Create_Data

December 19, 2019

```
In [0]: from google.colab import drive
        drive.mount('/content/drive',force_remount=True)
```

```
Mounted at /content/drive
```

```
In [0]: import pandas as pd
        import json
        from tqdm import tqdm
```

```
In [0]: path="/content/drive/My Drive/"
```

0.0.1 Read and creat data files

In this notebook, we will read the json files and save the features we need as CSV files.

First we want to pick out those users that rated at least five times as active users.

```
In [0]: line_count = len(open(path+'yelp_dataset/review.json').readlines())
        user_ids, business_ids, stars, dates = [], [], [], []
        i=1
        with open(path+"yelp_dataset/review.json") as f:
            for line in tqdm(f, total=line_count):
                blob = json.loads(line)
                user_ids += [blob["user_id"]]
                business_ids += [blob["business_id"]]
                stars += [blob["stars"]]
                dates += [blob["date"]]
        ratings = pd.DataFrame(
           {"user_id": user_ids, "business_id": business_ids, "rating": stars, "date": dates}
        )
```

```
100%|| 6685900/6685900 [03:08<00:00, 35490.78it/s]
```

```
In [0]: user_counts = ratings["user_id"].value_counts()
        active_users = user_counts.loc[user_counts >= 5].index.tolist()
```

```
In [0]: active_user=pd.DataFrame(active_users)
In [0]: len(active_user)
```

```
Out[0]: 286130
```

Save those active user.

```
In [0]: active_user.to_csv('active_user.csv',index=False)
```

```
In [0]: ratings=ratings.loc[ratings['user_id'].isin(active_users)]
```

```
In [0]: ratings.head(5)
```

```
Out[0]:                   user_id             business_id  rating                 date
        0  hG7b0MtEbXx5QzbzE6C_VA  ujmEBvifdJM6h6RLv4wQIg     1.0  2013-05-07 04:34:36
        2  n6-Gk65cPZL6Uz8qRm3NYw  WTqjgwHlXbSFevF32_DJVw     5.0  2016-11-09 20:09:03
        6  jlu4CztcSxrKx56ba1a5AQ  3fw2X5bZYeW9xCz_zGh0Hg     3.0  2016-05-07 01:21:02
        7  d6xvYpyzcfbF_AZ8vMB7QA  zv0-PJCpNk4fgAVUnExYAA     1.0  2010-10-05 19:12:35
        8  sG_h0dIzTKWa3Q6fmb4u-g  b2jN2mm9Wf3RcrZCgfo1cg     2.0  2015-01-18 14:04:18
```

```
In [0]: ratings.to_csv('ratings.csv',index=False)
```

Read business.json and save the features we need as CSV file.

```
In [0]: line_count = len(open(path+"yelp_dataset/business.json").readlines())
        business_ids= []
        name,address,city,state,postal_code,latitude,longtitude,stars,review_count=[],[],[],[]
        is_open,attributes,categories,hours=[],[],[],[]
        with open(path+"yelp_dataset/business.json") as f:
            for line in tqdm(f,total=line_count):
                blob = json.loads(line)
                #print(blob['attributes'])
                business_ids += [blob["business_id"]]
                name += [blob["name"]]
                address += [blob["address"]]
                city += [blob["city"]]
                state += [blob["state"]]
                postal_code += [blob["postal_code"]]
                latitude += [blob["latitude"]]
                longtitude += [blob["longitude"]]
                stars += [blob["stars"]]
                review_count += [blob["review_count"]]
                is_open += [blob["is_open"]]
                attributes += [blob["attributes"]]
                categories+= [blob["categories"]]
                hours += [blob["hours"]]

        business = pd.DataFrame(
           {"business_ids": business_ids, "name": name, "address": address, "city": city,
            'state':state,'postal_code':postal_code,'latitude':latitude,'longtitude':longtitud
```

```
             'stars':stars,'review_count':review_count,'is_open':is_open,'attributes':attributes
             'categories':categories,'hours':hours
            }
        )
```

```
100%|| 192609/192609 [00:03<00:00, 54133.39it/s]
```

```
In [0]: business.to_csv('business.csv',index=False)
```

Read user.json and save the features we need as CSV file.

```
In [0]: line_count = len(open(path+"yelp_dataset/user.json").readlines())
        user_id=[]
        name,review_count,yelping_since,friends,useful,funny,cool,fans,elite=[],[],[],[],[],[]
        average_stars,compliment_more,compliment_cute,compliment_funny=[],[],[],[]
        num_friends=[]
        num_elite=[]
        with open(path+"yelp_dataset/user.json") as f:
            for line in tqdm(f,total=line_count):
                blob = json.loads(line)
                #print(blob['attributes'])
                user_id += [blob["user_id"]]
                name += [blob["name"]]
                review_count += [blob["review_count"]]
                yelping_since += [blob["yelping_since"]]
                friends += [blob["friends"]]
                num_friends+=[blob["friends"].count(',')+1]
                useful += [blob["useful"]]
                funny += [blob["funny"]]
                cool += [blob["cool"]]
                fans += [blob["fans"]]
                elite += [blob["elite"]]
                num_elite+=[blob["elite"].count(',')+1]
                average_stars += [blob["average_stars"]]
                compliment_more += [blob["compliment_more"]]
                compliment_cute+= [blob["compliment_cute"]]
                compliment_funny += [blob["compliment_funny"]]

        user= pd.DataFrame(
           {"user_id": user_id, "name": name, "yelping_since": yelping_since, "review_count": 
            'friends ':friends ,'useful':useful,'funny':funny,'cool':cool,
            'fans':fans,'elite':elite,'average_stars':average_stars,'compliment_more':complimen
            'compliment_cute':compliment_cute,'compliment_funny':compliment_funny,
            'num_friends':num_friends,'num_elite':num_elite
           }
        )
```

```
100%|| 1637138/1637138 [01:32<00:00, 17658.90it/s]
```

Since we only want active users, so we filter out inactive users.

```
In [0]: user2=user.loc[user['user_id'].isin(active_users)]
```

```
In [0]: len(user2)
```

```
Out[0]: 286130
```

```
In [0]: user2.head()
```

```
Out[0]:                  user_id    name  ... num_friends  num_elite
        0  l6BmjZMeQD3rDxWUbiAiow  Rashmi  ...          99          3
        1  4XChL029mKr5hydo79Ljxg   Jenna  ...        1152          1
        2  bc8C_eETBWL0olvFSJJd0w   David  ...          15          1
        4  MM4RJAeH6yuaN8oZDSt0RA   Nancy  ...         231          4
        6  TEtzbpgA2BFBrC0y0sCbfw   Keane  ...        4326          8

        [5 rows x 16 columns]
```

```
In [0]: user2.to_csv('user2.csv',index=False)
```

Read review.json and save the features we need as CSV file.

```
In [0]: line_count = len(open(path+"yelp_dataset/review.json").readlines())
        user_ids, business_ids, stars, dates,text,useful,funny,cool = [], [], [], [],[],[],[],
        with open(path+"yelp_dataset/review.json") as f:
            for line in tqdm(f, total=line_count):
                blob = json.loads(line)
                user_ids += [blob["user_id"]]
                business_ids += [blob["business_id"]]
                stars += [blob["stars"]]
                dates += [blob["date"]]
                text+=[blob['text']]
                useful+=[blob['useful']]
                funny+=[blob['funny']]
                cool+=[blob['cool']]
        review = pd.DataFrame(
           {"user_id": user_ids, "business_id": business_ids, "rating_review": stars, "date_re
           'text_review':text,'useful_review':useful,'funny_review':funny,'cool_review':cool

           }
        )
        # user_counts = review["user_id"].value_counts()
        # active_users = user_counts.loc[user_counts >= 5].index.tolist()
```

```
100%|| 6685900/6685900 [03:05<00:00, 36072.32it/s]
```

Since we only want the users that rated at least 5 times, so we filter out inactive users.

```
In [0]: user_counts = review["user_id"].value_counts()
        active_users = user_counts.loc[user_counts >= 5].index.tolist()
```

```
In [0]: review=review.loc[review['user_id'].isin(active_users)]
```

```
In [0]: review.to_csv('review.csv',index=False)
```