Факультет
Компьютерных Наук

Москва
2022

Применение алгоритмов обучения с подкреплением в кредитном процессе

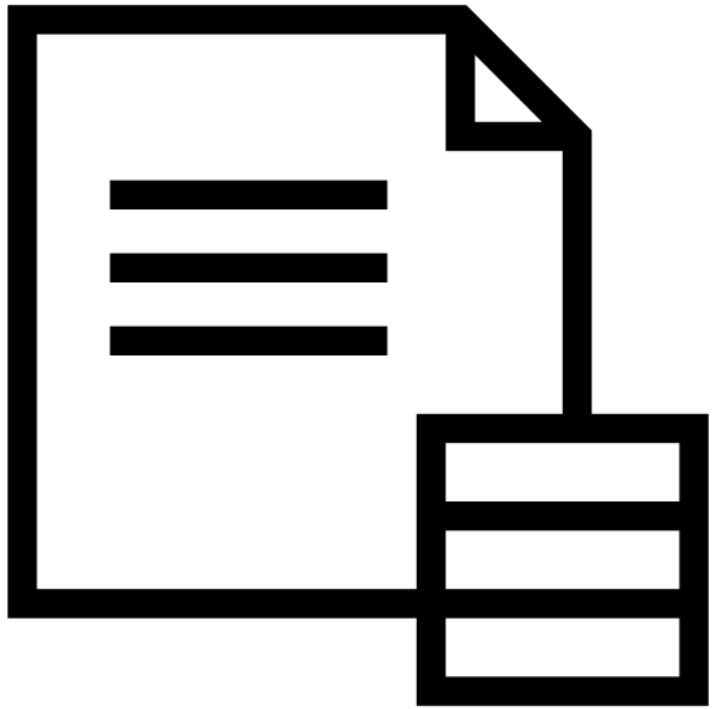
Выполнил:
Студент группы мФТиАД21, Попов Илья Иванович
Научный руководитель:
ПАО “Сбербанк”, исполнительный директор по исследованию данных, Щербаков Игорь Андреевич

Постановка задачи

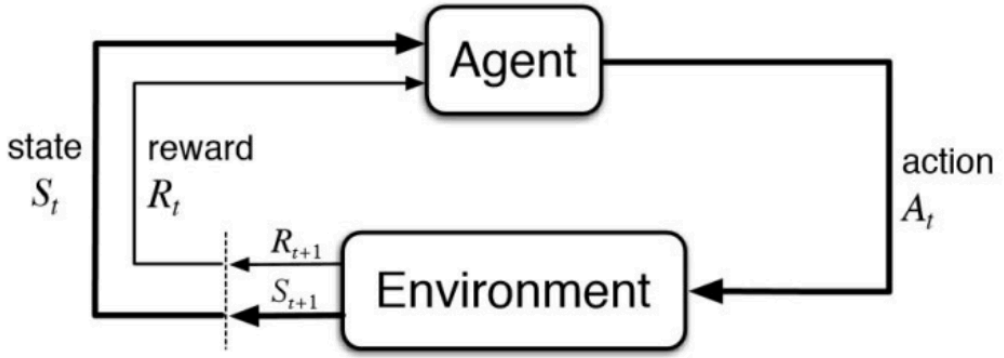
- В деятельности современных банков часто возникает задача выбора оптимального порога отсечения результатов скоринговых моделей для одобрения заявок по кредитам.
- Классические методы решения данной задачи имеют ряд недостатков, среди которых изменения потока кредитных заявок со временем, проблема смещения отбора и малая возможность адаптации к изменениям условий для правил, рассчитанных на исторических данных.
- Данные проблемы мешают банкам своевременно адаптироваться к изменениям в экономической среде.
- В данной работе рассматривается возможность применения алгоритмов машинного обучения с подкреплением.

Входные данные

- В качестве входных данных используются данные открытого датасета Landing Club
- Обучающая выборка: июнь 2007 - декабрь 2017
- Тестовая выборка: январь 2018 - июня 2018
- Произведен предварительный отбор и кодирование признаков
- На данных обучены две модели предсказания вероятности дефолта - CatBoost и LogReg
- Из вероятности дефолта на основе предсказания модели CatBoost был рассчитан кредитный рейтинг



Разработка среды



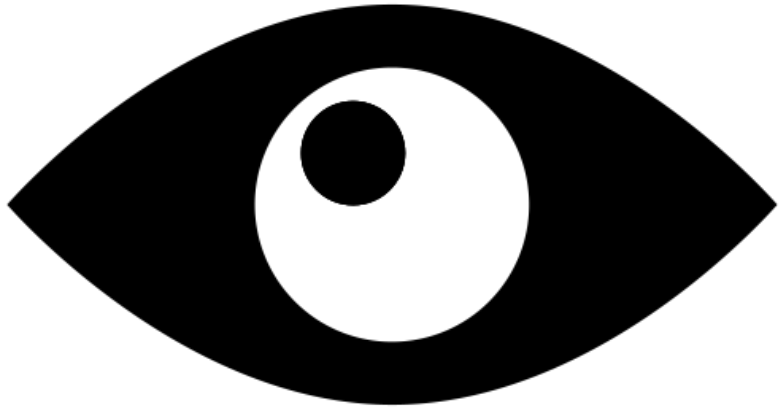
- Для корректной работы с входными данными и обеспечения необходимого уровня гибкости настройки была разработана собственная среда для обучения RL-агентов;
- Среда была разработана с использованием фреймворка OpenAI Gym

Основные вопросы при создании среды:

- Параметры, описывающие состояние среды;
- Логика работы шага среды;
- Режимы работы;
- Функция вознаграждения агента;

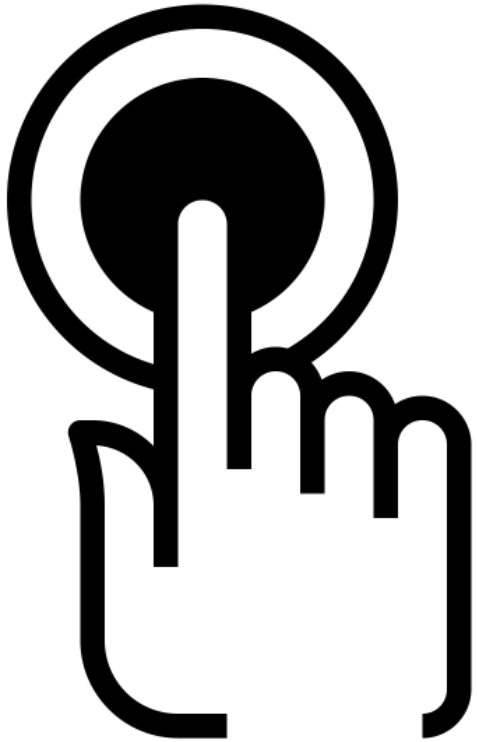
Состояние среды и структура наблюдения

- В качестве наблюдения среда отдает значение acceptance rate на прошлом шаге;
- В режиме score среда принимает значения порога отсечения как произвольное целое положительное число;
- В режиме pd среда принимает значения порога отсечения как произвольное вещественное число из интервала [0; 1];
- Это накладывает дополнительные ограничения на архитектуру агента.



Пространство действий агента

- Пространство действий среды - continious;
- В режиме score среда принимает значения порога отсечения как произвольное целое положительное число;
- В режиме pd реда принимает значения порога отсечения как произвольное вещественное число из интервала [0; 1];
- Это накладывает дополнительные ограничения на архитектуру агента.



Функция вознаграждения агента

- **Случай 1**: банк получает убыток в размере LDG • funded_amnt
- **Случай 2**: клиент объявлен дефолтом, но модель сработала корректно и не одобрила ему кредит - банк ничего не теряет;
- **Случай 3**: обратная ситуация, клиент не объявлен дефолтом, но модель сработала некорректно и не одобрила ему кредит - банк ничего не теряет, но ничего и не зарабатывает;
- **Случай 4**: банк получает прибыль в размере

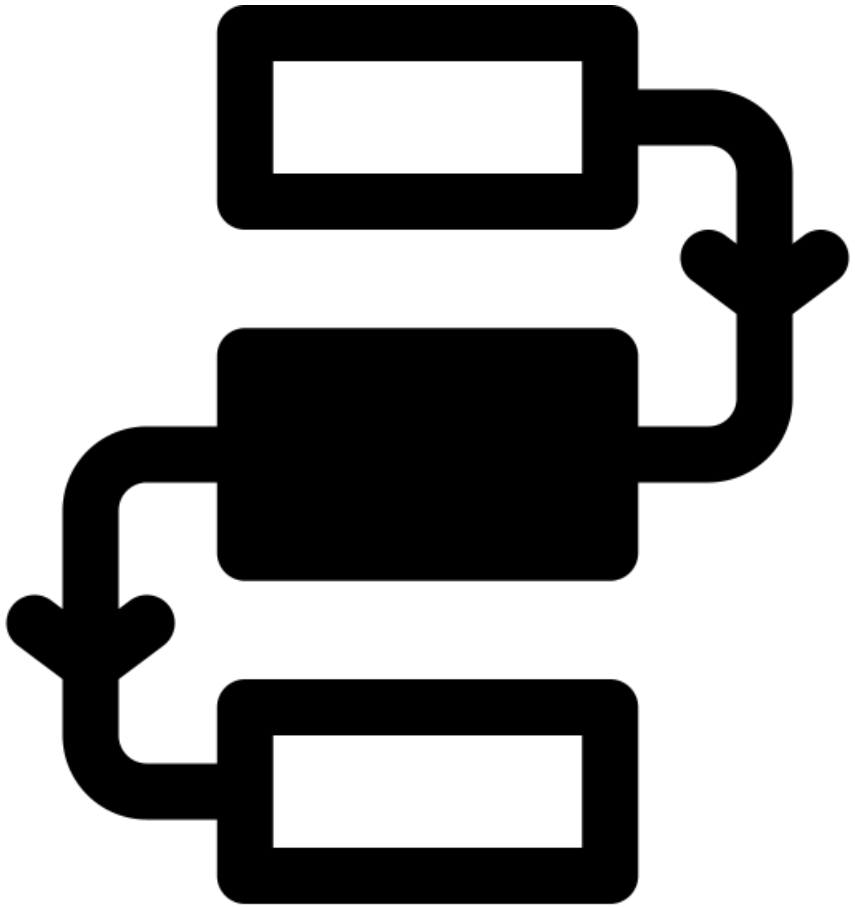
$$interest\_val = \frac{funded\_amnt}{int\_rate + \frac{int\_rate}{(1 + int\_rate)^{term} - 1} \bullet term}$$

		Была ли одобрена заявка на кредит?	
		Да	**Нет**
Был ли объявлен по заявке дефолт?	**Да**	Случай 1: банк теряет деньги	Случай 2: 0
	Нет	Случай 3: 0	Случай 4: банк получает прибыль

Шаг среды

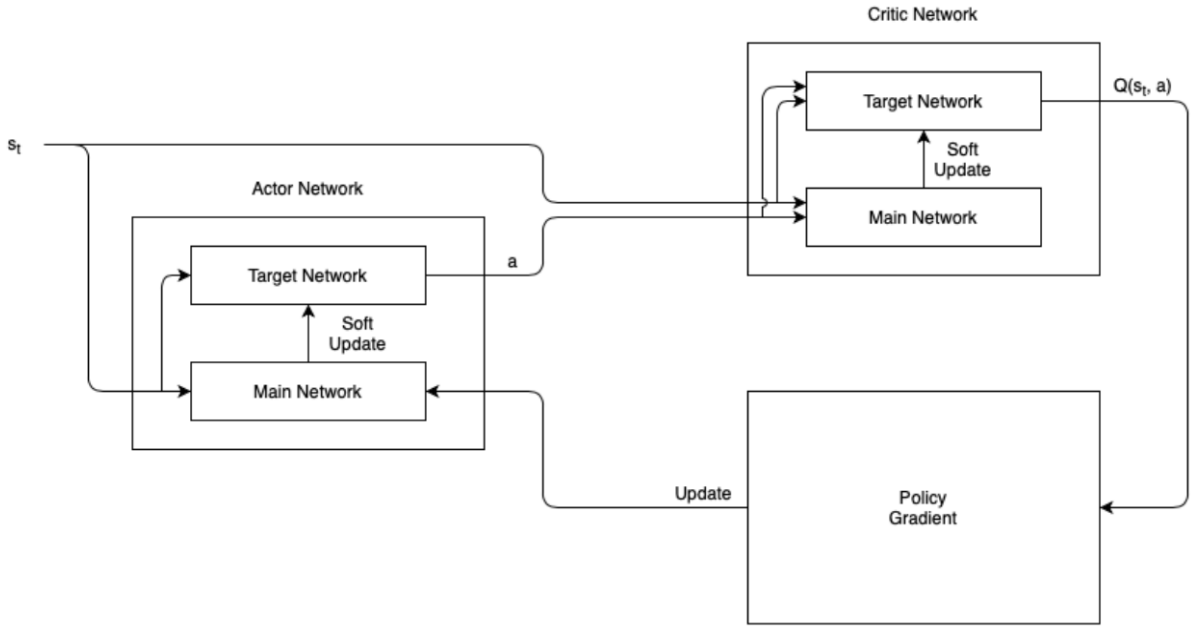
Для заявок, поступивших за месяц currMonth года currYear согласно полученной политике определяется, были ли они приняты или отклонены;

- Для всех принятых заявок рассчитывается вознаграждение агента:
  - Если среда работает в режиме мгновенного вознаграждения, то полученные вознаграждения суммируются и формируют reward;
  - Если среда работает в режиме отложенного вознаграждения, то вознаграждение добавляется к соответствующему месяцу в очереди. Reward объявляется равным первому значению отложенного вознаграждения в очереди, а в конец очереди добавляется 0;
- Рассчитывается observation = acceptance_rate = count(approved) / count(total);
- Если currMonth и currYear - последний месяц в истории наблюдений, то isDone = True, в противном случае - isDone = False;
- Цикл продолжается, пока isDone = False;



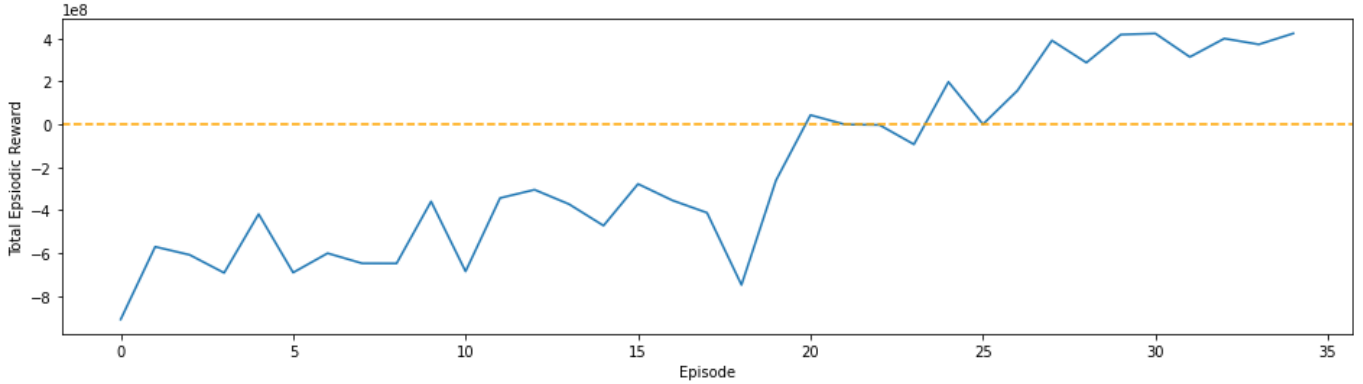
Архитектура агента

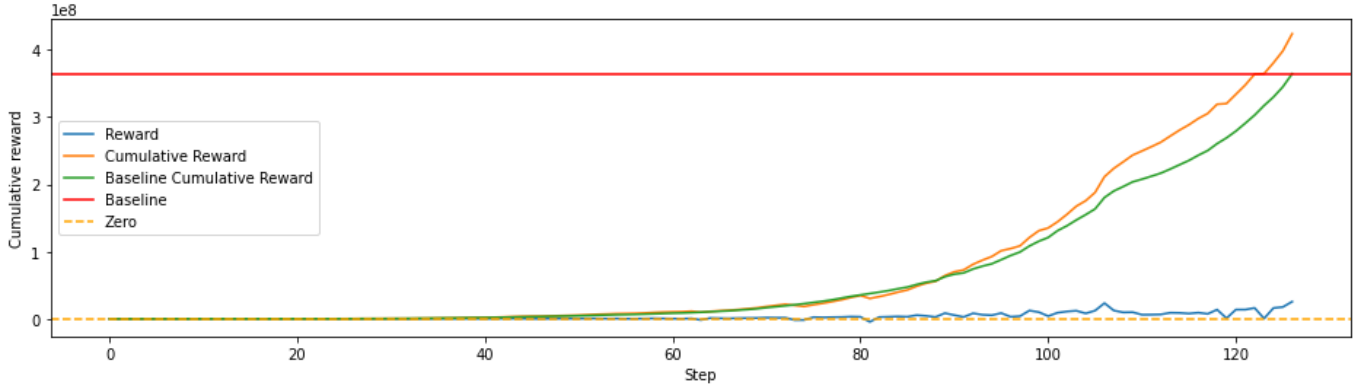
- Для реализации агента в работе был применён алгоритм Deep Deterministic Policy Gradient (DDPG);
- Алгоритм относится к семейству Actor-Critic аглоритмов, где Actor Network используется для выбора действия, а Critic Network - для приближения значения Q-функции для выбранного действия;
- Для стабилизации обучения используется механизм Soft Update, где каждая нейронная сеть заменяется двумя - Main и Target. Main Network используется для обучения, Target - для предсказания. Веса Target Network обновляются от весов Main Network с гиперпараметром τ, влияющим на скорость обновления весов;
- Для ускорения обучения используется механизм Prioritized Experience Replay.



Обучение и сравнение с бейзлайнами

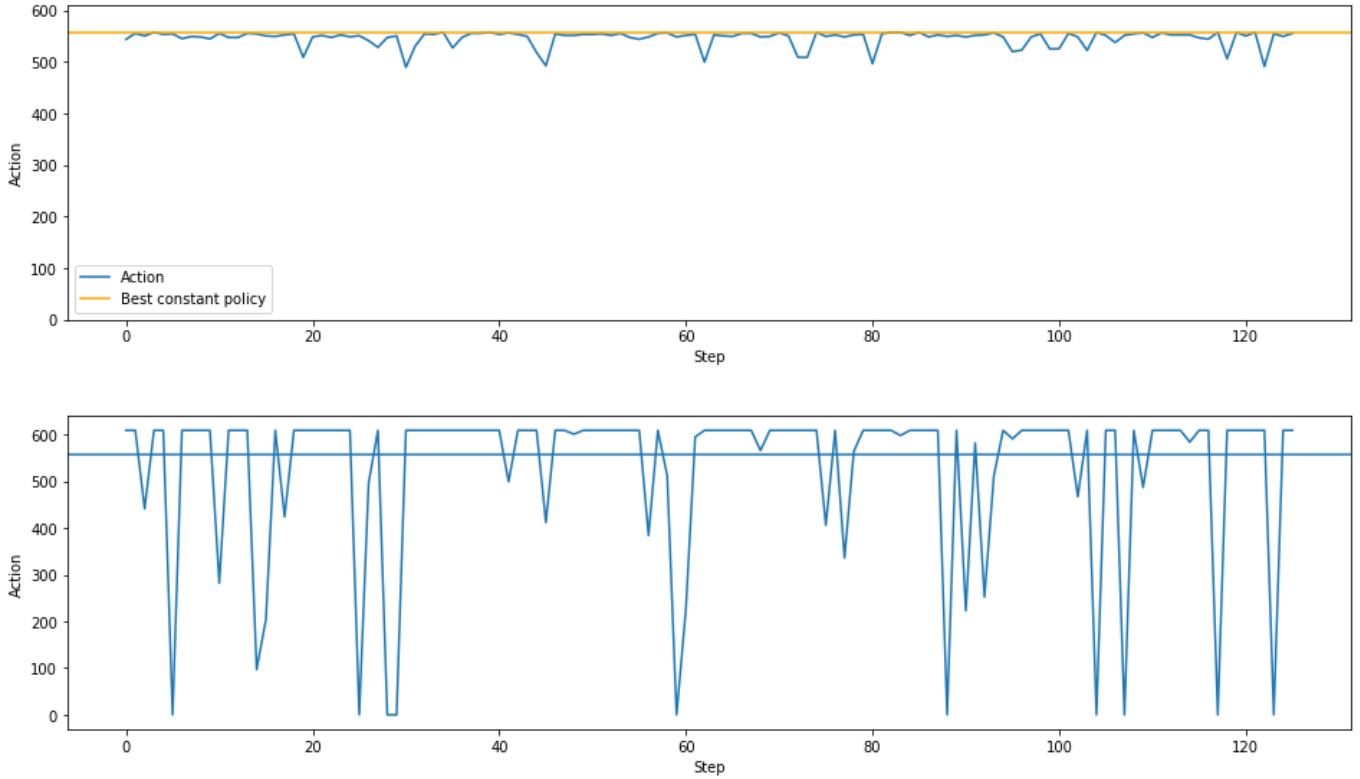
- Бейзлайн - наилучшая константная стратегия на обучающей выборке. В нашем случае - это постоянный выбор отсечки в 570;
- После 20 эпох обучения суммарное вознаграждение агента за эпоху впервые стало больше 0;
- После 30 эпох агент получил максимальное суммарное вознаграждение;
- Этот результат превзошёл тот, что показал бейзлайн - успех;
- Дальнейшее дообучение модели не привело к улучшению результата.





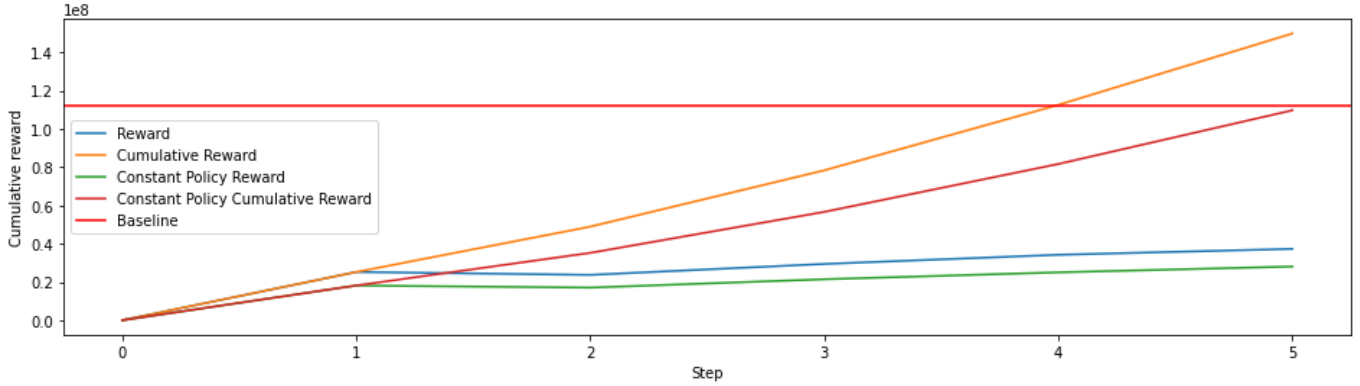
Обучение и сравнение с бейзлайнами

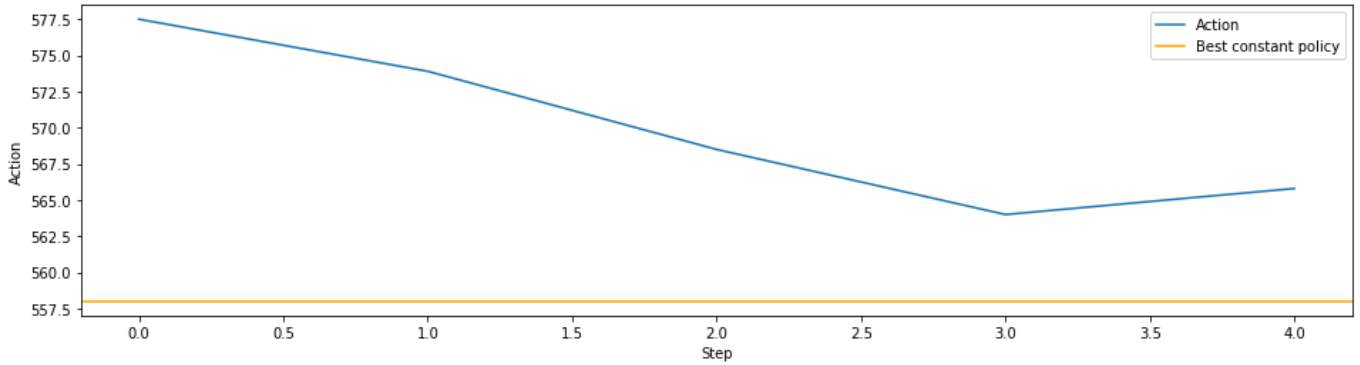
- Прирост прибыли у агента в сравнении с бейзлайном был достигнут за счет того, что агент периодически выбирал уровень отсечения сильно ниже бейзлайна, что позволяло выдавать больше кредитов и получать большую прибыль;
- В сравнении с действиями, выбранными агентом после 10 эпох обучения, заметно, что после следующих 20 эпох значения действия по модулю приблизились к бейзлайну, но большенство периодов колебаний сохранились;
- Вывод - модель достаточно быстро установила закономерности для определения периодов, где необходимо снизить порог, а в ходе следующих эпох сглаживался размер необходимого изменения.



Обучение и сравнение с бейзлайнами

- На тестовой выборке агент также показал результат, превосходящий бейзлайн;
- На этот агент достиг этого не за счет того, что он снижал порог и выдавал больше кредитов, а наоборот - ставил порог больше, чем бейзлайн, и выдавал меньше кредитов, но зато большего качества и предотвращал потери;
- Вывод - агент успешно адаптировался к новым условиям и в случае обучения на исторических данных может быть успешно применен в реальном бизнесе для подбора порога в реальном времени.





Заключение и дальнейшие шаги

Результаты работы:

- Проведен анализ существующих работ в области применения RL для задачи подбора порога отсечения в кредитном скоринге;
- На основе открытых данных была построена скоринговая модель;
- На ее основе была разработана среда для обучения и тестирования разных RL-агентов;
- В результате эксперимента было показано, что разработанный агент решает поставленную задачу лучше, чем бейзлайновая стратегия, как на обучающей выборке, так и на тестовой;

Дальнейшие шаги:

- Проведение экспериментов с использованием данных реального банка. Реализованная скоринговая модель далека от идеала, и была создана для демонстрации возможностей RL;
- Дополнительное обогащение входных данных модели основными макроэкономическими показателями и риск-факторами. Это должно помочь агенту лучше и быстрее адаптироваться в условиях неопределенности и резких изменений экономической обстановки в стране и мире;