GRAPH ALGORITHM APPLICATION

UNIT TITLE: DATA MINING

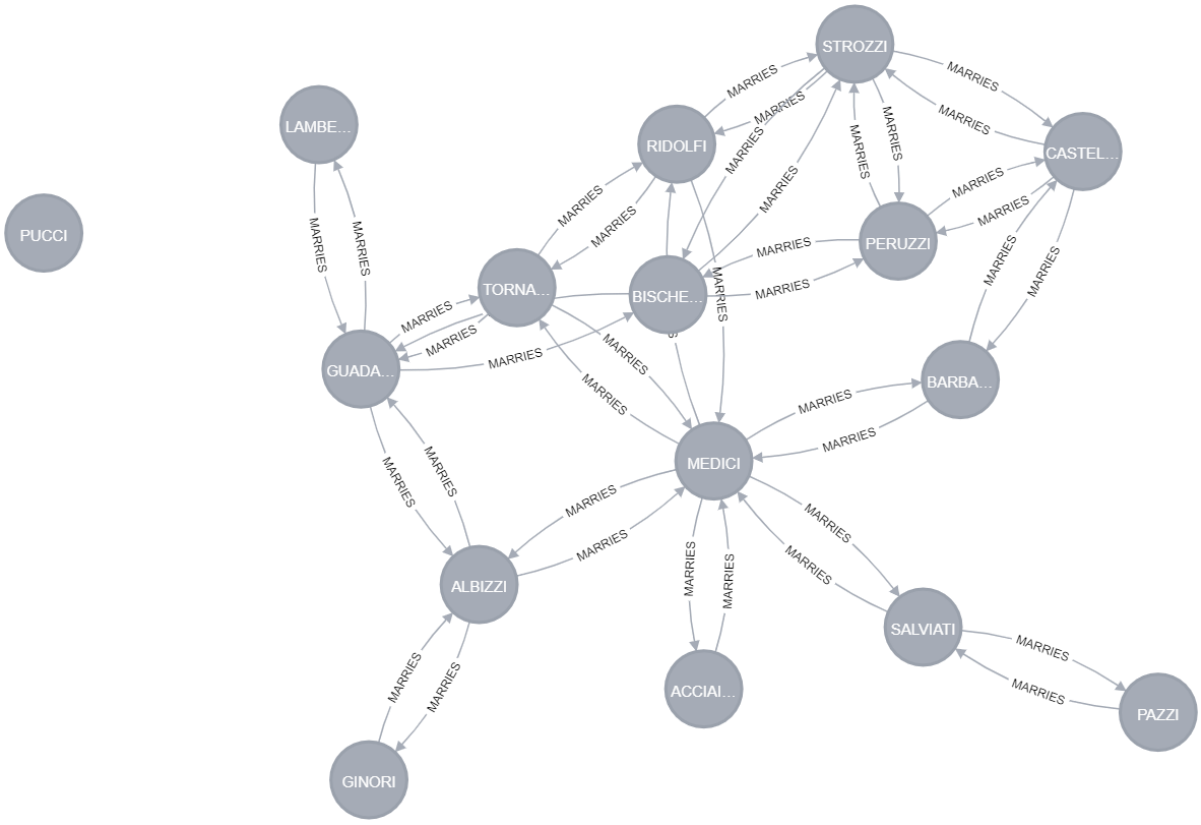
UNIT CODE: B9AI101

UNIT LEADER: TERRI HOARE

DATE OF SUBMISSION: 23 APRIL 2023

Submitter: Saiful Islam
Student ID: 10634911
Email: 10634911@mydbs.ie

Exercise-1: Cypher queries for data and relationship on marriage relationship among Florentine families is built based on the given graph. The queries are submitted in the separate cypher file named **exercise_1.cypher.** Here is the following graph after running the query: MATCH(n) RETURN n;



Computing measure of Centrality

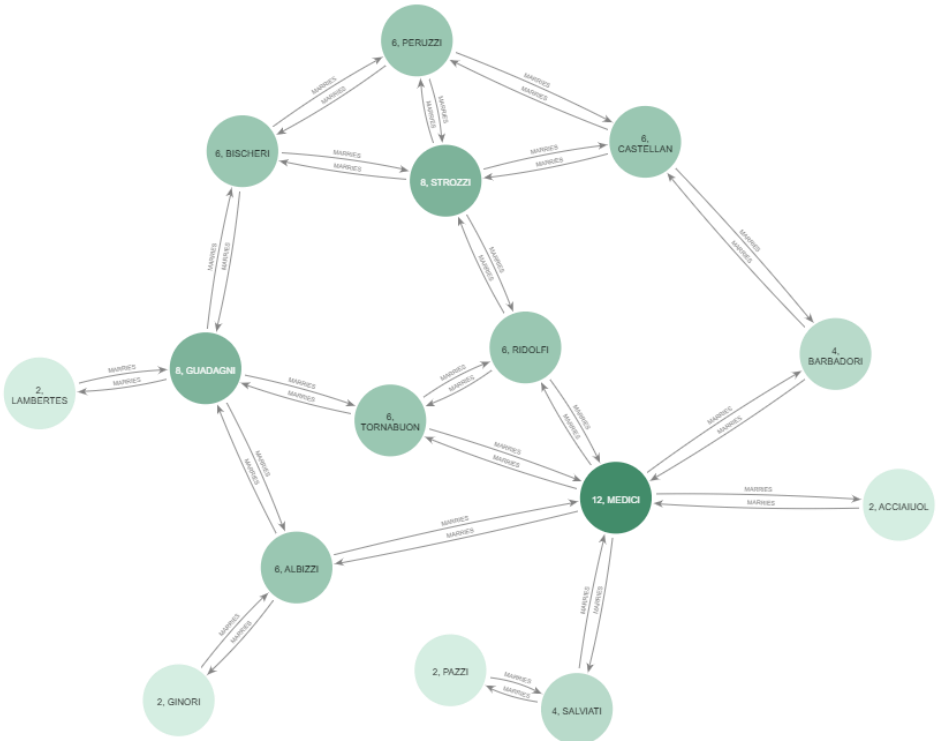
Since the relationship is bi-directional, the orientation of graph has to be **'UNDIRECTED'**. In order to measure centrality, a graph is created with the following cypher query:

CALL gds.graph.create(' marriageRelationGraph', 'Family', {MARRIES: {orientation: 'UNDIRECTED'}});

- **Degree Centrality**: Degree centrality is a measure in a network that counts how many relationships a node has. The following query creates a degree centrality graph on the database:

  CALL gds.degree.write('marriageRelationGraph', {writeProperty: 'degree_centrality_marriage'});

  After applying the degree centrality algorithm on this data from Neo4j Bloom, the following graph can be obtained with color gradient (higher value has deeper color):



The degree centrality scores for each node with Score descending order:

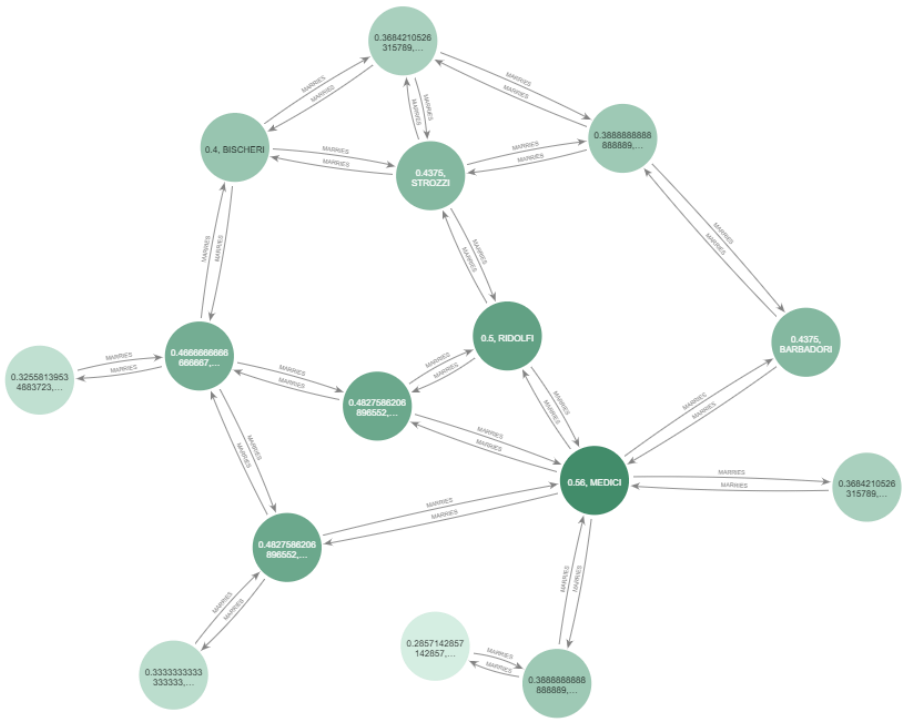
Family	Score
"MEDICI"	12
"STROZZI"	8
"GUADAGNI"	8
"CASTELLAN"	6
"PERUZZI"	6
"RIDOLFI"	6
"TORNABUON"	6
"BISCHERI"	6
"ALBIZZI"	6
"SALVIATI"	4
"BARBADORI"	4
"PAZZI"	2
"LAMBERTES"	2
"GINORI"	2
"ACCIAIUOL"	2
"PUCCI"	0

Here, it can be observed that the family “MEDICI” has the highest degree centrality score-12, followed by “STROZZI” & “GUADAGNI” each having score-8

- **Closeness Centrality**: Closeness centrality is a measure that detects the nodes having the shortest path to all other nodes. The following cypher code generates closeness centrality into the 'marriageRelationGraph':

```
CALL gds.alpha.closeness.write('marriageRelationGraph', {writeProperty: 'closeness_centrality_marriage'})
YIELD nodes, writeProperty;
```

After applying the degree centrality algorithm on this data from Neo4j Bloom, the following graph can be obtained with color gradient (higher value has deeper color):



The closeness centrality scores for each node with Score descending order:

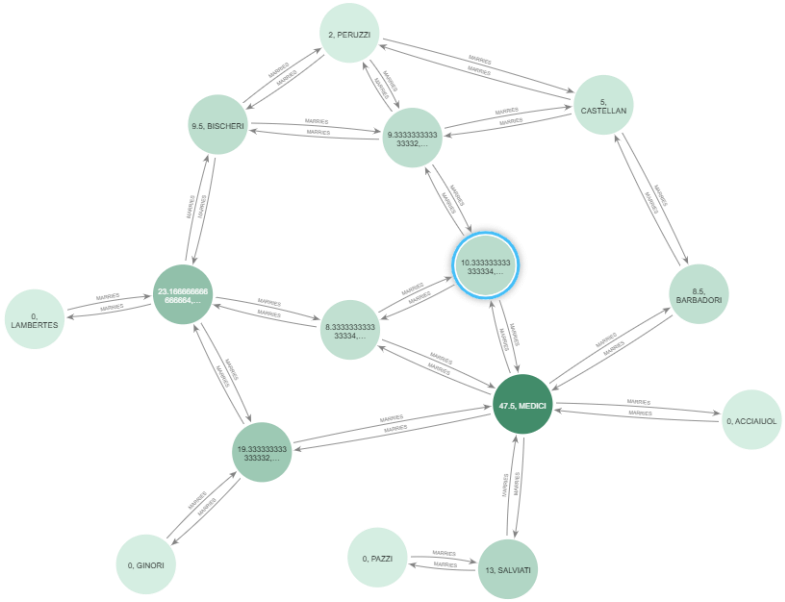
Family	Score
"MEDICI"	0.56
"RIDOLFI"	0.5
"TORNABUON"	0.482758621
"ALBIZZI"	0.482758621
"GUADAGNI"	0.466666667
"BARBADORI"	0.4375
"STROZZI"	0.4375
"BISCHERI"	0.4
"SALVIATI"	0.388888889
"CASTELLAN"	0.388888889
"PERUZZI"	0.368421053
"ACCIAIUOL"	0.368421053
"GINORI"	0.333333333
"LAMBERTES"	0.325581395
"PAZZI"	0.285714286
"PUCCI"	0

Here, it can be observed that the family "MEDICI" has the highest closeness centrality scores with value 0.56, followed by "RIDOLFI" with score 0.5 and "TORNABUON" with score 0.482758621

- **Betweenness Centrality**: Betweenness centrality is a detecting method in a network which counts how many shortest paths pass through a node. The following cypher code generates betweenness centrality into the 'marriageRelationGraph':

```
CALL gds.betweenness.write('marriageRelationGraph', { writeProperty: 'betweenness_centrality_marriage' })
YIELD centralityDistribution, nodePropertiesWritten
RETURN centralityDistribution.min AS minimumScore, centralityDistribution.mean AS meanScore, nodePropertiesWritten;
```

After applying the betweenness centrality algorithm on this data from Neo4j Bloom, the following graph can be obtained with color gradient (higher value has deeper color):



The betweenness centrality scores for each node with Score descending order:

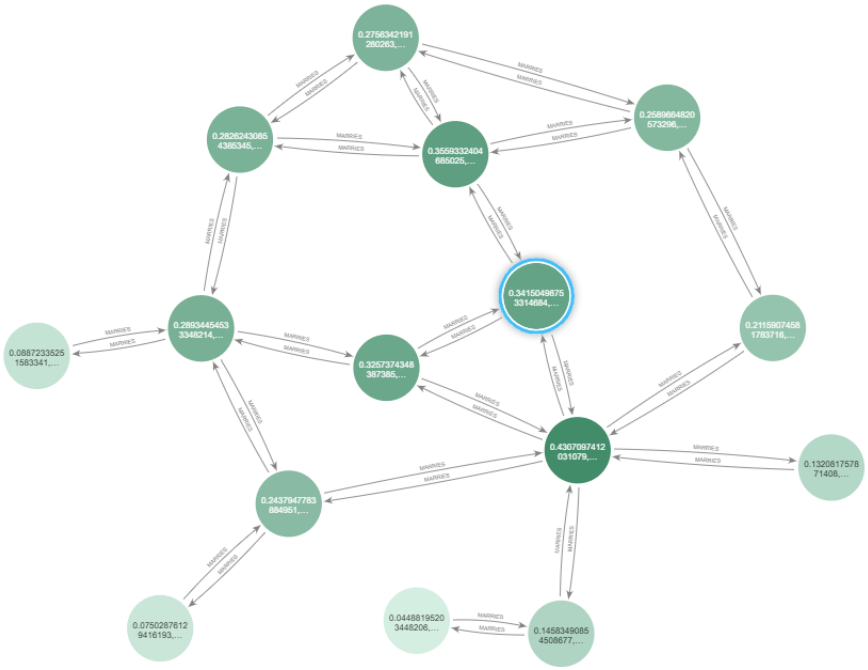
Family	Score
"MEDICI"	47.5
"GUADAGNI"	23.16666667
"ALBIZZI"	19.33333333
"SALVIATI"	13
"RIDOLFI"	10.33333333
"BISCHERI"	9.5
"STROZZI"	9.333333333
"BARBADORI"	8.5
"TORNABUON"	8.333333333
"CASTELLAN"	5
"PERUZZI"	2
"PAZZI"	0
"LAMBERTES"	0
"GINORI"	0
"ACCIAIUOL"	0
"PUCCI"	0

Here, it can be observed that the family “MEDICI” has the highest betweenness centrality scores with value 47.5 followed by “GUADANGI” having value 23.16666667 & “ALBIZZI” having value 19.33333333.

- **Eigenvector Centrality**: Eigenvector centrality is a algorithm that computes the influence of a node in a network. The following cypher code generates eigenvector centrality into the 'marriageRelationGraph':

```
CALL gds.eigenvector.write('marriageRelationGraph', {maxIterations: 20,
  writeProperty: 'eigen_centrality_marriage' }) YIELD nodePropertiesWritten, ranIterations;
```

After applying the eigenvector centrality algorithm on this data from Neo4j Bloom, the following graph can be obtained with color gradient (higher value has deeper color):



The eigenvector centrality scores for each node with Score descending order:

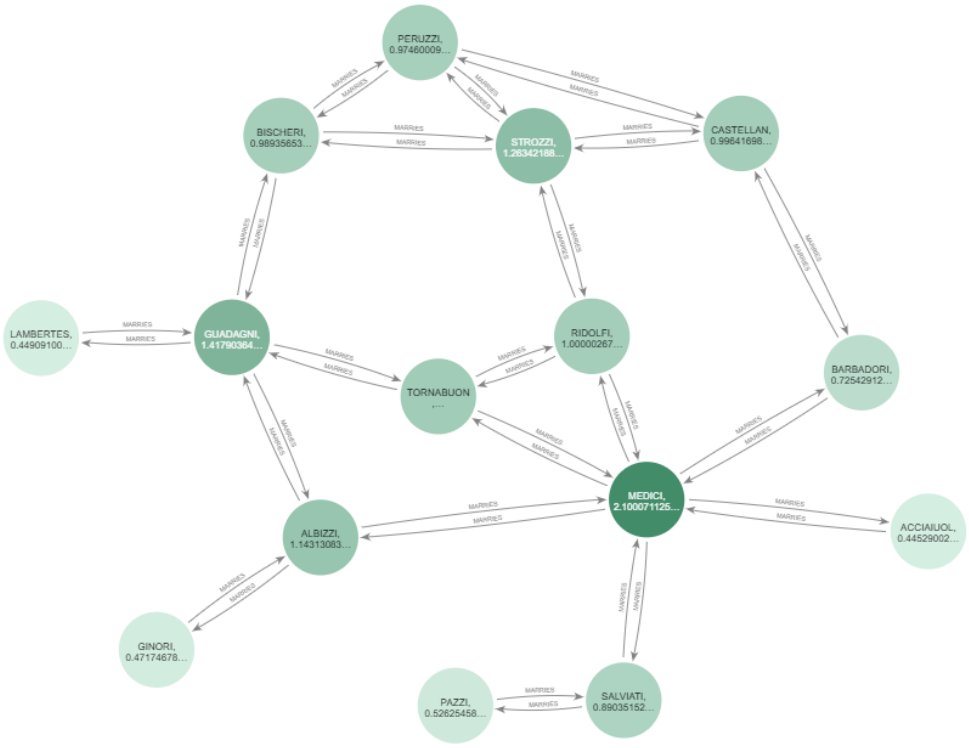
Family	Score
"MEDICI"	0.430709741
"STROZZI"	0.35593324
"RIDOLFI"	0.341504988
"TORNABUON"	0.325737435
"GUADAGNI"	0.289344545
"BISCHERI"	0.282624309
"PERUZZI"	0.275634219
"CASTELLAN"	0.258966482
"ALBIZZI"	0.243794778
"BARBADORI"	0.211590746
"SALVIATI"	0.145834909
"ACCIAIUOL"	0.132081758
"LAMBERTES"	0.088723353
"GINORI"	0.075028761
"PAZZI"	0.044881952
"PUCCI"	6.10E-148

Here, it can be observed that the family "MEDICI" has the highest eigenvector centrality score with value 0.430709741 followed by "STROZZI" having value 0.35593324 and "RIDOLFI" having value 0.341504988.

- **PageRank**: PageRank algorithm computes the importance of each node in a network by analyzing the linked neighbors. The following cypher code generates PageRank into the 'marriageRelationGraph':

```
CALL gds.pageRank.write('marriageRelationGraph', {
maxIterations: 20, dampingFactor: 0.85, writeProperty: 'pagerank_marriage'})
YIELD nodePropertiesWritten, ranIterations;
```

After applying the PageRank algorithm on this data from Neo4j Bloom, the following graph can be obtained with color gradient (higher value has deeper color):



The PageRank scores for each node with Score descending order:

Family	Score
"MEDICI"	2.100071126
"GUADAGNI"	1.417903641
"STROZZI"	1.263421885
"ALBIZZI"	1.14313084
"TORNABUON"	1.025540194
"RIDOLFI"	1.000002677
"CASTELLAN"	0.996416982
"BISCHERI"	0.989356539
"PERUZZI"	0.974600099
"SALVIATI"	0.890351526
"BARBADORI"	0.725429129
"PAZZI"	0.526254582
"GINORI"	0.471746784
"LAMBERTES"	0.449091008
"ACCIAIUOL"	0.445290023
"PUCCI"	0.15

Here, it can be observed that the family “MEDICI” has the highest PageRank score with value 2.100071126 followed by “GUADAGNI” having value 1.417903641 and “STROZZI” having value 1.263421885

Conclusions

Since, each family is connected with marriage relationship in the network, all the nodes are bi-directional. In all of the cases, “MEIDICI” has the highest value scores for all algorithms. Besides “MEIDICI”, "STROZZI" & "GUADAGNI" also have higher scores in all the algorithms.

Exercise-3: Findings on analyzing data based on CRISP-DM mythology

For data analyzing, Crime Investigation dataset was chosen. Link of the dataset: https://github.com/neo4j-graph-examples/pole

It is a public data source freely available from http://data.gov.uk

Phrases of CRISP-DM

Business Understanding

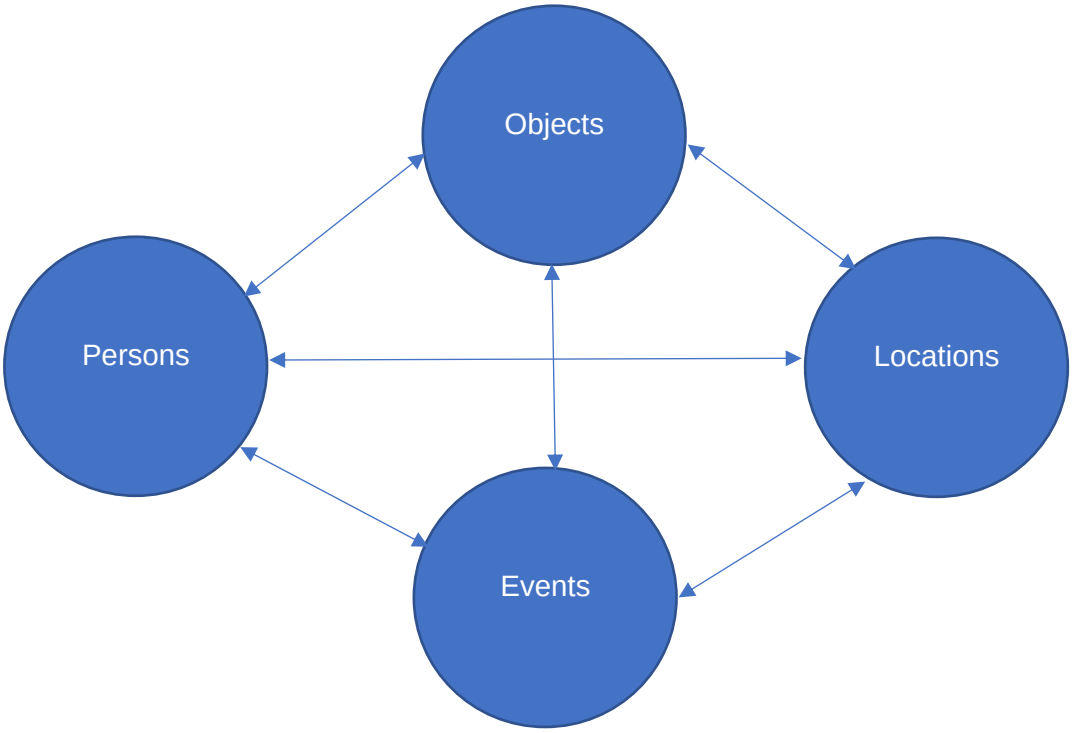
Crime Investigation data is a investigate dataset investigated by law enforcement agencies. This dataset maintains data of area/location-wise that means, in a certain area:

- Postal code of the area
- How many crimes happened in the area
- How many criminals living in the area
- Name and the details of the criminals
- Name and the details of residents living in the area
- Residents related to each of the neighbor
- Details of the officers
- List of officers investigating crimes in this area.

Data Understanding

The crime investigation dataset can be defined as POLE model:

(**P**erson+**O**bject+**L**ocation+**E**vent)



This POLE model includes:

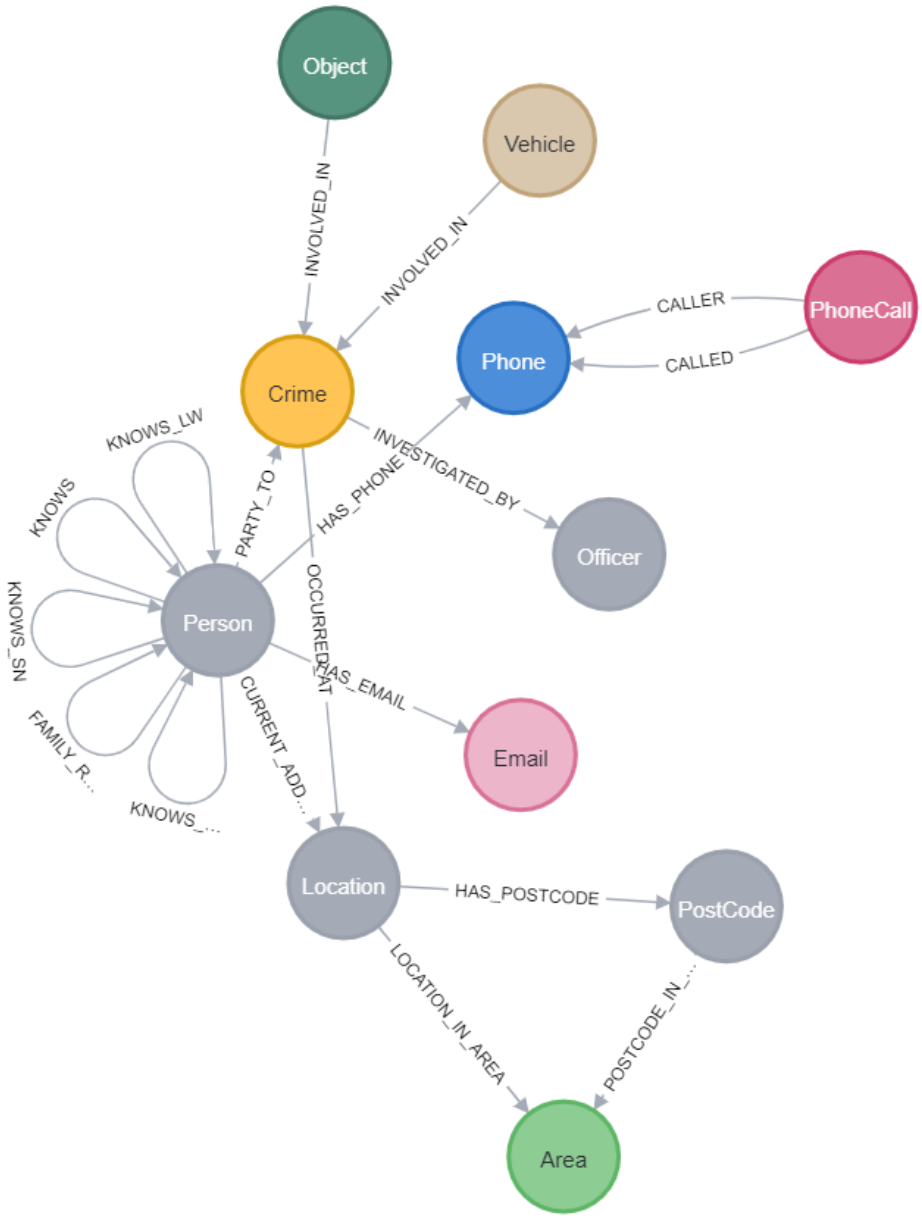
1. Policing
2. Counter Terrorism
3. Immigration
4. Child Protection
5. Insurance Fraud Investigation

Data Preparation

In this phase, a local database is created using Neo4j desktop community edition having the database version: 5.5.0. After that, from this GitHub link: https://github.com/neo4j-graph-examples/pole the zip file is downloaded and then extracted to local drive. After that, from data folder, **pole-50.dump** is imported into Neo4j and then the data is imported into selected database. Then in the jupyter notebook, necessary python library is loaded. After that, a database connection is established with jupyter notebook.

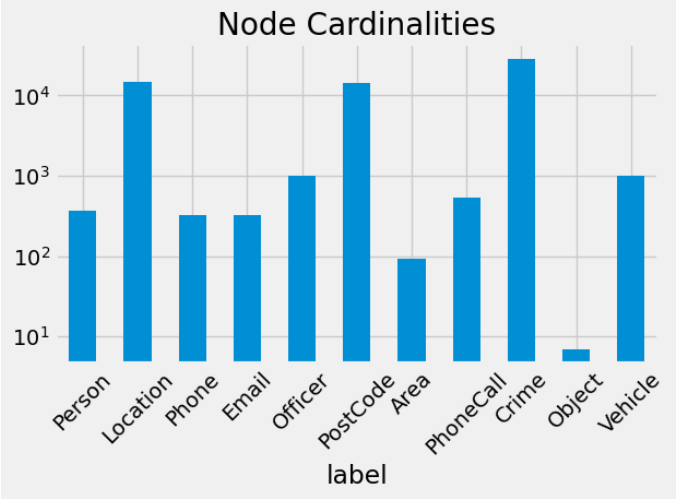
Exploratory Data Analysis EDA:

The visualization for this database is given below:



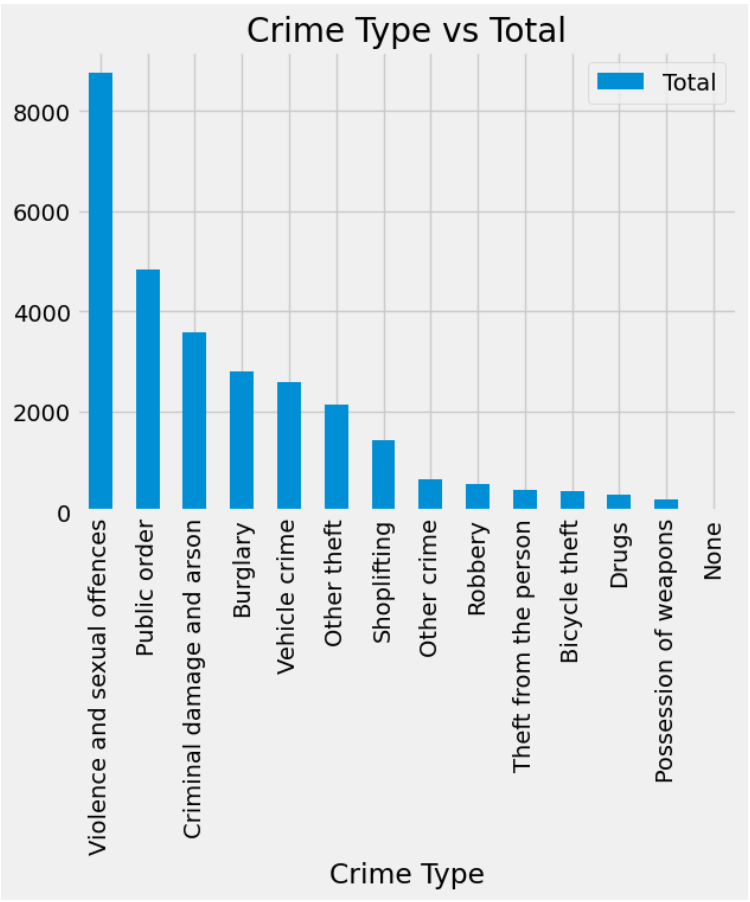
No of Nodes:

Label	Nodes
Object	7
Area	93
Phone	328
Email	328
Person	369
PhoneCall	534
Officer	1000
Vehicle	1000
PostCode	14196
Location	14904
Crime	28762

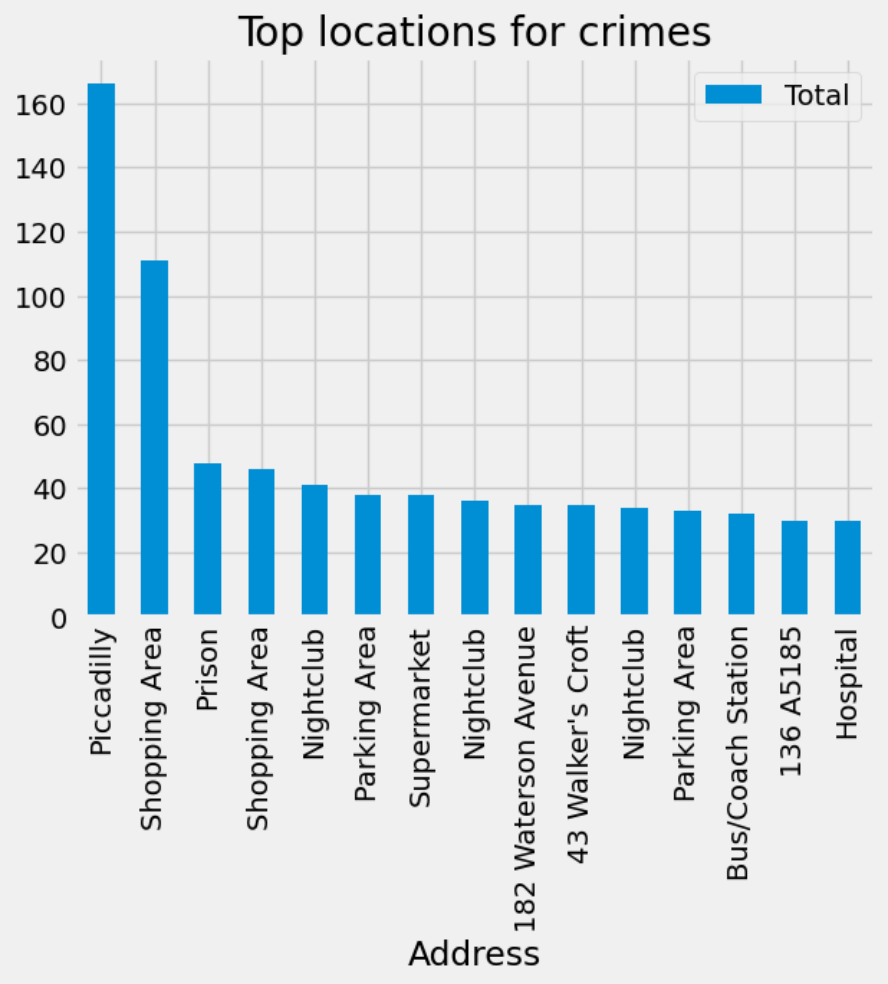


Total Crimes

Crime	Total
Violence and sexual offences	8765
Public order	4839
Criminal damage and arson	3587
Burglary	2807
Vehicle crime	2598
Other theft	2140
Shoplifting	1427
Other crime	651
Robbery	541
Theft from the person	423
Bicycle theft	414
Drugs	333
Possession of weapons	236
None	1

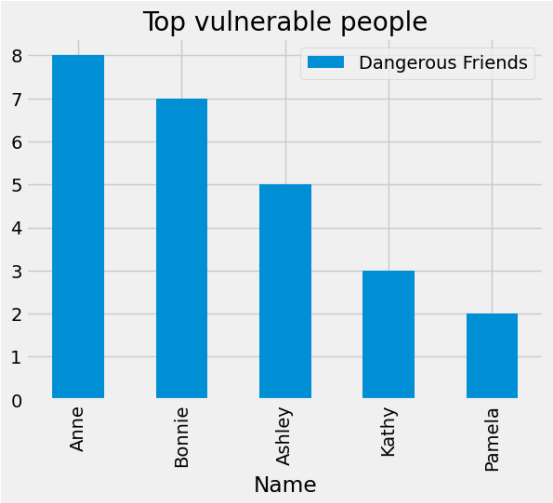


Top locations for Crimes:



Top vulnerable people:

Name	Surname	Dangerous Friends
Anne	Freeman	8
Bonnie	Gilbert	7
Ashley	Robertson	5
Kathy	Wheeler	3
Pamela	Gibson	2



Modeling

In this phase at first a graph is created named ‘social’. After that the following centrality graph algorithm has been applied into the graph

- Degree Centrality
- Closeness Centrality
- Betweenness Centrality
- Eigenvector Centrality
- PageRank

It can be observed that, in most of the algorithms, **Annie Duncan** has the highest scores since this person is connected to maximum no of dangerous friends who have criminal history in the past.

Furthermore, Community detection algorithm has been applied. It can be observed that **Joe** has the lowest community ID having value 0 and **Jessica** has the highest community ID having value 212.

After that, label propagation algorithm has been applied to the graph. Here it can be observed that **Carl** has the lowest community score having value 5 whereas **Joe** has the highest community score having value 61533.

Evaluation

After performing evaluation on this data, it can be said that some relationship is safer than others like **Family** or **Lives with**. It is also observed that some crimes are considered more dangerous than others like **Violence and Sexual Offences** is more dangerous crimes than **shoplifting** or other related crimes.

It is also found that people who has more dangerous friends have the tendency to indulge in crimes in the future.

Furthermore, some location has higher crime rates than others. Those locations are considered dangerous.

Deployment

The applied graph algorithms can be applied to similar law enforcement organizations those who deals with the followings:

- Child trafficking
- Drug trafficking
- Immigration
- Social Service
- Fraud detection

References

1. https://neo4j.com/docs/graph-data-science/current/algorithms/
2. https://neo4j.com/docs/bloom-user-guide/current/
3. Graph Algorithms: Practical Examples in Apache Spark and Neo4j (2019/05/16)[O'REILLY] --Needham, Mark
4. https://github.com/neo4j-graph-examples/pole
5. https://www.data.gov.uk
6. https://pandas.pydata.org/docs/user_guide/10min.html
7. https://www.datascience-pm.com/crisp-dm-2/
8. https://en.wikipedia.org/wiki/Cross-industry_standard_process_for_data_mining