Trabalho 3 - Mineração de Dados - Grupo 8 - Luan e Lucas

Junto ao pdf com o relatorio mandaremos o código e esse link para o colab: "https://colab.research.google.com/drive/1IJ5TCewguioyXlwTivOF71wtJ6N6m2Wz?usp=sharing"

Devido às falhas do colab em gerar o pdf que demonstra todo o código e com intuito de dar maior liberdade e informações pedimos que olhe ambos.

Importações de bibliotecas externas

```
#Importações e Drive

from google.colab import drive
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
import numpy as np
import sklearn.preprocessing as skp


drive.mount('/content/drive')
```

```
Drive already mounted at /content/drive; to attempt to forcibly remount, call
```

Importação dos dados e observação inicial

```
data = pd.read_csv("/content/drive/MyDrive/Mineracao-de-Dados/trabalho2_dados_8.csv
data.head()
```

	nome	plataforma	genero	editora	vendas	lancamento	avaliacao-criticos	c
0	Bladestorm: The Hundred Years' War	X360	Action	Tecmo Koei	0.09	6-Nov-07	63.0	
1	Sudoku Ball Detective	Wii	Puzzle	Playlogic Game Factory	0.03	13-Oct-09	NaN	
2	Family Game Night 4: The Game Show	Wii	Misc	Electronic Arts	0.12	1-Nov-11	NaN	
3	Rayman Origins	3DS	Platform	Ubisoft	0.08	6-Nov-12	71.0	

```
data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 3203 entries, 0 to 1570
Data columns (total 11 columns):
 #   Column              Non-Null Count  Dtype  
---  ------              --------------  -----  
 0   nome                3203 non-null   object 
 1   plataforma          3203 non-null   object 
 2   genero              3203 non-null   object 
 3   editora             3197 non-null   object 
 4   vendas              3203 non-null   float64
 5   lancamento          2461 non-null   object 
 6   avaliacao-criticos  2023 non-null   float64
 7   numero-criticos     2023 non-null   float64
 8   avaliacao-usuarios  2422 non-null   object 
 9   numero-usuarios     1906 non-null   float64
 10  fabricante          2451 non-null   object 
dtypes: float64(4), object(7)
memory usage: 300.3+ KB
```

Tratamento dos Dados

One-Hot-Encoding da Plataforma e do Gênero

Dado que diversos modelos de Aprendizado de Máquina não suportam dados categóricos, foi usada a técnica “One-Hot Encoding” para colocá-los em um formato numérico que esses modelos aceitem.

```
pd.get_dummies(data.plataforma, prefix='Plataforma').head()
```

	Plataforma_3DS	Plataforma_PS3	Plataforma_PS4	Plataforma_PSP	Plataforma_
0	0	0	0	0	
1	0	0	0	0	
2	0	0	0	0	
3	1	0	0	0	
4	0	0	0	1	

```
pd.get_dummies(data.genero, prefix='Genero').head()
```

	Genero_Action	Genero_Adventure	Genero_Fighting	Genero_Misc	Genero_Platf
0	1	0	0	0	
1	0	0	0	0	
2	0	0	0	1	

```
data_oh_enc = pd.concat([data, pd.get_dummies(data.plataforma, prefix='plataforma'
data_oh_enc.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 3203 entries, 0 to 1570
Data columns (total 30 columns):
 #   Column               Non-Null Count  Dtype
---  ------               --------------  -----
 0   nome                 3203 non-null   object
 1   editora              3197 non-null   object
 2   vendas               3203 non-null   float64
 3   lancamento           2461 non-null   object
 4   avaliacao-criticos   2023 non-null   float64
 5   numero-criticos      2023 non-null   float64
 6   avaliacao-usuarios   2422 non-null   object
 7   numero-usuarios      1906 non-null   float64
 8   fabricante           2451 non-null   object
 9   plataforma_3DS       3203 non-null   uint8
 10  plataforma_PS3       3203 non-null   uint8
 11  plataforma_PS4       3203 non-null   uint8
 12  plataforma_PSP       3203 non-null   uint8
 13  plataforma_PSV       3203 non-null   uint8
 14  plataforma_Wii       3203 non-null   uint8
 15  plataforma_WiiU      3203 non-null   uint8
 16  plataforma_X360      3203 non-null   uint8
 17  plataforma_XOne      3203 non-null   uint8
 18  genero_Action        3203 non-null   uint8
 19  genero_Adventure     3203 non-null   uint8
 20  genero_Fighting      3203 non-null   uint8
 21  genero_Misc          3203 non-null   uint8
 22  genero_Platform      3203 non-null   uint8
 23  genero_Puzzle        3203 non-null   uint8
 24  genero_Racing        3203 non-null   uint8
 25  genero_Role-Playing  3203 non-null   uint8
 26  genero_Shooter       3203 non-null   uint8
 27  genero_Simulation    3203 non-null   uint8
 28  genero_Sports        3203 non-null   uint8
 29  genero_Strategy      3203 non-null   uint8
dtypes: float64(4), object(5), uint8(21)
memory usage: 315.9+ KB
```

Tratamento de Inconsistências e transformações de tipo nas Avaliações e datas de Lançamento

```
data_oh_enc.lancamento = pd.to_datetime(data_oh_enc.lancamento, errors="coerce")
```

```
data_oh_enc.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 3203 entries, 0 to 1570
Data columns (total 30 columns):
 #   Column               Non-Null Count  Dtype         
---  ------               --------------  -----         
 0   nome                 3203 non-null   object        
 1   editora              3197 non-null   object        
 2   vendas               3203 non-null   float64       
 3   lancamento           2431 non-null   datetime64[ns]
 4   avaliacao-criticos   2023 non-null   float64       
 5   numero-criticos      2023 non-null   float64       
 6   avaliacao-usuarios   2422 non-null   object        
 7   numero-usuarios      1906 non-null   float64       
 8   fabricante           2451 non-null   object        
 9   plataforma_3DS       3203 non-null   uint8         
 10  plataforma_PS3       3203 non-null   uint8         
 11  plataforma_PS4       3203 non-null   uint8         
 12  plataforma_PSP       3203 non-null   uint8         
 13  plataforma_PSV       3203 non-null   uint8         
 14  plataforma_Wii       3203 non-null   uint8         
 15  plataforma_WiiU      3203 non-null   uint8         
 16  plataforma_X360      3203 non-null   uint8         
 17  plataforma_XOne      3203 non-null   uint8         
 18  genero_Action        3203 non-null   uint8         
 19  genero_Adventure     3203 non-null   uint8         
 20  genero_Fighting      3203 non-null   uint8         
 21  genero_Misc          3203 non-null   uint8         
 22  genero_Platform      3203 non-null   uint8         
 23  genero_Puzzle        3203 non-null   uint8         
 24  genero_Racing        3203 non-null   uint8         
 25  genero_Role-Playing  3203 non-null   uint8         
 26  genero_Shooter       3203 non-null   uint8         
 27  genero_Simulation    3203 non-null   uint8         
 28  genero_Sports        3203 non-null   uint8         
 29  genero_Strategy      3203 non-null   uint8         
dtypes: datetime64[ns](1), float64(4), object(4), uint8(21)
memory usage: 315.9+ KB
```

```
data_oh_enc.columns.to_list()
```

```
['nome',
 'editora',
 'vendas',
 'lancamento',
 'avaliacao-criticos',
 'numero-criticos',
 'avaliacao-usuarios',
 'numero-usuarios',
 'fabricante',
 'plataforma_3DS',
 'plataforma_PS3',
 'plataforma_PS4',
 'plataforma_PSP',
 'plataforma_PSV',
 'plataforma_Wii',
 'plataforma_WiiU',
 'plataforma_X360',
 'plataforma_XOne',
 'genero_Action',
```

```
       'genero_Adventure',
       'genero_Fighting',
       'genero_Misc',
       'genero_Platform',
       'genero_Puzzle',
       'genero_Racing',
       'genero_Role-Playing',
       'genero_Shooter',
       'genero_Simulation',
       'genero_Sports',
       'genero_Strategy']
```

Como podemos perceber a avaliação dos usuários não está no tipo: “float”

```
data_oh_enc['avaliacao-usuarios'].unique()
```

```
    array(['7.7', 'tbd', '6.1', nan, '7.3', '2.8', '8.3', '5.7', '6.7', '6.2',
           '8.7', '6.8', '4', '7.2', '7.5', '8.6', '8.8', '7', '7.6', '7.4',
           '8.2', '8.1', '7.1', '6.9', '8.4', '6.5', '5', '8', '5.8', '7.8',
           '6.4', '5.3', '4.9', '5.9', '6.6', '3.1', '4.3', '2.2', '5.2',
           '8.5', '6.3', '4.6', '4.5', '3.9', '7.9', '5.4', '4.8', '4.4',
           '2.7', '2.1', '5.5', '9.2', '6', '2.5', '2', '8.9', '1.4', '4.7',
           '2.4', '2.9', '4.2', '3.3', '3.7', '9', '5.1', '9.1', '3.2', '3.5',
           '4.1', '3.8', '0.8', '3', '1.3', '2.3', '5.6', '1.5', '3.6', '1',
           '1.8', '3.4', '0.7', '1.6', '0.9'], dtype=object)
```

Além disso, todos os dados podem ser transformados em float com exceção de "tbd", assim com intuito de transformar todos os valores em float, transformaremos os "tbd" em NaN(not a number).

```
data_oh_enc["avaliacao-usuarios"] = data_oh_enc["avaliacao-usuarios"].replace('tbd
```

Unicidade de Nome

Como existem múltiplos registros para o mesmo jogo, dado que cada plataforma tem sua própria versão, e é feita uma contagem individual de cada versão, iremos unir esses registros. Porém trataremos cada grupo de colunas de forma particular.

Faremos isso dividido o DataFrame em vários e tratando individualmente os casos de cada grupo de colunas. E seguindo os seguintes parâmetros :

As colunas "Nome", "Editora", "Lançamento" e "Fabricante" normalmente não mudam de plataforma em plataforma, portanto para cada grupo de instâncias com o mesmo nome, manteremos a primeira ocorrência não nula destes.

```
data_primeiro_editora = data_oh_enc[["nome","editora"]].sort_values("editora", asc
data_primeiro_lancamento = data_oh_enc[["nome","lancamento"]].sort_values("lancame
data_primeiro_fabricante = data_oh_enc[["nome","fabricante"]].sort_values("fabrica
```

```
data_primeiro = pd.concat([data_primeiro_editora, data_primeiro_lancamento, data_p
```

As colunas OneHot e vendas serão somadas em todas as instâncias, pois assim poderemos manter as informações sobre quais plataformas um jogo foi publicado e o número total de vendas do jogo em todas as plataformas.

```
data_sum = data_oh_enc[["nome","vendas", 'plataforma_3DS', 'plataforma_PS3', 'plata
 'plataforma_WiiU', 'plataforma_X360', 'plataforma_XOne', 'genero_Action', 'genero_
 'genero_Puzzle',  'genero_Racing',  'genero_Role-Playing',  'genero_Shooter',  'g
```

Para as colunas de avaliação, será feita uma média simples das avaliações, valores "NaN" serão ignorados.

```
data_avg = data_oh_enc[["nome", "avaliacao-criticos","avaliacao-usuarios"]].groupby
```

```
data_avg.head()
```

nome	avaliacao-criticos	avaliacao-usuarios
.hack: Sekai no Mukou ni + Versus	NaN	NaN
101-in-1 Sports Party Megamix	41.000000	NaN
11eyes: CrossOver	NaN	NaN
2010 FIFA World Cup South Africa	76.166667	7.6
3D Dot Game Heroes	77.000000	7.9

```
data_unified = pd.concat([data_primeiro, data_sum, data_avg], axis=1).reset_index(
data_unified.head()
```

	nome	editora	lancamento	fabricante	vendas	plataforma_3DS	plataf
0	.hack: Sekai no Mukou ni + Versus	Namco Bandai Games	NaT	None	0.03	0	

Lidando com os NaNs

Megamix

Agora, para lidar com “NaNs”, para os valores de "Vendas", "Avaliação-críticos" e "Avaliação-usuários":

Usamos as médias para preencher os “NaNs”. Para os usuários que avaliaram, julgamos que como não havia avaliações seria mais justo colocar 1 nos NaN, como se essa avaliação da média viesse de uma pessoa só.

3D Dot SouthPeak

```
mean_rows = data_unified[['vendas','avaliacao-criticos','avaliacao-usuarios']].mean
data_unified[['vendas','avaliacao-criticos','avaliacao-usuarios']].fillna(mean_rows
data_unified[['numero-criticos','numero-usuarios']].fillna(1, inplace=True)
```

```
/usr/local/lib/python3.6/dist-packages/pandas/core/series.py:4536: SettingWit
A value is trying to be set on a copy of a slice from a DataFrame

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/s
  downcast=downcast,
/usr/local/lib/python3.6/dist-packages/pandas/core/frame.py:4327: SettingWith
A value is trying to be set on a copy of a slice from a DataFrame

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/s
  downcast=downcast,
```

Para as datas de lançamento, editora e fabricante, optamos manter as colunas mesmo com os “NaNs” e ,caso os dados sejam utilizados, não iremos usar as instâncias que os têm como “NaN”.

Padronização de Valores

No fim, iremos padronizar as colunas de valores do tipo “Float”

```
AvUserScaler = skp.StandardScaler()
AvCritScaler = skp.StandardScaler()
VendasScaler = skp.MinMaxScaler()
```

```
data_unified['avaliacao-usuarios'] = AvUserScaler.fit_transform(data_unified['aval
data_unified['avaliacao-criticos'] = AvCritScaler.fit_transform(data_unified['aval
data_unified['vendas'] = VendasScaler.fit_transform(data_unified['vendas'].values.
```

```
data_unified['avaliacao-criticos'].plot.hist(bins=16)
```

```
<matplotlib.axes._subplots.AxesSubplot at 0x7fcb78a4b4e0>
```