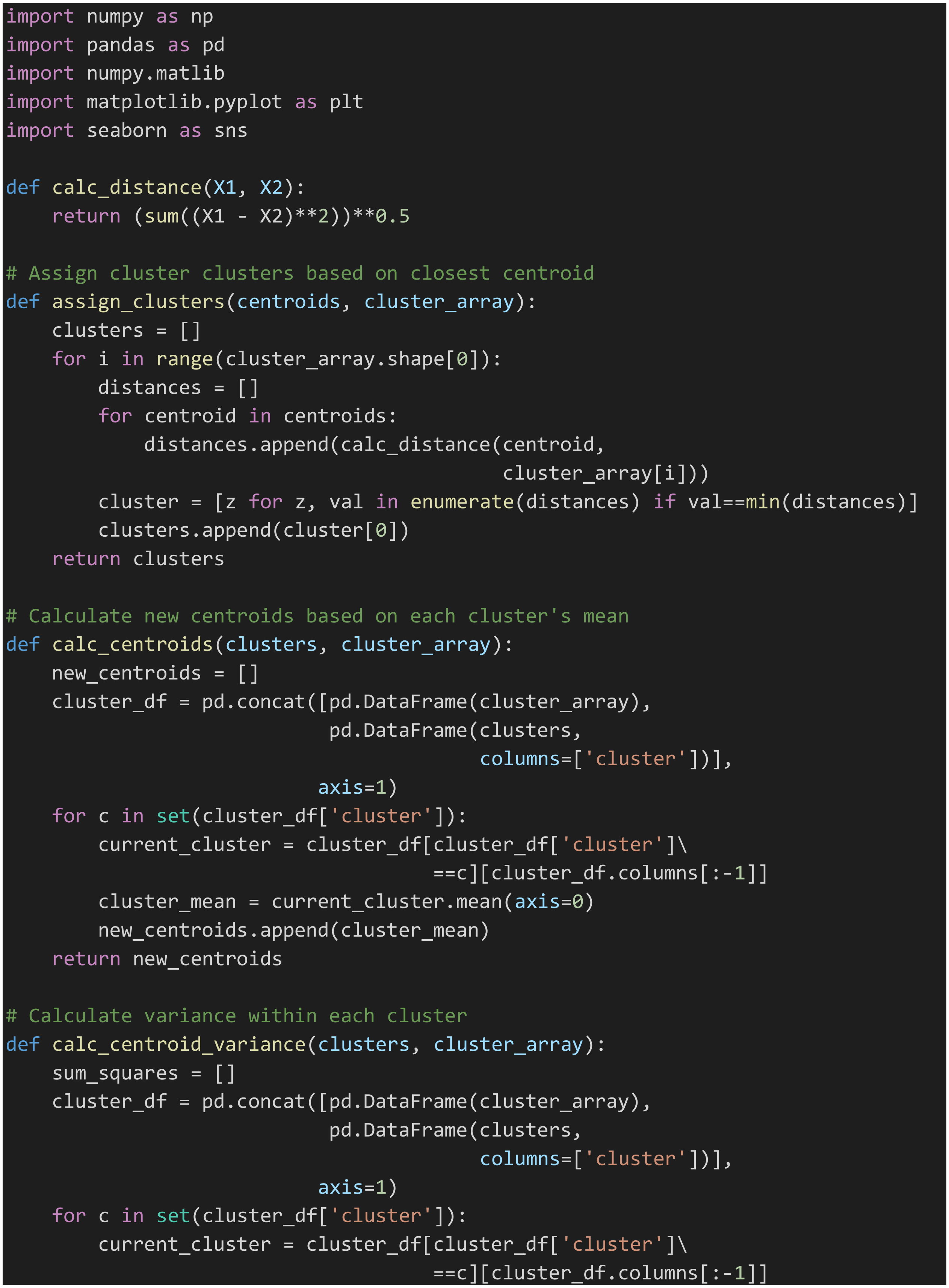
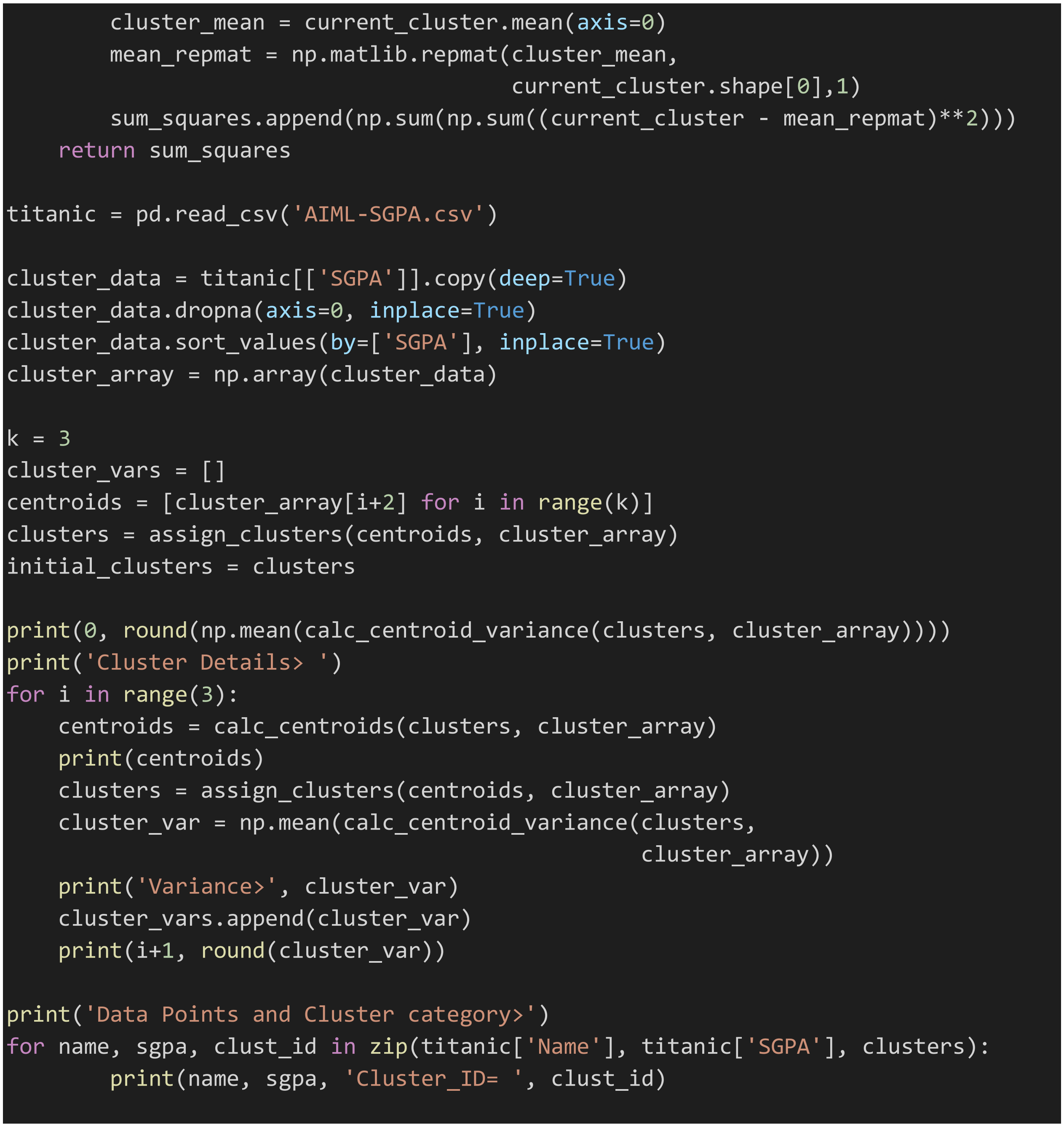
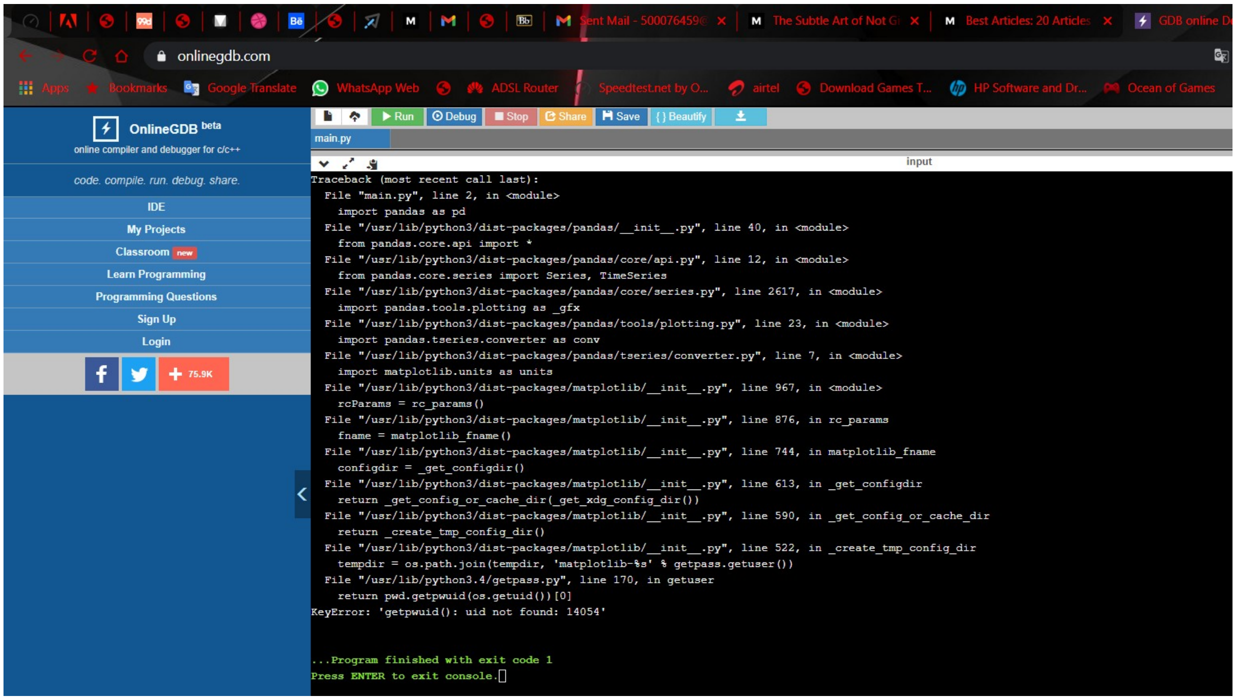


```python
import numpy as np
import pandas as pd
import numpy.matlib
import matplotlib.pyplot as plt
import seaborn as sns

def calc_distance(X1, X2):
    return (sum((X1 - X2)**2))**0.5

# Assign cluster clusters based on closest centroid
def assign_clusters(centroids, cluster_array):
    clusters = []
    for i in range(cluster_array.shape[0]):
        distances = []
        for centroid in centroids:
            distances.append(calc_distance(centroid,
                                           cluster_array[i]))
        cluster = [z for z, val in enumerate(distances) if val==min(distances)]
        clusters.append(cluster[0])
    return clusters

# Calculate new centroids based on each cluster's mean
def calc_centroids(clusters, cluster_array):
    new_centroids = []
    cluster_df = pd.concat([pd.DataFrame(cluster_array),
                            pd.DataFrame(clusters,
                                         columns=['cluster'])],
                           axis=1)
    for c in set(cluster_df['cluster']):
        current_cluster = cluster_df[cluster_df['cluster']\
                                     ==c][cluster_df.columns[:-1]]
        cluster_mean = current_cluster.mean(axis=0)
        new_centroids.append(cluster_mean)
    return new_centroids

# Calculate variance within each cluster
def calc_centroid_variance(clusters, cluster_array):
    sum_squares = []
    cluster_df = pd.concat([pd.DataFrame(cluster_array),
                            pd.DataFrame(clusters,
                                         columns=['cluster'])],
                           axis=1)
    for c in set(cluster_df['cluster']):
        current_cluster = cluster_df[cluster_df['cluster']\
                                     ==c][cluster_df.columns[:-1]]
```



```
        cluster_mean = current_cluster.mean(axis=0)
        mean_repmat = np.matlib.repmat(cluster_mean,
                                       current_cluster.shape[0],1)
        sum_squares.append(np.sum(np.sum((current_cluster - mean_repmat)**2)))
    return sum_squares

titanic = pd.read_csv('AIML-SGPA.csv')

cluster_data = titanic[['SGPA']].copy(deep=True)
cluster_data.dropna(axis=0, inplace=True)
cluster_data.sort_values(by=['SGPA'], inplace=True)
cluster_array = np.array(cluster_data)

k = 3
cluster_vars = []
centroids = [cluster_array[i+2] for i in range(k)]
clusters = assign_clusters(centroids, cluster_array)
initial_clusters = clusters

print(0, round(np.mean(calc_centroid_variance(clusters, cluster_array))))
print('Cluster Details> ')
for i in range(3):
    centroids = calc_centroids(clusters, cluster_array)
    print(centroids)
    clusters = assign_clusters(centroids, cluster_array)
    cluster_var = np.mean(calc_centroid_variance(clusters,
                                                 cluster_array))
    print('Variance>', cluster_var)
    cluster_vars.append(cluster_var)
    print(i+1, round(cluster_var))

print('Data Points and Cluster category>')
for name, sgpa, clust_id in zip(titanic['Name'], titanic['SGPA'], clusters):
        print(name, sgpa, 'Cluster_ID= ', clust_id)
```



```
Traceback (most recent call last):
  File "main.py", line 2, in <module>
    import pandas as pd
  File "/usr/lib/python3/dist-packages/pandas/__init__.py", line 40, in <module>
    from pandas.core.api import *
  File "/usr/lib/python3/dist-packages/pandas/core/api.py", line 12, in <module>
    from pandas.core.series import Series, TimeSeries
  File "/usr/lib/python3/dist-packages/pandas/core/series.py", line 2617, in <module>
    import pandas.tools.plotting as _gfx
  File "/usr/lib/python3/dist-packages/pandas/tools/plotting.py", line 23, in <module>
    import pandas.tseries.converter as conv
  File "/usr/lib/python3/dist-packages/pandas/tseries/converter.py", line 7, in <module>
    import matplotlib.units as units
  File "/usr/lib/python3/dist-packages/matplotlib/__init__.py", line 967, in <module>
    rcParams = rc_params()
  File "/usr/lib/python3/dist-packages/matplotlib/__init__.py", line 876, in rc_params
    fname = matplotlib_fname()
  File "/usr/lib/python3/dist-packages/matplotlib/__init__.py", line 744, in matplotlib_fname
    configdir = _get_configdir()
  File "/usr/lib/python3/dist-packages/matplotlib/__init__.py", line 613, in _get_configdir
    return _get_config_or_cache_dir(_get_xdg_config_dir())
  File "/usr/lib/python3/dist-packages/matplotlib/__init__.py", line 590, in _get_config_or_cache_dir
    return _create_tmp_config_dir()
  File "/usr/lib/python3/dist-packages/matplotlib/__init__.py", line 522, in _create_tmp_config_dir
    tempdir = os.path.join(tempdir, 'matplotlib-%s' % getpass.getuser())
  File "/usr/lib/python3.4/getpass.py", line 170, in getuser
    return pwd.getpwuid(os.getuid())[0]
KeyError: 'getpwuid(): uid not found: 14054'


...Program finished with exit code 1
Press ENTER to exit console.
```