05

Mãos à obra!

Vamos começar visualizando os centróides de cada grupo. Para isso, podemos falar para o Python imprimir os centróides e os nomes dos gêneros - que são as colunas do data frame `generos` :

```
print(generos.columns)
print(modelo.cluster_centers_)
```

```
Index(['(no genres listed)', 'Action', 'Adventure', 'Animation', 'Children',
       'Comedy', 'Crime', 'Documentary', 'Drama', 'Fantasy', 'Film-Noir',
       'Horror', 'IMAX', 'Musical', 'Mystery', 'Romance', 'Sci-Fi', 'Thriller',
       'War', 'Western'],
      dtype='object')
[[-0.05917995  1.77945047  0.55553895 -0.16381415 -0.26630279 -0.36360255
   0.30161844 -0.20993782 -0.28631303  0.07714909 -0.06158693  0.01905363
   0.32268722 -0.17405476 -0.01003755 -0.29857304  0.74304763  0.56544578
   0.15078538  0.05721408]
 [ 0.02425436 -0.47949498 -0.2730736  -0.25211343 -0.27045122  0.06715779
  -0.03984436  0.08495479  0.15795732 -0.139547    0.02967474  0.03096649
  -0.11464877 -0.00628803  0.02003386  0.11884858 -0.21319196 -0.09413105
  -0.01935    -0.00496482]
 [-0.05917995 -0.0987796   0.8968363   2.23664878  2.5878729   0.2519166
  -0.32826065 -0.20176991 -0.59169556  0.88866543 -0.09492563 -0.27250055
   0.19033986  0.40959419 -0.12974119 -0.27211478  0.0562637  -0.46926249
  -0.16838295 -0.0817667 ]]
```

Para facilitar o trabalho e a manipulação desses dados, vamos criar um data frame chamado `grupos` a partir dos centróides.

Portanto, falamos para o pandas (`pd`) criar um `DataFrame` a partir dos centróides e nomear as colunas (`columns`) com o nome dos gêneros:

```
grupos = pd.DataFrame(modelo.cluster_centers_,
            columns=generos.columns)
```

Podemos ver o data frame colocando a variável grupos como a última instrução da célula.

Vamos visualizar os centróides transpondo (`transpose`) o data frame de grupos e pedindo para o `pandas` plotar (`plot`) um gráfico de barras (`bar`). Como queremos que cada cluster tenha seu próprio gráfico, vamos falar que teremos `subplots` e para facilitar a visualização, vamos definir um tamanho para a figura (`figsize`) e dizer que não queremos compartilhar os labels do eixo `x` :

```
grupos.transpose().plot.bar(subplots=True,
               figsize=(25, 25),
               sharex=False)
```

Podemos visualizar os filmes pertencentes a algum grupo, por exemplo o grupo `0` . Basta realizar um filtro pelos `labels_` do modelo e pedir alguns dados da amostra:

```
grupo = 0

filtro = modelo.labels_ == grupo

dados_dos_filmes[filtro].sample(10)
```

Vamos plotar um gráfico de pontos. Porém, temos 20 gêneros, ou seja, 20 dimensões. Logo, antes de plotar o gráfico, temos que reduzir as dimensões. Para isso, vamos utilizar o algoritmo `TNSE` do módulo `manifold` da `sklearn`.

```
from sklearn.manifold import TSNE
```

A partir desse algoritmo podemos criar um objeto `TSNE` e utilizar o método `fit_transform`. Este método nos retorna um array do `numpy` com as features reduzidas.

```
tsne = TSNE()

visualizacao = tsne.fit_transform(generos_escalados)
```

Agora basta importamos o `seaborn` e plotar um gráfico de dispersão (`scatterplot`).

Mas antes, vamos atribuir um valor para o tamanho da figura (`figure.figsize`), apenas para facilitar a visualização:

```
import seaborn as sns

sns.set(rc={'figure.figsize': (13, 13)})



sns.scatterplot(x=visualizacao[:, 0],
                y=visualizacao[:, 1],
                hue=modelo.labels_,
                palette=sns.color_palette('Set1', 3))
```