# Restaurants Closure Forecasting on Yelp

**Yonnie Chen**
M.S. in Data Science, LSA
University of Michigan,
Ann Arbor

**Ping-Lun Lai**
M.S. in Data Science, LSA
University of Michigan,
Ann Arbor

## 1 Introduction

The restaurant failure rate is difficult to track nationwide, but the National Restaurant Association estimates a 30% failure rate in the restaurant industry. In other words, one in three restaurants won't survive their first year. Restaurants have been among the hardest-hit industries, especially during and after the pandemic. The project aims to build a machine-learning model that predicts restaurant closure within a one-year period time frame. It could help restaurants, restaurant lenders, and investors decide whether they should lend/invest in a particular restaurant based on the likelihood that it is going to fail within the next few years. Also, it would be alarming for potentially closed restaurants to adjust their future business strategies.

Yelp has become so popular over the last decade that more than 89 million monthly users have been recorded reviewing local businesses on their mobile devices. Reviews are one of the important factors customers have relied on to identify business quality and authenticity. A local consumer review survey published last year shows that 90% of consumers used the internet to find a local business in the previous year, and 89% of 35–54-year-olds trust online reviews as much as personal recommendations. Although Yelp's listings often have hundreds or thousands of reviews, many of those reviews are not trustworthy. Therefore, we decided to use this dataset for predicting restaurant closure. But before that, we would first filter out fake reviews by fake review detection techniques.

We divided tasks into two parts, the project workflow is following:

Part 1: Fake reviews detection
Feature Engineering → Classification → Model Selection → Models Evaluation → Filter out fake reviews → Roc curve interpretation

Part2: Closure forecasting
Feature Engineering → Model Selection → Hyperparameter Tuning → Final Classifier → Deal with Imbalanced dataset → Model Evaluation → Roc curve interpretation

The code for this project can be found in the GitHub repository below:
yonniechan/Restaurants-Closure-Forecasting-by-Fake-Reviews-Detection-NLP-and-ML-Techniques (github.com)

# 2 Datasets

For this project, the two datasets are Yelp datasets for academic updated in March 2022 on Kaggle.com (https://www.kaggle.com/datasets/yelp-dataset/yelp-dataset?datasetId=10100&sortBy=voteCount&language=Python&select=yelp_academic_dataset_review.json).

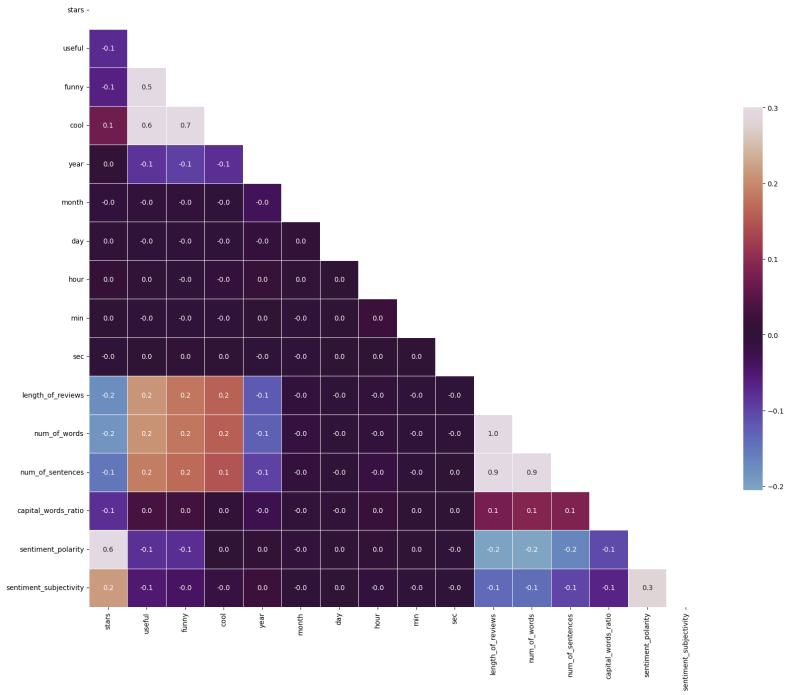
business.json file: It contains 8,814 observations and 14 features. Attributes include *business_id, name, address, city, state, postal_code, latitude, longitude, stars, review_count, is_open, attributes, categories, and hours*. The star column contains ratings on a scale of 1-5. In attribute and hours columns, there are dictionaries that have to expand to obtain more subset attributes.

reviews.json: It contains around 7,000,000 observations and 9 features. Attributes include *review_id, user_id, business_id, stars, useful, funny, cool, text, and date*.

## 2a) Part 1 preprocessing

In reviews.json, there are around 7,000,000 lines of data.

1. Filter out reviews for restaurants that are not in Pennsylvania
   We selected PA because the amount of data in PA is the highest among all the states. The time range of our data is 2005 to 2022.
2. Feature engineering
   We split *date* into *year, month, day, hour, min,* and *sec*. Besides, we generated *length_of_reviews, num_of_words, num_of_sentences,* and *capital_words_ratio* for those reviews. Among them, *num_of words* and *num_of_sentences* are generated using nltk. Then, we generated a heatmap to check the correlation between every feature in the dataset. We found that the time features we created have little correlation with the target variable ‘stars’, so we decided to drop them.



3. Sentiment analysis
   We stemmed, and remove stopwords and punctuation by using **PorterStemmer**, **WordNetLemmatizer**, and other techniques from the nltk library to clean the reviews. After cleaning, we used **TextBlob** to generate polarity and subjectivity from the text.

### 2b) Part 2 preprocessing

After filtering out the fake reviews, we used the groupby function to calculate some statistical data for each business, including the mean of *stars*, *polarity*, *subjectivity*, etc. Besides, in business.json, we broke down those parameters in the form of a dictionary to several columns in dataframe as one-hot encoding. Since there are some values recorded in the dictionary that have undesirable formats, we checked through all of them and adjusted them separately. For the missing data issue, we filled in "false" for these values because each row of the attributes does not contain every aspect.

# 3 Methodology

### 3a) Part 1

First, we defined the case as a classification problem. Restricted from the available dataset, we could only get the dataset with *stars*, *text*, and other attributes mentioned in the datasets instead of the labels about '1' for fake and '0' for not fake. In this project, we used "stars" as the variable we are going to predict in the model. In other words, if our predicted results go far from the actual value, we would consider the reviews fake.

Next, we divided the following tasks into six steps:

1. Classification
   Take only reviews with 1, 3, and 5 stars for the next step because we found there is no significant difference between stars between 2~4.
2. Deal with various variables
   Use **MinMaxScaler** for normalizing the variables and **LabelEncoder** for converting object datatype into categorical or integer.
3. Model Selection
   Try several classifiers including **KNeighbors**, **Decision Tree**, **Random Forest**, **Gradient Boosting**, and **XGBoost** to find the best model for this problem. The **XGBoost** classifier has the best accuracy score, we finally choose this as our final model.

| Classifier | Accuracy score |
|---|---|
| KNeighbors | 0.7256 |
| ** Decision Tree | 0.6715 (Benchmark) |
| Random Forest | 0.7720 |

| | |
|---|---|
| Gradient Boosting | 0.7737 |
| ** XGBoost | 0.7773 (Best Classifier) |

4. Hyperparameter tuning
   The reason we skip this part is because of computing resources limitation. While trying to run the grid search, the process stuck and did not react for over 9 hours.
5. Evaluation
   The final test accuracy is 0.7805.

   The confusion matrix is the following:
   [[189002 11638 40490]
   [ 33880 29224 111906]
   [ 26549 17346 641726]]

   Classification report is below:

| | precision | recall | f1-score | support |
|---|---|---|---|---|
| 0 | 0.76 | 0.78 | 0.77 | 241130 |
| 1 | 0.50 | 0.17 | 0.25 | 175010 |
| 2 | 0.81 | 0.94 | 0.87 | 685621 |
| accuracy | | | 0.78 | 1101761 |
| macro avg | 0.69 | 0.63 | 0.63 | 1101761 |
| weighted avg | 0.75 | 0.78 | 0.75 | 1101761 |

6. Results
   Remove those reviews that have a huge difference between actual values and predicted values (marked in the confusion matrix). Filter out fake reviews to make the review data pure for part 2 prediction.

3b) Part 2

Because our goal is to predict restaurant closure, part 2 is still a classification problem. Using the data retrieved from part 1, the target attribute is “is_open” with ‘1’ for open and ‘0’ for close. We then divided the tasks into n steps:

1. Separate data into three parts
   For model training, we used the data from 2005 to 2021, then split it into training and validation using the train_test_split function. To make the random effects influence smaller, we repeated the process five times. For predicting, we used the data which had reviews in 2022 and had the value ‘1’ in ‘is_open’ since we focused only on those restaurants that are open.
2. Deal with various variables
   First, we use **LabelEncoder** for converting object datatype into categorical or integer. Second, we delete columns with only one unique value in the dataframe to scale down the complexity.
3. Model Selection

Try several classifiers including **LogisticRegression**, **KNeighbors**, **Decision Tree**, **Random Forest**, **Gradient Boosting**, and **XGBoost** to find the best model for this problem. The **Logistic Regression** has the highest accuracy score, so we chose it as our final model due to the high complexity of our input data.

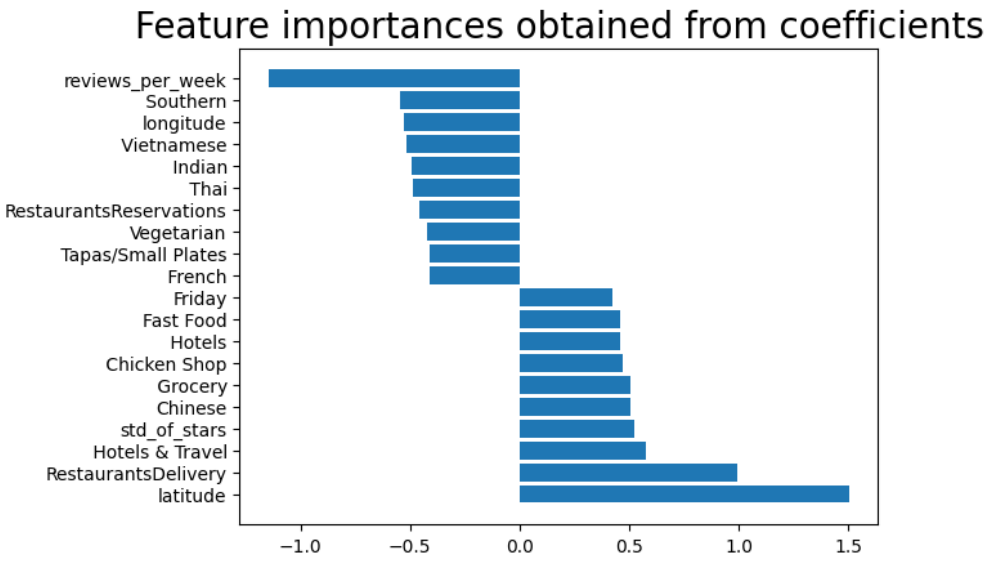
| Classifier | Accuracy score |
|---|---|
| ** Logistic Regression | 0.8887 (Best Classifier) |
| ** KNeighbors | 0.8152 (Benchmark) |
| Decision Tree | 0.8425 |
| Random Forest | 0.8568 |
| Gradient Boosting | 0.8695 |
| XGBoost | 0.8772 |

4. Hyperparameter tuning
   The code block below is the setup for out grid search:

```
X = remain_.drop(columns='is_open')
Y = remain_['is_open']
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size = .2, random_state=0)
params = {'penalty': ['none', 'l2', 'l1', 'elasticnet'],
        'solver': ['newton-cg', 'lbfgs', 'liblinear', 'sag', 'saga'],
        'C': [0.01, 0.1, 1.0, 10, 100]}
model = LogisticRegression()
clf = GridSearchCV(model, params, n_jobs=5, cv=5, scoring='accuracy')
clf.fit(X_train, y_train)
```

   The parameters we get from grid search are:
   'C': 0.1, 'penalty': 'l2, 'solver': 'newton-cg'

5. Evaluation
   The accuracy score and recall score we get from the validation set are 0.7255 and 0.8769 respectively. By these two values, we believe it's sufficient to find the feature importance of the whole model.




Using the coefficient as the importance of the Logistic Regression model, the top 20 features for predicting if a restaurant is open are shown below.

The final test for restaurants having reviews in 2022 has an accuracy of 0.9239. The highlighted value in the confusion matrix below is the number of restaurants predicted as closed, which represents they have a higher chance of closure in the future since they share some features with those closed restaurants before.

The confusion matrix:
[[ 0 0]
[77 935]]

6. Results

From the feature importance above, we analyzed some factors related to a restaurant's closure:

   a. *Food Style*
      As the figure shows, food styles like Indian and Thai have a negative coefficient while Chinese has a positive one. The results might represent the food preference or even the racial distribution in Pennsylvania.
   b. *Restaurants Delivery and Restaurants Reservations*
      Since the restaurants we predicted are those in 2022, it's quite interesting that delivery has a positive coefficient and reservation has a negative one. We believe this can be anecdotal evidence that the food delivery industry has made a huge impact on one's life.
   c. *Friday*
      Surprisingly, the coefficient of Friday is higher than those on weekends. This might show the strategy that many restaurants are using: making more deals on Friday.

# 4 Conclusion

**Summary**

In this project, we performed two parts - fake reviews detection and closure forecasting. We have predicted that 77 out of 1,012, about 7.6% of restaurants would potentially close within a year. The most important factor that defines whether a restaurant would remain open is location. For future work, the model can be improved if we could get further datasets with more demographic features, and more information about the surrounding.

**Challenges**

When developing this project, we faced some challenges. (1) In part 1, we are unable to access the data with fake labels. Thus, after fitting training data into the model, we tried to predict all data and evaluate the performance. Since the goal is to filter out possible fake reviews in all data, we might have a data leakage issue because of the data limitation. (2) The team tried to add more demographic features such as population density, salaries, rent charges, etc. But we found that the scale of these datasets is not desirable so we couldn't use them in our model. (3) The most prominent issue would be the restriction of computation resources. Reviews data is on a big scale and also complicated, which makes the runtime of using nltk and textblob time-consuming.

# 5 Reference


1. https://www.vendasta.com/blog/online-review-monitoring-yelp-reviews/
2. https://daniels.du.edu/assets/research-hg-parsa-part-3-2015.pdf
3. https://www.brightlocal.com/research/local-consumer-review-survey/#summary
4. https://www.nltk.org/
5. https://textblob.readthedocs.io/en/dev/