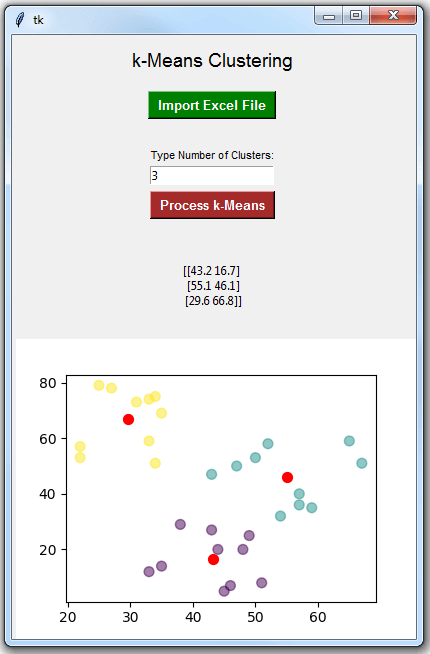
**Пример кластеризации K-средних в Python**

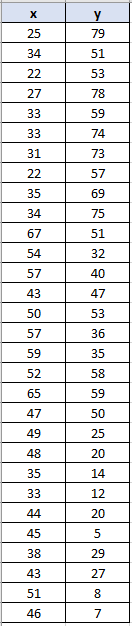
Кластеризация K-средних - это концепция, которая относится к неконтролируемому обучению. Этот алгоритм можно использовать для поиска групп в немаркированных данных. Чтобы продемонстрировать эту концепцию, я рассмотрю простой пример кластеризации K-средних в Python.

* Создание DataFrame для двумерного набора данных
* Нахождение центроидов для 3-х кластеров, а затем для 4-х кластеров
* Добавление графического пользовательского интерфейса (GUI) для отображения результатов



Пример кластеризации K-средних в Python

Для начала рассмотрим простой пример со следующим двумерным набором данных:



Затем вы можете [записать](https://datatofish.com/create-pandas-dataframe/) эти данные в Python с помощью [pandas DataFrame](https://datatofish.com/create-pandas-dataframe/) :

from pandas import DataFrame

Data = {'x': [25,34,22,27,33,33,31,22,35,34,67,54,57,43,50,57,59,52,65,47,49,48,35,33,44,45,38,43,51,46],

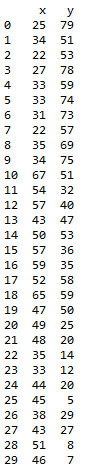
'y': [79,51,53,78,59,74,73,57,69,75,51,32,40,47,53,36,35,58,59,50,25,20,14,12,20,5,29,27,8,7]

}

df = DataFrame(Data,columns=['x','y'])

print (df)

Если вы запустите код на Python, вы получите следующий результат, который соответствует нашему набору данных:



Кластеризация K-средних в Python - 3 кластера

После того, как вы создали DataFrame на основе приведенных выше данных, вам необходимо импортировать 2 дополнительных модуля Python:

matplotlib - для [создания диаграмм на Python](https://datatofish.com/scatter-line-bar-charts-using-matplotlib/)

sklearn - для применения кластеризации K-средних в Python

from pandas import DataFrame

import matplotlib.pyplot as plt

from sklearn.cluster import KMeans

Data = {'x': [25,34,22,27,33,33,31,22,35,34,67,54,57,43,50,57,59,52,65,47,49,48,35,33,44,45,38,43,51,46],

'y': [79,51,53,78,59,74,73,57,69,75,51,32,40,47,53,36,35,58,59,50,25,20,14,12,20,5,29,27,8,7]

}

df = DataFrame(Data,columns=['x','y'])

kmeans = KMeans(n\_clusters=3).fit(df)

centroids = kmeans.cluster\_centers\_

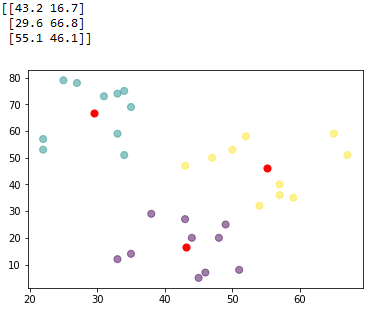
print(centroids)

plt.scatter(df['x'], df['y'], c= kmeans.labels\_.astype(float), s=50, alpha=0.5)

plt.scatter(centroids[:, 0], centroids[:, 1], c='red', s=50)

plt.show()

Запустите код на Python, и вы увидите 3 кластера с 3 отдельными центроидами:



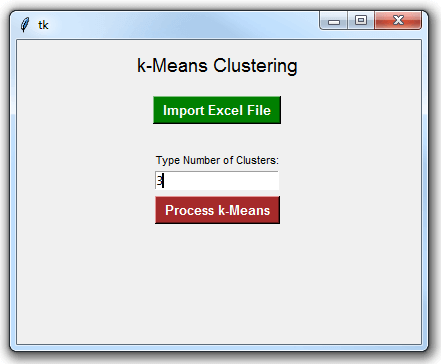
Обратите внимание, что центр каждого кластера (красный) представляет собой среднее значение всех наблюдений, принадлежащих этому кластеру.

Как вы также можете видеть, наблюдения, принадлежащие данному кластеру, находятся ближе к центру этого кластера по сравнению с центрами других кластеров.

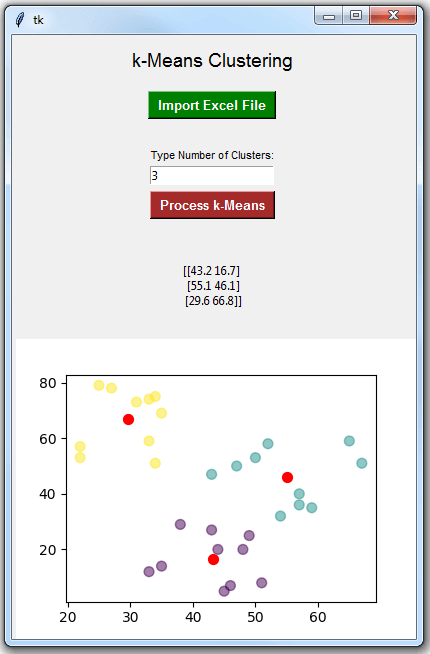
Графический интерфейс Tkinter

Нажмите зеленую кнопку, чтобы [импортировать файл Excel](https://datatofish.com/read_excel/) (откроется диалоговое окно, которое поможет вам найти и затем импортировать файл Excel).

После того, как вы импортировали файл Excel, введите количество кластеров в поле [ввода](https://datatofish.com/entry-box-tkinter/) , а затем нажмите красную кнопку, чтобы обработать k-средние. Например, я ввел 3 в поле ввода:



И вот результат, который у меня получился:



Полный код:

import tkinter as tk

from tkinter import filedialog

import pandas as pd

from pandas import DataFrame

import matplotlib.pyplot as plt

from sklearn.cluster import KMeans

from matplotlib.backends.backend\_tkagg import FigureCanvasTkAgg

root= tk.Tk()

canvas1 = tk.Canvas(root, width = 400, height = 300, relief = 'raised')

canvas1.pack()

label1 = tk.Label(root, text='k-Means Clustering')

label1.config(font=('helvetica', 14))

canvas1.create\_window(200, 25, window=label1)

label2 = tk.Label(root, text='Type Number of Clusters:')

label2.config(font=('helvetica', 8))

canvas1.create\_window(200, 120, window=label2)

entry1 = tk.Entry (root)

canvas1.create\_window(200, 140, window=entry1)

def getExcel ():

global df

import\_file\_path = filedialog.askopenfilename()

read\_file = pd.read\_excel (import\_file\_path)

df = DataFrame(read\_file,columns=['x','y'])

browseButtonExcel = tk.Button(text=" Import Excel File ", command=getExcel, bg='green', fg='white', font=('helvetica', 10, 'bold'))

canvas1.create\_window(200, 70, window=browseButtonExcel)

def getKMeans ():

global df

global numberOfClusters

numberOfClusters = int(entry1.get())

kmeans = KMeans(n\_clusters=numberOfClusters).fit(df)

centroids = kmeans.cluster\_centers\_

label3 = tk.Label(root, text= centroids)

canvas1.create\_window(200, 250, window=label3)

figure1 = plt.Figure(figsize=(4,3), dpi=100)

ax1 = figure1.add\_subplot(111)

ax1.scatter(df['x'], df['y'], c= kmeans.labels\_.astype(float), s=50, alpha=0.5)

ax1.scatter(centroids[:, 0], centroids[:, 1], c='red', s=50)

scatter1 = FigureCanvasTkAgg(figure1, root)

scatter1.get\_tk\_widget().pack(side=tk.RIGHT, fill=tk.BOTH)

processButton = tk.Button(text=' Process k-Means ', command=getKMeans, bg='brown', fg='white', font=('helvetica', 10, 'bold'))

canvas1.create\_window(200, 170, window=processButton)

root.mainloop()