Machine Learning Group Activity

Group No: GB9

Group Members:

1. Chaitanya Nawale (BECOB15) - Reduce dimensions using PCA technique
2. Ajinkya Patil (BECOB224) - Reduce dimensions using Sparse PCA
3. Mehul Suryavanshi (BECOB260) - Reduce dimensions using SelectKBest
4. Chinmayee Taralkar (BECOB262) - Kernel PCA (Non-linear Data set)

Reduce dimensions using PCA technique on fashion_mnist dataset -> BECOB215

```
from keras.datasets import fashion_mnist
import matplotlib.pyplot as plt
import numpy as np
from time import time
from sklearn.svm import SVC
from sklearn import metrics
from sklearn.metrics import confusion_matrix
from sklearn.preprocessing import StandardScaler, MinMaxScaler
from sklearn.decomposition import PCA
from sklearn.pipeline import Pipeline
from sklearn.manifold import TSNE
import umap
```

```
#load dataset
(x_train, y_train), (x_test, y_test) = fashion_mnist.load_data()
print(x_train.shape, y_train.shape)
print(x_test.shape, y_test.shape)

# Reshaping the images(28x28 pixel) into a single 784px vector using .reshape
x_train = np.reshape(x_train, (len(x_train), -1))/255
x_test = np.reshape(x_test, (len(x_test), -1))/255

print(x_train.shape, y_train.shape)
print(x_test.shape, y_test.shape)
```

```
(60000, 28, 28) (60000,)
(10000, 28, 28) (10000,)
```

```
(60000, 784) (60000,)
(10000, 784) (10000,)
```

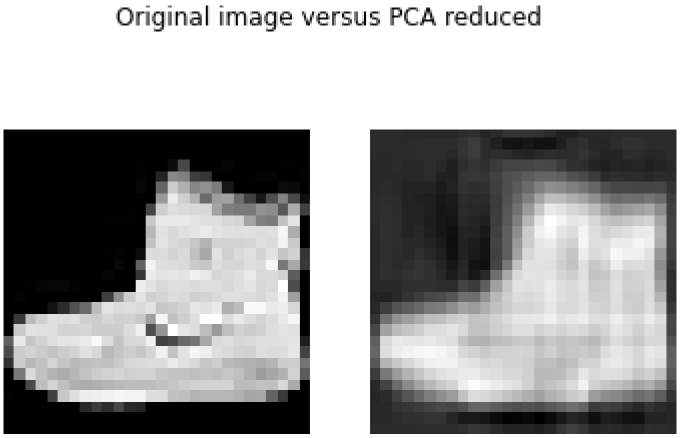
```
# Set number of components to extract and scale each feature to have a variance of 1
steps = [('scaling', StandardScaler()), ('pca', PCA(n_components=0.85))]
pipeline = Pipeline(steps)
pipeline.fit(x_train)
```

```
Pipeline(memory=None,
         steps=[('scaling',
                 StandardScaler(copy=True, with_mean=True, with_std=True)),
                ('pca',
                 PCA(copy=True, iterated_power='auto', n_components=0.85,
                     random_state=None, svd_solver='auto', tol=0.0,
                     whiten=False))],
         verbose=False)
```

```
#Check number of components extracted to account for 85% of the variance
pipeline['pca'].n_components_
```

```
81
```

```
reduced = pipeline.inverse_transform(pipeline.transform(x_train))

# Visualize the PCA reduced number
fig, (ax1, ax2) = plt.subplots(1, 2)
ax1.matshow(x_train[0].reshape(28,28), cmap='gray')
ax2.matshow(reduced[0].reshape(28,28), cmap='gray')
ax1.set_axis_off()
ax2.set_axis_off()
fig.suptitle("Original image versus PCA reduced".format(y_train[0]))
plt.show()
```



Reduce dimensions using Sparse PCA on fashion_mnist dataset -> BECOB224

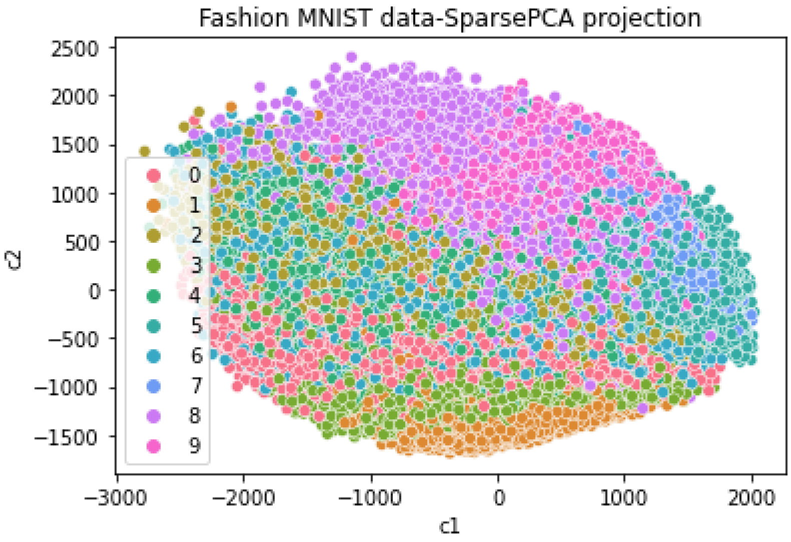
```
from sklearn.decomposition import SparsePCA
from numpy import reshape
import seaborn as sns
import pandas as pd
from keras.datasets import fashion_mnist
```

```
#load dataset
(x_train, y_train), (x_test , y_test) = fashion_mnist.load_data()
print(x_train.shape)
x_fashion_mnist = reshape(x_train, [x_train.shape[0], x_train.shape[1]*x_train.shape[2]])
print(x_fashion_mnist.shape)
```

```
(60000, 28, 28)
(60000, 784)
```

```
sparsePCA = SparsePCA(n_components=2, random_state=123)
z = sparsePCA.fit_transform(x_fashion_mnist)
df = pd.DataFrame()
df["y"] = y_train
df["c1"] = z[:,0]
df["c2"] = z[:,1]
sns.scatterplot(x="c1", y="c2", hue=df.y.tolist(),
                palette=sns.color_palette("husl", 10),
                data=df).set(title="Fashion MNIST data-SparsePCA projection")
```

```
[Text(0.5, 1.0, 'Fashion MNIST data-SparsePCA projection')]
```



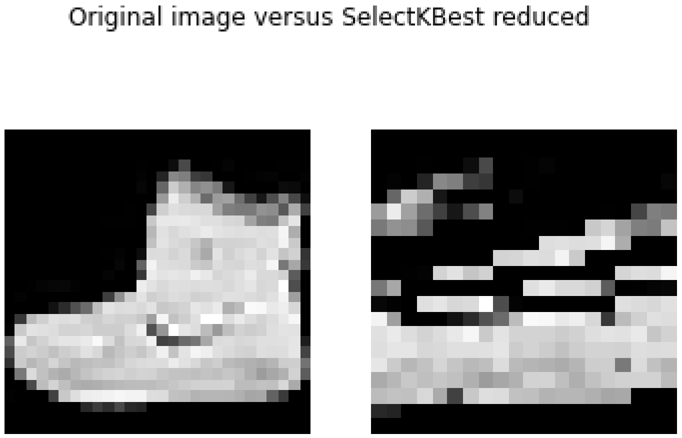
Reduce dimensions using SelectKBest on fashion_mnist dataset -> BECOB260

```
from keras.datasets import fashion_mnist
from keras.datasets import fashion_mnist
import matplotlib.pyplot as plt
import numpy as np
from time import time
from sklearn.svm import SVC
from sklearn import metrics
from sklearn.metrics import confusion_matrix
from sklearn.preprocessing import StandardScaler, MinMaxScaler
from sklearn.decomposition import PCA
from sklearn.pipeline import Pipeline
from sklearn.manifold import TSNE
from sklearn.feature_selection import SelectKBest,chi2
import umap
```

```
#load dataset
x_train = np.reshape(x_train, (len(x_train), -1))/255
x_test = np.reshape(x_test, (len(x_test), -1))/255
print(x_train.shape, y_train.shape)
X_new = SelectKBest(chi2, k=400).fit_transform(x_fashion_mnist, y_train)
print(X_new.shape)
```

```
(60000, 784) (60000,)
(60000, 400)
```

```
# visualize the PCA reduced number
fig, (a1, a2) = plt.subplots(1, 2)
a1.matshow(x_train[0].reshape(28,28), cmap='gray')
a2.matshow(X_new[0].reshape(20,20), cmap='gray')
a1.set_axis_off()
a2.set_axis_off()
fig.suptitle("Original image versus SelectKBest reduced".format(y_train[0]))
plt.show()
```



Kernel PCA (Non-linear Data set) -> BECOB262

```
from sklearn.decomposition import KernelPCA
import matplotlib.pyplot as plt
```

```
#load dataset
(x_train, y_train), (x_test, y_test) = fashion_mnist.load_data()
print(x_train.shape, y_train.shape)
print(x_test.shape, y_test.shape)

# Reshaping the images(28x28 pixel) into a single 784px vector using .reshape
x_train = np.reshape(x_train, (len(x_train), -1))/255
x_test = np.reshape(x_test, (len(x_test), -1))/255

print(x_train.shape, y_train.shape)
print(x_test.shape, y_test.shape)
```

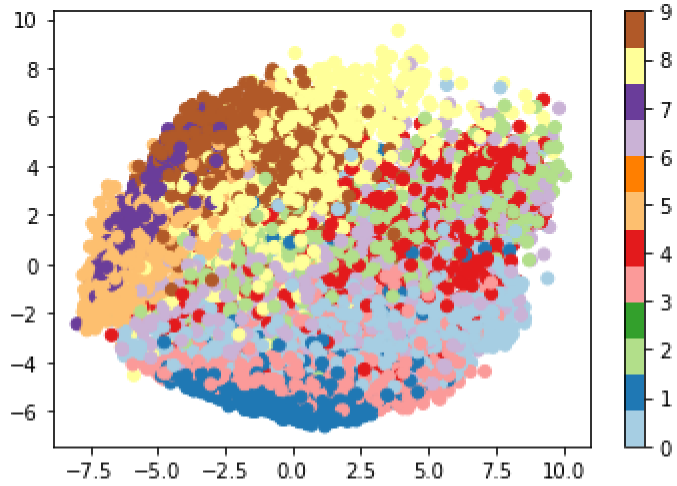
```
(60000, 28, 28) (60000,)
(10000, 28, 28) (10000,)
(60000, 784) (60000,)
(10000, 784) (10000,)
```

```
transformer1 = KernelPCA(n_components=81, kernel='linear')
X_transformed1 = transformer1.fit_transform(x_test)

print(X_transformed1.shape)
```

```
(10000, 81)
/usr/local/lib/python3.7/dist-packages/sklearn/utils/extmath.py:530: RuntimeWarning: inv
  v *= signs[:, np.newaxis]
```

```
#Plot a projection on first 2 principal axis

plt.figure()
plt.scatter(X_transformed1[:,0],X_transformed1[:,1],c=y_test,cmap="Paired")
plt.colorbar()
```

```
<matplotlib.colorbar.Colorbar at 0x7f9386a43810>
```



Could not connect to the reCAPTCHA service. Please check your internet connection and reload to get a reCAPTCHA challenge.