Mission2

October 31, 2024

[1]:
```
from google.colab import drive
import os, sys

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

#
%cd /content/drive/MyDrive/

#
!mkdir -p Mission2
```

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

1 Mission 2.

1.1 Mission 2-1

1. “{W/T}{ *ID*}{ }{ }{ }_{ ID}.json” .
2. & .
3. ID .
4. Train Validation .

[2]:
```
import os
import matplotlib.pyplot as plt
import pandas as pd
```

1.1.1 1.

1. DataFrame

[3]:
```
from IPython.display import display_html

t_img_path = "dataset/origin_dataset/training_image"
v_img_path = "dataset/origin_dataset/validation_image"

#
t_image_list = os.listdir(t_img_path)
```

```
v_image_list = os.listdir(v_img_path)

# train validation
print(f"train image: {len(t_image_list)}, validation image:␣
 ↪{len(v_image_list)}")

#     ->     Pandas
def img2data(image_list):
    data = []

    for i in range(len(image_list)):
        meta_data = image_list[i].split('_')

        row = {
            'file_name': image_list[i][:-4],    # ".jpg"
            'wt': meta_data[0],
            'image_id': meta_data[1],
            'time': meta_data[2],
            'style': meta_data[3],
            'gender': meta_data[4][0]
        }
        data.append(row)
    return pd.DataFrame(data)

#       HTML
def display_left(*args):
    html_str = ''
    for df in args:
        html_str += f'<div style="margin-right:30px;">{df.to_html()}</div>'
    display_html(f'<div style="display: flex;">{html_str}</div>', raw=True)

#
t_img_df = img2data(t_image_list)
v_img_df = img2data(v_image_list)

#
display_left(t_img_df.head(), v_img_df.head())
```

```
train image: 4070, validation image: 951
```

2. Label DataFrame 1. Label Mount Google Drive I/O Timeout .

2. training_label, validation_label /context (colab) .

[4]:
```
# dataset/
!mkdir -p /content/dataset
```

```
!cp -r /content/drive/MyDrive/dataset/origin_dataset/all_label.zip /content/
  ↪dataset
!unzip -o -qq /content/dataset/all_label.zip -d /content/dataset/
```

[5]:
```
from IPython.display import display_html

# t_label_path = "dataset/origin_dataset/training_label"
# v_label_path = "dataset/origin_dataset/validation_label"

# Google Drive  os.listdir  timeout     /content
t_label_path = "/content/dataset/training_label"
v_label_path = "/content/dataset/validation_label"

#
t_label_list = os.listdir(t_label_path)
v_label_list = os.listdir(v_label_path)

# train  validation     label
print(f"train label: {len(t_label_list)}, validation label:␣
  ↪{len(v_label_list)}")

#     ->     Pandas
def label2data(label_list):
    data = []

    for i in range(len(label_list)):
        meta_data = label_list[i].split('_')

        if len(meta_data) < 2:
            print(meta_data)
        row = {
            'file_name': label_list[i][:-5],    # ".json"
            'wt': meta_data[0],
            'image_id': meta_data[1],
            'time': meta_data[2],
            'style': meta_data[3],
            'gender': meta_data[4],
            'survey_id': meta_data[5].split('.')[0],
        }
        data.append(row)
    return pd.DataFrame(data)

#
t_lbl_df = label2data(t_label_list)
v_lbl_df = label2data(v_label_list)

#       HTML
```

```
def display_left(*args):
    html_str = ''
    for df in args:
        html_str += f'<div style="margin-right:30px;">{df.to_html()}</div>'
    display_html(f'<div style="display: flex;">{html_str}</div>', raw=True)

#
display_left(t_lbl_df.head(), v_lbl_df.head())
```

```
train label: 211346, validation label: 36383
```

3. Label

ID filtering

[6]:
```
#   ID  "00004"
#     ID         ("T_00004_90_hiphop_M"/"W_00004_50_ivy_M")
#      imageID  wt+image_id    ( 1-1  ID            )
t_lbl_df[t_lbl_df['image_id'] == '00004']
```

[6]:

	file_name	wt	image_id	time	style	gender	survey_id
21238	W_00004_50_ivy_M_153260	W	00004	50	ivy	M	153260
44261	W_00004_50_ivy_M_067526	W	00004	50	ivy	M	067526
85488	T_00004_90_hiphop_M_203010	T	00004	90	hiphop	M	203010
88028	T_00004_90_hiphop_M_206164	T	00004	90	hiphop	M	206164
88201	W_00004_50_ivy_M_185102	W	00004	50	ivy	M	185102
131236	W_00004_50_ivy_M_134675	W	00004	50	ivy	M	134675
134892	W_00004_50_ivy_M_179491	W	00004	50	ivy	M	179491
169817	T_00004_90_hiphop_M_071538	T	00004	90	hiphop	M	071538
170260	W_00004_50_ivy_M_060212	W	00004	50	ivy	M	060212
197689	W_00004_50_ivy_M_092597	W	00004	50	ivy	M	092597

labeling data filtering

[7]:
```
#         filtering
def filtering_labels(img_df, lbl_df):
    lbl_df['survey_imgname'] = lbl_df['file_name'].apply(lambda x: x[:-7]) #␣
 ↪ ID
    img_filenames = img_df['file_name'] #
    filtered_lbls = lbl_df[lbl_df['survey_imgname'].isin(img_filenames)].
↪reset_index(drop=True)    #

    return filtered_lbls.drop(columns=['survey_imgname'])

filtered_t_lbl = filtering_labels(t_img_df, t_lbl_df)
filtered_v_lbl = filtering_labels(v_img_df, v_lbl_df)

# filtering
```

```
print(f"filtered train label: {len(filtered_t_lbl)}, filtered validation label:␣
 ↪{len(filtered_v_lbl)}")
display_left(filtered_t_lbl.head(), filtered_v_lbl.head())
```

```
filtered train label: 16096, filtered validation label: 4105
```

filtering labeling data

[9]:
```
import shutil
from tqdm.notebook import tqdm

#
def filtering_label(df, dest_dir, state='train'):
    src_folder = '/content/dataset' #

    if state == 'train':
        folder = 'training_label'
    elif state == 'validation':
        folder = 'validation_label'

    #
    src_path = os.path.join(src_folder, folder)
    dest_path = os.path.join(dest_dir, folder)

    #
    if not os.path.exists(dest_path):
        os.makedirs(dest_path)

    #
    processed_count = 0
    error_count = 0

    file_list = os.listdir(src_path)

    #
    for file in tqdm(df, desc=f"Processing {folder}", unit='file'):

        file += '.json'
        src_file_path = os.path.join(src_path, file)
        dest_file_path = os.path.join(dest_path, file)

        try:
            if file in file_list:
                shutil.copy2(src_file_path, dest_file_path)
                processed_count += 1
            else:
                continue
        except Exception as e:
```

```
            error_count += 1
            print(f"Error: {e}")

    print(f"\nFolder: {folder}")
    print(f"Processed files: {processed_count}")
    print(f"Errors encountered: {error_count}\n")


dest_dir = "/content/filtered_label" #
if not os.path.exists(dest_dir):
    os.makedirs(dest_dir)

t_filenames = filtered_t_lbl['file_name'].values
v_filenames = filtered_v_lbl['file_name'].values

filtering_label(t_filenames, dest_dir, state='train')
filtering_label(v_filenames, dest_dir, state='validation')
```

```
Processing training_label:   0%|          | 0/16096 [00:00<?, ?file/s]


Folder: training_label
Processed files: 16096
Errors encountered: 0


Processing validation_label:   0%|          | 0/4105 [00:00<?, ?file/s]


Folder: validation_label
Processed files: 4105
Errors encountered: 0
```

4. 1. 1
2. 3-1 . “W_27750_60_mods_M_146696.json” .

[42]:
```
print(f"filtered_training_label: {len(os.listdir('/content/filtered_label/
  ↪training_label'))}")
print(f"filtered_validation_label: {len(os.listdir('/content/filtered_label/
  ↪validation_label'))}")

# W_27750_60_mods_M_146696.json
!rm /content/filtered_label/training_label/W_27750_60_mods_M_146696.json
!rm /content/filtered_label/validation_label/W_27750_60_mods_M_146696.json
```

```
print(f"filtered_training_label: {len(os.listdir('/content/filtered_label/
  ↪training_label'))}")
print(f"filtered_validation_label: {len(os.listdir('/content/filtered_label/
  ↪validation_label'))}")
```

```
filtered_training_label: 16095
filtered_validation_label: 4105
rm: cannot remove
'/content/filtered_label/training_label/W_27750_60_mods_M_146696.json': No such
file or directory
filtered_training_label: 16095
filtered_validation_label: 4104
```

1.1.2 2. &

[43]:
```
import os
import pandas as pd
from IPython.display import display_html

#
training_label_list = os.listdir('/content/filtered_label/training_label')
validation_label_list = os.listdir('/content/filtered_label/validation_label')

# train validation label
print(f"train label: {len(training_label_list)}, validation label:␣
  ↪{len(validation_label_list)}")

#  ->  Pandas
def path2data(label_list):
    data = []

    for i in range(len(label_list)):
        meta_data = label_list[i].split('_')

        row = {
            'wt': meta_data[0],
            'image_id': meta_data[1],
            'time': meta_data[2],
            'style': meta_data[3],
            'gender': meta_data[4],
            'survey_id': meta_data[5].split('.')[0]
        }
        data.append(row)
    return pd.DataFrame(data)

#  HTML
def display_left(*args):
```

```
    html_str = ''
    for df in args:
        html_str += f'<div style="margin-right:30px;">{df.to_html()}</div>'
    display_html(f'<div style="display: flex;">{html_str}</div>', raw=True)


#
training_df = path2data(training_label_list)
validation_df = path2data(validation_label_list)

#
display_left(training_df.head(), validation_df.head())

#
training_df.to_csv('Mission2/training_df.csv', index=False)
validation_df.to_csv('Mission2/validation_df.csv', index=False)
```

```
train label: 16095, validation label: 4104
```

groupby

[44]:
```
#  &
def get_gender_style_stats(df, state='train'):
    #
    group_data = df[['gender', 'style', 'survey_id']].groupby(['gender',␣
 ↪'style']).count()
    #      column
    group_data.rename(columns={'survey_id': f'{state}_count'}, inplace=True)

    return group_data

#
training_stats_data = get_gender_style_stats(training_df, state='train')
validation_stats_data = get_gender_style_stats(validation_df,␣
 ↪state='validation')

#
display_left(training_stats_data, validation_stats_data)

#
training_stats_data.to_csv('dataset/training_count_data.csv')
validation_stats_data.to_csv('dataset/validation_count_data.csv')

#
training_stats_data.to_csv('Mission2/training_count_data.csv')
validation_stats_data.to_csv('Mission2/validation_count_data.csv')
```

1.1.3 3.

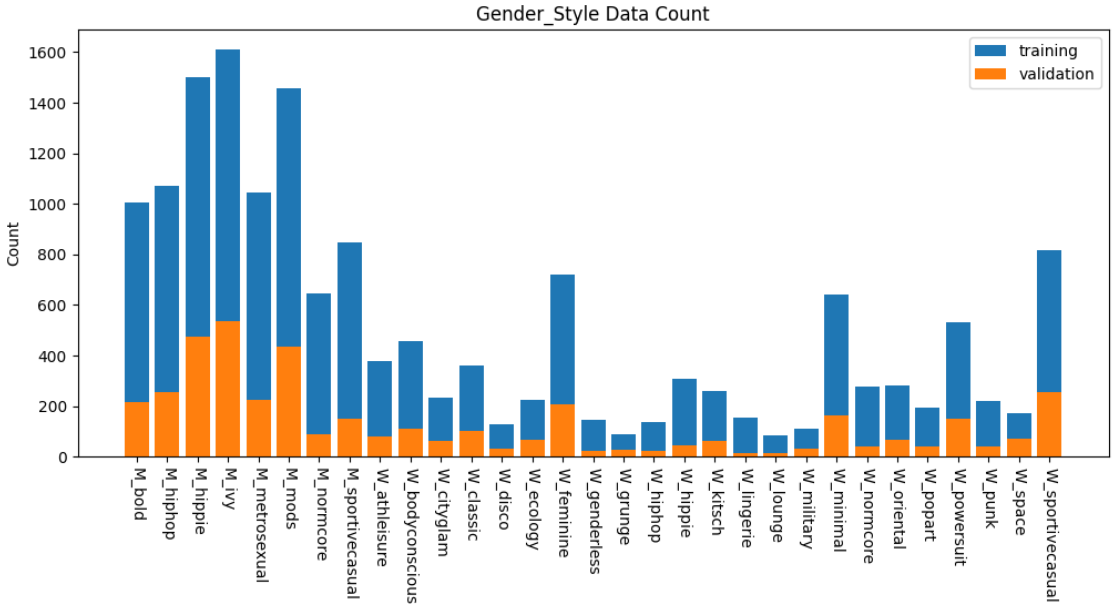
[45]:
```
import matplotlib.pyplot as plt

# Group
training_stats_data_re = training_stats_data.reset_index()
validation_stats_data_re = validation_stats_data.reset_index()

#          Gender  Style
training_stats_data_re['name'] = training_stats_data_re['gender'] + '_' +␣
 ↪training_stats_data_re['style']
validation_stats_data_re['name'] = validation_stats_data_re['gender'] + '_' +␣
 ↪validation_stats_data_re['style']

#
plt.figure(figsize=(12, 5))
plt.title('Gender_Style Data Count')
plt.ylabel('Count')
plt.xticks(rotation=270)
plt.bar(training_stats_data_re['name'], training_stats_data_re['train_count'],␣
 ↪label="training")
plt.bar(validation_stats_data_re['name'],␣
 ↪validation_stats_data_re['validation_count'], label="validation")

plt.legend()
plt.savefig('Mission2/gender_style_data_count.png', bbox_inches='tight')    #␣
 ↪
plt.show()
```



1.2 Mission 2-2

1.2.1 1. label “ ID” “ ”

[46]:
```
import os
import json
import pandas as pd

# filtering  JSON
train_path = '/content/filtered_label/training_label'
validation_path = '/content/filtered_label/validation_label'

def preference(label_me, output_file):
    # JSON
    folder_path = label_me

    #
    records = []

    #       JSON
    for file_name in tqdm(os.listdir(folder_path)):
        if file_name.endswith('.json'):
            file_path = os.path.join(folder_path, file_name)

            # JSON
            with open(file_path, 'r') as f:
                try:
                    data = json.load(f)

                    #      (user   )
                    r_id = data['user']['R_id']

                    # item     imgName  Q5
                    img_name = data['item']['imgName']
                    q5_value = data['item']['survey']['Q5']

                    # Q5
                    if q5_value == 2:
                        preference = '    '
                    elif q5_value == 1:
                        preference = '     '
                    records.append({'   ID': r_id, '  ': img_name, '      ':␣
 ↪preference})

                except json.JSONDecodeError as e:
```

```
                    print(f"Error decoding JSON from file {file_name}: {e}")
                except KeyError as e:
                    print(f"Missing key {e} in file {file_name}")

    #
    df = pd.DataFrame(records)

    #    ID
    df.sort_values(by='   ID', inplace=True)
    df.reset_index(drop=True, inplace=True)

    #
    duplicate_rows = df[df.duplicated()]
    #
    if not duplicate_rows.empty:
        df.drop_duplicates(inplace=True)
        print(f"     {len(duplicate_rows)}      .")

    #        (CSV  )
    df.to_csv(output_file, encoding='utf-8-sig', index=False)

    print(f"   {output_file}     .")

    return df  #

#
df_train = preference(train_path, 'Mission2/train_preference.csv')
df_val = preference(validation_path, 'Mission2/val_preference.csv')
```

```
  0%|          | 0/16095 [00:00<?, ?it/s]

    12      .
   Mission2/train_preference.csv      .

  0%|          | 0/4104 [00:00<?, ?it/s]

    5      .
   Mission2/val_preference.csv      .
```

1.2.2 2. train/valid R_id 100

**

[47]:
```
#
df_train
```

[47]:

	ID
0	12 W_03412_50_classic_W.jpg
1	25 W_12740_00_metrosexual_M.jpg

```
2          26        W_18990_50_feminine_W.jpg
3          27        W_17260_19_normcore_M.jpg
4          27          W_03007_70_hippie_M.jpg
…        …                                …       …
16090   68396          W_24537_70_hippie_M.jpg
16091   68398            W_25360_80_bold_M.jpg
16092   68729  T_00456_10_sportivecasual_M.jpg
16093   68743  T_01883_10_sportivecasual_M.jpg
16094   68747  T_02527_10_sportivecasual_M.jpg

[16083 rows x 3 columns]
```

[48]:
```
#    ID
train_count = df_train.groupby(['   ID']).size()
val_count = df_val.groupby(['   ID']).size()

#
train_count.name = 'train      '
val_count.name = 'val      '

#
df_sum = pd.concat([train_count, val_count],axis=1)
df_sum = df_sum.fillna(0).astype(int)    #    0
df_sum[' '] = df_sum['train      '] + df_sum['val      ']    # ' '
df_sum = df_sum.sort_values(by=' ', ascending=False)    # ' '

# user      df
df_sum
```

[48]:
```
         train        val
   ID
64747              45           15  60
63405              44           14  58
64561              46           12  58
64346              46           12  58
65139              46           12  58
…                …           …   ..
65051               0            1   1
65115               0            1   1
65125               0            1   1
65146               0            1   1
65285               0            1   1

[3480 rows x 3 columns]
```

[49]:
```
# df_sum          100     ID
top_100_ids = df_sum.head(100).index.tolist()
```

```
#
top100_train_df = df_train[df_train['   ID'].isin(top_100_ids)].
 ↪reset_index(drop=True)
top100_val_df = df_val[df_val['   ID'].isin(top_100_ids)].reset_index(drop=True)

#
top100_train_df.to_csv('Mission2/top100_train_preference.csv', index=False)
top100_val_df.to_csv('Mission2/top100_val_preference.csv', index=False)

#          100
top100_train_df['   ID'].nunique(), top100_val_df['   ID'].nunique()
```

[49]: (100, 100)

[50]:
```
#   100
top100_train_df
```

[50]:

	ID	
0	368	W_16264_80_bold_M.jpg
1	368	W_15340_50_ivy_M.jpg
2	368	W_02714_00_metrosexual_M.jpg
3	368	W_04604_00_metrosexual_M.jpg
4	368	W_16403_10_sportivecasual_M.jpg
…	…	… …
4445	67975	W_71920_60_mods_M.jpg
4446	67975	T_17798_19_normcore_M.jpg
4447	67975	T_17797_19_normcore_M.jpg
4448	67975	W_17754_80_bold_M.jpg
4449	67975	W_71933_60_mods_M.jpg

[4450 rows x 3 columns]

[51]:
```
top100_val_df
```

[51]:

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

```
[1100 rows x 3 columns]
```

1.2.3 3. 100 “ ”

pivot table

[52]:
```
# dataset    state
top100_train_df['dataset'] = 'Training'
top100_val_df['dataset'] = 'Validation'

#
combined_df = pd.concat([top100_train_df, top100_val_df], ignore_index=True)

# '   ID'  'dataset'        ,       ("\n")
grouped = combined_df.groupby(['   ID', 'dataset', '      ']).agg({
    '  ': lambda x: '\n'.join(x)
}).reset_index()

# '              'like'
grouped.sort_values(by=['   ID', 'dataset', '      '], ascending=[True, True,␣
 ↪False], inplace=True)
```

pivot table , HTML

[53]:
```
#
final_result = grouped.pivot(index='   ID', columns=['dataset', '      '],␣
 ↪values='  ')

# HTML
from IPython.display import HTML

def display_df(df, rows=5):
    styles = [
        dict(selector="th", props=[("text-align", "left")]),
        dict(selector="td", props=[("white-space", "pre-wrap")])
    ]
    return HTML(df.head(rows).style.set_table_styles(styles).to_html())

# 100
display(display_df(final_result, rows=100))
```

```
<IPython.core.display.HTML object>
```

[54]:
```
final_result.to_csv('Mission2/final_result.csv')
final_result
```

[54]:
```
dataset                                                 Training  \

   ID
368         W_04604_00_metrosexual_M.jpg\nW_16403_10_sport…
837         W_00829_10_sportivecasual_M.jpg\nW_09157_60_mo…
7658        W_08410_00_cityglam_W.jpg\nW_18560_70_military…
7905        W_02845_60_mods_M.jpg\nW_24765_60_mods_M.jpg\n…
9096        W_06437_90_grunge_W.jpg\nW_19075_50_classic_W…
…                                                         …
66469       W_59268_70_hippie_M.jpg\nT_02558_19_normcore_M…
66513       T_07416_19_lounge_W.jpg\nW_14828_50_classic_W…
66592       T_09717_19_genderless_W.jpg\nW_02343_60_space_…
66731       W_04137_60_minimal_W.jpg\nT_14085_19_genderles…
67975       W_52583_50_ivy_M.jpg\nW_07095_00_metrosexual_M…

dataset                                                       \

   ID
368         W_16264_80_bold_M.jpg\nW_15340_50_ivy_M.jpg\nW…
837         W_27782_90_hiphop_M.jpg\nW_24381_70_hippie_M.j…
7658        W_10510_60_space_W.jpg\nW_00682_70_punk_W.jpg\…
7905        W_10076_50_ivy_M.jpg\nW_15545_70_hippie_M.jpg\…
9096        W_08232_19_normcore_W.jpg\nW_14393_70_hippie_W…
…                                                         …
66469       T_06076_60_mods_M.jpg\nT_07605_00_metrosexual_…
66513       W_67337_90_grunge_W.jpg\nW_10984_50_feminine_W…
66592       W_52969_00_ecology_W.jpg\nW_05140_50_feminine_…
66731       W_67040_00_oriental_W.jpg\nW_47122_80_powersui…
67975       T_21986_70_hippie_M.jpg\nW_71922_60_mods_M.jpg…

dataset                                               Validation  \

   ID
368         W_06864_10_sportivecasual_M.jpg\nW_04678_50_iv…
837         W_06590_90_hiphop_M.jpg\nW_00829_10_sportiveca…
7658        W_04927_50_feminine_W.jpg\nW_09731_19_genderle…
7905             W_02845_60_mods_M.jpg\nW_32034_80_bold_M.jpg
9096        W_18714_90_kitsch_W.jpg\nW_19205_00_oriental_W…
…                                                         …
66469       W_52231_50_ivy_M.jpg\nT_01123_90_hiphop_M.jpg\…
66513                                 W_14828_50_classic_W.jpg
66592       T_00253_60_popart_W.jpg\nW_46907_80_powersuit_…
66731                                                   NaN
67975       W_07074_00_metrosexual_M.jpg\nW_17738_80_bold_…

dataset
```

```
   ID
368         W_16034_80_bold_M.jpg\nW_15340_50_ivy_M.jpg\nW…
837         W_27700_70_hippie_M.jpg\nW_15661_70_hippie_M.j…
7658        W_05312_80_bodyconscious_W.jpg\nW_13688_90_hip…
7905        W_28909_19_normcore_M.jpg\nW_24535_70_hippie_M…
9096        W_00191_10_sportivecasual_W.jpg\nW_14225_50_fe…
…                                                       …
66469       W_24647_70_hippie_M.jpg\nW_58887_00_metrosexua…
66513       W_60553_00_cityglam_W.jpg\nW_39793_80_powersui…
66592       W_35400_80_powersuit_W.jpg\nW_22056_70_hippie_…
66731       W_64332_80_powersuit_W.jpg\nW_22783_70_hippie_…
67975       T_21992_70_hippie_M.jpg\nW_52521_50_ivy_M.jpg\…

[100 rows x 4 columns]
```