

1. Техническое задание

Задание состоит из 2 частей:

- Вам нужно определить собственные движения цены фьючерса ETHUSDT, исключив из них движения вызванные влиянием цены BTCUSDT. Опишите, какую методику вы выбрали, какие параметры подобрали, и почему (обоснование можно оформить в README).
- Напишите программу на Python, которая в реальном времени (с минимальной задержкой) следит за ценой фьючерса ETHUSDT и используя выбранный вами метод, определяет собственные движение цены ETH. При изменении цены на 1% за последние 60 минут, программа выводит сообщение в консоль. При этом программа должна продолжать работать дальше, постоянно считывая актуальную цену.

2. Выбор модели

Для определения собственных движений цены фьючерса ETHUSDT, исключив влияние цены BTCUSDT, можем использовать метод ковариации или корреляции. Так величины всего две и сложные методы типа метода градиентного спуска нам не понадобятся.

Выбор методики:

- Ковариация: Измеряет степень изменения двух переменных вместе. Положительная ковариация указывает на положительное взаимосвязанное изменение, а отрицательная на обратное.
- Корреляция: Нормализованная мера взаимосвязи между двумя переменными. Корреляция принимает значения от -1 до 1, где 1 указывает на положительную линейную связь, -1 на отрицательную, и 0 на отсутствие связи.

Параметры:

- Соберем временные ряды цен ETHUSDT и BTCUSDT.
- Выберем временной интервал для анализа (например, несколько минут, час, день). Длинные исторические ряды нам не помогут уточнить быстрые движения.
- Рассмотрим несколько периодов, чтобы учесть краткосрочные тренды.

3. Анализ данных (запускаемый файл research.py)

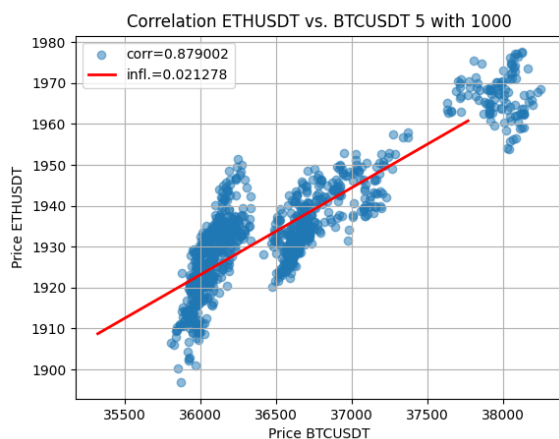
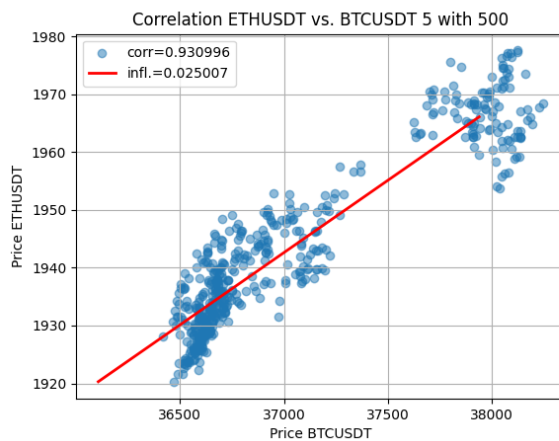
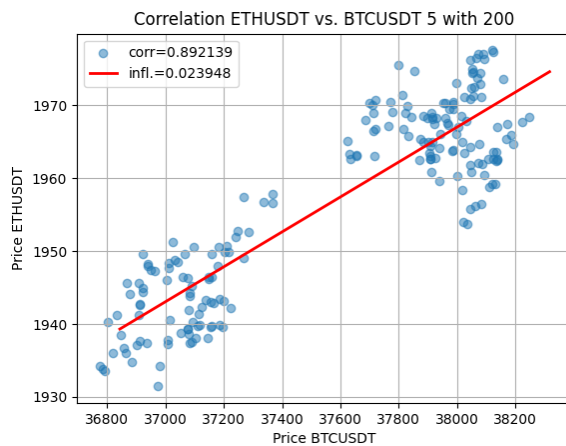
В данном проекте интуитивно угадывается, что связь между величинами существует и достаточно сильна. Исследуем данное предположение с помощью расчета линейного коэффициента корреляции (Пирсона)

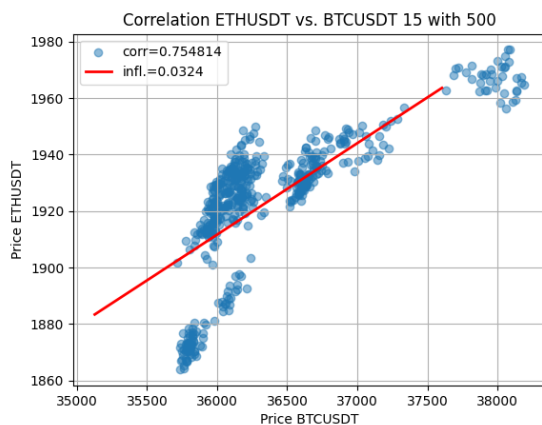
$$r = \frac{COV_{xy}}{s_x s_y} = \frac{\sum (X_i - \bar{X})(Y_i - \bar{Y})}{(n-1)s_x s_y} = \frac{\sum (X_i - \bar{X})(Y_i - \bar{Y})}{\sqrt{\sum (X_i - \bar{X})^2 \sum (Y_i - \bar{Y})^2}}$$

Коэффициент корреляции показывает тесноту линейной взаимосвязи и изменяется в диапазоне от -1 до 1. -1 (минус один) означает полную (функциональную) линейную обратную взаимосвязь. 1 (один) – полную (функциональную) линейную положительную взаимосвязь. 0 – отсутствие линейной корреляции (но не обязательно взаимосвязи). На практике всегда получаются промежуточные значения.

Соберем несколько выборок разной длительности и оценим коэффициенты корреляции для каждой из них. Будем опираться на данные по закрытиям «свеч» для интервалов 5, 15 и 360 при глубинах выборки 200, 500 и 1000 значений.

Для прогнозирования цен будем применять одну из регрессионных моделей (линейная, нелинейная, многофакторная). В нашем случае подойдет линейная регрессионная модель если корреляция между величинами будет достаточно сильной (коэффициент более 0,72). Линейная регрессионная модель даст нам коэффициент наклона в линейной функции типа $y = A + Bx$. Наложим эту кривую на наши выборки.

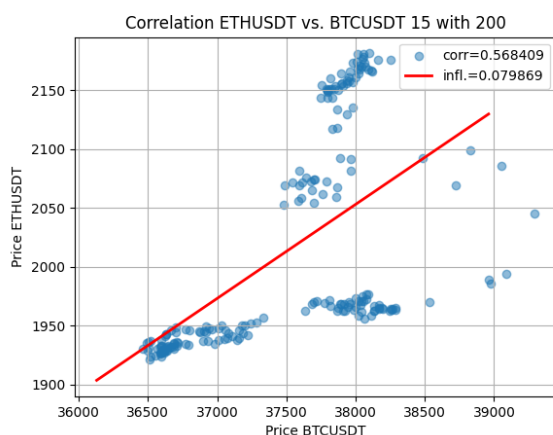




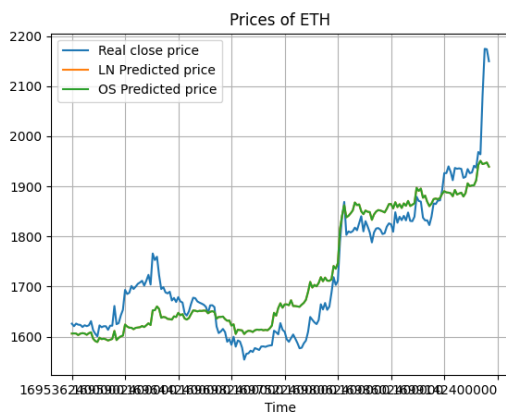
Видим что коэффициенты корреляции выше 0,7 что в целом характеризует устойчивую зависимость одной величины от другой.

Однако нужно отметить что коэффициенты для разных выборок достаточно сильно различаются. Так что в дальнейшем имеет смысл динамически подбирать глубину выборки для получения более устойчивой модели.

Пример – при активном движении цен наблюдаются случаи достижения коэффициента корреляции порядка 0,55 что говорит о достаточно слабой взаимосвязи двух величин (большом разбросе) и даст большую ошибку при прогнозировании цены для линейной регрессионной модели.



В целом линейная регрессия дает весьма неплохой предсказательный результат



4. API для получения данных

Для получения данных в реальном времени можем использовать API любой из криптовалютных бирж. На настоящий момент выбран API биржи ByBit (предоставляет данные в реальном времени и исторические данные при наличии ключа для АПИ). В настоящий момент реализован механизм http-запросов с темпом опроса раз в секунду, в последствии механизм может быть реализован через websocket, темп обновления информации при такой работе в целом соизмерим с имеющимся.

При включенном режиме динамической перестройки модели в зависимости от глубины и частоты выборки данных (по наилучшему коэффициенту корреляции) потребуется доступ к API с предоставлением данных за прошедший период. В этом случае нам необходим ключ доступа к API ByBit (см. раздел установка и конфигурирование)

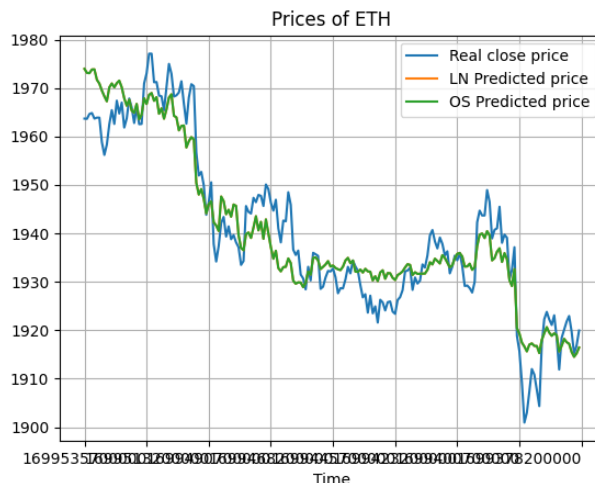
5. Инструменты

Для расчета прогностических цен воспользуемся доступными python-библиотеками

statmodel.api

sklearn

Для динамического подбора выборки имеет смысл асинхронно в одном потоке обрабатывать текущие замеры и в другом периодически сверять действующую глубину выборки для выбора наиболее эффективной.



Обе библиотеки обеспечивают одинаковый результат – LN и OS на графике полностью накладываются друг на друга.

6. Установка и конфигурирование

Описаны в файле readme.md