Mašinsko učenje

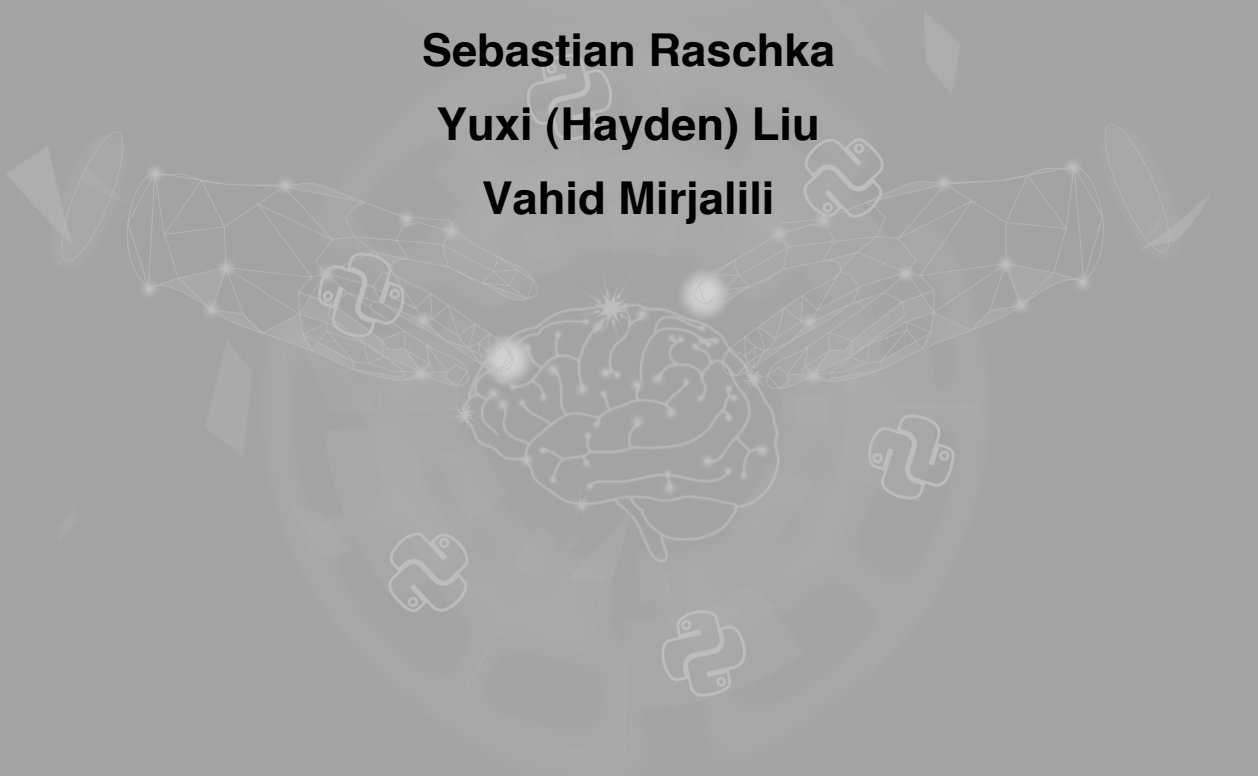
PyTorch i Scikit-Learn

Razvoj modela mašinskog učenja i dubokog učenja pomoću programskog jezika Python

Sebastian Raschka

Yuxi (Hayden) Liu

Vahid Mirjalili



kompjuter biblioteka

Packt›

Izdavač:

kompjuter biblioteka

Obalskih radnika 4a, Beograd

Tel: 011/2520272

e-mail: kombib@gmail.com

internet: www.kombib.rs

Urednik: Mihailo J. Šolajić

Za izdavača, direktor:
Mihailo J. Šolajić

Autor: Sebastian Raschka
Vahid Mirjalili
Yuxi (Hayden) Liu

Prevod: Slavica Prudkov

Lektura: Nemanja Lukić

Slog: Zvonko Aleksić

Znak Kompjuter biblioteke:
Miloš Milosavljević

Štampa: „Pekograf“, Zemun

Tiraž: 500

Godina izdanja: 2022.

Broj knjige: 554

Izdanje: Prvo

ISBN: 978-86-7310-577-2

Machine Learning with PyTorch and Scikit-Learn

Sebastian Raschka
Yuxi (Hayden) Liu
Vahid Mirjalili

ISBN 978-1-80181-931-2

Predgovor

Poslednjih godina, metodi mašinskog učenja sa njihovom sposobnošću da daju smisao ogromnim količinama podataka i da automatizuju odluke, našli su široku primenu u zdravstvu, robotici, biologiji, fizici, potrošačkim proizvodima, internet uslugama i raznim drugim industrijama.

Ogromni skokovi u nauci obično dolaze iz kombinacije moćnih ideja i sjajnih alata. Mašinsko učenje nije izuzetak. Uspeh metoda učenja zasnovanih na podacima zasniva se na genijalnim idejama hiljada talentovanih istraživača ove oblasti tokom proteklih 60 godina. Ali njihova nedavna popularnost je, takođe, podstaknuta evolucijom hardverskih i softverskih rešenja koja ih čine skalabilnim i dostupnim. Ekosistem odličnih biblioteka za numeričko računanje, analizu podataka i mašinsko učenje izgrađenih oko programskog jezika Python, kao što su NumPy i scikit-learn, stekao je široku primenu u istraživanju i industriji.To je u velikoj meri pomoglo da Python postane najpopularniji programski jezik.

Ogromna poboljšanja u zadacima kompjuterskog vida, teksta, govora i drugih zadataka koje je uvela nedavna pojava tehnika dubokog učenja predstavljaju primer ove teme. Pristupi se oslanjaju na teoriju neuronskih mreža od poslednje četiri decenije, koje su počele da rade izuzetno dobro u kombinaciji sa GPU-ovima i visokooptimizovanim računarskim rutinama.

Naš cilj u izgradnji biblioteke PyTorch u proteklih pet godina bio je da istraživačima pružimo najfleksibilniji alat za izražavanje algoritama dubokog učenja koji vodi računa o osnovnim inženjerskim složenostima. Imali smo koristi od odličnog Python ekosistema. Zauzvrat, imali smo sreću da vidimo zajednicu veoma talentovanih ljudi koji grade napredne modele dubokog učenja u različitim domenima povrh biblioteke PyTorch. Među njima su bili i autori ove knjige.

Poznajem Sebastijana iz ove bliske zajednice već nekoliko godina. Ima talenat bez premca u lakom objašnjavanju informacija i činjenju kompleksa dostupnim. Sebastijan je doprineo u mnogim široko korišćenim softverskim paketima za mašinsko učenje i autor je desetina odličnih tutorijala o dubokom učenju i vizuelizaciji podataka.

Za primenu mašinskog učenja u praksi potrebno je i vladanje idejama i alatima. Početak može da deluje zastrašujuće, od sticanja smisla o teorijskim konceptima do otkrivanja koje softverske pakete je potrebno da instalirate.

Na sreću, knjiga koju držite u rukama odlično kombinuje koncepte mašinskog učenja i praktične inženjerske korake koji će vas voditi na ovom putovanju. Očekuje vas divna tura od osnova tehnika vođenih podacima do najnovih arhitektura dubokog učenja. U okviru svakog poglavlja naći ćete konkretne primere koda koji primenjuju predstavljene metode u praktičnim zadacima.

Kada je objavljeno prvo izdanje 2015. godine, knjiga je postavila veoma visoka očekivanja za kategoriju knjiga o mašinskom učenju i programskom jeziku Python. Ali izvrsnost se tu nije zaustavila. Sa svakim izdanjem, Sebastijan i tim su nadograđivali i usavršavali materijal kako se evolucija dubokog učenja odvijala u novim domenima. U ovom novom izdanju biblioteke PyTorch naći ćete nova poglavlja o arhitekturi transformatora i grafovskim neuronskim mrežama. Ovi pristupi su najsavremeniji u dubokom učenju i velikom brzinom su preuzeli oblast razumevanja teksta i molekularne strukture u poslednje dve godine. Moći ćete da ih vežbate korišćenjem novih, ali veoma popularnih softverskih paketa u ekosistemu, kao što su Hugging Face, PyTorch Lightning i PyTorch Geometric.

Odličan balans teorije i prakse u ovoj knjizi nije iznenađenje s obzirom na kombinaciju napredne istraživačke ekspertize autora i iskustva u praktičnom rešavanju problema. Sebastijan Raška i Vahid Mirjalili to crpe iz iskustva u istraživanju dubokog učenja za računarski vid i računarsku biologiju. Hejden Liu unosi iskustvo primene metoda mašinskog učenja za predviđanje događaja, sisteme preporuka i druge zadatke u industriji. Svi autori dele duboku strast prema obrazovanju, a to se ogleda u pristupačnom načinu na koji se knjiga kreće od jednostavnih do naprednih tema.

Uveren sam da će vam ova knjiga biti od neprocenjive vrednosti, kao opširan pregled uzbudljive oblasti mašinskog učenja i kao riznica praktičnih uvida. Nadam se da će vas inspirisati da primenite mašinsko učenje za opšte dobro u bilo kom problematičnom području.

Dmitro Dzhulgakov

PyTorch Core Maintaine

O autorima

Dr Sebastian Raschka je asistent profesora statistike na Univerzitetu Viskonsin u Medisonu, fokusiran na mašinsko učenje i duboko učenje. Fokus njegovog istraživanja su opšti izazovi, kao što je few-shot učenje za upotrebu ograničenih podataka i razvoj dubokih neuronskih mreža za redne ciljeve. Sebastian je takođe strastven saradnik u zajednici otvorenog koda, a u ulozi vodećeg predavača o veštačkoj inteligenciji na Grid.ai, planira da sledi svoju strast da pomaže ljudima koji žele da započnu karijeru u oblasti mašinskog učenja i veštačke inteligencije.

Zahvaljujem Jitianu Zhaou i Benu Kaufmanu sa kojima sam imao zadovoljstvo da radim na novim poglavljima o transformatorima i grafovskim neuronskim mrežama. Takođe sam veoma zahvalan na Hajdenovoj i Vahidovoj pomoći - ova knjiga ne bi bila moguća bez vas. Konačno, zahvaljujem se Andrea Panizzai, Toni Giteru i Adamu Bielskom za korisne diskusije o odeljcima rukopisa.

Yuxi (Hayden) Liu je softverski inženjer mašinskog učenja u Google-u i radio je kao istraživač mašinskog učenja u različitim domenima vođenim podacima. Hajden je autor serije knjiga o mašinskom učenju. Njegova prva knjiga, *Python Machine Learning By Example*, bila je rangirana kao najprodavanija u svojoj kategoriji na veb sajtu Amazona 2017. i 2018. godine, a prevedena je na mnogo jezika. Njegove ostale knjige uključuju *R Deep Learning Project, Hands-On Deep Learning Architectures with Python* i *PyTorch 1.x Reinforcement Learning Cookbook*.

Želeo bih da zahvalim svim dobrim ljudima sa kojima sam radio, posebno ko-autorima, urednicima u Packt-u i recenzentima. Bez njih ova knjiga bi bila mnogo teža za čitanje i za primenu na probleme iz stvarnog sveta. Konačno, želeo bih da zahvalim svim čitaocima na podršci koja me je podstakla da napišem PyTorch izdanje ove najprodavanije knjige o mašinskom učenju.

Dr Vahid Mirjalili je istraživač dubokog učenja fokusiran na aplikacije računarskog vida. Vahid je doktorirao mašinsko inženjerstvo i računarske nauke na Michigan State univerzitetu. Tokom svojih doktorskih studija razvio je nove algoritme za računarski vid za rešavanje problema iz stvarnog sveta i publikovao je nekoliko istraživačkih članaka koji su veoma cenjeni u zajednici računarskog vida.

Ostali saradnici

Benjamin Kaufman je doktorski kandidat na Univerzitetu Viskonsin Medison u oblasti nauke o biometrijskim podacima. Njegovo istraživanje je fokusirano na razvoj i primenu metoda mašinskog učenja za otkrivanje leka. Njegov rad u ovoj oblasti pružio je bolje razumevanje grafovskih neuronskih mreža.

Jitian Zhao je student doktorskih studija na Univerzitetu Viskonsin Medison, gde je razvila svoje interesovanje za modele jezika velikih razmera. Veoma je zainteresovana za duboko učenje u razvoju aplikacija iz stvarnog sveta i teorijskoj podršci.

Želim da zahvalim roditeljima na podršci. Oni su me podstakli da uvek pratim svoje snove i motivisali me da budem dobra osoba.

O recenzentu

Roman Tezikov je inženjer industrijskog istraživanja i entuzijasta dubokog učenja sa preko četiri godine iskustva u naprednom računarskom vidu, NLP-u i MLOps-u. Kao suosnivač ML-REPA zajednice organizovao je nekoliko radionica i sastanaka o reproduktivnosti mašinskog učenja i automatizaciji pipeline-a. Jedan od njegovih aktuelnih poslovnih izazova je primena računarskog vida u modnoj industriji. Roman je takođe bio glavni programer za Catalyst - PyTorch biblioteke za ubrzano duboko učenje.

Kratak sadržaj

Sadržaj

POGLAVLJE 8

POGLAVLJE 9

POGLAVLJE 10

POGLAVLJE 11

POGLAVLJE 12

POGLAVLJE 15

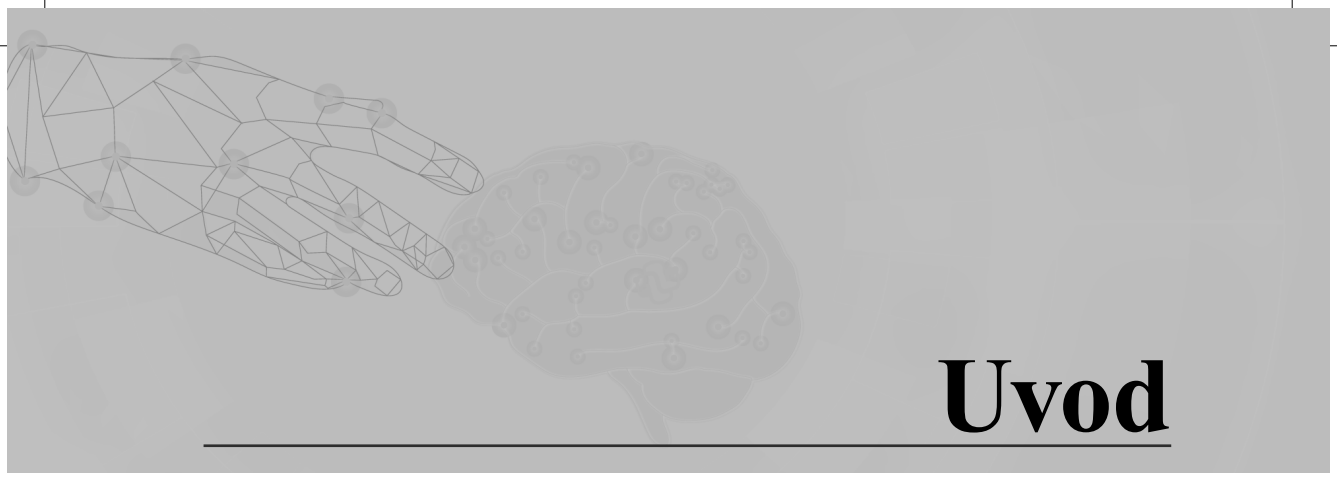
Modelovanje sekvencijalnih podataka upotrebom rekurentnih neuronskih mreža 499

POGLAVLJE 16

Transformatori – poboljšanje obrade prirodnog jezika pomoću mehanizama pažnje 539

POGLAVLJE 18

POGLAVLJE 19



Uvod

Verovatno vam je poznata činjenica, iz vesti i društvenih medija, da je mašinsko učenje postalo jedna od najuzbudljivijih tehnologija našeg vremena. Velike kompanije, kao što su „Microsoft“, „Google“, „Meta“, „Apple“, „Amazon“, IBM i mnoge druge, sasvim razumljivo, investiraju u istraživanje mašinskog učenja i u njegovu primenu. Iako se možda čini da je mašinsko učenje postala vrlo popularna reč našeg vremena, to svakako nije preterivanje. Ova uzbudljiva oblast otvara put novim mogućnostima i postala je neophodna u svakodnevnom životu. Razmislite samo o razgovoru sa glasovnim pomoćnikom na pametnom telefonu, preporučivanju odgovarajućih proizvoda za kupce, sprečavanju prevare kreditnim karticama, filtriranju spem poruka iz elektronskog poštanskog sandučeta i detektovanju i dijagnostikovanju bolesti i tako dalje.

Ako želite da postanete praktikant mašinskog učenja ili da bolje rešavate probleme koji se javljaju, ili možda razmišljate o karijeri u istraživanju mašinskog učenja, onda je ova knjiga za vas! Međutim, za nove korisnike teorijski koncepti mašinskog učenja mogu da budu previše teški, ali izdato je mnogo knjiga poslednjih godina koje će vam pomoći da započnete da koristite mašinsko učenje implementiranjem moćnih algoritama učenja.

Pregledanje praktičnih primera koda i izrada primera primene mašinskog učenja odlični su načini da započnete učenje ove oblasti. Konkretni primeri pomažu pri ilustrovanju širih koncepata postavljanjem naučenog materijala direktno u praksu. Međutim, ne zaboravite da sa velikom moći dolazi i velika odgovornost! U ovoj knjizi, osim što ćemo vam pomoći da steknete praktično iskustvo u mašinskom učenju upotrebom Python programskog jezika i biblioteka mašinskog učenja zasnovanih na Pythonu, predstavićemo i matematičke koncepte u algoritmima mašinskog učenja, koji su važni za uspešnu upotrebu mašinskog učenja. Prema tome, ova knjiga se razlikuje od čisto praktične knjige; u njoj su opisani potrebni detalji u sa vezi konceptima mašinskog učenja i obezbeđena su intuitivna i informativna objašnjenja kako funkcionišu algoritmi mašinskog učenja, kako da ih upotrebite i, najvažnije, kako da izbegnete najčešće zamke.

U ovoj knjizi ćemo krenuti na uzbudljivo putovanje i opisaćemo sve važne teme i koncepte da bismo vam pomogli da započnete rad u ovoj oblasti. Ako otkrijete da vaša glad za znanjem nije zadovoljena, možete da upotrebite mnoge korisne resurse koji su referencirani u ovoj knjizi da biste pratili najnovija dostignuća u ovoj oblasti.

Za koga je ova knjiga

Ova knjiga je idealna pomoć za učenje kako da primenite mašinsko učenje i duboko učenje na širok raspon zadataka i skupova podataka. Ako ste programer koji želi da bude u toku sa najnovijim trendovima u tehnologiji, ova knjiga je definitivno za vas. Takođe, ako ste student i razmišljate o promeni karijere, ova knjiga će vam biti uvod i sveobuhvatan vodič u svet mašinskog učenja.

Šta obuhvata ova knjiga

U poglavlju 1, „*Kako da računarima pružite mogućnost da uče iz podataka*“, predstavićemo glavne podoblasti mašinskog učenja koje se koriste za rešavanje različitih problema. Osim toga, opisaćemo osnovne korake za kreiranje tipične faze obrade (pipeline) izgradnje modela mašinskog učenja, koji će nas pratiti u narednim poglavljima.

U poglavlju 2, „*Obučavanje jednostavnih algoritama mašinskog učenja za klasifikaciju*“, vraćamo se na početke mašinskog učenja i predstavljamo binarne klasifikatore perceptrona i adaptivne linearne neurone. Predstavićemo osnove klasifikacije obrazaca i fokusiraćemo se na interakciju algoritama optimizacije i mašinskog učenja.

U poglavlju 3, „Predstavljanje klasifikatora mašinskog učenja pomoću biblioteke scikit-learn“, opisaćemo važne algoritme mašinskog učenja za klasifikaciju i obezbedićemo praktične primere upotrebom biblioteke scikit-learn, jedne od najpopularnih i sveobuhvatnijih biblioteka mašinskog učenja otvorenog koda.

U poglavlju 4, „*Izgradnja dobrih skupova podataka za obuku - pretprocesiranje podataka*“, opisaćemo kako se rešavaju najčešći problemi u neobrađenim skupovima podataka, kao što je podatak koji nedostaje. Takođe ćemo predstaviti nekoliko pristupa za identifikaciju najinformativnijih atributa u skupovima podataka i način kako se pripremaju promenljive različitih tipova kao pravilni unosi za algoritme mašinskog učenja.

U poglavlju 5, „*Kompresovanje podataka upotrebom redukcije dimenzionalnosti*“, upoznaćete osnovne tehnike za redukciju broja atributa u skupovima podataka na manje skupove, uz zadržavanje većine njihovih korisnih i diskriminitornih informacija. Osim toga, opisaćemo standardan pristup redukciji dimenzionalnosti analizom glavnih komponenata i njihovim poređenjem sa nadgledanim tehnikama nelinearne transformacije.

U poglavlju 6, „*Učenje najbolje prakse za procenu modela i podešavanje hiperparametara*“, saznaćete šta treba, a šta ne treba da radite pri proceni performansi prediktivnih modela. Upoznaćete i različite metrike za merenje performansi modela i tehnika za fino podešavanje algoritama mašinskog učenja.

U poglavlju 7, „*Kombinovanje različitih modela za učenje u ansamblu*“, predstavićemo različite koncepte efikasnog kombinovanja većeg broja algoritama učenja. Istražićemo kako se grade ansambli stručnjaka za prevazilaženje slabosti pojedinačnih učenika, što dovodi do tačnijih i pouzdanijih predviđanja.

U poglavlju 8, „*Primena mašinskog učenja na analizu mišljenja*“, opisaćemo osnovne korake za transformisanje tekstualnih podataka u smislene reprezentacije za algoritme mašinskog učenja za predviđanje mišljenja ljudi na osnovu njihovog pisanja.

U poglavlju 9, „*Predviđanje kontinualnih ciljnih promenljivih pomoću regresione analize* “, opisaćemo osnovne tehnike za modelovanje linearnog odnosa između cilja i promenljivih odgovora za izvršavanje kontinualnog predviđanja. Nakon predstavljanja različitih linearnih modela, biće reči o polinomnoj regresiji i pristupima zasnovanim na stablu.

U poglavlju 10, „*Upotreba neoznačenih podataka - analiza klasterovanja*“, fokus prebacujemo na različite podoblasti mašinskog učenja, odnosno na nenadgledano učenje. Opisaćemo algoritme iz tri osnovne familije algoritama klasterovanja za pronalaženje grupe objekata koji dele određeni stepen sličnosti.

U poglavlju 11, „*Implementiranje višeslojnih veštačkih neuronskih mreža ‚od nule'*“, proširićemo koncept optimizacije zasnovane na gradijentu, koju smo predstavili u poglavlju 2 „Obučavanje jednostavnih algoritama mašinskog učenja za klasifikaciju“ da bismo izgradili moćne višeslojne neuronske mreže na osnovu popularnog algoritma povratne propagacije u programskom jeziku Python.

Poglavlje 12, „*Paralelizacija obuke neuronske mreže pomoću biblioteke PyTorch*“, nadovezuje se na znanje stečeno u prethodnom poglavlju za obezbeđivanje efikasnijeg praktičnog vodiča za obuku neuronskih mreža. Fokus u ovom poglavlju je na biblioteci PyTorch, Python biblioteci otvorenog koda koja omogućava da iskoristimo više jezgara modernih procesora (GPU) i konstruišemo duboke neuronske mreže iz zajedničkih gradivnih blokova pomoću jednostavnog i fleksibilnog API-ja.

U poglavlju 13, „*Detaljnije - mehanika biblioteke PyTorch*“, nastavićemo razmatranje teme iz prethodnog poglavlja i predstavićemo naprednije koncepte i funkcionalnosti biblioteke PyTorch. PyTorch je izuzetno velika i sofisticirana biblioteka i u ovom poglavlju ćemo vas provesti kroz koncepte, kao što su dinamičko izračunavanje grafova i automatska diferencijacija. Takođe ćete učiti kako da koristite objektno-orijentisan API biblioteke PyTorch za implementiranje kompleksnih neuronskih mreža i kako biblioteka PyTorch Lightning pomaže u najboljoj praksi i minimizira šablonski kod.

U poglavlju 14, „*Klasifikovanje slika pomoću dubokih konvolucionih neuronskih mreža*“, predstavićemo **konvolucione neuronske mreže (CNN)**. CNN predstavlja određeni tip arhitekture duboke neuronske mreže koja je posebno dobro prilagođena skupovima podataka slika. Zbog svoje superiorne performanse u odnosu na tradicionalne pristupe, CNN se sada koristi u računarskom vidu za postizanje vrhunskih rezultata za različite zadatke prepoznavanja slika. U ovom poglavlju ćete naučiti kako konvolucioni slojevi mogu da se upotrebe kao moćni ekstraktori atributa za klasifikaciju slika.

U poglavlju 15, „Modelovanje sekvencijalnih podataka upotrebom rekurentnih neuronskih mreža“, upoznaćete još jednu popularnu arhitekturu neuronske mreže za duboko učenje, koja je posebno dobro prilagođena za upotrebu teksta i drugih tipova sekvencijalnih podataka i podataka vremenskih serija. Kao vežbu zagrevanja, u ovom poglavlju predstavićemo rekurentne neuronske mreže za predviđanje mišljenja za recenzije filmova. Zatim ćemo opisati učenje rekurentnih mreža da prebacuju informacije iz knjiga da bi generisale potpuno novi tekst.

U poglavlju 16, „*Transformatori - poboljšanje obrade govornog jezika pomoću mehanizama pažnje*“, fokusiramo se na najnovije trendove obrade govornog jezika i objasnićemo kako mehanizmi pažnje pomažu u modelovanju složenih odnosa u dugim sekvencama. Konkretno, u ovom poglavlju opisujemo arhitekturu uticajnog transformatora i najsavremenije modele transformatora, kao što su BERT i GPT.

U poglavlju 17, „*Generativne suparničke mreže za sintetizovanje novih podataka*“, predstavićemo popularni suparnički režim obuke za neuronske mreže koji može da se upotrebi za generisanje novih slika realističnog izgleda. Poglavlje ćemo započeti kratkim uvodom u autoenkodere koji su poseban tip arhitekture neuronske mreže koji može da se upotrebi za kompresovanje podataka. Zatim ćemo prikazati kako se kombinuje deo dekodera autoenkodera sa drugom neuronskom mrežom, koja može da razlikuje stvarne i sintetizovane slike. Omogućavanjem nadmetanja dve neuronske mreže u pristupu suparničke obuke implementiraćemo generativnu suparničku mrežu koja generiše nove ručno pisane cifre.

U poglavlju 18, „*Grafovske neuronske mreže za otkrivanje zavisnosti u grafički strukturiranim podacima*“, predstavićemo nešto više osim tabelarnih skupova podataka, slika i teksta. U ovom poglavlju ćemo predstaviti grafovske neuronske mreže koje obrađuju grafički strukturirane podatke, kao što su mreže društvenih medija i molekuli. Nakon objašnjenja osnova konvolucije grafova, pronaći ćete instrukcije koje vam pokazuju kako da implementirate prediktivne modele za molekularne podatke.

U poglavlju 19, „*Učenje uslovljavanjem za donošenje odluka u kompleksnim okruženjima*“, obuhvatićemo potkategoriju mašinskog učenja koja se često koristi za obučavanje robota i drugih autonomnih sistema. Prvo ćemo predstaviti osnove **učenja uslovljavanjem (RL)** da biste upoznali interakciju agenta/okruženja, procesom nagrađivanja RL sistema i konceptom učenja iz iskustva. Nakon što naučite osnovne kategorije učenja uslovljavanjem, implementiraćete i obučiti agenta koji može da se kreće kroz mrežu okruženja upotrebom Q-learning algoritma. Na kraju ćemo predstaviti deep Q-learning algoritam koji je varijanta Q-learning algoritma koji koristi duboke neuronske mreže.

Da biste dobili maksimum iz ove knjige

Idealno bi bilo da ste već upoznati sa programiranjem u programskom jeziku Python da biste mogli da pratite primere koda koje obezbeđujemo za ilustraciju i primenu različitih algoritama i modela. Da biste dobili maksimum iz ove knjige takođe je korisno da budete upoznati sa matematičkom notacijom.

Običan laptop ili desktop računar bi trebalo da bude dovoljan za pokretanje većine koda iz ove knjige, a u prvom poglavlju obezbeđujemo instrukcije za Python okruženje. U narednim poglavljima ćemo predstaviti dodatne biblioteke i preporuke za instalaciju, kada to bude potrebno.

Novija grafička procesna jedinica (GPU) može da ubrza izvršenje koda u poglavljima o dubokom učenju. Međutim, GPU nije obavezan, a takođe obezbeđujemo instrukcije za upotrebu besplatnih cloud resursa.

Preuzimanje primera koda

Svi primeri koda su dostupni za preuzimanje sa GitHub-a, na adresi https://github.com/rasbt/machine-learning-book. Takođe, imamo i druge pakete koda iz našeg bogatog kataloga knjiga i video snimaka dostupne na adresi https://github.com/PacktPublishing/. Pogledajte!

Iako preporučujemo upotrebu Jupyter Notebook-a za interaktivno izvršenje koda, svi primeri koda su dostupni u formatima Python skripta (na primer, ch02/ch02.py) i Jupyter Notebooka (na primer, ch02/ch02.ipynb). Štaviše, preporučujemo da pregledate README.md fajl koji objedinjuje dodatne informacije i ažuriranja za svako pojedinačno poglavlje.

Preuzimanje kolornih slika

Takođe smo obezbedili PDF fajl koji sadrži kolorne slike ekrana/dijagrama koji su upotrebljeni u knjizi. Možete da preuzmete ovaj fajl na adresi:

https://static.packt-cdn.com/downloads/9781801819312_ColorImages.pdf.

Pored toga, kolorne slike niže rezolucije su ugrađene u code notebook za ovu knjigu, koji se nalazi u fajlovima primera koda.

Konvencije

U ovoj knjizi pronaći ćete više različitih stilova za tekst.

Evo nekih primera tih stilova i objašnjenja njihovog značenja. Reči koda u tekstu su prikazane na sledeći način: „A već instalirani paketi mogu da budu ažurirani pomoću oznake --upgrade“.

Blok koda je postavljen na sledeći način:

```
def __init__(self, eta=0.01, n_iter=50, random_state=1):
    self.eta = eta
    self.n_iter = n_iter
    self.random_state = random_state
```

Svi unosi u Python interpreteru su ispisani na sledeći način (vidite simbol >>>). Očekivan ispis će biti prikazan bez simbola >>>:

```
>>> v1 = np.array([1, 2, 3])
>>> v2 = 0.5 * v1
>>> np.arccos(v1.dot(v2) / (np.linalg.norm(v1) *
...           np.linalg.norm(v2)))
0.0
```

Svi unosi ili ispisi komandne linije su napisani na sledeći način:

```
pip install gym==0.20
```

Novi termini i **važne reči** su napisani podebljanim slovima. Reči koje vidite na ekranu - na primer, u menijima ili okvirima za dijalog, prikazane su u tekstu na sledeći način: „Kliknite na dugme **Next** da biste se prebacili na sledeći ekran“.

Upozorenja ili važne **napomene** se prikazuju u ovakvom okviru.

Saveti i trikovi se prikazuju ovako.

Stupite u kontakt

Povratne informacije naših čitalaca su uvek dobrodošle.

Opšte povratne informacije: Ako imate pitanja o bilo kom aspektu ove knjige, pošaljite nam email na adresu kombib@gmail.com.

Štamparske greške: Iako smo preduzeli sve mere da bismo obezbedili tačnost sadržaja, greške su moguće. Ako pronađete grešku u nekoj od naših knjiga - u tekstu ili u kodu, bili bismo zahvalni ako biste nam to javili. Na taj način možete da pomognete drugim čitaocima da izbegnu frustracije i pomognete nama da poboljšamo sledeće verzije ove knjige. Ako pronađete grešku, molimo vas da nas o tome obavestite na email kombib@gmail.com.

Piraterija: Ako pronađete ilegalnu kopiju naših knjiga, u bilo kojoj formi na Internetu, molimo vas da nas o tome obavestite i da nam pošaljete adresu lokacije ili naziv veb sajta. Pošaljite nam poruku na adresu kombib@gmail.com i pošaljite nam link ka sumnjivom materijalu.

Pregledi

Kada pročitate ovu knjigu Mašinsko učenje pomoću biblioteka PyTorch i scikit-learn, voleli bismo da čujemo vaše mišljenje! Molim vas kliknite ovde da biste otvorili stranicu recenzije na Amazon veb sajtu za ovu knjigu i napišite svoje mišljenje.

Vaše mišljenje je važno za nas i tehničku zajednicu i pomoći će nam da budemo sigurni da isporučujemo sadržaj odličnog kvaliteta.

Predlozi za prevod

Oni koji kupuju naša izdanja su nam, prethodnih godina, veoma pomagali da izaberemo knjigu za prevod na srpski jezik.

To možete da uradite i vi. Posetite stranu predloga za prevod:

http://bit.ly/2NVhHRg

i ukoliko je među knjigama koje smo ponudili i ona koja je vama potrebna, napišite komentar. Svaki komentar ćemo nagraditi.

A ukoliko među knjigama koje smo ponudili nema one koja je vama potrebna, pošaljite nam mail sa vašim predlogom na kombib@gmail.com. Ukratko objasnite zašto bi baš ta knjiga bila zanimljiva, a ukoliko je budemo objavili vi ćete dobiti knjigu na poklon.

Postanite član Kompjuter biblioteke

Kupovinom jedne naše knjige stekli ste pravo da postanete član Kompjuter biblioteke. Kao član možete da kupujete knjige u pretplati sa 40% popustai učestvujete u akcijama kada ostvarujete popuste na sva naša izdanja. Potrebno je samo da se prijavite preko formulara na našem sajtu. Link za prijavu: http://bit.ly/2TxekSa

Skenirajte QR kod
registrujte knjigu
i osvojite nagradu

1

Kako da računarima pružite mogućnost da uče iz podataka

Po mom mišljenju, **mašinsko učenje**, primena i nauka o algoritmima koji imaju smisla za podatke, predstavlja najuzbudljiviju oblast od svih računarskih nauka! Živimo u doba kada postoji ogroman broj podataka; upotrebom samoučećih algoritama iz oblasti mašinskog učenja možemo da pretvorimo ove podatke u znanje. Zahvaljujući mnogobrojnim moćnim bibliotekama otvorenog koda koje su razvijene poslednjih godina, verovatno nikada nije bilo bolje vreme za proboj u oblast mašinskog učenja i za učenje kako da iskoristite moćne algoritme za pronalaženje obrazaca u podacima i za predviđanje o budućim događajima.

U ovom poglavlju ćete učiti o glavnim konceptima i različitim tipovima mašinskog učenja. Zajedno sa osnovnim uvodom u relevantnu terminologiju, postavićemo temelj za uspešnu upotrebu tehnika mašinskog učenja za rešavanje praktičnih problema.

Ovim poglavljem obuhvaćene su sledeće teme:

- osnovni koncepti mašinskog učenja
- tri tipa učenja i osnovna terminologija
- gradivni blokovi za uspešno projektovanje sistema mašinskog učenja
- instaliranje i podešavanje programskog jezika Python za analizu podataka i za mašinsko učenje

Izgradnja inteligentnih mašina za transformisanje podataka u znanje

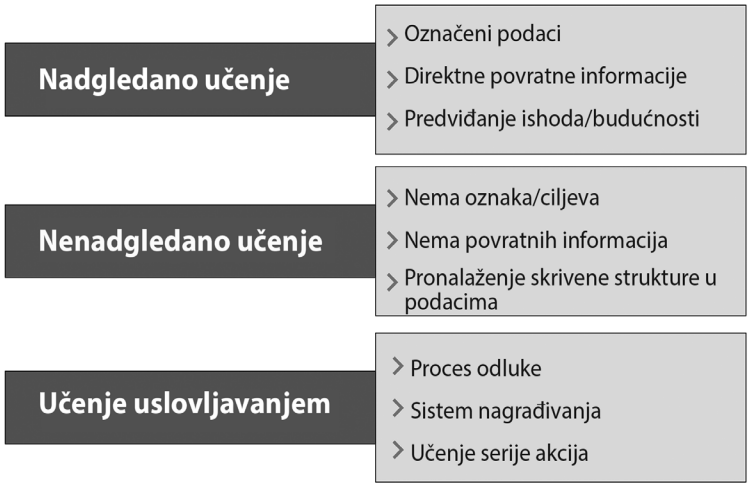
U ovo doba moderne tehnologije dostupan nam je jedan resurs u ogromnim količinama: velika količina strukturiranih i nestrukturiranih podataka. U drugoj polovini 20. veka mašinsko učenje se razvijalo kao podoblast **veštačke inteligencije (artificial intelligence - AI)** i uključivalo je algoritme koji samostalno uče, a za predviđanja izvode znanje iz podataka.

Umesto potrebe da ljudi ručno izvode pravila i grade modele analiziranjem velike količine podataka, mašinsko učenje pruža efikasniju alternativu za čuvanje znanja u podacima za postepeno poboljšanje performansi prediktivnih modela i donošenje odluka vođenih podacima.

Ne samo da mašinsko učenje postaje sve važnije u istraživanjima računarske nauke, već igra sve veću ulogu i u svakodnevnom životu. Zahvaljujući mašinskom učenju, mi uživamo u robusnim filterima neželjene pošte, softverima za prepoznavanje teksta i glasa, pouzdanim veb pretraživačima, preporukama za gledanje interesantnih filmova, mobilnim depozitima čekova, procenjenom vremenu za isporuku pošiljke i još mnogo toga. Nadamo se da ćemo uskoro imati i bezbedne i efikasne samovozeće automobile na ovoj listi. Osim toga, primetan je napredak i u primeni u medicini; na primer, istraživači su predstavili da modeli dubokog učenja mogu da detektuju rak kože skoro ljudskom tačnošću (https://www.nature.com/articles/nature21056). Veliki napredak su nedavno postigli i istraživači u DeepMindu, koji su koristili duboko učenje za predviđanje 3D proteinskih struktura i prvi put su nadmašili rezultate pristupa zasnovanih na fizici (https://deepmind.com/blog/artucke/alphafold-a-solution-to-a-50-year-old-grand-challenge-in-biology). Iako tačna predviđanja 3D proteinske strukture igraju važnu ulogu u istraživanjima u oblasti biologije i farmacije, u poslednje vreme postoji mnogo važnih primena mašinskog učenja u oblasti zdravstva. Na primer, istraživači su projektovali sisteme za predviđanje potrebe za kiseonikom za COVID-19 pacijente za četiri dana unapred da bi pomogli bolnicama da pripreme resurse za te potrebe (https://ai.facebook.com/blog/new-ai-research-to-help-predict-covid-19-resource-needs-from-a-series-of-x-rays/). Još jedna važna tema su klimatske promene, koja predstavlja jedan od najvećih i najvažnijih izazova. U današnje vreme se mnogo pažnje usmerava na razvoj inteligentnih sistema koji se bore protiv ovih klimatskih promena (https://www.forbes.com/sites/robtoews/2021/06/20/these-are-the-startups-applying-ai-to-tackle-climate-change). Jedan od mnogih pristupa borbi protiv klimatskih promena je nova oblast precizne poljoprivrede. Ovde istraživači imaju za cilj da projektuju sisteme mašinskog učenja zasnovane na računarskoj viziji za optimizovanje raspoređivanja resursa da bi smanjili upotrebu i rasipanje đubriva.

Tri različita tipa mašinskog učenja

U ovom odeljku ćemo opisati tri tipa mašinskog učenja: **nadgledano učenje, nenadgledano učenje i učenje uslovljavanjem**. Učićemo o osnovnim razlikama između ta tri različita tipa učenja i upotrebom konceptualnih primera ćemo razviti razumevanje domena praktičnih problema u kojima se oni mogu primeniti:

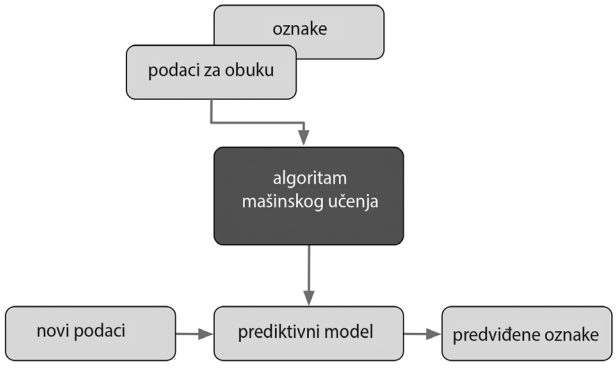


Slika 1.1 Tri različita tipa mašinskog učenja

Predviđanje budućnosti pomoću nadgledanog učenja

Glavni cilj nadgledanog učenja je da obučimo model iz označenih podataka za obuku koji omogućavaju da izvršimo predviđanje o neviđenim ili budućim podacima. Ovde se termin „nadgledano" odnosi na skup primera za obuku (ulaznih podataka) gde su signali željenog izlaza (oznake) već poznati. Prema tome, nadgledano učenje je proces modelovanja odnosa između ulaznih podataka i oznaka. Stoga, takođe možemo da smatramo nadgledano učenje kao „učenje oznakama".

Na *slici 1.2* rezimiran je tipičan tok rada nadgledanog učenja, u kojem su prosleđeni označeni podaci za obuku u algoritam mašinskog učenja za usklađivanje prediktivnog modela koji može da izvrši predviđanje na novim, neoznačenim ulaznim podacima:



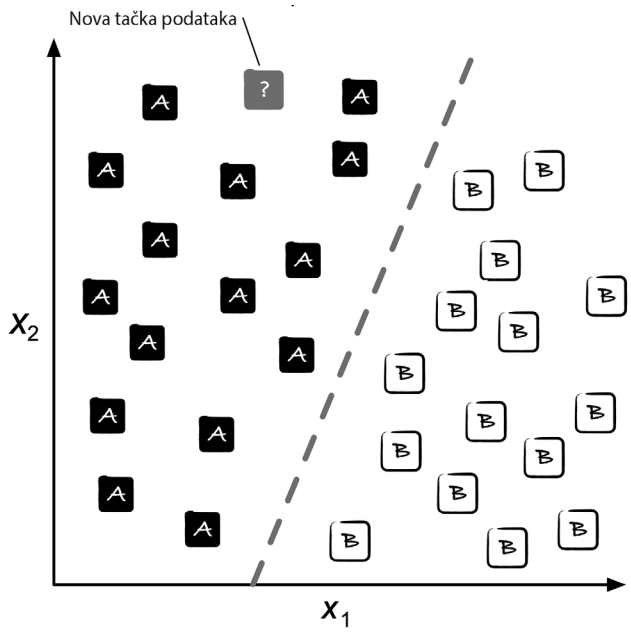
Slika 1.2 Proces nadgledanog učenja

Razmatranjem primera filtriranja neželjene elektronske pošte možemo da obučimo model upotrebom algoritma nadgledanog mašinskog učenja u grupi označenih e-mailova, koji su korektno označeni kao spem ili ne-spem, za predviđanje da li novi e-mail pripada jednoj od ove dve kategorije. Zadatak nadgledanog učenja sa diskretnim oznakama klase, kao što je prethodni primer filtriranja neželjene pošte, naziva se i **zadatak klasifikacije**. Još jedna potkategorija nadgledanog učenja je **regresija**, gde je izlazni signal kontinualna vrednost.

Klasifikacija za predviđanje oznaka klase

Klasifikacija je potkategorija nadgledanog učenja gde je cilj predvideti kategoričke oznake klasa novih instanci ili tačaka podataka na osnovu ranijih opažanja. Te oznake klase su diskretne, neuređene vrednosti koje mogu da se razumeju kao grupno članstvo tačaka podataka. Prethodno pomenuti primer detekcije neželjene pošte predstavlja tipičan primer zadatka binarne klasifikacije, gde algoritam mašinskog učenja uči skup pravila da bi razlikovao dve moguće klase: spam i ne-spam e-mail poruke.

Na *slici 1.3* prikazan je koncept zadatka binarne klasifikacije gde je dato 30 primera za obuku: 15 je označeno kao klasa A, a 15 kao klasa B. U ovom scenariju naš skup podataka je dvodimenzionalan, što znači da svaki primer ima dve povezane vrednosti: x1 i x2. Sada možemo da upotrebimo algoritam nadgledanog mašinskog učenja za učenje pravila (granica odluke predstavljena je isprekidanom linijom) koje može da razdvoji te dve klase i da klasifikuje nove podatke u svaku od te dve kategorije, uzimajući u obzir njihove x1 i x2 vrednosti:



Slika 1.3: Klasifikovanje nove tačke podataka

Međutim, skup oznaka klase ne treba da bude binaran. Prediktivni model naučen pomoću algoritma nadgledanog učenja može da dodeli bilo koju oznaku klase, koja je predstavljena u skupu podataka za obuku, novoj i neoznačenoj tački podataka.

Tipičan primer zadatka **višeklasne klasifikacije** je prepoznavanje ručno pisanog karaktera. Možemo da sakupimo skup podataka za obuku, koji se sastoji od više ručno pisanih primera svakog slova u abecedi. Slova (A, B, C i tako dalje) će predstavljati različite neuređene kategorije ili oznake klasa koje želimo da predvidimo. Ako korisnik unese novi ručno pisani karakter pomoću uređaja za unos, naš prediktivni model će moći da predvidi tačno slovo u abecedi sa određenom tačnošću. Međutim, naš sistem mašinskog učenja neće moći tačno da prepozna bilo koju od cifara između 0 i 9 ako nisu deo skupa podataka za obuku.

Regresija za predviđanje neprekidnog ishoda

U prethodnom odeljku ste naučili da je zadatak klasifikacije dodela kategoričkih, neuređenih oznaka instancama. Drugi tip nadgledanog učenja je predviđanje neprekidnog ishoda, a naziva se i **regresiona analiza**. U regresionoj analizi dati su broj prediktorskih (**objašnjavajućih**) promenljivih i neprekidna ciljna promenljiva (**ishod**) i pokušavamo da pronađemo odnos između tih promenljivih koje nam omogućavaju da predvidimo ishod.

Imajte na umu da u oblasti mašinskog učenja prediktorske promenljive obično nazivamo atributi, a promenljive odgovora obično nazivamo ciljne promenljive. Mi ćemo usvojiti te konvencije u ovoj knjizi.

Na primer, pretpostavimo da smo zainteresovani za predviđanje matematičkih SAT rezultata studenata (SAT je standardizovan test koji se često koristi za prijemni ispit u Sjedinjenim Državama). Ako postoji veza između vremena utrošenog u učenju za test i konačnih rezultata, možemo da je upotrebimo kao podatke za obučavanje modela koji koristi vreme učenja za predviđanje rezultata testa budućih studenata koji planiraju da ga polažu.

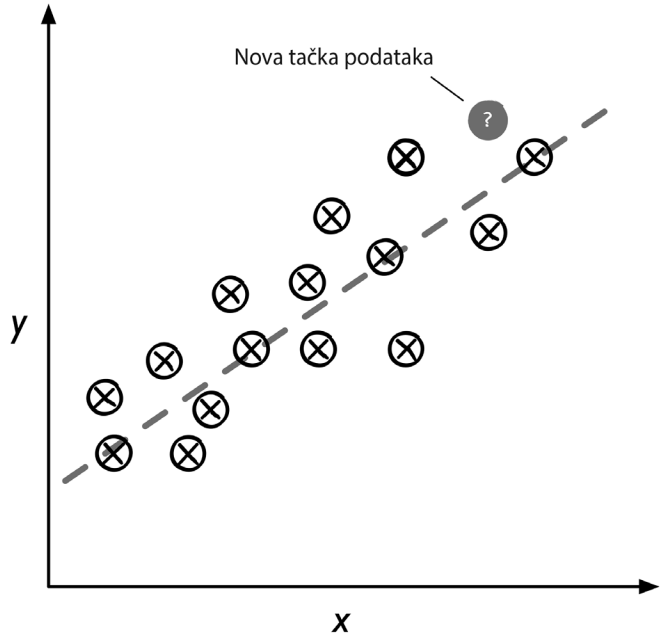
Regresija prema srednjoj vrednosti

Termin regresija osmislio je 1886. godine Francis Galton u svom članku „Regression towards Mediocrity in Hereditary Stature". Galton je opisao biološki fenomen da se varijansa visine populacije ne uvećava vremenom.

Primetio je da se visina roditelja ne prenosi na njihovu decu, već visina njihove dece napreduje ka srednjoj vrednosti populacije.

Na *slici 1.4* ilustrovan je koncept linearne regresije. Dati su atribut x i ciljna promenljiva y. Na osnovu njih ćemo uklopiti pravu liniju za ove podatke, koja minimalizuje odstupanje (obično je to prosečan kvadrat odstupanja) između tačaka podataka i uklopljene linije.

Sada možemo da upotrebimo presek i nagib za predviđanje ciljne promenljive novih podataka:

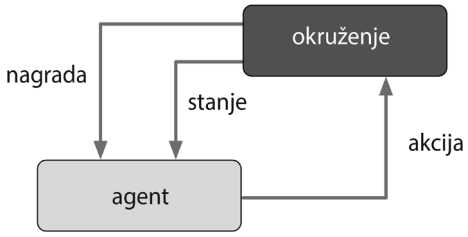


Slika 1.4: Primer linearne regresije

Rešavanje interaktivnih problema pomoću učenja uslovljavanjem

Još jedan tip mašinskog učenja je **učenje uslovljavanjem** (eng. reinforcement learning), u kojem je cilj razviti sistem (**agent**) koji poboljšava svoje performanse na osnovu interakcija sa okruženjem. Pošto informacije o aktuelnom stanju okruženja obično uključuju takozvani **signal nagrade**, možemo da zamislimo učenje uslovljavanjem kao oblast koja se odnosi na nadgledano učenje. Međutim, u učenju uslovljavanjem ova povratna informacija nije tačna oznaka ili vrednost istine, već je to mera koliko dobro je funkcija nagrade izmerila akciju. Kroz interakciju sa okruženjem agent može da upotrebi učenje uslovljavanjem za učenje serije akcija koje maksimalizuju ovu nagradu pomoću istraživačkog pristupa pokušaja i greške ili promišljenim planiranjem.

Popularan primer učenja uslovljavanjem je šah. Ovde agent odlučuje o serijama poteza, u zavisnosti od stanja na tabli (okruženje), a nagrada će biti definisana kao **pobeda** ili **poraz** na kraju igre:



Slika 1.5: Proces učenja uslovljavanjem

Postoje mnogi različiti podtipovi učenja uslovljavanjem. Međutim, osnovna šema je da agent u učenju uslovljavanjem pokuša da maksimalizuje nagradu kroz serije interakcija sa okruženjem. Svako stanje može da bude povezano sa pozitivnom ili negativnom nagradom, a nagrada može da bude definisana kao postizanje uopštenog cilja, kao što su pobeda ili poraz u igri šaha. Na primer, u šahu ishod svakog poteza može da se zamisli kao različito stanje okruženja.

Da bismo dalje istražili primer šaha, razmislimo o određenim konfiguracijama na šahovskoj tabli koje su povezane sa stanjima koja će najverovatnije dovesti do pobede - na primer, uklanjanje šahovske figure protivnika sa table ili pretnja kraljici. Međutim, druge pozicije su povezane sa stanjima koja će najverovatnije dovesti do poraza u igri, kao što je gubitak šahovske figure u korist protivnika u sledećem potezu. Sada nagrada (pozitivna za pobedu, ili negativna za gubitak partije) neće biti data do kraja igre. Osim toga, finalna nagrada će takođe zavisiti od toga kako protivnik igra. Na primer, protivnik može da žrtvuje kraljicu, ali na kraju da pobedi u igri.

Učenje uslovljavanjem se bavi učenjem odabira serije akcija koje maksimalizuju ukupnu nagradu, koja može da se dobije ili odmah nakon izvršavanja akcije ili pomoću *odložene* povratne informacije.

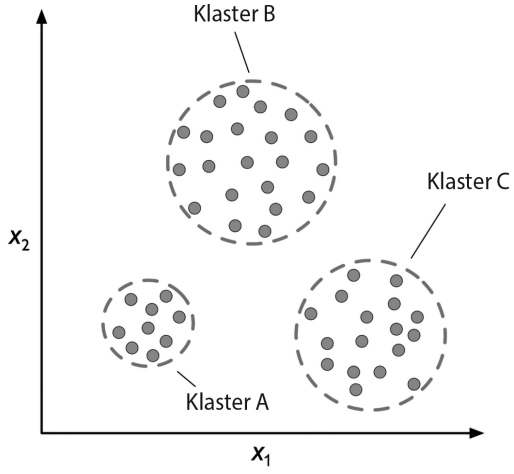
Otkrivanje skrivenih struktura pomoću nenadgledanog učenja

U nadgledanom učenju unapred znamo tačan odgovor (oznaku ili ciljnu promenljivu) kada obučavamo model, a u učenju uslovljavanjem definišemo meru nagrade za određene akcije koje izvršava agent. Međutim, u nenadgledanom učenju koristimo neoznačene podatke ili podatke nepoznate strukture. Upotrebom tehnika nenadgledanog učenja možemo da istražimo strukturu podataka za izdvajanje značajnih informacija, bez smernica poznate promenljive ishoda ili funkcije nagrade.

Pronalaženje podgrupa pomoću klasterovanja

Klasterovanje je istraživačka tehnika analize podataka ili otkrivanja obrazaca koja omogućava da organizujemo grupu informacija u značajne podgrupe (**klastere**), bez potrebe da imamo neko predznanje o članovima grupe. Svaki klaster koji se javlja u toku analize definiše grupu objekata koji dele određeni stepen sličnosti, ali su više različiti od objekata u drugim klasterima; zbog toga se klasterovanje ponekad naziva i **nenadgledana klasifikacija**. Klasterovanje je odlična tehnika za strukturiranje informacija i za izvođenje značajnih odnosa iz podataka. Na primer, omogućava trgovcima da otkriju grupe kupaca na osnovu njihovog interesovanja da bi razvili različite marketinške programe.

Na *slici 1.6* prikazano je kako klasterovanje može da se primeni za organizovanje neoznačenih podataka u tri različite grupe ili klastera (A, B i C, u proizvoljnom redosledu) na osnovu sličnosti njihovih atributa x1 i x2:

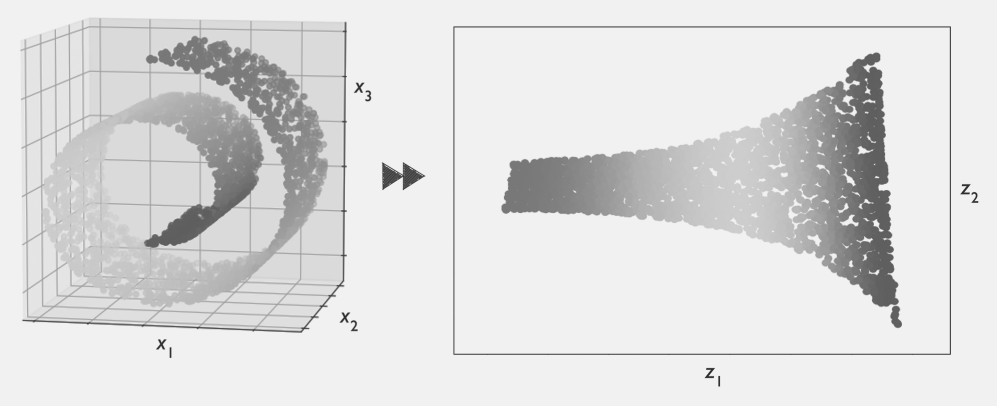


Slika 1.6: Kako funkcioniše klasterovanje

Redukcija dimenzionalnosti za kompresovanje podataka

Još jedna podoblast nenadgledanog učenja je **redukcija dimenzionalnosti**. Često koristimo podatke visoke dimenzionalnosti (svako opažanje uključuje visok broj merenja), što može predstavljati izazov za ograničavanje prostora za skladištenje i performanse izračunavanja za algoritme mašinskog učenja. Nenadgledana redukcija dimenzionalnosti je pristup koji se uobičajeno koristi u pretprocesiranju atributa za uklanjanje šuma iz podataka, koji takođe može da degradira prediktivnu performansu određenih algoritama. Redukcija dimenzionalnosti kompresuje podatke u manje dimenzionalne podoblasti dok se zadržavaju najrelevantnije informacije.

Redukcija dimenzionalnosti može ponekad da bude korisna i za vizuelizaciju podataka; na primer visokodimenzionalni skup atributa može da bude projektovan u jednodimenzionalnim, dvodimenzionalnim ili trodimenzionalnim prostorima atributa da bi se vizuelizovali pomoću 2D ili 3D dijagrama rasturanja ili histograma. Na *slici 1.7* prikazan je primer gde je primenjena nelinearna redukcija dimenzionalnosti za kompresovanje 3D Swiss Rolla u novi 2D potprostor atributa:



Slika 1.7: Primer redukcije dimenzionalnosti iz trodimenzionalnog u dvodimenzionalni prikaz

Uvod u osnovnu terminologiju i notacije

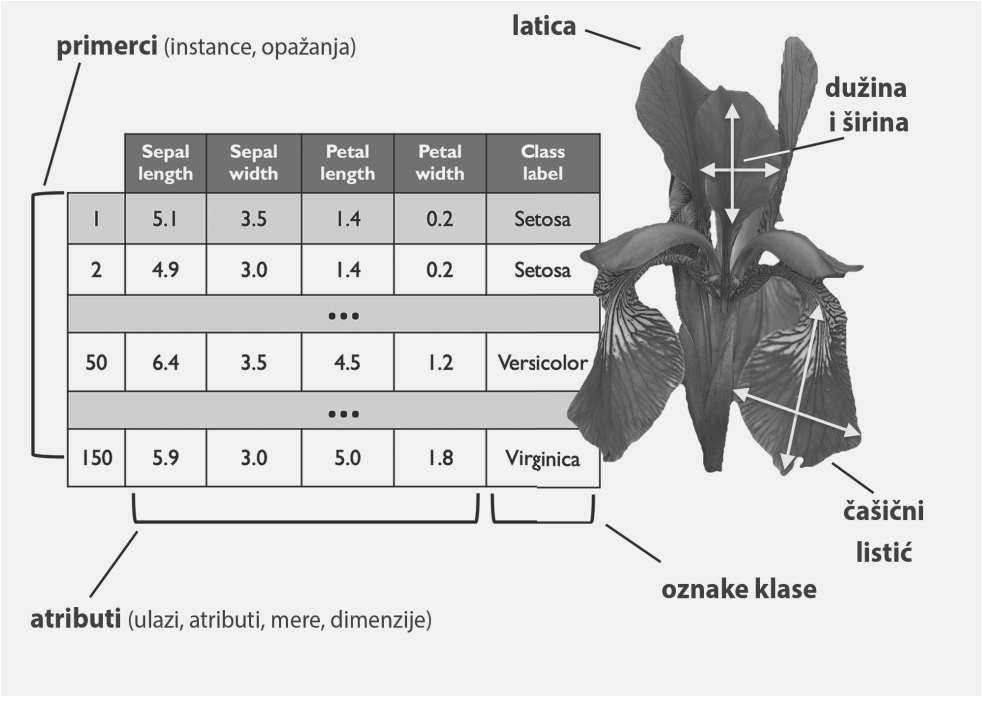
Sada, kada smo opisali tri široke kategorije mašinskog učenja - nadgledano, nanadgledano učenje i učenje uslovljavanjem, predstavićemo osnovnu terminologiju koju ćemo koristiti u ovoj knjizi. U sledećim odeljcima opisaćemo uobičajene termine koje ćemo koristiti kada govorimo o različitim aspektima skupa podataka i matematičke notacije koje će nam pomoći da preciznije i efikasnije komuniciramo.

Pošto je mašinsko učenje ogromna i interdisciplinarna oblast, pre ili kasnije sigurno ćete naići na mnogo različitih termina koji se odnose na iste koncepte. Najčešće upotrebljavane termine koji se mogu pronaći u literaturi mašinskog učenja opisaćemo u drugom pododeljku, koji može biti koristan kao referentan odeljak kada čitate raznovrsnu literaturu o mašinskom učenju.

Notacije i konvencije upotrebljene u ovoj knjizi

Na *slici 1.8* prikazan je deo skupa podataka Iris, koji je tipičan primer za oblast mašinskog učenja (pronaći ćete više informacija na adresi https://archive.ics.uci.edu/ml/datasets/iris). Iris skup podataka sadrži mere 150 cvetova Irisa iz tri različite vrste - Setosa, Versicolor i Virginica.

Ovde svaki primer cveta predstavlja jedan red u skupu podataka, a mere cveta u centimetrima su sačuvane kao kolone, koje nazivamo i **atributi** skupa podataka:



	Sepal length	Sepal width	Petal length	Petal width	Class label
1	5.1	3.5	1.4	0.2	Setosa
2	4.9	3.0	1.4	0.2	Setosa
...					
50	6.4	3.5	4.5	1.2	Versicolor
...					
150	5.9	3.0	5.0	1.8	Virginica

Slika 1.8: Skup podataka Iris

Da bi notacija i implementacija ostale jednostavne, a ipak efikasne, upotrebićemo osnove linearne algebre. U sledećim poglavljima ćemo upotrebiti notaciju matrice za referenciranje podataka. Pratićemo uobičajenu konvenciju za predstavljanje svakog primera kao posebnog reda u matrici atributa X, gde je svaki atribut sačuvan u posebnoj koloni.

Iris skup podataka se sastoji od 150 primera i četiri atributa i može da bude napisan kao matrica 150×4, formalno označena kao $\boldsymbol{X} \in \mathbb{R}^{150\times 4}$:

$$\begin{bmatrix} x_1^{(1)} & x_2^{(1)} & x_3^{(1)} & x_4^{(1)} \\ x_1^{(2)} & x_2^{(2)} & x_3^{(2)} & x_4^{(2)} \\ \vdots & \vdots & \vdots & \vdots \\ x_1^{(150)} & x_2^{(150)} & x_3^{(150)} & x_4^{(150)} \end{bmatrix}$$

Konvencije notacije

U ostatku ove knjige, osim ako je naznačeno drugačije, upotrebićemo superskript i da bismo ukazali na i-ti primer obuke i indeks j da bismo ukazali na j-tu dimenziju skupa podataka za obuku.

Upotrebićemo mala podebljana slova da bismo ukazali na vektore ($\boldsymbol{x} \in \mathbb{R}^{\boldsymbol{n}\times\boldsymbol{1}}$), velika podebljana slova da bismo ukazali na matrice ($\boldsymbol{X} \in \mathbb{R}^{\boldsymbol{n}\times\boldsymbol{m}}$ i iskošena slova ($x^{(n)}$ ili $x_m^{(n)}$) da bismo ukazali na pojedinačne elemente u vektoru ili matrici.

Na primer, $x_1^{(150)}$ se odnosi na prvu dimenziju primera cveta 150, odnosno sepal length. Prema tome, matrica X predstavlja jednu instancu cveta i može da bude napisana kao četvorodimenzionalni neobrađeni vektor $\boldsymbol{x}^{(i)} \in \mathbb{R}^{1\times4}$:

$$\boldsymbol{X}^{(i)} = \begin{bmatrix} x_1^{(i)} & x_2^{(i)} & x_3^{(i)} & x_4^{(i)} \end{bmatrix}$$

I svaka dimenzija atributa je 150-dimenzionalni vektor kolone $\boldsymbol{X}^{(i)} \in \mathbb{R}^{150\times1}$ - na primer:

$$\boldsymbol{x_j} = \begin{bmatrix} x_j^{(1)} \\ x_j^{(2)} \\ \cdots \\ x_j^{(150)} \end{bmatrix}$$

Slično tome, predstavićemo ciljne promenljive (oznake klasa) kao 150-dimenzionalni vektor kolone:

$$\boldsymbol{y} = \begin{bmatrix} y^{(1)} \\ \cdots \\ y^{(150)} \end{bmatrix}, \text{ where } y^{(i)} \in \{\text{Setosa, Versicolor, Virginica}\}$$

Terminologija mašinskog učenja

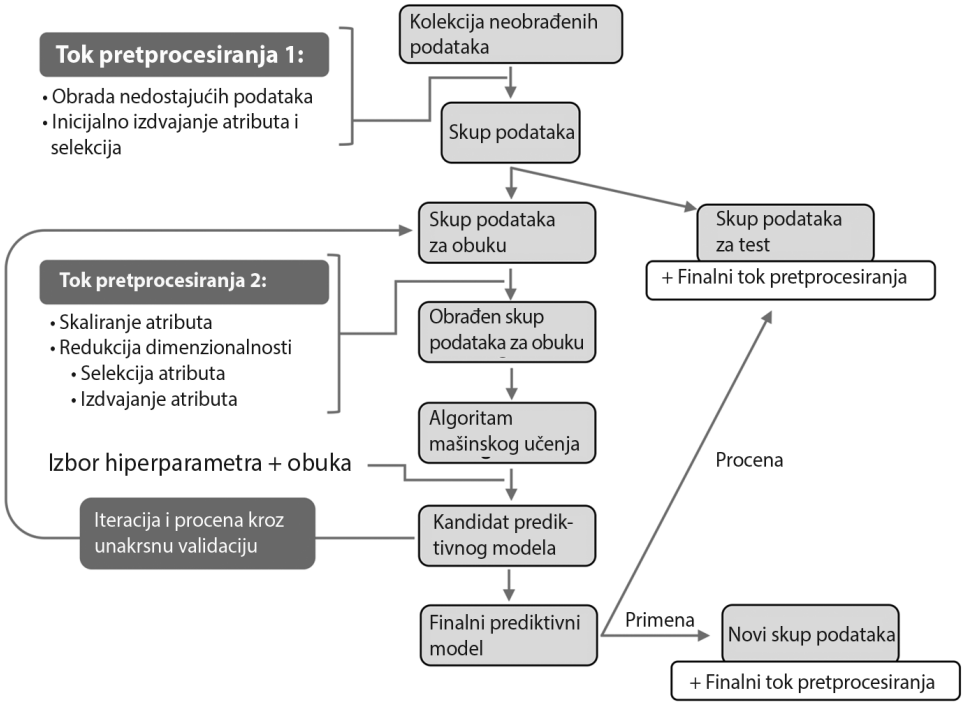
- Mašinsko učenje je obimna i interdisciplinarna oblast, jer spaja mnoge naučnike iz drugih oblasti istraživanja. Kao što se često dešava, mnogi termini i koncepti su ponovo otkriveni ili redefinisani i možda su vam već biti poznati, ali se javljaju pod različitim nazivima. U sledećoj listi možete pronaći selekciju uobičajeno upotrebljavanih termina i njihovih sinonima, koji će vam biti korisni dok čitate ovu knjigu, ili drugu literaturu o mašinskom učenju:
- **primer za obučavanje**- red u tabeli koji predstavlja skup podataka i sinonim je za opažanje, zapis, instancu ili uzorak (u većini konteksta, uzorak se odnosi na kolekciju primera za obučavanje)
- **obučavanje** - uklapanje modela za parametarske modele slično proceni parametara
- **atribut, skraćeno x** - kolona u tabeli podataka ili matrici podataka (dizajn); sinonim je za prediktor, promenljivu, ulaz, atribut ili kovarijanse

- **cilj, skraćeno y** - sinonim za ishod, izlaz, promenljivu odgovora, zavisnu promenljivu, oznaku (klase) i istine
- **funkcija gubitka** - Često se koristi kao sinonim za *cost* funkciju. Ponekad se funkcija gubitka naziva i *error* funkcija. U nekoj literaturi termin gubitak se odnosi na gubitak meren iz jedne tačke podataka, a cost je mera koja izračunava gubitak (prosečno ili zbirno) u celom skupu podataka.

Mapa za izgradnju sistema mašinskog učenja

U prethodnim odeljcima smo predstavili osnovne koncepte mašinskog učenja i tri različita tipa učenja. U ovom odeljku ćemo govoriti o drugim važnim delovima sistema mašinskog učenja, zajedno sa algoritmima učenja.

Na *slici 1.9* prikazan je tipičan tok rada mašinskog učenja u prediktivnom modelovanju, o čemu ćemo govoriti u sledećim pododeljcima:



Slika 1.9: Tok rada prediktivnog modelovanja

Pretprocesiranje - oblikovanje podataka

Prvo ćemo opisati mapu za izgradnju sistema mašinskog učenja. Neobrađeni podaci retko dolaze u obliku koji je potreban za optimalne performanse algoritma mašinskog učenja. Prema tome, pretprocesiranje podataka je jedan od najvažnijih koraka u svakoj primeni mašinskog učenja.

Ako, kao primer, upotrebimo skup podataka Iris cveta iz prethodnog odeljka, možemo da zamislimo neobrađene podatke kao seriju slika cveća, iz kojih možemo da izdvojimo značajne atribute. Korisni atributi mogu da budu boja cvetova ili visina, dužina i širina cvetova.

Mnogi algoritmi mašinskog učenja takođe zahtevaju da su izabrani atributi na istoj skali za optimalnu performansu, što se često postiže transformisanjem atributa u rasponu [0, 1] ili pomoću standardne normalne raspodele srednje vrednosti i jedinične varijanse, kao što ćete videti u sledećim poglavljima.

Neki od izabranih atributa mogu da budu povezani i, prema tome, suvišni do određenog stepena. U tim slučajevima tehnike redukcije dimenzionalnosti su korisne za kompresovanje atributa u potprostore niže dimenzije. Redukcija dimenzionalnosti prostora atributa ima prednosti, jer je potrebno manje prostora za skladištenje, a algoritam obučavanja može da se pokreće mnogo brže. U određenim slučajevima redukcija dimenzionalnosti takođe može da poboljša prediktivnu performansu modela ako skup podataka sadrži veliki broj nerelevantnih atributa (ili šum), odnosno ako skup podataka ima slab odnos signala i šuma.

Da bismo odredili da li se algoritam mašinskog učenja izvršava dobro u skupu podataka za obučavanje i da li se dobro generalizuje u novim podacima, takođe ćemo nasumično da razdvojimo skup podataka u poseban skup podataka za obučavanje i skup podataka za testiranje. Upotrebićemo skup podataka za obučavanje i optimizaciju modela mašinskog učenja, dok ćemo zadržati skup podataka za testiranje do samog kraja za procenu finalnog modela.

Obučavanje i selektovanje prediktivnog modela

Kao što ćete videti u narednim poglavljima, mnogo različitih algoritama mašinskog učenja je razvijeno za rešavanje različitih problema. Važna tačka koja se može rezimirati iz poznatog članka Davida Wolperta *„No free lunch theorems“* je da ne možemo da učimo „besplatno“ (*„The Lack of A Priori Distinctions Between Learning Algorithms“*, D. H. Wolpert, 1996; *„No free lunch theorems for optimization“*, D. H. Wolpert i W. G. Macready, 1997). Ovaj koncept možemo povezati sa popularnom izrekom Abrahama Maslowa (1966) *„Ako vam je jedini alat čekić, pretpostavljam da je primamljivo da sve tretirate kao ekser“*. Na primer, svaki algoritam klasifikacije ima svoje nerazdvojive pomake i ni jedan model klasifikacije nije superioran ako ne postoje pretpostavke u vezi sa zadatkom. Prema tome, u praksi je važno uporediti bar nekoliko različitih algoritama učenja da bismo obučili i selektovali model sa najboljom performansom. Međutim, pre nego što uporedimo različite modele, prvo bi trebalo da donesemo odluku o metrici za merenje performansi. Jedna od uobičajeno upotrebljavanih metrika je tačnost klasifikacije, koja je definisana kao udeo tačno klasifikovanih instanci.

Kako možemo da znamo koji se model dobro izvršava u finalnom skupu podataka za testiranje i u podacima iz realnog sveta ako ne koristimo ovaj skup podataka za testiranje za selekciju modela, već ga zadržavamo za procenu finalnog modela? Da bismo rešili taj problem, možemo upotrebiti različite tehnike koje nazivamo unakrsna validacija. U unakrsnoj validaciji mi dalje razdvajamo skup podataka u podskupove za obučavanje i validaciju da bismo procenili performansu generalizacije modela.

Na kraju, takođe ne možemo da očekujemo da će podrazumevani parametri različitih algoritama za učenje koje obezbeđuju biblioteke softvera biti optimalni za specifičan problem. Prema tome, u narednim poglavljima često ćemo koristiti tehnike optimizacije hiperparametara koje nam pomažu da fino podesimo performanse modela.

Možemo čak da zamislimo te hiperparametre kao parametre koji nisu naučeni iz podataka, već predstavljaju ručice modela koje možemo da okrenemo da bismo poboljšali performanse konkretnog modela. Sve ovo će vam postati mnogo jasnije u narednim poglavljima kada budete videli stvarne primere.

Procena modela i predviđanje neviđenih instanci podataka

Nakon što smo selektovali model koji je usklađen u skupu podataka za obučavanje, možemo da upotrebimo skup podataka za testiranje da bismo procenili koliko se model dobro izvršava u ovim neviđenim podacima da bismo procenili takozvanu *grešku generalizacije*. Ako smo zadovoljni performansom, možemo da upotrebimo ovaj model za predviđanje novih, budućih podataka. Važno je da naglasimo da se parametri za prethodno pomenute procedure, kao što su skaliranje atributa i redukcija dimenzionalnosti, dobijaju samo iz skupa podataka za obučavanje, a isti parametri su kasnije ponovo primenjeni za transformisanje skupa podataka za testiranje, kao i bilo koje instance novih podataka - performansa merena u skupu podataka za testiranje može, u suprotnom, biti preterano optimistična.

Upotreba programskog jezika Python za mašinsko učenje

Python je jedan od najpopularnijih programskih jezika za istraživanje podataka. Zahvaljujući veoma aktivnim programerima i zajednici otvorenog koda, razvijen je veliki broj korisnih biblioteka za naučna izračunavanja i mašinsko učenje.

Iako su performanse interpretiranih jezika, kao što je Python, za računski intenzivne zadatke inferiorne u odnosu na programske jezike nižeg nivoa, razvijene su proširene biblioteke, kao što su NumPy i SciPy, za nadgradnju Fortran i C implementacija nižeg nivoa za brze vektorizovane operacije u višedimenzionalnim nizovima.

Za zadatke programiranja mašinskog učenja uglavnom ćemo koristiti biblioteku scikit-learn, koja je trenutno najpopularnija biblioteka mašinskog učenja otvorenog koda. U narednim poglavljima, kada se budemo fokusirali na podoblast mašinskog učenja pod nazivom *duboko učenje*, upotrebićemo najnoviju verziju PyTorch biblioteke, koja je specijalizovana za veoma efikasno obučavanje takozvanih modela *duboke neuronske mreže* upotrebom grafičkih kartica.

Instaliranje Pythona i paketa iz Python Package Indexa

Python je dostupan za sva tri glavna operativna sistema - Microsoft Windows, macOS i Linux, a instalacioni fajl i dokumentacija mogu da se preuzmu sa zvaničnog Python veb sajta https://www.python.org.

Primeri koda iz ove knjige su napisani za Python verziju 3.9; preporučujemo da upotrebite najnoviju verziju Pythona 3, koja je trenutno dostupna. Neki kod može da bude kompatibilan sa verzijom Python 2.7, ali pošto je zvanična podrška za Python 2.7 okončana 2019. godine, a većina biblioteka otvorenog koda je već prestala da podržava Python 2.7 (https://python-3statement.org), preporučujemo da upotrebite Python 3.9 ili noviju verziju.

Možete da proverite aktuelnu Python verziju pomoću sledeće komande:

```
python --version
```

Ili

```
python3 --version
```

u terminalu (ili PowerShell-u ako koristite Windows).

Dodatni paketi koje ćemo koristiti u ovoj knjizi mogu da budu instalirani pomoću pip instalacionog programa, koji je deo Python Standard Library biblioteke od verzije Python 3.3. Više informacija o pip programu možete pronaći na adresi https://docs.python.org/3/installing/index.html.

Nakon što uspešno instalirate Python, možete da izvršite pip iz terminala da biste instalirali dodatne Python pakete:

```
pip install SomePackage
```

Već instalirani paketi mogu da budu ažurirani pomoću oznake --upgrade:

```
pip install SomePackage --upgrade
```

Upotreba Anaconda Python distribucije i upravljača paketima

Preporučen sistem za upravljanje paketima otvorenog koda za instaliranje programskog jezika Python za kontekste naučnih izračunavanja je conda od kompanije Continuum Analytics. Conda je besplatna i licencirana je pod licencom otvorenog koda. Cilj je da pomogne u instaliranju i upravljanju verzijama Python paketa za istraživanje podataka, matematiku i inženjerstvo na različitim operativnim sistemima. Ako želite da upotrebite sistem conda, možete da birate različite verzije: Anaconda, Miniconda i Miniforge:

- Anaconda ima mnogo unapred instaliranih paketa za naučna izračunavanja. Instalacioni program Anaconda možete da preuzmete sa adrese https://docs.anaconda.com/anaconda/install/, a vodič za upotrebu Anaconda sistema dostupan je na adresi https://docs.anaconda.com/anaconda/user-guide/getting-started/.
- Miniconda je jednostavnija alternativa za Anaconda sistem (https://docs.conda.io/en/latest/miniconda.html). U suštini je sličan Anaconda sistemu ali bez unapred instaliranih paketa, što mnogo ljudi (uključujući i autore) preferira.
- Miniforge je sličan Miniconda sistemu, ali ga održava zajednica i koristi drugačija skladišta paketa (conda-forge) od Miniconda i Anaconda sistema. Smatramo da je Miniforge odlična alternativa Miniconda sistemu. Instalacioni program i instrukcije za instalaciju možete pronaći na GitHub skladištu, na adresi https://github.com/conda-forge/miniforge.

Nakon što uspešno instalirate Anaconda, Miniconda ili Miniforge sistem, možete da instalirate nove Python pakete upotrebom sledeće komande:

```
conda install SomePackage
```

Postojeći paketi mogu da se ažuriraju upotrebom sledeće komande:

```
conda update SomePackage
```

Paketi koji nisu dostupni na zvaničnom conda kanalu, dostupni su putem conda-forge projekta, koji zajednica podržava (https://conda-forge.org), koji mogu da budu specifikovani oznakom --channel conda-forge. Na primer:

```
conda install SomePackage --channel conda-forge
```

Paketi koji nisu dostupni na podrazumevanom conda kanalu ili conda-forge projektu mogu da budu instalirani pomoću programa pip, kao što je ranije opisano. Na primer:

```
pip install SomePackage
```

Paketi za naučna izračunavanja, istraživanje podataka i mašinsko učenje

U prvoj polovini ove knjige ćemo uglavnom koristiti višedimenzionalne nizove biblioteke NumPy za skladištenje podataka i manipulisanje njima. Povremeno ćemo upotrebiti pandas biblioteku, koja je izgrađena na osnovi NumPy biblioteke, a obezbeđuje dodatne alatke za manipulaciju podacima višeg nivoa i olakšava upotrebu tabelarnih podataka. Da biste poboljšali iskustvo u učenju i vizuelizovali kvantitativne podatke, što je često veoma korisno da biste ih bolje razumeli, upotrebite veoma prilagodljivu biblioteku Matplotlib.

Glavna biblioteka mašinskog učenja koju ćemo upotrebiti u ovoj knjizi je scikit-learn (*poglavlja od 3 do 11*). U *poglavlju 12 „Paralelizovanje obučavanja neuronske mreže pomoću biblioteke PyTorch"* ćemo predstaviti biblioteku PyTorch za duboko učenje.

Brojevi verzija glavnih Python paketa koji su upotrebljeni za pisanje ove knjige prikazani su u sledećoj listi. Uverite se da su brojevi verzija vaših instaliranih paketa isti da biste bili sigurni da se primeri koda pravilno pokreću:

- NumPy 1.21.2
- SciPy 1.7.0
- scikit-learn 1.0
- Matplotlib 3.4.3
- pandas 1.3.2

Nakon instaliranja ovih paketa možete dodatno da proverite instalirane verzije importovanjem paketa u Python i pristupanjem njegovom atributu `_version_`, na primer:

```
>>> import numpy
>>> numpy.__version__
'1.21.2'
```

Dodali smo i `python-environment-check.py` skript u besplatno skladište kodova za ovu knjigu, na adresi https://github.com/rasbt/machine-learning-book, da biste izvršenjem ovog skripta mogli da proverite Python verziju i verzije paketa.

Za određena poglavlja potrebni su dodatni paketi i biće obezbeđene informacije o instalacijama. Na primer, ne brinite sada o instaliranju PyTorch biblioteke. U *poglavlju 12* pronaći ćete savete i instrukcije kada vam budu potrebne.

Ako pronađete greške čak i ako se vaš kod podudara sa kodom u poglavlju, preporučujemo da prvo proverite brojeve verzija osnovnih paketa, pre nego što utrošite vreme na otklanjanje greške ili se obratite izdavaču ili autorima. Ponekad, novije verzije biblioteka prikazuju promene koje nisu kompatibilne sa starijim verzijama, što može da bude objašnjenje za te greške.

Ako ne želite da promenite verziju paketa u glavnoj Python instalaciji, preporučujemo da upotrebite virtuelno okruženje za instaliranje paketa upotrebljenih u ovoj knjizi. Ako koristite Python bez conda upravljača, možete da upotrebite `venv` biblioteku za kreiranje novog virtuelnog okruženja. Na primer, možete da kreirate i aktivirate virtuelno okruženje pomoću sledeće dve komande:

```
python3 -m venv /Users/sebastian/Desktop/pyml-book
source /Users/sebastian/Desktop/pyml-book/bin/activate
```

Imajte na umu da je potrebno da aktivirate virtuelno okruženje svaki put kada otvorite nov terminal ili PowerShell. O tome možete pronaći više informacija na stranici https://docs.python.org/3/library/venv.html.

Ako koristite Anaconda sistem sa conda upravljačem paketa, možete da kreirate i aktivirate virtuelno okruženje na sledeći način:

```
conda create -n pyml python=3.9
conda activate pyml
```

Rezime

U ovom poglavlju smo istražili mašinsko učenje na veoma visokom nivou i videli ste „širu sliku“ i glavne koncepte koje ćemo istražiti detaljnije u sledećim poglavljima. Naučili ste da je nadgledano učenje sastavljeno od dve važne podoblasti: od klasifikacije i regresije. Dok nam modeli klasifikacije omogućavaju da kategorizujemo objekte u poznate klase, regresionu analizu možemo da upotrebimo za predviđanje kontinualnih ishoda ciljnih promenljivih. Nenadgledano učenje ne obezbeđuje samo korisne tehnike za otkrivanje struktura u neoznačenim podacima, već može da bude korisno i za kompresovanje podataka u koracima pretprocesiranja atributa.

Ukratko smo pregledali tipičnu mapu za primenu mašinskog učenja za rešavanje problema, što ćemo upotrebiti kao osnovu za detaljnije razmatranje i praktične vežbe u narednim poglavljima. Na kraju smo podesili Python okruženje i instalirali i ažurirali potrebne pakete da bismo se pripremili za pregled mašinskog učenja u akciji.

Kasnije u ovoj knjizi, osim samog mašinskog učenja, predstavićemo i različite tehnike za pretpocesiranje skupa podataka, što će vam pomoći da dobijete najbolje performanse iz različitih algoritama mašinskog učenja. Detaljno ćemo opisati algoritme klasifikacije, a takođe ćemo istražiti različite tehnike za regresionu analizu i klasterovanje.

Pred vama je veoma uzbudljiv „put“, jer ćemo opisati mnoge moćne tehnike u ogromnoj oblasti mašinskog učenja. Međutim, pristupićemo mašinskom učenju korak po korak, postepeno nadgrađujući znanje kroz poglavlja knjige. U sledećem poglavlju ćemo započeti ovo „putovanje“ implementiranjem jednog od najranijih algoritama mašinskog učenja za klasifikaciju, što će nas pripremiti za *poglavlje 3, „Predstavljanje klasifikatora mašinskog učenja pomoću biblioteke scikit-learn“*, u kome ćemo opisati naprednije algoritme mašinskog učenja upotrebom biblioteke mašinskog učenja scikit-learn otvorenog koda.

Pridružite se Discord prostoru ove knjige

Pridružite se Discord radnom prostoru ove knjige, da biste učestvovali u Ask me Anything sesijama sa autorima: https://packt.link/MLwPyTorch