H-index Prediction

Paul Theron

I. What to aim at

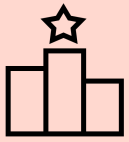
- Investigating whether the **collaboration patterns** of an author are **good predictors** of the author's short term future *h*-index.
- **Crucial impact** to enable professor recruitment
- **Regression target:** *h*-index prediction

II. Exploration data analysis

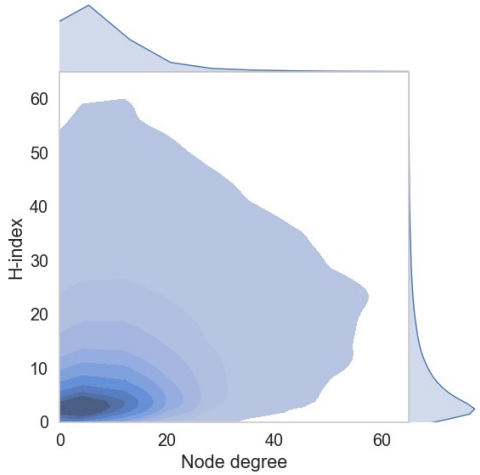
- **217,801 authors** with their co-authorship, abstracts and 5 most cited publications.
- Main challenge: **3 very different datatypes**

Partial co-authorship graph (1.7M edges)

Full abstracts text (NLP task with 624.181 abstracts)



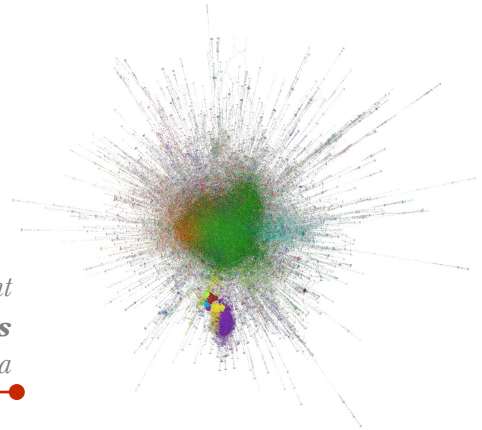
Popularity of the 5 most cited papers per author



Node density over number of co-authorship vs *h*-index

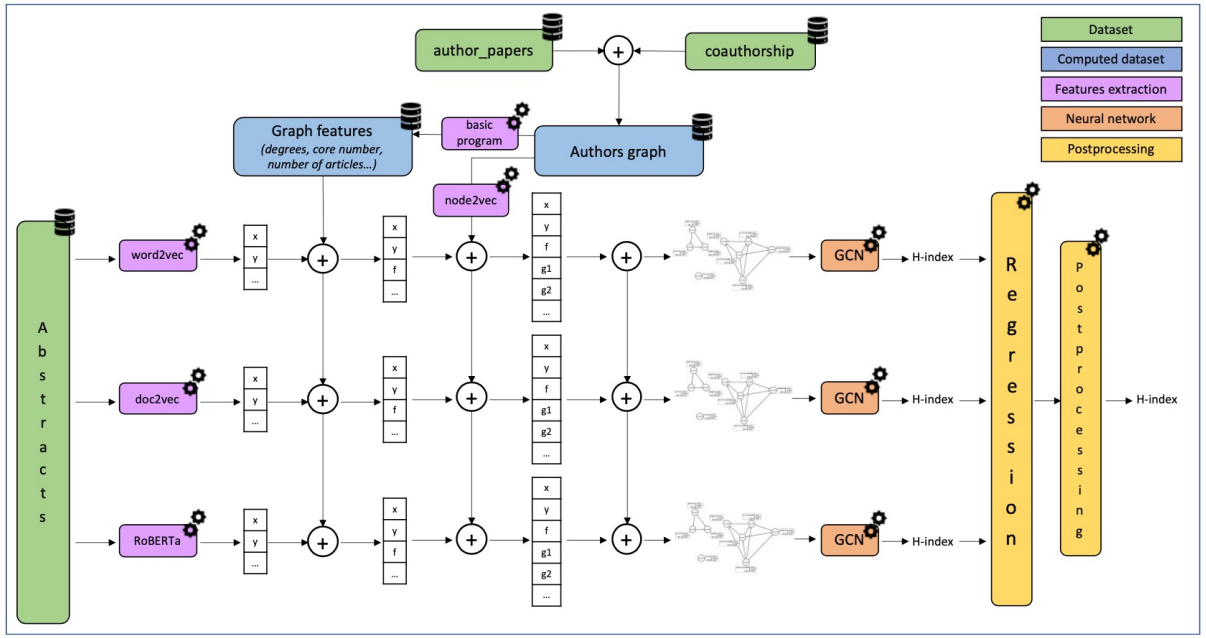
Imbalanced h-index repartition *and* ***high variability*** *due to outliers*
Average h-index: 10.08

Highlighting the different ***scientific fields*** *represented in the data*



Graph visualization of the dataset with visual **clustering of the nodes**

III. Global pipeline approach

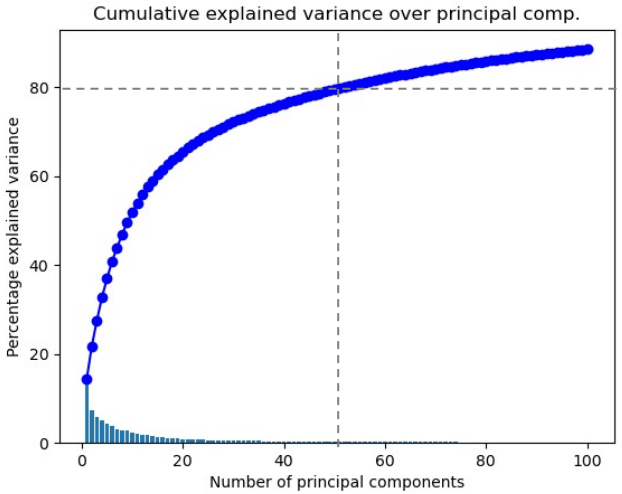


H-index Prediction

Paul Theron

IV. Preprocessing task

- Performed network science centralities and metrics to **capture different dimensions of a node's impact on the graph**, Page Rank score, core centrality
- Performed state of the art 300 dimension **embedding of abstract**, after stop words removal and cleaning
- **Performed a PCA** at 80% threshold to avoid the curse of dimensionality



V. Model tuning

- Used 2 layers neural networks for prediction, created training & testing **masks** to fit the **graph architecture**
- Compared different architectures: **Multi layer perceptron, Graph convolutional network, Sage Graph**.
- Hyperparameter tuning with **Randomized Search**

Best model

Sage Graph
7.74 Root mean squared error
with a 10.08 average h-index

VI. Postprocessing task

- Normalized predictions to fit **statistic distribution** of training set.
- **Combined results from 3 different models** to improve predictions

Best improvement

7.61 Root mean squared error

VII. What was done well

- Construction of an **efficient machine learning pipeline**
- Assessing the issues of ***h*-index imbalance** and **curse of dimensionality** during preprocessing
- Efficiently combining **diverse types of data**

VIII. Margin for improvement

- Dig deeper into the **high-level features**, which could be extracted from the graph (Shannon Entropy, community centrality)
- Use **convolutional neural networks** or **sentence embeddings** to create efficient embeddings for abstracts
- Explore other prediction models such as **Light gradient boosting.**