

# Grafos de Dependências no Ecossistema de IA em Python



Alice Maria Fonseca Victorino Freire  
Projeto 1: Algoritmos e Estruturas de Dados II

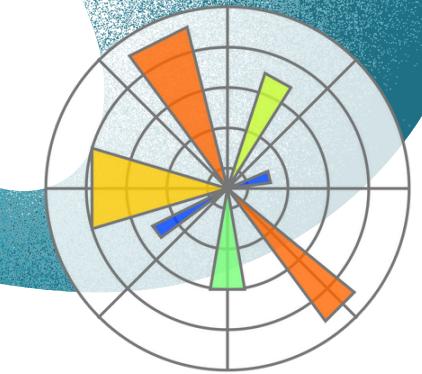
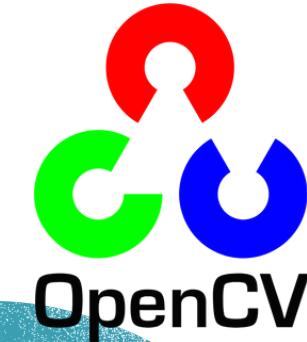
# Porque mapear o ecossistema de IA?

O ecossistema de IA em Python é uma rede complexa. Este projeto usa a Teoria dos Grafos para mapear essa estrutura e responder perguntas-chave como:

- Quais são as bibliotecas mais centrais e influentes?
- Existem "tribos" ou sub-ecossistemas concorrentes?

The screenshot shows a code editor interface with two tabs. The left tab is titled 'streamlit\_control.py' and contains a single line of code: 'import streamlit as st'. The right tab is titled 'requirements.txt' and lists the following dependencies:

```
1 altair
2 boto3
3 ccxt
4 google
5 gspread
6 pandas
7 regex
8 requests
9 s3fs==0.4.*
10 streamlit==1.1.*
```



## Objetivos:

- 1 Modelar a rede de dependências como um grafo.
- 2 Analisar a estrutura com métricas-chave (grau, densidade, assortatividade).
- 3 Descobrir as comunidades (clusters) e as "pontes" entre elas.

# Metodologia e Ferramentas

## 1 Coleta e Modelagem (NetworkX)

As dependências de cada biblioteca foram extraídas via pip e modeladas como um grafo direcionado, onde uma aresta  $A \rightarrow B$  significa "A depende de B".



## 2 Análise e Métricas (Jupyter & NetworkX)

O grafo foi analisado quantitativamente, focando em centralidade de grau, densidade para medir a coesão, e assortatividade para entender a arquitetura da rede.

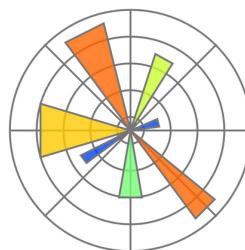


## 3 Visualização e Comunidades (Gephi)

O software Gephi foi usado para a visualização criativa, aplicando o layout ForceAtlas2 para agrupar nós e algoritmos de detecção de comunidades para revelar os clusters.



# Bibliotecas de Visualização de Dados



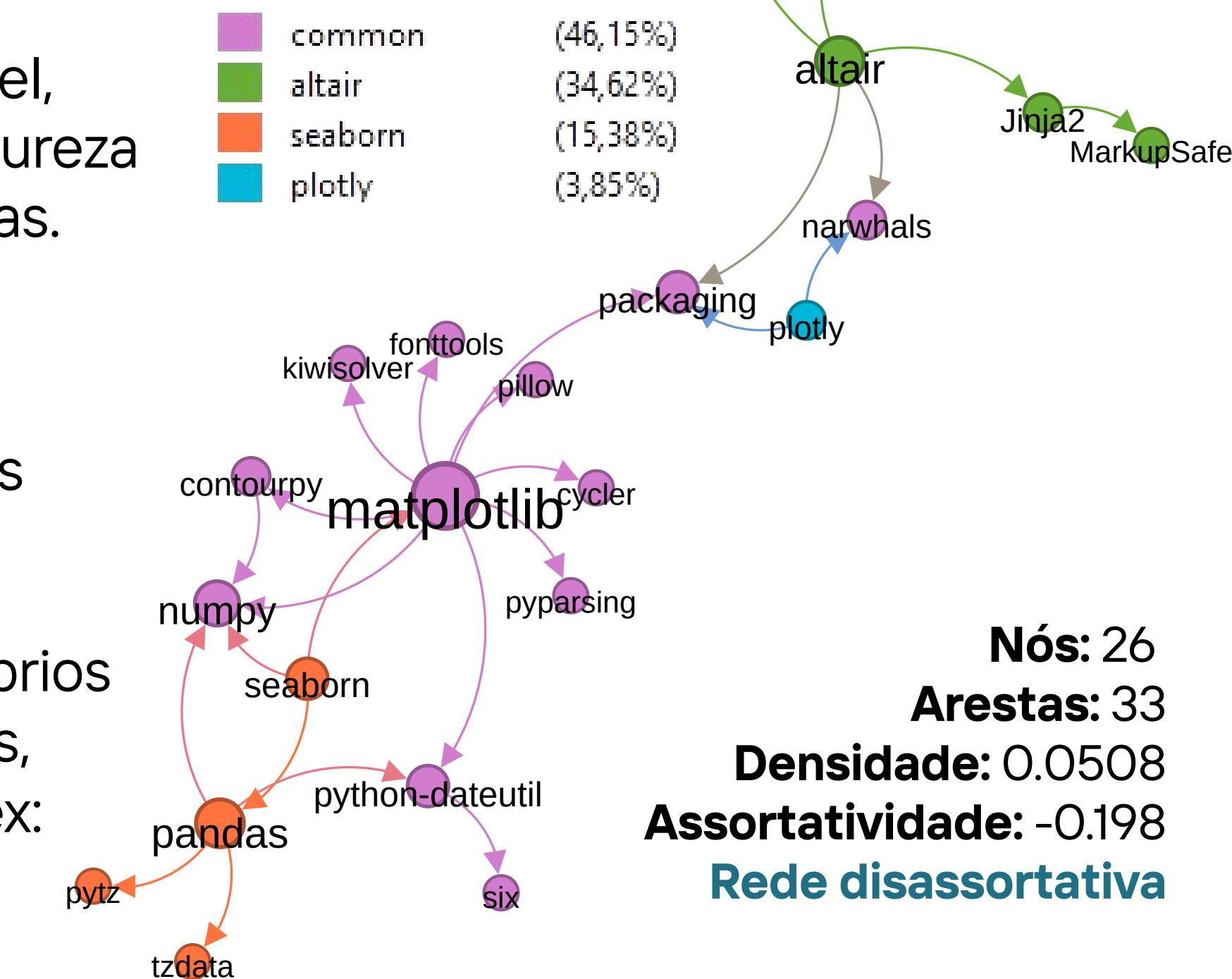
**Matplotlib:** Atua como a base de baixo nível, com o maior número de conexões. Sua natureza é ser uma fundação para outras ferramentas.



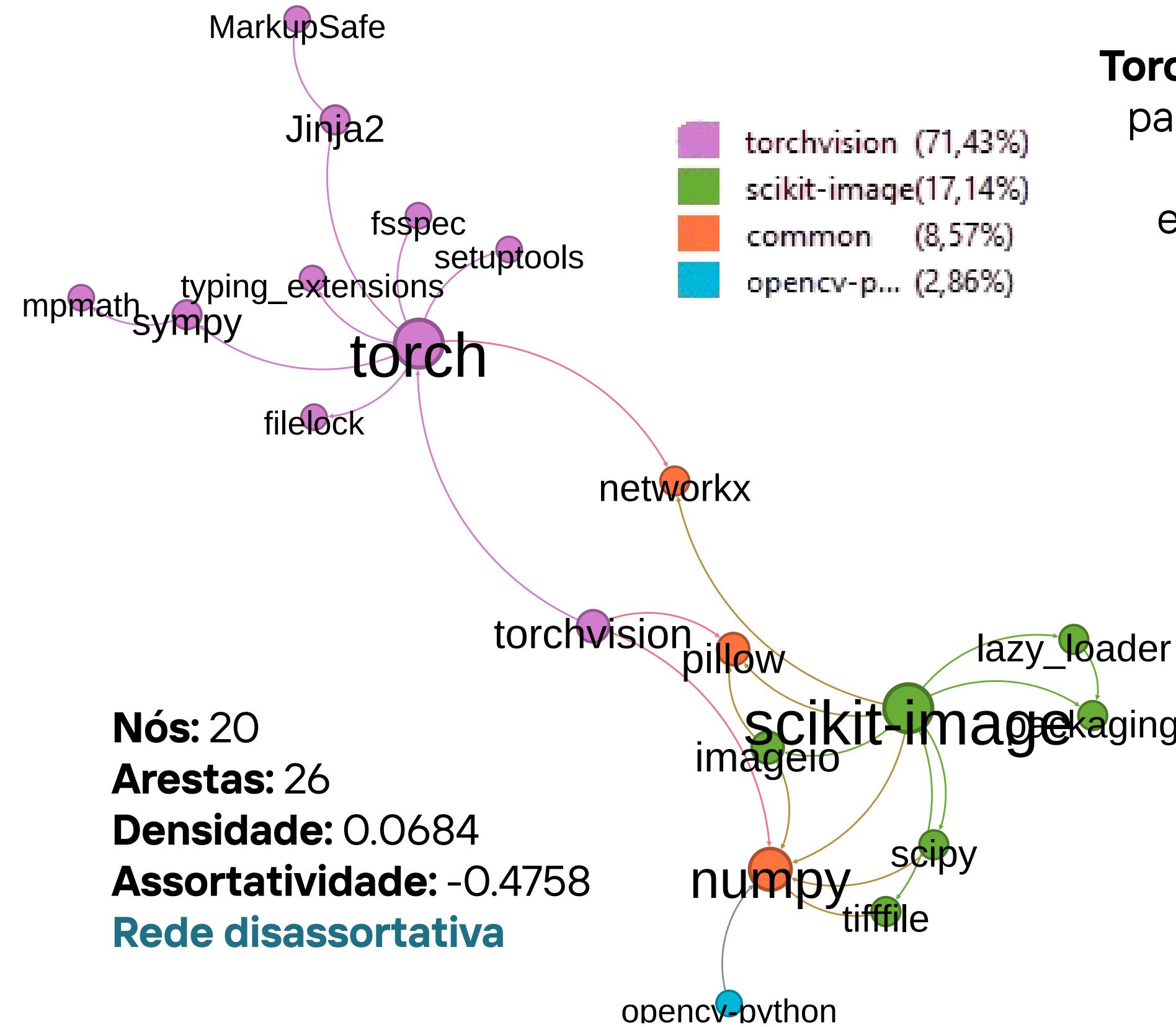
**Seaborn:** Uma interface de alto nível que depende diretamente do Matplotlib para simplificar a criação de gráficos estatísticos complexos.



**Altair e Plotly:** Formam ecossistemas próprios com poucas dependências compartilhadas, refletindo filosofias de design diferentes (ex: abordagem declarativa do Altair).



# Bibliotecas de Visão Computacional



**Torchvision:** Atua como a interface de alto nível para visão com torch, mas funciona como uma ponte, pois depende de bibliotecas do ecossistema clássico (numpy e Pillow) para o pré-processamento de dados.



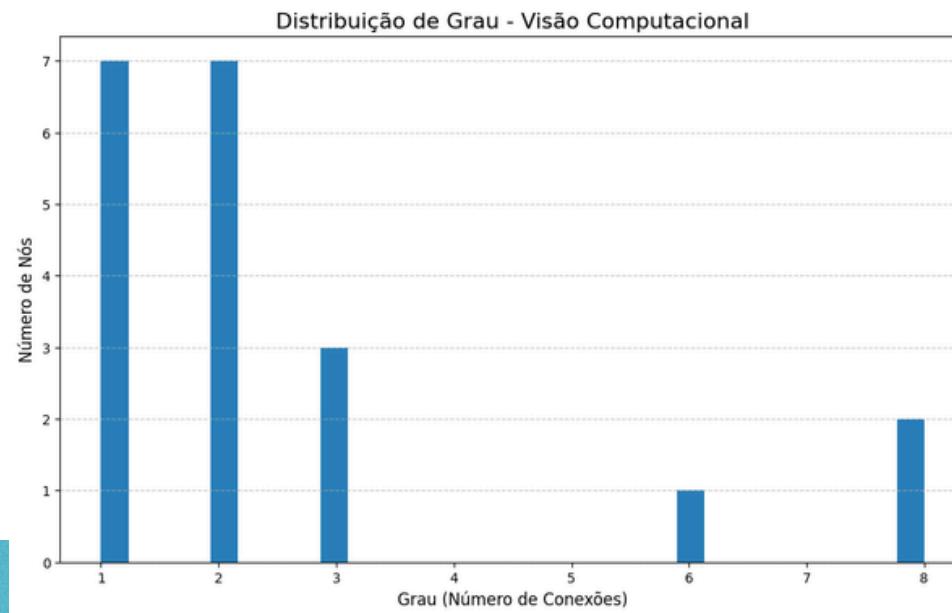
**Scikit-image:** O núcleo do ecossistema clássico de processamento de imagem (verde), fornecendo as ferramentas fundamentais para algoritmos tradicionais.



**OpenCV:** Outra importante biblioteca do ecossistema clássico, que se conecta e depende do **numpy** para suas operações fundamentais.



# Bibliotecas de Visão Computacional Modularização

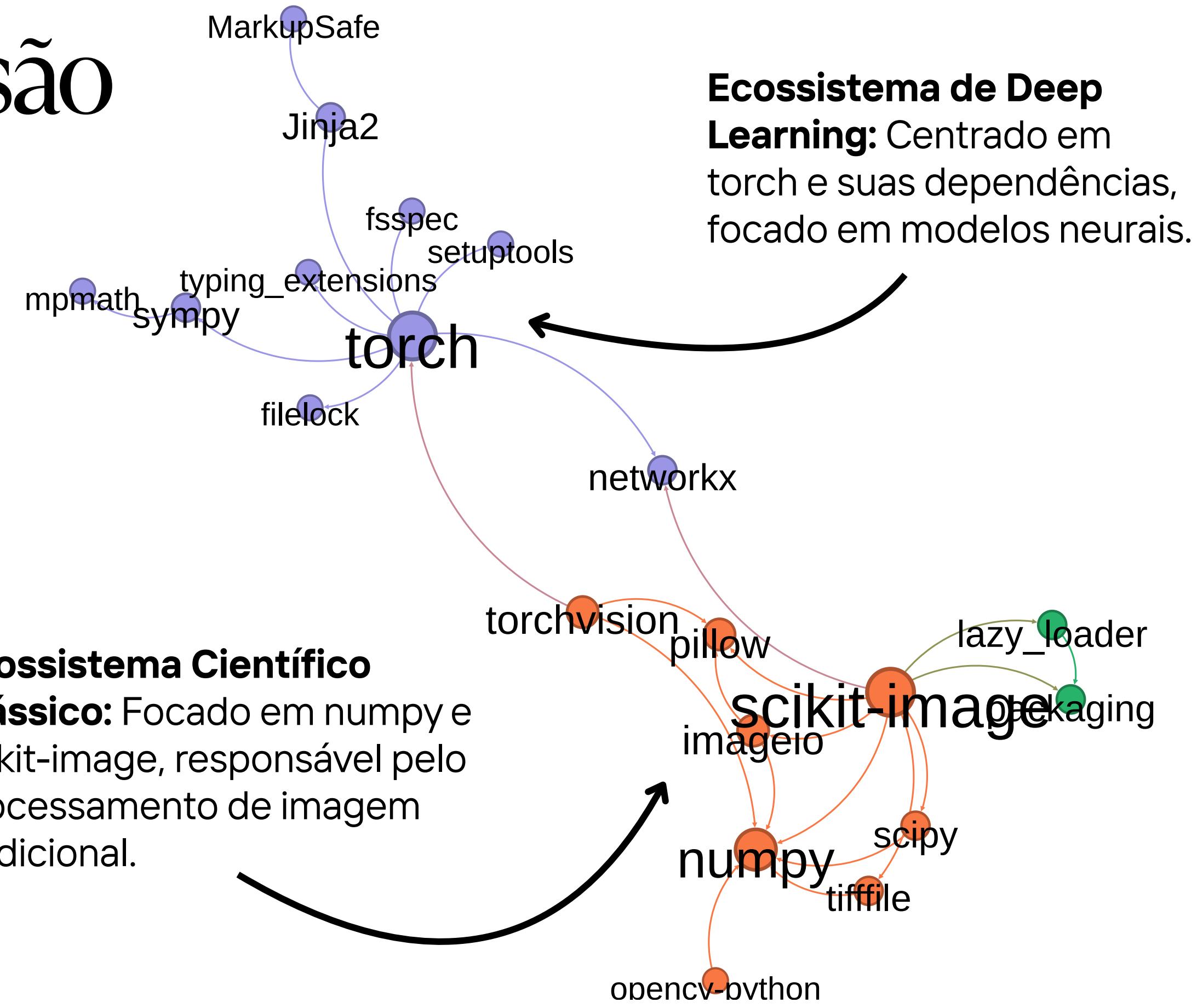


## Distribuição de grau:

- **Padrão de cauda longa:** a maioria dos nós (14) possui 1-2 conexões
- **Hubs principais:** scikit-image e torch (8 conexões); numpy (6 conexões)

## Ecossistema Científico

**Clássico:** Focado em numpy e scikit-image, responsável pelo processamento de imagem tradicional.



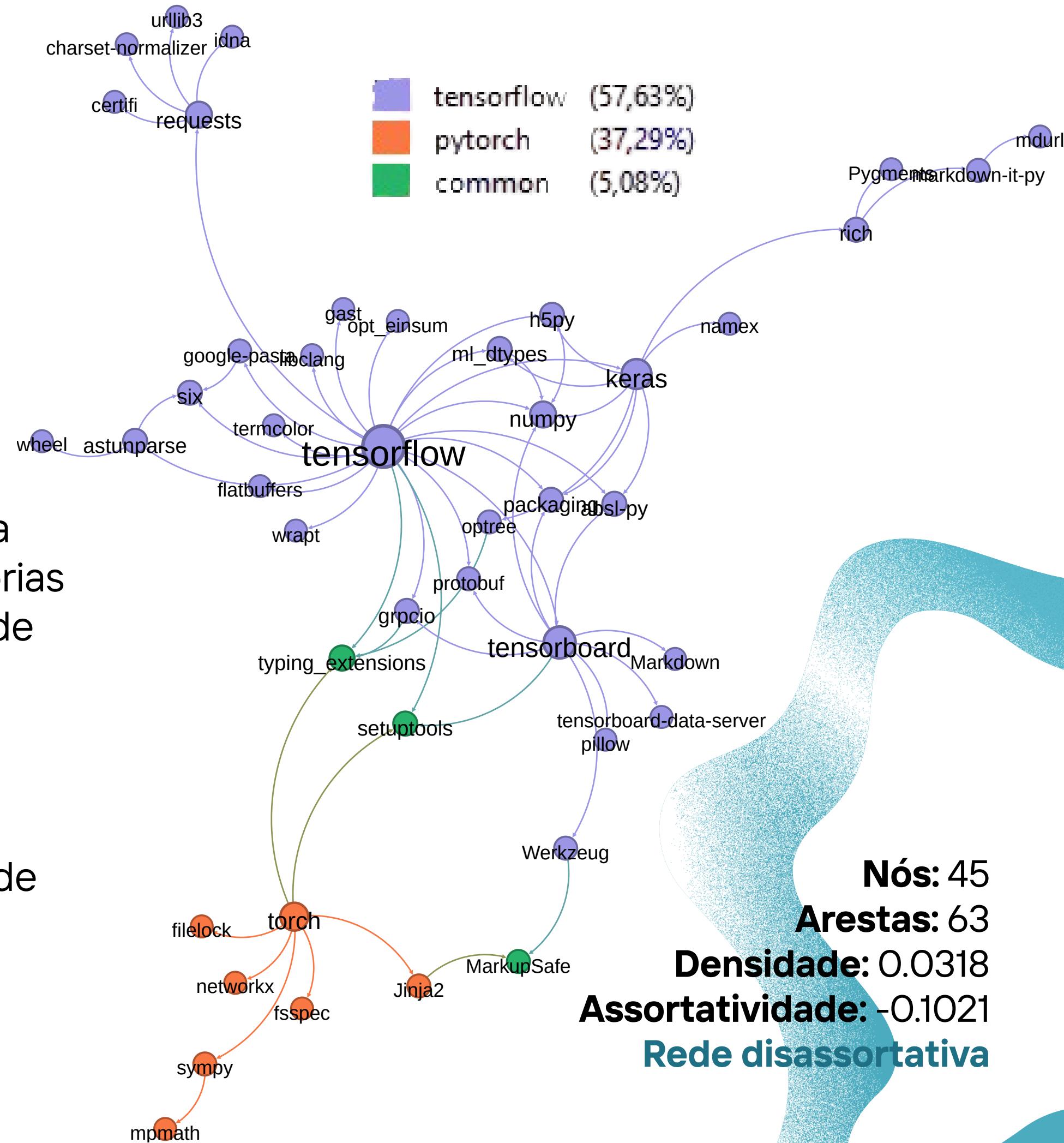
# Bibliotecas de Deep Learning



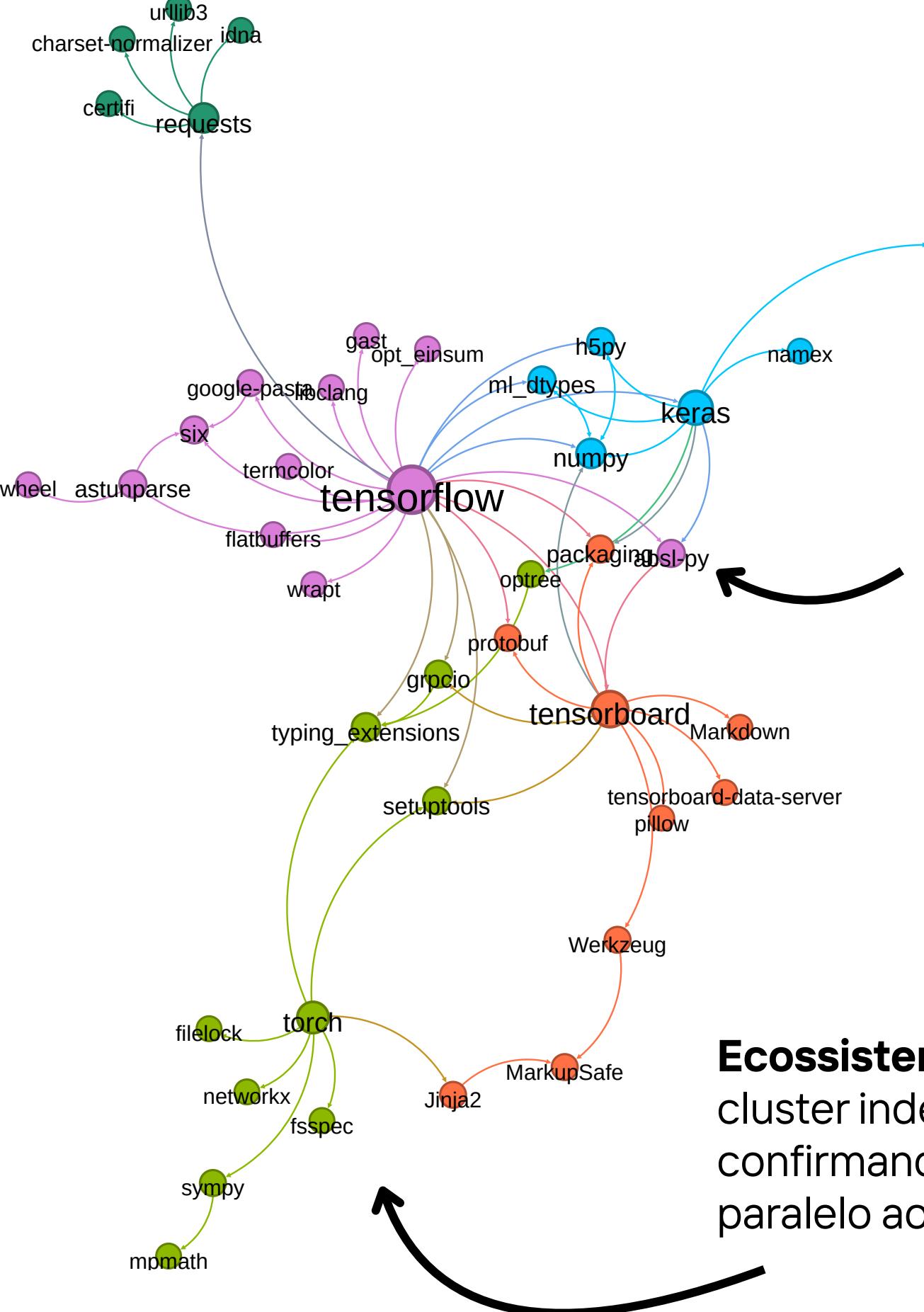
**TensorFlow:** O hub dominante do grafo (cluster roxo), formando um ecossistema vasto e autocontido que inclui suas próprias APIs de alto nível (Keras) e ferramentas de visualização (TensorBoard).



**PyTorch:** O centro do segundo grande ecossistema (cluster laranja), operando de forma paralela e independente do TensorFlow, com sua própria cadeia de dependências especializadas.

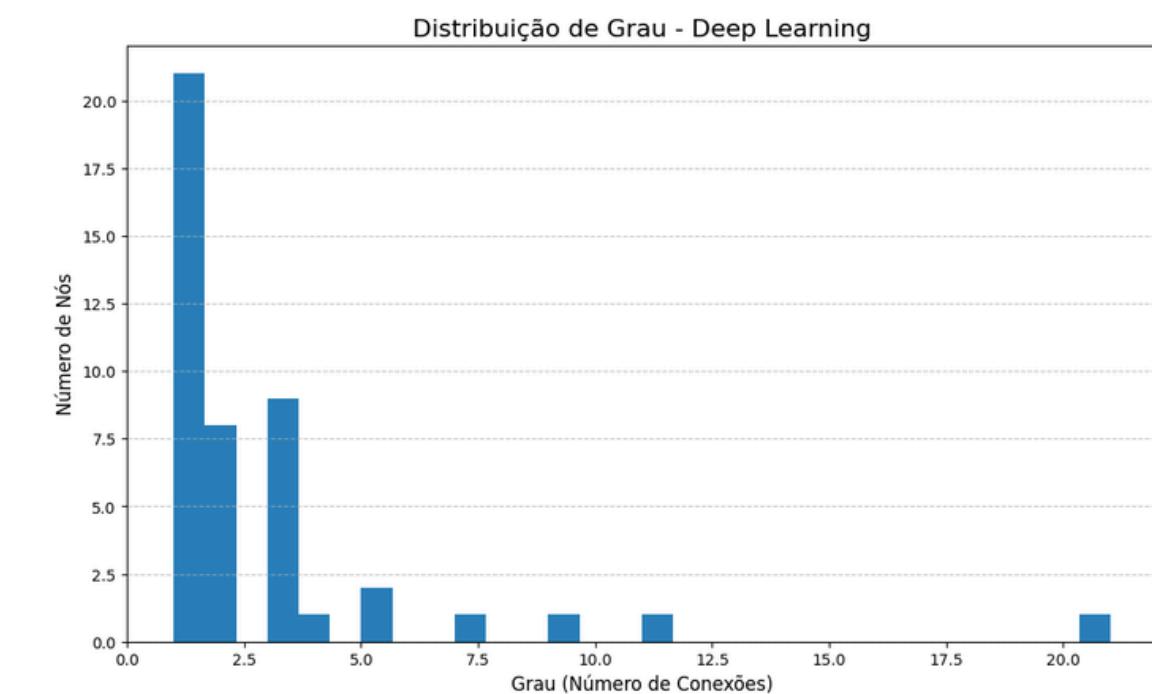


# Bibliotecas de Deep Learning



**Ecossistema TensorFlow (Roxo, Azul, Laranja):** Um "mega-cluster" composto por três comunidades interdependentes: o núcleo tensorflow, a API Keras e a ferramenta de visualização TensorBoard.

**Ecossistema PyTorch (Verde):** Aparece como um cluster independente e autossuficiente, confirmando sua posição como um universo paralelo ao TensorFlow.



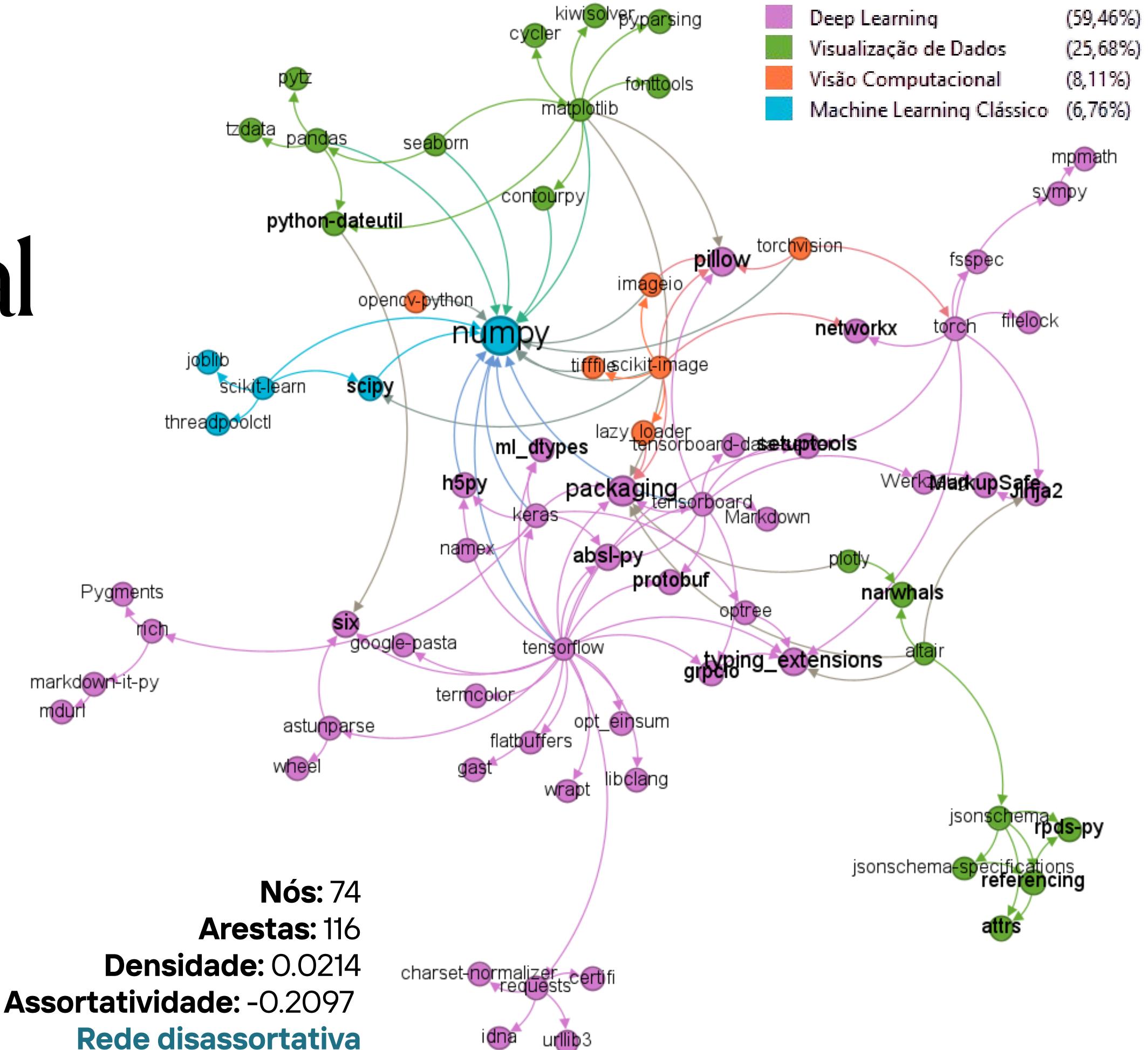
## Distribuição de grau:

- Padrão de cauda longa mais extremo que os anteriores: quase metade dos nós ( $>20$ ) possui apenas 1 conexão
  - Hub dominante: TensorFlow com 21 conexões

# Ecossistema de Inteligência Artificial

- **Núcleo Clássico (ML, Visão, Visualização):** Forma um ecossistema denso e fortemente interligado.
- **Deep Learning (Roxo):** É a maior e mais complexa categoria, mas demonstra clara dependência do núcleo clássico.
- **Numpy:** Atua como o hub universal que conecta e unifica todos os diferentes temas. Tem o **maior grau de entrada**.

**Nós:** 74  
**Arestas:** 116  
**Densidade:** 0.0214  
**Assortatividade:** -0.2097  
**Rede disassortativa**



# Ecossistema de Inteligência Artificial

- **Núcleo Científico (Verde):** A modularidade revela um cluster central massivo que une Manipulação de Dados, ML Clássico e Visualização, servindo como a fundação de toda a rede.
  - **Ecossistemas de Deep Learning (Roxo, Azul):** TensorFlow e PyTorch formam seus próprios clusters grandes e separados, que se conectam ao núcleo científico para tarefas de dados.
  - **Clusters Independentes:** Ferramentas com arquiteturas únicas, como requests e rich, aparecem como comunidades próprias e isoladas.

