How to Combine **Complex Systems** and **Data Science** ?

Uzay Çetin
Istanbul Bilgi University

kahve

https://uzay00.github.io/kahve/orta.html

Data Science → Complex Systems

- Model parameters of a Complex System can be determined by analysing data
- Complex Systems generates huge amount of data to be analysed.

Complex Systems → Data Science

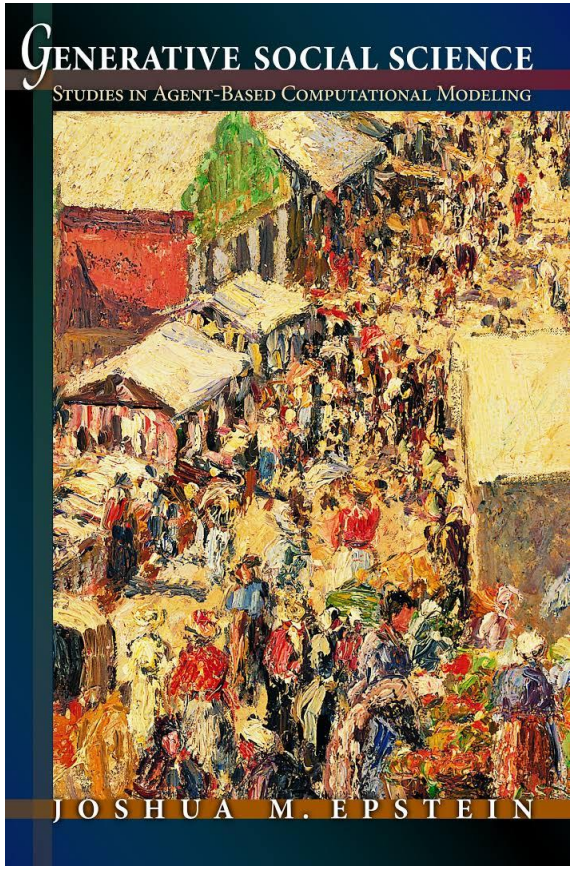
- Agent based models can be transformed to optimisation algorithms to be used in classical machine learning problems.
- Complex network analysis help a lot to analyse graph data.

Complex Systems

More is different !!

- One water molecule is not fluid
- One neuron is not intelligent
- One amino acid is not alive

What makes the difference?

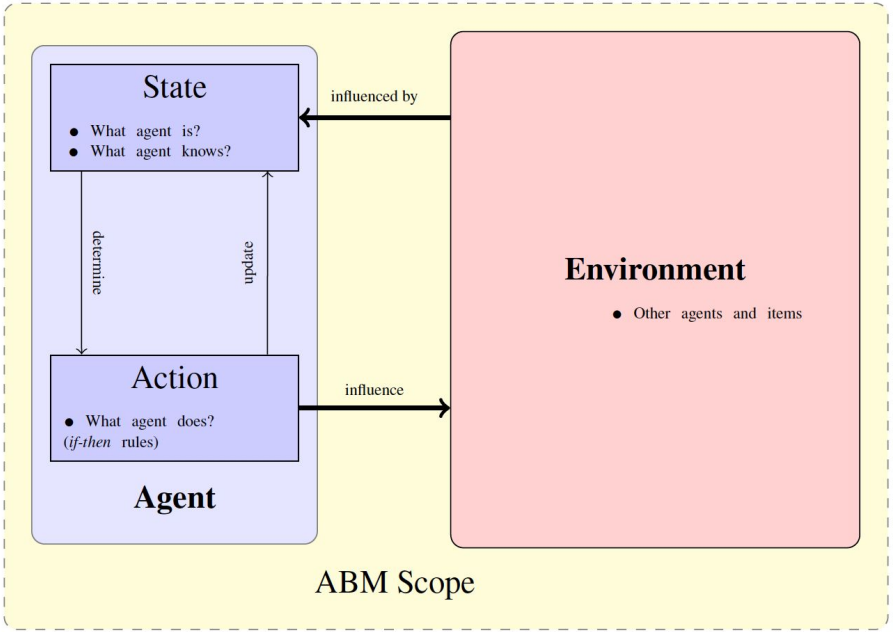


An Agent

is an *autonomous* computational unit

Relationships are primary, agents are secondary!!

Structure Generates Behavior!!



Research Question I

- **How** to improve **improve** the capabilities of **agent-based models** in order to use them for **prediction** just **like** the standard **machine learning** algorithms?

Darpa announced a new program seeking to develop **simulated** social systems of **varying complexity** against which to test their explanatory and predictive performance **social science modeling** methods.

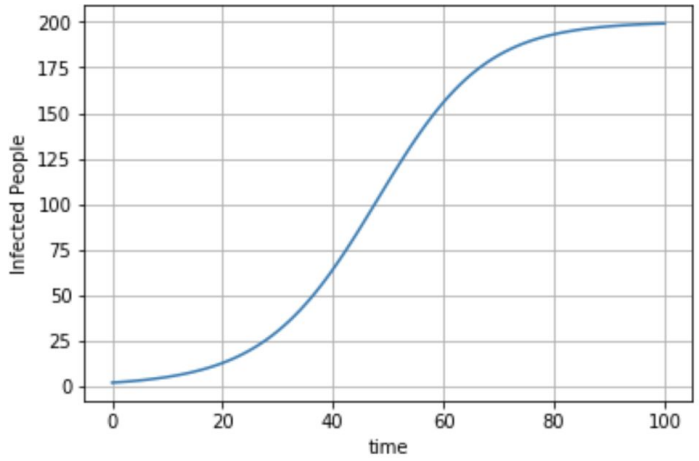
https://www.darpa.mil/news-events/2017-04-07

Modeling Spread of Disease

$$\frac{dI}{dt} = k(M - I)I$$

M	Population size
I	Infected people
k	Probability of spread

```
M, k, dt, I = 200, 0.1, 0.005, [2]
for t in range(100):
    dI = k * (M-I[t]) * I[t] * dt
    I_new = I[t] + dI
    I.append(I_new)
plt.plot(I)
plt.xlabel('time');plt.ylabel('Infected People')
plt.grid(); plt.show()
```

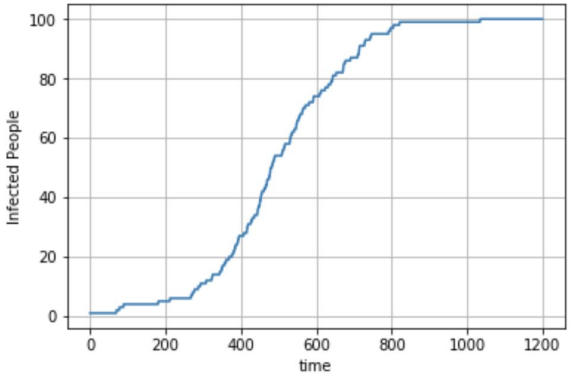


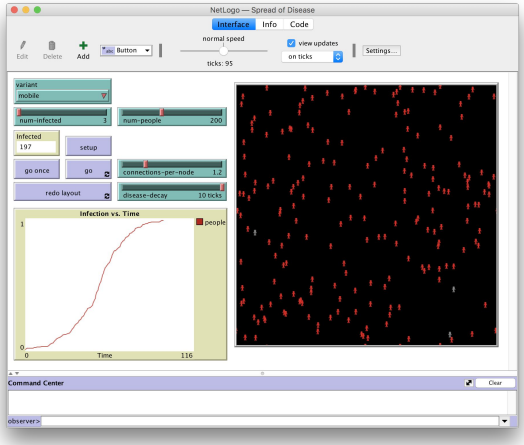
“A **Third** Way of Doing **Science**” - Axelrod, R.

(2003). Advancing the art of simulation in the social sciences.

```
class socialMan():
    def __init__(self, ID, idea = 0):
        self.ID, self.idea = ID, idea
    def give(self):
        return self.idea
    def take(self, new_idea):
        if(self.idea == 0):
            self.idea = new_idea

class socialWorld():
    def __init__(self, N = 100, time = 1200):
        self.N, self.time = N, time
        self.socialMen = [socialMan(i) for i in range(self.N)]
        self.socialMen[0].take(1) # Initially, only one adopter

    def exchange(self):
        # Select two random socialMan, A:giver and B:taker
        A, B = self.socialMen[randint(0, self.N-1)], self.socialMen[randint(0, self.N-1)]
        if A.idea == 1:
            B.take(A.give())

    def ideas(self):
        return [man.idea for man in self.socialMen]

    def runTheWorld(self):
        adopters = [0] * self.time
        for t in range(self.time):
            world.exchange()
            adopters[t] = sum(world.ideas())
        return adopters

world = socialWorld()
plt.plot(world.runTheWorld())
plt.xlabel('time'); plt.ylabel('Infected People');plt.grid()
```

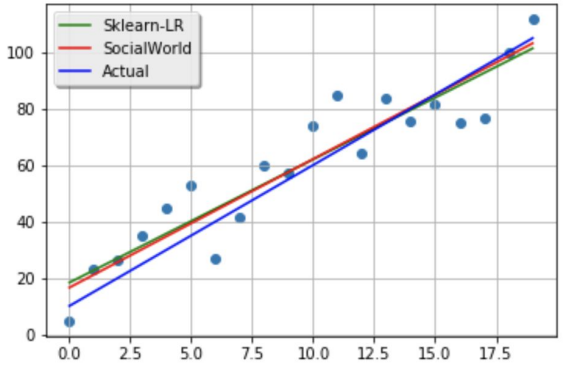
Simulation, AI, OOP





Linear Regression With Social Optimisation

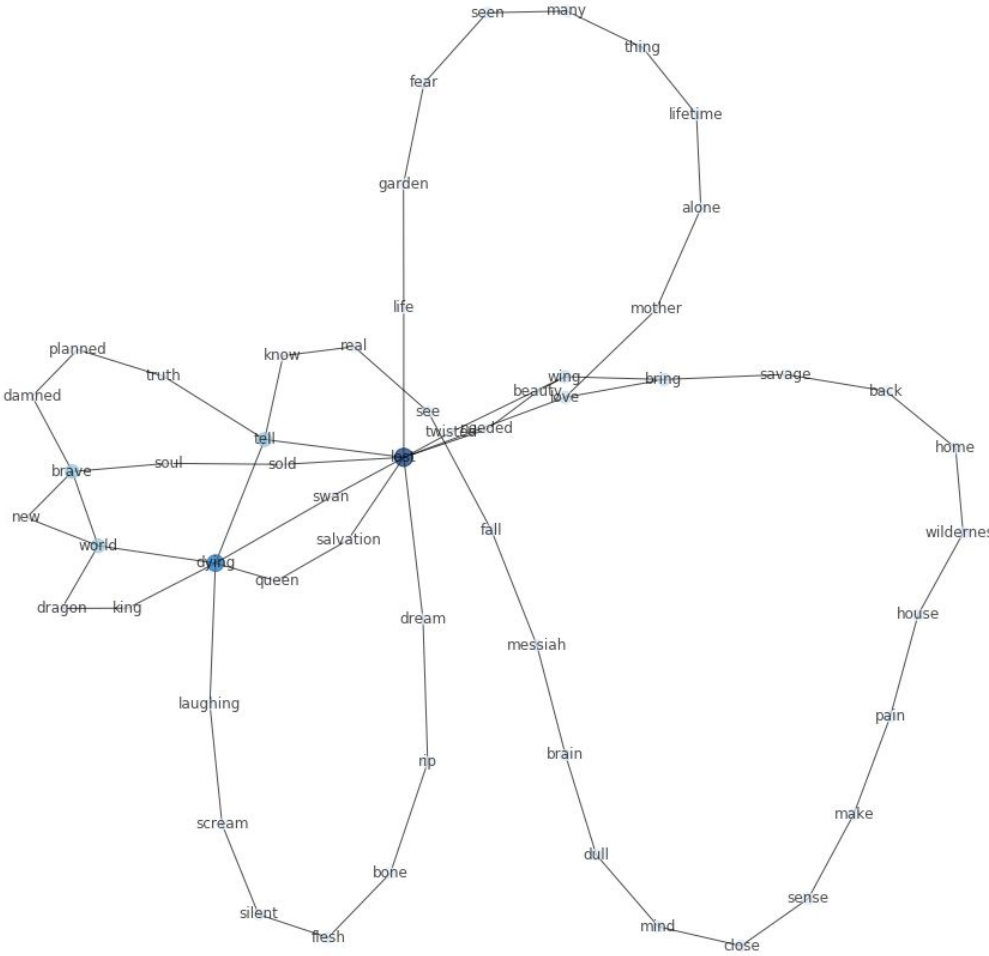
```python
class socialMan():
    def __init__(self, ID, X, y, max_w, w_shape):
        self.ID, self.X, self.y, self.max_w = ID, X, y, max_w
        self.W = self.max_w * np.random.rand(*(w_shape))

    def immitate(self, other): # immitate betters
        if other.performance() > self.performance():
            row = np.random.randint(other.W.shape[0])
            self.W[row,:] = other.W[row,:]
        if np.random.rand() < 0.1: # Go on your own
            row = np.random.randint(self.W.shape[0])
            self.W[row,:] = self.max_w * np.random.rand(self.W.shape[1])

    def performance(self):
        return 1/(1+np.sum(np.power(self.y - self.W.T.dot(self.X),2)))

class socialWorld():
    def __init__(self, X, y, N = 20, time = 10000, max_w = 20, w_shape = (2,1)):
        self.X, self.y, self.N, self.time = X, y, N, time
        self.socialMen = [socialMan(i, X=X, y=y, max_w = max_w, w_shape = w_shape)
                          for i in range(self.N)]

    def best(self):
        score, who = self.socialMen[0].performance(), 0
        for i in range(self.N):
            if self.socialMen[i].performance() > score:
                score, who = self.socialMen[i].performance(), i
        return self.socialMen[who].W

    def predict(self, X):
        return self.best().T.dot(X)

    def runTheWorld(self):
        for i in range(self.time):
            pair = np.random.randint(self.N, size = 2)
            A, B = pair[0], pair[1]
            self.socialMen[A].immitate(self.socialMen[B])
        return self.best()
```




```python
# Generate Data From Correct (Hidden) Function with Noise
correctW, M = np.array([[10],[5]]), 20 # M observations
X = np.array([np.ones(M),np.arange(M)])
y = correctW.T.dot(X) + np.random.randn(X.shape[1])
# Linear Regression from Scikit-Learn
from sklearn.linear_model import LinearRegression
X_train = np.arange(M).reshape(M,1)
lr = LinearRegression()
lr.fit(X_train, y.T)
y_pred = lr.predict(X_train)
# Social World Optimization
world = socialWorld(X=X, y=y)
world.runTheWorld()
# Plot Results
plt.scatter(X_train, y ); #Points are observations
plt.plot(X_train, y_pred, 'g')
plt.plot(X_train, world.predict(X).T, 'r')
plt.plot(X_train, correctW.T.dot(X).T, 'b')
plt.grid()
plt.legend(("Sklearn-LR", "SocialWorld", "Actual"))
```

Research Question II

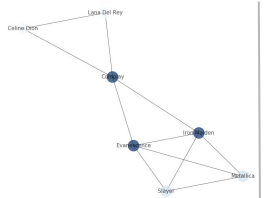
- What can **Complex Network** analysis bring and **offer** to text mining, **sentiment analysis** and **data curation?**



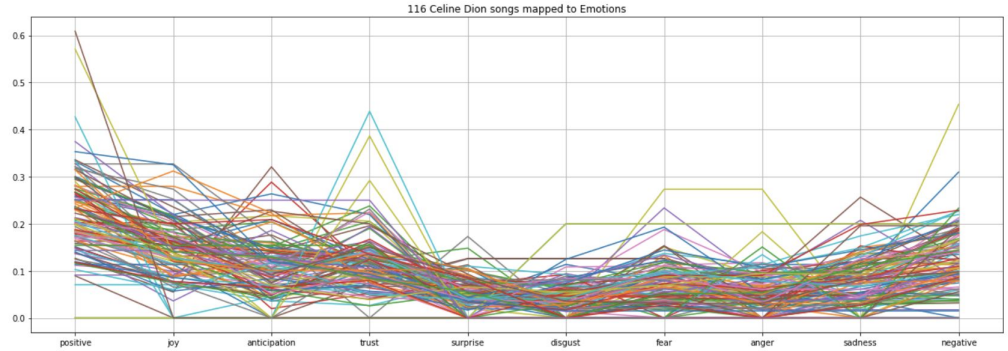
Iron Maiden - Brave New World

Artists and emotions

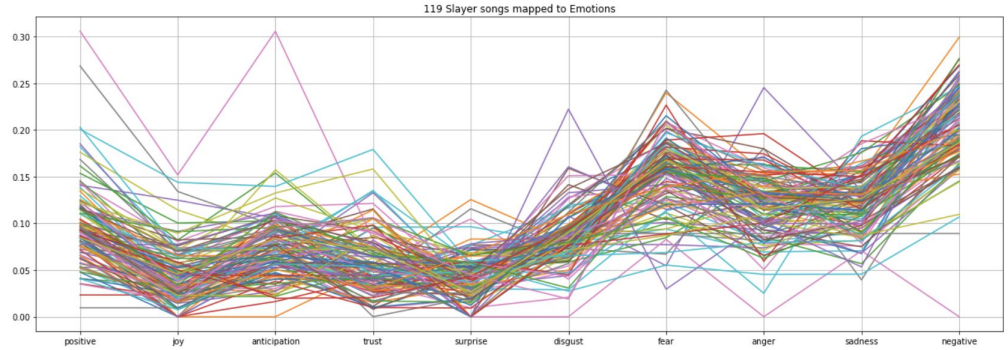
expressed in their lyrics.




```
V_Celine_Dion = drawArtistEmotion(artist_name = 'Celine Dion')
```




```
V_Slayer = drawArtistEmotion(artist_name = 'Slayer')
```

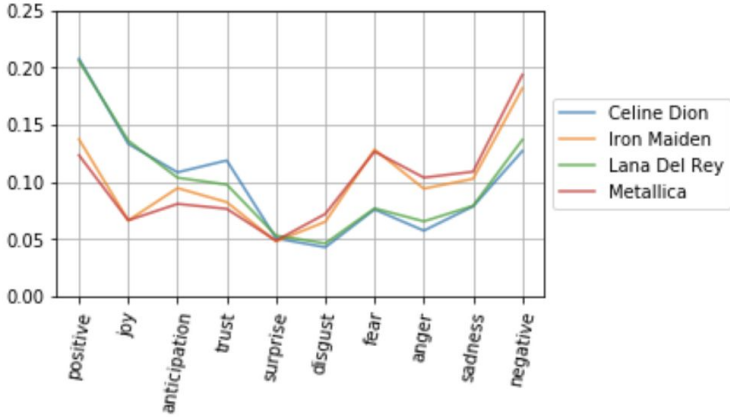



```
In [182]: M['Slayer'].sort_values(ascending=False)[:10]
```

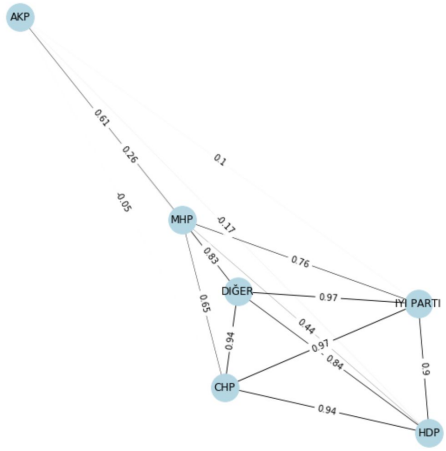
```
Out[182]: Slayer        1.000000
          Misfits       0.988156
          X-Raided      0.986683
          W.A.S.P.      0.981440
          Unearth       0.976719
          Venom         0.974885
          Megadeth      0.973782
          Death         0.956182
          Korn          0.955097
          Zebrahead     0.952159
          Name: Slayer, dtype: float64
```

```
In [186]: M['Slayer'].sort_values(ascending=False)[-12:-2]
```

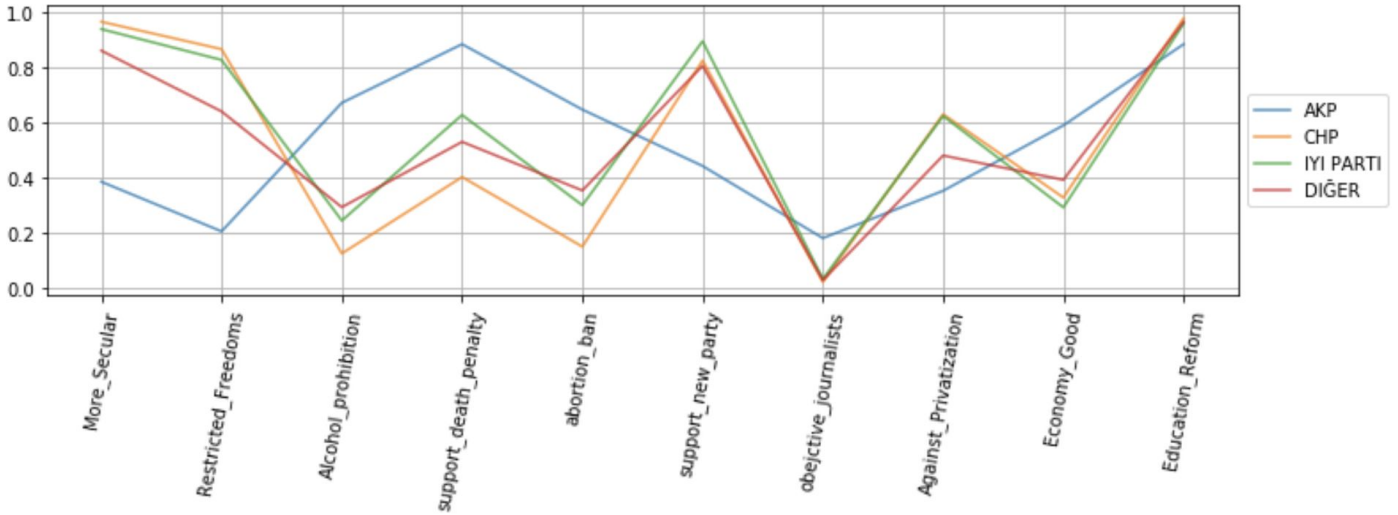
```
Out[186]: Gloria Gaynor      -0.198972
          Andy Williams      -0.199561
          Planetshakers      -0.209186
          Raffi              -0.215793
          Christmas Songs    -0.216911
          Israel             -0.218784
          Jose Mari Chan     -0.223031
          Eppu Normaali      -0.254548
          Dewa 19            -0.258662
          Vera Lynn          -0.351095
          Name: Slayer, dtype: float64
```



Political forecasting
With Machine Learning

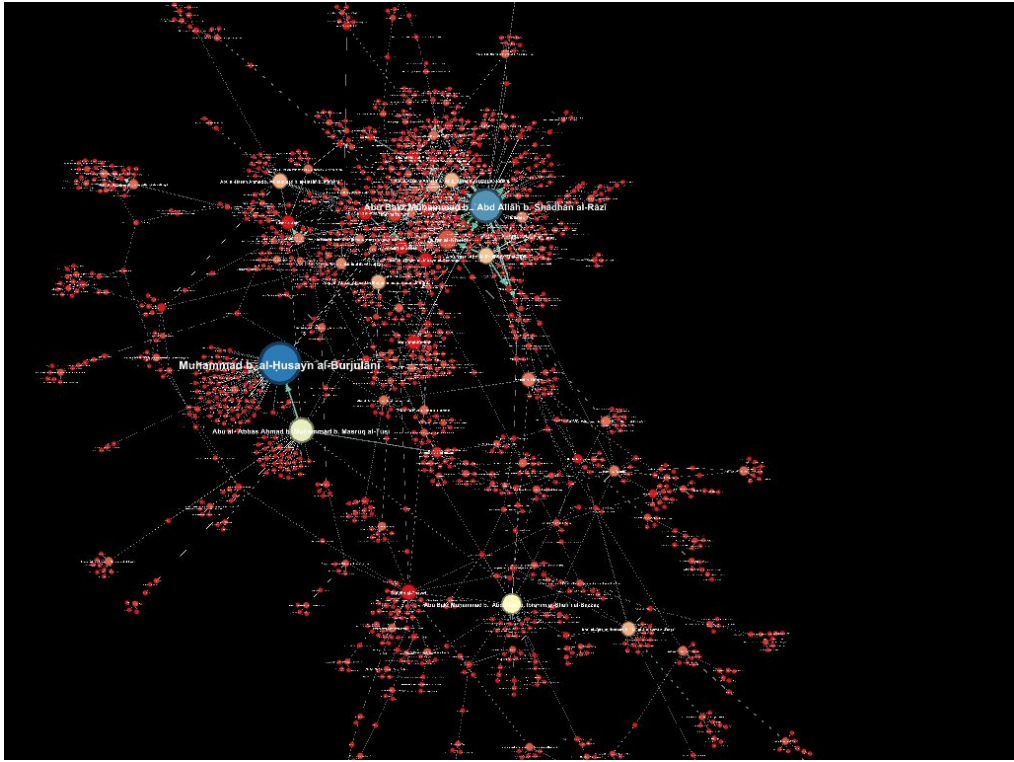


	features	importance
10	More_Secular	0.222866
12	Restricted_Freedoms	0.201417
9	Alcohol_prohibition	0.182448
7	support_death_penalty	0.145602
11	abortion_ban	0.095982
13	support_new_party	0.054736
8	obejctive_journalists	0.023909
1	Age	0.022315
6	Against_Privatization	0.020549
3	Education	0.010892
2	Region	0.007523
4	Economy_Good	0.005733
5	Education_Reform	0.003522
0	Sex	0.002505



Networked Metrics of Leadership: An Analysis of Early Sufism (Iraq, 9th-10th c.)

حَدَّثَنَا أَبُو مُحَمَّدٍ ازْدِيَارُ بْنُ سُلَيْمَانَ، ثَنَا جَعْفَرُ بْنُ مُحَمَّدٍ، قَالَ: قَالَ أَبُو الْحَسَنِ الْمَزَيِّنُ: «التَّصَوُّفُ قَمِيصٌ قَمَّصَهُ اللهُ أَقْوَامًا , فَإِنْ أُلْهِمُوا عَلَيْهِ الشُّكْرَ , وَإِلَّا كَانَ خَصْمَهُمْ فِي ذَلِكَ اللهُ عَزَّ وَجَلَّ» وَسُئِلَ الْخَوَّاصُ عَنِ التَّصَوُّفِ فَقَالَ: «اسْمٌ يُغَطَّى بِهِ عَنِ النَّاسِ، إِلَّا أَهْلَ الدِّرَايَةِ , وَقَلِيلٌ مَا هُمْ»



Two set of keywords: **Asceticism** vs **Mysticism**

- "**low entropy**"/*homogenous* messages
- "**high entropy**"/*heterogeneous* message

RQ: How the **diversity** in the sayings of a religous person affects his popularity?

Thank You

Mail:
uzay.cetin@bilgi.edu.tr

Web:
https://uzay00.github.io