Train_Test_Split

December 19, 2019

```
In [0]: import pandas as pd
        import numpy as np
        import time
```

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

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

0.0.1 In this notebook, we want to split the dataset into train and test set. We save the row index of the test set as CSV file so we can use them later in all the other tasks.

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

Read the review csv.

```
In [0]: review=pd.read_csv(path+'review.csv')
```

```
In [0]: review.head(2)
```

```
Out[0]:                  user_id             business_id  ...  funny_review cool_review
        0  hG7b0MtEbXx5QzbzE6C_VA  ujmEBvifdJM6h6RLv4wQIg  ...           1.0         0.0
        1  n6-Gk65cPZL6Uz8qRm3NYw  WTqjgwHlXbSFevF32_DJVw  ...           0.0         0.0

        [2 rows x 8 columns]
```

```
In [0]: del review['text_review']
```

```
In [0]: review.head(2)
```

```
Out[0]:                  user_id             business_id  ...  funny_review cool_review
        0  hG7b0MtEbXx5QzbzE6C_VA  ujmEBvifdJM6h6RLv4wQIg  ...           1.0         0.0
        1  n6-Gk65cPZL6Uz8qRm3NYw  WTqjgwHlXbSFevF32_DJVw  ...           0.0         0.0

        [2 rows x 7 columns]
```

```
In [0]: review['freq_business'] = review.groupby('business_id')['business_id'].transform('coun
```

```
In [0]: review2=review.loc[review['freq_business']>2]
        review2['freq_user'] = review2.groupby('user_id')['user_id'].transform('count')
        review3=review2.loc[review2['freq_user']>=5]
```

```
/usr/local/lib/python3.6/dist-packages/ipykernel_launcher.py:2: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: http://pandas.pydata.org/pandas-docs/stable/user_guide/i
```

Take the last rating of each user as test set.

```
In [0]: result5=review3.sort_values(by =['user_id', 'date_review'])
        test1=result5.drop_duplicates(['user_id'],keep='last')
```

```
In [0]: result5.head(6)
```

```
Out[0]:      index                 user_id  ... freq_business  freq_user
        0  3520769  ---1lKK3aKOuomHnwAkAow  ...           136        126
        1  4032237  ---1lKK3aKOuomHnwAkAow  ...           910        126
        2  2931695  ---1lKK3aKOuomHnwAkAow  ...            69        126
        3  1982498  ---1lKK3aKOuomHnwAkAow  ...           178        126
        4   550727  ---1lKK3aKOuomHnwAkAow  ...           741        126
        5  2377380  ---1lKK3aKOuomHnwAkAow  ...            21        126

        [6 rows x 10 columns]
```

```
In [0]: #test1=test1.reset_index()
        result5=result5.reset_index()
```

```
In [0]: train1=result5.loc[~result5['index'].isin(test1['index'])]
```

Extract another two ratings from each user randomly and use them as test set too, because we want to compare the predicted ranking with the true ranking of the businesses in the test set.

```
In [0]: test2=train1.groupby('user_id').apply(lambda x: x.sample(2,random_state=42))
```

```
In [0]: test2
```

```
Out[0]:                                    index  ... freq_user
        user_id                                   ...
        ---1lKK3aKOuomHnwAkAow 18        2978757  ...       126
                               42         852488  ...       126
        --0kuuLmuYBe3Rmu0Iycww 134       1888012  ...        11
                               127       4129878  ...        11
        --2HUmLkcNHZp0xw6AMBPg 189       3112157  ...        65
        ...                                  ...  ...       ...
        zzvV3l9IqTRX7Db8nxThbA 4462177    363369  ...         5
```

```
zzw0Z6-_VDp9ShIRSKIsQw 4462186  2633163  ...        10
                       4462180   782777  ...        10
zzxZoMmjbUjXcWZzrE3PIw 4462190  3199076  ...         6
                       4462193  1819230  ...         6

[562730 rows x 10 columns]
```

```
In [0]: test3=test2
        test3.index = test2.index.set_names(['Trial', 'measurement'])
```

```
In [0]: test3
```

```
Out[0]:                         Trial  measurement  ...  freq_business freq_user
        0       ---1lKK3aKOuomHnwAkAow           18  ...             69       126
        1       ---1lKK3aKOuomHnwAkAow           42  ...              7       126
        2       --0kuuLmuYBe3Rmu0Iycww          134  ...             15        11
        3       --0kuuLmuYBe3Rmu0Iycww          127  ...             22        11
        4       --2HUmLkcNHZp0xw6AMBPg          189  ...            140        65
        ...                        ...          ...  ...            ...       ...
        562725  zzvV3l9IqTRX7Db8nxThbA      4462177  ...            232         5
        562726  zzw0Z6-_VDp9ShIRSKIsQw      4462186  ...             70        10
        562727  zzw0Z6-_VDp9ShIRSKIsQw      4462180  ...            119        10
        562728  zzxZoMmjbUjXcWZzrE3PIw      4462190  ...            148         6
        562729  zzxZoMmjbUjXcWZzrE3PIw      4462193  ...            356         6

        [562730 rows x 12 columns]
```

```
In [0]: test3=test3.reset_index()
```

```
In [0]: all_test_idx=list(test3['index'])+list(test1['index'])
```

Save the index of the test set, since when only save the index, it would not take much space and we can use them later.

```
In [0]: all_test_idx_df=pd.DataFrame(all_test_idx)
        all_test_idx_df.to_csv('all_test_idx_df2.csv',index=False)
```