

Автоматизация участия в котировочных сессиях

Банда пяти



Знакомство с командой



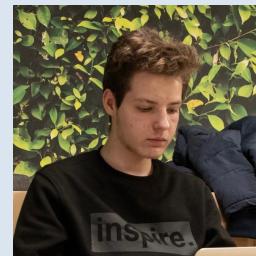
Антон Гайворонский
(ИТМО ФИТИП ИС)
Создатель баз данных,
объединяющий участки кода в одно



Антон Субботин
(ИТМО ИКТ)
В субБОТу писал бота (в пт и вс
тоже)



Мила (*ИТМО ФИТИП ИС*)
Математик-алгоритмист

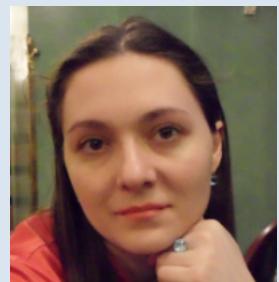


Саша (*ИТМО ФИТИП ИС*)
Программист-алгоритмист



Артём (*ИТМО ФИТИП КТ*)
Создатель презентации

Команда в полном составе



Аня
Математик



**Антон
Гайворонский**

Создатель баз данных,
объединяющий участки
кода в одно

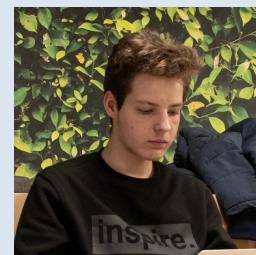


Антон Суббота

В субБОТу писал бота (в пт и вс
тоже)



Мила
Математик-алгоритмист



Саша
Программист-алгоритмист



Артём
Создатель презентации

Повестка дня

- Эмуляция котировочной сессии
- Бот: кнопки, аккаунт и токены
- Режим 0: наивный
- Режим 1: полуавтоматический
- Режим 2: доверительный
- Наш умный алгоритм
- Что ещё?

Эмуляция



```
def test():
    Ivan = EmulatorCustomer("Ivan")
    Alex = EmulatorProvider("Alex")
    Leo = EmulatorProvider("Leo")

    IS = EmulatorSession("IS", Ivan, 1000, 2.5, 3)
    print(IS.name, IS.customer, IS.current_value)

    for chelick in (Alex, Leo, Alex):
        IS.add(chelick)
        print(chelick.name, "added:", IS.name, IS.current_value)

    IS.end()
    print("End", IS.current_value, IS.winner)
    print("Alex", Alex.history.get(IS))
    print("Leo", Leo.history.get(IS))
```

Эмуляция



```
def test():
    Ivan = EmulatorCustomer("Ivan")
    Alex = EmulatorProvider("Alex")
    Leo = EmulatorProvider("Leo")

    IS = EmulatorSession("IS", Ivan, 1000, 2.5, 3)
    print(IS.name, IS.customer, IS.current_value)

    for chelick in (Alex, Leo, Alex):
        IS.add(chelick)
        print(chelick.name, "added:", IS.name, IS.current_value)

    IS.end()
    print("End", IS.current_value, IS.winner)
    print("Alex", Alex.history.get(IS))
    print("Leo", Leo.history.get(IS))
```

Эмуляция



```
def test():
    Ivan = EmulatorCustomer("Ivan")
    Alex = EmulatorProvider("Alex")
    Leo = EmulatorProvider("Leo")

    IS = EmulatorSession("IS", Ivan, 1000, 2.5, 3)
    print(IS.name, IS.customer, IS.current_value)

    for chelick in (Alex, Leo, Alex):
        IS.add(chelick)
        print(chelick.name, "added:", IS.name, IS.current_value)

    IS.end()          925      Alex
    print("End", IS.current_value, IS.winner)
    print("Alex", Alex.history.get(IS)) winner
    print("Leo", Leo.history.get(IS)) participant
```

Инициализация события

```
if __name__ == '__main__':
    engine = create_engine(CONNECTION_STRING)
    Base.metadata.create_all(engine) # Создает таблицы, если они ещё не созданы

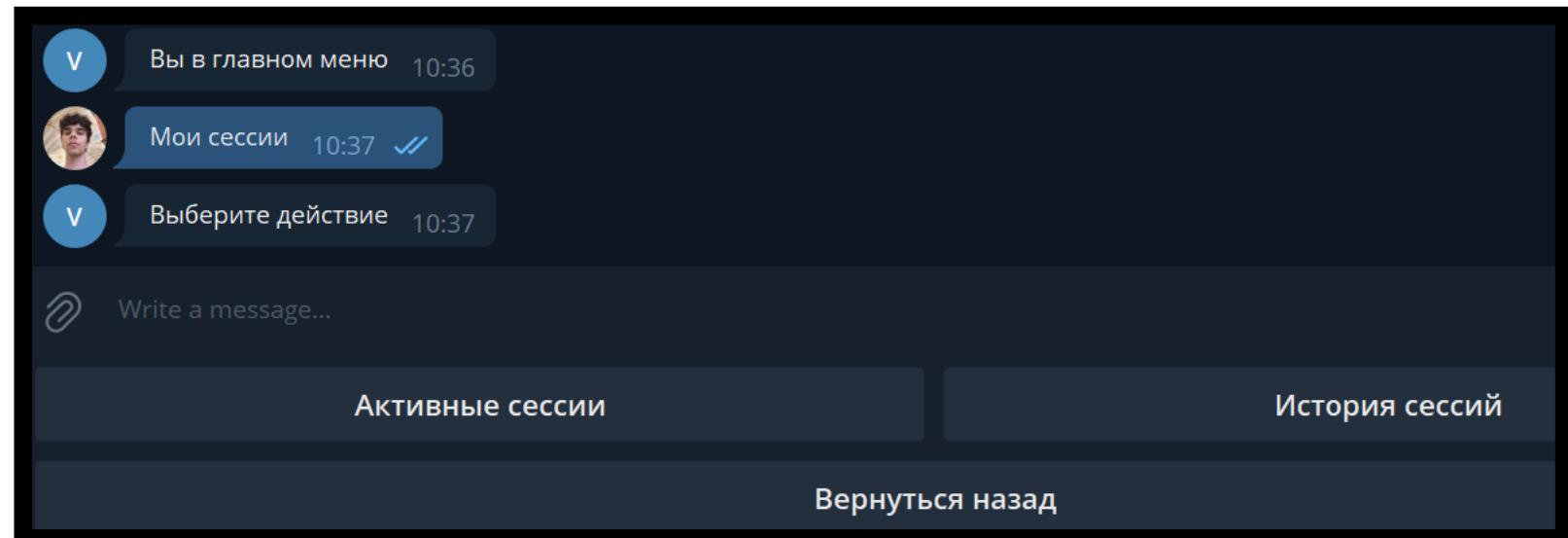
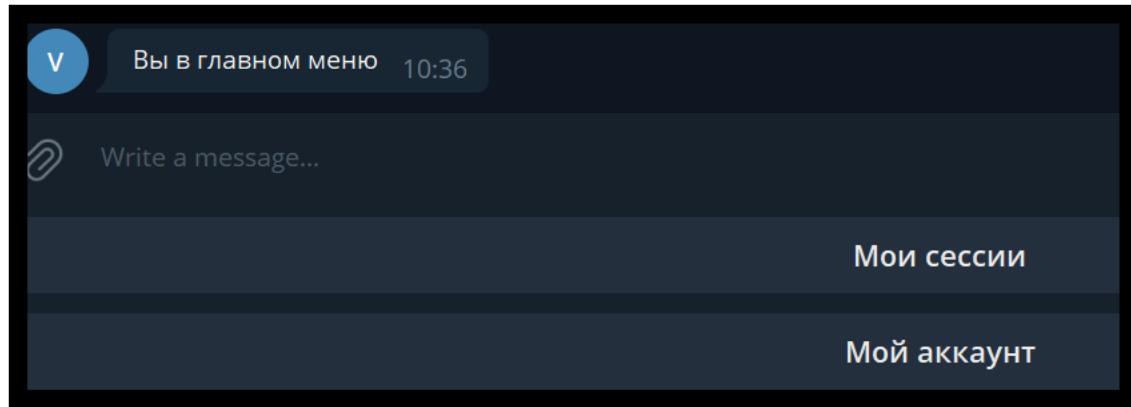
    Ivan = EmulatorCustomer("Ivan")
    Artyom = EmulatorCustomer("Artyom")
    Maria = EmulatorCustomer("Maria")

    IS = EmulatorSession("IS", Ivan, 1000, 2.5, 3)
    VT = EmulatorSession("VT", Artyom, 2000, 1, 3)
    CT = EmulatorSession("CT", Maria, 500, 5, 3)

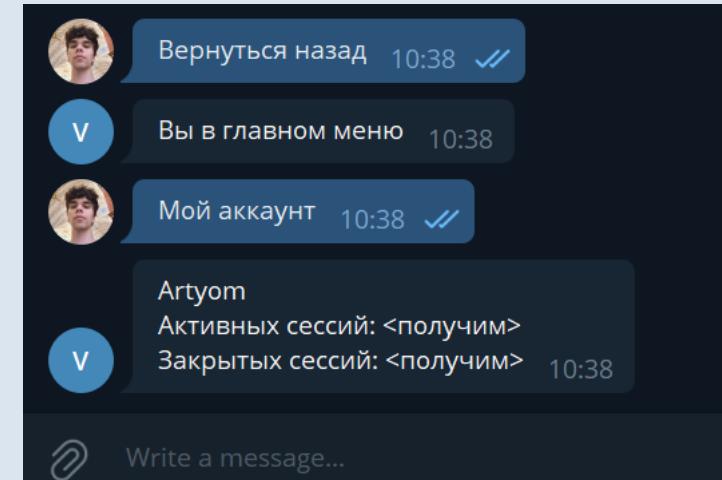
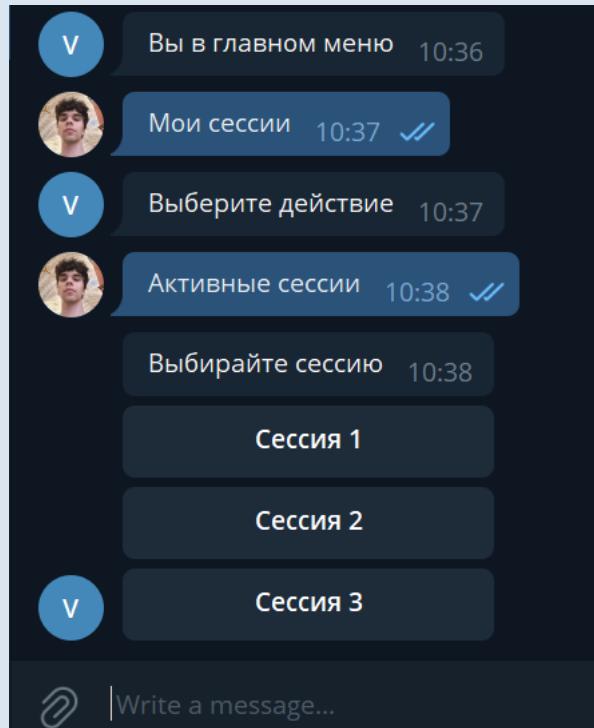
    Alex = EmulatorProvider("Alex")
    Leo = EmulatorProvider("Leo")
    Mila = EmulatorProvider("Mila")
    Anton = EmulatorProvider("Anton")
    Ksenia = EmulatorProvider("Ksenia")

    tokens: Dict[str, EmulatorUser] = {
        Ivan.password: Ivan,
        Artyom.password: Artyom,
        Maria.password: Maria,
        Alex.password: Alex,
        Leo.password: Leo,
        Mila.password: Mila,
        Anton.password: Anton,
        Ksenia.password: Ksenia,
    }
```

Бот



Бот



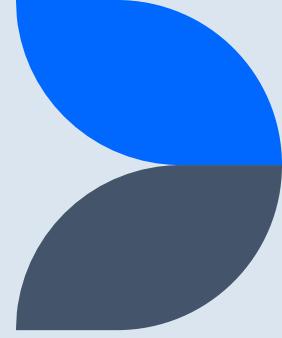
Почему телеграмм-бот?

Благодаря телеграмм-боту информация о текущих сессиях всегда будет у вас под рукой. Несколько нажатий - и вы уже в курсе о таких факторах, как активные/неактивные сессии, текущая ставка, положение вас относительно других поставщиков, предполагаемое снижение ставки.

Телеграмм бот реализует такие функции, как изменение режима работы, изменение предельной ставки, определение наиболее выгодных сессий.

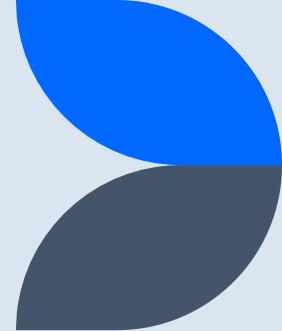


Режим 0



Обычный режим. Пользователь может вручную поставить ставку прямо в боте, наблюдать за текущей ставкой и оставшимся временем. Он будет уведомлён о финальной стадии сессии за 5 минут до конца сессии

Режим 1



Полуавтоматический режим. Поставщик может выбрать минимальную сумму, за которую он готов взяться за заказ. Ставки будет делать бот, пока текущая стоимость сессии больше минимальной суммы.

Как только текущая стоимость становится меньше минимальной суммы или равной ей, пользователь получает уведомление с вариантами: больше не делать ставки в этой сессии или снизить минимальную сумму и продолжить участие.

Бот переходит в режим 0 за 5 минут до конца сессии.

Режим 2

Умный алгоритм определяет насколько выгодна каждая сессия относительно других.

В связи с показателем выгодности сессии сортируются по убыванию показателя. Поставщик, на основе проанализированных данных, выбирает свою дальнейшую стратегию.

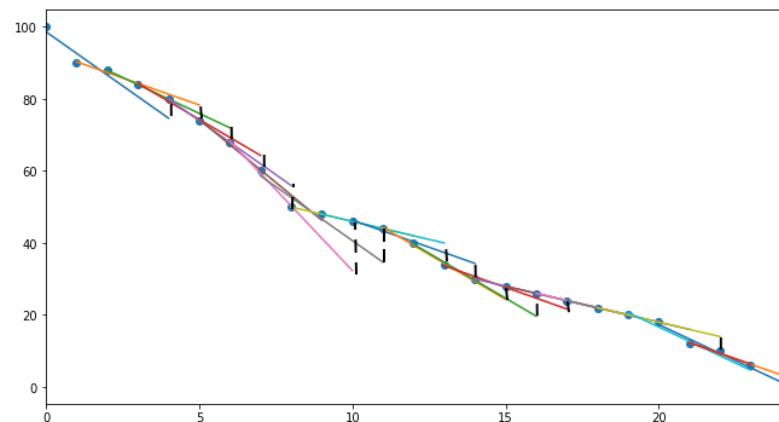
Алгоритм также создаёт рекомендации в режимах 0 и 1, основываясь на имеющихся данных.



Наш алгоритм

Линейное предсказание

На основе некоторого приращения значения ставки δP за диапазон времени δt выявляется **линия тренда** путем дискретной линейной аппроксимации (метод наименьших квадратов)



На определенный период времени (параметризуется) выявляется возможное изменение ставки сессии.

Позднее, предполагаемое убывание функции сравнивается с фактическим

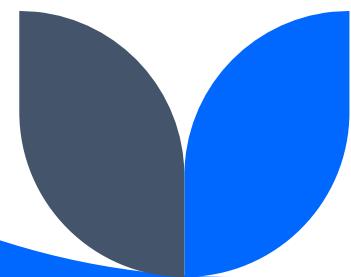
Предполагаемое убывание функции продажи доступно поставщику как рекомендация, а также используется в режиме 2.

Наш алгоритм

Коэффициент выгоды участия в тендере

На основе скорости падения цены и предполагаемой прибыли в определенный момент времени рассчитывается коэффициент, показывающий, насколько выгодна для поставщика данная сессия относительно других.

Этот коэффициент - значение целевой функции $y = S_{cur.price} * (1 - V_{price})$, аргументы которой - экспоненциально нормированные V_{price} и $S_{cur.price}$



Преимущества нашего решения

- 1) Не сбивается цена сессии, которая не выгодна поставщику-пользователю, но может быть выгодна другим поставщикам.
- 2) У поставщика есть несколько активных сессий, в победе которых он относительно уверен. Это позволяет планировать его стратегию ведения продаж. Например, искать и приобретать/подготавливать необходимый заказчику товар.



Перспективы развития

- Нуждается в доработке и финансировании
- Использование технологий для работы с big data и ML для минимизации погрешности предсказаний
- Анализ расходов и прибыли поставщика, регулярный отчет по доходам/расходам
- Развитие доверительного режима на основе этого анализа

Спасибо

Артём Швецов

shvetsovart55@gmail.com

@shvetsovart

<https://github.com/inter-malchik/Tenderhack-hackaton>

