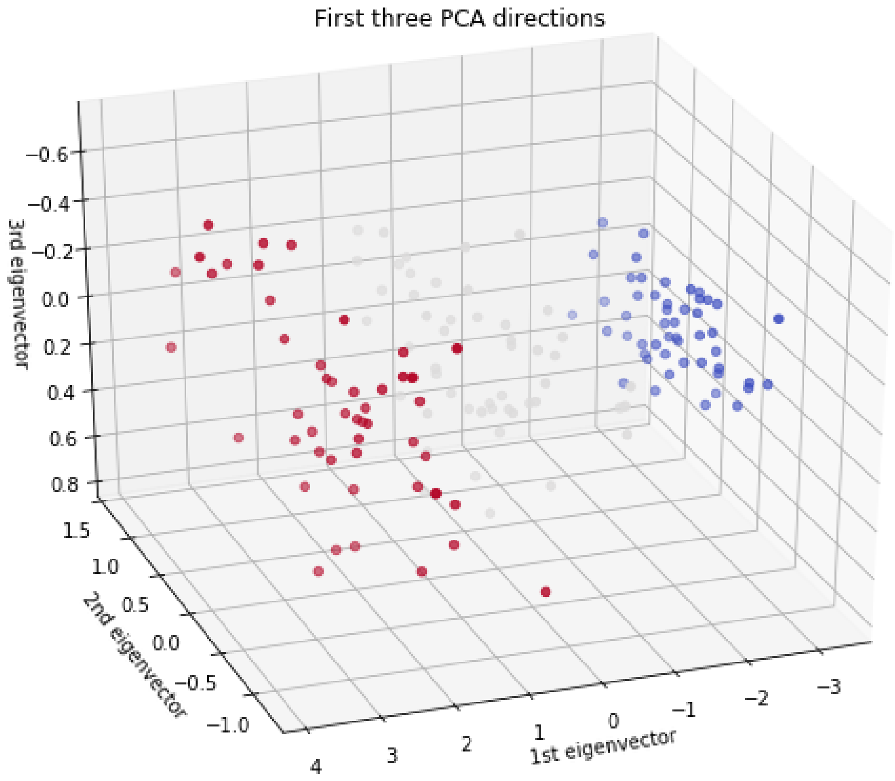
ML_10 Assignment PCA and PLOT

April 9, 2019

```
In [1]: """In this assignment students have to transform iris data into 3 dimensions and plot
        chart with transformed dimensions and color each data point with specific class.
        Hint:
        import numpy as np
        import matplotlib.pyplot as plt
        from mpl_toolkits.mplot3d import Axes3D
        from sklearn import decomposition
        from sklearn import datasets"""


        # Importing important libraries

        import numpy as np
        import pandas as pd
        import matplotlib.pyplot as plt
        %matplotlib inline

In [2]: # Load the dataset from sklearn

        from sklearn import datasets

        iris = datasets.load_iris()
        iris.data.shape

Out[2]: (150, 4)

In [4]: #Applying PCA to reduce the dimension to 3
        #and plotting a 3D graph


        X = iris.data
        y = iris.target

        from sklearn.decomposition import PCA
        from mpl_toolkits.mplot3d import Axes3D

        fig = plt.figure(1, figsize=(8, 6))
        ax = Axes3D(fig, elev=-150, azim=110)
```

```
X_reduced = PCA(n_components=3).fit_transform(iris.data)

ax.scatter(X_reduced[:, 0], X_reduced[:, 1], X_reduced[:, 2], c=y,
           cmap=plt.cm.coolwarm)

ax.set_title("First three PCA directions")
ax.set_xlabel("1st eigenvector")
ax.set_ylabel("2nd eigenvector")
ax.set_zlabel("3rd eigenvector")
```

```
Out[4]: Text(0.5, 0, '3rd eigenvector')
```



```
In [5]: # Reducing the data into 3 dimensions using PCA

        from sklearn.decomposition import PCA

        pca = PCA(n_components = 3)
        principalComponents = pca.fit_transform(X)
        newDf=pd.DataFrame(data=principalComponents,columns=['principal component 1','principa
        newDf.head()
```

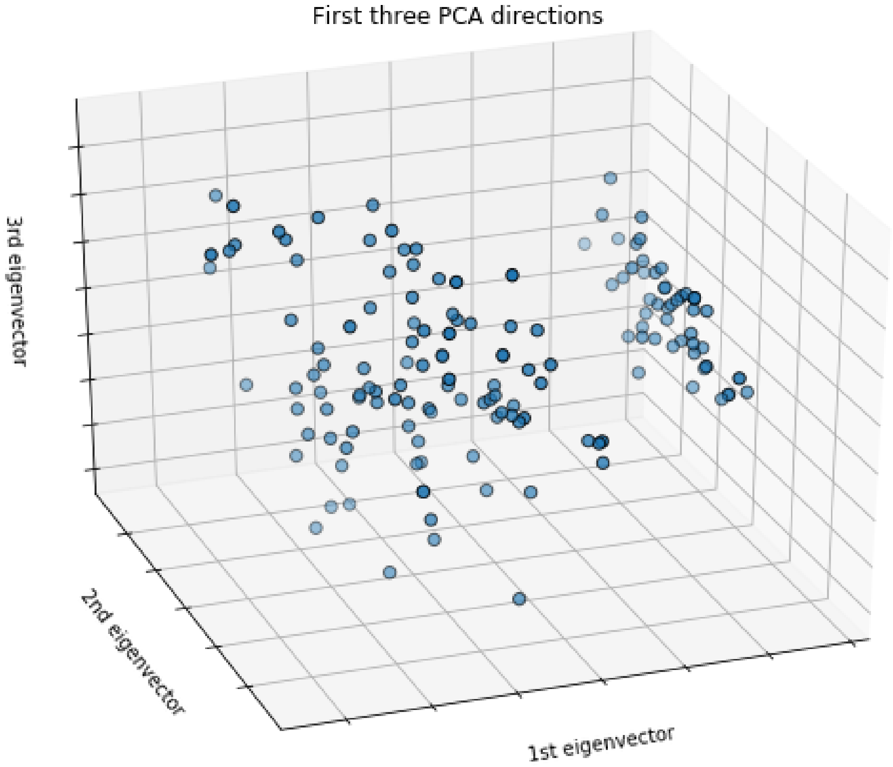
```
Out[5]:    principal component 1  principal component 2  principal component 3
        0              -2.264542               0.505704              -0.121943
        1              -2.086426              -0.655405              -0.227251
        2              -2.367950              -0.318477               0.051480
        3              -2.304197              -0.575368               0.098860
        4              -2.388777               0.674767               0.021428
```

```
In [6]: # Concatinating the principal components with target column

        finaldf = pd.concat([newDf,df[['target']]],axis = 1)
        finaldf.head()
```

```
Out[6]:    principal component 1  principal component 2  principal component 3  \
        0              -2.264542               0.505704              -0.121943
        1              -2.086426              -0.655405              -0.227251
        2              -2.367950              -0.318477               0.051480
        3              -2.304197              -0.575368               0.098860
        4              -2.388777               0.674767               0.021428

                target
        0  Iris-setosa
        1  Iris-setosa
        2  Iris-setosa
        3  Iris-setosa
        4  Iris-setosa
```

```
In [ ]:
```

```
In [8]: from mpl_toolkits.mplot3d import Axes3D

        fig = plt.figure(1, figsize=(8, 6))
        ax = Axes3D(fig, elev=-150, azim=110)
        ax.scatter(principalComponents[:, 0], principalComponents[:, 1], principalComponents[:
        ax.set_title("First three PCA directions")
        ax.set_xlabel("1st eigenvector")
        ax.w_xaxis.set_ticklabels([])
        ax.set_ylabel("2nd eigenvector")
        ax.w_yaxis.set_ticklabels([])
        ax.set_zlabel("3rd eigenvector")
        ax.w_zaxis.set_ticklabels([])
        plt.show()
```


First three PCA directions
3rd eigenvector
2nd eigenvector
1st eigenvector