Team members:
Sirisha Lanka - PES1201700294
Malavikka R - PES1201700794
Kevin Arulraj - PES1201700659

Assignment 2

Problem Statement:

Using a dataset write a code that generates principal component analysis (PCA) and singular value decomposition (SVD).

Dataset:

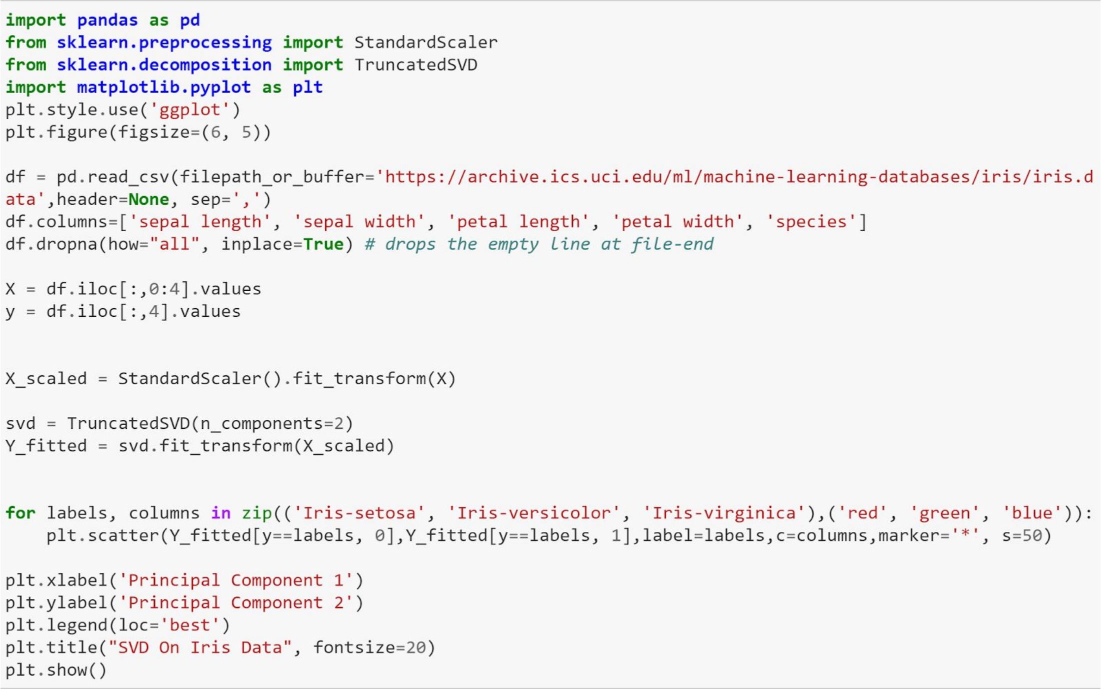
Iris dataset

Code:

#1 PCA

```
features = ['sepal length', 'sepal width', 'petal length', 'petal width']
x = df.loc[:, features].values

y = df.loc[:,['target']].values

x = StandardScaler().fit_transform(x)
pd.DataFrame(data = x, columns = features).head()
pca = PCA(n_components=2)

principalComponents = pca.fit_transform(x)
principalDf = pd.DataFrame(data = principalComponents, columns = ['principal component 1', 'principal component 2'])
principalDf.head(5)

df[['target']].head()
finalDf = pd.concat([principalDf, df[['target']]], axis = 1)
finalDf.head(5)
fig = plt.figure(figsize = (8,8))
ax = fig.add_subplot(1,1,1)
ax.set_xlabel('Principal Component 1', fontsize = 15)
ax.set_ylabel('Principal Component 2', fontsize = 15)
ax.set_title('2 Component PCA', fontsize = 20)


targets = ['Iris-setosa', 'Iris-versicolor', 'Iris-virginica']
colors = ['r', 'g', 'b']
for target, color in zip(targets,colors):
    indicesToKeep = finalDf['target'] == target
    ax.scatter(finalDf.loc[indicesToKeep, 'principal component 1'], finalDf.loc[indicesToKeep, 'principal component 2'], c = color, s = 50)
ax.legend(targets)
ax.grid()
```

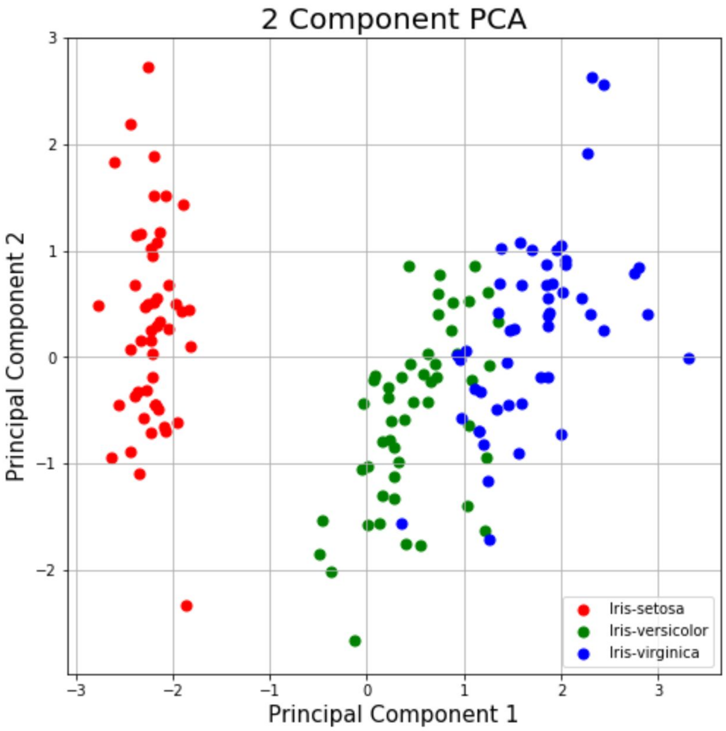
#2 SVD



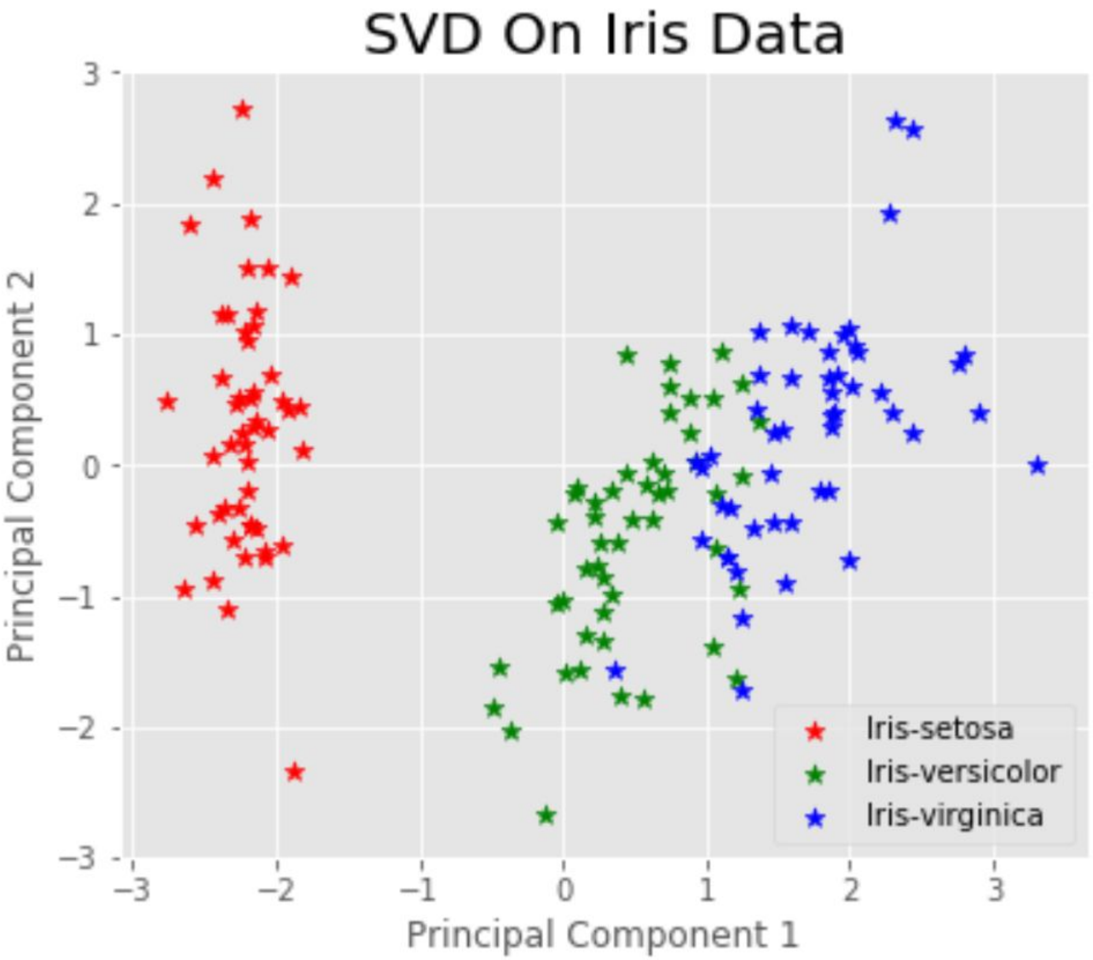
```
import pandas as pd
from sklearn.preprocessing import StandardScaler
from sklearn.decomposition import TruncatedSVD
import matplotlib.pyplot as plt
plt.style.use('ggplot')
plt.figure(figsize=(6, 5))

df = pd.read_csv(filepath_or_buffer='https://archive.ics.uci.edu/ml/machine-learning-databases/iris/iris.d
ata',header=None, sep=',')
df.columns=['sepal length', 'sepal width', 'petal length', 'petal width', 'species']
df.dropna(how="all", inplace=True) # drops the empty line at file-end

X = df.iloc[:,0:4].values
y = df.iloc[:,4].values


X_scaled = StandardScaler().fit_transform(X)

svd = TruncatedSVD(n_components=2)
Y_fitted = svd.fit_transform(X_scaled)


for labels, columns in zip(('Iris-setosa', 'Iris-versicolor', 'Iris-virginica'),('red', 'green', 'blue')):
    plt.scatter(Y_fitted[y==labels, 0],Y_fitted[y==labels, 1],label=labels,c=columns,marker='*', s=50)

plt.xlabel('Principal Component 1')
plt.ylabel('Principal Component 2')
plt.legend(loc='best')
plt.title("SVD On Iris Data", fontsize=20)
plt.show()
```

Results:

#1



#2


SVD On Iris Data
Principal Component 2
Principal Component 1
3
2
1
0
−1
−2
−3
−3
−2
−1
0
1
2
3
★ Iris-setosa
★ Iris-versicolor
★ Iris-virginica