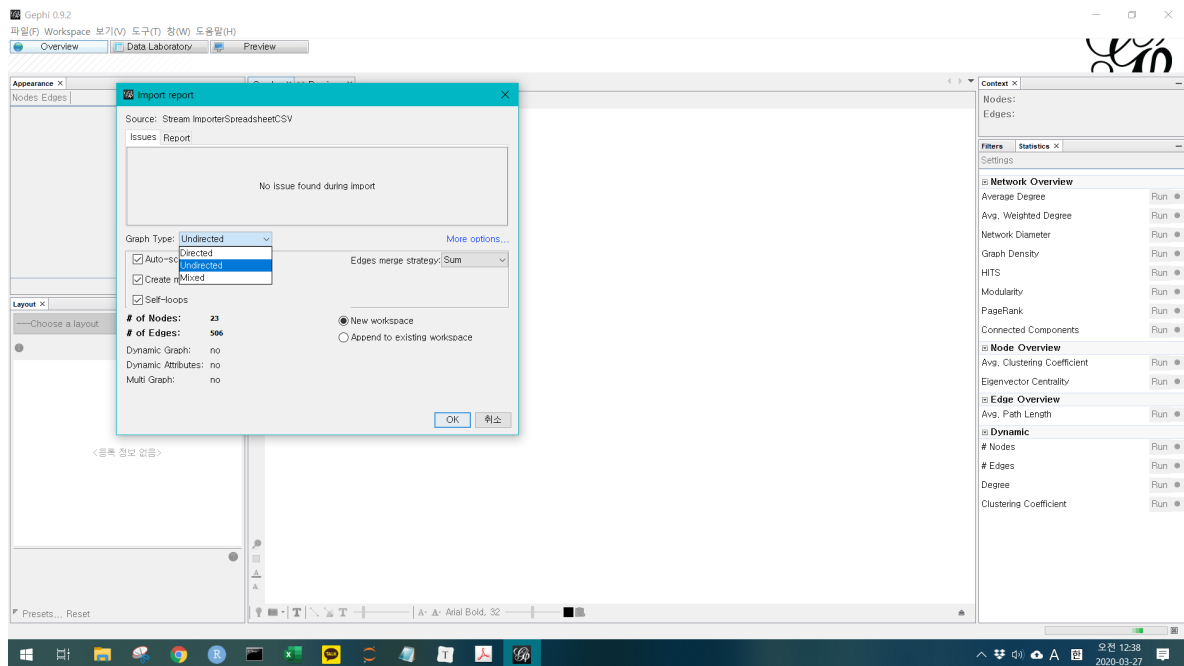


# Gephi 기본 조작

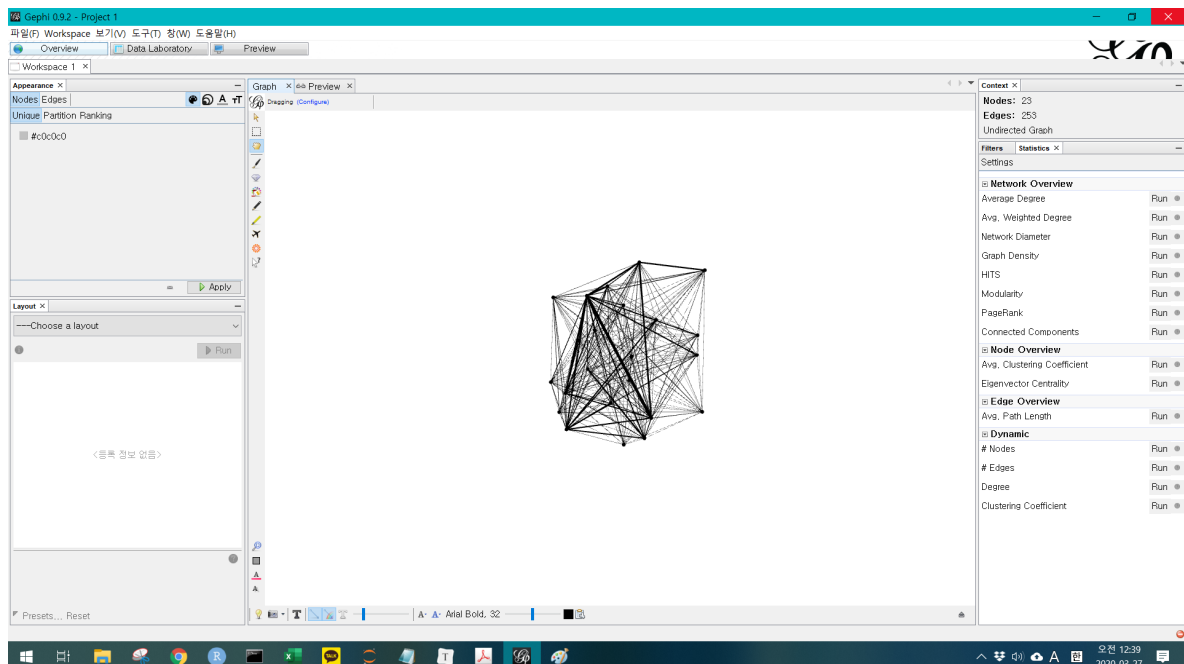
Gephi에서 파일을 불러오고 이쁜 시각화를 위해 해야하는 몇 가지 요소들에 대해서 살펴보자.

## 1. Co-occurrence 행렬 불러오기

co-occurrence 행렬에 대한 개념은 [여기](#)를 참조하면 된다. 주의할 점은, 구축한 co-occurrence 행렬은 대칭 행렬이므로, 방향성이 존재하지 않는다. 예를 들어, les misérables 데이터는 등장인물 A가 B에게 말하는 방향성이 존재하지만, co-occurrence 행렬은 이러한 개념이 없다. 따라서 아래 그림의 Graph Type에서 Undirected를 선택해야 한다.



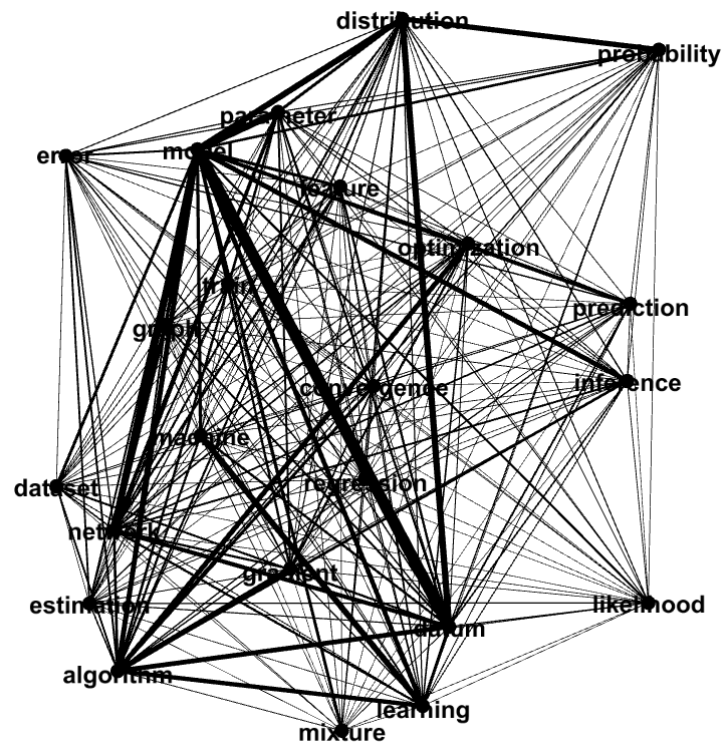
데이터를 불러오면, 아래와 같이 messy한 형태가 보일 것이다.



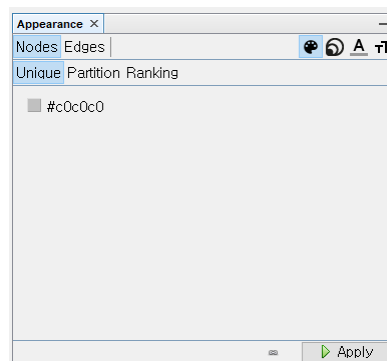
## 2. Overview에서 설정하기

node 관련 설정으로는 node의 라벨, node의 크기, node의 색 조정이 있다.

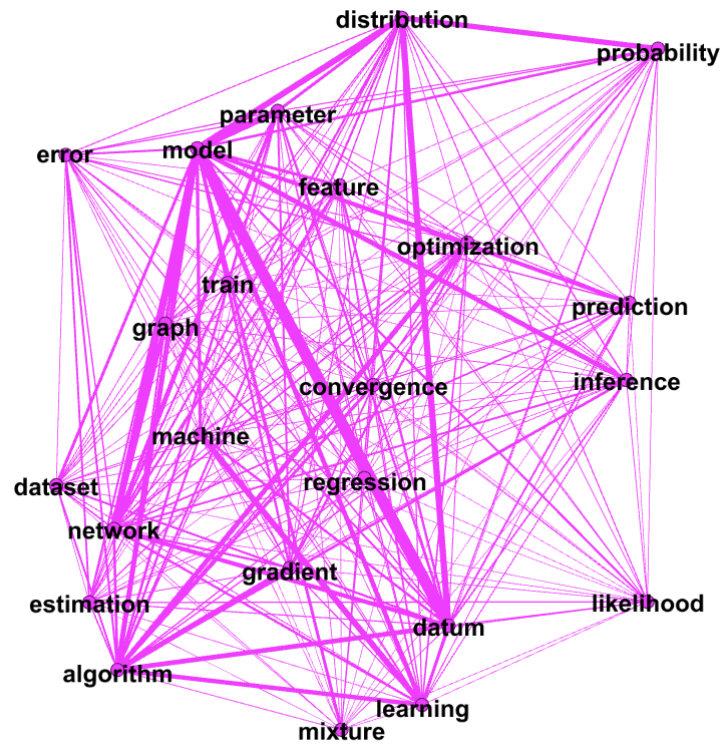
먼저 node의 라벨은 위 그림의 왼쪽 아래 T를 눌러 활성화할 수 있다.



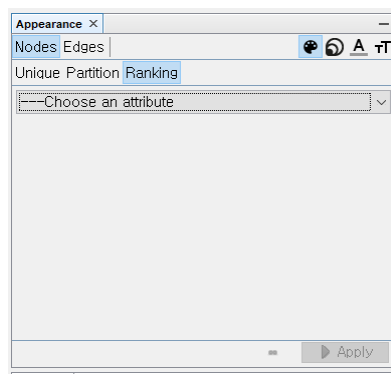
여전히 지저분한데, node와 node의 라벨이 동일한 검정색이기 때문이다. node의 색을 바꿔보자.



위 그림에서 우상단에 네 개의 옵션이 있는데 차례대로 node의 색, node의 크기, label의 색, label의 크기이다. 또한 unique, partition, ranking이 있는데, partition은 아직 파악을 못했고, unique는 모든 node에 동일한 색을 적용하고 ranking은 '중요도'에 따라서 색을 달리 조정하는 것이다. 여기서 '중요도'는 pagerank와 eigenvector centrality가 있는데, 이는 다음 글에서 자세하게 알아본다. 아래는 마음에 드는 색을 고르고 Apply를 누른 결과이다.



위의 그림은 unique 옵션을 적용했을 때의 결과이고, 중요도에 따라 선의 그라데이션을 다르게 하고 싶다면 아래에서 attribute을 선택하면 된다.

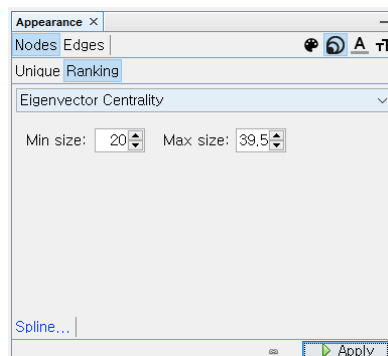


처음에는 Degree 값만 보일텐데, pagerank나 eigenvector centrality을 계산하지 않아서 그렇다. 오른쪽에서 statistics을 누르면 아래와 같이 보일 것이다.

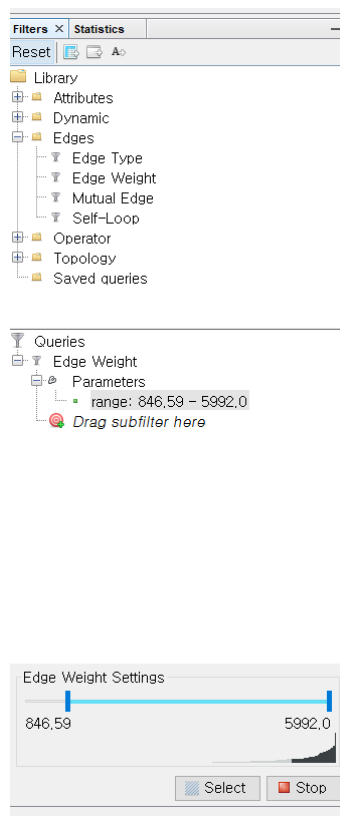


여기서 중간에 pagerank나 eigenvector centrality가 보일 것이다. 또는 다른 값들을 사용해도 된다. 'Run'을 클릭하면 값을 계산해주고 위의 Choose an attribute에서 계산한 지표들이 보일 것이다.

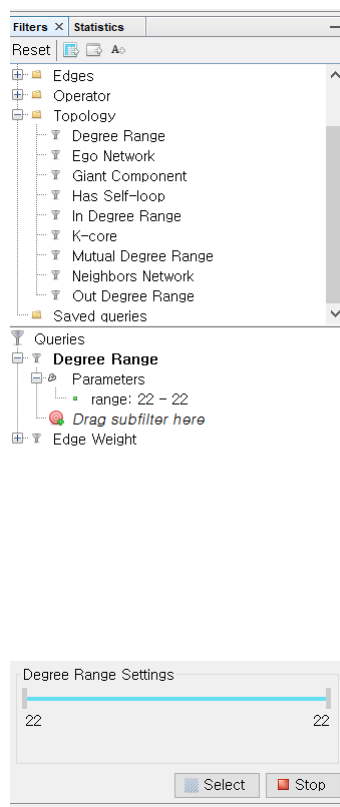
이제 계산한 '중요도'에 비례하게 노드의 크기나 색, 라벨의 크기나 색을 설정하면 된다. Ranking 옵션은 이 넷에 모두 있다. 필자는 node의 크기를 eigenvector centrality에 비례하게 설정하였다.



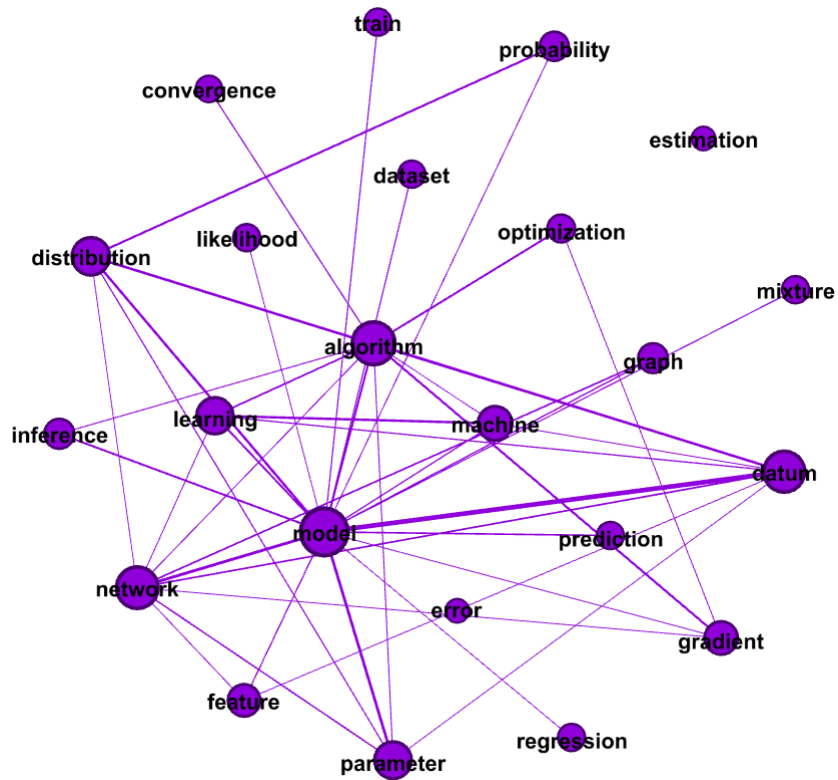
여기서 더 깔끔하게 하고 싶으면, 작은 weight을 가지는 edges를 필터링할 필요가 있다. 이는 아래의 Edge Weight을 Queries로 드래그하여서 스크롤 바를 움직이며 할 수 있다.



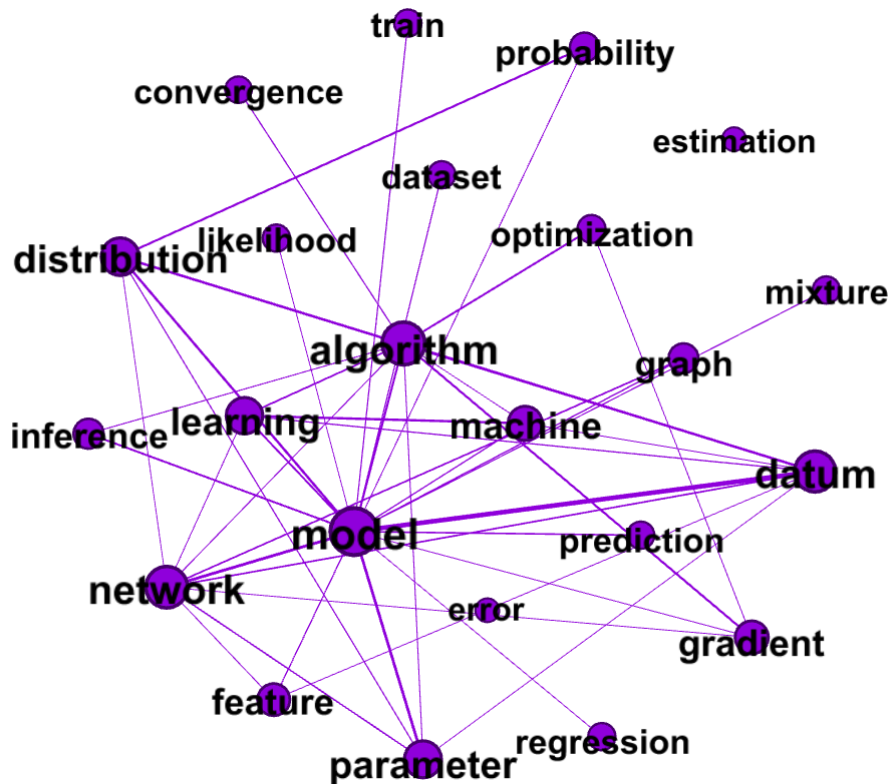
만약 node들을 미리 선별하지 않아서 매우 많은 node들이 있다면, 아래 옵션을 통해서 덜 중요한 node를 필터링할 수 있다.



위에서 언급한 내용을 수행한 결과는 아래와 같다.



노드들의 크기가 서로 상이하고 edges들도 깔끔하게 정리되었다. 이제 라벨들의 크기도 eigenvector centrality에 비례하게 설정해보자.



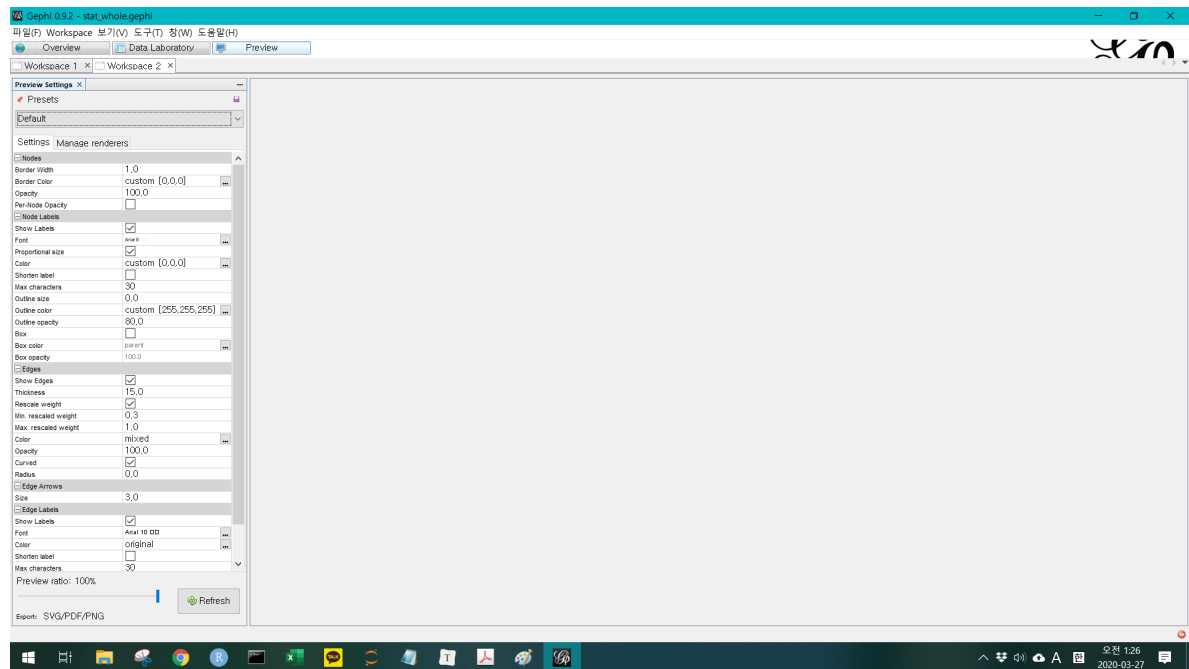
상대적으로 다른 노드들과 많이 연결된 model, algorithm, distribution 등의 단어들이 라벨과 노드의 크기가 크다.

## 여기서 꿀 Tip

node edges를 filtering 하고, eigenvector centrality나 pagerank를 계산하는 것이 좋다. 그렇지 않으면 값이 한 개나 두 개로 나와서 이 값들로 ranking을 하는 것이 의미가 없어진다.

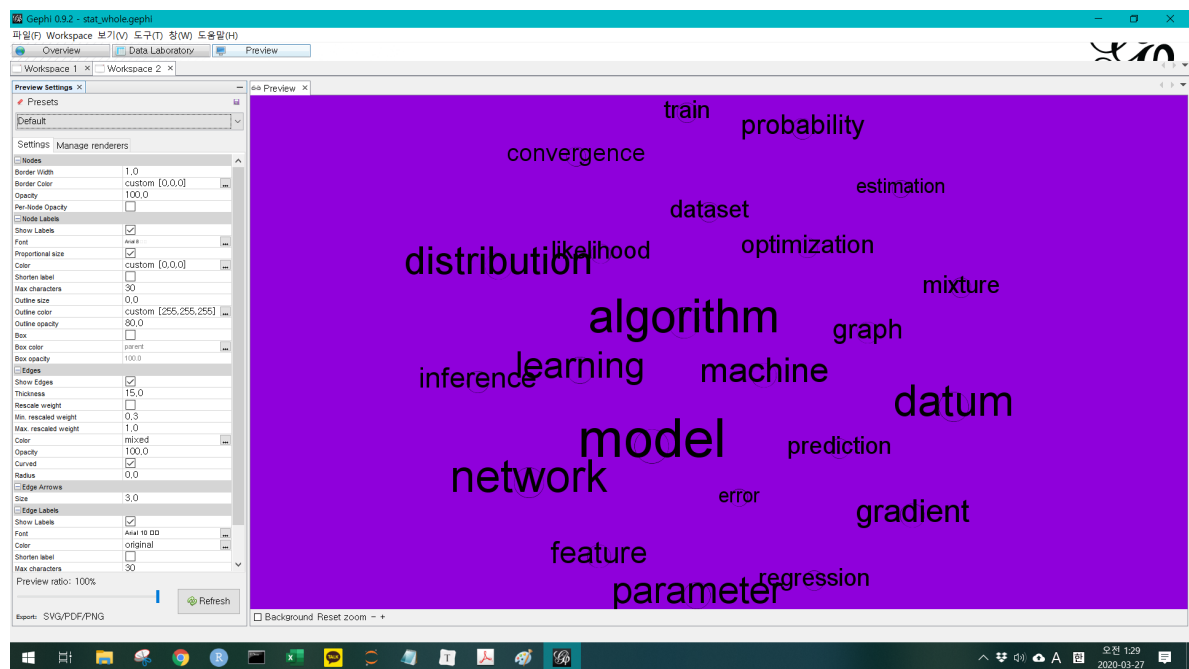
### 3. Preview에서 설정하기

Preview에서 보이는 output이 실제 우리가 얻는 결과물이다. Preview에 들어가서 Refresh를 누르면 되는데, 간혹 Refresh를 눌러도 변화가 전혀 없을 때가 있다.



위와 같이 미동도 없는 경우이다. 이때는 창 -> Preview를 클릭하면 된다.

근데, 상당수는 Refresh를 누르면, 아래와 같이 뜨거나, 아예 검정색 화면일 수 있다.

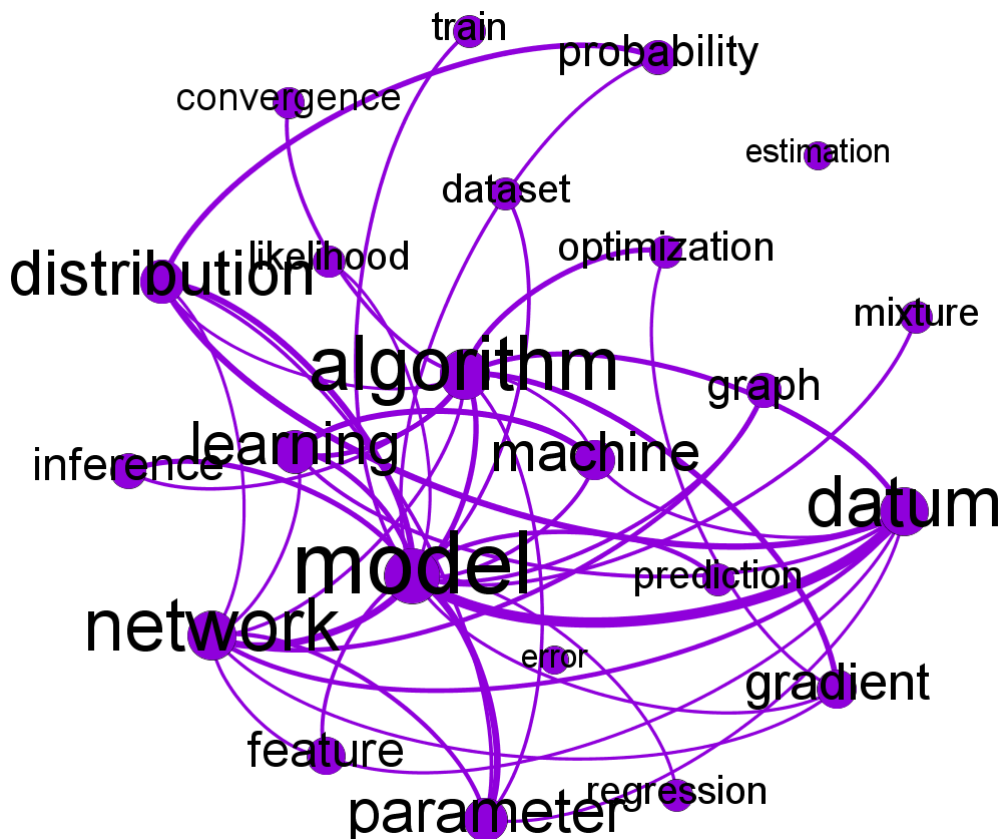


이는, co-occurrence 행렬에서 값들의 차이가 심해서 발생하는 현상이다. 위 그림에서 전체적으로 보라색을 띄는데, 이는 node의 색으로 한 node의 색이 극명하게 커져서 이 node가 전체를 dominant한 것이라고 보면 된다. 이를 위해 아래의 rescale 옵션을 반드시 해야한다.

Edges	
Show Edges	<input checked="" type="checkbox"/>
Thickness	15.0
Rescale weight	<input checked="" type="checkbox"/>
Min. rescaled weight	0.3
Max. rescaled weight	1.0
Color	mixed
Opacity	100.0
Curved	<input checked="" type="checkbox"/>
Radius	0.0

Min rescaled weight이 작을수록 노드간의 크기 차이가 크게 나타나고 Max rescaled weight와 비슷할수록 노드간의 크기가 비슷할 것이다.

Rescale weight을 조정하였다면, edges들의 두께가 조금 얇게 나타날 수도 있다. 이때는 위 그림의 Thickness의 값을 크게 하여 조정하면 된다. 아래 그림은 위 설정을 적용한 결과이다.



중심이 되는 단어의 label과 node가 다른 단어에 비해서 부각 되어있고, edges의 두께도 서로 상이하여 시각화로 써먹을 정도가 되었다.

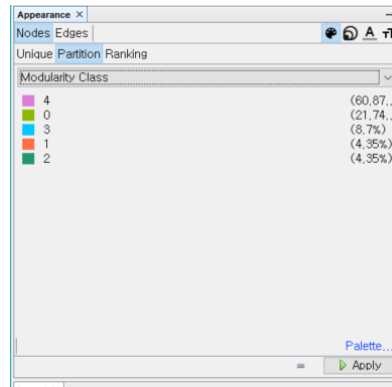
## 원하는 node 별로 다른 edges 색 적용하기

modularity 값을 계산하여, 'Partition'을 적용한다. 우선 modularity는 여기서 값을 계산해줘야 한다.

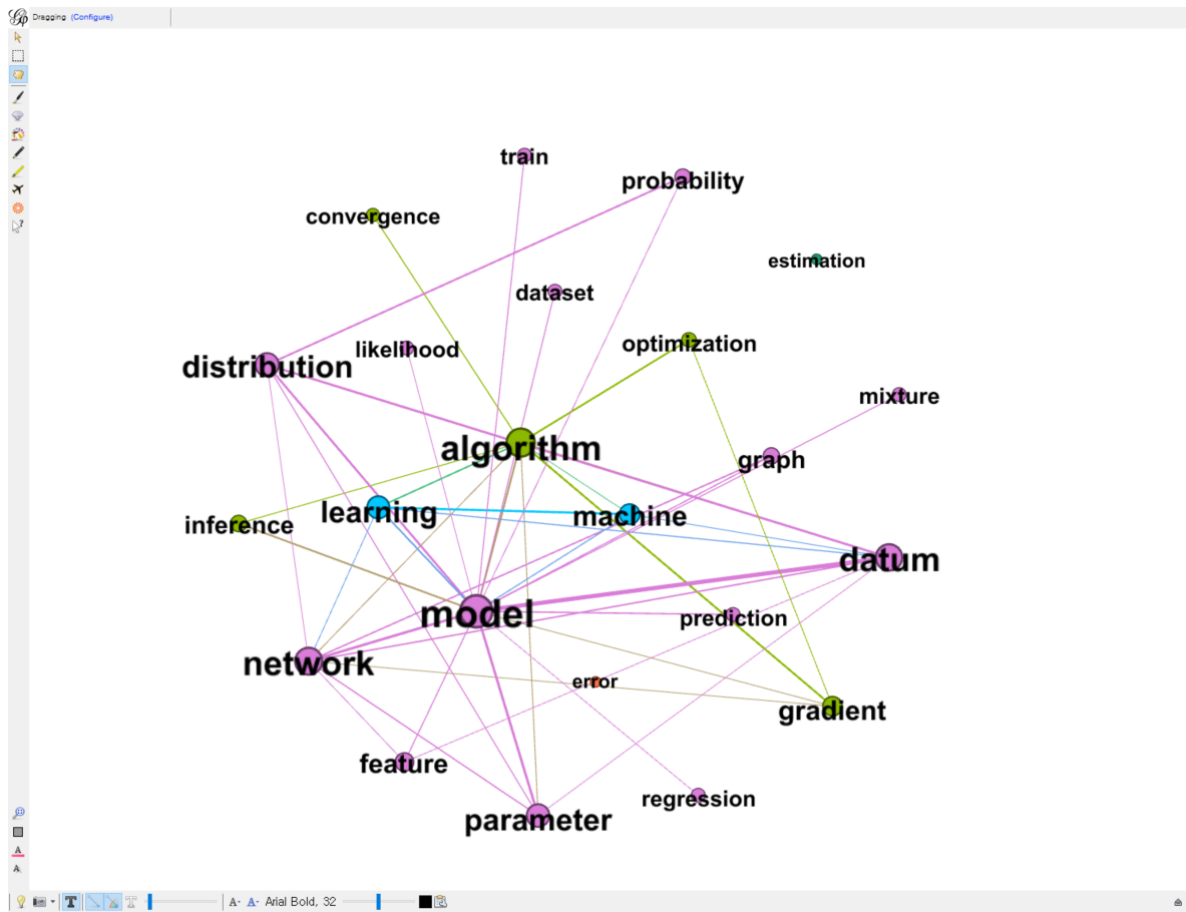
Filters		Statistics X	
Settings			
Network Overview			
Average Degree	22	Run	
Avg. Weighted Degree		Run	
Network Diameter		Run	
Graph Density		Run	
HITS		Run	
Modularity		Run	
PageRank		Run	
Connected Components		Run	
DBSCAN		Run	
Node Overview			
Avg. Clustering Coefficient	1	Run	
Eigenvector Centrality		Run	
Edge Overview			
Avg. Path Length		Run	
Dynamic			
# Nodes		Run	
# Edges		Run	
Degree		Run	
Clustering Coefficient		Run	



이제 Partition에서 아래와 같이 Modularity 옵션이 뜰 것이다.



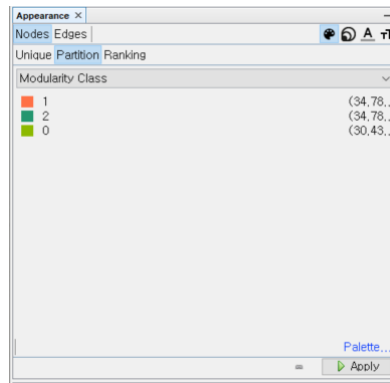
그러면 계산한 modularity에 따라서 색이 임의로 배정된 상태를 확인할 수 있다. 여기서 Apply를 눌러 보면 아래와 같이 modularity에 해당하는 node와 edges의 색을 바꿔준다.



그런데, 여기서 '원하는' node들끼리 같은 색을 취하게 해주고 싶으면, 조금의 수작업이 필요하다. Data Laboratory로 들어가보자.

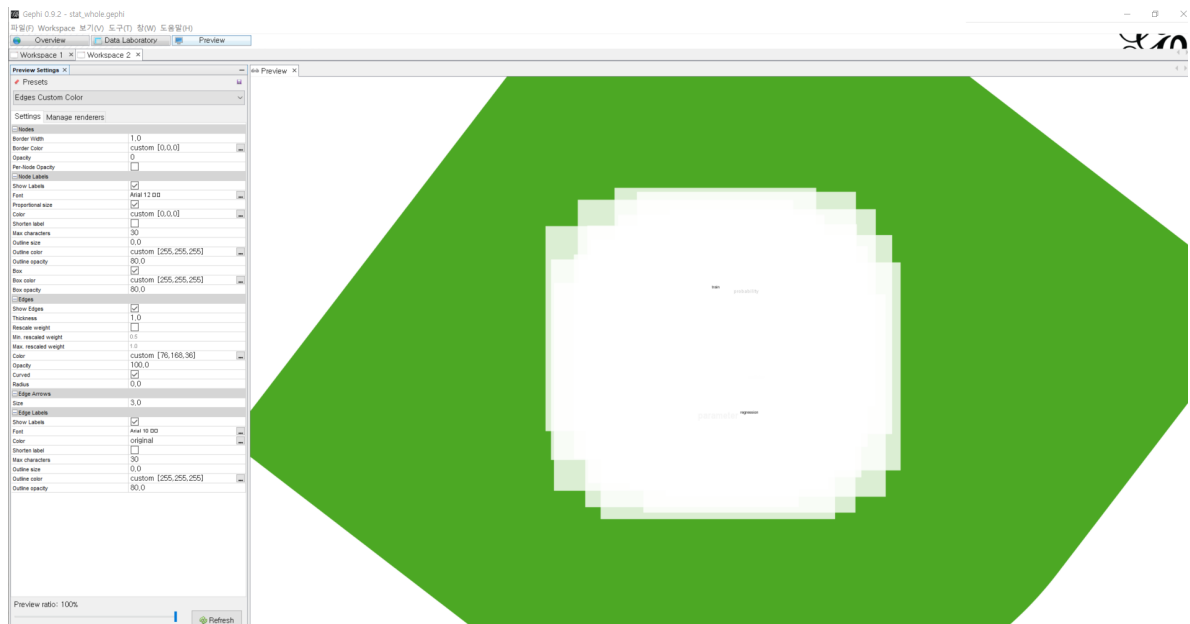
id	Label	Interval	PageRank	Eigenvector Centrality	Clustering Coefficient	Number of Triangles	Degree	Modularity Class
0	algorithm	0.043470	0.813060	0.117724	1.0	231	22	0
1	convergence	0.043470	0.146632	0.146632	1.0	231	22	4
2	dataset	0.043470	0.795315	0.146632	1.0	231	22	4
3	datum	0.043470	0.565258	0.146632	1.0	231	22	4
4	distribution	0.043470	0.0	0.0	1.0	231	22	1
5	error	0.043470	0.0	0.0	1.0	231	22	2
6	estimation	0.043470	0.0	0.0	1.0	231	22	4
7	feature	0.043470	0.361377	0.146632	1.0	231	22	0
8	gradient	0.043470	0.450745	0.146632	1.0	231	22	4
9	graph	0.043470	0.254466	0.146632	1.0	231	22	4
10	inference	0.043470	0.254356	0.146632	1.0	231	22	0
11	learning	0.043470	0.5444	0.146632	1.0	231	22	3
12	likelihood	0.043470	0.146632	0.146632	1.0	231	22	4
13	machine	0.043470	0.448116	0.146632	1.0	231	22	3
14	mixture	0.043470	0.146632	0.146632	1.0	231	22	4
15	model	0.043470	1.0	0.117724	1.0	231	22	4
16	network	0.043470	0.795315	0.146632	1.0	231	22	4
17	optimization	0.043470	0.17635	0.146632	1.0	231	22	0
18	parameter	0.043470	0.565344	0.146632	1.0	231	22	4
19	prediction	0.043470	0.146632	0.146632	1.0	231	22	4
20	probability	0.043470	0.232276	0.146632	1.0	231	22	4
21	regression	0.043470	0.146632	0.146632	1.0	231	22	4
22	train	0.043470	0.146632	0.146632	1.0	231	22	4

우상단에서 Modularity Class를 선택하고, Modularity Class에 해당하는 column을 원하는 클러스터별로 지정해주면 된다. 여기서의 예시를 위해, 위에서부터 0, 1, 2를 지정하겠다. 그리고 Overview로 돌아와서 Partition을 보면, 설정한 0, 1, 2 값으로 업데이트 되어 있는 것을 확인할 수 있다.

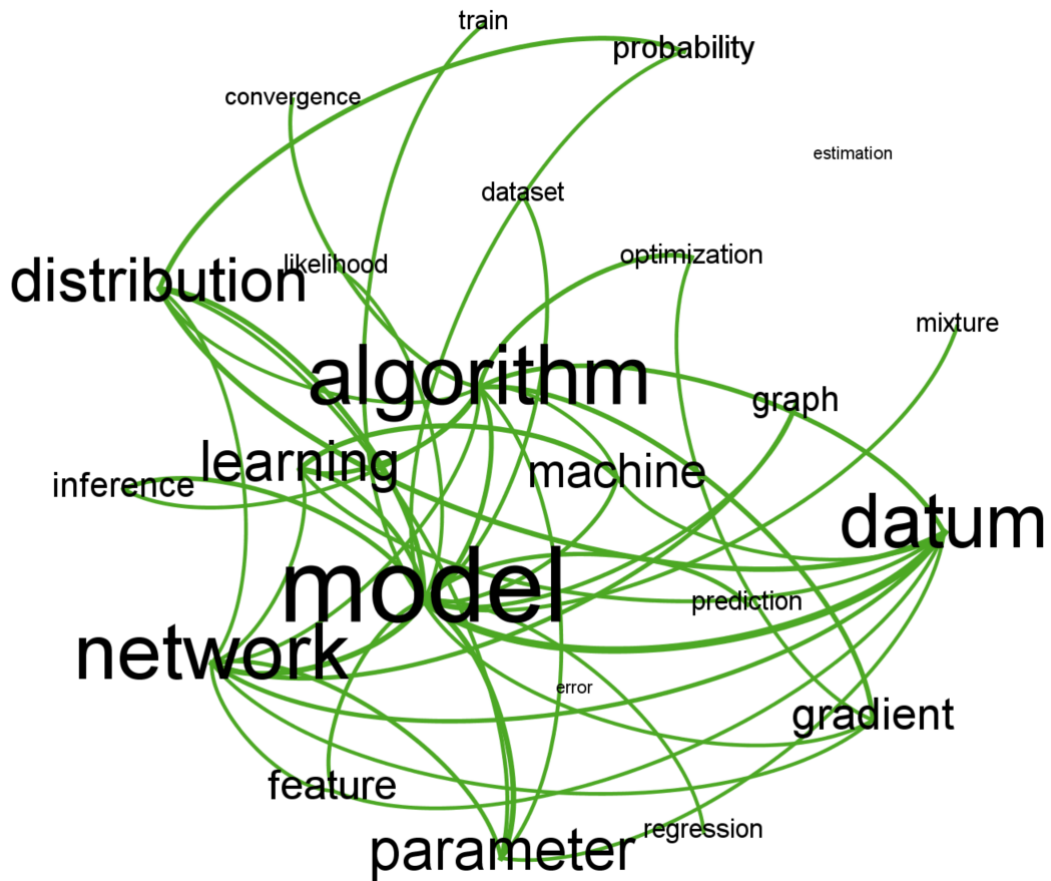


그리고 Apply를 누르면 원하는 그룹이 동일한 색을 지니게 된다.

여기까지 설정하고 Preview를 들어가보자. Final output에서도 modularity 별로 색을 다르게 하고 싶으면, Edges Custom Color를 선택해야 한다. Preview에 들어가보면 역시나 뭔가 이상한 것이 띄워져 있을 것이다. (아닐 수도 있지만, 대부분 그렇다.)

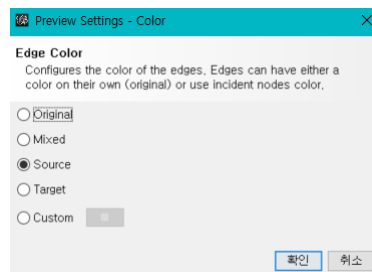


당황하지 말고, Rescale 옵션을 통해서 크기를 조정해주자. 또한 Thickness를 크게 하여 두께도 조절하자.

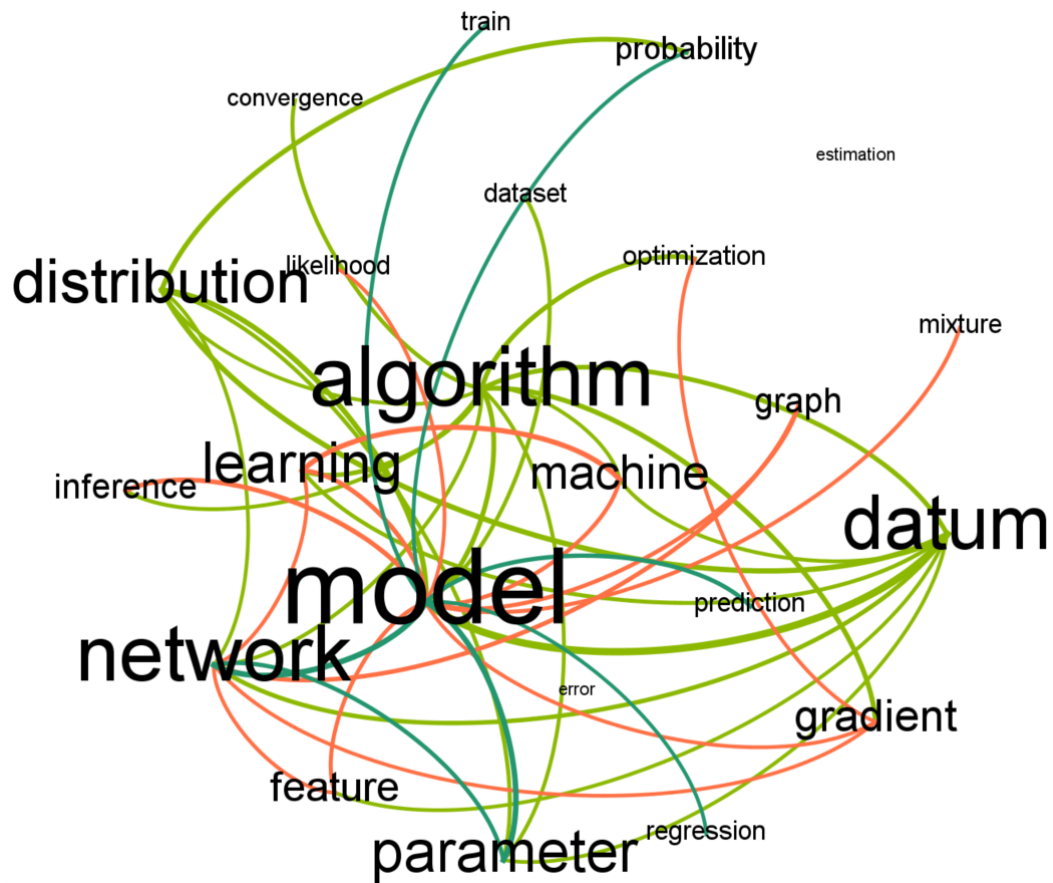


매우 편안해졌다. 하지만 우리가 Overview에서 Modularity에 따라 색을 달리 설정했는데, Preview에서는 그게 반영되어있지 않다. 이를 따로 설정해줘야하는데, Edges의 Color로 들어가보자.

여기서 아래의 Source 옵션을 선택해야 한다.



여기까지 했으면 Preview에서도 Overview와 비슷하게 Modularity에 따라서 색이 다르게 설정된 모습을 확인할 수 있다.



### 꿀팁 하나 더!

아무리 네트워크를 저장했어도 다시 불러오면, edges의 weight 필터링이 안 되어 있고 초기화 되어 있다. 이에 따라서 eigenvector centrality 등으로 ranking한 것도 모두 초기화 되어 있다. 따라서 저장한 네트워크를 다시 불러온다면 edges의 weight를 다시 필터링하고 (오른쪽에서 Filter를 클릭하기만 하면 된다.) eigenvector centrality도 다시 계산하자. 그러면 저장할 당시의 네트워크를 복원할 수 있다.