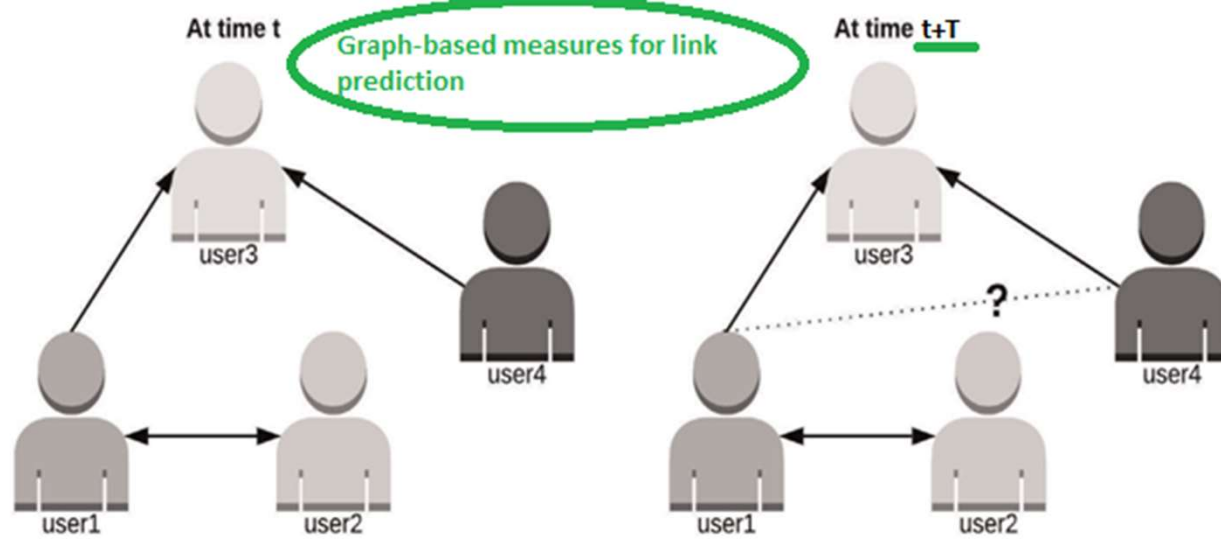
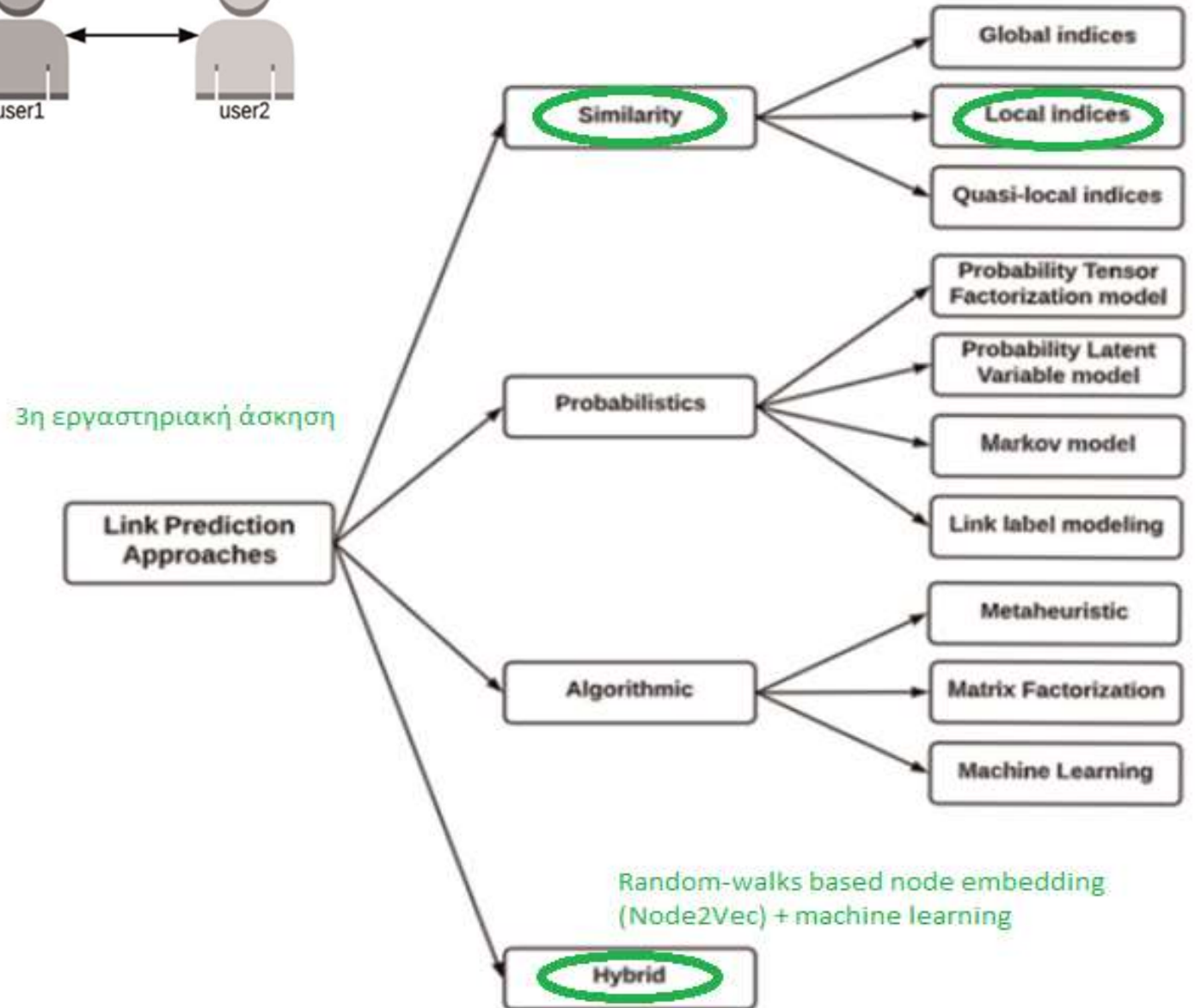


# ΑΚΔ – 3η Εργαστηριακή Άσκηση

## Link Prediction - Πρόβλεψη συνδέσμων



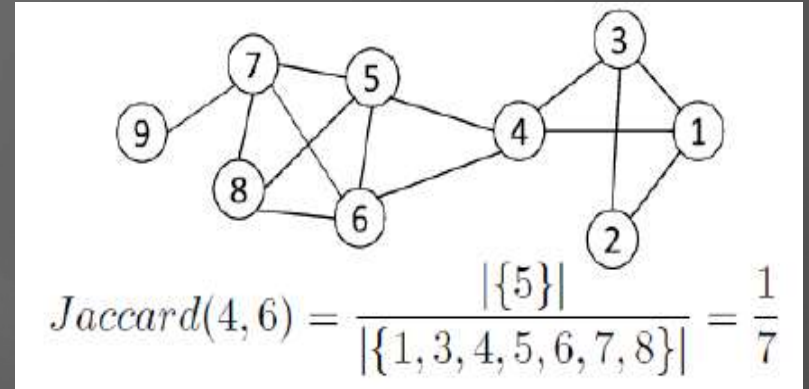
[1] Daud, Hamid, Saadoon, Sahran, Anuar, Applications of link prediction in social networks: A review. Journal of Network and Computer Applications, Volume 166, 2020, ELSEVIER.



# Similarity metrics (Local indices)

▣ Jaccard Coefficient:

$$\frac{|\Gamma(u) \cap \Gamma(v)|}{|\Gamma(u) \cup \Gamma(v)|}$$



▣ Preferential Attachment:

$$|\Gamma(u)| |\Gamma(v)|$$

▣ Resource Allocation:

$$\sum_{w \in \Gamma(u) \cap \Gamma(v)} \frac{1}{|\Gamma(w)|}$$

# NODE EMBEDDING ΒΑΣΙΣΜΕΝΟ ΣΕ ΤΥΧΑΙΟΥΣ ΠΕΡΙΠΑΤΟΥΣ: NODE2VEC [4]

## ΒΗΜΑΤΑ:

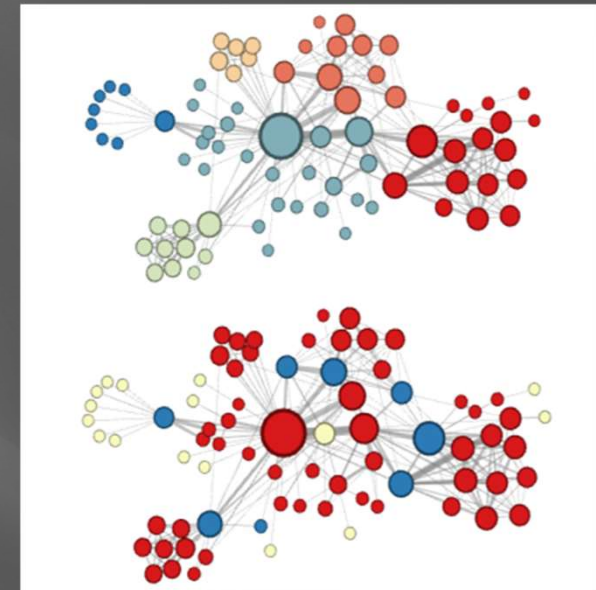
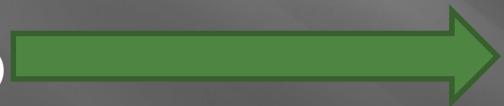
1. Τυχαίοι περίπατοι για παραγωγή προτάσεων από γράφο. Κάθε πρόταση είναι μια λίστα από node ids. Όλες οι προτάσεις συνιστούν ένα corpus.
2. Το corpus χρησιμοποιείται για να απεικονιστεί ένας embedding vector για κάθε κόμβο στον γράφο. Κάθε κόμβος θεωρείται μοναδική λέξη στο λεξικό που έχει μέγεθος ίσο με το πλήθος των κόμβων στον γράφο. Για τον υπολογισμό των embedding vectors χρησιμοποιείται το Word2Vec.

Node2Vec: Απεικόνιση γειτονικών κόμβων βάσει:

- Communities (homophily)

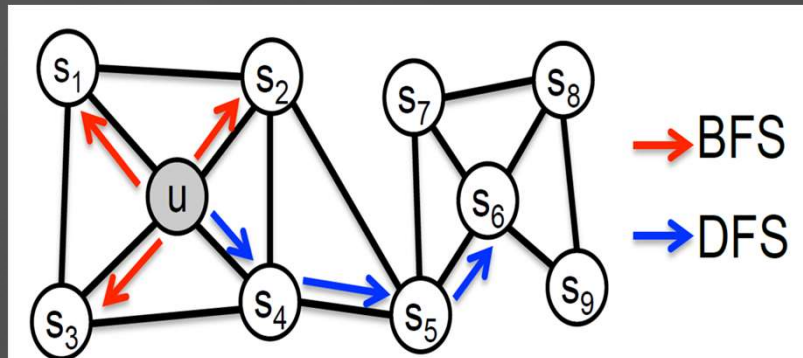


- Network roles (structural equivalence)



Les Misérables

- Κοντινό embedding κόμβων που ανήκουν στην ίδια κοινότητα ( $u, s_1$ ).
- Κόμβοι με όμοιους ρόλους (π.χ. hubs) πρέπει να έχουν παρόμοια embeddings ( $u, s_6$ ).





# ΠΑΡΑΓΩΓΗ ΤΥΧΑΙΩΝ ΠΕΡΙΠΑΤΩΝ ΜΕ ΤΟΝ NODE2VEC (1/2)

**Algorithm 1** The *node2vec* algorithm.

**LearnFeatures** (Graph  $G = (V, E, W)$ , Dimensions  $d$ , Walks per node  $r$ , Walk length  $l$ , Context size  $k$ , Return  $p$ , In-out  $q$ )  
 $\pi = \text{PreprocessModifiedWeights}(G, p, q)$   
 $G' = (V, E, \pi)$   
Initialize *walks* to Empty  
**for**  $iter = 1$  **to**  $r$  **do**  
    **for all** nodes  $u \in V$  **do**  
         $walk = \text{node2vecWalk}(G', u, l)$   
        Append *walk* to *walks*  
 $f = \text{StochasticGradientDescent}(k, d, walks)$   
**return**  $f$

**node2vecWalk** (Graph  $G' = (V, E, \pi)$ , Start node  $u$ , Length  $l$ )  
Initialize *walk* to  $[u]$   
**for**  $walk\_iter = 1$  **to**  $l$  **do**  
     $curr = walk[-1]$   
     $V_{curr} = \text{GetNeighbors}(curr, G')$   
     $s = \text{AliasSample}(V_{curr}, \pi)$   
    Append  $s$  to *walk*  
**return** *walk*

Objective function:

$$\max_f \sum_{u \in V} \log \Pr(N_S(u) | f(u))$$

Για κάθε τυχαίο περίπατο:

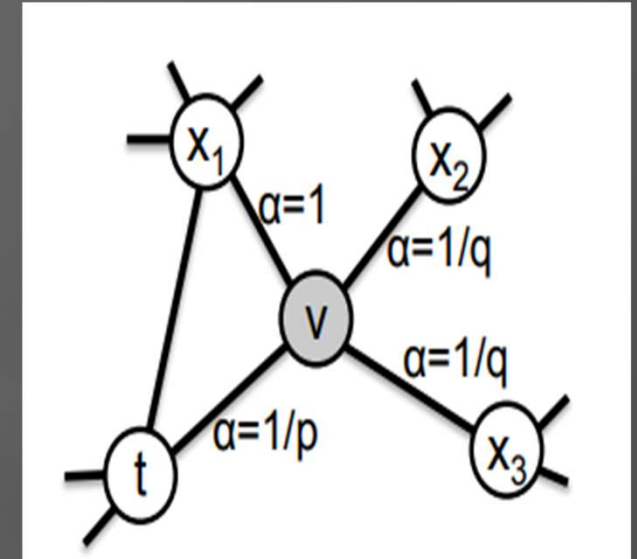
- Βήμα 1: αρχικός κόμβος
- Βήμα 2: επιλογή γειτονικού κόμβου ως επόμενου
- Βήμα 3: επανάληψη βήματος 2 μέχρι το walk length να γίνει 1

$$P(c_i = x \mid c_{i-1} = v) = \begin{cases} \frac{\pi_{vx}}{Z} & \text{if } (v, x) \in E \\ 0 & \text{otherwise} \end{cases}$$

# ΠΑΡΑΓΩΓΗ ΤΥΧΑΙΩΝ ΠΕΡΙΠΑΤΩΝ ΜΕ ΤΟΝ NODE2VEC (2/2)

Bias στην επιλογή του επόμενου κόμβου:

- Πιο πρόσφατη ακμή στο random walk :  $t \rightarrow v$
- Βρισκόμαστε στον  $v$



## Return parameter $p$ :

- $(p > \max(q, 1))$  λιγότερο πιθανό να επιστρέψουμε σε κόμβο που επισκεφθήκαμε
- $(p < \min(q, 1))$  backtrack κάποιου βήματος

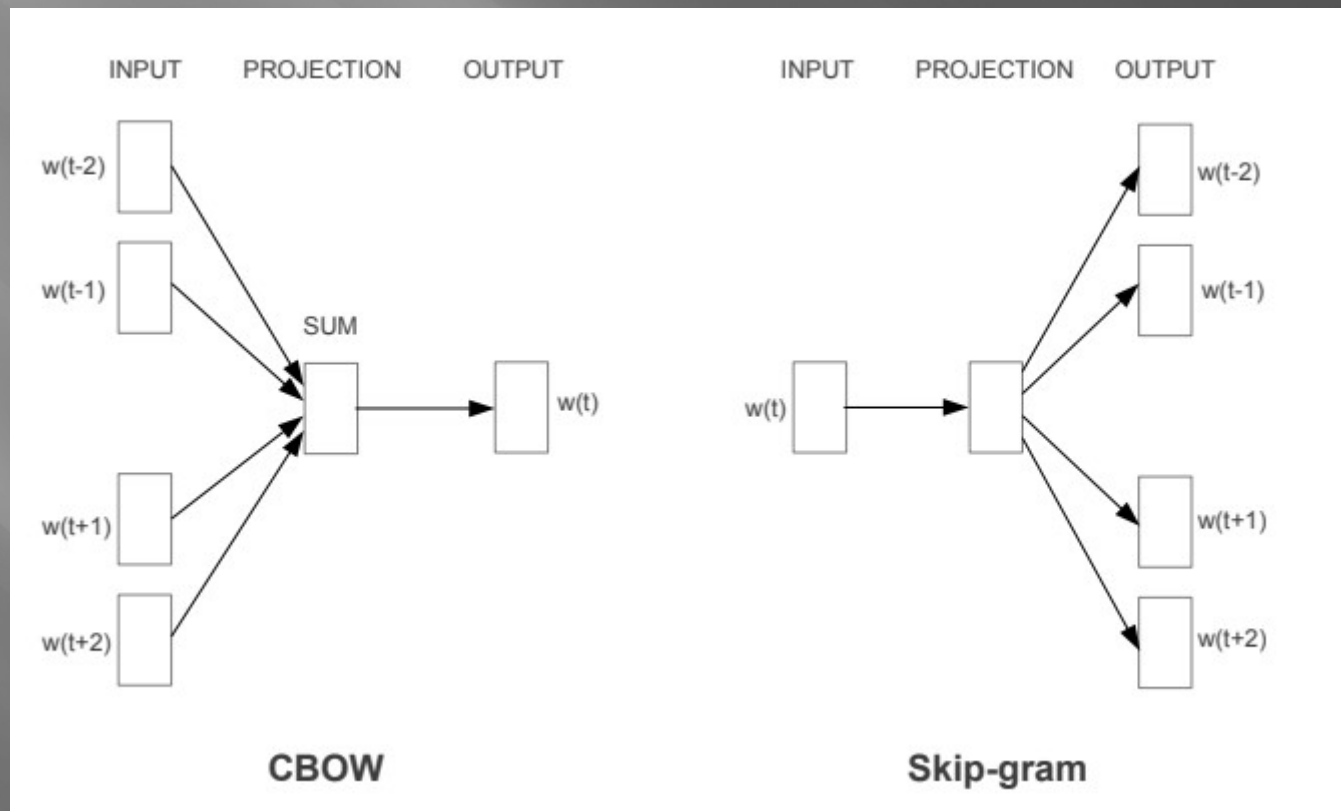
## In-out parameter $q$ :

- $q > 1$  walk biased να επισκεφθεί κόμβους κοντά στον  $t$  (local view/ BFS).
- $q < 1$  walk biased να επισκεφθεί κόμβους μακριά από τον  $t$  (global view/ DFS).

$$\alpha_{pq}(t, x) = \begin{cases} \frac{1}{p} & \text{if } d_{tx} = 0 \\ 1 & \text{if } d_{tx} = 1 \\ \frac{1}{q} & \text{if } d_{tx} = 2 \end{cases}$$

# WORD2VEC flavours

- CBOW**: Πρόβλεψη τρέχουσας λέξης  $w(t)$  βάσει του context
- Skip-gram**: Πρόβλεψη context βάσει της  $w(t)$



Mikolov et al., Exploiting Similarities among Languages for Machine Translation, 2013, arXiv

# WORD2VEC SKIP-GRAM MODEL

- objective function:

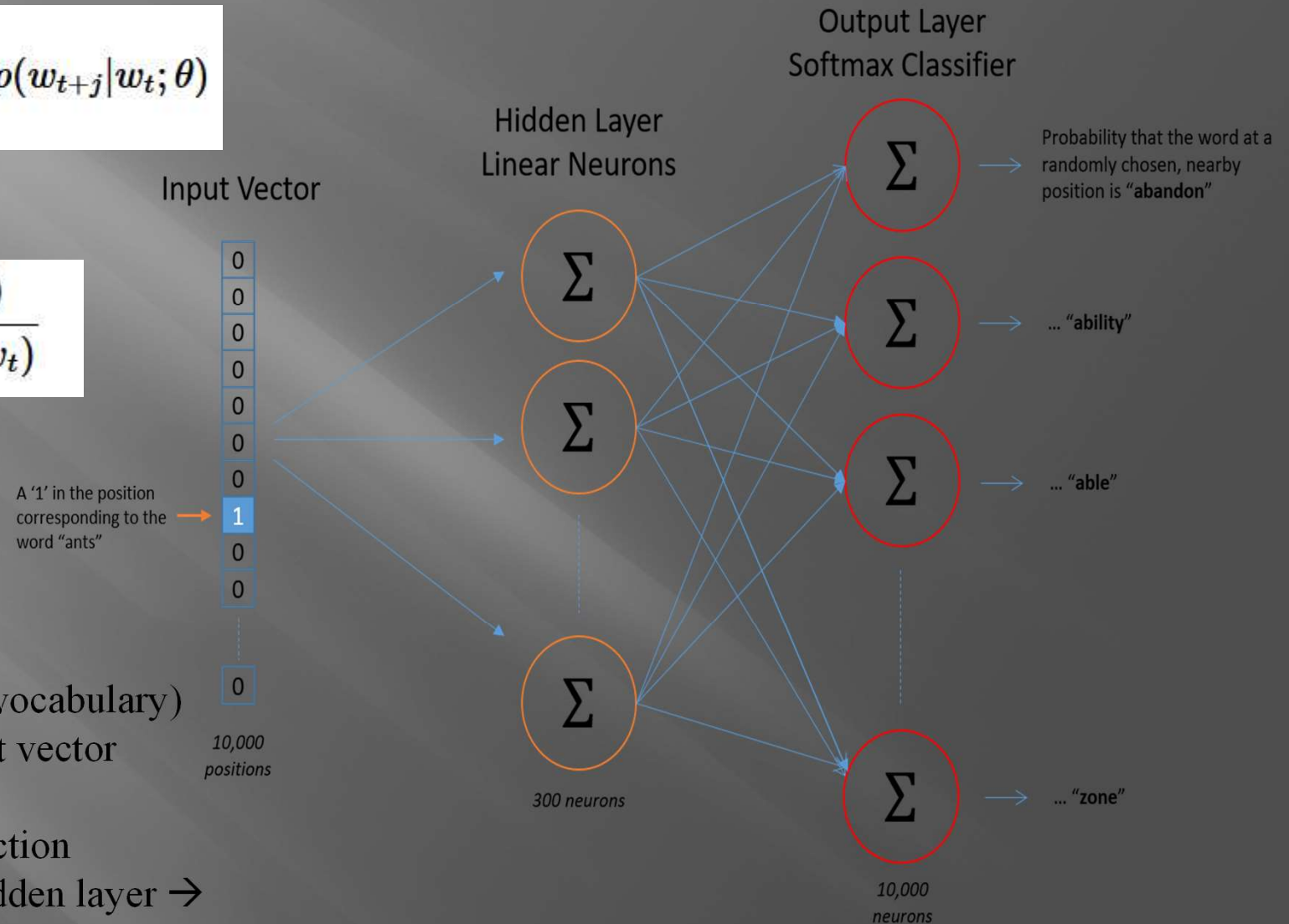
$$\operatorname{argmax}_{\theta} \frac{1}{T} \sum_{t=1}^T \sum_{j \in \mathcal{C}, j \neq 0} \log p(w_{t+j} | w_t; \theta)$$

- Softmax:

$$p(w_i | w_t; \theta) = \frac{\exp(\theta w_i)}{\sum_t \exp(\theta w_t)}$$

Παράδειγμα:

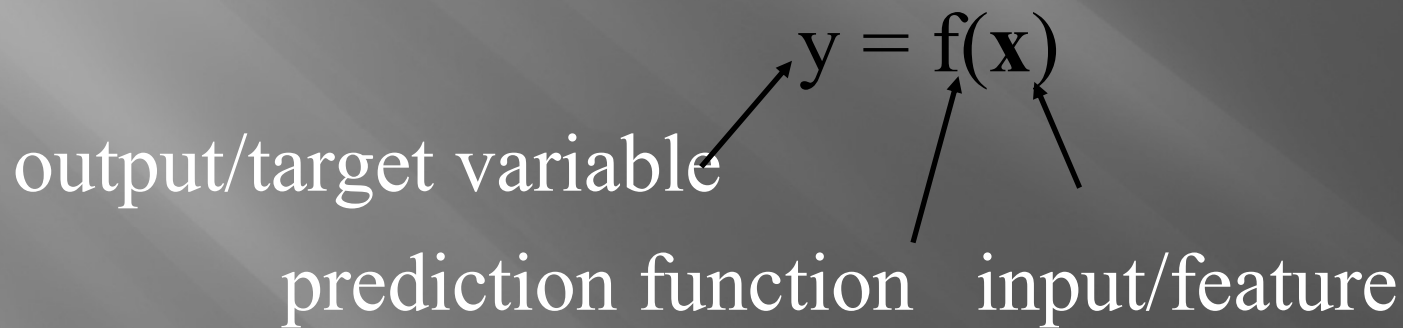
- $V = 10000$  (μέγεθος vocabulary)
- input: “ants” (one-hot vector encoding)
- Χωρίς activation function
- 300 νευρώνες στο hidden layer  $\rightarrow$  300 features





# Μηχανική μάθηση (Machine Learning)

- ▣ **Training:** training set από labeled examples  $\{(\mathbf{x}_1, y_1), \dots, (\mathbf{x}_N, y_N)\}$ , υπολογισμός της prediction function  $f$
- ▣ **Testing:** εφαρμογή της  $f$  σε νέα test examples  $\mathbf{x}$  και υπολογισμός της  $y = f(\mathbf{x})$



# Μηχανική μάθηση (Machine Learning)

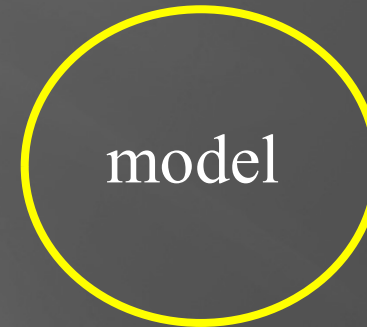
training data  
(labeled)

Terrain	Unicycle-type	Weather	Go-For-Ride?
Trail	Normal	Rainy	NO
Road	Normal	Sunny	YES
Trail	Mountain	Sunny	YES
Road	Mountain	Rainy	YES
Trail	Normal	Snowy	NO
Road	Normal	Rainy	YES
Road	Mountain	Snowy	YES
Trail	Normal	Sunny	NO
Road	Normal	Snowy	NO
Trail	Mountain	Snowy	YES



Terrain	Unicycle-type	Weather	Go-For-Ride?
Trail	Normal	Rainy	NO
Road	Normal	Sunny	YES
Trail	Mountain	Sunny	YES
Road	Mountain	Rainy	YES
Trail	Normal	Snowy	NO
Road	Normal	Rainy	YES
Road	Mountain	Snowy	YES
Trail	Normal	Sunny	NO
Road	Normal	Snowy	NO
Trail	Mountain	Snowy	YES

learn



pre-processing

test data

Terrain	Unicycle-type	Weather	Go-For-Ride?
Trail	Normal	Rainy	NO
Road	Normal	Sunny	YES
Trail	Mountain	Sunny	YES
Road	Mountain	Rainy	YES
Trail	Normal	Snowy	NO
Road	Normal	Rainy	YES
Road	Mountain	Snowy	YES
Trail	Normal	Sunny	NO
Road	Normal	Snowy	NO
Trail	Mountain	Snowy	YES

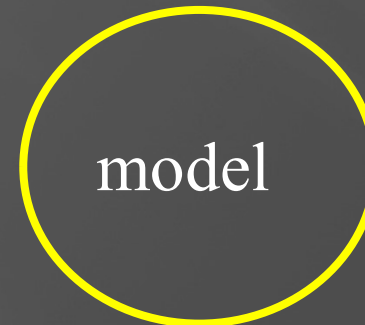


Terrain	Unicycle-type	Weather	Go-For-Ride?
Trail	Normal	Rainy	NO
Road	Normal	Sunny	YES
Trail	Mountain	Sunny	YES
Road	Mountain	Rainy	YES
Trail	Normal	Snowy	NO
Road	Normal	Rainy	YES
Road	Mountain	Snowy	YES
Trail	Normal	Sunny	NO
Road	Normal	Snowy	NO
Trail	Mountain	Snowy	YES



Classify (discrete output variable)

Regress (continuous output variable)



prediction

# Random Forest Classifier

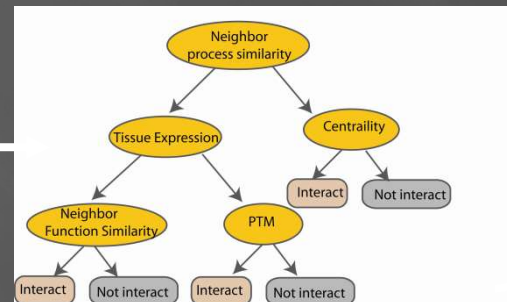
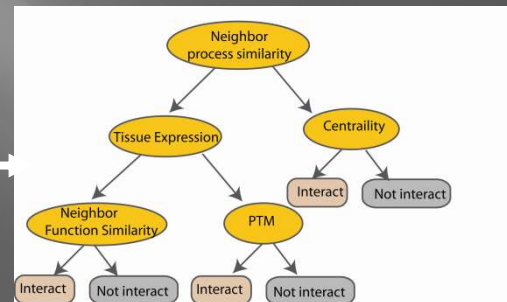
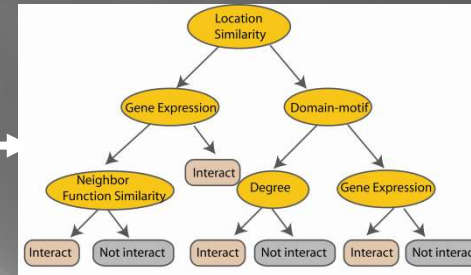
$$\hat{f}_{\text{bag}}(x) = \frac{1}{B} \sum_{b=1}^B \hat{f}^{*b}(x).$$

B bootstrap samples  
from training data

M features

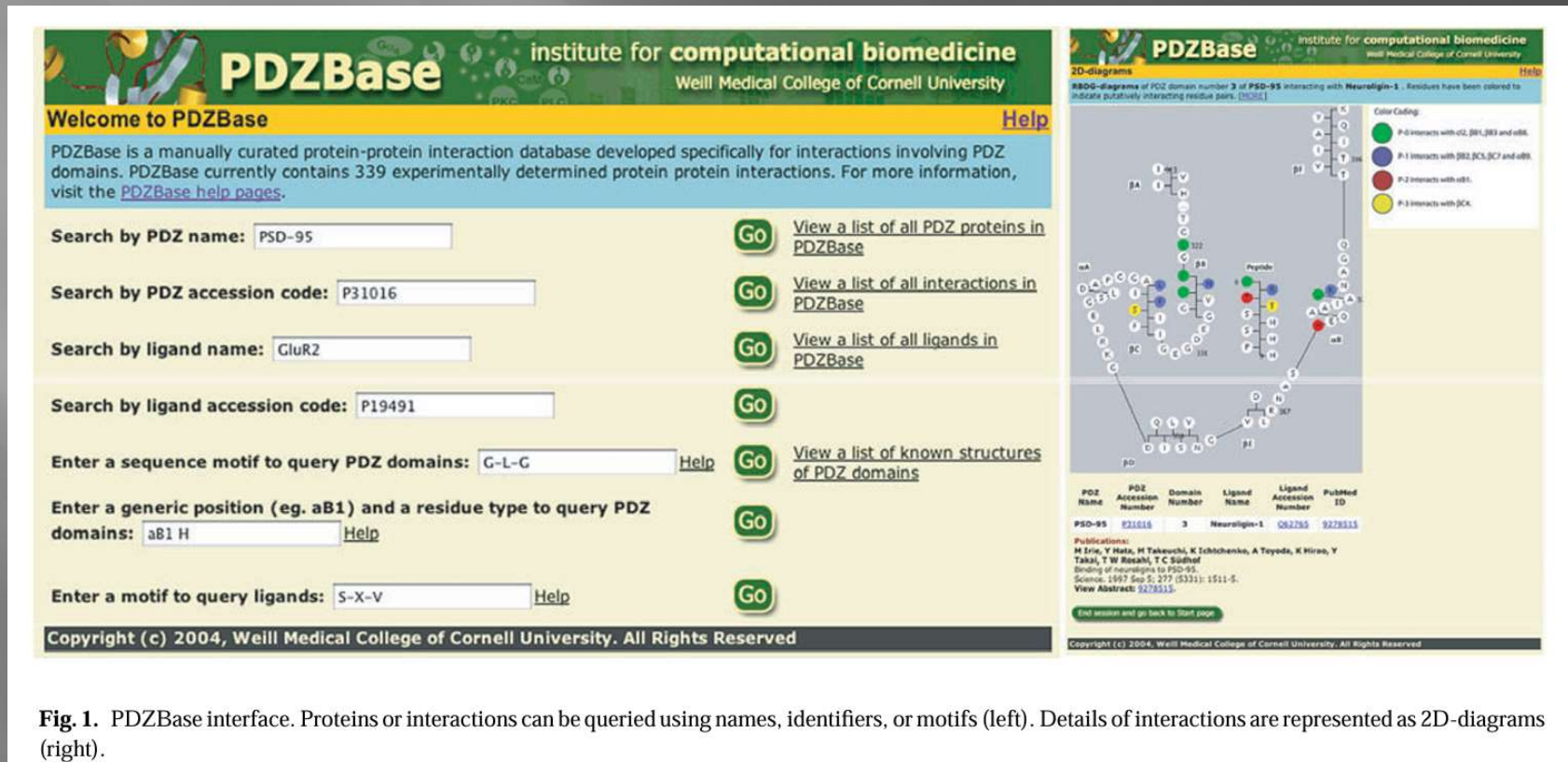
N examples

Majority  
voting



# 1. Κατασκευή γράφου και υπολογισμός μετρικών

- 1(i). Κατασκευή γράφου από PDZBase <http://konect.cc/networks/maayan-pdzbases/>
- Κόμβοι: πρωτεΐνες, ακμές: αλληλεπίδραση μεταξύ τους.
- 1(ii).  $G: |V|, |E|, \text{avg}(\text{degree}), \# \text{conn. components}$ . Αφαίρεση όλων των self-edges.

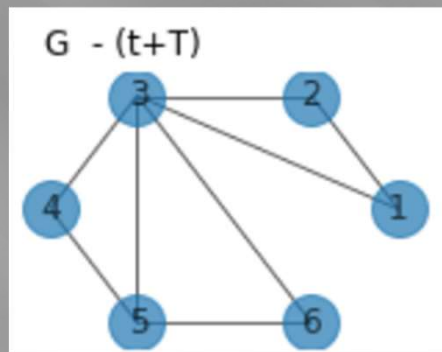




## 2. Προεργασία για την πρόβλεψη συνδέσμων

- 2(i). Υπολογίστε όλα τα ασύνδετα ζεύγη κόμβων στον γράφο χρησιμοποιώντας τον πίνακα γειτνίασης (adjacency matrix) (μη κατευθυνόμενος γράφος, συμμετρικός πίνακας)

π.χ.



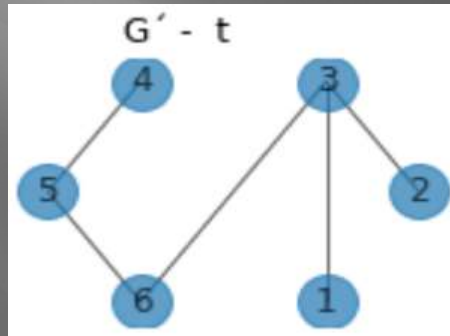
[	0.	1.	1.	0.	0.	0.	]
[	1.	0.	1.	0.	0.	0.	]
[	1.	1.	0.	1.	1.	1.	]
[	0.	0.	1.	0.	1.	0.	]
[	0.	0.	1.	1.	0.	1.	]
[	0.	0.	1.	0.	1.	0.	]

unconnected\_edges={ (1, 4), (1, 5), (1, 6), (2, 4), (2, 5), (2, 6), (4, 6) }

- 2(ii). Υπολογίστε το σύνολο των ακμών που μπορούν να αφαιρεθούν από τον γράφο χωρίς να υπάρχει κατάτμηση (splitting) του γράφου (αριθμός των συνεκτικών συνιστωσών αμετάβλητος). Ελέγξτε, επίσης, ότι δεν μειώνεται ο αριθμός των κόμβων (εφόσον δεν υπάρχει ξεχωριστό αρχείο μόνο με κόμβους στο dataset).

π.χ. removable\_edges={ (1, 2), (3, 4), (3, 5) }

- ▣ **2(iii).** Κατασκευάστε dataframe που συνενώνει τις απαντήσεις των ερωτημάτων 2(i) και 2(ii) (unconnected+removables). Χρησιμοποιήστε την μεταβλητή *link* που θα παίρνει τιμή 1 ή 0 ανάλογα με την ύπαρξη συνδέσμου ή μη. Το dataframe αυτό συγκεντρώνει τα θετικά και αρνητικά δείγματα για την πρόβλεψη των μελλοντικών συνδέσεων.
- ▣ **2(iv).** Κατασκευάστε τον γράφο  $G'$  που προκύπτει από την αφαίρεση από τον γράφο  $G$  των ακμών του ερωτήματος 2(ii).



- ▣ **2(v).**  $G'$ :  $|V|$ ,  $|E|$ , avg(degree), #conn.components.

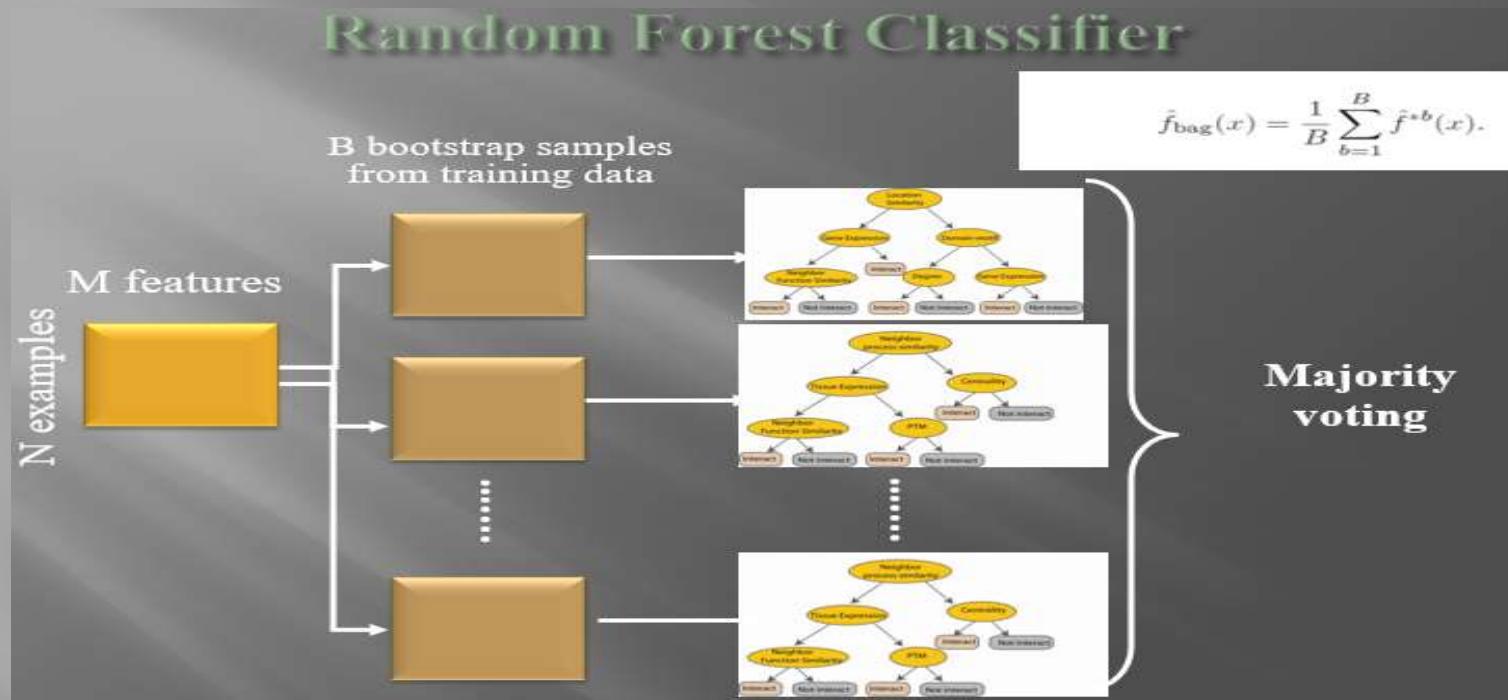
### 3. Πρόβλεψη συνδέσμων με embedding βασισμένο σε τυχαίους περιπάτους (Random Walks)

- 3(i). Εξάγετε τα χαρακτηριστικά (features) του γράφου  $G'$  με τον αλγόριθμο Node2vec.

p	q	dimensions	num_walks	walk_length	window_size	workers
1.0	1.0	128	10	80	10	multiprocessing.cpu_count()

- p - παράμετρος τυχαίων περιπάτων p που καθορίζει την πιθανότητα “1/p” επιστροφής στον κόμβο προέλευσης (source node),
- q - παράμετρος τυχαίων περιπάτων q που καθορίζει την πιθανότητα “1/q” μετακίνησης σε κόμβο μακριά από τον κόμβο προέλευσης (source node),
- dimensions - πλήθος διαστάσεων των node2vec embeddings,
- num\_walks - αριθμός περιπάτων από κάθε κόμβο,
- walk\_length - μήκος τυχαίου περιπάτου,
- window\_size - μέγεθος παραθύρου context για τον αλγόριθμο Word2Vec,
- num\_iter - αριθμός SGD επαναλήψεων (epochs),
- workers - αριθμός workers για τον Word2Vec

- 3(ii). Εφαρμόστε τον αλγόριθμο Random Forest Classifier, για να προβλέψετε τις συνδέσεις.



- 3(iii). Υπολογίστε την ακρίβεια (accuracy) του μοντέλου.



## Βιβλιογραφία:

- [1] Nur Nasuha Daud, Siti Hafizah Ab Hamid, Muntadher Saadoon, Firdaus Sahran, Nor Badrul Anuar, Applications of link prediction in social networks: A review, Journal of Network and Computer Applications, Volume 166, 2020, ELSEVIER, ISSN 1084-8045
- [2] Jérôme Kunegis. KONECT – The Koblenz Network Collection. In Proc. Int. Conf. on World Wide Web Companion, pages 1343–1350 2013.
- [3] Beuming, T., Skrabanek, L., Niv, M.Y., Mukherjee, P. and Weinstein, H., 2005. PDZBase: a protein–protein interaction database for PDZ-domains. Bioinformatics, 21(6), pp.827-828.
- [4] A. Grover, J. Leskovec, Node2vec: Scalable Feature Learning for Networks. In ACM KDD, 2016.
- [5] T. Mikolov, I. Sutskever, K. Chen, G. S. Corrado, and J. Dean. Distributed representations of words and phrases and their compositionality. In NIPS, 2013.