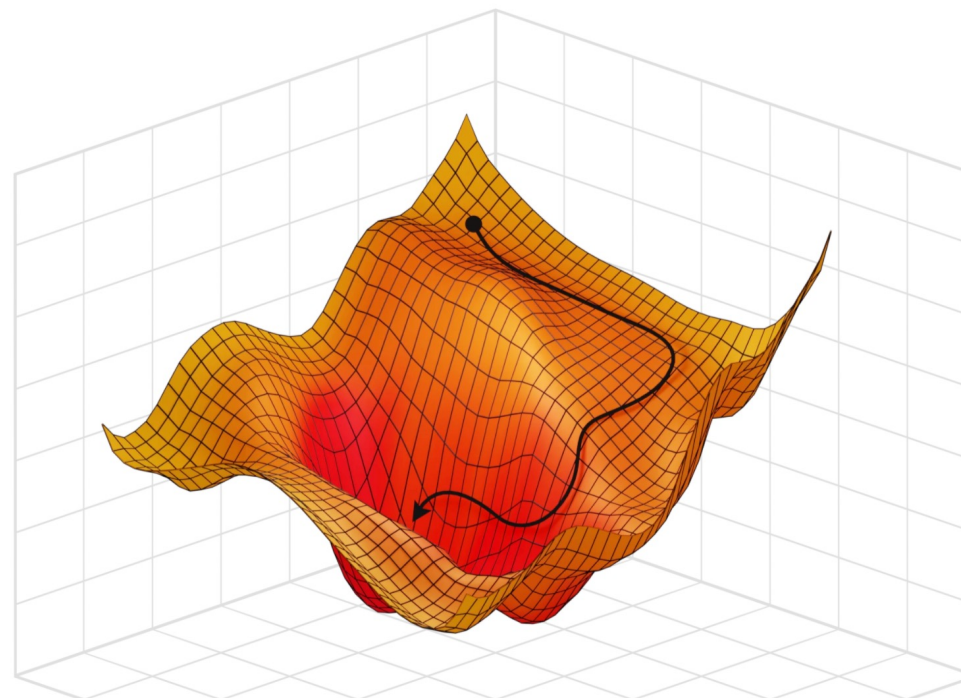


Команда: Dsteam

Буслов Дмитрий

Фиронов Сергей

Задача 1: Продуйте металл через
Data Science



**EVRAZ AI
CHALLENGE**



Постановка задачи

Задача - разработать алгоритм предсказания содержания углерода и температуру чугуна.



Дополнительная задача - придумать как обобщить алгоритм прогноза на ситуацию реального производства. В реальном времени мы не знаем, когда наступит конец проливки, а должны определить это сами на основании прогнозируемых параметров чугуна.

- Что нужно добавить?
- Как перейти от модельной задачи к задаче из реального мира?
- Как развить алгоритм так, чтобы это было максимально полезно бизнесу?

Справочная минутка по чтению данных для DS

Пропустить, если
мало времени

Когда лень писать код:

```
files_train = glob.glob(DATA_DIR+'*train.csv')
files_test = glob.glob(DATA_DIR+'*test.csv')
```

```
def gen_read(files, train_type):
    for i in files:
        print(f'df_{i[8:].split("_")[0]}_{train_type}=fread(\'{i}\').to_pandas()')
gen_read(files_train, 'train')
print('-'*50)
gen_read(files_test, 'test')
```

```
df_sip_train=fread('../data/sip_train.csv').to_pandas()
df_lom_train=fread('../data/lom_train.csv').to_pandas()
df_chronom_train=fread('../data/chronom_train.csv').to_pandas()
df_produv_train=fread('../data/produv_train.csv').to_pandas()
df_target_train=fread('../data/target_train.csv').to_pandas()
df_chugun_train=fread('../data/chugun_train.csv').to_pandas()
df_gas_train=fread('../data/gas_train.csv').to_pandas()
df_plavki_train=fread('../data/plavki_train.csv').to_pandas()
-----
df_gas_test=fread('../data/gas_test.csv').to_pandas()
df_produv_test=fread('../data/produv_test.csv').to_pandas()
df_lom_test=fread('../data/lom_test.csv').to_pandas()
df_chugun_test=fread('../data/chugun_test.csv').to_pandas()
df_chronom_test=fread('../data/chronom_test.csv').to_pandas()
df_plavki_test=fread('../data/plavki_test.csv').to_pandas()
df_sip_test=fread('../data/sip_test.csv').to_pandas()
```

Когда pandas заставляет ждать...

```
%%time
df_sip=pd.read_csv('../data/sip_train.csv')
df_lom=pd.read_csv('../data/lom_train.csv')
df_chronom=pd.read_csv('../data/chronom_train.csv')
df_produv=pd.read_csv('../data/produv_train.csv')
df_target=pd.read_csv('../data/target_train.csv')
df_chugun=pd.read_csv('../data/chugun_train.csv')
df_gas=pd.read_csv('../data/gas_train.csv')
df_plavki=pd.read_csv('../data/plavki_train.csv')
```

CPU times: user 15.1 s, sys: 1.4 s, total: 16.5 s
Wall time: 16.5 s

```
%%time
df_sip=fread('../data/sip_train.csv').to_pandas()
df_lom=fread('../data/lom_train.csv').to_pandas()
df_chronom=fread('../data/chronom_train.csv').to_pandas()
df_produv=fread('../data/produv_train.csv').to_pandas()
df_target=fread('../data/target_train.csv').to_pandas()
df_chugun=fread('../data/chugun_train.csv').to_pandas()
df_gas=fread('../data/gas_train.csv').to_pandas()
df_plavki=fread('../data/plavki_train.csv').to_pandas()
```

CPU times: user 7.08 s, sys: 834 ms, total: 7.91 s
Wall time: 2.41 s

Нюансы в данных

Adversarial валидация - дает 0.96 ROC_AUC

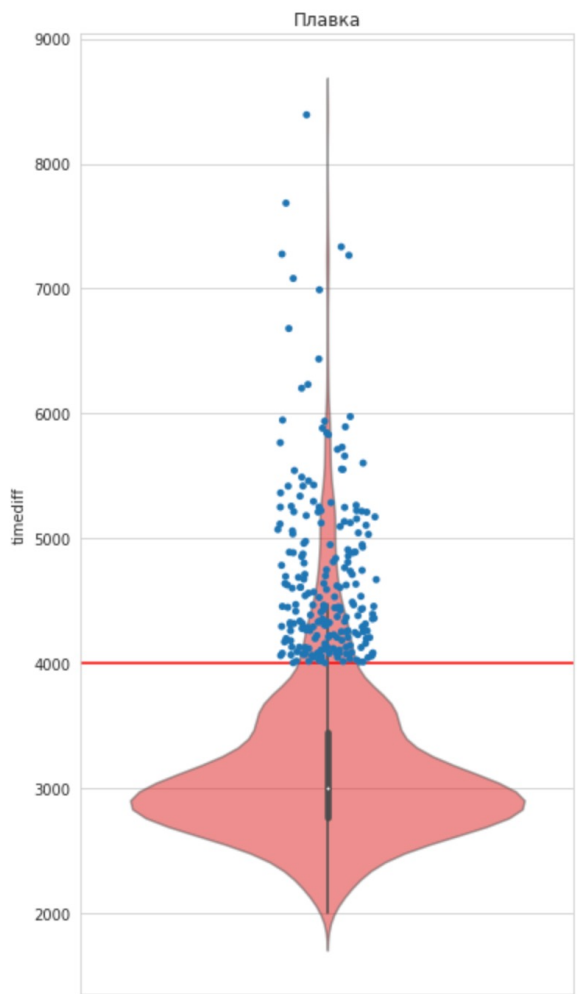
Плавка 511135 - 75 разных стойкостей фурмы и головки фурмы

Отличия в сыпучих

Test dataset

Train dataset

	index	NMSYP		index	NMSYP
0	изв_ЦОИ	0.603312	0	изв_ЦОИ	0.737717
1	Уголь ТО	0.245282	1	Уголь ТО	0.100268
2	ФЛЮМАГ	0.092768	2	ФЛЮМАГ	0.080482
3	Флюс ФОМИ	0.051418	3	Флюс ФОМИ	0.032594
4	изв отсев	0.007156	4	изв_отсев	0.019499
5	Долом(сыр)	0.000063	5	агл_офл_с	0.014720
			6	долом_сыр	0.014051
			7	кокс_25_40	0.000669



**Здесь должен был быть
этот мем, но у нас
“серьезная”
презентация...**

***Поэтому, он у нас
находится в
суперпозиции - он
вроде как есть, но его
как бы и нет...***

**Да че там на эти данные смотреть,
давай сетки учить!**



Модель



Ключевые моменты:

На вход модели подаются данные до конца плавки (не заглядываем в будущее и обрезаем)

Низкочастотные категории отбрасываем, заменяя их на 'none' (для эмбединга)

Аналогичный подход для категорий теста

Мы индексируем события по времени с начала плавки и модель может предиктить значения в любой момент времени, они меняются плавно

Преимущества подхода

Возможность учитывать все состояние внутри, в дальнейшем возможность собирать эмбединг для плавки и поиска ближайших плавок / субстояний (Metric - learning)

Возможность учиться батчами и предиктить (необходимая параллелизация для оптимизации)

Учесть оценку влияния отдельных событий по сыпучим и лому в заданные моменты времени

Легко перестроить на предикт наилучших действий

Визуализация отдельных эмбедингов в Projector-e

Экстраполяция (деревянные модели не умеют очень слабы) - можно работать со сдвигом признаков

Почти не нужны гиперпараметры и улучшения скоря возможны просто при добавлении данных

Интерфейс

Что важно бизнесу?

- ~~Когда какая будет температура~~
- ~~Когда какой будет углерод~~
- **Какие управляющие параметры менять и когда**
- **Как изменение параметров повлияет на результат**
- **Какая экономия**



Предложения

Быстрый результат - это только прогноз для информации оператора и поиск аналогичных плавок.

Оценка влияния на результат статичного изменения отдельных управляющих параметров

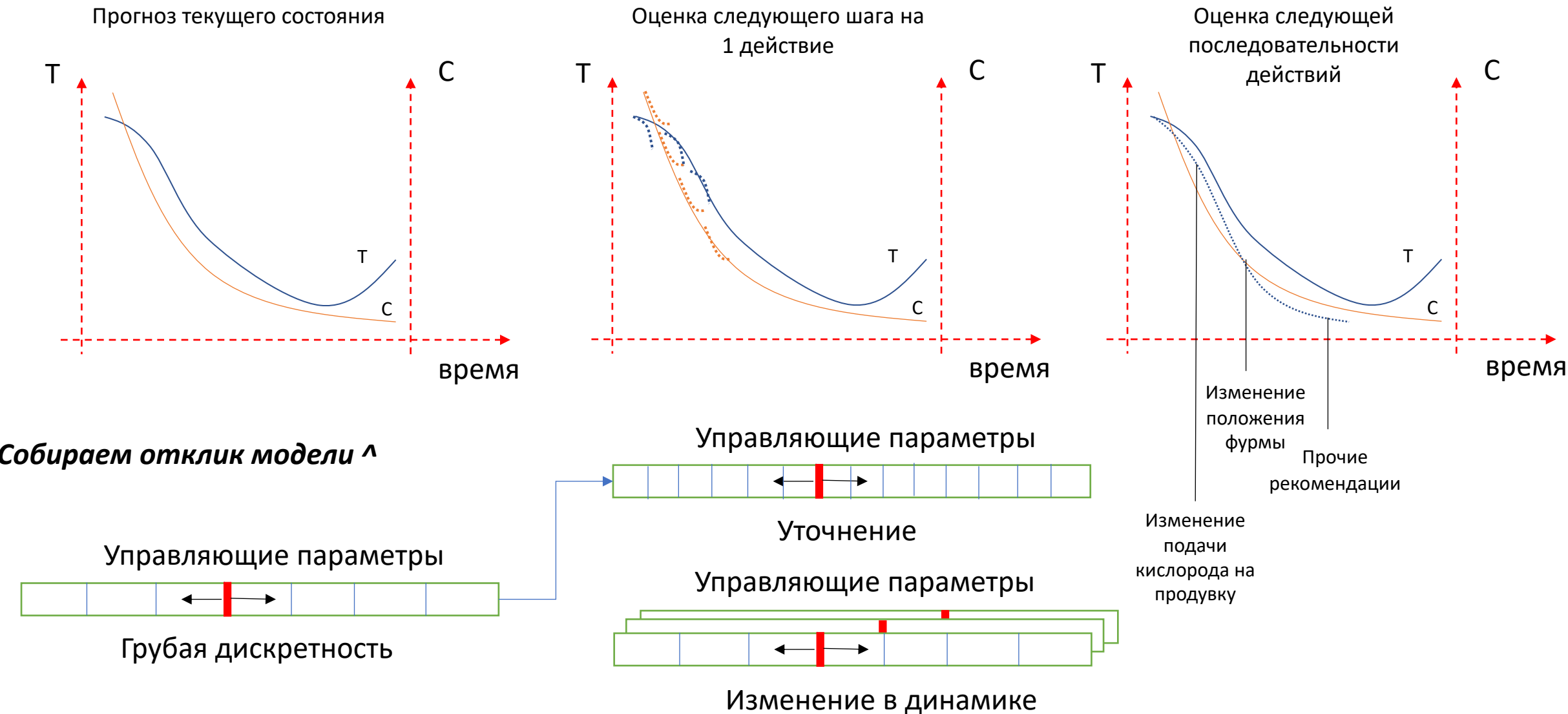
Оценка влияния на результат совместного изменения управляющих параметров

Использования моделей для формирования среды симуляции (gym)

Использование Decision Transformer (RL) <https://arxiv.org/abs/2106.01345>

А еще может можно собрать больше данных с других заводов?

Оптимизация



Доменные знания -> модель

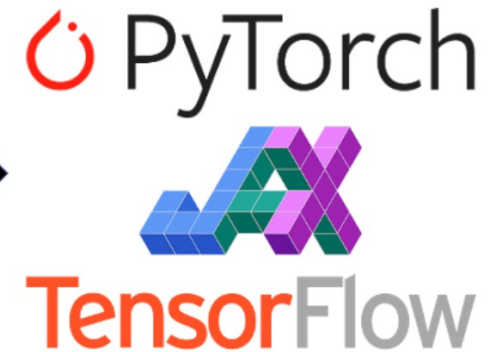
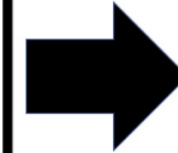
Ограничения на отдельные признаки и их интеракции имеет смысл закладывать сразу в модель.

Для нейронок - дифференцируемые слои `cvxpylayers`

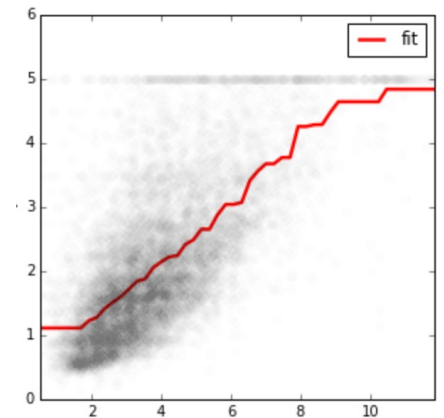
Для бустинга - монотонные ограничения

Мы не успели, но очень хотели...

$$\begin{array}{l} \text{CVXPY} \\ x^*(\theta) = \underset{x}{\operatorname{argmin}} f(x; \theta) \\ \text{subject to } g(x; \theta) \leq 0 \\ h(x; \theta) = 0 \end{array}$$



dmlc
XGBoost
Monotonic Constraints



Металл

В любом исходе - мы сможем говорить, что мы **продули!**

Резюме «на 8 баллов» ?

Target - Собрать прогнозную модель и сделать концепт UI, показав что его можно использовать с моделью

ROI - Компания сократит доплаты (в будущем) благодаря поиску оптимальных порогов остановки. Также, модель позволит обучить операторов и оценить вклады отдельных признаков (ICE, PDP)

Exploration - Прошлая плавка и состояние головок не менее важно, чем действия во время продувки.
только по газовому состоянию можно получить скор до 0.56 на этих данных.

Alternatives - Ничего не делать и собирать данные☺ Собрать модель оценки отдельных действий по изменениям параметров (ручная разметка)

Solution - (см. архитектуру сети) Наибольшие перспективы для учета всех параметров.

UI Demo - Есть, но в реальности необходимо прорабатывать с технологами

Roadmap - Работа над моделью, работа над данными. MVP для технологов (разработка)

Estimate - 0.65+ потенциал улучшения для текущих данных. 0.78 при X2 данных по плавкам