دانشگاه علم و صنعت ایران

دانشکده مهندسی کامپیوتر

عنوان: پروژه سوم درس داده‌کاوی

خوشه‌بندی مقالات علمی بااستفاده از مجموعه داده **covid-19**

نام و نام خانوادگی: آیلین نائب‌زاده

شماره دانشجویی: ۹۹۵۲۲۱۸۵

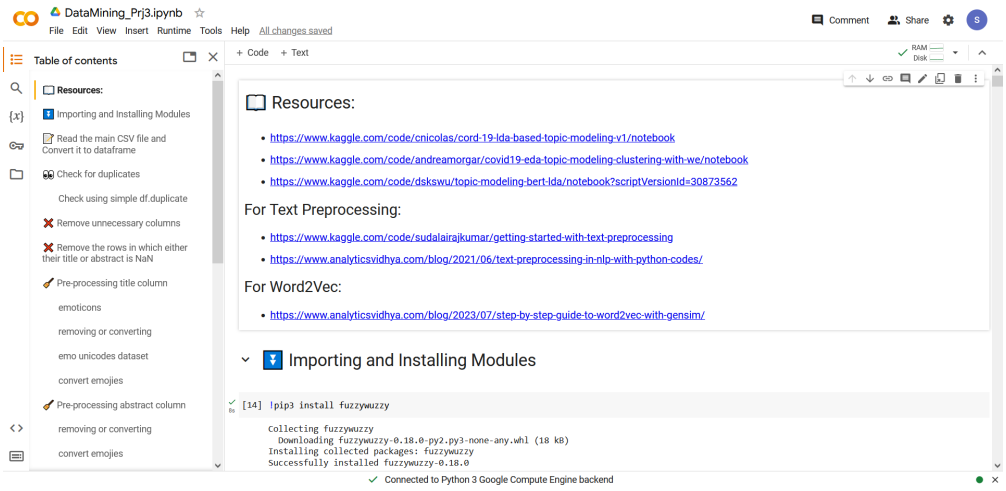
نیم‌سال تحصیلی: پائیز ۱۴۰۲

مدرّس: دکتر حسین رحمانی

فهرست مطالب

۱ گام اول

در طی این پروژه، هدف تحلیل مقالات مرتبط با بیماری covid-19 می‌باشد. فایل ورودی شامل ۱۴ ستون می‌باشد. در اولین قدم با مراجعه به سایت Google Colab و ساخت یک پروژه جدید و فراخوانی بعضی از کتابخانه‌های معروف زبان Python شروع به مشاهده داده‌های موجود در هر فایل و انجام تحلیل‌های ابتدایی می‌کنیم. محیط کلی پروژه را در تصویر زیر می‌توانید مشاهده کنید.



شکل ۱: نمای کلی از پروژه notebook در فضای Google Colab

در مرحله اول بااستفاده از کتابخانه pandas و تابع read_csv، فایل ورودی را به آبجکتی از نوع dataframe تبدیل می‌کنیم تا در قدم‌های بعدی راحت‌تر بتوانیم تحلیل‌های مورد نیاز را انجام بدهیم. همچنین بااستفاده از توابع info()، head() و describe() اطلاعات بیشتری نسبت به داده‌های موجود کسب می‌کنیم.

شکل ۲: خواندن فایل ورودی

همچنین در بخش زیر می‌توانید خروجی مربوط به تابع ()info. را مشاهده کنید. همانطور که مشاهده می‌کنید، در هر ستون تعدادی مقادیر تهی یا به اصطلاح NaN موجود است، که در گام بعدی مورد پردازش قرار می‌گیرند.

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 29500 entries, 0 to 29499
Data columns (total 14 columns):
 #   Column                        Non-Null Count  Dtype
---  ------                        --------------  -----
 0   sha                           17420 non-null  object
 1   source_x                      29500 non-null  object
 2   title                         29491 non-null  object
 3   doi                           26357 non-null  object
 4   pmcid                         27337 non-null  object
 5   pubmed_id                     16730 non-null  float64
 6   license                       17692 non-null  object
 7   abstract                      26909 non-null  object
 8   publish_time                  18604 non-null  object
 9   authors                       28903 non-null  object
 10  journal                       17791 non-null  object
 11  Microsoft Academic Paper ID  1134 non-null   float64
 12  WHO #Covidence                1236 non-null   object
 13  has_full_text                 17420 non-null  object
dtypes: float64(2), object(12)
memory usage: 3.2+ MB
None
```

۲ گام دوم

در این مرحله که مربوط به پیش‌پردازش داده‌ها می‌باشد، باتوجه به این موضوع که داده‌هایی که در اختیار داریم از نوع متنی می‌باشند، نیاز است بعضی از پیش‌پردازش‌های معمول را اجرا کنیم. بطور خلاصه مراحل زیر برروی داده‌های انجام شده‌اند:

- حذف ردیف‌های تکراری: براساس جفت عنوان و خلاصه مقاله، داده‌های تکراری شناخته و پیدا شده‌اند و تنها آخرین نمونه تکراری از هر گروه برای ادامه مراحل باقی می‌ماند.
- حذف تمامی ستون‌ها به‌جز ستون‌های title و abstract
- حذف تمامی ردیف‌هایی که حداقل یکی از مقادیر title یا abstract در آن‌ها تهی می‌باشد.
- تبدیل محتوای ستون title و abstract به string
- تبدیل تمامی حروف موجود در ستون‌های title و abstract به حروف کوچک
- حذف تمامی علائم نگارشی از ستون‌های title و abstract
- حذف برخی از حروف پرتکرار مانند for، must و ... از ستون‌های title و abstract
- برگرداندن کلمات موجود در ستون‌های title و abstract به ریشه اصلی آن‌ها با استفاده از عملیات stemming و lemmatization
- حذف emoticons از محتوای ستون‌های title و abstract
- حذف آدرس‌ها از محتوای ستون‌های title و abstract
- حذف تگ‌های html از محتوای ستون‌های title و abstract
- حذف برخی از رشته‌های خلاصه‌شده خاص از محتوای ستون‌های title و abstract

* خروجی تمامی مراحل بالا در دو ستون جدا به نام‌های cleaned_title و cleaned_abstract و در آبجکت df_dropped ذخیره می‌شوند.

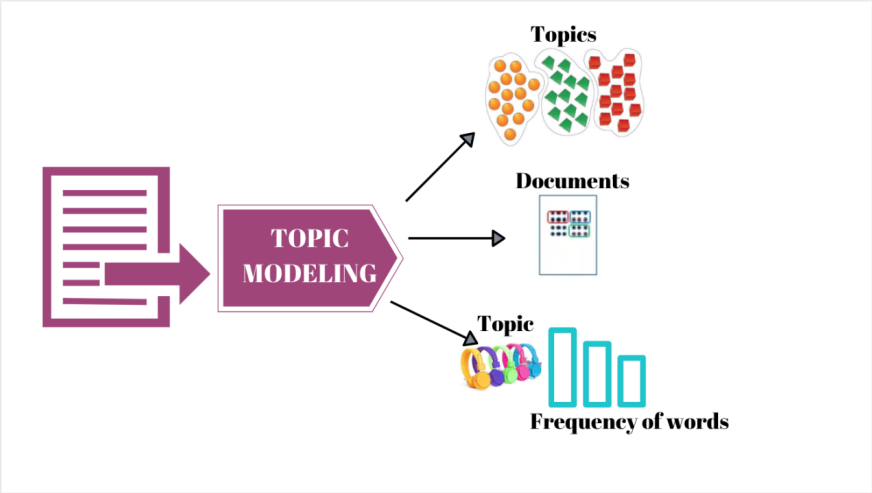
۳ گام سوم

این مرحله مربوط به استخراج ویژگی از داده‌های موجود می‌باشد. در این مرحله باتوجه به داده‌های متنی که در اختیار داریم، نیاز است تا برای پردازش راحت‌تر، کلمات را به مقادیر عددی تبدیل کنیم که برای انجام این کار از دو روش TF-IDF و Bag of Words استفاده شده‌است. همچنین باتوجه به این موضوع که از ما خواسته‌شده است که تحلیل‌ها را بر روی دو ستون title و abstract انجام دهیم، درواقع مقادیر تجمیع شده این دو ستون را به‌عنوان ورودی به دو تابع TfidfVectorizer و CountVectorizer به‌عنوان ورودی می‌دهیم. این دو تابع در کتابخانه sklearn.feature_extraction.text موجود می‌باشند. و نحوه استفاده از این دو تابع را در کدهای زیر می‌توانید مشاهده کنید:

```
1  tfidf = TfidfVectorizer()
2
3  tf_idf_matrix = tfidf.fit_transform(df_dropped['cleaned_title'] + ' ' +
       df_dropped['cleaned_abstract'])
4
5  """
6  tf_idf_matrix
7
8  <22819x99910 sparse matrix of type '<class 'numpy.float64'>'
9          with 2082915 stored elements in Compressed Sparse Row format>
10 """
11
12 bow_vectorizer = CountVectorizer(max_df = 0.90, min_df = 2, max_features =
       1000, stop_words='english')
13 bow_matrix = bow_vectorizer.fit_transform(df_dropped['cleaned_title'] + ' ' +
       df_dropped['cleaned_abstract'])
14
15 """
16 bow_matrix
17
18 <22819x1000 sparse matrix of type '<class 'numpy.int64'>'
19         with 1286156 stored elements in Compressed Sparse Row format>
20 """
```

۴ گام چهارم

در این مرحله به‌منظور انجام Topic Modeling از الگوریتم Latent Dirichlet Allocation استفاده می‌کنیم. این الگوریتم را به‌طور مجزا یک بار با استفاده از ماتریس خروجی از الگوریتم TF-IDF انجام می‌دهیم و بار دیگر نیز الگوریتم را برروی خروجی Bag of Words اجرا می‌کنیم.



شکل ۳: الگوریتم LDA

کدهای مربوط به این بخش را در بخش زیر می‌توانید مشاهده کنید:

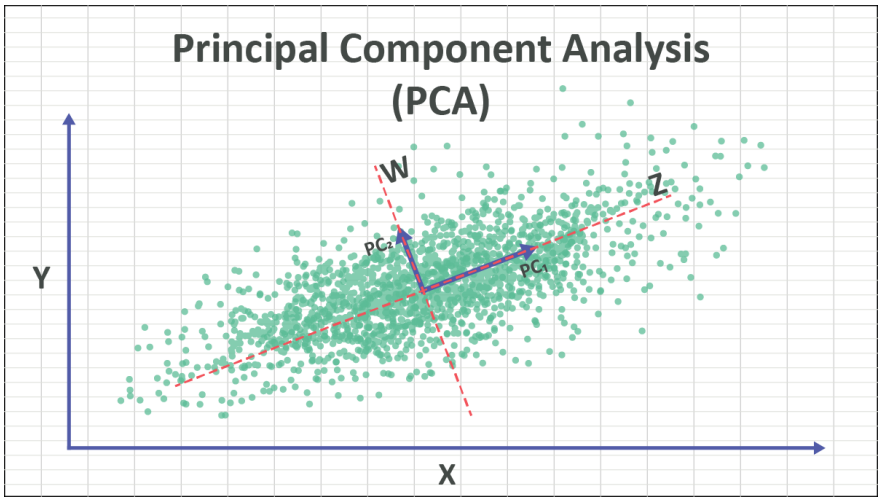
```
# Create an LDA object with 20 topics
lda_tf_idf = LatentDirichletAllocation(n_components=20)
lda_bow = LatentDirichletAllocation(n_components=20)

tf_idf_topics = lda_tf_idf.fit_transform(tf_idf_matrix)
print(tf_idf_topics.shape) # Output will be (22819, 20)

bow_topics = lda_bow.fit_transform(bow_matrix)
print(bow_topics.shape) # Output will be(22819, 20)
```

۵ گام پنجم

در این مرحله از ما خواسته‌شده است تا ابعاد ماتریس‌هایی که در مرحله قبلی محاسبه شده بودند را کاهش دهیم. به‌منظور انجام این کار از الگوریتم Principle Component Analysis استفاده می‌کنیم. در هنگام استفاده از این الگوریتم همانگونه که در تصویر زیر مشاهده می‌کنیم، تعداد بخش‌های اصلی را با ۲ مقداردهی می‌کنیم. همانند مرحله پیشین نیاز است یک بار الگوریتم را با استفاده از ماتریس خروجی TF-IDF و بار دیگر با استفاده از ماتریس خروجی Bag of Words اجرا نمائیم.

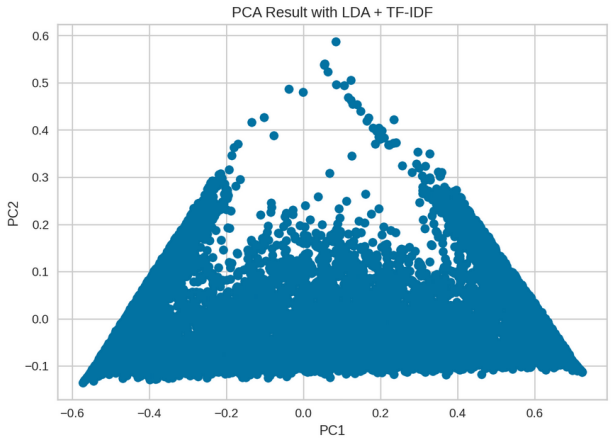


شکل ۴: الگوریتم PCA

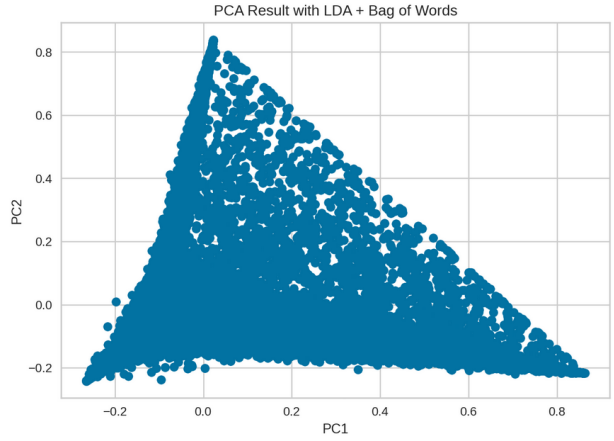
کدهای مربوط به این بخش را در بخش زیر می‌توانید مشاهده کنید:

```
1 # Create a PCA object with 2 components
2 pca_tf_idf = PCA(n_components=2)
3 pca_bow = PCA(n_components=2)
4
5 tf_idf_reduced = pca_tf_idf.fit_transform(tf_idf_topics)
6 print(tf_idf_reduced.shape) # Output will be (22819, 2)
7
8 bow_reduced = pca_bow.fit_transform(bow_topics)
9 print(bow_reduced.shape) # Output will be (22819, 2)
```

همچنین تصاویر مربوط به خروجی این الگوریتم را برروی دو ماتریس TF-IDF و Bag of Words می‌توانید مشاهده کنید:



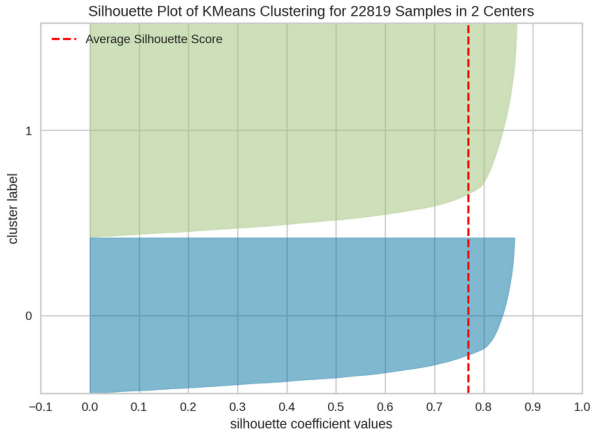
شکل ۵: خروجی الگوریتم PCA برای ماتریس ورودی TF-IDF



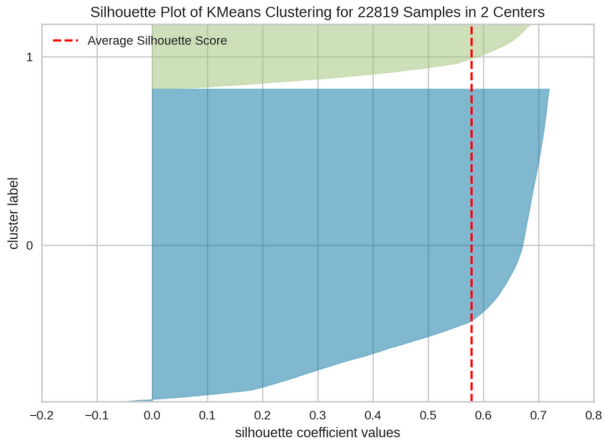
شکل ۶: خروجی الگوریتم PCA برای ماتریس ورودی BoW

۶ گام ششم

در این مرحله که مربوط به استفاده از الگوریتم‌های خوشه‌بندی می‌باشد، همانطور که در تصویر زیر مشاهده می‌کنید از دو الگوریتم KMeans و DBSCAN استفاده می‌کنیم. البته هر یک از این دو الگوریتم را بصورت جداگانه بااستفاده از دو ماتریس TF-IDF و Bag of Words مورد آموزش قرار می‌دهیم. همچنین هنگام استفاده از الگوریتم KMeans بااستفاده از الگوریتم Elbow و Silhoute سعی می‌کنیم تا بهترین مقدار K یا به‌عبارتی بهینه‌ترین تعداد خوشه‌ها را انتخاب کنیم. مقدار بازه تعداد خوشه‌ها را از ۲ تا ۲۰ تعریف می‌کنیم و همانطور که در خروجی مشخص است، در هر دو حالت تعداد خوشه دو بهترین پاسخ را باتوجه به معیار Silhoute می‌دهد. چراکه همانطور که می‌دانیم هر چه مقدار این معیار به ۱ نزدیک‌تر باشد، یعنی خوشه‌بندی بهتر انجام شده‌است.

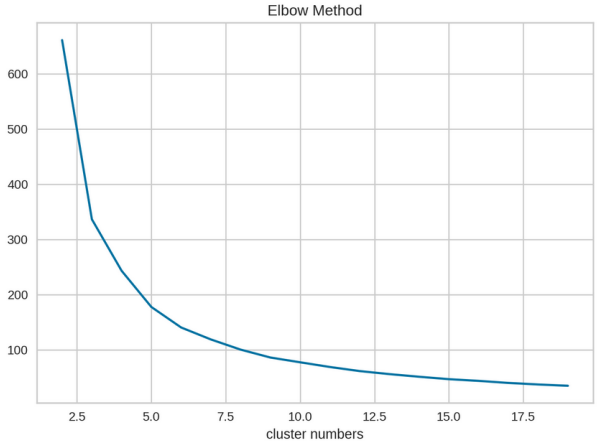


شکل ۷: خروجی تابع Silhoute برای الگوریتم KMeans در حالت استفاده از الگوریتم TF-IDF

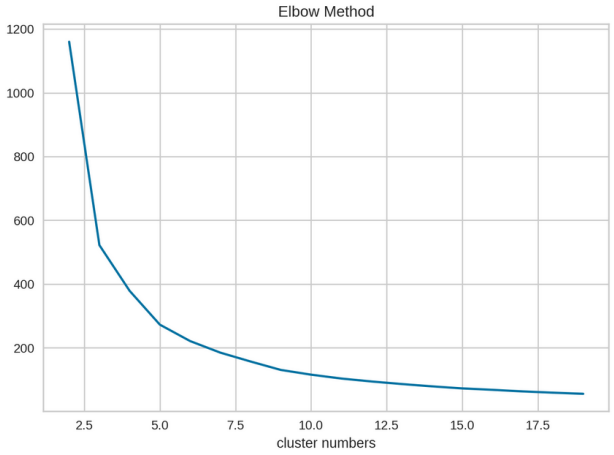


شکل ۸: خروجی تابع Silhoute برای الگوریتم KMeans در حالت استفاده از الگوریتم Bag of Words

نمودارهای Elbow مربوط به الگوریتم‌های KMeans را نیز می‌توانید در تصاویر زیر مشاهده کنید:



شکل ۹: Elbow for KMeans TF-IDF



شکل ۱۰: Elbow for KMeans BoW

*تمامی خروجی‌های مربوط به این بخش در خروجی کدها موجود است.

۷ نتیجه‌گیری

۱.۷ کدام یک از روش استخراج ویژگی و مدل خوشه‌بندی عملکرد بهتری دارند؟

در حالت استفاده از الگوریتم KMeans و ماتریس نتایج TF-IDF به‌همراه K=2 به بهترین عملکرد می‌رسیم.

۲.۷ دلیل خود برای این انتخاب را ذکر کنید.

سادگی پیاده‌سازی - قابلیت فهم بیشتر الگوریتم و نحوه عملکرد - کمبود زمان:)

۳.۷ با بررسی خوشه ها تحلیل کنید که هر خوشه نماینده ی کدام نوع از موضوعات می باشند.

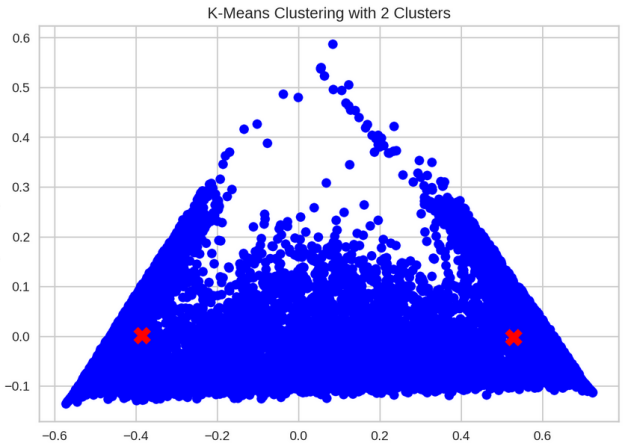
همانطور که در بخش زیر می‌توانید مشاهده کنید، ده عنوان پرتکرار در حالتی که تنها دو خوشه داشته باشیم، به‌صورت زیر می‌باشند:

```
1
2  Cluster 0:
3  Title + Abstract:
4  [('vte', 1.678757792866809), ('cistran', 1.6842943889715567), ('rp',
       1.7129443327427645), ('hbz', 1.9769338882057215), ('de',
       1.9923199119796313), ('ppias', 2.2181387904324343), ('nw',
       2.2671036443696733), ('stat3', 4.269071400018623), ('cypa',
       5.930938308329624), ('cyclophilin', 7.042773950588994)]
5  Cluster 1:
6  Title + Abstract:
7  [('csa', 3.3479223037548627), ('nendou', 3.463791236165143), ('omtas',
       3.5563027857379423), ('la', 3.58139521730169), ('rc', 3.7748316966902937),
        ('nsp14', 4.299709957546419), ('nsp16', 4.828983563894078), ('nsp11',
       5.2362523138395884), ('de', 5.585708033407686), ('cchfv',
       8.213499671505836)]
8  ..................................................................
9
10 Cluster 0:
11 Title + Abstract:
12 [('recent', 1470.153988758314), ('research', 1594.534216970629), ('global',
       1638.6642640546884), ('new', 1987.7613982795474), ('drug',
       2023.8265750613432), ('emerg', 2292.504527338654), ('review',
       2448.7748359447), ('health', 3010.6509597508707), ('develop',
       3332.1425793052204), ('diseas', 4115.477387357077)]
13 Cluster 1:
14 Title + Abstract:
15 [('anim', 1172.0006459933547), ('syndrom', 1454.5165365477444), ('virus',
       1591.727393376288), ('infect', 1689.7183916321692), ('respiratori',
       1699.9138172302232), ('east', 1796.7737427966777), ('middl',
       1877.3738146026203), ('bat', 3053.279585291326), ('human',
       3479.644308713773), ('merscov', 3559.049999997403)]
```

۴.۷ نمایش داده‌ها

تصویر مربوط به خروجی الگوریتم KMeans را با دو خوشه می‌توانید در تصویر زیر مشاهده کنید. باتوجه به اینکه پیش از استفاده از خوشه‌بندی، از الگوریتم کاهش ابعاد استفاده کردیم، دقت مدل متاسفانه کاهش یافته است و به این دلیل که داده‌ها را به دو بعد کاهش داده‌ایم، امکان دارد که بعد دوم در زیر این بعد قرار داشته باشد.

* سایر نمایش‌های مربوط به خوشه‌ها در خروجی برنامه قرار دارند.



شکل ۱۱: Clustering Visualize