Graph Algorithm Application

**Unit Title**: Data Mining

**Unit Code**: B9AI101

**Unit Leader**: Terri Hoare

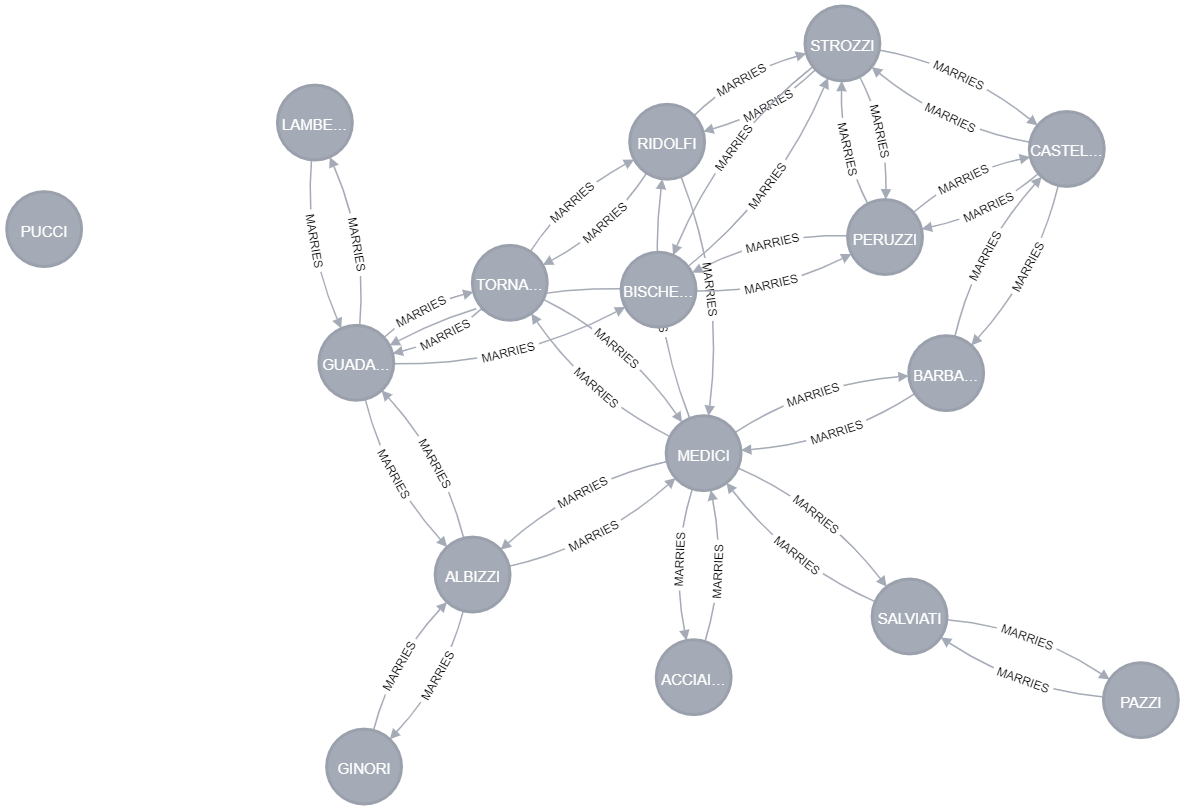
**Date of Submission**: 23 April 2023

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

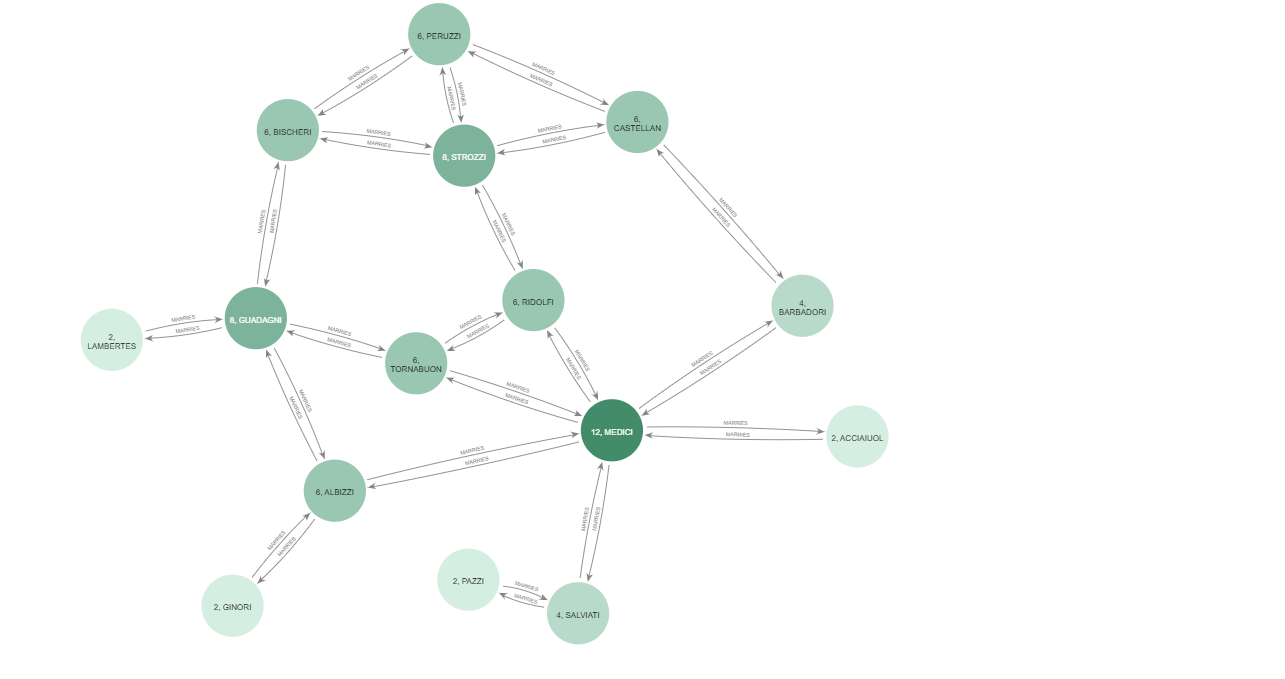
Since the relationship is bi-directional, the orientation of graph has to be **'UNDIRECTED'**. In order to measure centrality, a graph is created with the following cypher query:

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:

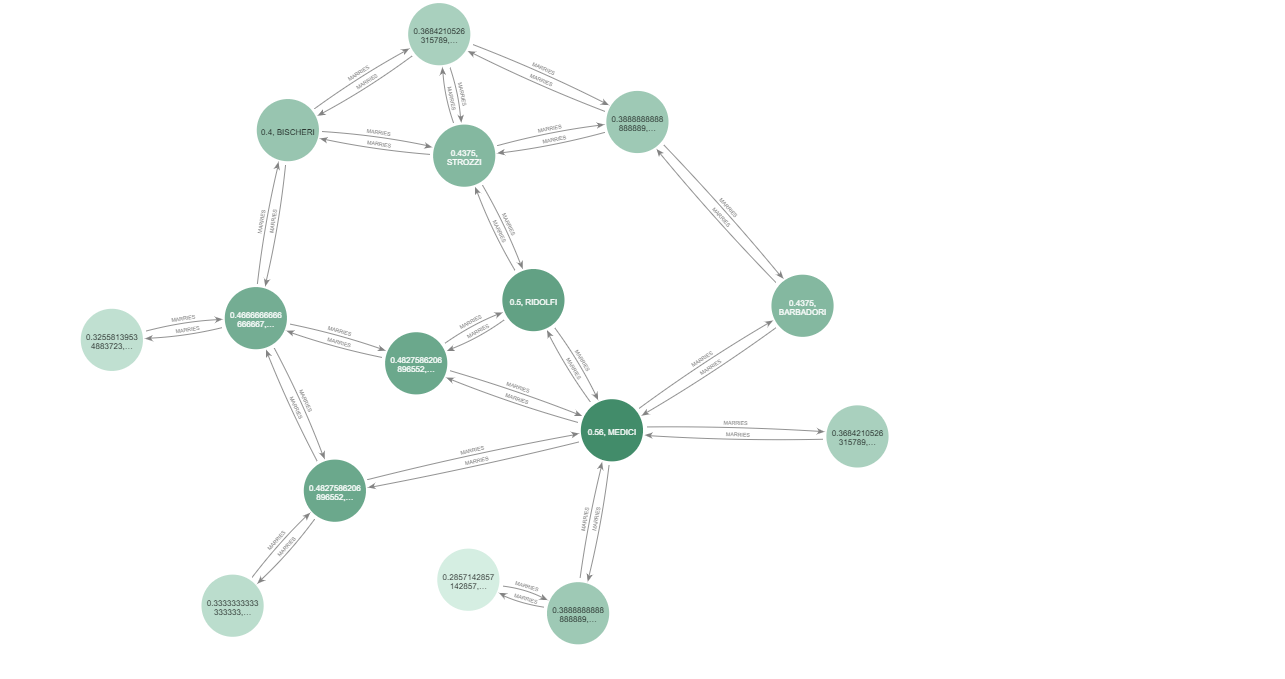
Here, it can be observed that the family “MEDICI” has the highest degree centrality score-12, followed by “STROZZI” & “GUADAGNI” each having score-8

|  |  |
| --- | --- |
| **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 |
|  |  |

* 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): 

|  |  |
| --- | --- |
| **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 |

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

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

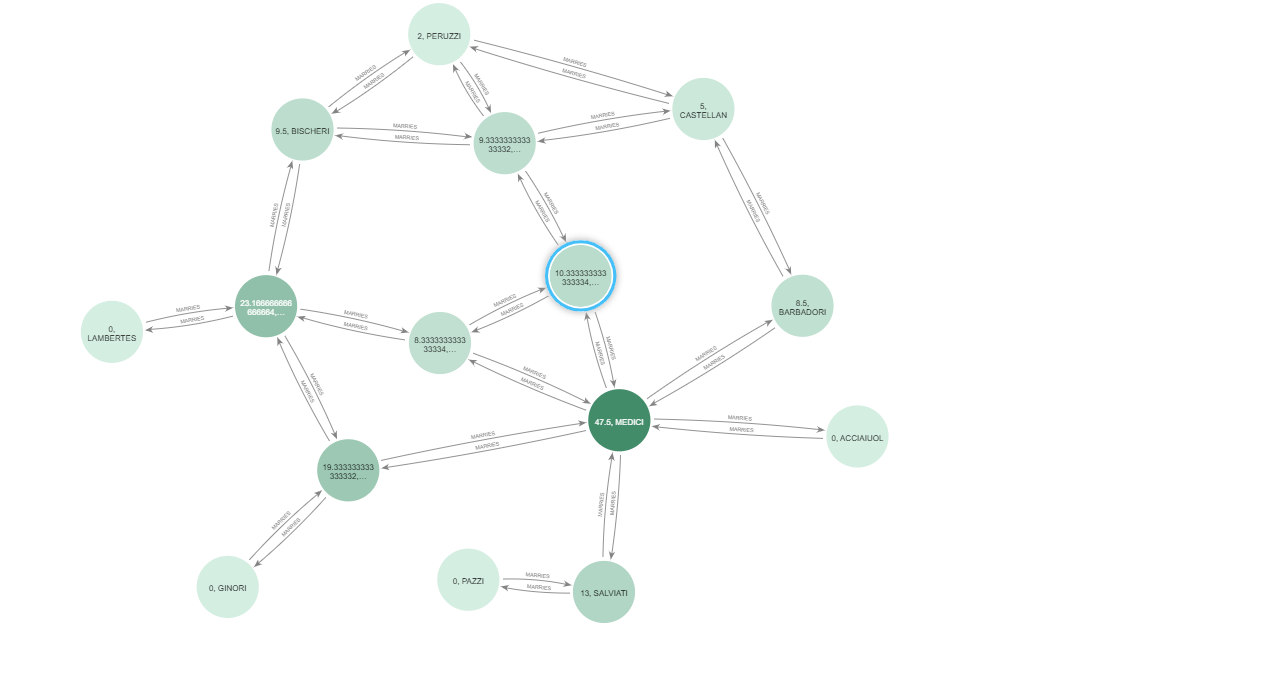
|  |  |
| --- | --- |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |

* 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:

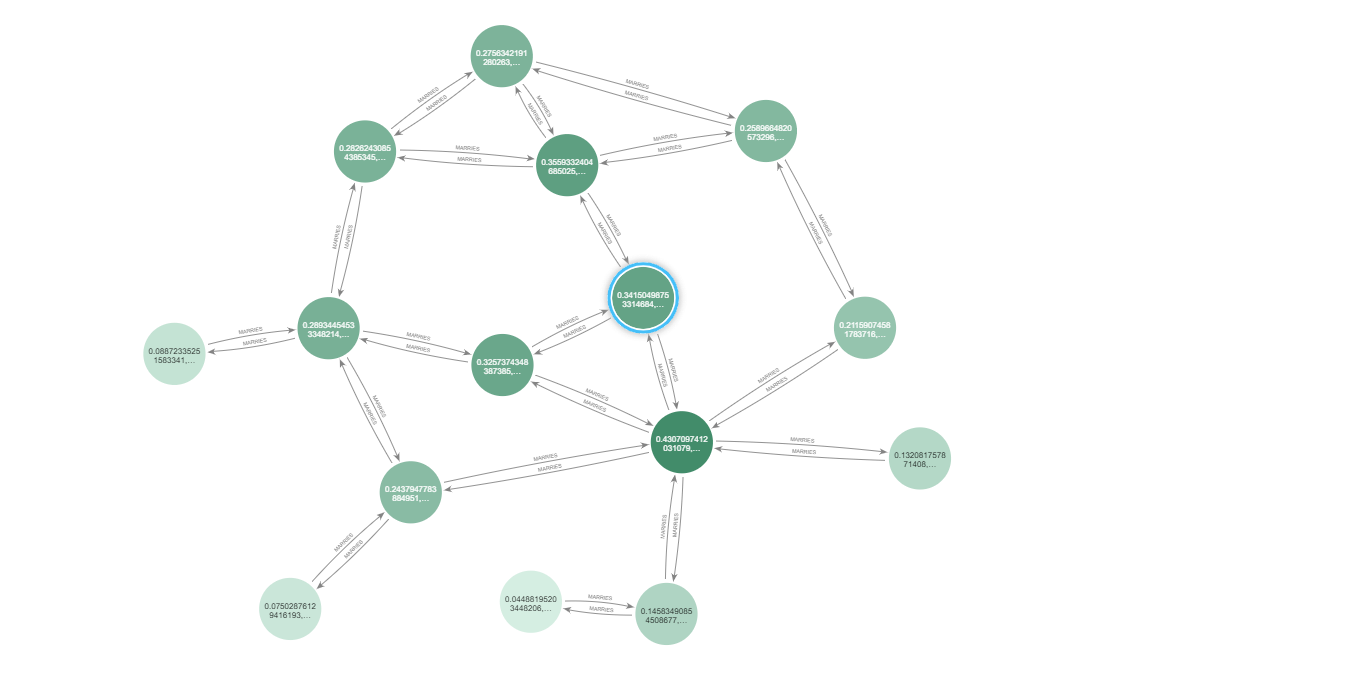
Here, it can be observed that the family “MEDICI” has the highest betweenness centrality scores with value 47.5 followed by “GUADANGI” having value 23.16666667 & “ALBIZZI” having value 19.33333333.

|  |  |
| --- | --- |
| **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 |

* 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:

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.

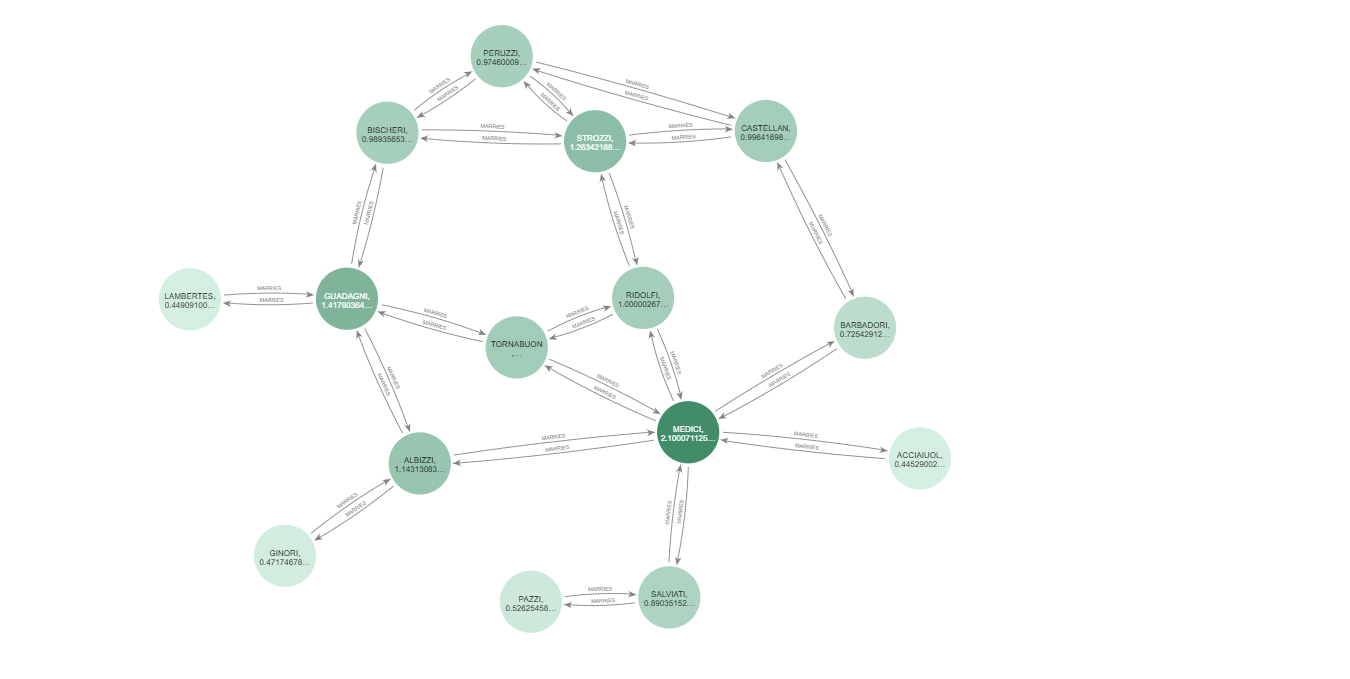
|  |  |
| --- | --- |
| **Family** | **Score** |
| "MEDICI" | 0.430709741 |
| "STROZZI" | 0.35593324 |
| "RIDOLFI" | 0.341504988 |
| "TORNABUON" | 0.325737435 |
| "GUADAGNI" | 0.289344545 |
| "BISCHERI" | 0.282624309 |
| "PERUZZI" | 0.275634219 |
| "CASTELLAN" | 0.258966482 |
| "ALBIZZI" | 0.243794778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

* 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:

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

|  |  |
| --- | --- |
| **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 |

## 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**:

## 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