ECE 408/208 ART OF MACHINE LEARNING FINAL REPORT

Authors: Ozyurt, Mehmet Emin and Pinco, Ilan

ABSTRACT

Many companies use five-star ratings to determine public sentiment towards their products. However, many comments on their products are made in various locations on the internet, all in pure text without an accompanying rating. This project attempts to create a machine learning model capable of giving positive or negative ratings to these comments. Such data would allow a more accurate approximation of public sentiment.

Multiple models were used and fed preprocessed data from a public dataset. One model was abandoned early on as it had great difficulty working with data that was further processed. The other three models were trained and had their accuracy and precision determined.

The Logistic Regression model had the highest accuracy at approximately 90%. The other models also provided high accuracy with Linear SVM giving a score just below Logistic Regression, and Random Forest giving a score just above 85%. This relatively high accuracy with only the beginnings of experimentation suggests that using Machine Learning models to predict public sentiment is both possible and very effective.

1. INTRODUCTION

This project aims to determine the viability of having machine learning models evaluate the public sentiment of company products through analyzing their online commentary on the relevant products. Through this project, we attempt to provide evidence towards this viability by training and evaluating multiple machine learning models for the task in question. Finaly, future possibilities for improvement are explored so that this idea is open for additional refinement.

2. METHOD

At the beginning of our code file, we use a dataset by the name of “yelp.csv” [1] which contains ten thousand text reviews for restaurants with their five-star ratings included.

We begin to determine our preprocessing steps by importing two additional pieces of data in the form og lists. The first is a list of all punctuation, taken from the string class using the string.punctuation feature. The second is a list of stopwords. These words include various neutral words such as “the” and “do”. Upon gaining these lists, a function is created to clean any input text files and output an array of strings which contains only words that are not stop words. A CountVectorizer is then made with this function as its analyzer. It is fitted with the 1 and 5 star reviews from the yelp dataset to determine which words it should count from future input text data. The entire yelp dataset is then run through the vectorizer, creating a dataset that can be inputted into a model and therefore used for training and testing.
Example text processing before inputting into a vectorizer:

This message:

```
My wife took me here on my birthday for breakfast and it was excellent.  The weather was perfect which made sitting outside overlooking their grounds an absolute pleasure.  Our waitress was excellent and our food arrived quickly on the semi-busy Saturday morning.  It looked like the place fills up pretty quickly so the earlier you get here the better.

Do yourself a favor and get their Bloody Mary.  It was phenomenal and simply the best I've ever had.  I'm pretty sure they only use ingredients from their garden and blend them fresh when you order it.  It was amazing.

While EVERYTHING on the menu looks excellent, I had the white truffle scrambled eggs vegetable skillet and it was tasty and delicious.  It came with 2 pieces of their griddled bread with was amazing and it absolutely made the meal complete.  It was the best "toast" I've ever had.

Anyway, I can't wait to go back!
```

Is turned into this array:

```
['wife', 'took', 'birthday', 'breakfast', 'excellent', 'weather', 'perfect', 'made', 'sitting', 'outside', 'overlooking', 'grounds', 'absolute', 'pleasure', 'waitress',
```

```
'excellent', 'food', 'arrived', 'quickly',
'semibusy', 'Saturday', 'morning', 'looked',
'like', 'place', 'fills', 'pretty', 'quickly',
'earlier', 'get', 'better', 'favor', 'get',
'Bloody', 'Mary', 'phenomenal', 'simply',
'best', 'Ive', 'ever', 'Im', 'pretty', 'sure',
'use', 'ingredients', 'garden', 'blend',
'fresh', 'order', 'amazing', 'EVERYTHING',
'menu', 'looks', 'excellent', 'white',
'truffle', 'scrambled', 'eggs', 'vegetable',
'skillet', 'tasty', 'delicious', 'came', '2',
'pieces', 'griddled', 'bread', 'amazing',
'absolutely', 'made', 'meal', 'complete',
'best', 'toast', 'Ive', 'ever', 'Anyway',
'cant', 'wait', 'go', 'back']
```

The first experiment conducted was with the multinomial naive bayes model [2]. The model was given the entire dataset to train on, and it was determined that it was giving the initial expected output of a five-star rating. No significant data was collected from this; it was only a confirmation of functionality.

The second experiment was with the same model, but with the dataset split into training and testing portions. After training a new instance of the model, its accuracy with the testing data was determined while the outputs for both the training and testing data were displayed for analysis.

At this point, an additional step of data processing is performed. This processing includes the feature TF-IDF, or Term Frequency-Inverse Document Frequency. The TF-IDF of a word determines its importance in a collection of documents. This variable comes in the form of a decimal value that acts as a weight which represents the importance of the relevant word within a document. Using this value as the input to the chosen models can allow it to give outputs with greater accuracy. The processing, using this feature, gives an array of decimal values corresponding to each significant word in the review. A new dataset is created from the yelp dataset using this process.

The third experiment takes place after the additional data processing step was added. With the new TF-IDF input, the same multinomial naive bayes model was trained. However, the model outputs an extremely biased set of predictions, far from what would be considered acceptable. It was at this point that other models were tested and the multinomial naive bayes model was abandoned.

The fourth experiment was a comparison of the accuracies between three different models. Each model was trained using the processed data and then had their results displayed for analysis. The models chosen were logistic regression [3], linear SVM [4], and random forest [5]. The logistic regression model had the highest accuracy during this experiment.

An error analysis was then conducted to determine possible points of improvement. After filtering the outputs for only false negatives and false positives from the logistic regression model, the reviews were analyzed. Four types of errors were identified.

To test a possible improvement to the model, the logistic regression model was used to predict the test data again, but this time the data was sorted into a new display. Using sets of keywords, each review had its relevancy towards certain restaurant-based aspects listed. If a review had words relevant to food and service, it would be listed twice, once tagged with food, second tagged with service. This allowed an analysis as to whether the model had trouble evaluating reviews that contained certain subjects. This was the final conducted experiment.

3. EXPERIMENTS

The first experiment resulted in a success. The model was able to give a positive or negative prediction depending on the input text.

The second experiment resulted in an accuracy of 92% with the test data with some positive results from the confusion matrices. These values are shown in table 1. The main source of inaccuracy for both matrices seemed to be false positives, where more than 70% of incorrect predictions fall into that category for both. This can be seen in the confusion matrices of figures 1 and 2

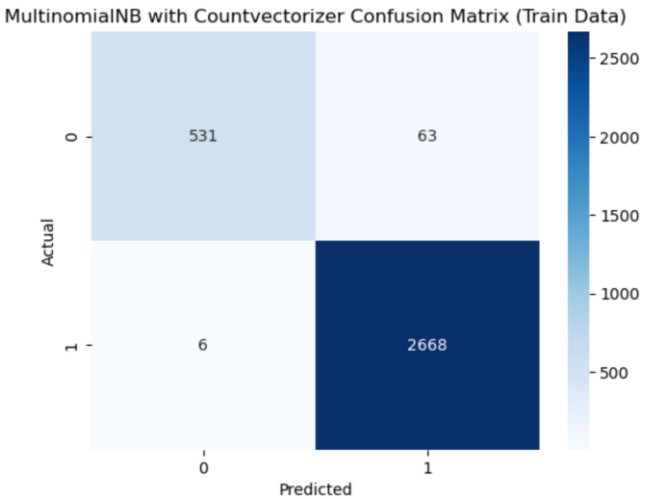

Fig.1. MultinomialNB with CountVectorizer Confusion Matrix on train data

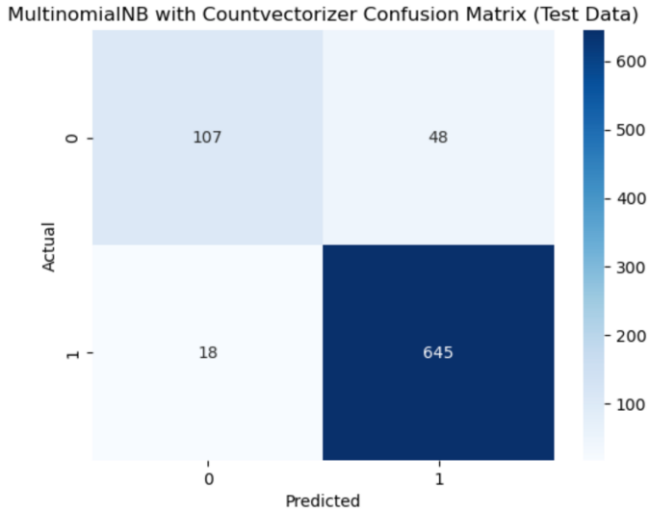

Fig.2. MultinomialNB with CountVectorizer Confusion Matrix on test data

Table.1. Statistics from the Multinomial Naive Bayes model:

	precision	recall	F1-score	support
1	0.86	0.69	0.76	155
5	0.93	0.97	0.95	663
accuracy			0.92	818
macro avg	0.89	0.83	0.86	818
weighted avg	0.92	0.92	0.92	818

The third experiment resulted in a heavily biased and fairly inaccurate set of predictions, as shown in figure 3 and 4, where only 5 of the 3,473 reviews were predicted as negative. A likely reason for this bias is due to the relatively high concentration of positive reviews compared to negative reviews. It was due to this relative shift in results that the current model was abandoned and other models were tested.

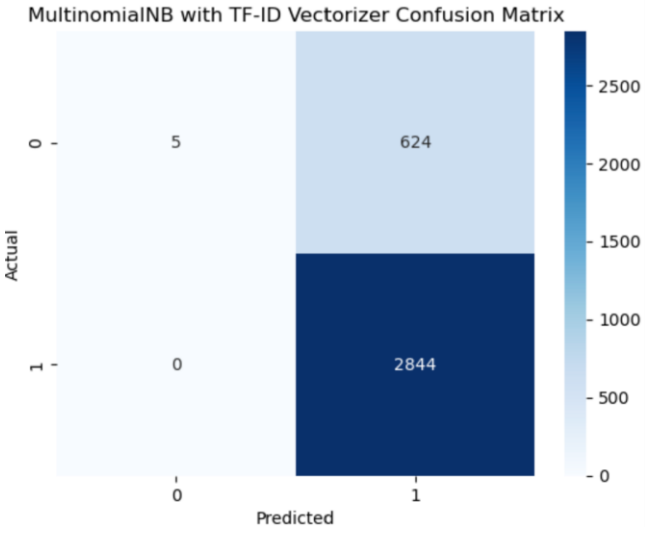

Fig.3. MultinomialNB with TF-ID vectorizer Confusion Matrix on train data

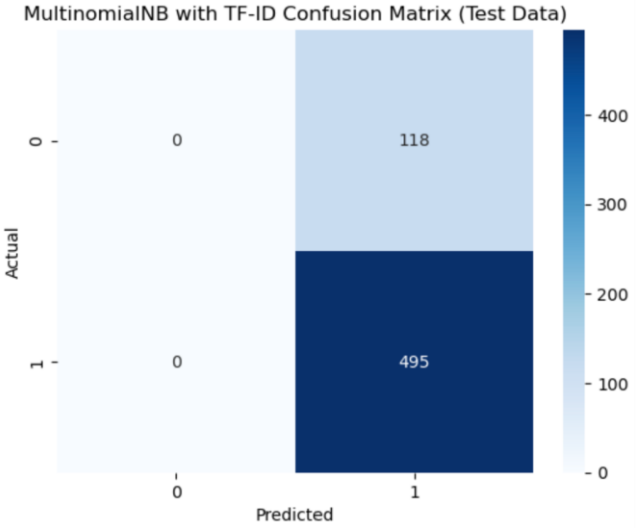

Fig.4. MultinomialNB with TF-ID vectorizer Confusion Matrix on train data

The fourth experiment gave a set of accuracies for the three tested models, shown in table 2. As shown by Figures 5 to 7, All three models were biased towards giving more positive predictions than negatives, echoing the same issue found in the previous experiments where false positives were the majority contributor towards the error. This is likely due to the biased input data, where much of the data is positive while a much smaller portion is negative.

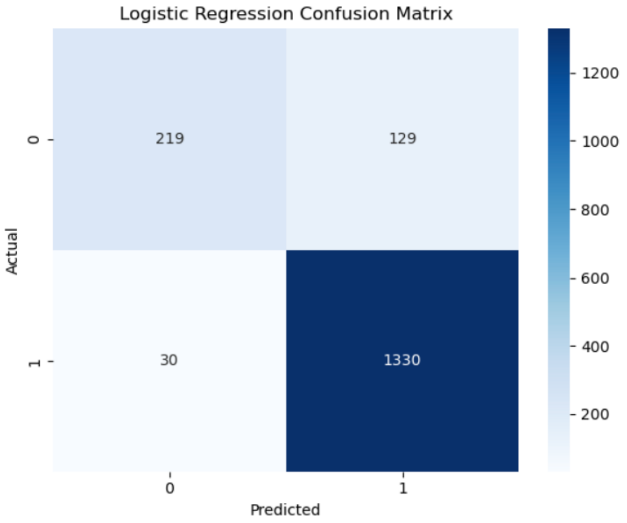

Fig.5. Logistic Regression Confusion Matrix on test data

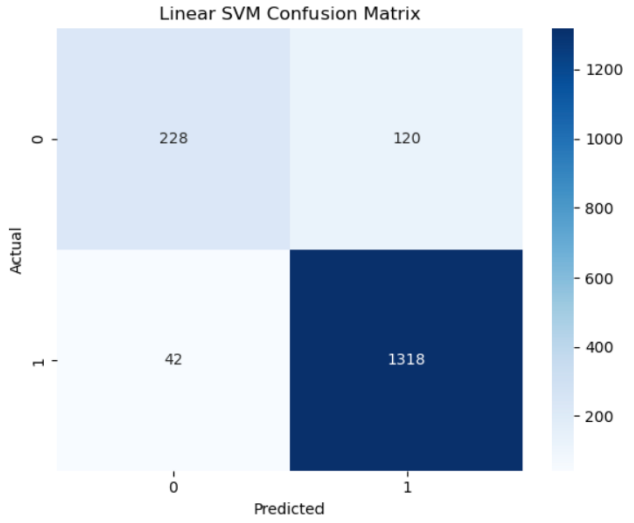

Fig.6. Linear SVM Confusion Matrix on test data

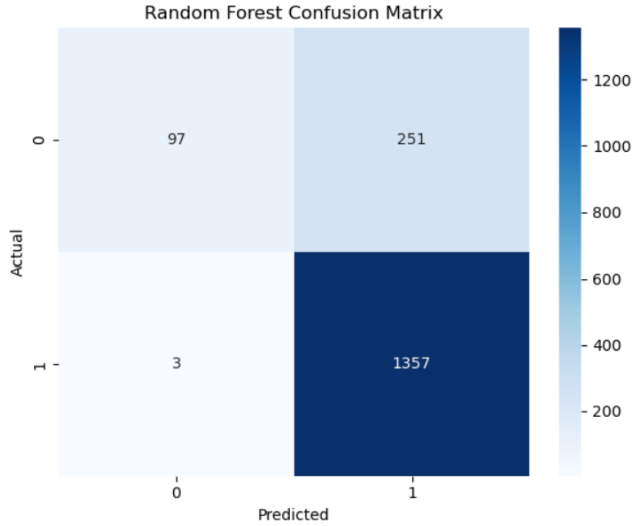

Fig.7. Random Forest Confusion Matrix on test data

Table.2. statistic comparisons between 3 models:

	Accuracy	Precision	Recall	F1-score
Logistic Regression	0.906909	0.911583	0.977941	0.943597
Linear SVM	0.905152	0.916551	0.969118	0.942102
Random Forest	0.851288	0.843905	0.997794	0.914420

The error analysis showed that four types of errors were the most prevalent. False positives were mainly due to mixed sentiment reviews that confused the model, or reviews that were only mildly negative with no obvious or heavy sentiment. False negatives were also made from mild reviews that lacked any strong keywords. A portion of errors were also made from reviews containing informal language and liberal use of capitalization. For improvements, many could be made during preprocessing. A simple normalization of the reviews, such as removing or applying all capitalization, could assist accuracy. A method for detecting mixed or conflicting sentiment has a possibility of positively refining the data. It is also possible to manually augment the data for edge cases.

The improvement test resulted in a set of accuracies for each restaurant-based aspect. At best, reviews containing service relevant keywords were slightly over 95% accurate, while at worst, reviews containing location-based keywords were a bit over 90% accurate. These results show that errors can occur in greater frequency due to the subject matter of the review.

4. CONCLUSIONS

In summary, it is viable to train and use a machine learning model to determine the public sentiment towards a product using text-based commentary. With only a few basic models trained, over 90% accuracy was achieved with the best result. The results of these experiments provide directions in which to further improve the functionality of the model. It is certainly possible to gain high enough accuracy to apply this approach in real world situations.

5. REFERENCES

[1] Yelp, INC. Chris Crawford, and sebastien, "Yelp Dataset," Yelp, INC. https://www.kaggle.com/datasets/yelp-dataset/yelp-dataset , 2022.

[2] Andreas Lindholm, Niklas Wahlström, Fredrik Lindsten, Thomas B. Schön, Machine Learning: A First Course for Engineers and Scientists, Cambridge University Press, https://smlbook.org/book/sml-book-draft-latest.pdf , pp.217-243, 2022.

[3] Andreas Lindholm, Niklas Wahlström, Fredrik Lindsten, Thomas B. Schön, Machine Learning: A First Course for Engineers and Scientists, Cambridge University Press, https://smlbook.org/book/sml-book-draft-latest.pdf , pp.45-54, 2022.

[4] Andreas Lindholm, Niklas Wahlström, Fredrik Lindsten, Thomas B. Schön, Machine Learning: A First Course for Engineers and Scientists, Cambridge University Press, https://smlbook.org/book/sml-book-draft-latest.pdf , pp.208-213, 2022.

[5] Andreas Lindholm, Niklas Wahlström, Fredrik Lindsten, Thomas B. Schön, Machine Learning: A First Course for Engineers and Scientists, Cambridge University Press, https://smlbook.org/book/sml-book-draft-latest.pdf , pp.171-174, 2022.