Final Project – Hotel Review Classification

Michael K. Zoucha

College of Science & Information Technology, Bellevue University

DSC 550: Data Mining

Professor Werner

16 February 2022

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.

In this analysis, I will be training models to classify the rating (score of 1-5) of the hotel review by the contents of its text. The reviews were obtained through labelled datasets available

on Kaggle. To start, I will train the model with reviews from only one source using readily available models such as BERT and GPT-3 and see how it performs on the labelled reviews from another source. If the model can generalize across different sources, we can infer it will generalize well no matter the source of text (chat, web, blog, social media, etc.). I will also train a model based on the combination of all sources to compare its performance. If the second model performs better, we can infer the model(s) need as many different sources as possible to be effective regardless of source.

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.

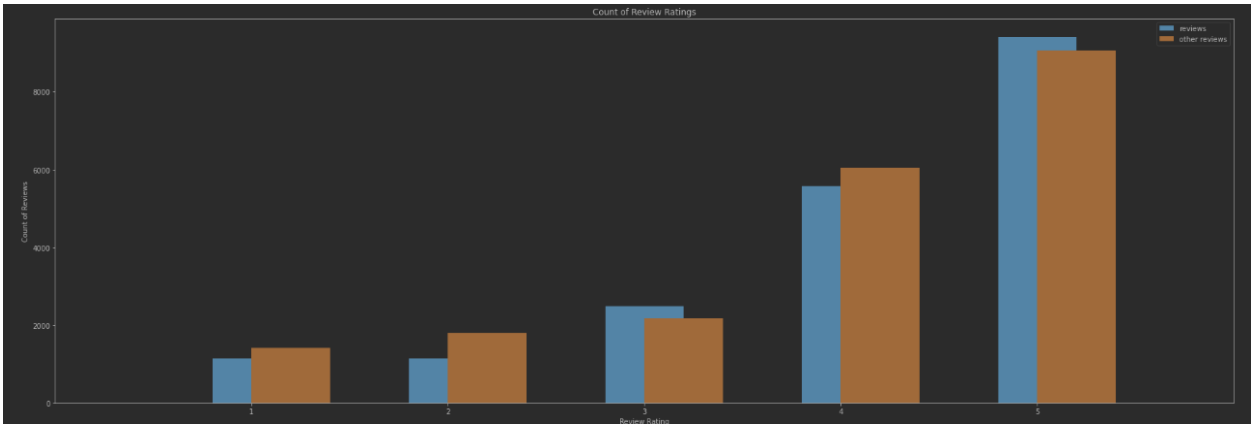

Figure 1: Count of Reviews by Rating

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.

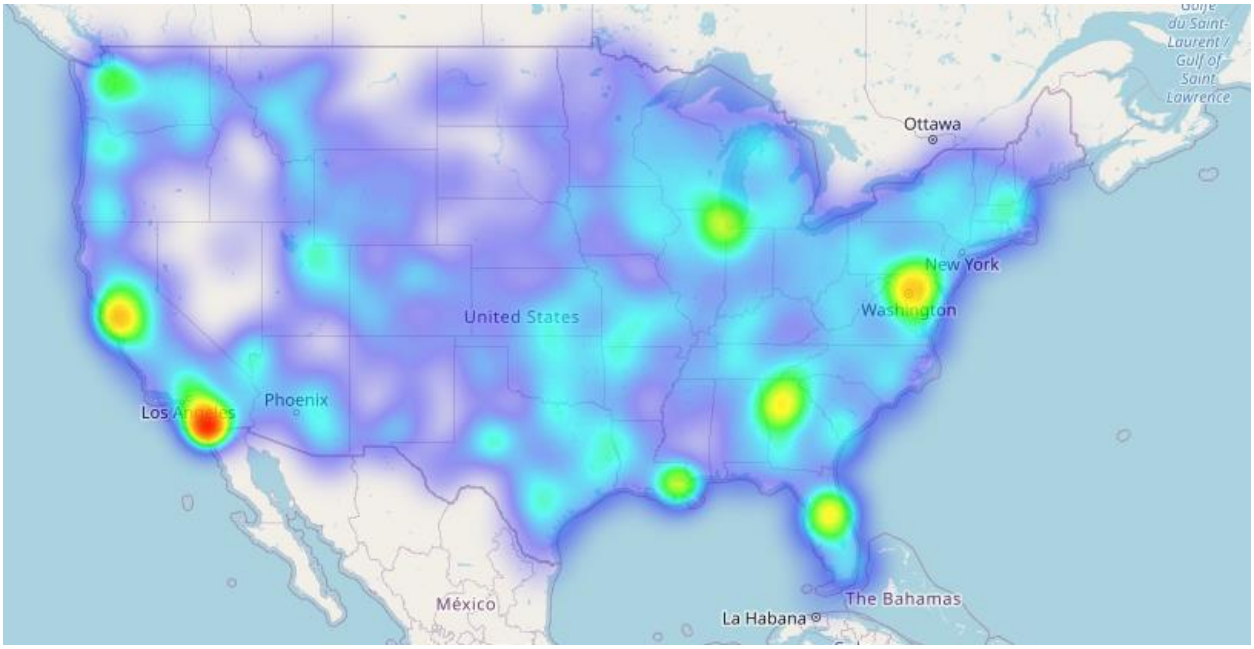

Figure 2: Heat Map of Review Distribution

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.). In order 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.

Model Evaluation

As we can clearly see from the models above, 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. 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 lead 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. 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.

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 pre-trained tokenizers like BERT and GPT-3, but without a GPU, I couldn't get meaningful results in the amount of time I had to complete that milestone. 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.