Отчет по проекту: Задачи по случайным графам

Бахурин Виктор и Стахова Екатерина

30 мая 2025 г.

Содержание

1 Введение

Часть I. Исследование свойств характеристики

2 Описание кода

2.1 Используемые инструменты

- Язык программирования: Python 3.10
- Основные библиотеки: numpy, networkx, matplotlib, scikit-learn
- Система контроля версий: Git (GitHub/GitLab)
- Дополнительные инструменты: Jupyter Notebook, PyCharm, Google Colab

2.2 Реализованные классы

2.2.1 MyClassifier()

- **Назначение**: Простая модель бинарного классификатора по двум признакам.
- **Встроенные методы**: $fit(X\_train, y\_train)$ и $predict(X\_test)$

2.3 Реализованные алгоритмы

2.3.1 $fast\_chromatic\_number()$

- **Назначение**: Вычисление хроматического числа для случайного графа построенного на данной выборке.
- **Входные данные**: list - выборка
- **Выходные данные**: int - хроматическое число
- **Сложность**: O(nlog(n))

2.3.2 $greedy()$

- **Назначение**: Жадное построение множества A, максимизирующие мощность критерия, при заданной допустимой ошибки первого рода.
- **Входные данные**: T_H_0, T_H_1, α - два набора наблюдений и максимальная допустимая ошибка первого рода.
- **Выходные данные**: A, $current\_error$, power - множество A, ошибка первого рода, мощность критерия.
- **Сложность**: O(nlog(n))

3 Описание экспериментов

3.1 Эксперимент 1

3.1.1 Цель

Исследовать, как ведет себя числовая характеристика T в зависимости от параметров распределений и , зафиксировав размер выборки и параметр процедуры построения графа KNN.

3.1.2 Результаты

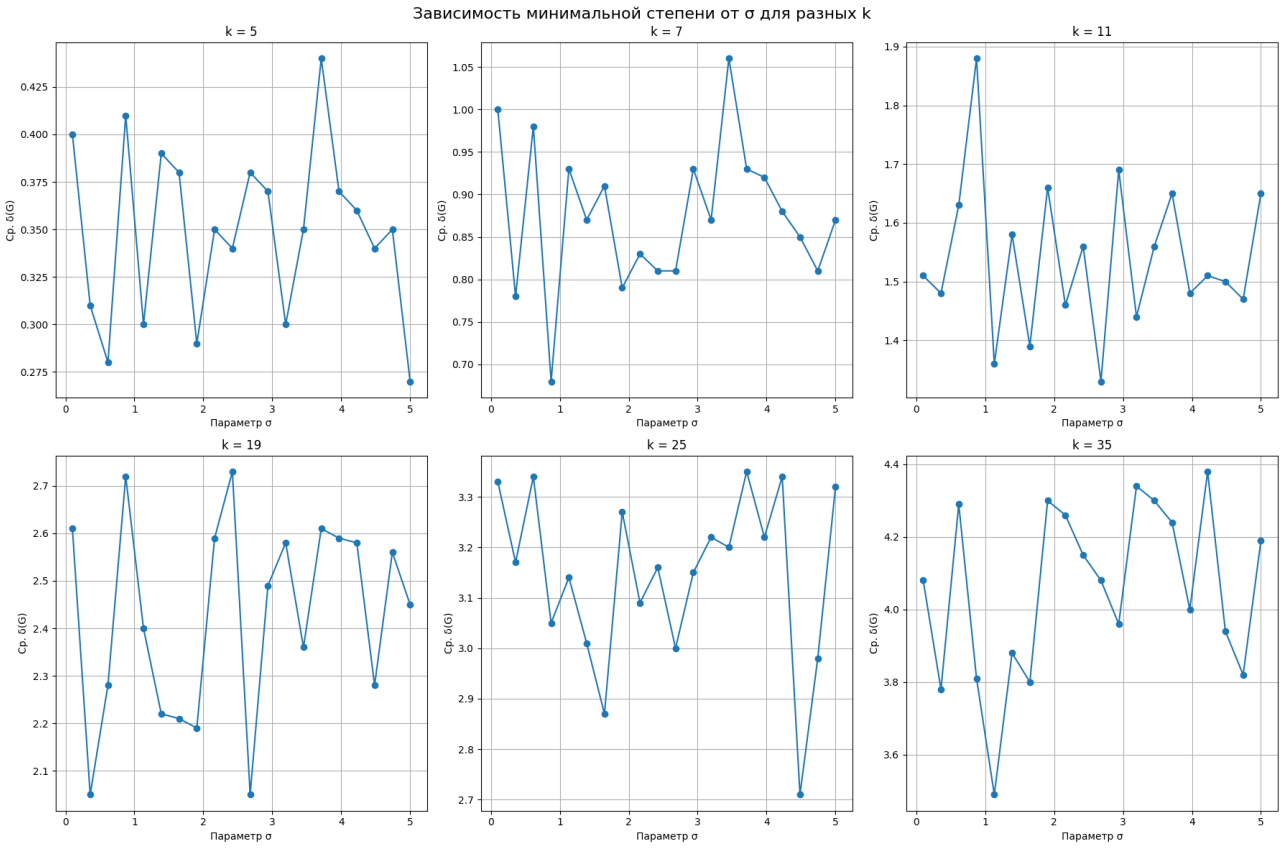

Рис. 1: $E[in\_\delta(G)]$ для KNN графа построенного на $Normal(0, \sigma)$

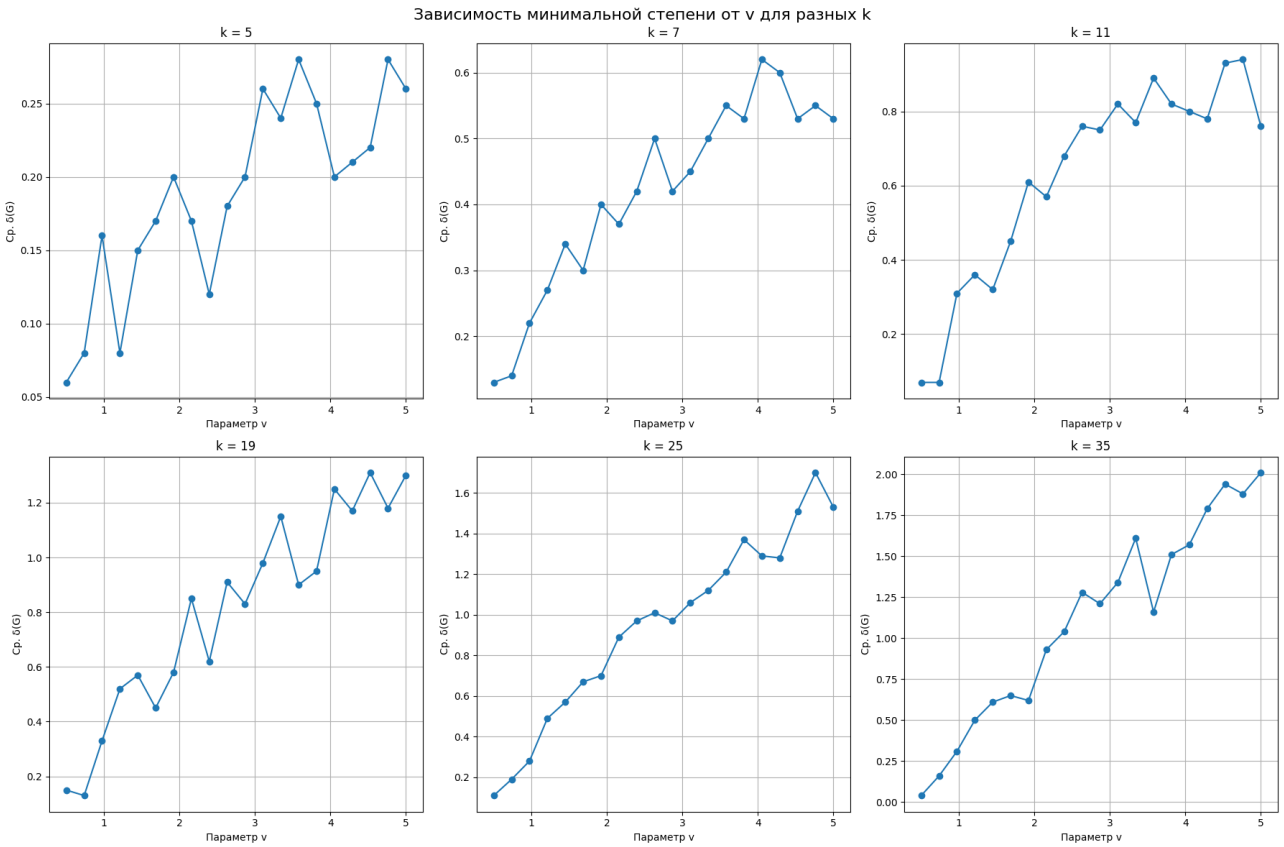

Рис. 2: $E[in\_\delta(G)]$ для KNN графа построенного на $Student - t()$

Мы получили интересный результат. График для нормального распределения выглядит хаотичнее, чем график для Student-t(); в графике Student-t() прослеживается рост $E[in\_\delta(G)]$ с ростом параметра . И еще одно интересное наблюдение: для интересующих нас параметров распределений v_0 и $_0$ график распределения Student-t() ниже графика нормального распределения.

3.2 Эксперимент 2

3.2.1 Цель

Исследовать, как ведет себя числовая характеристика T в зависимости от параметров распределений и , зафиксировав размер выборки и параметр процедуры построения графа dist.

3.2.2 Результаты

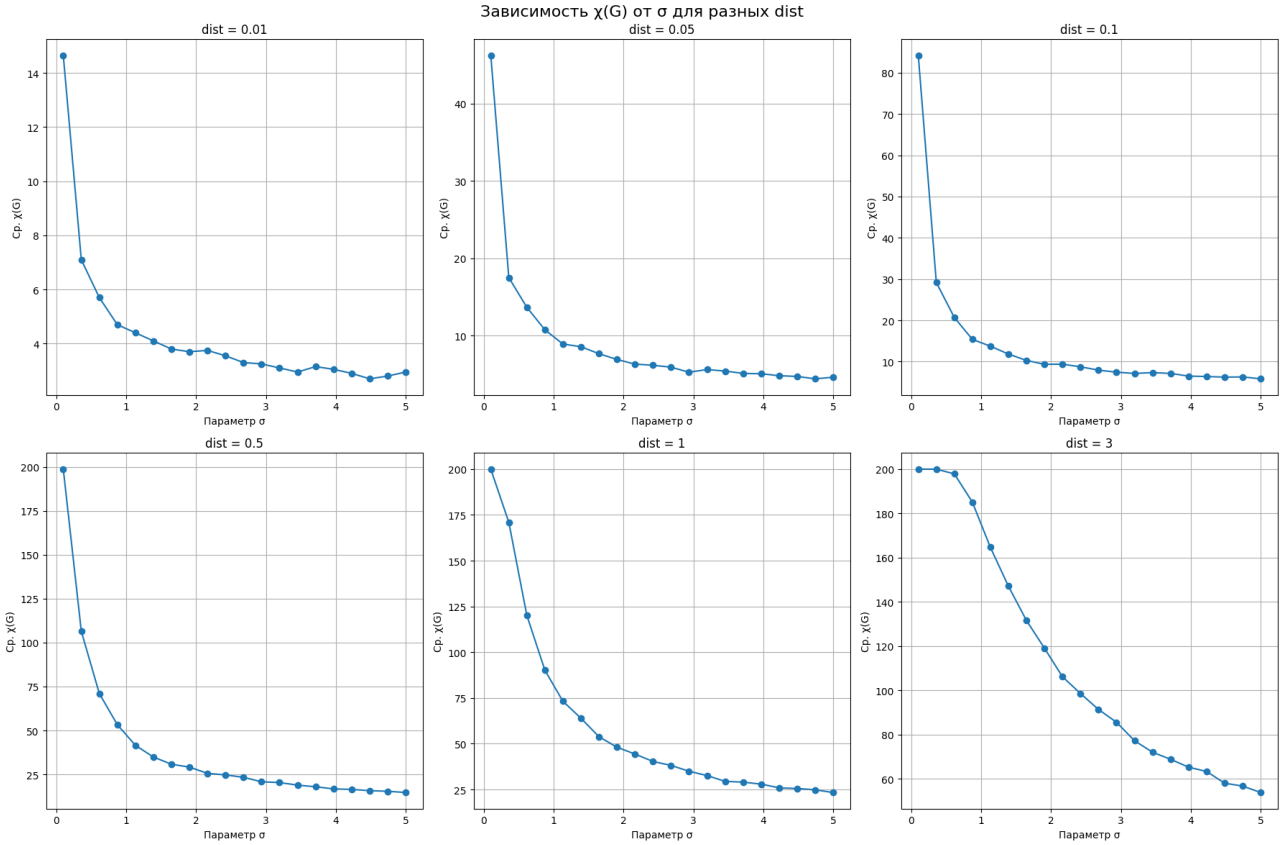

Рис. 3: $E[\chi(G)]$ для dist графа построенного на $Normal(0, \sigma)$

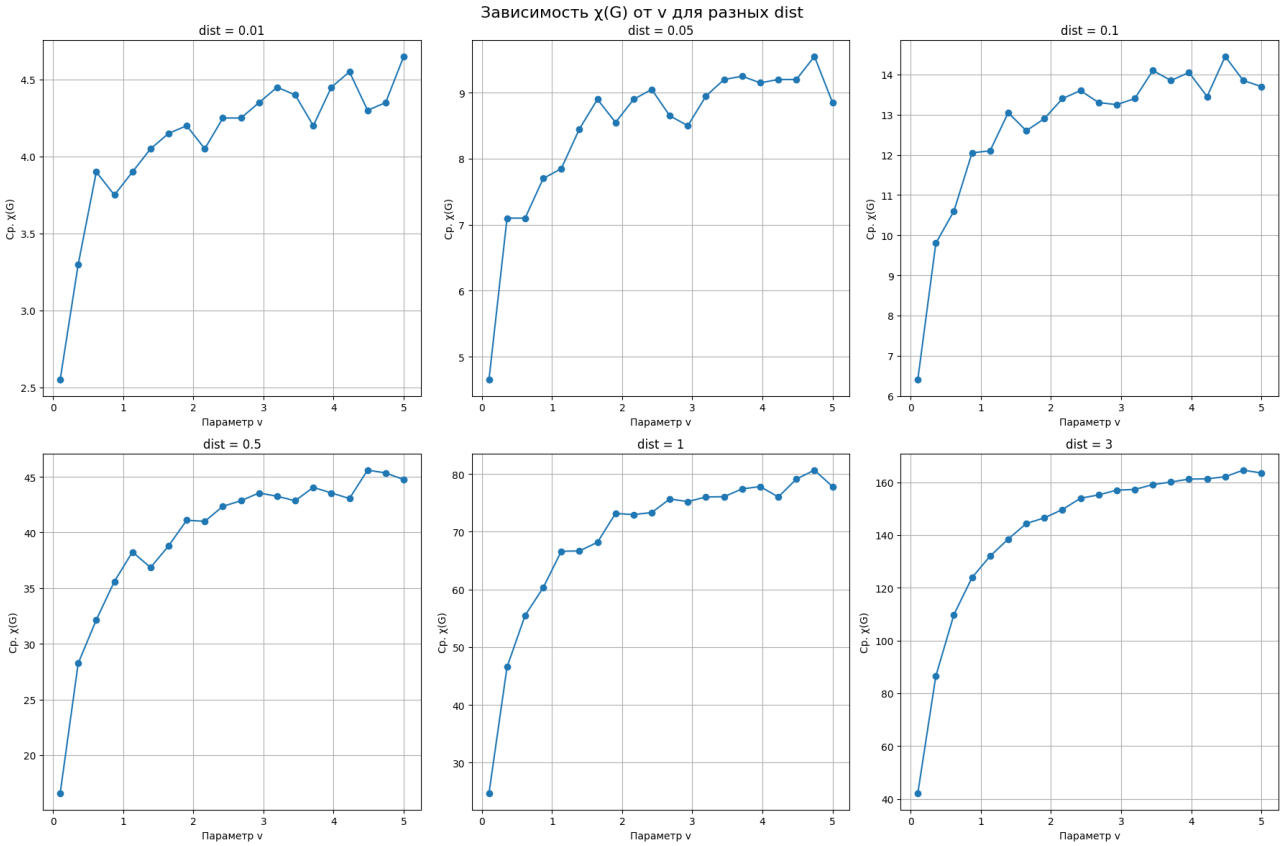

Рис. 4: $E[\chi(G)]$ для dist графа построенного на $Student - t()$

Характеристика (G) на дистанционном графе показывает разные результаты для разных выборок. Для нормального распределения с ростом параметра σ хроматическое число убывает, а для распределения Student-t() с ростом параметра v $\chi(G)$ наоборот растет.

3.3 Эксперимент 3

3.3.1 Цель

Исследовать, как ведет себя числовая характеристика T в зависимости от параметров процедуры построения графа KNN и размера выборки при фиксированных значениях $\theta = \theta_0$ и $v = v_0$.

3.3.2 Результаты

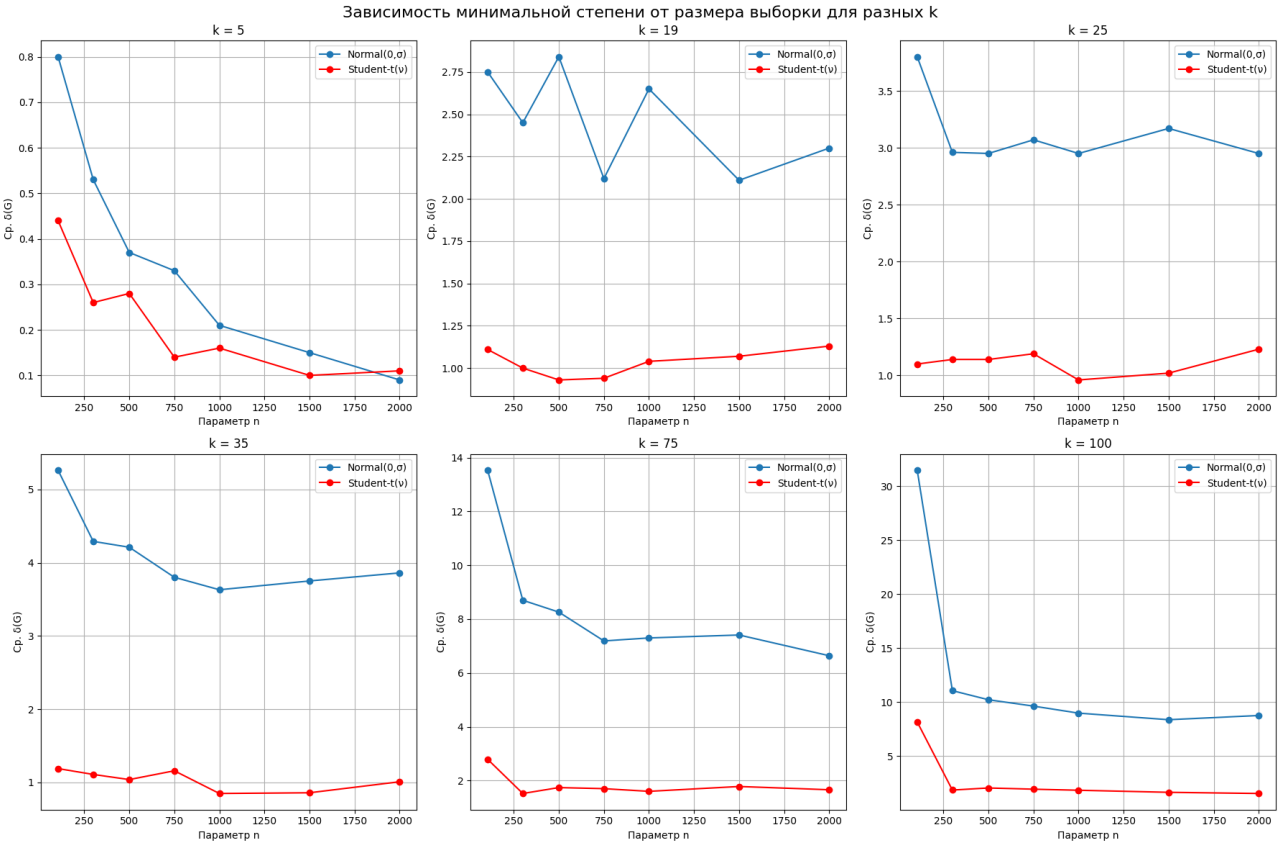

Рис. 5: $E[in\_\delta(G)]$ для KNN графа

График для Normal выше, чем график для Student. Это может помочь в проверке истинности H_0 и H_1.

3.4 Эксперимент 4

3.4.1 Цель

Исследовать, как ведет себя числовая характеристика T в зависимости от параметров процедуры построения графа и размера выборки при фиксированных значениях $\theta = \theta_0$ и $v = v_0$.

3.4.2 Результаты

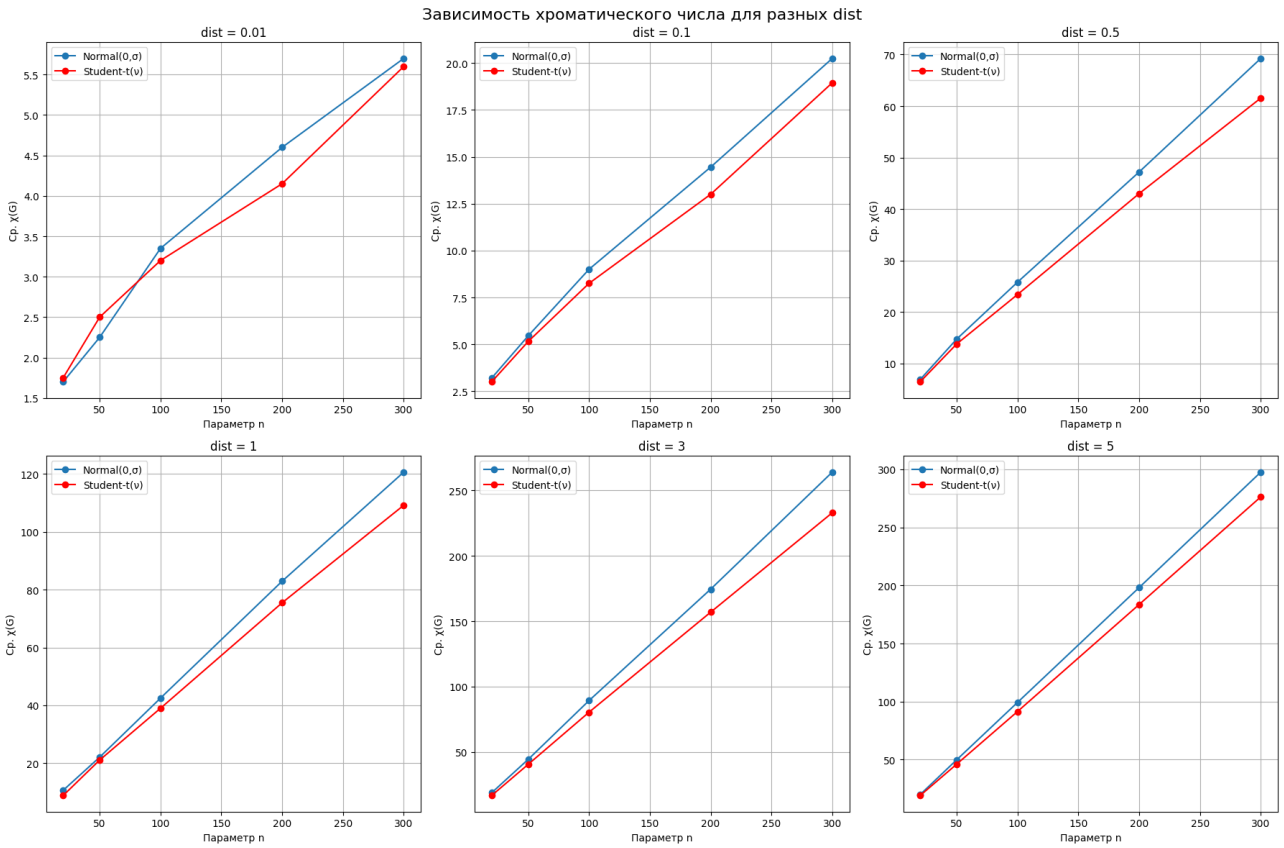

Рис. 6: $E[\chi(G)]$ для dist графа

К сожалению, данные графики не сильно отличаются, в среднем график для Student-t() ниже, чем график $Normal(0, \sigma)$.

3.5 Промежуточный вывод

Если обобщить результаты, полученные в предыдущих пунктах, то можно заметить, что каждая из характеристик показывает разные значения на случайных графах, построенных на распределениях $Student - t()$ и нормальном распределении $Normal(0, \sigma)$. Это означает, что существует возможность использовать их для проверки истинности гипотез H_0 и H_1.

3.6 Эксперимент 5

3.6.1 Цель

Построить множество A в предположении $\theta = \theta_0$ и $v = v_0$ при максимальной допустимой вероятности ошибки первого рода $\alpha = 0.055$. Оценить мощность полученного критерия.

3.6.2 Результаты

Для каждой характеристики удалось построить множество A.
Используя характеристику $in\_\delta(G)$ на графе KNN получен следующий результат:
Ошибка первого рода $\alpha = 0.035$. Мощность полученного критерия 0.717.
Используя характеристику $\chi(G)$ на графе dist получен следующий результат:
Ошибка первого рода $\alpha = 0.045$. Мощность полученного критерия 0.594.
В первом случае результат значительно лучше.

3.7 Эксперимент 6

3.7.1 Цель

Исследование важности характеристик, как признаков классификации. Узнать, меняется ли важность характеристик с ростом n.
В качестве графа был выбран dist, так как он неориентированный.

3.7.2 Результаты

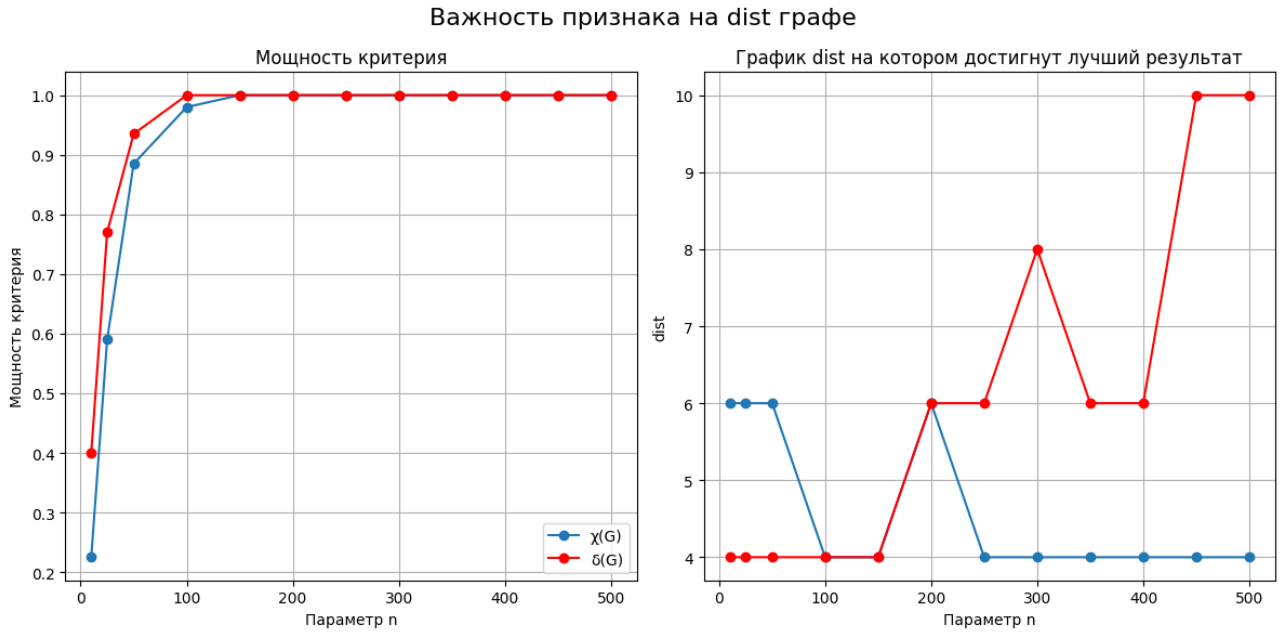

Рис. 7: Важность признаков на dist графе

Мы видим, что каждый из признаков дает хорошую мощность критерия на значениях n от 50. При этом мы также видим значение dist, на котором было получено лучшее значение мощности критерия.

3.8 Эксперимент 7

3.8.1 Цель

Применить разные классификационные алгоритмы и оценить метрики качества. В качестве классификаторов были выбраны следующие: LogisticRegression, RandomForestClassifier и MyClassifier.

3.8.2 Результаты

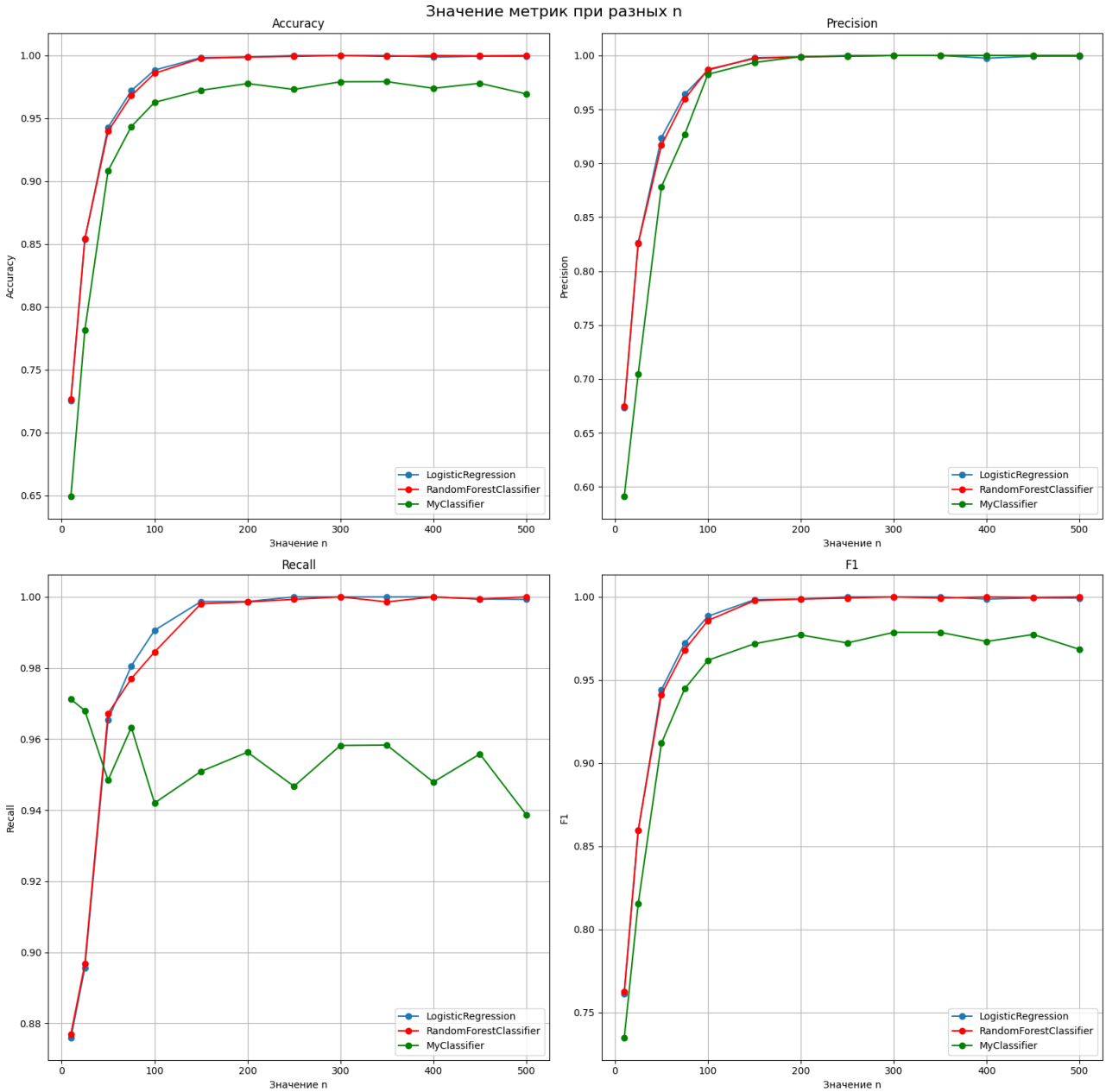

Рис. 8: Значения метрик

Мы видим, что используя обе характеристики, мы получаем точность 95% уже при n=75. Я думаю, что это хороший результат. Стоит отметить, что мой классификатор показывает не очень хорошую, относительно других классификаторов, метрику recall, это вызвано его реализацией: в случае, когда характеристики указывают на разные гипотезы, я выбираю

ответ равновероятно, это и дает данную погрешность. В остальных метриках классификаторы ведут себя схоже: RandomForestClassifier чуть лучше, LogisticRegression чуть хуже, MyClassifier еще чуть хуже.

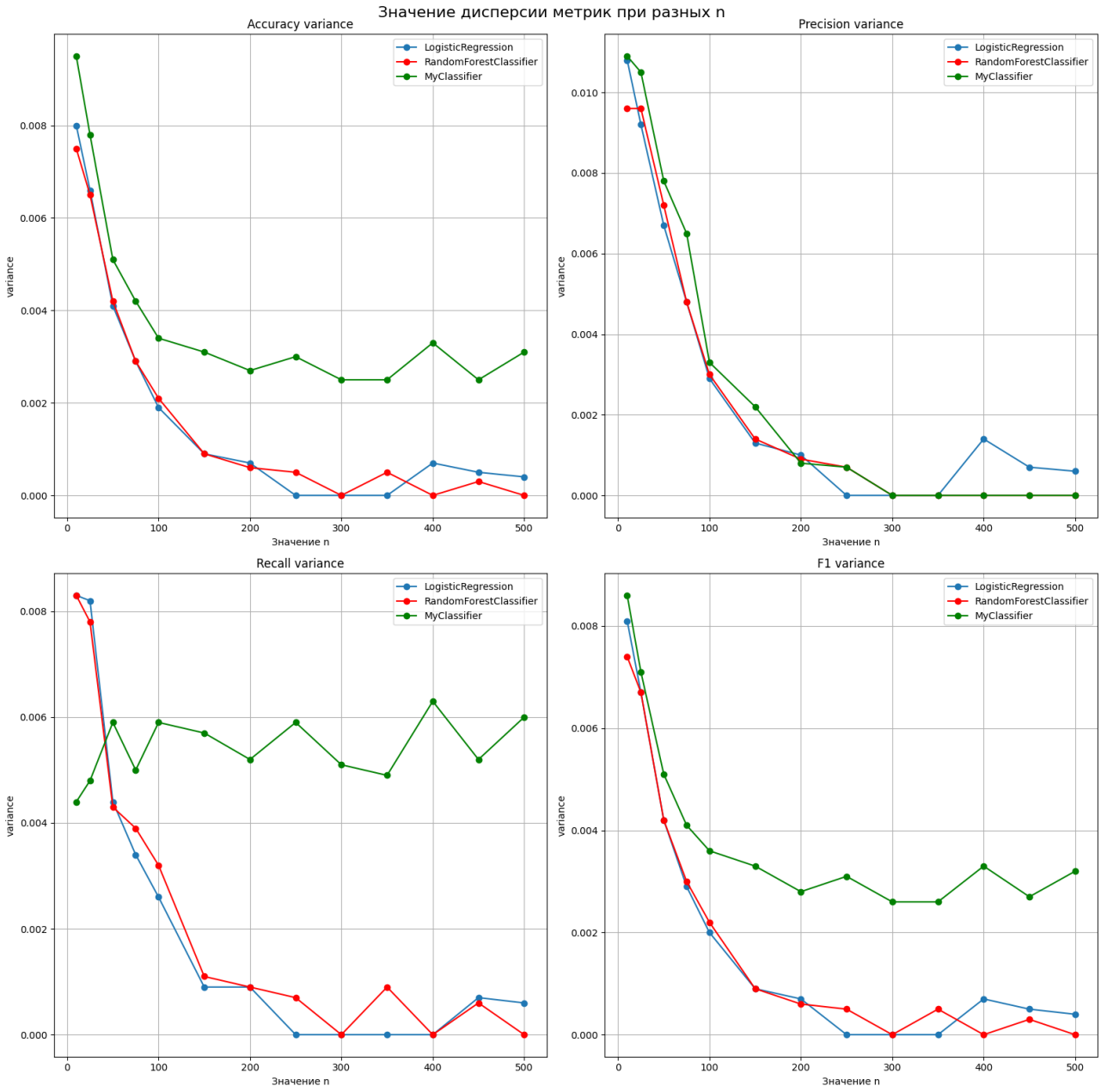

Рис. 9: Значения дисперсий метрик

Посмотрим на дисперсию метрик, для всех метрик и классификаторов верно, что коэффициент вариации находится в диапазоне от 10% до 0% и уменьшается с ростом n (исключение метрика recall и MyClassifier, но причину этого я уже описывал выше). Хороший ли это показатель, зависит от задачи.

3.9 Оценка собственного классификатора

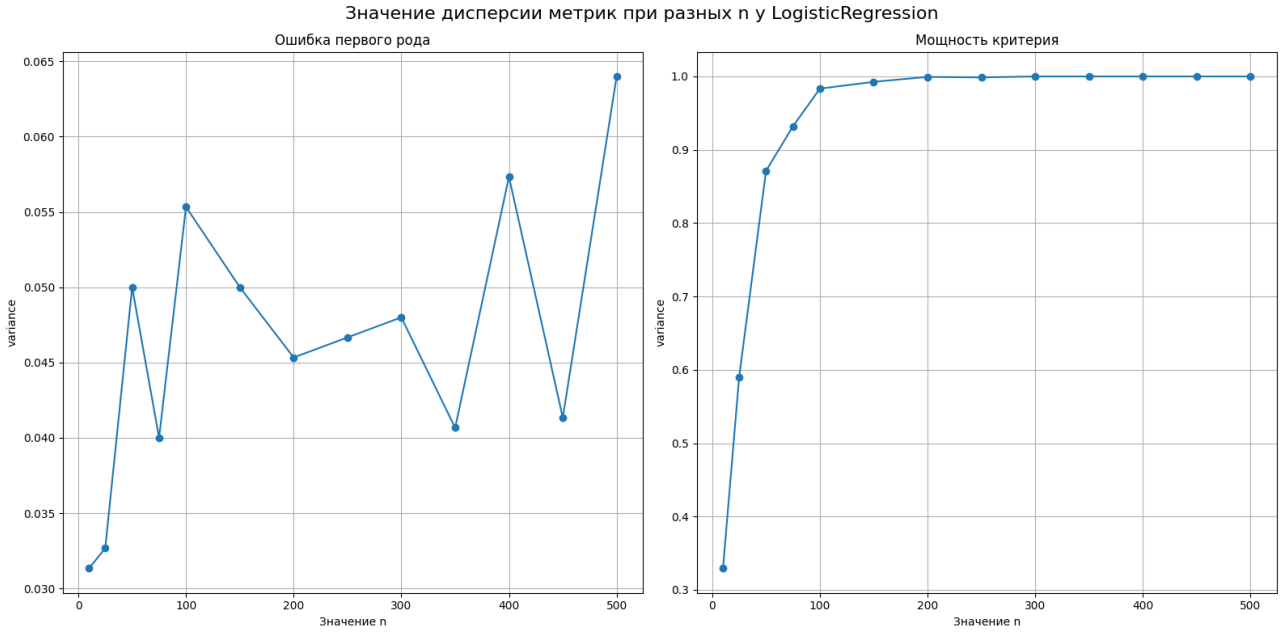

Рис. 10: Ошибка первого рода и мощность критерия

По графику заметно, что при значении ошибки первого рода в диапазоне 0.055 мы получили хорошие результаты мощности критерия при n от 50, мощность критерия в районе 90%. Я считаю, что для двух характеристик это неплохой результат. Возможно, если реализовывать классификатор другим способом, можно добиться лучшего(например, если руками реализовать RandomForestClassifier), но для столь наивной реализации результат неплох.