Sadržaj

Predgovor ..... **xi**

1. Pregled sistema mašinskog učenja **1**
Kada koristiti mašinsko učenje 3
Primeri upotrebe mašinskog učenja 9
Razumevanje sistema mašinskog učenja 12
Mašinsko učenje u istraživanju u odnosu na primenu 12
Sistemi mašinskog učenja nasuprot tradicionalnog softvera 22
Rezime 24

2. Uvod u projektovanje sistema mašinskog učenja **25**
Poslovni i ML ciljevi 26
Zahtevi za ML sisteme 28
Pouzdanost 29
Skalabilnost 29
Održivost 31
Prilagodljivost 31
Iterativan proces 32
Formulisanje problema mašinskog učenja 34
Vrste ML zadataka 35
Ciljne funkcije 39
Um protiv podataka 42
Rezime 45

3. Osnove inženjeringa podataka **47**
Izvori podataka 48
Formati podataka 51
JSON 52
Format po redovima nasuprot formata po kolonama 52
Tekstualni nasuprot binarnom formatu 55
Modeli podataka 56
Relacioni model 57
NoSQL 61
Strukturisani nasuprot nestrukturisanih podataka 64

Mašine za skladištenje podataka i obradu 65
Transakciona i analitička obrada 66
ETL: ekstrakcija, transformacija, učitavanje 68
Režimi protoka podataka 70
Prenos podataka kroz baze podataka 70
Prenos podataka kroz servise 71
Prenos podataka transportom u realnom vremenu 72
Paketna obrada nasuprot obrade u toku 76
Rezime 78

4. Podaci za obuku 81
Uzorkovanje 82
Uzorkovanje bez verovatnoće 83
Jednostavno nasumično uzorkovanje (Simple Random Sampling) 84
Slojevito uzorkovanje (Stratified Sampling) 84
Težinsko uzorkovanje (Weighted Sampling) 85
Uzorkovanje rezervoara (Reservoir Sampling) 86
Uzorkovanje po važnosti (Importance Sampling) 87
Označavanje (Labeling) 88
Ručne oznake 88
Prirodne oznake 91
Rešavanje nedostatka oznaka 94
Neravnoteža klasa 103
Izazovi neravnoteže klasa 103
Rešavanje problema neravnoteže klasa 105
Povećanje skupa podataka 114
Jednostavne transformacije koje čuvaju oznake 115
Perturbacija 115
Sinteza podataka 117
Rezime 119

5. Inženjering karakteristika 121
Naučene karakteristike nasuprot projektovanih karakteristika 121
Uobičajene operacije inženjeringa karakteristika 125
Rukovanje sa nedostajućim vrednostima 125
Normalizacija 128
Diskretizacija 130
Kodiranje kategorijskih karakteristika 131
Ukrštanje karakteristika 134
Diskretno i kontinuirano poziciono umetanje 135
Curenje podataka 137
Uobičajeni uzroci curenja podataka 139
Detektovanje curenja podataka 142

Inženjering dobrih karakteristika 142
Važnost karakteristika 143
Generalizacija karakteristika 146
Rezime 147

6. Razvoj modela i oflajn evaluacija 149
Razvoj i obučavanje modela 150
Evaluacija ML modela 150
Ansambli 156
Praćenje eksperimenata i verzionisanje 162
Distribuirana obuka 168
AutoML 172
Oflajn evaluacija modela 178
Osnove 179
Metode evaluacije 181
Rezime 189

7. Implementacija modela i servis za predikciju 191
Mitovi o implementaciji mašinskog učenja 193
Mit 1: Implementirate samo jedan ili dva ML modela odjednom 194
Mit 2: Ako ništa ne radimo, performanse modela ostaju iste 195
Mit 3: Nećete morati često ažurirati vaše modele 196
Mit 4: Većina ML inženjera ne mora brinuti o skaliranju 196
Paketno predviđanje nasuprot onlajn predviđanja 197
Od predviđanja u grupama ka onlajn predviđanju 201
Ujedinjavanje paketnog i strimovanog cevovoda 203
Kompresija modela 205
Faktorizacija niske rangiranosti 206
Destilacija znanja 207
Smanjenje dimenzija – orezivanje 207
Kvantizacija 208
ML na oblaku i na ograničenim resursima 211
Kompajliranje i optimizacija modela za uređaje kod korisnika 213
ML u veb čitačima 220
Rezime 222

8. Promena raspodele podataka i praćenje 225
Uzroci neuspeha ML sistema 226
Neuspesi softverskog sistema 227
Neuspesi specifični za ML 228
Promene u raspodeli podataka 236
Vrste promena u raspodeli podataka 237
Opšte promene raspodele podataka 241

Otkrivanje promena u raspodeli podataka 242
Adresiranje promene raspodele podataka 247
Praćenje i posmatranje 249
Metrike specifične za ML 251
Skup alata za praćenje 256
Posmatranje 259
Rezime 261

9. Neprekidno učenje i testiranje u proizvodnji 263

Neprekidno učenje 264
Obuka bez stanja nasuprot obuke sa stanjem 265
Zašto neprekidno učenje? 268
Izazovi neprekidnog učenja 270
Četiri faze neprekidnog učenja 275
Koliko često ažurirati modele 280
Testiranje u proizvodnji 282
Skrivena primena 283
A/B testiranje 283
Puštanje kanarinca 285
Prepletani eksperimenti 286
Banditi 288
Rezime 292

10. Infrastruktura i alati za MLOps 293

Skladištenje i računanje 297
Javni oblak nasuprot privatnih centara podataka 299
Okruženje za razvoj 302
Postavljanje okruženja za razvoj 303
Standardizacija razvojnih okruženja 306
Od razvoja do proizvodnje: kontejneri 308
Upravljanje resursima 311
Cron, planeri i orkestratori 311
Upravljanje radnim tokovima u nauci o podacima 314
ML Platforma 320
Primena modela 321
Skladište modela 321
Skladište karakteristika 326
Graditi ili kupiti 328
Rezime 330

11. Ljudski faktor u mašinskom učenju 333
Korisničko iskustvo 333
Obezbeđivanje doslednosti korisničkog iskustva 334
Borba protiv „uglavnom tačnih“ predviđanja 334
Neprimetno otkazivanje 336
Struktura tima 336
Saradnja funkcionalnih timova 337
Naučnik podataka vodi proces od početka do kraja 337
Odgovorna AI 341
Neodgovorna AI: studije slučajeva 342
Okvir za odgovornu AI 348
Rezime 355

Epilog 357

Indeks 359

Predgovor

Od kada sam držala prvi kurs o mašinskom učenju na Stanfordu 2017. godine, mnogo ljudi mi je tražilo savet kako da primene modele mašinskog učenja u svojim organizacijama. Ta pitanja mogu biti opšta, kao što su „Koji model treba da koristim?“ „Koliko često treba da obučavam svoj model?“ „Kako mogu da otkrijem promene u distribuciji podataka?“ „Kako da osiguram da su karakteristike korišćene tokom obučavanja konzistentne sa karakteristikama korišćenim tokom inferencije (predviđanja)?“

Ta pitanja mogu biti i specifična, kao na primer „Uveren sam da će prelazak sa paketnog predviđanja (batch prediction) na predviđanje u realnom vremenu (online prediction) poboljšati performanse našeg modela, ali kako da ubedim svog menadžera da mi to dozvoli?“ ili „Ja sam najstariji naučnik podataka u mojoj kompaniji i nedavno mi je povereno da postavim našu prvu platformu za mašinsko učenje; odakle da počnem?“

Moj kratki odgovor na sva ova pitanja uvek je: „To zavisi“. Moji detaljni odgovori često traži sate diskusije kako bih razumela odakle dolazi osoba koja pita, šta zaista pokušava da postigne i prednosti i mane različitih pristupa za njihov specifičan slučaj.

Sistemi mašinskog učenja su istovremeno složeni i jedinstveni. Složeni su jer se sastoje od mnogih različitih komponenti (ML algoritmi, podaci, poslovna logika, merni pokazatelji evaluacije, osnovna infrastruktura itd.) i uključuju mnogo različitih subjekata (naučnici podataka inženjeri mašinskog učenja, poslovni lideri, korisnici, pa čak i društvo u širem smislu). Sistemi mašinskog učenja su jedinstveni jer zavise od podataka, a podaci se drastično razlikuju od slučaja do slučaja.

Na primer, dve kompanije mogu biti iz iste oblasti (e-trgovina) i imati isti problem koji žele da reše pomoću mašinskog učenja (sistem preporuka), ali njihovi rezultujući ML sistemi mogu imati različitu arhitekturu modela, koristiti različite skupove karakteristika, biti evaluirani na različitim metrikama (mernim pokazateljima) i doneti različiti povraćaj investicije.

Mnogo blogova i tutorijala o mašinskom učenju fokusira se na odgovaranje na jedno specifično pitanje. Iako fokus pomaže u prenošenju poente, to može stvoriti utisak da je moguće razmatrati svako od ovih pitanja izolovano. Međutim,

promene u jednoj komponenti verovatno će uticati na druge komponente. Zato je neophodno razmatrati sistem u celini prilikom pokušaja donošenja bilo kakve projektne odluke.

Ova knjiga zauzima holistički pristup sistemima mašinskog učenja. Uzima u obzir različite komponente sistema i ciljeve različitih učesnika koji su uključeni. Sadržaj ove knjige ilustrovan je koristeći stvarne studije slučaja, mnoge od onih na kojima sam lično radila, podržan je brojnim referencama i pregledan od strane praktikanata mašinskog učenja u akademiji i industriji. Sekcije koje zahtevaju dubinsko znanje o određenoj temi – npr., obrada u paketima nasuprot obradi u stvarnom vremenu infrastruktura za skladištenje i računske operacije, odgovorna veštačka inteligencija – dalje su pregledane od strane stručnjaka čiji rad se fokusira na tu temu. Drugim rečima, ova knjiga je pokušaj da se daju nijansirani odgovori na pomenuta pitanja i više od toga.

Kada sam prvi put napisala beleške za predavanje koje su postavile temelj za ovu knjigu, mislila sam da ih pišem za svoje studente kako bih ih pripremila za zahteve njihovih budućih poslova kao naučnika podataka i inženjera mašinskog učenja. Međutim, ubrzo sam shvatila da sam i ja mnogo naučila tokom tog procesa. Početni nacrti koje sam podelila sa ranih čitaocima pokrenuli su mnogo razgovora koji su testirali moje pretpostavke, primorali me da razmotrim različite perspektive i upoznala sam se sa novim problemima i novim pristupima.

Nadam se da će se ovaj proces učenja nastaviti sada kada je knjiga u vašim rukama, jer imate iskustva i perspektive koje su jedinstvene za vas. Slobodno sa mnom podelite bilo koji komentar koji imate o ovoj knjizi, preko MLOps Discord servera koji vodim (gde možete pronaći i druge čitaoce ove knjige), Twittera, LinkedIna ili drugih kanala koje možete pronaći na mojoj veb stranici.

Kome je namenjena knjiga

Ova knjiga je za svakoga ko želi da iskoristi mašinsko učenje za rešavanje stvarnih problema. ML u ovoj knjizi odnosi se i na duboko učenje (deep learning) i klasične algoritme, sa naglaskom na sisteme mašinskog učenja na velikoj skali, kao što su oni viđeni u srednjim do velikim preduzećima i brzo rastućim startapovima. Sistemi manje skale imaju tendenciju da budu manje složeni i mogu manje koristiti od sveobuhvatnog pristupa izloženog u ovoj knjizi.

Budući da mi je struka inženjerstvo, jezik ove knjige je usmeren ka inženjerima, uključujući inženjere mašinskog učenja, naučnike podataka inženjere podataka inženjere platformi mašinskog učenja i menadžere inženjeringa. Možda se možete poistovetiti sa jednim od sledećih scenarija:

- Dobili ste poslovni problem i puno sirovih podataka. Želite da konstruišete ove podatke i izaberete prave metrike za rešavanje ovog problema.
- Vaši početni modeli dobro rade u oflajn eksperimentima i želite da ih primenite.
- Imate malo povratnih informacija o performansama vaših modela nakon što su primenjeni i želite da pronađete način da brzo otkrijete, otklonite i adresirate bilo koji problem sa kojim se vaši modeli mogu suočiti u produkciji.
- Proces razvoja, evaluacije, primene i ažuriranja modela za vaš tim je uglavnom bio ručni, spor i sklon greškama. Želite da automatizujete i poboljšate ovaj proces.
- Svaki ML slučaj korišćenja u vašoj organizaciji je primenjen koristeći sopstveni tok rada i želite da postavite temelj (npr. skladište modela, skladište karakteristika, alate za monitoring) koji se može deliti i ponovo koristiti preko slučajeva korišćenja.
- Brinete da vaši sistemi mašinskog učenja mogu imati pristrasnosti i želite da učinite vaše sisteme odgovornim.

Možete imati koristi od knjige ako pripadate jednoj od sledećih grupa:

- Razvojni programeri alata koji žele da identifikuju oblasti koje su nedovoljno iskorišćene u proizvodnji ML-a i shvate kako da pozicionirate vaše alate u ekosistemu.
- Pojedinci koji traže poslove u vezi sa ML-om u industriji.
- Tehnički i poslovni lideri koji razmišljaju o usvajanju ML rešenja za poboljšanje vaših proizvoda i/ili poslovnih procesa. Čitaoci bez jake tehničke pozadine mogli bi imati najviše koristi od Poglavlja 1, 2 i 11.

Šta ova knjiga nije

Ova knjiga nije uvod u ML. Postoje mnoge knjige, kursevi i resursi dostupni za teoriju ML-a i stoga se ova knjiga odmiče od ovih koncepta kako bi se fokusirala na praktične aspekte ML-a. Da budemo specifični, knjiga pretpostavlja da čitaoci imaju osnovno razumevanje sledećih tema:

- *ML modeli* kao što su klasterisanje, logistička regresija, stabla odluka, kolaborativno filtriranje i različite arhitekture neuronskih mreža uključujući napredne, rekurentne, konvolutivne i transformatorske
- *ML tehnike* kao što su nadgledano nasuprot nenadgledanom učenju, gradijentni spust, cilj/funkcija gubitka, regularizacija, generalizacija i podešavanje hiperparametara

- *Metrike* kao što su tačnost, F1 skor, preciznost, odziv, ROC, srednja kvadratna greška i log-verovatnoća
- *Statistički koncepti* kao što su varijansa, verovatnoća i normalna/sa dugim repom distribucija
- *Uobičajeni ML zadaci* kao što su modelovanje jezika, detekcija anomalija, klasifikacija objekata i mašinski prevod

Nije potrebno da ove teme znate u detalje – za koncepte čije se tačne definicije teže pamte, npr. F1 skor, uključujemo kratke beleške kao referencu – ali trebalo bi imati grubu ideju šta oni znače.

Iako ova knjiga pominje tekuće alate da bi ilustrovala određene koncepte i rešenja, knjiga nije udžbenik. Tehnologije se razvijaju tokom vremena. Alati brzo ulaze i izlaze iz mode, ali osnovni pristupi rešavanju problema trebalo bi da traju malo duže. Ova knjiga pruža okvir kako biste procenili koji alat najbolje funkcioniše za vaše slučajeve upotrebe. Kada postoji alat koji želite da koristite, obično je lako pronaći udžbenika za njega na internetu. Kao rezultat toga, ova knjiga sadrži malo delova koda, a umesto toga se fokusira na pružanje velikog izlaganja o kompromisima, prednostima i nedostacima i konkretnim primerima.

Kako se kretati kroz ovu knjigu

Poglavlja u ovoj knjizi su organizovana tako da odražavaju probleme sa kojima se naučnici podataka mogu susresti tokom životnog ciklusa ML projekta. Prva dva poglavlja postavljaju osnovu za uspostavljanje uspeha ML projekta, počevši od osnovnog pitanja: da li vašem projektu treba ML? Pokriva i izbor ciljeva za vaš projekat i kako postaviti vaš problem na način koji vodi ka jednostavnijim rešenjima. Ako ste već upoznati sa ovim razmatranjima i nestrpljivi da stignete do tehničkih rešenja, slobodno preskočite prva dva poglavlja.

Poglavlja 4 do 6 pokrivaju fazu ML projekta pre implementacije i primene: od kreiranja podataka za trening i inženjeringa osobina do razvoja i evaluacije vaših modela u razvojnom okruženju. Ovo je faza u kojoj su posebno potrebni ekspertiza i u ML-u i u domenu problema.

Poglavlja 7 do 9 pokrivaju implemenatciju i primenu i potonju fazu ML projekta. Saznaćemo kroz priču sa kojom se mnogi čitaoci mogu povezati da postavljanje modela nije kraj procesa primene. Primenjen model treba da bude nadgledan i kontinualno ažuriran prema promenama okruženja i poslovnim zahtevima.

Poglavlja 3 i 10 fokusiraju se na infrastrukturu potrebnu da omogući saradnicima iz različitim predznanjima da rade zajedno na isporuci uspešnih ML sistema. Poglavlje 3 se fokusira na sisteme podataka, dok se Poglavlje 10 fokusira na

kompjutersku infrastrukturu i ML platforme. Dugo sam razmišljala o tome koliko duboko da idem u sisteme podataka i gde da ih uvedem u knjigu. Sistemi podataka, uključujući baze podataka, formate podataka, premeštanje podataka i procesore obrade podataka, tendenciono su slabo pokriveni u ML kursu i stoga mnogi naučnici podataka mogu misliti o njima kao na nešto niskog nivoa ili nebitno. Posle konsultovanja sa mnogim kolegama, odlučila sam da zato što ML sistemi zavise od podataka, prethodno pokrivanje osnova sistema podataka pomoći će nam da budemo na istoj stranici kada raspravljamo o pitanjima podataka u ostatku knjige.

Iako pokrivamo mnoge tehničke aspekte ML sistema u ovoj knjizi, ML sisteme grade ljudi, za ljude i mogu imati veliki uticaj na živote mnogih. Bio bi propust napisati knjigu o ML u proizvodnji bez poglavlja o ljudskoj strani toga, što je fokus Poglavlja 11, poslednjeg poglavlja.

Zapamtite da je „naučnik podataka“ uloga koja se dosta razvijala u proteklih nekoliko godina i bilo je mnogo diskusija da se odredi šta ova uloga treba da obuhvati – ući ćemo u neka od ovih razmatranja u Poglavlju 10. U ovoj knjizi, koristimo „naučnik podataka“ (data scientist) kao opšti pojam koji obuhvata svakoga ko radi na razvoju i primenu ML modela, uključujući ljude čiji su nazivi poslova možda ML inženjeri, inženjeri podataka, analitičari podataka itd.

GitHub skladište i zajednica

Ova knjiga je praćena GitHub skladištem koji sadrži:

- Pregled osnovnih ML koncepta
- Spisak referenci korišćenih u ovoj knjizi i drugih naprednih, ažuriranih resursa
- Delove koda korišćenog u ovoj knjizi
- Listu alata koje možete koristiti za određene probleme sa kojima se možete susresti u vašim radnim procesima

Vodim i Discord server o MLOps-u gde ste pozvani da diskutujete i postavljate pitanja o knjizi.

Konvencije korišćene u ovoj knjizi

Sledeće tipografske konvencije su korišćene u ovoj knjizi:

Kurziv

Označava nove termine, URL-ove, email adrese imena fajlova i ekstenzije fajlova.

`Konstantna širina`

Koristi se za programsku listu, kao i unutar paragrafa za pozivanje na programske elemente kao što su imena varijabli ili funkcija, baze podataka, tipovi podataka, environment varijable, naredbe i ključne reči.





Ovaj element ukazuje na upozorenje ili oprez.

Upotreba primera koda

Kao što je pomenuto, dodatni materijal (primeri koda, vežbe itd.) dostupan je za preuzimanje na *https://oreil.ly/designing-machine-learning-systems-code*.

Ako imate tehničko pitanje ili problem u korišćenju primera koda, molimo da pošaljete email na *bookquestions@oreilly.com*.

Ova knjiga je ovde da vam pomogne da obavite svoj posao. Generalno, ako je primer koda ponuđen sa ovom knjigom, možete ga koristiti u svojim programima i dokumentaciji. Ne treba da nas kontaktirate za dozvolu osim ako reprodukujete značajan deo koda. Na primer, pisanje programa koji koristi nekoliko delova koda iz ove knjige ne zahteva dozvolu. Prodaja ili distribucija primera iz O'Reilly knjiga zahteva dozvolu. Odgovaranje na pitanje citiranjem ove knjige i navođenjem primera koda ne zahteva dozvolu. Uključivanje značajne količine primera koda iz ove knjige u dokumentaciju vašeg proizvoda zahteva dozvolu.

Cenimo, ali generalno ne zahtevamo, navođenje. Navođenje obično uključuje naslov, autora izdavača i ISBN. Na primer: „Mašinsko učenje: projektovanje sistema“ od Chip Huyen (Mikro knjiga). Copyright 2024 Huyen Thi Khanh Nguyen, 978-86-7555-474-5.“

Ako smatrate da vaša upotreba primera koda izlazi izvan fer upotrebe ili dozvole navedene iznad, slobodno nas kontaktirajte na *permissions@oreilly.com*.

POGLAVLJE 1

Pregled sistema mašinskog učenja

U novembru 2016. godine, Google je objavio da je integrisao svoj višejezični neuronski sistem za mašinsko prevođenje u Google Translate, označavajući jednu od prvih uspešnih priča dubokih veštačkih neuronskih mreža u proizvodnji i u velikom obimu.[1] Prema Googlu, sa ovim ažuriranjem, kvalitet prevođenja se poboljšao više nego u prethodnih 10 godina zajedno.

Ovaj uspeh dubokog učenja obnovio je interesovanje za mašinsko učenje (ML, machine learning) uopšte. Od tada, sve više kompanija se okreće ML-u za rešavanje svojih najizazovnijih problema. Za samo pet godina, ML je pronašao svoj put u gotovo svaki aspekt našeg života: kako pristupamo informacijama, kako komuniciramo, kako radimo, kako pronalazimo ljubav. Širenje ML-a bilo je tako brzo da je već teško zamisliti život bez njega. Ipak, još uvek postoji mnogo više upotreba za ML koje čekaju da budu istražene u oblastima kao što su zdravstvo, transport, poljoprivreda, pa čak i u pomoći pri razumevanju univerzuma.[2]

Mnogi ljudi, kada čuju „sistem za mašinsko učenje," razmišljaju samo o ML algoritmima koji se koriste, kao što su logistička regresija ili različite vrste neuronskih mreža. Međutim, algoritam je samo mali deo ML sistema u proizvodnji. Sistem uključuje i poslovne zahteve koji su stvorili ML projekat na prvom mestu, interfejs gde korisnici i razvojni inženjeri komuniciraju sa vašim sistemom, stek podatke i logiku za razvoj, praćenje i ažuriranje modela, kao i infrastrukturu koja omogućava isporuku te logike. Slika 1-1 vam prikazuje različite komponente ML sistema i u kojim poglavljima ove knjige će biti obrađene.

[1] Mike Schuster, Melvin Johnson i Nikhil Thorat, „Zero-Shot Translation with Google's Multilingual Neural Machine Translation System" *Google AI Blog*, November 22, 2016, *https://oreil.ly/2R1CB*.

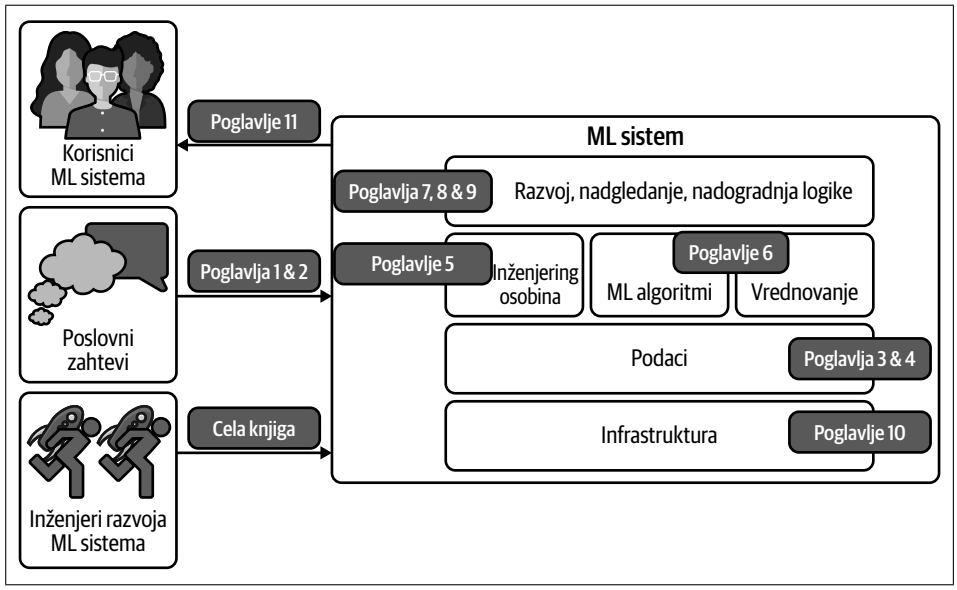
[2] Larry Hardesty, „A Method to Image Black Holes" *MIT News*, June 6, 2016, *https://oreil.ly/HpL2F*.



Odnos između MLOps i dizajna ML sistema

Ops u MLOps dolazi od DevOps, skraćeno od Razvoj i Operacije (Developments and Operations). Operacionalizacija nečega znači dovesti ga u proizvodnju, što uključuje implementaciju, nadgledanje i održavanje. MLOps je skup alatki i najboljih praksi za donošenje ML-a u proizvodnju.

Dizajn ML sistema koristi sistemski pristup MLOps-u, što znači da razmatra ML sistem holistički kako bi se obezbedilo da svi delovi i njihovi zainteresovani akteri mogu zajedno raditi kako bi zadovoljili navedene ciljeve i zahteve.



Slika 1-1. Različite komponente ML sistema. „ML algoritmi" obično je ono o čemu ljudi razmišljaju kada kažu mašinsko učenje, ali to je samo mali deo celog sistema.

Postoji mnogo odličnih knjiga o različitim algoritmima za mašinsko učenje. Ova knjiga ne pokriva specifične algoritme u detalje, već pomaže čitaocima da razumeju ceo sistem za mašinsko učenje kao celinu. Drugim rečima, cilj ove knjige je da vam pruži okvir za razvoj rešenja koje najbolje funkcioniše za vaš problem, bez obzira na to koji algoritam na kraju koristite. Algoritmi mogu brzo zastareti jer se neprestano razvijaju novi, ali okvir predložen u ovoj knjizi trebalo bi da funkcioniše i sa novim algoritmima.

Prvo poglavlje knjige ima za cilj da vam pruži pregled onoga što je potrebno da se mašinski model uvede u primenu. Pre diskusije o tome kako razviti sistem za mašinsko učenje, važno je postaviti osnovno pitanje kada koristiti a kada ne koristiti

mašinsko učenje. Pokrićemo neke popularne primene mašinskog učenja kako bismo ilustrovali ovu tačku.

Nakon primena, preći ćemo na izazove uvođenja sistema za mašinsko učenje, i to tako što ćemo uporediti mašinsko učenje u proizvodnom procesu sa mašinskim učenjem u istraživanju, kao i sa tradicionalnim softverom. Ako ste već bili na bojnom polju razvoja namenskih sistema za mašinsko učenje, možda ste već upoznati sa onim što je napisano u ovom poglavlju. Međutim, ako ste imali samo iskustvo sa mašinskim učenjem u akademskom okruženju, ovo poglavlje će vam pružiti iskren pogled na mašinsko učenje u stvarnom svetu i učiniće vašu prvu primenu uspešnom.

Kada koristiti mašinsko učenje

Kako je brzo prihvatanje u industriji raslo, mašinsko učenje se pokazalo kao moćan alat za širok spektar problema. Uprkos neverovatnoj količini uzbuđenja i hajpa koji su izazvali ljudi i unutar i van oblasti, mašinsko učenje nije čarobni alat koji može rešiti sve probleme. Čak i za probleme koje mašinsko učenje može rešiti, rešenja sa mašinskim učenjem možda nisu optimalna rešenja. Pre nego što započnete projekat mašinskog učenja, možete da se zapitate da li je mašinsko učenje neophodno ili ekonomično.[3]

Da bismo razumeli šta mašinsko učenje može da uradi, pogledajmo šta uopšteno čine rešenja mašinskog učenja:

> Mašinsko učenje je pristup za (1) *učenje* (2) *složenih obrazaca* iz (3) *postojećih podataka* i korišćenje ovih obrazaca za (4) *pravljenje predviđanja* na (5) *neviđenim podacima*.

Proučićemo svaku od navedenih ključnih fraza, napisanih kosim pismom, u gornjem opisu kako bismo razumeli njihove implikacije za probleme koje mašinsko učenje može rešiti:

1. Učenje: sistem ima sposobnost da uči

Relaciona baza podataka nije sistem za mašinsko učenje jer nema kapacitet da uči. Možete eksplicitno navesti vezu između dve kolone u relacionoj bazi podataka, ali je malo verovatno da će sama moći da otkrije vezu između ovih dveju kolona.

Da bi mašinski sistem mogao da uči, mora postojati nešto od čega može da uči. U većini slučajeva, mašinsko učenje, uči iz podataka. U superviziranom (eng. supervised) učenju, na osnovu primera ulaznih i izlaznih parova, mašinsko učenje uči kako da generišu izlaze za proizvoljne ulaze. Na primer, ako želite da izgradite sistem za mašinsko učenje koji će naučiti da predviđa cenu

[3] Nisam pitao da li je mašinsko učenje dovoljno, jer je odgovor uvek ne.

najma za Airbnb oglase, morate obezbediti skup podataka gde svaki ulaz predstavlja oglas sa relevantnim karakteristikama (kvadratura, broj soba, kvart, pogodnosti, ocena tog oglasa itd.) i pripadajući izlaz je cena najma tog oglasa. Nakon što se nauči, ovaj sistem za mašinsko učenje treba da predviđa cenu novog oglasa na osnovu njegovih karakteristika.

2. *Složeni obrasci: postoje obrasci za učenje i oni su složeni*

   Rešenja za mašinsko učenje su korisna samo kada postoje obrasci za učenje. Normalni ljudi ne ulažu novac u izgradnju sistema za mašinsko učenje kako bi predviđali sledeći ishod nenameštene kocke jer ne postoji obrazac u tome kako se ishodi generišu.[4] Međutim, postoje obrasci u tome kako se cene akcija formiraju, pa su kompanije uložile milijarde dolara u izgradnju sistema za mašinsko učenje kako bi naučili te obrasce.

   Da li obrazac postoji možda nije očigledno, ili ako obrasci postoje, vaš skup podataka ili algoritmi za mašinsko učenje možda nisu dovoljni da ih uhvate. Na primer, može postojati obrazac u tome kako tvitovi Ilona Maska utiču na cene kriptovaluta. Međutim, ne biste znali dok svoje modele za mašinsko učenje niste rigorozno obučili i razvili na njegovim tvitovima. Čak i ako svi vaši modeli ne uspeju da daju razumne prognoze cena kriptovaluta, to ne znači da obrazac ne postoji.

   Razmotrite veb sajt kao što je Airbnb sa mnogo oglasa za kuće; svaki oglas dolazi sa poštanskim brojem. Ako želite da sortirate oglase u države u kojima se nalaze, nećete trebati sistem za mašinsko učenje. S obzirom na to da je obrazac jednostavan – svaki poštanski broj odgovara poznatoj državi – možete jednostavno koristiti tabelu za pretragu (eng. lookup table).

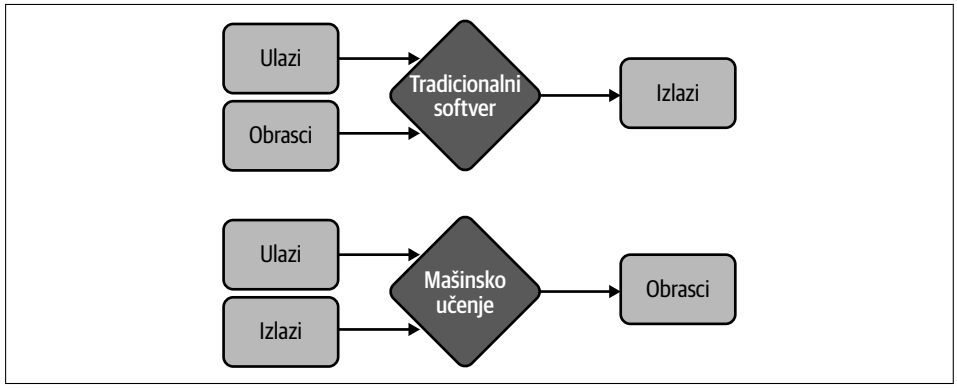
   Veza između cene najma i svih njegovih karakteristika prati mnogo složeniji obrazac, koji bi bilo vrlo teško ručno specificirati. Mašinsko učenje je dobro rešenje za ovo. Umesto da vašem sistemu kažete kako da izračuna cenu na osnovu liste karakteristika, možete obezbediti cene i karakteristike i pustiti vaš sistem za mašinsko učenje da otkrije obrazac. Razlika između rešenja za mašinsko učenje i rešenja sa tabelom pretrage, kao i opšta rešenja tradicionalnog softvera prikazana je na Slici 1-2. Iz ovog razloga, mašinsko učenje se naziva i Softver 2.0.[5]

   Mašinsko učenje (ML) je postiglo veliki uspeh u rešavanju zadataka sa složenim obrascima, kao što su detekcija objekata i prepoznavanje govora. Ono što je složeno za mašine razlikuje se od onog što je složeno za ljude. Mnogi zadaci koji su teški za ljude, mašinama su laki, na primer, podizanje broja na stepen 10. S druge

[4] Obrasci se razlikuju od distribucija. Znamo distribuciju ishoda nenameštene kocke, ali nema obrazaca po kome se ishodi generišu.

[5] Andrej Karpathy, „Software 2.0“ *Medium*, November 11, 2017, *https://oreil.ly/yHZrE*.

strane, mnogi zadaci koji su laki za ljude mogu biti teški za mašine, na primer, odlučivanje da li je na slici mačka.



Slika 1-2. Umesto zahtevanja ručno definisanih obrazaca za izračunavanje rezultata, ML rešenja uče obrasce iz ulaza i izlaza

3. Postojeći podaci: podaci su dostupni ili ih je moguće prikupiti

Pošto ML uči iz podataka, moraju postojati podaci iz kojih može da uči. Zabavno je razmišljati o izgradnji modela za predviđanje koliko poreza osoba treba da plati godišnje, ali to nije moguće osim ako imate pristup podacima o porezu i prihodima velike populacije.

U kontekstu učenja bez uzoraka (ponekad poznatog kao učenje bez podataka), moguće je da ML sistem daje dobre prognoze za zadatak iako nije treniran na podacima za taj zadatak. Međutim, taj ML sistem je prethodno treniran na podacima za druge zadatke, često slične zadatku koji se razmatra. Dakle, iako sistem ne zahteva podatke za treniranje za trenutni zadatak i dalje zahteva podatke za učenje.

Takođe je moguće pokrenuti ML sistem bez podataka. Na primer, u kontekstu kontinualnog učenja, ML modeli mogu biti implementirani bez prethodnog treniranja na bilo kakvim podacima, ali će učiti iz dolaznih podataka tokom primene.[6]Međutim, pružanje nedovoljno obučenih modela korisnicima nosi određene rizike, kao što je loše iskustvo korisnika.

Bez podataka i bez kontinualnog učenja, mnoge kompanije primenjuju pristup „fake-it-til-you make it“ (pretvaraj se dok ne uspeš): isporučuju proizvode koji pružaju predviđanja koja su napravili ljudi, umesto ML modela, s nadom da će kasnije koristiti generisane podatke za obuku ML modela.

[6] Proučićemo online učenje u Poglavlju 9.

4. Predviđanja: radi se o prediktivnom problemu

ML modeli prave predviđanja, tako da mogu rešiti samo probleme koji zahtevaju prediktivne odgovore. ML može biti posebno privlačan kada možete koristiti veliku količinu jeftinih, ali približnih predviđanja. Na engleskom jeziku, „predict" znači „proceniti vrednost u budućnosti". Na primer, kakvo će biti vreme sutra? Ko će pobediti na Super Bowlu ove godine? Koju će film korisnik želeti da pogleda sledeći?

Kako se prediktivne mašine (npr. ML modeli) postaju sve efikasnije, sve više problema se ulazi u sferu prediktivnih problema. Bez obzira na pitanje koje imate, uvek ga možete postaviti kao: „Kakav bi bio odgovor na ovo pitanje?" bez obzira na to da li je ovo pitanje o nečemu u budućnosti, sadašnjosti ili čak prošlosti.

Problemi koji zahtevaju intenzivno računanje su jedna klasa problema koja su veoma uspešno ušli u sferu prediktivnih. Umesto da računate tačan ishod procesa, što može biti više vremenski i računski zahtevno od ML-a, možete postaviti problem kao: „Kakav bi izgledao ishod ovog procesa?" i aproksimirati rezultat koristeći ML model. Rezultat će biti približna vrednost tačnog rezultata, ali često je i dovoljno dobar. Možete videti mnogo primera u grafičkim prikazima, kao što su smanjenje šuma na slici i prosvetljenje slike.[7]

5. Nepoznati podaci: nepoznati podaci dele obrasce sa podacima za obuku

Obrasci koje vaš model uči iz postojećih podataka su korisni samo ako obrasci važe i na nepoznatim podacima. Model koji predviđa da li će se aplikacija preuzimati na Božić 2020. godine neće se dobro ponašati ako je obučen na podacima iz 2008. godine, kada je najpopularnija aplikacija na App Store bila „Koi Pond". Šta je „Koi Pond"? Tačno.

U tehničkom smislu, to znači da vaši nepoznati podaci i podaci za obuku treba da potiču iz sličnih distribucija. Možda ćete se zapitati: „Ako podaci nisu viđeni, kako znamo iz koje distribucije dolaze?" Ne znamo, ali možemo da napravimo pretpostavke – na primer, možemo pretpostaviti da će ponašanje korisnika sutra biti slično ponašanju korisnika danas – i nadamo se da će naše pretpostavke biti tačne. Ako nisu, imaćemo model koji se loše ponaša, što možemo otkriti putem nadgledanja, kao što je opisano u Poglavlju 8 i testiranja u eksploataciji, kao što je opisano u Poglavlju 9.

---

[7] Steke Bako, Thijs Vogels, Brian McWilliams, Mark Meyer, Jan Novák, Alex Harvill, Pradeep Sen, Tony Derose i Fabrice Rousselle, „Kernel-Predicting Convolutional Networks for Denoising Monte Carlo Renderings" *ACM Transactions on Graphics* 36, no. 4 (2017): 97, *https://oreil.ly/EeI3j*; Oliver Nalbach, Elena Arabadzhiyska, Dushyant Mehta, Hans-Peter Seidel i Tobias Ritschel, „Deep Shading: Convolutional Neural Networks for Screen-Space Shading" *arXiv*, 2016, *https://oreil.ly/dSspz*.

Zbog načina na koji većina ML algoritama uči danas, ML rešenja će posebno briljirati ako vaš problem ima sledeće dodatne karakteristike:

6. Repetitivan je

Ljudi su veoma dobri u učenju na osnovu malog broja primera: možete pokazati deci nekoliko slika mačaka i većina će prepoznati mačku sledeći put kada je vide. Uprkos uzbudljivom napretku u istraživanju učenja na osnovu malog broja primera, većina ML algoritama i dalje zahteva mnogo primera da bi naučili obrazac. Kada je zadatak repetitivan, svaki obrazac se ponavlja više puta, što mašinama olakšava učenje.

7. Cena netačnih predviđanja je niska

Ako performanse vašeg ML modela nisu uvek 100%, što je veoma malo verovatno za bilo koje značajne zadatke, vaš model će praviti greške. ML je posebno prikladan kada je cena pogrešne prognoze niska. Na primer, jedna od najvećih upotreba ML danas je u sistemima preporuka jer je kod sistema za preporuke loša preporuka obično oproštena – korisnik jednostavno neće kliknuti na preporuku.

Ako jedna greška u prognozi može imati katastrofalne posledice, ML i dalje može biti odgovarajuće rešenje ako, u proseku, koristi od tačnih prognoza prevazilaze troškove netačnih prognoza. Razvoj samovozećih automobila je izazovan jer algoritamska greška može dovesti do smrti. Međutim, mnoge kompanije i dalje žele razvijati samovozeće automobile jer imaju potencijal da spasu mnoge živote kada samovozeći automobili statistički budu sigurniji od ljudskih vozača.

8. U velikom je obimu

ML rešenja često zahtevaju značajna početna ulaganja u podatke, računarstvo, infrastrukturu i talente, pa bi bilo logično ako možemo koristiti ova rešenja mnogo puta.

Izraz „u velikom obimu“ znači različite stvari za različite zadatke, ali uopšteno se odnosi na pravljenje velikog broja prognoza. Primeri uključuju sortiranje miliona e-pošta godišnje ili dnevno predviđanje koje odeljenje treba da bude zaduženo za hiljade zahteva za podršku.

Problem može izgledati kao pojedinačna prognoza, ali zapravo se radi o nizu prognoza. Na primer, model koji predviđa ko će pobediti na izborima za predsednika SAD čini se da pravi samo jednu prognozu svakih četiri godine, ali zapravo može činiti prognozu svaki sat ili čak češće jer se ta prognoza mora stalno ažurirati kako bi se uključile nove informacije.

Imati problem u velikom obimu takođe znači da imate mnogo podataka koje možete prikupiti, što je korisno za obuku ML modela.

9. Obrasci se konstantno menjaju

Kulture se menjaju. Ukusi se menjaju. Tehnologije se menjaju. Ono što je danas trend može biti zastarelo već sutra. Razmislite o zadatku klasifikacije spam e-pošte. Danas je indikacija spam e-pošte nigerijski princ, ali sutra to može biti uznemireni vietnamski pisac.

Ako vaš problem uključuje jedan ili više konstantno promenljivih obrazaca, rešenja bazirana na ručno napisanim pravilima mogu brzo zastareti. Pronalaženje načina kako se vaš problem promenio kako biste mogli ažurirati ručno napisana pravila može biti previše skupo ili nemoguće. Budući da ML uči iz podataka, možete ažurirati svoj ML model novim podacima bez potrebe da saznate kako su se podaci promenili. Takođe je moguće postaviti sistem da se prilagodi promenljivim distribucijama podataka, što je pristup koji ćemo razmotriti u odeljku „Kontinuirano učenje".

Spisak upotreba može da se nastavi i da dalje raste kako se primena mašinskog učenja razvija u industriji. Iako mašinsko učenje može da reši određeni skup problema vrlo dobro, ne može da reši i/ili ne bi trebalo da se koristi za mnoge probleme. Većinu današnjih algoritama za mašinsko učenje ne bi trebalo koristiti pod sledećim uslovima:

- Ako je neetično. Razmotrićemo jedan studijski slučaj gde se može raspravljati da je upotreba algoritama za mašinsko učenje neetična u delu „Studijski slučaj I: Biases automatizovanog ocenjivača".
- Jednostavnija rešenja su često dovoljna. U poglavlju 6, pokrićemo četiri faze razvoja modela mašinskog učenja, gde prva faza treba da bude primena rešenja koja ne uključuju mašinsko učenje.
- Ako nije isplativo.

Međutim, čak i ako mašinsko učenje ne može da reši vaš problem, moguće je razbiti problem na manje komponente i koristiti mašinsko učenje da reši neke od njih. Na primer, ako ne možete da napravite četbota koji odgovara na sve upite vaših korisnika, možda je moguće napraviti model za mašinsko učenje koji predviđa da li upit odgovara jednom od često postavljanih pitanja. Ako da, uputite korisnika na odgovor. Ako ne, uputite ih na korisničku podršku.

Takođe želim da upozorim da ne treba odbacivati novu tehnologiju samo zato što trenutno nije isplativa kao postojeće tehnologije. Većina tehnoloških napredaka je inkrementalna. Vrsta tehnologije možda nije efikasna sada, ali može da bude tokom vremena uz više investicija. Ako čekate da tehnologija dokaže svoju vrednost ostatku industrije pre nego što se uključite, možete završiti godinama ili decenijama iza svojih konkurenata.

Primeri upotrebe mašinskog učenja

Mašinsko učenje sve više nalazi primenu kako u preduzećima tako i u aplikacijama za potrošače. Od sredine 2010-ih godina, došlo je do eksplozije aplikacija koje koriste mašinsko učenje kako bi pružile superiorne ili prethodno nemoguće usluge potrošačima.

Uz eksploziju informacija i usluga, bilo bi veoma izazovno pronaći ono što želimo bez pomoći mašinskog učenja, bilo da se manifestuje u obliku *pretraživača* ili *sistema za preporuke*. Kada posetite veb sajt poput Amazona ili Netflixa, preporučuju vam se predmeti koji su predviđeni da najbolje odgovaraju vašem ukusu. Ako vam se ne sviđaju preporučeni predmeti, možda ćete želeti da tražite određene predmete, a vaši rezultati pretrage verovatno su dobijeni mašinskim učenjem.

Ako imate pametan telefon, verovatno već koristite veštačku inteligenciju u mnogim svakodnevnim aktivnostima. Pisanje na vašem telefonu postaje lakše uz *prediktivno kucanje*, ML vam daje sugestije o tome šta biste mogli da napišete sledeće. ML može raditi u vašoj aplikaciji za uređivanje fotografija kako bi predložio kako najbolje poboljšati vaše fotografije. Možda ćete autentifikovati svoj telefon koristeći otisak prsta ili vaše lice, što zahteva ML sistem da predvidi da li otisak prsta ili lice odgovaraju vašem.

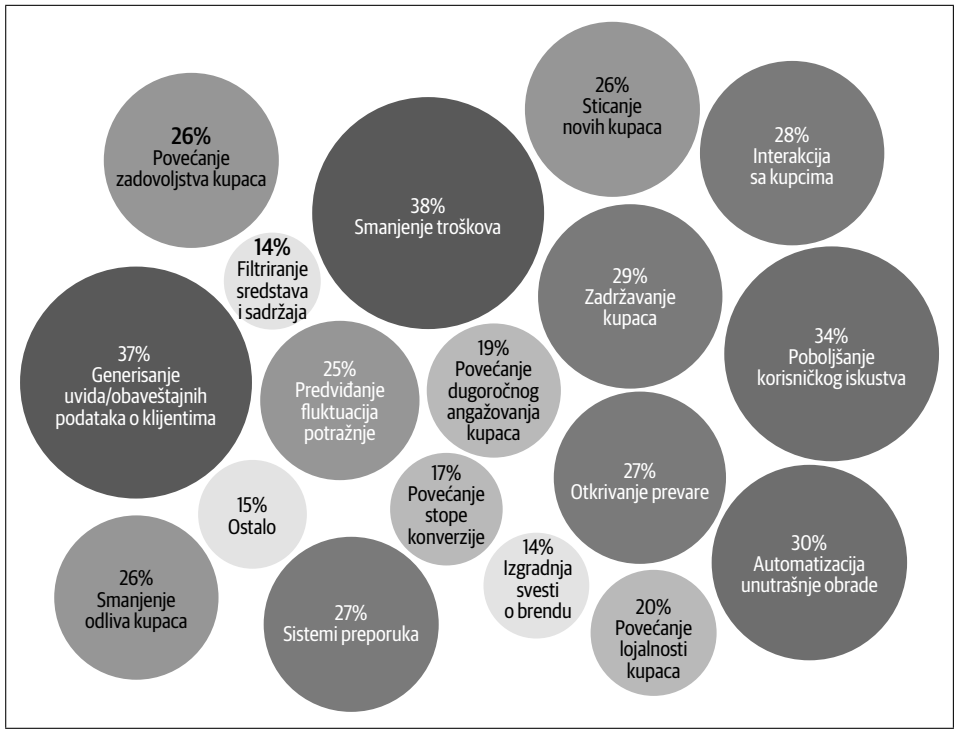
Primer upotrebe mašinskog učenja (ML) koji me privukao ovom polju bio je *mašinsko prevođenje*, automatsko prevođenje sa jednog jezika na drugi. To ima potencijal da omogući ljudima iz različitih kultura da komuniciraju međusobno, brišući jezičku barijeru. Moji roditelji ne govore engleski, ali zahvaljujući Google Prevodu (Google Translate), sada mogu čitati moje tekstove i razgovarati s mojim prijateljima koji ne govore vietnamski jezik.

ML je sve prisutniji u našim domovima putem pametnih ličnih asistenata kao što su Alexa i Google Assistant. Pametne sigurnosne kamere mogu vam reći kada vaši ljubimci napuste kuću ili ako imate neželjenog gosta. Jedan moj prijatelj se brinuo za svoju stariju majku koja živi sama – ako padne, niko nije tu da joj pomogne da ustane – pa se oslanjao na sistem za praćenje zdravlja kod kuće koji predviđa da li je neko pao u kući.

Iako je tržište za potrošačke primene ML usponu, većina upotreba ML je i dalje u poslovnom svetu. ML aplikacije u poslovnom svetu obično imaju potpuno drugačije zahteve i razmatranja u odnosu na potrošačke aplikacije. Postoji mnogo izuzetaka, ali u većini slučajeva, poslovne aplikacije mogu imati strože zahteve za tačnost, ali mogu biti tolerantnije prema zahtevima za latenciju (kašnjenje). Na primer, poboljšanje tačnosti sistema za prepoznavanje govora sa 95% na 95,5% možda neće biti primetno većini potrošača, ali poboljšanje efikasnosti sistema za alokaciju resursa za samo 0,1% može pomoći kompanijama kao što su Google ili General Motors da uštede milione dolara. Istovremeno, latencija od jedne sekunde

može ometati potrošača i naterati ga da otvori nešto drugo, ali poslovni korisnici mogu biti tolerantniji prema visokoj latenciji. Za ljude koji su zainteresovani za razvoj kompanija zasnovanih na ML aplikacijama, potrošačke aplikacije mogu biti lakše distribuirane, ali su mnogo teže monetizovanje. Međutim, većina poslovnih slučajeva primene nije očigledna ako se nisu susreli s njima lično.

Prema istraživanju Algoritamija o primeni mašinskog učenja u preduzećima iz 2020. godine, ML aplikacije u preduzećima su raznovrsne i služe kako internim slučajevima upotrebe (smanjenje troškova, generisanje uvida i inteligencije o korisnicima, unutrašnja automatizacija obrade), tako i spoljnim slučajevima upotrebe (unapređenje iskustva korisnika, zadržavanje korisnika, interakcija sa korisnicima), kako je prikazano na Slici 1-3.[8]



Slika 1-3. Stanje primene mašinskog učenja u preduzećima 2020. Izvor: Prilagođeno sa slike od strane Algorithmia

Otkrivanje prevare je jedna od najstarijih primena ML-a u svetu preduzeća. Ako vaš proizvod ili usluga uključuje transakcije bilo koje vrednosti, podložan je prevari. Korišćenjem ML rešenja za otkrivanje anomalija, možete imati sisteme koji uče na osnovu istorijskih prevara i predviđaju da li će buduća transakcija biti prevara.

[8] „2020 State of Enterprise Machine Learning" *Algorithmia*, 2020, *https://oreil.ly/wKMZB*.

Odlučivanje koliko naplatiti za svoj proizvod ili uslugu verovatno je jedna od najtežih poslovnih odluka; zašto ne biste to prepustili ML-u? *Optimizacija cena* je proces procene cene u određenom vremenskom periodu radi maksimiziranja definisane ciljne funkcije, kao što su marža kompanije, prihod ili stopa rasta. ML bazirana optimizacija cena najviše je pogodna za slučajeve sa velikim brojem transakcija gde se tražnja menja i potrošači su voljni da plate dinamičnu cenu, na primer, internet oglasi, avionske karte, rezervacije smeštaja, deljenje vožnje i događaji.

Za vođenje posla važno je biti u mogućnosti da se prognozira potražnja potrošača kako biste mogli da pripremite budžet, uskladištite inventar, dodelite resurse i ažurirate strategiju cena. Na primer, ako vodite prodavnicu prehrambenih proizvoda, želite da imate dovoljno zaliha kako bi kupci pronašli ono što traže, ali ne želite da imate previše zaliha, jer ako to uradite, vaša roba može da propadne i gubite novac.

Sticanje novog korisnika je skupo. Prema podacima iz 2019. godine, prosečni trošak za aplikaciju da privuče korisnika koji će izvršiti kupovinu u aplikaciji iznosi 86,61 dolara.[9] Trošak privlačenja korisnika za kompaniju Lyft procenjuje se na 158 dolara.[10] Ovaj trošak je mnogo veći za korporativne korisnike. Trošak privlačenja korisnika označava se kao ubica startapa od strane investitora.[11] Smanjenje troškova privlačenja korisnika za mali iznos može rezultirati velikim povećanjem profita. To se može postići boljim identifikovanjem potencijalnih korisnika, prikazivanjem bolje ciljanih oglasa, davanjem popusta u pravo vreme itd. – sve to su zadaci pogodni za ML.

Nakon što ste potrošili toliko novca na privlačenje korisnika, bilo bi šteta ako odu. Trošak privlačenja novog korisnika procenjuje se kao 5 do 25 puta veći od troška zadržavanja postojećeg korisnika.[12] *Prognostika gubitka* je predviđanje kada će određeni korisnik prestati da koristi vaše proizvode ili usluge kako biste mogli da preduzmete odgovarajuće mere kako biste ih ponovo privukli. Prognostika gubitka može se koristiti ne samo za korisnike, već i za zaposlene.

Kako biste sprečili odlazak korisnika, važno je da ih zadovoljavate tako što ćete reagovati na njihove brige čim se pojave. Automatizovana klasifikacija podrške putem tiketa može pomoći u tome. Ranije, kada bi korisnik otvorio podršku putem tiketa ili poslao e-poštu, bilo je potrebno da se prvo obradi, a zatim da se prosledi

[9] „Average Mobile App User Acquisition Costs Worldwide from September 2018 to August 2019, by User Action and Operating System" *Statista*, 2019, *https://oreil.ly/2pTCH*.

[10] Jeff Henriksen, „Valuing Lyft Requires a Deep Look into Unit Economics" *Forbes*, May 17, 2019, *https://oreil.ly/VeSt4*.

[11] David Skok, „Startup Killer: The Cost of Customer Acquisition" *For Entrepreneurs*, 2018, *https://oreil.ly/L3tQ7*.

[12] Amy Gallo, „The Value of Keeping the Right Customers" *Harvard Business Review*, October 29, 2014, *https://oreil.ly/OlNkl*.

različitim odeljenjima dok ne stigne u inbox osobe koja može da je obradi. ML sistem može analizirati sadržaj tiketa i predvideti gde bi trebalo da ode, što može skratiti vreme odgovora i poboljšati zadovoljstvo korisnika. Takođe se može koristiti za klasifikaciju internih IT tiketa.

Još jedan popularan slučaj upotrebe ML-a u poslovanju je praćenje brenda. Brend je vredna imovina preduzeća.[13] Važno je pratiti kako javnost i vaši korisnici percipiraju vaš brend. Možda želite da znate kada/gde/kako se pominje, kako eksplicitno (npr. kada neko pomene „Google“) tako i implicitno (npr. kada neko kaže „pretraživački div“), kao i sentiment koji je povezan s njim. Ako odjednom dođe do naglog porasta negativnog sentimenta u pominjanju vašeg brenda, želećete da se time bavite što je pre moguće. Analiza sentimenta je tipičan zadatak ML-a.

Skup slučajeva upotrebe ML-a koji je nedavno izazvao veliko uzbuđenje nalazi se u oblasti zdravstva. Postoje ML sistemi koji mogu otkriti rak kože i dijagnostikovati dijabetes. Iako su mnoge primene u oblasti zdravstva usmerene ka potrošačima, zbog svojih strožih zahteva u vezi sa tačnošću i privatnošću, obično se pružaju putem pružalaca zdravstvenih usluga kao što je bolnica ili se koriste za pomoć lekarima pri postavljanju dijagnoze.

Razumevanje sistema mašinskog učenja

Razumevanje sistema mašinskog učenja će biti korisno pri njihovom dizajniranju i razvoju. U ovom odeljku, videćemo kako su ML sistemi različiti od istraživanja ML (ili kako se često predaju u školama) i tradicionalnog softvera, što opravdava potrebu za ovom knjigom.

Mašinsko učenje u istraživanju u odnosu na primenu

Kako se upotreba ML-a u industriji još uvek smatra relativno novom, većina ljudi sa stručnošću u oblasti ML-a je stekla kroz akademiju: pohađajući kurseve, istražujući, čitajući akademske radove. Ako navedeno opisuje vas, za vas može biti strm uspon da razumete izazove implementacije sistema mašinskog učenja i da se snađete u preplavljujućem nizu rešenja za izazove koje rešavate. ML u primeni je vrlo različit od ML-a u istraživanju. Tabela 1-1 prikazuje pet glavnih razlika.

[13] Marty Swant, „The World's 20 Most Valuable Brands“ *Forbes*, 2020, *https://oreil.ly/4uS5i*.

Tabela 1-1. Ključne razlike između mašinskog učenja u istraživanju i mašinskog učenja u primeni

	Istraživanje	Primena
Zahtev	Odlične performanse modela na repernim skupovima podataka	Različiti zainteresovani imaju različite zahteve
Računarski prioritet	Brza obuka, visok protok	Brza inferencija (predikcija), niska latencija
Podaci	Statički[a]	Konstantno promenljivi
Pravičnost	Često nije u fokusu	Mora se uzeti u obzir
Interpreternost	Često nije u fokusu	Mora se uzeti u obzir

[a] Jedno područje istraživanja fokusira se na kontinuirano učenje: razvijanje modela koji rade sa promenljivim raspodelama podataka. O tome ćemo govoriti u Poglavlju 9.

Različiti zainteresovani subjekti i zahtevi

Ljudi uključeni u istraživački projekat (stakeholders) i projekat tabele rangiranja (eng. leaderboard) često se usmereni na jedan cilj. Najčešći cilj je performansa modela – razviti model koji postiže vrhunske rezultate na repernim skupovima podataka. Da bi se ostvarilo i najmanje poboljšanje u performansi, istraživači često koriste tehnike koje čine modele suviše kompleksnim za praktičnu upotrebu.

U procesu dovođenja ML sistema u eksploataciju, uključeni su mnogi zainteresovani. Svaki zainteresovani ima svoje zahteve. Imajući različite, često suprotstavljene zahteve, može biti teško dizajnirati, razvijati i odabrati ML model koji zadovoljava sve zahteve.

Razmotrite mobilnu aplikaciju koja preporučuje restorane korisnicima. Aplikacija zarađuje tako što naplaćuje restoranima proviziju od 10% na svaku porudžbinu. To znači da skupi restorani donose više novca aplikaciji od jeftinih restorana. U projektu su uključeni inženjeri za ML, tim za prodaju, menadžeri proizvoda, inženjeri infrastrukture i menadžer:

Inženjeri za ML

Žele model koji preporučuje restorane koje će korisnici najverovatnije naručiti i veruju da to mogu postići korišćenjem složenijeg modela sa više podataka.

Tim za prodaju

Želi model koji preporučuje skuplje restorane jer takvi restorani donose više provizije.

Proizvodni tim

Obratite pažnju da svako povećanje latencije dovodi do smanjenja narudžbina, pa žele model koji može da ponudi preporučene restorane za manje od 100 milisekundi.

Tim za ML platformu

Kako raste saobraćaj, ovaj tim je budan usred noći zbog problema sa skaliranjem njihovog postojećeg sistema, pa žele da odlože ažuriranje modela kako bi prioritetno unapredili ML platformu.

Menadžer

Želi da maksimizira maržu, a jedan način da to postigne može biti da se odrekne ML tima.[14]

„Preporučivanje restorana na koje će korisnici najverovatnije kliknuti" i „preporučivanje restorana koji će doneti najviše novca aplikaciji" su dva različita cilja i u odeljku „Razdvajanje ciljeva" govorićemo o tome kako razviti ML sistem koji zadovoljava različite ciljeve. Za nestrpljive: razvijaćemo jedan model za svaki cilj i kombinovati njihove predikcije.

Za sada zamislimo da imamo dva različita modela. Model A je model koji preporučuje restorane na koje će korisnici najverovatnije kliknuti, a model B je model koji preporučuje restorane koji će doneti najviše novca aplikaciji. A i B mogu biti veoma različiti modeli. Koji model treba da bude implementiran korisnicima? Da bi odluka bila još teža, ni A ni B ne ispunjavaju zahtev postavljen od strane menadžerskog tima: ne mogu vratiti preporuke restorana za manje od 100 milisekundi.

Pri razvoju ML projekta, važno je da ML inženjeri razumeju zahteve svih učesnika i koliko su ovi zahtevi strogi. Na primer, ako je mogućnost vraćanja preporuka u roku od 100 milisekundi obavezan zahtev – kompanija je utvrdila da ako vaš model zahteva više od 100 milisekundi da preporuči restorane, 10% korisnika će izgubiti strpljenje i zatvoriti aplikaciju – tada ni model A ni model B neće raditi. Međutim, ako je to samo željeni zahtev, možda ćete i dalje razmotriti model A ili model B.

Stvarana primena, tj. proizvodnja, imajući različite zahteve u odnosu na istraživanje je jedan od razloga zašto uspešni istraživački projekti ponekad nisu uvek upotrebljivi u proizvodnji. Na primer, ansambliranje (eng. ensembling) je tehnika popularna među pobednicima mnogih takmičenja u oblasti ML, uključujući čuvenu Netflix nagradu od milion dolara, ali nije široko korišćena u proizvodnji. Ansambliranje kombinuje „više algoritama za učenje kako bi se postigla bolja prediktivna performansa nego što bi se mogla postići bilo kojim od pojedinačnih algoritama za učenje samostalno."[15] Iako može da poboljša performanse vašeg ML sistema,

[14] Nije neobično da timovi za mašinsko učenje i nauku o podacima budu među prvim tokom masovnog otpuštanja u kompaniji, kako je prijavljeno u IBM-u, Uberu, Airbnb-u. Vidi i Sejuti Das's analysis „How Data Scientists Are Also Susceptible to the Layoffs Amid Crisis" *Analytics India Magazine*, May 21, 2020, *https://oreil.ly/jobmz*.

[15] Wikipedia, s.v. „Ensemble learning" *https://oreil.ly/5qkgp*.

ansambliranje obično čini sistem suviše složenim za upotrebu u proizvodnji, na primer, usporava davanje predikcija ili otežava tumačenje rezultata. Više ćemo razgovarati o ansambliranju u odeljku „Ansambliranje".

Za mnoge zadatke, mala poboljšanja u performansama mogu rezultirati ogromnim povećanjem prihoda ili uštedom troškova. Na primjer, poboljšanje stopa klikova za 0,2% u sistemu preporuke proizvoda može rezultirati milionima dolara povećanja prihoda za internet prodavnicu. Međutim, za mnoge zadatke, mala poboljšanja možda neće biti primetna korisnicima. Za drugi tip zadatka, ako jednostavan model može obaviti zadovoljavajući posao, kompleksni modeli moraju biti značajno bolji da bi opravdali svoju složenost.

Računarski prioriteti

Prilikom dizajniranja sistema za mašinsko učenje, ljudi koji nisu implementirali sistem za mašinsko učenje često prave grešku fokusirajući se previše na deo razvoja modela, a premalo na deo implementacije i održavanja modela.

Tokom procesa razvoja modela, možete obučiti mnoge različite modele, a svaki model prolazi kroz više iteracija nad obukom podataka. Svaki obučen model zatim generiše predviđanja na validacionim podacima jednom kako bi prijavio rezultate. Validacioni podaci obično su mnogo manji od podataka za obuku. Tokom razvoja modela, obuka je usko grlo. Međutim, nakon što je model implementiran, njegov zadatak je da generiše predviđanja, pa je inferencija usko grlo. Inferencija je proces korišćenja obučenog modela za donošenje predviđanja ili zaključaka iz novih, neviđenih podataka. Istraživanje obično prioritizuje brzu obuku, dok proizvodnja obično prioritizuje brzu inferenciju.

Jedna posledica ovoga je da istraživanje prioritizuje visoku propusnost (eng. throughput), dok proizvodnja prioritizuje nisku latenciju (eng. latency). U slučaju da vam je potrebno osveženje, latencija se odnosi na vreme koje je potrebno da se dobije rezultat nakon što se primi upit. Propusnost se odnosi na broj upita koji se obradi u određenom vremenskom periodu.

Kritika tabela rangiranja u mašinskom učenju

U poslednjih nekoliko godina bilo je mnogo kritika tabela rangiranja u mašinskom učenju, kako takmičenja poput Kagglea, tako i tabela rangiranja u istraživanjima kao što su ImageNet ili GLUE.

Očigledan argument je da su u ovim takmičenjima potrebni mnogi teški koraci za izgradnju sistema za mašinsko učenje već obavljeni za vas.[16]

Manje očigledan argument je da zbog scenarija testiranja s više hipoteza koji se dešava kada više timova testira na istom skupu test podataka, model može da se pokaže boljim od drugih samo slučajno.[17]

Neusklađenost interesa između istraživanja i proizvodnje (stvarne primene) primećena je od strane istraživača. U radu sa EMNLP 2020, Ethayarajh i Jurafsky su argumentovali da su testovi pomogli napredak u obradi prirodnog jezika (NLP) tako što su podstakli stvaranje tačnijih modela na štetu drugih kvaliteta koje cenimo, kao što su kompaktnost, pravičnost i energetska efikasnost.[18]



Sukob terminologije

Neki autori prave razliku između latencije i vremena odgovora. Prema Martinu Klepmanu u njegovoj knjizi *Designing Data-Intensive Applications*, „Vreme odgovora je ono što klijent vidi: osim stvarnog vremena potrebnog za obradu zahteva (vreme usluge), ono uključuje i kašnjenja u mreži i čekanja u redovima. Latencija je trajanje tokom kojeg zahtev čeka da bude obrađen – tokom kojeg je zahtev latentan, čekajući uslugu."[19]

U ovoj knjizi, radi pojednostavljenja diskusije i da bismo bili dosledni terminologiji koja se koristi u zajednici za mašinsko učenje, koristimo latenciju kako bismo se odnosili na vreme odgovora, tako da latencija zahteva se odnosi na vreme od trenutka kada se zahtev šalje do trenutka kada se odgovor primi.

Na primer, prosečna latencija servisa Google Translate je prosečno vreme koje je potrebno od trenutka kada korisnik klikne na Translate do prikaza prevoda, a protok je broj zahteva koje obrađuje i servisira u sekundi.

[16] Julia Evans, „Machine Learning Isn't Kaggle Competitions" 2014, *https://oreil.ly/p8mZq*.

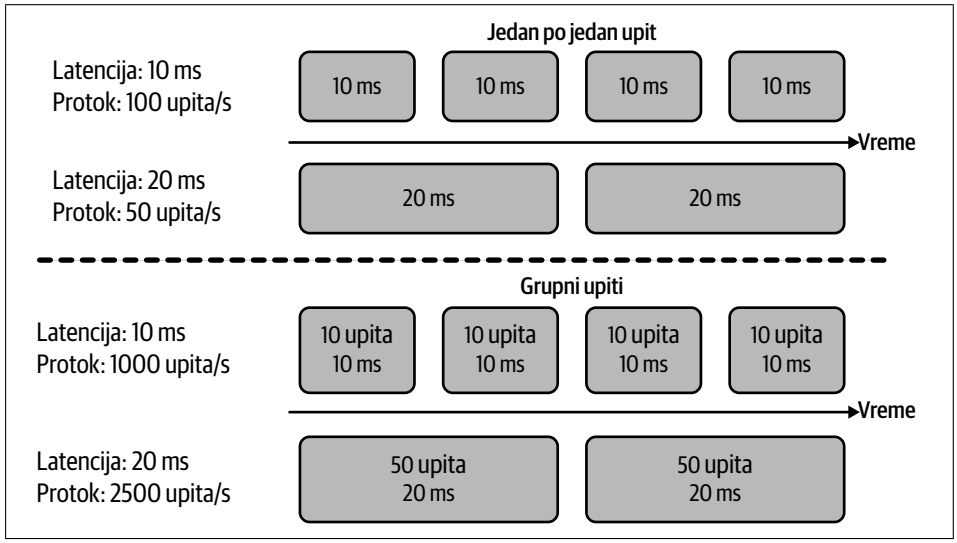
[17] Lauren Oakden-Rayner, „AI Competitions Don't Produce Useful Models" September 19, 2019, *https://oreil.ly/X6RlT*.

[18] Kawin Ethayarajh i Dan Jurafsky, „Utility Is in the Eye of the User: A Critique of NLP Leaderboards" EMNLP, 2020, *https://oreil.ly/4Ud8P*.

[19] Martin Kleppmann, *Designing Data-Intensive Applications* (Sebastopol, CA: O'Reilly, 2017).

Ako vaš sistem uvek obrađuje samo jedan zahtev istovremeno, veća latencija znači manji protok. Ako je prosečna latencija 10 ms, što znači da je potrebno 10 ms za obradu zahteva, protok je 100 zahteva u sekundi. Ako je prosečna latencija 100 ms, protok je 10 zahteva u sekundi.

Međutim, zato što većina modernih distribuiranih sistema grupiše zahteve kako bi ih zajedno, često istovremeno, obradila, *veća latencija takođe može značiti veći protok*. Ako obrađujete 10 zahteva istovremeno i potrebno je 10 ms da se izvrši grupa, prosečna latencija i dalje iznosi 10 ms, ali je protok sada 10 puta veći – 1.000 zahteva u sekundi. Ako obrađujete 50 zahteva istovremeno i potrebno je 20 ms da se izvrši grupa, prosečna latencija je sada 20 ms, a protok je 2.500 zahteva u sekundi. I latencija i protok su povećani! Razlika u odnosu latencije i protoka prilikom obrade zahteva jedan po jedan i obrade zahteva u grupama prikazana je na Slici 1-4.



Slika 1-4. Kada se obrađuju zahtevi jedan po jedan, veća latencija znači manji protok. Međutim, obrada zahteva u grupama takođe može značiti veći protok.

Ovo postaje još komplikovanije ako želite grupisati online zahteve. Grupisanje (eng. batching) upita zahteva da vaš sistem sačeka dovoljno zahteva da stignu u grupi pre nego što ih obradi, što dodatno povećava latenciju.

U istraživanju, važnije vam je koliko uzoraka možete obraditi u sekundi (protok), a manje koliko vremena traje obrada svakog uzorka (latencija). Spremni ste da povećate latenciju kako biste povećali protok, na primer, agresivnim grupisanjem.

Međutim, kada implementirate svoj model u stvarnom svetu, latencija ima veliku važnost. U 2017. godini, istraživanje kompanije Akamai je pokazalo da kašnjenje

od 100 ms može smanjiti stope konverzije za 7%.[20] U 2019. godini, Booking.com je otkrio da povećanje latencije od oko 30% košta oko 0,5% u stopama konverzije – „relevantan trošak za naš posao."[21] U 2016. godini, Google je otkrio da će više od polovine korisnika mobilnih uređaja napustiti stranicu ako se učitava duže od tri sekunde.[22] Korisnici danas imaju još manje strpljenja.

Da biste smanjili latenciju u proizvodnji (stvarnoj upotrebi), možda ćete morati smanjiti broj upita koje možete obraditi na istom hardveru istovremeno. Ako vaš hardver može obraditi mnogo više upita odjednom, njegovo korišćenje za obradu manje upita znači da ne iskorišćavate vaš hardver u punom kapacitetu, što povećava troškove obrade svakog upita.

Kada razmišljate o latenciji, važno je imati na umu da latencija nije jedan broj, nego distribucija. Iskušenje je da pojednostavite ovu distribuciju koristeći jedan broj poput prosečne (aritmetičke sredine) latencije svih zahteva unutar vremenskog prozora, ali ovaj broj može biti zavaravajući. Zamislite da imate 10 zahteva čije su latencije 100 ms, 102 ms, 100 ms, 100 ms, 99 ms, 104 ms, 110 ms, 90 ms, 3000 ms, 95 ms. Prosečna latencija iznosi 390 ms, što čini da vaš sistem izgleda sporije nego što zapravo jeste. Možda se dogodila mrežna greška koja je usporila jedan zahtev u odnosu na ostale trebalo bi istražiti taj problematični zahtev.

Obično je bolje razmišljati u percentilima, jer vam oni nešto govore o određenom procentu vaših zahteva. Najčešći percentil je 50. percentil, skraćeno p50. Takođe se naziva medijanom. Ako je medijan 100 ms, polovina zahteva traje duže od 100 ms, a polovina zahteva traje manje od 100 ms.

Viši percentili takođe vam pomažu da otkrijete izuzetke, koji mogu biti simptomi nečega što nije u redu. Tipično, percentili koje ćete želeti da pogledate su p90, p95 i p99. 90. percentil (p90) za 10 zahteva iznad iznosi 3.000 ms, što je anomalija (eng. outlier).

Viši percentili su važni za razmatranje, jer iako čine mali procenat vaših korisnika, ponekad mogu biti najvažniji korisnici. Na primer, na Amazonovoj veb stranici, korisnici sa najsporijim zahtevima često su oni koji imaju najviše podataka na svojim nalozima jer su napravili mnogo kupovina – to jest, najvredniji su korisnici.[23]

---

[20] Akamai Technologies, *Akamai Online Retail Performance Report: Milliseconds Are Critical*, April 19, 2017, *https://oreil.ly/bEtRu*.

[21] Lucas Bernardi, Themis Mavridis i Pablo Estevez, „150 Successful Machine Learning Models: 6 Lessons Learned at Booking.com" KDD '19, August 4–8, 2019, Anchorage, AK, *https://oreil.ly/G5QNA*.

[22] „Consumer Insights" Think with Google, *https://oreil.ly/JCp6Z*.

[23] Kleppmann, *Designing Data-Intensive Applications*.

Uobičajena praksa je korišćenje visokih percentila za specificiranje zahteva za performanse vašeg sistema; na primer, menadžer proizvoda može zadati da 90. percentil ili 99,9. percentil latencije sistema mora biti ispod određenog broja.

Podaci

U fazi istraživanja, skupovi podataka s kojima radite često su čisti i dobro formatirani, omogućavajući vam da se usredsredite na razvoj modela. Prirodno su statični kako bi zajednica mogla da ih koristi za testiranje novih arhitektura i tehnika. To znači da mnogi ljudi mogu da koriste i diskutuju o istim skupovima podataka, a osobine tih skupova su poznate. Čak možete pronaći skript otvorenog koda za obradu i unos podataka direktno u vaše modele.

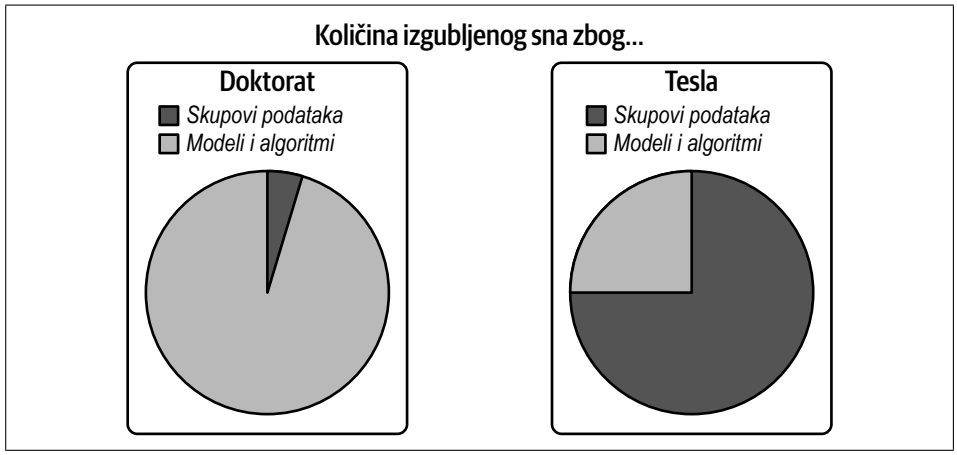
U proizvodnji, podaci, ako su dostupni, mnogo su neuredniji. Šum je prisutan, verovanto su nestrukturisani i konstantno se menjaju. Verovatno su pristrasni (eng. biased), a verovatno ne znate na koji način su pristrasni. Oznake, ako postoje, mogu biti retke, neizbalansirane ili netačne. Promene u projektu ili poslovnim zahtevima mogu zahtevati ažuriranje nekih ili svih vaših postojećih oznaka. Ako radite sa korisničkim podacima, takođe ćete morati brinuti o pitanjima privatnosti i regulativi. U odeljku „Studija slučaja II: Opasnost od 'anonimizovanih' podataka" razmotrićemo studiju slučaja u kome su korisnički podaci nedovoljno obrađeni (strana 346).

U istraživanju, uglavnom radite sa istorijskim podacima, npr. podacima koji već postoje i čuvaju se negde. U proizvodnji, verovatno ćete morati raditi i sa podacima koji se konstantno generišu od strane korisnika, sistema i podataka trećih strana.

Slika 1-5 je adaptirana sjajna grafika autora Andreja Karpathya, direktora za veštačku inteligenciju u kompaniji Tesla, koja ilustruje probleme sa podacima sa kojima se suočavao tokom svog doktorata u poređenju sa vremenom provedenim u kompaniji Tesla.

Pravičnost

Tokom faze istraživanja, model se još uvek ne koristi na ljudima, pa je istraživačima lako da odlože razmatranje pravičnosti (eng. fairness) kao naknadnu brigu: „Hajde da prvo postignemo najbolje rezultate i da se brinemo o pravičnosti kada dođemo do proizvodnje." Kad dođe do proizvodnje, već je kasno. Ako optimizujete modele za veću tačnost ili nižu latenciju, možete pokazati da vaši modeli nadmašuju vrhuske rezultate. Ali, u trenutku pisanja ove knjige, ne postoji ekvivalentan vrhuskih rezultata u metrikama pravičnosti.



Slika 1-5. Podaci u istraživanju nasuprot podacima u proizvodnji. Izvor: Adaptirana slika autora Andreja Karpathyja[24]

Vi ili neko u vašem životu već ste možda bili žrtva pristrasnih matematičkih algoritama, a da to niste znali. Vaša prijava za kredit može biti odbijena jer ML algoritam uzima u obzir vašu poštansku šifru, koja nosi pristrasnost prema socijalnom statusu. Vaša biografija može biti rangirana niže jer sistem rangiranja poslodavaca obraća pažnju na vaš pravopis. Vaša hipoteka može imati višu kamatnu stopu jer se delimično oslanja na kreditne ocene, koje favorizuju bogate i kažnjavaju siromašne. Druge primere pristrasnosti u stvarnom svetu putem ML algoritama nalazimo u algoritmima za prediktivno obavljanje policijskih poslova, testovima ličnosti koje sprovode potencijalni poslodavci i rangiranju na fakultetima.

U 2019. godini, „Berkeley istraživači su otkrili da su kako lice u lice, tako i onlajn zajmodavci odbili ukupno 1,3 miliona kreditno sposobnih crnih i latino aplikanta između 2008. i 2015. godine." Kada su istraživači „koristili prihod i kreditne ocene odbačenih prijava, ali su izbrisali identifikatore rase, prijava za hipoteku je prihvaćena."[25] Za još više uznemirujućih primera, preporučujem knjigu Cathy O'Neil „Weapons of Math Destruction".[26]

ML algoritmi ne predviđaju budućnost, već dešifuju prošlost, čime se nastavlja pristrasnosti u podacima a i više od toga. Kada se ML algoritmi primene u velikom obimu, mogu diskriminisati ljude u velikom obimu. Ako ljudski operater može donositi široke sudove o samo nekoliko pojedinaca odjednom, ML

[24] Andrej Karpathy, „Building the Software 2.0 Stack," Spark+AI Summit 2018, video, 17:54, *https://oreil.ly/Z21Oz*.

[25] Khristopher J. Brooks, „Disparity in Home Lending Costs Minorities Millions, Researchers Find" *CBS News*, November 15, 2019, *https://oreil.ly/UiHUB*.

[26] Cathy O'Neil, *Weapons of Math Destruction* (New York: Crown Books, 2016).

algoritam može doneti široke sudove o milionima u deliću sekunde. To može posebno štetiti članovima manjinskih grupa jer pogrešna klasifikacija kod njih može imati samo blagi uticaj na ukupne metrike performansi modela.

Ako algoritam već može da tačno predviđa 98% populacije, a poboljšanje predviđanja za preostalih 2% bi donelo višekratne troškove, neke kompanije, nažalost, mogu odlučiti da to ne rade. Tokom istraživanja McKinsey & Company iz 2019. godine, samo 13% velikih kompanija koje su anketirane izjavile su da preduzimaju korake za ublažavanje rizika u vezi sa pravičnošću i jednakošću, kao što su algoritamske pristrasnosti i diskriminacija.[27] Međutim, ovo se brzo menja. O pravičnosti i drugim aspektima odgovorne veštačke inteligencije ćemo govoriti u poglavlju 11.

Interpreterabilnost

U početku 2020. godine, dobitnik Turingove nagrade, profesor Geoffrey Hinton, postavio je žestoko raspravljano pitanje o važnosti interpreterabilnosti u ML sistemima. „Pretpostavimo da imate rak i da morate da birate između AI hirurga crne kutije koji ne može objasniti kako radi, ali ima stopu izlečenja od 90% i ljudskog hirurga sa stopom izlečenja od 80%. Da li biste želeli da AI hirurg bude nelegalan?“[28]

Nekoliko nedelja kasnije, kada sam postavila ovo pitanje grupi od 30 izvršnih direktora u oblasti tehnologije u javnim nontehnološkim kompanijama, samo polovina njih bi želela da ih operiše veoma efikasan, ali nesposoban da objasni AI hirurg. Druga polovina je želela ljudskog hirurga.

Iako se većina nas oseća udobno koristeći mikrotalasnu pećnicu bez razumevanja kako funkcioniše, mnogi se još uvek ne osećaju isto prema veštačkoj inteligenciji, posebno ako ta veštačka inteligencija donosi važne odluke o njihovim životima.

Budući da se većina istraživanja u mašinskom učenju i dalje ocenjuje na osnovu jednog cilja, performansi modela, istraživači nisu motivisani da rade na interpretaciji modela. Međutim, interpretacija nije samo opciona za većinu slučajeva upotrebe mašinskog učenja u industriji, već je obavezna.

Prvo, interpretacija je važna za korisnike, kako poslovne lidere tako i krajnje korisnike, kako bi razumeli zašto je doneta određena odluka kako bi mogli da veruju modelu i otkriju potencijalne pristrasnosti koje su prethodno pomenute.[29] Drugo, važno je da programeri mogu da otklanjaju probleme i poboljšavaju model.

---

[27] Stanford University Human-Centered Artificial Intelligence (HAI), *The 2019 AI Index Report*, 2019, *https://oreil.ly/xs8mG*.

[28] Tweet by Geoffrey Hinton (@geoffreyhinton), February 20, 2020, *https://oreil.ly/KdfD8*.

[29] Za određene slučajeve u nekim zemljama, korisnici imaju „pravo na objašnjenje“: pravo da dobiju objašnjenje za rezultat algoritma.

Samo zato što je interpretacija obavezna ne znači da je svako radi. Kako je 2019. godine, samo 19% velikih kompanija radi na poboljšanju objašnjivosti svojih algoritama.[30]

Diskusija

Neke osobe bi mogle tvrditi da je u redu poznavati samo akademsku stranu mašinskog učenja jer postoji mnogo poslova u istraživanju. Prvi deo – da je u redu poznavati samo akademsku stranu mašinskog učenja – jeste tačan. Drugi deo je netačan.

Iako je važno pratiti čisto istraživanje, većina kompanija to ne može da priušti osim ako ne dovodi do primene kratkoročnih poslovnih aplikacija. Ovo je posebno tačno sada kada je istraživačka zajednica usvojila pristup „veće, bolje". Često, novi modeli zahtevaju ogromnu količinu podataka i desetine miliona dolara samo za računarske resurse.

Pošto istraživanje u oblasti mašinskog učenja i dostupni modeli postaju pristupačniji, sve više ljudi i organizacija će želeti da pronađe primene za njih, što povećava potražnju za mašinskim učenjem u proizvodnji.

Većina poslova vezanih za mašinsko učenje će biti, i već jesu, u stvarnoj, proizvodnoj upotrebi mašinskog učenja.

Sistemi mašinskog učenja nasuprot tradicionalnog softvera

Pošto je mašinsko učenje deo inženjeringa softvera (software engineering, SWE), a softver se uspešno koristi u proizvodnji već više od pola veka, neki se možda pitaju zašto jednostavno ne preuzmemo proverene najbolje prakse u inženjeringu softvera i primenimo ih na mašinsko učenje.

To je odlična ideja. Zapravo, proizvodnja ML sistema bi bila mnogo bolje mesto ako bi stručnjaci za mašinsko učenje bili bolji softverski inženjeri. Mnogi tradicionalni alati za softversko inženjerstvo mogu se koristiti za razvoj i implementaciju aplikacija za mašinsko učenje.

Međutim, mnogi izazovi karakteristični za ML aplikacije su jedinstveni i zahtevaju svoje alate. U softverskom inženjeringu postoji osnovna pretpostavka da su kod i podaci odvojeni. Zapravo, u softverskom inženjeringu želimo da stvari ostanu što modularnije i odvojene koliko je moguće (vidite Wikipedia stranicu o odvajanju nadležnosti – separation of concerns).

Nasuprot tome, sistemi za mašinsko učenje sastoje se delimično od koda, delimično od podataka i delimično od artefakata koji su stvoreni iz ta dva dela. Trend u poslednjem deceniji pokazuje da aplikacije razvijene sa najboljim podacima

[30] Stanford HAI, *The 2019 AI Index Report*.

pobeđuju. Umesto da se fokusirate na unapređenje algoritama za mašinsko učenje, većina kompanija će se fokusirati na unapređenje svojih podataka. Budući da se podaci mogu brzo menjati, aplikacije za mašinsko učenje moraju biti prilagodljive promenljivom okruženju, što može zahtevati brže cikluse razvoja i implementacije.

U tradicionalnom softverskom inženjeringu, potrebno je samo fokusirati se na testiranje i verzionisanje koda. Sa mašinskim učenjem, moramo takođe testirati i verzionisati naše podatke i to je teži deo. Kako verzionisati velike skupove podataka? Kako znati da li je uzorak podataka dobar ili loš za vaš sistem? Svi uzorci podataka nisu jednaki – neki su vredniji za vaš model od drugih. Na primer, ako je vaš model već treniran na milion skeniranja normalnih pluća i samo hiljadu skeniranja kancerogenih pluća, skeniranje kancerogenih pluća mnogo je vrednije od skeniranja normalnih pluća. Ravnopravno prihvatanje svih dostupnih podataka može nauditi performansama vašeg modela i čak ga učiniti podložnim napadima sa lošim podacima.[31]

Veličina modela za mašinsko učenje predstavlja još jedan izazov. Kao što je slučaj u 2022. godini, uobičajeno je da modeli za mašinsko učenje imaju stotine miliona, ako ne i milijarde parametara, što zahteva gigabajte radne memorije (RAM) da se učitaju u memoriju. Za nekoliko godina, milijarda parametara može delovati zastarelo – kao „Možete li verovati da je računar koji je poslao ljude na Mesec imao samo 32 MB RAM-a?"

Međutim, trenutno je veliki izazov uvesti ove velike modele u proizvodnju, posebno na uređajima u kritičnim primenama,[32] i postavlja se pitanje kako postići da ovi modeli rade dovoljno brzo i da budu korisni. Model za automatsko dovršavanje je beskoristan ako vreme koje mu treba da predloži sledeći znak traje duže od vremena koje vam je potrebno da kucate.

Takođe nije trivijalno pratiti i otkrivati greške ovih modela koji rade u proizvodnji. Kako modeli za mašinsko učenje postaju složeniji, uz nedostatak nadgledivosti njihov rad, teško je utvrditi šta je krenulo loše ili biti brzo obavešten kada je nešto krenulo loše.

Dobra vest je da se ovi inženjerski izazovi rešavaju brzim koracima. Još 2018. godine, kada je prvi put objavljen rad o Bidirekcionom enkoderu reprezentacije iz transformersa (Encoder Representations from Transformers, BERT), ljudi su govorili kako je BERT suviše velik, suviše složen i suviše spor da bi bio praktičan. Prethodno trenirani veliki BERT model ima 340 miliona parametara i zauzima

[31] Xinyun Chen, Chang Liu, Bo Li, Kimberly Lu i Dawn Song, „Targeted Backdoor Attacks on Deep Learning Systems Using Data Poisoning" *arXiv*, December 15, 2017, *https://oreil.ly/OkAjb*.

[32] Uređaje u kritičnim primenama pokrićemo u Poglavlju 7.

1.35 GB.[33] Brzo napredovanje dve godine kasnije, BERT i njegove varijacije su već korišćeni u gotovo svakoj engleskoj pretrazi na Googlu.[34]

Rezime

Ovo početno poglavlje ima za cilj da čitaocima pruži razumevanje onoga što je potrebno da ML uđe u stvarni svet. Počeli smo turom kroz širok spektar slučajeva upotrebe ML-a u današnjoj proizvodnji. Dok su većini ljudi poznate primene ML-a u potrošačkim aplikacijama, većina slučajeva upotrebe ML-a odnosi se na poslovne aplikacije. Takođe smo diskutovali o tome kada bi rešenja zasnovana na ML-u bila adekvatna. Iako ML može da reši mnoge probleme veoma dobro, ne može rešiti sve probleme i sigurno nije prikladan za sve probleme. Međutim, za probleme koje ML ne može rešiti, moguće je da ML bude jedan deo rešenja.

Ovo poglavlje ističe razlike između ML-a u istraživanju i ML-a u stvarnoj upotrebi, tj. proizvodnji. Razlike sadrže uključenost zainteresovanih strana, računarski prioritet, karakteristike korišćenih podataka, ozbiljnost problema pravičnosti i zahteva za pojašnjivost (intepretabilnost). Ova sekcija je najkorisnija onima koji dolaze iz akademske sfere u proizvodni ML. Govorili smo o tome kako se ML sistemi razlikuju od tradicionalnih softverskih sistema, što motiviše potrebu za ovom knjigom.

ML sistemi su složeni i sastoje se od mnogo različitih komponenti. Naučnici podatka (eng. data scientist) i inženjeri za ML koji rade sa ML sistemima u proizvodnji verovatno će zaključiti da se fokusiranje samo na deo sa ML algoritmima ne dovodi do cilja. Važno je znati i o drugim aspektima sistema, uključujući stek podataka, implementaciju, praćenje, održavanje, infrastrukturu itd. Ova knjiga pristupa razvoju ML sistema sa sistemskim pristupom, što znači da ćemo razmatrati sve komponente sistema holistički, umesto da se samo posmatraju ML algoritmi. Detalje o tome šta ova holistička metoda podrazumeva, daćemo u narednom poglavlju.

[33] Jacob Devlin, Ming-Wei Chang, Kenton Lee i Kristina Toutanova, „BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding“ *arXiv*, October 11, 2018, *https://oreil.ly/TG3ZW*.

[34] Google Search On, 2020, *https://oreil.ly/M7YjM*.