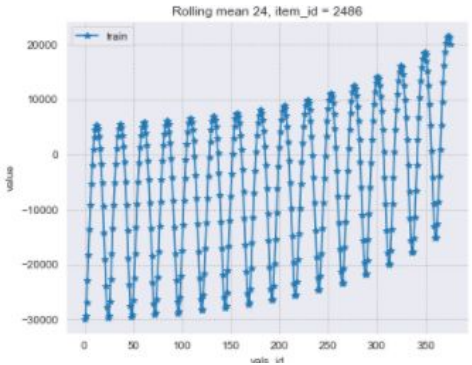
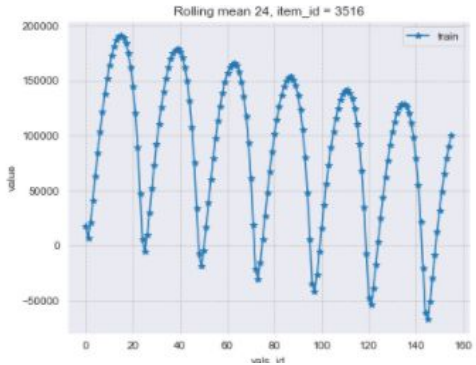
C1. Прогнозирование временных рядов

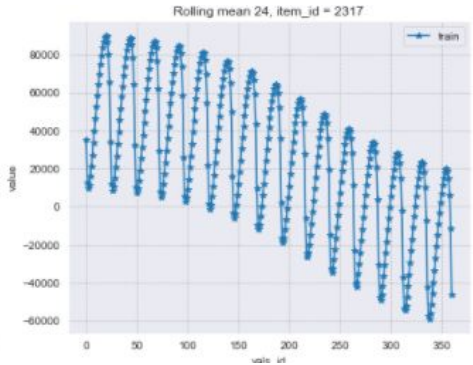
Ozon Masters ML2
Студент: Арешин Станислав Олегов

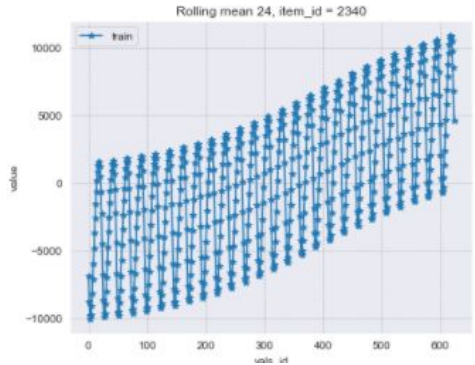
Постановка задачи

- Требуется предсказать следующие значения периодического временного ряда.
- Метрика SMAPE
- Всего 3600 временных рядов в обучении и тесте.
- В обучении 3115734 значений.
- В тесте 780762 значения



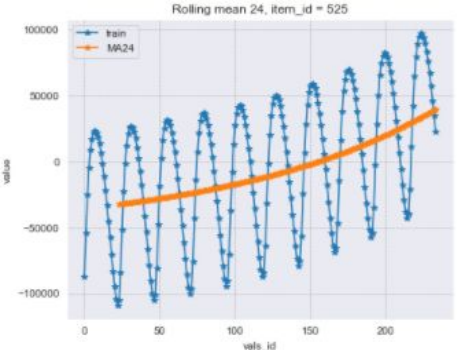


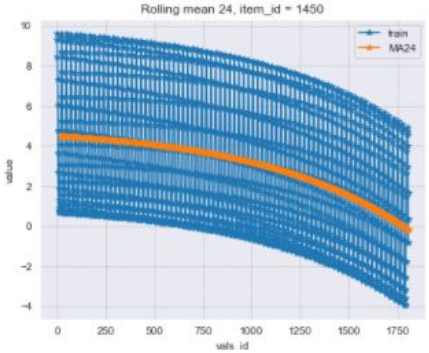


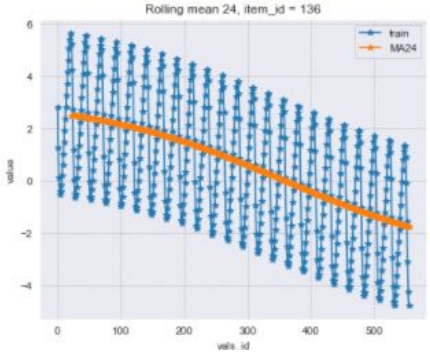


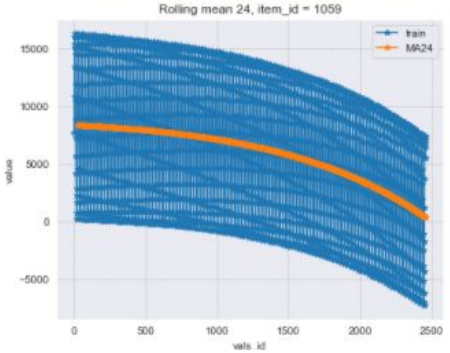
Анализ данных. Скользящие статистики.

Скользящее среднее, окно 24

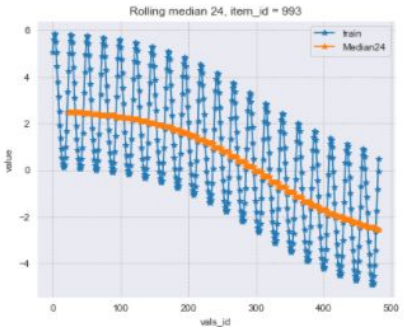


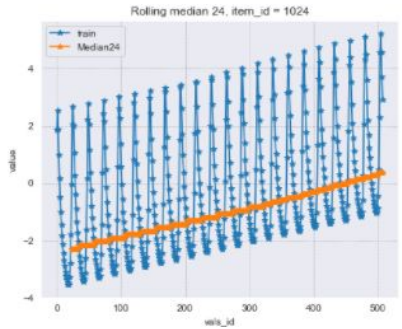


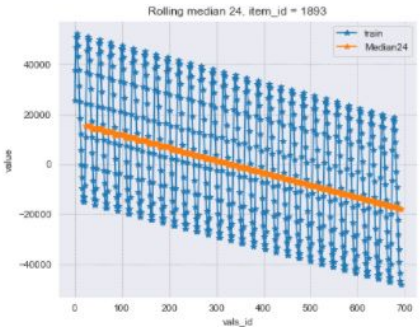


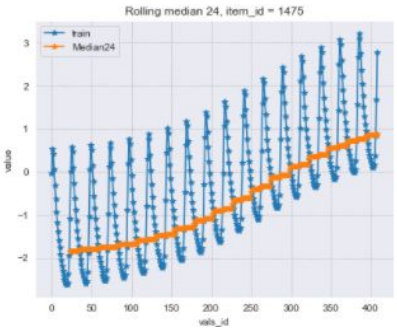


Скользящая медиана, окно 24



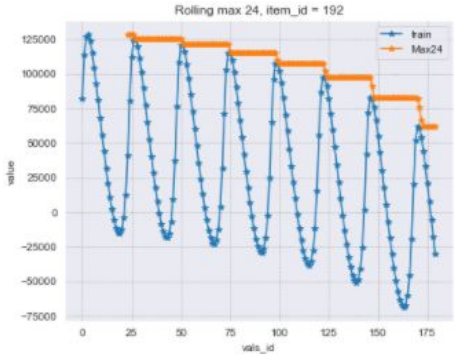


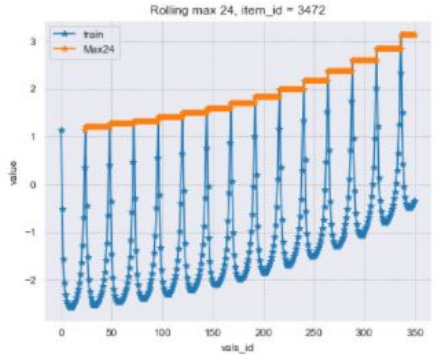


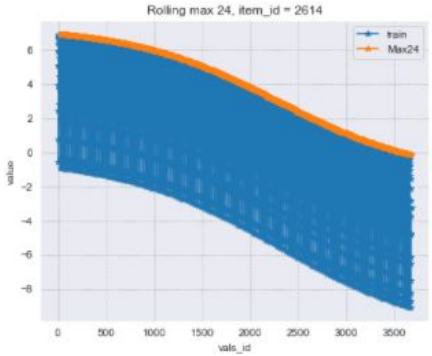


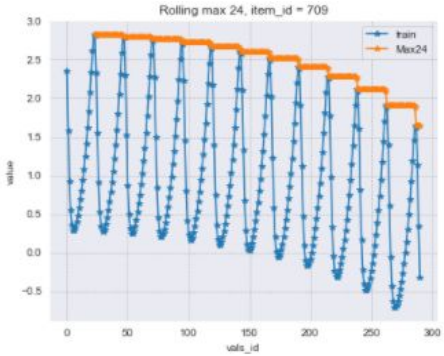
Анализ данных. Скользящие статистики.

Скользящий максимум, окно 24

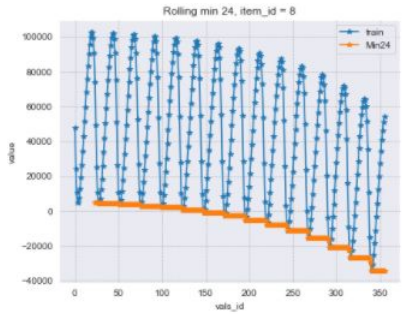


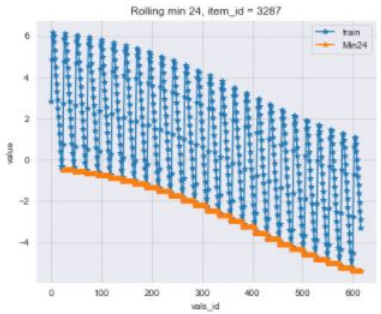


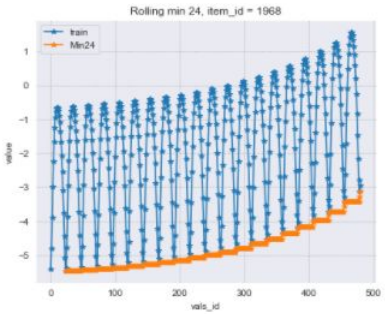


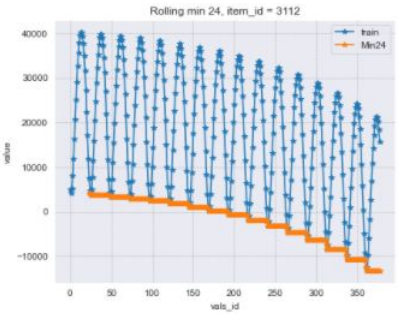


Скользящий минимум, окно 24









Анализ данных. Итоги.

- Минимальная длина ряда 75, максимальная 5055, средняя 865, медиана 540.
- Период цикла 24 значения, для обучения стоит брать либо лаги цикла, либо чуть больше.
- Скользящее среднее с окном сглаживания 24 хорошо выделяет тренд.
- Тренд не всегда линейный, а скорее чаще всего нелинейный.
- Скользящая медиана хуже среднего, использовать не стоит.
- С помощью скользящего минимума и максимума можно ограничить ряд сверху и снизу, что в совокупности с выявленным трендом задает рамки для рядов.
- Много похожих между собой рядов.
- Как идея - брать максимально возможное количество лагов в зависимости от длины ряда.

Первые попытки. Классические алгоритмы.

- SARIMA

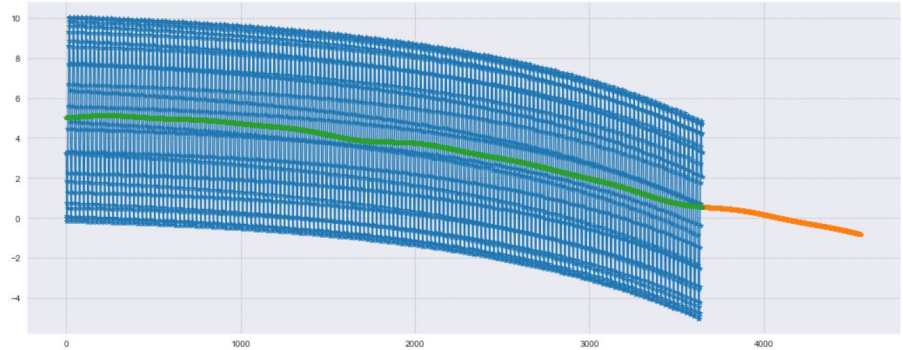
Результат: отдельные ряды прогнозирует хорошо, если подобрать параметры.

Проблема: невозможно подобрать универсальные параметры, которые подходили бы каждому ряду, сложно и долго подбирать параметры для каждого ряда отдельно, долго работает.

- FB Prophet

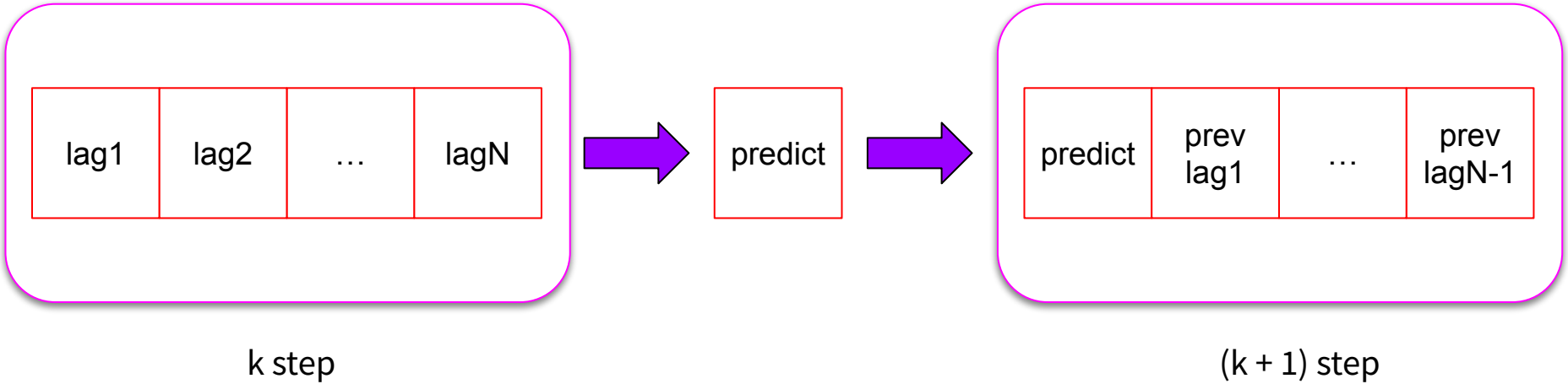
Результат: плохой прогноз.

Проблема: кажется, проблема кроется в автонастройке параметров prophet, так как встроена настройка сезонности по дням, неделям и годам, а формат данных решаемой задачи этого не предполагает. Пример прогноза представлен ниже:



Основная идея

Основная идея алгоритма заключается в схеме предсказаний. Возьмем N лагов как признаки для обучения. На k-ом шаге предсказания получаем значение, добавляем его как первый лаг для k+1 предсказания, остальные лаги сдвигаем.



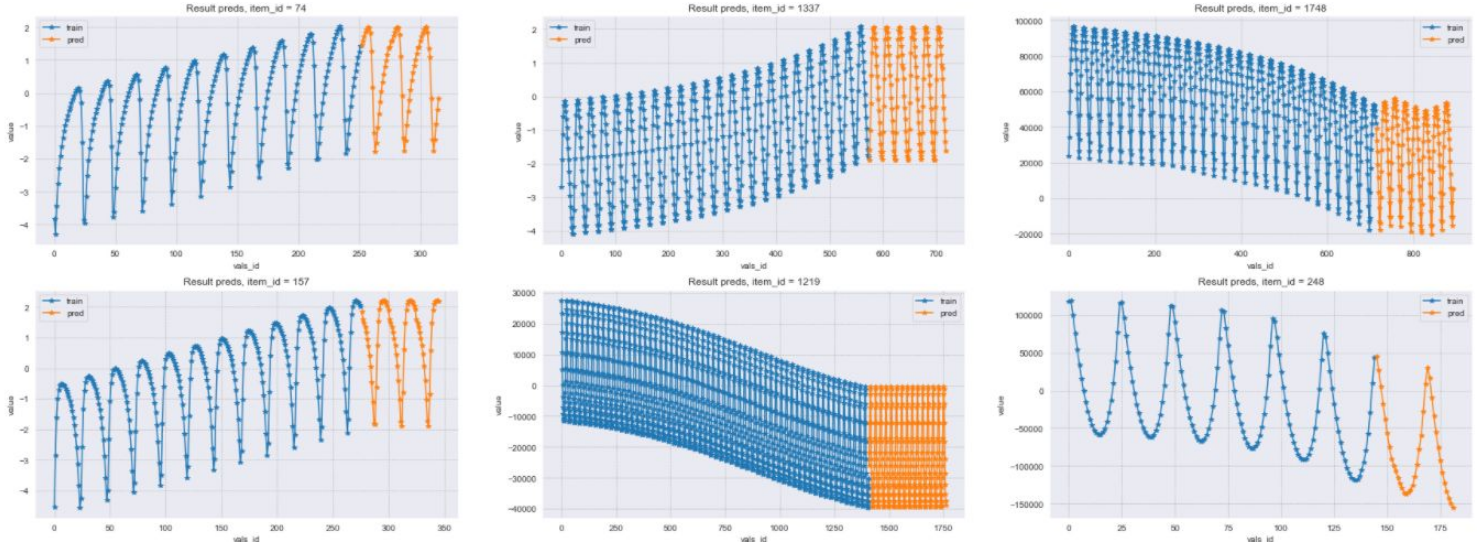
Первые попытки. Алгоритмы ML.

- Один алгоритм для всех рядов. Градиентный бустинг (Catboost), 24 лага.

Результат: константные решения для большей части рядов -> плохой скор.

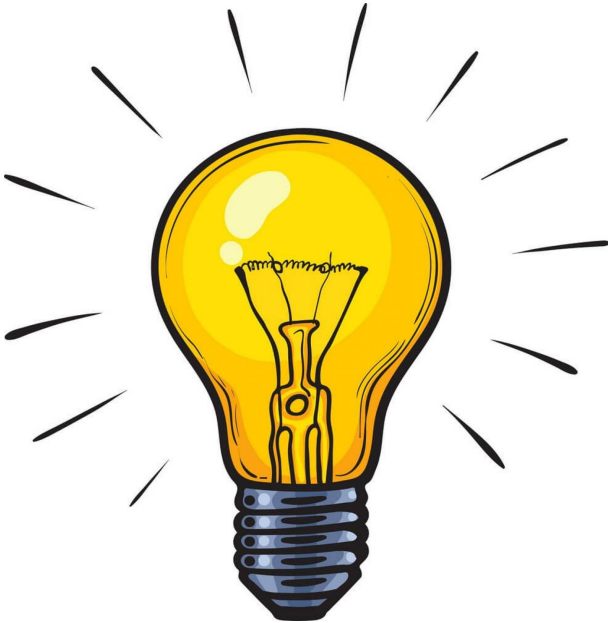
- Один алгоритм для всех рядов. Метод k ближайших соседей, k=3, взвешенный алгоритм, 30 лагов.

Результат: неплохой прогноз, но не улавливает тренд рядов -> плохой скор. Ниже представлены несколько графиков:



Переломный момент

А что если обучать простую линейную регрессию на лагах отдельно для каждого временного ряда? Это работает !

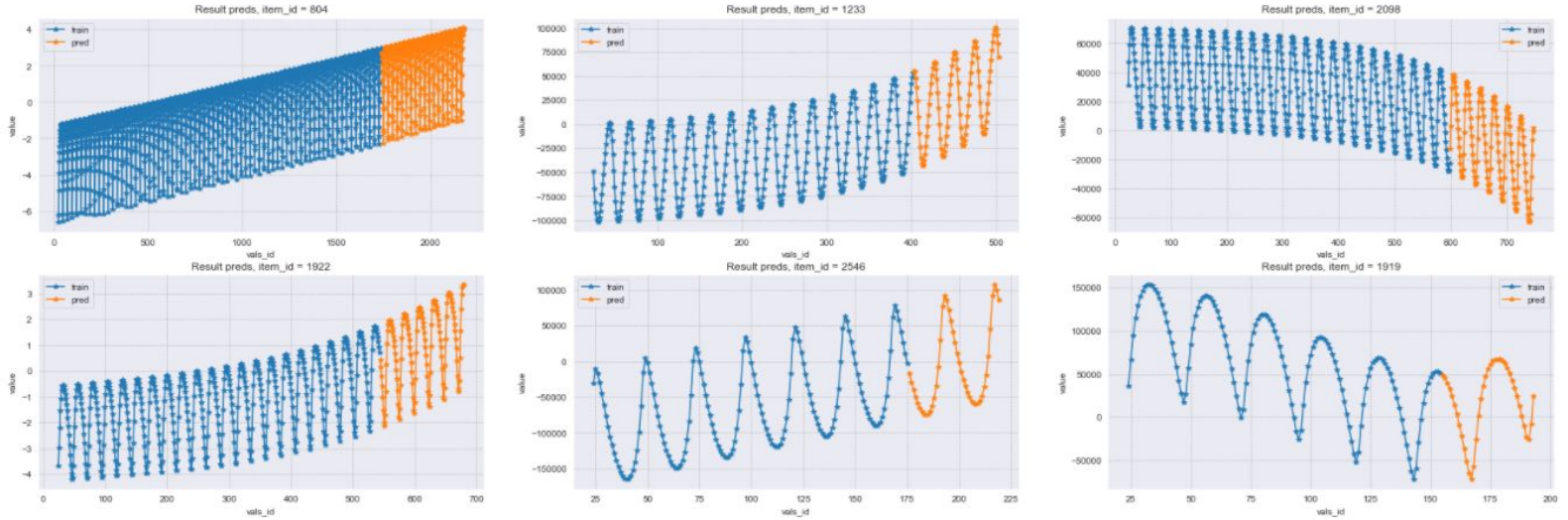


Baseline 1. Алгоритм.

- Используем 24 лага.
- Для выделения тренда обучается линейная регрессия на vals_id и vals_id^2 (для выделения нелинейности), в качестве целевой величины используем value. Такая операция проводится для каждого ряда, делаются предсказания на тест и полученные значения используются в качестве признаков.
- Аналогичные действия проводим, используя min24 и max24 как целевую величину, прогнозируем на будущее и используем как признаки.
- Таким образом, с помощью ограничиваем диапазон каждого ряда.
- Так как признаков довольно много, используем Ridge регрессию.
- Используем предложенную выше схему предсказаний, обучая под каждый ряд свою модель Ridge регрессии.

Baseline 1. Результаты.

Скор решения на паблике 15.92324. По графикам видно, особенно если позапускать несколько раз, что решение не идеальное и некоторые ряды утягиваются за счёт скорее всего признаков тренда, скользящего минимума и максимума, но при этом решение пробивает первый бейслайн. Нужно дальше развивать идею, посмотрим, что получится.

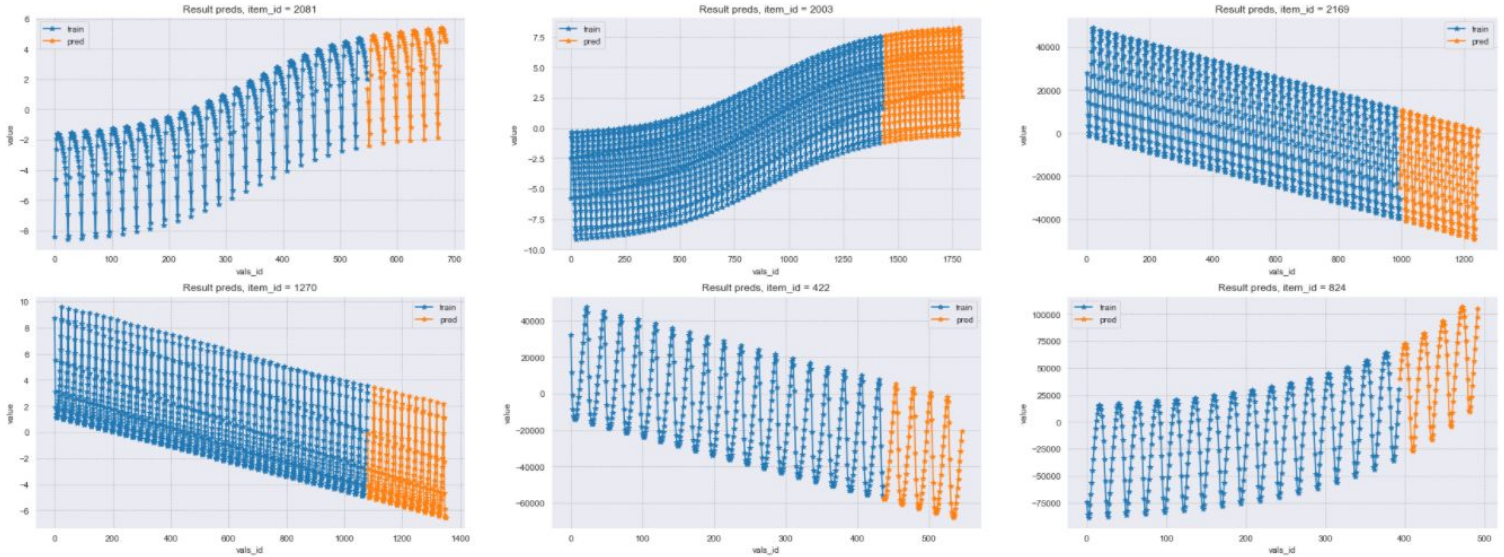


Baseline 2. Алгоритм.

- В ходе экспериментов выявлено, что чем больше лагов брать, тем лучше становится решение. Минимальная длина ряда 75 значений, поэтому было решено для коротких рядов (длиной меньше 100 значений) использовать 24 лага, а для длинных использовать 48 лагов.
- Было решено отказаться от признаков trend, min(window) и max(window), так как они утягивают некоторые ряды и нарушают их поведение.
- Линейная регрессия при данном подходе дает результаты точнее Ridge регрессии, поэтому используем её.
- Используем предложенную выше схему предсказаний, обучая под каждый ряд свою модель линейной регрессии.

Baseline 1. Результаты.

Скор решения на паблике 3.17186, итоговое место 9/74. При нескольких запусках видно, что не все временные ряды прогнозируются идеально, но на плохой прогноз надо постараться наткнуться. Результат гораздо лучше, чем был при первом успешном решении.



Итог

- Главный минус данного подхода - очень долгий процесс предсказания (лучшее решение на моём слабеньком ноутбуке предсказывало 4 часа), но это можно попробовать оптимизировать.
- Итоговый результат в топ 10 на паблике, что вполне достойно для такого простого решения.
- Дополнительно можно было попробовать отдельно настраивать алгоритм на ряды, у которых большое значение метрики SMAPE на обучении, чтобы улучшить прогноз таких рядов.
- Мне кажется, что из своего решения я получил практически максимум, для улучшения скора нужно думать над другими подходами.