distinct terms throughout all documents.

Document frequency is the number of documents containing a specific term. Document frequency indicates how common the term is.

Inverse document frequency (IDF) is the weight of a term, it aims to reduce the weight of a term if the term’s occurrences are scattered throughout all the documents

The *max_feature* of the vectorizer is set to 6,500, which would mean creating a feature matrix out of the most 6,500 frequent words across text documents. Also, Bi-Gram strategy is used.

3) Oversampling

After plotting the dataset with MatplotLib, we faced with an imbalanced dataset. As it shows, the number of people who voted five star is far more than the others. So, the data set is

not balanced. The strategy is oversampling the data to make a good balance among rates.

Oversampling is the practice of selecting respondents so that some groups make up a larger share of the survey sample than they do in the population. Oversampling small groups can be difficult and costly, but it allows polls to shed light on groups that would otherwise be too small to report on

To do so, **imblearn** library is used. After the operation, the dimensionality of the matrix of features is 72,100 by 6,500.

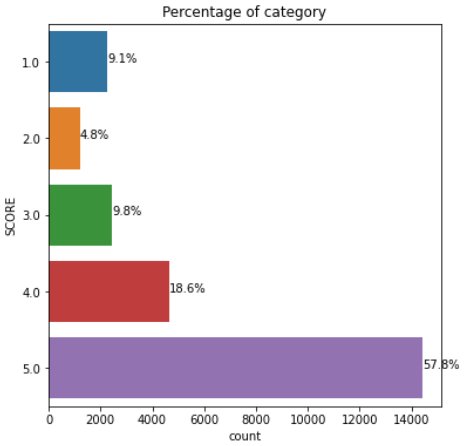

Fig. 02
Shape of imbalanced Dataset

Also, after getting the word cloud, we can see the most used words in the dataset. For instance, 'use', 'case', 'great', phone. The size of the words indicates the feature importance.

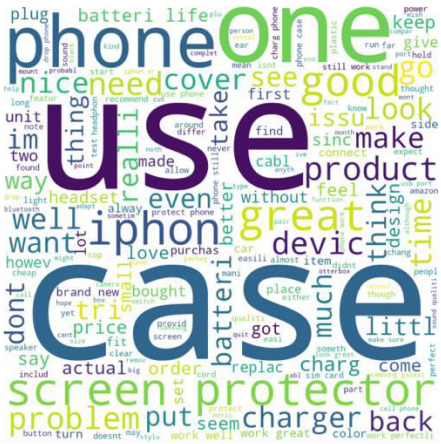

Fig. 03
Dataset's Word Cloud

4) Train-Test Split

Sk-learn's *model_selection* library is used to split the Train and Test set, sk-learn's model_selection library is used. Only %20 of the dataset is dedicated as a Test set. So, %80 of the whole dataset is consequently considered a Train set. Note that the dataset must be shuffled at the end. The final shapes if the Train set and test set are 57,680 by 6,500 and 14,420 by 6,500 respectively. Also, the *random_state* is set to 42.

D. Classification

In the following section, all of the classification techniques that used in this project are described in detail. The classifications are done by five different popular classifiers with Sk-Learn and PyTorch:

- Gaussian Naïve Bayes
- Support Vector Machine
- Long-Short Time Memory (LSTM)
- K-Nearest Neighbors
- Random Forest

1) Gaussian Naïve Bayes

Gaussian Naïve Bayes with the Sk-Learn has only one hyper parameter which is called *var_smoothing* which is a portion of the largest variance of all features that is added to variances for calculation stability. This value is set 1e-3.

2) Support Vector Machine

In this case we used **L2 regularization** and its parameter, C is equal to 100 which must be strictly positive. Also, the *random_state* is set to *zero* and the model will iterate for 10000 times.

3) Random Forest

In case of working correctly, **GINI** indexing [3] is used with *random_state=0*. Also, the model reuses the solution of the previous call to fit and add more estimators to the ensemble. This means, the *warm_start* is equal to zero

$$gini\ index = 1 - \sum p(x = k)^2$$

4) K-NN

For the model, *n_neighbors* is set to two and distance metric is considered as *minkowski* with *p-value* of two.

$$minkowski\ distance = (\sum_{i=1}^{n} |x_i - y_i|^p)^{\frac{1}{p}}$$

5) Long-Short Term Memory (LSTM)

As far as our range of labels are between 1 to 5, so we have five number of classes. 40 Batches are considered for training data. For the Validation set only eight batches are set.

As mentioned above, the RNN model has trained with 25 epochs with learning rate 0.002 and *hidden_size* 256. Embedding dimension is set on 128 and Vocab_Size 30000. Furthermore, the *input_size* of the LSTM is equal to the length of matrix of features vocabs for train data. It is obviously that the output_dimension would be equal to the number of classes.

III. SUCCESS MEASURES

In order to evaluate how far the model was successful, four famous methods were used as follows.

A. *Confusion Matrix*

A Confusion Matrix [4] is a summary of prediction results on a classification problem.

The number of correct and incorrect predictions are summarized with count values and broken down by each class. This is the key to the confusion matrix.

The confusion matrix shows the ways in which your classification model is confused when it makes predictions.

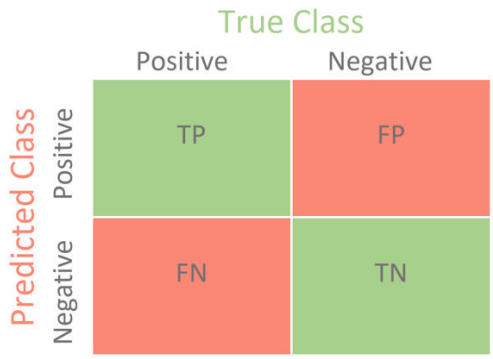

Fig. 04
Confusion Matrix

B. *Mean Squared Error*

The **mean squared error** (MSE) tells you how close a regression line is to a set of points. It does this by taking the distances from the points to the regression line (these distances are the "errors") and squaring them. The squaring is necessary to remove any negative signs. It also gives more weight to larger differences. It's called the **mean** squared error as you're finding the average of a set of errors. The lower the MSE, the better the forecast. [5]

C. *F1-Score*

The F1-score combines the precision and recall of a classifier into a single metric by taking their harmonic mean. It is primarily used to compare the performance of two classifiers. Suppose that classifier A has a higher recall, and classifier B has higher precision.

$$f1_{score} = \frac{(\beta^2 + 1)precision \times Recall}{\beta^2 \times Precision + Recall}$$

1) *Precision*

Precision (also called positive predictive value) is the fraction of relevant instances among the retrieved instances. [6]

$$Precision = \frac{True\ Positive}{True\ Positive + False\ Positive}$$

2) *Recall*

recall (also known as sensitivity) is the fraction of relevant instances that were retrieved. Both precision and recall are therefore based on relevance. [6]

$$Recall = \frac{True\ Positive}{True\ Positive + False\ Negative}$$

D. *Accuracy Score*

In multilabel classification, this function computes subset accuracy: the set of labels predicted for a sample must exactly match the corresponding set of labels in y_true. [7]

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN}$$

IV. EXPERIMENT RESULTS

After classifying the dataset with five different models, it is time to review the results of each classifier based on their Accuracy Score.

A. *Acceptance Condition*

The Accuracy Score is considered as the main condition to accept the model. If the accuracy of predicted class labels is less that %75, the model is considered as a failed model.

B. *Results*

Now, it is time to review the results.

1) *Gaussian Naïve Bayes*

Measure	Amount
Accuracy Score	%70

F1-Score	%68.9
MSE	1.274

TABLE I
Metrics of NB

$\hat{y}$ \ *y_true*	**1**	**2**	**3**	**4**	**5**
1	2602	148	79	40	70
2	95	2681	63	36	22
3	371	302	1916	181	90
4	574	257	381	1333	282
5	194	185	338	624	1556

TABLE II
Confusion Matrix of NB

2) K-NN

Measure	**Amount**
Accuracy Score	%77
F1-Score	%71.7
MSE	1.230

TABLE III
Metrics of K-NN

$\hat{y}$ \ *y_true*	**1**	**2**	**3**	**4**	**5**
1	2789	0	21	112	17
2	2	2895	0	0	0
3	44	0	2679	133	4
4	137	1	79	2598	12
5	731	14	320	1619	213

TABLE IV
Confusion Matrix of K-NN

C. LSTM

Measure	**Amount**
Accuracy Score	%78
F1-Score	%78
MSE	0.469

TABLE V
Metrics of LSTM

$\hat{y}$ \ *y_true*	**1**	**2**	**3**	**4**	**5**
1	25	2	2	0	0
2	1	5	1	0	0
3	1	1	3	1	1
4	0	0	0	4	1
5	0	0	0	0	1

TABLE VI
Confusion Matrix of LSTM

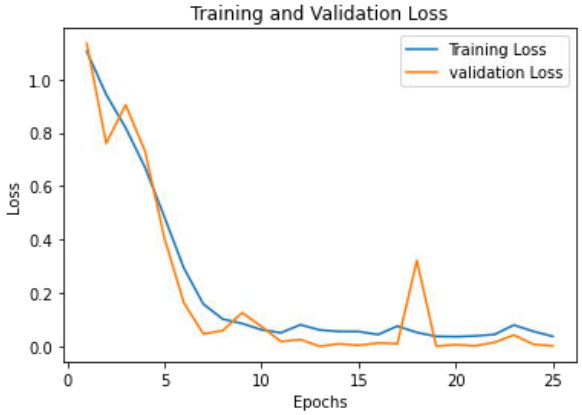

Fig. 01
Train and Validation Loss of LSTM

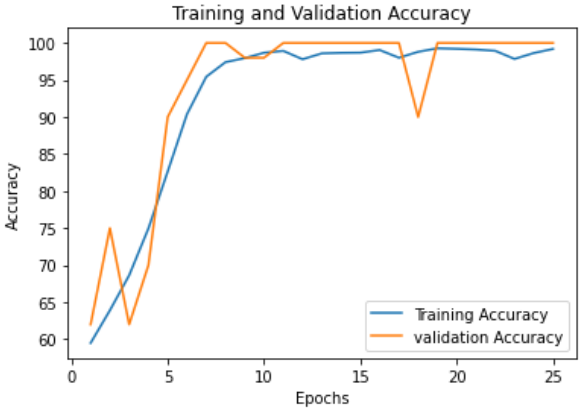

Fig. 02
Train and Validation Accuracy of LSTM

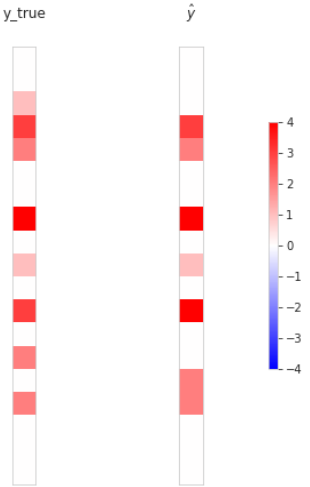

Fig. 03
Comparing the results of y_true and ŷ

D. Linear SVC

Measure	**Amount**
Accuracy Score	%88
F1-Score	%87.4
MSE	0.477

TABLE VII
Metrics of Linear SVC

$\hat{y}$ \ *y_true*	1	2	3	4	5
1	2881	3	15	10	30
2	0	2892	3	2	2
3	7	4	2782	51	16
4	25	28	76	2297	401
5	169	92	300	524	1812

TABLE VIII
Confusion Matrix of LSTM

E. Random Forest

Measure	Amount
Accuracy Score	%92
F1-Score	%92
MSE	0.296

TABLE IX
Metrics of Random Forest

$\hat{y}$ \ *y_true*	1	2	3	4	5
1	2881	3	15	10	30
2	0	2892	3	2	2
3	7	4	2782	51	16
4	25	28	76	2297	401
5	169	92	300	524	1812

TABLE X
Confusion Matrix of Random Forest

F. Accuracy Score of all Models

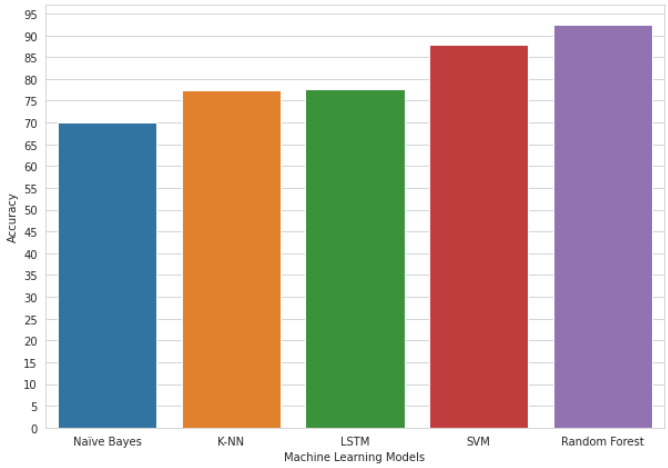

Fig. 04
Accuracy of all models

G. F1-Score of all models

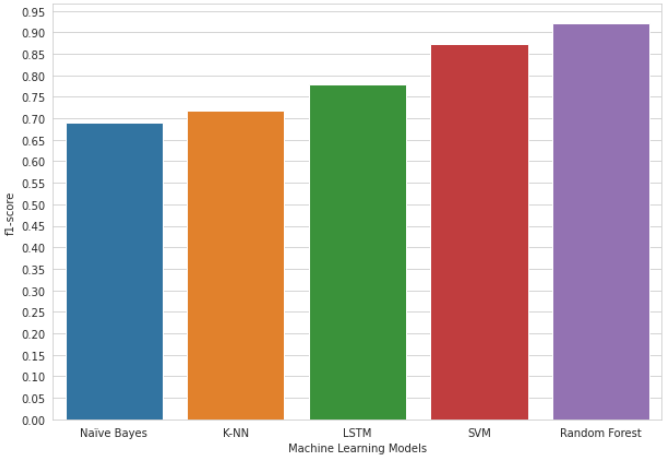

Fig. 05
F1-Score of all models

V. CONCLUSION AND FUTURE WORKS

In this project we focused on addressing the issue with regard to predicting the right rate for a comment using Cell phones and Accessories dataset that is downloaded from GitHub and predict Amazon comments based on the trained dataset.

First, we started with importing Libraries such as Numpy, BS4 and so on. Second, the dataset was imported and after that, the iPhone XR's comments from amazon was crawled. Then, Data Preprocessing part begins with cleaning text and finally send it to tf-idf vectorizer and train-test split which were the final part just before classification.

After classifying the data, it was obvious that almost all models worked well. It was also observed that all models are highly sensitive to data changes because the results changed after each time when the project was executed. Therefore, it is recommended that the crawled data and dataset be in one category. For more curiosity, the data used for LSTM was without resampling. Surprisingly, the result was good but fragile.

We have in mind to try with different and larger datasets with different categories such as Foods, Musical Instruments, Books, etc. Also, Implementing Sentiment Analysis could be interesting.

Apart from Review-Rate prediction, we also intend to enter into areas such as recognizing immoral content or recognizing emotions

VI. RESOURCES

All codes, model weights parameters and resources can be found in GitHub. [8]

REFERENCES

[1] Ni, J., n.d. Amazon review data on Cellphones and Accessories. [online] Nijianmo.github.io. Available at: <https://nijianmo.github.io/amazon/index.html>.

[2] Amazon.com. 2022. iPhone XR. [online] Available at: <https://www.amazon.com/Apple-iPhone-XR-Fully-Unlocked/product reviews/B07P6Y7954/ref=cm_cr_arp_d_paging_btm_next_0?ie=UTF8 &reviewerType=all_reviews&pageNumber=0>.

[3] Medium. 2022. Entropy, Information gain and Gini Index; the crux of a Decision Tree. [online] Available at: <https://blog.clairvoyantsoft.com/entropy-information-gain-and-gini-index-the-crux-of-a-decision-tree-99d0cdc699f4?gi=b8896eb22e32#:~:text=Gini%20Index%3A%20It%20is%20calculated,is%20chosen%20for%20a%20split.> [Accessed 9 April 2022].

[4] Brownlee, J., 2022. What is a Confusion Matrix in Machine Learning. [online] Machine Learning Mastery. Available at: <https://machinelearningmastery.com/confusion-matrix-machine-learning/#:~:text=A%20confusion%20matrix%20is%20a%20summary%20of%20prediction%20results%20on,key%20to%20the%20confusion%20matrix> [Accessed 9 April 2022].

[5] MSE, M., 2022. Mean Squared Error: Definition and Example. [online] Statistics How To. Available at: <https://www.statisticshowto.com/probability-and-statistics/statistics-definitions/mean-squared-error/> [Accessed 9 April 2022].

[6] En.wikipedia.org. 2022. Precision and recall - Wikipedia. [online] Available at: <https://en.wikipedia.org/wiki/Precision_and_recall> [Accessed 9 April 2022].

[7] scikit-learn. 2022. sklearn.metrics.accuracy_score. [online] Available at: <https://scikit-learn.org/stable/modules/generated/sklearn.metrics.accuracy_score.html > [Accessed 9 April 2022].

[8] https://github.com/ebahador/NLP-With-Amazon-Crawling-Comments-with-LSTM