



دانشگاه صنعتی امیر کبیر

(پلی تکنیک تهران)

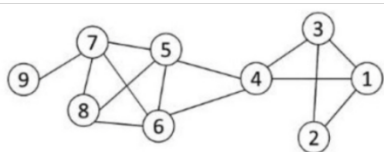
نام و نام خانوادگی: پیمان هاشمی

شماره دانشجویی: 400131032

درس: تحلیل شبکه های پیچیده

تمرین: تمرین شماره 2

سوال (1)



سوال 1

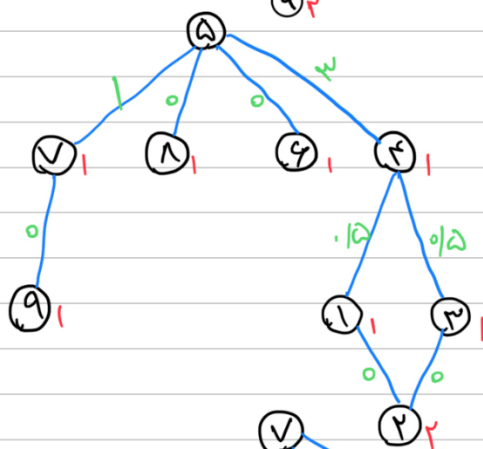
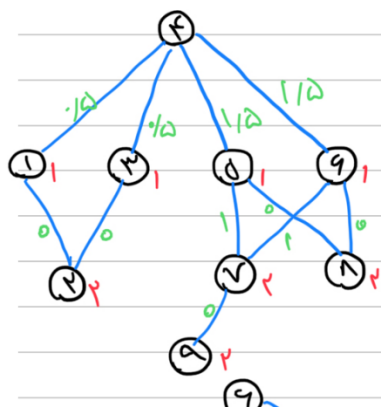
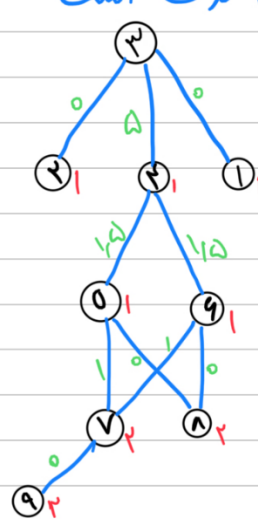
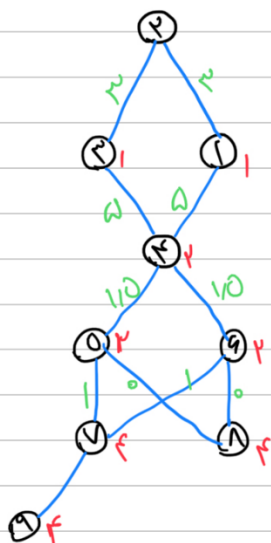
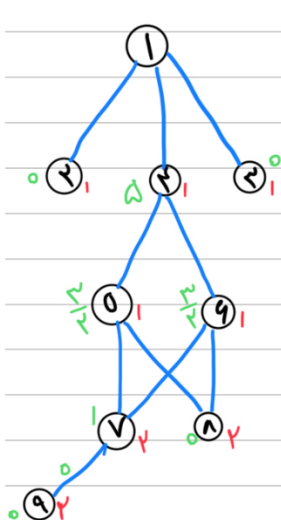
الف)

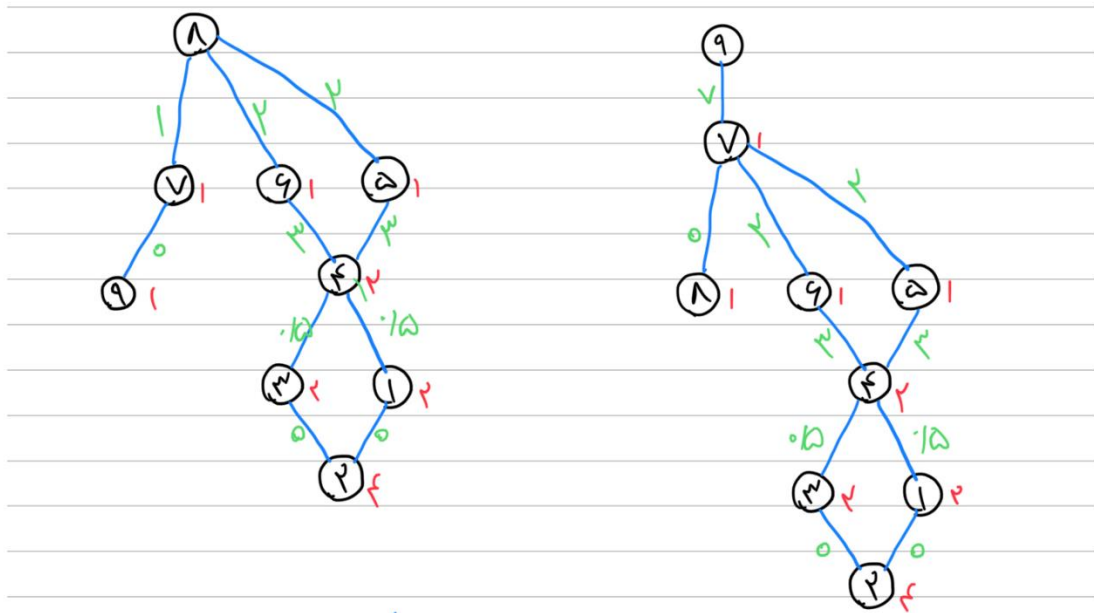
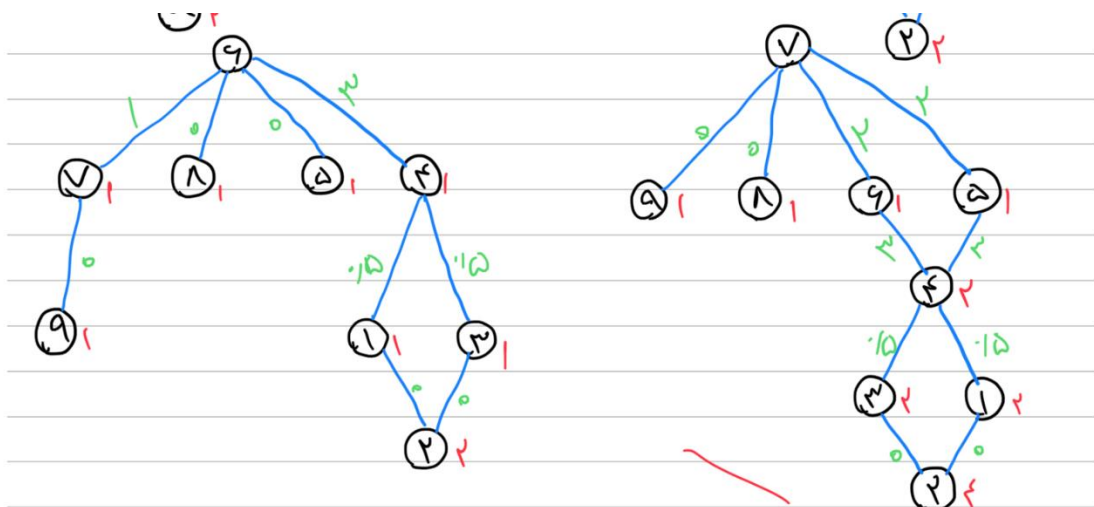
ابتدا باید 9 گرهان DAG با شروع از گره 1 از

9 فرد تشکیل دهیم، مقدار Betweenness Centrality را به هر گره از 9 گرهان

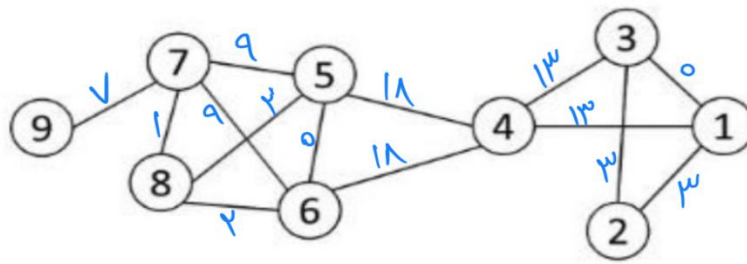
به دست آوریم. در نهایت میزان Centrality گره 1 را برابر با جمع Centrality هایش در

این 9 گرهان است





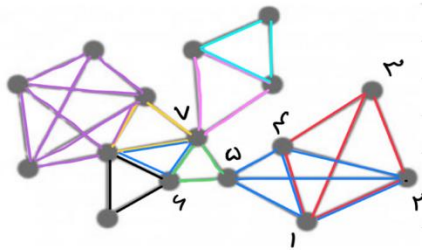
شکل مفاتی



برای مشخص کردن Community های گراف، حد آستانه‌ی ۵ را برای
 Betweenness Centrality یا حد نقطه‌ی گزینیم و حدیابی که Betweenness
 بزرگتر باشد را در نقطه‌ی گزینیم پس ۲ یا ۵، ۴۴ حدی می‌شوند و
 Community های زیر تشکیل می‌شوند.



(ب)



حداکثر Max Clique

زمانی که $K=4$ باشد.

رشته‌ها را محدود می‌کنیم و داریم

| ① | ② | ③ |
|----------------------|---|-------------------------------------|
| $Q = \{1\}$ | $\{1, 2\}$ | $\{1, 2, 3\}$ |
| $R = \{2, 3, 4, 5\}$ | $\{2, 3, 4, 5\} \cap \{2\} = \{2, 4, 5\}$ | $\{2, 3, 4, 5\} \cap \{3\} = \{4\}$ |

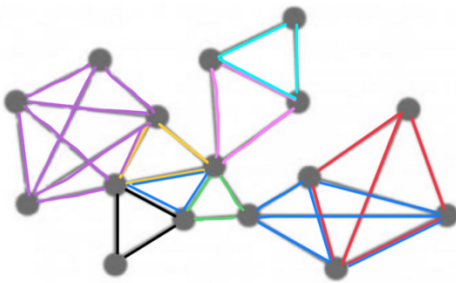
| ④ | ⑤ | ⑥ |
|----------------------|-------------------------------|----------------------------------|
| $Q = \{1, 2, 3, 4\}$ | back track $\{1, 2, 4\}$ | $\{1, 2, 4, 5\}$ |
| $R = \{1\}$ | $\{2, 5\} \cap \{4\} = \{5\}$ | $\{5\} \cap \{5\} = \emptyset$ * |

پس ۱، ۲، ۳، ۴، ۵، ۶، ۷، ۸، ۹ Max clique هستند.
 حال برای بقیه رشته‌ها هم همین روند را طی می‌کنیم و جدول زیر را تشکیل می‌دهیم و داریم

ب

در این روش جدول تعداد همسایگی ریزه‌ها را

محاسبه می‌کنیم



| | ۱ | ۲ | ۳ | ۴ | ۵ | ۶ | ۷ | ۸ | ۹ | ۱۰ |
|----|---|---|---|---|---|---|---|---|---|----|
| ۱ | ۰ | ۰ | ۰ | ۰ | ۰ | ۰ | ۰ | ۰ | ۰ | ۰ |
| ۲ | ۰ | ۰ | ۰ | ۰ | ۰ | ۰ | ۰ | ۰ | ۰ | ۰ |
| ۳ | ۰ | ۰ | ۰ | ۰ | ۰ | ۰ | ۰ | ۰ | ۰ | ۰ |
| ۴ | ۰ | ۰ | ۰ | ۰ | ۰ | ۰ | ۰ | ۰ | ۰ | ۰ |
| ۵ | ۰ | ۰ | ۰ | ۰ | ۰ | ۰ | ۰ | ۰ | ۰ | ۰ |
| ۶ | ۰ | ۰ | ۰ | ۰ | ۰ | ۰ | ۰ | ۰ | ۰ | ۰ |
| ۷ | ۰ | ۰ | ۰ | ۰ | ۰ | ۰ | ۰ | ۰ | ۰ | ۰ |
| ۸ | ۰ | ۰ | ۰ | ۰ | ۰ | ۰ | ۰ | ۰ | ۰ | ۰ |
| ۹ | ۰ | ۰ | ۰ | ۰ | ۰ | ۰ | ۰ | ۰ | ۰ | ۰ |
| ۱۰ | ۰ | ۰ | ۰ | ۰ | ۰ | ۰ | ۰ | ۰ | ۰ | ۰ |

| | ۱ | ۲ | ۳ | ۴ | ۵ | ۶ | ۷ | ۸ | ۹ | ۱۰ |
|----|---|---|---|---|---|---|---|---|---|----|
| ۱ | ۰ | ۰ | ۰ | ۰ | ۰ | ۰ | ۰ | ۰ | ۰ | ۰ |
| ۲ | ۰ | ۰ | ۰ | ۰ | ۰ | ۰ | ۰ | ۰ | ۰ | ۰ |
| ۳ | ۰ | ۰ | ۰ | ۰ | ۰ | ۰ | ۰ | ۰ | ۰ | ۰ |
| ۴ | ۰ | ۰ | ۰ | ۰ | ۰ | ۰ | ۰ | ۰ | ۰ | ۰ |
| ۵ | ۰ | ۰ | ۰ | ۰ | ۰ | ۰ | ۰ | ۰ | ۰ | ۰ |
| ۶ | ۰ | ۰ | ۰ | ۰ | ۰ | ۰ | ۰ | ۰ | ۰ | ۰ |
| ۷ | ۰ | ۰ | ۰ | ۰ | ۰ | ۰ | ۰ | ۰ | ۰ | ۰ |
| ۸ | ۰ | ۰ | ۰ | ۰ | ۰ | ۰ | ۰ | ۰ | ۰ | ۰ |
| ۹ | ۰ | ۰ | ۰ | ۰ | ۰ | ۰ | ۰ | ۰ | ۰ | ۰ |
| ۱۰ | ۰ | ۰ | ۰ | ۰ | ۰ | ۰ | ۰ | ۰ | ۰ | ۰ |

حال در جدولی که K را در نظر بگیریم

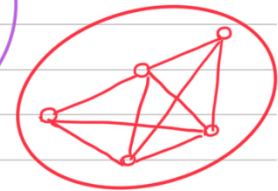
ببینیم در جدول در هر سطر اگر $K-1$ از a

باشد برابر ۳ در نظر گرفته می‌شود.

رصداری که ۱ سطر Community های

ما را تشکیل می‌دهند.

Communities



تعداد Community ها برابر ۳ است.

سوال 2)

(الف)

(تمام نتایج بدست آمده از این بخش در فولدر **Result** با فرمت اعلام شده ذخیره شده است)

در این قسمت در ابتدا دو مجموعه داده را خوانده و گراف آن ها تشکیل می‌دهیم:

```
Airport Dataset :  
Number of nodes: 1574  
Number of edges: 17215  
Average degree: 21.874205844980942  
  
Bible Dataset :  
Number of nodes: 1773  
Number of edges: 9131  
Average degree: 10.300056401579244
```

Closeness:

در یک گراف متصل، مرکزیت نزدیکی (یا نزدیکی) یک گره معیاری از مرکزیت در یک شبکه است که به عنوان مجموع طول کوتاه‌ترین مسیرها بین گره و سایر گره‌های گراف محاسبه می‌شود. بنابراین هر چه یک گره مرکزی تر باشد، به سایر گره ها نزدیک تر است.

$$C_i = \frac{n}{\sum_j d_{ij}}$$

(برای شلوغ نبودن گزارش تنها قسمت از نتایج مربوط به Airport آورده می‌شود – فایل اصلی شامل تمام نتایج برای دو مجموعه داده است که در قسمت **Result** ذخیره شده اند)

```
1 : 0.2509199652825258  
47 : 0.3349706539094009  
832 : 0.32558674889222533  
2 : 0.2512413999858501  
679 : 0.3356155172003495  
3 : 0.24397489393743332  
424 : 0.26949545567015354  
1688 : 0.32250823081431323  
4 : 0.29470370828543085  
685 : 0.41795485959287
```

Efficiency

Efficiency یک جفت گره در یک نمودار، معکوس ضربی کوتاه ترین فاصله مسیر بین گره ها است. بازده محلی یک گره در نمودار، میانگین بازده کلی زیرگراف القا شده توسط همسایگان گره است. متوسط بازده محلی میانگین بازده محلی هر گره است.

$$C_i = \frac{1}{n-1} \sum_{j(\neq i)} \frac{1}{d_{ij}}$$

نتایج بدست آمده برای Efficiency:

0 : 3.972681067344346
1 : 2.9758576874205844
2 : 3.0616264294790345
3 : 3.9675984752223634
4 : 2.9701397712833546
5 : 4.08576874205845
6 : 3.698856416772554
7 : 3.090851334180432
8 : 3.3824650571791612
9 : 2.3850063532401524
10 : 3.972681067344346

Degree

درجه نود ها بر کل نود ها می باشد. نتایج بدست آمده برای Degree:

1 : 0.0012706480304955528
47 : 0.025412960609911054
832 : 0.005717916137229987
2 : 0.0006353240152477764
679 : 0.017153748411689963
3 : 0.0012706480304955528
424 : 0.0038119440914866584
1688 : 0.007623888182973317
4 : 0.0006353240152477764
685 : 0.13024142312579415

Katz

katz برای اندازه گیری میزان نسبی تأثیر یک بازیگر (یا گره) در یک شبکه اجتماعی استفاده می شود. بر خلاف معیارهای مرکزی معمولی که فقط کوتاه ترین مسیر را بین یک جفت بازیگر در نظر می گیرند، مرکزیت کاتز با در نظر گرفتن تعداد کل پیاده روی بین یک جفت بازیگر تأثیر می گذارد. این شبیه به رتبه صفحه گوگل و مرکزیت بردار ویژه است. مرکزیت کاتز تأثیر نسبی یک گره در یک شبکه را با اندازه گیری تعداد همسایه های مستقیم (گره های درجه یک) و همچنین تمام گره های دیگر در شبکه که از طریق این همسایه های فوری به گره مورد نظر متصل می شوند، محاسبه می کند. با این حال، اتصالات ایجاد شده با همسایگان دور توسط یک عامل تضعیف alpha جریمه می شود. به هر مسیر یا اتصال بین یک جفت گره، وزنی که توسط alpha تعیین می شود و فاصله بین گره ها به عنوان alpha_d تعیین می شود. نتایج بدست آمده برای katz:

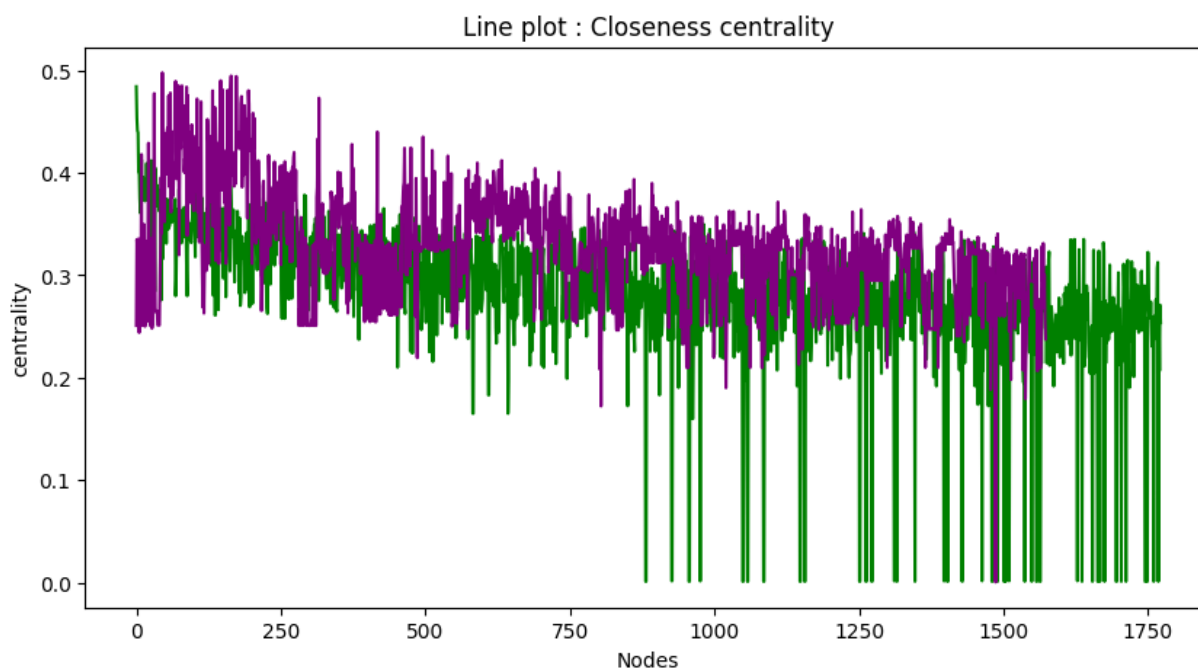
1 : 180.10000000000005
47 : 4248.400000000002
832 : 2393.3000000000025
2 : 122.70000000000002
679 : 4474.0999999999985
3 : 139.6
424 : 1023.8999999999995
1688 : 11929.300000000001
4 : 2914.4999999999995
685 : 330147.89999999954

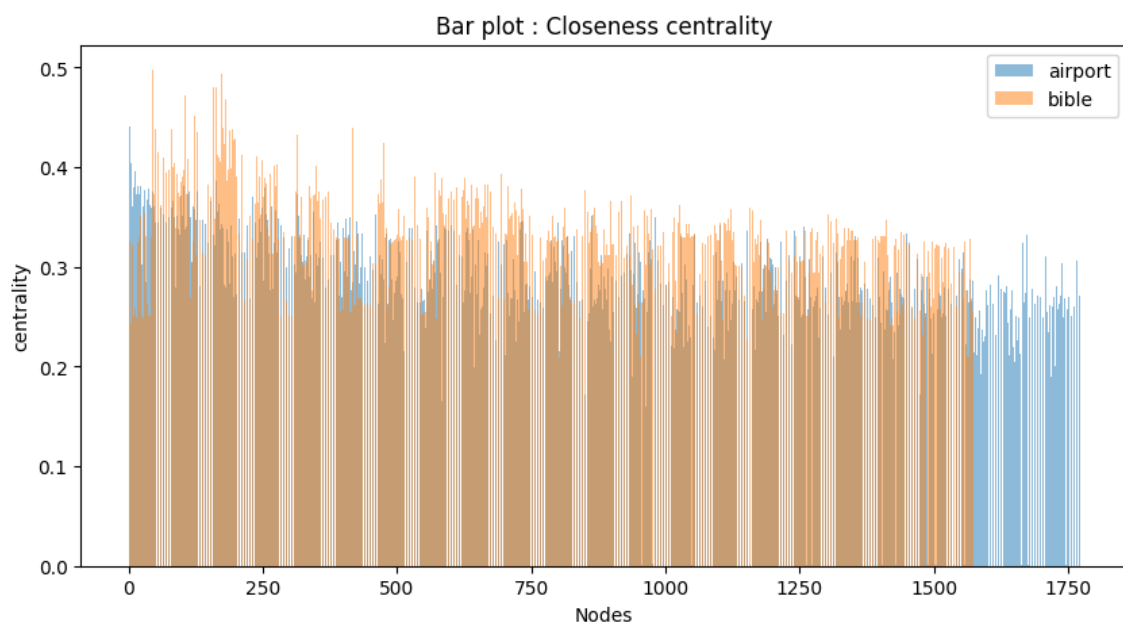
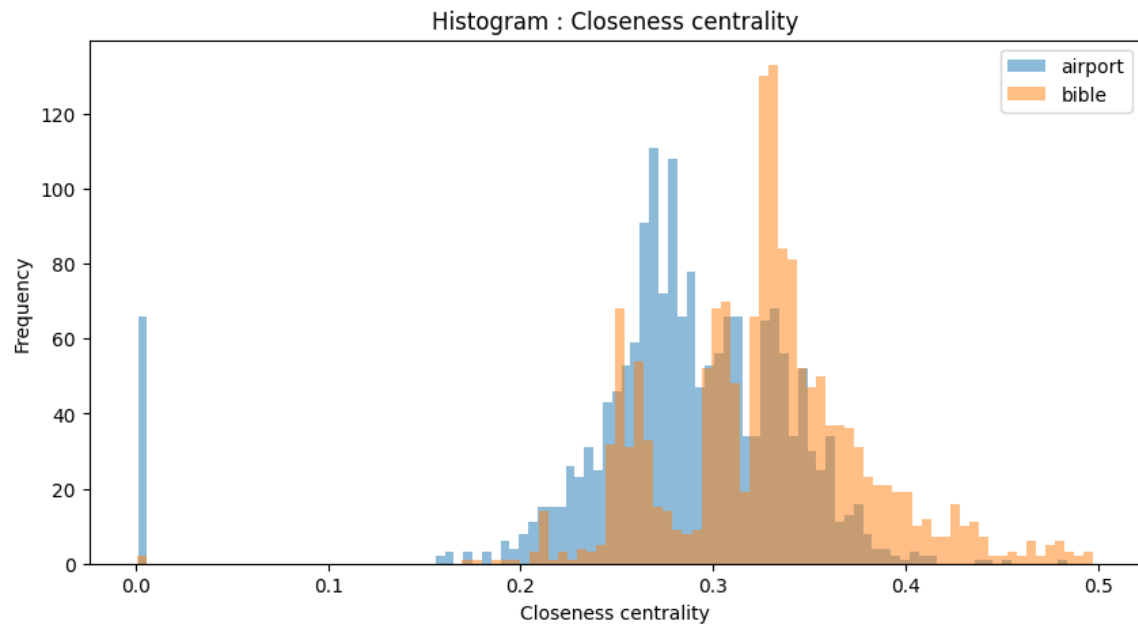
(ب)

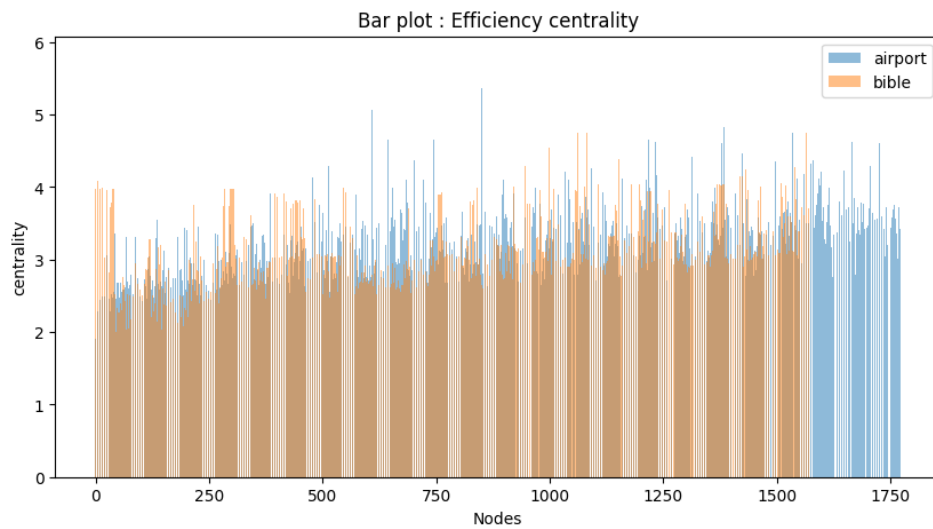
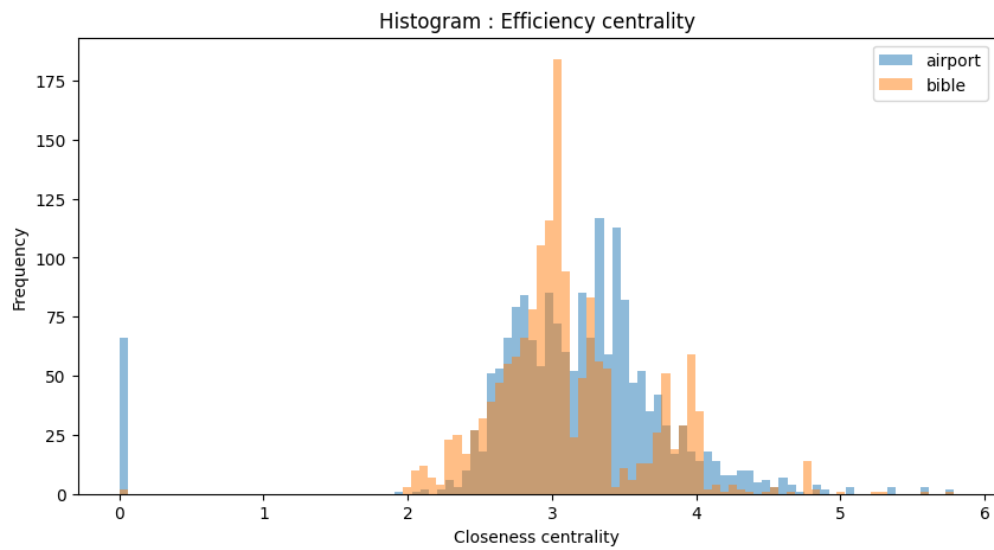
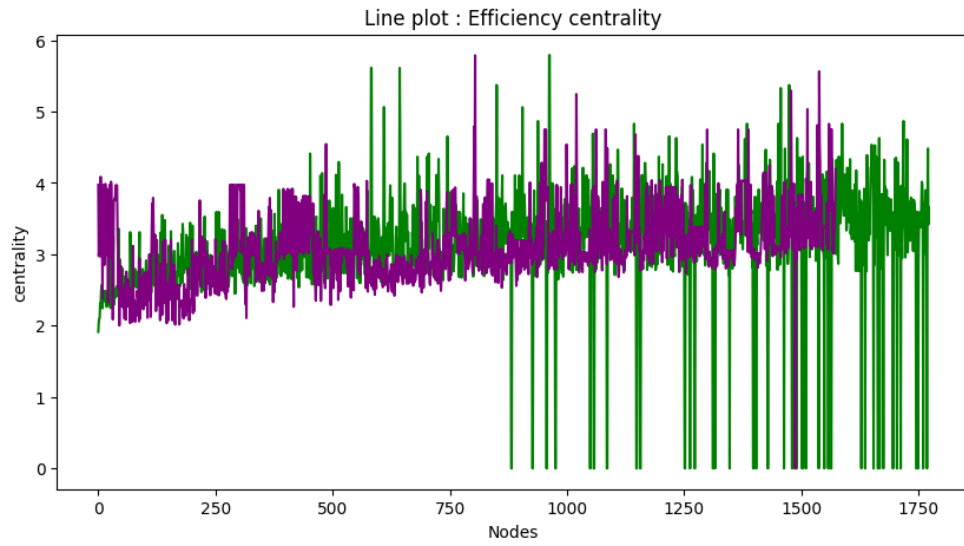
در این بخش 3 نمودار رسم شده است:

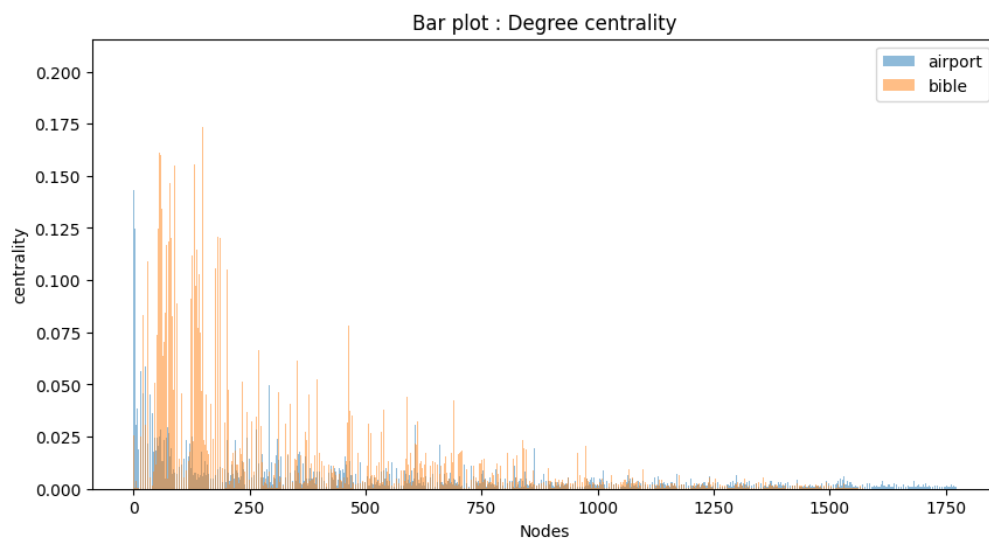
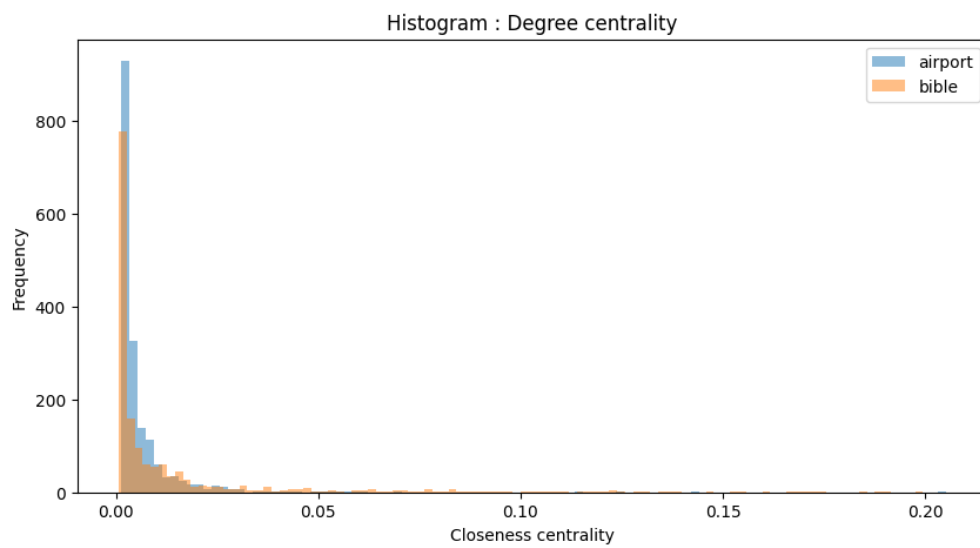
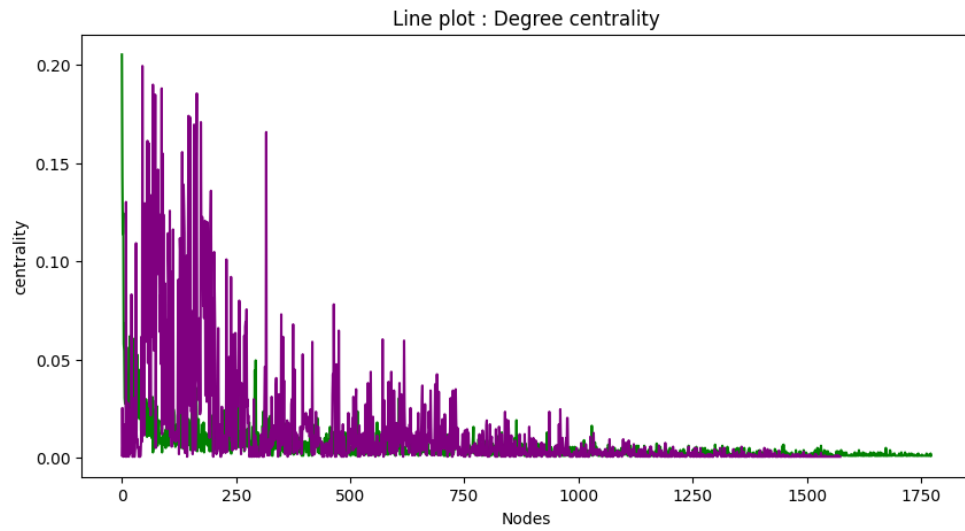
نمودار خواسته شده، هیستوگرام، نمودار bar

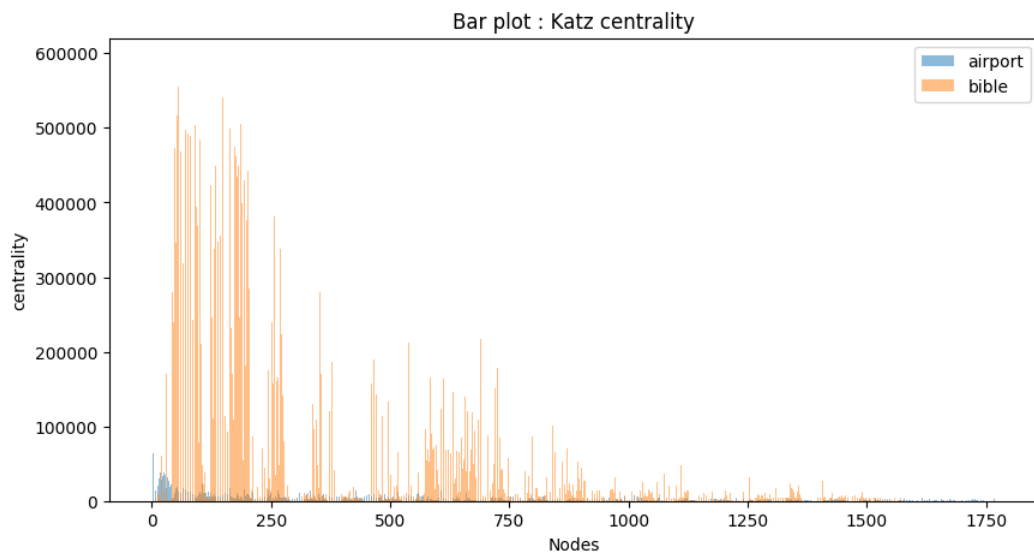
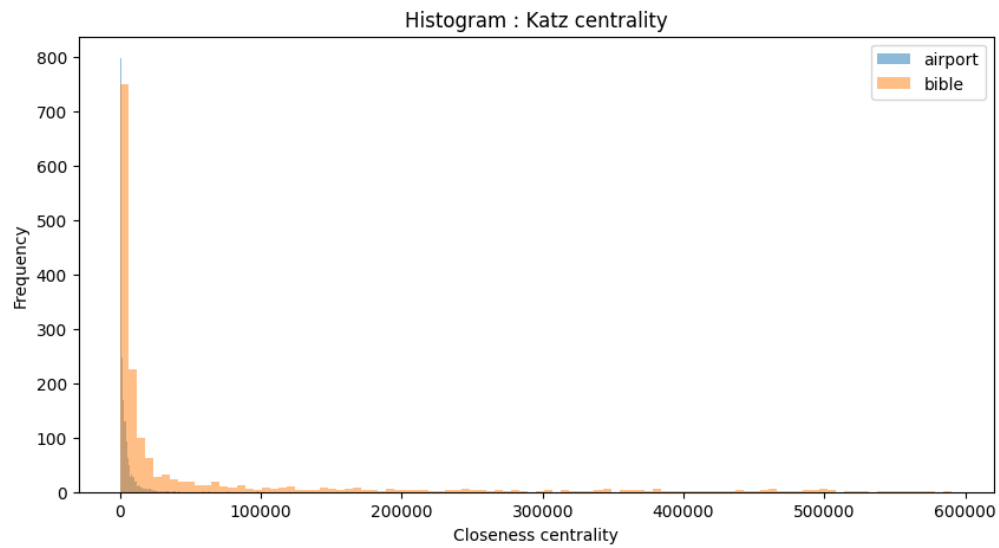
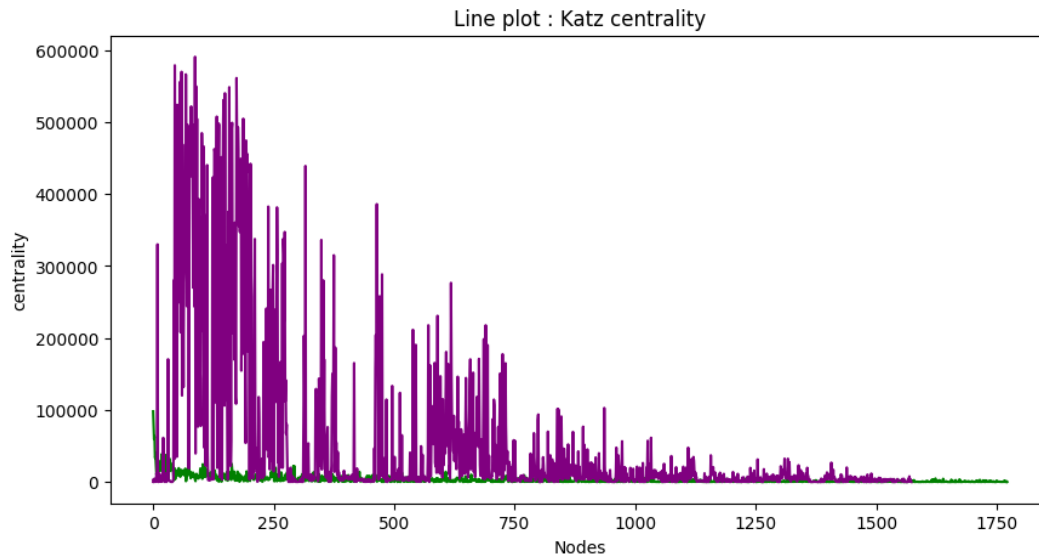
نتایج به صورت زیر است: (نمودار های سبز برای مجموعه داده bible و نمودار بنفش برای مجموعه داده airport است)











(ج)

:Closeness

در تحلیل نمودار های آن مشاهده می کنیم، مجموعه داده airport در اکثر موارد در یک محدوده مشخص Closeness آن ها تغییر میکند و نشان می دهد فاصله نود ها و نزدیکی آن ها به یکدیگر تقریبا به یک اندازه است ولی در Bible مشاهده می شود که بعضی از نود ها Closeness خیلی پایین یا خیلی بالایی دارند که نشان میدهد بعضی از نود ها نزدیکی بیشتری با نود های دیگر دارند و بعضی خیلی مرتبط نیستند.

:Efficiency

در این قسمت نیز مثل قبل مشاهده می شود در مجموعه داده airport در اکثر موارد در یک محدوده مشخص Efficiency آن ها تغییر میکند و نشان می دهد فاصله نود ها و نزدیکی آن ها به یکدیگر تقریبا به یک اندازه است ولی در Bible مشاهده می شود که بعضی از نود ها Efficiency خیلی پایین یا خیلی بالایی دارند که نشان میدهد بعضی از نود ها توانایی انتقال اطلاعات (ability to spread information) بالا و بعضی توانایی کم دارند.

:Degree

از روی نمودار ها میتوان تحلیل کرد که در گراف airport نود هایی با درجه بالاتر بیشتر از گراف bible هستند. یعنی نود ها بیشتر بهم متصل هستند و مثلا هر نود به نود های زیادی متصل است. این نکته تایید کننده دو تحلیل قبلی می باشد.

:Katz

از روی نمودار ها می توان تحلیل کرد، در مجموعه داده airport داده ها تاثیر گذارتر از مجموعه داده bible هستند، یعنی نود های مستقیم و همسایگان بیشتری دارند و گره های بیشتری در شبکه از طریق این همسایه های فوری به گره مورد نظر متصل می شوند.

در این قسمت نیز داده های ذخیره ده را میخوانیم. سپس در مجموعه داده های مربوط به معیار های Katz, Efficiency, Closeness و 10 Degree تا نود با بزرگترین مقدار را به عنوان بهترین نود ها انتخاب میکنیم و در یک دیتافریم میریزیم. سپس عدد هر نود را با ID آن در مجموعه داده نام مطابقت داده و در نهایت اسم نود ها را جایگزین ID آن ها میکنیم.

| ID | AirportName |
|------|-------------|
| 0 | 1 |
| 1 | 2 |
| 2 | 3 |
| 3 | 4 |
| 4 | 5 |
| ... | ... |
| 1853 | 1854 |
| 1854 | 1855 |
| 1855 | 1856 |
| 1856 | 1857 |
| 1857 | 1858 |

| ID | biblicalNouns |
|------|---------------|
| 0 | 1 |
| 1 | 2 |
| 2 | 3 |
| 3 | 4 |
| 4 | 5 |
| ... | ... |
| 1768 | 1769 |
| 1769 | 1770 |
| 1770 | 1771 |
| 1771 | 1772 |
| 1772 | 1773 |

| top results for airport is : | | | | |
|------------------------------|-----------|------------|--------|------|
| | closeness | efficiency | degree | katz |
| 0 | 114 | 804 | 114 | 417 |
| 1 | 877 | 1538 | 709 | 1025 |
| 2 | 1068 | 1478 | 1200 | 405 |
| 3 | 389 | 1020 | 877 | 1089 |
| 4 | 709 | 1513 | 766 | 1511 |
| 5 | 766 | 1559 | 389 | 1777 |
| 6 | 982 | 1534 | 500 | 1801 |
| 7 | 1200 | 802 | 1068 | 1815 |
| 8 | 500 | 1062 | 711 | 1842 |
| 9 | 711 | 1365 | 1016 | 1849 |

| | closeness | efficiency | degree | katz |
|---|-----------|------------|--------|------|
| 0 | ATL | KFP | ATL | DPA |
| 1 | LAX | SWB | IAD | MKL |
| 2 | MSP | SLW | ORD | DMA |
| 3 | DEN | MIL | LAX | MVW |
| 4 | IAD | STE | JFK | SSW |
| 5 | JFK | TAO | DEN | YGE |
| 6 | MCO | SVO | EWR | YPW |
| 7 | ORD | KEH | MSP | YTG |
| 8 | EWR | MS2 | IAH | ZEL |
| 9 | IAH | RKS | MIA | ZTS |

| top results for bible is : | | | |
|----------------------------|-----------|------------|--------|
| | closeness | efficiency | degree |
| 0 | 1 | 962 | 114 |
| 1 | 2 | 963 | 709 |
| 2 | 3 | 582 | 1200 |
| 3 | 4 | 583 | 877 |
| 4 | 10 | 643 | 766 |
| 5 | 27 | 850 | 389 |
| 6 | 29 | 1474 | 500 |
| 7 | 19 | 1456 | 1068 |
| 8 | 34 | 609 | 711 |
| 9 | 6 | 610 | 1016 |

| | closeness | efficiency | degree |
|---|-----------|------------|----------|
| 0 | israel | jimnah | levi |
| 1 | judah | jared | achsah |
| 2 | jerusalem | zur | jose |
| 3 | david | almodad | zimmah |
| 4 | egypt | sheshai | rahab |
| 5 | ephrain | mehetabel | ziklag |
| 6 | manasseh | titus | henadad |
| 7 | benjamin | paulus | timnath |
| 8 | joseph | pontius | magdiel |
| 9 | moses | riphath | jehdeiah |

(۵)

معيار Closeness:

در این جداول به نظر میرسد در جدول airport، فرودگاه هایی که بسیار مهم هستند و معمولاً shortest path هستند را نشان میدهد یعنی برای سفر به جایی میتوان اول به ATL سفر کرد و بعد به مقصد رفت. در جدول bible نیز میتوان از کلمه isreal، در مسیر کوتاه تری به بقیه کلمات رسید.

معيار Efficiency:

نود ها در این قسمت نشان می دهند که بهترین فرودگاه ها و کلمات کارآمد در کل مجموعه داده کدام هستند.

معيار Degree:

نشان میدهند که کدام فرودگاه ها، از مقاصد مختلف پرواز دارند، شاید تعداد این پرواز ها از تعداد پرواز های انجام شده در قسمت closeness کمتر باشد، ولی نشان می دهد فرودگاه های بیشتری به فرودگاه مثلا IAD پرواز دارند نسبت به ORD. یا در bible کلمه levi در کنار کلمات مختلف بیشتری استفاده شده است.

هر کدام از معیار ها میتواند بنا به درخواست ما مهم باشد و با توجه به توضیحات داده شده میتوان متناسب با خروجی دلخواه، معیار مناسب انتخاب کنیم.

سوال 3

در این سوال در ابتدا فایل edges را میخوانیم و سپس نمایش می دهیم.

| | 0 | 1 | 2 |
|-----------------------|------|-----|-----|
| 0 | 1 | 2 | 100 |
| 1 | 3 | 2 | 78 |
| 2 | 4 | 1 | 73 |
| 3 | 4 | 5 | 67 |
| 4 | 1 | 6 | 67 |
| ... | ... | ... | ... |
| 9126 | 558 | 111 | 1 |
| 9127 | 1042 | 424 | 1 |
| 9128 | 130 | 6 | 1 |
| 9129 | 280 | 414 | 1 |
| 9130 | 353 | 528 | 1 |
| 9131 rows × 3 columns | | | |

فایل bible.edges

سپس گرافی ساخته و ستون های 0 و 1 را به عنوان نود های متصل می دهیم. در نهایت گراف زیر تشکیل می شود.

Graph with 1773 nodes and 9131 edges

(الف)

در این قسمت ماتریس مجاورت متناظر با گراف و ماتریس درجه را می سازیم:

```
matrix of degrees is :
[[364.  0.  0. ...  0.  0.  0.]
 [ 0. 254.  0. ...  0.  0.  0.]
 [ 0.  0. 202. ...  0.  0.  0.]
 ...
 [ 0.  0.  0. ...  2.  0.  0.]
 [ 0.  0.  0. ...  0.  4.  0.]
 [ 0.  0.  0. ...  0.  0.  2.]]
```

```
laplacian matrix is :
[[364. -1. -1. ... 0. 0. 0.]
 [ -1. 254. -1. ... 0. 0. 0.]
 [ -1. -1. 202. ... 0. 0. 0.]
 ...
 [ 0. 0. 0. ... 2. 0. 0.]
 [ 0. 0. 0. ... 0. 4. 0.]
 [ 0. 0. 0. ... 0. 0. 2.]]
laplacian shape is : (1773, 1773)
```

در نهایت با استفاده از فرمول روبه‌رو، ماتریس لاپلاسیان را تشکیل می‌دهیم:

$$L = A - D$$

و نتیجه برابر زیر است:

```
[[364. -1. -1. ... 0. 0. 0.]
 [ -1. 254. -1. ... 0. 0. 0.]
 [ -1. -1. 202. ... 0. 0. 0.]
 ...
 [ 0. 0. 0. ... 2. 0. 0.]
 [ 0. 0. 0. ... 0. 4. 0.]
 [ 0. 0. 0. ... 0. 0. 2.]]
```

ماتریس لاپلاسیان

(ب)

در این قسمت به منظور استخراج بردارها و مقادیر ویژه از دستور `np.linalg.eig(Laplacian)` استفاده میکنیم و سپس مقادیر `EigenVector` ها را استخراج کرده و به ترتیب نزولی مرتب می‌کنیم:

```
sorted eigenvalues are :
[-4.46139904e-14 -8.90326803e-15 -6.47322745e-15 ...  2.22074285e+02
 2.55063946e+02  3.65027518e+02]

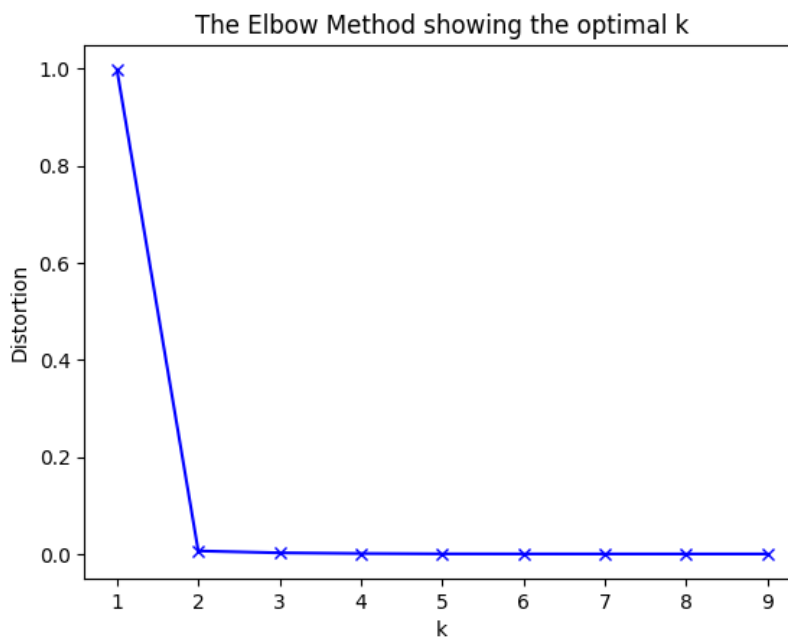
sorted eigenvectors are :
[[ 2.42037824e-02  2.73271444e-21 -6.72048445e-18 ...  2.96473093e-03
  2.99922860e-03  9.98546027e-01]
 [ 2.42037824e-02  1.79210931e-17 -5.75859294e-18 ...  1.62543265e-02
  9.97615066e-01 -4.93642513e-03]
 [ 2.42037824e-02 -1.62052269e-17 -1.02230029e-17 ... -3.47483019e-02
 -1.05113376e-02 -4.04574543e-03]
 ...
 [ 2.42037824e-02  4.51310128e-17 -1.77730863e-17 ...  1.55249574e-09
 -7.05962420e-08 -2.30515516e-08]
 [ 2.42037824e-02  2.09743367e-16  1.10346001e-16 ...  5.05846856e-05
 -2.18184319e-06  1.71219811e-05]
 [ 2.42037824e-02 -6.62863372e-16  1.92721272e-17 ... -6.39479041e-07
 1.56394236e-05  7.61276370e-06]]
```

سپس بنا به خواسته سوال بردار متناظر با دومین کوچک ترین را نمایش می‌دهیم:

```
second smallest eigenvalue is : -8.903268033317957e-15
second smallest eigenvector is :
[[ 2.73271444e-21]
 [ 1.79210931e-17]
 [-1.62052269e-17]
 ...
 [ 4.51310128e-17]
 [ 2.09743367e-16]
 [-6.62863372e-16]]
```

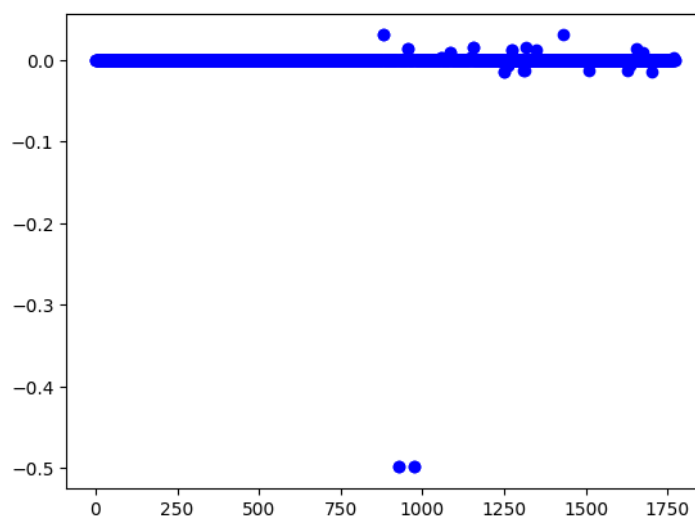
(ج)

از بردار استخراج شده در قسمت قبل به عنوان بردار ویژگی برای خوشه بندی با الگوریتم kmeans استفاده می کنیم.
برای آنکه بهترین مقدار را برای k استفاده کنیم از روش elbow در الگوریتم kmeans استفاده میکنیم. که شکل 1 نتیجه حاصل از آن می باشد:



شکل 1. نمودار روش elbow در الگوریتم k-means

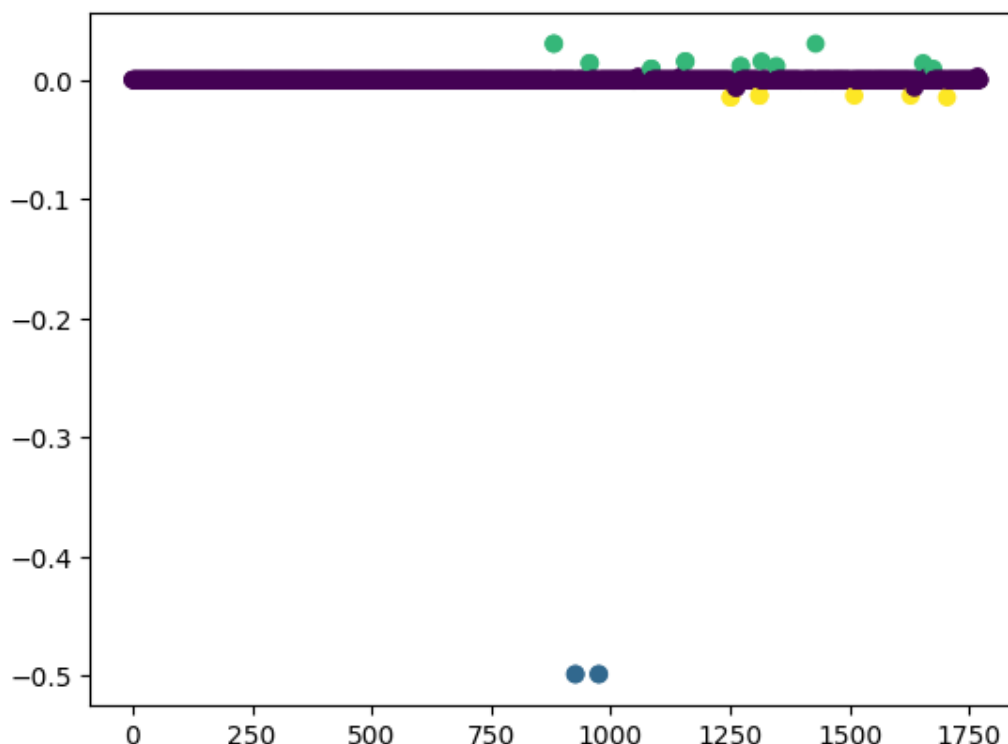
برای تحلیل بهتر این نمودار، نود ها را در یک بعد رسم میکنیم که شکل 2 نتیجه آن است:



شکل 2. نمودار یک بعدی نود ها

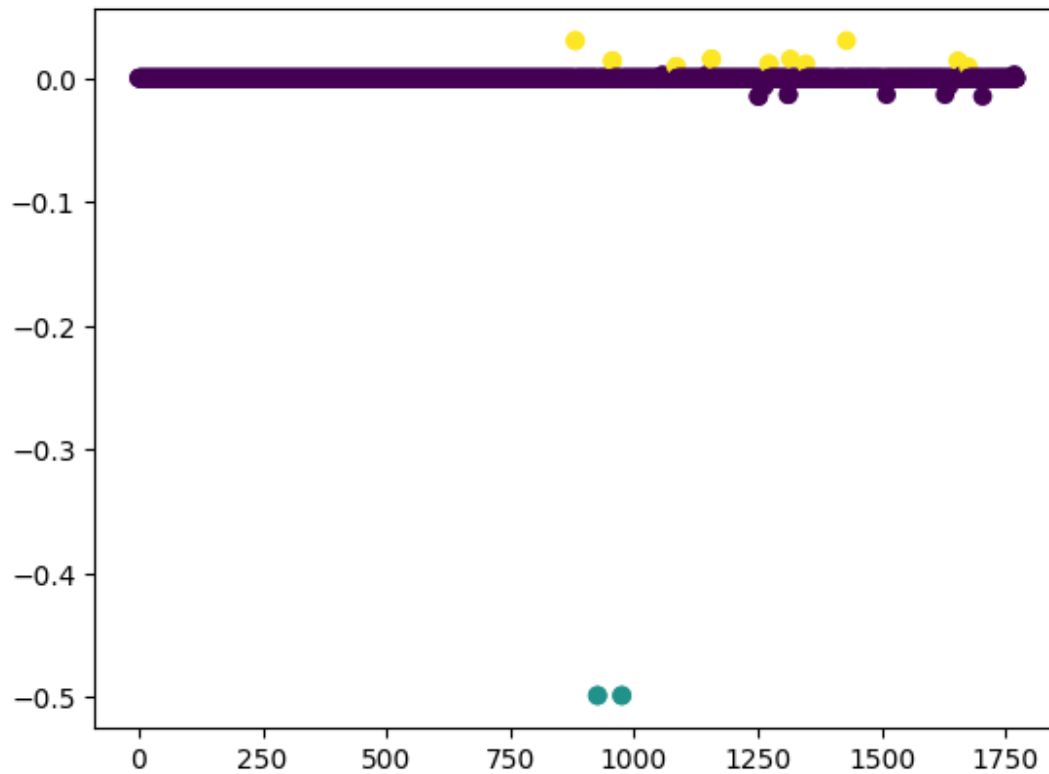
همانطور که در شکل 1 میبینید، با افزایش نود ها، distortion نیز کاهش می یابد. Distortion بعد از آنکه مقدار k زیاد می شود، تقریباً همگرا میشود. با توجه به نمودار، به نظر می رسد بهترین تعداد k میتواند 4 باشد. چرا که یک دسته از نود ها بر روی صفر قرار دارند، دسته ای دیگر بیشتر از یک، کمتر از یک، و یک دسته با فاصله کمتر از یک می باشد. اما در صورتی که مقدار k برابر 2 در نظر گرفته شود، دو دسته تشکیل می شود. دسته اول مقادیر در محدوده صفر و دسته ایی دیگر نود هایی که با صفر فاصله بیشتری دارند.

برای پیدا کردن بهترین مقدار برای k ، الگوریتم k -means با مقادیر 2، 3، و 4 برای k پیاده سازی شده است و نتایج حاصل از دسته بندی به صورت زیر است:



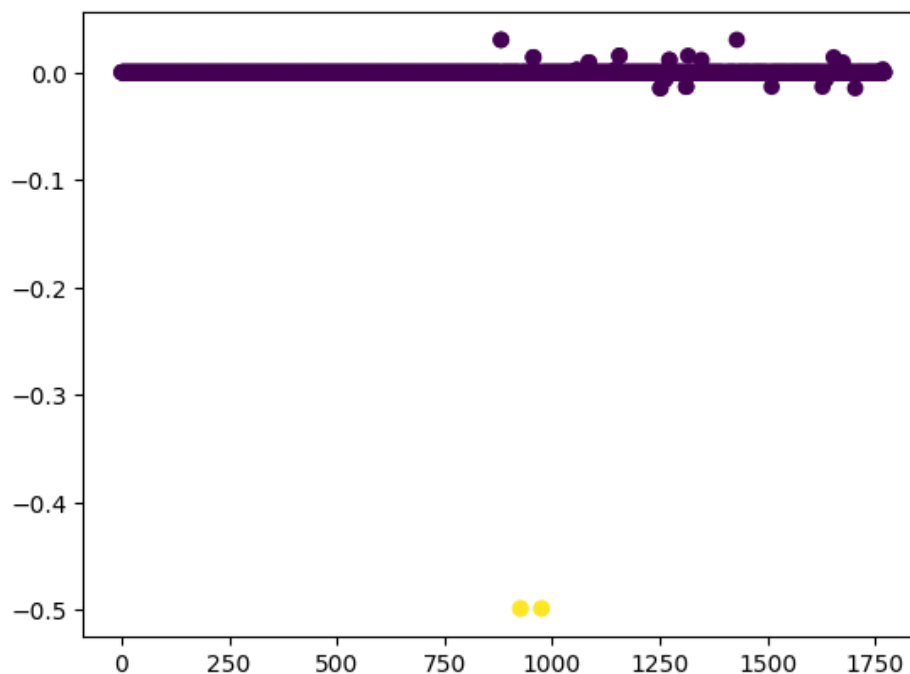
شکل 3. Kmean – $k=4$

```
k is 4 :
cluster centers are : [[-6.81182728e-06]
[-4.98438366e-01]
[ 1.60824561e-02]
[-1.39456314e-02]]
cluster labels are : [0 0 0 ... 0 0 0]
cluster labels length is : (1773,)
```



شكل 4. k=3 - Kmean

```
k is 3 :
cluster centers are : [[-6.24399554e-05]
[-4.98438366e-01]
[ 1.60824561e-02]]
cluster labels are : [0 0 0 ... 0 0 0]
cluster labels length is : (1773,)
```



شکل 5. Kmean – k=2

```
k is 2 :
cluster centers are : [[ 7.44585412e-05]
 [-4.98438366e-01]]
cluster labels are : [0 0 0 ... 0 0 0]
cluster labels length is : (1773,)
```

(د)

در این قسمت به محاسبه min cut و modularity پرداخته می‌شود. min cut معیاری است که تعداد یال‌های که بین دو کلاستر وجود دارند را نشان می‌دهد. پس هر چقدر معیار min cut کمتر باشد، کلاسترینگ بهتر انجام شده است.

این معیار برای خوشه‌بندی‌های با مقادیر 2، 3، و 4 برای k اعمال شده است و نتایج آن به شرح زیر است:

```
mincut for kmeans with k=4 is : 0.0
mincut for kmeans with k=3 is : 0.0
mincut for kmeans with k=2 is : 0.0
```

این نتایج نشان‌دهنده این است که هر سه مقدار، نتیجه بسیار خوبی داشته‌اند و خوشه‌بندی به خوبی انجام شده است. ولی در مقدار k=4 تعداد خوشه‌ها زیاده‌تر است و همچنین بین خوشه‌ها یالی وجود ندارد. پس می‌توان آن را ترجیح داد.

Modularity معیاری است که خوشه بندی را ارزیابی می‌کند و از فرمول زیر بدست می‌آید:

$$Q = \frac{1}{2m} \sum_{i,j} (A_{ij} - \gamma \frac{k_i k_j}{2m}) \delta(c_i, c_j)$$

که A ماتریس مجاورت و K_i برابر درجه نود i است. در صورتی که i و j در یک خوشه باشند، مقدار زیر برابر 1 و در غیر اینصورت برابر صفر است.

$$\delta(c_i, c_j)$$

مقادیر modularity برای خوشه‌های با مقادیر 2، 3، و 4 برای k اعمال شده است و نتایج آن به شرح زیر است:

```
modularity for k=4 is : 0.006556125270962652
modularity for k=3 is : 0.0045912954819048325
modularity for k=2 is : 0.001313340792245858
```

همانطور که مشاهده میشود، گرچه این مقادیر تفاوت چندانی باهم ندارند ولی همانطور که در قسمت خوشه بندی دیدیم، خوشه بندی با $k=4$ بهتر بوده و modularity بالاتری دارد.

سوال 4

در ابتدا مجموعه داده مورد نظر را خوانده و تعداد نود و یال برابر زیر است:

```
Number of nodes: 7115
Number of edges: 100762
```

در این قسمت در ابتدا تابعی برای محاسبه مقدار modularity matrix می‌نویسیم. که از فرمول زیر بدست می‌آید.

$$B_{ij} = A_{ij} - \frac{k_i k_j}{2m}$$

سپس تابعی برای محاسبه بزرگترین بردار ویژه می‌نویسیم و در نهایت تابعی به نام binary_classification می‌نویسیم تا گراف را به دو بخش تقسیم کند. حال در ابتدا مقدار اولیه modularity را برابر صفر در نظر می‌گیریم و سپس به صورت بازگشتی روند زیر را طی می‌کنیم:

بزرگترین بردار ویژگی متناظر را بدست می‌آوریم

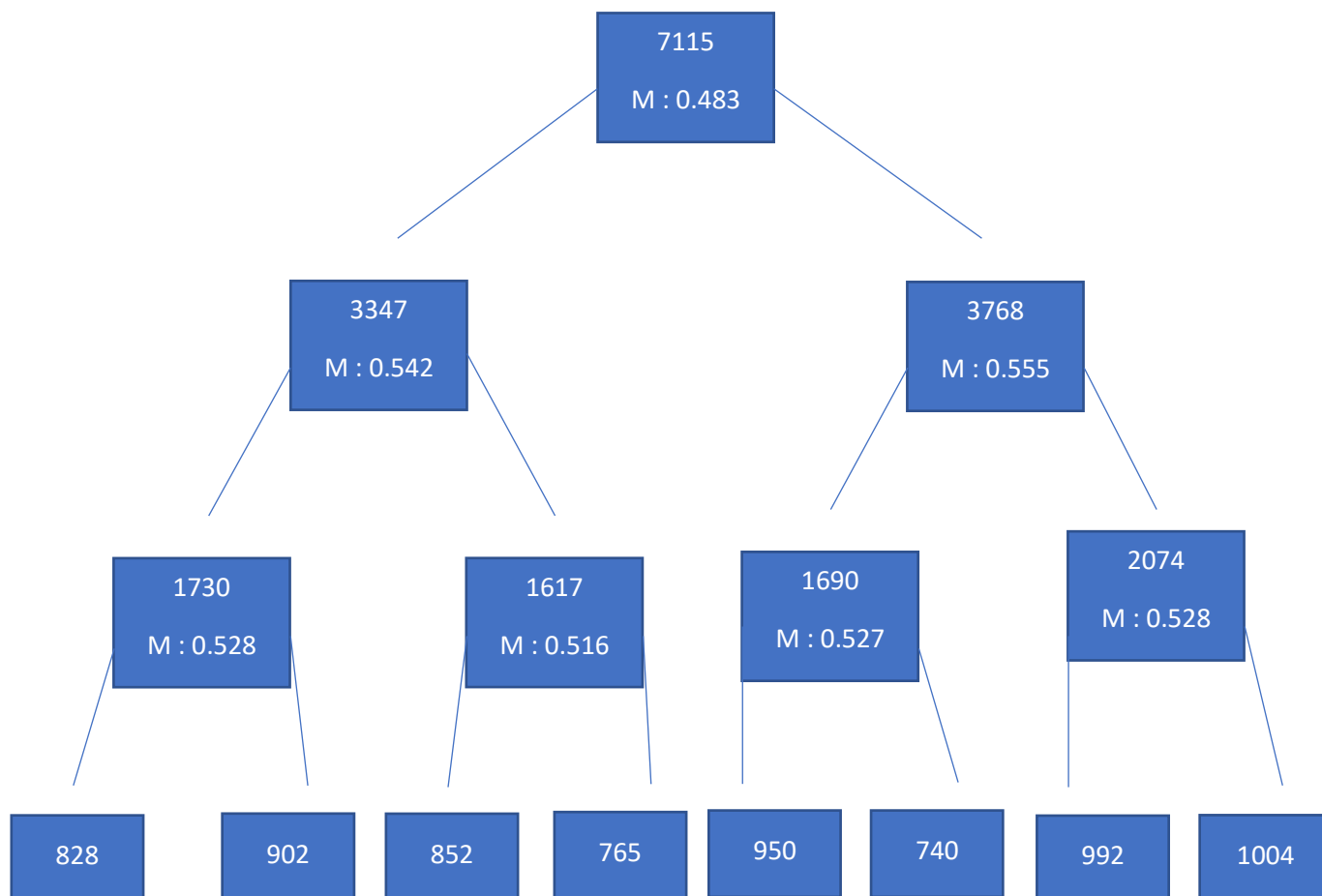
آن را به تابع binary_classification می‌دهیم و گراف را به دو قسمت تقسیم می‌کنیم و هر دوی آن را ذخیره می‌کنیم.

سپس مقدار modularity را برای گراف با بردار ویژگی در نظر گرفته شده محاسبه می‌کنیم

و آن را با مقدار قبلی مقایسه می‌کنیم، در صورتی که مقدار آن بزرگ تر بود، قسمت های تقسیم شده را دوباره وارد سیستم می‌کنیم.

این کار را تا زمانی انجام می‌دهیم که مقدار modularity از مقدار قبلی کمتر شده باشد. در این حالت، دیگر خوشه بندی در آن قسمت را ادامه نمی‌دهیم.

نتیج حاصل از اعمال این کد به شرح زیر است:



بر طبق نتایج بدست آمده، خوشه بندی فقط تا زمانی که 4 خوه به وجود بیاید ادامه می یابد چرا که modularity کاهش می یابد. یعنی 4 دسته با تعداد نود های 1730، 1617، 1690، 2074 به وجود می آید. و با کاهش modularity یگر این خوشه ها به دو قسمت دیگر تقسیم نمی شوند.

