LA Assignment 2 Report

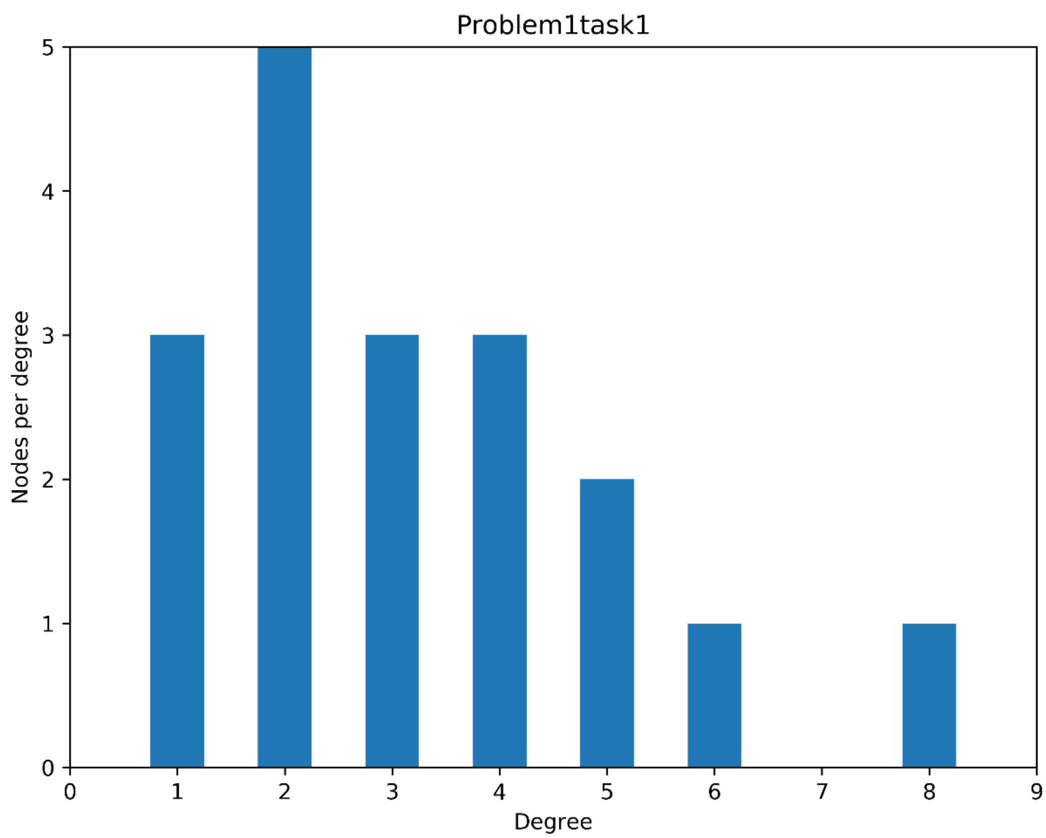
Mohit Gupta

15755, M.Tech, CSA

Problem 1

Task 1

Plot→

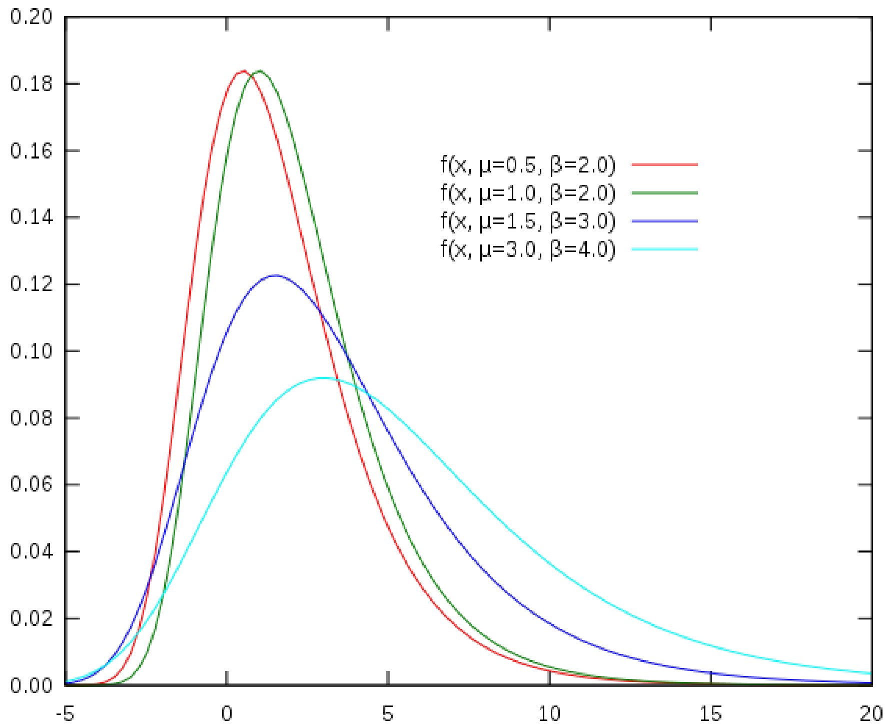


We Infer→

High no. Nodes with less degree (like deg 2 on X-axis in above plot).

Very few nodes with high degree(like deg 8 in X-axis in above plot).

The Degree Distribution above we have got is something discrete case of log-normal distribution . This is how a continuous log-normal distribution(which is skewed on one side) looks like.



Task 2:

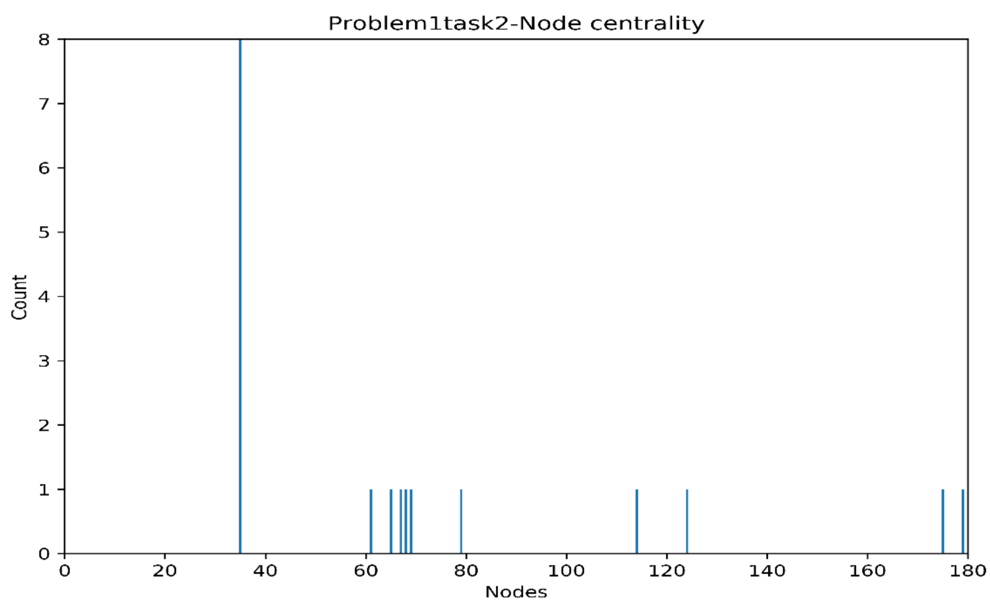


Figure 1 Calculated using my own code.

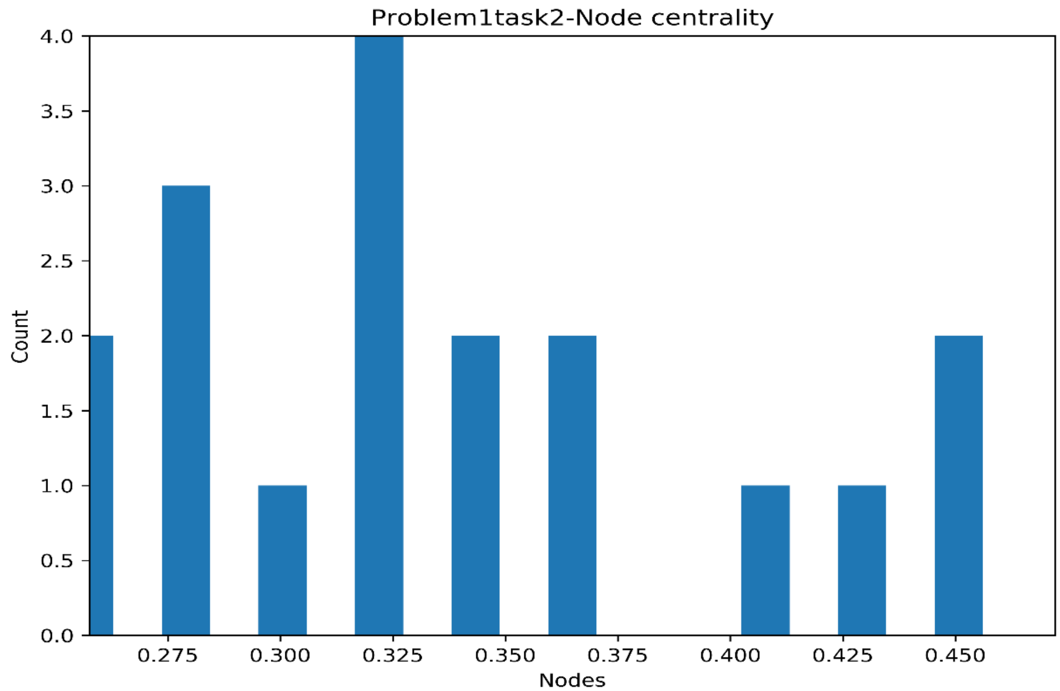


Figure 2 Calculated using Networkx- node centrality.

Top two central nodes <By my code>:-

Little-finger, Varys.

Task 3:

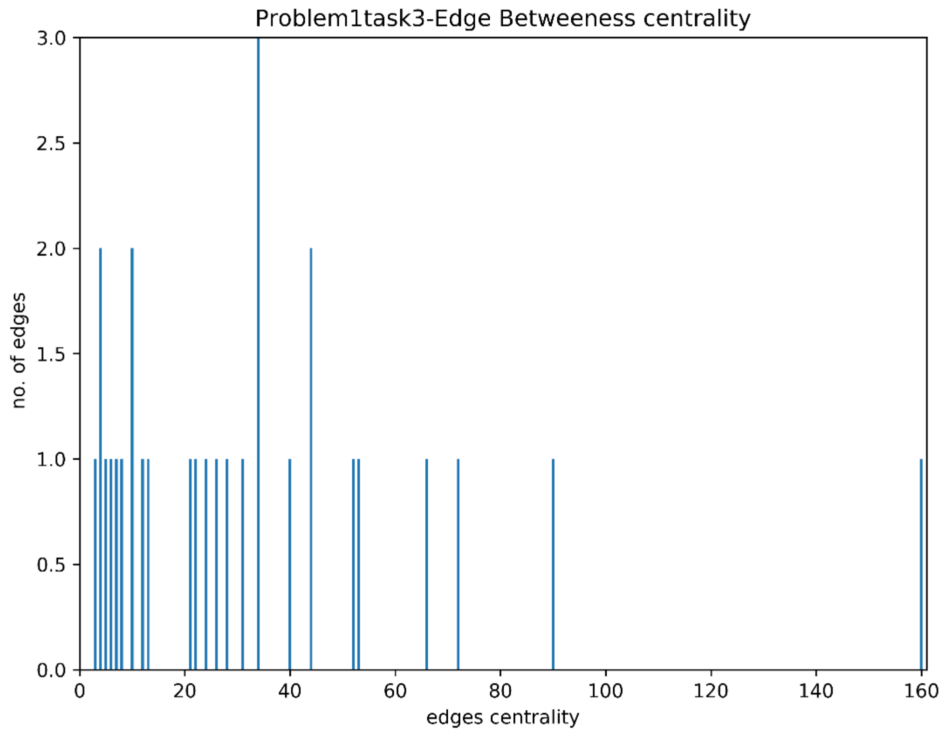


Figure 3 Calculated using my own code

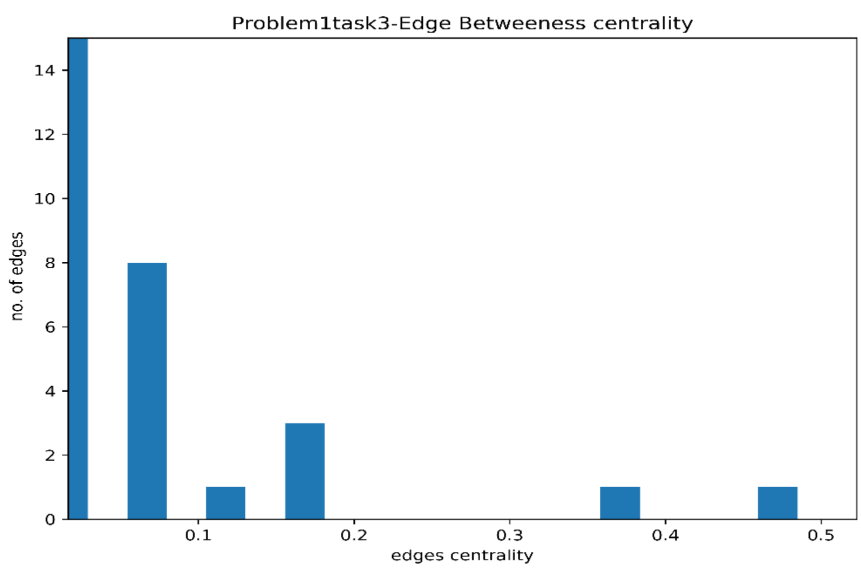


Figure 4 Calculated using Networkx- edge betweeness.

Most central edge <By my Code>:

Edge between Varys and Little-finger is the central one.

Task 4:

Eigen Values for above case are Real.

Also for symmetric matrix, Eigen values are always real.

Proof (cited from math.stackexchange.com):

$Av = \lambda v$ combined with $A = A^T$ gives

$$\langle Av, Av \rangle = v^* A^* Av = v^* A^T Av = v^* A^2 v = \lambda^2 ||v||^2.$$

Now $\lambda^2 = \frac{\langle Av, Av \rangle}{||v||^2} \geq 0$, as the quotient of a nonnegative real number by a positive one. So λ must be real.

Task 5

Getting zero as the smallest eigen value. Also getting eigen vector components equal to 1. No such difference we get.

Smallest and Second smallest eigen values/vectors:-

smallest one:

value: 0.0

vector: [0.23570226 0.23570226 0.23570226 0.23570226 0.23570226 0.23570226

0.23570226 0.23570226 0.23570226 0.23570226 0.23570226 0.23570226

0.23570226 0.23570226 0.23570226 0.23570226 0.23570226 0.23570226]

second smallest one:

value: 0.12152974241615154

vector: [-0.2960552 -0.24646062 -0.27048116 -0.26007583 -0.27690165 -0.17887273

-0.26007569 -0.28244098 0.2068688 0.18646789 0.202129 0.22522628

0.22095186 0.24313206 0.23009202 0.27676755 0.2068688 0.07285959]

Task 6

Eigen vector corresponding to second smallest eigen value tries to divide the graph into two parts, each of which have high information within it & less mutual information between these two divided clusters.

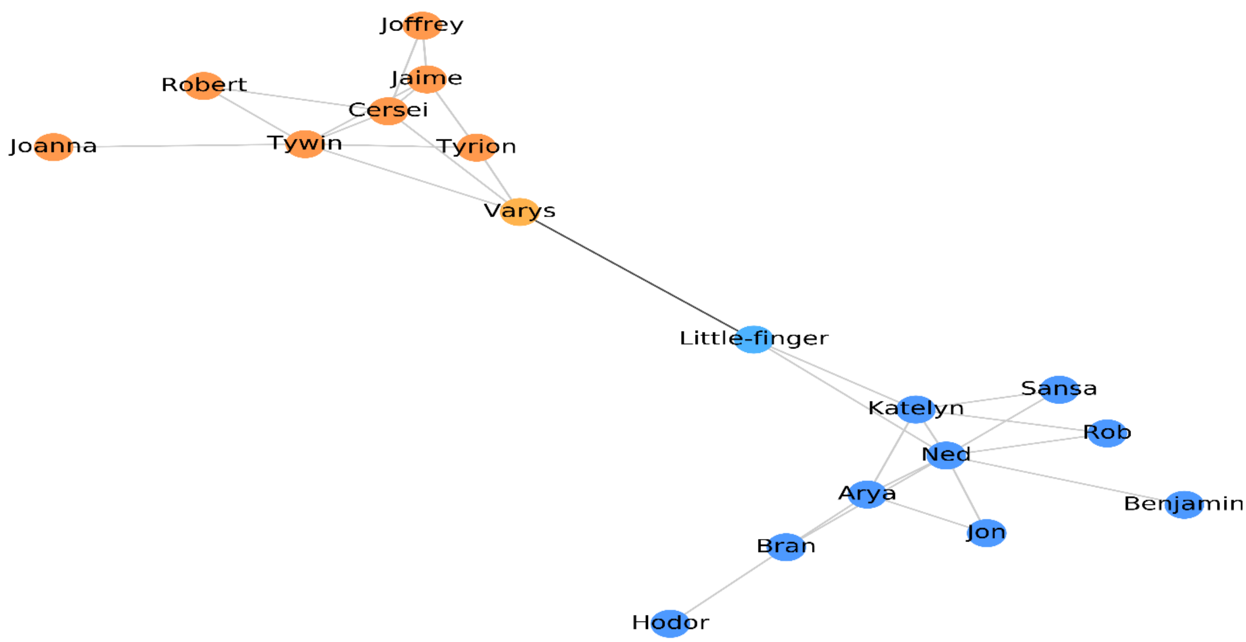


Figure 5 Done as per given question.

Based on the second small eigen value, we get a graph where two divided clusters are connected by the central most edge. This gives an inference of each cluster nodes containing high information amongst themselves and least information with another cluster nodes.

We can also use K-means clustering algorithm(k==2 here) for grouping into two houses .

Here clustering is done based on cohesion/ centroid in the graph. I have used SKLEARN to compute the below result.

See the results below:-

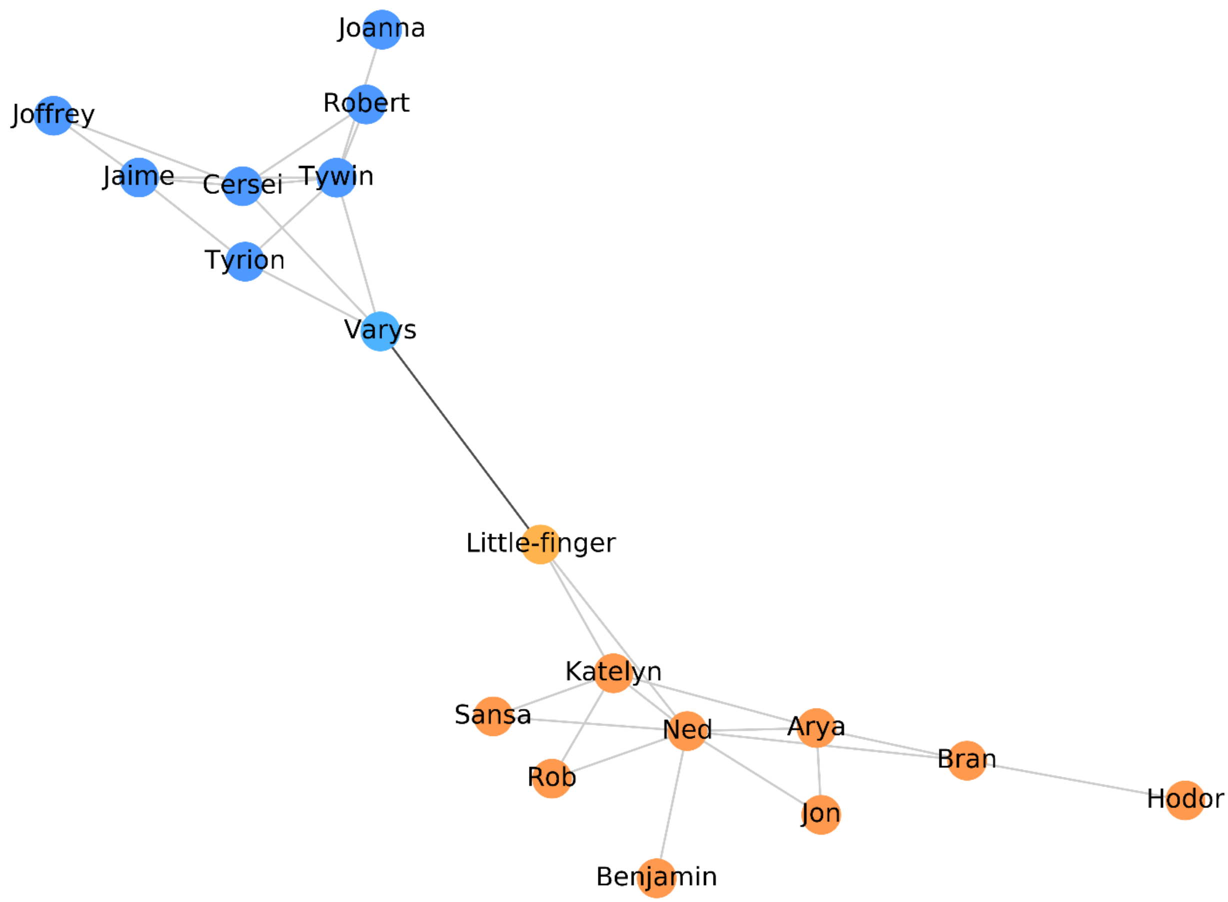


Figure 6 Done clustering using K-means. Here K=2

Bonus 1:

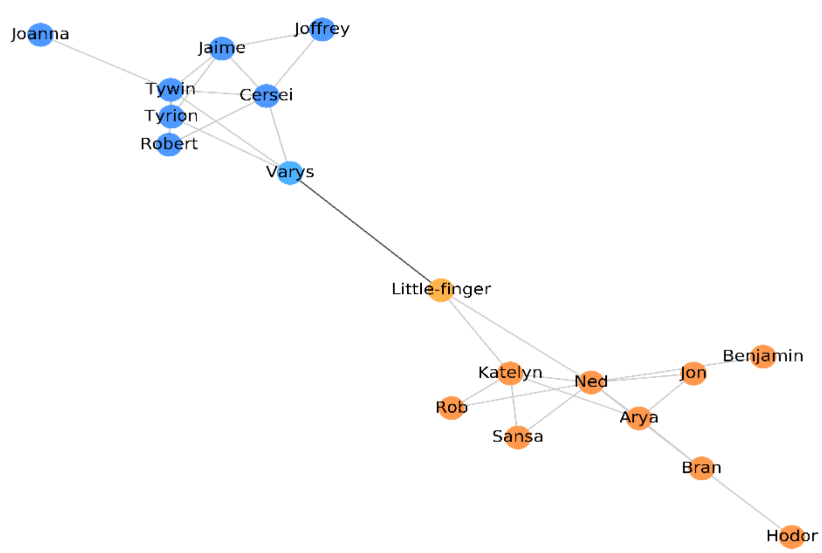


Figure 7 Using Numpy eigen vectors

No such difference found from results.

Bonus 2

We can use the Hierarchical division algorithm to divide the nodes into K-groups.

The algo can be explained as:

Take kth smallest eigen vector where k=2,3,4…

Using it we can divide the nodes into two clusters.

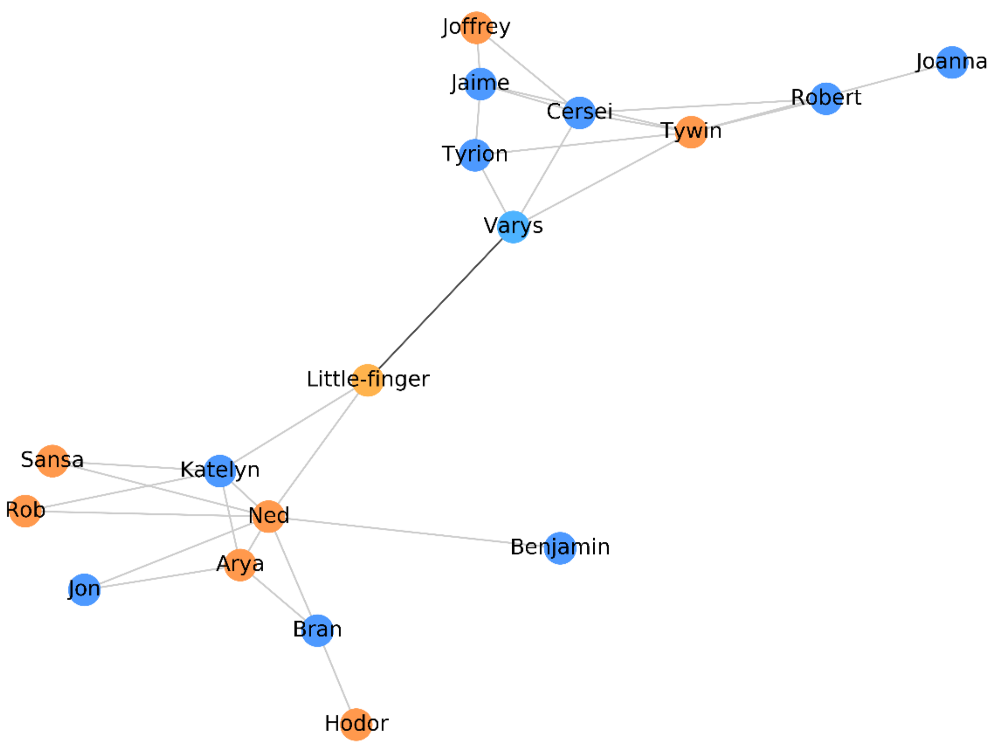
Take next smallest eigen vector i.e., (k+1)th.

Pick any cluster based on criteria like variance and divide it into two parts using the same technique we used to divide the nodes at first.

We repeat the step until we get K-groups.

Bonus 3

If Eigen vector corresponding to second largest eigen value is taken, can't divide the nodes into two groups as we did in case of second smallest eigen vector.



It looks as if nodes colored with orange only trying to cover max no. of edges (something similar to vertex cover).

It also looks like an attempt is being done to make it as a 2-coloring graph(partial bi-partite graph) with some nodes not following the constraint like two nodes connected to each other should have different color.

Bonus 4

Most central edge in the graph is acting as a bridge.

Problem 2

(Task2): Eigen vectors turns out to be orthogonal.

Task 4 and 5 results:-

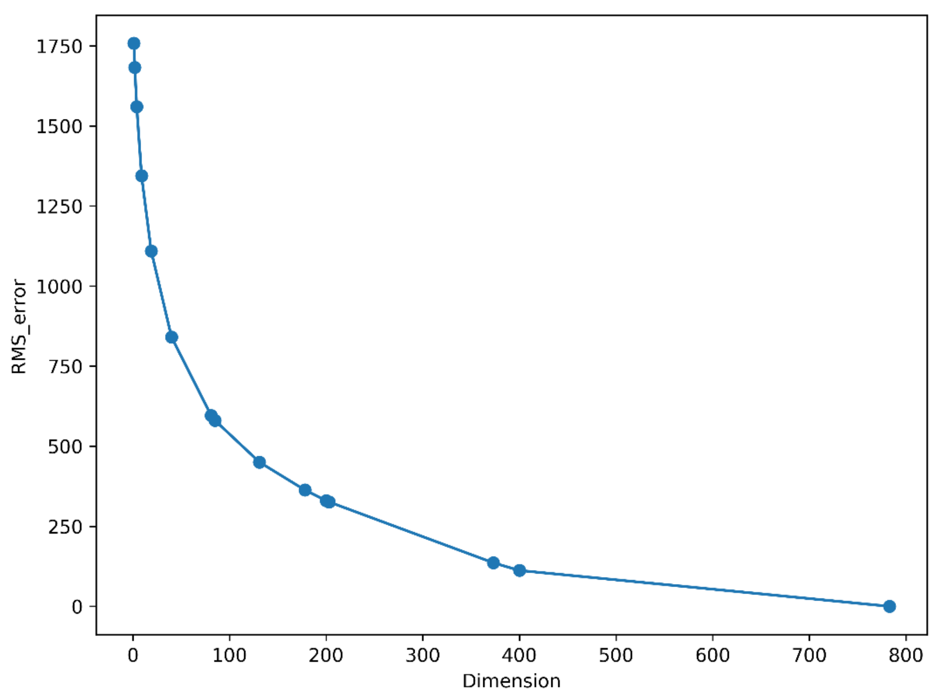


Figure 8 Reconstruction error

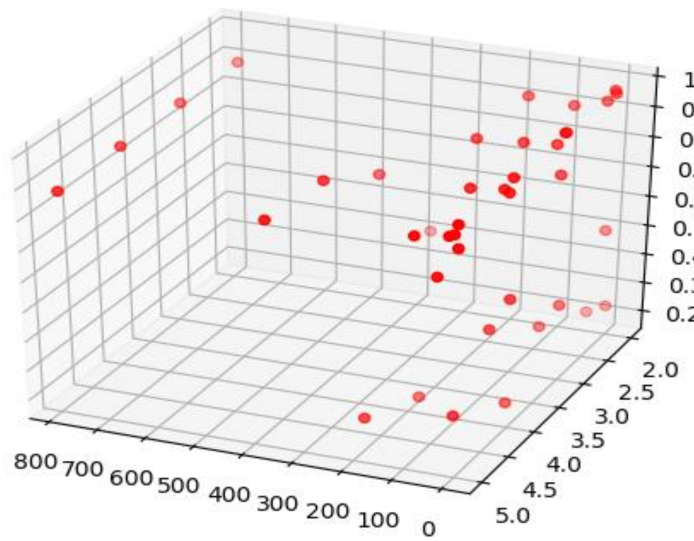
(Task6):

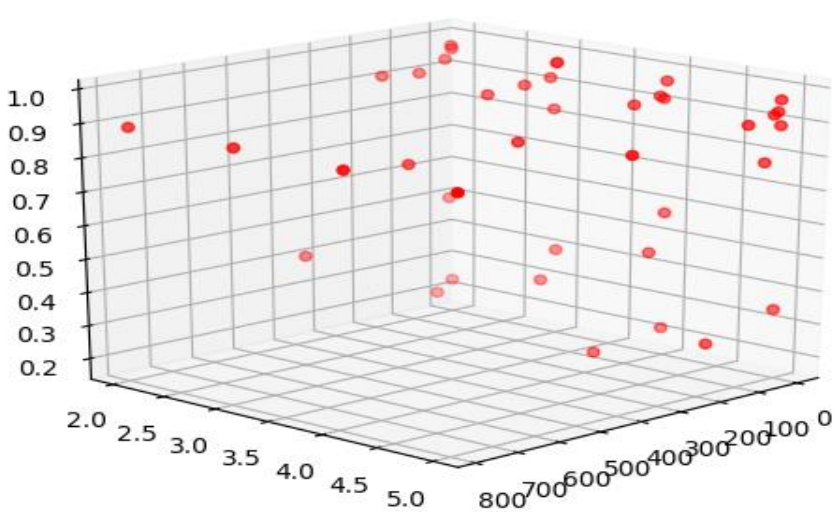
Based on results from training set, have picked k=2 and M=40.

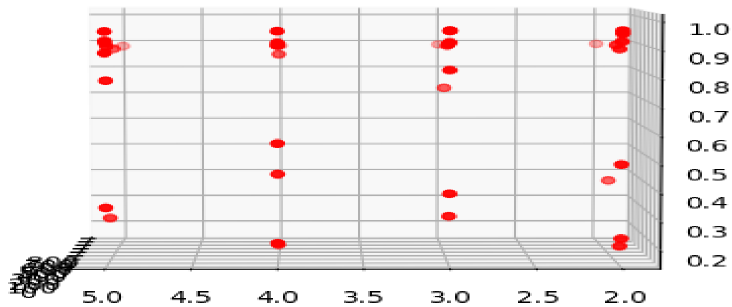
Rest task outputs can be found in output_data folder.

Please consider the following labelling (forgot to label them while generating the plot).

Values 0 to 800 along X-axis is M. Values 2 to 5 along Z-axis is K and Values 0 to 1 along Y-axis is accuracy.

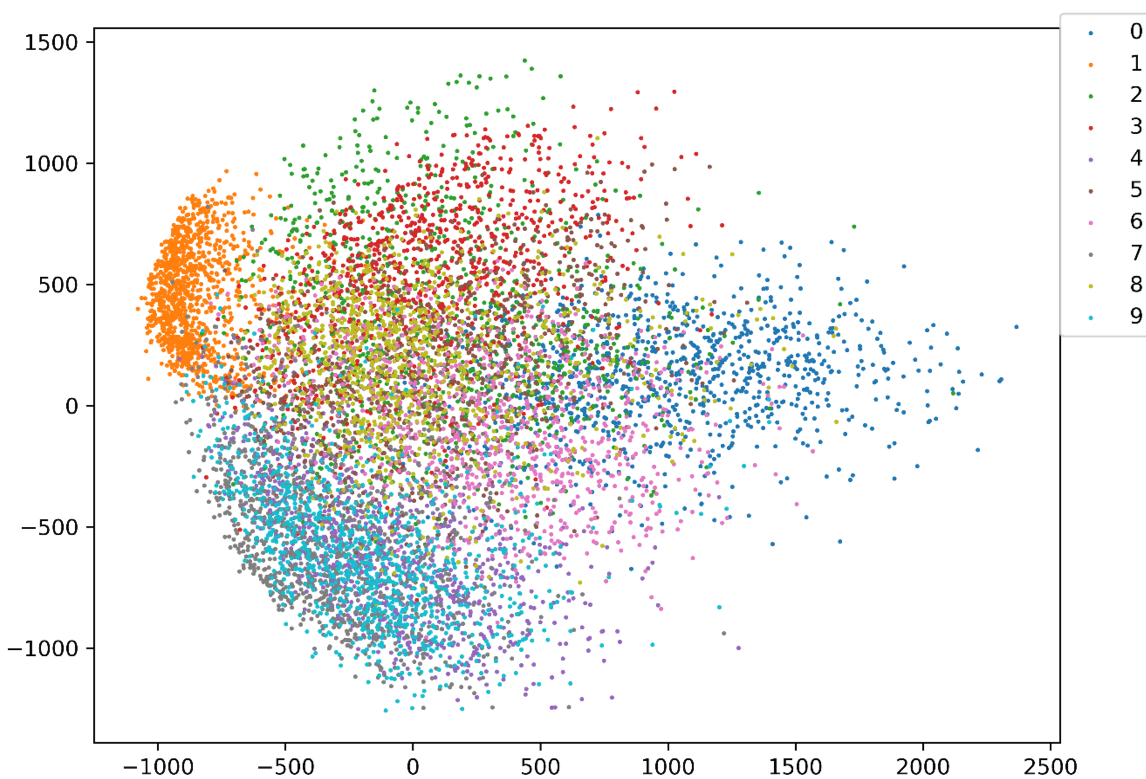






Bonus 1

The Plotting is shown below as well stored in output_plots folder. We can see that for each class, data is spread so much as well as overlapping very much that it becomes very hard for ML based algo to predict with great accuracy.



We see Class 0 and 9 are much of overlapping with each other.

The Class 1 with color ‘orange’ is the one where least overlapping is happening meaning Image with label 1 has good chances of getting rightly classified.

As such we see no such uniformity in patterns.

BONUS 2:-

Based on results we got on training sklearn K-NN model and ours.

We see, for M=40 and K=3 we get best accuracy from sklearn. And for M=40 and K=2 we get best accuracy on our model with both models having 96.65% accuracy.

Also we see some different results from sklearn KNN model. When we go for higher dimensions (M), we see poor performance in terms of accuracy which in our model doesn’t tend to happen.