Final-Report

Group number: 85

Members: Yunjie Wu, Yunhe Wang

In order to do some user-defined filtering and statistical analysis, or even make a high-quality recommendation system. Large datasets with specified quantitative evaluations of business from a huge number of users are valuable and necessary. The Yelp Review Dataset is an appropriate choice with all the information we need. Based on this dataset, we can emulate a distributed file system and create an UI to implement various searching and analytic functions.

The Dataset

We collect the dataset from the official offset Yelp Dataset and use three datasets: business.json, user.json, and review_train.json. Since the dataset could be updated regularly, when using the UI for some analysis, we rename the datasets as the format like: business_2022_10_31.json in order not to operate the files with the same name. The below figures show the sample of the dataset. Since the datasets are all structured dataset, we use MySQL as our backend database.

business.json

Contains business data including location data, attributes, and categories.

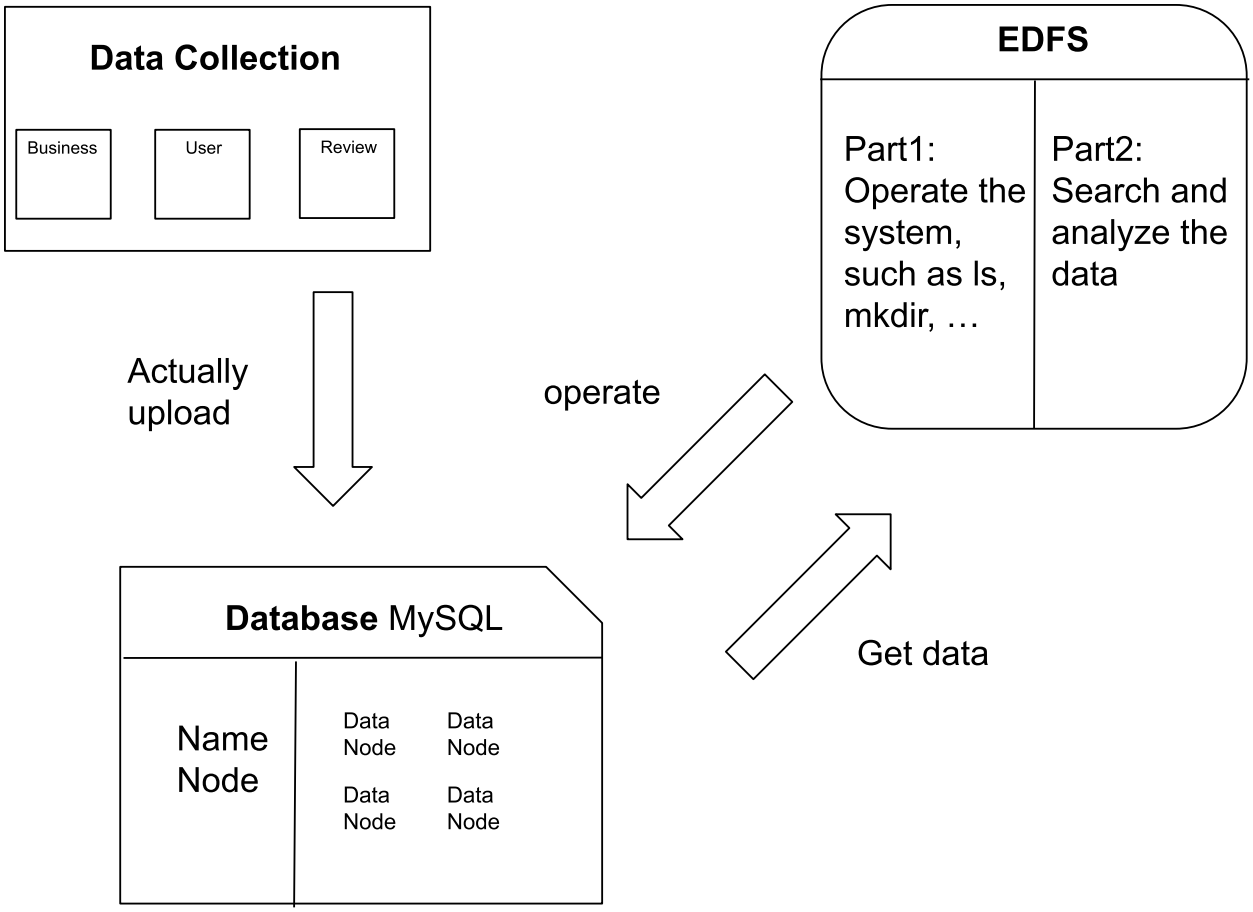
```
{
    // string, 22 character unique string business id
    "business_id": "tnhfDv5Il8EaGSXZGiuQGg",

    // string, the business's name
    "name": "Garaje",

    // string, the full address of the business
    "address": "475 3rd St",

    // string, the city
    "city": "San Francisco",

    // string, 2 character state code, if applicable
    "state": "CA",

    // string, the postal code
    "postal code": "94107",

    // float, latitude
    "latitude": 37.7817529521,

    // float, longitude
    "longitude": -122.39612197,

    // float, star rating, rounded to half-stars
    "stars": 4.5,

    // integer, number of reviews
    "review_count": 1198,
```

review.json

Contains full review text data including the user_id that wrote the review and the business_id the review is written for.

```
{
    // string, 22 character unique review id
    "review_id": "zdSx_SD6obEhz9VrW9uAWA",

    // string, 22 character unique user id, maps to the user in user.json
    "user_id": "Ha3iJu77CxlrFm-vQRs_8g",

    // string, 22 character business id, maps to business in business.json
    "business_id": "tnhfDv5Il8EaGSXZGiuQGg",

    // integer, star rating
    "stars": 4,

    // string, date formatted YYYY-MM-DD
    "date": "2016-03-09",

    // string, the review itself
    "text": "Great place to hang out after work: the prices are decent, and

    // integer, number of useful votes received
    "useful": 0,

    // integer, number of funny votes received
    "funny": 0,
```

The idea to implement the api



In MySQL, we create 5 database to store the information about the data, 4 “DataNode”, which are used to store the actual files, and 1 “NameNode”, which are used to store the metadata of the files.
In NameNode database, there are 4 tables records the metadata.

metadata

It records the **meta_id. name, type, “hasFile”** of each node. One node is a directory or the file. For example, if there is a path in the EDFS: /yelp/user/user2022_10_31.json, then it will contains 3 node “directory:yelp”, “directory:user”, “file:user2022_10_31.json”. If the node indicates a directory, “hasFile” will record how many actual files under this

directory

It records the parent-child relationship between each nodes, If there is a path “/yelp/user”, then “yelp” is parent node, while “user” is the chilid node.

partitions

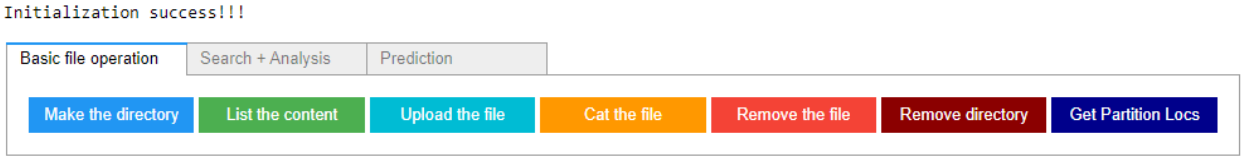
It records the **file_meta_id, partition name of that file and the DataNode the partition belongs to**. Basically, each time the api upload a file, it will be splitted into many partitions, and for each partition file we create **3 replicas** and distribute into different NameNode database. This table is used to record these information.

Fileinfometadata

It records some information about the file nodes like **meta_id** of it, **file content** of it (business, user, or review), **partition number** of it (which is default 5 but could be set by the api), **file name and location** on the file system.

Api

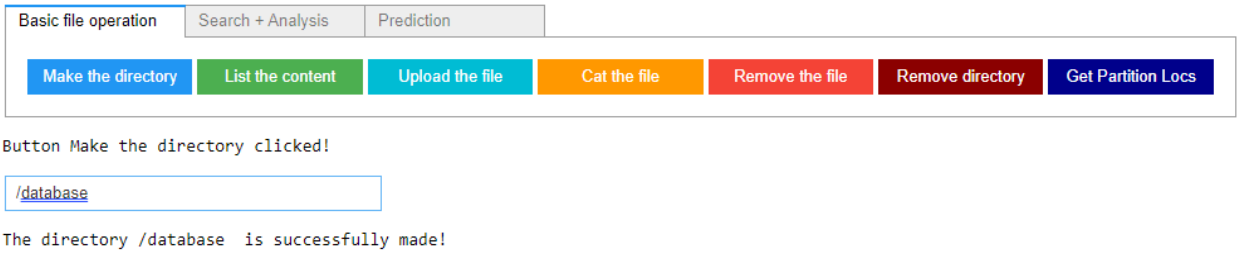
Run the function to call api using the host, port, user, password of the MySQL on your local machine, and you would see the following UI. The UI is based on jupyter notebook. You can run the cell on the Jupyter notebook to refresh the UI.



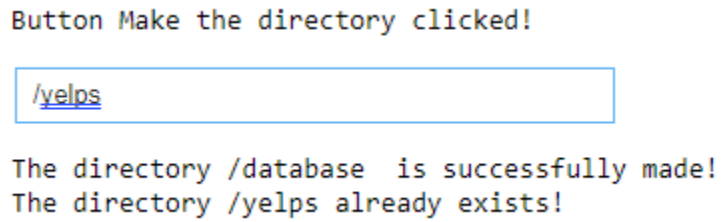
Part1: basic system operation (Wu)

Make the directory:

This will create a directory as the user input. Press the button “Make the directory” then the user could use this function, the same as other functions.

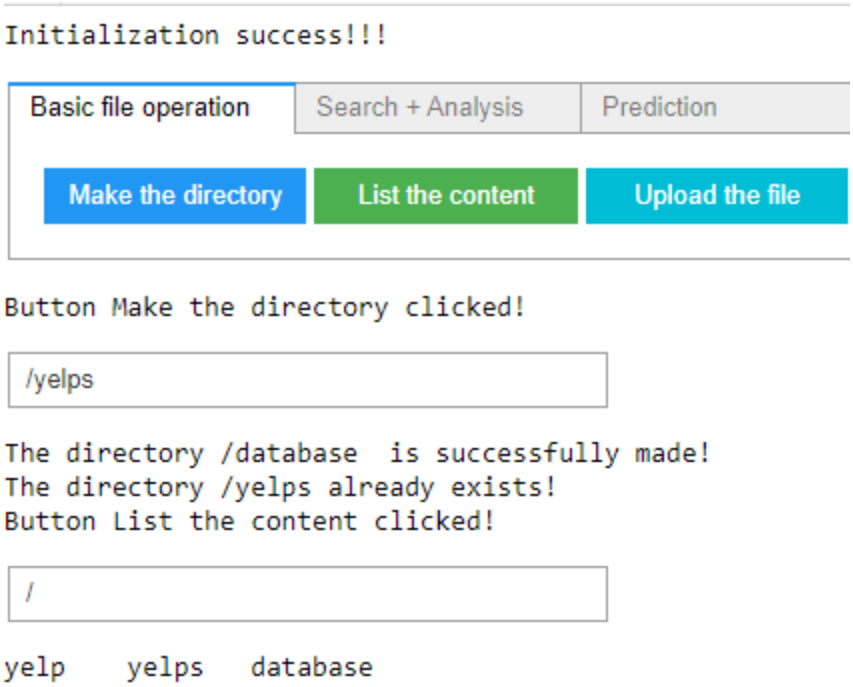


After the user input the path, press [Enter] to run the function. If you input the existing directory, then it will tells that it has been created.



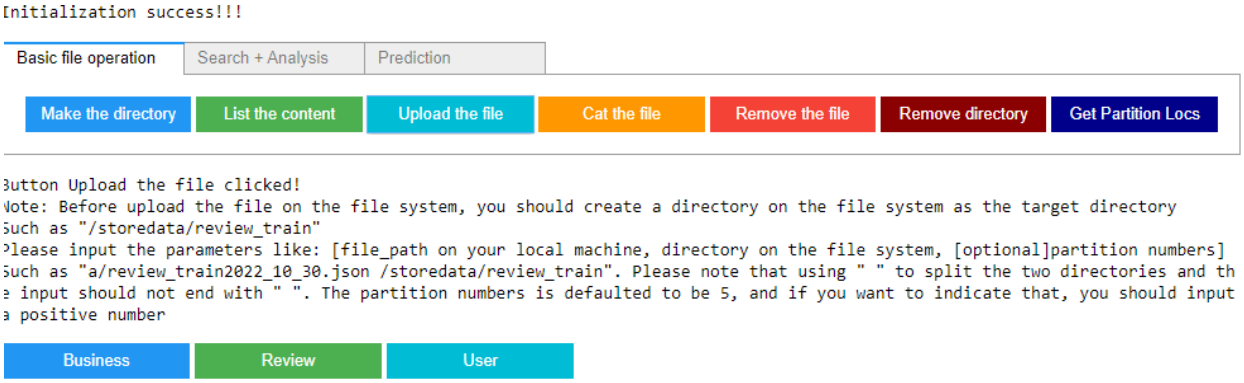
List the content

It will return all of the child node the input path has. If the path does not exist, then it will tell the not existing path.



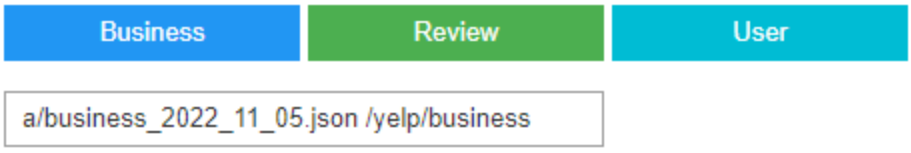
Upload the file

It will show 3 buttons for each content. If you want to upload a business file on the system, you should choose business button. If you choose others, it will fail.

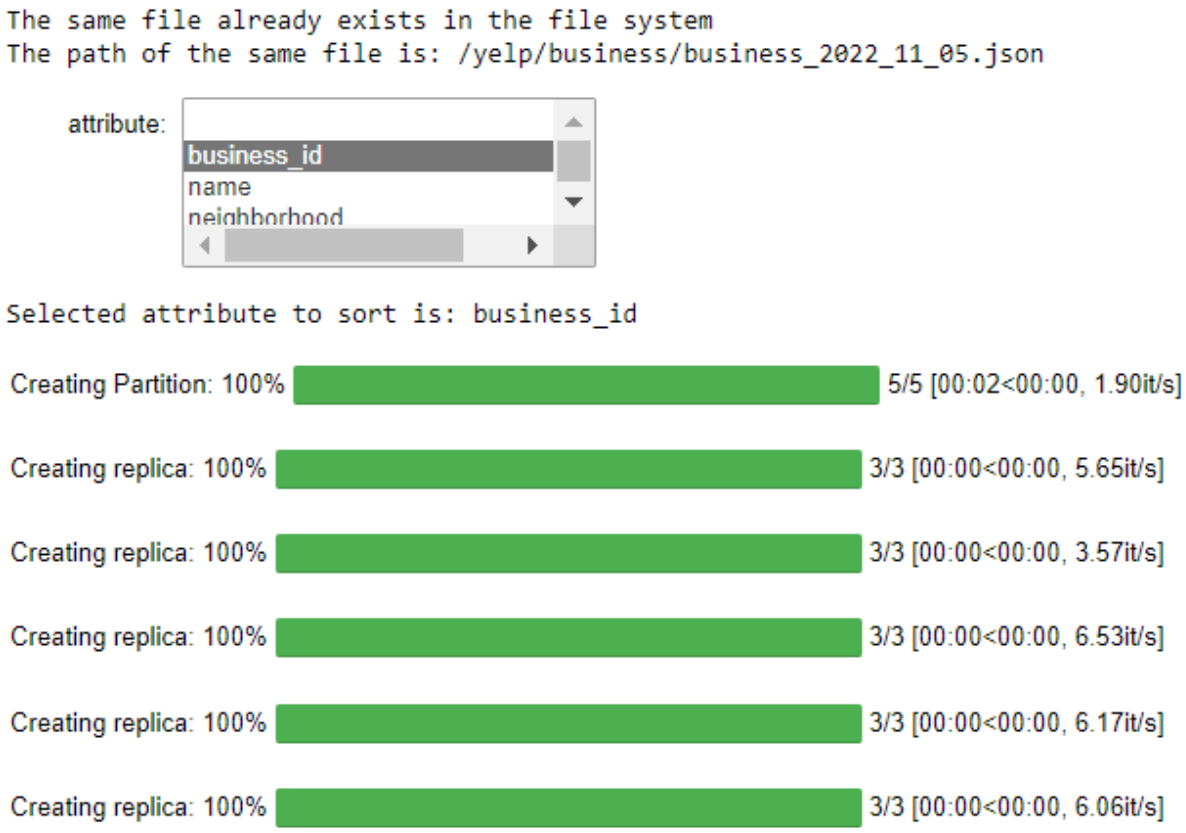


The user should input the parameters like: [file_path on your local machine, directory on the file system, [optional]partition numbers]

Such as "a/business_2022_11_06.json /yelp/business". Please note that using " " to split the two directories and the input should not end with " ". The partition numbers is defaulted to be 5, and if you want to indicate that, you should input a positive number. If the file with the same name has existed in the file system, it will tell the information.



After input the parameters in the text box, press [Enter] and the api will show the attribute which is used to sort the file and be used as the split key. Select one attribute of it and then upload will begin.



Each partition will be replica 3 times and be uploaded into different DataNode.

Cat the file

It will show the subset of the dataset. Due to the showing limit, this function only shows at most 20 lines of the data. The user should input the parameters like: [file_path on the file system, offset line, length of the data]. If the subset of the file cover two partitions, then it will automatically search the contents in these two partitions.

```
Button Cat the file clicked!
Due to the showing limit, this function only shows at most 20 lines of the data
Please input the parameters like: [file_path on the file system, offset line, length of the data]
Such as "/storedata/business/business_2022_10_30_11.json 37710 20". This means that reading the file from 37710 line to 37729 l
ine
```

p/business/business_2022_10_31.json 37710 20

	business_id	name	neighborhood	address	city	state	postal_code	latitude	longitude	stars	...	attributes.Music	attr
0	BqGM1NCltdaweP_WB1Klhg	Bisteces		10649 N 43rd Ave	Phoenix	AZ	85304	33.583473	-112.150986	4.0	...	None	
1	BqGWgwUAUi9rtlh_bKNMNg	Saguaro Landscaping & Pools		835 W Warner Rd, Ste 101-475	Gilbert	AZ	85233	33.334041	-111.807576	2.5	...	None	
2	BqHX49ixrSuPoMaZ9Lzz1g	The Nailery		1100 Grove Rd	Pittsburgh	PA	15234	40.367521	-80.014219	2.5	...	None	
3	BqIbxfNBEHbjHuYb2Y2ZvQ	AA Auto Care	Southeast	2070 E Warm Springs Rd	Las Vegas	NV	89119	36.057795	-115.123905	3.5	...	None	
4	BqKFAnGL8MLI7NtYNUhSYg	Boîte à Fromages	Ville-Marie		Montréal	QC		45.508670	-73.553993	3.5	...	None	
5	BqKq0tKOAEvyHdjGtWcEHQ	The Gaming Goat	Westside	4575 W Charleston Blvd	Las Vegas	NV	89102	36.158628	-115.202574	3.0	...	None	
6	BqPDyBgRnvPWCTcMjonnxA	The Shave Shack	Spring Valley	7095 S Durango Dr	Las Vegas	NV	89147	36.059403	-115.279713	4.0	...	None	

Remove the file

This function will remove the file on the system. If the input is not a file, it will warn.

```
Button Remove the file clicked!
Please input the file path you want to delete. End with ".json"
```

/datanode

```
Please search for a file to delete, rather than a directory
```

```
Button Remove the file clicked!
Please input the file path you want to delete. End with ".json"
```

/yelp/business/business_2022_11_09.json

```
The file has been deleted
```

Remove the directory

This is used to remove the directory on the file system. If there are some files in the children directory of the input path or the input path does not exist, it will warn the user.

```
Button Remove directory clicked!
Please input the directory path you want to delete.
```

/datanode

```
The directory does not exist!
Button List the content clicked!
```

/

```
yelp    yelps   database
Button Remove directory clicked!
Please input the directory path you want to delete.
```

/database

```
The directory has been deleted
Button List the content clicked!
```

/

```
yelp    yelps
Button Remove directory clicked!
Please input the directory path you want to delete.
```

/yelp

```
The dir has files in it, please check: /yelp/user
```

Get Partition Locs

This function will return one of the replica location for each partition of the input file path. It is used for the map-reduce function.

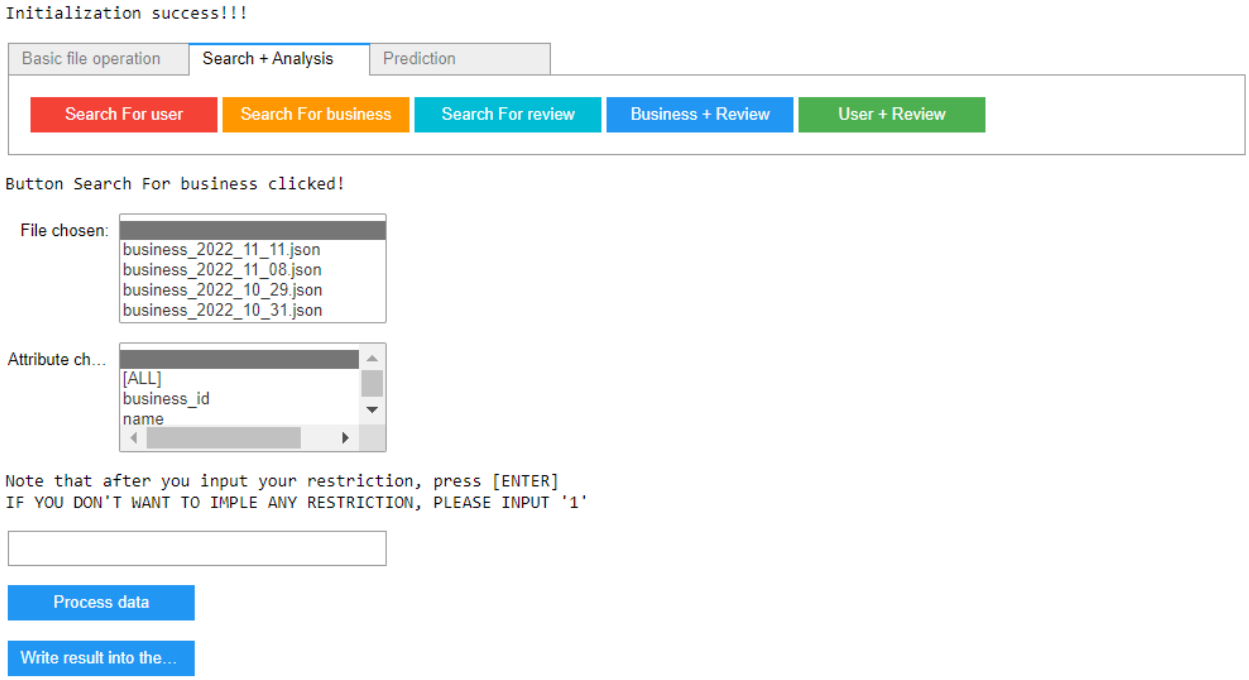
```
Button Get Partition Locs clicked!
```

/yelp/business/business_2022_10_31.json

```
Partition_name: business_2022_10_31_partition0-> DataNode: storage1
Partition_name: business_2022_10_31_partition1-> DataNode: storage0
Partition_name: business_2022_10_31_partition2-> DataNode: storage0
Partition_name: business_2022_10_31_partition3-> DataNode: storage1
Partition_name: business_2022_10_31_partition4-> DataNode: storage0
```

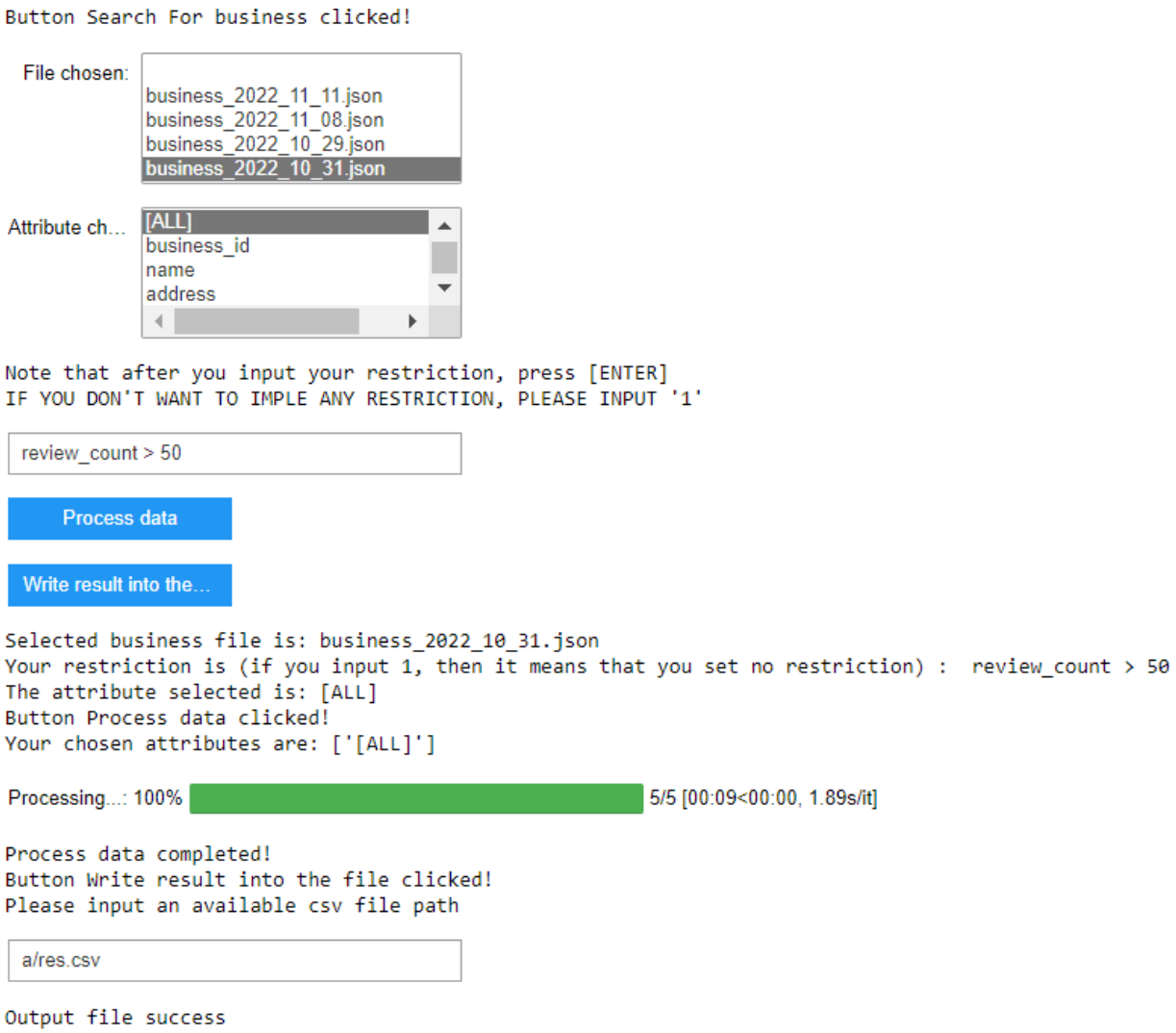
Part2: Search + Analysis (Wu)

Search function will allow the user to choose one of the file and some features about the file with some restrictions. After process the data, they could be written into the files and return to the local machine.



For the attribute, if you select [ALL], the system will return all the attributes of the file. After you input the restrictions, you should PRESS [ENTER]. And if you don't want to choose any attribute, please input "1".

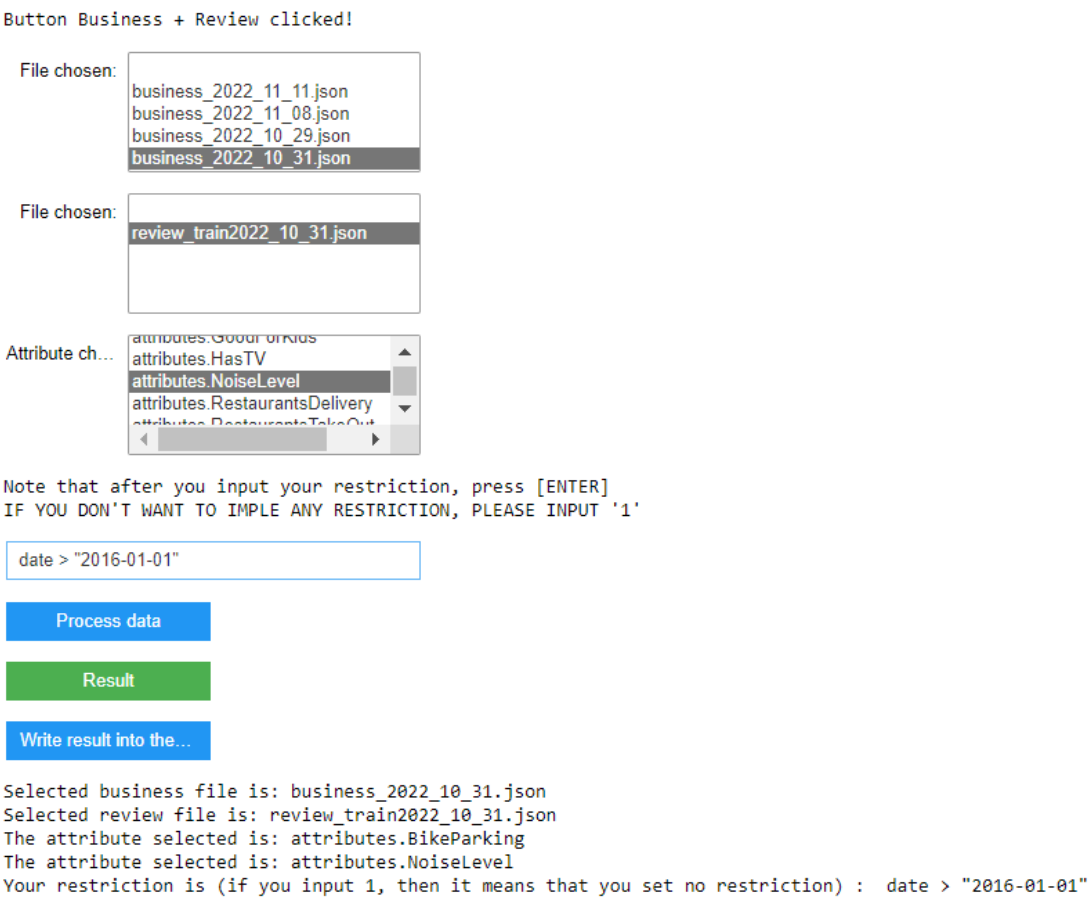
Having input the parameters, you could orderly press "Process the data", and after the Processing press "Write the result into the local machine" button. After input the path on the local, the search result will be downloaded in the local machine.



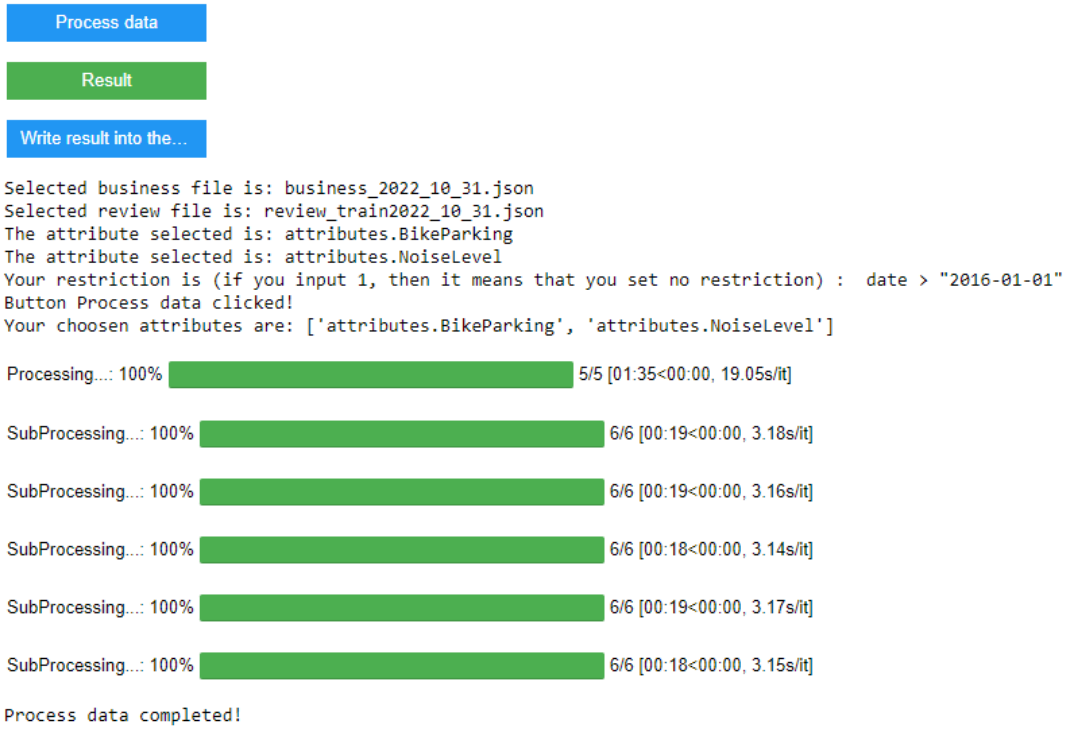
The same are for user and review.

Analysis: Combine of review + business/user

These functions will aggregate review file and business/user file using the attributes you choose as the groupby keys in order to see some relations between the attributes and the “stars”. **Since the dataset is the for recommendation, “stars” and “count of the review” are two metrics we want to focus,** so the target variable are the stars in the review file . You could only choose at most 2 keys for these functions in our api.



After you have chosen all the required parameters, press “Process the data”. It will find the partitions of review file and partitions of business/user file, and map function will apply these parameters to each combination of two partitions. Finally, it will return the pandas Dataframe format result.



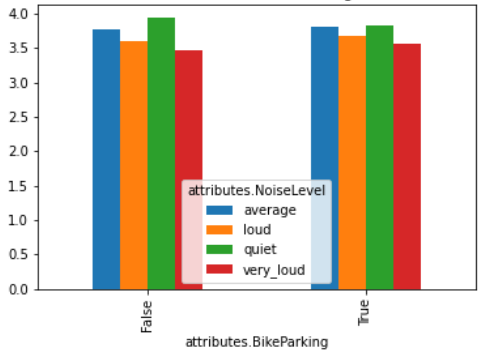
Then, press “Result”, it will show the result and plot the result.

SubProcessing...: 100% 6/6 [00:18<00:00, 3.15s/it]

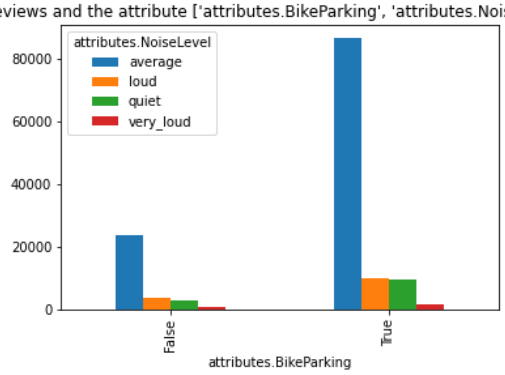
```
Process data completed!
Button Result clicked!
```

attributes.BikeParking	attributes.NoiseLevel	cnt	summation	avg
False	average	23682	89158.0	3.764800
	loud	3539	12780.0	3.611190
	quiet	2869	11291.0	3.935518
	very_loud	730	2538.0	3.476712
True	average	86314	329614.0	3.818778
	loud	9790	36002.0	3.677426
	quiet	9372	35937.0	3.834507
	very_loud	1630	5798.0	3.557055

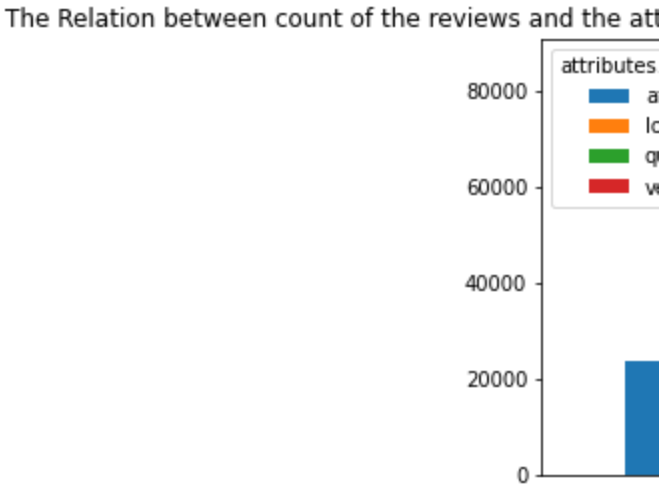
The Relation between average of the stars and the attribute ['attributes.BikeParking', 'attributes.NoiseLevel'] for business under the restrictions

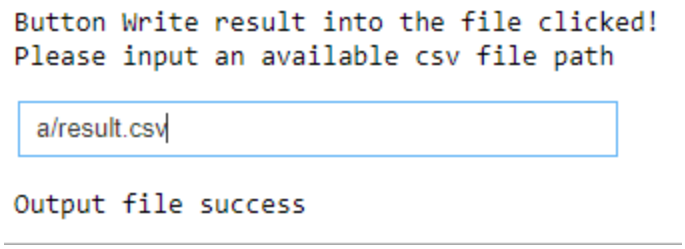


The Relation between count of the reviews and the attribute ['attributes.BikeParking', 'attributes.NoiseLevel'] for business under the restrictions



Finally, if you press “Write result into the local machine”, then it will be downloaded into the local machine on the path you give.





Part3: Prediction (Wang)

As we stated in the guideline of our project, one of our purposes of choosing this dataset is to build a recommender system based on the map and reduce functions we implemented in the distributed file system. Here we basically join three datasets with user id and business id as joining attributes and use data in the Review table as training data to build the regression model as the recommender system. Quantitative variables such as average stars and review counts are selected in the joining process. In the predicting phase, users can just randomly choose a user id and a business id from the user dataset and business dataset respectively, and type into the input box in our UI, the UI also supports datasets selection from our file system. Here are some examples of the predictions.

```
Button Recommender clicked!
```

File chosen: user1.json

File chosen: business6.json

File chosen: review1.json

```
Please input user id and business id split by " "
```

XvLBr-9smbI0m_a7dXtB7w O8S5hYJ1SMc8fA4(

```
Selected user file is: user1.json
Selected business file is: business6.json
Selected review file is: review1.json
[03:05:30] WARNING: /Users/runner/work/xgboost/xgboost/python-package/build/temp.macosx-11.0-arm64-cpython-38/xgboos
t/src/objective/regression_obj.cu:213: reg:linear is now deprecated in favor of reg:squarederror.
The predicted star of business  O8S5hYJ1SMc8fA4QBtVujA  from user  XvLBr-9smbI0m_a7dXtB7w  is  5.5394754
```

This prediction implies that this specific user probably will like this business, and the business is an ideal recommendation.

```
Please input user id and business id split by " "
```

lzlZwIpuSWXEnNS91wxjHw 8USyCYqpScwiNEb

```
Selected user file is: user1.json
Selected business file is: business6.json
Selected review file is: review1.json
[03:07:14] WARNING: /Users/runner/work/xgboost/xgboost/python-package/build/temp.macosx-11.0-arm64-cpython-38/xgboos
t/src/objective/regression_obj.cu:213: reg:linear is now deprecated in favor of reg:squarederror.
The predicted star of business  8USyCYqpScwiNEb58Bt6CA  from user  lzlZwIpuSWXEnNS91wxjHw  is  0.7391845
```

Here is another example where the predicted star is relatively low. And this business will not be a good choice to recommend to this user.

Besides, if the input is some strange strings which can not match with the records in our user and business datasets, the UI will tell the user that the user id or business id does not exist.

Please input user id and business id split by " "

sadsasa sdfssd

user id or business id not exists

Group members learning experience

Yunjie Wu:

In this report, I learn how to use ipywidgets to create UI on the jupyter notebook, pymysql to connect MySQL with python. There are many tricks about ipywidgets, especially the way to asynchronously wait for the result in the last step and after the result has been finished, then run the following programs. If not using asynchronous functions, the program will not wait for the user to input the parameters and just run for the afterward programs, which will just cause the error .I also learn how to deal with the mysql result and convert them into the pandas dataframe.

Yunhe Wang:

During this project, I have a better understanding of the distributed file system, the specific meaning and functions of datanodes and namenodes. I also reviewed the SQL knowledge while implementing the system commands with the SQL emulation. Besides, I also mastered the transformation between the SQL tables and pandas dataframes while implementing the analysis and prediction parts. Some challenges I met were the formats of testing data input of the prediction model, dealing with the outputs of the invoked functions, and the connections between the users' intentions and the internal operations in the file system as well. I also learned how to use pymysql to implement MySQL operations in python language and ipywidgets to better interact with the user of our application.

The links to the project

url:https://drive.google.com/drive/folders/1-ZJ0vFGuvKqOxO-vp7lq-zkMj4tkyNnG

Demo video links

Yunjie Wu: https://www.youtube.com/watch?v=QSaF0YNs5Qc

Yunhe Wang: https://youtu.be/d6A_8hBehq8

Link to the code: YELP_EDFS.ipynb - Colaboratory (google.com)