Министерство цифрового развития, связи и
массовых коммуникаций Российской Федерации

Федеральное государственное бюджетное образовательное учреждение высшего образования «Сибирский государственный университет телекоммуникаций и информатики» (СибГУТИ)

Отчёт
по лабораторной работе №4
по дисциплине «**Прикладные задачи теории вероятностей**»

Выполнил:
студент гр. ИС-142
«__» декабря 2023 г. _____ /Наумов А.А./

Проверил:
профессор кафедры В.С.,
«__» декабря 2023 г. _____ /Родионов А.С./

Оценка « _____»

Новосибирск 2023

ВЫПОЛНЕНИЕ РАБОТЫ

Цель работы

Целью данной работы является изучение метода кластеризации k-средних и анализ влияния уровня разброса (стандартного отклонения) данных на качество кластеризации.

Генерация данных

Данные генерируются с использованием нормального распределения. Формула нормального распределения для генерации случайной величины X с математическим ожиданием μ и стандартным отклонением σ выглядит следующим образом:

$$\boldsymbol{X} \sim \boldsymbol{N}(\mu, \sigma^2)$$

Для каждого набора данных используется свое значение математического ожидания (1, 2, 3), а стандартное отклонение изменяется от 0.1 до 1.0. Каждая выборка будет содержать 100 точек, распределенных по трем столбцам Xi, Yi, Zi (выборкам 1, 2, 3).

Кластеризация методом k-средних (K-means clustering)

Метод k-средних (K-means clustering) представляет собой алгоритм машинного обучения для кластеризации данных. Кластеризация - это задача разделения набора данных на группы (кластеры) таким образом, чтобы объекты внутри одной группы были более похожи друг на друга, чем на объекты в других группах. Кластеризация - это форма без учителя (unsupervised learning), где модель обучается выявлять структуру в данных без использования меток классов.

Принцип работы метода k-средних следующий:

1. Инициализация центроидов:
- Выбирается количество кластеров (k).
- Инициализируются случайным образом центроиды для каждого кластера. Центроид - это центр масс (среднее) точек внутри кластера.

2. Присвоение точек к кластерам:
- Каждая точка в наборе данных присваивается к кластеру, центроид которого находится ближе всего к этой точке. Расстояние может быть измерено, например, с использованием евклидова расстояния.

3. Пересчет центроидов:
- Пересчитываются центроиды как среднее значение точек внутри каждого кластера.

4. Повторение шагов 2-3:
- Шаги 2 и 3 повторяются до тех пор, пока центроиды не стабилизируются или не достигнут критерия остановки.

5. Вывод результата:
- Когда процесс завершен, каждая точка данных принадлежит к какому-то

кластеру. Полученные кластеры могут быть визуализированы, проанализированы, или использованы в дальнейших задачах, таких как предсказание принадлежности к кластеру для новых данных.

Основные параметры метода k-средних включают количество кластеров (k), а также различные параметры, такие как критерий остановки, способ инициализации центроидов и т. д.

Метод k-средних чувствителен к начальному выбору центроидов и может сходиться к локальному минимуму. В связи с этим, для улучшения результатов, иногда используются несколько случайных начальных инициализаций и выбирается лучшая.

Весь метод можно реализовать через python-библиотеку SciLearn.
Весь подсчет будем делать через код.

Код на Python демонстрирует использование метода k-средних (k-means) для кластеризации данных.

1. Генерация данных:
   - Создаются три выборки данных, каждая из которых представляет собой 100 точек, сгенерированных из нормального распределения с матожиданием 1, 2 и 3 соответственно и стандартным отклонением `sigma = 0.1`.
   - Все три выборки объединяются в одну 2D-матрицу `data_2d_sigma_05`.
2. Применение метода k-средних:
   - Используется библиотека `scikit-learn` для применения метода k-средних с `n_clusters=3` (3 кластера).
   - Модель обучается на сгенерированных данных, и каждой точке присваивается метка кластера.
3.Визуализация результатов:
   - Создается график размером 12x8 дюймов.
   - Исходные данные каждой выборки отображаются на графике с различными цветами (`red`, `green`, `blue`) и прозрачностью `alpha=0.5`.
   - Кластеризованные данные также отображаются на графике с цветами кластеров (`orange`, `magenta`, `black`), их границы обозначены черными кромками.
   - Центроиды каждого кластера обозначаются черными крестами (`marker='x'`).
   - График снабжен заголовком и метками для осей X и Y, а также легендой для легкости восприятия.
4. Определение точек, попавших не в свой кластер:
   - Для каждой из трех исходных выборок определяются точки, которые были неправильно классифицированы, и они сохраняются в списке `misplaced_points`.
5. Подсчет количества ошибочно классифицированных точек в каждом кластере:
   - Для каждого кластера подсчитывается количество точек, которые были

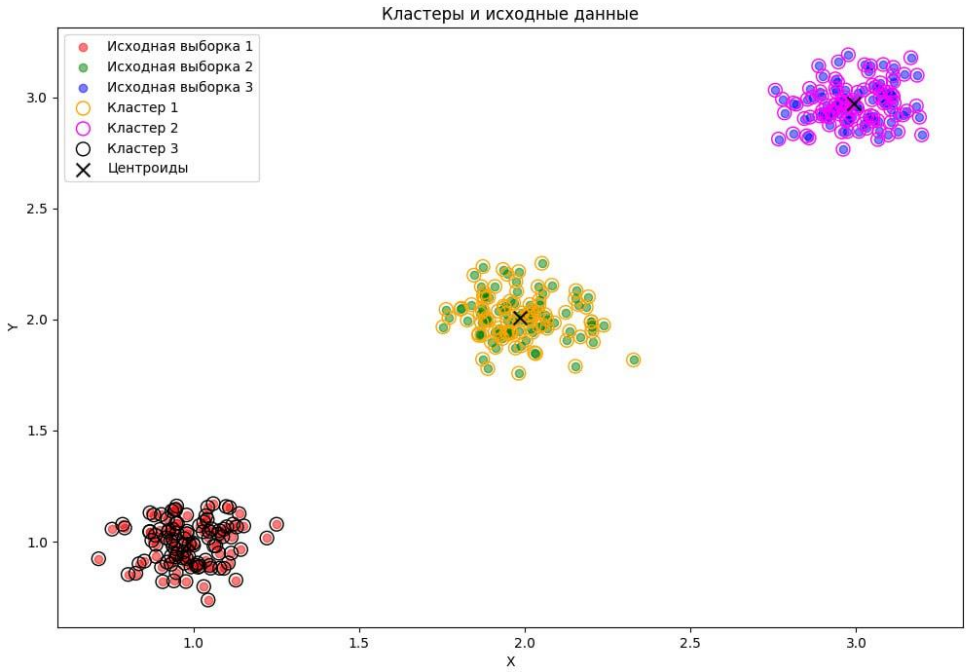
ошибочно классифицированы, и результат сохраняется в списке `misplaced_points_counts`.

Этот код полезен для иллюстрации процесса кластеризации методом k-средних и визуализации результатов, включая выделение точек, которые были неправильно классифицированы.

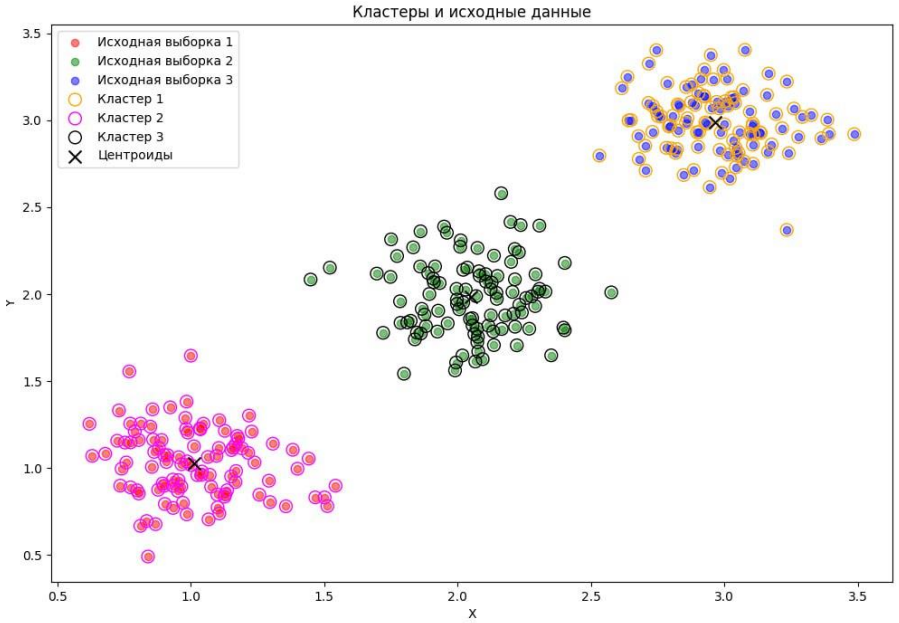
Визуализация данных

Библиотека matplotib будет использоваться для визуализации данных в коде. Данные из каждого набора будут визуализированы с использованием различных цветов.

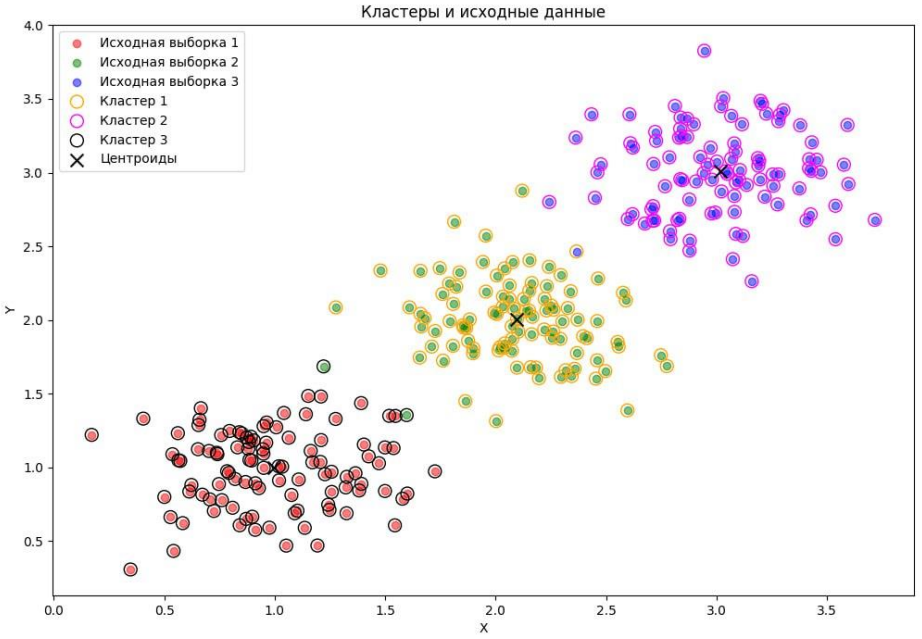
σ=0.1



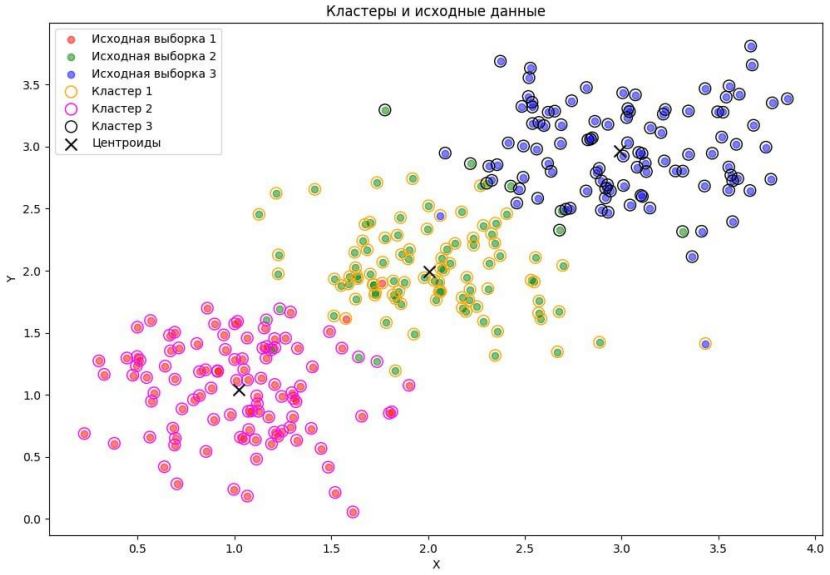
σ=0.2



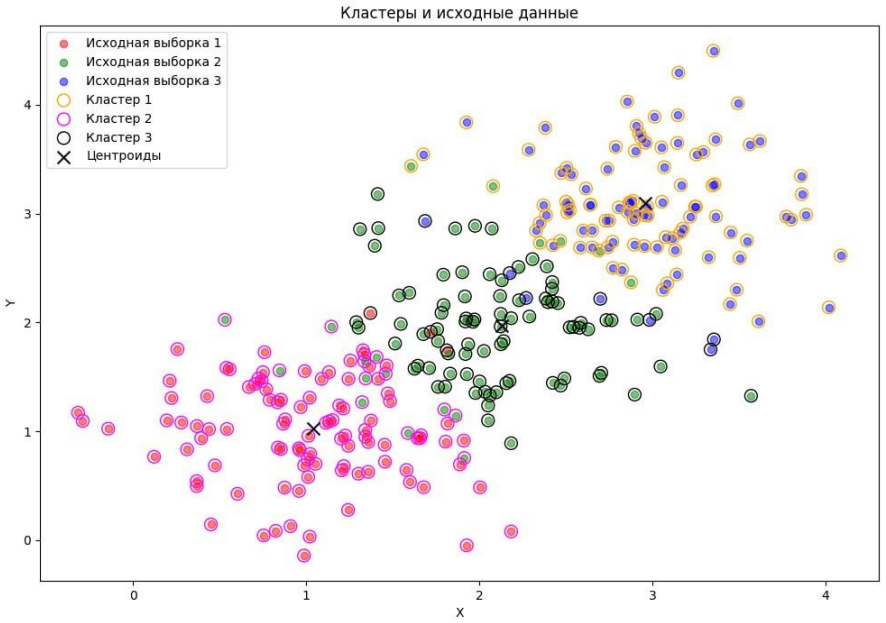
σ=0.3(3 промаха)



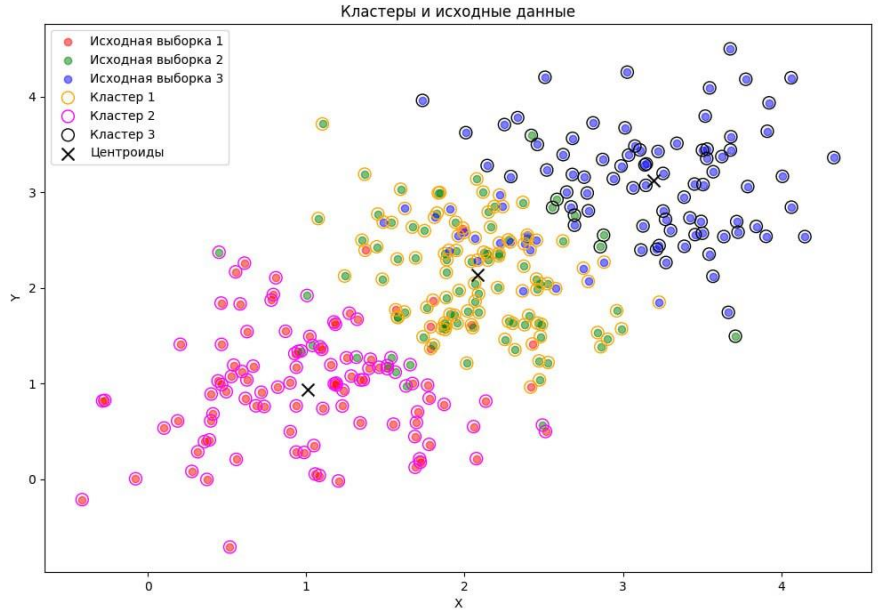
σ=0.4(16 промахов)



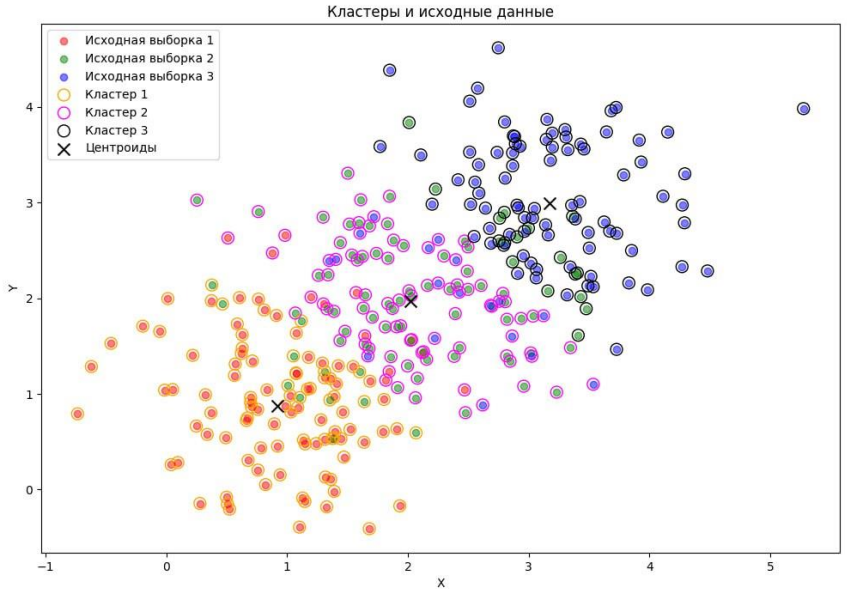
σ=0.5(28 промахов)



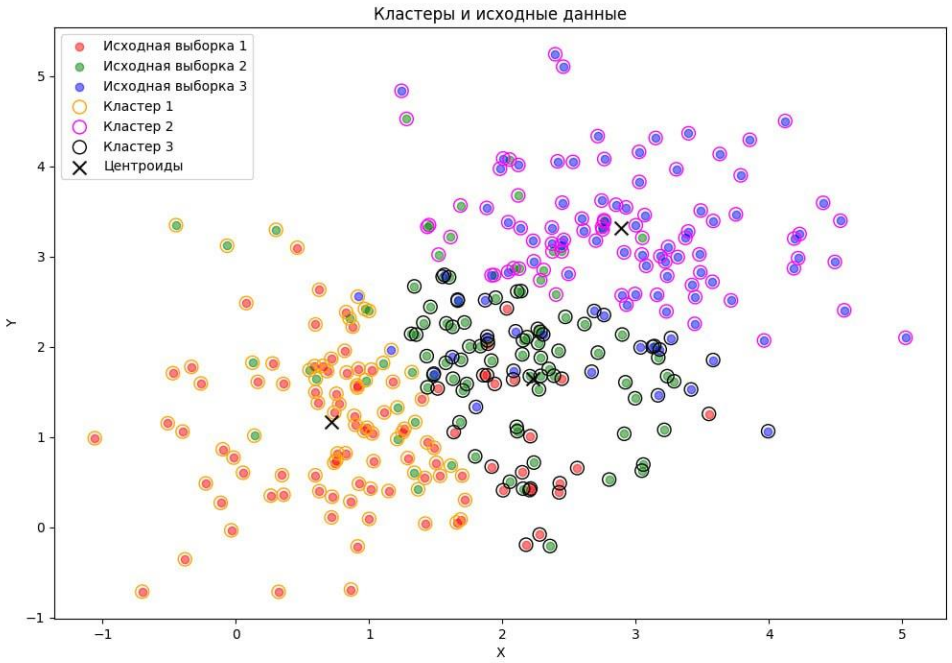
σ=0.6(47 промахов)



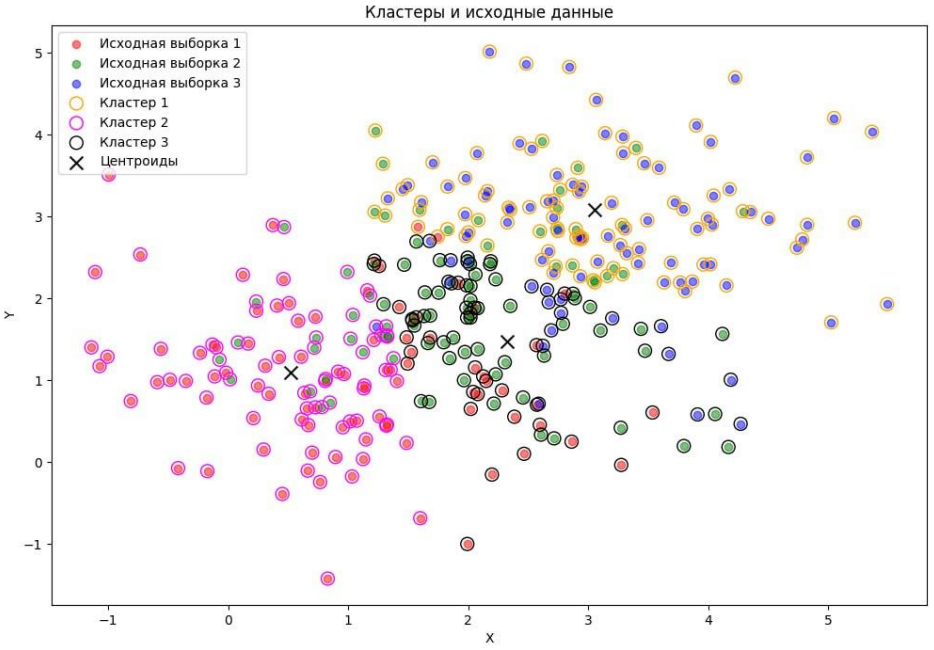
σ=0.7(63 промахов)



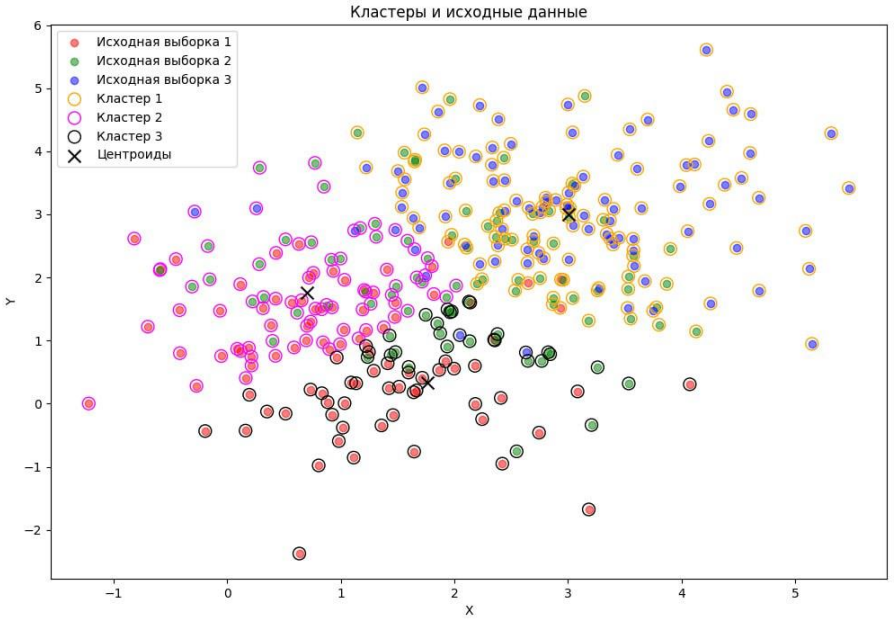
σ=0.8(75 промахов)



σ=0.9(93 промаха)



σ=1.0(109 промахов)



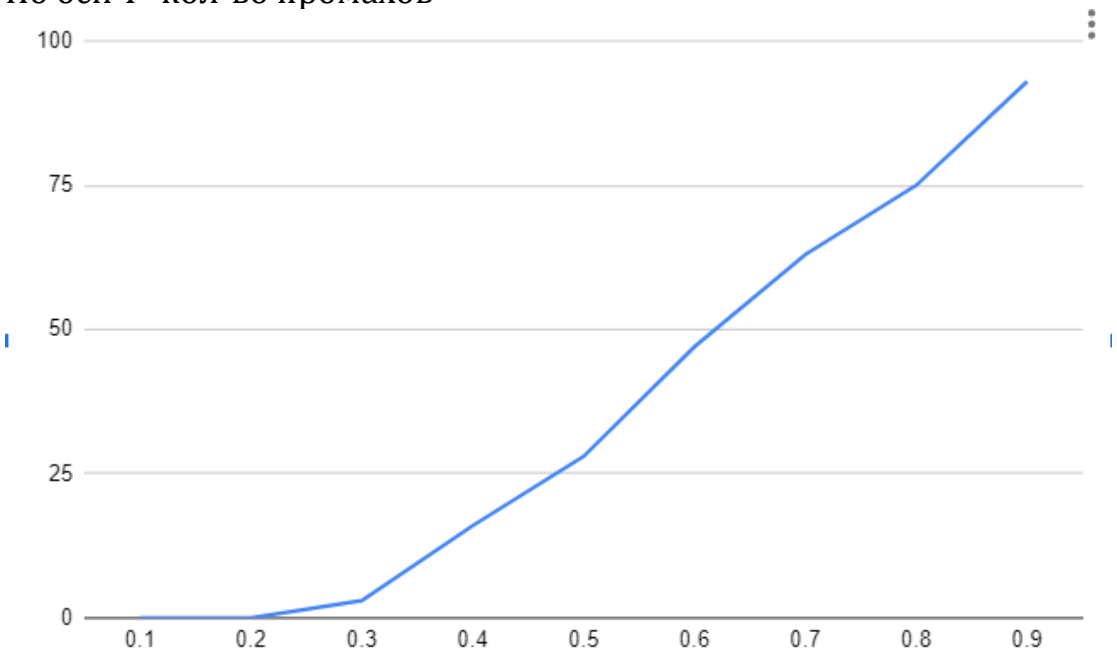
Программа для генерации данных, определения кластеров методом K-средних и последующая за ним визуализация результатов на python.

```
import matplotlib.pyplot as plt
import numpy as np
from sklearn.cluster import KMeans

# Генерация данных с матожиданиями 1, 2, 3 и отклонением sigma
means = [1, 2, 3]
sigma = 1
data_2d_sigma_05 = []

for mean in means:
    data_2d_sigma_05.append(np.random.normal(mean, sigma, (100, 2)))

data_2d_sigma_05 = np.concatenate(data_2d_sigma_05, axis=0)

# Применение метода k-средних
kmeans = KMeans(n_clusters=3)
kmeans.fit(data_2d_sigma_05)
labels = kmeans.predict(data_2d_sigma_05)

# Визуализация кластеров
plt.figure(figsize=(12, 8))

# Цвета для исходных данных
original_colors = ['red', 'green', 'blue']

# Отображение исходных данных с их цветами
for i in range(3):
    sample_data = data_2d_sigma_05[i*100:(i+1)*100]
    plt.scatter(sample_data[:, 0], sample_data[:, 1], alpha=0.5, label=f'Исходная выборка {i+1}', color=original_colors[i])

# Цвета для кластеров
cluster_colors = ['orange', 'magenta', 'black']

# Отображение кластеризованных данных с цветами кластеров
```

```
for i in range(3):
    cluster_data = data_2d_sigma_05[labels == i]
    plt.scatter(cluster_data[:, 0], cluster_data[:, 1], alpha=1.0,
edgecolor=cluster_colors[i], facecolor='none', s=100, label=f'Кластер {i+1}')

# Отображение центроидов кластеров
centroids = kmeans.cluster_centers_
plt.scatter(centroids[:, 0], centroids[:, 1], color='black', marker='x', s=100,
label='Центроиды')

plt.title('Кластеры и исходные данные')
plt.xlabel('X')
plt.ylabel('Y')
plt.legend()
plt.show()

# Определение точек, попавших не в свой кластер
misplaced_points = []
for i in range(3):
    original_sample = data_2d_sigma_05[i*100:(i+1)*100]
    misplaced = original_sample[labels[i*100:(i+1)*100] != i]
    misplaced_points.append(misplaced)

# Подсчёт количества ошибочно классифицированных точек в каждом кластере
misplaced_points_counts = [len(mp) for mp in misplaced_points]

# Вывод общего количества ошибочно классифицированных точек
print(f'Количество ошибочно классифицированных точек: {misplaced_points_counts}')
```

Анализ промахов

С увеличением значения нормального отклонения **σ,** количество промахов растет почти с линейным отклонением. Ниже представлен график
по оси X – значение нормального отклонения
По оси Y- кол-во промахов



Вывод

1. Влияние стандартного отклонения

Один из ключевых выводов из проведенной работы заключается в том, что точность кластеризации методом k-средних существенно зависит от стандартного отклонения в данных. При уменьшении значения стандартного отклонения кластеры формируются более четко и с высокой точностью. Однако при увеличении стандартного отклонения наблюдается увеличение количества ошибок классификации. Это может быть обусловлено увеличением перекрытия между кластерами, что затрудняет их четкое разграничение.

2. Линейная зависимость ошибок от σ

Имеется предположение о возможной линейной зависимости между количеством ошибочно классифицированных точек и стандартным отклонением данных. Это наблюдение подчеркивает важность тщательного анализа вариации данных при выборе метода кластеризации.

3. Влияние начального выбора центроидов

Метод k-средних инициализируется случайным выбором начальных центров кластеров, что может существенно влиять на конечный результат. Особенно при большом разбросе данных начальный выбор центроидов может привести к значительным различиям в результатах кластеризации. Это подчеркивает важность многократного запуска алгоритма с разными начальными условиями для получения более надежных результатов.

4. Применение в реальных сценариях

На основе проведенного анализа можно предположить, что метод k-средних будет наиболее эффективен в сценариях, где данные хорошо разделяются и обладают низким стандартным отклонением. В случаях с высоким уровнем шума или значительным перекрытием между классами может потребоваться применение более сложных методов кластеризации или предварительная обработка данных.

В целом, результаты подчеркивают, что метод k-средних представляет собой мощный инструмент для кластеризации данных, но его эффективность может варьироваться в зависимости от характеристик конкретного набора данных. Тщательный анализ и выбор параметров являются ключевыми при применении этого метода в практических сценариях.