**Node2vec** algorithm seems feasible to meet latency requirements and processing user persona files.

**Graph will be heterogenous and there can be a vertices considering data points from data points list presented in data lake document.**

Main domains -

**1)User id**

**2)IAB Categories**

**3)Publisher Segments**

**4)ML segments**

**5)Color Segments**

**6)Demographic Segments**

**7)Third party databases segments (google trends, dbpedia and more) (keyword affinity scores computed based on campaigns domains on Aajtak/Indiatoday).**

**These Segments - Augmentation databases are also available using Entity Metadata Enrichment Algorithms to enhance graph quality.**

**Below is the list of operations one can perform on vertices.**

aajtak\_adamic\_adar Adamic Adar Topological Link Prediction

aajtak\_article\_rank Article rank

aajtak\_astar A\* search

aajtak\_betweenness\_cent Betweenness centrality

aajtak\_bfs Breadth-first search

aajtak\_closeness\_cent\_approx Approximate closeness centrality

aajtak\_closeness\_cent Closeness centrality

aajtak\_common\_neighbors Common neighbors topological link prediction

aajtak\_cosine\_batch Cosine similarity for each pair of vertices, computed in batches

aajtak\_cosine\_nbor\_ap Cosine similarity for each pair of vertices

aajtak\_cosine\_nbor\_ss Cosine similarity from a single vertex

aajtak\_cycle\_detection Rocha–Thatte algorithm for cycle detection; output the cycles

aajtak\_cycle\_detection\_count Rocha–Thatte algorithm for cycle detection; output the number of cycles

aajtak\_degree\_cent Degree centrality

aajtak\_eigenvector\_cent Eigenvector centrality

aajtak\_embedding\_cosine\_sim One-to-Many embedding cosine similarity

aajtak\_embedding\_pairwise\_cosine\_sim Pairwise embedding cosine similarity

aajtak\_estimate\_diameter Heuristic estimate of graph diameter

aajtak\_fastRP FastRP graph embedding

aajtak\_greedy\_graph\_coloring Greedy graph coloring

aajtak\_harmonic\_cent Harmonic centrality

aajtak\_influence\_maximization\_CELF Influence maximization using CELF

aajtak\_influence\_maximization\_greedy Influence maximization using greedy method

aajtak\_jaccard\_batch Jaccard similarity for each pair of vertices, computed in batches

aajtak\_jaccard\_nbor\_ap Jaccard similarity for each pair of vertices

aajtak\_jaccard\_nbor\_ss Jaccard similarity from a single vertex

aajtak\_kcore K-Core

aajtak\_kmeans K-Means

aajtak\_knn\_cosine\_all k-Nearest Neighbor classification, using cosine similarity, batch

aajtak\_knn\_cosine\_cv Cross validation for k-Nearest Neighbor, using cosine similarity

aajtak\_knn\_cosine\_ss k-Nearest Neighbor classification, using cosine sim., single source

aajtak\_label\_prop Label propagation method for community detection

aajtak\_lcc Local clustering coefficient

aajtak\_louvain\_distributed Distributed & parallel Louvain Modularity optimization

aajtak\_louvain\_parallel Parallel Louvain Modularity optimization

aajtak\_maxflow Maxflow

aajtak\_maximal\_indep\_set Maximal independent set

aajtak\_msf Minimum spanning forest (MSF)

aajtak\_mst Minimum spanning tree (MST)

aajtak\_node2vec node2vec graph embedding

aajtak\_pagerank\_pers Personalized PageRank

aajtak\_pagerank\_wt Weighted PageRank

aajtak\_pagerank PageRank measurement of relative influence of each vertex

aajtak\_preferential\_attachment Preferential attachment topological link prediction

aajtak\_random\_walk Random walk generator

aajtak\_random\_walk\_batch Random walk generator, in batches for greater memory efficiency

aajtak\_resource\_allocation Resource allocation topological link prediction

aajtak\_same\_community Same community topological link prediction

aajtak\_scc Strongly connected component detection

aajtak\_scc\_small\_world Strongly connected component detection

aajtak\_shortest\_ss\_any\_wt Single-Source shortest paths

aajtak\_shortest\_ss\_no\_wt Single-Source shortest paths without weight

aajtak\_shortest\_ss\_pos\_wt Single-Source shortest paths with positive weight

aajtak\_slpa Speaker-Listener Label Propagation

aajtak\_tri\_count\_fast Count all the triangles, faster but using more memory

aajtak\_tri\_count Count all the triangles, memory efficient

aajtak\_total\_neighbors Total neighbors topological link prediction

aajtak\_wcc Weakly (undirect) Connected component detection

aajtak\_wcc\_small\_world Weakly (undirect) Connected component detection

aajtak\_weighted\_random\_walk Weighted random walk generator

aajtak\_weighted\_random\_walk\_batch Weighted random walk generator, in batches for greater memory efficiency

**Functions above can be used to optimize**

1)Cross device algorithm

2)Category churn

3)Next category prediction

4)Cookie pool derivation based on custom segments (all data points in data lake)