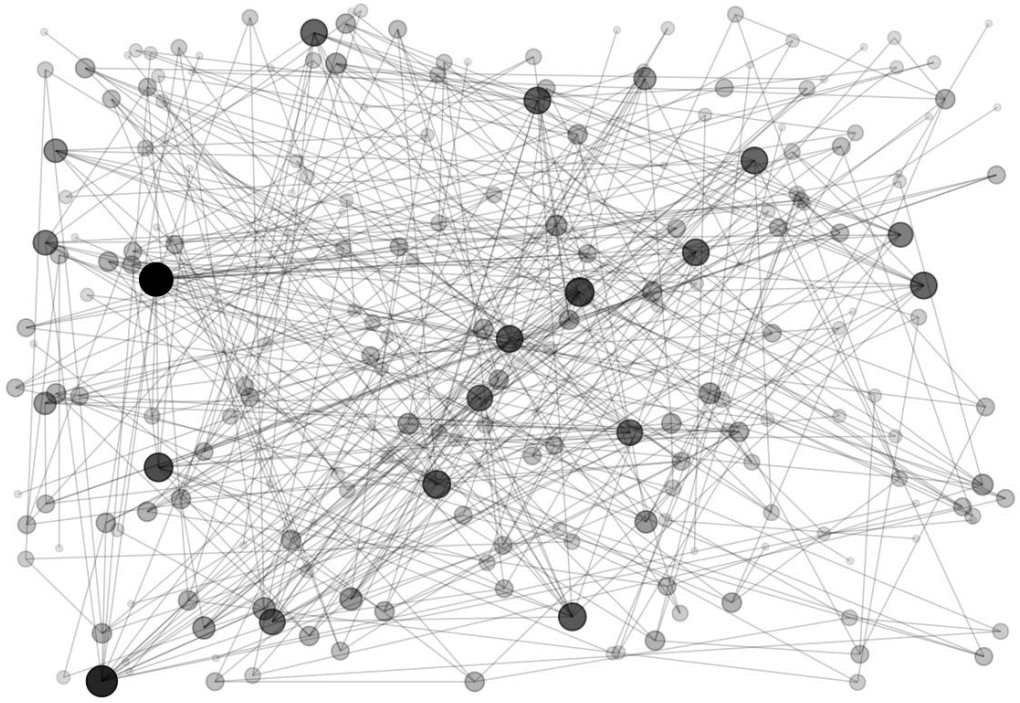
Domain Text Classification

Domain Node Classification | Project Scope

- **Objective:** Use Machine Learning techniques to classify web domains based on textual data and link structures.

- **Approach:** Combine text classification and node classification methods for optimal prediction of domain classes minizing the multiclass log loss

$$\mathcal{L} = -\frac{1}{N}\sum_{i=1}^{N}\sum_{j=1}^{C} y_{ij}\log(p_{ij})$$



Data Preprocessing & Preparation

Data Sources | Greek web domains

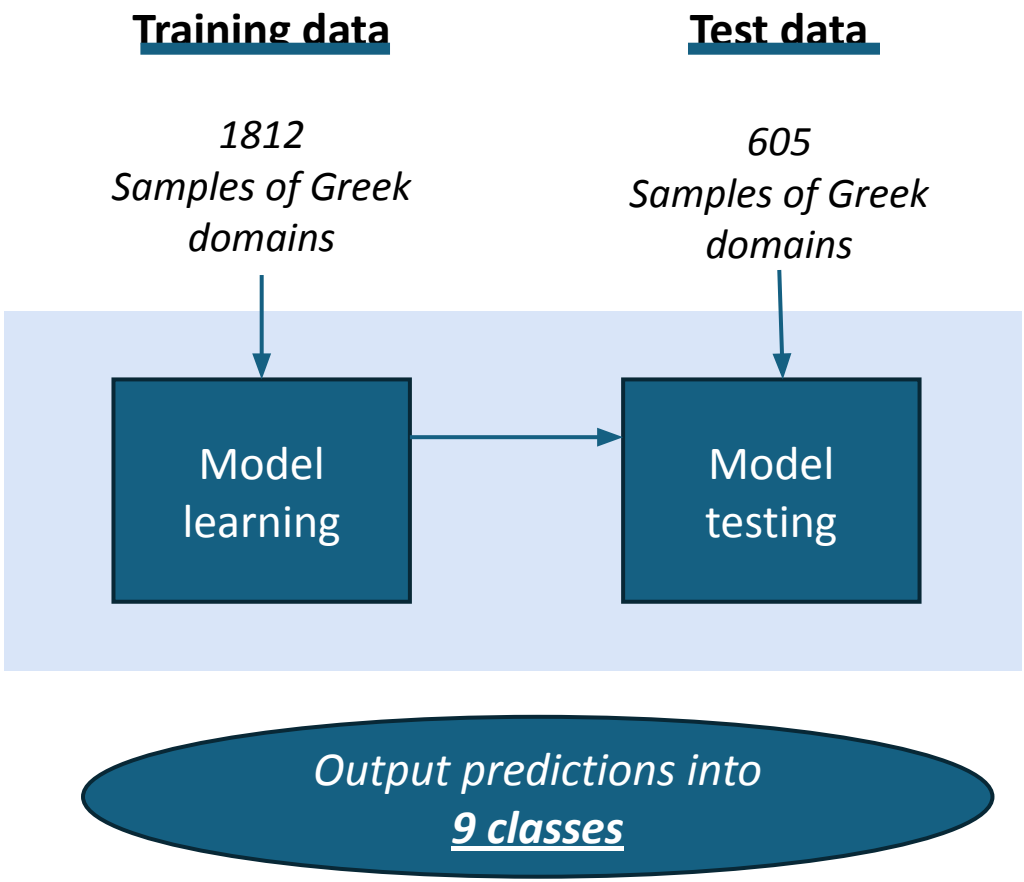
Data Categories

Textual Content

- Text data of the domains in scope that have been produced by web scraping

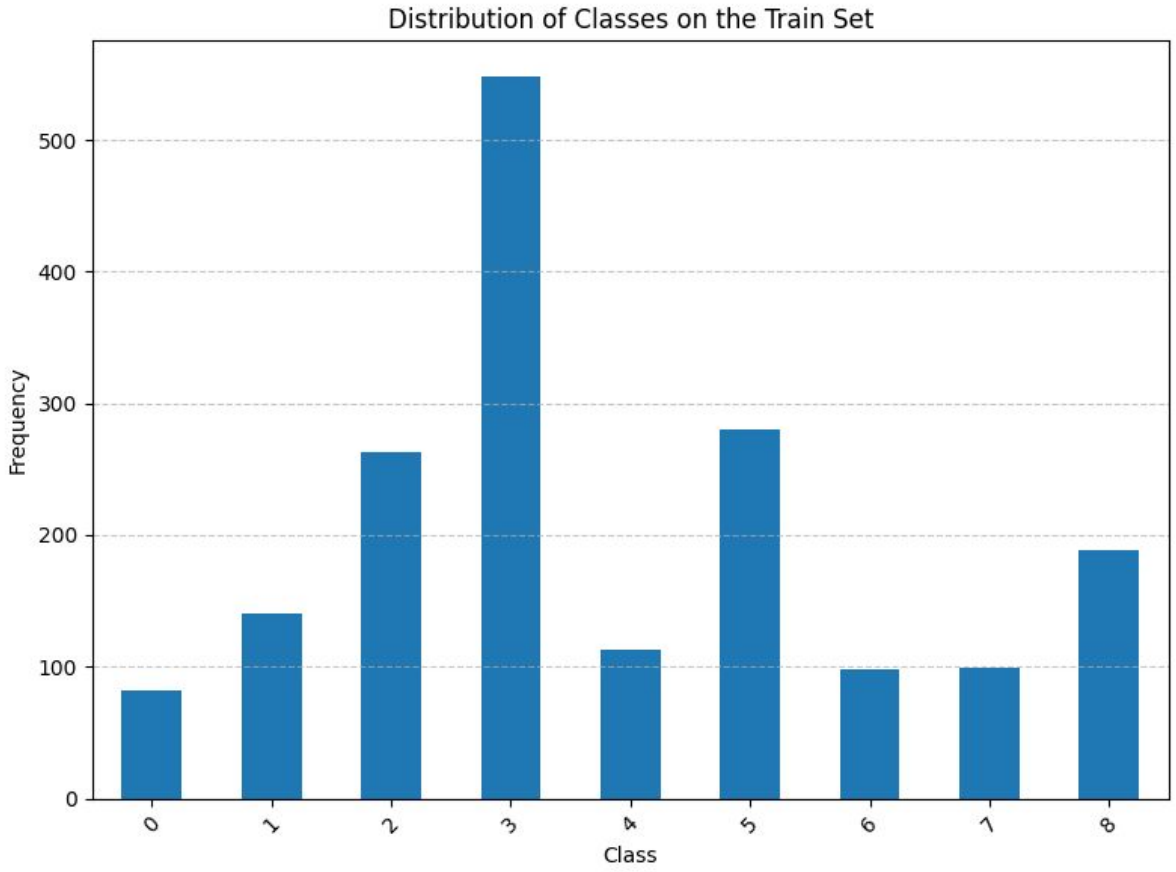
Graph-related data

- Graph data represent the connectivity of domains within the Greek web, where nodes are domain names and edges are the hyperlinks between them



Data Exploration | Class Distribution

Analyzing the Class Distribution in Training Data to Understand Imbalance and Representation

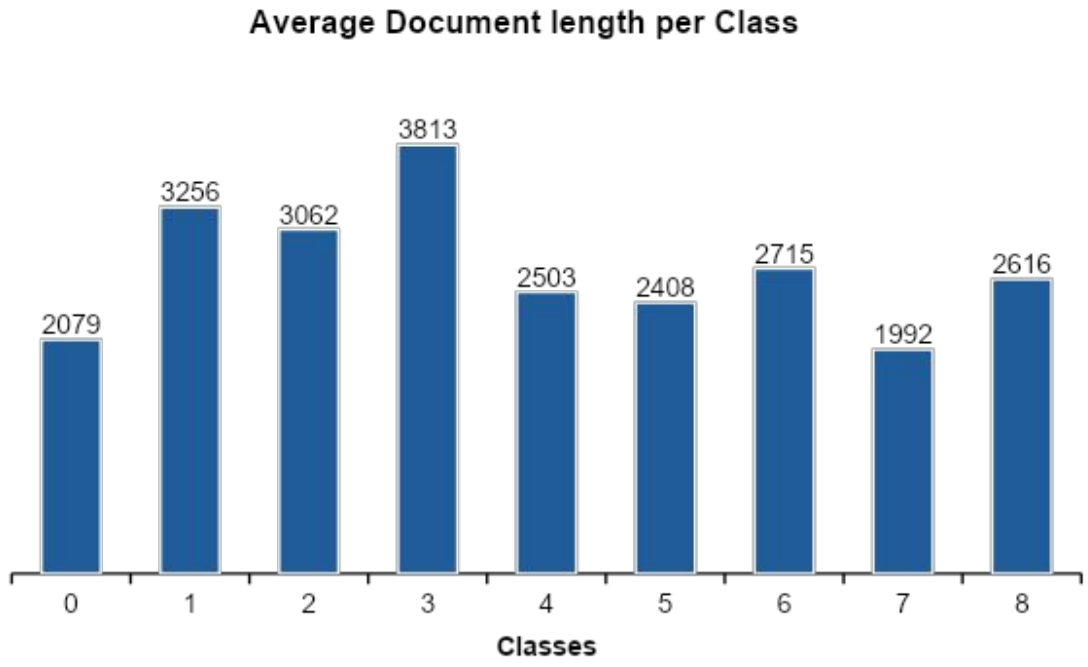


- Significant ***imbalance*** among the classes.
- ***Class 3*** is the most dominant with over 500 instances.
- ***Classes 0, 6, and 7*** seem underrepresented with fewer than 100 instances each.

Data Exploration | Domain Node Classification

Deep dive into the Classes

Class	*Domains*		*Top words*	
0	Autocarnet.gr	Mydirect.gr	Αυτοκινήτου	Παραδόθηκε
1	Coachbasket.gr	Basketplus.gr	Παιδαγωγικός	Προπονητής
2	Yourate.gr	Rockandroll.gr	Υπερρεαλιστικά	Μακιαβελικό
3	Athensgo.gr	Onalert.gr	Κοινοποίηση	Καταλάβετε
4	Oaed.gr	Education.gr	Πληροφορική	Αναβάθμιση
5	Queen.gr	Hostplus.gr	Φωτογραφία	Τηλεοπτική
6	Topgamos.gr	Spitistaleuka.gr	Μοιραστείτε	Επισκέπτες
7	Holiday.gr	Samothraki.gr	Αεροδρόμιο	Αναμνήσεις
8	Psychotherapia.net.gr	Sweetandbalance.gr	Γλυκαιμικός	Πανεπιστήμιο



- The top words for each class give ***insights*** into the ***key themes*** and ***topics*** covered within each domain.
- The average ***document length*** varies significantly across classes.

Data Exploration | Graph features

Analysis of Domain Popularity per Class: **In-Degrees** and **Out-Degrees**

Class	*Highest In-degrees*	*Highest Out-degrees*
0	Car.gr	Moto.gr
1	gazzetta.gr	Podilates.gr
2	sansimera.gr	Slang.gr
3	tovima.gr	In2life.gr
4	uoa.gr	Auth.gr
5	google.gr	Freestuff.gr
6	mothersblog.gr	Hamomilaki.gr
7	Oasa.gr	Arttravel.gr
8	Iatronet.gt	Iator.gr

- Domains with ***high in – degrees*** are the most ***popular*** or influential within each class.
- Domains with ***high out – degrees*** are potential ***hubs*** within each class.

*This comprehensive analysis facilitates the **identification** of **core themes** and key domains per Class.*

Data Exploration | Domains' Themes

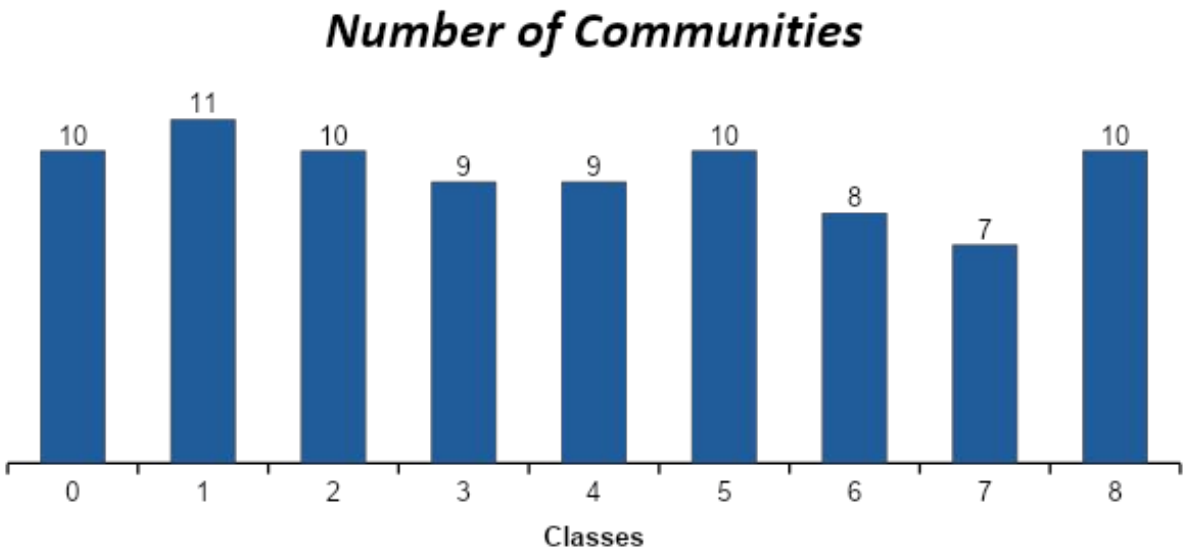
Combination of the above analyses to understand the class thematics

Class	*Highest In-degrees*
0	Driving/ Cars
1	Sports
2	Art
3	News
4	Education
5	Weather/ Online Transactions
6	Housing / Home-ware
7	Travelling
8	Health

- Combining the results of the above analyses, we conclude on specific thematic per Class.
- The mapping explains some of the above details e.g., higher document length in "News" domain.

Data Exploration | Communities Detection

Examination of the structural organization and cohesiveness of nodes within each class



Higher Number of Communities (10-11):

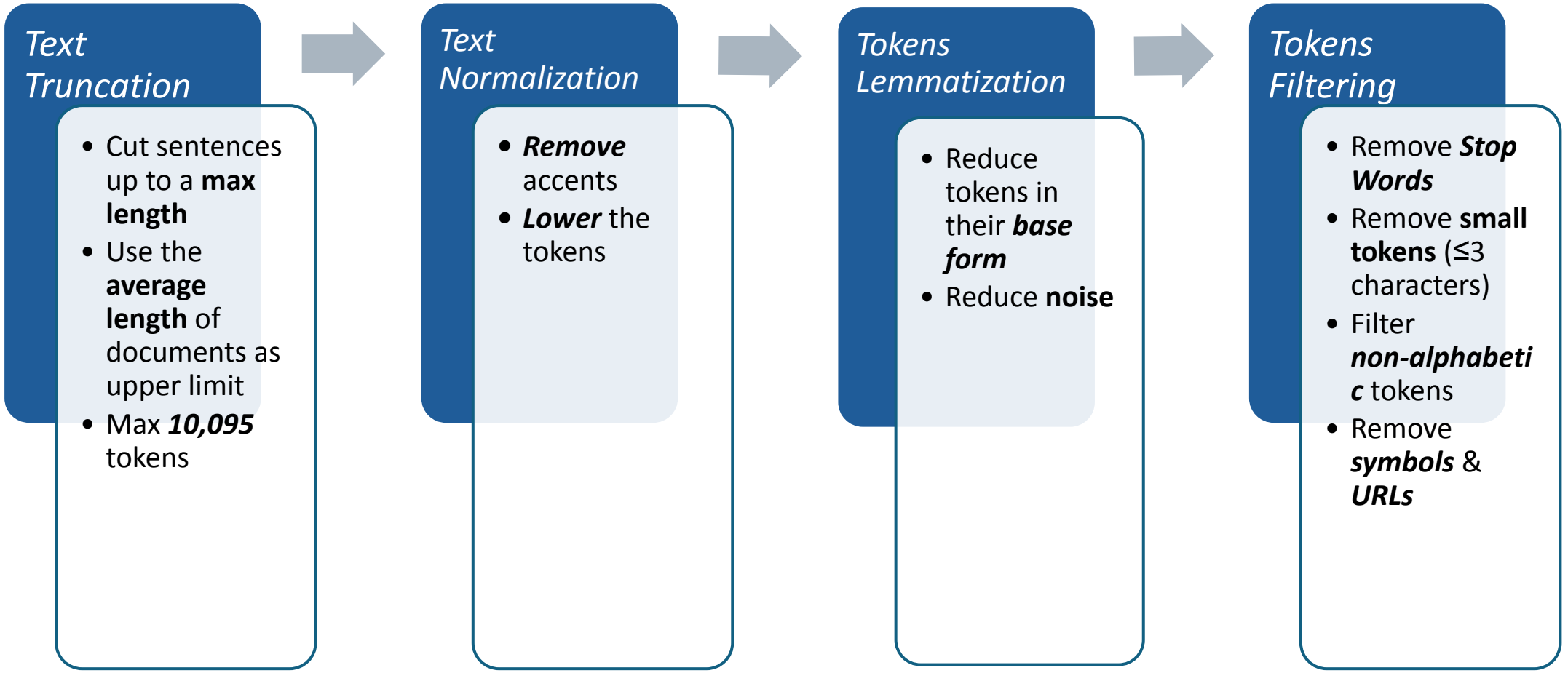
- Classes 0, 1, 2, 5, and 8 exhibit more communities.
- Complex structure with diverse subgroups.
- This indicates varied content or areas of interest within these classes.

Fewer Communities (7-9):

- Classes 3, 4, 6, and 7 show fewer communities.
- More cohesive structure with stronger interconnections among nodes.
- This suggests more homogenous content.

Data Preprocessing| Cleaning up

Utilize SpaCy to preprocess the textual representation of the domains and reduce the noise



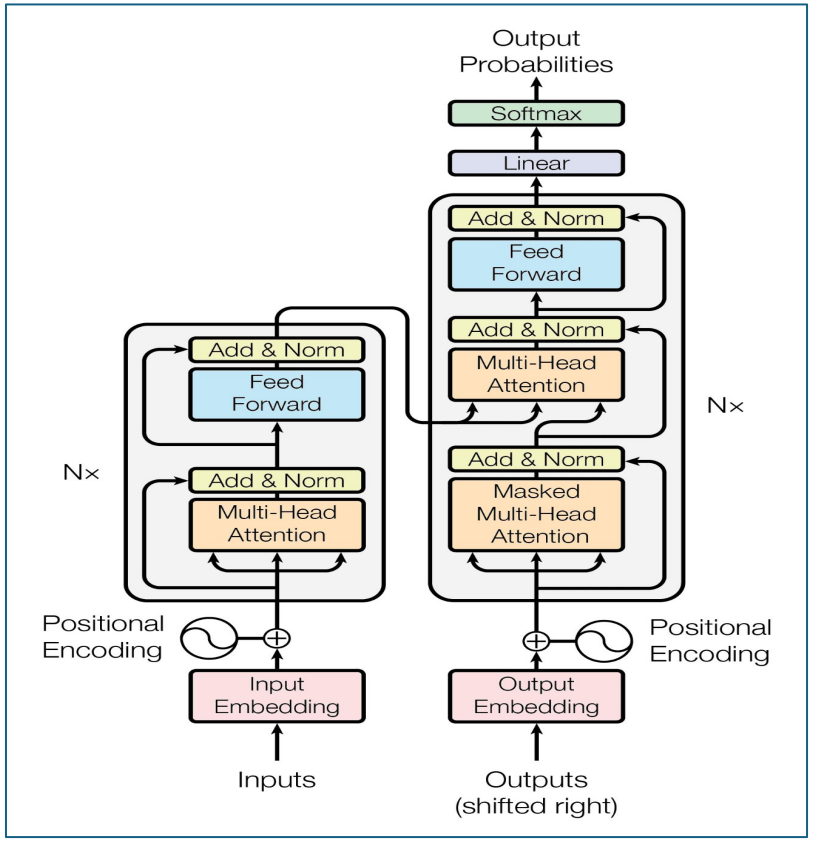
Feature Engineering

Feature Engineering | Text Representations

Creating a suitable representation for the textual information, using Greek Bert k-Sentence Transformer.

We produced our embeddings using Greek Bert k-Sentence Transformer* where k was the mean length of the Sentences.

Alternatively, we obtained the embeddings from a different BERT model** of the tokens of each sentence and produced the sentence embedding by averaging the tokens' embeddings.

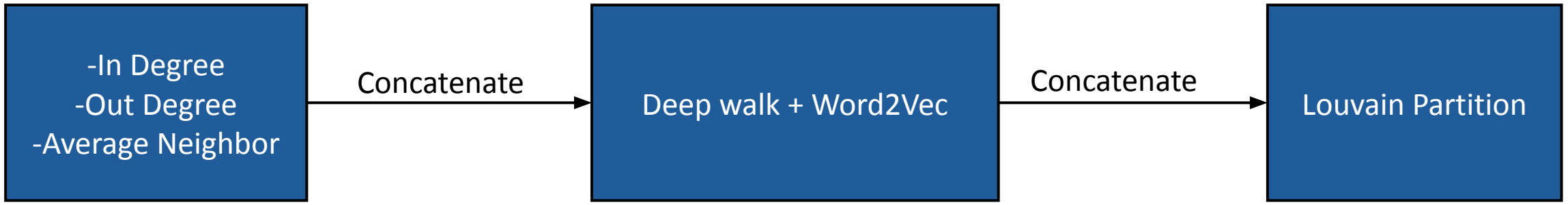


*Bert-based Greek uncased v1

**st Greek media Bert-based Greek uncased

Feature Engineering | Graph Representations

Finding Suitable and Useful Representations for the Information Contained Inside the Graph Structure.



We calculate some basic features for each node of the graph and we create first features matrix.

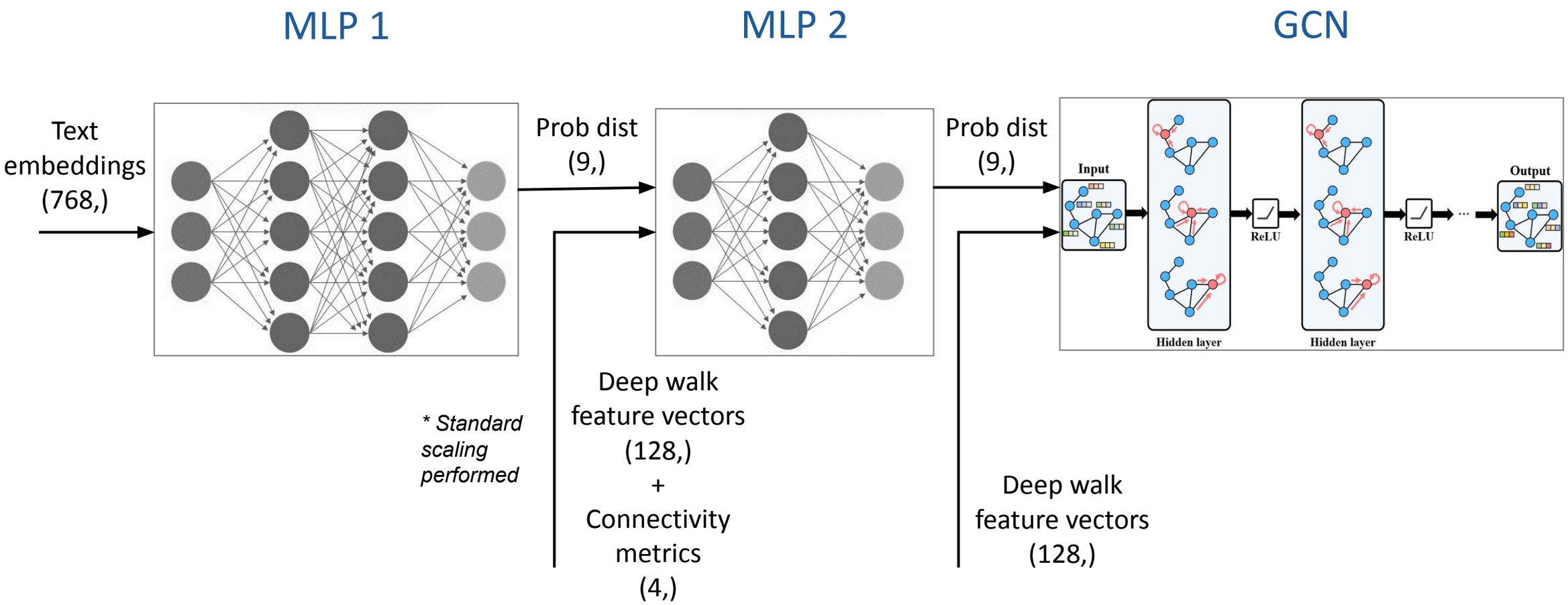
We create 20 Random Walks for each node and then we transform the lists returned from Deep Walk to embeddings, using Word2Vev (trained for all nodes).

We create clusters to classify our nodes to. We use the Louvain Method to create for each node the information of the community.

Modeling & Evaluation

Modeling | An integrated pipeline

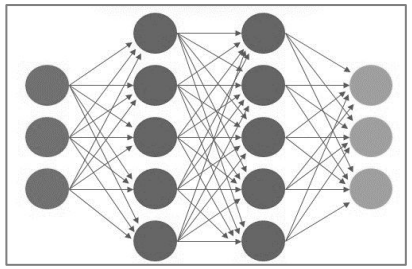
The following model architecture was developed that effectively integrates both the text and the graph data, capturing the interplay between textual content and network structure



Modeling | An MLP model for textual information representation

Architecture

MLP 1



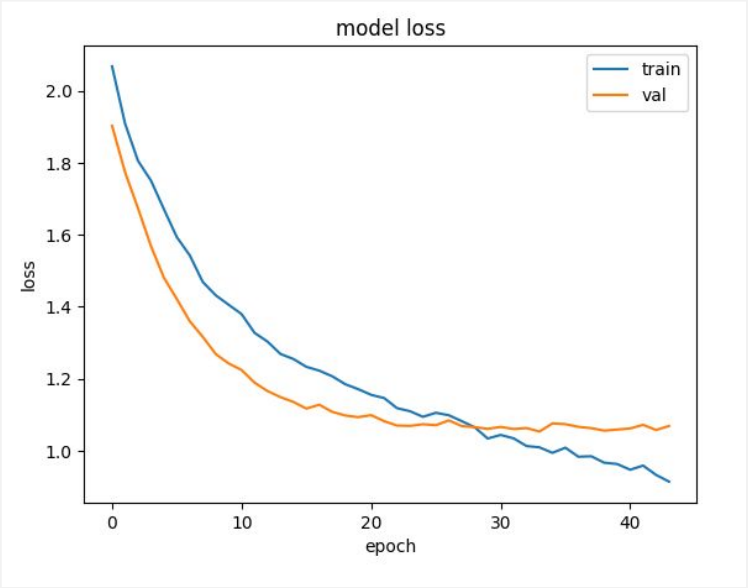
- Number of layers: 2
- Hidden layer 1: 896
- Dropout: 0.2
- Hidden layer 1 : 384
- Dropout: 0.4

** Hyper-parameter tuning performed using Optuna*
** Early stopping mechanism added*

Results

- Train loss=0.8964 and validation loss=1.0685
- Signs of overfitting
- Better performance than the baseline logistic regression model

Evaluation on Validation data (90%-10%)

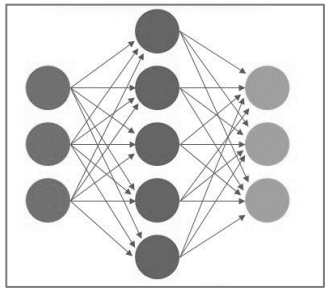


	precision	recall	f1-score	support
0	0.83	0.62	0.71	8
1	0.82	0.64	0.72	14
2	0.59	0.70	0.64	27
3	0.66	0.84	0.74	55
4	0.86	0.55	0.67	11
5	0.61	0.50	0.55	28
6	0.83	0.50	0.62	10
7	0.75	0.60	0.67	10
8	0.68	0.68	0.68	19
accuracy			0.68	182
macro avg	0.74	0.63	0.67	182
weighted avg	0.69	0.68	0.67	182

Modeling | An MLP model for graph-related information representation

Architecture

MLP 2



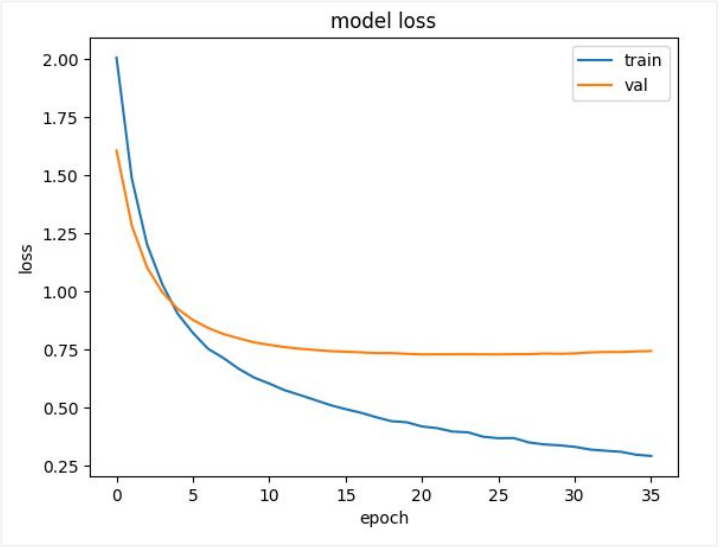
- Number of layers: 1
- Hidden layer 1: 896
- Dropout: 0.3

** Hyper-parameter tuning performed using Optuna*
** Early stopping mechanism added*

Results

- Train loss=0.2842 and validation loss=0.7445
- Overfitting – huge gap between training and validation loss
- Improved performance than the first MLP model on the validation data
- Do not generalize well on unseen test data

Evaluation on Validation data (90%-10%)

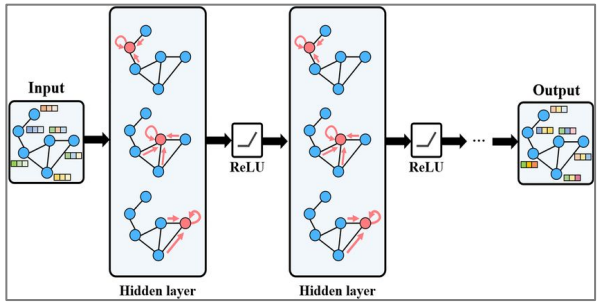


	precision	recall	f1-score	support
0	1.00	0.62	0.77	8
1	1.00	0.86	0.92	14
2	0.68	0.63	0.65	27
3	0.75	0.85	0.80	55
4	0.88	0.64	0.74	11
5	0.57	0.61	0.59	28
6	0.80	0.40	0.53	10
7	0.82	0.90	0.86	10
8	0.78	0.95	0.86	19
accuracy			0.75	182
macro avg	0.81	0.72	0.75	182
weighted avg	0.76	0.75	0.74	182

Modeling | A GCN to leverage more graph information

Architecture

GCN

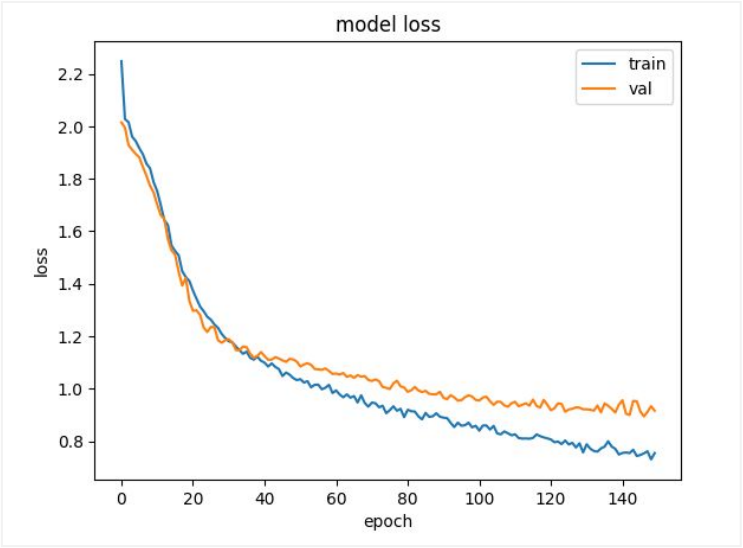


- 2 GCN layers
- Hidden layer 1: 64
- Hidden layer 2: 128
- 1 fully connected layer
- Dropout: 0.4

Results

- Train loss=0.7550 and validation loss=0.9162
- Overfitting issue is less than the previous model
- Did not perform better than the previous model on the validation data, but it achieved better results on the test data (**Private loss of 0.9196 and a Public loss of 0.8620)**

Evaluation on Validation data (90%-10%)

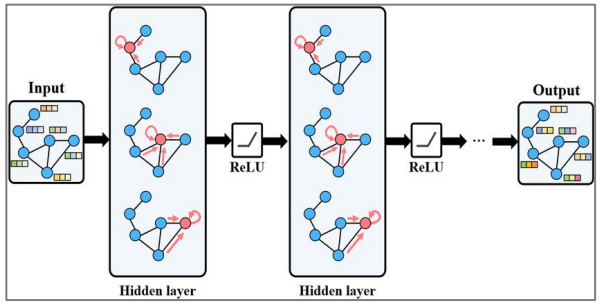


	precision	recall	f1-score	support
0	0.57	0.50	0.53	8
1	0.83	0.71	0.77	14
2	0.73	0.70	0.72	27
3	0.77	0.85	0.81	55
4	0.82	0.82	0.82	11
5	0.54	0.50	0.52	28
6	0.44	0.40	0.42	10
7	0.75	0.90	0.82	10
8	0.78	0.74	0.76	19
accuracy			0.71	182
macro avg	0.69	0.68	0.68	182
weighted avg	0.71	0.71	0.71	182

Modeling | A GCN to leverage more graph information

Architecture

GCN

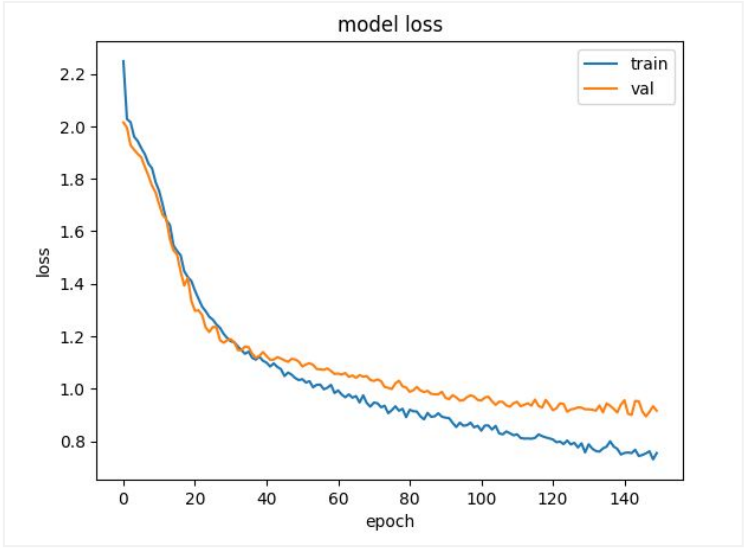


- 2 GCN layers
- Hidden layer 1: 64
- Hidden layer 2: 128
- 1 fully connected layer
- Dropout: 0.4

General Information:

- **Even Thought We Had Already Leveraged The Topological Information, When Creating Our Embeddings, We Thought that The GCN Might Reveal a More Complex Pattern, to Increase our Accuracy.**
- **Because of the Topological Nature of Our Embeddings, our GCN Both Converged and Overfitted Incredibly Early.**
- **There Was no Reason to Continue the Training Because it is Very Rare, Because of the GCN Architecture to Fall to a Local Instead of Local Minimal, for an extensive Period of Time.**

Evaluation on Validation data (90%-10%)



	precision	recall	f1-score	support
0	0.57	0.50	0.53	8
1	0.83	0.71	0.77	14
2	0.73	0.70	0.72	27
3	0.77	0.85	0.81	55
4	0.82	0.82	0.82	11
5	0.54	0.50	0.52	28
6	0.44	0.40	0.42	10
7	0.75	0.90	0.82	10
8	0.78	0.74	0.76	19
accuracy			0.71	182
macro avg	0.69	0.68	0.68	182
weighted avg	0.71	0.71	0.71	182

Other trials | Different combinations of features and models tested

Modeling trials

Training and Testing models using different feature sets **independently**:

- MLP model using the text embeddings alone
- MLP model using graph-based features alone, both with and without Louvain community labels

Experimentation with text embeddings and graph-based features combined:

- MLP model using the full text embeddings (768,) combined with graph-based features (132,)
- MLP model using the truncated text embeddings (100,) combined with graph-based features

Combined Feature Models

Feature Engineering

Evaluation of the results for every feature set combination:

- Scaling for improved convergence
- Dimensionality reduction to remove noise and keep the important information
- Various text representation techniques (e.g. fast-text Greek model, bert embeddings, bert sentence transformers)

Conclusions & Future Work

Conclusions | Challenges & Limitations

- **Graph-based** features enhanced domain representations and boosted classification accuracy by capturing local domain connectivity
- **Simpler** models, such as MLPs with 1-2 layers, performed better due to their simpler architecture
- The small dataset with limited training samples created issues such as model **overfitting**
- **GCN** did not significantly improve performance, indicating that MLPs might suffice with better features
- Improved **feature engineering** is needed, as modeling attempts did not significantly enhance accuracy

Further Improvements | What we could do better?

- Address discrepancies in validation and test losses, performing **cross validation for fine tuning**
- Handle overfitting issues using **regularization techniques** such as batch normalization layers
- Handle the imbalance dataset performing techniques such as **data augmentation or random sampling** to produce more data samples
- Explore advanced feature engineering methods such as **topic modeling** to extract topic distributions from texts and leverage the information for modeling
- Improve and **boost graph-related features** incorporating additional measures that could provide even richer representation