USC Viterbi
School of Engineering

ISE 540 Text Analytics Project:

Yelp Review Helpfulness Prediction Based on Multiple Features

Jingzhi Zhou
Yidan Sun
Fandi Ma
Jiade Song

Repository: https://github.com/toyhtoza/Yelp-Review

Part 1: Executive Summary

1.1 Motivating Background

The development and popularization of mobile e-ecology have reached an unprecedented scale. As the leading e-commerce platform for life services, Yelp has connected millions of businesses and hundreds of millions of users, accumulating billions of genuine user reviews. Reviews from users provide a reference for consumers to make consumption decisions and are also an important channel for merchants to obtain consumer feedback. A user may only read a limited number of reviews before deciding. However, low-quality reviews can often cause inconvenience to review readers. Under such circumstances, our goal is to construct a system to predict the usefulness of yelp reviews and recommend useful ones to users to improve user experience. Besides identifying which reviews are "useful," this paper also explores the essential factors that make reviews helpful to a consumer in the process of making a purchase decision.

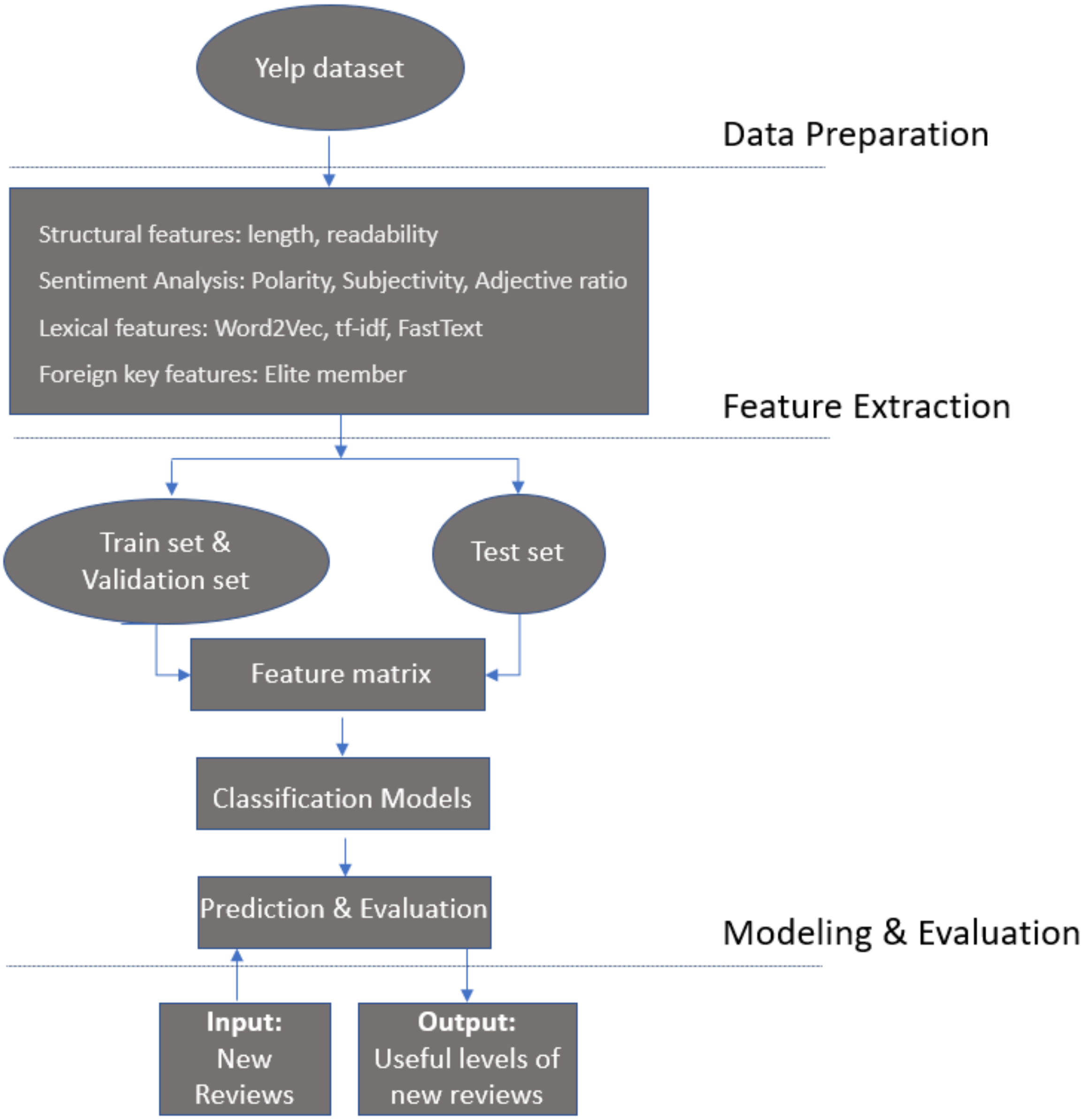
1.2 Solutions

This paper analyzes 55187 reviews from the Yelp restaurant dataset to explore over 20 features' effects on review helpfulness. Besides the features initially included in the dataset, e.g. useful rates, stars, etc., we have mined ten more linguistic and semantic features to examine their impact on the number of helpfulness reviews received, e.g. readability, polarity, subjectivity, etc. Various text mining methods were used to determine the best model for Yelp restaurant's review helpfulness, including TF-IDF, Word2Vec, and FastText. The results show that reviews for different product types have other psychological and linguistic characteristics, and the factors affecting the review helpfulness are also various. This paper sufficiently discusses the implications of our findings for each implementation, both in theory and in practice.

1.3 Outcomes

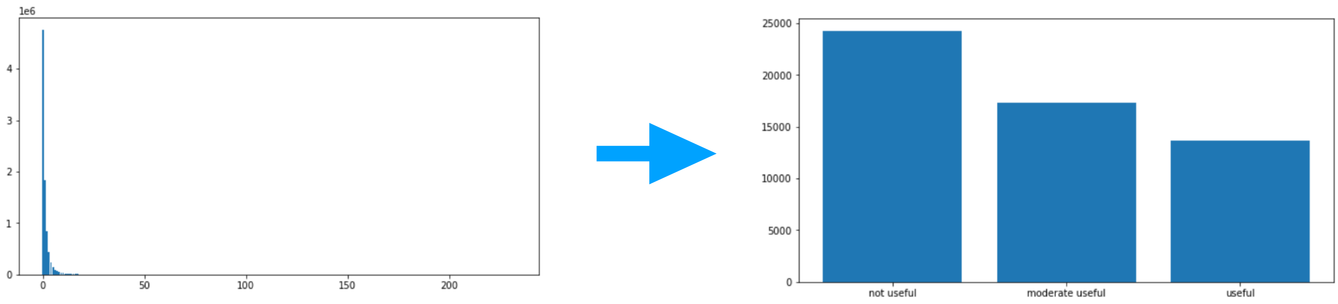
We conducted eight major experiments by combining these features and using classifiers, including Logistic Regression, XGBoost, GaussianNB, and Gradient Boosting Classifier. We examined their effectiveness by comparing the results using Accuracy, F1 Score, Precision, Recall, AUC, and Training Time. Among all the models, **the XGBoost model utilizing TF-IDF(lexicon) features with all other features** reaches the highest test accuracy of 70% but takes up the longest time. If we use precision as our metric, **the XGBoost model utilizes TF-IDF(lexicon) features with all other features. The XGBoost model utilizes all other features only,** and **FastText** gives equally good results of 0.69. To our surprise, the FastText method only used a training time of 1.28s.

Part 2: Proposed Approach


Yelp dataset
Data Preparation
Structural features: length, readability
Sentiment Analysis: Polarity, Subjectivity, Adjective ratio
Lexical features: Word2Vec, tf-idf, FastText
Foreign key features: Elite member
Feature Extraction
Train set & Validation set
Test set
Feature matrix
Classification Models
Prediction & Evaluation
Modeling & Evaluation
Input: New Reviews
Output: Useful levels of new reviews

2.1 Data Preparation

The original Yelp Review Dataset (https://www.yelp.com/dataset) contains over 8630000 records. We extracted a subset of the original dataset for this project for efficiency and divided the reviews into 3 groups by their "useful" votes. Specifically, reviews with 0 or 1 vote were labelled "not useful"; reviews with 2-20 votes were labelled "moderate useful"; reviews with over 20 votes were labelled "useful". Because the reviews with votes less than 5 takes up the majority of the original dataset, we kept all the reviews with more than 29 votes and randomly extracted a fraction of reviews from each level to provide each useful level enough data to analyze with. The sampled dataset contains 55187 Reviews.



The subsequent data cleaning process includes dropping null values, screening out reviews only in English, removing extra spaces and newlines, removing punctuation marks, converting text to lowercase, stemming, and finally, lemmatization. The cleaned version of review texts is kept as a new column called “Text_clean” and used for feature extraction and word embedding.

2.2 Feature Extraction

The features we mined from the reviews for prediction are categorized into four types: lexical features, structural features, sentiment features, and foreign key features.

- **Lexical features**: Indices generated using word-embedding techniques, which are TF-IDF, Word2Vec, and FastText.
- **Structural features** : The word count of each review; the readability of the review, measured by the *Flesch Reading Ease* method that gives a text a score between 1 and 100, with 100 being the highest readability score.
- **Sentiment features**: Polarity, which lies between [-1,1] with -1 representing a negative sentiment and 1 representing a positive sentiment; Subjectivity, lying between [0,1] where a higher score means more personal opinions instead of factual information; the ratio of adjectives to the number of words of each review.
- **Foreign key feature**: The number of years the reviewer has been an elite member of Yelp.

Together, structural, sentiment analysis, and foreign key features are called “Other Features” in this project, in opposition to the word-embedded matrix that directly represents for the cleaned review texts.

2.3 Models

2.3.1 Proposed model

To train and evaluate models, the selected 55187 Reviews are split into three parts: Train, Test and Validation sets, stratified by the y value “useful_level”. We set 72% of the dataset as the train set, 18% as the validation set, and 20% as the test set.

- **Proposed model—Word2Vec with Other Features - Single Classifier: A** single classifier to predict useful levels using the combination of Other Features and Word2Vec embedded matrix features. Various classifiers are trained separately and then compared to select the best classifier for this proposed method.

2.3.2 Classifiers for fitting the models

The classifiers we trained for fitting the models are:

- **Logistic regression** is used for finding the linear relationship between binary classes. However, with multinomial logistic regression we could apply that with our case (that is 3 classes). If there exists a linear relation between them, it could be a good fit for the model.
- **Naive Bayes** is to determine what the class this data belongs to. It is classified by the probability of a given event and the events happen independently. This model will calculate in what probability this event will occur. Our data happens independent which means the choice is reasonable.
- **Decision Tree** is to break the data into smaller datasets. Based on each subset the data falls, it could determine which class the data belongs to. If decision trees are connected, we could use random forest as an ensemble. Our data have lots of features and 3 classes, which could be a good fit to use random forests.
- **XGBoost** is an updated version of decision trees. It is optimized by updating the decision trees using the loss function of every tree, so it has a higher accuracy than traditional trees. However, this model always takes longer to train the data.
- **KNN** is also a good classifier, it classifies data by finding small groups and updating the group center by computing the errors.

2.3.3 Baseline models

To verify if our proposed model is the optimal solution, we perform predictions using the other eight baseline models for the comparison:

1. **Random assignment**: Randomly assign the reviews to the three classes of "useful_level" following the idea of Multinomial distribution.

2. **Other Features only**: Consider only Other Features, i.e., structural, sentiment analysis, and foreign key features. Train and fit a single classifier to predict the useful level for each review. The best classifier is picked from Logistic Regression, XGBoost, GaussianNB, Gradient Boosting Classifier.

3. **TF-IDF only**: Use the TF-IDF technique to transform the cleaned texts to the matrix and then use the best classifier to predict the useful levels based on the TF-IDF matrix. The best classifier is selected from Naive Bayes, Logistic Regression, Decision Tree, KNN, XGBoost, Random Forest, and MLP with different input parameters.

4. **TF-IDF with Other Features - Single Classifier**: The same process as our proposed model but uses TF-IDF for word embedding instead of Word2Vec. The best classifier is picked from Logistic Regression, Decision Tree, KNN, XGBoost, Random Forests and MLP.

5. **TF-IDF with Other Features - Two-layer Blending**: Use the combination of other features and the TF-IDF matrix for prediction. The method for classifier here follows the blending technique of ensemble learning and has two layers. First, we train two classifiers, one separately on the TF-IDF matrix and the combination of the Other Features, which generates the first layer of prediction result. Then, the second layer of the learner using logistic regression is performed to validate the prediction and pass the final result. Four combinations of classifiers for the first layer are compared: Logistic Regression & XGB, Gradient Boosting Classifier & XGB, GaussianNB & Gradient Boosting Classifier, XGB & Random Forest Classifier.

6. **Word2Vec**: The same process as Baseline Model 2 but uses Word2Vec to generate the word embedded matrix. The classifiers being compared are the same as Baseline Model 2.

7. **Word2Vec with Other Features- Two-layer Blending**: The same process as Baseline Model 5 but uses Word2Vec to generate the word embedded matrix. Four combinations of classifiers for the first layer are compared: Logistic Regression & XGB, Random Forest Classifier & XGB, Logistic Regression & Random Forest Classifier, Gradient Boosting Classifier & XGB.

8. **FastText**: Use FastText to automatically complete the text classification process on the cleaned review texts. No other features or classifiers are used to generate the prediction.

Part 3: Experimental Results

3.1 Metrics

	Description
Accuracy	(TP+TN)/(TP+FP+FN+TN) Ratio of correctly predicted observation to the total observations.
F1 Score	2*(Recall * Precision) / (Recall + Precision) Weighted average of Precision and Recall
Precision	TP/(TP+FP) Ratio of correctly predicted positive observations to the total predicted positive observations
Recall	TP/(TP+FN) Ratio of correctly predicted positive observations to all observations in actual class
AUC	measures the area under the ROC curve
Training Time	Measures the time of each classifier during training process

We applied the above metrics and evaluated models using our validation set. We firstly evaluated the model's overall performance on the dataset and then we moved to model evaluation on each class of y (label set: usefulness levels).

3.2. Evaluation

- **Ground Truth:** Reviews with 0 or 1 vote are in level "not useful" and we give this class label 0; reviews with 2-20 votes are in level "moderate useful" and we give this class label 1; reviews with over 20 votes are in level "useful"and we give this class label 1.

3.2.1 Random Model

- **Overall evaluation**

Accuracy	F1 Score	Precision	Recall
0.35	0.33	0.33	0.33

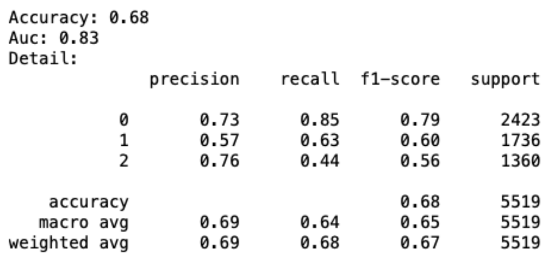
3.2.2 Other Features Only

- **Overall evaluation**

Classifiers	Accuracy	AUC	F1 Score	Precision	Recall	Training Time (s)
Logistic Regression	0.57	0.76	0.46	0.5	0.5	0.73
XGBoost	**0.68**	**0.83**	**0.65**	**0.69**	**0.64**	**4.76**
GaussianNB	0.58	0.77	0.5	0.56	0.51	0.039
Gradient Boosting Classifier	**0.65**	**0.81**	**0.61**	**0.65**	**0.61**	**7.31**

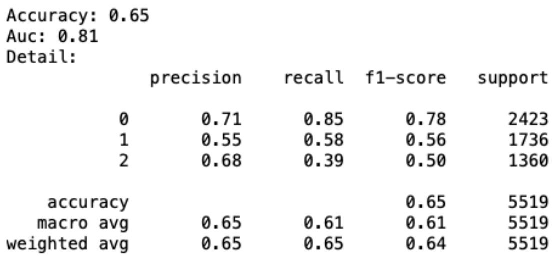
- **The best 2 classifiers**

XGB:



```
Accuracy: 0.68
Auc: 0.83
Detail:
              precision    recall  f1-score   support

           0       0.73      0.85      0.79      2423
           1       0.57      0.63      0.60      1736
           2       0.76      0.44      0.56      1360

    accuracy                           0.68      5519
   macro avg       0.69      0.64      0.65      5519
weighted avg       0.69      0.68      0.67      5519
```

Gradient Boosting:



```
Accuracy: 0.65
Auc: 0.81
Detail:
              precision    recall  f1-score   support

           0       0.71      0.85      0.78      2423
           1       0.55      0.58      0.56      1736
           2       0.68      0.39      0.50      1360

    accuracy                           0.65      5519
   macro avg       0.65      0.61      0.61      5519
weighted avg       0.65      0.65      0.64      5519
```

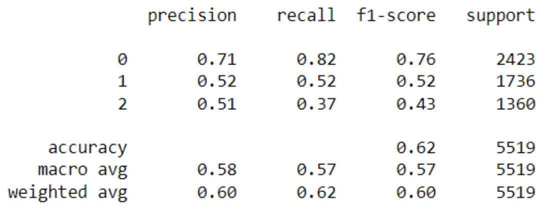
3.2.3 TF-IDF

- **Overall Evaluation**

Classifiers	Accuracy	AUC	F1 Score	Precision	Recall	Training Time (s)
Naive Bayes	0.56	0.72	0.4885	0.5432	0.5026	1.77
Logistic Regression	**0.62**	**0.78**	**0.5706**	**0.5801**	**0.5706**	**9.95**
Decision Tree	0.51	0.61	0.4749	0.4755	0.4748	51.31
KNN	0.47	0.65	0.3284	0.4794	0.3866	51.63
XGBoost	**0.61**	**0.78**	**0.5566**	**0.5783**	**0.5592**	**118.17**
Random Forest	0.59	0.77	0.5043	0.5796	0.5277	197.5
MLP	0.58	0.74	0.5497	0.5501	0.5495	313.48

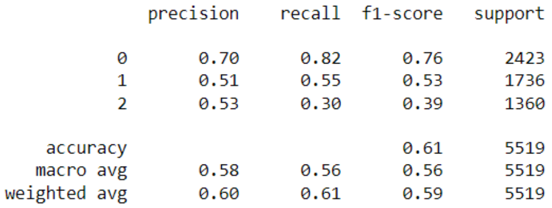
- **The best 2 classifiers**

Logistic Regression:

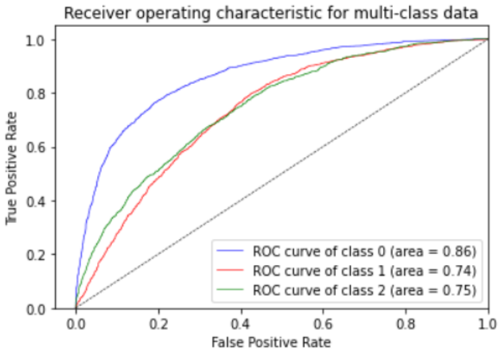


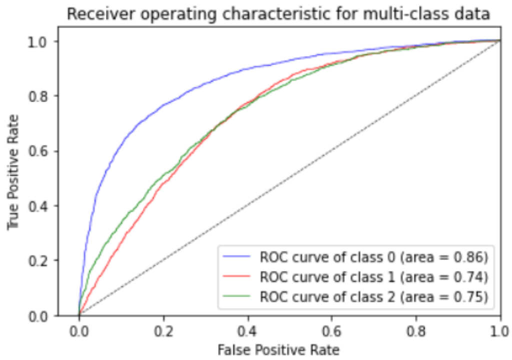
```
              precision    recall  f1-score   support

           0       0.71      0.82      0.76      2423
           1       0.52      0.52      0.52      1736
           2       0.51      0.37      0.43      1360

    accuracy                           0.62      5519
   macro avg       0.58      0.57      0.57      5519
weighted avg       0.60      0.62      0.60      5519
```

XGB



```
              precision    recall  f1-score   support

           0       0.70      0.82      0.76      2423
           1       0.51      0.55      0.53      1736
           2       0.53      0.30      0.39      1360

    accuracy                           0.61      5519
   macro avg       0.58      0.56      0.56      5519
weighted avg       0.60      0.61      0.59      5519
```





3.2.4 Fasttext

- **Overall Evaluation**

Fasttext reached precision 0.7

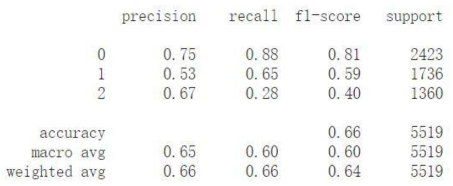
3.2.5 TF-IDF with other Features (Single Classifier)

- **Overall Evaluation**

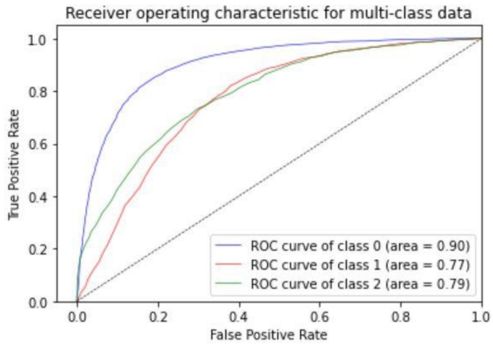
Classifiers	Accuracy	AUC	F1 Score	Precision	Recall	Training Time (s)
Logistic Regression	0.65	0.8	0.5785	0.6325	0.5866	123.32
Decision Tree	0.6	0.68	0.5693	0.5696	0.5692	38.77
KNN	0.63	0.75	0.5878	0.6096	0.5842	69.29
XGBoost	**0.7**	**0.85**	**0.6612**	**0.6899**	**0.6562**	**94.36**
Random Forest	**0.66**	**0.82**	**0.5976**	**0.6525**	**0.6035**	**71.19**
MLP	0.64	0.80	0.6011	0.6049	0.6059	493.13

- **The best 2 classifiers to evaluate by class**

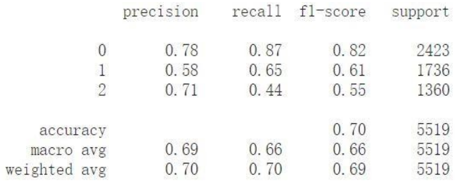
XGBoost:



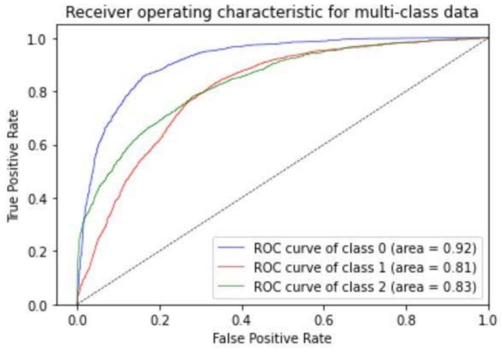
```
              precision    recall  f1-score   support

           0       0.75      0.88      0.81      2423
           1       0.53      0.65      0.59      1736
           2       0.67      0.28      0.40      1360

    accuracy                           0.66      5519
   macro avg       0.65      0.60      0.60      5519
weighted avg       0.66      0.66      0.64      5519
```



Random Forest:



```
              precision    recall  f1-score   support

           0       0.78      0.87      0.82      2423
           1       0.58      0.65      0.61      1736
           2       0.71      0.44      0.55      1360

    accuracy                           0.70      5519
   macro avg       0.69      0.66      0.66      5519
weighted avg       0.70      0.70      0.69      5519
```



3.2.6 TF-IDF with other Features (Two-Layer Blending)

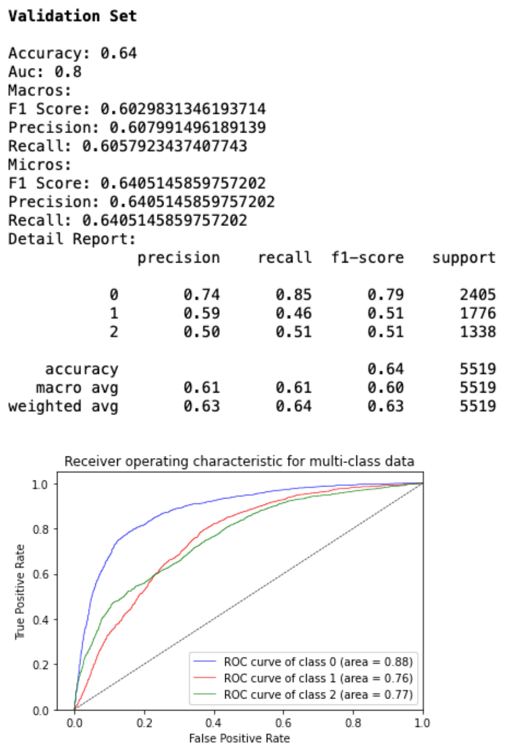
- **Overall Evaluation**

Classifiers	Accuracy	AUC	F1 Score	Precision	Recall	Training Time (s)
Logistic Regression & XGB	**0.65**	**0.81**	**0.61**	**0.61**	**0.61**	**19.15**
Gradient Boosting Classifier & XGB	0.63	0.80	0.58	0.60	0.58	127.31

GaussianNB & Gradient Boosting Classifier	0.63	0.79	0.57	0.61	0.58	11.94
XGB & Random Forest Classifier	**0.7**	**0.85**	**0.6612**	**0.6899**	**0.6562**	**94.36**
MLP	0.64	0.80	0.6011	0.6049	0.6059	493.13

- **The best 2 classifiers**

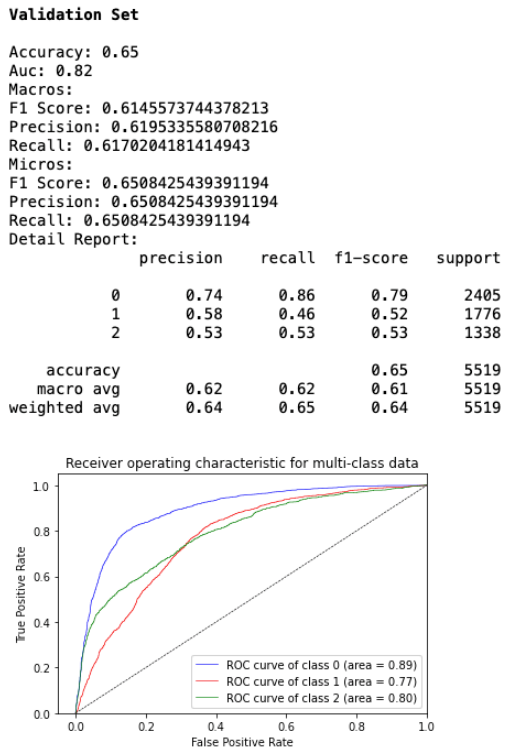
XGB & Random Forest Classifier:



```
Validation Set

Accuracy: 0.64
Auc: 0.8
Macros:
F1 Score: 0.6029831346193714
Precision: 0.607991496189139
Recall: 0.6057923437407743
Micros:
F1 Score: 0.6405145859757202
Precision: 0.6405145859757202
Recall: 0.6405145859757202
Detail Report:
              precision    recall  f1-score   support

           0       0.74      0.85      0.79      2405
           1       0.59      0.46      0.51      1776
           2       0.50      0.51      0.51      1338

    accuracy                           0.64      5519
   macro avg       0.61      0.61      0.60      5519
weighted avg       0.63      0.64      0.63      5519
```

Receiver operating characteristic for multi-class data
True Positive Rate
False Positive Rate
0.0
0.2
0.4
0.6
0.8
1.0
ROC curve of class 0 (area = 0.88)
ROC curve of class 1 (area = 0.76)
ROC curve of class 2 (area = 0.77)

Logistic Regression & XGB:



```
Validation Set

Accuracy: 0.65
Auc: 0.82
Macros:
F1 Score: 0.6145573744378213
Precision: 0.6195335580708216
Recall: 0.6170204181414943
Micros:
F1 Score: 0.6508425439391194
Precision: 0.6508425439391194
Recall: 0.6508425439391194
Detail Report:
              precision    recall  f1-score   support

           0       0.74      0.86      0.79      2405
           1       0.58      0.46      0.52      1776
           2       0.53      0.53      0.53      1338

    accuracy                           0.65      5519
   macro avg       0.62      0.62      0.61      5519
weighted avg       0.64      0.65      0.64      5519
```

Receiver operating characteristic for multi-class data
True Positive Rate
False Positive Rate
0.0
0.2
0.4
0.6
0.8
1.0
ROC curve of class 0 (area = 0.89)
ROC curve of class 1 (area = 0.77)
ROC curve of class 2 (area = 0.80)

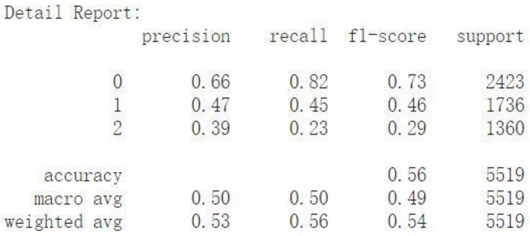
3.2.7 Word2Vec

- **Overall Evaluation**

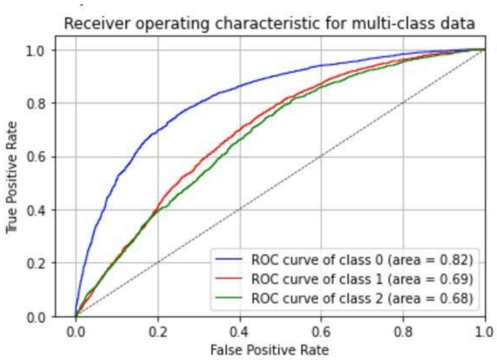
Classifiers	Accuracy	AUC	F1 Score	Precision	Recall	Training Time (s)
Naive Bayes	0.53	0.59	0.4179	0.4881	0.4472	0.37
Logistic Regression	0.54	0.64	0.4492	0.4928	0.4784	4.99
Decision Tree	0.47	0.59	0.4393	0.4392	0.4396	4.89
KNN	0.49	0.62	0.3645	0.4331	0.4039	494.21
XGBoost	**0.56**	**0.73**	**0.4942**	**0.5046**	**0.5026**	**28.61**
Random Forest	**0.57**	**0.73**	**0.4785**	**0.5152**	**0.5012**	**20.69**
MLP	0.44	0.62	0.2206	0.4317	0.3396	26.95

- **The best 2 classifiers**

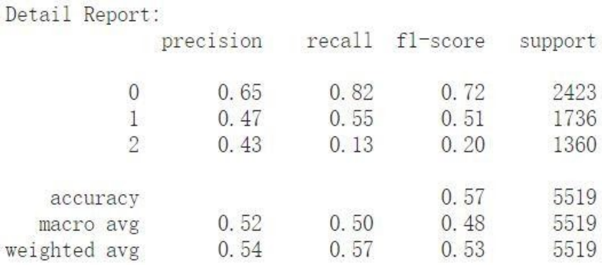
Xgboost:



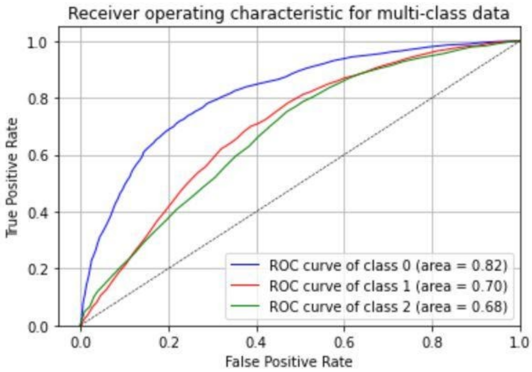
```
Detail Report:
              precision    recall  f1-score   support

           0       0.66      0.82      0.73      2423
           1       0.47      0.45      0.46      1736
           2       0.39      0.23      0.29      1360

    accuracy                           0.56      5519
   macro avg       0.50      0.50      0.49      5519
weighted avg       0.53      0.56      0.54      5519
```



Random Forest:



```
Detail Report:
              precision    recall  f1-score   support

           0       0.65      0.82      0.72      2423
           1       0.47      0.55      0.51      1736
           2       0.43      0.13      0.20      1360

    accuracy                           0.57      5519
   macro avg       0.52      0.50      0.48      5519
weighted avg       0.54      0.57      0.53      5519
```



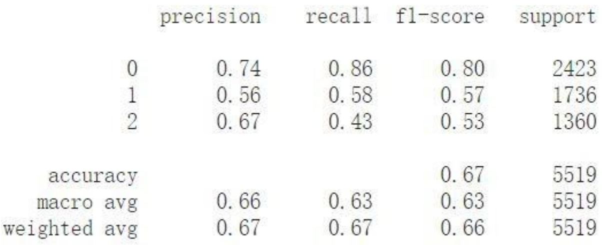
3.2.8 Word2Vec with other Features (Single Classifier)

- **Overall Evaluation**

Classifiers	Accuracy	AUC	F1 Score	Precision	Recall	Training Time (s)
Logistic Regression	0.60	0.72	0.5255	0.5630	0.5954	2.24
Decision Tree	0.56	0.66	0.5400	0.5377	0.5402	4.80
KNN	0.50	0.64	0.3800	0.4588	0.4166	270.39
XGBoost	**0.67**	**0.83**	**0.6318**	**0.6567**	**0.6271**	**20.00**
Random Forest	**0.63**	**0.8**	**0.5761**	**0.6112**	**0.5785**	**24.00**
MLP	0.58	0.72	0.4399	0.4774	0.5121	28.33

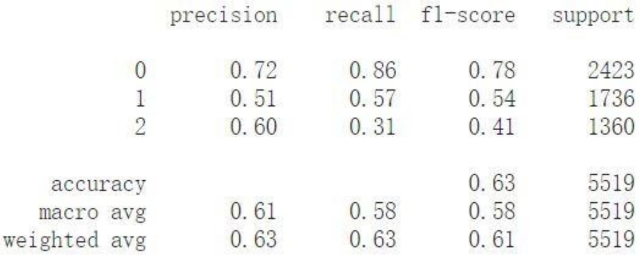
- **The best 2 classifiers**

XGBoost:

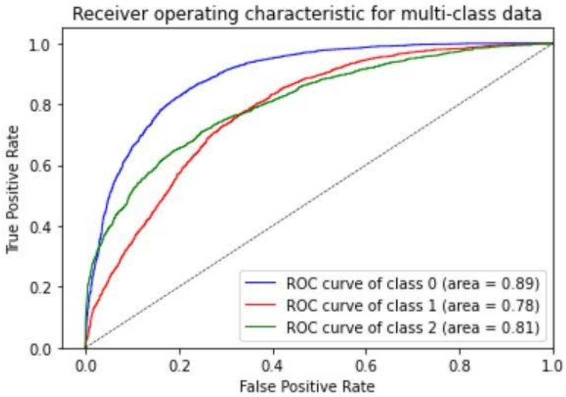


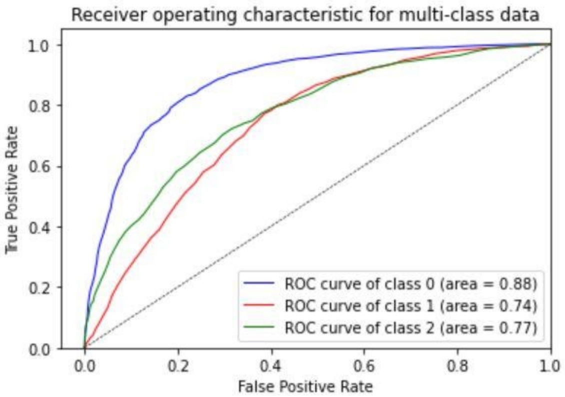
```
              precision    recall  f1-score   support

           0       0.74      0.86      0.80      2423
           1       0.56      0.58      0.57      1736
           2       0.67      0.43      0.53      1360

    accuracy                           0.67      5519
   macro avg       0.66      0.63      0.63      5519
weighted avg       0.67      0.67      0.66      5519
```

Random forest:



```
              precision    recall  f1-score   support

           0       0.72      0.86      0.78      2423
           1       0.51      0.57      0.54      1736
           2       0.60      0.31      0.41      1360

    accuracy                           0.63      5519
   macro avg       0.61      0.58      0.58      5519
weighted avg       0.63      0.63      0.61      5519
```





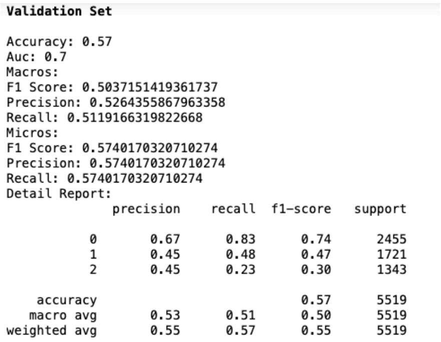
3.2.9 Word2Vec with other Features (2-layer blending)

- **Overall Evaluation**

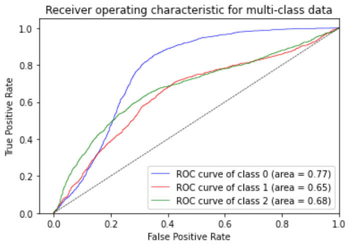
(For Word2vec & Other Features)	Accuracy	AUC	F1 Score macro	Precision	Recall	Training Time (s)
Logistic Regression & XGB	**0.57**	**0.70**	**0.51**	**0.53**	**0.51**	**20.1**
Random Forest Classifier & XGB	**0.57**	**0.69**	**0.51**	**0.52**	**0.51**	**1.6**
Logistic Regression & Random Forest Classifier	0.56	0.73	0.48	0.49	0.49	1.9
Gradient Boosting Classifier & XGB	0.56	0.71	0.49	0.5	0.5	61.99

- **The best 2 classifiers:**

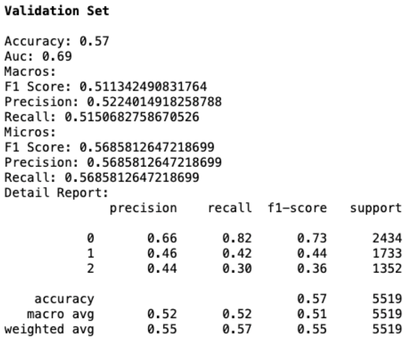
Logistic Regression & XGB:



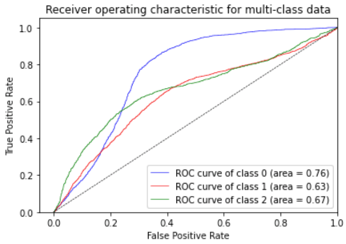
```
Validation Set

Accuracy: 0.57
Auc: 0.7
Macros:
F1 Score: 0.5037151419361737
Precision: 0.5264355867963358
Recall: 0.5119166319822668
Micros:
F1 Score: 0.5740170320710274
Precision: 0.5740170320710274
Recall: 0.5740170320710274
Detail Report:
              precision    recall  f1-score   support

           0       0.67      0.83      0.74      2455
           1       0.45      0.48      0.47      1721
           2       0.45      0.23      0.30      1343

    accuracy                           0.57      5519
   macro avg       0.53      0.51      0.50      5519
weighted avg       0.55      0.57      0.55      5519
```



Random Forest Classifier & XGB:



```
Validation Set

Accuracy: 0.57
Auc: 0.69
Macros:
F1 Score: 0.511342490831764
Precision: 0.5224014918258788
Recall: 0.5150682758670526
Micros:
F1 Score: 0.5685812647218699
Precision: 0.5685812647218699
Recall: 0.5685812647218699
Detail Report:
              precision    recall  f1-score   support

           0       0.66      0.82      0.73      2434
           1       0.46      0.42      0.44      1733
           2       0.44      0.30      0.36      1352

    accuracy                           0.57      5519
   macro avg       0.52      0.52      0.51      5519
weighted avg       0.55      0.57      0.55      5519
```



Part 4: Discussion

4.1 Summary of the model evaluation result

Models	Classifiers	Accuracy	AUC	F1 Score	Precision	Recall	Training Time (s)
Word2Vec with Other Features (Proposed)	**XGB**	**0.67**	**0.83**	**0.6318**	**0.6567**	**0.6271**	**20**
Random	-	0.35	-	0.33	0.33	0.33	0.11
Other features only	**XGB**	**0.68**	**0.83**	**0.65**	**0.69**	**0.64**	**4.76**
TF-IDF-only	Logistic Regression	0.62	0.78	0.5706	0.5801	0.5706	9.95
Word2Vec only	XGB	0.56	0.73	0.4942	0.5046	0.5026	28.61
FastText only	**-**	**-**	**-**	**-**	**0.69**	**-**	**1.28**
TF-IDF with Other Features	**XGB**	**0.7**	**0.85**	**0.6612**	**0.69**	**0.6562**	**94.36**
TF-IDF with Other Features (blending)	XGB & Random Forest	0.65	0.81	0.61	0.63	0.61	131.61
Word2Vec with Other Features (blending)	Logistic Regression & XGB	0.58	0.73	0.54	0.54	0.54	20.1

4.1 Summary of model evaluation results

In this paper, we built several models to predict the usefulness of a Yelp customer review. Among all the classifiers, the most powerful and effective one is the XGBoost, which performs well in each model. When using accuracy as the evaluation metric, the best model is the **XGBoost model utilizing TF-IDF(lexicon) features with all other features**, reaching a test accuracy of 70% on the test dataset, which is very effective.

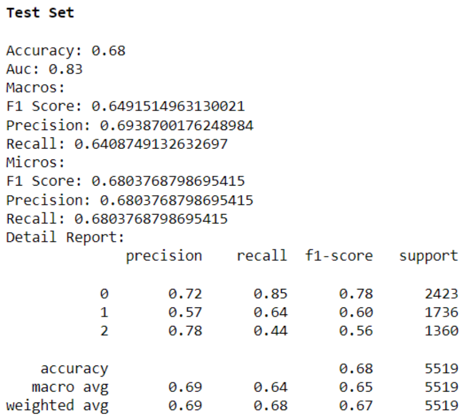
In our experiment, the model using other features (structural, sentiment analysis, and foreign key features) only, TF-IDF-only, and Word2Vec only, all give relatively good results. In contrast, other features (structural, sentiment analysis, and foreign key features) seemed to be more informative for predicting review usefulness than lexicon features (Indices generated by TF-IDF, Word2Vec). This indicates that the combined feature matrix using readability, polarity, subjectivity, etc., has a more significant effect on review helpfulness than term frequency. This reveals that easy-to-understand text can improve review usefulness. In *Estimating the helpfulness and economic impact of product reviews: Mining text and reviewer characteristics*[1], it is found that the readability of reviews has a positive effect on perceived helpfulness, and spelling errors hurt perceived helpfulness. Besides, the polarity of emotions that a review also contains critically affects review usefulness because reviews

[1] Ghose, A.; Ipeirotis, P.G. Estimating the helpfulness and economic impact of product reviews: Mining text and reviewer characteristics. IEEE Trans. Knowl. Data Eng. 2011, 23.

with more positive/negative emotions embedded tend to have more information than unbiased reviews. A factor that could be easily neglected is that the reviewer's reputation could also affect the usefulness of reviews. The reviewer's level of credibility directly affects the review's usefulness. Yelp Elites' reviews are more likely to be recommended by the system and are more helpful for potential consumers to make decisions. Generally speaking, the more complete a user's personal information is (e.g., those users who have IDs and photos), the higher credibility the reviews they leave are[2]. Comparatively, bag-of-word and term frequency appear less significant, indicating that "important words" are relatively less helpful in contributing review usefulness.

Interestingly, TF-IDF combining other features performs much better than word2vec combining other features, with the latter one even performing worse than using other features only in prediction. However, although TF-IDF combining other features gives the best result overall, this method is considerably time-consuming, requiring robust hashing power of target CPUs. Therefore, considering time cost, **the XGBoost model utilizing all other features only** is also considered a good model, reaching an accuracy of 0.68 but essentially reducing training time.

Also, it is worth mentioning that if using precision as our metric, **the XGBoost model utilizes TF-IDF(lexicon) features with all other features, the XGBoost model utilizes all other features only,** and **FastText** gives equally good results of 0.69. Using grid search cross validation to tune the parameters of XGBoost model, we obtained the optimal parameters are: "learning_rate": 0.1, "max_dept": 8, and "n_estimators": 60.



```
Test Set

Accuracy: 0.68
Auc: 0.83
Macros:
F1 Score: 0.6491514963130021
Precision: 0.6938700176248984
Recall: 0.6408749132632697
Micros:
F1 Score: 0.6803768798695415
Precision: 0.6803768798695415
Recall: 0.6803768798695415
Detail Report:
              precision    recall  f1-score   support

           0       0.72      0.85      0.78      2423
           1       0.57      0.64      0.60      1736
           2       0.78      0.44      0.56      1360

    accuracy                           0.68      5519
   macro avg       0.69      0.64      0.65      5519
weighted avg       0.69      0.68      0.67      5519
```

To our surprise, the FastText method only used a training time of 1.28s, which is impressive. FastText method is a neural network-based open-source library that allows users to learn text representations and text classifiers. The FastText classifier takes in the text without any features input, which is very convenient to implement.

`

[2] Mudambi, S.M.; Schuff, D. What makes a helpful online review? A study of customer review on Amazon.com. MIS Q. 2010, 34.

Part 5: Conclusion, future work and lessons learned

5.1 Conclusions

In this project, we attempt to predict whether a Yelp customer's review is helpful or not. We used The Yelp dataset (https://www.yelp.com/dataset) published by Yelp and generated four significant features from it—lexical features, structural features, sentiment features, and foreign features key features with the last three categories combined as "other features." We conducted eight major experiments by combining these features and using classifiers, including Logistic Regression, XGBoost, GaussianNB, and Gradient Boosting Classifier. We examined their effectiveness by comparing the results using Accuracy, F1 Score, Precision, Recall, AUC, and Training Time. Among all the models, **the XGBoost model utilizing TF-IDF(lexicon) features with all other features** reaches the highest test accuracy of 70% but takes up the longest time. If we use precision as our metric, **the XGBoost model utilizes TF-IDF(lexicon) features with all other features. The XGBoost model utilizes all other features only,** and **FastText** gives equally good results of 0.69. To our surprise, the FastText method only used a training time of 1.28s, which is impressive.

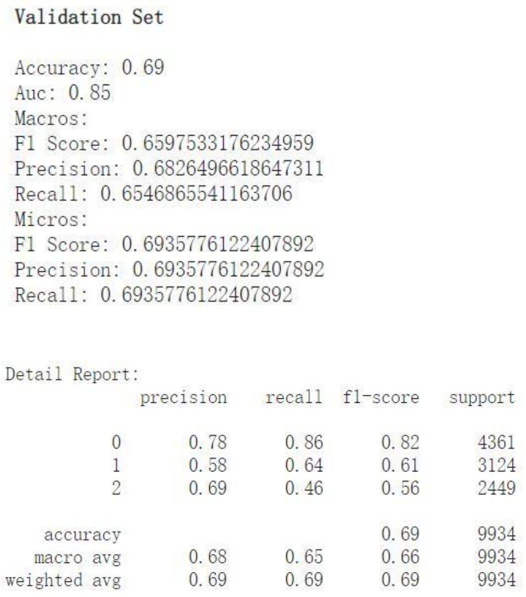
5.2 Future work and lessons learned

There are also many limitations to this project. Firstly, the data we used for this project is the "global" data, which is too broad. Future work can be done in applying our methods to different product categories and even mixed categories. For example, we can consider incorporating area factors (different cities, other parts of the country, etc.), restaurant type (fine dining, street food, etc.), and the business scale into our prediction model. Moreover, restaurants that have been operating for a longer time tend to have more reviews. Therefore, the standard of choosing "useful reviews" should be changed accordingly. Besides, our sampling method to balance the useful, moderately helpful and unhelpful reviews is effective in the experiment, but not in reality. As a result, applying the same approach in a real-world dataset might get very different results.

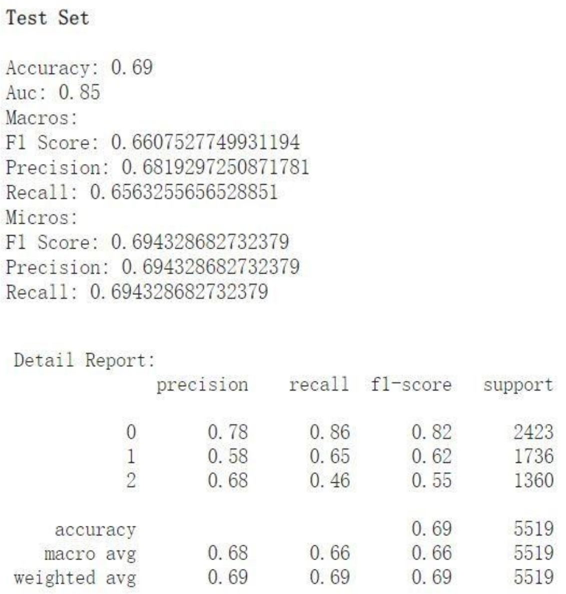
Our work can be implemented as a system used to help businesses improve their ranking in Yelp and support customers in their consumption decision-making process. Besides, this system can also encourage customers to write high-quality reviews continuously.

Appendix

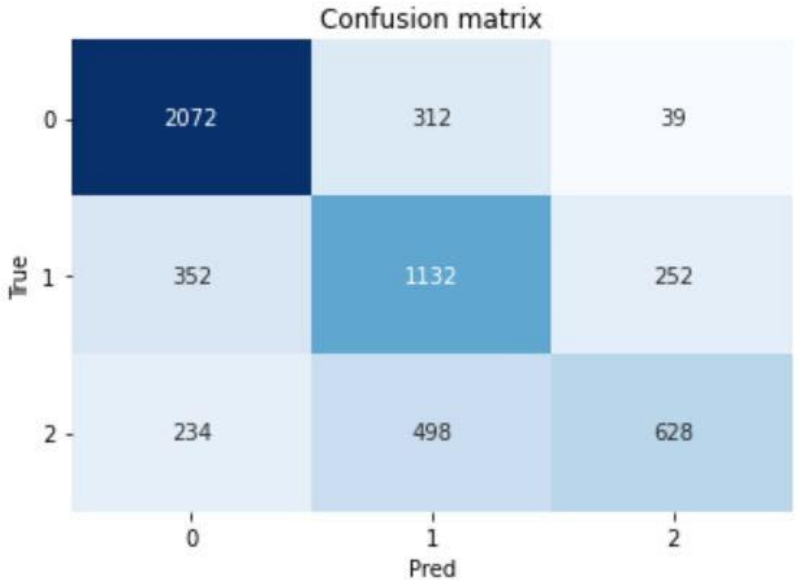
Appendix 1: Fitting outcomes of TF-IDF with Other Features using XGBoost

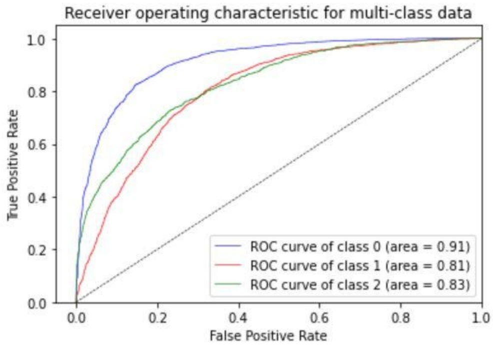


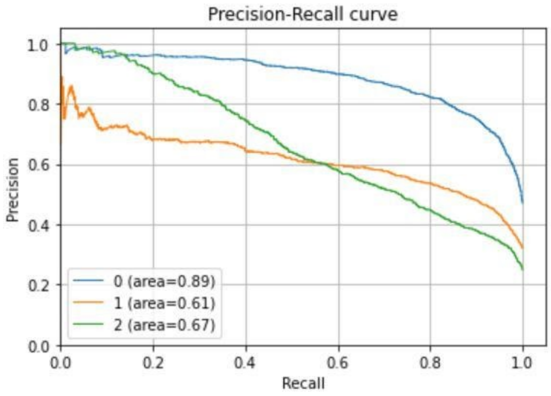
```
Validation Set

Accuracy: 0.69
Auc: 0.85
Macros:
F1 Score: 0.6597533176234959
Precision: 0.6826496618647311
Recall: 0.6546865541163706
Micros:
F1 Score: 0.6935776122407892
Precision: 0.6935776122407892
Recall: 0.6935776122407892


Detail Report:
              precision    recall  f1-score   support

           0       0.78      0.86      0.82      4361
           1       0.58      0.64      0.61      3124
           2       0.69      0.46      0.56      2449

    accuracy                           0.69      9934
   macro avg       0.68      0.65      0.66      9934
weighted avg       0.69      0.69      0.69      9934
```



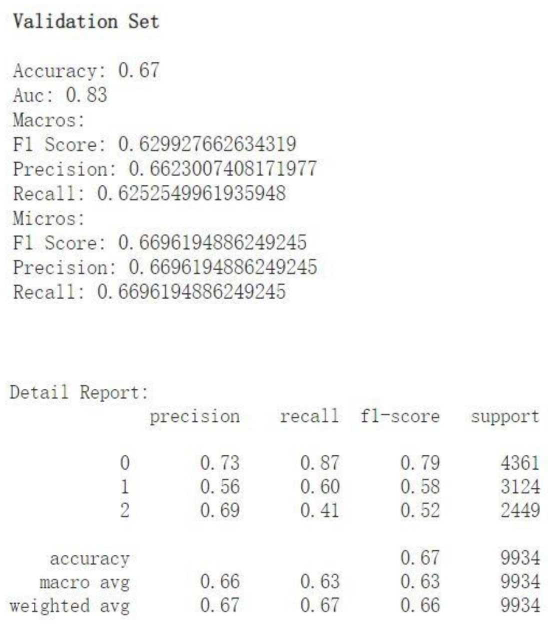
```
Test Set

Accuracy: 0.69
Auc: 0.85
Macros:
F1 Score: 0.6607527749931194
Precision: 0.6819297250871781
Recall: 0.6563255656528851
Micros:
F1 Score: 0.694328682732379
Precision: 0.694328682732379
Recall: 0.694328682732379


Detail Report:
              precision    recall  f1-score   support

           0       0.78      0.86      0.82      2423
           1       0.58      0.65      0.62      1736
           2       0.68      0.46      0.55      1360

    accuracy                           0.69      5519
   macro avg       0.68      0.66      0.66      5519
weighted avg       0.69      0.69      0.69      5519
```



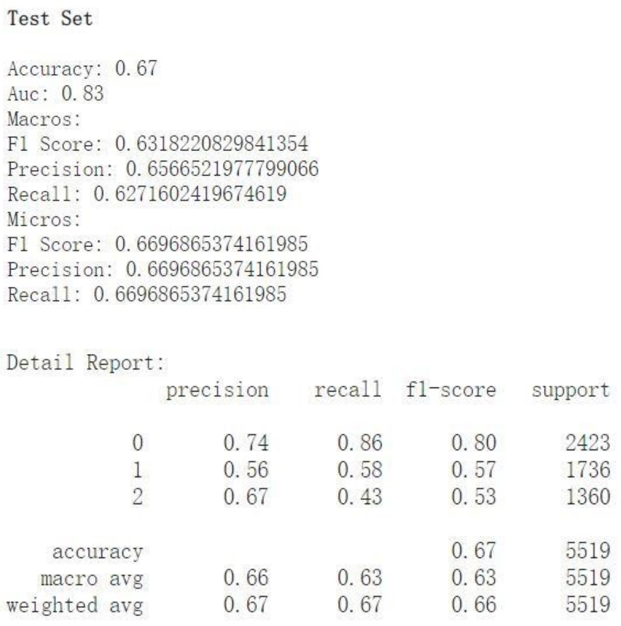




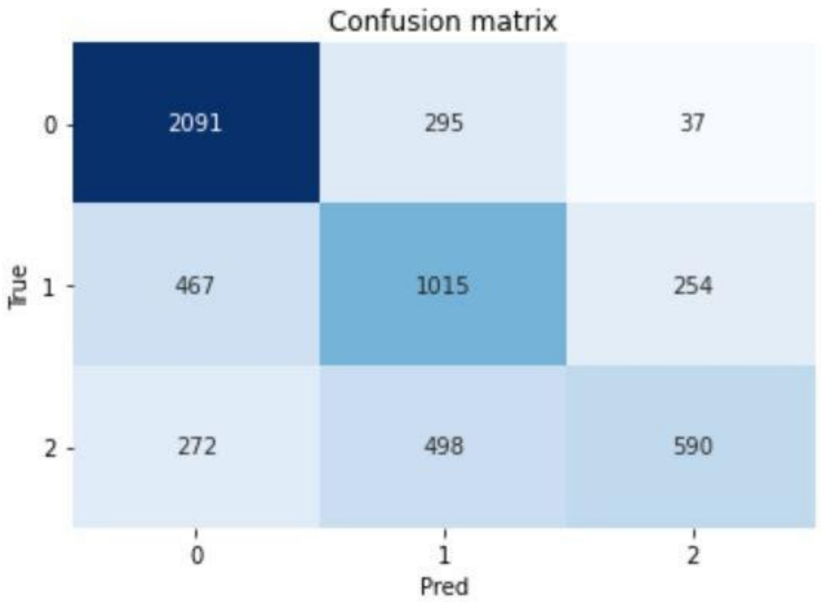
Appendix 2: Fitting outcomes of Word2vec with Other Features using XGBoost

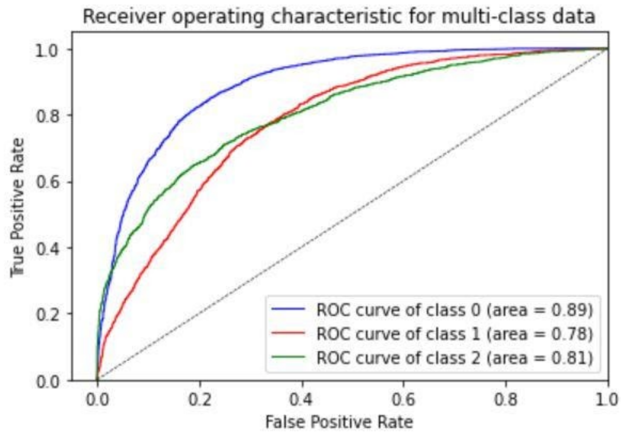


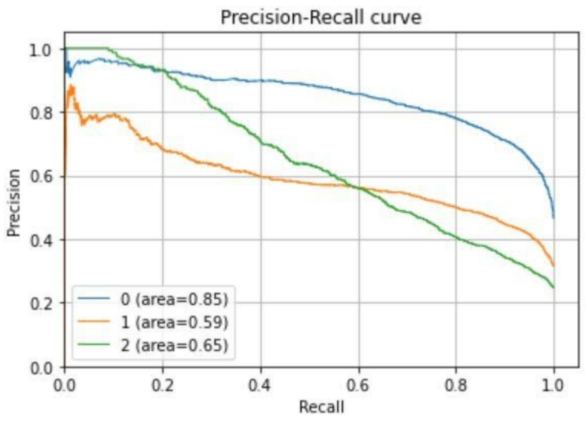
```
Validation Set

Accuracy: 0.67
Auc: 0.83
Macros:
F1 Score: 0.629927662634319
Precision: 0.6623007408171977
Recall: 0.6252549961935948
Micros:
F1 Score: 0.6696194886249245
Precision: 0.6696194886249245
Recall: 0.6696194886249245


Detail Report:
              precision    recall  f1-score   support

           0       0.73      0.87      0.79      4361
           1       0.56      0.60      0.58      3124
           2       0.69      0.41      0.52      2449

    accuracy                           0.67      9934
   macro avg       0.66      0.63      0.63      9934
weighted avg       0.67      0.67      0.66      9934
```



```
Test Set

Accuracy: 0.67
Auc: 0.83
Macros:
F1 Score: 0.6318220829841354
Precision: 0.6566521977799066
Recall: 0.6271602419674619
Micros:
F1 Score: 0.6696865374161985
Precision: 0.6696865374161985
Recall: 0.6696865374161985

Detail Report:
              precision    recall  f1-score   support

           0       0.74      0.86      0.80      2423
           1       0.56      0.58      0.57      1736
           2       0.67      0.43      0.53      1360

    accuracy                           0.67      5519
   macro avg       0.66      0.63      0.63      5519
weighted avg       0.67      0.67      0.66      5519
```


Confusion matrix
True
Pred


Receiver operating characteristic for multi-class data
True Positive Rate
False Positive Rate
ROC curve of class 0 (area = 0.89)
ROC curve of class 1 (area = 0.78)
ROC curve of class 2 (area = 0.81)


Precision-Recall curve
Precision
Recall
0 (area=0.85)
1 (area=0.59)
2 (area=0.65)