# 0_prepare_datatable

February 7, 2020

## 1 Prepare datatable used to benchmark the Balance Faces in the Wild (BFW) dataset.

Load table in `data/bfw-datatable.pkl` to extract all features and store in the datatable. Overwrites the table to `data/bfw-datatable.pkl`.

### 1.1 Add project code to PYTHONPATH, if not already there

Check that *path_package* is set to *code* directory on respective system

[20]:

```
import pandas as pd
import pathlib
from sklearn.preprocessing import LabelEncoder
%matplotlib inline
%reload_ext autoreload
%autoreload 2
version_bfw="0.1.5"
dir_meta = f'../../data/bfw/meta/'
```

### 1.2 Load list of pairs

Load lists of pairs, with *p1* and *p2* representing samples for either pair, *label* is set as 1 if *genuine*; else, 0 for *imposter*, and the fold. Note there is no overlap in subjects between folds. Finally, the experiments are 5-fold (hence, *fold* $\in \{1, 2, 3, 4, 5\}$

[22]:

```
fin = f'{dir_meta}bfw-v{version_bfw}-pairs.csv'
data = pd.read_csv(fin)
data.head()
```

[22]:

```
   fold                                 p1                                 p2  \
0     1  asian_females/n000009/0010_01.jpg  asian_females/n000009/0043_01.jpg
1     1  asian_females/n000009/0010_01.jpg  asian_females/n000009/0120_01.jpg
2     1  asian_females/n000009/0010_01.jpg  asian_females/n000009/0122_02.jpg
3     1  asian_females/n000009/0010_01.jpg  asian_females/n000009/0188_01.jpg
4     1  asian_females/n000009/0010_01.jpg  asian_females/n000009/0205_01.jpg

   label
0      1
```

```
1        1
2        1
3        1
4        1
```

## 1.3 Display stats

For each fold, how many *genuine* (i.e., 1) and *imposter* (i.e., 0)

[23]:

```
dfcounts = pd.DataFrame(data.groupby(by=['fold', 'label']).count()[['p1']])
dfcounts.columns = ['stats']
dfcounts.head(len(dfcounts))
```

[23]:

```
                 stats
fold label
1    0         136194
     1          48514
2    0         136185
     1          48469
3    0         136284
     1          48527
4    0         136242
     1          48512
5    0         136474
     1          48497
```

[24]:

```
del dfcounts
```

## 1.4 Add metadata to table, set format appropriately each step

There is rich information in the pairs list, for which preparing the datatable will be convenient later

Set attributes, IDs (str), and abbreviated variants of attribute, gender, and ethnicity for both *p1* and *p2*

[25]:

```
data['att1'] = data.p1.apply(lambda x: x.split('/')[0]).astype('category')
data['att2'] = data.p2.apply(lambda x: x.split('/')[0]).astype('category')



data['e1'] = data.att1.apply(lambda x: x.split('_')[0][0].upper())
data['e2'] = data.att2.apply(lambda x: x.split('_')[0][0].upper())

data['g1'] = data.att1.apply(lambda x: x.split('_')[1][0].upper())
data['g2'] = data.att2.apply(lambda x: x.split('_')[1][0].upper())

data['a1'] = (data['e1'] + data['g1']).astype('category')
data['a2'] = (data['e2'] + data['g2']).astype('category')
```

```
data['e1'] = data['e1'].astype('category')
data['e2'] = data['e2'].astype('category')
data['g1'] = data['g1'].astype('category')
data['g2'] = data['g2'].astype('category')
data['score'] = pd.np.nan
data.head()
```

```
[25]:    fold                                p1                                p2  \
      0     1  asian_females/n000009/0010_01.jpg  asian_females/n000009/0043_01.jpg
      1     1  asian_females/n000009/0010_01.jpg  asian_females/n000009/0120_01.jpg
      2     1  asian_females/n000009/0010_01.jpg  asian_females/n000009/0122_02.jpg
      3     1  asian_females/n000009/0010_01.jpg  asian_females/n000009/0188_01.jpg
      4     1  asian_females/n000009/0010_01.jpg  asian_females/n000009/0205_01.jpg

         label           att1           att2 e1 e2 g1 g2  a1  a2  score
      0      1  asian_females  asian_females  A  A  F  F  AF  AF    NaN
      1      1  asian_females  asian_females  A  A  F  F  AF  AF    NaN
      2      1  asian_females  asian_females  A  A  F  F  AF  AF    NaN
      3      1  asian_females  asian_females  A  A  F  F  AF  AF    NaN
      4      1  asian_females  asian_females  A  A  F  F  AF  AF    NaN
```

### 1.4.1 assign unique ID tags per subject

Encode N subjects as 0, 1, ...., N - 1

```
[26]: le = LabelEncoder()



      subject_names = list(set(["/".join(p1.split('/')[:-1]) for p1 in data['p1'].
       ↪unique()] + ["/".join(p2.split('/')[:-1]) for p2 in data['p2'].unique()]))
      le.fit(subject_names)

      data['ids1'] = le.transform(data['p1'].apply(lambda x: "/".join(x.split('/')[:
       ↪-1])))
      data['ids2'] = le.transform(data['p2'].apply(lambda x: "/".join(x.split('/')[:
       ↪-1])))
      data.head()
```

```
[26]:    fold                                p1                                p2  \
      0     1  asian_females/n000009/0010_01.jpg  asian_females/n000009/0043_01.jpg
      1     1  asian_females/n000009/0010_01.jpg  asian_females/n000009/0120_01.jpg
      2     1  asian_females/n000009/0010_01.jpg  asian_females/n000009/0122_02.jpg
      3     1  asian_females/n000009/0010_01.jpg  asian_females/n000009/0188_01.jpg
      4     1  asian_females/n000009/0010_01.jpg  asian_females/n000009/0205_01.jpg

         label           att1           att2 e1 e2 g1 g2  a1  a2  score  ids1  ids2
      0      1  asian_females  asian_females  A  A  F  F  AF  AF    NaN     0     0
```

```
1      1  asian_females  asian_females  A  A  F  F  AF  AF    NaN     0     0
2      1  asian_females  asian_females  A  A  F  F  AF  AF    NaN     0     0
3      1  asian_females  asian_females  A  A  F  F  AF  AF    NaN     0     0
4      1  asian_females  asian_females  A  A  F  F  AF  AF    NaN     0     0
```

[27]:

```
data.sample(10)
```

[27]:

```
        fold                                  p1  \
891525     5     white_males/n003055/0035_01.jpg
731787     2   indian_females/n004671/0070_01.jpg
238634     1   indian_females/n008665/0203_01.jpg
542338     4     asian_males/n003489/0271_01.jpg
316318     5    indian_males/n006214/0235_01.jpg
644695     2     black_males/n002141/0435_01.jpg
102977     3     asian_males/n006614/0135_02.jpg
455642     4     white_males/n001111/0389_01.jpg
250475     3   indian_females/n008310/0006_01.jpg
227505     5     black_males/n002012/0163_01.jpg

                                        p2  label            att1  \
891525     black_males/n003539/0051_01.jpg      0     white_males
731787     black_males/n002398/0120_01.jpg      0  indian_females
238634  indian_females/n008665/0124_01.jpg      1  indian_females
542338     asian_males/n001496/0171_01.jpg      0     asian_males
316318    indian_males/n006214/0110_01.jpg      1    indian_males
644695     black_males/n009210/0080_01.jpg      0     black_males
102977     asian_males/n008150/0048_01.jpg      0     asian_males
455642     white_males/n003485/0397_01.jpg      0     white_males
250475  indian_females/n008310/0223_01.jpg      1  indian_females
227505     black_males/n006257/0036_03.jpg      0     black_males

                  att2 e1 e2 g1 g2  a1  a2  score  ids1  ids2
891525     black_males  W  B  M  M  WM  BM    NaN   735   347
731787     black_males  I  B  F  M  IF  BM    NaN   430   330
238634  indian_females  I  I  F  F  IF  IF    NaN   498   498
542338     asian_males  A  A  M  M  AM  AM    NaN   127   106
316318    indian_males  I  I  M  M  IM  IM    NaN   535   535
644695     black_males  B  B  M  M  BM  BM    NaN   320   399
102977     asian_males  A  A  M  M  AM  AM    NaN   163   175
455642     white_males  W  W  M  M  WM  WM    NaN   714   746
250475  indian_females  I  I  F  F  IF  IF    NaN   494   494
227505     black_males  B  B  M  M  BM  BM    NaN   315   367
```

### 1.5 Save datatable

if file does not exist, write to disc

```
[28]: fout = f'{dir_meta}/bfw-{version_bfw}-datatable.pkl'
      if not pathlib.Path().is_file():
          data.to_pickle(fout)
```