In [1]:

```
import numpy as np
import pandas as pd
```

In [2]:

```
names = ['user_id', 'item_id', 'rating', 'timestamp']
df =pd.read_csv('ml-100k/u.data', sep='\t', names=names)
df.head()
```

Out[2]:

	user_id	item_id	rating	timestamp
0	196	242	3	881250949
1	186	302	3	891717742
2	22	377	1	878887116
3	244	51	2	880606923
4	166	346	1	886397596

In [4]:

```
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
```

In [5]:

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

Out[5]:

```
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.]])
```

In [6]:

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

```
Sparsity: 6.30%
```

In [8]:

```
def train_test_split(ratings):
    test = np.zeros(ratings.shape)
    train = ratings.copy()
    for user in range(ratings.shape[0]):
        test_ratings = np.random.choice(ratings[0, :].nonzero()[0],
                                        size=10
                                        size=10,
                                        replace=False)
        train[user, test_ratings] = 0.
        test[user, test_ratings] = ratings[user, test_ratings]

    # Test and training are truly disjoint
    assert(np.all((train * test) == 0))
    return train, test

train, test = train_test_split(ratings)
```

In [9]:

```
def slow_similarity(ratings, kind='user'):
    if kind == 'user':
        axmax = 0
        axmin = 1
    elif kind == 'item':
        axmax = 1
        axmin = 0
    sim = np.zeros((ratings.shape[axmax], ratings.shape[axmax]))
    for u in range(ratings.shape[axmax]):
        for uprime in range(ratings.shape[axmax]):
            rui_sqrd = 0.
            ruprimei_sqrd = 0.
            for i in range(ratings.shape[axmin]):
                sim[u, uprime] = ratings[u, i] * ratings[uprime, i]
                rui_sqrd += ratings[u, i] ** 2
                ruprimei_sqrd += ratings[uprime, i] ** 2
            sim[u, uprime] /= rui_sqrd * ruprimei_sqrd
    return sim
```

In [10]:

```
def fast_similarity(ratings, kind='user'):
    if kind == 'user':
        sim = ratings.dot(ratings.T)
    elif kind == 'item':
        sim = ratings.T.dot(ratings)
    norms = np.array([np.sqrt(np.diagonal(sim))])
    return sim / norms / norms.T
```

In [11]:

```
#%timeit slow_user_similarity(train)
fast_similarity(train, kind='user')

user_similarity = fast_similarity(train, kind='user')
item_similarity = fast_similarity(train, kind='item')
print(item_similarity[:4, :4])
```

```
[[1.         0.38107207 0.32627046 0.4284299 ]
 [0.38107207 1.         0.25828144 0.48551038]
 [0.32627046 0.25828144 1.         0.30768179]
 [0.4284299  0.48551038 0.30768179 1.        ]]
```

In [12]:

```
def predict_slow_simple(ratings, similarity, kind='user'):
    pred = np.zeros(ratings.shape)
    if kind == 'user':
        for i in range(ratings.shape[0]):
            for j in range(ratings.shape[1]):
                pred[i, j] = similarity[i, :].dot(ratings[:, j])\
                             /np.sum(np.abs(similarity[i, :]))
        return pred
    elif kind == 'item':
        for i in range(ratings.shape[0]):
            for j in range(ratings.shape[1]):
                pred[i, j] = similarity[j, :].dot(ratings[i, :].T)\
                             /np.sum(np.abs(similarity[j, :]))

        return pred
```

In [13]:

```
def predict_fast_simple(ratings, similarity, kind='user'):
    if kind == 'user':
        return similarity.dot(ratings) / np.array([np.abs(similarity).sum(axis=1)]).T
    elif kind == 'item':
        return ratings.dot(similarity) / np.array([np.abs(similarity).sum(axis=1)])
```

In [14]:

```
from sklearn.metrics import mean_squared_error
```

In [15]:

```
def get_mse(pred, actual):
    # Ignore nonzero terms.
    pred = pred[actual.nonzero()].flatten()
    actual = actual[actual.nonzero()].flatten()
    return mean_squared_error(pred, actual)
```

In [17]:

```
item_prediction = predict_fast_simple(train, item_similarity, kind='item')
user_prediction = predict_fast_simple(train, user_similarity, kind='user')

print('User-based CF MSE: ' + str(get_mse(user_prediction, test)))
print('Item-based CF MSE: ' + str(get_mse(item_prediction, test)))
```

```
User-based CF MSE: 7.293962040310653
Item-based CF MSE: 9.958893513581314
```

In [18]:

```
def predict_topk(ratings, similarity, kind='user', k=40):
    pred = np.zeros(ratings.shape)
    if kind == 'user':
        for i in range(ratings.shape[0]):
            top_k_users = [np.argsort(similarity[:,i])[:-k-1:-1]]
            for j in range(ratings.shape[1]):
                pred[i, j] = similarity[i, :][top_k_users].dot(ratings[:, j][top_k_users])
                pred[i, j] /= np.sum(np.abs(similarity[i, :][top_k_users]))
    if kind == 'item':
        for j in range(ratings.shape[1]):
            top_k_items = [np.argsort(similarity[:,j])[:-k-1:-1]]
            for i in range(ratings.shape[0]):
                pred[i, j] = similarity[j, :][top_k_items].dot(ratings[i, :][top_k_items].T)
                pred[i, j] /= np.sum(np.abs(similarity[j, :][top_k_items]))

    return pred
```

In [19]:

```
pred = predict_topk(train, user_similarity, kind='user', k=40)
print('Top-k User-based CF MSE: ' + str(get_mse(pred, test)))

pred = predict_topk(train, item_similarity, kind='item', k=40)
print('Top-k Item-based CF MSE: ' + str(get_mse(pred, test)))
```

```
F:\CONDA\lib\site-packages\ipykernel_launcher.py:7: FutureWarning: Using a non-tuple sequence for
multidimensional indexing is deprecated; use `arr[tuple(seq)]` instead of `arr[seq]`. In the futur
e this will be interpreted as an array index, `arr[np.array(seq)]`, which will result either in an
error or a different result.
  import sys
F:\CONDA\lib\site-packages\ipykernel_launcher.py:8: FutureWarning: Using a non-tuple sequence for
multidimensional indexing is deprecated; use `arr[tuple(seq)]` instead of `arr[seq]`. In the futur
e this will be interpreted as an array index, `arr[np.array(seq)]`, which will result either in an
```

```
error or a different result.
```

```
Top-k User-based CF MSE: 4.042916821517707
```

```
F:\CONDA\lib\site-packages\ipykernel_launcher.py:13: FutureWarning: Using a non-tuple sequence for
multidimensional indexing is deprecated; use `arr[tuple(seq)]` instead of `arr[seq]`. In the futur
e this will be interpreted as an array index, `arr[np.array(seq)]`, which will result either in an
error or a different result.
  del sys.path[0]
F:\CONDA\lib\site-packages\ipykernel_launcher.py:14: FutureWarning: Using a non-tuple sequence for
multidimensional indexing is deprecated; use `arr[tuple(seq)]` instead of `arr[seq]`. In the futur
e this will be interpreted as an array index, `arr[np.array(seq)]`, which will result either in an
error or a different result.
```

```
Top-k Item-based CF MSE: 4.427883840161299
```

In [20]:

```
k_array = [5, 15, 30, 50, 100, 200]
user_train_mse = []
user_test_mse = []
item_test_mse = []
item_train_mse = []

def get_mse(pred, actual):
    pred = pred[actual.nonzero()].flatten()
    actual = actual[actual.nonzero()].flatten()
    return mean_squared_error(pred, actual)

for k in k_array:
    user_pred = predict_topk(train, user_similarity, kind='user', k=k)
    item_pred = predict_topk(train, item_similarity, kind='item', k=k)

    user_train_mse += [get_mse(user_pred, train)]
    user_test_mse += [get_mse(user_pred, test)]

    item_train_mse += [get_mse(item_pred, train)]
    item_test_mse += [get_mse(item_pred, test)]
```

```
F:\CONDA\lib\site-packages\ipykernel_launcher.py:7: FutureWarning: Using a non-tuple sequence for
multidimensional indexing is deprecated; use `arr[tuple(seq)]` instead of `arr[seq]`. In the futur
e this will be interpreted as an array index, `arr[np.array(seq)]`, which will result either in an
error or a different result.
  import sys
F:\CONDA\lib\site-packages\ipykernel_launcher.py:8: FutureWarning: Using a non-tuple sequence for
multidimensional indexing is deprecated; use `arr[tuple(seq)]` instead of `arr[seq]`. In the futur
e this will be interpreted as an array index, `arr[np.array(seq)]`, which will result either in an
error or a different result.

F:\CONDA\lib\site-packages\ipykernel_launcher.py:13: FutureWarning: Using a non-tuple sequence for
multidimensional indexing is deprecated; use `arr[tuple(seq)]` instead of `arr[seq]`. In the futur
e this will be interpreted as an array index, `arr[np.array(seq)]`, which will result either in an
error or a different result.
  del sys.path[0]
F:\CONDA\lib\site-packages\ipykernel_launcher.py:14: FutureWarning: Using a non-tuple sequence for
multidimensional indexing is deprecated; use `arr[tuple(seq)]` instead of `arr[seq]`. In the futur
e this will be interpreted as an array index, `arr[np.array(seq)]`, which will result either in an
error or a different result.

F:\CONDA\lib\site-packages\ipykernel_launcher.py:7: FutureWarning: Using a non-tuple sequence for
multidimensional indexing is deprecated; use `arr[tuple(seq)]` instead of `arr[seq]`. In the futur
e this will be interpreted as an array index, `arr[np.array(seq)]`, which will result either in an
error or a different result.
  import sys
F:\CONDA\lib\site-packages\ipykernel_launcher.py:8: FutureWarning: Using a non-tuple sequence for
multidimensional indexing is deprecated; use `arr[tuple(seq)]` instead of `arr[seq]`. In the futur
e this will be interpreted as an array index, `arr[np.array(seq)]`, which will result either in an
error or a different result.

F:\CONDA\lib\site-packages\ipykernel_launcher.py:13: FutureWarning: Using a non-tuple sequence for
multidimensional indexing is deprecated; use `arr[tuple(seq)]` instead of `arr[seq]`. In the futur
```

```
e this will be interpreted as an array index, `arr[np.array(seq)]`, which will result either in an
error or a different result.
  del sys.path[0]
F:\CONDA\lib\site-packages\ipykernel_launcher.py:14: FutureWarning: Using a non-tuple sequence for
multidimensional indexing is deprecated; use `arr[tuple(seq)]` instead of `arr[seq]`. In the futur
e this will be interpreted as an array index, `arr[np.array(seq)]`, which will result either in an
error or a different result.

F:\CONDA\lib\site-packages\ipykernel_launcher.py:7: FutureWarning: Using a non-tuple sequence for
multidimensional indexing is deprecated; use `arr[tuple(seq)]` instead of `arr[seq]`. In the futur
e this will be interpreted as an array index, `arr[np.array(seq)]`, which will result either in an
error or a different result.
  import sys
F:\CONDA\lib\site-packages\ipykernel_launcher.py:8: FutureWarning: Using a non-tuple sequence for
multidimensional indexing is deprecated; use `arr[tuple(seq)]` instead of `arr[seq]`. In the futur
e this will be interpreted as an array index, `arr[np.array(seq)]`, which will result either in an
error or a different result.

F:\CONDA\lib\site-packages\ipykernel_launcher.py:13: FutureWarning: Using a non-tuple sequence for
multidimensional indexing is deprecated; use `arr[tuple(seq)]` instead of `arr[seq]`. In the futur
e this will be interpreted as an array index, `arr[np.array(seq)]`, which will result either in an
error or a different result.
  del sys.path[0]
F:\CONDA\lib\site-packages\ipykernel_launcher.py:14: FutureWarning: Using a non-tuple sequence for
multidimensional indexing is deprecated; use `arr[tuple(seq)]` instead of `arr[seq]`. In the futur
e this will be interpreted as an array index, `arr[np.array(seq)]`, which will result either in an
error or a different result.

F:\CONDA\lib\site-packages\ipykernel_launcher.py:7: FutureWarning: Using a non-tuple sequence for
multidimensional indexing is deprecated; use `arr[tuple(seq)]` instead of `arr[seq]`. In the futur
e this will be interpreted as an array index, `arr[np.array(seq)]`, which will result either in an
error or a different result.
  import sys
F:\CONDA\lib\site-packages\ipykernel_launcher.py:8: FutureWarning: Using a non-tuple sequence for
multidimensional indexing is deprecated; use `arr[tuple(seq)]` instead of `arr[seq]`. In the futur
e this will be interpreted as an array index, `arr[np.array(seq)]`, which will result either in an
error or a different result.

F:\CONDA\lib\site-packages\ipykernel_launcher.py:13: FutureWarning: Using a non-tuple sequence for
multidimensional indexing is deprecated; use `arr[tuple(seq)]` instead of `arr[seq]`. In the futur
e this will be interpreted as an array index, `arr[np.array(seq)]`, which will result either in an
error or a different result.
  del sys.path[0]
F:\CONDA\lib\site-packages\ipykernel_launcher.py:14: FutureWarning: Using a non-tuple sequence for
multidimensional indexing is deprecated; use `arr[tuple(seq)]` instead of `arr[seq]`. In the futur
e this will be interpreted as an array index, `arr[np.array(seq)]`, which will result either in an
error or a different result.

F:\CONDA\lib\site-packages\ipykernel_launcher.py:7: FutureWarning: Using a non-tuple sequence for
multidimensional indexing is deprecated; use `arr[tuple(seq)]` instead of `arr[seq]`. In the futur
e this will be interpreted as an array index, `arr[np.array(seq)]`, which will result either in an
error or a different result.
  import sys
F:\CONDA\lib\site-packages\ipykernel_launcher.py:8: FutureWarning: Using a non-tuple sequence for
multidimensional indexing is deprecated; use `arr[tuple(seq)]` instead of `arr[seq]`. In the futur
e this will be interpreted as an array index, `arr[np.array(seq)]`, which will result either in an
error or a different result.

F:\CONDA\lib\site-packages\ipykernel_launcher.py:13: FutureWarning: Using a non-tuple sequence for
multidimensional indexing is deprecated; use `arr[tuple(seq)]` instead of `arr[seq]`. In the futur
e this will be interpreted as an array index, `arr[np.array(seq)]`, which will result either in an
error or a different result.
  del sys.path[0]
F:\CONDA\lib\site-packages\ipykernel_launcher.py:14: FutureWarning: Using a non-tuple sequence for
multidimensional indexing is deprecated; use `arr[tuple(seq)]` instead of `arr[seq]`. In the futur
e this will be interpreted as an array index, `arr[np.array(seq)]`, which will result either in an
error or a different result.

F:\CONDA\lib\site-packages\ipykernel_launcher.py:7: FutureWarning: Using a non-tuple sequence for
multidimensional indexing is deprecated; use `arr[tuple(seq)]` instead of `arr[seq]`. In the futur
e this will be interpreted as an array index, `arr[np.array(seq)]`, which will result either in an
error or a different result.
  import sys
F:\CONDA\lib\site-packages\ipykernel_launcher.py:8: FutureWarning: Using a non-tuple sequence for
multidimensional indexing is deprecated; use `arr[tuple(seq)]` instead of `arr[seq]`. In the futur
e this will be interpreted as an array index, `arr[np.array(seq)]`, which will result either in an
error or a different result.
```

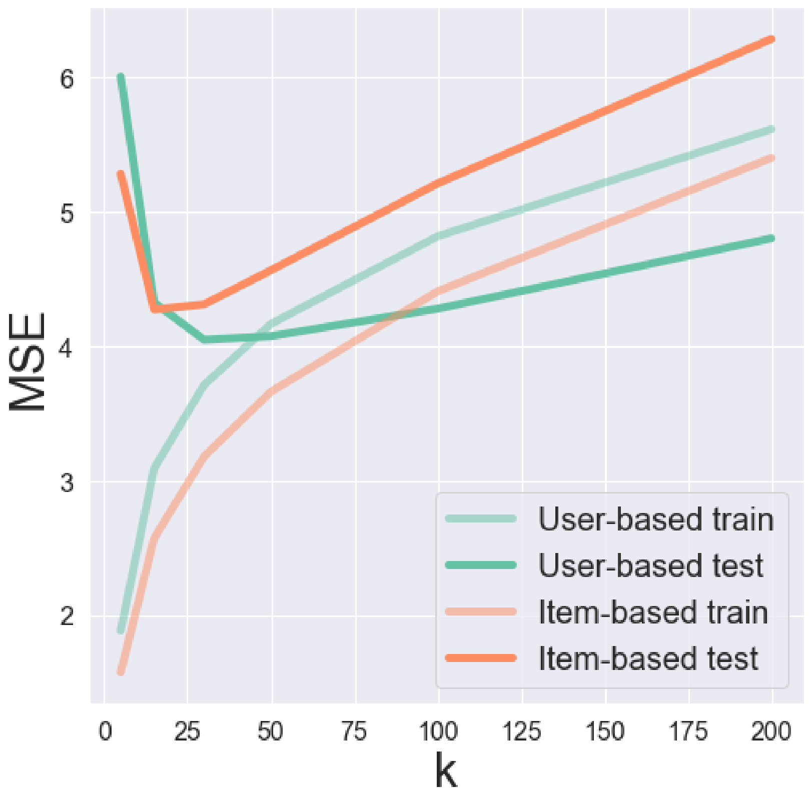
```
F:\CONDA\lib\site-packages\ipykernel_launcher.py:13: FutureWarning: Using a non-tuple sequence for
multidimensional indexing is deprecated; use `arr[tuple(seq)]` instead of `arr[seq]`. In the futur
e this will be interpreted as an array index, `arr[np.array(seq)]`, which will result either in an
error or a different result.
  del sys.path[0]
F:\CONDA\lib\site-packages\ipykernel_launcher.py:14: FutureWarning: Using a non-tuple sequence for
multidimensional indexing is deprecated; use `arr[tuple(seq)]` instead of `arr[seq]`. In the futur
e this will be interpreted as an array index, `arr[np.array(seq)]`, which will result either in an
error or a different result.
```

In [21]:

```python
import matplotlib.pyplot as plt
import seaborn as sns
sns.set()

pal = sns.color_palette("Set2", 2)

plt.figure(figsize=(8, 8))
plt.plot(k_array, user_train_mse, c=pal[0], label='User-based train', alpha=0.5, linewidth=5)
plt.plot(k_array, user_test_mse, c=pal[0], label='User-based test', linewidth=5)
plt.plot(k_array, item_train_mse, c=pal[1], label='Item-based train', alpha=0.5, linewidth=5)
plt.plot(k_array, item_test_mse, c=pal[1], label='Item-based test', linewidth=5)
plt.legend(loc='best', fontsize=20)
plt.xticks(fontsize=16);
plt.yticks(fontsize=16);
plt.xlabel('k', fontsize=30);
plt.ylabel('MSE', fontsize=30);
```



In [22]:

```python
def predict_nobias(ratings, similarity, kind='user'):
    if kind == 'user':
        user_bias = ratings.mean(axis=1)
        ratings = (ratings - user_bias[:, np.newaxis]).copy()
        pred = similarity.dot(ratings) / np.array([np.abs(similarity).sum(axis=1)]).T
        pred += user_bias[:, np.newaxis]
    elif kind == 'item':
        item_bias = ratings.mean(axis=0)
        ratings = (ratings - item_bias[np.newaxis, :]).copy()
        pred = ratings.dot(similarity) / np.array([np.abs(similarity).sum(axis=1)])
        pred += item_bias[np.newaxis, :]

    return pred

user_pred = predict_nobias(train, user_similarity, kind='user')
print('Bias-subtracted User-based CF MSE: ' + str(get_mse(user_pred, test)))

item_pred = predict_nobias(train, item_similarity, kind='item')
print('Bias-subtracted Item-based CF MSE: ' + str(get_mse(item_pred, test)))
```

```
Bias-subtracted User-based CF MSE: 6.821500672783424
Bias-subtracted Item-based CF MSE: 7.334858113878549
```

In [23]:

```
def predict_topk_nobias(ratings, similarity, kind='user', k=40):
    pred = np.zeros(ratings.shape)
    if kind == 'user':
        user_bias = ratings.mean(axis=1)
        ratings = (ratings - user_bias[:, np.newaxis]).copy()
        for i in range(ratings.shape[0]):
            top_k_users = [np.argsort(similarity[:,i])[:-k-1:-1]]
            for j in range(ratings.shape[1]):
                pred[i, j] = similarity[i, :][top_k_users].dot(ratings[:, j][top_k_users])
                pred[i, j] /= np.sum(np.abs(similarity[i, :][top_k_users]))
        pred += user_bias[:, np.newaxis]
    if kind == 'item':
        item_bias = ratings.mean(axis=0)
        ratings = (ratings - item_bias[np.newaxis, :]).copy()
        for j in range(ratings.shape[1]):
            top_k_items = [np.argsort(similarity[:,j])[:-k-1:-1]]
            for i in range(ratings.shape[0]):
                pred[i, j] = similarity[j, :][top_k_items].dot(ratings[i, :][top_k_items].T)
                pred[i, j] /= np.sum(np.abs(similarity[j, :][top_k_items]))
        pred += item_bias[np.newaxis, :]

    return pred
```

In [24]:

```
k_array = [5, 15, 30, 50, 100, 200]
user_train_mse = []
user_test_mse = []
item_test_mse = []
item_train_mse = []

for k in k_array:
    user_pred = predict_topk_nobias(train, user_similarity, kind='user', k=k)
    item_pred = predict_topk_nobias(train, item_similarity, kind='item', k=k)

    user_train_mse += [get_mse(user_pred, train)]
    user_test_mse += [get_mse(user_pred, test)]

    item_train_mse += [get_mse(item_pred, train)]
    item_test_mse += [get_mse(item_pred, test)]
```

```
F:\CONDA\lib\site-packages\ipykernel_launcher.py:9: FutureWarning: Using a non-tuple sequence for
multidimensional indexing is deprecated; use `arr[tuple(seq)]` instead of `arr[seq]`. In the futur
e this will be interpreted as an array index, `arr[np.array(seq)]`, which will result either in an
error or a different result.
  if __name__ == '__main__':
F:\CONDA\lib\site-packages\ipykernel_launcher.py:10: FutureWarning: Using a non-tuple sequence for
multidimensional indexing is deprecated; use `arr[tuple(seq)]` instead of `arr[seq]`. In the futur
e this will be interpreted as an array index, `arr[np.array(seq)]`, which will result either in an
error or a different result.
  # Remove the CWD from sys.path while we load stuff.
F:\CONDA\lib\site-packages\ipykernel_launcher.py:18: FutureWarning: Using a non-tuple sequence for
multidimensional indexing is deprecated; use `arr[tuple(seq)]` instead of `arr[seq]`. In the futur
e this will be interpreted as an array index, `arr[np.array(seq)]`, which will result either in an
error or a different result.
F:\CONDA\lib\site-packages\ipykernel_launcher.py:19: FutureWarning: Using a non-tuple sequence for
multidimensional indexing is deprecated; use `arr[tuple(seq)]` instead of `arr[seq]`. In the futur
e this will be interpreted as an array index, `arr[np.array(seq)]`, which will result either in an
error or a different result.
```

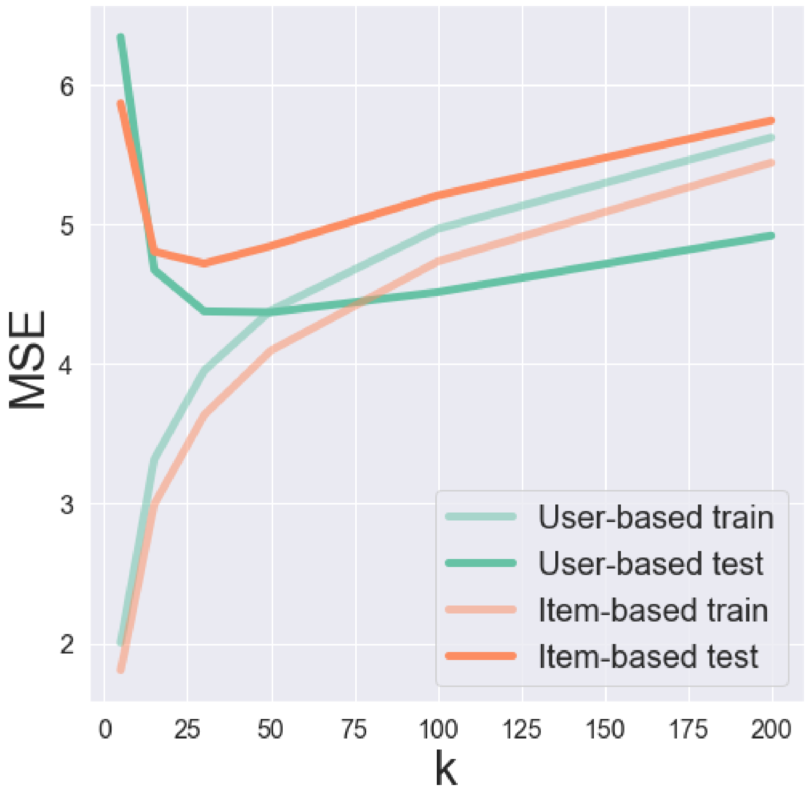
```
F:\CONDA\lib\site-packages\ipykernel_launcher.py:9: FutureWarning: Using a non-tuple sequence for
multidimensional indexing is deprecated; use `arr[tuple(seq)]` instead of `arr[seq]`. In the futur
e this will be interpreted as an array index, `arr[np.array(seq)]`, which will result either in an
error or a different result.
  if __name__ == '__main__':
F:\CONDA\lib\site-packages\ipykernel_launcher.py:10: FutureWarning: Using a non-tuple sequence for
multidimensional indexing is deprecated; use `arr[tuple(seq)]` instead of `arr[seq]`. In the futur
e this will be interpreted as an array index, `arr[np.array(seq)]`, which will result either in an
error or a different result.
  # Remove the CWD from sys.path while we load stuff.
F:\CONDA\lib\site-packages\ipykernel_launcher.py:18: FutureWarning: Using a non-tuple sequence for
multidimensional indexing is deprecated; use `arr[tuple(seq)]` instead of `arr[seq]`. In the futur
e this will be interpreted as an array index, `arr[np.array(seq)]`, which will result either in an
error or a different result.
F:\CONDA\lib\site-packages\ipykernel_launcher.py:19: FutureWarning: Using a non-tuple sequence for
multidimensional indexing is deprecated; use `arr[tuple(seq)]` instead of `arr[seq]`. In the futur
e this will be interpreted as an array index, `arr[np.array(seq)]`, which will result either in an
error or a different result.
F:\CONDA\lib\site-packages\ipykernel_launcher.py:9: FutureWarning: Using a non-tuple sequence for
multidimensional indexing is deprecated; use `arr[tuple(seq)]` instead of `arr[seq]`. In the futur
e this will be interpreted as an array index, `arr[np.array(seq)]`, which will result either in an
error or a different result.
  if __name__ == '__main__':
F:\CONDA\lib\site-packages\ipykernel_launcher.py:10: FutureWarning: Using a non-tuple sequence for
multidimensional indexing is deprecated; use `arr[tuple(seq)]` instead of `arr[seq]`. In the futur
e this will be interpreted as an array index, `arr[np.array(seq)]`, which will result either in an
error or a different result.
  # Remove the CWD from sys.path while we load stuff.
F:\CONDA\lib\site-packages\ipykernel_launcher.py:18: FutureWarning: Using a non-tuple sequence for
multidimensional indexing is deprecated; use `arr[tuple(seq)]` instead of `arr[seq]`. In the futur
e this will be interpreted as an array index, `arr[np.array(seq)]`, which will result either in an
error or a different result.
F:\CONDA\lib\site-packages\ipykernel_launcher.py:19: FutureWarning: Using a non-tuple sequence for
multidimensional indexing is deprecated; use `arr[tuple(seq)]` instead of `arr[seq]`. In the futur
e this will be interpreted as an array index, `arr[np.array(seq)]`, which will result either in an
error or a different result.
F:\CONDA\lib\site-packages\ipykernel_launcher.py:9: FutureWarning: Using a non-tuple sequence for
multidimensional indexing is deprecated; use `arr[tuple(seq)]` instead of `arr[seq]`. In the futur
e this will be interpreted as an array index, `arr[np.array(seq)]`, which will result either in an
error or a different result.
  if __name__ == '__main__':
F:\CONDA\lib\site-packages\ipykernel_launcher.py:10: FutureWarning: Using a non-tuple sequence for
multidimensional indexing is deprecated; use `arr[tuple(seq)]` instead of `arr[seq]`. In the futur
e this will be interpreted as an array index, `arr[np.array(seq)]`, which will result either in an
error or a different result.
  # Remove the CWD from sys.path while we load stuff.
F:\CONDA\lib\site-packages\ipykernel_launcher.py:18: FutureWarning: Using a non-tuple sequence for
multidimensional indexing is deprecated; use `arr[tuple(seq)]` instead of `arr[seq]`. In the futur
e this will be interpreted as an array index, `arr[np.array(seq)]`, which will result either in an
error or a different result.
F:\CONDA\lib\site-packages\ipykernel_launcher.py:19: FutureWarning: Using a non-tuple sequence for
multidimensional indexing is deprecated; use `arr[tuple(seq)]` instead of `arr[seq]`. In the futur
e this will be interpreted as an array index, `arr[np.array(seq)]`, which will result either in an
error or a different result.
F:\CONDA\lib\site-packages\ipykernel_launcher.py:9: FutureWarning: Using a non-tuple sequence for
multidimensional indexing is deprecated; use `arr[tuple(seq)]` instead of `arr[seq]`. In the futur
e this will be interpreted as an array index, `arr[np.array(seq)]`, which will result either in an
error or a different result.
  if __name__ == '__main__':
F:\CONDA\lib\site-packages\ipykernel_launcher.py:10: FutureWarning: Using a non-tuple sequence for
multidimensional indexing is deprecated; use `arr[tuple(seq)]` instead of `arr[seq]`. In the futur
e this will be interpreted as an array index, `arr[np.array(seq)]`, which will result either in an
error or a different result.
  # Remove the CWD from sys.path while we load stuff.
F:\CONDA\lib\site-packages\ipykernel_launcher.py:18: FutureWarning: Using a non-tuple sequence for
multidimensional indexing is deprecated; use `arr[tuple(seq)]` instead of `arr[seq]`. In the futur
e this will be interpreted as an array index, `arr[np.array(seq)]`, which will result either in an
error or a different result.
F:\CONDA\lib\site-packages\ipykernel_launcher.py:19: FutureWarning: Using a non-tuple sequence for
multidimensional indexing is deprecated; use `arr[tuple(seq)]` instead of `arr[seq]`. In the futur
e this will be interpreted as an array index, `arr[np.array(seq)]`, which will result either in an
error or a different result.
F:\CONDA\lib\site-packages\ipykernel_launcher.py:9: FutureWarning: Using a non-tuple sequence for
multidimensional indexing is deprecated; use `arr[tuple(seq)]` instead of `arr[seq]`. In the futur
e this will be interpreted as an array index, `arr[np.array(seq)]`, which will result either in an
error or a different result.
  if __name__ == '__main__':
```

```
F:\CONDA\lib\site-packages\ipykernel_launcher.py:10: FutureWarning: Using a non-tuple sequence for
multidimensional indexing is deprecated; use `arr[tuple(seq)]` instead of `arr[seq]`. In the futur
e this will be interpreted as an array index, `arr[np.array(seq)]`, which will result either in an
error or a different result.
  # Remove the CWD from sys.path while we load stuff.
F:\CONDA\lib\site-packages\ipykernel_launcher.py:18: FutureWarning: Using a non-tuple sequence for
multidimensional indexing is deprecated; use `arr[tuple(seq)]` instead of `arr[seq]`. In the futur
e this will be interpreted as an array index, `arr[np.array(seq)]`, which will result either in an
error or a different result.
F:\CONDA\lib\site-packages\ipykernel_launcher.py:19: FutureWarning: Using a non-tuple sequence for
multidimensional indexing is deprecated; use `arr[tuple(seq)]` instead of `arr[seq]`. In the futur
e this will be interpreted as an array index, `arr[np.array(seq)]`, which will result either in an
error or a different result.
```

In [25]:

```
pal = sns.color_palette("Set2", 2)

plt.figure(figsize=(8, 8))
plt.plot(k_array, user_train_mse, c=pal[0], label='User-based train', alpha=0.5, linewidth=5)
plt.plot(k_array, user_test_mse, c=pal[0], label='User-based test', linewidth=5)
plt.plot(k_array, item_train_mse, c=pal[1], label='Item-based train', alpha=0.5, linewidth=5)
plt.plot(k_array, item_test_mse, c=pal[1], label='Item-based test', linewidth=5)
plt.legend(loc='best', fontsize=20)
plt.xticks(fontsize=16);
plt.yticks(fontsize=16);
plt.xlabel('k', fontsize=30);
plt.ylabel('MSE', fontsize=30);
```



In [ ]:

```
import requests
import json

response = requests.get('http://us.imdb.com/M/title-exact?Toy%20Story%20(1995)')
print(response.url.split('/')[-2])

# Get base url filepath structure. w185 corresponds to size of movie poster.
headers = {'Accept': 'application/json'}
payload = {'api_key': 'Plz insert your key here '}
response = requests.get("http://api.themoviedb.org/3/configuration", params=payload, headers=header
s)
response = json.loads(response.text)
base_url = response['images']['base_url'] + 'w185'
```

In [28]:

```
def get_poster(imdb_url, base_url):
    # Get IMDB movie ID
    response = requests.get(imdb_url)
    movie_id = response.url.split('/')[-2]

    # Query themoviedb.org API for movie poster path.
    movie_url = 'http://api.themoviedb.org/3/movie/{:}/images'.format(movie_id)
    headers = {'Accept': 'application/json'}
    payload = {'api_key': 'INSERT API_KEY HERE'}
    response = requests.get(movie_url, params=payload, headers=headers)
    try:
        file_path = json.loads(response.text)['posters'][0]['file_path']
    except:
        # IMDB movie ID is sometimes no good. Need to get correct one.
        movie_title = imdb_url.split('?')[-1].split('(')[0]
        payload['query'] = movie_title
        response = requests.get('http://api.themoviedb.org/3/search/movie', params=payload, headers
=headers)
        movie_id = json.loads(response.text)['results'][0]['id']
        payload.pop('query', None)
        movie_url = 'http://api.themoviedb.org/3/movie/{:}/images'.format(movie_id)
        response = requests.get(movie_url, params=payload, headers=headers)
        file_path = json.loads(response.text)['posters'][0]['file_path']

    return base_url + file_path
```

In [30]:

```
toy_story = 'http://us.imdb.com/M/title-exact?Toy%20Story%20(1995)'

# Load in movie data
idx_to_movie = {}
with open('ml-100k/u.item', 'r') as f:
    for line in f.readlines():
        info = line.split('|')
        idx_to_movie[int(info[0])-1] = info[4]

def top_k_movies(similarity, mapper, movie_idx, k=6):
    return [mapper[x] for x in np.argsort(similarity[movie_idx,:])[:-k-1:-1]]

idx = 0 # Toy Story
movies = top_k_movies(item_similarity, idx_to_movie, idx)
```

In [31]:

```
print(movies)
```

```
['http://us.imdb.com/M/title-exact?Toy%20Story%20(1995)', 'http://us.imdb.com/M/title-exact?
Star%20Wars%20(1977)', 'http://us.imdb.com/M/title-exact?Independence%20Day%20(1996)',
'http://us.imdb.com/M/title-exact?Return%20of%20the%20Jedi%20(1983)', 'http://us.imdb.com/M/title-
exact?Mission:%20Impossible%20(1996)', 'http://us.imdb.com/M/title-exact?Rock,%20The%20(1996)']
```

Conclusion

Recommended movies on basis of Toy Story(1995) :

- Star Wars(1997)
- Independence Day(1996)
- Return of Jedi(1983)
- Mission Impossible (1996)
- Rock (1996)

In [ ]:

Similarity Calculation

In [33]:

```
import pandas as pd
import scipy
import sklearn

r_cols=['user_id','movie_id','rating']
ratings=pd.read_csv('ml-latest/ratings.csv',names=r_cols,usecols=range(3))
ratings.head()
```

Out[33]:

	user_id	movie_id	rating
0	userId	movieId	rating
1	1	1	4.0
2	1	3	4.0
3	1	6	4.0
4	1	47	5.0

In [34]:

```
m_cols=['movie_id','title']
movies=pd.read_csv('ml-latest/movies.csv',names=m_cols,usecols=range(2))
movies.head()
```

Out[34]:

	movie_id	title
0	movieId	title
1	1	Toy Story (1995)
2	2	Jumanji (1995)
3	3	Grumpier Old Men (1995)
4	4	Waiting to Exhale (1995)

In [35]:

```
ratings=pd.merge(movies,ratings)
ratings.head()
```

Out[35]:

	movie_id	title	user_id	rating
0	movieId	title	userId	rating
1	1	Toy Story (1995)	1	4.0
2	1	Toy Story (1995)	5	4.0
3	1	Toy Story (1995)	7	4.5
4	1	Toy Story (1995)	15	2.5

In [36]:

```
movieRatings=ratings.pivot_table(index=['user_id'],columns=['title'],values='rating',aggfunc =
lambda x: x)
movieRatings.head()
```

Out[36]:

title	'71 (2014)	'Hellboy': The Seeds of Creation (2004)	'Round Midnight (1986)	'Salem's Lot (2004)	'Til There Was You (1997)	'Tis the Season for Love (2015)	'burbs, The (1989)	'night Mother (1986)	(500) Days of Summer (2009)	*batteries not included (1987)	...	[REC] (2007)	[REC]² (2009)	[REC]³ 3 Génesis (2012)	anoh Flc We That - M (2
user_id															
1	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	...	NaN	NaN	NaN	
10	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	...	NaN	NaN	NaN	
100	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	...	NaN	NaN	NaN	
101	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	...	NaN	NaN	NaN	
102	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	...	NaN	NaN	NaN	

5 rows × 9720 columns

```
In [ ]:
```