



**UNIP – Universidade Paulista**

**Bacharelado em Ciência da Computação**

**“DESENVOLVIMENTO DE UM SISTEMA BIOMÉTRICO CAPAZ DE  
IDENTIFICAR ESPÉCIES DE PLANTAS A PARTIR DE SUAS FOLHAS”**

Piero Henrique Roquetti - F033686

Yan Carlos de Oliveira - N4139E5

Igor Geovani Cavalini - F0362J8

Michael vieira do Nascimento - D8636G2

ARARAQUARA

2021

## **Índice:**

**Pg 3: .....Introdução**

**Pg 4: .....Objetivo & Motivação**

**Pg 5: .....Fundamentos das Principais Técnicas Biométricas**

**Pg 7: .....Extração de características a partir de imagens**

**Pg 9: .....Aprendizado de Máquina e classificação**

**Pg 18: .....Relatório sobre o processo experimental e análises**

**Pg 27: .....Resultado, análise e conclusão**

**Pg 34: .....Bibliografia**

**Pg 35: .....Repositório do Projeto**

## **Introdução**

O uso da Biometria começou em 1858, quando o astrônomo William Hersche se dedicou a um hobby peculiar: recolher impressões digitais retiradas na parte de trás de contratos. Contudo, somente na década de 1970, que foram estabelecidos os primeiros sistemas automatizados.

Os sistemas biométricos funcionam de diversas maneiras para realizar suas identificações, podendo ocorrer por meios de leitura de diferentes características corporais de um indivíduo, sendo os olhos, as impressões digitais, retina, dentre outros. Além disso, os sistemas biométricos mais sofisticados realizam sua identificação por meios de voz, ou até por suas características físicas, através de comparações.

A biometria é um dos caminhos mais seguros para a identificação de pessoas e proteção de dados.

Cada sistema biométrico possui um funcionamento diferente, mas no geral, possui três componentes básicos para a obtenção dos dados, sendo eles a captura que é o momento que uma amostra biométrica é adquirida (a digital, por exemplo), a extração que é a retirada das informações da amostra das características do indivíduo (template) e a comparação que é a informação obtida recentemente com a que já está armazenada no template.

Portanto, este projeto visa apresentar o desenvolvimento de um programa capaz de realizar a identificação de espécies de plantas através de suas folhas, utilizando algumas técnicas de classificação e manipulando informações. Portanto, extraindo características para o reconhecimento ser consistente e para trazer credibilidade aos resultados.

## **Objetivo & Motivação**

Visamos explorar algumas técnicas de processamento de imagem com um sistema inteligente capaz de identificar diversas espécies de plantas a partir de suas folhas, fazendo a utilização de informações reais, extraíndo atributos necessários para sua identificação e realizando a classificação com inteligência artificial.

Iremos realizar testes, a fim de obter uma consistência no desempenho e no resultado final da abstração dos dados.

## Fundamentos das Principais Técnicas Biométricas (Conceitos gerais)

A palavra Biometria (do latim, *Bio* + *Metria*) é a medição da vida, em termos gerais, o estudo estatístico das características físicas e comportamentais. Dentro da tecnologia, a biometria consiste em uma aplicação de métricas a atributos biológicos, para fins de identificação de um indivíduo.

Dentre os mais populares tipos de biometria, podemos citar os mais populares como: Impressão digital, reconhecimento facial, identificação pela íris, reconhecimento pela retina, voz, reconhecimento da assinatura digital e reconhecimento da digitação.

O método de reconhecimento biométrico pela **Impressão digital** é o mais antigo e de menor custo para a implementação, dada a baixíssima mutabilidade dos dados ao longo do tempo, ela é de extrema confiabilidade.

O **Reconhecimento facial**, por sua vez, está presente até nos celulares, ele consiste em mapear um rosto, seja em 3D ou 2D, é criada uma imagem das pessoas e a mesma vai ser utilizada como desbloqueio e identificação.

A biometria utilizando a **íris** é de extrema confiabilidade, já que a membrana permanece a mesma ao longo de toda a vida, reduzindo os riscos de envelhecimento e/ou alteração. Diferentemente do método de leitura de retina, por exemplo, é muito menos invasivo, tornando-se difícil de contornar.

O **Reconhecimento de voz** realiza uma análise físicas e comportamentais, como por exemplo, sotaques, entonação da voz, etc. E tendo como resultado um perfil sonoro. Comumente utilizado para assinaturas biométricas, ele possui um custo de implementação baixo, pois a confiabilidade se torna baixa, pois qualquer ruído pode comprometer a coleta e a análise da voz, e por causas naturais, como por exemplo a perda da voz e o envelhecimento, derrubam os índices de acertos deste reconhecimento. Uma das grandes biometrias mais seguras já criadas, já que a disposição dos vasos sanguíneos os irriga, é a de **retina**, pois variam de pessoa pra pessoa e não mudam. Os meios necessários para a coleta e a leitura dos dados não são tão simples, o que torna a falsificação dos mesmos muito difícil. Para a realização da leitura, o usuário que se dispõe para o reconhecimento, deve olhar para um dispositivo e uma luz infravermelha de baixa intensidade fará a “leitura” da retina. Por mais seguro que seja, é algo bem invasivo e incômodo.

É pouco invasivo, é baseado na análise do ritmo e cadência do usuário, quando se trata do método de **Reconhecimento pela digitação**, pois cada pessoa possui um estilo de

digitação próprio, quando comparada a velocidade de digitação, força ao pressionar as teclas, dentre outros. Pensando no baixo custo, este é um método de captação não tão simples e pouco confiável, pois o estilo de digitação pode mudar de forma inconsciente ou intencional.

## Extração de características a partir de imagens

- A palavra “Padrão” em reconhecimento de imagens, se refere a qualquer elemento que possa ser definido quantitativamente mesmo que sujeito a variações.
- O reconhecimento pode ser feito por dois tipos de métodos, sendo eles por **diferenciação** ou por **classificação** (ou ambos).
- Já as classes padrão que são reconhecidas são formadas a partir destas características que são extraídas das imagens.
- São utilizados descritores para caracterizar o objeto/padrão. Cada tipo de descritor é mais adequado a determinado aspecto, como por exemplo, forma, dimensão, textura, cor, dentre outros.

O fluxograma abaixo, mostra como é realizada a etapa de um sistema de reconhecimento de padrões.



1.

captura a imagem;

2. Separa o objeto ou o padrão de interesse (segmentação);

3. Obtem um conjunto de características extraídas do padrão;

3. É armazenado no banco de dados de padrões;

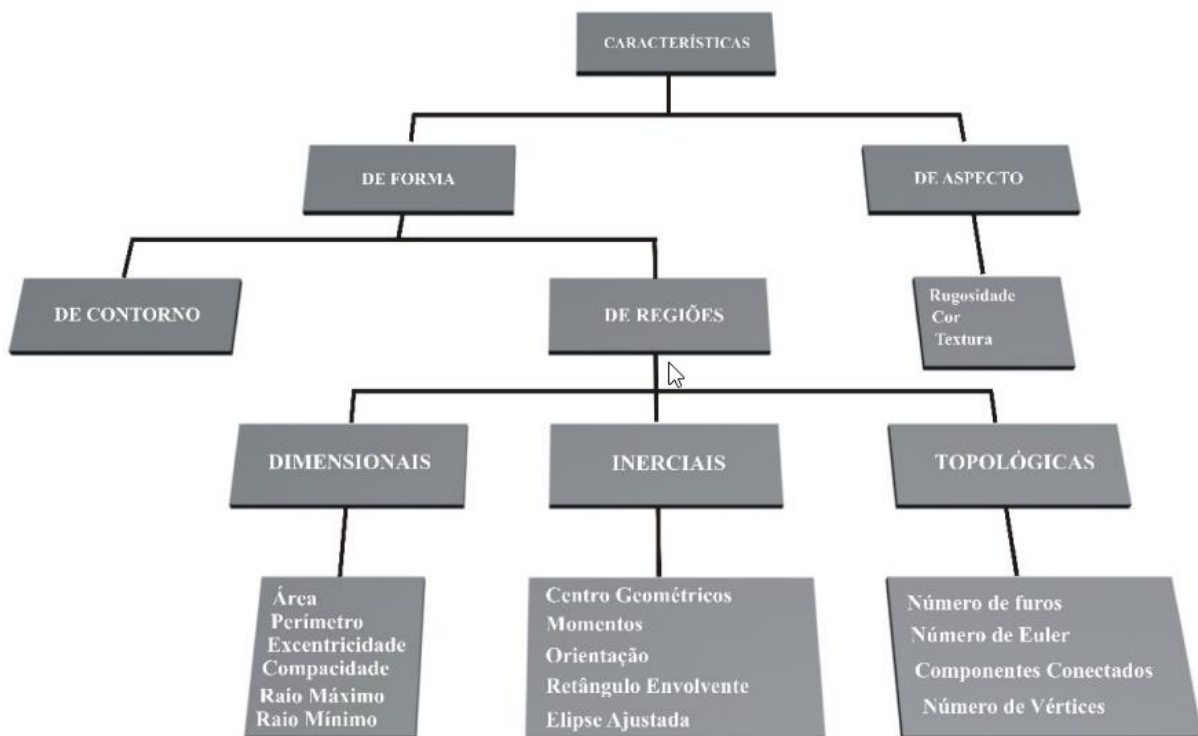
4. Classificador; e

5. Reconhece o padrão

Uma das técnicas de segmentação é dividir uma imagem em regiões que possuem o mesmo conteúdo no contexto da aplicação, ela pode ser baseada em descontinuidades e similaridades dos diferentes aspectos da imagem.

Na técnica de segmentação existem diversos outros tipos, sendo eles, Rotulação, propriedades do pixel e outras técnicas de segmentação como por exemplo, filtragem no domínio espacial, da frequência, transformação para um espaço de medida específico, baseado na morfologia matemática e contornos ativos, todos eles presentes na imagem.

Abaixo, pode-se observar os tipos de características:



Características:

1. De Forma:

1.1. De Contorno;

1.2. De Regiões:

- Dimensionais → Área, perímetro, excentricidade, compacidade, raio máximo, raio mínimo.

- Inerciais → Centros geométricos, momentos, orientação, retângulo envolvente e elipse ajustada.

- Topológicas → Número de furos, números de Euler, componentes conectados e número de vértices.

2. De Aspecto → Rugosidade, cor e textura.



## Aprendizado de Máquina e classificação

Machine Learning (ML) é uma importante área da inteligência artificial onde é possível criar algoritmos para ensinar uma determinada máquina a desempenhar tarefas.

Arthur Samuel, pioneiro nos campos dos jogos de computador, inteligência artificial e aprendizado de máquina, definiu machine learning como a “área de pesquisa que dá aos computadores a habilidade de aprender sem terem sido programados para tal”.

Um algoritmo de ML possibilita pegar um conjunto de dados de entrada e com base em determinados padrões encontrados, gerar as saídas. Cada entrada desse conjunto de dados possui suas próprias *features* (característica/atributo que descreve um objeto), e ter um conjunto delas é o ponto inicial fundamental para qualquer algoritmo de ML.

É possível, com o ML, reconhecer e extrair padrões de um grande volume de dados, construindo dessa maneira, um modelo de aprendizado. Esse aprendizado é baseado na observação e análise de dados como: exemplos, experiência direta ou instrução. Uma vez que tenham aprendido, são capazes de executar tarefas complexas e dinâmicas, prever com mais precisão, reagir em situações diversas e comportar-se de forma inteligente.

Os algoritmos são classificados em diferentes categorias (aprendizado supervisionado, aprendizado não - supervisionado, aprendizado semi - supervisionado e aprendizado por reforço), sendo duas delas muito utilizadas. São elas:

- **Aprendizado supervisionado:** Os algoritmos de aprendizagem supervisionada relacionam uma saída com uma entrada com base em dados rotulados. Neste caso, o usuário alimenta ao algoritmo pares de entradas e saídas conhecidos, normalmente na forma de vetores. Para cada saída é atribuído um rótulo, que pode ser um valor numérico ou uma classe. O algoritmo determina uma forma de prever qual o rótulo de saída com base em uma entrada informada. Como exemplo, podemos utilizar o próprio objetivo deste trabalho, que visa ensinar a máquina reconhecer a qual espécie uma folha pertence, a partir de seus dados rotulados.
- **Aprendizado não – supervisionado:** Com base em um número grande de dados não rotulados, o algoritmo busca padrões e similaridades entre os dados, permitindo identificar grupos de itens similares ou similaridade de itens novos com grupos já definidos. Estes algoritmos podem ser divididos em algoritmos de transformação e algoritmos de agrupamento. Os algoritmos de transformação são utilizados para criar uma representação de um conjunto de dados que seja mais conveniente que a original, seja

para facilitar a interpretação humana ou para melhorar o desempenho de outros algoritmos de aprendizagem. Os algoritmos de agrupamento (clustering) particionam os dados em grupos com características similares com base em critérios pré-estabelecidos, permitindo encontrar padrões entre os dados fornecidos. Diversos métodos de agrupamento podem ser aplicados, podendo estes serem baseados na distância geométrica entre os pontos, em distribuições estatísticas específicas ou levar em conta a densidade de pontos em áreas específicas do conjunto de dados. Como exemplo, o algoritmo teria a tarefa de identificar um grupo de clientes de um e-commerce para montar e-mail marketing e vitrines personalizadas.

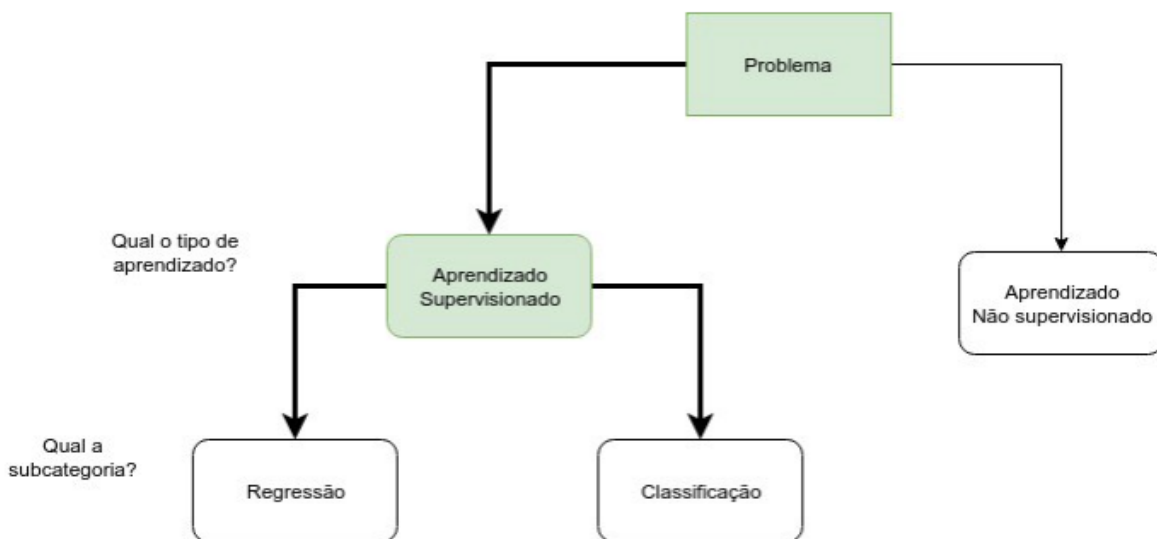


Figura 3 <https://medium.com/horadecodar/aprendizado-supervisionado-com-exemplos-supervised-learning-f9856fed2445>

Os algoritmos classificadores escolhidos para este projeto são:

### 1.1. Algoritmo KNN (K-Nearest Neighbor – Vizinho mais próximo)

O KNN é um algoritmo de classificação, e felizmente é um dos algoritmos de machine learning mais fáceis de se compreender. Seu objetivo é determinar a qual grupo uma determinada amostra vai pertencer com base nas amostras vizinhas.

Ao contrário dos outros algoritmos ele não constrói um modelo, ele faz somente o cálculo de distância. Por conta desta característica, ele é considerado um método do tipo preguiçoso (lazy).

### Funcionamento:

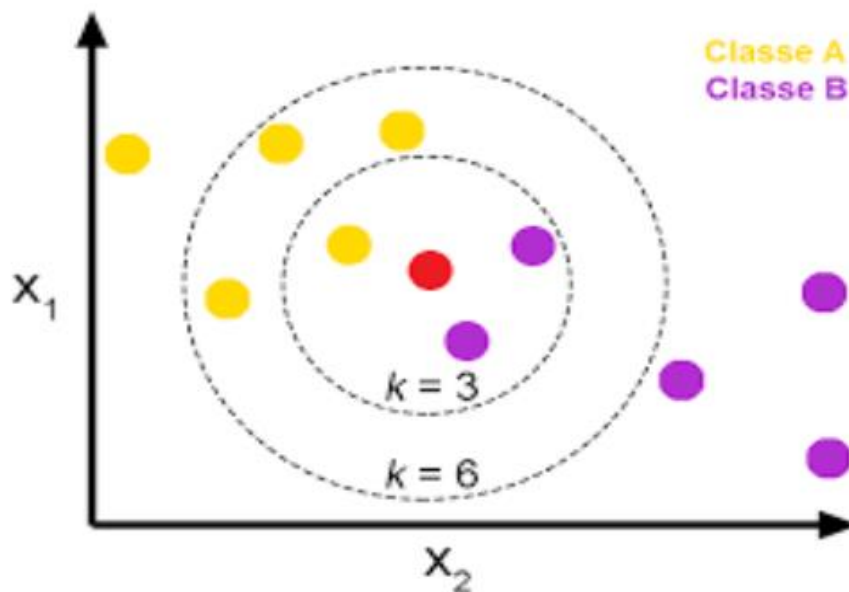


Figura 4 <https://minerandodados.com.br/machine-learning-na-pratica-knn-python/>

Nesse exemplo temos 5 amostras da classe A e 5 amostras da classe B, o objetivo é saber a qual classe a nova amostra (bolinha vermelha) vai pertencer.

Dada uma nova amostra, o algoritmo vai calcular a distância dessa amostra com todas as amostras da base de dados de treinamento.

As distâncias mais utilizadas são: Euclidiana, Manhattan, Minkowski e Ponderada.

Precisamos especificar um valor para o parâmetro K, ou seja, informar qual será o número de vizinhos que serão comparados.

No exemplo temos  $K = 3$ , então ele vai pegar somente as três amostras mais próximas para fazer a comparação.

Quando  $K$  é pequeno, a classificação fica mais sensível a regiões bem próximas (podendo ocorrer o problema de *overfitting*). Com  $K$  sendo um valor grande, a classificação fica menos sujeita a ruídos e pode ser considerada mais robusta, porém se  $K$  for grande demais, pode ser que ocorra o problema de *underfitting*.

Por fim, o processo de classificação do algoritmo KNN pode ser descrito ou resumido nas seguintes etapas:

1. Recebe uma amostra não classificada;
2. Mede a distância da nova amostra com todas as outras amostras já classificadas;
3. Obtém as  $K$  menores distâncias;
4. Verifica a classe de cada uma das amostras que têm a menor distância e conta a quantidade de cada classe que aparece;
5. Classifica a nova amostra como pertencente a classe que mais apareceu.

## **1.2. Algoritmo de Naive Bayes**

O algoritmo de Naive Bayes é um classificador probabilístico baseado no “Teorema de Bayes”, o qual foi criado por Thomas Bayes (1701-1761) para tentar provar a existência de Deus.

Tomando como premissa a suposição de independência entre as variáveis (por isso o nome Naive - Ingênuo) do problema, o modelo de Naive Bayes realiza uma classificação probabilística de observações, caracterizando-as em classes pré-definidas.

Sendo um modelo adequado para classificação de atributos discretos, o algoritmo tem aplicações na análise de créditos, diagnósticos médicos ou busca por falhas em sistemas mecânicos.

O método pode ser usado quando os atributos que descrevem as instâncias forem condicionalmente independentes. Ou seja, o teorema de Bayes trata sobre probabilidade condicional. Isto é, qual a probabilidade de o evento  $A$  ocorrer, dado o evento  $B$ .

Um problema simples que exemplifica bem o teorema é o cálculo de probabilidades em cima de diagnósticos de doenças.

Imagine que testes estão sendo feitos para o diagnóstico de uma nova doença.

Após realizar testes, coletas, análises com 100 pessoas distintas, descobre-se que 20 pessoas possuíam a doença (20%) e 80 pessoas estavam saudáveis (80%).

De todas as pessoas que possuíam a doença, 90% receberam 'Positivo' no teste. Já 30% das pessoas que não possuíam a doença também receberam o teste positivo.

### Vamos para a tabulação dos dados:

**100** pessoas realizaram o teste.

**20%** das pessoas que realizaram o teste possuíam a doença.

**90%** das pessoas que possuíam a doença, receberam positivo no teste.

**30%** das pessoas que não possuíam a doença, testaram positivo.

A partir destes dados, surge o problema: se uma nova pessoa realizar o teste e receber um resultado positivo, qual a probabilidade de ela realmente possuir a doença?

Essa probabilidade a *posteriori* é resolvida pelo Naive Bayes. Para isso, é preciso multiplicar a probabilidade a *priori* (possuir a doença) pela probabilidade de “receber um resultado positivo, dado que tem a doença”.

Com esses dados, também podemos calcular a probabilidade a *posteriori* da negação (não possuir a doença, dado que recebeu um resultado positivo).

Ou seja:

$$P(\text{doença} \mid \text{positivo}) = 20\% * 90\%$$

$$P(\text{doença} \mid \text{positivo}) = 0,2 * 0,9$$

$$P(\text{doença} \mid \text{positivo}) = 0,18$$

$$P(\text{não doença} \mid \text{positivo}) = 80\% * 30\%$$

$$P(\text{não doença} \mid \text{positivo}) = 0,8 * 0,3$$

$$P(\text{não doença} \mid \text{positivo}) = 0,24$$

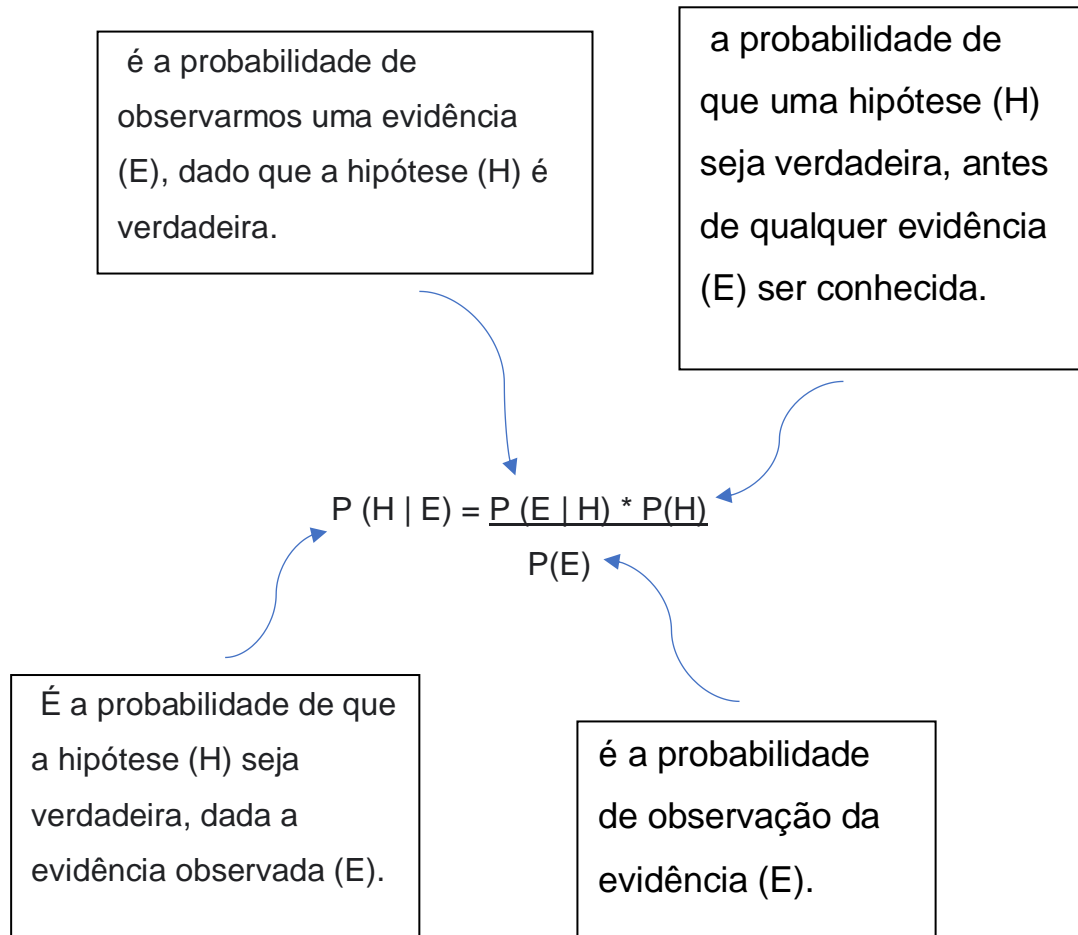
Finalizado o cálculo inicial, é preciso normalizar os dados, para que a soma das duas probabilidades resulte 100% ou 1. Para gerar os dados normalizados dividimos o resultado pela soma das duas probabilidades.

Exemplo:

$$P(\text{doença} \mid \text{positivo}) = 0,18 / (0,18 + 0,24) = 0,4285$$

$P(\text{não doença} \mid \text{positivo}) = 0,24 / (0,18 + 0,24) = 0,5714$   
 $0,4285 + 0,5714 = 0,9999...$  (ou aproximadamente 1)

Por fim, é interessante saber como foi definido o teorema de Bayes:

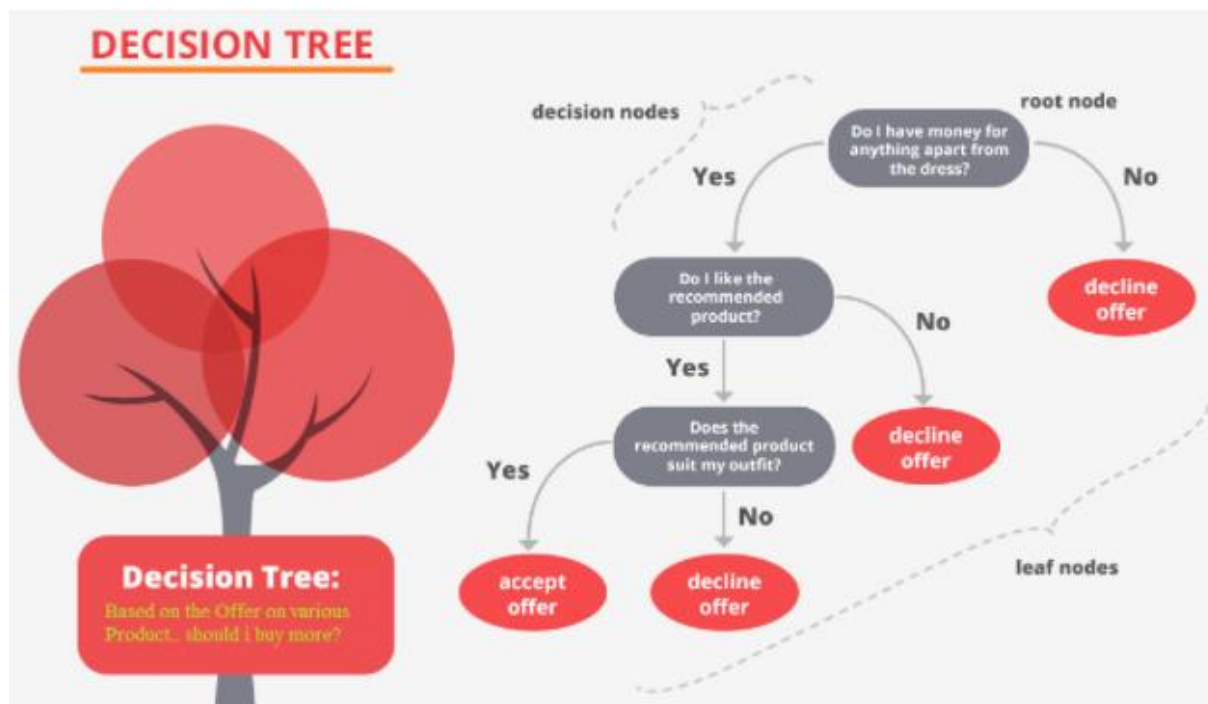


### 1.3. Algoritmo de Tree

O algoritmo de Tree (Árvore de decisão) é classificado em divisão de dados em grupos homogêneos que pode ser usado em um cenário de classificação ou regressão. Com isso o objetivo deste algoritmo é encontrar o atributo que gera a melhor divisão dos dados, subconjunto com maior pureza, ou seja, qual será a métrica utilizada para decidir qual é o melhor atributo que divide os nossos dados gerando a partição mais pura. E como o próprio nome diz, neste algoritmo vários pontos de decisão serão criados. Estes pontos são os “nós” da árvore e em cada um deles o resultado da decisão será seguir por um caminho, ou por outro. Os caminhos existentes são os “ramos”. Esta é a estrutura básica

de uma árvore de decisão. Os nós são responsáveis pelas conferências que irão indicar um ramo ou outro para sequência do fluxo.

Assim como um fluxograma, a árvore de decisão estabelece **nós** (decision nodes) que se relacionam entre si por uma hierarquia. Existe o **nó-raiz** (root node), que é o mais importante, e os **nós-folha** (leaf nodes), que são os resultados. No contexto de machine learning, o raiz é um dos atributos da base de dados e o nó-folha é a classe ou o valor que será gerado como resposta.



Na ligação entre nós, temos regras de “se então”. Ao chegar em um nó A, o algoritmo se pergunta acerca de uma regra, uma condição, como “se a característica X do registro analisado é menor do que 15?”. Se for menor, então ele vai para um lado da árvore; se for maior, então ele vai para outro. No próximo nó, segue a mesma lógica.

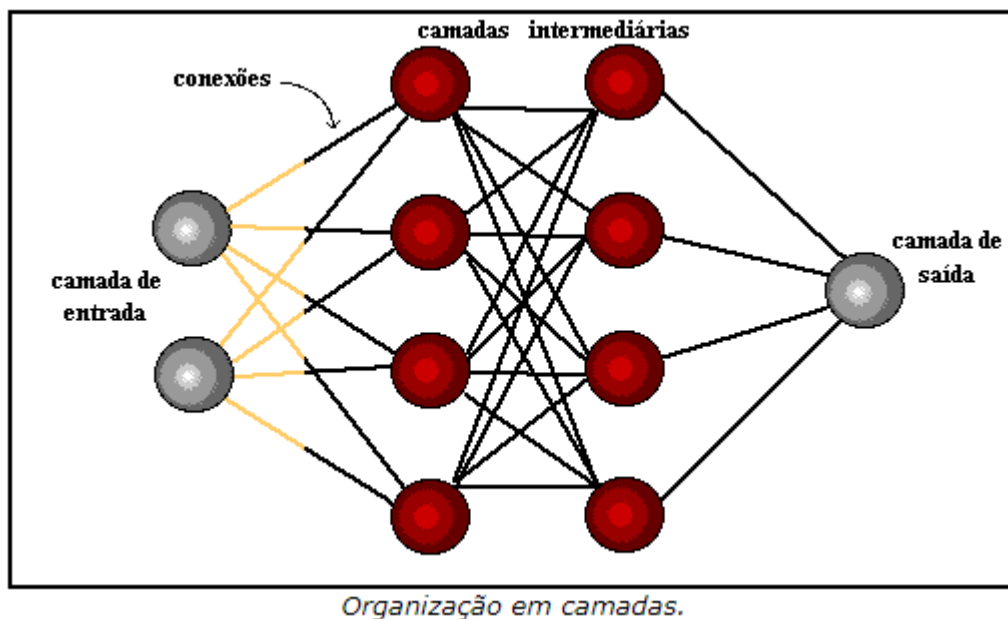
#### 1.4. Algoritmo de Neural Network

O algoritmo de Neural Network é classificado como sistemas de computação com nós interconectados que funcionam como os neurônios do cérebro humano, com isso elas podem reconhecer padrões escondidos e correlações em dados brutos, agrupá-los e classificá-los e com o tempo aprender e melhorar continuamente. A primeira rede neural foi concebida por Warren McCulloch e Walter Pitts em 1943. As redes neurais também

são idealmente desenvolvidas para ajudar as pessoas a resolver problemas complexos em diversas situações da vida real.

A maioria dos modelos de redes neurais possui alguma regra de treinamento, onde os pesos de suas conexões são ajustados de acordo com os padrões apresentados. Em outras palavras, elas aprendem através de exemplos.

Arquiteturas neurais são tipicamente organizadas em camadas, com unidades que podem estar conectadas às unidades da camada posterior.



Usualmente as camadas são classificadas em três grupos:

- **Camada de Entrada:** onde os padrões são apresentados à rede;
- **Camadas Intermediárias ou Escondidas:** onde é feita a maior parte do processamento, através das conexões ponderadas; podem ser consideradas como extratoras de características;
- **Camada de Saída:** onde o resultado é concluído e apresentado.

Uma rede neural é especificada, principalmente pela sua topologia, pelas características dos nós e pelas regras de treinamento. A seguir, serão analisados os processos de aprendizado.

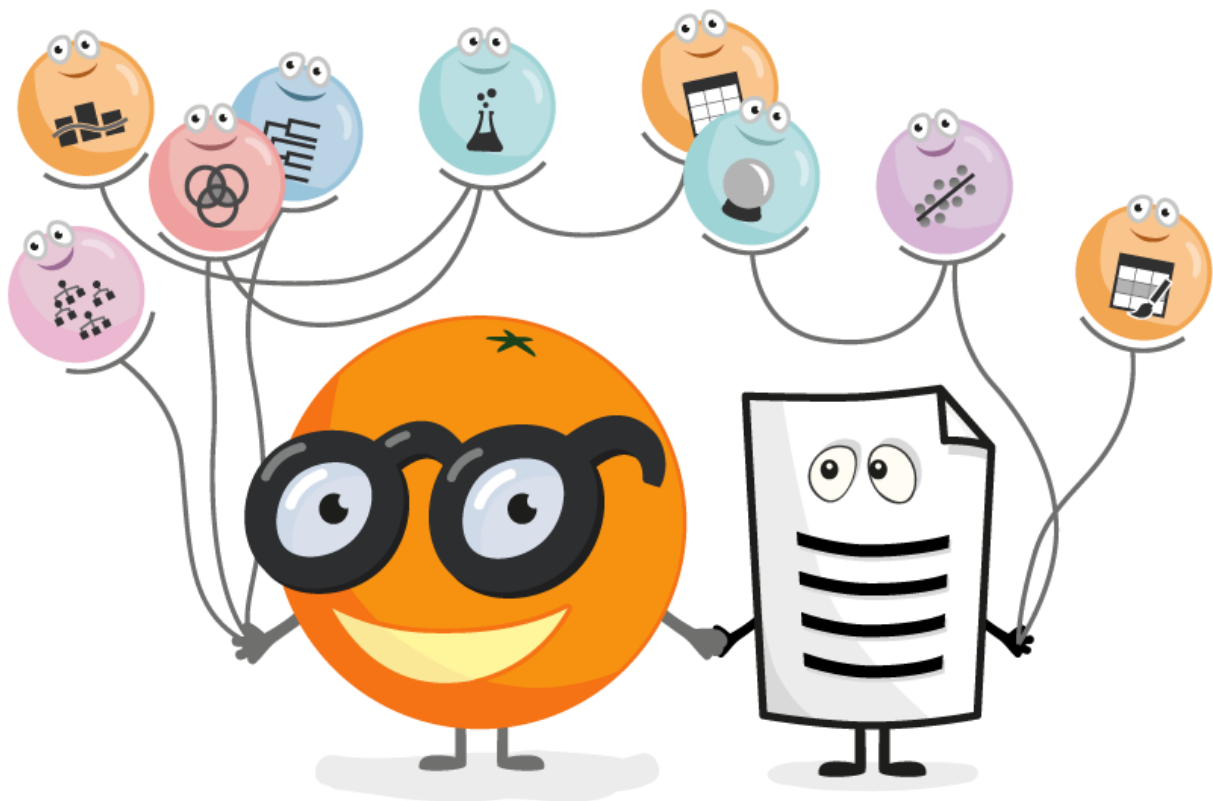


## Relatório sobre o processo experimental e análises

### 7.1 Conhecendo a ferramenta Orange;

Para nosso sistema, utilizaremos o Software Orange Data Mining que foi desenvolvido na University of Ljubljana.

Trata-se de uma ferramenta open source que permite criar todo fluxo de trabalho de um projeto de data mining, sem necessidade de código. Ideal para quem quer praticar machine learning ou para iniciantes que pretendem aplicar alguns conceitos, bem como para experts no assunto. O Orange Data Mining possui uma interface drag and drop.



### 7.2 Processo experimental

Agora que conhecemos a ferramenta, vamos conhecer a base de dados utilizada.

Assim como mencionado na proposta do trabalho, utilizaremos o dataset do LEAF, que seria uma base de imagens da University of California Irvine (UCI). Neste banco de imagens, temos 40 espécies de plantas registradas com várias imagens de suas folhas em diferentes ângulos, totalizando um montante de mais de 800 imagens reais.

Para iniciarmos os testes, selecionaremos 8 espécies de plantas, que seriam elas: Acer Palmaturu, Aesculus Californica, Betula Pubescens, Ilex Aquifolium, Papaver Sp, Populus Nigra, Pseudosasa Japonica & Schinus Terebinthifolius.

A intenção inicial, seria utilizar a entrada de imagens para montar uma base de dados com os atributos vetorizados de cada uma das imagens, a fim de criar uma instância de cada imagem e compara-las, criando assim "Clusters" (agrupamentos) baseados em referências semelhantes.

Assim como proposto, utilizaremos três métodos de classificação baseado nos seguintes paradigmas: Baseado em Instâncias; Conexionista e Probabilístico.

A sequência de criação do fluxo de dados, seria a seguinte:

**1 - Import Images:** Primeiramente importamos as imagens contidas em algum diretório (no caso, na própria máquina);

**2 - Image Embedding:** Após termos todas as imagens carregadas no Framework, iremos convertê-las em vetores contendo cada um dos atributos de cada imagem. Se direcionarmos uma ferramenta de tabela ao fluxo, podemos visualizar estes valores (imagem a seguir):

Data Table

Info

95 instances (no missing data)

2048 features

No target variable.

5 meta attributes

Variables

☒ Show variable labels (if present)

☐ Visualize numeric values

☒ Color by instance classes

Selection

☒ Select full rows

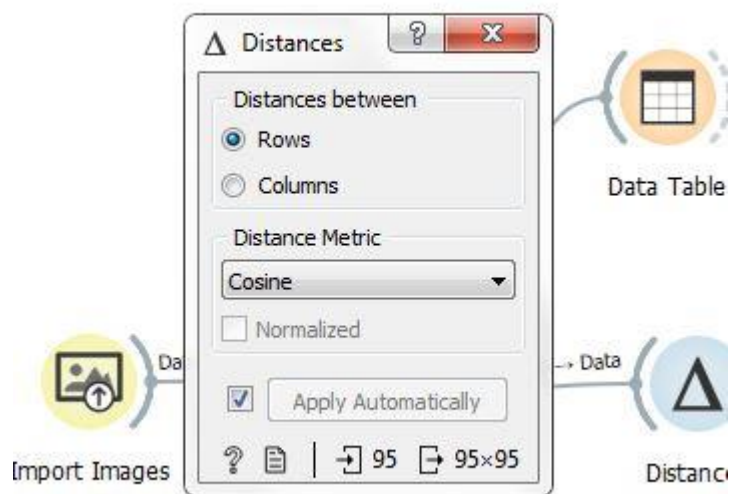
Restore Original Order

☒ Send Automatically

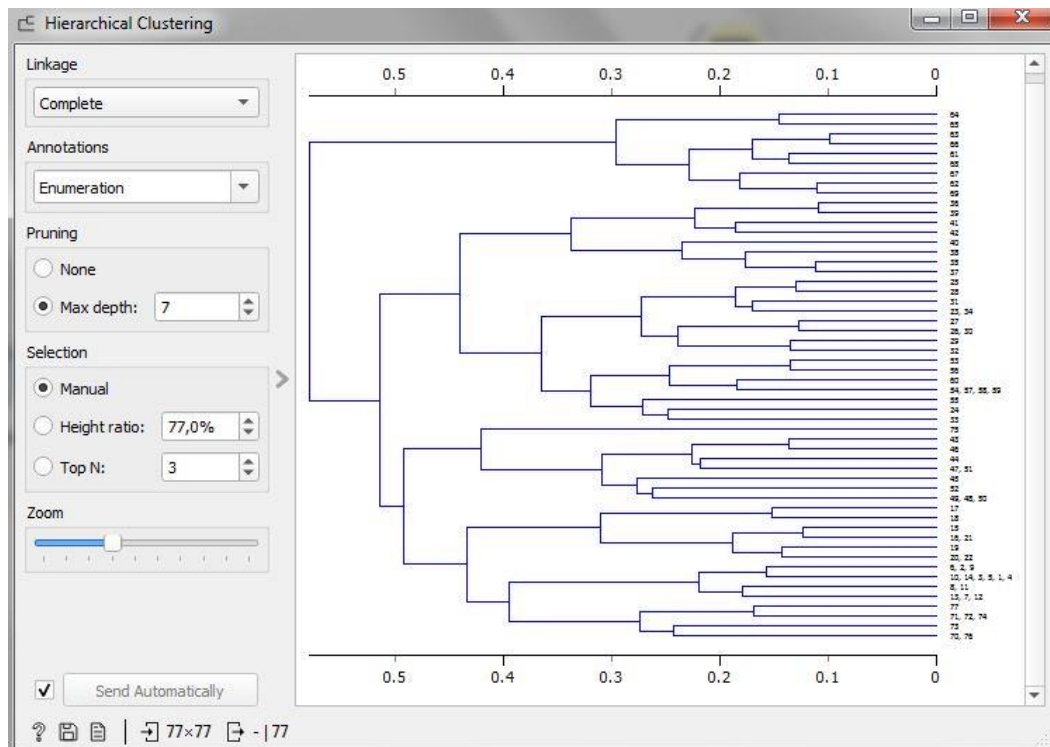
hidden origin	image name	image	size	width	height	n0 True	n1 True	n2 True	n3 True	n4 True	n5 True	n6 True
1	iPAD2_C01_EX01	iPAD2_C01_EX01...	144804	720	960	0.102298	0.451842	0.377023	0.45006	0.148807	0.0935487	0
2	iPAD2_C01_EX02	iPAD2_C01_EX02...	144368	720	960	0.208359	0.35715	0.537948	0.549029	0.0325427	0.0552013	0
3	iPAD2_C01_EX03	iPAD2_C01_EX03...	147521	720	960	0.344283	0.229643	0.155658	0.28854	0.0872608	0.0452082	0
4	iPAD2_C01_EX04	iPAD2_C01_EX04...	153654	720	960	0.19482	0.279378	0.399332	0.388099	0.354891	0.1098	0
5	iPAD2_C01_EX05	iPAD2_C01_EX05...	148829	720	960	0.278901	0.0918939	0.498009	0.25432	0.0620067	0.139913	0
6	iPAD2_C01_EX06	iPAD2_C01_EX06...	151832	720	960	0.195406	0.105473	0.175764	0.0429737	0.200718	0.19743	0
7	iPAD2_C01_EX07	iPAD2_C01_EX07...	161043	720	960	0.138366	0.050533	0.354605	0.218066	0.0803916	0.0477454	0
8	iPAD2_C01_EX08	iPAD2_C01_EX08...	142272	720	960	0.396503	0.116703	0.567055	0.397165	0.0705003	0.0363338	0
9	iPAD2_C01_EX09	iPAD2_C01_EX09...	158715	720	960	0.388248	0.0856571	0.275069	0.0874195	0.012988	0.198424	0
10	iPAD2_C01_EX10	iPAD2_C01_EX10...	137081	720	960	0.658262	0.0922212	0.141277	0.15349	0.0370803	0.142768	0
11	iPAD2_C01_EX11	iPAD2_C01_EX11...	150532	720	960	0.625942	0.135203	0.0908561	0.422705	0.132189	0.0628131	0
12	iPAD2_C01_EX12	iPAD2_C01_EX12...	134669	720	960	0.406204	0.0717851	0.123054	0.29188	0.0690344	0.0887459	0
13	iPAD2_C05_EX01	iPAD2_C05_EX01...	182703	720	960	0.0381384	0.0949689	0.117966	0.990387	1.14266	0.422401	0
14	iPAD2_C05_EX02	iPAD2_C05_EX02...	183540	720	960	0.0524618	0.442575	0.0460512	1.0894	1.26274	0.307764	0
15	iPAD2_C05_EX03	iPAD2_C05_EX03...	194340	720	960	0.13397	0.0257663	0.198159	0.764275	0.891262	0.331627	0
16	iPAD2_C05_EX04	iPAD2_C05_EX04...	188627	720	960	0.0677734	0.0216536	0.337845	0.669035	0.973015	0.395717	0
17	iPAD2_C05_EX05	iPAD2_C05_EX05...	181400	720	960	0.0631901	0.184979	0.460185	0.781531	0.949504	0.2501	0
18	iPAD2_C05_EX06	iPAD2_C05_EX06...	183773	720	960	0.0377785	0.0323392	0.11965	0.759727	0.463561	0.217653	0
19	iPAD2_C05_EX07	iPAD2_C05_EX07...	197665	720	960	0.0617535	0.03612	0.223525	0.599542	0.898866	0.17721	0
20	iPAD2_C05_EX08	iPAD2_C05_EX08...	155021	720	960	0.060258	0.0454815	0.166072	1.3081	0.698054	0.192824	0
21	iPAD2_C05_EX09	iPAD2_C05_EX09...	195519	720	960	0.0120706	0	0.322207	1.18297	0.961721	0.128878	0
22	iPAD2_C05_EX10	iPAD2_C05_EX10...	164361	720	960	0.0526267	0.0250278	0.129376	1.27845	0.646024	0.370734	0
23	iPAD2_C05_EX11	iPAD2_C05_EX11...	217322	720	960	0.102028	0	0.197766	1.21034	1.05373	0.358564	0
24	iPAD2_C05_EX12	iPAD2_C05_EX12...	170362	720	960	0.0457092	0.0226837	0.463037	1.23777	0.819104	0.159134	0
25	iPAD2_C08_EX01	iPAD2_C08_EX01...	157127	720	960	0.184127	0.82556	0.594327	0.370541	0.046176	0.138911	0
26	iPAD2_C08_EX02	iPAD2_C08_EX02...	156306	720	960	0.195117	0.831785	0.486665	0.246534	0.0864734	0.127644	0
27	iPAD2_C08_EX03	iPAD2_C08_EX03...	160370	720	960	0.403369	1.20633	0.660206	0.328204	0.0277367	0.186192	0
28	iPAD2_C08_EX04	iPAD2_C08_EX04...	155346	720	960	0.456085	0.668842	0.480728	0.277353	0.0327491	0.175087	0
29	iPAD2_C08_EX05	iPAD2_C08_EX05...	161515	720	960	0.412625	0.739383	0.496209	0.236545	0.0309567	0.162347	0
30	iPAD2_C08_EX06	iPAD2_C08_EX06...	161997	720	960	0.522727	0.719935	0.342964	0.344369	0.229102	0.331662	0
31	iPAD2_C08_EX07	iPAD2_C08_EX07...	165802	720	960	0.278691	0.897435	0.533015	0.329552	0.0159129	0.168294	0
32	iPAD2_C08_EX08	iPAD2_C08_EX08...	168970	720	960	0.473614	0.868028	0.416862	0.532448	0.0378584	0.237739	0
33	iPAD2_C08_EX09	iPAD2_C08_EX09...	161794	720	960	0.625504	0.559522	0.343985	0.588943	0.0156081	0.101796	0
34	iPAD2_C08_EX10	iPAD2_C08_EX10...	158595	720	960	0.396123	0.744136	0.691701	0.529316	0.00739283	0.326579	0

**3 - Distances:** Com as imagens vetorizadas e os valores em mãos, conseguimos calcular a distância de cada atributo, a fim de estabelecer um padrão e assim, podermos agrupá-las de acordo com os atributos semelhantes;

**3.1 -** É valido ressaltar que, como métrica, utilizaremos a função trigonométrica de "Cosseno";

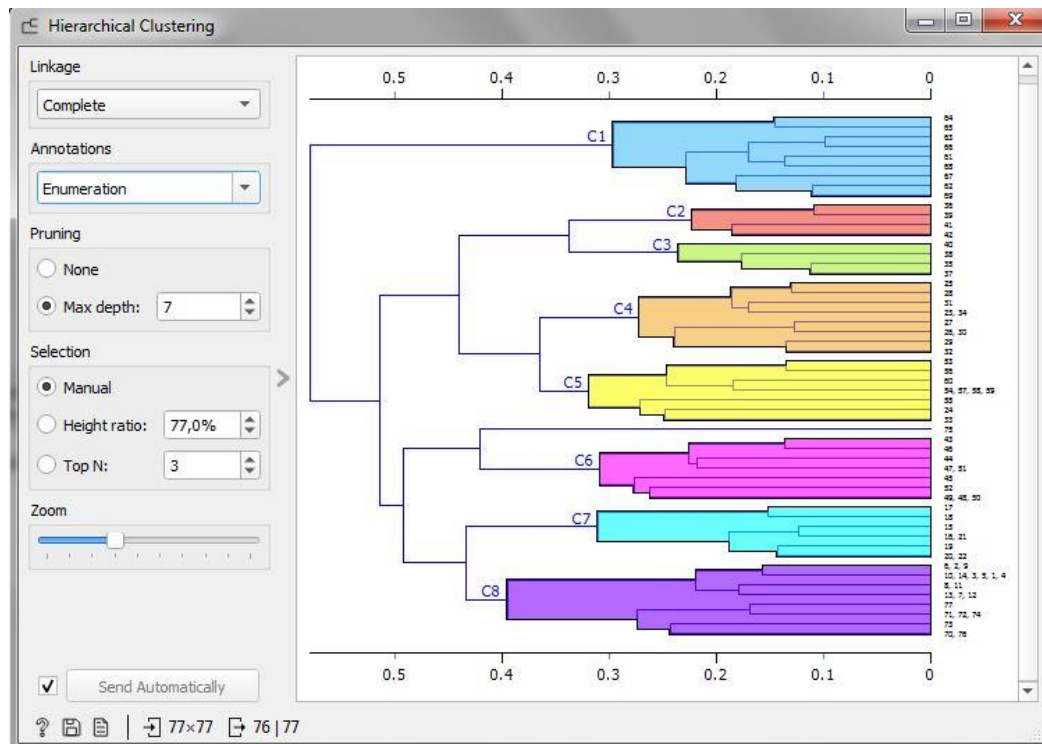


**4 - Hierarchical Clustering:** Chegamos a uma parte crucial de nossa aplicação. Na ferramenta anterior, definimos alguns padrões de distancia e com estes dados, conseguimos estabelecer uma correlação entre cada uma das imagens, fornecendo assim um dendrograma que nos apresenta alguns Clusters:



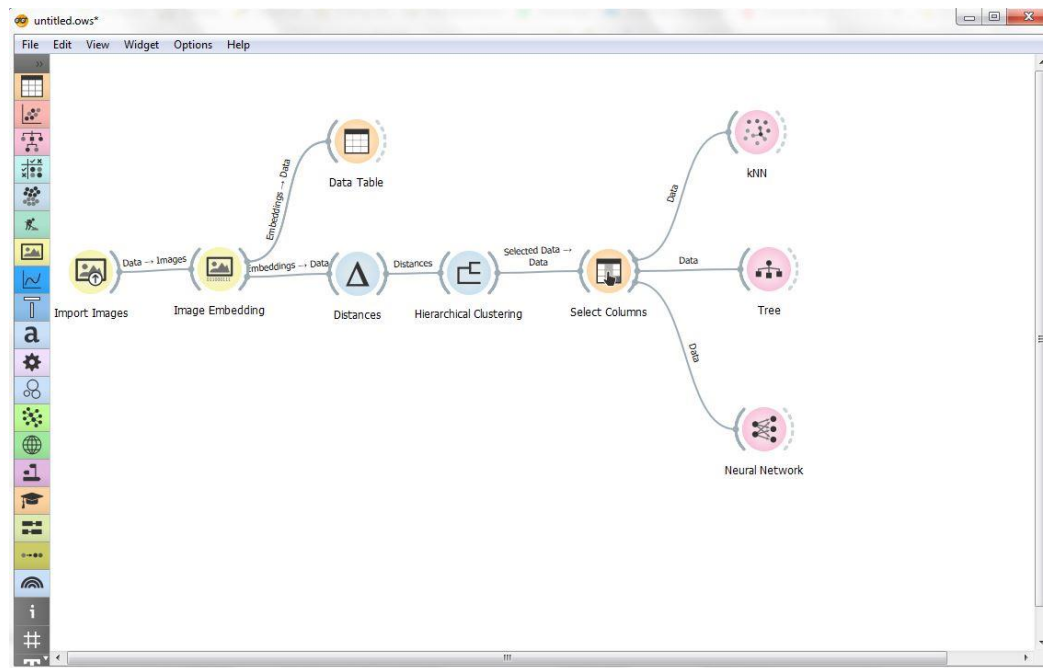
**4.1** - Neste dendrograma, é possível identificar alguns agrupamentos, que provavelmente seriam as imagens semelhantes. Neste ponto, é importantíssimo que o administrador da aplicação, identifique as instâncias que se assemelham, e crie Clusters correspondentes as instâncias das imagens. Neste caso, em específico, nossa base de dados esta dividida em 8 Clusters, que seriam os 8 tipos de plantas. Logo, deveremos separar os Clustes seguindo este padrão de agrupamento:

- Primeiro Agrupamento:



**5 - Select Columns:** Agora podemos selecionar as Colunas representando cada um dos Clusters, e utilizar estes dados para estabelecer a Variável-Alvo.

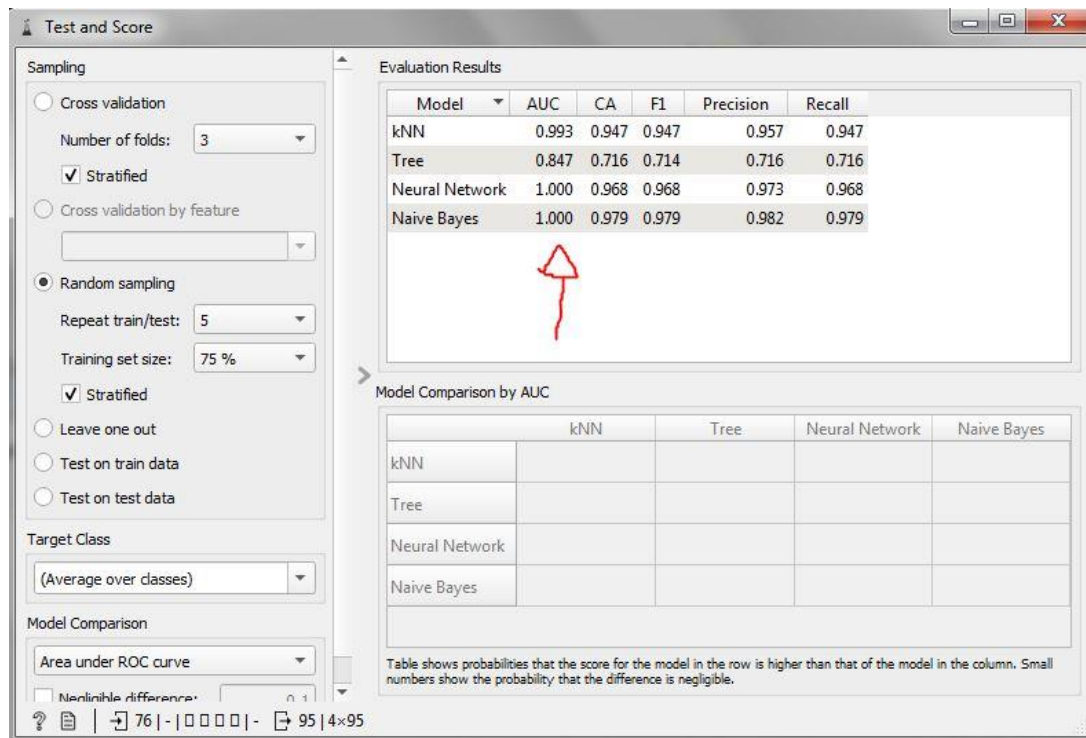
**6 - Algoritmos de Classificação:** Neste ponto, já temos os dados agrupados (Clusters) e com isto, conseguimos classificar estes Clusters, afim de formular o aprendizado e obtermos as métricas para comparação das técnicas utilizadas (K-NN, Naive Bayes, Arvore de Decisão e Redes Neurais);



7 - Test & Score: Hora de testarmos os Algoritmos. Com a Ferramenta de teste e pontuação, comparamos os dados antes do refinamento dos algoritmos de classificação, e após os mesmos. Assim, conseguindo estabelecer um referencial e comparar sua eficácia (no caso, sua acurácia).

**7.1** - Neste teste, realizamos 5 repetições (diferente de 3 repetições que seria o padrão do Orange);

**7.2** - Vale ressaltar que, uma acurácia acima de 80%, é o que nosso grupo considera "bom" para esta aplicação:

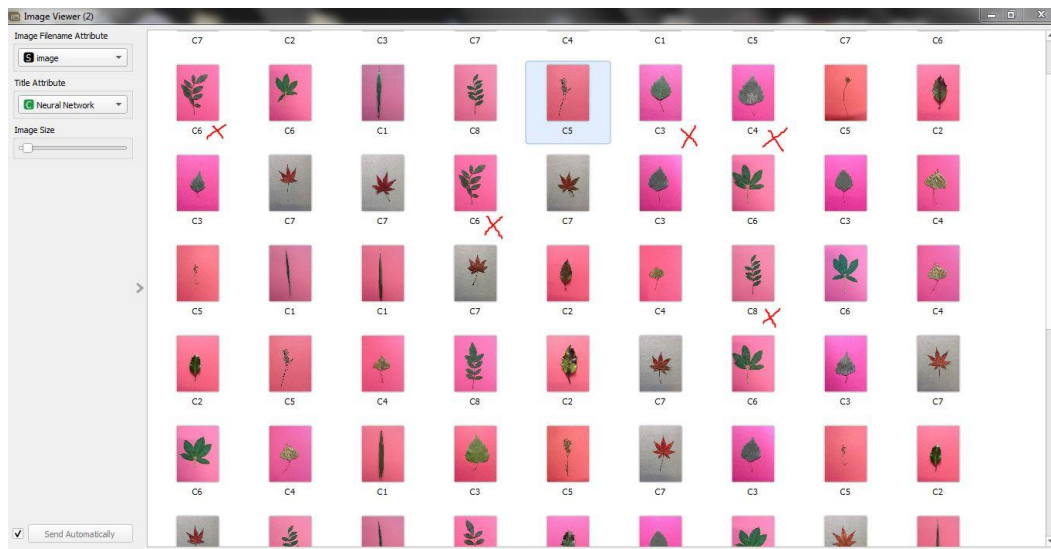


**8 - Finalização:** Por fim, podemos verificar os resultados e dar início a entrada de novas imagens, para verificar a performance do sistema na classificação das imagens da base de dados e de imagens novas.

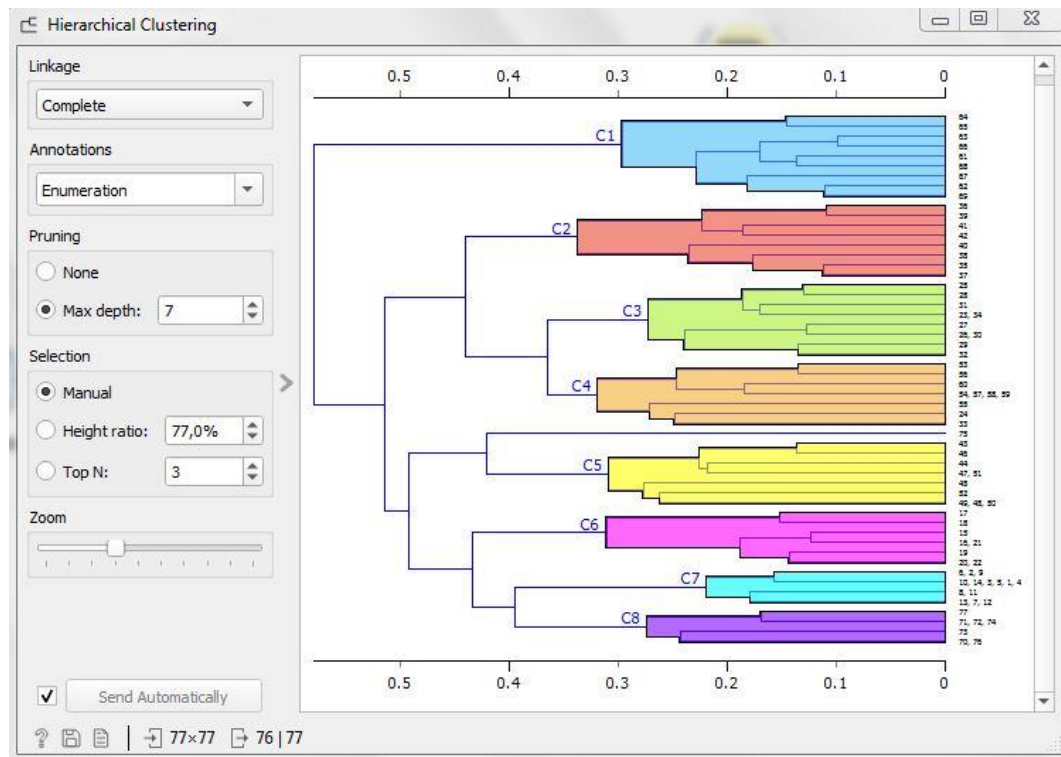
## • Resultado

**1 -** Com base nestas configurações, obtivemos 8 Cluster's de imagens, porém, ao verificarmos estes grupos, notamos que havia incoerência no agrupamento, como pode ver na imagem a seguir:

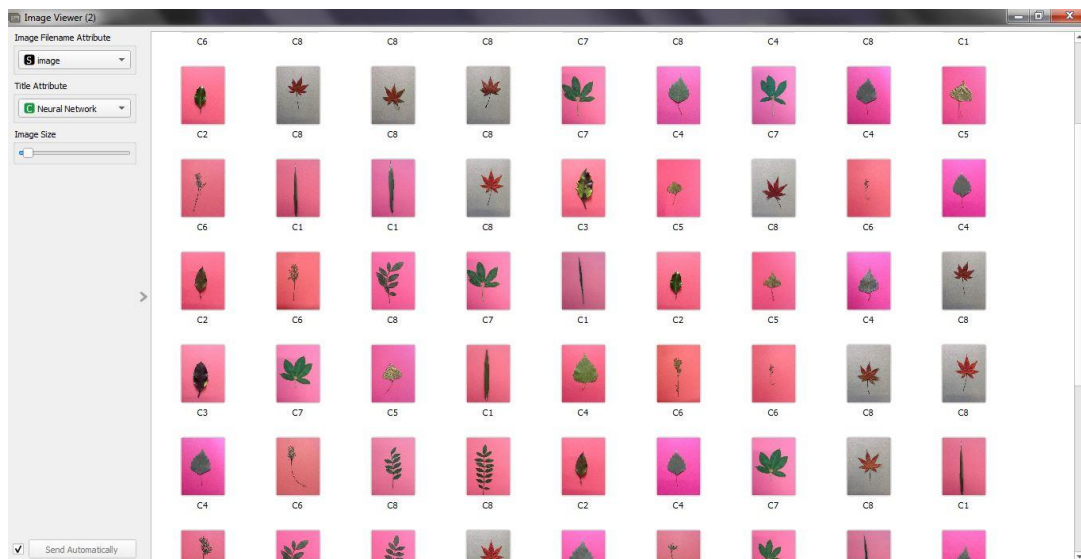




**1.1** - Desta forma, retrocedemos até a Hierarquia de Agrupamento (Hierarchical Clustering) para refazer o agrupamento, baseando-se na melhor acurácia, dentre os algoritmos de classificação, que no caso, foi as "Redes Neurais". Assim, refinamos o agrupamento e efetuamos outro teste, como mostra a imagem a seguir:



**2** - Neste agrupamento, conseguimos a melhor das classificações, utilizando Redes Neurais como algoritmo classificador. Como resultado, uma acurácia de 99,8 considerada "Boa", a separação das imagens retornou um agrupamento mais confiável, como pode ver na imagem a seguir:



- **Entrada de Imagens**

A partir deste momento, utilizaremos a entrada de imagem, para avaliar a capacidade de agrupamento de novas imagens, de acordo com a aplicação gerada acima.

Nesta etapa, utilizaremos as seguintes ferramentas do Orange, para os devidos procedimentos:

**1 - Import Images** - Importaremos as novas imagens de um diretório interno;

**2 - Image Embedding** - Aplicaremos a vetorização destas imagens, para obtermos os valores vetorizados de cada instância. Assim como no teste supervisionado;

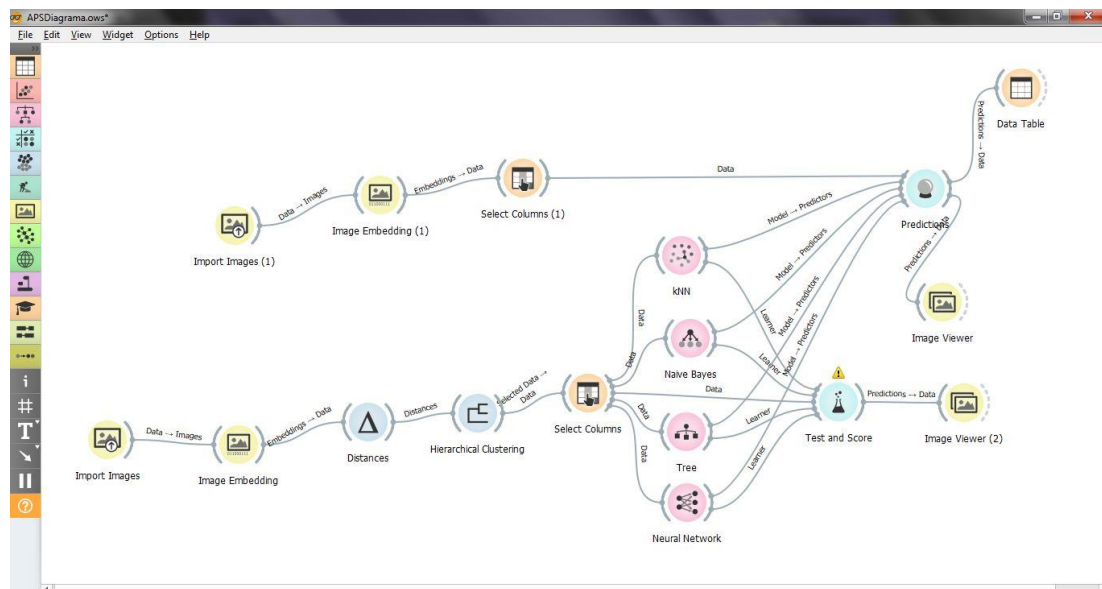
**3 - Image Embedding** - Como base de dados, utilizaremos as colunas, para efetuar a comparação.

**4 - Predictions** - Neste momento, utilizamos os dados de entrada (imagens a serem agrupadas) juntamente com a base de conhecimento gerada na aplicação. Desta forma, a ferramenta faz uma comparação dos dados de entrada e "prevê" através de probabilidade, a qual cada um dos conjuntos de atributos das imagens de entrada, esta associado, dentre os agrupamentos anteriormente gerados (Cluster's). O resultado, é uma atribuição de agrupamento para cada uma das imagens inseridas;

**5 - Image Viewer** - Para verificarmos o resultado, utilizamos a ferramenta de visualização de imagem, recebendo a previsão dos dados. Assim, obtemos as imagens agrupadas de acordo com sua probabilidade de ser pertencente a devido grupo;

**6 - Table** - Por fim, utilizamos uma tabela para podemos organizar os resultados e desta forma, analisar friamente cada uma das entradas e sua classificação com cada algoritmo;

Assim, nossa aplicação ficou da seguinte forma:



**Resultado, análise e conclusão**

Com todo o fluxo concluído, podemos agora analisar os resultados.

Resultado do Teste com a base:

\* O teste com a base de imagem e a nova Hierarquia de Cluster diminuiu muito os erros de agrupamento.

Dentre os 4 Algoritmos, destaca-se Redes Neurais, fornecendo uma base de dados com apenas 1 Erro de agrupamento (Schinus Terebinthifolius no Cluster 7, quando deveria-se estar no Cluster 9).

Cluster	Image name	image	size	width	height	rural Netwo	KNN	Tree	Naive Bayes	rural Network (C)	rural Network (C)
53	C6	Papaver Sp (22)	Papaver Sp (22)...	175532	720	960	C6	C2	C3	5.29444e-06	8.07656e-05
50	C6	Papaver Sp (1)	Papaver Sp (1) J...	192409	720	960	C6	C9	C2	8.89385e-07	2.29176e-07
38	C6	Papaver Sp (19)	Papaver Sp (19)...	198032	720	960	C6	C6	C6	3.33277e-09	2.71654e-07
28	C6	Papaver Sp (22)	Papaver Sp (22)...	175532	720	960	C6	C6	C2	6.51982e-06	7.88943e-05
17	C6	Papaver Sp (22)	Papaver Sp (22)...	175532	720	960	C6	C6	C2	4.65042e-06	0.000131112
14	C6	Papaver Sp (14)	Papaver Sp (14)...	176990	720	960	C6	C6	C9	4.18126e-08	6.57076e-06
7	C6	Papaver Sp (15)	Papaver Sp (15)...	195435	720	960	C6	C6	C6	9.45671e-08	1.29327e-06
89	C7	Aesculus Califormica (5)	Aesculus Califo...	192896	720	960	C7	C7	C7	8.82665e-09	1.90057e-06
86	C7	Aesculus Califormica (9)	Aesculus Califo...	187172	720	960	C7	C8	C7	5.55099e-09	4.72882e-07
70	C7	Aesculus Califormica (7)	Aesculus Califo...	185687	720	960	C7	C7	C7	1.05009e-07	1.49252e-06
61	C7	Aesculus Califormica (9)	Aesculus Califo...	187172	720	960	C7	C9	C7	1.30961e-08	9.99038e-07
46	C7	Aesculus Califormica (9)	Aesculus Califo...	187172	720	960	C7	C8	C7	2.19668e-08	1.38967e-06
43	C7	Aesculus Califormica (8)	Aesculus Califo...	195952	720	960	C7	C7	C7	5.17809e-08	2.69222e-05
35	C7	Aesculus Califormica (1)	Aesculus Califo...	193379	720	960	C7	C6	C7	5.25871e-08	4.02312e-06
25	C7	Aesculus Califormica (6)	Aesculus Califo...	195952	720	960	C7	C7	C7	7.97243e-09	7.84993e-06
22	C9	Schinus Terebinthifolius (8)	Schinus Terebi...	184669	720	960	C7	C7	C2	1.3554e-05	0.0010211
11	C7	Aesculus Califormica (5)	Aesculus Califo...	192896	720	960	C7	C7	C7	1.25729e-08	1.24567e-06
9	C7	Aesculus Califormica (7)	Aesculus Califo...	185687	720	960	C7	C6	C7	5.65966e-08	6.93671e-07
95	C8	Acer Palmaturu (5)	Acer Palmaturu...	187827	720	960	C8	C4	C3	1.30688e-10	2.60809e-07
88	C8	Acer Palmaturu (14)	Acer Palmaturu...	203745	720	960	C8	C8	C8	6.36705e-12	5.57265e-08
87	C8	Acer Palmaturu (13)	Acer Palmaturu...	202492	720	960	C8	C8	C8	1.21028e-11	2.44766e-08
85	C8	Acer Palmaturu (7)	Acer Palmaturu...	197711	720	960	C8	C3	C8	6.05665e-10	1.43974e-07
67	C8	Acer Palmaturu (4)	Acer Palmaturu...	202098	720	960	C8	C8	C8	1.62695e-10	7.43041e-07
65	C8	Acer Palmaturu (14)	Acer Palmaturu...	203745	720	960	C8	C8	C8	4.7004e-12	5.99853e-08
62	C8	Acer Palmaturu (12)	Acer Palmaturu...	187243	720	960	C8	C8	C8	3.81483e-11	6.14087e-08
55	C8	Acer Palmaturu (5)	Acer Palmaturu...	187827	720	960	C8	C8	C2	1.23222e-10	2.08233e-07
51	C8	Acer Palmaturu (14)	Acer Palmaturu...	203745	720	960	C8	C8	C8	1.11268e-11	9.9208e-08
45	C8	Acer Palmaturu (8)	Acer Palmaturu...	209790	720	960	C8	C7	C8	6.09853e-11	1.13289e-07
42	C8	Acer Palmaturu (10)	Acer Palmaturu...	199170	720	960	C8	C8	C8	2.14464e-10	1.37218e-07
31	C8	Acer Palmaturu (2)	Acer Palmaturu...	224313	720	960	C8	C8	C2	2.62423e-07	0.000157716
23	C8	Acer Palmaturu (3)	Acer Palmaturu...	222415	720	960	C8	C8	C8	2.44809e-09	5.22098e-06
21	C8	Acer Palmaturu (15)	Acer Palmaturu...	199107	720	960	C8	C8	C8	2.31291e-09	1.13876e-06
20	C8	Acer Palmaturu (5)	Acer Palmaturu...	187827	720	960	C8	C8	C2	1.16402e-09	2.79632e-06
8	C8	Acer Palmaturu (8)	Acer Palmaturu...	209790	720	960	C8	C7	C8	4.63872e-12	1.05612e-08

Assim como Redes Neurais, o algoritmo KNN (Enésimo-Vizinho-Próximo), obteve o mesmo resultado no agrupamento, deixando apenas 1 erro de Cluster (o mesmo de Redes Neurais), como pode-se verificar abaixo:



Data Table												
Embedded images		Embedded images										
Cluster	image name	image	size	width	height	Neural Network	kNN	Tree	Naive Bayes	Neural Network (C)	Neural Network (C)	Neural Network (C)
hidden origin	image	ite/Desktop/APS										
76	C5	Populus Nigra (2)	155284	720	960	C5	C5	C7	C2	1.14301e-05	0.000712365	
81	C5	Populus Nigra (8)	158897	720	960	C5	C5	C2	C2	6.18123e-07	0.00103292	
82	C5	Populus Nigra (3)	184557	720	960	C5	C5	C3	C3	1.59132e-07	1.2967e-06	
7	C6	Papaver Sp (15)	195435	720	960	C6	C6	C6	C6	9.45671e-08	1.29327e-06	
14	C6	Papaver Sp (14)	176090	720	960	C6	C6	C9	C2	4.18126e-08	6.57076e-06	
17	C6	Papaver Sp (22)	175532	720	960	C6	C6	C2	C3	4.65042e-06	0.000131112	
28	C6	Papaver Sp (22)	175532	720	960	C6	C6	C6	C2	6.51982e-06	7.88943e-05	
38	C6	Papaver Sp (19)	198032	720	960	C6	C6	C6	C6	3.33277e-09	2.71654e-07	
50	C6	Papaver Sp (1)	192409	720	960	C6	C6	C9	C2	8.89385e-07	2.29176e-07	
53	C6	Papaver Sp (22)	175532	720	960	C6	C6	C2	C3	5.29444e-06	8.07656e-05	
64	C6	Papaver Sp (18)	204350	720	960	C6	C6	C6	C6	4.28859e-08	5.13406e-07	
69	C6	Papaver Sp (21)	185096	720	960	C6	C6	C9	C2	0.000280929	0.000606175	
77	C6	Papaver Sp (14)	176090	720	960	C6	C6	C1	C3	4.61956e-08	4.60963e-06	
78	C6	Papaver Sp (19)	198032	720	960	C6	C6	C6	C6	6.57218e-08	2.59663e-06	
90	C6	Papaver Sp (21)	185096	720	960	C6	C6	C6	C6	0.000259136	0.000341102	
9	C7	Aesculus Californica (7)	185687	720	960	C7	C7	C6	C7	5.65966e-08	6.93671e-07	
11	C7	Aesculus Californica (5)	192896	720	960	C7	C7	C7	C7	1.25729e-08	1.24567e-06	
22	C7	Schinus Terebinthifolius (8)	184669	720	960	C7	C7	C7	C7	1.3554e-05	0.0010211	
25	C7	Aesculus Californica (6)	195952	720	960	C7	C7	C7	C7	7.97243e-09	7.84993e-06	
35	C7	Aesculus Californica (1)	193379	720	960	C7	C7	C6	C7	5.25871e-08	4.02312e-06	
43	C7	Aesculus Californica (6)	195952	720	960	C7	C7	C7	C7	5.17809e-08	2.6922e-05	
46	C7	Aesculus Californica (9)	187172	720	960	C7	C7	C8	C7	2.19668e-08	1.38967e-06	
61	C7	Aesculus Californica (9)	187172	720	960	C7	C7	C9	C7	1.30961e-08	9.99038e-07	
70	C7	Aesculus Californica (7)	185687	720	960	C7	C7	C7	C7	1.05009e-07	1.49252e-06	
86	C7	Aesculus Californica (9)	187172	720	960	C7	C7	C8	C7	5.55099e-09	4.72882e-07	
89	C7	Aesculus Californica (5)	192896	720	960	C7	C7	C7	C7	8.82665e-09	1.90057e-06	
1	C8	Acer Palmaturu (8)	202098	720	960	C8	C8	C7	C8	9.98463e-11	1.81771e-07	
4	C8	Acer Palmaturu (10)	199170	720	960	C8	C8	C8	C8	3.50015e-11	2.24075e-08	
8	C8	Acer Palmaturu (8)	209790	720	960	C8	C8	C7	C8	4.63872e-12	1.05612e-08	
20	C8	Acer Palmaturu (5)	187827	720	960	C8	C8	C8	C8	1.16402e-09	2.79632e-06	
21	C8	Acer Palmaturu (15)	199107	720	960	C8	C8	C8	C8	2.31291e-09	1.13876e-06	
23	C8	Acer Palmaturu (3)	222415	720	960	C8	C8	C8	C8	2.44809e-09	5.22098e-06	
31	C8	Acer Palmaturu (2)	224313	720	960	C8	C8	C8	C2	2.62433e-07	0.000157716	

Já o algoritmo Árvore de Decisões, não fora tão efetivo. Analisando sua tabela de agrupamentos, identificamos vários erros, como pode ser observado a seguir:

Data Table												
Embedded images		Embedded images										
Cluster	image name	image	size	width	height	Neural Network	kNN	Tree	Naive Bayes	Neural Network (C)	Neural Network (C)	Neural Network (C)
hidden origin	image	ite/Desktop/APS										
33	C3	Populus Nigra (8)	158897	720	960	C3	C3	C1	C2	1.27255e-06	0.0023513	
26	C4	Betula Pubescens (13)	185035	720	960	C4	C4	C2	C2	3.18911e-05	0.000882216	
41	C2	Ilex Aquifolium (9)	154549	720	960	C2	C2	C3	C2	0.00464072	0.918405	
18	C2	Ilex Aquifolium (9)	154549	720	960	C2	C2	C3	C2	0.0141075	0.917775	
27	C5	Populus Nigra (2)	155284	720	960	C5	C5	C4	C2	3.01914e-06	0.000406817	
93	C4	Betula Pubescens (3)	175569	720	960	C4	C4	C4	C3	4.01014e-07	1.9496e-06	
92	C4	Betula Pubescens (6)	171682	720	960	C4	C4	C4	C2	8.26477e-08	9.71269e-06	
75	C4	Betula Pubescens (12)	161599	720	960	C4	C4	C4	C2	1.22022e-06	0.000440064	
72	C4	Betula Pubescens (11)	174940	720	960	C4	C4	C4	C2	4.15376e-08	2.85518e-06	
62	C4	Betula Pubescens (3)	175569	720	960	C4	C4	C4	C2	1.5239e-06	1.75829e-05	
50	C4	Betula Pubescens (12)	161599	720	960	C4	C4	C4	C2	3.3299e-07	7.33902e-05	
44	C4	Betula Pubescens (6)	171682	720	960	C4	C4	C4	C2	7.3364e-08	4.49637e-05	
19	C4	Betula Pubescens (4)	187914	720	960	C4	C4	C4	C2	7.34633e-07	2.02015e-05	
15	C4	Betula Pubescens (11)	174940	720	960	C4	C4	C4	C2	2.74157e-06	4.01469e-05	
37	C2	Ilex Aquifolium (2)	155530	720	960	C2	C2	C4	C2	0.103245	0.8539	
31	C3	Populus Nigra (8)	158897	720	960	C3	C3	C5	C2	6.18123e-07	0.00103292	
39	C3	Populus Nigra (8)	158897	720	960	C3	C3	C5	C2	2.45447e-07	0.000804196	
5	C3	Populus Nigra (9)	173872	720	960	C3	C3	C5	C2	2.96038e-06	0.000357578	
24	C4	Betula Pubescens (3)	175569	720	960	C4	C4	C5	C2	2.10495e-06	5.02648e-05	
83	C2	Ilex Aquifolium (9)	154549	720	960	C2	C2	C5	C2	0.0197471	0.836128	
22	C9	Schinus Terebinthifolius (8)	184669	720	960	C7	C7	C6	C2	1.3554e-05	0.0010211	
28	C6	Papaver Sp (22)	175532	720	960	C6	C6	C6	C2	6.51982e-06	7.88943e-05	
76	C5	Populus Nigra (2)	155284	720	960	C5	C5	C7	C2	1.14301e-05	0.000712365	
55	C9	Acer Palmaturu (5)	187827	720	960	C8	C8	C3	C2	1.23223e-10	2.08233e-07	
31	C8	Acer Palmaturu (2)	224313	720	960	C8	C8	C8	C2	2.62433e-07	0.000157716	
20	C8	Acer Palmaturu (5)	187827	720	960	C8	C8	C8	C2	1.16402e-09	2.79632e-06	
59	C2	Ilex Aquifolium (9)	154549	720	960	C2	C2	C2	C2	0.00628789	0.906583	
69	C6	Papaver Sp (21)	185096	720	960	C6	C6	C9	C2	0.000280929	0.000606175	
50	C6	Papaver Sp (1)	192409	720	960	C6	C6	C9	C2	8.89385e-07	2.29176e-07	
14	C6	Papaver Sp (14)	176090	720	960	C6	C6	C9	C2	4.18126e-08	6.57076e-06	
77	C6	Papaver Sp (14)	176090	720	960	C6	C6	C1	C3	4.61956e-08	4.60963e-06	
53	C6	Papaver Sp (22)	175532	720	960	C6	C6	C2	C3	5.29444e-06	8.07656e-05	
17	C6	Papaver Sp (22)	175532	720	960	C6	C6	C2	C3	4.65042e-06	0.000131112	
94	C3	Ilex Aquifolium (1)	167182	720	960	C3	C3	C3	C3	0.00046573	0.000313331	


Por fim, o algoritmo Naive Bayes retornou um agrupamento errôneo, assim como Árvore de Decisões. Segue sua tabela de agrupamento, com os erros evidenciados:

Data Table												
Embedded images		Embedded images										
	Cluster	image name	image	size	width	height	Neural Network	kNN	Tree	Naive Bayes	Neural Network (C1)	Neural Network (C2)
hidden origin			ite/Desktop/APS image									
33	C5	Populus Nigra (8)	Populus Nigra (...)	158897	720	960	C5	C5	C1	C2	1.27255e-06	0.0023513
26	C4	Betula Pubescens (13)	Betula Pubescen...	185035	720	960	C4	C4	C2	C2	3.18911e-05	0.000882216
41	C2	Ilex Aquifolium (9)	Ilex Aquifolium ...	154549	720	960	C2	C2	C3	C2	0.00464072	0.918405
18	C2	Ilex Aquifolium (9)	Ilex Aquifolium ...	154549	720	960	C2	C2	C3	C2	0.0141075	0.917775
27	C5	Populus Nigra (2)	Populus Nigra (...)	155284	720	960	C5	C5	C4	C2	3.31934e-06	0.000240817
93	C4	Betula Pubescens (3)	Betula Pubescen...	175569	720	960	C4	C4	C4	C2	4.10194e-07	1.96896e-06
92	C4	Betula Pubescens (6)	Betula Pubescen...	171682	720	960	C4	C4	C4	C2	8.26477e-08	9.71269e-06
75	C4	Betula Pubescens (12)	Betula Pubescen...	161599	720	960	C4	C4	C4	C2	1.22022e-06	0.000440064
72	C4	Betula Pubescens (11)	Betula Pubescen...	174940	720	960	C4	C4	C4	C2	4.15376e-08	2.55518e-06
60	C4	Betula Pubescens (3)	Betula Pubescen...	175569	720	960	C4	C4	C4	C2	1.5239e-06	1.75829e-05
52	C4	Betula Pubescens (12)	Betula Pubescen...	161599	720	960	C4	C4	C4	C2	3.3299e-07	7.3902e-05
44	C4	Betula Pubescens (6)	Betula Pubescen...	171682	720	960	C4	C4	C4	C2	7.3364e-08	4.49637e-05
19	C4	Betula Pubescens (4)	Betula Pubescen...	187914	720	960	C4	C4	C4	C2	7.34633e-07	2.02015e-05
15	C4	Betula Pubescens (11)	Betula Pubescen...	174940	720	960	C4	C4	C4	C2	2.74157e-06	4.01469e-05
37	C2	Ilex Aquifolium (8)	Ilex Aquifolium ...	155330	720	960	C2	C2	C4	C2	0.106345	0.8388
81	C5	Populus Nigra (8)	Populus Nigra (...)	158897	720	960	C5	C5	C5	C2	6.18123e-07	0.00103292
39	C5	Populus Nigra (8)	Populus Nigra (...)	158897	720	960	C5	C5	C5	C2	2.45447e-07	0.000804196
5	C5	Populus Nigra (9)	Populus Nigra (...)	173872	720	960	C5	C5	C5	C2	2.96038e-06	0.000357578
24	C4	Betula Pubescens (3)	Betula Pubescen...	175569	720	960	C4	C4	C5	C2	2.10495e-06	5.02648e-05
83	C2	Ilex Aquifolium (9)	Ilex Aquifolium ...	154549	720	960	C2	C2	C5	C2	0.0197471	0.835138
22	C9	Schinus Terebinthifolius (8)	Schinus Terebin...	184669	720	960	C7	C7	C6	C2	1.3554e-05	0.0010211
28	C6	Papaver Sp (22)	Papaver Sp (22)...	175532	720	960	C6	C6	C6	C2	6.51982e-06	7.88943e-05
76	C5	Populus Nigra (2)	Populus Nigra (...)	155284	720	960	C5	C5	C7	C2	1.14301e-05	0.900712365
55	C8	Acer Palmaturu (5)	Acer Palmaturu...	187827	720	960	C8	C8	C8	C2	1.23222e-10	2.08233e-07
31	C8	Acer Palmaturu (2)	Acer Palmaturu...	224313	720	960	C8	C8	C8	C2	2.62433e-07	0.000157716
20	C8	Acer Palmaturu (5)	Acer Palmaturu...	187827	720	960	C8	C8	C8	C2	1.16402e-09	2.79632e-06
59	C2	Ilex Aquifolium (9)	Ilex Aquifolium ...	154549	720	960	C2	C2	C8	C2	0.00628789	0.906583
69	C6	Papaver Sp (21)	Papaver Sp (21)...	185096	720	960	C6	C6	C9	C2	0.000280929	0.000606175
50	C6	Papaver Sp (1)	Papaver Sp (1)...	192409	720	960	C6	C6	C9	C2	8.89385e-07	2.29176e-07
14	C5	Papaver Sp (14)	Papaver Sp (14)...	176090	720	960	C5	C5	C9	C2	4.18125e-08	6.57075e-05
77	C6	Papaver Sp (14)	Papaver Sp (14)...	176090	720	960	C6	C6	C1	C3	4.61956e-08	4.60962e-06
53	C6	Papaver Sp (22)	Papaver Sp (22)...	175532	720	960	C6	C6	C2	C3	5.29444e-06	8.07656e-05
17	C6	Papaver Sp (22)	Papaver Sp (22)...	175532	720	960	C6	C6	C2	C3	4.65042e-06	0.000131112
84	C2	Ilex Aquifolium (1)	Ilex Aquifolium ...	167182	720	960	C2	C2	C2	C2	0.00446572	0.000212271

- Entrada de Dados

Isolando os algoritmos "Redes Neurais" e "KNN", como os melhores, podemos agrupar as imagens de entrada utilizando ambos. Desta maneira obtivemos os seguintes agrupamentos:

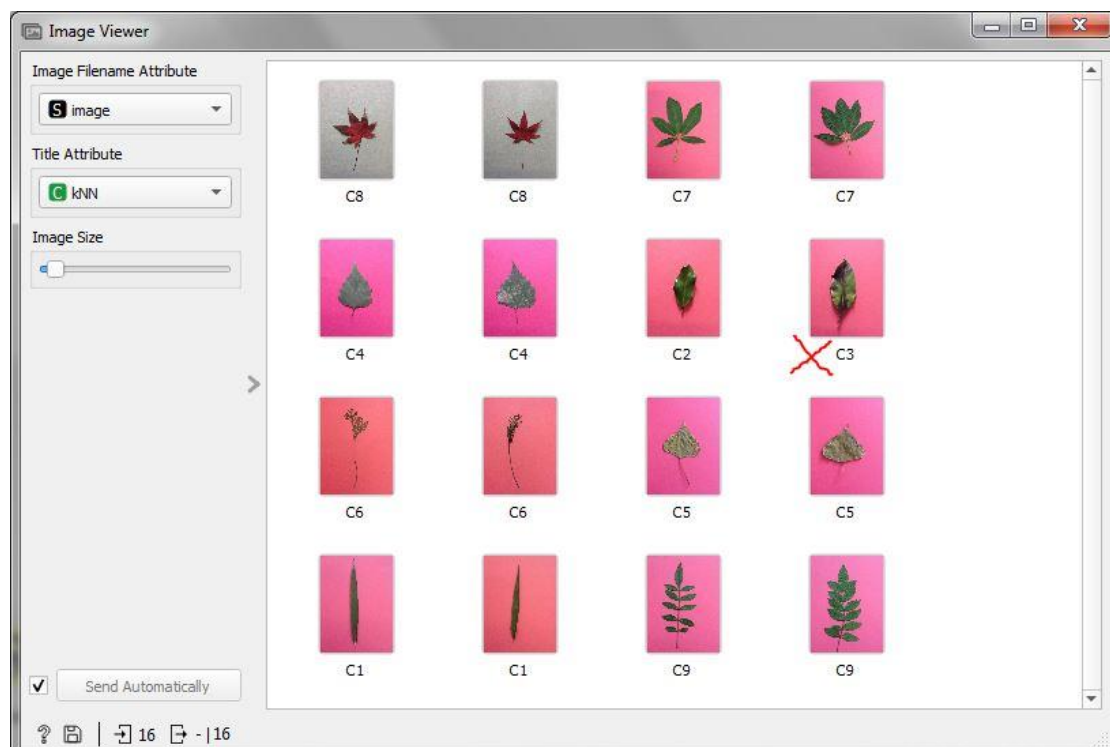
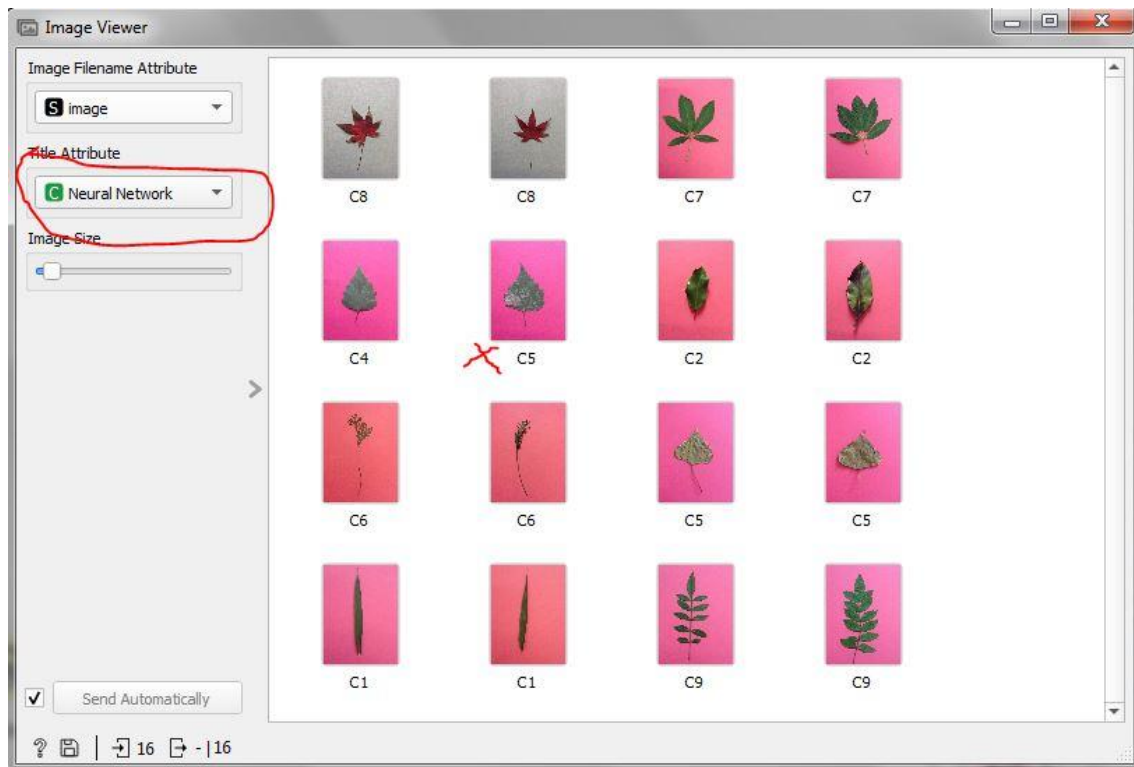
image name		image	size	width	height	kNN	kNN (C1)
hidden origin		ote/Desktop/AP5 image					
16	Pseudosasa Jap...	Pseudosasa Jap...	183827	720	960	C1	
15	Pseudosasa Jap...	Pseudosasa Jap...	189607	720	960	C1	
13	Ilex Aquifolium ...	Ilex Aquifolium ...	152396	720	960	C2	
14	Ilex Aquifolium ...	Ilex Aquifolium ...	152336	720	960	C3	
9	Betula Pubesce...	Betula Pubesce...	203050	720	960	C4	
12	Betula Pubesce...	Betula Pubesce...	174302	720	960	C4	
11	Populus Nigra (7)	Populus Nigra (...)	172508	720	960	C5	
10	Populus Nigra (6)	Populus Nigra (...)	174089	720	960	C5	
8	Papaver Sp (23)	Papaver Sp (23)...	189187	720	960	C6	
7	Papaver Sp (16)	Papaver Sp (16)...	191927	720	960	C6	
6	Aesculus Califo...	Aesculus Califo...	195487	720	960	C7	
5	Aesculus Califo...	Aesculus Califo...	186025	720	960	C7	
4	Acer Palmaturu...	Acer Palmaturu...	219124	720	960	C8	
3	Acer Palmaturu...	Acer Palmaturu...	201429	720	960	C8	
2	Schinus Terebi...	Schinus Terebi...	206145	720	960	C9	
1	Schinus Terebi...	Schinus Terebi...	211825	720	960	C9	



Embedded images		Embedded images					
hidden origin time	image name	image note/Desktop/AP5 image	size	width	height	neural Netwo	neural Networ
14	Pseudosasa Jap...	Pseudosasa Jap...	183827	720	960	C1	0.999
13	Pseudosasa Jap...	Pseudosasa Jap...	189607	720	960	C1	0.999
8	Ilex Aquifolium ...	Ilex Aquifolium ...	152336	720	960	C2	0.0104
7	Ilex Aquifolium ...	Ilex Aquifolium ...	152396	720	960	C2	3.23957
5	Betula Pubesce...	Betula Pubesce...	174302	720	960	C4	7.099
12	Populus Nigra (7)	Populus Nigra (...)	172508	720	960	C5	1.33651
11	Populus Nigra (6)	Populus Nigra (...)	174089	720	960	C5	1.04054
6	Betula Pubesce...	Betula Pubesce...	203050	720	960	C5	6.53598
10	Papaver Sp (23)	Papaver Sp (23)...	189187	720	960	C6	4.71607
9	Papaver Sp (16)	Papaver Sp (16)...	191927	720	960	C6	1.31296
4	Aesculus Califo...	Aesculus Califo...	195487	720	960	C7	4.01612
3	Aesculus Califo...	Aesculus Califo...	186025	720	960	C7	1.07549
2	Acer Palmaturu...	Acer Palmaturu...	219124	720	960	C8	4.07744
1	Acer Palmaturu...	Acer Palmaturu...	201429	720	960	C8	0.0063
16	Schinus Terebi...	Schinus Terebi...	206145	720	960	C9	3.25784
15	Schinus Terebi...	Schinus Terebi...	211825	720	960	C9	6.74029

E o resultado fora um agrupamento de imagens **87.5 %** correto com o Algoritmo "KNN" e **87.5 %** com Redes Neurais:





- **Conclusão**

Com isto, foi possível observar e atestar sobre a utilização de Algoritmos Conexionistas & Baseado em Instancias, no processo de identificação e agrupamento de imagens. Vale ressaltar que exclusivamente neste teste, os Algoritmos de Naive Bayes e Arvore de Decisões, não foram tão bem nos testes como os demais. Mas sabemos de sua importância e que podem (e devem) ser aplicados na Ciência de Dados, Aprendizado de Maquina e Computação em Geral.

Com base neste sistema, podemos ampliar ainda mais o foco e, abstraindo mais informação e mais "aprendizado", podemos gerar inúmeras possibilidades para serem testadas, que vão desde identificação de objetos, ate identificadores de rostos. Algo muitas vezes sonhado quando se inicia-se uma graduação como esta.

Acreditamos que assim como nossas maquinas podem aprender testando, este teste nos serviu para aprendermos ainda mais e darmos passos significativos em direção ao conhecimento.

## **Bibliografia**

<https://blog.gigasecurity.com.br/biometria-entenda-como-ela-revolucionou-a-seguranca-eletronica/>

<https://tecnoblog.net/273655/o-que-e-biometria-tecnologia/>

<https://computacaografica.ic.uff.br/transparenciasvol2cap6.pdf>

<https://minerandodados.com.br/machine-learning-na-pratica-knn-python/>

<https://didatica.tech/o-que-e-e-como-funciona-o-algoritmo-knn/>

<https://www.datageeks.com.br/naive-bayes/>

<https://www.digitalhouse.com/br/blog/naive-bayes>

<https://ichi.pro/pt/algoritmo-naive-bayes-101099747947170>

<https://blog.somostera.com/data-science/arvores-de-decisao>

<https://didatica.tech/como-funciona-o-algoritmo-arvore-de-decisao/>

<https://sites.icmc.usp.br/andre/research/neural/>

[https://www.sas.com/pt\\_br/insights/analytics/neural-networks.html](https://www.sas.com/pt_br/insights/analytics/neural-networks.html)

<https://medium.com/ensina-ai/machine-learning-sem-c%C3%B3digo-636d1a8f9081>

## **Repositório do Projeto**

<https://github.com/br-Yan1993/APS-6-PDI-ML>