Mission3

October 31, 2024

[ ]:
```
from google.colab import drive
import os, sys

#
drive.mount('/content/drive')

#
%cd /content/drive/MyDrive/

#
!mkdir -p Mission3
```

```
Drive already mounted at /content/drive; to attempt to forcibly remount, call
drive.mount("/content/drive", force_remount=True).
/content/drive/MyDrive
```

1 Mission 3.

1.1 Mission 3-1

1.1.1 (1) Item-based filtering

: item-user matrix

[ ]:
```
import pandas as pd

# Mission2-2     top100
t_top100_pref = pd.read_csv('Mission2/top100_train_preference.csv')
v_top100_pref = pd.read_csv('Mission2/top100_val_preference.csv')

#        "   ID"  "  "
dup_top100_pref = pd.merge(v_top100_pref, t_top100_pref, on=['   ID', '  ', ' ␣
↪    '], how='inner')

#
print(f"  v_top100_pref    : {len(v_top100_pref)}")
print(f"     : {len(dup_top100_pref)}")
```

```
  v_top100_pref    : 1100
```

: 646

[ ]:
```
#
dup_top100_pref.head()
```

[ ]:
```
      ID
0     368          W_04678_50_ivy_M.jpg
1     368  W_01703_00_metrosexual_M.jpg
2     368          W_15340_50_ivy_M.jpg
3     368         W_06551_60_mods_M.jpg
4     368          W_12817_50_ivy_M.jpg
```

[ ]:
```
# train        item-user Martix
def make_item_user_matrix(df):
    #         1  -1
    df['      '] = df['      '].apply(lambda x: 1 if x == '    ' else -1)

    #          user-item matrix
    user_item_matrix = df.pivot_table(index='   ID', columns='  ', values='    ␣
 ↪ ')

    # item-based filtering          item-user matrix
    item_user_matrix = user_item_matrix.T

    #       item
    mask = item_user_matrix.isna()

    #    0
    item_user_matrix.fillna(0, inplace=True)

    return item_user_matrix, mask

t_item_user_matrix, t_mask = make_item_user_matrix(t_top100_pref)
```

[ ]:
```
t_item_user_matrix      # train     item-user matrix
```

[ ]:
```
   ID                          368    837    7658   7905   9096   20768  \

T_00253_60_popart_W.jpg          0.0    0.0    0.0    0.0    0.0    0.0
T_00456_10_sportivecasual_M.jpg  0.0    0.0    0.0    0.0    0.0    0.0
T_00588_10_sportivecasual_M.jpg  0.0    0.0    0.0    0.0    0.0    0.0
T_00770_60_minimal_W.jpg         0.0    0.0    0.0    0.0    0.0    0.0
T_00893_90_hiphop_W.jpg          0.0    0.0    0.0    0.0    0.0    0.0
…                                …      …      …      …      …      …
W_71923_60_mods_M.jpg            0.0    0.0    0.0    0.0    0.0    0.0
W_71933_60_mods_M.jpg            0.0    0.0    0.0    0.0    0.0    0.0
W_71934_60_mods_M.jpg            0.0    0.0    0.0    0.0    0.0    0.0
```

```
W_71935_60_mods_M.jpg              0.0    0.0    0.0    0.0    0.0    0.0
W_71936_60_mods_M.jpg              0.0    0.0    0.0    0.0    0.0    0.0

   ID                           21432  22324  28371  28571  …  64633  \
                                                              …
T_00253_60_popart_W.jpg            0.0    0.0    0.0    0.0  …    0.0
T_00456_10_sportivecasual_M.jpg    0.0    0.0    0.0    0.0  …    0.0
T_00588_10_sportivecasual_M.jpg    0.0    0.0    0.0    0.0  …    0.0
T_00770_60_minimal_W.jpg           0.0    0.0    0.0    0.0  …    0.0
T_00893_90_hiphop_W.jpg            0.0    0.0    0.0    0.0  …    0.0
…                                  …      …      …      …    …    …
W_71923_60_mods_M.jpg              0.0    0.0    0.0    0.0  …    0.0
W_71933_60_mods_M.jpg              0.0    0.0    0.0    0.0  …    0.0
W_71934_60_mods_M.jpg              0.0    0.0    0.0    0.0  …    0.0
W_71935_60_mods_M.jpg              0.0    0.0    0.0    0.0  …    0.0
W_71936_60_mods_M.jpg              0.0    0.0    0.0    0.0  …    0.0

   ID                           64662  64747  65071  65139  66469  66513  \

T_00253_60_popart_W.jpg            0.0    0.0    0.0    0.0    0.0    0.0
T_00456_10_sportivecasual_M.jpg    0.0    0.0    0.0    0.0    1.0    0.0
T_00588_10_sportivecasual_M.jpg    0.0    0.0    0.0    0.0    1.0    0.0
T_00770_60_minimal_W.jpg           0.0    0.0    0.0    0.0    0.0    0.0
T_00893_90_hiphop_W.jpg            0.0    0.0    0.0    0.0    0.0    0.0
…                                  …      …      …      …      …      …
W_71923_60_mods_M.jpg              0.0    0.0    0.0    0.0    0.0    0.0
W_71933_60_mods_M.jpg              0.0    0.0    0.0    0.0    0.0    0.0
W_71934_60_mods_M.jpg              0.0    0.0    0.0    0.0    0.0    0.0
W_71935_60_mods_M.jpg              0.0    0.0    0.0    0.0    0.0    0.0
W_71936_60_mods_M.jpg              0.0    0.0    0.0    0.0    0.0    0.0

   ID                           66592  66731  67975

T_00253_60_popart_W.jpg            1.0    0.0    0.0
T_00456_10_sportivecasual_M.jpg    0.0    0.0    0.0
T_00588_10_sportivecasual_M.jpg    0.0    0.0    0.0
T_00770_60_minimal_W.jpg           0.0    1.0    0.0
T_00893_90_hiphop_W.jpg            1.0    0.0    0.0
…                                  …      …      …
W_71923_60_mods_M.jpg              0.0    0.0   -1.0
W_71933_60_mods_M.jpg              0.0    0.0    1.0
W_71934_60_mods_M.jpg              0.0    0.0    1.0
W_71935_60_mods_M.jpg              0.0    0.0    1.0
W_71936_60_mods_M.jpg              0.0    0.0    1.0

[4070 rows x 100 columns]
```

: Item

[ ]:
```
#
from sklearn.metrics.pairwise import cosine_similarity

t_item_similarity_matrix = cosine_similarity(t_item_user_matrix)

#      DataFrame
t_item_similarity_df = pd.DataFrame(t_item_similarity_matrix,␣
 ↪index=t_item_user_matrix.index, columns=t_item_user_matrix.index)

#
t_item_similarity_df
```

[ ]:

	T_00253_60_popart_W.jpg	\
T_00253_60_popart_W.jpg	1.0	
T_00456_10_sportivecasual_M.jpg	0.0	
T_00588_10_sportivecasual_M.jpg	0.0	
T_00770_60_minimal_W.jpg	0.0	
T_00893_90_hiphop_W.jpg	1.0	
…	…	
W_71923_60_mods_M.jpg	0.0	
W_71933_60_mods_M.jpg	0.0	
W_71934_60_mods_M.jpg	0.0	
W_71935_60_mods_M.jpg	0.0	
W_71936_60_mods_M.jpg	0.0	

	T_00456_10_sportivecasual_M.jpg	\
T_00253_60_popart_W.jpg	0.0	
T_00456_10_sportivecasual_M.jpg	1.0	
T_00588_10_sportivecasual_M.jpg	1.0	
T_00770_60_minimal_W.jpg	0.0	
T_00893_90_hiphop_W.jpg	0.0	
…	…	
W_71923_60_mods_M.jpg	0.0	
W_71933_60_mods_M.jpg	0.0	
W_71934_60_mods_M.jpg	0.0	
W_71935_60_mods_M.jpg	0.0	
W_71936_60_mods_M.jpg	0.0	

	T_00588_10_sportivecasual_M.jpg	\
T_00253_60_popart_W.jpg	0.0	
T_00456_10_sportivecasual_M.jpg	1.0	
T_00588_10_sportivecasual_M.jpg	1.0	
T_00770_60_minimal_W.jpg	0.0	

```
T_00893_90_hiphop_W.jpg                                    0.0
…                                                          …
W_71923_60_mods_M.jpg                                      0.0
W_71933_60_mods_M.jpg                                      0.0
W_71934_60_mods_M.jpg                                      0.0
W_71935_60_mods_M.jpg                                      0.0
W_71936_60_mods_M.jpg                                      0.0

                                T_00770_60_minimal_W.jpg  \

T_00253_60_popart_W.jpg                                    0.0
T_00456_10_sportivecasual_M.jpg                            0.0
T_00588_10_sportivecasual_M.jpg                            0.0
T_00770_60_minimal_W.jpg                                   1.0
T_00893_90_hiphop_W.jpg                                    0.0
…                                                          …
W_71923_60_mods_M.jpg                                      0.0
W_71933_60_mods_M.jpg                                      0.0
W_71934_60_mods_M.jpg                                      0.0
W_71935_60_mods_M.jpg                                      0.0
W_71936_60_mods_M.jpg                                      0.0

                                T_00893_90_hiphop_W.jpg  \

T_00253_60_popart_W.jpg                                    1.0
T_00456_10_sportivecasual_M.jpg                            0.0
T_00588_10_sportivecasual_M.jpg                            0.0
T_00770_60_minimal_W.jpg                                   0.0
T_00893_90_hiphop_W.jpg                                    1.0
…                                                          …
W_71923_60_mods_M.jpg                                      0.0
W_71933_60_mods_M.jpg                                      0.0
W_71934_60_mods_M.jpg                                      0.0
W_71935_60_mods_M.jpg                                      0.0
W_71936_60_mods_M.jpg                                      0.0

                                T_01322_19_normcore_M.jpg  \

T_00253_60_popart_W.jpg                                    0.0
T_00456_10_sportivecasual_M.jpg                            1.0
T_00588_10_sportivecasual_M.jpg                            1.0
T_00770_60_minimal_W.jpg                                   0.0
T_00893_90_hiphop_W.jpg                                    0.0
…                                                          …
W_71923_60_mods_M.jpg                                      0.0
W_71933_60_mods_M.jpg                                      0.0
W_71934_60_mods_M.jpg                                      0.0
```

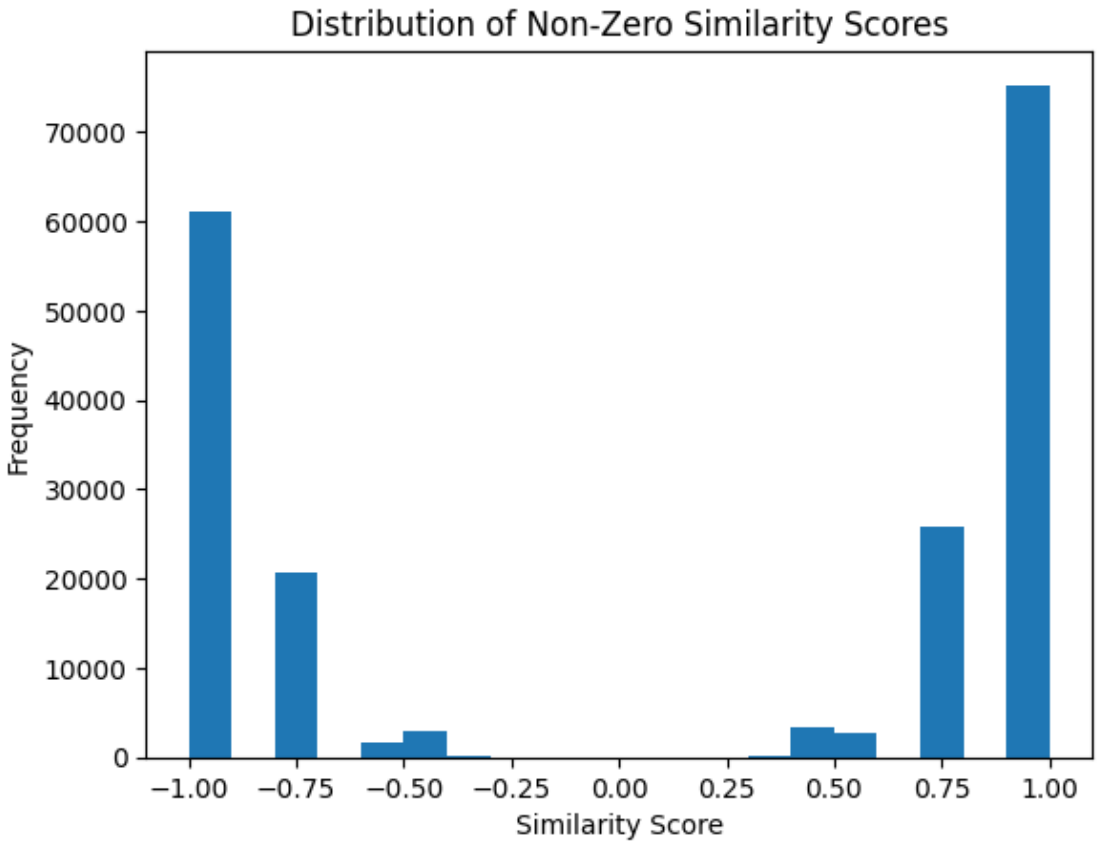
```
W_71935_60_mods_M.jpg                                0.0
W_71936_60_mods_M.jpg                                0.0

                                T_01883_10_sportivecasual_M.jpg  \

T_00253_60_popart_W.jpg                                0.0
T_00456_10_sportivecasual_M.jpg                        1.0
T_00588_10_sportivecasual_M.jpg                        1.0
T_00770_60_minimal_W.jpg                               0.0
T_00893_90_hiphop_W.jpg                                0.0
…                                                      …
W_71923_60_mods_M.jpg                                  0.0
W_71933_60_mods_M.jpg                                  0.0
W_71934_60_mods_M.jpg                                  0.0
W_71935_60_mods_M.jpg                                  0.0
W_71936_60_mods_M.jpg                                  0.0

                                T_02527_10_sportivecasual_M.jpg  \

T_00253_60_popart_W.jpg                                0.0
T_00456_10_sportivecasual_M.jpg                        1.0
T_00588_10_sportivecasual_M.jpg                        1.0
T_00770_60_minimal_W.jpg                               0.0
T_00893_90_hiphop_W.jpg                                0.0
…                                                      …
W_71923_60_mods_M.jpg                                  0.0
W_71933_60_mods_M.jpg                                  0.0
W_71934_60_mods_M.jpg                                  0.0
W_71935_60_mods_M.jpg                                  0.0
W_71936_60_mods_M.jpg                                  0.0

                                T_02558_19_normcore_M.jpg  \

T_00253_60_popart_W.jpg                              0.0
T_00456_10_sportivecasual_M.jpg                      1.0
T_00588_10_sportivecasual_M.jpg                      1.0
T_00770_60_minimal_W.jpg                             0.0
T_00893_90_hiphop_W.jpg                              0.0
…                                                    …
W_71923_60_mods_M.jpg                                0.0
W_71933_60_mods_M.jpg                                0.0
W_71934_60_mods_M.jpg                                0.0
W_71935_60_mods_M.jpg                                0.0
W_71936_60_mods_M.jpg                                0.0

                                T_02820_10_sportivecasual_W.jpg  …  \
                                                                 …
```

```
T_00253_60_popart_W.jpg                                  1.0  …
T_00456_10_sportivecasual_M.jpg                          0.0  …
T_00588_10_sportivecasual_M.jpg                          0.0  …
T_00770_60_minimal_W.jpg                                 0.0  …
T_00893_90_hiphop_W.jpg                                  1.0  …
…                                                        …  …
W_71923_60_mods_M.jpg                                    0.0  …
W_71933_60_mods_M.jpg                                    0.0  …
W_71934_60_mods_M.jpg                                    0.0  …
W_71935_60_mods_M.jpg                                    0.0  …
W_71936_60_mods_M.jpg                                    0.0  …

                                 W_68175_19_normcore_W.jpg  \

T_00253_60_popart_W.jpg                                  0.0
T_00456_10_sportivecasual_M.jpg                          0.0
T_00588_10_sportivecasual_M.jpg                          0.0
T_00770_60_minimal_W.jpg                                 0.0
T_00893_90_hiphop_W.jpg                                  0.0
…                                                        …
W_71923_60_mods_M.jpg                                    0.0
W_71933_60_mods_M.jpg                                    0.0
W_71934_60_mods_M.jpg                                    0.0
W_71935_60_mods_M.jpg                                    0.0
W_71936_60_mods_M.jpg                                    0.0

                                 W_68199_10_sportivecasual_W.jpg  \

T_00253_60_popart_W.jpg                                  0.0
T_00456_10_sportivecasual_M.jpg                          0.0
T_00588_10_sportivecasual_M.jpg                          0.0
T_00770_60_minimal_W.jpg                                 0.0
T_00893_90_hiphop_W.jpg                                  0.0
…                                                        …
W_71923_60_mods_M.jpg                                    0.0
W_71933_60_mods_M.jpg                                    0.0
W_71934_60_mods_M.jpg                                    0.0
W_71935_60_mods_M.jpg                                    0.0
W_71936_60_mods_M.jpg                                    0.0

                                 W_71920_60_mods_M.jpg  W_71921_60_mods_M.jpg  \

T_00253_60_popart_W.jpg                            0.0                    0.0
T_00456_10_sportivecasual_M.jpg                    0.0                    0.0
T_00588_10_sportivecasual_M.jpg                    0.0                    0.0
T_00770_60_minimal_W.jpg                           0.0                    0.0
T_00893_90_hiphop_W.jpg                            0.0                    0.0
```

```
…                                          …                    …
W_71923_60_mods_M.jpg                       1.0                  1.0
W_71933_60_mods_M.jpg                      -1.0                 -1.0
W_71934_60_mods_M.jpg                      -1.0                 -1.0
W_71935_60_mods_M.jpg                      -1.0                 -1.0
W_71936_60_mods_M.jpg                      -1.0                 -1.0

                                 W_71922_60_mods_M.jpg  W_71923_60_mods_M.jpg  \

T_00253_60_popart_W.jpg                     0.0                  0.0
T_00456_10_sportivecasual_M.jpg             0.0                  0.0
T_00588_10_sportivecasual_M.jpg             0.0                  0.0
T_00770_60_minimal_W.jpg                    0.0                  0.0
T_00893_90_hiphop_W.jpg                     0.0                  0.0
…                                          …                    …
W_71923_60_mods_M.jpg                       1.0                  1.0
W_71933_60_mods_M.jpg                      -1.0                 -1.0
W_71934_60_mods_M.jpg                      -1.0                 -1.0
W_71935_60_mods_M.jpg                      -1.0                 -1.0
W_71936_60_mods_M.jpg                      -1.0                 -1.0

                                 W_71933_60_mods_M.jpg  W_71934_60_mods_M.jpg  \

T_00253_60_popart_W.jpg                     0.0                  0.0
T_00456_10_sportivecasual_M.jpg             0.0                  0.0
T_00588_10_sportivecasual_M.jpg             0.0                  0.0
T_00770_60_minimal_W.jpg                    0.0                  0.0
T_00893_90_hiphop_W.jpg                     0.0                  0.0
…                                          …                    …
W_71923_60_mods_M.jpg                      -1.0                 -1.0
W_71933_60_mods_M.jpg                       1.0                  1.0
W_71934_60_mods_M.jpg                       1.0                  1.0
W_71935_60_mods_M.jpg                       1.0                  1.0
W_71936_60_mods_M.jpg                       1.0                  1.0

                                 W_71935_60_mods_M.jpg  W_71936_60_mods_M.jpg

T_00253_60_popart_W.jpg                     0.0                  0.0
T_00456_10_sportivecasual_M.jpg             0.0                  0.0
T_00588_10_sportivecasual_M.jpg             0.0                  0.0
T_00770_60_minimal_W.jpg                    0.0                  0.0
T_00893_90_hiphop_W.jpg                     0.0                  0.0
…                                          …                    …
W_71923_60_mods_M.jpg                      -1.0                 -1.0
W_71933_60_mods_M.jpg                       1.0                  1.0
W_71934_60_mods_M.jpg                       1.0                  1.0
W_71935_60_mods_M.jpg                       1.0                  1.0
```

```
W_71936_60_mods_M.jpg                    1.0                    1.0

[4070 rows x 4070 columns]
```

[ ]:
```python
#
import numpy as np
import matplotlib.pyplot as plt

#     (      1)  0(item           0     )
similarities = t_item_similarity_df.values[~np.eye(t_item_similarity_df.
  ↪shape[0], dtype=bool)]
non_zero_similarities = similarities[similarities != 0]

#
plt.hist(non_zero_similarities, bins=20)
plt.title("Distribution of Non-Zero Similarity Scores")
plt.xlabel("Similarity Score")
plt.ylabel("Frequency")
plt.show()
```



, Item 0.7

[ ]:
```
def predict_rating(dup_top100_pref, item_similarity_df, mask_df,␣
↪item_user_matrix):
    predicted_ratings = []      #

    for i in range(len(dup_top100_pref)):
        user_id = dup_top100_pref.iloc[i, 0]
        item = dup_top100_pref.iloc[i, 1]

        # user       index( )
        rated_items = mask_df[user_id].loc[mask_df[user_id] == False].index

        #
        similar_items = item_similarity_df[item].loc[rated_items]

        #
        similar_items = similar_items.sort_values(ascending=False)
        top_similar_items = similar_items[similar_items >= 0.7]     #   : 0.7

        # top_similar_items
        weighted_ratings_sum = 0    #      ( )
        similarity_sum = 0          #       ( )

        #
        for similar_item in top_similar_items.index:
            if similar_item == item:    #
                continue
            similarity = top_similar_items[similar_item]
            rating = item_user_matrix[[user_id]].loc[similar_item].values
            weighted_ratings_sum += similarity * rating
            similarity_sum += similarity

        if similarity_sum != 0:
            predicted_rating = weighted_ratings_sum / similarity_sum
        else:
            predicted_rating = 0

        #
        predicted_ratings.append([user_id, item, predicted_rating])

    #      DataFrame
    predicted_df = pd.DataFrame(predicted_ratings, columns=['   ID', '  ', ' ␣
↪ '])

    return predicted_df
```

```
#
predicted_ratings = predict_rating(dup_top100_pref, t_item_similarity_df,␣
 ↪t_mask, t_item_user_matrix)
```

[ ]:
```
#
total_dup_pref = dup_top100_pref.merge(predicted_ratings, on=['   ID', '  '],␣
 ↪how='left')
total_dup_pref['      '] = total_dup_pref['      '].apply(lambda x: 1 if x ==␣
 ↪'    ' else -1)

#          1 ,         -1   (0        -1    )
total_dup_pref['     '] = total_dup_pref['     '].apply(lambda x: 1 if x > 0␣
 ↪else -1)

#       (Accuracy / Precision / Recall)
from sklearn.metrics import accuracy_score, precision_score, recall_score

accuracy = accuracy_score(total_dup_pref['      '], total_dup_pref['     '])
precision = precision_score(total_dup_pref['      '], total_dup_pref['     '])
recall = recall_score(total_dup_pref['      '], total_dup_pref['     '])

print(f"Accuracy: {accuracy:.4f}, Precision: {precision:.4f}, Recall: {recall:.
 ↪4f}")

#
total_dup_pref
```

```
Accuracy: 0.9737, Precision: 1.0000, Recall: 0.9306
```

[ ]:

	ID			
0	368	W_04678_50_ivy_M.jpg	1	-1
1	368	W_01703_00_metrosexual_M.jpg	1	1
2	368	W_15340_50_ivy_M.jpg	-1	-1
3	368	W_06551_60_mods_M.jpg	-1	-1
4	368	W_12817_50_ivy_M.jpg	1	-1
..	…	…	…	…
641	67975	W_07074_00_metrosexual_M.jpg	1	1
642	67975	W_17747_80_bold_M.jpg	-1	-1
643	67975	W_17738_80_bold_M.jpg	1	1
644	67975	T_21986_70_hippie_M.jpg	-1	-1
645	67975	W_26965_90_hiphop_M.jpg	-1	-1

[646 rows x 4 columns]

1.1.2 (2) User-based filtering

: user-item matrix

[ ]:
```
import pandas as pd

# top100
t_top100_pref = pd.read_csv('Mission2/top100_train_preference.csv')
v_top100_pref = pd.read_csv('Mission2/top100_val_preference.csv')

#        "   ID"  "  "
dup_top100_pref = pd.merge(v_top100_pref, t_top100_pref, on=['   ID', '  ', '  ␣
↪    '], how='inner')

#
print(f"  v_top100_pref    : {len(v_top100_pref)}")
print(f"     : {len(dup_top100_pref)}")
```

```
 v_top100_pref    : 1100
    : 646
```

[ ]:
```
#
dup_top100_pref.head()
```

[ ]:

	ID	
0	368	W_04678_50_ivy_M.jpg
1	368	W_01703_00_metrosexual_M.jpg
2	368	W_15340_50_ivy_M.jpg
3	368	W_06551_60_mods_M.jpg
4	368	W_12817_50_ivy_M.jpg

[ ]:
```
def make_user_item_matrix(df):
    #         1  0
    df['      '] = df['      '].apply(lambda x: 1 if x == '    ' else -1)

    #          user-item matrix
    user_item_matrix = df.pivot_table(index='   ID', columns='  ', values='    ␣
↪ ')

    #       user
    mask = user_item_matrix.isna()

    #    0
    user_item_matrix.fillna(0, inplace=True)

    return user_item_matrix, mask

t_user_item_matrix, t_mask = make_user_item_matrix(t_top100_pref)
```

[ ]:
```
t_user_item_matrix      # train    user-item matrix
```

[ ]:

ID	T_00253_60_popart_W.jpg	T_00456_10_sportivecasual_M.jpg	\
368	0.0	0.0	
837	0.0	0.0	
7658	0.0	0.0	
7905	0.0	0.0	
9096	0.0	0.0	
…	…	…	
66469	0.0	1.0	
66513	0.0	0.0	
66592	1.0	0.0	
66731	0.0	0.0	
67975	0.0	0.0	

ID	T_00588_10_sportivecasual_M.jpg	T_00770_60_minimal_W.jpg	\
368	0.0	0.0	
837	0.0	0.0	
7658	0.0	0.0	
7905	0.0	0.0	
9096	0.0	0.0	
…	…	…	
66469	1.0	0.0	
66513	0.0	0.0	
66592	0.0	0.0	
66731	0.0	1.0	
67975	0.0	0.0	

ID	T_00893_90_hiphop_W.jpg	T_01322_19_normcore_M.jpg	\
368	0.0	0.0	
837	0.0	0.0	
7658	0.0	0.0	
7905	0.0	0.0	
9096	0.0	0.0	
…	…	…	
66469	0.0	1.0	
66513	0.0	0.0	
66592	1.0	0.0	
66731	0.0	0.0	
67975	0.0	0.0	

ID	T_01883_10_sportivecasual_M.jpg	T_02527_10_sportivecasual_M.jpg	\
368	0.0	0.0	
837	0.0	0.0	
7658	0.0	0.0	

```
7905                                0.0                                0.0
9096                                0.0                                0.0
…                                   …                                  …
66469                               1.0                                1.0
66513                               0.0                                0.0
66592                               0.0                                0.0
66731                               0.0                                0.0
67975                               0.0                                0.0

       T_02558_19_normcore_M.jpg  T_02820_10_sportivecasual_W.jpg  …  \
  ID                                                                 …
368                          0.0                                0.0  …
837                          0.0                                0.0  …
7658                         0.0                                0.0  …
7905                         0.0                                0.0  …
9096                         0.0                                0.0  …
…                            …                                  …  …
66469                        1.0                                0.0  …
66513                        0.0                                0.0  …
66592                        0.0                                1.0  …
66731                        0.0                                0.0  …
67975                        0.0                                0.0  …

       W_68175_19_normcore_W.jpg  W_68199_10_sportivecasual_W.jpg  \
  ID
368                          0.0                                0.0
837                          0.0                                0.0
7658                         0.0                                0.0
7905                         0.0                                0.0
9096                         0.0                                0.0
…                            …                                  …
66469                        0.0                                0.0
66513                        0.0                               -1.0
66592                        0.0                                0.0
66731                        0.0                                0.0
67975                        0.0                                0.0

       W_71920_60_mods_M.jpg  W_71921_60_mods_M.jpg  W_71922_60_mods_M.jpg  \
  ID
368                      0.0                    0.0                    0.0
837                      0.0                    0.0                    0.0
7658                     0.0                    0.0                    0.0
7905                     0.0                    0.0                    0.0
9096                     0.0                    0.0                    0.0
…                        …                      …                      …
66469                    0.0                    0.0                    0.0
66513                    0.0                    0.0                    0.0
```

```
66592                     0.0                     0.0                     0.0
66731                     0.0                     0.0                     0.0
67975                    -1.0                    -1.0                    -1.0

       W_71923_60_mods_M.jpg  W_71933_60_mods_M.jpg  W_71934_60_mods_M.jpg  \
   ID
368                       0.0                     0.0                     0.0
837                       0.0                     0.0                     0.0
7658                      0.0                     0.0                     0.0
7905                      0.0                     0.0                     0.0
9096                      0.0                     0.0                     0.0
…                         …                       …                       …
66469                     0.0                     0.0                     0.0
66513                     0.0                     0.0                     0.0
66592                     0.0                     0.0                     0.0
66731                     0.0                     0.0                     0.0
67975                    -1.0                     1.0                     1.0

       W_71935_60_mods_M.jpg  W_71936_60_mods_M.jpg
   ID
368                       0.0                     0.0
837                       0.0                     0.0
7658                      0.0                     0.0
7905                      0.0                     0.0
9096                      0.0                     0.0
…                         …                       …
66469                     0.0                     0.0
66513                     0.0                     0.0
66592                     0.0                     0.0
66731                     0.0                     0.0
67975                     1.0                     1.0

[100 rows x 4070 columns]
```

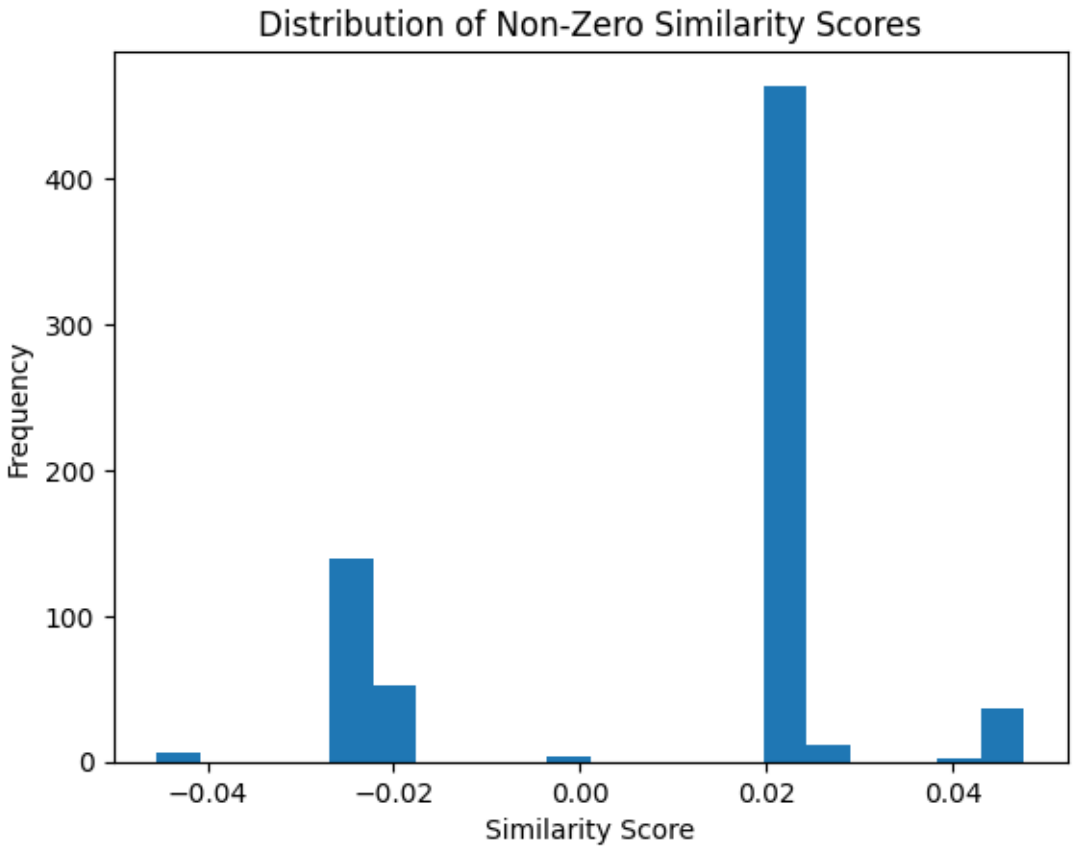
: User

[ ]:
```
#
from sklearn.metrics.pairwise import cosine_similarity

t_user_similarity_matrix = cosine_similarity(t_user_item_matrix)

#      DataFrame
t_user_similarity_df = pd.DataFrame(t_user_similarity_matrix,␣
 ↪index=t_user_item_matrix.index, columns=t_user_item_matrix.index)

# User
t_user_similarity_df
```

[ ]:
```
ID  368    837    7658   7905   9096   20768     21432     22324     28371  \
ID
368       1.0    0.0    0.0    0.0    0.0    0.0  0.000000  0.00000  0.000000
837       0.0    1.0    0.0    0.0    0.0    0.0  0.000000  0.00000  0.022751
7658      0.0    0.0    1.0    0.0    0.0    0.0  0.000000  0.00000  0.000000
7905      0.0    0.0    0.0    1.0    0.0    0.0  0.000000  0.02299 -0.023531
9096      0.0    0.0    0.0    0.0    1.0    0.0  0.000000  0.00000  0.000000
…       …    …    …    …    …    …       …       …       …
66469     0.0    0.0    0.0    0.0    0.0    0.0  0.022751  0.00000  0.000000
66513     0.0    0.0    0.0    0.0    0.0    0.0  0.000000  0.00000  0.000000
66592     0.0    0.0    0.0    0.0    0.0    0.0  0.000000  0.00000  0.000000
66731     0.0    0.0    0.0    0.0    0.0    0.0  0.000000  0.00000  0.000000
67975     0.0    0.0    0.0    0.0    0.0    0.0  0.022228  0.00000  0.000000

ID     28571  …     64633  64662     64747  65071  65139  66469  66513  \
ID            …
368    0.022473  …  0.022990    0.0  0.000000    0.0    0.0    0.0    0.0
837    0.000000  …  0.000000    0.0  0.000000    0.0    0.0    0.0    0.0
7658   0.000000  …  0.000000    0.0  0.000000    0.0    0.0    0.0    0.0
7905   0.000000  …  0.023256    0.0  0.000000    0.0    0.0    0.0    0.0
9096   0.000000  …  0.000000    0.0  0.000000    0.0    0.0    0.0    0.0
…         …  …       …    …       …    …    …    …    …
66469  0.000000  …  0.000000    0.0  0.000000    0.0    0.0    1.0    0.0
66513  0.000000  …  0.000000    0.0  0.000000    0.0    0.0    0.0    1.0
66592  0.000000  …  0.000000    0.0  0.021979    0.0    0.0    0.0    0.0
66731  0.000000  …  0.000000    0.0  0.000000    0.0    0.0    0.0    0.0
67975  0.000000  …  0.000000    0.0  0.000000    0.0    0.0    0.0    0.0

ID  66592  66731  67975
ID
368       0.0    0.0    0.0
837       0.0    0.0    0.0
7658      0.0    0.0    0.0
7905      0.0    0.0    0.0
9096      0.0    0.0    0.0
…       …    …    …
66469     0.0    0.0    0.0
66513     0.0    0.0    0.0
66592     1.0    0.0    0.0
66731     0.0    1.0    0.0
67975     0.0    0.0    1.0

[100 rows x 100 columns]
```

[ ]:
```
#
import numpy as np
import matplotlib.pyplot as plt
```

```
#     (      1)  0(item           0     )
similarities = t_user_similarity_df.values[~np.eye(t_user_similarity_df.
 ↪shape[0], dtype=bool)]
non_zero_similarities = similarities[similarities != 0]

#
plt.hist(non_zero_similarities, bins=20)
plt.title("Distribution of Non-Zero Similarity Scores")
plt.xlabel("Similarity Score")
plt.ylabel("Frequency")
plt.show()
```



[ ]:
```
def predict_rating(dup_top100_pref, user_similarity_df, mask_df,␣
 ↪user_item_matrix):
    predicted_ratings = []      #

    for i in range(len(dup_top100_pref)):
```

```
        user_id = dup_top100_pref.iloc[i, 0]
        item = dup_top100_pref.iloc[i, 1]

        #   item    user  index(userID)
        rated_user = mask_df[item].loc[mask_df[item] == False].index

        #    user
        similar_users = user_similarity_df[user_id].loc[rated_user]

        #
        similar_users = similar_users.sort_values(ascending=False)
        top_similar_users = similar_users[similar_users >= 0.7]

        # top_similar_users
        weighted_ratings_sum = 0    #       ( )
        similarity_sum = 0          #       ( )

        # item    user
        for similar_user in top_similar_users.index:
            if similar_user == user_id:    #
                continue
            similarity = top_similar_users[similar_user]
            rating = user_item_matrix[[item]].loc[similar_user].values
            weighted_ratings_sum += similarity * rating
            similarity_sum += similarity

        if similarity_sum != 0:
            predicted_rating = weighted_ratings_sum / similarity_sum
        else:
            predicted_rating = 0

        #
        predicted_ratings.append([user_id, item, predicted_rating])

    #      DataFrame
    predicted_df = pd.DataFrame(predicted_ratings, columns=['   ID', '  ', '  ␣
 ↪ '])

    return predicted_df

#
predicted_ratings = predict_rating(dup_top100_pref, t_user_similarity_df,␣
 ↪t_mask, t_user_item_matrix)
```

[ ]:
```
#
total_dup_pref = dup_top100_pref.merge(predicted_ratings, on=['   ID', '  '],␣
 ↪how='left')
```

```
total_dup_pref['      '] = total_dup_pref['      '].apply(lambda x: 1 if x ==␣
 ↪'    ' else -1)

#          1 ,        -1
total_dup_pref['    '] = total_dup_pref['    '].apply(lambda x: 1 if x > 0␣
 ↪else -1)

#      (Accuracy / Precision / Recall)
from sklearn.metrics import accuracy_score, precision_score, recall_score

accuracy = accuracy_score(total_dup_pref['      '], total_dup_pref['    '])
precision = precision_score(total_dup_pref['      '], total_dup_pref['    '])
recall = recall_score(total_dup_pref['      '], total_dup_pref['    '])

print(f"Accuracy: {accuracy:.4f}, Precision: {precision:.4f}, Recall: {recall:.
 ↪4f}")

#
total_dup_pref
```

```
Accuracy: 0.6207, Precision: 0.0000, Recall: 0.0000
```

```
/usr/local/lib/python3.10/dist-packages/sklearn/metrics/_classification.py:1531:
UndefinedMetricWarning: Precision is ill-defined and being set to 0.0 due to no
predicted samples. Use `zero_division` parameter to control this behavior.
  _warn_prf(average, modifier, f"{metric.capitalize()} is", len(result))
```

[ ]:

	ID			
0	368	W_04678_50_ivy_M.jpg	1	-1
1	368	W_01703_00_metrosexual_M.jpg	1	-1
2	368	W_15340_50_ivy_M.jpg	-1	-1
3	368	W_06551_60_mods_M.jpg	-1	-1
4	368	W_12817_50_ivy_M.jpg	1	-1
..	…	…	…	…
641	67975	W_07074_00_metrosexual_M.jpg	1	-1
642	67975	W_17747_80_bold_M.jpg	-1	-1
643	67975	W_17738_80_bold_M.jpg	1	-1
644	67975	T_21986_70_hippie_M.jpg	-1	-1
645	67975	W_26965_90_hiphop_M.jpg	-1	-1

[646 rows x 4 columns]

user user .

[ ]:
```
# user
t_user_similarity_df
```

[ ]:
```
ID  368    837    7658   7905   9096   20768     21432     22324     28371  \
ID
368       1.0    0.0    0.0    0.0    0.0    0.0  0.000000  0.00000  0.000000
837       0.0    1.0    0.0    0.0    0.0    0.0  0.000000  0.00000  0.022751
7658      0.0    0.0    1.0    0.0    0.0    0.0  0.000000  0.00000  0.000000
7905      0.0    0.0    0.0    1.0    0.0    0.0  0.000000  0.02299 -0.023531
9096      0.0    0.0    0.0    0.0    1.0    0.0  0.000000  0.00000  0.000000
…       …    …    …    …    …    …       …       …       …
66469     0.0    0.0    0.0    0.0    0.0    0.0  0.022751  0.00000  0.000000
66513     0.0    0.0    0.0    0.0    0.0    0.0  0.000000  0.00000  0.000000
66592     0.0    0.0    0.0    0.0    0.0    0.0  0.000000  0.00000  0.000000
66731     0.0    0.0    0.0    0.0    0.0    0.0  0.000000  0.00000  0.000000
67975     0.0    0.0    0.0    0.0    0.0    0.0  0.022228  0.00000  0.000000

ID     28571  …     64633  64662     64747  65071  65139  66469  66513  \
ID            …
368    0.022473  …  0.022990    0.0  0.000000    0.0    0.0    0.0    0.0
837    0.000000  …  0.000000    0.0  0.000000    0.0    0.0    0.0    0.0
7658   0.000000  …  0.000000    0.0  0.000000    0.0    0.0    0.0    0.0
7905   0.000000  …  0.023256    0.0  0.000000    0.0    0.0    0.0    0.0
9096   0.000000  …  0.000000    0.0  0.000000    0.0    0.0    0.0    0.0
…         …  …       …    …       …    …    …    …    …
66469  0.000000  …  0.000000    0.0  0.000000    0.0    0.0    1.0    0.0
66513  0.000000  …  0.000000    0.0  0.000000    0.0    0.0    0.0    1.0
66592  0.000000  …  0.000000    0.0  0.021979    0.0    0.0    0.0    0.0
66731  0.000000  …  0.000000    0.0  0.000000    0.0    0.0    0.0    0.0
67975  0.000000  …  0.000000    0.0  0.000000    0.0    0.0    0.0    0.0

ID  66592  66731  67975
ID
368       0.0    0.0    0.0
837       0.0    0.0    0.0
7658      0.0    0.0    0.0
7905      0.0    0.0    0.0
9096      0.0    0.0    0.0
…       …    …    …
66469     0.0    0.0    0.0
66513     0.0    0.0    0.0
66592     1.0    0.0    0.0
66731     0.0    1.0    0.0
67975     0.0    0.0    1.0

[100 rows x 100 columns]
```

[ ]:
```
#     (0.7  ) user
import numpy as np
user_similar_df = t_user_similarity_df.copy()
```

```
#      0
np.fill_diagonal(user_similar_df.values, 0)

#    0.7            0
df_filtered = user_similar_df.where(user_similar_df >= 0.7, 0)

#     0
result = df_filtered.apply(lambda x: x[x != 0])

#
result
```

[ ]:
```
Empty DataFrame
Columns: [368, 837, 7658, 7905, 9096, 20768, 21432, 22324, 28371, 28571, 28912,
30790, 35514, 58251, 59083, 59506, 59523, 59637, 59642, 59704, 59812, 60173,
60184, 60234, 60465, 61104, 61250, 61493, 61859, 62113, 62155, 62264, 62349,
62361, 62525, 62625, 62653, 62868, 62952, 63057, 63156, 63207, 63316, 63359,
63369, 63392, 63405, 63424, 63430, 63435, 63473, 63479, 63481, 63505, 63508,
63526, 63545, 63569, 63571, 63583, 63601, 63644, 63740, 63742, 63748, 63759,
63769, 63910, 63913, 63927, 63930, 63934, 64216, 64221, 64223, 64252, 64280,
64295, 64310, 64336, 64345, 64346, 64364, 64397, 64441, 64460, 64503, 64561,
64571, 64598, 64633, 64662, 64747, 65071, 65139, 66469, 66513, 66592, 66731,
67975]
Index: []

[0 rows x 100 columns]
```

User 0.047 .

[ ]:
```
#
user_similar_df.values.max()
```

[ ]:
```
0.04767312946227962
```

, user 1 . item user (100), . , user .

1.2 Mission 3-2

1.2.1 train data feature vector

[ ]:
```
#
from google.colab import drive
import os, sys

#
drive.mount('/content/drive')
```

```
#
%cd /content/drive/MyDrive/

#
!mkdir -p Mission3
```

```
Drive already mounted at /content/drive; to attempt to forcibly remount, call
drive.mount("/content/drive", force_remount=True).
/content/drive/MyDrive
```

[ ]:
```
#      import
import os
import random
import numpy as np
import pandas as pd
from PIL import Image
import torch
import torch.nn as nn
import torchvision.transforms as transforms


#
def set_random_seed(seed_value=42):
    # Python
    random.seed(seed_value)
    # NumPy
    np.random.seed(seed_value)
    # PyTorch       (CPU)
    torch.manual_seed(seed_value)
    # PyTorch       (GPU)
    torch.cuda.manual_seed(seed_value)
    torch.cuda.manual_seed_all(seed_value)
    # CuDNN
    torch.backends.cudnn.deterministic = True
    torch.backends.cudnn.benchmark = False

set_random_seed()
```

[ ]:
```
import torchvision.models as models

# GPU
device = torch.device("cuda" if torch.cuda.is_available() else "cpu")
print(device)

#
model = models.resnet18(weights=None, num_classes=31)
```

```
# Mission1  Loss
model.load_state_dict(torch.load('Mission1/best_loss_exp/best_model.pth',␣
 ↪weights_only=True))
model = model.to(device)
model.eval()    #           eval
```

cuda

[ ]:
```
ResNet(
  (conv1): Conv2d(3, 64, kernel_size=(7, 7), stride=(2, 2), padding=(3, 3),
bias=False)
  (bn1): BatchNorm2d(64, eps=1e-05, momentum=0.1, affine=True,
track_running_stats=True)
  (relu): ReLU(inplace=True)
  (maxpool): MaxPool2d(kernel_size=3, stride=2, padding=1, dilation=1,
ceil_mode=False)
  (layer1): Sequential(
    (0): BasicBlock(
      (conv1): Conv2d(64, 64, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1),
bias=False)
      (bn1): BatchNorm2d(64, eps=1e-05, momentum=0.1, affine=True,
track_running_stats=True)
      (relu): ReLU(inplace=True)
      (conv2): Conv2d(64, 64, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1),
bias=False)
      (bn2): BatchNorm2d(64, eps=1e-05, momentum=0.1, affine=True,
track_running_stats=True)
    )
    (1): BasicBlock(
      (conv1): Conv2d(64, 64, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1),
bias=False)
      (bn1): BatchNorm2d(64, eps=1e-05, momentum=0.1, affine=True,
track_running_stats=True)
      (relu): ReLU(inplace=True)
      (conv2): Conv2d(64, 64, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1),
bias=False)
      (bn2): BatchNorm2d(64, eps=1e-05, momentum=0.1, affine=True,
track_running_stats=True)
    )
  )
  (layer2): Sequential(
    (0): BasicBlock(
      (conv1): Conv2d(64, 128, kernel_size=(3, 3), stride=(2, 2), padding=(1,
1), bias=False)
      (bn1): BatchNorm2d(128, eps=1e-05, momentum=0.1, affine=True,
track_running_stats=True)
```

```
      (relu): ReLU(inplace=True)
      (conv2): Conv2d(128, 128, kernel_size=(3, 3), stride=(1, 1), padding=(1,
1), bias=False)
      (bn2): BatchNorm2d(128, eps=1e-05, momentum=0.1, affine=True,
track_running_stats=True)
      (downsample): Sequential(
        (0): Conv2d(64, 128, kernel_size=(1, 1), stride=(2, 2), bias=False)
        (1): BatchNorm2d(128, eps=1e-05, momentum=0.1, affine=True,
track_running_stats=True)
      )
    )
    (1): BasicBlock(
      (conv1): Conv2d(128, 128, kernel_size=(3, 3), stride=(1, 1), padding=(1,
1), bias=False)
      (bn1): BatchNorm2d(128, eps=1e-05, momentum=0.1, affine=True,
track_running_stats=True)
      (relu): ReLU(inplace=True)
      (conv2): Conv2d(128, 128, kernel_size=(3, 3), stride=(1, 1), padding=(1,
1), bias=False)
      (bn2): BatchNorm2d(128, eps=1e-05, momentum=0.1, affine=True,
track_running_stats=True)
    )
  )
  (layer3): Sequential(
    (0): BasicBlock(
      (conv1): Conv2d(128, 256, kernel_size=(3, 3), stride=(2, 2), padding=(1,
1), bias=False)
      (bn1): BatchNorm2d(256, eps=1e-05, momentum=0.1, affine=True,
track_running_stats=True)
      (relu): ReLU(inplace=True)
      (conv2): Conv2d(256, 256, kernel_size=(3, 3), stride=(1, 1), padding=(1,
1), bias=False)
      (bn2): BatchNorm2d(256, eps=1e-05, momentum=0.1, affine=True,
track_running_stats=True)
      (downsample): Sequential(
        (0): Conv2d(128, 256, kernel_size=(1, 1), stride=(2, 2), bias=False)
        (1): BatchNorm2d(256, eps=1e-05, momentum=0.1, affine=True,
track_running_stats=True)
      )
    )
    (1): BasicBlock(
      (conv1): Conv2d(256, 256, kernel_size=(3, 3), stride=(1, 1), padding=(1,
1), bias=False)
      (bn1): BatchNorm2d(256, eps=1e-05, momentum=0.1, affine=True,
track_running_stats=True)
      (relu): ReLU(inplace=True)
      (conv2): Conv2d(256, 256, kernel_size=(3, 3), stride=(1, 1), padding=(1,
```

```
1), bias=False)
      (bn2): BatchNorm2d(256, eps=1e-05, momentum=0.1, affine=True,
track_running_stats=True)
    )
  )
  (layer4): Sequential(
    (0): BasicBlock(
      (conv1): Conv2d(256, 512, kernel_size=(3, 3), stride=(2, 2), padding=(1,
1), bias=False)
      (bn1): BatchNorm2d(512, eps=1e-05, momentum=0.1, affine=True,
track_running_stats=True)
      (relu): ReLU(inplace=True)
      (conv2): Conv2d(512, 512, kernel_size=(3, 3), stride=(1, 1), padding=(1,
1), bias=False)
      (bn2): BatchNorm2d(512, eps=1e-05, momentum=0.1, affine=True,
track_running_stats=True)
      (downsample): Sequential(
        (0): Conv2d(256, 512, kernel_size=(1, 1), stride=(2, 2), bias=False)
        (1): BatchNorm2d(512, eps=1e-05, momentum=0.1, affine=True,
track_running_stats=True)
      )
    )
    (1): BasicBlock(
      (conv1): Conv2d(512, 512, kernel_size=(3, 3), stride=(1, 1), padding=(1,
1), bias=False)
      (bn1): BatchNorm2d(512, eps=1e-05, momentum=0.1, affine=True,
track_running_stats=True)
      (relu): ReLU(inplace=True)
      (conv2): Conv2d(512, 512, kernel_size=(3, 3), stride=(1, 1), padding=(1,
1), bias=False)
      (bn2): BatchNorm2d(512, eps=1e-05, momentum=0.1, affine=True,
track_running_stats=True)
    )
  )
  (avgpool): AdaptiveAvgPool2d(output_size=(1, 1))
  (fc): Linear(in_features=512, out_features=31, bias=True)
)
```

[ ]:
```python
#       Class
class FeatureExtractor(nn.Module):
    def __init__(self, original_model):
        super(FeatureExtractor, self).__init__()
        self.features = nn.Sequential(*list(original_model.children())[:-1])
 ↪#    fc layer

    def forward(self, x):
        x = self.features(x)
```

```
        return torch.flatten(x, 1)

#
feature_extractor = FeatureExtractor(model)
feature_extractor = feature_extractor.to(device)
```

[ ]:
```
import torch.nn as nn
import torchvision.models as models
import torchvision.transforms as transforms
from PIL import Image
import numpy as np
import pandas as pd
import os
from tqdm import tqdm
from concurrent.futures import ThreadPoolExecutor, as_completed

#
def _preprocess_image(image_path):
    transform = transforms.Compose([
        transforms.Resize((224, 224)),
        transforms.ToTensor(),
        transforms.Normalize(mean=[0.485, 0.456, 0.406], std=[0.229, 0.224, 0.
 ↪225])
    ])
    image = Image.open(image_path).convert('RGB')
    return transform(image).unsqueeze(0)  #

# feature vector
def _extract_features(image_path, device, feature_extractor):
    image = _preprocess_image(image_path).to(device)
    with torch.no_grad():
        features = feature_extractor(image)
    return features.cpu().numpy().flatten()

#       feature vector   (Colab I/O               , os.cpu_count() == 12 -> L4␣
 ↪GPU        )
def extract_features_from_images(dir_path, image_list, device,␣
 ↪feature_extractor):
    features = {}
    with ThreadPoolExecutor(max_workers=8) as executor:
        futures = {executor.submit(_extract_features, os.path.join(dir_path,␣
 ↪img), device, feature_extractor): img for img in image_list}

        for future in tqdm(as_completed(futures), total=len(futures),␣
 ↪desc="Extracting features"):
            img = futures[future]
            try:
```

```
                features[img] = np.array(future.result())
            except Exception as e:
                print(f"{img}        : {e}")
    return features

def extract_and_save(state='train', save=True):
    img_path = './dataset/origin_dataset'
    if state == 'train':
        dir_path = os.path.join(img_path, 'training_image')
    else:
        dir_path = os.path.join(img_path, 'validation_image')
    img_list = os.listdir(dir_path)

    extracted_features = extract_features_from_images(dir_path, img_list,␣
 ↪device, feature_extractor)

    # feature  shape
    print(f"    : {len(extracted_features)}, feature shape:␣
 ↪{extracted_features[img_list[0]].shape}")

    #    feature vector  DataFrame
    feature_vectors = pd.DataFrame(extracted_features).T
    if save:
        os.makedirs('Mission3', exist_ok=True)
        feature_vectors.to_csv(f'Mission3/{state}_feature_vectors.csv')

    return feature_vectors

# feature vector
t_feature_vectors = extract_and_save('train', save=True)
v_feature_vectors = extract_and_save('val', save=True)
```

```
Extracting features: 100%|      | 4070/4070 [05:10<00:00, 13.09it/s]
```

```
   : 4070, feature shape: (512,)
```

```
Extracting features: 100%|      | 951/951 [01:16<00:00, 12.36it/s]
```

```
   : 951, feature shape: (512,)
```

1.2.2 feature vector

[ ]:
```
import numpy as np
import pandas as pd

# Mission2-2     top100
t_top100_pref = pd.read_csv('Mission2/top100_train_preference.csv')
v_top100_pref = pd.read_csv('Mission2/top100_val_preference.csv')
```

```
#      1  -1   (Mission 3-1      )
t_top100_pref['      '] = t_top100_pref['      '].apply(lambda x: 1 if x == '  ␣
 ↪ ' else -1)
v_top100_pref['      '] = v_top100_pref['      '].apply(lambda x: 1 if x == '  ␣
 ↪ ' else -1)
```

[ ]:
```
def feature_preprocessing(csv_path):
    feature_vectors = pd.read_csv(csv_path, index_col=0)        # feature␣
 ↪vector
    feature_vectors['feature_vector'] = feature_vectors.values.tolist()      #␣
 ↪512   feature vector    list

    # '  ' 'feature_vector'
    feature_reset = feature_vectors.reset_index().rename(columns={'index':␣
 ↪'  '})
    feature_simplified = feature_reset[['  ', 'feature_vector']]

    return feature_simplified

#    feature vector csv
t_feature_simplified = feature_preprocessing('Mission3/train_feature_vectors.
 ↪csv')
v_feature_simplified = feature_preprocessing('Mission3/val_feature_vectors.csv')
```

[ ]:
```
# train  validation    feature vector
from sklearn.metrics.pairwise import cosine_similarity

# feature_vectors  numpy
train_features = np.stack(t_feature_simplified['feature_vector'].values)
val_features = np.stack(v_feature_simplified['feature_vector'].values)

#
similarity_matrix = cosine_similarity(train_features, val_features)

#    DataFrame
similarity_df = pd.DataFrame(similarity_matrix,
                             index=t_feature_simplified['  '],
                             columns=v_feature_simplified['  '])

# train-validation
similarity_df
```

[ ]:
```
                              W_20598_70_military_W.jpg  \

W_13846_60_minimal_W.jpg                       0.578833
W_10981_50_feminine_W.jpg                      0.758445
W_14489_19_normcore_W.jpg                      0.667769
```

```
W_16836_19_normcore_M.jpg                        0.450144
W_17123_19_normcore_M.jpg                        0.470672
…                                                   …
W_25400_19_normcore_M.jpg                        0.412683
W_17108_19_normcore_M.jpg                        0.430460
W_09233_60_mods_M.jpg                            0.337876
W_12814_19_normcore_M.jpg                        0.468611
W_06682_00_metrosexual_M.jpg                     0.460755

                              W_02498_50_feminine_W.jpg  \

W_13846_60_minimal_W.jpg                         0.652536
W_10981_50_feminine_W.jpg                        0.900691
W_14489_19_normcore_W.jpg                        0.616123
W_16836_19_normcore_M.jpg                        0.412058
W_17123_19_normcore_M.jpg                        0.470980
…                                                   …
W_25400_19_normcore_M.jpg                        0.397502
W_17108_19_normcore_M.jpg                        0.448311
W_09233_60_mods_M.jpg                            0.295941
W_12814_19_normcore_M.jpg                        0.436388
W_06682_00_metrosexual_M.jpg                     0.503961

                              W_11610_90_grunge_W.jpg  \

W_13846_60_minimal_W.jpg                       0.407351
W_10981_50_feminine_W.jpg                      0.533222
W_14489_19_normcore_W.jpg                      0.684993
W_16836_19_normcore_M.jpg                      0.511955
W_17123_19_normcore_M.jpg                      0.549426
…                                                 …
W_25400_19_normcore_M.jpg                      0.447764
W_17108_19_normcore_M.jpg                      0.575467
W_09233_60_mods_M.jpg                          0.374216
W_12814_19_normcore_M.jpg                      0.532495
W_06682_00_metrosexual_M.jpg                   0.656902

                              W_47169_70_hippie_W.jpg  \

W_13846_60_minimal_W.jpg                       0.390600
W_10981_50_feminine_W.jpg                      0.578887
W_14489_19_normcore_W.jpg                      0.591835
W_16836_19_normcore_M.jpg                      0.390615
W_17123_19_normcore_M.jpg                      0.514430
…                                                 …
W_25400_19_normcore_M.jpg                      0.373020
W_17108_19_normcore_M.jpg                      0.547781
```

```
W_09233_60_mods_M.jpg                          0.285780
W_12814_19_normcore_M.jpg                      0.418036
W_06682_00_metrosexual_M.jpg                   0.601679

                               W_05628_00_cityglam_W.jpg  \

W_13846_60_minimal_W.jpg                         0.593423
W_10981_50_feminine_W.jpg                        0.633572
W_14489_19_normcore_W.jpg                        0.768282
W_16836_19_normcore_M.jpg                        0.493622
W_17123_19_normcore_M.jpg                        0.490963
…                                                   …
W_25400_19_normcore_M.jpg                        0.472459
W_17108_19_normcore_M.jpg                        0.576505
W_09233_60_mods_M.jpg                            0.340084
W_12814_19_normcore_M.jpg                        0.511976
W_06682_00_metrosexual_M.jpg                     0.611730

                               W_38588_19_genderless_W.jpg  \

W_13846_60_minimal_W.jpg                         0.616600
W_10981_50_feminine_W.jpg                        0.524184
W_14489_19_normcore_W.jpg                        0.678110
W_16836_19_normcore_M.jpg                        0.874314
W_17123_19_normcore_M.jpg                        0.875660
…                                                   …
W_25400_19_normcore_M.jpg                        0.853628
W_17108_19_normcore_M.jpg                        0.871043
W_09233_60_mods_M.jpg                            0.630628
W_12814_19_normcore_M.jpg                        0.885744
W_06682_00_metrosexual_M.jpg                     0.659312

                               W_22510_80_powersuit_W.jpg  \

W_13846_60_minimal_W.jpg                         0.698037
W_10981_50_feminine_W.jpg                        0.584434
W_14489_19_normcore_W.jpg                        0.505030
W_16836_19_normcore_M.jpg                        0.512953
W_17123_19_normcore_M.jpg                        0.521476
…                                                   …
W_25400_19_normcore_M.jpg                        0.467378
W_17108_19_normcore_M.jpg                        0.554438
W_09233_60_mods_M.jpg                            0.393971
W_12814_19_normcore_M.jpg                        0.505399
W_06682_00_metrosexual_M.jpg                     0.499949

                               W_30988_90_kitsch_W.jpg  \
```

```
W_13846_60_minimal_W.jpg                    0.452228
W_10981_50_feminine_W.jpg                   0.544127
W_14489_19_normcore_W.jpg                   0.544305
W_16836_19_normcore_M.jpg                   0.498940
W_17123_19_normcore_M.jpg                   0.523792
…                                              …
W_25400_19_normcore_M.jpg                   0.470208
W_17108_19_normcore_M.jpg                   0.611663
W_09233_60_mods_M.jpg                       0.500380
W_12814_19_normcore_M.jpg                   0.523485
W_06682_00_metrosexual_M.jpg                0.843724

                            W_39164_00_oriental_W.jpg  \

W_13846_60_minimal_W.jpg                      0.706288
W_10981_50_feminine_W.jpg                     0.937815
W_14489_19_normcore_W.jpg                     0.671321
W_16836_19_normcore_M.jpg                     0.406927
W_17123_19_normcore_M.jpg                     0.398665
…                                                …
W_25400_19_normcore_M.jpg                     0.398322
W_17108_19_normcore_M.jpg                     0.394036
W_09233_60_mods_M.jpg                         0.346020
W_12814_19_normcore_M.jpg                     0.416810
W_06682_00_metrosexual_M.jpg                  0.456766

                            W_44330_10_sportivecasual_W.jpg  …  \
                                                                …
W_13846_60_minimal_W.jpg                               0.402619  …
W_10981_50_feminine_W.jpg                              0.525235  …
W_14489_19_normcore_W.jpg                              0.557363  …
W_16836_19_normcore_M.jpg                              0.419327  …
W_17123_19_normcore_M.jpg                              0.453136  …
…                                                         …   …
W_25400_19_normcore_M.jpg                              0.364277  …
W_17108_19_normcore_M.jpg                              0.430416  …
W_09233_60_mods_M.jpg                                  0.261456  …
W_12814_19_normcore_M.jpg                              0.480167  …
W_06682_00_metrosexual_M.jpg                           0.604209  …

                            W_16067_80_bold_M.jpg  W_20593_70_punk_W.jpg  \

W_13846_60_minimal_W.jpg                 0.799044                0.712388
W_10981_50_feminine_W.jpg                0.638641                0.720997
W_14489_19_normcore_W.jpg                0.584284                0.665826
W_16836_19_normcore_M.jpg                0.689034                0.442448
```

W_17123_19_normcore_M.jpg	0.694488	0.430364
…	…	…
W_25400_19_normcore_M.jpg	0.642619	0.430696
W_17108_19_normcore_M.jpg	0.690477	0.517597
W_09233_60_mods_M.jpg	0.588320	0.376860
W_12814_19_normcore_M.jpg	0.640152	0.473857
W_06682_00_metrosexual_M.jpg	0.569013	0.655061

	W_41341_19_lounge_W.jpg \
W_13846_60_minimal_W.jpg	0.747899
W_10981_50_feminine_W.jpg	0.701771
W_14489_19_normcore_W.jpg	0.773254
W_16836_19_normcore_M.jpg	0.497870
W_17123_19_normcore_M.jpg	0.560454
…	…
W_25400_19_normcore_M.jpg	0.449661
W_17108_19_normcore_M.jpg	0.546617
W_09233_60_mods_M.jpg	0.336279
W_12814_19_normcore_M.jpg	0.509045
W_06682_00_metrosexual_M.jpg	0.590116

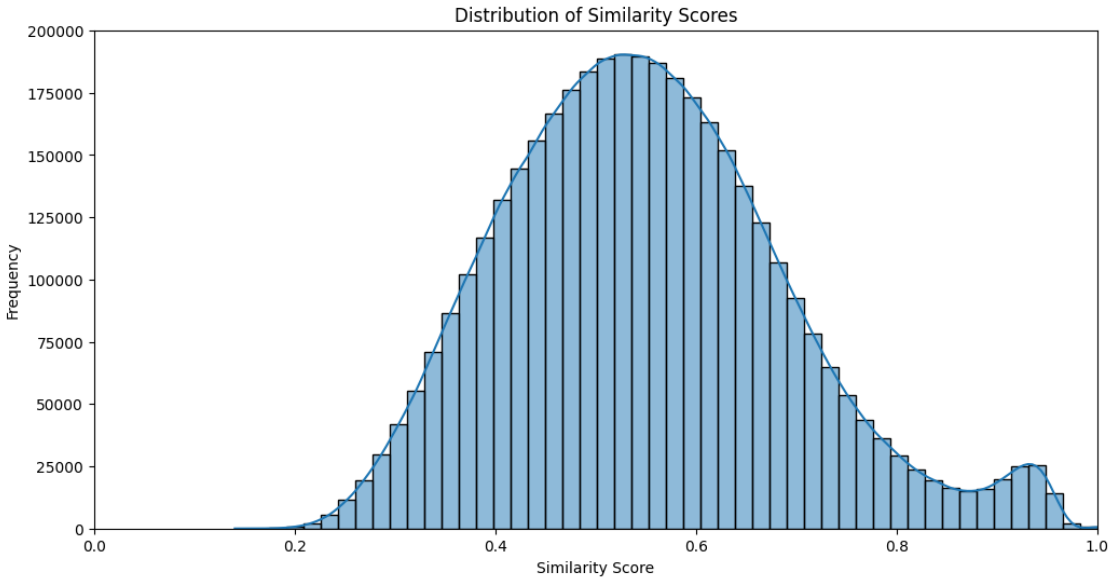
	W_29693_19_normcore_M.jpg \
W_13846_60_minimal_W.jpg	0.587182
W_10981_50_feminine_W.jpg	0.511369
W_14489_19_normcore_W.jpg	0.647131
W_16836_19_normcore_M.jpg	0.883176
W_17123_19_normcore_M.jpg	0.882393
…	…
W_25400_19_normcore_M.jpg	0.833703
W_17108_19_normcore_M.jpg	0.850023
W_09233_60_mods_M.jpg	0.514431
W_12814_19_normcore_M.jpg	0.879441
W_06682_00_metrosexual_M.jpg	0.621707

	W_17783_80_bold_M.jpg	W_24517_70_hippie_M.jpg \
W_13846_60_minimal_W.jpg	0.657697	0.284166
W_10981_50_feminine_W.jpg	0.634490	0.361793
W_14489_19_normcore_W.jpg	0.538943	0.527656
W_16836_19_normcore_M.jpg	0.623670	0.532336
W_17123_19_normcore_M.jpg	0.678168	0.607623
…	…	…
W_25400_19_normcore_M.jpg	0.567004	0.531308
W_17108_19_normcore_M.jpg	0.652534	0.610223
W_09233_60_mods_M.jpg	0.478550	0.504567

```
W_12814_19_normcore_M.jpg                 0.628956                 0.558276
W_06682_00_metrosexual_M.jpg              0.576781                 0.711815

                            W_12635_70_hippie_M.jpg  W_12817_50_ivy_M.jpg  \

W_13846_60_minimal_W.jpg                  0.278237             0.427809
W_10981_50_feminine_W.jpg                 0.315936             0.462535
W_14489_19_normcore_W.jpg                 0.530435             0.709773
W_16836_19_normcore_M.jpg                 0.533661             0.635083
W_17123_19_normcore_M.jpg                 0.571459             0.590698
…                                            …                    …
W_25400_19_normcore_M.jpg                 0.527698             0.651332
W_17108_19_normcore_M.jpg                 0.570484             0.583682
W_09233_60_mods_M.jpg                     0.484216             0.624612
W_12814_19_normcore_M.jpg                 0.571421             0.711849
W_06682_00_metrosexual_M.jpg              0.620586             0.468826

                            W_25649_19_normcore_M.jpg  \

W_13846_60_minimal_W.jpg                    0.594407
W_10981_50_feminine_W.jpg                   0.617854
W_14489_19_normcore_W.jpg                   0.561760
W_16836_19_normcore_M.jpg                   0.487970
W_17123_19_normcore_M.jpg                   0.517074
…                                              …
W_25400_19_normcore_M.jpg                   0.455389
W_17108_19_normcore_M.jpg                   0.561777
W_09233_60_mods_M.jpg                       0.564366
W_12814_19_normcore_M.jpg                   0.523550
W_06682_00_metrosexual_M.jpg                0.739916

                            W_25790_90_hiphop_M.jpg

W_13846_60_minimal_W.jpg                  0.430533
W_10981_50_feminine_W.jpg                 0.612102
W_14489_19_normcore_W.jpg                 0.502561
W_16836_19_normcore_M.jpg                 0.492199
W_17123_19_normcore_M.jpg                 0.586279
…                                            …
W_25400_19_normcore_M.jpg                 0.468339
W_17108_19_normcore_M.jpg                 0.493322
W_09233_60_mods_M.jpg                     0.414932
W_12814_19_normcore_M.jpg                 0.537728
W_06682_00_metrosexual_M.jpg              0.518468

[4070 rows x 951 columns]
```

[ ]:
```python
#    item
import matplotlib.pyplot as plt
import seaborn as sns

#        1
similarity_values = similarity_df.values.flatten()

#
print("    :")
print(f" : {similarity_values.mean():.4f}")
print(f"  : {np.median(similarity_values):.4f}")
print(f"   : {similarity_values.std():.4f}")
print(f"  : {similarity_values.min():.4f}")
print(f"  : {similarity_values.max():.4f}")

#
plt.figure(figsize=(12, 6))

#
sns.histplot(similarity_values, kde=True, bins=50)

plt.title("Distribution of Similarity Scores")
plt.xlabel("Similarity Score")
plt.ylabel("Frequency")

# x
plt.xlim(0, 1)
plt.show()
```

```
   :
 : 0.5493
  : 0.5406
   : 0.1402
  : 0.1402
  : 1.0000
```



0.75 - mean + 1.5 * std = 0.7596

(1) top1 item

[ ]:
```
#         item
def _method1(top_similar_items, user_rated_df):
    predicted_rating = user_rated_df.loc[user_rated_df['  '] ==␣
 ↪top_similar_items.index[0]]['      '].values[0]
    return predicted_rating

def predict_preference(v_top100_pref, t_top100_pref, similarity_df):
    pred_list = []

    for i in range(len(v_top100_pref)):
        userID = v_top100_pref.iloc[i, 0]
        item = v_top100_pref.iloc[i, 1]

        # userID              items
        user_rated_df = t_top100_pref.loc[t_top100_pref['   ID'] == userID]
        rated_items = user_rated_df['  '].values

        #              item
        similar_items = similarity_df.loc[rated_items, item]

        #        item
        similar_items = similar_items.sort_values(ascending=False)
        top_similar_items = similar_items.head(1)
```

```
        #
        predicted_rating = _method1(top_similar_items, user_rated_df)

        #
        pred_list.append([userID, item, predicted_rating])

    pred_df = pd.DataFrame(pred_list, columns=['   ID', '  ', '     '])

    return pred_df

#
pred_df = predict_preference(v_top100_pref, t_top100_pref, similarity_df)
pred_df
```

[ ]:

	ID		
0	368	W_06864_10_sportivecasual_M.jpg	1
1	368	W_04678_50_ivy_M.jpg	1
2	368	W_16034_80_bold_M.jpg	-1
3	368	W_00551_19_normcore_M.jpg	1
4	368	W_01703_00_metrosexual_M.jpg	1
…	…	…	…
1095	67975	W_17738_80_bold_M.jpg	1
1096	67975	T_21986_70_hippie_M.jpg	-1
1097	67975	T_21988_70_hippie_M.jpg	-1
1098	67975	W_52578_50_ivy_M.jpg	-1
1099	67975	W_26965_90_hiphop_M.jpg	-1

[1100 rows x 3 columns]

[ ]:
```
# v_top100_pref  merge
merge_df = v_top100_pref.merge(pred_df, on=['   ID', '  '], how='left')

#       (Accuracy / Precision / Recall)
from sklearn.metrics import accuracy_score, precision_score, recall_score

accuracy = accuracy_score(merge_df['      '], merge_df['     '])
precision = precision_score(merge_df['      '], merge_df['     '])
recall = recall_score(merge_df['      '], merge_df['     '])

print(f"Accuracy: {accuracy:.4f}, Precision: {precision:.4f}, Recall: {recall:.
 ↪4f}")

merge_df
```

```
Accuracy: 0.8191, Precision: 0.7763, Recall: 0.7710
```

[ ]:

	ID				
0	368	W_06864_10_sportivecasual_M.jpg	1	1	
1	368	W_04678_50_ivy_M.jpg	1	1	
2	368	W_16034_80_bold_M.jpg	-1	-1	
3	368	W_00551_19_normcore_M.jpg	1	1	
4	368	W_01703_00_metrosexual_M.jpg	1	1	
…	…	…	…	…	
1095	67975	W_17738_80_bold_M.jpg	1	1	
1096	67975	T_21986_70_hippie_M.jpg	-1	-1	
1097	67975	T_21988_70_hippie_M.jpg	1	-1	
1098	67975	W_52578_50_ivy_M.jpg	1	-1	
1099	67975	W_26965_90_hiphop_M.jpg	-1	-1	

[1100 rows x 4 columns]

(2) n items voting

[ ]:
```
#     15 items       voting
def _method2(top_similar_items, user_rated_df):
    pref_lists = []
    for similar_item in top_similar_items.index:
        rating = user_rated_df.loc[user_rated_df['  ']==similar_item]['      '].
 ↪values[0]
        pref_lists.append(rating)
    # pref_lists
    predicted_rating = max(pref_lists, key=pref_lists.count)

    return predicted_rating

def predict_preference(v_top100_pref, t_top100_pref, similarity_df):
    pred_list = []

    for i in range(len(v_top100_pref)):
        userID = v_top100_pref.iloc[i, 0]
        item = v_top100_pref.iloc[i, 1]

        # userID              items
        user_rated_df = t_top100_pref.loc[t_top100_pref['   ID'] == userID]
        rated_items = user_rated_df['  '].values

        #              item
        similar_items = similarity_df.loc[rated_items, item]

        #     15  item
        similar_items = similar_items.sort_values(ascending=False)
        top_similar_items = similar_items.head(15)
```

```
        #
        predicted_rating = _method2(top_similar_items, user_rated_df)

        #
        pred_list.append([userID, item, predicted_rating])

    pred_df = pd.DataFrame(pred_list, columns=['   ID', '  ', '     '])

    return pred_df

#
pred_df = predict_preference(v_top100_pref, t_top100_pref, similarity_df)
pred_df
```

[ ]:

	ID		
0	368	W_06864_10_sportivecasual_M.jpg	-1
1	368	W_04678_50_ivy_M.jpg	1
2	368	W_16034_80_bold_M.jpg	-1
3	368	W_00551_19_normcore_M.jpg	-1
4	368	W_01703_00_metrosexual_M.jpg	1
…	…	…	…
1095	67975	W_17738_80_bold_M.jpg	-1
1096	67975	T_21986_70_hippie_M.jpg	-1
1097	67975	T_21988_70_hippie_M.jpg	-1
1098	67975	W_52578_50_ivy_M.jpg	-1
1099	67975	W_26965_90_hiphop_M.jpg	-1

[1100 rows x 3 columns]

[ ]:
```
# v_top100_pref  merge
merge_df = v_top100_pref.merge(pred_df, on=['   ID', '  '], how='left')

#       (Accuracy / Precision / Recall)
from sklearn.metrics import accuracy_score, precision_score, recall_score

accuracy = accuracy_score(merge_df['      '], merge_df['     '])
precision = precision_score(merge_df['      '], merge_df['     '])
recall = recall_score(merge_df['      '], merge_df['     '])

print(f"Accuracy: {accuracy:.4f}, Precision: {precision:.4f}, Recall: {recall:.
 ↪4f}")

merge_df
```

```
Accuracy: 0.6909, Precision: 0.6384, Recall: 0.5283
```

[ ]:

	ID			
0	368	W_06864_10_sportivecasual_M.jpg	1	-1
1	368	W_04678_50_ivy_M.jpg	1	1
2	368	W_16034_80_bold_M.jpg	-1	-1
3	368	W_00551_19_normcore_M.jpg	1	-1
4	368	W_01703_00_metrosexual_M.jpg	1	1
…	…	…	…	…
1095	67975	W_17738_80_bold_M.jpg	1	-1
1096	67975	T_21986_70_hippie_M.jpg	-1	-1
1097	67975	T_21988_70_hippie_M.jpg	1	-1
1098	67975	W_52578_50_ivy_M.jpg	1	-1
1099	67975	W_26965_90_hiphop_M.jpg	-1	-1

[1100 rows x 4 columns]

(3)

[ ]:
```
# top_similar_items                    (       )
def _method3(top_similar_items, user_rated_df):
    weighted_ratings_sum = 0    #      ( )
    similarity_sum = 0          #      ( )

    #           top_similar_items
    for similar_item in top_similar_items.index:
        similarity = top_similar_items[similar_item]
        rating = user_rated_df.loc[user_rated_df['  '] == similar_item]['
 ↪ '].values[0]
        weighted_ratings_sum += similarity * rating
        similarity_sum += similarity

    if similarity_sum != 0:
        predicted_rating = weighted_ratings_sum / similarity_sum
    else:
        predicted_rating = 0

    return predicted_rating

def predict_preference(v_top100_pref, t_top100_pref, similarity_df):
    pred_list = []

    for i in range(len(v_top100_pref)):
        userID = v_top100_pref.iloc[i, 0]
        item = v_top100_pref.iloc[i, 1]

        # userID              items
        user_rated_df = t_top100_pref.loc[t_top100_pref['   ID'] == userID]
        rated_items = user_rated_df['  '].values
```

```
        #              item
        similar_items = similarity_df.loc[rated_items, item]

        #              (!      )
        similar_items = similar_items.sort_values(ascending=False)
        top_similar_items = similar_items[similar_items >= 0.75]    #   : 0.75

        #     (         )
        predicted_rating = _method3(top_similar_items, user_rated_df)

        #
        pred_list.append([userID, item, predicted_rating])

    pred_df = pd.DataFrame(pred_list, columns=['   ID', '  ', '     '])

    return pred_df

#
pred_df = predict_preference(v_top100_pref, t_top100_pref, similarity_df)
pred_df
```

[ ]:

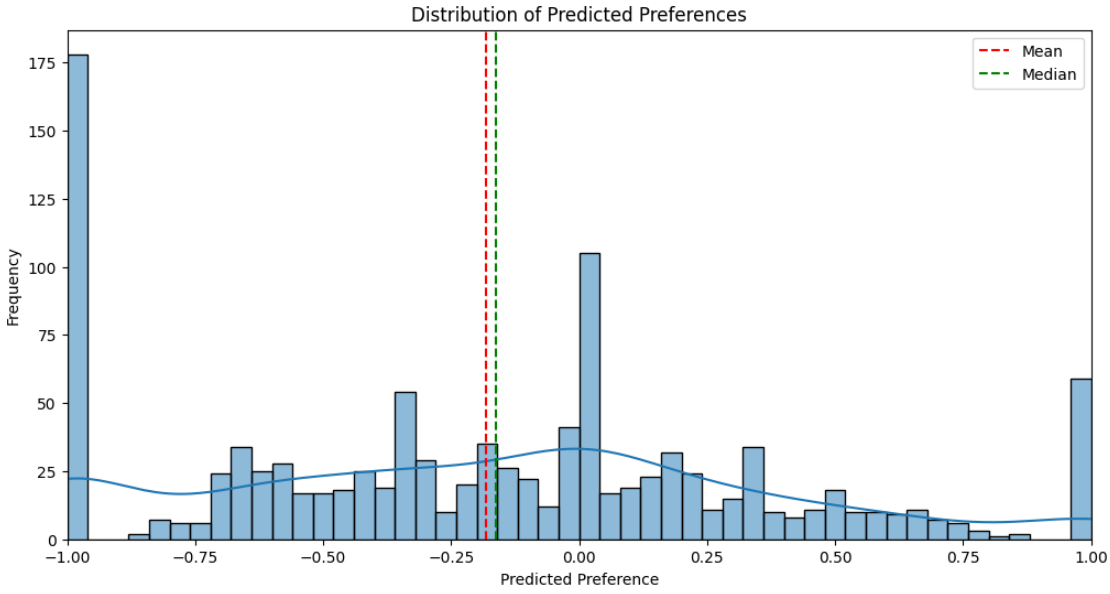
	ID		
0	368	W_06864_10_sportivecasual_M.jpg	-0.001472
1	368	W_04678_50_ivy_M.jpg	0.340583
2	368	W_16034_80_bold_M.jpg	-0.332617
3	368	W_00551_19_normcore_M.jpg	0.197463
4	368	W_01703_00_metrosexual_M.jpg	0.033871
…	…	…	…
1095	67975	W_17738_80_bold_M.jpg	-0.577791
1096	67975	T_21986_70_hippie_M.jpg	-0.603113
1097	67975	T_21988_70_hippie_M.jpg	-0.093612
1098	67975	W_52578_50_ivy_M.jpg	-0.463443
1099	67975	W_26965_90_hiphop_M.jpg	-0.671873

[1100 rows x 3 columns]

[ ]:
```
#
import matplotlib.pyplot as plt
import seaborn as sns

#
stats = pred_df['     '].describe()

#
print("<     >")
```

```
print(f" : {stats['count']:.0f} |  : {stats['mean']:.4f} |  : {stats['50%']:.
  ↪4f} |  : {stats['std']:.4f}")


#
plt.figure(figsize=(12, 6))

#     KDE
sns.histplot(data=pred_df, x='    ', kde=True, bins=50)

#
plt.axvline(stats['mean'], color='r', linestyle='--', label='Mean')
plt.axvline(stats['50%'], color='g', linestyle='--', label='Median')

plt.title("Distribution of Predicted Preferences")
plt.xlabel("Predicted Preference")
plt.ylabel("Frequency")
plt.legend()

# x
plt.xlim(-1, 1)

plt.show()
```

```
<     >
 : 1100 |  : -0.1831 |  : -0.1630 |  : 0.5522
```



[ ]:
```
# pred_df    0          '    '
pred_df['    '] = pred_df['    '].apply(lambda x: 1 if x > 0 else -1)
pred_df
```

[ ]:

	ID		
0	368	W_06864_10_sportivecasual_M.jpg	-1
1	368	W_04678_50_ivy_M.jpg	1
2	368	W_16034_80_bold_M.jpg	-1
3	368	W_00551_19_normcore_M.jpg	1
4	368	W_01703_00_metrosexual_M.jpg	1
…	…	…	…
1095	67975	W_17738_80_bold_M.jpg	-1
1096	67975	T_21986_70_hippie_M.jpg	-1
1097	67975	T_21988_70_hippie_M.jpg	-1
1098	67975	W_52578_50_ivy_M.jpg	-1
1099	67975	W_26965_90_hiphop_M.jpg	-1

[1100 rows x 3 columns]

[ ]:
```
# v_top100_pref  merge
merge_df = v_top100_pref.merge(pred_df, on=['   ID', '  '], how='left')

#       (Accuracy / Precision / Recall)
from sklearn.metrics import accuracy_score, precision_score, recall_score

accuracy = accuracy_score(merge_df['      '], merge_df['    '])
precision = precision_score(merge_df['      '], merge_df['    '])
recall = recall_score(merge_df['      '], merge_df['    '])

print(f"Accuracy: {accuracy:.4f}, Precision: {precision:.4f}, Recall: {recall:.
↪4f}")

merge_df
```

```
Accuracy: 0.7291, Precision: 0.6810, Recall: 0.6100
```

[ ]:

	ID			
0	368	W_06864_10_sportivecasual_M.jpg	1	-1
1	368	W_04678_50_ivy_M.jpg	1	1
2	368	W_16034_80_bold_M.jpg	-1	-1
3	368	W_00551_19_normcore_M.jpg	1	1
4	368	W_01703_00_metrosexual_M.jpg	1	1
…	…	…	…	…
1095	67975	W_17738_80_bold_M.jpg	1	-1
1096	67975	T_21986_70_hippie_M.jpg	-1	-1
1097	67975	T_21988_70_hippie_M.jpg	1	-1
1098	67975	W_52578_50_ivy_M.jpg	1	-1

```
1099   67975          W_26965_90_hiphop_M.jpg         -1       -1

[1100 rows x 4 columns]
```

[ ]: