CS 498 AML HW3

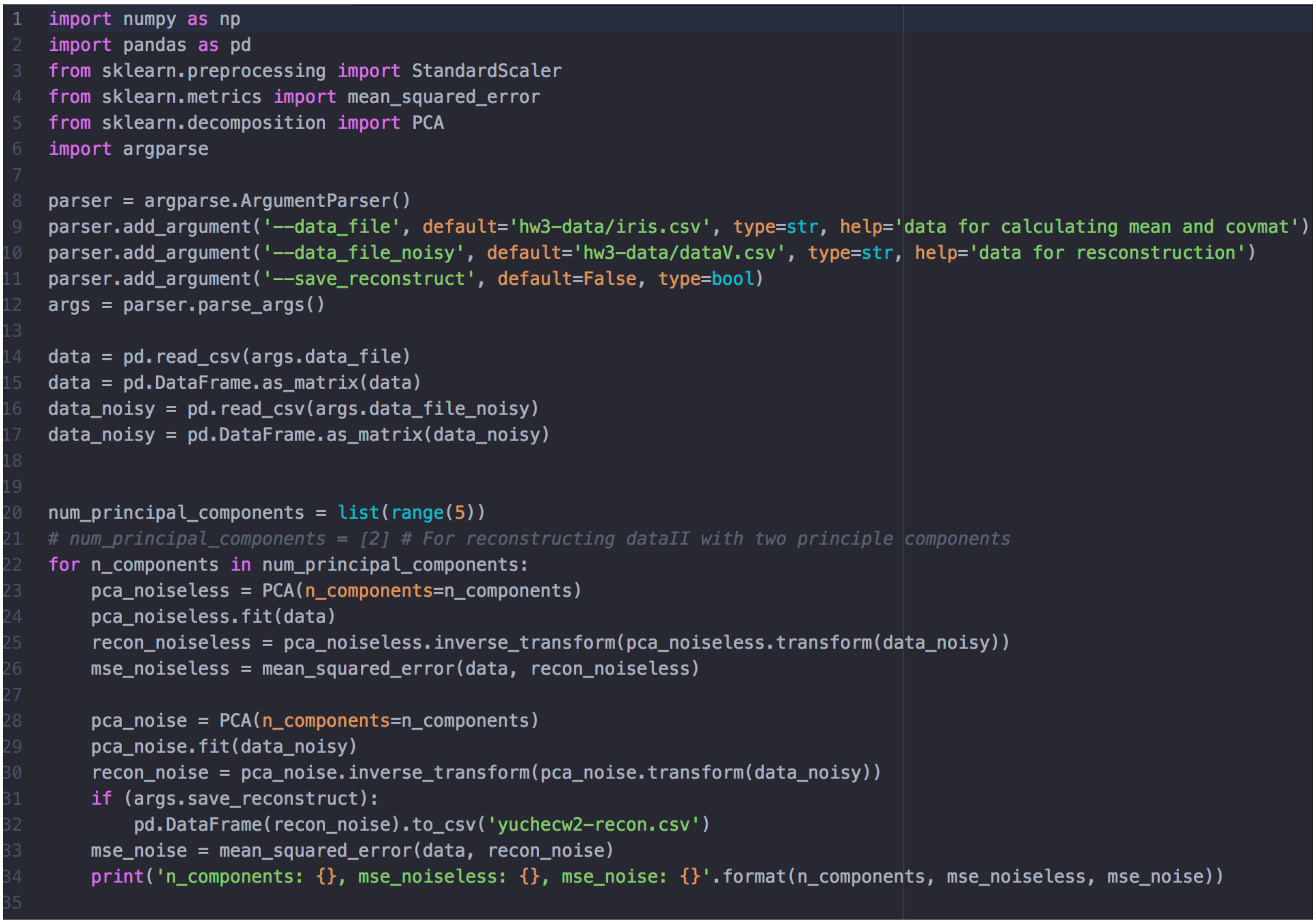
Yu Che Wang/ yuchecw2

Table of 50 MSE numbers

Number of PCs	0N	1N	2N	3N	4N	0c	1c	2c	3c	4c
Dataset I	1.13561	0.0958	0.0438	0.03544	0.0402	1.13577	0.09615	0.04445	0.03611	0.0402
Dataset II	1.13561	0.16027	0.1789	0.22709	0.27891	1.13738	0.1621	0.1876	0.2354	0.2789
Dataset III	1.13561	0.32259	0.49181	0.66271	0.9133	1.13936	0.33086	0.5299	0.75684	0.9134
Dataset IV	1.13561	0.1999	0.20702	0.24623	0.2985	1.1415	0.21015	0.3017	0.31779	0.299
Dataset V	1.13561	0.47944	0.83293	1.13706	1.2848	1.22998	0.70891	1.16285	1.24281	1.28481

Code:

Main function



```
import numpy as np
import pandas as pd
from sklearn.preprocessing import StandardScaler
from sklearn.metrics import mean_squared_error
from sklearn.decomposition import PCA
import argparse

parser = argparse.ArgumentParser()
parser.add_argument('--data_file', default='hw3-data/iris.csv', type=str, help='data for calculating mean and covmat')
parser.add_argument('--data_file_noisy', default='hw3-data/dataV.csv', type=str, help='data for resconstruction')
parser.add_argument('--save_reconstruct', default=False, type=bool)
args = parser.parse_args()

data = pd.read_csv(args.data_file)
data = pd.DataFrame.as_matrix(data)
data_noisy = pd.read_csv(args.data_file_noisy)
data_noisy = pd.DataFrame.as_matrix(data_noisy)


num_principal_components = list(range(5))
# num_principal_components = [2] # For reconstructing dataII with two principle components
for n_components in num_principal_components:
    pca_noiseless = PCA(n_components=n_components)
    pca_noiseless.fit(data)
    recon_noiseless = pca_noiseless.inverse_transform(pca_noiseless.transform(data_noisy))
    mse_noiseless = mean_squared_error(data, recon_noiseless)

    pca_noise = PCA(n_components=n_components)
    pca_noise.fit(data_noisy)
    recon_noise = pca_noise.inverse_transform(pca_noise.transform(data_noisy))
    if (args.save_reconstruct):
        pd.DataFrame(recon_noise).to_csv('yuchecw2-recon.csv')
    mse_noise = mean_squared_error(data, recon_noise)
    print('n_components: {}, mse_noiseless: {}, mse_noise: {}'.format(n_components, mse_noiseless, mse_noise))

```