SEAS6402 Project Final Report

Yelp Restaurant Future Success Prediction

Ran Wei[1], Rui Zhang[2], Bite Xiong[3]

1. Problem Statement

Many new restaurants do not survive their first few years in business (Parsa et al., 2005) [1]. Various factors impact the success of restaurants. Finding those critical factors can help investors and business owners understand the critical factors contributing to a restaurant's success.

This project aims to extract restaurant characteristics to predict whether a restaurant will still be open after two years. Specifically, we use a restaurant's text and non-text features from the 2018 Yelp dataset [2] to predict if the restaurant is still open until 2020. Unlike previous successful prediction studies, we will make text features more valuable by using an advanced NLP method. Various predictive models will be used in this case study, and we will compare the performance and find the best one.

2. Related Work

Prior work using the Yelp dataset paid more attention to customers' reviews. Dai et al. (2012) [3] offered a structural approach to construct an average rating of a given set of reviews. Applying this approach to customers' review stars, they constructed optimal ratings for all restaurants and compared them to the rating stars displayed by Yelp. Their study only focused on numerical data. Productive literature has begun to use text analysis to extract informational content from text reviews.

Huang et al. (2014) [4] used the Latent Dirichlet Allocation algorithm to exploit the hidden subtopic in text corpora. In the research, the breakdown of topics overall reviews in the Yelp dataset showed that "service" is the keyword that customers care about most and extended their findings to that of temporal information regarding restaurants' peak hours. Their work explored more insights into text reviews. As the Yelp dataset is an information-rich dataset, some non-text features unrelated to reviews should be considered in research.

Chen and Xia (2020) [5] predicted restaurants' ratings for 2019 by using both text features and non-text features of 2018. For text features, unigram features were used to distinguish the attitude of users; bigram features involved customers' opinions on the environment, location, service, and taste. Non-text features are generated from the business attributes of each restaurant and divided into three groups.

The Yelp dataset has also become the main resource for predicting the success and failure of restaurants. Lu et al. (2018) [6] generated their model with both text and non-text features and analyzed features that influence the most for the future success of the restaurant. The result showed that their text features failed to have significant indications for the future success of the restaurant, while non-text features had a strong correlation with future restaurant performance. Since Luca (2016) [7] found a one-star increase in Yelp

[1] Ran Wei, GWID: G20123964, email: ran_wei@gwu.edu

[2] Rui Zhang, GWID: G38902280, email: ruizhang@gwu.edu

[3] Bite Xiong, GWID: G32496599, email: peter75977@gwu.edu

review ratings leads to a 5-9 percent increase in revenue. We think text features should have a stronger impact.

Lu et al. (2018) [6] generated a domain-specific keywords dictionary and counted word occurrences as features. Based on the idea, they designed two unigram features and eight bigram features. But all the features reflect sentiment rather than semantic information. Ajrwi et al. (2021) [8] performed sentiment analysis using combinations of different features like unigrams, bigrams, and TF-IDF. Those features were trained on Supervised Learning algorithms, and the authors found the combination of unigrams, bigrams, and TF-IDF has the best performance. Using unigrams and bigrams usually requires choosing a large number of n-grams which means high-dimensional features. Since we only have limited rows in our dataset, too many features may cause overfitting easily. In our project, we try to represent text reviews in low-dimensional space and can reflect all information of the reviews.

3. Data

In this section, we provide a general description of the dataset and the processing of data formation and feature extraction.

3.1 Dataset Description

The dataset we used for this project is the Yelp dataset. We obtained two Yelp datasets from Kaggle.com, and they have the same format and matched business entities but different release dates. One was released in December 2018, and the other one was released in February 2020. Each dataset includes over 160 thousand businesses, 7 million reviews, and 200 thousand users. The size of the data table is more than 10 gigabytes. The raw data has 100 million rows and more than 50 columns. The dataset contains five JSON tables. Our project mainly uses three tables: “business”, “review”, and “checkin”. Table 1 shows the attributes of each table we used in this project.

Table 1. Attributes of Each Table

Table_Name	Attributes
business	business_id, name, state, stars, review_count, attributes, categories, is_open
review	business_id, user_id, stars, text, date
checkin	business_id,date

“business_id” in both of the three tables is the primary key. We used it to match data and merge tables. “is_open” in the business table is the target variable. If “is_open” equals 1, it means the restaurant is open; if “is_open” equals 0, it means the restaurant is closed.

3.2 Dataset Formation

The Yelp dataset contains several business types, not only restaurants. We first selected all the businesses in 2018 which "category" attribute contained "Restaurant" or "restaurant", then filtered out closed restaurants by using the "is_open" attribute. We got 41,718 open restaurants in 2018.

We matched with restaurants in 2020 by "business_id." There were 27,619 restaurants still open in 2020 and 2,063 closed restaurants. Besides, we found 12,036 restaurants missing the target label, which means we do not know whether those restaurants are still open or closed.

To impute the missing label. We found the date of the last review in all the closed restaurants in 2020. If a restaurant had a review after this date, we identified the restaurant as open and labeled 1; otherwise, we discard these missing-label records. Applying this method, we got two open restaurants and discard 12,034 missing-label restaurants. Figure 1 shows the dataset formation process.

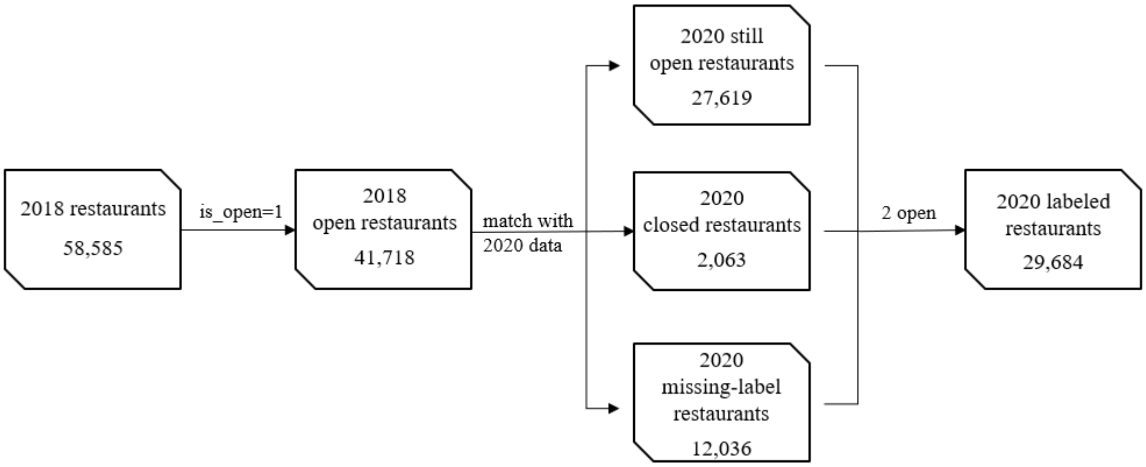


Figure 1. Dataset Formation Process

3.3 Feature Extraction

Our features for this project include non-text features and text features, as shown in Table 2.

Table 2. Features Categorizations

	Name	**Number of Columns**
Non-text Features	Review Star	1
	Review Count	1
	Checkin Count	1
	Chain Restaurant	1
	Return Guest Count	1
	Restaurant Category	10
	Location	1
	Other Attributes	27
Text Features	Reviews	384
Total		**427**

Non-text Features

Some non-text features could be directly obtained from the data.

- Review Star – The total rating stars of a restaurant provided by Yelp.com range from 1 to 5. We obtained this feature from the “stars” attribute in the “business” table.
- Review Count – The number of reviews a restaurant has. We obtained this feature from the “review_count” attribute in the “business” table.

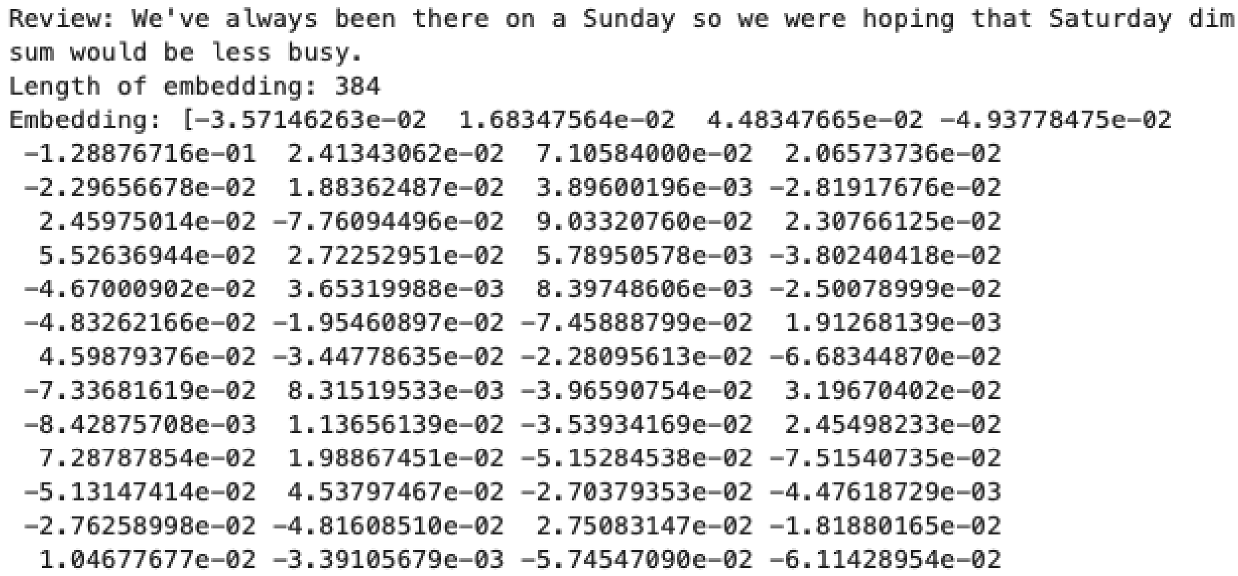
Other non-text features needed processing to parse them.

- Checkin Count – We believed more checkin might indicate the restaurant is popular. However, there was no such attribute in the data. In the "checkin" table, there was a "date" attribute that stored all the checkin dates for each restaurant. We counted the number of checkin dates as checkin count.
- Chain Restaurant – Chain restaurants might have a more stable operation style and are more unlikely to be closed. We identified the restaurant's name appeared more than twice in the "business" table as a chain restaurant and marked it as 1; others were marked as 0.
- Return Guest Count – A higher number of returned customers means stronger user loyalty. We used the "user_id" attribute in the "review" table to identify the return guest. If an "user_id" had two or more reviews for a particular restaurant, we identified the user as a return guest. Then, counted the number of returned guests for each restaurant.
- Restaurant Category – The restaurant category might also affect the success of a restaurant. We extracted categories from the "*category*" attribute in the "business" table and chose the top 10 restaurant categories as 10 category features: "cat_Pizza," "cat_Chines," "cat_Mexican," "cat_Italian," "cat_FastFood," "cat_American (Traditional)," "cat_Thai," "cat_Vienamese," "cat_Indian," and "cat_Breakfast & Brunch." If a restaurant had the features' keyword in its "category" attribute, the corresponding feature value would become 1. Otherwise, it would become 0.
- Location – The economic status of a certain city might affect the business situation for all restaurants in it. We believed that compared with location, the income data of each state is more important to our predictive model, and it can directly reflect the household consumption level. So, we collected the median annual household income in 2018 for each state (income data source: US [9] and Canada [10]) and normalized it as a feature.
- Other Attributes – Some useful features were nested in dictionary format in the 'attributes' field of the "business" table. We applied for loop to extract the keys from every dictionary, and choose the top 15 attributes with higher frequency as our features. Then extract the value for each record based on these 15 features. Not every restaurant contains all 15 features, so we marked the record with each feature as 1 if a restaurant has this attribute, and the rest were marked as 0. Some features are nested in these 15 features and can be extracted as individual features. So, we got 27 attribute features in total, which are listed below:
    1. RestaurantsTakeOut
    2. RestaurantsPriceRange2
    3. RestaurantsReservations
    4. RestaurantsGoodForGroups
    5. RestaurantsDelivery
    6. GoodForKids
    7. OutdoorSeating
    8. RestaurantsAttire
    9. HasTV
    10. BikeParking
    11. Alcohol
    12. WiFi
    13. NoiseLevel
    14. BusinessParking_garage

15. BusinessParking_street
16. BusinessParking_validated
17. BusinessParking_lot
18. BusinessParking_valet
19. Ambience_romantic
20. Ambience_intimate
21. Ambience_classy
22. Ambience_touristy
23. Ambience_trendy
24. Ambience_casual
25. Ambience_upscale
26. Ambience_hipster
27. Ambience_divey

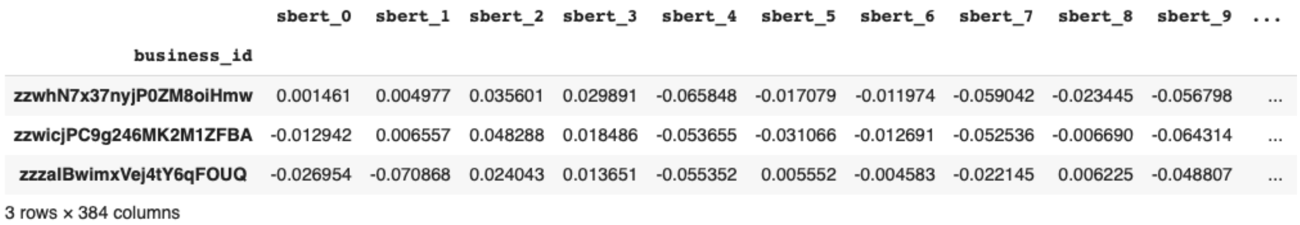
Text Features

We had 3.55 million reviews in dataset v1 and 2.51 million reviews in dataset v2. We embedded the text into dense and low-dimensional space by using Sentence-BERT (SBERT). SBERT can effectively create sentence embeddings that capture semantic information. Each review was converted to a numerical vector of length 384. Figure 2 showed an example of SBERT output. For each restaurant, we averaged all review vectors. So, each restaurant would have 384 text features, as shown in Figure 3. More detail about SBERT will be discussed in the next section.



```
Review: We've always been there on a Sunday so we were hoping that Saturday dim
sum would be less busy.
Length of embedding: 384
Embedding: [-3.57146263e-02  1.68347564e-02  4.48347665e-02 -4.93778475e-02
 -1.28876716e-01  2.41343062e-02  7.10584000e-02  2.06573736e-02
 -2.29656678e-02  1.88362487e-02  3.89600196e-03 -2.81917676e-02
  2.45975014e-02 -7.76094496e-02  9.03320760e-02  2.30766125e-02
  5.52636944e-02  2.72252951e-02  5.78950578e-03 -3.80240418e-02
 -4.67000902e-02  3.65319988e-03  8.39748606e-03 -2.50078999e-02
 -4.83262166e-02 -1.95460897e-02 -7.45888799e-02  1.91268139e-03
  4.59879376e-02 -3.44778635e-02 -2.28095613e-02 -6.68344870e-02
 -7.33681619e-02  8.31519533e-03 -3.96590754e-02  3.19670402e-02
 -8.42875708e-03  1.13656139e-02 -3.53934169e-02  2.45498233e-02
  7.28787854e-02  1.98867451e-02 -5.15284538e-02 -7.51540735e-02
 -5.13147414e-02  4.53797467e-02 -2.70379353e-02 -4.47618729e-03
 -2.76258998e-02 -4.81608510e-02  2.75083147e-02 -1.81880165e-02
  1.04677677e-02 -3.39105679e-03 -5.74547090e-02 -6.11428954e-02
```

Figure 2. An Example of SBERT Output (Part of the result)



	sbert_0	sbert_1	sbert_2	sbert_3	sbert_4	sbert_5	sbert_6	sbert_7	sbert_8	sbert_9	...
business_id											
zzwhN7x37nyjP0ZM8oiHmw	0.001461	0.004977	0.035601	0.029891	-0.065848	-0.017079	-0.011974	-0.059042	-0.023445	-0.056798	...
zzwicjPC9g246MK2M1ZFBA	-0.012942	0.006557	0.048288	0.018486	-0.053655	-0.031066	-0.012691	-0.052536	-0.006690	-0.064314	...
zzzalBwimxVej4tY6qFOUQ	-0.026954	-0.070868	0.024043	0.013651	-0.055352	0.005552	-0.004583	-0.022145	0.006225	-0.048807	...

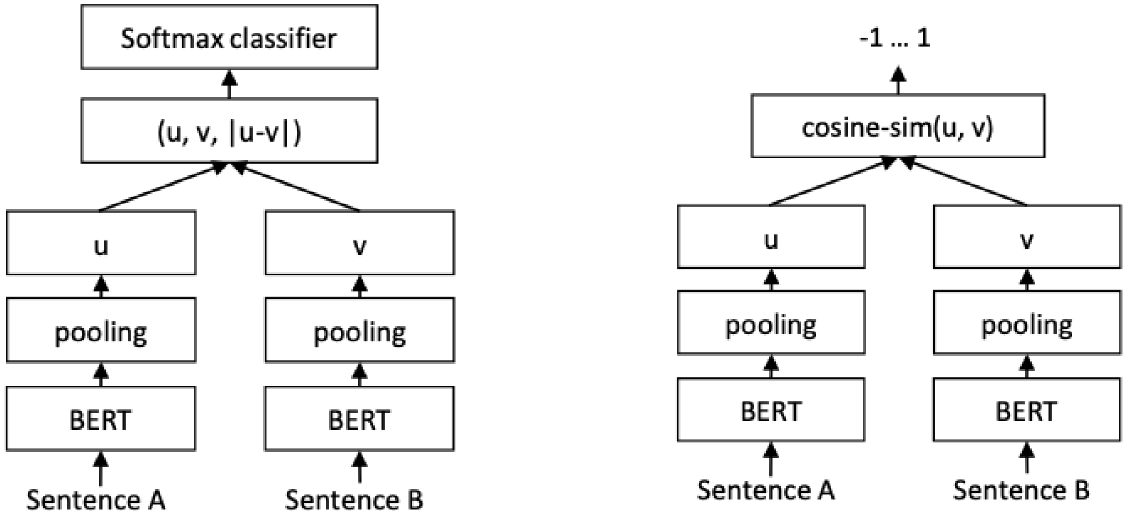
3 rows × 384 columns

Figure 3. Example of Text Feature (Part of the result)

4. Methods

Sentence-BERT (SBERT)

The SBERT is a modification of the pre-trained BERT network presented by Reimers and Gurevych (2019) [11]. BERT is a very powerful language model. But it does not have a method to represent a sentence. SBERT provides an easy way to produce text embedding. Reimers and Gurevych (2019) added a pooling operation to the output of BERT to derive a fixed-sized sentence embedding and created siamese and triplet networks to fine-tune BERT. Figure 4 shows the SBERT architecture.



(a) SBERT Architecture with Classification Objective Function (Siamese network structure)

(b) SBERT Architecture at Inference (Regression objective function and triplet objective function)

Figure 4. SBERT Architecture (Reimers and Gurevych, 2019)

To implement SBERT, we used the python library: sentence-transformers [11] and did the computation on Google Colab Pro GPU.

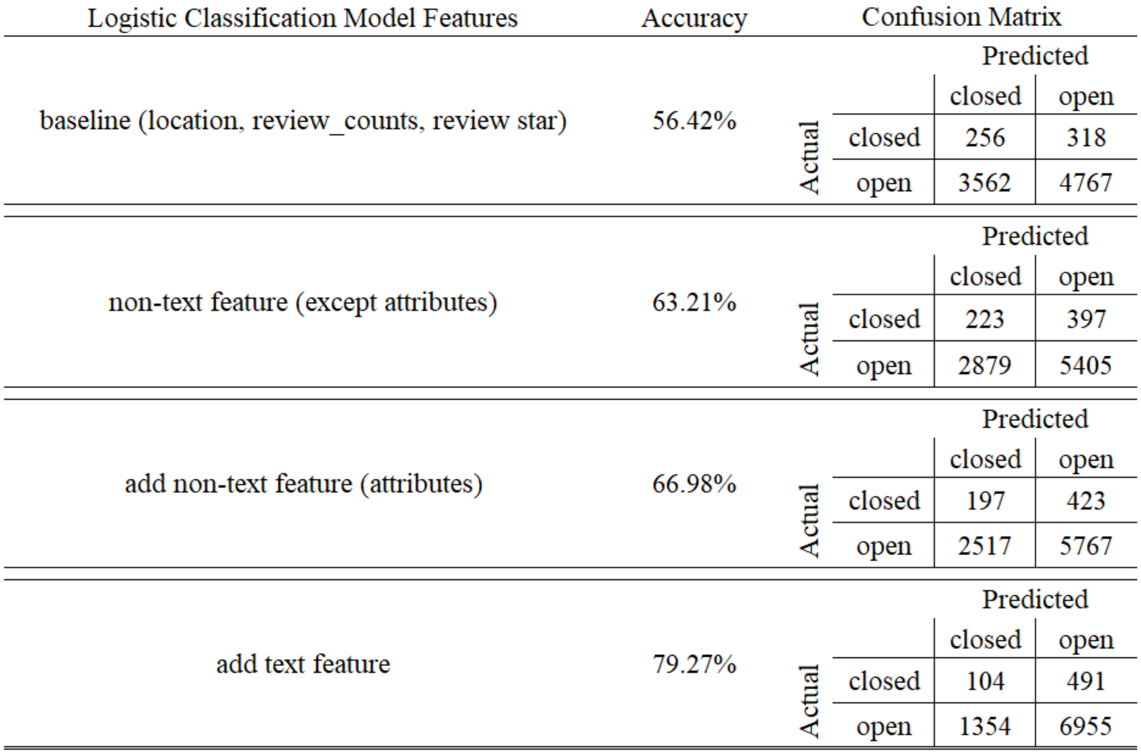
Logistic Regression

For the baseline modeling, we applied binary logistic regression through the python sklearn.linear_model [13] library on all of the non-text features except other attributes. However, since the proportion of each target class (open or closed) was imbalanced, class imbalance arises due to the fact that the model is trained predominantly on the label of the majority class and very little on the minority class. To be specific, since the proportion of the majority class (open) in our dataset was 93% in dataset v2 (with fewer closed labels) and 67% (with more closed labels) in dataset v1, the model would predict everything to be of the majority class, and we will end up with an accuracy of 93% and 67%. But this result is meaningless and useless. So, we applied the Synthetic Minority Over-sampling Technique (the SMOTE). This approach is described by Nitesh Chawla, et al. (2002) [14]. Through SMOTE, which is in imblearn.over_sampling library [15], we created synthetic observations using K-Means in the minority class, although these examples don't add any new information to the model. Instead, new examples can be synthesized from the existing examples.

5. Result

We applied Logistic Classification as our basic model. Starting with three features, which could be obtained directly from the dataset, the accuracy results improved generally by adding more features into the model, as shown in Table 3. The accuracy of Logistic Classification reached 79.27% when all of the features, including non-text or text features, were taken into consideration.

Table 3. Basic Model Results



Logistic Classification Model Features	Accuracy	Confusion Matrix			
				Predicted	
				closed	open
baseline (location, review_counts, review star)	56.42%	Actual	closed	256	318
			open	3562	4767
				Predicted	
				closed	open
non-text feature (except attributes)	63.21%	Actual	closed	223	397
			open	2879	5405
				Predicted	
				closed	open
add non-text feature (attributes)	66.98%	Actual	closed	197	423
			open	2517	5767
				Predicted	
				closed	open
add text feature	79.27%	Actual	closed	104	491
			open	1354	6955

We also tried several advanced models, like Decision Tree, Random Forest Classification, and XGBoost classification, as shown in Table 4. Using 0.5 as threshold, the decision tree model yielded the best performance with 83.96% accuracy. The top 25 factors, which are more important to predict if a restaurant will still be open after the next two years, in the Decision Tree model are shown in Figure 5. The most crucial factor is whether the ambiance of a restaurant is casual or not, which has an importance of about 0.28. Whether the restaurant provides delivery service is also an important factor with importance about 0.1.

Table 4. All Model Results

Model	Accuracy	Confusion Matrix			
				Predicted	
				closed	open
Baseline	56.42%	Actual	closed	256	318
			open	3562	4767
				Predicted	
				closed	open
Logistic Classification	79.27%	Actual	closed	223	397
			open	2879	5405
				Predicted	
				closed	open
Decision Tree	**83.96%**	Actual	closed	73	547
			open	881	7403
				Predicted	
				closed	open
Random Forest Classification	93.03%	Actual	closed	0	620
			open	1	8283
				Predicted	
				closed	open
XGBoost Classification	93.04%	Actual	closed	0	620
			open	0	8284

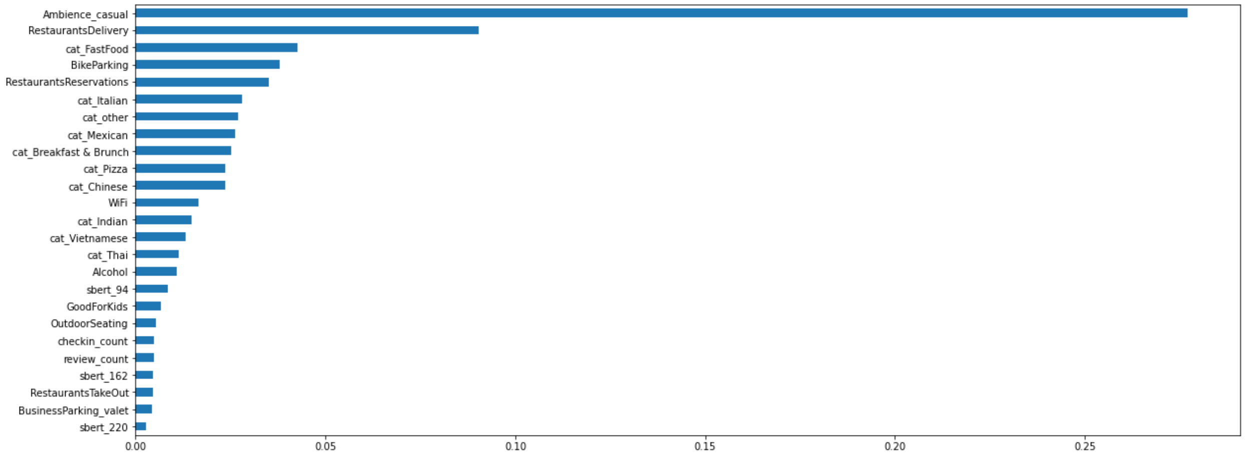


Figure 5. Decision Tree Feature Importance (Top 25)

In addition to the performance at threshold=0.5, we can evaluate the models considering all possible thresholds. Figure 6 shows the Precision-Recall curves and the AUC values. The findings are consistent with our previous results-the decision tree model has the best prediction power. The reason could be that the complicated models tend to overfit the training data.

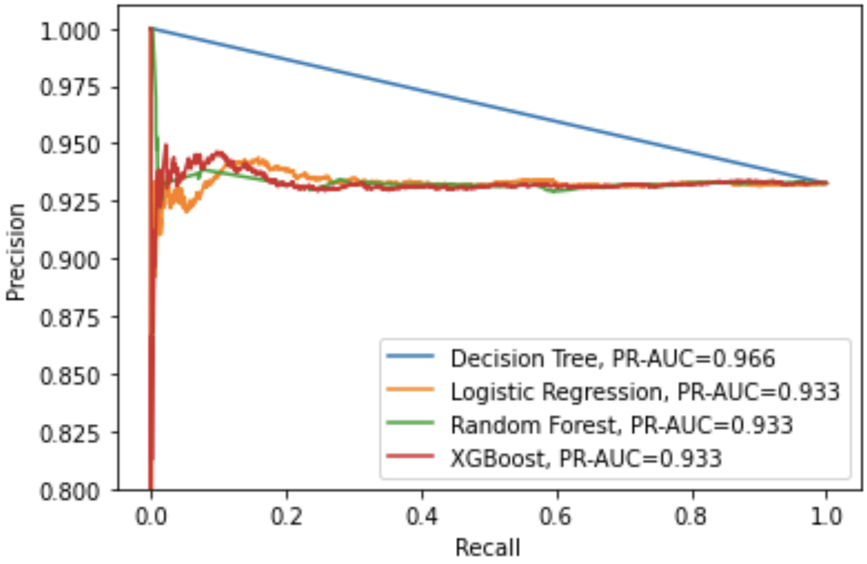


Figure 6. Precision-Recall Curves

6. Future Work

It's worth noticing that although the accuracy results of the Random Forest and XGBoost model are higher than the Decision Tree, the confusion matrix result looks meaningless with no predictive ability. In fact, this accuracy rate is the same as the ratio of positive to negative labels (93%) in our dataset. This means that the current way to balanced data seems didn't work well in these two models. We will search for more ways to balance data, like Outlier Detection or One Class Learning.

In addition, we also want to complete more analysis and add the pandemic factor if new data is available on Yelp's official website. A predictive calculator will be developed as well to help investors make the decision. By just inputting several key attributes of the restaurant, in which they want to invest, the calculator can provide how likely the restaurant will be open after the following two years.

References

[1] Parsa, H. G., et al. "Why Restaurants Fail." *Cornell Hotel and Restaurant Administration Quarterly*, vol. 46, no. 3, 2005, pp. 304-322.

[2] Yelp Dataset. Retrieved from https://www.kaggle.com/datasets/yelp-dataset/yelp-dataset (2018 & 2020)

[3] Dai, Weijia, et al. "Optimal aggregation of consumer ratings: an application to yelp. com." NBER Working Paper Series: 18567, 2012.

[4] Huang, James, Stephanie Rogers, and Eunkwang Joo. "Improving restaurants by extracting subtopics from yelp reviews." *iConference 2014 (Social Media Expo)*, 2014.

[5] Chen, Yifan and Xia, Fanzeng. "Restaurants' Rating Prediction Using Yelp Dataset." *2020 IEEE International Conference on Advances in Electrical Engineering and Computer Applications (AEECA)*. IEEE, 2020.

[6] Lu, Xiaopeng, et al. "Should I Invest it? Predicting Future Success of Yelp Restaurants." *Proceedings of the Practice and Experience on Advanced Research Computing*. 2018, pp. 1-6.

[7] Luca, Michael. "Reviews, reputation, and revenue: The case of Yelp. com." *Com (March 15, 2016). Harvard Business School NOM Unit Working Paper*, 2016, pp. 12-16.

[8] Al Ajrawi, Shams, et al. "Evaluating business yelp's star ratings using sentiment analysis." *Materials Today: Proceedings*, 2021.

[9] US median annual household income in 2018: https://nces.ed.gov/programs/digest/d19/tables/dt19_102.30.asp

[10] Canada median annual household income in 2018: https://www150.statcan.gc.ca/t1/tbl1/en/cv!recreate.action?pid=1110019001&selectedNodeIds=1D2,1D7,1D8,1D12,1D13,1D14,1D15,1D16,1D17,1D18,1D19,1D20,1D21,2D8,3D2&checkedLevels=0D1,2D1&refPeriods=20180101,20180101&dimensionLayouts=layout3,layout2,layout2,layout2&vectorDisplay=false

[11] Reimers, Nils, and Iryna Gurevych. "Sentence-BERT: Sentence Embeddings using Siamese BERT-Networks." *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*. 2019.

[12] sentence-transformers: https://www.sbert.net

[13] sklearn.linear_model: https://scikit-learn.org/stable/modules/linear_model.html

[14] Chawla, N. V., et al. "Smote: Synthetic Minority over-Sampling Technique." *Journal of Artificial Intelligence Research*, vol. 16, 2002, pp. 321–357.

[15] imblearn.over_sampling: https://imbalanced-learn.org/stable/references/generated/imblearn.over_sampling.SMOTE.html

Appendix A

Group Member Contribution

Rui Zhang: documentation, feature processing, modeling, presentation & report
Ran Wei: documentation, data preparation, EDA, feature processing, modeling, presentation & report
Bite Xiong: documentation, presentation & report