Main Idea

# The objective of this project is to provide data consulting services to Yelp for the enhancement in community engagement and review creditability. For the first part, we will focus on the business attributes of Yelp’s restaurant sector and restaurant scores, to identify top factors contributing to a high or low score. We perform analysis with different techniques for validation purposes. With our analysis, Yelp can help the restaurant owners/vendors identify shortcomings and refine their Yelp profile accordingly to improve their scores. For the second part, we propose applicable technologies such as NLP with implementation details for deceptive reviews identification and handling to enhance the review creditability.

# Potential Benefit

Every restaurant owner wants a high rating, as it is a driving factor in attracting customers. Business attributes such as parking availability, take-out options, and other such features sometimes get neglected by restaurant managers, even though these service attributes greatly contribute to the customer experience. Our analysis would identify the business attributes that are the most relevant regarding a restaurant’s rating/score. This can be an opportunity for Yelp to initiate a consultation service targeted towards restaurants that aids in profile enhancements for higher ratings, such as suggesting which services to provide or how a restaurant should edit their profile. This service will help Yelp gain a competitive edge in the market, as the closest data-driven insights service currently available is the “UberEATS for Restaurant” app, but it offers nothing more than a dashboard summarizing the high-level operation data.

Despite Yelp's slogan being "real people real review", the reliability of reviews on Yelp has been questioned by some marketplace organizations and user communities. Deceptive reviews intend to manipulate business ratings and reviews to put other businesses at a disadvantage. Our proposed natural language processing (NLP) method provides an automated, data-driven detection of deceptive reviews, which reduces human error, saves effort, and ultimately increases customers’ confidence in Yelp.

# Implementation

Yelp collects a large amount of data and as part of this project, seven tables were provided. However, for our proposed initiatives of restaurant profile refinement and deceptive review elimination, most of them are not useful.

**Restaurant Profile Refinement:**

Before jumping into exploring the predictors and building models, we must first take steps to clean the data. For the scope of this study, restaurants are the only business type being considered, so we should first filter out unnecessary variables and rows as it will aid in conserving computational resources. Using a new data frame (df\_restaurants), we can restrict the rows from the yelp\_business table to only include observations that have either “restaurant” or “food” listed in the “categories” column. We can also group related predictors together into smaller data frames such as GoodForMeal or BusinessParking. Aside from making it easier to run going forward, this will allow us to leave out all irrelevant predictors (i.e., predictors that do not make sense in a restaurant setting) such as the columns beginning with “HairSpecializesIn\_” in the yelp\_business\_attributes table. Using inner join to merge these smaller data frames with df\_restaurants through business\_id, we can ensure that only the observations concerning restaurants in the yelp\_business\_attributes table remain and are considered during the next steps. To conclude data pre-processing, we will need to update all the cells containing “Na” in the original tables with numpy.nan and dummify all the categorical/binary predictors. These steps need to be taken as Python cannot directly utilize strings in the subsequent model-building techniques. In terms of dummifying binary variables, we will simply change all the “False” to 0 and all the “True” to 1. At this stage, we can finally begin building models.

A few different techniques may be used to identify the relationship between various attributes of a restaurant and its Yelp score, but the two we will be exploring are Linear Regression and Random Forest Classifier. Before doing Linear Regression, we must first utilize Recursive Factor Elimination to determine which predictors should be used in building our model. While we could in theory forego this step and explore every single predictor, doing this allows us to simplify the final model and only include the most important features, both saving time and increasing the interpretability of our model. As for why we begin with Linear Regression, it is because it is a good baseline to start with for any model where the target variable is continuous.

With that said, we can also transform the target variable of star rating into a categorical one by creating a list of different labels and assigning classification values to the lips, mapping those with <1 star as “very poor” and those with >4 stars as “very good”. The main reason we might want to do this instead is that there is little difference between a rating of 4.45 and one of 4.48 in terms of a customer understanding the overall quality of the restaurant. Doing this will also allow us to take advantage of Random Forest Classifier in which multiple Decision Trees are aggregated together to have more accurate and stable predictions. There are a few benefits to Random Forest Classification, but some of the main reasons that we choose this technique over other classification methods are that it works well with high dimensional data since it splits all the features into a subset at each step and because it is resistant to outliers and non-linear data which may affect our dataset. While it may be computationally intensive to perform, we can begin by utilizing Principal Component Analysis to reduce the number of features for the Random Forest to process. The most concerning for Yelp’s consultants will of course be that the model will lack interpretability, but the model will be a great validation tool.

**Deceptive Review Identification:**

We may examine the content of and identify fake reviews using the natural language processing (NLP) method of sentimental analysis. Three datasets are required as inputs, the first being reviews which we have been given. The second should label industry (restaurant in our case) specific vocabulary as “positive” or “negative” and the last is a list of words that offer no value to the analysis like “the” or “and”. Whilst such datasets can easily be found through the likes of Kaggle, they may be riddled with special characters, inconsistent capitalization, and other problems that Python has in recognizing the text. We must thus begin by correcting these mistakes and tokenize the data (breaking down texts into individual terms).

After implementing transformation and tokenizing the data, we can count the occurrence of positive and negative words in a review to calculate the sentiment score for said review. There are different methodologies for calculating the sentiment score, but our recommended method would be: Degree of positivity/negativity of a review = [count(positive words) - count(negative words)]/count(total words) because it normalizes the score by considering the length of the review.

After assigning a sentiment score to each review, we must then devise a strategy to flag fake reviews. There are mainly two types of such: computer-generated ones (created using AI text generation algorithms) and human-generated ones (negative reviews to harm competitors, positive reviews to improve their image). The issue at hand concerning both types is that Yelp has not recorded any reviews as being suspicious, so we do not have a training set for our model to learn how to identify fake reviews. However, computer-generated fake reviews tend to exhibit a detectable pattern. If a restaurant has a low rating and the fake review’s purpose is to increase the rating, it will fill the written review with “negative” words to by-pass ML algorithms meant to catch bots but give the restaurant a high star rating. In short, there is a discrepancy between the feeling portrayed in the text and the number of stars awarded.

After sentiment analysis, we give weightage to each review as good or bad. If that weightage goes against the star provided by the user for that review, we flag it for a human to label that review as fake/correct. After labeling reviews as "fake" or "correct" using human verification, we may split the dataset and test models like Random Forest Classifier, K Neighbors Classifier, and Bagging Classifier to determine the best model to flag reviews in the future.

# Post-implementation

Unfortunately, the table containing the most information, yelp\_business\_attributes, was riddled with nulls. If we were to remove all the rows with missing values, there would be no rows left. If we remove all the columns with missing values, we will be left with no predictors. As such, going forward, it will be in Yelp’s best interest to require businesses fill out all the information it is able to provide so that the models used in providing restaurants with advice will be accurate and bring them value. In addition, since there are many attributes that are specific to a particular business type, they should be stored on a separate table.

The lack of review flagging allows much space for improvement in the deployment of false review detection. With the use of human verification, we may identify evaluations as "fake" or "correct." We may partition the dataset and test several ML models to choose the one with the greatest accuracy to flag reviews in the future.

We can also establish a connection between the quantity of erroneous reviews and the date the account was created. We can find patterns by merging user and review table, which we can then utilize to enhance our testing models.

Yelp's review table also includes tags like "useful," "funny," and "cool" so we may further enhance our model by examining their relationships with flagged reviews.