**Final Project**

Michael K. Zoucha

College of Science & Information Technology, Bellevue University

DSC 680: Applied Data Science

Professor Williams

09 July 2022

**Milestone 1**

**Introduction**

Reviews are a wonderful tool for gauging a company's performance from the eyes of their customer(s). These short snippets of text also provide immense meaning and context to other potential customers. There is so much information available to buyers, it is almost a conscious decision to NOT be an informed consumer. The hospitality industry experiences make or break moments with these reviews and social media posts and bad interactions are broadcast further and faster than ever. The pandemic did serious damage to all industries, but maybe none more than hospitality. Any assistance to keep the customers happy (and returning) will ultimately help speed up the recovery of the industry. The digital age has presented a unique opportunity for companies to interact with their customers on a more personal level and if taken advantage of, can have direct effect on bottom lines.

Hotel reviews also present a unique opportunity for training natural language processing (NLP) tasks because the one responsible for writing the review is the one who labels the data - with their 'star' rating. With the exponentially increasing use of social media, SMS text messages, and chats, companies now have massive amounts of raw text data to process, and, unlike reviews, these sources do not come with pre-labeled ratings. The application of training from these pre-labeled datasets can then be applied back to the un-labeled sources to help these companies flag the issues that need dealt with the most, and quickest, to limit the backlash it could cause. Chats can be flagged and sent to escalation departments before ever being answered by a live agent. Social media posts can be logged and reported if rectification is warranted in the situation.

**Dataset**

For this project I will be using a hotel review dataset from [Kaggle](https://www.kaggle.com/datasets/datafiniti/hotel-reviews) that is a compilation of reviews from different travel and review sites for hotels all around the US. The dataset contains includes a significant amount of information about the hotel itself, including name and address information, as well as the review itself, including source, date, rating, and the text from the rating. The only information I will be using for this model is the rating and the text from the rating. There are a total of 19,746 reviews and ratings in my final dataset and is a combination of two CSV files containing the same columns. I also compiled a separate dataset from a different review site to see if the model from one dataset will perform well against a different dataset of the same content.

**Models**

Like my first project, this is another classification problem, but one that is relevant to my current projects at work. I plan on using GridSearchCV to test all the ‘usual’ suspects for classification, including Random Forest, Logistic Regression, Multinomial Naive Bayes, and linear support vector classification to see which model performs best on the dataset. I am also planning on trying a neural network to see if it can perform better than the more traditional machine learning methods. If I can find the time, I may even try using popular third-party NLP tools such as BERT and GPT-3 to see for myself if they are as great as everyone says they are! Neither are technically classification tools, but both can be used to easily assist in classification tasks.

**Risks / Contingencies**

This dataset is imbalanced with many, many more reviews with 4- and 5-star ratings than with 1-, 2-, and 3-star ratings. This is surprising to me from my experience in the hospitality industry – usually the only ones who take the time to leave reviews are upset about something. There is also a risk of running my bill up significantly with the third-party tools if I do not properly research and prepare so I can use it as little as possible. I am aware there will be costs but am also using it as a test run of sorts for my job. I am also fortunate enough to have the computer resources at home to really dig into some of those tools.

I am concerned the subjective nature of hotel reviews and the people leaving them could confuse the model, especially between the rating level adjacent to the rating of the review (e.g., predicting a 3 or 5 for a 4-star review). There is also a wide range of what a guest desires in their hotel stay, for instance, one guest may want to be left completely alone and not talk to hotel staff except for check-in and check-out, while another may want their every need attended to. There is also a chance the model may not transfer well to my second dataset, but I think it will perform very similarly since the keywords for what makes a ‘good’ and ‘bad’ hotel review are the same.

**EDA / Graphic Analysis**

Early analysis shows that we have a lot more 'good' reviews than 'bad' ones, which will need to be accounted for when attempting to build a classification model.

Chart, bar chart

Description automatically generated

*Figure 1: Count of Reviews by Rating*

The heatmap and state histogram show that we have more reviews from certain areas of the United States, particularly California. If trying to encode and use the state column for classification, the number of total reviews from each state will also need to be equalized to ensure fairness in the model.

Map

Description automatically generated

*Figure 2: Heat Map of Review Distribution*

Preliminary results from separating the adjectives in the reviews give a good idea of the words used most in both the best and worst reviews. The crossover in words between the two alludes to the fact that phrases will need to be tokenized, as well as words (i.e., 'clean' is in the word cloud for the worst reviews, which leads me to believe the reviews are actually saying 'not clean' or 'need clean', etc.).



*Figure 3: Common Words from 5-Star Reviews Figure 4: Common Words from 1-Star Reviews*

**Data Preparation**

The two data frames were combined (each from different segments of time) and any duplicates dropped. I also dropped all unnecessary columns since we are only concerned with the rating and the text data for the initial classification model. Dummy variables were created for the reviews rating for the final classification target. From here, I will be using different methods for pre-processing the reviews text depending on which model/pipeline the data will be fed to (custom classification, keras, etc.). To give the model(s) the ability to generalize better without the risk of overfitting, all reviews were down sampled to balance between the classes. It removed quite a bit of data, which is unfortunate, but it also created a much more accurate model to use. With so many examples of the minority class still, I am interested what the results would have been had I not balanced the classes right out of the gate.

**Model Evaluation**

As we can clearly see from the models below, the closeness of the reviews' ratings (labeled by the user themselves) made it very difficult for the models to accurately distinguish reviews from those in close, but different, rating categories (e.g. distinguishing 4 star from 3 or 5 star reviews). I believe this is due to the subjective nature of online reviews and the labels they are given by the user. One person's 3-star review is another's 5-star or 1-star review.

Calendar

Description automatically generated Calendar

Description automatically generated

Figure 5 - Logistic Regression Results (1-5) Figure 6 - Neural Network Results (1-5)

I was pleased the model didn't have too difficult of a time distinguishing between the 5-star and 1- or 2-star reviews, or the 1-star from the 4- or 5-star reviews. This led to me changing my strategy for modelling the reviews.

By grouping the reviews by 'good' (4- and 5-star reviews) and 'bad' (1- and 2-star reviews) and dropping the 3-star reviews, I was able to double every performance metric of the models, including accuracy and precision. From a business perspective, this model is about quickly discovering, escalating, and handling the 'bad' text data that comes through in different channels, so I now believe classifying the individual 'star' rating is of less value.

Calendar

Description automatically generated Calendar

Description automatically generated

Figure 7 - Logistic Regression Results (binary) Figure 8 - Neural Network Results (binary)

This model is now able to predict the binary category of 'good vs bad' much more quickly, and accurately than I was ever able to get otherwise, regardless of the amount of tuning I did, or layers I added to a neural network. When testing the model on the second dataset, I was pleasantly surprised to get performance than with the initial dataset – 93.41% using logistic regression.

Conclusion

With the untimely demise of my old computer, I had to reassess both what I was going to do to complete the project and how I was going to go about it. I was hoping to use common pretrained tokenizers like BERT and GPT-3, but unfortunately, I did not have the time to dive deep into those tools. My first few runs were taking a day or more to complete between each minor tweak of the model. That is the one downside of using such powerful pre-trained resources is the power needed to run such models. Switching to a binary target since I did not have the computation power necessary to accurately predict individual star rating allowed for much higher performing models.

This model is ready to be deployed for the use case described above – as an initial text, chat, social media, and email routing tool to escalate the lowest rated text samples to the appropriate teams to try and eliminate the frustration that comes with already being upset with a company and being forced to be transferred through multiple departments before finally talking to the right person. There is certainly some fine tuning that could be done and gathering even more text data from the actual business would help increase performance as well, since reviews don’t go into some of the most significant issues a hotel company sees within their customer service teams, such as rewards accounts, reservations, cancellations, etc.

**Assumptions**

A significant assumption made in this project was that all sources of text data (social media posts, chats, emails, etc.) will contain the same words and phrases for good / bad as the reviews did. It is possible that sources like Facebook and Twitter posts would not contain enough text for the model to perform well in those instances.

**Limitations**

I do not foresee this model having many limitations if it is regularly updated with new reviews. I would really like to see how it performs in a production setting at work using ‘real-world’ hotel text data. It is possible separate models need to be made for certain sources of text, but the framework would be the same, it is just the training data that would change.

**Challenges**

The only real challenge I faced during this project was the short amount of time I had to complete it. I would have like to branch out to use publicly available third-party tools to see how they perform. I was mostly curious if they would perform well enough to be able to do a full multi-class classification instead of condensing it to binary classification.

**Future Uses**

We recently started ramping up a new NLP program at work and this is one of the first items on my list! We have so many teams that are randomly looking through hundreds or thousands of instances of phone calls, emails, reviews, posts, etc., for specific occurrences, it will save everyone a ton of time and effort if I am able to point them to what they need to focus on. This model (or a similar one) will first be deployed to the social media team.

**Recommendations**

I recommend the company acquire the licenses necessary to scrape popular social media sites from the backend, instead of having associates manually looking at each hotel’s individual pages on a scheduled basis. This would allow us to search for occurrences and direct them to the right pages. My only recommendation for the model implementation is full testing on all different sources of text individually and as a whole to see if the model needs updated for specific contact types

**Implementation Plan**

This model only uses two fields for training and one for predictions, so implementation is as simple as feeding it a cleaned text column from a data frame or SQL table. We already have the resources and frameworks available to set up and support this model, so no additional work is necessary.

**Ethical Assessment**

I don’t have any concerns surrounding the ethics of this project. The data is readily available on the internet (and reviews/posts are put there specifically to be readily available on the internet) so there are no privacy concerns that come to mind. The model itself will not be used to make any drastic decisions; it is meant as more of a ‘triage’ tool.

Questions

1. What cleaning methods did you use on the text?

* I used stopword removal, stemming, and non-alpha character removal

1. Why do you think there was such an overlap between the 5- and 1-star word clouds?

* I am assuming the difference comes from the removal of stopwords (e.g. a 5-star review has the word ‘clean’ while a 1-star review would contain the phrase ‘not clean’)

1. It seems like most of the reviews are ‘good’, what makes this tool so helpful?

* Since most reviews are good, it can be difficult to comb through them all to find the ‘bad’ ones. By bringing them immediately to the surface, they can be resolved much quicker.

1. How often will the model need to be updated?

* We will have auditing processes in place to monitor the drift of the model to see if it needs to be updated in the future.

1. Why didn’t you add any other metrics to the model?

* By using text only, we allow the model to be used on all different sources of text

1. How could this be fine-tuned, if at all, for specific hotel companies / brands?

* Custom stopwords and brand specific model training could possibly increase performance for the model

1. What is the resource cost of implementing a model like this?

* All infrastructure is already in place, so no extra costs are necessary

1. Could this model be run in real time?

* Yes, this model could act as a ‘filter’ for incoming text data to ‘flag’ and send it to the appropriate team

1. Can the model be trained in other languages, or does it only ‘know’ English?

* Currently, the model was only trained in English, but the NLP packages used can handle other languages. We would need a separate model for each language.

1. Would a larger training set net even better results?

* It is possible, it could also decrease performance by increasing the amount of noise in each rating category.