MOVIE CATALOG
DATASET

Iniciar

https://github.com/MovieCatalogDS/MovieCatalogDS

Motivação e Contexto

- Filmes: meio de entretenimento muito popular.
- Serviços de streaming de vídeo tentando pegar a sua fatia em um mercado crescente.
- Grande quantidade de serviços, além da popularidade das franquias e universos cinematográficos.
- Escolher quais filmes e onde assisti-los: tarefa difícil.

Motivação e Contexto

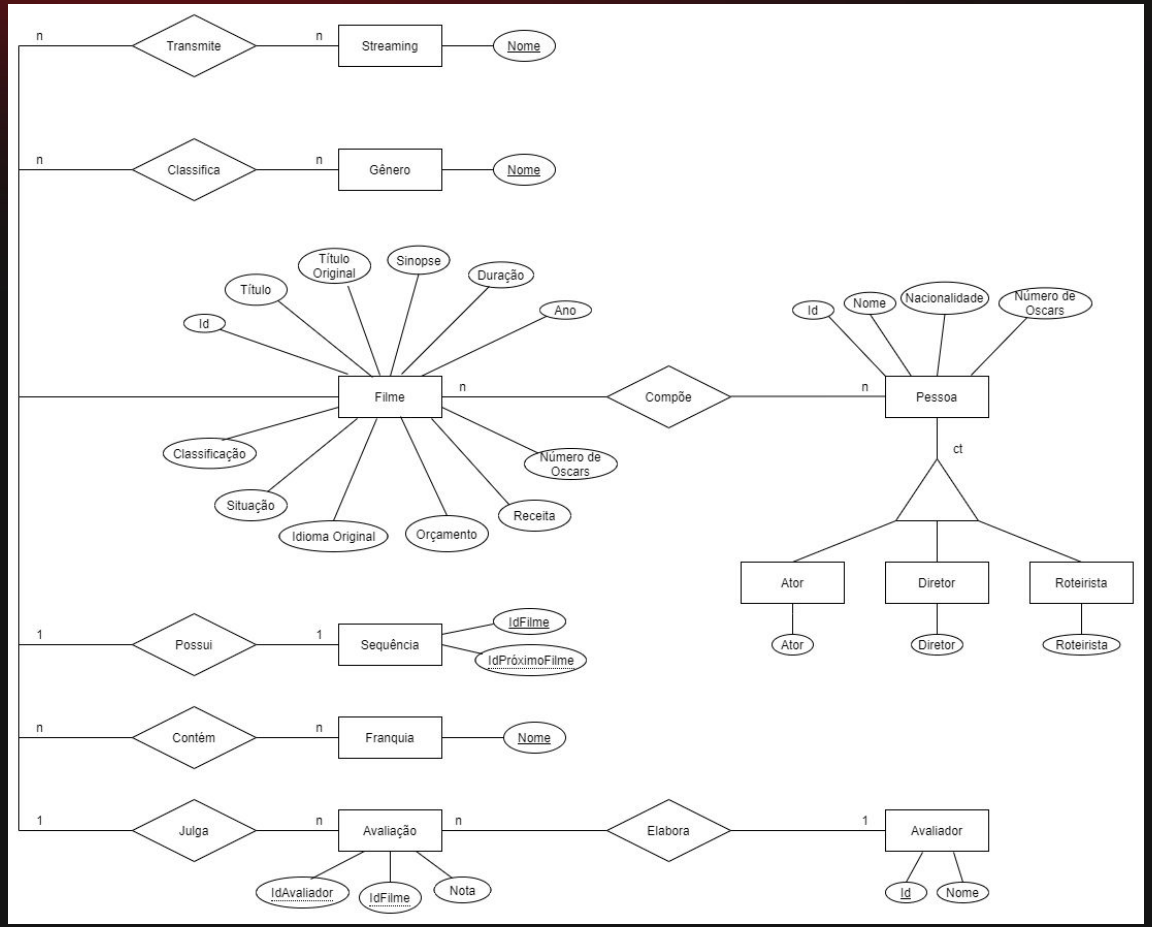
- Segundo o IMDB, a média de filmes produzidos por ano é de 2577.
- Empresas da indústria cinematográfica estão explorando maneiras de aumentar seu faturamento bruto de bilheteria.
- É difícil saber o que o público gosta antes de ouvir suas críticas.
- Produção de filmes: tarefa de risco.

Descrição

O Movie Catalog Dataset objetiva-se a ser uma base de dados sobre a indústria cinematográfica, permitindo a construção de mecanismos de busca e análise a respeito de diversos aspectos relacionados ao cinema.

Sendo que alguns deles são: gêneros, pessoas que participaram de filmes (diretores, roteiristas e atores) e os filmes por si só.

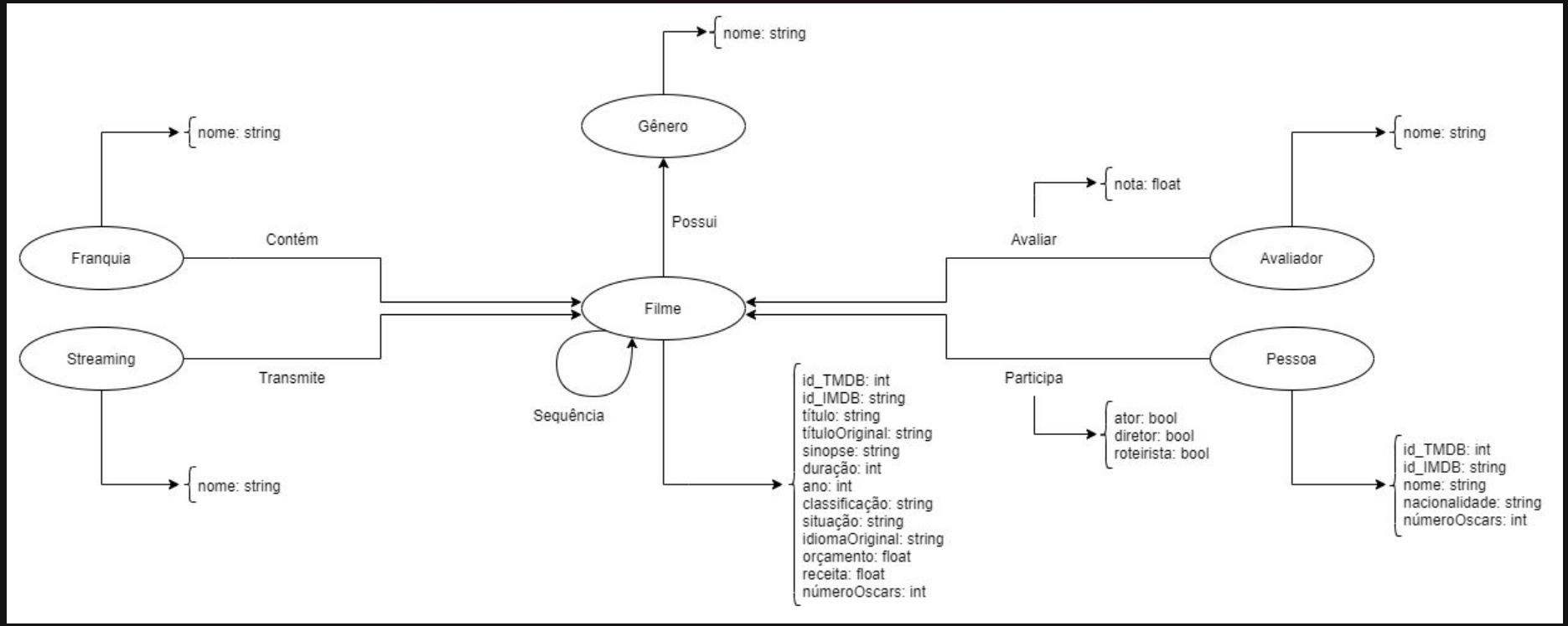
Modelo Conceitual



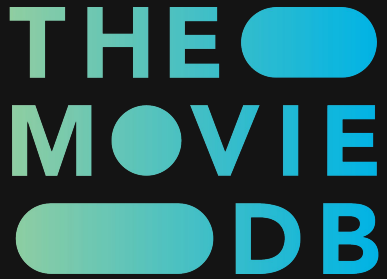
Modelo Lógico Relacional

```
FILME(_id_TMDB_, id_IMDB, titulo, titulo_original, sinopse, duracao, ano, classificacao, situacao,
idioma_original, orcamento, receita, num_oscars)
AVALIADOR(_id_, nome)
AVALIACAO(_id_avaliador_, _id_filme_TMDB_, nota)
  id_avaliador chave estrangeira -> AVALIADOR(id)
  id_filme_TMDB chave estrangeira -> FILME(id_TMDB)
FRANQUIA(_nome_)
FRANQUIAFILME(_nome_franquia_, _id_filme_TMDB_)
  nome_franquia chave estrangeira -> FRANQUIA(nome)
  id_filme_TMDB chave estrangeira -> FILME(id_TMDB)
GENERO(nome)
GENEROFILME(_nome_genero_, _id_filme_TMDB_)
  nome_genero chave estrangeira -> GENERO(nome)
  id_filme_TMDB chave estrangeira -> FILME(id_TMDB)
PESSOA(_id_TMDB_, id_IMDB, nome, nacionalidade, num_oscars)
PESSOAFILME(_id_pessoa_TMDB_, _id_filme_TMDB_, ator, diretor, roteirista)
  id_pessoa_TMDB chave estrangeira -> PESSOA(id_TMDB)
  id_filme_TMDB chave estrangeira -> FILME(id_TMDB)
SEQUENCIA(_id_filme_TMDB_, _id_filme_sequencia_TMDB_)
  id_filme_TMDB chave estrangeira -> FILME(id_TMDB)
  id_filme_sequencia_TMDB chave estrangeira -> FILME(id_TMDB)
STREAMING(_nome_)
STREAMINGFILME(_nome_streaming_, _id_filme_TMDB)
  nome_streaming chave estrangeira -> STREAMING(nome)
  id_filme_TMDB chave estrangeira -> FILME(id_TMDB)
```

Modelo Lógico de Grafos



Fontes de Dados



Ferramentas Utilizadas

Operações realizadas para a construção do Dataset

Operações aplicadas:

- Agregação de dados obtidos a partir da API do TMDB;
- Extração de dados páginas do IMDb via IMDbPY;
- Transformação de dados vindos do IMDb (cálculo do nº de Oscars);
- Exclusão de registros com dados essenciais faltantes;
- Integração de dados entre as diferentes fontes utilizadas;
- Paralelização das operações de extração de dados do IMDb.

Perguntas de Pesquisa/Análise com Resposta Implementada

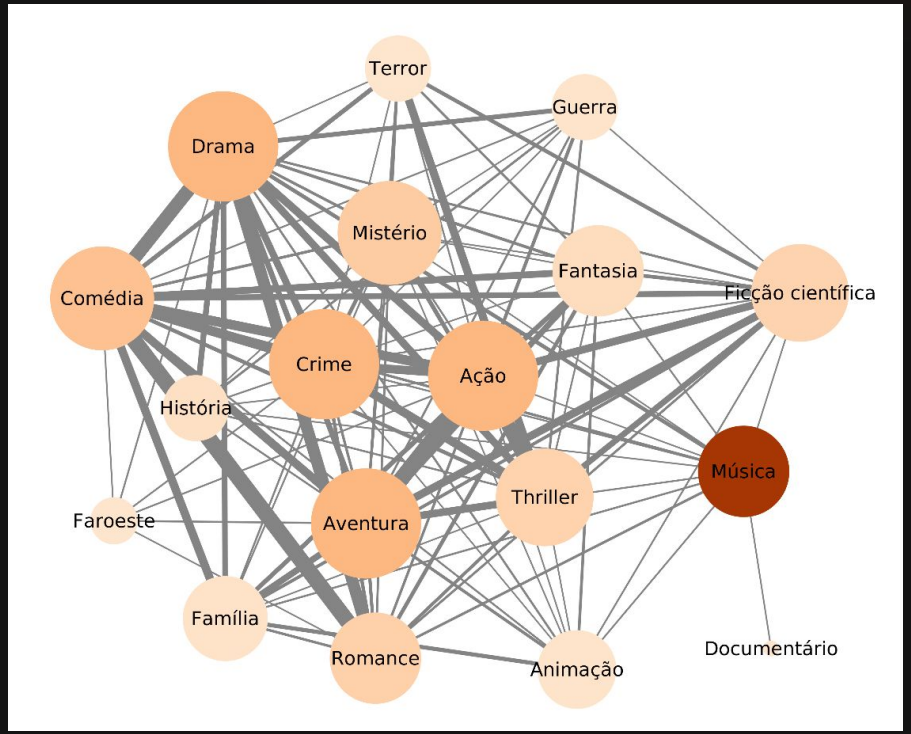
- Os filmes que mais fizeram sucesso com o público também são aqueles que mais fizeram sucesso com a crítica?
- Como os gêneros que classificam os filmes se relacionam em uma determinada década?
- Quais são as comunidades de pessoas que podem ser mapeadas? E quais são as pessoas mais relevantes dentre elas?

“Os filmes que mais fizeram sucesso com público também são aqueles que mais fizeram sucesso com a crítica?”

index	TITULO	RECEITA	NOTA_MEDIA
0	Avatar	2847246203	7.833333333333333
1	Vingadores: Ultimato	2797800564	8.700000000000001
2	Titanic	2187463944	8.200000000000001
3	Star Wars: O Despertar da Força	2068223624	8.1
4	Vingadores: Guerra Infinita	2046239637	8.4
5	Jurassic World: O Mundo dos Dinossauros	1671713208	6.899999999999995
6	O Rei Leão	1667635327	6.399999999999995
7	Os Vingadores: The Avengers	1518815515	8.299999999999999
8	Velozes & Furiosos 7	1515047671	7.533333333333332
9	Frozen II	1450026933	7.266666666666667

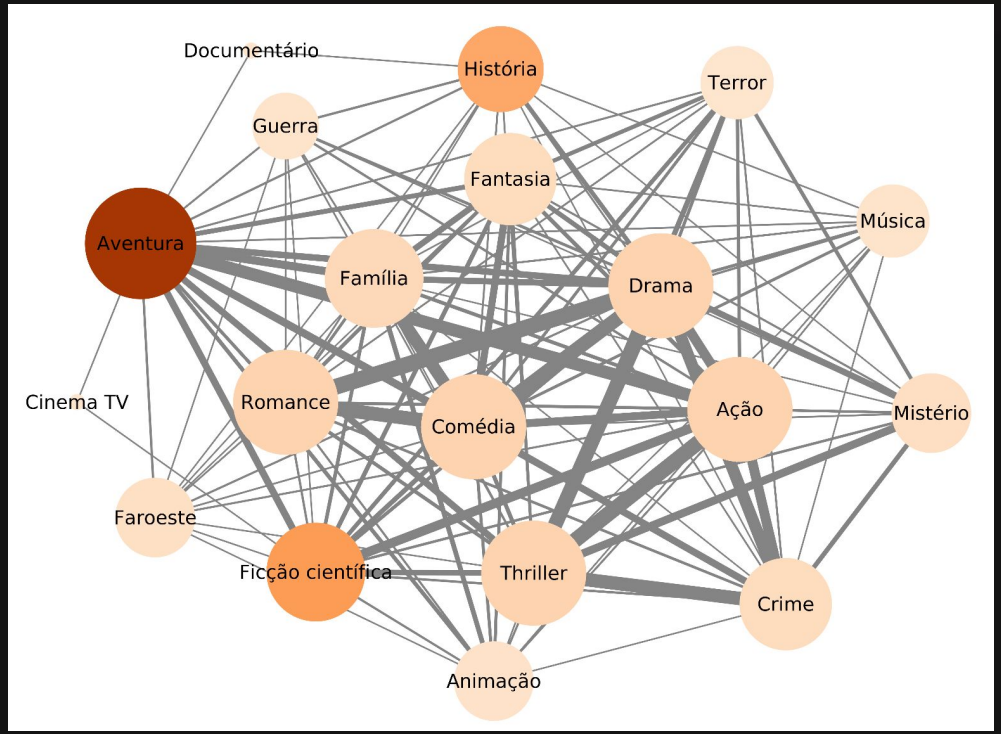
index	TITULO	RECEITA	NOTA_MEDIA
0	O Poderoso Chefão	245066411	9.233333333333333
1	A Lista de Schindler	321365567	9.066666666666666
2	Um Sonho de Liberdade	28341469	9.033333333333333
3	Batman: O Cavaleiro das Trevas	1004558444	8.966666666666667
4	Os Bons Companheiros	46835000	8.933333333333332
5	O Senhor dos Anéis: O Retorno do Rei	1118888979	8.9
6	Pulp Fiction: Tempo de Violência	214179088	8.866666666666665
7	Homem-Aranha: No Aranhaverso	375540831	8.833333333333334
8	O Senhor dos Anéis: As Duas Torres	926287400	8.833333333333334
9	O Silêncio dos Inocentes	272742922	8.833333333333334

“Como os gêneros que classificam os filmes se relacionam em uma determinada década?”



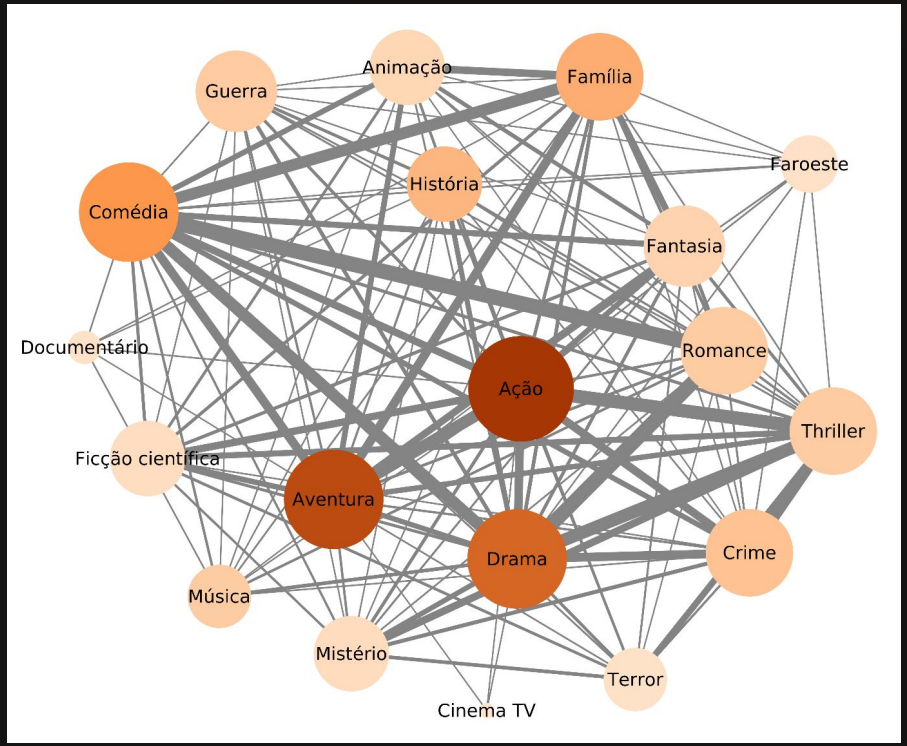
1980

“Como os gêneros que classificam os filmes se relacionam em uma determinada década?”



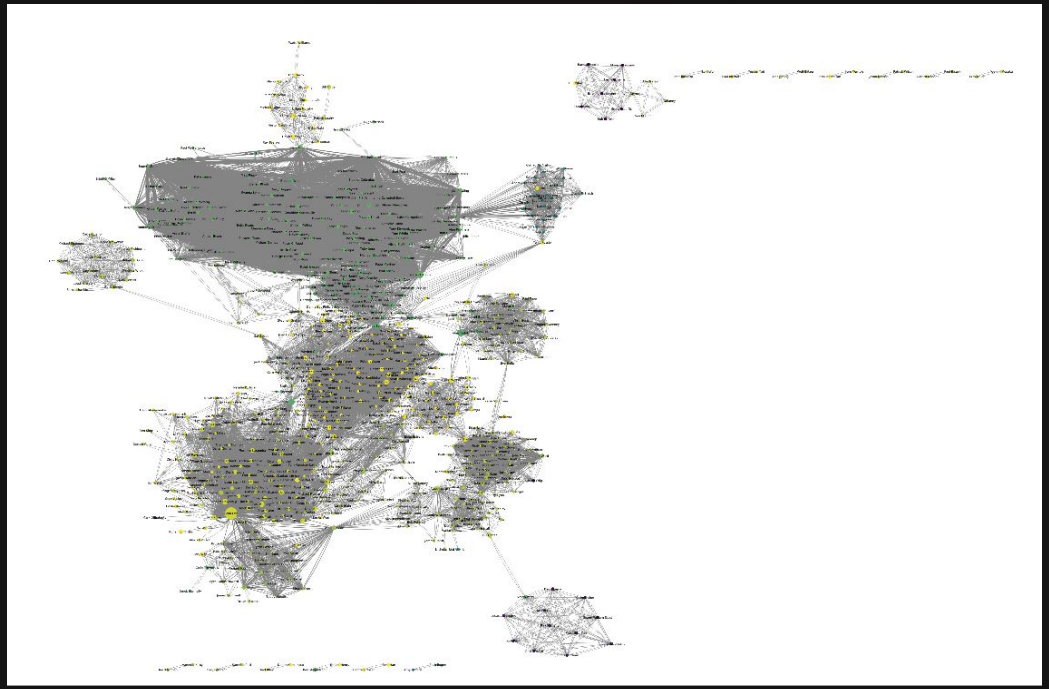
1990

“Como os gêneros que classificam os filmes se relacionam em uma determinada década?”

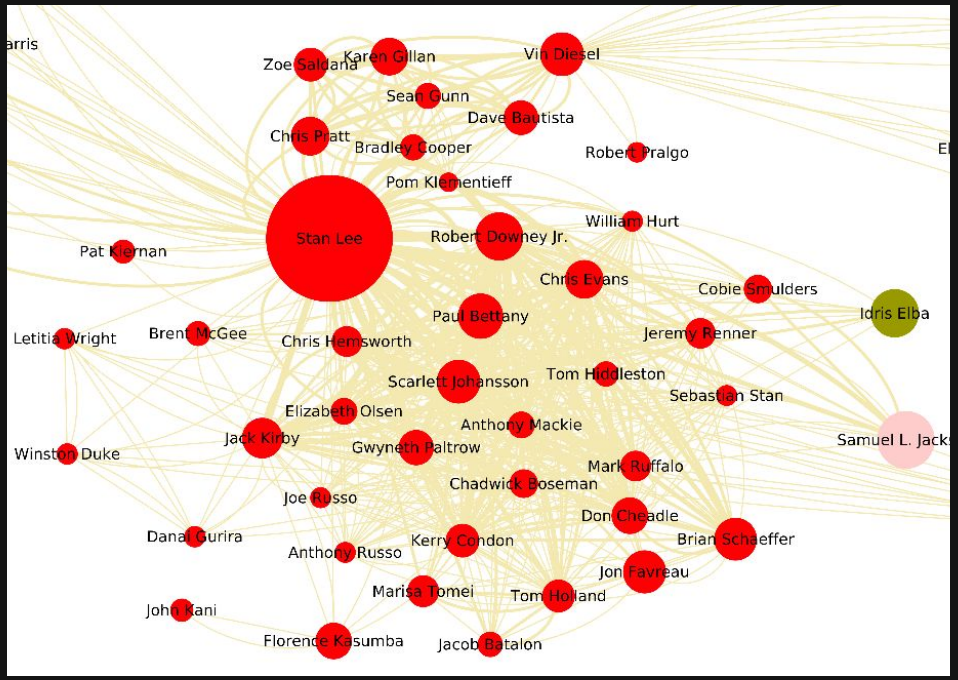


2000

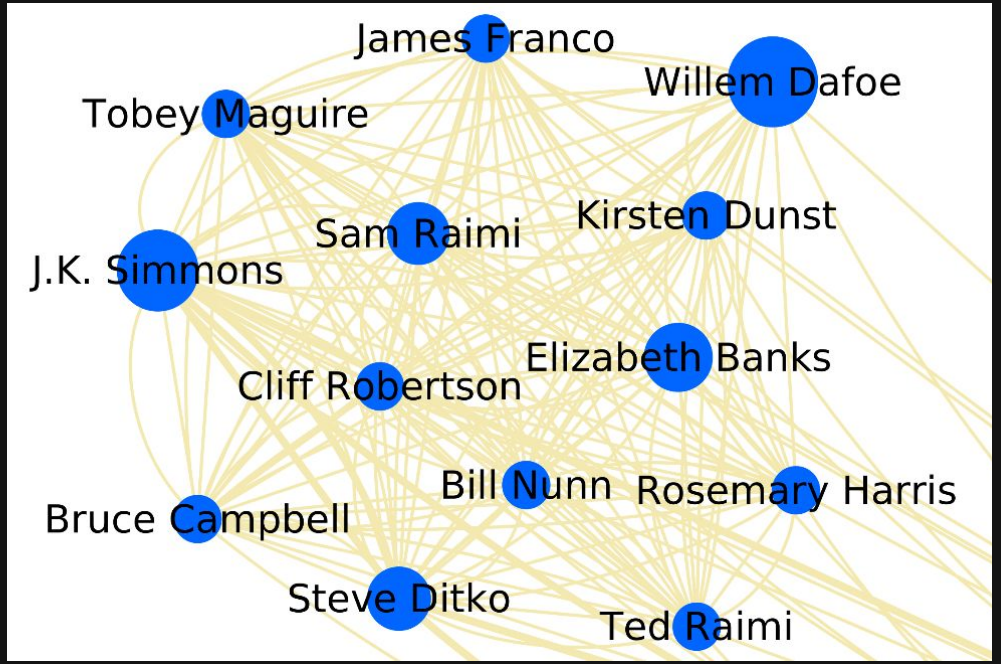
“Quais são as comunidades de pessoas que podem ser mapeadas? E quais são as pessoas mais relevantes dentre elas?”



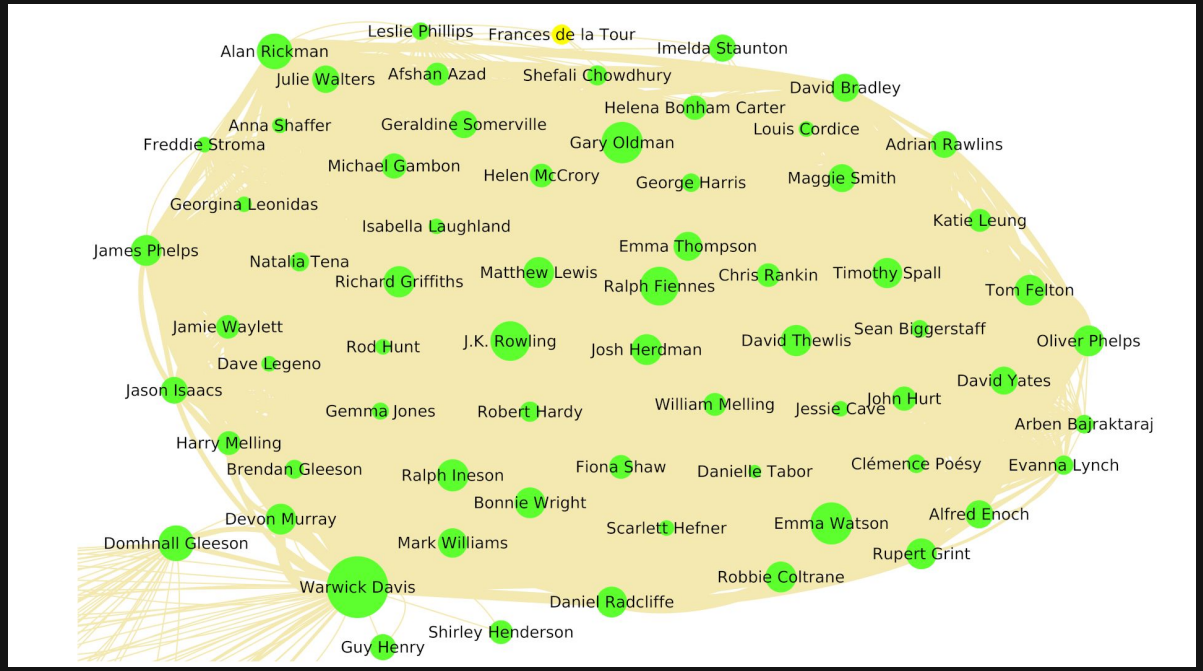
“Quais são as comunidades de pessoas que podem ser mapeadas? E quais são as pessoas mais relevantes dentre elas?”



“Quais são as comunidades de pessoas que podem ser mapeadas? E quais são as pessoas mais relevantes dentre elas?”



“Quais são as comunidades de pessoas que podem ser mapeadas? E quais são as pessoas mais relevantes dentre elas?”



Perguntas de Pesquisa/Análise sem Resposta Implementada

- Sabendo que uma pessoa X trabalhou com uma pessoa Y no filme A e com uma pessoa Z no filme B, qual é a probabilidade das pessoas Y e Z trabalharem juntas em um filme C?
- Quais são as características de um filme que faz sucesso com o público?
- Quem são as pessoas mais relevantes em cada gênero em uma determinada década?

Destaques de Código

```python
def build_franchise_table():
    config_tmdb()
    films = pd.read_csv('../data/processed/Filme.csv')
    films_ids = list(films['id_TMDB'])

    # Número de Núcleos Lógicos do processador
    num_pp = cpu_count()

    # Quantidade de trabalho para cada Processo
    ipp = len(films_ids)//num_pp
    pr_array = list()

    # Dividir os indices dos filmes entre os Processos
    start = 0
    for i in range(num_pp-1):
        stop = ipp*(i+1)
        ids_clip = list(films_ids[start: stop])
        start = stop
        # Criar e iniciar o processo
        pr = Process(target=franchise_job, args=(ids_clip, i))
        pr_array.append(pr)
        pr.start()
        print(f"ID do processo p{i}: {pr.pid}")
```

```python
# Iniciar o último processo (Pode possuir alguns ids a mais)
ids_clip = list(films_ids[stop: ])
pr =Process(target=franchise_job, args=(ids_clip, num_pp-1))
pr_array.append(pr)
pr.start()

# Esperar que todos os Processos terminem
for i in range(len(pr_array)):
    pr_array[i].join()

# Unir a tabela de franquias
concatenar(('FranquiaFilme', num_pp))
```

Destaques de Código

```
def load_csv(file_name="", index_col="id_TMDB"):
    ''' Código responsável por carregar tabelas ja geradas para reutilização.
        Lê uma tabela gerada previamente e retorna um dataframe para manipulação '''
    try:
        file_name += '.csv'
        return pd.read_csv(load_path + file_name, index_col=index_col)

    except FileNotFoundError:
        print("Arquivo [" + file_name + "] nao encontrado!")
```

Destaques de Código

```sql
/* Relação entre sucesso com o público (receita)
   e sucesso com a crítica (nota média) dos filmes */

DROP TABLE IF EXISTS FilmeReceitaNota;
DROP TABLE IF EXISTS FilmeAvaliacao;

CREATE VIEW FilmeAvaliacao AS
    SELECT A.id_filme, SUM(A.nota) nota_total, COUNT(A.id_filme) qtd_avaliacoes
        FROM Avaliacao A
        GROUP BY A.id_filme;

CREATE VIEW FilmeReceitaNota AS
    SELECT A.id_filme, F.titulo, F.ano, F.receita, (A.nota_total / A.qtd_avaliacoes) nota_media
        FROM Filme F, FilmeAvaliacao A
        WHERE A.id_filme = F.id_TMDB
          AND qtd_avaliacoes > 2;

-- Ordenação decrescente por receita
SELECT titulo, receita, nota_media
    FROM FilmeReceitaNota
    ORDER BY receita DESC LIMIT 10;

-- Ordenação decrescente por nota média
SELECT titulo, receita, nota_media
    FROM FilmeReceitaNota
    ORDER BY nota_media DESC LIMIT 10;
```

Evolução do Projeto

- Mudança de foco;
- Decisão de utilizar APIs;
- Expansão da quantidade de dados do dataset;
- Necessidade de otimizar os scripts de coleta de dados;
- Adição de dados do Rotten Tomatoes.

Obrigado!

Alguma pergunta?

https://github.com/MovieCatalogDS/MovieCatalogDS

CREDITS: This presentation template was created by Slidesgo, including icons by Flaticon and infographics & images by Freepik

Please keep this slide for attribution