DEO I

Temelji i gradivni blokovi

POGLAVLJE 1

Opis inženjerstva podataka

Ako radite u oblasti podataka ili softvera, možda ste primetili da se inženjerstvo podataka izvuklo iz senke i sada deli scenu sa naukom o podacima. Inženjerstvo podataka je jedno od najtraženijih polja u oblasti podataka i tehnologije i to s dobrim razlogom. On postavlja temelje za nauku o podacima i analitiku u proizvodnji (produciton). Ovo poglavlje istražuje šta je inženjerstvo podataka, kako je to polje nastalo i kako se razvijalo, veštine inženjera podataka i s kim sarađuje.

Šta je inženjerstvo podataka?

Uprkos trenutnoj popularnosti inženjerstva podataka, postoje mnoge zablude o tome šta inženjerstvo podataka znači i šta inženjeri podataka rade. Inženjerstvo podataka postoji u nekom obliku otkako su kompanije počele da rade sa podacima – poput prediktivne analize, deskriptivne analitike i izveštaja – i došo je fokus zajedno sa usponom nauke o podacima tokom 2010-ih godina. Za potrebe ove knjige, ključno je definisati šta *inženjerstvo podataka* (data engineering) i *inženjer podataka* (data engineer) znače.

Prvo, pogledajmo prostor u kome se inženjerstvo podataka opisuje i da razvijemo neku terminologiju koju možemo koristiti kroz celu knjigu. Postoji beskrajno mnogo definicija *inženjerstva podataka*. Početkom 2022. godine, Google pretraživanje sa tačnim poklapanjem za „šta je inženjerstvo podataka?“ vraća preko 91.000 jedinstvenih rezultata. Pre nego što damo našu definiciju, evo nekoliko primera kako neki stručnjaci definišu inženjerstvo podataka:

> Inženjerstvo podataka je skup operacija usmerenih na kreiranje interfejsa i mehanizama za protok i pristup informacijama. Potrebni su posvećeni stručnjaci – inženjeri podataka – da održavaju podatke tako da budu dostupni i korisni za druge. Ukratko inženjeri podataka uspostavljaju i upravljaju infrastrukturom podataka organizacije, pripremajući je za dalju analizu od strane analitičara podataka i naučnika.
>
> Od „Data Engineering and Its Main Conceptsi“ od AlexSoft[1]

[1] „Data Engineering and Its Main Concepts“, AlexSoft, ažurirano 26. avgusta 2021, *https://oreil.ly/e94py*.

> Prvi tip inženjerstva podataka je fokusiran na SQL. Rad i primarna skladišta podataka su u relacionim bazama podataka. Sva obrada podataka radi se sa SQL-om ili jezikom baziranim na SQL-u. Ponekad se ova obrada podataka vrši sa ETL alatom.[2] Drugi tip inženjerstva podataka je fokusiran na Big Data (veliki podaci). Rad i primarna skladišta podataka su u Big Data tehnologijama kao što su Hadoop, Cassandra i HBase. Sva obrada podataka vrši se u Big Data okvirima kao što su MapReduce, Spark i Flink. Dok se SQL koristi, primarna obrada se vrši programskim jezicima poput Jave, Scale i Pythona.
>
> Jesse Anderson[3]

> U odnosu na prethodno postojeće uloge, polje inženjerstva podataka može se smatrati nadskupom poslovne inteligencije i skladišta podataka koji donosi više elemenata iz softverskog inženjerstva. Ova disciplina integriše specijalizaciju oko upravljanja tzv. „big data" distribuiranih sistema, zajedno sa konceptima oko proširenog Hadoop ekosistema, obrade podataka u realnom vremenu i računanja u velikom obimu.
>
> Maxime Beauchemin[4]

> Inženjerstvo podataka se svodi na kretanje, manipulaciju i upravljanje podacima.
>
> Lewis Gavin[5]

Sjajno! Sasvim je razumljivo ako ste bili zbunjeni u vezi inženjerstva podataka. To je samo mali deo definicija i sadrže ogroman raspon mišljenja o značenju *inženjerstva podataka*.

Definicija inženjerstva podataka

Kada prođemo kroz tokako različiti ljudi definišu inženjerstvo podataka, pojavljuje se očigledan obrazac: inženjer podataka dobija podatke, skladišti ih i priprema ih za upotrebu od strane naučnika podataka, analitičara i drugih. Definišemo *inženjerstvo podataka* i *inženjera podataka* na sledeći način:

> *Inženjerstvo podataka* je razvoj implementaciju i održavanje sistema i procesa koji uzimaju sirove podatke i proizvode kvalitetne, konzistentne informacije koje podržavaju dalje slučajeve upotrebe, kao što su analiza i mašinsko učenje. Inženjerstvo podataka je presečna tačka bezbednosti, upravljanja podacima, DataOpsa, arhitekture podataka, orkestracije i softverskog inženjerstva. *Inženjer podataka* upravlja životnim ciklusom inženjerstva podataka, počevši od dobijanja podataka iz izvornih sistema do serviranja podataka za slučajeve upotrebe, kao što su analiza ili mašinsko učenje.

[2] *ETL* je od *extract, transform, load*, uobičajeni obrazac u ovoj knjizi.

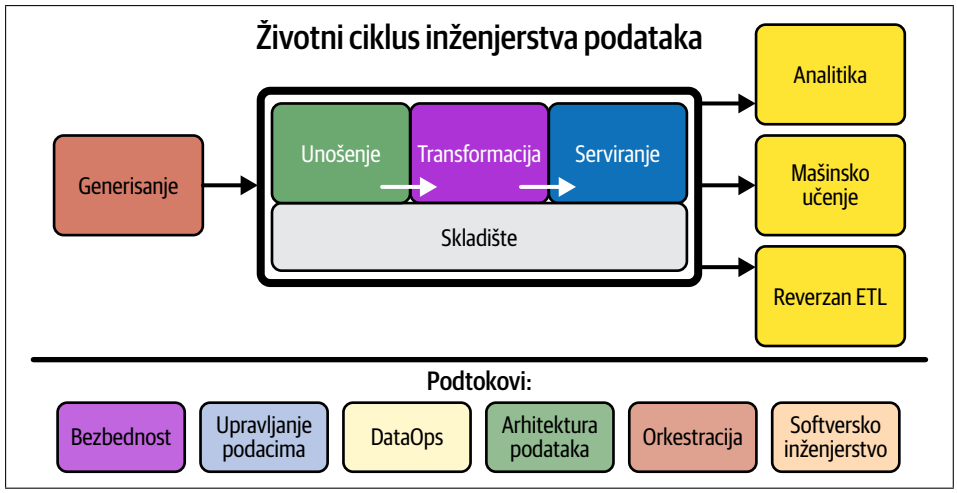
[3] Jesse Anderson, „The Two Types of Data Engineering", 27. jun 2018, *https://oreil.ly/dxDt6*.

[4] Maxime Beauchemin, „The Rise of the Data Engineer", 20. januar 2017, *https://oreil.ly/kNDmd*.

[5] Lewis Gavin, *What Is Data Engineering?* (Sebastapol, CA: O'Reilly, 2020), *https://oreil.ly/ELxLi*.

Životni ciklus inženjerstva podataka

Previše je lako fokusirati se isključivo na tehnologiju i izgubiti iz vida celokupnu sliku. Ova knjiga se fokusira oko velike ideje nazvane *životni ciklus inženjerstva podataka* (Slika 1-1), za koju verujemo da daje inženjerima podataka holistički kontekst za sagledavanje njihove uloge.



Slika 1-1. Životni ciklus inženjerstva podataka

Životni ciklus inženjerstva podataka menja perspektivu sa tehnologije na same podatke i krajnje ciljeve koje oni moraju da zadovolje. Faze ciklusa inženjerstva podataka su sledeće:

- Generisanje
- Skladištenje
- Unošenje
- Transformacija
- Serviranje

Životni ciklus inženjerstva podataka ima pojam *podtokovi* (undercurrent) – ključne ideje koje prevladavaju tokom celokupnog ciklusa. To su bezbednost, upravljanje podacima, DataOps, arhitektura podataka, orkestracija i softversko inženjerstvo. O ciklusu inženjerstva podataka i njegovim podtokovima govorićemo više u Poglavlju 2. Ipak, uvodimo ih ovde jer su suštinski važni za našu definiciju inženjerstva podataka i izlaganje koji sledi u ovom poglavlju.

Sada kada imate radnu definiciju inženjerstva podataka i uvod u njegov životni ciklus, hajde da se malo vratimo unazad i pogledamo malo istorije.

Evolucija inženjera podataka

> Istorija se ne ponavlja, ali se rimuje.
>
> – Poznata izreka često pripisivana Marku Tvenu

Razumevanje inženjerstva podataka danas i sutra zahteva kontekst o evoluciji ove oblasti. Ovaj deo nije lekcija iz istorije, ali osvrt na prošlost je neprocenjiv da bismo razumeli gde se danas nalazimo i kuda stvari idu. Jedna zajednička tema se stalno ponavlja: ono što je staro, ponovo je novo.

Rani dani: 1980 do 2000, od skladišta podataka do veba

Rođenje inženjera podataka se moglo reći da je ukorenjeno u skladištenju podataka, datirajući još od 1970-ih, kada se oblik *poslovnog skladišta podataka* formirao tokom 1980-ih i kada je Bil Inmon zvanično skovao izraz *skladište podataka* (data warehouse) 1989. godine. Nakon što su inženjeri u IBM-u razvili relacionu bazu i i Jezik strukturisanih upita (SQL, Structured Query Language), Oracle je popularizovao tu tehnologiju. Kako su sistemi podataka u povoju rasli, biznisi su zahtevali namenske alate i cevovode podataka tj. protočnu obradu (data pipelines) za izveštavanje i poslovnu inteligenciju (BI, business intelligence). Da bi ljudi pravilno modelovali svoju poslovnu logiku u skladištu podataka, Ralf Kimball i Inmon su razvili svoje istoimene tehnike i pristupe modelovanju podataka, koji se i dandanas široko koriste.

Skladišta podataka su donela prvu eru skalabilnih analitika, sa novim bazama podataka sa masivno paralelnim obradama (MPP) koje koriste više procesora za obradu velike količine podataka na tržištu i podršku bez presedana velikog obima podataka. Uloge kao što su inženjer poslovne inteligencije inženjer razvoja ETL-a i inženjer skladišta podataka su se pojavile da zadovolje različite potrebe skladišta podataka. Skladištenje podataka i inženjerstvo poslovne inteligencije su preteče današnjeg inženjerstva podataka i dalje imaju ključnu ulogu u ovoj disciplini.

Internet je postao mejnstrim oko sredine 1990-ih, stvarajući potpuno novu generaciju kompanija orijentisanih na veb, poput AOL-a, Yahoo-a i Amazona. Dotcom bum je izazvao masu aktivnosti u veb aplikacijama i backend sistemima koji ih podržavaju – serverima, bazama podataka i skladištima. Većina infrastrukture je bila skupa, monolitna i sa opterećujućim licencama. Prodavci ovih backend sistema verovatno nisu mogli predviđati ogromno skaliranje podataka koje bi veb aplikacije proizvele.

Rane 2000-te: Rođenje savremenog inženjerstva podataka

Brzo idemo dalje do ranih 2000-ih, kada je dot-com bum kasnih ’90-ih propao, ostavljajući iza sebe malu grupu preživelih. Neka od ovih kompanija, kao što su Yahoo, Google i Amazon, porasla su u moćne tehnološke kompanije. U početku, ove kompanije su nastavile da se oslanjaju na tradicionalne monolitne, relacione

baze podataka i skladišta podataka iz 1990-ih, gurajući ove sisteme do krajnjih granica. Kako su ovi sistemi počeli da popuštaju, bili su potrebni novi pristupi da se podnese rast podataka. Novi naraštaj sistema mora biti isplativ, skalabilan, dostupan i pouzdan.

Uz eksploziju podataka, hardver – poput servera, RAM-a, diskova i fleš diskova – postao je jeftin i svuda prisutan. Nekoliko inovacija je omogućilo distribuirani račun i skladištenje na ogromnim računarskim klasterima na velikoj skali. Ove inovacije su počele decentralizovati i razbijati tradicionalno monolitne usluge. *Big data* era je počela.

Oksfordski engleski rečnik definiše *big data* kao „izuzetno velike skupove podataka koji se mogu računski analizirati da bi se otkrili obrasci, trendovi i povezanost, posebno u odnosu na ljudsko ponašanje i interakcije". Drugi čuven i sažet opis big data je tri *V* podataka: brzina, raznolikost i obim (velocity, variety, volume).

Google je 2003. godine objavio rad o Google File System (GFS), a ubrzo nakon toga, 2004. godine i rad o MapReduce, ultra skalabilnoj paradigmi za obradu podataka. Istina je da veliki podaci (big data) imaju svoje prethodnike u MPP skladištima podataka i sistemu upravljanja podacima za eksperimentalne projekte fizike, ali Googlove publikacije su predstavljale „veliki prasak" za tehnologije obrade podataka i temelje inženjerstva podataka kakvog danas poznajemo. Više o MPP sistemima i MapReduce moći ćete saznati u Poglavlju 3 i 8, respektivno.

Inspirisan radovima kompanije Google, tim inženjera u Yahoou je razvio, a kasnije i otvorio izvorni kôd Apache Hadoop-a 2006. godine.[6] Teško je preuveličati uticaj Hadoopa. Softverski inženjeri zainteresovani za probleme obrade velikih količina podataka bili su privučeni mogućnostima ovog novog otvorenog ekosistema tehnologija. Kako su kompanije svih veličina i tipova uočile rast svojih podataka na stotine terabajta pa čak i petabajta, rodilo se doba inženjera velikih podataka.

U isto vreme, Amazon je morao da se izbori sa svojim sopstvenim eksplodirajućim potrebama za podacima i kreirao je elastična računarsko okruženja (Amazon Elastic Compute Cloud, tj. EC2), beskrajno skalabilne sisteme za skladištenje (Amazon Simple Storage Service, tj. S3), visoko skalabilne NoSQL baze podataka (Amazon DynamoDB) i mnoge druge osnovne gradivne komponente za podatke.[7] Amazon je odlučio da ove usluge ponudi za unutrašnju i spoljašnju upotrebu kroz *Amazon Web Services* (AWS), postajući prva popularna javna klaud platforma. AWS je stvorio ultra fleksibilno tržište resursa, gde se plaća po potrošnji, virtualizacijom i preprodajom ogromnih bazena (pools) široko dostupne hardverske

[6] Cade Metz, „How Yahoo Spawned Hadoop, the Future of Big Data", *Wired*, 18. oktobar 2011, *https://oreil.ly/iaD9G*.

[7] Ron Miller, „How AWS Came to Be", *TechCrunch*, 2. jul 2016, *https://oreil.ly/VJehv*.

opreme. Umesto kupovine hardvera za data centar, razvojni programeri su jednostavno mogli iznajmiti računarsku snagu i skladište od AWS-a.

Kako je AWS postajao visoko profitabilan motor rasta za Amazon, uskoro su usledile i druge javne klaud (cloud – oblak) platforme, kao što su Google Cloud, Microsoft Azure i DigitalOcean. Javni klaud se može smatrati jednom od najznačajnijih inovacija 21. veka i pokrenuo je revoluciju u načinu na koji se razvijaju i primenjuju softverske i aplikacije podataka.

Rane alatke za obradu velikih podataka i javni klaud su postavili temelje današnjeg ekosistema podataka. Savremeni prostor podataka – i inženjerstvo podataka kakvog sada poznajemo – ne bi postojao bez ovih inovacija.

2000-te i 2010-e: Inženjerstvo velikih podataka

Alatke otvorenog koda za rad sa velikim podacima u Hadoop ekosistemu su brzo sazrele i proširile se iz Silicijumske doline ka tehnološki naprednim kompanijama širom sveta. Prvi put, svaki posao je imao pristup istim naprednim alatima za podatke koje su koristile vodeće tehnološke kompanije. Još jedna revolucija se dogodila sa prelaskom sa obrade u grupama ili paketima (batch computing) na strim (u toku, stream) obradu događaja, uvodeći novu eru velikih „u realnom vremenu" podataka. Obradi u grupama i strim obradi događaja naučićete kroz ovu knjigu.

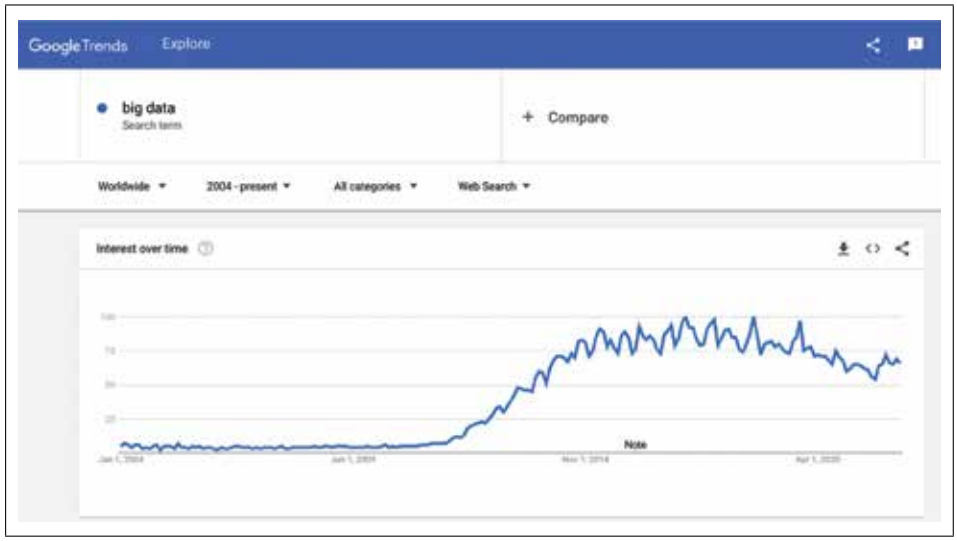
Inženjeri su mogli izabrati najnovije i najbolje – Hadoop, Apache Pig, Apache Hive, Dremel, Apache HBase, Apache Storm, Apache Cassandra, Apache Spark, Presto i brojne druge nove tehnologije koje su se pojavile na sceni. Tradicionalni orijentisani ka preduzeću i alati za podatke bazirani na grafičkom korisničkom interfejsu (GUI) odjednom su delovali zastareli, a prvo kôd inženjerstvo je bilo u trendu sa usponom MapReduce. Mi (autori) smo bili prisutni u to vreme i činilo se kao da stare dogme umiru iznenadnom smrću na oltaru velikih podataka.

Eksplozija alatki za podatke krajem 2000-ih i 2010-ih je označilo pojavu *inženjera velikih podataka*. Da bi efikasno koristili ove alatke i tehnike – tj. Hadoop ekosistem koji sadrži Hadoop, YARN, Hadoop Distributed File System (HDFS) i MapReduce – inženjeri velikih podataka su morali biti vešti u razvoju softvera i radu na niskom nivou infrastrukture, ali sa promenjenim težištem. Inženjeri velikih podataka su obično održavali ogromne klastere široko dostupne hardverske opreme da bi isporučivali podatke na velikoj skali. Iako bi povremeno slali zahteve za osnovni kôd Hadoopa, fokus im se premeštao od razvoja osnovne tehnologije ka isporuci podataka.

Veliki podaci su brzo postali žrtva sopstvenog uspeha. Kao savremen izraz, *big data* je postao popularan tokom ranih 2000-ih do sredine 2010-ih. Veliki podaci su zarobili maštu kompanija koje su pokušavale da nađu smisao u stalnom rastu količina podataka i stalnom bombardovanju bezobraznim marketingom od strane

kompanija koje prodaju alatke i usluge za velike podatke. Zbog velikog hajpa, bilo je uobičajeno videti kompanije koje koriste alatke za velike podatke za male probleme sa podacima, ponekad postavljajući Hadoop klaster da bi se obradilo samo nekoliko gigabajta. Delovalo je kao da svi žele deo akcije velikih podataka. Dan Ariely je tvitovao, „Veliki podaci su kao seks u tinejdžerskim danima: svi pričaju o tome, niko zapravo ne zna kako to da uradi, svi misle da svi drugi to rade, pa tako svi tvrde da to rade".

Slika 1-2 prikazuje snimak iz Google Trend-a za termin pretrage „big data" da bi se dobila ideja o usponu i padu velikih podataka.



Slika 1-2. Google Trends za „big data" (mart 2022)

Uprkos popularnosti termina interes za velike podatke (big data) je oslabio. Šta se desilo? Jedna reč: pojednostavljenje. Uprkos moći i sofisticiranosti alata otvorenog koda za obradu velikih podataka, njihovo upravljanje je zahtevalo puno rada i stalnu pažnju. Često su kompanije zapošljavale čitave timove inženjera za velike podatke koji su koštali milione dolara godišnje da bi „čuvali" ove platforme. Inženjeri za velike podatke su često provodili previše vremena održavajući komplikovane alate i moglo bi se reći da nisu provodili dovoljno vremena na pružanju uvida i vrednosti važnih za poslovanje.

Razvojni inženjeri otvorenog koda, oblaci i treće strane počeli su da traže načine da apstrahuju, pojednostave i omoguće pristup velikim podacima bez visokih administrativnih troškova i cene upravljanja njihovim klasterima, kao i instalaciju, konfiguraciju i nadogradnju njihovog otvorenog koda. Termin *veliki podaci* je u suštini postao relikt koji opisuje određeno vreme i pristup obradi velike količine podataka.

Danas se podaci kreću brže nego ikad i rastu sve više, ali obrada velikih podataka je postala toliko dostupna da više ne zaslužuje poseban termin; svaka kompanija teži rešavanju svojih problema sa podacima, bez obzira na stvarnu veličinu tih podataka. Inženjeri za velike podatke su sada jednostavno *inženjeri podataka*.

2020-te: Inženjerstvo za životni ciklus podataka

U trenutku pisanja ove knjige, uloga inženjera podataka brzo se razvija. Očekujemo da će ova evolucija nastaviti ubrzano u doglednoj budućnosti. Dok su inženjeri podataka istorijski bili skloni detaljima na niskom nivou monolitnih okvira poput Hadoopa, Sparka ili Informaticae, trend se pomera ka decentralizovanim, modularizovanim, upravljanim i visoko apstraktnim alatima.

Zaista, alati za podatke su se razmnožili neverovatnom brzinom (Slika 1-3). Popularni trendovi početkom 2020-ih uključuju *savremen stek podataka*, koji predstavlja kolekciju gotovih proizvoda otvorenog koda i proizvoda trećih strana sklopljenih da bi se olakšao rad analitičarima. Istovremeno izvori podataka i formati podataka rastu kako po raznolikosti, tako i po veličini. Inženjerstvo podataka je sve više disciplina interoperabilnosti, povezivanje različitih tehnologija poput LEGO kockica, za postizanje krajnjih poslovnih ciljeva.

Slika 1-3. Matt Truckov prostor podataka iz 2012. u odnosu na 2021.

Inženjera podataka o kojem govorimo u ovoj knjizi možemo preciznije opisati kao *inženjera životnog ciklusa podataka*. S većom apstrakcijom i pojednostavljenjem inženjer životnog ciklusa podataka više nije opterećen teškim detaljima jučerašnjih okvira za velike podatke. Iako inženjeri podataka održavaju veštine programiranja na niskom nivou podataka i koriste ih po potrebi, sve češće nalaze svoju ulogu usmerenu ka stvarima više u lancu vrednosti: bezbednost, upravljanje podacima, DataOps, arhitektura podataka, orkestracija i opšte upravljanje životnim ciklusom podataka.[8]

[8] DataOps je skraćenica za *operacije sa podacima* (data operations). Ovu temu pokrivamo u Poglavlju 2. Za više informacija, pročitajte DataOps Manifesto.

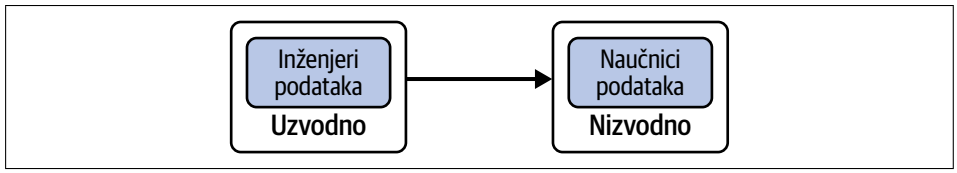
Kako se alati i radni procesi pojednostavljuju, primetili smo značajnu promenu u stavovima inženjera podataka. Umesto fokusiranja na to ko ima „najveće podatke“, otvoreni projekti i servisi se sve više bave upravljanjem i upravljanju podacima, čineći ih lakšim za korišćenje i otkrivanje i poboljšavajući njihov kvalitet. Inženjeri podataka se sada upoznaju sa skraćenicama poput *CCPA* i *GDPR*;[9] dok projektuju cevovode, bave se privatnošću, anonimizacijom, skupljanjem „otpada“ podataka i usklađivanjem sa propisima.

Ono što je staro je ponovo novo. Dok su „korporativne“ stvari poput upravljanja podacima (uključujući kvalitet i upravljanje) bile uobičajene za velika preduzeća pre ere velikih podataka, one nisu bile široko usvojene u manjim kompanijama. Sada kada su mnogi izazovni problemi jučerašnjih sistema podataka rešeni, lepo upakovani i proizvedeni, tehnolozi i preduzetnici su ponovo usmerili fokus na „korporativne“ stvari, ali sa naglaskom na decentralizaciju i agilnost, što je u suprotnosti sa tradicionalnim korporativnim pristupom naredbi i kontrole.

Smatramo da je sadašnji trenutak zlatno doba upravljanja životnim ciklusom podataka. Inženjeri podataka koji upravljaju životnim ciklusom inženjerstva podataka imaju bolje alate i tehnike nego ikada pre. O životnom ciklusu inženjerstva podataka i njegovim podtokovima biće detaljnije rečeno u sledećem poglavlju.

Inženjerstvo podataka i nauka o podacima

Gde se inženjerstvo podataka uklapa u nauku o podacima (data science)? Postoje neke debate, sa nekima koji tvrde da je inženjerstvo podataka poddisciplina nauke o podacima. Mi verujemo da je inženjerstvo podataka *odvojen* od nauke o podacima i analitike. Oni se dopunjuju, ali su jasno različiti. Inženjerstvo podataka se nalazi uzvodno od nauke o podacima (vidi sliku 1-4), što znači da inženjeri podataka pružaju ulaze koje koriste naučnici podataka (nizvodno od inženjerstva podataka), koji ove ulaze pretvaraju u nešto korisno.



Slika 1-4. Inženjerstvo podataka nalazi se uzvodno od nauke o podacima

Razmotrite hijerarhiju potreba nauke o podacima (Slika 1-5). Monica Rogati je 2017. godine objavila ovu hijerarhiju u članku koji je pokazao gde se veštačka

[9] Ove skraćenice stoje za *Zakon o zaštiti privatnosti potrošača Kalifornije* (California Consumer Privacy Act) i *Opšta uredba o zaštiti podataka* (General Data Protection Regulation), respektivno.

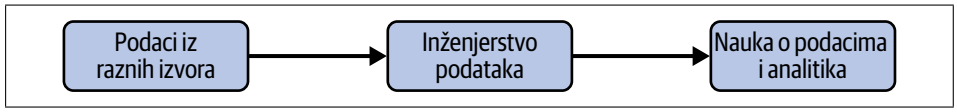
inteligencija (AI) i mašinsko učenje (ML) nalaze u blizini „rutinskih" oblasti kao što su kretanje/skladištenje podataka, sakupljanje i infrastruktura.

Slika 1-5. Hijerarhija potreba nauke o podacima

Iako mnogi naučnici podataka sa nestrpljenjem očekuju izgradnju i podešavanje ML modela, realnost je da procenjenih 70% do 80% njihovog vremena provedu baveći se donjim segmentima hijerarhije – sakupljanjem podataka, čišćenjem podataka, obradom podataka – i samo malim delom svog vremena na analizi i ML. Rogati tvrdi da kompanije moraju izgraditi čvrstu osnovu za podatke (tri osnovna nivoa hijerarhije) pre nego što se pozabave oblastima poput AI i ML.

Naučnici podataka obično nisu obučeni za inženjerstvo podataka na nivou proizvodnje i često obavljaju ovaj posao neplanski jer nemaju podršku i resurse inženjera podataka. U idealnom svetu, trebalo bi da naučnici podataka provedu više od 90% svog vremena fokusirajući se na najviše slojeve piramide: analitika, eksperimentisanje i ML. Kada se inženjeri podataka fokusiraju na ove donje delove hijerarhije, oni grade čvrst temelj za uspeh naučnika podataka.

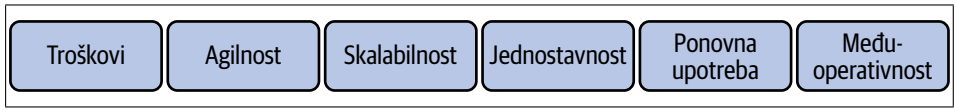
Sa naukom podataka koja pokreće napredne analitike i ML inženjerstvo podataka premošćuje razliku između dobijanja podataka i dobijanja vrednosti od podataka (Slika 1-6). Verujemo da je inženjerstvo podataka jednako važan i vidljiv kao i nauka o podacima, sa inženjerima podataka koji igraju ključnu ulogu u uspehu nauke o podacima u proizvodnji.



Slika 1-6. Inženjer podataka uzima podatke i stvara vrednost iz podataka

Veštine i aktivnosti inženjerstva podataka

Skup veština inženjera podataka obuhvata „podtokove" inženjerstva podataka: bezbednost, upravljanje podacima, DataOps, arhitektura podataka i softversko inženjerstvo. Ovaj skup veština zahteva razumevanje kako se procenjuju alati za podatke i kako se uklapaju kroz životni ciklus inženjerstva podataka. Ključno je znati kako se podaci proizvode u izvornim sistemima i kako će analitičari i naučnici podataka konzumirati i stvarati vrednost nakon obrade i kultivacije podataka. Na kraju inženjer podataka upravlja mnogim složenim pokretnim delovima i mora stalno da optimizuje duž osa troškova, agilnosti, skalabilnosti (proširivosti), jednostavnosti, ponovne upotrebe i međuoperativnost (Slika 1-7). O ovim temama ćemo govoriti detaljnije u predstojećim poglavljima.



Slika 1-7. Balansiranje inženjerstva podataka

Kao što smo nedavno izneli, od inženjera podataka se očekivalo da poznaje i razume kako se koristi nekoliko moćnih i monolitnih tehnologija (Hadoop, Spark, Teradata, Hive i mnoge druge) za stvaranje rešenja za podatke. Korišćenje ovih tehnologija često zahteva sofisticirano razumevanje softverskog inženjerstva, mrežnog povezivanja, distribuiranog računarstva, skladištenja ili drugih detalja niskog nivoa. Njihov rad bi bio posvećen administraciji klastera i održavanju, upravljanju preopterećenjem i pisanju poslova za cevovod i transformaciju između ostalog.

Danas je prostor alata za podatke dramatično manje komplikovan za upravljanje i implementaciju. Savremeni alati za podatke značajno apstrahuju i pojednostavljuju radne tokove. Kao rezultat inženjeri podataka su sada fokusirani na balansiranje najjednostavnijih i najisplativijih, najboljih usluga koje pružaju vrednost poslovanju. Od inženjera podataka očekuje se da stvori agilne arhitekture podataka koje evoluiraju kako se pojavljuju novi trendovi.

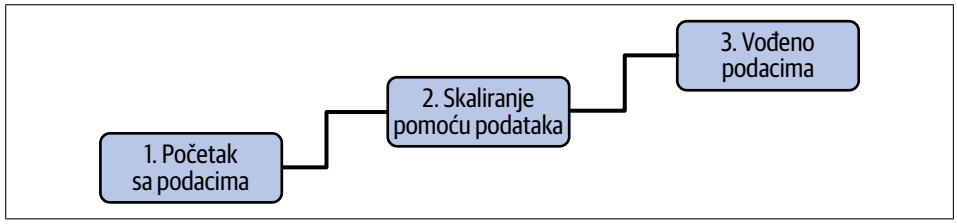
Koje su od nekih stvari koje inženjer podataka *ne* radi? Inženjer podataka obično ne gradi direktno ML modele, ne kreira izveštaje ili kontrolne table (dashboards), ne obavlja analizu podataka, ne gradi ključne pokazatelje performansi (KPI, key performance indicators) niti razvija softverske aplikacije. Inženjer podataka treba da ima dobro funkcionalno razumevanje ovih oblasti da bi najbolje služio interesima zainteresovanih strana.

Zrelost podataka i inženjer podataka

Nivo složenosti inženjerstva podataka unutar kompanije u velikoj meri zavisi od zrelosti podataka te kompanije. To značajno utiče na svakodnevne radne odgovornosti i napredovanje karijere inženjera podataka. Šta je zapravo zrelost podataka?

Zrelost podataka je napredovanje ka većem korišćenju podataka, sposobnostima i integraciji širom organizacije, ali zrelost podataka ne zavisi jednostavno od starosti ili prihoda kompanije. Startap u ranoj fazi može imati veću zrelost podataka od stogodišnje kompanije sa godišnjim prihodima u milijardama. Ono što je važno jeste način na koji se podaci koriste kao konkurentska prednost.

Modeli zrelosti podataka imaju mnogo verzija, kao što su Model zrelosti upravljanja podacima (DMM, Data Management Maturity) i drugi i teško je odabrati onaj koji je jednostavan i koristan za inženjerstvo podataka. Zato ćemo stvoriti svoj sopstveni pojednostavljeni model zrelosti podataka. Naš model zrelosti podataka (Slika 1-8) ima tri faze: početnu s podacima, skalirajuću s podacima i vođenje podacima. Hajde da pogledamo svaku od ovih faza i šta obično radi inženjer podataka u svakoj fazi.



Slika 1-8. Naš pojednostavljen kompanijski model zrelosti podataka.

Faza 1: Početak sa podacima

Preduzeće koje započinje rad sa podacima je, po definiciji, u vrlo ranoj fazi svoje zrelosti podataka. Kompanija može imati nejasne, slabo definisane ciljeve ili ih uopšte nemati. Arhitektura i infrastruktura podataka su u vrlo ranim fazama planiranja i razvoja. Usvojenost i korišćenje su verovatno niski ili nepostojeći. Tim za podatke je mali, često sa brojem zaposlenih u jednocifrenim brojevima. U ovoj fazi inženjer podataka obično je univerzalista i često obavlja više drugih uloga, kao što su naučnik podataka ili softverski inženjer. Cilj inženjera podataka je da brzo napreduje, stekne pažnju i da dodaje vrednost.

Praktičnosti dobijanja vrednosti iz podataka su tipično slabo razumljive, ali želja postoji. Izveštajima ili analizama nedostaju formalne strukture i većina zahteva za podacima je ad hoc. Iako je primamljivo odmah zaroniti u mašinsko učenje (ML, machine learning) u ovoj fazi, ne preporučujemo to. Videli smo bezbroj timova za podatke koji su zapeli i nisu uspeli kada su pokušali da skoče u ML bez izgradnje čvrste osnove podataka.

To ne znači da ne možete postići uspeh sa ML u ovoj fazi – to je retko, ali moguće. Bez čvrste osnove podataka, verovatno nećete imati podatke za obučavanje pouzdanih ML modela niti sredstva za njihovu implementaciju u proizvodnju na skalabilan i ponovljiv način. Šaljivo se nazivamo „oporavljeni naučnici podataka", uglavnom iz ličnog iskustva preranog ulaska u projekte nauke podataka bez adekvatne zrelosti podataka ili podrške inženjerstva podataka.

Inženjer podataka bi trebalo da se fokusira na sledeće u organizacijama koje započinju rad sa podacima:

- Da dobije podršku ključnih zainteresovanih strana, uključujući izvršni menadžment. Idealno, trebalo bi da inženjer podataka ima sponzora za ključne inicijative da bi projektovao i izgradio arhitekturu podataka koja podržava ciljeve kompanije.
- Da definiše pravu arhitekturu podataka (obično sami, budući da arhitekta podataka verovatno ne postoji). To znači određivanje poslovnih ciljeva i konkurentske prednosti koju ciljate postići svojom inicijativom. Radite na arhitekturi podataka koja podržava te ciljeve. Pogledajte Poglavlje 3 za naš savet o „dobroj" arhitekturi podataka.
- Da identifikujte i procenite podatke koji će podržati ključne inicijative i delovati unutar arhitekture podataka koju ste projektovali.
- Da izgradi čvrstu osnovu podataka za buduće analitičare i naučnike podataka da bi generisali izveštaje i modele koji pružaju konkurentsku vrednost. U međuvremenu, možda ćete i sami morati da generišete te izveštaje i modele dok se ovaj tim ne zaposli.

Ovo je delikatna faza sa mnogo zamki. Evo nekih saveta za ovu fazu:

- Organizaciona volja može oslabiti ako se ne postignu mnogi vidljivi uspesi sa podacima. Postizanje brzih uspeha uspostaviće važnost podataka unutar organizacije. Samo imajte na umu da će brzi uspesi verovatno stvoriti tehnički dug. Imajte plan za smanjenje ovog duga, jer će inače dodati usporenje za buduću isporuku.
- Izađite i razgovarajte sa ljudima i izbegavajte rad u izolaciji. Često vidimo tim za podatke koji „radi u svom balonu", ne komunicirajući sa ljudima izvan svojih odeljenja i ne dobijajući perspektive i povratne informacije od poslovnih zainteresovanih strana. Opasnost je da ćete provesti mnogo vremena radeći na stvarima od malo koristi za ljude.
- Izbegavajte neodređeni teški rad. Nemojte se opterećivati nepotrebnom tehničkom složenošću. Koristite gotova, ključ-u-ruke rešenja gde god je to moguće.
- Izgradite prilagođena rešenja i kôd samo tamo gde to stvara konkurentsku prednost.

Faza 2: Skaliranje pomoću podataka

U ovoj tački, preduzeće je prešlo sa ad hoc zahteva za podacima na formalne prakse sa podacima. Sada je izazov stvaranje skalabilnih arhitektura podataka i planiranje za budućnost gde je kompanija zaista vođena podacima. Uloge inženjera podataka prelaze od univerzalaca do specijalista, sa ljudima koji se fokusiraju na određene aspekte životnog ciklusa inženjerstva podataka.

U organizacijama koje su u fazi 2 zrelosti podataka, ciljevi inženjera podataka su:

- Uspostavljanje formalne prakse sa podacima
- Kreiranje skalabilne i robustne arhitekture podataka
- Usvajanje DevOps i DataOps prakse
- Izgradnja sistema koji podržavaju ML
- Nastavljanje izbegavanja neodređenog teškog rada osim kada iz toga proizlazi konkurentska prednost

Vratićemo se ovim ciljevima kasnije u knjizi.

Problemi na koje treba obratiti pažnju uključuju sledeće:

- Kako postajemo sofisticiraniji s podacima, postoji iskušenje da usvojimo tehnologije koje su na samom vrhu inovacija na osnovu društvenog dokaza kompanija iz Silicijumske doline. Retko kada je to dobra upotreba vašeg vremena i energije. Sve odluke o tehnologiji treba da budu vođene vrednošću koju će one doneti vašim klijentima.
- Glavna prepreka za skaliranje nisu čvorovi klastera, skladištenje ili tehnologija, već tim za inženjerstvo podataka. Fokusirajte se na rešenja koja su jednostavna za primenu i upravljanje da biste povećali propusni opseg (throughput) vašeg tima.
- Bićete u iskušenju da se predstavite kao tehnolog, genijalac za podatke koji može da isporuči magične proizvode. Preusmerite svoj fokus umesto na pragmatično vođstvo i započnite prelazak na sledeću fazu zrelosti; komunicirajte s drugim timovima o praktičnoj korisnosti podataka. Naučite organizaciju kako da konzumira i iskoristi podatke.

Faza 3: Vođenje podacima

U ovoj fazi, kompanija je vođena podacima. Automatizovani tokovi podataka i sistemi kreirani od strane inženjera podataka omogućavaju ljudima unutar kompanije da obavljaju analitiku i mašinsko učenje samostalno. Uvođenje novih izvora podataka je besprekorno i izvučena je opipljiva vrednost. Inženjeri podataka sprovode odgovarajuće kontrole i prakse da bi se uverili da su podaci uvek dostupni

ljudima i sistemima. Uloge inženjera podataka nastavljaju se specijalizovati dublje nego u fazi 2.

U organizacijama u fazi 3 zrelosti podataka inženjer podataka će nastaviti rad na prethodnim fazama, plus će raditi sledeće:

- Stvarati automatizaciju za besprekorno uvođenje i korišćenje novih podataka
- Fokusirati se na izgradnju prilagođenih alata i sistema koji koriste podatke kao konkurentsku prednost
- Fokusirati se na vrhunske aspekte podataka, kao što su upravljanje podacima (uključujući upravljanje podacima i kvalitet) i DataOps
- Primeniti alate koji izlažu i šire podatke kroz organizaciju, uključujući kataloge podataka, alate za poreklo podataka i sisteme za upravljanje metapodacima
- Efikasno sarađivati sa softverskim inženjerima, inženjerima mašinskog učenja, analitičarima i drugima
- Stvaranje zajednice i okruženja gde ljudi mogu sarađivati i otvoreno govoriti, bez obzira na njihovu ulogu ili poziciju

Problemi na koje treba obratiti pažnju uključuju sledeće:

- U ovoj fazi, samozadovoljstvo je značajna opasnost. Jednom kada organizacije dostignu fazu 3, moraju konstantno da se fokusiraju na održavanje i poboljšanje ili rizikuju da padnu nazad na nižu fazu.
- Tehnološka skretanja su veća opasnost ovde nego u ostalim fazama. Postoji iskušenje da se prate skupi hobi projekti koji ne donose vrednost poslovanju. Koristite prilagođenu tehnologiju samo tamo gde pruža konkurentsku prednost.

Pozadina i veštine inženjera podataka

Inženjerstvo podataka je brzo rastuće polje i još uvek postoji puno pitanja o tome kako postati inženjer podataka. Pošto je inženjerstvo podataka relativno nova disciplina, malo formalnog obrazovanja je dostupno za ulazak u ovu oblast. Univerziteti nemaju standardizovan put ka inženjerstvu podataka. Iako postoje pojedinačni inženjerski kampovi i onlajn tutorijali koji pokrivaju različite teme, zajednički kurikulum za ovu oblast još uvek ne postoji.

Ljudi koji ulaze u inženjerstvo podataka dolaze sa različitim obrazovanjem, karijerom i veštinama. Svi koji ulaze u ovu oblast trebalo bi da očekuju da će investirati značajnu količinu vremena u samostalno učenje. Čitanje ove knjige je dobar početni korak; jedan od osnovnih ciljeva ove knjige je da vam pruži temelj za znanje i veštine za koje mislimo da su neophodne da biste uspeli kao inženjer podataka.

Ako usmeravate svoju karijeru u inženjerstvo podataka, otkrili smo da je prelazak najlakši kada se kreće iz srodnih oblasti, kao što su softversko inženjerstvo, razvoj ETL-a, administracija baza podataka, nauka o podacima ili analiza podataka. Ove discipline su obično „svesne podataka" i pružaju dobar kontekst za uloge podataka u organizaciji. Opremljuju ljude sa relevantnim tehničkim veštinama i kontekstom za rešavanje problema inženjerstva podataka.

Uprkos nedostatku formalizovanog puta, postoji određeno osnovno znanje koje verujemo da inženjer podataka treba da zna da bi bio uspešan. Po definiciji inženjer podataka mora razumeti i podatke i tehnologiju. U pogledu podataka, to uključuje poznavanje različitih najboljih praksi oko upravljanja podacima. Kada je u pitanju tehnologija inženjer podataka mora biti obavešten o različitim opcijama za alate, njihovom međusobnom delovanju i kompromisima. To zahteva dobro razumevanje softverskog inženjerstva, DataOpsa i arhitekture podataka.

Kada se posmatra šira slika inženjer podataka mora da razume zahteve korisnika podataka (analitičara podataka i naučnika podataka) i šire implikacije podataka unutar organizacije. Inženjerstvo podataka je celovita praksa; najbolji inženjeri podataka svoje odgovornosti posmatraju kroz poslovne i tehničke aspekte.

Poslovne odgovornosti

U ovoj odeljku ćemo dati spisak makro odgovornosti koje nisu ekskluzivne za inženjere podataka, ali su ključne za svakoga ko radi u oblasti podataka ili tehnologije. Zato što jednostavna Google pretraga pruža tone resursa za učenje o ovim oblastima, mi ćemo ih ovde samo nabrojati radi sažetosti:

Znati kako da komunicirate sa tehnički neobrazovanim i tehnički obrazovanim korisnicima.

Komunikacija je ključna i morate biti sposobni da uspostavite dobar odnos i poverenje sa ljudima širom organizacije. Predlažemo da pažljivo pratite hijerarhiju organizacije, ko kome izveštava, kako ljudi međusobno interaguju i koja postojanje odeljenja koja su izolovana. Ova zapažanja će biti neprocenjiva za vaš uspeh.

Razumeti kako definisati i prikupiti poslovne i proizvodne zahteve.

Morate znati šta da gradite i postarati se da se vaši deoničari slažu sa vašom procenom. Morate razviti osećaj kako odluke o podacima i tehnologiji utiču na poslovanje.

Razumeti osnove kulture Agile, DevOps i DataOps.

Mnogi tehnolozi pogrešno veruju da su ove prakse rešene tehnologijom. Mi smatramo da je to opasno pogrešno. Agile, DevOps i DataOps su u kulturne osnove, zahtevajući prihvatanje širom organizacije.

Upravljati troškovima.

Bićete uspešni kada možete da održite niske troškove pružajući nesrazmernu vrednost. Znati kako optimizovati posao da se potroši što manje vremena, ukupan trošak vlasništva i propuštene prilike. Naučite nadgledati troškove da biste izbegli iznenađenja.

Kontinuirano učenje.

Polje sa podacima deluje da se menja brzinom svetlosti. Ljudi koji uspevaju u njemu su odlični u usvajanju novih stvari uz izoštravanje osnovnog znanja. Oni su dobri u filtriranju, određivanju koja su nova dostignuća najrelevantnija za njihov rad, koja su još uvek nezrela i koja su samo prolazni trendovi. Ostanite informisani o oblasti i naučite kako da učite.

Uspešan inženjer podataka uvek izlazi iz okvira i shvata širu sliku i kako postići nesrazmernu vrednost za poslovanje. Komunikacija je vitalna, kako za tehničke, tako i za netehničke ljude. Često vidimo da timovi za podatke postižu uspeh na osnovu svoje komunikacije sa ostalim zainteresovanima; uspeh ili neuspeh retko je tehnološko pitanje. Znajući kako da se snalazite u organizaciji, da definišete i prikupite zahteve, da kontrolišete troškove uz kontinuirano učenje će vas izdvojiti od inženjera podataka koji se oslanjaju isključivo na svoje tehničke sposobnosti za napredovanje u karijeri.

Tehničke odgovornosti

Morate razumeti kako da gradite arhitekture koje optimiziraju performanse i troškove na visokom nivou, koristeći gotove komponente ili komponente napravljene u preduzeću. Na kraju, arhitekture i tehnologije koje ih čine su građevinski blokovi koji služe životnom ciklusu inženjerstva podataka. Podsetite se na faze životnog ciklusa inženjerstva podataka:

- Generisanje
- Skladištenje
- Unos podataka
- Transformacija
- Pružanje usluga

Podtokovi životnog ciklusa inženjerstva podataka su sledeći:

- Bezbednost
- Upravljanje podacima
- DataOps
- Arhitektura podataka

- Orkestracija
- Softversko inženjerstvo

Zumirajući malo, u ovoj sekciji razmatramo neke od taktičkih podataka i tehnoloških veština koja će vam biti potrebna kao inženjeru podataka; to ćemo detaljnije obrađivati u narednim poglavljima.

Pitanje koje se često postavlja glasi: da li inženjer podataka treba da zna da programira? Kratak odgovor: da. Inženjer podataka treba da ima programersku veštinu na nivou proizvodnje. Uočavamo da se priroda softverskih razvojnih projekata kojima se bave inženjeri podataka fundamentalno promenila u poslednjih nekoliko godina. Potpune usluge sada zamenjuju veliki deo programiranja niskog nivoa koje se ranije očekivalo od inženjera, koji sada koriste uslužne open source i jednostavne plug-and-play softver-kao-uslugu (SaaS, software-as-a-service) ponude. Na primer inženjeri podataka sada se fokusiraju na apstrakcije visokog nivoa ili pisanje cevovoda kao koda unutar okvira orkestracije.

Čak i u više apstraktnom svetu, najbolje prakse softverskog inženjerstva pružaju konkurentsku prednost i inženjeri podataka koji mogu da se udube u duboke arhitektonske pojedinosti koda daju svojim kompanijama prednost kada se pojave specifične tehničke potrebe. Ukratko inženjer podataka koji ne zna da piše proizvodni kôd biće ozbiljno ugrožen i ne vidimo da će se to uskoro promeniti. Inženjeri podataka ostaju softverski inženjeri, pored svojih mnogih drugih uloga.

Koje jezike inženjer podataka treba da zna? Mi delimo jezike programiranja za inženjerstvo podataka na primarne i sekundarne kategorije. U trenutku pisanja ovog teksta, primarni jezici inženjerstva podataka su SQL, Python, jezik Java Virtual Machine (JVM) obično Java ili Scala i bash:

SQL

Najčešće korišćen interfejs za baze podataka i jezera podataka (data lakes). Nakon što je na kratko bio stavljen u drugi plan zbog potrebe za pisanjem prilagođenog MapReduce koda za obradu velikih podataka, SQL (u raznim oblicima) se ponovo nametnuo kao lingua franca podataka.

Python

Jezik koji služi kao most između inženjerstva podataka i nauke o podacima. Sve veći broj alata za inženjerstvo podataka je napisan na Pythonu ili ima Python API-je. Poznat je kao „drugi najbolji jezik za sve". Python leži u osnovi popularnih alata za podatke kao što su pandas, NumPy, Airflow, sci-kit learn, TensorFlow, PyTorch i PySpark. Python je lepak između osnovnih komponenti i često je jezik prvog reda za API interakciju sa nekim okvirom.

JVM jezici kao što su Java i Scala

Prevlađujući za Apache open source projekte poput Spark, Hive i Druid. JVM je generalno boljih performansi od Pythona i može pružiti pristup funkcijama nižeg nivoa nego Python API (na primer, to je slučaj za Apache Spark i Beam). Razumevanje Jave ili Scale će biti korisno ako koristite popularni open source okvir za podatke.

bash

Komandno-linijski interfejs (CLI, command-line interface) za Linux operativne sisteme. Poznavanje bash komandi i osećaj udobnosti pri korišćenju CLI-ja znatno će povećati vašu produktivnost i tok rada kada treba da pišete skripte ili obavljate operacije na operativnom sistemu. Čak i danas inženjeri podataka često koriste komandno-linijske alate poput awk ili sed za obradu datoteka u cevovodu (pipline) podataka u ili pozivaju bash komande iz okvira za orkestraciju. Ako koristite Windows, slobodno koristite PowerShell umesto bash.

Nerazumna efikasnost SQL-a

Pojavom MapReduce i ere velikih podataka, SQL je bio proglašen zastarelim. Od tada, razni razvoji značajno su povećali korisnost SQL-a u životnom ciklusu inženjerstva podataka. Spark SQL, Google BigQuery, Snowflake, Hive i mnogi drugi alati za podatke mogu obraditi ogromne količine podataka koristeći deklarativne, set-teorijske SQL semantike. SQL je podržan i od mnogih okvira za striming, kao što su Apache Flink, Beam i Kafka. Verujemo da kompetentni inženjeri podataka treba da budu visoko stručni u SQL-u.

Da li kažemo da je SQL jezik koji može sve? Nipošto. SQL je moćan alat koji može brzo rešiti kompleksne analitičke i probleme transformacije podataka. S obzirom da je vreme primarno ograničenje za protok inženjerskog tima za podatake, trebalo bi da inženjeri prihvataju alate koji kombinuju jednostavnost i visoku produktivnost. Inženjeri podataka rade dobro kad razviju stručnost u komponovanju SQL-a sa drugim operacijama, bilo unutar okvira poput Spark i Flink ili korišćenjem orkestracije za kombinovanje više alata. Inženjeri podataka trebalo bi da uče savremene SQL semantike za bavljenje JavaScript Object Notation (JSON) parsiranjem i ugnežđenim podacima i da razmotre korišćenje SQL okvira za upravljanje kao što je dbt (Data Build Tool).

Iskusan inženjer podataka prepoznaje kada SQL nije odgovarajući alat za posao i može izabrati i kodirati u odgovarajućoj alternativi. Ekspert za SQL bi verovatno mogao napisati upit za izolovanje i tokenizaciju sirovog teksta u obradi govornog jezika (NLP, natural language processing) u cevovodu ali bi prepoznao da je kodiranje u nativnom Sparku daleko superiorna alternativa ovom mazohističkom vežbanju.

Inženjeri podataka možda će trebati da razviju veštine u sekundarnim programskim jezicima, uključujući R, JavaScript, Go, Rust, C/C++, C# i Julia. Razvoj na ovim jezicima često je neophodan kada su popularni unutar kompanije ili se koriste sa specifičnim alatima za podatke. Na primer, JavaScript se pokazao popularnim kao jezik za korisnički definisane funkcije u klaud skladištima podataka. Istovremeno, C# i PowerShell su suštinski u kompanijama koje koriste Azure i Microsoft ekosistem.

Držati korak u polju koje se brzo menja

Jednom kada te nova tehnologija pregazi, ako nisi deo valjka, onda si deo puta.

– Stewart Brand

Kako održati svoje veštine na visini u brzo promenljivom polju poput inženjerstva podataka? Da li se fokusirati na najnovije alate ili dubinsko proučavanje osnova? Evo našeg saveta: usredsredite se na osnovne principe da biste razumeli šta se neće menjati; obratite pažnju na tekući razvoj da biste znali u kom smeru se polje kreće. Nove paradigme i prakse se uvode sve vreme i na vama je da ostanete u toku. Težite da razumete kako nove tehnologije mogu biti korisne u životnom ciklusu.

Kontinuum uloga inženjera podataka, od A do B

Iako opisi poslova inženjera podataka prikazuju kao „jedinstvenog" koji mora posedovati svaku zamisivu veštinu u vezi sa podacima, svi inženjeri podataka ne obavljaju iste tipove posla niti imaju identičan skup veština. Zrelost podataka je koristan vodič za razumevanje vrsta izazova s podacima s kojima će se kompanija susresti kako razvija svoje sposobnosti u vezi sa podacima. Korisno je pogledati neke kritične razlike u vrstama poslova koje inženjeri podataka obavljaju. Iako su ove razlike pojednostavljene, one razjašnjavaju šta rade naučnici podataka odnosno inženjeri podataka i izbegavaju guranje bilo koje uloge u kategoriju jedinstvenog.

U nauci o podacima postoji pojam tipa A i tipa B naučnika podataka.[10] *Naučnici podataka tipa A* – gde *A* stoji za *analizu* – fokusiraju se na razumevanje i izvođenje uvida iz podataka. *Naučnici podataka tipa B* – gde *B* stoji za *izgradnju* (build) – deli slično pozadinsko znanje kao naučnici podataka tipa A i poseduju dobre programerske veštine. Naučnik podataka tipa B gradi sisteme koji omogućavaju rad nauci podataka u proizvodnji. Pozajmljujući iz ovog kontinuuma naučnika podataka, stvorićemo sličnu razliku za dva tipa inženjera podataka:

[10] Robert Chang, „Doing Data Science at Twitter", *Medium*, 20. jun 2015, *https://oreil.ly/xqjAx*.

Inženjeri podataka tipa A

A stoji za *apstrakciju*. U ovom slučaju inženjer podataka izbegava nekreativno teške poslove, čuvajući arhitekturu podataka što je moguće više apstraktnom i jednostavnom i ne izmišljajući točak iznova. Inženjeri podataka tipa A upravljaju životnim ciklusom inženjerstva podataka uglavnom koristeći potpuno gotove proizvode, uslužne servise i alate. Inženjeri podataka tipa A rade u kompanijama različitih industrija i na svim nivoima zrelosti podataka.

Inženjeri podataka tipa B

B potiče od *build* (izgradnja). Inženjeri podataka tipa B izrađuju alate i sisteme za podatke koji skaliraju i koriste ključnu kompetenciju i konkurentsku prednost kompanije. U opsegu zrelosti podataka inženjeri podataka tipa B češće se nalaze u kompanijama koje su na drugom i trećem stepenu (skaliranje i vođenje podacima) ili kada je početni slučaj korišćenja podataka toliko jedinstven i ključan za misiju da su potrebni prilagođeni alati za podatke da se započne.

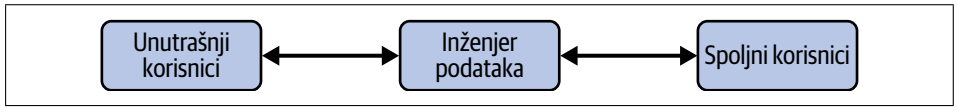
Inženjeri podataka tipa A i tipa B mogu raditi u istoj kompaniji i mogu čak biti ista osoba! Češće se prvo angažuje inženjer podataka tipa A da bi postavio temelj, dok se veštine inženjera podataka tipa B ili uče ili se angažuju kako se potreba pojavi unutar kompanije.

Inženjeri podataka unutar organizacije

Inženjeri podataka ne rade u vakuumu. U zavisnosti od toga na čemu rade interaguju s tehničkim i netehničkim osobljem i okreću se u različitim smerovima (interno i eksterno). Hajde da istražimo šta inženjeri podataka rade unutar organizacije i sa kim su u interakciji.

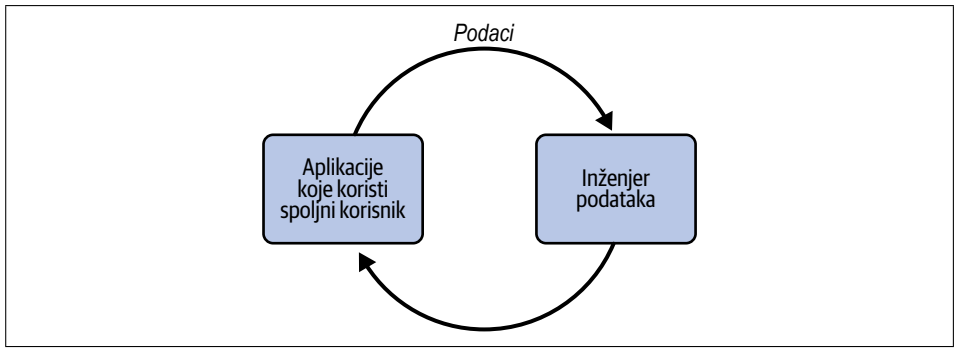
Inženjeri podataka okrenuti prema unutra u odnosu na one okrenute prema spolja

Inženjer podataka služi više krajnjih korisnika i okreće se u mnogim internim i eksternim smerovima (Slika 1-9). Pošto poslovi i odgovornosti inženjera podataka nisu isti, važno je razumeti kome inženjer podataka služi. U zavisnosti od krajnjih slučajeva upotrebe, primarne odgovornosti inženjera podataka su uperene prema spolja, prema unutra ili su kombinacija ova dva.



Slika 1-9. Pravci u kojima inženjer podataka može biti usmeren

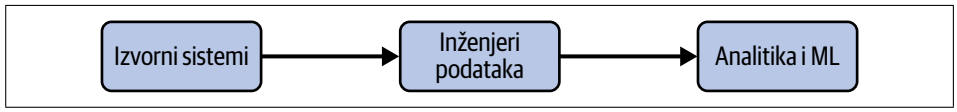
Jedan *inženjer podataka okrenut prema spolja* tipično se usklađuje sa korisnicima aplikacija okrenutih prema spolja, kao što su društvene mreže, uređaji Internet of Things (IoT) i e-trgovinske platforme. Ovaj inženjer podataka projektuje, izrađuje i upravlja sistemima koji prikupljaju, skladište i obrađuju transakcijske i događajne podatke iz ovih aplikacija. Sistemi koje grade ovi inženjeri imaju povratnu petlju od aplikacije do cevovoda za podatke i zatim nazad do aplikacije (Slika 1-10).



Slika 1-10. Sistemi inženjera podataka okrenuti prema spolja

Inženjeri za obradu podataka koji se fokusiraju na spoljašnje sisteme suočeni su sa jedinstvenim skupom problema. Spoljašnji sistem za obradu upita često rukuje sa znatno većim brojem istovremenih opterećenja u odnosu na sisteme usmerene ka unutrašnjim potrebama. Inženjeri moraju razmotriti postavljanje striktnih ograničenja na upite koje korisnici mogu da pokrenu da bi ograničili uticaj pojedinačnog korisnika na infrastrukturu. Pored toga, bezbednost je mnogo složeniji i osetljiviji problem za spoljašnje upite, posebno ako su podaci koji se pretražuju multitenant (podaci od mnogih klijenata smešteni u jednoj tabeli).

Inženjer za obradu podataka sa unutrašnjim fokusom obično se fokusira na aktivnosti od suštinskog značaja za potrebe poslovanja i unutrašnje zainteresovane strane (Slika 1-11). Primeri uključuju kreiranje i održavanje protoka podataka, cevovoda (pipeline) i skladišta podataka za BI kontrolne table (dashobards) izveštaje, poslovne procese, nauku o podacima i modele mašinskog učenja.



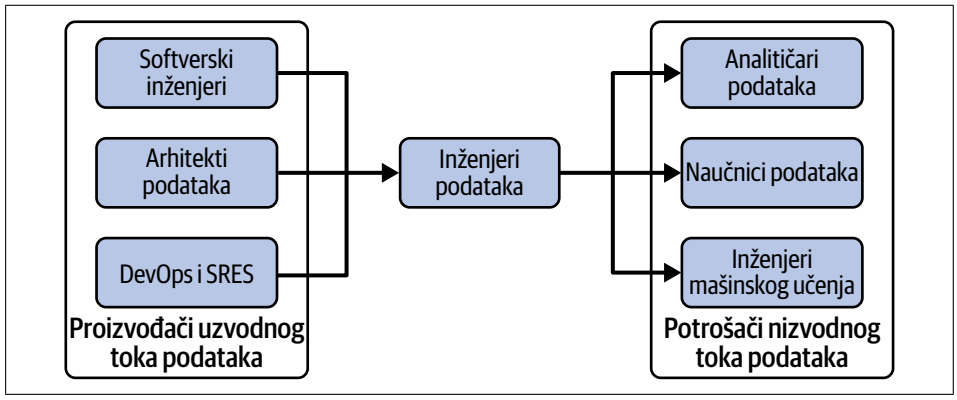
Slika 1-11. Inženjer za obradu podataka sa unutrašnjim fokusom

Odgovornosti usmerene ka spoljašnosti i unutrašnjosti često su mešane. U praksi, podaci sa unutrašnjim fokusom obično su preduslov za podatke usmerene ka spoljašnjosti. Inženjer za obradu podataka ima dva skupa korisnika sa vrlo različitim zahtevima za istovremenošću obrade upita, bezbednosti i još toga.

Inženjeri za obradu podataka i ostale tehničke uloge

U praksi, životni ciklus inženjerstva podataka se preseca sa mnogim domenima odgovornosti. Inženjeri podataka nalaze se u centru različitih uloga, direktno ili preko menadžera interagujući sa mnogim organizacionim jedinicama.

Pogledajmo ko bi mogao biti pod uticajem inženjera za obradu podataka. U ovoj sekciji, razmotrićemo tehničke uloge povezane sa inženjerstvom podataka (Slika 1-12).



Slika 1-12. Ključni tehnički zainteresovani subjekti inženjerstva podataka

Inženjer podataka je čvorište između *proizvođača podataka*, kao što su softverski inženjeri, arhitekti podataka i inženjeri za DevOps ili pouzdanost sistema (SRE, site-reliability engineers) i *potrošača podataka*, kao što su analitičari podataka, naučnici koji se bave podacima i inženjeri za mašinsko učenje. Pored toga inženjeri za obradu podataka će interagovati sa onima koji se nalaze u operativnim ulogama, poput DevOps inženjera.

S obzirom na tempo kojim se nove uloge u podacima pojavljuju na sceni (npr. inženjeri za analizu i mašinsko učenje), ovo ni izdaleka nije iscrpan spisak.

Subjekti zainteresovani za uzvodni tok podataka

Da biste bili uspešni kao inženjer podataka, potrebno je da razumete arhitekturu podataka koju koristite ili dizajnirate, kao i izvorne sisteme koji proizvode podatke koje ćete obrađivati. Dalje, govorićemo o nekoliko poznatih strana zainteresovanih za uzvodni tok podataka: arhitekti podataka, softverski inženjeri i DevOps inženjeri.

Arhitekti podataka. Arhitekti podataka (data architects) funkcionišu na nivou apstrakcije koji je jedan korak udaljen od inženjera za obradu podataka. Arhitekti podataka projektuju planove za upravljanje podacima organizacije, mapirajući

procese i opštu arhitekturu i sisteme podataka.[11] Oni služe kao most između tehničke i netehničke strane organizacije. Uspešni arhitekti podataka uglavnom imaju „ožiljke iz borbe" proizašle iz širokog inženjerskog iskustva, omogućavajući im da vode i asistiraju inženjerima dok uspešno komuniciraju inženjerske izazove netehničkim poslovnim zainteresovanim stranama.

Arhitekti podataka sprovode politike za upravljanje podacima preko silosa (izolovane kolekcije podataka koje nisu lako dostupne drugim delovima organizacije) i poslovnih jedinica, usmeravaju globalne strategije kao što su upravljanje podacima (data management) i upravljanje upotrebom podataka (data governance) i usmeravaju značajne inicijative. Arhitekti podataka često igraju centralnu ulogu u migracijama u oblak i dizajniranju sistema u oblaku od nule.

Pojava oblaka pomakla je granicu između arhitekture podataka i inženjerstva podataka. Arhitekture podataka u oblaku su mnogo fluidnije od sistema na licu mesta, tako da su odluke o arhitekturi koje su tradicionalno uključivale opsežna istraživanja, duge pripreme, ugovore o nabavci i instaliranje hardvera, sada često donesene tokom procesa implementacije, kao samo jedan korak u široj strategiji. Bez obzira na to, arhitekti podataka će biti uticajni vizionari u preduzećima, radeći ruku pod ruku sa inženjerima za obradu podataka da bi odredili veliku sliku arhitekture i strategija podataka.

U zavisnosti od zrelosti i obima podataka kompanije inženjer podataka može se preklapati sa ili preuzimati odgovornosti arhitekte podataka. Stoga inženjer za obradu podataka treba da ima dobro razumevanje najboljih praksi arhitekture i načina pristupa.

Imajte na umu da smo arhitekte podataka smestili u sekciju *uzvodni zainteresovani subjekti*. Arhitekti podataka često pomažu u dizajniranju slojeva podataka aplikacija koji su izvorni sistemi za inženjere za obradu podataka. Arhitekti mogu interagovati sa inženjerima za obradu podataka u različitim drugim fazama životnog ciklusa obrade podataka. Pokrivamo „dobru" arhitekturu podataka u Poglavlju 3.

Softverski inženjeri. Inženjeri softvera (software engineers) grade softvere i sisteme koji pokreću poslovanje; oni su u velikoj meri odgovorni za generisanje *internih podataka* koje će inženjeri podataka konzumirati i procesuirati. Sistemi koje izrađuju inženjeri softvera obično generišu podatke o događajima u aplikacijama i logove, koji su sami po sebi značajna sredstva. Ovi interni podaci se razlikuju od *eksternih podataka* dobijenih iz SaaS platformi ili partnera u poslovanju. U tehnički dobro vođenim organizacijama inženjeri softvera i inženjeri podataka koordinišu se od početka novog projekta da bi osmislili podatke aplikacija za konzumiranje u analitičkim i ML (machine learning) aplikacijama.

[11] Paramita (Guha) Ghosh, „Data Architect vs. Data Engineer", Dataversity, 12. novembar 2021, *https://oreil.ly/TlyZY*.

Trebalo bi da inženjer podataka radi zajedno sa inženjerima softvera da bi razumeo aplikacije koje generišu podatke, obime, frekvenciju i format generisanih podataka, kao i sve ostalo što će uticati na životni ciklus inženjerstva podataka, poput bezbednosti podataka i regulative o usaglašenosti. Na primer, to bi moglo da znači postavljanje inicijalnih očekivanja o tome šta inženjeri softvera treba da urade da bi obavljali svoje poslove. Inženjeri podataka moraju blisko sarađivati s inženjerima softvera.

Inženjeri DevOps i inženjeri pouzdanosti sajta. DevOps i SRE-ovi često proizvode podatke kroz operativni monitoring. Klasifikujemo ih kao uzvodne u odnosu na inženjere podataka, ali oni mogu biti i nizvodni, konzumirajući podatke kroz kontrolne table ili interagujući direktno sa inženjerima podataka u koordinaciji operacija sa sistemima podataka.

Nizvodni korisnici

Inženjerstvo podataka postoji da bi služilo nizvodnim (downstream) korisnicima podataka i slučajevima upotrebe. Ovaj odeljak diskutuje kako inženjeri podataka interaguju s različitim nizvodnim ulogama. Upoznaćemo se i sa nekoliko modela usluga, uključujući centralizovane timove inženjerstva podataka i interdisciplinarne timove.

Naučnici podataka. Naučnici podataka ili data naučnici (data scientist) grade modele koji gledaju u budućnost da bi pravili predviđanja i preporuke. Ovi modeli se zatim evaluiraju na živim podacima da bi pružili vrednost na različite načine. Na primer, bodovanje modela može odrediti automatizovane akcije kao odgovor na uslove u realnom vremenu, preporučiti proizvode kupcima na osnovu istorije pretrage u njihovoj trenutnoj sesiji ili praviti trenutna ekonomska predviđanja koja koriste trgovci.

Prema uobičajenim industrijskim anegdotama, data naučnici provedu 70% do 80% svog vremena prikupljajući, čisteći i pripremajući podatke.[12] Prema našem iskustvu, ovi brojevi često odražavaju nezrele prakse nauke podataka i inženjerstva podataka. Posebno, mnogi popularni okviri nauke podataka mogu postati uska grla ako se ne skaliraju adekvatno. Data naučnici koji rade isključivo na jednoj radnoj stanici primorani su da smanjuju uzorak podataka, čime značajno komplikuju pripremu podataka i potencijalno kompromituju kvalitet modela koje proizvode. Štaviše, lokalno razvijeni kodovi i okruženja često su teški za implementaciju u proizvodnji, a nedostatak automatizacije znatno otežava radne

[12] Postoji različita referentna literatura za ovaj koncept. Iako je ova fraza opštepoznata, pojavila se zdrava rasprava oko njene validnosti u različitim praktičnim okruženjima. Za više detalja, pogledajte Leigh Dodds, „Do Data Scientists Spend 80% of Their Time Cleaning Data? Turns Out, No?“ Lost Boy blog, 31. januar 2020, *https://oreil.ly/szFww*; i Alex Woodie, „Data Prep Still Dominates Data Scientists' Time, Survey Finds“, *Datanami*, 6. jul 2020, *https://oreil.ly/jDVWF*.

procese naučnika podataka. Ako inženjeri podataka obave svoj posao i uspešno sarađuju, data naučnici ne treba da troše svoje vreme prikupljajući, čisteći i pripremajući podatke nakon početne istraživačke faze. Trebalo bi da inženjeri podataka automatizuju ovaj posao što je više moguće.

Potrebom za proizvodno spremnom naukom podataka značajno se pokreće pojava profesije inženjerstva podataka. Inženjeri podataka treba da pomognu naučnicima podataka da omoguće put do proizvodnje. Zapravo, mi (autori) prešli smo iz nauke o podacima u inženjerstvo podataka nakon što smo prepoznali ovu osnovnu potrebu. Inženjeri podataka rade na pružanju automatizacije podataka i skaliranosti koje čine nauku podataka efikasnijom.

Analitičari podataka. Analitičari podataka ili data analitičari (data analysts) ili poslovni analitičari teže da razumeju poslovne učinke i trendove. Dok su naučnici podataka okrenuti ka budućnosti, analitičar podataka se tipično fokusira na prošlost ili sadašnjost. Analitičari podataka obično izvršavaju SQL upite u skladištu podataka ili u jezeru podataka. Mogu koristiti tabele za proračun i analizu i različite BI alate kao što su Microsoft Power BI, Looker ili Tableau. Data analitičari su stručnjaci u domenu podataka s kojima često rade i postaju blisko upoznati sa definicijama podataka, karakteristikama i problemima kvaliteta. Uobičajeni nizvodni potrošači data analitičara su poslovni korisnici, menadžment i izvršni direktori.

Inženjeri podataka rade sa analitičarima podataka na izgradnji cevovoda za nove izvore podataka potrebne poslovanju. Stručnost data analitičara u domenu je izuzetno vredna u poboljšanju kvaliteta podataka i oni često sarađuju sa inženjerima podataka u ovom kapacitetu.

Inženjeri mašinskog učenja i istraživači veštačke inteligencije. Inženjeri mašinskog učenja (ML inženjeri) preklapaju se sa inženjerima podataka i naučnicima podataka. ML inženjeri razvijaju napredne ML tehnike, obučavaju modele i dizajniraju i održavaju infrastrukturu koja pokreće ML procese u okruženju proizvodnog razmera. ML inženjeri često imaju napredno radno znanje o ML i tehnikama dubokog učenja, kao i okvirima poput PyTorch ili TensorFlow.

ML inženjeri razumeju hardver, usluge i sisteme potrebne za pokretanje ovih okvira, kako za obuku modela tako i za njihovo implementiranje u okruženju proizvodnje. Uobičajeno je da ML protokoli rade u okruženju u oblaku gde ML inženjeri mogu po potrebi brzo da pokreću i skaliraju infrastrukturne resurse ili da se oslone na upravljane servise.

Kao što smo spomenuli, granice između ML inženjerstva inženjerstva podataka i nauke o podacima su nejasne. Inženjeri podataka mogu imati neke operativne odgovornosti nad ML sistemima, a naučnici podataka mogu tesno sarađivati sa ML inženjerstvom u dizajniranju naprednih ML procesa.

Svet ML inženjerstva se ubrzano širi i paralelno se razvija mnogo istih trendova koji se dešavaju i u inženjerstvu podataka. Dok je pre nekoliko godina pažnja na ML bila fokusirana na to kako graditi modele, ML inženjerstvo sada sve više naglašava uključivanje najboljih praksi iz domena mašinskog učenja operacija (MLOps, machine learning operations) i drugih zrelih praksi koje su prethodno usvojene u softverskom inženjerstvu i DevOps-u.

Istraživači veštačke inteligencije (AI) rade na novim, naprednim ML tehnikama. Istraživači AI mogu raditi unutar velikih tehnoloških kompanija, specijalizovanih startapova za intelektualno vlasništvo (OpenAI, DeepMind) ili akademskih institucija. Neki praktičari su posvećeni istraživanju kao dodatnom radu u kombinaciji sa obavezama ML inženjerstva unutar kompanije. Oni koji rade unutar specijalizovanih ML laboratorija često su 100% posvećeni istraživanju. Istraživački problemi mogu biti usmereni na neposredne praktične primene ili na više apstraktne demonstracije veštačke inteligencije. DALL-E, Gato AI, AlphaGo i GPT-3/GPT-4 su odlični primeri projekata istraživanja ML. S obzirom na tempo napretka u ML, ovi primeri će najverovatnije biti zastareli za par godina. Naveli smo neke reference u „Dodatnim izvorima“ (strana 33).

Istraživači AI u dobro finansiranim organizacijama su visoko specijalizovani i rade sa timovima inženjera koji olakšavaju njihov rad. Akademski ML inženjeri obično imaju manje resursa ali se oslanjaju na timove studenata diplomaca, postdoktoranata i univerzitetskog osoblja za pružanje inženjerske podrške. ML inženjeri koji su delimično posvećeni istraživanju često se oslanjaju na iste timove za podršku istraživanju i proizvodnji.

Inženjeri podataka i poslovno liderstvo

Govorili smo o tehničkim ulogama sa kojima inženjer podataka komunicira. Ali inženjeri podataka deluju šire kao organizacijski konektori, često u netehničkom kapacitetu. Poslovanje se sve više oslanja na podatke kao ključni deo mnogih proizvoda ili kao proizvod sam po sebi. Inženjeri podataka sada učestvuju u strateškom planiranju i vode ključne inicijative koje se protežu izvan granica IT-a. Inženjeri podataka često podržavaju arhitekte podataka tako što služe kao lepak između poslovanja i nauke o podacima/analitike.

Podaci koje koriste izvršni direktori

Izvršni direktori (C suite) sve više se uključuju u podatke i analitiku, jer ih prepoznaju kao značajnu vrednost za savremene preduzetnike. Na primer, generalni direktori sada brinu o inicijativama koje su nekad bile isključiva domena IT-a, kao što su migracije u oblak ili uvođenje novih platformi za podatke o klijentima.

Generalni direktor. Generalni direktori (CEO, chief executive officers) u netehnološkim kompanijama generalno se ne bave sitnicama oko okvira podataka i softvera.

Umesto toga, oni definišu viziju u saradnji sa tehničkim rukovodstvom i rukovodstvom podacima kompanije. Inženjeri podataka pružaju uvid u ono što je moguće sa podacima. Inženjeri podataka i njihovi menadžeri održavaju mapu podataka dostupnih organizaciji – kako interno tako i trećih strana – i u kojem vremenskom okviru. Zaduženi su i za proučavanje primarnih izmena arhitekture podataka u saradnji sa drugim inženjerskim ulogama. Na primer inženjeri podataka su često intenzivno uključeni u migracije u oblak, migracije na nove sisteme podataka ili uvođenje tehnologija strimovanja.

Direktor za informacione tehnologije. Direktor informacionih tehnologija (CIO, chief information officer) je viši upravni izvršilac odgovoran za informacionu tehnologiju unutar organizacije; to je uloga usmerena ka unutra. CIO mora posedovati duboko znanje o informacionoj tehnologiji i poslovnim procesima – samo jedno ili drugo nije dovoljno. CIO-i upravljaju organizacijom informacione tehnologije, postavljajući tekuće politike, ali definišu i izvršavaju značajne inicijative pod upravom CEO-a.

CIO često sarađuje sa liderima inženjerstva podataka u organizacijama sa dobro razvijenom kulturom podataka. Ako organizacija nije posebno napredna u smislu zrelosti podataka, CIO će pomoći u oblikovanju njene kulture podataka. CIO će raditi sa inženjerima i arhitektama da bi napravio mapu glavnih inicijativa i donosio strateške odluke o usvajanju glavnih arhitektonskih elemenata, kao što su sistemi za planiranje resursa preduzeća (ERP, enterprise resource planning) i sistemi za upravljanje odnosima sa kupcima (CRM, customer relationship management), migracije u oblak, sistemi podataka i IT usmeren ka internim potrebama.

Direktor za tehnologiju. Direktor za tehnologiju (CTO, chief technology officer) je sličan CIO-u, ali je okrenut ka spoljnim poslovima. CTO poseduje ključnu tehnološku strategiju i arhitekture za aplikacije usmerene ka korisnicima, kao što su mobilne aplikacije, veb aplikacije i IoT – sve su kritični izvori podataka za inženjere podataka. CTO je verovatno vešt sa tehnologijom i ima dobro razumevanje osnova softverskog inženjerstva i arhitekture sistema. U nekim organizacijama bez CIO-a, CTO a ponekad operativni direktor (COO, chief operating officer) preuzima ulogu CIO-a. Inženjeri podataka često direktno ili indirektno izveštavaju CTO-a.

Direktor za podatke. Direktor za podatke (CDO, chief data officer) je funkcija koja je prvi put stvorena 2002. godine u kompaniji Capital One kao priznanje rastućeg značaja podataka kao poslovnog resursa. CDO je odgovoran za data imovinu i strategiju kompanije. CDO se fokusira na poslovnu upotrebljivost podataka, ali trebalo bi da ima čvrsto tehničko znanje. CDO nadgleda data proizvode, strategiju inicijative i osnovne funkcije kao što su upravljanje glavnim podacima i privatnost. Povremeno, CDO upravlja poslovnom analitikom i inženjerstvom podataka.

Direktor analitike. Direktor analitike (CAO, chief analytics officer) je varijanta uloge CDO-a. Tamo gde oba položaja postoje, CDO se fokusira na tehnologiju i organizaciju potrebnu za dostavljanje podataka. CAO je odgovoran za analitičku strategiju i donošenje odluka za poslovanje. CAO može nadgledati nauku o podacima i mašinsko učenje (ML) iako ovo u velikoj meri zavisi od toga da li kompanija ima CDO ili CTO ulogu.

Direktor za algoritme. Direktor za algoritme (CAO-2, chief algorithms officer) je nedavna inovacija među C-ovima, veoma tehnička uloga fokusirana specifično na nauku o podacima i ML. CAO-2 obično ima iskustvo kao pojedinačni saradnici i vođe timova u projektima nauke o podacima ili ML. Često imaju profesionalno iskustvo u istraživanju ML-a i odgovarajući napredni stepena obrazovanja.

Od CAO-2 se očekuje da budu upućeni u aktuelna istraživanja ML-a i da imaju duboko tehničko znanje o ML inicijativama njihove kompanije. Pored stvaranja poslovnih inicijativa, oni pružaju tehničko vođstvo, postavljaju agende za istraživanje i razvoj i formiraju istraživačke timove.

Inženjeri podataka i menadžeri projekta

Inženjeri podataka (data engeneers) često rade na značajnim inicijativama, koje se protežu na više godina. Dok pišemo ovu knjigu, mnogi inženjeri podataka rade na migracijama u oblak, premeštajući cevovode i skladišta podataka na sledeću generaciju alata za podatke. Ostali inženjeri podataka započinju projekte od nule, praveći nove arhitekture podataka od početka birajući između brojnih najboljih mogućih arhitektonskih opcija i alata.

Ove velike inicijative često imaju koristi od *upravljanja projektima* (za razliku od upravljanja proizvodima, koje će biti opisano dalje). Dok inženjeri podataka funkcionišu u kapacitetu infrastrukture i isporuke usluga, projektni menadžeri usmeravaju saobraćaj i služe kao kontrolori. Većina projektnih menadžera radi prema nekoj varijanti Agila i Scruma, sa povremenim pojavljivanjem Vodopada (Waterfall metodologije). Poslovanje nikada ne spava i poslovni deoničari često imaju značajnu listu stvari koje žele da adresiraju i novih inicijativa koje žele da pokrenu. Projektni menadžeri moraju filtrirati dugu listu zahteva i prioritetizirati ključne isporučive stavke da bi projekti ostali na pravom putu i bolje služili kompaniji.

Inženjeri podataka stupaju u interakciju s projektnim menadžerima, često planirajući sprintove (vremenski period u Scrum metodologiji) za projekte i prateće kratke sastanke (stand-ups) u vezi sa sprintom. Povratne informacije idu u oba smera, sa inženjerima podataka koji informišu projektnog menadžera i ostale zainteresovane strane o napretku i preprekama, kao i projektni menadžeri koji usklađuju ritam tehnoloških timova sa stalno promenljivim potrebama poslovanja.

Inženjeri podataka i menadžeri proizvoda

Menadžeri proizvoda nadgledaju razvoj proizvoda, često posedujući linije proizvoda. U kontekstu inženjera podataka, ti proizvodi se nazivaju *proizvodi podataka*. Proizvodi podataka se ili grade od temelja ili su inkrementalna poboljšanja postojećih proizvoda. Inženjeri podataka interaguju češće sa *menadžerima proizvoda* (product managers) kako je korporativni svet usvojio fokus na podatke. Kao i projektni menadžeri, menadžeri proizvoda usklađuju delovanje tehnoloških timova sa potrebama kupaca i poslovanja.

Inženjeri podataka i druge menadžerske uloge

Inženjeri podataka komuniciraju sa raznim menadžerima osim menadžera projekta i proizvoda. Međutim, ove interakcije obično prate ili model usluge (usluge za analizu podataka ili pristup podacima drugim timovima unutar organizacije) ili model prekograničnih funkcija (saradnja između različitih funkcija radi postizanja ciljeva projekta ili inicijative). Inženjeri podataka ili služe različitim dolazećim zahtevima kao centralizovani tim ili rade kao resurs dodeljen određenom menadžeru, projektu ili proizvodu.

Za više informacija o timovima podataka i kako ih strukturirati, preporučujemo knjigu Džona Tompsona *Building Analytics Teams* (Packt) i knjigu Džesija Andersona *Data Teams* (Apress). Obe knjige pružaju čvrste okvire i perspektive o ulogama izvršnih direktora za podatke, koga zaposliti i kako izgraditi najefikasniji tim podataka za vašu kompaniju.

Kompanije ne zapošljavaju inženjere samo zbog koda u izolaciji. Da bi bili dostojni svog naziva inženjeri bi trebalo da razviju duboko razumevanje problema koji treba rešiti, tehnoloških alata koji su im na raspolaganju, kao i ljudi s kojima rade i koje služe.

Zaključak

Ovo poglavlje vam je pružilo kratak pregled prostora inženjerstva podataka, uključujući sledeće:

- Definisanje inženjerstva podataka i opisivanje šta inženjeri podataka rade
- Opisivanje vrsta zrelosti podataka u kompaniji
- Tip A i tip B inženjera podataka
- S kim inženjeri podataka rade

Nadamo se da je ovo prvo poglavlje probudilo vaše interesovanje, bilo da ste praktičar u razvoju softvera, naučnik podataka inženjer mašinskog učenja, poslovni akter, preduzetnik ili kapitalista za rizična ulaganja. Naravno, mnogo toga još uvek treba pojasniti u sledećim poglavljima. Poglavlje 2 pokriva životni ciklus

inženjerstva podataka, a zatim arhitekturu u poglavlju 3. Sledeća poglavlja ulaze u srž tehnoloških odluka za svaki deo životnog ciklusa. Celokupno polje podataka je u pokretu i koliko god je to moguće, svako poglavlje se fokusira na *neizmenljive* – perspektive koje će ostati važeće mnogo godina usred neprestanih promena.

Dodatni izvori

- „The AI Hierarchy of Needs", Monica Rogati
- Veb stranica istraživanja AlphaGo
- „Big Data Will Be Dead in Five Years", Lewis Gavin
- *Building Analytics Teams,* John K. Thompson (Packt)
- *What Is Data Engineering?,* Lewis Gavin, Poglavlje 1 knjige (O'Reilly)
- „Data as a Product vs. Data as a Service", Justin Gage
- „Data Engineering: A Quick and Simple Definition", James Furbush (O'Reilly)
- *Data Teams,* Jesse Anderson (Apress)
- „Doing Data Science at Twitter", Robert Chang
- „The Downfall of the Data Engineer", Maxime Beauchemin
- „The Future of Data Engineering Is the Convergence of Disciplines", Liam Hausmann
- „How CEOs Can Lead a Data-Driven Culture", Thomas H. Davenport i Nitin Mittal
- „How Creating a Data-Driven Culture Can Drive Success", Frederik Bussler
- Web stranica Body of Knowledge o upravljanju informacijama, Wikipedia
- „Information Management" Wikipedia
- „On Complexity in Big Data", Jesse Anderson (O'Reilly)
- „OpenAI's New Language Generator GPT-3 Is Shockingly Good – and Completely Mindless", Will Douglas Heaven
- „The Rise of the Data Engineer", Maxime Beauchemin
- „A Short History of Big Data", Mark van Rijmenam
- „Skills of the Data Architect", Bob Lambert
- „The Three Levels of Data Analysis: A Framework for Assessing Data Organization Maturity", Emilie Schario
- „What Is a Data Architect? IT's Data Framework Visionary", Thor Olavsrud
- „Which Profession Is More Complex to Become, a Data Engineer or a Data Scientist?" tema na stranici Quora
- „Why CEOs Must Lead Big Data Initiatives", John Weathington