DATA ANALYTICS
Assignment 2

Amrutha S - USN : PES120700829

Yoshitha - USN : PES1201701744

Dhruv - USN : PES1201700122

Swathi - USN : PES1201701826

PROBLEM STATEMENT :
Perform dimensionality reduction on iris dataset using PCA and SVD.

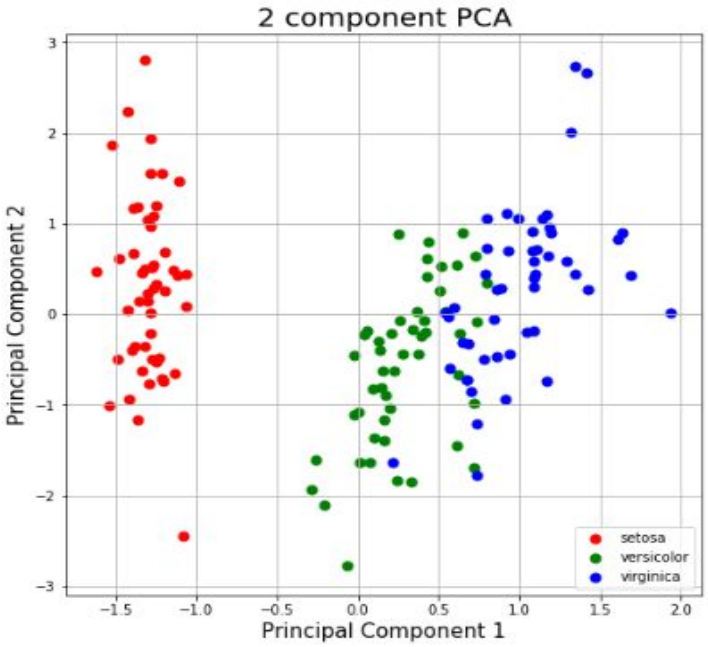
In [14]:
```
pca=PCA(n_components=2,whiten='True')
principalComponents=pca.fit_transform(x)
principalDf=pd.DataFrame(data=principalComponents,columns=['principal component 1', 'principal component
2'])
finaldf=pd.concat([principalDf,df[['species']]],axis=1)
```

In [15]:
```
finaldf.head(10)
```

Out[15]:

	principal component 1	principal component 2	species
0	-1.321232	0.500417	setosa
1	-1.214037	-0.702770	setosa
2	-1.379296	-0.356432	setosa
3	-1.341465	-0.622771	setosa
4	-1.394238	0.674312	setosa
5	-1.210927	1.552436	setosa
6	-1.425851	0.049668	setosa
7	-1.302647	0.232627	setosa
8	-1.362034	-1.162705	setosa
9	-1.274341	-0.488937	setosa

In [16]:
```
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 = ['setosa', 'versicolor', 'virginica']
colors = ['r', 'g', 'b']
for target, color in zip(targets,colors):
    indicesToKeep = finaldf['species'] == target
    ax.scatter(finaldf.loc[indicesToKeep, 'principal component 1'], finaldf.loc[indicesToKeep, 'principal
component 2'],c = color,s = 50)
ax.legend(targets)
ax.grid()
```



In [27]:

```
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.data',header=None, s
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.ix[:,0:4].values
y = df.ix[:,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()
```