

Emoji Recognition using RandomForest

IMPORT NECESSARY LIBRARY

```
In [1]: import os  
import warnings  
warnings.simplefilter('ignore')
```

```
In [2]: import pandas as pd  
import numpy as np
```

```
In [3]: import matplotlib.pyplot as plt  
%matplotlib inline
```

```
In [4]: from skimage.io import imshow, imread  
from skimage.transform import resize  
from skimage.color import rgb2gray
```

IMPORT IMAGE FILE VIA DIRECTORY

```
In [5]: happy=os.listdir("emoji_img/happy")
```

```
In [6]: sad=os.listdir("emoji_img/sad")
```

```
In [7]: angry=os.listdir("emoji_img/angry")
```

```
In [8]: limit=10  
happy_images=[None]*limit  
j=0  
for i in happy:  
    if (j<limit):  
        happy_images[j]=imread("emoji_img/happy/"+i)  
    j+=1
```

```
    else:  
        break
```

```
In [9]: limit=10  
sad_images=[None]*limit  
j=0  
for i in sad:  
    if (j<limit):  
        sad_images[j]=imread("emoji_img/sad/"+i)  
        j+=1  
    else:  
        break
```

```
In [10]: limit=10  
angry_images=[None]*limit  
j=0  
for i in angry:  
    if (j<limit):  
        angry_images[j]=imread("emoji_img/angry/"+i)  
        j+=1  
    else:  
        break
```

```
In [11]: limit=10  
happy_gray =[None]*limit  
j = 0  
for i in happy:  
    if (j < limit):  
        happy_gray[j]=rgb2gray(happy_images[j])  
        j+=1  
    else:  
        break
```

```
In [12]: limit=10  
sad_gray =[None]*limit  
j = 0  
for i in sad:  
    if (j < limit):  
        sad_gray[j]=rgb2gray(sad_images[j])
```

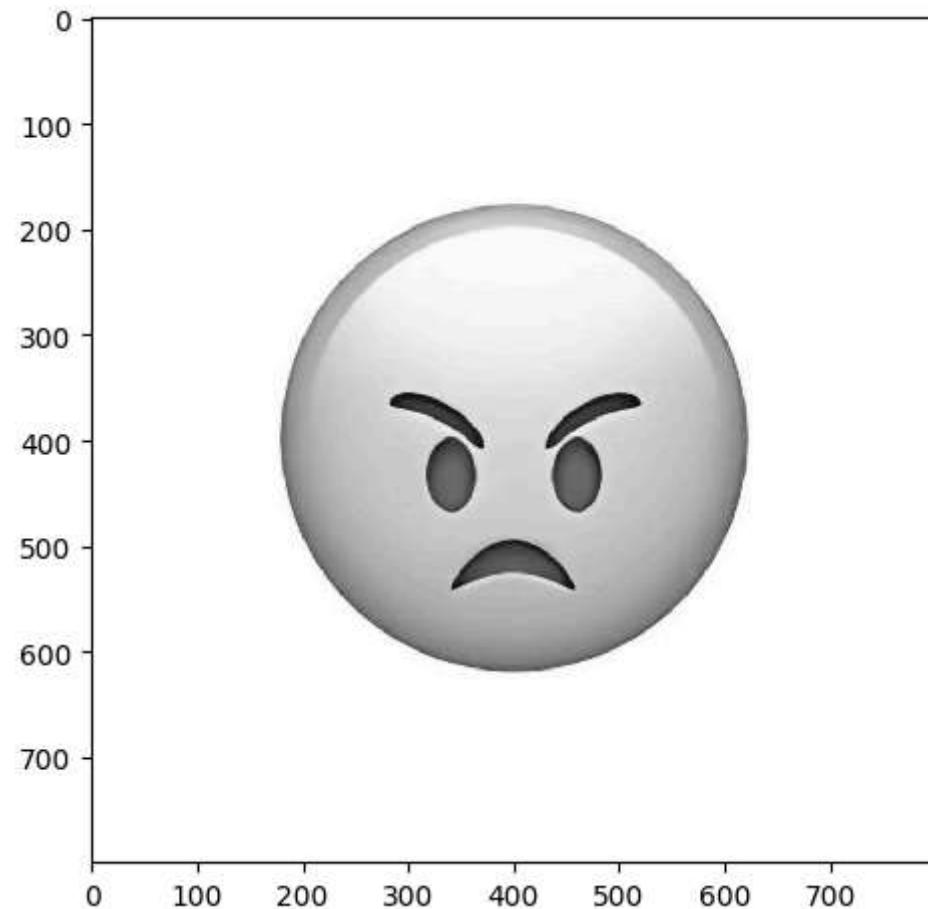
```
j+=1
else:
    break
```

```
In [13]: limit=10
angry_gray =[None]*limit
j = 0
for i in angry:
    if (j < limit):
        angry_gray[j]=rgb2gray(angry_images[j])
        j+=1
    else:
        break
```

```
In [14]: happy_gray[0]
```

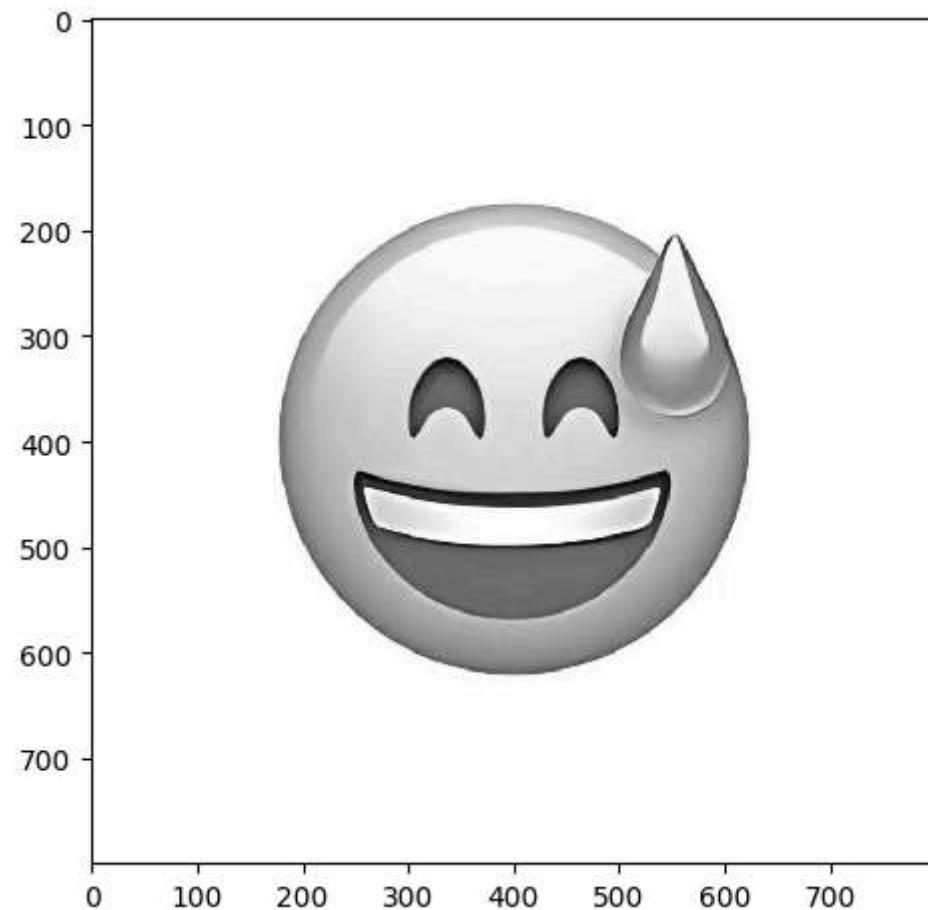
```
Out[14]: array([[1., 1., 1., ..., 1., 1., 1.],
                 [1., 1., 1., ..., 1., 1., 1.],
                 [1., 1., 1., ..., 1., 1., 1.],
                 ...,
                 [1., 1., 1., ..., 1., 1., 1.],
                 [1., 1., 1., ..., 1., 1., 1.],
                 [1., 1., 1., ..., 1., 1., 1.]])
```

```
In [15]: view=imshow(angry_gray[0])
plt.show()
```

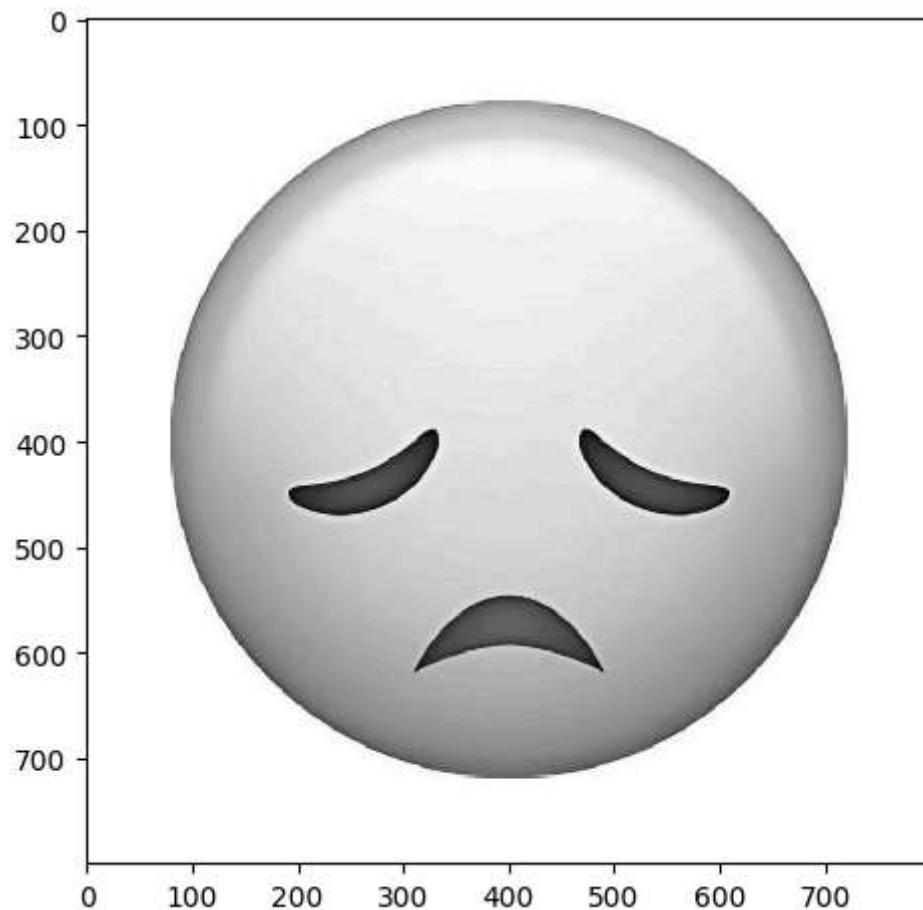


VIEW IMAGE

```
In [16]: view=imshow(happy_gray[0])
plt.show()
```



```
In [17]: view=imshow(sad_gray[0])
plt.show()
```



Check the image matrix size before resizing

```
In [18]: angry_gray[5].shape
```

```
Out[18]: (800, 800)
```

```
In [19]: happy_gray[2].shape
```

```
Out[19]: (800, 800)
```

```
In [20]: sad_gray[2].shape
```

```
Out[20]: (800, 800)
```

```
In [21]: angry_gray[2].shape
```

```
Out[21]: (800, 800)
```

RESIZE IMAGE INTO 512x512

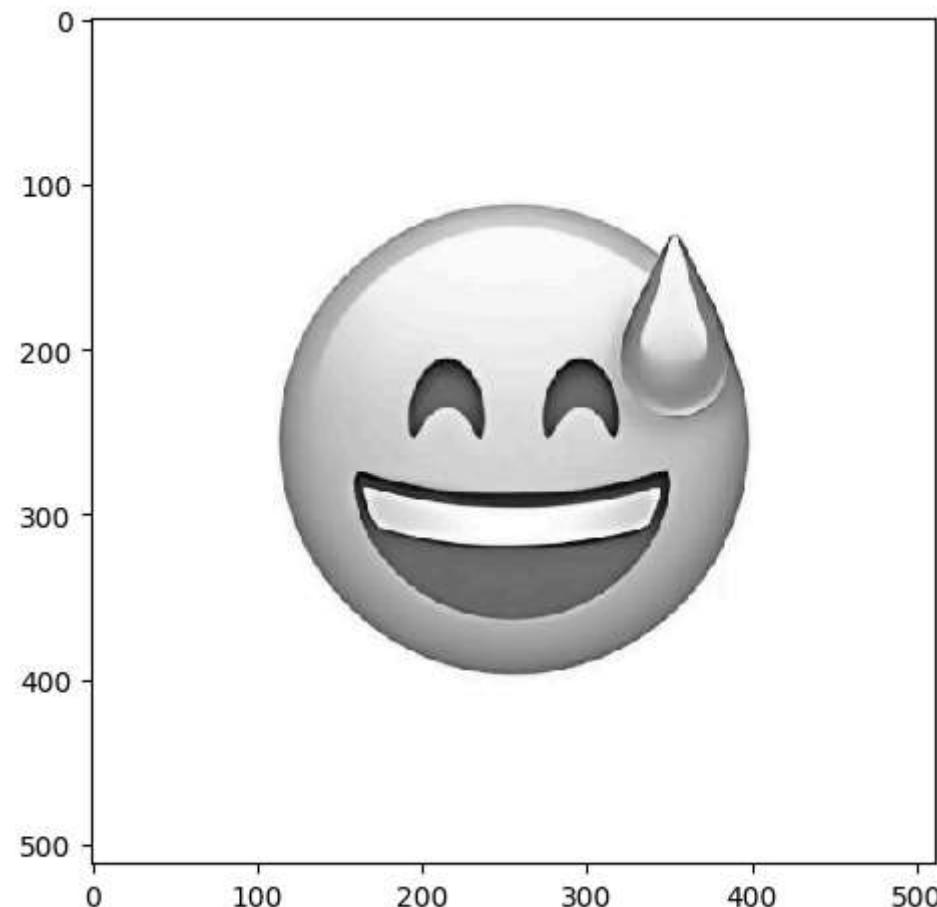
```
In [37]: for j in range(10):
    happy=happy_gray[j]
    happy_gray[j]=resize(happy,(512,512))
```

```
In [38]: for j in range(10):
    sad=sad_gray[j]
    sad_gray[j]=resize(sad,(512,512))
```

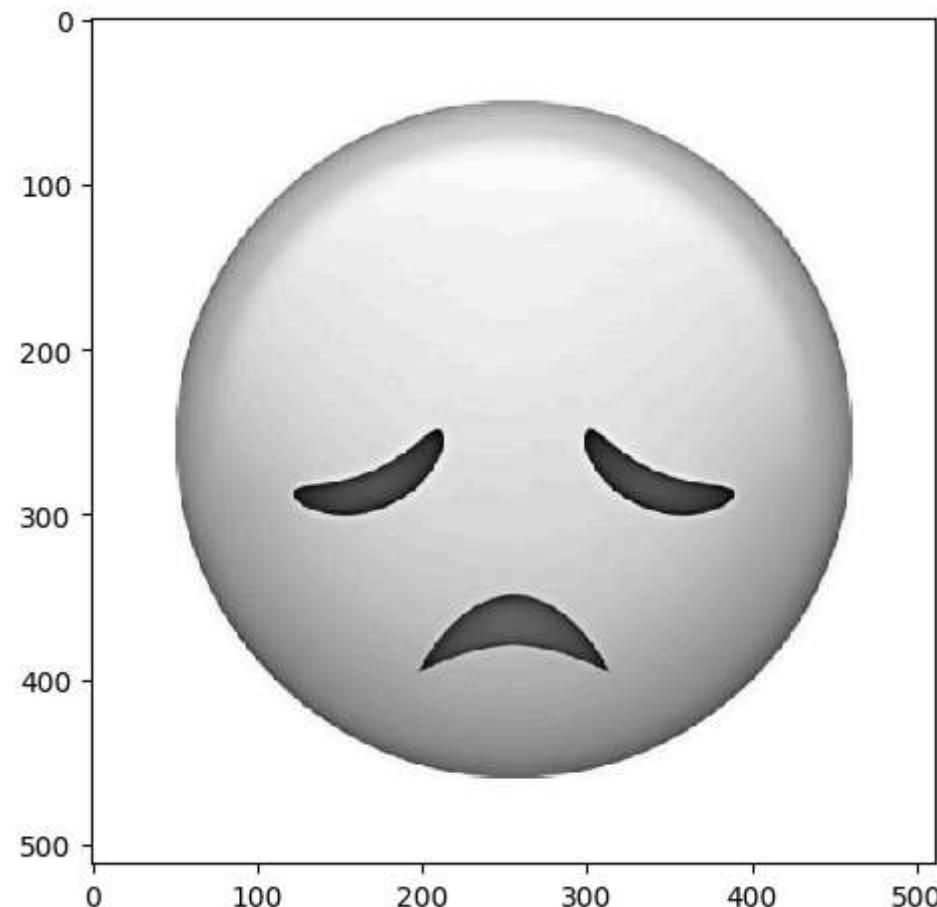
```
In [43]: for j in range(10):
    angry=angry_gray[j]
    angry_gray[j]=resize(ag,(512,512))
```

VIEW RESIZED IMAGE

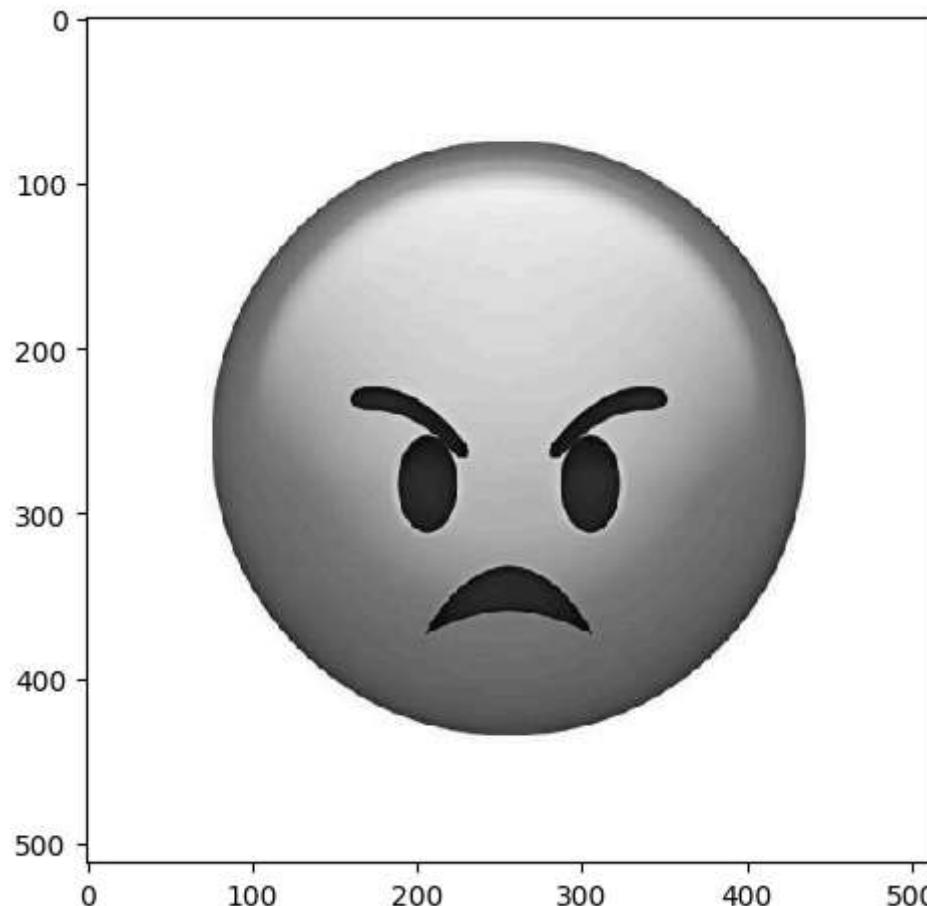
```
In [44]: view=imshow(happy_gray[0])
plt.show()
```



```
In [45]: view=imshow(sad_gray[0])
plt.show()
```



```
In [46]: view=imshow(angry_gray[0])
plt.show()
```



FIND NUMBER OF GRayscale IMAGE

```
In [47]: length_happy=len(happy_gray)  
length_happy
```

```
Out[47]: 10
```

```
In [48]: length_sad=len(sad_gray)  
length_sad
```

Out[48]: 10

```
In [49]: length_angry=len(angry_gray)  
length_angry
```

Out[49]: 10

CREATE VARIABLE FLATTEN IMAGE FOR THREE INPUT EMOJI

```
In [50]: happy_size=happy_gray[0].shape  
happy_size
```

Out[50]: (512, 512)

```
In [51]: sad_size=sad_gray[0].shape  
sad_size
```

Out[51]: (512, 512)

```
In [52]: angry_size=angry_gray[0].shape  
angry_size
```

Out[52]: (512, 512)

FLATTEN IMAGE FOR THREE INPUT EMOJI

```
In [53]: flatten_happy=happy_size[0]*happy_size[1]  
flatten_happy
```

Out[53]: 262144

```
In [54]: flatten_sad=sad_size[0]*sad_size[1]  
flatten_sad
```

Out[54]: 262144

```
In [55]: flatten_angry=angry_size[0]*angry_size[1]
flatten_angry
```

```
Out[55]: 262144
```

NOW FALTEN IMAGE INTO VECTOR

```
In [56]: for i in range(length_happy):
    happy_gray[i]=np.ndarray.flatten(happy_gray[i]).reshape(flatten_happy,1)
```

```
In [57]: for i in range(length_sad):
    sad_gray[i]=np.ndarray.flatten(sad_gray[i]).reshape(flatten_sad,1)
```

```
In [58]: for i in range(length_angry):
    angry_gray[i]=np.ndarray.flatten(angry_gray[i]).reshape(flatten_angry,1)
```

STACK THE BOTH INPUT EMOJI AS ARRAY

```
In [59]: happy_gray=np.dstack(happy_gray)
happy_gray
```

```
Out[59]: array([[[1., 1., 1., ..., 1., 1., 1.]],
               [[1., 1., 1., ..., 1., 1., 1.]],
               [[1., 1., 1., ..., 1., 1., 1.]],
               ...,
               [[1., 1., 1., ..., 1., 1., 1.]],
               [[1., 1., 1., ..., 1., 1., 1.]],
               [[1., 1., 1., ..., 1., 1., 1.]]])
```

```
In [60]: sad_gray=np.dstack(sad_gray)
sad_gray
```

```
Out[60]: array([[[1., 1., 1., ..., 1., 1., 1.]],  
                [[1., 1., 1., ..., 1., 1., 1.]],  
                [[1., 1., 1., ..., 1., 1., 1.]],  
                ...,  
                [[1., 1., 1., ..., 1., 1., 1.]],  
                [[1., 1., 1., ..., 1., 1., 1.]],  
                [[1., 1., 1., ..., 1., 1., 1.]]])
```

```
In [61]: angry_gray=np.dstack(angry_gray)  
angry_gray
```

```
Out[61]: array([[[1., 1., 1., ..., 1., 1., 1.]],  
                [[1., 1., 1., ..., 1., 1., 1.]],  
                [[1., 1., 1., ..., 1., 1., 1.]],  
                ...,  
                [[1., 1., 1., ..., 1., 1., 1.]],  
                [[1., 1., 1., ..., 1., 1., 1.]],  
                [[1., 1., 1., ..., 1., 1., 1.]]])
```

ROLLAXIS

```
In [62]: happy_gray=np.rollaxis(happy_gray, axis=2, start=0)  
happy_gray
```

```
Out[62]: array([[[1.],
   [1.],
   [1.],
   ...,
   [1.],
   [1.],
   [1.]],

   [[1.],
   [1.],
   [1.],
   ...,
   [1.],
   [1.],
   [1.]],

   [[1.],
   [1.],
   [1.],
   ...,
   [1.],
   [1.],
   [1.]],

   ...,

   [[1.],
   [1.],
   [1.],
   ...,
   [1.],
   [1.],
   [1.]],

   [[1.],
   [1.],
   [1.],
   ...,
   [1.],
   [1.],
   [1.]]])
```

```
[1.]],  
[[1.],  
[1.],  
[1.],  
...,  
[1.],  
[1.],  
[1.]]])
```

```
In [63]: sad_gray=np.rollaxis(sad_gray, axis=2, start=0)  
sad_gray
```

```
Out[63]: array([[[1.],
   [1.],
   [1.],
   ...,
   [1.],
   [1.],
   [1.]],

   [[1.],
   [1.],
   [1.],
   ...,
   [1.],
   [1.],
   [1.]],

   [[1.],
   [1.],
   [1.],
   ...,
   [1.],
   [1.],
   [1.]],

   ...,

   [[1.],
   [1.],
   [1.],
   ...,
   [1.],
   [1.],
   [1.]],

   [[1.],
   [1.],
   [1.],
   ...,
   [1.],
   [1.],
   [1.]]])
```

```
[1.]],  
[[1.],  
[1.],  
[1.],  
...,  
[1.],  
[1.],  
[1.]]])
```

```
In [64]: angry_gray=np.rollaxis(angry_gray, axis=2, start=0)  
angry_gray
```

```
Out[64]: array([[[1.],
   [1.],
   [1.],
   ...,
   [1.],
   [1.],
   [1.]],

   [[1.],
   [1.],
   [1.],
   ...,
   [1.],
   [1.],
   [1.]],

   [[1.],
   [1.],
   [1.],
   ...,
   [1.],
   [1.],
   [1.]],

   ...,

   [[1.],
   [1.],
   [1.],
   ...,
   [1.],
   [1.],
   [1.]],

   [[1.],
   [1.],
   [1.],
   ...,
   [1.],
   [1.],
   [1.]]])
```

```
[1.]],  
[[1.],  
[1.],  
[1.],  
...,  
[1.],  
[1.],  
[1.]])
```

In [65]: `happy_gray.shape`

Out[65]: (10, 262144, 1)

In [66]: `sad_gray.shape`

Out[66]: (10, 262144, 1)

In [67]: `angry_gray.shape`

Out[67]: (10, 262144, 1)

In [68]: `happy_gray=happy_gray.reshape(length_happy,flatten_happy)`
`happy_gray`

Out[68]: array([[1., 1., 1., ..., 1., 1., 1.],
[1., 1., 1., ..., 1., 1., 1.],
[1., 1., 1., ..., 1., 1., 1.],
...,
[1., 1., 1., ..., 1., 1., 1.],
[1., 1., 1., ..., 1., 1., 1.],
[1., 1., 1., ..., 1., 1., 1.]])

In [69]: `sad_gray=sad_gray.reshape(length_happy,flatten_happy)`

In [71]: `angry_gray=angry_gray.reshape(length_angry,flatten_angry)`

In [72]: `happy_data=pd.DataFrame(happy_gray)`

In [73]: happy_data

| | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | ... | 262134 | 262135 | 262136 | 262137 | 262138 | 262139 | 262140 | 262141 | 262142 | 26214 |
|---|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|--------|--------|--------|--------|--------|--------|--------|--------|--------|-------|
| 0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | ... | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1 |
| 1 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | ... | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1 |
| 2 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | ... | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1 |
| 3 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | ... | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1 |
| 4 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | ... | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1 |
| 5 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | ... | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1 |
| 6 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | ... | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1 |
| 7 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | ... | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1 |
| 8 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | ... | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1 |
| 9 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | ... | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1 |

10 rows × 262144 columns



In [74]: sad_data=pd.DataFrame(sad_gray)

In [75]: sad_data

Out[75]:

| | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | ... | 262134 | 262135 | 262136 | 262137 | 262138 | 262139 | 262140 | 262141 | 262142 | 26214 |
|---|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|--------|--------|--------|--------|--------|--------|--------|--------|--------|-------|
| 0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | ... | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1 |
| 1 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | ... | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1 |
| 2 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | ... | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1 |
| 3 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | ... | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1 |
| 4 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | ... | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1 |
| 5 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | ... | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1 |
| 6 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | ... | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1 |
| 7 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | ... | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1 |
| 8 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | ... | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1 |
| 9 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | ... | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1 |

10 rows × 262144 columns



In [76]: angry_data=pd.DataFrame(angry_gray)

In [78]: angry_data

Out[78]:

| | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | ... | 262134 | 262135 | 262136 | 262137 | 262138 | 262139 | 262140 | 262141 | 262142 | 26214 |
|---|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|--------|--------|--------|--------|--------|--------|--------|--------|--------|-------|
| 0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | ... | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1 |
| 1 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | ... | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1 |
| 2 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | ... | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1 |
| 3 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | ... | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1 |
| 4 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | ... | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1 |
| 5 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | ... | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1 |
| 6 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | ... | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1 |
| 7 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | ... | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1 |
| 8 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | ... | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1 |
| 9 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | ... | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1 |

10 rows × 262144 columns



In [79]: happy_data['label']='happy'

In [80]: sad_data['label']='sad'

In [81]: angry_data['label']='angry'

In [82]: happy_data

Out[82]:

| | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | ... | 262135 | 262136 | 262137 | 262138 | 262139 | 262140 | 262141 | 262142 | 262143 | label |
|---|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|--------|--------|--------|--------|--------|--------|--------|--------|--------|-------|
| 0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | ... | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | happy |
| 1 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | ... | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | happy |
| 2 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | ... | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | happy |
| 3 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | ... | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | happy |
| 4 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | ... | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | happy |
| 5 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | ... | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | happy |
| 6 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | ... | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | happy |
| 7 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | ... | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | happy |
| 8 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | ... | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | happy |
| 9 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | ... | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | happy |

10 rows × 262145 columns



In [83]: sad_data

Out[83]:

| | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | ... | 262135 | 262136 | 262137 | 262138 | 262139 | 262140 | 262141 | 262142 | 262143 | label |
|---|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|--------|--------|--------|--------|--------|--------|--------|--------|--------|-------|
| 0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | ... | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | sad |
| 1 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | ... | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | sad |
| 2 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | ... | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | sad |
| 3 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | ... | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | sad |
| 4 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | ... | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | sad |
| 5 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | ... | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | sad |
| 6 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | ... | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | sad |
| 7 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | ... | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | sad |
| 8 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | ... | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | sad |
| 9 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | ... | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | sad |

10 rows × 262145 columns



In [293...]

emoji=pd.concat([happy_data,sad_data,angry_data])

In [294...]

emoji

Out[294...]

| | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | ... | 262135 | 262136 | 262137 | 262138 | 262139 | 262140 | 262141 | 262142 | 262143 | label |
|---|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|--------|--------|--------|--------|--------|--------|--------|--------|--------|-------|
| 0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | ... | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | happy |
| 1 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | ... | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | happy |
| 2 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | ... | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | happy |
| 3 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | ... | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | happy |
| 4 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | ... | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | happy |
| 5 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | ... | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | happy |
| 6 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | ... | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | happy |
| 7 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | ... | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | happy |
| 8 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | ... | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | happy |
| 9 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | ... | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | happy |
| 0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | ... | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | sad |
| 1 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | ... | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | sad |
| 2 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | ... | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | sad |
| 3 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | ... | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | sad |
| 4 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | ... | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | sad |
| 5 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | ... | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | sad |
| 6 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | ... | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | sad |
| 7 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | ... | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | sad |
| 8 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | ... | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | sad |
| 9 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | ... | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | sad |
| 0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | ... | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | angry |
| 1 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | ... | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | angry |

| 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | ... | 262135 | 262136 | 262137 | 262138 | 262139 | 262140 | 262141 | 262142 | 262143 | label |
|---|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|--------|--------|--------|--------|--------|--------|--------|--------|--------|-------|
| 2 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | ... | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | angry |
| 3 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | ... | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | angry |
| 4 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | ... | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | angry |
| 5 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | ... | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | angry |
| 6 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | ... | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | angry |
| 7 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | ... | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | angry |
| 8 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | ... | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | angry |
| 9 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | ... | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | angry |

30 rows × 262145 columns

NOW USE SHUFFLE LIBRARY TO SHUFFLE

In [295...]: `from sklearn.utils import shuffle`

In [374...]: `emoji_index=shuffle(emoji).reset_index()`

In [375...]: `emoji_index`

Emoji_Recognition_RandomForest

Out[375...]

| | index | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | ... | 262135 | 262136 | 262137 | 262138 | 262139 | 262140 | 262141 | 262142 | 262143 | I |
|----|-------|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|--------|--------|--------|--------|--------|--------|--------|--------|--------|-----|
| 0 | 8 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | ... | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | a |
| 1 | 4 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | ... | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 |
| 2 | 6 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | ... | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | ha |
| 3 | 9 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | ... | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | a |
| 4 | 0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | ... | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | a |
| 5 | 1 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | ... | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 |
| 6 | 5 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | ... | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | a |
| 7 | 6 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | ... | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 |
| 8 | 5 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | ... | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 |
| 9 | 3 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | ... | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | a |
| 10 | 4 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | ... | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | a |
| 11 | 9 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | ... | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | ha |
| 12 | 8 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | ... | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | ha |
| 13 | 8 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | ... | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 |
| 14 | 2 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | ... | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | a |
| 15 | 4 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | ... | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | ha |
| 16 | 2 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | ... | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 |
| 17 | 3 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | ... | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | ha |
| 18 | 2 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | ... | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | ha |
| 19 | 6 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | ... | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | a |
| 20 | 5 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | ... | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | ha |
| 21 | 1 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | ... | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | a |

| | index | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | ... | 262135 | 262136 | 262137 | 262138 | 262139 | 262140 | 262141 | 262142 | 262143 | I |
|----|-------|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|--------|--------|--------|--------|--------|--------|--------|--------|--------|----|
| 22 | 0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | ... | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | ha |
| 23 | 0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | ... | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | ha |
| 24 | 7 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | ... | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | a |
| 25 | 3 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | ... | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | ha |
| 26 | 7 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | ... | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | ha |
| 27 | 9 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | ... | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | ha |
| 28 | 1 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | ... | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | ha |
| 29 | 7 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | ... | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | ha |

30 rows × 262146 columns

In [376]: emoji1=emoji_index.drop(['index'],axis=1)

In [377]: emoji1

Out[377...]

| | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | ... | 262135 | 262136 | 262137 | 262138 | 262139 | 262140 | 262141 | 262142 | 262143 | label |
|----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|--------|--------|--------|--------|--------|--------|--------|--------|--------|-------|
| 0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | ... | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | angry |
| 1 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | ... | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | sad |
| 2 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | ... | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | happy |
| 3 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | ... | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | angry |
| 4 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | ... | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | angry |
| 5 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | ... | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | sad |
| 6 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | ... | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | angry |
| 7 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | ... | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | sad |
| 8 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | ... | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | sad |
| 9 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | ... | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | angry |
| 10 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | ... | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | angry |
| 11 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | ... | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | happy |
| 12 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | ... | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | happy |
| 13 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | ... | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | sad |
| 14 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | ... | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | angry |
| 15 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | ... | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | happy |
| 16 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | ... | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | sad |
| 17 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | ... | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | happy |
| 18 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | ... | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | happy |
| 19 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | ... | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | angry |
| 20 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | ... | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | happy |
| 21 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | ... | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | angry |

| | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | ... | 262135 | 262136 | 262137 | 262138 | 262139 | 262140 | 262141 | 262142 | 262143 | label |
|----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|--------|--------|--------|--------|--------|--------|--------|--------|--------|-------|
| 22 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | ... | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | happ |
| 23 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | ... | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | sad |
| 24 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | ... | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | angry |
| 25 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | ... | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | sad |
| 26 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | ... | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | sad |
| 27 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | ... | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | sad |
| 28 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | ... | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | happ |
| 29 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | ... | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | happ |

30 rows × 262145 columns

```
emoji1.to_csv('emojidata.csv')
```

In [378...]

```
x=emoji1.values[:, :-1]
y=emoji1.values[:, -1]
```

CREATE TRAINING & TESTING SET

In [379...]

```
from sklearn.model_selection import train_test_split
```

In [380...]

```
x_train,x_test,y_train,y_test=train_test_split(x,y,test_size=0.2,random_state=0)
```

In [381...]

```
x_train.shape
```

Out[381...]

```
(24, 262144)
```

In [382...]

```
y_train.shape
```

```
Out[382... (24,)
```

```
In [383... x_test
```

```
Out[383... array([[1.0, 1.0, 1.0, ..., 1.0, 1.0, 1.0],  
   [1.0, 1.0, 1.0, ..., 1.0, 1.0, 1.0],  
   [1.0, 1.0, 1.0, ..., 1.0, 1.0, 1.0],  
   [1.0, 1.0, 1.0, ..., 1.0, 1.0, 1.0],  
   [1.0, 1.0, 1.0, ..., 1.0, 1.0, 1.0],  
   [1.0, 1.0, 1.0, ..., 1.0, 1.0, 1.0]], dtype=object)
```

```
In [384... y_test.shape
```

```
Out[384... (6,)
```

PERFORM PCA

```
In [385... from sklearn import decomposition
```

```
In [386... pca=decomposition.PCA(n_components=20,whiten=True,random_state=0)
```

```
In [387... pca
```

```
Out[387... ▾ PCA ⓘ ⓘ  
PCA(n_components=20, random_state=0, whiten=True)
```

```
In [388... pca.fit(x_train)
```

```
Out[388... ▾ PCA ⓘ ⓘ  
PCA(n_components=20, random_state=0, whiten=True)
```

```
In [389... from sklearn.decomposition import PCA  
pca = PCA(n_components=20)
```

```
x_train_pca = pca.fit_transform(x_train)
```

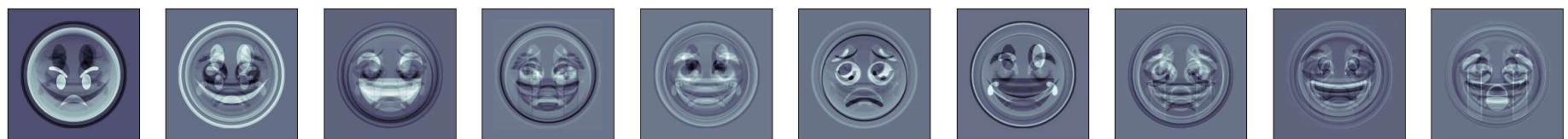
```
In [390... x_test_pca=pca.transform(x_test)
```

```
In [391... eigen=(np.reshape(x[10],(512,512)).astype(np.float64))
```

```
In [392... eigen
```

```
Out[392... array([[1., 1., 1., ..., 1., 1., 1.],
   [1., 1., 1., ..., 1., 1., 1.],
   [1., 1., 1., ..., 1., 1., 1.],
   ...,
   [1., 1., 1., ..., 1., 1., 1.],
   [1., 1., 1., ..., 1., 1., 1.],
   [1., 1., 1., ..., 1., 1., 1.]])
```

```
In [393... fig=plt.figure(figsize=(30,30))
for i in range(10):
    ex=fig.add_subplot(3, 10, i+1, xticks=[], yticks[])
    ex.imshow(pca.components_[i].reshape(eigen.shape),cmap=plt.cm.bone)
plt.show()
```



```
In [394... from sklearn.ensemble import RandomForestClassifier
from sklearn.metrics import classification_report, accuracy_score
```

```
In [395... rf_classifier = RandomForestClassifier()
rf_classifier.fit(x_train_pca, y_train)
```

```
Out[395... ▾ RandomForestClassifier ⓘ ?
```

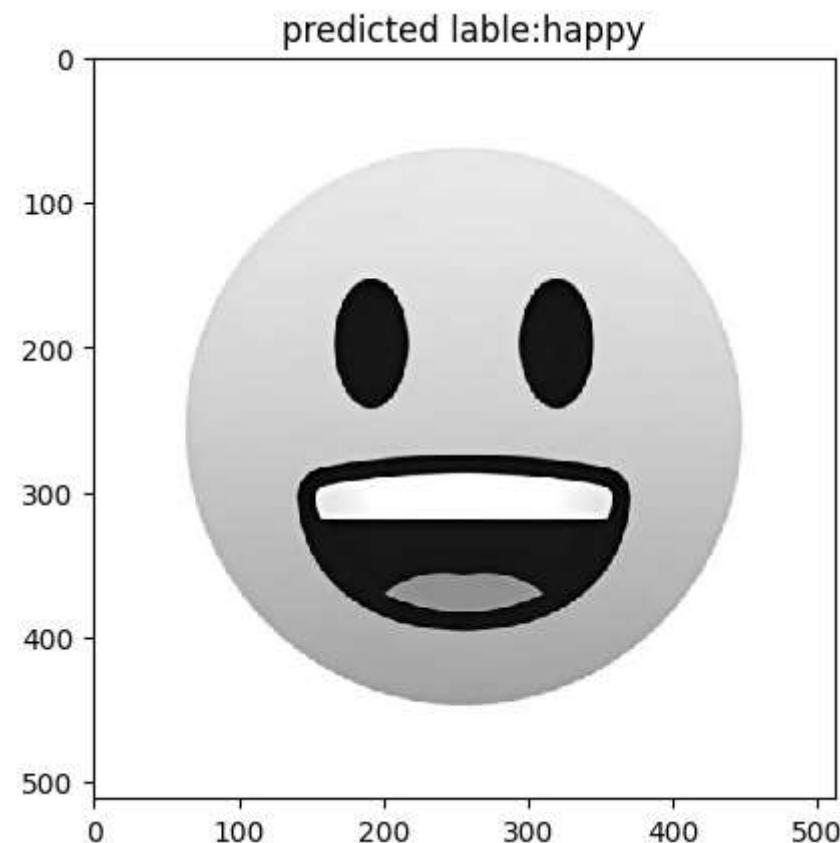
```
RandomForestClassifier()
```

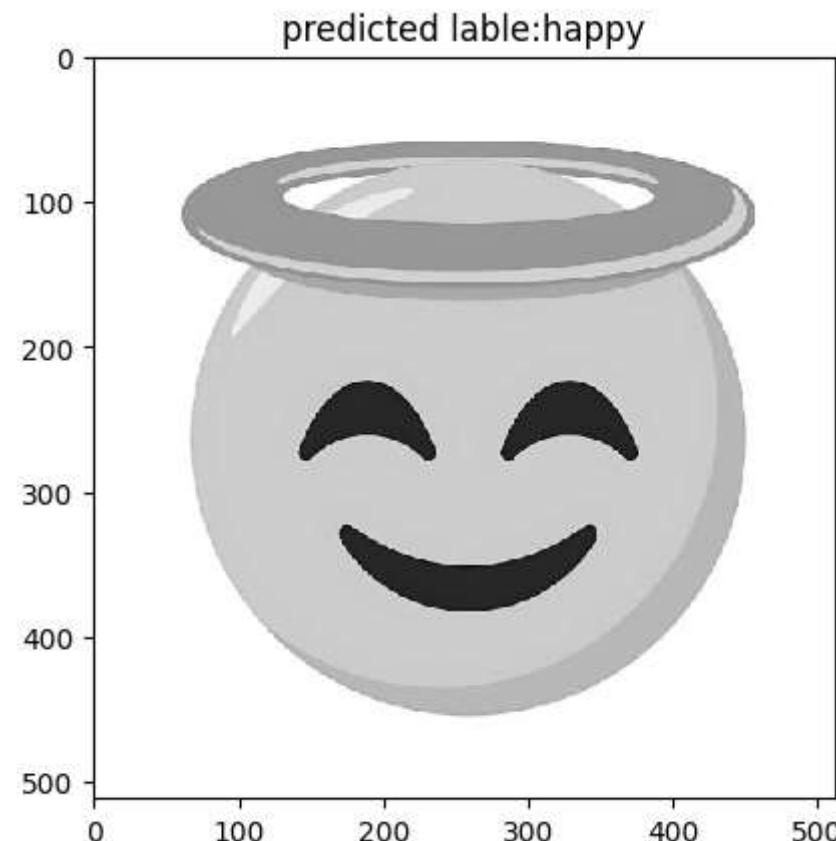
```
In [396... y_pred=rf_classifier.predict(x_test_pca)
```

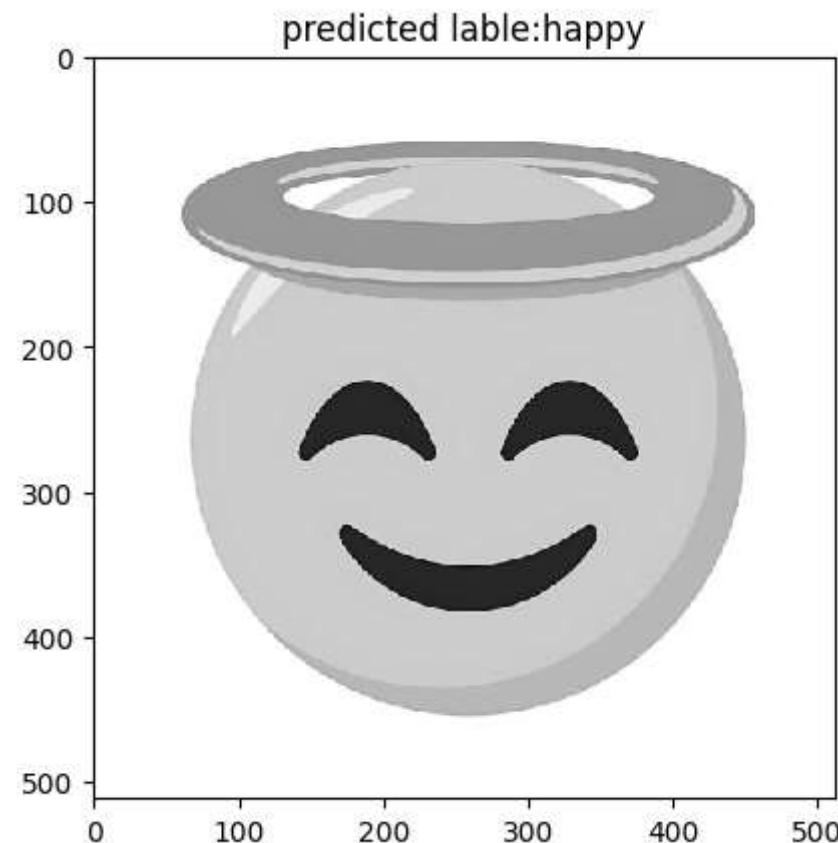
```
In [397... y_pred
```

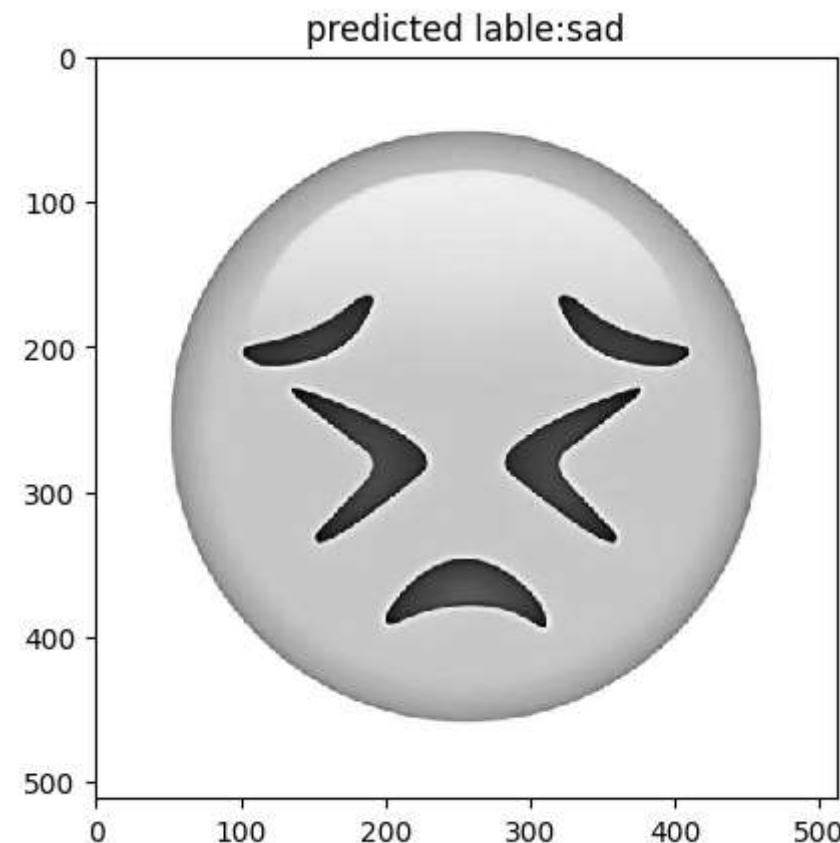
```
Out[397... array(['happy', 'happy', 'sad', 'angry', 'sad', 'angry'], dtype=object)
```

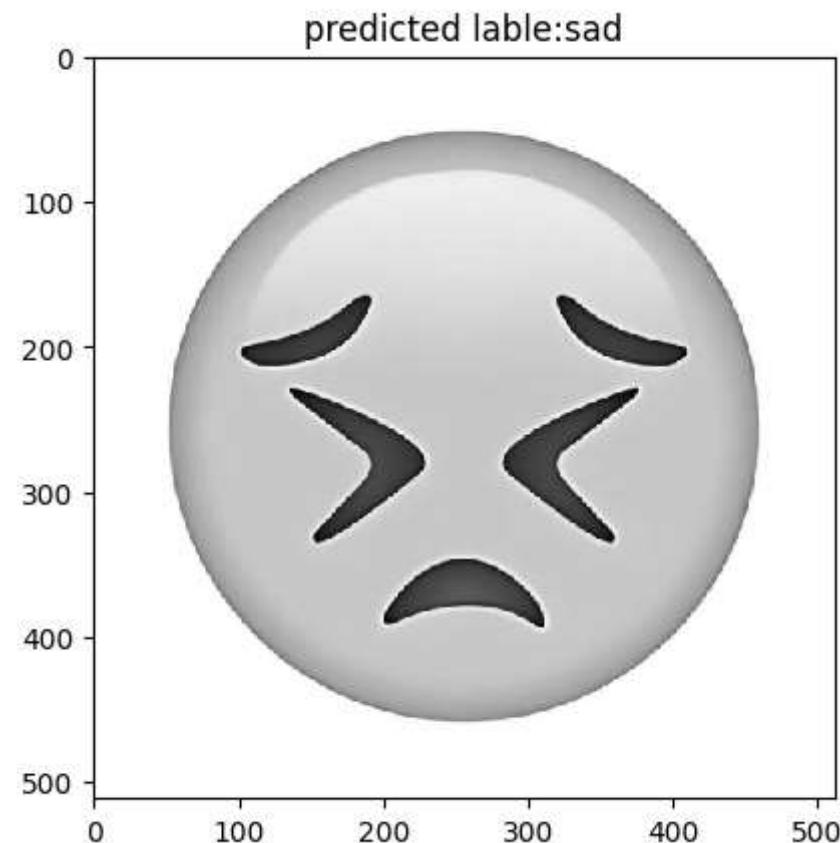
```
In [398... for i in (np.random.randint(0,6,6)):
    predicted_image=(np.reshape(x_test[i],(512,512)).astype(np.float64))
    plt.title('predicted lable:{0}'.format(y_pred[i]))
    plt.imshow(predicted_image, interpolation='nearest',cmap='gray')
    plt.show()
```

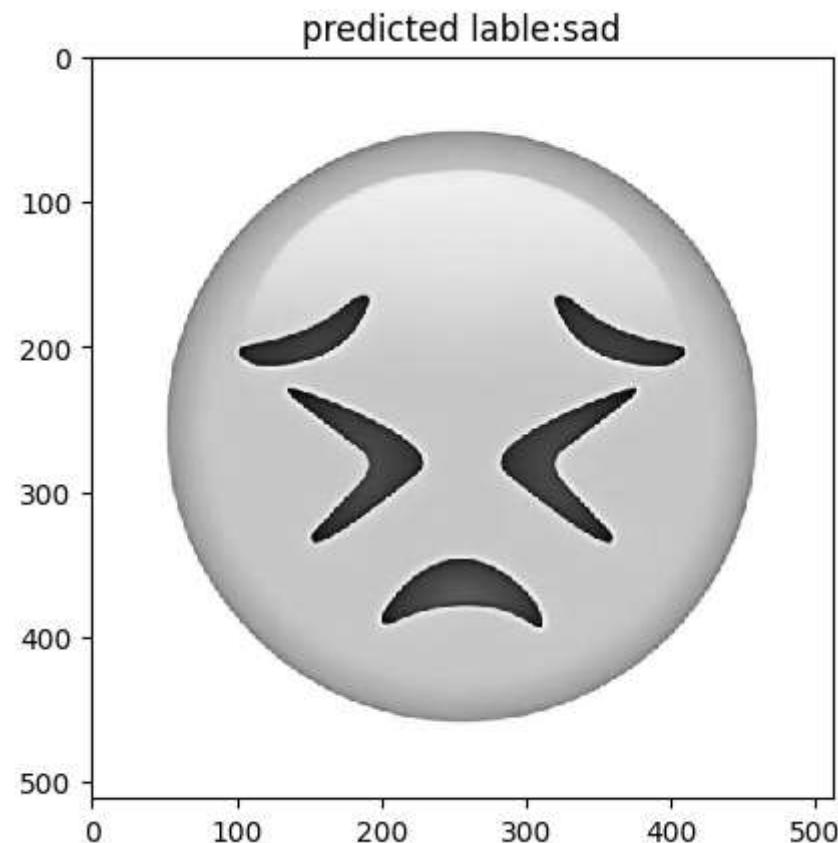












In [399...]

```
rf_classifier = RandomForestClassifier(n_estimators=100, random_state=42)
rf_classifier.fit(x_train_pca, y_train)
y_pred = rf_classifier.predict(x_test_pca)

accuracy = accuracy_score(y_test, y_pred)
print(f"Accuracy: {accuracy*100:.2f}%")

print("Classification Report:")
print(classification_report(y_test, y_pred))
```

Accuracy: 100.00%

Classification Report:

| | precision | recall | f1-score | support |
|--------------|-----------|--------|----------|---------|
| angry | 1.00 | 1.00 | 1.00 | 2 |
| happy | 1.00 | 1.00 | 1.00 | 2 |
| sad | 1.00 | 1.00 | 1.00 | 2 |
| accuracy | | | 1.00 | 6 |
| macro avg | 1.00 | 1.00 | 1.00 | 6 |
| weighted avg | 1.00 | 1.00 | 1.00 | 6 |

In []:

In []:

In []: