

UMETNA INTELIGENCA ZA TELEBANE – PLATFORME STROJNEGA UČENJA

SAŠO KARAKATIČ, GREGA VRBANČIČ, JERNEJ FLISAR IN VILI PODGORELEC

Povzetek: Uporaba procesov strojnega učenja za večino podjetij predstavlja velik izziv, saj zahteva spremembo obstoječih poslovnih procesov, drag in talentiran kader ter njihovo dodatno izobrazbo. Najnovejše tehnike strojnega učenja predstavljajo velik napredek pri obdelavi podatkov, vendar je področje preobširno za spoznavanje »čez vikend« ali »v popoldanskem času« za ljudi, ki niso strokovnjaki s področja podatkovne znanosti (angl. Data Science). Če gradimo nov Uber, Netflix ali Facebook je zahtevnost res ogromna, ampak uporaba obstoječih storitev strojnega učenja znatno olajša ta postopek. Z uporabo oblčnih storitev strojnega učenja lahko začnemo graditi svoje prve modele umetne inteligence in tako vključiti dragocene napovedi in inteligenco v obstoječe sisteme. V članku bomo razpravljali o obstoječih platformah strojnega učenja – tako o spletnih kakor tudi o lokalnih.

Ključne besede: • strojno učenje • podatkovna znanost • platforme • storitve

NASLOV AVTORJEV: dr. Sašo Karakatič, docent, Univerza v Mariboru, Fakulteta za elektrotehniko, računalništvo in informatiko, Koroška cesta 46, 2000 Maribor, Slovenija, e-pošta: saso.karakatic@um.si. Grega Vrbančič, mladi raziskovalec, Univerza v Mariboru, Fakulteta za elektrotehniko, računalništvo in informatiko, Koroška cesta 46, 2000 Maribor, Slovenija, e-pošta: grega.vrbancic@um.si. Jernej Flisar, asistent, Univerza v Mariboru, Fakulteta za elektrotehniko, računalništvo in informatiko, Koroška cesta 46, 2000 Maribor, Slovenija, e-pošta: jernej.flisar@um.si. dr. Vili Podgorelec, redni profesor, Univerza v Mariboru, Fakulteta za elektrotehniko, računalništvo in informatiko, Koroška cesta 46, 2000 Maribor, Slovenija, e-pošta: vili.podgorelec@um.si.

1. UVOD

Strojno učenje predstavlja velik mejnik pri analizi velepodatkov, vendar je njegova uporaba zastrašujoča za ljudi, ki niso strokovnjaki za področje podatkovne znanosti in niso tehnično podkovani iz algoritmov strojnega učenja. Platforme podatkovne znanosti in strojnega učenja uporabljajo znanstveniki in razvijalci pri opravljanju nalog v celotnem podatkovnem in analitičnem procesu. Vključujejo naloge, ki se nanašajo na kreacijo in zajemanje podatkov, pripravo podatkov v primerno obliko, interaktivno raziskovanje in vizualizacijo, napredno modeliranje, testiranje, učenje, uporabo ter inženiring modelov strojnega učenja.

Izzivi uporabe podatkovne znanosti in platform za strojno učenje niso omejeni na izbiro prave platforme za zadostitev analitičnih potreb organizacij. Hkrati je potrebno obravnavati podatke, ljudi in procese, kar predstavlja še dodatni nivo težavnosti pri upravljanju organizacije. Upravljanje informacij ima zato pomembno vlogo pri zagotavljanju, da modeli temeljijo na najnovejših znanstvenih dosežkih tega področja in dobrih praksah, ki pridejo le z izkušnjami. Mnogokrat pa organizacije ne zaposlujejo strokovnjakov, ki bi posedovali ta znanja in izkušnje. Namesto najema zunanjih človeških virov, zaposlovanja in formiranja ekip podatkovnih znanstvenikov, je vedno bolj popularna uporaba namenskih storitev, ki so enostavne za uporabo in rešujejo specializirane naloge podatkovne znanosti.

Včasih so algoritme in ogrodja strojnega učenja večinoma uporabljali znanstveniki, tehnološki strokovnjaki ali domenski strokovnjaki. Vendar pa vse več organizacij zdaj uporablja storitve strojnega učenja, ki so vse bolj dostopne širšemu spektru razvijalcev in raziskovalcev. Podobno, kakor tradicionalne spletne storitve pomagajo razvijalcem ustvariti aplikacije, tako tudi storitve strojnega učenja (angl. *machine learning as a service* ali *MLaaS*) omogočajo enostavno uporabo strojnega učenja povprečnemu razvijalcu. Storitve strojnega učenja prikrivajo kompleksnost pri ustvarjanju in uporabi modelov strojnega učenja, tako da se lahko razvijalci osredotočijo na pripravo podatkov, uporabniško izkušnjo, oblikovanje, eksperimentiranje in uporabo znanja ter odkrivanje vzorcev iz podatkov.

Storitve strojnega učenja ponujajo abstraktni sloj za razvijalce, ki jim omogoča integracijo strojnega učenja v aplikacije v realnem svetu, ne da bi morali skrbeti za skaliranje algoritmov na svoji infrastrukturi in se ukvarjati s podrobnostmi metod strojnega učenja. Razvijalci aplikacij vedno iščejo različne načine za olajšanje življenja svojih uporabnikov z uvedbo novih in inovativnih funkcij, ki uporabnikom omogočajo prihranek časa. To je razlog za priljubljenost storitev strojnega učenja pri razvijalcih aplikacij. Nekateri standardni primeri teh storitev vključujejo inteligentno označevanje, priporočila, napovedovanje, segmentiranje in kategoriziranje. [1]

Najbolj pomemben del uporabe storitev strojnega učenja je prepoznavna poslovnega problema, ki ga je treba rešiti in oblikovanje toka podatkov skozi celotno aplikacijo. Vsake težave ni mogoče rešiti s storitvami strojnega učenja, zato je pomembno identificirati problem, cilj in scenarij uporabe metod strojnega učenja. Mnogokrat namreč uporaba takih storitev ni mogoča, saj zastavljeni primeri uporabe zahtevajo aplikacijo podatkovne znanosti, ki jih ponujene storitve ne omogočajo.

Tekom naše raziskave smo identificirali tri tipe uporabnikov storitev in platform strojnega učenja:

- (1) **Tipični razvijalci programske opreme, hobi znanstveniki, podatkovni raziskovalci in novinarji**, ki potrebujejo podatkovno znanost in strojno učenje za reševanje specifičnega problema ali primera uporabe.
- (2) **Operativni delavci, razvijalce strojne opreme in odločevalci**, ki vsakodnevno sprejemajo odločitve na podlagi modelov strojnega učenja.
- (3) **Visokokvalificirani inženirji strojnega učenja in podatkovni znanstveniki**, ki oblikujejo poskuse in učijo modele strojnega učenja za predstavljanje in optimizacijo poslovnih odločitev.

Vsak tip uporabnikov lahko uporabi platformo strojnega učenja, ki je prilagojena in optimizirana za njegov primer uporabe. V sledečem poglavju so opisane specializirane storitve, ki rešujejo ozko določen primer uporabe in so namenjene specifičnim zahtevam. Temu sledi poglavje, ki predstavi storitve in

platforme z vizualnim uporabniškim vmesnikom in so zasnovane bolj široko – potrebujejo prilagoditev in posledično poznavanje strojnega učenja. Zadnje poglavje pa predstavi platforme, ki so osnova vsem storitvam strojnega učenja in jih lahko uporabimo tudi v naših aplikacijah, ampak za to zahtevajo tako odlično poznavanje algoritmov strojnega učenja kakor tudi programiranja v splošnem.

2. SPECIALIZIRANE STORITVE STROJNEGA UČENJA

V zadnjih nekaj letih je v podjetjih prišlo do premika paradigme gradnje tehnoloških skladov v smeri platform in mikrostoritev. Ta premik je bil omogočen zaradi razcveta računalništva v oblaku, še posebej pa zaradi povečane rasti števila javnih oblačnih storitev, ki jih ponujajo glavni igralci na področju računalništva v oblaku (Amazon, Google, Microsoft). Omenjena podjetja so močno poudarjala in zagovarjala poslovni model "kot storitev" (angl. *as a service*), kateri zunanjim podjetjem omogoča izbiro zgolj tistih mikrostoritev, ki so za njih potrebne oz. nujne.

Infrastruktura kot storitev (angl. *infrastructure as a service* ali *IaaS*) ter platforma kot storitev (angl. *platform as a service* ali *PaaS*) sta dve najpogosteje uporabljeni storitveni ponudbi ponudnikov računalništva v oblaku. S hitrim napredkom in razvojem strojnega učenja in umetne inteligence v splošnem pa se v zadnjem letu ali dveh na trgu vedno bolj razširjajo namenske – specializirane storitve. Za takšne specializirane storitve strojnega učenja sta se uveljavila dva izraza in sicer strojno učenje kot storitev (angl. *machine learning as a service* ali *MLaaS*) ter umetna inteligenca kot storitev (angl. *artificial intelligence as a service* – *AIaaS*). Z integracijo orodij in storitev umetne inteligence oz. strojnega učenja lahko podjetja izboljšajo zmožnosti svojih produktov, bolje komunicirajo s strankami, racionalizirajo poslovanje in ustvarjajo natančne napovedne poslovne strategije. MLaaS storitve omogočajo podjetjem brez namenskih oddelkov za strojno učenje oz. umetno inteligenco, da s svojim obstoječim kadrom – razvijalci, hitro in učinkovito oplemenitijo in nadgradijo svoje produkte. Podatki so gonilna sila strojnega učenja in ker velika podjetja – glavni igralci na področju IaaS in PaaS – ustvarjajo ogromne količine podatkov do katerih imajo tudi dostop, obenem pa imajo tudi veliko računskih virov, so sami zmožni zgraditi in naučiti modele, kar jim omogoča, da te ponujajo v obliki MLaaS zunanjim podjetjem. Takšne storitve torej vsebujejo že v naprej pripravljene algoritme in modele, za katere bi sicer potrebovali ogromno virov, da bi jih zgradili popolnoma od začetka. Hitrost, enostavnost in stroškovna učinkovitost integracije z obstoječimi produkti so tako ključne prednosti MLaaS v primerjavi s klasičnimi pristopi k strojnemu učenju oz. umetni inteligenci. [2]




MLaaS oz. strojno učenje kot storitev je krovna definicija avtomatiziranih in delno avtomatiziranih oblačnih platform, ki naslavljajo problematiko večine infrastrukturnih problemov kot so predprocesiranje podatkov, učenje modelov, evalvacija naučenih modelov in napovedovanje s pomočjo takšnih modelov. Rezultati napovednih modelov so z interno infrastrukturo podjetja povezani preko REST API povezav. [4]

Amazon Machine Learning Services, Microsoft Azure Machine Learning in Google Cloud AI so trije vodilni ponudniki oblačnih MLaaS storitev, ki omogočajo hitro učenje modelov in njihovo namestitve z malo ali celo brez domenskega znanja s področja podatkovne znanosti in strojnega učenja. Ponudniki MLaaS storitev ponujajo širok nabor funkcionalnosti teh storitev vse od prepoznave obrazov, zaznave predmetov iz slik, pretvarjanja govora v besedilo in obratno, pa vse do namenskih funkcionalnosti kot so pogovorni roboti (angl. *chatbot*) in podobno. V splošnem lahko glavne funkcionalnosti vodilnih podjetij na področju MLaaS razdelimo v tri osnovne kategorije (glej Slika 1):

- procesiranje govora in besedil,
- analiza slik in video posnetkov, ter
- namenske storitve.

Kategorija procesiranja govora in besedil je najbolj zastopana pri treh vodilnih ponudnikih. V to kategorijo spadajo storitve, z integracijo katerih lahko podjetje svoje produkte nadgradi na nivoju

interakcije z uporabniki. Z apliciranjem takšnih storitev lahko na primer omogočimo strojno prevajanje besedila v različne jezike ali pa analizo čustev iz besedila, s pomočjo katere lahko dodatno obogatimo uporabniško izkušnjo. Tehnike procesiranja naravnega besedila so med drugim tudi primarna tehnologija, ki stoji za raznimi pogovornimi roboti ali asistenti. V to skupino storitev pa spadajo tudi storitve, ki omogočajo "razumevanje" govora in so ga zmožne tudi pretvarjati v besedilo. Najbolj poznani primeri aplikacij storitev te kategorije so pametni asistenti **Apple Siri**, **Google Now**, **Microsoft Cortana** ter **Amazon Alexa**.

	Procesiranje govora in besedil	Analiza slik in video posnetkov	Namenske storitve
	Amazon Transcribe Amazon Polly Amazon Comprehend Amazon Translate	Amazon Rekognition Image Amazon Rekognition Video	Amazon Lex
	Speech APIs Language APIs	Face API Computer Vision API	Azure Service Bot Search APIs Knowledge APIs
	Dialogflow Cloud natural language Cloud Speech-to-Text Cloud Text-to-Speech Cloud translation API	Cloud Vision API Cloud Video Intelligence	Cloud Job Discovery

Slika 1. MLaaS storitve ponudnikov Amazon, Microsoft in Google

Storitve iz kategorije analiza slik in video posnetkov v glavnem bazirajo na metodah in tehnikah globokega učenja. S pomočjo teh lahko podjetja svoje produkte nadgradijo z zmožnostjo prepoznavne in "razumevanja" slik ter video posnetkov. Storitve omogočajo analizo vizualnih vsebin vključno z identifikacijo objektov kot tudi klasifikacijo zaznanih objektov. Dodatno storitve omogočajo napredno analizo obraznih mimik subjektov, sledenje subjektom ipd.

V kategorijo namenske storitve so uvrščene storitve, ki so namenjene reševanju točno določenega problema. Na primer **Amazon Lex** in **Azure Service Bot** sta namenski storitvi za gradnjo pogovornega robota s katerim lahko komuniciramo preko govora ali besedila. Microsoftovi skupini aplikacijskih programskih vmesnikov **Search API** in **Knowledge API** med drugim omogočata dostop do funkcionalnosti iskalnika Bing in pa do naprednih funkcionalnosti za upravljanje z znanjem. **Google Cloud Job Discovery** je storitev, ki omogoča kadrovskim službam podjetij "plug and play" dostop do iskalnika Google in njihovih zmogljivosti strojnega učenja, s čimer združujejo celoten ekosistem: strani podjetij za iskanje kadrov, zaposlitvene oglase, sisteme sledenja prošenj za službo in kadrovske agencije.

Ena od največjih prednosti uporabe MLaaS v primerjavi s klasičnimi metodami in tehnikami strojnega učenja je, da MLaaS podjetjem oz. organizacijam omogoča dostop do zmogljive infrastrukture, ki si je sama najverjetneje ne bi mogla ali ne morejo privoščiti. Strojno učenje zahteva veliko računske moči in sistemi, ki zagotavljajo to raven moči, so tradicionalno zelo dragi. Druga prednost, povezana s stroški, je dostop do cenovno ugodne podatkovne hrambe. Količina podatkov kontinuirano raste in mnoga podjetja se odločajo, da je stroškovno učinkoviteje, če podatke hranijo v oblaku. Ko ima podjetje podatke že v oblaku, pa je tudi uporaba MLaaS obstoječega ponudnika oblaknih storitev še bolj smiselna. Ključna prednost MLaaS je hitra in enostavna uporaba ter integracija v lastne produkte, brez potrebnega predhodnega znanja o metodah in tehnikah strojnega učenja.

Kljub številnim prednostim MLaaS se podjetja oz. organizacije, ki uporabljajo te storitve, lahko soočijo tudi z nekaj slabostmi. Največji problem pri uporabi MLaaS je skupen vsem storitvam, ki tečejo v javnem oblaku in to je odvisnost od ponudnika. Podjetja skrbi, da v primeru, da bi uporabljali preveč storitev posameznega ponudnika, prehod na drugega več ne bi bil mogoč. Obenem pa lahko to predstavlja tveganje v primeru, če ponudnik storitev zviša ceno svojih storitev. Vključevanje podatkov iz različnih virov prav tako lahko predstavlja oviro pri uporabi MLaaS. Mnogi produkti, ki uporabljajo strojno učenje, se opirajo na podatke, ti pa navadno prihajajo iz veliko različnih virov. Takšno zbiranje podatkov ter njihovo pred-procesiranje in obdelovanje je lahko težavna naloga ne glede na to ali uporabljamo MLaaS ali ne. [2, 3, 5]

3. STROJNO UČENJE Z UPORABNIŠKIM VMESNIKOM

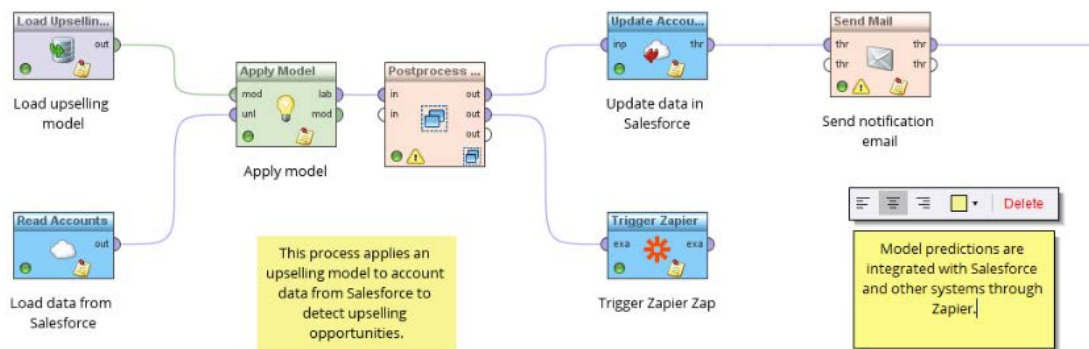
Zaradi razvijajočih se tehnologij strojnega učenja ter možnosti njihove uporabe v realnih projektih in aplikacijah se je pojavila zahteva po enostavni integraciji takih tehnologij na različnih nivojih in domenah aplikacij [5]. Metode strojnega učenja so namreč v domeni znanstvenikov oz. raziskovalcev v raziskovalnih institucijah, ki imajo veliko domenskega znanja, katerega pa razvijalci v podjetjih, ki razvijajo večinoma programske opreme, nimajo. To je podaljšalo razvoj in apliciranje metod strojnega učenja v podjetjih. S časom so se razvila orodja, ki omogočajo enostaven razvoj inteligentnih aplikacij oz. integracijo naprednih metod strojnega učenja, brez naprednega domenskega znanja o samih metodah strojnega učenja.

Orodja za uporabo strojnega učenja pri razvoju oz. apliciranju le-tega lahko uporabljamo tudi preko naprednih grafičnih vmesnikov. Zaradi enostavne uporabe ni večje potrebe po poznavanju različnih algoritmov in metod strojnega učenja. Reševanja domensko specifičnih problemov se lahko lotimo s tako-imenovanim “*povleci in spusti*” načinom, kjer posamezne komponente odlagamo na delovno površino ter jih povežemo v smiseln proces (ang. *workflow*), kateri nam na koncu zgradi zelen model. Model v tem primeru lahko predstavlja naučen klasifikator, ki nam npr. klasificira dokumente v določene kategorije. Vsaka posamezna komponenta v procesu predstavlja posamezno nalogo iz domene strojnega učenja za gradnjo učnega modela. Na voljo imamo številne metode podatkovnega rudarjenja, od priprave in čiščenja podatkov, preko algoritmov gručenja in klasifikacije vse do validacije zgrajenih modelov.

Ker moramo sami definirati cevovod procesa strojnega učenja, potrebujemo ustrezno osnovno znanje s področja podatkovnega rudarjenja in strojnega učenja, saj se morajo posamezne komponente ustrezno uporabiti. Večina orodij sicer preprečuje “*napačno*” uporabo komponent, ki nas vodi pri razvoju modela. Poleg velikega števila posameznih algoritmov in metod imamo pri teh na voljo še mnogo različnih nastavljivih parametrov, ki jih lahko dober strokovnjak s področja podatkovnega rudarjenja optimizira glede na specifično problema, ki ga rešuje, kar lahko občutno izboljša uspešnost razvitega modela. Kljub temu je mogoče dobro razviti oz. aplicirati modele strojnega učenja tudi s pomanjkljivim domenskim znanjem, saj določena orodja sama predlagajo metode ter njihove nekatere privzete vrednosti.

Vzdrževanje modela, zgrajenega z grafičnim vmesnikom, je zelo enostavno. Zaradi vizualizacije je zgrajen model zelo pregleden. Tudi nadgrajevanje, spreminjanje in izboljševanje modela je dokaj

enostavno, saj lahko na enostaven način spremenimo podatkovni vir, iz katerega model črpa učne podatke, ali pa zamenjamo klasifikacijski algoritem, katerega želimo uporabiti. Tipični predstavniki orodij z grafičnim vmesnikom za namen obdelave podatkov so: RapidMiner (primer na Sliki 2), Knime, Azure ML in Waikato Weka.

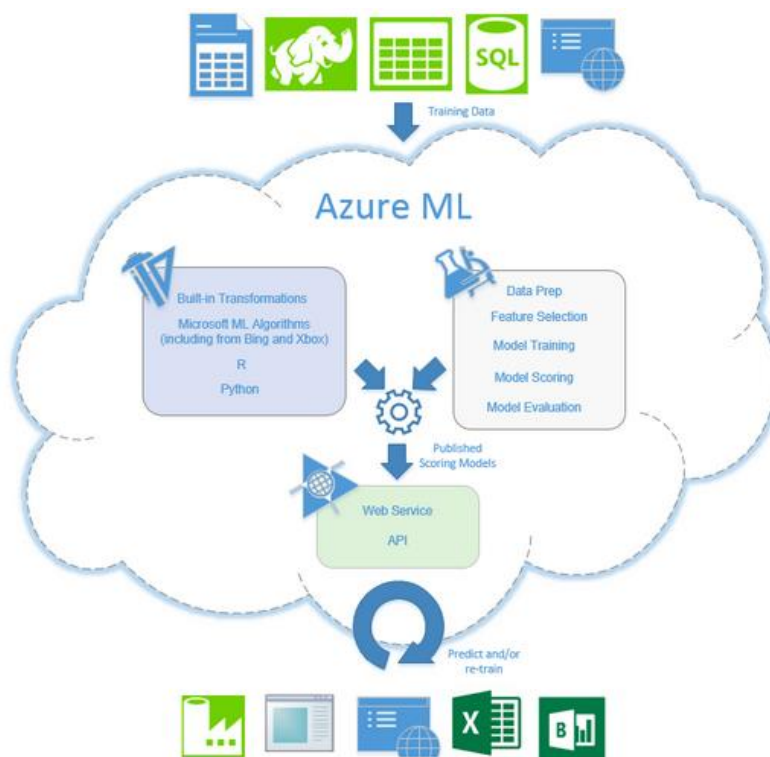


Slika 2. Primer gradnje modela v RapidMiner-ju.

Orodja strojnega učenja z uporabniškim vmesnikom lahko razdelimo v dve skupini, glede na “lokacijo” izvajanja – **lokalno** ali na **strežnikih** (“oblakih”).

Orodja lahko namestimo in zaganjamo lokalno. V tem primeru potrebujemo dovolj dobro oz. problemu primerno strojno opremo, saj so lahko nekateri napredni algoritmi strojno zelo zahtevni. Prednost takšnih orodij je enostavnejša integracija z lastnim, že obstoječim sistemom, ali celo lastnimi aplikacijami. Pomemben je tudi nadzor nad podatki, saj se ti vedno nahajajo na lastni infrastrukturi in tako niso izpostavljeni na zunanjih strežnikih, ki niso v celoti pod našim nadzorom.

Na voljo so tudi orodja, ki se izvajajo na zunanjih strežnikih, ki smo jih že prej imenovali MLaaS. Takšna orodja so primerna predvsem, kadar imamo potrebe po obdelavi velikih količin podatkov, saj so infrastrukturo že pripravljena za izvajanje preko različnih sistemov. Slaba lastnost uporabe orodij v oblaku je izguba popolnega nadzora nad podatki ter izguba nadzora nad konfiguracijo in parametrizacijo nekaterih učnih algoritmov. Primer takšnih orodij so npr: MS Azure ML, AWS Machine Learning, RapidMinerCloud. Slika 3 prikazuje arhitekturni pregled MLaaS.



Slika 3. Primer Arhitekturni pregled MLaaS za storitev Azure ML. [10]

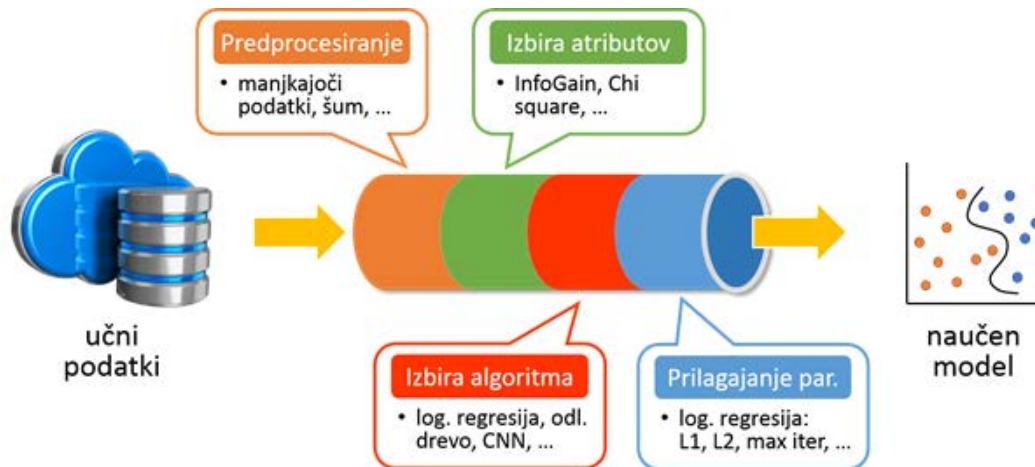
Tabela 1 prikazuje prednosti in slabosti razvoja modelov strojnega učenja z uporabo grafičnih vmesnikov v orodjih.

Tabela 1. Prednosti in slabosti orodij strojnega učenja z uporabniškim vmesnikom.

Prednosti	Slabosti
Hitrejši in enostavnejši razvoj	Slabša kakovost/točnost modelov
Manjša razvojna ekipa	Počasnejše izvajanje
Enostavnejše vzdrževanje	Težja implementacija za specifične probleme
Manjša potreba po domenskih znanjih	
Vizualizacija procesa	

4. STROJNO UČENJE IZ NIČ

Tretji možen pristop k razvoju inteligentnih informacijskih rešitev z uporabo metod in tehnik strojnega učenja je popolnoma samostojen razvoj, tako rekoč iz nič. Kot velja že za ostala področja razvoja, lahko ugotovimo tudi tukaj – samostojnega razvoja »iz nič« se je smiselno lotiti predvsem takrat, ko želimo imeti popoln nadzor nad uporabljenimi metodami in algoritmi v celotnem življenjskem ciklu razvoja in uporabe tehnik strojnega učenja – t.i. cevovodu strojnega učenja (Slika 4). Ker obstoječe rešitve MLaaS neenakomerno pokrivajo posamezne korake ML cevovoda (Slika 5), je tudi od tega, kateremu izmed korakov želimo posvetiti večjo pozornost, odvisno, ali in v kolikšni meri se bomo lotili samostojnega razvoja.

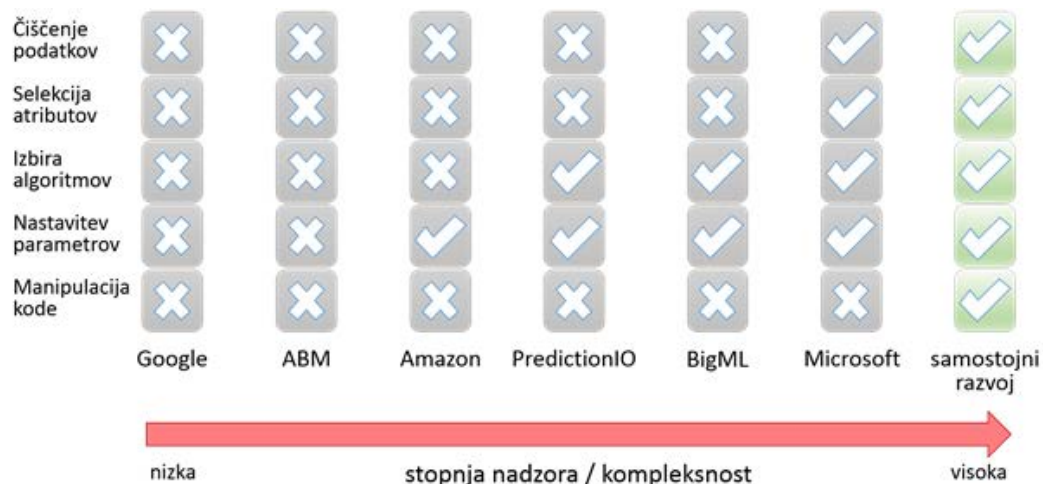


Slika 4. Standardni cevovod strojnega učenja. Od posamezne platforme je odvisno, katere korake je mogoče nadzorovati.

Povsem intuitivno je, da večji nadzor nad vsakim korakom v cevovodu omogoča dobro obveščeni uporabnik, da gradijo bolj kakovostne modele [6]. Funkcija, model in izbira parametrov lahko pomembno vplivajo na uspešnost nalog strojnega učenja (npr. točnost napovedi). Vendar pa je za uspešno optimizacijo vsakega koraka potrebno preseči precejšnjo kompleksnost, kar je težko brez poglobljenega znanja in izkušenj. Po drugi strani pa seveda ni samo po sebi umevno, da lahko specializirane storitve, katerih uporaba seveda precej zmanjša kompleksnost, samodejno opravijo učinkovito upravljanje cevovoda ter nastavitve parametrov algoritmov in metod do te mere, kot je včasih potrebno.

Pri razumevanju odnosov med kompleksnostjo, uspešnostjo in preglednostjo na platformah MLaaS se lahko osredotočimo na tri ključna vprašanja:

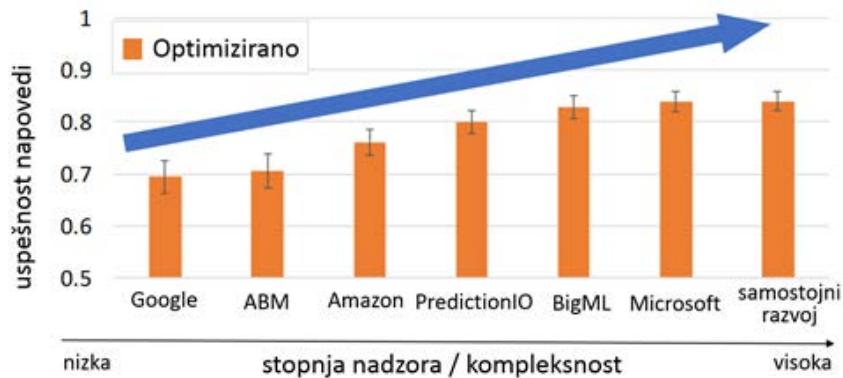
- Kako kompleksnost (stopnja nadzora) sistemov strojnega učenja vpliva na točnost modelov?
- Ali lahko povečan nadzor vodi do večjih tveganj pri oblikovanju slabih modelov?
- Do kakšne mere lahko sistemi MLaaS optimizirajo avtomatizirane dele svojega cevovoda?



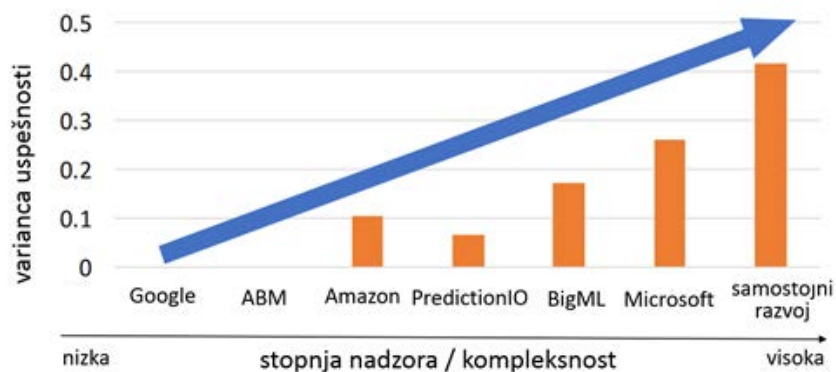
Slika 5. Različne platforme strojnega učenja omogočajo različno stopnjo nadzora nad posameznimi koraki v procesu strojnega učenja; podatki povzeti po [6].

V [6] so avtorji opravili eksperimentalno primerjavo napovedne uspešnosti posameznih platform strojnega učenja nad 119 podatkovnimi množicami. Rezultati so pokazali, da z večanjem kompleksnosti

oz. stopnje nadzora posamezne platforme raste tudi uspešnost najboljših zgrajenih modelov (Slika 6) – tako lahko pri samostojnem razvoju dejansko zgradimo najboljše napovedne modele. A po drugi strani z večanjem kompleksnosti platforme zelo raste tudi tveganje, da bomo dejansko uspeli zgraditi najboljši možni model, ki ga orodje zmore (Slika 7). Preprost sklep bi lahko bil: uporabljajmo tisto platformo oz. orodje, ki smo ga sposobni obvladati.



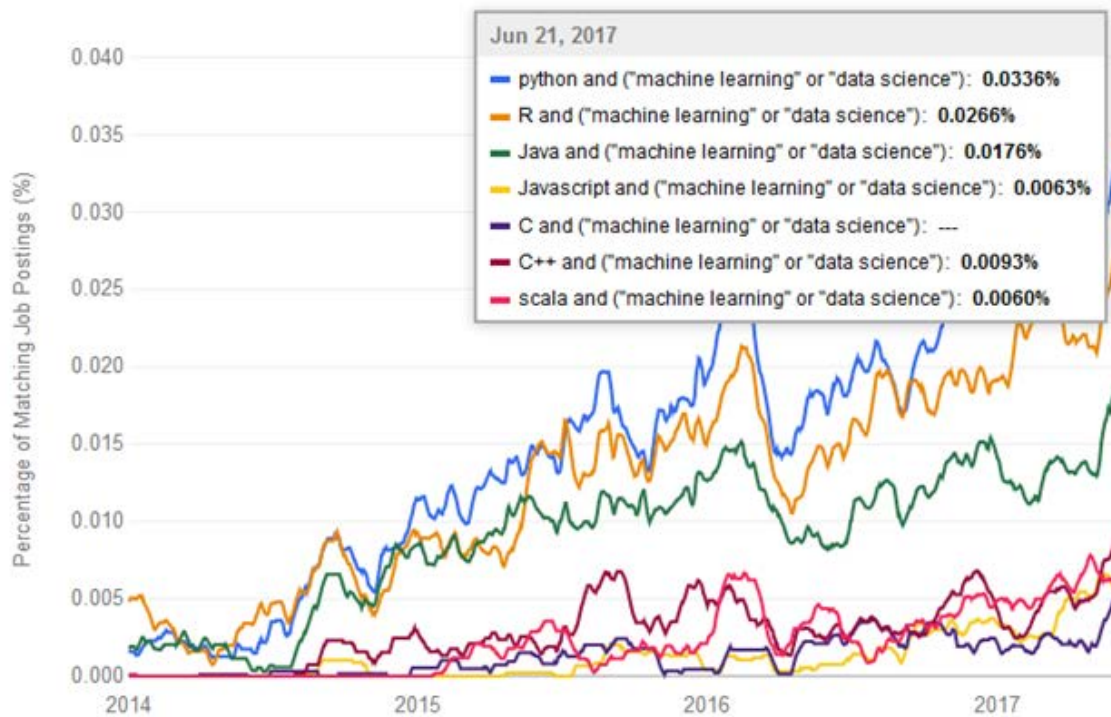
Slika 6. Z večanjem stopnje kompleksnosti se večja uspešnost zgrajenih modelov; povzeto po [6].



Slika 7. Z večanjem stopnje kompleksnosti se zelo večja potencialno tveganje; povzeto po [6].

No, resnici na ljubo »nič«, iz katerega gradimo rešitve strojnega učenja, zajema kopico odličnih programskih knjižnic, večinoma odprtokodnih in podprtih z zelo močno skupnostjo razvijalcev. Tako se potencialnemu razvijalcu ni potrebno ukvarjati z implementacijskimi podrobnostmi samih metod strojnega učenja, hkrati pa njihova programska uporaba omogoča poln dostop do vseh parametrov, ki vplivajo na učinkovitost izvajanja ter uspešnost izvedbe.

Med programskimi jeziki, v katerih se razvijalci najpogosteje lotevajo razvoja, izstopajo predvsem Python, R, Java, C, C++, Scala in morda nekoliko presenetljivo tudi Javascript. Slika 8 prikazuje trend števila oglasov za službe, ki zajemajo strojno učenje oz. podatkovno znanost, in zahtevajo poznavanje določenega programskega jezika. Vidimo lahko, da predvsem Python vse bolj prednjači, medtem ko Java med prvimi tremi jeziki, ki sicer od ostalih že precej odstopajo, nekoliko izgublja zagon.



Slika 8. Zahtevano znanje programskih jezikov v oglasih za službe na področju strojnega učenja in podatkovne znanosti [vir: <https://www.indeed.com/jobtrends/>].

Pri podatkih na Sliki 8 je treba upoštevati dejstvo, da v prikaz niso zajeti le oglasi za razvijalce, pač pa tudi podatkovne analitike ipd. Prav jezik R pa je, spet poleg Pythona, najbolj pogost prav med analitiki, statistiki in drugimi, ki pri svojem delu obdelujejo in analizirajo podatke, ne razvijajo pa programske opreme. Posledično velja, da so za razvoj inteligentnih rešitev najpogosteje uporabljeni jeziki Python, Java, C/C++, R in Javascript. Tako kar 57% razvijalcev že sedaj uporablja Python, še 33% uporabnikov pa meni, da je prav Python ustrezen za razvoj tovrstnih rešitev [7]. R po drugi strani sicer uporablja 31% uporabnikov, ki se ukvarjajo s strojnimi učenjem, vendar bi ga za namen razvoja uporabilo zgolj 5%. Java in C/C++ sta precej izenačena, saj za oba jezika okrog 20% razvijalcev meni, da sta primerna izbira. V Tabeli 2 je zbranih nekaj podatkov o uporabi različnih programskih jezikov za namene razvoja inteligentnih rešitev.

Kot je razvidno iz Tabele 2, so jeziki C/C++, R in Javascript namenjeni precej specifičnim namenom – tistim, pri katerih se tudi sicer posamezni programski jezik najpogosteje uporablja: C/C++ za razvoj iger in vgrajenih sistemov, R za namene statističnih obdelav in analitike, Javascript pa za razvoj spletnih rešitev, predvsem na strani odjemalca.

Tako ostaneta Python in Java tista programska jezika, ki sta najbolj primerna za razvoj raznovrstnih inteligentnih rešitev. Pri tem se, spet podobno kot sicer, Java osredotoča na poslovno uporabo in predvsem večje poslovne sisteme, medtem ko je Python primeren za praktično karkoli in je daleč najbolj priljubljen predvsem pri vseh vrstah zagonskih podjetij in manjših, drznejših projektih.

Tabela 2. Pregled uporabe programskih jezikov v projektih strojnega učenja.

	Python	Java	C/C++	R	Javascript
Najpogostejše aplikacije	procesiranje naravnega jezika, analiza sentimenta, tekstovno in rudarjenje po spletu	upravljanje podpore strankam, omrežna varnost in detekcija vdorov, detekcija goljufij	AI v igrah, upravljanje robotov, omrežna varnost in detekcija vdorov	bioinženiring, bioinformatika, analiza sentimenta, detekcija anomalij	iskalniki, upravljanje podpore strankam, različne specifične naloge
Najmanj pogoste aplikacije	AI v igrah, omrežna varnost in detekcija vdorov	analiza sentimenta, bioinženiring, bioinformatika, specifične naloge	detekcija goljufij, priporočilni sistemi, analiza sentimenta	prepoznavanje govora, AI v igrah, upravljanje robotov	diagnostika v industriji, bioinženiring, bioinformatika
Poklicno ozadje	podatkovni znanstvenik	razvijalec aplikacij za namizne računalnike	inženir za vgrajene sisteme	podatkovni analitik, statistik	spletni razvijalec
Razlogi za uporabo AI	priključiti se valu vpeljave metod strojnega učenja	navodilo vodstva podjetja	dodajanje umetne inteligence v obstoječe aplikacije	podatkovna znanost je (bila) del študija	zagotovitev oz. pridobitev profitabilnih projektov

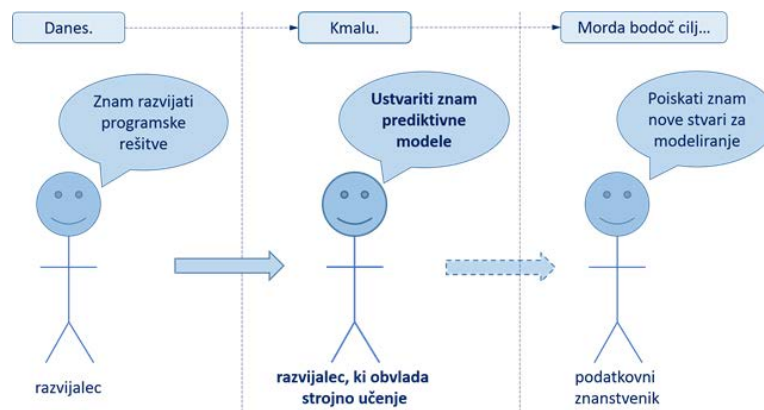
Za konec omenimo še nekatera trenutno najpogostejše uporabljana ogrodja in knjižnice za strojno učenje. Za Java je smiselno pregledati naslednje: ADAMS, Deeplearning4j, ELKI, JavaML, JSAT, Mahout, MALLETT, Massive Online Analysis, RapidMiner in Weka. Za Python pa je nabor še bistveno obširnejši. Med temeljne knjižnice sodijo: NumPy, SciPy in Pandas. Za namene vizualizacije so na voljo: Matplotlib, Seaborn, Bokeh in Plotly. Najpogostejša ogrodja za strojno učenje so: SciKit-Learn, Theano, TensorFlow in Keras. Za obdelavo naravnega jezika imamo na voljo NLTK in Gensim, medtem ko je Scrapy odličen za zajem podatkov iz (nestrukturiranih) vsebin, za statistiko pa velja uporabiti Statsmodels.

5. BISTVENO VPRAŠANJE RAZVOJA INTELIGENTNIH REŠITEV NI VPRAŠANJE IZBIRE TE ALI ONE TEHNOLOGIJE

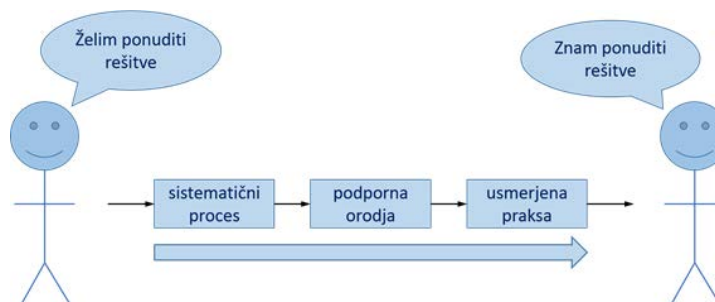
Tipična opravila strojnega učenja, ki se danes vse pogosteje vgrajujejo v sodobne aplikacije, z vidika razumevanja strojnega učenja običajno niso preveč zahtevna. Samodejno identifikacijo neželene e-pošte, razvrščanje izdelkov v priporočilnih sistemih, profiliranje uporabnikov spletne storitve in podobne primere uporabe metod strojnega učenja bi praviloma povprečno usposobljen razvijalec moral obvladati že po osnovni seznanitvi s področjem strojnega učenja. Tudi na videz zahtevne naloge, kot so npr. prepoznavna obrazov, identifikacija objektov na slikah ali ugotavljanje sentimenta v podanem besedilu, ob uporabi sodobnih knjižnic ne bi smele predstavljati pretrdega oreha. Praviloma je v teh primerih potrebno zbrati znane podatke, z njimi naučiti model znanja, ki ga lahko nato uporabimo za predikcijo novih, neznanih situacij. Nekaj klicev ustreznih funkcij in rešitev je na dlani. S tega vidika izbira med specializiranimi rešitvami za strojno učenje (MLaaS), uporabo metod strojnega učenja preko uporabniškega vmesnika ali samostojen razvoj v izbranem programskem jeziku in z izbrano programsko knjižnico ni vprašanje obvladanja strojnega učenja, pač pa drugih, poslovnih in osebnih faktorjev.

Vse nekaj drugega pa je, ko v teoriji delujoč in priporočen recept v praksi ne deluje. Ko rezultati niso niti približno takšni, kot smo pričakovali ali si jih obetali. Takrat ni pravo vprašanje, katera tehnologija je boljša, temveč kako rešiti problem, pred katerim smo se ustavili. Največji problem, ki ga mora razvijalec na svoji poti do uspešnega razvijalca inteligentnih rešitev premagati, tako ni v obvladovanju tehnologij, kompleksnosti knjižnic ali obilju algoritmov in njihovih nastavitev. Največji problem je ustrezno povezati teorijo, algoritme in matematiko strojnega učenja s problemom, ki ga moramo rešiti. Ko priporočen recept, ki običajno dobro deluje, in smo mu dosledno sledili, ne pričara pravega rezultata, se zavemo prave, ogromne vrzeli.

Proces transformacije iz (običajnega) razvijalca v razvijalca, ki obvlada strojno učenje in morda še naprej v podatkovnega znanstvenika, je prikazan na Sliki 9. Za čim lažjo dosegtovrstne transformacije se velja držati top-down pristopa, prikazanega na Sliki 10. Za učinkovito uporabo strojnega učenja se ne smemo ustaviti zgolj pri učnih algoritmih in ogrodjih, ki jih implementirajo in nam ponujajo možnost njihove uporabe. Potrebujemo sistematičen pristop od zgoraj navzdol, kjer se osredotočimo na dejanski rezultat, ki ga želimo: razvijati resnične rešitve »na ključ« z uporabo najboljših, sodobnih orodij in platform. V ta namen pa moramo razumeti, za kaj pri strojnem učenju dejansko gre, ne zgolj znati uporabiti vnaprej pripravljene rešitve.



Slika 9. Transformacija iz razvijalca v razvijalca, ki obvlada strojno učenje.



Slika 10. Pristop, ki se začne z dejanskimi problemi strojnega učenja in konča z rešitvijo »na ključ«.

6. ZAKLJUČEK

V prispevku smo pregledali tri nivoje storitev, kjer vsak nivo služi svojemu primeru uporabe. Če se navežemo na tri tipe uporabnikov takih storitev, vidimo da so razvijalske ekipe s strokovno usposobljenim kadrom na področju strojnega učenja in izdelkom, katerega osrednji del je inteligentna obdelava podatkov, primarni uporabniki platform na najnižjem nivoju. Te namreč ponujajo tem uporabnikom največjo svobodo pri implementaciji, a poleg strokovno usposobljenega kadra zahtevajo mnogo več časa za implementacijo ter višje stroške vzdrževanja in skaliranja teh storitev. Tisti, ki s pomočjo metod podatkovne znanosti in rezultatov teh metod sprejemajo poslovne odločitve, so primarni uporabniki storitev na drugem nivoju, kjer platforme z vizualnim uporabniškim vmesnikom omogočajo rešitve strojnega učenja. Te namreč omogočajo dovolj velik spekter prilagodljivosti, uporabi se jih lahko v oblaku ali lokalno, in ne zahtevajo programerskega znanja. Razvijalci programske opreme, ki ne temeljijo na strojnem učenju, raziskovalni novinarji in raziskovalci pa so primarni uporabniki specializiranih in že optimiziranih storitev, ki jih kot storitve ponujajo različni ponudniki.

Uporaba tehnologije podatkov in strojnega učenja je glavna prioriteta ali celo izdelek mnogih organizacij. Na voljo so vedno bolj izpopolnjeni analitični postopki in storitve strojnega učenja, ki jim omogočajo, da se ogromne količine podatkov, ki so na voljo, v celoti izkoristijo za namen optimizacije poslovnega procesa. Zelo enostavno se je izgubiti v različnih razpoložljivih rešitvah. Te se razlikujejo po ponujenih algoritmih, primerih uporabe in zahtevanem poznavanju strojnega učenja. Ustvarjanje mostu med podatkovno znanostjo in poslovno vrednostjo je težaven proces, predvsem če primanjkuje znanja o podatkovni znanosti ali domenskih izkušnj. Spodnji diagram prikazuje napoved Gartner-ja o vodilnih platformah in vizionarjih na področju platform strojnega učenja. Gartner v naslednjih dveh do treh letih pričakuje nadaljevanje turbulence na trgu podatkovne znanosti in platform strojnega učenja. V njem bodo še naprej v celoti novi ponudniki, pa tudi obstoječi ponudniki sosednjih trgov (kot so analitika in BI). [8]

Prihodki iz računalniških znanosti in platform za strojno učenje so se v letu 2016 povečali za 9,3%, na 2,4 milijarde USD. Ta rast je več kot dvakrat večja od celotnega analitičnega trga (4,5%), 2,4 milijarde pa predstavlja 14,1% celotnega svetovnega analitičnega in BI prihodka (v letu 2015 je bil ta delež 13,5%). Rast trga podatkovne znanosti in strojnega učenja spodbuja željo končnih uporabnikov, da uporabijo bolj napredne analitike za izboljšanje odločanja v celotnem podjetju. [7]

Po vseh prognozah rasti trga podatkovne znanosti je tako pričakovano, da bo število različnih ponujenih platform, na vseh treh nivojih, le še raslo. To bo vsekakor imelo pozitiven vpliv na kvaliteto storitev, saj bodo ponudniki takih platform in storitev tekmovali za enak ali podoben profil uporabnikov in razvijalcev, kar pa je za končnega uporabnika pozitivno v smislu višje kvalitete storitev in nižjih stroškov uporabe teh storitev.



Slika 11. Gartner-jev kvadrant za leto 2017 za podjetja, ki razvijajo platforme strojnega učenja [8].

7. LITERATURA

- [1] <https://www.gartner.com/document/3868668>, Market Guide: Machine Learning Infrastructure as a Service, obiskano 17.5.2018.
- [2] <https://blog.g2crowd.com/blog/trends/artificial-intelligence/2018-ai/machine-learning-service-mlaas/>, AI Trends 2018: Machine Learning as a Service (MLaaS), obiskano 17.5.2018.

- [3] <https://www.datamation.com/cloud-computing/artificial-intelligence-as-a-service-ai-meets-the-cloud.html>, Artificial Intelligence as a Service: AI Meets the Cloud - Datamation, obiskano 17.5.2018.
- [4] <https://www.altexsoft.com/blog/datascience/comparing-machine-learning-as-a-service-amazon-microsoft-azure-google-cloud-ai/>, Comparing MLaaS: Amazon AWS, MS Azure, Google Cloud AI, obiskano 17.5.2018.
- [5] <https://www.datamation.com/cloud-computing/cloud-machine-learning-is-it-right-for-you.html>, Cloud Machine Learning: Is It Right for You? - Datamation, obiskano 17.5.2018.
- [6] Yao Y, Xiao Z, Wang B, Viswanath B, Zheng H, Zhao BY. Complexity vs. performance: empirical analysis of machine learning as a service. Proceedings of the 2017 Internet Measurement Conference 2017, ACM, pp. 384-397.
- [7] <https://towardsdatascience.com/what-is-the-best-programming-language-for-machine-learning-a745c156d6b7>, What is the best programming language for Machine Learning?, Developer Economics, 2017, obiskano 17.5.2018.
- [8] <https://www.gartner.com/document/3860063>, Magic Quadrant for Data Science and Machine-Learning Platforms, obiskano 17.5.2018.
- [9] <https://www.gartner.com/document/3772081>, Hype Cycle for Data Science and Machine Learning, 2017, obiskano 17.5.2018.
- [10] <https://www.blue-granite.com/blog/bid/404378/azure-machine-learning-an-overview>, Azure Machine Learning: An Overview, obiskano 16.5.2018.