

In [1]:

```
import py_entitymatching as em #Import megallan entity matching library
import math
import warnings
warnings.filterwarnings('ignore')

# Pandas has useful data-structure and analysis tool
import pandas as pd
# numpy has a lots of useful math related modules
import numpy as np
# to check execution time
import time
# Helpful function to display intermittent result
from IPython.display import display
```

In [39]:

```
matching_records = em.read_csv_metadata("filtered_predictions.csv",key="id")
yelp_original = em.read_csv_metadata("yelp_original.csv",key="id")
zomato_original = em.read_csv_metadata("zomato_original.csv",key="id")
matching_records.head(1)
```

Out[39]:

	Unnamed: 0	id	Unnamed: 1	restaurant_name	phone	zipcode	state	city	address	delivery	tz
0	0	1	512	McSorley's Old Ale House	(212) 473- 9148	10003	NY	New York	15 E 7th Street	0	

Load the dataset

In [3]:

```
#sqlalchemy for sqlite3 use
from sqlalchemy import create_engine
import sqlite3
#create a database where we'll load the dataset from the csv file
engine = create_engine('sqlite:///yelpzomato.db')
connection = engine.connect()
```

In [4]:

```
import string

for data in pd.read_csv("filtered_predictions.csv",
    iterator=True, encoding='utf-8'):
    data.to_sql('data', engine, if_exists='append')
```

impact of delivery facility on pricing and rating

In [5]:

```
a = connection.execute("select AVG(rating) from data where delivery = 0")
b = connection.execute("select AVG(rating) from data where delivery = 1")
for data in zip(a,b):
    print data
```

```
((3.97051282051282,,), (4.057500000000001,))
```

Insight -

Takeout facility hasn't affected the rating significantly.

In [6]:

```
a = connection.execute("select AVG(pricy) from data where delivery = 0")
b = connection.execute("select AVG(pricy) from data where delivery = 1")
for data in zip(a,b):
    print data
```

```
((2.3361702127659574,,), (1.9,))
```

Insight -

Restaurants that do not offer delivery facility on average are more pricy.

impact of takeout facility on pricing and rating

In [7]:

```
a = connection.execute("select AVG(rating) from data where takeout = 0")
b = connection.execute("select AVG(rating) from data where takeout = 1")
for data in zip(a,b):
    print data
```

```
((3.9120000000000004,,), (4.010050251256281,))
```

Insight -

Takeout facility hasn't affected the rating significantly, which means that perhaps people do not care much about outdoor seating

In [8]:

```
a = connection.execute("select AVG(pricy) from data where takeout = 0")
b = connection.execute("select AVG(pricy) from data where takeout = 1")
for data in zip(a,b):
    print data
```

```
((2.763157894736842,), (2.0854271356783918,))
```

Insight -

Restaurants that do not offer takeout facility on average are more pricy.

impact of outdoor seating on pricing and rating

In [9]:

```
a = connection.execute("select AVG(rating) from data where outdoor_seating = 0")
b = connection.execute("select AVG(rating) from data where outdoor_seating = 1")
for data in zip(a,b):
    print data
```

```
((3.999371069182388,), (3.9608695652173904,))
```

Insight -

Outdoor seating hasn't affected the rating significantly, which means that perhaps people do not care much about outdoor seating.

In [30]:

```
a = connection.execute("select AVG(pricy) from data where takeout = 0")
b = connection.execute("select AVG(pricy) from data where takeout = 1")
for data in zip(a,b):
    print data
```

```
((2.763157894736842,), (2.0854271356783918,))
```

Insight -

Restaurants that do not offer outdoor seating on average are more pricy.

impact of delivery and takeout on rating and pricing

In [11]:

```
a = connection.execute("select AVG(rating) from data where delivery = 0 and takeout = 0")
b = connection.execute("select AVG(rating) from data where delivery = 0 and takeout = 1")
c = connection.execute("select AVG(rating) from data where delivery = 1 and takeout = 0")
d = connection.execute("select AVG(rating) from data where delivery = 1 and takeout = 1")
for data in zip(a,b,c,d):
    print data
```

```
((3.9120000000000004,), (3.998113207547168,), (None,), (4.057500000000001,))
```

In [12]:

```
a = connection.execute("select AVG(pricy) from data where delivery = 0 and takeout = 0")
b = connection.execute("select AVG(pricy) from data where delivery = 0 and takeout = 1")
c = connection.execute("select AVG(pricy) from data where delivery = 1 and takeout = 0")
d = connection.execute("select AVG(pricy) from data where delivery = 1 and takeout = 1")
for data in zip(a,b,c,d):
    print data
```

```
((2.763157894736842,), (2.1320754716981134,), (None,), (1.9,))
```

impact of delivery and outdoor seating on rating and pricing

In [13]:

```
a = connection.execute("select AVG(rating) from data where delivery = 0 and outdoor_seating = 0")
b = connection.execute("select AVG(rating) from data where delivery = 0 and outdoor_seating = 1")
c = connection.execute("select AVG(rating) from data where delivery = 1 and outdoor_seating = 0")
d = connection.execute("select AVG(rating) from data where delivery = 1 and outdoor_seating = 1")
for data in zip(a,b,c,d):
    print data
```

```
((3.96439393939394,), (3.978431372549019,), (4.170370370370371,), (3.8230769230769233,))
```

In [14]:

```
a = connection.execute("select AVG(pricy) from data where delivery = 0 and takeout = 0")
b = connection.execute("select AVG(pricy) from data where delivery = 0 and takeout = 1")
c = connection.execute("select AVG(pricy) from data where delivery = 1 and takeout = 0")
d = connection.execute("select AVG(pricy) from data where delivery = 1 and takeout = 1")
for data in zip(a,b,c,d):
    print data
```

```
((2.763157894736842,), (2.1320754716981134,), (None,), (1.9,))
```

impact of takeout and outdoor seating on rating and pricing

In [15]:

```
a = connection.execute("select AVG(rating) from data where takeout = 0 and outdoor_seating = 0")
b = connection.execute("select AVG(rating) from data where takeout = 0 and outdoor_seating = 1")
c = connection.execute("select AVG(rating) from data where takeout = 1 and outdoor_seating = 0")
d = connection.execute("select AVG(rating) from data where takeout = 1 and outdoor_seating = 1")
for data in zip(a,b,c,d):
    print data
```

```
((3.9078431372549027,), (3.9208333333333333,), (4.042592592592592,), (3.9714285714285693,))
```

In [16]:

```
a = connection.execute("select AVG(pricy) from data where takeout = 0 and outdoor_seating = 0")
b = connection.execute("select AVG(pricy) from data where takeout = 0 and outdoor_seating = 1")
c = connection.execute("select AVG(pricy) from data where takeout = 1 and outdoor_seating = 0")
d = connection.execute("select AVG(pricy) from data where takeout = 1 and outdoor_seating = 1")
for data in zip(a,b,c,d):
    print data
```

```
((2.9423076923076925,), (2.375,), (2.1296296296296298,), (2.032967032967033,))
```

impact of delivery, takout and outdoor seating on rating and pricing

In [17]:

```
a = connection.execute("select AVG(rating) from data where delivery = 0 and takeout = 0 and outdoor_seating = 0")
b = connection.execute("select AVG(rating) from data where delivery = 0 and takeout = 0 and outdoor_seating = 1")
c = connection.execute("select AVG(rating) from data where delivery = 0 and takeout = 1 and outdoor_seating = 0")
d = connection.execute("select AVG(rating) from data where delivery = 0 and takeout = 1 and outdoor_seating = 1")
e = connection.execute("select AVG(rating) from data where delivery = 1 and takeout = 0 and outdoor_seating = 0")
f = connection.execute("select AVG(rating) from data where delivery = 1 and takeout = 0 and outdoor_seating = 1")
g = connection.execute("select AVG(rating) from data where delivery = 1 and takeout = 1 and outdoor_seating = 0")
h = connection.execute("select AVG(rating) from data where delivery = 1 and takeout = 1 and outdoor_seating = 1")
for data in zip(a,b,c,d,e,f,g,h):
    print data
```

```
((3.9078431372549027,), (3.9208333333333333,), (4.0000000000000001,), (3.9961538461538444,), (None,), (None,), (4.170370370370371,), (3.8230769230769233,))
```

Interesting Insight -

- Customers prefer restaruants that have delivery and takeout facility and rate them the highest.
- Restaurants that offer all three services overall have received lowest average rating, perhaps since they are cheapest, they do not live upto the standard of the costlier peers in the game.
- Other categories lie in the middle.

In [18]:

```
a = connection.execute("select AVG(pricy) from data where delivery = 0 and takeout = 0")
b = connection.execute("select AVG(pricy) from data where delivery = 0 and takeout = 1")
c = connection.execute("select AVG(pricy) from data where delivery = 1 and takeout = 0")
d = connection.execute("select AVG(pricy) from data where delivery = 1 and takeout = 1")
e = connection.execute("select AVG(pricy) from data where delivery = 0 and takeout = 0")
f = connection.execute("select AVG(pricy) from data where delivery = 0 and takeout = 1")
g = connection.execute("select AVG(pricy) from data where delivery = 1 and takeout = 0")
h = connection.execute("select AVG(pricy) from data where delivery = 1 and takeout = 1")
for data in zip(a,b,c,d,e,f,g,h):
    print data
```

```
((2.9423076923076925,), (2.375,), (2.197530864197531,), (2.0641025641025643,)), (None,), (None,), (1.9259259259259258,), (1.8461538461538463,))
```

Interesting Insight -

- The priciest restaurants do not offer delivery, takeout and outdoor seating facility. They belong to luxurious class.
- The next category is has the restaurants which are slightly less pricy but just offer outdoor seating facility.
- The trend goes on like this and the restaurants which offer all three facilites are the cheapest.

Data Distribution based on city

In [19]:

```
a = connection.execute("select city, count(city) from data group by city")
for data in a:
    print data
```

```
(u'Brooklyn', 5)
(u'Chicago', 91)
(u'New York', 60)
(u'San Francisco', 85)
(u'Seattle', 34)
```

In []:

Brooklyn has very few instances, hence any inference drawn might be inaccurate.

impact of city on rating and pricing

In [20]:

```
a = connection.execute("select city, AVG(pricy), AVG(rating) from data group by city")
for data in a:
    print data
```

```
(u'Brooklyn', 2.6, 4.0600000000000005)
(u'Chicago', 2.3076923076923075, 4.04175824175824)
(u'New York', 2.5833333333333335, 4.114999999999999)
(u'San Francisco', 2.1058823529411765, 3.8440476190476196)
(u'Seattle', 2.0, 3.9264705882352935)
```

- Shows that Seattle is least pricy but has overall average user rating.
- Restaurants in San Francisco have overall worst rating.
- New York has pricest restaurants with highest user rating.
- Brooklyn - no inference due to lack to adequate number of instances.

Restaurants in Chicago which are most expensive

In [27]:

```
a = connection.execute("select restaurant_name,pricy from data where city = 'Chicago'")
for data in a:
    print data
```

```
(u'Benny\u2019s Chop House', 4)
(u'Tru', 4)
(u'Joe\u2019s Seafood', 4)
(u'Alinea', 4)
(u'Schwa', 4)
(u'mk The Restaurant', 4)
(u'North Pond', 4)
```

Number of most pricy(4) restaurants in each city

In [29]:

```
a = connection.execute("select count(restaurant_name),city from data where pricy = 4")
for data in a:
    print data
```

```
(1, u'Brooklyn')
(7, u'Chicago')
(14, u'New York')
(5, u'San Francisco')
```

- The above result suggests that 'New York' has over all more number of pricy restaurants.
- Based the above result and also the data distribution city wise, we observe that, Seattle is the cheapest of

all followed by San Francisco, then Chicago and New York being the most expensive one.

- Brooklyn has not been considered since there are very few instances of Brooklyn.

Impact of Location on Price and Rating

In [34]:

```
a = connection.execute("select min(zipcode), max(zipcode), city from data group by city")
for data in a:
    print data
```

```
(11201, 11231, u'Brooklyn')
(60603, 60661, u'Chicago')
(10001, 10023, u'New York')
(94102, 94133, u'San Francisco')
(98101, 98199, u'Seattle')
```

- The above query gives the range of zipcode for restaurant in records for each city.
- This data can be used for location based analysis.

Location based analysis on New York

In [38]:

```
a = connection.execute("select avg(pricy),avg(rating),count(restaurant_name) from data where city='New York'")
b = connection.execute("select avg(pricy),avg(rating),count(restaurant_name) from data where city='New York'")
c = connection.execute("select avg(pricy),avg(rating),count(restaurant_name) from data where city='New York'")
d = connection.execute("select avg(pricy),avg(rating),count(restaurant_name) from data where city='New York'")
for data in zip(a,b,c,d):
    print data
```

```
((2.0625, 4.175, 16), (2.7142857142857144, 4.128571428571428, 7), (2.5
9375, 4.078124999999999, 32), (4.0, 4.14, 5))
```

Insight -

- We observe that zipcode 10016 - 10023 has costliest restaurants with highest pricy rating.
- Zipcode 10001 - 10005 has cheapest restaurants.
- Maximum number of restaurants lie in zipcode range 10011 - 10015 and the avg rating rating is also 4.07 which is pretty decent.

In [22]:

```
#connection.close()
#engine.dispose()
```


