BARI ANKIT (56)

Exp : Principle Component Analysis

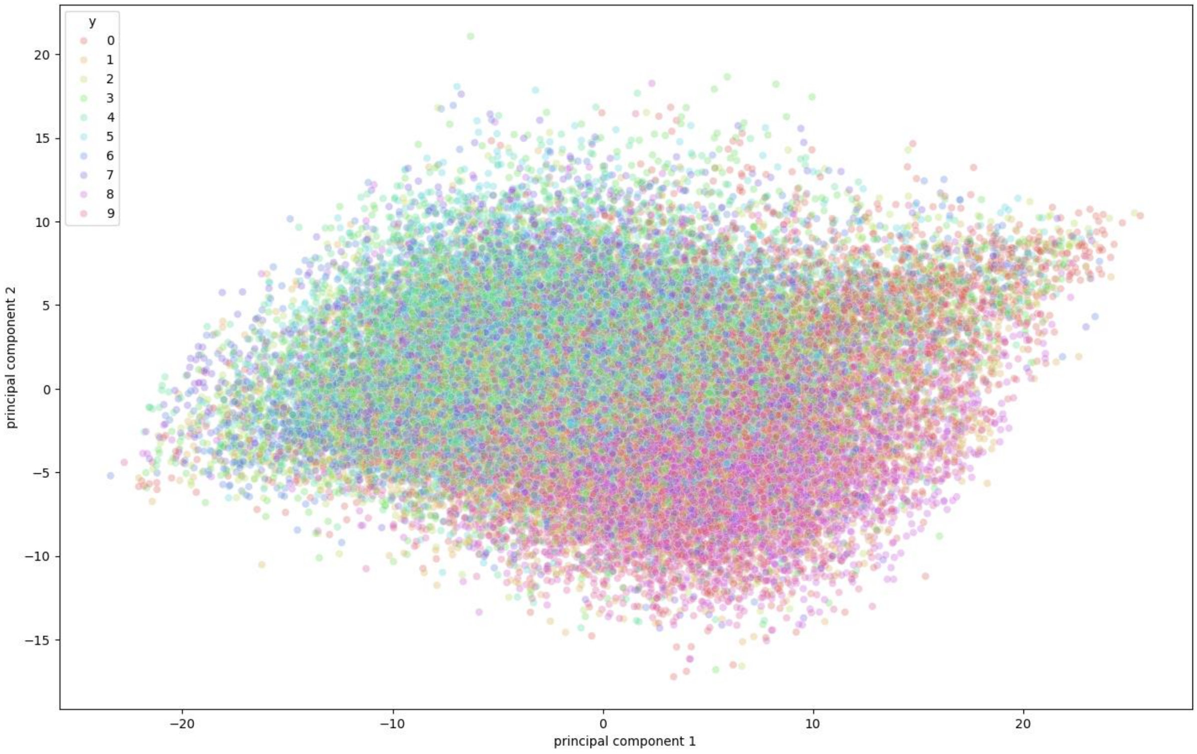
Code :

```
# %%
from tensorflow.keras.datasets import cifar10


# %%
(X_train, y_train), (X_test, y_test) = cifar10.load_data()


# %%
print('Traning data shape:', X_train.shape)
print('Testing data shape:', X_test.shape)


# %%
y_train.shape, y_test.shape


# %%
import matplotlib.pyplot as plt
%matplotlib inline


# %%
label_dict = {
 0: 'airplane',
 1: 'automobile',
 2: 'bird',
 3: 'cat',
 4: 'deer',
 5: 'dog',
 6: 'frog',
 7: 'horse',
 8: 'ship',
 9: 'truck',
}


# %%
import numpy as np


# %%
plt.figure(figsize=[5,5])


# Display the first image in training data
plt.subplot(121)
curr_img = np.reshape(X_train[0], (32,32,3))
plt.imshow(curr_img)
print(plt.title("(Label: " + str(label_dict[y_train[0][0]]) + ")"))


# Display the first image in testing data
plt.subplot(122)
curr_img = np.reshape(X_test[0],(32,32,3))
plt.imshow(curr_img)
print(plt.title("(Label: " + str(label_dict[y_test[0][0]]) + ")"))


# %%
```

```
np.min(X_train), np.max(X_train)


# %%

X_train = X_train / 255.0


# %%

np.min(X_train), np.max(X_train)


# %%

X_train.shape


# %%

x_train_flat = X_train.reshape(-1,3072)


# %%

feat_cols = ['pixel'+str(i) for i in range(x_train_flat.shape[1])]


# %%

import pandas as pd


# %%

df_cifar = pd.DataFrame(x_train_flat, columns=feat_cols)


# %%

df_cifar['label'] = y_train

print('Dataframe Size : {}'.format(df_cifar.shape))

# %%

from sklearn.decomposition import PCA
```

```
# %%
pca_cifar = PCA(n_components=2)
principalComponents_cifar = pca_cifar.fit_transform(df_cifar.iloc[:,:-1])

# %%
principal_cifar_Df = pd.DataFrame(data = principalComponents_cifar
             , columns = ['principal component 1', 'principal component 2'])
principal_cifar_Df['y'] = y_train

# %%
principal_cifar_Df.head()

# %%
print('Explained variation per PCA: {}'.format(pca_cifar.explained_variance_ratio_))

# %%
import seaborn as sns
plt.figure(figsize=(16,10))
sns.scatterplot(
    x="principal component 1", y="principal component 2",
    hue="y",
    palette=sns.color_palette("hls", 10),
    data=principal_cifar_Df,
    legend="full",
    alpha=0.3
)
# %%
```

Output :



y
0
1
2
3
4
5
6
7
8
9
20
15
10
5
0
−5
−10
−15
principal component 2
−20
−10
0
10
20
principal component 1