


Sprint 2

Задача: Получить рабочую модель для прогона на протезе

Изменения вводных относительно Sprint 1:

- Новые данные для двух разных пилотов
- Ещё что-то важное, чего я не заметил

team:

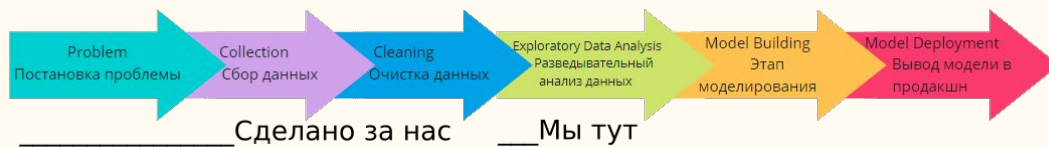
 [Alex1iv](#) - Александр

 [Yyalexx](#) - Алексей

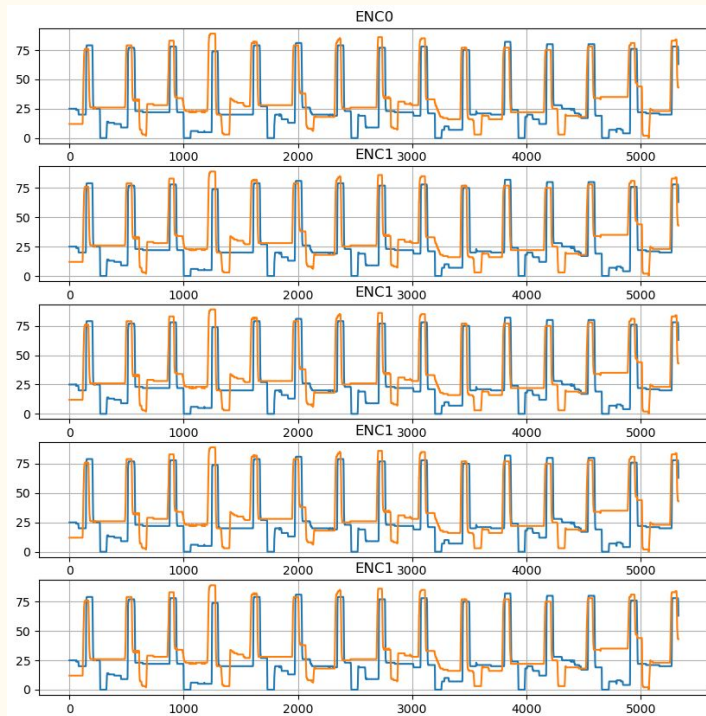
 [MapleBloom](#) - Марина

 [hoittoken](#) - Михаил

 [ostrebko](#) - Олег



Таргеты разных пилотов



Проверим какие датчики у пилотов активные:

Активных датчиков:

первый пилот: 16

второй пилот: 16

Пассивных датчиков:

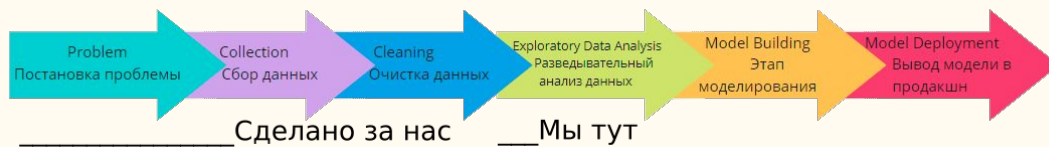
первый пилот: 34

второй пилот: 34

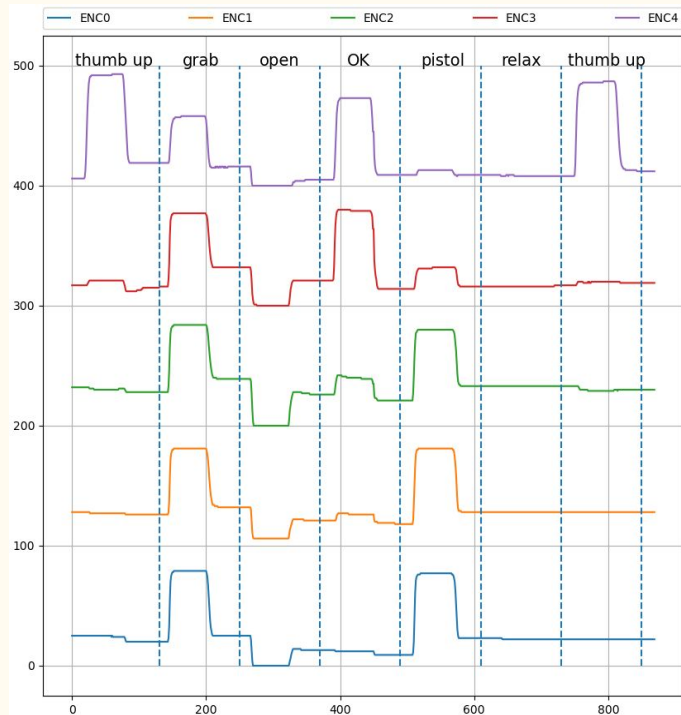
Номера этих датчиков совпадают: True

Гипотезы для проверки:

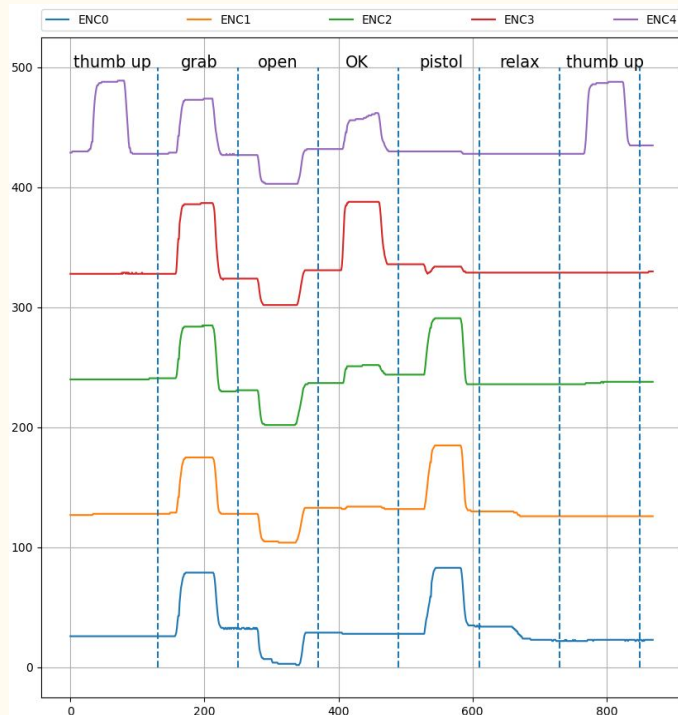
1. Датчики **взаимозаменяемы** (при разных монтажах /пилотах показания датчика А могут быть сняты с датчика В)
2. **Существует отчетливый паттерн** показаний датчиков для конкретных жестов (данный паттерн может перемещаться в пространстве датчиков)
3. На free_movements протез должен реагировать **спокойствием**
4. Конструкция с классификатором предпроцессинге для **определения NOGO** позволит существенно улучшить метрики
5. Задачу можно и нужно решить **без нейросетей**
6. **Сглаживание команд** на выходе можно организовать по сколь угодно большому накоплению предыдущей информации, с реальной задержкой по выходному сигналу **в 1 единицу времени**



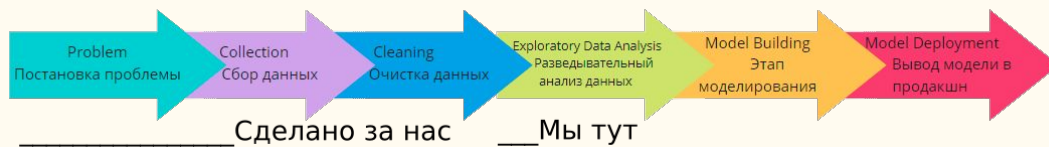
Пилот 1, диапазон [0:870]



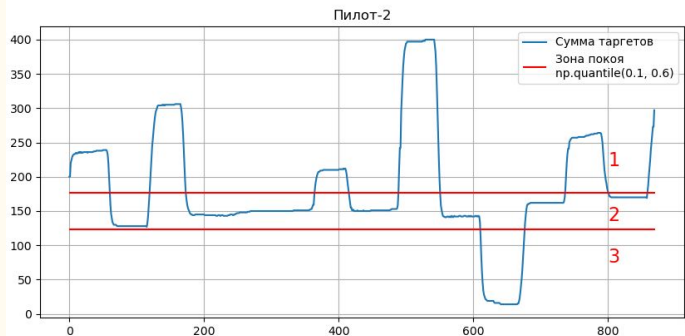
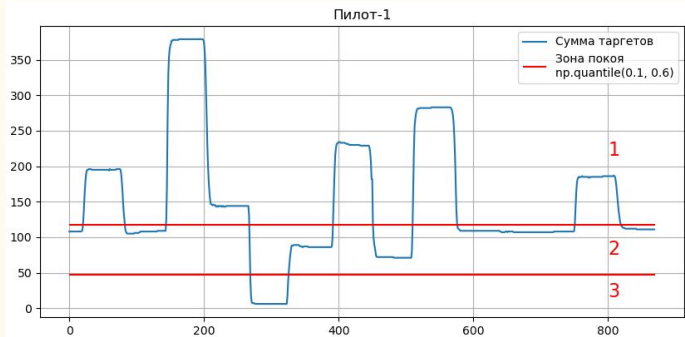
Пилот 2, диапазон [330:1200]



Повторяющийся паттерн жестов для обоих пилотов может быть полезен при препроцессинге. Особенно если в препроцессинг мы добавляем классификатор. Как-будто это может ощутимо улучшить качество предсказаний (если модель на входе уже “знает”, что от неё ждут на выходе)



Разобъём таргет на зоны



1 - Сгибание от положения покоя

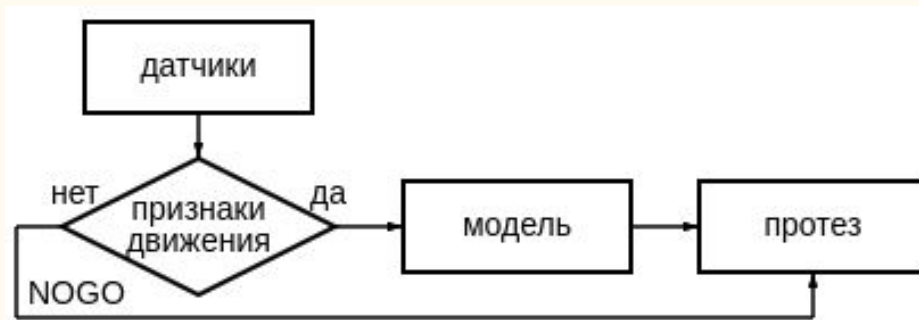
2 - Зона покоя (расслабленная рука)

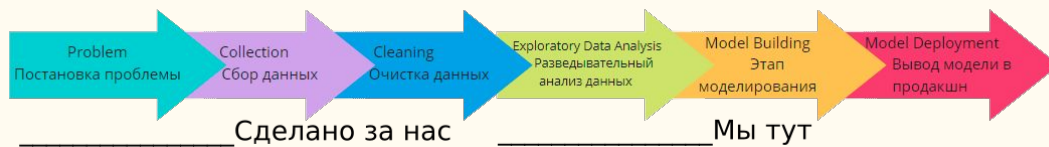
3 - Разгибание от положения покоя

Зона покоя привязана к **10-му** и **60-му** квантилю

Всё ещё хочу попробовать прикрутить классификатор на предпроцессинг.

Сюда можно было бы прикрутить классификатор, хотя бы не явный классификатор (на конструкте if... else...)





Базовые модели на объединённых данных (без пред и пост обработки)

INDEX	lr_pilot_1	lr_pilot_2	vr_pilot_1	vr_pilot_2	en_pilot_1	en_pilot_2	xgb_pilot_1	xgb_pilot_2	lgb_pilot_1	lgb_pilot_2
ENC0	144.452417	149.452798	112.900060	118.686712	143.110811	154.396532	92.579431	132.193758	86.738974	152.108946
ENC1	101.116083	117.938012	88.186855	84.640438	100.382778	122.247244	57.232394	77.110563	60.442739	102.552523
ENC2	117.696557	108.818361	93.291931	70.456218	116.366246	111.354833	72.078710	48.156578	78.668825	59.070663
ENC3	107.812676	180.603738	86.885266	101.398573	105.482038	182.943771	65.373375	66.720286	72.426480	98.867079
ENC4	413.214784	105.420510	346.636066	80.025143	415.856617	102.187213	293.561001	62.553216	284.330516	122.703637
ENC5	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000001	0.000001	0.000000	0.000000
MSE mean	147.382086	110.372236	121.316696	75.867847	146.866415	112.188265	96.804152	64.455734	97.101256	89.217141

Подсвечены наиболее точные предсказания:

Зелёным для первого пилота

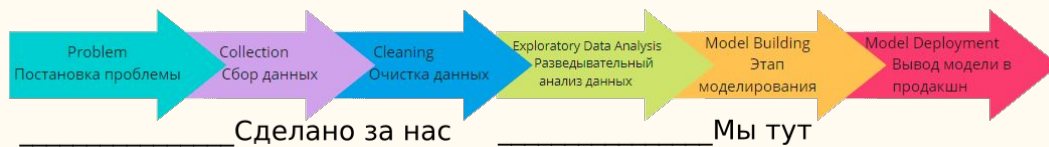
Голубым для второго пилота

xgboost.XGBRegressor (оптимизирован optun'ой)

lightgbm.LGBMRegressor (оптимизирован optun'ой)

Тройка фаворитов:

1. MultiOutputRegressor(VotingRegressor(DecisionTreeRegressor())LassoLars())
2. xgboost.XGBRegressor()
3. MultiOutputRegressor(lightgbm.LGBMRegressor())



Время инференса на разных базовых моделях (данные по тысяче прогонов)

```
runs = []
```

```
for i in np.arange(1000):
    start_time = time.time()
```

```
    preprocessing(omg_sample)
    inference(preprocessing(omg_sample))
    postprocessing(model_vr.predict([preprocessing(omg_sample)]))
```

```
    commands(postprocessing(model_vr.predict([preprocessing(omg_sample)])),
              postprocessing(model_vr.predict([preprocessing(omg_sample)])))
```

```
    run_time = pd.Series((time.time() - start_time))
    runs.append(run_time)
```

За время инференса принят промежуток:

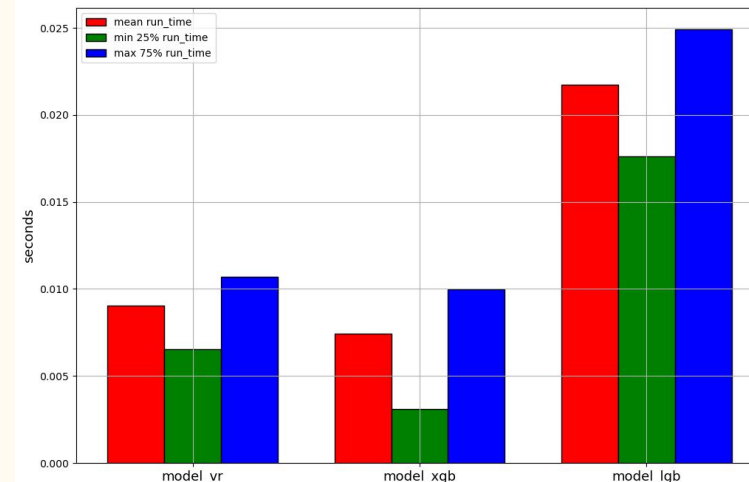
от поступления показаний датчиков

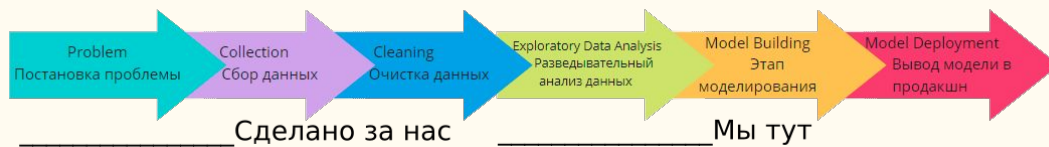
до выдачи команды протезу

Для усреднения результатов время измерялось:

1000 раз

	model_vr	model_xgb	model_lgb
25%	0.006533	0.003104	0.017601
mean	0.009060	0.007444	0.021726
75%	0.010713	0.009987	0.024906

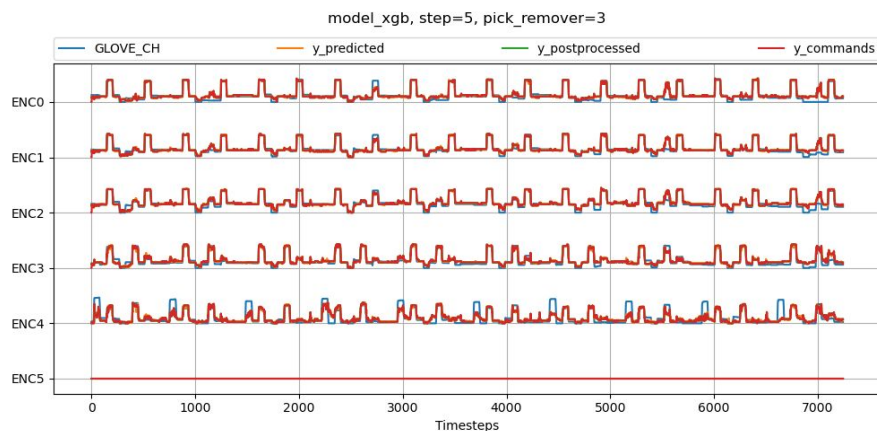




Модель обучена на протокольных жестах обоих пилотов

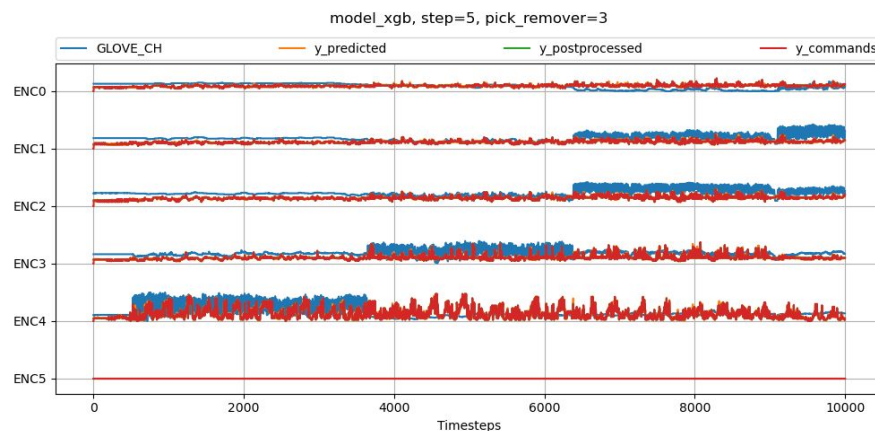
Предсказания для:

первого пилота, жесты по протоколу

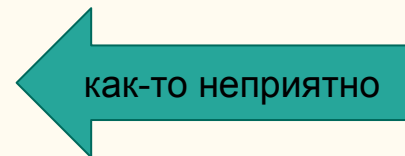


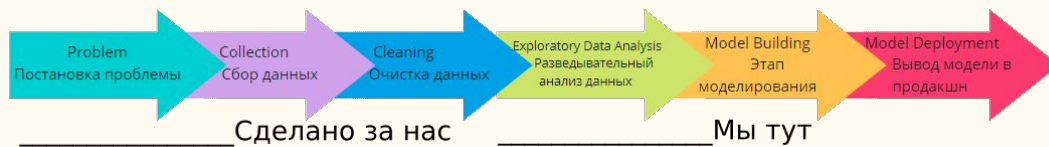
ENC0 98.163446
ENC1 61.849117
ENC2 77.117477
ENC3 71.455411
ENC4 298.688294

первого пилота, свободные жесты



ENC0 137.832500
ENC1 320.229200
ENC2 470.573900
ENC3 391.351500
ENC4 659.926400

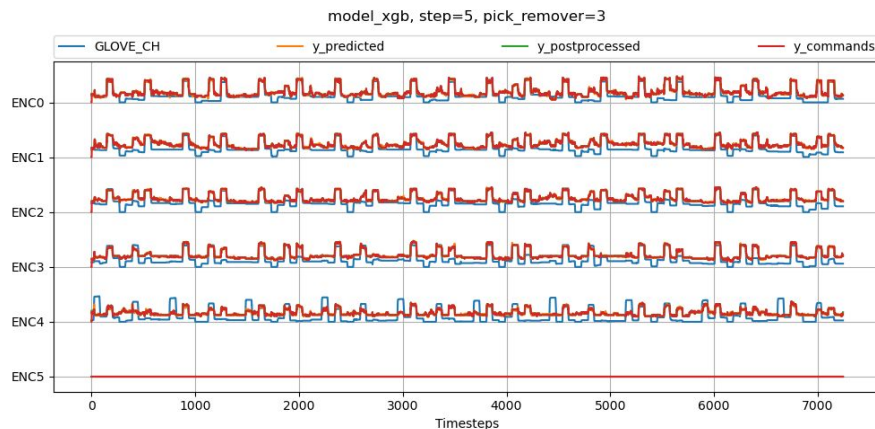




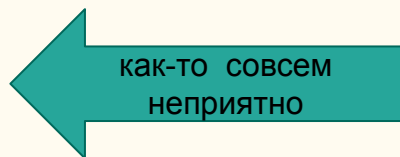
Модель обучена на свободных жестах обоих пилотов

Предсказания для:

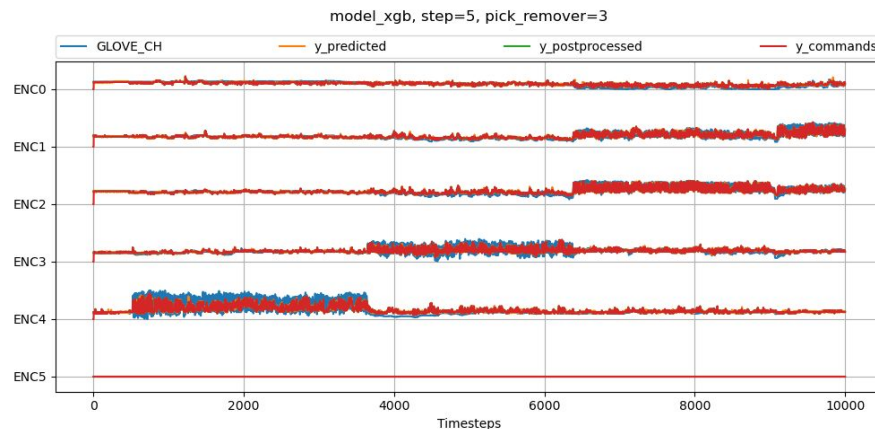
первого пилота, жесты по протоколу



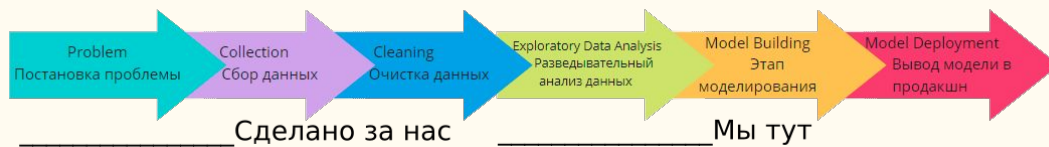
ENC0 435.057427
ENC1 464.522363
ENC2 392.008421
ENC3 447.472805
ENC4 687.425870



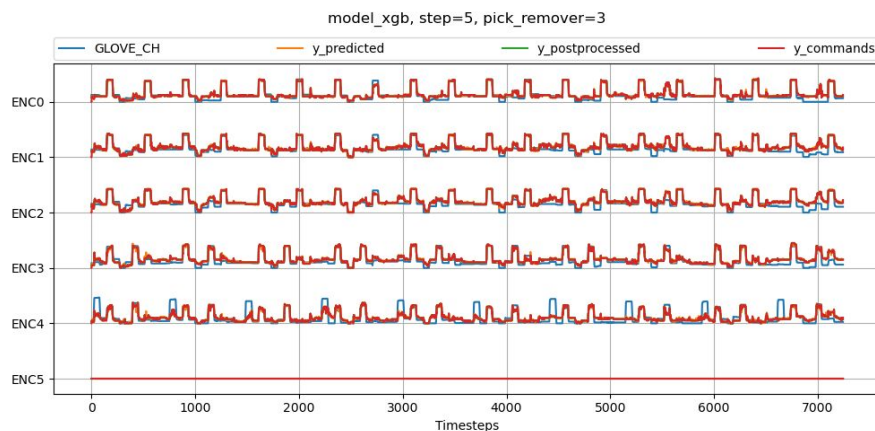
первого пилота, свободные жесты



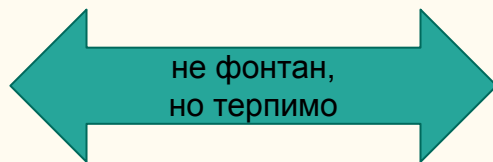
ENC0 46.283500
ENC1 37.229200
ENC2 39.490400
ENC3 49.364500
ENC4 180.440400



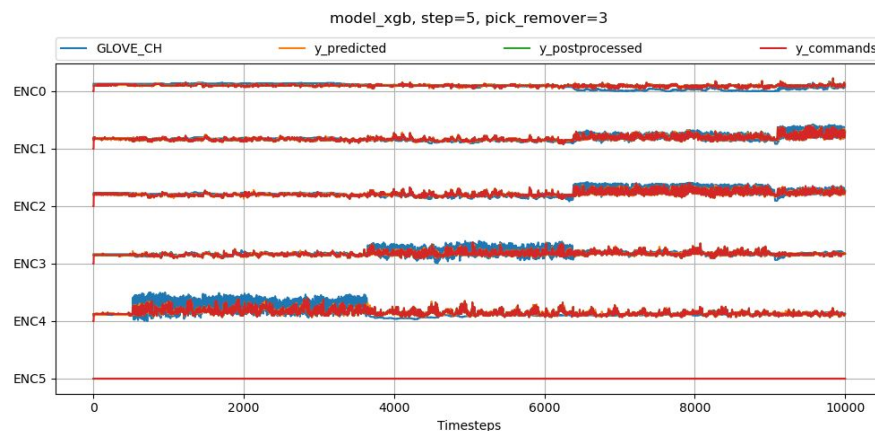
Модель обучена на всех доступных данных
 Предсказания для:
 первого пилота, жесты по протоколу



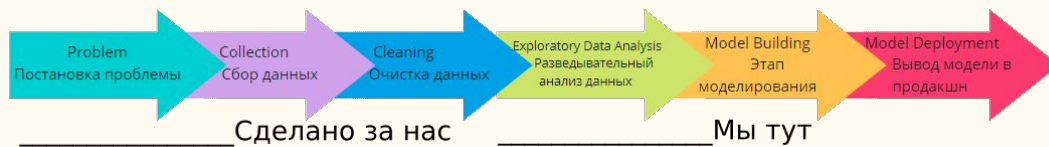
ENC0 112.614716
 ENC1 129.751518
 ENC2 135.725980
 ENC3 127.193263
 ENC4 359.715351



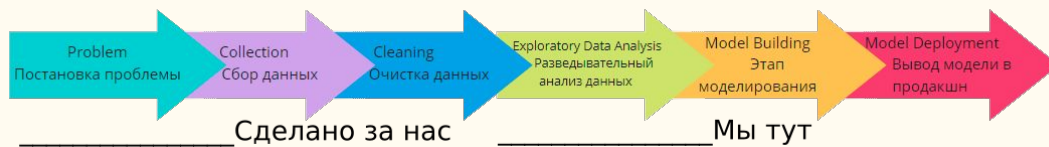
первого пилота, свободные жесты



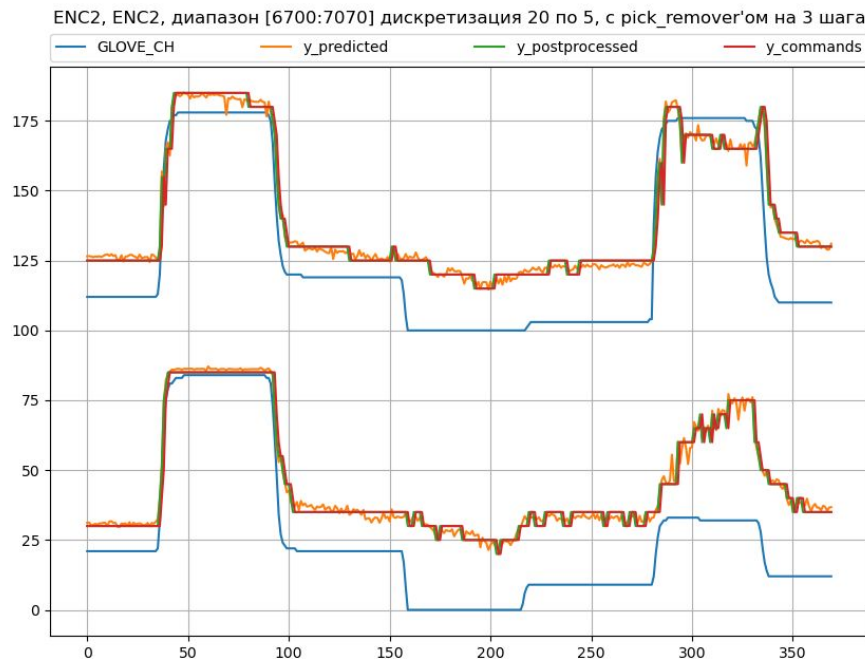
ENC0 82.6600
 ENC1 57.6857
 ENC2 82.0324
 ENC3 88.2130
 ENC4 298.0554



1. Данные для моделирования - последовательно соединённые данные 2-х пилотов (**train** протокольных жестов и **free_movements**), дополнительно нарезанные **sklearn.model_selection.TimeSeriesSplit**
2. Подготовлены к работе **3 модели** (**VotingRegressor**, **XGBRegressor**, **LGBMRegressor**)
3. Предпроцессинг - **добавляем 1 признак** (которая является флагом действия/бездействия)
4. Постпроцессинг:
 - **дискретизация таргетов** на заданное количество диапазонов (по умолчанию 20 диапазон, по 5 единиц в каждом)
 - **сглаживатель пиков** по предыдущим показаниям с шагом 3, даёт задержку между предсказанием и выводом на протез в 1 временной шаг



Укрупнённый фрагмент обработки модели для тестовых данных первого пилота ENC2, ENC3



в диапазоне [6700:7070]

дискретизация 20 по 5

сглаживание по 3 шагам

(что даёт реальную задержку
между снятием сигнала и
отправкой команды 30мс)

Вопросы:

1. Как подаётся NOGO (при отсутствии изменений сигналов датчиков выше заданного порога?)
2. Насколько гладкий сигнал должен быть на выходе (чтобы протез не испытывал “биение”?)
3. Как протез должен реагировать на пики (резкие изменения показаний датчиков?)