# Darbo užduotis

Sukurti programą SPAMui klasifikuoti panaudojant Bajeso teoremą. Ištirti priklausomybę tarp programoje naudojamų nustatymų ir klasifikatoriaus darbo efektyvumo (*žr.reikalavimus ataskaitai*). Programavimo kalba pasirenkama laisvai.

**Ataskaitoje turi būti pateikta:**

[ ] atlikti kryžminės patikros eksperimentai (segmentų N skaičius - 10);   
 apskaičiuotas vidutinis visų 10 eksperimentų tikslumas

[ ] klasifikatorius vaizduoja testuojamo failo(-ų) spamiškumo simbolinę ir skaitinę įvertį (pvz. 97%, Spamas)

[ ] ataskaitoje pateikta grafikų pagalba priklausomybės tarp programoje naudojamų nustatymų ir   
 klasifikatoriaus darbo efektyvumo

{ } leksemos, sutinkamos pirmą kartą, spamiškumo tikimybės įverčio pokyčio (pvz. 0.45 ; 0.40 ; 0.35 )   
 įtaka į klasifikatoriaus darbo tikslumą\*

{ } iš analizuojamo failo pasirenkamų leksemų skaičiaus N reikšmių (pvz. 8, 16, 20) pokyčio įtaka į   
 klasifikatoriaus darbo tikslumą\*

{ } spamiškumo slenksčio reikšmių pokyčio įtaka į klasifikatoriaus darbo tikslumą\*.

(\*) klasifikatoriaus darbo tikslumas turi būti vertinamas procentais skaičiuojant true positive ir false positive reikšmių vidutinę reikšmę, čia

1. true positive - vidutinė reikšmė, įvertinanti santykius: nespamas klasifikuotas kaip nespamas ir spamas klasifikuotas kaip spamas;
2. false positive - vidutinė reikšmė, įvertinanti santykius: nespamas klasifikuotas kaip spamas ir spamas klasifikuotas kaip nespamas.

Pateikiant priklausomybes grafikų pagalba vaizduoti true positive ir false positive reikšmių sandarą (kiekvienai dedamąjai, kuri buvo aprašyta aukščiau, turi atitikti atskira kreivė) taip pat grafike kartu pateikti kreivę, kuri atitinka bendrą klasifikatoriaus darbo tikslumą.

**Darbo vykdymo rekomendacijos**

Skaičiuojant kiekvienos leksemos (simbolių seka iš a..Z, 0..9, $, ', "; Visi kiti simboliai - yra skyrikliai tarp leksemų) pasirodymų skaičių kiekviename duomenų rinkinyje patartina naudoti *hash* lenteles.

**Šaltiniai:**

1. <http://www.paulgraham.com/spam.html>
2. <http://en.wikipedia.org/wiki/Bayesian_spam_filtering>

# Darbo atlikimo schema (viena iš galimų)

1. Sudaromi 2 failų katalogai su SPAMu ir NESPAMu (tai apsimokymo duomenys). Tarkime, SPAM katalogo failuose yra 300 žodžių, NESPAM – 250 žodžių.
2. Sudaroma duomenų struktūra (pvz. *hash* lentelė), į kurią įrašomi kiekvienos leksemos pasirodymo kiekiai SPAM ir NESPAM failuose, pvz. leksema „africa“ apsimokymo duomenyse – SPAM kataloge sutinkama 50 kartų, NESPAM kataloge – 15 kartų; leksema „earn“ – 200 ir 33 kartų atitinkamai

|  |  |  |
| --- | --- | --- |
|  | SPAM | NESPAM |
| africa | 50 | 15 |
| earn | 200 | 33 |

1. Apskaičiuojama kiekvienos leksemos spamiškumo tikimybė, t.y. tikimybę, kad failas su šiuo žodžiu yra SPAMas. Naudojama formulė:



čia P(W|S) – tikimybė, kad leksema W yra SPAMe, P(W|H) – tikimybė, kad leksema W yra NESPAMe.

Mūsų atveju:

|  |  |  |  |
| --- | --- | --- | --- |
| W | P(W|S) | P(W|H) | P(S|W) |
| africa |  |  |  |
| earn |  |  |  |

Apsimokymo duomenų leksemų spamiškumo tikimybes P(S|W) tikslinga saugoti atskirai, bet ne perskaičiuoti kiekvieną kartą, kai reikia atlikti failo SPAMo klasifikacijos analizę.

1. Pateiktas SPAM klasifikacijai naujo failo turinys suskaidomas į leksemas. Nustatoma kiekvienos leksemos spamiškumo tikimybė (žr. ankstesnį punktą). Žodžiams, kurie sutinkami primą kartą, priskiriama spamiškumo tikimybė – 0.4.  
   Pvz.: naujo failo, pateikto analizei, turinys:   africa earn zzz.
2. Iš analizuojamo failo pasirenkamas tam tikras leksemų skaičius *N* (pvz. 15-20), kurių spamiškumo tikimybės yra maksimaliai nutolusios **nuo** neutralios (pvz. 0.5)
3. Įvertinama tikimybė, kad failo pasirinktos leksemos rodo į jo priklausomumą SPAMui. Naudojama formulė

,

čia *pi* – pasirinktos leksemos spamiškumo tikimybė. Mūsų atveju, iš failo išrenkame 2   
leksemas – „africa“ ir „earn“:



**Pastabos:** Jei turime situacija

|  |  |  |
| --- | --- | --- |
|  | SPAM | NESPAM |
| top | 11 | 0 |
| bottom | 0 | 44 |

Tuomet šių leksemų spamiškumas įvertinamas

|  |  |  |  |
| --- | --- | --- | --- |
| W | P(W|S) | P(W|H) | P(S|W) |
| top |  |  | 0.99, jei P(W|H)=0 |
| bottom |  |  | 0.01, jei P(W|S)=0 |