

```

import pandas as pd
import matplotlib.pyplot as plt
from sklearn.decomposition import PCA

# code from https://github.com/mGalarnyk/Python_Tutorials/blob/master/Sklearn/PCA/PCA_Data_Visualization_Iris_Dataset_Blog.ipynb

df = pd.read_csv('data_authors.csv') # read csv file

# list of features which will be used to create the PCA model
features = ['av_sentences', 'av_voc', 'lex_richness', 'av_punct', 'av_num_word_ns',
            'av_num_words_sentences', 'av_num_words_sentences_ns', 'av_num_ns_sentences',
            'av_words_length', 'av_ego', 'av_modal', 'emphasis', 'comparison', 'contrast',
            'addition',
            'illustration', 'comma', 'dotcomma', 'excl', 'qmark', 'dash', 'slash', 'twodot',
            'CC', 'CD', 'DT', 'EX',
            'FW', 'IN', 'JJ', 'JJR', 'JJS', 'LS', 'MD', 'NN', 'NNS', 'NNP', 'NNPS', 'PDT',
            'POS', 'PRP', 'PRP$', 'RB', 'RBR', 'RBS', 'RP', 'TO', 'UH', 'VB', 'VBD', 'VBG',
            'VBN', 'VBP', 'VBZ',
            'WDT', 'WP', 'WP$', 'WRB']

# extract features
x = df.loc[:, features].values
y = df.loc[:, ['Authors']].values

# start creating the PCA model with 2 components
pca = PCA(n_components=2)
principalComponents = pca.fit_transform(x)
principalDf = pd.DataFrame(data=principalComponents
                           , columns=['principal component 1', 'principal component 2'])
finalDf = pd.concat([principalDf, df[['Authors']]], axis=1)

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('PCA : Authors profiles', fontsize=20)

# authors to take into account
targets = ['Bacon', 'Bentham', 'Berkeley', 'Burke', 'Cavendish', 'Clifford', 'Dewey',
            'Emerson', 'Fellerton', 'Godwin', 'Goldman', 'Hide', 'Hobbes',
            'Hume', 'James', 'Jordan', 'Martineau', 'Mill',
            'Russell', 'Santayana', 'Sidgwick', 'Spencer', 'Thoreau',
            'Wollstonecraft']

colors = ['#C0C0C0', '#808080', '#5D6D7E', '#FF0000', '#F41F53', '#F08080', '#800000', '#F39C12',
            '#E67E22', '#D35400', '#FFFF00', '#FFFE91', '#808000', '#00FF00', '#70E770',
            '#008000', '#00B9B9', '#00FFFF', '#00D1A0', '#008080', '#0000FF', '#000080', '#8C8CEC',
            '#FF00FF']

# create the graph
for target, color in zip(targets, colors):
    indicesToKeep = finalDf['Authors'] == target
    ax.scatter(finalDf.loc[indicesToKeep, 'principal component 1'],
               finalDf.loc[indicesToKeep, 'principal component 2'],
               c=color,
               s=50)
    ax.annotate(target, (finalDf.loc[indicesToKeep, 'principal component 1'], finalDf.loc[indicesToKeep, 'principal component 2'])) # to annotate each point with the name of the author

ax.grid()
plt.show()

```