**RELATÓRIO EXPLICATIVO DO CÓDIGO: GRAFO DE INCIDÊNCIA**

**1. Introdução**

Este relatório tem como objetivo explicar, de forma clara e estruturada, o funcionamento de um script desenvolvido em Python que realiza a visualização de um **grafo de incidência bipartido**. O grafo conecta duas categorias distintas de entidades: **pessoas** (no caso, nomes fornecidos em uma lista) e **bandas musicais favoritas**. O foco está em destacar bandas que foram mencionadas por mais de uma pessoa, promovendo uma representação gráfica limpa e significativa.

**2. Bibliotecas Utilizadas**

O código utiliza as seguintes bibliotecas:

* pandas: para manipulação e estruturação dos dados em forma de tabela (DataFrame).
* networkx: para construção e manipulação do grafo.
* matplotlib.pyplot: para exibição gráfica do grafo.
* ast: para converter strings representando listas em objetos de lista reais.
* collections.Counter: para contar a frequência de cada banda.
* networkx.drawing.layout.bipartite\_layout: para posicionamento automático dos nós bipartidos.

**3. Estrutura do Código**

**3.1 Definição dos Dados**

Os dados consistem em um dicionário Python com duas chaves: nome e bandas. Cada entrada da chave nome representa uma pessoa, e a entrada correspondente na chave bandas é uma lista com os nomes das bandas favoritas dessa pessoa. Esses dados são convertidos em um DataFrame utilizando a biblioteca pandas.

Além disso, como as listas de bandas estão inicialmente armazenadas como strings (ex: "['rock', 'metal']"), é necessário convertê-las em listas reais utilizando ast.literal\_eval().

**3.2 Filtragem das Bandas Populares**

Em seguida, o código coleta todas as bandas mencionadas por todas as pessoas e usa o Counter para contar a frequência de cada banda. Apenas as bandas citadas por **duas ou mais pessoas** são consideradas para o grafo final. Isso ajuda a reduzir a poluição visual do grafo e a destacar interesses musicais compartilhados.

**3.3 Construção do Grafo**

É criado um grafo não direcionado (nx.Graph()) com estrutura **bipartida**:

* Os **nós do tipo 0** representam pessoas.
* Os **nós do tipo 1** representam bandas populares.

Para cada pessoa, é adicionado um nó, e para cada banda popular citada por essa pessoa, é criada uma **aresta** entre os dois.

**3.4 Definição de Layout e Visualização**

A disposição dos nós é feita usando bipartite\_layout, que organiza os dois grupos (pessoas e bandas) em lados opostos do grafo. Em seguida:

* A cor dos nós diferencia os dois grupos (azul para pessoas, verde para bandas).
* O **tamanho dos nós** é proporcional ao número de conexões (grau).
* O grafo é exibido com matplotlib de forma legível e colorida.

**4. Conclusão**

O script fornece uma representação gráfica eficiente e visualmente clara da relação entre indivíduos e suas bandas favoritas. Ele permite identificar, de forma rápida, **quais bandas são mais populares dentro do grupo**, além de visualizar **quem compartilha gostos musicais**. Essa técnica pode ser aplicada em diversos contextos onde se busca entender a conexão entre dois grupos distintos de entidades, como alunos e disciplinas, clientes e produtos, entre outros.