

# Supervised Learning

## Dry Beans Dataset

---

**Unidade Curricular:** Inteligência Artificial

**Grupo 36:**

Diogo Filipe de Oliveira Santos  
Jéssica Mireie Fernandes do Nascimento  
João Vítor Freitas Fernandes

**Créditos:** KOKLU, M. and OZKAN, I.A., (2020), “Multiclass Classification of Dry Beans Using Computer Vision and Machine Learning Techniques.” Computers and Electronics in Agriculture, 174, 105507.

# Descrição do problema

---

O objetivo deste trabalho é utilizar *Supervised Learning* para prever o tipo de *dry bean*. Para isso, contamos com um *dataset* de 13611 amostras.

Estas amostras contêm os seguintes atributos:

- |                   |                 |               |                |
|-------------------|-----------------|---------------|----------------|
| • Area            | • AspectRation  | • Extent      | • ShapeFactor1 |
| • Perimeter       | • Eccentricity  | • Solidity    | • ShapeFactor2 |
| • MajorAxisLength | • ConvexArea    | • Roundness   | • ShapeFactor3 |
| • MinorAxisLength | • EquivDiameter | • Compactness | • ShapeFactor4 |

Este dataset é constituído por 7 tipos de feijões: *Seker*, *Barbunya*, *Bombay*, *Cali*, *Horoz*, *Sira*, *Dermason*.

Tirando partido dos diferentes valores destes atributos, utilizaremos vários classificadores para avaliar a possível classe de cada feijão, com uma taxa de acerto aceitável.

# Ferramentas e Algoritmos utilizados

---

Para desenvolver este projeto utilizou-se a linguagem de programação *Python3* com o auxílio do *Jupyter Notebook* e das seguintes *libraries*: *Pandas*, *Numpy*, *Scipy*, *Scikit-Learn*, *Matplotlib* e *Seaborn*.

Durante um pré processamento dos dados foram removidos outliers, duplicados e decidimos utilizar um sampling de 500 dados de cada classe, uma vez que o número total de dados de cada classe era variável e o menor continha 522 dados (classe: *Bombay*).

Para a implementação da técnica de classificação Decision Tree, utilizou-se um teste size de 25% e obtemos um score de cerca de 91%.

## Funções utilizadas:

```
(training_inputs, testing_inputs, training_classes, testing_classes) = train_test_split(inputs, labels, test_size= 0.25, random_state = 1)
```

```
decision_tree_classifier = DecisionTreeClassifier() #classificador
```

```
decision_tree_classifier.fit(training_inputs, training_classes) #modelo
```

```
decision_tree_classifier.score(testing_inputs, testing_classes)
```

# Ferramentas e Algoritmos utilizados

---

Pretendemos ainda implementar as seguintes técnicas de classificação:

- **Nearest Neighbor (K-NN):**  
Tenta classificar um objeto com base nos K elementos mais próximos/semelhantes.
- **Naïve Bayes (NB):**  
Assume que todos os valores são independentes, tirando partido disso para a previsão de um elemento.
- **Support Vector Machines (SVM):**  
Tenta traçar uma fronteira entre as várias classes e classificar o elemento consoante a região a que pertence.
- **Neural Networks (ANN):**  
Rede constituída por vários “neurônios” que comunicam entre si de forma a prever um dado *outcome*.