BME-VIK Méréstechnikai és Információs Rendszerek Tanszék
Intelligens rendszerek szakirány

EGÉSZSÉGÜGYI KÓDOLÁSTÁMOGATÓ RENDSZER FEJLESZTÉSE

BSc szakdolgozat

Készítette: Sárándi István
Külső konzulens: Héja Gergely (GYEMSZI)
Tanszéki konzulens: Strausz György

2012. január 5.

Tartalom

- Bevezető
- Osztályozó módszerek
- Implementáció
- Eredmények, további lehetőségek

BEVEZETŐ

A problémáról

- Egészségügyi kódolás
  - Betegségek formalizált ábrázolására
  - Pl. BNO-rendszer
  - Diagnózisok, zárójelentések kódolása
    - Statisztika, finanszírozás
- Automatizálás igénye
  - Manuálisan lassú, drága
  - Legalább részben automatizálni

Mi automatizálható?

- Teljes automatizálás nehéz
  - Szakértők is tévednek
- Kevesebb kódnak kelljen manuálisan utánanézni
  1. Felhasználó begépeli a diagnózist
  2. A rendszer visszaad egy tipplistát
  3. A felhasználó megnézi a talált kódok leírását és dönt
     - **Ehhez szakértelem szükséges!**
- Felhasználó–szolgáltatás kommunikáció
  - Webes felület

Példa

`fibrillatio auricula paroxysmalis`

Kód	Bizonyosság
I48H0	0,762
I4710	0,144
I4900	0,048
I5130	0,027
I4990	0,019

- Bizonyosságértékek (relevancia)
  - Mivel érdemes kezdeni
  - Mennyire biztos magában a rendszer
  - Nem feltétlenül valószínűség (%)

Megközelítési módok

- Szabályalapú
  - Elkészítéséhez tárgyterületi tudás kell
- Természetesnyelv-feldolgozás (NLP)
  - Morfológiai elemzés
    - Előfeldolgozás
  - Szintaktikai elemzés itt nem releváns
  - Szemantika
    - Ontológiákkal
- **Gépi tanulás, statisztikai osztályozás**
  - Nem kell tárgyterületi szaktudás
  - Minták kellenek
  - Számításigényes
    - De egyre gyorsabb gépek

OSZTÁLYOZÓ MÓDSZEREK

Gépi tanulás, statisztika

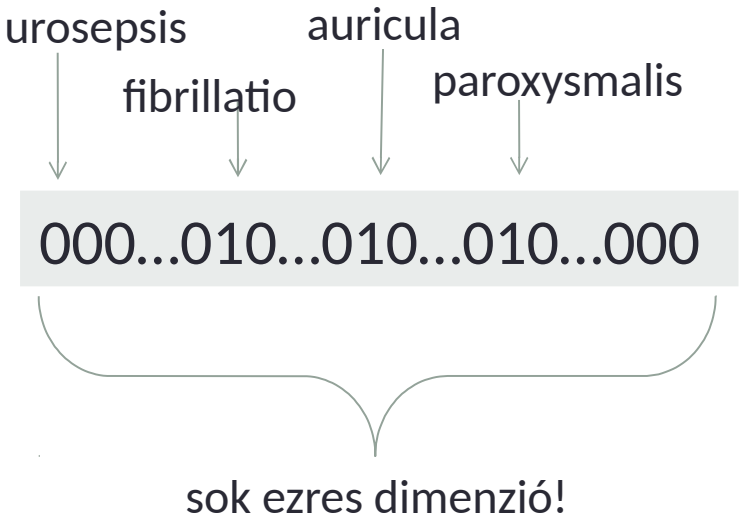
- Vektortér
- Naiv Bayes-háló
- Neurális hálózatok
  - Többrétegű perceptron
- SVM
- Kevert módszerek

Vektorizálás

- Vektorbemenet kell
- Szóhalmaz-modell (*bag of words*)
  - Elvész információ
    - Sorrend
      - Diagnózisoknál nem lényeges
    - Szóhasonlóságok (pl. morfológia, szinonimák)
      - Előfeldolgozás

Példa

„fibrillatio aricula paroxysmalis”



Vektortér

- Folytonossági hipotézis
- Összehasonlítás sokdimenziós vektortérben
  - Koszinuszos hasonlóságmérték
    - $cosSim(\boldsymbol{v}, \boldsymbol{w}) = \frac{\boldsymbol{v} \cdot \boldsymbol{w}}{\|\boldsymbol{v}\| \cdot \|\boldsymbol{w}\|}$
- Gondok
  - Sok összehasonlítás, nem hatékony
    - Invertált index (96%-kal kevesebb összehasonlítás)
  - Azonosan kezeli a dimenziókat (szavakat)
    - Súlyozás kell

IDF-súlyozás

- „inverse document frequency”
- Gyakran szereplő szó kevésbé fontos
- Információelméletileg is alátámasztható

$$\boldsymbol{v'}_j = \boldsymbol{v}_j \cdot \log\left(\frac{|D|}{\left|\{\boldsymbol{w} \in D \mid \boldsymbol{w}_j = 1\}\right|}\right)$$

Bayes

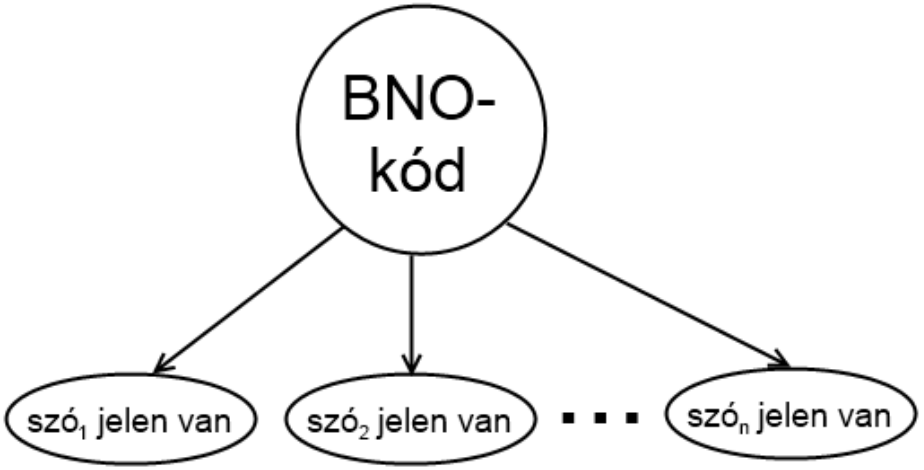
- Relevancia: feltételes valószínűség

$$p(C_i|\boldsymbol{v}) = p(\boldsymbol{v}|C_i) \cdot \frac{p(C_i)}{p(\boldsymbol{v})}$$

- feltételes együttes eloszlás becslése mintákból
  - $p(\boldsymbol{v}|C_i)$ feltételes együttes eloszlás becslése mintákból
  - Rengeteg tárolandó paraméter → gyenge becslés
- Bayes-háló
  - Kompaktabb leírás feltételes függetlenségek fennállásakor
- Naív Bayes
  - Minden attribútum (szó) páronként feltételesen független, ismerve a célváltozót (BNO-kód)

Naiv Bayes

- $p(\boldsymbol{v}|C_i) = \prod_j p(v_j|C_i)$

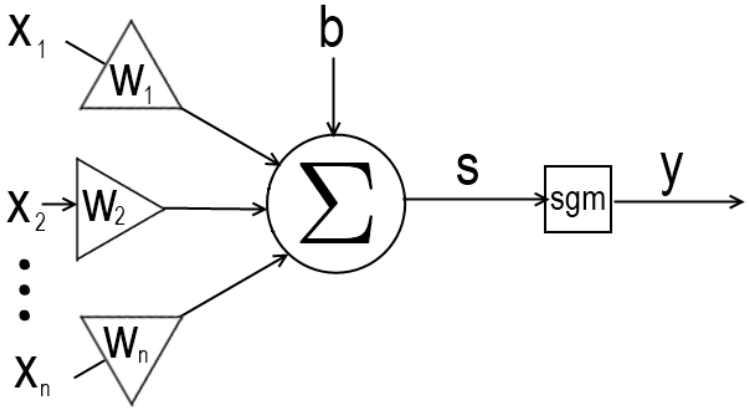


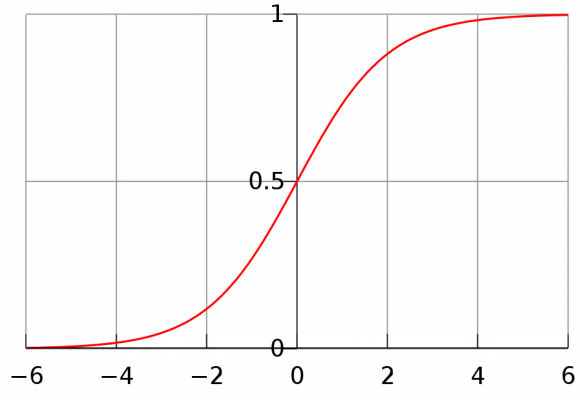
- Ha a mintából becsülve 0, elvész információ
  - Laplace-simítással nem lehet 0
- Ha $p(v_j|C_i)$ a mintából becsülve 0, elvész információ
  - Laplace-simítással nem lehet 0

$$\hat{p}(v_j|C_i) = \frac{\theta + |\{\boldsymbol{w} \mid w_j = v_j \wedge \mathcal{D}(\boldsymbol{w}) = C_i\}|}{2 \cdot \theta + |\{\boldsymbol{w} \mid \mathcal{D}(\boldsymbol{w}) = C_i\}|}$$

Neurális hálózatok

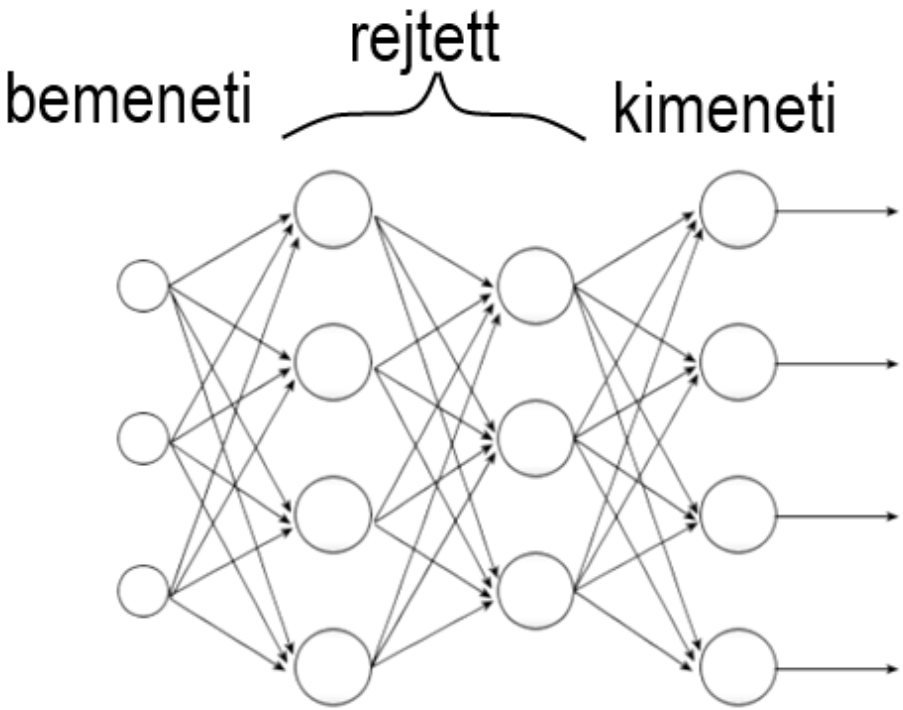
- Biológiai mechanizmusok alapján konstruált számító rendszerek
- Felügyelt tanulásra egyik legnépszerűbb a többrétegű perceptron (MLP)
  - Elemi neuronja: perceptron, kimenetén szigmoid nemlinearitással





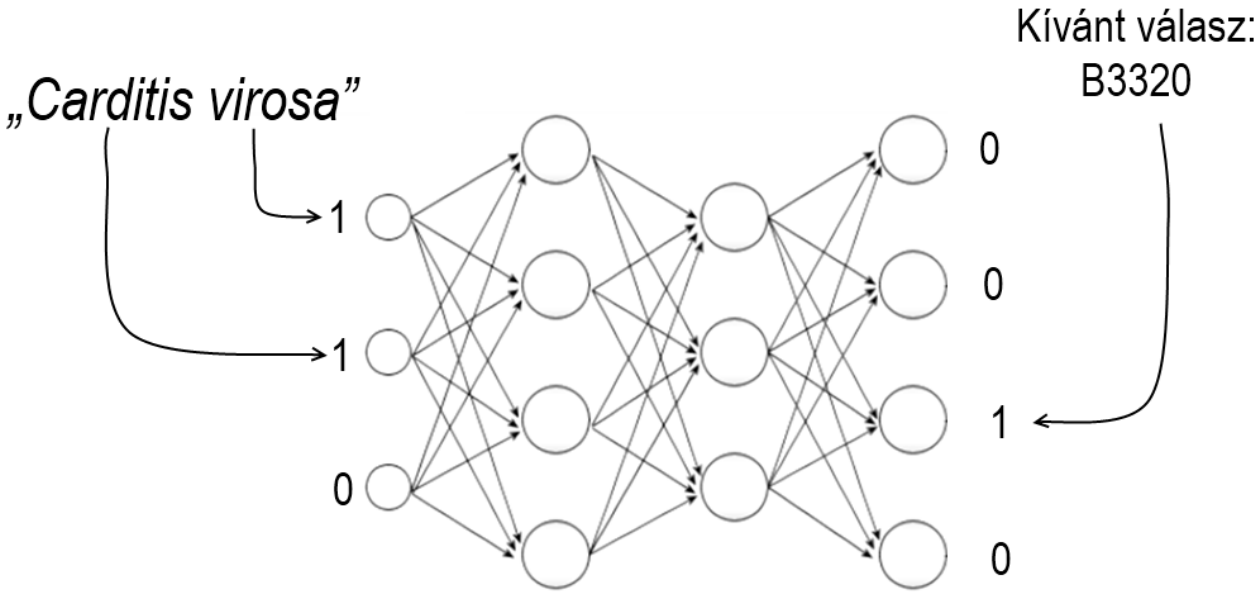
$$sgm(s) = \frac{1}{1 + \exp(-s)}$$

Többrétegű perceptron



Többrétegű perceptron

- Kimeneti neuronok a kódok relevanciáját becslik
  - Többezer kimeneti neuron



Többrétegű perceptron

- Tanítása hibakritérium alapján
  - Várható négyzetes hiba
  - Maximum likelihood (más néven keresztentrópia)
- Súlytérben a hiba negatív gradiense felé mozdulunk
- Tanítás ciklusokban
  - Hiba-visszaterjesztés (láncszabály alapján)
  - Amíg a hiba csökken (csúszóablak)

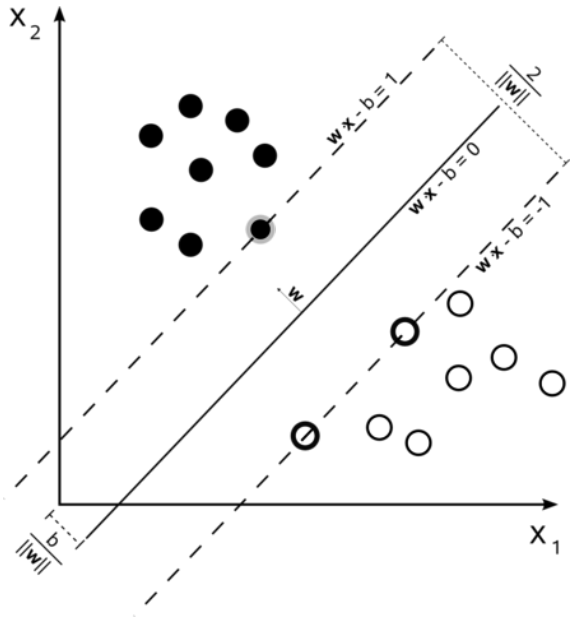
$$J(W) = \mathbb{E}_{\boldsymbol{x}}\left\{\sum_{j=1}^{n}\left[d_j(\boldsymbol{x}) - y_j(\boldsymbol{x}, W)\right]^2\right\}$$

$$J(W) = \mathbb{E}_{\boldsymbol{x}}\left\{-\sum_{j}\left(d_j(\boldsymbol{x}) \cdot \log\left[y_j(\boldsymbol{x}, W)\right] + \left(1 - d_j(\boldsymbol{x})\right) \cdot \log\left[1 - y_j(\boldsymbol{x}, W)\right]\right)\right\}$$

$$\Delta w = -a\frac{\partial J}{\partial w}$$

Szupport vektor gép (SVM)

- Lineáris változat
  - Nagy margó (biztonsági sáv) az elválasztó hipersík körül



- Lágy lineáris
  - „Rossz helyen” lévő minta megengedett, de büntetjük
  - Együttes optimalizálás (súlyozva)
- Nemlineáris
  - Nagydimenziójú jellemzőtérben lineáris
  - Szövegosztályozásnál általában nem szükséges

Többosztályos SVM

- Dekomponálás bináris feladatokra
- Megkülönböztethetünk
  - Egy osztályt a többitől (őket egybevonva)
  - Páronként az osztályokat
- SVM feladat átfogalmazása (több feltétel)

Keveréses módszerek

Kód	Bizonyosság
B	0,27
C	0,26
A	**0,25**
D	0,11
E	0,11

Kód	Bizonyosság
D	0,28
A	**0,25**
B	0,17
E	0,16
C	0,14

0,5

0,5

Kód	Bizonyosság
A	**0,25**
B	0,22
C	0,20
D	0,195
E	0,135

Kiértékelési mérce

- Találati arány (teljesség, recall)
  - A lista megengedett hosszától függ
  - $R(h) = \frac{\#\ helyes\ kód\ a\ listán}{\#\ tesztbemenetek}$
  - Súlyozás lehetséges a listán elfoglalt hely szerint (nem csak, hogy rajta van-e), relevancia szerint stb.

Keveréses módszerek

- Több osztályozó eredménylistájának összevonása
- Jobb eredmény, mint a legjobb különálló
  - Ha a hibás kódok „véletlenszerűek” vagy a sorrendjük véletlenszerű, de a jó kód nagyrészükön szerepel
    - Optimális konstans súlyok keresése (**konstans súlyozó**)
  - Ha a bemeneti tér különböző részein jók
    - Bemenetfüggő optimális súly tanulása (**súlybecslő**)
    - Tanuljuk meg, melyik hol milyen jó (**jóságbecslő**)

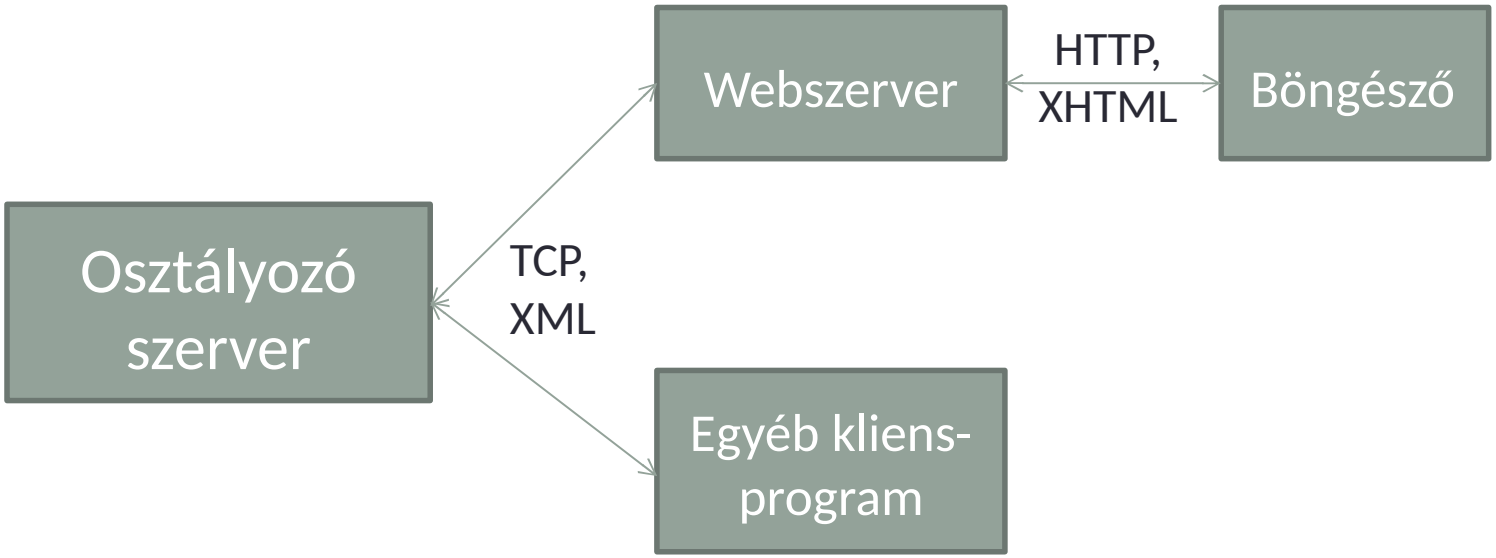
Keveréses módszerek

- Keverendő osztályozók előállítása
  - Eltérő modell
  - Eltérő tanítóminták
    - BNO hierarchikus
    - Főcsoportonként külön osztályozó (**szakértőegyüttes**, mixture of experts)
      - Külön kapuzó osztályozó tanulja: melyik bemenet melyik főcsoporthoz

IMPLEMENTÁCIÓ

Implementáció

- Alapvető felépítés



Osztályozóprogram

- Különböző módok
  - Tanítás és a kapott osztályozó fájlba szerializálása
  - Fájlból beolvasott osztályozóval TCP-n figyelés, érkező kérések (diagnózis) kiszolgálása (kódlista)
  - Keresztkiértékelés
- Szükséges
  - Osztályozó felépítésének megadása (XML)
  - Üzenetek (kérdés, válasz) formátuma (XML)

XML példák: Lekérés szervertől

```
<?xml version="1.0" encoding="UTF-8"
standalone="no"?>
<request>
  <inputs>
    <input>ruptura corpus lutea haemorrhagia</input>
  </inputs>
  <resultcount>5</resultcount>
</request>
```

XML példák: Szerver válasza

```
<?xml version="1.0" encoding="UTF-8" standalone="no"?>
<resultsets>
  <resultset>
    <result>
      <class>N8310</class>
      <confidence>0.38768336305190293</confidence>
    </result>
    <result>
      <class>D6990</class>
      <confidence>0.17026160892877706</confidence>
    </result>
    <result>
      <class>H4310</class>
      <confidence>0.15751956314998392</confidence>
    </result>
    <result>
      <class>R58H0</class>
      <confidence>0.15043689961141468</confidence>
    </result>
    <result>
      <class>K2280</class>
      <confidence>0.13409856525792138</confidence>
    </result>
  </resultset>
</resultsets>
```

Összetett osztályozó megadása (részlet)

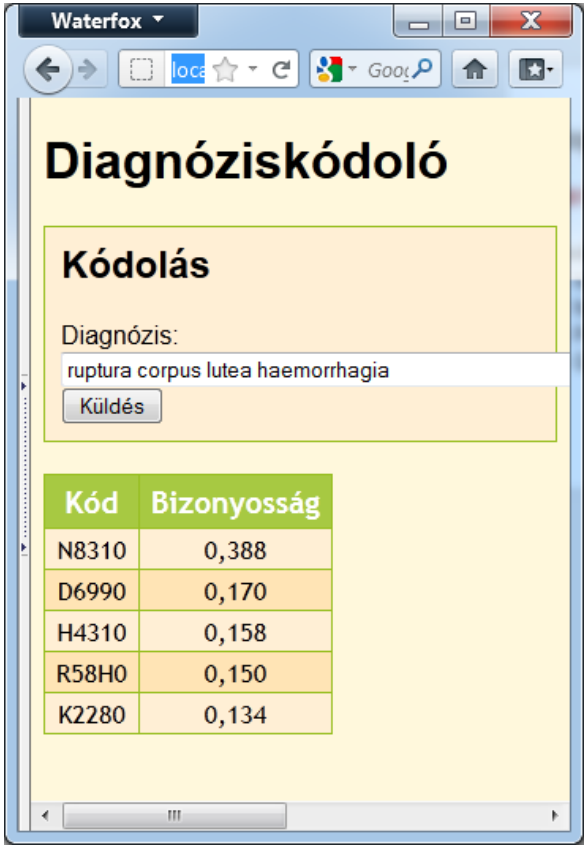
```
<classifier id="team">
    <type>ConstantWeightTeam</type>
    <param name="c1">
        <classifier ref="moe" />
    </param>
    <param name="c2">
        <classifier ref="idf" />
    </param>
    <param name="ratio">0.4</param>
</classifier>
```

Keretrendszer

- Java nyelven készítettem el
- Osztályozási feladatok kezelése általánosan
  - Tanítás
  - Kiértékelés
  - Bemeneti és kimeneti transzformációk
  - Mintahalmazok manipulációja (keverés, vágás, iteráció)
  - Eredményhalmazok kezelése
  - Párhuzamosítás egyszerűen
- Egyszerű bővíthetőség
  - Új osztályozók
  - Új kiértékelési mércék
  - Új transzformációk (pl. nyelvi feldolgozás)

Webes felület

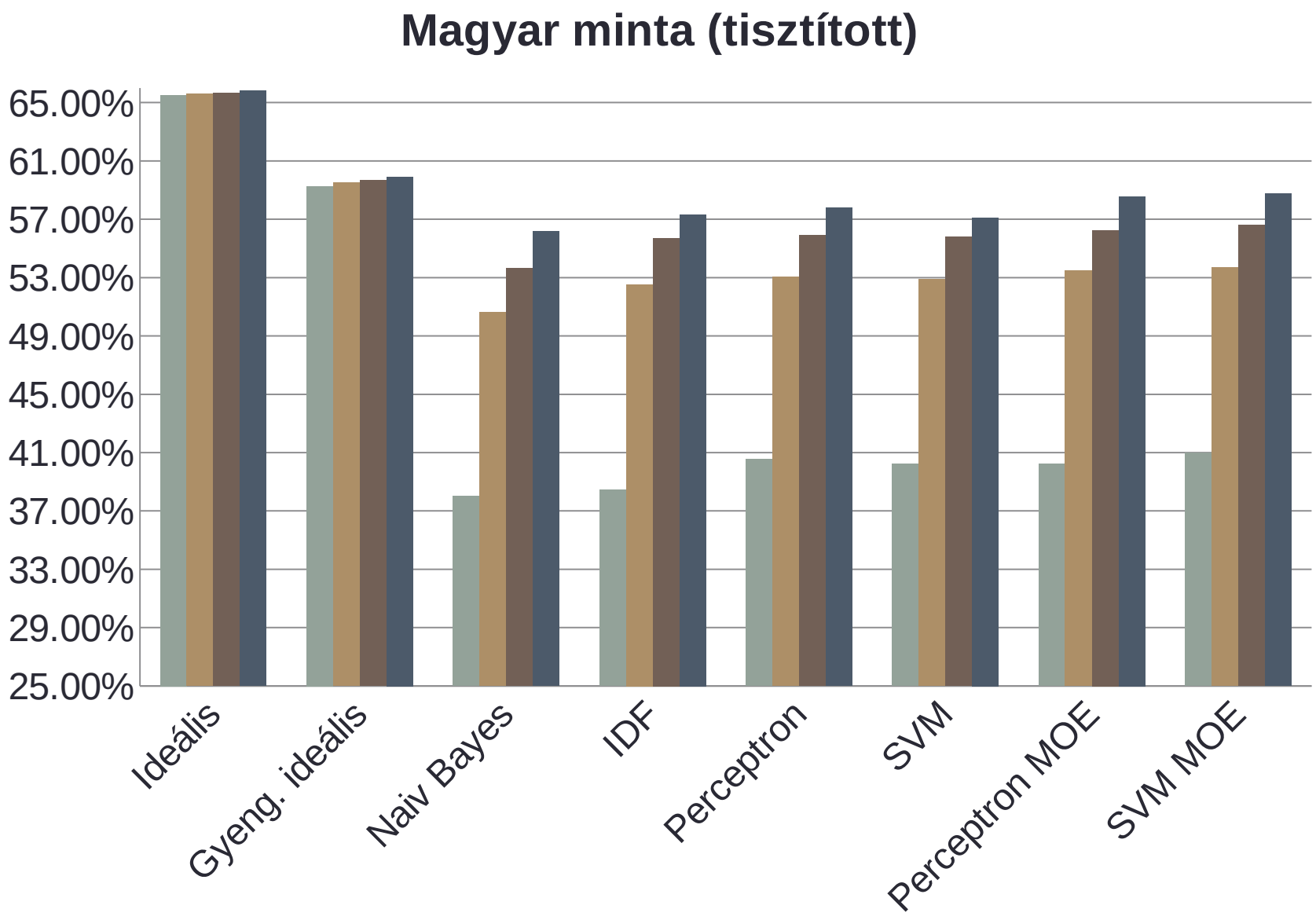
- JSP prototípus oldal

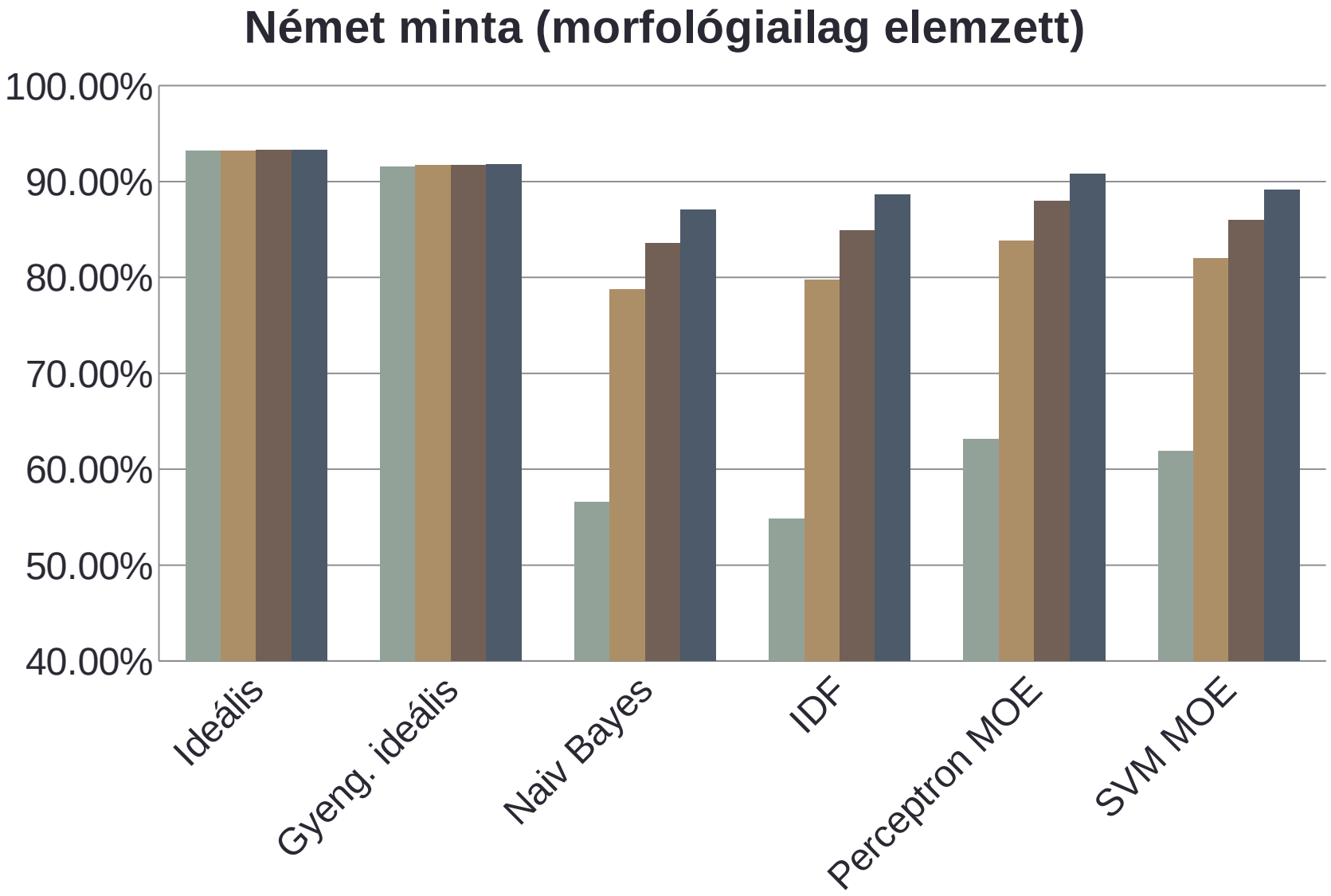


EREDMÉNYEK

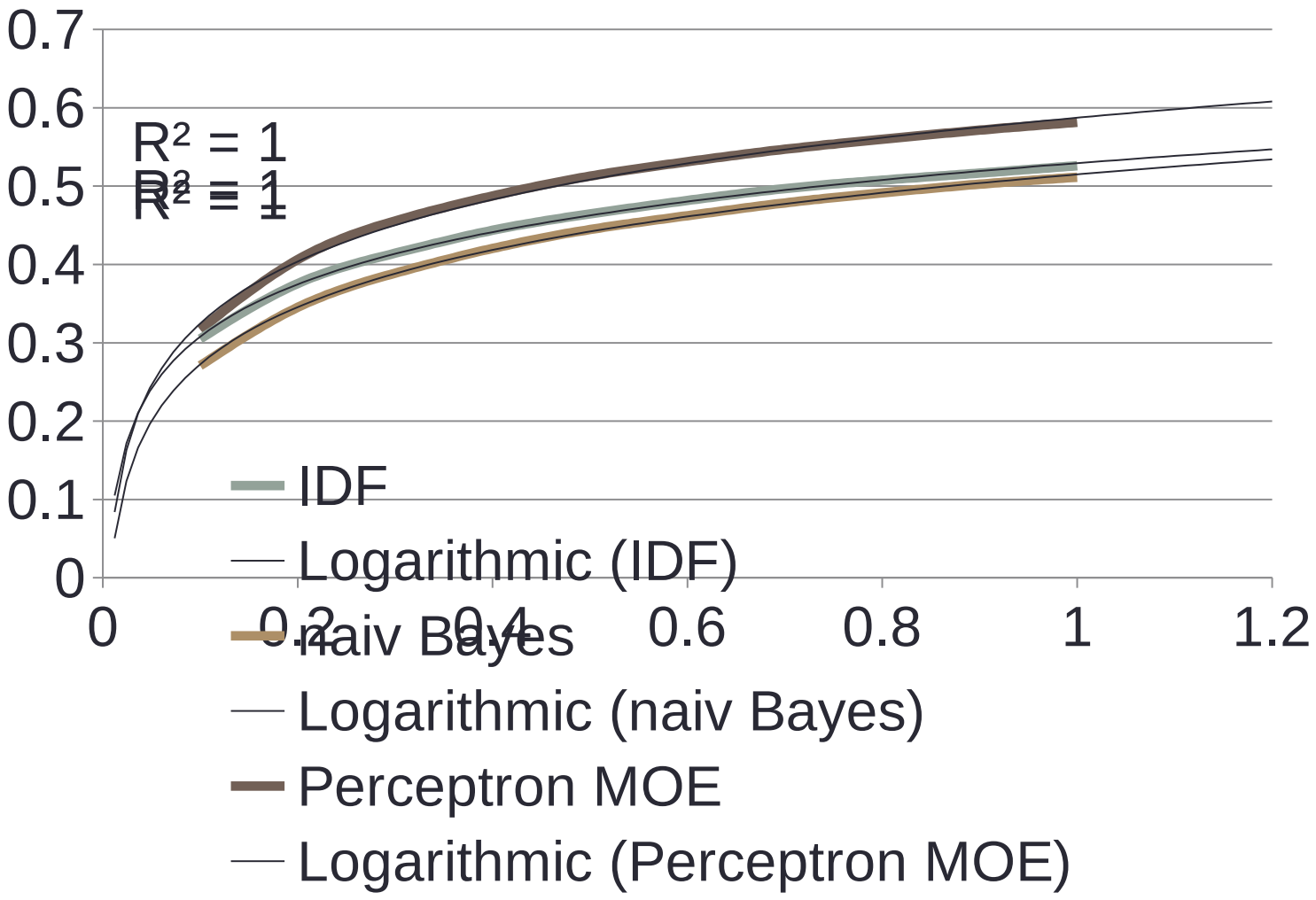
Mintahalmazok

- Magyar (3081 minta)
  - Tisztítatlan
  - Tisztított
- Német (93863 minta)
  - Feldolgozatlan
  - Szótövezett
  - Morfológiailag elemzett


Magyar minta (tisztított)
65.00%
61.00%
57.00%
53.00%
49.00%
45.00%
41.00%
37.00%
33.00%
29.00%
25.00%
Ideális
Gyeng. ideális
Naiv Bayes
IDF
Perceptron
SVM
Perceptron MOE
SVM MOE


Német minta (morfológiailag elemzett)
100.00%
90.00%
80.00%
70.00%
60.00%
50.00%
40.00%
Ideális
Gyeng. ideális
Naiv Bayes
IDF
Perceptron MOE
SVM MOE

Tanulási görbék (német feldolgozatlan)



Továbbfejlesztési lehetőségek

- A különböző algoritmusok hasonlóan teljesítenek
- Nagyobb különbséget okoz a minta jósága, mérete

- Előfeldolgozás finomítása
  - Morfológiai elemzés
- Ontológiák használata
  - Pl. speciális fogalmak általánosítása
  - Szinonimák felismerése

- Felhasználói felület bővítése, BNO-adatbázissal összekötés

KÖSZÖNÖM A FIGYELMET!