Gina Cody School of Engineering

Department of Computer Science and Software Engineering (CSSE)

PROJECT REPORT TITLE

Review_Rating prediction with crawling live data from Amazon

April 12, 2022

Machine Learning (COMP 6321)

LECTURER

Dr. Mirco Ravanelli

SUBMITTED BY

Amir Bahador Eizadkhah

MS. Applied Computer Science

40165791

a_eizadk@live.concordia.ca

eizadkhah.bit@gmail.com

Amin Nazarian Saralang

MS. Electrical and Computer ENG

40219812

am_naza@encs.concordia.ca

aminnazarian1987@gmail.com

Winter 2022

Table Of Contents

1. 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.

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

2. TOOLS AND PACKAGES

The vast majority of tools and packages are used in order to complete the project. Some of the essential libraries and tools are listed below.

2.1. Microsoft Excel

In order to explore the dataset, the MS Excel is used.

Microsoft Excel [1] is a spreadsheet developed by Microsoft for Windows, macOS, Android, and iOS. It features calculation or computation capabilities, graphing tools, pivot tables, and a macro programming language called Visual Basic for Applications. Excel forms part of the Microsoft Office suite of software.

This application also appropriate for read/write the .csv files.

2.2. Python 3

Python [2] is an interpreted, object-oriented, high-level programming language with dynamic semantics. Its high-level built-in data structures, combined with dynamic typing and dynamic binding, made it very attractive for Rapid Application Development and used as a scripting or glue language to connect existing components. Python's simple, easy-to-learn syntax emphasizes readability and, therefore, reduces program maintenance costs. Python supports modules and packages, which encourages program modularity and code reuse. The Python

interpreter and the extensive standard library are available in source or binary form without charge for all major platforms and can be freely distributed.

2.2.1. Pandas

Pandas [3] is a software library written for the Python programming language for data manipulation and analysis. In particular, it offers data structures and operations for manipulating numerical tables and time series. It is free software released under the three-clause BSD license. The name is derived from the term "panel data", an econometrics term for data sets that include observations over multiple time periods for the same individuals. Its name is a play on the phrase "Python data analysis" itself. Wes McKinney started building what would become pandas at AQR Capital while he was a researcher there from 2007 to 2010.

2.2.2. NumPy

NumPy [4] is the fundamental package for scientific computing in Python. It is a Python library that provides a multidimensional array object, various derived objects (such as masked arrays and matrices), and an assortment of routines for fast operations on arrays, including mathematical, logical, shape manipulation, sorting, selecting, I/O, discrete Fourier transforms introductory linear algebra, basic statistical procedures, random simulation and much more.

2.2.3. MatplotLib

Matplotlib [5] is a comprehensive library for creating static, animated, and interactive visualizations in Python. Matplotlib makes easy things easy and hard things possible.

- Create publication quality plots.
- Make interactive figures that can zoom, pan, update.
- Customize visual style and layout.
- Export to many file formats.
- Embed in JupyterLab and Graphical User Interfaces.
- Use a rich array of third-party packages built on Matplotlib.

2.2.4. Beautiful Soup 4

Beautiful Soup [6] is a Python package for parsing HTML and XML documents (including having malformed markup, i.e. non-closed tags, so named after tag soup). It creates a parse tree for parsed pages that can be used to extract data from HTML, which is useful for web scraping.

2.2.5. SciKit-Learn

Scikit-learn [7] (sklearn) is the most useful and robust library for machine learning in Python. It provides a selection of efficient tools for machine learning and statistical modelling including

classification, regression, clustering and dimensionality reduction via a consistence interface in Python.

2.2.6. PyTorch

PyTorch [8] is a library for Python programs that facilitates building deep learning projects. We like Python because is easy to read and understand. PyTorch emphasizes flexibility and allows deep learning models to be expressed in idiomatic Python.

In a simple sentence, think about Numpy, but with strong GPU acceleration. Better yet, PyTorch supports dynamic computation graphs that allow you to change how the network behaves on the fly, unlike static graphs that are used in frameworks such as TensorFlow.

2.2.7. NLTK

NLTK [9] is a leading platform for building Python programs to work with human language data. It provides easy-to-use interfaces to over 50 corpora and lexical resources such as WordNet, along with a suite of text processing libraries for classification, tokenization, stemming, tagging, parsing, and semantic reasoning, wrappers for industrial-strength NLP libraries, and an active discussion forum.

NLTK has been called “a wonderful tool for teaching, and working in, computational linguistics using Python,” and “an amazing library to play with natural language.”

Natural Language Processing with Python provides a practical introduction to programming for language processing. Written by the creators of NLTK, it guides the reader through the fundamentals of writing Python programs, working with corpora, categorizing text, analysing linguistic structure, and more.

2.3. Google Colab

Collaboratory, or “Colab” for short, is a product from Google Research [10]. Colab allows anybody to write and execute arbitrary python code through the browser and is especially well suited to machine learning, data analysis, and education. More technically, Colab is a hosted Jupyter notebook service that requires no setup to use while providing access free of charge to computing resources, including GPUs.

In order to make the most of PyTorch, the **Pro Version of Google Colab** were used in this project.

3. DATASET

Our Dataset is derived from the Kaggle Fine Foods dataset [11] which has .cvs suffix. Generally, the Fine Foods Dataset has exactly 10 columns such as Id, ProductId, UserId, ProfileName, 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	I have bought several of the Vitality canned d...	5
1	Product arrived labeled as Jumbo Salted Peanut...	1
2	This is a confection that has been around a fe...	4
3	If you are looking for the secret ingredient i...	2
4	Great taffy at a great price. There was a wid...	5

Fig 1: example of dataset

4. WEB CRAWLING

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 a kind of Cat Food with 3,718 reviews in total [12].

After crawling, the final shape of the crawled data frame was 2,368 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.

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

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 11,840 by 2.

5. DATA PREPROCESSING

Data preprocessing 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.

5.1. 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.

5.2. 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 5,000, which would mean creating a feature matrix out of the most 5,000 frequent words across text documents.

5.3. 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 33,825 by 5,000.

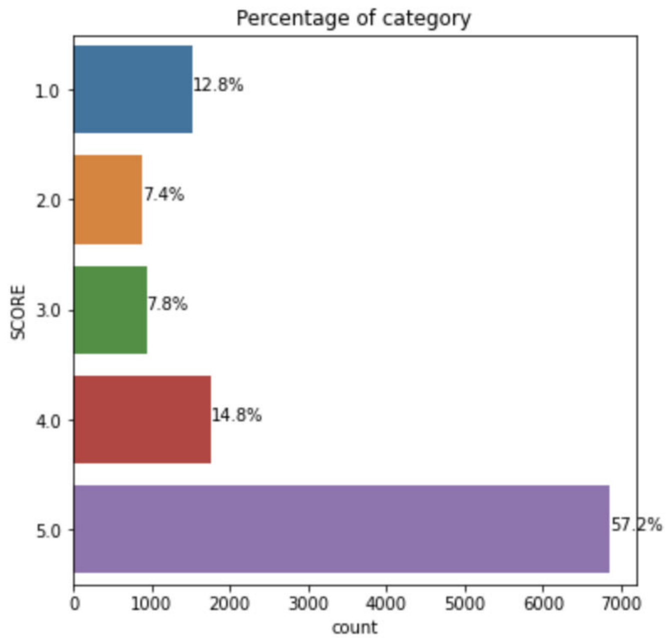

Fig 2: shape of imbalanced the dataset

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

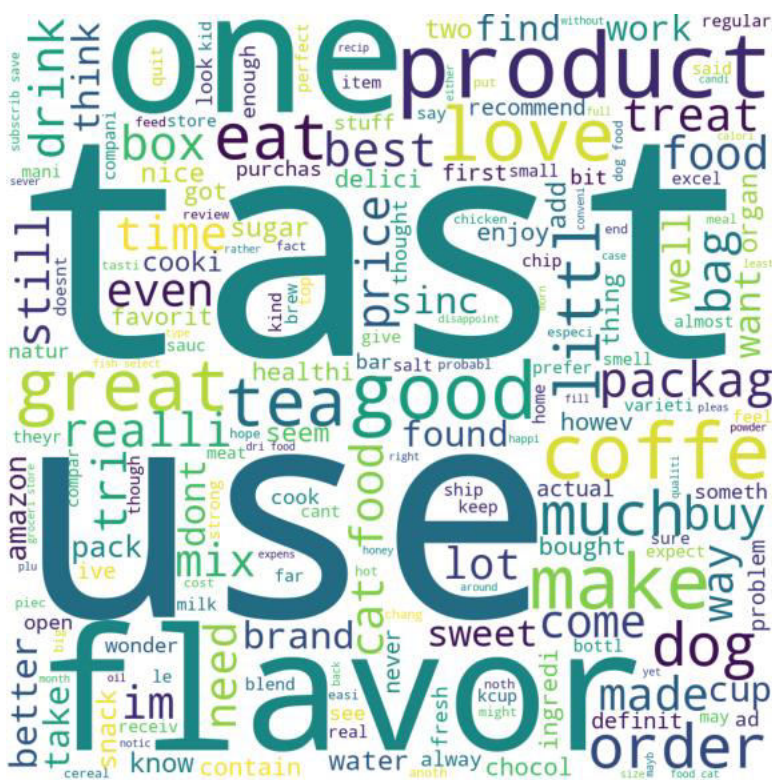

Fig 3: dataset’s word cloud

5.4. 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 27,060 by 5,000 and 6,765 by 5,000 respectively.

6. CLASSIFICATION

In machine learning, classification refers to a predictive modeling problem where a class label is predicted for a given example of input data.

Examples of classification problems include:

- Given an example, classify if it is spam or not.
- Given a handwritten character, classify it as one of the known characters.
- Given recent user behavior, classify as churn or not.

From a modeling perspective, classification requires a training dataset with many examples of inputs and outputs from which to learn.

A model will use the training dataset and will calculate how to best map examples of input data to specific class labels. As such, the training dataset must be sufficiently representative of the problem and have many examples of each class label.

In the following section, all of the classification techniques that used in this project are described in detail.

6.1. Gaussian Naïve Bayes

The Naive Bayes method makes the assumption that the predictors contribute equally and independently to selecting the output class. Although the Naive Bayes model's assumption that all predictors are independent of one another is unfeasible in real-world circumstances, this assumption produces a satisfactory outcome in the majority of instances.

Naive Bayes is often used for text categorization since the dimensionality of the data is frequently rather large.

$$P(y|x1, x2, x3, \dots, xN) = \frac{P(x1|y).P(x2|y).P(x3|y)\dots P(xN|y).P(y)}{P(x1).P(x2).P(x3)\dots P(xN)}$$

6.1.1. Hyper Parameters

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.

6.2. Support Vector Machines

Support vector machines (SVMs) are a set of supervised learning methods used for classification, regression and outliers' detection [13].

- The advantages of support vector machines are:
- Effective in high dimensional spaces.
- Still effective in cases where number of dimensions is greater than the number of samples.
- Uses a subset of training points in the decision function (called support vectors), so it is also memory efficient.
- Versatile: different Kernel functions can be specified for the decision function. Common kernels are provided, but it is also possible to specify custom kernels.

The disadvantages of support vector machines include:

- If the number of features is much greater than the number of samples, avoid over-fitting in choosing Kernel functions and regularization term is crucial.
- SVMs do not directly provide probability estimates, these are calculated using an expensive five-fold cross-validation

For the classification **Linear Support Vector Classifier** is used. The primal problem can be equivalently formulated as

$$min(w,b)\frac{1}{2}w^Tw + C\sum_{i=1} max(0,1 - y_i(w^T\phi(x_i) + b))$$

where we make use of the hinge loss [14]. This is the form that is directly optimized by LinearSVC, but unlike the dual form, this one does not involve inner products between samples, so the famous kernel trick cannot be applied. This is why only the linear kernel is supported by LinearSVC (ϕ is the identity function).

In machine learning, the hinge loss is a loss function used for training classifiers. The hinge loss is used for "maximum-margin" classification, most notably for support vector machines (SVMs)

6.2.1. Hyper Parameters

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.

6.3. Random Forest

Random Forest [15] is a popular machine learning algorithm that belongs to the supervised learning technique. It can be used for both Classification and Regression problems in ML. It is

based on the concept of ensemble learning, which is a process of *combining multiple classifiers to solve a complex problem and to improve the performance of the model.*

As the name suggests, "Random Forest is a classifier that contains a number of **Decision Trees** on various subsets of the given dataset and takes the average to improve the predictive accuracy of that dataset." Instead of relying on one decision tree, the random forest takes the prediction from each tree and based on the majority votes of predictions, and it predicts the final output. The greater number of trees in the forest leads to higher accuracy and prevents the problem of overfitting.

There are mainly four sectors where Random Forest mostly used:

- Banking: Banking sector mostly uses this algorithm for the identification of loan risk.
- Medicine: With the help of this algorithm, disease trends and risks of the disease can be identified.
- Land Use: We can identify the areas of similar land use by this algorithm.
- Marketing: Marketing trends can be identified using this algorithm.

6.3.1. Hyper Parameters

In case of working correctly, **GINI** indexing [16] 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

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

6.4. K-Nearest Neighbour

In KNN, K is the number of nearest neighbors. The number of neighbors is the core deciding factor. K is generally an odd number if the number of classes is 2. When K=1, then the algorithm is known as the nearest neighbor algorithm. This is the simplest case. Suppose P1 is the point, for which label needs to predict. First, you find the one closest point to P1 and then the label of the nearest point assigned to P1.

Suppose P1 is the point, for which label needs to predict. First, you find the k closest point to P1 and then classify points by majority vote of its k neighbors. Each object votes for their class and the class with the most votes is taken as the prediction. For finding closest similar points, you find the distance between points using distance measures such as Euclidean distance, Hamming distance, Manhattan distance and Minkowski distance. KNN has the following basic steps:

- Calculate distance
- Find closest neighbors
- Vote for labels

6.4.1. Hyper Parameters

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

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

6.5. Long-Short Term Memory (LSTM)

Long Short-Term Memory (LSTM) [17] networks are a type of recurrent neural network capable of learning order dependence in sequence prediction problems. This is a behavior required in complex problem domains like machine translation, speech recognition, and more. LSTMs are a complex area of deep learning. It can be hard to get your hands around what LSTMs are, and how terms like bidirectional and sequence-to-sequence relate to the field.

In order to make out prediction about the rates better, LSTM with 50 epochs has been used as one of the classifiers. In this case, PyTorch instead of Sk-Learn has been used.

The dimensionality of the input is equal to a one by two matrix, composed of sentence length and batch size. First, X as input is passed to the embedding method. Word embeddings are dense vectors of real numbers, one per word in your vocabulary. In NLP, it is almost always the case that the features are words. Second, the embedded output with size of one by three ([sentence_length, batch_size, embedding_dimension]) will be passed to RNN function. Finally, the output will send to for Linear Transformation.

Having said that, Adam is used as the model optimizer.

6.5.1. Hyper Parameters

As far as our range of labels are between 1 to 5, so we have five number of classes. 64 Batches are considered for training data. For the Validation set only two batches are set. As mentioned above, the RNN model has trained with 50 epochs with learning rate 0.00146 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.

7. SUCCESS MEASURES

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

7.1. Confusion Matrix

A Confusion Matrix [18] 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.

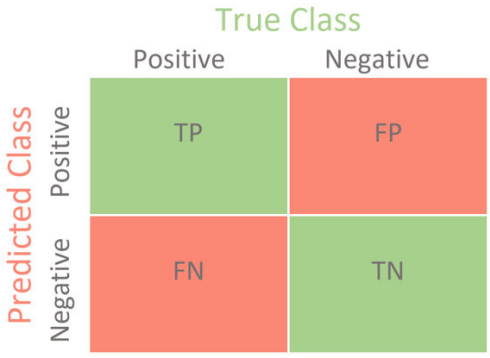

Fig 4: Confusion Matrix

7.2. 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. [19]

7.3. 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.

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

7.3.1. Precision

Precision (also called positive predictive value) is the fraction of relevant instances among the retrieved instances. [20]

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

7.3.2. Recall

recall (also known as sensitivity) is the fraction of relevant instances that were retrieved. Both precision and recall are therefore based on relevance. [20]

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

7.4. 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. [21]

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

8. 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.

8.1. 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.

8.2. Results

Now, it is time to review the results.

8.2.1. Gaussian Naïve Bayes

Measure	Amount
Accuracy Score	%72
F1-Score	%70.8
MSE	1.178

Table 1: metrics of NB

		y_true				
		1	2	3	4	5
ŷ	1	910	219	111	27	74
	2	19	1241	40	12	57
	3	18	67	1209	19	53
	4	78	212	199	594	263
	5	87	91	132	137	896

Table 2: Confusion Matrix of NB

8.2.2. K-NN

Measure	Amount
Accuracy Score	%77
F1-Score	%70.4
MSE	1.259

Table 3: metrics of K-NN

ŷ \ y_true	1	2	3	4	5
1	1295	8	32	5	1
2	54	1284	0	28	3
3	3	0	1361	2	0
4	34	89	28	1193	2
5	63	637	36	536	71

Table 4: Confusion Matrix of K-NN

8.2.3. LSTM

Measure	Amount
Accuracy Score	%87
F1-Score	%80.8
MSE	0.261

Table 5: metrics of LSTM

ŷ \ y_true	1	2	3	4	5
1	13	0	0	0	0
2	2	1	0	0	0
3	0	0	2	0	1
4	0	0	0	2	0
5	0	0	0	0	2

Table 6: Confusion Matrix of LSTM

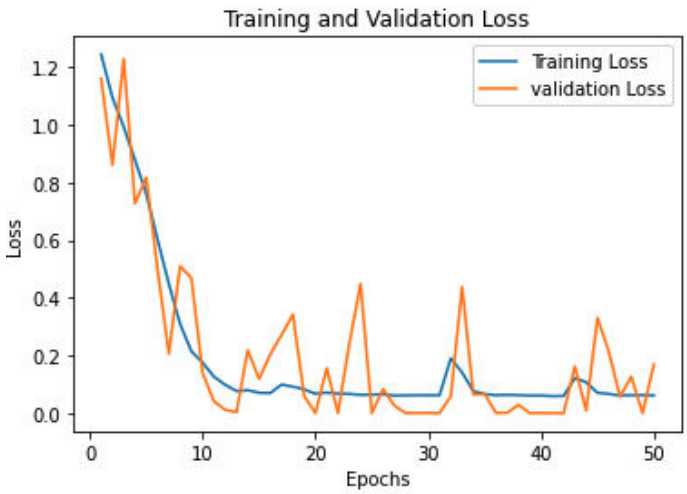

Fig 5: Train and Validation Loss of LSTM

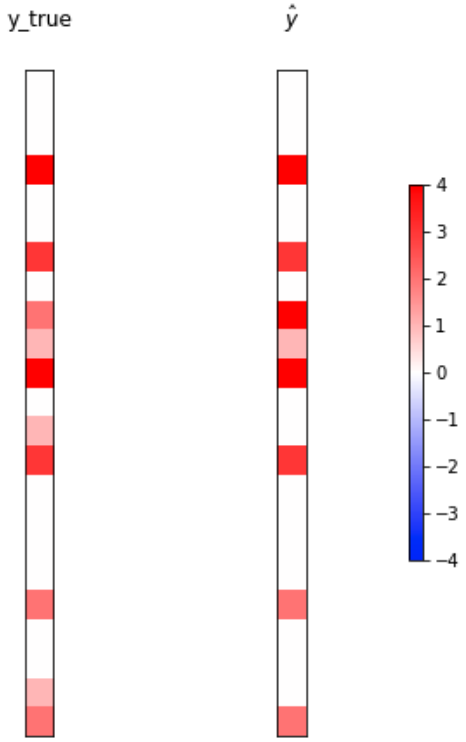

Fig 6: Comparing the results of y_true and ŷ

8.2.4. Linear SVC

Measure	Amount
Accuracy Score	%88
F1-Score	%87.4
MSE	0.592

Table 7: metrics of Linear SVC

ŷ \ y_true	1	2	3	4	5
1	1253	38	34	10	6
2	3	1358	2	0	6
3	8	0	1344	9	5
4	41	67	35	1150	53
5	103	61	87	253	839

Table 8: Confusion Matrix of Linear SVC

8.2.5. Random Forest

Measure	Amount
Accuracy Score	%91
F1-Score	%90.8
MSE	0.511

Table 9: metrics of Random Forest

ŷ \ y_true	1	2	3	4	5
1	1223	34	31	3	50
2	2	1347	0	6	14
3	6	6	1313	4	37
4	36	45	28	1108	129
5	75	15	17	77	1159

Table 10: Confusion Matrix of Random Forest

9. CHALLENGES AND LIMITATIONS

During the project, we encountered a number of challenges and limitations, which I will explain below.

9.1. Non-English Comments

As mentioned above, initially, we did not pay attention to this problem. Besides English, many people who live in Canada speak and write in French and other languages. We tried to look for items and products from the United Kingdom to mitigate this issue, where most people speak and write in pure English. But the problem remained unchanged. Even on UK amazon, many people who comment on a product were from other parts of Europe such as Italy, Germany, Spain, etc.

After exploring, we find out the ***googletrans*** library [22] for Python 3 is one of the most commonly used tools for translating texts into a desirable language such as English. The googletrans library accepts 750 consecutive words for real-time translation, and it means the

sentences with more than 750 words must be eliminated. This problem is solved with a simple if/else condition.

Another challenge was that the amount of data sent to Google for translation was too large, and Google blocked our work because it thought it is a DDOS attack.

To make sure this will not happen again, we used a language detector to ensure just non-English words will be translated.

9.2. Hardware Limitations

Working with PyTorch sometimes is turned to a really heavy job especially in NLP. One way is to train the model with CPU. But it is very time consuming. As a determined solution, we used Google Colab Pro in order to boost the Training Process.

10. CONCLUSION AND FUTURE WORKS

In this project we focused on addressing the issue with regard to predicting the right rate for a comment using Fine Food Dataset on Kaggle. First, we started with importing Libraries such as Numpy, BS4 and so on. Second, the dataset was imported and after that, the Cat Food 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 observed 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 project execution. 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 excellent.

We have in mind to try with different and larger datasets with different categories such as Electronics, Musical Instruments, 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.

All resources and codes of the project could be found on ***GitHub*** [23].

References

1. En.wikipedia.org. 2022. *Microsoft Excel - Wikipedia*. [online] Available at: <https://en.wikipedia.org/wiki/Microsoft_Excel> [Accessed 8 April 2022].
2. Python.org. 2022. *What is Python? Executive Summary*. [online] Available at: <https://www.python.org/doc/essays/blurb/> [Accessed 8 April 2022].
3. En.wikipedia.org. 2022. *pandas (software) - Wikipedia*. [online] Available at: <https://en.wikipedia.org/wiki/Pandas_(software)> [Accessed 8 April 2022].
4. Numpy.org. 2022. *What is NumPy? — NumPy v1.22 Manual*. [online] Available at: <https://numpy.org/doc/stable/user/whatisnumpy.html> [Accessed 8 April 2022].
5. Matplotlib.org. 2022. *Matplotlib — Visualization with Python*. [online] Available at: <https://matplotlib.org/> [Accessed 8 April 2022].
6. En.wikipedia.org. 2022. *Beautiful Soup (HTML parser) - Wikipedia*. [online] Available at: <https://en.wikipedia.org/wiki/Beautiful_Soup_(HTML_parser)#:~:text=Website,is%20useful%20for%20web%20scraping.> [Accessed 8 April 2022].
7. Tutorialspoint.com. 2022. *Scikit Learn - Introduction*. [online] Available at: <https://www.tutorialspoint.com/scikit_learn/scikit_learn_introduction.htm> [Accessed 8 April 2022].
8. Medium. 2022. *What is PyTorch?*. [online] Available at: <https://towardsdatascience.com/what-is-pytorch-a84e4559f0e3> [Accessed 8 April 2022].
9. Nltk.org. 2022. *NLTK :: Natural Language Toolkit*. [online] Available at: <https://www.nltk.org/> [Accessed 8 April 2022].
10. Research.google.com. 2022. *Google Colab*. [online] Available at: <https://research.google.com/colaboratory/faq.html#:~:text=Colaboratory%2C%20or%20%E2%80%9CColab%E2%80%9D%20for,learning%2C%20data%20analysis%20and%20education.> [Accessed 8 April 2022].
11. Kaggle.com. 2022. *Amazon Fine Food Reviews*. [online] Available at: <https://www.kaggle.com/datasets/snap/amazon-fine-food-reviews> [Accessed 9 April 2022].
12. Amazon.co.uk. 2022. *Amazon.co.uk:Customer reviews: Lifelong Complete Food for Adult Cats Mixed In Jelly Selection, 24 x 100g*. [online] Available at: <https://www.amazon.co.uk/product-reviews/B07HN58Y53/ref=cm_cr_arp_d_paging_btm_next_1?ie=UTF8&reviewerType=all_reviews&pageNumber=0> [Accessed 9 April 2022].
13. scikit-learn. 2022. *1.4. Support Vector Machines*. [online] Available at: <https://scikit-learn.org/stable/modules/svm.html> [Accessed 9 April 2022].
14. En.wikipedia.org. 2022. *Hinge loss - Wikipedia*. [online] Available at: <https://en.wikipedia.org/wiki/Hinge_loss> [Accessed 9 April 2022].
15. www.javatpoint.com. 2022. *Machine Learning Random Forest Algorithm - Javatpoint*. [online] Available at: <https://www.javatpoint.com/machine-learning-random-forest-algorithm> [Accessed 9 April 2022].

16. 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].
17. Brownlee, J., 2022. *A Gentle Introduction to Long Short-Term Memory Networks by the Experts*. [online] Machine Learning Mastery. Available at: <https://machinelearningmastery.com/gentle-introduction-long-short-term-memory-networks-experts/> [Accessed 9 April 2022].
18. 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].
19. 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].
20. En.wikipedia.org. 2022. *Precision and recall - Wikipedia*. [online] Available at: <https://en.wikipedia.org/wiki/Precision_and_recall> [Accessed 9 April 2022].
21. 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].
22. PyPI. 2022. *googletrans*. [online] Available at: <https://pypi.org/project/googletrans/> [Accessed 9 April 2022].
23. https://github.com/ebahador/Review_Rating_prediction_with-crawling_Amazon_data