Классификация распределения с помощью случайных графов

Кочетков А.В.

Дата: 29 мая 2025 г.

Используемые библиотеки и инструменты

- `networkx` – библиотека с реализованными методами на графах
- `numpy, matplotlib, seaborn, pandas` – библиотеки для визуализации и работы с данными
- `sklearn` – библиотека с алгоритмами машинного обучения, в том числе алгоритмов классификации
- `pytest` – библиотека для проведения тестов

Файл utils.py:

- `build_knn_nx` – построение графа KNN с заданными параметрами по переданному списку
- `build_dist_nx` – построение дистанцированного графа с заданными параметрами по переданному списку
- `calculate_connected_components` — Вычисляет количество связных компонент графа
- `calculate_chromatic_number` — Вычисляет хроматическое число графа (минимальное число цветов для раскраски)
- `calculate_clique_number` — Вычисляет число клики (размер максимальной клики в графе)
- `calculate_size_maximal_independent_set` — Вычисляет размер максимального независимого множества
- `calculate_size_dom_set` — Вычисляет размер доминирующего множества

- `class DataGenerator` — Генератор случайных данных для тестирования гипотез H0 и H1.
- `monte_carlo_experiment` — Проводит Монте-Карло эксперимент для оценки статистических свойств графов
- `monte_carlo_experiment_for_several_characteristics` — Проводит Монте-Карло эксперимент для оценки нескольких характеристик графа.
  Подробное описание всех функций и классов есть в utils.py

Часть 1

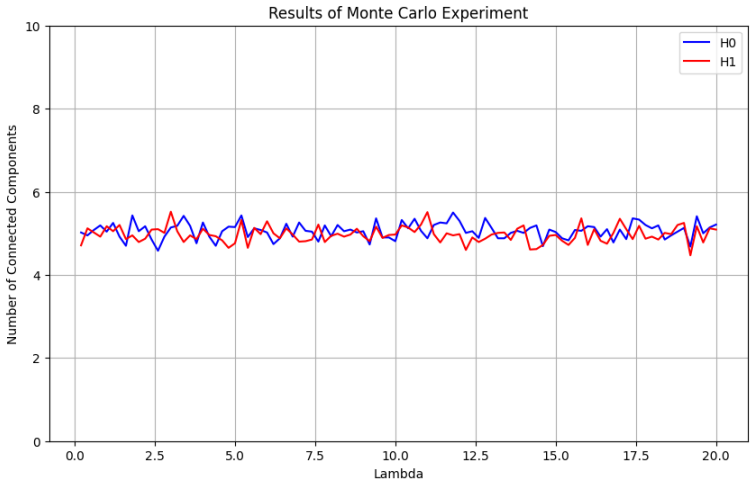
Исследуем, как ведет себя число компонент связности графа T^{KNN} в зависимости от параметров построения, experiments_1

Зафиксируем параметры:

K = 4	число соседей для knn
n_sampels = 100	число итераций в эксперименте Монте-Карло
N = 200	размер набора генерируемых данных

Зависимость от параметра lambda

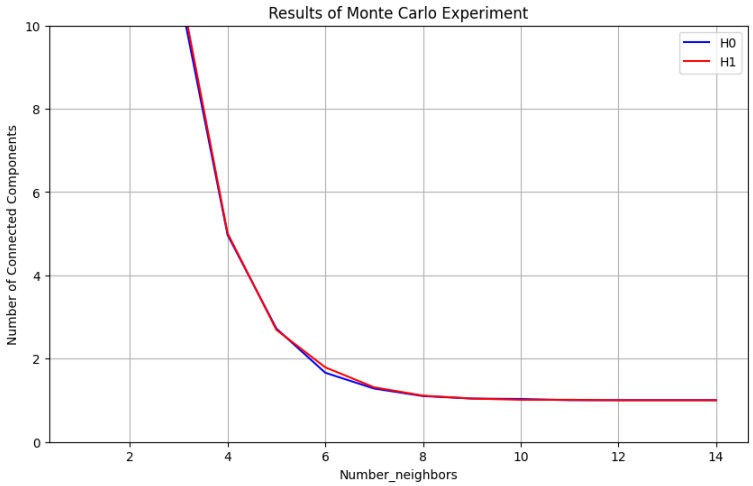
Строим набор параметров lambda от 0.2 до 20 с шагом 0.2 и для каждого параметра и распределения проводим эксперимент Монте-Карло. Считаем матожидание характеристики.



Как мы видим, особой зависимости от параметра lambda нет ни у одного из распределений.

Зависимость от числа соседей

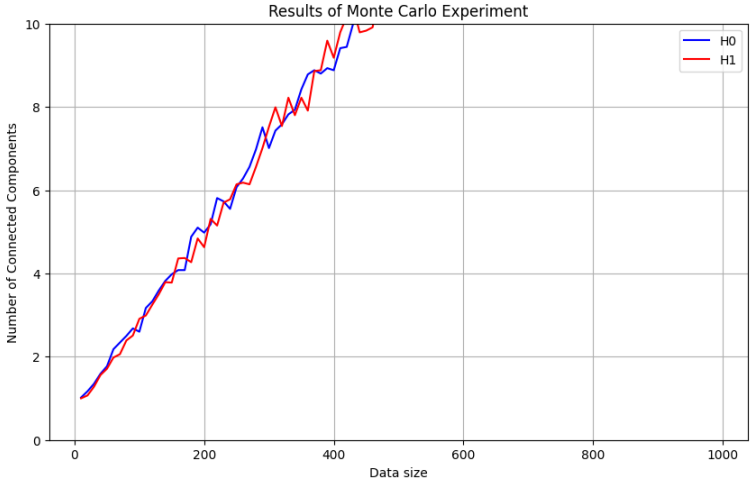
Строим набор параметров от 1 до 15 с шагом 1 и для каждого числа соседей и распределения проводим эксперимент Монте-Карло. Считаем матожидание характеристики.



Для обоих распределений есть зависимость – число компонент связности уменьшается при увеличении числа соседей. Причем для обоих распределений результаты почти идентичны.

Зависимость от размера набора данных

Строим набор размеров данных от 10 до 1000 с шагом 10 и для каждого размера и распределения проводим эксперимент Монте-Карло. Считаем матожидание характеристики.



Для обоих распределений число компонент связности растет при увеличении размера выборки, причем почти одинаково.

Выводы

Характеристика ведет себя почти идентично для обоих распределений, поэтому использовать ее как критерий классификации – плохая идея.

Исследуем, как ведет себя хроматическое число графа T^{dist} в зависимости от параметров построения, experiments_2

Зафиксируем параметры:

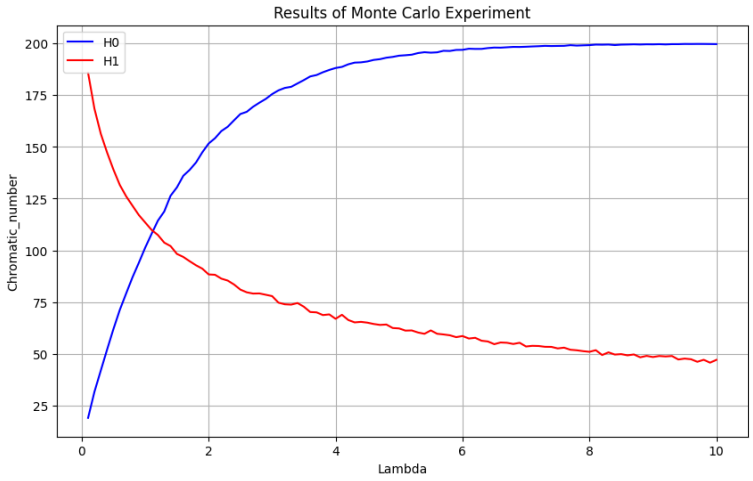
D = 0.7 параметри dist

n_sampels = 100 число итераций в эксперименте Монте-Карло

N = 200 размер набора генерируемых данных

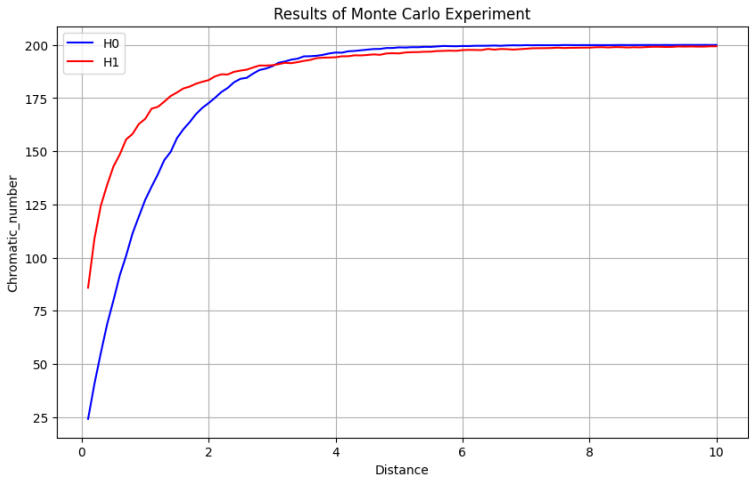
Во всех пунктах делаются действия, аналогичные пунктам выше.

Зависимость от параметра lambda



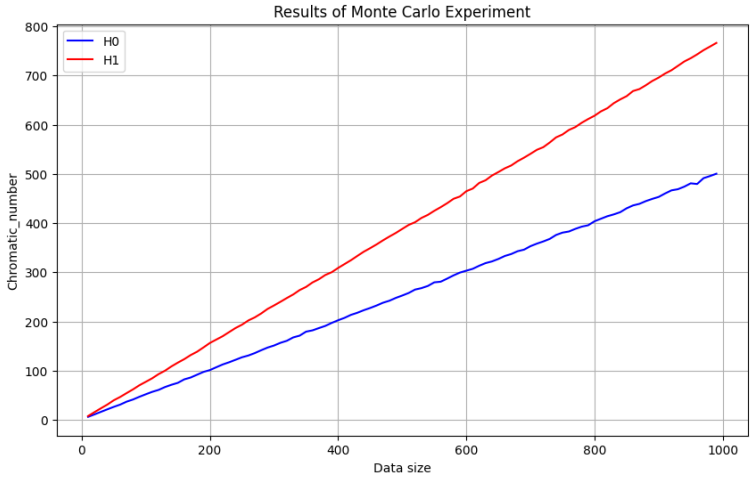
Видим, что при разных распределениях характеристика ведет себя по разному при увеличении lambda (для H0 – возрастает, для H1 – убывает)

Зависимость от параметра dist



Для обоих распределений характеристика возрастает, но для H1 характеристика в начале растет быстрее.

Зависимость от размера данных



Для обоих распределений характеристика возрастает, но для H1 характеристика растет быстрее.

Выводы

При изменении параметров построения характеристики ведут себя по разному. Это говорит о том, что хроматическое число можно использовать как критерий классификации.

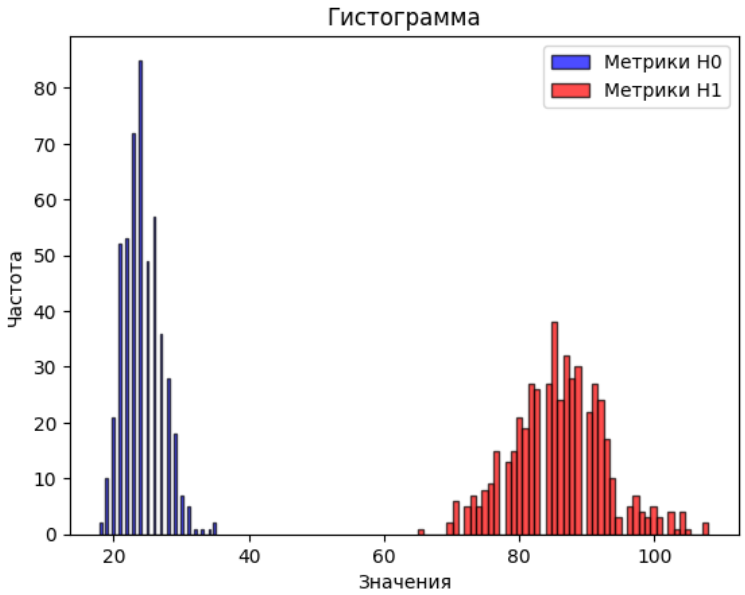
Посмотрим на разделение данных и построим множество A, experiments_3

Зафиксируем параметры:

D = 0.1	параметри dist
n_sampels = 500	число итераций в эксперименте Монте-Карло
N = 200	размер набора генерируемых данных

Исходя из прошлых экспериментов, для построения множества A будем использовать дистанцированный граф, а в качестве характеристики его хроматическое число.

Посмотрим, как хорошо разделяются данные



Для выбранных параметров данные разделяются очень хорошо.

Построим множество A

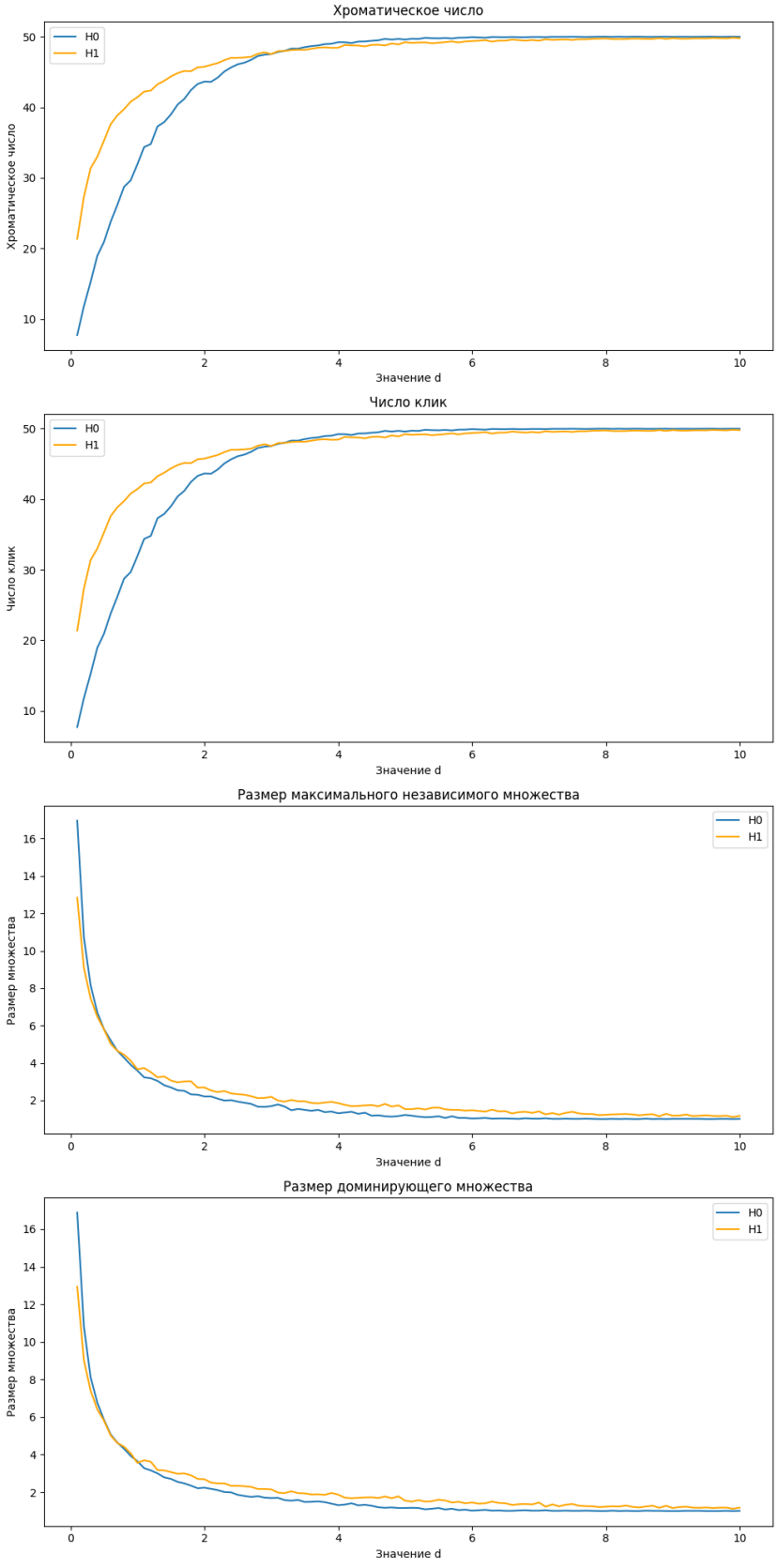
Построим область A так, что левее ее левой границы лежит не более 5% результатов. При фиксированных параметрах получили, что $A = [0, 29]$. Ошибка на H0 получилась ≤ 0.05, полнота H1 $= 1$. Это очень хороший результат

Часть 2

Теперь будем работать только с дистанцированным графом. Выберем характеристики – хроматическое число, кликовое число, размер макс. независимого множества и число доминирования.

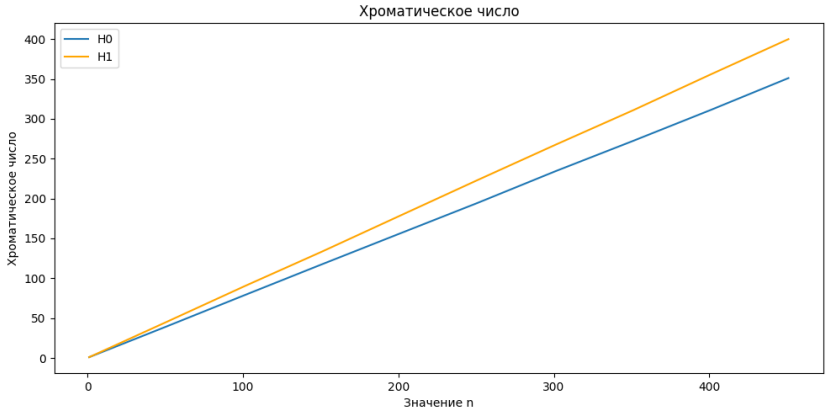
Изучение характеристик для дистанцированного графа, experiments_1

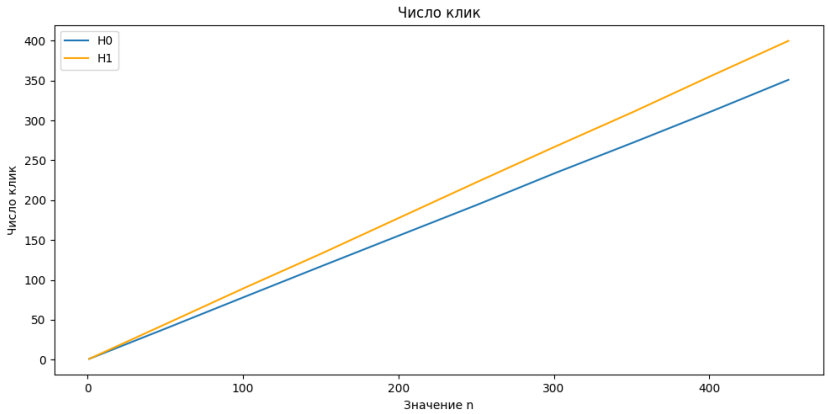
Подберем параметр dist так, чтобы характеристики хорошо разделялись

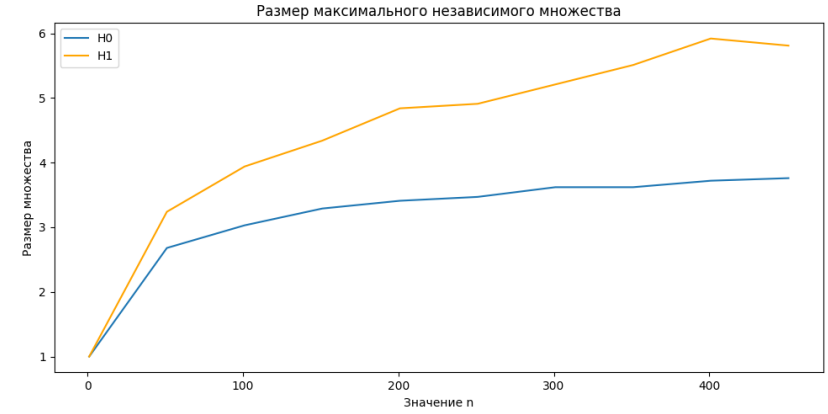


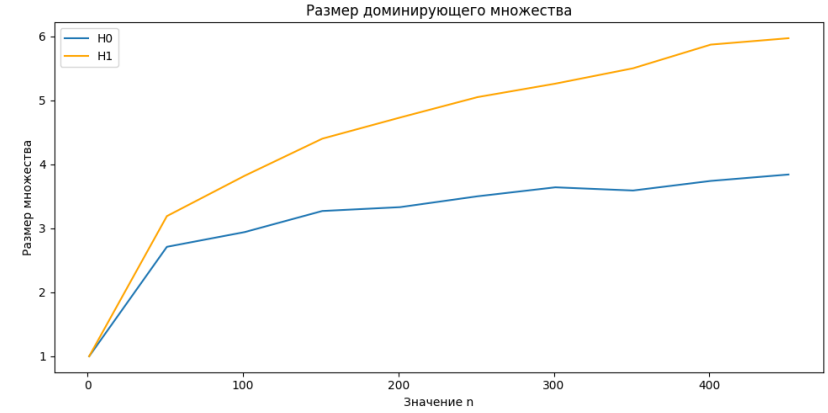
При dist=1.5 все характеристики хорошо разделяются, будем использовать его.

Посмотрим на зависимость характеристик от размера данных



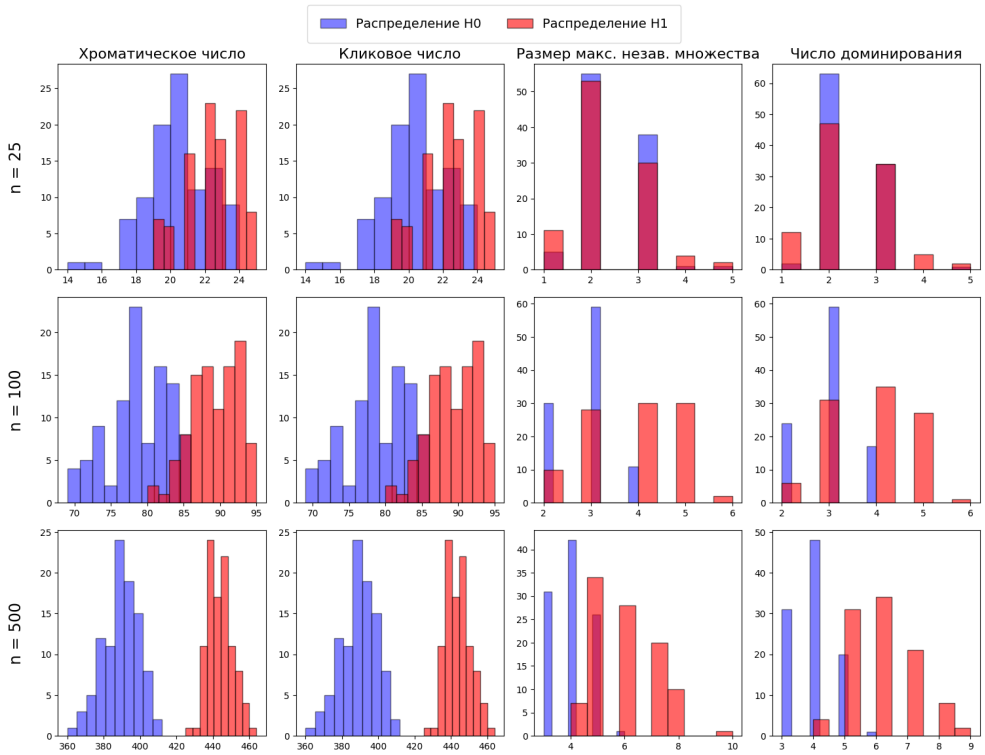






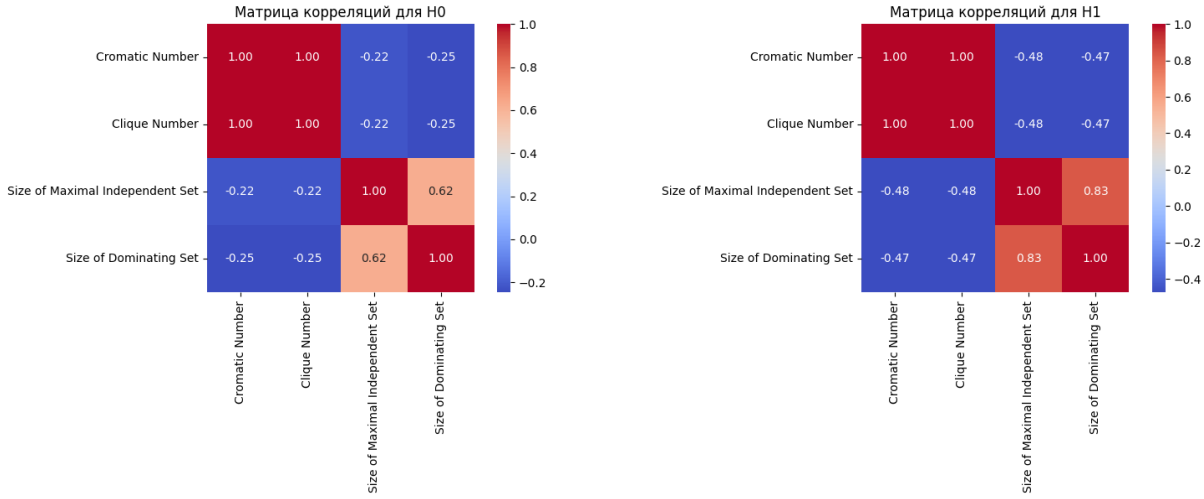
Как мы видим, все характеристики растут при увеличении n, при этом при фиксированном n для H1 каждая характеристика всегда больше, чем для H0.

Посмотрим, как хорошо разделяются наши характеристики при размерах выборки 25, 100 и 500



При данных размера 25 характеристики отделяются не очень хорошо. Зато при большом размере данных разделимость характеристик повышается.

Посмотрим на корреляцию характеристик для обоих распределений



У хроматического числа и кликового числа корреляция равна 1, у размера макс. независимого множества и числа доминирования корреляция большая при обоих распределениях.

В дальнейших экспериментах и при построении классификаторов будем исользовать только хроматические число и размер макс. независимого множества.

Построение классификатора, experiments_2

Наш классификатор будет работать таким образом: по выборке строится дистанцированный граф, для него считается хроматическое число и размер максимального независимого множества, по ним и размеру выборки какой-то алгоритм классификации делает предсказание. Пусть принятие нулевой гипотеры – False, а первой – True. Тогда ошибка первого рода это $FPR = \frac{FP}{FP+TN}$, а мощность это $Recall = \frac{TP}{TP+FN}$

Сгенерируем данные с такими колонками:

Хроматическое число

Размер макс. независимого множества

Размер выборки

Тип распределения (0 или 1)

Разобьем данные на тренировочную и тестовую выборки.

3.2.1 Обучение алгоритмов классификации

Обучим алгоритмы KNN, LogReg и RandomForest на наших данных и сравним их метрики:

Таблица 1: Сравнение метрик классификаторов

Метрика	**KNN**	**LogReg**	**Random Forest**
FPR	0.0317	0.0312	0.0342
Recall	0.9754	0.9546	0.9700
Accuracy	0.9719	0.9617	0.9679

Лучше всего работает KNN. FPR чуть лучше у LogReg, но у нее сильно хуже Recall. Далее по обученной модели KNN построим pipeline, который будет принимать на вход уже список точек, а не список характеристик. Оценим его на выборках размера 25, 100 и 500:

Таблица 2: Сравнение метрик для разных параметров

Метрика	**25**	**100**	**500**
FPR	0.2550	0.0650	0.0100
Recall	0.7050	0.9500	1.0000
Accuracy	0.7250	0.9425	0.9950

Итого

Были проведены эксперименты, в которых изучалась зависимость характеристик от параметра построения графа. По ним выбрались лучшие параметры построения и характеристики, которые могут являться признаками классификации. Был сгенерирован набор данных, на которых были обучены три модели классификации. На основе лучшей модели построен pipeline, который делает предсказание на основе входного списка точек.