13. Assignment on Dimensionality Reduction using PCA.

In [2]:
```
import pandas as pd

# load dataset into Pandas DataFrame
df = pd.read_csv("iris.csv")
df.head()
```

Out[2]:

	Id	SepalLengthCm	SepalWidthCm	PetalLengthCm	PetalWidthCm	Species
0	1	5.1	3.5	1.4	0.2	Iris-setosa
1	2	4.9	3.0	1.4	0.2	Iris-setosa
2	3	4.7	3.2	1.3	0.2	Iris-setosa
3	4	4.6	3.1	1.5	0.2	Iris-setosa
4	5	5.0	3.6	1.4	0.2	Iris-setosa

In [11]:
```
from sklearn.preprocessing import StandardScaler
features = ['SepalLengthCm','SepalWidthCm','PetalLengthCm','PetalWidthCm']
# Separating out the features
x = df.loc[:,features].values
# Separating out the target
y = df.loc[:,['Species']].values
# Standardizing the features
x = StandardScaler().fit_transform(x)
from sklearn.model_selection import train_test_split
X_train, X_test, y_train, y_test = train_test_split(x, y, test_size=0.3, random_state
```

In [12]:
```
from sklearn.decomposition import PCA
pca = PCA(n_components=3)
principalComponents = pca.fit_transform(x)
principalDf = pd.DataFrame(data = principalComponents
             , columns = ['principal component 1', 'principal component 2', 'principa
```

In [14]:
```
finalDf = pd.concat([principalDf, df[['Species']]], axis = 1)
```

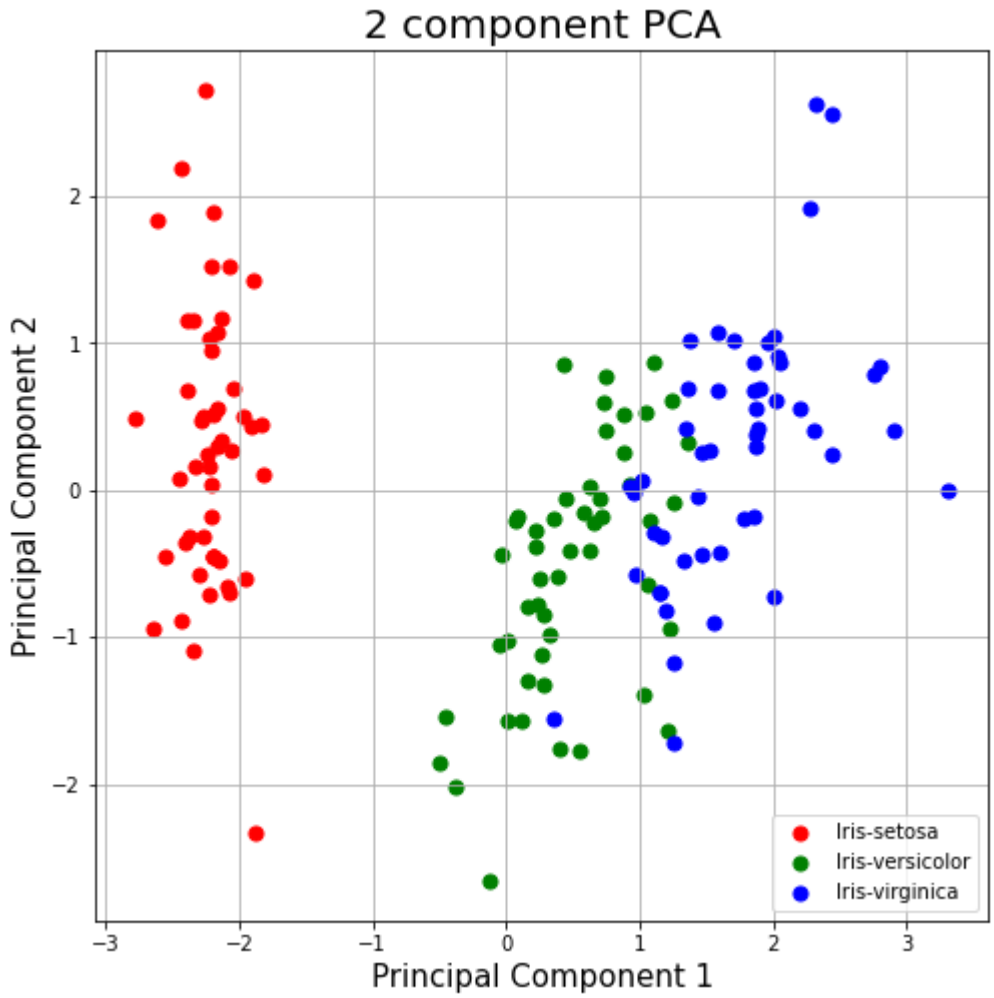
In [15]:
```
finalDf.head()
```

Out[15]:

	principal component 1	principal component 2	principal component 3	Species
0	-2.264542	0.505704	-0.121943	Iris-setosa
1	-2.086426	-0.655405	-0.227251	Iris-setosa
2	-2.367950	-0.318477	0.051480	Iris-setosa
3	-2.304197	-0.575368	0.098860	Iris-setosa
4	-2.388777	0.674767	0.021428	Iris-setosa

In [17]:
```
import matplotlib.pyplot as plt
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['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 [18]:
```
pca.explained_variance_ratio_
```

Out[18]:
```
array([0.72770452, 0.23030523, 0.03683832])
```

In [19]:
```
from sklearn.ensemble import RandomForestClassifier
```

In [20]:
```
#using original data
model = RandomForestClassifier()
model.fit(X_train, y_train)
predictions = model.predict(X_test)
```

```
C:\Users\danes\AppData\Local\Temp\ipykernel_7964\1694027823.py:3: DataConversionWarni
ng: A column-vector y was passed when a 1d array was expected. Please change the shap
e of y to (n_samples,), for example using ravel().
  model.fit(X_train, y_train)
```

In [21]:
```
predictions
```

Out[21]:
```
array(['Iris-setosa', 'Iris-setosa', 'Iris-setosa', 'Iris-virginica',
       'Iris-versicolor', 'Iris-virginica', 'Iris-versicolor',
       'Iris-versicolor', 'Iris-virginica', 'Iris-setosa',
       'Iris-virginica', 'Iris-setosa', 'Iris-setosa', 'Iris-virginica',
       'Iris-virginica', 'Iris-versicolor', 'Iris-versicolor',
       'Iris-versicolor', 'Iris-setosa', 'Iris-versicolor',
       'Iris-versicolor', 'Iris-setosa', 'Iris-versicolor',
       'Iris-versicolor', 'Iris-versicolor', 'Iris-versicolor',
       'Iris-versicolor', 'Iris-virginica', 'Iris-setosa', 'Iris-setosa',
       'Iris-virginica', 'Iris-versicolor', 'Iris-virginica',
       'Iris-versicolor', 'Iris-virginica', 'Iris-versicolor',
       'Iris-versicolor', 'Iris-versicolor', 'Iris-versicolor',
```

```
       'Iris-virginica', 'Iris-setosa', 'Iris-setosa', 'Iris-setosa',
       'Iris-virginica', 'Iris-versicolor'], dtype=object)
```

In [22]:
```
#Display the accuracy score
from sklearn.metrics import accuracy_score
accuracy_score(y_test, predictions)
```

Out[22]:
```
0.9555555555555556
```

In [24]:
```
# Separating out the features
x = finalDf.drop(["Species"], axis = 1)
x = StandardScaler().fit_transform(x)
from sklearn.model_selection import train_test_split
X_train, X_test, y_train, y_test = train_test_split(x, y, test_size=0.3, random_state
```

In [25]:
```
predictions
```

Out[25]:
```
array(['Iris-setosa', 'Iris-setosa', 'Iris-setosa', 'Iris-virginica',
       'Iris-versicolor', 'Iris-virginica', 'Iris-versicolor',
       'Iris-versicolor', 'Iris-virginica', 'Iris-setosa',
       'Iris-virginica', 'Iris-setosa', 'Iris-setosa', 'Iris-virginica',
       'Iris-virginica', 'Iris-versicolor', 'Iris-versicolor',
       'Iris-versicolor', 'Iris-setosa', 'Iris-versicolor',
       'Iris-versicolor', 'Iris-setosa', 'Iris-versicolor',
       'Iris-versicolor', 'Iris-versicolor', 'Iris-versicolor',
       'Iris-versicolor', 'Iris-virginica', 'Iris-setosa', 'Iris-setosa',
       'Iris-virginica', 'Iris-versicolor', 'Iris-virginica',
       'Iris-versicolor', 'Iris-virginica', 'Iris-versicolor',
       'Iris-versicolor', 'Iris-versicolor', 'Iris-versicolor',
       'Iris-virginica', 'Iris-setosa', 'Iris-setosa', 'Iris-setosa',
       'Iris-virginica', 'Iris-versicolor'], dtype=object)
```

In [26]:
```
#Accuracy score of the altered dataset
accuracy_score(y_test, predictions)
```

Out[26]:
```
0.9555555555555556
```

In [ ]:
```
```