PREDICTIVE ANALYTICS USING NETWORKED DATA IN R

Extract a dataset

María Óskarsdóttir, Ph.D.
Post-doctoral researcher

Getting the dataset

```
V(g)$degree<-degree(g)
V(g)$triangles<-count_triangles(g)
V(g)$betweeness<-betweenness(g,normalized=TRUE)
V(g)$transitivity<-transitivity(g,type='local',isolates='zero')

A <- get.adjacency(g)
preference <- c(1,1,1,1,1,1,0,0,0,0)
age <- c(23,65,33,36,28,45,41,24,38,39)
V(g)$rNeighbors <- as.vector(A%*%preference)
V(g)$averageAge <- as.vector(A%*%age/V(g)$degree)

V(g)$pageRank<-page.rank(g)$vector
V(g)$personalizePageRank<-page.rank(g,
  personalized = c(1,0,0,0,0,0,0,0,0,0))$vector
```

```
g
IGRAPH UN-- 10 19 --
 attr: name (v/c), degree (v/n), triangles (v/n), transitivity
| (v/n), rNeighbors (v/n), averageAge (v/n), pageRank (v/n),
| pPageRank (v/n), label (e/c)
 edges (vertex names):
 [1] A--B A--C A--D A--E B--C B--D C--D C--G D--E D--F D--G E--F F--G F--I
[15] G--I G--H H--I H--J I--J
```

Getting the dataset

```
g
IGRAPH UN-- 10 19 --
 attr: name (v/c), degree (v/n), triangles (v/n), transitivity
| (v/n), rNeighbors (v/n), averageAge (v/n), pageRank (v/n),
| pPageRank (v/n), label (e/c)
 edges (vertex names):
 [1] A--B A--C A--D A--E B--C B--D C--D C--G D--E D--F D--G E--F F--G F--I
[15] G--I G--H H--I H--J I--J
```

```
as_data_frame(g,what='vertices')

  name degree triangles transitivity rNeighbors averageAge   pageRank  pPageRank
A    A      4         4    0.6666667          4   40.50000 0.10238312 0.25528911
B    B      3         3    1.0000000          3   30.66667 0.07917232 0.10363533
C    C      4         4    0.6666667          3   41.25000 0.10164910 0.12156935
D    D      6         7    0.4666667          5   39.16667 0.14693274 0.16625582
E    E      3         2    0.6666667          3   34.66667 0.07953551 0.09366836
F    F      4         3    0.5000000          2   35.75000 0.10335821 0.07466596
G    G      5         4    0.4000000          3   35.20000 0.12732387 0.08473039
H    H      3         2    0.6666667          0   39.33333 0.08675903 0.03285162
I    I      4         3    0.5000000          1   37.25000 0.10994175 0.04785657
J    J      2         1    1.0000000          0   31.00000 0.06294435 0.01947748
```

Preprocessing - Missing values

name	degree	triangles	transitivity	rNeighbors	averageAge	pageRank	pPageRank
A	4	4	0.6666667	4	40.50000	0.10238312	0.25528911
B	3	3	1.0000000	3	30.66667	0.07917232	0.10363533
C	NA	4	0.6666667	3	41.25000	0.10164910	0.12156935
D	6	7	0.4666667	5	39.16667	0.14693274	0.16625582
E	3	2	0.6666667	3	34.66667	0.07953551	0.09366836
F	4	3	0.5000000	2	35.75000	0.10335821	0.07466596
G	5	4	0.4000000	3	35.20000	0.12732387	0.08473039
H	3	2	0.6666667	0	39.33333	0.08675903	0.03285162
I	NA	3	0.5000000	1	37.25000	0.10994175	0.04785657
J	2	1	1.0000000	0	31.00000	0.06294435	0.01947748

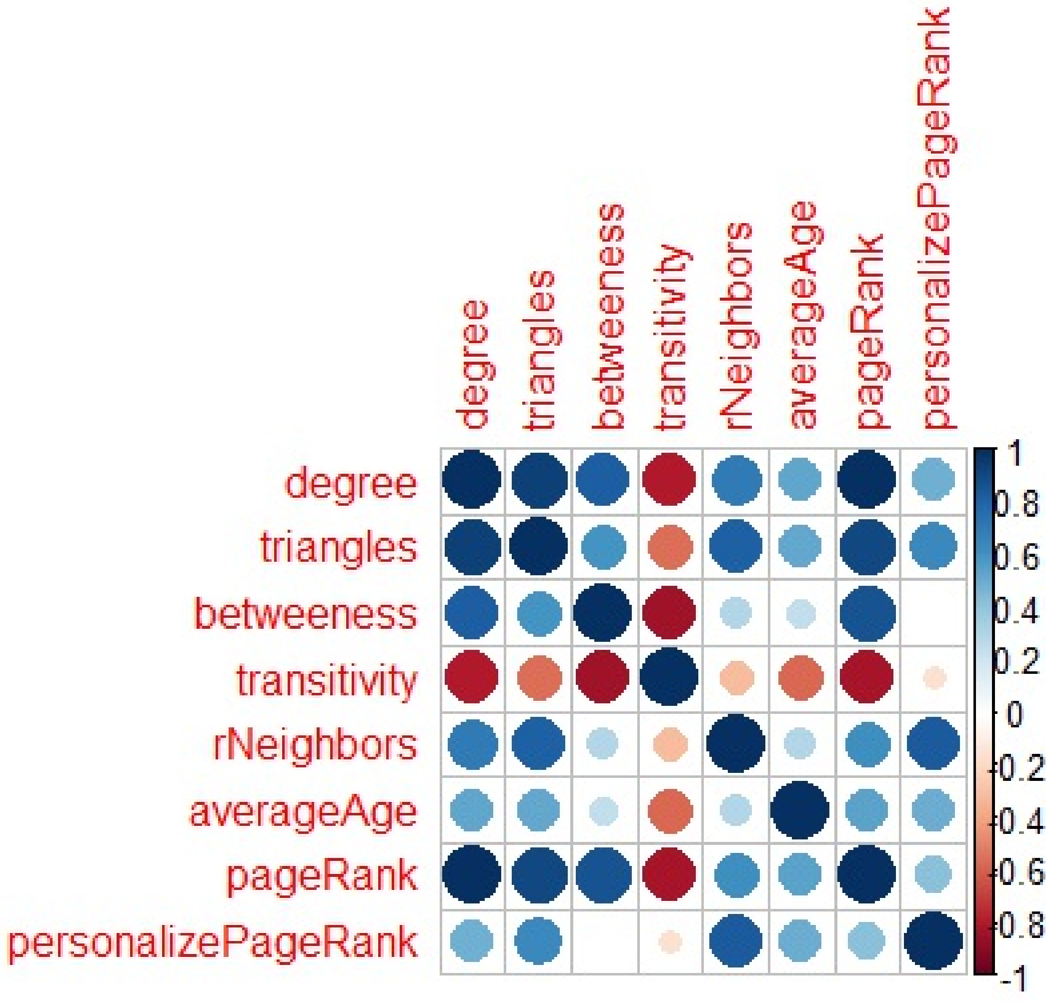
```
sum(is.na(dataset$degree))
[1] 2
```

Preprocessing - Correlated variables

```
library(corrplot)

M <- cor(dataset[,-1])

corrplot(M, method = 'circle')
```



PREDICTIVE ANALYTICS USING NETWORKED DATA IN R

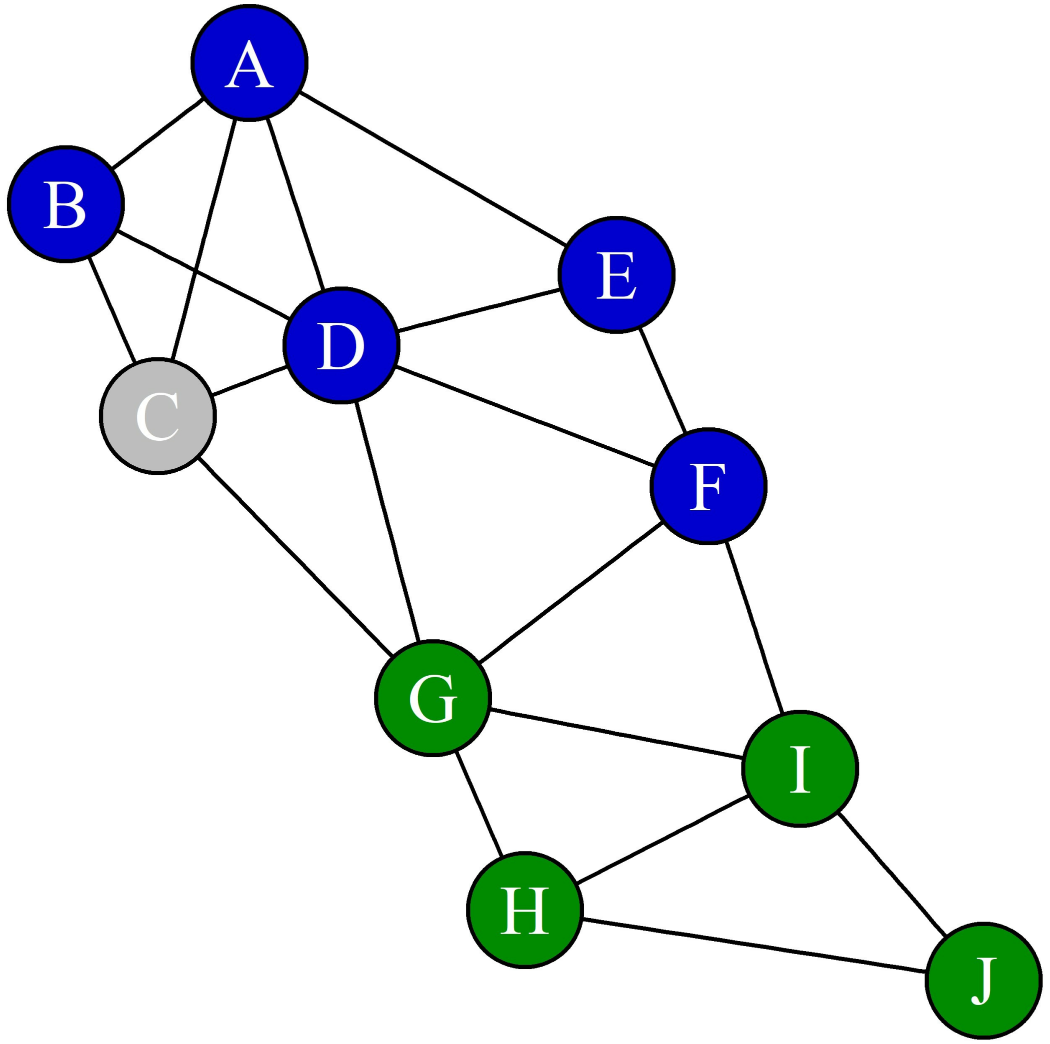
Let's practice!

PREDICTIVE ANALYTICS USING NETWORKED DATA IN R

Building a predictive model

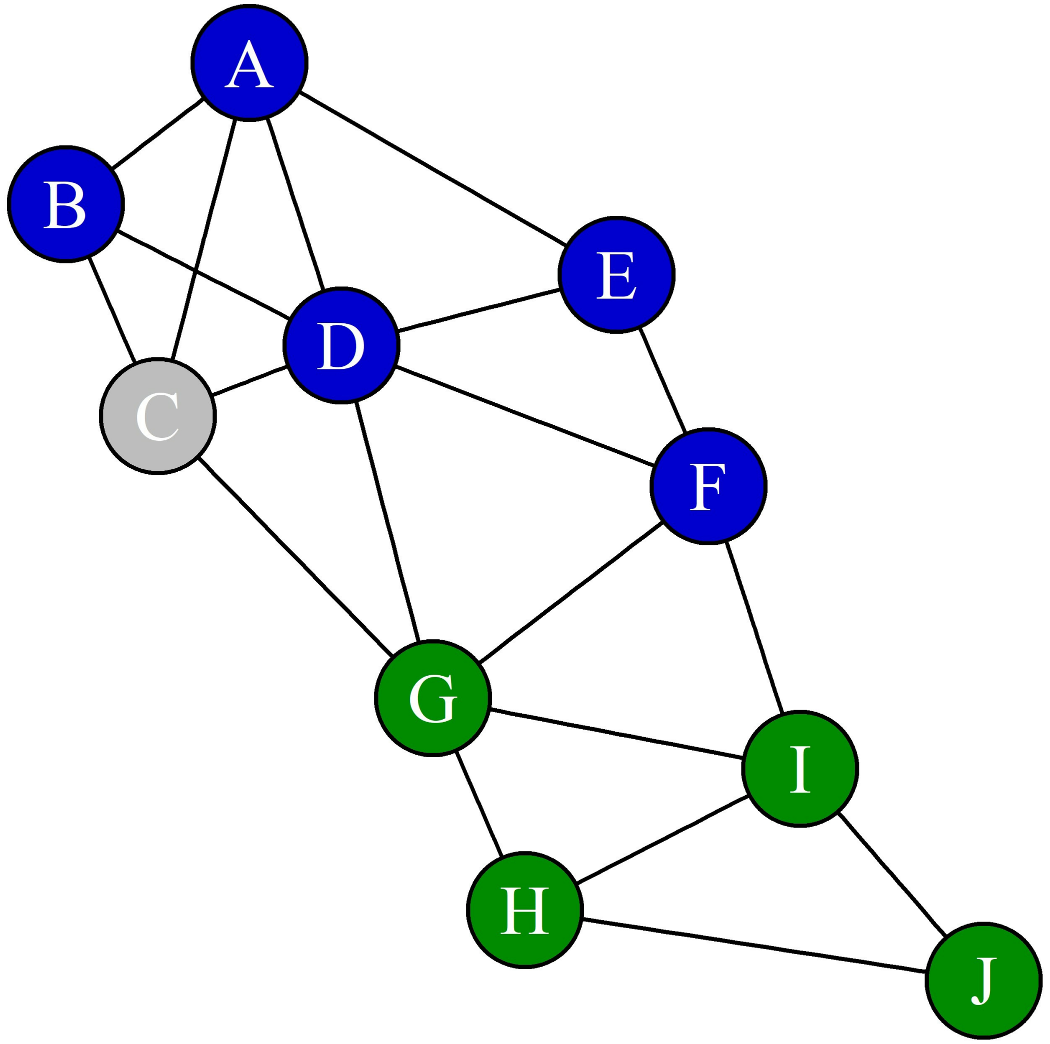
María Óskarsdóttir, Ph.D.
Post-doctoral researcher

Predictive modeling




```
dataset$preference<-c(rep('R',2),'?',
rep('R',3),rep('P',4))
> dataset[,c(1,9)]
  name preference
A    A          R
B    B          R
C    C          ?
D    D          R
E    E          R
F    F          R
G    G          P
H    H          P
I    I          P
J    J          P
```

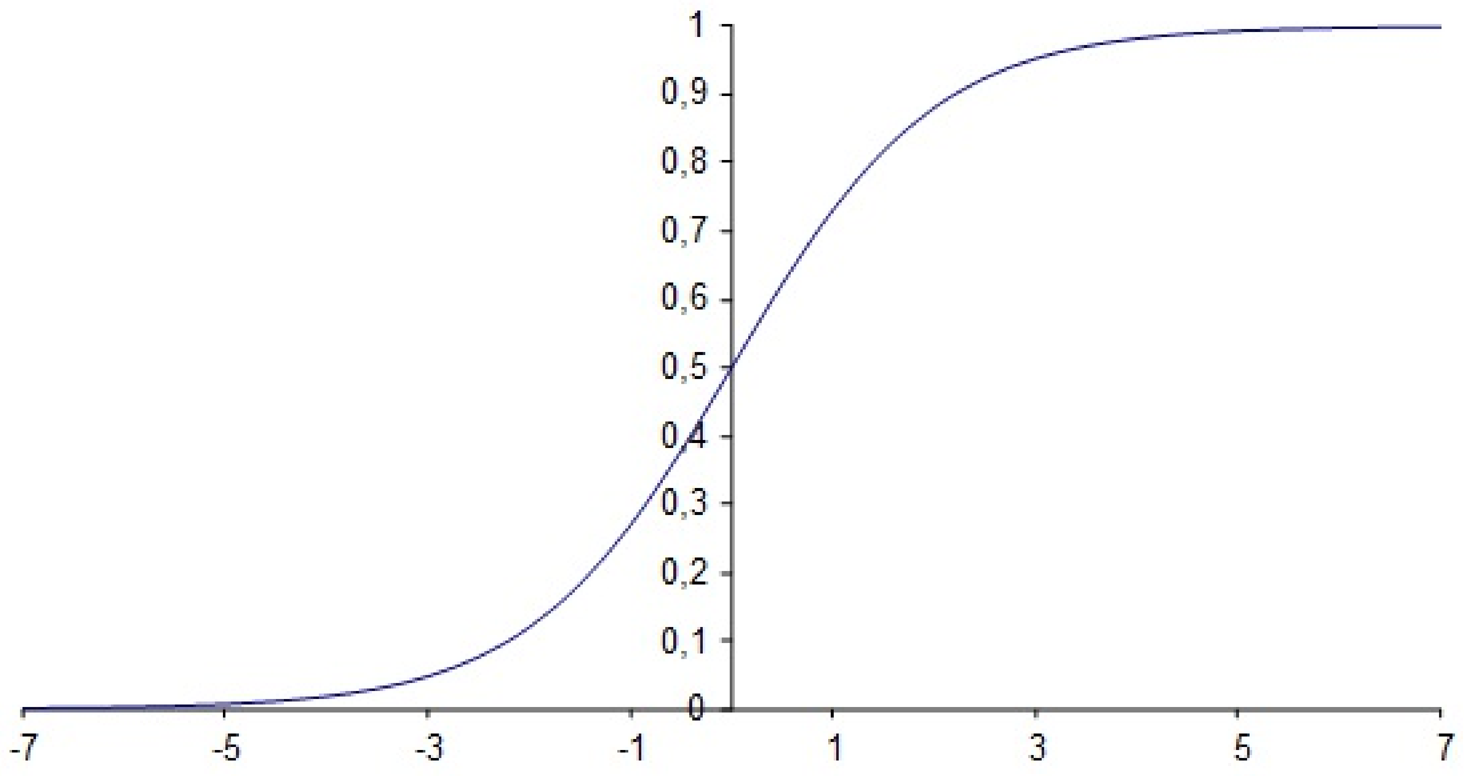
Predictive modeling




```
 dataset$R<-c(1,1,'?',1,1,1,0,0,0,0)
> dataset[,c(1,9,10)]
  name preference R
A    A          R 1
B    B          R 1
C    C          ? ?
D    D          R 1
E    E          R 1
F    F          R 1
G    G          P 0
H    H          P 0
I    I          P 0
J    J          P 0
```

```
training_set<-dataset[-3,-9]
test_set<-dataset[3,-9]
```

Logistic regression



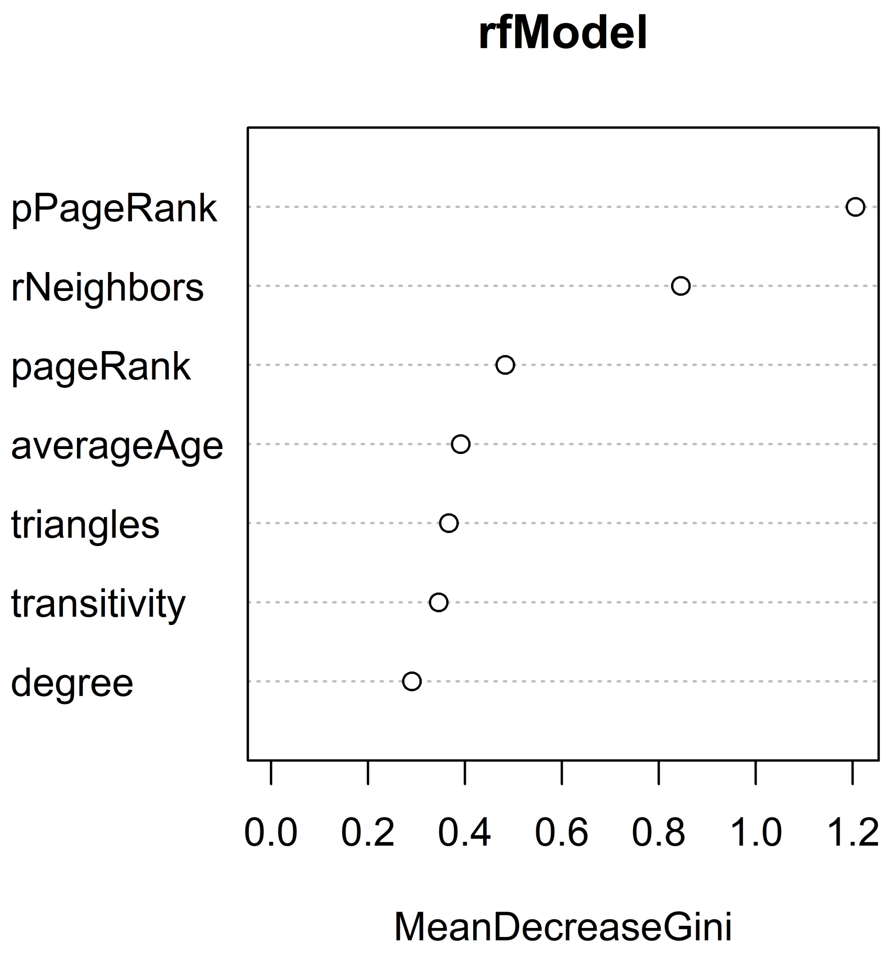

```
glm(R~degree+pageRank, dataset=training_set,family='binomial')
```

```
glm(R~., dataset=training_set,family='binomial')
```

Random forests

```
library(randomForest)
rfModel<-randomForest(R~., dataset=training_set)
```

```
varImpPlot(rfModel)
```



PREDICTIVE ANALYTICS USING NETWORKED DATA IN R

Let's practice!

PREDICTIVE ANALYTICS USING NETWORKED DATA IN R

Evaluating model performance

María Óskarsdóttir, Ph.D.
Post-doctoral researcher

Making predictions

```
library(pROC)
```

- Logistic regression

```
logPredictions <- predict(logModel, newdata = test_set, type = "response")
```

- Random forest

```
rfPredictions<- predict(rfModel, newdata = test_set, type='prob')
rfPredictions
    0     1
C 0.136 0.864
attr(,"class")
[1] "matrix" "votes"
```

AUC

- Probability that a randomly chosen churner gets a higher score than a randomly chosen non-churner
- Displays the trade-off between the model's sensitivity and specificity
- A number between:
  - **0.5**: random model
  - **1**: perfect model

```
library(pROC)
auc(test_set$label, logPredictions)
```

Top decile lift

- How much better is the prediction model at identifying churners, compared to a random sample of customers
- Computes the proportion of actual churners amongst the 10% of customers with the highest predicted churn probability
- Lift value greater than 1 means that the model is better than a random model
- If, in the top 10% of the highest scores there are **60%** churners and in the whole population there are **10%** churners, then the lift is $60/10 = 6$

```
library(lift)
TopDecileLift(test_set$label, predictions, plot=TRUE)
```

PREDICTIVE ANALYTICS USING NETWORKED DATA IN R

Let's practice!

PREDICTIVE ANALYTICS USING NETWORKED DATA IN R

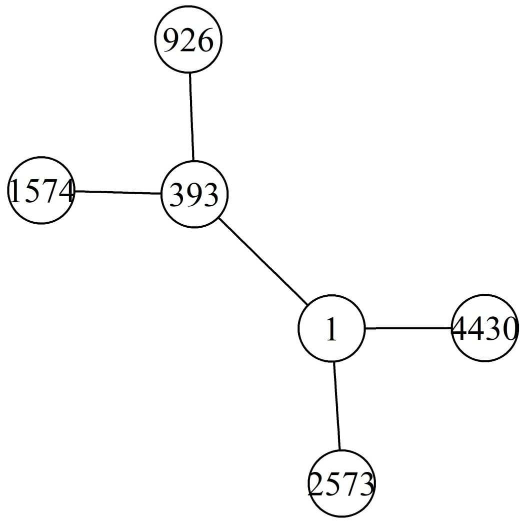
Summary and final thoughts

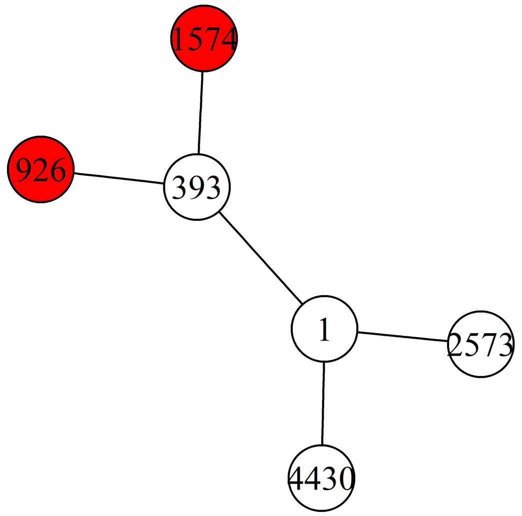
Bart Baesens, Ph.D.
Professor of Data Science, KU Leuven and University of Southampton

Labeled networks

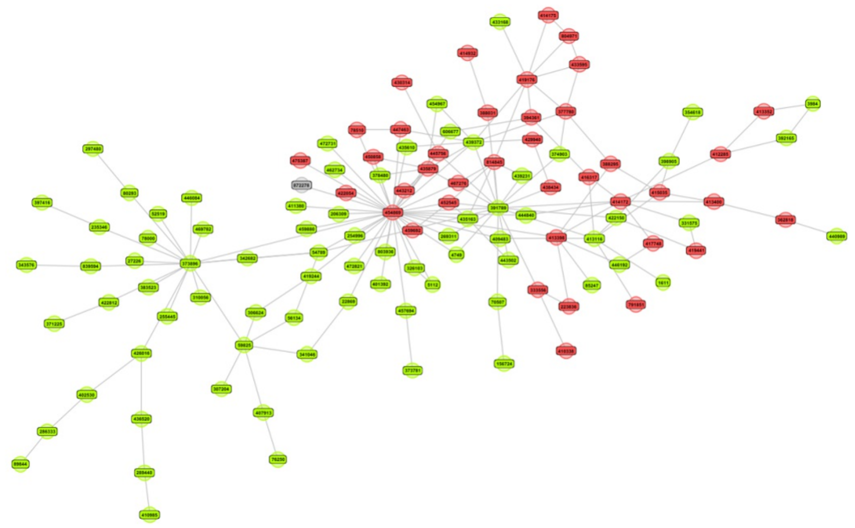
```
> edgeList
  from   to
1    1  393
2    1 2573
3    1 4430
4  393  926
5  393 1574
```

```
> customers
    id churn
1    1     0
2  393     0
3 2573     0
4 4430     0
5  926     1
6 1574     1
```





Homophily



Birds of a feather flock together

Dyadicity: connectedness between nodes with same label

Heterophilicty: connectedness between nodes with opposite labels

Network Featurization

```
g
IGRAPH UN-- 10 19 --
 attr: name (v/c), label (e/c)
 edges (vertex names):
 [1] A--B A--C A--D A--E B--C B--D C--D C--G D--E D--F D--G E--F F--G F--I
[15] G--I G--H H--I H--J I--J
```

```
V(g)$degree<-degree(g)
```

```
g
IGRAPH UN-- 10 19 --
 attr: name (v/c), degree (v/n), triangles (v/n), transitivity
| (v/n), rNeighbors (v/n), averageAge (v/n), pageRank (v/n),
| pPageRank (v/n), label (e/c)
 edges (vertex names):
 [1] A--B A--C A--D A--E B--C B--D C--D C--G D--E D--F D--G E--F F--G F--I
[15] G--I G--H H--I H--J I--J
```

Model building and evaluation

1. Extract dataframe:

```
dataset <- as_data_frame(g, what='vertices')
```

2. Preprocess data set:

   - Missing values, outliers, correlated variables, and normalization

3. Build model:

```
glm(R~., dataset=training_set, family='binomial')
```

4. Make predictions:

```
logPredictions <- predict(logModel, newdata=test_set, type="response")
```

5. Measure performance:

```
auc(test_set$label, logPredictions)
TopDecileLift(test_set$label, predictions, plot=TRUE)
```

PREDICTIVE ANALYTICS USING NETWORKED DATA IN R

Congratulations!