KAUNO TECHNOLOGIJOS UNIVERSITETAS
Informatikos fakultetas

Intelektikos pagrindai projekto ataskaita

Tema: Filmo žanro prognozavimas

Atliko:
Augustas Maslauskas,
Nerijus Dulkė,
Arūnas Bendoraitis
2019 m. gegužė 28 d.

Priėmė:
Lekt. Germanas Budnikas

KAUNAS, 2019

Turinys

Santrauka

Duomenų rinkinio pavadinimas: wiki_movie_plots_deduped.csv. Duomenis parsisiuntėme iš https://www.kaggle.com/jrobischon/wikipedia-movie-plots. Juos sudaro 8 atributai : [Release Year, Genre, Wiki Page, Plot, Cast, Director, Title, Origin/Ethnicity].

Problema: Iš filmo aprašymo (Plot) prognozuoti žanrą (Genre).

Panaudoti mašininio mokymo metodai:

- Naive Bayes,
- K-nearest neighbors
- Neural Network.

Atlikėjų sąrašas

Vardas	Pavardė	Užsiėmimo dieną ir laiką	Atsakomybės projekte	Parengtas skyrius ataskaitoje
Augustas	Maslauskas	Antradienis 09:00	k-nearest neighbors algoritmas	Ataskaita parengta bendrai
Nerijus	Dulkė	Antradienis 09:00	Neural Network algoritmas	Ataskaita parengta bendrai
Arūnas	Bendoraitis	Antradienis 09:00	Naive Bayes algoritmas	Ataskaita parengta bendrai

Programinės sistemos sukūrimas/ pritaikymas duomenims surinkti

Programa kurta JavaScript programavimo kalba. Naudojamas node.js framework. Duomenys buvo parsisiūsti iš https://www.kaggle.com/jrobischon/wikipedia-movie-plots.

Duomenų failą sudaro atributai : Release Year, Genre, Wiki Page, Plot, Cast, Director, Title, Origin/Ethnicity.

Duomenų surinkimas

Duomenys buvo parsisiųsti. Atributų skaičius : 8 - [Release Year, Genre, Wiki Page, Plot, Cast, Director, Title, Origin/Ethnicity], bet mes naudojame tiktai 2: [Plot, Genre], nes to prašo mūsų užduotis. Rinkinio dydis - ~35,000 įrašų.

Pavyzdys :

1910,drama,Mary Pickford plays Priscilla an unemployed maid who finds work at a farm. There she meets a no-good peddler who starts flirting with her and makes her fall in love with him. He runs up a gambling bill and asks her to help him pay his debts or he won't be able to marry her.[1],"Mary Pickford, Mack Sennett",D.W. Griffith,An Arcadian Maid,American,https://en.wikipedia.org/wiki/An_Arcadian_Maid

Duomenų paruošimas ir valymas

k-nearest neighbors - atveju filmų aprašymai išskaidomi į nepasikartojančius žodžius ir kiekvienam žodžiui priskiriama reikšmė. Taip sudaromas sąrašas unikalių žodžių ir jiem priskirtų reikšmių. Taip atrodo suformuotas sąrašas:

```
with: 52,
dynamite: 53,
others: 54,
fireman: 55,
halt: 56,
disconnect: 57,
The: 58,
passengers: 59,
off: 60,
rifle: 61,
them: 62,
their: 63,
belongings: 64,
One: 65,
passenger: 66,
tries: 67,
escape: 68,
but: 69,
instantly: 70,
shot: 71,
down: 72,
Carrying: 73,
loot: 74,
in: 75,
later: 76,
stopping: 77,
valley: 78,
horses: 79,
had: 80,
been: 81,
left: 82,
hoping: 83,
quiet: 84,
life: 85,
Things: 86,
start: 87,
go: 88,
```

Neural network atveju buvo naudojamas “Bag-of-words” metodas tekstui paruošti. Šiuo atveju tekstas pateikiamas kaip masyvas, nepaisant žodžių reikšmės bet koncentruojantis į jų pasikartojimų skaičių.

Dimensijų sumažinimas

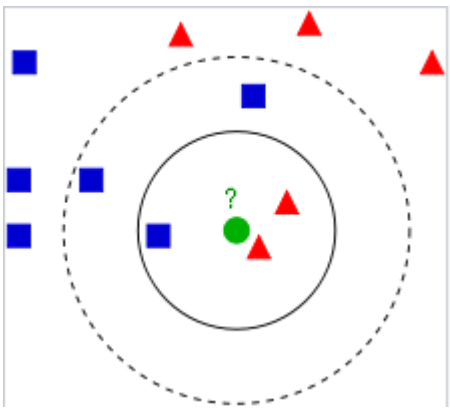
Duomenys išvalomi nuo nereikšmingų žodžių (he, she, the ir t.t.). Taip pat sugrupuojami panašūs žodžiai (pvz.: am, are, is -> be; car, cars, car’s -> car).

Pirmojo mašininio mokymosi metodo su mokytoju panaudojimas

k-artimiausi kaimynai - tai neparametrinis metodas, naudojamas klasifikavimui ir regresijai. Abiem atvejais įvestis susideda iš k artimiausių mokymo pavyzdžių, esančių erdvėje. Išėjimas priklauso nuo to, ar k-NN naudojamas klasifikavimui ar regresijai:

- K-NN klasifikacijoje produkcija yra narystė klasėje. Objektas yra klasifikuojamas pagal jo kaimynų balsų daugumą, kai objektas priskiriamas prie labiausiai paplitusios klasės k artimiausių kaimynų (k yra teigiamas sveikasis skaičius, paprastai mažas). Jei k = 1, tada objektas yra tiesiog priskiriamas tos pačios artimiausios kaimyno klasei.
    - K-NN regresijoje išėjimas yra objekto nuosavybės vertė. Ši vertė yra k artimiausių kaimynų verčių vidurkis

Paveikslėlyje matome žalias rutuliukas kaip naujas objektas dar nepriskirtas jokiai kategorijai (galimos dvi raudonas trikampis, mėlynas kvadratas). Matome galimus du variantus kai k -3 daugiau artimų figūrėlių raudonų todėl bus žalias rutuliukas priskirtas raudonam trikampiui, tai matyti pagal tiesios linijos ratą. kai k - 5 bus priskirtas mėlynam kvadratui (ribos iki brūkšninės linijos rato- matome daugiau mėlynų kvadratų).



Šaltinis : https://en.wikipedia.org/wiki/K-nearest_neighbors_algorithm

Atlikta kryžminė patikra naudojant https://www.npmjs.com/package/ml-cross-validation biblioteką. Ji automatiškai parenka segmentų skaičių. Buvo naudojama nedideli duomenų kiekiai dėl sistemos apribojimų.

Duomenų skaičius	Tikslumas (procentais)
100	0.22
400	0.2125
1000	0.144
1500	0.18733333333333332
2000	0.18

Matome, kad paduodant skirtingą duomenų skaičių gaunamas skirtingas tikslumas, tai yra dėl to, nes ženkliai padidėja naujų žodžių ir žanrų, todel klasifikuoti tampa labai sunku.

Antrojo mašininio mokymosi metodo su mokytoju panaudojimas

Neuroniniai tinklai yra tam tikros struktūros matematinės funkcijos, kurios naudojamos kaip funkcijų aproksimatoriai.

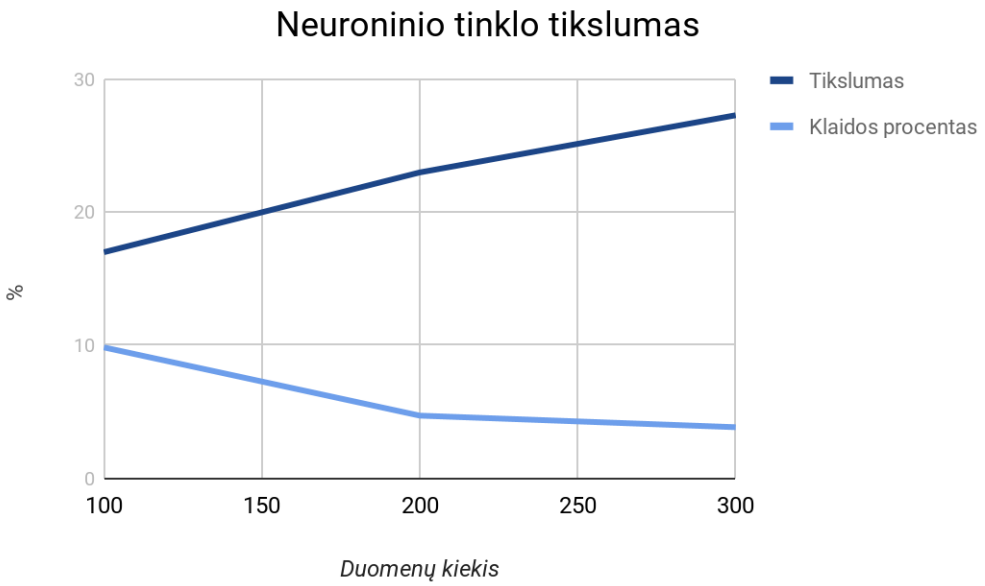
Šiai užduočiai spręsti buvo panaudota brain.js biblioteka, kuri duoda neuroninio tinklo implementaciją. Del duomenu kiekio ir sistemos apribojimų teko dirbti su mažesniais duomenų kiekiais (200-400 filmų). Dėl mažų duomenų kiekių galima didesnė paklaida, nes neuroninis tinklas negali būti tinkamai apmokytas.

Atlikta kryžminė patikra, duomenys skaidomi į 10 segmentų, su devyniais iš jų yra apmokomas neuroninis tinklas, o su dešimtu testuojamas.

Neuroninis tinklas mokosi kol pasiekia tam tikrą paklaidos ribą arba įvykdo tam tikra skaičių mokymosi iteracijų.

Žemiau pateikta lentelė su bandymų rezultatais:

Duomenų kiekis	**Klaidos procentas**	**Teisingai nustatytų žanrų skaičius**	**Tikslumas procentais**
100	9,84%	17	17%
200	4,71%	46	23%
300	3,84%	82	27,3%



Iš bandymų rezultatų matome, kad didėjant duomenų kiekiui, didėja ir tikslumas, tačiau tai žymiai padidina vykdymo trukmę (100 duom. - ~1,5 min., 300 duom - ~10 min.).

Trečiojo mašininio mokymosi metodo su mokytoju panaudojimas

Prieš klasifikuojant, duomenys tokenizuojami, pašalinami visi dažnai pasikartojantys anglų kalbos žodžiai. Toliau, trečiam metodui pasirinkta naudoti bajeso teoremą kuri formuluojama taip:

P(A | B) = (P(B | A)* P(A)) / (P(B))

Formulę galima apibūdinti pavyzdžiu.

P(Komedija |Three buddies wake up from a bachelor party in Las Vegas) = (P(Three buddies wake up from a bachelor party in Las Vegas |Komedija)* P(Komedija))/(P(Three buddies wake up from a bachelor party in Las Vegas))

Kadangi šis aprašymas nėra mokymosi duomenyse, filmų aprašymai išskaidomi į atskiras tikimybes kiekvienam žodžiui.

P(Three buddies wake up from a bachelor party in Las Vegas) = P(Three)* P(buddies)*P(wake)* P(up)* P(from)* P(a)* P(bachelor)*P(party)* P(in)* P(Las)* P(Vegas)

Po šio žingsnio, išvardinti žodžiai pasirodo mokymosi duomenyse ir turėsime prasmingas tikimybes.
Toliau skaičiuosime tikimybę pagal bajeso teoremą. Tam reikia apskaičiuoti kiekvienai klasei tikimybę

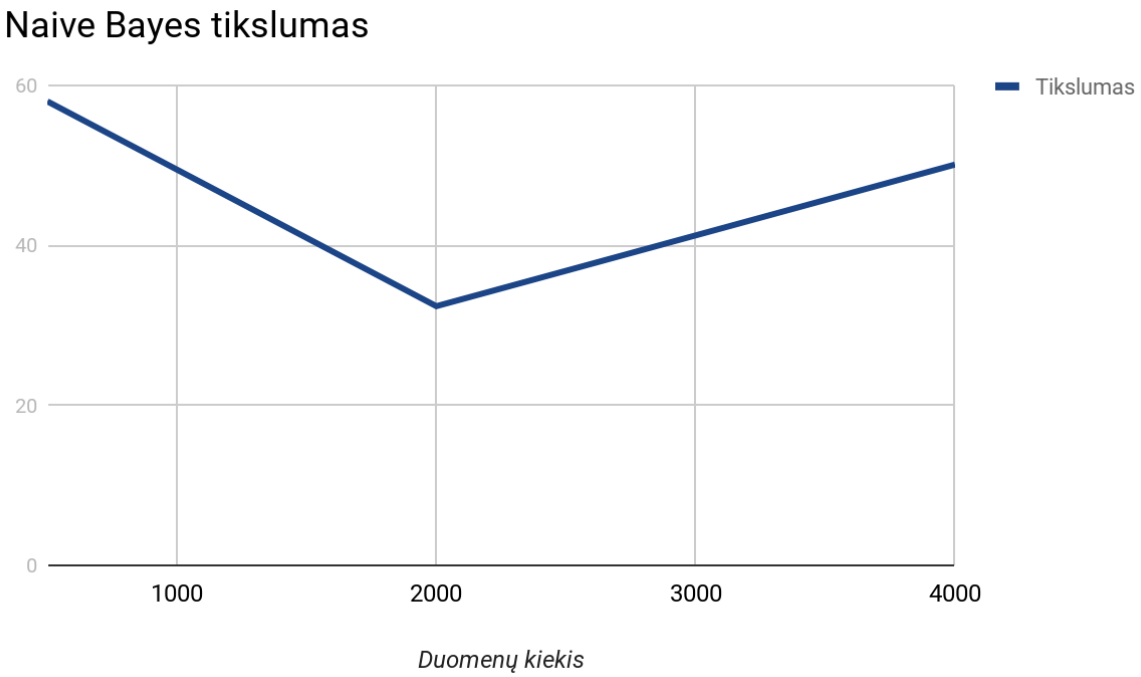
P(A) = (klasės pasikartojimas duomenyse) / (visi įrašai)

Šitaip apskaičiuojame P(A) esantį bajeso teoremoje. Sekantis žingsnis P(B|A), tai apskaičiuojame skaičiuodami kiek kartų žodis B pasikartoja klasėje A. Tačiau to nepakanka, nes bet kuris žodis, kuris yra nerastas mokymosi duomenyse sugadins aukščiau esančią tikimybių sandaugą tarp žodžių, kadangi daugyba iš 0 visada gražins nulinę tikimybę. Tokiu tikslu naudojama Laplaso lyginimas (angl. Laplace smoothing) iš kurio gauname formulę:

P(B|A) = (B pasikartojimas klasėje A+1) / (visi žodžiai klasėje A+ visi žodžiai)

Naudojant k=10 kryžminę patikrą, gautas šio metodo tikslumo įvertinimas:

Su 2000 įrašų vidutinis tikslumas 32.4%
Su 4000 įrašų vidutinis tikslumas 50.1%



Su mažu duomenų kiekiu, klasifikatorius turi didesnį tikslumą, tačiau tai yra tiesiog atspėjama kadangi drama ir komedija yra dažniausi žanrai duomenyse.

Literatūra

https://www.kaggle.com/jrobischon/wikipedia-movie-plots
https://en.wikipedia.org/wiki/K-nearest_neighbors_algorithm
https://www.npmjs.com/package/ml-knn
https://medium.com/@tech_fort/classifying-text-with-neural-networks-and-mimir-in-javascript-94c9de20c0ac
https://github.com/BrainJS/brain.js
https://github.com/machinelearningmindset/machine-learning-course
https://hackernoon.com/machine-learning-with-javascript-part-2-da994c17d483