Data and Programming Analytics

Yelp Review Sentiment Analysis

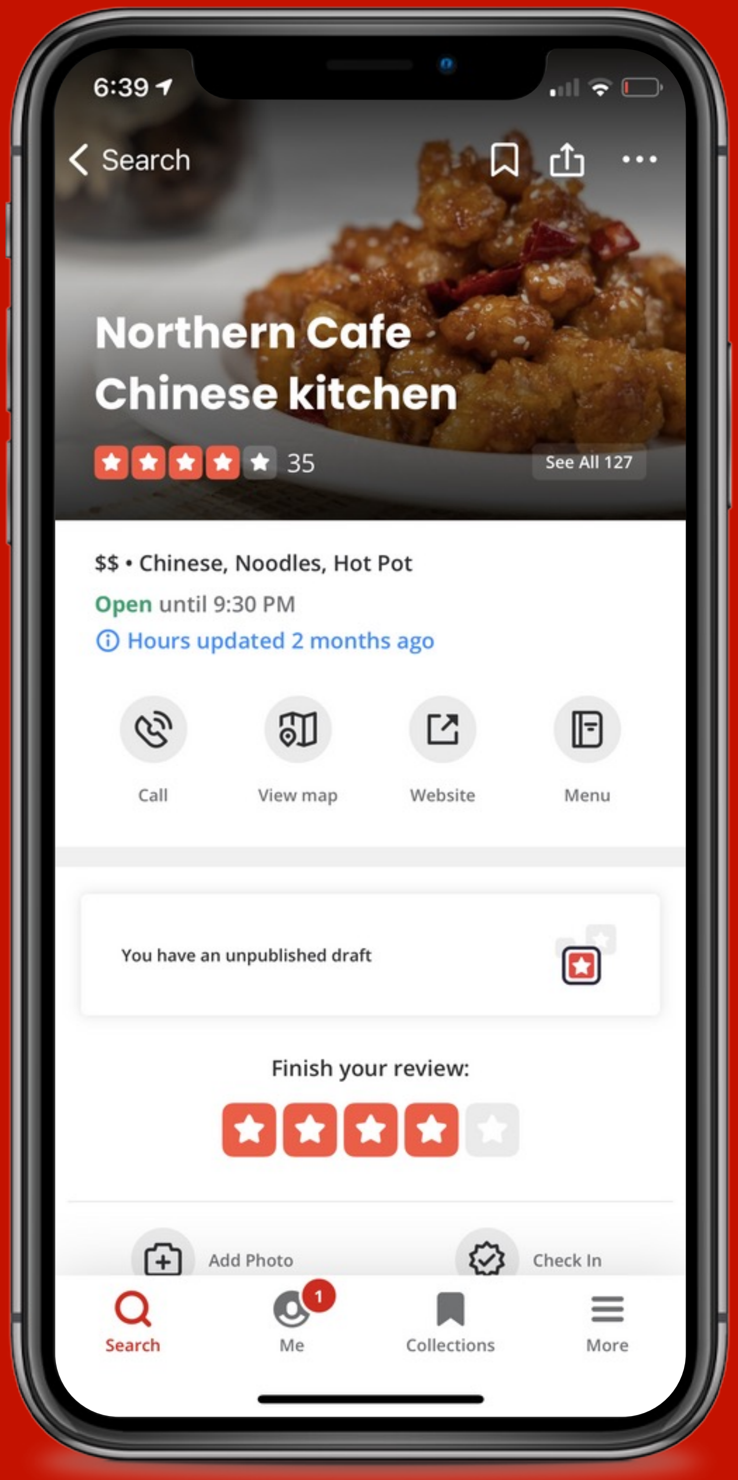
Group 18

Yaohui Wu, Jai Agrawal, Jennie Ryu, Xiaofan Zhu, Sahithi Muddana

Table of Content

2021 Q2 revenue is more than

$257M



Yelp rating can increase
a business owner's sales by

5-9%

PROJECT TASK

Topic modeling (Positive)

Topic modeling to extract the reasons why people give positive comments to business.

Topic modeling (Negative)

Topic modeling to extract the reasons why people give negative comments to business.

Topic modeling (Neutral)

Sentiment Analysis and topic modeling extract actual negative and positive comments from neutral comments

DATA INTRODUCTION

business.json

- business_id
- stars
- city
- state
- review_count.

Review.json

- review_id
- business_id
- stars
- useful
- etc
- Dataset size: 8.6 millons

```
with open('drive/MyDrive/Colab Notebooks/yelp_dataset/yelp_academic_dataset_business.json') as f:
    for i in f:
        line = eval(i.replace('null','"null"',100))
        if count == 0:
            if 'business_id' not in business_info.keys():
                business_info['business_id'] = [line.get('business_id')]
            else:
                business_info['business_id'].append(line.get('business_id'))
            if 'name' not in business_info.keys():
                business_info['name'] = [line.get('name')]
            else:
                business_info['name'].append(line.get('name'))
            if 'address' not in business_info.keys():
                business_info['address'] = [line.get('address')]
            else:
                business_info['address'].append(line.get('address'))
            if 'city' not in business_info.keys():
                business_info['city'] = [line.get('city')]
            else:
                business_info['city'].append(line.get('city'))
            if 'state' not in business_info.keys():
                business_info['state'] = [line.get('state')]
            else:
                business_info['state'].append(line.get('state'))
            if 'postal_code' not in business_info.keys():
                business_info['postal_code'] = [line.get('postal_code')]
            else:
                business_info['postal_code'].append(line.get('postal_code'))
            if 'stars' not in business_info.keys():
                business_info['stars'] = [line.get('stars')]
            else:
                business_info['stars'].append(line.get('stars'))
            if 'review_count' not in business_info.keys():
                business_info['review_count'] = [line.get('review_count')]
            else:
                business_info['review_count'].append(line.get('review_count'))
```

RESERVOIR SAMPLING ALGORITHM

```
#reservoir sampling algorithm code
def reservoir_sampling(sampled_num, total_num):
    pool = []
    for i in range(0, total_num):#i is from 0 to 100000-1
        if i < sampled_num:
            pool.append(i)
        else:
            r = random.randint(0, i)
            if r < sampled_num:
                pool[r] = i
    return pool
```

1. Reservoir Sampling Method
2. Randomly pick 100,000 value from 0 to 8,635,403
3. Use as Index

RESERVOIR SAMPLING ALGORITHM

```
#pool is a 100000 length list contains value from 0 to 8635402
import datetime
start = datetime.datetime.now()
if 8665403 in pool:
    print(’a’)
end = datetime.datetime.now()
print(end-start)
```

```
0:00:00.011099
```

```
pool_hash = set(pool)
#pool_hash is a 100000 length list contains value from 0 to 8635402
import datetime
start = datetime.datetime.now()
if 8665403 in pool_hash:
    print(’a’)
end = datetime.datetime.now()
print(end-start)
```

```
0:00:00.000104
```

```
print(0.011099/0.000104)
```

```
106.72115384615385
```

20 Hours

Hashing

20 Minutes

DATA VISUALIZAION

```
1 df.info()
```

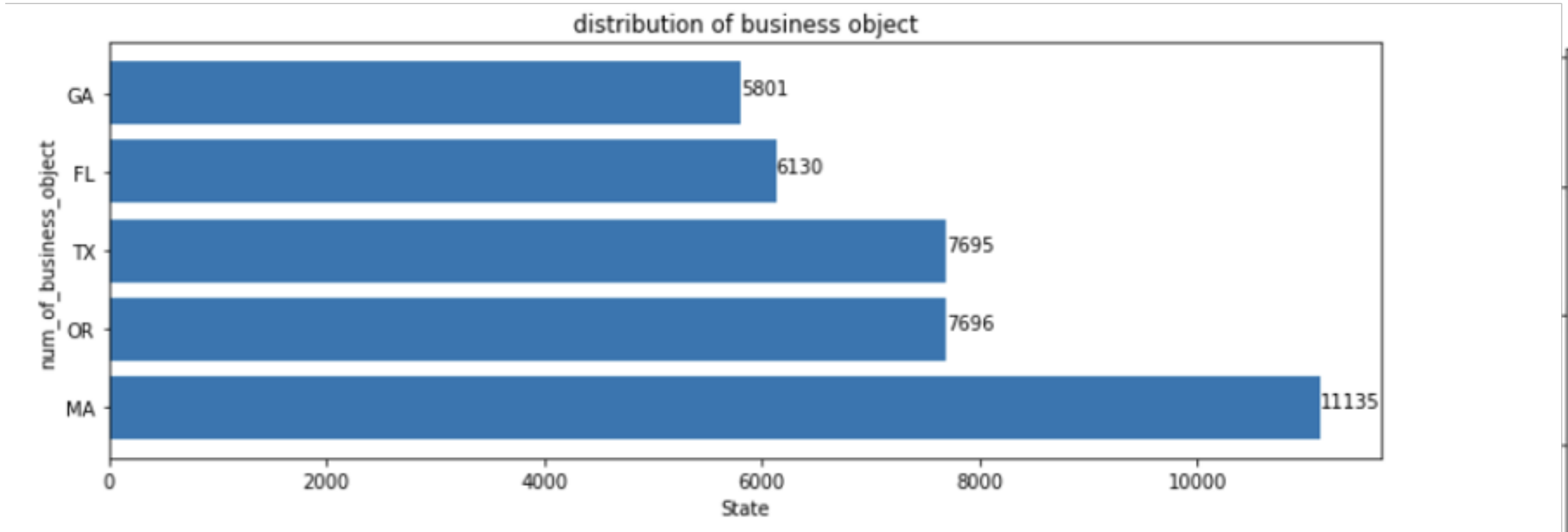
```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 100000 entries, 0 to 99999
Data columns (total 16 columns):
 #   Column        Non-Null Count   Dtype
---  ------        --------------   -----
 0   review_id     100000 non-null  object
 1   user_id       100000 non-null  object
 2   business_id   100000 non-null  object
 3   stars         100000 non-null  float64
 4   useful        100000 non-null  int64
 5   funny         100000 non-null  int64
 6   cool          100000 non-null  int64
 7   text          100000 non-null  object
 8   date          100000 non-null  object
 9   name          100000 non-null  object
 10  address       100000 non-null  object
 11  city          100000 non-null  object
 12  state         100000 non-null  object
 13  postal_code   100000 non-null  object
 14  avg_stars     100000 non-null  float64
 15  review_count  100000 non-null  int64
dtypes: float64(2), int64(4), object(10)
memory usage: 13.0+ MB
```

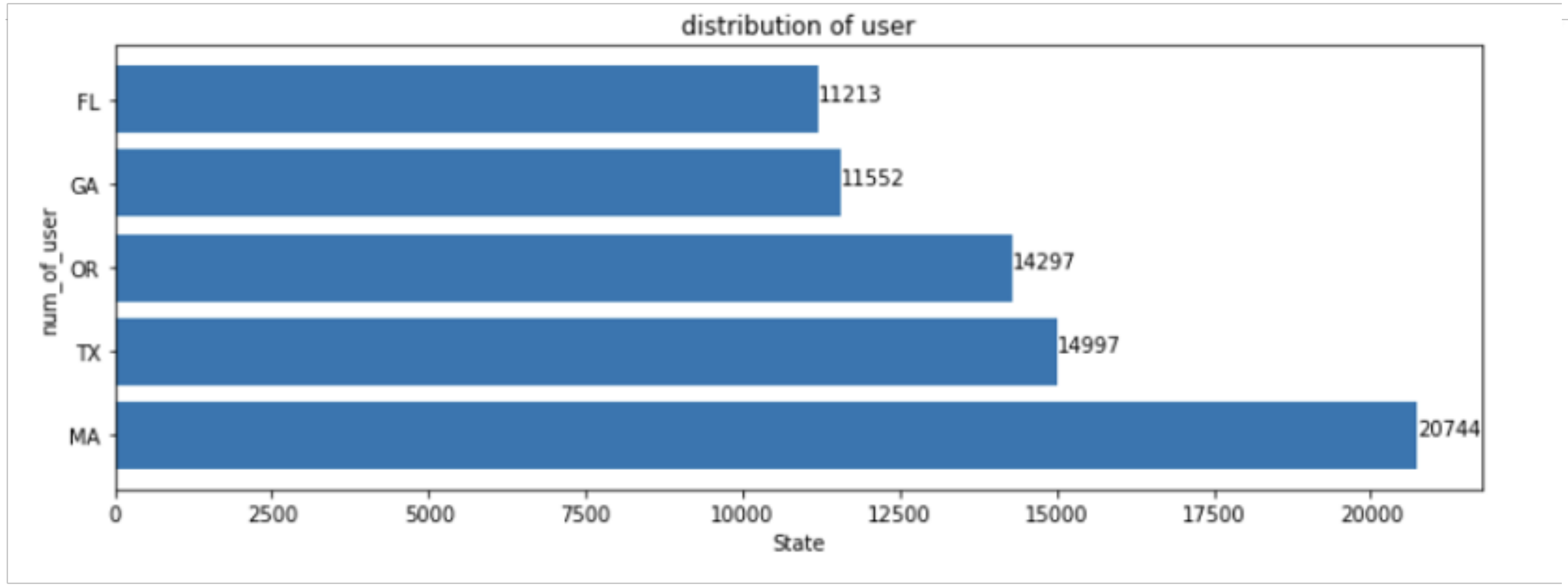
100,000 Rows

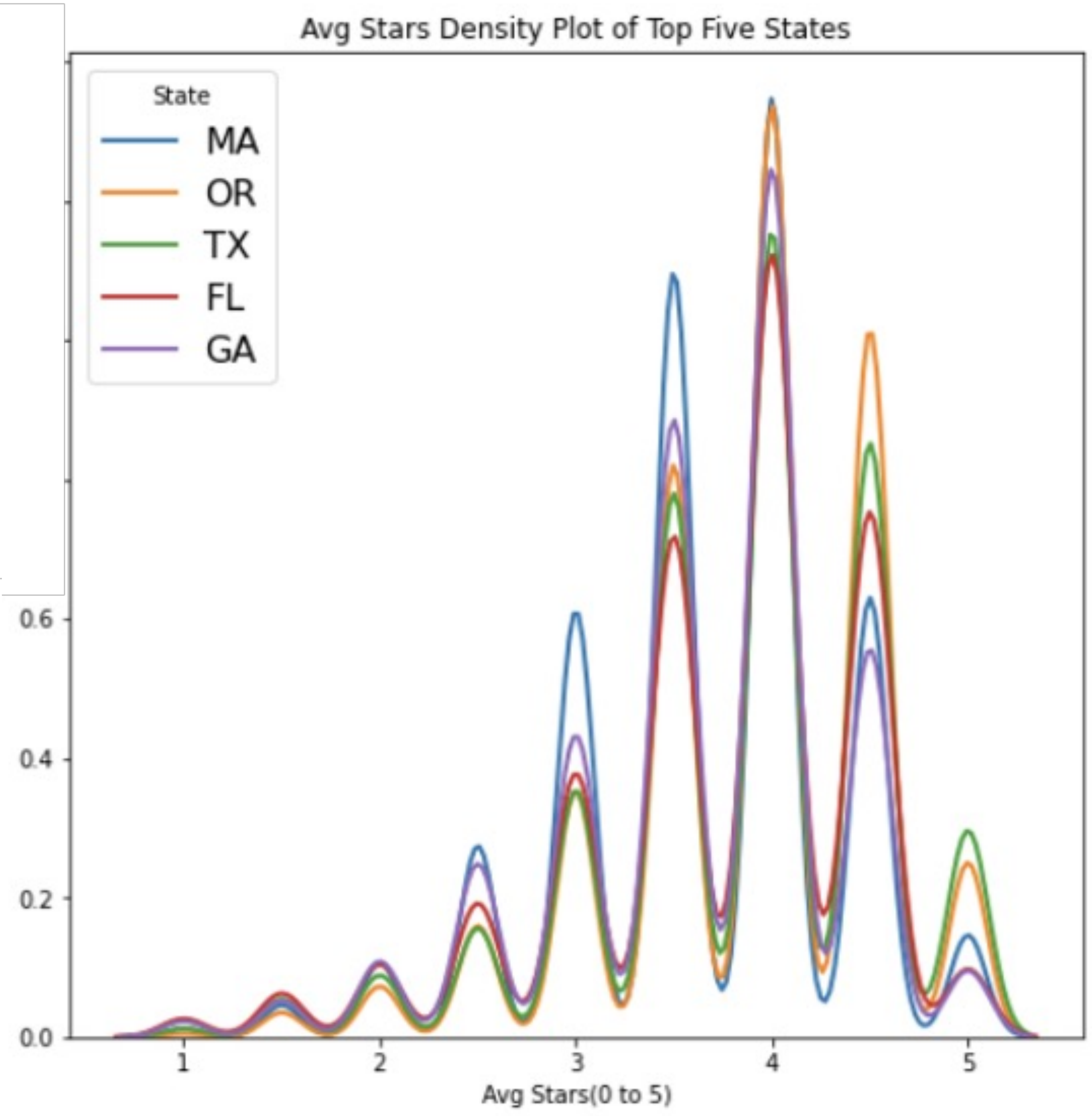
16 Columns

0 N//A Values

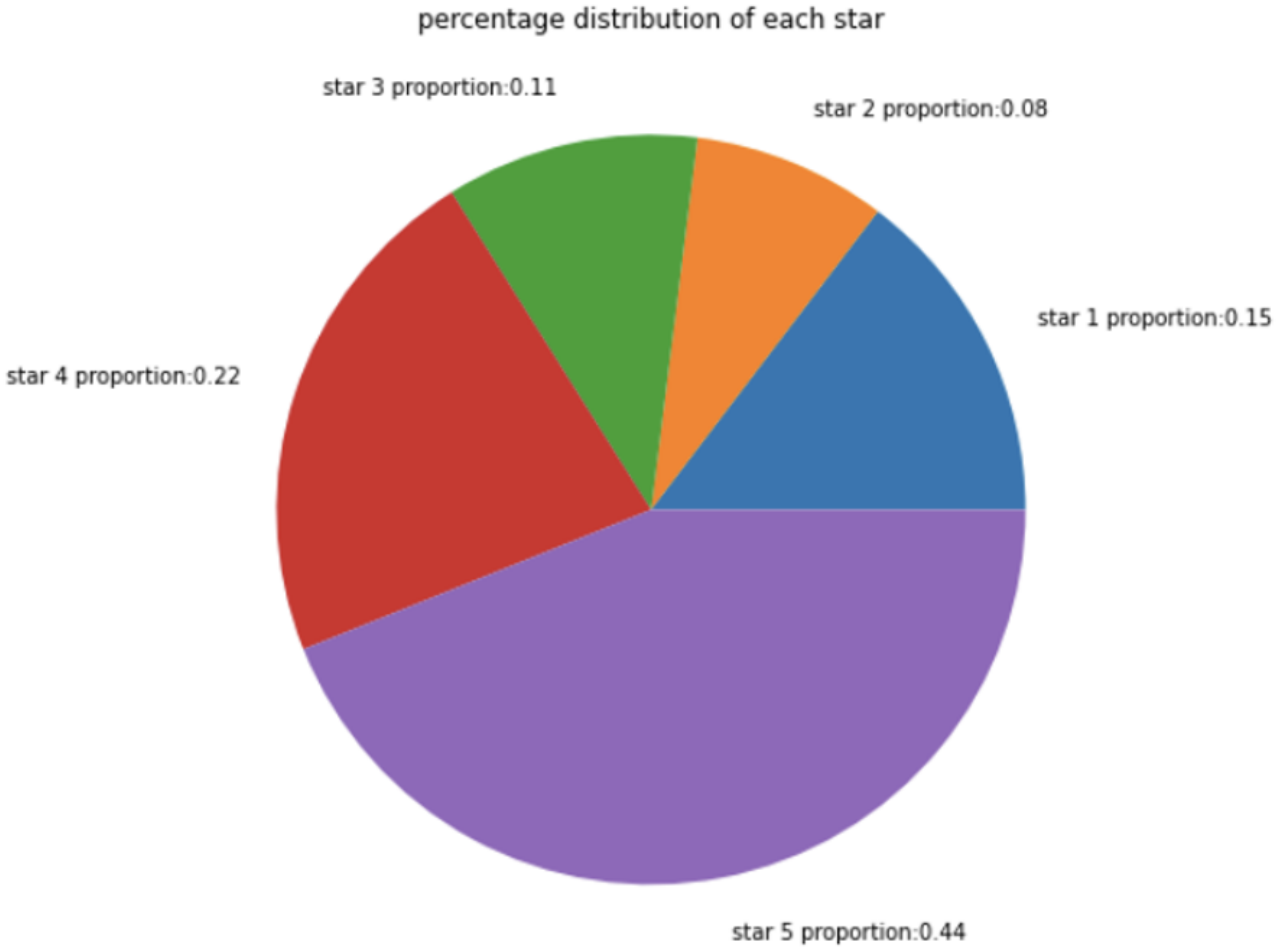
GA(Georgia),FL(Florida),TX(Texas),OR(Oregon),MA(Massachusetts) are the top five states

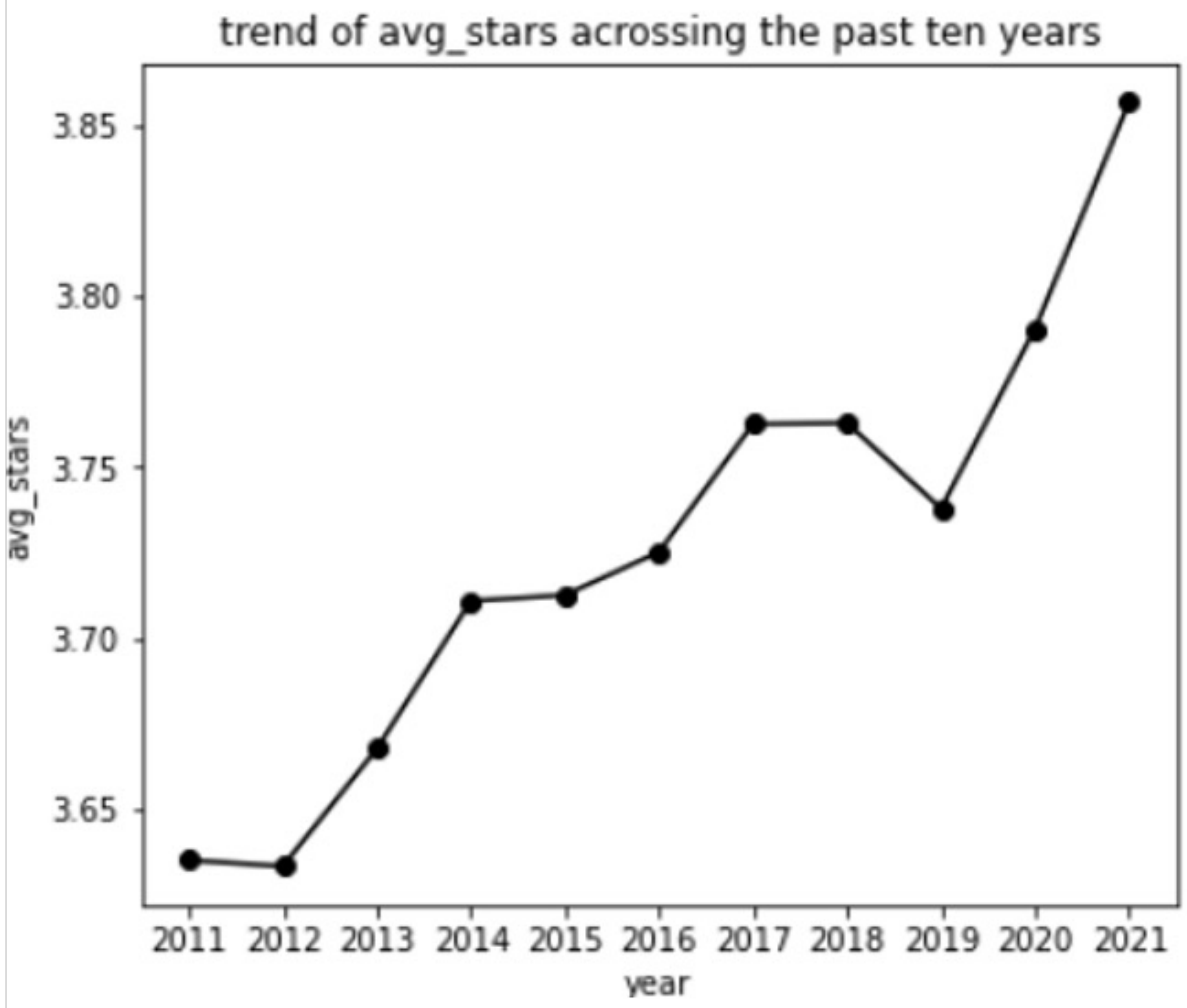






positive(4 or 5 stars) comments takes 66%, negative(1 or 2 stars) take 23%, neutral(3 stars) takes 11%





DATA CLEANING

Remove all of the reviews in foreign languages

```
807      ディビジョンのローカルなフローズンヨーグルト屋さん。自前のキャラも作って、お店も広くて清潔。...
1075     Solid modern Shanghainese food. Quality ingred...
2218     店員のおねーちゃんがタイ人かな？とっても笑顔が素敵でした。お水もなくなったらすぐ入れに来てく...
2627     一月十四日我們四人叫了菜，其中一道油豆腐已經壞了，有告知服務人員，他說另外給我們一份油豆腐，...
3687     20180504 after my interview, I went downtown V...
                                    ...
92533    没想到吃火锅会食物中毒！ 我和先生前天晚上10月11号晚5点后到达火锅店，以前也来过这家两次...
93090    Restaurant was rather packed and required some...
94282    小さな屋台である。希望を聞いて巻いてくれるが、スパムと野菜を巻いてもらって美味しかった。ソウ...
97901    We really like their fish bbq 烤鱼 and also real...
99499    Among all, Kung Fu (功夫茶 is probably the sounde...
Name: text, Length: 64, dtype: object
```

DATA PRE-PROCESSING

- Make all text "lower case"
- Remove "stopwords" from text
- "Amplify" the voice of each user’s comment

SENTIMENT ANALYSIS

NEGATIVE

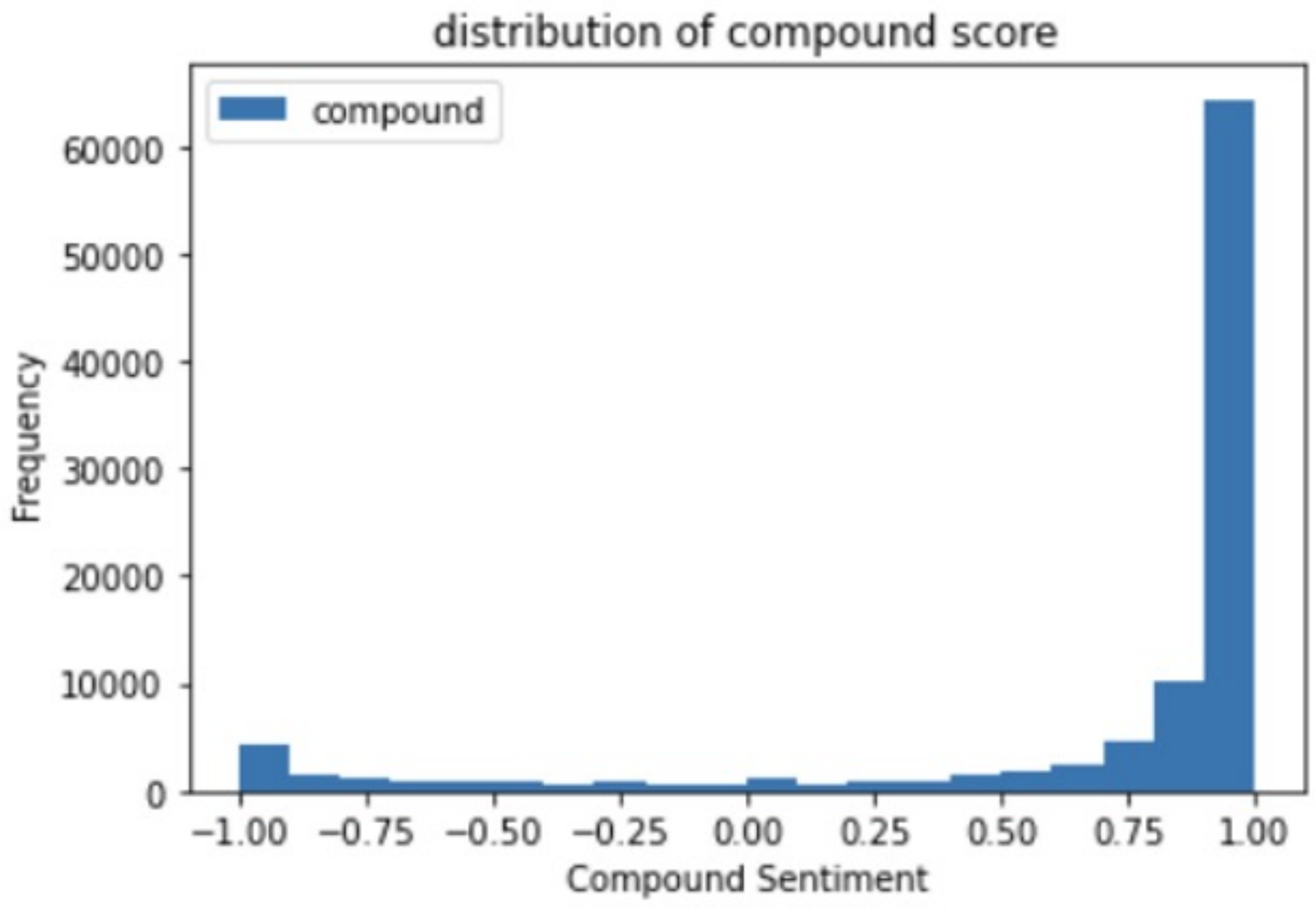
1~2 stars

NEUTRAL

POSITIVE

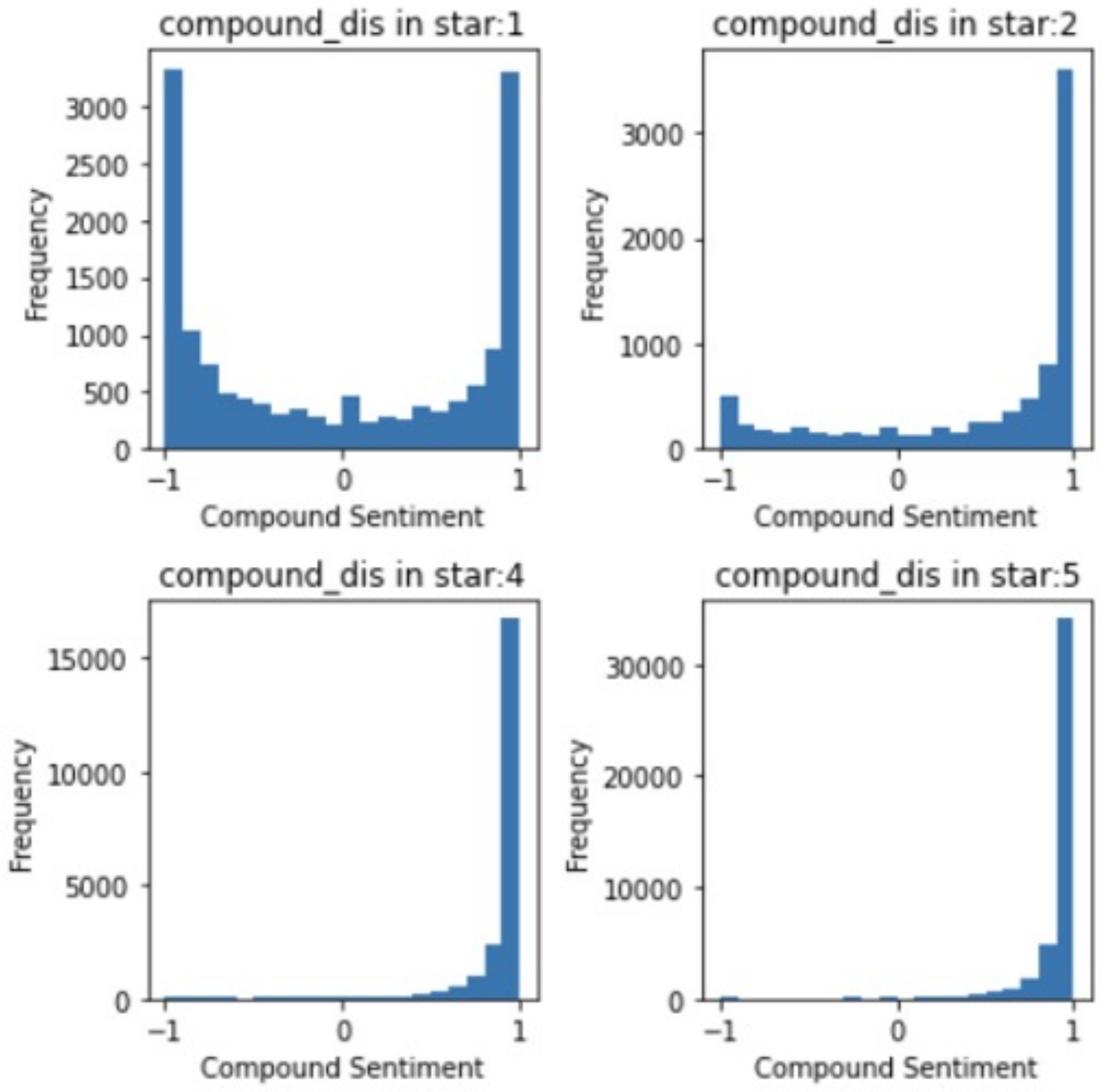
4~5 stars

SENTIMENT ANALYSIS



- Average stars of the dataset is around 3.5
- Compound score is skewed to positive 1

SENTIMENT ANALYSIS



- Distribution of compound score change when star increases
- Compound score moved to positive 1

TOPIC MODELING

Model Building

Create our own stopword package

- Stopword(English) From spacy.lang.en.stop_words: 326
- Stopword(English) From NLTK: 179
- Manually created stopwords: 20
- Our own stopword: 525

```
my_stopwords = ['good','great','bad','recommend','like','pizza','chicken','horrible','better','worst','worse',
                'really','best','amazing','excellent','really','nice','love','highly','pretty']
# stopword from spacy
from spacy.lang.en.stop_words import STOP_WORDS as en_stop
# stopword from nltk
stopword_forTfidfVectorizer = list(stopwords.words('english')) + list(en_stop) + my_stopwords
```

TOPIC MODELING

```
vectorizer = TfidfVectorizer(stop_words=stopword_forTfidfVectorizer,
                             #stop_words='english',
                             max_features= 500 # keep top 1000 terms
                             ,max_df = 0.35
                             ,min_df = 0.05
                             )
```

(A, B)	C
(0, 125)	0.3483511471529684
(0, 112)	0.358021509022570 4
(0, 32)	0.3984352082029299
(0, 34)	0.4389343533429925
(0, 24)	0.4091533930501054
(0, 78)	0.4813020041893511
(1, 111)	0.2065173166552533 3
(1, 110)	0.3498094344052366
(1, 9)	0.2609792801954902
(1, 3)	0.3348766806511680 5
(1, 30)	0.3608773291313086
(1, 88)	0.3100336163837944 5
(1, 107)	0.3375006685332691
(1, 47)	0.3316450335079065

TfidfVectorizer

- A : Index for Comment
- B : Index for Keyword
- C : Level of Importance

TOPIC MODELING

```
svd_model = TruncatedSVD(n_components=topicsNum, algorithm='randomized', n_iter=100, random_state=122)
```

```
components_ output shape (10, 133)
[[ 0.05375653  0.05193305  0.04573669 ...  0.0534104   0.05281103
   0.05644966]
 [-0.01670479 -0.00139009 -0.00413807 ...  0.01598738 -0.01965294
  -0.04217895]
 [ 0.02186757  0.03559468 -0.01549099 ...  0.02746867 -0.0054664
   0.00959735]
 ...
 [ 0.01222876 -0.01604516 -0.04206287 ...  0.03657012  0.05360057
   0.04330051]
 [ 0.00607374  0.00245995  0.01883561 ...  0.03427292 -0.03636514
  -0.00046736]
 [ 0.01498355  0.03766232 -0.00737106 ...  0.04572849  0.01750786
   0.02769219]]
```

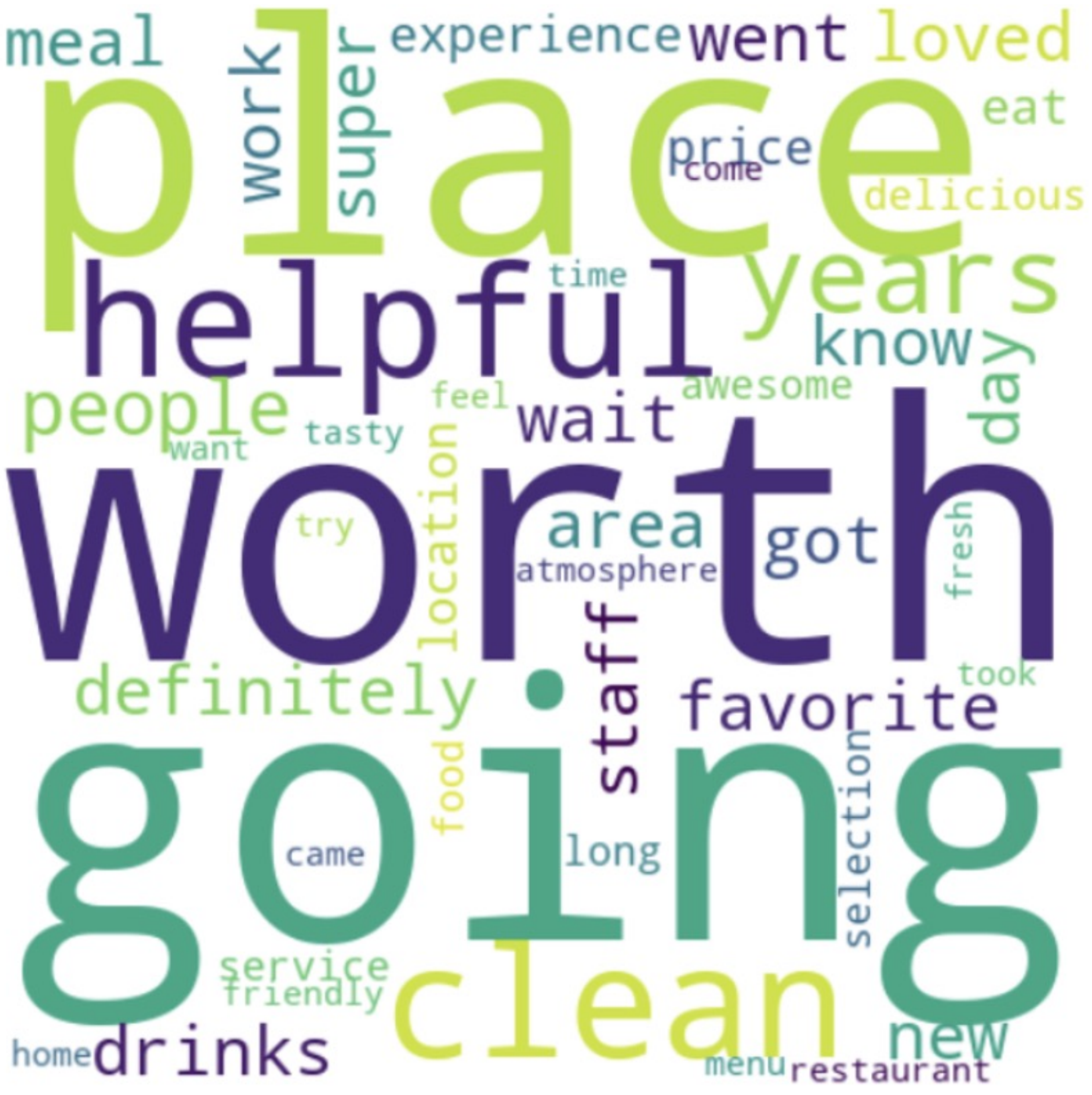
TruncatedSVD

- Row : Topic
- Value : Importance of the keyword

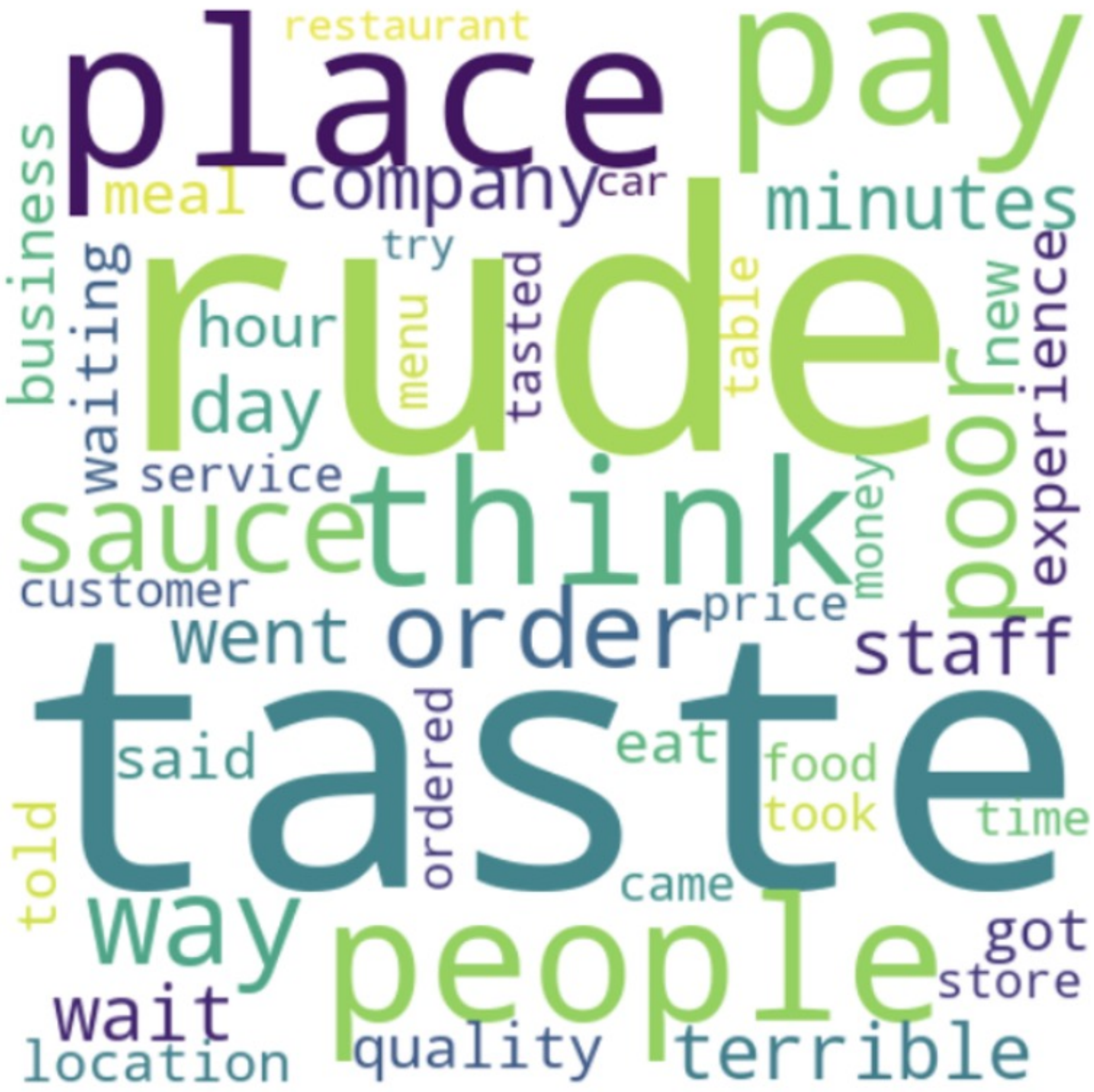
TOPIC MODELING

```python
from  sklearn.feature_extraction.text  import  TfidfVectorizer
from  sklearn.decomposition  import  TruncatedSVD
def  getTopics(df,  topicsNum):
        re  =  ''
        vectorizer  =  TfidfVectorizer(stop_words=stopword_forTfidfVectorizer,
                                                                    #stop_words='english',
                                                                    max_features= 500  #  keep  top  1000  terms
                                                                    ,max_df  =  0.35
                                                                    ,min_df  =  0.05
                                                                    )
        X  =  vectorizer.fit_transform(df['amplified_text'])
        svd_model  =  TruncatedSVD(n_components=topicsNum,  algorithm='randomized',  n_iter=100,  random_state=122)
        svd_model.fit(X)

        terms  =  vectorizer.get_feature_names()
        print(len(terms))
        print(svd_model.components_.shape)
        print(svd_model.feature_names_in_.shape)

        for  i,  comp  in  enumerate(svd_model.components_):
                terms_comp  =  zip(terms,  comp)
                sorted_terms  =  sorted(terms_comp,  key=  lambda  x:x[1],  reverse=True)[:10]
                string  =  "Topic  "+str(i+1)+":  "
                for  t  in  sorted_terms:
                        string  =  string  +  t[0]  +  '  '
                        re  =  re  +  t[0]  +  '  '
                print(string)
        return  re
```

TOPIC MODELING



POSITIVE



NEGATIVE

NEUTRAL COMMENT

neu_neg 3 stars comment with compound score <= Q1

neu_pos 3 stars comment with compound score > Q3

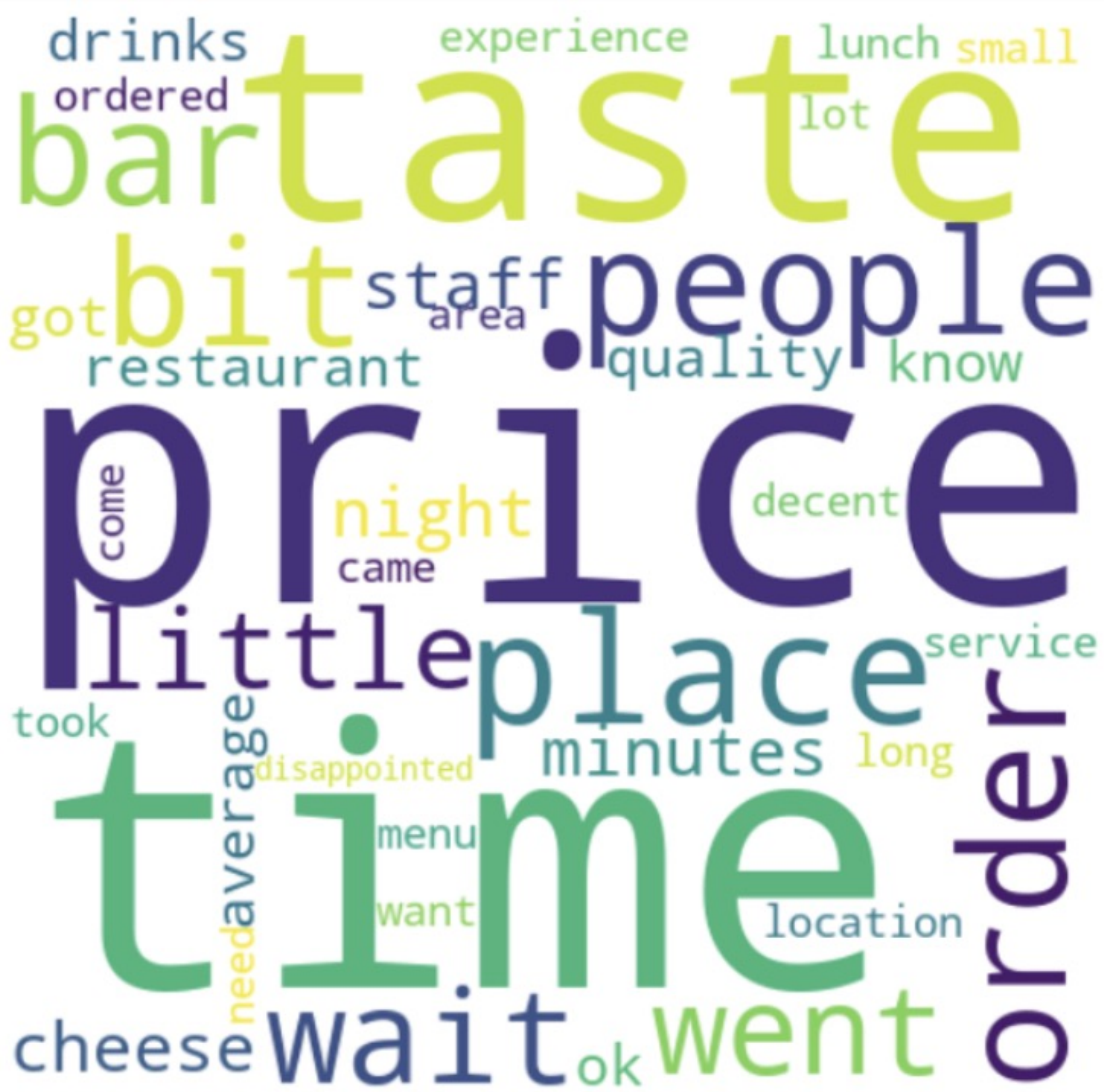
neu_nue 3 stars comment with compound between Q1 and Q3

neu_neg	2728
neu_neu	5457
neu_pos	2721
not_neu	89030

	sentiment	Q1_compound	Q2_compound	Q3_compound
0	negative	-0.7351	0.4019	0.9393
1	neutral	0.7574	0.9459	0.9896
2	positive	0.9081	0.9689	0.9924

TOPIC MODELING

POSITIVE



NEGATIVE

TOPIC MODELING

BOOK STORE

CAR RENTAL

CONCLUSION

- Successfully extract why users give positive or negative comments
- Extract the negative and positive comment from neutral comment
- Understand why the neutral comment is skewed toward positive or negative
- Offer accurate advice to the other business

Q&A