|  |  |  |
| --- | --- | --- |
|  | |  |
| МИНОБРНАУКИ РОССИИ | |  |
| Федеральное государственное бюджетное образовательное учреждение  высшего образования  **"МИРЭА - Российский технологический университет"**  **РТУ МИРЭА** | |  |
|  | Институт информационных технологий (ИТ) | |
|  | Кафедра прикладной математики | |

|  |  |
| --- | --- |
| **Отчет по шестой практической работе.** | |
| **по дисциплине** | |
| **«Технологии и инструментарий анализа больших данных»** | |
|  | |
| Выполнил студент группы ИКБО- 22-20 | Никулин К.В. |
| Принял | Парамонов А.А. |

Москва 2023 г.

СОДЕРЖАНИЕ

[1. ВЫГРУЗКА ДАННЫХ 3](#_Toc151044784)

[2. ЛОГИЧЕСКАЯ РЕГРЕССИЯ 4](#_Toc151044785)

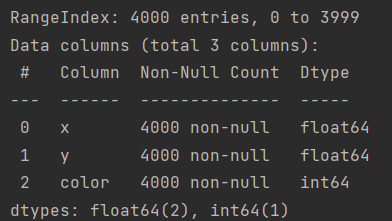
[3. МЕТОД ОПОРНЫХ ВЕКТОРОВ (SVM) 6](#_Toc151044786)

[4. KNN (K БЛИЖАЙШИХ СОСЕДЕЙ) 7](#_Toc151044787)

[ПРИЛОЖЕНИЕ 9](#_Toc151044789)

1. ДАННЫЕ ДЛЯ РАБОТЫ

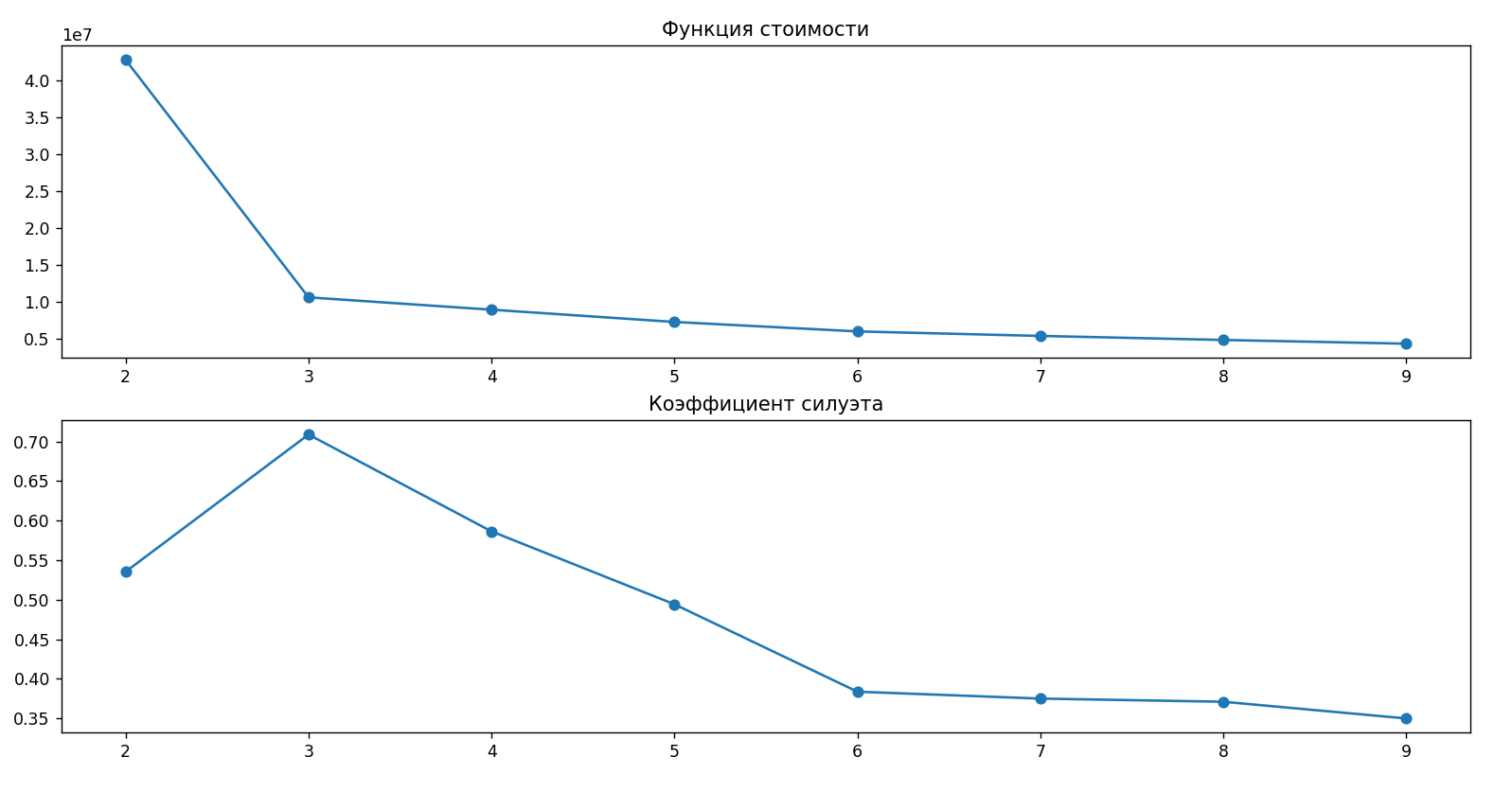
Датасет специально созданный для кластеризации данных, для 3 кластеров.



**Рисунок 1 – Данные датасета**

1. K-MEANS КЛАСТЕРИЗАЦИЯ

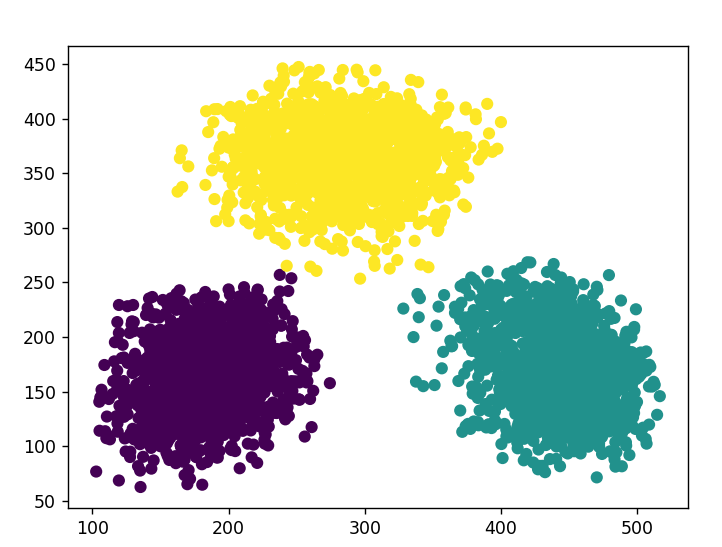
Выбор количества кластеров в K-means: построим график зависимости функции стоимости от числа кластеров для определения количества кластеров «правилом локтя».

****

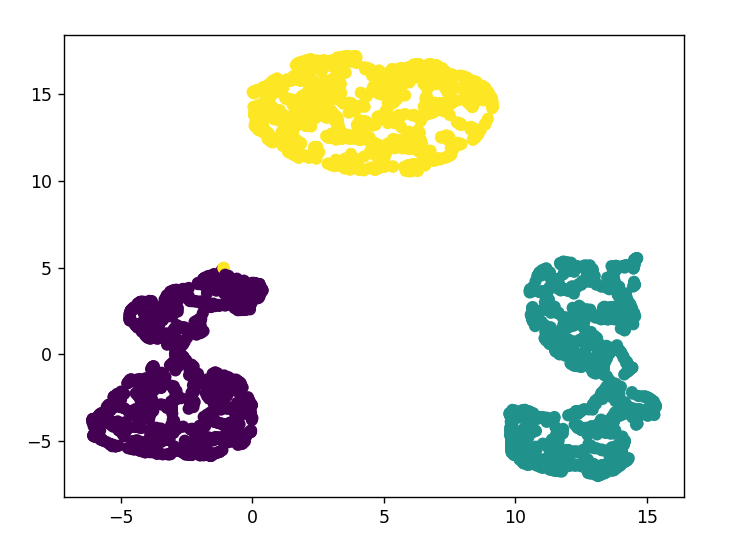
**Рисунок 2 – Значения функции стоимости и коэффициента силуэта**

«Правило локтя» не всегда позволяет понять какое количество кластеров взять, поэтому можем воспользоваться другим методом: «Коэффициент силуэта». Коэффициент силуэта демонстрирует пиковое значение, нежели плавный изгиб в методе локтя. Коэффициент силуэта рассчитывается с использованием среднего внутри кластерного расстояния a и среднего расстояния до ближайшего кластера b для каждой выборки.

Коэффициент силуэта достигает максимума при k=3, Результат кластеризации для 3 кластеров:

****

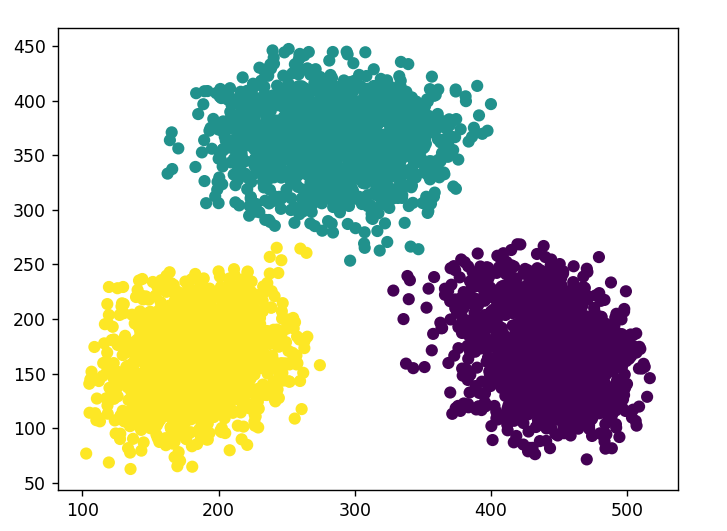
**Рисунок 3 – Визуализация k-means**

****

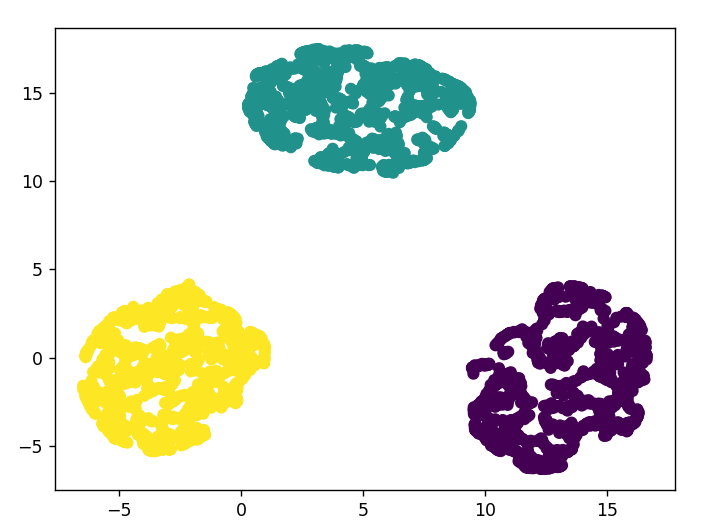
**Рисунок 4 – Визуализация k-means с помощью UMAP**

1. ИЕРАРХИЧЕСКАЯ АГЛОМЕРАТИВНАЯ КЛАСТЕРИЗАЦИЯ

В результате кластеризации получаем 3 кластера:



**Рисунок 5 – Иерархическая агломеративная кластеризация**

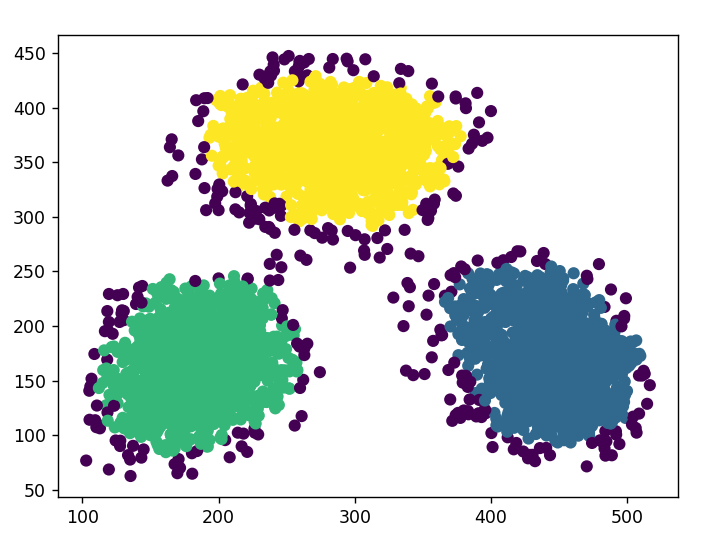
****

**Рисунок 6 – Визуализация иерархической агломеративной кластеризации с помощью UMAP**

1. DBSCAN КЛАСТЕРИЗАЦИЯ

Результат кластеризации DBSCAN с параметрами eps = 10 и min\_samples = 10,

Синим цветом (-1) выделены шумовые объекты.



**Рисунок 6 – DBSCAN кластеризация**

ПРИЛОЖЕНИЕ

import pandas as pd  
import numpy as np  
import matplotlib.pyplot as plt  
from sklearn.cluster import KMeans, DBSCAN  
from sklearn.metrics import silhouette\_score  
from sklearn.cluster import AgglomerativeClustering  
import umap  
  
  
def kmeans(df):  
 models = []  
 score1 = []  
 score2 = []  
 for i in range(2, 10):  
 model = KMeans(n\_clusters=i, random\_state=123, init='k-means++').fit(df)  
 models.append(model)  
 score1.append(model.inertia\_)  
 score2.append(silhouette\_score(df, model.labels\_))  
 fig, ax = plt.subplots(nrows=2, ncols=1)  
 ax[0].plot(np.arange(2, 10), score1, marker='o')  
 ax[1].plot(np.arange(2, 10), score2, marker='o')  
 ax[0].set\_title('Функция стоимости')  
 ax[1].set\_title('Коэффициент силуэта')  
 plt.show()  
  
 model1 = KMeans(n\_clusters=4, random\_state=123, init='k-means++')  
 model1.fit(df)  
 df['Claster'] = model1.labels\_  
 standard\_embedding = umap.UMAP(random\_state=42).fit\_transform(df)  
 plt.scatter(standard\_embedding[:, 0], standard\_embedding[:, 1], c=df['Claster'])  
 # plt.scatter(df['x'], df['y'], c=df['Claster'])  
 plt.show()  
 return  
  
  
def hac(df):  
 model2 = AgglomerativeClustering(3, compute\_distances=True)  
 clastering = model2.fit(df)  
 df['Claster'] = clastering.labels\_  
 standard\_embedding = umap.UMAP(random\_state=42).fit\_transform(df)  
 plt.scatter(standard\_embedding[:, 0], standard\_embedding[:, 1], c=df['Claster'])  
 # plt.scatter(df['x'], df['y'], c=df['Claster'])  
 plt.show()  
 return  
  
  
def dbscan(df):  
 model3 = DBSCAN(eps=10, min\_samples=10).fit(df)  
 df['Claster'] = model3.labels\_  
 # standard\_embedding = umap.UMAP(random\_state=42).fit\_transform(df)  
 # plt.scatter(standard\_embedding[:, 0], standard\_embedding[:, 1], c=df['Claster'])  
 plt.scatter(df['x'], df['y'], c=df['Claster'])  
 plt.show()  
 return  
  
  
def main():  
 df = pd.read\_csv("basic5.csv")  
 print(df.info())  
 kmeans(df)  
 # hac(df)  
 # dbscan(df)  
 return  
  
  
if \_\_name\_\_ == "\_\_main\_\_":  
 main()