**Navn:** Majdi Alali

**Uio-brukernavn:** majdioa

**Mitt svar på Oblig1, IN2110**

**Oppgave 1**

**A)**

Jeg tror at koden min forklarer seg selv.

**B)**

**Første del,**

Etter bruk av den orginale funksjonen tokenise(), er antall tokens i hele korpuset 10014784 og

antall oredtyper er: 689501.

**Andre del,**

Etter bruk av nltk sin funksjon word\_tokenise(), er antall tokens : 11474658 og

antall ordtyper er: 439247

**Tredje del,**

Jeg har endret på funksjonen tokenize() ved å gjøre alle bokstaver til små og fjerne punctuation marks og stopwords

antall tokens etter mine endring i tokenise(): 9586530

antall ordtyper etter mine endring i tokenise(): 386972

**Fjerde del,**

Det er helt klar at tokaniseringen etter endringene jeg har gjort på funksjonen tokenise(), gir lavest tokens og ordtyper.

**C)**

**Første del,**

|  |  |
| --- | --- |
| **Kategori's navn** | **Antall dokumenter pr kategori** |
| “gaming” | 1413 |
| “literature” | 2821 |
| “restauranter” | 428 |

Flest dokumenter tilhører til kategori literature. Kanskje fordi korpuset er litt gammel eller innholder messer artikler til verk av Shakespeare.Det er tydlig at dokumenter som har noe å gjøre med mat og drikke er lavest.

**Ekstra ting om oppgaven:**

- Jeg hadde et Memory Error når jeg prøvde å ta unike ord for alle dok . Jeg skjønte ikke om det fantes en funksjon son heter set().

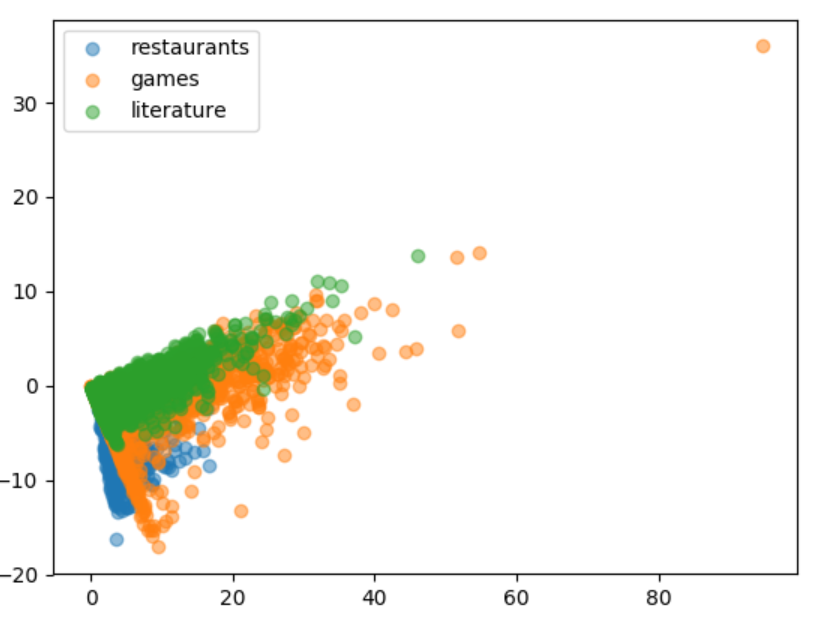
-Jeg har importert noen ekstra klasser for å hjelpe med å utføre noen operasjoner på string-variabler.

**Oppgave 2**

**A)**

Først setter jeg alle tekster/dokumenter (som har en av de tre kategoriene) i en liste med navn listeOver\_deTreKategorier. Så i konstruktøren har jeg gjort mange ting i en setningskode, Første har jeg laget et objekt av CountVectorizer med variabelnavn min\_Vectorizer og bruke min toknize(), som fjerner funksjonord, tegnsetting og gjøre alle ord til små bokstaver. I tillegg har jeg begrenset antall ord ved max\_features. Merk atfuksjonen vec.train() brukte jeg fit() for å har ordbok hvor unike ordene danner nøklene og tall av indekser. Merk at ordene ligger alfabetisk I rekkefølge I forhold til indeksene.

B)



Plasseringen til hvert dokument/punkt sier hvor nært deter til andre dokumentene.

Jeg ser det er noen dokumenter/prikker med kategori “games” som sprer seg litt lang fra hverandre roundt i grafen. Dvs. at dokumentenefor i denne kategorien har forskjellige ord. Dermed har de stort sett forskjellige temaer om spill. Mens dokumentene som hører til “restaurants”-kategori ligger nært til hverandre. Dvs. At de har mange ordtyper til felles. Også bør vi legge også merke til at noen dokumenter mellom de tre kategoriene har mange ord til felles fordi de legge over hverandre (selv om jeg ikke kan se dem).

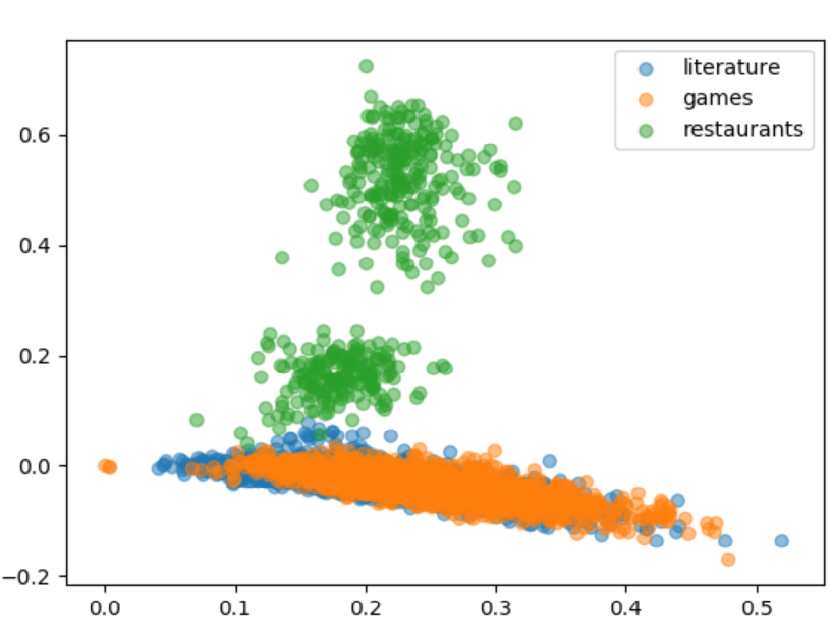
Merk:

#har merkert at selv om jeg har fjerntet de norske funksjon ord via tokanise() er det fortsatt noen uønskede ord/tegn som har høy forekomester.

#merk at jeg har gjordt toknise til små bokstaver, derfor er det ikke nødvendig å bruke lowercase() som rgument i countvectorizer

#Jeg har ikke forstått forskjellen mellom vec\_train() og vec\_test()

C)



Grafen har forandret seg på grunn av at det nye ordforrådet/ordtyper. Dvs. at unødvendige ord har blitt fjernet med etter bruk av tf-df. Jeg merker også at kolonnene or radene er mellom 0 og 1.

**alarm:**

Når jeg kjører metodene vec\_train() med returnverdi vec eller vec\_tfidf, går det fint men når jeg kjører programmet med både vec og vec\_tfidf, får jeg en feilmelding (ValueError: Expected 2D array, got 1D array instead:array=[list([]) ).

**Oppgave 3**

**A)**

Jeg fikk noen unntakk når jeg kjøret metoden creatknn() som:

1) raise ValueError("Unknown label type: %r" % y\_type)

Når jeg sendte listen med labels hvor typen for elementene var ordbok, ikke string.

En annen unntakk var:

2) ValueError: Found input variables with inconsistent numbers of samples: [9324, 4662]

Problemet her var at jeg trente dataene to ganger. Det vil si at antall labels var mismatchet me antall dataene/tekstene for dokumentene.

En forskjellig feil var:

3) ValueError: Expected 2D array, got 1D array instead:

array=[<10487x5000 sparse matrix of type '<class 'numpy.int64'>'

Dette skjedde pga at jeg brukte dataene som parameter, ikke dokumentvektorer som har to deminsjoner.

**B)**

Merk at jeg bruker tfidf for å få beste resultater

Valideringstabell:

|  |  |
| --- | --- |
| k | accuracy |
| K med verdi 1 | 1.0 |
| K med verdi 2 | 0.3833476394849785 |
| K med verdi 10 | 0.9843776824034335 |
| K med verdi 100 | 0.9905579399141631 |

Det er sant at acuraccy er best i starten mens reduserer kraftig når k øker. Uten tvil, K med verdi 1 gir best resultatet.

**C)**

teststabell:

|  |  |
| --- | --- |
| k | accuracy |
| K med verdi 1 | 1.0 |
| K med verdi 2 | 0.3833476394849785 |
| K med verdi 10 | 0.9905579399141631 |
| K med verdi 100 | 0.9905579399141631 |

Resultatene fra testtabell gir akurat samme resultater fra valideringstabell. Dvs. Klassifikatoren gir veldig bra for å klassifisere nye data.