Predicting the Star Ratings on the Yelp Dataset: a Research on Text Mining and Sentiment Analysis

Yizhang Liu A15910821
Yikai Mao A15887137
Allan Sun A15937284
Alan Zhang A16013308

In this assignment, we will be using the review dataset released by Yelp, the most popular business rating app. The original dataset contains around 8.7 million reviews, all shuffled. In this assignment, we will only use the first 200,000 reviews.

Exploratory Data Analysis

Since the dataset was provided by Yelp and all entries are automatically generated, there was no missing data and thus no cleaning was required. The dataset contains the following features: “review id”, “user id”, “business id”, “stars”, “useful”, “funny”, “cool”, “text” and “date”. The “review id”, “user id” and “business id” are all hashed code, and therefore don’t contain valuable information except for personal identification purposes. We chose to focus our research on text mining so we mostly ignored them in our research.

The “stars” column is an ordinal categorical value as the numbers themselves have a meaning and a bigger number implies better service/products. “Useful”, “funny”, and “cool” are quantitative variables that only have integer values. Upon initial inspection, 75% of the reviews fell above 3 stars and more than half of the reviews received 4 stars or more. This indicates that the dataset is skewed to the left instead of normally distributed as shown in Figure 1.

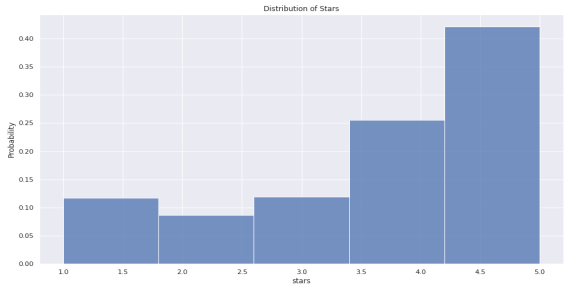

Figure 1, frequency of star ratings

The features “useful”, “funny”, and “cool” are also very spread out with high variance. The following chart shows that over 75% of the users do not use the feature in Yelp at all.

```
[20]: df.describe()
```

[20]:

	stars	useful	funny	cool
count	100000.000000	100000.000000	100000.000000	100000.000000
mean	3.777250	0.989390	0.336430	0.375620
std	1.376391	2.129099	1.212221	1.205016
min	1.000000	0.000000	0.000000	0.000000
25%	3.000000	0.000000	0.000000	0.000000
50%	4.000000	0.000000	0.000000	0.000000
75%	5.000000	1.000000	0.000000	0.000000
max	5.000000	128.000000	42.000000	47.000000

Table 1, statistics features of the dataset

Figure 2 shows that the average rating of businesses has trended upwards with time. Comparing this graph to the number of reviews submitted by users each year, we can conclude that the average rating scaled up in a logarithmic manner until it reached the horizontal asymptote at 689 reviews per year. The average bounces back and forth until a decrease in the number of reviews in 2018 leads to an increase in average user rating. This is presumably because businesses are more aware of their ratings on Yelp over the years and are optimizing it as part of their performance metric. Alternatively, a possible change in perception that a 4/5 star rating as the baseline for a business that fulfills its intended purpose could have also contributed to the rise of the average rating. This is a possible cause as 50% of the ratings received are 4 stars and above as shown in figure 1. From these two figures, we can safely conclude that

year/date and changes in the number of reviews per year are bad linear predictors of rating.

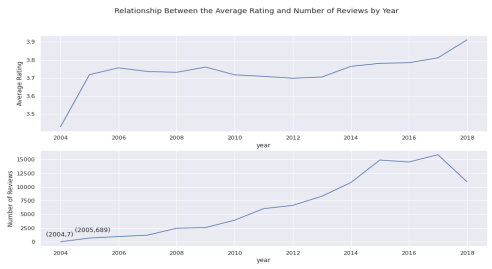

Figure 2: Relationship Between the Average Rating and Number of Reviews by Year

The oscillating line in figure 3 shows that none of the features provided by Yelp are good linear predictors of the average rating. This pushed us towards using review text as the primary feature in our predictive model as we can extract a lot of models from the reviews.

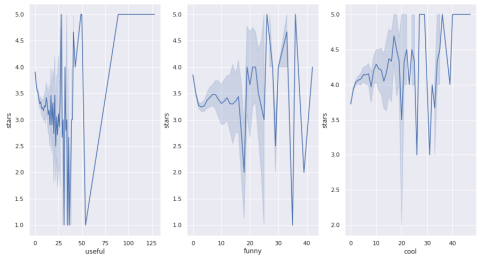

Figure 3: Useful/Funny/Cool as predictors of rating

Fitting a linear line to the dataset displays a negative correlation between review length and average rating. We will be using the logistic regression model as our baseline 1 and user average as baseline 2.

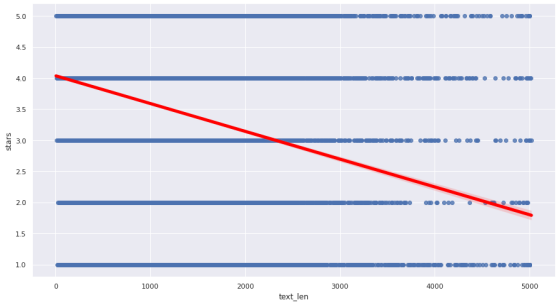

Figure 4: Relationship between Review Length and Rating

A final observation is that there is no strong correlation between stars and other variables other than text length, so we will be sticking with the review text as our primary feature.

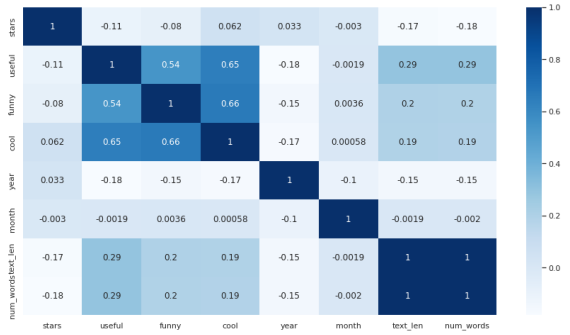

Figure 5:Heatmap of the correlation between any two variables

2. Predictive task:

The goal of this project is to predict the star rating of a particular review. The metric we will be using to evaluate our models is the accuracy of the prediction, calculated by the total number of correct predictions divided by the size of the testing dataset. Since star rating is a categorical variable that scales from 1 to 5, the accuracy of the predictions better reflects the success of our model in real life applications.

Baseline Models:

The objective of the baseline models is to beat out a random guess. Since we have 5 categorical variables, a random guess would yield a ⅕ or 20% accuracy, so all of our baseline models should be above 20% accuracy.

<u>Baseline 1: Use the length of the review text as a linear predictor of the stars rating</u>

The advantage of this model is that it's computationally cheap to extract the length of the review and will serve as a great baseline for complicated NLTK based models. The disadvantage is that it does not accommodate words that have a positive/negative connotation. It simply implies that longer review length associates with either a more positive experience or negative experience. We implemented this

model using sklearn's logistic regression to make the classification. The accuracy of the model on our validation set is 42% and 41.954% on our test set.

Baseline 2: Predict the rating based on the user's average rating

If the user has not posted a review before, use the global average across all users. The intention of this model is to see if users are more likely to write a review if their experience is negative. If this model yields a high accuracy, then it will imply that people with more extreme opinions are more likely to write a review and thus using the user's average rating is a good predictor of the future rating they will give to a business. To obtain the user's rating average, we simply totaled up all of the ratings under the user and got the average rating. The advantage of this model is that it is very easy to implement and can scale indefinitely. The disadvantage of this model is that it requires more data to be accurate, particularly more data about the user. Our model yielded a training accuracy of 84.93%, validation accuracy of 26.4%, and test accuracy of 25.6%. Upon further investigation, we discovered that only 13% of the users wrote more than 1 review and only 0.12% of the users wrote more than 10 reviews. Therefore, we would require a lot more, potentially billions of data points in order for this model to catch any trends in the reviews users write. Since this is too computationally expensive to run on a personal computer, we will be disregarding this model. However, this model can be appealing if the workload can be managed by a server or through cloud computing.

Baseline 3: Predict the rating based on the business's average rating

This model assumes that the rating of a business does not change over time. The advantage of this model is similar to baseline 2. It can scale infinitely, but would require a massive amount of data for the model to be accurate. The downside is that either businesses have a lot of reviews or less than 10 reviews, which increases the variance of the data. The performance of this model is as follows. The training accuracy is 36.7%, validation accuracy is 32.01%, and the test accuracy is 25.69%. The conclusion for this baseline is that business ratings do not stay the same over time and suggests that external factors affect a business's rating over time.

Baseline model chosen: Baseline 1
We chose the text length as our baseline because it is easy to implement and computationally cheap to determine if future models are viable models. We consider a model viable if it yields an accuracy above 42%.

3. Models Description

Random Forest Classifier:

Random Forest classifier is a strong model to deal with numerical and categorical data. As discussed in the EDA section, we are not sure if "useful", "funny", and "cool" provided in the dataset are informative features for the prediction, so we include them, as well as review's length, year and month of the review to build a random forest classifier. To control overfitting, we apply cross-validation with 5 sub-samples to find the best combination of hyperparameters. Model parameters are:

max_depth = 50
n_estimator = 100
min_samples_leaf = 20

The training accuracy of the model is 48.5%, and the test accuracy is 43.6%, which is slightly better than the baseline model 1.
One disadvantage of this model is that it is slow and complex to get the best combination of hyperparameters. Also, it is computationally complex to find out which features are more informative for the classifier. However, the problem of overfitting is not obvious in our case since the accuracy of training and testing datasets do not vary a lot.

N-gram:

For this report, we use bi-gram and 5-gram separately as features for the model. To clean the text, we remove stop words, punctuation, and convert the text into lowercase. We also

compared the models with and without stemming words to determine whether stemming increases accuracy.

Train accuracy:

	With stemming	Without stemming
bi-gram	23.5%	23.4%
5-gram	23.2%	23.2%

Test accuracy:

	With stemming	Without stemming
bi-gram	33.9%	34.1%
5-gram	37.1%	37.5%

We concluded that stemming has a negligible effect on the accuracy of the models whereas the 5-gram model increased accuracy in the test set. A possible explanation for the increase in accuracy is that a 5-gram is better at capturing the context and connotation of the text than a bi-gram.

Due to the size of the dataset, it takes longer to train a 5-gram model than a bigram, which is one disadvantage when dealing with very complex and huge datasets. Another issue with both of the two models is the difficulty to clean the text data. For example, words/phrases like “fnc”, “excellen”, “rue” were caused by typos but were regarded as unique grams by the model.

5-gram and more with LightGBM:

Besides all the models above and the models we learned from the class, we also experimented with LightGBM. In the real world, LightGBM is widely used in NLP, which is similar to our text-mining algorithm. The features we choose are “useful”, “funny” and “cool”, as well as the most popular 1000 words in 5-gram without stemming, as shown in the previous part that the effect of stemming is negligible. Previously with Ridge in the sklearn linear model provided in lecture material, we achieved near 40% accuracy. This time, we will still use the first 90000 reviews for training, the following 10000 reviews for validating, and the following 100000 for testing. With the vanilla LightGBM classifier, the result achieves an accuracy of around 62.9% at validating stage, without any parameters tuning. We then further test the model we have on the dataset with a size of 100000, and the result stays above 61%, showing that our model is not overfitting the training dataset.

Then we test the features without the 1000-dimension vector representing the number of most popular 5-gram words, but only other features like “funny” and the length of the “text”, etc. The result significantly drops to 43.4% accuracy, showing that the 5-gram features are the most essential.

4. Literature search

In this section, we will talk about the literature we reviewed and how it inspired us in this project.

The review dataset we used, also known as the “Yelp Review Dataset” is officially released by Yelp and it has been commonly used for benchmarking tasks on recommendation systems. It has also been studied by machine learning researchers who focus their studies on sentiment analysis. This dataset plays an important role in educational institutions as well as in academia, as it is often used as a dataset to teach natural language processing and benchmarking. More recently, machine learning experts have also found another interesting and meaningful task using this dataset – the detection of fake reviews. This has brought more attention to the “Yelp dataset”.

During our research, we also found some similar datasets that were used in similar prediction tasks. These datasets are similar in the sense of scenarios (they all fall under the category of review data) and what it is to be predicted (star rating). One of them is the “restaurant reviews dataset” published by Carnegie Mellon

University (Sharifi, 2006). The "restaurant review dataset" also uses a 1-5 stars rating scale, which is identical to the Yelp Review Dataset. A closer examination of the "restaurant reviews dataset" also confirms one of our hypotheses: the existence of voluntary response bias in online reviews. (Figure 6)

Figure 6: Restaurant reviews dataset rating distribution. The restaurant reviews dataset has a similar rating distribution with the Yelp dataset (see figure 1), as reviewers dominantly responded either very positive (fours and fives), or extremely negative (ones). This behavior gives us more belief that the review data are often biased due to the fact that people are more likely to submit a review when they feel strongly (either positive or negative) about something. This is known as the voluntary response bias. (Cheung et al. 2017)

Due to the dominance of Yelp in the United States and Yelp's influential nature, much prior research has been conducted on it (Luca 2011). The same prediction task (star rating) has been done before by researchers in the field. Here are two of the state-of-the-art models that were applied to the Yelp dataset and yield the best results:

Model	Error rate (%, lower the better)
XLNet (Yang et al., 2019)	27.80
BERT_large+ITPT(Sun et al., 2019)	28.62

Table 2: Error rates were used as the metric to measure how successful a model is on the Yelp dataset. We choose to see this prediction task as a five-class classification problem and use the same metric inspired by them. Data from nlpprogress.com is presented in the table.

Given that XLNet and BERT both use transformers and XLNet is built upon BERT, we chose BERT as our study focus. BERT stands for Bidirectional Encoder Representations from Transformers, and it is a deep learning model made by Google. BERT's essential component is its transformers, and more specifically, its encoders. The groundbreaking innovation introduced by Google is that the encoders are bidirectional, meaning that instead of parsing a sentence by its natural ordering, a bidirectional encoder can take in a sentence of arbitrary length altogether (Figure 7). The parallel way of processing each word allows the model to understand the context of a sentence better compared to traditionally how sentences are parsed in a sequential way. The contexts of the words are thus determined not only by the words that come before but also by the one that follows (Devlin, Chang et al., 2018). This groundbreaking mechanism significantly boosted the results of many natural language processing-related tasks.

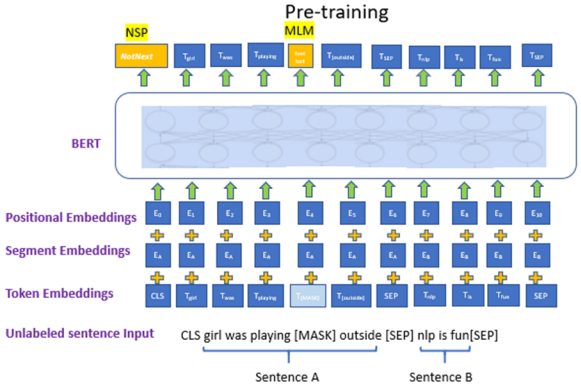

Figure 7: diagram of BERT Pre-Training with

MLM and NSP (and the bidirectional encoders)
Figure by Khandelwal, (2020).

5. Conclusion

In this report, we explored several models to predict the star ratings in Yelp reviews:
We tried some naive models such as linear regression with length of the review as the feature, predictions based on average rating of users, and predictions based on average rating on businesses.

We also explored and experimented with the more complex models, i.e. Random Forest Classifier, bi-gram and 5-gram predictor with Ridge classifier, and with LightGBM classifier.

Among all these models, we conclude that LightGBM on 5-gram predictor as the most successful model that has test accuracy up to 62.9%. Compared with it, we have less successful ones, including linear regressor (length of the review) and random forest tree classifier, and some unsuccessful models, such as the baseline models that use user's average rating or business's average rating, as well as the N-gram models. For baseline models, they reach pretty high train accuracy but they fail at validation and test dataset due to overfitting.

Here are some reasons that may have caused the unsuccessful models to underperform. One is our choice of metrics. For rating prediction, our regression models output float values but the dataset only has integer values for ratings. In real world applications, we are trying to find the exact ratings of a review based on its features instead of a statistically meaningful float number. Therefore, we tried to approach this by rounding our predictions to the nearest integers to check if the model predicts correctly. However, this is problematic for that the stars only scale up to 5, and therefore rounding stars with up to 0.5 greatly affects the accuracy.

On the other hand, the LightGBM model succeeds for its distinguished performance and scalability. In fact, we have also tried AutoGluon to find the best model, but the performance of our personal computers limits the capability to find the best model — the 16GB RAM is far from enough for fitting a 90000 reviews dataset. The efficiency of LightGBM helps keep us from waiting in front of the screen while maintaining accuracy, as its leaf-wise tree growth algorithm cuts off unusable branches, saving time and performance.

References:

Sharifi Mehrbod, Restaurant review dataset,, Carnegie Mellon University, (2006). http://www.cs.cmu.edu/~mehrbod/RR/

Cheung, K.L., ten Klooster, P.M., Smit, C. *et al.* The impact of non-response bias due to sampling in public health studies: A comparison of voluntary versus mandatory recruitment in a Dutch national survey on adolescent health. *BMC Public Health* 17, 276, (2017). https://doi.org/10.1186/s12889-017-4189-8

Luca, Michael. Reviews, Reputation, and Revenue: The Case of Yelp.com, Harvard Business School, (2011).

Ruder, Sebastian. NLP-progress.com http://nlpprogress.com/english/sentiment_analysis.html, (2021).

Khandelwal, Renu. Intuitive Explanation of BERT- Bidirectional Transformers for NLP. Toward Data Science, (2020). https://towardsdatascience.com/intuitive-explanation-of-bert-bidirectional-transformers-for-nlp-cdc1efc69c1e,

Devlin, J., Chang, M.-W., Lee, K. & Toutanova, K. (2018). BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding (cite arxiv:1810.04805Comment: 13 pages)