In this assignment students have to transform iris data into 3 dimensions and plot a 3d 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 NOTE: The solution shared through Github should contain the source code used and the screenshot of the output.

from sklearn import datasets

import numpy as np

import matplotlib.pyplot as plt

from mpl\_toolkits.mplot3d import Axes3D

from sklearn import decomposition

iris = datasets.load\_iris()

#print(iris.DESCR)

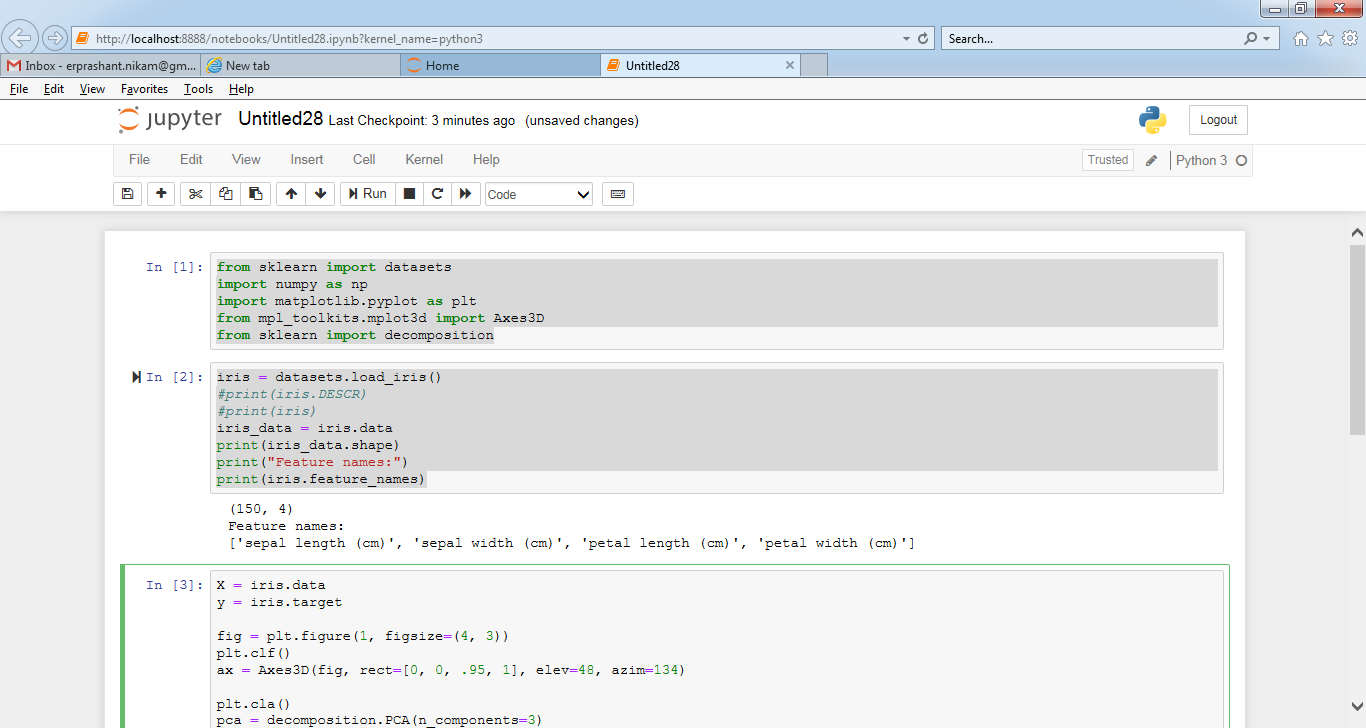
#print(iris)

iris\_data = iris.data

print(iris\_data.shape)

print("Feature names:")

print(iris.feature\_names)



X = iris.data

y = iris.target

fig = plt.figure(1, figsize=(4, 3))

plt.clf()

ax = Axes3D(fig, rect=[0, 0, .95, 1], elev=48, azim=134)

plt.cla()

pca = decomposition.PCA(n\_components=3)

pca.fit(X)

X = pca.transform(X)

for name, label in [('Setosa', 0), ('Versicolour', 1), ('Virginica', 2)]:

ax.text3D(X[y == label, 0].mean(),

X[y == label, 1].mean() + 1.5,

X[y == label, 2].mean(), name,

horizontalalignment='center',

bbox=dict(alpha=.5, edgecolor='w', facecolor='w'))

# Reorder the labels to have colors matching the cluster results

y = np.choose(y, [1, 2, 0]).astype(np.float)

ax.scatter(X[:, 0], X[:, 1], X[:, 2], c=y, cmap=plt.cm.nipy\_spectral,

edgecolor='k')

ax.w\_xaxis.set\_ticklabels([])

ax.w\_yaxis.set\_ticklabels([])

ax.w\_zaxis.set\_ticklabels([])

plt.show()

