Mašinsko učenje

Analiza sentimenta

Nemanja Maček

Sadržaj

Uvodne napomene

Da li sa slike možete da zaključite šta je analiza sentimenta?

Izvor slike: https://www.quantzig.com/blog/sentiment-analysis-social-media-strategy

Uvodne napomene

Da li sa slike možete da zaključite šta je analiza sentimenta?

Izvor slike: http://www.polyvista.com/blog/how-sentiment-analysis-helps-businesses-strengthen-customer-experience

Uvodne napomene

Primer iz prakse.

Izvor slike: https://github.com/ayushoriginal/Sentiment-Analysis-Twitter

Uvodne napomene

Primer iz prakse.

- Na stranici www.oceniprofesora.com čitamo sledeće komentare za izvesnog „profesora“:
  - „Predavanja ovog profesora ne sluze apsolutno nicemu, covek cita knjigu i to ne rado, vidi se da mu nije stalo da prenese studentima znanje, da li zbog toga sto je covek u godinama ili neceg drugog, nije bitno. Takodje, medju mnogima sam koji na diplomu cekaju samo zbog ispita kod ovog profesora jer zahteva da se nauce stvari koje su besmislene tj. koje zasigurno necemo koristiti u praksi.“
  - „Apsolutno nezanimljiva predavanja, vežbe takođe. Predavanja se svode na čitanje iz knjige, a vežbe na čitanje sa slajdova. Smatram da profesor ne bi položio svoj ispit, a kamoli asistentkinja.“
  - „Los predavac, neinteresantan i nejasan.“
- Obradom prethodno pomenutih i sličnih komentara, ovog „profesora“ bi identifikovali kao „negativnog“, u kontekstu analize sentimenta.

Gde se koristi analiza sentimenta?

- Neki od primera primene analize sentimenta su:
  - analiza sentimenta i mišljenja o proizvodima i uslugama,
  - analiza sentimenta u finansijskom sektoru,
  - analiza sentimenta u poslovnoj inteligenciji,
  - mišljenja iskazana na blogovima, forumima i društvenim mrežama (npr. analiza sentimenta se može primeniti i za potrebe praćenja konzistentnosti i nekonzistentnosti između datih izjava i akcija na državnom nivou, a rezultati izbora se mogu uspešnije predviđati praćenjem sentimenta u diskusijama na blogovima i forumima), itd.

Metodologija analize sentimenta.

- Analiza sentimenta je složen analitički proces koji se realizuje kroz nekoliko sekvencijalnih koraka:
  - pronalaženje i prikupljanje podataka,
  - priprema podataka,
  - izdvajanje obeležja,
  - izdvajanje sentimenta,
  - klasifikacija sentimenta i njihovo sumiranje, i
  - izveštavanje i vizualizacija rezultata.

Pretraga, preuzimanje i čišćenje podataka.

- Komentari o proizvodima i uslugama se mogu pronaći na zvaničnim sajtovima kompanija, različitim forumima na internetu, specijalizovanim blogovima, Facebook stranicama ili drugim sajtovima društvenih medija.
- Radi sticanja sveobuhvatnog uvida u diskusije vođene o proizvodima ili uslugama, potrebno je preuzeti sa Interneta relevantne komentare.
- Za ove potrebe se koriste tzv. pretraživači Veba (engl. *Web crawlers*) koji pretražuju Internet, preuzimaju sadržaj ciljne Veb stranice i čuvaju ga na lokalnom disku.
- Alternativno, koristi se kod kreiran prema strukturi sajta koji će ciljati i preuzimati sadržaj između određenih HTML tagova.

Preprocesiranje podataka.

- Preuzete Veb stranice mogu sadržati komentare o proizvodima ili uslugama koje nisu predmet analize ili su van okvira domena, te je potrebno izvršiti filtriranje teksta kako bi se zadržao sadržaj koji se odnosi samo na ciljani proizvod ili uslugu.
- Za ove potrebe se može primeniti klasifikacija prikupljenih tekstova kako bi se utvrdilo da li Veb stranica sadrži diskusiju o relevantnom proizvodu/usluzi.
  - Klasifikator se obučava skupom relevantnih i irelevantnih pojmova na osnovu kojih će se vršiti procena sadržaja dokumenta.
  - Na osnovu odluke klasifikatora Veb stranica se ili zadržava za dalju analizu ili se odbacuje.
- Dodatno, u ovom koraku se realizuje filtriranje i uklanjanje sadržaja koji nisu u tekstualnom formatu, poput HTML tagova, vrši se uklanjanje praznih polja, proširivanje skraćenica, steming i uklanjanje stop reči.

Izdvajanje obeležja.

- Izdvajanje obeležja podrazumeva njihovo izdvajanje iz članka ili diskusije.
- Ukoliko su, npr., predmet analize komentari o fakultetima, bitna obeležja bi bila nazivi konkretnih fakulteta – ukoliko obeležja nisu adekvatno izdvojena, kompletna analiza sentimenta ili mišljenja postaje irelevantna.
- Neki od načina za izdvanjanje obeležja su:
  - kreiranje rečnika (obezbeđuje listu relevantnih pojmova za analizirani domen, ali se zahteva često ažuriranje, naročito u slučaju komentara o komercijalnim proizvodima kod kojih se često lansiraju novi modeli),
  - formulisanje pravila (tražene ključne reči obično ispoljavaju određeni šablon – ovaj pristup je robusniji od prethodnog, ali se pravila moraju ručno ažurirati kako bi se obuhvatila sva odstupanja od predefinisanog stringa pretrage),
  - mašinsko učenje, itd.

Analiza sentimenta

Izdvajanje sentimenta.

- Izdvajanje sentimenta se najčešće obavlja upotebom rečnika sentiment izraza* i njihove semantičke orijentacije.
- Ovaj pristup nosi određena ograničenja – na primer, reč „visoka“ može imati negativan sentiment kada se koristi u kontekstu školarine, dok u kontekstu ocene nosi pozitivan sentiment.
- Kada se u tekstu identifikuje relevantan entitet ili predefinisana sentiment reč, strukturirani sentiment se izdvaja iz rečenice u formi: entitet/reči i ocena.
- Pridružena ocena se odnosi na pozitivan ili negativan polaritet identifikovane reči u rečenici.

* Termini sentiment izraz i sentiment reč koriste ravnopravno kako bi označili reči, fraze i idiome koje nose sentiment.

Analiza sentimenta

Izdvajanje sentimenta.

- Posebnu pažnju treba posvetiti negaciji – tehnike obrade prirodnog jezika se koriste kako bi se utvrdili efekti negacije na pridruženu sentiment reč.
- Zavisno od strukture, kao i izvora prikupljenih podataka, u ovom koraku je potrebno izvršiti i procenu subjektivnosti iskazanog dokumenta, tako da se zadrže rečenice u kojima je prisutna subjektivnost, dok se one kojima se iskazuju činjenice odbacuju.
  - Napomena: ovaj korak je potrebno realizovati u uslovima kada je izvor podataka forum, blog, društvene mreže, dok se na sajtovima za recenziranje uglavnom iskazuju subjektivni stavovi.
- Nakon ovog koraka, izdvojene sentiment reči su u strukturiranom formatu.

Analiza sentimenta

Klasifikacija sentimenta i njihovo sumiranje.

- Zavisno od nivoa na kom se radi klasifikacija (dokument, rečenica, reč ili fraza, aspekt) iskazani sentiment se najčešće klasifikuje u pozitivnu ili negativnu kategoriju primenom različitih algoritama mašinskog učenja.
- Kako bi izdvojeni sentiment iz prethodnog koraka bio smislen i značajan za izveštavanje, potrebno ga je agregirati.
- Postoje različiti nivoi agregiranja i sumiranja.
- Na primeru fakulteta, prirodan način agregiranja bilo bi grupisanje svih pozitivnih i negativnih reči prema pojedinačnim fakultetima, predmetima i drugim kriterijumima.

Izazovi u analizi sentimenta.

- Neki od izazova analize sentimenta su:
  - domenska zavisnost,
  - jezička zavisnost,
  - izdvajanje atributa,
  - detekcija i upravljanje negacijom, sarkazmom i ironijom,
  - utvrđivanje polariteta,
  - utvrđivanje subjektivnosti dokumenata, itd.

Domenska zavisnost.

- Potrošači iskazuju svoja mišljenja o temama iz različitih domena, pri čemu su mišljenja i osećanja iskazana na različite načine, različitim vokabularom, stilom pisanja, različitom dužinom teksta, upotrebom žargona, itd.
- Domenska zavisnost je delimično posledica promene u vokabularu, te identični izrazi mogu nositi potpuno drugačiji sentiment u različitim domenima – na primer, „pročitaj knjigu“ se može tretirati kao pozitivan sentiment kada je reč o kritikama knjige, a negativan kada je reč o kritikama filma.
- Klasifikator obučen skupom kritika o jednom tipu proizvoda često ne postiže istu performanse kada se primeni nad skupom kritika o drugom proizvodu.
- Usled domenske zavisnosti, potrebno je kreirati anotirane skupove podataka za svaki domen i razviti domen-specifičan klasifikator.

Analiza sentimenta

Jezička zavisnost.

- S obzirom na zastupljenost engleskog jezika u društvenim medijima u vidu brojnih statusa i komentara, a pogotovo značaja koji ovaj jezik ima u naučnoj zajednici, razumljivo je da je i najveći broj resursa (rečnika sa sentiment oznakama i korpusa), kao i analitičkih pristupa i primenjenih tehnika i metoda analize sadržaja razvijen upravo za englesko govorno područje.
- Izrazi novog jezika se mogu uskladiti sa izrazima jezika za koji već postoje razvijeni resursi upotrebom višejezičkih rečnika, paralelnih korpusa ili mašinskog prevođenja.

Analiza sentimenta

Izdvajanje obeležja.

- Svaki dokument ili rečenica, koji je predmet analize sentimenta, najčešće se predstavlja kao vektor svih reči koje se u njemu pojavljuju.
- Jedan od osnovnih problema je visoka dimenzionalnost vektora obeležja kojima se obično opisuju tekstualni podaci.
- Svako jedinstveno obeležje koje se pojavljuje u kolekciji dokumenata predstavlja jednu dimenziju u vektorskom prostoru, a njihov broj može dostići i hiljade različitih obeležja.
- Visoka dimenzionalnost može dovesti do problema prenaučenosti.
- Iz tog razloga su metode odabira i izdvajanja obeležja važne kako bi se smanjila dimenzionalnost vektora, omogućila generalizacija i skratilo vreme za obučavanje.

Izdvajanje obeležja.

- Obeležja koja se najčešće koriste su:
  - prisutnost i frekvencija reči,
  - sentiment reči i fraze,
  - negacija, itd.

Prisutnost i frekvencija reči.

- Atributi ovog tipa predstavljeni su brojem njihovog pojavljivanja u dokumentima.
- Pored merenja učestalosti pojavljivanja reči prebrojavanjem, može se posmatrati značaj reči koristeći meru TF-IDF koja meri težinu termina, odnosno stepen u kom reč doprinosi sadržaju dokumenta.
- TF-IDF se izračunava pomoću frekvencije termina u dokumentu TF (engl. *term frequency*) i inverzne frekvencije termina u skupu dokumenata IDF (engl. *inverse document frequency*), koja proverava broj dokumenata u kojima se pojavljuje konkretna reč.

Prisutnost i frekvencija reči.

- TF-IDF mere se može izračunati na sledeći način:

$$tfidf(j) = t(f) \times idf(j)$$

$$idf(j) = \log\left(\frac{N}{df(j)}\right)$$

- gde je j termin, $tf(j)$ broj pojavljivanja datog termina u dokumentu, $df(j)$ broj dokumenata u skupu dokumenata koji sadrže dati termin, a N broj dokumenata u skupu.
- Kada se reč pojavi u velikom broju dokumenata, smatra se nevažnom i pridružena težina je manja. Kada je reč relativno jedinstvena i pojavljuje se u nekoliko dokumenata, smatra se važnom i pridružena težina je veća.

Analiza sentimenta

Sentiment reči i fraze.

- Sentiment reči i fraze predstavljaju reči koje se učestalo koriste u iskazivanju mišljenja, poput dobro, loše itd.
- Pojedine fraze ukazuju na jasno mišljenje, stav ili osećanje, a da u njima nije upotrebljena konkretna sentiment reč (npr. kapa dole), što predstavlja poseban izazov.

Detekcija i upravljanje negacijom, sarkazmom i ironijom.

- Negacija predstavlja učestalu jezičku konstrukciju koja menja značenje reči (odnosno, dela rečenice ili cele rečenice), kao i orijentaciju iskazanog sentimenta i potrebno ju je uzeti u obzir pri analizi mišljenja i osećanja.
- Prisustvo negacije u rečenici ne znači da će se polaritet sentimenta uvek izmeniti.
- Naredni primeri ilustruju problematiku negacije:

(1) Dopada mi se predmet.
(2) Ne dopada mi se predmet.
(3) Ne samo da je predmet dosadan, nego su i materijali zastareli.
(4) Ne dopada mi se predmet, ali su predavanja zanimljiva.

- Šta se dešava sa sentimentima usled upotrebe negacije u prethodnim primerima?

Detekcija i upravljanje negacijom, sarkazmom i ironijom.

(1) Dopada^{+} mi se predmet.

- U primeru (1) iskazan je pozitivan stav sentiment izrazom „dopada“.

Detekcija i upravljanje negacijom, sarkazmom i ironijom.

(2) [Ne dopada^{+}]$^{-}$ mi se predmet.

- Negacija prethodne rečenice upotrebom ključne reči „ne“ menja njeno značenje i iskazuje negativan stav.

Detekcija i upravljanje negacijom, sarkazmom i ironijom.

(3) Ne samo da je predmet dosadan, nego su i materijali zastareli.

- U primeru (3) negacija, odnosno ključna reč „ne“ ne menja sveukupni sentiment rečenice koja ostaje negativna.

Detekcija i upravljanje negacijom, sarkazmom i ironijom.

(4) Ne dopada mi se predmet, ali su predavanja zanimljiva.

- U primeru (4) negacija menja polaritet sentimenta u prvom delu rečenice, dok se drugim delom rečenice iskazuje pozitivan stav.
- Dakle, prisustvo negacije u rečenici ne znači da se svako iskazano mišljenje negira, odnosno da se njihovo značenje menja.

Detekcija i upravljanje negacijom, sarkazmom i ironijom.

- Kada se rukuje negacijom, potrebno je ispravno identifikovati njen opseg važenja, odnosno koji deo rečenice će imati izmenjeno značenje usled prisustva negacije.
- U većini slučajeva upotreba negacije nije jednostavna, kao u primeru (2).
- Pored standardnih ključnih reči negacije („ne“, „nije“, „nema“ i sl.) vrlo često se koriste neutralizatori (reči koje umanjuju značenje, poput „malo“, „donekle“ i slične), veznici i drugi konstrukti koji negiraju značenje ili umanjuju iskazani sentiment.

Detekcija i upravljanje negacijom, sarkazmom i ironijom.

- Negacija se može iskazati i na vrlo suptilan način upotrebom sarkazma i ironije, koji se prilično teško identifikuju.
- Neki primeri njihove upotrebe su:

(5) Jedna žena koja je jako posvećena svom poslu. Toliko posvećena da održava konsultacije jednom mesečno a i kad ih održi nigde je nema :)
(6) Zanimljiva osoba... Ako niste žensko, onda vam da 3 ocene manje u startu.

- Sarkazam i ironija su uobičajeni u recenzijama proizvoda i usluga, a učestalo se pojavljuju i u političkim diskusijama i komentarima.

Detekcija i upravljanje negacijom, sarkazmom i ironijom.

- Jedan od osnovnih problema u vezi sa zadatkom identifikovanja ironije je nepostojanje saglasnosti među istraţivačima po pitanju formalne definicije ironije ili sarkazma i njihove strukture.
- Smatra se da je primena ironije i sarkazma, kao i termini koji ukazuju na njih, promenjiva i da sarkazam ima regionalne varijacije.
- Iz tog razloga nije moguće definisati jasna uputstva za identifikovanje ironije i sarkazma.
- Međutim, ljudi generalno razumeju značenje ironičnih izraza i mogu ih pouzdano identifikovati.
- Dodatno, određena pojavljivanja sarkastičnih izjava mogu se shvatiti samo kada su smeštena u određeni kontekst.
- Zadaci identifikovanja negacije, sarkazma i ironije predstavljaju bitan korak analize sentimenta, jer njihovo ispravno detektovanje poboljšava performanse sistema analize sentimenta.

Utvrđivanje polariteta.

- Upotreba rečnika polarizovanih reči i fraza (pozitivnih i negativnih) predstavlja uobičajeni pristup sentiment analizi.
- Ovakvi rečnici se mogu kreirati ručno ili automatski.
- Generisanje polariteta reči i fraza je aktivni istraživački pravac, a neke od predloženih tehnika za za učenje polariteta su:
  - tehnike bazirane na korpusu, koje podrazumevaju manuelno označavanje pozitivnih i negativnih reči,
  - tehnike koje koriste informacije o leksičkim odnosima i resurse poput WordNet mreže pojmova na osnovu kojih se kreira sistem za automatsko izdvajanje pojmova, itd.
- Pojedine reči posmatrane samostalno mogu nositi jedan polaritet koji će se izmeniti kada su one upotrebljene u određenom kontekstu.

Analiza sentimenta

Utvrđivanje subjektivnosti dokumenta.

- Nasuprot objektivnim iskazima koji odražavaju činjenice, subjektivni iskazuju lična osećanja, mišljenja i uverenja.
- Subjektivni iskazi se mogu pojaviti u formi mišljenja, želje, verovanja, sumnje, osude, itd.
- Mnogi od njih podrazumevaju sentiment, ali u određenim situacijama subjektivni iskazi ne sadrže sentiment (npr. želim foto-aparat koji će praviti kvalitetne fotografije).
- Detektovanje subjektivnosti i analiza sentimenta su dva različita zadatka obrade prirodnog jezika – identifikovanje subjektivnosti svodi se na problem razdvajanja iskaza koji sadrže činjenice i one koji sadrže subjektivne iskaze.
- Nad izdvojenim subjektivnim iskazima se može sprovesti sentiment analiza kako bi se izvršila klasifikacija sadržaja prema polaritetu – uklanjanje objektivnih rečenica iz tekstova pre analize sentimenta rezultovaće višom tačnošću klasifikacije.

Skup podataka

Šta nam je potrebno?

- Potrebni su nam tvitovi i odgovarajuće labele koje ukazuju da li je sentiment tvita pozitivan, negativan ili neutralan.
- Koristićemo Niek Sandersov korpus koji sadrži ručno obeležene tvitove (za detalje o preuzimanju v. knjigu, str. 124, pri čemu se skript za preuzimanje nalazi u dodatku knjige).
- Podaci sadrže četiri različite labele sentimenata:

```
>>> X, Y = load_sanders_data()
>>> classes = np.unique(Y)
>>> for c in classes: print("#%s: %i" % (c, sum(Y==c)))
#irrelevant: 490
#negative: 487
#neutral: 1952
#positive: 433
```

Naivni Bajes

Bajesovo pravilo.

$$P(H|E) = \frac{P(E|H) \times P(H)}{P(E)}$$

- gde je:
  - H – hipoteza (engl. *hypothesis*),
  - E – opažaj (engl. *evidence*),
  - $P(H)$ – verovatnoća hipoteze H (engl. *prior probability*),
  - $P(E)$ – verovatnoća opažaja, tj. stanja na koje ukazuju prikupljeni podaci,
  - $P(E|H)$ – (uslovna) verovatnoća opažaja E ukoliko važi hipoteza H,
  - $P(H|E)$ – (uslovna) verovatnoća hipoteze H ukoliko imamo opažaj E.

Naivni Bajes

Bajesovo pravilo – primer.

- Pretpostavite sledeće:
  - jednog jutra ste se probudili sa povišenom temperaturom,
  - prethodnog dana ste čuli da je u gradu počela da se širi virusna infekcija, ali da je verovatnoća zaraze mala, svega 2,5%,
  - takođe ste čuli da je u 50% slučajeva virusna infekcija praćena povišenom temperaturom
  - u vašem slučaju, povišena temperatura se javlja svega par puta u godini, tako možemo reći da je verovatnoća da imate povišenu temperaturu 5%.
- Pitanje: kolika je verovatnoća da, pošto imate povišenu temperaturu, da imate i virusnu infekciju?

Bajesovo pravilo – primer.

Teorija	**Primer**
Hipoteza (H)	Imate virusnu infekciju.
P(H)	0,025
Opažaj (E)	Imate povišenu temperaturu.
P(E)	0,05
(uslovna) verovatnoća opažaja E ukoliko važi hipoteza H: P(E|H)	Verovatnoća da je virusna infekcija praćena povišenom temperaturom je 0,5
(uslovna) verovatnoća hipoteze H ukoliko imamo opažaj E: P(H|E)	Verovatnoća da pošto imate povišenu temperaturu da imate i virusnu infekciju?

$$P(H|E) = \frac{P(E|H) \times P(H)}{P(E)} = \frac{0{,}5 \times 0{,}025}{0{,}05} = 0{,}25$$

Još jedan primer.

- Lekar zna da meningitis u 50% slučajeva prouzrokuje kočenje vrata.
- Verovatnoća da bilo koji pacijent ima meningitis (M) je 1/50.000.
- Verovatnoća da bilo koji pacijent ima ukočen vrat (S) je 1/20.
- Ako pacijent ima ukočen vrat, koja je verovatnoća da ima i meningitis?

$$P(M|S) = \frac{P(S|M)P(M)}{P(S)} = \frac{0{,}5 \times \frac{1}{50.000}}{\frac{1}{20}} = 0{,}0002 = 0{,}02\%$$

Naivni Bajesov klasifikator.

- Naivni Bajesov klasifikator (engl. *Naïve Bayes classifier*) je klasifikator zasnovan na Bajesovom pravilu.
- Uvode se dve „naivne“ pretpostavke nad atributima:
  - sva obležja su a priori podjednako važna, i
  - sva obeležja su statistički nezavisna (vrednost jednog obeležja nam ne govori ništa o vrednosti drugog obeležja).
- Ove pretpostavke najčešće nisu tačne, ali i pored toga, u praksi ovaj klasifikator daje dobre rezultate.

Naivni Bajesov klasifikator.

- Pretpostavka nezavisnosti obeležja značajno pojednostavljuje računanje uslovnih verovatnoća.
- Neka je E_i opažaj (raspoloživi podaci) vezan za obeležje i.

$$P(E|H) = P(E_1, E_2, \dots, E_n|H) = P(E_1|H) \times P(E_2|H) \times \cdots \times P(E_n|H)$$

$$P(H|E) = \frac{\big(P(E_1|H) \times P(E_2|H) \times \cdots \times P(E_n|H)\big) \times P(H)}{P(E)}$$

Naivni Bajes

U kontekstu analize sentimenta tvitova ...

- Takozvani Bernulijev model vodi računa samo o bulovskim obeležjima, drugim rečima, da li se određena reč pojavila jednom ili više puta u određenom tvitu nije bitno.
- Nasuprot tome, multinomijalni model koristi broj reči kao obeležja, odnosno, broj pojavljivanja neke reči u tvitu je bitan.
- Da bi pojednostavili „slučaj“ koristićemo Bernulijev model za objašnjenje primene naivnog Bajesa u analizi sentimenta.

Slučaj sa dva obeležja.

- Neka je:
  - C klasa tvita (pozitivan ili negativan),
  - F_1 bulovsko obeležje pojavljivanja jedne reči u tvitu (tj. da li se reč barem jedan put pojavila ili ne),
  - F_2 bulovsko obeležje pojavljivanja druge reči u tvitu.
- Verovatnoću pripadnosti klasi C, tj. $P(C|F_1, F_2)$ određujemo na osnovu vrednosti obeležja F_1 i F_2, koristeći Bajesovu teoremu:

$$P(A) \times P(B|A) = P(B) \times P(A|B)$$

$$P(F_1, F_2) \times P(C|F_1, F_2) = P(C) \times P(F_1, F_2|C) \rightarrow P(C|F_1, F_2) = \frac{P(C) \times P(F_1, F_2|C)}{P(F_1, F_2)}$$

Slučaj sa dva obeležja.

- $P(C)$ je verovatnoća da poznajemo klasu bez pozavanja podataka koja se može odrediti na osnovu procenta instanci koje pripadaju datoj klasi u obučavajućem skupu, $P(F_1, F_2)$ je opažaj za obeležja F_1 i F_2, a $P(F_1, F_2|C)$ određujemo koristeći pretpostavku o nezavisnosti obeležja:

$$P(F_1, F_2|C) = P(F_1|C) \times P(F_2\ |C)$$

- Iz ovog sledi:

$$P(C|F_1, F_2) = \frac{P(C) \times \left(P(F_1|C) \times P(F_2|C)\right)}{P(F_1, F_2)}$$

- Da ponovimo: pretpostavke o a priori podjednakoj važnosti i statističkoj nezavisnosti obeležja najčešće nisu tačne, ali ovaj klasifikator u praksi daje dobre rezultate.

Upotreba naivnog Bajesa za klasifikaciju.

- Za novi tvit, verovatnoće računamo na sledeći način:

$$P(C = "pos"|F_1, F_2) = \frac{P(C = "pos") \times \left(P(F_1|C = "pos") \times P(F_2|C = "pos")\right)}{P(F_1, F_2)}$$

$$P(C = "neg"|F_1, F_2) = \frac{P(C = "neg") \times \left(P(F_1|C = "neg") \times P(F_2|C = "neg")\right)}{P(F_1, F_2)}$$

$$c_{best} = \underset{c \in C}{\operatorname{argmax}}\, P(C = c) \times \left(P(F_1|C = c) \times P(F_2|C = c)\right)$$

Primer.

- Radi jednostavnosti objašnjenja, pretpostavićemo da Twitter dozvoljava upotrebu samo dve reči – „awesome“ i „crazy“ – i da smo ručno klasifikovali određeni broj tvitova.
- Uzmite u obzir da u opštem slučaju tvit „crazy“ može biti pozitivan i negativan, zavisno od konteksta („crazy song“, odnosno „crazy idiot“).

Tvit	Klasa
awesome	Pozitivan tvit
awesome	Pozitivan tvit
awesome crazy	Pozitivan tvit
crazy	Pozitivan tvit
crazy	Negativan tvit
crazy	Negativan tvit

Primer.

$$P(C = "pos") = 4/6$$
$$P(C = "neg") = 2/6$$

$$P(F_1 = 1|C = "pos") = \frac{broj\ pozitivnih\ tvitova\ koji\ sadrže\ reč\ "awesome"}{broj\ svih\ pozitivnih\ tvitiova} = 3/4$$

$$P(F_1 = 0|C = "pos") = 1 - P(F_1 = 1|C = "pos") = 1/4$$

$$P(F_2 = 1|C = "pos") = 2/4$$

$$P(F_2 = 0|C = "pos") = 2/2$$

- Na sličan način se računaju verovatnoće koje se odnose na negativne tvitove.

Primer.

$$P(F_1, F_2) = P(F_1, F_2 | C = "pos") \times P(C = "pos") + P(F_1, F_2 | C = "neg") \times P(C = "neg")$$

$$P(F_1 = 1, F_2 = 1) = \frac{3}{4} \times \frac{2}{4} \times \frac{4}{6} + 0 \times 1 \times \frac{2}{6} = \frac{1}{4}$$

$$P(F_1 = 1, F_2 = 0) = \frac{3}{4} \times \frac{2}{4} \times \frac{4}{6} + 0 \times 0 \times \frac{2}{6} = \frac{1}{4}$$

$$P(F_1 = 0, F_2 = 1) = \frac{1}{4} \times \frac{2}{4} \times \frac{4}{6} + \frac{2}{2} \times \frac{2}{2} \times \frac{2}{6} = \frac{5}{12}$$

Primer – klasifikacija novog tvita (1).

- Tvit: „awesome“, F_1=1, F_2=0.

$$P(C = "pos" | F_1 = 1, F_2 = 0) = \frac{\frac{3}{4} \times \frac{2}{4} \times \frac{4}{6}}{\frac{1}{4}} = 1$$

$$P(C = "neg" F_1 = 1, F_2 = 0) = \frac{\frac{0}{2} \times \frac{2}{2} \times \frac{2}{6}}{\frac{1}{4}} = 0$$

- Klasifikovan kao pozitivan.

Primer – klasifikacija novog tvita (2).

- Tvit: „crazy“, F_1=0, F_2=1.

$$P(C = "pos" | F_1 = 0, F_2 = 1) = \frac{\frac{1}{4} \times \frac{2}{4} \times \frac{4}{6}}{\frac{5}{12}} = \frac{1}{5}$$

$$P(C = "neg" F_1 = 0, F_2 = 1) = \frac{\frac{2}{2} \times \frac{2}{2} \times \frac{2}{6}}{\frac{5}{12}} = \frac{4}{5}$$

- Klasifikovan kao negativan.

Primer – klasifikacija novog tvita (3).

- Tvit: „awesome crazy“, F_1=1, F_2=1.

$$P(C = "pos" | F_1 = 1, F_2 = 1) = \frac{\frac{3}{4} \times \frac{2}{4} \times \frac{4}{6}}{\frac{1}{4}} = 1$$

$$P(C = "neg" F_1 = 1, F_2 = 1) = \frac{\frac{0}{2} \times \frac{2}{2} \times \frac{2}{6}}{\frac{1}{4}} = 0$$

- Klasifikovan kao pozitivan.

Rešavanje problema prethodno neviđenih reči.

- Klasifikacioni model trivijalnim tvitovima dodeljuje ispravne labele.
- Postavlja se, međutim, sledeće pitanje: šta možemo da uradimo sa rečima koje se ne nalaze u korpusu za obuku?
- Uzevši u obzir prethodne jednačine, njima će uvek biti dodeljene verovatnoće jednake nuli.
- Problem se praktično rešava tehnikom dodavanja jedinice (engl. *add-one smoothing*) koja se zasniva na pretpostavci da čak iako nismo videli reč u našem korpusu, postoji mogućnost da je naš korpus samo deo većeg korpusa koji ne sadrži tu reč.

Rešavanje problema prethodno neviđenih reči.

- Pretpostavićemo da smo svaku reč videli jednom više, pa umesto $P(F_1 = 1|C = "pos") = 3/4$ računamo:

$$P(F_1 = 1|C = "pos") = \frac{3+1}{4+2} = 0{,}67$$

- Broj 2 se u imeniocu dodaje zato što imamo dva obeležja – drugim rečima, za *N* obeležja dodali bi *N*.
- Važi i dalje:

$$P(F_1 = 1|C = "pos") + P(F_1 = 0|C = "pos") = \frac{3+1}{4+2} + \frac{1+1}{4+2} = 1$$

Problem pri obradi vrlo malih brojeva.

- Povećanjem korpusa, odnosno broja reči neke verovatnoće će biti jako male.
- Numpy ne može da obradi neke jako male brojeve.
- Kako možemo da „očešemo“ donji limit?
- Na primer, pokušajte da izmnožite 64 uslovnih verovatnoća od 0,0001 (što znači da imamo 64 slabo verovatnih vrednosti obeležja) i imaćete problem sa nečim što se naziva *arithmetic underflow*.

Rešenje problema obrade vrlo malih brojeva.

- Ako jednačinu:

$$\log(x \times y) = \log(x) + \log(y)$$

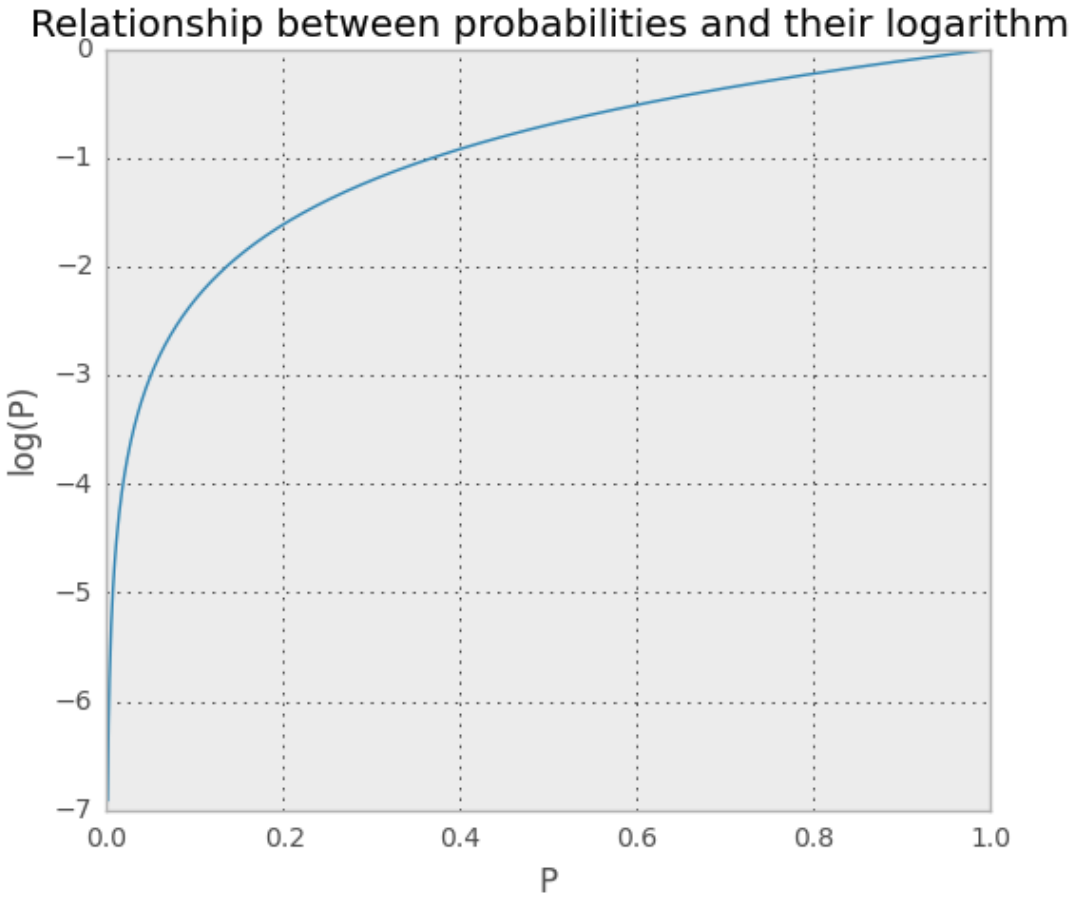
- primenimo na naš problem, dobićemo:

$$\log P(C) = \log P(F_1|C) + \log P(F_2|C)$$

- Verovatnće pripadaju intervalu [0, 1], a logaritam verovatnoća intervalu (-∞, 0] – v. str. 55.
- I u ovom slučaju su veći brojevi jači indikator ispravne klase (jedino što su sada negativni), tj:

$$P(C = "pos"|F_1, F_2) > P(C = "neg"|F_1, F_2) \rightarrow \log P(C = "pos"|F_1, F_2) > \log P(C = "neg"|F_1, F_2)$$

Rešenje problema obrade vrlo malih brojeva.



Rešenje problema obrade vrlo malih brojeva.

- Primenjujemo logaritam na jednačinu i dobijamo:

$$c_{best} = \underset{c \in C}{\operatorname{argmax}} P(C = c) \times \left(P(F_1|C = c) \times P(F_2|C = c)\right)$$

$$c_{best} = \underset{c \in C}{\operatorname{argmax}} (\log P(C = c) + \log P(F_1|C = c) + \log P(F_2|C = c))$$

- Generalizacija za k obeležja:

$$c_{best} = \underset{c \in C}{\operatorname{argmax}} \left(\log P(C = c) + \sum_{i=1}^{k} \log P(F_k|C = c) \right)$$

Binarna klasifikacija sentimenta

Vrste naivnih Bajesovih klasifikatora.

- Naivni Bajesovi klasifikatori nalaze se u paketu sklearn.naive_bayes.
- Postoje različite vrste klasifikatora: Gausov, multinomijalni i Bernulijev.
  - GaussianNB – pretpostavlja se normalna distribucija obeležja, što je ispravno za, npr. klasifikaciju pola osobe na osnovu visine i težine, ali ne i za klasifikaciju sentimenta tvitova (normalna distribucija se ne odnosi na brojanje reči).
  - MultinomialNB – pretpostavlja se da se obeležja odnose na broj pojava, što je za naš slučaj upotrebljivo.
  - BernoulliNB – sličan je multinomijalnom ali je pogodniji za binarne slučajeve (reč se pojavljuje ili ne) nego za broj pojavljivanja reči.

Binarna klasifikacija sentimenta

Vrste naivnih Bajesovih klasifikatora.

- Ukoliko malo bolje pogledamo podatke sa Tvitera, možemo da zaključimo da tvitovi nisu isključivo pozitivni ili negativni.
- Veliki broj tvitova ne sadrži nikakav sentiment, već su neutralni ili neznačajni, odnosno sadrže informacije tipa: „novi album benda Cannibal Corpse će biti objavljen sredinom jula meseca“.
- Drugim rečima, analiza sentimenta nije problem binarne klasifikacije zato što u izlaznom prostoru imamo četiri moguće vrednosti klasnog obeležja.
- Da bi, za prvo vreme, uprostili slučaj, koristićemo samo pozitivne i negativne tvitove.

Binarna klasifikacija sentimenta

Filtriranje podataka.

```
>>> # najpre kreiramo bulovsku listu koja sadrži true za
>>> # tvitove koji su ili pozitivni ili negativni
>>> pos_neg_idx = np.logical_or(Y=="positive", Y=="negative")
>>> # indeks koristimo za filtriranje podataka i labela
>>> X = X[pos_neg_idx]
>>> Y = Y[pos_neg_idx]
>>> # labele transformišemo u bulovske vrednosti
>>> Y = Y=="positive"
```

- Nakon filtriranja, X sadrži sirov tekst tvita, a Y klasu (0 za negativnu, odnosno 1 za pozitivnu).
- Upotrebom TfidfVectorizer-a izdvojićemo TF-IDF vrednosti obeležja koja nadalje koristimo za obuku klasifikatora.
- Klasom Pipeline vezujemo TfidfVectorizer i klasifikator.

Binarna klasifikacija sentimenta

Obučavanje modela.

```
from sklearn.feature_extraction.text import TfidfVectorizer
from sklearn.naive_bayes import MultinomialNB
from sklearn.pipeline import Pipeline
def create_ngram_model():
  tfidf_ngrams = TfidfVectorizer(ngram_range=(1, 3), analyzer="word", binary=False)
  clf = MultinomialNB()
  return Pipeline([('vect', tfidf_ngrams), ('clf', clf)])
```

- *Pipeline* instance koju vraća funkcija `create_ngram_model()` može se dalje koristiti za obuku klasifikatora.
- Podsetnik: konstruktor *pipeline*-a zahteva listu parova tipa (str, clf); svaki par odgovara jednom koraku u *pipeline*-u, pri čemu je prvi element svakog para string koji imenuje korak, dok je drugi element objekat koji izvršava transformaciju.

Binarna klasifikacija sentimenta

Obučavanje modela.

- Definisaćemo funkciju `train_model()` koja obučava klasifikator:

```
from sklearn.metrics import precision_recall_curve, auc
from sklearn.cross_validation import ShuffleSplit
def train_model(clf_factory, X, Y):
  cv = ShuffleSplit(n=len(X), n_iter=10, test_size=0.3, random_state=0)
  scores = []
  pr_scores = []
```

Obučavanje modela.

```
for train, test in cv:
  X_train, y_train = X[train], Y[train]
  X_test, y_test = X[test], Y[test]
  clf = clf_factory()
  clf.fit(X_train, y_train)
  train_score = clf.score(X_train, y_train)
  test_score = clf.score(X_test, y_test)
  scores.append(test_score)
  proba = clf.predict_proba(X_test)
  precision, recall, pr_thresholds = precision_recall_curve(y_test, proba[:,1])
  pr_scores.append(auc(recall, precision))
  summary = (np.mean(scores), np.std(scores), np.mean(pr_scores), np.std(pr_scores))
  print("%.3f\t%.3f\t%.3f\t%.3f" % summary)
```

Obučavanje modela – prvi model.

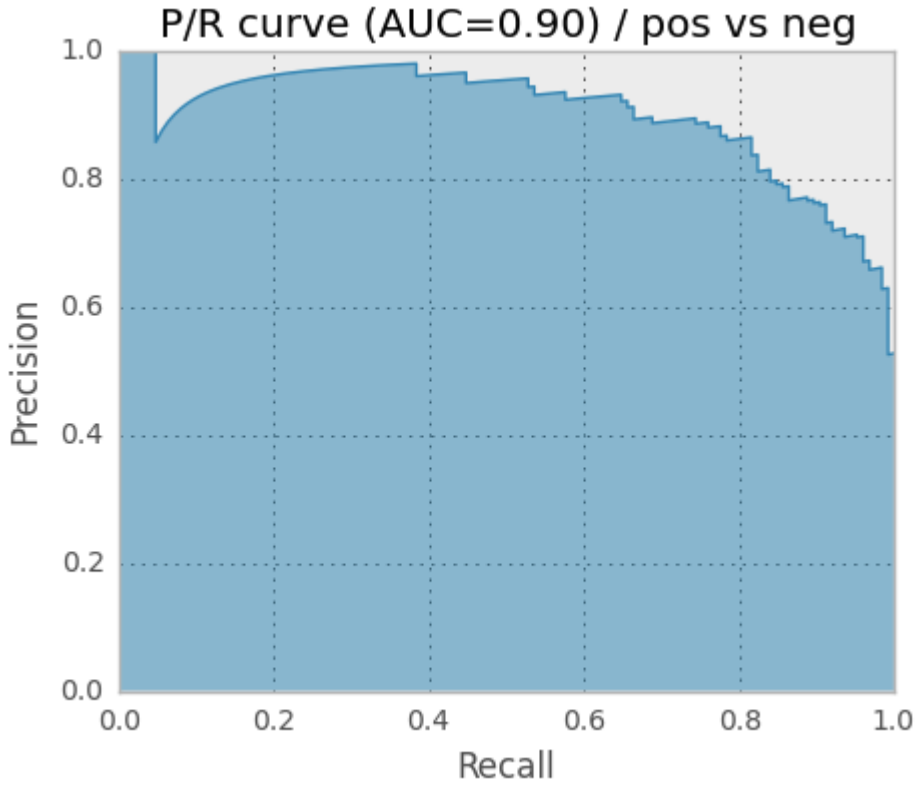
- Šta dobijamo kada obučimo prvi model?

```
>>> X, Y = load_sanders_data()
>>> pos_neg_idx = np.logical_or(Y=="positive", Y=="negative")
>>> X = X[pos_neg_idx]
>>> Y = Y[pos_neg_idx]
>>> Y = Y=="positive"
>>> train_model(create_ngram_model, X, Y)
0.788   0.024   0.882   0.036
```

- Obuka Naivnog Bajesa obeležjima zasnovanim na TF-IDF trigramima daje tačnost od 78,8% i P/R AUC (površina ispod krive za grafik odziv-preciznost) od 88,2%, što je dobar rezultat jer je u analizi sentimenta teško postići visoku tačnost.

Binarna klasifikacija sentimenta

Obučavanje modela – prvi model.



Upotreba svih klasa

Ponašanje klasifikatora.

- Pomoćna funkcija (za redefinisanje klasnih obeležja):

```
def tweak_labels(Y, pos_sent_list):
  pos = Y==pos_sent_list[0]
  for sent_label in pos_sent_list[1:]:
    pos |= Y==sent_label
  Y = np.zeros(Y.shape[0])
  Y[pos] = 1
  Y = Y.astype(int)
```

Upotreba svih klasa

Ponašanje klasifikatora.

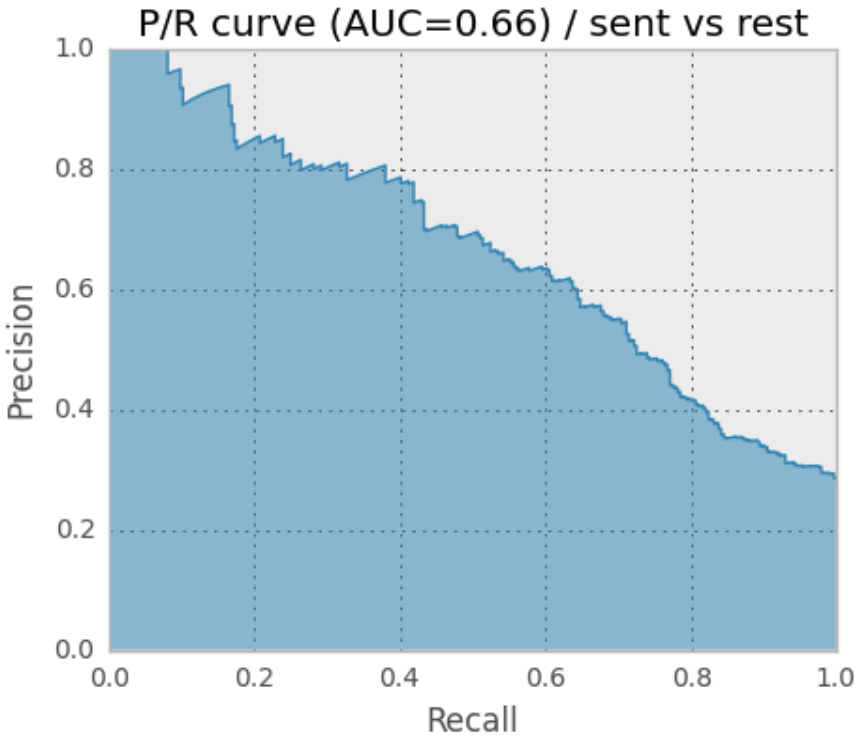
- Kako Naivni Bajes razdvaja tvitove koji sadrže sentiment od ostatka tvitova?

```
>>> Y = tweak_labels(Y, ["positive", "negative"])
# Y je sada 1 (pozitivna klasa) za sve tvitove koji su pozitivni ili negativni.
# Y je 0 za (negativna klasa) za sve tvitove koji su neutralni ili neznačajni.
>>> train_model(create_ngram_model, X, Y, plot=True)
0.750   0.012   0.659   0.023
```

Upotreba svih klasa

Ponašanje klasifikatora.

- Kako Naivni Bajes razdvaja tvitove koji sadrže sentiment od ostatka tvitova?



Upotreba svih klasa

Ponašanje klasifikatora.

- Kako Naivni Bajes razdvaja tvitove koji sadrže sentiment od ostatka tvitova?
- P/R AUC je pala na 66%, što je očekivano.
- Tačnost je i dalje visoka zato što je skup podataka nebalansiran (od 3.362 tvita, 920 su pozitivni ili negativni, što je oko 27% skupa).
- Drugim rečima, ako formiramo „klasifikator“ koji bi bilo koju instancu skupa klasifikovao kao tvit bez sentimenta, dostigli bi tačnost od 73%.
- Na osnovu toga zaključujemo da u slučaju nebalansiranih skupova podataka neophodno obratiti pažnju na odziv i preciznost.

Upotreba svih klasa

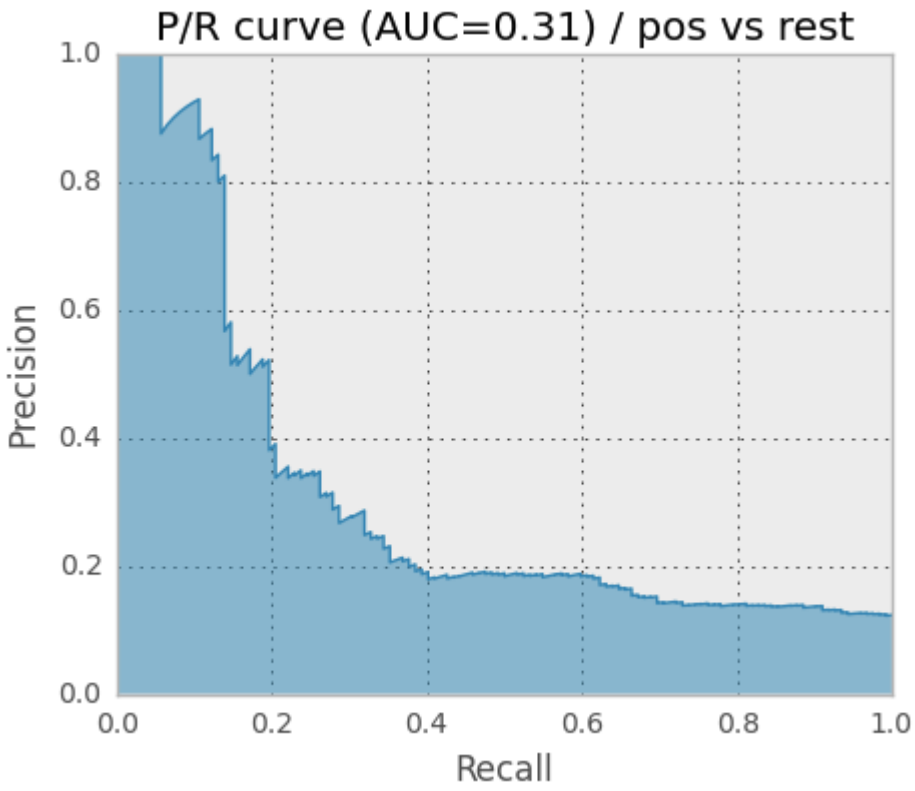
Ponašanje klasifikatora.

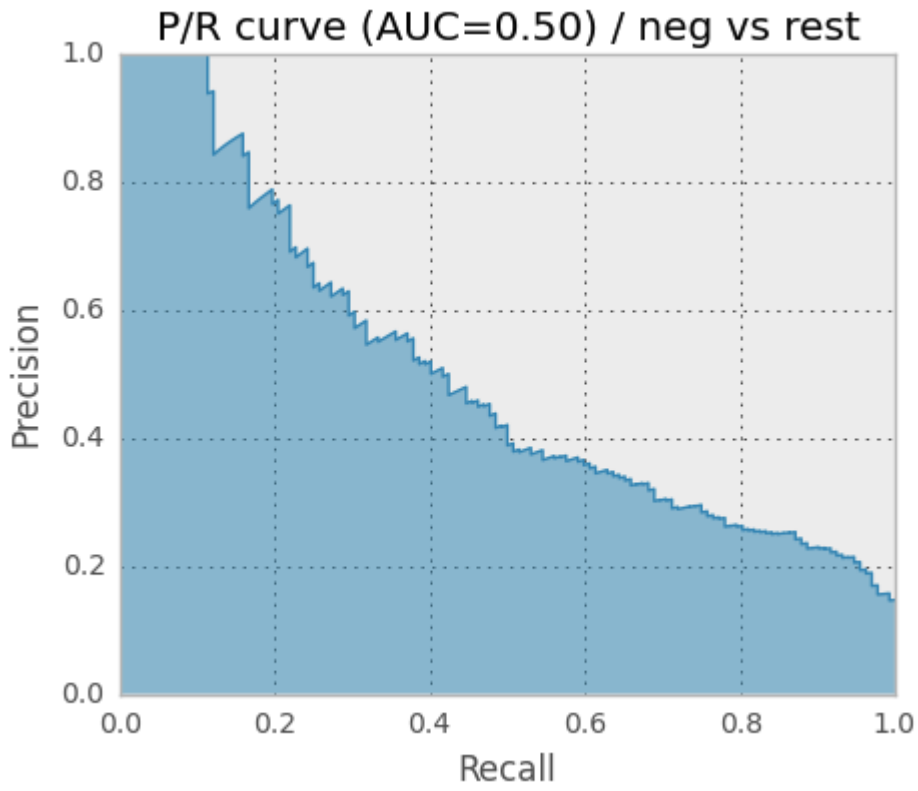
- Kako Naivni Bajes razdvaja pozitivne tvitove od ostatka tvitova?
- Kako Naivni Bajes razdvaja negativne tvitove od ostatke tvitova?
- Odgovor na oba pitanja je: izuzetno loše.
- Ukoliko pogledate P/R krive, videćete da je nemoguće naći kompromis između preciznosti i odziva.

```
== Pos vs. rest ==
0.873   0.009   0.305   0.026
== Neg vs. rest ==
0.861   0.006   0.497   0.026
```

Upotreba svih klasa

Ponašanje klasifikatora.





Upotreba svih klasa

Podešavanje parametara.

- Parametri koje možemo da podešavamo za TfidfVectorizer su:
  - podešavanja koja se odnose na n-grame – korišćenje unigrama (1,1), unigrama i bigrama (1,2), unigrama, bigrama i trigrama (1,3),
  - izmena vrednosti parametra koji se odnosi na tretiranje retkih reči,
  - izmena uticaja IDF na TF-IDF,
  - uklanjanje stop reči,
  - upotreba logaritma prilikom brojanja reči,
  - način brojanja reči – da li se reči broje ili se jednostavno proverava da li se pojavljuju ili ne.
- Parametar koje možemo da podešavamo za MultinomialNB je alfa parametar koji se odnosi na način uglađivanja (engl. *smoothing*): Laplasovo (1), Lidstonovo (0.01, 0.05, 0.1, or 0.5), bez uglađivanja – 0.

Upotreba svih klasa

Pretraga po rešetci (engl. *grid search*).

- Pretraga prostora parametara je česta aktivnost u mašinskom učenju.
  - Pretpostavite da imate dva parametra sa po tri moguće vrednosti, pri čemu za svaki par parametra klasifikator postiže različitu tačnost unakrsne validacije.
  - Cilj je naći optimalne vrednosti parametara za koje se postiže najveća tačnost.
  - U opštem slučaju, tačnost ne mora biti ciljna mera – umesto tačnosti se može, npr. koristiti i balansirana F-mera (koju ćemo koristiti u našem slučaju).

	P1 = 1	P1 = 2	P1 = 3
P2 = 1	80%	81%	**83%**
P2 = 2	75%	76%	77%
P2 = 3	69%	70%	71%

Upotreba svih klasa

Pretraga po rešetci (engl. *grid search*).

```
from sklearn.grid_search import GridSearchCV
from sklearn.metrics import f1_score
def grid_search_model(clf_factory, X, Y):
  cv = ShuffleSplit(n=len(X), n_iter=10, test_size=0.3,random_state=0)
  param_grid = dict(vect__ngram_range=[(1, 1), (1, 2), (1, 3)],
                    vect__min_df=[1, 2],
                    vect__stop_words=[None, "english"],
                    vect__smooth_idf=[False, True],
                    vect__use_idf=[False, True],
                    vect__sublinear_tf=[False, True],
                    vect__binary=[False, True],
                    clf__alpha=[0, 0.01, 0.05, 0.1, 0.5, 1])
```

Upotreba svih klasa

Pretraga po rešetci (engl. *grid search*).

```
  grid_search = GridSearchCV(clf_factory(),
                             param_grid=param_grid,
                             cv=cv,
                             score_func=f1_score,
                             verbose=10)
  grid_search.fit(X, Y)
  return grid_search.best_estimator_
```

- Nakon pokretanja funkcije treba „malo“ sačekati jer se ispituje 3×2×2×2×2×2×2 = 1152 mogućih kombinacija, pri čemu se za svaku izvodi 10-struka unakrsna validacija.

```
clf = grid_search_model(create_ngram_model, X, Y) # za f-ju create_ngram_model v. str. 58
print(clf)
```

Upotreba svih klasa

Pretraga po rešetci (engl. *grid search*).

- Optimalna kombinacija parametara dovodi do P/R AUC vrednosti 70,2% prilikom razdvajanja tvitova koji sadrže sentiment od ostatka tvitova.

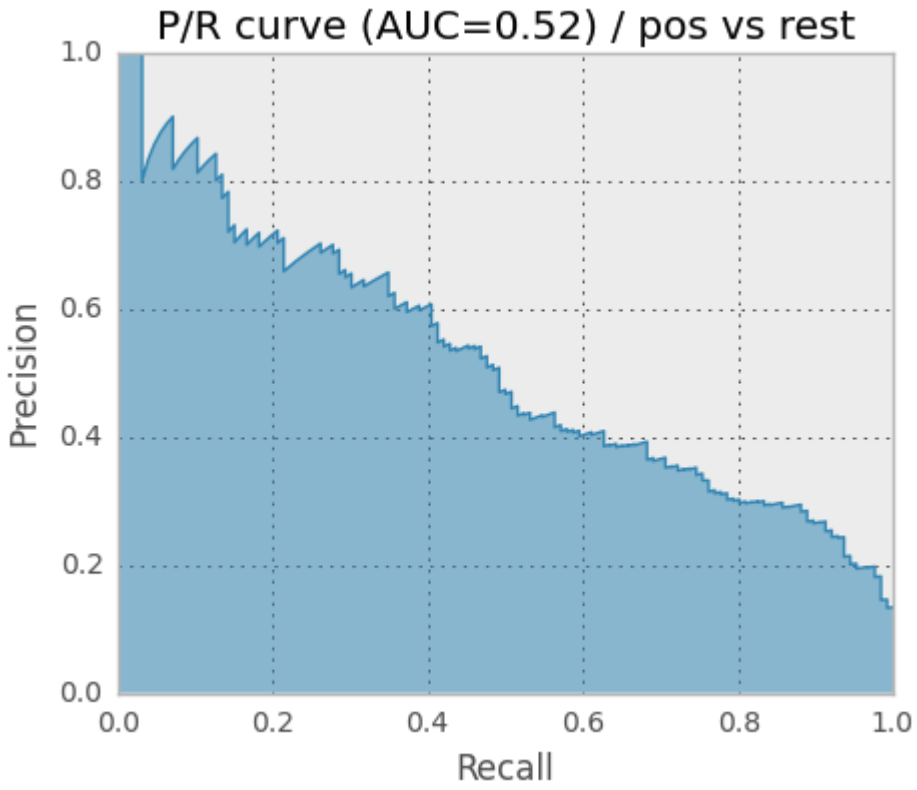
```
0.795   0.007   0.702   0.028
```

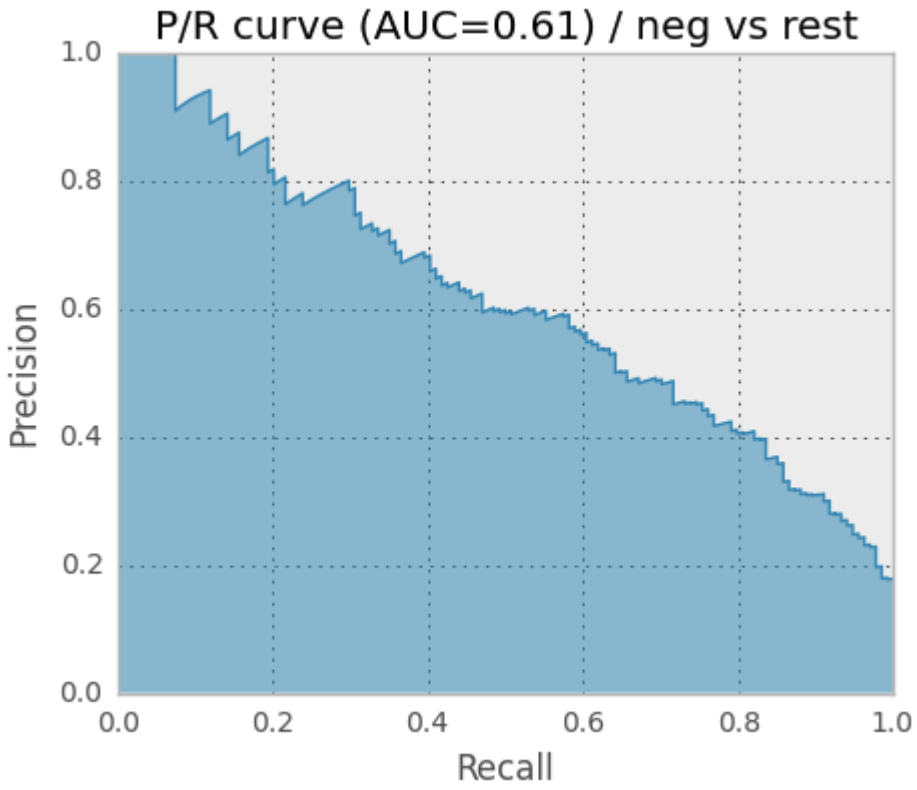
- Prilikom razdvaja pozitivnih tvitova od ostatka tvitova i negativnih tvitova od ostatka tvitova dobijamo sledeće rezultate:

```
== Pos vs. rest ==
0.889 0.010 0.509 0.041
== Neg vs. rest ==
0.886 0.007 0.615 0.035
```

Upotreba svih klasa

Ponašanje klasifikatora (sa optimalnim parametrima).





Upotreba svih klasa

Da li je klasifikator sada upotrebljiv?

- Iako P/R krive izgledaju bolje, ovakav klasifikator je i dalje neupotrebljiv.
- Zato primenjujemo dodatne operacije poput čišćenja tvitova, određivanja vrste reči, itd.

Čišćenje tvitova

Zašto je neophodno da očistimo tvitove?

- Pošto je dužina teksta ograničena na određeni broj karaktera, ljudi su osmislili „nove jezičke skraćenice“ (emotikone i neke druge skraćenice) kako bi ono što žele da kažu mogli reći sa što manjim brojem karaktera.
- Pošto smo do sada ovakve jezičke konstrukte ignorisali, postavljamo pitanje – da li možemo da povećamo performanse klasifikatora ukoliko ih uzmemo u obzir.
- Da bi smo to postigli potrebno je da napravimo predprocesor za TfidfVectorizer.

Čišćenje tvitova

Generisanje rečnika za zamenu emotikona.

```
emo_repl = {
  # pozitivni emotikoni
  "&lt;3": " good ", ":d": " good ", ":dd": " good ", "8)": " good ", ":-)": " good ",
  ":)": " good ", ";)": " good ", "(-:": " good ", "(:": " good ",
  # negativni emotikoni
  ":/": " bad ", ":&gt;": " sad ", ":')": " sad ", ":-(": " bad ", ":(": " bad ",
  ":S": " bad ", ":-S": " bad ",
}
# osigurati da se npr. :dd zameni pre :d
emo_repl_order = [k for (k_len,k) in reversed(sorted([(len(k),k) for k in emo_repl.keys()]))]
```

Čišćenje tvitova

Generisanje rečnika za zamenu skraćenica.

```
re_repl = {
  r"\br\b": "are", r"\bu\b": "you",
  r"\bhaha\b": "ha", r"\bhahaha\b": "ha",
  r"\bdon't\b": "do not", r"\bdoesn't\b": "does not",
  r"\bdidn't\b": "did not", r"\bhasn't\b": "has not",
  r"\bhaven't\b": "have not", r"\bhadn't\b": "had not",
  r"\bwon't\b": "will not", r"\bwouldn't\b": "would not",
  r"\bcan't\b": "can not", r"\bcannot\b": "can not",
}
```

Čišćenje tvitova

Uvođenje predprocesora.

```
def create_ngram_model(params=None):
  def preprocessor(tweet):
    tweet = tweet.lower()
    for k in emo_repl_order:
      tweet = tweet.replace(k, emo_repl[k])
      for r, repl in re_repl.items():
        tweet = re.sub(r, repl, tweet)
  return tweet
tfidf_ngrams = TfidfVectorizer(preprocessor=preprocessor, analyzer="word")
# ...
```

Čišćenje tvitova

Kako se ponaša klasifikator sa predprocesorom?

- Iako ima veliki broj skraćenica koje su korišćene, ovaj ograničeni rečnik za zamenu doveo je do sledećeg unapređenja klasifikacije (P/R AUC je 70,7% prilikom razdvajanja tvitova koji sadrže sentiment od ostatka tvitova, odnosno uvećan je za 0,5%):

```
== Pos vs. neg ==
0.808 0.024 0.885 0.029
== Pos/neg vs. irrelevant/neutral ==
0.793 0.010 0.685 0.024
== Pos vs. rest ==
0.890 0.011 0.517 0.041
== Neg vs. rest ==
0.886 0.006 0.624 0.033
```

Uzimanje vrste reči u obzir

Šta smo do sada radili?

- Do sada smo obučavali klasifikator pod predpostavkom da su reči međusobno nezavisne, odnosno pokušali smo da procenimo sentiment koristeći pristup vreće reči.
- Međutim, posmatranjem tvitova sa sentimentom i tvitova bez sentimenta možete uočiti da:
  - neutralni tvitovi uglavnom sadrže veći broj imenica, dok su
  - pozitivni i negativni tvitovi „šarenoliki“, odnosno da sadrže više prideva i glagola.
- Postavljamo sledeće pitanje: da li se ova lingvistička informacija može primeniti u analizi sentimenta tvitova?

Uzimanje vrste reči u obzir

Kako funkcioniše POS označavanje?

- POS označavanjem (engl. *part-of-speech tagging*, POS *tagging*) je raščlanjivanje cele rečenice u cilju formiranja stabla zavisnosti u kome svaki čvor odgovara datoj reči, a odnosi roditelj-dete određuju od koje reči data reč zavisi.
- Pomoću ovog stabla, moguće je odrediti da li je reč „book“ imenica ili glagol.
- Na primer,
  - reč „book je imenica u rečenici „This is a good book“,
  - a glagol u rečenici „Could you please book the flight?“

Uzimanje vrste reči u obzir

Kako funkcioniše POS označavanje?

- NLTK POS označavač za dati ulaz niza tokena vraća listu tuplea, u kojoj se svaki element sastoji od dela ulazne rečenice i dodeljenog POS taga.

```
>>> import nltk
>>> nltk.pos_tag(nltk.word_tokenize("This is a good book."))
[('This', 'DT'), ('is', 'VBZ'), ('a', 'DT'), ('good', 'JJ'), ('book','NN'), ('.', '.')]
>>> nltk.pos_tag(nltk.word_tokenize("Could you please book the flight?"))
[('Could', 'MD'), ('you', 'PRP'), ('please', 'VB'), ('book', 'NN'), ('the', 'DT'),
('flight', 'NN'), ('?', '.')]
```

- Koje su oznake od značaja za naš slučaj?

Uzimanje vrste reči u obzir

Kako funkcioniše POS označavanje?

- Za nas su od značaja sve koje počinju sa:
  - NN (imenice, engl. *nouns*),
  - VB (glagoli, engl. *verbs*),
  - JJ (pridevi, engl, *adjectives*),
  - RB (prilozi, engl. *adverbs*).
- Na primer:
  - NN (imenica, jednina) – *book*,
  - NNS (imenica, množina) – *books*,
  - NNP (vlastita imenica, jednina) – *Sean*,
  - NNPS (vlastita imenica, množina) – Vikings,
  - itd.

Šta je SentiWordNet?

- SentiWordNet je datoteka veličine 13MB dostupna na adresi http://sentiwordnet.isti.cnr.it koja sadrži veliki broj engleskih reči i njima dodeljene pozititvne i negativne vrednosti sentimenta.
- Datoteka sadrži nekoliko kolona pri čemu:
  - kolona POS omogućava da razlikujemo vrste reči (na primer, glagol „*book*“ i imenicu „*book*“),
  - PosScore and NegScore nam omogućuju da odredimo neutralnost reči kao: 1-PosScore-NegScore.
  - SynsetTerms je popis svih reči u skupu koje su sinonimi, dok
  - Kolone ID i Description možemo (u ovom slučaju) slobodno da ignorišemo.

Šta je SentiWordNet?

- Obratite pažnju na brojeve dodate rečima u koloni SynsetTerms.
- Neke reči mogu da imaju drugačija značenja u drugačijim kontekstima, što će za posledicu imati različite pozitivne i negativne vrednosti.

POS	ID	PosScore	NegScore	SynsetTerms	Description
v	01636859	0.375	0	fantasize#2 fantasise#2	Portray in the mind; "he is fantasizing the ideal wife"
v	01637368	0	0.125	fantasy#1 fantasize#1 fantasise#1	Indulge in fantasies; "he is fantasizing when he says he plans to start his own company"

Šta je SentiWordNet?

- Da bi odabrali odgovarajući Synset (odnosno reč) potrebno je da razumemo značenje tvita, što je posebna istraživačka oblast koja prevazilazi okvir ovog predavanja, tako da ćemo jednostavno računati srednju pozitivnu i srednju negativnu vrednost svih reči.
- Na primer, za reč „fantasize“:
  - PosScore će biti (0,375 + 0) / 2 = 0,1875,
  - NegScore će biti (0 + 0,125) / 2 = 0,0625.

Zaključne napomene

- Kod koji primenjuje do sada sve pobrojane metode prilikom konstrukcije klasifikatora za analizu sentimenta dat je u dodatku knjige.
- Na osnovu rezultata možemo da zaključimo da ne želimo da koristimo klasifikator koji razdvaja tvit sa pozitivnim sentimentom od ostalih ili tvit sa negativnim sentimentom odstalih, već da:
  - utvrdimo da li ima sentimenta u tvitu (pozitivan ili negativan / neutralan ili nebitan),
  - klasifikujemo sentiment, ukoliko ga ima, kao pozitivan, odnosno negativan.

```
== Pos vs. neg ==
0.810 0.023 0.890 0.025
== Pos/neg vs. irrelevant/neutral ==
0.791 0.007 0.691 0.022
== Pos vs. rest ==
0.890 0.011 0.529 0.035
== Neg vs. rest ==
0.883 0.007 0.617 0.033
```

Literatura

- Beleške pripremljene prema knjizi – Luis Pedro Coelho, Willi Richert (2015): „Building Machine Learning Systems with Python, Second Edition“. Packt Publishing.
- Dodatni materijali korišćeni prilikom pripreme beleški:
  - Olivera Grljević (2016): „Sentiment u sadržajima sa društvenih mreža kao instrument unapređenja poslovanja visokoškolskih institucija“, doktorska disertacija, Univerzitet u Novom Sadu.
  - Nikola Milikić (2016): „Naivni Bajes – klasifikacija“. Dostupno na sajtu Fakulteta organizacionih nauka, Univerziteta u Beogradu.

Hvala na pažnji

Pitanja su dobrodošla.