Yelp business analysis and location recommendation Report
CMPT 732 Project

Xiaoxiao Duan(xda38)
Jingyi Huang(jha364)
Ziyao Cui(zca123)
Zipeng Liang(zla269)

Problem Definition

The Yelp challenge dataset was chosen because we want to work on a scope with potentially real business needs. After doing a simple explanatory analysis of the dataset and integrating the feedback from the project proposal, our team decided to initiate a project on offering data analysis targeting business owners and potential investors by concentrating on the following products: a dashboard for an overview of the market, an analysis of users' reviews using natural language processing to provide insight, and a machine learning algorithm offering recommendation about where to choose the location for the chain restaurants. A specific region, Alberta province was chosen for project demonstration.

Challenges in Data Processing:

1. Use the restricted/limited dataset to train the Machine Learning model.
2. Train and test by Scikit-learn tools: the 'BallTree' model to generate the store's neighbors' indices and the distances from the corresponding store, then report each neighbor's information.
3. Some bad reviews start with compliments so top words could be the same as good reviews. And reviews are very objective and we can't decide if it's useful information only based on one overall rating.

Challenges in Visualizing data:

4. Make use of the suitable libraries in frontend to draw informative data visualization charts.
5. Identify the data we need when drawing each data visualization graph/map.
6. Ensure low latency when loading the data visualization charts with thousands of data records.

Methodology

Application pipeline is shown below:

Data Processing: Spark, Spark SQL, Pandas

Machine Learning: Scikit-learn

Scikit learn[2] is a free software machine learning library for the Python programming language. It integrated well with many other Python libraries, such as Pandas DataFrames, and Numpy for array vectorization.

Database: MongoDB
The MongoDB Atlas was chosen as the data storage for the web application for several reasons. First, this non-relational document database provides a flexible data model that enables storing unstructured data, which suits storing JSON files that this project requires. And then, scalability: the cloud databases can be offered as a managed database-as-a-service; finally, its simple and expressive query API allows the data to be retrieved easily.

Web backend: flask
This lightweight web framework allows a wide variety of plugins. It can work easily with the python libraries that the dashboard requires.

Web frontend: React.js, Material-UI.js, Chart.js, leaflet.js, Leaflet.markercluster.js,
A variety of JavaScript libraries were used in building the frontend modules.
React is one of the most widely used and the most popular frontend JavaScript framework in the world. It is very easy and suitable for building a single-page web application such as our project.
Material-UI.js is the popular CSS library used with React framework.
Chart.js was chosen because we could easily master it and use it to draw some beautiful graphs for our data visualization. In addition, it integrated well with the React framework.
Leaflet.js/Leaflet.markercluster.js was chosen over Google Maps because it supports more customized components and design and many easy-to-control interactive layer groups.

Natural language processing:
Natural language processing helps resolve ambiguity in language and adds useful numeric structure to the data for further analysis, using NLTK tools for text processing.

Problems
Dashboard:

1. Redefine the category for each business to provide informative dashboard components
2. Add a cluster to provide meaningful location visualization

In the business.json file, the attribute “categories" are similar to the concept of tags. Each consists of a list of descriptives originally added by the business owners, which poses difficulties for visualization. To solve this problem, the frequency of each “tag” in all businesses was extracted and counted using spark, and top categories were selected for reassigning each business.

Location visualization for big data is challenging. By utilizing appropriate JavaScript libraries to create clusters, multiple-layer groups with a larger amount of markers (up to 10000) and other display components can be added with minimal lagging.

Natural language processing:

1. Clean the text data
2. Append the top words for each business into a file

Another technique we used is sentiment analysis. By analyzing customer reviews, we can understand how they think positively or negatively about the restaurant. We hope to help each business to improve their customer's experience, so we did sentiment analysis on reviews. We first divided the dataset into two parts based on review stars. One is for good reviews and the other is for bad reviews. The process starts with tokenizing the sentence to split long reviews into smaller units. Next, we converted all letters into lowercase. Then, we got rid of the stop words, since these words normally don't pack much meaning in a given context. We also added some neutral words as stop words in the reviews, for example, "restaurant", and "place". In order to make the word cloud dynamic, we designed a function to automate the process of generating the top 20 words and frequency in each business. We could give restaurant business suggestions based on top words in customer reviews.

Machine learning:

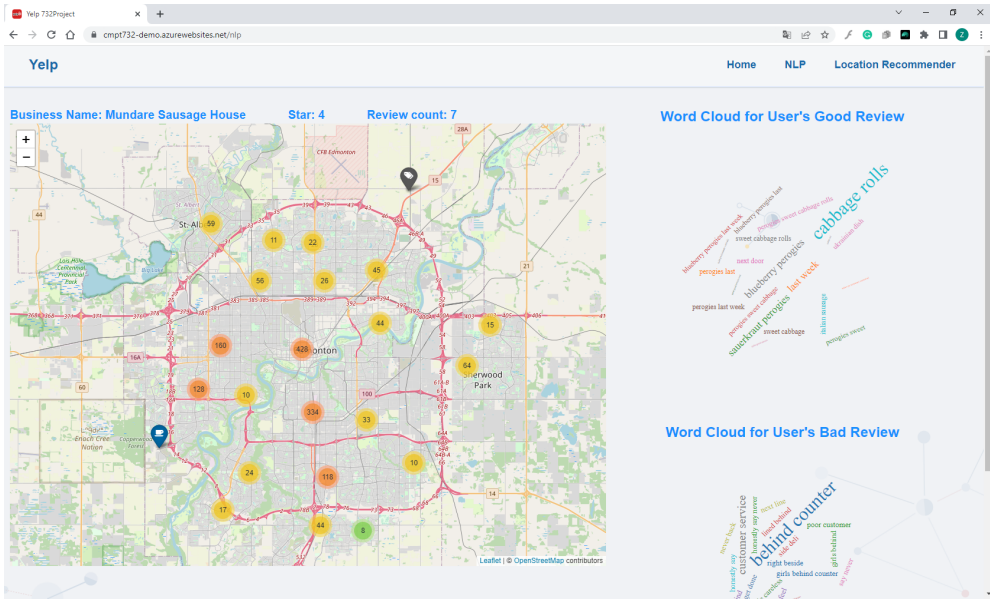
1. Transfer between Numpy array and pandas DataFrame
2. Finding the corresponding neighbors of each store in the dataset

We first intended to use another model to produce the neighbors' information, but this did not turn out to be the case. As a result, we implemented Scikit-learn NearestNeighbors, which offers capabilities for both supervised and unsupervised neighbor-based learning approaches. To be more precise, we extracted and cleaned the dataset after importing the data into a Spark DataFrame, excluding the chain restaurants. The data is then divided into a 25% train set and a 75% test set after being transferred from Spark DataFrame to Pandas DataFrame, and we also store the test data in a .csv file. But the training data is actually our entire dataset since we want to generate each store's neighbors. Next, we determined the separations between two points using the built-in API of BallTree and the "distance" method. As a result, we can generate the query point's neighbors at random inside the specified range, which our users decide. We also included a new column called "counts" that displays the number of stores that each store has as neighbors. Finally, we save this as a trained model called 'neighbors-model' and output info as out.zip. Now, we could use the test-data.csv to generate the neighbors' indices and distances from the stores you want within any range in the test data.

Results

The outcomes of the project have shown in the following video, which is a short overview of the website. Link: video link

1. Web dashboard and NLP

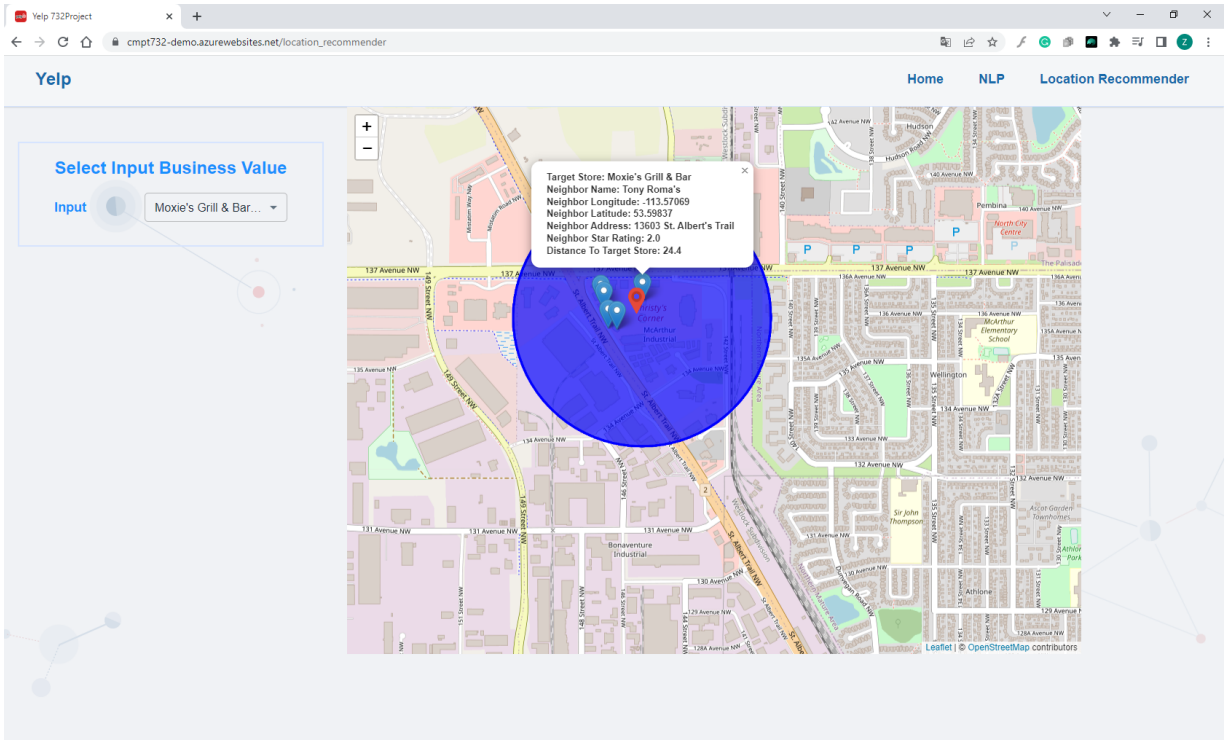


2. Machine learning

These are parts of our data frames and test data:

```
Pandas DataFrame:
          id                      name   latitude   longitude    i
0        105                    Subway  53.518303 -113.505608    0
1        173                  Filistix  53.538696 -113.504257    1
2        335  Big Al's House Of Blues   53.570473 -113.536293    2
3        338             Grandma Pizza  53.494556 -113.578392    3
4        431             Mucho Burrito  53.467327 -113.494499    4
..       ...                       ...        ...          ...  ...
742   149663               Cinnaholic   53.517929 -113.496613  742
743   149686                  Wok Box   53.540787 -113.494522  743
744   149737              Burger Baron  53.506645 -113.486649  744
745   150192                Top Donair  53.607231 -113.468224  745
746   150213        Bourbon St. Grill   53.522312 -113.622429  746

[747 rows x 5 columns]
Train data looks like:
 [[53.47426223754883 -113.39321899414062]
 [53.591365814208984 -113.41795349121094]
 [53.49844741821289 -113.51255798339844]
 ...
 [53.52376937866211 -113.61653900146484]
 [53.54643249511719 -113.52555084228516]
 [53.434814453125 -113.603034973144531]
```



Lesson learned - Data Analysis

Based on the data we have, we found that:

1. Every region has a very prosperous area of its own. Businessmen/businesswomen usually choose the city center first. Additionally, the second store's site may be established further from the city's core. Since Edmonton has fewer restaurants than some major cities, it is possible to estimate the city size and its levels of economic prosperity by counting restaurants and other retail establishments.
2. Data Analysis and Data Visualization are closely related. Visualization can help us observe data more intuitively. We could imagine data visualization as the graphical representation of information and data in a pictorial or graphical format. From our website, we could find that the average star of business is between 3.5 and 4. And restaurants account for a large portion of Edmonton's business.
3. Sentiment analysis is very important for a business. It helps the business to communicate better with customers and develop more relevant messages. A business doesn't want to guess customers' feelings and emotions, so generating word clouds can help businesses through visualization in the form of tags or words and make data-driven decisions.

Lesson learned - Implementation

If a good beginning is half done, then a good plan is a half project done. After we made the plan for our project, to know what to do next, we first deal with the data's structure and determine the

types of purposes we have for it, then it would be easier to implement it in the future. Data processing well will increase the effectiveness of any subsequent visualization because data is a crucial building block.

After finishing this project, we realized that Spark is a good tool that could handle big data. Moreover, we found that React is a nice frontend framework to build a single webpage and we also learned how to use some common JavaScript libraries to draw graphs for data visualization.

Project Summary

Getting the data	0
ETL: Extract-Transform-Load work and clean the dataset	1
Problem: Work on defining the problem itself and motivation for the analysis	3
Algorithm Work: Work on the algorithm needed to work with the data, including integrating data mining and machine learning techniques	3
Bigness/Parallelization	3
UI	4
Visualization	3
Technologies	3

Reference:
[1] *Yelp*. Retrieved From https://www.yelp.com/dataset
[2] *Scikit learn*. 1.6. Nearest Neighbors. Retrieved From https://scikit-learn.org/stable/modules/neighbors.html