### Содержание

[Содержание](#_vkt3z46pos0w) 1

[**Future move price prediction - data-driven approaches for predicting an asset price future move over a mid-term horizon**](#_3e8rhzit5z13) **1**

[Описание проекта](#_y61zb3b8yva2) 1

[Реализация торговой симуляции](#_tb7nu5z9h47r) 2

[Задание](#_evwbezhl1jnr) 3

[Пример расчета PnL по сделкам](#_dqi653wze548) 4

## Future move price prediction - data-driven approaches for predicting an asset price future move over a mid-term horizon

### Описание проекта

Задача берет корни в высокочастотной торговле, где для обычно необходимо решать две задачи:

1. pricing (price prediction, т.е. где в ближайшее время будет цена торгуемого актива)
2. ordering (какие нужно отправлять на рынок ордера для эксплуатации prediction).

В рамках проекта предлагается выполнять эти задачи в хронологическом порядке.

Предлагается использовать готовые (собранные) данные, которые представляют собой L1+L2 с нарезкой по секунде (5 уровней стакана) + агрегированные данные о том, что произошло внутри этой секунды. В данных присутствуют агрегаты по полям ask\_price\_i, bid\_price\_i (цены), ask\_quantity\_i, bid\_quantity\_i (объемы), trade\_price (цена трейда), trade\_quantity (объем трейда) - с разбивкой на Buy / Sell, count - количество событий внутри агрегации, price - колонка с текущей ценой. Необходимо предсказать цену через N шагов.

Состав данных может меняться по ходу течения проекта, но структура останется такой же, т.е. предполагается, что все наработки на старых данных будут применимы к новым (и чем лучше результат на старых данных, тем лучше на новых, новые данные могут содержать дополнительную информацию для улучшения моделей).

Примерный план работы (и примерные сроки реализации):

1. Разработка предсказательной модели с наилучшей точностью (2 месяца)
   1. Бизнес смысл - построить модель с наилучшей точностью в целях получения доходности, превосходящей издержки (величина деградации сигнала в связи с задержкой и уплаченные комиссии биржи)
   2. Подзадача будет проходить в виде kaggle-like соревнования, в котором участникам будет предоставлен train/val датасет, а на лидерборде будет виден скор по части test датасета (скор на полном датасете - при завершении соревнования).
   3. Соревнование будет проходить путем загрузки обученных участниками model inference в систему EvalAI, внутри (скрыто от участников) будет проходить inference согласно предоставленной модели на новых данных (недоступных во время train) и получение скора для ранжирования участников на лидерборде
2. Реализация торговой симуляции (1 месяц)
   1. Симуляция на исторических данных гипотетических трейдов, сделанных согласно модели, и анализ различных метрик (PnL, mark out)
3. В качестве задачи со звездочкой - добавление второго датасета (с другого актива), с которым проделать следующие подзадачи (1-2 месяца):
   1. Применить 1. и 2. для второго актива (т.е. будет 2 модели)

Улучшить 1. и 2. на обоих активах, имея данные с обоих активов в распоряжении

* 1. Построить одну модель, которая будет работать на совокупных данных обоих активов и делать предсказания сразу для двух активов. Точность предсказаний с одной модели должны быть как минимум не хуже точности предсказаний индивидуальных моделей

В рамках проекта предполагается работа над моделью - инжиниринг признаков, выбор архитектуры модели (возможно попробовать что-то из решений-победителей опубликованных с прошлых трейдинговых соревнований на кагле, попробовать что-то из статей), правильное обучение и кросс-валидация, регуляризация, проверка перформанса на unseen данных (как с точки зрения ML-метрик, так с точки зрения «бизнес» результата, т.е. PnL или mark out), исследование задержек и комиссий в целях определения наихудших условий (наибольшая задержка и наибольшая комиссия), при которых наблюдается положительный перформанс стратегии

### Реализация торговой симуляции

Предлагается реализовать простую стратегию на бэктестере, которая будет совершать сделки в случае, если предсказанная будущая цена достаточно сильно превышает текущую цену в стакане.

А именно, предлагается совершать покупку если предсказанная цена отличается от лучшего ASK более чем на порог Т, продажу - если предсказанная цена отличается от лучшего BID более чем на порог Т. Предполагается, что сделка совершается по цене лучшего ASK или лучшего BID.

Сделка совершается сайзом S, при достижении позиции M разрешаются сделки только в обратную сторону (риск-менеджмент). За совершение любой сделки платится комиссия C базисных пунктов.

Меняя T можно совершать больше / меньше сделок. По совершенным (просимулированным на исторических данных) сделкам следует рассчитать PnL, сводя друг с другом открывающие и закрывающие сделки на покупку / продажу. Варьируя T и совершая достаточно большое количество сделок можно проверить, достаточна ли предсказательная сила у построенной модели, чтобы рассчитанный по сделкам PnL был положительным. Также с помощью симулятора можно проводить сравнение друг с другом разных моделей, производя сделки на симуляторе с помощью каждой из них и сравненивая PnL.

По результатам прогона стратегии через бэктест следует собрать различные метрики, по которым можно измерять производительность стратегии. К примеру, такими метриками могут быть PnL, Sharpe ratio, количество сделок, различные визуализации (предлагается подумать, какие метрики и графики могут быть полезными для анализа).

Дополнительные вопросы:

* Исследовать, стоит ли менять логику стратегии\* на каждом шаге в зависимости от того, в какой позиции на этом шаге она находится.
* В случае если даже при малых значениях T сделок совершается не очень много - подумать, можно ли что-то изменить в логике стратегии\*, чтобы увеличить количество совершаемых сделок.
* Исследовать бэктест в случае наличия комиссий за сделку, варируя их от 0 до 7 базисных пунктов с шагом 1 - оптимизировать частоту совершения сделок для каждого уровня комиссий.

\*логика стратегии - алгоритм, по которому определяется когда, по какой цене, какие сделки совершать

### Задание

Язык программирования - Python.

1. Написать программу, реализующую стратегию с параметрами T, S, M, C описанную выше
   1. на вход программе подаются
      1. временной ряд рыночных цен (аски и биды первого уровня стакана)
      2. предсказанная будущая цена по модели в каждый момент времени
      3. значения параметров T, S, M, C
   2. на выходе програма должна выдавать датафрейм с совершенными сделками со следующими колонками:
      1. Timestamp
      2. Цена сделки
      3. Объем сделки
      4. Направление сделки (покупка или продажа)
2. Реализовать модуль анализа совершенных сделок, рассчитывающий ключевые метрики, которые должны включать:
   1. Итоговый PnL
   2. Размер уплаченных комиссий
   3. Количество сделок
   4. Объем торгов
   5. Sharpe ratio
   6. Средневзвешенный PnL за сделку
   7. Средневзвешенная комиссия за сделку
   8. Пропорция сделок с положительным и отрицательным PnL
   9. Максимальная просадка PnL за период
3. Реализовать модуль, производящий графики анализа торгов, включающий:
   1. Временной ряд цен, по которым совершаются сделки
   2. Временной ряд рыночных цен
   3. Временной ряд предсказанной цены
   4. Кумулятивные (по сделкам)
      1. PnL до комиссии
      2. PnL после комиссии
      3. Позиция
      4. Объем торгов
   5. Распределение PnL сделок
4. Найти в интернете релевантные статьи и дополнить предыдущие два пункта метриками и графиками (со ссылками на статьи), которые рассчитываются в этих статьях, но не перечислены выше
5. Оптимизировать параметры T, S, M, C путем прогона программы из п. 1 на разных параметрах и используя в качестве целевых метрик
   1. Итоговый PnL (без учета комиссий и с учетом комиссий)
   2. Средневзвешенный PnL (без учета комиссий и с учетом комиссий)
   3. Sharpe ratio
6. Собрать все метрики и графики по итогам п. 2-5 в отчет (формат - Jupyter Notebook или PDF)
7. Подставить вместо предсказанной цены истинную будущую цену и повторить п. 2-6

### Пример расчета PnL по сделкам

https://drive.google.com/file/d/1MbafZjxEE2\_EpwbUlB4L\_-qvA0Xuoi6U/view?usp=sharing

По ссылке представлен датафрейм с примером совершенных сделок и расчета PnL. Датафрейм имеет вид как на скриншоте ниже и содержит следующие колонки:

* ask - рыночная цена аска в момент совершения сделки
* bid - рыночная цена бида в момент совершения сделки
* trade\_price - цена совершенной сделки
* trade\_quantity - объем совершенной сделки
* side - сторона сделки (покупка или продажа)
* pnl - реализованный PnL по формуле (close\_price - open\_price) \* quantity

Замечание: для расчета реализованного PnL считается, что открывающая и закрывающая сделки совершены одинаковым объемом, однако на практике закрывающая сделка может быть совершена объемом меньше или больше, чем открывающая - тогда помимо реализованного PnL возникает mark-to-market PnL (от незакрытой части), который станет реализованным после того, как вся незакрытая часть будет закрыта. В прикрепленном примере в колонке pnl указана только реализованная часть PnL, нереализованная (mark-to-market) часть “переносится” на последующие сделки.

