Applying Analytical Patterns

Day 1 Lab
Data Engineering Design Patterns at Meta - Growth Accounting

Transcript:

52:44
the end of the day one lecture if you're taking this class for credit make sure to switch to the other tab so that you
52:50
get credit for it excited you check out the lab first off we're going to be creating this ddl here I'll explain how
52:58
this ddl like kind of works we're just going to step through this first
53:03
so um and we're actually going to be using this same table for their Survivor analysis as well and I think that will
53:11
be uh very useful um you'll see here we have user grow users growth accounting
53:18
and then uh we have user ID obviously then we have uh first active date so
53:23
this is uh the first day that that user showed uh this might not like this is where
53:30
cumulative table design can kind of mess up where like they like if you start the
53:35
accumulation on the wrong day if you actually don't start at the beginning of of History then you might uh end up
53:42
saying a user's first active date is the wrong day and they were actually active much earlier but like you just didn't
53:48
start the accumulation and that's fine that's like one of those things that like essentially that uh goes away over
53:54
time and then uh you have last active date is going to be the last day they
53:59
were active and then you have daily active State weekly active state so these are going to be our um values that
54:07
are going to be uh like growth accounting you remember those like churned resurrected those kind of values
54:13

and then dates active and date so dates active is we're just we're keeping this around this is very similar to remember
54:20
in the week two when we did the dates active stuff we're going to be doing a very similar thing in this case so can
54:27
go ahead and uh run that query that's fine so what we're going to say here is we're going to say with yesterday as and
54:34
then in this case we're going to say uh select star from users growth accounting and then we're going to say where date
54:40
is equal to and we're going to start uh like I want to just start with 2023
54:46
0228 and then so what we're going to do is we're essentially going to build uh out growth accounting for the month of
54:51
March cuz that's when I quit my job and that's like when the the the data is going to be the most interesting so when
54:58
I have quit my job launch the company all that kind of stuff so then uh then we have today so we want to start with
55:04
March uh 1st and we're going to be building this off of the events table so
55:10
in this case what we're going to say is we're going to say uh select from events where uh there's yeah want to use
55:19
date trunk here date trunk uh day then we have event time and you want to cast
55:27
this to a timestamp and then this is equal to date 2023
55:33
0301 and then in here what we want to do is we want to look at essentially
55:40
user ID and then we want to have count
55:45
one and this will be group by user ID and this will give us essentially all
55:51
the users for that day and we can also have uh we can also put like a we can
55:58
maybe group on both of these because we want to have the time we want want this is like today
56:05
date so we can have uh both of those so what we're going to start with here is
56:12

this hopefully looks very similar to some of the other cumulative table designs that we were working on before
56:19
and you know yesterday is going to have no data or wait hold up I need to do a quick delete because I um my my table
56:26
will actually have data run this real
56:32
quick okay so now that's that's gone and that'll be deleted so now we have our
56:37
yesterday and our today very similar to our cumulative table before so we can say select star from today T um full
56:45
outer join yesterday Y and then in this case we have on on t.
56:53
user ID equals y. user ID so that makes sense so far and so what
57:01
we want to do is uh we want to do the the first thing we
57:06
want to do right is the cols here of like it'll be t. user ID y. user
57:12
ID um as user ID because we want to get this to match right we want we want to
57:17
let me grab this guy again so we can and then I'm just going to copy and we're
57:22
going to comment him out though so we can just have him for reference okay so we have user ID and then we have
57:30
first active date so in this case we have two values here right so in this
57:35
case we have um y. first active date and t. today date as first activ date and
57:43
this is U the the order of these coalesces matters and I'm just going to uh I think you'll understand what I mean
57:50
by this and so
57:56
okay so this will make oh this is going to make sense here so for this first one
58:02
here we have first active date so essentially how it works is if yesterday has a value great we're going to pick
58:09
that if not we're going to pick today's because that means it's a new user and they don't have a first active date

58:15
that's earlier because they're new um otherwise we can pick whatever the one from yesterday was last active date is
58:23
the opposite where we say okay if they were active today then that means that's the latest date that they were active
58:30
and if they're not active today if today is null then we want to pick whatever it was the um the last time they were
58:37
active so you see how like The Ordering of the cols here makes a big difference
58:43
um then we have a daily active State okay so this is going to be a case
58:49
when statement um I'm just going to uh essentially put we're going to put this as uh daily
58:56
active State and this will and then we have weekly active State as well right
59:03
weekly active State let's going to put some case ones in here I want to finish out the rest of this stuff real quick so this date's active I'm just going to
59:09
pull from the query real quick because like that doesn't matter that much because we did that last time that's
59:14
going to be I'm just going to grab both of these real quick and uh just kind of copy these from the query below from
59:22
okay so this is essentially just going to build out that array of date and then this is going to essentially
59:29
give us the dates that we're looking for so now this is the this is the meat of the query that I want to go over with
59:35
y'all is the these case when statements here so let's talk about the very first
59:41
case which is going to be when uh yesterday is null and today is not null
59:49
that is the easiest case so we're going to say when y. user ID is null uh and uh t. user ID is not null
59:59
then new uh because that means that they
1:00:04
showed up today they were active today but they uh like and they weren't they
1:00:09
weren't in the data set before so that one that one's that one's pretty easy so

1:00:15
H the they get more complicated as we go on here so now let's talk about uh what
1:00:21
about if they are um resurrected then in that case we need to see that their
1:00:28
active date is more than one day uh like their last activate has to
1:00:35
be it has to have a bigger difference than one day uh between uh today's date
1:00:42
and their last active date so if we say like y. last active date and then in
1:00:47
this case is going to be less than um T do uh in this case we're going to say
1:00:53
today8 minus interval uh one day
1:00:59
so um then this is resurrected because what that means is
1:01:05
there was a gap right uh that's going to be um or we have
1:01:11
retained as well so let's put retained in first because I think retained uh makes more sense where we say okay if
1:01:18
last active date is equal to today date minus one then it's retained
1:01:26
um and then we have uh so we're getting there right we we have new we have
1:01:31
retained we have resurrected so we need a churn turned is going to be uh one and
1:01:38
then we also have stale those are going to be the two that I think are going to be important here so churned is going to
1:01:45
be so in that case we're going to say when t. today date is null so this is uh
1:01:52
this one's this one's a little bit trickier because uh um we we don't have uh today date so
1:02:01
because it's null because there's no there's no daily data for that day uh but we have and then we can say and so
1:02:08
this is going to be his y. we're going need to look at last active date and this is going to be is equal to and then
1:02:16
there's actually y do but we have y.d dat so y. date is like the partition
1:02:22
date and not the last active date so we can use this essentially instead to see

1:02:28
like what's going on here so in this case we can say okay if today is null
1:02:34
and uh the last active date is equal to uh the date then this is going to be
1:02:41
uh um what should I say here this is going to be
1:02:46
churn and then uh the only other thing that's possible is stale and then let's make
1:02:53
this a like an empty string for now so I'm going to run this query and it's
1:02:59
going to be like very okay oh yeah we got a cast we got to cast these got to
1:03:05
cast this as a text because um the the types are
1:03:12
weird okay there we go so now we have our um we have our user ID we have our
1:03:18
first active date we have our last active date oh w we have user IDs that are null oh
1:03:24
great uh what's actually um filter those guys out we say and user ID is not null
1:03:30
that's like weird okay there we go so you'll see
1:03:36
okay you'll see essentially everyone here is going to be daily
1:03:42
active like always you'll see like or the daily activate is going to be new
1:03:47
but that's because we haven't started the cumulation yet so essentially what this means is we're saying this is the
1:03:53
beginning of time and every user that's active on this day that's their first day and some of these users might have
1:03:59
been active before they and so that's where like when you pick uh where you
1:04:05
start the cumulation matters a lot and but I'm not going to do that because I don't want to do like a 100 or 200
1:04:13
because Tech Creator I guess my platform if we wanted to actually do this the right way we would need to run this uh
1:04:20
starting in 2019 and I would need to run this query like a thousand times to have it be absolutely perfect but I'm not
1:04:28

going to do that because we have 30 minutes so we're just going to assume that everyone who uh started on uh March
1:04:36
1st that they are new users even if they're not actually new so in this case
1:04:42
we only need one more thing here right we're going to need a insert into uh users growth accounting so one of the
1:04:49
best things about growth accounting right is this first statement is easy
1:04:56
it's the same that one's easy okay so let's go ahead and grab these two and I
1:05:02
think this will also be not too crazy so for
1:05:08
retained in this case we essentially like it's not an equal to for retained
1:05:14
because essentially what it means is that we want to see if they were active at any point in the last seven days so
1:05:21
in this case we essentially want to say if uh instead of equal to here we're
1:05:27
going to say greater than or equal to and then instead of today
1:05:33
date um we want this to be y. dat right
1:05:39
because the reason why this needs to be y do date instead of today date is because they might not be active today
1:05:45
and so we want to use uh y. dat and that will be our uh comparison value and then
1:05:53
but this is going to be minus 7 so this will give us our retained um but
1:05:59
the good news is is uh resurrected is going to be it we do use today date for
1:06:06
resurrected so let's actually put this one above because I think that makes more sense so for uh resurrected we have
1:06:13
last active date and then so that means that they have to have been active or
1:06:18
the last time they were active needs to be more than seven days ago because they needed to have churned so that means
1:06:25
that this pretty straightforward right so we just say Okay their last active date was um s days ago or more and then

1:06:32
that means that they were resurrected and then it has to be more this is more than seven days because this is a strictly less than right and
1:06:40
then uh if if it wasn't then they're retained then we have uh the the ones
1:06:47
that get a little bit more complicated here all right so then we have a churn churn is going to be next right and
1:06:54
because we have the other ones are already done here so in the case of churn we're going to want to do
1:07:02
uh okay so we have uh when y do last
1:07:07
active date is uh okay this is going to be a a less
1:07:14
than um let me think about this because hold up I have I have that one I
1:07:21
I I this one I like this this is where like I I spent a little bit of time on this so let me let me just grab this oh
1:07:28
yeah this is where um for that's how it's done that that makes sense so for
1:07:34
um for churn you actually have to have uh why is that like oh there's a that's
1:07:42
way okay so churn you actually have this to have the um the T do today date is
1:07:48
null because you have to have that check because they can't be active today and then the the last time they were active
1:07:55
needs to be longer than this many days ago and then but it needs to be greater
1:08:02
than wait a minute I think actually this is not quite right because this is going to catch this isn't going to catch stale
1:08:10
because um it needs to be okay the last active date was more than seven days
1:08:16
ago no it has to be equal this is actually a this is an equal sign right because that means it has to be exactly
1:08:23
seven days ago because that means that uh they're churning today and then uh
1:08:29
then yeah that makes sense to me like because that because that means that like they were uh and then otherwise

1:08:35
they're stale that's a this is actually an equal sign here because that's that one moment in time
1:08:42
and so uh and then we have an Els here and we have stale so now this is our weekly active
1:08:50
State obviously for the first run everything is new so and even and the thing that is is crazy for weekly active
1:08:58
is that for the first um uh for the first seven days is that uh wait a minute oh
1:09:08
that's actually another thing we want to be careful about here because uh resurrected and retained here oh
1:09:14
retained does actually have to go above or there needs to be another condition and the reason for that is because uh
1:09:21
retained and uh oh wait no no no it doesn't matter because this today date
1:09:26
is going to be uh well yeah it actually doesn't matter never mind I I I was I thought my my
1:09:34
date logic there was a little bit off and the conditions needed to be cascaded away no but this is fine so those
1:09:39
actually do not overlap so I'm pretty sure this is going to work so essentially you see how like you have like this like kind of crazy like date
1:09:45
math that you need to do um uh this is actually not the way that they do it at
1:09:51
Facebook they actually uh the way they do it is with bit math so you know like the date the date list in and the bit
1:09:58
count and all those things where they shift the bits over if you have the date list for yesterday and the date list for
1:10:04
today and how they shifted you have like what the value is today and then you have the so you have the two date lists
1:10:10
and then you can compare the bits on either side you could actually get all of these different states a lot easier
1:10:16
like it's like the more efficient way to do it I wanted to show how you can do things a bunch of different ways though
1:10:21

because like that doesn't matter quite as much in these cases so um what we're going to do is we are going to uh we're

1:10:29

going to run this and uh make sure uh in the code that I gave you guys uh for for churn

1:10:37

this is uh this was like a less than make sure you change that to an equal that was a mistake um okay so that ran

1:10:43

and so now what we want to do is we want to just essentially change this to uh move everything up a date so it's the

1:10:51

same sort of thing where you change the two dates and then you run the run the query again okay so let's just go ahead and

1:10:59

query this real quick so we can say like select star from uh users growth

1:11:05

accounting where date is equal to date uh 2023

1:11:11

0302 we have the second okay so you'll see

1:11:17

here so you see this user here is and this it makes sense right so he's uh

1:11:22

retained because he was active on both days daily retained and so um let's find

1:11:29

a uh there's going to be some users here who are not retained though let's say um and daily active State equals churned

1:11:36

because there's going to be people who like for daily active right there we go so you'll see like these are people who

1:11:42

were just active on the first but they were not active on the second but that doesn't mean that they're not weekly

1:11:47

active because they they're still weekly active for the next seven days and so they're still weekly active but now they

1:11:54

have churned for the day daily active so you'll see how now like we can see these

1:11:59

two different ways that these these turns so you'll see daily active actually the state here will change a

1:12:04

lot more frequently because it needs to be tested essentially on a daily basis whereas uh the weekly one gets tested

1:12:11

more on like a you know a weekly basis so it doesn't change quite as quickly so like essentially you're going to get a
1:12:17
lot more retains for the first like uh seven days right of the of the weekly
1:12:23
state so let's go ahead and just quickly uh run this a bunch more times actually
1:12:30
I don't I realize I don't need to do that I can just click the Run yeah run the insert into then we can do a
1:12:38
three and four and four and
1:12:48
five and five and six
1:12:56
six and [Music]
1:13:04
seven and then let's do seven and eight and then we'll just go do one more well
1:13:09
let's go to the eighth because the eth will be where we we start to see a drop off on the um or we'll do the ninth so
1:13:16
that we can guaranteed to see a drop off here okay cool so now we have data all the way through to the ninth and let's
1:13:23
go ahead and look at that let's look at the the ninth here so we should see some
1:13:28
some users who have like dropped off okay there we go so you see we have some like stale users and then we have like
1:13:34
oh look this guy he's retained right because he came back right or oh he's just this person's like active every
1:13:40
single freaking day that's cool but um you'll see that uh there are some people
1:13:45
who ched here because they were they were they were active on the eth but not on the ninth so but well let's uh let's
1:13:53
do a little group by here cuz maybe is there a bug I don't think there's a bug but it just feels like there's an awful
1:14:00
lot of retains there oh it's still just new and
1:14:06
retained okay well I guess there is a bug here for this is
1:14:12
uh where last active date is greater than or equal

1:14:18
to y. date minus 7 days is
1:14:23
retained that's seems right though because that means that they weren't act like how is it like they are
1:14:31
all retained though hold up because this minus
1:14:38
interval okay in here we have retained
1:14:43
is it's saying it's why active dat is and it's saying it has to be but this
1:14:50
is with today date but this today date is null it can it can be null and that's
1:14:56
fine um interesting interesting
1:15:05
because okay so you have the retained and then and the the and today day is null
1:15:13
that's how because I got this to work I definitely got this to work this is wild that this is like a little bit off just
1:15:19
a little bit off that's what like one of those things about these queries that's like wild is that they um they can be
1:15:24
just slightly off where um
1:15:30
so because if you subtract seven from this and then the date is the last
1:15:36
activate is greater than or equal to that minus 7 because okay hold up because if we if
1:15:43
we query this and we just say select Star right and look at the
1:15:49
data here because we should have like some data here that is a little bit off right because let's see um
1:15:57
is first okay and then we can say uh and last active date equals date 2023 0301
1:16:04
so we're just going to do a little bit of troubleshooting here real quick see what's going on so you see okay there are users in here that we have that were
1:16:11
uh essentially only active on the first see and then this is weird because
1:16:17
the why is it getting hit like why is why is this retained that doesn't make any sense because this is uh this would

1:16:26
be where last active date is um greater than or equal to date
1:16:31
minus 7 which would mean that the first is greater than the second which
1:16:38
is you're saying for for here where uh yeah because well
1:16:45
like these these should all be still because these are not active these have only been active on the first right and
1:16:50
this is now the ninth right this is this is for the ninth and that's what I'm saying is that like this y. date here is
1:16:57
the 9th like that's okay that's wild okay um anyways there's there is a bug
1:17:02
here that I think uh what I'm what I'm going to do here because I want to give time for the rest of the lab here is
1:17:08
there's a small bug on the kind of the way that this is working because essentially there this is saying this is
1:17:15
this is like because wait a minute because somehow
1:17:21
this is this this condition is being triggered every single time like that's so weird but um what I'm gonna do is
1:17:27
like I want to put a a cap on this because the whole idea here is I know that the daily active state is working
1:17:34
and that's what matters for the rest of the class and we can see what's going on there's there must be a small bug with
1:17:40
these conditions here but the uh the daily active state is working the way we expect because it's simpler and uh the
1:17:46
idea here right is now you can kind of see like if we go ahead and we say date
1:17:52
and then we say daily active state and we say count one and then we can say uh
1:17:58
we can say Group by uh date weekly or and then daily active State we run this
1:18:04
query you can see uh okay we can see when people showed up you'll see and
1:18:11

when people okay so stale is like uh all the people who like churned out and they never came back but okay you'll see on
1:18:18
uh there was 3,000 new users who showed up on the thir right and it it was
1:18:24
really hard for me to retain daily active users essentially so there was 287 people who came back on the 4th
1:18:32
right and like some people ended up coming back like after churning a little bit but like for the most part they uh
1:18:40
like are mostly stale now anyways um that's kind of the idea for uh growth
1:18:45
accounting here now what I want to show y'all is how you can use this with first
1:18:52
active state to show y'all how to look at retention analysis for cohorts so in
1:19:01
this case what I want to do is first we're just going to say select star from users growth accounting and we're going
1:19:07
to say where first active date equals date and we're going to use 2023 0301 so
1:19:12
we're going to look at this group of users so you'll see we have a weird
1:19:20
number of users there's like hundreds and hundreds and hundreds of users here that uh essentially were here now what
1:19:25
we can do is let's go ahead and look at this uh over time so if we look at uh
1:19:32
date and count you'll see if we group here Group by
1:19:40
date you see how it's always the same it's always
1:19:46
171 which is like kind of what we would expect um so if we have 100 so this is a
1:19:54
cohort of 171 users that's the total number of users in this cohort um but
1:20:01
what we can see is uh we can say what if
1:20:06
we say count uh case when daily active State equals uh or we can say da active
1:20:13
state in and we want to say retained
1:20:19
resurrected or or new uh then one end

1:20:25
so let's look at this I think this will uh kind of give a pretty good idea here
1:20:31
so if we uh sort on date here you'll see this
1:20:37
is essentially that J curve that I was talking about so we can it's it's not
1:20:44
too crazy to build this out actually like especially if you already do the growth accounting piece of it so what we
1:20:50
can do right is we have our counts and then let's go ahead and say this is um
1:20:56
as number active and then what we want to do is we want to have one more here let's do a
1:21:04
percent right so we have this divided by count one we got to cast this though
1:21:10
otherwise it's going to give uh this is um
1:21:19
cast cast this is as real
1:21:25
what why is it mad aggregation calls are not allowed
1:21:33
here why not that this query looks like that like okay one second this looks
1:21:38
fine to me oh this it doesn't like the oh it's
1:21:44
because there's oh oh I think I missed a oh that looks right count case when cast
1:21:50
is real divide but but but you can't
1:21:57
divide hold up does it will it give it to me this way
1:22:03
great case when in okay fine we'll just we only care
1:22:10
about the percent anyways so we can move him and we'll just divide here because it's probably going to okay there must
1:22:16
have been some I must have uh done something weird but this looks like it's working okay so now this is going to
1:22:23
work and you'll see this is oh let's say this is percent
1:22:28
active Okay okay so we have our percent active
1:22:35

and then uh one of the things that's important here though is we actually
1:22:41
don't like this is working great for one cofort but this is not that good for uh
1:22:49
like all the cohorts so what we can do right is we want to say
1:22:55
date minus first active date as days since sign up or days since um we'll say
1:23:02
days since first active we'll sa that and you'll see if we do oh we got to then we got a group on
1:23:10
that so this is going to give us that kind of Survivor analysis a little bit
1:23:16
so you'll see okay uh the the first day uh they are great right it's exactly
1:23:23
one and then as things go on like things uh get a little bit different so one of
1:23:30
the things that's really cool though is now we can actually greatly expand the cohorts here because we can actually
1:23:37
remove this filter condition and now this is actually going to give us okay so now we have a a bunch
1:23:45
of users who some of them like some cohorts are bigger than others because we've had more data right and you'll see
1:23:52
but like one of the things things you'll notice is oh it's sort on uh days since
1:23:59
active here so you'll see uh days since first active we have
1:24:05
uh 100% of people which is exactly what we're
1:24:10
expecting you you remember that chart where it's like a curve and it all starts at 100% And then it slowly goes
1:24:17
down and so you'll see that like uh really we only like these last two days
1:24:24
don't want to care about as much because there's not enough data for these two days uh or because we we need to load
1:24:31
more data in because you see the number of users here is not very high but for
1:24:36
these other days we have a lot of users so you'll see that like okay for uh
1:24:42
first day it looks like it goes 100% then 5% and then it then it kind of
1:24:47

stabilizes at about 2% so it seems like my retention rate for my website is
1:24:53
about 2% of people come back which I don't know that's like it's it's
1:24:59
probably better now I mean this is obviously like six months ago so things have changed quite a bit but this is the
1:25:05
idea behind how to uh how to do this right and um like for
1:25:13
example like let me just show you how powerful this actually is though right so isn't there like there's a way to do
1:25:20
this where uh can we do like um it's got to be a way to do like a month
1:25:26
right or gota be a way to like so you can add other columns into the table
1:25:32
right or like you can group on other dimensions of the cohort or of the users in that cohort so that you can see like
1:25:39
oh do users who sign up on Wednesday versus um um Monday or whatever they
1:25:46
have uh isn't there like date part okay so um extract day of a week
1:25:51
and then this is going to be from um first active date as day of
1:25:57
week and then we we group on that as well and you'll see in this case this is
1:26:04
going to give us a different perspective so still first day looking great right
1:26:11
and then you can kind of see okay it looks like looks like what
1:26:17
Wednesday look like Wednesday Wednesday is looking pretty fire 133% but it's also like not enough data in summer as
1:26:24
well right so you can see that you can see uh what day of the week like you can
1:26:30
kind of compare these like day of the week like you can see how this could be like a nice line chart of like oh this day of the week is better than that day
1:26:36
of the week and then then you can kind of use that to your advantage to like uh
1:26:42
optimize your ad campaigns stuff like that uh so just letting youall know I don't think we're going to get have time
1:26:48

for the uh kind of the Rolling uh window function stuff cuz we only have like four or five minutes left for to today's

1:26:54

uh lab I was a little bit ambitious on uh how much content to cover here I covered a little bit in too much in the

1:27:01

presentation but um I think that's pretty much it for what I was planning

1:27:07

to cover today congrats on getting to the end of the day one lab if you're taking this class for credit make sure to switch over to the other tab so you

1:27:13

can get credit for the day two