Data Visualization and Impact

Day 1 Lecture

Transcript:

0:00
data Engineers that don't actually impact decision-making are just running up Cloud bills and moving bits around in
0:05
the cloud without actually having any tangible impact in this 2-hour course we're going to be covering all of the
0:11
things necessary to land an impact as a data professional whether that be pushing back with stakeholders on low
0:17
value requests because the whole can I get this data real quick meme is a meme for a reason and data engineers get
0:24
burnt out from that over time and so how do you push back there so that you can have more time to do robust data
0:30
modeling that actually impacts decisions not just for this quarter or next quarter but for the next four or five
0:35
quarters and along with this stuff we know that another layer in Impact is communication and visualization if you
0:42
can display your data in a very compelling way that is a very powerful way to impact decision-making processes
0:48
I'm excited for you to check out this course this is the last of the courses in the 6 we free data engineering boot
0:54
camp so I hope you enjoy it [Music]

Types of Impact

1:02
the types of impact um you got measurable hard hard to measure and
1:07
immeasurable so if you think about measurable it's you made a pipeline more
1:12
efficient congratulations or you enabled an experiment to run that moved a metric
1:19

that's another easy way to have measurable impact it's like you provided metrics that moved uh that enabled an
1:27
experiment to run and that experiment caused this these metrics to move so you can claim some of that impact usually
1:33
you need to claim that impact in collaboration with uh data scientists
1:40
and product managers and stuff like that you got to do that like um it's not like it's just yours unless you're like a
1:47
data unicorn and you're doing all of that and I'm hoping that most of y'all will have more of those skills kind of those experimentation skills as well as
1:54
you you kind of learned a little bit in the boot camp uh this time uh then you have hard to measure
2:00
um Quality improvements um for example uh if you are if you don't have any data
2:07
quality errors like you one of the things that's hard about measuring errors is it's like nothing going wrong
2:14
is the measurement the measurement is zero the measurement is like no problems and obviously you can measure those
2:20
things you can measure like how many errors are showing up and all that stuff but like and that's why it's hard to measure because it's not necessarily
2:27
that like you can't measure it it's just that like is is it worth your time to actually measure it like and in my
2:33
experience like most of the time in big Tech the answer to that question is no like they don't really keep track of like all of the quality errors that
2:39
happen and how many are happening on each team over time uh so that can and because the thing is is like the number
2:45
of quality errors is not necessarily uh indicative of the quality because you could also have more
2:51
pipelines it could be indicative of the number of pipelines or it can be indic indicative of the amount of data quality
2:58
checks that you're doing so so uh like more quality airs might actually mean
3:03
higher quality because you're actually catching more bugs so that's where like

3:08
that measurement is a very hard measurement to figure out and um that's a good example of uh something that
3:15
matters I mean I'd say data quality matters uh but it's also something that
3:21
is tricky um Team enablement outcomes that's another one so I'm just going to give a quick example here about team
3:28
enablement things something that I did at airb I me at Netflix that um made my
3:33
life a lot easier was I wrote this groovy script thing that what it did was
3:38
it made the the amount of data that I had to ship to run my spark job a lot
3:44
less it like cut it by like 60% so then um instead of taking four minutes to
3:49
deploy it took one minute to to deploy and I was able to get that code adopted
3:54
across the team and so we all saved some time but it's hard to know how much time we actually saved I mean the only way
4:00
we'd be able to do that is measure like how many times do each person on the team deploy and then multiply that across the board and obviously that's
4:07
like probably some amount of time like and I I tried to get a kind of a back of the envelope calculation there but it's
4:13
not something that you have like the hard numbers for and or would I ever like create a data pipeline about right
4:20
I would never actually like spend the time to actually go and measure the impact of that Improvement because the
4:27
value of measuring that is not worth it like and it's the back of the envelope calculation is probably good enough to
4:34
give you a good enough measurement but like it's obviously not very accurate so but it's still impactful and it's
4:40
something that can uh definitely save a lot of headache for my team but whether
4:45
it's good or bad I or whether it's measurable or not is like and how accurate that measure is makes a big

4:52
difference um and then obviously you have the third type which is uh immeasurable impact um the big one here
5:00
is uh changing intuitions where if you present data to someone and then like if
5:05
you present data to a leader and then it changes their intuition on how to make decisions that can be one of the most
5:11
impactful things you can do because it makes it so that their uh default thinking is more correct but that um is
5:20
very hard to measure like the only way you could measure that is if you had leadership take some sort of like bias
5:25
test and like they're not going to do that and so uh changing intuition is a big one team culture improvements like
5:30
if you're a good team player and you're very enjoyable to work with and uh all that stuff that's another way to um have
5:39
a good impact but like it's one of those things that is uh again you could almost
5:45
put this in the hard to measure because you could say like okay it's a measurement of retention but then you have like Team Dynamics where it's like
5:51
you might be the best team player ever but like if there's another person on your team who's toxic or your boss is
5:56
toxic then like you might be adding to the culture but then the other people on your team are removing from the culture
6:02
and it's like is the fact that there's massive uh um you know ATT attrition on
6:08
your team is that because of you even if you are the person who is the you know the person who's actually contributing a
6:15
lot to the culture so I put it in the immeasurable immeasurable bucket there because Team Dynamics are very
6:20
complicated but that being said that doesn't necessarily mean that uh you shouldn't be a good team player because
6:28

it is something that matters uh the last one here is called being a glue person I found being a glue person
6:33
to be a very important piece of this puzzle uh what I mean by Glue person is you're kind of the person that holds the
6:39
team together and you uh you do a lot of the work that is not like as shiny or as
6:46
sexy as like a lot of the other work like for example one of the things I did to be a glue person um at Netflix was I
6:54
combined all of our teams code into one repo and then I showed everyone how to use this like single single repo and
7:00
that was a great way that I got everyone on the same page um to uh for code sharing that was cool um I found that to
7:08
be a really fun and great time uh but like at the same time it's like what was that what was the value there the value
7:15
there was mostly in the collaboration model not necessarily in like oh I Sav Netflix and number of dollars because
7:21
now we are all using one code base it's like hard very hard to measure very hard to measure like what the difference
7:27
would be if we use 10 code bases and like how many much time we would save it's a very hard one you you know you
7:32
could see that there's impact there but you don't know what it is like you don't you don't know exactly like how many dollars it is or how much money they're
7:39
going to save or how much time they're going to save you don't really know but you know that there's impact there and so and that's why being a glue person is
7:45
super important because like if if teams don't have glue people they kind of fall apart and then you just have a bunch of
7:51
uh you have a bunch of like really talented um lone wolves who don't support each other and that can be very
7:58
terrible and I've found that like the glue people are the people who get promoted to leadership the glue people
8:03

are the people who actually end up uh becoming managers and leaders because they are very good at banding people
8:10
together whereas the lone wolves like sometimes they they can be good too they just like are more like I think they're
8:16
more likely to become very technical ic's because they could be kind of a lone wolf in those cases and kind of build up that way but even then um It's
8:23
Tricky for uh if you're working in that Lone Wolf mindset to uh get promoted
8:29
Beyond like senior engineer because of the fact that if you're at staff or higher up on
8:35
the ladder then you also need to lead and band people together and kind of
8:40
sell people on a technical Vision so like it's very hard to work alone and
8:46
become a leader uh that's just kind of I hope that makes sense like the the leader who is alone that like sounds
8:53
like an oxymoron and it kind of is and that's where like there's a lot of value in being a glue person so I I know I've
8:59
just harped on that a little bit but yeah let's let's keep going okay

How does your time as a data engineer translate into value

9:06
so how does your time as a data engineer translate into value well there's a bunch of bunch of
9:13
things here uh I think the the number one way that people like as a data engineer like if you if if you um you
9:20
know if your grandma asks you so what do you do you like say oh I Supply information to the business I think
9:26
that's like what most people would say is the um kind of at a very fundamental
9:33
level that's what data Engineers do these Supply insights um I think that
9:39
that is one of many things that they actually do and uh there is going to be
9:45
a lot more that they can also do as well that like it's not just uh supplying insights that like there's so many other
9:52

tasks and ways that data Engineers can have impact that is super important um
9:59
obviously you have preventing bugs and maintaining Data Trust because if you have if you're in an organization that
10:05
has low quality data then people are just going to use their intuition more and their intuition is oftentimes a lot
10:13
more likely to be wrong than right and that makes uh the business it's riskier
10:19
for the business because data is data is also wrong a lot of the time like I mean
10:24
but like it's wrong less of the time than your intuition is and that's the whole point and why we use data and why
10:31
we like data um also like pipelines aren't free you need pipelines that are
10:36
efficient you got to make you got you can save a lot of money if you make your pipelines efficient and you either save
10:43
a lot of money or you save a lot of time because like um if there's an air if you have a very efficient pipeline that runs
10:48
really quickly then uh you can uh bounce back from a data Quality Air a lot
10:54
faster like for example when I was working at Airbnb one of the things I did was I added staging step to the
11:00
pricing and availability Pipeline and uh the old pipeline if you wanted to backfill all of history it took like two
11:07
weeks and uh with the staging step it went from taking two weeks to taking one day and so um and since it only took one
11:15
day like uh like we were able to iterate way faster on the the definitions of
11:20
pricing and availability at Airbnb and I also didn't feel so nervous about making
11:26
mistakes because my pipeline was a lot more efficient and so like I didn't have to like eat
11:31
all of that backfill pain of two weeks of like babysitting in my Pipeline and hoping that it doesn't break by just
11:38

making the pipeline more efficient so that's another great way that you can have a lot of good impact and I know
11:44
that the next data engineer he's going to be very happy with that whoever took over my job at Airbnb um you also have
11:50
uh enabling other Engineers to work faster and this can be in a couple different ways uh one of them is um
11:57
through processes like code review um if you provide code review that helps
12:04
people adopt best practices um more consistently then that's a great way to
12:11
work faster because then you're just it goes back to point two here of preventing bugs and maintaining data
12:17
Data Trust right because if if you use best practices you're going to prevent bugs and maintain Data Trust and that
12:23
and if you prevent bugs you're going to work faster because you don't have all the maintenance overhead that you would normally have from Bad data quality so
12:32
um that's going to be another great way to think about it but there's all sorts of other ways there's so many other ways
12:38
that you can um enable other Engineers to work faster um I want to go over a couple more here I so I talked about
12:44
that thing that made it so I could deploy my pipelines faster um another thing I worked on um at Facebook was I
12:51
worked on this framework called Milky Way and Milky Way allowed you to do all
12:56
sorts of um analytical patterns um essentially for free so how it worked was you described the schema of your
13:03
input table and then you described the analytical pattern that you wanted to apply whether that be um and we and
13:10
almost all the analytical patterns uh that were available in Milky Way we have covered in this boot camp uh one of the
13:16
analytical patterns was cumulation like cumulative table table design another one was uh growth accounting or that

13:23
kind of state change tracking another one was uh retention that kind of retention tracking so that's one of the
13:29
the reasons why I wanted to cover all that stuff because I found those analytical patterns to be heavily used at that Facebook and but we built a
13:35
framework that made it so you didn't have to write the SQL for that you just had to describe the input schema and then that was it and then everything
13:42
else would be done for you and so that enabled data Engineers to work a lot faster it was like almost like the chat
13:47
GPT of 2017 uh where we were able to just be like yep apply this pattern with this
13:52
schema take this schema and this pattern and then you're then you have your output data set and so uh that was
13:58
really fun but that's another example where you can build Frameworks and libraries that enable other Engineers to
14:04
work faster and better and with fewer airs like uh you could think of like another example here is like if you have
14:12
um Engineers like like for example Max Max bman he worked at um Airbnb and he
14:20
created airf flow and he um open SCE to airf flow as a technology because he was
14:25
like the orchestration engines that we have right now Uzi Uzi is terrible I don't want to write my freaking my etls
14:32
with XML XML is terrible and like it it's painful and like I want to be able to use loops and expressive um stuff and
14:40
that's where you get a lot of that with um airf flow and that's another way that
14:45
uh Max enabled people to be a lot more effective and obviously that has been
14:51
you can have that be a very high level if you think about like Max's impact there it wasn't just at his own company
14:58
because air flows used like across the entire industry and so if if you want to really be at that next level that's what
15:03

you want to do right um that's going to be the next one right if if you are building things that make it so other
15:10
Engineers can work faster that's going to be some of the most impactful and important work of your career um last
15:16
one here I think is important as well which is around kind of like shifting culture and shifting um decision-making
15:23
processes because a lot of times um companies like if you're not in big Tech and you're other companies like they
15:29
might not be as uh data driven as big Tech is and so you want to also spend
15:37
some time to not just like show like hey here's this high quality data but also
15:43
be like yo this data is like you should be making decisions with this data and if you're not like you're you're Flying Blind and you're taking on unnecessary
15:50
risk to the business that you don't that you don't need so anyways those are going to be like the five ways that you
15:56
can essentially change your time for impact and time for value and the more
16:03
effective you are at each one of these buckets you're going the more you're going to uh be a better data engineer
16:11
and you're going to climb the ladder faster especially like if you can learn how to do every single one of these buckets uh effectively then you're going
16:18
to climb really quickly but we're going to dive a little bit deeper into each one of these buckets because I think it's

When is an Insight valuable

16:24
important so so um
16:30
when is an Insight uh valuable uh when you create a data set right like for
16:36
example um one of the things I did right in that experimentation uh week last week and I
16:42
did that red and blue experiment one of the things that I learned from that experiment is that really it doesn't
16:48

matter it doesn't matter if it's red or blue it really doesn't matter and so if it doesn't matter then uh that kind of
16:55
goes into that third category of it fails to support or contradict intuitions CU I predicted that the red
17:01
button was going to do a little bit better but like statistically speaking they are exactly the same so but you can
17:07
reinforce intuitions contradict intuitions or kind of like fail to reinforce or contradict intuitions you
17:13
can do all three uh and that can be a very important thing to remember is that no result is a result like especially
17:21
when you're doing experiments like if your experiment says there is no relationship between these two variables
17:27
and it when like that a lot of times like data scientists they might consider that experiment like a a failure quote
17:33
unquote failure when that's not the case like you learn something you learn that like these two things are not as
17:39
correlated as you thought that change doesn't matter that much so like when when you fail to reject the null
17:45
hypothesis like that is still a valuable thing I think data engineers and data
17:51
scientists they get into this kind of uh a rut when that happens though because
17:57
it comes back to that measurable versus immeasurable impact because when the null is rejected and there is a
18:04
statistically significant result then you can measure that result and you can say this many dollars this many users
18:12
this many revenues and that is good but like when you don't reject the null like
18:18
the the answer is like it doesn't matter like the change doesn't matter like and it has no no tangible impact and so that
18:25
is something to think about like another thing to remember here uh in data engineering is supplying
18:32
insights in general is often overvalued uh I think it can be a big part of our
18:39

job and it often is it's often is like a very very big part of the data engineering job but it's not everything
18:47
and a lot of times I think there's people out there in the world who think that it is everything especially analytical Partners analytical Partners
18:53
often times will be like hey can you just pull this this data real quick or you know that kind of meme everything
18:59
and because it's not often that like supplying insights is just quick and

Overvaluing Insights

19:04
easy okay so here are some signs that um
19:09
your organization is overvaluing insights because insights comes at the
19:15
expense like come like the the rate at which you create insights comes at a
19:21
tradeoff with the rate at which you create technical debt and they there's a
19:28
Balancing Act there where if you want to just churn out pipelines every day and like not really care about quality and
19:34
you just want to like have a bunch of data points like you can do that quickly I could I could create 50 terrible
19:40
pipelines in a month all right and I could do that and then they might have like the data might be 80% right and
19:47
like it could give us directionality even like in in the startup world that might be a a good good play because it's
19:54
all about like getting answers as quickly as possible but like you also want to remember that you want to create
20:01
your pipelines and all your stuff in a way where once you build it it can stick around for a long time so if you want to
20:07
come back to it you can do that so I'm going to talk about I have a couple anecdotes um in on this slide as well
20:14
for uh when an organization is overvaluing insights because analytical
20:20

Partners they uh when business is normal analytical partners are very patient for
20:27
the most part and and they are going to be like okay we can wait we can we can get in with your long-term planning or
20:34
your quarterly planning we can we can get in on that but that all kind of goes
20:39
out the window when something uh crazy happens so I'm going to give one example
20:44
real quick so when I worked at Facebook near the end of my time there like Facebook saw its first diping growth
20:49
ever and um that was what was considered a crisis and I essentially got pulled
20:56
off all of my normal work I was doing and and I got put onto this like Squad to troubleshoot this crisis as quickly
21:02
as possible and um I learned a lot from that process
21:09
initially I felt very excited to be a part of that because I felt very important where I was like wow I'm going to troubleshoot this very hard problem
21:16
um I learned later on as I've kind of done and as I've had more of these crisises in my career that like I don't
21:24
like I don't know I I I don't necessarily think that it's like the best thing to do because it's uh like
21:30
you end up creating a lot of pipelines that might be be low quality or like you're not taking enough time on them
21:37
because the analytical partners are not patient enough and that can be uh a problem um and so one of the things that
21:44
happened for me is like I know I churned out like so many different pipelines and measurements and stuff like that and that's when I like kind of discovered
21:50
different patterns and that's where like in these crises like if you're churning out a lot of pipelines you should be
21:56
able to notice like some higher level pattern like maybe that if you go back to like week two when we were talking about
22:02
daily metrics or weekly metrics or monthly metrics or uh slowly changing Dimensions or whatever because most of

22:09
the time when a crisis happens what analytics is looking for is the ability to slice and dice and cut the data in
22:16
every possible way that they can and if you're doing your data engineering right that usually means that you don't have
22:23
to do anything and that data is just available for them and that uh that like that's the long-term vision for data
22:30
engineering is that like you build the data infrastructure for them and then they can answer whatever questions they
22:36
want to answer and you don't have to go and write ad hoc pipeline after ad hoc Pipeline and burn yourself out and so
22:43
anyways for me that's what happened like I was writing like these ad hoc pipelines and trying to solve these problems and like I got so burnt out I
22:50
felt so tired from from working on this all the time and one of the things that I realized was that I was overvaluing
22:56
insights and I was undervaluing Tech debt and my own sanity and all sorts of
23:02
other things right and so that's where it's important to think about things in a longer term vision and that's another
23:08
thing that can happen right is in a crisis like a lot of times it's more like I need this answer tomorrow or I
23:14
need this answer yesterday and you don't have like oh I'm gonna I'm going to deliver this to you in two weeks or
23:20
three weeks and then like it's going to be like uh there's like a vision for
23:26
what you're going to be doing for the next couple days or the next week or two and you get this kind of like I need to
23:31
Sprint the marathon sort of thing I think we talked about a that a little bit earlier last week but um and then
23:38
the last one is ad hoc request hell where you just have so many requests coming in and uh you don't know which
23:45

ones you need to actually uh fulfill and which ones are not worth it and which
23:51
ones like should be passed or delegated or like even like removed from your task
23:56
list so a lot of that like ad requests like hopefully that's something that you can work with your manager on and you
24:02
can get uh a good kind of way of like getting the stakeholder to do The
24:08
Upfront work where because a lot of times I've noticed with ad hoc requests if you make it so the stakeholder has to
24:15
do more work themselves to give you a request as opposed to them just being like hey can you pull this data because
24:20
that's like easy it's very easy that's a very low effort for them to make a request like if you make it so that
24:26
there's a little bit more of a barrier for them to make a request where you have to they have to fill out a dock or fill out a form or fill out fill out
24:33
something and like uh explain like why they need this data and all that stuff that stuff that can be a great way for
24:40
you to have a healthier relationship with your analytical stakeholders because then you're not just like
24:46
running around like a chicken with your head cut off and you can actually um uh
24:51
plan ahead and requests come in at a slower Pace cuz um you make because they
24:57
don't want to do it cuz like they don't want to they're only going to make a request when they really really need it in those cases because it's work for
25:04
them and that's a good thing uh it it's one of those things that I kind of went back and forth on for a while because a
25:10
part of me feels like well we should be we shouldn't slow down business communication but this is uh something
25:16
especially in data engineering that I think is kind of unique to this because
25:21
software Engineers already have this they already have a lot of this like okay you want a fe you want a new feature okay put in a request it's not

25:28
like oh you want a new feature and we're just going to give you that data tomorrow like they don't do that they
25:34
don't do that and and I think that that's one of the things that's changing in data engineering is it's becoming
25:39
more like software engineering where you can kind of have more of that long-term planning and so that's a key thing to
25:46
remember here these are the four things that can happen when you're or when it looks like your organization is overvaluing insights that is the big
25:54
thing to remember here because we're talking about value here we're talking about impact so in this cas Cas this is
25:59
the these are four signals that you can see uh that would be important to see
26:06
that your your company's probably has the bias a little bit wrong and is a little bit out of whack um there's
26:11
another example I have actually um for a crisis so I worked um at Netflix in 2019
26:17
and in 2019 that was when Disney plus came out and again that was another example of where uh people were freaking
26:24
out because they were like worried that like uh Disney plus was going to know take over and be really powerful and all
26:31
this stuff right and again we had some more like analytical like really intense analytical Sprints and stuff like that
26:36
to like solve problems and so you can have all sorts of different things that can happen in your company and remember
26:44
as well that like these crisises will pass and don't take them personally like because I I definitely felt like when I
26:49
was working at Facebook at least that like I needed to solve Facebook's growth problem like by myself or like I needed
26:55
to do everything in my Poss everything I could possibly do to solve this problem and I don't know I think that that was
27:01

like I I I definitely over indexed on like trying to solve this big problem when like I probably should have focused

27:08

more on like myself and my health and my well-being but anyways that's those are

27:14

overvaluing insights indicators so how do you push back pushing back and saying no is so important because if you just

Push Back Say No

27:21

say yes to everything as a data engineer uh you're going to get burnt out you're going to get roasted you're not going to

27:26

last very long here I mean a lot of people in data engineering don't last very long like many many people I think

27:31

a lot of people drop out I think it's like 80% of people drop out in data engineering after 5 years so uh why uh

27:39

this is a big reason right here so uh don't be afraid to say no um don't be

27:44

afraid to push back uh and tell tell your stakeholders that things can wait

27:50

and that like you're doing things as quickly as you can and while maintaining quality obviously and push back say no I

27:57

love saying no say like that's like one of my favorite things in the world is to tell tell people no that like it can't

28:02

happen and that can be a beautiful beautiful way to remember that like

28:08

you're in this career for the long term and like if you're getting all these requests that are happening on this like

28:13

weird lurchy short-term basis you're not going to stay in data engineering for the long term you're not going to at all

28:19

like and so that's a big thing um leverage your manager uh always If

28:26

your manager uh should be able to support you in these uh in saying no and

28:32

getting the prioritization right if they don't find a new manager like find someone else quit the job find a new job

28:39

because if they aren't able to like help you like maintain a good sanity and a good work life balance it's not it's not
28:45
the job for you um uh obviously another big thing here is around you want to
28:51
communicate with your stakeholders about the fact that like if we go slower we can go faster because if we can build
28:57
comp comprehensive and robust data models then all of these oneoff pipelines that they're asking for don't
29:03
even need to exist and they can just query the data and it's just available to them and that's where like if you
29:09
have robust data models like it's very very very powerful it's insanely powerful it saves your it saves your own
29:16
sanity as a data engineer if you can get these robust data models because then a lot of the analytical Partners they
29:22
don't need one-off pipelines anymore they can just go to the data and query it themselves they might need help with
29:27
like some of the query like they might not know how to do those like array functions or anything like that if you're doing like one big table but
29:33
maybe you can teach them and then they then once they know then you can you're good to go and then they won't have to
29:40
like bother you anymore and you can again focus on more robust data models and that stuff is very very powerful and
29:47
the the key takeaway from this slide is remember don't be afraid to say no and that like the odds like I think that a
29:54
lot of data Engineers have fear about that because they fear like they're going to get fired or they fear that um
30:00
that there's going to be a fight or they fear that there's going to be like I don't know some sort of push back and
30:05
that's not what happens like I've like I've generally found that like when you uh when you push back that's when you
30:14

get the most value like and because then you actually saying no to something opens up the possibility of a
30:21
conversation about priorities and opens the possibility of a conversation around
30:26
sustainable work environment and so because saying yes doesn't saying yes does nothing saying yes just gives
30:32
them their value and like if you keep doing that you're going to eventually burn yourself out so those are uh some
30:40
good antidote antidotes to uh when insights are overvalued and you're kind of treated like a sequel monkey these
30:47
are going to be uh great kind of tips that I've used in my career to uh kind
30:52
of maintain balance okay uh some of the stuff we

Maintain Trust

30:58
talked about in week three but um how do you maintain trust in data sets you produce right quality checks obviously
31:04
you got to do write audit publish pattern uh documentation write a spec just like what we were talking about
31:09
before because this trust remember trust is the impact um here in this category then you
31:16
have good engineering practices use cumulative table when you can uh minimize the amount of data that you're
31:22
processing and uh all that stuff uh clear expectations make sure you have like an SLA
31:28
so that people are aware of like what is uh when this data will arrive because that could be another way to minimize
31:34
like communication overhead because people are like oh is this data delayed it's like nope it's uh it will it's only
31:41
delayed if it's been over a day or it's only delayed if it's been over eight hours or whatever whatever your SLA is
31:48
and so um and then also document the gaps I think that this is a an important one to talk about real quick so

31:54
sometimes people might see your data set as like all powerful and that like it's
32:01
uh they put more trust into the data set than they probably should and that is
32:08
something that you don't want uh I don't know if y'all ever saw the thing was Zillow uh they like they' tried to like
32:15
predict the housing market they spent like hundreds of millions of dollars on the housing market in places and they
32:21
lost a ton of money and so that is um an example of where uh
32:28
they had data sets that they believed in but those data sets were not complete
32:34
they didn't cover all of the nuances of the housing market because that's a very complicated system to model so that's
32:42
going to be an example of a gap and hopefully you can think about like things that might not be in the data set
32:48
that could make uh the data more accurate but that's going to be um we
32:55
clear expectations um I find Data Trust uh it's interesting
33:01
because this area I think has kind of gone back and forth and it really depends on the organization whether this
33:07
is overvalued or undervalued I found at Airbnb it was like accurately valued
33:12
like it was like very fair Facebook it was undervalued where like Facebook they really did care about just like they
33:19
overvalued insights and they kind of undervalued uh Data Trust and data quality and Netflix was kind of more
33:25
like Airbnb where they kind of Fairly valued the but um generally speaking there's a lot
33:30
of companies out there that like really just want you to give them a number and they don't really care as much about
33:35
like like the sustainability maintainability or trust of that number which is weird found this so weird
33:42

that's the case like like why like well don't you want to care about like whether or not you can actually trust
33:47
this number so anyways quality is an undervalued way of um delivering impact
33:55
as a data engineer so remember that as you're kind of going throughout your career so what about efficiencies

Efficiency Quality

34:01
efficiencies and quality are related uh efficiencies is almost like a subset of quality but um we talked about a couple
34:08
of these things uh proper data modeling is a great way right uh that was weeks
34:14
one and two if you data model correctly you're going to save a lot of time and pain and frustration uh improving on call if
34:21
you're in the combine track uh like we did a lot of stuff last week on data pipeline maintenance and so that's going
34:27
to be a great place to see what you can do there um reducing data volumes this
34:33
could be like through sampling or compression techniques like leveraging Park format we're also going to be
34:39
covering a lot of Park stuff in the spark week this week uh it should be fun
34:44
um picking the right tool for the job so um I think this is an important one to
34:50
talk about real quick I have a another anecdote for y'all so when I was working at Facebook uh there was a big push to
34:56
get off of Hive because Apache Hive was like the the main Big Data technology we used because Hive is slow and expensive
35:03
and terrible and they were like okay we need to move to presto and I was like okay
35:10
and uh one of the things about Presto was is if you have more than one terabyte of data uh Presto is going to
35:17
run out of memory because Presto had like a one terabyte memory limit and I was like well my pipelines are like 10
35:24

terabytes so uh I guess I can't use Presto and
35:29
that was like kind of a crazy thing for a little while there where I was like what do I even do and I was like I want
35:35
to make my pipelines more efficient but I can't use Presto and Presto is what Facebook wants everyone to move to and
35:41
so what I did was I waited a bit and then I talked with the data platform team and I was like yo can we get spark
35:47
can we get spark on on my team because I know if I get spark I'll be able to freaking process my data way more efficiently if we move to spark and then
35:55
I got uh I got access to spark machines started migrating things saw a massive
36:02
Improvement but you see like one of the things about this is like picking the right tool for the job like I did try
36:07
Presto for a while I actually did try to use Presto for wherever I could and I was like this is dumb like this doesn't
36:14
work like and like Hive literally the older the older technology was the better choice in those cases and like I
36:20
was surprised by that because I was like I thought Hive was old and slow and terrible but like it was old and slow and terrible but it actually ran so like
36:28
that was made it better so um you never know what tool might be the right one like I definitely noticed this as well
36:34
when uh moving stuff from hive to spark is that like uh spark can actually in some cases be more prone to running out
36:41
of memory because it leverages a lot more like Ram compute and because of
36:46
that um spark actually can um fail more often than Hive so in some cases it's
36:53
not a clear trade-off it's not just like oh it's so much more efficient but it could fail more often as well so you
36:58
want to be careful when you're like migrating and upgrading and and being very aware of the trade-offs that you're
37:04

making okay so another thing to think about here is uh like simplifying model
37:10
and reducing capabilities a little bit so sometimes uh your data science
37:16
Partners like they want you to have every column Under the Sun like a lot of times it's like quote unquote like just
37:22
in case and then like you put it in there and then you you regret because
37:28
like then you look like after 6 months they've never queried it and you're like
37:34
okay why did we why did we why did we bring this column with us like why why if they've never actually used it and
37:41
it's like oh are they're going to we put this column in for the one chance that they query at one time in five years I
37:47
don't know if that's worth it I think that that's probably not worth it and so um that's where obviously there's um you
37:53
have to be aware of like the impact of these things um like for example there was an uh I have
38:01
a kind of a tra an example of like the other side right of like when that line of thinking actually doesn't work and
38:08
like that was like when I worked at Netflix and I worked in security I worked a lot on like security audit logs
38:14
and like auditing like if a security incident happened and like and in those cases like those logs are not quered
38:21
that much they they're only queried in like in the case of a security incident for the most part and so and those
38:27
didn't happen very often so in those cases like yeah it makes sense to have that data on hand or like another
38:33
example is like maybe you need to hold on to data because of legal reasons or privacy reasons or uh things like that
38:40
or um and if you need that like then that makes sense like if you're like Bound by law then that's fine I think
38:47
generally speaking uh the the way I look at it is like if it's like something related to security

38:54
legal or lawyers or privacy then uh there's going to be probably more of a
39:02
um a flexible nature of this number five bullet point but if you're in pretty
39:08
much anything else like then you really want to think about whether or not to bring in the column because if so many
39:15
times data scientists have told me like just in case and then they never query it and so uh in those cases it would
39:21
probably be better to just not have the column and there would be no uh Downstream or any sort of impact from

End of Day 1

39:28
that congrats on getting to the end of the day one lecture if you're taking this class for credit make sure to
39:34
switch to the other tab so that you get credit for it what I want y'all to do is um I actually there is a repo uh you'll