Final report

Our group project: “Yelp business insights”, mainly want to stand on investors’ perspectives to see which states’ restaurants are worth to invest. And there are mainly five parts in our project as shown in Fig 1.

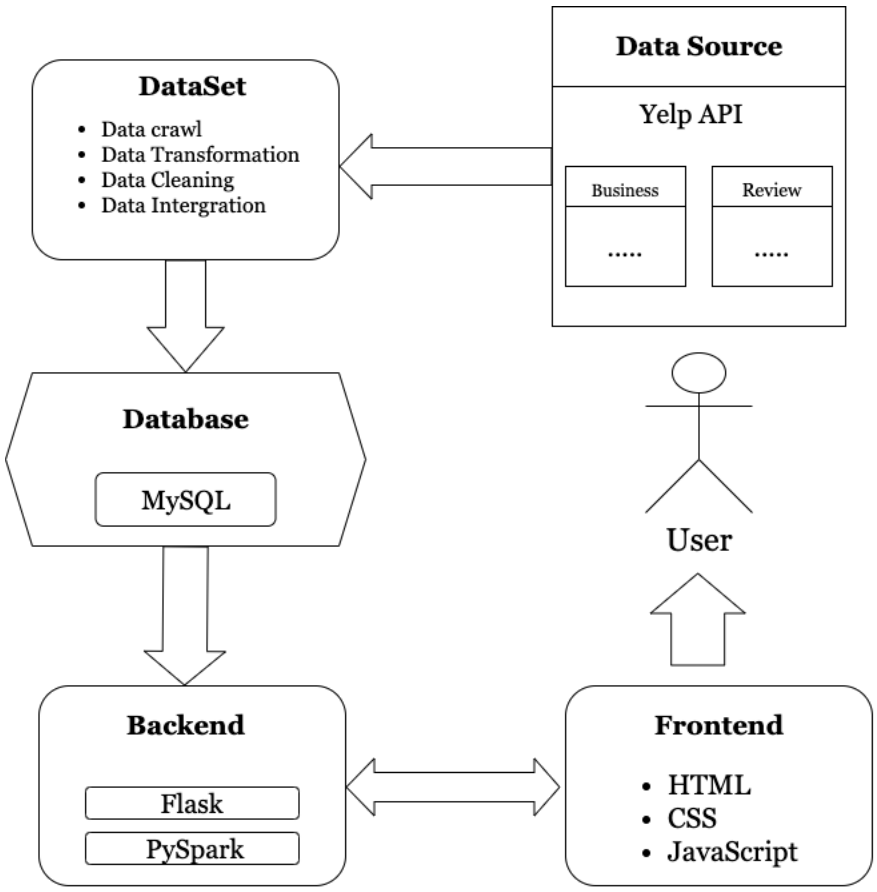

Fig 1: the project architecture

We collect our dataset by using Yelp’s API. https://api.yelp.com/v3/businesses/{id} and https://api.yelp.com/v3/businesses/{id}/reviews. From these two APIs, we can gain restaurant basic information, including each restaurants’ rating score, rating count, geographical location, belonged states, and so forth. We also explore representative comments for those restaurants, to judge whether this restaurant is worth trying or should avoid. Then, we use data crawling techniques in python to generate two csv(business.csv and reviews.csv). business.csv contains comprehensive information, like illustrated above, and reviews.csv contains users’ comments for each restaurant in business.csv. Since our dataset are well-structured

and organized into format repository and it only contains text and number, Thus, MySQL is most suitable to store data. We build two SQL tables corresponding to the business dataset and reviews dataset. And we use these SQL tables in later parallel processing.

Since our dataset is large, exceed 140000 rows in each dataset, so we choose Spark to process data in parallel. By using Spark, memory-based operations tasks can be handled faster and Spark also implements a DAG execution engine that can effectively process data based on memory. Thus, we use Spark to calculate statistical parameters for overall restaurant information in each state of the US, including average score rating, total restaurants opened, fluctuation patterns of rating scores, standard deviations, and most popular dishes' types. Then, we send each state's summary report into backends.

In the front-end and back-end part, we have achieved three main functions:

1. On the home page, we will show our product logo and brief introduction.
2. Users can select the state they are interested in and submit their email addresses.
3. After the back-end receives the information from the front-end, the back-end will automatically use Spark to deal with our dataset and generate the state's restaurant analysis report and then send it to the user's mailbox.

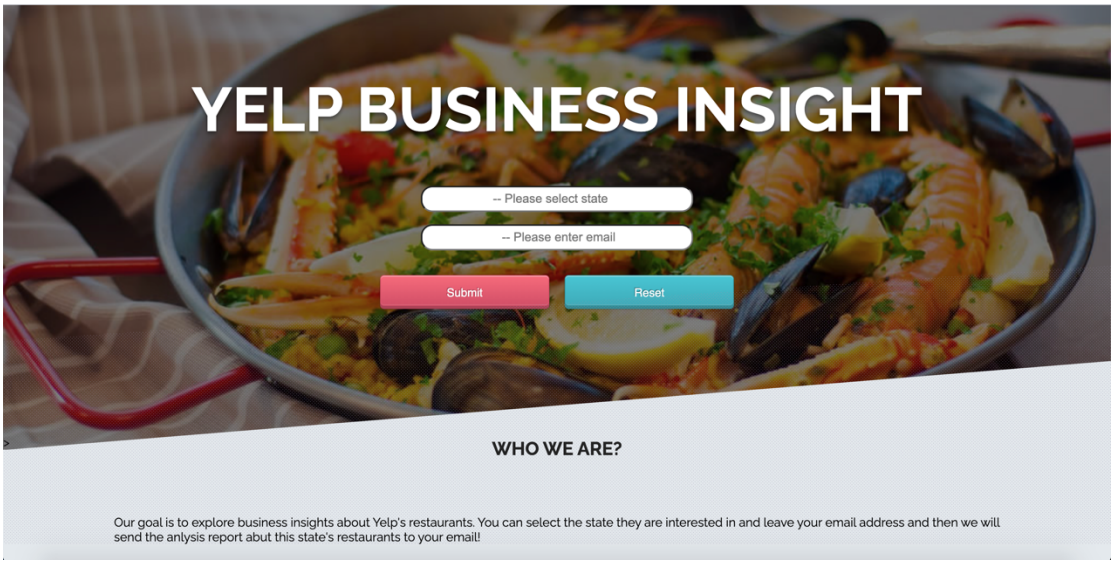

Fig 2: homepage of our website

The user can see the title and brief introduction of our product. The users can select the state they are interested in, and then enter their mailbox. If they hit the submit button, the front-end will send request to the back-end.

As the back-end, we use the flask as the back-end framework. The back-end will parse the parameters passed by the user, and the interface with these parameters will send the specific state's restaurant analysis report to the user. The jdbc connection driver is used to read the dataset from the database and PySpark is our tool to do parallel processing. Fig 3 is an example of the specific state's restaurant analysis report.

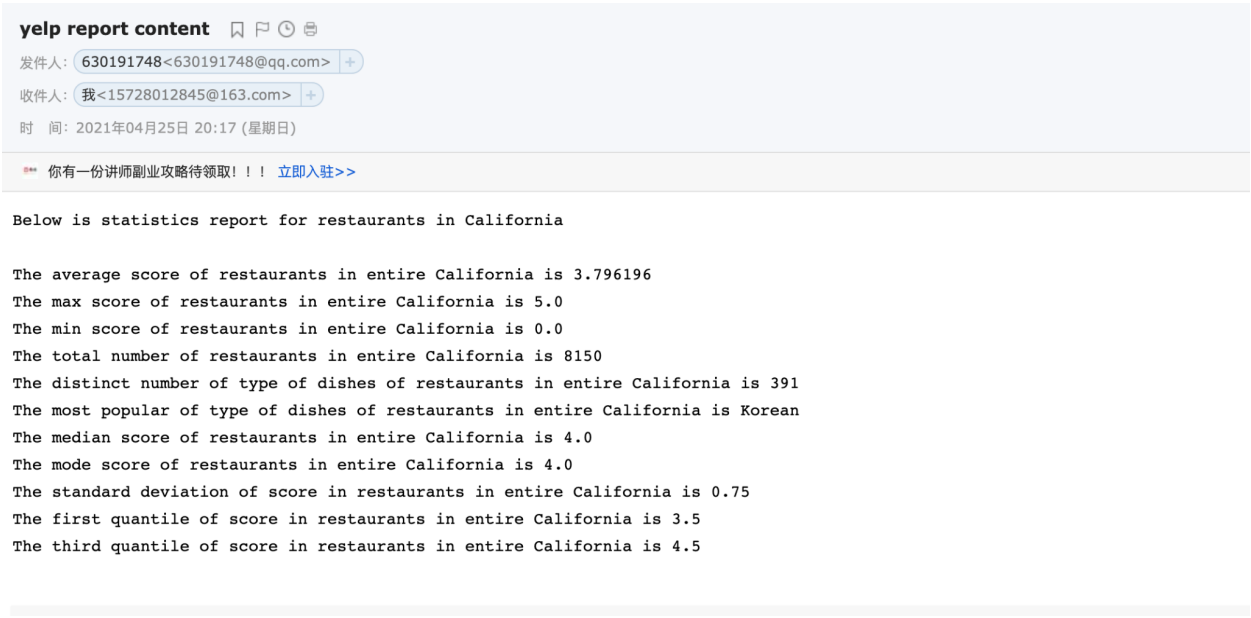

Fig 3: An example of the specific state's restaurant analysis report

Group members information and learning experience:

Ruichao Ma:

Graduated from University of Wisconsin-Madison, with majors in math and statistics. Have experiences in python, R and SQL, Spark. Have data science skills including data warehouse, machine learning, data mining and deep learning.

In this project, I learn how to use APIs to crawl data and ways to store data in suitable database. And also learn how to use big data platforms (Hadoop, Spark) to handle data and some front-end knowledge and UI principles. Very valuable learning experiences.

Wenxuan Li:

Graduated from Zhejiang University of technology, with majors in automation. Relevant past experience has focused on data science issues in the field of transportation. Moreover, has nearly one year of working experience in the application of image algorithm and logic algorithm in intelligent high-speed scene.

In this project, I learn how to use API to get data source and then clean, transform and integrate these data source into useful dataset. Also, I have a simple understanding of the front-end(HTML, CSS, JS) and back-end(Flask) technology, and can achieve certain functions.

Data Science Challenge

We finish data challenge created by group 7. In this data challenge, we mainly need to calculate twitter user count in different time zone in one day. Since half hour is one unit, one day with 24 hours can be divided into 48 partitions. Then we need to do visualization work to show trends of twitter user count in one day. Below is screenshot of our plot.

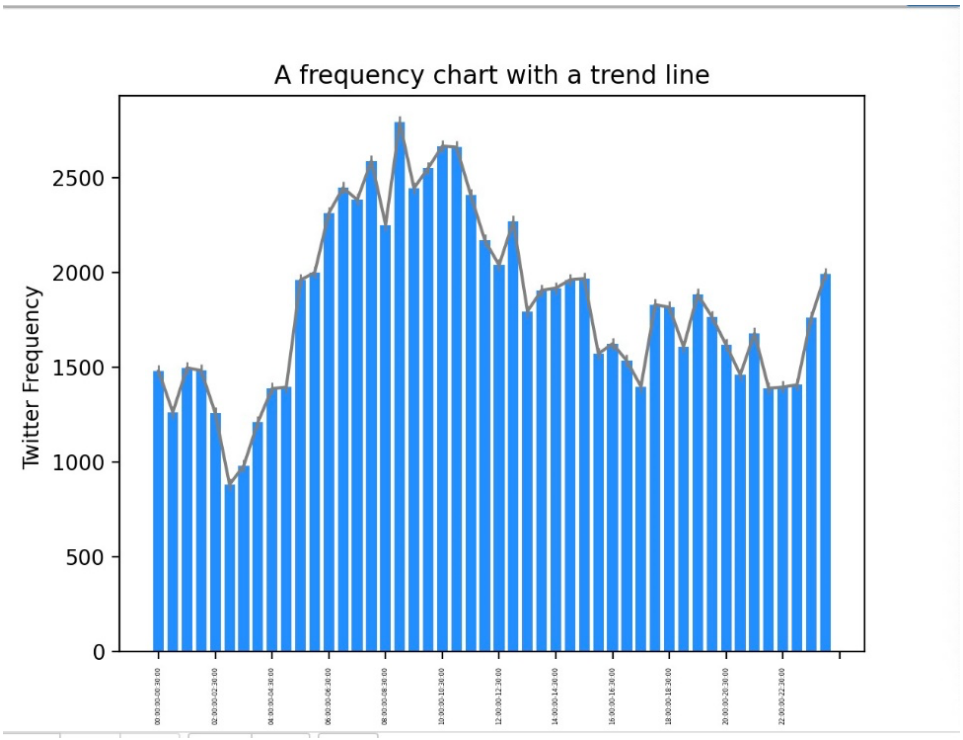

Fig 4: A frequency chart with a trend line

This plot directly shows that from 2:00AM to 8:00AM. The trend is dramatically increasing, meaning lots of twitter user increase in this time period. However, from 11:00AM to 5:00PM, the trend gradually decreasing. From 5:00PM to 11:59PM, there is small fluctuations in twitter frequency.