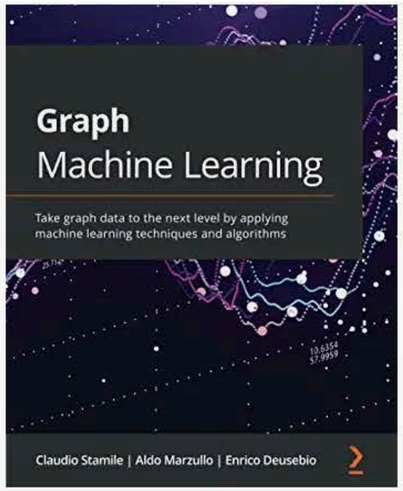
CLAUDIO STAMILE - 23/06/2022 - CLAUDIOSTAMILE@GMAIL.COM

A GENTLE INTRODUCTION TO GRAPH MACHINE LEARNING IN PYTHON

ABOUT ME

- Computer Scientist di formazione
- Data Scientist di adozione (💰)
- Un paio di PhD
- Un altro paio di brevetti
- Tipo un'ottantina di pubblicazioni

ABOUT ME



Graph Machine Learning: Take graph data to the next level by applying machine learning techniques and algorithms

by Claudio Stamile , Aldo Marzullo, et al. | Jun 25, 2021

★★★★☆ 46

Paperback

$44.99

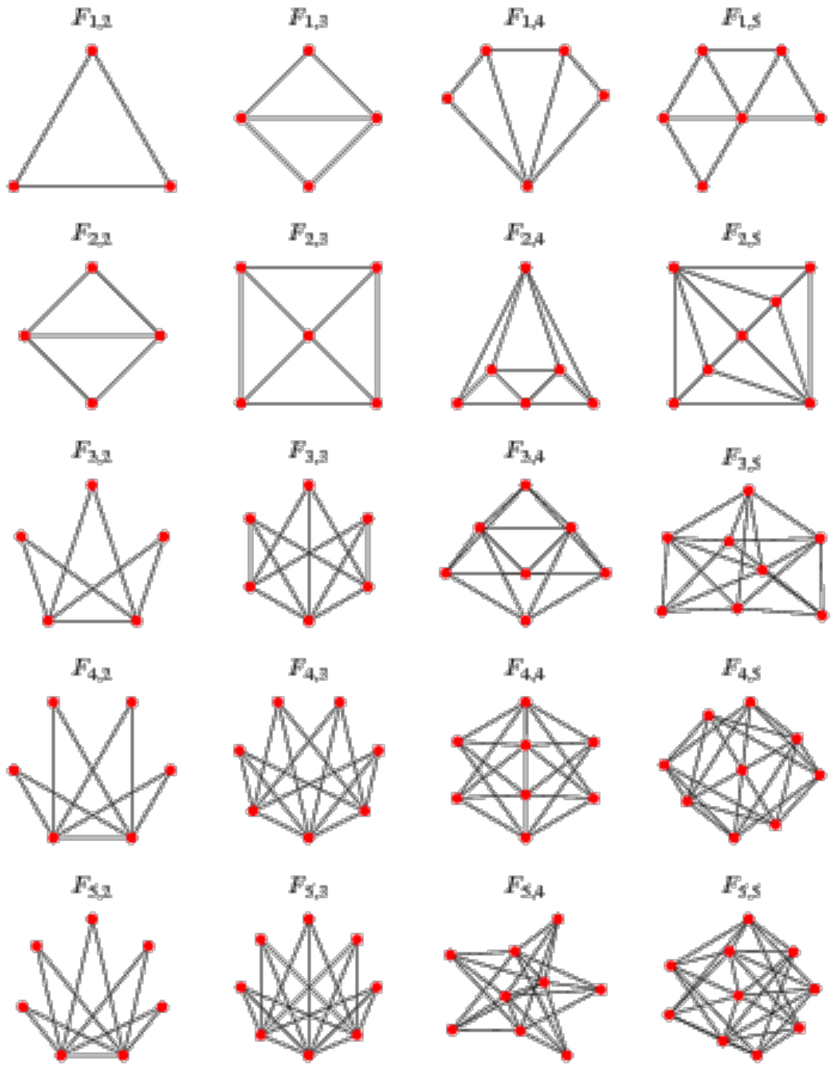
Ships to Italy

More Buying Choices

$39.95 (12 used & new offers)

Kindle

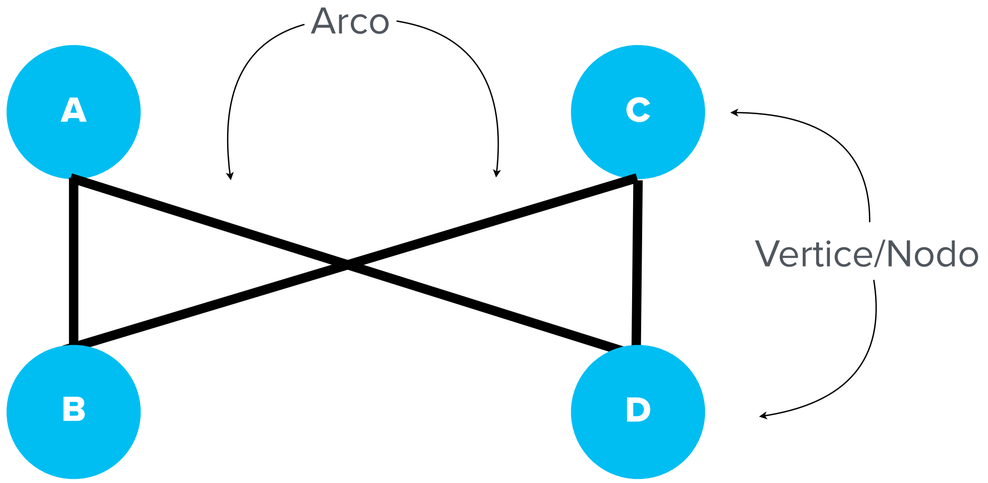
AGENDA



- Grafi ? Parliamone
- Python ❤️ grafi
- Come si usano i grafi in data science

AEROPORTI E GRAFI, UNA SIMILITUDINE FACILE

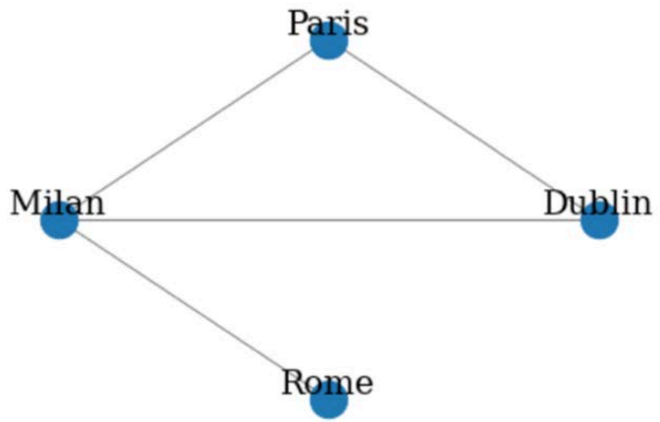
http://www.martingrandjean.ch/connected-world-air-traffic-network/



- Ciascun aeroporto è un punto (Nodo/Vertice)
- Ciascuna connessione tra aeroporti è una linea (Arco)

PYTHON ❤️ I GRAFI

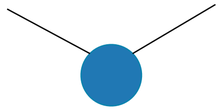
G=(V,E)
V = {Paris, Milan, Dublin, Rome}
E={{Paris, Milan}, {Paris, Dublin}, {Milan, Dublin}, {Milan, Rome}}




```
import networkx as nx
G = nx.Graph()
V = {'Dublin', 'Paris', 'Milan', 'Rome'}
E = [('Milan','Dublin'), ('Milan','Paris'),
('Paris','Dublin'), ('Milan','Rome')]
G.add_nodes_from(V)
G.add_edges_from(E)
```

```
nx.to_pandas_adjacency(G)
```

	Paris	Rome	Dublin	Milan
Paris	0.0	0.0	1.0	1.0
Rome	0.0	0.0	0.0	1.0
Dublin	1.0	0.0	0.0	1.0
Milan	1.0	1.0	1.0	0.0

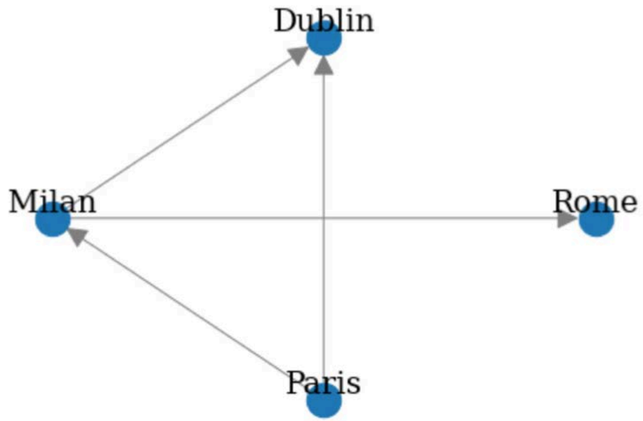


Node Degree: G.degree('Paris')

Graph Order: G.number_of_nodes()
Graph Size: G.number_of_edges()

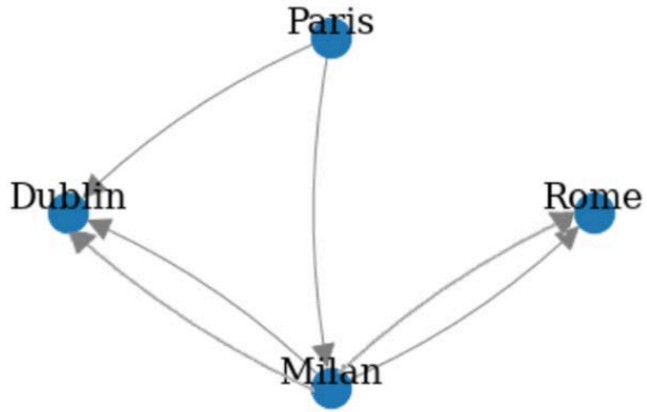
UNO, NESSUNO, CENTOMILA GRAFI

- Directed Graph



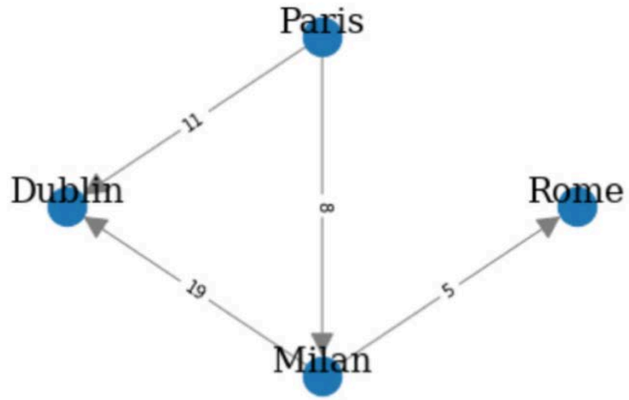

```
import networkx as nx
G = nx.DiGraph()
V = {'Dublin', 'Paris', 'Milan', 'Rome'}
E = [('Milan','Dublin'), ('Milan','Paris'),
('Paris','Dublin'), ('Milan','Rome')]
G.add_nodes_from(V)
G.add_edges_from(E)
```

- Multi Graph



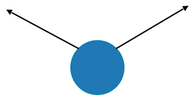

```
dmg = nx.MultiDiGraph()
umg = nx.MultiGraph()
V = {'Dublin', 'Paris', 'Milan', 'Rome'}
E = [('Milan','Dublin'), ('Milan','Dublin'), ('Paris','Milan'),
('Paris','Dublin'), ('Milan','Rome'), ('Milan','Rome')]
dmg.add_nodes_from(V)
umg.add_nodes_from(V)
dmg.add_edges_from(E)
umg.add_edges_from(E)
```

- Weighed Graph

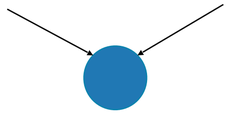



```
G = nx.DiGraph()
V = {'Dublin', 'Paris', 'Milan', 'Rome'}
E = [('Milan','Dublin', 19), ('Paris','Milan',
8), ('Paris','Dublin', 11), ('Milan','Rome', 5)]
G.add_nodes_from(V)
G.add_weighted_edges_from(E)
```

Out Degree



In Degree



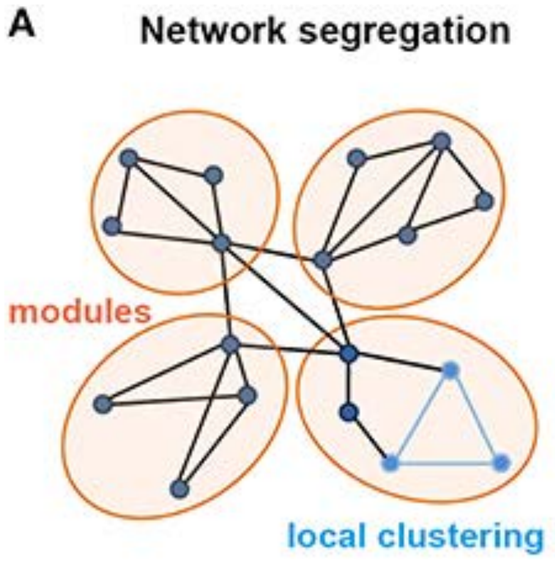
SÌ OK, MA COME TOLGO FUORI DEI NUMERI DAI GRAFI ? (METRICHE)

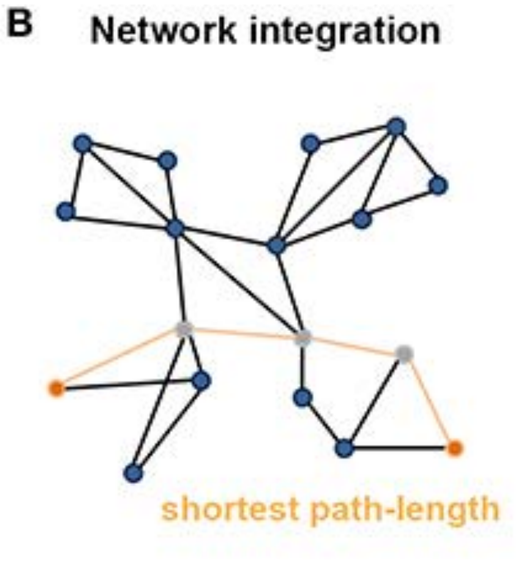
A. Quantifica la presenza di gruppi di nodi interconnessi, denominati con *community* o *modules*

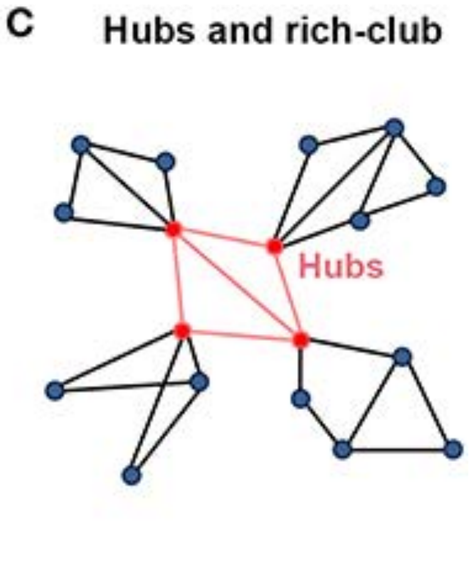
B. Misura come i nodi tendono ad essere interconnessi tra loro

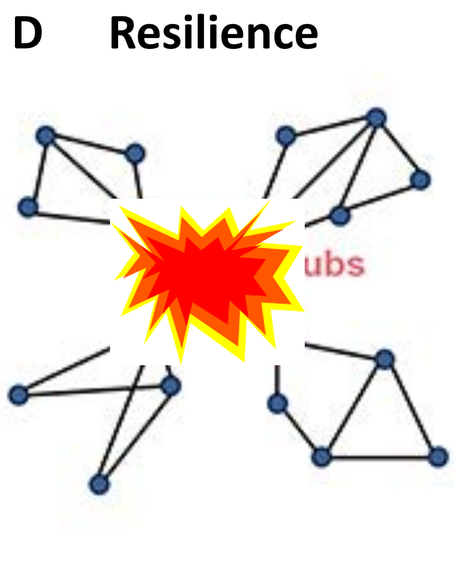
C. Quantifica l'importanza di ogni singolo nodo all'interno del grafo

D. Misura quanto un grafo è capace di adattarsi e di trasmettere l'informazione quando alcune condizioni critiche (perdita di nodi o archi) accadono





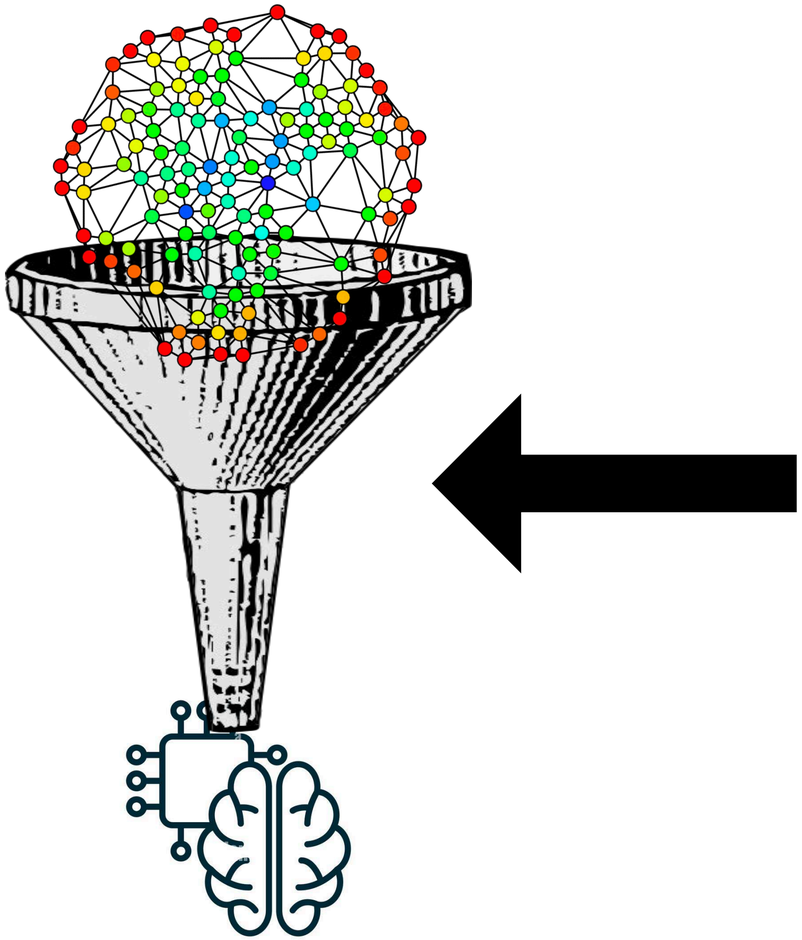




LE METRICHE SUI GRAFI SONO UN SACCO

1. VertexCount
2. EdgeCount
3. VertexDegree
4. VertexInDegree
5. VertexOutDegree
6. GraphDistance
7. GraphDistanceMatrix
8. VertexEccentricity
9. GraphRadius
10. GraphDiameter
11. Connectivity Measures
12. VertexConnectivity
13. EdgeConnectivity
14. Centrality Measures
15. ClosenessCentrality
16. BetweennessCentrality
17. DegreeCentrality
18. EigenvectorCentrality
19. KatzCentrality
20. PageRankCentrality
21. HITSCentrality
22. RadialityCentrality
23. StatusCentrality
24. EdgeBetweennessCentrality
25. GraphReciprocity
26. GlobalClusteringCoefficient
27. MeanClusteringCoefficient
28. LocalClusteringCoefficient
29. GraphAssortativity
30. VertexCorrelationSimilarity
31. MeanNeighborDegree
32. MeanDegreeConnectivity
33. VertexDiceSimilarity
34. VertexJaccardSimilarity
35. VertexCosineSimilarity

COS'È IL GRAPH MACHINE LEARNING

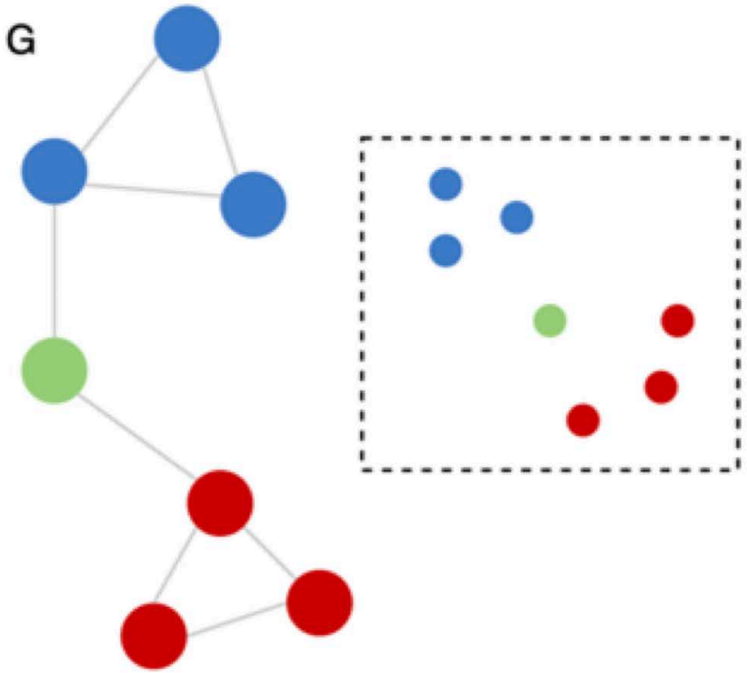


MACHINE LEARNING

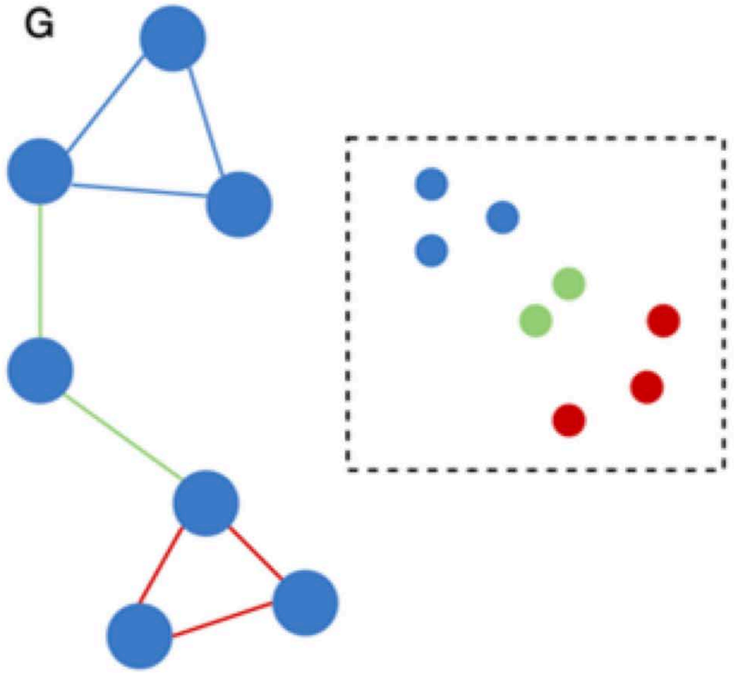
- Ottenere una rappresentazione compatta dei dati
- La rappresentazione deve essere in grado di conservare tutte le proprietà del grafo
- I nodi o i vertici possono inoltre contenere informazioni aggiuntive

COS'È IL GRAPH MACHINE LEARNING - A DIVERSI LIVELLI

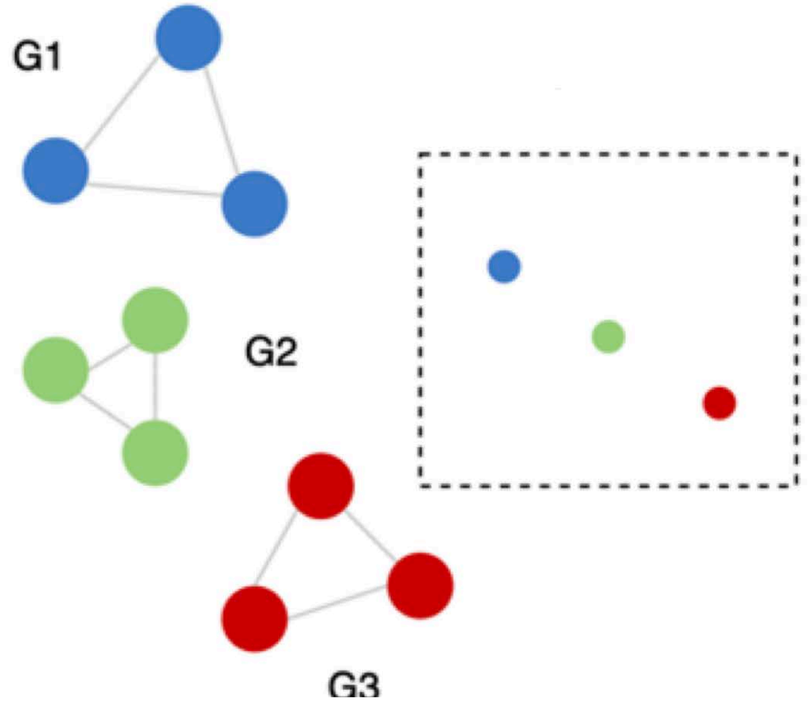
- Node Level Features



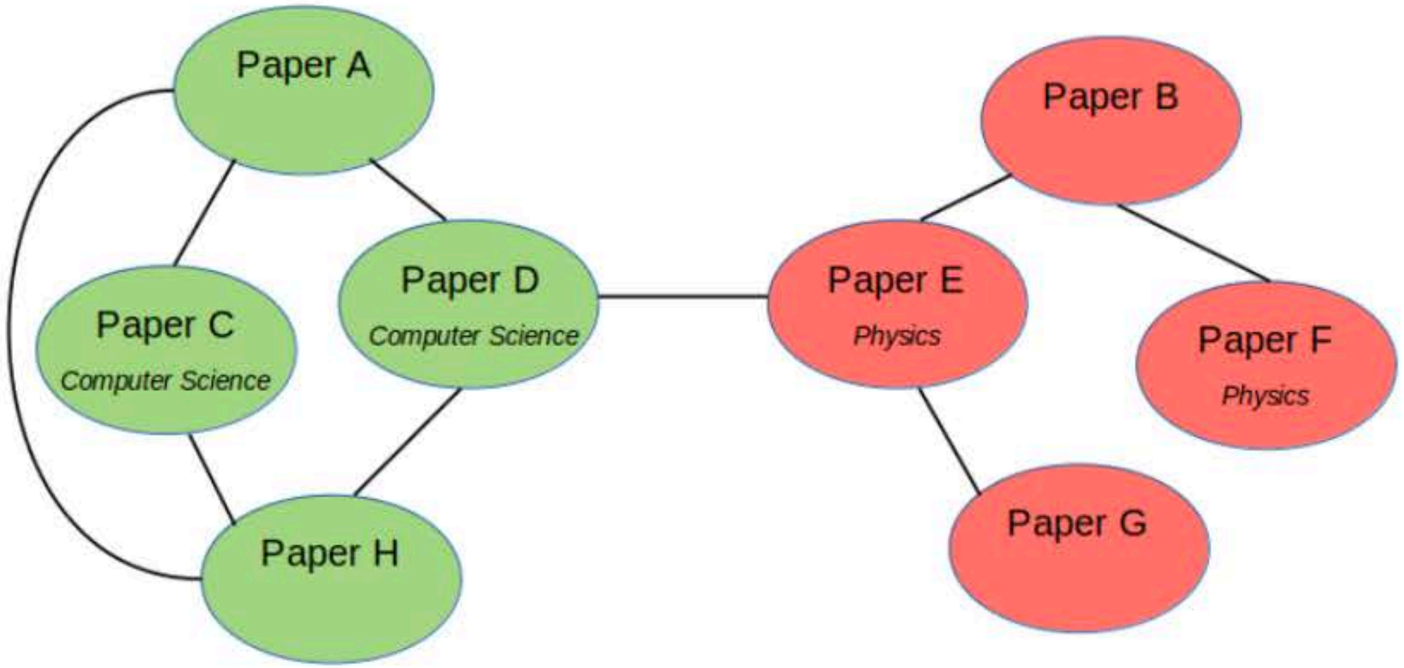
- Edge Level Features



- Graph Level Features



COS'È IL GRAPH MACHINE LEARNING - CLASSIFICARE NODI



Di quale argomento un articolo scientifico parla ?

COS'È IL GRAPH MACHINE LEARNING - CLASSIFICARE ARCHI

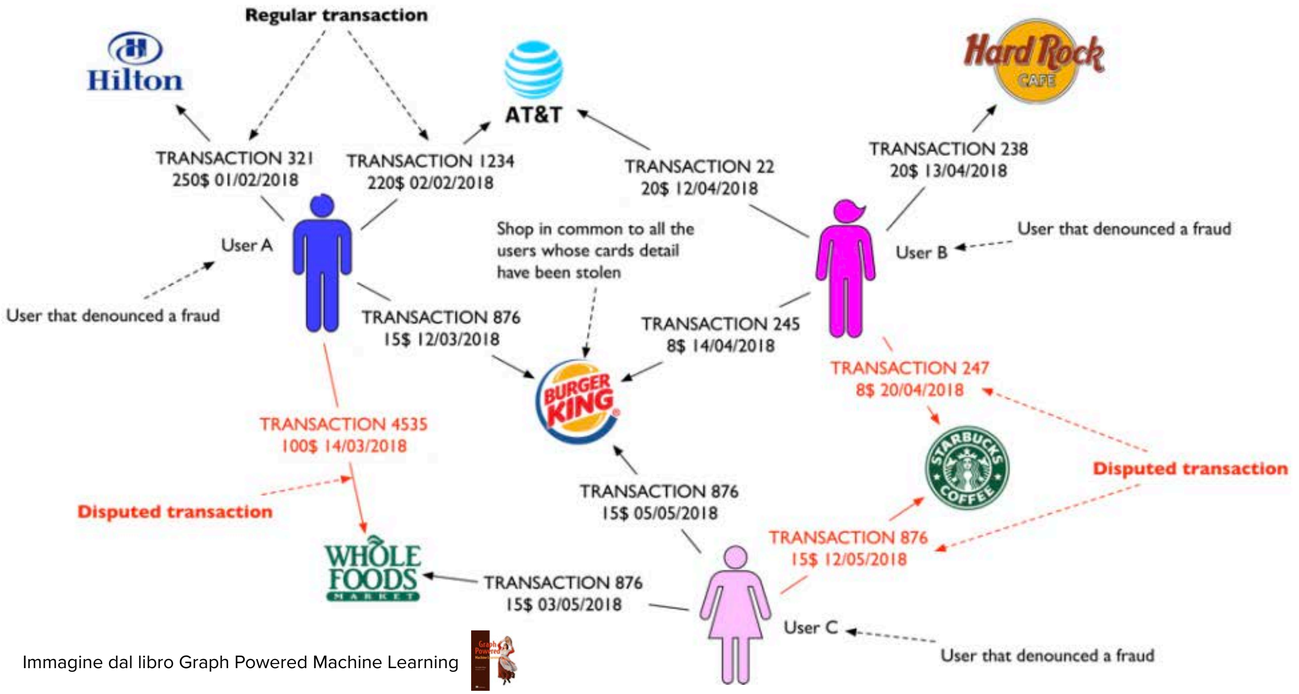
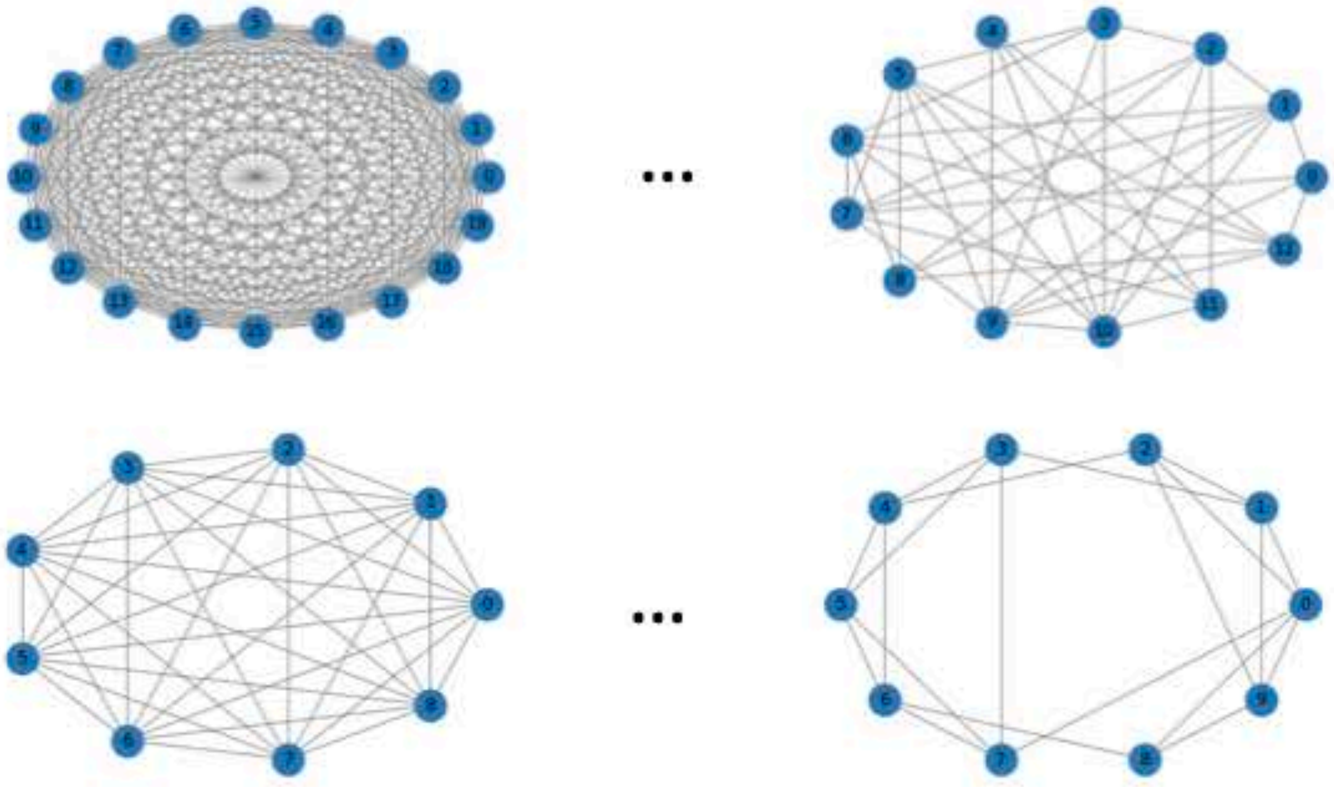


Immagine dal libro Graph Powered Machine Learning

Quali transazioni (archi) sono fraudolenti

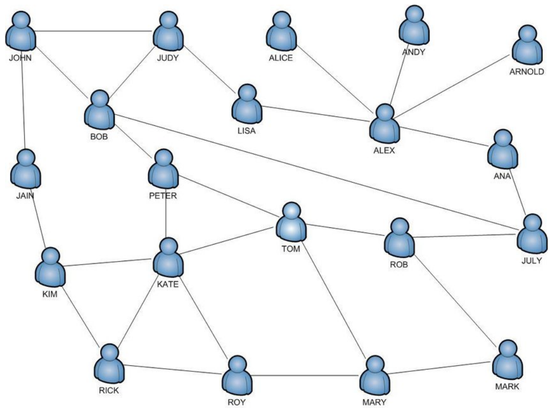
COS'È IL GRAPH MACHINE LEARNING - CLASSIFICARE GRAFI



Quali grafi rappresentano linee ferroviarie danneggiate ?

MACHINE LEARNING ON GRAPH - UN APPROCCIO SEMPLICE

- Un approccio «classico» nell'utilizzare I grafi all'interno di algoritmi di ML è creare un set di feature
- Supponiamo quindi di voler classificare I nodi, possiamo creare un feature vector utilizzando diverse metriche locali



Feature engineering

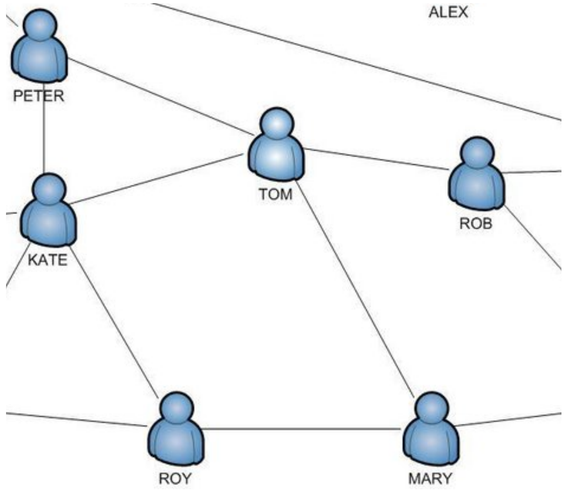
NodeId	Node Degree	…	Age
Alice	10	0.12	28
..	20	0.88	…
Bob	7	0.76	31

- *Node Attributes*: Feature specifiche del problema (Età, Sesso, Numero di figli, ..).
- *Local Structural Features*: Feature dei nodi come *degree* (count of adjacent nodes), *mean of degrees* dei nodi vicini, *number of triangles* che un nodo forma con alto, etc.

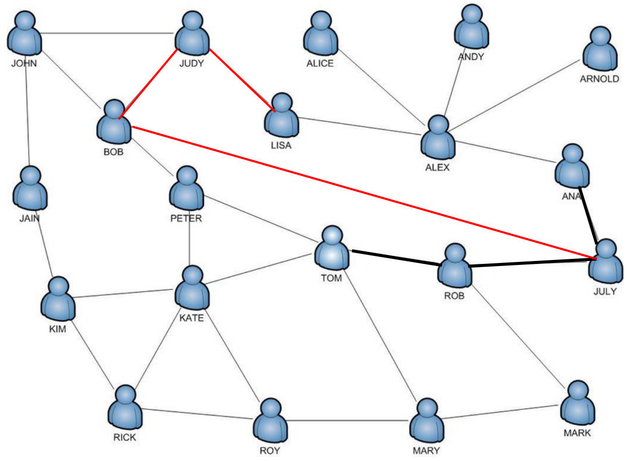
LIMITAZIONI DELL'APPROCCIO SEMPLICE

1. *Node Attributes:* Sappiamo che i nodi in un grafo rappresentano entità e queste entità hanno i loro attributi caratteristici (Age).
2. *Local Structural Features:* Feature di nodi come degree (count of adjacent nodes), mean of degrees of neighbor nodes, number of triangles etc.
3. *Embeddings:* **Le caratteristiche sopra discusse contengono solo informazioni relative al vicinato local nodo. Non acquisiscono le informazioni sul contesto di un nodo (i nodi a lui collegati direttamente o indirettamente).**

- **Local Structure Features**

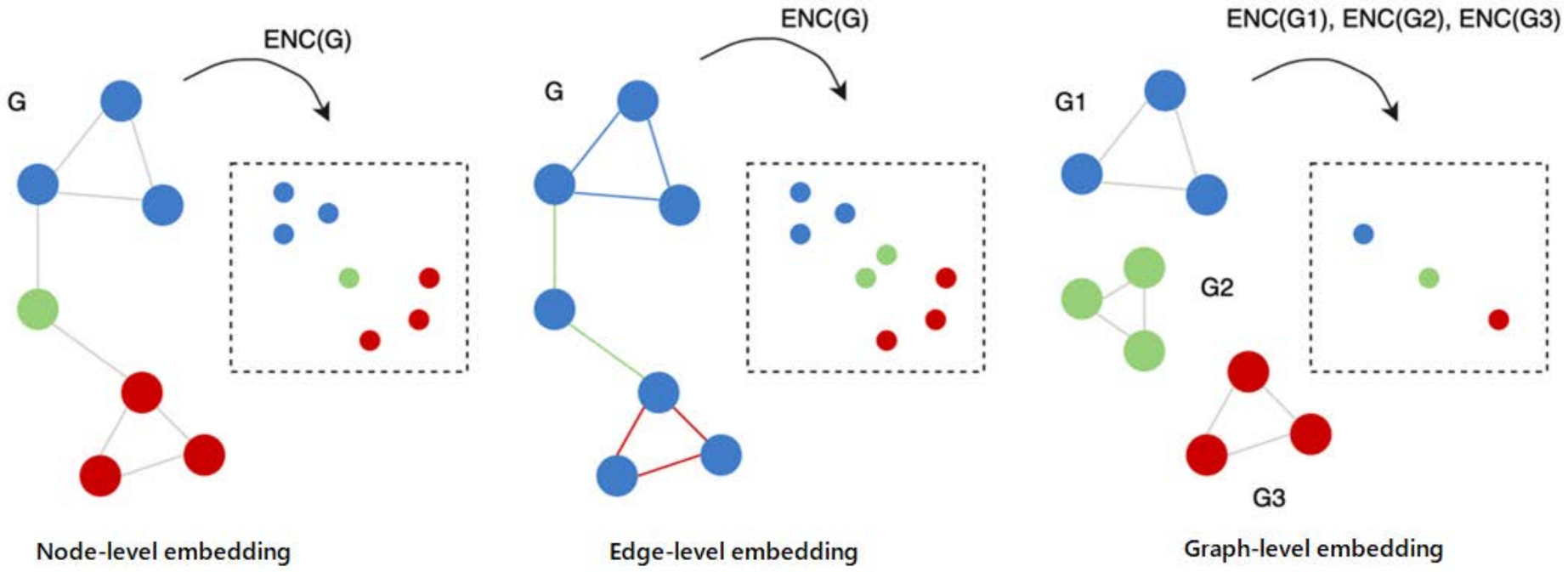


- **Embeddings**



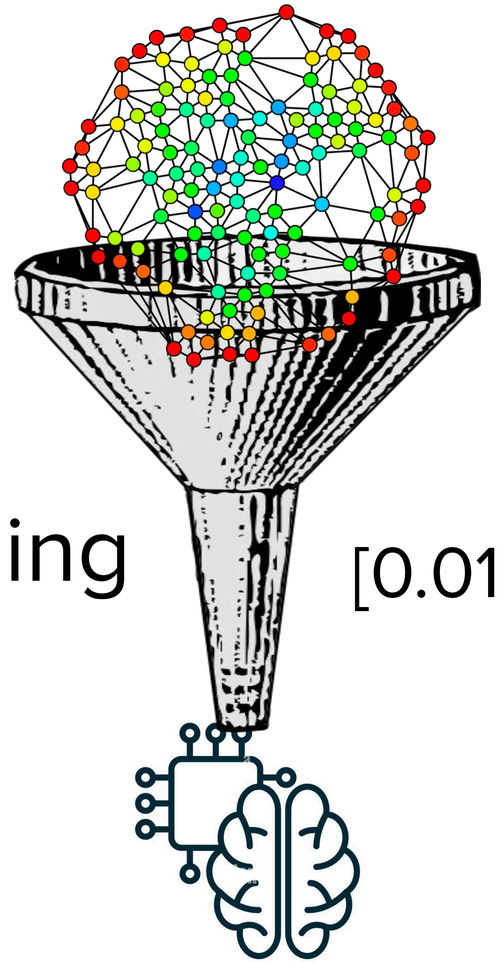
GRAPH MACHINE LEARNING - EMBEDDING, UNA INTRO NOISE

- *Representation learning* (network embedding) ha come scopo quello di apprendere una funzione di mapping $f: G \rightarrow \mathbb{R}^d$ dal dominio discreto dei grafi ad un dominio continuo.
- Una funzione capace di fornire una rappresentazione vettoriale compatta in modo che le proprietà del grafo (globali e locali) siano mantenute.



COS'È IL GRAPH MACHINE LEARNING

Graph Embedding



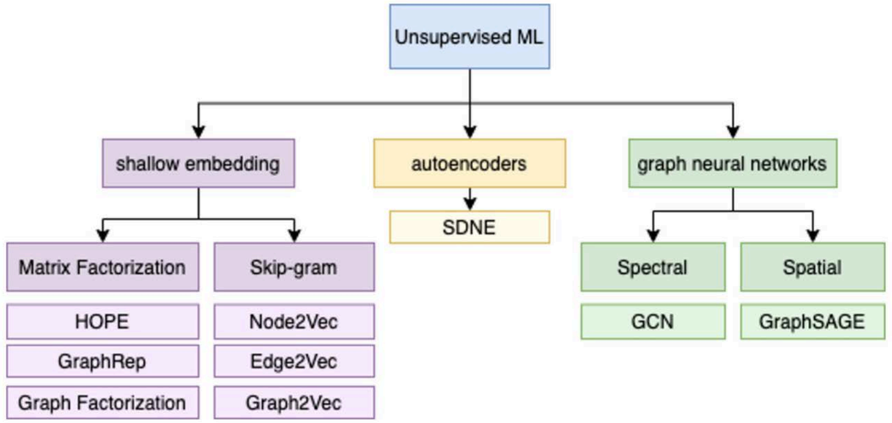
[0.01, 1.8, ..., 0.12]

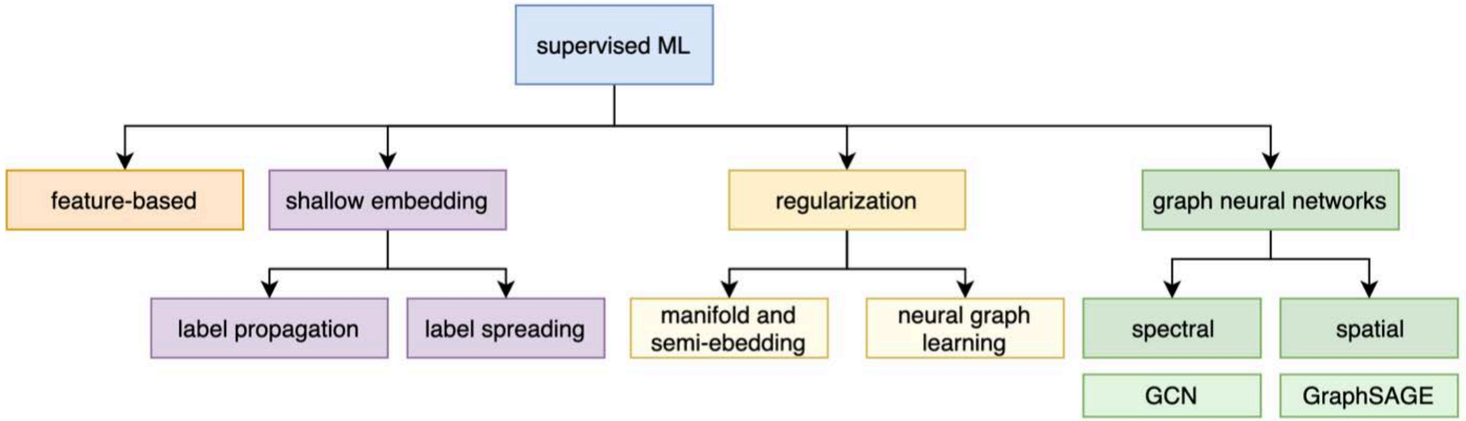
MACHINE LEARNING

GRAPH EMBEDDING ALGORITHMS

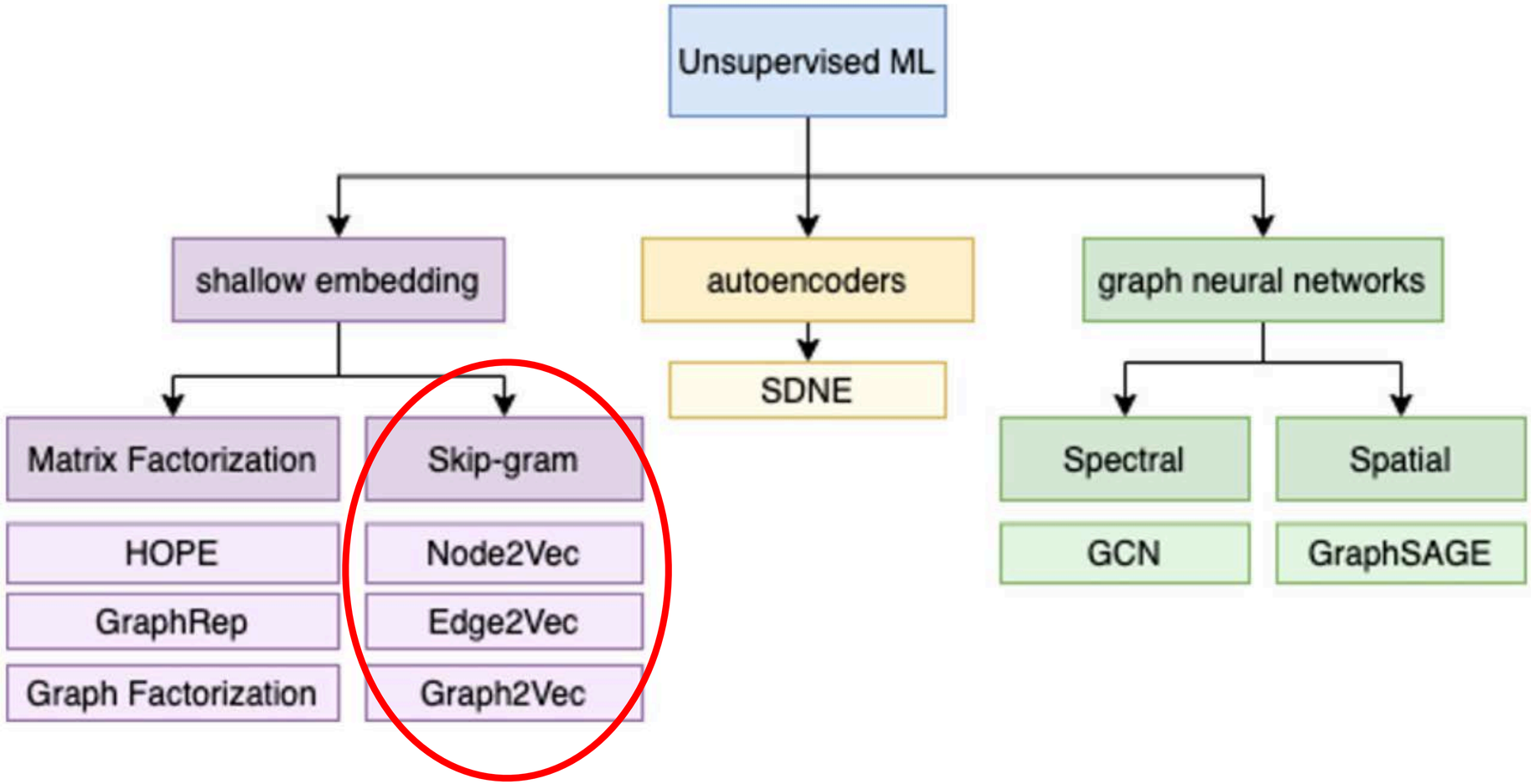
Machine Learning on Graphs:
A Model and Comprehensive Taxonomy

Ines Chami*†, Sami Abu-El-Haija‡, Bryan Perozzi††, Christopher Ré‡‡, and Kevin Murphy††



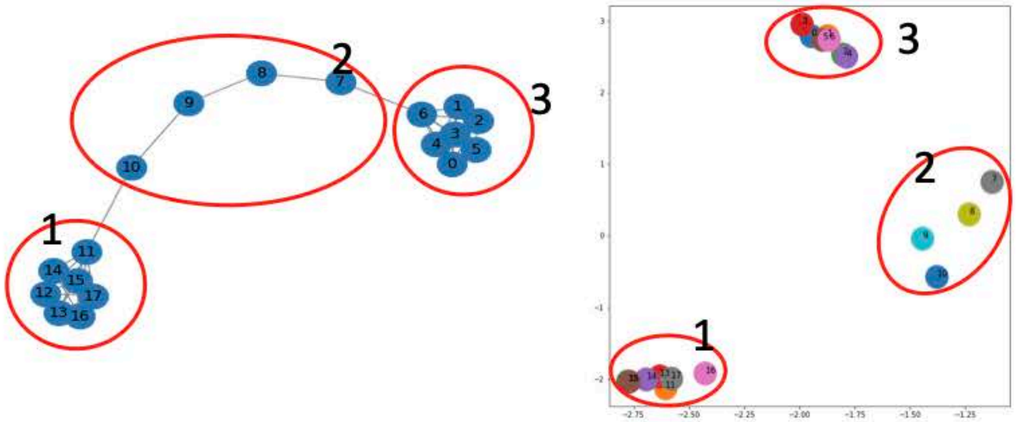


GRAPH EMBEDDING ALGORITHMS

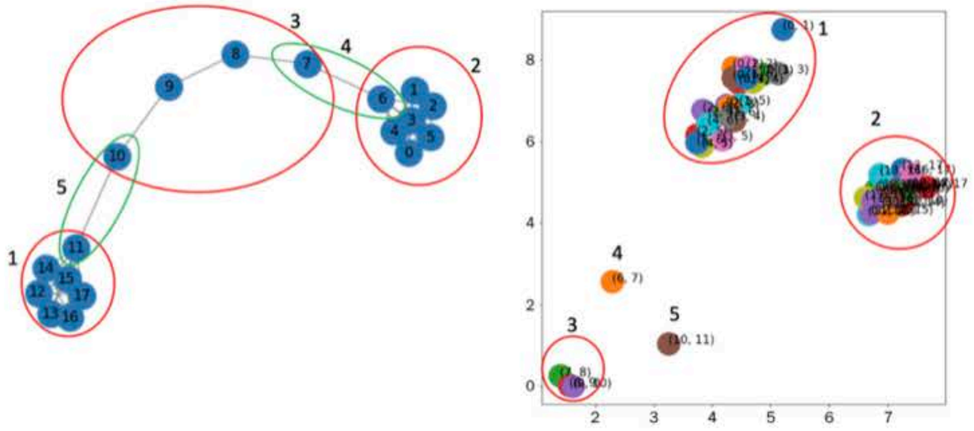


SKIP-GRAM BASED EMBEDDING ALGORITHMS– NODE2VEC, EDGE2VEC, GRAPH2VEC

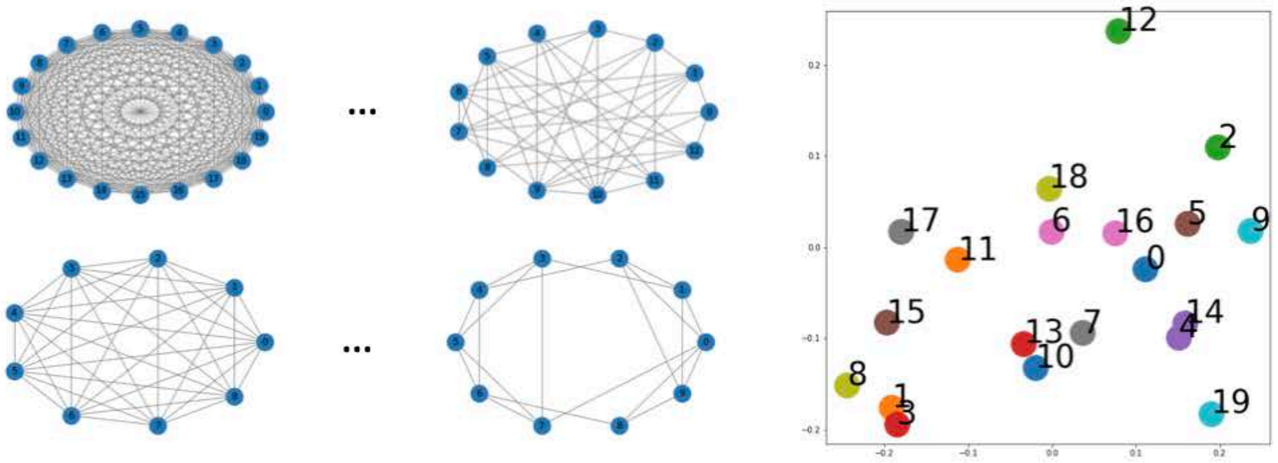
Node2Vec



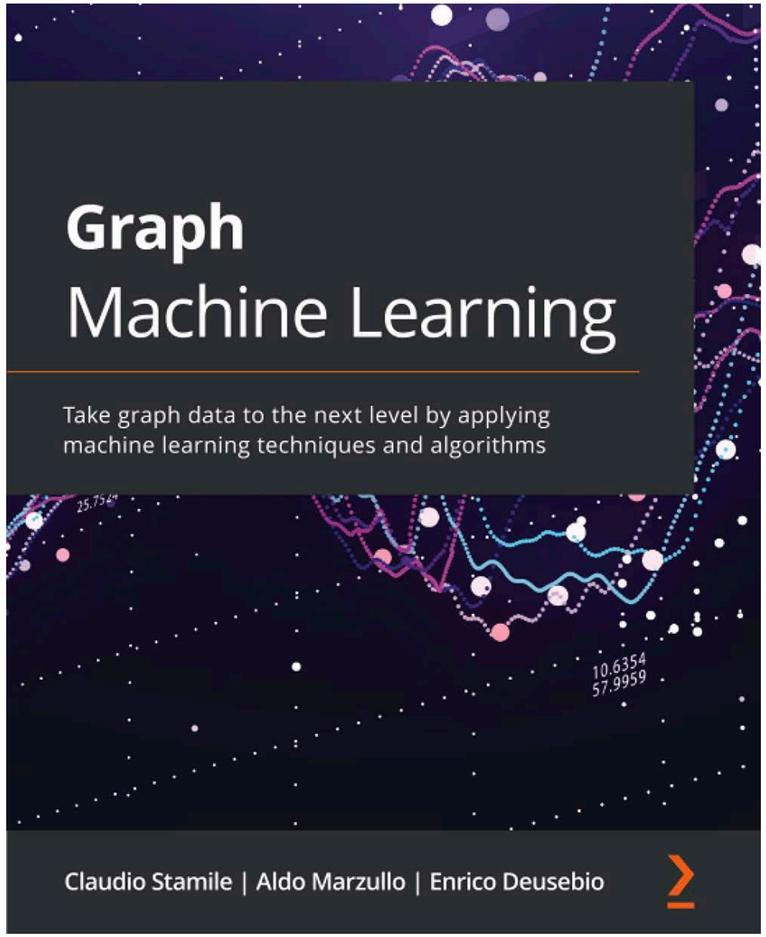
Edge2Vec

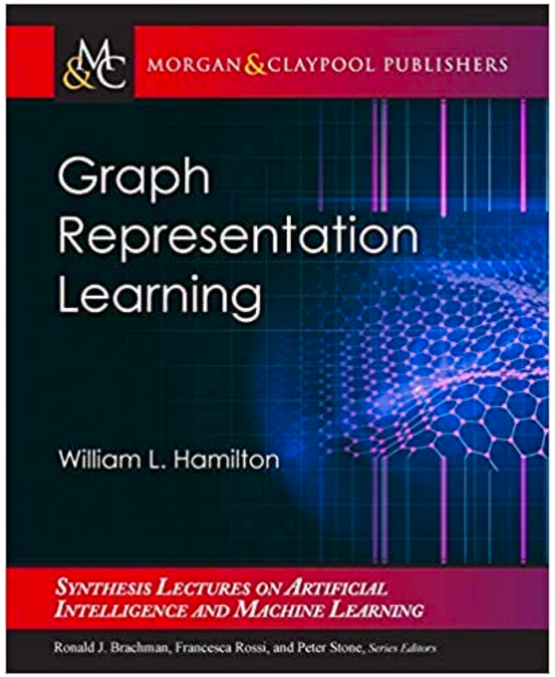


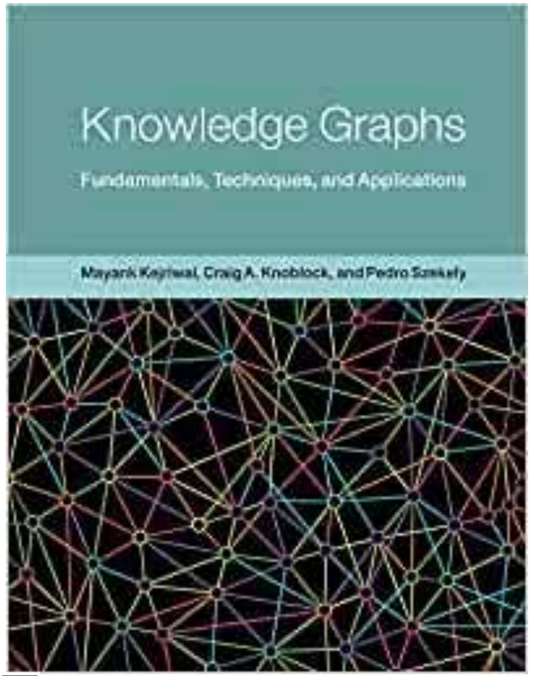
Graph2Vec

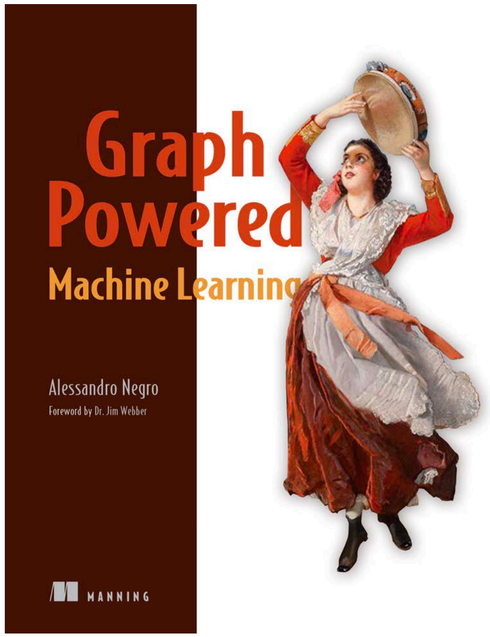


RIFERIMENTI









https://github.com/PacktPublishing/Graph-Machine-Learning

CLAUDIO STAMILE - 23/06/2022

THANK YOU

A GENTLE INTRODUCTION TO GRAPH MACHINE LEARNING IN PYTHON