Sentiment Analysis for Amazon Fine Food Reviews

Team: Ziye Liu, JinzeWang, Duping Gao **Project Mentor TA:** Yufei Wang

1) Abstract

Nowadays, as people are more keen to leave their words on the internet, sentiment analysis becomes a heated topic. For this project, we aim to translate Amazon Fine Food reviews to raw scores, ranging from 1 to 5. If an accurate model can be designed, people can build a system that translates people's reviews to raw scores and then rank food based on those numerical scores to provide customers with good suggestions.

After doing in-depth research, our group decided to implement MLP as well as Bert. As it is a classification problem, we used one of the most traditional classification models, logistic regression model, as our baseline. Because of the nature of our data (highly skewed to the highest scores), purely using accuracy metric cannot reflect the result accurately. Therefore, we use F-1 score as our evaluation metric. It turned out that our baseline model logistic regression has an F-1 score around 0.46. As for MLP, the F-1 score is around 0.58 after fine tuning, higher than the baseline model. The most complex model Bert achieves an F-1 score of 0.62, which is the highest among all three models.

2) Introduction

We are using the data from Kaggle, named Amazon Find Food Review. The raw data includes 568,454 reviews and 10 columns. For the prediction purpose, we select the columns time, summary, and text as the inputs and score as the output. Time seems to be irrelevant to our model. However, after categorizing reviews based on the seasons (derived from the column time), we found that people tend to give the lowest score in winter while more neural scores in spring and summer, which means seasons affect people's behavior to some extent. Therefore, we decided to include it as a potentially useful feature and one-hot encoded it (however, later on, we showed that it was not a useful predictor for our models). As for summary and text features, we concatenate the two columns together, and then use a pretrained model 'all-MiniLM-L6-v2', a variant of the MiniLM model, to obtain the embeddings. We use this specific pretrained model since we are looking for a sentence-transformers model, which can map sentences and paragraphs to a multi-dimensional vector, and 'all-MiniLM-L6-v2' is the sample usage provided by the Website. Also, it is a smaller version of BERT, making it computationally more efficient than the BERT. Moreover, the model is trained on a diversified dataset including Wikipedia and the Web, making it more robust and suitable for our task.

Aside from using the Logistic Regression model as our baseline model, we build a MLP model using the embeddings obtained from the pretrained model. Finally, we fine tuned a BERT-base variant of Bert model, which consists of 12 layers in the encoder.

Since the data is imbalanced with more scores-5, we use F1 score to evaluate our result. To further solve the problem of imbalance, we downsampled our data with score-5. Our Machine Learning model, with the feature season, logistic regression achieves F-1 score of 0.46 for the test set after downsampling. After fine-tuning the MLP, we got a F-1 score of 0.57. When excluding the feature season, logistic regression and MLP have F-1 score of 0.46 and 0.58

respectively. The relatively same F-1 scores imply that the feature season did not improve the performance of our model. Therefore, we only consider reviews as input for the Bert model, which has an F-1 score of 0.62 after tuning, highest among all the models we selected.

2) How We Have Addressed Feedback From the Proposal Evaluations

In our first milestone, after data exploration, we extracted “season” as a potentially useful feature when predicting the scores. However, because the data type of that feature is not consistent with the shape of embeddings, we did not include that into our model. To address the problem, we concatenate one-hot encoding with embeddings. However, by comparing the F-1 score of models with and without F-1 score, we found that the feature season is not as useful as what we thought previously. The second problem is that our data is imbalanced. It is highly skewed to the highest score, 5, because the reviews came from fine food rather than the regular food. To solve this, we downsample our training data, we downsampled our data with score-5 to make data more balanced. We chose to keep half of the data with score 5, which left score-5 still the majority class, however less unbalanced. The reason behind that is, it is a dataset about fine tuning. Our target does tend to give higher scores, and we do not want to lose the information. However, since we observed a F-1 score below 0.5 while doing data shifting to test data, we realized that we do need to deal with imbalanced data. After downsampling data with score 5, we observed a smaller F-1 score difference on shifted test data. [Without downsampling, the f1 score on original test data is 0.429, while on shifted test data is 0.366, after downsampling, the difference of f1 score between the original test data and shifted test data is narrowed down to 0.035, from 0.463 to 0.428]. We are also advised to elaborate more on the selection of the pretrained model we are using, which are explained in the introduction.

3) Background

A. https://www.kaggle.com/code/shashanksai/text-preprocessing-using-python

This notebook gives many ways to generate embeddings, however, each of them has some shortcomings, for instance, Bad of Words method generates embeddings that are very sparse. The TF-IDF method may assign low values to words that are relatively important. Additionally, it is not possible for Word2Vec to generate embeddings for words it has never seen before. Thus, we need to think of better ways to generate embeddings.

B. https://www.kaggle.com/code/chirag9073/amazon-fine-food-reviews-sentiment-analysi

We found our data from this kaggle and we also planned to build our model on the top of it. The shortcoming of this relevant work is that in this notebook, score we need to predict is one-hot encoded, in which case, we punish those severe mistakes of misclassifying score 5 to score 1 the same amount compared to less severe mistakes of misclassifying score 5 to 4.

4) Summary of Our Contributions

Unlike other machine learning tasks, we used sentences as inputs, which requires the “translation” of text to data types that can be used to train various models. Our group used the most popular technique, embeddings. After obtaining embeddings from a pretrained model "all-MiniLM-L6-v2", which was imported from python library sentence-transformers, we used those embeddings as our major inputs for logistic regression and MLP.

We first trained a logistic regression model as our baseline model since we are predicting multiple labels. We then fine-tuned a MLP model by selecting the most appropriate batch size, learning rate, optimizer and architecture. For both the logistics regression model and MLP, we trained two versions of them, one with the feature season and one without the feature season to evaluate the usefulness of that feature. Since it turned out that season is not helpful in terms of achieving a higher F-1 score, we only used reviews as input to fine-tune a Bert-base model. Notice that in Bert, we did not use embeddings generated previously. Instead, we applied Bert tokenizer to re-preprocess the reviews to the format needed for the Bert model.

When using the feature season by concatenating with the embeddings, the F-1 score we get from the baseline model is 0.46. As for the deep learning model MLP, its F-1 score is 0.57 after fine-tuning learning rate, optimizer, batch size and other hyperparameters.

We trained a MLP model, which has a f1 score of 0.57 after fine-tuning learning rate, optimizer, batch size and other hyperparameters. Without using the feature season, the F-1 scores are 0.46 and 0.58 for logistic regression and MLP respectively. For the Bert, it has the highest F-1 score, which is 0.62 without using the feature season.

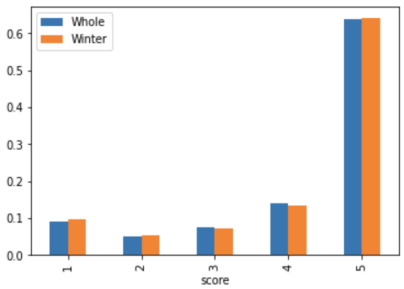
5) Detailed Description of Contributions

5.1 Implementation Contributions

Our project is Option 1. We are implementing different Machine Learning and Deep Learning models such as logistic regression, MLP, and Bert-base to predict scores from reviews.

During Exploratory Data Analysis, we find out that we can use season as a good feature for predicting, which can be derived from the original feature ‘time’. As can be seen from Figure 1 below, people tend to give the lowest score in winter while more neural scores in spring and summer. Therefore, we one hot encoded the ‘season’ feature and treated it as a candidate of the predictor. Apart from season, we also utilize text information. We combined column ‘summary’ and column ‘Text’ together, and then used a pretrained BERT model to generate embeddings. With the help of sentence-transformers, we are able to map our text data to a vector of length 384. After transforming all the features we want into numeric forms, we created two versions of the dataset to verify the usefulness of the feature season. One purely consists of text data as input and score as the output, and one uses both text and season as inputs through concatenating the one-hot encoded season feature and embeddings together into a 388-dim vector.

Figure 1: Relative percentage of people leaving certain score



After EDA and transforming the text data to embeddings, we then feed data into a Logistic Regression model for two versions of data separately. After that, we use a MLP model

with three fully-connected layers. We first tuned a MLP using both text and season as inputs. For the hyperparameters, we tune the batch_size, layer_size and learning rate. Additionally, we tried different optimizers, and found out that Adam has way better performance than the other optimizers. We applied similar logic when training the MLP with only the text data as the input.

For the third model, Bert-base, rather than building upon the embeddings, we re-preprocessed the text (reviews) by applying Bert tokenizer to make the input agree with the format needed for the Bert-case model. The next step is to fine tune the model. Unlike other models we trained, Bert-case is extremely expensive as it has 12 layers in the encoder. Thus, we only tuned the learning rate and batch size. As for the optimizer, as Bert is very complex, we used AdamW rather than commonly used Adam since it is an extension of Adam with weight decay, which prevents the problem of overfitting. For the epoch, we only used one because of the limited resources.

5.2 Evaluation Contribution

Our primary goal is to build models, which translates reviews in the format of sentences to raw scores ranging from 1 to 5. In addition to that, we want to verify whether the feature season, extracted from the variable time for the original dataset, is useful in the prediction or not. As suggested by the name of the dataset, ‘fine food’, our dataset is highly imbalanced, skewed to the highest score, 5, as shown by figure 2. Therefore, we selected F-1 score as our evaluation metric because the most commonly used performance metric accuracy does not take the problem of imbalance into account. As mentioned previously, our dataset came from fine food reviews. Therefore, we consider another possible scenario where the scores are more evenly distributed as the dataset shifts (Figure 3). We obtain the “new” data by changing the proportion of the original data. After data shifting, the F-1 score of our test set for the logistic regression decreases from 0.46 to 0.428, which is tolerable.

When comparing different models, model complexity, hyperparameters, and preprocessing are responsible for a model to perform better than others, and our group took those factors into account when building our model. There are many existing machine learning models. Since we are aiming to solve a classification problem, we limit the scope to classification models only. The data type of the inputs, sentences, further reduces the number of possible models. To make the sentences usable in various models in the first place, we need to create embeddings for them. Because of the limited size of our data and the computing power, it is almost impossible for us to train a brand new model starting from scratch for the purpose of embeddings. Therefore, we decided to use pretrained models. Among many pretrained models, we used "all-MiniLM-L6-v2" because it is designed for sentences rather than words, which is especially important for review type of sentences. Its efficient computation is another reason. After embedding, each review has high dimensions. Therefore, we only consider complex models to avoid overfitting. Therefore, we consider the MLP model. After preprocessing the data appropriately, we fine-tuned them. Lastly, we fine tune a state-of-art model, Bert-base.

Results are shown in the table1 below. For the question related to the usefulness of season, it turned out that it is not a useful predictor since based on the F-1 score from logistic regression and MLP, adding season as an additional feature did not help with improving the performance, which contradicts with our previous assumption. One possible explanation is that people’s mood influenced by the season has already been reflected in the text. Therefore,

adding season did not add more information. Another reason is related to the size of inputs. After one-hot encoding and concatenation, season is expressed using only 4 numbers such as 0,0,0,1, which is small compared to the size of embeddings for the whole sentence. Therefore, unless the season has a significant influence on the result, its impact on performance is not clear. For our major question, applying various models to predict scores from reviews, both MLP and Bert-case have better performance compared to the baseline model. Even though the differences are minor, Bert-case model has the best performance, followed by MLP (Table 1). Possible reason is that we did not truly fine tune the Bert model given limited resources. It is possible that if we train more epochs and try more combinations of parameters such as batch size and learning rate, the F-1 score will be improved significantly.

Table 1: F-1 score for 3 models

	Logistic regression	MLP	Bert-case
Test F-1 score(with season)	0.46	0.57	–
Test F-1 score (no season)	0.46	0.58	0.62

Figure 2: Original distribution of score

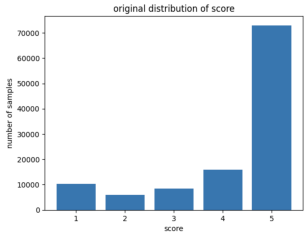
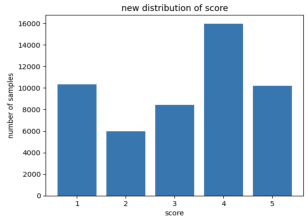


Figure 3: Distribution of score after data shifting



5.3 Any difficulties encountered

First, the BERT model is complex and has many parameters, and thus it takes a long time to train. Second, neither accuracy nor f1 score is the best choice to evaluate our model's performance, since the punishment made to wrongly classify 5 as 4 compared to wrongly classify 5 as 1 should be different, but both accuracy and f1 score can not take this into consideration. Finally, people's scoring and comments are subjective, it might be the case that the comment itself is a very positive one, while the user only gives a score of 4 since he/she is conservative. It is hard for us to achieve good performance considering this.

6) Compute/Other Resources Used

We primarily use Google Colab as the computing resource because of its shareability, which is useful in building models in groups. We also use various packages from Python such as sklearn, torch, and sentence_transformers to not only generate the embeddings for our sentences but also train our models.

7) Conclusions

This project aims to develop a model that can predict raw scores using reviews from websites. We build three models, logistic regression, MLP and Bert-base. Among them, Bert-base has the best performance in terms of the highest F-1 score using only 1 epoch. In addition to that, we try to evaluate the usefulness of seasons in the prediction since it seems to be a useful predictor during the EDA. However, the result contradicts our assumption. One important lesson is that even though EDA can provide important information, because of the complexity of the world, it is always better to try to evaluate the result before taking it.

In hindsight, we tried to build a CNN model with generated embeddings at the very beginning, however, since the embedding we generated is one dimensional, which contains little position information which can be utilized by CNN model. Therefore, we switched to the idea of building a BERT model as our advanced model. Additionally, we want to utilize the feature 'season' as discovered in the EDA session. However, after comparing the training result on Logistic Regression and MLP model of data with and without season as feature in different ways (Method 1: Concatenate season to Text as a string and then generate the embeddings, which was suggested by Yufei. Method 2: One-hot encoded season feature and concatenate it with Text embeddings), we ended up with a conclusion that season may not be as useful as we thought. Moreover, we thought that little work should be done considering imbalanced data, since the data set tends to generate more positive rating scores. However, after we added data shifts to our testing data, we realized that we should get more insight into imbalancing. We really appreciate Yufei's help during the whole process, she gives us suggestions on how to better utilize the season as a feature and let us know more about the pretrained model we choose. As for the future, we recommend training more epochs with BERT.

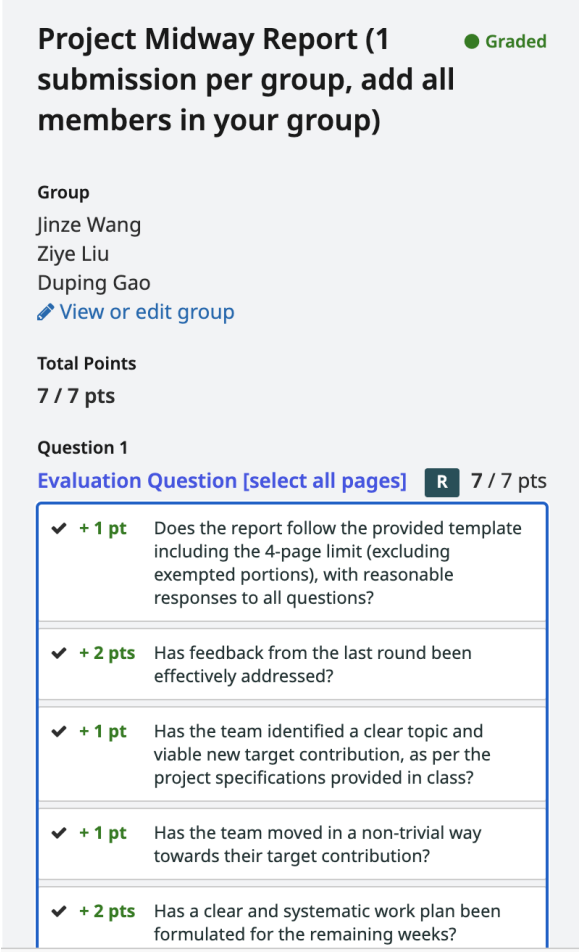
From the perspective of the dataset itself, our project does not have any ethical consideration since we did not use any personal information such as the profile name and gender in our model, making it impossible to expose customers' privacy. However, the model itself might be unfair to some restaurants. It is highly possible that if the model is widely used and customers heavily rely on the scores predicted by the model to decide which restaurant they are going to, owners of restaurants will notice that. To attract more customers, some owners tend to deploy the model by hiring people leaving fake "good" reviews on the website to achieve a higher score, which is unfair to startup restaurants. Since our model is unable to detect and filter out the fake reviews, fairness is not guaranteed. Another potential problem associated with our model is its possible negative impact on the environment. Since the inputs are sentences, which have high dimensions compared to other types of inputs, training an accurate model using large amounts of training data requires strong computing power especially if we want to create useful embeddings ourselves instead of relying on pre-trained models, which will possibly negatively affect the environment.

8) Roles of team members:

Ziye Liu came up with the idea of using 'all-MiniLM-L6-v2' to create embeddings and build MLP on top of it. Jinze Wang pre-processed the data using Bert tokenier and implemented the Bert-base model on top of that. Duping Gao evaluated the three models and the usefulness of the feature season and then wrote the report.

(Exempted from page limit) Other Prior Work / References (apart from Sec 3) that are cited in the text:

(Exempted from page limit) Attach your midway report here, as a series of screenshots from Gradescope, starting with a screenshot of your main evaluation tab, and then screenshots of each page, including pdf comments. This is similar to how you were required to attach screenshots of the proposal in your midway report.



Project Midway Report (1 submission per group, add all members in your group) ● Graded

Group
Jinze Wang
Ziye Liu
Duping Gao
View or edit group

Total Points
7 / 7 pts

Question 1
Evaluation Question [select all pages] R **7** / 7 pts

✔ + 1 pt	Does the report follow the provided template including the 4-page limit (excluding exempted portions), with reasonable responses to all questions?
✔ + 2 pts	Has feedback from the last round been effectively addressed?
✔ + 1 pt	Has the team identified a clear topic and viable new target contribution, as per the project specifications provided in class?
✔ + 1 pt	Has the team moved in a non-trivial way towards their target contribution?
✔ + 2 pts	Has a clear and systematic work plan been formulated for the remaining weeks?

responses to all questions?

✔ **+ 2 pts**	Has feedback from the last round been effectively addressed?
✔ **+ 1 pt**	Has the team identified a clear topic and viable new target contribution, as per the project specifications provided in class?
✔ **+ 1 pt**	Has the team moved in a non-trivial way towards their target contribution?
✔ **+ 2 pts**	Has a clear and systematic work plan been formulated for the remaining weeks?

Great job guys. Glad to see your progress. I really like your data analysis part about season as a potential influential feature and missing data. I guess I would suggest you guys to elaborate more about the pretrained model you are using, like what kind of architecture does "all-MiniLM-L6-v2" have (I assumed that you found it in a kaggle submission? This is okay but you have to justify more about why this is suitable). Also could you guys clarify your plan about including the season feature? Because it seems like you then focus on using embeddings generated by pretrained language models. How are you incorporating these features at the same time? Do you just ignore it for the BERT model which seems to be the main contribution you are going to work on for the next milestone? I do agree with the imbalance data treatment part you mentioned since investigating data shift can also be an important part of your project. It would be great if you guys can stop by my office hour or we can arrange a meeting to briefly discuss these points.

Sentiment Analysis for Amazon Fine Food Reviews

Team: Jinze Wang (CIS 5190) , Ziye Liu (CIS 5190) , Duping Gao (CIS 5190) .
Project Mentor TA: Yufei

1) Introduction
Our group aims to use the reviews (sentences) to predict rating score (1 is the lowest score and 5 is the highest score). We are using the data from Kaggle, named Amazon Find Food Review. The raw data includes 568,454 reviews from October 1999 to October 2012. There are 10 columns in the original dataset, which includes id, product id, user id, profile name, number of users who found the review helpful, number of users who indicated whether they found the review helpful or not, score, time, summary, and text.

For the prediction purpose, we select the columns time, summary, and text as the inputs and score as the output. Time seems to be irrelevant to our model. However, we still decided to include it because after categorizing reviews based on the seasons (derived from the column time), we found that people tend to give the lowest score in winter while more neural scores in spring and summer, which means seasons affect people's behavior to some extent. Therefore we include season as a feature and one-hot encodes it. There are 27 missing values in the column summary. Among those rows which contain null values, we witnessed abnormalities. Since it turns out that there is one customer who comments on different products with the same reviews at the same time. We decided to drop these rows. For the columns summary and text, we combine it as one column named full text since they contain similar information and we want to extract information from both of them. Also, we further use the pretrained model 'all-MiniLM-L6-v2' to obtain the embeddings of the full text and use it as one of the inputs. For the output, we do not one-hot encode it and leave it as it is since more punishment should be made if we wrongly assign Score-5 to Score-1 compared to wrongly assigning Score-5 to Score-4.

Implementation:
For the implementation, we first use a pretrained model 'all-MiniLM-L6-v2' as the sentence transformer to create the embeddings for the reviews. Then, use the embedding as inputs to train models including logistic regression, CNN, decision tree, multi-layer perceptron (MLP). Among those models, we treat logistic regression as the baseline model.

Evaluation:
Since the data is imbalanced with highly more scores of 5, we use F1 score to evaluate our result. We do not downsample our data to make data balanced because the target of the reviews is fine dining, and thus compared to the normal restaurants, they are more likely to receive high ratings. Therefore, to maintain the distribution, we decide to use a different evaluation method rather than downsampling the data. [Update: after observing f1 score below 0.5, we think we should pay attention to imbalance data, we will handle this before the next milestone. Currently, we are thinking of keeping 50% data for score-5, which can mitigate the

problem of imbalancing to some extent, and at the same time contain useful information(fine dining datasets tend to give high scores) for future prediction.]

2) How We Have Addressed Feedback From the Proposal Evaluations

For milestone 1, we mentioned that we planned to use existing sentiment analyzer and other pretrained models as one of our models. However, we did not explicitly state how we are going to use them. Considering the limited resources, we are going to use a pretrained model 'all-MiniLM-L6-v2' to create the embeddings for the reviews as our initial step, and the resulting embeddings are used as input to train different models such as MLP. In the future, we will try to fine tune the bert model.

3) Prior Work We are Closely Building From

A. https://www.kaggle.com/code/nayansakhiya/text-classification-using-bert
   We searched for many pre-trained models to create the embeddings for our text and we finally decided to use BERT to preprocess our text data.
B. https://www.kaggle.com/code/chirag9073/amazon-fine-food-reviews-sentiment-analysis
   We found our data from this kaggle and we also planned to build our model on the top of it.

4) Contributions

Our group has already finished data pre-processing including feature selection, one-hot encoding the feature "season", dealing with nulls, and creating text embedding for the input variable full text using pretrained model 'all-MiniLM-L6-v2', and train-test split.

4.1 Implementation Contributions

The first model we implemented is a pretrained model, BERT, which is used in the text preprocessing stage. In milestone1 we considered using Bag of Words to generate word embedding, however, we later learnt in class that the resulting vector will be very sparse. TF-IDF is not a good choice considering that it may assign low value to rather important words. After comparing Word2Vec, we decided to use BERT from sentence transformers to generate text embedding which contains position information.

We also have finished implementing the baseline model logistic regression from sklearn. We would like to build a MLP, since it has a good performance in extracting features from data. Currently, we only built a simple MLP with 3 layers, 10 epoches, batch size of 64, and a learning rate of 0.001. We will future fine tune the MLP model to select the "best" architecture by trying different architectures, learning rate, and batch size. In the future, we will also train a CNN model, with Conv2d, pooling layer, normalization layer, dropout layer, fully-connected layer and activation layers, we are going to explore the appropriate kernel size, dropout fraction and which activation function to use, whether to use a maxpooling layer or an avgpooling layer instead.

Currently we have implemented word embeddings using the pretrained BERT model, and feed those embeddings into some machine learning models. But we think we can further improve our predictions using a fine-tuned BERT model. We will build a BERT classifier model by adding a pretrained BERT model into our neural network architecture, so that we can update weights of the BERT model during the training process. Since the BERT model may have a very complex structure which will consume lots of computing power to train, we may need to utilize AWS or other possible methods to accelerate the training process.

4.2 Evaluation Contribution

Given the text of a food review, we want to predict the rating(1-5) the reviewer gave to that food. So it is a multi-class classification problem. And since the labels(rating) are unbalanced, we will use F1 score as our metrics.

The baseline model is the logistic regression with an F1 score of 0.428 on the test set. We also train a simple MLP (without fine tuning), which has an F1 score of 0.50 on the test data set. Without tuning, MLP has slightly better performance on the test set compared to the baseline model logistic regression. We do not consider dealing with imbalance in the first place, since we think that the dataset itself has a tendency to predict more positive scores, and since it can predict scores other than 5, we do not take imbalance seriously. However, the bad performance illustrated by the f1 score made us think that imbalance should be handled. We will handle this problem before the next milestone.

Since our original data is imbalanced by its nature, skewed to a score of 5 (Figure 1), we change the distribution of the test data by decreasing the proportion of data with output of score of 5 (Figure 2). With the data shift, the F1 score for our baseline model logistic regression decreases from 0.428 to 0.364, which implies that our F1 score is somehow influenced by the data with output of score of 5 and with more training data with score of 5, our model tends to predict the score as 5. However, if the model is used to predict the score for fine dining, it is still reasonable.

5) Risk Mitigation Plan

The hardest part or the most special part of our project is how to make raw text usable in various machine learning algorithms. The most common approach is to create the embeddings and use them instead as the inputs. However, the creation of embeddings takes up lots of resources because of its high dimensionality. To mitigate the risk, our group starts from a simplified setting to at least have some early results by first using a pretrained model BERT to create the embeddings for the text.

If time allowed, we are going to fine tune the BERT model. During this process we may encounter some problems like limited computing resources. And we may find some difficulties when we run our code using AWS resources.

In our exploration, we find out that our MLP model does not outperform the baseline logistic regression model very much. So it is possible that our complex model may not be able to generate better predictions. However, we will further fine tune the complex model to verify our assumption.

(Exempted from page limit) Supplementary Materials if any (but not guaranteed to be considered during evaluation):

Link to the repository:
https://github.com/Zoey-ZiyeLiu/Amazon_Fine_Food_Reviews