

UNIVERZITET U BEOGRADU  
MATEMATIČKI FAKULTET

Nikola Z. Vidič

# **PRIMENA MAŠINSKOG UČENJA U VERIFIKACIJI SOFTVERA**

master rad

Beograd, 2018.

**Mentor:**

dr Milena VUJOŠEVIĆ JANIČIĆ, docent  
Univerzitet u Beogradu, Matematički fakultet

**Članovi komisije:**

dr Mladen NIKOLIĆ, docent  
Univerzitet u Beogradu, Matematički fakultet

dr Filip MARIĆ, vanredni profesor  
Univerzitet u Beogradu, Matematički fakultet

**Datum odbrane:** \_\_\_\_\_



**Naslov master rada:** Primena mašinskog učenja u verifikaciji softvera

**Rezime:**

**Ključne reči:**

# Sadržaj

<b>1</b>	<b>Uvod</b>	<b>1</b>
<b>2</b>	<b>Mašinsko učenje</b>	<b>2</b>
2.1	Algoritam $k$ najbližih suseda . . . . .	3
2.1.1	Dodeljivanje klase objektu . . . . .	4
2.2	Logistička regresija . . . . .	5
2.3	Slučajne šume . . . . .	7
2.4	Mašinsko učenje u Python-u . . . . .	10
2.4.1	SciPy ekosistem . . . . .	10
<b>3</b>	<b>Verifikacija softvera</b>	<b>12</b>
<b>4</b>	<b>Zaključak</b>	<b>13</b>
	<b>Bibliografija</b>	<b>14</b>

# Glava 1

## Uvod

## Glava 2

# Mašinsko učenje

Sama ideja *mašinskog učenja* (eng. *machine learning*) javlja se još četrdesetih godina dvadesetog veka u radovima Alana Turinga (Alan Turing). Razvoj mašinskog učenja vođen je željom da se razume i oponaša ljudski potencijal za učenje. Pedesetih godina mašinsko učenje se razvija zajedno sa pojmom perceptrona, pretka neuronskih mreža. Razvoj mašinskog učenja u formi neuronskih mreža nastavlja se sve do devedesetih godina. Početkom dvehiljaditih dešava se proboj na polju razvoja *veštačke inteligencije* (eng. *artificial intelligence*) i mnogi problemi za koje se smatralo da će jos dugo ostati nerešeni bivaju rešeni, velikim delom zahvaljujući mašinskom učenju. Uopšteno, „mašinsko učenje predstavlja disciplinu koja se bavi konstrukcijom sistema koji se prilagođavaju i popravljaju svoje performanse sa povećanjem iskustva, oličenog u količini relevantnih podataka”[mladen]. Mašinsko učenje proučava indukcijski način zaključivanja tj. generalizaciju (uopštavanje ka univerzalnim zaključcima). Primene mašinskog učenja su brojne: prepoznavanje lica na slikama, oblika na slikama, prepoznavanje tumora, autonomno upravljanje automobilima, autonomno letenje, igranje igara na tabli kao što je šah, klasifikacija teksta, prepoznavanje cifara i mnoge druge [12].

Mašinsko učenje za rešavanje problema koristi različite metode. Ove metode se prema prirodi problema učenja svrstavaju u jednu od tri grupe: *nadgledano učenje* (eng. *supervised learning*), *nenadgledano učenje* (eng. *unsupervised learning*) ili *učenje potkrepljivanjem* (eng. *reinforcement learning*). Nadgledano učenje se odlikuje ulaznim podacima tj. podacima na osnovu kojih se uči i izlaznim podacima tj. podacima koje je potrebno naučiti. Naziv je posledica sličnosti postupka nadgledanog učenja i učenja u kome profesor zada učeniku zadatke i nakon što ih učenik reši, da

učeniku odgovore radi poređenja rezultata. Algoritmi mašinskog učenja se opisuju kao učenje ciljne funkcije  $f$  koja najbolje opisuje vezu između ulaznih promenljivih  $x$  i izlazne promenljive  $y$ , odnosno  $y = f(x)$ . Naučena veza (ciljna funkcija  $f$ ) se kasnije koristi za buduća predviđanja izlaza  $y$  na osnovu novih vrednosti ulaza  $x$ . Najčešće je ulaz predstavljen vektorom vrednosti promenljivih koje se nazivaju *atributima* (eng. *features*), a izlaz kao jedna promenljiva koja se zove *ciljna promenljiva* (eng. *target variable*). Kako se u današnje vreme raspolaže ogromnim količinama podataka merenim gigabajtima i terabajtima, neophodno je pronaći metode koje automatski pronalaze veze između promenljivih. Veze tj. izgrađene ciljne funkcije se nazivaju *modelima mašinskog učenja*. Postoji veliki broj modela i ne opisuju svi podjednako dobro veze među podacima. Od kvalitetnog modela se očekuje da vrši dobru generalizaciju tj. da prilikom predviđanja retko greši [12, 10].

Osnovne vrste nadgledanog učenja su *klasifikacija* i *regresija* [12].

- „Klasifikacija je problem predviđanja kategoričke ciljne promenljive.” Vrednosti kategoričke promenljive pripadaju nekom konačnom skupu, pri čemu ne postoji uređenje među tim vrednostima.
- „Regresija je problem predviđanja neprekidne ciljne promenljive”. Neprekidne promenljive uzimaju vrednosti iz neograničenog skupa vrednosti.

U nastavku će detaljnije biti opisani modeli klasifikacije zasnovani na algoritmima  $k$  najbližih suseda, slučajnim šumama i logističkoj regresiji.

## 2.1 Algoritam $k$ najbližih suseda

Klasifikatori zasnovani na algoritmu  $k$  najbližih suseda spadaju u klasifikatore zasnovane na instancama. Klasifikatori zasnovani na instancama spadaju u neparametarske modele. Takvi modeli ne mogu se opisati konačnim skupom parametara i oni moraju da čuvaju skup podataka za trening na osnovu koga se vrši klasifikacija novih instanci. Dakle, kod modela zasnovanih na instancama ne postoji eksplicitan model već je model sadržan u skupu trening instanci. Sva izračunavanja vrše se u fazi predviđanja. Ovi klasifikatori svoju predikciju zasnivaju na lokalnim podacima pa su stoga podložni greškama zbog postojanja šuma. Zavisni su od izbora mere bliskosti i adekvatnog preprocesiranja podataka. Ako vrednosti atributa imaju skale

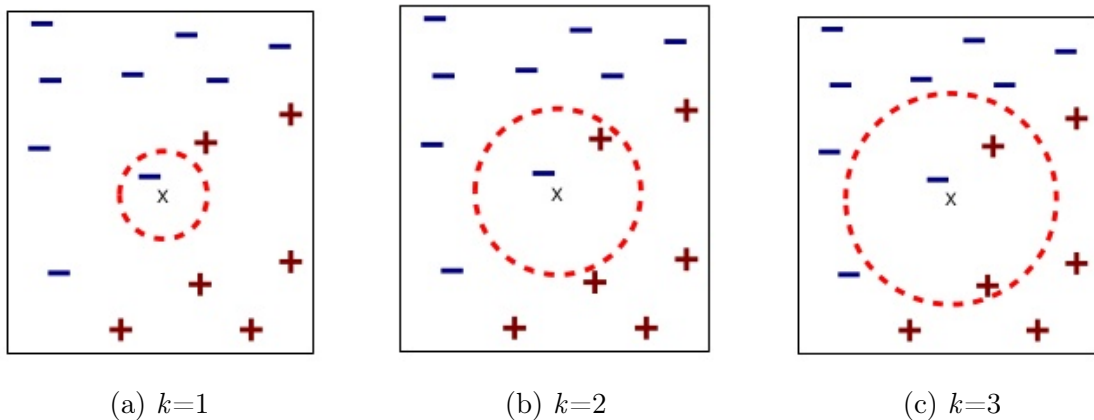


različitih vrednosti, jedan atribut može više uticati na ishod klasifikacije a da to nije opravdano. Ovo se može rešiti svođenjem atributa na istu skalu standardizacijom. Pored svoje jednostavnosti, ovi modeli nalaze široku primenu [12, 13].

Objekti podaci koje koristi metod najbližih suseda najčešće su predstavljeni kao tačke u  $d$ -dimenzionom prostoru, gde je  $d$  broj atributa objekata. Pretpostavlja se da postoji rastojanje među objektima kao i metrika koja definiše to rastojanje. Ideja vodilja primene metoda najbližih suseda je izreka : „Ako hoda kao patka, priča kao patka i izgleda kao patka, onda je najverovatnije patka” [12, 13].

### 2.1.1 Dodeljivanje klase objektu

Prilikom klasifikacije nepoznatog objekta prvo se među poznatim objektima pronade njegovih  $k$  najbližih suseda na osnovu izabrane mere bliskosti (često je u upotrebi *Euklidsko rastojanje*). Algoritam novom objektu pridružuje klasu koja se *najčešće javlja među njegovim susedima* (eng. *majority voting*). U slučaju nerešenog ishoda, klasa nepoznatog objekta se dobija slučajnim izborom iz skupa najzastupljenijih klasa. Da bi se izbegli nerešeni ishodi, česta je praksa da se za  $k$  uzima neparan broj [13, 12].



Slika 2.1: Susedstva i dodeljivanje klasa novoj instanci za  $k=1, 2$  i  $3$

Na slici 2.1 prikazana su susedstva instance koja se nalazi u sredini kruga kada algoritam  $k$  najbližih suseda uzima vrednosti 1, 2 i 3 za  $k$ . Instanci će biti dodeljena klasa na osnovu klasa njenih najbližih suseda. Na slici 2.1a klasa najbližeg suseda je (-) pa će i klasa instance biti (-). U slučaju 2.1c dva suseda su klase (+), a jedan klase (-) pa će klasa instance biti (+). Slika 2.1b ilustruje situaciju podjednake

zastupljenosti klasa (jedan (+) i jedan (-)). U ovakvim situacijama, instanci se dodeljuje klasa slučajnim izborom. [13]

Za rezultat klasifikacije jako je bitan izbor parametra  $k$ . U slučaju male vrednosti parametra  $k$  može doći do preprilagođavanja. Do grešaka dolazi jer je mali broj suseda uključen u razmatranje. Često se može javiti greška zbog prisustva šuma. Sa druge strane, velika vrednost parametra  $k$  vodi ka potprilagođavanju. U tom slučaju do greške može doći jer se razmatraju klase onih objekata koji nisu u neposrednom susedstvu [12, 13].

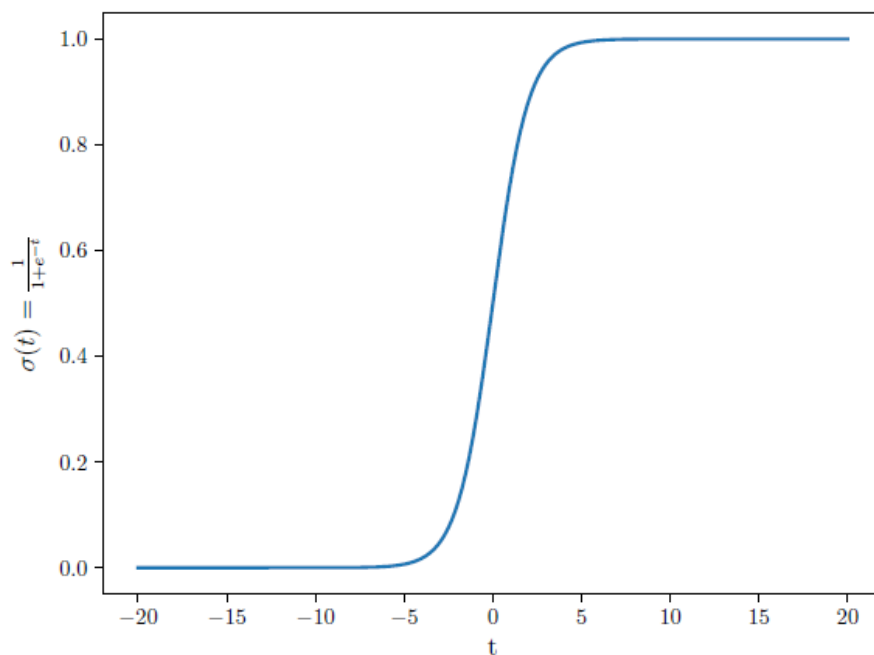
## 2.2 Logistička regresija

Logistička regresija predstavlja probabilistički model klasifikacije. Koristi se i za probleme binarne i višeklasne klasifikacije. Kod probabilističkih modela je potrebno definisati raspodelu verovatnoće koju kasnije treba oceniti na osnovu podataka. Da bi ocena raspodele bila računski izvodljiva potrebno je uvesti pretpostavke o raspodeli podataka i međusobnoj zavisnosti promenljivih. Pogrešne pretpostavke o podacima, njihovoj raspodeli i zavisnosti, vode ka lošijim rezultatima predviđanja. U daljem tekstu detaljnije je opisana binarna logistička regresija [12].

Logistička regresija pretpostavlja međusobnu nezavisnost vrednosti ciljne promenljive  $y$  i *Bernulijevu raspodelu* ciljne promenljive  $y$  za date vrednosti atributa  $x$ . To znači da postoji parametar  $\mu$  iz intervala  $[0,1]$  takav da je  $p(y=1|x) = \mu$ , a  $p(y=0|x) = 1-\mu$ . Ovako definisan model logističke regresije nije kompletan. Potrebno je definisati zavisnost parametra  $\mu$  od vrednosti atributa  $x$ . Da bi se parametar  $\mu$  uklopio u definiciju verovatnoće, njegove vrednosti moraju biti u intervalu  $[0,1]$ . Poželjno bi bilo koristiti model linearne regresije zbog njegove jednostavnosti i lake interpretabilnosti, ali na prvi pogled ovo nije moguće jer vrednosti linearnog modela pripadaju intervalu  $[-\infty, \infty]$ . Međutim, moguće je koristiti model linearne regresije ako se njegove vrednosti transformišu nekom nenegativnom, monotonom, neprekidnom i diferencijabilnom funkcijom u interval  $[0,1]$ . U tu svrhu koristi se *sigmoidna funkcija*  $\sigma$  (moguća je upotreba i nekih drugih funkcija):

$$\sigma(t) = \frac{1}{1 + e^{(-t)}}$$

Sigmoidna (logistička) funkcija, čiji je grafik prikazan na slici 2.2, uzima proizvoljan realan broj i dodeljuje mu vrednost iz intervala  $(0,1)$ .



Slika 2.2: Sigmoidna funkcija

Transformacijom vrednosti linearnog modela, definisana je i zavisnost parametra Bernulijeve raspodele  $\mu$  od vrednosti atributa  $x$ . Slično modelu linearne regresije, forma modela logističke regresije je jednačina:

$$p_w(y = 1|x) = \sigma(w \cdot x)$$

Izlaz modela je linearna kombinacija njegovog ulaza i koeficijenata. Koeficijenti se dobijaju iz podataka prilikom samog treniranja modela i oni predstavljaju reprezentaciju modela u memoriji [12, 11].

Treniranje modela linearne regresije odgovara oceni njegovih parametara i zasniva se na **principu maksimalne verodostojnosti**. Kao što smo videli, da bi se precizirao statistički model potrebno je uključiti neke parametre. Da bi dobijeni rezultat imao smisla, izbor vrednosti parametara je jako bitan. Jedan od principa izbora ovih vrednosti, princip maksimalne verodostojnosti, je neprihvatanje onih vrednosti parametara za koje su posmatrani podaci malo verovatni. Na taj način biće prihvaćene one vrednosti parametara za koje su posmatrani podaci visoko verovatni [12].

## 2.3 Slučajne šume

*Grupni metodi* (eng. *Ensemble methods*) su tehnike koje za cilj imaju poboljšanje tačnosti klasifikacije koje se postiže kombinovanjem predviđanja većeg broja klasifikatora. Ovi metodi konstruišu veći broj *baznih klasifikatora* (eng. *base classifiers*) i kombinovanjem njihovih rezultata vrše svoje predviđanje. Rezultat predviđanja je srednja vrednost u slučaju regresije ili najzastupljenija vrednost u slučaju klasifikacije. Rezultati dobijeni na ovaj način su bolji od rezultata dobijenih korišćenjem samo jednog klasifikatora.

Da bi grupni klasifikatori zaista davali bolje rezultate od pojedinačnih klasifikatora, potrebno je da budu ispunjena dva uslova [13]:

- 1) bazni klasifikatori treba da budu međusobno nezavisni. U tom slučaju konačna predikcija će biti pogrešna samo ako više od polovine baznih klasifikatora pogreši u predikciji. Potpuna nezavisnost klasifikatora je teško ostvariva, ali se u praksi pokazalo da nije neophodna da bi se ostvarili bolji rezultati.
- 2) bazni klasifikatori treba da budu bolji od slučajnog klasifikatora

Osnovna ideja grupnih metoda je kreiranje velikog broja klasifikatora na osnovu trening podataka i neki vid agregacije njihovih rezultata u slučaju nepoznate instance. Grupa klasifikatora može se konstruisati na neki od sledećih načina [13]:

- 1) *Manipulacijom trening podataka* Ponovnim izborom, u skladu sa izabranom raspodelom, iz originalnog skupa trening podataka kreira se veći broj novih skupova podataka. Raspodela utiče na verovatnoću da će podatak biti izabran za novi trening skup. Zatim se za svaki novi trening skup kreira model klasifikacije izabranim algoritmom (npr. stablo odlučivanja). Dva grupna metoda koja rade na ovaj način su upakivanje i podsticanje.

- *Upakivanje* (eng. *bagging*) je tehnika koja iznova vrši izbor uzoraka sa zamenom na osnovu uniformne raspodele verovatnoća. Kako sve instance imaju jednaku verovatnoću da budu izabrane, ova tehnika je manje podložna greškama usled preprilagođavanja. Dobijeni skup uzoraka je iste veličine kao i originalni skup. Kako se izbor uzoraka vrši sa zamenom neki podaci će biti izabrani više puta a neki nijednom. Na dobijenim skupovima se treniraju

klasifikatori. Nakon što se kreiraju svi klasifikatori moguće je klasifikovati nepoznate instance. Novoj instanci će biti pridružena najzastupljenija klasa.

- *Podsticanje* (eng. *boosting*) je iterativna procedura koja postepeno menja raspodelu trening podataka i na taj način favorizuje one podatke koji su teži za klasifikaciju. Svakoj trening instanci je pridružen koeficijent težine koji se može promeniti po završetku iteracije. Težine se mogu koristiti bilo kao raspodela pri kreiranju novih skupova podataka bilo za treniranje pristrasnih modela.

- 2) Manipulacijom ulaznih atributa Novi skupovi trening podataka nastaju kao podskupovi originalnih podataka, slučajnim izborom podataka početnog skupa ili analizom domena. Ovaj pristup daje dobre rezultate u slučaju redundantnih atributa. Slučajne šume su primer metoda koji manipuliše ulaznim atributima i koristi klasifikatore zasnovane na stablima odlučivanja u svojoj osnovi.
- 3) Manipulacijom algoritma učenja Moguće je menjati i sam algoritam učenja. Ovaj način nalazi praktičnu primenu ubacivanjem faktora slučajnosti prilikom treniranja skupa stabala odlučivanja. Umesto izbora najboljeg atributa za podelu u čvoru, moguće je slučajnim izborom odabrati jedan od nekoliko najboljih atributa podele.
- 4) Manipulacijom oznaka klasa Ako je broj klasa dovoljno, veliki višeklasni problem je moguće transformisati u binarni problem podelom oznaka klasa, na slučajan način, u dva disjunktne skupa  $A_0$  i  $A_1$ . Trening podacima čije klase pripadaju skupu  $A_0$  pridružuje se klasa 0, a podacima čije klase pripadaju skupu  $A_1$  klasa 1. Podaci sa novim oznakama klasa se koriste za treniranje baznog klasifikatora. Grupa klasifikatora se dobija višestrukim ponavljanjem kreiranja skupova  $A_0$  i  $A_1$  i treniranja klasifikatora nad dobijenim skupovima. Dodeljivanje klase nepoznatoj instanci vrši se tako što svaki bazni klasifikator  $C_i$  vrši predviđanje. Ako je predviđena klasa 0, sve klase skupa  $A_0$  dobijaju glas. Analogno, ako je predviđena klasa 1, sve klase skupa  $A_1$  dobijaju glas. Glasovi se prebrojavaju i rezultujuća klasa će biti ona sa najviše glasova.

*Slučajne šume* (eng. *Random forests*) pripadaju klasi grupnih metoda. Kao bazni klasifikatori se koriste stabla odlučivanja koja se konstruišu nad skupom nezavisnih slučajnih vektora koji nastaju metodom manipulacije trening podataka.

*Upakivanje sa korišćenjem stabala odlučivanja* (eng. *Bagging using decision trees*) je specijalan slučaj slučajnih šuma u kom se slučajnost dodaje u proces pravljenja modela (do sada je postojala u trening podacima) tako što se novi skupovi podataka kreiraju metodom slučajnog izbora sa zamenom od elemenata polaznog skupa. Dodavanje slučajnosti smanjuje korelaciju između stabala, a samim tim i grešku generalizacije cele grupe. Upakivanje popravljja grešku generalizacije smanjujuću varijansu baznih klasifikatora. Pored toga, upakivanje je manje podložno i greškama usled preprilagođavanja, koje je velika mana stabala odlučivanja, zato što nijedna trening instanca nema prednost prilikom izbora, već sve imaju podjednaku verovatnoću da će biti izabrane [13].

Svako stablo odlučivanja za podelu u čvoru koristi odgovarajući slučajni vektor podataka koji se dobija na jedan od sledećih načina [13]:

- 1) Slučajnim izborom se određuje  $F$  atributa koji će se koristiti za podelu u svakom od čvorova (umesto da se za podelu razmatraju svi atributi) nakon čega se stablo konstruiše do kraja bez odsecanja. Ako postoji potreba za većim faktorom slučajnosti moguće je koristiti upakivanje za kreiranje trening skupova. Izbor broja atributa  $F$  utiče na snagu i korelaciju modela slučajnih šuma. Za male vrednosti  $F$  stabla šume su slabo korelisana, dok veliko  $F$  omogućava jače modele zbog korišćenja većeg broja atributa. Često se kao ravnoteža bira  $F = \log_2 d + 1$ , gde je  $d$  broj ulaznih atributa. Ovaj pristup vodi manjem vremenu izvršavanja algoritma jer se prilikom podele u čvoru ne razmatraju svi atributi.
- 2) Ako početnih atributa  $d$  ima malo, teško je pronaći nezavisni skup slučajnih atributa koji se koriste pri izgradnji modela. Ovo se može prevazići proširenjem skupa atributa njihovom linearnom kombinacijom. Na nivou svakog čvora novi atribut nastaje tako što se metodom slučajnog izbora izabere  $L$  atributa iz polaznog skupa, nakon čega se ti atributi linearno kombinuju sa koeficijentima dobijenim iz uniformne raspodele na intervalu  $[-1,1]$ . Svaki čvor će dobiti  $F$  novih atributa, a najbolji od njih će biti korišćen za podelu čvora.
- 3) Mogući pristup podeli unutar čvora je da se umesto najboljeg atributa podele na slučajan način izabere jedan od  $F$  najboljih atributa. Stabla dobijena na ovaj način mogu imati veći stepen korelacije u slučaju nedovoljno velikog parametra  $F$ . Pored toga, ovaj pristup ne daje uštedu u vremenu izvršavanja kao prethodna dva pristupa jer je potrebno ispitati svaki atribut podele u svakom čvoru stabla.

## 2.4 Mašinsko učenje u Python-u

„Python je interpretirani, objektno-orijentisani viši programski jezik sa dinamičkom semantikom. Pythonova jednostavna sintaksa se lako uči i naglašava čitljivost koda koja za posledicu ima nisku cenu održavanja.” Python podržava veliki broj paketa i modula što vodi ponovnoj upotrebljivosti i modularnosti koda. Nepostojanje faze kompilacije ubrzava čitav proces pisanja i testiranja koda. Debugovanje je prilično jednostavno, greške proizvode izuzetke i ako se oni ne uhvate interpreter ispisuje *stanje steka* (eng. *stack trace*). Štaviše, često je najjednostavniji način debugovanja ubacivanje `print` naredbi [9].

Autor Pythona je Holanđanin Gido Van Rosum (hol. Guido Van Rossum). Ideju o Pythonu Gido dobija krajem 1980-ih kada je radio kao programer na Državnom institutu za matematiku i informatiku CWI (Centrum voor Wiskunde en Informatica) u jeziku ABC kojim je inspirisan. Implementaciju započinje decembra 1989-te, a prva verzija jezika (verzija 0.9.0) puštena je 1991-ve. Dalji razvoj Pythona se nastavlja. Januara 1994-te izlazi verzija 1.0, unapređena verzija 2.0 izlazi u oktobru 2000-te, a verzija 3.0 u decembru 2008-me [2, 1].

### 2.4.1 SciPy ekosistem

**SciPy** je ekosistem slobodnog softvera za matematiku, nauku i inženjerstvo. Njegovu osnovu čini programski jezik Python i grupa paketa od kojih su za mašinsko učenje najznačajniji [6]:

1) **NumPy** je osnovni paket za naučna izračunavanja u Pythonu. Ovaj paket definiše numeričke nizove, matrične tipove, kao i osnovne operacije nad njima. Glavni objekat paketa je homogeni višedimenzioni niz koji je predstavljen kao tabela elemenata (najčešće brojeva), pri čemu su svi istog tipa i indeksirani torkom pozitivnih celih brojeva. Dimenzije niza se nazivaju *osama* (eng. *axis*). Klasa nizova NumPy paketa se naziva *ndarray*.

- Tačka prostora npr. `[1, 2, 1]` sa svoje tri koordinate ima jednu osu (dimenziju). Osa se sastoji od 3 elementa pa se kaže da je dužine 3.
- Dvodimenzioni niz, npr.

`[[1., 0., 0.],`

[0., 1., 2.]

ima dve ose. Prva osa ima 2 elementa, a druga 3 [8, 7].

- 2) **Pandas** je paket koji obezbeđuje strukture podataka za jednostavan i intuitivan rad sa podacima. Predstavlja nadgradnju NumPy biblioteke i napravljen je tako da se lako integriše u razna okruženja koja se koriste za naučna izračunavanja. Pogodan je za podatke koji su smešteni u tabelama, uređene i neuređene vremenske serije, proizvoljne matrice i sve oblike podataka dobijenih na osnovu statističkih posmatranja. Pandas uvodi dve strukture podataka, Series za jednodimenzione podatke i DataFrame za dvodimenzione [4].
- 3) **Matplotlib** je biblioteka koja služi za kreiranje različitih vrsta dijagrama. Većinu dijagrama moguće je kreirati u svega nekoliko linija koda, a sami dijagrami se mogu prilagoditi potrebama korisnika izborom stilova linija (pune, isprekidane,...), svojstava fontova, svojstava koordinatnih osa, itd [3].

**Scikit-learn** je modul koji implementira veliki broj algoritama mašinskog učenja. Pored algoritama za rešavanje problema klasifikacije, regresije i klasterovanja ovaj modul podržava i razne tehnike preprocesiranja podataka, kao i tehnike ocenjivanja modela. Zasniva se na bibliotekama NumPy, SciPy i matplotlib [5].



## Glava 3

# Verifikacija softvera

Glava 4

Zaključak

# Bibliografija

- [1] A brief timeline of python. <https://python-history.blogspot.com/2009/01/brief-timeline-of-python.html>.
- [2] History of python. [https://www.python-course.eu/python3\\_history\\_and\\_philosophy.php](https://www.python-course.eu/python3_history_and_philosophy.php).
- [3] Matplotlib.org. <https://matplotlib.org/>.
- [4] pandas: powerful Python data analysis toolkit. <https://pandas.pydata.org/pandas-docs/stable/>.
- [5] scikit-learn: Machine Learning in Python. <http://scikit-learn.org/stable/index.html>.
- [6] Scipy.org. <https://www.scipy.org/>.
- [7] Scipy.org: Quickstart tutorial. <https://docs.scipy.org/doc/numpy/user/quickstart.html>.
- [8] Scipy.org: Scientific computing tools for python. <https://www.scipy.org/about.html>.
- [9] What is python? executive summary. <https://www.python.org/doc/essays/blurb/>.
- [10] Jason Brownlee. How machine learning algorithms work (they learn a mapping of input to output). <https://machinelearningmastery.com/how-machine-learning-algorithms-work/>.
- [11] Jason Brownlee. Logistic regression for machine learning. <https://machinelearningmastery.com/logistic-regression-for-machine-learning/>.

## *BIBLIOGRAFIJA*

---

- [12] Mladen Nikolić and Anđelka Zečević. Mašinsko učenje. 2018.
- [13] Pang-Ning Tan, Michael Steinbach, and Vipin Kumar. *Introduction to Data Mining*. Addison-Wesley Longman Publishing Co., Inc., 2005.

## Biografija autora