



Universidade Presbiteriana Mackenzie

UNIVERSIDADE PRESBITERIANA MACKENZIE

TECNÓLOGO EM CIÊNCIAS DE DADOS

PARTICIPANTES DO GRUPO

Adrieli Machado Zaluski - 22503668

Caroline Ribeiro Ferreira - 22514635

Lais César Fonseca - 22500790

Liliane Gonçalves de Brito Ferraz - 22501142

Múcio Emanuel Feitosa Ferraz Filho - 22515925

Otávio Bernardo Scandiuzzi - 22511921

COLLAM FILMS

SÃO PAULO

2023



Sumário

| | | |
|-----|-------------------------------------|----|
| 1. | GLOSSÁRIO..... | 3 |
| 2. | INTRODUÇÃO | 5 |
| 3. | OBJETIVOS E METAS..... | 6 |
| 4. | APRESENTAÇÃO DA EMPRESA | 6 |
| 5. | METODOLOGIA | 8 |
| 6. | BIBLIOTECAS PYTHON | 10 |
| 7. | METADADOS | 16 |
| 8. | SISTEMA DE RECOMENDAÇÃO | 18 |
| 9. | ANÁLISE EXPLORATÓRIA DE DADOS | 22 |
| 10. | TRATAMENTO DE DADOS | 29 |
| 11. | MÉTODO DE RECOMENDAÇÃO..... | 31 |
| 12. | CONCLUSÃO | 35 |
| 13. | DIRETÓRIO GITHUB..... | 36 |
| 14. | REFERÊNCIAS BIBLIOGRÁFICAS | 37 |
| 15. | ANEXOS | 38 |



1. GLOSSÁRIO

- **Clustering (Agrupamento):** É uma técnica estatística usada para classificar elementos em grupos, de forma que elementos dentro de um mesmo cluster sejam muito parecidos, e os elementos em diferentes clusters sejam distintos entre si.
- **Colaboratory:** Conhecido também como “Colab”, é um produto do Google Research, área de pesquisas científicas do Google. O Colab permite que qualquer pessoa escreva e execute código Python arbitrário pelo navegador e é especialmente adequado para aprendizado de máquina, análise de dados e educação.
- **DataFrame:** É uma estrutura de dados bidimensional com os dados alinhados de forma tabular em linhas e colunas.
- **Datasets:** conjuntos de dados organizados em um formato similar ao das tabelas, com linhas e colunas que contém informações sobre determinado tema.
- **GitHub:** GitHub é uma plataforma de hospedagem de código-fonte e arquivos com controle de versão usando o Git. Ele permite que programadores, utilitários ou qualquer usuário cadastrado na plataforma contribuam em projetos privados e/ou Open Source de qualquer lugar do mundo.
- **IMDb:** Internet Movie Database, é uma base de dados online de informação sobre cinema, TV, música e games.
- **Kaggle:** É uma plataforma para aprendizado de ciência de dados. É também uma comunidade, a maior da internet, para assuntos relacionados com Data Science.
- **Machine Learning (Aprendizado de máquina):** É um subcampo da Engenharia e da ciência da computação que evoluiu do estudo de reconhecimento de padrões e da teoria do aprendizado computacional em inteligência artificial.
- **Python:** É uma linguagem de programação de alto nível, interpretada de script, imperativa, orientada a objetos, funcional, de tipagem dinâmica e forte. Foi lançada por Guido van Rossum em 1991.
- **Software:** É uma sequência de instruções escritas para serem interpretadas por um computador para executar tarefas específicas. Também pode ser definido como os



programas, dados e instruções que comandam o funcionamento de um computador, smartphone, tablet e outros dispositivos eletrônicos.

- **Streaming:** Fluxo contínuo, fluxo de média, fluxo de mídia ou transmissão contínua, é uma forma de distribuição digital, em oposição à descarga de dados.
- **Storytelling:** é a habilidade de contar histórias utilizando enredo elaborado, narrativa envolvente, e recursos audiovisuais. A técnica, cujo caráter é persuasivo, ajuda a promover o seu negócio e a vender seus serviços de forma indireta. Pode ser aplicada na produção de conteúdo, em vendas e em consultorias.



2. INTRODUÇÃO

Neste projeto, propomos o desenvolvimento de um sistema de recomendação de filmes utilizando técnicas de aprendizado de máquina e análise estatística preditiva. O objetivo principal é melhorar a experiência do usuário ao fornecer recomendações personalizadas individual com base em seu histórico de filmes assistidos, considerando fatores como gênero, ator, elenco e roteirista.

Utilizaremos um dataset público, que possuem acesso livre no site Kaggle e IMDb para criar um modelo eficaz de recomendação de filmes. Os resultados esperados incluem a correção de um sistema de recomendação funcional e a melhoria da precisão das recomendações à medida que mais dados são coletados e o modelo é refinado.

A recomendação de filmes desempenha um papel crucial na satisfação do público em serviços de *streaming* e na indústria cinematográfica como um todo. À medida que a quantidade de conteúdo disponível cresce, os usuários enfrentam o desafio de encontrar filmes que se alinhem com seus gostos e preferências individuais. Neste contexto, este projeto visa desenvolver um sistema de recomendação de filmes que aborde esse problema, e traga ao usuário uma experiência mais ágil e prazerosa na hora de escolher seus próximos filmes.

A abordagem do projeto combina os princípios da análise estatística preditiva, aprendizado de máquina, aquisição e preparação de dados, introdução à engenharia de *software* e tópicos de banco de dados. Utilizaremos conjuntos de dados obtidos no site da Kaggle e do IMDb para nosso sistema alimentar, permitindo que ele faça recomendações com base em atributos como: gênero e notas dadas por outros usuários.

3. OBJETIVOS E METAS

Nosso objetivo com a proposta de desenvolvimento da ferramenta “Collam Films”, é proporcionar ao usuário a facilidade de escolher um novo filme para assistir, oferecendo sugestões personalizadas com base no seu histórico de filmes já assistidos.

Esta ferramenta visa melhorar a experiência do usuário com relação aos próximos filmes a serem assistidos e reduzir o tempo na escolha de um novo filme de acordo com suas preferências.

E para realizar este sistema de recomendação de filmes iremos implementar métodos como:

Filtragem Colaborativa: método que analisa as preferências do usuário e as comparam com as de outros usuários semelhantes.

Filtragem Baseada em Conteúdo: utilizaremos informações como gênero, para recomendar filmes que compartilham de similaridades.

Esperamos que o sistema seja capaz de melhorar as recomendações de filmes à medida que mais dados sejam coletados e o modelo seja ajustado. Além disso, buscamos tornar o sistema escalável e eficiente para lidar com grandes volumes de dados de filmes, garantindo que os usuários sejam informados de filmes que lhes agradem.

4. APRESENTAÇÃO DA EMPRESA

A origem do nome da empresa “COLLAM FILMS”, nasceu da paixão por filmes e séries e da necessidade de tornar a experiência de assistir filmes ainda mais cativantes. Seu nome é uma fusão das iniciais dos integrantes do grupo que deram vida a essa iniciativa, representando nossa colaboração e dedicação.

O nome “Collam” é uma celebração da união e a diversidade de habilidades que traremos a este projeto.

Logo: Representado por um ícone central que ilustra uma fita de filme estilizada, esta fita se desenrola de forma dinâmica, sugerindo movimento e aventura. A fita também forma uma curva que se assemelha a um sorriso, indicando a alegria e o prazer de encontrar filmes interessantes.

A paleta de cores, foi definida por cores vibrantes e atraentes com tons de azul e amarelo dourado. Azul, representa confiança e confiabilidade e o amarelo evoca alegria e otimismo. No entanto, esta combinação de cores cria um equilíbrio entre seriedade e diversão.



Missão: É simplificar e aprimorar a forma como as pessoas descobrem e desfrutam de filmes e séries. Através da aplicação de métodos de aprendizado de máquina e da linguagem Python, buscamos oferecer recomendações personalizadas que encantem os usuários, conectando-os a conteúdos que realmente gostam e cada vez mais ágil.

Visão: Queremos ser reconhecidos no campo da recomendação de entretenimento audiovisual. Para isso, iremos aplicar o conhecimento adquirido até o momento e o que será adquirido no decorrer deste semestre, para criar um ecossistema onde cada pessoa encontre, de forma fácil e eficiente as histórias que a apaixonam. Buscaremos a inovação e a excelência técnica para proporcionar uma experiência de entretenimento de ponta.

Valores: Paixão pelo Entretenimento, Ética, Inovação, Eficiência, Integridade, Colaboração e Diversidade, Foco no Usuário.

5. METODOLOGIA

A metodologia deste projeto envolve várias etapas interconectadas. Começaremos coletando e disponibilizando os dados de filmes, garantindo que estejam prontos para análise. Em seguida, utilizaremos algoritmos de aprendizado de máquina para criar um modelo de recomendação de filmes. Inicialmente, exploraremos a filtragem colaborativa como abordagem principal.

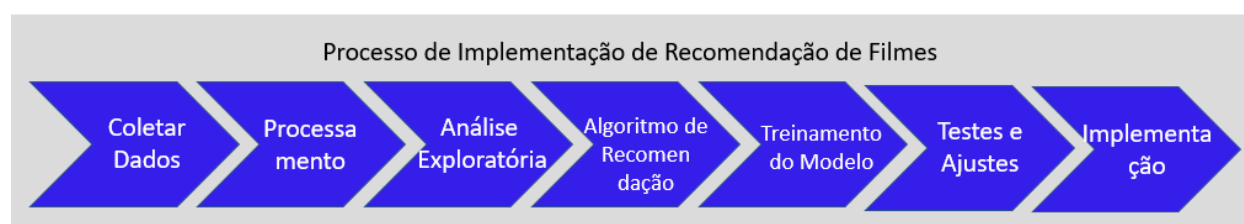
A aquisição e o armazenamento de dados serão realizados em um banco de dados adequado. Quanto à engenharia de software, desenvolveremos uma interface de usuário amigável para que os usuários finais possam interagir com o sistema de recomendação.

Será realizado um estudo das bibliotecas (pacotes) *python* que se faz necessário para a implementação deste projeto de forma mais assertiva e eficiente.

Utilizaremos de métodos de análise exploratória dos dados para realizar o entendimento de todo o conteúdo dos dados presente no dataset, será aplicado o tratamento necessário para a melhor adequação ao modelo proposto e assim obter a medidas de acurácia mais precisas.

O processo para a implementação deste sistema de recomendação de filmes utilizando Machine Learning, requer seguir as seguintes etapas demonstrada na figura 1.

Figura 1: Processo de Recomendação

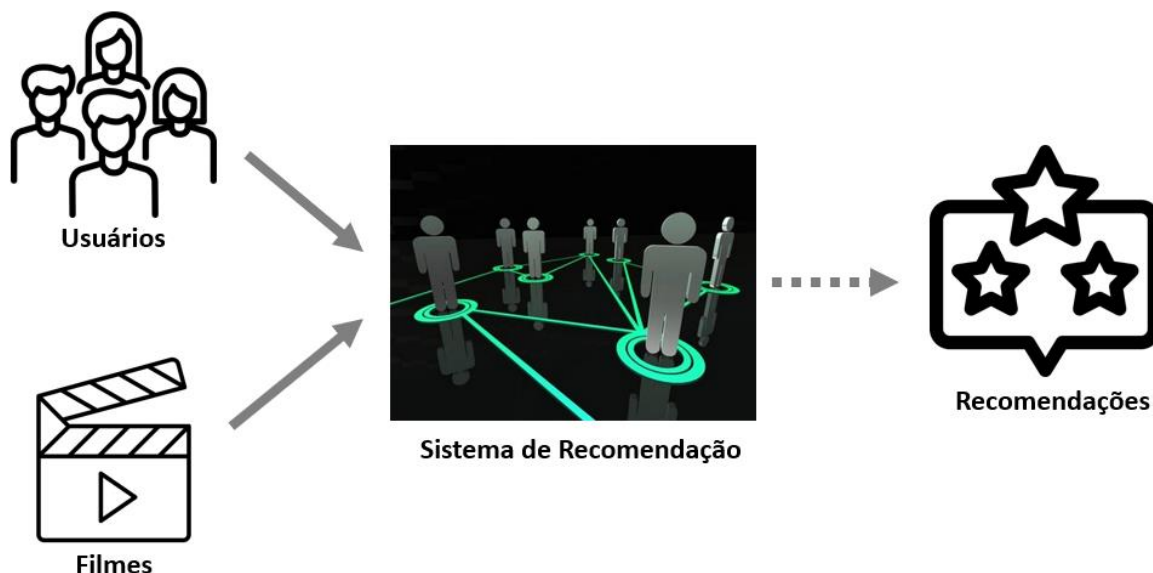


Elaborado pelo autor, 2023.

Para o desenvolvimento do sistema de recomendação de filmes abordaremos o método de “recomendação” através de algoritmos que oferecem sugestões personalizadas com base nas preferências do usuário. Esses sistemas, utilizados em diversos setores, incluindo comércio eletrônico e streaming, melhoram a experiência do usuário, aumentam o engajamento e auxiliam na descoberta de novos produtos.

Este sistema de recomendação de filmes pode ser ilustrado de forma bem simples como apresentado na figura 2 onde podemos observar como funciona a lógica deste sistema.

Figura 2: Modelo de Sistema de Recomendação



Elaborado pelo autor, 2023.

Será importante as etapas de pré-processamento dos dados que envolve a limpeza, engenharia de recursos e transformação. O treinamento do modelo inclui a escolha e implementação de métodos de recomendação, como filtragem colaborativa. Essa técnica, baseada no comportamento histórico dos usuários, a filtragem colaborativa baseada em item é eficiente em termos computacionais e robusta a mudanças no comportamento dos usuários.

A filtragem colaborativa baseada no usuário, destaca-se o cálculo de similaridades entre usuários, geralmente realizado através de métricas como a distância euclidiana. Essa abordagem identifica usuários com comportamentos semelhantes e recomenda itens com base nas preferências dos usuários mais similares.

Para a filtragem colaborativa baseada no item, o foco está no cálculo de similaridades entre gêneros de filmes. Utilizando a distância euclidiana, o sistema identifica a proximidade entre os gêneros dos filmes, recomendando itens semelhantes ao que o usuário já avaliou positivamente. Essa estratégia busca oferecer as recomendações mais alinhadas aos gostos do usuário, aprimorando a precisão do sistema e proporcionando uma experiência mais personalizada.

6. BIBLIOTECAS PYTHON

Para a implementação do sistema de recomendação, após estudos e análises realizadas sobre o tema, avaliamos que as bibliotecas da linguagem *Python* que serão utilizadas neste projeto para coletar os dados, processamento e tratamento dos dados, modelagem e avaliação. Selecionamos as bibliotecas a seguir:

1. Pandas:

A biblioteca Pandas é uma poderosa ferramenta de linguagem de programação *Python*, de código aberto e gratuito, que desempenha um papel fundamental na análise, limpeza e manipulação de dados. Além disso, ela permite a criação de gráficos e a manipulação de tabelas, tornando-a uma escolha essencial para programadores e cientistas de dados. *Python* é amplamente utilizado em diversas áreas, incluindo aprendizado de máquina, cibersegurança, mineração de dados, ciência de dados, programação web e muitas outras. A biblioteca Pandas é uma das razões pelas quais *Python* é tão popular para lidar com grandes estruturas de dados.

A biblioteca Pandas oferece uma ampla gama de recursos e funcionalidades para programadores e analistas de dados e suas funcionalidades são:

- **Manipulação de Dados:** O Pandas permite importar, manipular e processar dados de diversas fontes, como arquivos CSV, TSV ou bancos de dados SQL. Ele transforma esses dados em objetos *Python* chamados DataFrames, que se assemelham a tabelas, facilitando a análise e a manipulação.
- **Análise de Dados:** Com o Pandas, é possível realizar análises elaboradas dos dados. Ele oferece funções para agregar, agrupar, filtrar e calcular estatísticas, tornando a análise de dados eficiente e poderosa.
- **Limpeza de Dados:** A biblioteca simplifica a tarefa de limpar dados, permitindo a detecção e remoção de valores ausentes, duplicados e inconsistentes. Isso resulta em conjuntos de dados mais confiáveis e prontos para análise.
- **Visualização de Dados:** O Pandas se integra à biblioteca Matplotlib, facilitando a criação de gráficos e visualizações de dados. Isso torna a comunicação dos insights obtidos a partir dos dados mais eficazes.
- **Manipulação de Séries Temporais:** O Pandas oferece suporte robusto para lidar com dados de séries temporais, permitindo análises avançadas de dados ao longo do tempo.
- **Combinação de DataFrames:** É possível combinar DataFrames horizontal ou verticalmente, o que é útil quando se lida com grandes conjuntos de dados fragmentados.

- **Trabalho com Dados Categóricos:** O Pandas facilita a categorização de dados, simplificando a criação de modelos de aprendizado de máquina e a visualização de dados categóricos.

A biblioteca Pandas oferece várias vantagens distintas para os programadores e analistas de dados, são estas as principais vantagens:

- **Produtividade Elevada:** O Pandas é altamente produtivo e eficiente, economizando tempo na análise e manipulação de dados.
- **Facilidade de Acesso:** A biblioteca é conhecida por sua facilidade de uso e acessibilidade, tornando-a adequada para iniciantes e especialistas.
- **Versatilidade:** O Pandas é extremamente versátil e pode ser aplicado em diversas áreas, desde análise de dados até aprendizado de máquina.
- **Comunidade Ativa:** Com uma comunidade de colaboradores ativos, o Pandas está sempre em constante desenvolvimento e melhoria.
- **Integração com Outras Bibliotecas:** A integração com bibliotecas como Matplotlib e NumPy amplia ainda mais suas capacidades.
- **Manipulação de Grandes Dados:** Mesmo em grandes conjuntos de dados, o Pandas mantém seu desempenho e eficiência, tornando-o uma escolha sólida para projetos de qualquer escala.

2. Numpy:

NumPy, abreviatura de "*Numeric Python*", é uma biblioteca poderosa da linguagem de programação Python que se destaca por suas estruturas de dados multidimensionais, conhecidas como arrays. Além disso, o NumPy oferece uma extensa coleção de rotinas e funções que facilitam o processamento de arrays,

O NumPy é extremamente reconhecido por fornecer um conjunto abrangente de recursos e operações que simplifica o desenvolvimento de cálculos numéricos. Esses cálculos desempenham um papel fundamental em diversas áreas, incluindo:

- **Modelos de Machine Learning:** Em algoritmos de *Machine Learning*, é comum realizar uma variedade de cálculos numéricos, como multiplicação de matrizes, transposição e adição. O NumPy oferece uma biblioteca eficiente para executar esses cálculos de maneira fácil e rápida. Os arrays do NumPy são frequentemente usados para armazenar dados de treinamento e intervalos de modelos de *Machine Learning*.



- **Processamento de Imagem e Computação Gráfica:** Para manipular imagens de forma eficiente, o NumPy fornece funções que simplificam tarefas como espelhamento e rotação de imagens, entre outras operações de processamento de imagem.
- **Tarefas Matemáticas:** O NumPy é uma ferramenta útil para executar diversas tarefas matemáticas, incluindo integração numérica, diferenciação, interpolação e extrapolação. Além disso, a biblioteca oferece funções internas para álgebra linear e geração de números aleatórios. O NumPy é frequentemente combinado com outras bibliotecas, como SciPy e Matplotlib, para realizar tarefas complexas de análise e visualização de dados. Ele também é considerado uma alternativa ao MATLAB para aplicações matemáticas.

3. Matplotlib:

A biblioteca Matplotlib é uma ferramenta poderosa na linguagem de programação *Python*, voltada para a plotagem de gráficos 2D. Ela foi lançada em 2003 e seu desenvolvimento foi liderado pelo neurologista americano John D. Hunter. A origem do Matplotlib está ligada à pesquisa de pós-doutorado de Hunter, onde ele visualiza dados de eletrocorticografia em pacientes com epilepsia.

O Matplotlib oferece uma ampla gama de funcionalidades e recursos para criar gráficos 2D de alta qualidade e suas funcionalidades são:

- **Visualização de Dados:** A principal função do Matplotlib é criar gráficos e visualizações de dados de maneira eficaz. Ela suporta uma variedade de tipos de gráficos, incluindo gráficos de dispersão, barras, linhas, histogramas, entre outros.
- **Personalização:** A biblioteca permite personalizar todos os aspectos dos gráficos, incluindo núcleos, tamanhos, fontes e estilos. Isso possibilita a criação de visualizações únicas e informativas.
- **Suporte a Diferentes Backends:** A Matplotlib oferece suporte a uma ampla variedade de backends e saídas. Isso significa que os gráficos criados podem ser salvos em diferentes formatos de arquivo e exibidos em várias plataformas, tornando-os altamente portáteis.
- **Integração com outras bibliotecas:** O Matplotlib é frequentemente usado em conjunto com outras bibliotecas de análise de dados, como Pandas e NumPy, facilitando a criação de visualizações a partir de dados processados por essas ferramentas.

Existem várias vantagens em escolher a Matplotlib para criar gráficos e visualizações de dados, sendo elas:



- **Facilidade de Uso:** O Matplotlib é conhecido por sua facilidade de uso, tornando-a acessível tanto para iniciantes quanto para profissionais experientes.
- **Ampla Comunidade:** A biblioteca possui uma comunidade ativa de desenvolvedores e usuários.

4. Seaborn:

O Seaborn é uma biblioteca de visualização de dados em *Python* que se baseia no popular Matplotlib. Ela foi projetada para criar gráficos estatísticos elegantes e informativos com facilidade, exigindo apenas algumas linhas de código. O Seaborn é particularmente útil ao lidar com dados complexos, fornecendo diversas ferramentas para simplificar o processo de visualização e apresentação de resultados. Suas principais funções são:

- **Gráficos de Barras:** Os gráficos de barras são ideais para visualizar dados categóricos. O Seaborn oferece diversos tipos de gráficos de barras, incluindo gráficos de barras agrupadas, empilhadas e horizontais. Esses gráficos ajudam a representar informações de forma clara e eficaz.
- **Gráficos de Dispersão:** Os gráficos de dispersão são usados para visualizar a relação entre duas variáveis. O Seaborn oferece vários tipos de gráficos de dispersão, como aqueles com linhas de regressão e gráficos de dispersão com hexágonos, que ajudam a identificar tendências e padrões nos dados.
- **Gráficos de Caixa:** Os gráficos de caixa são úteis para representar a distribuição de uma variável numérica. O Seaborn oferece diferentes tipos de gráficos de caixa, incluindo aqueles com distribuição e pontos, permitindo a análise da dispersão e dos valores atípicos nos dados.
- **Gráficos de Densidade:** Os gráficos de densidade ajudam a visualizar a distribuição de uma variável numérica. O Seaborn oferece gráficos de densidade uni variada e bivariada, fornecendo insights sobre a distribuição conjunta de duas variáveis numéricas.

Suas vantagens de utilização são:

- **Facilidade de Uso:** O Seaborn é reconhecido por sua facilidade de uso, permitindo que os usuários criem visualizações complexas com código conciso.
- **Estilo Elegante:** A biblioteca oferece uma ampla variedade de estilos e paletas de cores elegantes, tornando as visualizações atraentes e informativas.
- **Integração com o Matplotlib:** O Seaborn é baseado no Matplotlib, o que significa que você pode combinar as funcionalidades dessas duas bibliotecas, aproveitando o poder do Matplotlib com a simplicidade do Seaborn.



- **Visualização Estatística:** O foco do Seaborn está na visualização estatística, o que o torna uma escolha qualitativa para análises exploratórias de dados e apresentação de resultados em um formato informativo.
- **Flexibilidade:** Apesar de sua simplicidade, a Seaborn oferece opções avançadas de personalização para atender às necessidades específicas de visualização.

5. Surprise Lib:

Surprise é uma biblioteca *Python* voltada para a construção e análise de sistemas de recomendação que trabalham com dados de classificação explícita. Ela é uma parte do conjunto de bibliotecas scikit e foi desenvolvida para simplificar o processo de criação e avaliação de sistemas de recomendação. Suas principais características são:

- **Controle do Usuário:** O Surprise foi projetado para oferecer aos usuários um controle preciso sobre seus experimentos. A documentação é um aspecto crucial, com ênfase na clareza e na precisão, abordando detalhes dos algoritmos.
- **Facilidade de Uso de Dados:** Para tornar o conjunto de dados mais simples, os usuários podem aproveitar conjuntos de dados integrados, como Movielens e Jester, ou usar seus próprios conjuntos de dados personalizados.
- **Implementação de Novas Ideias:** O Surprise é flexível e facilita a implementação de novos algoritmos de recomendação, permitindo que os desenvolvedores experimentem novas ideias.
- **Avaliação e Análise de Desempenho:** Uma biblioteca fornece ferramentas para avaliar, analisar e comparar o desempenho dos algoritmos. Isso inclui procedimentos de validação cruzada e pesquisa exaustiva de parâmetros.

6. SKLearn ou Scikit-Learn:

O *Scikit-learn*, originalmente chamado de *scikits.learn*, é uma biblioteca de aprendizado de máquina de código aberto para *Python*. Oferece uma ampla variedade de algoritmos para classificação, regressão e agrupamento, como máquinas de vetores de suporte, florestas aleatórias, gradient boosting, k-means e DBSCAN. Essa biblioteca é projetada para integrar-se perfeitamente com as bibliotecas *Python* numéricas e científicas, como NumPy e SciPy. Além disso, o Scikit-learn é uma ferramenta gratuita e versátil para modelagem estatística, análise de dados e aprendizado supervisionado e não supervisionado, tornando-o uma escolha popular para machine learning em *Python*. Suas principais aplicações são:



- **Algoritmos de Classificação:** Identificam categorias associadas aos dados, úteis para tarefas como classificar e-mails como spam ou não.
- **Algoritmos de Regressão:** Criam modelos para compreender a relação entre dados de entrada e saída, usados, por exemplo, para prever o comportamento dos preços das ações.
- **Algoritmos de Agrupamento (*Clustering*):** Agrupam automaticamente dados com características semelhantes, como segmentar clientes por idade ou localização.
- **Redução de Dimensionalidade:** Diminuem o número de variáveis para análise, aprimorando eficiência na visualização e processamento de dados.
- **Seleção de Modelo:** Oferecem ferramentas para comparar, validar e selecionar os melhores modelos e parâmetros para projetos de ciência de dados.
- **Pré-processamento:** Extrai e normaliza recursos nos dados, sendo útil para transformar dados de entrada, como texto, durante a análise.

Seus principais recursos são:

- **Pré-processamento:** Realiza transformações e manipulações nos dados brutos, incluindo tratamento de valores ausentes, conversão de valores categóricos em formatos numéricos e seleção de recursos.
- **Estimadores:** Oferece uma variedade de algoritmos predefinidos para aprendizado supervisionado e não supervisionado, como classificadores, regressões, SVM, árvores de decisão e algoritmos de *clustering*.
- **Avaliação do Modelo:** Fornece métricas estatísticas para avaliar o desempenho dos modelos, incluindo validação cruzada e funções de métricas individuais.
- **Otimização do Modelo:** Permite a otimização de hiper parâmetros, incluindo aprendizado conjunto, pesquisa em grade e pesquisa aleatória para melhorar o desempenho dos modelos de aprendizado de máquina

7. Scipy:

O projeto SciPy é uma coleção de bibliotecas Python open-source para matemática, ciência e engenharia, incluindo o NumPy e o matplotlib. Por outro lado, parte do projeto é a biblioteca scipy, que chamaremos apenas de SciPy. Ela contém como submódulos a maioria das ferramentas que se espera de um software para cientistas, incluindo funções especiais, integração, otimização, interpolação, transformadas de Fourier, processamento de sinais, álgebra linear, estatística e processamento de imagens. Suas principais características são:



- **Funcionalidades Avançadas:** O SciPy oferece um conjunto abrangente de funcionalidades avançadas para cálculos numéricos e científicos, incluindo otimização, álgebra linear, integração numérica, interpolação, processamento de sinais, estatísticas e muito mais.
- **Construído sobre o NumPy:** O SciPy é construído sobre o NumPy, outra biblioteca fundamental para computação numérica em *Python*. Isso significa que ele herda a capacidade de lidar com arrays multidimensionais eficientemente, o que é essencial para muitos tipos de cálculos científicos.
- **Integração com Outras Bibliotecas:** O SciPy é frequentemente usado em conjunto com outras bibliotecas científicas, como Matplotlib para visualização de dados e Pandas para análise de dados, tornando-o parte de um ecossistema poderoso para a ciência de dados e engenharia.
- **Licença de Código Aberto:** O SciPy é distribuído sob uma licença de código aberto (BSD), o que significa que é gratuito para uso e pode ser estendido e modificado conforme necessário.

7. METADADOS

Para o desenvolvimento do nosso projeto de recomendação de filmes, foi escolhido um dataset com as seguintes descrições:

7.1. Tipo de arquivo

A base de dados adquirida é de extensão csv. Denominada “movies_metadata.csv”.

7.2. Origem dos dados

Os dados são de domínio público/aberto, do site da Kaggle.

<https://www.kaggle.com/datasets/rounakbanik/the-movies-dataset/>

7.3. Sensibilidade / LGPD

Esta base de dados não possui dados sensíveis e está de acordo com a Lei Geral de Proteção de Dados Pessoais – LGPD.

7.4. Proprietário do dado

Rounak Banik

7.5. Descrição e atributos dos dados

Conjunto de Dados Full MovieLens - Metadados de Filmes é uma coleção de informações abrangentes sobre filmes lançados até julho de 2017. Ele inclui metadados detalhados sobre aproximadamente 45.000 filmes, oferecendo uma riqueza de informações relacionadas à indústria cinematográfica. Este dataset possui 45.466 linhas e 24 colunas.

O arquivo `movies_metadata.csv` é o principal metadados de filmes. Ele contém informações sobre os filmes incluídos no conjunto de dados. Alguns dos principais recursos disponíveis neste arquivo incluem:

- ✓ Pôsteres: Imagens associadas aos filmes.
- ✓ Cenários: Detalhes sobre o enredo ou resumo dos filmes.
- ✓ Orçamento: Informações sobre os custos de produção dos filmes.
- ✓ Receitas: Dados relacionados à receita gerada pelos filmes.
- ✓ Datas de Lançamento: Informações sobre as datas de lançamento dos filmes.
- ✓ Idiomas: Idiomas em que os filmes estão disponíveis.
- ✓ Países de Produção: Países onde os filmes foram produzidos.
- ✓ Empresas: Informações sobre as empresas de produção envolvidas na criação dos filmes.

O conjunto de dados Full MovieLens - Metadados de Filmes é uma fonte de informações valiosa para uma variedade de aplicações, incluindo sistemas de recomendação de filmes, análise de tendências da indústria cinematográfica e estudos de mercado. Os metadados detalhados permitem uma análise aprofundada dos filmes e podem ser utilizados para desenvolver algoritmos de recomendação mais precisos.

Os dados foram coletados e compilados pelo GroupLens a partir de fontes diversas, incluindo informações de elenco, equipe, palavras-chave de enredo, entre outros. Além disso, o conjunto de dados inclui avaliações de filmes fornecidos por aproximadamente 270.000 usuários em uma escala de classificação de 1 a 5.

É importante notar que, além dos metadados de filmes, este conjunto de dados inclui informações sobre as avaliações de usuários, que podem ser exploradas em projetos adicionais relacionados à recomendação e análise de avaliações de filmes.

A base de dados é composta com os atributos, seguidos de sua respectiva tradução, descrição sobre e o tipo do atributo que foi identificado através do comando “`dtypes`”.

- ✓ `Adult` – Adulto: tipo de filme (object);
- ✓ `belongs_to_collection` – Pertence à coleção, informação do filme e sua coleção (object);
- ✓ `budget` – Orçamento: quanto foi gasto na produção do filme (object);
- ✓ `genres` – Gêneros (object);
- ✓ `homepage` - A página do filme na internet (object);
- ✓ `id` – Um número identificador (object);



- ✓ imdb_id – Código identificador (object);
- ✓ original_language – Língua original (object);
- ✓ original_title – Título original (object);
- ✓ overview – Uma descrição básica do filme (object);
- ✓ popularity – Uma espécie de "nota" de popularidade calculada pelo próprio TMDb (object);
- ✓ poster_path – Caminho do poster (object);
- ✓ production_companies – As empresas envolvidas na produção (object);
- ✓ production_countries – Países em que o filme foi produzido (object);
- ✓ release_date – Data de lançamento (object);
- ✓ revenue – Faturamento (float64);
- ✓ runtime – Tempo de duração (em minutos) - (float64);
- ✓ spoken_language – As línguas faladas no filme (object);
- ✓ status – Se o filme foi lançado ou não (object);
- ✓ tagline – Uma rápida chamada do filme (como encontramos em propagandas) (object);
- ✓ title – Título (object);
- ✓ video – Vídeo (object);
- ✓ vote_average – Uma nota média do filme (float64);
- ✓ vote_count – Mostra o número de notas atribuídas ao filme (float64).

8. SISTEMA DE RECOMENDAÇÃO

Sistemas de recomendação são algoritmos e técnicas que oferecem sugestões personalizadas de itens ou informações para usuários com base em suas preferências, comportamentos passados ou características similares de outros usuários, como foi também demonstrado na figura 2 apresentado no tópico da metodologia deste projeto.

Um mecanismo de recomendação é uma classe de aprendizado de máquina que oferece sugestões relevantes ao cliente. Antes do sistema de recomendação, a grande tendência para comprar era aceitar sugestões de amigos. Mas agora o Google sabe quais notícias você vai ler, o Youtube sabe que tipo de vídeos você vai assistir com base em seu histórico de pesquisa, histórico de exibição ou histórico de compra.

Um sistema de recomendação ajuda uma organização a criar clientes fiéis e construir a confiança deles nos produtos e serviços desejados para os quais vieram em seu site. Os sistemas de recomendação de hoje são tão poderosos que também podem lidar com o novo cliente que visitou o site pela primeira vez. Eles recomendam os produtos que estão em alta ou com alta classificação e também podem recomendar os produtos que trazem o máximo de lucro para a empresa.



Um sistema de recomendação de filmes é um tipo de sistema de recomendação em que temos que recomendar filmes semelhantes ao usuário com base em seu interesse.

Esses sistemas são amplamente utilizados em diversos contextos, como comércio eletrônico, streaming de conteúdo, redes sociais, entre outros. Existem três tipos principais de sistemas de recomendação:

- **Recomendação baseada em conteúdo:** Este tipo de sistema analisa os atributos e características dos itens recomendados e os compara com as preferências do usuário. Por exemplo, em um sistema de recomendação de filmes, se um usuário assistiu e gostou de filmes de ação, o sistema pode recomendar outros filmes de ação com características semelhantes.
- **Filtragem colaborativa:** Essa abordagem identifica padrões entre usuários com gostos semelhantes. Se um usuário A tem preferências semelhantes a um usuário B em relação a determinados itens, o sistema de recomendação pode sugerir itens que o usuário B gostou, mas que o usuário A ainda não viu ou experimentou.
- **Recomendação híbrida:** Combinação de abordagens baseadas em conteúdo e colaborativas para melhorar a precisão e a robustez das recomendações. Isso ajuda a superar algumas limitações de cada abordagem isolada.

Os sistemas de recomendação são fundamentais para melhorar a experiência do usuário, aumentar o engajamento e auxiliar os usuários na descoberta de novos produtos, serviços ou conteúdos que possam ser de seu interesse. Eles são aplicados em uma variedade de setores, desde plataformas de streaming de vídeo e música até lojas online e redes sociais.

1. Pré-processamento dos Dados:

- Limpeza dos dados, como tratamento de valores ausentes e duplicados.
- Engenharia de recursos, se necessário, para criar novos recursos relevantes.
- Transformação de dados, como coincidência de variáveis categóricas.

2. Treinamento do Modelo:

- Escolher e implementar métodos de recomendação, como filtragem colaborativa ou modelos baseados em conteúdo.
- Treine e avalie o modelo usando detalhes detalhados.

3. Filtragem Colaborativa:

A filtragem colaborativa é uma técnica comumente utilizada em sistemas de recomendação para fazer observações ou recomendações com base no comportamento passado dos usuários ou em informações de itens (produtos, filmes, livros etc.) que foram avaliados pelos usuários. O princípio fundamental da filtragem colaborativa é que as preferências de um usuário possam ser prejudicadas com base nas opiniões de outros usuários semelhantes. Existem dois principais tipos de filtragem colaborativa:

Filtragem Colaborativa Baseada no Usuário

A filtragem colaborativa baseada no usuário, também conhecida como "user-based CF", utiliza a similaridade entre usuários para fazer recomendações. O conceito por trás desse método é que usuários que tiveram interações semelhantes no passado tiveram a ter gostos e tendências semelhantes no futuro. O processo geral envolve os seguintes passos:

- Calcule a semelhança entre o usuário alvo (para o que queremos fazer recomendações) e todos os outros usuários com base em suas avaliações passadas ou comportamento.
- Identifique os usuários mais semelhantes ao usuário alvo.
- Recomendamos itens que os usuários similares tenham gostado e que o usuário alvo ainda não tenha avaliado.

Essa abordagem é intuitiva e eficaz, mas pode ser computacionalmente cara, especialmente em conjuntos de grandes dados, devido ao design de semelhanças entre todos os pares de usuários.

Filtragem Colaborativa Baseada no Item

A filtragem colaborativa baseada no item, também chamada de "CF baseada em item", se concentra nas características dos itens e como eles se relacionam entre si. A ideia por trás desse método é que itens semelhantes tendem a ser apreciados pelos mesmos usuários. O processo geral envolve os seguintes passos:

- Calcule a similaridade entre todos os pares de itens com base nas avaliações dos usuários.
- Para um usuário específico, identifique os itens que ele já avaliou com certeza.
- Recomendando itens semelhantes aos que o usuário já avaliou com certeza.

A filtragem colaborativa baseada em nenhum item é eficiente em termos computacionais, uma vez que o design de similaridades é feito entre itens, não entre usuários. Além disso, é robusta a mudanças no comportamento dos usuários.

4. Métricas e Processos de Avaliação:

A métrica MAE (Mean Absolute Error) é uma medida comum usada para avaliar o desempenho de modelos de machine learning, especialmente em problemas de regressão. Ela quantifica o erro médio absoluto entre as previsões do modelo e os valores reais do conjunto de dados. Quanto menor o valor do MAE, melhor o desempenho do modelo, indicando que as previsões estão mais próximas dos valores reais.

E para calcular o MAE a fórmula é:

$$MAE = \frac{1}{n} \sum_{i=1}^n |y_i - p_i|$$

Onde:

n é o número de amostras

y é o valor observado para cada amostra

p é o valor previsto pelo modelo para cada amostra

| | representa o valor absoluto

O processo de avaliação que inclui a métrica MAE normalmente envolve a divisão dos dados em conjuntos de treinamento, teste e, às vezes, validação cruzada. Aqui está como o processo pode ser descrito:

➤ **Divisão dos Dados:**

Os dados disponíveis são divididos em pelo menos dois conjuntos: o conjunto de treinamento e o conjunto de teste. Em alguns casos, pode haver um terceiro conjunto chamado conjunto de validação.

➤ **Conjunto de Treinamento:**

O conjunto de treinamento é usado para treinar o modelo. Os algoritmos de machine learning aprendem com esses dados para fazer previsões.

➤ **Conjunto de Teste:**

O conjunto de teste é reservado para avaliar o desempenho do modelo. O modelo faz previsões com base nos dados de teste, e o MAE é calculado para medir o quão bem as previsões correspondem aos valores reais.

➤ **Validação Cruzada:**

A validação cruzada é uma técnica que divide os dados em várias dobras (*folds*) e realiza várias iterações de treinamento e teste. Isso é útil para avaliar a capacidade do modelo de generalizar para dados não vistos. O MAE é calculado para cada dobra e pode ser usado para determinar o desempenho médio do modelo.

9. ANÁLISE EXPLORATÓRIA DE DADOS

Para iniciar a etapa de análise exploratória de dados, importante entendermos mais sobre a importância desta etapa dentro da ciência de dados e para a realização de qualquer tarefa a ser desenvolvida utilizando dados como insumo para estudos e aplicações. Conhecer inicialmente os dados é imprescindível para seguir nas próximas etapas de forma mais eficaz.

A fase de entendimento dos dados começa com a coleta de dados e segue com atividades que permitem com que você se familiarize com os dados, identifique problemas de qualidade de dados, os primeiros insights sobre os dados e/ou detecte subconjuntos interessantes para formar hipóteses sobre informações ocultas.

É uma fase de modelos e hipóteses simples, mas que já fornece muitas respostas relevantes ao problema tratado. A Análise Exploratória de Dados (EDA, Exploratory Data Analysis) é usada para analisar e investigar conjuntos de dados e resumir suas características principais, que se destacam algumas delas na figura 3, para melhor compreensão deste processo.

Figura 3: Modelo de Sistema de Recomendação



Elaborado pelo autor, 2023.

A análise exploratória emprega métodos quantitativos, como estatísticas e sumarizações dos dados, e técnicas de visualização dos dados para obter uma série de análises. Em geral, elas envolvem:

- Análise uni variada dos dados, o que inclui a análise de cada atributo dos dados individualmente a partir de estatísticas dos dados, análise da distribuição dos valores, identificação de anomalias e padrões, entre outros.
- Análise bivariada e multivariada, o que avalia a relação entre cada variável no conjunto de dados com uma variável objetivo, ou entre várias variáveis.
- Análise de agrupamentos, o que inclui a redução da dimensão dos dados em grupos de dados que se assemelham e se diferenciam dos demais.

Após este entendimento, foi realizado a análise da base de dados aplicando os métodos estatísticos de análise exploratória de dados, através do uso da linguagem Python, utilizando a ferramenta Colaboratory. Existem diversas maneiras de analisarmos um dataframe do Pandas. Uma delas é a função `describe()`, que exibirá informações das colunas numéricas do conjunto de dados, conforme apresentado na figura 4 a forma de como é exibido os dados.

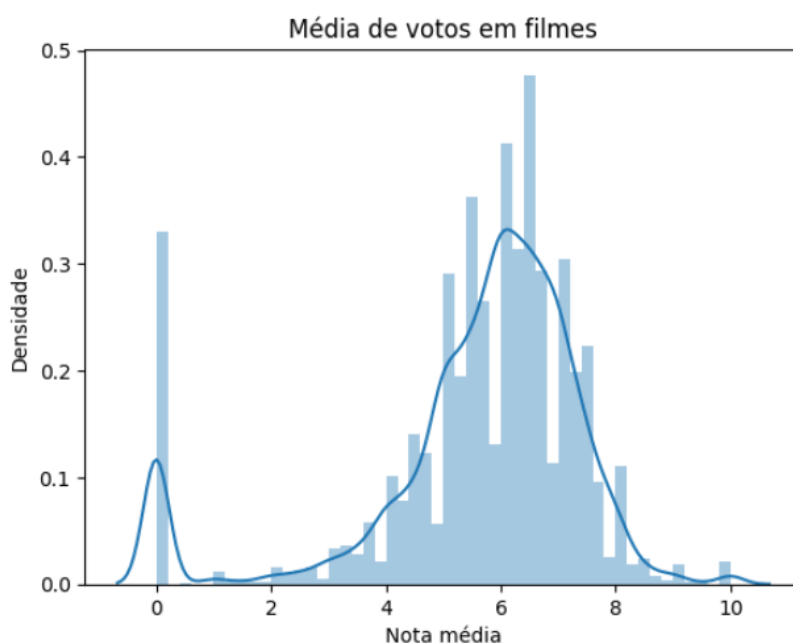
Figura 4: Discribe Dataset movies_metadata.csv

| | Contagem | Média | Desvio Padrão | Mínimo | 25% | 50% | 75% | Máximo |
|-------------------|----------|-------|---------------|--------|-----|-----|-----|--------|
| Receita | 45.460 | 1,12 | 6,43 | 0 | 0 | 0 | 0 | 2,79 |
| Tempo de execução | 45.203 | 9,41 | 3,84 | 0 | 85 | 95 | 107 | 1,26 |
| Votação média | 45.460 | 5,62 | 1,92 | 0 | 5 | 6 | 6,8 | 1,00 |
| Contagem de votos | 45.460 | 1,10 | 4,91 | 0 | 3 | 10 | 34 | 1,41 |

Elaborado pelo autor, 2023.

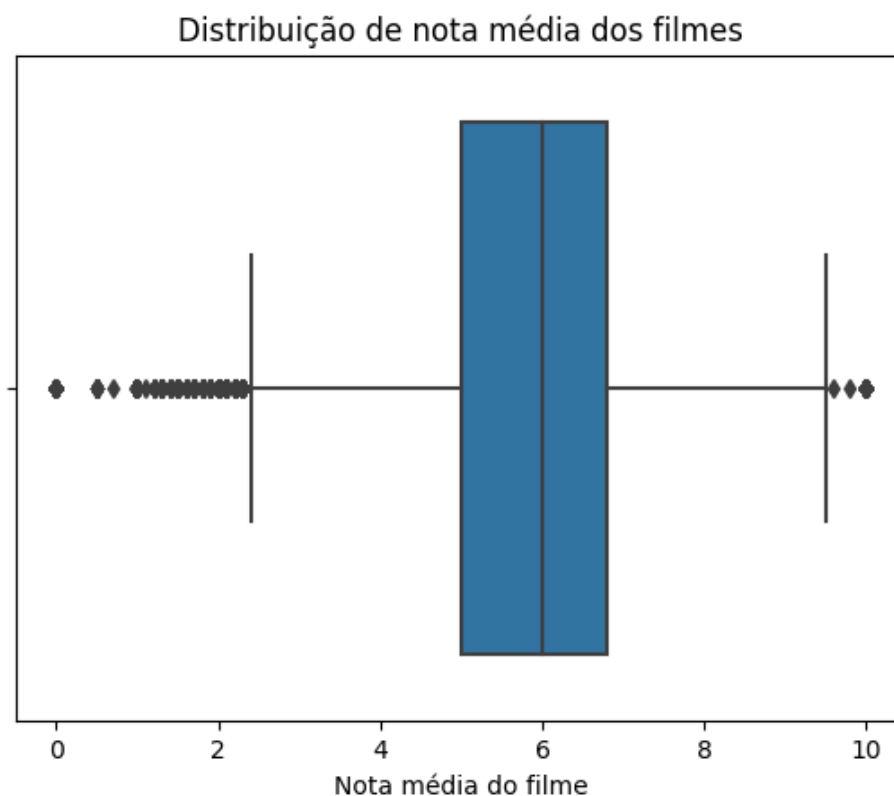
Na tabela resultante, é possível verificar, por exemplo, que o valor mínimo para votação média (`vote_average`) é 0, e o máximo é 10. Essa é uma maneira tabular de visualizarmos essas informações, e ela nos permite, inclusive, verificar a mediana (que figura na linha 50%, e que representa o valor que divide o conjunto de dados ao meio - neste caso, 6.0) e os quartis (25% e 75%). Uma maneira gráfica de visualizarmos essas informações é o histograma com o seaborn, utilizando a função `distplot()`, conforme demonstrado nas figuras 5 e 6.

Figura 5: Gráfico distplot



Fonte: Google Colab – Elaborado pelo autor, 2023.

Figura 6: Gráfico distplot



Fonte: Google Colab – Elaborado pelo autor, 2023.

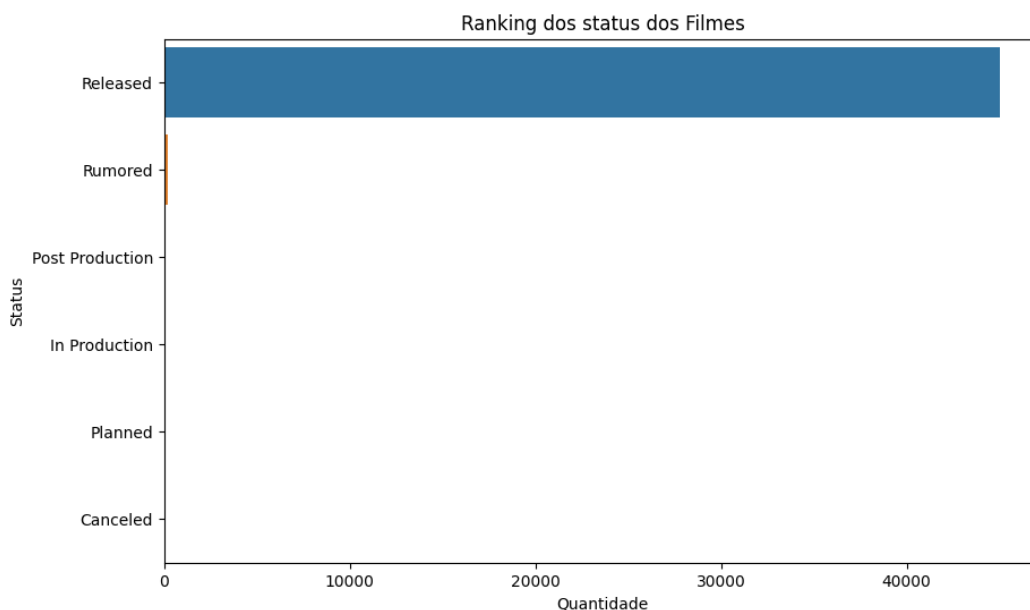
Com o recurso de groupby e nunique analisamos o status dos filmes sem repetição, onde obtivemos o resultado apresentado na figura 7 (demonstrado em tabela) e figura 8 (demonstrado em gráfico de barras), que do total de filmes que constam na base de dados, 99% são filmes lançados.

Figura 7: Tabela de Status dos filmes

| | Status | Imdb_id |
|---|--------------|---------|
| 0 | Cancelado | 2 |
| 1 | Em Produção | 20 |
| 2 | Planejado | 15 |
| 3 | Pós-produção | 98 |
| 4 | Lançado | 44970 |
| 5 | Rumores | 227 |

Elaborado pelo autor, 2023.

Figura 8: Status em Gráfico de barras

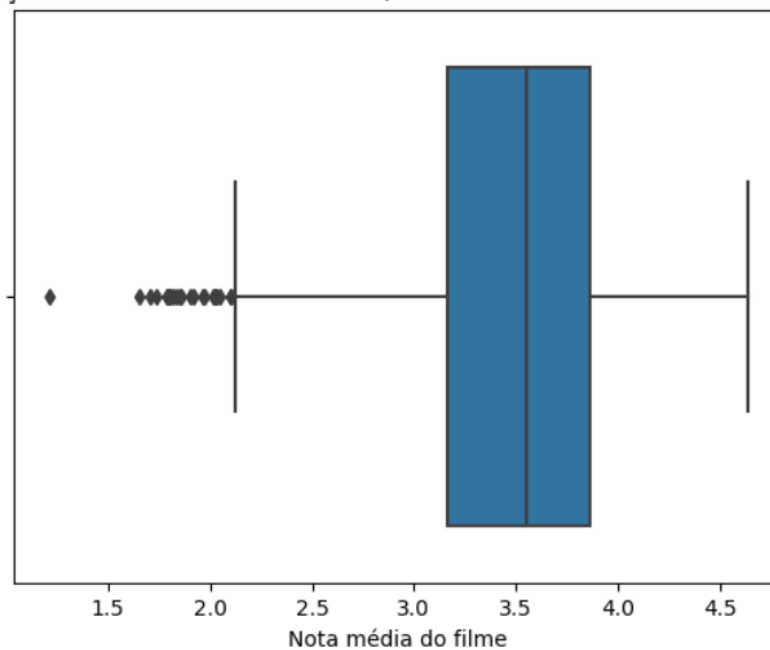


Fonte: Google Colab – Elaborado pelo autor, 2023.

Realizamos também análise na base de dados de avaliações dos filmes “ratings_small.csv” para entender a média de votos, distribuição de notas médias dos filmes, onde foi considerado apenas os filmes com mais de 10 votos, onde representamos esta análise em um gráfico de boxplot representado na figura 9, que identifica a concentração da nota média em torno de 3,5.

Figura 9: Gráfico de boxplot

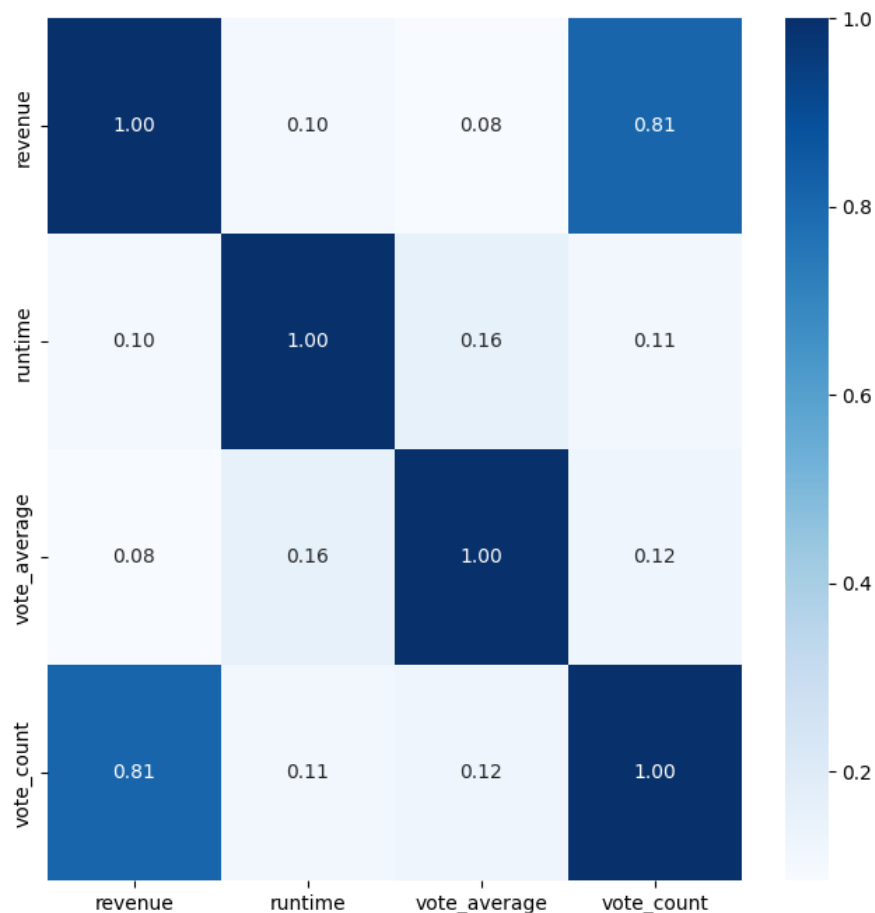
Distribuição de nota média dos filmes, dentre os filmes com 10 ou mais votos



Fonte: Google Colab – Elaborado pelo autor, 2023.

Foi realizado a correlação entre as variáveis: receita, tempo de execução, votação média e contagem de votos, utilizando o comando de “corr”, apresentando as informações em um indicador de heatmap (mapa de calor), conforme demonstrado na figura 10, onde se análise que as variáveis se correlacionam entre si.

Figura 10: Gráfico de heatmap

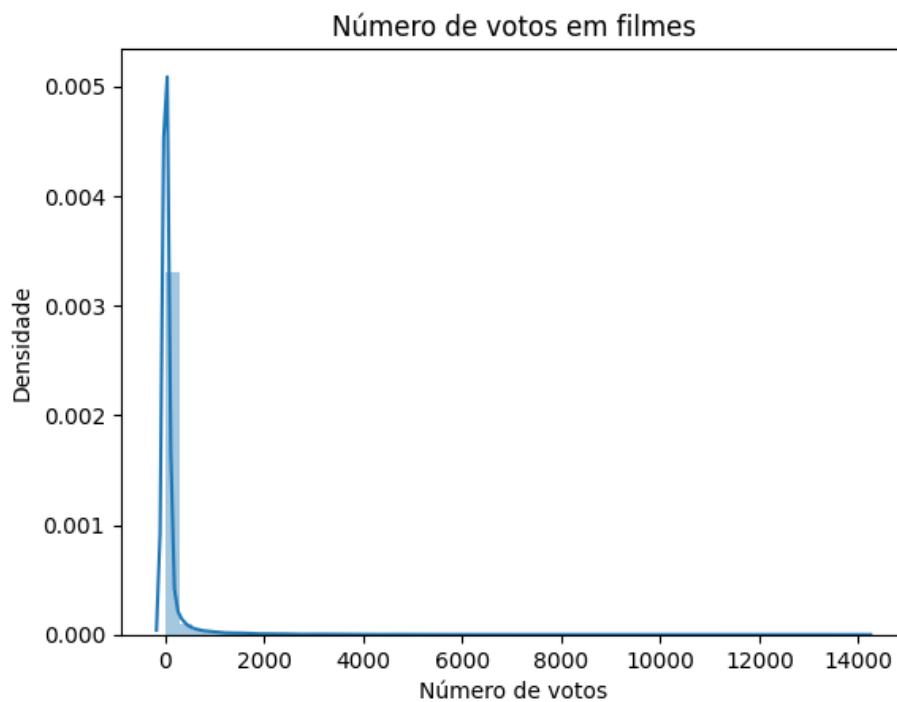


Fonte: Google Colab – Elaborado pelo autor, 2023.

Avaliamos na base de dados através do comando de “query” quais filmes possuem votos igual a 0 e igual a 10. Através do comando de `isnull().sum()` analisamos a soma de valores vazios onde foi identificado para o atributo tempo de execução que constam 257 sem preenchimento da informação.

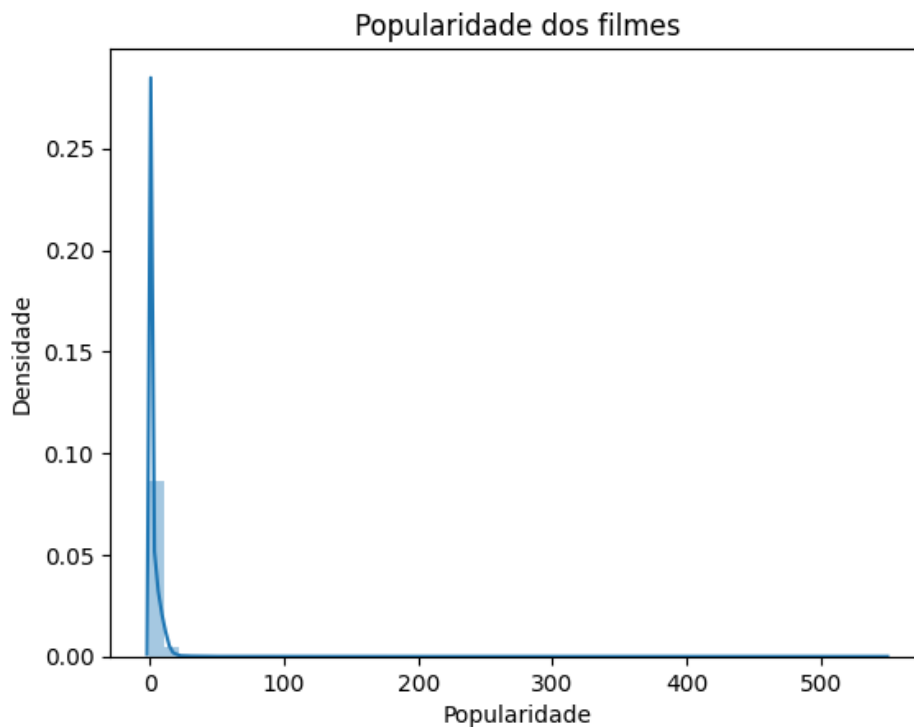
Através do gráfico de distribuição “distplot”, realizamos análise de quantidade de votos dos filmes apresentado na figura 11, popularidade dos filmes apresentado na figura 12 e a duração dos filmes apresentado na figura 13.

Figura 11: Gráfico de Número de votos em filmes



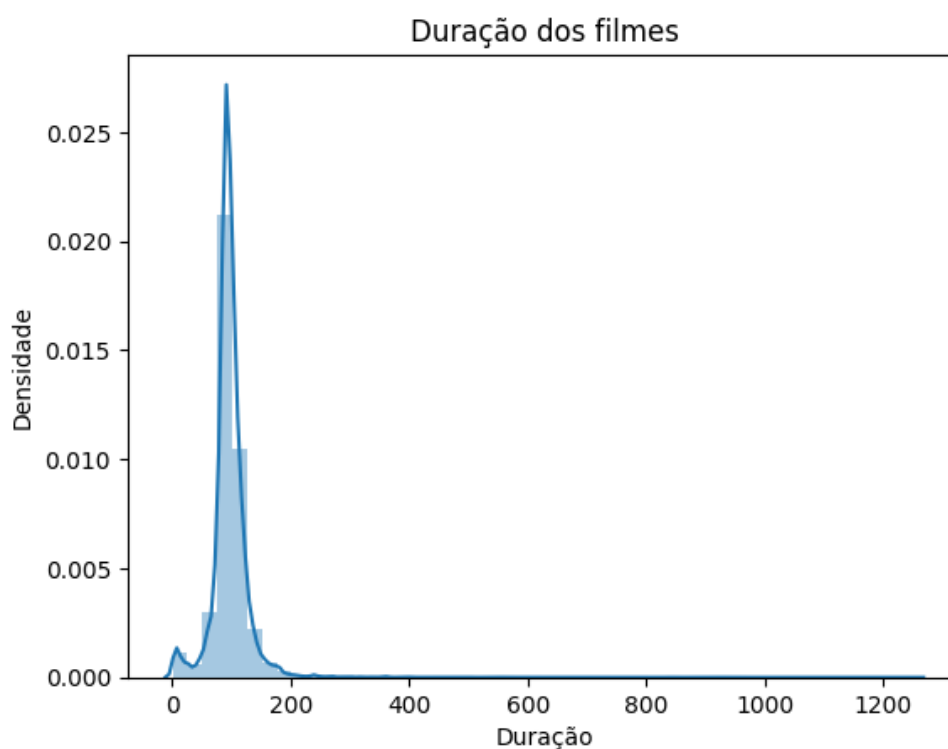
Fonte: Google Colab – Elaborado pelo autor, 2023.

Figura 12: Gráfico de Popularidade de Filmes



Fonte: Google Colab – Elaborado pelo autor, 2023.

Figura 13: Gráfico de Duração de Filmes



Fonte: Google Colab – Elaborado pelo autor, 2023.

Através da aplicação do comando “from statsmodels.stats.weightstats import zconfint”, foi realizado um teste da amostra, aplicando o intervalo de confiança da média com t e ztest, no qual foi analisando as médias dos filmes. Isso não quer dizer que todos os filmes do mundo terão uma média 3.47, que é a média dos filmes da base de dados analisada que apresentam pelo menos 10 votos, mas queremos criar uma generalização, a partir dessa amostra, sobre os filmes que existem no mundo.

O detalhamento completo da análise exploratória do dataset está disponível no GitHub, no link: https://github.com/OtavioBer/ProjetoAplicadoII/blob/main/Projeto_Aplicado_II.ipynb

10. TRATAMENTO DE DADOS

Para o desenvolvimento de um sistema de recomendação de filmes, uma abordagem fundamentada na filtragem colaborativa foi utilizada. Este trabalho visa apresentar uma solução que se destaca pela eficiência e pela capacidade de proporcionar recomendações de filmes altamente satisfatórias aos usuários.

Para que este sistema seja estruturado de forma mais assertiva o tratamento dos dados é essencial. E este processo requer seguir as etapas ilustradas na figura 14, são métodos indicados dentro da ciência de dados para a análise em qualquer área de exploração desejável.

Figura 14: Processo de tratamento de dados



Elaborado pelo autor, 2023.

Neste contexto, descreveremos a metodologia utilizada e uma análise aprofundada do código Python desenvolvido. Para iniciar a criação do código, importamos os pacotes pandas e numpy e os Datasets que serão utilizados.

Foi implementado uma seleção das colunas dos Dataset “Filmes” e de “Avaliações” que utilizaremos no sistema de recomendação e com a aplicação do comando de “rename columns” renomeamos algumas de suas variáveis.

Para encontrar valores nulos foi utilizado a função “isna” e devido ter identificado poucos valores ausentes, fizemos a remoção dos dados, pois não teremos nenhum impacto, aplicando o comando de “dropna(inplace = True)”.



No Dataset “Avaliações” aplicamos um comando para verificar a quantidade de avaliações feitas por usuários. Para o nosso sistema utilizaremos somente os usuários que fizeram mais de 400 avaliações, e para separar os dados em um novo dataframe aplicamos o comando: `qt_avaliacoes = avaliacoes[ID_USUARIO].value_counts() > 400`”.

Com a avaliação feita acima vimos que ainda temos um Dataset com mais de 20 milhões de linhas, com isso, resolvemos utilizar apenas os usuários que avaliaram mais de 400 vezes. Com isso reduzimos o tamanho do Dataset pela metade.

Para nosso sistema de recomendação utilizaremos filmes que possuem uma quantidade de avaliações superior a 30. Após a seleção dos filmes agrupamos os mesmos pela sua respectiva linguagem, aplicando o código a seguir: `filmes = filmes[filmes[QT_AVALIACOES] > 30]`”.

No sistema de recomendação utilizaremos apenas filmes na linguagem inglês (EN), aplicando a seleção: `filmes = filmes[filmes[LINGUAGEM] == 'en']`”.

Precisamos converter a variável ID_FILME em número inteiro aplicando o comando: `filmes[ID_FILME] = filmes[ID_FILME].astype(int)`”.

Através do comando merge realizamos a concatenação dos dataframes de filmes e avaliações, com o seguinte código: `avaliacoes_e_filmes = avaliacoes.merge(filmes, on = 'ID_FILME')`”. Após a concatenação dos Dataframes verificamos se há valores duplicados e descartamos os mesmos para que não tenhamos problemas de termos o mesmo usuário avaliando o mesmo filme mais de uma vez. E para remover valores duplicados aplicamos o código: `avaliacoes_e_filmes.drop_duplicates(['ID_USUARIO', 'ID_FILME'], inplace = True)`”.

O detalhamento completo do tratamento do dataset está disponível no GitHub, no link: https://github.com/OtavioBer/ProjetoAplicadoll/blob/main/Collam_FilmesV5.ipynb.

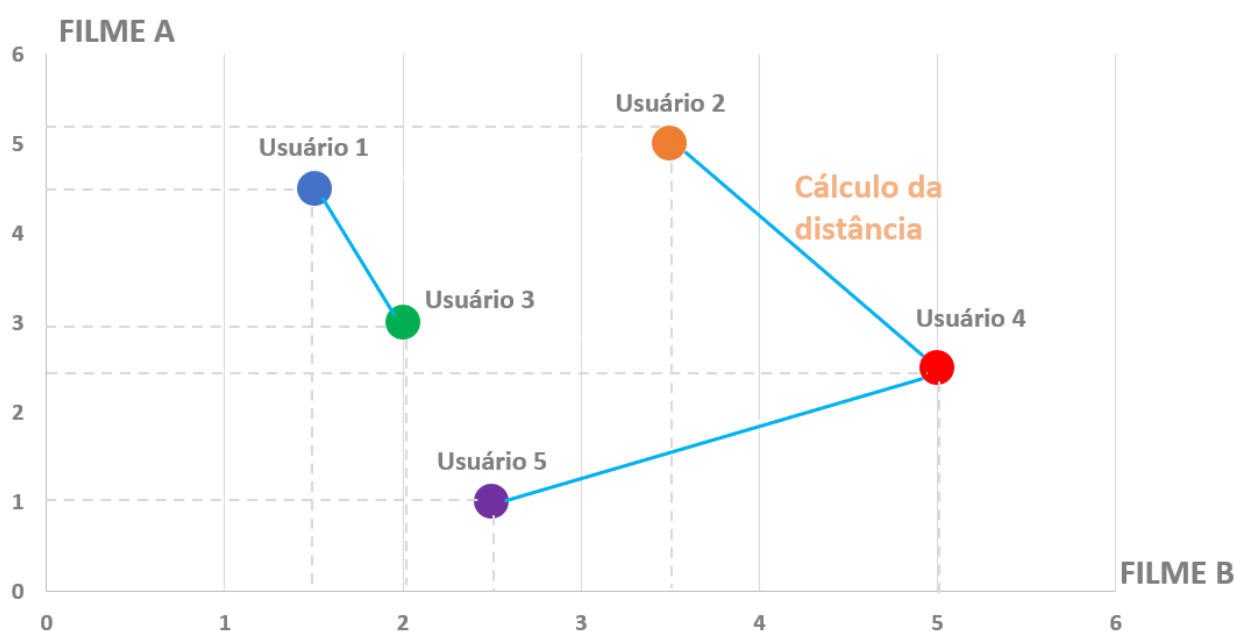
11. MÉTODO DE RECOMENDAÇÃO

Para realizar a aplicação de análise de sistema de recomendação, utilizamos do método de cálculo para definir a distância euclidiana entre usuários, vamos utilizar os dados que viemos trabalhando para criarmos a base de um algoritmo de Machine Learning. No nosso conjunto de dados, temos duas informações muito importantes: as características dos filmes e as notas que os usuários deram a eles.

O método de recomendação em aprendizado de máquina envolve a previsão de preferências do usuário com base em dados históricos. A distância euclidiana é uma métrica utilizada para medir a semelhança entre dois pontos em um espaço multidimensional, na figura 15 representamos como funciona este processo do cálculo da distância entre os usuários com base nas notas de cada um para determinado filme. No contexto de recomendação, essa distância pode ser aplicada para calcular a proximidade entre perfis de usuários ou itens. A similaridade entre dois pontos é medida pela distância euclidiana entre eles. Quanto menor a distância, maior a semelhança.

No caso de recomendação de itens, por exemplo, os usuários com históricos semelhantes terão perfis mais próximos no espaço, indicando gostos similares. Assim, itens preferidos por usuários com perfis próximos podem ser recomendados.

Figura 15: Distância Euclidiana



Elaborado pelo autor, 2023.



Matriz de usuários x filmes

1. Cria uma matriz, "matriz_usuario_filme" que representa as avaliações dos usuários para diferentes filmes.
2. Extrai os títulos únicos de filmes da matriz de avaliações.
3. Cria um Dataframe "df_titulos_unicos" com os títulos únicos e o exporta para um arquivo Excel chamado "titulos_unicos.xlsx".
4. Chama a função "recomendar_filmes_por_titulo()" com o título do filme "10,000 BC" e recupera as recomendações com base na similaridade euclidiana.
5. Imprime as recomendações.

A implementação da matriz de avaliações de usuários para filmes utilizando a função "pivot_table" para organizar as avaliações dos usuários para diferentes filmes. A lista de títulos de filmes únicos é extraída do Dataframe "avaliacoes_e_filmes". Por fim chamamos a função "recomendar_filmes_por_titulos" para obter recomendações de filmes com base em um filme de referência. O número de recomendações definido foi 3. Após executar a função acima as recomendações serão impressas no console.

Em resumo, o código é uma implementação básica de um sistema de recomendação de filmes que utiliza a distância euclidiana para encontrar filmes similares com base nas avaliações dos usuários. O resultado é uma lista de filmes recomendados com base na similaridade de avaliações em relação ao filme de entrada. O número de recomendações é definido como 3 por padrão, mas pode ser ajustado.

Com base nessas distâncias, saberemos dizer quais usuários são mais similares entre si, o que servirá de base para implementarmos a continuação do algoritmo.

Medida de Acurácia:

Em sistemas de recomendação de filmes, a medida de acurácia é usada para avaliar o desempenho do sistema, geralmente comparando as recomendações feitas pelo sistema com as classificações reais dos usuários. Uma métrica comum usada para isso é a "Mean Absolute Error" (Erro Médio Absoluto - MAE). O MAE mede a diferença absoluta entre as classificações previstas e as reais. Quanto menor o MAE, melhor o desempenho do sistema.

A função "calcular_mae" calcula o MAE com base nas recomendações feitas pelo sistema e nas classificações reais dos usuários. Ele percorre as recomendações, calcula o erro absoluto para cada recomendação e, em seguida, calcula a média desses erros.

O MAE mede o erro médio entre as classificações previstas e as reais. Quanto menor o valor do MAE, melhor é o desempenho do sistema de recomendação, pois indica que as recomendações estão mais próximas das classificações reais dos usuários.

Essa métrica de acurácia ajuda a avaliar o quão bom o sistema de recomendação está funcionando e pode ser usada para ajustar e otimizar o algoritmo de recomendação.

Avaliação de Similaridade

A distância euclidiana é uma métrica de analisar a similaridade entre filmes, isso implica em medir a diferença entre os gostos ou avaliações dadas por esses usuários aos filmes. Quanto menor a distância euclidiana, maior a similaridade entre os usuários. Segue uma descrição das funções e do fluxo do código que foi construído, a aplicação do código de calcular a distância euclidiana realiza as seguintes etapas:

- A função “`calcular_distancia_euclidiana (filme1, filme2, matriz_usuario_filme)`”: É aplicada para calcular a distância euclidiana (distância entre dois pontos, que pode ser provada pela aplicação repetida do teorema de Pitágoras) entre dois filmes, representados por “filme1” e “filme2”, com base nas avaliações dadas pelos usuários em “matriz_usuario_filme”.
- Já a função “`recomendar_filmes_por_titulo (titulo_filme, matriz_usuario_filme, dados, num_recomendacoes=3)`”: Recebe um título de filme (“titulo_filme”) como entrada e calcula a distância euclidiana entre esse filme e todos os outros filmes na matriz, “matriz_usuario_filme”. Em seguida, ela classifica os filmes com base na distância euclidiana, retornando os filmes mais similares como recomendações.
- A função “`user_similarity`”, calcula a similaridade entre dois usuários com base na distância euclidiana de suas avaliações de filmes. Quanto mais próximas as avaliações de dois usuários, maior será a similaridade.
- A função “`genre_similarity`”, calcula a similaridade entre dois conjuntos de gêneros de filmes. Se os conjuntos forem iguais, a similaridade é 1; caso contrário, é 0.

Implementando a Recomendação

Neste método os gêneros do filme de referência são comparados com os gêneros de outros filmes. Essa comparação gera uma pontuação de similaridade de gêneros para cada filme, proporcionando uma abordagem mais abrangente para as recomendações, considerando tanto as preferências dos usuários quanto os gêneros dos filmes.

- Recebe o título de um filme de interesse e um DataFrame “`avaliacoes_e_filmes`” contendo avaliações de filmes.
- Filtra o DataFrame para excluir o filme de referência.



- Calcula a similaridade entre os usuários (com base nas avaliações) e entre os gêneros dos filmes.
- Combina as similaridades com pesos de 70% para similaridade de usuários e 30% para similaridade de gêneros.
- Ordena os filmes por sua pontuação de similaridade total e retorna as principais recomendações.

Recomendação de Filmes

O sistema de recomendação utiliza da similaridade que foi construída e explicada no tópico anterior para comparar o filme de referência (por exemplo, “Toy Story”) com outros filmes, buscando usuários que tenham avaliações semelhantes. Usando a função “sistema_de_recomendacao” informando um filme de preferência o sistema recomenda filmes de similaridade. Realizamos o teste para recomendações dos filmes:

- ✓ “Toy Story” – recebendo indicação dos filmes: A Close Shave e Monsters, Inc.
- ✓ “Sin City” – recebendo indicação dos filmes: Raw Deal, Kill Bill: Vol. 2, Point Break.
- ✓ “A Close Shave” – recebendo indicação do filme: Meet the Robinsons.
- ✓ “Point Break” – recebendo indicação dos filmes: Raw Deal, Kill Bill: Vol. 2, Kiss of the Dragon.

Detalhamento completo do método de recomendação está disponível no GitHub, no link: https://github.com/OtavioBer/ProjetoAplicadoll/blob/main/Collam_Filmes_V6.ipynb

12. CONCLUSÃO

Nosso estudo possibilitou uma exploração minuciosa e eficaz no desenvolvimento de um sistema de recomendação, utilizando o método de distância euclidiana e adotando uma abordagem integrada à similaridade entre usuários e gêneros de filmes. O processo, desde a manipulação e tratamento dos dados até a implementação da função de recomendação, foi conduzido de maneira meticulosa, evidenciando um entendimento aprofundado dos conceitos e técnicas envolvidos.

Ao avaliarmos o desempenho do sistema com filmes específicos, como "Toy Story" e "Sin City", constatamos resultados satisfatórios, revelando uma notável capacidade de sugerir filmes com base na afinidade, tanto em termos de avaliações quanto de gêneros.

É crucial ressaltar que todo o processo de limpeza e processamento de dados, especialmente no que diz respeito à gestão de valores ausentes e à seleção criteriosa de usuários com um número substancial de avaliações, desempenhou um papel fundamental na robustez e confiabilidade dos resultados apresentados. Nosso trabalho não se limitou a oferecer uma implementação prática e funcional do sistema de recomendação, mas também enfatizou a importância de aspectos essenciais no processo, contribuindo para uma compreensão mais profunda do papel desses sistemas e suas aplicações.

Dessa forma, a exportação da lista de títulos proporciona uma ferramenta valiosa para futuras análises, evidenciando nosso comprometimento com a utilidade prática dos resultados obtidos. Assim, nossa pesquisa não apenas demonstra habilidades técnicas sólidas, mas também reflete uma compreensão reflexiva e estratégica das nuances envolvidas na implementação de sistemas de recomendação baseados na distância euclidiana.

Podemos ressaltar a importância do tratamento de dados para o sucesso do sistema de recomendação baseado em filtragem colaborativa. Destacamos que as etapas apresentadas neste projeto representam métodos consagrados na ciência de dados, contribuindo para a estruturação eficiente do sistema. A abordagem adotada visa assegurar a qualidade e precisão das recomendações de filmes, demonstrando a robustez do processo de análise de dados empregado.

Concluimos que a implementação bem-sucedida do sistema de recomendação de filmes, utilizando o código em Python com funções de distância euclidiana e avaliação de similaridade, demonstrou eficácia na identificação de padrões de preferência do usuário. A análise



comparativa das avaliações de filmes permitiu uma recomendação precisa, contribuindo para uma experiência personalizada. A aplicação do modelo evidenciou sua capacidade de lidar com diferentes conjuntos de dados, destacando a viabilidade e utilidade prática desse sistema na recomendação de filmes da empresa “COLLAM FILMS”.

13. DIRETÓRIO GITHUB

Todo o conteúdo do projeto estará disponível no site da GitHub, que poderá ser acessado pelo link:

<https://github.com/OtavioBer/ProjetoAplicadoII>

O diretório está organizado por pastas:

Pasta “Códigos” será disponibilizado os códigos em Python que foram utilizados para realizar a análise exploratória, tratamento dos dados e o sistema de recomendação.

Pasta “Dados” temos os arquivos utilizados para o estudo.

Pasta “Documentos” temos o cronograma de entregas do projeto, as versões de entrega deste documento.

Temos também o arquivo README.md com informações relevantes do projeto.

14. REFERÊNCIAS BIBLIOGRÁFICAS

NETTO, Amilcar; NETO, Francisco. Python Para Data Science e Machine Learning Descomplicado. Rio de Janeiro: Alta Books, 2021.

GRUS, Joel; Data Science do Zero. Editora Alta Books, 2021.

RICCI, F.; ROKACH, L.; SHAPIRA, B. Introdução ao Manual de Sistemas de Recomendação. Springer, 2015.

KOREN, Y.; BELL, R. Avanços na filtragem colaborativa. Springer, 2015.

Su, X.; KHOSHGOFTAAR, TM. Uma pesquisa de técnicas de filtragem colaborativa. Avanços em Inteligência Artificial, 2009.

HERLOCKER, JL; KONSTAN, JA; TERVEEN, LG; RIEDL, JT. Avaliando Sistemas de Recomendação de Filtragem Colaborativa. Transações ACM em Sistemas de Informação, 2004.

RESNICK, P.; VARIAN, HR. Sistemas de recomendação. Comunicações da ACM, 1997.

Material de apoio dos componentes curriculares: Aprendizado de Máquina, Análise Estatística Preditiva, Aquisição e Preparação de Dados, Introdução a Engenharia e Tópicos de Banco de Dados.

<https://www.kaggle.com/datasets/rounakbanik/the-movies-dataset/>

<https://www.diva-portal.org/smash/get/diva2:927356/FULLTEXT01.pdf>

<https://www.voitto.com.br/blog/artigo/biblioteca-pandas>

<https://medium.com/ensina-ai/entendendo-a-biblioteca-numpy-4858fde63355>

<https://www.crawly.com.br/blog/python-e-big-data-fique-por-dentro-de-3-bibliotecas-essenciais>

<https://dadosaocubo.com/analise-de-dados-com-seaborn-python/>

<https://surpriselib.com/>

<https://awari.com.br/scikit-learn/>

<https://pt.wikipedia.org/wiki/Scikit-learn>

<https://gepac.github.io/2019-05-17-intro-scipy/>

<http://pyscience-brasil.wikidot.com/module:scipy>

<https://sicit.uit.br/wp-content/uploads/2018/05/APO9.pdf>

<https://mariofilho.com/mae-erro-medio-absoluto-em-machine-learning/#:~:text=A%20f%C3%B3rmula%20do%20erro%20m%C3%A9dio,erros%20de%20todas%20as%20amostras.>



15. ANEXOS

✓ **Cronograma:**

O desenvolvimento do cronograma do projeto, foi realizado para reportar-se o percentual de evolução de entregas referente as ações propostas pelo componente curricular de Projeto Aplicado II do curso de Tecnologia em Ciências de Dados em cada etapa. E está disponível no GitHub no link:

https://github.com/OtavioBer/ProjetoAplicadoII/blob/main/Documentos/Cronograma_Collam%20Films.pdf

✓ **Storytelling:**

O desenvolvimento da narrativa deste projeto encontra-se disponível no GitHub no link:

https://github.com/OtavioBer/ProjetoAplicadoII/blob/main/Documentos/Storytelling_Collam%20Films_V1_061123.pdf

✓ **Figuras:**

A elaboração das figuras que foram elaboradas e apresentadas neste projeto está disponível no GitHub no link:

https://github.com/OtavioBer/ProjetoAplicadoII/blob/main/Documentos/Figuras_Collam%20Films.pdf

✓ **Vídeo:**

A gravação do vídeo realizada com a apresentação do projeto, com a participação de todos os integrantes do grupo está disponível no Youtube no link: <https://youtu.be/aQOG7zXN9LE>

✓ **Código do Sistema de Recomendação:**

O detalhamento completo do método de recomendação está disponível no GitHub, no link: https://github.com/OtavioBer/ProjetoAplicadoII/blob/main/Collam_Filmes_V6.ipynb

✓ **Análise Exploratória:**

O detalhamento completo da análise exploratória do dataset está disponível no GitHub, no link: https://github.com/OtavioBer/ProjetoAplicadoII/blob/main/Projeto_Aplicado_II.ipynb