

Intelligentni sistemi

INDUKCIONI SISTEMI

Šta je to mašinsko učenje?

Disciplina koja omogućava računarima da uče bez *eksplicitnog* programiranja (Arthur Samuel 1959).

1. Generalizacija znanja na osnovu prethodnog **iskustva** (podataka o pojavama/entitetima koji su predmet učenja)

2. Dobijeno znanje koristi se kako bi se dali odgovori na pitanja za entitete/pojave koji nisu ranije viđeni

Definicija (Tom Mitchell 1998)

Za kompjuterski program se kaže da uči iz iskustva ***E (experience)***, vezanog za zadatak ***T (task)***, i meru performansi ***P (performance)***, ukoliko se njegove performanse na zadatku ***T***, *merene* metrikama ***P***, *unapređuju sa iskustvom* ***E***

Prednosti ML-a?

1. Vrlo je teško precizno (algoritamski) opisati neke vrste zadataka koje ljudi lako rešavaju. Primeri: prepoznavanje lica (face detection), prepoznavanje govora (speech recognition)
2. Za neke vrste zadataka mogu se definisati algoritmi za rešavanje, ali su ti algoritmi vrlo složeni i/ili zahtevaju velike baze znanja
Primer: automatsko prevođenje (MT)

Prednosti ML-a?

3. U mnogim oblastima se kontinuirano prikupljaju podaci sa ciljem da se iz njih “nešto sazna”; npr:
 - u medicini: podaci o pacijentima i terapijama
 - u marketingu: o korisnicima/kupcima i tome šta su kupili, za šta su se interesovali, kako su proizvode ocenili,...
- Analiza podataka ovog tipa zahteva pristupe koji će omogućiti da se otkriju pravilnosti, zakonitosti u podacima koje nisu ni poznate, ni očigledne, a mogu biti korisne (Data mining)

Najnoviji rezultati 12/2022

- **GoalsEye:** Learning High Speed Precision Table Tennis on a Physical Robot
 - <https://sites.google.com/view/goals-eye?pli=1>
- **AlphaCode:** Competitive programming with AlphaCode
 - <https://www.deepmind.com/blog/competitive-programming-with-alphacode?s=09>
 - <https://www.science.org/doi/10.1126/science.abq1158>
 - As part of DeepMind's mission to solve intelligence, we created a system called AlphaCode that writes computer programs at a competitive level. AlphaCode achieved an estimated rank within the top 54% of participants in programming competitions by solving new problems that require a combination of critical thinking, logic, algorithms, coding, and natural language understanding.
- **ChatGPT:** ChatGPT: Optimizing Language Models for Dialogue
 - <https://openai.com/blog/chatgpt/?s=09>
 - We've trained a model called ChatGPT which interacts in a conversational way. The dialogue format makes it possible for ChatGPT to answer followup questions, admit its mistakes, challenge incorrect premises, and reject inappropriate requests.
- **Diplomacy:** Human-level play in the game of Diplomacy by combining language models with strategic reasoning
 - <https://www.science.org/doi/10.1126/science.ade9097?s=09>

Podela

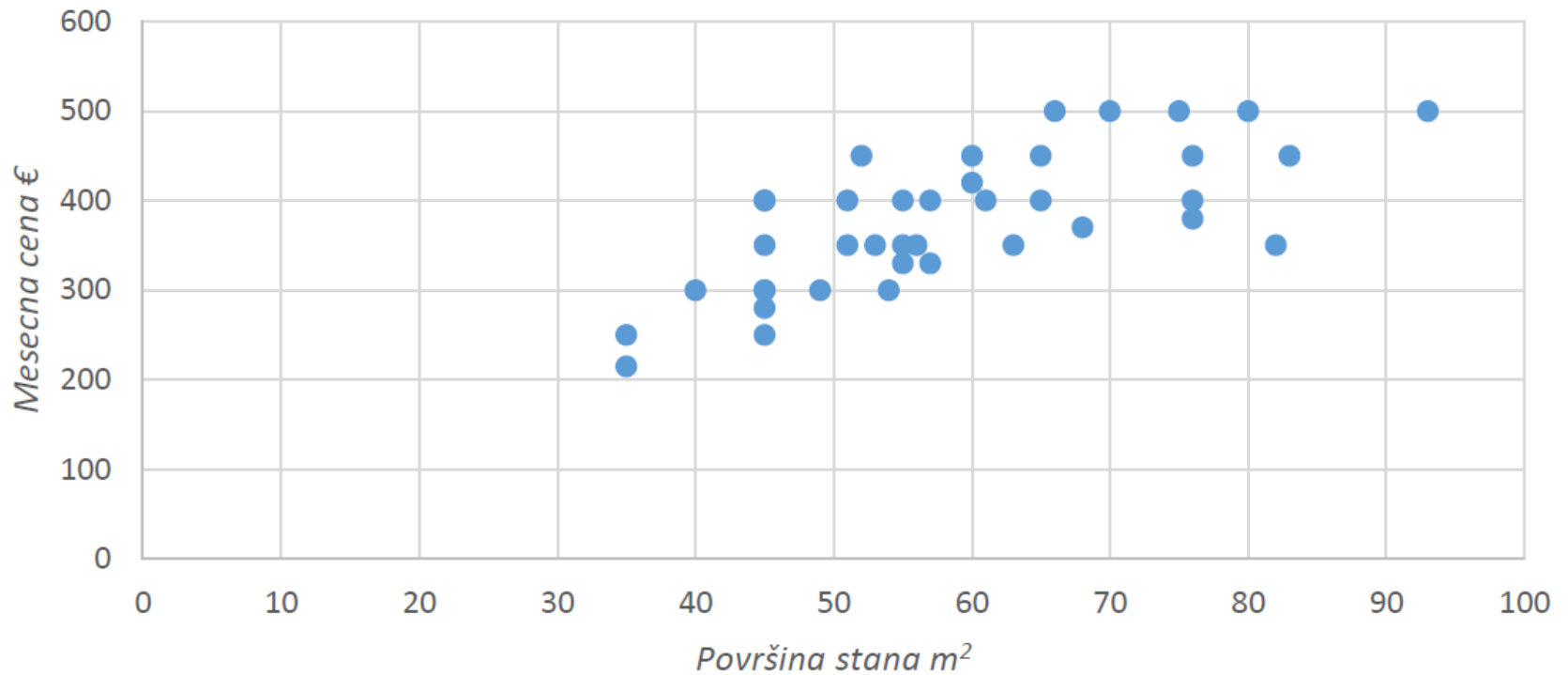
- Postoji nekoliko različitih tipova algoritama u mašinskom učenju. Dva najčešća i najosnovnija oblika mašinskog učenja su:
- Supervised learning (nadgledano učenje)
- Unsupervised learning

Nadgledano učenje

- Supervised learning (nadgledano učenje) predstavlja oblik mašinskog učenja na osnovu obeleženog skupa podataka za treniranje.
- Obeležen primer iz skupa podataka za treniranje, sastoji se od ulaznih podataka, tipično predstavljenih nizom realnih brojeva, kao i željene očekivane izlazne vrednosti.
- Ovi algoritmi, jednom trenirani sa obeleženim skupom podataka, kao svrhu imaju da izračunaju rezultat, za neobeleženi skup podataka
- Izlazna vrednost može biti kontinualan realan broj ili neka diskretna vrednost.

Primer

Cena iznajmljivanja stana na osnovu kvadrature



Nagledano učenje

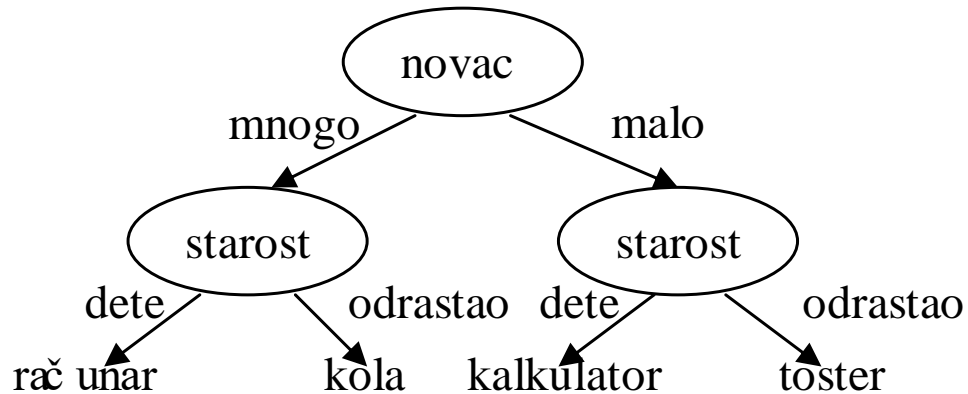
- Ukoliko je izlazna vrednost diskretnog tipa, tada se govori o klasifikaciji
- Neki od primera klasifikacija su:
 - Medicinsko testiranje radi utvrđivanja da li pacijent ima bolest ili ne
 - Prošao/pao metod testiranja kod provere kvaliteta u fabrikama
 - Testiranje krvne grupe koje pokazuje da li pacijent ima A, B, AB, ili O
 - Određivanje da li je dobijen email, spam ili ne

Indukcioni sistemi

- Učenje putem primera – indukcija - proces izvođenja opštih pravila iz znanja koja se sadrže u konačnom skupu primera
- Induktivno učenje se može posmatrati kao pretraživanje prostora problema radi nalaženja rešenja.
- Primer - problem nalaženja poklona. Neka su glavni koncepti problema: novac, starost osobe, pokloni.
- Usvajaju se sledeće pretpostavke
 - Novac – mnogo ili malo
 - Starost – dete ili odrasla osoba

Indukcioni sistemi

Faktori odlučivanja		Rezultati
Novac	Starost	Poklon
mnogo	odrastao	Kola
mnogo	dete	Računar
malo	odrastao	Toster
malo	dete	Kalkulator



Indukcioni sistemi

- IF mnogo novca

AND kupuje se poklon za odraslu osobu

THEN kupiti kola

- IF mnogo novca

AND kupuje se poklon za dete

THEN kupiti računar

- IF malo novca

AND kupuje se poklon za odraslu osobu

THEN kupiti toster

- IF malo novca

AND kupuje se poklon za dete

THEN kupiti kalkulator

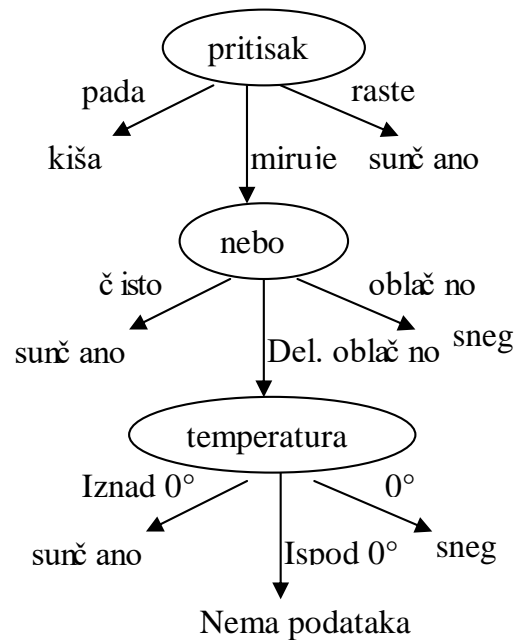
ID3

- ID3 (Iterative Dichotomiser 3 – Ross Quinlan) je induktivni algoritam koji se komercijalno koristio
- ID3 uzima skup primera problema i generiše stablo odlučivanja
- Izgradjuje rešenje sa što manje pretpostavki, odnosno izgraditi što manje stablo
- Primer je kombinacija:
 - faktora odlučivanja,
 - vrednosti faktora odlučivanja,
 - akcije specifične za taj primer.

ID3

Faktori odlučivanja				Rezultat
Temperatura	Vetar	Nebo	Pritisak	Prognoza
Iznad 0°	Zapadni	Oblačno	Pada	Kiša
Ispod 0°	*	Oblačno	Miruje	Sneg
Iznad 0°	Istočni	Oblačno	Raste	Sunčano
Iznad 0°	*	Delimično oblačno	Miruje	Sunčano
*	*	Čisto nebo	Miruje	Sunčano
Iznad 0°	Južni	Čisto nebo	Pada	Kiša
0°	Severni	Delimično oblačno	Miruje	Sneg

* - označava da taj faktor u tom primeru nije bitan.



ID3

- Izbor najpre najvažnijeg elementa – prva odluka je izbor atributa barometarski pritisak kao korena stabla. Ispitivanje prvo ovog faktora je najefikasnije, pošto dva od tri moguća odgovora vode neposrednom predviđanju, bez potrebe da se ispituju faktori nebo ili temperatura. Dakle, ID3 smešta najvažnije faktore odlučivanja blizu korena stabla.
- Nema podataka – povremeno se dešavaju slučajevi da se ne može dobiti rezultat. To označava situaciju za koju ne postoje primeri na osnovu kojih se može izvući zaključak. Ako se naiđe na ovakve situacije, to je obično znak da je skup primera nedovoljan i da ga je potrebno upotpuniti novim primerima koji bi podržali te mogućnosti.
- Isključivanje nevažnih faktora – još jedna važna odlika ID3 je da ustanovi da originalni skup sadrži atribut smer vetra, ali ovaj faktor se ne pojavljuje nigde u stablu odlučivanja. ID3 je odlučio, na osnovu skupa primera, da je ovaj faktor nebitan za predviđanje vremena. Čovek koji je ekspert za predviđanje vremena može da bude nesvestan da je ovaj faktor nevažan.

Algoritam rada ID3

- ID3 je naslednik sistema za učenje koncepata (CLS – Concept Learning System).
- CLS rešava zadatke učenja sa jednim konceptom i koristi naučene koncepte da klasifikuje nove primere.
- CLS može da otkrije klasifikaciono pravilo ili stablo odlučivanja za skup primera koji pripadaju dvema klasama i da koristi ovo pravilo da klasifikuje novi primer u jednu od ove dve klase.
- Primer se klasifikuje polazeći od čvora stabla, obavljajući testiranje, i prateći grane dok se ne dostigne čvor, koji se označava sa YES ili NO, da li je primer u klasi.

CLS

- CLS algoritam počinje praznim stablom odlučivanja i iterativno gradi stablo dodavanjem čvorova odlučivanja dok stablo ne klasifikuje ispravno sve primere obučavanja u skupu C.
- Ovaj algoritam se odvija na sledeći način:
 - Ako su svi primeri u C pozitivni, formirati čvor YES i stati. Ako su svi primeri u C negativni, formirati čvor NE i stati. Inače, odabrati, prema nekom heurističkom kriterijumu, atribut A sa vrednostima V_1, V_2, \dots, V_n , i formirati čvor odlučivanja
 - Podeliti priemre za obučavanje u skupu C u podskupove prema vrednostima C_1, C_2, \dots, C_n , za V
 - Primeniti algoritam rekurzivno na svaki podskup C_i

ID3 algoritam

- CLS algoritam zahteva da svi primeri za obučavanje budu raspoloživi u prvom koraku, što ograničava broj primera koji se mogu efektivno rešiti. ID3 algoritam radi sa podskupovima primera radi rešavanja složenih problema koji uključuju veliki broj primera.
 - Heuristika korišćena u prvom koraku služi za izbor najpre najdiskriminatornijeg atributa. ID3 koristi informacioni pristup za procenu diskriminatorske snage svakog atributa.
 - Za neki dati primer, stablo odlučivanja generiše poruku vezanu za klasu primera, t.j. ako je primer vezan za klasu YES, daje se poruka »plus«. Ako su verovatnoće »plus« i »minus« poruka p^+ i p^- respektivno, tada je informacioni sadržaj
- $p^+ \log_2 p^+ - p^- \log_2 p^-$ (1)

ID3 algoritam

- p^+ je dato proporcijom primera u C klase »plus« a p^- dato proporcijom primera u klase »minus«.
- $M(C)$ se sada može definisati da predstavlja očekivani informacioni sadržaj poruke iz stabla odlučivanja za skup C primera, i dat je jednačinom (1).
- Izbor najdiskriminativnijeg atribut - atribut kojim treba testirati susedne grane podskupove C_i od C.
- Za bilo koji atribut sa vrednostima V_1, V_2, \dots, V_n , novočekivana informacija koja treba da se dobije biranjem ovog atributa je data sa

$$B(C, A) = \sum(\text{verovatnoća da je vrednost od A jednaka } V_i) \quad (2)$$

- Jednačina (2) se može primeniti na svaki atribut i odabira se onaj koji maksimizira očekivani informativni dobitak prema

$$M(C) - B(C, A) \quad (3)$$

- Novoočekivana količina informacija treba da bude mala kada se bira ovaj atribut A da bi preostali deo imao što manje da radi.

ID3 algoritam

1. Odabrati slučajni podskup veličine W iz celokupnog skupa primera za obučavanje (W se naziva prozor)
2. Primeniti CLS algoritam za formiranje stabla odlučivanja ili pravila za odabrani prozor W .
3. Pretražiti celokupan skup primera (ne samo prozor) da bi se pronašli izuzeci od tekućeg pravila.
4. Ako postoje izuzeci, uključiti ih u prozor i ponoviti korak 2, inače stati i prikazati poslednje pravilo.

ID3 algoritam

- Algoritam konvergira iterativno konačnom pravilu koje obuhvata koncept. Svrha koraka 1 je da se formira mali skup iz svih primera, prozor, jer pun skup primera može da bude suviše veliki. Ovim se korakom povećava efikasnost algoritma.
- CLS algoritam služi kao potprogram ID3 algoritmu. CLS algoritam prvo nalazi atribut (čvor odlučivanja) koji najbolje diskriminiše između pozitivnih i negativnih primera, i deli podatke u odnosu na taj atribut. ID3 algoritam koristi heuristiku zasnovanu na informacionom sadržaju poruke dobijene od stabla odlučivanja za dati primer.
- Po izboru najdiskriminativnijeg atributa, podaci se dele u dva podskupa. Svaki podskup se zatim deli na sličan način sve dok se ne dobiju podskupovi koji sadrže podatke samo jedne klase. Konačan rezultat je stablo odlučivanja koje se može iskoristiti da se klasifikuju novi primeri.

ID3 - primer

Faktori odlučivanja				Rezultat
	Nebo	Pritisak	Vetar	Kiša
1	Č	R	S	-
2	O	R	J	+
3	O	S	S	+
4	Č	O	S	-
5	O	O	S	+
6	O	R	S	+
7	O	O	J	-
8	Č	R	J	-

Nebo

Čisto

- 1. čisto, raste, severni -
- 4. čisto, pada, severni -
- 8. čisto, raste, južni -

Oblačno

- 2. oblačno, raste, južni +
- 3. oblačno, stabilno, severni +
- 5. oblačno, pada, severni +
- 6. oblačno, raste, severni +
- 7. oblačno, pada, južni -

ID3 - primer

Podskup za granu č isto sadrži samo primer jedne klase i ne zahteva dalju razradu. Razvijajuć i dalje atribut pritisak daje sledeće rezultate

oblač no
pritisak

pada

stabilno

raste

5. oblač no, pada, severni + 3. oblač no, stabilno, severni + 2. oblač no, raste, juž ni +

7. oblač no, pada, juž ni - 6. oblač no, raste, severni +

Dalje, potrebno je izvršiti dalju ekspanziju za podskup pritisak pada.

pada
vetar

severni

juž ni

5. oblač no, pada, severni +

7. oblač no, pada, juž ni -

ID3 - primer

Ostaje da se preveri da li je najbolji diskriminatorskič vor nebo.

Skup primera C sadži i 4 u klasi + i 4 u klasi -. Prema jednoj od prethodnih jednač ina

$$M(C) = -(4/8)\log_2(4/8) - (4/8)\log_2(4/8) = 1 \text{ bit}$$

Ako se prvo testira atribut vetar dobija se jednonivoosko stablo:

		vetar	
severni		juž ni	
1	-	2	+
3	+	7	-
5	+	8	-
6	+		

ID3 - primer

- Informacija potrebna za granu severni može se pronaći iz

$$M(\text{severni}) = -(3/5)\log_2(3/5) - (2/5)\log_2(2/5) = 0,971 \text{ bita}$$

- a za granu južni

$$M(\text{južni}) = -(1/3)\log_2(1/3) - (2/3)\log_2(2/3) = 0,918 \text{ bita}$$

- Očekivani informacioni sadržaj je

$$B(C, \text{vetar}) = (5/8) * 0,971 + (3/8) * 0,918 = 0,951 \text{ bita}$$

- Informacija koja se dobija korišćenjem ovog algoritma

$$M(C) - B(C, \text{vetar}) = 1 - 0,951 = 0,049 \text{ bita}$$

- Ako se isti postupak primeni na atribut nebo dobija se

$$M(\text{čisto}) = -(3/3)\log_2(3/3) = 0 \text{ bita}$$

$$M(\text{oblačno}) = -(4/5)\log_2(4/5) - (1/5)\log_2(1/5) = 0,722 \text{ bita}$$

$$B(C, \text{nebo}) = (3/8) * 0 + (5/8) * 0,722 = 0,4512 \text{ bita}$$

- Informacija koja se dobija koristeći atribut nebo je

$$M(C) - B(C, \text{nebo}) = 1 - 0,452 = 0,548 \text{ bita}$$

- Ako se isti postupak primeni na atribut pritisak, dobija se 0,156 bita.

- Znači da je najpogodniji diskriminatorski čvor nebo.

Prednosti ID3

- **Biranje skupa problema** – I CLS i ID3 algoritmi rade dobro kada se radi o problemima sa samo jednim konceptom. Izbor podskupa problema za obučavanje, prozora, jedna od osnovnih snaga. Ovaj metod izbora prozora se često naziva filtriranje upravljano izuzecima. Ova tehnika omogućava programu da se fokusira na one primere za obučavanje koje su protivne očekivanjima, koji su primeri potrebni da se poboljša predstava koncepta koji se uči.
- **Izbor najboljeg diskriminatorskog čvora** – ID3 primenjuje informacijski pristup u izboru najboljeg diskriminatorskog čvora. Ovaj pristup obezbeđuje efikasnost sistema.

Nedostaci ID3

- pravila nisu probabilistička
- nekoliko identičnih primera nema više efekta od jednog
- ne radi sa protivrečnim primerima
- rezultati su stoga jako osetljivi na manje izmene u primerima za obučavanje.
- Praktičan problem sa stablima odlučivanja je da je teško razumeti koncept koji se uči za velika stabla odlučivanja. Uprkos tome, ID3 se pokazao efektivnim u izgradnji dobrih ekspertskih sistema.

Razvoj induktivnog ekspertskog sistema

- **Odrediti cilj:** Pretraživanje stabla odlučivanja će dovesti do jedne od konačnog skupa odluka na osnovu putanje kroz stablo. Svaka odluka se odnosi za neki unapred određeni cilj. U problemu izbora poklona cilj je bio da se izabere pravi poklon za određenu osobu.
- **Odrediti faktore odlučivanja:** faktori odlučivanja su čvorovi atributa u stablu odlučivanja. Ovi faktori uključuju odlike problema koji bi se razmatrali kod donošenja konačne odluke. Na primer, u problemu izbora poklona, uzeti su u obzir kao što su novac i starost i pol osobe za koju se kupuje poklon. Ovo je često najteži zadatak u razvoju induktivnog sistema. Ako su faktori odlučivanja slabo izabrani ili nepotpuni, krajnji rezultat bi bio netačan.
- **Odrediti vrednosti faktora odlučivanja:** Potrebno je zadati listu mogućih vrednosti za attribute stabla odlučivanja. Na primer, faktor odlučivanja novac ima vrednost (malo, mnogo). Dobar izbor vrednosti je važan. Ako se posmatra navedeni slučaj, korisnik može da utvrdi da je ovo suviše subjektivno, ili pak želi da potroši prosečno novca – vrednost koja nedostaje u listi. Stoga se korisnik mora uključiti rano u projekat da bi pomogao u definisanju efektivnog skupa vrednosti.

Razvoj induktivnog ekspertskog sistema

- **Odrediti rešenje:** Mora se odrediti lista konačnih odluka koju sistem mora da učini - listovi stabla odlučivanja. Na primer, mogući pokloni.
- **Formiranje skupa primera:** Primeri sadrže poznavanje problema i koriste se za donošenje inteligentnih odluka. Ovi primeri povezuju vrednosti faktora odlučivanja sa konačnim ishodom, i mogu da se dobiju od eksperta domena ili iz zapisa prošlih događaja. Ako primeri dolaze iz nekoliko izvora, potrebno je čuvati zapis o njihovom izvoru. U slučaju da performanse sistema ne zadovolje očekivanja, tada brisanje primera iz manje pouzdanog izvora bi moglo da poboljša situaciju.
- **Formiranje stabla odlučivanja:** Potrebno je da se koriste induktivni algoritmi kao što je ID3 da bi se formiralo stablo odlučivanja. U slučaju da se neki faktor odlučivanja ne pojavi u stablu, ne uklanjati ga iz skupa podataka. Kasnije kada se dodaju drugi primeri, faktor se može pojaviti kada se indukuje novo stablo.

Razvoj induktivnog ekspertskog sistema

- **Testirati stablo:** Pri testiranju se upoređuju predviđeni rezultati sistema sa stvarnim studijama slučaja. Mogu se izabrati slučajevi iz skupa originalnih primera koji nisu bili korišćeni pri formiranju stabla odlučivanja ili koristiti skup budućih primera.
- **Izvršiti reviziju sistema:** Revizija sistema se obično sastoji od dodavanja novih faktora odlučivanja ili dodavanja ili brisanja primera. Tipični problemi koji se dešavaju u toku testiranja spadaju u neku od sledećih kategorija: netačan rezultat, nema rezultata, protivrečan rezultat.
 - Netačan rezultat je onaj koji se razlikuje od očekivanog. Ovaj problem se dešava jer je poznavanje sistema slabo, sistem nema dovoljno faktora odlučivanja, ili sistemu nedostaju vrednosti faktora odlučivanja. Sistem je slab ako je izvor znanja nepouzdan ili skup primera ne pokriva širok opseg problema. Problemi se mogu pojaviti ako sistemu nedostaju važni faktori odlučivanja ili vrednosti.
 - Kada nema rezultat, znači da skup primera ne uključuje vrednosti faktora odlučivanja za date test slučajeve - uključiti nove primere.
 - Protivrečan rezultat dolazi kada se dobije više od jednog rezultata za isti skup vrednosti faktora odlučivanja. Radi korekcije, treba dodati nove faktore ili vrednosti koje će dovesti do odgovarajuće diskriminacije.

Primer razvoja sistema

Cilj: napraviti ekspertski sistem koji će predvideti pobjedu ili poraz našeg tima u sledećem susretu.

Faktori odlučivanja: lokacija meča, vreme, istorija našeg tima i istorija protivnika.

Vrednosti faktora odlučivanja:

Lokacija	Vreme	Istorija tima	Istorija protivnika
Kod kuće	Kiša	Slabo	Slabo
Na strani	Hladno	Prosečno	Prosečno
	Umereno	Dobro	Dobro
	Vruće		

Rešenje: binarno – naš tim ili dobija ili gubi.

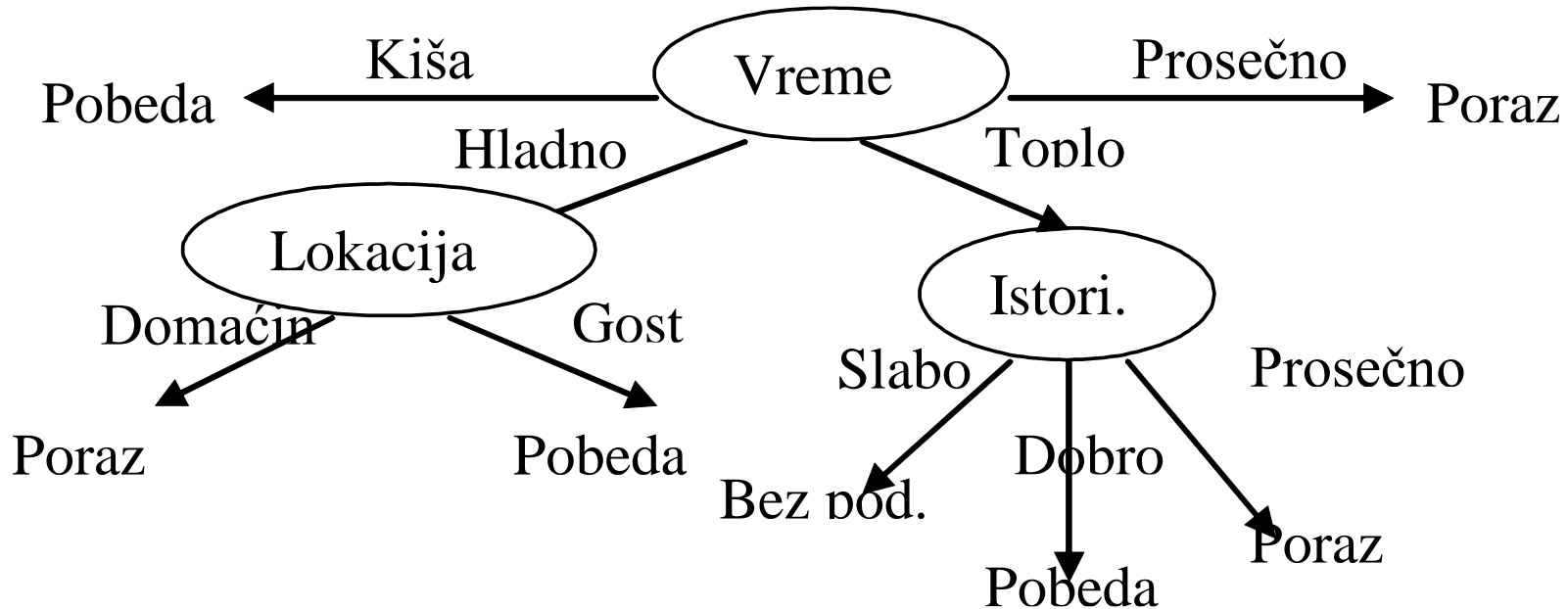
Primer razvoja sistema

- Primeri: Potrebno je sakupiti skup primera koji bi se iskoristili za generisanje stabla odlučivanja. Neka je polovina sezone prošla, te se iz prethodnih osam igara mogu dobiti primeri sa vrednostima faktora odlučivanja.
- Stablo odlučivanja: Čvorovi stabla odlučivanja su vreme, lokacija i istorija našeg tima. ID3 algoritam je odlučio da je faktor istorija privremeno nevažan za predviđanje ishoda igre. Međutim, kasnije će se ispostaviti da je on ipak važan.
- Testiranje: Stablo odlučivanja se može iskoristiti da se predvidi da li će naš tim dobiti ili izgubiti buduće mečeve. Polazeći od njega dobija se tabela. Tabela predviđanja pokazuje da neka predviđanja nisu dobra. Samo tri prognoze su dobre. Očigledno je da treba utvrditi razloge za loše prognoze. Došlo se do zaključka da je potrebno uzeti u obzir još jedan faktor – zdravlje tima, sa vrednostima slabo, prosečno i dobro.
- Kao sledeći korak, potrebno je vratiti se natrag nad prošlim igrama i dobiti vrednost zdravlja tima i dodati ovaj faktor i novi primer u naš sistem. U stablu se sada pojavljuje zapis o protivniku. Upoređivanjem rezultata predviđanja dobijenih uvođenjem novog faktora sa stvarnim rezultatima, vidi se da su predviđanja odlična.

Primer razvoja sistema

Faktori odlučivanje					Rezultat
Nedelja	Lokacija	Vreme	Istorija	Istorija prot.	Naš tim
1	Domaćin	Vruće	Dobro	Dobro	Pobeda
2	Domaćin	Kiša	Dobro	Prosečno	Pobeda
3	Gost	Umereno	Dobro	Prosečno	Poraz
4	Gost	Vruće	Dobro	Slabo	Pobeda
5	Domaćin	Hladno	Dobro	Dobro	Poraz
6	Gost	Vruće	Prosečno	Prosečno	Poraz
7	Domaćin	Umereno	Prosečno	Dobro	Poraz
8	Gost	Hladno	Slabo	Prosečno	Pobeda

Primer razvoja sistema



Primer razvoja sistema

Faktori odlučivanje					Rezultat	
Nedelja	Lokacija	Vreme	Istorija	Istorija prot.	Pred.	Aktuelan
9	Domaćin	Vruće	Slabo	Slabo	Bez podataka	Pobeda*
10	Domaćin	Umereno	Dobro	Prosečno	Poraz	Pobeda*
11	Gost	Hladno	Dobro	Dobro	Pobeda	Pobeda
12	Domaćin	Vruće	Dobro	Prosečno	Pobeda	Poraz*
13	Domaćin	Umereno	Dobro	Prosečno	Poraz	Pobeda*
14	Gost	Hladno	Dobro	Prosečno	Pobeda	Poraz*
15	Domaćin	Hladno	Prosečno	Dobro	Poraz	Poraz
16	Gost	Umereno	Slabo	Slabo	Poraz	Poraz

Primer razvoja sistema

Faktori odlučivanje						Rezultat	
Nedelja	Zdravlje	Lokacija	Vreme	Istorija	Istori. prot.	Pred.	Aktuelan
1	Dobro	Domaćin	Vruće	Dobro	Dobro	Pobeda	Pobeda
2	Prosečno	Domaćin	Kiša	Dobro	Prosečno	Pobeda	Pobeda
3	Prosečno	Gost	Umereno	Dobro	Prosečno	Poraz	Poraz
4	Prosečno	Gost	Vruće	Dobro	Slabo	Pobeda	Pobeda
5	Dobro	Domaćin	Hladno	Dobro	Dobro	Poraz	Poraz
6	Dobro	Gost	Vruće	Prosečno	Prosečno	Poraz	Poraz
7	Slabo	Domaćin	Umereno	Prosečno	Dobro	Poraz	Poraz
8	Slabo	Gost	Hladno	Slabo	Prosečno	Pobeda	Pobeda
9	Prosečno	Domaćin	Vruće	Slabo	Slabo	Pobeda	Pobeda
10	Prosečno	Domaćin	Umereno	Dobro	Prosečno	Pobeda	Pobeda
11	Dobro	Gost	Hladno	Dobro	Dobro	Pobeda	Pobeda
12	Dobro	Domaćin	Vruće	Dobro	Prosečno	Poraz	Poraz
13	Prosečno	Domaćin	Umereno	Dobro	Prosečno	Pobeda	Pobeda
14	Prosečno	Gost	Hladno	Dobro	Prosečno	Poraz	Poraz
15	Slabo	Domaćin	Hladno	Prosečno	Dobro	Poraz	Poraz
16	Prosečno	Gost	Umereno	Slabo	Slabo	Poraz	Poraz

Primer razvoja sistema

Faktori odlučivanje					Rezultat	
Nedelja	Zdravlje	Vreme	Istorija	Istori. prot.	Pred.	Aktuelan
1	Dobro	Vruće	Dobro	Dobro	Pobeda	Pobeda
2	Prosečno	Kiša	Dobro	Prosečno	Pobeda	Pobeda
3	Prosečno	Umereno	Dobro	Prosečno	?	Poraz
4	Prosečno	Vruće	Dobro	Slabo	Pobeda	Pobeda
5	Dobro	Hladno	Dobro	Dobro	?	Poraz
6	Dobro	Vruće	Prosečno	Prosečno	Poraz	Poraz
7	Slabo	Umereno	Prosečno	Dobro	Poraz	Poraz
8	Slabo	Hladno	Slabo	Prosečno	Pobeda	Pobeda
9	Prosečno	Vruće	Slabo	Slabo	Pobeda	Pobeda
10	Prosečno	Umereno	Dobro	Prosečno	?	Pobeda
11	Dobro	Hladno	Dobro	Dobro	?	Pobeda
12	Dobro	Vruće	Dobro	Prosečno	Poraz	Poraz
13	Prosečno	Umereno	Dobro	Prosečno	?	Pobeda
14	Prosečno	Hladno	Dobro	Prosečno	Poraz	Poraz
15	Slabo	Hladno	Prosečno	Dobro	Poraz	Poraz
16	Prosečno	Umereno	Slabo	Slabo	Poraz	Poraz

Prednosti indukcije

- **Otkrivanje pravila iz primera:** Na osnovu urađenih primera o nekom problemu iz kojih se može izvesti skup pravila odlučivanja.
- **Izbegavanje problema izbegavanje znanja:** Indukcija nudi tehniku kod koje se znanje sistema stiže direktno preko prošlih primera. Ova tehnika može da izbegne probleme vezane za pokušaje sticanja znanja direktno od eksperta.
- **Proizvodnja novog znanja:** Induktivno sredstvo može da proizvede ekspertski sistem koji bi upravljao donošenjem budućih odluka, čak iako ekspert nije eksplicitno svestan znanja za odlučivanje. Ovo je moguće jer indukcija može da otkrije oblike odlučivanja koji nisu očigledni, čak ni ekspertu domena.
- **Otkrivanje kritičnih faktora odlučivanja:** Indukcija otkriva najvažnije faktore odlučivanja. Ovo preimućstvo može da vodi ka sistemu koji odlučuje na osnovu samo nekoliko faktora, koji značajno poboljšavaju inteligenciju i efikasnost sistema.
- **Eliminisanje nevažnih faktora odlučivanja:** Često, se za faktore koji u početku izgledaju važni u donošenju konačnih odluka, utvrdi kroz indukciju da su nebitni. Ovo je značajno za samog eksperta i efikasnost sistema, ali se mora ipak biti obazriv, jer se dodavanjem novih primera može desiti da uklonjeni faktori ponovo postanu važni.

Prednosti indukcije

- **Otkrivanje protivrečnosti:** Zbog načina na koji se primeri unose u indukcione sisteme, neka razvojna okruženja mogu lako da otkriju primere koji daju protivrečne rezultate za isti skup vrednosti faktora odlučivanja.

Faktori odlučivanja		Rezultati
Pritisak pumpe	Temperatura pumpe	Uslovi
Nizak	Visok	Oticanje pumpe
Nizak	Visok	Normalna pumpa

- Induktivno razvojno okruženje može da otkrije protivrečne rezultate i o tome obavesti projektanta. U nekim slučajevima, protivrečnost može da bude prihvatljiva. Za date vrednosti faktora odlučivanja, oba rezultata mogu da budu logična.
- U većini slučajeva, otkrivena protivrečnost označava problem. Problem može da bude u lošem primeru, ili još češće, ili u faktorima odlučivanja ili vrednostima koje su neadekvatne.

Prednosti indukcije

Ako se posmatra ponovo primer pumpe i otkrivanje protivrečnosti. Ekspert može da utvrdi da bi trebalo uzeti u obzir broj obrta/min motora koji kontrolišu pumpu. Ekspert može da da i vrednosti za ovaj faktor koje vode ka neprotivrečnim rezultatima.

Faktori odlučivanja			Rezultati
Broj obrtaja/min	Pritisak pumpe	Temperatura pumpe	Uslovi
Visok	Nizak	Visok	Oticanje pumpe
Nizak	Nizak	Visok	Normalna pumpa

Dodavanje ovog novog faktora odlučivanja ilustruje ne samo kako se može eliminisati protivrečnost, već otkriveni problemi mogu da odvedu ka otkrivanju dodatnog znanja domena.

Nedostaci

- **Teškoće izbora dobrih faktora odlučivanja:** Efektivnost sistema zavisi od izbora dobrih faktora odlučivanja. Na primer, koje faktore treba uzeti u obzir pri razvoju sistema za predviđanje pobednika fudbalske igre.
- **Teškoće u razumevanju pravila:** Većina induktivnih sredstava proizvode rezultate u formi stabla odlučivanja. Kod složenih problema, teško je razumeti proces odlučivanja praćenjem kroz stablo. Ovo stablo postaje naročito složeno kada se sredstvo koristi za određivanje pravila radi njihovog uključivanja u standardne produkcione sisteme.
- **Primenjivi su samo kod klasifikacionih problema:** Indukcija je dobra za probleme u kojima se skup vrednosti atributa može klasifikovati u neke očekivane rezultate. Na primer, problem dijagnoze pokušava da klasifikuje skup simptoma u zadato stanje greške. Drugi tipovi problema, kao što su planiranje ili projektovanje, teško se mogu rešiti induktivnim pristupom.

C 4.5

- C4.5 je proširenje ID3 algoritma (Weka J48)
- Gradi stabla odlučivanja iz skupa podataka za obučavanje na isti način kao ID3, koristeći koncept informacione entropije
- Skup za obučavanje je skup $S = \{s_1, s_2, s_3, \dots\}$ od već klasifikovanih instanci
- Svaka instanca s_i se sastoji od p - dimenzionog vektora sa vrednostima atributa koji se odnosi na instancu

C 4.5

- U svakom čvoru stabla se bira atribut koji najefikasnije deli svoj skup instanci na podskupove sa najzastupljenijom klasom izabranog atributa.
- Kriterijum deljenja je normalizovani dobitak informacija
- Za donošenje odluke bira se atribut sa najvećim normalizovanim informacionim dobitkom.
- Algoritam C4.5 se zatim rekurzivno ponavlja na podeljenim podskupovima.

C 4.5

C4.5 (Иницијални скуп D , Циљани Атрибут, Атрибути)

- Креирати корени чвор за стабло
- Базични случајеви
 - Све инстанце из скупа припадају истој класи - враћа се корени чвор који садржи ту класу.
 - Ниједан од атрибута не даје никакав информациони добитак - враћа корени чвор који садржи најзаступљенију класу.
 - Ако се добије инстанца класе која није претходно виђена - враћа корени чвор који садржи најзаступљенију класу.
- Ако ниједан од базичних случајева није испуњен (иначе):
 - A_{best} = атрибут који има највећи информациони добитак
 - Формира се корени чвор стабла са најбољим атрибутом A_{best}
 - D_v = скуп свих подскупова скупа D подељених према атрибуту A_{best}
 - **foreach** (D_v)
 - $Tree_v = C4.5$ (Скуп D_v , Циљани_Атрибут A_{best} , преостали Атрибути)
 - Поставља се $Tree_v$ као нова грана стабла $Tree$ где је вредност на грани према вредности дељења атрибута A_{best}
- **end_for**
- Вратити корени чвор.

Prednosti u odnosu na ID3

- Rukovanje kontinualnim i diskretnim atributima
 - kreira se prag, a zatim deli skup instanci na one čija je vrednost atributa iznad praga i one koje su manje ili jednake njemu.
- Rukovanje podacima za obučavanje koji imaju nedostajuće vrednosti atributa
 - Dozvoljena je da vrednosti atributa bude označena kao ?
 - Nedostajuće vrednosti atributa jednostavno se ne koriste u proračunima dobitka i entropije.
- Rukovanje atributima koji imaju različite cene koštanja
- Čišćenje stabala od suvišnih grana
 - povratak nazad kroz kreirano stablo
 - i pokušava se da se ukloni grane koje ne pomažu zamenjujući ih čvorovima lista