

# Постановка задачи и исходные данные

Необходимо разработать приложение, которое будет принимать на вход массив исходных данных в .txt, группировать эти данные по заданному алгоритму и затем выявлять определенные закономерности.

Прикладной смысл приложения – поиск и фильтрация паттернов (формаций, закономерностей) в потоке данных по биржевым сделкам. В качестве исходных данных используется массив исторических данных (т.е. сделок, уже совершенных ранее) в формате .txt. В потоке данных содержится информация по каждой сделке: дата, время, цена, объем, тип сделки (покупка/продажа).

Формат исходных данных:

[Date] [Time] [MS];[Price];[Bid];[Ask];[Volume]

[Date] – это дата совершения сделки (ГГГГММДД).

[Time] – это время совершения сделки (ЧЧММСС).

[MS] – время в миллисекундах (возможно), параметр не нужен для анализа.

[Price] – цена, по которой была совершена сделка (float, от 0 до 8-10 знаков после запятой).

[Bid] – цена, по которой в момент сделки было предложение от покупателя (далее «бид»).

[Ask] – цена, по которой в момент сделки было предложение от продавца (далее «аск»).

[Volume] – объем сделки, то есть кол-во контрактов, купленных/проданных в данной сделке. (int)

Для анализа необходимо произвести агрегацию сырых данных в бары (от “bar” - полоска).

Бар – это объект, внутри которого содержатся просуммированные значения [Volume] для каждого ключа [Price].

Причем значения Volume суммируются отдельно по [Bid] и по [Ask].

```
if (Price == Bid) Bar[Price].Bid += Volume;
```

```
else if (Price == Ask) Bar[Price].Ask += Volume;
```

Все бары имеют одинаковую размерность. К примеру, если задана размерность бара = 4, то производится суммирование [Volume] по [Price] до того момента, пока разница между наименьшим значением ключа [Price] и наибольшим значением ключа [Price] не достигнет значения «4 \* Шаг\_Изменения\_Ключа» (на примере выше Шаг\_Изменения\_Ключа = 0.25). Как только разница между min и max значением ключа достигла порогового значения – запись начинает производиться уже в следующий бар.

**К примеру**, если в исходных данных нам встретились Price = {2921, 2921.25, 2921, 2920.75, 2920.50, 2920.25, 2920}, то для [Price] = {2921, 2921.25, 2921, 2920.75, 2920.50, 2920.25} мы добавляем [Volume] в первый бар, а вот для [Price] = {2920} уже начинаем запись во второй бар, а первый закрываем для записи. Потому что Bar.MaxKey == 2921.25, Bar.MinKey = 2920.25, и (Bar.MaxKey – Bar.MinKey)/0.25 == 4, то есть значение {2920} уже не помещается в наш первый бар.

**Во избежание недопонимания:** как только открывается следующий бар, все записи производятся уже в него, предыдущий бар уже заполнили и забыли о нем. То есть если после [Price] = {2920} снова появится [Price] = {2920.25}, то мы не возвращаемся обратно в первый бар, а продолжаем добавлять [Volume] во второй бар. Как только возникнет ситуация, что новое значение [Price] не помещается во второй бар – то открываем третий и т.д. На скриншоте справа показано графическое представление баров.

К примеру, на среднем баре видно, что для [Price] = {2917.75} (см. в верхнюю часть среднего бара) сумма всех [Volume] по [Ask] равна 113, а сумма всех [Volume] по [Bid] равна 0.

MES 11.05.2020.txt – Блокнот

Файл Правка Формат Вид Справка

```
20200510 220000 0700000;2921;2921;2921.5;3
20200510 220000 0700000;2921;2921;2921.5;1
20200510 220000 0820000;2921;2921;2921.5;3
20200510 220000 0820000;2921;2921;2921.5;2
20200510 220000 0820000;2920.75;2920.75;2921.5;3
20200510 220000 0820000;2920.5;2920.5;2921.5;1
20200510 220000 0820000;2920.5;2920.5;2921.5;1
20200510 220000 0820000;2920.5;2920.5;2921.5;1
```



## Цель исследования

Основная задача приложения – выявлять закономерности в данных, которые позволят заключать сделки с положительным матожиданием.

Упрощенно говоря – делать ставки на то, что цена в ближайшее время увеличится или уменьшится на значение  $N$ , а при появлении предпосылок к тому, что цель достигнута не будет, быстро выходить из сделки, минимизируя убыток.

Для создания и тренировки модели необходимо понимать, как именно заключаются сделки.

Начнем с терминологии.

**Тик** – это минимальный шаг изменения цены. В примере выше он равен 0.25, то есть при изменении цены с 2921.00 на 2921.25 мы говорим, что «цена выросла на 1 тик».

**Сделка в лонг** – это ставка на то, что цена вырастет на  $N$  тиков.

**Сделка в шорт** – это ставка на то, что цена упадет на  $N$  тиков.

**Тейк-профит**, или просто **тейк** – это то самое значение  $N$ . К примеру, «лонг на 2921.00 с тейком 8» - это ставка на то, что с 2921.00 цена вырастет до  $(2921 + 0.25 * 8) = 2923.00$ . И если это происходит, то сделка закрывается с прибылью 8 тиков.

**Стоп-лосс**, или просто **стоп** – это пороговое значение, при котором считается, что сделка закрывается с убытком. В примере выше, если мы добавим «... тейком 8 и стопом 4», то если цена дойдет до  $(2921 - 0.25 * 4) = 2920.00$  до того, как она дойдет до тейка – то сделка закрывается с убытком в 4 тика.

**Комиссия** – это сумма, которая вычитается из результата сделки вне зависимости от того, с каким результатом эта сделка закрывается. В примере выше, если у нас тейк 8 и стоп 4, и при этом комиссия 0.5 тика, то в случае прибыльной сделки чистая прибыль составит  $(8 - 0.5) = 7.5$  тиков, а в случае убыточной сделки убыток составит  $(-4 - 0.5) = -4.5$  тика.

Итак, теперь вы понимаете эльфийский, а значит, мы можем перейти к описанию ключевого процесса – входу/выходу из сделки.

Задача нейросети – находить закономерности, которые позволят заключать сделки с положительным матожиданием прибыли.

$EV = P\_profit * (takeProfitSize - commission) - P\_loss * (stopLossSize + commission)$ , где  $P\_Profit$  и  $P\_loss$  – это вероятность получения прибыли/убытка в сделке, а  $takeProfitSize$  и  $stopLossSize$  – размер тейка и стопа в тиках.

Есть 2 способа обеспечить положительное матожидание.

**Способ #1:** выбирать хорошие точки входа. К примеру, некоторые паттерны, которые будут показаны на следующей странице, образуются в тех местах, где крупный игрок начинает набирать позицию, и часто это приводит к тому, что имеющееся движение цены останавливается и начинается движение в противоположную сторону. Соответственно, вероятность  $P\_profit$  значительно выше, чем  $P\_loss$ , и это позволяет иметь положительное матожидание прибыли даже при условии одинаковых размеров тейка и стопа.

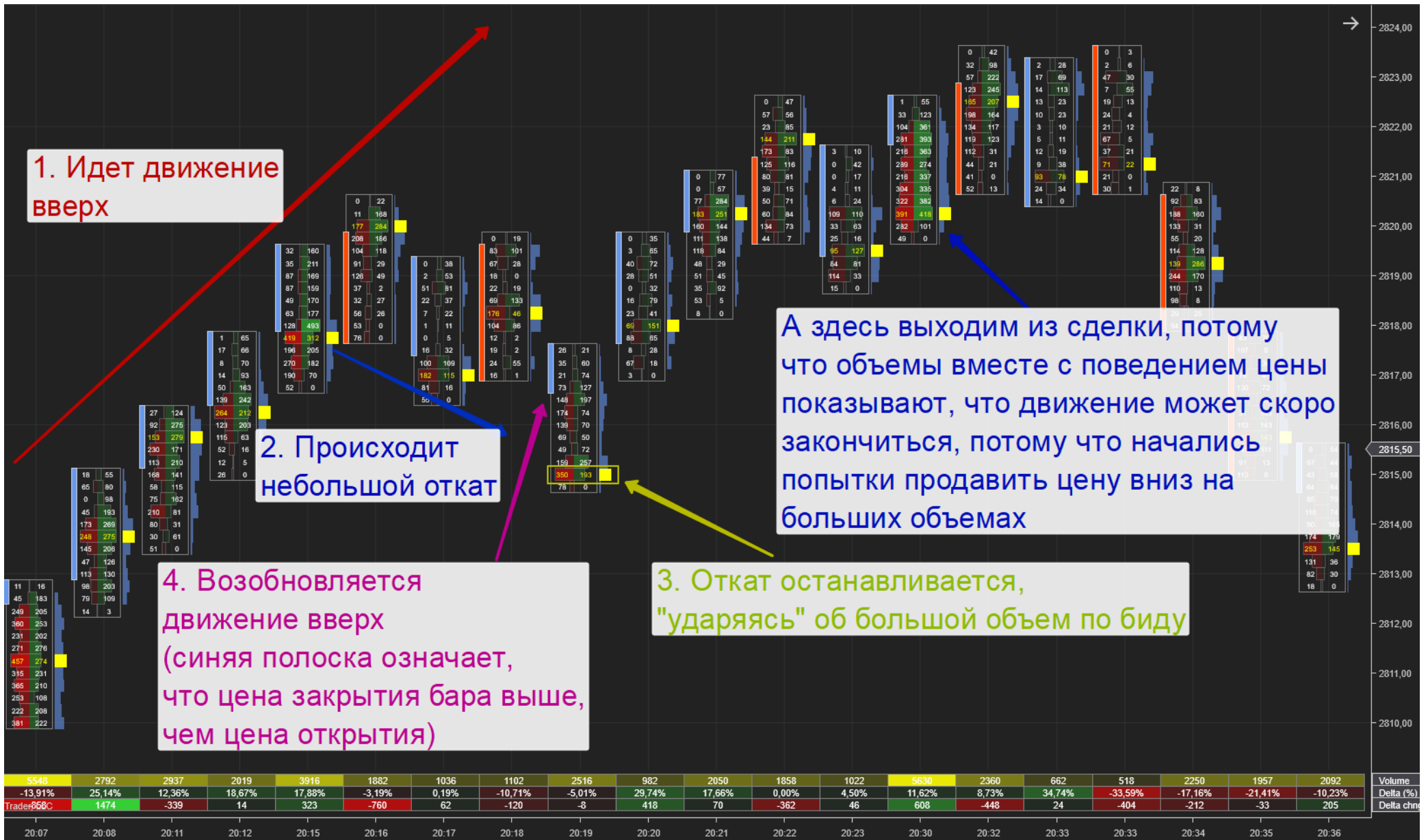
**Способ #2:** управлять позицией, то есть двигать тейк и стоп в зависимости от того, как развивается ситуация. В этом случае убыточные сделки быстро закрываются, а прибыльные удерживаются «до упора», что обеспечивает соотношение «тейк\стоп» на уровне, необходимом для положительного матожидания прибыли от сделки. В этом случае

И идеальным способом будет являться сочетание способов #1 и #2, то есть поиск хорошей точки входа вместе с управлением позицией.

**Обратите внимание:** речь идет не о попытке предсказать, какое именно значение будет у цены спустя время  $t$ , а о том, чтобы сначала найти точки входа, которые позволят с нужной вероятностью зайти в движение цены в нужном направлении, а затем в процессе принимать решение о выходе из позиции на основании поведения цены и поведения объемов.

На следующей странице будут показаны примеры паттернов, которые указывают на хорошие точки входа, а также удачные и неудачные сделки, вход в которые был осуществлен именно на основании этих паттернов. Там же будет показано, как изменение объемов может стать сигналом для входа/выхода из сделки.

# Пример паттерна и изменения поведения цены



Вот пример того, как поведение цены вместе с объемами может дать явный сигнал о продолжении движения (см. п. 3 и 4, откат вниз и слом «растущей» тенденции не удался) или о его прекращении (см. правую часть скриншота, где в баре была попытка продать цену вниз на аномально высоких объемах, что свидетельствует о том, что здесь вступил в игру крупный игрок, чьи действия могут создать достаточно серьезное движение цены).

## Механика движения цены (часть 1\2)

Для того, чтобы у вас сформировалось полное понимание того, почему все эти цифры в барах должны хоть как-то влиять на поведение цены и давать возможность находить хорошие точки входа, далее будет информация о том, за счет чего конкретно происходит изменение цены при торгах на бирже. Это поможет правильным образом описать и обучить модель.

Итак, каждый участник торгов либо продает, либо покупает.

И продавать/покупать он может одним из двух способов – рыночным ордером или лимитным ордером.

Лимитный ордер – это «хочу купить/продать ровно по такой-то цене, ставлю заявку по этой цене и готов ждать, пока кто-то согласится мне продать/у меня купить по этой цене».

Рыночный ордер – это «хочу купить/продать **прямо сейчас**, по какой цене дадут прямо сейчас – по той и куплю/продам».

Рыночные ордера исполняются об лимитные.

К примеру, давайте представим, что у нас по цене 2900.00, 2900.25 и 2900.50 стоит по 100 контрактов на продажу.

И кто-то захотел купить по рынку 20 контрактов.

В этом случае его рыночный ордер на покупку 20 контрактов исполнится о лимитный ордер на продажу по цене 2900.00, и на этом уровне останется только 80 контрактов на продажу.

И в этом случае цена последней сделки (которая и записана в наших исходных данных в столбце [Price]) будет равна 2900.00.

Но что произойдет, если кто-то захочет купить по рынку не 20, а 220 контрактов?

В этом случае он заберет все 100 контрактов по 2900.00, все 100 контрактов по 2900.25 и еще 20 из 100 по цене 2900.50.

И цена последней сделки будет равна уже 2900.50.

Итого – цена выросла на 2 тика.

Таким образом, получается что рыночные ордера двигают цену, а лимитные, наоборот, мешают ей двигаться, останавливают ее.

И чтобы сформировалась полная картина, остается только понять, что одновременно на рынке присутствуют и рыночные покупатели, и лимитные покупатели, и рыночные продавцы, и лимитные продавцы. Поэтому цена не идет строго вверх или строго вниз – она постоянно колеблется то вверх, то вниз. Но чем больше перевес одной из сторон (покупателей/продавцов), тем больше цена сдвигается в соответствующую сторону.

К примеру, если лимитных покупателей и лимитных продавцов примерно поровну, но при этом рыночных покупателей гораздо больше, чем рыночных продавцов – то цена полетит вверх, потому что более активные рыночные покупатели будут «съедать» гораздо больше лимитных ордеров на покупку, чем рыночные продавцы съедят лимитных ордеров на продажу. Соответственно, возникнет ситуация вида «на 1 тик вниз, на 2 тика вверх», что и приведет к росту цены в течение определенного времени.

В качестве утрированного примера рассмотрим ситуацию, когда кто-то поставил огромную лимитную заявку на продажу 100 000 контрактов (при среднем объеме 100-300).

В этом случае цена не вырастет ни на 1 тик до того момента, пока покупатели не разберут все эти 100 000 контрактов (а это займет очень много времени).

И рассмотрим обратную ситуацию – когда на 1 уровне цены стоит в среднем 100-300 контрактов, и тут кто-то решил купить по рынку 100 000 контрактов.

В этом случае цена моментально взлетит на несколько сотен тиков, потому что для удовлетворения рыночного ордера на 100 000 контрактов придется собрать по 100-300 контрактов с 300-1000 уровней цены.

## Механика движения цены (часть 2\2)

И теперь с учетом этой информации рассмотрим, что же произошло в предыдущем примере.

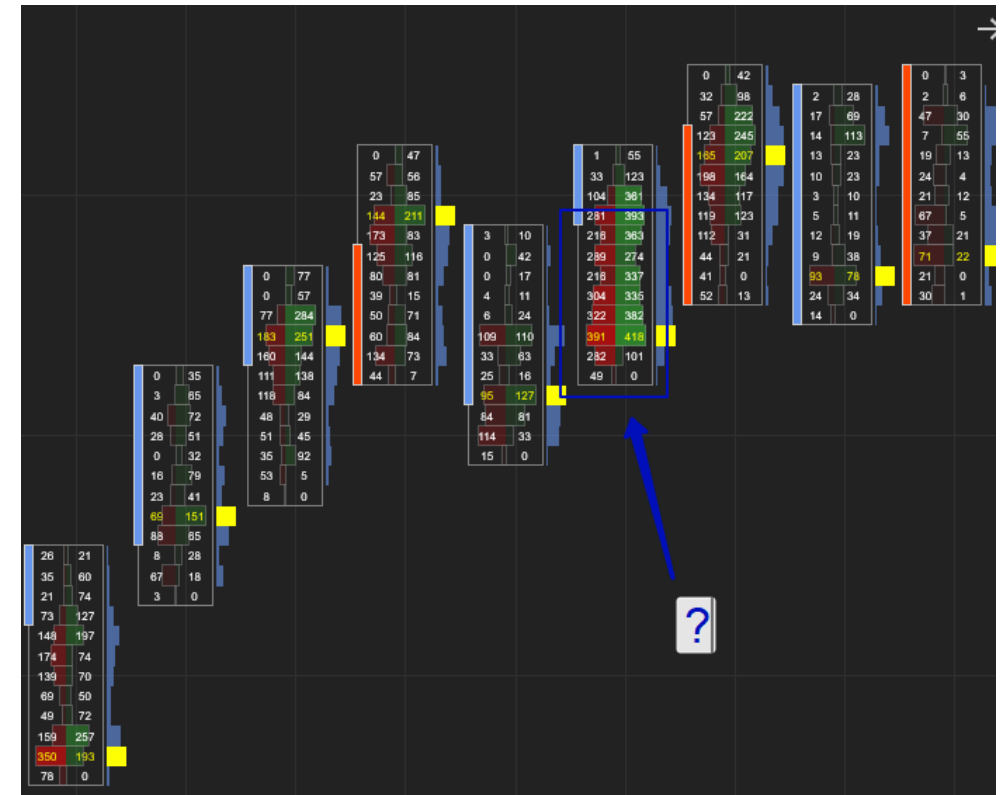
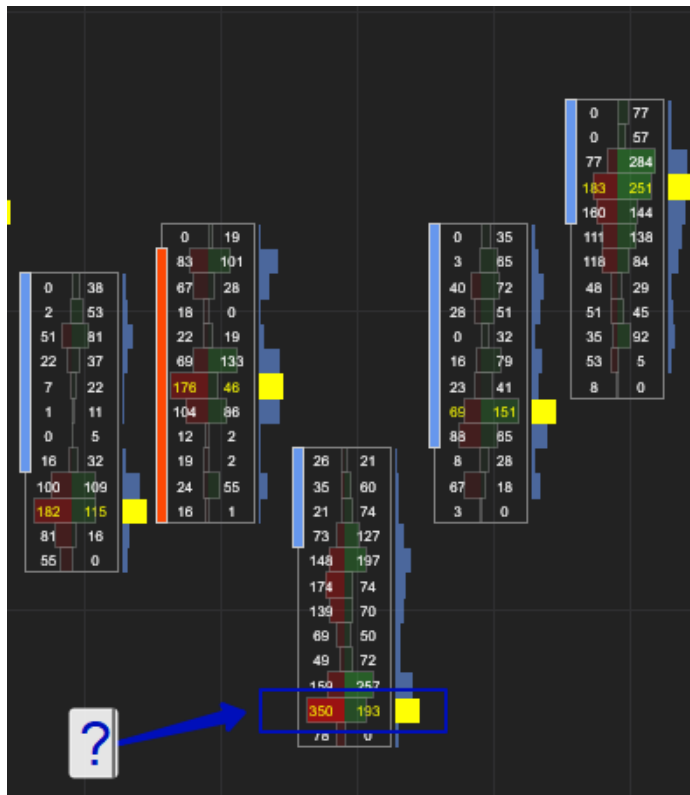
Как видно, активность и покупателей, и продавцов здесь была не очень высокой, но как только цена опустилась до отмеченного уровня – то все рыночные продажи были «поглощены» лимитными заявками на покупку, что и привело к образованию крупного объема в 350 контрактов по биду.

И сразу после этого произошел откат вверх, что говорит о том, что всю активность рыночных продавцов поглотил какой-то крупный лимитный покупатель. Также появились хорошие объемы по аску (193 и 257 внизу бара), что тоже является показателем того, что на этом уровне кто-то активно закупается, то есть толкает цену вверх рыночными ордерами на покупку и при этом не пускает вниз лимитными ордерами на покупку.

Соответственно, такие паттерны – это хорошие точки входа в лонг (т.е. с расчетом на то, что цена будет расти и дальше, потому что рыночные покупатели будут и дальше активно разбирать лимитки на продажу, и при этом лимитные покупатели будут сдерживать рыночных продавцов, то есть не будут давать им двигать цену вниз).

А что происходит здесь? На первый взгляд может показаться, что ситуации похожи – тоже был откат, внизу его встретили лимитки на покупку (это видно по левому столбцу, по бидам), и цена откатилась обратно.

Но! Стоит только сравнить активность продавцов в первой ситуации и в этой ситуации, и станет ясно, что это «две большие разницы». Если в первой ситуации активность продавцов была околонулевая, и ее «впитали» лимитки на покупку буквально на 1 уровне объемом на 350 контрактов, то здесь активность продавцов в разы выше, и более того – еще и резко выросшие объемы по аску (правый столбец, зеленый) показывают, что к борьбе присоединились еще и лимитные продавцы, которые пытаются не пустить цену вверх. Сравните с предыдущими барами – там практически не было ни лимитных продавцов, ни рыночных продавцов, а здесь и тех и тех просто толпа. Соответственно, такой сигнал стоит воспринимать как начало сильной борьбы между покупателями и продавцами, без четкого перевеса, а значит, из сделки стоит выйти, так как ситуация стала неопределенной.



## Практическое применение

Подведем итог. Задача модели машинного обучения – входить в сделки и выходить из них, анализируя объемы и цену.

Целевое значение матожидания прибыли  $\geq 2$  тиков.

Это значение обусловлено тем, что реальные сделки осуществляются роботом на C# в торговом терминале, и в реальной торговле всегда есть проскальзывание, т.е. исполнение рыночного ордера по худшей цене, чем была запланирована.

Соответственно, при  $EV = 2$  в реальности речь будет идти о +- нулевой прибыльности.

Паттерны, описанные в предыдущем блоке, будучи описанными в виде четкого алгоритма, обеспечивают  $EV = 2.7$  тика, в связи с этим есть надежда, что машинное обучение поможет улучшить эти показатели.

Также нужно учесть, что в случае положительных результатов при тестировании на исторических данных понадобится обеспечить шлюз, который сможет принимать данные из торгового терминала (формат вывода можно манипулировать как угодно, данные отдавать смогу так, как будет нужно) и в ответ отдавать информацию о том, что сейчас нужно зайти или выйти из сделки (формат и процесс передачи данных, опять же, может быть каким угодно, т.к. endpoint на C#).

Стоимость и сроки доработки по обмену данными прошу указывать сразу же, но это будет отдельная задача, к реализации которой приступим лишь в том случае, если в синтетических условиях модель покажет более интересные результаты, чем обеспечивает алгоритм, жестко описывающий паттерны и вход/выход из сделки.