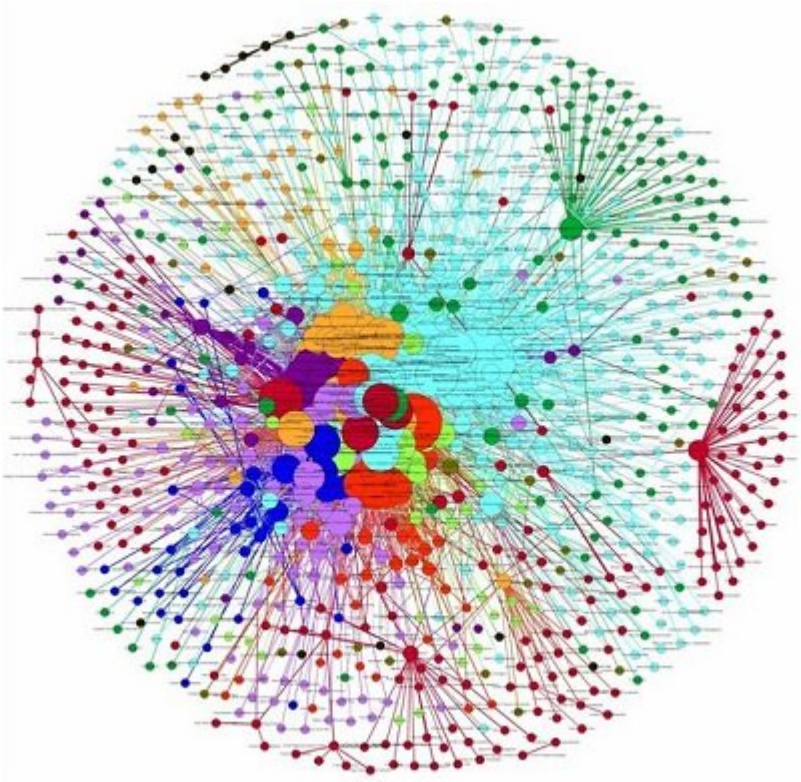
تحلیل اطلاعات اجتماعی

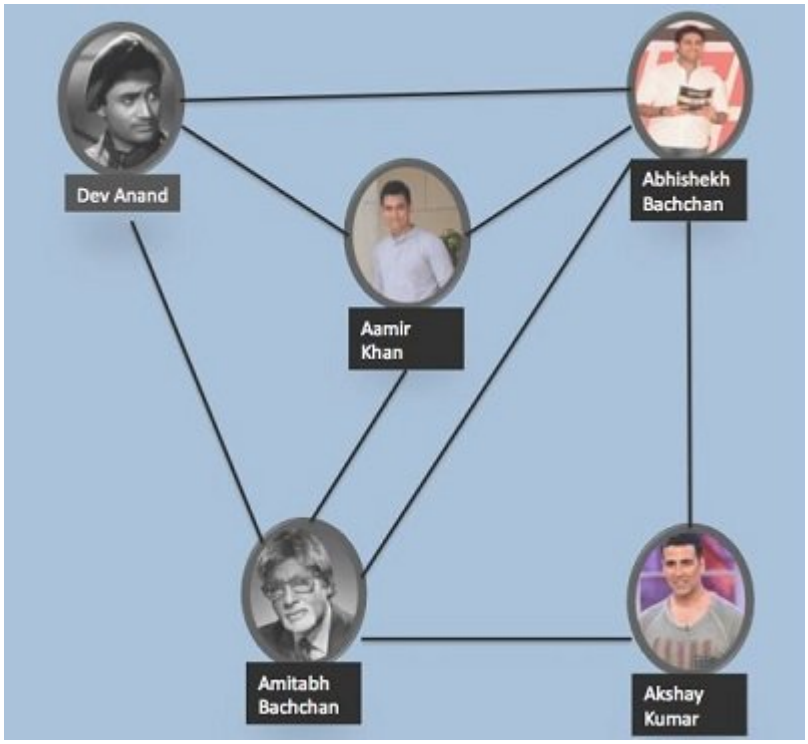
چگونه می‌توانیم با پایتون شبکه‌های اجتماعی را تجزیه‌وتحلیل کنیم؟



شبکه‌های اجتماعی و ارتباطی به اشکال مختلف در هر مکانی قرار دارند. شبکه جاده‌ای، شبکه دوستان و دنبال‌کنندگان در جامعه مجازی و شبکه‌ای از همکاران اداری از جمله این موارد هستند. هرکدام از این شبکه‌ها نقش خاص خود را در زندگی روزمره ما ایفا می‌کنند که از انتشار و پخش اخبار مفید گرفته تا برگزاری انتخابات ملی را شامل می‌شوند. توانایی تجزیه‌وتحلیل این شبکه‌ها و اتخاذ تصمیمات آگاهانه مبتنی بر این اطلاعات مهارتی است که هر تحلیل‌گر داده‌ای باید از آن برخوردار باشد. هدف از این مقاله آموزش تجزیه‌وتحلیل شبکه اجتماعی (SNA) با استفاده از پایتون و NetworkX به‌عنوان یک کتابخانه پایتون برای مطالعه و بررسی ساختار، پویایی و عملکرد شبکه‌های پیچیده است. در این راهنما فرض بر این است که خواننده با اصول مرتبط با دستورات پایتون آشنا است. برای درک این مطالب ضرورتی ندارد که پیش‌نیاز قبلی در ارتباط با نحوه تجزیه‌وتحلیل شبکه اجتماعی (SNA) داشته باشید.

معرفی

اجازه دهید ابتدا کمی درباره مفهوم شبکه‌های اجتماعی بحث کنیم. در شکل1 شبکه‌ای از بازیگران بالیوود را مشاهده می‌کنید که با عنوان نود یا گره با یکدیگر ارتباط دارند. هر کدام از این خط‌های متصل نشان‌دهنده آن است که آن‌ها حداقل در یک فیلم مشترک با هم همکاری داشته‌اند.



این یک شبکه اجتماعی محسوب می‌شود. هر شبکه با وضعیت ارتباطات بین هر یک از افراد مستقل درون آن تعریف می‌شود. این اتصالات رابطه بین اشخاص را مشخص می‌کند. زمانی‌که این ارتباطات و اتصالات حجم بسیار گسترده‌ای پیدا می‌کنند، کارشناسان می‌توانند درباره یک گروه از مردم و حتی مردمی که در یک کشور زندگی می‌کنند، اطلاعات تحلیل‌شده دقیقی را به دست آورند. تجزیه‌وتحلیل این شبکه‌ها درک وسیعی در مورد افراد درون آن شبکه، از جمله این‌که چه کسانی تاثیر واقعی روی دیگران دارند یا چه کسانی بیشترین ارتباطات را دارند و ... در اختیار ما قرار می‌دهد.

مطلب پیشنهادی

کدنویسی منطقی و درست
اگر برنامه‌نویس پایتون هستید از این دو اشتباه دوری کنید

عناصر تشکیل‌دهنده یک شبکه

گره‌ها (Nodes): نشان‌دهنده تک‌تک افراد موجود در شبکه‌ای است که ما می‌سازیم، مانند شبکه بازیگران.
لبه‌ها (Edges): نشان‌دهنده ارتباط بین این گره‌ها است. لبه‌ها بیانگر رابطه بین گره‌های موجود در یک شبکه است. در مثال ما این رابطه هم‌بازی بودن بازیگران را نشان می‌دهد.

ساخت یک شبکه با استفاده از NetworkX

انواع مختلفی از شبکه‌ها وجود دارد. ما از NetworkX می‌توانیم برای توسعه و تجزیه‌وتحلیل شبکه‌های مختلف استفاده کنیم. برای شروع NetworkX را نصب کرده و از فرمان زیر استفاده کنید:

```
pip install networkx
```

اگر در Anaconda کار می‌کنید، از این فرمان استفاده کنید:

```
conda install -c anaconda networkx
```

این فرامین آخرین نسخه از Networkx را نصب خواهد کرد. کدهای به‌کاررفته در این مقاله در نسخه Python=3.5, NetworkX = 2.0 اجرا شده‌اند.

شبکه‌های متقارن

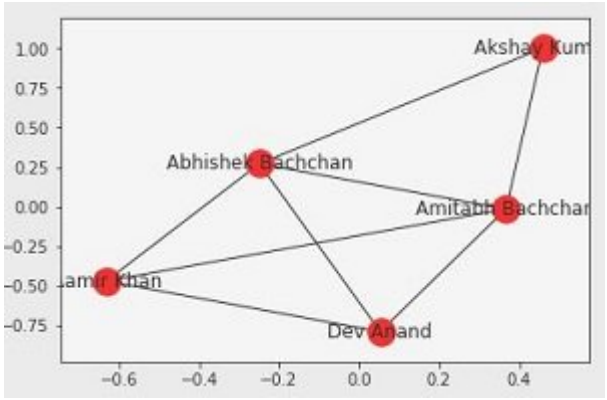
شبکه بازیگرانی که ما در بالا ایجاد کردیم، یک شبکه متقارن است، زیرا رابطه «کار با یکدیگر در یک فیلم» یک

رابطه متقارن است. اگر A مرتبط با B است، B نیز مرتبط با A است. در ادامه شبکه‌ای را که در بالا مشاهده کردید در NetworkX می‌سازیم.
ما از متد Graph() برای ساخت یک شبکه جدید و add_edge() برای اضافه کردن لبه و مرز بین دو گره استفاده می‌کنیم. (فهرست 1)

```
import networkx as nx
G_symmetric = nx.Graph()
G_symmetric.add_edge('Amitabh Bachchan','Abhishek Bachchan')
G_symmetric.add_edge('Amitabh Bachchan','Aamir Khan')
G_symmetric.add_edge('Amitabh Bachchan','Akshay Kumar')
G_symmetric.add_edge('Amitabh Bachchan','Dev Anand')
G_symmetric.add_edge('Abhishek Bachchan','Aamir Khan')
G_symmetric.add_edge('Abhishek Bachchan','Akshay Kumar')
G_symmetric.add_edge('Abhishek Bachchan','Dev Anand')
G_symmetric.add_edge('Dev Anand','Aamir Khan')
```

فهرست 1

برای نمایش بصری این شبکه از nx.draw_networkx(G_symmetric) استفاده می‌کنیم. (شکل 2)



شبکه‌های نامتقارن

اگر رابطه بین گره‌ها «والد فرزندی» باشد، این رابطه دیگر متقارن نخواهد بود. به‌عبارت‌دیگر، اگر A فرزند B باشد، بنابراین دیگر B فرزند A نخواهد بود. یک شبکه نامتقارن، شبکه‌ای است که روابط آن نامتقارن هستند. (اگر A به B مرتبط است، لزوما به این معنا نیست که B هم به A مرتبط است.) ما می‌توانیم در NetworkX با استفاده از متد DiGraph (سرنام Directional Graph) یک شبکه نامتقارن بسازیم. (فهرست 2)

```
G_asymmetric = nx.DiGraph()
G_asymmetric.add_edge('A','B')
G_asymmetric.add_edge('A','D')
G_asymmetric.add_edge('C','A')
G_asymmetric.add_edge('D','E')
```

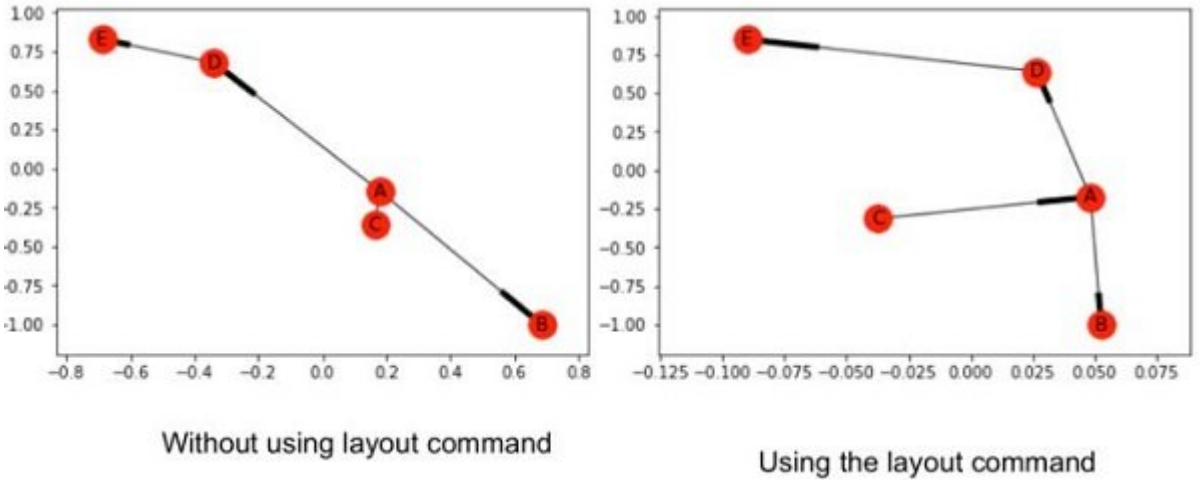
فهرست 2

همانند مثال قبل ما می‌توانیم با استفاده از تابع draw_networkx() آن را به شکل قابل رویت درآوریم. اما این احتمال وجود دارد که گره‌ها به‌صورت مجزا نباشند و در ترسیم شبکه به‌صورت مجزا نمایش داده شوند. برای رسیدگی به این موضوع ما می‌توانیم از تابع
spring_layout() همراه با تابع draw_networkx() استفاده کنیم.

```
nx.spring_layout(G_asymmetric)
```

nx.draw_networkx(G_asymmetric)

در شکل (3) این شبکه را همراه و بدون فرمان layout مشاهده می‌کنید. در نمونه‌ای که از فرمان layout استفاده‌شده، نظم و ترتیب بیشتری مشاهده می‌شود و ساختار آن واضح‌تر است.



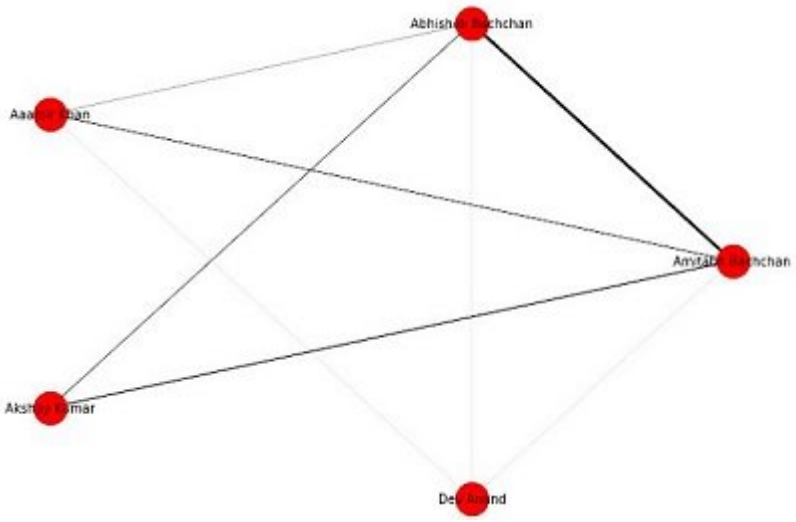
شبکه‌های توزیع‌شده

تا اینجا شبکه‌های ما وزنی نداشتند، اما این امکان وجود دارد که شبکه‌ها به‌صورت توزیع‌شده ساخته شوند. برای مثال، اگر در شبکه اولیه ما تعداد فیلم‌هایی را که بازیگران ما با یکدیگر بازی کرده‌اند به‌عنوان وزن در نظر بگیریم، یک شبکه توزیع‌شده را خواهیم داشت. اجازه دهید یک‌بار دیگر شبکه بازیگران را بسازیم، اما این بار ما به این شبکه وزن اضافه می‌کنیم. هر لبه وزنی دارد که نشان‌دهنده تعداد فیلم‌هایی است که آن‌ها با هم کار کرده‌اند. (فهرست 3)

```
G_weighted = nx.Graph()
G_weighted.add_edge('Amitabh Bachchan','Abhishek Bachchan', weight=25)
G_weighted.add_edge('Amitabh Bachchan','Aaamir Khan', weight=8)
G_weighted.add_edge('Amitabh Bachchan','Akshay Kumar', weight=11)
G_weighted.add_edge('Amitabh Bachchan','Dev Anand', weight=1)
G_weighted.add_edge('Abhishek Bachchan','Aaamir Khan', weight=4)
G_weighted.add_edge('Abhishek Bachchan','Akshay Kumar',weight=7)
G_weighted.add_edge('Abhishek Bachchan','Dev Anand', weight=1)
G_weighted.add_edge('Dev Anand','Aaamir Khan',weight=1)
```

فهرست 3

شکل (4) این شبکه توزیع‌شده از بازیگران را در یک طرح مدور نشان می‌دهد. دقت کنید در این مدل شبکه‌ها عرض لبه‌ها وزن بین دو لبه را مشخص می‌کنند.



اتصالات شبکه

شبکه ما ساخته شد اما باید اطلاعات بیشتری در مورد هر یک از گره‌های موجود در این شبکه به دست آوریم.

رتبه

رتبه (Degree) یک گره نشان‌دهنده تعداد اتصالاتی است که آن گره دارد. NetworkX تابعی به نام Degree دارد که ما می‌توانیم از آن برای مشخص کردن رتبه یک گره در شبکه استفاده کنیم.

```
nx.degree(G_symmetric, ‘Dev Anand`)
```

از آنجا که Dev Anand در شبکه ما تنها با سه بازیگر دیگر کار کرده است، خروجی این تابع 3 خواهد بود.

ضریب خوشه‌بندی

بدون شک افرادی که در یک شبکه اجتماعی ارتباط برقرار می‌کنند، تمایل به تشکیل انجمن دارند. به‌عبارت‌دیگر، در یک شبکه اجتماعی تمایل به ایجاد گروه یا در اصطلاح فنی خوشه (Cluster) وجود دارد. ما می‌توانیم با استفاده از ضریب خوشه محلی (Local Clustering Coefficient) خوشه‌های یک گره را مشخص کنیم. برای مشخص کردن ضریب خوشه محلی ما از تابع nx.clustering(Graph, Node) استفاده می‌کنیم.
در شبکه متقارن بازیگران مشاهده می‌کنید که مقدار ضریب خوشه محلی Dev Anand برابر با 1 است و Abhishek Bachchan ضریب خوشه محلی 0.67 را دارد. میانگین ضریب خوشه‌بندی (مجموع ضرایب خوشه‌بندی محلی تقسیم بر تعداد گره‌ها) برای این شبکه متقارن از بازیگران 0.867 است و ما می‌توانیم این مقدار را با استفاده از فرمان زیر به دست آوریم:

```
nx.average_clustering(G_symmetric)
```

مسیر

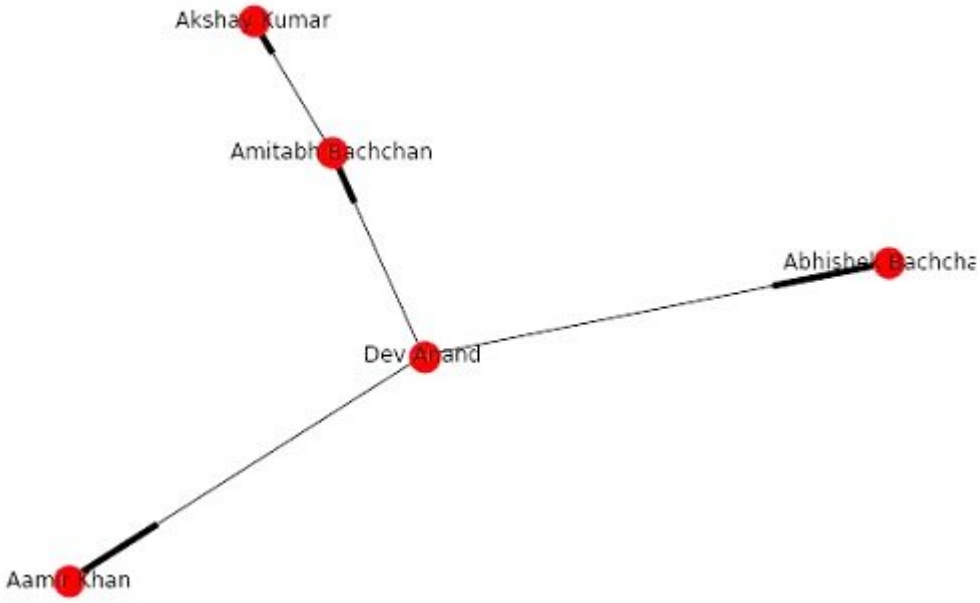
همچنین می‌توانیم کوتاه‌ترین مسیر بین دو گره و طول آن را به ترتیب با استفاده از توابع nx.shortest_path(Graph, Node1, Node2) و nx.shortest_path_length(Graph, Node1, Node2) در NetworkX به دست آوریم. برای این منظور باید فرمان زیر را اجرا کنیم:

```
nx.shortest_path(G_symmetric, ‘Dev Anand’, ‘Akshay Kumar’)
```

با اجرای فرمان فوق، خروجی زیر را مشاهده خواهید کرد:

```
[‘Dev Anand’, ‘Amitabh Bachchan’, ‘Akshay Kumar’]
```

ما می‌توانیم با استفاده از الگوریتم جست‌وجوی Breadth-First فاصله یک گره از سایر گره‌های دیگر موجود در شبکه را پیدا کنیم. NetworkX برای این کار تابع Bfs_Tree را در نظر گرفته است. بنابراین با اجرای فرمان T = nx.bfs_tree(G_symmetric, ‘Dev Anand’) و ترسیم این ساختار شبکه می‌توانیم مسیر رسیدن به سایر گره‌های شبکه را که از Dev Anand آغاز می‌شود، به دست آوریم. (شکل 5)



دوری از مرکز

برای آن‌که بتوانیم فاصله گره A و سایر گره‌های موجود در شبکه را مشخص کرده و به این شکل دوری از مرکز را برای یک گره محاسبه کنیم باید از تابع nx.eccentricity() استفاده کنیم. در شبکه متقارن بازیگران ما، دوری از مرکز Dev Anand برابر با 2 است و مقدار دوری از مرکز Abhishek Bachchan معادل 1 است (به همه متصل است).

تأثیرگذارترین عوامل موجود در شبکه

تا به اینجا ما با برخی از اندازه‌گیری‌های مسیر و فاصله در شبکه آشنا شدیم. این اطلاعات برای آگاهی از وضعیت گستردگی و توازن در یک شبکه بسیار مفید هستند. اما چطور می‌توانیم مهم‌ترین گره‌های موجود در یک شبکه را پیدا کنیم. پیدا کردن مهم‌ترین گره‌های موجود در یک شبکه بر مبنای پارامترهایی انجام می‌شود که ما به آن‌ها معیارهای مرکزیت یا Centrality Measures می‌گوییم. یک دانش‌آموز شایسته در یک مدرسه یا یک ورزشکار مطرح را در نظر بگیرید؛ این‌ها افرادی هستند که می‌توانند در مطرح‌شدن یا نشدن آن مدرسه تاثیرگذار باشند. اما چه عواملی چنین قدرتی را به آن‌ها می‌دهد؟ پاسخ، معیارهای مرکزیت است. معیارهای مرکزیت به ما کمک می‌کنند تا محبوبیت، بیشترین علاقه و تاثیرگذارترین افراد موجود در یک شبکه را پیدا کنیم.

مطلب پیشنهادی

همه آن چیزی که باید بدانید
چگونه یک شبکه ۱۰ گیگابیت بسازیم؟

مرکزیت رتبه

افراد محبوب بیشتر مورد علاقه دیگران هستند و این‌ها همان‌هایی هستند که دوستان بیشتری دارند. مرکزیت رتبه معیاری است که تعداد اتصالات یک گره خاص در شبکه را نشان می‌دهد. این معیار بر این پایه استوار است که

گره‌های مهم‌تر اتصالات بیشتری دارند. برای محاسبه رتبه مرکزیت NetworkX تابع degree_centrality() را در اختیار شما قرار می‌دهد.

مرکزیت بردار ویژه

این معیار نه‌تنها به تعداد اتصالات و روابطی که یک نفر می‌تواند داشته باشد مرتبط است، بلکه نوع افرادی را که یک نفر با آن‌ها در ارتباط است، در نظر می‌گیرد. به‌بیان‌دیگر، این معیار میزان نفوذ یک گره در شبکه را مشخص می‌کند. با استفاده از تابع eigenvector_centrality() در NetworkX می‌توانید معیار مرکزیت بردار ویژه یک گره در شبکه را مشخص کنید.

مرکزیت بینابینی

معیار مرکزیت بینابینی مرکزیت کنترل را تعیین می‌کند. این معیار تعداد دفعاتی را که یک گره خاص به کوتاه‌ترین مسیر انتخابی بین دو گره دیگر می‌رسد، مشخص می‌کند. گره‌هایی که مرکزیت بینابینی بالایی دارند، نقش مهمی در چرخه ارتباطی / اطلاعاتی درون یک شبکه ایفا می‌کنند. گره‌هایی با مرکزیت بینابینی بالا می‌توانند یک کنترل و نفوذ استراتژیک روی سایرین داشته باشند. یک فرد با چنین موقعیت استراتژیکی می‌تواند روی کل مجموعه تاثیرگذار باشد. Networkx یک تابع به نام betweenness_centrality() دارد که به‌وسیله آن می‌توانید این معیار را در شبکه اندازه‌گیری کنید. برای متعادل کردن مقادیر بینابینی گزینه‌هایی برای این تابع تعریف‌شده که از آن‌ها نیز می‌توانید استفاده کنید.

سخن آخر

در این مقاله سعی بر این بوده تا نحوه تجزیه‌وتحلیل یک شبکه اجتماعی ساده را با استفاده از پایتون نشان دهیم. همان‌گونه که مشاهده کردید برای تحلیل درست ارتباطات درون یک شبکه اجتماعی نمی توانید فقط بر ابزارها و کتابخانه‌ها متمرکز باشید، بلکه باید درباره معیارهای مختلفی که در این مقاله با چند مورد از مهم‌ترین آن‌ها آشنا شدید، درک درستی داشته باشید و به‌مرور با افزایش سطح مهارت‌های شما در این زمینه می‌توانید به سراغ شبکه‌های بزرگ‌تری همچون فیس‌بوک و.... بروید.

تاریخ انتشار:
05 اسفند 1397

نشانی منبع:
https://www.shabakeh-mag.com/workshop/14586/%DA%86%DA%AF%D9%88%D9%86%D9%87-%D9%85%DB%8C%E2%80%8C%D8%AA%D9%88%D8%A7%D9%86%DB%8C%D9%85-%D8%A8%D8%A7-%D9%BE%D8%A7%DB%8C%D8%AA%D9%88%D9%86-%D8%B4%D8%A8%DA%A9%D9%87%E2%80%8C%D9%87%D8%A7%DB%8C-%D8%A7%D8%AC%D8%AA%D9%85%D8%A7%D8%B9%DB%8C-%D8%B1%D8%A7-%D8%AA%D8%AC%D8%B2%DB%8C%D9%87%E2%80%8C%D9%88%D8%AA%D8%AD%D9%84%DB%8C%D9%84-%DA%A9%D9%86%DB%8C%D9%85%D8%9F