Projeto da Disciplina - Algoritmos de Inteligência Artificial para clusterização

***As partes 1,2,3 se encontram no notebook no repositorío:**
https://github.com/Annallisboa/clusteringpos

Parte 4

1. **Escreva em tópicos as etapas do algoritmo de K-médias até sua convergência.**
    1. Escolher o número de clusters e iniciar os centróides
    2. Atribuir cada ponto nos dados ao centróide mais próximo através da distância formando os clusters;
    3. Repetir os passos anteriores e recalcular o centróide com novos atributos;
    4. Para o algoritmo parar algumas condições precisam ser atendidas, como: os centróides não mudar, o número de interações máximo é atingido e as atribuições do cluster não mudam.

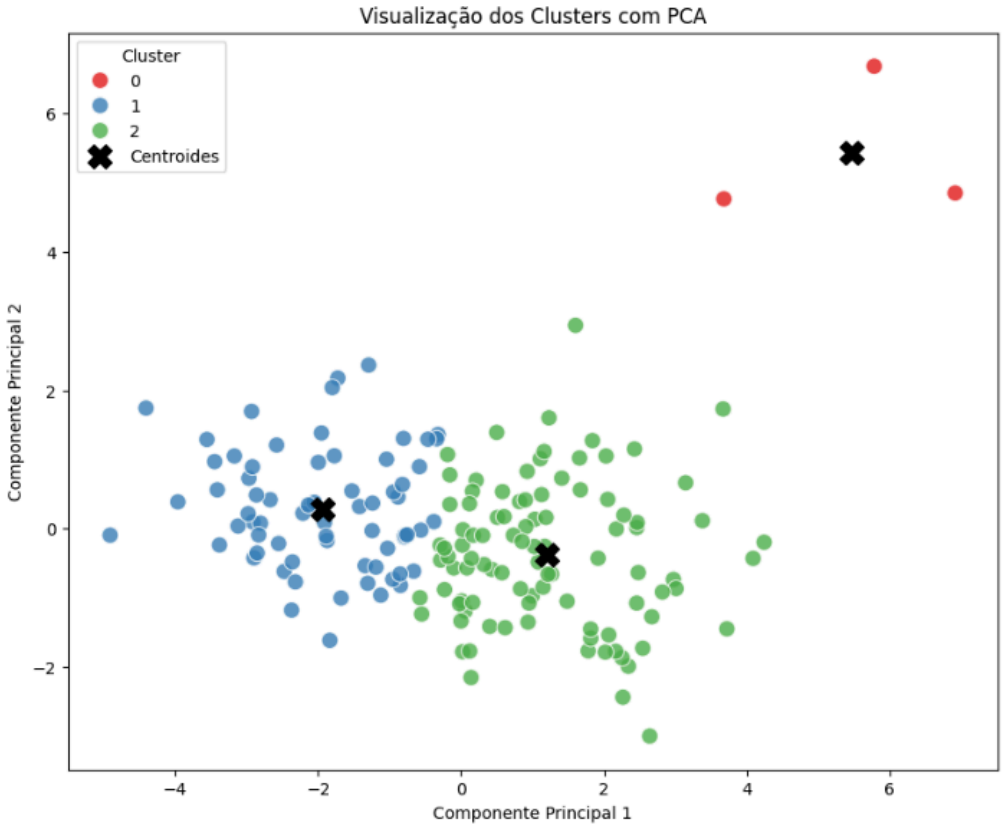
2. O algoritmo de K-médias converge até encontrar os centróides que melhor descrevem os clusters encontrados (até o deslocamento entre as interações dos centróides ser mínimo). Lembrando que o centróide é o baricentro do cluster em questão e não representa, em via de regra, um dado existente na base. Refaça o algoritmo apresentado na questão 1 a fim de garantir que o cluster seja representado pelo dado mais próximo ao seu baricentro em todas as iterações do algoritmo.
Obs: nesse novo algoritmo, o dado escolhido será chamado medoids.
Não consegui rodar o algoritmo:

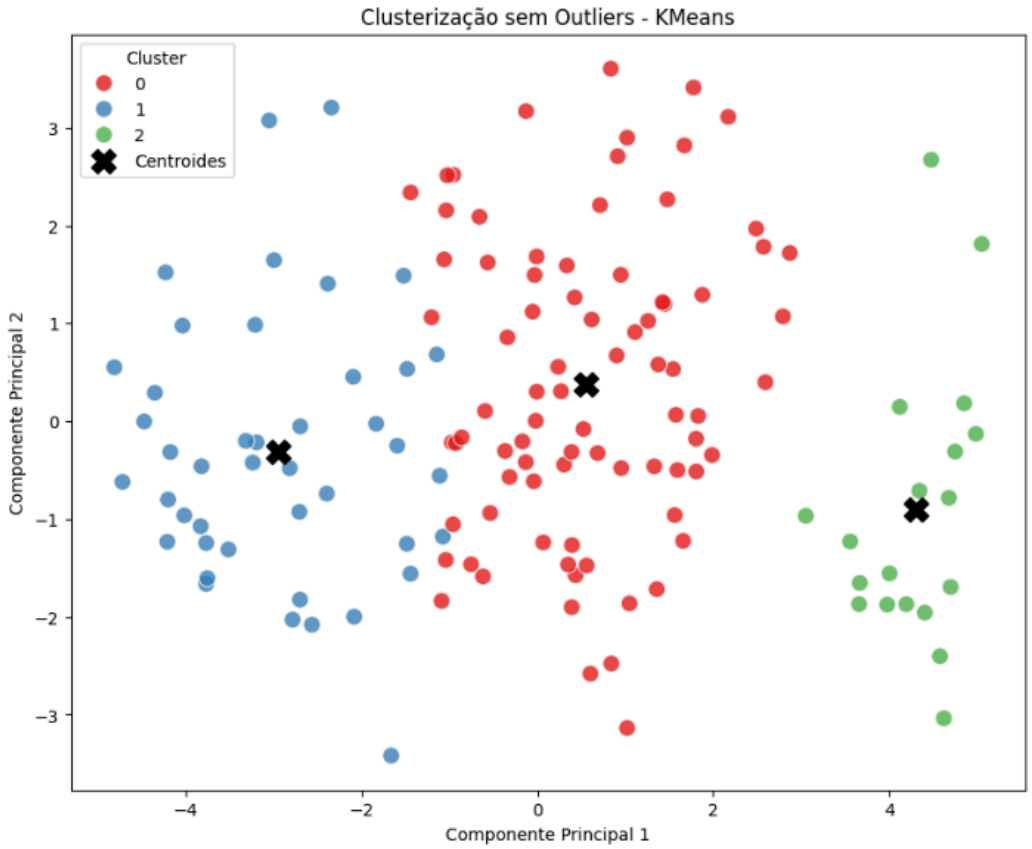
```
from sklearn_extra.cluster import KMedoids
from sklearn.metrics import pairwise_distances_argmin_min
```

```
---------------------------------------------------------------------------
ValueError                                Traceback (most recent call last)
Cell In[98], line 1
----> 1 from sklearn_extra.cluster import KMedoids
      2 from sklearn.metrics import pairwise_distances_argmin_min

File ~\anaconda3\envs\meu_ambiente\lib\site-packages\sklearn_extra\__init__.py:1
----> 1 from . import kernel_approximation, kernel_methods  # noqa
      3 from ._version import __version__
      5 __all__ = ["__version__"]

File ~\anaconda3\envs\meu_ambiente\lib\site-packages\sklearn_extra\kernel_approximation\__init__.py:1
----> 1 from ._fastfood import Fastfood
      4 __all__ = ["Fastfood"]

File ~\anaconda3\envs\meu_ambiente\lib\site-packages\sklearn_extra\kernel_approximation\_fastfood.py:11
      8 from sklearn.base import TransformerMixin
      9 from sklearn.utils import check_array, check_random_state
---> 11 from ..utils._cyfht import fht2 as cyfht
     14 class Fastfood(BaseEstimator, TransformerMixin):
     15     """Approximates feature map of an RBF kernel by Monte Carlo approximation
     16     of its Fourier transform.
     17
   (...)
     55
     56     """

File sklearn_extra\utils\_cyfht.pyx:1, in init sklearn_extra.utils._cyfht()

ValueError: numpy.dtype size changed, may indicate binary incompatibility. Expected 96 from C header, got 88 from PyObject
```

3. O algoritmo de K-médias é sensível a outliers nos dados. Explique.

Como o próprio nome diz, o K-médias utiliza médias aritméticas para calcular seus centróides, sendo assim, com as médias são influenciadas por valores outliers, esses valores podem “puxar” o centróide, afetando a posição dos clusters. Por exemplo, abaixo vemos o impacto no k-means com outliers e sem outliers:





4. Por que o algoritmo de DBScan é mais robusto à presença de outliers?

Diferente do k-means, que como dito acima precisa das médias para calcular seus centróides, o DBSCAN trata esses outliers sem “obrigá-los” a encaixar num clusters, já que o algoritmo é baseado em densidade dos pontos, ignorando os outliers que não atendem aos critérias da densidade mínima do algoritmo.