DATA ANALYTICS ASSIGNMENT

NIVEDITHA C U

PES1201701640

SECTION 'C'

RACHANA H S

PES1201701726

SECTION 'E'

PROBLEM STATEMENT:

APPLYING PCA AND SVD ON A DATASET.

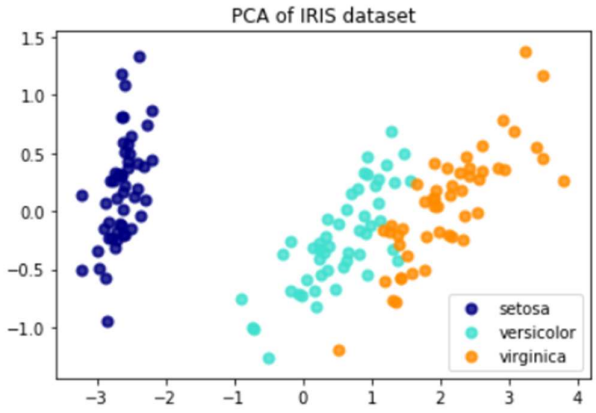
SCREENSHOTS

PCA

In [2]:

```
import matplotlib.pyplot as plt

from sklearn import datasets
from sklearn.decomposition import PCA

iris = datasets.load_iris()

X = iris.data
y = iris.target
target_names = iris.target_names

pca = PCA(n_components=2)
X_r = pca.fit(X).transform(X)
print('explained variance ratio (first two components): %s'
      % str(pca.explained_variance_ratio_))

plt.figure()
colors = ['navy', 'turquoise', 'darkorange']
lw = 2

for color, i, target_name in zip(colors, [0, 1, 2], target_names):
    plt.scatter(X_r[y == i, 0], X_r[y == i, 1], color=color, alpha=.8, lw=lw,
                label=target_name)
plt.legend(loc='best', shadow=False, scatterpoints=1)
plt.title('PCA of IRIS dataset')

plt.figure()
plt.show()
```

```
explained variance ratio (first two components): [0.92461872 0.05306648]
```

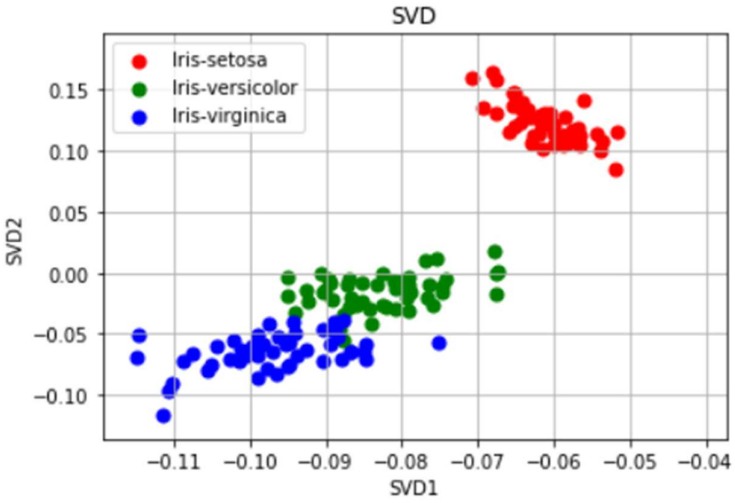



```
<Figure size 432x288 with 0 Axes>
```

SVD

In [2]:

```
from numpy import *
import operator
import matplotlib.pyplot as plt
import pandas as pd
from numpy.linalg import *

df = pd.read_csv('iris.data', names=["sepal_length","sepal_width","petal_length","petal_width","target"])


data = df[["sepal_length","sepal_width","petal_length","petal_width"]]
#calculate SVD
n = 2 # We will take two Singular Values
U, s, V = linalg.svd( data )
# eye() creates a matrix with ones on the diagonal and zeros elsewhere
Sig = mat(eye(n)*s[:n])
newdata = U[:,:n]
newdata = pd.DataFrame(newdata)
newdata.columns=["SVD1","SVD2"]
newdata.head()

newdata["target"]=df["target"]
fig = plt.figure()
ax = fig.add_subplot(1,1,1)
ax.set_xlabel("SVD1")
ax.set_ylabel("SVD2")
ax.set_title("SVD")
targets = ["Iris-setosa", "Iris-versicolor","Iris-virginica"]
colors = ["r", "g", "b"]
for target, color in zip(targets,colors):
 indicesToKeep = newdata["target"] == target
 ax.scatter(newdata.loc[indicesToKeep, "SVD1"]
 , newdata.loc[indicesToKeep, "SVD2"]
 , c = color
 , s = 50)
ax.legend(targets)
ax.grid()
```


SVD
Iris-setosa
Iris-versicolor
Iris-virginica
0.15
0.10
0.05
0.00
−0.05
−0.10
SVD2
−0.11
−0.10
−0.09
−0.08
−0.07
−0.06
−0.05
−0.04
SVD1