**Universidade Federal do Ceará**

**Campus Sobral**

**Curso de Engenharia de Computação**

**FRANCISCO RODRIGO ROCHA**

**Matrícula - 473962**

**RELATÓRIO FINAL DO TRABALHO PRÁTICO**

**Introdução ao Scikit-Learn - Python (individual)**

**Sobral-CE**

**2019**

**SUMÁRIO**

[PARTE 1 3](#_Toc25673677)

[1.1 Introdução 3](#_Toc25673678)

[1.2 Conjunto de dados utilizados 3](#_Toc25673679)

[1.3 Algoritmos Utilizados 4](#_Toc25673680)

[1.4 Metodologia 4](#_Toc25673681)

[1.5 Resultados Obtidos 4](#_Toc25673682)

[1.6 *Link* para o código-fonte no *GitHub* 6](#_Toc25673683)

[PARTE 2 7](#_Toc25673684)

[2.1 Introdução 7](#_Toc25673685)

[2.2 Conjunto de dados utilizados 7](#_Toc25673686)

[2.3 Algoritmos Utilizados 8](#_Toc25673687)

[2.4 Metodologia 8](#_Toc25673688)

[2.5 Resultados Obtidos 8](#_Toc25673689)

[2.6 *Link* para o código-fonte no *GitHub* 10](#_Toc25673690)

[REFERÊNCIAS 10](#_Toc25673691)

# PARTE 1

## Introdução

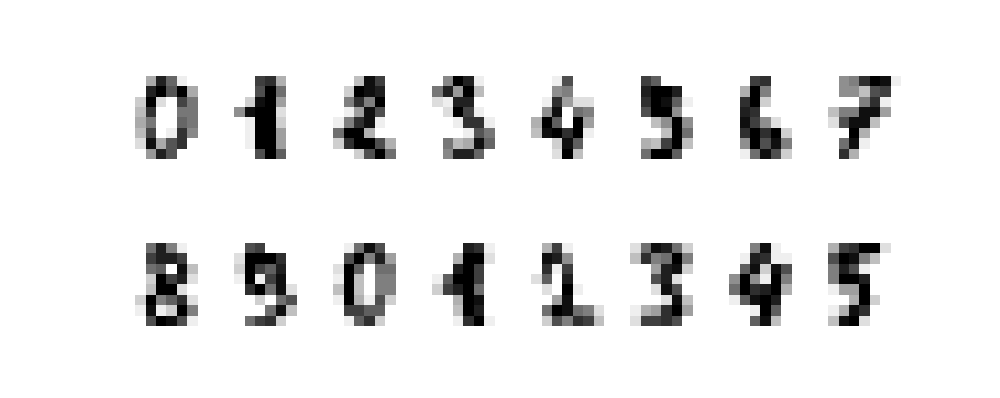
Na primeira parte do trabalho, é feito um tutorial [Ref. 1] onde o centro do estudo é o Machine Learning e a bibioteca utilizada é a *Scikit-learn*, nessa parte entenderemos como funciona o apredizado e como são feitas as predições dos modelos criados por essa biblioteca. *Machine Learnig*, ou aprendizado de máquina, é uma área da ciência da computação onde estuda-se formas com que os computadores tomem uma decisão de acordo com a situação e com as informações que foram inseridas neles. Nessa parte será utilizada a base de dados *sklearn.datasets.load\_digits* onde os dados são imagens de 8x8 pixels e o modelo reconhece q número de 0 a 9 representa a imagem.

## 1.2 Conjunto de dados utilizados

O conjunto de dados utilizados é um conjunto ja incluído na biblioteca, *sklearn.datasets.load\_digits.*

Consiste em imagens de 8x8 pixels de números com uma resolução baixa. Seque alguns exemplos:

**Figura 1.1** – Exemplos dos números dos dados utilizados



[Ref. 2]

## 1.3 Algoritmos Utilizados

Support Vector Machines – SVM - sklearn.svm.SVC

KneighborsClassifier - sklearn.neighbors.KNeighborsClassifier

DecisionTreeClassifier - sklearn.tree.DecisionTreeClassifier

## 1.4 Metodologia

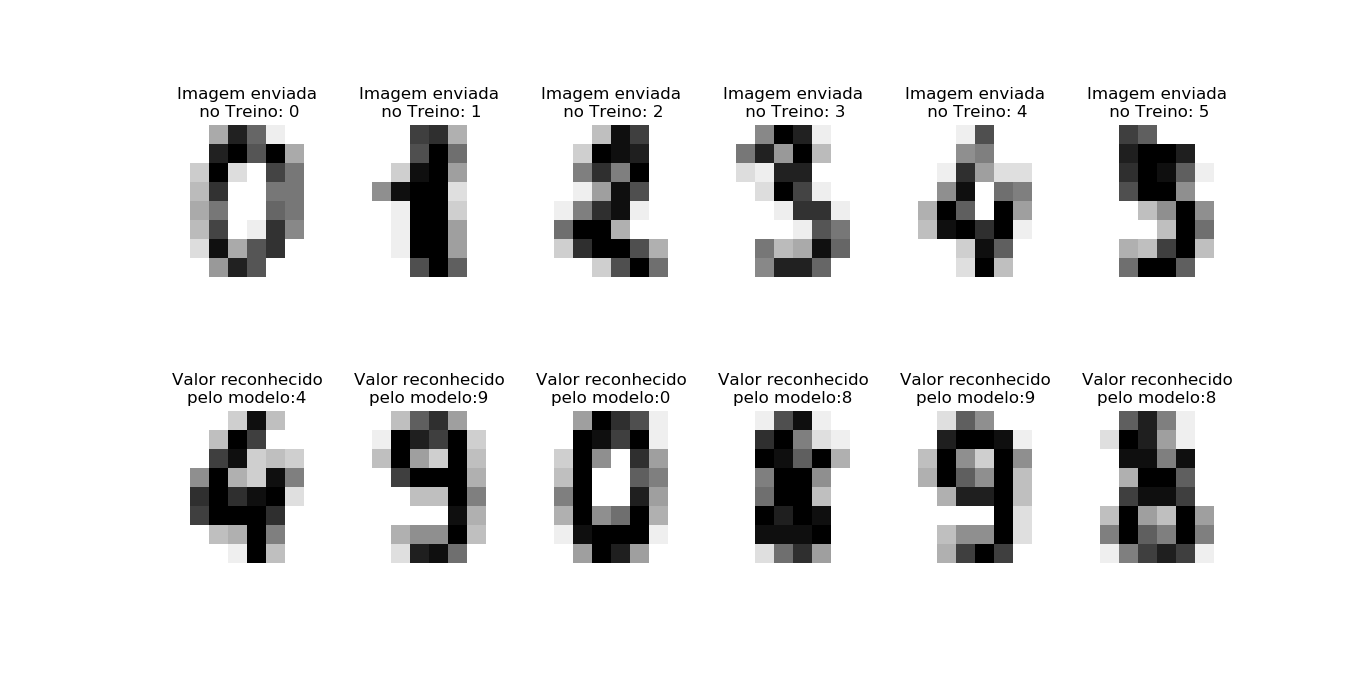
As entradas são imagens de 8x8 pixels, e as saídas são números de 0 a 9.

Para executar a predição precisamos, primeiramente, fazer o treinamento. Para a realização do treinamento foi selecionada a primeira metade dos dados dos dígitos e seus valores correspondentes (números de 0 a 9) como parâmetros, e na realização dos testes foram utilizadas a segunda metade dos dados.

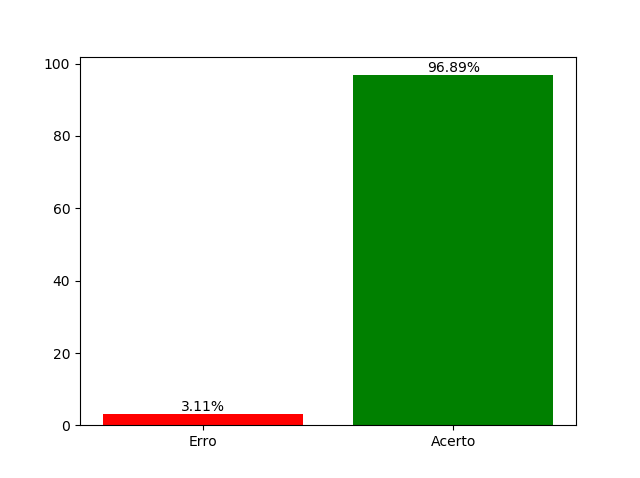
## 1.5 Resultados Obtidos

Support Vector Machines(SVM) - sklearn.svm.SVC:

**Figura 1.2** – Exemplos de imagens de treino e teste com classificador SVM

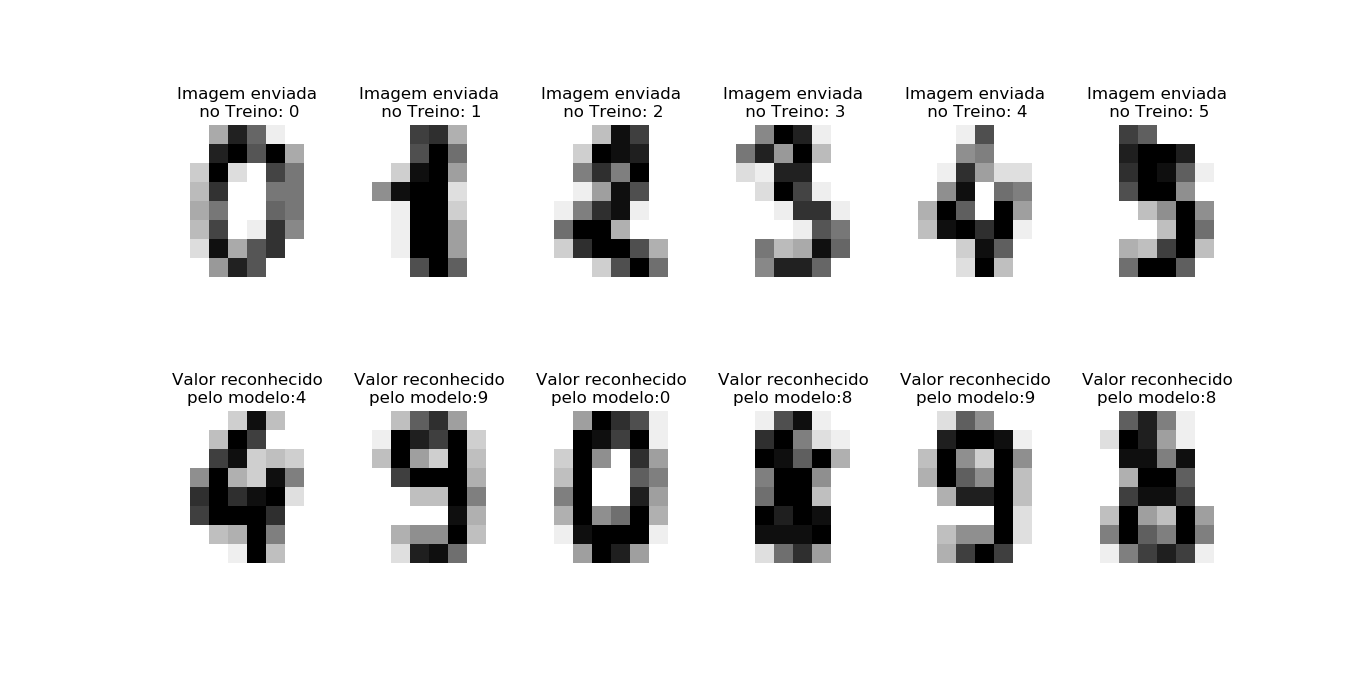


**Figura 1.3** – porcentagem de acertos e erros com classificador SVM

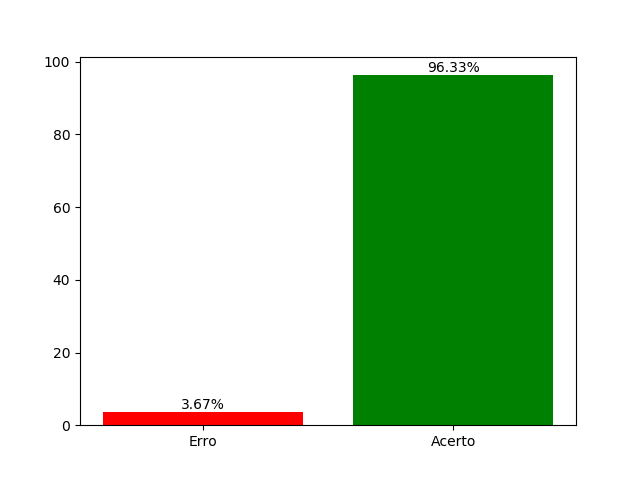


KneighborsClassifier - sklearn.neighbors.KneighborsClassifier:

**Figura 1.4** – Exemplos de imagens de treino e teste com classificador KneighborsClassifier

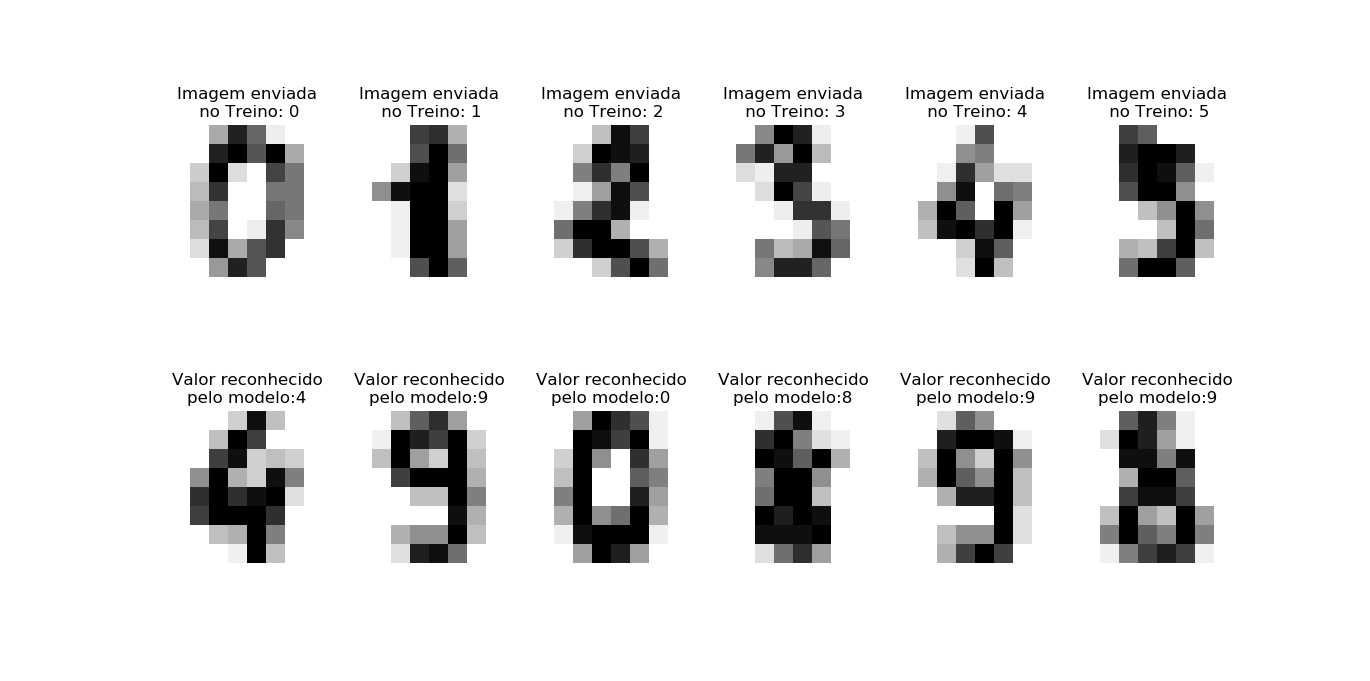


**Figura 1.5** – porcentagem de acertos e erros com classificador KneighborsClassifier

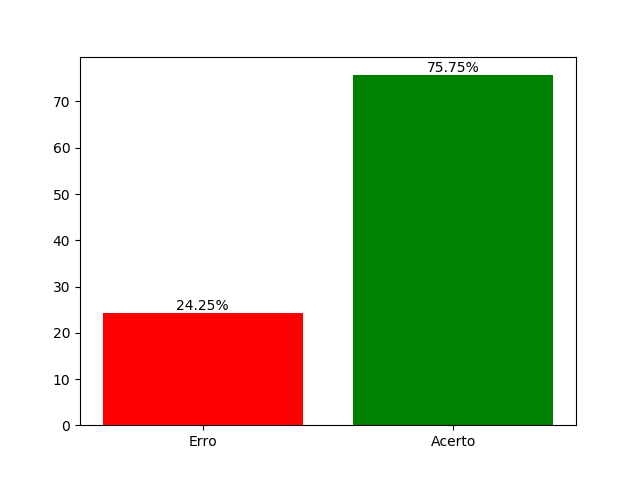


DecisionTreeClassifier - sklearn.tree.DecisionTreeClassifier:

**Figura 1.6** – Exemplos de imagens de treino e teste com classificador DecisionTreeClassifier

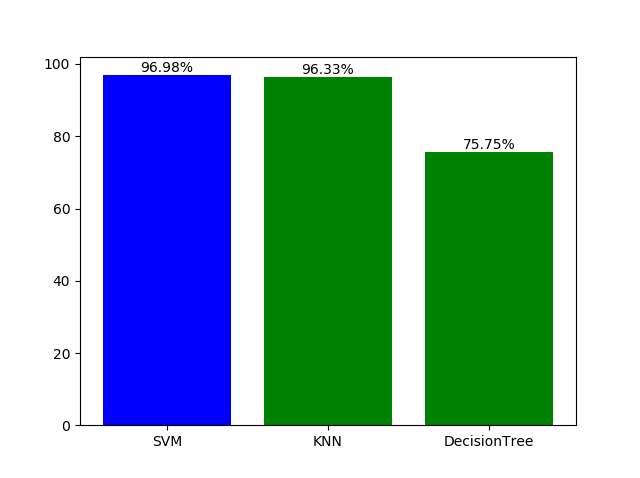


**Figura 1.7** – porcentagem de acertos e erros com classificador DecisionTreeClassifier



Comparação entre os três algoritmos:

**Figura 1.8** – Comparação entre acertos dos classificadores utilizados



[Ref. 2]

## 1.6 *Link* para o código-fonte no *GitHub*

<https://github.com/rodrocha444/trabalho/tree/master/parte1>

# PARTE 2

## 2.1 Introdução

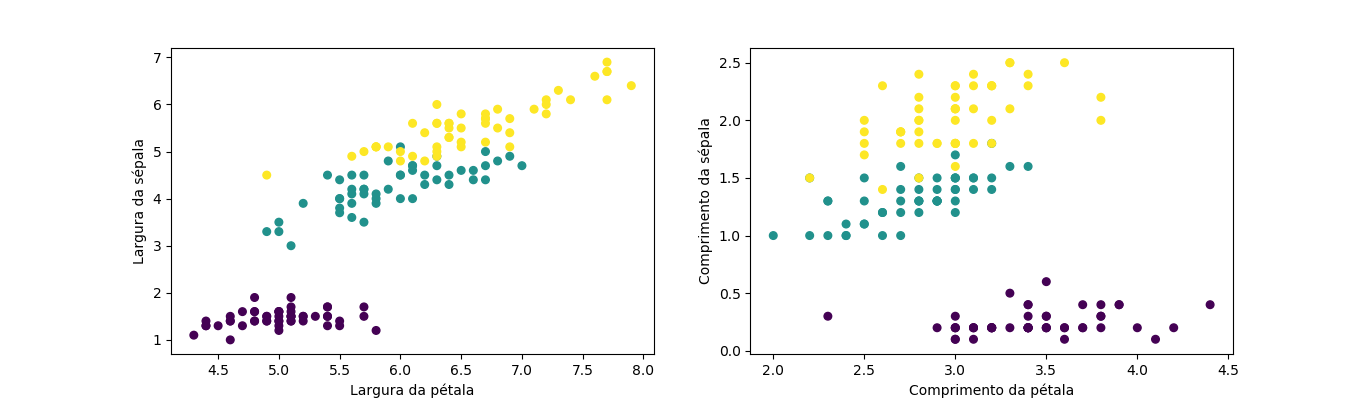
Na segunda parte do trabalho, é feito um código onde são utilizados 4 algoritmos de *Machine Learning* da biblioteca *Scikit-learn.* Nessa parte usaremos os métodos aprendidos na PARTE 1 para desenvolvermos os algoritmos para a predição de uma base de dados diferente. Machine Learnig, ou aprendizado de máquina, é uma área da ciência da computação onde visa o objetico é que os computadores tomem uma decisão de acordo com a situação e com as informações que foram inseridas nele. Será utilizada a base de dados *sklearn.datasets.load\_iris* onde os dados são largura e comprimento das pétalas e sépalas e o modelo reconhece q tipo de iris representa a imagem (0,1,2).

## 2.2 Conjunto de dados utilizados

O conjunto de dados utilizados é um conjunto ja incluído na biblioteca, *sklearn.datasets.load\_iris.*

Consiste em 4 dados q correspondem a largura e comprimento das sépalas e pétalas. Seque os dados distribuídos em gráficos, onde os tipos de iris são distinguíveis por cor:

**Figura 2.1** – Distribuição das iris



[Ref. 3]

## 2.3 Algoritmos Utilizados

Support Vector Machines – SVM - sklearn.svm.SVC

KneighborsClassifier - sklearn.neighbors.KNeighborsClassifier

DecisionTreeClassifier - sklearn.tree.DecisionTreeClassifier

AdaBoostClassifier - sklearn.ensemble.AdaBoostClassifier

## 2.4 Metodologia

As entradas são 4 dados, comprimento e largura da sépala e da pétala, e as saídas são números de 0 a 2, q representam os 3 tipos de iris classificáveis.

Para executar a predição precisamos, primeiramente, fazer o treinamento.

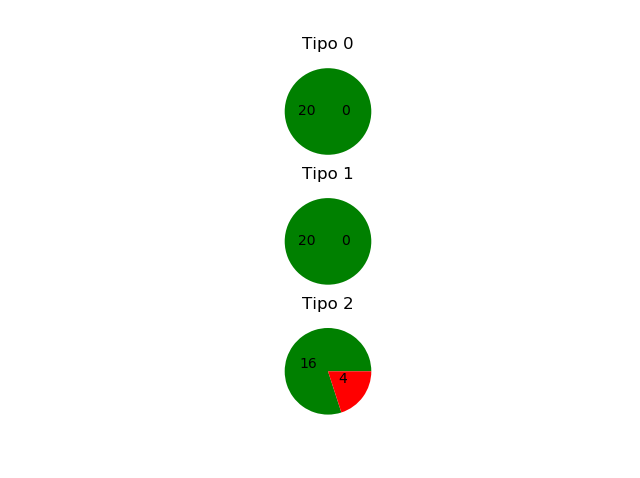
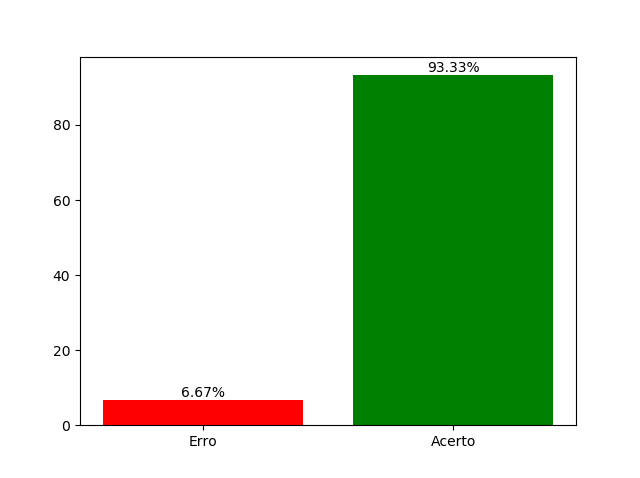
OBSERVAÇÃO: os dados das iris(150 no total) estão arranjados da seguinte forma, as 50 primeiras amostras sao do tipo 0, as outras 50 seguintes são do tipo 1 e as 50 últimas são do tipo 2.

Para a realização do treinamento foi selecionada 30 amostras de cada tipo de iris e seus valores correspondentes (números de 0 a 2) como parâmetros, e na realização dos testes foram utilizadas as 20 amostras restantes de cada tipo dos dados.

## 2.5 Resultados Obtidos

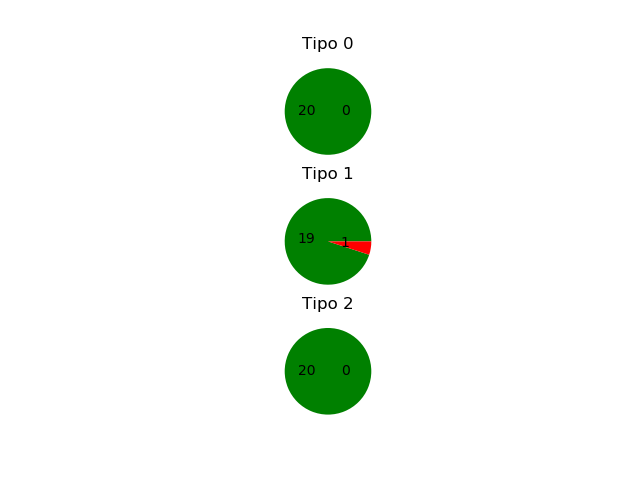
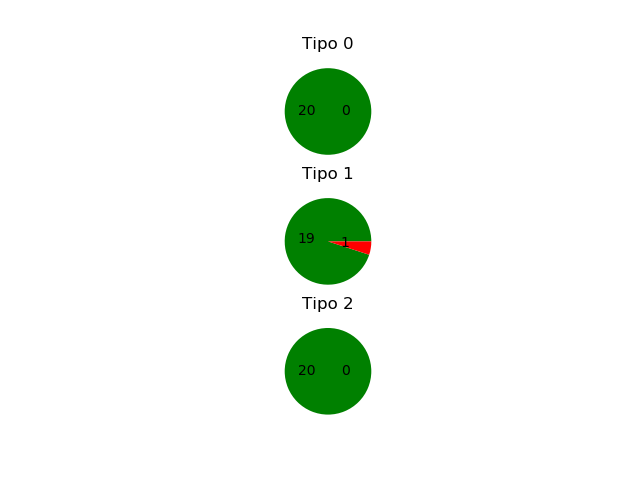
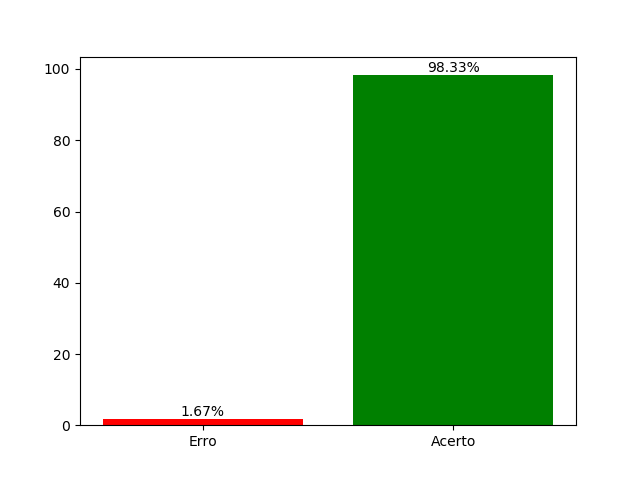
Support Vector Machines(SVM) - sklearn.svm.SVC:

**Figura 2.2** - Porcentagem e distribuição de acertos e erros com classificador sklearn.svm.SVC



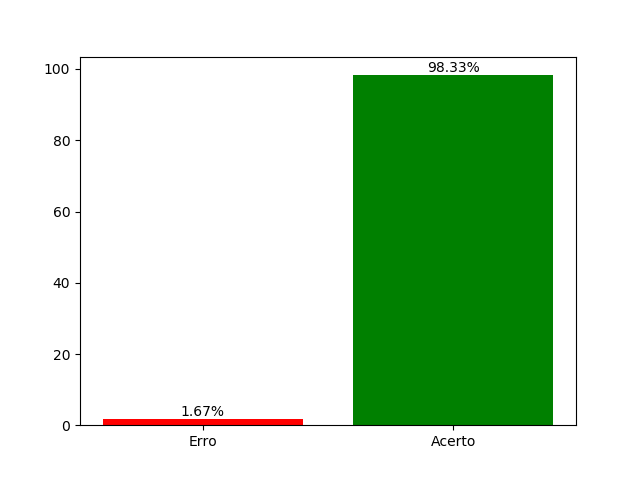
KneighborsClassifier - sklearn.neighbors.KneighborsClassifier:

**Figura 2.3** - Porcentagem e distribuição de acertos e erros com classificador KneighborsClassifier



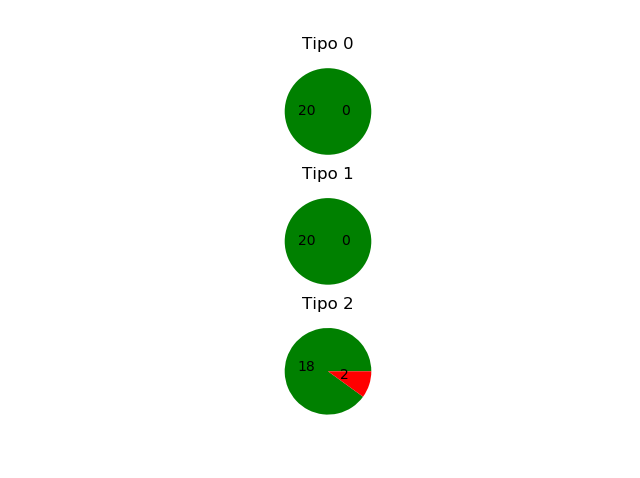
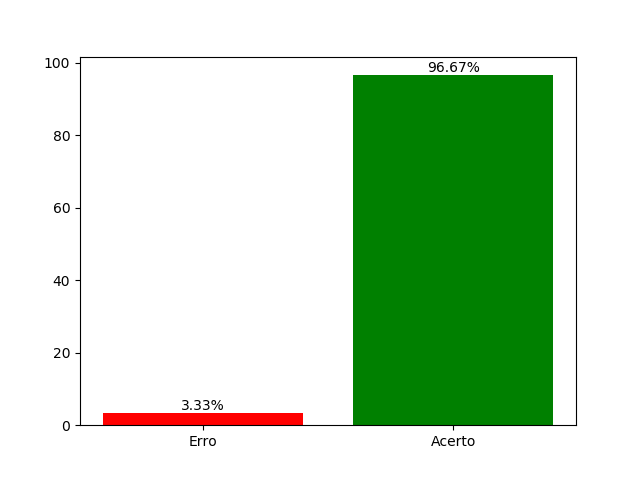
DecisionTreeClassifier - sklearn.tree.DecisionTreeClassifier:

**Figura 2.4** - Porcentagem e distribuição de acertos e erros com classificador DecisionTreeClassifier



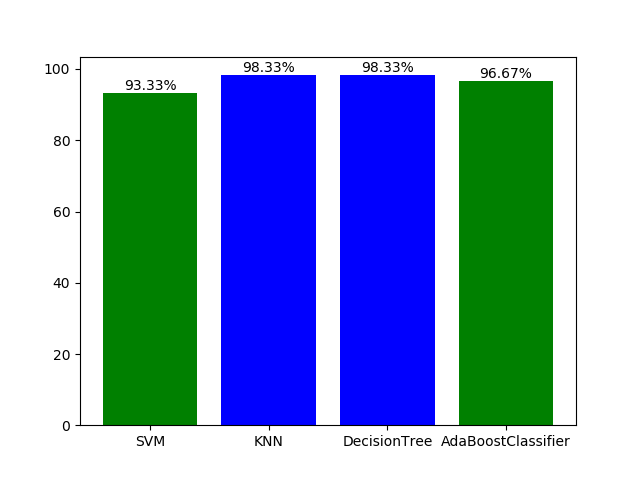
AdaBoostClassifier - sklearn.ensemble.AdaBoostClassifier:

**Figura 2.5** - Porcentagem e distribuição de acertos e erros com classificador AdaBoostClassifier



Comparação entre os quatro algoritmos:

**Figura 2.6** – Comparação entre acertos dos classificadores utilizados



## 2.6 *Link* para o código-fonte no *GitHub*

<https://github.com/rodrocha444/trabalho/tree/master/parte2>

# REFERÊNCIAS

[Ref. 1] Tutorial: <https://scikit-learn.org/stable/tutorial/basic/tutorial.html>

[Ref. 2] Figura 1.1: <https://github.com/rodrocha444/trabalho/blob/master/parte1/grafico_adicional1.py> [Ref. 3] Figura 1.8: <https://github.com/rodrocha444/trabalho/blob/master/parte1/grafico_adicional2.py>

[Ref. 3] Figura 2.1: <https://github.com/rodrocha444/trabalho/blob/master/parte2/grafico_adicional1.py> [Ref. 3] Figura 2.6: <https://github.com/rodrocha444/trabalho/blob/master/parte2/grafico_adicional2.py>