Классификация распределения с помощью случайных графов

Соколовский С.П., Григоренко М.Д.

Дата: 28 мая 2025 г.

Предисловие

Договоримся об обозначениях:

- n — размер вектора реализаций случайной величины
- k, d — параметры построения KNN и дистанционного графов соответственно
- θ, v — параметры распределений
- T^{KNN}, T^{dist} — характеристики случайных графов

Часть I. Исследование свойств характеристики

Используемые инструменты Соколовского С.П.

Весь код в ветке `Crazy-Explorer31/first_part`, в директории `src/`:

- `graphs.py` — реализации KNN и дистанционного графов (построение и отрисовка)
- `characteristics_experimental.py` — функции для получения характеристик графов, построенных при данных параметрах (распределений, построения графов...). Самая важная — `get_average_characteristics`, возвращающая средние характеристики графов, построенных при переданных параметрах
- `characteristics_applied.py` — функции, возвращающие хар-ки данного графа
- `visualisations.py` — функции для удобного построения графиков
- `metrics.py` — функции, приближенно считающие ошибку I рода и мощность для данного $\mathcal{A}$. Считается по методу Монте-Карло, используя переданное в функцию множество точек (число компонент, хроматическое число), принадлежащих какому-то распределению.
- `classifier.py` — класс, генерирующий характеристики графов по данным параметрам и строящий $\mathcal{A}$ как *жадный рюкзак*

Используемые инструменты Григоренко М.Д.

Весь код в ветке `maxGrigorenko/first_part`, в директории `src/`:

- `graph_common_functions.py` — реализации KNN и дистанционного графов (у каждого есть метод для построения из значений случайной величины, а также методы вычисления характеристик)
- `distribution_functions.py` — функции для генерации выборки и вычисления матожидания характеристики методом Монте-Карло.

Шаг 1. Фиксируем n. Исследуем взаимосвязь между θ, v и T^{KNN}, T^{dist}

Результаты Соколовского С.П.

В файле `experiments_first_part_1.ipynb` происходит следующее:

- Для каждой тройки (распределение, тип графа, характеристика) перебираются параметры трех перечисленных объектов, после чего вычисляются характеристики полученных графов.
- Для каждой тройки строится диаграмма рассеивания, в которой по горизонтальной оси — параметр распределения, а по вертикальной — характеристика графа

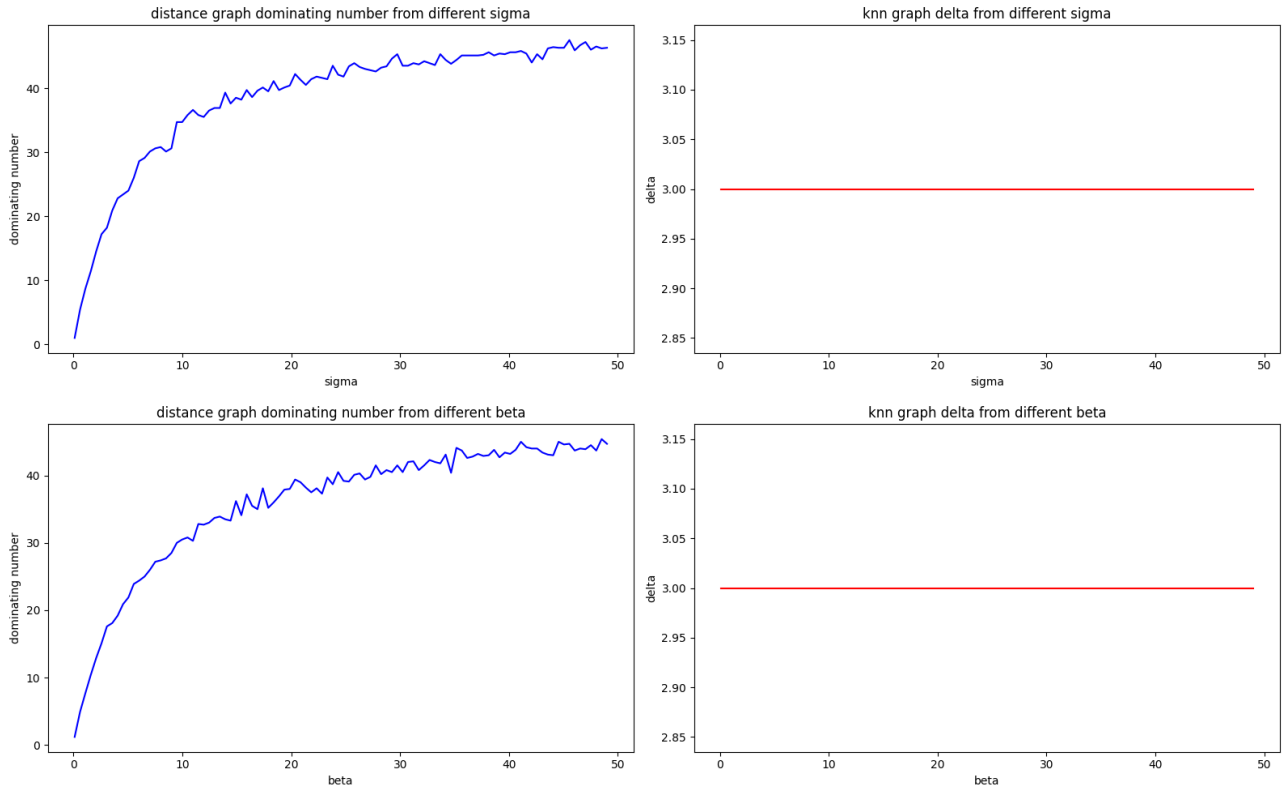
Из графиков заметно, что лишь с дистанционным графом хочется продолжать работать

Результаты Григоренко М.Д.

В файле `experiments_first_part_1.ipynb` происходит следующее:

- Реализованы функции `plot_sigma` и `plot_beta`, перебирающие значения соответствующих параметров распределений и выводящих график зависимости характеристики графов (knn и dist) от перебираемого параметра
- При фиксированном размере выборки проведены эксперименты с различными параметрами d и k.

В результате всех экпериментов delta графа knn была константной, то есть эта характеристика никак не связана с параметрами распределений. А вот доминирующее число дистанционного графа в среднем увеличивалось при увеличении параметра sigma. На двух графиках ниже показана зависимость среднего числа характеристик в завимости от параметров распределений:



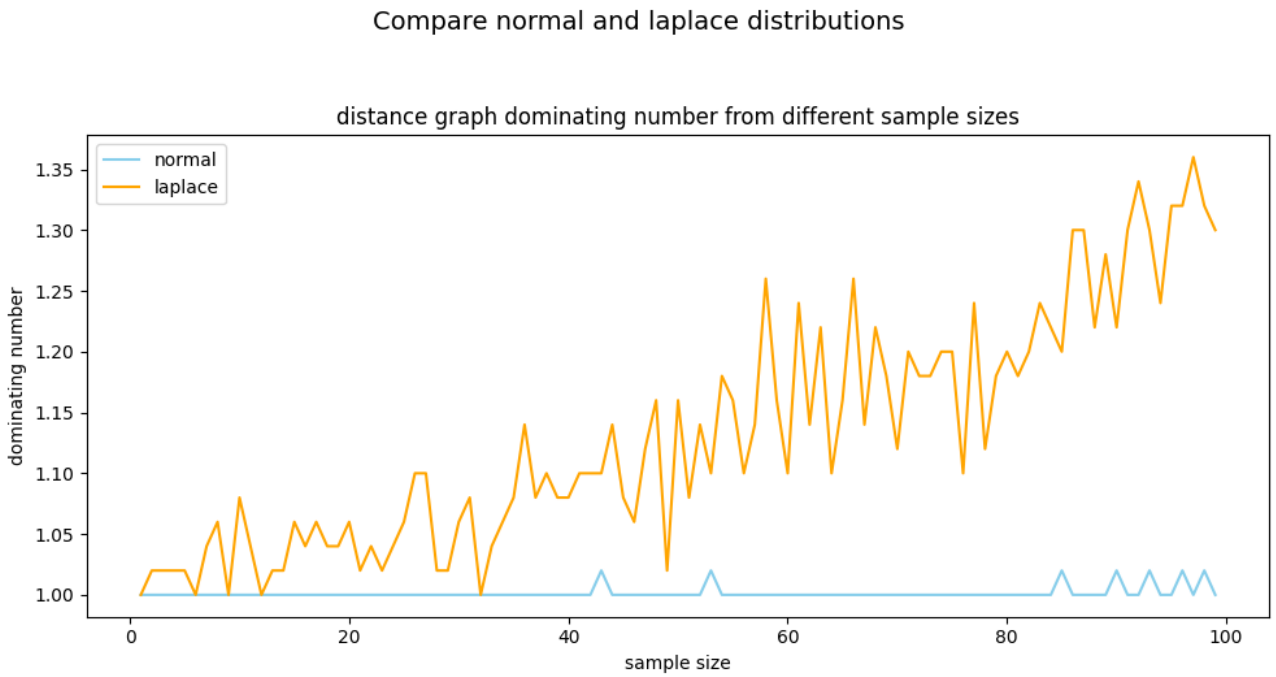
Шаг 2. Фиксируем θ, v. Исследуем взаимосвязь между n, k, d и T^{KNN}, T^{dist}

Результаты Соколовского С.П.

В файле `experiments_first_part_2.ipynb`, аналогично первому шагу, генерятся много налюдений для всех комбинаций распределений, типов графов, их характеристик. Далее на диаграммах рассеивания по оси Ox откладываются параметры построения графов, по Oy — их характеристики, и ещё цветом отражена, при каком n было получено наюлюдение. Выводы аналогичные первому эксперименту

Результаты Григоренко М.Д.

В файле `experiments_first_part_2.ipynb` зафиксированы параметры распределений и отрисованы графики зависимости характеристик графов от размера выборки. delta графа knn оказалась неинформативной характеристикой. А вот доминирующее число дистанционного графа немного по-разному меняется при изменении размера выборки, в особенности, если в качестве параметра дистанционного графа установить значение $d >= 3$, то характеристика графа из нормального распределения становится почти всегда равной 1, а вот при распределении Лапласа немного больше. Снизу график зависимости среднего числа доминирования от размера выборки при $d = 3.5$:



Шаг 3. Фиксируем θ, v. Строим $\mathcal{A}$ для переданного n

Результаты Соколовского С.П.

Файл `experiments_first_part_3.ipynb` поделен на два раздела. В первом фиксируются все параметры и строится $\mathcal{A}$. Во втором рассуждения, изложенные в первом разделе обобщаются, и приведена реализация класса, строящая $\mathcal{A}$ по переданному в конструктор n Используется следующий алгоритм построения $\mathcal{A}$:

1. Строятся точки с координатами (число компонент, хроматическое число) по генерирующимся векторам случайных величин
2. За изначальное $\mathcal{A}$ берется множество всех сгенерированных точек, полученных по первому распределению (Exp).
3. Далее пытаемся удалить точку из $\mathcal{A}$ так, чтобы ошибка I рода не превысила 0.05, а мощность была максимальной (ошибка I рода и мощность считаются на основе точек, сгенерированных в начале). Для этого перебираем все варианты и выбираем наилучший
4. Пытаемся так удалить что-то из $\mathcal{A}$ много раз
5. В итоге получаем искомое $\mathcal{A}$

Результаты Григоренко М.Д.

В файле `experiments_first_part_3.ipynb` реализован алгоритм конструирования множества $\mathcal{A}$, которое должно удовлетворять двум условиям:

1. Контроль ошибки первого рода: вероятность ошибочно отвергнуть нулевую гипотезу H_0 (данные имеют нормальное распределение) при её справедливости не превышает $\alpha = 0.05$.

2. Максимизация мощности: вероятность корректно отвергнуть H_0 в пользу альтернативы H_1 (например, распределение Лапласа) должна быть максимальной.

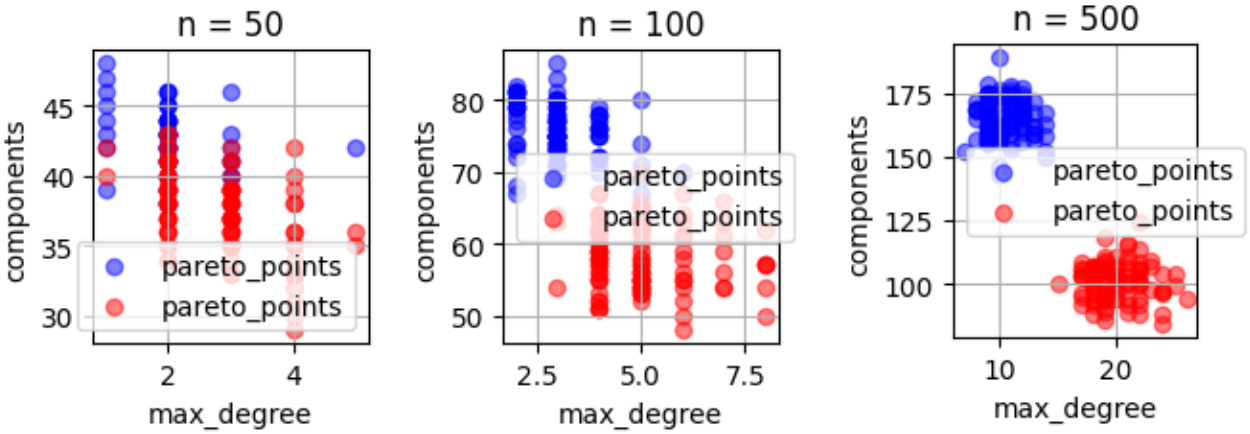
Множество $\mathcal{A}$ конструируется итеративно следующим образом: На каждом шаге генерируется большое число выборок (number_of_experiments) из нормального распределения. Для каждой выборки вычисляется характеристика графа. Если значение характеристики не принадлежит текущему множеству $\mathcal{A}$, оно считается "ошибочным"(ложным отклонением H_0). Ошибка первого рода оценивается как доля таких "ошибочных"случаев: Пока ошибка err $> \alpha$, в $\mathcal{A}$ добавляется наиболее частое значение характеристики из "ошибочных"результатов (мода). Это снижает долю ошибок за счёт включения типичных для H_0 значений. Процесс останавливается, когда err $\leq \alpha$, либо когда "ошибочные"значения исчерпаны.

Часть II. Несколько характеристик проверки гипотезы

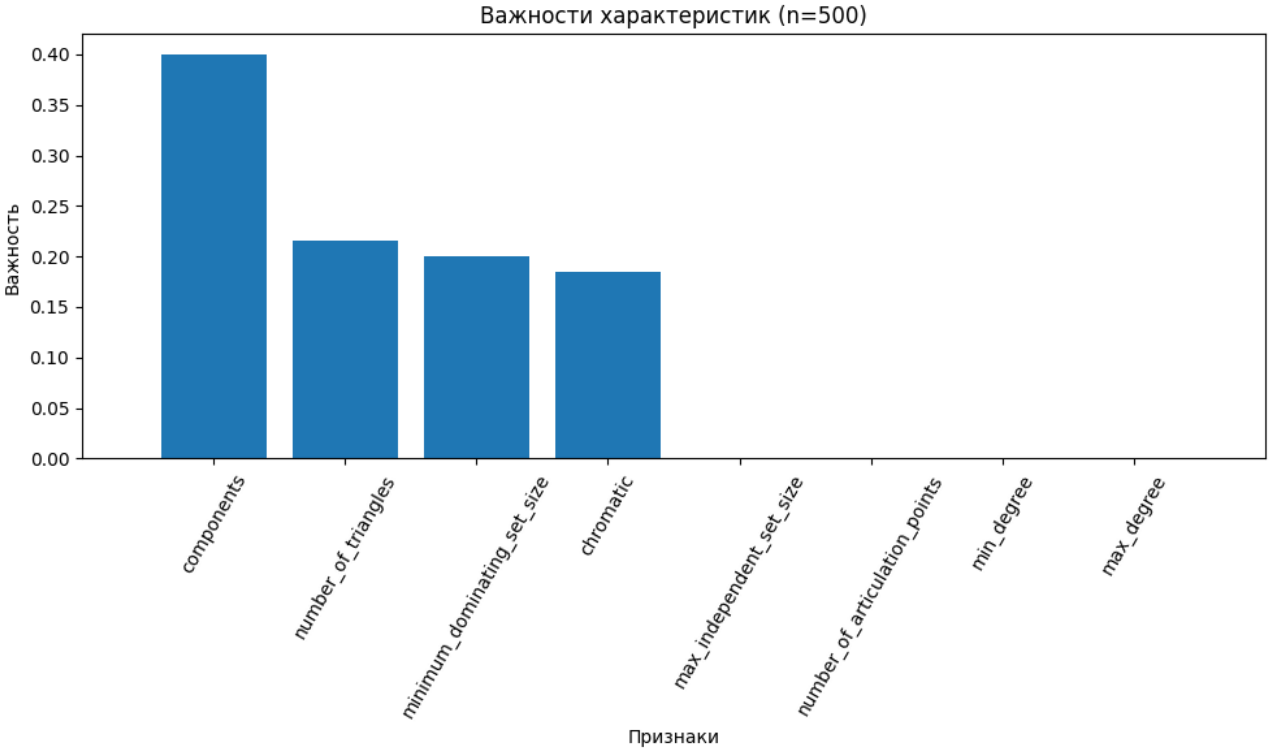
Шаг 1. Исследуем важность характиристик

Результаты Соколовского С.П.

В файле `experiments_second_part_1.ipynb` исследование важности характеристик состоит из визуальной и аналитической частей. В визуальной с помощью функции `describe_features_importances` для разных n построены матрицы скатерплотов, изображающих взаиморасположение точек с координатами, равными характеристикам графов, соответствующим векторам чисел из экспоненциального и Парето распределений. Заметно, что при росте n граница между точками разного характера становится очевиднее.



В аналитической части важность признаков изучается с помощью случайного леса: для разных n на сгенерированных данных обучается случайный лес (с перебором гиперпараметров), а затем с помощью встроенного метода класса случайного леса из sklearn достаются важности признаков. Ниже представлена гистограма важностей признаков для $n = 500$.



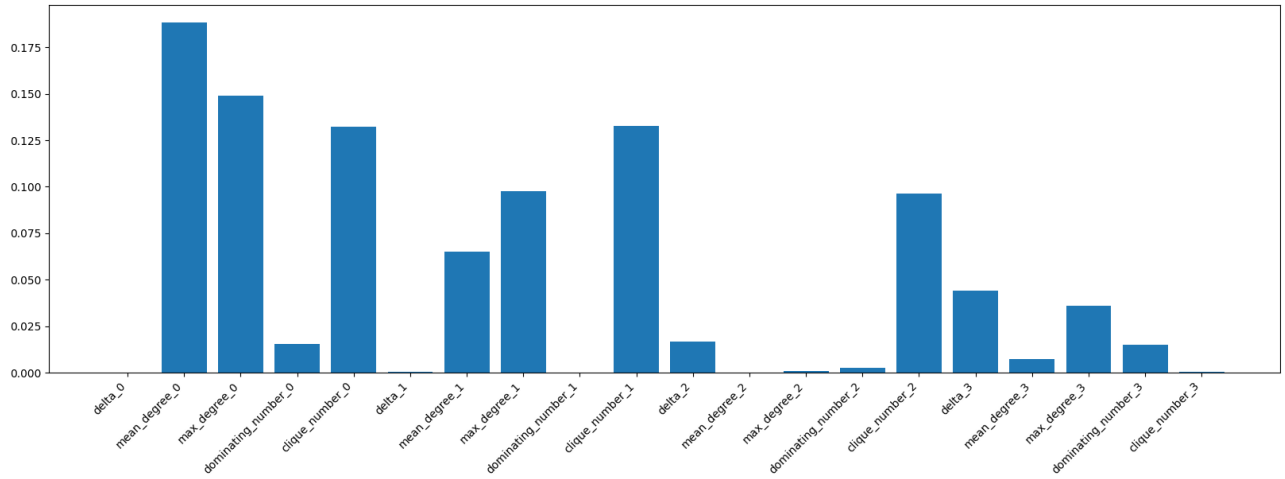
При обучении моделей буду использовать именно эти четыре характеристики: число компонент, число треугольников, размер наименьшего доминирующего множества, хроматическое число.

Результаты Григоренко М.Д.

В файле `experiments_second_part.ipynb` написан класс `DistribituionClassifier`, принимающий на вход параметр n - размер выборки и модель классификации, которую предстоит обучить. Для выявление признаков по данной выборке строится 4 дистанционных графа с различным параметром d, для каждого графа считаеются следующие характеристики:

1. Минимальная степень вершины
2. Средняя степень вершины
3. Максимальная степень вершины
4. Число доминирования
5. Кликовое число

Важность характеристик слабо меняется при разных n. Самыми важными характеристиками оказались средняя и максимальная степени дистанционного графа при $d = 0.3$ (самое маленькое значени d). Ниже предствлена гистограмма важности признаков при $n = 500$:

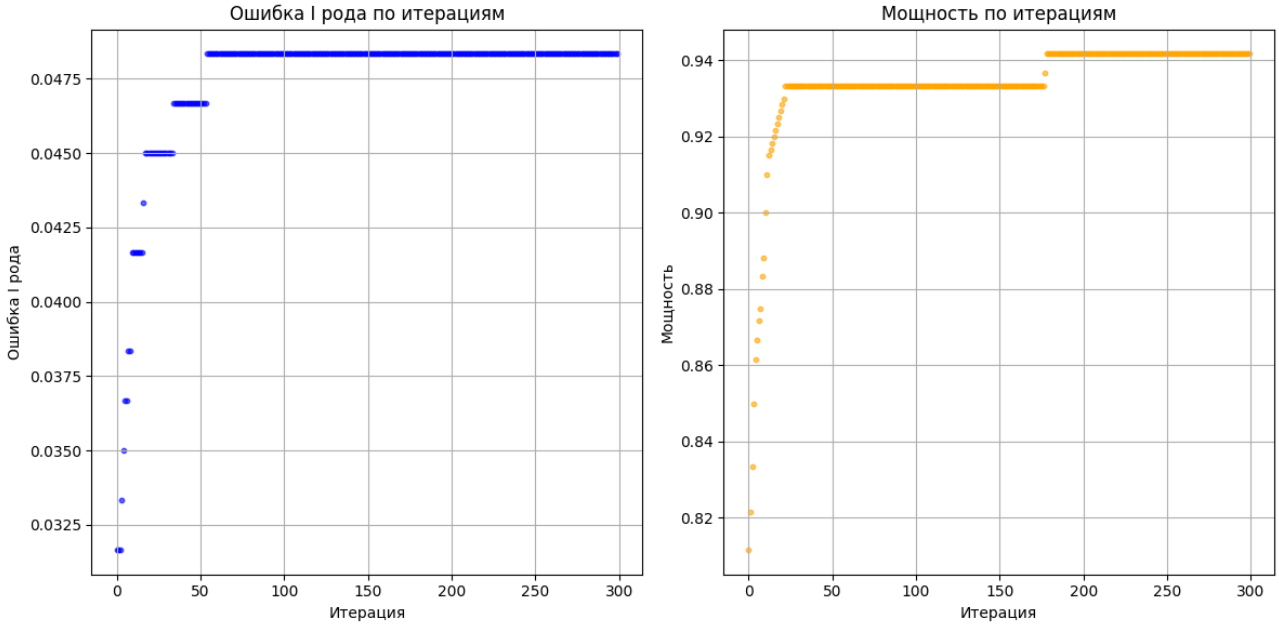


Шаг 2. Обучаем разные модели и мерим качество для разных размеров $\hat{\xi}$

Результаты Соколовского С.П.

Всё в файле `experiments_second_part_2.ipynb`. Для данной задачи были опробованы следующие модели: жадная (формирующая $\mathcal{A}$ как в задаче про рюкзак), случайный лес, логистическая регресия. Пройдемся по каждой:

1. Реализация в `classifier.py`. Жадник работал следующим образом: сперва генерировались характеристики (далее — точки с их координатами) для графов разных распределений, затем $\mathcal{A}$ инициализировалось как множество уникальных точек экспоненциального распределения, и, наконец, в цикле, пока ошибка I рода не превышает 0.05, из $\mathcal{A}$ выкидывалась точка так, чтобы максимально увеличить мощность. Отмечу, что здесь за $\mathcal{A}$ берется не множество точек, а их выпуклая оболочка (статистический критерий — принадлежность данной точки этой выпуклой оболочке). Из-за вычислительной сложности, качественно удалось обучить лишь для $n = 75$



n	Accuracy	Precision	Recall	F1 Score	ROC AUC	I_error	Power
75	0.9375	0.9397	0.935	0.9373	0.9375	0.0483	0.9417

Метрики модели рюкзака

2. По аналогии с предыдущим шагом был обучен случайный лес. Метрики у него тем более выдающиеся, чем больше n. Однако, как в предыдущем шаге, контролировать ошибку I рода нельзя (но при больших n необходимость в этом отпадает)

n	Accuracy	Precision	Recall	F1 Score	ROC AUC	I_error	Power
50	0.8750	0.8866	0.8600	0.8731	0.8750	0.1100	0.8600
100	0.9850	0.9802	0.9900	0.9851	0.9850	0.0200	0.9900
500	1.0000	1.0000	1.0000	1.0000	1.0000	0.0000	1.0000

Метрики модели случайного леса

3. Почти такого же качества получилась модель логистической регрессии. Однако преимущество её над случайным лесом в том, что, варьируя её пороговое значение, мы можем контролировать ошибку I рода (пока не актуально — работает хорошо из коробки)

n	Accuracy	Precision	Recall	F1 Score	ROC AUC	I_error	Power
50	0.8575	0.8865	0.8200	0.8519	0.8575	0.1050	0.8200
100	0.9850	0.9899	0.9800	0.9849	0.9850	0.0100	0.9800
500	1.0000	1.0000	1.0000	1.0000	1.0000	0.0000	1.0000

Метрики модели логистической регрессии

Результаты Григоренко М.Д.

Для каждого n (размер выборки) были обучены классификаторы `RandomForestClassifier`, с подобранными гиперпараметрами `max_depth` и `min_samples_split`, и `BaggingClassifier` с подобранным гиперпараметром `n_estimators` (из библиотеки `sklearn`). Для всех выборок чуть лучше (или примерно одинаково) оказался классификатор `RandomForestClassifier`. Лучшие метрики (за положительный класс взято H_1 (распределение Лапласа)):

1. $n = 25$: `accuracy = 0.77, precision = 0.79, recall = 0.74.`. Измерение проводилось на 2000 примерах.
2. $n = 100$: `accuracy = 0.95, precision = 0.94, recall = 0.96`. Измерение проводилось на 200 примерах.
3. $n = 500$: `accuracy = 1.0, precision = 1.0, recall = 1.0`. Измерение проводилось на 200 примерах (из-за сложности построения дистанционного графа на 500 вершинах).

Шаг 3. Выводы о вероятности ошибки первого рода и мощности подхода

Результаты Соколовского С.П.

В качестве исследуемой архитектуры взята логистическая регрессия (потому что она лучше рюкзака и всё ещё можно контролировать ошибку I рода). Измерение метрик проводилось так: для каждого $n \in \{50, 100, 500\}$ обучалась модель выбранной архитектуры, 10 раз измерялись её метрики на новых сгенерированных точках, а затем брались статистики по полученным метрикам. При $n = 50$, для модели пришлось взять threshold равным 0.8, чтоб ошибка I рода была небольшая. Ниже приведены результаты этих измерений

n = 50		
	I_error	Power
min	0.019868	0.6150
mean	0.041746	0.6705
max	0.058824	0.7400

n = 100		
	I_error	Power
min	0.000000	0.945
mean	0.014645	0.969
max	0.029557	0.985

n = 500		
	I_error	Power
min	0.0	1.0
mean	0.0	1.0
max	0.0	1.0

Статистики по метрикам логистической регрессии по разным n

Результаты Григоренко М.Д.

Мощность критерия по определению равна метрике `recall`. Вероятность ошибки первого рода была измерена отдельно:

1. $n = 25$: `error = 0.22`. Измерение проводилось на 2000 примерах.
2. $n = 100$: `error = 0.055`. Измерение проводилось на 2000 примерах.
3. $n = 500$: `error = 0.0`. Измерение проводилось на 200 примерах (из-за сложности построения дистанционного графа на 500 вершинах). Таким образом, можно оценить ошибку сверху `error < 0.01`.

Заключение

В ходе работы были успешно подобраны модели для классификации распределения случайной величины по данному вектору из её реализаций:

- Для классификации *Нормальное* $\leftrightarrow$ *Лаплас* — `RandomForest`
- Для классификации *Экспоненциальное* $\leftrightarrow$ *Парето* — `LogisticRegression`

Также было замечено, что при увеличении n качество модели повышается, так как характеристики графов разной природы становятся более "непохожими".
Данная работа свидетельствует о том, что такие абстрактные объекты, как случайные графы, имеют хорошее практическое применение. В данном случае — при проверке гипотезы согласия