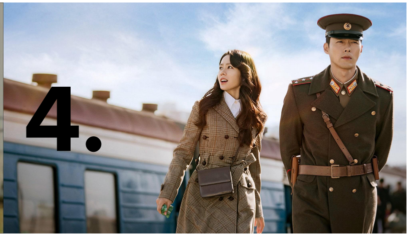
SAAS

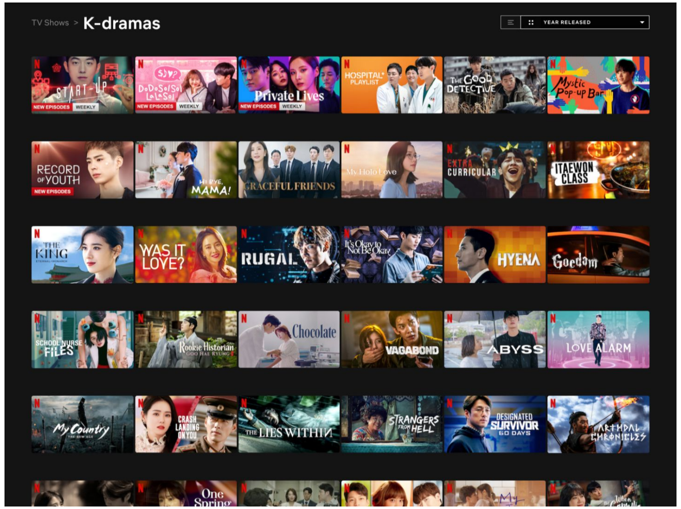
Presentation Contents

K-drama Recommendation System



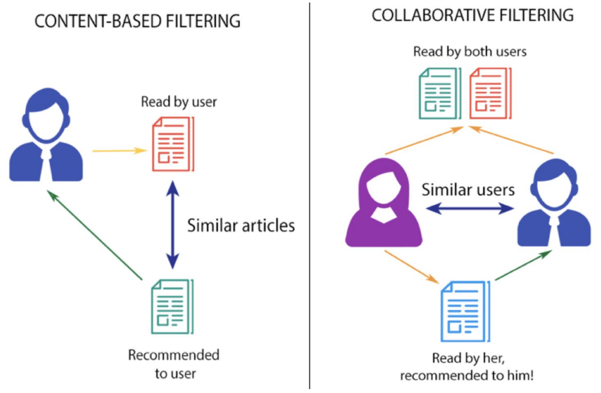
SAAS

Project Introduction



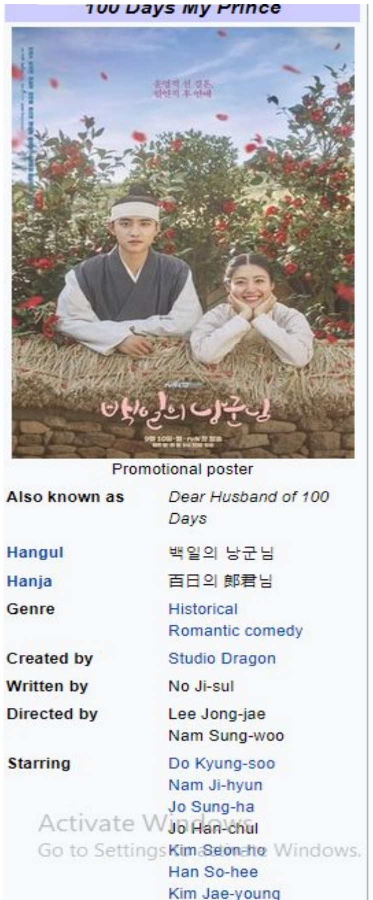
Problem: Understand how products/services are recommended to people

Applications: Game Recommendations, Movie Recommendations, Song Recommendation, many more!!!



SAAS

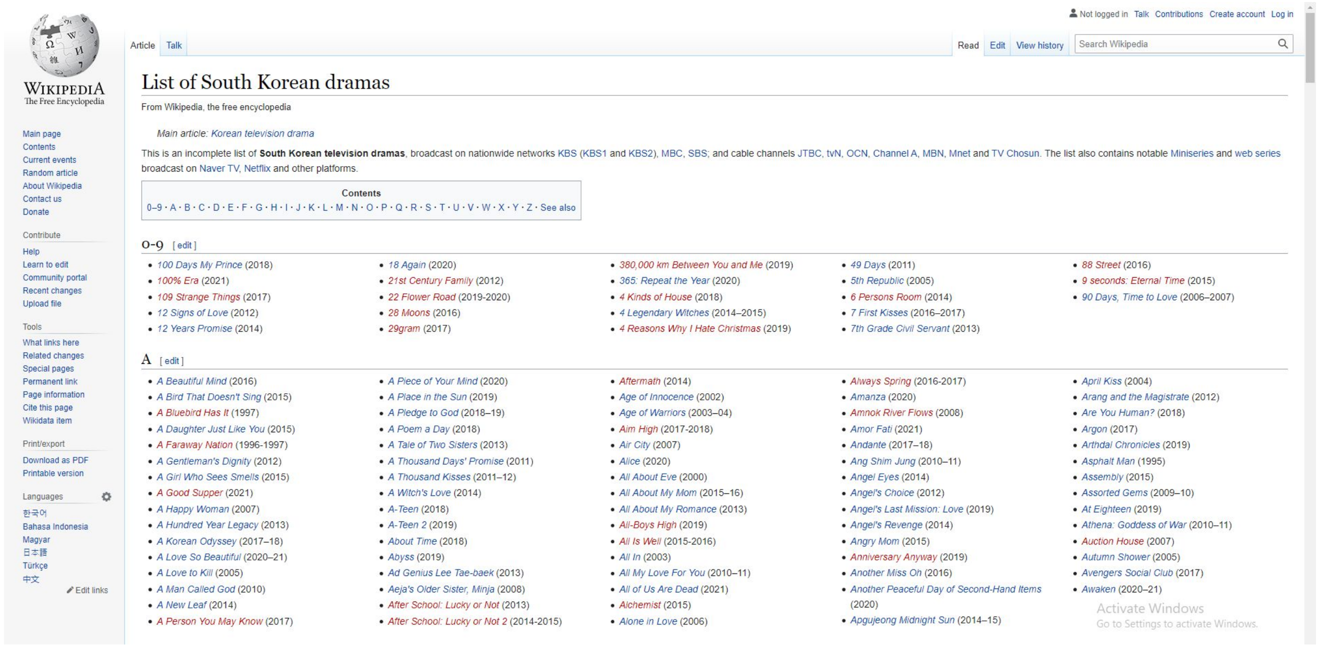
Data Collection



100 Days My Prince

Promotional poster

Also known as	Dear Husband of 100 Days
Hangul	백일의 낭군님
Hanja	百日의 郎君님
Genre	Historical Romantic comedy
Created by	Studio Dragon
Written by	No Ji-sul
Directed by	Lee Jong-jae Nam Sung-woo
Starring	Do Kyung-soo Nam Ji-hyun Jo Sung-ha Jo Han-chul Kim Seon-ho Han So-hee Kim Jae-young



Not logged in Talk Contributions Create account Log in

Article Talk Read Edit View history Search Wikipedia

WIKIPEDIA
The Free Encyclopedia

Main page
Contents
Current events
Random article
About Wikipedia
Contact us
Donate

Contribute
Help
Learn to edit
Community portal
Recent changes
Upload file

Tools
What links here
Related changes
Special pages
Permanent link
Page information
Cite this page
Wikidata item

Print/export
Download as PDF
Printable version

Languages
한국어
Bahasa Indonesia
Magyar
日本語
Türkçe
中文
Edit links

List of South Korean dramas

From Wikipedia, the free encyclopedia

Main article: Korean television drama

This is an incomplete list of **South Korean television dramas**, broadcast on nationwide networks KBS (KBS1 and KBS2), MBC, SBS; and cable channels JTBC, tvN, OCN, Channel A, MBN, Mnet and TV Chosun. The list also contains notable Miniseries and web series broadcast on Naver TV, Netflix and other platforms.

Contents
0–9 · A · B · C · D · E · F · G · H · I · J · K · L · M · N · O · P · Q · R · S · T · U · V · W · X · Y · Z · See also

0-9 [edit]

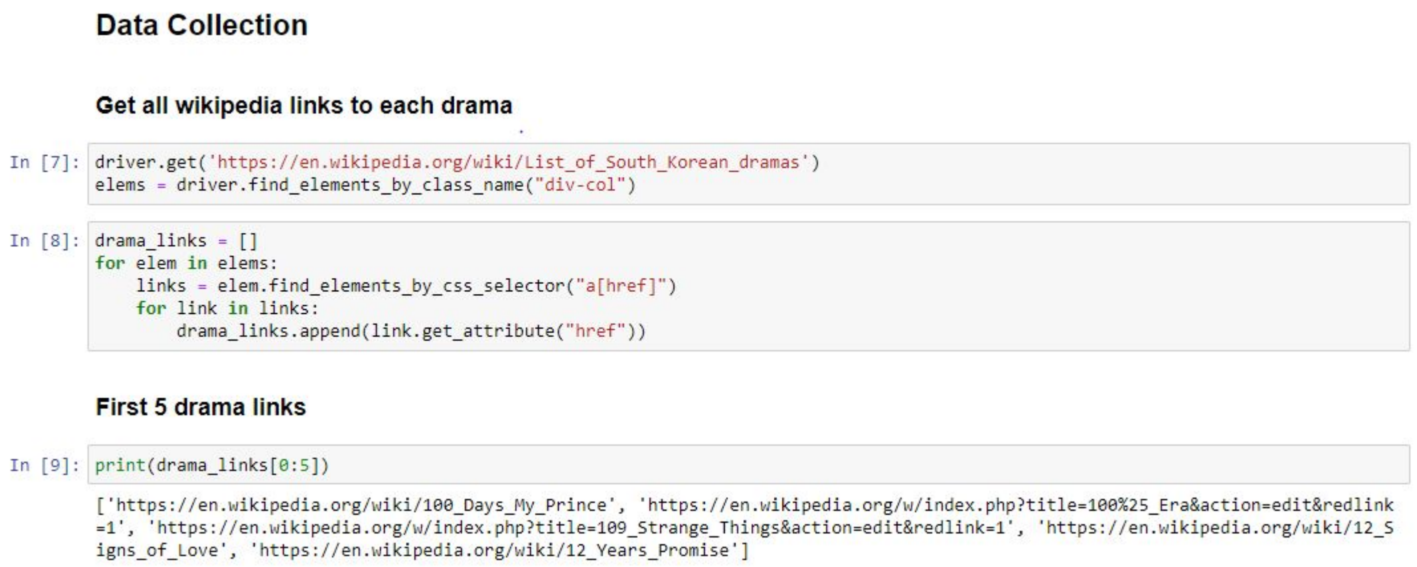
- *100 Days My Prince* (2018)
- *100% Era* (2021)
- *109 Strange Things* (2017)
- *12 Signs of Love* (2012)
- *12 Years Promise* (2014)
- *18 Again* (2020)
- *21st Century Family* (2012)
- *22 Flower Road* (2019-2020)
- *28 Moons* (2016)
- *29gram* (2017)
- *380,000 km Between You and Me* (2019)
- *365: Repeat the Year* (2020)
- *4 Kinds of House* (2018)
- *4 Legendary Witches* (2014–2015)
- *4 Reasons Why I Hate Christmas* (2019)
- *49 Days* (2011)
- *5th Republic* (2005)
- *6 Persons Room* (2014)
- *7 First Kisses* (2016–2017)
- *7th Grade Civil Servant* (2013)
- *88 Street* (2016)
- *9 seconds: Eternal Time* (2015)
- *90 Days, Time to Love* (2006–2007)

A [edit]

- *A Beautiful Mind* (2016)
- *A Bird That Doesn't Sing* (2015)
- *A Bluebird Has It* (1997)
- *A Daughter Just Like You* (2015)
- *A Faraway Nation* (1996-1997)
- *A Gentleman's Dignity* (2012)
- *A Girl Who Sees Smells* (2015)
- *A Good Supper* (2021)
- *A Happy Woman* (2007)
- *A Hundred Year Legacy* (2013)
- *A Korean Odyssey* (2017–18)
- *A Love So Beautiful* (2020–21)
- *A Love to Kill* (2005)
- *A Man Called God* (2010)
- *A New Leaf* (2014)
- *A Person You May Know* (2017)
- *A Piece of Your Mind* (2020)
- *A Place in the Sun* (2019)
- *A Pledge to God* (2018–19)
- *A Poem a Day* (2018)
- *A Tale of Two Sisters* (2013)
- *A Thousand Days' Promise* (2011)
- *A Thousand Kisses* (2011–12)
- *A Witch's Love* (2014)
- *A-Teen* (2018)
- *A-Teen 2* (2019)
- *About Time* (2018)
- *Abyss* (2019)
- *Ad Genius Lee Tae-baek* (2013)
- *Aeja's Older Sister, Minja* (2008)
- *After School: Lucky or Not* (2013)
- *After School: Lucky or Not 2* (2014-2015)
- *Aftermath* (2014)
- *Age of Innocence* (2002)
- *Age of Warriors* (2003–04)
- *Aim High* (2017-2018)
- *Air City* (2007)
- *Alice* (2020)
- *All About Eve* (2000)
- *All About My Mom* (2015–16)
- *All About My Romance* (2013)
- *All-Boys High* (2019)
- *All Is Well* (2015-2016)
- *All In* (2003)
- *All My Love For You* (2010–11)
- *All of Us Are Dead* (2021)
- *Alchemist* (2015)
- *Alone in Love* (2006)
- *Always Spring* (2016-2017)
- *Amanza* (2020)
- *Amnok River Flows* (2008)
- *Amor Fati* (2021)
- *Andante* (2017–18)
- *Ang Shim Jung* (2010–11)
- *Angel Eyes* (2014)
- *Angel's Choice* (2012)
- *Angel's Last Mission: Love* (2019)
- *Angel's Revenge* (2014)
- *Angry Mom* (2015)
- *Anniversary Anyway* (2019)
- *Another Miss Oh* (2016)
- *Another Peaceful Day of Second-Hand Items* (2020)
- *Apgujeong Midnight Sun* (2014–15)
- *April Kiss* (2004)
- *Arang and the Magistrate* (2012)
- *Are You Human?* (2018)
- *Argon* (2017)
- *Arthdal Chronicles* (2019)
- *Asphalt Man* (1995)
- *Assembly* (2015)
- *Assorted Gems* (2009–10)
- *At Eighteen* (2019)
- *Athena: Goddess of War* (2010–11)
- *Auction House* (2007)
- *Autumn Shower* (2005)
- *Avengers Social Club* (2017)
- *Awaken* (2020–21)

SAAS

Data Collection



Data Collection

Get all wikipedia links to each drama

```
In [7]: driver.get('https://en.wikipedia.org/wiki/List_of_South_Korean_dramas')
        elems = driver.find_elements_by_class_name("div-col")
```

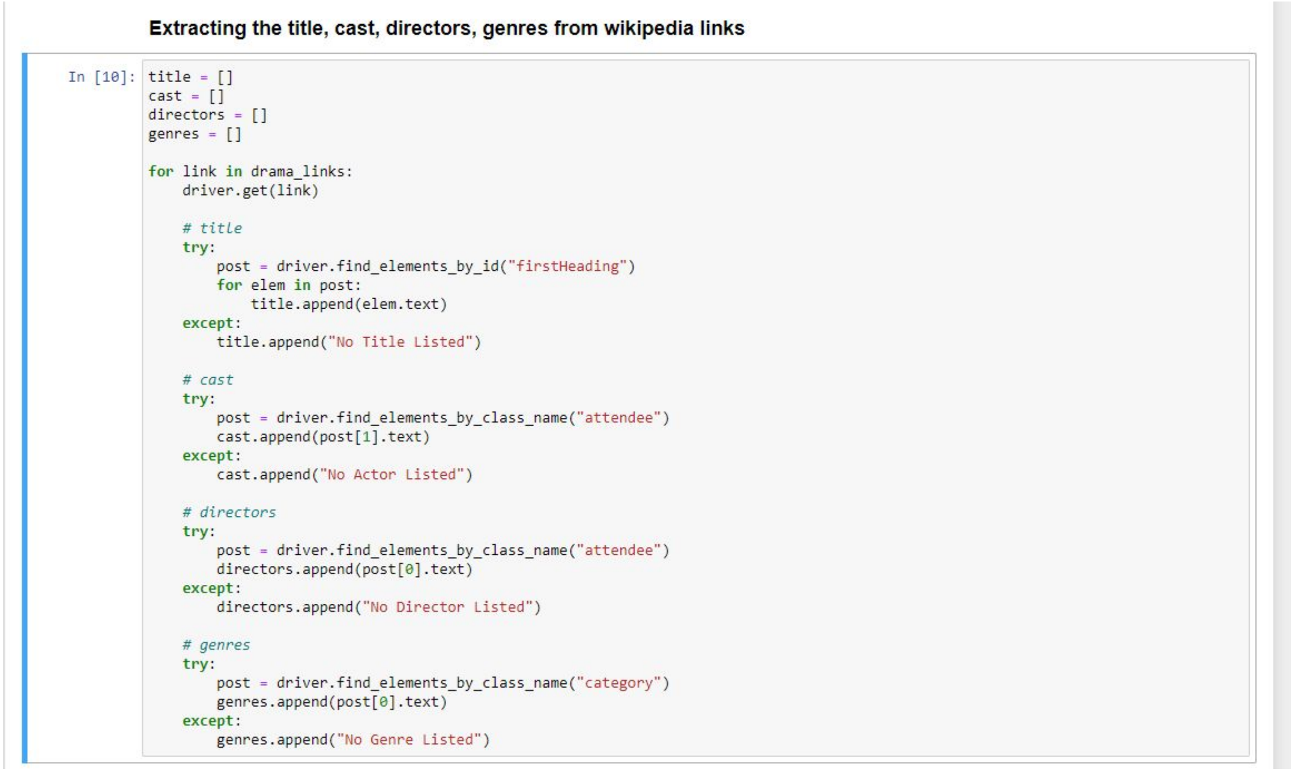
```
In [8]: drama_links = []
        for elem in elems:
            links = elem.find_elements_by_css_selector("a[href]")
            for link in links:
                drama_links.append(link.get_attribute("href"))
```

First 5 drama links

```
In [9]: print(drama_links[0:5])
```

```
['https://en.wikipedia.org/wiki/100_Days_My_Prince', 'https://en.wikipedia.org/w/index.php?title=100%25_Era&action=edit&redlink
=1', 'https://en.wikipedia.org/w/index.php?title=109_Strange_Things&action=edit&redlink=1', 'https://en.wikipedia.org/wiki/12_S
igns_of_Love', 'https://en.wikipedia.org/wiki/12_Years_Promise']
```

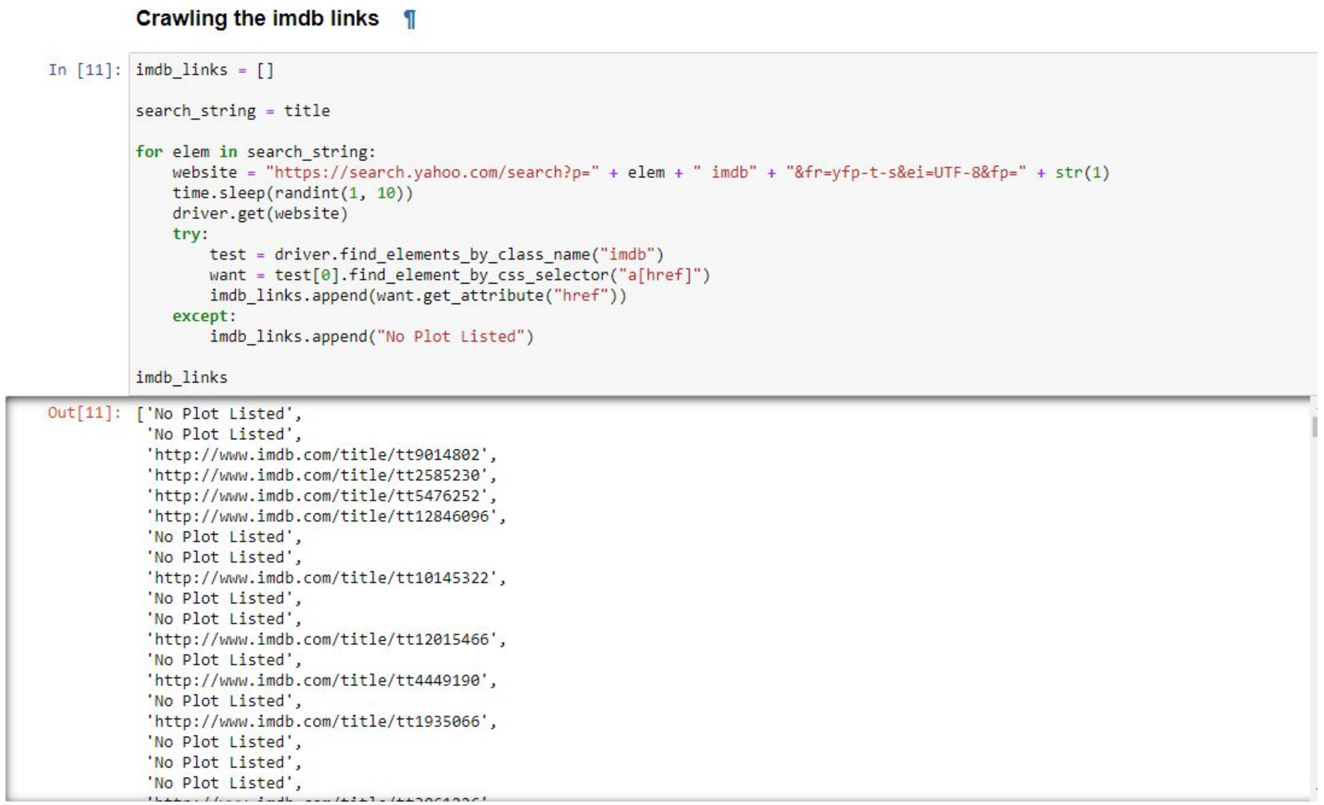
Data Collection



Extracting the title, cast, directors, genres from wikipedia links

In [10]:
```
title = []
cast = []
directors = []
genres = []

for link in drama_links:
    driver.get(link)

    # title
    try:
        post = driver.find_elements_by_id("firstHeading")
        for elem in post:
            title.append(elem.text)
    except:
        title.append("No Title Listed")

    # cast
    try:
        post = driver.find_elements_by_class_name("attendee")
        cast.append(post[1].text)
    except:
        cast.append("No Actor Listed")

    # directors
    try:
        post = driver.find_elements_by_class_name("attendee")
        directors.append(post[0].text)
    except:
        directors.append("No Director Listed")

    # genres
    try:
        post = driver.find_elements_by_class_name("category")
        genres.append(post[0].text)
    except:
        genres.append("No Genre Listed")
```

Data Collection

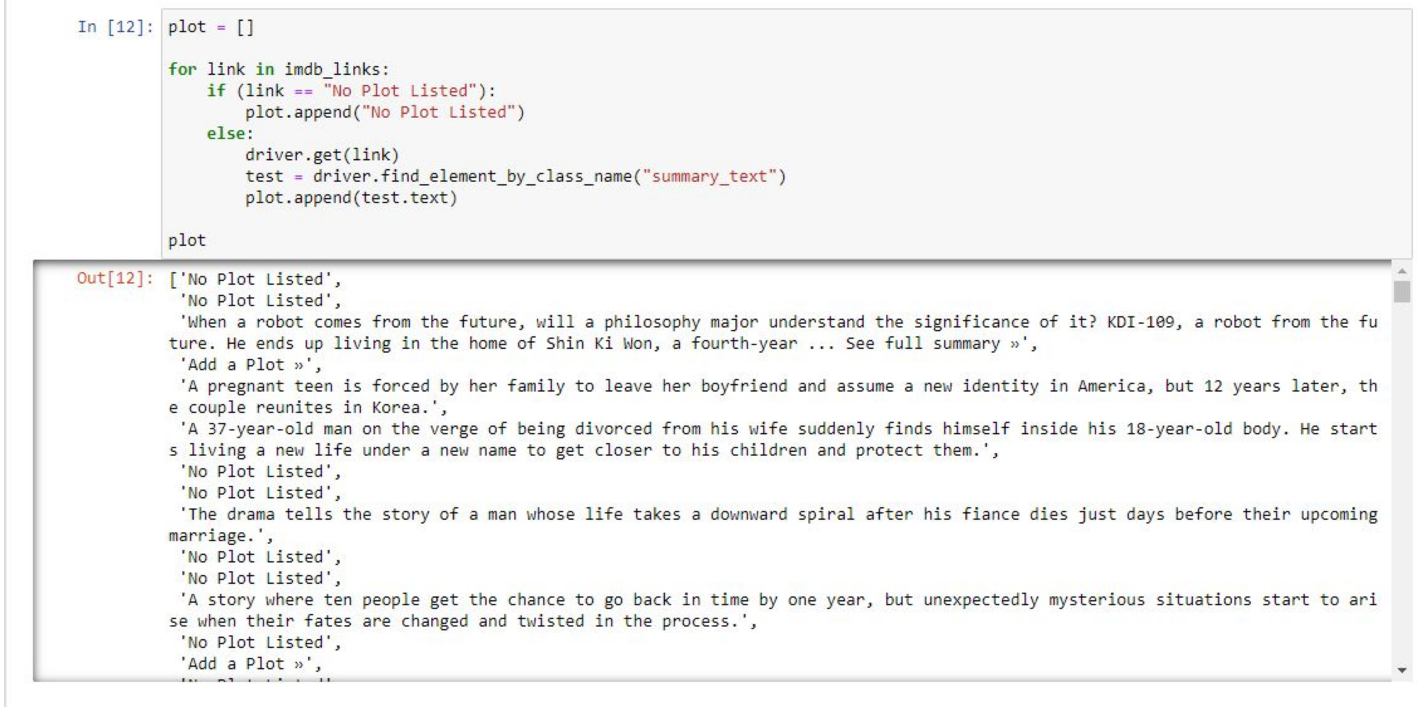


Crawling the imdb links ¶

```
In [11]: imdb_links = []

         search_string = title

         for elem in search_string:
             website = "https://search.yahoo.com/search?p=" + elem + " imdb" + "&fr=yfp-t-s&ei=UTF-8&fp=" + str(1)
             time.sleep(randint(1, 10))
             driver.get(website)
             try:
                 test = driver.find_elements_by_class_name("imdb")
                 want = test[0].find_element_by_css_selector("a[href]")
                 imdb_links.append(want.get_attribute("href"))
             except:
                 imdb_links.append("No Plot Listed")

         imdb_links
```

```
Out[11]: ['No Plot Listed',
          'No Plot Listed',
          'http://www.imdb.com/title/tt9014802',
          'http://www.imdb.com/title/tt2585230',
          'http://www.imdb.com/title/tt5476252',
          'http://www.imdb.com/title/tt12846096',
          'No Plot Listed',
          'No Plot Listed',
          'http://www.imdb.com/title/tt10145322',
          'No Plot Listed',
          'No Plot Listed',
          'http://www.imdb.com/title/tt12015466',
          'No Plot Listed',
          'http://www.imdb.com/title/tt4449190',
          'No Plot Listed',
          'http://www.imdb.com/title/tt1935066',
          'No Plot Listed',
          'No Plot Listed',
          'No Plot Listed',
```

Data Collection

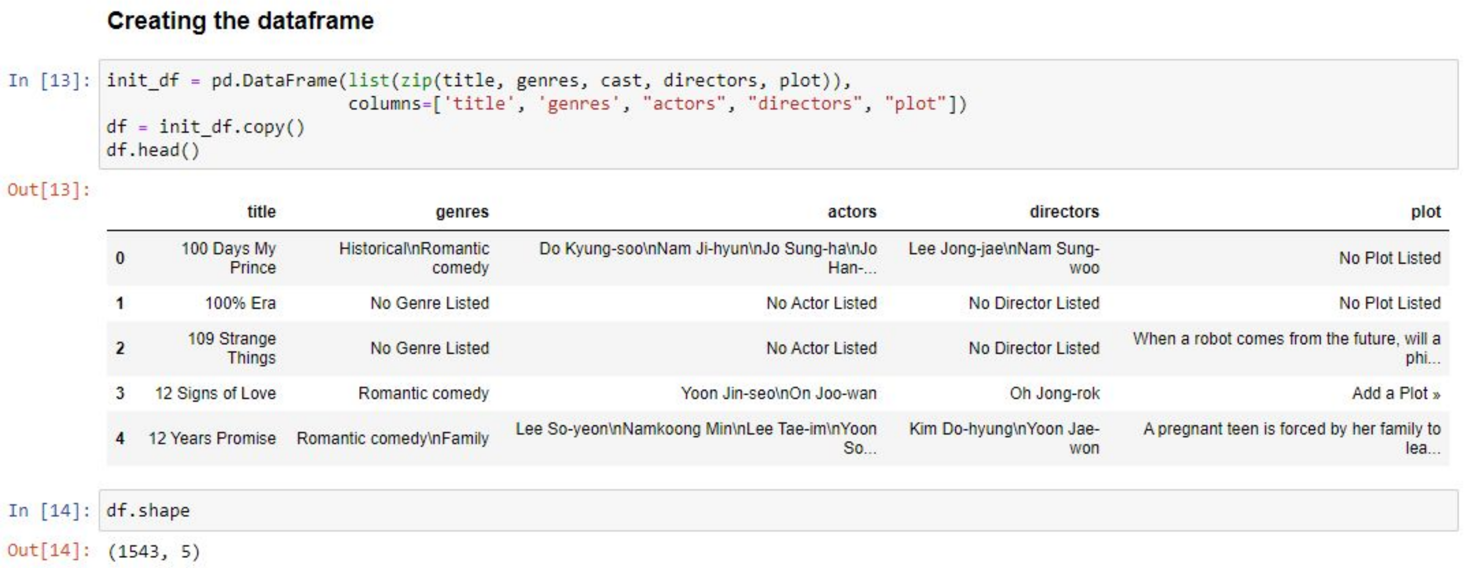


```
In [12]: plot = []

         for link in imdb_links:
             if (link == "No Plot Listed"):
                 plot.append("No Plot Listed")
             else:
                 driver.get(link)
                 test = driver.find_element_by_class_name("summary_text")
                 plot.append(test.text)

         plot
```

```
Out[12]: ['No Plot Listed',
          'No Plot Listed',
          'When a robot comes from the future, will a philosophy major understand the significance of it? KDI-109, a robot from the fu
         ture. He ends up living in the home of Shin Ki Won, a fourth-year ... See full summary »',
          'Add a Plot »',
          'A pregnant teen is forced by her family to leave her boyfriend and assume a new identity in America, but 12 years later, th
         e couple reunites in Korea.',
          'A 37-year-old man on the verge of being divorced from his wife suddenly finds himself inside his 18-year-old body. He start
         s living a new life under a new name to get closer to his children and protect them.',
          'No Plot Listed',
          'No Plot Listed',
          'The drama tells the story of a man whose life takes a downward spiral after his fiance dies just days before their upcoming
         marriage.',
          'No Plot Listed',
          'No Plot Listed',
          'A story where ten people get the chance to go back in time by one year, but unexpectedly mysterious situations start to ari
         se when their fates are changed and twisted in the process.',
          'No Plot Listed',
          'Add a Plot »',
```

SAAS

Data Collection



Creating the dataframe

```
In [13]: init_df = pd.DataFrame(list(zip(title, genres, cast, directors, plot)),
                                columns=['title', 'genres', "actors", "directors", "plot"])
         df = init_df.copy()
         df.head()
```

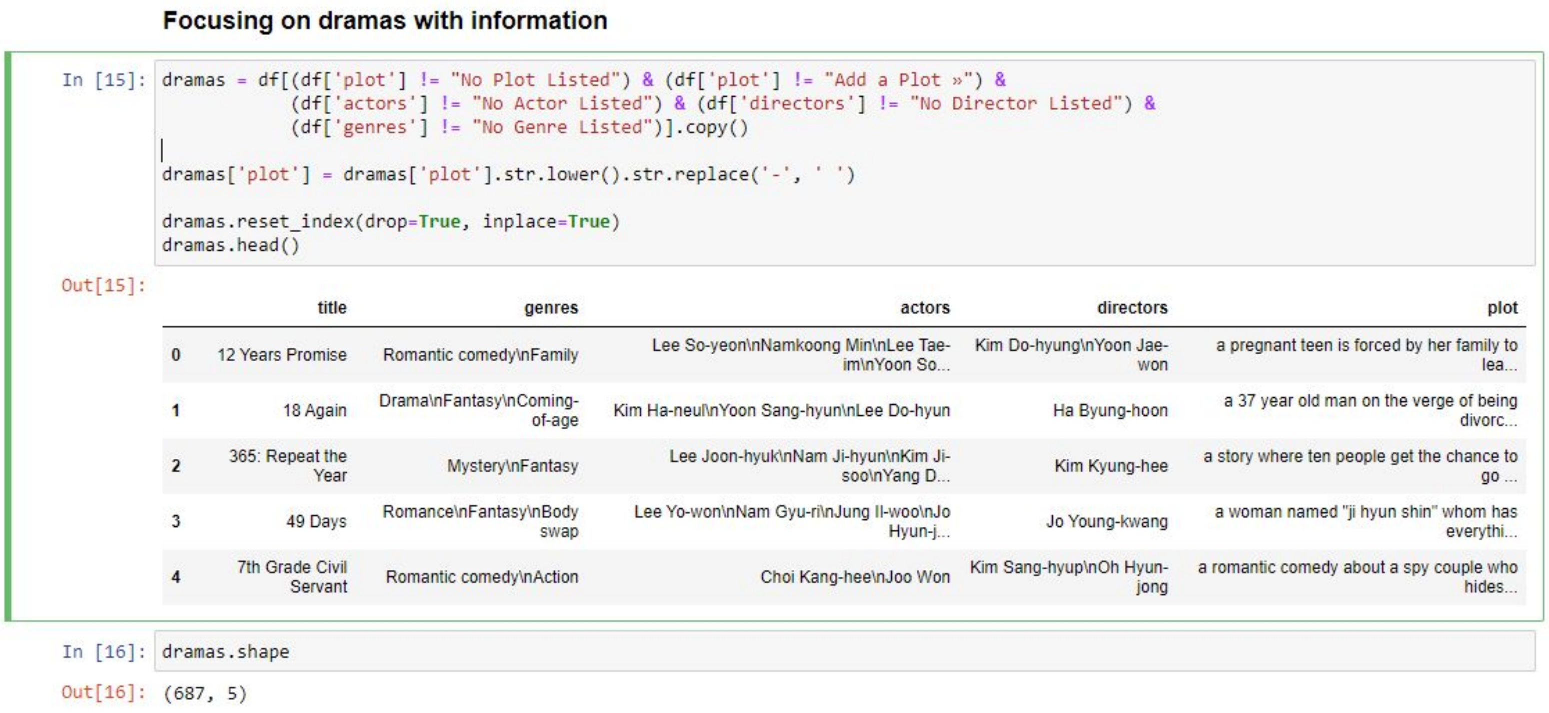
Out[13]:

	title	genres	actors	directors	plot
0	100 Days My Prince	Historical\nRomantic comedy	Do Kyung-soo\nNam Ji-hyun\nJo Sung-ha\nJo Han-...	Lee Jong-jae\nNam Sung-woo	No Plot Listed
1	100% Era	No Genre Listed	No Actor Listed	No Director Listed	No Plot Listed
2	109 Strange Things	No Genre Listed	No Actor Listed	No Director Listed	When a robot comes from the future, will a phi...
3	12 Signs of Love	Romantic comedy	Yoon Jin-seo\nOn Joo-wan	Oh Jong-rok	Add a Plot »
4	12 Years Promise	Romantic comedy\nFamily	Lee So-yeon\nNamkoong Min\nLee Tae-im\nYoon So...	Kim Do-hyung\nYoon Jae-won	A pregnant teen is forced by her family to lea...

```
In [14]: df.shape

Out[14]: (1543, 5)
```

SAAS

Data Collection



Focusing on dramas with information

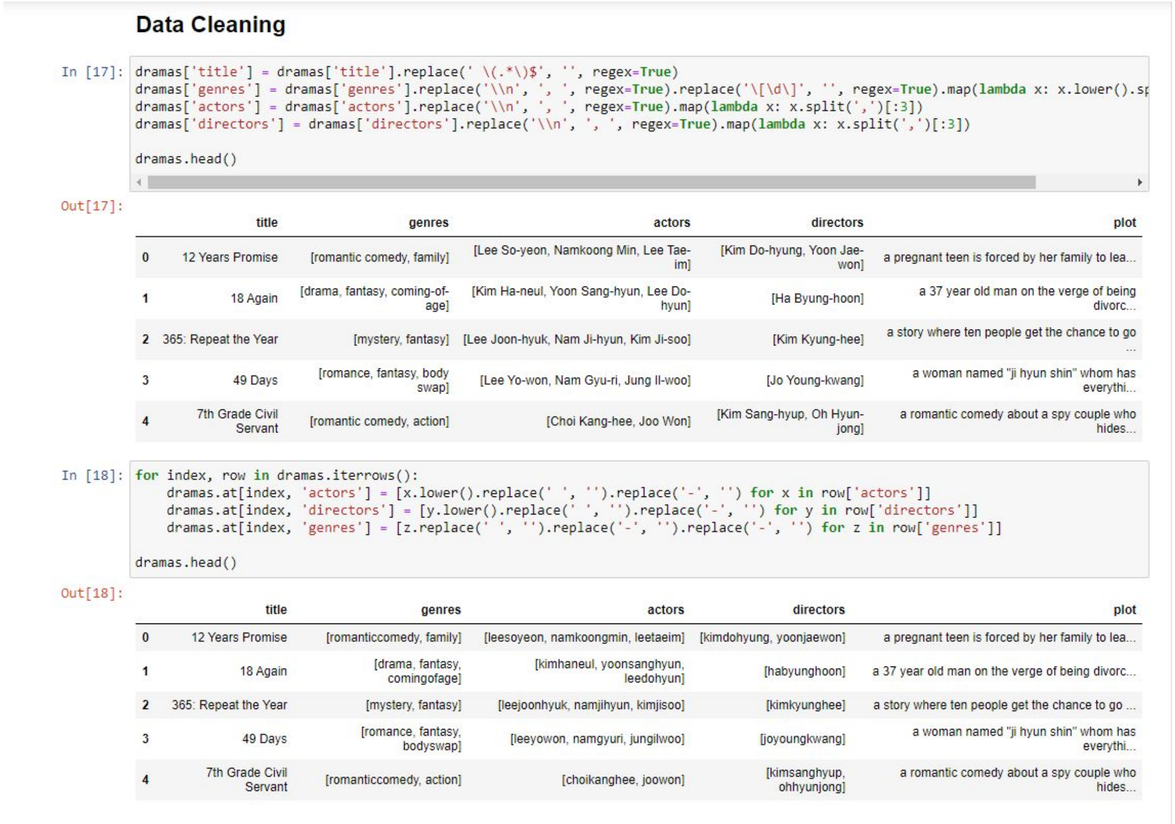
```
In [15]: dramas = df[(df['plot'] != "No Plot Listed") & (df['plot'] != "Add a Plot »") &
                     (df['actors'] != "No Actor Listed") & (df['directors'] != "No Director Listed") &
                     (df['genres'] != "No Genre Listed")].copy()

         dramas['plot'] = dramas['plot'].str.lower().str.replace('-', ' ')

         dramas.reset_index(drop=True, inplace=True)
         dramas.head()
```

Out[15]:

	title	genres	actors	directors	plot
0	12 Years Promise	Romantic comedy\nFamily	Lee So-yeon\nNamkoong Min\nLee Tae-im\nYoon So...	Kim Do-hyung\nYoon Jae-won	a pregnant teen is forced by her family to lea...
1	18 Again	Drama\nFantasy\nComing-of-age	Kim Ha-neul\nYoon Sang-hyun\nLee Do-hyun	Ha Byung-hoon	a 37 year old man on the verge of being divorc...
2	365: Repeat the Year	Mystery\nFantasy	Lee Joon-hyuk\nNam Ji-hyun\nKim Ji-soo\nYang D...	Kim Kyung-hee	a story where ten people get the chance to go ...
3	49 Days	Romance\nFantasy\nBody swap	Lee Yo-won\nNam Gyu-ri\nJung Il-woo\nJo Hyun-j...	Jo Young-kwang	a woman named "ji hyun shin" whom has everythi...
4	7th Grade Civil Servant	Romantic comedy\nAction	Choi Kang-hee\nJoo Won	Kim Sang-hyup\nOh Hyun-jong	a romantic comedy about a spy couple who hides...

```
In [16]: dramas.shape

Out[16]: (687, 5)
```

Data Cleaning



Data Cleaning

In [17]:
```python
dramas['title'] = dramas['title'].replace(' \(.*\)$', '', regex=True)
dramas['genres'] = dramas['genres'].replace('\\n', ', ', regex=True).replace('\[\d\]', '', regex=True).map(lambda x: x.lower().sp
dramas['actors'] = dramas['actors'].replace('\\n', ', ', regex=True).map(lambda x: x.split(',')[:3])
dramas['directors'] = dramas['directors'].replace('\\n', ', ', regex=True).map(lambda x: x.split(',')[:3])

dramas.head()
```

Out[17]:

	title	genres	actors	directors	plot
0	12 Years Promise	[romantic comedy, family]	[Lee So-yeon, Namkoong Min, Lee Tae-im]	[Kim Do-hyung, Yoon Jae-won]	a pregnant teen is forced by her family to lea...
1	18 Again	[drama, fantasy, coming-of-age]	[Kim Ha-neul, Yoon Sang-hyun, Lee Do-hyun]	[Ha Byung-hoon]	a 37 year old man on the verge of being divorc...
2	365: Repeat the Year	[mystery, fantasy]	[Lee Joon-hyuk, Nam Ji-hyun, Kim Ji-soo]	[Kim Kyung-hee]	a story where ten people get the chance to go ...
3	49 Days	[romance, fantasy, body swap]	[Lee Yo-won, Nam Gyu-ri, Jung Il-woo]	[Jo Young-kwang]	a woman named "ji hyun shin" whom has everythi...
4	7th Grade Civil Servant	[romantic comedy, action]	[Choi Kang-hee, Joo Won]	[Kim Sang-hyup, Oh Hyun-jong]	a romantic comedy about a spy couple who hides...

In [18]:
```python
for index, row in dramas.iterrows():
    dramas.at[index, 'actors'] = [x.lower().replace(' ', '').replace('-', '') for x in row['actors']]
    dramas.at[index, 'directors'] = [y.lower().replace(' ', '').replace('-', '') for y in row['directors']]
    dramas.at[index, 'genres'] = [z.replace(' ', '').replace('-', '').replace('-', '') for z in row['genres']]

dramas.head()
```

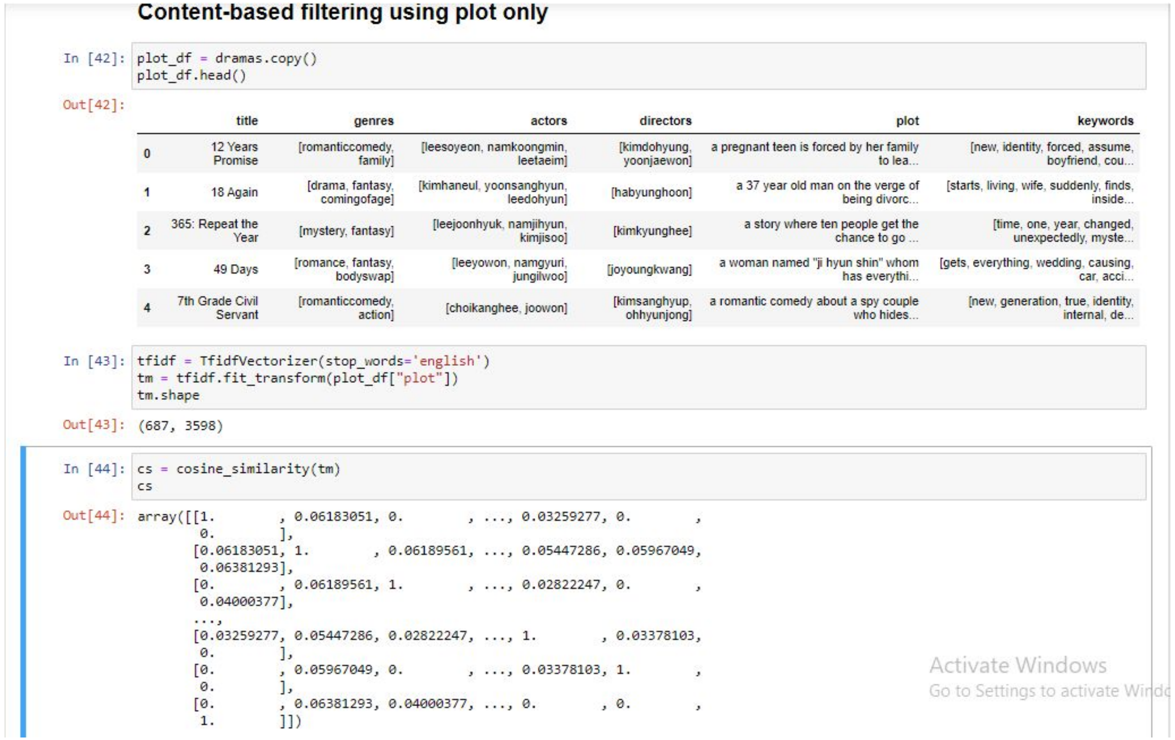
Out[18]:

	title	genres	actors	directors	plot
0	12 Years Promise	[romanticcomedy, family]	[leesoyeon, namkoongmin, leetaeim]	[kimdohyung, yoonjaewon]	a pregnant teen is forced by her family to lea...
1	18 Again	[drama, fantasy, comingofage]	[kimhaneul, yoonsanghyun, leedohyun]	[habyunghoon]	a 37 year old man on the verge of being divorc...
2	365: Repeat the Year	[mystery, fantasy]	[leejoonhyuk, namjihyun, kimjisoo]	[kimkyunghee]	a story where ten people get the chance to go ...
3	49 Days	[romance, fantasy, bodyswap]	[leeyowon, namgyuri, jungilwoo]	[joyoungkwang]	a woman named "ji hyun shin" whom has everythi...
4	7th Grade Civil Servant	[romanticcomedy, action]	[choikanghee, joowon]	[kimsanghyup, ohhyunjong]	a romantic comedy about a spy couple who hides...

SAAS

Building the system

Type 1: **Plot** Recommendation



Content-based filtering using plot only

```
In [42]: plot_df = dramas.copy()
         plot_df.head()
```

Out[42]:

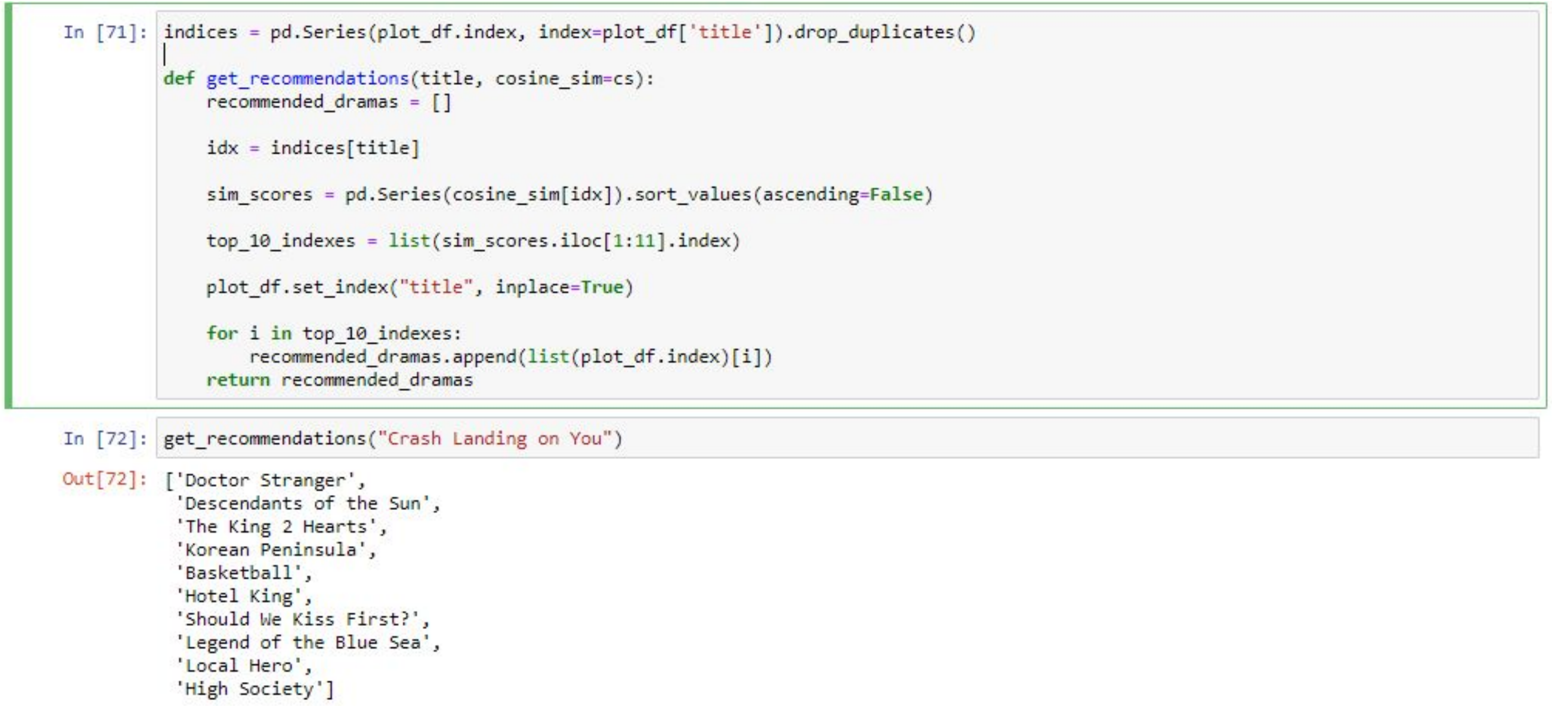
	title	genres	actors	directors	plot	keywords
0	12 Years Promise	[romanticcomedy, family]	[leesoyeon, namkoongmin, leetaeim]	[kimdohyung, yoonjaewon]	a pregnant teen is forced by her family to lea...	[new, identity, forced, assume, boyfriend, cou...
1	18 Again	[drama, fantasy, comingofage]	[kimhaneul, yoonsanghyun, leedohyun]	[habyunghoon]	a 37 year old man on the verge of being divorc...	[starts, living, wife, suddenly, finds, inside...
2	365: Repeat the Year	[mystery, fantasy]	[leejoonhyuk, namjihyun, kimjisoo]	[kimkyunghee]	a story where ten people get the chance to go ...	[time, one, year, changed, unexpectedly, myste...
3	49 Days	[romance, fantasy, bodyswap]	[leeyowon, namgyuri, jungilwoo]	[joyoungkwang]	a woman named "ji hyun shin" whom has everythi...	[gets, everything, wedding, causing, car, acci...
4	7th Grade Civil Servant	[romanticcomedy, action]	[choikanghee, joowon]	[kimsanghyup, ohhyunjong]	a romantic comedy about a spy couple who hides...	[new, generation, true, identity, internal, de...

```
In [43]: tfidf = TfidfVectorizer(stop_words='english')
         tm = tfidf.fit_transform(plot_df["plot"])
         tm.shape

Out[43]: (687, 3598)
```

```
In [44]: cs = cosine_similarity(tm)
         cs

Out[44]: array([[1.        , 0.06183051, 0.        , ..., 0.03259277, 0.        ,
                 0.        ],
                [0.06183051, 1.        , 0.06189561, ..., 0.05447286, 0.05967049,
                 0.06381293],
                [0.        , 0.06189561, 1.        , ..., 0.02822247, 0.        ,
                 0.04000377],
                ...,
                [0.03259277, 0.05447286, 0.02822247, ..., 1.        , 0.03378103,
                 0.        ],
                [0.        , 0.05967049, 0.        , ..., 0.03378103, 1.        ,
                 0.        ],
                [0.        , 0.06381293, 0.04000377, ..., 0.        , 0.        ,
                 1.        ]])
```

Building the system

Type 1: **Plot** Recommendation



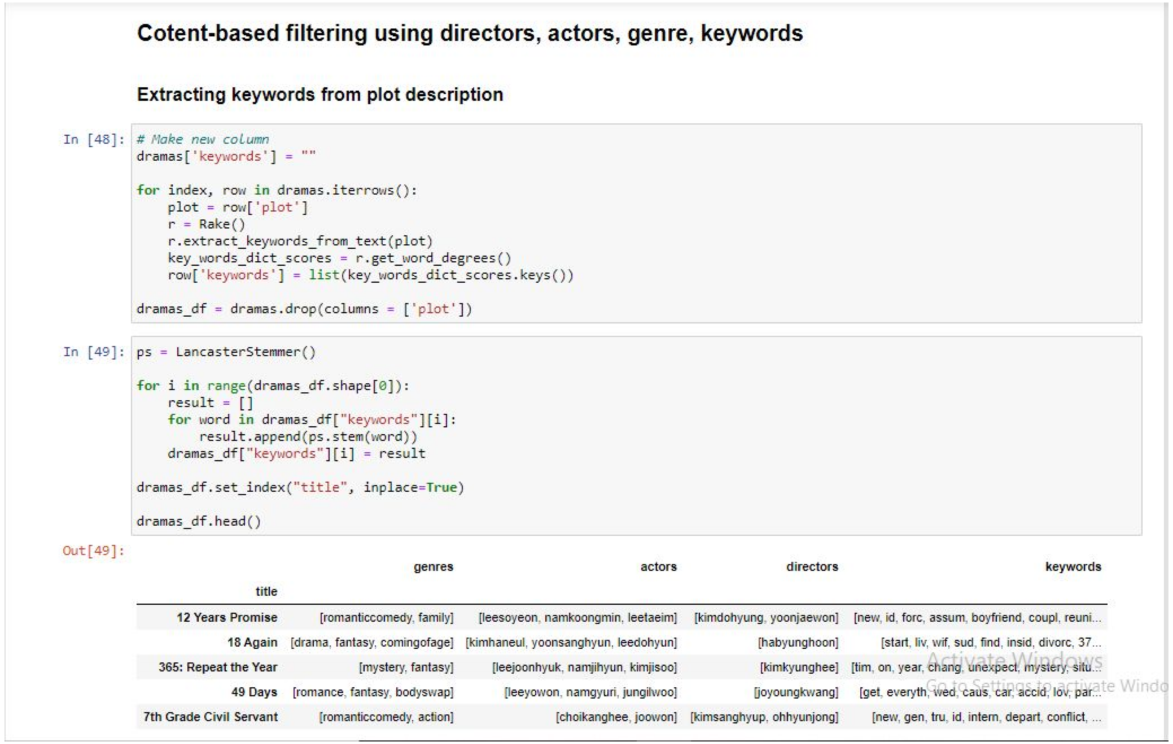
```
In [71]: indices = pd.Series(plot_df.index, index=plot_df['title']).drop_duplicates()

         def get_recommendations(title, cosine_sim=cs):
             recommended_dramas = []

             idx = indices[title]

             sim_scores = pd.Series(cosine_sim[idx]).sort_values(ascending=False)

             top_10_indexes = list(sim_scores.iloc[1:11].index)

             plot_df.set_index("title", inplace=True)

             for i in top_10_indexes:
                 recommended_dramas.append(list(plot_df.index)[i])
             return recommended_dramas
```

```
In [72]: get_recommendations("Crash Landing on You")
```

```
Out[72]: ['Doctor Stranger',
          'Descendants of the Sun',
          'The King 2 Hearts',
          'Korean Peninsula',
          'Basketball',
          'Hotel King',
          'Should We Kiss First?',
          'Legend of the Blue Sea',
          'Local Hero',
          'High Society']
```

Building the system

Type 2: **Genre, Actor, Director, Keyword** Recommendation



Cotent-based filtering using directors, actors, genre, keywords

Extracting keywords from plot description

In [48]:
```python
# Make new column
dramas['keywords'] = ""

for index, row in dramas.iterrows():
    plot = row['plot']
    r = Rake()
    r.extract_keywords_from_text(plot)
    key_words_dict_scores = r.get_word_degrees()
    row['keywords'] = list(key_words_dict_scores.keys())

dramas_df = dramas.drop(columns = ['plot'])
```

In [49]:
```python
ps = LancasterStemmer()

for i in range(dramas_df.shape[0]):
    result = []
    for word in dramas_df["keywords"][i]:
        result.append(ps.stem(word))
    dramas_df["keywords"][i] = result

dramas_df.set_index("title", inplace=True)

dramas_df.head()
```

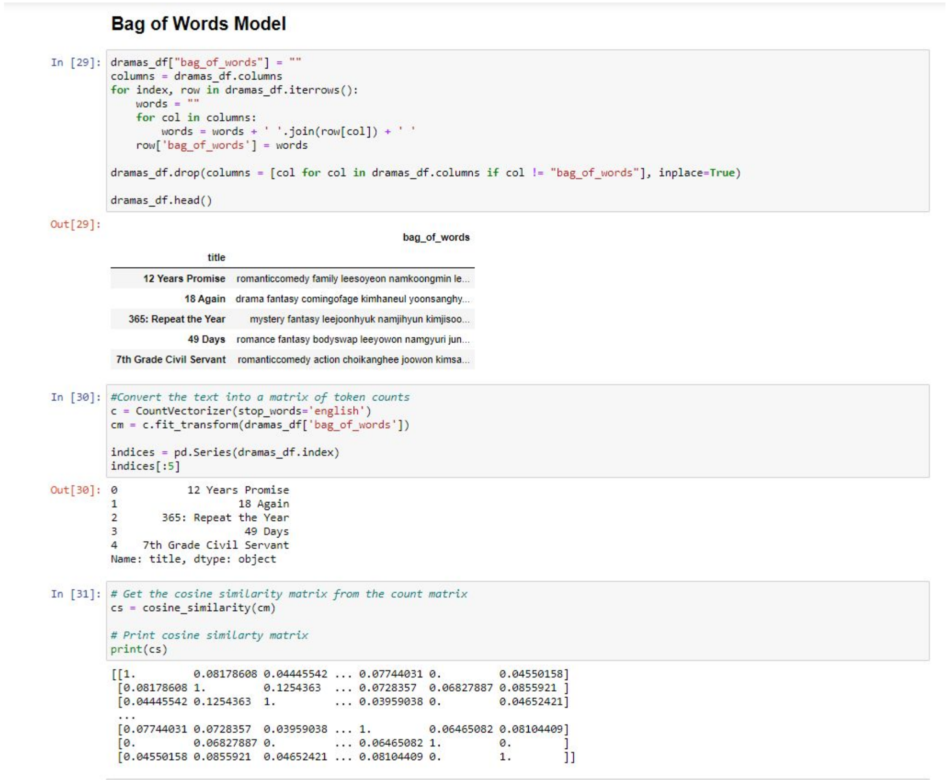
Out[49]:

	genres	actors	directors	keywords
title				
12 Years Promise	[romanticcomedy, family]	[leesoyeon, namkoongmin, leetaeim]	[kimdohyung, yoonjaewon]	[new, id, forc, assum, boyfriend, coupl, reuni...
18 Again	[drama, fantasy, comingofage]	[kimhaneul, yoonsanghyun, leedohyun]	[habyunghoon]	[start, liv, wif, sud, find, insid, divorc, 37...
365: Repeat the Year	[mystery, fantasy]	[leejoonhyuk, namjihyun, kimjisoo]	[kimkyunghee]	[tim, on, year, chang, unexpect, mystery, situ...
49 Days	[romance, fantasy, bodyswap]	[leeyowon, namgyuri, jungilwoo]	[joyoungkwang]	[get, everyth, wed, caus, car, accid, lov, par...
7th Grade Civil Servant	[romanticcomedy, action]	[choikanghee, joowon]	[kimsanghyup, ohhyunjong]	[new, gen, tru, id, intern, depart, conflict, ...

SAAS

Building the system

Type 2: **Genre, Actor, Director, Keyword** Recommendation



Bag of Words Model

```
In [29]: dramas_df["bag_of_words"] = ""
         columns = dramas_df.columns
         for index, row in dramas_df.iterrows():
             words = ""
             for col in columns:
                 words = words + ' '.join(row[col]) + ' '
             row['bag_of_words'] = words

         dramas_df.drop(columns = [col for col in dramas_df.columns if col != "bag_of_words"], inplace=True)

         dramas_df.head()
```

Out[29]:

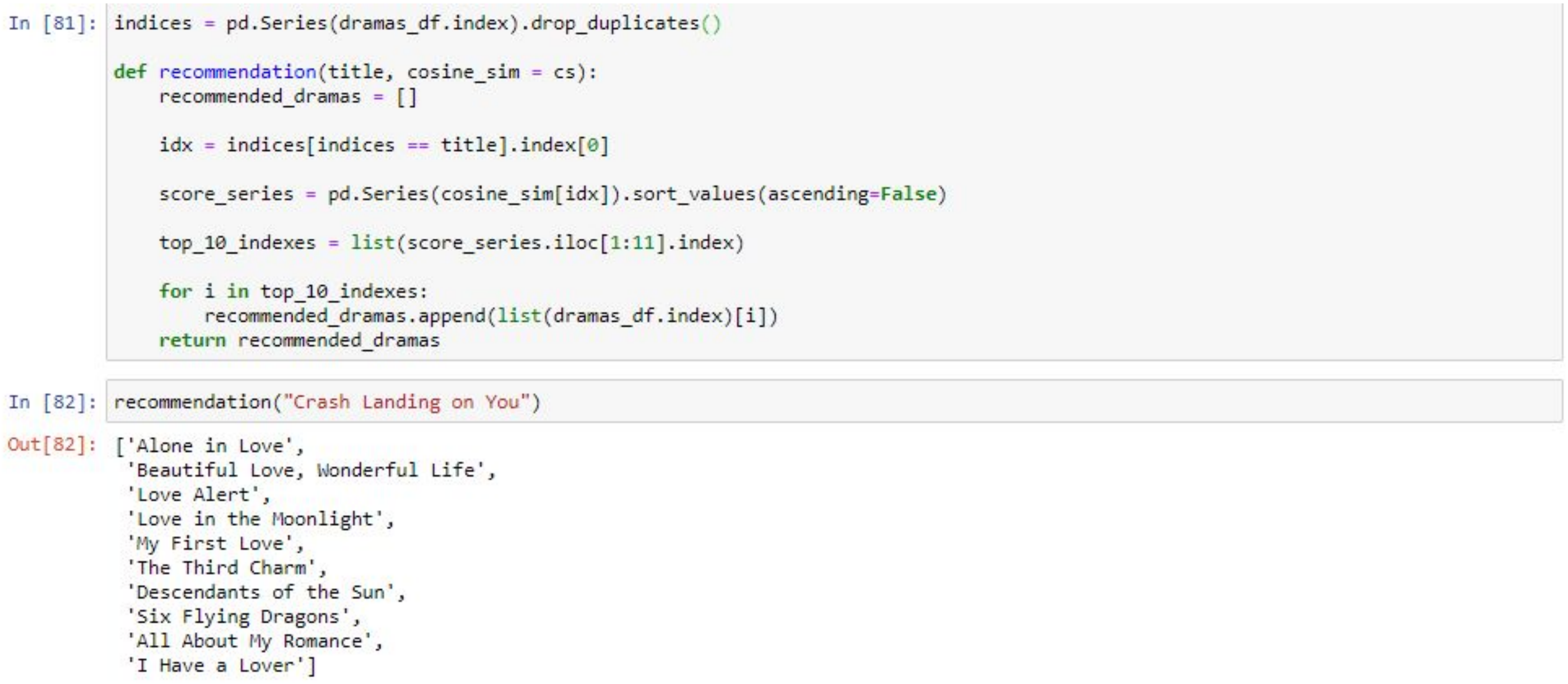
title	bag_of_words
12 Years Promise	romanticcomedy family leesoyeon namkoongmin le...
18 Again	drama fantasy comingofage kimhaneul yoonsanghy...
365: Repeat the Year	mystery fantasy leejoonhyuk namjihyun kimjisoo...
49 Days	romance fantasy bodyswap leeyowon namgyuri jun...
7th Grade Civil Servant	romanticcomedy action choikanghee joowon kimsa...

```
In [30]: #Convert the text into a matrix of token counts
         c = CountVectorizer(stop_words='english')
         cm = c.fit_transform(dramas_df['bag_of_words'])

         indices = pd.Series(dramas_df.index)
         indices[:5]

Out[30]: 0           12 Years Promise
         1                   18 Again
         2       365: Repeat the Year
         3                    49 Days
         4    7th Grade Civil Servant
         Name: title, dtype: object
```

```
In [31]: # Get the cosine similarity matrix from the count matrix
         cs = cosine_similarity(cm)

         # Print cosine similarty matrix
         print(cs)

         [[1.         0.08178608 0.04445542 ... 0.07744031 0.         0.04550158]
          [0.08178608 1.         0.1254363  ... 0.0728357  0.06827887 0.0855921 ]
          [0.04445542 0.1254363  1.         ... 0.03959038 0.         0.04652421]
          ...
          [0.07744031 0.0728357  0.03959038 ... 1.         0.06465082 0.08104409]
          [0.         0.06827887 0.         ... 0.06465082 1.         0.        ]
          [0.04550158 0.0855921  0.04652421 ... 0.08104409 0.         1.        ]]
```

Building the system

Type 2: **Genre, Actor, Director, Keyword** Recommendation



```
In [81]: indices = pd.Series(dramas_df.index).drop_duplicates()

         def recommendation(title, cosine_sim = cs):
             recommended_dramas = []

             idx = indices[indices == title].index[0]

             score_series = pd.Series(cosine_sim[idx]).sort_values(ascending=False)

             top_10_indexes = list(score_series.iloc[1:11].index)

             for i in top_10_indexes:
                 recommended_dramas.append(list(dramas_df.index)[i])
             return recommended_dramas

In [82]: recommendation("Crash Landing on You")

Out[82]: ['Alone in Love',
          'Beautiful Love, Wonderful Life',
          'Love Alert',
          'Love in the Moonlight',
          'My First Love',
          'The Third Charm',
          'Descendants of the Sun',
          'Six Flying Dragons',
          'All About My Romance',
          'I Have a Lover']
```

Next Steps

Questions?

- Finish display results of the system on Tableau using tabPy
- Look for better sources of data
- Implement other methods