Adithya Kiran || Chirag P Tubakad || Arpit Agarwal || Nandakrishna
PES1201700231||PES1201700896||PES1201701084||PES12017010901

ASSIGNMENT 7 : Collaborative Filtering

Take a dataset with identifiers and use collaborative filtering to filter through the noise in the dataset and generate MSE for user and item datasets.

Dataset Details (number of items and users, utility matrix construction, and sparse nature of utility matrix)

```
n_users = df.user_id.unique().shape[0]
n_items = df.item_id.unique().shape[0]
print(str(n_users) + ' users')
print(str(n_items) + ' items')
```

```
943 users
1682 items
```

```
# Varaible ratings holds our user-item rating matrix. Similar to our utility matrix.

ratings = np.zeros((n_users, n_items))
for row in df.itertuples():
    ratings[row[1]-1, row[2]-1] = row[3]
ratings
```

```
array([[5., 3., 4., ..., 0., 0., 0.],
       [4., 0., 0., ..., 0., 0., 0.],
       [0., 0., 0., ..., 0., 0., 0.],
       ...,
       [5., 0., 0., ..., 0., 0., 0.],
       [0., 0., 0., ..., 0., 0., 0.],
       [0., 5., 0., ..., 0., 0., 0.]])
```

```
sparsity = float(len(ratings.nonzero()[0]))
sparsity /= (ratings.shape[0] * ratings.shape[1])
sparsity *= 100
print('Sparsity: {:4.2f}%'.format(sparsity))

#This gives us an idea of how sparse our matrix is. This means that 6.3% of the user-item ratings have a value.
```

```
Sparsity: 6.30%
```

Item similarity and user similarity calculations

```
def fast_similarity(ratings, kind='user', epsilon=1e-9):
    # epsilon -> small number for handling dived-by-zero errors
    if kind == 'user':
        sim = ratings.dot(ratings.T) + epsilon
    elif kind == 'item':
        sim = ratings.T.dot(ratings) + epsilon
    norms = np.array([np.sqrt(np.diagonal(sim))])
    return (sim / norms / norms.T)
```

```
user_similarity = fast_similarity(train, kind='user')
item_similarity = fast_similarity(train, kind='item')
# This matrix is actually very large , we are just printing first 4 rows and columns for representation.
print(item_similarity[:4 ,:4])
```

```
[[1.         0.38003054 0.33412145 0.45924272]
 [0.38003054 1.         0.25924382 0.46065394]
 [0.33412145 0.25924382 1.         0.33497463]
 [0.45924272 0.46065394 0.33497463 1.        ]]
```

MSE of user-user relation and item-item relation

```
User-based CF MSE: 8.454547772457058
Item-based CF MSE: 11.608851716966937
```