

Analysis Report: Yelp Restaurants Data Preprocessing and Model Evaluation

Authors: James Galbraith, Hannah Phung, Max Saparov and Pasindu.

Date: August 18, 2023

GitHub: <https://github.com/ajgalbraith/restaurant-survival-analysis>

Abstract

This report details the methods, data preprocessing techniques, challenges, and results associated with analyzing the Yelp restaurants dataset. The ultimate goal is to prepare the data for model building and predict if a restaurant is open or not, based on its features.

Machine learning models, including but not limited to DecisionTree, RandomForest, GradientBoosting, and LogisticRegression, are trained on the extracted features to predict the survival outcome of restaurants over a specific time period. The survival outcome is defined as the ability of a restaurant to remain operational or cease operations during the observed time frame. The predictive models are rigorously evaluated and fine-tuned using various performance metrics such as accuracy, precision, recall, and F1-score. Cross-validation techniques are employed to ensure the robustness and generalizability of the models. Additionally, feature importance analysis is conducted to identify the most influential factors contributing to a restaurant's survival rate.

Introduction

In today's dynamic culinary landscape, the restaurant industry is characterized by a high level of competitiveness and volatility. Understanding the factors that contribute to the success or failure of restaurants is crucial for both entrepreneurs and investors. This project aims to explore the potential of utilizing restaurant reviews from Yelp, a popular user-generated review platform, to predict the survival rate of restaurants.

Besides reviews, Yelp datasets also provide a vast amount of information about businesses, including their attributes such as opening hours and price range. Analyzing this data can yield insights into the factors that affect a restaurant's operational status.

This paper delves into the intriguing realm of restaurant survival prediction by harnessing the combined power of restaurant reviews and attributes sourced from Yelp. We embark on a journey to uncover the latent insights concealed within the vast expanse of online reviews and intrinsic characteristics of restaurants. By leveraging advanced data analysis techniques, including Natural Language Processing (NLP) and machine learning, we aim to construct predictive models that shed light on the factors influencing the longevity of restaurants in this highly competitive industry.

In the early stages of our research, we explored the potential of utilizing Chat GPT for sentiment analysis. Chat GPT, with its vast knowledge base and sophisticated natural language processing capabilities, appeared promising for our needs. However, after rigorous testing and evaluation, it became clear that the model did not align perfectly with our specific requirements for sentiment analysis. The intricacies and challenges faced during this experimental phase can be found in the section titled "Chat GPT Attempt".

Literature review

Customer Experience Analysis through Google Maps Reviews

- **Objective:** Predict restaurant performance in the UK using customer reviews.
 - **Methods:**
 - Google Maps data extraction
 - Sentiment assessment via the VADER tool
 - **Dataset Overview:**
 - Restaurants: 5,010
 - Reviews: 935,386
 - **Key Findings:**
 - **Food:** Dominant factor for achieving 5-star ratings.
 - **Service:** Essential for preventing 1-star reviews.
 - **Atmosphere:** Crucial for turning 2-star ratings into 3-stars.
 - **Value:** Deterrent for 5-star ratings but aids in securing lower ratings.
 - **Valence Analysis:** Alcohol enhances customer experiences, dietary options meet customer needs, while basic food items have subdued impact.
 - **Recommendations:**
 - Address key facets like food, service, ambiance, and value.
 - Strategize enhancements tailored to specific rating tiers.
 - Recognize the contribution of alcoholic offerings and diverse dietary options.
-

Assessing Restaurant Longevity via Customer Feedback

- **Objective:** Predict the viability of restaurants using content from customers.
 - **Methods:**
 - Marrying Aspect-Based Sentiment Analysis (ABSA) with the Conditional Survival Forest (CSF) approach.
 - **Dataset Source:**
 - Extensive Yelp data that encompasses restaurant critiques, ratings, and pertinent details.
 - **Model Description:**
 - Elicit sentiment from reviews pinpointing vital aspects.
 - Forecast longevity using the derived attributes.
 - **Principal Discoveries:**
 - ABSA-CSF model excels over competitors.
 - Geographical positioning and sentiment around 'tastiness' are pivotal for predicting longevity.
 - Influential factors differ according to the nature of the restaurant.
 - **Recommendations:**
 - Acknowledge the role of online feedback in the survival of a business.
 - Strategically distribute resources considering the pivotal factors.
 - **Conclusion:** The ABSA-CSF model proficiently forecasts restaurant longevity using online feedback.
-

Decoding the International Restaurant Success Formula with Yelp

- **Objective:** Decipher the primary elements leading to the success of restaurants internationally via the Yelp Dataset.
 - **Methods:**
 - **Feature Selection:** Univariate method
 - **Classification Techniques:** Array including Naive Bayes, Logistic Regression, SVM, Decision Trees, Random Forest, and GDA.
 - **Text Analysis:** Employing the Naive Bayes classifier for scrutinizing reviews.
 - **Dataset Insight:**
 - Coverage: Restaurants from the US, UK, Canada, and Germany.
 - Source: Yelp Dataset Challenge.
 - **Noteworthy Outcomes:**
 - Universal Attributes: Emphasis on factors like street parking, reservations, volume of reviews, ambiance, noise level, and dress code.
 - Region-specific Traits: Divey ambiance is preferred in the US, parking is a priority in North America, and alcohol availability stands out in Europe.
 - Foremost Model: GDA exhibiting test accuracy rates between 55% and 60%.
-

Methods

Data Used

1. `yelp-restaurants.csv`: Contains attributes of Yelp restaurants.
2. `yelp-reviews.csv`: Contains reviews related to the Yelp restaurants.

Preprocessing Steps

1. **Loading necessary libraries:** `dask`, `distributed`, `AST`, and `tqdm`.
 - The Dask library in Python provides a powerful and flexible framework for parallel and distributed computing. It is particularly beneficial when working with large datasets that cannot fit into memory or when dealing with computationally intensive tasks. Dask enables parallel execution of tasks, which can significantly speed up computations. It automatically divides the data and tasks into smaller chunks and processes them concurrently, utilizing all available CPU core
 - Tqdm library provides progress bars to loops and iterators, allowing the tracking of progress, especially for dealing with large datasets and lengthy computations
2. **Loading and basic cleaning:** Remove columns with identifiers, filter for non-restaurants, and deal with missing values.
3. **Filter popular restaurants:** Only take restaurants that have more than 50 reviews. Restaurants with too few reviews might have risk of bias opinions
4. **Attributes mapping:** Extract and convert attribute columns like 'BusinessParking', 'GoodForMeal', 'Ambience' from nested structures to individual columns.
5. **Feature reduction:** Columns with more than 20% missing values are dropped.

6. **Type conversion:** String values are converted to integer types where applicable using one hot encoding or manual conversion. Some categories are self-translated into numerical based on their values instead of one hot encoding to avoid having too many columns. For example, noise level of quiet, average, loud and very loud would have the value of 0, 1, 2, 3.
7. **Feature encoding:** Some features are numerically encoded, like the 'Alcohol' column being mapped to ordinal numbers.
8. **Merging dataframes:** The processed restaurant attributes are then merged with their corresponding reviews to produce a comprehensive dataset.

Feature Selection

1. Correlation heatmaps were used to identify relevant features.
2. Over and under sampling techniques were applied to address class imbalances.
3. **SelectKBest** was used to identify the top 5 relevant features. Avoid overfitting by using too many irrelevant features that might not have strong predictive values.
4. Principal Component Analysis (PCA) was employed to further reduce dimensionality.

Model Building and Evaluation

1. **Baseline Model:** A RandomForest classifier was trained on the data.
2. **Ensemble Strategy:** Stacking Classifier was used with estimators like DecisionTree, RandomForest, GradientBoosting, and LogisticRegression. Aim to even out errors and eliminate model variance.
3. **Hyperparameter tuning:** GridSearchCV was employed on the Stacking Classifier to find optimal hyperparameters.
4. Models were evaluated using metrics such as accuracy, ROC-AUC, and precision-recall curves.

Challenges & Improvements

1. **Nested Attributes:** Many attributes were nested, necessitating complex transformations to be usable.
2. **Missing Data:** A large chunk of data had missing values. An approach of dropping columns with more than 20% missing values might be too aggressive.
3. **Class Imbalance:** The target variable, 'is_open', may be imbalanced. Random over-sampling and under-sampling techniques were employed to address this, but more sophisticated methods like SMOTE could also be considered.
4. **Feature Selection:** While **SelectKBest** was used, other techniques or domain expertise could further refine feature selection.
5. **Small Yelp dataset:** Yelp's research dataset does not contain big cities like New York or LA. Yelp's API is only for developers, not researchers. Hard to combine with other alternative data sources that are only publicly available for big cities.
6. **Aspect sentiment analysis of reviews:** Using GPT to get sentiment proved impractical (much better and cheaper tools exist, slow api requests, service breaks often, inconsistent responses)
7. **Google Trends Data:**


- **Insight into Popularity:** Incorporating Google trends data can offer valuable insights into the temporal popularity of certain restaurants or cuisines. This real-time popularity measure, combined with sentiment analysis, can enhance the accuracy of predictions by factoring in current trends.
- **Understanding Seasonal Variations:** Google trends can also capture seasonal variations in preferences, allowing the model to adjust predictions based on time of year or specific events, leading to more nuanced and context-aware sentiment analysis.

8. Time Series:

- **Temporal Patterns:** Using a time series approach would allow us to capture evolving sentiment over time. By recognizing and adapting to these patterns, the model could become more predictive of future sentiments based on past trends.
- **Incorporating External Events:** Time series analysis can help the model factor in external events, such as holidays, local events, or global occurrences, which might impact public sentiment. This added dimension could greatly refine and improve the sentiment analysis results.

Discussion of Results

1. Feature Importance:

Correlation heatmaps and `SelectKBest` suggest that certain features like 'stars', 'review_count', and 'RestaurantsPriceRange2' play pivotal roles. Here is the heat map: Correlation Heatmap

Correlation heatmap shows the correlation coefficients, which are statistical measures that quantify the strength and direction of the linear relationship between two variables. Hence, any other types of relationship between the features and the label will not be represent


```
Best 5 features:  
Index(['review_count', 'HasTV', 'RestaurantsDelivery', 'street',  
      'lot'], dtype='object')
```

SelectKBest helps select the top k features that have the strongest relationship with the target variable, but does not take into account the interaction between different features.

2. Exploratory Data Analysis (EDA)

Before diving deep into our modeling, it was crucial for us to understand the nature, structure, and relationships within our dataset. Exploratory Data Analysis (EDA) served as our foundational step in this direction. EDA provided us with insightful visualizations and summary statistics, helping to identify patterns, anomalies, and potential features for our sentiment analysis. One significant aspect of our EDA was the generation of a heatmap, specifically focusing on correlations between different features. Heatmaps are a powerful visualization tool, translating numerical values into a color spectrum. Through this, we could quickly identify and interpret relationships between multiple variables. In our heatmap, features with high correlation were highlighted, indicating a strong relationship between them, while low correlation features were distinctly visible with contrasting colors. This heatmap was invaluable, not only for understanding our data's structure but also for guiding feature selection and engineering in subsequent modeling stages.


3. **PCA:**

The scree plot indicated the importance of each principal component. It's valuable to decide how many components to retain. This is the PCA Plot: PCA Plot

4. **Dealing with imbalanced data:**


after filtering, there are 17060 open restaurants and 5374 closed restaurants. Although there are fewer closed restaurants, the proportion is not too imbalanced since closed restaurants make up about 24% of the data points (usually we have to concern when one label is less than 5%). However, we still tried methods of oversampling and undersampling. Oversampling gives almost near perfect performance, which flags a sign of overfitting since testing data might replicate training data due to bootstrapping.

5. **Model Performance:**

The baseline RandomForest model showed decent accuracy and ROC-AUC values. Model Performance However, the Stacking Classifier, after hyperparameter tuning, showed promise in improving the predictive power.

6. **Precision-Recall Curves:**

Both models (RandomForest and Stacking Classifier) were compared using precision-recall curves, which is crucial in imbalanced datasets.

Precision = True Positives / (True Positives + False Positives)
Recall = True Positives / (True Positives + False Negatives) Precision-Recall Curves

7. **Confusion Matrices:** These matrices provided insights into the true positive, false positive, true negative, and false negative values for the models, allowing for a deeper understanding of model performance.

- Default random forest (no resampling)

	Predicted: Open	Predicted: Closed
Actual: Open	4848	235
Actual: Closed	1152	496

- Default random forest (oversampling)

	Predicted: Open	Predicted: Closed
Actual: Open	4450	656
Actual: Closed	297	4833

- Default random forest (undersampling)

	Predicted: Open	Predicted: Closed
Actual: Open	1090	515
Actual: Closed	524	1096

- Ensemble model with grid search to optimize accuracy (oversampling)

	Predicted: Open	Predicted: Closed
--	-----------------	-------------------

	Predicted: Open	Predicted: Closed
Actual: Open	4844	262
Actual: Closed	414	4716

Chat GPT Attempt

In our pursuit of leveraging cutting-edge AI technologies for sentiment analysis, we turned to Chat GPT. The initial appeal was obvious: Chat GPT's ability to understand and generate human-like text made it a prime candidate for evaluating sentiments within vast datasets.

We began with training sessions and a series of tests to understand how the model gauges sentiments from given text inputs. While the model demonstrated impressive comprehension skills, there were inconsistencies in its sentiment scoring mechanism. For instance, nuances in language, context-dependent phrases, and certain idiomatic expressions posed challenges. The sentiment scores at times lacked the precision we aimed for, especially when the textual data deviated from standard forms.



Additionally, the overhead of integrating and constantly communicating with the model for large datasets was another challenge. The time and computational resources required were not sustainable for our project's scale.

In conclusion, while Chat GPT remains an impressive feat in the realm of AI and NLP, its application for our specific sentiment analysis needs was not optimal. We decided to pivot our approach, focusing on traditional methods and other machine learning algorithms which provided more consistent and reliable results for our dataset.

We used the following prompt to get the sentiment of a review:

```
def get_sentiment(text):
    prompt = text + "\n" + "What is the sentiment of this review?"
    return ask_gpt(prompt)['choices'][0]['text'].strip()

def get_feels_ranking(text):
    prompt = text + "\n" + "On a scale of 1 to 10, how positive or negative is this review?"
    rank = None
    for i in range(0, 10):
        try:
            resp = ask_gpt(prompt)['choices'][0]['text']
            rank = float(resp)
            break
        except ValueError:
            continue
    return rank
```

The key drawback was that the self-reported sentiment differed from the predicted sentiment. The following graph shows the predicted sentiment:  Predicted Sentiment And this graph shows the actual sentiment distribution:  Actual Sentiment

Conclusion

Through rigorous preprocessing and modeling techniques, we've managed to build predictive models on the Yelp dataset to determine if a restaurant remains open. The Stacking Classifier seems to be the more promising approach, outperforming the baseline RandomForest model. This study can be beneficial for stakeholders interested in understanding the dynamics of restaurant operations using Yelp data.

Future research can explore more sophisticated preprocessing methods, employ different feature selection techniques, and utilize more advanced modeling strategies for better results.

Also, since we have only been working with a small dataset, we hope to have the chance to explore our methodology using other sources of data, potentially from Google reviews.