pr_dom6

May 30, 2021

[62]:
```
from sklearn.datasets import fetch_olivetti_faces
import matplotlib.pyplot as plt
import numpy as np
import seaborn as sns
from sklearn.decomposition import PCA
import pandas as pd
from sklearn.model_selection import train_test_split
from sklearn.metrics import mean_squared_error
```

[5]:
```
data=fetch_olivetti_faces()
```

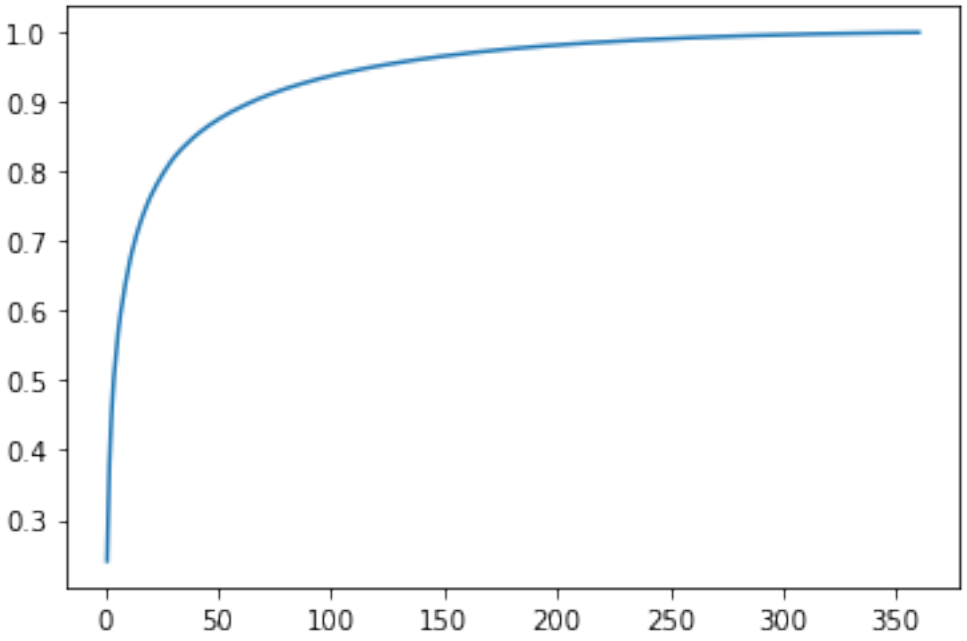
[8]:
```
data.data.shape
```

[8]:
```
(400, 4096)
```

1 Wybór ilości komponentów

[67]:
```
X=data.data
y=data.target
x_train,x_test,y_train,y_test=train_test_split(X,y,test_size=0.
 ↪1,random_state=2342)
pca=PCA().fit(x_train)
ans=[]
for i in range(350):
    model=PCA(n_components=i)
    m=model.fit_transform(x_train)
    ans.append(mean_squared_error(model.inverse_transform(m),x_train))
```

[67]:
```
[<matplotlib.lines.Line2D at 0x24d557f8b20>]
```



[189]:
```
pca=PCA(n_components=200)
tran=pca.fit_transform(x_train)
df=pca.inverse_transform(tran)
```

2 Stopień kompresji

[252]:
```
x_train[0].shape[0]/tran[0].shape[0]
```

[252]: 20.48

3 Wybrane obrazy

[190]:
```
plt.gray()
nrow=4
ncol=6
plt.figure(figsize=(15,10))
for i in range(nrow*ncol):
    ax=plt.subplot(nrow,ncol,i+1)
    ax.matshow(x_train[i].reshape((64,64)))
```

```
<Figure size 432x288 with 0 Axes>
```

4 Transformacja odwrotna

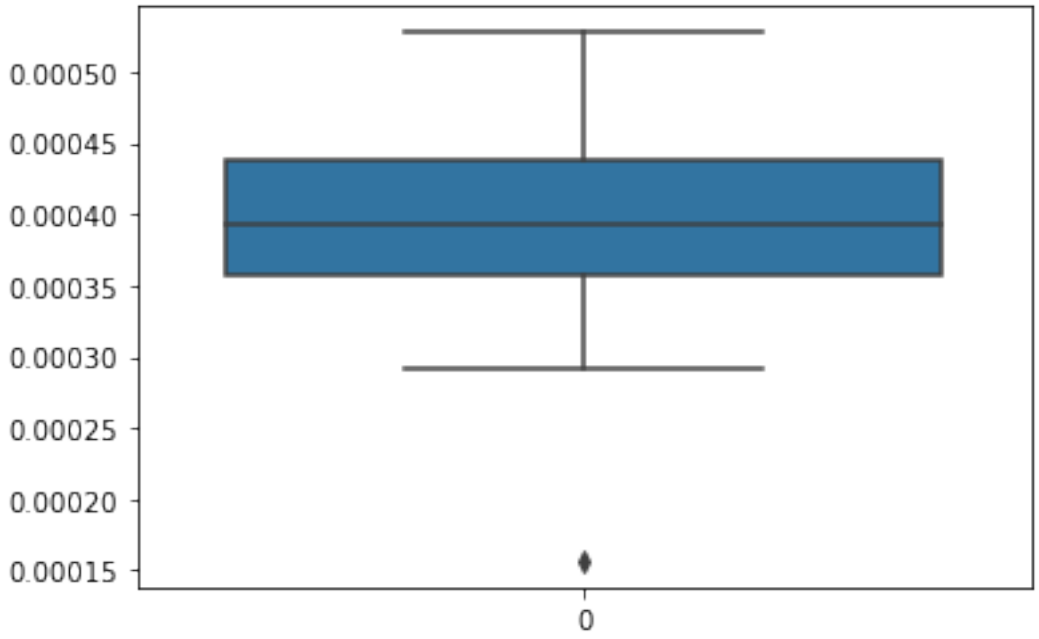
[228]:
```
plt.gray()
answer=[]
plt.figure(figsize=(15,10))
for i in range(nrow*ncol):
    ax=plt.subplot(nrow,ncol,i+1)
    ans=[x_train[i]]
    ans=pca.transform(ans)
    ans=pca.inverse_transform(ans)
    ax.matshow(ans.reshape((64,64)))
    answer.append(ans)
```

```
<Figure size 432x288 with 0 Axes>
```

4.1 Błąd rekonstrukcji

[249]:
```
error=[0]*24
for i in range(24):
    error[i]=mean_squared_error(x_train[i],answer[i].reshape(64*64))
sns.boxplot(data=[error])
```

[249]: <AxesSubplot:>



5 Zmodyfikowane obrazy

5.0.1 Obrócone

[192]:
```
transposed=np.transpose(x_train[1].reshape((64,64))).reshape((1,64*64))
```

[241]:
```
plt.gray()
answer0=[]
plt.figure(figsize=(15,10))
for i in range(nrow*ncol):
    ax=plt.subplot(nrow,ncol,i+1)
    ans=[x_train[i].reshape((64,64)).transpose().reshape(64*64)]
    ans=pca.transform(ans)
    ans=pca.inverse_transform(ans)
    ax.matshow(ans.reshape((64,64)))
    answer0.append(ans)
```

```
<Figure size 432x288 with 0 Axes>
```

5.0.2 Przyciemnione

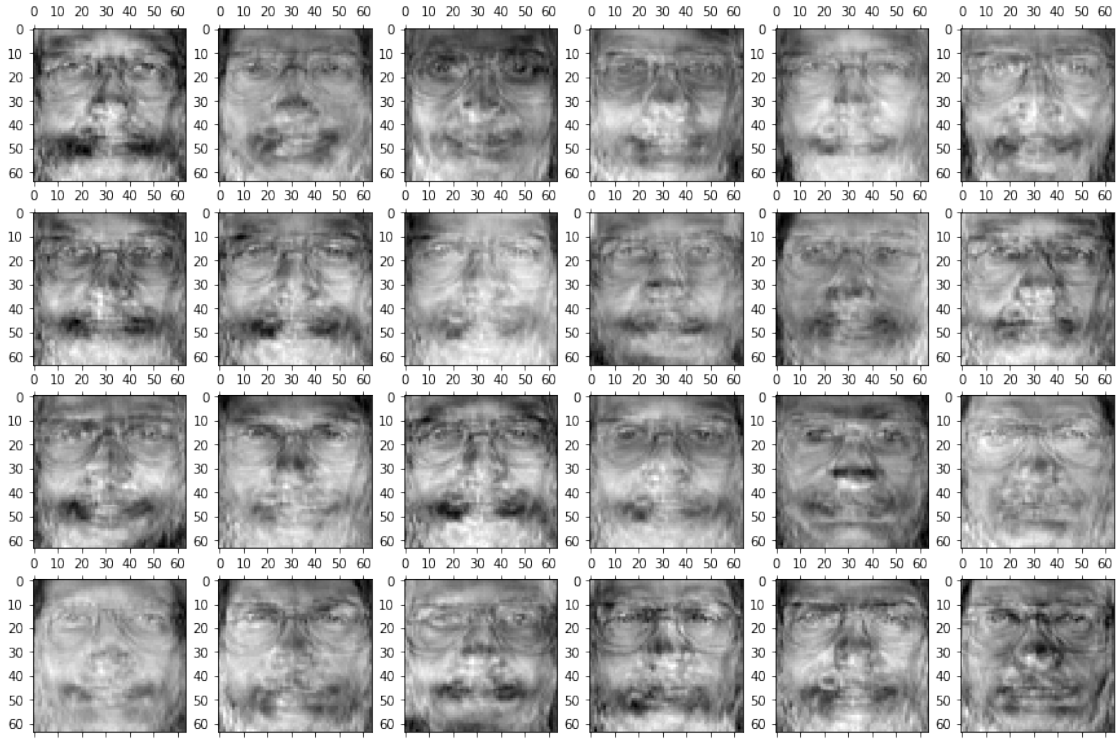
[258]:
```
plt.gray()
answer1=[]
plt.figure(figsize=(15,10))
for i in range(nrow*ncol):
    ax=plt.subplot(nrow,ncol,i+1)
    ans=x_train[i]+0.5
    ans=[ans]
    ans=pca.transform(ans)
    ans=pca.inverse_transform(ans)
    ax.matshow(ans.reshape((64,64)))
    answer1.append(ans)
```

```
<Figure size 432x288 with 0 Axes>
```

5.0.3 Do góry nogami

[244]:
```
plt.gray()
answer2=[]
plt.figure(figsize=(15,10))
for i in range(nrow*ncol):
    ax=plt.subplot(nrow,ncol,i+1)
    ans=[np.flip(x_train[i].reshape((64,64)),axis=0).reshape(64*64)]
    ans=pca.transform(ans)
    ans=pca.inverse_transform(ans)
    ax.matshow(ans.reshape((64,64)))
    answer2.append(ans)
```

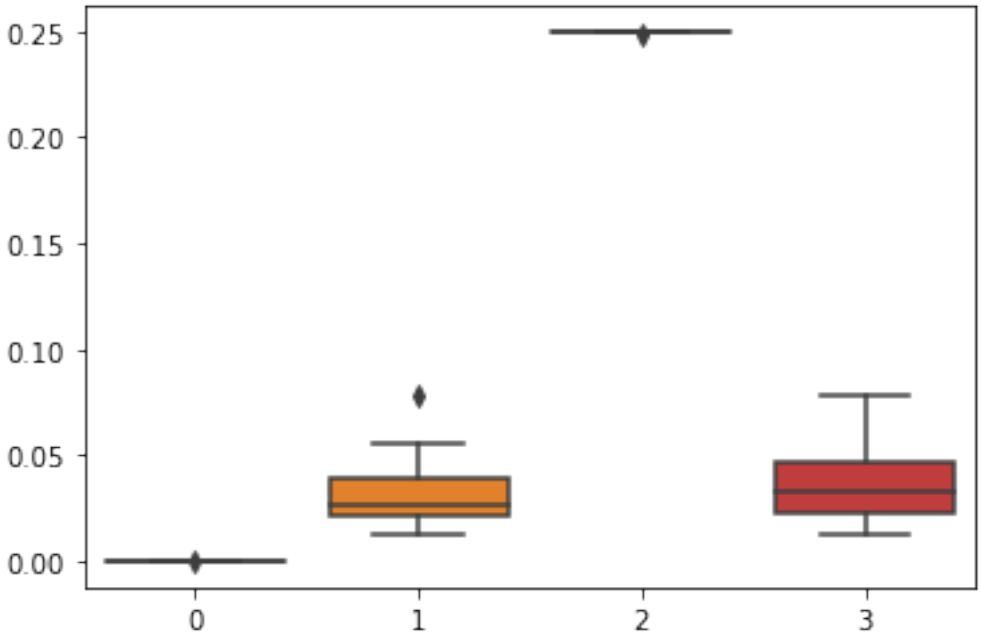
```
<Figure size 432x288 with 0 Axes>
```



6 Błąd rekonstrukcji dla modyfikacji

[259]:
```
error=[0]*24
error0=[0]*24
error1=[0]*24
error2=[0]*24
for i in range(24):
    error[i]=mean_squared_error(x_train[i],answer[i].reshape(64*64))
    error0[i]=mean_squared_error(x_train[i],answer0[i].reshape(64*64))
    error1[i]=mean_squared_error(x_train[i],answer1[i].reshape(64*64))
    error2[i]=mean_squared_error(x_train[i],answer2[i].reshape(64*64))
sns.boxplot(data=[error,error0,error1,error2])
```

[259]:
```
<AxesSubplot:>
```



7 Wnioski

RMSE daje znacznie większe wartości dla zmodyfikowanych obrazów, co może okazać się pomocne przy próbie wykrycia, czy obrazy były modyfikowane. Co ciekawe zarówno przyciemnienie, jak i niezmienione obrazy mają bardzo skupione wartości błędu, różnią się natomiast tym ,że niezmienione obrazy mają RMSE bliskie 0, a przyciemnione nie.