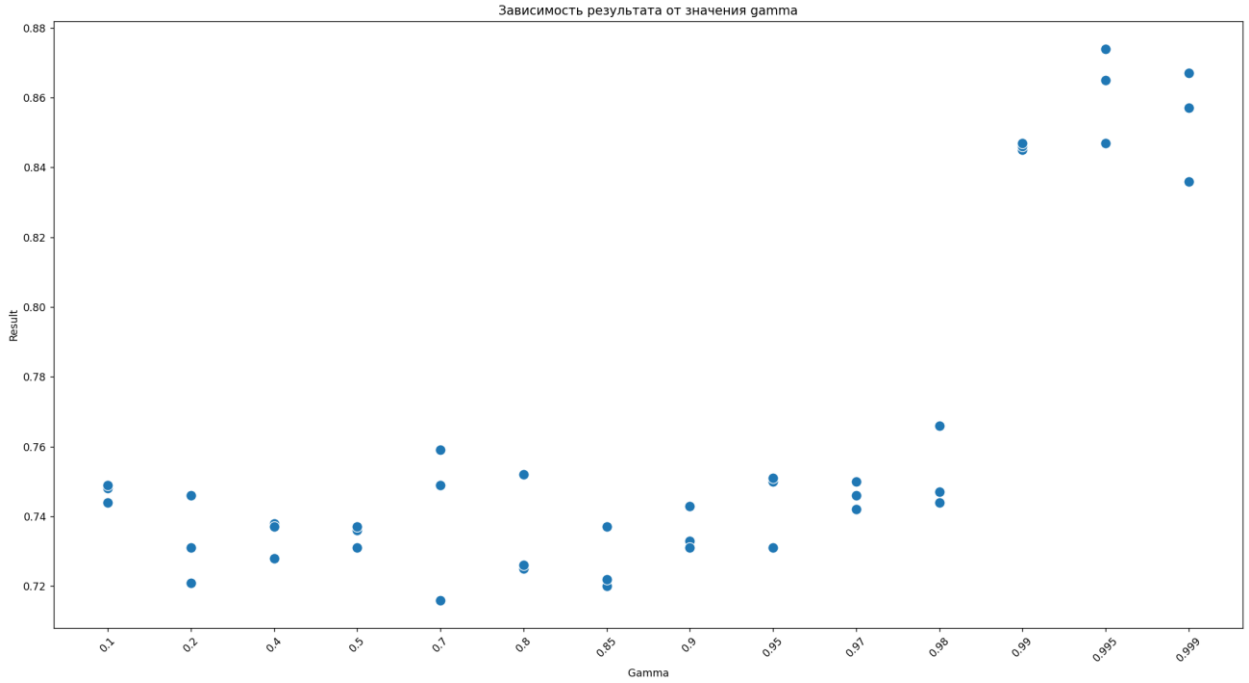
Трапер Максим. СПБГУ. Магистратура 1 курс «Искусственный интеллект и наука о данных»

P.S: я опять вспомнил о Grid только на середине… I`m sorry…

Задание №1: В алгоритме Policy Iteration важным гиперпараметром является gamma. Требуется ответить на вопрос, какой gamma лучше выбирать. Качество обученной политики можно оценивать например запуская среду 1000 раз и взяв после этого средний total_reward.

Эксперимент: проитерируемся по списку gammas = [0.1, 0.2, 0.4, 0.5, 0.7, 0.8, 0.85, 0.9, 0.95, 0.97, 0.98, 0.99, 0.995, 0.999]. K = 20, L = 20. По каждому из этих значений проитерируемся по 3 раза. Среда запускается 1000 раз.

Результат эксперимента представлен на графике

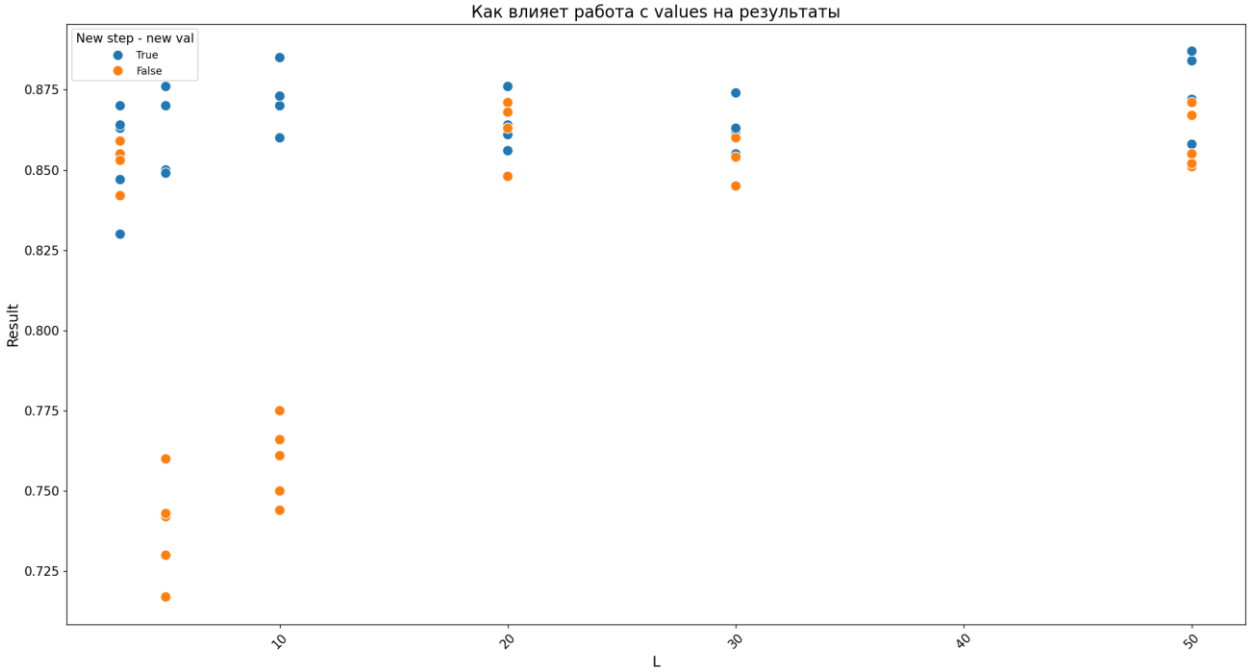


Вывод: исходя из графика можно сделать однозначный вывод, что для данной задачи заметное улучшение даёт только смена гаммы до значений около 1. Всё, что < 0.98, очень схоже по конечным результатам. А в трех наибольших выбранных гамма результаты очень похожи, хоть и отличаются ввиду стохастичности среды. Думаю, в данной задаче гамма = 0.995 – оптимальный вариант.

Задание №2: На шаге Policy Evaluation мы каждый раз начинаем с нулевых values. А что будет если вместо этого начинать с values обученных на предыдущем шаге? Будет ли алгоритм работать? Если да, то будет ли он работать лучше?

Эксперимент: K = 20, L = [3, 5, 10, 20, 30, 50, 100], gamma = 0.995. На каждом L проитерируемся по одним и тем же параметрам 5 раз. Итерируемся по списку [‘True’, ‘False’] который означает, будем ли мы передавать значения values на следующем шагу

или будет их инициализировать по новой. Для большей наглядности посмотрим это на различных значениях L.

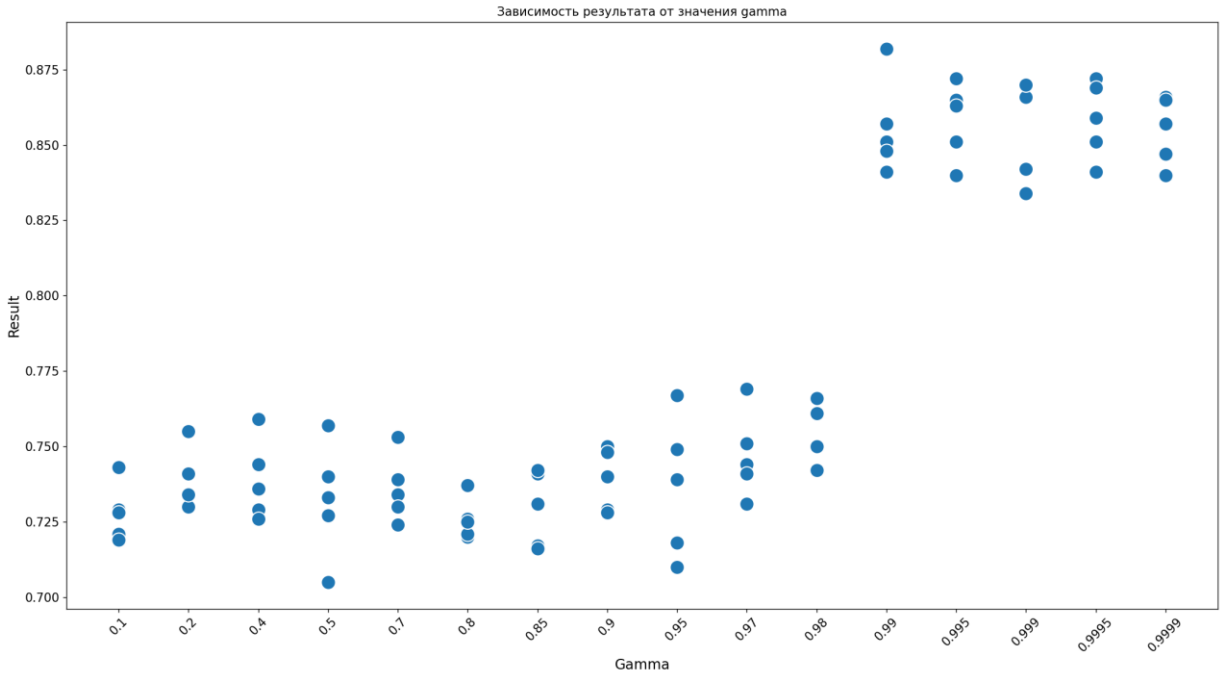


Вывод: Во первый да, решение передавать values с прошлого шага работает. Также можно чётко отметить, что решение передавать values с прошлого шага, а не каждый раз инициализировать новые привело к улучшению результатов. При больших значениях L улучшение результатов примерно на 1-2%. Изредка метод без передачи values выигрывает, но спишем это на стохастичность среды.

При любом раскладе, такое решение, как минимум, не ухудшает результаты а в некоторых случаях кратно улучшает. Пример: при малом значении L = 10, применение этого метода позволило добиться качества результатов сходное с работой при значении L = 50. Очевидно, что это выигрыш ещё и в производительности.

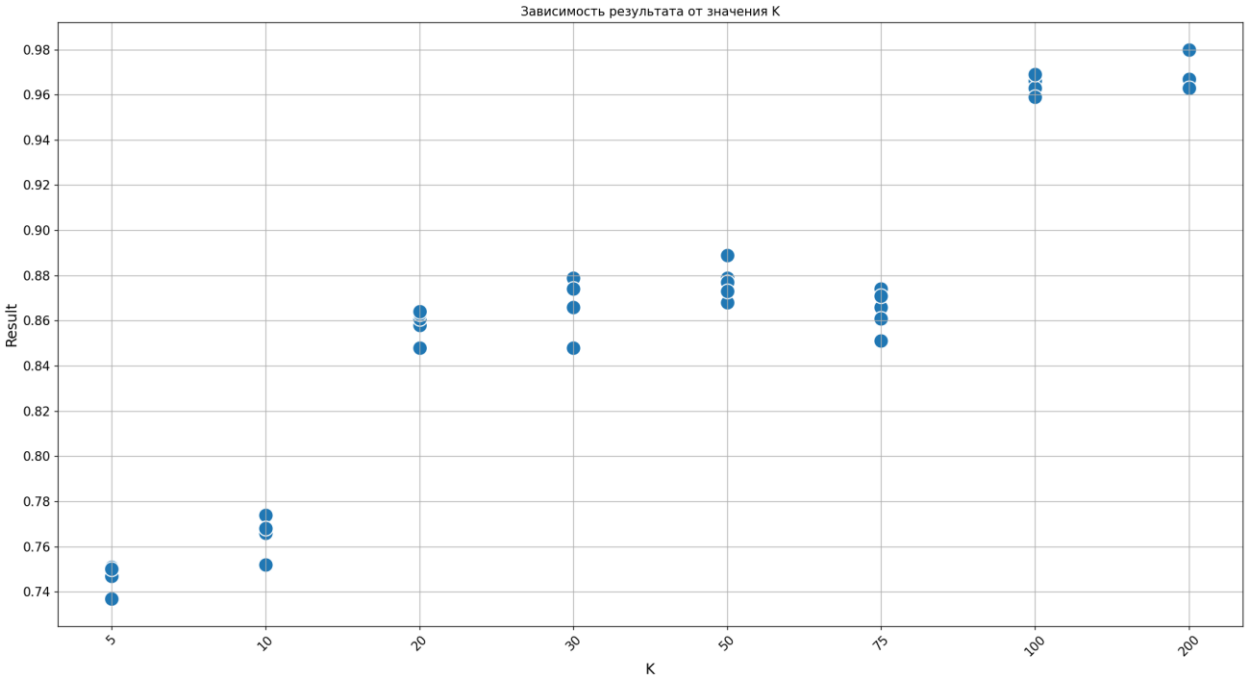
Задание №3: Написать Value Iteriation. Исследовать гиперпараметры (в том числе gamma). Сравнить с Policy Iteration. Поскольку в Policy Iteration есть еще внутренний цикл, то адекватным сравнением алгоритмов будет не графики их результативности относительно внешнего цикла, а графики относительно, например, количества обращения к среде.

Эксперимент №1: K = 20. Переберём гамму = [0.1, 0.2, 0.4, 0.5, 0.7, 0.8, 0.85, 0.9, 0.95, 0.97, 0.98, 0.99, 0.995, 0.999, 0.9995, 0.9999]. По каждому значению итерируемся 5 раз.



Вывод: алгоритм Value Iteration однозначно выигрывает от более высоких значений gamma. Пока будем брать гамма = 0.99.

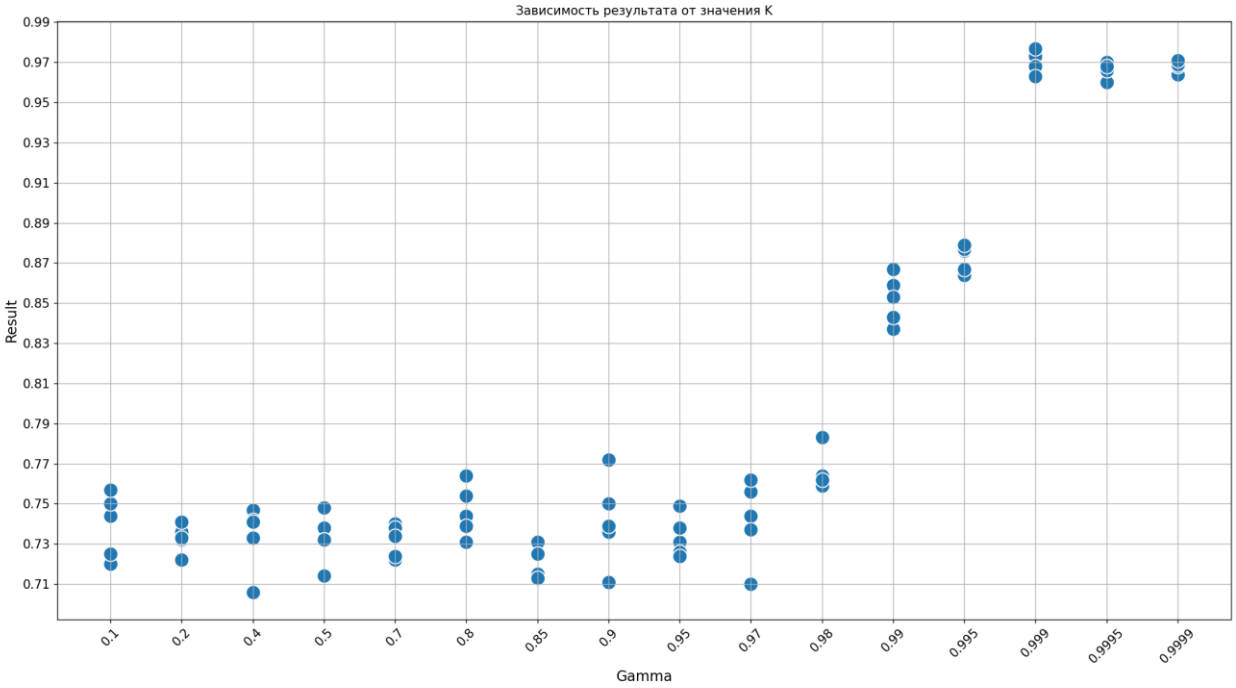
Эксперимент №2: Гамма = 0.99. Переберём K = [5, 10, 20, 30, 50, 75, 100]. По каждому значению итерируемся 5 раз.



Вывод: ну опять же, чем больше K, тем лучше). Но без фанатизма. Смена K со 100 на 200 дало в основном более высокие затраты на вычисления, но не дало качества.

Берём K = 100.

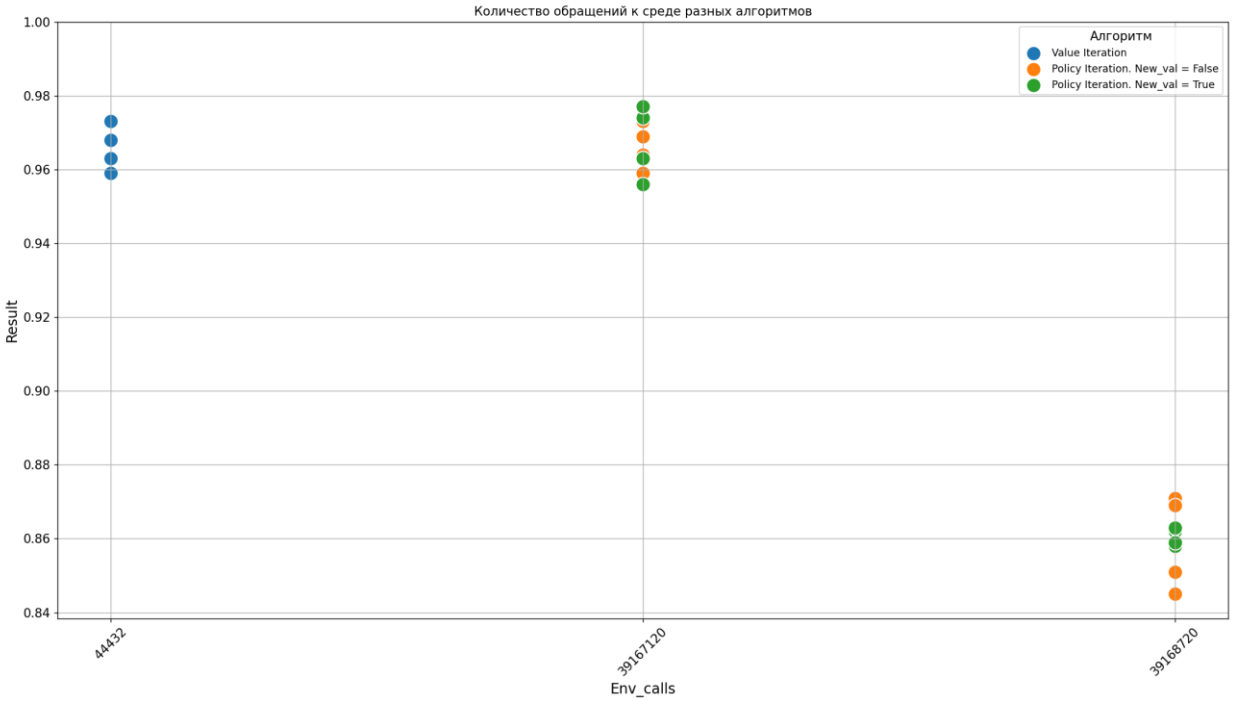
Эксперимент №3: Попробуем провести эксперимент №1 ещё раз, но возьмём K = 100. Переберём гамму = [0.1, 0.2, 0.4, 0.5, 0.7, 0.8, 0.85, 0.9, 0.95, 0.97, 0.98, 0.99, 0.995, 0.999, 0.9995, 0.9999]. По каждому значению итерируемся 5 раз.



Вывод: гипотеза**,** что значения гамма > 0.99 при K >> 20 дадут реальное улучшение, подтвердилась. Улучшения очень заметны и прибавка в качестве составляет около 10%. Может быть можно ещё добиться улучшения, путём повышения того и другого параметра, но тогда количество операций будет также расти. Думаю, что полученное качество ≈ 0.97 не такое плохое.

Итоговые параметры: K = 100, gamma = 0.999.

Эксперимент №4: прогоним оба алгоритма с лучшими параметрами, выбранными выше.



Вывод: оба алгоритма могут достичь около максимального результата, но неизвестно, стоит ли использовать Policy_Iteration с передачей values на каждом шаге, а не Value_Iteration, непонятно…

P.S: я не уверен в том, что понял это задание правильно. Лично мне тяжело сделать какой-то хороший вывод из этого, кроме того, что выше