

A decorative background pattern consisting of a network graph. It features numerous nodes, represented by circles of varying sizes and shades of gray, connected by thin, light gray lines. Some nodes are highlighted with a blue outline, and a few are solid blue dots. The pattern is more dense on the left and right sides of the slide, with the central area being mostly white space containing the title.

# Link prediction in the Greek Web

A decorative network diagram in the top-left corner, featuring a cluster of interconnected nodes. Some nodes are represented by concentric circles, while others are simple dots. The connections are thin, light-gray lines.

1.

# The problem

Let's start by describing the problem

A decorative network diagram in the bottom-right corner, similar to the one in the top-left, showing a cluster of interconnected nodes with some concentric circles and thin gray lines.

## Link prediction

The setting involves a directed graph which nodes correspond to webpages and edges represent a link from one page to another

## Edge Classification

Since our task is to predict edges we will use edges as observations to extract meaningful features. An edge consists of two nodes and the directed link between these two.






## Dataset

### **Graph Data**

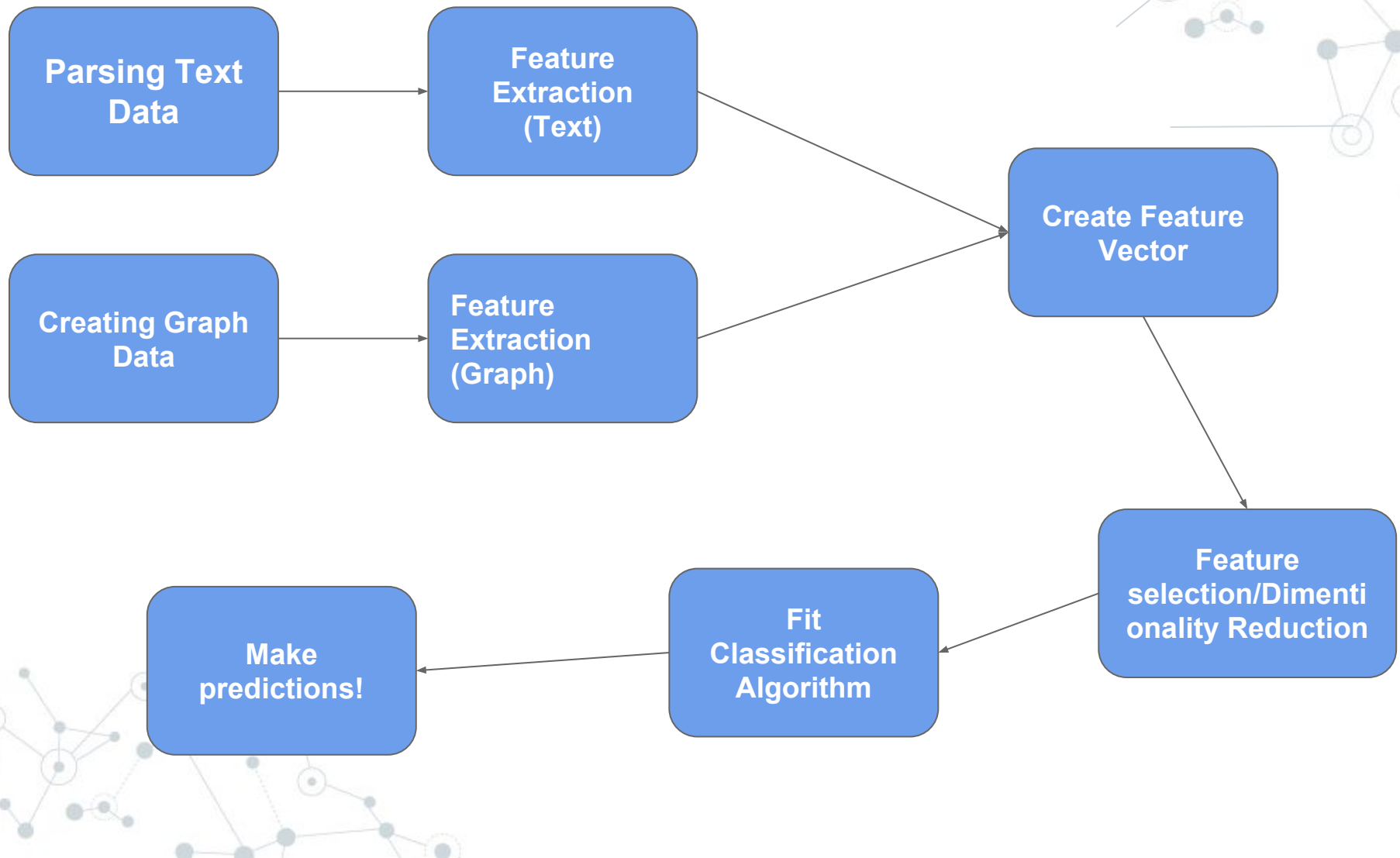
We have 2683 edges between that show links between 2041 webpages.

### **Text Data**

We also have some raw texts from each webpage.



# Classification pipeline





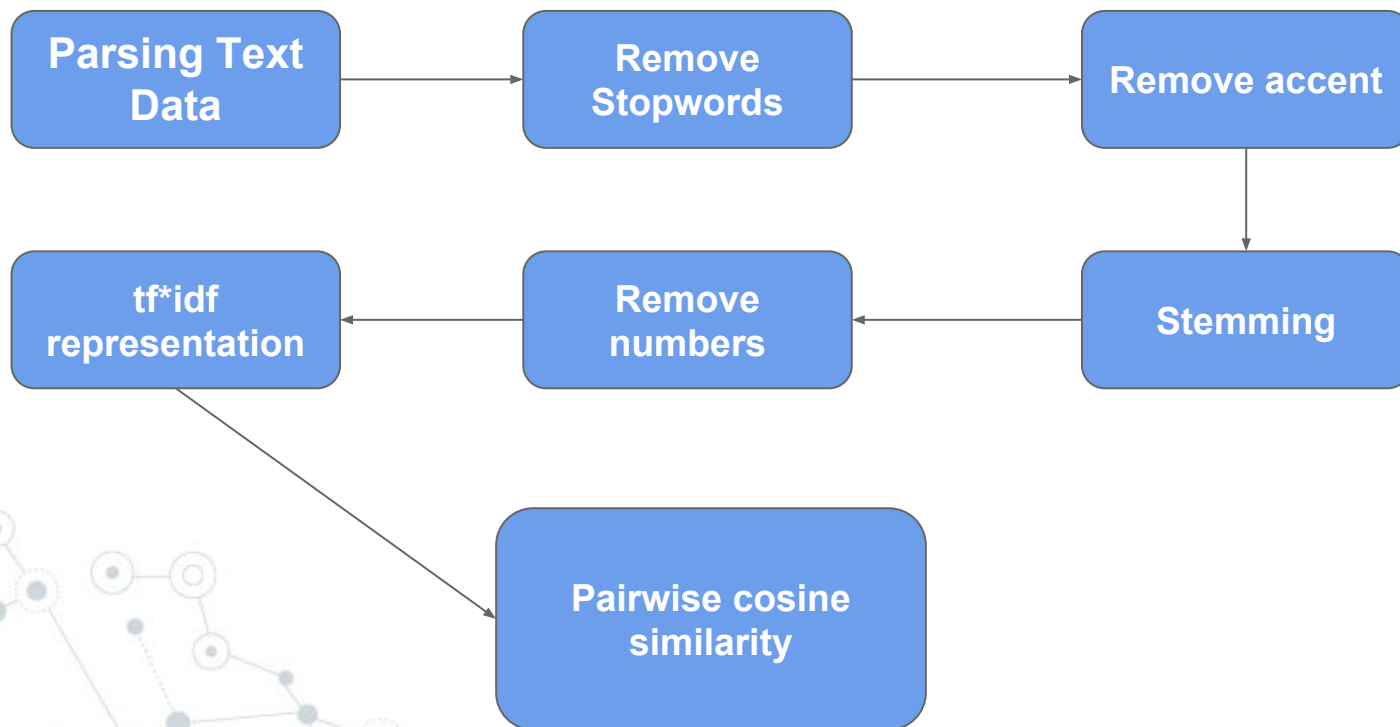
2.

# Feature Extraction



## Text Similarity

We compute text similarity between each pair of nodes





## Graph Features

- ◎ Source out-degree
- ◎ Destination in-degree
- ◎ # common neighbors
- ◎ # of second neighbors



## Graph Features: Centrality Measures

- ◎ Pagerank
- ◎ Eigenvector
- ◎ Betweenness
- ◎ Closeness
- ◎ Katz



## Graph Features

- ◎ K-cores
- ◎ # of triangles
- ◎ Simrank
- ◎ Adamic adar
- ◎ Jaccard coefficient
- ◎ Preferential Attachment
- ◎ Resource allocation index






## Graph Features: Community Detection

After applying the louvain method on the graph we obtain 605 communities. By exploring them we deal with some interesting communities like the following:

1: techgear.gr , prasinanea.gr , news247.gr , 24media.gr , oneman.gr , sport24.gr , huffingtonpost.gr , macuser.gr , ladylike.gr , contra.gr , redplanet.gr , olapaok.gr

2: arkadiapress.gr , arcadia-news.gr , arcadiaportal.gr , spartakos-dei.gr , kafeneio-megalopolis.gr , e-gortynia.gr


3: shootandgoal.com , briefingnews.gr , ikypros.com , tothemaonline.com , riknews.com.cy , lay-out.gr , dialogos.com.cy , onlycy.com , cyprusnet.gr , pafospress.com , sae.gr , cyprusrodos.gr , offsite.com.cy , cyprusnews.eu





## Graph Features: Community Detection

We use 2 features that have to do with the extracted communities

- ◎ # of common neighbors in dst/src communities
  - ◎ If src/dst co-exist in a community (boolean)
- 

A decorative network diagram in the top-left corner, consisting of various sized circles (nodes) connected by thin lines (edges). Some nodes are solid grey, while others are hollow with a grey outline. The network is dense and irregular.

3.

# Feature Selection

A decorative network diagram in the bottom-right corner, similar to the one in the top-left, featuring a cluster of interconnected nodes and edges, with some nodes highlighted in solid grey.

## Feature Selection

We want to select the smallest subset of features that better describe our data. These features better describe the variation in the response (if an edge should exist or not).

We are interested in examining the information gain of each candidate to answer the following:

**How much does it help to reduce our uncertainty about the correct class?**



## Feature Selection

There are four approaches that can be followed for feature selection:

- © Manual Feature Selection. We calculate statistics about each feature (min-max values, mean, variance). We discard features with very low variance (up to a threshold). Afterwards we also calculate the pearson correlation between each pair of features and choose to remove one of the features in each pair that shows a high correlation. High correlation between two features can make our model really unstable.





## Feature Selection

- ◎ Use the `feature_selection` package available in `scikit learn` to do feature selection.
- ◎ Use a model that provides feature importance (eg Random Forests).
- ◎ Do no feature selection and use all features in a neural network or a tree based model



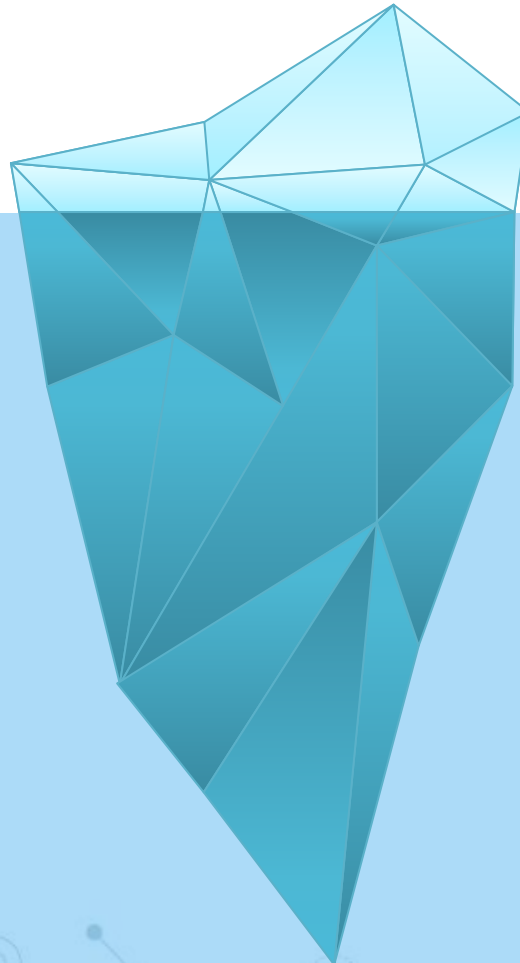
# Considerations

## Feature collinearity

Good feature subsets contain features highly correlated with the classification, yet uncorrelated to each other

## Complexity

We could examine other features like edge betweenness by adding and removing a candidate edge each time, but it would be too time-consuming!



## Imbalanced dataset

We have a big number of non existing edge, opposed to the ones that actually exist

## The Real World

Although we may find high similarity between two nodes eg. by examining text features, in the real world it would be unlikely for example that news247.gr has a link to newsit.gr!

A decorative network diagram in the top-left corner, consisting of various sized circles (nodes) connected by thin lines (edges). Some nodes are solid grey, while others are hollow with a grey outline. The connections form a complex, branching structure.

# 4. **Classification**

A decorative network diagram in the bottom-right corner, similar to the one in the top-left, featuring a cluster of interconnected nodes and edges, with some nodes highlighted in solid grey.

## Classification Algorithms/Techniques

- ◎ Logistic Regression
- ◎ SVM
- ◎ Multilayer Perceptron
- ◎ XGBoost
- ◎ Random Forest
- ◎ Gradient Boosting Classifier
- ◎ Voting Classifier

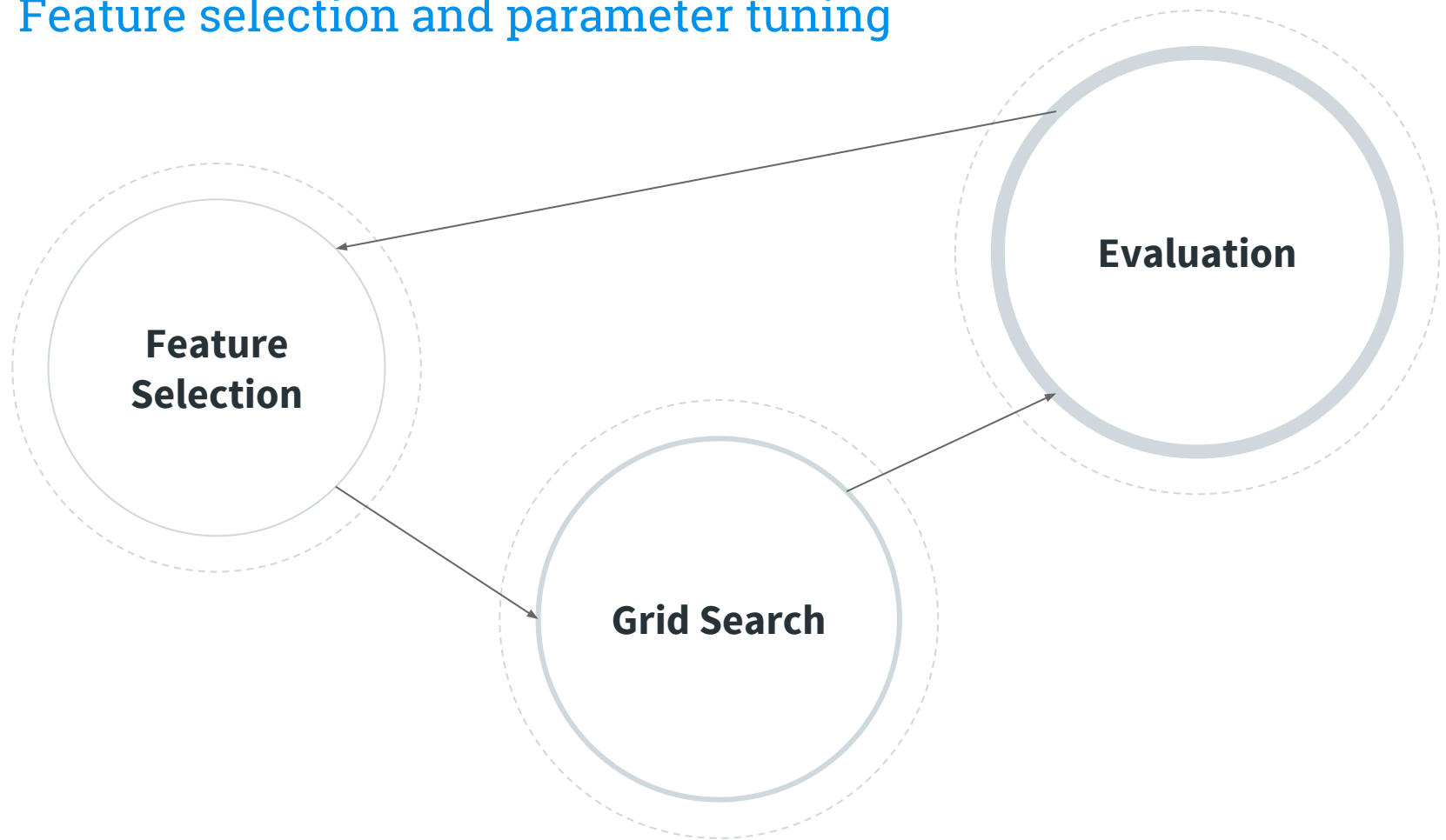


## Challenges

- ◎ Avoid overfitting (regularization)
- ◎ Hyper-parameter tuning (Grid Search)



## Feature selection and parameter tuning



## And tables to compare data

---

### Accuracy

Logistic Regression

**6.4%**

SVM

**8,6%**

XGBoost

**10,15%**

Random Forest

**10,31%**

---





**Thanks!**

**Any questions?**

