ПРАВИТЕЛЬСТВО РОССИЙСКОЙ ФЕДЕРАЦИИ
НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ
«ВЫСШАЯ ШКОЛА ЭКОНОМИКИ»

Факультет компьютерных наук
Образовательная программа «Прикладная математика и информатика»

Отчет о программном проекте

на тему «Разработка систем предсказания успешногозавершения учебной дисциплины»

Выполнил:
студент группы БПМИ186 _____ Д. А. Чужмаров
Подпись И.О. Фамилия
23.06.2020
Дата

Принял:
руководитель проекта Андрей Андреевич Паринов
Имя, Отчество, Фамилия
м.н.с. МНУЛ ИССА ФКН НИУ ВШЭ
Должность Место работы

Дата _____2020 _____ _____
Оценка (по 10-тибалльной шкале) Подпись

Москва 2020

Анотация

Данный отчет является одной частью командной работы. Основной задачами данной работы являются: изучение, сравнение, реализация и применение на практике алгоритмов кластеризации. В данной отчете представлены алгоритмы агломеративной кластеризации и спектральной кластеризации. Описан принцип работы двух алгоритмов и представлены результаты работы на реальных данных. Изначально планировалось брать данные студентов и основываясь на их текущих оценках давать предсказания успешного/неуспешного завершения программы, однако данные студентов были недоступны. В качестве датасета для тестирования использовался оригинальный датасет репетиторов, собранный с сайта https://repetitors.info/. Также была реализована клиент-серверная архитектура API на фреймворке Flask.

Содержание

Введение

Кластерный анализ данных используется в качестве общего метода в современных научных исследованиях, которые проходят через коммуникационные науки, компьютерные науки и биологические науки. Кластеризация, как основной состав анализа данных, играет значительную роль. С одной стороны, было создано много инструментов для кластерного анализа, наряду с увеличением объема информации и пересечением тематики. С другой стороны, каждый алгоритм кластеризации имеет свои сильные и слабые стороны, обусловленные сложностью информации. Далее мы разберем, что такое кластеризация, рассмотрим алгоритмы агломеративной и спектральной кластеризации.

Кластеризация, рассматриваемая как наиболее важный процесс неконтролируемого обучения, имеет дело с разделением структуры данных в неизвестной области и является основой для дальнейшего обучения. Полное определение кластеризации описывается следующим образом:

1) Экземпляры, находящиеся в одном кластере, должны быть максимально похожи;
2) Экземпляры в разных кластерах должны быть максимально разными;
3) Измерение сходства и несходства должно быть четким и иметь практическое значение

Агломеративная кластеризация

Идея

Объединять объекты в кластер, используя некоторую меру сходства или расстояние между объектами.

Шаги алгоритма

1) Присваиваем каждому объекту свой кластер
2) Сортируем попарные расстояния между кластерами
3) Берём пару ближайших кластеров, склеиваем их в один
4) Повторяем 2 и 3 пункт пока все объекты не попадут в один кластер

Методы объединения точек

1) *Метод одиночной связи* – в основе этого метода лежит минимальное расстояние
2) *Метод полной связи* – в основе этого метода лежит максимальное расстояние между объектами
3) *Метод средней связи* – в основе этого метода лежит среднее расстояние между каждой парой объектов из разных кластеров
4) *Центроидный метод* – в основе этого метода лежит минимальное расстояние между центрами разных кластеров

Сложность алгоритма

$O(n + n^2d + n^2log(n^2) + n^3) = O(n^3)$

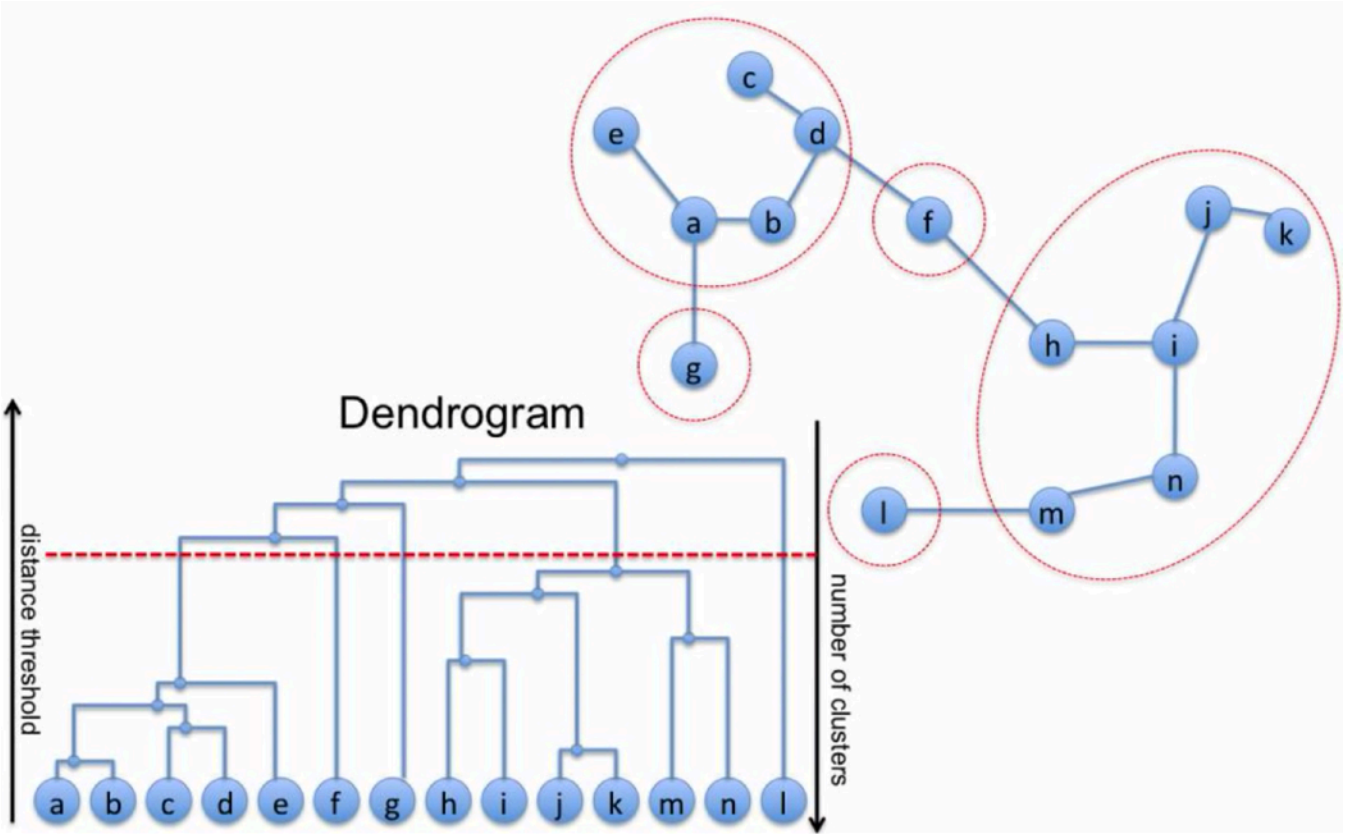
1-ый пункт: $O(n)$

2-ой пункт: $O(n^2d)$ – создание массива n на n, где каждый элемент стоит *d*

3-ий пункт: $O(n^2log(n^2))$

4-5-ые пункты: $O(n^3)$ ищем какому кластеру принадлежат точки, затем обновляем их кластеры за n и повторяем это n^2 раз

Результат работы такого алгоритма



Дерево склеивания кластеров - граф показывающий работу алгоритма, как следствие показывает, на каком этапе нужно остановить алгоритм

Реализация агломеративной кластеризации

В моем случае был использован метод одиночной связи: Мы объединяем 2 кластера, если расстояние между выбранным элементом первого кластера и выбранным элементом второго кластера меньше, чем расстояние между любыми другими парами элементов любых других кластеров.

Альтернативой был метод средней связи, где находилось расстояние между средними значениями кластеров. Такой метод идейно не отличается от выбранного, однако реализацию такого сказать нельзя. Я подсчитал сложность этого алгоритма: самая затратная часть – это перерасчет расстояния на каждом шаге, однако это гарантировало, что с каждой итерацией количество кластеров будет изменяться поэтому сложность его осталась $O(n^3)$.

Поскольку сложность алгоритмов одинакова, а точность в кластеризации понятие субъективное и разное в каждой задаче, я выбрал первый алгоритм так как по моему мнению он более понятный

Была написана функция поиска расстояния между точками, которая принимает пару индексов точек и сам список точек, возвращает расстояние между ними. В Евклидовой метрике.

В качестве координат точек были взяты оценки репетиторам и названия предметов

Далее были написаны оговоренные ранее шаги.

Выводом из алгоритма был словарь, где ключом была точка, значением был номер кластера. Номер кластера

соответствует номеру точки, которая первой оказалась в этом кластере.

Спектральная кластеризация

Определение

Спектральной кластеризацией называются все методы, которые разбивают множество на кластеры с помощью собственных векторов матрицы схожести или других матриц, полученных из нее

В основе данного алгоритма лежит алгоритм разделение графа с помощью матрицы Лапласа. В данном алгоритме мы представляем наши данные как элементы графа.

Шаги алгоритма

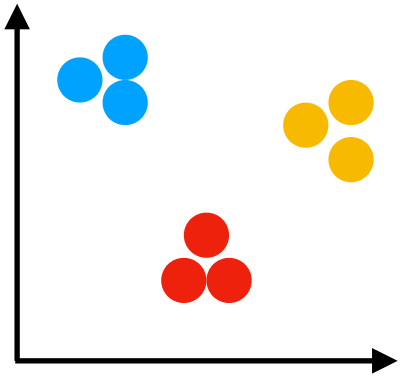
1. Строим матрицу схожести
2. Строим диагональную матрицу
3. Находим матрицу Лапласа
4. Затем вычисляем два самых маленьких собственных значения величины и соответствующие собственные вектора
5. Разделяем кластер в два кластера с помощью вектора Fiedler
6. Далее работаем с вектором Fiedler: Элемент присваиваем первому кластеру, если он имеет положительное значение в w
7. Для дальнейшего разбиения повторяем 1–6 шаги для любого из кластеров

Использованные термины

- *Матрица схожести*
  - Квадратная матрица, где a_{ij} = схожесть точек с номером i и j, в нашем случае расстояние в Евклидовой метрике между i и j
- *Диагональная матрица*
  - Матрица, где a_{ii} = сумме всех элементов матрицы схожести в i-ом ряду, остальные элементы = 0
- *Матрица Лапласа*
  - Матрица = Матрица схожести - Диагональная матрица
- *Вектор Fiedler*
  - Собственный вектором, соответствующим второму самому маленькому собственному значению графика.

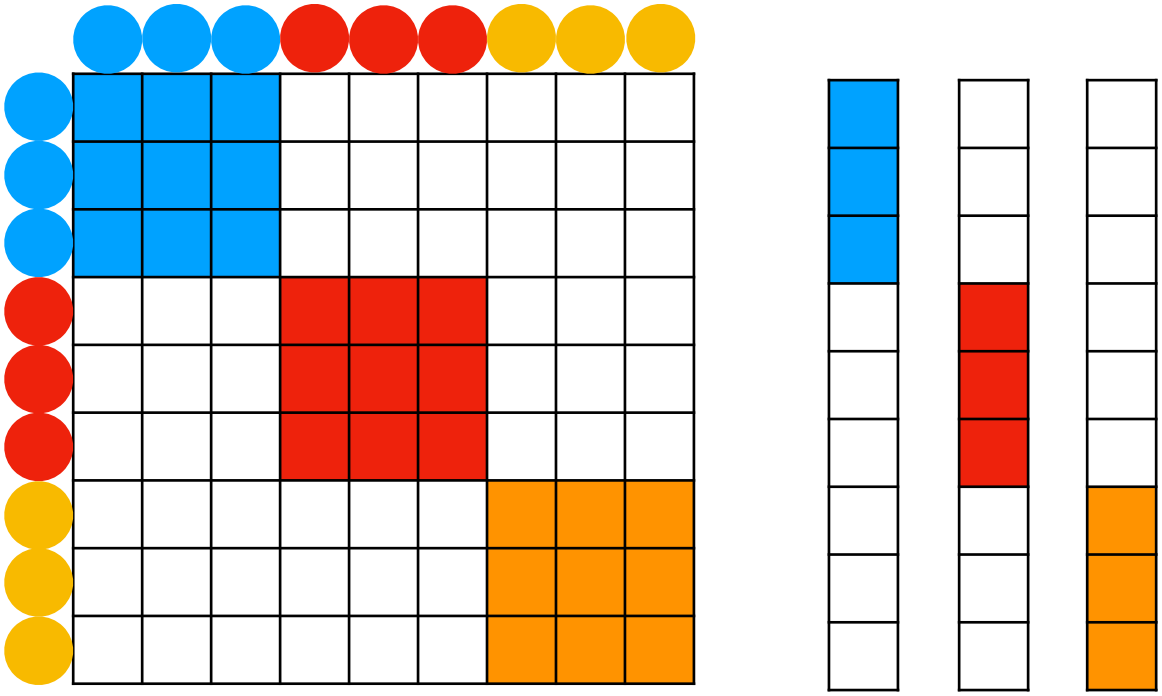
Иллюстрация работы алгоритма

Пусть нам на вход подаются следующие точки:



Цвета сделаны для удобства

Построим матрицу сходства для этих точек:



Цветом отметим места с максимальным сходством.

Найдем собственный вектор:

Видно, что собственный вектор разбил точки на нужные кластеры

Реализация спектральной кластеризации

Описаный алгоритм в позволяет разрезать граф по наименьшему потоку. Чем меньше вес между двумя ребрами в графе, тем проще их разделить, однако в нашей задаче за вес ребер мы берем расстояние между ними. Если расстояние большое, то и вес будет большим, а нам нужно наоборот. Поэтому мы дополнительно модифицируем матрицу схожести: (-1) * A + B, где B это квадратная матрица, где каждый элемент равен максимальному в A. Поскольку каждый элемент имеет ребро с каждым мы можем так сделать.

Далее были написаны оговоренные ранее шаги.

Выводом из алгоритма был словарь, где ключом была точка, значением был номер кластера. Номер кластера соответствует номеру точки, которая первой оказалась в этом кластере.

Задача второго этапа проекта:

Реализация клиент-серверной архитектуры

Клиент серверная-архитектура API

API (от англ. ***a****pplication* ***p****rogramming* ***i****nterface*) — описание способов которыми одна компьютерная программа может взаимодействовать с другой программой. Обычно входит в описание какого-либо интернет-протокола программного каркаса (фреймворка). Используется программистами при написании всевозможных приложений.

REST и RESTful

REST – аббревиатура от Representational State Transfer («передача состояния представления»). Это согласованный набор архитектурных принципов для создания более масштабируемой и гибкой сети.

Архитектурные принципы:

1) **Клиент-сервер**

   Первое ограничение указывает, что сеть должна состоять из клиентов и серверов. Сервер — это компьютер, который имеет требуемые ресурсы, а клиент — это компьютер, которому нужно взаимодействовать с ресурсами, хранящимися на сервере.

2) **Отсутствие состояния**

   Клиент и сервер не отслеживают состояние друг друга. Когда клиент не взаимодействует с сервером, сервер не имеет представления о его существовании.

3) **Единообразие интерфейса**
   Ограничение гарантирует, что между серверами и клиентами существует общий язык, который позволяет каждой части быть заменяемой или изменяемой, без нарушения целостности системы. Это достигается через 4 дополнительных ограничения: идентификация ресурсов, манипуляция ресурсами через представления, «самодостаточные» сообщения и гипермедиа.
4) **Кэширование**
   Ответы сервера должны помечаться как кэшируемые или некэшируемые. Кэшируемый ответы сохраняются у пользователя.

Flask

Для реализации моего сервера я выбрал фреймворк Flask. Flask – это фреймворк, служащий для создания вебсайтов на языке Python.

Я написал два класса: Experiment и ExperimentsList.
http://127.0.0.1/experiments/get<int:exp_id>
показывает результат эксперимента<int:exp_id>

http: // 127.0.0.1/experiments/post<>
создает новый эксперимент

Тестирование

В ходе алгоритмы тестировались на реальных данных. В качестве датасета для тестирования использовался оригинальный датасет репетиторов, собранный с сайта https://repetitors.info/.

Для работы с датасетом репетиторов необходимо было считать данные с Excel для этого было использовано расширение Pandas.
Сложности:

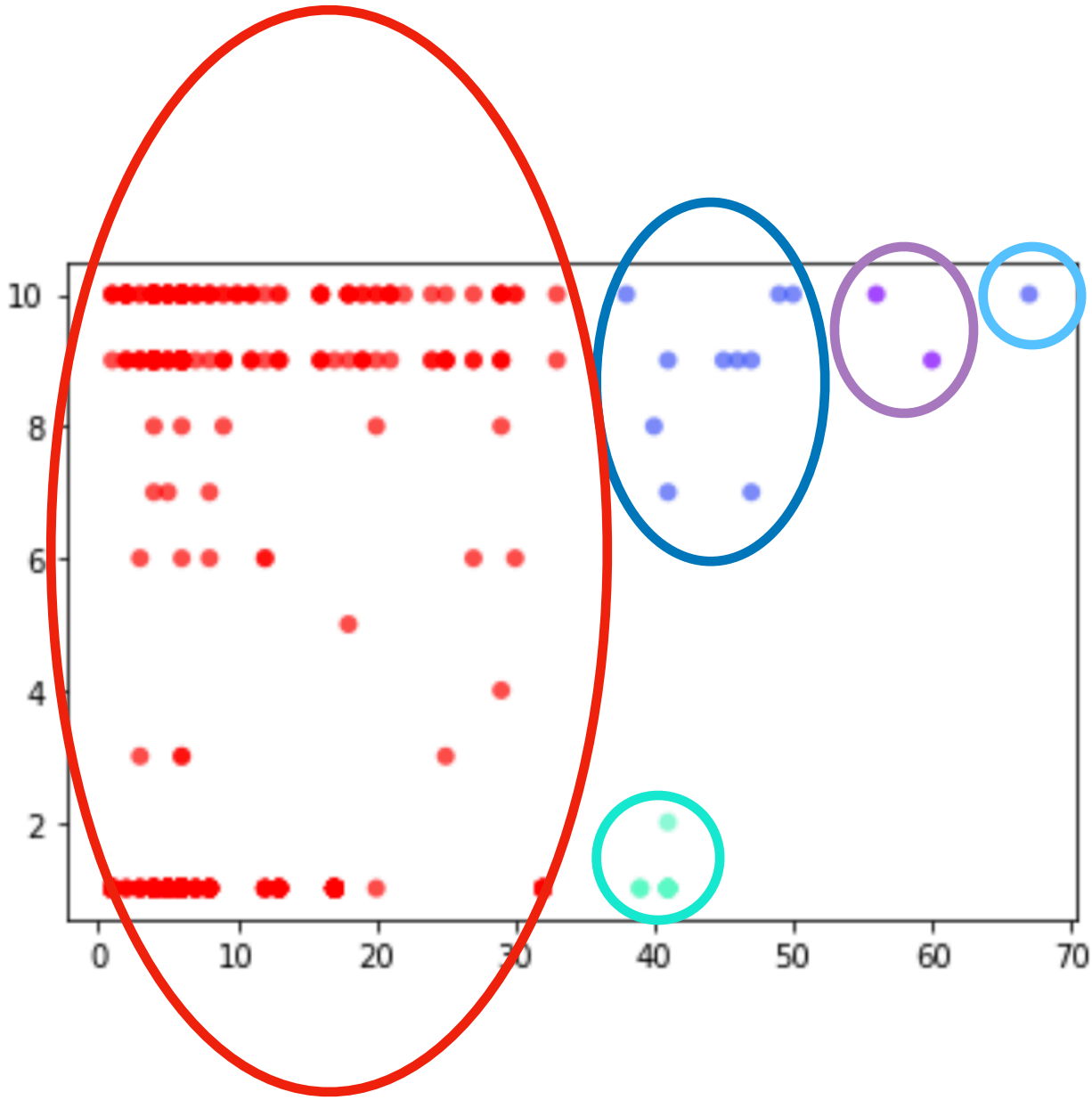
1) Необходимо конвертировать данные из Excel Pandas.dataDataFrame
2) Конкатенируем страницы таблицы
3) Обработать данные, убрав ненужные и перевести оценки и предметы в числовой формат

Далее идет экспериментальная часть, где были подобраны оптимальные параметры алгоритма для эффективной и точной работы кластеризации.

Результаты работы алгоритма

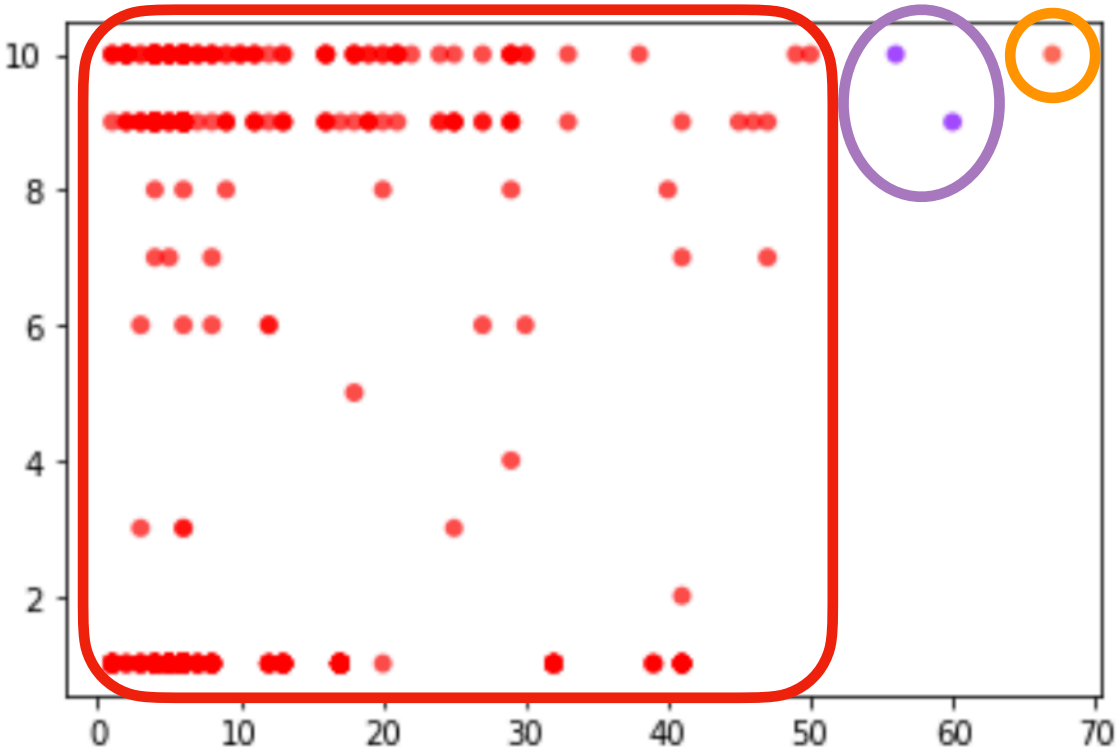
Далее представлены результаты работы на оригинальном дата сете.

Агломеративная кластеризация

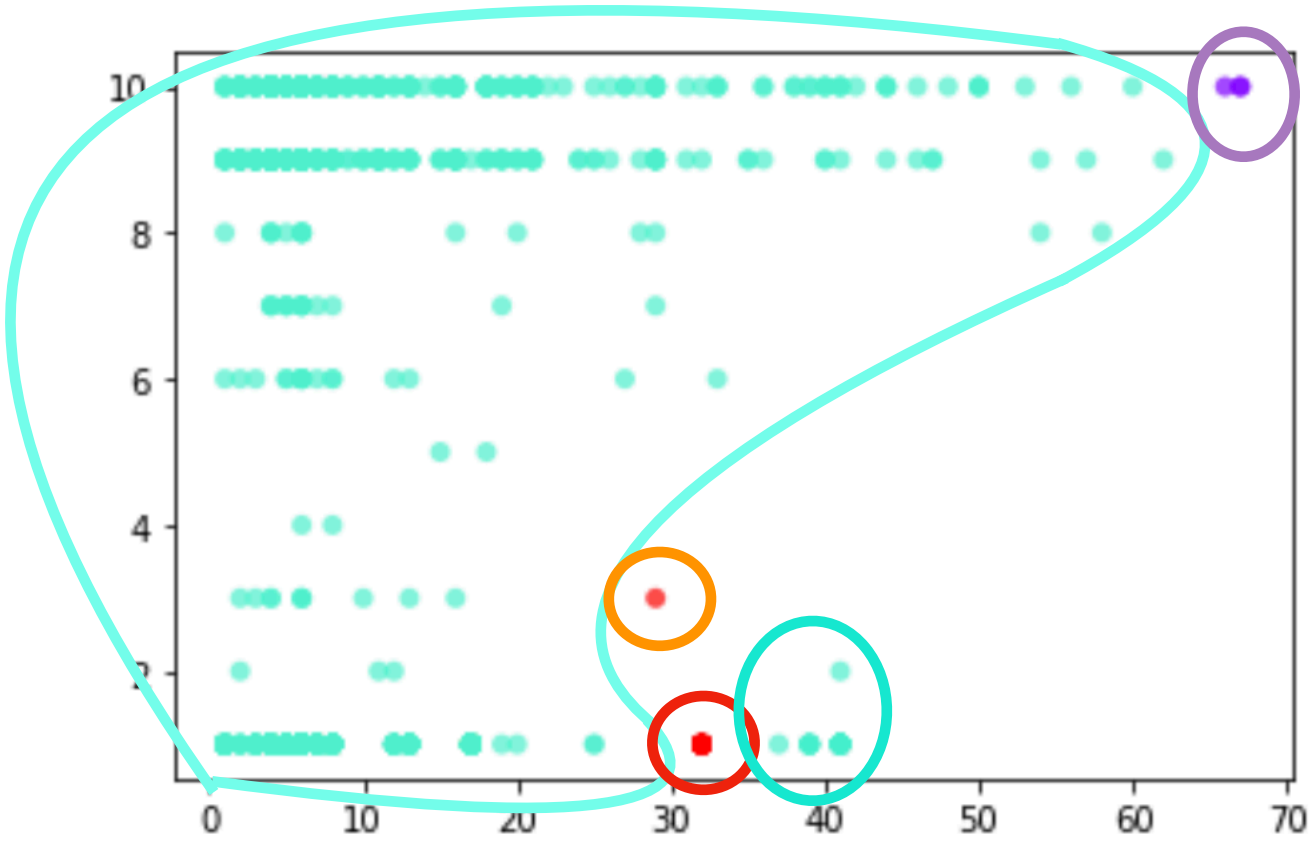


Разбиение на 5 кластеров

Разбиение на 3 кластера



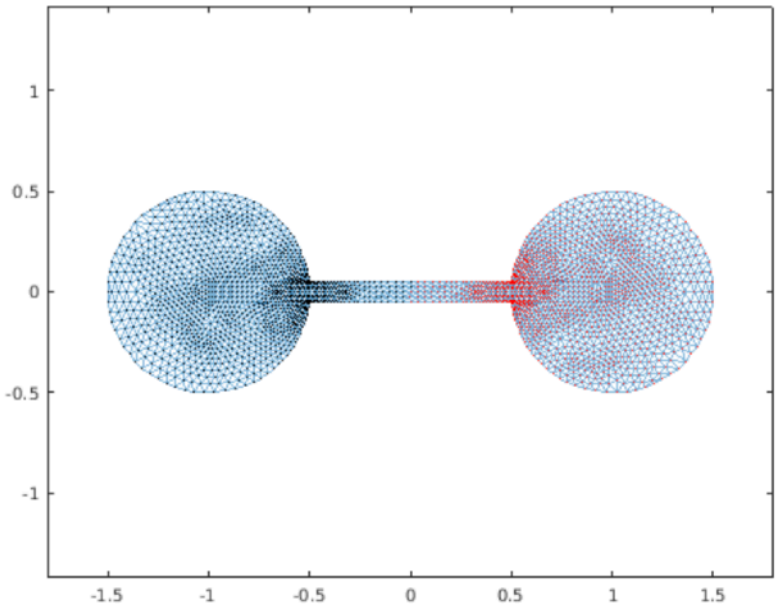
Разбиение на 5 кластеров с большим числом



элементов (в 2 раза больше)

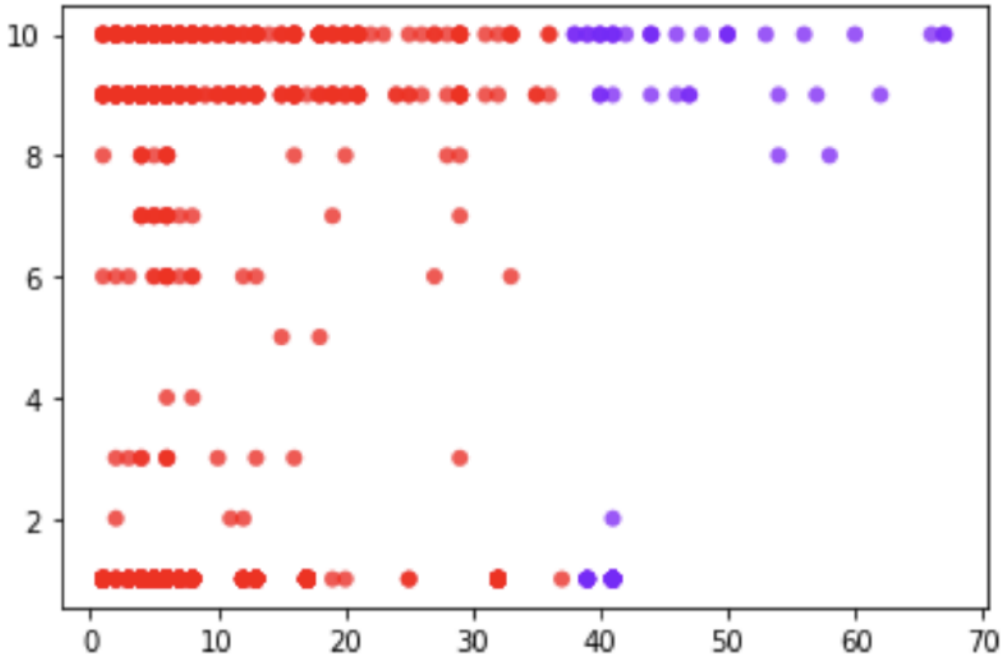
Спектральная кластеризация

Работа спектральной кластеризации:

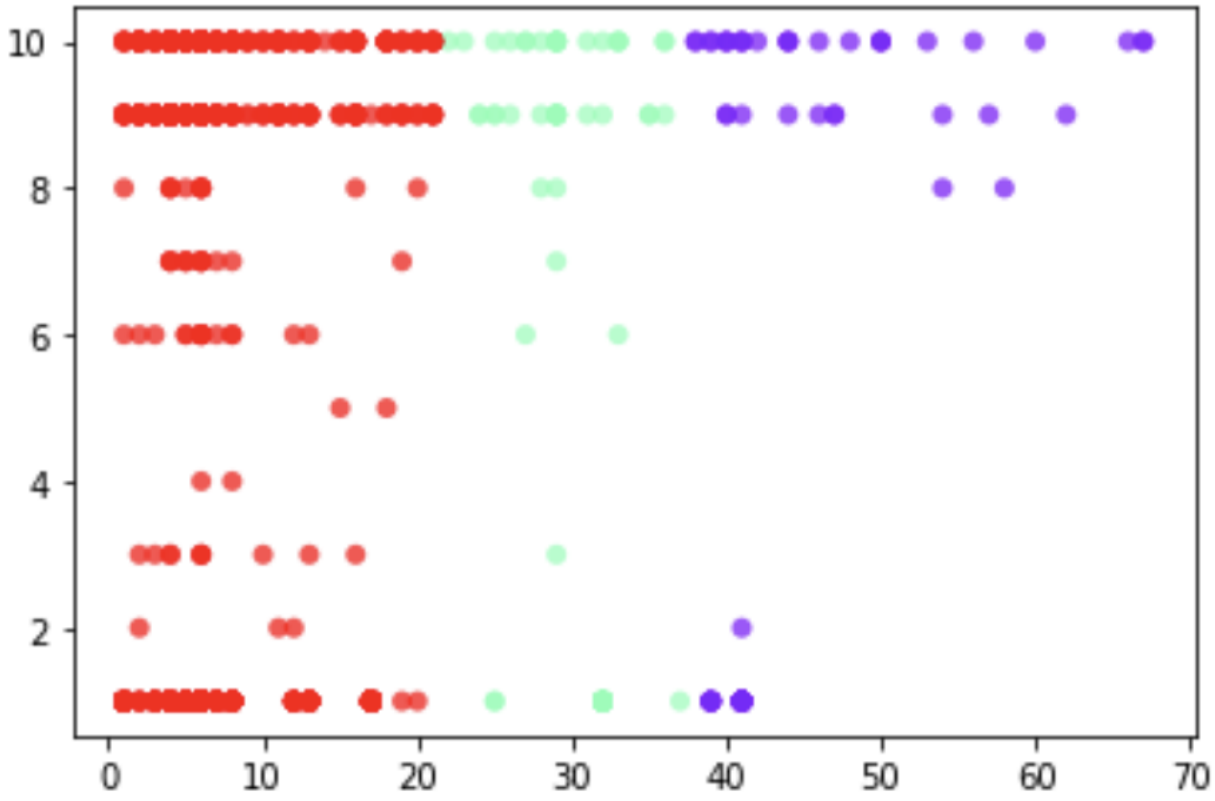


Тут красным помечены элементы одного кластера, черным другого

Однако на нашей базе данных разбиение выглядит так:



Для трех кластеров



Выводы:

Видно, что агломеративный и спектральные алгоритмы хорошо справились с поставленной задачей. Однако не все так прекрасно, из-за сложности равной о от н в кубе алгоритмы не способны обработать всю базу данных, вернее это занимает огромное количество времени. Поэтому мне пришлось искусственно обрезать базу данных, для такой базы данных как наша я не рекомендую использовать эти алгоритмы.

Ссылки на литературу

Агломеративная кластеризация:

1) YouTube: https://www.youtube.com/watch?v=XJ3194AmH40
2) Habr: https://habr.com/ru/company/ods/blog/325654/
3) Wikipedia: https://clck.ru/JuWgj
4) https://lektsii.org/2-87430.html

Спектральная кластеризация:

1) Wikipedia: https://clck.ru/MAby3
2) Habr: https://habr.com/ru/company/ods/blog/325654/
3) YouTube: https://youtu.be/zkgm0i77jQ8
4) https://nsu.ru/xmlui/bitstream/handle/nsu/463/Text_MachulskisSV.pdf

API

1) https://ru.wikipedia.org/wiki/API

REST и RESTful

1) https://habr.com/ru/company/hexlet/blog/274675/
2) https://habr.com/ru/company/dataart/blog/277419/

Flask

1) https://www.freecodecamp.org/news/build-a-simple-json-api-in-python/
2) https://flask-restful.readthedocs.io/en/latest/

Pandas

1) https://pandas.pydata.org/pandas-docs/stable/reference/api/pandas.DataFrame

GitHub

1) https://github.com/Res0nanceD/HSECommandProject2019