



MEMBRO DA REDE
ILUMNO

Universidade Veiga de Almeida (UVA)

Ciência da Computação

Análise de Negócios com Uso de Big Data

PROJETO A4 DE ANÁLISE DE NEGOCIOS COM USO DE BIG DATA

Guilherme Suares de Freitas – 1220107152

Rafael de Oliveira Fragoso – 1220300458

Matheus Vasconcellos Novaes- 1220104301

Gabriel Mattos Barroso Serra - 1220107868

Rio de Janeiro

2022

Problemática

Uma startup recém-inaugurada no mercado está prestes a lançar seu jogo inaugural e está atualmente enfrentando uma decisão crucial em relação às plataformas de lançamento. Inicialmente, o jogo foi projetado para plataformas de alto desempenho, como o PlayStation 5, o Xbox Series X e PCs de última geração. No entanto, a empresa está ponderando se deve ou não investir na adaptação do jogo para as plataformas da Nintendo, notando que o console da Nintendo possui um hardware menos potente e exigiria investimentos adicionais significativos para garantir um desempenho adequado.

Para tomar essa decisão estratégica, a empresa decidiu realizar uma análise minuciosa do mercado. Utilizando um Dataset que compreendeu as vendas de jogos até o ano de 2016, e a partir desses dados, conduziu uma filtragem criteriosa e gerou gráficos representativos. Essas análises têm como objetivo fornecer uma visão clara do potencial de vendas e do mercado existente nas plataformas da Nintendo, auxiliando a empresa na determinação da viabilidade de lançar o jogo nesse ambiente específico. A seguir, apresentamos a análise detalhada e as conclusões que embasam a melhor decisão estratégica.

Análise de Dataset e aplicação de ciências de dados com uso de Python e Jupyter

```
In [13]: #Seção de imports
import pandas as pd
import matplotlib.pyplot as plt
```

Importante as bibliotecas Pandas e matplotlib

```
In [8]: #declaração do dataframe
path = r'C:\Trabalhobigdata\Video Games Sales 2016.csv'
```

```
In [9]: #exibição do local do arquivo
path
```

```
Out[9]: 'C:\\Trabalhobigdata\\Video_Games_Sales2016.csv'
```

Declarando o Dataframe, Exibindo o Local onde o Dataframe foi salvo.

```
In [10]: #Inicialização e visualização
dataset = pd.read_csv(path)
display(dataset)
```

[illegible]

Exibindo o DataFrame

```
In [169]: #Demonstração do número de colunas
print(dataset.columns)

Index(['Name', 'Platform', 'Year_of_Release', 'Genre', 'Publisher', 'NA_Sales',
       'EU_Sales', 'JP_Sales', 'Other_Sales', 'Global_Sales', 'Critic_Score',
       'Critic_Count', 'User_Score', 'User_Count', 'Developer', 'Rating'],
      dtype='object')
```

Mostrando as colunas para melhor visualização delas

```
In [170]: #Agrupar os dados por empresa e somar as vendas
vendas_por_empresa = dataset.groupby('Publisher')['Global_Sales'].sum().reset_index()

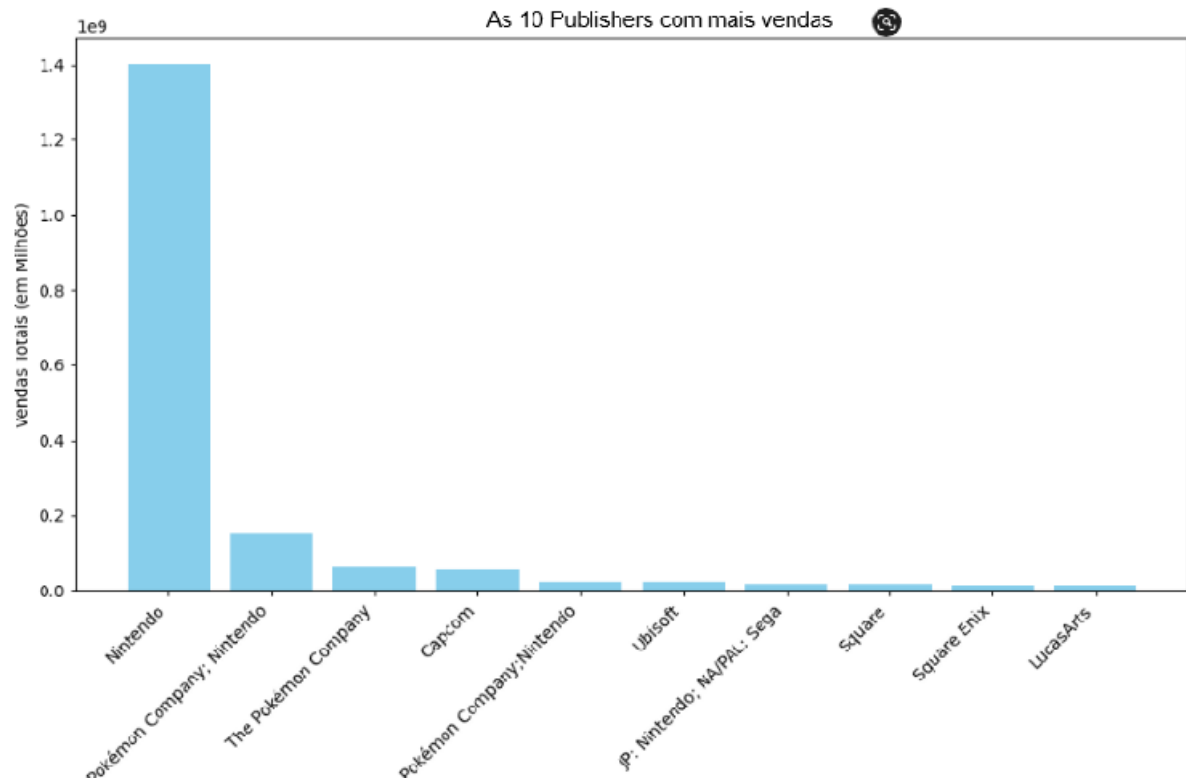
In [171]: #Ordene as empresas por vendas em ordem decrescente
vendas_por_empresa = vendas_por_empresa.sort_values(by='Global_Sales', ascending=False)

In [172]: #As 10 empresas que mais venderam
top_10_empresas = vendas_por_empresa.head(10)
```

Agrupando os dados por empresas e somando vendas, ordenando as empresas por vendas em ordem decrescente e mostrando as 10 empresas que mais venderam nesse período

```
#criação do gráfico de barras
plt.figure(figsize=(12, 6))
plt.bar(top_10_empresas['Publisher'], top_10_empresas['Sales'], color='skyblue')
plt.xlabel('Empresa')
plt.ylabel('Vendas Totais (em Milhões)')
plt.title('As 10 Empresas Mais Vendidas')
plt.xticks(rotation=45, ha='right')
```

```
([0, 1, 2, 3, 4, 5, 6, 7, 8, 9],
 [Text(0, 0, 'Nintendo'),
  Text(1, 0, 'The Pokémon Company; Nintendo'),
  Text(2, 0, 'The Pokémon Company'),
  Text(3, 0, 'Capcom'),
  Text(4, 0, 'The Pokémon Company; Nintendo'),
  Text(5, 0, 'Ubisoft'),
  Text(6, 0, 'JP: Nintendo; NA/PAL: Sega'),
  Text(7, 0, 'Square'),
  Text(8, 0, 'Square Enix'),
  Text(9, 0, 'LucasArts')])
```



Apresentação do gráfico das Publishers com mais vendas

```

In [*]: #Filtragem de dados
print("Escolha um dado para ser filtrado em razão do número de vendas:")
metrica = input("1-Game\n2-Developer\n3-Publisher\n4-Release date\n5-Platform\n")

if metrica == '1':
    vendas_por_jogo = dataset.groupby('Name')['Global_Sales'].sum().reset_index()
    vendas_por_jogo = vendas_por_jogo.sort_values(by='Global_Sales', ascending=False)
    print(vendas_por_jogo)

elif metrica == '2':
    vendas_por_desenvolvedor = dataset.groupby('Developer')['Global_Sales'].sum().reset_index()
    vendas_por_desenvolvedor = vendas_por_desenvolvedor.sort_values(by='Global_Sales', ascending=False)
    print(vendas_por_desenvolvedor)

elif metrica == '3':
    vendas_por_distribuidora = dataset.groupby('Publisher')['Global_Sales'].sum().reset_index()
    vendas_por_distribuidora = vendas_por_distribuidora.sort_values(by='Global_Sales', ascending=False)
    print(vendas_por_distribuidora)

elif metrica == '4':
    vendas_por_data = dataset.groupby('Year_of_Release')['Global_Sales'].sum().reset_index()
    vendas_por_data = vendas_por_data.sort_values(by='Global_Sales', ascending=False)
    print(vendas_por_data)

elif metrica == '5':
    vendas_por_plataforma = dataset.groupby('Platform')['Global_Sales'].sum().reset_index()
    vendas_por_plataforma = vendas_por_plataforma.sort_values(by='Global_Sales', ascending=False)
    print(vendas_por_plataforma)

else:
    print("Escolha inválida!")

1-Game
2-Developer
3-Publisher
4-Release date
5-Platform


Escolha um dado para ser filtrado em razão do número de vendas:

```

Filtragem de dados com um menu para que se possa escolher qual informação você quer obter sendo elas Jogo mais vendido, Desenvolvedora com mais vendas, Publisher com mais vendas, Data em que teve o jogo com mais vendas e a plataforma com mais venda.

```

In [*]: #Cálculo dos índices
print("Escolha um dado para ter seus indicadores calculados em razão do número de vendas:")
metrica = input("1-Game\n2-Developer\n3-Publisher\n4-Release date\n5-Platform\n")

if metrica == '1':
    media = dataset.groupby('Name')['Global_Sales'].mean()
    print("Média:", media)

    mediana = dataset.groupby('Name')['Global_Sales'].median()
    print("Mediana:", mediana)

    contagem = dataset.groupby('Name')['Global_Sales'].count()
    print("Contagem de elementos:", contagem)

elif metrica == '2':
    media = dataset.groupby('Developer')['Global_Sales'].mean()
    print("Média:", media)

    mediana = dataset.groupby('Developer')['Global_Sales'].median()
    print("Mediana:", mediana)

    contagem = dataset['Developer'].count()
    print("Contagem de elementos:", contagem)

elif metrica == '3':
    media = dataset.groupby('Publisher')['Global_Sales'].mean()
    print("Média:", media)

    mediana = dataset.groupby('Publisher')['Global_Sales'].median()
    print("Mediana:", mediana)

    contagem = dataset['Publisher'].count()
    print("Contagem de elementos:", contagem)

elif metrica == '4':
    media = dataset.groupby('Year_of_Release')['Global_Sales'].mean()
    print("Média:", media)

    mediana = dataset.groupby('Year_of_Release')['Global_Sales'].median()
    print("Mediana:", mediana)

    contagem = dataset['Year_of_Release'].count()
    print("Contagem de elementos:", contagem)
elif metrica == '5':
    media = dataset.groupby('Platform')['Global_Sales'].mean()
    print("Média:", media)

    mediana = dataset.groupby('Platform')['Global_Sales'].median()
    print("Mediana:", mediana)

    contagem = dataset['Platform'].count()
    print("Contagem de elementos:", contagem)
else:
    print("Escolha inválida!")

1-Game
2-Developer
3-Publisher
4-Release date
5-Platform

```

Cálculo de índices com menu

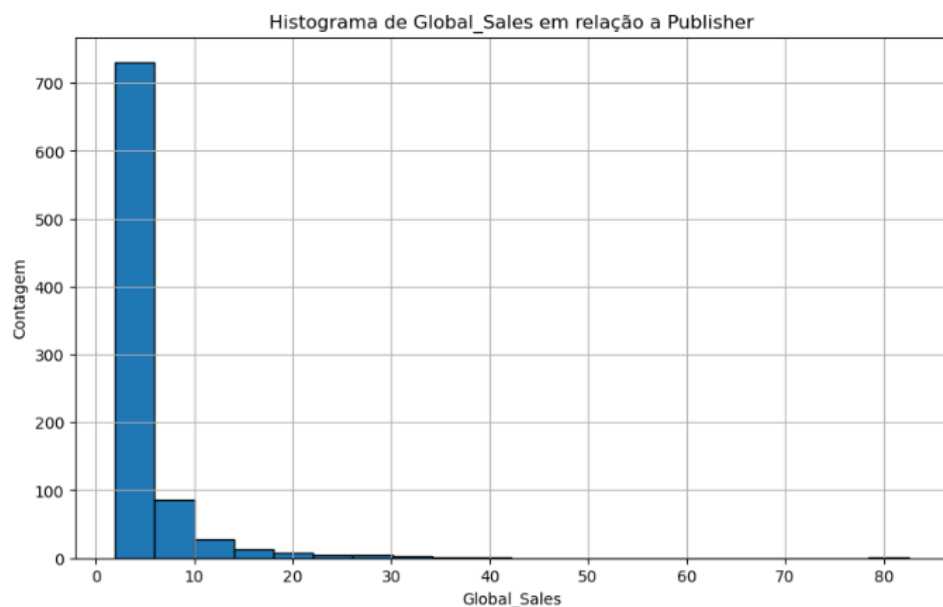
```
In [208]: #Análise da distribuição das amostras
elemento_x = 'Publisher'
elemento_y = 'Global_Sales'

dados_filtrados = dataset[[elemento_x, elemento_y]]

plt.figure(figsize=(10, 6))
plt.hist(dados_filtrados[elemento_y], bins=20, edgecolor='k')
plt.xlabel(elemento_y)
plt.ylabel('Contagem')
plt.title(f'Histograma de {elemento_y} em relação a {elemento_x}')
plt.grid(True)
plt.show()

vendas_por_publisher = dataset.groupby('Publisher')['Global_Sales'].sum()
publisher_mais_vendida = vendas_por_publisher.idxmax()
vendas_maximas = vendas_por_publisher.max()

print(f'A publisher com o maior número de vendas é "{publisher_mais_vendida}" com um total de vendas de {vendas_maximas} unidades
```



A publisher com o maior número de vendas é "Nintendo" com um total de vendas de 1470.4 unidades.

Análise da distribuição das amostras

Conclusão

Como solução para essa problemática, a startup implementou um código em Python, no ambiente Jupyter Notebook, que aproveitou o Dataset de vendas de jogos até 2016. Este código realizou uma filtragem criteriosa dos dados, identificando as 10 Publishers que mais obtiveram vendas, ou seja, as "best sellers" no mercado de jogos.

Essa abordagem permitiu à empresa obter insights valiosos sobre as tendências de vendas e o desempenho das principais Publishers nas diferentes plataformas. Com base nesses resultados, a startup está em uma posição melhor para tomar uma decisão informada sobre se deve ou não adaptar seu jogo para as plataformas da Nintendo, por exemplo.

Essa análise de "best sellers" permite a publisher avaliar o potencial de sucesso do jogo na plataforma da Nintendo, levando em consideração o histórico de vendas das principais empresas nesse ambiente específico. Dessa forma, a startup pode tomar uma decisão estratégica embasada em dados concretos e maximizar suas chances de sucesso no mercado de jogos.

Demandas da avaliação

Utilizando a biblioteca pandas - crie o dataframe correspondente:

```
In [27]: #Seção de imports
import pandas as pd
import matplotlib.pyplot as plt
```

```
In [28]: #declaração do dataframe
path = r'C:\Trabalhobigdata\Video_Games_Sales2016.csv'
```

```
In [29]: #exibição do local do arquivo
path
```

```
Out[29]: 'C:\\Trabalhobigdata\\Video_Games_Sales2016.csv'
```

```
In [30]: #Inicialização e visualização
dataset = pd.read_csv(path)
display(dataset)
```

	Name	Platform	Year_of_Release	Genre	Publisher	NA_Sales	EU_Sales	JP_Sales	Other_Sales	Global_Sales	Critic_Score	Critic_Count	User_Score
0	Wii Sports	Wii	2006.0	Sports	Nintendo	41.36	28.96	3.77	8.45	82.63	76.0	51.0	8.0
1	Super Mario Bros.	NES	1985.0	Platform	Nintendo	29.08	3.58	6.81	0.77	40.24	NaN	NaN	NaN
2	Mario Kart Wii	Wii	2008.0	Racing	Nintendo	15.68	12.76	3.79	3.29	35.52	82.0	73.0	8.0
3	Wii Sports Resort	Wii	2009.0	Sports	Nintendo	15.61	10.93	3.28	2.95	32.77	80.0	73.0	8.0
4	Pokemon Red/Pokemon Blue	GB	1996.0	Role-Playing	Nintendo	11.27	8.89	10.22	1.00	31.37	NaN	NaN	NaN
...
874	Resident Evil 6	X360	2012.0	Shooter	Capcom	1.12	0.60	0.07	0.16	1.95	67.0	71.0	6.0
	Grand Theft Auto V	PlayStation 4	2013.0	Action	Rockstar Games	15.00	10.00	0.00	0.00	25.00	80.0	70.0	8.0

```
In [31]: #Demonstração do número de colunas
print(dataset.columns)

Index(['Name', 'Platform', 'Year_of_Release', 'Genre', 'Publisher', 'NA_Sales',
      'EU_Sales', 'JP_Sales', 'Other_Sales', 'Global_Sales', 'Critic_Score',
      'Critic_Count', 'User_Score', 'User_Count', 'Developer', 'Rating'],
      dtype='object')
```

Bibliotecas, Dataframe e Colunas para melhor manipulação dos dados.

Manipulação de colunas (inserção e/ou deleção e/ou filtragem);

```
In [40]: #Filtragem de dados
print("Escolha um dado para ser filtrado em razão do número de vendas:")
metrica = input("1-Game\n2-Developer\n3-Publisher\n4-Release date\n5-Platform\n")

if metrica == '1':
    vendas_por_jogo = dataset.groupby('Name')['Global_Sales'].sum().reset_index()
    vendas_por_jogo = vendas_por_jogo.sort_values(by='Global_Sales', ascending=False)
    print(vendas_por_jogo)

elif metrica == '2':
    vendas_por_desenvolvedor = dataset.groupby('Developer')['Global_Sales'].sum().reset_index()
    vendas_por_desenvolvedor = vendas_por_desenvolvedor.sort_values(by='Global_Sales', ascending=False)
    print(vendas_por_desenvolvedor)

elif metrica == '3':
    vendas_por_distribuidora = dataset.groupby('Publisher')['Global_Sales'].sum().reset_index()
    vendas_por_distribuidora = vendas_por_distribuidora.sort_values(by='Global_Sales', ascending=False)
    print(vendas_por_distribuidora)

elif metrica == '4':
    vendas_por_data = dataset.groupby('Year_of_Release')['Global_Sales'].sum().reset_index()
    vendas_por_data = vendas_por_data.sort_values(by='Global_Sales', ascending=False)
    print(vendas_por_data)

elif metrica == '5':
    vendas_por_plataforma = dataset.groupby('Platform')['Global_Sales'].sum().reset_index()
    vendas_por_plataforma = vendas_por_plataforma.sort_values(by='Global_Sales', ascending=False)
    print(vendas_por_plataforma)

else:
    print("Escolha inválida!")
```

Escolha um dado para ser filtrado em razão do número de vendas:

1-Game
2-Developer
3-Publisher
4-Release date
5-Platform
4

	Year_of_Release	Global_Sales
29	2009.0	290.83
26	2006.0	285.25
30	2010.0	278.03
27	2007.0	273.01
28	2008.0	246.22
31	2011.0	230.74
33	2013.0	190.66
25	2005.0	187.02
34	2014.0	167.50
22	2002.0	162.54
32	2012.0	160.69
21	2001.0	159.38
24	2004.0	151.20
35	2015.0	131.28
23	2003.0	116.50
19	1999.0	113.08
16	1996.0	112.38
18	1998.0	112.01
17	1997.0	92.01
20	2000.0	77.53

Exemplo Utilizando a filtragem por data de lançamento

Análise da distribuição das amostras ou itens (sendo a estatística à sua escolha):

```
#Análise da distribuição das amostras
elemento_x = 'Publisher'
elemento_y = 'Global_Sales'

dados_filtrados = dataset[[elemento_x, elemento_y]]

plt.figure(figsize=(10, 6))
plt.hist(dados_filtrados[elemento_y], bins=20, edgecolor='k')
plt.xlabel(elemento_y)
plt.ylabel('Contagem')
plt.title(f'Histograma de {elemento_y} em relação a {elemento_x}')
plt.grid(True)
plt.show()

vendas_por_publisher = dataset.groupby('Publisher')['Global_Sales'].sum()
publisher_mais_vendida = vendas_por_publisher.idxmax()
vendas_maximas = vendas_por_publisher.max()

print(f'A publisher com o maior número de vendas é "{publisher_mais_vendida}" com um total de vendas de {vendas_maximas} unidades
```



A publisher com o maior número de vendas é "Nintendo" com um total de vendas de 1470.4 unidades.

Mostrando um gráfico em que a Publisher com maior número de vendas foi a Nintendo

Link do original do dataset: <https://www.kaggle.com/datasets/sidtwr/videogames-sales-dataset/code>