

#4EduTech Pipeline обработки данных

Получение данных, преобразование из формата CSV в формат SQL

Данные были предоставлены файлами CSV, т.к. в будущем проект будет получать данные непосредственно из базы данных клиента, делать автоматическую загрузку не было смысла. В процессе работы с данными были сделаны предварительные изменения с названиями полей

users_age_timezone.csv

userID	timeZone	age
982	+03:00	16
2566	+03:00	16
2614	+03:00	15

Table: users_age_timezone

Название	#	Тип данных
123 user_id	1	int4
ABC time_zone	2	varchar(50)
123 age	3	int4

CSV: authorization.csv

user_id	created_at	user_agent	window_size
7722	2022-12-07 17:43:14	Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/106.0.0.0 YaBrowser/22.11.0.2500 Yowser/2.5 Safari/537.36	1872x932
7722	2022-12-07 19:51:34	Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/106.0.0.0 YaBrowser/22.11.0.2500 Yowser/2.5 Safari/537.36	1872x918
2614	2022-12-08 00:59:43	Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/106.0.0.0 YaBrowser/22.11.0.2500 Yowser/2.5 Safari/537.36	1528x755

Table: authorization

Название	#	Тип данных
123 user_id	1	int4
🕒 created_at	2	timestamp
ABC user_agent	3	varchar(256)
ABC window_size	4	varchar(50)

CSV: schedule.csv

course_id	type	taskID	activityID	activityType	isAttestation	visibility	flows	dateShown
3	активность	81	8719	slide	0	{}	1	2023-12-05 11:00:00
3	активность	81	8720	slide	0	{}	1	2023-12-05 11:00:00
3	активность	81	8722	slide	0	{}	1	2023-12-05 11:00:00

Table: schedule

Название	#	Тип данных
123 course_id	1	int4
ABC type	2	varchar(50)
123 task_id	3	int4
123 activity_id	4	int4
ABC activity_type	5	varchar(50)
123 is_attestation	6	int4
ABC visibility	7	varchar(50)
123 flows	8	int4
🕒 date_shown	9	timestamp

CSV: activity_history_viewed.csv

user_id	created_at	page_type	page_id	module	attestation	activity_type
982	2024-01-24 19:14:11	занятие	3015	2	1	
982	2024-01-24 19:34:09	занятие	3015	2	1	
982	2024-01-28 22:35:18	занятие	3015	2	1	

Table: activity_history_viewed

Название	#	Тип данных
123 user_id	1	int4
🕒 created_at	2	timestamp
ABC page_type	3	varchar(50)
123 page_id	4	int4
123 module	5	int4
123 attestation	6	int4
ABC activity_type	7	varchar(50)

CSV: webinars_logs.csv

userId	dateTime	eventName	webinarId	формат подключения	вводный вебинар	module
74952	2023-12-23 12:10:03	Подключение	13378	офлайн	1	2
74952	2023-12-23 12:15:44	Отключение	13378	офлайн	1	2
74952	2023-12-23 12:20:58	Подключение	13378	офлайн	1	2

Table: webinars_logs

Название	#	Тип данных
123 user_id	1	int4
🕒 datetime	2	timestamp
ABC event_name	3	varchar(50)
123 webinar_id	4	int4
ABC conn_format	5	varchar(50)
123 webinar_vvod	6	int4
123 module	7	int4

CSV: exercise_results.csv

module	activityId	userId	createdAt	result	success
2	13278	72002	2023-12-01 08:53:00	100	1
2	13278	66421	2023-12-01 09:18:55	100	1
2	13278	66405	2023-12-01 09:23:24	100	1

Table: exercise_results





Название	#	Тип данных
123 module	1	int4
123 activity_id	2	int4
123 user_id	3	int4
🕒 created_at	4	timestamp
ABC result	5	varchar(50)
123 success	6	int4

CSV: users_logs.csv

user_id	created_at	event	comment
2566	2023-11-14 15:07:40	tags-changed	Были хештеги: #entry-email #офлайн, стали: #entry-email #онлайн

2614	2024-02-19 17:20:49	tags-changed	Были хештеги: #мотив #онлайн #M1-завершил #сверка-дат #M1-оплачен #M2-завершил, стали: #мотив #онлайн #M1-завершил #M1-оплачен #M2-завершил
13849	2024-02-19 17:22:00	tags-changed	Были хештеги: #онлайн #M1-завершил #сверка-дат #M1-оплачен #M2-завершил, стали: #онлайн #M1-завершил #M1-оплачен #M2-завершил









Table: users_logs

Название	#	Тип данных
 user_id	1	int4
 created_at	2	timestamp
 event	3	varchar(50)
 comment	4	varchar(256)

CSV: users.csv

unti_id	userID	course_id	flow_num	tgBot	M2_progress	M2_attestation	M2_attestation_date
1051004	982	3	1.00	—	13	Не сдана	
1118021	2566	3	1.00	—	0	Не сдана	
1120146	2614	77	1.00	подключен	100	80	2024-03-05 17:29:37

Table: users

Название	#	Тип данных
 unti_id	1	int4
 user_id	2	int4
 course_id	3	int4
 flow_num	4	float4
 tg_bot	5	varchar(50)
 m2_progress	6	varchar(50)
 m2_attestation	7	varchar(50)
 m2_attestation_date	8	timestamp

CSV: activities_guide.csv

courseID	Курс	Провайдер	Модуль	themeID	Тема	taskID	Занятие	Task Position	Признак Аттестации	activityID
3	Нейро.PY	1T	2	13	Функции и классы	81	Функции графики и запроса данных	1	0	14823
3	Нейро.PY	1T	2	13	Функции и классы	81	Функции графики и запроса данных	1	0	14824
3	Нейро.PY	1T	2	13	Функции и классы	81	Функции графики и запроса данных	1	0	14825

продолжение

Тип Активности	Активность	Признак Обязательного	Видимость
CodeExercise	Тренажер 1	0	{}
CodeExercise	Тренажер 2	0	{}
CodeExercise	Тренажер 3	0	{}

Table: activities_guide

Название	#	Тип данных
123 course_id	1	int4
ABC course	2	varchar
ABC provider	3	varchar
123 modul	4	int4
123 theme_id	5	int4
ABC theme	6	varchar
123 task_id	7	int4
ABC exercise	8	varchar
123 task_position	9	int4
123 att_priznak	10	int4
123 activity_id	11	int4
ABC activity_type	12	varchar
ABC activity	13	varchar
123 obyaz_priznak	14	int4
ABC visibility	15	varchar

Работа с данными в базе PostgreSQL

Фильтр выборки обучающихся по данным был задан заказчиком, брать тех кто имеет статус **онлайн**

```
with
t1 as(
select distinct user_id, created_at, comment
from public.users_logs logs
where comment like 'Были хештеги:%онлайн%стали:%онлайн%'
or comment like 'Установлен хештег "#онлайн" --'%Установлен%онлайн%'
or comment like 'Были хештеги: , стали:%онлайн%'
or comment like 'Были хештеги%оулайн%стали:%онлайн%'
or comment like 'Были хештеги%оглайн%стали:%онлайн%')

select *
from users
where "userID" in(select distinct user_id from t1)
```

@TatianaGlu

В процессе работы с данными аналитиками были сформированы представления по которым проверялись гипотезы

Коллеги, позволил себе нескромность сделать новые вьюшки из датасетов, которые собрали мы с Таней. По крайней мере мне так удобнее пользоваться, если работать напрямую с базой.

dataset_v - 1й датасет, где все даты расположены по вертикали (V)

dataset_h - 2й датасет, где часть событий сопоставлена по горизонтали (H)

Важное уточнение! Убрал в 1м датасете фильтрацию только успешных студентов (строк стало больше), но сохранил фильтрацию тех студентов, по которым есть след во всех таблицах.

2й датасет сразу был собран по такому принципу.

@ryurikovich_37

dataset_h	dataset_h_v3	dataset_v	glu_dataset
123 user_id	123 user_id	123 user_id	123 user_id
123 course_id	123 course_id	123 course_id	123 age
ABC type	ABC type	ABC event	ABC time_zone
ABC activity_type	ABC activity_type	ABC type	123 course_id
123 task_id	123 task_id	123 task_id	ABC event
123 activity_id	123 activity_id	123 activity_id	ABC time
🕒 date_shown	🕒 date_shown	🕒 time	123 task_id
123 is_attestation	123 is_attestation	123 result	ABC type
🕒 created_at	🕒 created_at	123 success	123 activity_id
123 module	123 module	123 m2_progress	ABC activity_type
123 attestation	123 attestation	123 m2_attestation	123 obyzaz_priznak
🕒 result_time	🕒 result_time	123 age	123 result
123 result	123 result	ABC time_zone	123 success
123 success	123 success	123 age	123 m2_progress
ABC tg_bot	ABC tg_bot		123 m2_attestation
123 m2_progress	123 m2_progress		
🕒 m2_attestation_date	🕒 m2_attestation_date		
123 m2_attestation	123 m2_attestation		
123 age	123 age		
ABC time_zone	ABC time_zone		

Новую версию сохранил как dataset_h_v3

Изменения:

- 1) Данные на основе датасета students_v3 (см. выше)
 - 2) Не стал выкидывать студентов, которые засветились не во всех таблицах
 - 3) Добавил столбец обязательного признака задания
- Датасет стал тяжелым (1 217 959 строк), выполняется долго.

@ryurikovich_37

```
with
cross_schedule as (
select sv2.user_id, sv.course_id, sv."type", sv.activity_type, sv.task_id, sv.activivty_id, sv.date_shown, sv.is_attestation
from schedule_v2 sv
full outer join students_v3 sv2
on sv2.course_id = sv.course_id
),
cross_history as (
select
cross_schedule.user_id, cross_schedule.course_id, cross_schedule."type", cross_schedule.activity_type, cross_schedule.task_id,
cross_schedule.activivty_id, cross_schedule.date_shown, cross_schedule.is_attestation,
ahvv.created_at, ahvv."module", ahvv.attestation
from cross_schedule
left join activity_history_viewed_v2 ahvv
on (ahvv.user_id = cross_schedule.user_id) and (ahvv.page_id = cross_schedule.task_id)
where type = 'занятие'
union
select cross_schedule.user_id, cross_schedule.course_id, cross_schedule."type", cross_schedule.activity_type, cross_schedule.task_id,
cross_schedule.activivty_id, cross_schedule.date_shown, cross_schedule.is_attestation,
ahvv.created_at, ahvv."module", ahvv.attestation
from cross_schedule
left join activity_history_viewed_v2 ahvv
on (ahvv.user_id = cross_schedule.user_id) and (ahvv.page_id = cross_schedule.activivty_id)
where type = 'активность'),
history_results as (
select cross_history.*, erv.created_at as result_time,
case when erv.result = 'Пропуск' then -1 else cast(erv.result as int4) end as result,
erv.success
from cross_history
left join exercise_results_v2 erv
on (erv.user_id = cross_history.user_id) and (erv.activity_id = cross_history.activivty_id)
)
select history_results.*, agv.obyaz_priznak, sv.tg_bot,
case when sv.m2_progress = 'Нет данных' then -1 else cast(sv.m2_progress as int4) end as m2_progress,
sv.m2_attestation_date,
case when sv.m2_attestation = 'Не сдана' then -1 else cast(sv.m2_attestation as int4) end as m2_attestation,
uatv.age, uatv.time_zone
from history_results
left join students_v3 sv
on sv.user_id = history_results.user_id
left join users_age_timezone_v2 uatv
on history_results.user_id = uatv.user_id
left join activities_guide_v2 agv
on agv.activity_id = history_results.activivty_id
order by history_results.user_id, history_results.date_shown
```

m_course_params 123 course_id ABC name 🕒 min_date_shown 🕒 max_date_shown 123 num_tasks 123 num_activity 123 num_attestation_activity ABC attestation_activity_types 123 num_webinar_v2 123 num_questions 123 num_exercise 123 num_interactive 123 num_codeexercise 123 num_slide 123 num_empty_activity	m_online_users_exercise 123 user_id 123 course_id 123 task_id 123 m 🕒 activity_datetime_shown 🕒 result_datetime_upd 123 success 123 num_success 123 num_activities_all 123 max_num_repeat 123 is_attestation ABC m2_progress ABC m2_attestation 🕒 m2_attestation_date	m_user_tags_history 123 user_id 🕒 from_datetime 🕒 to_datetime ABC last_tag ABC tag <input checked="" type="checkbox"/> from_users_logs ABC comment	m_exercise_lag_time_series 123 user_id 123 course_id 123 day_num 🕒 date_column 123 sum_lag ABC m2_progress ABC m2_attestation 🕒 m2_attestation_date
		m_online_students 123 unti_id 123 user_id 123 course_id 123 flow_num ABC tg_bot ABC m2_progress ABC m2_attestation 🕒 m2_attestation_date ABC learning_format	m_date_interval 123 day_num 🕒 date_column

students_v2 123 unti_id 123 user_id 123 course_id 123 flow_num ABC tg_bot ABC m2_progress ABC m2_attestation 🕒 m2_attestation_date	students_v3 123 unti_id 123 user_id 123 course_id 123 flow_num ABC tg_bot ABC m2_progress ABC m2_attestation 🕒 m2_attestation_date
-------------------------------------------------------------------------------------------------------------------------------------------------------------------	-------------------------------------------------------------------------------------------------------------------------------------------------------------------

По представлениям много было испробовано и отброшено общим решением DA, после долгих споров, анализов и уточнений у заказчиков оставили текущие версии.

запросы, которыми формировали стартовые датасеты?

интересно, как считали sum_m2_progress и остальные значения

```

with
events as(
select user_id, created_at as time, 'authorization' as event, " as type, " as activity_type, 0 as task_id, 0 as activity_id, 0 as result, 0 as success, 0 as m2_progress, 0 as
m2_attestation
from authorization_v2 av
where user_id in (select user_id from public.students_v3)
union
select user_id, created_at as time, event, " as type, " as activity_type, 0 as task_id, 0 as activity_id, 0 as result, 0 as success, 0 as m2_progress, 0 as m2_attestation
from users_logs_v2 ulv
where user_id in (select user_id from public.students_v3)
union
select user_id, m2_attestation_date as time, 'attestation' as event, " as type, " as activity_type, 0 as task_id, 0 as activity_id, 0 as result, 0 as success,
case when m2_progress = 'Нет данных' then -1 else cast(m2_progress as int4) end as m2_progress,
case when m2_attestation = 'Не сдана' then -1 else cast(m2_attestation as int4) end as m2_attestation
from students_v2

```

```

where user_id in (select user_id from public.students_v3)
union
select user_id, created_at as time, 'history' as event, page_type as type, activity_type, 0 as task_id, null as activity_id, 0 as result, 0 as success, 0 as m2_progress, 0 as
m2_attestation
from public.activity_history_viewed_v2
where (page_type = 'занятие') and (user_id in (select user_id from public.students_v3))
union
select ahvv.user_id, ahvv.created_at as time, 'history' as event, page_type as type, ahvv.activity_type
, sv.task_id as task_id, ahvv.page_id as activity_id, 0 as result, 0 as success, 0 as m2_progress, 0 as m2_attestation
from public.activity_history_viewed_v2 ahvv
left join schedule_v2 sv on ahvv.page_id = sv.activity_id
where (ahvv.page_type = 'активность') and (ahvv.user_id in (select user_id from public.students_v3))
union
select user_id, created_at as time, 'results' as event, '' as type, '' as activity_type, 0 as task_id, activity_id,
case when result = 'Пропуск' then -1 else cast(result as int4) end as result, success as success, 0 as m2_progress, 0 as m2_attestation
from public.exercise_results_v2 erv
where user_id in (select user_id from public.students_v3)
union
select user_id, datetime as time, event_name as event, 'Вебинар' as type, '' as activity_type, 0 as task_id, webinar_id as activity_id, 0 as result, 0 as success, 0 as
m2_progress, 0 as m2_attestation
from public.webinars_logs_v2
where user_id in (select user_id from public.students_v3)
)
,
--выбираем юзеров сдавших аттестацию по модулю вовремя
success_intime_users as(
select user_id from public.users_v2 --where m2_attestation_date between '2023-12-01 00:00:00.000' and '2024-02-01 00:00:00.000' закомментировали для неуспешных
)
,
--собрали user_id, кто сдал аттестацию
success_users as(
select user_id from events where m2_attestation >= 50 and m2_progress >= 50 and user_id in (select user_id from success_intime_users)
)
,
--подтягиваем номер группы, часовой пояс и возраст
events_with_personal_info as(
select ev.user_id, users.course_id, ev.time, ev.event, ev.type, ev.activity_type, ev.task_id, ev.activity_id, ev.result, ev.success, ev.m2_progress, ev.m2_attestation
, uat.time_zone, uat.age
from events ev
left join public.users_v2 users
on ev.user_id=users.user_id
left join public.users_age_timezone_v2 uat
on ev.user_id=uat.user_id
union
select 0 as user_id, course_id, date_shown as time, 'schedule' as event, type, activity_type, task_id, activity_id, 0 as result, 0 as success, 0 as m2_progress, 0 as
m2_attestation, '' as time_zone, 0 as age
from public.schedule_v2 sv2
order by user_id, time
)
,
--выбрали юзеров из 49 группы успешно сдавших модуль и разбили их по неделям
dataset_vsev_suc as(
select user_id, age
, date_part('year', time) as year, date_part('week', time) week
-- , count(event) filter (where event = 'authorization') authorization_count
, count(event) filter (where event = 'history') all_activity_count
, count(event) filter (where event = 'attestation') attestation_event_count
, count(event) filter (where event = 'results') results_event_count
-- , count(activity_type) filter (where activity_type = 'CodeExercise') "CodeExercise_count"

```



```

-- , count(activity_type) filter (where activity_type = 'interactive') interactive_count
-- , count(activity_type) filter (where activity_type = 'slide') slide_count
-- , count(activity_type) filter (where activity_type = 'exercise') exercise_count
, sum(result) total_score
, sum(success) success_attempts
, count(success) total_attempts
, round((1.0*sum(success)/count(success)), 2) success_rate --удачные попытки / все попытки
, case when sum(success)!= 0 then 1.0*sum(result)/sum(success) else 0 end as avg_success_score --ср.балл удачных попыток сдачи
, case when count(success)!= 0 then 1.0*sum(result)/count(success) else 0 end as avg_score --средний балл всех попыток сдачи заданий
, sum(m2_progress) sum_m2_progress
from events_with_personal_info
where user_id in(select user_id from success_users) and user_id != 0
and course_id=49
and date_part('week', time) in(1, 2, 3, 4, 5, 48, 49, 50, 51, 52)
group by user_id, year, week, age
order by user_id, year, week
)

select * from dataset_vsev_suc

```

@TatianaGlu

20241024_dataset_timeseries_with_metrics_v_gavrilova
123 id
123 user_id
123 payment
123 time_zone
123 age
123 unti_id
123 course_id
123 flow_num
123 tg_bot
123 m2_progress
123 m2_attestation
123 module
123 m2_delay
123 sum_auth
123 sum_schedule_activities
123 sum_required_activity
123 sum_attestation_activity
123 view_delay_first
123 view_delay_sum
123 sum_activity_viewed
123 sum_required_activity_viewed
123 sum_attestation_activity_viewed
123 w_view_hours
123 sum_exercise
123 sum_required_exercises
123 sum_attestation_exercises
123 exercise_delay_first
123 exercise_delay_sum
123 result_delay_mean
123 result_delay_sum
123 sum_exercise_attempts_mean
123 mean_required_result
123 mean_non_req_result
123 mean_attestation_result
123 sum_result
123 conn
123 online_rate
123 mean_result
123 progress
ABC cur_date

Одно из представлений сделали таблицей и вывели в общее использование.

Работа DS над данными

Код разработан пользователем с телеграмм username @zloy

Прогнозирование показателя результативности обучения на курсах

Исходные данные: выборки, содержащие показатели прохождения курса обучающимися.

1. course_49_lubov_dataset.csv - датасет с данными учеников 49 группы.

Задача: используя модели машинного обучения, спрогнозировать показатель результативности прохождения курса.

	user_id	age	m2_delay_1_week	m2_delay_2_week	m2_delay_3_week	m2_delay_4_week	m2_delay_5_week	m2_delay_6_week	m2_delay_7_week
1	23052	15	-58.458332	-58.458332	-58.458332	-58.458332	-58.458332	-58.458332	-58.458332
2	29079	15	-58.458332	-58.458332	-58.458332	-58.458332	-58.458332	-58.458332	-58.458332
	m2_delay_8_week	m2_delay_9_week	m2_delay_10_week	view_delay_first_1_w...	view_delay_first_2_w...	view_delay_first_3_w...	view_delay_first_4_w...	view_delay_first_5_w...	view_delay_first_6_w...
1	-58.458332	-58.458332	-58.458332	0.0	0.0	0.0	0.0	0.0	0.0
2	-58.458332	-58.458332	-58.458332	0.0	0.0	0.0	0.0	0.0	0.0
	view_delay_first_7_w...	view_delay_first_8_w...	view_delay_first_9_w...	view_delay_first_10_w...	view_delay_sum_1_w...	view_delay_sum_2_w...	view_delay_sum_3_w...	view_delay_sum_4_w...	view_delay_sum_5_w...
1	-0.18401042	-0.17836145	-0.17836145	-0.17836145	0.0	0.0	0.0	0.0	0.0
2	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
	view_delay_sum_6_w...	view_delay_sum_7_w...	view_delay_sum_8_w...	view_delay_sum_9_w...	view_delay_sum_10_w...	sum_activity_viewed...	sum_activity_viewed...	sum_activity_viewed...	sum_activity_viewed...
1	0.0	-15.484445	-24.888681	-24.888681	-24.888681	0.0	0.0	0.0	0.0
2	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
	sum_activity_viewed...	sum_activity_viewed...	sum_activity_viewed...	sum_activity_viewed...	sum_activity_viewed...	sum_activity_viewed...	sum_activity_viewed...	sum_activity_viewed...	sum_activity_viewed...
1	0.0	0.0	70.0	111.0	111.0	111.0	0.0	0.0	0.0
2	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
	sum_required_activity...	sum_required_activity...	sum_required_activity...	sum_required_activity...	sum_required_activity...	sum_required_activity...	sum_required_activity...	sum_required_activity...	sum_required_activity...
1	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
2	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
	sum_exercise_1_week	sum_exercise_2_week	sum_exercise_3_week	sum_exercise_4_week	sum_exercise_5_week	sum_exercise_6_week	sum_exercise_7_week	sum_exercise_8_week	sum_exercise_9_week
1	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
2	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
	sum_exercise_10_week	sum_exercise_11_week	sum_exercise_12_week	sum_exercise_13_week	sum_exercise_14_week	sum_exercise_15_week	sum_exercise_16_week	sum_exercise_17_week	sum_exercise_18_week
1	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
2	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
	sum_exercise_19_week	sum_exercise_20_week	sum_exercise_21_week	sum_exercise_22_week	sum_exercise_23_week	sum_exercise_24_week	sum_exercise_25_week	sum_exercise_26_week	sum_exercise_27_week
1	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
2	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
	sum_exercise_28_week	sum_exercise_29_week	sum_exercise_30_week	sum_exercise_31_week	sum_exercise_32_week	sum_exercise_33_week	sum_exercise_34_week	sum_exercise_35_week	sum_exercise_36_week
1	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
2	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
	sum_exercise_37_week	sum_exercise_38_week	sum_exercise_39_week	sum_exercise_40_week	sum_exercise_41_week	sum_exercise_42_week	sum_exercise_43_week	sum_exercise_44_week	sum_exercise_45_week
1	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
2	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
	sum_exercise_46_week	sum_exercise_47_week	sum_exercise_48_week	sum_exercise_49_week	sum_exercise_50_week	sum_exercise_51_week	sum_exercise_52_week	sum_exercise_53_week	sum_exercise_54_week
1	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
2	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
	sum_exercise_55_week	sum_exercise_56_week	sum_exercise_57_week	sum_exercise_58_week	sum_exercise_59_week	sum_exercise_60_week	sum_exercise_61_week	sum_exercise_62_week	sum_exercise_63_week
1	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
2	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
	sum_exercise_64_week	sum_exercise_65_week	sum_exercise_66_week	sum_exercise_67_week	sum_exercise_68_week	sum_exercise_69_week	sum_exercise_70_week	sum_exercise_71_week	sum_exercise_72_week
1	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
2	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
	sum_exercise_73_week	sum_exercise_74_week	sum_exercise_75_week	sum_exercise_76_week	sum_exercise_77_week	sum_exercise_78_week	sum_exercise_79_week	sum_exercise_80_week	sum_exercise_81_week
1	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
2	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
	sum_exercise_82_week	sum_exercise_83_week	sum_exercise_84_week	sum_exercise_85_week	sum_exercise_86_week	sum_exercise_87_week	sum_exercise_88_week	sum_exercise_89_week	sum_exercise_90_week
1	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
2	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
	sum_exercise_91_week	sum_exercise_92_week	sum_exercise_93_week	sum_exercise_94_week	sum_exercise_95_week	sum_exercise_96_week	sum_exercise_97_week	sum_exercise_98_week	sum_exercise_99_week
1	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
2	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
	sum_exercise_100_week	sum_exercise_101_week	sum_exercise_102_week	sum_exercise_103_week	sum_exercise_104_week	sum_exercise_105_week	sum_exercise_106_week	sum_exercise_107_week	sum_exercise_108_week
1	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
2	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
	sum_exercise_109_week	sum_exercise_110_week	sum_exercise_111_week	sum_exercise_112_week	sum_exercise_113_week	sum_exercise_114_week	sum_exercise_115_week	sum_exercise_116_week	sum_exercise_117_week
1	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
2	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
	sum_exercise_118_week	sum_exercise_119_week	sum_exercise_120_week	sum_exercise_121_week	sum_exercise_122_week	sum_exercise_123_week	sum_exercise_124_week	sum_exercise_125_week	sum_exercise_126_week
1	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
2	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
	sum_exercise_127_week	sum_exercise_128_week	sum_exercise_129_week	sum_exercise_130_week	sum_exercise_131_week	sum_exercise_132_week	sum_exercise_133_week	sum_exercise_134_week	sum_exercise_135_week
1	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
2	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
	sum_exercise_136_week	sum_exercise_137_week	sum_exercise_138_week	sum_exercise_139_week	sum_exercise_140_week	sum_exercise_141_week	sum_exercise_142_week	sum_exercise_143_week	sum_exercise_144_week
1	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
2	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
	sum_exercise_145_week	sum_exercise_146_week	sum_exercise_147_week	sum_exercise_148_week	sum_exercise_149_week	sum_exercise_150_week	sum_exercise_151_week	sum_exercise_152_week	sum_exercise_153_week
1	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
2	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
	sum_exercise_154_week	sum_exercise_155_week	sum_exercise_156_week	sum_exercise_157_week	sum_exercise_158_week	sum_exercise_159_week	sum_exercise_160_week	sum_exercise_161_week	sum_exercise_162_week
1	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
2	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
	sum_exercise_163_week	sum_exercise_164_week	sum_exercise_165_week	sum_exercise_166_week	sum_exercise_167_week	sum_exercise_168_week	sum_exercise_169_week	sum_exercise_170_week	sum_exercise_171_week
1	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
2	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
	sum_exercise_172_week	sum_exercise_173_week	sum_exercise_174_week	sum_exercise_175_week	sum_exercise_176_week	sum_exercise_177_week	sum_exercise_178_week	sum_exercise_179_week	sum_exercise_180_week
1	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
2	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
	sum_exercise_181_week	sum_exercise_182_week	sum_exercise_183_week	sum_exercise_184_week	sum_exercise_185_week	sum_exercise_186_week	sum_exercise_187_week	sum_exercise_188_week	sum_exercise_189_week
1	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
2	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
	sum_exercise_190_week	sum_exercise_191_week	sum_exercise_192_week	sum_exercise_193_week	sum_exercise_194_week	sum_exercise_195_week	sum_exercise_196_week	sum_exercise_197_week	sum_exercise_198_week
1	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
2	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
	sum_exercise_199_week	sum_exercise_200_week	sum_exercise_201_week	sum_exercise_202_week	sum_exercise_203_week	sum_exercise_204_week	sum_exercise_205_week	sum_exercise_206_week	sum_exercise_207_week
1	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
2	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
	sum_exercise_208_week	sum_exercise_209_week	sum_exercise_210_week	sum_exercise_211_week	sum_exercise_212_week	sum_exercise_213_week	sum_exercise_214_week	sum_exercise_215_week	sum_exercise_216_week
1	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
2	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
	sum_exercise_217_week	sum_exercise_218_week	sum_exercise_219_week	sum_exercise_220_week	sum_exercise_221_week	sum_exercise_222_week	sum_exercise_223_week	sum_exercise_224_week	sum_exercise_225_week
1	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
2	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
	sum_exercise_226_week	sum_exercise_227_week	sum_exercise_228_week	sum_exercise_229_week	sum_exercise_230_week	sum_exercise_231_week	sum_exercise_232_week	sum_exercise_233_week	sum_exercise_234_week
1	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
2	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
	sum_exercise_235_week	sum_exercise_236_week	sum_exercise_237_week	sum_exercise_238_week	sum_exercise_239_week	sum_exercise_240_week	sum_exercise_241_week	sum_exercise_242_week	sum_exercise_243_week
1	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
2	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
	sum_exercise_244_week	sum_exercise_245_week	sum_exercise_246_week	sum_exercise_247_week	sum_exercise_248_week	sum_exercise_249_week	sum_exercise_250_week	sum_exercise_251_week	sum_exercise_252_week
1	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
2	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
	sum_exercise_253_week	sum_exercise_254_week	sum_exercise_255_week	sum_exercise_256_week	sum_exercise_257_week	sum_exercise_258_week	sum_exercise_259_week	sum_exercise_260_week	sum_exercise_261_week
1	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
2	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
	sum_exercise_262_week	sum_exercise_263_week	sum_exercise_264_week	sum_exercise_265_week	sum_exercise_266_week	sum_exercise_267_week	sum_exercise_268_week	sum_exercise_269_week	sum_exercise_270_week
1	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
2	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
	sum_exercise_271_week	sum_exercise_272_week	sum_exercise_273_week	sum_exercise_274_week	sum_exercise_275_week	sum_exercise_276_week	sum_exercise_277_week	sum_exercise_278_week	sum_exercise_279_week
1	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
2	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
	sum_exercise_280_week	sum_exercise_281_week	sum_exercise_282_week	sum_exercise_283_week	sum_exercise_284_week	sum_exercise_285_week	sum_exercise_286_week	sum_exercise_287_week	sum_exercise_288_week
1	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
2	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
	sum_exercise_289_week	sum_exercise_290_week	sum_exercise_291_week	sum_exercise_292_week	sum_exercise_293_week	sum_exercise_294_week	sum_exercise_295_week	sum_exercise_296_week	sum_exercise_297_week
1	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
2	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
	sum_exercise_298_week	sum_exercise_299_week	sum_exercise_300_week	sum_exercise_301_week	sum_exercise_302_week	sum_exercise_303_week	sum_exercise_304_week	sum_exercise_305_week	sum_exercise_306_week
1	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
2	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
	sum_exercise_307_week	sum_exercise_308_week	sum_exercise_309_week	sum_exercise_310_week	sum_exercise_311_week	sum_exercise_312_week	sum_exercise_313_week	sum_exercise_314_week	sum_exercise_315_week
1	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
2	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
	sum_exercise_316_week	sum_exercise_317_week	sum_exercise_318_week	sum_exercise_319_week	sum_exercise_320_week	sum_exercise_321_week	sum_exercise_322_week	sum_exercise_323_week	sum_exercise_324_week
1	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
2	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
	sum_exercise_325_week	sum_exercise_326_week	sum_exercise_327_week	sum_exercise_328_week	sum_exercise_329_week	sum_exercise_330_week	sum_exercise_331_week	sum_exercise_332_week	sum_exercise_333_week
1	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
2	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
	sum_exercise_334_week	sum_exercise_335_week	sum_exercise_336_week	sum_exercise_337_week					

```

# Формируем стратифицированную обучающую выборку

import pandas as pd

# Предполагаем, что df уже определен

# Шаг 1: Выбор 30 записей из группы course_id=3
group_3_sample = df[df['course_id'] == 3].sample(n=30, replace=True)

# Шаг 2: Размножение выбранных записей до 500
group_3_expanded = group_3_sample.sample(n=500, replace=True)

# Шаг 3: Выбор 15 записей из группы course_id=77
group_77_sample = df[df['course_id'] == 77].sample(n=15, replace=True)

# Шаг 4: Размножение выбранных записей до 500
group_77_expanded = group_77_sample.sample(n=500, replace=True)

# Шаг 5: Получение стратифицированной выборки по 500 образцов из остальных групп
other_groups = df[df['course_id'] != 77]
other_groups = other_groups[other_groups['course_id'] != 3] # Исключаем группу course_id=3

# Получаем уникальные значения course_id для остальных групп
other_course_ids = other_groups['course_id'].unique()

# Проверяем, что у нас есть как минимум две другие группы
if len(other_course_ids) < 2:
    raise ValueError("Ожидается как минимум две другие группы.")

# Шаг 6: Выбор по 500 образцов из каждой другой группы
stratified_samples = []
for course_id in other_course_ids:
    samples = other_groups[other_groups['course_id'] == course_id].sample(n=500, replace=True)
    stratified_samples.append(samples)

# Объединение всех выборок
train_sample = pd.concat([group_3_expanded, group_77_expanded] + stratified_samples)

# Проверка результата
print("\nРазмер итоговой выборки:", len(train_sample))
print("\nКоличество записей из course_id=3:", len(train_sample[train_sample['course_id'] == 3]))
print("\nКоличество записей из course_id=77:", len(train_sample[train_sample['course_id'] == 77]))
print("\nКоличество записей из других групп:")
for course_id in other_course_ids:
    print(f"course_id={course_id}: {len(train_sample[train_sample['course_id'] == course_id])}")

```

Размер итоговой выборки: 2000

Количество записей из course_id=3: 500

Количество записей из course_id=77: 500

Количество записей из других групп:

course_id=71: 500

course_id=49: 500

```
# Создаем валидационную выборку методом исключения из датасета обучающих примеров
val_sample = df.drop(train_sample.index)
```

```
# Проверка результатов
print("Обучающая стратифицированная выборка:")
print(len(train_sample))
print("\nВалидационная выборка:")
print(len(val_sample))
```

Обучающая стратифицированная выборка:
2000

Валидационная выборка:
1880

```
# остатки 77 группы для тестирования
df_71_test = val_sample[val_sample['course_id'] == 77]
df_71_test = val_sample[val_sample['course_id'] == 71]
df_49_test = val_sample[val_sample['course_id'] == 49]
df_3_test = val_sample[val_sample['course_id'] == 3]
```

df_3_test

	user_id	course_id	age	payment	tg_bot	m2_delay	view_delay_first_1_week	view_delay_first_2_week	view_delay_first_3_week
50	31678	3	15	0	1	-58.458332	-4.496157	-2.752473	-2.886936
73	33026	3	19	0	1	-45.245810	-3.194155	-2.963096	-3.352257
238	51059	3	19	0	2	0.725949	266.864600	25.554272	25.554272
341	55218	3	15	1	1	0.671539	0.000000	0.000000	0.000000
494	66184	3	18	0	1	-58.458332	12.727256	12.727256	12.727256
508	66405	3	19	0	1	8.577789	13.070785	12.799987	12.799987
1335	69383	3	19	0	2	-0.379537	0.000000	-4.597754	-4.212048

Выводы:

- в датасете нет пропусков, он хорошо подходит для обучения моделей;
- для прогнозирования целевого признака m2_progress в различные периоды необходимо сформировать датасеты для каждой недели, что позволит исключить "подглядывание в будущее".

1.3. Создание отдельных датасетов для каждой недели

1.3. Создание отдельных датасетов для каждой недели

```
# train_sample.columns.tolist()
```

```
from itertools import chain

# Определяем названия столбцов для каждой недели
sum_exercise_week_10 = [f'sum_exercise_{i}_week' for i in range(1, 11)]
sum_required_exercises_week_10 = [f'sum_required_exercises_{i}_week' for i in range(1, 11)]
result_delay_sum_week_10 = [f'result_delay_sum_{i}_week' for i in range(1, 11)]
sum_exercise_attempts_mean_week_10 = [f'sum_exercise_attempts_mean_{i}_week' for i in range(1, 11)]
sum_result_week_10 = [f'sum_result_{i}_week' for i in range(1, 11)]
mean_result_week_10 = [f'mean_result_{i}_week' for i in range(1, 11)]
progress_week_10 = [f'progress_{i}_week' for i in range(1, 11)]

view_delay_first_week_10 = [f'view_delay_first_{i}_week' for i in range(1, 11)]
view_delay_sum_week_10 = [f'view_delay_sum_{i}_week' for i in range(1, 11)]
# sum_activity_viewed_week_10 = [f'sum_activity_viewed_{i}_week' for i in range(1, 11)]
# sum_required_activity_viewed_week_10 = [f'sum_required_activity_viewed_{i}_week' for i in range(1, 11)]
# exercise_delay_first_week_10 = [f'exercise_delay_first_{i}_week' for i in range(1, 11)]
exercise_delay_sum_week_10 = [f'exercise_delay_sum_{i}_week' for i in range(1, 11)]

# Уменьшаем количество элементов в списке для каждой последующей недели
def generate_weekly_columns(base_list, weeks):
    return [base_list[:i] for i in range(1, weeks + 1)]

# Генерируем названия столбцов по неделям
sum_exercise_weeks = generate_weekly_columns(sum_exercise_week_10, 10)
sum_required_exercises_weeks = generate_weekly_columns(sum_required_exercises_week_10, 10)
result_delay_sum_weeks = generate_weekly_columns(result_delay_sum_week_10, 10)
sum_exercise_attempts_mean_weeks = generate_weekly_columns(sum_exercise_attempts_mean_week_10, 10)
sum_result_weeks = generate_weekly_columns(sum_result_week_10, 10)
mean_result_weeks = generate_weekly_columns(mean_result_week_10, 10)
progress_weeks = generate_weekly_columns(progress_week_10, 10)

view_delay_first_weeks = generate_weekly_columns(view_delay_first_week_10, 10)
view_delay_sum_weeks = generate_weekly_columns(view_delay_sum_week_10, 10)
# sum_activity_viewed_weeks = generate_weekly_columns(sum_activity_viewed_week_10, 10)
# sum_required_activity_viewed_weeks = generate_weekly_columns(sum_required_activity_viewed_week_10, 10)
# exercise_delay_first_weeks = generate_weekly_columns(exercise_delay_first_week_10, 10)
exercise_delay_sum_weeks = generate_weekly_columns(exercise_delay_sum_week_10, 10)
```

```
# Формируем столбцы для каждой недели
columns = []
for i in range(10):
    columns.append(list(chain(
        ['user_id', 'age', 'payment', 'tg_bot', 'm2_delay'],
        view_delay_first_weeks[i],
        view_delay_sum_weeks[i],
        sum_exercise_weeks[i],
        sum_required_exercises_weeks[i],
        # exercise_delay_first_weeks[i],
        result_delay_sum_weeks[i],
        sum_exercise_attempts_mean_weeks[i],
        sum_result_weeks[i],
        mean_result_weeks[i],
        progress_weeks[i],
        # view_delay_sum_weeks[i],
        # sum_activity_viewed_weeks[i],
        # sum_required_activity_viewed_weeks[i],
        exercise_delay_sum_weeks[i],
        ['m2_progress']
    )))

# Теперь на позиции `columns[0]` у нас находятся столбцы для 1-й недели,
# на `columns[1]` для 2-й и так далее до 10-й недели.
```

Week 1 - Mean Absolute Error: 7.9924794391680525
Week 2 - Mean Absolute Error: 6.924550653267929
Week 3 - Mean Absolute Error: 6.700902867699872
Week 4 - Mean Absolute Error: 6.300852912703257
Week 5 - Mean Absolute Error: 5.999639314988462
Week 6 - Mean Absolute Error: 5.704828925945423
Week 7 - Mean Absolute Error: 4.9113169553779565
Week 8 - Mean Absolute Error: 3.818703643099225
Week 9 - Mean Absolute Error: 2.934018585007375
Week 10 - Mean Absolute Error: 2.8606435373430092

rf_models

Сериализация моделей!!!

```
import joblib # Импортируем библиотеку для сериализации
import os
```

Укажите директорию для сохранения моделей

```
model_dir = '/home/shared_notebooks/zloy/saved_models'
```

Создаем директорию, если она не существует

```
os.makedirs(model_dir, exist_ok=True)
```












```
{'rf_model_week_1': RandomForestRegressor(max_features='log2', n_estimators=50),
 'rf_model_week_2': RandomForestRegressor(max_features='log2', n_estimators=50),
 'rf_model_week_3': RandomForestRegressor(max_features='log2', n_estimators=50),
 'rf_model_week_4': RandomForestRegressor(max_depth=30, max_features='log2', n_estimators=200),
 'rf_model_week_5': RandomForestRegressor(max_features='log2', n_estimators=50),
 'rf_model_week_6': RandomForestRegressor(max_features='log2', n_estimators=200),
 'rf_model_week_7': RandomForestRegressor(max_features='sqrt', n_estimators=200),
 'rf_model_week_8': RandomForestRegressor(max_features='sqrt', n_estimators=200),
 'rf_model_week_9': RandomForestRegressor(max_features='sqrt', n_estimators=200),
 'rf_model_week_10': RandomForestRegressor(max_features='sqrt', n_estimators=200)}
```

Сериализация моделей на диск

```
for model_name, model in rf_models.items():
    model_filename = os.path.join(model_dir, f'{model_name}.joblib') # Создаем имя файла для модели
    joblib.dump(model, model_filename) # Сохраняем модель в файл
    print(f"Модель '{model_name}' успешно сохранена в '{model_filename}'")

print(f"\nВсе модели успешно сохранены в директорию: {model_dir}")
```

В результате получены модели

 rf_model_week_1.joblib	8 минут назад
 rf_model_week_10.joblib	8 минут назад
 rf_model_week_2.joblib	8 минут назад
 rf_model_week_3.joblib	8 минут назад
 rf_model_week_4.joblib	8 минут назад
 rf_model_week_5.joblib	8 минут назад
 rf_model_week_6.joblib	8 минут назад
 rf_model_week_7.joblib	8 минут назад
 rf_model_week_8.joblib	8 минут назад
 rf_model_week_9.joblib	8 минут назад
 Untitled.ipynb	24 минуты назад

На данный момент идет работа и проверка гипотез