



Факультет компьютерных наук

Машинное Обучение и  
Высоконагруженные Системы

Стратегии предсказания временных  
рядов на несколько точек  
вперед

# tsururu или Стратегии предсказания временных рядов на несколько точек вперед

Команда:

- Костромина Алина\* @elineii \*Sber AI Lab
- Ларченков Михаил @laplan
- Зуйкова Ольга @s21\_fernando
- Карагодин Никита @Einstein\_30

Куратор: Романенко Алексей





## Мотивация

- В области time series forecasting много прикладных задач, но относительно мало исследований.
- Существует множество уникальных методов и подходов (статистические, ML, DL).
- Прослеживается тесная связь с другими областями анализа данных.
- Финансы, производство, социальные и экономические исследования, климатология и многие другие области постоянно обращаются к анализу временных рядов.

## Временные ряды – это важно! (и интересно)





## Представляем tsururu:

- Инструмент для прогнозирования одномерных и многомерных временных рядов.
- 3 режима предсказаний: local-, global-, multivariate-modelling.
  - local-modelling: одна модель на один ряд.
  - global-modelling: одна модель на все ряды.
  - multivariate-modelling: одна модель на все ряды, при этом используются кросс-признаки (лаги рядов являются признаками друг для друга).
- Аккуратная предобработка ряда (трансформеры можно применять на ряд как при создании как признаков, так и целевых значений).
- 5+ стратегий.
- Совсем скоро выпуск в open-source (поэтому любые советы очень ценны)!

! Для получения доступа пришлите e-mail в TG @elineii

```
class TransformersFactory:
    def __init__(self):
        self.models = {
            "StandardScalerTransformer": StandardScalerTransformer,
            "LabelEncodingTransformer": LabelEncodingTransformer,
            "OneHotEncodingTransformer": OneHotEncodingTransformer,
            "LastKnownNormalizer": LastKnownNormalizer,
            "DifferenceNormalizer": DifferenceNormalizer,
            "TimeToNumGenerator": TimeToNumGenerator,
            "DateSeasonsGenerator": DateSeasonsGenerator,
            "LagTransformer": LagTransformer,
        }
```

Предобработка ряда

Признаки от времени

Работа с категориальными признаками

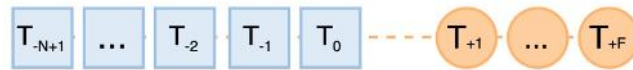
Прошлые значения ряда

+ поддержка экзогенных признаков

```
class StrategiesFactory:
    def __init__(self):
        self.models = {
            "RecursiveStrategy": RecursiveStrategy,
            "DirectStrategy": DirectStrategy,
            "MIMOStrategy": MIMOStrategy,
            "DirRecStrategy": DirRecStrategy,
            "FlatWideMIMOStrategy": FlatWideMIMOStrategy,
        }
```

## Стратегии предсказаний

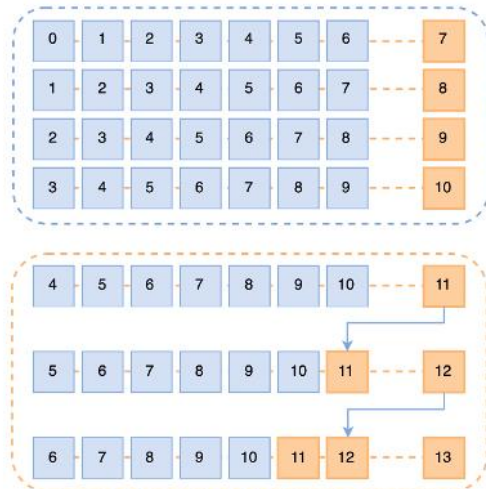
Общая постановка задачи регрессии:



Пусть  $N = 10$ ;  $F = 3$ ;  $H$  (history) = 7:

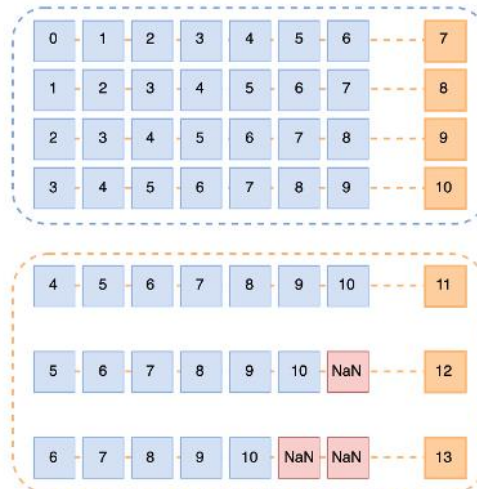


### Recursive strategy



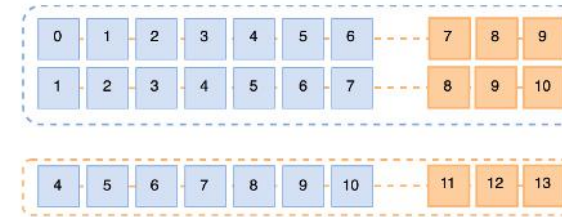
- ARIMA, ETS
- RecursivePipeline (ETNA)
- LGBM (Merlion)

### Reduced direct strategy

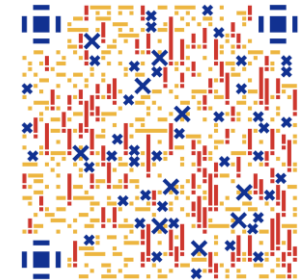


- DirectPipeline (ETNA)

### MIMO (multi-input-multi-output) strategy



- ANNs



Другие стратегии доступны по ссылке

Примеры библиотек и методов,  
которые основаны на этих  
стратегиях



## Наша команда, роли и задачи

- Общая цель: доработать и улучшить tsururu: open-source решение для анализа временных рядов.
- Главная цель: посмотреть на интересные подобласти анализа временных рядов и попробовать себя в них.



Карагодин Никита

Построит нейронку любой широты и глубины



Ларченков Михаил

Докажет, что даже простые бейзлайны дают результаты, а ансамблирование моделей – так вообще, лучшая вещь на свете



Зуйкова Ольга

Знает, что такое реконсильяция, сожмет и разожмет данные как угодно во времени



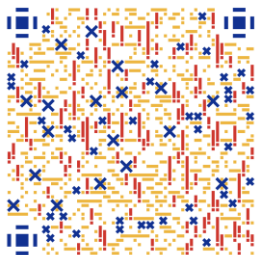
Костромина Алина

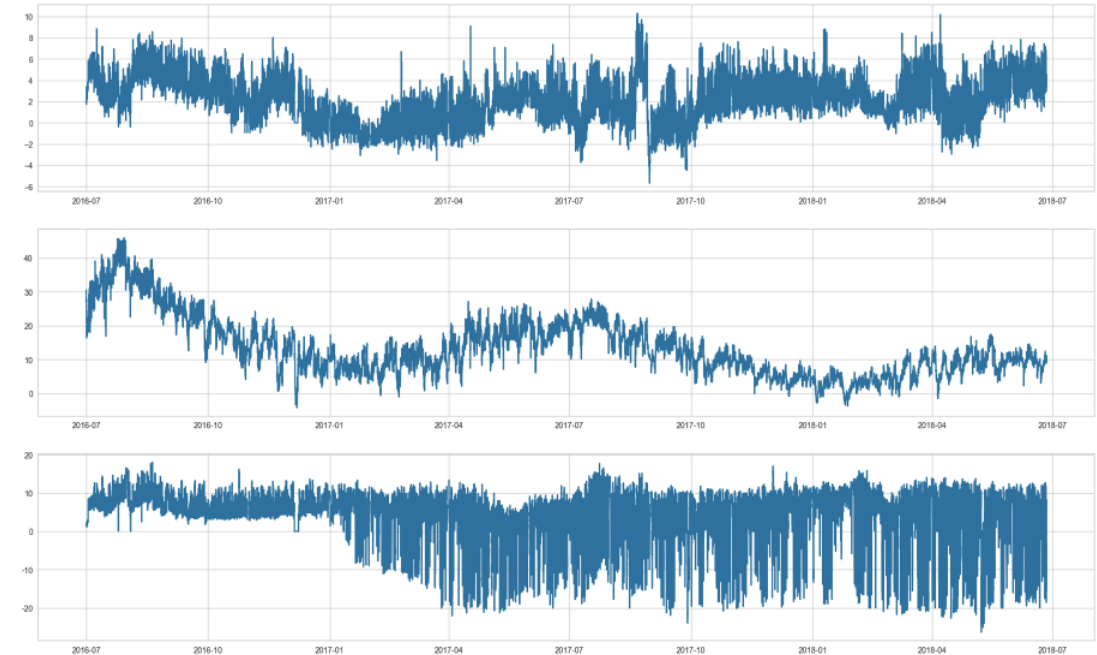
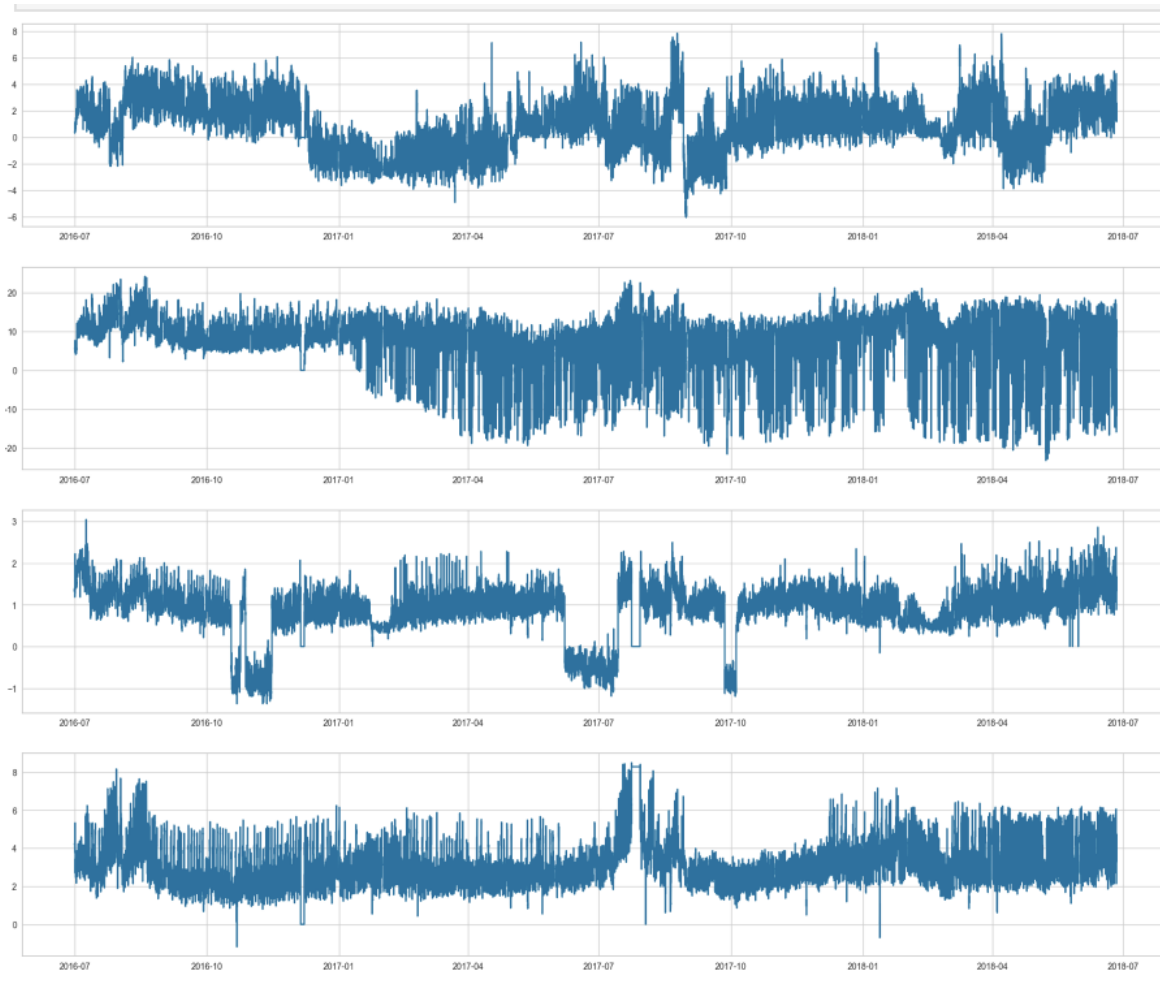
Соберет воедино командные усилия и постарается сделать библиотеку более дружелюбной к пользователям

## Прошлые результаты: Exploratory data analysis: 2 этапа

- Сделали анализ литературы по временным рядам и поняли, какие временные ряды чаще всего берут исследователи
- Отрисовали и проанализировали два больших бенчмарка с датасетами
- Подумали, какие паттерны во временных рядах могут быть потенциально интересны
- *Собрали сводную таблицу характеристик временных рядов*

- Для начала выбрали наиболее разнообразный, но при этом не слишком большой датасет ETTh1, связанный с электричеством



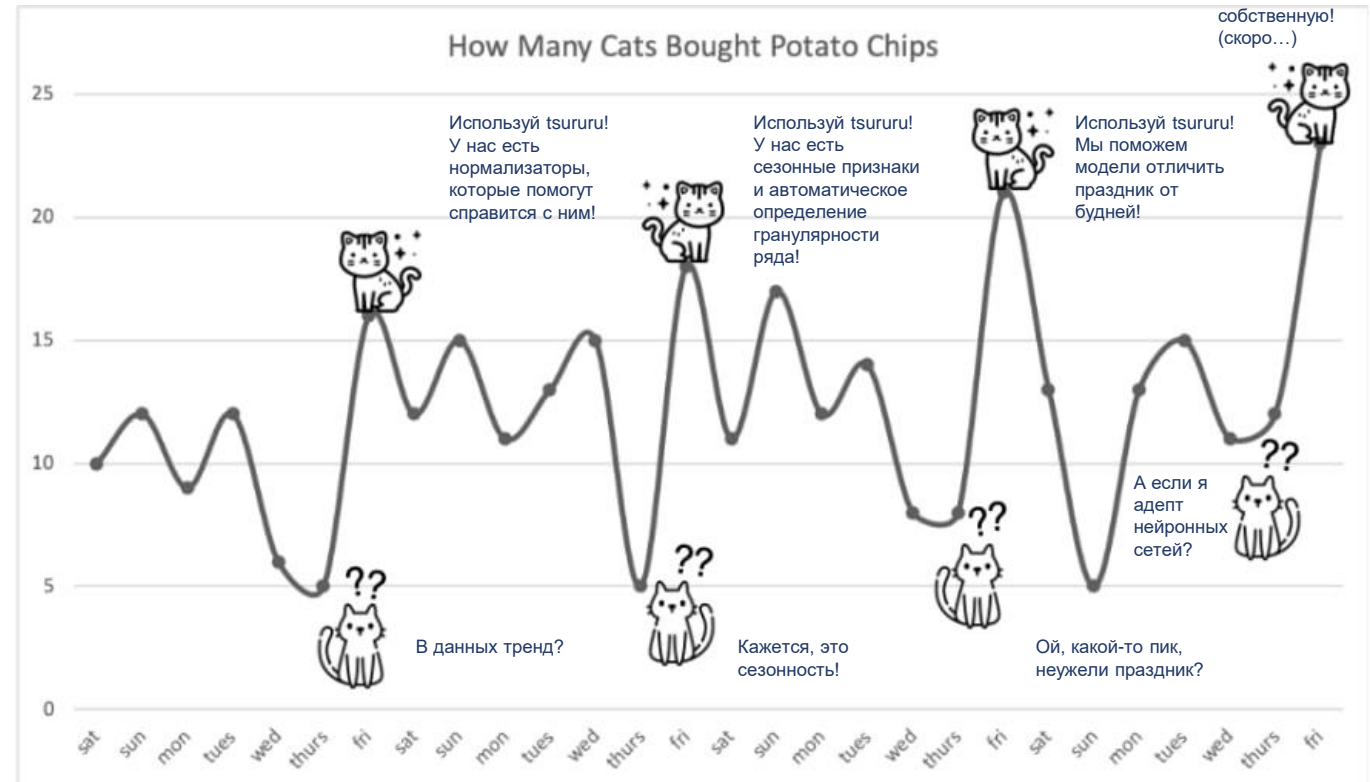


- Гранулярность: от 15 минут (что удобно для задачи иерархии по времени).
- Разнообразие паттернов: точки смены режима, локальные и глобальные тренды → интересно наблюдать, как разные модели справятся с такими трудностями.
- Минус: практически нет паттернов сезонности.



## Не бейте, что нет финальной ML модели

- Мы сосредоточились на том, чтобы не просто решить какую-то одну задачу, а развить универсальный инструмент для решения целого рода задач прогнозирования временных рядов.
- И тут нам не помешал бы совет, как лучше сделать? Хороший ли вариант выбрать ряд (в нашем случае ряд прогнозирования электричества) или несколько и показать, как можно добиться классного результата, используя tsururu (например, использовать нормализаторы, если у нас есть тренд или подобрать временные фичи, если у нас есть сезонность или использовать иерархию по времени, если мы хотим прогнозировать как на день, так и на месяцы вперед)?







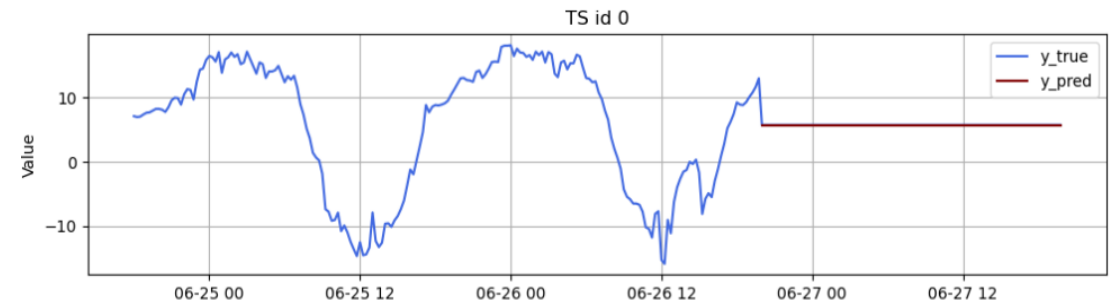
## Прошлые результаты: наивные методы (1)

- Имплементированы следующие методы:
  - Mean method (1)
  - Naive method (2)
  - Seasonal naive method (3)
  - Drift method (4)

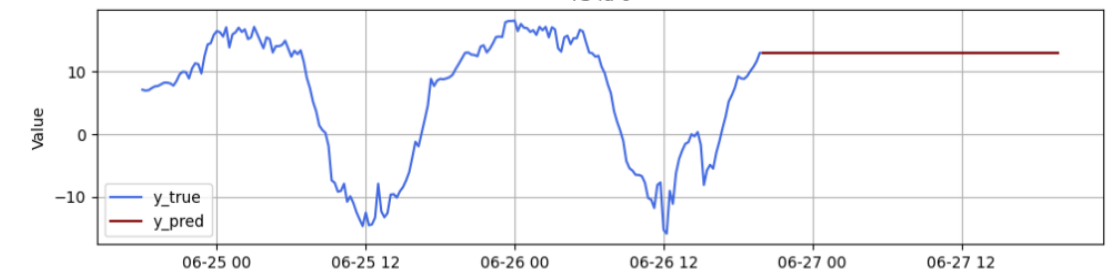
Ларченков Михаил (ветка `feature/new_models`)\*

\* Из-за того, что работа над частями проекта велась параллельно, то каждый участник команды вел разработку в своей ветке; на данный момент мы занимаемся финализацией и объединением результатов в общей ветке.

(1)



(2)



(3)



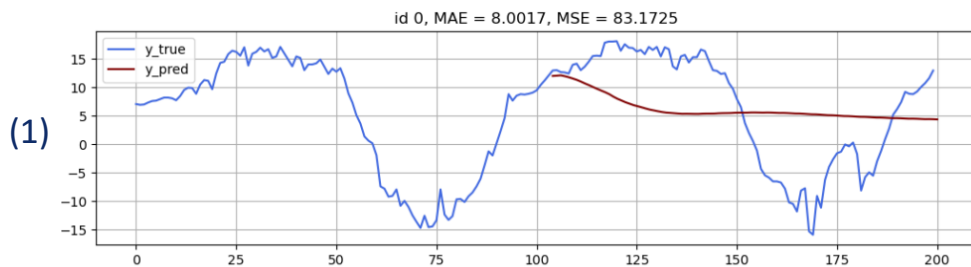
(4)





## Прошлые результаты: методы машинного обучения

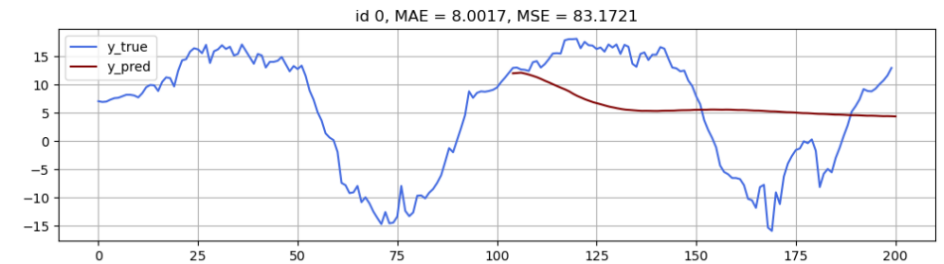
- Изначально был только CatBoost...
- Имплементированы следующие ML модели:
  - Vanilla linear regression (1)
  - Lasso regression (2)
  - Ridge regression (3)
  - Random forest (4)
  - Catboost (5)



(2)



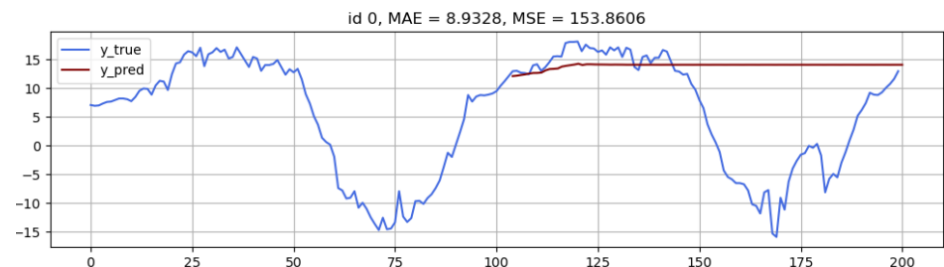
(3)



(4)



(5)





А вот как один из трансформеров помогает немного подтянуть  
предсказания CatBoost

**True and predicted values**





## Новые результаты: статистические методы

- Имплементированы следующие методы:
  - AutoETS
  - AutoARIMA
  - AutoTheta
- Все наивные и статистические модели встроены в общий пайплайн.

## Планы

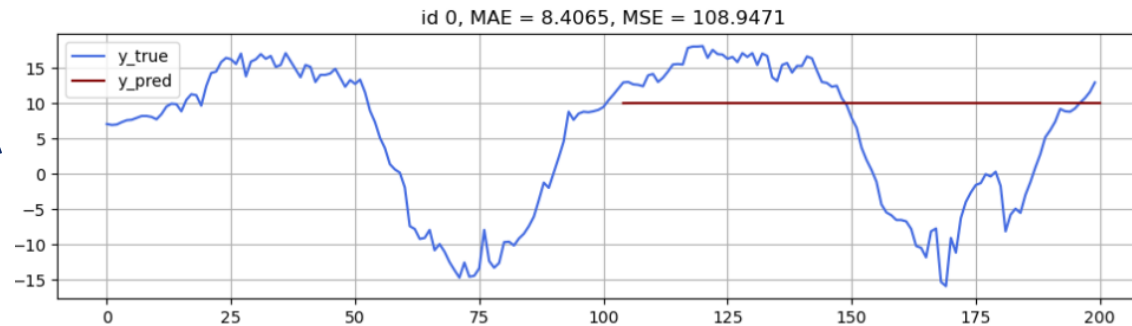
- Модели детрендинга и десезонализации
- Ансамблирование статистических и ML моделей



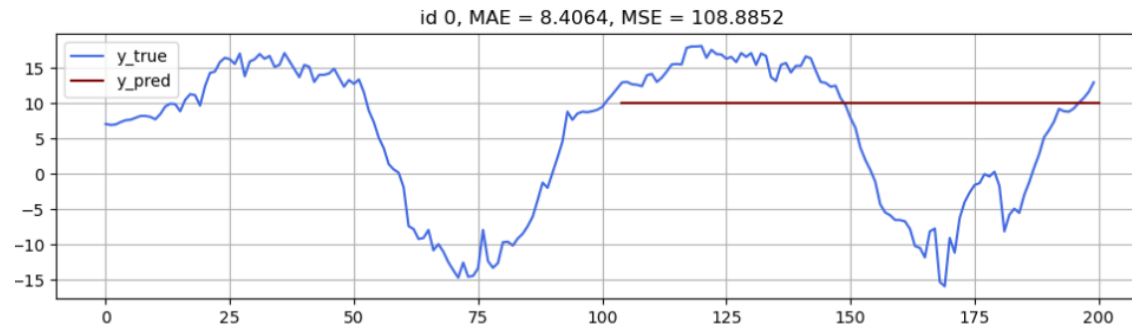
AutoTheta



AutoARIMA



AutoETS

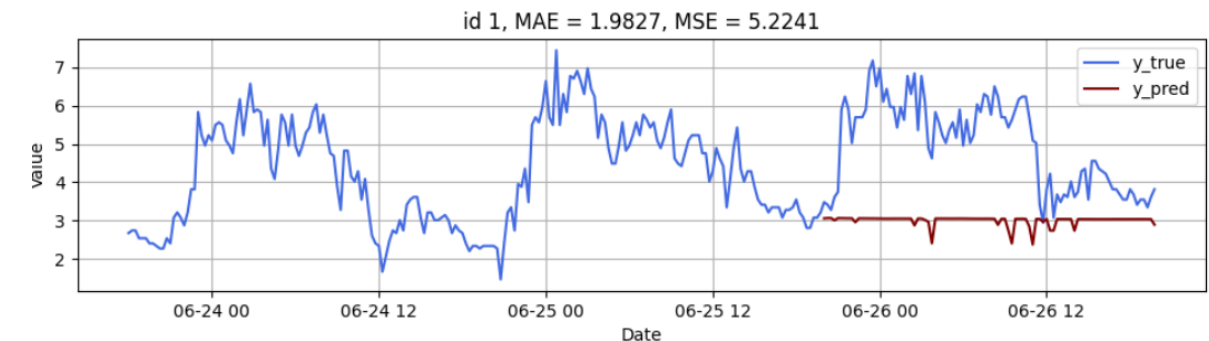
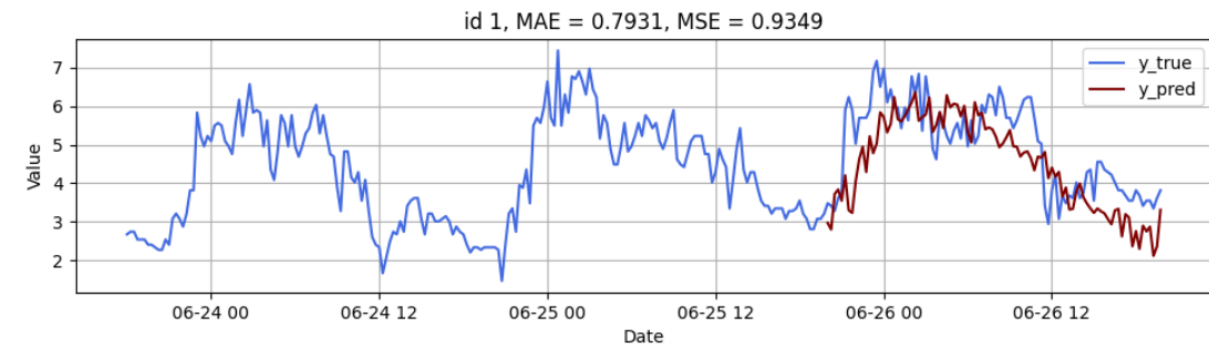
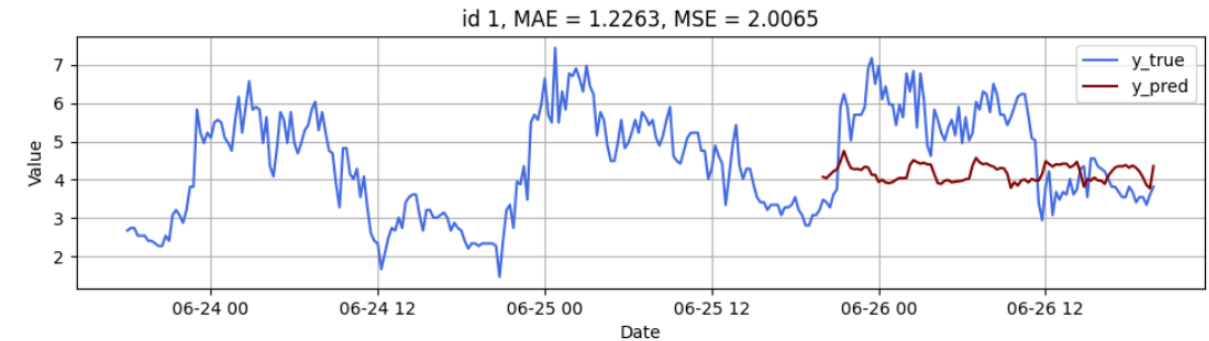




## Прошлые результаты: нейронные сети

- Проанализирована литература по использующимся архитектурам в области прогнозирования TS

*Autoformer*



*PathchTST*

*Informer*

### Основные архитектуры:

- *Convolutional neural network (CNN)*
- *Recurrent neural network (RNN)*
- *Graph neural network (GNN)*
- *Transformers (показывают SOTA-результат; было решено остановиться на них)*





## Новые результаты: нейронные сети

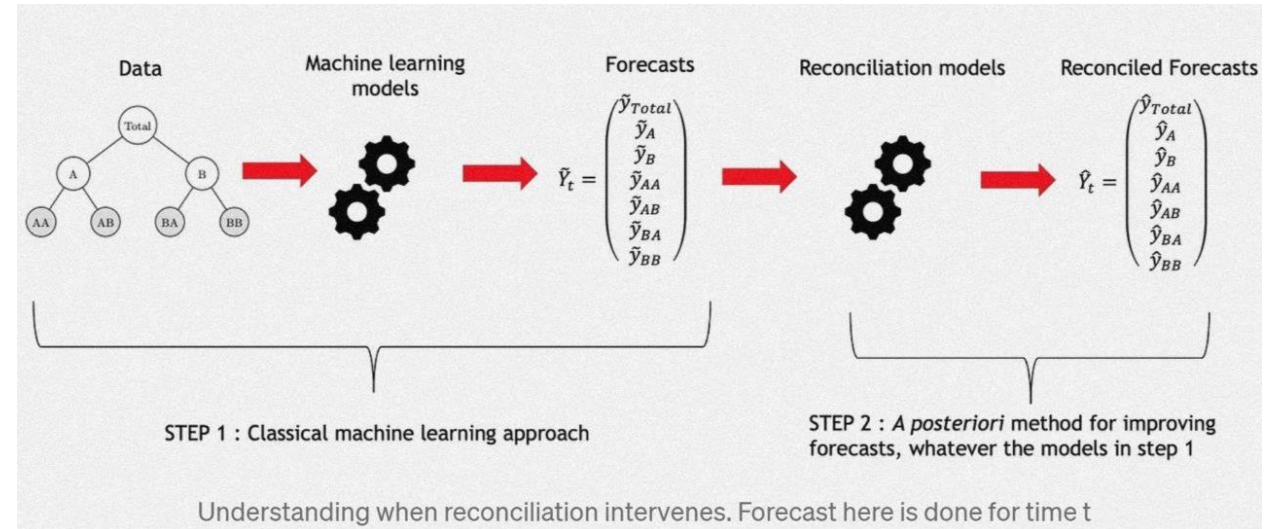
- Осуществлен переход с torch lightning на torch (для лучшего контроля настроек обучения)
- В tsururu добавлены код подготовки данных для нейронных сетей (так же для использования в различных стратегиях) и код модели в нужном формате (на примере Dlinear), чтобы можно было добавлять свои уже имеющиеся либо кастомные архитектуры.

## Планы

- Довести до конца имплементацию нейронных сетей и исследовать поведение моделей в разрезе различных стратегий.

## Результаты

- Был создан класс трансформеров по преобразованию временной гранулярности на уровни «вверх» и «вниз»
  - «Вверх» = resample (например, день → месяц)
  - «Вниз» = интерполяция (например, месяц → день)



### Выводы:

1. Будем пробовать применять более сложные и точные методы преобразования гранулярности
2. Не будем дробить временную иерархию “вниз” (все более мелкий шаг), нужно построить прогноз на более агрегированном уровне (скажем, прогноз нужно сделать по часам, но мы еще делаем прогноза по дням и по неделям, потому что при каждой из таких агрегаций, получается лучше всего учесть конкретные паттерны: недельную и годовую сезонность)
3. Самой главное в иерархическом прогнозировании - подобрать подходящий класс моделей для каждого из уровня иерархии

Зуйкова Ольга (hts\_reconciliation)\*



### Основные шаги:

- Построение рядов на разных уровнях гранулярности
- Предсказание иерархических рядов и смещение прогнозов

Делаем прогнозирование по summing matrix как линейной интерполяции:

2012-01

2012-02

2012-03

2012-01/2011-12-26

2012-01/2012-01-02

...

0 или 1 отмечалось на каждом уровне иерархии

Таблица с результатами обычного прогноза и реконсильированного при помощи Bottom-up метода. Пока что наблюдаются не лучшие результаты.

ARIMA	MSE		MAE	
	no_rec	BU	no_rec	BU
weekly	1685585.71	7762581.31	1260.57	1870.9
daily	41674.07	280839.13	142.67	251.09
hourly	587.62	587.62	18.29	18.29

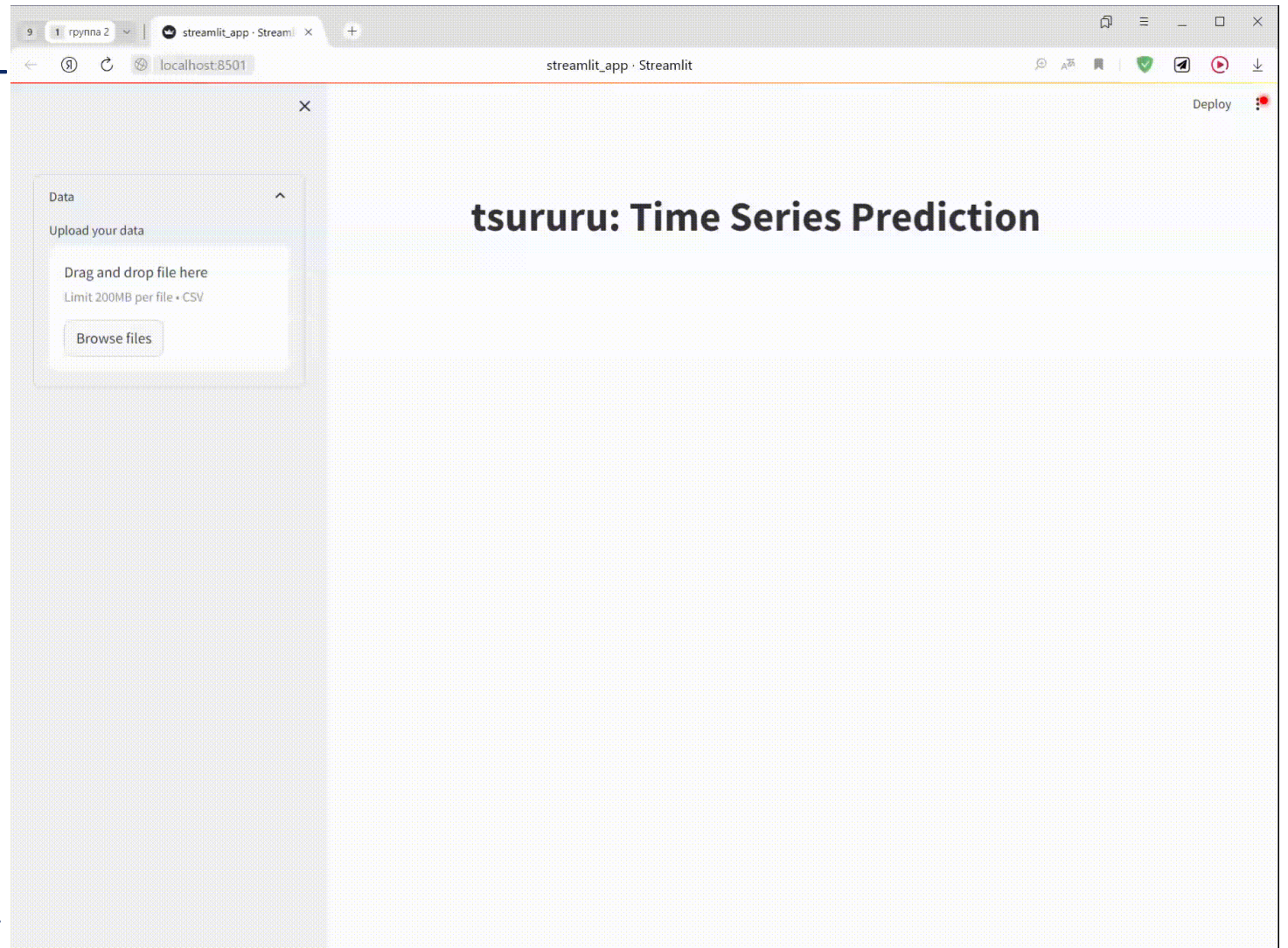
### Планы:

1. Протестировать все методы реконсильации и подобрать лучший, как и более выгодный класс моделей.
2. Учесть split-недельный прогноз, когда неделя попадает на 2 календарных месяца. Тогда недельный прогноз превращается в сплит-недельный прогноз, а уже потом месячный прогноз реконсильруется с сплит-недельным



## Прошлые результаты: Streamlit-сервис

- **Предскажем ваши временные ряды за несколько секунд (или минут... или десятков минут...).**
- Возможности будут расширяться по мере улучшений и слияний наработок в основную ветку.





## Новые результаты

- Добавлена работа с категориальными признаками.
- Перед релизом исправлены основные баги и недочеты (их было много, спасибо ребятам).
- Добавление автоматического распознавания гранулярности ряда (какой ряд: дневной, месячный, квартальный?)
- Помощь ребятам, объяснение устройства библиотеки и другие вопросы.

## Планы

- Помощь с имплементацией нейронных сетей (доработка пайплайна генерации признаков, чтобы можно было генерировать признаки по батчам, а не все сразу).
- Улучшение юзабилити (добавление новых tutorиалов и других материалов).
- Помощь в объединении наших усилий в main-ветке.
- Доработка Streamlit-приложения.
- Ну и, конечно, поиск и исправление багов 😊

