UNIVERSIDADE FEDERAL DO PIAUÍ – UFPI.
CENTRO DE CIÊNCIAS DA NATUREZA – CCN.
CURSO: CIÊNCIA DA COMPUTAÇÃO.
DISCIPLINA: TÓPICOS EM INTELIGÊNCIA ARTIFICIAL.
INTEGRANTES: JOÃO VICTOR CAMPELO DO VALE; JOSÉ VITOR SEGUNDO; ONOFRE MACHADO VIEIRA NETO

# Requisitos, Documentação e Considerações Finais Sobre a Função de Rotulação Automática Utilizando o Método CAIBAL

## Requisitos:

- Ter instalado o Python na versão 3 ou superior.
- Criação do ambiente virtual e instalação das bibliotecas utilizadas no projeto de acordo com sistema operacional utilizado:
- Linux/macos
    - apt install python3-pip
    - pip3 install virtualenv
    - virtualenv ../venv -p python3
    - source ../venv/bin/activate
    - pip install -r requirements.txt
- Windows
    - python3 get-pip.py
    - pip3 install virtualenv
    - virtualenv ..\venv -p python3
    - ..\venv\Scripts\activate
    - pip install -r requirements.txt

## Execução:

- Fazer o import da classe Caibal do arquivo caibal.py: from caibal import Caibal
- Atribuir a classe: *var_classe = Caibal().*
- Executar o método *execute()* com seus respectivos parâmetros. Ex:
    - var_classe.execute(path='Datasets/ffiresData.csv', cluster_alg=1, discretization_type=2, var=10, n_clusters=2, prepro=False, is_sklearn_df=False, sklearn_df_name="wine", has_index_column=False, index_column_name="1", has_target=False, has_column_name=False, verbose=True).

## Parâmetros:

- **path** (String, Default=None)
    - Caminho do arquivo .CSV caso o tipo de base de dados não seja da SKlearn.
- **cluster_alg** (Integer, [1,..., 5], Default=1)
    - Escolha de qual o algoritmo de cluster será usado.
    - São 5 tipos de algoritmos de cluster.
        - 1 - K Means
        - 2 - Affinity Propagation
        - 3 - Agglomerative Clustering
        - 4 - Mean shift
        - 5 - Spectral Clustering
- **discretization_type** (Integer, [1,2], Default=1)
    - Escolha do tipo de discretização que será usada.
    - No Caibal existem 2 métodos, o padrão e o alternativo.
        - 1 - Método Padrão
        - 2 - Método Alternativo
- **var** (Float, Default=0)
    - Caso seja selecionado a discretização do tipo alternativo, pode ser passado o valor da variação.
    - O valor da variação aumenta o número de rótulos retornados para cada grupo.

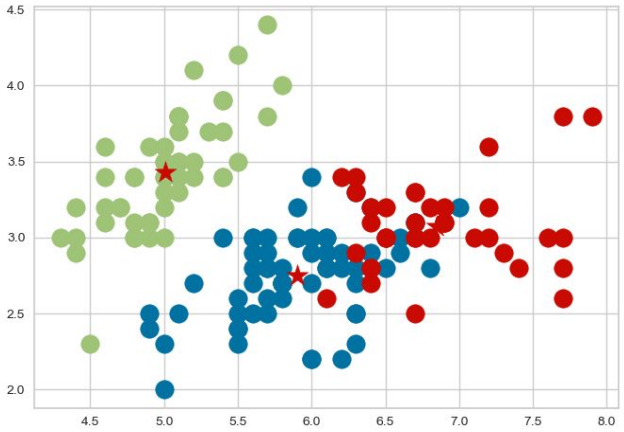
- **n_clusters** (Integer, [0,..., n] | n ∈ ℕ, Default=0)
    - Caso seja permitido pelo algoritmo de cluster selecionado, pode-se passar o número de clusters que será gerado.
    - Caso não seja informado este valor, ou informado como 0, será feito um processo de seleção do número de clusters apropriado automaticamente.
    - A métrica usada para avaliar o melhor número de clusters de forma automática é a Calinski Harabasz Score.
- **prepro** (Boolean, Default=False)
    - Caso deseja-se fazer uma padronização dos dados, utilizando o método *StandardScaler()* do Sklearn, deve-se passar esse parâmetro com o valor True.
- **is_sklearn_df** (Boolean, Default=False)
    - Caso a base de dados que se deseja testar seja proveniente dos datasets da SKlearn basta passar esse parâmetro como True.
- **sklearn_df_name** (String, Default=None)
    - Caso a fonte das bases de dados seja o Sklearn, deve-se selecionar qual a base de dados será utilizada.
    - Estão disponíveis 3 bases de dados: Iris, Wine, Breast Cancer.
    - Para selecionar alguma, passa-se como parâmetros os seguintes valores:
        - "iris" - Seleciona a base de dados Iris
        - "wine" - Seleciona a base de dados Wine
        - "cancer" - seleciona a base de dados Breast Cancer
- **has_index_column** (Boolean, Default=False)
    - Se a base de dados utilizada para estudo for proveniente externamente, através de um caminho na variável path e com a **is_sklearn_df=False,** deve-se informar se esses dados possuem uma coluna index.
    - Coluna Index é uma coluna especial que apenas enumera cada linha, sem valor de estudo.
- **index_column_name** (String, Default=None)
    - Caso o dataset externo possua um index, informado no parâmetro **has_index_column=True,** deve-se informar o nome da coluna index. Geralmente o nome dessa coluna é "index".
- **has_target** (Boolean, Default=False)
    - Se o dataset inserido por meios externos possuir uma coluna dedicada a classe de cada registro, então deve-se atribuir a este parâmetro o valor True.
- **has_column_name** (Boolean, Default=True)
    - Se o dataset inserido por meios externos não possuir um cabeçalho informando o nome de cada coluna (atributo), deve-se atribuir a este parâmetro o valor False.
    - Caso seja atribuído o valor False, será feita a adição automática dos nomes das colunas por meio padronizado.
    - O resultado dos nomes das colunas adicionadas de forma automática será:
        - Atr_1, Atr_2, …, Atr_n
        - n pertence ao conjunto dos número inteiros positivos.
- **verbose** (Boolean, Default=False)
    - Controla a exibição das informações internas.
    - Caso desejar ver os resultados dos métodos utilizados, bem como a plotagem dos clusters, atribuir a este parâmetro o valor True.

# Saída:

- A saída será uma lista de String.
- Essas Strings correspondem aos rótulos feitos de acordo com o método selecionado (Padrão ou Alternativo).
- Cada posição da lista corresponde a um cluster. Posição 0 corresponde ao cluster 0, posição 1 corresponde ao cluster 1, …, etc.
- Mesmo com o verbose=False, ainda será exibido o resultado processamento do método CAIM.
- Exemplo de retorno da função:
    - Principais parâmetros usados:
        - cluster_alg=1, discretization_type=1, n_clusters=3, prepro=False, is_sklearn_df=True, sklearn_df_name="iris".

    - Retorno:
        - ['(Attribute name: petal length (cm) Interval: (5.1, 6.9))',
        - '(Attribute name: petal width (cm) Interval: (0.1, 0.6))',
        - '(Attribute name: petal length (cm) Interval: (1.9, 5.1))']

# Exemplo de saída com Verbose=True e explicação de cada resultado:




Plotagem da classificação feita pelo algoritmo de cluster Kmean.

- **Execução do CAIM:**

  Categorical []
  # 0 GLOBAL CAIM 34.218864468864474
  # 1 GLOBAL CAIM 22.68402981294548
  # 2 GLOBAL CAIM 47.41414141414142
  # 3 GLOBAL CAIM 40.77950461389024

- **Resultado da execução do CAIM:**

  Caim splitted scheme:
  {0: [4.3, 5.4, 6.3, 7.9], 1: [2.0, 3.0, 3.3, 4.4], 2: [1.0, 1.9, 5.1, 6.9], 3: [0.1, 0.6, 1.9, 2.5]}

- **Quebra das faixas de valores por cluster:**

  Range of values ??by clusters obtained by the CAIM method
  {0: [(4.3, 5.4), (2.0, 3.0), (1.0, 1.9), (0.1, 0.6)], 1: [(5.4, 6.3), (3.0, 3.3)
  , (1.9, 5.1), (0.6, 1.9)], 2: [(6.3, 7.9), (3.3, 4.4), (5.1, 6.9), (1.9, 2.5)]}
  {0: [(4.3, 5.4), (2.0, 3.0), (1.0, 1.9), (0.1, 0.6)], 1: [(5.4, 6.3), (3.0, 3.3)
  , (1.9, 5.1), (0.6, 1.9)], 2: [(6.3, 7.9), (3.3, 4.4), (5.1, 6.9), (1.9, 2.5)]}

- **Acurácia do algoritmo caso possua uma coluna com a classe alvo:**

  Accuracy:
  (0.7514854021988339, 0.7649861514489816, 0.7581756800057786)

- **Dados das faixas de valores de cada atributo em cada clusters:**

  cluster 0
  76 (5.4, 6.3) 62 15
  89 (2.0, 3.0) 62 7
  100 (1.9, 5.1) 62 0
  95 (0.6, 1.9) 62 3

  cluster 1
  90 (4.3, 5.4) 50 5
  66 (3.3, 4.4) 50 17
  100 (1.0, 1.9) 50 0
  100 (0.1, 0.6) 50 0

  cluster 2
  95 (6.3, 7.9) 38 2
  61 (3.0, 3.3) 38 15

95 (5.1, 6.9) 38 2
74 (1.9, 2.5) 38 10

Accuracy of each attribute in each cluster
[[[76, (5.4, 6.3), 62, 15], [89, (2.0, 3.0), 62, 7], [100, (1.9, 5.1), 62, 0], [95, (0.6, 1.9), 62, 3]], [[90, (4.3, 5.4), 50, 5], [66, (3.3, 4.4), 50, 17], [100, (1.0, 1.9), 50, 0], [100, (0.1, 0.6), 50, 0]], [[95, (6.3, 7.9), 38, 2], [61, (3.0, 3.3), 38, 15], [95, (5.1, 6.9), 38, 2], [74, (1.9, 2.5), 38, 10]]]

- **Resultado dos rótulos gerados com informações sobre o nome do atributo, a acurácia, a faixa de valores, número de elementos e o número de erro:**

  Labels by Standard Method:
  Cluster 0
  Attribute name: petal length (cm), Accuracy: 100%, Interval: (1.9, 5.1), Number of elements: 62, Erro: 0
  Cluster 1
  Attribute name: petal width (cm), Accuracy: 100%, Interval: (0.1, 0.6), Number of elements: 50, Erro: 0
  Cluster 2
  Attribute name: petal length (cm), Accuracy: 95%, Interval: (5.1, 6.9), Number o f elements: 38, Erro: 2

# Limitações e problemas:

- Não é feita limpeza dos dados, logo espera-se que os dados sejam inseridos após ser feito o processo de pré-processamento.
- Alguns algoritmos, para bases de dados diferentes, podem apresentar resultados ruim. No momento, não está sendo feito uma verificação de viabilidade de execução do algoritmo selecionado para a base de dados selecionada. O K means, entretanto, apresentou bons resultados para todas as bases de dados testada (Iris, Wine, Breast Cancer, Glass, Forest Fire, Seeds).
- Se a base de dados inserida não conter colunas nomeadas e mesmo assim possuir uma coluna com característica de Index deve-se inserir o nome do primeiro registro que componhe a coluna com dados de index.
- De acordo com a dissertação “CAIBAL - Cluster-Attribute Interdependency Based Automatic Labeler” feita pelo Marcel Raimundo de Souza Moura, para a base de dados Glass foram geradas mais de um rótulo no método padrão. No método criado neste trabalho não foi criada uma forma de adicionar mais de um rótulo por cluster no método padrão.
- O parâmetro has_target somente aceita, caso seja atribuído o valor True, que o nome da coluna que contém a variável classificadora seja “target”, logo, deve ser alterada o nome caso seja diferente de “target”.
- Como requisitos descritos no escopo do trabalho, pedia-se a entrada do método discretizador, porém, neste trabalho, está sendo utilizando apenas o CAIM como método de discretização, logo não possui necessário adicionar um campo com o tipo de discretizador

# Trabalhos futuros:

- Adicionar controle de exceções para evitar o cancelamento da execução do programa após erros de algoritmos.
- Refinar a escolha de outros algoritmos de clusters para aumentar a base de comparação.
- Adicionar um campo para a seleção do tipo de métrica a ser utilizada para calcular os melhores parâmetros, atualmente está sendo usada o calinski harabasz score, mas pode ser adicionado o silhouette score para ser testado os resultado, e, assim, obter melhores resultados.
- Adicionar comentários aos métodos.
- Tornar público apenas o método execute.
- Construir uma plataforma Web que utilize essa função.
- Adicionar novos métodos de pré-processamento.
- Adicionar inserção do nome da coluna que contém a variável classificadora. No momento apenas será considerado o nome “target”.
- Modularizar a função para poder adicionar novos métodos de discretização.