MTS ML CUP 2023

Решение 3-го места, Ишков Денис

13.04.23

Соревнование на ODS MTS ML CUP

- Задача – предсказание пола и возраста по интернет-следу пользователя
- Данные – история переходов с сайтов*

Train / Test **- 270 000** / **144 725**

- Метрика – свёртка **GINI** и взвешенной **F1-меры**:

score = 2*roc_auc(is_male) - 1 +
2*f1_weighted(age_bin)

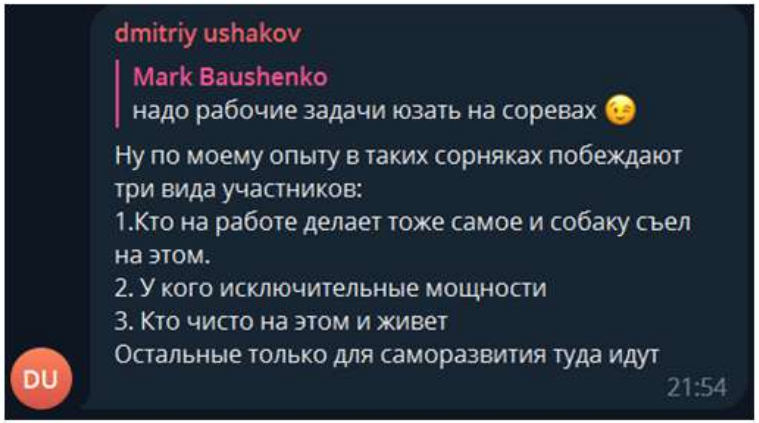
public_lb – **55%**, private_lb – **45%**

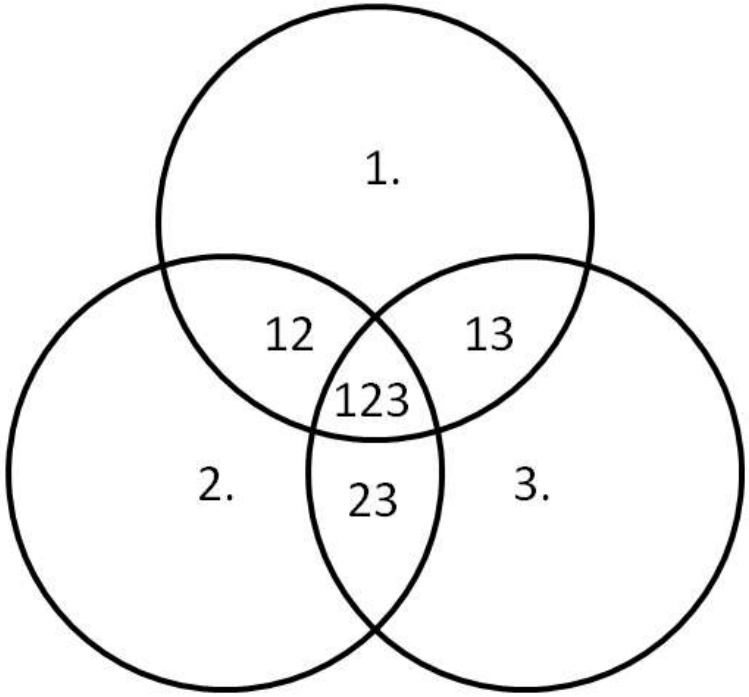
1	Vladimir Bazhenov	1,7825
2	flow	1,7802
3	Ishkov Denis	1,7788
4	сансеймайсамба	1,7744
5	Stanislav Chistyakov	1,7739
6	Anton Karasev	1,7710
7	John Galt	1,7674
8	tg kaggle_fucker	1,7638
9	More Trees Stack	1,7634
10	Ivan Alexandrov	1,7617
11	DS48_DMVL	1,7616
12	room722	1,7591
13	AK	1,7585

Мой бэкграунд

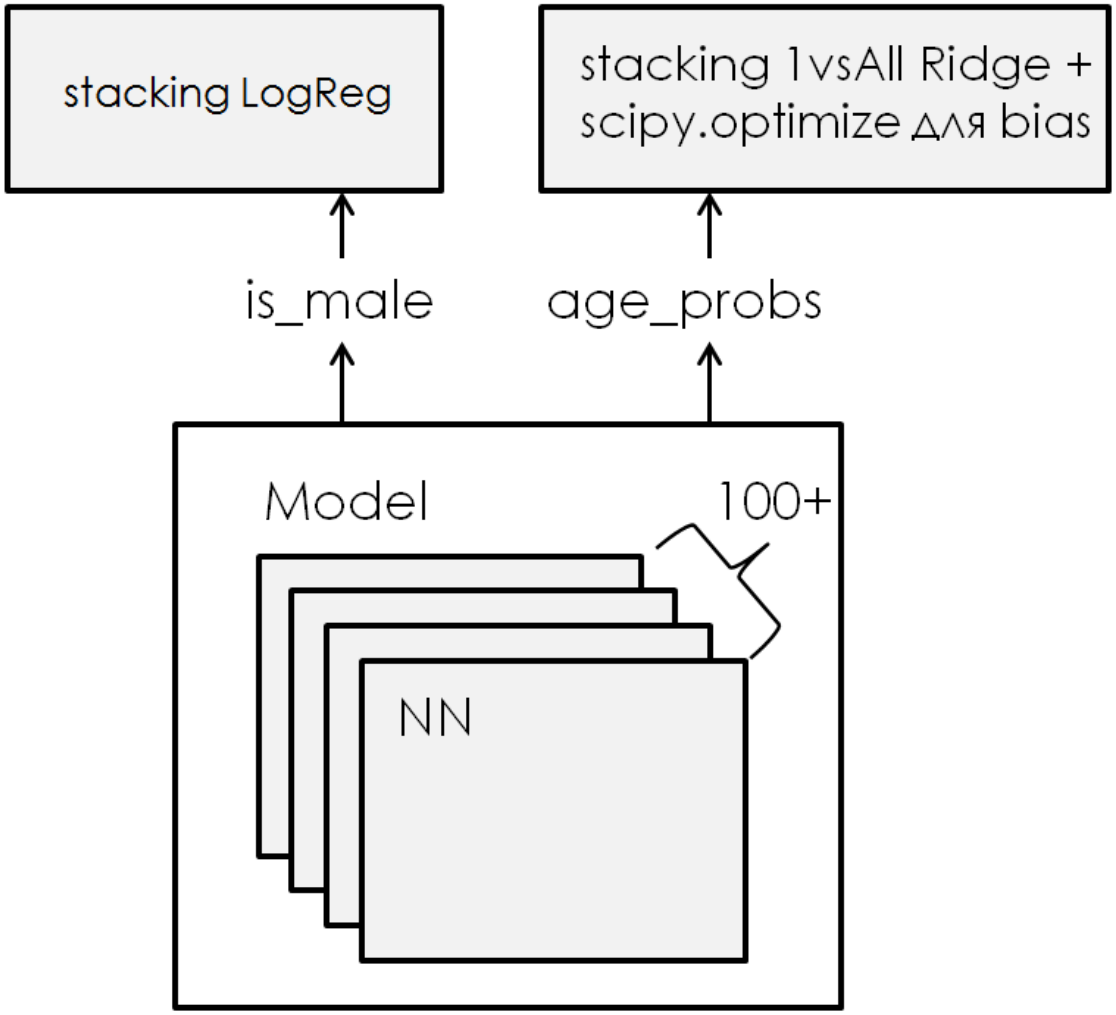
Coursera+Youtube+ПетПроекты
+Научные конференции

- До 2020г. – видео-лекции на Youtube
- 05.2020 – Coursera "How to Win a Data Science Competition" + ещё 4 курса от авторов
- 09.2020-06.2022 – магистратура с ML и DL
- 09.2021 – Пет-проект и статьи в Scopus по дистилляции знаний для задачи OCR
- 03.2022-… – VK. Аналитик в B2B
- 09.2022-… – BMSTU. Курсы по DL/CV
- **03.2023 – MTS ML CUP, 3 место**





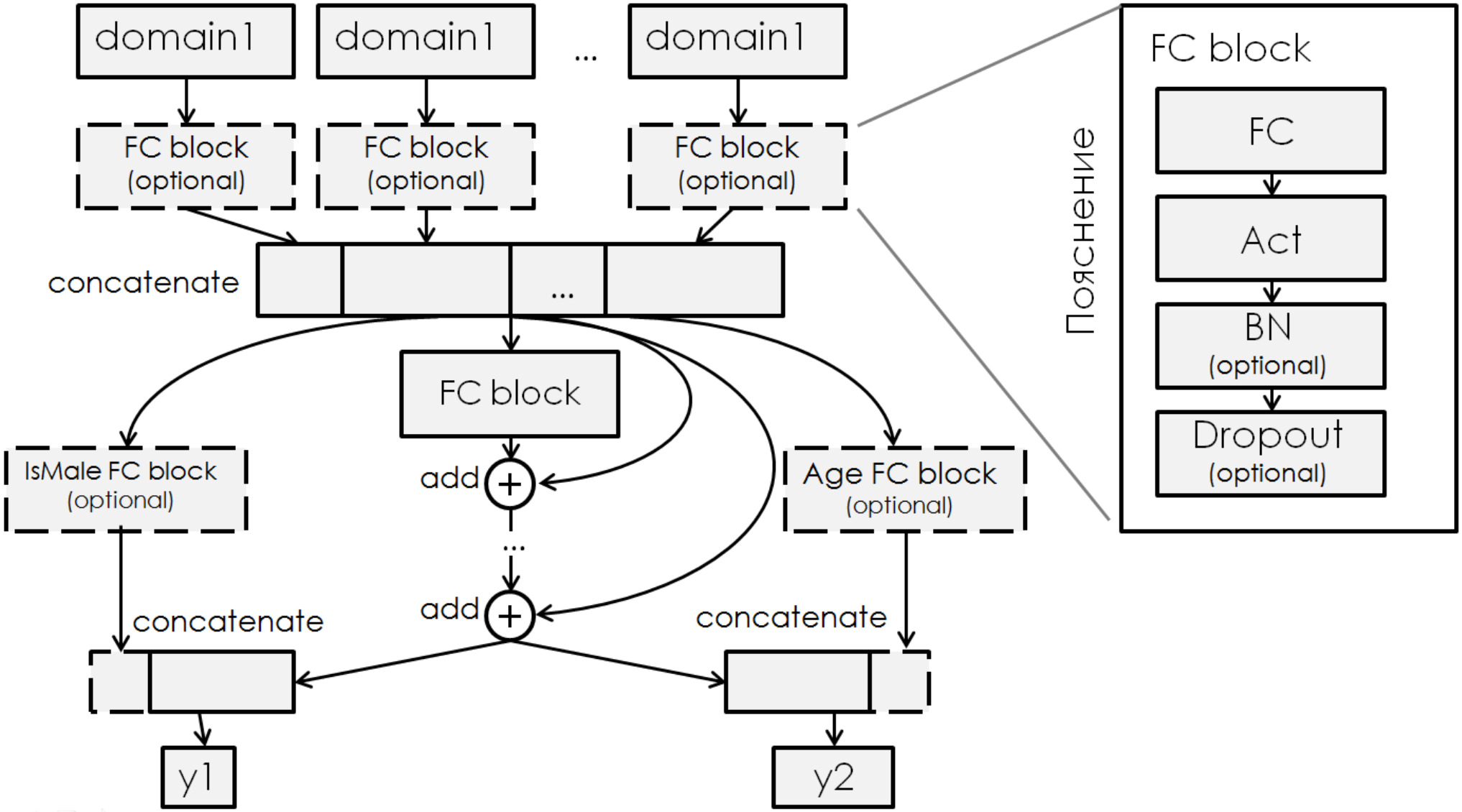
Архитектура решения. Стэкинг



Лучший скор 1 модели – 1,7638
Стэкинга – 1,7750

Архитектура решения. Основная модель

Полносвязная multi-domain multi-output нейронная сеть с dense-connections

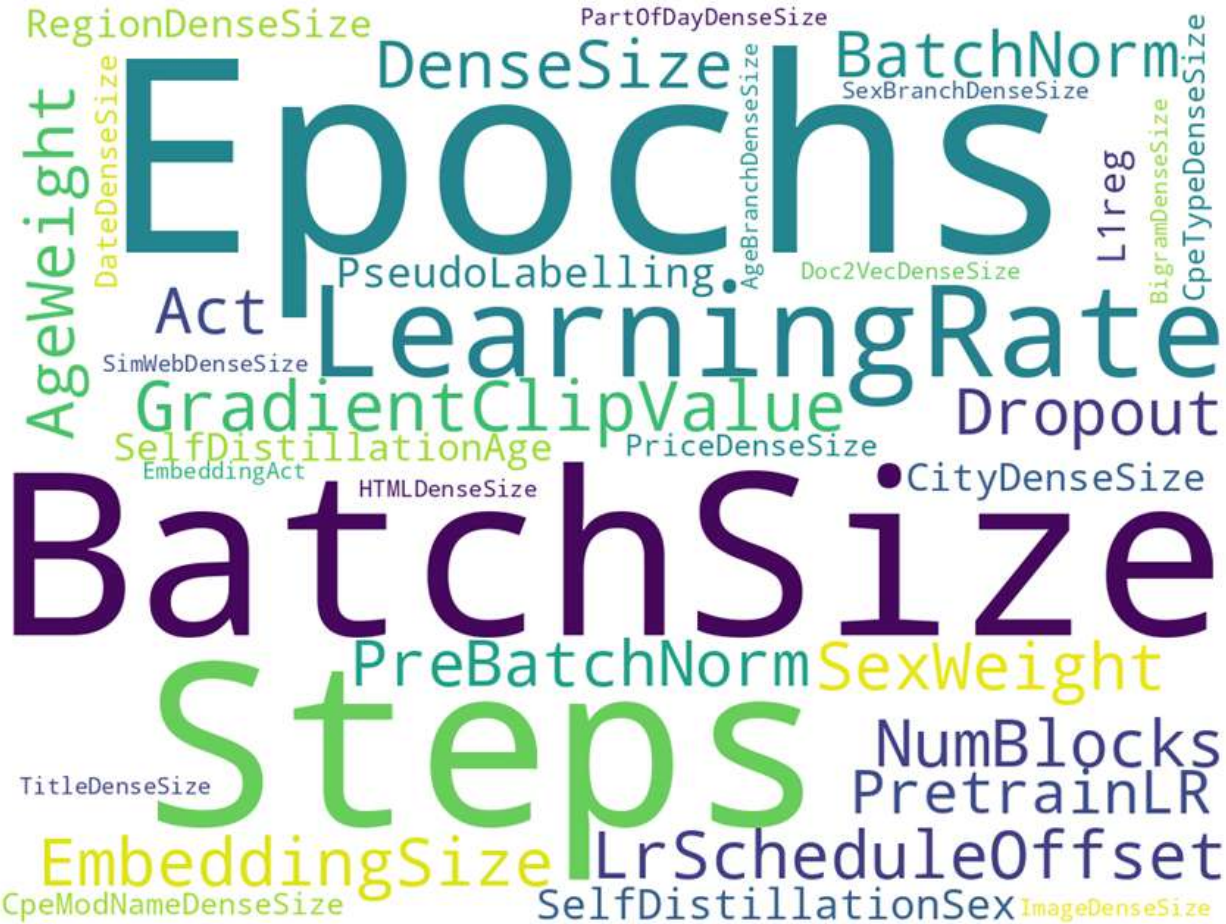


Архитектура решения. Пример модели

Порядка 70 млн. параметров

Подбор гиперпараметров

- Гиперпараметры нейронной сети подбираеются с помощью hyperopt
- Скор конфигурации оценивается на 2-3 разбиениях CV с повторением 2 раза (т.е на 1 пробу нужно 4/6 fit'ов)
- Если очевидно плохая конфигурация NN, то ранний выход из функции оценки
- Максимизация либо по полу, либо по возрасту
- На каждый поднабор фолдов в среднем закладывается где-то 300-400 проб



Множество подбираемых гиперпараметров

Домены признаков

Обозначение	Признаки	Предобработка	Размерность
domain1	url_host+request_cnt	BoW	20144
domain2	region_name	ArgMax	1
domain3	city_name	ArgMax	1
domain4	cpe_model_name	ArgMax	1
domain5	cpe_manufacturer_name	ArgMax	1
domain6	cpe_type_cd	ArgMax	1
domain7	price	ArgMax	1
domain8	date	ArgMax	1
domain9	part_of_day	ArgMax	1
domain10	domain2..9	BoW	1407
domain11	sequences of domain1,2,4,8+9	Doc2Vec	512
domain12	url_host's screenshots	CLIP	768
domain13	url_host's titles	BoW	13118
domain14	url_host's html	BoW+SVD	256
domain15	url_host's similar web stats	BoW	199
domain16	sequences of url_host bigrams	BoW+SVD	512
			Итого: 36924

Домены признаков. domain1

Рецепт:

- Для каждого юзера собрать BoW с учётом значений request_cnt
- Из 200к уникальных значений url_host отобрать 20к самых частотных по Train+Test
- К элементам разряженной матрицы BoW для сглаживания экстремальных значений применить sqrt-преобразование

Домены признаков. domain2..9

Рецепт:

- Для каждого юзера собраны BoW с учётом значений request_cnt
- Из уникальных значений признака оставляем только самые частотные (>40) по Train+Test
- Каждого юзера кодируем индексом его моды из BoW

Домены признаков. domain10

Рецепт:

- Для каждого юзера собраны BoW по признакам из domain2..9 с учётом значений request_cnt
- Из уникальных значений признака оставляем только самые частотные (>40) по Train+Test
- Финальный признаковый домен – конкатенация BoW по столбцам

Домены признаков. domain11

Рецепт:

- Отсортировать последовательности признаков domain1,2,4,8,9 по (user_id, дата, время суток)
- Выставить тэги документа как значения user_id.
- Обучить gensim.models.doc2vec.Doc2Vec на Train+Test для каждой последовательности признаков
- Получить эмбеддинги юзеров как model.docvecs.vector_docs
- Сконкатенировать эмбеддинги юзеров по каждому признаку (url_host, region_name, …) вдоль колонок

Домены признаков. domain12

Рецепт:

- Спарсить топ-20к частотных url_host с помощью Selenium
- Сделать скриншоты главной страницы url_host в разрешении 600x300
- Получить эмбеддинг изображений с помощью CLIP ViT-L/14
- Получить эмбеддинг юзера как TfIdf взвешенной суммы эмбеддингов изображений
- Применить L2 нормализацию

Домены признаков. domain13

Рецепт:

- Спарсить топ-20к частотных url_host с помощью Selenium
- Сохранить title главных страниц
- Векторизовать с помощью BoW для униграм и биграм токенов title
- Обрезать низкочастотные сочетания токенов (<2)
- Получить эмбеддинг юзера как dot-произведение между BoW от url_host и BoW title
- К элементам разряженной матрицы BoW для сглаживания экстремальных значений применить log-преобразование

Домены признаков. domain14

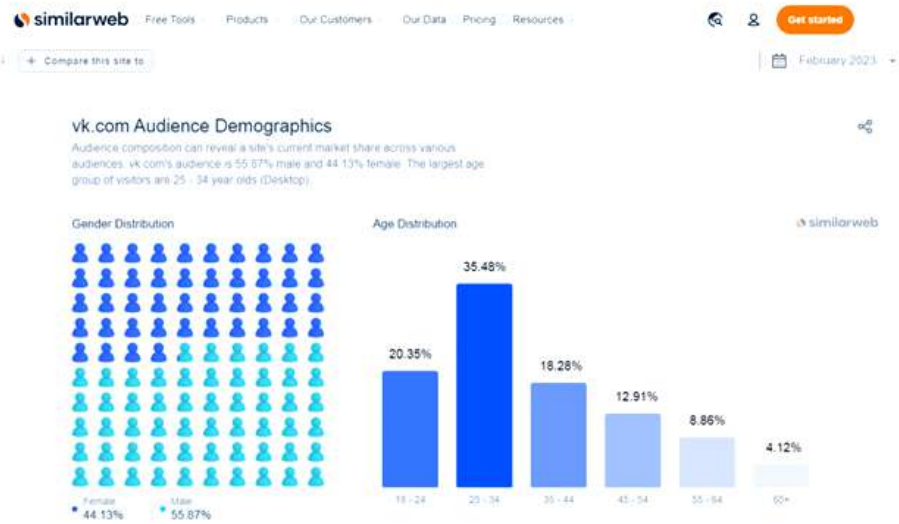
Рецепт:

- Спарсить топ-20к частотных url_host с помощью Selenium
- Сохранить HTML-дерево главных страниц
- Векторизовать с помощью BoW для униграм и биграм токенов HTML
- Обрезать низкочастотные сочетания токенов (<300)
- Обрезать высокочастотные сочетания токенов (>25%)
- Обучить TfIdf на полученном BoW
- Понизить размерность с помощью SVD
- Получить эмбеддинг юзера как TfIdf взвешивание SVD html по BoW от url_host
- Применить L2 нормализацию

Домены признаков. domain15

Рецепт:

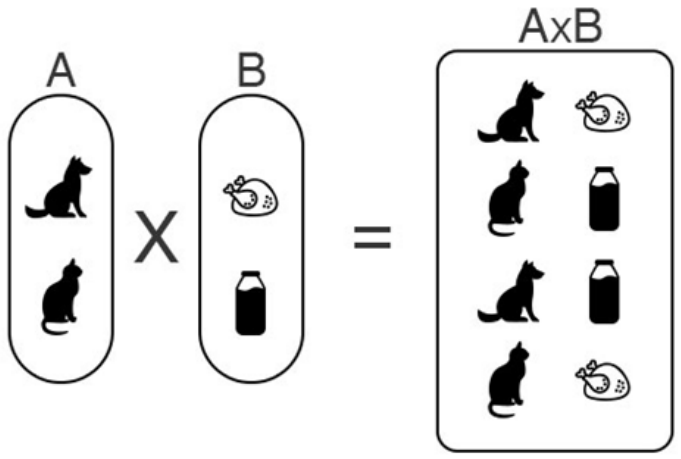
- Спарсить SimilarWeb статистики по топ-20к частотных url_host с помощью Selenium
- С помощью регулярок выделить информацию о ранге сайта, его категории контента и демографии посетителей
- Каждую категориальную переменную закодировать с помощью Tfidf, для возраста и пола вычесть среднее значение по всем сайтам
- Получить эмбеддинг юзера как Tfidf взвешивание по BoW от url_host



Домены признаков. domain16

Рецепт:

- Упорядочить url_host по (user_id, дата, время дня)
- Отфильтровать редкие (<50) url_host
- Посчитать биграммы* значений url_host как декартово произведение между соседними временными окнами (например, вечер 2022-06-10 и ночь 2022-06-11)
- Оставить только самые частотные пары токенов (>80)
- Обучить BoW
- Сгладить экстремальные значения с помощью sqrt-преобразования
- Понизить размерность с помощью SVD



Cartesian Product of Two Sets.

Валидация

public_lb			private_lb			
1	Vladimir Bazhenov	1,7853	1	Vladimir Bazhenov	1,7825	-0,0028
2	Ishkov Denis	1,7751	2	flow	1,7802	+0,0070
3	сансеймайсамба	1,7737	3	Ishkov Denis	1,7788	+0,0037
4	flow	1,7732	4	сансеймайсамба	1,7744	+0,0007
5	Stanislav Chistyakov	1,7715	5	Stanislav Chistyakov	1,7739	+0,0024
6	More Trees Stack	1,7680	6	Anton Karasev	1,771	+0,0048
7	tg kaggle_fucker	1,7667	7	John Galt	1,7674	+0,0033
8	Anton Karasev	1,7662	8	tg kaggle_fucker	1,7638	-0,0029
9	John Galt	1,7641	9	More Trees Stack	1,7634	-0,0046
10	AK	1,7613	10	Ivan Alexandrov	1,7617	+0,0032
11	room722	1,7606	11	DS48_DMVL	1,7616	+0,0041
12	Ivan Alexandrov	1,7585	12	room722	1,7591	-0,0015
13	DS48_DMVL	1,7575	13	AK	1,7585	-0,0028

призовые

- Валидация – стратифицированная кросс-валидация по (пол, бин возраста) на 10 разбиениях
- Гиперпараметры NN подбирались байесовским поиском на одном наборе разбиений, качество удачных архитектур замерялось на альтернативном наборе разбиений
- Использование информации о количестве бинов возраста в public_lb даёт согласованную оценку локальной валидации
- F1_weighted нестабильная метрика, когда классы несбалансированны (класс 66+ встречается всего в 2% всех примеров)

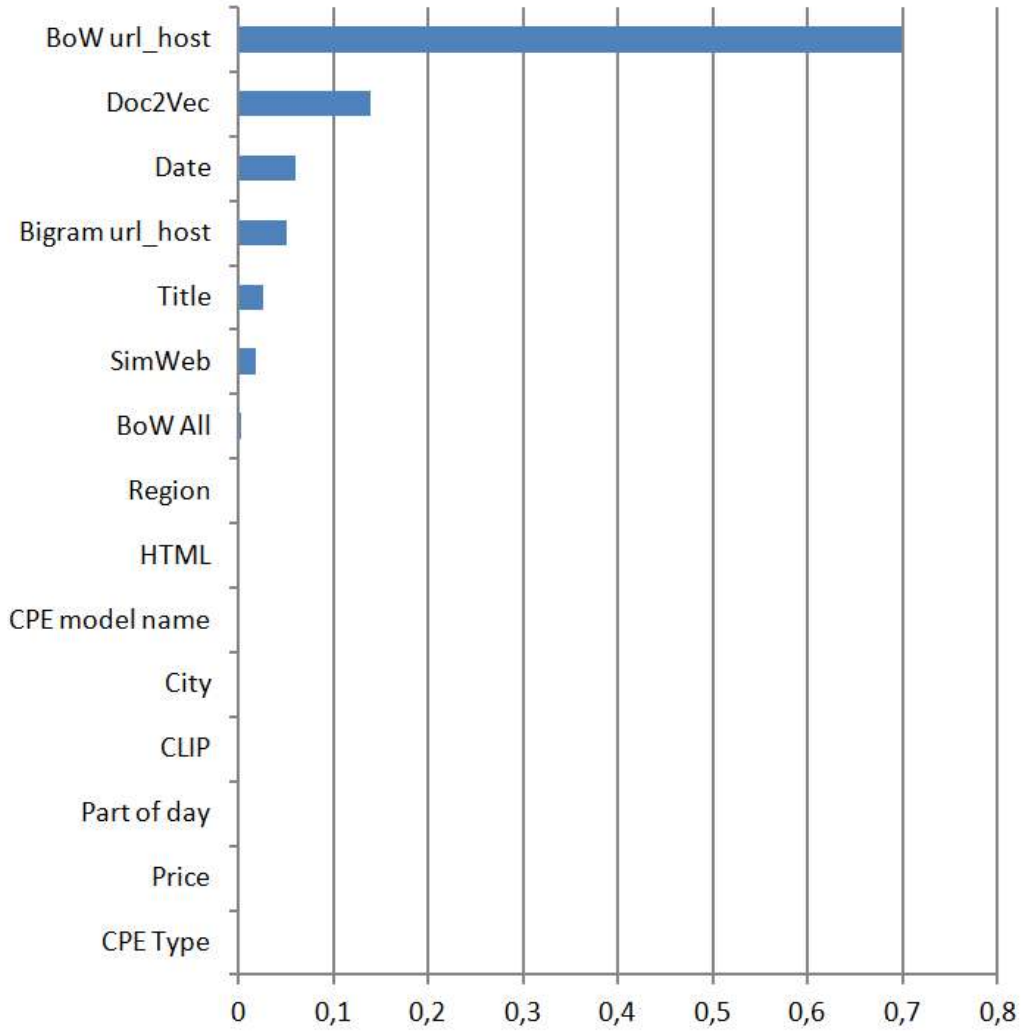
Подходы, которые помогли

- Использование Multi-Output даёт значительные приросты (4 место в последние выходные поднялся на ~5 позиций за счёт этого)
- Ансамблирование скоров до стэкинга (усреднение 3x10 скоров во время локальной валидации одной модели, абсолютный прирост 0,25-0,5%)
- Подбирать гиперпараметры, отдельно максимизируя только GINI или только F1_weighted
- Увеличивать разнообразие моделей за счёт подбора архитектуры на 2-3 тестовых фолдах, а не всех 10
- Сглаживание меток/self-distillation – учим не на оригинальный таргет, а на линейную комбинацию предсказаний самой модели и оригинального таргета
- Pseudo-labelling – обучаем модель на 9 Train фолдах, размечаем ей Train+Test, используем скоры как pretrain грязную выборку, потом fine-tune на исходных 9 фолдах и чистом таргете (данный трюк помог оставаться топ-2 в течение 2 недельного перерыва)
- Алгоритм дискретной оптимизации COBYLA находил хорошие смещения для предсказаний возраста, прирост 0.005 к F1_weighted

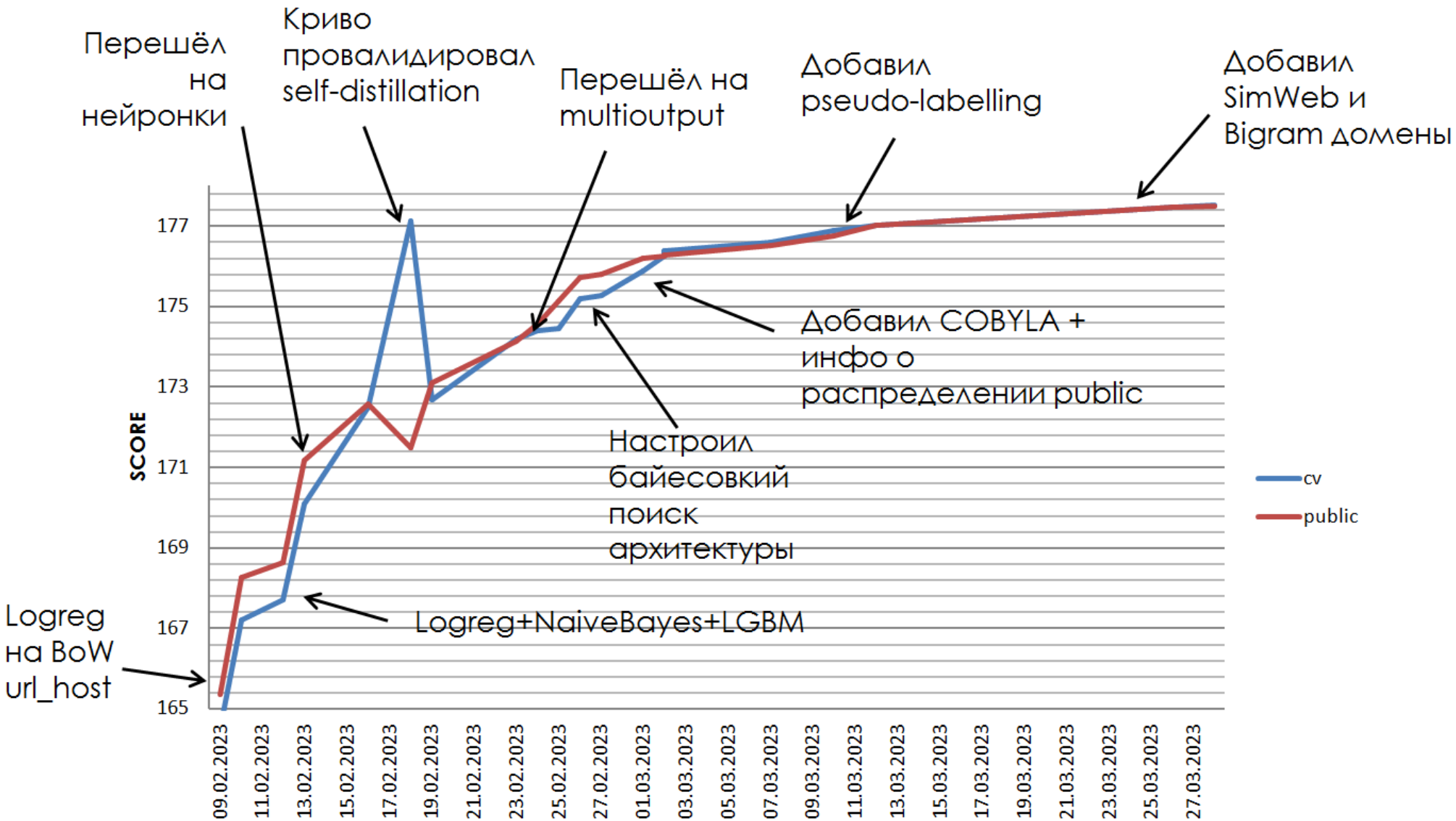
Подходы, которые НЕ помогли

- Ordinal, 1vs1, 1vsRest модели для возраста
- Прямой-обратный перевод, очистка, punycode декодирование url_host
- Регрессия в возраст/бин возраста
- TfIdf преобразование BoW url_host ухудшало качество модели
- Признаки ближайших соседей по BoW url_host
- UMAP признаки на BoW url_host
- Латентное представление автоэнкодера как доп признак
- Бутстрэппирование (обычное, дикое) обучающих фолдов для увеличения разнообразия моделей
- Бустинг нейронных сетей (регрессия на ошибках предыдущей модели)
- Архитектуры на последовательностях (очень прожорливые, нестабильные и долго обучаются)
- Бустинг на деревьях, как на фичах, так и признаках предпоследнего слоя NN*
- Mixup-аугментация примеров
- Кастомные функции потерь (bce -> roc_auc_star, cce -> f1)
- Отбор признаков для стэкинга (BORUTA, Forward, Random)

Permutation feature importance



Сравнение CV и лидерборда



Спасибо за внимание!

- **e-mail**: defasium@yandex.ru
- **tg**: @molibden4ik