فاز اول پروژه

پیشبینی رده سنی بر اساس خلاصه فیلم

پردازش زبان و گفتار

استاد درس

دکتر مینایی

دانشجویان

امیرحسین احمدی – ملیکا احمدی رنجبر

بهار ۱۴۰۱

فهرست مطالب

خلاصه فعالیت ها

در فاز اول این پروژه با Crawling روی صفحات مربوط به رده های سنی (MPA) در سایت IMDb اطلاعات مربوط به فیلم و خلاصه فیلم را بدست آورده و ذخیره میکنیم. در ادامه داده های بدست آمده را نرمال سازی کرده و داده های تمیز شده را ذخیره میکنیم. در انتها نیز آمار خواسته شده مربوط به داده ها را بدست میاوریم.

استخراج اطلاعات فیلم از IMDb

ابتدا تابع crawl را تعریف کرده که با گرفتن MPA (G, PG, PG-13, R) یک Request به صفحه مربوط به آن زده و اطلاعات آن را دریافت میکنیم. (در هر صفحه ۵۰ فیلم وجود دارد به همین دلیل Index مربوط به شروع صفحه را در هر Request ۵۰ تا زیاد میکنیم تا تمام فیلم های مربوط به آن رده را دریافت کنیم.) نمونه لینک برای رده سنی PG-13 :

https://www.imdb.com/search/title/?title_type=feature,tv_movie,tv_special,documentary,short,tv_short&certificates=US%3A**PG-13**&start=1

سپس با استفاده از ابزار BeautifulSoup اطلاعات درون صفحه را استخراج کرده و Title، Plot و MPA آن را درون لیست raw_list اضافه میکنیم.

```
def crawl(mpa):
    for current_page in tqdm(range(1, 9952, 50)):
        response = requests.get('https://www.imdb.com/search/title/?'
                                + 'title_type=feature,tv_movie,tv_special,documentary,short,tv_short'
                                + '&certificates=US%3A' + mpa
                                + '&start=' + str(current_page))

        soup = BeautifulSoup(response.text, 'html.parser')

        for i in range(len(soup.select('h3.lister-item-header a'))):
            title = soup.select('h3.lister-item-header a')[i].get_text()
            plot = soup.select('p.text-muted')[2 * i + 1].get_text()
            raw_list.append([title, mpa, plot])
```

در ادامه با صدا زدن تابع crawl برای تمام MPA ها، raw_list را آپدیت میکنیم و با تبدیل raw_list به Pandas DataFrame، ۵ رکورد اول آن را نمایش میدهیم.(در ابتدای هر Plot یک '\n' آمده که آن را نیز در این مرحله حذف میکنیم.)

```
crawl('G')
```

```
100%|██████████| 200/200 [03:32<00:00,  1.06s/it]
```

```
crawl('PG')
```

```
100%|██████████| 200/200 [05:15<00:00,  1.58s/it]
```

```
crawl('PG-13')
```

```
100%|██████████| 200/200 [05:08<00:00,  1.54s/it]
```

```
crawl('R')
```

```
100%|██████████| 200/200 [05:16<00:00,  1.58s/it]
```

```
raw_data = pd.DataFrame(raw_list, columns = ['Title', 'MPA', 'Plot'])
raw_data.Plot = raw_data.Plot.apply(lambda p: p.replace('\n', ''))
raw_data.head()
```

	Title	MPA	Plot
0	The Lion King	G	Lion prince Simba and his father are targeted ...
1	Cars	G	A hot-shot race-car named Lightning McQueen ge...
2	Luck	G	The curtain is pulled back on the millennia-ol...
3	2001: A Space Odyssey	G	The Monoliths push humanity to reach for the s...
4	Ratatouille	G	A rat who can cook makes an unusual alliance w...

در آخر با ساخت پوشه های مورد نیاز DataFrame ساخته شده را در قالب csv ذخیره میکنیم.

```
if not os.path.isdir(base_dir + 'data/'):
    os.mkdir(base_dir + 'data/')

if not os.path.isdir(base_dir + 'data/raw/'):
    os.mkdir(base_dir + 'data/raw/')

raw_data.to_csv(base_dir + 'data/raw/data.csv')
```

پیش پردازش

در ابتدا داده خام ذخیره شده در مرحله قبل را میخوانیم.

```
df = pd.read_csv(base_dir + 'data/raw/data.csv', index_col=0)
df
```

	Title	MPA	Plot
0	The Lion King	G	Lion prince Simba and his father are targeted ...
1	Cars	G	A hot-shot race-car named Lightning McQueen ge...
2	Luck	G	The curtain is pulled back on the millennia-ol...
3	2001: A Space Odyssey	G	The Monoliths push humanity to reach for the s...
4	Ratatouille	G	A rat who can cook makes an unusual alliance w...
...	...	...	...
28105	Zombie Diaries	R	An unknown virus begins spreading and within w...
28106	Gangsta Rap: The Glockumentary	R	The hardest group you've never heard of is bac...
28107	Satan's Sadists	R	The "Satans" are a very cruel biker gang led b...
28108	Train of Life	R	In 1941, the inhabitants of a small Jewish vil...
28109	The Devil's Female	R	After the gruesome death of her father, a youn...

28110 rows × 3 columns

سپس داده های مورد نیاز کتابخانه nltk را دریافت کرده که در ادامه از آن ها استفاده خواهیم کرد.

```
nltk.download('stopwords')
nltk.download('punkt')
nltk.download('wordnet')
```

```
[nltk_data] Downloading package stopwords to /root/nltk_data...
[nltk_data]   Unzipping corpora/stopwords.zip.
[nltk_data] Downloading package punkt to /root/nltk_data...
[nltk_data]   Unzipping tokenizers/punkt.zip.
[nltk_data] Downloading package wordnet to /root/nltk_data...
[nltk_data]   Unzipping corpora/wordnet.zip.
True
```

بعضی از فیلم های موجود در IMDb خلاصه فیلم ندارند که آن ها را از داده ها حذف میکنیم. همچنین بعضی از خلاصه ها به دلیل بلند بودن در صفحه ها نیامده اند و عبارت ‘See full summary’ در انتهای Plot آن ها نوشته شده که این عبارت را از خلاصه ها حذف میکنیم.

در ادامه Normalization های مرسوم را با استفاده از nltk و کتابخانه string در Python روی داده ها اعمال میکنیم. ابتدا تمام کلمات Plot ها را Lowercase کرده، سپس علامت گذاری ها و اعداد را حذف میکنیم. در نهایت Plot ها را Tokenized کرده، Stop Word ها را حذف کرده و عمل Lemmatization را روی آن انجام میدهیم و در ستون Normalized_Plot ذخیره میکنیم.

```
df = df.drop(df[df.Plot == 'Add a Plot'].index)
df = df.reset_index().drop(columns = 'index')
df['Plot'] = df['Plot'].str.replace('See full summary', '')

df['Normalized_Plot'] = df['Plot'].str.lower()
df['Normalized_Plot'] = df['Normalized_Plot'].str.translate(str.maketrans('', '', string.punctuation + '»' + '«'))
df['Normalized_Plot'] = df['Normalized_Plot'].str.translate(str.maketrans('', '', string.digits))
df['Normalized_Plot'] = df['Normalized_Plot'].apply(word_tokenize)
df['Normalized_Plot'] = df['Normalized_Plot'].apply(lambda lst : [word for word in lst if word not in set(stopwords.words('english'))])
df['Normalized_Plot'] = df['Normalized_Plot'].apply(lambda lst : [WordNetLemmatizer().lemmatize(w) for w in lst])
```

در آخر با ساخت پوشه های مورد نیاز داده های تمیز شده را در قالب **csv** ذخیره میکنیم.

```
if not os.path.isdir(base_dir + 'data/'):
    os.mkdir(base_dir + 'data/')

if not os.path.isdir(base_dir + 'data/cleaned/'):
    os.mkdir(base_dir + 'data/cleaned/')

df.to_csv(base_dir + 'data/cleaned/data.csv')
```

آمار

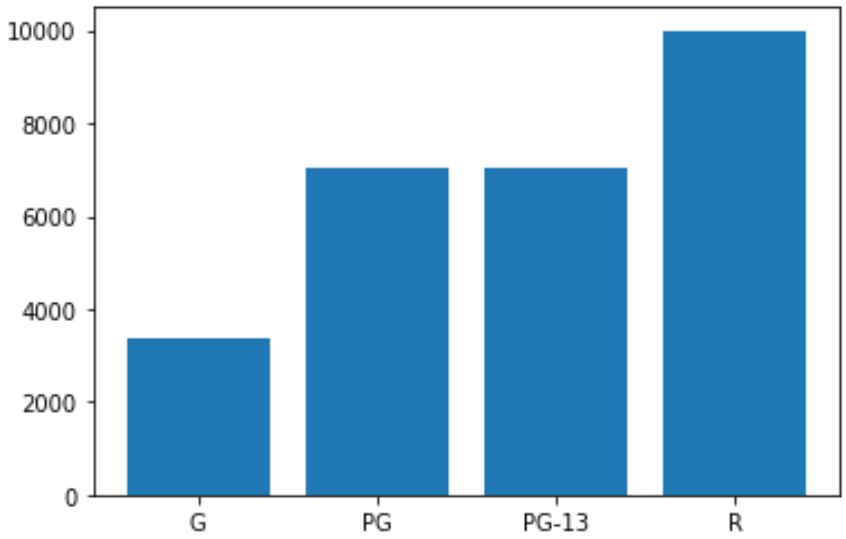
ابتدا داده های تمیز شده در مرحله قبل را میخوانیم.

```
df = pd.read_csv(base_dir + 'data/cleaned/data.csv', index_col=0)
df
```

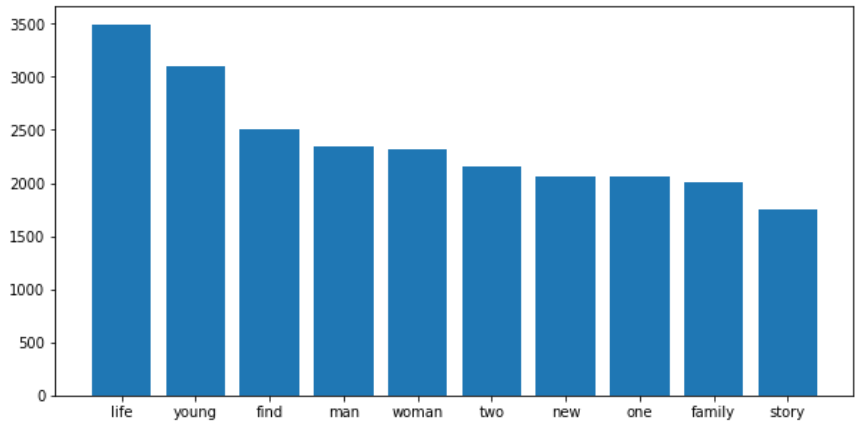
	Title	MPA	Plot	Normalized_Plot
0	The Lion King	G	Lion prince Simba and his father are targeted ...	['lion', 'prince', 'simba', 'father', 'targete...
1	Cars	G	A hot-shot race-car named Lightning McQueen ge...	['hotshot', 'racecar', 'named', 'lightning', '...
2	Luck	G	The curtain is pulled back on the millennia-ol...	['curtain', 'pulled', 'back', 'millenniaold', ...
3	2001: A Space Odyssey	G	The Monoliths push humanity to reach for the s...	['monolith', 'push', 'humanity', 'reach', 'sta...
4	Ratatouille	G	A rat who can cook makes an unusual alliance w...	['rat', 'cook', 'make', 'unusual', 'alliance',...
...	...	...	...	...
27396	Zombie Diaries	R	An unknown virus begins spreading and within w...	['unknown', 'virus', 'begin', 'spreading', 'wi...
27397	Gangsta Rap: The Glockumentary	R	The hardest group you've never heard of is bac...	['hardest', 'group', 'youve', 'never', 'heard'...
27398	Satan's Sadists	R	The "Satans" are a very cruel biker gang led b...	['satan', 'cruel', 'biker', 'gang', 'led', 'an...
27399	Train of Life	R	In 1941, the inhabitants of a small Jewish vil...	['inhabitant', 'small', 'jewish', 'village', '...
27400	The Devil's Female	R	After the gruesome death of her father, a youn...	['gruesome', 'death', 'father', 'young', 'beau...

27401 rows × 4 columns

سپس تعداد داده های مربوط به هر MPA را درون یک Bar Plot نمایش میدهیم که در زیر مشاهده میکنید. همان طور که پیداست تعداد فیلم های با درجه سنی G در IMDb از بقیه کمتر و بیشترین فیلم به درجه سنی R تعلق دارد.



در ادامه با استفاده از تابع Counter در کتابخانه collections تعداد کلمات موجود در Dataset را شمرده و در Bat Plot زیر نمایش داده ایم.



در آخر تعداد جمله ها، تعداد کل کلمات بعد از Normalize کردن داده ها و تعداد کلمات یکتا را میتوانید مشاهده کنید.

```
print('Sentence count:', df['Plot'].apply(sent_tokenize).apply(len).sum())
print('All words(preprocessed words):', sum(counter.values()))
print('Unique words:', len(counter))
```

```
Sentence count: 39327
All words(preprocessed words): 422248
Unique words: 34538
```