In [1]:
```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import cv2
from PIL import Image
%matplotlib inline
```

In [2]:
```
import  pickle
```

In [4]:
```
# Loading image

df = pickle.load(open('./dataa/dataframe_images_100_100.pickle','rb'))
```

In [5]:
```
df.head()
```

Out[5]:

	gender	0	1	2	3	4	5	6	7	8	...	9990	9991	9992	9993	9994	9995	9996	9997	9998	9999
0	female	188	180	184	188	173	179	192	178	216	...	109	111	115	116	122	116	119	120	117	111
1	female	32	24	32	27	29	29	29	32	36	...	58	47	27	33	22	28	22	36	62	17
2	female	22	30	39	36	30	61	11	17	10	...	156	171	177	186	176	185	186	190	177	177
3	female	35	35	35	35	35	35	35	35	35	...	75	82	90	92	86	70	89	84	84	74
4	female	86	86	71	54	45	49	33	20	18	...	35	34	32	32	30	32	34	34	33	30

5 rows × 10001 columns

In [6]:
```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 5458 entries, 0 to 6057
Columns: 10001 entries, gender to 9999
dtypes: object(1), uint8(10000)
memory usage: 52.1+ MB
```

In [9]:
```
# checking missing values

df.isnull().sum()
```

Out[9]:
```
gender    0
0         0
1         0
2         0
3         0
         ..
9995      0
9996      0
9997      0
9998      0
9999      0
Length: 10001, dtype: int64
```

Data Preprocessing

-remove missing -Data normalization(Min Max Scaling)

In [10]:
```
# removing missing values

df.dropna(axis=0,inplace=True)
```

In [11]:
```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 5458 entries, 0 to 6057
Columns: 10001 entries, gender to 9999
dtypes: object(1), uint8(10000)
memory usage: 52.1+ MB
```

In [12]:
```
# split into two parts

X = df.iloc[:,1:].values #independent features
Y = df.iloc[:,0].values #dependent features
```

In [13]:
```
X.shape
```

Out[13]:
```
(5458, 10000)
```

Min Max Scaling

$$\text{Xnorm} = \frac{x - minValue}{maxValue - minValue}$$

In [14]:
```
X.min(), X.max()
```

Out[14]:
```
(0, 255)
```

In [15]:
```
Xnorm = X / 255
```

In [16]:
```
Xnorm
```

Out[16]:
```
array([[0.7372549 , 0.70588235, 0.72156863, ..., 0.47058824, 0.45882353,
        0.43529412],
       [0.1254902 , 0.09411765, 0.1254902 , ..., 0.14117647, 0.24313725,
        0.06666667],
       [0.08627451, 0.11764706, 0.15294118, ..., 0.74509804, 0.69411765,
        0.69411765],
       ...,
       [0.11764706, 0.11764706, 0.11764706, ..., 0.4627451 , 0.5372549 ,
        0.52941176],
       [0.08235294, 0.10588235, 0.12156863, ..., 0.07843137, 0.08627451,
        0.09803922],
       [0.01568627, 0.01176471, 0.00784314, ..., 0.35294118, 0.35294118,
        0.36470588]])
```

In [17]:
```
Xnorm.shape
```

Out[17]:
```
(5458, 10000)
```

In [19]:
```
Y
```

Out[19]:
```
array(['female', 'female', 'female', ..., 'male', 'male', 'male'],
      dtype=object)
```

In [23]:
```
# female=1, male=0
y_norm = np.where(Y=='female',1,0)
```

In [26]:
```
# save X and y
np.savez('./dataa/data_10000_norm.npz',Xnorm,y_norm)
```

In [ ]:
```

```