بسمه تعالی

دانشکده مهندسی کامپیوتر

داده کاوی

نام استاد: دکتر حسین رحمانی

پروژه دوم

آرمان حیدری

شماره دانشجویی: ۹۷۵۲۱۲۵۲

فروردین ۱۴۰۱

فهرست

خواندن فایل ورودی

همانطور که واضح است دو فایل movie_synopsis و movie_info را با هم ترکیب میکنیم و معیار را برابر بودن local_id میگیریم. اشتباها در یکی از فایل ها به اسم locale_id ثبت شده که آن را در کد اصلاح میکنیم. و جدولی از داده های خاممان می‌سازیم.

که ۱۶۸۲ سطر (تعداد فیلم ها) و ۷ ستون (تعداد ویژگی مه از هر فیلم داریم، یعنی local_id، plot_source، imdbId، plot_synopsis، title، genre_imdb، id_imdb) دارد. نمونه ای از دیتاست:

	imdbID	plot_synopsis	local_id	plot_source	title	id_imdb	genre_imdb
0	tt0114709	A boy called Andy Davis (voice: John Morris) u...	1.0	imdb	Toy Story (1995)	tt0114709	Animation|Adventure|Comedy|Family|Fantasy
1	tt0113189	The story opens in 1986, in the Cold War Sovie...	2.0	imdb	GoldenEye (1995)	tt0113189	Action|Adventure|Thriller
2	tt0113101	The film begins with Ted the Bellhop (Tim Roth...	3.0	imdb	Four Rooms (1995)	tt0113101	Comedy
3	tt0113161	Chilli Palmer (John Travolta) is a loan shark ...	4.0	imdb	Get Shorty (1995)	tt0113161	Comedy|Crime|Thriller
4	tt0112722	After giving a guest lecture on criminal psych...	5.0	imdb	Copycat (1995)	tt0112722	Drama|Mystery|Thriller
5	tt0115012	Tang Shuisheng (Wang Xiaoxiao) has arrived in ...	6.0	imdb	Shanghai Triad (Yao a yao yao dao waipo qiao) ...	tt0115012	Crime|Drama|History|Romance|Thriller

پیش‌پردازش داده

سوال ۱:

Stemming و Lemmatization هر دو شکل ریشه کلمات را ایجاد می کنند. تفاوت این است که stem ممکن است یک کلمه واقعی نباشد در حالی که lem یک کلمه واقعی زبان است.

به طور کلی stemming از ریشه یابی استفاده میکند. و الگوریتمی با مراحلی دارد که آن را سریعتر می کند. برای مثال در زبان انگلیسی ing انتهای کلمه را حذف میکند، یا s آخر کلمه را. درواقع بستگی به دستورالعمل از پیش تعیین شده ای و نه با توجه به نوع و نقش کلمه آن را کوتاه می‌کند، یا تغییری نمی دهد.

Lemmatization نتایج بهتری را با انجام تجزیه و تحلیلی که به POS (نقش کلمه در جمله) بستگی دارد و تولید کلمات فرهنگ لغت واقعی ارائه می‌دهد. در نتیجه، پیاده‌سازی آن سخت‌تر و در مقایسه با stemming کندتر است. مثلا برای افعال یا اسم ها روش متفاوتی دارد. و خروجی آن یک ریشه واقعی در زبان مربوطه است.

```
nltk stemming results:
rocks : rock
exhaustive : exhaust
corpora : corpora
better : better

nltk lemmatization results:
rocks : rock
exhaustive : exhaustive
corpora : corpus
better : good
```

میبینیم که در مثال اجرا شده، میبینیم هر دو s جمع را به درستی حذف کرده اند. اما exhaust کلمه بی معنی است که از الگوریتم های stemming حاصل شده است در حالی که lemmatization آن را عوض نمیکند. و در دو مورد بعدی هم به درستی لم کلمات پیدا شده که با معنی هستند و ریشه گروهی از کلمات محسوب میشوند، در حالی که Stemming کلمه را عوض نکرده و احتمالا better و best را نمیتواند از یک ریشه تشخیص دهد.

تمرین ۱:

به ترتیب این پیش پردازش ها را انجام می دهیم:

- بررسی که سطرهای nan چقدر هستند و چون تعدادشان کم بود همه را حذف کردیم و به ۱۵۹۶ نمونه میرسیم.
- بررسی کردیم که دو ستون مربوط به آیدی imdb آیا متفاوت است یا خیر که همگی برابر بودند و این یعنی داده نویزی نداریم. البته چون این ستون اهمیت دیگری ندارد به کل هر دو را حذف کرده ایم.
- کدی برای مقایسه lemmatization و stemming زدیم و طبق توضیحات قسمت قبل و چون داده ها چندان زیاد نیست که سرعت مهم باشد، از lemmatization استفاده میکنیم.
- برای ستون های متنی، یعنی نامم فیلم و متن توضیحی آن، یک تابع text_preprocessing تعریف کرده ایم که یک متن را میگیرد و کار های زیر را به ترتیب انجام میدهد:

۱. تمام متن را tokenize میکند، و در حین این کار با توجه به نوع tokenization استفاده شده از NLTK، علائم نگارشی و کلا هر چیزی غیر از کلمات یا اعداد را دور میریزد. فواصل زیاد و enter و spaceو ... همگی حذف میشوند و در token ها نمی آیند.

۲. تمام token ها را به حالت lowercase میبریم.

۳. تمام stopword های زبان انگلیسی را از nltk دانلود میکنیم و یک set میسازیم که جستجو در آن سریع باشد. سپس تک تک توکن ها را اگر stopword بودند حذف میکنیم.

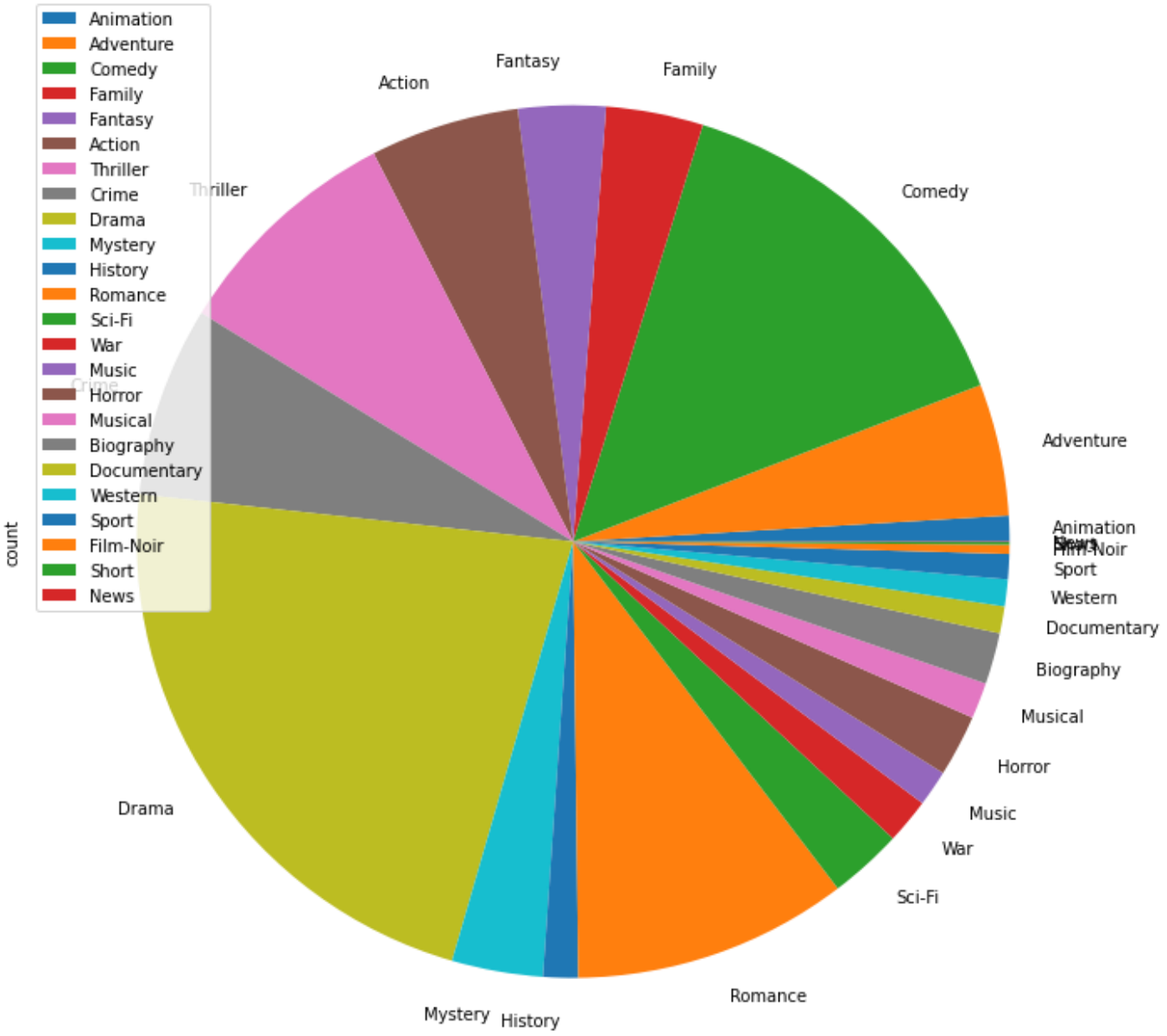
۴. توکن هایی با طول ۱ را حذف میکنیم. مواردی مانند ... I'd like دو تا توکن با طول ۱ ایجاد میکنند که I و d ارزشی ندارند.

۵. توکن های حاوی اعداد را حذف میکنیم.

۶. تمام توکن ها را با lemmatize شده آن ها جایگزین میکنیم. برای این کار هم از API موجود در کتابخانه NLTK استفاده میکنیم. نام آن WordNetLemmatizer است و بسیار معروف است و از مجموعه داده بزرگی به نام wordnet به زبان انگلیسی استفاده میکند.

- Plot_source فقط دو مقدار wiki و imdb را دارد که آن ها را با ۰ و ۱ جایگزین میکنیم تا عددی شوند و بتوانیم به مدل بدهیم.
- ژانر های فیلم ها را ابتدا جمع میکنیم (با split کردن کاراکتر "|") و سپس آن ها را با یک بردار ۲۴ تایی از ۰ که فقط ژانر های موجود هر record در آن ۱ هستند جایگزین میکنیم. به این صورت تمام ژانر ها با یک

آرایه ۲۴ تایی از ۰ و ۱ برای هر نمونه نمایش داده میشوند. در این مرحله پراکندگی ژانرهای مختلف را در یک نمودار آورده ایم که دیدنش خالی از لطف نیست:



پس از اتمام پیش پردازش علاوه بر این که سایز جدول به 1596*5 به جای 1681*7 میرسد:

	plot_synopsis	local_id	plot_source	title	genre_imdb
0	boy called andy davis voice john morris us toy...	1	0	toy story	[1.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, ...
1	story open cold war soviet union british secre...	2	0	goldeneye	[0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, ...
2	film begin ted bellhop tim roth room filled ho...	3	0	four room	[0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, ...
3	chilli palmer john travolta loan shark living ...	4	0	get shorty	[0.0, 0.0, 1.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, ...
4	giving guest lecture criminal psychology local...	5	0	copycat	[0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, ...
...	...	...	...	...	...
1675	angel celebrates birth daughter taking first h...	1677	1	sweet nothing	[0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, ...
1677	alan jared harris schoolteacher london also mo...	1679	1	monkey	[0.0, 0.0, 1.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, ...
1678	film follows helen quilley gwyneth paltrow you...	1680	0	sliding door	[0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, ...
1679		1681	1	crazy	[0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 1.0, ...
1680		1682	1	scream stone schrei au stein	[0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, ...

1596 rows x 5 columns

میتوانیم تاثیر تابع text_processing هم به طور مجزا در برخی موارد ببینیم. که در نوتبوک ۵۰ مورد رندوم نمایش داده شده است. یک نمونه را باهم ببینیم. متن اصلی:

```
While growing up in Oklahoma, young Lane Frost (Cameron Finley) learns the tricks of the bull riding trade at the hand of his father, Clyde (James Rebhorn), an accomplished rodeo bronco rider himself. As he enters his teenage and early adult years, Lane (Luke Perry) travels the western rodeo circuit with his best friends Tuff Hedeman (Stephen Baldwin) and Cody Lambert (Red Mitchell). He meets and falls in love with a young barrel racer, Kellie Kyle (Cynthia Geary), and they eventually marry in 1984.As Lane's legend and fame increase, so does the amount of pressure he puts on himself, to be what everyone wants him to be, and he wants to show that he is as good as they say he is. His ascent to the world championship is marred by a cheating incident, questions about Kellie's devotion, and a near broken neck. The film also follows him through the true life series between himself and Red Rock, a bull that no cowboy had ever been able to stay on for 8 seconds. It cuts the series down to three rides. In 1989, he is the second-to-last rider at the Cheyenne Frontier Days Rodeo. While riding on the bull known as "Takin' Care Of Business", he dismounts after his 8-second ride but the bull turns back and hits him in the side with his horn, breaking some ribs and severing a main artery. As a result of excessive internal bleeding, he dies on the arena floor before he can be transported to the hospital.The final scene shows Hedeman later that same year at the National Finals Rodeo riding for the world championship. After the 8 second bell sounds, he continues to ride and stays on an additional 8 seconds as a tribute to his fallen best friend.
```

پس از انجام پیش پردازش:

```
growing oklahoma young lane frost cameron finley learns trick bull riding trade hand father clyde james rebhorn accomplished rodeo bronco rider enters teenage early adult year lane luke perry travel western rodeo circuit best friend tuff hedeman stephen baldwin cody lambert red mitchell meet fall love young barrel racer kellie kyle cynthia geary eventually marry lane legend fame increase amount pressure put everyone want want show good say ascent world championship marred cheating incident question kellie devotion near broken neck film also follows true life series red rock bull cowboy ever able stay second cut series three ride second last rider cheyenne frontier day rodeo riding bull known takin care business dismount second ride bull turn back hit side horn breaking rib severing main artery result excessive internal bleeding dy arena floor transported hospital final scene show hedeman later year national final rodeo riding world championship second bell sound continues ride stay additional second tribute fallen best friend
```

استخراج ویژگی

سوال ۲:

میتوان به tf-idf و word embedding و word frequency و bag of words اشاره کرد.

Tf-idf: TF-IDF مخفف عبارت فرکانس معکوس سند فرکانس است. موضوع خاصی را برجسته می کند که ممکن است در مجموعه ما زیاد نباشد، اما اهمیت زیادی دارد. مقدار TF–IDF متناسب با تعداد دفعاتی که یک کلمه در سند ظاهر می شود افزایش می یابد و با تعداد اسناد موجود در مجموعه حاوی کلمه کاهش می یابد. از ۲ بخش فرعی تشکیل شده است که عبارتند از :

۱. Term frequency: فراوانی عبارت مشخص می‌کند که یک عبارت چقدر در کل سند ظاهر می‌شود. می‌توان آن را به عنوان احتمال یافتن یک کلمه در سند در نظر گرفت. تعداد دفعاتی که یک کلمه w_i در یک بررسی r_j رخ می‌دهد را با توجه به تعداد کل محاسبه می‌کند. از کلمات در بررسی r_j .به صورت فرموله شده است:

$$tf(w_i, r_j) = \frac{No.\,of\,times\,w_i\,occurs\,in\,r_j}{Total\,no.\,of\,words\,in\,r_j}$$

۲. Inverse Document Frequency(IDF): فرکانس معکوس سند معیاری است که نشان می دهد یک عبارت نادر یا متداول در اسناد در کل مجموعه است. آن کلماتی را که در اسناد بسیار کمی در سراسر مجموعه وجود دارد برجسته می کند، یا به زبان ساده، کلماتی که نادر هستند دارای امتیاز IDF بالایی هستند. IDF یک مقدار نرمال شده log است که از تقسیم تعداد کل اسناد D موجود در مجموعه بر تعداد اسناد حاوی عبارت t و گرفتن لگاریتم عبارت کلی به دست می آید.

$$idf(d, D) = \log \frac{|D|}{\{d \epsilon D : t \epsilon D\}}$$

از آنجایی که نسبت داخل تابع log IDF باید همیشه بزرگتر یا مساوی ۱ باشد، بنابراین مقدار IDF (و بنابراین tf–idf) بزرگتر یا مساوی ۰ است. وقتی یک عبارت در تعداد زیادی از اسناد ظاهر می شود، نسبت درون لگاریتم به ۱

نزدیک می‌شود و IDF به ۰ نزدیک‌تر است. فرکانس فرکانس معکوس سند TF-IDF (TF-IDF) حاصل ضرب TF و IDF است. به صورت فرموله شده است:

$$tfidf(t,d,D) = tf(t,d) * idf(d,D)$$

امتیاز TF-IDF بالا با عبارتی به دست می آید که دارای فرکانس بالا در یک سند و فرکانس سند پایین در بدنه باشد. برای کلمه‌ای که تقریباً در همه اسناد ظاهر می‌شود، مقدار IDF به ۰ نزدیک می‌شود، که باعث می‌شود tf-idf به ۰. زمانی که مقادیر IDF و TF هر دو بالا باشند، یعنی کلمه در کل سند نادر است. اما در یک سند مکرر است که آن را با ارزش می‌کند.

تمرین ۲:

با استفاده از روش word2vec، بردارهایی که از پیش آموزش داده شده اند را برای هر کلمه استفاده میکنیم.

این کار را با استفاده از مجموعه بردارهای کتابخانه spacy انجام میدهیم. برای هر کدام از plot_synopsis ها و title ها، میانگین بردار مربوط به کلمات آن ها را جایگزین میکنیم. پس داده هایمان به این شکل کاملا عددی می‌شوند:

	local_id	plot_source	genre_imdb	text_features
0	1	0	[1.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, ...	[-0.7138522, 0.073373705, -0.1398024, -0.10776...
1	2	0	[0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, ...	[-0.67880213, 0.08571929, -0.015693752, -0.004...
2	3	0	[0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, ...	[-0.6802255, 0.07970264, -0.024893358, -0.0727...
3	4	0	[0.0, 0.0, 1.0, 0.0, 0.0, 0.0, 0.0, 0.0, ...	[-0.70416576, 0.12861355, -0.06847692, -0.0346...
4	5	0	[0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, ...	[-0.718952, 0.12190968, -0.0152674485, -0.0501...
...	...	...	...	...
1675	1677	1	[0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, ...	[-0.6812775, 0.09497944, -0.14746328, -0.07139...
1677	1679	1	[0.0, 0.0, 1.0, 0.0, 0.0, 0.0, 0.0, 0.0, ...	[-0.70034236, 0.17377904, -0.052543465, -0.060...
1678	1680	0	[0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, ...	[-0.7026811, 0.15511484, -0.055853795, -0.0132...
1679	1681	1	[0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 1.0, ...	[-0.67887, -0.35292, -0.32589, -0.11129, -0.24...
1680	1682	1	[0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, ...	[-0.662062, 0.045938797, 0.01871007, -0.134907...

1596 rows × 4 columns

ستون text_features جدید اضافه شده و جای plot_synopsis و title را گرفته است. همان میانگین embedding تمام کلمات آن هاست. یعنی با یک بردار ۳۰۰ تایی که میانگین کلمات موجود در توضیحات فیلم ها و نام آن هاست، سعی میکنیم محتوای معنایی فیلم را نمایش دهیم. که این کار امکان پذیر است چون embedding ها بردارهایی مبتنی بر معنا هستند.

پردازش داده

سوال ۳:

برخی مزایای روش kmeans عبارتند از:

- سادگی پیاده سازی
- قابلیت scale به دیتاست های بزرگ که چون در اینجا دیتاست نسبتا کوچی داریم پس مناسب است.
- تضمین همگرا شدن به دسته بندی: میخواهیم حتما خوشه بندی انجام شود و نیمه کاره نماند.
- به راحتی با نمونه های جدید سازگار می شود، و ما خیلی خیلی فیلم داریم که میتواند به این دیتا اضافه شود.
- به خوشه هایی با اشکال و اندازه های مختلف، مانند خوشه های بیضی تعمیم می یابد. و ما تضمینی برای دایره ای بودن خوشه هایمان نداریم پس این خیلی مفید است.

اما برخی معایب هم دارد:

- انتخاب دستی k: که سعی کردم با چند بار اجرای اعداد مختلف بین ۲ تا ۱۰ بهترین را برگزینم.
- وابسته بودن به مقادیر اولیه مثل مرکز کلاسترها: بله این مشکل وجود دارد. اما کتابخانه scikit-learn با ارائه اعداد رندومی که شروط خاصی دارند تقریبا این مشکل را حل میکند.
- وابستگی به outlier: که به نوعی مهمترین ضعف آن محسوب میشود و مهم است که اینجا آن را نداریم. چون ستون های Genre و source که بررسی کردیم و فقط برخی مقادیر خاص را داشتند که به جای آن ها ۰ و ۱ گذاشتیم. و ۲ تا ستونی که متن و نم فیلم بودند هم از embedding استفاده کردیم که تمرین داده شده روی دیتای بزرگ برای هر کلمه است و در نتیجه بردارش خطایی ندارد. و میانگین آن ها هم قطعا عدد پرتی نخواهد بود.

تمرین ۳:

با استفاده از kmeans موجود در کتابخانه scikit-learn روی دیتاست کلاسترینگ انجام میدهیم. البته قبل از آن مقادیر را به آرایه تبدیل میکنیم و آیدی را ازآن حذف میکنیم. تا فقط مقادیر عددی فیچرهای مختلف و plot-source بماند. با اعداد مختلفی امتحان کردم و با ۸ label به نتیجه معقولی از جهت تنوع بین کلاستر ها رسیدم.

تعداد اعضای هر کلاستر:

```
Counter({0: 194, 1: 105, 2: 232, 3: 311, 4: 199, 5: 315, 6: 61, 7: 179})
```

نمونه برچسب دار دیتاست اصلی، فقط پس از حذف شدن سطر و ستون های اضافی:

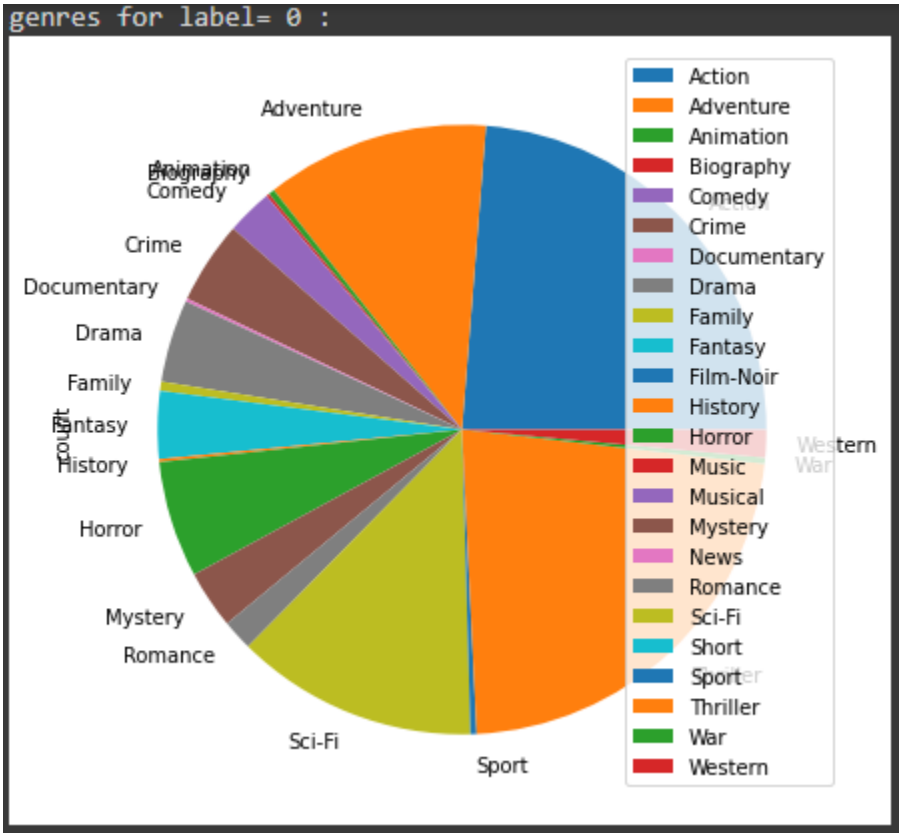
	plot_synopsis	local_id	plot_source	title	genre_imdb	label
0	A boy called Andy Davis (voice: John Morris) u...	1.0	imdb	Toy Story (1995)	Animation|Adventure|Comedy|Family|Fantasy	1
1	The story opens in 1986, in the Cold War Sovie...	2.0	imdb	GoldenEye (1995)	Action|Adventure|Thriller	0
2	The film begins with Ted the Bellhop (Tim Roth...	3.0	imdb	Four Rooms (1995)	Comedy	5
3	Chilli Palmer (John Travolta) is a loan shark ...	4.0	imdb	Get Shorty (1995)	Comedy|Crime|Thriller	2
4	After giving a guest lecture on criminal psych...	5.0	imdb	Copycat (1995)	Drama|Mystery|Thriller	2
...	...	...	...	...	...	...
1675	Angel celebrates the birth of his daughter by ...	1677.0	wiki	Sweet Nothing (1995)	Drama	3
1677	Alan (Jared Harris) is a schoolteacher in Lond...	1679.0	wiki	B. Monkey (1998)	Crime|Drama|Romance|Thriller	2
1678	The film follows Helen Quilley (Gwyneth Paltro...	1680.0	imdb	Sliding Doors (1998)	Comedy|Drama|Fantasy|Romance	4
1679		1681.0	wiki	You So Crazy (1994)	Documentary|Comedy	5
1680		1682.0	wiki	Scream of Stone (Schrei aus Stein) (1991)	Drama	6

1596 rows × 6 columns

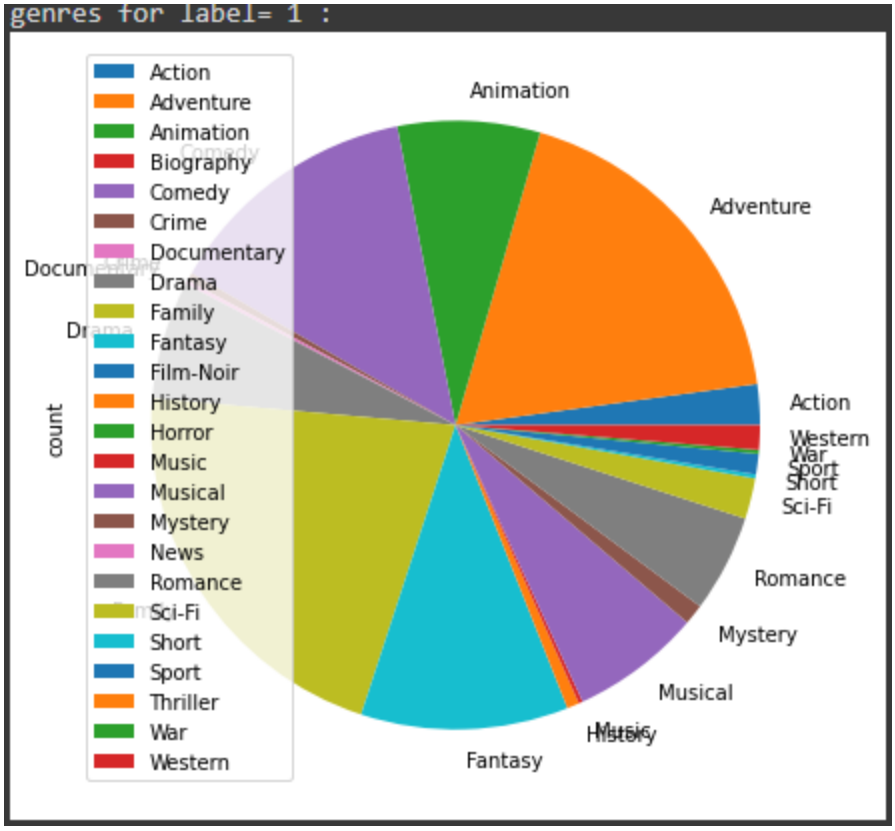
پساپردازش

تمرین ۴:

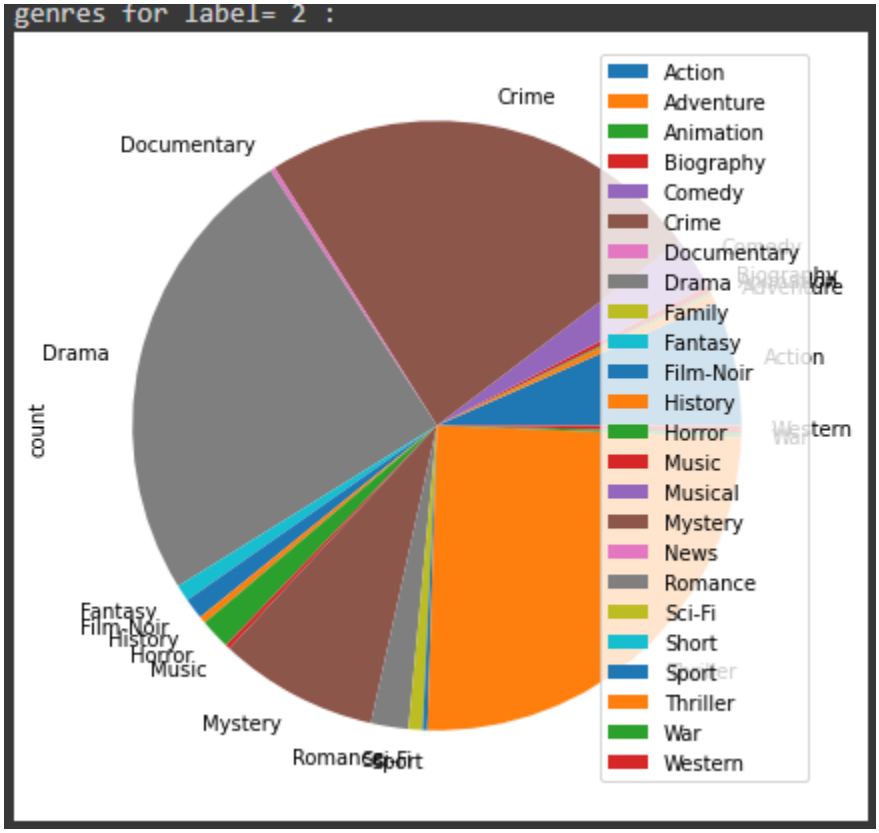
بهترین ویژگی که میتوانیم با نتیجه خوشه بندی آن را مقایسه کنیم همان ژانر فیلم است. که ببینیم چقدر خوب از روی کلمات به تنوع های مختلف ژانری رسیده است. به این منظور نمودارهایی از تنوع ژانرها برای هر خوشه تهیه میکنیم تا بهتر متوجه شویم:



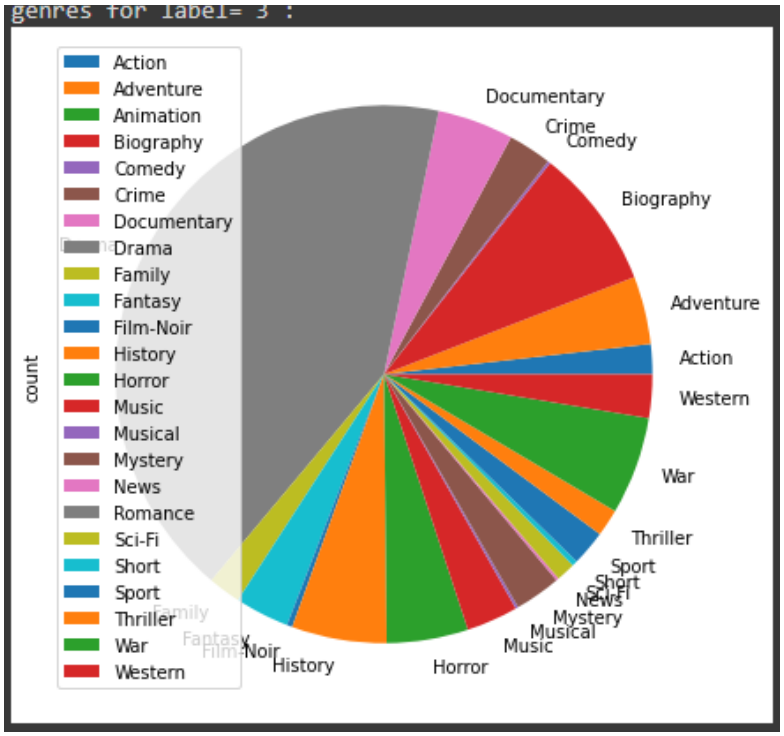
فیلم هایی که حالت اکشن و هیجان انگیز و ماجراجویی دارند در این دسته قرار گرفته اند. که میتوان حس کرد به نوعی محتوای این فیلم ها به علمی تخیلی نیز نزدیک است که تعداد آن هم بالاست.



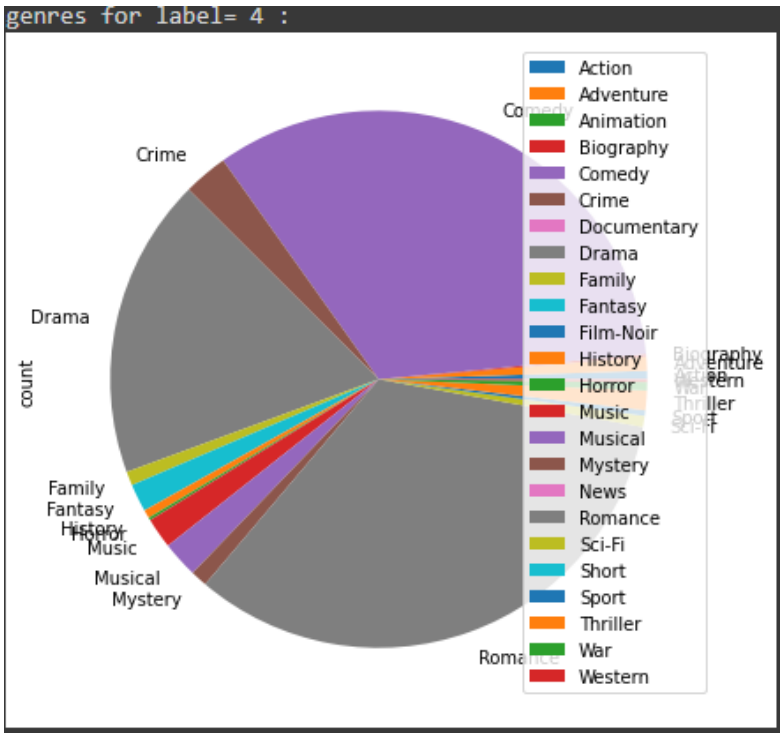
در این دسته فیلم های ماجراجویانه ای که برخلاف قبل به جای محتوای اکشن، محتوای فانتزی و یا علمی تخیلی دارند آمده است. درواقع فیلم هایی آرام تر که بعضا کمدی هم هستند ولی در عین حال ماجراجویی دارند.



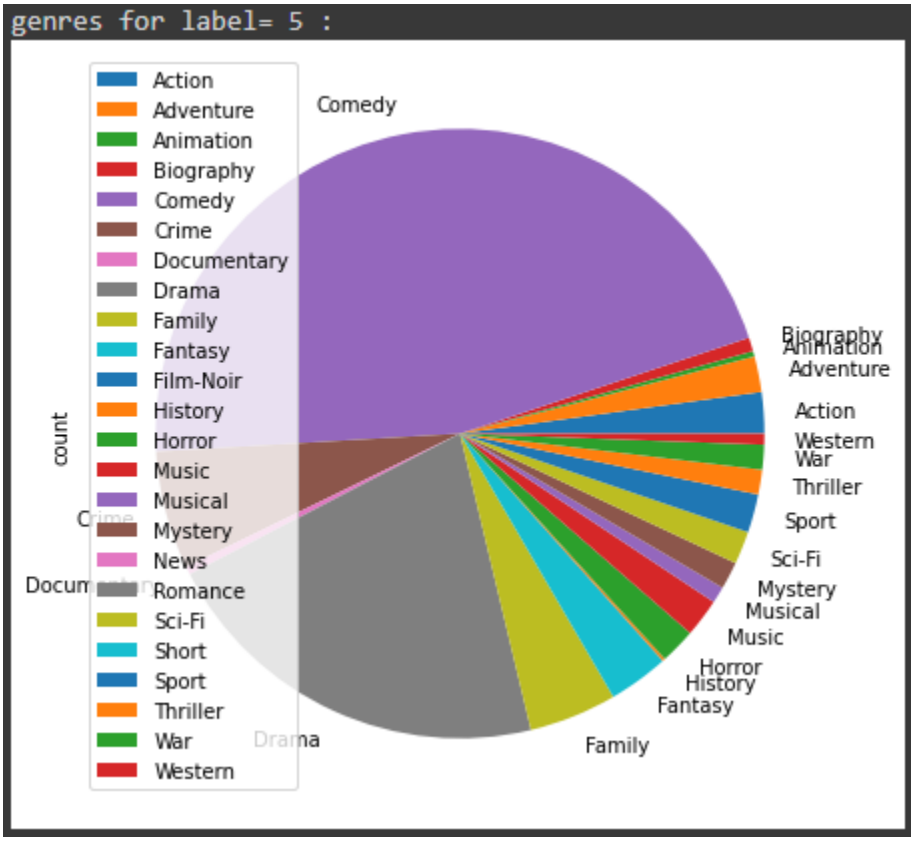
میبینیم که این دسته با قبلی ها کاملا متفاوت است. (چون هر رنگ در تمام نمودارها نشاندهنده یک دسته است، این را میتوان با یک نگاه متوجه شد.) فیلم هایی با ژانر جنایی که حالت معما برانگیز دارند و در کنار آن داستانی عاشقانه هم دارند این دسته را میسازند.



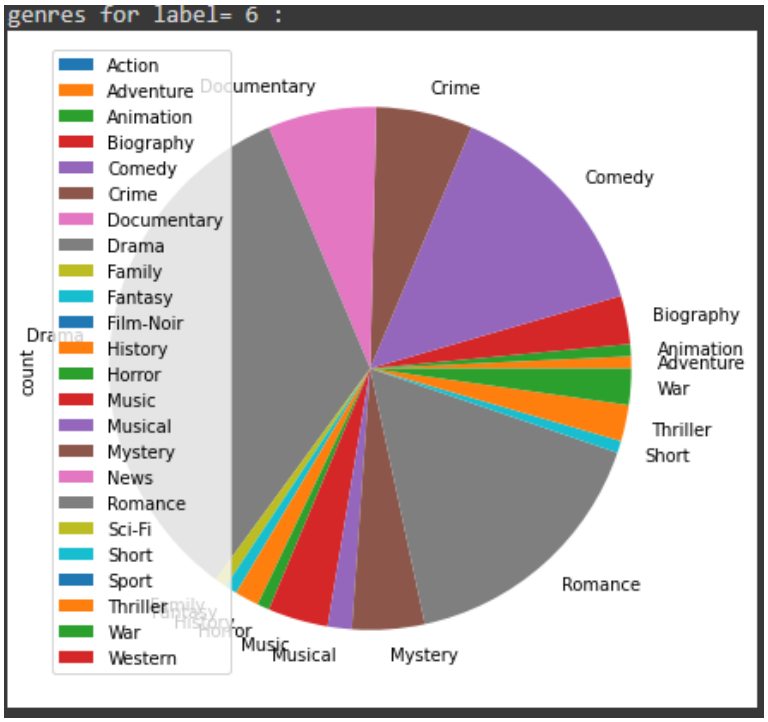
تقریبا ژانر وسیع dram در این دسته قرار گرفته که بعضا با هر ژانر دیگری ترکیب شده است.



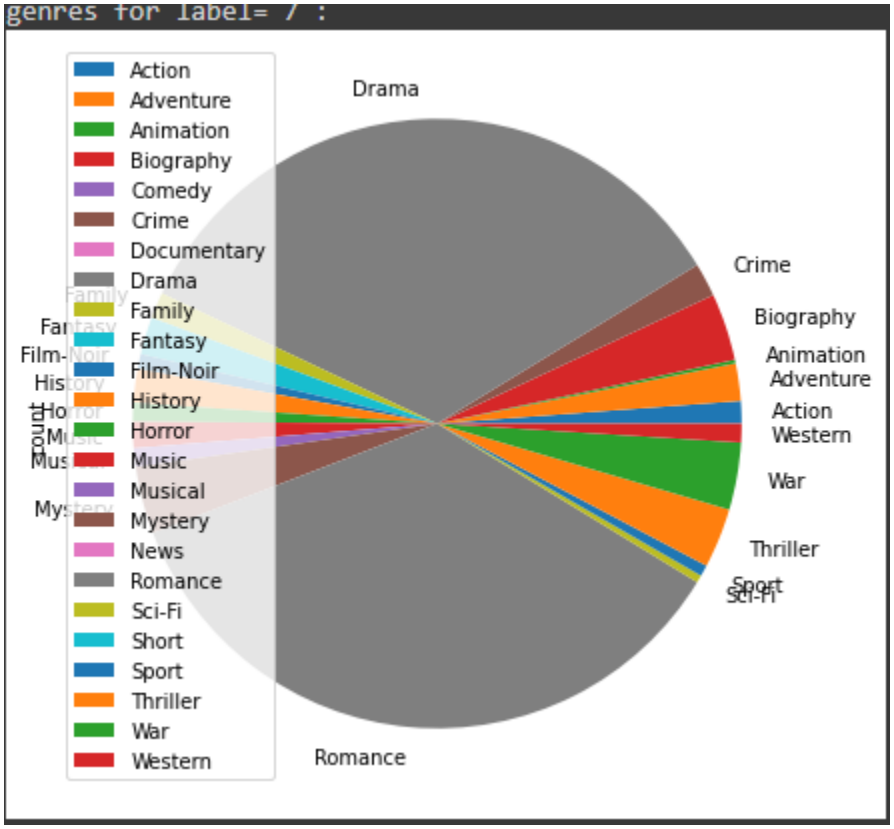
فیلم های کمدی ماجراجویانه را در دسته ۱ دیدیم اما اگر حالت رومانتیک و درام داشته باشند در این دسته قرار میگیرند.



میتوان گفت این دسته تاحدی مشابه دسته قبل است ولی یک تفاوت چشمگیر در تعداد فیلم های romance وجود دارد. که یعنی به خوبی درام های کمدی عاشقانه را از غیر عاشقانه تمییز داده است.



نکته اصلی در مقایسه با دسته های قبلی زیاد شدن مستندهاست. که در کل تعدادشان کم است و اکثرا در این دسته قرار گرفته اند.



میدانیم بخش مهمی از سینما درام های عاشقانه است. که اگر غیر کمدی باشند در این دسته قرار گرفته اند.

منابع

- https://www.baeldung.com/cs/stemming-vs-lemmatization
- https://blog.bitext.com/what-is-the-difference-between-stemming-and-lemmatization/#:~:text=Stemming%20and%20lemmatization%20are%20methods,more%20detailed%20explanations%20and%20examples.
- https://www.geeksforgeeks.org/feature-extraction-techniques-nlp/
- https://developers.google.com/machine-learning/clustering/algorithm/advantages-disadvantages