

**Yelp Dataset**

(Recommendation Engine & Sentiment Analysis)

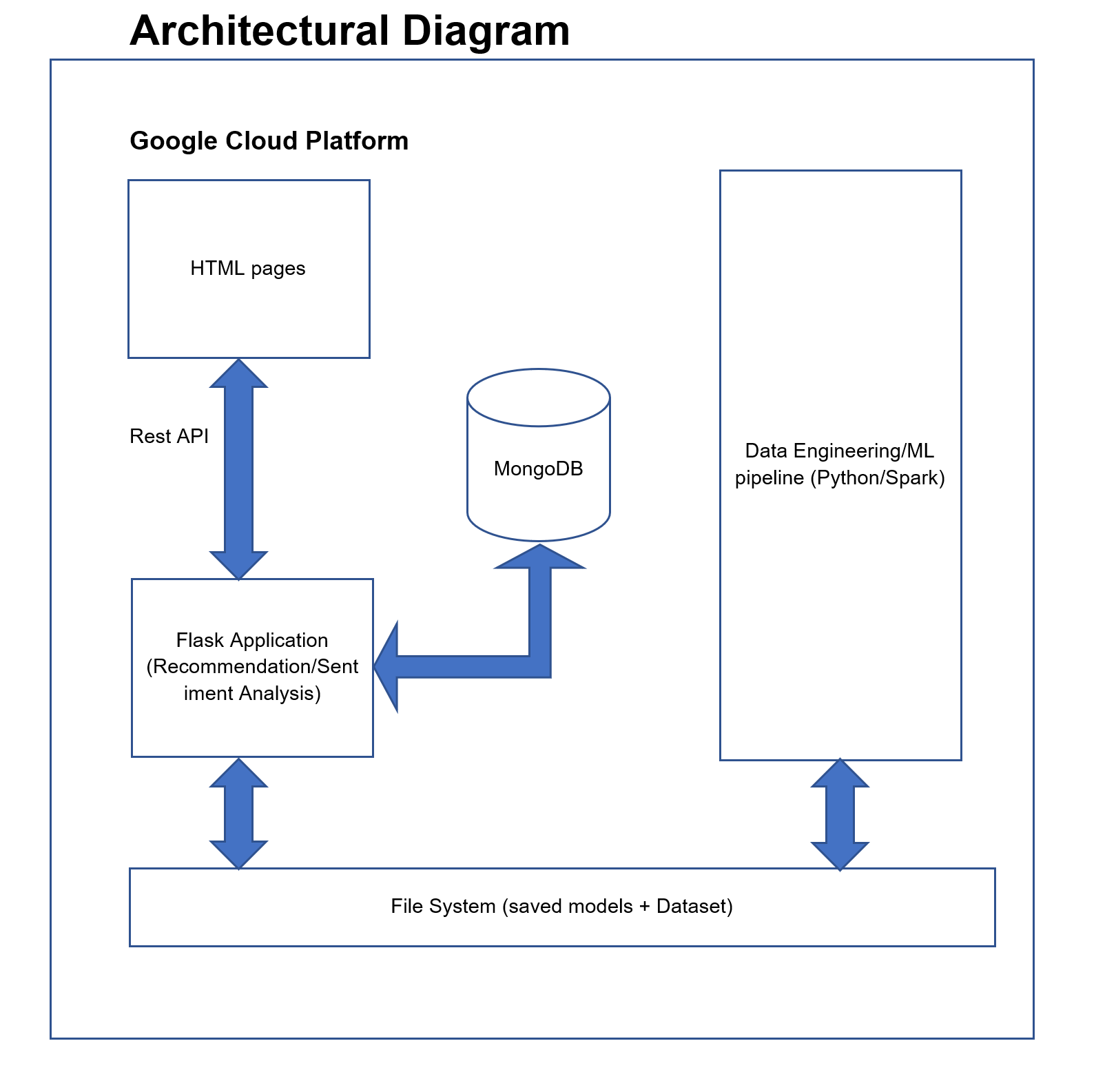
Devesh Kandpal

Soham Dasneogi

**Introduction**

Yelp.com is the go-to website for anybody looking for recommendations about local businesses. Over the period, it has become quite popular for restaurant recommendations as users find it easier to judge a local business by reading reviews about that business posted by other users. In addition to writing reviews and rating local businesses, user can react to reviews, plan events or start up a discussion forum. Yelp has about 135 million monthly customers and 95 million reviews.

**Architecture Diagram**



**Data Download and pre-processing**

Yelp dataset was downloaded from yelp.com/dataset. It consisted of JSON files

business.json

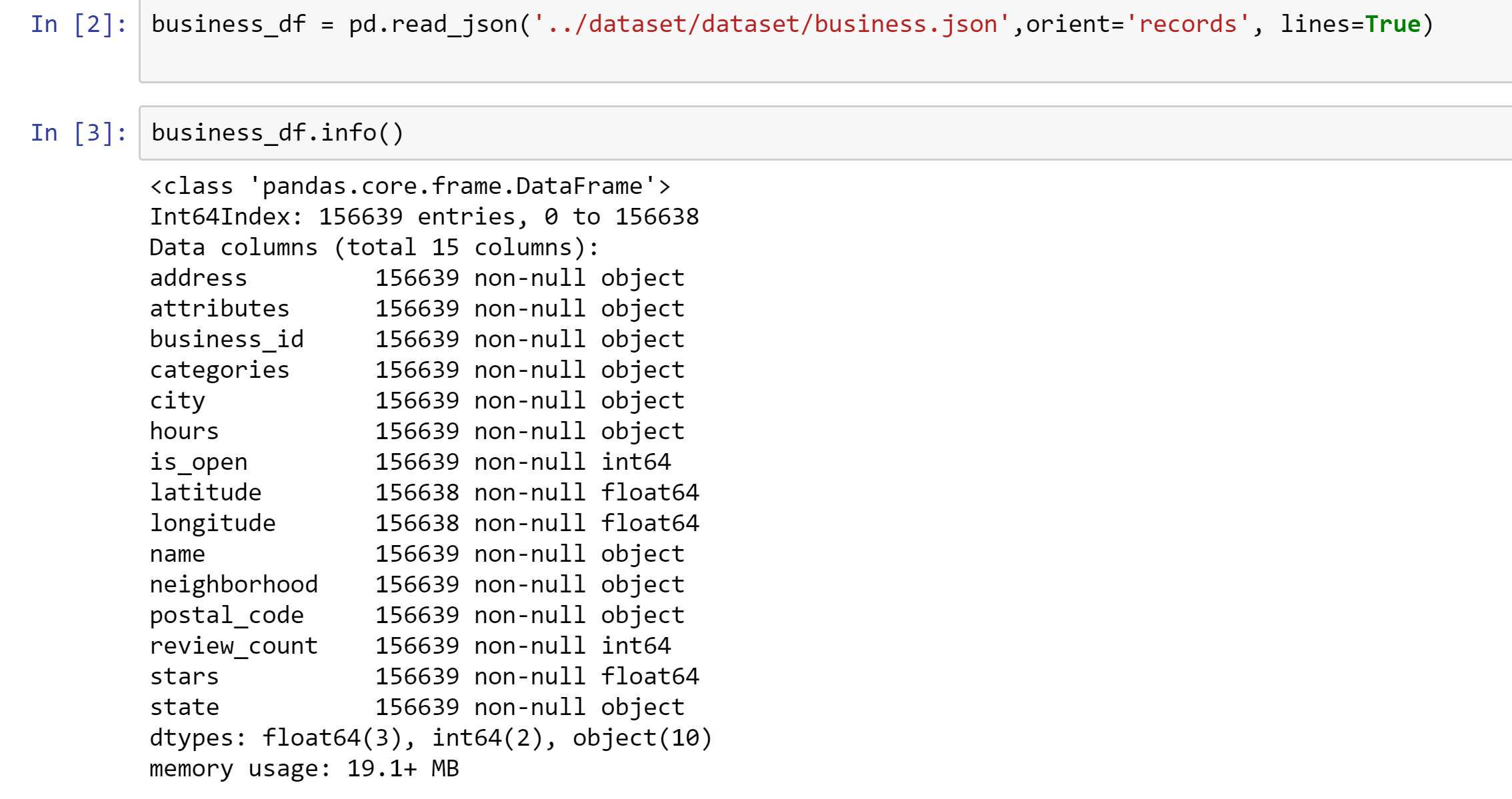
review.json

user.json

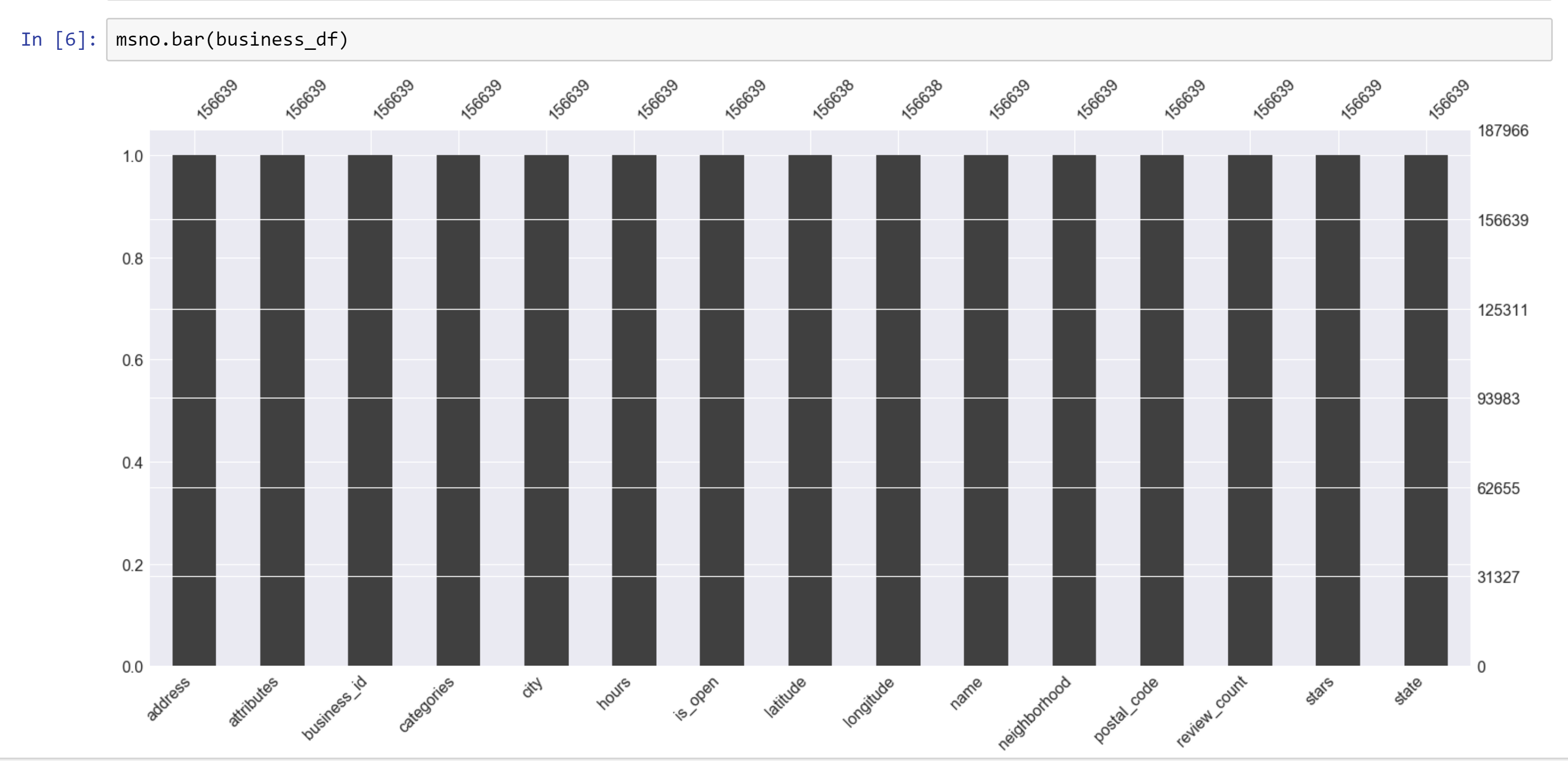
Extracting the json files and reading them into the pandas dataframe

**Exploratory Data Analysis**

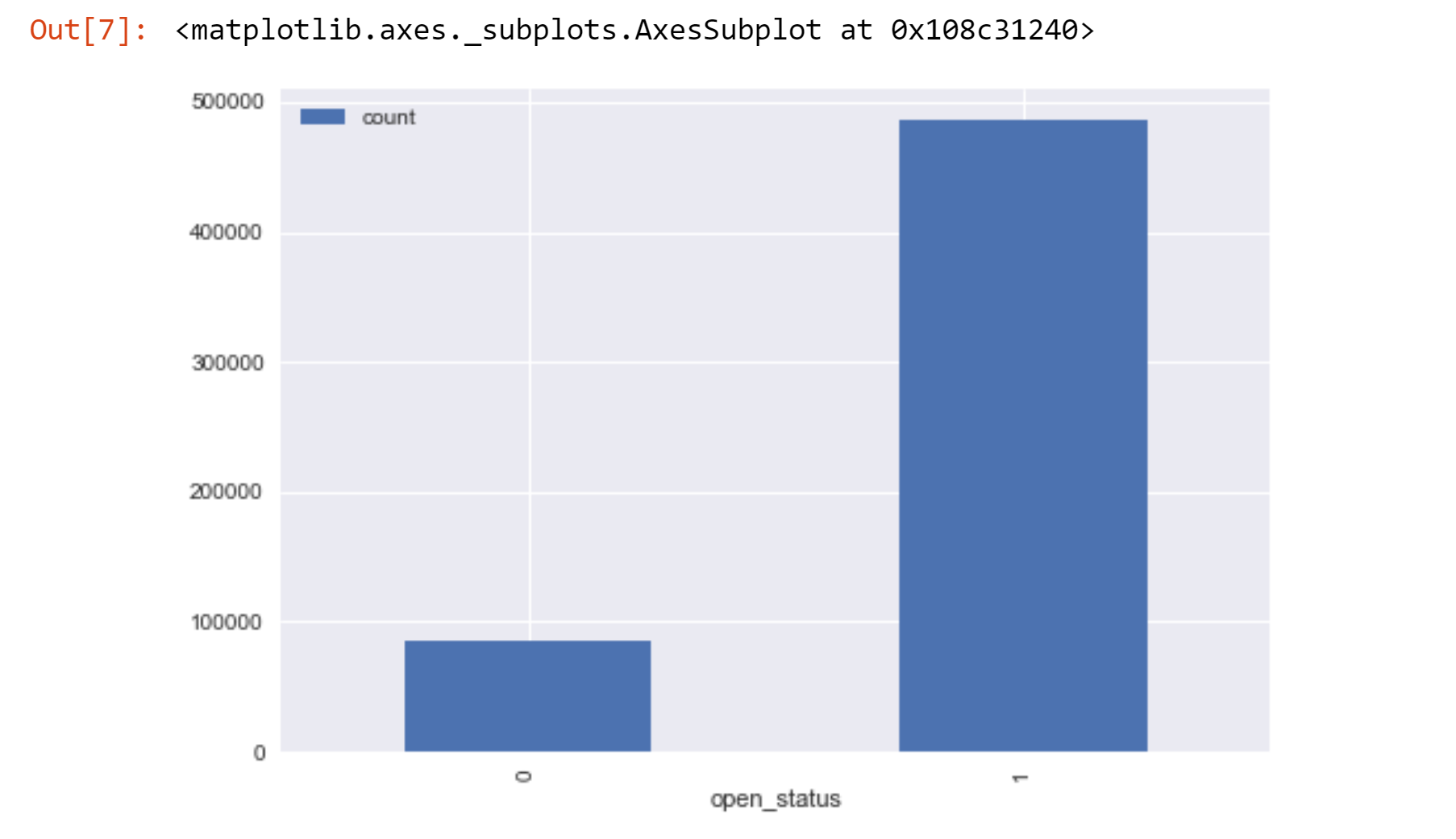
Reading the business.json and checking the type of data that we have there.



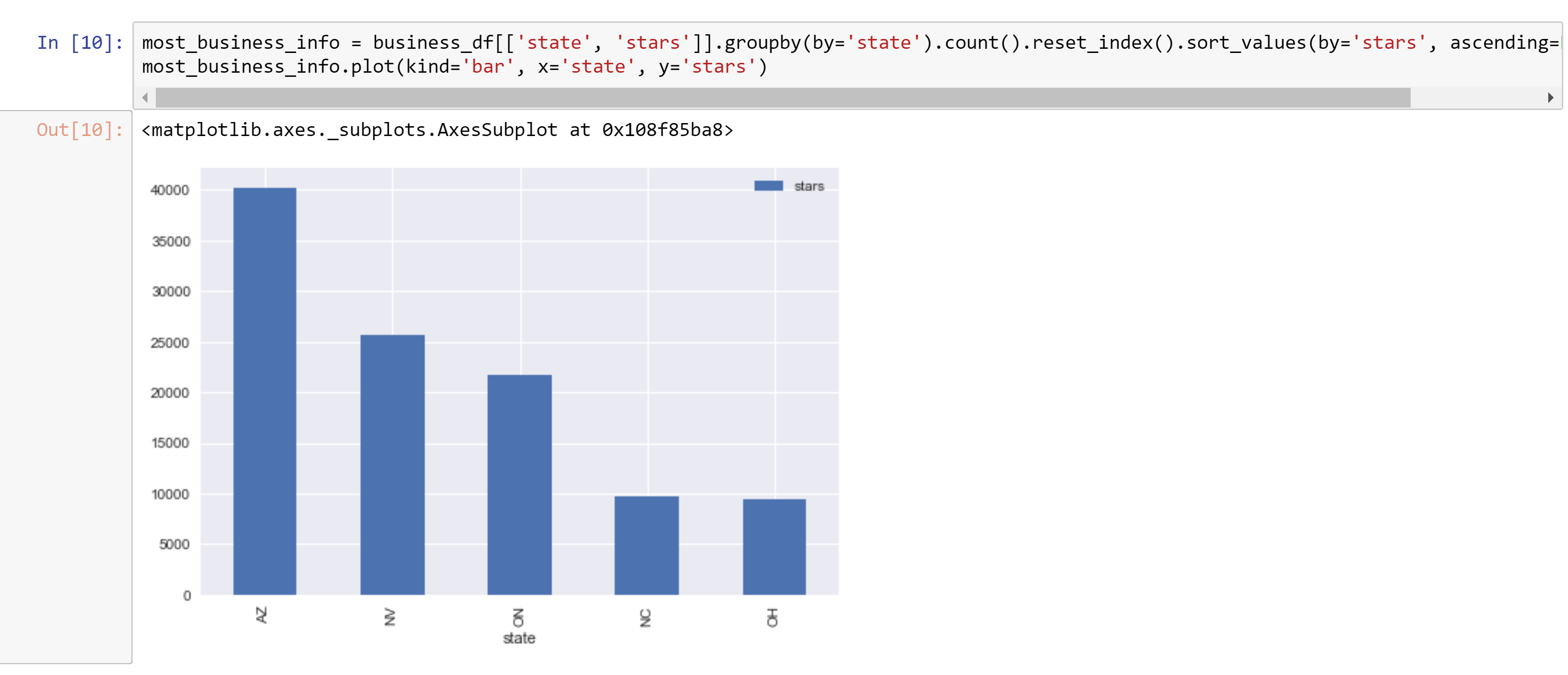
There was no missing data to handle but the file was large, so we had to do a lot of data processing to find out the relevant data to do recommendation on restaurants.



Checking the attribute of the restaurant if its open or not and eliminating the restaurant which is closed



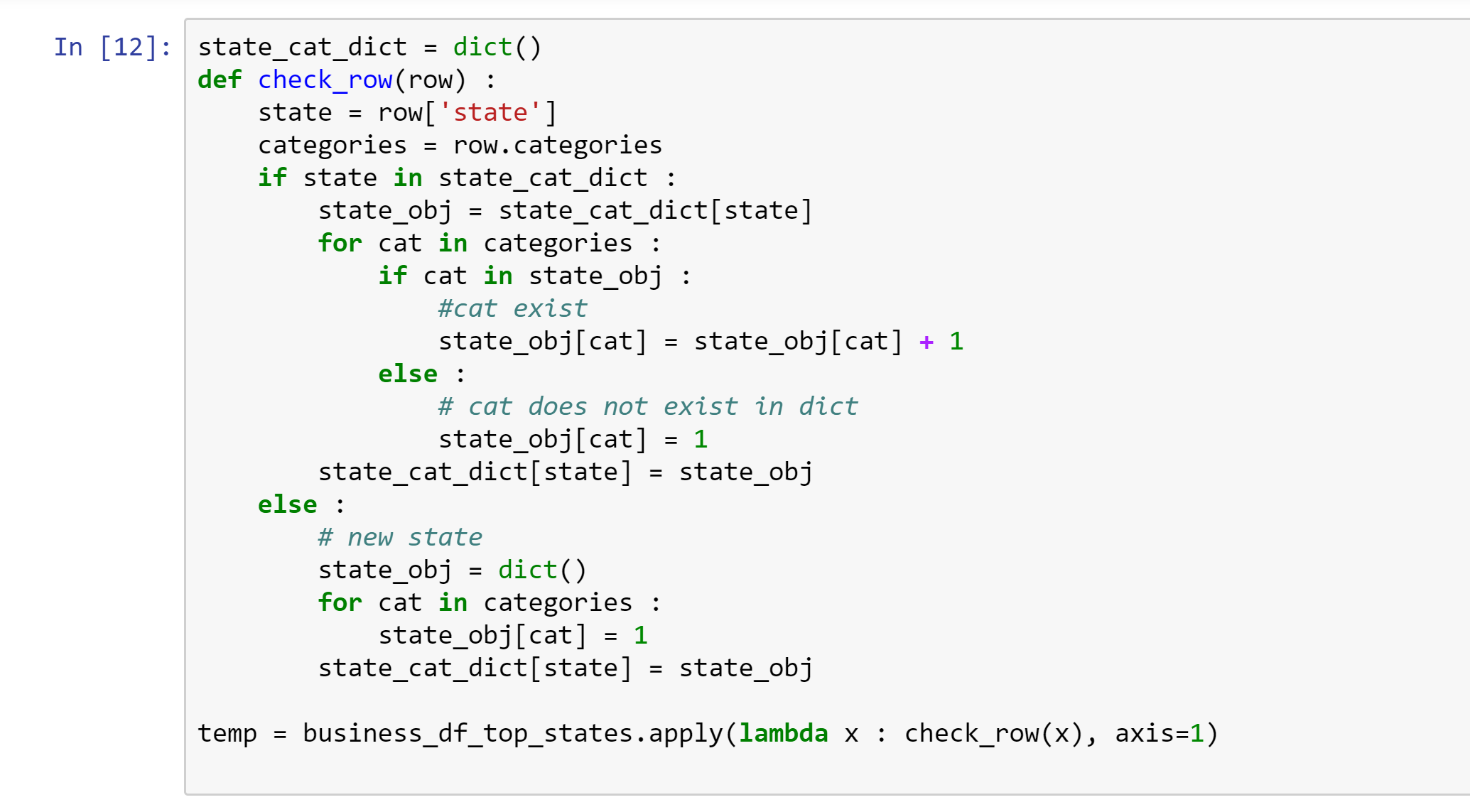
Selecting the top states with most restaurants



We find out that the states

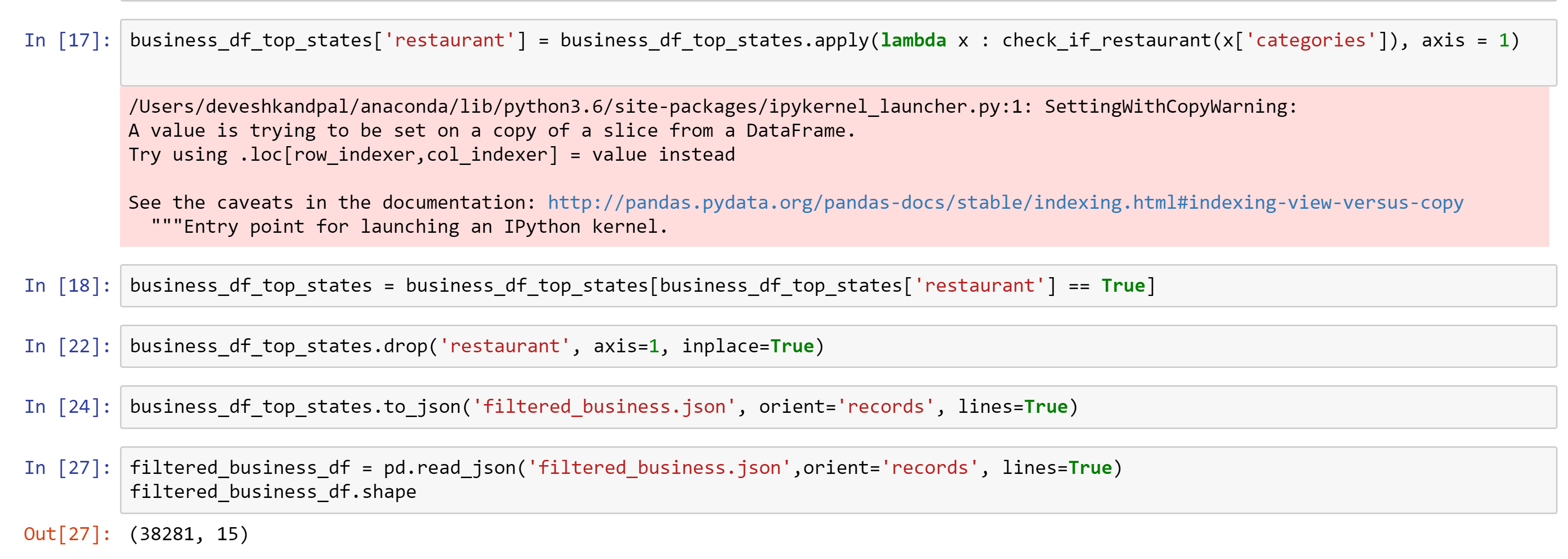
Arizona, Nevada, Ontario, North Carolina and Ohio has the most number of restaurants in the data set so we eliminated the other states as they were significantly low in number

Looking at the categories that the business were segregated into, we cherry picked the categories that were related to restaurants and food





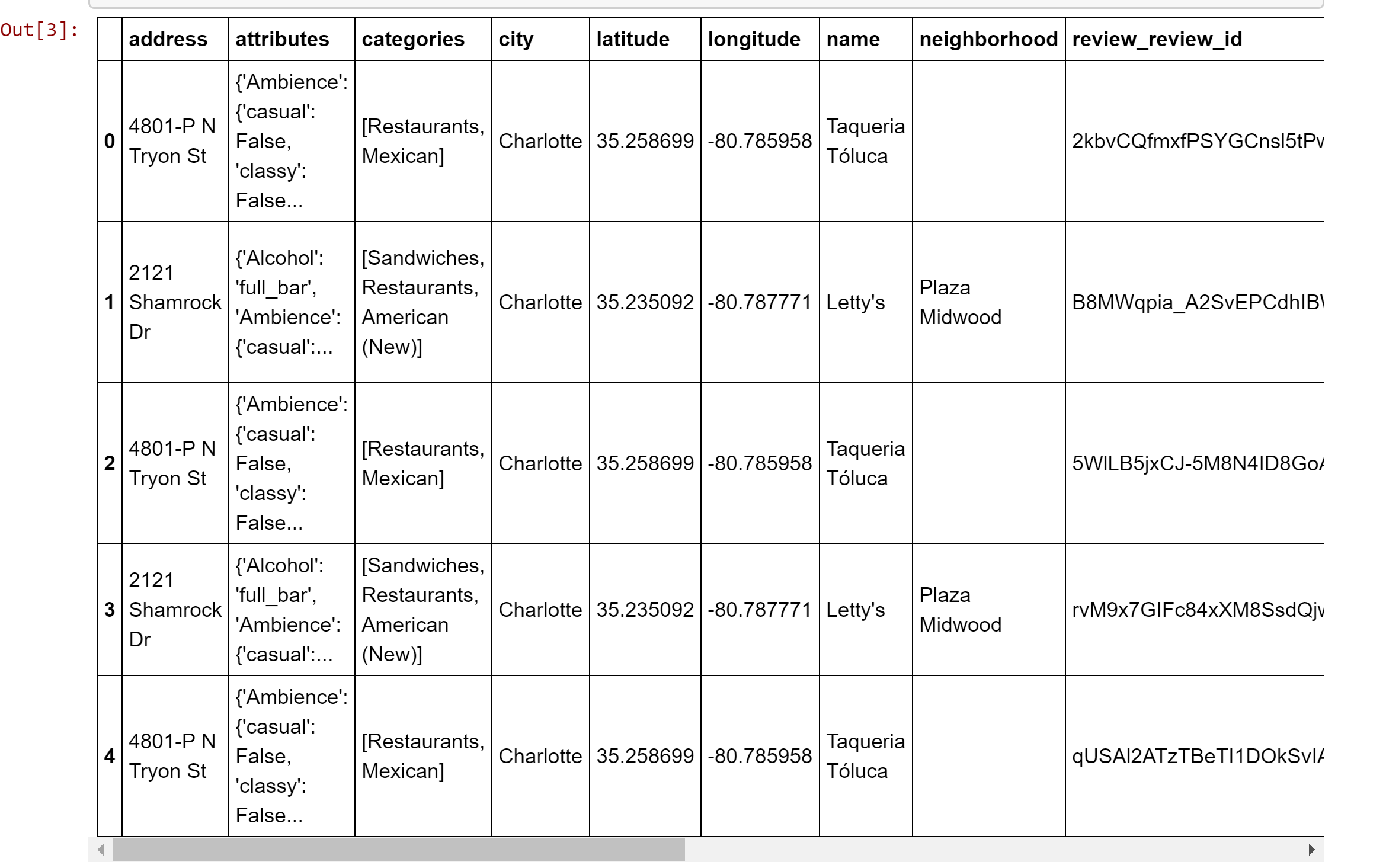
Then we filtered out the data frame and cleaned the data based on the analysis done.



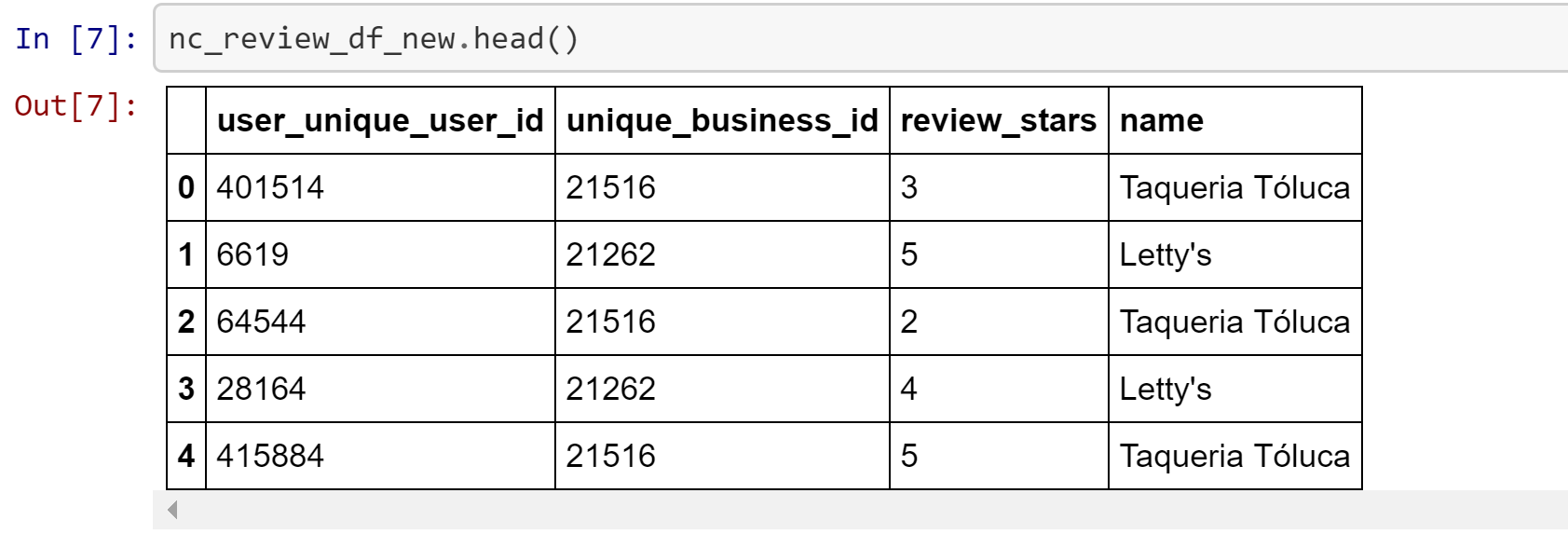
With the help of the filtered business dataset, we selected the reviews based on those businesses and merged the files for computing the results.



Our data frame now looks like



Checking out the data frame for one state NC



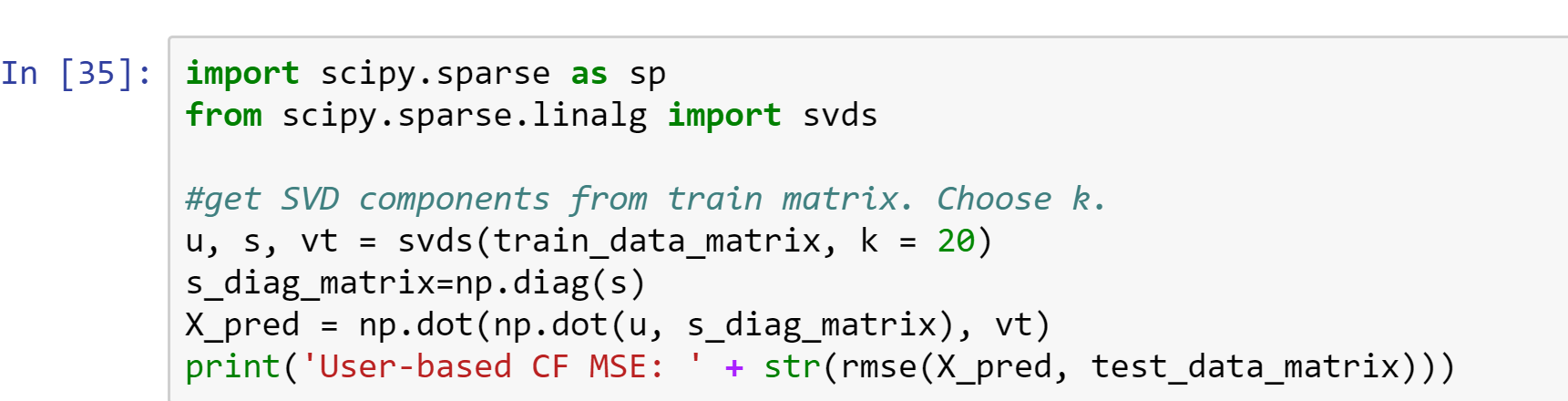
After cleaning and segregating the data and merging appropriate files, we get the data frame which has only the relevant features needed for a recommendation model

**Building and Evaluating Models:**

1. **Recommendation Model**

We tried out various recommendation models for recommending restaurants to a user.

**SVD model**



The RMSE of the model wasn’t as per standards and the model was taking more time that expected to train.

**ALS model**

We used Spark ML to build the ALS Matrix Factorization Model for restaurant recommendation.

We create for different ALS models (one per state) which we pickle and store in the file system after creation.

This model is capable of helping us get restaurant recommendations for an existing user as well recommendation for users when given a restaurant.



Steps

1. For each state, we load the json file containing a combined view of user reviews and restaurants
2. We select user\_id, business\_id, and rating and do a 80:20 random split on the data
3. The training data is used to train the ALS model
4. The tesing dats is used to compute the RMSE
5. We get and **RMSE score of 1.3**
6. The models are pickled and saved so that it can be consumed by the flask application.
7. Sentiment Analysis

We are doing two things here

For each restaurant in each state, we are analyzing each review and calculating the sentiment polarizing score using a library called textblob.

Secondly, for each restaurant we are doing topic modelling using LDA by creating a single document of reviews per restaurant. We do this for all restaurants across all the states.

The goal of topic modelling is to identify good and bad reviews.





Steps

1. Combine all reviews of a single restaurant into a single document
2. The entire document is tokenized and cleaned using stop words provided in the gensim library.
3. We then used textblob on this cleaned text to determine sentiment polarity
4. We then used gensim library for first creating a dictionary followed by corpus by using the cleaned text
5. Once the corpus and dictionary are created, we used an unsupervised learning technique called LDA for topic modelling
6. We repeat this process for all the restaurants across all the states
7. We then use LDA model to identify the top topics in the corpus and store the top 15 tokens in each topic in a dictionary which is then dumped into mongo db along with the calculated polarity of the reviews and other details

