Istraživanje podataka

Vežbe 8

9. April 2021

Outline

Outline

Naivni Bajesovski klasifikatori

- Uslovna verovatnoća

$$P(C|A) = \frac{P(A,C)}{P(A)}$$

- Verovatnoću da se zajedno dese događaj A i događaj C možemo računati sa

$$P(A,C) = P(C|A) * P(A)$$

kao i sa

$$P(A,C) = P(A|C) * P(C)$$

Naivni Bajesovski klasifikatori

- Bajesova teorema

$$P(C|A) = \frac{P(A|C) * P(C)}{P(A)}$$

- Pretpostavka o nezavisnosti između atributa

$$P(C|A_1, A_2, ..., A_n) = \frac{\prod_{i=1}^{n} P(A_i|C) * P(C)}{P(A)}$$

- Određivanje klase

$$\widehat{C} = \arg\max_{C} \prod_{i=1}^{n} P(A_i|C) * P(C)$$

Zadatak 1

Dat je skup podataka:

Record	A	B	C	Class
1	0	0	0	+
2	0	0	1	−
3	0	1	1	−
4	0	1	1	−
5	0	0	1	+
6	1	0	1	+
7	1	0	1	−
8	1	0	1	−
9	1	1	1	+
10	1	0	1	+

Predvideti oznaku klase za test primer (A=0, B=1,C=0) koristeći naivan Bajesov pristup.

Zadatak 1

Da bismo odredili klasu test instance X, računamo uslovne verovatnoće $P(+|X)$ i $P(-|X)$.

$$P(+|X) = \frac{P(A = 0|+) * P(B = 1|+) * P(C = 0|+) * P(+)}{P(X)}$$

$$P(A = 0|+) = \frac{2}{5}$$

$$P(B = 1|+) = \frac{1}{5}$$

$$P(C = 0|+) = \frac{1}{5}$$

$$P(+) = \frac{1}{2}$$

Zadatak 1

$$P(+|X) = \frac{P(A = 0|+) * P(B = 1|+) * P(C = 0|+) * P(+)}{P(X)} =$$

$$\frac{\frac{2}{5} * \frac{1}{5} * \frac{1}{5} * \frac{1}{2}}{P(X)} =$$

$$\frac{\frac{1}{5^3}}{P(X)}$$

Zadatak 1

$$P(-|X) = \frac{P(A = 0|-) * P(B = 1|-) * P(C = 0|-) * P(-)}{P(X)}$$

$$P(A = 0|-) = \frac{3}{5}$$

$$P(B = 1|-) = \frac{2}{5}$$

$$P(C = 0|-) = 0$$

$$P(-) = \frac{1}{2}$$

Zadatak 1

$$P(-|X) = \frac{P(A = 0|-) * P(B = 1|-) * P(C = 0|-) * P(-)}{P(X)} =$$

$$\frac{\frac{3}{5} * \frac{2}{5} * 0 * \frac{1}{2}}{P(X)} = 0$$

Kako je $P(+|X) > P(-|X)$ test instancu X klasifikujemo klasom $+$.

Zadatak 2

Dati su podaci :

Boja	Veličina	Vrsta	Osoba	Naduvan
Žut	Mali	Duguljast	Odrasla	T
Žut	Mali	Duguljast	Dete	T
Žut	Mali	Okrugao	Dete	T
Ljubičast	Veliki	Okrugao	Odrasla	T
Žut	Veliki	Okrugao	Dete	F
Žut	Veliki	Duguljast	Dete	F
Ljubičast	Mali	Okrugao	Dete	F
Ljubičast	Veliki	Duguljast	Odrasla	F

Zadatak 2

Korišćenjem naivnog Bajesovog algoritma na osnovu prethodno datih podataka klasifikovati sledeće instance i izračunati preciznost.

Boja	Veličina	Vrsta	Osoba	Naduvan
Ljubičast	Mali	Okrugao	Odrasla	T
Žut	Mali	Okrugao	Odrasla	T
Ljubičast	Veliki	Okrugao	Dete	T
Ljubičast	Veliki	Duguljast	Odrasla	F

Zadatak 2

Izračunate verovatnoće na osnovu trening skupa koje su potrebne za klasifikaciju test instanci:

X	$P(X\|T)$	$P(X\|F)$
Boja=Ljubičast	$\frac{1}{4}$	$\frac{1}{2}$
Boja=Žut	$\frac{3}{4}$	$\frac{1}{2}$
Veličina=Mali	$\frac{3}{4}$	$\frac{1}{4}$
Veličina=Veliki	$\frac{1}{4}$	$\frac{3}{4}$
Vrsta=Duguljast	$\frac{1}{2}$	$\frac{1}{2}$
Vrsta=Okrugao	$\frac{1}{2}$	$\frac{1}{2}$
Osoba=Odrasla	$\frac{1}{2}$	$\frac{1}{4}$
Osoba=Dete	$\frac{1}{2}$	$\frac{3}{4}$

$P(T) = \frac{1}{2}$
$P(F) = \frac{1}{2}$

Zadatak 2

Klasifikacija test instanci:

$X_1 = (Boja = Ljubicast, Velicina = Mali, Vrsta = Okrugao, Osoba = Odrasla)$

$$P(T|X_1) = \frac{P(Ljubicast|T)*P(Mali|T)*P(Okrugao|T)*P(Odrasla|T)*P(T)}{P(X_1)} = \frac{\frac{1}{4}*\frac{3}{4}*\frac{1}{2}*\frac{1}{2}*\frac{1}{2}}{P(X_1)} = \frac{\frac{3}{2^7}}{P(X_1)}$$

$$P(F|X_1) = \frac{P(Ljubicast|F)*P(Mali|F)*P(Okrugao|F)*P(Odrasla|F)*P(F)}{P(X_1)} = \frac{\frac{1}{2}*\frac{1}{4}*\frac{1}{2}*\frac{1}{4}*\frac{1}{2}}{P(X_1)} = \frac{\frac{1}{2^7}}{P(X_1)}$$

Kako je $P(T|X_1) > P(F|X_1)$, instancu X_1 klasifikujemo klasom T.

Zadatak 2

$X_2 = (Boja = Zut, Velicina = Mali, Vrsta = Okrugao, Osoba = Odrasla)$

$$P(T|X_2) = \frac{P(Zut|T)*P(Mali|T)*P(Okrugao|T)*P(Odrasla|T)*P(T)}{P(X_2)} = \frac{\frac{3}{4}*\frac{3}{4}*\frac{1}{2}*\frac{1}{2}*\frac{1}{2}}{P(X_2)} = \frac{\frac{9}{2^7}}{P(X_2)}$$

$$P(F|X_2) = \frac{P(Zut|F)*P(Mali|F)*P(Okrugao|F)*P(Odrasla|F)*P(F)}{P(X_2)} = \frac{\frac{1}{2}*\frac{1}{4}*\frac{1}{2}*\frac{1}{4}*\frac{1}{2}}{P(X_2)} = \frac{\frac{1}{2^7}}{P(X_2)}$$

Kako je $P(T|X_2) > P(F|X_2)$, instancu X_2 klasifikujemo klasom T.

Zadatak 2

$X_3 = (Boja = Ljubicast, Velicina = Veliki, Vrsta = Okrugao, Osoba = Dete)$

$$P(T|X_3) = \frac{P(Ljubicast|T)*P(Veliki|T)*P(Okrugao|T)*P(Dete|T)*P(T)}{P(X_3)} = \frac{\frac{1}{4}*\frac{1}{4}*\frac{1}{2}*\frac{1}{2}*\frac{1}{2}}{P(X_3)} = \frac{\frac{1}{2^7}}{P(X_3)}$$

$$P(F|X_3) = \frac{P(Ljubicast|F)*P(Veliki|F)*P(Okrugao|F)*P(Dete|F)*P(F)}{P(X_3)} = \frac{\frac{1}{2}*\frac{3}{4}*\frac{1}{2}*\frac{3}{4}*\frac{1}{2}}{P(X_3)} = \frac{\frac{9}{2^7}}{P(X_3)}$$

Kako je $P(T|X_3) < P(F|X_3)$, instancu X_3 klasifikujemo klasom F.

Zadatak 2

$X_4 = (Boja = Ljubicast, Velicina = Veliki, Vrsta = Duguljast, Osoba = Odrasla)$

$P(T|X_4) = \frac{P(Ljubicast|T)*P(Veliki|T)*P(Duguljast|T)*P(Odrasla|T)*P(T)}{P(X_4)} = \frac{\frac{1}{4}*\frac{1}{4}*\frac{1}{2}*\frac{1}{2}*\frac{1}{2}}{P(X_4)} = \frac{\frac{1}{2^7}}{P(X_4)}$

$P(F|X_4) = \frac{P(Ljubicast|F)*P(Veliki|F)*P(Duguljast|F)*P(Odrasla|F)*P(F)}{P(X_4)} = \frac{\frac{1}{2}*\frac{3}{4}*\frac{1}{2}*\frac{1}{4}*\frac{1}{2}}{P(X_4)} = \frac{\frac{3}{2^7}}{P(X_4)}$

Kako je $P(T|X_4) < P(F|X_4)$, instancu X_4 klasifikujemo klasom F.

Zadatak 2

Preciznost izračunata na test instancama je $\frac{3}{4}$ jer su 3 instance (X_1, X_2 i X_4) od 4 dobro klasifikovane.

Naivni Bajesovski klasifikator za kategoričke atribute

- *sklearn.naive_bayes.CategoricalNB*
  Verovatnoća za vrednost t u atributu x_i za klasu c se procenjuje sa:

$$P(x_i = t|y = c) = \frac{N_{tic} + \alpha}{N_c + \alpha * n_i}$$

gde je

- N_{tic} broj pojavljivanja vrednosti t u atributu x_i u instancama klase c
- N_c broj instanci klase c
- α - parametar za uglađivanje (default=1.0)
- n_i - broj različitih vrednosti u atributu x_i

Podrazumeva se da atributi imaju vrednosti $(0, 1, ..., n_i - 1)$.

Naivni Bajesovski klasifikator za kategoričke atribute

- parametri
  - *alpha* - parametar za uglađivanje (default=1.0)
- neki atributi
  - *category_count_*: sadrži podatak o broju pojavljivanja svake vrednosti atributa u svakoj klasi
  - *class_count_* - broj instanci u svakoj klasi

Naivni Bajesovski klasifikator za kategoričke atribute

- neke metode
  - *fit(x, y)* - pravi model
  - *predict(x)* - određuje klase instancama
  - *predict_proba(X)* - vraća procenjenu verovatnoću za test instance

Kodiranje kategoričkog atributa u numerički

- *sklearn.preprocessing.OrdinalEncoder*
  Kodiranje kategoričkog atributa u numeričke vrednosti $(0, 1, ..., n_i - 1)$ gde je n_i broj različitih vrednosti kategoričkog atributa.

Kodiranje kategoričkog atributa u numerički

- parametri
  - *categories* - kategorije atributa skupa
    - *auto*: određuju se na osnovu trening skupa (default)
    - lista čiji elementi su liste kategorija za svaki atribut skupa
  - *handle_unknown* - kako postupiti sa vrednostima koje se pojave u skupu za transformaciju, a nisu u listi kateforija
    - *error* - prijavljuje se greška (default)
    - *use_encoded_value* - dodeljuje se vrednost zadata parametrom *unknown_value*

Kodiranje kategoričkog atributa u numerički

- metode
  - *fit* - određuje vrednosti za transformaciju
  - *fit_transform* - određuje vrednosti za transformaciju i transformiše skup zadat kao argument
  - *transform* - transformiše skup zadat kao argument

Outline

Term-matrica

Podaci o dokumentima se predstavljaju sa term-matricom: atributi su termi (reči); broj atributa je veličina rečnika. Za svaki dokument i svaki term (reč) čuva se broj pojavljivanja tog terma u tom dokumentu. Umesto broja pojavljivanja terma može da se koristi td-idf mera.

Term-matrica

Primer term-matrice:

	term-matrica					
tekst	beijing	chinese	japan	macao	shanghai	tokyo
Chinese Beijing Chinese	1	2	0	0	0	0
Chinese Chinese Shanghai	0	2	0	0	1	0
Chinese Macao	0	1	0	1	0	0
Tokyo Japan Chinese	0	1	1	0	0	1

Term-matrica

Umesto broja pojavljivanja terma može da se koristi *tf* – *idf* (*term-frequency* - *inverse document frequency*) mera u kojoj je

- *tf* - frekvencija reči (*term-frequency*)
- *idf* - inverzna frekvencija dokumenta (*inverse document frequency*) je težina kojom se određuje značajnost terma u kolekciji tekstualnih dokumenata

Term-matrica

Ako su:

- t - term
- d - dokument
- n - ukupan broj dokumenata
- $df(t)$ - broj dokumenata koji sadrže term t

formule za $tf - idf$ meru i idf su:

$$tf - idf(t, d) = tf(t, d) * idf(t)$$

$$idf(t) = \log[n/df(t)] + 1$$

Naivni Bajes za klasifikaciju teksta

- Za klasifikaciju teksta koristiti se varijanta naivnog Bajesa - multinomijalni naivni Bajes
- Svakoj klasi c se na osnovu trening skupa dodeljuje vektor parametara $\Theta_c = (\Theta_{c1}, \Theta_{c2}, ..., \Theta_{cn})$, gde je n broj terma (atributa), a Θ_{ci} verovatnoća da se term i pojavi u instanci koja pripada klasi c.
- $\Theta_{ci} = \frac{N_{ci}+\alpha}{N_c+\alpha*n}$
  - N_{ci} broj pojavljivanja terma (reči) i u dokumentima klase c
  - N_c ukupan broj pojavljivanja svih reči u klasi c

Naivni Bajesovski klasifikatori

- klasifikacija dokumenta d sa termina $\langle t_1, t_2, \ldots t_{nd} \rangle$

$$\widehat{c} = \arg\max_c P(c) \prod_{i=1}^{nd} P(t_i|c) = \arg\max_c P(c) \prod_{i=1}^{nd} \Theta_{ci}$$

Zadatak 3

Dati su podaci :

Id teksta	reči u dokumentu	klasa
1̌	Chinese Beijing Chinese	yes
2̌	Chinese Chinese Shanghai	yes
3̌	Chinese Macao	yes
4̌	Tokyo Japan Chinese	no

Primenom naivnog Bajesa za klasifikaciju teksta klasifikovati tekst *Chinese Chinese Chinese Tokyo Japan* ako je $\alpha = 1$.

Zadatak 3

Verovatnoće za klase u trening skupu su:
$P(yes) = \frac{3}{4}$
$P(no) = \frac{1}{4}$
Podaci potrebni za određivanje verovatoće terma u klasi ($\Theta_{ci} = \frac{N_{ci}+\alpha}{N_c+\alpha*n}$):

- broj različitih reči koji se javlja u celom skupu:6
- broj reči u tekstovima klase *yes*: 8
- broj reči u tekstovima klase *no*: 3

Zadatak 3

Verovatnoće za reči u klasi *yes* u trening skupu:

$P(Chinese|yes) = \frac{5+1}{8+6} = \frac{6}{14} = \frac{3}{7}$

$P(Tokyo|yes) = \frac{0+1}{8+6} = \frac{1}{14}$

$P(Japan|yes) = \frac{0+1}{8+6} = \frac{1}{14}$

Verovatnoća da instanca pripada klasi *yes*:

$P(yes|X) = P(yes) * P(Chinese|yes)^3 * P(Tokyo|yes) *$

$P(Japan|yes) = \frac{3}{4} * \frac{3}{7}^3 * \frac{1}{14} * \frac{1}{14} \approx 0,0003$

Zadatak 3

Verovatnoće za reči u klasi *no* su:

$P(Chinese|no) = \frac{1+1}{3+6} = \frac{2}{9}$

$P(Tokyo|no) = \frac{1+1}{3+6} = \frac{2}{9}$

$P(Japan|no) = \frac{1+1}{3+6} = \frac{2}{9}$

Verovatnoća da instanca pripada klasi *no*:

$P(no|X) = P(no) * P(Chinese|no)^3 * P(Tokyo|no) * P(Japan|no) =$

$\frac{1}{4} * \frac{2}{9}^3 * \frac{2}{9} * \frac{2}{9} \approx 0,0001$

Zadatak 3

Kako je $P(yes|X) > P(no|X)$ test instanci dodeljujemo klasu *yes*.

Izdvajanje terma iz teksta

- *sklearn.feature_extraction.text*
  - *CountVectorizer* - pretvara kolekciju tekst dokumenata u term-matricu sa brojem pojavljivanja terma u dokumentu
  - *TfidfVectorizer(* - pretvara kolekciju tekst dokumenata u matricu sa *tf-idf* atributima
  - *TfidfTransformer* - pretvara matricu sa brojem pojavljivanja u matricu sa *tf-idf* atributima

Izdvajanje terma iz teksta

tf-idf : *term-frequency - inverse document frequency*

- *tf* - term-frequency (u klasifikacija teksta: frekvencija reči)
- *tf-idf* je *tf*idf*, smanjuje uticaj terma koji se često javlja u datom korpusu
- formula

$$tf - idf(t,d) = tf(t,d) * idf(t)$$

$$idf(t) = \log[n/df(t)] + 1$$

  - t - term
  - d - dokument
  - n - ukupan broj dokumenata
  - $df(t)$ - broj dokumenata koji sadrže term t
  - zbog 1 u $idf(t)$ term koji se javlja u svim dokumentima neće u potpunosti biti ignosrisan

Izdvajanje terma iz teksta

- parametri (*TfidfVectorizer* i *CountVectorizer*)
  - *input* - šta je ulaz (‘filename‘, ‘file‘, ‘content‘) (default=‘content‘)
  - *lowercase* - sva slova će biti pretvorena u mala pre izdvajanja (default=True)
  - *stop_words* - reči koje će biti uklonjene (default=‘english‘)
  - *max_df* - ignoriše reči koje imaju dokument-frekvenciju iznad zadatog praga (zadaje se procenat ili broj dokumenata) (default=1.0)
  - *min_df* -gnoriše reči koje imaju dokument-frekvenciju ispod zadatog praga (zadaje se procenat ili broj dokumenata) (default=1.0)
  - *binary* - term koji se javlja u dokumentu ima vrednos 1 umesto broja pojavljivanja (default=False)

Izdvajanje terma iz teksta

- parametri (*TfidfVectorizer* i *TfidfTransformer*)
  - *use_idf* -(default=True)
- atributi (*TfidfVectorizer* i *CountVectorizer*)
  - *vocabulary_* - rečnik sa rečima kao ključevima, a vrednosti su indeksi odgovarajućih atributa

Izdvajanje terma iz teksta

- metode
  - *fit* - uči rečnik
  - *fit_transform* - uči rečnik i vraća term matricu
  - *transform* - pretvara tekstove u term matricu
  - *get_feature_names* - vraća imena atributa (*TfidfVectorizer* i *CountVectorizer*)
  - *get_stop_words* - vraća stop reči (*TfidfVectorizer* i *CountVectorizer*)

Izdvajanje terma iz rečnika

- *sklearn.feature_extraction.DictVectorizer* - transformiše rečnike sa podacima u obliku atribut-vrednost u vektore.
  - kada su vrednosti atributa niske primenjuje se kodiranje *jedan-od-c*
  - metode
    - *fit*
    - *fit_transform*
    - *get_feature_names*
    - transform

Multinomijalni naivni Bajes

- *sklearn.naive_bayes.MultinomialNB*
  - parametri
    - *alpha* - (default=1.0)
    - *fit_prior* - da li uči verovatnoće klasa (default=True)
    - *class_prior* - verovatnoće klasa (default=None)

Naivni Bajesovski klasifikatori

- atributi
  - *class_count_* - broj instanci po klasama
  - *feature_count_* - broj terma po klasi

Naivni Bajesovski klasifikatori

- metode
  - *fit(x, y)* - pravi model
  - *predict(x)* - određuje klase instancama
  - *predict_proba(X)* - vraća procenjenu verovatnoću za test instance

Outline

Obrada teksta na engleskom jeziku

1. eliminacija stop reči
2. svođenje reči na koren pomoću Porterovog stemera `https://tartarus.org/martin/PorterStemmer/`
3. pravljenje term matrice