```
Timer unit: 1e-06 s

Total time: 157.043 s
File: /tmp/ipykernel_24079/899591159.py
Function: fit_predict at line 18

Line #      Hits         Time  Per Hit   % Time  Line Contents
==============================================================
    18                                             def fit_predict(self, k_num:int = 3, max_step:int = 500, conv_threshold: float = 1e-5
    19                                               '''
    20                                               Membuat model KMeans dengan K tertentu. Akan mengkembalikan hasil prediksi cluster.
    21                                               Poin kluster akan disimpan pada variable point
    22                                               '''
    23                                               # Setting up cluster arry for every record
    24         5       2338.0    467.6      0.0      cluster = np.zeros(len(self.training_arr))
    25
    26                                               # normalize data
    27         5      19998.0   3999.6      0.0      data = self.__normalize_data__(self.training_arr)
    28
    29                                               # Initialize centroid using KMeans++
    30         5   34688788.0 6937757.6     22.1      point = self.__initialize_centroids__(data, k_num)
    31
    32                                               # Setup convergence and counter
    33         5         17.0      3.4      0.0      convergence = False
    34         5          5.0      1.0      0.0      step = 0
    35
    36        23         34.0      1.5      0.0      while not convergence and (step < max_step):
    37        18         42.0      2.3      0.0        initial_point = point
    38        18   96640393.0 5368910.7     61.5        distance = self.__calculate_distance__(data, point)
    39        18    8219250.0 456625.0      5.2        cluster = self.__clustering__(distance)
    40        18    7495574.0 416420.8      4.8        new_point = self.__point_nomralization__(data, point, cluster)
    41        18       2921.0    162.3      0.0        convergence = self.__convergence_check__(initial_point, new_point, conv_threshold
    42
    43        18         35.0      1.9      0.0        if convergence:
    44         1          1.0      1.0      0.0          point = new_point
    45         1        129.0    129.0      0.0          print("It's convergence!")
    46                                                 else:
    47        17         14.0      0.8      0.0          point = new_point
    48        17         19.0      1.1      0.0          step += 1
    49        17       3239.0    190.5      0.0          print("STEP:", step)
    50
    51
    52         5    9740132.0 1948026.4      6.2      self.inertia = self.__calculate_inertia__(data, cluster, point)
    53         5     230011.0  46002.2      0.1      self.point = self.__denormalize_point__(point, self.training_arr)
    54         5         11.0      2.2      0.0      return cluster
```