Pratham Nagar - 59 Exp 9

```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
from sklearn.decomposition import PCA
import seaborn as sns
from tensorflow.keras.datasets import fashion_mnist
%matplotlib inline
```

```
# Load dataset
(X_train, y_train), (X_test, y_test) = fashion_mnist.load_data()

# Print shapes
print('Training data shape:', X_train.shape)
print('Testing data shape:', X_test.shape)
print('Training labels shape:', y_train.shape)
print('Testing labels shape:', y_test.shape)
```

```
Downloading data from https://storage.googleapis.com/tensorflow/tf-keras-datasets/train-labels-idx1-ubyte.gz
29515/29515 ━━━━━━━━━━━━━━━━━━━━ 0s 0us/step
Downloading data from https://storage.googleapis.com/tensorflow/tf-keras-datasets/train-images-idx3-ubyte.gz
26421880/26421880 ━━━━━━━━━━━━━━━━━━━━ 0s 0us/step
Downloading data from https://storage.googleapis.com/tensorflow/tf-keras-datasets/t10k-labels-idx1-ubyte.gz
5148/5148 ━━━━━━━━━━━━━━━━━━━━ 0s 0us/step
Downloading data from https://storage.googleapis.com/tensorflow/tf-keras-datasets/t10k-images-idx3-ubyte.gz
4422102/4422102 ━━━━━━━━━━━━━━━━━━━━ 0s 0us/step
Training data shape: (60000, 28, 28)
Testing data shape: (10000, 28, 28)
Training labels shape: (60000,)
Testing labels shape: (10000,)
```

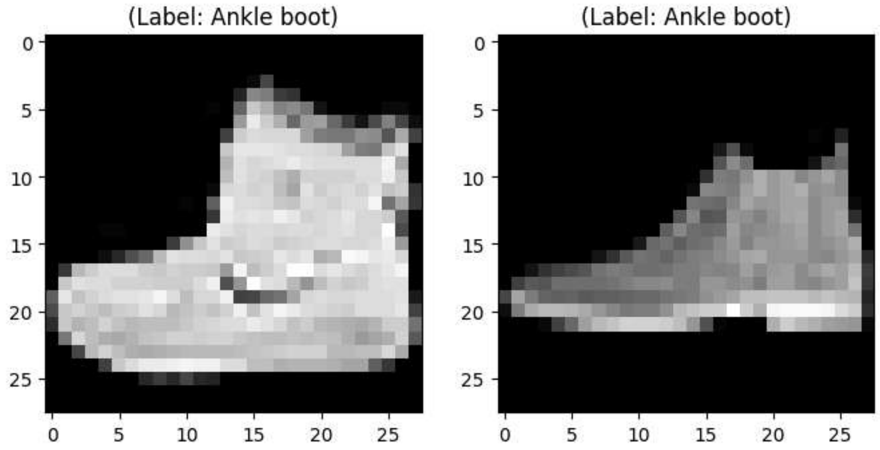
```
# Fashion MNIST labels
label_dict = {
    0: 'T-shirt/top',
    1: 'Trouser',
    2: 'Pullover',
    3: 'Dress',
    4: 'Coat',
    5: 'Sandal',
    6: 'Shirt',
    7: 'Sneaker',
    8: 'Bag',
    9: 'Ankle boot'
}

# Display sample images
plt.figure(figsize=[8,5])

# First training image
plt.subplot(121)
plt.imshow(X_train[0], cmap='gray')
plt.title("(Label: " + str(label_dict[y_train[0]]) + ")")

# First test image
plt.subplot(122)
plt.imshow(X_test[0], cmap='gray')
plt.title("(Label: " + str(label_dict[y_test[0]]) + ")")

plt.show()
```



```
# Normalize pixel values
X_train = X_train / 255.0
X_test = X_test / 255.0

# Check new range
print('Min value:', np.min(X_train))
print('Max value:', np.max(X_train))
```

```
Min value: 0.0
Max value: 1.0
```

```
# Reshape from 28x28 to 784-dimensional vector
x_train_flat = X_train.reshape(-1, 784)

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

# Create DataFrame
df_fashion = pd.DataFrame(x_train_flat, columns=feat_cols)
df_fashion['label'] = y_train
print('DataFrame size:', df_fashion.shape)
```

```
DataFrame size: (60000, 785)
```

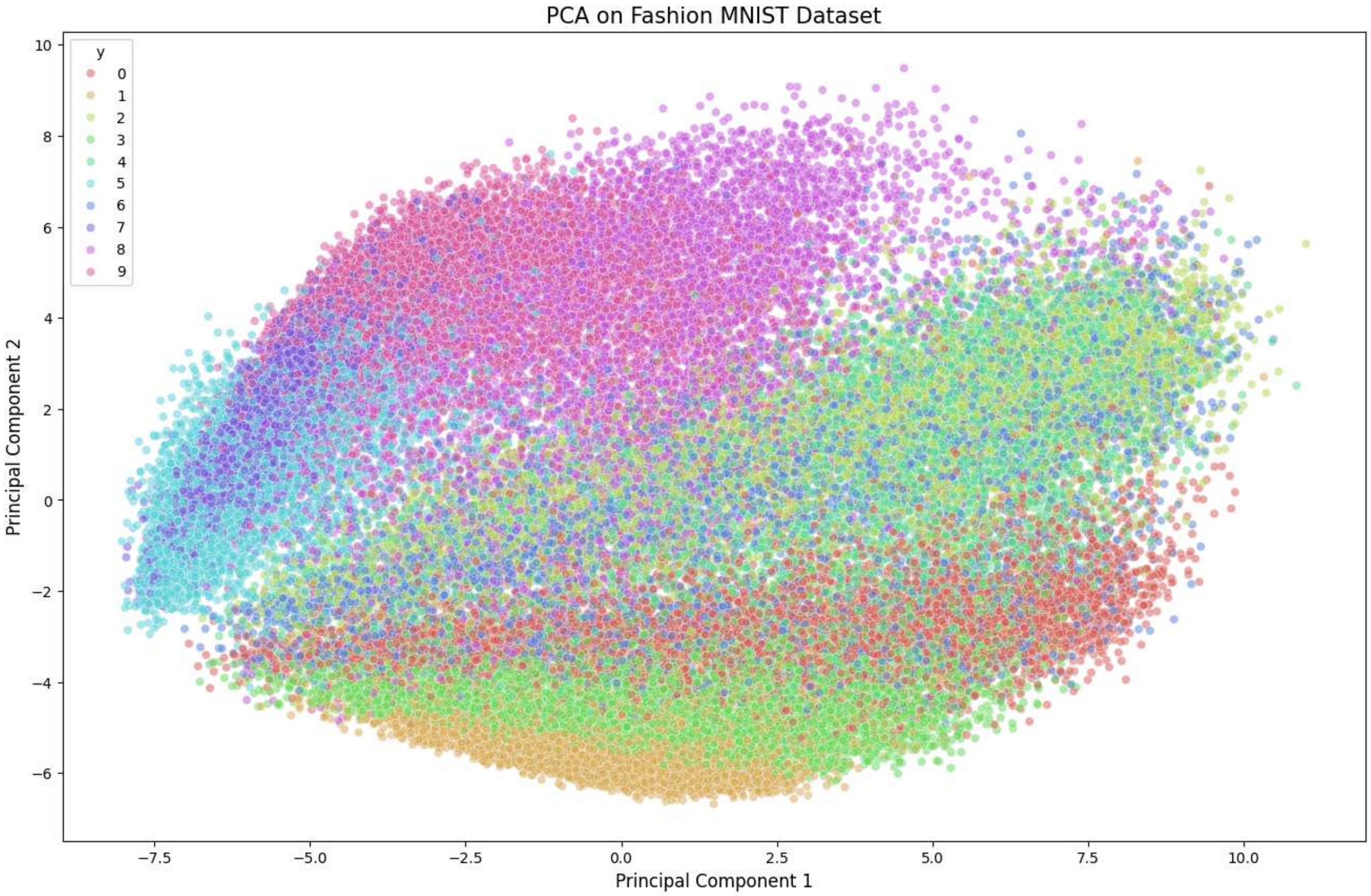
```
# Apply PCA
pca_fashion = PCA(n_components=2)
principalComponents_fashion = pca_fashion.fit_transform(df_fashion.iloc[:,:-1])

# Create DataFrame with principal components
principal_fashion_Df = pd.DataFrame(data=principalComponents_fashion,
                                  columns=['principal component 1', 'principal component 2'])
principal_fashion_Df['y'] = y_train

# Show explained variance
print('Explained variation per principal component: {}'.format(pca_fashion.explained_variance_ratio_))
```

```
Explained variation per principal component: [0.29039228 0.1775531 ]
```

```
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_fashion_Df,
    legend="full",
    alpha=0.5
)
plt.title('PCA on Fashion MNIST Dataset', fontsize=15)
plt.xlabel('Principal Component 1', fontsize=12)
plt.ylabel('Principal Component 2', fontsize=12)
plt.show()
```


PCA on Fashion MNIST Dataset
y
0
1
2
3
4
5
6
7
8
9
Principal Component 2
10
8
6
4
2
0
−2
−4
−6
Principal Component 1
−7.5
−5.0
−2.5
0.0
2.5
5.0
7.5
10.0