LADATA
liga acadêmica de ciência de dados

Autoria do material: Sérgio Manhães, Francisco Almir, Thiago Castro, Edimilton Rocha

Área: Ciência de Dados

Palavras-chave: Ciência de dados, Ferramentas, Programação

Guia LADATA de ferramentas para Ciência de Dados

2024, São Cristóvão

Sumário

1. Introdução

O estudo da Ciência de dados é algo que está em seu estado de ascensão na atualidade e que, por isso, passa por muitas atualizações em suas tecnologias, sempre em busca de soluções mais eficientes, mais baratas e com mais funcionalidades que resolvam seus problemas. Com isso, esse documento tem como objetivo apresentar ferramentas utilizadas no contexto empresarial e acadêmico no mundo dos dados.

2. Fundamentação Teórica

2.1. Versionamento de Código

O uso de versionamento de código na computação se tornou um pilar muito importante, principalmente pela colaboração mútua em grandes projetos, dada a necessidade de grandes equipes atualizarem de maneira não concorrente o código e arquivos de projetos.

No contexto desse material, o foco será na ferramenta Git e suas plataformas (Github/Gitlab), por conta de seu compartilhamento no mercado e a requisição de tal tecnologia nas vagas de emprego da área. O controle de versões nessa tecnologia é feito pelo uso de branches para trabalho paralelo, de commits para publicar versões. Além disso, as plataformas contam com diversas ferramentas de gerenciamento de projetos e feedback entre desenvolvedores e gestores, por exemplo, podendo se estender para os Stakeholders.

2.2. Visualização de Dados

A visualização de dados é um componente essencial da análise de dados, pois permite traduzir grandes volumes de informações complexas em representações gráficas compreensíveis e intuitivas. No contexto da ciência de dados, a visualização atua como uma ponte entre os dados brutos e os insights, auxiliando na comunicação eficaz de padrões, tendências e anomalias.

No contexto de Python, a linguagem oferece uma variedade de ferramentas para visualização, cada uma atendendo a diferentes necessidades. Matplotlib é amplamente reconhecida por sua flexibilidade e controle detalhado sobre gráficos, sendo uma das bibliotecas pioneiras nesse domínio. Seaborn, construída sobre o Matplotlib, proporciona uma interface mais amigável e é especialmente útil para visualizações estatísticas, além do mais atual e bastante utilizado Streamlit, que se baseia em criar uma plataforma completa de visualização em formato de portal Web . Em suma, a visualização de dados em Python é um componente indispensável na exploração e comunicação de dados, contribuindo significativamente para a interpretação e disseminação de informações.

2.3. Engenharia de Dados

A engenharia de dados é uma área fundamental no ecossistema de ciência de dados, sendo responsável por estruturar, processar e disponibilizar dados de maneira eficiente e escalável. Seu objetivo principal é transformar dados brutos, provenientes de diversas fontes, em informações organizadas e prontas para análises. Essa prática é indispensável em um mundo orientado por dados, onde a tomada de decisão depende cada vez mais da capacidade de processar grandes volumes de informações em tempo real.

A importância da engenharia de dados reside na sua habilidade de garantir a qualidade e integridade dos dados, permitindo que empresas e organizações extraiam valor de suas informações. Além disso, a construção de pipelines eficientes possibilita a integração de dados de diferentes fontes, promovendo uma visão unificada que serve como base para análises estratégicas.

2.4. Ciência de Dados

A ciência de dados é um campo interdisciplinar que combina técnicas de estatística, matemática, computação e conhecimento de domínio para extrair informações e gerar insights a partir de dados. No cenário atual, onde grandes volumes de informações são gerados continuamente, a ciência de dados desempenha um papel crucial ao transformar esses dados brutos em conhecimento útil e acionável. Essa área vai além da simples análise de dados, englobando processos que incluem a coleta, organização, processamento, modelagem e interpretação de informações, frequentemente com o apoio de ferramentas avançadas de aprendizado de máquina e inteligência artificial.

3. Resumo das Ferramentas

3.1. Codificação online

3.1.1. Jupyter Notebooks

O Jupyter Notebooks é um ambiente interativo que permite criar e compartilhar documentos contendo código, texto explicativo, gráficos e resultados de execução, sendo muito utilizado para análises exploratórias e prototipagem. Um dos seus pontos mais fortes é o intercalamento entre código e escrita, o que torna a visualização e compreensão mais intuitiva.

3.1.2. Kaggle

O Kaggle é uma plataforma voltada para a comunidade de ciência de dados que oferece datasets prontos, competições de aprendizado de máquina e um ambiente baseado em notebooks para execução de código diretamente na nuvem, promovendo o aprendizado e a colaboração entre profissionais.

3.1.3. Google Colab

O Google Colab é um serviço gratuito que combina a flexibilidade do Jupyter com o poder de processamento do Google, oferecendo notebooks na nuvem com acesso a GPUs e TPUs para projetos que exigem maior capacidade computacional. Essas ferramentas complementam-se e são indispensáveis para pesquisadores e engenheiros que trabalham com dados e inteligência artificial.

3.2. Aquisição de dados

Além da estruturação e ferramentas de desenvolvimento de código, é importante falar das opções de aquisição de dados para o tratamento posterior.

Uma das maneiras mais comuns de aquisição de dados é a utilização de tabelas já prontas, muitas vezes povoadas manualmente por empresas, ou geradas automaticamente por meio de relatórios, etc. Nesses exemplos, é muito comum que esses arquivos estejam no formato de tabelas no Excel. A grande vantagem de saber tratar esse tipo de tabelas, é a maior gama de bases de dados nesse formato, contudo vemos muitos exemplos com dados incompletos e com algumas inconsistências de tipo, um problema que pode ser contornado na etapa de tratamento.

Além disso, existem outras fontes de dados e de bases de dados. Dois exemplos muito famosos e em ascensão hoje em dia, que são o Kaggle e o HuggingFace. Que são portais que contém desde bases de dados, geralmente tratadas, até modelos de aprendizagem de máquina e interfaces de programação online. Por serem sites da comunidade, é muito interessante a variedade de temas que estes cobrem, o que é bastante interessante principalmente para quem está começando no mundo da programação.

Por fim, um dos tipos de fontes de dados mais impactantes que podem ser encontrados são os dados abertos, isso pois esses incluem dados governamentais, que são de imensa importância, já que dizem bastante sobre informações públicas e que podem servir para análises sociais e econômicas.

3.3. ETL e Visualização de dados

ETL (Extract, Transform, Load) é um processo essencial no gerenciamento de dados que envolve a extração de informações de diversas fontes, sua transformação para padronização e limpeza, e, por fim, o carregamento desses dados em um repositório de destino, como um banco de dados ou data warehouse. A etapa de extração coleta dados brutos de sistemas heterogêneos, como bancos de dados, APIs ou arquivos. Na fase de transformação, os dados passam por operações de limpeza, integração, normalização e enriquecimento para garantir consistência e qualidade. Por fim, na etapa de carregamento, os dados tratados são armazenados em um ambiente estruturado, preparados para análise e uso por ferramentas de Business Intelligence (BI) ou aprendizado de máquina.

O tratamento de dados é uma parte fundamental do ETL e visa garantir a qualidade e integridade das informações. Isso inclui remover duplicatas, corrigir erros, lidar com valores ausentes, identificar outliers e realizar transformações como conversões de tipos e agregações. Um processo de ETL bem estruturado melhora a confiabilidade das análises, otimiza processos de tomada de decisão e possibilita insights mais precisos a partir dos dados.

Para essa parte, geralmente quem fica encarregada, em projetos reais, é a pessoa engenheira de dados, que deve garantir que os dados a serem utilizados estejam disponíveis e íntegros. Para essa tarefa, existem tecnologias que são essenciais, como o Apache Airflow, AWS Glue, Apache Spark.

3.4. Colaboração e Aprendizagem

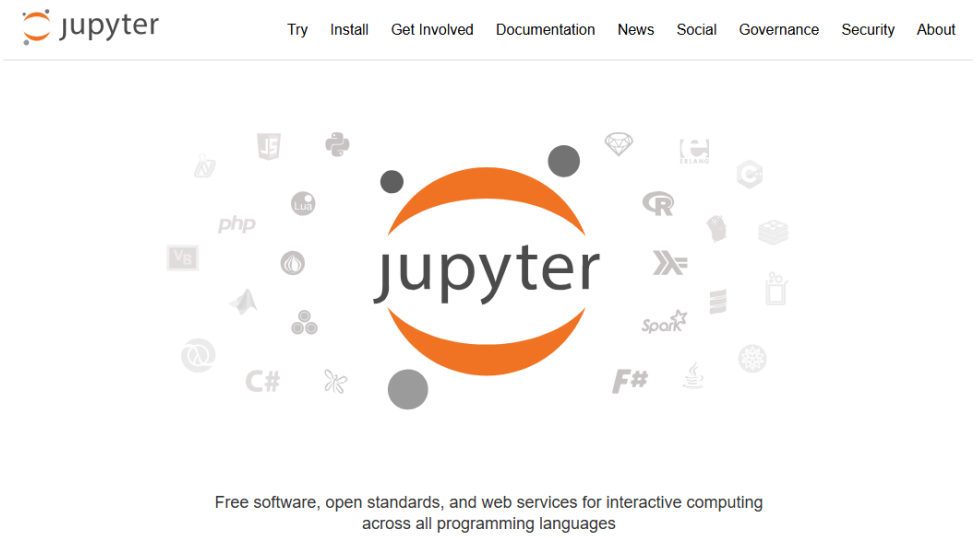
Outra coisa essencial quando trabalhamos com ciência de dados é o uso de ferramentas para compartilhamento e colaboração de código, dada a grande necessidade do trabalho mútuo e paralelo em projetos.

4. Acesso e uso das ferramentas

4.1. Jupyter Notebooks

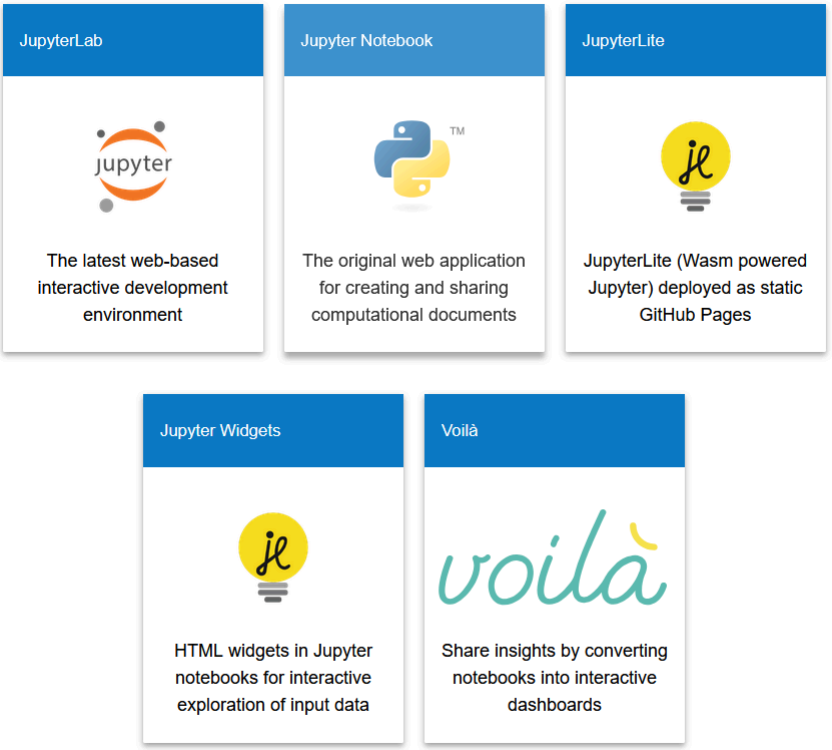
Como acessar:

Para fazer uso das funcionalidades da ferramenta, deve ser acessado o site oficial, disponível em https://jupyter.org/ :



Página Inicial jupyter

Ao selecionar a opção “Try”, será redirecionado para uma nova página com algumas opções de como utilizar a ferramenta, nesse caso iremos utilizar a versão disponibilizada para uso em nuvem, “**JupyterLab**”:



Lista de opções de como utilizar o jupyter

Ao selecionar a opção indicada, o notebook já estará disponível para uso e desenvolvimento de aplicações python e markdown.

Uso básico:

O jupyter notebook é uma ferramenta versátil que permite ao desenvolvedor flexibilidade para desenvolvimento do código fonte de sua aplicação e amostragem de dados, através do uso da linguagem de programação **python** e a linguagem de marcação **markdown** no corpo do notebook.

Principais funcionalidades:

Dentre as funcionalidades disponibilizadas pelo jupyter notebook, merecem destaque as seguintes:

- **Células interativas:** divisão do código em blocos menores para execução incremental.
- **Gráficos e visualizações integrados:** suporte a gráficos interativos diretamente no notebook.
- **Exportação de notebooks:** salvar notebooks criados em formato .ipynb ou como PDF/HTML

Exemplo prático: Analisar um Dataframe pandas sobre vendas e gerar um gráfico.

```
# Importando bibliotecas necessárias
import pandas as pd
import matplotlib.pyplot as plt
```

```
# Criando um DataFrame com dados fictícios de vendas
dados_vendas = {
    "produto": ["Notebook", "Mouse", "Teclado", "Monitor", "Cadeira", "Headset", "Impressora", "Smartphone"],
    "quantidade": [10, 50, 30, 20, 15, 25, 5, 18],
    "valor": [4500, 120, 200, 1000, 600, 350, 800, 2500],
    "data": ["2024-01-10", "2024-01-12", "2024-01-15", "2024-01-18", "2024-01-20", "2024-01-22", "2024-01-25", "2024-01-28"]
}
```

```
# Criando o DataFrame
df_vendas = pd.DataFrame(dados_vendas)
```

```
# Criando o DataFrame
df_vendas = pd.DataFrame(dados_vendas)
```

```
# Criando um gráfico de barras para visualizar o valor total por produto
plt.figure(figsize=(10, 6))
plt.bar(df_vendas['produto'], df_vendas['valor'], color='skyblue', edgecolor='black')
plt.title("Valor de Vendas por Produto", fontsize=16)
plt.xlabel("Produto", fontsize=12)
plt.ylabel("Valor (R$)", fontsize=12)
plt.xticks(rotation=45, fontsize=10)
plt.grid(axis='y', linestyle='--', alpha=0.7)
plt.tight_layout()
```

Código fonte de exemplo prático jupyter notebook

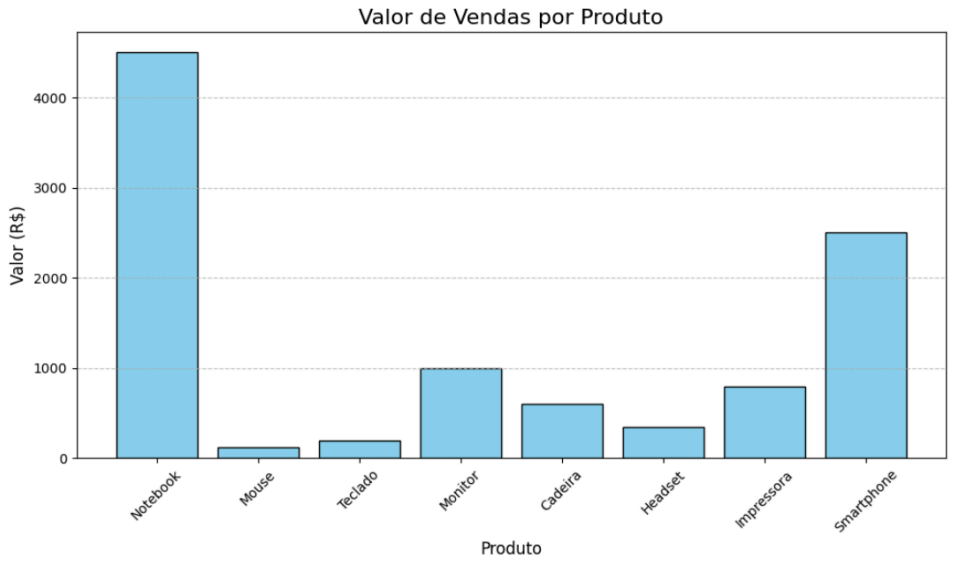


Gráfico de barras resultante do código fonte do exemplo prático

4.2. Kaggle

Como acessar:

Inicialmente a ferramenta está disponível para acesso online, através de um navegador, para ter acesso às funcionalidades, deve ser acessado o site oficial do

Kaggle, disponível em: https://www.kaggle.com/. Para começar, acesse sua conta ou crie uma nova caso seja o primeiro acesso.

Uso básico:

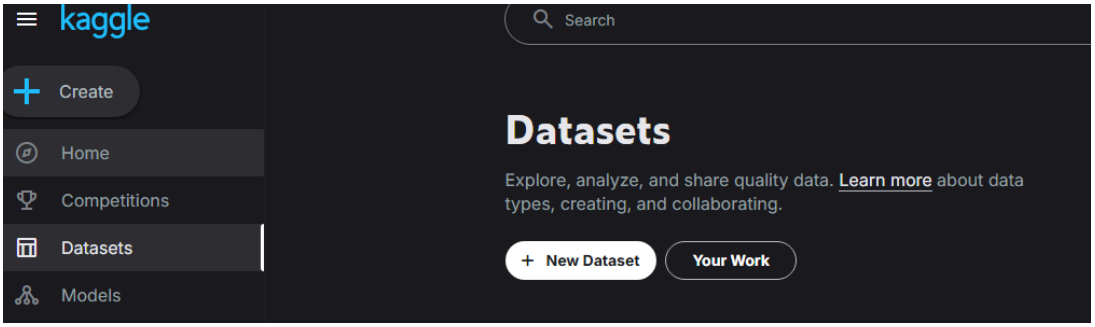
Uma das motivações de se utilizar a ferramenta é a possibilidade de participar em competições online relacionadas com a área de ciência de dados, onde são apresentados problemas e a comunidade deve desenvolver soluções adequadas, promovendo muitas vezes o trabalho colaborativo e uma rede de aprendizado mútua. Além disso, a ferramenta é utilizada devido a disponibilidade de acesso a datasets públicos que podem ser utilizados para projetos pessoais dos usuários.

Principais funcionalidades:

Dentre as funcionalidades disponibilizadas, destacam-se o ambiente de codificação online, que permite o desenvolvimento de projetos inteiramente em nuvem, garantindo versatilidade e segurança e o generoso repositório de datasets e notebooks compartilhados entre os usuários.

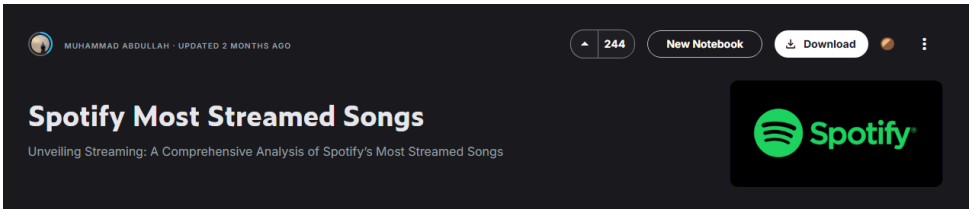
Exemplo prático: Criando um novo notebook e acessando um dataset público

Inicialmente, vamo acessar a aba de "Datasets" para encontrar um dataset para o nosso exemplo:



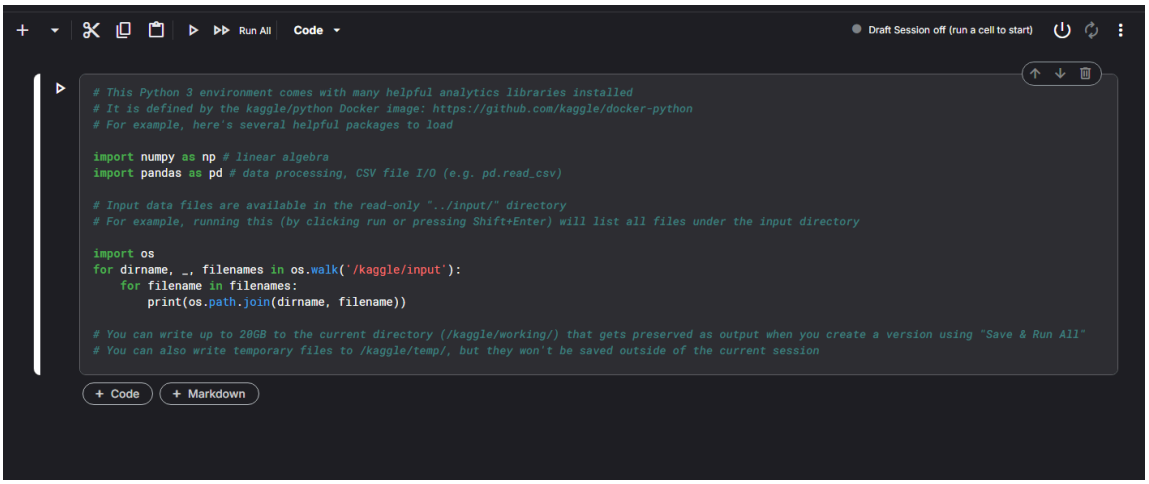
Aba de Datasets do Kaggle

Iremos escolher o dataset "Spotify Most Streamed Songs", ao acessar o dataset, teremos a opção de criar um novo notebook a partir do dataset selecionado, através da opção "**New Notebook**":

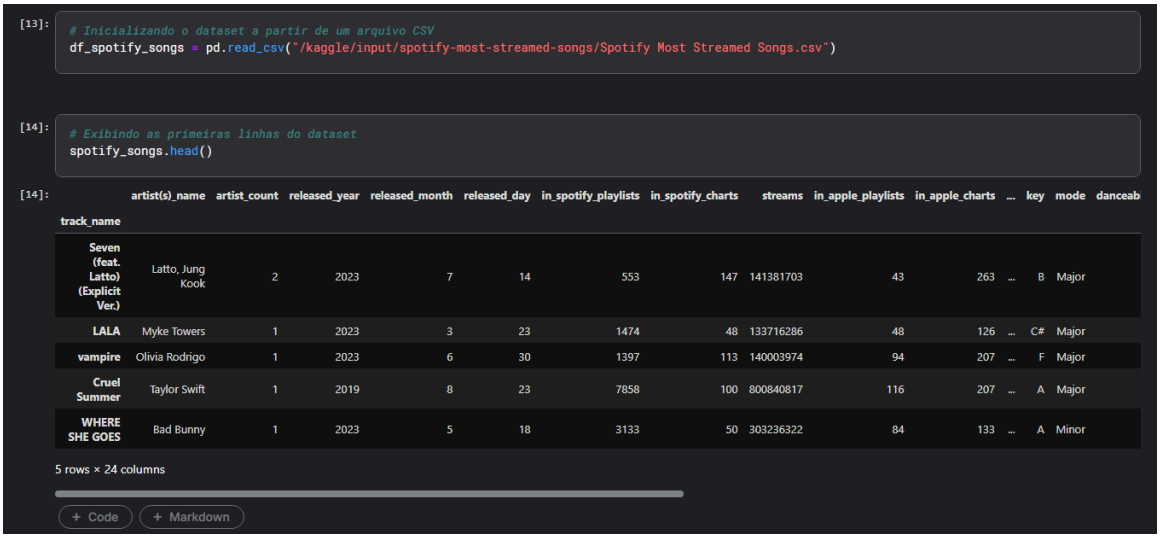


Página do dataset selecionado **"Spotify Most Streamed Songs"**

Ao criar um novo notebook, o kaggle nos redireciona para uma nova página já com o notebook aberto e para tornar ainda mais prático o desenvolvimento, já adiciona uma célula inicial importando bibliotecas importantes como **numpy** e **pandas**.



Visualização inicial ao criar um novo notebook a partir de um dataset



Criando dataframe e exibindo amostra dos dados

```
[15]: # Importando bibliotecas necessárias para análise e visualização
      import matplotlib.pyplot as plt

      + Code  + Markdown

[16]: # Preparando os dados: Convertendo a coluna 'streams' para numérica
      df_spotify_songs['streams'] = pd.to_numeric(df_spotify_songs['streams'], errors='coerce')

      # Ordenando os dados pelas músicas mais transmitidas
      top_songs = df_spotify_songs.nlargest(10, 'streams')

[17]: # Visualizando os 10 artistas mais transmitidos
      plt.figure(figsize=(12, 6))
      plt.barh(top_songs['track_name'], top_songs['streams'], color='lightgreen', edgecolor='black')
      plt.xlabel("Transmissões (Streams)", fontsize=12)
      plt.ylabel("Músicas", fontsize=12)
      plt.title("Top 10 Músicas Mais Transmitidas no Spotify", fontsize=16)
      plt.gca().invert_yaxis()
      plt.grid(axis='x', linestyle='--', alpha=0.7)
      plt.tight_layout()
```

Importando biblioteca pyplot, ordenando dados e definindo gráfico

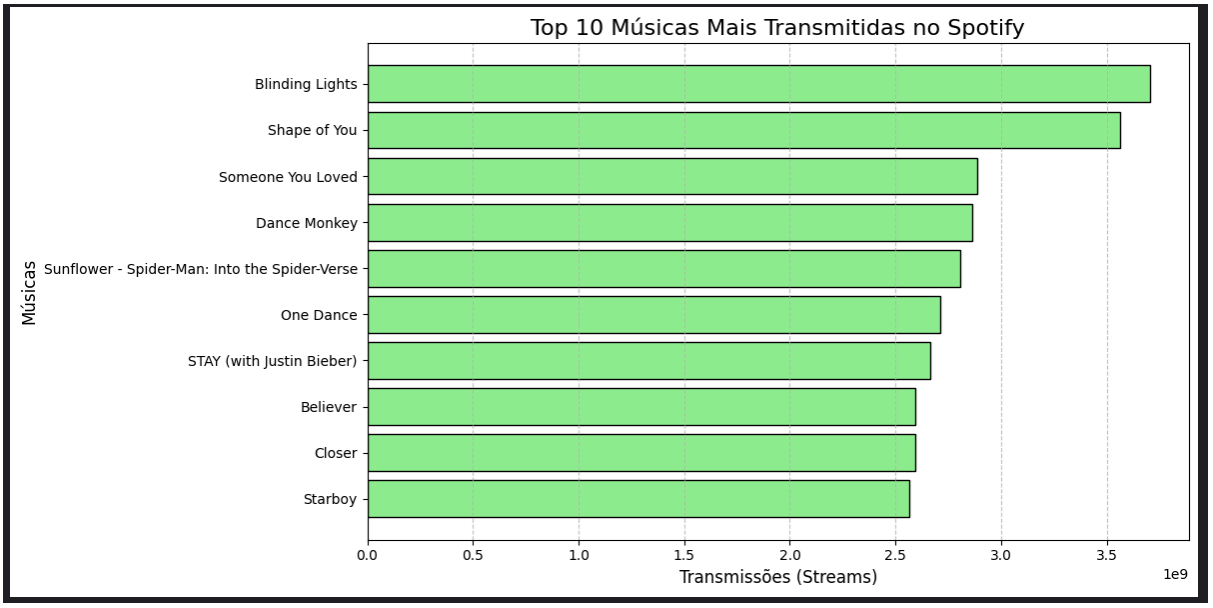


Gráfico resultante do código fonte produzido no exemplo prático Kaggle

4.3. Google Colab

O Google Colab é uma ferramenta gratuita que permite criar e compartilhar notebooks baseados em Jupyter diretamente no navegador. Ele é especialmente útil para projetos de aprendizado de máquina, ciência de dados e programação em Python.

Passo 1: Acessar o site do Google Colab

1. Abra seu navegador de preferência.

2. Na barra de endereços, digite o seguinte link: https://colab.research.google.com/ e pressione **Enter**.

Passo 2: Fazer login com sua conta Google

1. Se você ainda não estiver logado, a página solicitará que insira suas credenciais da conta Google.
2. Digite seu e-mail e senha. (Caso não tenha uma conta Google, você precisará criar uma antes de continuar.)

Passo 3: Interface do Google Colab

Assim que estiver logado, você verá a tela inicial do Google Colab. Aqui estão algumas opções principais:

1. **Criar um novo notebook**:
    - Clique no botão **"Novo Notebook"** (ou "New Notebook", dependendo do idioma).
    - Isso abrirá um notebook em branco onde você pode começar a escrever código Python.
2. **Abrir um notebook existente**:
    - Você pode abrir notebooks de:
        - **Google Drive**: Clique na aba "Google Drive" e escolha um arquivo do seu drive.
        - **GitHub**: Conecte-se ao seu repositório do GitHub para abrir notebooks hospedados lá.
        - **Computador local**: Clique em "Fazer upload" para enviar um arquivo `.ipynb` do seu computador.
3. **Explorar exemplos**:
    - Clique na aba **"Exemplos"** para ver notebooks de demonstração que podem ajudá-lo a começar.

Passo 4: Executar código no Google Colab

1. Dentro do notebook, digite o código Python na célula de entrada.
2. Pressione **Shift + Enter** ou clique no botão de **play** (ao lado da célula) para executar o código.
3. Você verá a saída logo abaixo da célula.

Passo 5: Salvar seu trabalho

1. O Google Colab salva automaticamente os notebooks no seu Google Drive.

2. Para salvar manualmente ou criar uma cópia:
    - Clique em **Arquivo > Salvar uma cópia no Drive**.
    - Você também pode fazer o download como arquivo `.ipynb` ou `.py` para o seu computador.

Dicas adicionais

- **Acesso a GPUs e TPUs**: Para acelerar cálculos, vá em **Editar > Configurações do Notebook** e selecione **GPU** ou **TPU**.
- **Bibliotecas Python**: Use `!pip install` para instalar bibliotecas adicionais diretamente no notebook.

Referências

Datacamp . **Tutorial de Python Seaborn para iniciantes**: Comece a visualizar dados. Datacamp,2024. Disponível em: <https://www.datacamp.com/pt/tutorial/seaborn-python-tutorial>

Datasets/ Kaggle. **Spotify Most Streamed Songs**. Kaggle, 2024. Disponível em: <https://www.kaggle.com/datasets/abdulszz/spotify-most-streamed-songs>

Learn/Microsoft . **Comece a criar com o Power BI** . Microsoft,2024 . Disponível em : <https://learn.microsoft.com/pt-br/training/modules/get-started-with-power-bi/>

Learn/Microsoft . **Curso de Streamlit – Como Criar Apps e Sites com Streamlit**. Microsoft,2024 . Disponível em: <https://www.hashtagtreinamentos.com/streamlit-python#:~:text=O%20Streamlit%20%C3%A9%20uma%20biblioteca,interativos%20e%20visualiza%C3%A7%C3%B5es%20de%20dados/>

Support Microsoft / Microsoft . **O que é o Excel?** . Microsoft, 2024 . Disponível em : <https://support.microsoft.com/pt-br/office/criar-uma-pasta-de-trabalho-no-excel-94b00f50-5896-479c-b0c5-ff74603b35a3>