2020

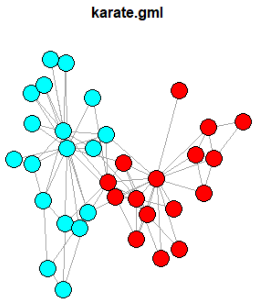
Social Network Analysis

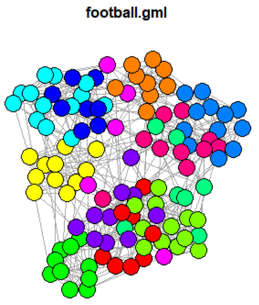
Professor:

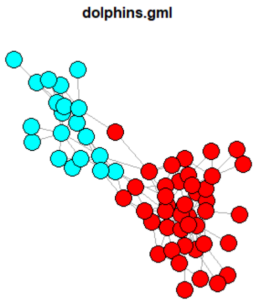
Rushed Kanawati

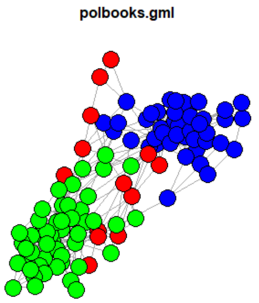
Arjun Singh
Tejas Bhor
Sudhanshu Chaudhari

Table of Contents



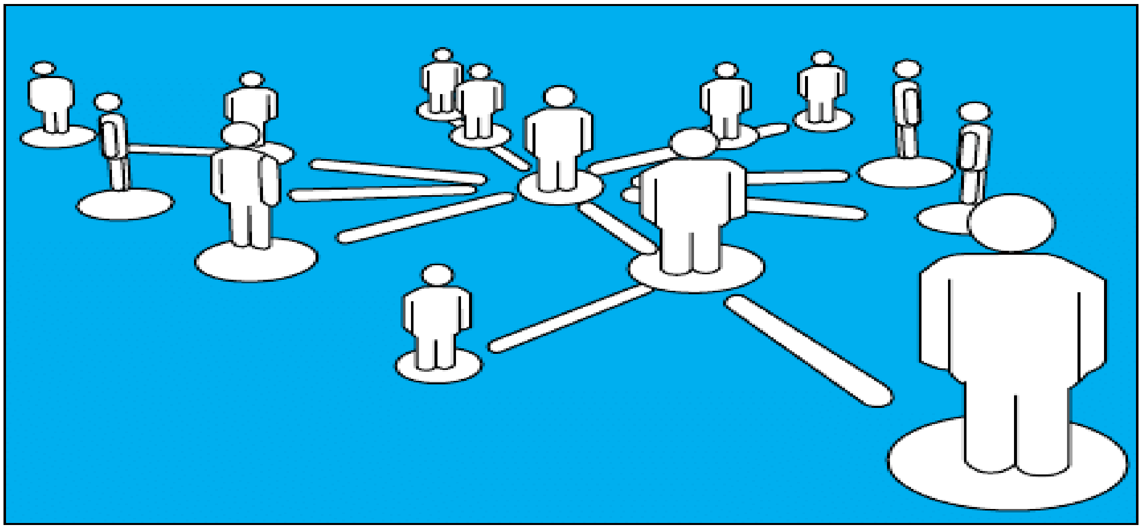






INTRODUCTION

- The defining feature of social network analysis is its focus on the structure of relationships, ranging from casual acquaintance to close bonds.



- Social network analysis assumes that relationships are important. It maps and measures formal and informal relationships to understand what facilitates or impedes the knowledge flows that bind interacting units, viz., who knows whom, and who shares what information and knowledge with whom by what communication media (e.g., data and information, voice, or video communications).
- Social network analysis is a method with increasing application in the social sciences and has been applied in areas as diverse as psychology, health, business organization, and electronic communications.

BENEFITS

- Identify the individuals, teams, and units who play central roles.
- Discern information breakdowns7, bottlenecks8, structural holes, as well as isolated individuals, teams, and units.
- Make out opportunities to accelerate knowledge flows across functional and organizational boundaries.
- Strengthen the efficiency and effectiveness of existing, formal communication channels.
- Raise awareness of and reflection on the importance of informal networks and ways to enhance their organizational performance.
- Advantage peer support.
- Improve innovation and learning.
- Refine strategies.

CENTRALITY MEASURES

Various measures of the centrality of a vertex within a graph determine the relative importance of a vertex within the graph. There are many measures of centrality that are widely used in network analysis: degree centrality, betweenness, closeness and many more.

Degree Centrality: -

- The first, and simplest, is degree centrality. Degree centrality is defined as the number of links incident upon a node (i.e., the number of ties that a node has).
- Degree is often interpreted in terms of the immediate risk of node for catching whatever is flowing through the network

$$\boxed{C_d(v) = \frac{\|(\Gamma(v)\|}{max_{u \in V}\|\Gamma(u)\|}}$$

Betweenness centrality :-

- Betweenness is a centrality measure of a vertex within a graph.
- Vertices that occur on many shortest paths between other vertices have higher betweenness than those that do not.

$$\boxed{C_c(v) = \frac{1}{\sum_{u \in V} sp(v,u)}}$$

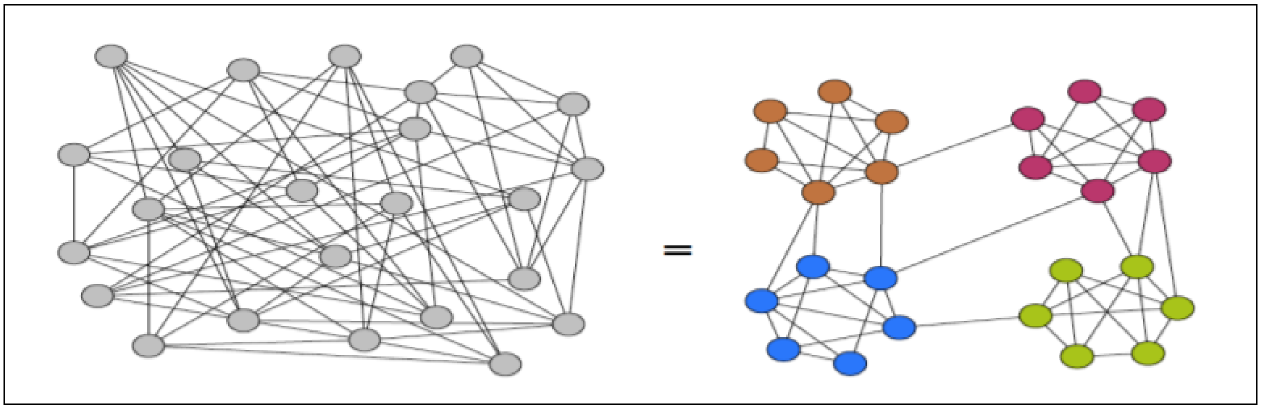
Closeness centrality:-

- In topology and related areas in mathematics, closeness is one of the basic concepts in a topological space. Intuitively we say two sets are close if they are arbitrarily near to each other.
- The concept can be defined naturally in a metric space where a notion of distance between elements of the space is defined, but it can be generalized to topological spaces where we have no concrete way to measure distances.

$$\boxed{C_i(v) = \sum_{s,t \in V, stv} \frac{\sigma_{s,t}(v)}{\sigma_{s,t}}}$$

COMMUNITY DETECTION

- A dense subgraph loosely coupled to other modules in the network.
- A community is a set of nodes seen as one by nodes outside the community. A subgraph where almost all nodes are linked to other nodes in the community.



Local Community Quality Function:

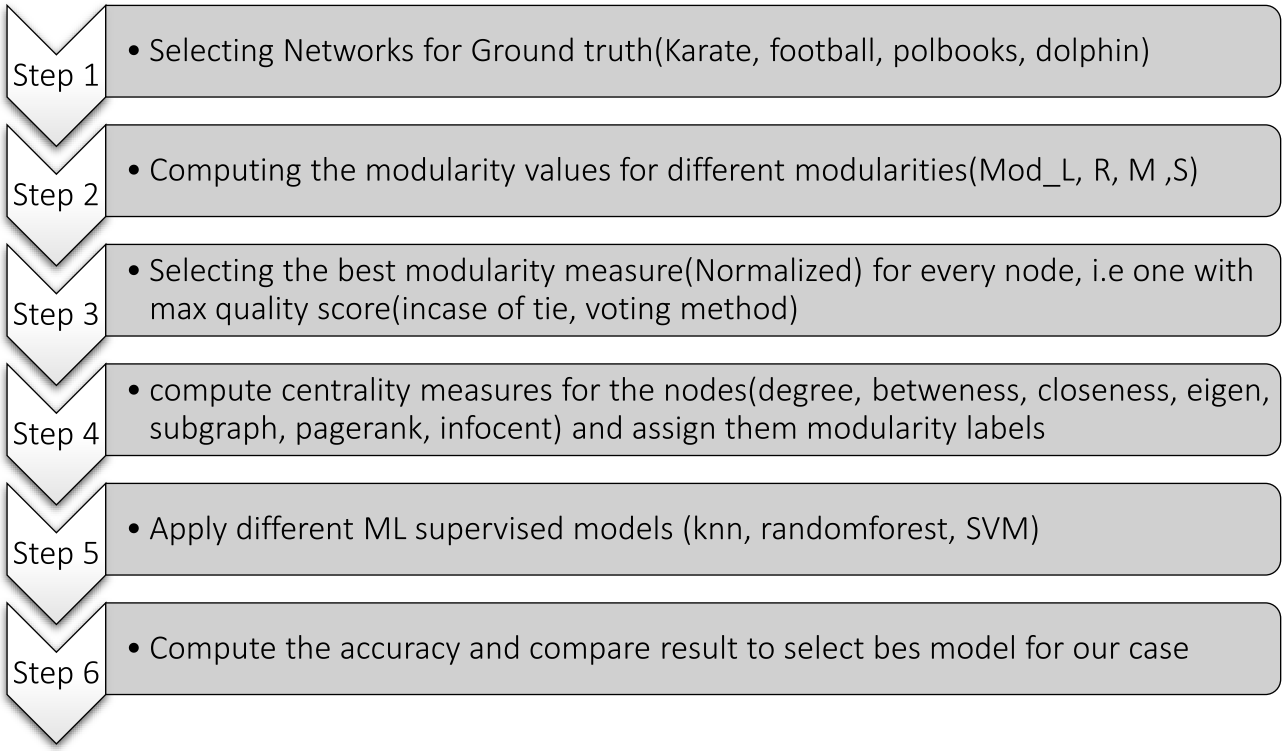
$$R = \frac{B_{in}}{B_{in}+B_{out}}$$

$$M = \frac{D_{in}}{D_{out}}$$

$$L = \frac{L_{in}}{L_{ex}} \text{ where : } L_{in} = \frac{\sum\limits_{i\in D} \|\Gamma(i)\cap D\|}{\|D\|} \text{ , } L_{ex} = \frac{\sum\limits_{i\in B} \|\Gamma(i)\cap S\|}{\|B\|}$$

PROJECT APPROACH

Model Approach



Step 1
- Selecting Networks for Ground truth(Karate, football, polbooks, dolphin)

Step 2
- Computing the modularity values for different modularities(Mod_L, R, M ,S)

Step 3
- Selecting the best modularity measure(Normalized) for every node, i.e one with max quality score(incase of tie, voting method)

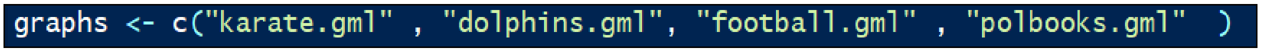
Step 4
- compute centrality measures for the nodes(degree, betweness, closeness, eigen, subgraph, pagerank, infocent) and assign them modularity labels

Step 5
- Apply different ML supervised models (knn, randomforest, SVM)

Step 6
- Compute the accuracy and compare result to select bes model for our case

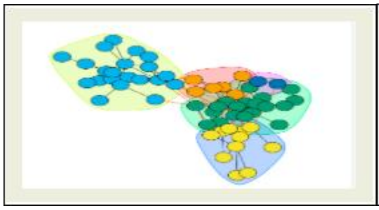
Ground Truth Information

- Benchmark datasets used for ground-truth information.

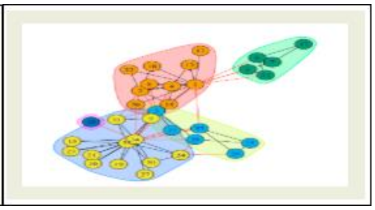


```
graphs <- c("karate.gml" , "dolphins.gml", "football.gml" , "polbooks.gml"  )
```

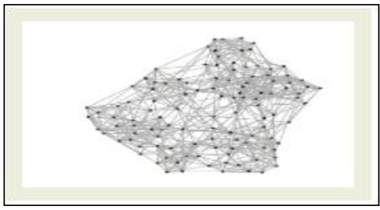
Dolphin



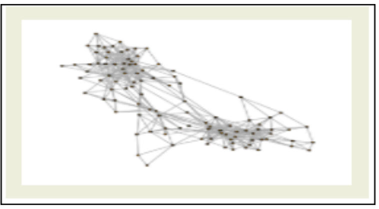
Karate



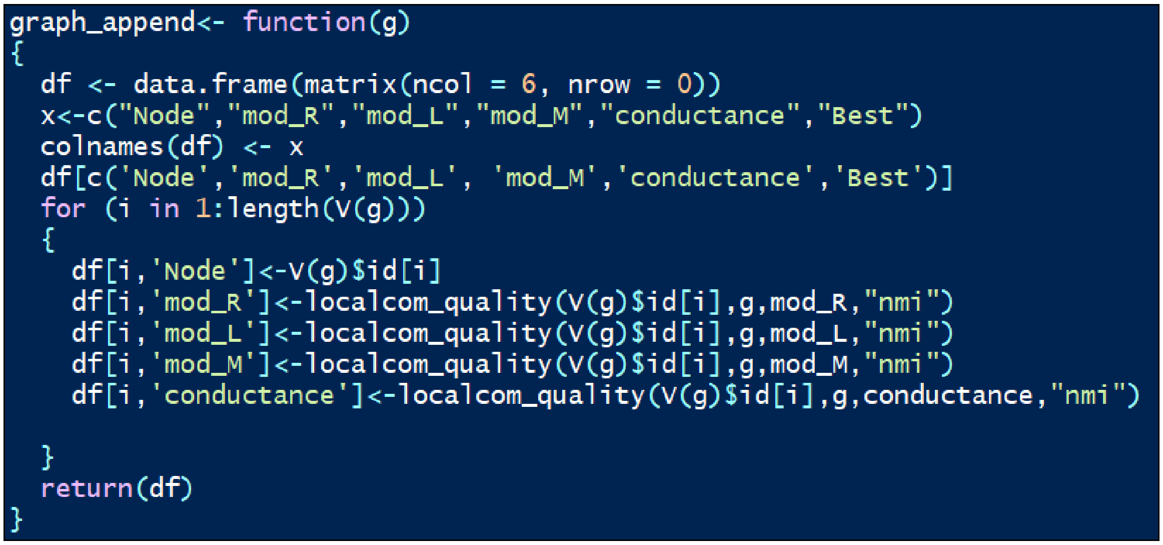
Soccer



Polbooks

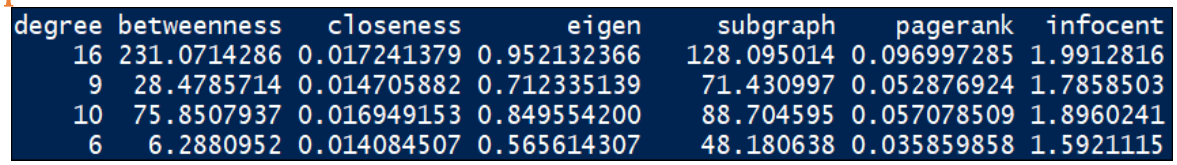


- Each node can be then described by a vector of attributes given its different **centralities** values in the network.



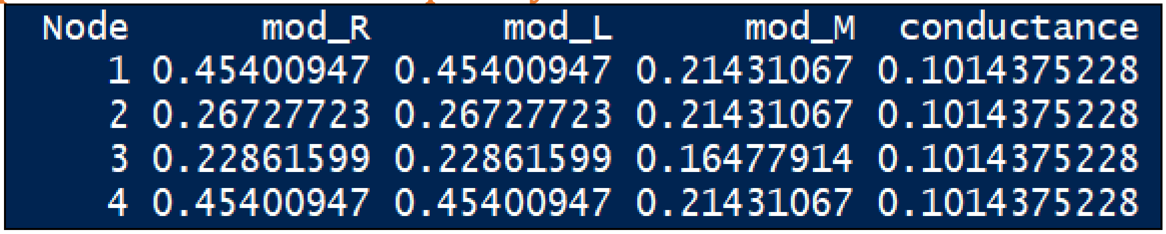
```
graph_append<- function(g)
{
  df <- data.frame(matrix(ncol = 6, nrow = 0))
  x<-c("Node","mod_R","mod_L","mod_M","conductance","Best")
  colnames(df) <- x
  df[c('Node','mod_R','mod_L', 'mod_M','conductance','Best')]
  for (i in 1:length(V(g)))
  {
    df[i,'Node']<-V(g)$id[i]
    df[i,'mod_R']<-localcom_quality(V(g)$id[i],g,mod_R,"nmi")
    df[i,'mod_L']<-localcom_quality(V(g)$id[i],g,mod_L,"nmi")
    df[i,'mod_M']<-localcom_quality(V(g)$id[i],g,mod_M,"nmi")
    df[i,'conductance']<-localcom_quality(V(g)$id[i],g,conductance,"nmi")

  }
  return(df)
}
```

Sample Data frame with centralities:



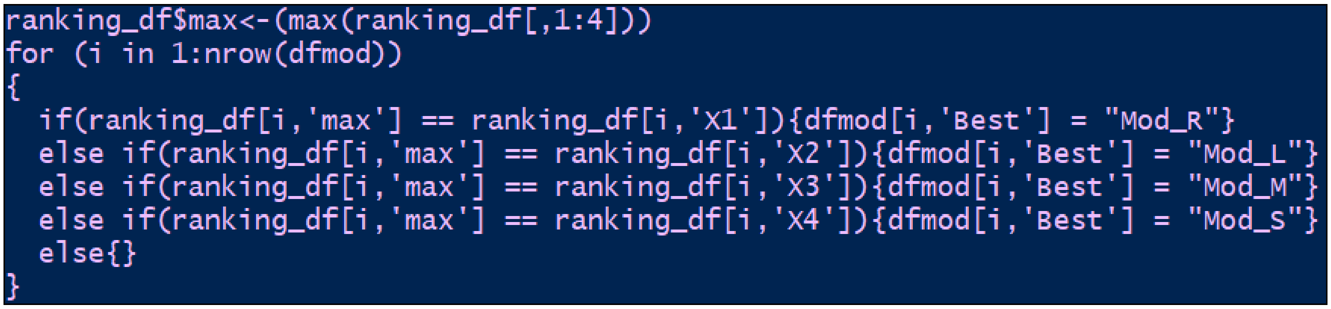
```
degree betweenness    closeness        eigen     subgraph     pagerank  infocent
    16 231.0714286 0.017241379 0.952132366   128.095014 0.096997285 1.9912816
     9  28.4785714 0.014705882 0.712335139    71.430997 0.052876924 1.7858503
    10  75.8507937 0.016949153 0.849554200    88.704595 0.057078509 1.8960241
     6   6.2880952 0.014084507 0.565614307    48.180638 0.035859858 1.5921115
```

Sample Data frame with quality function:

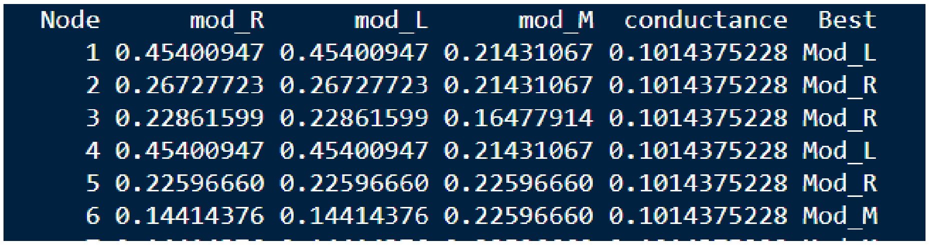


```
  Node      mod_R      mod_L      mod_M  conductance
     1 0.45400947 0.45400947 0.21431067 0.1014375228
     2 0.26727723 0.26727723 0.21431067 0.1014375228
     3 0.22861599 0.22861599 0.16477914 0.1014375228
     4 0.45400947 0.45400947 0.21431067 0.1014375228
```

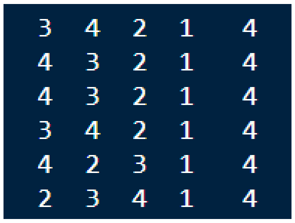
- We then compute for each node its local community applying different quality functions **(L/R/M).** Using the ground truth information, we can readily select the best quality function that yields the best result. For this ranking was one logic that needs a lot to be pondered upon. We had quite a lot of observation of modularity's where two modularity measures were same i.e. it is a tie. To break tied we ranked the modularity vector for every node and when tied applied randomness, probability of 0.5 that anyone can be winner and henceforth we used this logic to compute best modularity measure for every node when tied modularity's.



```
ranking_df$max<-(max(ranking_df[,1:4]))
for (i in 1:nrow(dfmod))
{
  if(ranking_df[i,'max'] == ranking_df[i,'X1']){dfmod[i,'Best'] = "Mod_R"}
  else if(ranking_df[i,'max'] == ranking_df[i,'X2']){dfmod[i,'Best'] = "Mod_L"}
  else if(ranking_df[i,'max'] == ranking_df[i,'X3']){dfmod[i,'Best'] = "Mod_M"}
  else if(ranking_df[i,'max'] == ranking_df[i,'X4']){dfmod[i,'Best'] = "Mod_S"}
  else{}
}
```

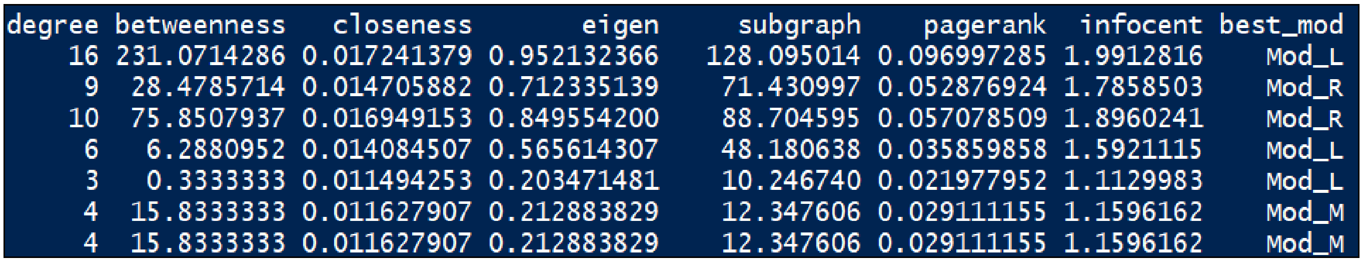


```
  Node       mod_R      mod_L      mod_M  conductance  Best
     1 0.45400947 0.45400947 0.21431067 0.1014375228 Mod_L
     2 0.26727723 0.26727723 0.21431067 0.1014375228 Mod_R
     3 0.22861599 0.22861599 0.16477914 0.1014375228 Mod_R
     4 0.45400947 0.45400947 0.21431067 0.1014375228 Mod_L
     5 0.22596660 0.22596660 0.22596660 0.1014375228 Mod_R
     6 0.14414376 0.14414376 0.22596660 0.1014375228 Mod_M
```



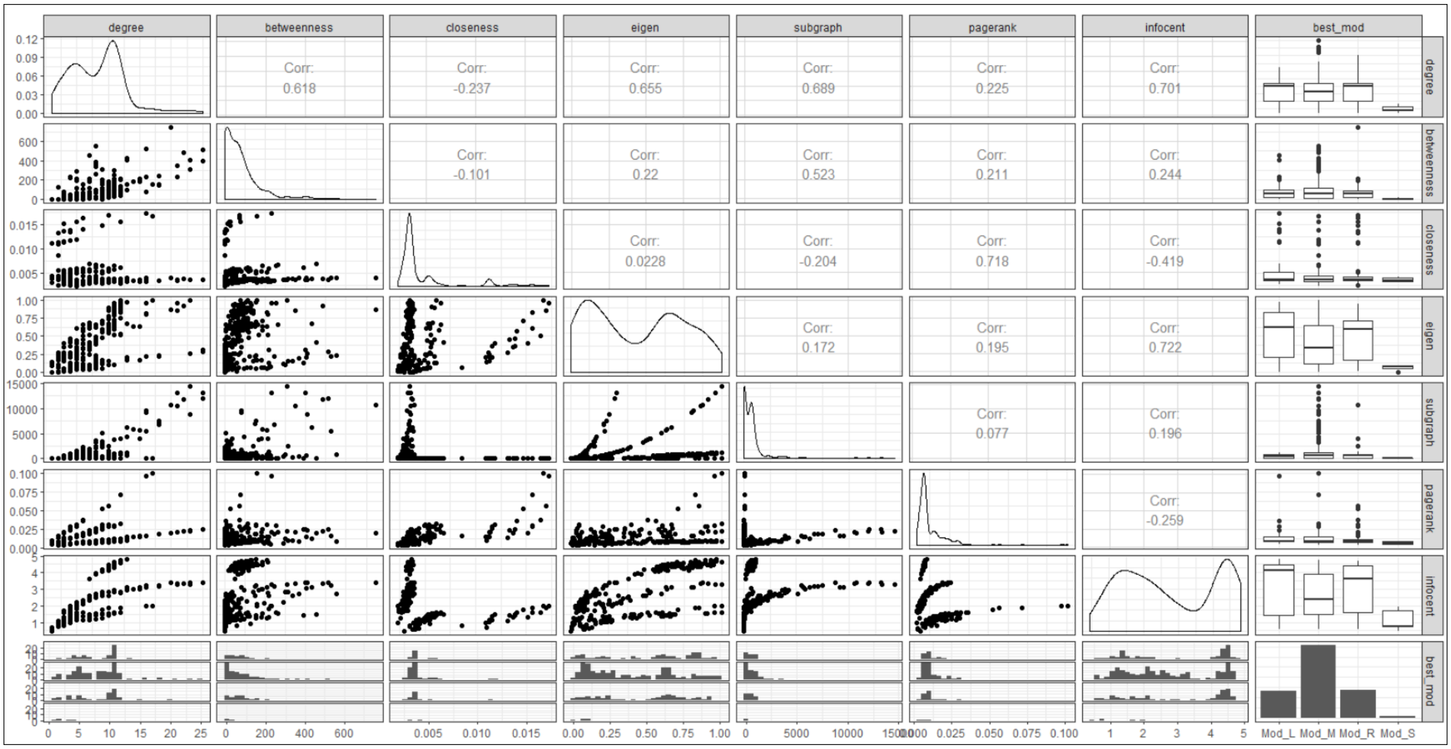
```
 3  4  2  1   4
 4  3  2  1   4
 4  3  2  1   4
 3  4  2  1   4
 4  2  3  1   4
 2  3  4  1   4
```

Sample DataFrame with best quality function.



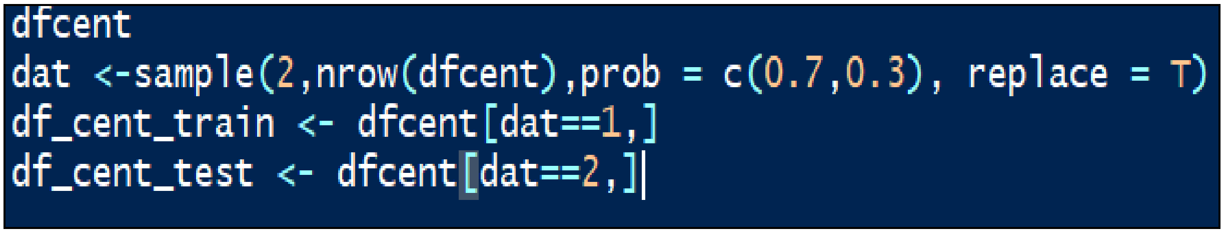
```
degree betweenness   closeness        eigen     subgraph    pagerank  infocent best_mod
    16 231.0714286 0.017241379 0.952132366   128.095014 0.096997285 1.9912816    Mod_L
     9  28.4785714 0.014705882 0.712335139    71.430997 0.052876924 1.7858503    Mod_R
    10  75.8507937 0.016949153 0.849554200    88.704595 0.057078509 1.8960241    Mod_R
     6   6.2880952 0.014084507 0.565614307    48.180638 0.035859858 1.5921115    Mod_L
     3   0.3333333 0.011494253 0.203471481    10.246740 0.021977952 1.1129983    Mod_L
     4  15.8333333 0.011627907 0.212883829    12.347606 0.029111155 1.1596162    Mod_M
     4  15.8333333 0.011627907 0.212883829    12.347606 0.029111155 1.1596162    Mod_M
```

Correlation between centralities: -



Multi-label supervised classification

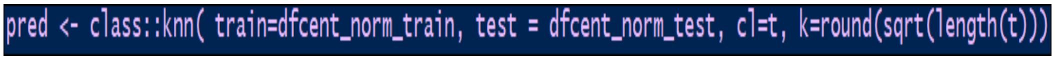
- We now have the required ground-truth information from the benchmark networks.
- We find solution to problem of selecting the best quality function to apply by reformulating it as multi-label supervised classification problem.
- **Splitting Data: -** To begin with classification, we first split the data in to Train and Test with a ratio of 7/3.



```
dfcent
dat <-sample(2,nrow(dfcent),prob = c(0.7,0.3), replace = T)
df_cent_train <- dfcent[dat==1,]
df_cent_test <- dfcent[dat==2,]
```

1. K-Nearest Neighbors

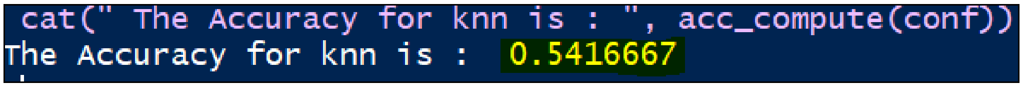
- **Creating and training KNN model:**



```
pred <- class::knn( train=dfcent_norm_train, test = dfcent_norm_test, cl=t, k=round(sqrt(length(t)))
```

- **Testing the model:**

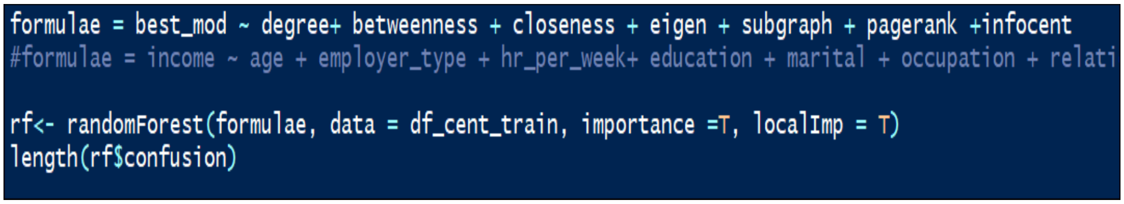
Accuracy for Random Forest is approx. 54.1%



```
 cat(" The Accuracy for knn is : ", acc_compute(conf))
The Accuracy for knn is :  0.5416667
```

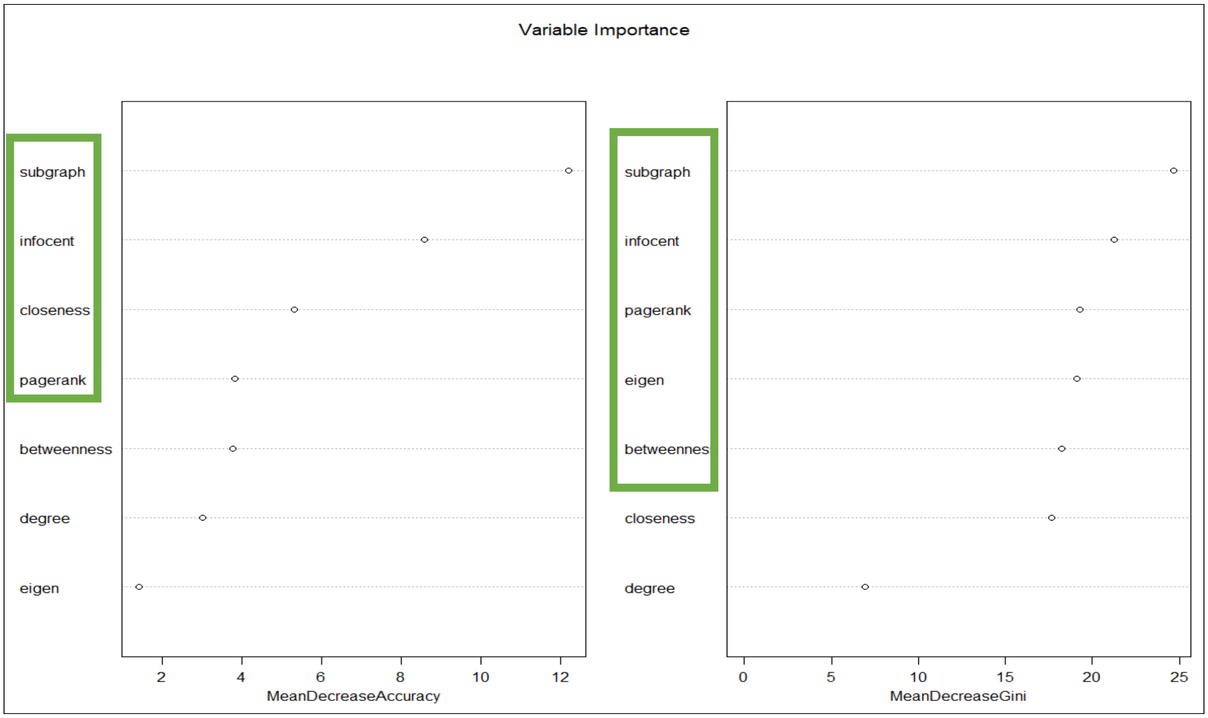
2. Random Forest

- **Creating and training Random Forest model.**

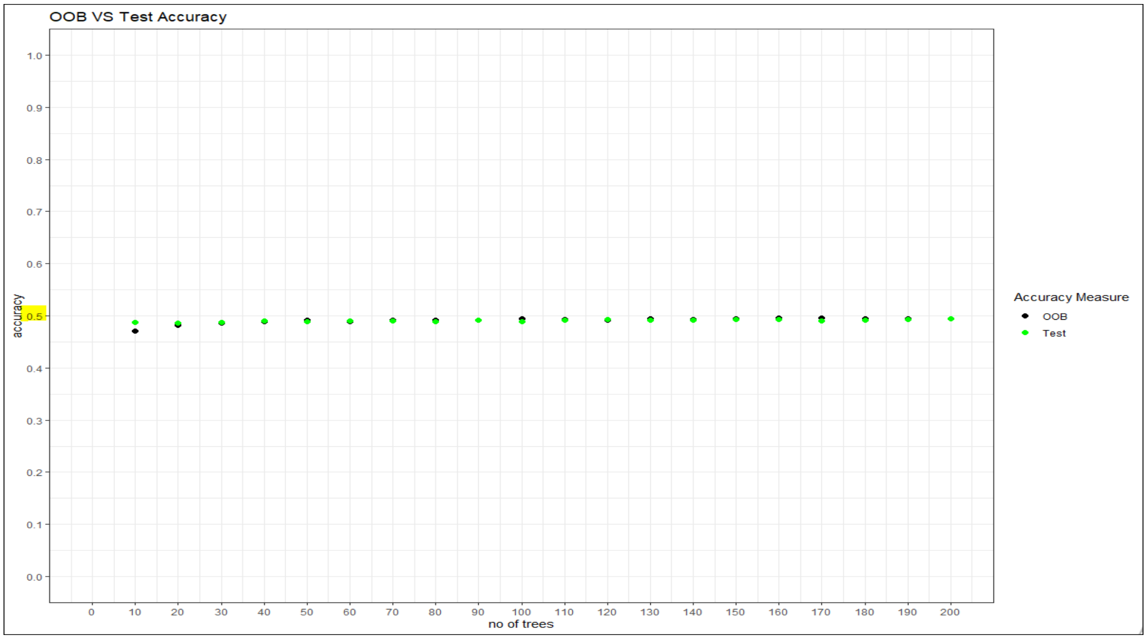


```
formulae = best_mod ~ degree+ betweenness + closeness + eigen + subgraph + pagerank +infocent
#formulae = income ~ age + employer_type + hr_per_week+ education + marital + occupation + relati

rf<- randomForest(formulae, data = df_cent_train, importance =T, localImp = T)
length(rf$confusion)
```

- **Testing the model:** As per the variable importance test, the below highlighted centralities play important role in our classification

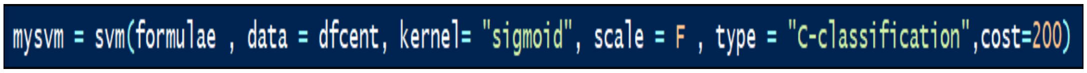


Accuracy for Random Forest is approx. 50% **(for 10-200 trees and mtry = 7)**



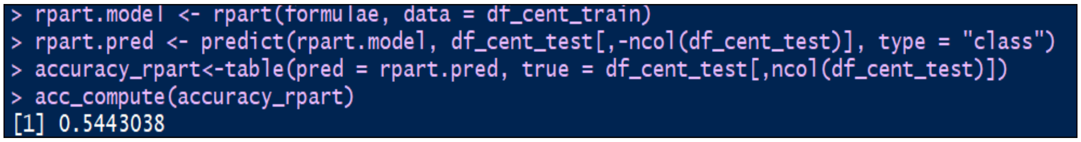
3. Support Vector Machine

- **Creating and training SVM model.**



```
mysvm = svm(formulae , data = dfcent, kernel= "sigmoid", scale = F , type = "C-classification",cost=200)
```

- **Testing the SVM**



```
> rpart.model <- rpart(formulae, data = df_cent_train)
> rpart.pred <- predict(rpart.model, df_cent_test[,-ncol(df_cent_test)], type = "class")
> accuracy_rpart<-table(pred = rpart.pred, true = df_cent_test[,ncol(df_cent_test)])
> acc_compute(accuracy_rpart)
[1] 0.5443038
```

Accuracy for SVM is approx. 54%

CONCLUSION

The Results of model yield different and not very good results. Corrective measures need to address two major issues.

1. The ranking method in case of a tied modularity measure e.g. node1 M1=2, M2=2, M3=1 in this case we assigned random winner the full points. We need to figure out the importance of a modularity measure before selecting this.
2. For random forest (or decision trees), we need levels in the parameters. At present we are using numerical values to create trees as such we have a lot of rules and accuracy is suffering if the new test data doesn't fit into rules generated by the model.
3. This approach is a very good way to identify local community just based on centrality values , instead of looking for the modularity's and figuring out the best(building ground truth every time) a predictive modelling to fit best modularity based in centrality values