PROJECT BESZÁMOLÓ

Bevezetés a Természetes Nyelvű Szövegfeldolgozásba

Tartalom

[Project bemutatása 2](#_Toc68182587)

[Csapat tagok 2](#_Toc68182588)

[Fejlesztési idő 2](#_Toc68182589)

[Adathalmaz 2](#_Toc68182590)

[ELőfeldolgozás 2](#_Toc68182591)

[Modell 2](#_Toc68182592)

[Kiértékelés 3](#_Toc68182593)

[További fejlesztési lehetősége, Tapasztalatok 3](#_Toc68182594)

# Project bemutatása

# *A többcímkés osztályozás egy címkés probléma általánosítása, és egyetlen példány több osztályhoz is tartozhat. A scikit-learn csomag dokumentációja szerint "Ez úgy gondolható, hogy megjósolják a minta tulajdonságait, amelyek nem zárják ki egymást." Nincsenek korlátozások azon osztályok számával kapcsolatban, amelyekhez egy példány hozzárendelhető egy többcímkés feladatban. Hasonló kontextusban létezik a többosztályos osztályozási probléma. A legfontosabb különbség azonban az a tény, hogy a többcímkés osztályozás feltételezi, hogy a tulajdonságok nem zárják ki egymást.*

# *A valós problémák sora több címkés osztályozási problémaként jeleníthető meg, például a cikkek kategorizálása témaköröként. Az ebben a projektben szereplő probléma a Netflix filmjeinek és műsorainak műfaji kategorizálása, amelyek egyszerre több kategóriába is tartozhatnak.*

# Csapat tagok

* *Lucski Attila*

# Fejlesztési idő

*82 óra*

# Adathalmaz

* *A példa project adathalmaza 93480 rekordot tartalmaz.*
* *A rekordokhoz tartozó mezők:* 
  + ***show\_id****: Unique ID for every Movie / Tv Show*
  + ***type:*** *Identifier - A Movie or TV Show*
  + ***title:*** *Title of the Movie / Tv Show*
  + ***director:*** *Director of the Movie*
  + ***cast:*** *Actors involved in the movie / show*
  + ***country:*** *Country where the movie / show was produced*
  + ***date\_added:*** *Date it was added on Netflix*
  + ***release\_year:*** *Actual Release year of the move / show*
  + ***rating:*** *TV Rating of the movie / show*
  + ***duration:*** *Total Duration - in minutes or number of seasons*
  + ***listed\_in:*** *Genere*
  + ***description:*** *The summary description*
* *Az adatok címkézettek*
* *A címkék többes osztályúak*
* Link: <https://www.kaggle.com/shivamb/netflix-shows>

# ELőfeldolgozás

* *Betöltjük az adattáblát a memóriába, szemlélteti az adathalmaz alakját és kiírja az adathalmaz címkéit*
* *Az adatkészlet sok olyan mezőt tartalmaz, amelyek nem relevánsak a probléma szempontjából, és az összes műfaj csoportosított ugyanazon oszlopon belül. Ezeket eltávolítjuk, a kategóriákat szegmentáljuk. A kategóriák 3 csoportját felbontjuk.*
* *Ezzel az adatok címkeit 0, 1 számokká konvertáltuk attól függően, hogy melyik leírás melyik műfajokba tartozik.*
* *Megnézzük a címek között a leggyakoribb műfajokat, és túl kevés adatponttal ellenőrizzük a kategóriákat. Egyes kategóriákban nincs elegendő adatpont, és ez kevésbé segíti az előrejelzést. Ezért egy tetszőleges 200 -as küszöböt állítottam be, és a küszöb alatti kategóriák egy új Others nevű kategóriába csoportosulnak, így összesen 42-ből 21 műfaj lett.*
* *A több címkével ellátott címek számának megszámlálása és a szó gyakoriságának kiszámítása szintén statisztikát készítünk.*
* *Adatok normalizálása:*
  + *szavakat kisbetűssé konvertáljuk, ugye ez nem befolyásolja a szó jelentését.*
  + *eltávolítjuk az írásjeleket, számokat, mivel nincs különös jelentésük.*
  + *Stopword-öket eltávolítjuk, ugye az irrelevánsan gyakori szavakat, mivel nem adnának semmiféle értéket a modellnek. Nem segíti a modellt a döntésben, mivel bárhol előfordulhatnak.*
  + *Stemming alkalmazása, ezzel eltávolítva a szavak toldalékát, ezzel szótőre hozva a szót. Elkerülve, hogy például a kategória nevét tartalmazza ragozva és ne ismerje fel.*
  + *Lematizáció alkalmazása, a szóösszehúzások, rövidítések lecserélése teljes formájukra, szótári alakjukra. Mivel a jelentésük ugyanaz, így nem célszerű különbözőnek venni őket, egyes esetekben csökkentené a döntés biztosságát. Illetve a rövidítéseket stopword-nek vettem.*
* *Felosztottuk az adathalmazt tanítási és teszt készletre 70/30%-ban.*
* *A szavak vektorizálására van még szükség a modell oktatása előtt, és itt a tf-idf vektorizálót fogjuk használni, ami az alapján működik, hogy mennyire fontos egy szó a szótárban.*
* *Lehetőség van a WordEmbedding réteg betanítására vagy egy előre betanított réteg használatára transzfertanulás révén, például a word2vec és a GloVe segítségével.*
* *A Tensorflow modelleknél az alkalmazott vektorozás a text\_to\_sequences volt, amely átalakítja a szavakat számokká, és a pad\_sequences biztosítja, hogy az összes vektor azonos hosszúságú legyen.*
* *Az osztályok súlyát a kategóriák egyensúlyhiányának kezelésére számoltuk ki.*

# *.*Modell

* *Többfajta modellt is kipróbáltunk az adathalmazra:*
  + *Scikitlearn:*
    - *OneVsAll: LogisticRegression, MultinomialNB, LinearSVC, RandomFOrestClassifier*
    - *BinaryRevelance: GasussianNB*
    - *ClassifierChain: LogiticRegression*
    - *MultiOutput: KNeighboursClassifier*
  + *Tensorflow:*
    - *DNN with WordEmbedding*
    - *CNN with WordEmbedding*
    - *LSTM GloVe WordEmbedding-gel*
* *A pontosságot az AUC mérőszámmal mértük, ami a ROC görbe alatti terület*
* *A tanítást 5 különböző seeden keresztül végeztük, hogy csökkentsük az egyoldalúságot.*

# Kiértékelés

* *A Deep Neural Network mutatta a legjobb AUC pontszámot, de a különbség minimális a mély tanulási modellek között, és a CNN és az LSTM is hasonló teljesítményt nyújt. A scikit-learn csomagban elérhető algoritmusok jóval alacsonyabb pontszámot mutatnak be.*
* *A legjobb pontosság DNN: ~89%*

# További fejlesztési lehetősége, Tapasztalatok

*Azt tapasztaltuk, hogy minden egyes újabb preprocessing lépés bevezetésével jelentősen javult a hatékonyság, ezért az adathalmaz vizsgálata további statisztikai eszközökkel indokolt lehet azért, hogy újabb tisztítási lehetőségeket találjunk a további javulás érdekében. Például a szótőzésre egy másik algoritmus alkalmazása, mint például a Porter. Illetve a PyTorch könyvtár kipróbálása mélytanuláshoz, illetve a Wordembedding-hez a word2vec réteg használata. Más tanulási/teszt arány alkalmazása*