Help with Yelp

Project Summary

We aim to create an interactive tool which would allow business owners to gain insight into the factors which may be responsible for the successes and failures of their own business and others in its category. To accomplish this, we will use a business, review, and user dataset from Yelp, a crowd-sourced business review platform.

With our tool, businesses can easily get a sense of the most common attributes of successful businesses (e.g. businesses with an average user rating > 4) in its category (e.g. sushi restaurants in Chicago). A business would be able to search by area, tags, and other attributes which would allow it to plan out a viable opening strategy, for example. We will also allow businesses to update the database from which our tool pulls to limit manual maintenance. For example, a business would be able to indicate that it is permanently closed or that its hours have changed.

Description

Small businesses often fail due to a lack of understanding of customer behavior. However, a wealth of customer data exists; the website Yelp, for example, houses millions of crowd-sourced business reviews. Its customer-centric approach to displaying these reviews limits its utility for businesses. For example, a sushi restaurant in Chicago may want to better understand the behavior of successful sushi restaurants in the area. What is the ambience of these restaurants? Are they open late? Do they have outdoor seating? Currently, it would be both difficult and time-consuming for an individual business owner to parse through this information for every business in its category in the area.

Help with Yelp aims to remedy this; we will offer small businesses a means to gain insights into the factors which correlate with success. Business owners will be able to select the general categories to which their business belongs (e.g. Chicago business, sushi restaurant), and we will use a subset of Yelp customer review data to suggest a business strategy given these categories. Then, by better understanding the needs of their customers, owners can ensure the success of their businesses.

Usefulness

Our application is useful because there is no perfect business; few businesses with a large number of ratings have an average rating of a 5.0 (perfect score) on Yelp. Thus, businesses almost always seek to improve in areas which would allow for a higher rating (to improve the customer satisfaction/“retention rate”, gain new customers, etc.). Businesses must also always balance cost considerations in making these improvements; our tool is also valuable in that it allows owners to gauge the relative importance of possible improvements (based on their bearing in contributing to a high review). We will use techniques like data visualization and, if time permits, machine learning algorithms on data in the same location (as certain locations might have preferences for certain attributes).

Yelp.com and the Yelp mobile application are the most similar existing websites/applications. These tools allow users to view and submit reviews for businesses. They differ from our tool in that they are user-centric; their primary purpose is to allow customers to determine whether to go to a business or use a service. Businesses would not be able to effectively use the Yelp website or mobile app to construct a business model. Doing so would likely involve manually parsing through hundreds of reviews for hundreds of businesses and attempting to find trends. Help with Yelp, on the other hand, is business-centric; we help businesses readily see trends in the strategies of similar successful businesses.

Realness

We will use a dataset from Kaggle, a dataset hosting platform. Yelp published this dataset on Kaggle, and it is a subset of all of Yelp’s crowd-sourced review data. The data is stored in five JSON files each dealing with an aspect of business, user, and review data.

We will divide this data into five tables:

1. Businesses: The Businesses table includes information such as the business’ location, contact information, and other attributes (e.g. Italian).
2. Users: The Users table will let us gather information about the type of review a user makes. For example, one of its columns categorizes reviews as funny, cool, elite, etc.
3. Reviews: The Reviews table indicates the number of stars and category of the review.
4. Compliments: The Compliments table shows the date, text, and number of compliments users have given a business
5. Climate: We will call the Visual Crossing Weather API to find historical weather data corresponding to the reviews in the Weather table. With this information, we will determine if there is a correlation between weather and ratings (e.g. Do businesses receive higher ratings in warmer months?).

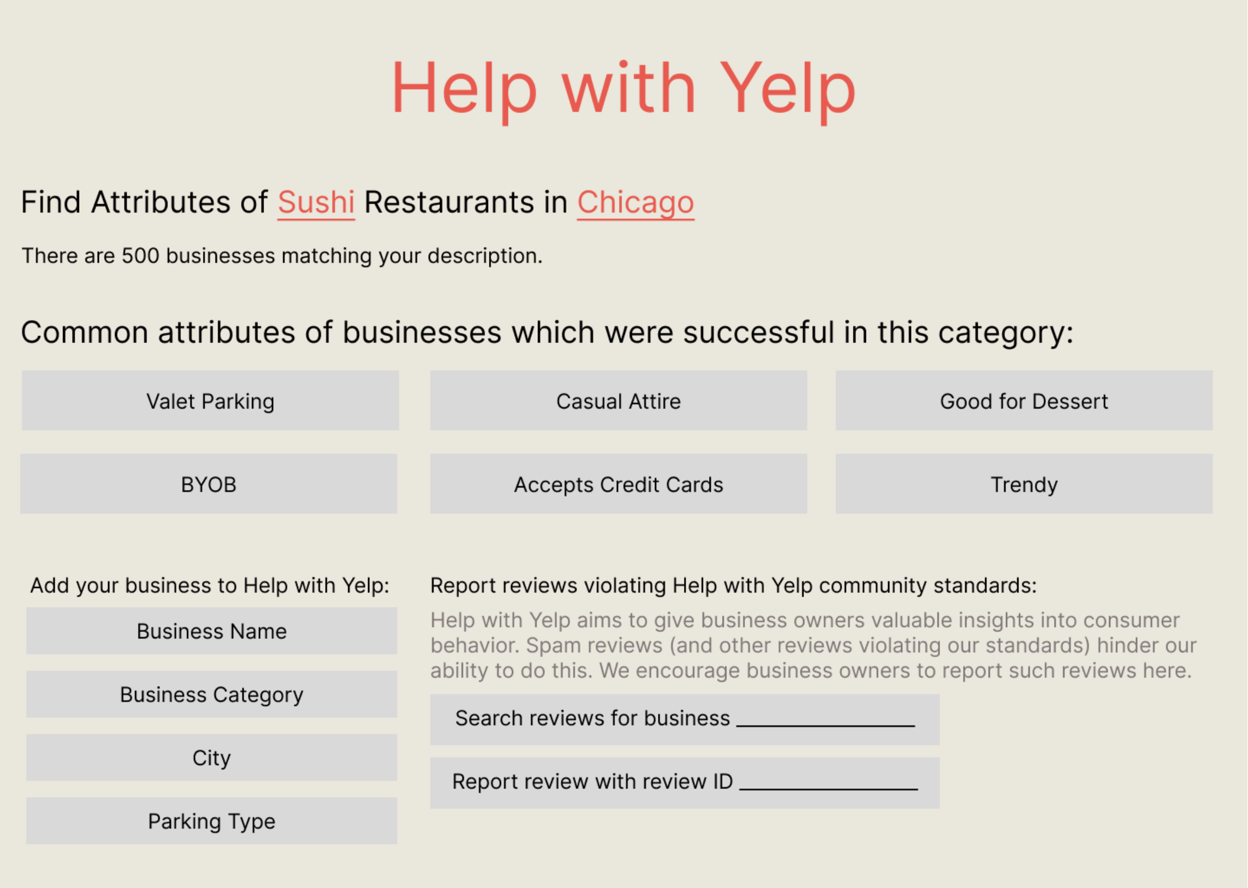
Functionality

By default, the user, a business owner, would be able to search for the attributes of businesses in its category with high reviews (defined as an average rating of greater than 4 stars on a 5-star rating system). For example, the user can search for “Sushi restaurant in Chicago,” and the application would return specific attributes that correlated with restaurant success in the category (e.g., certain opening times, whether or not they accept credit cards, etc). We will also return the most successful restaurants in this category, and the user would then be able to view reviews for those specific top restaurants to get more information on what customers like. The user can then use this information in determining what to emulate for success.

The user can also search for the attributes of businesses in its category with a rating above or below a user-defined threshold. For example, the user may want to see the attributes common in “Sushi restaurants in Chicago” with a rating of less than 2 stars. The user can then use these attributes to identify practices to avoid.

New business owners can add their business to the database, and all business owners can help remove unhelpful/spam reviews from the database. If businesses have updated statistics, they can also update these attributes on Help for Yelp. For example, a restaurant that was formerly open from 9 AM - 5 PM and is now open 24 hours a day can make this change with a form on the user interface.

UI Mockup



Help with Yelp

Find Attributes of Sushi Restaurants in Chicago

There are 500 businesses matching your description.

Common attributes of businesses which were successful in this category:

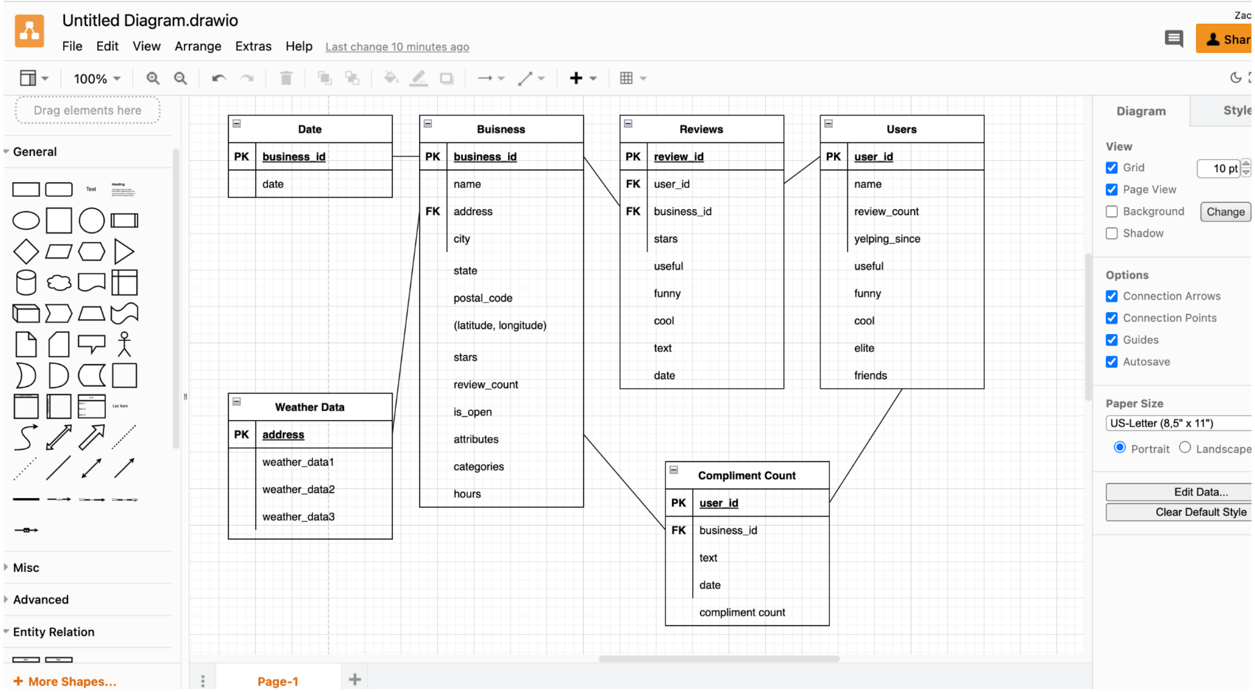
Add your business to Help with Yelp:

Report reviews violating Help with Yelp community standards:

Help with Yelp aims to give business owners valuable insights into consumer behavior. Spam reviews (and other reviews violating our standards) hinder our ability to do this. We encourage business owners to report such reviews here.

Search reviews for business ________________

Report review with review ID ________________



☑ Grid 10 pt
☑ Page View
☐ Background Change
☐ Shadow

Options
☑ Connection Arrows
☑ Connection Points
☑ Guides
☑ Autosave

Paper Size
US-Letter (8,5" x 11")
● Portrait ○ Landscape

Date	
PK	**business_id**
	date

Buisness	
PK	**business_id**
	name
FK	address
	city
	state
	postal_code
	(latitude, longitude)
	stars
	review_count
	is_open
	attributes
	categories
	hours

Reviews	
PK	**review_id**
FK	user_id
FK	business_id
	stars
	useful
	funny
	cool
	text
	date

Users	
PK	**user_id**
	name
	review_count
	yelping_since
	useful
	funny
	cool
	elite
	friends

Weather Data	
PK	**address**
	weather_data1
	weather_data2
	weather_data3

Compliment Count	
PK	**user_id**
FK	business_id
	text
	date
	compliment count

Work Distribution

We would all like exposure to both the frontend and backend of this problem. Thus, we will not rigidly define roles. Instead, we will use a subtask system for a JIRA-esque delegation of tasks. Each week, we will determine subtasks we need to complete to achieve that week's larger goal, assign each subtask a certain number of points based on its relative difficulty (1 for a simple task; 10 for a particularly difficult one), and choose subtasks in a roughly equitable manner depending on our interests (e.g. each team member will be responsible for subtasks totaling 40 points). Should certain tasks be more or less time consuming than intended, we will communicate this on Discord, and we will meet to adjust the workload.

Whenever we meet we try to organize based on deadlines what needs to be done. As a start we will meet once a week with everyone in order to work on either the project of GA's. At the end of the meeting we will go over what still needs to be done and divide the work as evenly as possible between everyone. We will use a React frontend and a Flask backend. Michael and Zach will lead the frontend. Vrush and Maya will work on creating the weather data and developing the backend. We will roughly evenly split general database setup (each team member will be responsible for ~1 table).