kaggle Search kaggle Competitions Datasets Kernels Discussion Learn Sign In

Swaroop Kallakuri

Yelp Data Analysis

▲ 17 voters

last run 3 months ago · IPython Notebook HTML · 1,860 views
using data from Yelp Dataset · Public

Notebook Code Data (1) Comments (1) Log Versions (3) Forks (1) Fork Notebook

Tags data visualization beginner eda tutorial food and drink

Notebook

Abstract

The objective is to design system which will use existing yelp data to provide insightful analytics and help existing business owners, future business owners to make important decisions regarding new business or business expansion.40% of world population has internet connection today compared to 1% in 1995. Almost 3 exabytes of data is created per day using internet. Storing huge amount of data and retrieving knowledge out of it is challenging task these days. Yelp is a website which publishes crowd sourced reviews about local businesses (Restaurants, Department Stores, Bars, Home-Local Services, Cafes, Automotive). It provides opportunity to business owners to improve their services and users to choose best business amongst available.

Introduction

Yelp is a local business directory service and review site with social networking features. It allows users to give ratings and review businesses. The review is usually short text consisting of few lines with about hundred words. Often, a review describes various dimensions about a business and the experience of user with respect to those dimensions.This dataset is a subset of Yelp's businesses, reviews, and user data. It was originally put together for the Yelp Dataset Challenge which is a chance for students to conduct research or analysis on Yelp's data and share their discoveries. In the dataset you'll find information about businesses across 11 metropolitan areas in four countries.

Glossary of terms

1. **Yelp**: A website which publishes crowd source reviews to help users and business owners . Business: A local body listed on yelp like Restaurants, Department Stores, Bars, Home-Local Services, Cafes, Automotive.
2. **Existing business owner**: A person who has listed his business on Yelp site and getting views from yelp users.
3. **Future business owner**: A person who wants to start new business in future time
4. **User**: A person who has registered on yelp who is writing reviews about different business after vising them or a person who is using yelp reviews to choose business.
5. **Analytics**: Extract knowledge out of data which can be used by system users to make important decisions which is very difficult just by looking at the data.
6. **Review**: It is text written by user after vising business about the over-all experience. I is also a numeric representation (out of 5) to compare it with other business.

In [1]:

```
# This Python 3 environment comes with many helpful analyt
# This Python 3 environment comes with many helpful analyt
ics libraries installed
# It is defined by the kaggle/python docker image: http
s://github.com/kaggle/docker-python
# For example, here's several helpful packages to load in

import numpy as np # linear algebra
import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt
%matplotlib inline
from matplotlib.pylab import rcParams
rcParams['figure.figsize'] = 12, 4
#import warnings
#warnings.filterwarnings('ignore')
pd.set_option('display.max_columns', 500)
pd.set_option('display.max_columns', 100)
# Input data files are available in the "../input/" direct
ory.
# For example, running this (by clicking run or pressing S
hift+Enter) will list the files in the input directory

from subprocess import check_output
print(check_output(["ls", "../input"]).decode("utf8"))

# Any results you write to the current directory are saved
 as output.
```

```
Dataset_Challenge_Dataset_Agreement.pdf
yelp_business.csv
yelp_business_attributes.csv
yelp_business_hours.csv
yelp_checkin.csv
yelp_review.csv
yelp_tip.csv
yelp_user.csv
```

In [2]:

```
business = pd.read_csv('../input/yelp_business.csv')
```

In [3]:

```
business.head(5)
```

Out[3]:

	business_id	name	neighborhoo
0	FYWN1wneV18bWNgQjJ2GNg	"Dental by Design"	NaN
1	He-G7vWjzVUysIKrfNbPUQ	"Stephen Szabo	NaN

		Salon"	
2	KQPW8lFf1y5BT2MxiSZ3QA	"Western Motor Vehicle"	NaN
3	8DShNS-LuFqpEWIp0HxijA	"Sports Authority"	NaN
4	PfOCPjBrlQAnz_NXj9h_w	"Brick House Tavern + Tap"	NaN

In [4]:

```
business_hours = pd.read_csv("../input/yelp_business_hour
s.csv")
```

In [5]:

```
business_hours.head()
```

Out[5]:

	business_id	monday	tuesday	wec
0	FYWN1wneV18bWNgQjJ2GNg	7:30-17:0	7:30-17:0	7:3
1	He-G7vWjzVUysIKrfNbPUQ	9:0-20:0	9:0-20:0	9:0
2	KQPW8lFf1y5BT2MxiSZ3QA	None	None	Nor
3	8DShNS-LuFqpEWIp0HxijA	10:0-21:0	10:0-21:0	10:
4	PfOCPjBrlQAnz_NXj9h_w	11:0-1:0	11:0-1:0	11:

In [6]:

```
business.columns
```

Out[6]:

```
Index(['business_id', 'name', 'neighborhood',
'address', 'city', 'state',
       'postal_code', 'latitude', 'longitud
e', 'stars', 'review_count',
       'is_open', 'categories'],
      dtype='object')
```

In [7]:

```
business.shape
```

Out[7]:

```
(174567, 13)
```

In [8]:

```
#Null Values...
business.isnull().sum().sort_values(ascending=False)
```

Out[8]:

```
neighborhood    106552
postal_code        623
longitude            1
latitude             1
state                1
city                 1
categories           0
is_open              0
review_count         0
stars                0
address              0
name                 0
business_id          0
dtype: int64
```

In [9]:

```
#are all business Id's unique?
business.business_id.is_unique #business_id is all unique
```

Out[9]:

```
True
```

In [10]:

```
business.city.value_counts()
```

Out[10]:

```
Las Vegas                     26775
Phoenix                       17213
Toronto                       17206
Charlotte                      8553
Scottsdale                     8228
Pittsburgh                     6355
Mesa                           5760
Montréal                       5709
Henderson                      4465
Tempe                          4263
Chandler                       3994
Edinburgh                      3868
Cleveland                      3322
```

```
Madison                        3213
Glendale                       3206
Gilbert                        3128
Mississauga                    2726
Stuttgart                      2000
Peoria                         1706
Markham                        1564
North Las Vegas                1393
Champaign                      1195
Scarborough                    1095
North York                     1092
Surprise                       1018
Richmond Hill                   888
Concord                         864
Brampton                        839
Goodyear                        772
Vaughan                         768
                              ...
M7                                1
Pincourt                          1
Dollard-des Ormeaux               1
East Gwilimbury                   1
Canonsburd                        1
Chateau                           1
Lake Park                         1
Middleburg Hts.                   1
Chertsey                          1
Currie                            1
Allegheny                         1
Bedford Hts.                      1
Hemmingford                       1
N W Las Vegas                     1
Chester Township                  1
Shaker Hts                        1
Cleveland Hghts.                  1
Lübeck                            1
Ben Avon                          1
Northfield Center Township        1
Baie-D'urfe                       1
Mesa Arizona                      1
Shandwick                         1
Plan                              1
Median                            1
Monreoville                       1
Leaside                           1
Henderson and Las vegas           1
Côte-Saint-Luc                    1
Vaughn City                       1
Name: city, Length: 1093, dtype: int64
```

Top 50 most reviewed businesses

In [11]:

```
business[['name', 'review_count', 'city', 'stars']].sort_v
alues(ascending=False, by="review_count")[0:50]
```

```
Out[11]:
```

	name	review_count	city	stars
97944	"Mon Ami Gabi"	7361	Las Vegas	4.0
119907	"Bacchanal Buffet"	7009	Las Vegas	4.0
69993	"Wicked Spoon"	5950	Las Vegas	3.5
81212	"Gordon Ramsay BurGR"	5447	Las Vegas	4.0
139699	"Earl of Sandwich"	4869	Las Vegas	4.5
19191	"Hash House A Go Go"	4774	Las Vegas	4.0
80590	"The Buffet"	4018	Las Vegas	3.5
124412	"Lotus of Siam"	3964	Las Vegas	4.0
21006	"Serendipity 3"	3910	Las Vegas	3.0
93038	"The Buffet at Bellagio"	3838	Las Vegas	3.5
26748	"ARIA Resort & Casino"	3794	Las Vegas	3.5
80626	"The Cosmopolitan of Las Vegas"	3772	Las Vegas	4.0
25096	"Secret Pizza"	3741	Las Vegas	4.0
6670	"Luxor Hotel and Casino Las Vegas"	3621	Las Vegas	2.5
6782	"Bouchon at the Venezia Tower"	3570	Las Vegas	4.0
10567	"MGM Grand Hotel"	3444	Las Vegas	3.0
169223	"McCarran International Airport"	3284	Las Vegas	3.5
170798	"Gangnam Asian BBQ Dining"	3262	Las Vegas	4.5
112523	"The Venetian Las Vegas"	3101	Las Vegas	4.0
116002	"Bachi Burger"	3065	Las Vegas	4.0
50087	"Hash House A Go Go"	3050	Las Vegas	4.0
43069			Las	

[illegible]	"Mesa Grill"	3012	Las Vegas	4.0
106267	"Flamingo Las Vegas Hotel & Casino"	2938	Las Vegas	2.5
138776	"Gordon Ramsay Steak"	2935	Las Vegas	4.0
60845	"XS Nightclub"	2884	Las Vegas	4.0
126918	"Bellagio Hotel"	2780	Las Vegas	3.5
124334	"Holsteins Shakes and Buns"	2771	Las Vegas	4.0
24586	"The Peppermill Restaurant & Fireside Lounge"	2703	Las Vegas	4.0
69105	"Mandalay Bay Resort & Casino"	2687	Las Vegas	3.5
8709	"Planet Hollywood Las Vegas Resort & Casino"	2681	Las Vegas	3.0
36120	"Guy Fieri's Vegas Kitchen & Bar"	2674	Las Vegas	3.5
128975	"Egg & I"	2595	Las Vegas	4.5
166920	"Pho Kim Long"	2594	Las Vegas	3.5
27862	"Shake Shack"	2549	Las Vegas	4.0
2803	"Monte Carlo Hotel And Casino"	2507	Las Vegas	2.5
800	"Excalibur Hotel"	2504	Las Vegas	2.5
118712	"Gordon Ramsay Pub & Grill"	2502	Las Vegas	3.5
108433	"Grand Lux Cafe"	2490	Las Vegas	4.0
100272	"Tacos El Gordo"	2448	Las Vegas	4.0
154617	"Wynn Las Vegas"	2441	Las Vegas	4.0
161630	"Burger Bar"	2440	Las Vegas	4.0
155142	"Caesars Palace Las Vegas Hotel & Casino"	2393	Las Vegas	3.0
73007	"Yardbird Southern Table & Bar"	2360	Las Vegas	4.5
5068	"Giada"	2349	Las	3.5

			Vegas	
89974	"Rollin Smoke Barbeque"	2320	Las Vegas	4.5
13125	"Vdara Hotel"	2315	Las Vegas	4.0
20329	"Monta Ramen"	2291	Las Vegas	4.0
73708	"The Palazzo Las Vegas"	2248	Las Vegas	4.0
102148	"Treasure Island"	2237	Las Vegas	3.0
4137	"Phoenix Sky Harbor International Airport"	2215	Phoenix	3.0

Number of businesses listed in different cities

In [12]:

```
city_business_counts = business[['city', 'business_id']].g
roupby(['city'])\
['business_id'].agg('count').sort_values(ascending=False)
```

In [13]:

```
city_business_counts = pd.DataFrame(data=city_business_cou
nts)
```

In [14]:

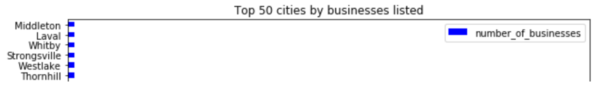
```
city_business_counts.rename(columns={'business_id' : 'numb
er_of_businesses'}, inplace=True)
```

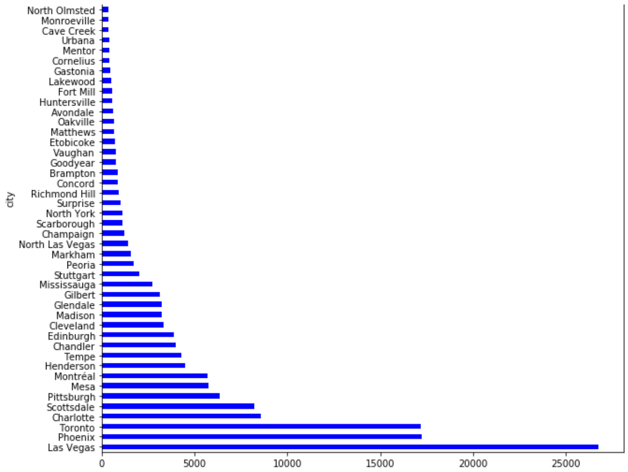
In [15]:

```
city_business_counts[0:50].sort_values(ascending=False, by
="number_of_businesses")\
.plot(kind='barh', stacked=False, figsize=[10,10], colorma
p='winter')
plt.title('Top 50 cities by businesses listed')
```

Out[15]:

```
Text(0.5,1,'Top 50 cities by businesses liste
d')
```





Cities with most reviews and best ratings for their businesses

In [16]:

```
city_business_reviews = business[['city', 'review_count',
'stars']].groupby(['city']).\
agg({'review_count': 'sum', 'stars': 'mean'}).sort_values(
by='review_count', ascending=False)
city_business_reviews.head(10)
```

Out[16]:

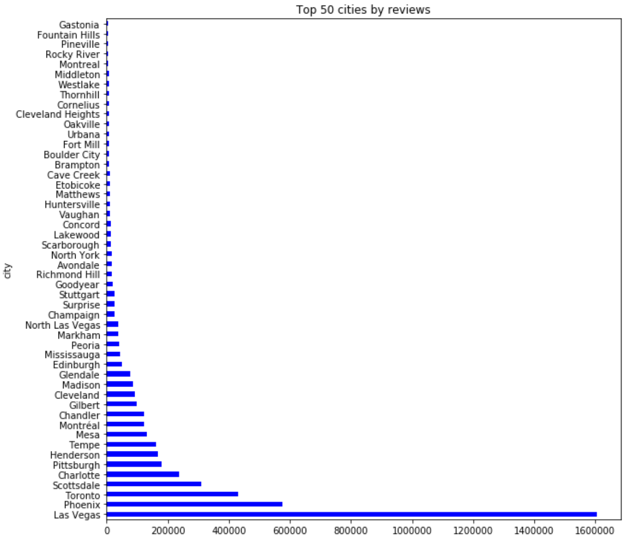
	review_count	stars
city		
Las Vegas	1604173	3.709916
Phoenix	576709	3.673793
Toronto	430923	3.487272
Scottsdale	308529	3.948529
Charlotte	237115	3.571554
Pittsburgh	179471	3.629819
Henderson	166884	3.789362
Tempe	162772	3.729885
Mesa	130883	3.636024
Montréal	122620	3.706604

In [17]:

```
city_business_reviews['review_count'][0:50].plot(kind='bar
h', stacked=False, figsize=[10,10], \
                                              colormap=
'winter')
plt.title('Top 50 cities by reviews')
```

Out[17]:

```
Text(0.5,1,'Top 50 cities by reviews')
```

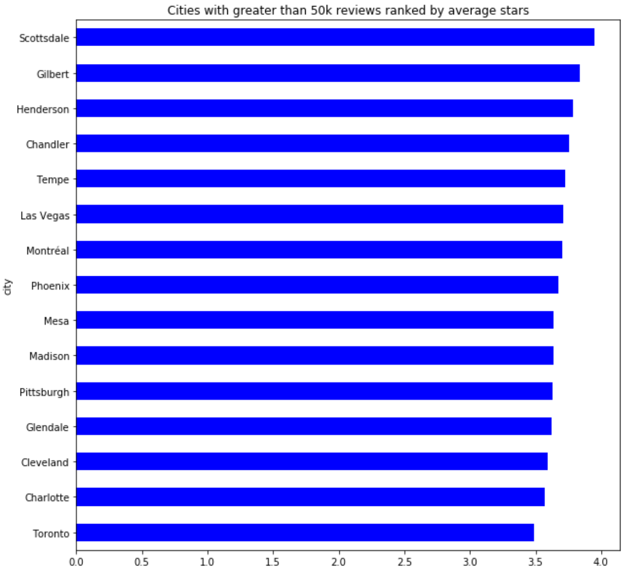


In [18]:

```
city_business_reviews[city_business_reviews.review_count >
 50000]['stars'].sort_values()\
.plot(kind='barh', stacked=False, figsize=[10,10], colorma
p='winter')
plt.title('Cities with greater than 50k reviews ranked by
 average stars')
```

Out[18]:

```
Text(0.5,1,'Cities with greater than 50k revi
ews ranked by average stars')
```



Distribution of stars

In [19]:

In [19]:

```
business['stars'].value_counts()
```

Out[19]:

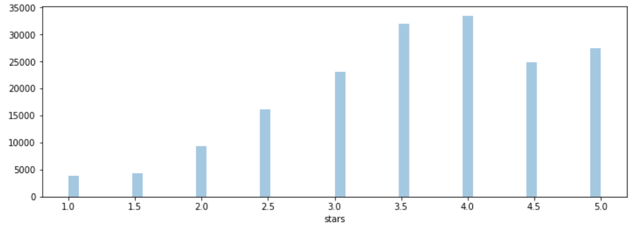
```
4.0    33492
3.5    32038
5.0    27540
4.5    24796
3.0    23142
2.5    16148
2.0     9320
1.5     4303
1.0     3788
Name: stars, dtype: int64
```

In [20]:

```
sns.distplot(business.stars, kde=False)
```

Out[20]:

```
<matplotlib.axes._subplots.AxesSubplot at 0x7
fddf08c5358>
```



How many are open and how many closed?

In [21]:

```
business['is_open'].value_counts()
```

Out[21]:

```
1    146702
0     27865
Name: is_open, dtype: int64
```

Lets look into user tips on businesses before looking at reviews

In [22]:

```
tip = pd.read_csv('../input/yelp_tip.csv')
```

In [23]:

```
tip.head(10)
```

Out[23]:

	text	date	likes	business_i
0	Great breakfast large portions and friendly wa...	2015-08-12	0	jH19V2I9fIslnNhDzPmdkA
1	Nice place. Great staff. A fixture in the tow...	2014-06-20	0	dAa0hB2yrnHzVmsCkN4Yv
2	Happy hour 5-7 Monday - Friday	2016-10-12	0	dAa0hB2yrnHzVmsCkN4Yv
3	Parking is a premium, keep circling, you will ...	2017-01-28	0	ESzO3Av0b1_TzKOiqzbQYQ
4	Homemade pasta is the best in the area	2017-02-25	0	k7WRPbDd7rztjHcGGkEjlw
5	Excellent service, staff is dressed profession...	2017-04-08	0	k7WRPbDd7rztjHcGGkEjlw
6	Come early on Sunday's to avoid the rush	2016-07-03	0	SqW3igh1_Png336VIb5DUA
7	Love their soup!	2016-01-07	0	KNpcPGqDORDdvtekXd348
8	Soups are fantastic!	2016-05-22	0	KNpcPGqDORDdvtekXd348
9	Thursday night is $5 burger night	2016-06-09	0	KNpcPGqDORDdvtekXd348

In [24]:

```
tip.shape
```

Out[24]:

```
(1098324, 5)
```

How many of the selected words are used in the user tips?

In [25]:

```
selected_words = ['awesome', 'great', 'fantastic', 'amazin
g', 'love', 'horrible', 'bad', 'terrible',
                  'awful', 'wow', 'hate']
selected_words
```

Out[25]:

```
['awesome',
 'great',
 'fantastic',
 'amazing',
 'love',
 'horrible',
 'bad',
 'terrible',
 'awful',
 'wow',
 'hate']
```

In [26]:

```
from sklearn.feature_extraction.text import CountVectorize
r
vectorizer = CountVectorizer(vocabulary=selected_words, lo
wercase=False)
#corpus = ['This is the first document.','This is the seco
nd second document.']
#print corpus
selected_word_count = vectorizer.fit_transform(tip['text']
.values.astype('U'))
vectorizer.get_feature_names()
```

Out[26]:

```
['awesome',
 'great',
 'fantastic',
 'amazing',
 'love',
 'horrible',
 'bad',
 'terrible',
 'awful',
 'wow',
 'hate']
```

In [27]:

```
word_count_array = selected_word_count.toarray()
word_count_array.shape
```

Out[27]:

```
(1098324, 11)
```

In [28]:

```
word_count_array.sum(axis=0)
```

Out[28]:

```
array([22354, 77169,  5168, 26547, 27972,  32
33, 10207,  2589,  1338,
         862,  1214])
```

In [29]:

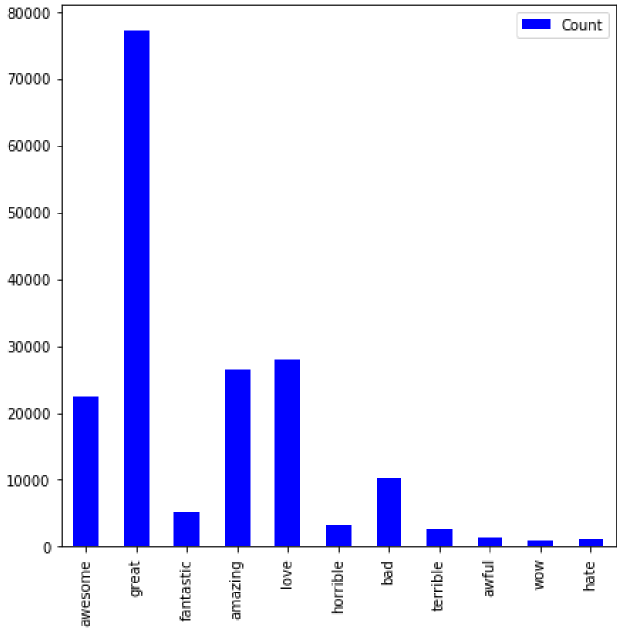
```
temp = pd.DataFrame(index=vectorizer.get_feature_names(),
 \
                    data=word_count_array.sum(axis=0)).ren
ame(columns={0: 'Count'})
```

In [30]:

```
temp.plot(kind='bar', stacked=False, figsize=[7,7], colorm
ap='winter')
```

Out[30]:

```
<matplotlib.axes._subplots.AxesSubplot at 0x7
fdd11c57470>
```



We see that most of the tips are positive rather than negative!

Lets look at one restaurant with high star rating and one with low star rating and see what the user tips look like

Lets look at "Earl of Sandwich" restaurant in Las Vegas which has 4.5 rating

In [31]:

```
business[(business['city'] == 'Las Vegas') & (business['st
ars'] == 4.5)]
```

Out[31]:

	business_id	name	nei
26	VBHEsoXQb2AQ76J9l8h1uQ	"Alfredo's Jewelry"	So
41	1Jp_hmPNUZArNqzpbm7B0g	"Task Electric"	Sp
60	v2GJWvZqEAjUc22hZUYzYw	"John Armond Actor's Studio"	We
82	bOOgAB_CEWWsxalAthnRSw	"Tenors of Rock"	Th
110	ZmMCgM4RCqCXJ0Lswu6yxw	"A Professional Appliance Repair"	Na
153	PJ-VbAtIOso1dqd2frQqqg	"Donut Tyme"	Su
176	cehTmoCXPi0a3FwCE3Tq2Q	"Red Wing Shoes"	So
177	Uy3_5nLo3sYkAuSX6mjdmg	"Geebee's Bar & Grill"	So
193	P5TLch0Fu9p3o6W2hRSz0g	"Terrible Herbst"	So
225	xiSEUnaX77EhNz-l3ag7RA	"LV Nail Lounge"	Sp
274	dPxZI9lrKTl5dvFfnb1_lg	"Trattoria Italia"	An
303	WfB_SsYeKy83QQsqAAyGVQ	"Cancun Bar & Grill"	So

		& Grill	
333	oDMiR7xWFWNSG4zOXFajdg	"Coral Academy of Science"	So
366	u29lf2yPd-qK5ThAS9FRQQ	"Kinthai"	We
457	1e2ZRUm9lpX3vrmraRx-yQ	"Tacos Colima"	Na
505	bs5ZQW4z83ml6kuZWd-Q1A	"Enliven Skin and Beauty"	We
518	lil28runuoryt7uMQLXSRQ	"Flipperspiel Underground Arcade Club"	So
574	0Yeb_P24sj6MwG2qmuehkA	"Till's Bar"	Na
596	hTzcHtk4-0QJnFUbkKpd5Q	"Citi Trends"	Na
691	_ewxwEwJM-IYflYnKpQOZw	"Mexicali Raspados"	Ea
778	v0byOL8VL6v6muGa1anxFA	"The Hummus Factory"	We
783	i57cZRoLUU9QUPCI0ErWGQ	"Dad's Bail Bonds"	Do
839	3MBON3dW2a1NKjgo9H780Q	"Anson Edwards & Higgins Plastic Surgery Assoc...	Sp
883	VOHbo_5g1rLwu65AkMHacQ	"Hair'z Melinda"	Su
918	crGdKSRKi25R2vorZ7skzg	"Tortilleria San Diego"	Do
952	kvRJjMN1XZtfgLxhVP_BPw	"Modern Landscape, LLC"	Ce
974	4Zqv7NyeiuqMOV8wWlhB4A	"ReVamp Extensions"	Sp
990	zcVZc4SadqLgUHKWL2ZilQ	"ChefKas Catering"	Sp
1070	USMFeacfapj3IXvZahIj-A	"Vape Kraze Vapors"	Do
1077	Njydzc1qePniw9hdkikYbg	"Gettinger Chiropractic"	Do
...	...	...	...
173097	gKj0wkJhQ0YT-Aw65e7hQQ	"Stephen K Montoya, MD"	Ea

173149	1La-mfFF5RTYOhujtSUVaQ	"911 Keys & Sound"	Ea
173184	N0YNLwSNejIpJ1E3qQorhg	"FiveStar Mobile iPhone Repair"	Th
173206	pvASvwMIwe7-zY5rN9oBAQ	"Fast-Fix Jewelry & Watch Repairs"	So Su
173225	VIW-4GDAKGaXUdLtTVSFqQ	"Khoury's Fine Wine & Spirits"	An
173343	maUGKOLNdvYKUVQoyh0-gg	"The Law Office of Joseph P Reiff"	Do
173393	fnfh9LuxfmLw59vC_kEBCA	"Madewell"	Th
173461	1_RnfPCfPKAhEolhM5yeUw	"Scott Biggs, DDS - Micro Endodontics"	No
173500	ad7NXTxHr2BmjyQvGnfotQ	"Specialty Surgery Center"	Na
173518	LIU7IcJtD9Vieolo__wd9Q	"Thai Pan Cuisine"	We
173580	QMozb3XreozGjEMrabs__A	"The Joint Chiropractic - Blue Diamond"	So
173612	EwN1LCoJXB0z_a-LxLFKyQ	"Paleteria Y Neveria Mexicana"	Su
173781	BD18SKv935HDmIKrLPkhLA	"Chocolate Swan"	Na
173825	kd96_x4saxuoe_eJHbS87A	"Martin Garage Doors of Nevada"	Na
173915	lmpJvZl_F9iBmNoHi4jW1A	"Bestcuts & More"	Na
174011	ekTUBCcsRTheOvYa7fPQpQ	"Sweet Bubble Bath Confections in Excalibur"	Th
174020	1dl7wV-zwqliz0xOrzLBUQ	"Universal Solar Direct of Las Vegas"	Na
174098			

174098	Tefx_N6A6nrsdj4jHHnbYg	"Le Petit Café & Bakery"	Sp
174131	ZEouZiCVwjIa4ePWboaVkg	"Raysco"	Na
174184	_rQb4DXr4i-XOb3c_LOKdg	"Bikram Yoga Westside"	So Su
174195	WBdgcjOt9qJfaeGcdzaMLA	"Chic Cache"	We
174220	muriGdv1pnJaNZTQfZq9CQ	"Kim Layson Beauty"	Sp
174321	m3_NFDiJ8ib2fUzoqwzm5Q	"Buy Buy Baby"	So Su
174332	RgLA2YwJ53xoeMMgc7L7oA	"Scooter Nation"	So
174355	Nu-SSGx_BFb9eMOm8qO4Mg	"Paul's Auto Service"	Do
174380	l3l3RvS7lXogVpanFu6QlA	"Nulook Floor"	Na
174386	EZ0pK8z6jG8uv4DNZhrRuA	"11th Street Records"	Do
174417	MKrvEEejLBeUsjZRBtVxrQ	"Green Valley Shoe Repair"	So
174455	Fv4EXwV30rwGD2NzN1ekgA	"Gorilla Sushi"	Ea
174539	swjz4q8gl79Ndg4APuHEUA	"Stonegate Real Estate Services"	We

4006 rows × 13 columns

In [32]:

```
business[business.name=='"Earl of Sandwich"']
```

Out[32]:

	business_id	name	neighb
107416	Ffhe2cmRvloz3CCdRGvHtA	"Earl of	NaN

		Sandwich"	
131049	3fT1kcQ-MVEImGHa3hll5w	"Earl of Sandwich"	South Summ
139699	DkYS3arLOhA8si5uUEmHOw	"Earl of Sandwich"	The St
166792	fE7x3Ui2mzdwdfJnd7r_1g	"Earl of Sandwich"	The St

- Points to remember:
  1. There are 4 branches
  2. Two of them are on the strip
  3. Since there are multiple, lets pick by index

In [33]:

```
# This is where  have been to :)
business.loc[139699,:]
```

Out[33]:

```
business_id                               Dk
YS3arLOhA8si5uUEmHOw
name
  "Earl of Sandwich"
neighborhood
           The Strip
address                                  "36
67 Las Vegas Blvd S"
city
           Las Vegas
state
                  NV
postal_code
               89109
latitude
             36.1082
longitude
            -115.172
stars
                 4.5
review_count
                4869
is_open
                   1
categories      Caterers;Sandwiches;Restauran
ts;Food Delivery ...
Name: 139699, dtype: object
```

```
Name: 139699, dtype: object
```

In [34]:

```
earl_of_sandwich = tip[tip.business_id==business.loc[13969
9,:].business_id]
```

In [35]:

```
earl_of_sandwich_selected_word_count = \
vectorizer.fit_transform(earl_of_sandwich['text'].values.a
stype('U'))
```

In [36]:

```
word_count_array = earl_of_sandwich_selected_word_count.to
array()
temp = pd.DataFrame(index=vectorizer.get_feature_names(),
 \
                    data=word_count_array.sum(axis=0)).ren
ame(columns={0: 'Count'})
temp
```

Out[36]:

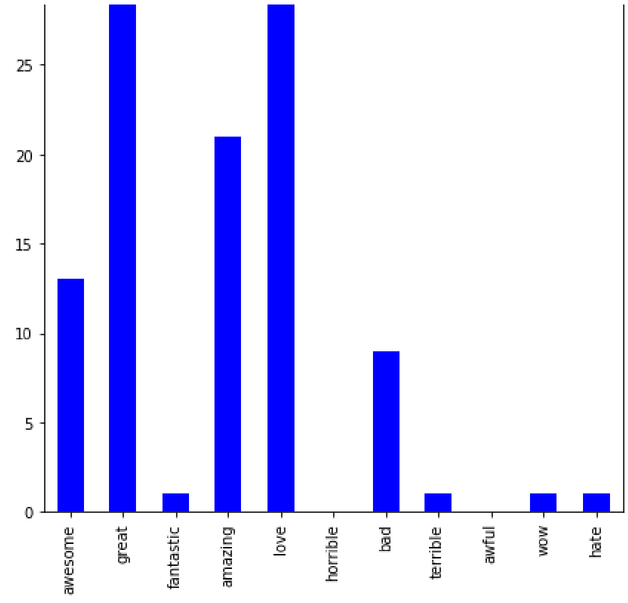
	Count
awesome	13
great	30
fantastic	1
amazing	21
love	30
horrible	0
bad	9
terrible	1
awful	0
wow	1
hate	1

In [37]:

```
temp.plot(kind='bar', stacked=False, figsize=[7,7], colorm
ap='winter')
```

Out[37]:

```
<matplotlib.axes._subplots.AxesSubplot at 0x7
fddf7ca12b0>
```



Did you find this Kernel useful?
Show your appreciation with an upvote

17

Comments (1)

All Comments

Sort by Hotness

Please sign in to leave a comment.

Venu Gopal • Posted on Version 2 • 4 months ago • Options

0

business[business.name=='"Earl of Sandwich"'] has 4 different locations or branches and have different ratings. If I want to get all the restaurants with multiple branches and their star ratings and can I correlate with the reviews? how can I do that?

Similar Kernels

Yelp Ratings EDA

Chocolate Dataset Visualisation For Beginners

Wine EDA

Data ScienceTutorial For Beginners

Titanic Survival Prediction End To End ML Pipeline

Our Team Terms Privacy Contact/Support