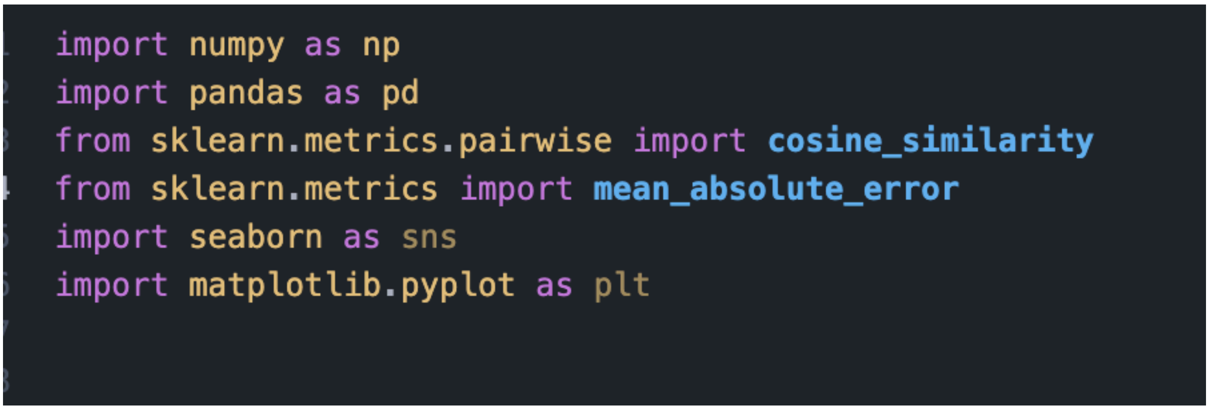
CF ASSIGNMENT

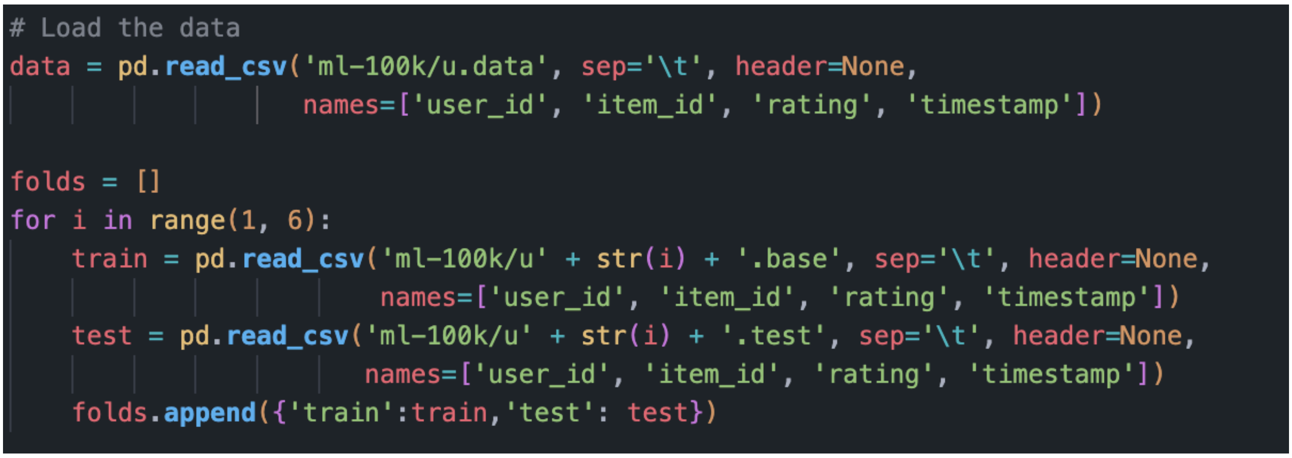
Name: Aayush Gakhar
Roll no: 2020006

Step 1: Loading required libraries



```python
import numpy as np
import pandas as pd
from sklearn.metrics.pairwise import cosine_similarity
from sklearn.metrics import mean_absolute_error
import seaborn as sns
import matplotlib.pyplot as plt
```

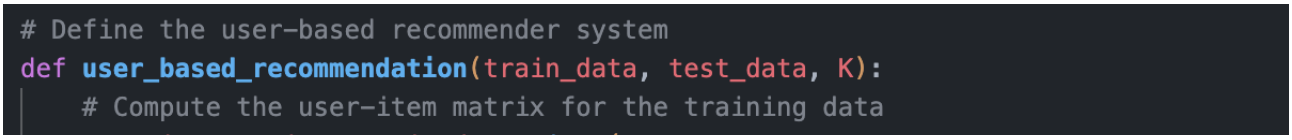
Step 2: Loading data



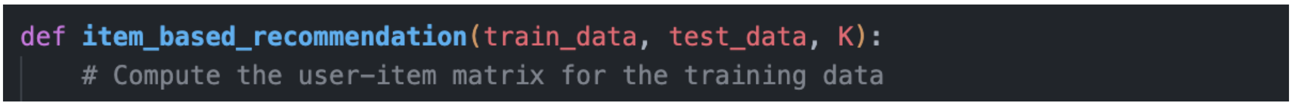
```python
# Load the data
data = pd.read_csv('ml-100k/u.data', sep='\t', header=None,
                   names=['user_id', 'item_id', 'rating', 'timestamp'])

folds = []
for i in range(1, 6):
    train = pd.read_csv('ml-100k/u' + str(i) + '.base', sep='\t', header=None,
                        names=['user_id', 'item_id', 'rating', 'timestamp'])
    test = pd.read_csv('ml-100k/u' + str(i) + '.test', sep='\t', header=None,
                       names=['user_id', 'item_id', 'rating', 'timestamp'])
    folds.append({'train':train,'test': test})
```

Step 3: Made functions to calculate the recommendations mae

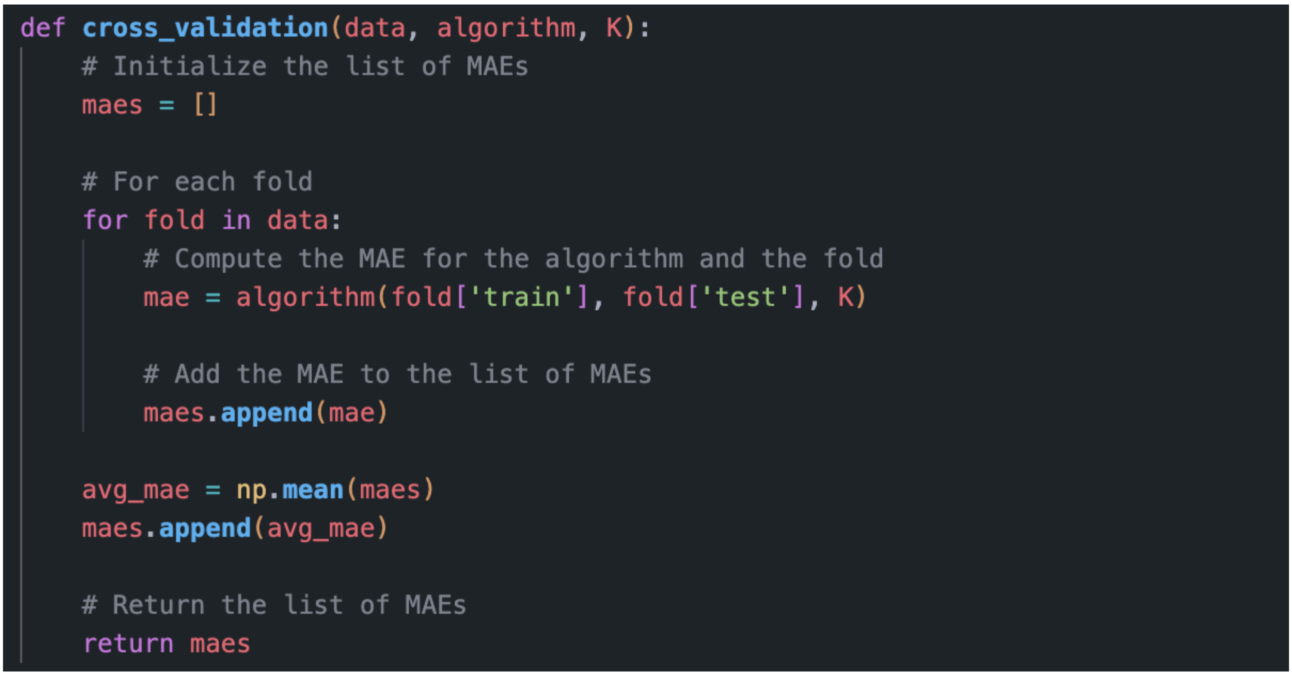


```python
# Define the user-based recommender system
def user_based_recommendation(train_data, test_data, K):
    # Compute the user-item matrix for the training data
```



```python
def item_based_recommendation(train_data, test_data, K):
    # Compute the user-item matrix for the training data
```

Step 4: Iterated over all folds and K values



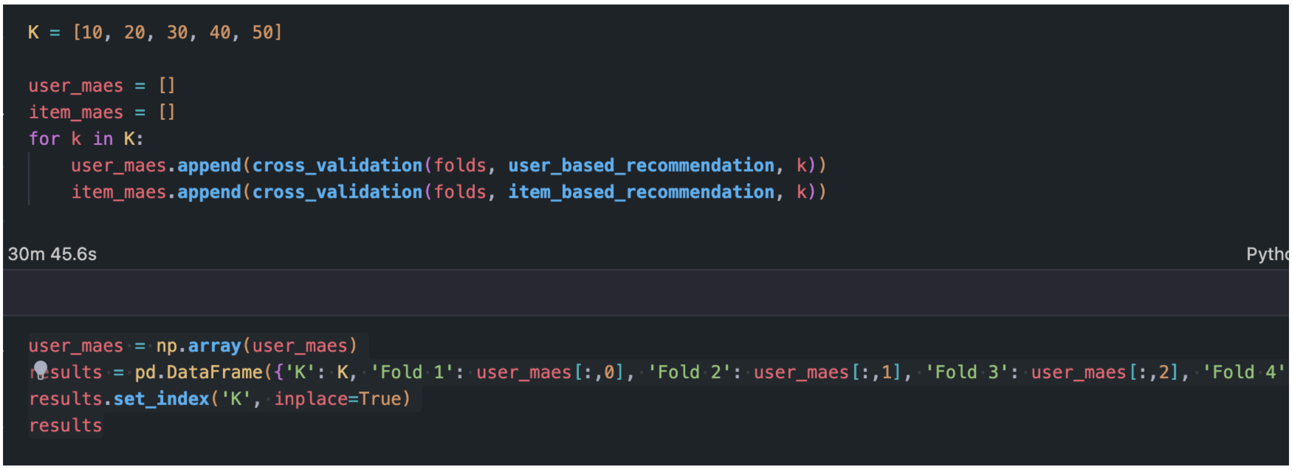
```
def cross_validation(data, algorithm, K):
    # Initialize the list of MAEs
    maes = []

    # For each fold
    for fold in data:
        # Compute the MAE for the algorithm and the fold
        mae = algorithm(fold['train'], fold['test'], K)

        # Add the MAE to the list of MAEs
        maes.append(mae)

    avg_mae = np.mean(maes)
    maes.append(avg_mae)

    # Return the list of MAEs
    return maes
```



```
K = [10, 20, 30, 40, 50]

user_maes = []
item_maes = []
for k in K:
    user_maes.append(cross_validation(folds, user_based_recommendation, k))
    item_maes.append(cross_validation(folds, item_based_recommendation, k))
```

30m 45.6s

Pytho

```
user_maes = np.array(user_maes)
results = pd.DataFrame({'K': K, 'Fold 1': user_maes[:,0], 'Fold 2': user_maes[:,1], 'Fold 3': user_maes[:,2], 'Fold 4'
results.set_index('K', inplace=True)
results
```

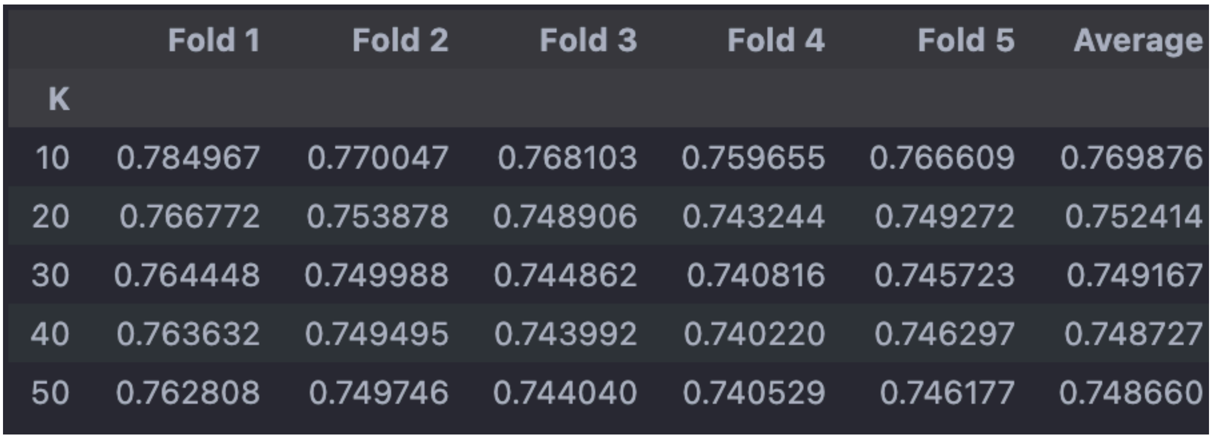
Cosine Similarity

$$w(a,i) = \sum_j \frac{v_{a,j}}{\sqrt{\sum_{k\in I_a} v_{a,k}^2}} \frac{v_{i,j}}{\sqrt{\sum_{k\in I_i} v_{i,k}^2}}$$

Formula to get the predicted value

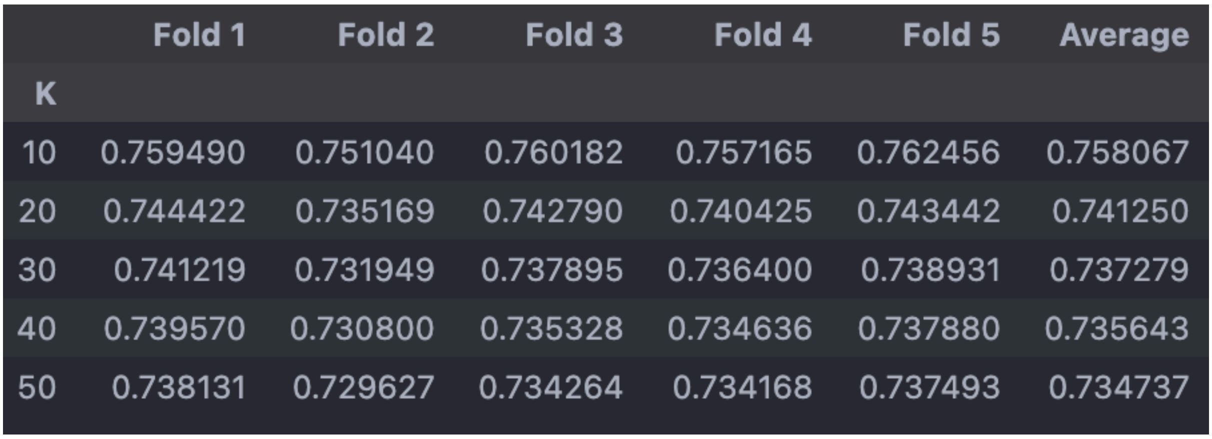
$$p_{a,j} = \overline{v}_a + \sum_{i=1}^{n} \frac{w_{a,i}(v_{i,j} - \overline{v}_i)}{\sum_i w_{a,i}}$$

Table for user based:



	Fold 1	Fold 2	Fold 3	Fold 4	Fold 5	Average
K						
10	0.784967	0.770047	0.768103	0.759655	0.766609	0.769876
20	0.766772	0.753878	0.748906	0.743244	0.749272	0.752414
30	0.764448	0.749988	0.744862	0.740816	0.745723	0.749167
40	0.763632	0.749495	0.743992	0.740220	0.746297	0.748727
50	0.762808	0.749746	0.744040	0.740529	0.746177	0.748660

Table for item based:



	Fold 1	Fold 2	Fold 3	Fold 4	Fold 5	Average
K						
10	0.759490	0.751040	0.760182	0.757165	0.762456	0.758067
20	0.744422	0.735169	0.742790	0.740425	0.743442	0.741250
30	0.741219	0.731949	0.737895	0.736400	0.738931	0.737279
40	0.739570	0.730800	0.735328	0.734636	0.737880	0.735643
50	0.738131	0.729627	0.734264	0.734168	0.737493	0.734737