Principal Component Analysis, PCA

The PCA method allows us to analyze complex data having many dimensions, such as marketing data, and to find the principal drivers of outcomes.
This example takes a four dimensional model using test data from the UCI Machine Learning database and projects it onto two dimensions. PCA also shows how much these new dimensions contribute to the outcome. If below 85 % then too much information is being lost.

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
from sklearn.decomposition import PCA
from sklearn.preprocessing import StandardScaler

url="https://archive.ics.uci.edu/ml/machine-learning-databases/iris/iris.data"
df=pd.read_csv(url,names=['sepal length','sepal width','petal length','petal width','target'])
#define the independent variables - the features
features = ['sepal length','sepal width','petal length','petal width']
x=df.loc[:,features].values
y=df.loc[:,'target'].values #dependent variable - the question
x=StandardScaler().fit_transform(x)
print(pd.DataFrame(data=x,columns=features).head())
pca = PCA(n_components=2)
principalComponents=pca.fit_transform(x)
principalDf = pd.DataFrame(data=principalComponents,columns=['Principal Comp 1','Principal Comp 2
'])
finalDf = pd.concat([principalDf,df[['target']]],axis=1)
print('Contribution of Principal Comps 1 and 2 to model ', pca.explained_variance_ratio_)
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)
```

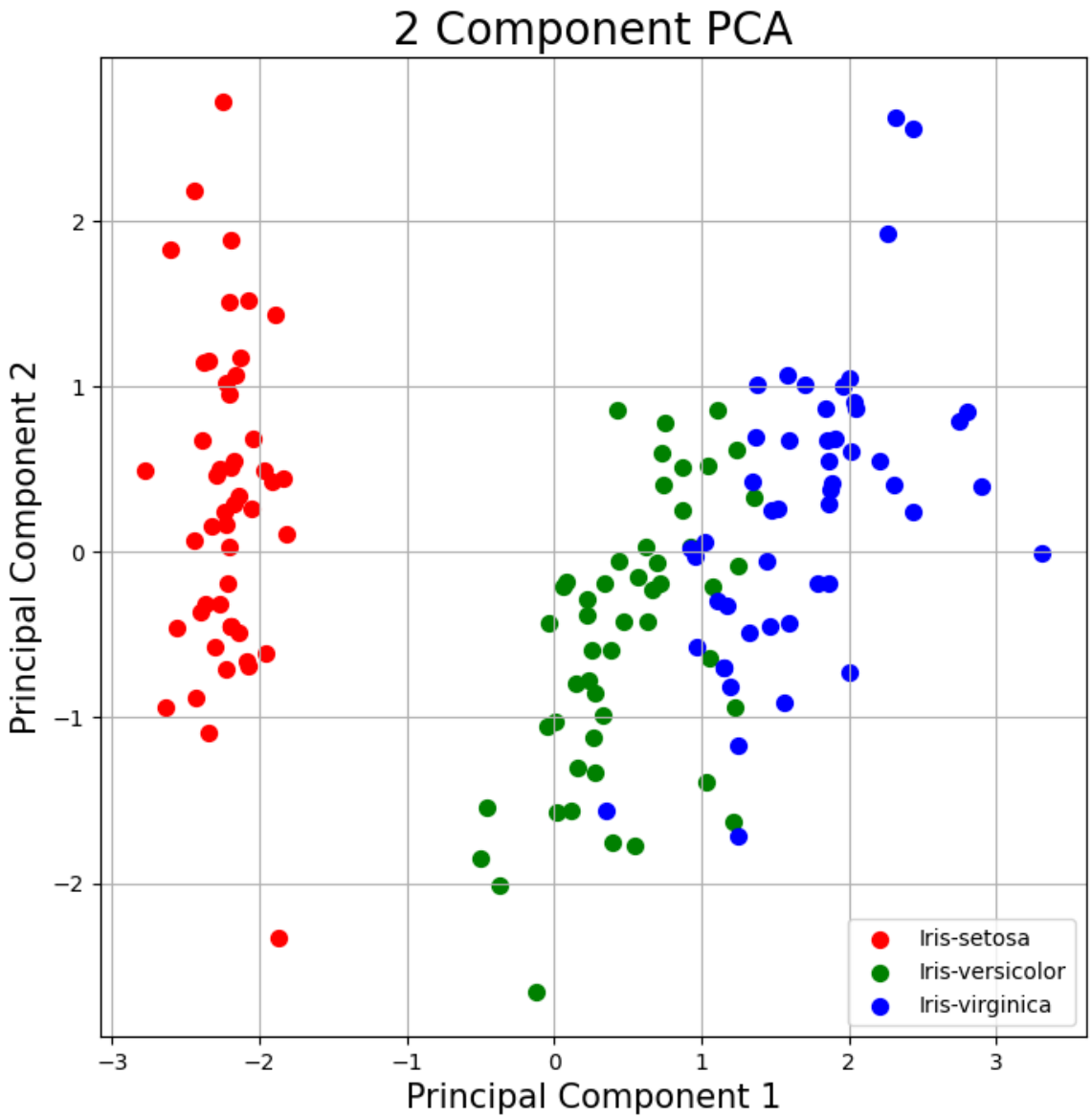
```
ax.set_title('2 Component PCA',fontsize=20)
targets = ['Iris-setosa','Iris-versicolor','Iris-virginica']
colors= ['r','g','b']
for target,color in zip(targets,colors):
    indicesToKeep = finalDf['target'] == target
    ax.scatter(finalDf.loc[indicesToKeep,'Principal Comp 1']
               ,finalDf.loc[indicesToKeep,'Principal Comp 2']
               ,c=color
               ,s=50)

ax.legend(targets)
ax.grid()
plt.show()
```

How much information do these two principal components contribute? 96% so very little information is lost by doing this.

Contribution of Principal Comps 1 and 2 to model [0.72770452 0.23030523]

Figure 1



x=−0.80 y=1.829