UNIVERSIDADE DO ESTADO DO AMAZONAS

Escola Superior de Tecnologia

Processo seletivo para projeto P&D UEA/IA

# Projeto de Machine Learning para classificação de espécies de Íris (flor)

Autora:

- Karoline Santos Pereira

Avaliadores:

- Rodrigo Choji de Freitas
- Fabio Santos
- Ramayana Júnior
- Tiago Ramos de Sá

MANAUS / AM

23/05/2022

# Sumário

## 1. Introdução

Nesse projeto de *machine learning (*aprendizagem de máquinas) foi apresentado um método de classificação de espécies de flor de Íris, no qual se utiliza uma base de dados com a coleta de três tipos de espécies, e suas determinadas medições de comprimento e largura. Sendo possível determinar a classificação de variante por meio das medidas de sua sépala e pétala, classificando-as comumente como setosa, versicolor ou virginica (Figura 1).

*Figura 1: Classificação de espécie das Flores de Íris*


*Fonte:https://pt.wikipedia.org/wiki/Conjunto_de_dados_flor_Iris#/media/Ficheiro:Flores_de_%C3%8Dris.png*

Para classificação de dados presentes nesta base foi utilizada o *scikit-learn* (biblioteca de aprendizagem de máquina em código aberto) no Python, onde será feita a análise exploratória e tratamento de dados, por meio de alguns processos, como o treinamento de modelo por meio de Árvore de Decisão e métricas de validação do código por meio da execução de predições e avaliação de acurácia do modelo.

A fim de criar um modelo de tomada de decisão, foi utilizado o algoritmo *RandomForest* (Floresta aleatória), o qual cria uma variedade de árvores de decisão, com o intuito de formar uma sequência de etapas para verificação de uma condição, sendo a mesma atendida, segue para outra etapa, senão, para outra decisão, chegando após diversos caminhos a um resultado de análise/classificação final.

A execução da análise preditiva foi realizada como forma de “previsão” de uma possível classificação de espécies em dados de teste, e posteriormente aplicada a avaliação de acurácia

destas predições, que seria o quão preciso foi determinado dado. A partir destes processos foi possível realizar uma análise exploratória e tratamento de dados, e logo após uma definição e validação do modelo de classificação de espécies.

## 2. Descrição da construção do modelo

Inicialmente foi realizada a importação do dataset íris pro modelo (Figura 2), o qual apresentou uma base de 150 dados, os "*targets"* que seriam os valores padrões que devem ser apresentados, os quais definem a classificação dessas espécies (Figura 3) e em seguida os "*target_names"* que definem o valor 0 para setosa, 1 para versicolor e 2 para virginica, e finalmente os "*feature_names"* que indicam o que são as variáveis para cada coluna, contendo: comprimento de sépala, largura de sépala, comprimento de pétala e largura de pétala (Figura 4).

*Figura 2:Import de dataset íris*

```
In [2]: from sklearn.datasets import load_iris
        retorno = load_iris()

In [3]: retorno

Out[3]: {'data': array([[5.1, 3.5, 1.4, 0.2],
                        [4.9, 3. , 1.4, 0.2],
                        [4.7, 3.2, 1.3, 0.2],
                        [4.6, 3.1, 1.5, 0.2],
                        [5. , 3.6, 1.4, 0.2],
                        [5.4, 3.9, 1.7, 0.4],
                        [4.6, 3.4, 1.4, 0.3],
                        [5. , 3.4, 1.5, 0.2],
                        [4.4, 2.9, 1.4, 0.2],
                        [4.9, 3.1, 1.5, 0.1],
                        [5.4, 3.7, 1.5, 0.2],
                        [4.8, 3.4, 1.6, 0.2],
                        [4.8, 3. , 1.4, 0.1],
                        [4.3, 3. , 1.1, 0.1],
                        [5.8, 4. , 1.2, 0.2],
                        [5.7, 4.4, 1.5, 0.4],
                        [5.4, 3.9, 1.3, 0.4],
                        [5.1, 3.5, 1.4, 0.3],
                        [5.7, 3.8, 1.7, 0.3],
```

*Figura 3:targets de classificação;*

```
'target': array([0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
       0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
       0, 0, 0, 0, 0, 0, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,
       1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,
       1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2,
       2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2,
       2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2]),
'frame': None,
'target_names': array(['setosa', 'versicolor', 'virginica'], dtype='<U10'),
```

*Figura 4: Indicação de variáveis de acordo com as colunas de dados.*

```
'feature_names': ['sepal length (cm)'
 'sepal width (cm)',
 'petal length (cm)',
 'petal width (cm)'],
```

Com o intuito de preparar os dados, realizou-se a classificação de dados entre setosa, versicolor e virginica, e em seguida inserido o conjunto de dados em um DataFrame, indicando o nome da variável para cada coluna, e adicionado as colunas de *targer e target_name.* (Figura 5), a partir disso, foi apresentado uma tabela com dados de medições e classificações (Figura 6) que dará início à análise exploratória.

*Figura 5:Preparção de dados para classificação*

```
In [4]: def defineNames(target):
            if(target == 0):
               return 'setosa'
            elif(target == 1):
                return 'versicolor'
            elif(target == 2):
                 return 'virginica'
```

```
In [5]: iris = pd.DataFrame(retorno.data)
        iris.columns = ['comp sepal','larg sepal','comp petal','larg petal']
```

```
In [6]: iris['target'] = retorno.target
        iris['target_name'] = iris['target'].apply(defineNames)
```

*Figura 6:Tabela para análise de dados de medição e classificação*

```
In [7]: display(iris)
```

| | comp sepal | larg sepal | comp petal | larg petal | target | target_name |
|---|---|---|---|---|---|---|
| **0** | 5.1 | 3.5 | 1.4 | 0.2 | 0 | setosa |
| **1** | 4.9 | 3.0 | 1.4 | 0.2 | 0 | setosa |
| **2** | 4.7 | 3.2 | 1.3 | 0.2 | 0 | setosa |
| **3** | 4.6 | 3.1 | 1.5 | 0.2 | 0 | setosa |
| **4** | 5.0 | 3.6 | 1.4 | 0.2 | 0 | setosa |
| **...** | ... | ... | ... | ... | ... | ... |
| **145** | 6.7 | 3.0 | 5.2 | 2.3 | 2 | virginica |
| **146** | 6.3 | 2.5 | 5.0 | 1.9 | 2 | virginica |
| **147** | 6.5 | 3.0 | 5.2 | 2.0 | 2 | virginica |
| **148** | 6.2 | 3.4 | 5.4 | 2.3 | 2 | virginica |
| **149** | 5.9 | 3.0 | 5.1 | 1.8 | 2 | virginica |

150 rows × 6 columns

A análise exploratória se iniciou com a obtenção de dados estáticos presente no dataset, como a média, valor máximo, valor mínima, quartis e demais dados que possibilitam um maior conhecimento do que esperar em relação aos dados, e uma melhor análise crítica caso sejam encontrados valores muito fora desses padrões (Figura 7). Assim, para melhor visualização, foi gerado um gráfico boxplot que possui uma melhor e mais clara explanação sobre esses dados (Figura 8), com a observação das diferenças entre cada variável de medição, como também, a presença de outliers (valores extremos) no caso da largura da sépala, ao verificar esses outliers, foi apresentado 4 dados fora do padrão, sendo três deles da espécie setosa, e um versicolor (Figura 9).

*Figura 7:dados estatísticos do dataset íris.*

```
In [8]: iris.describe()
```

Out[8]:

| | comp sepal | larg sepal | comp petal | larg petal | target |
|---|---|---|---|---|---|
| **count** | 150.000000 | 150.000000 | 150.000000 | 150.000000 | 150.000000 |
| **mean** | 5.843333 | 3.057333 | 3.758000 | 1.199333 | 1.000000 |
| **std** | 0.828066 | 0.435866 | 1.765298 | 0.762238 | 0.819232 |
| **min** | 4.300000 | 2.000000 | 1.000000 | 0.100000 | 0.000000 |
| **25%** | 5.100000 | 2.800000 | 1.600000 | 0.300000 | 0.000000 |
| **50%** | 5.800000 | 3.000000 | 4.350000 | 1.300000 | 1.000000 |
| **75%** | 6.400000 | 3.300000 | 5.100000 | 1.800000 | 2.000000 |
| **max** | 7.900000 | 4.400000 | 6.900000 | 2.500000 | 2.000000 |

*Figura 8: dados estatísticos para cada variável de medição.*

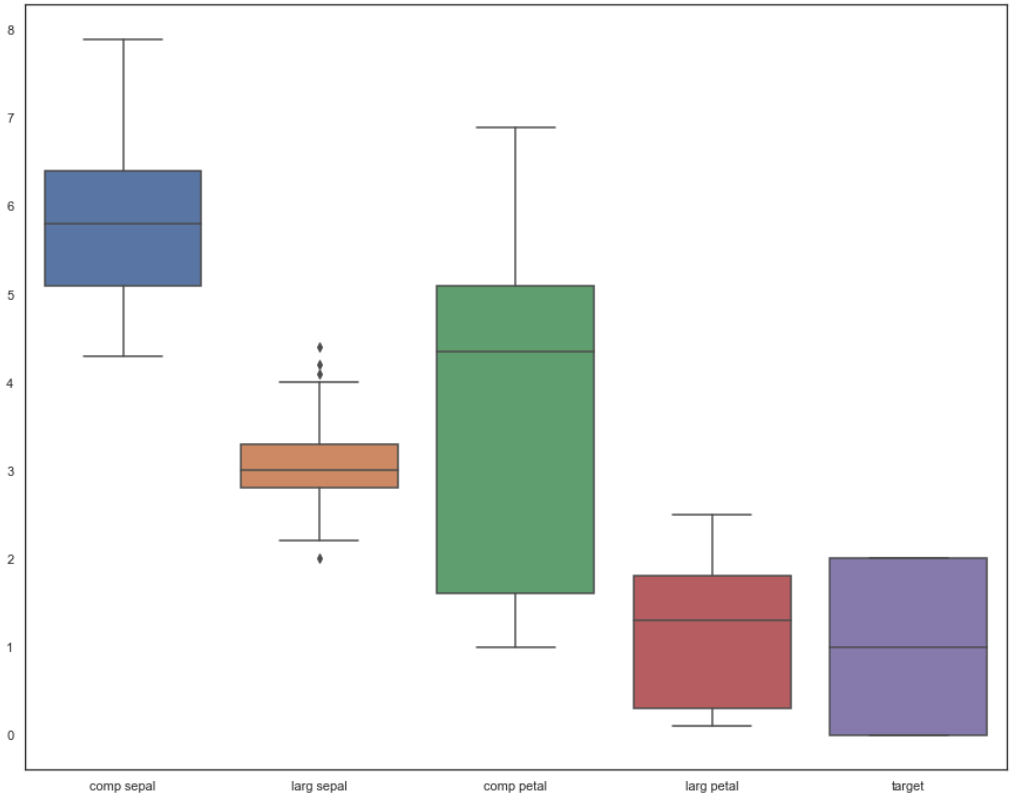



*Figura 9: Verificação da presença de outliers.*

```
In [12]: #verificar outliers
         iris[(iris['larg sepal'] <2.2) | (iris['larg sepal'] > 4)]
```

Out[12]:

| | comp sepal | larg sepal | comp petal | larg petal | target | target_name |
|---|---|---|---|---|---|---|
| **15** | 5.7 | 4.4 | 1.5 | 0.4 | 0 | setosa |
| **32** | 5.2 | 4.1 | 1.5 | 0.1 | 0 | setosa |
| **33** | 5.5 | 4.2 | 1.4 | 0.2 | 0 | setosa |
| **60** | 5.0 | 2.0 | 3.5 | 1.0 | 1 | versicolor |

Subsequente a isso, foi feita a verificação da presença de dados nulos (Figura 10), sendo observado a inexistência deles no conjunto de dados, e em sequência, a correlação entre as variáveis de medição (Figura 11), classificando em:

*Tabela 1: Classificação de correlação entre variáveis.*

| **Classificação** | **Tipo** | **Comportamento** |
|---|---|---|
| 1 | Correlação Forte | Proporcional |
| 0 | Correlação Nula | Nulo |
| -1 | Correlação Forte | Inverso |

*Figura 10: Verificação de dados nulos.*

```
In [11]: #Verificar a presença de dados nulos
         iris.info()

<class 'pandas.core.frame.DataFrame'>
RangeIndex: 150 entries, 0 to 149
Data columns (total 6 columns):
 #   Column       Non-Null Count  Dtype
---  ------       --------------  -----
 0   comp sepal   150 non-null    float64
 1   larg sepal   150 non-null    float64
 2   comp petal   150 non-null    float64
 3   larg petal   150 non-null    float64
 4   target       150 non-null    int32
 5   target_name  150 non-null    object
dtypes: float64(4), int32(1), object(1)
memory usage: 6.6+ KB
```

*Figura 11: Correlação entre variáveis.*

```
In [13]: #verificar a correlação entre variáveis
         #corr=1 (correlação forte) corr=-1 (crescimento inverso) corr=0 (correlação fraca)
         iris.corr()
```

Out[13]:

| | comp sepal | larg sepal | comp petal | larg petal | target |
|---|---|---|---|---|---|
| comp sepal | 1.000000 | -0.117570 | 0.871754 | 0.817941 | 0.782561 |
| larg sepal | -0.117570 | 1.000000 | -0.428440 | -0.366126 | -0.426658 |
| comp petal | 0.871754 | -0.428440 | 1.000000 | 0.962865 | 0.949035 |
| larg petal | 0.817941 | -0.366126 | 0.962865 | 1.000000 | 0.956547 |
| target | 0.782561 | -0.426658 | 0.949035 | 0.956547 | 1.000000 |

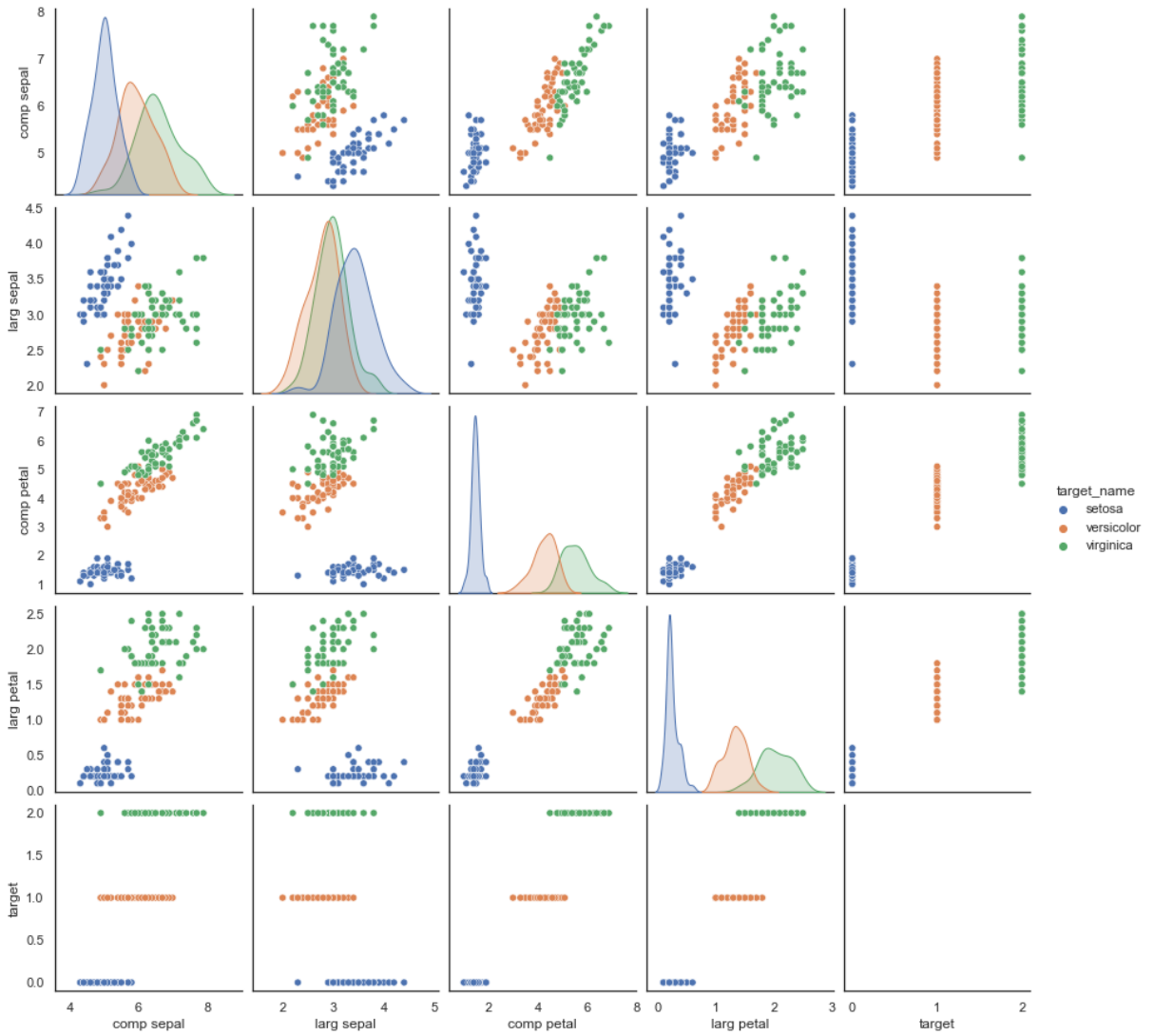
Por fim, seguindo os passos de análise exploratória de dados, tornou-se viável a separação de dados em treino e teste para criação do modelo de machine learning.

## 3. Resultados

### 3.1. Correlação entre variáveis de medição.

A correlação de dados para as variáveis medidas (Figura 12) apresentou uma correlações muito fortes entre medidas iguais, como quando se analisa largura de sépala em x com largura de sépala em y, além destas, observou-se expressiva correlação >0,95 para os casos entre largura e comprimento de pétala, onde é visto uma facilidade de separação entre as espécies, em que a setosa fica isolada em um extremo em relação a versicolor e virginica. Para casos de correlações fortes, porém inversas, as correlações entre comprimento de sépala e largura de sépala, comprimento de pétala e largura de sépala, e largura de pétala e largura de sépala, definem bem a inversão entre ambas, quando uma cresce e a outra decresce na mesma proporção. Para estes dados de medidas foi inexistente as correlações nulas, mostrando que são dados de correlações muito fortes, mesmo que separados pela proporcionalidade ou inversão dos mesmos.

*Figura 12: Correlação entre comprimento e largura para sépala e pétala.*




## 3.2. Separação de dados em treino e teste.

### 3.2.1. Dados de treino.

Ao separar os dados em treino e teste, criou-se uma base de treino (Figura 13) e plotou-se um gráfico para análise destes dados, na qual se percebeu uma clara divisão da setosa para a versicolor e virginica (Figura 14). Para propósito de divisão de targets, foi realizado um algoritmo para delitimitação dos pontos para um modelo de classificação (Figura 15), o qual traçou duas retas que separam as espécies uma da outra (Figura 16).

*Figura 13: separação de dados de treino.*

```
In [19]: treino = pd.DataFrame(x_treino)
         treino.columns = ['comp sepal', 'larg sepal', 'comp petal', 'larg petal']
         treino['target'] = y_treino

In [20]: display(treino)
```

| | comp sepal | larg sepal | comp petal | larg petal | target |
|---|---|---|---|---|---|
| **0** | 6.5 | 2.8 | 4.6 | 1.5 | 1 |
| **1** | 6.7 | 2.5 | 5.8 | 1.8 | 2 |
| **2** | 6.8 | 3.0 | 5.5 | 2.1 | 2 |
| **3** | 5.1 | 3.5 | 1.4 | 0.3 | 0 |
| **4** | 6.0 | 2.2 | 5.0 | 1.5 | 2 |
| **...** | ... | ... | ... | ... | ... |
| **107** | 6.3 | 2.8 | 5.1 | 1.5 | 2 |
| **108** | 6.4 | 3.1 | 5.5 | 1.8 | 2 |
| **109** | 6.3 | 2.5 | 4.9 | 1.5 | 1 |
| **110** | 6.7 | 3.1 | 5.6 | 2.4 | 2 |
| **111** | 4.9 | 3.6 | 1.4 | 0.1 | 0 |

112 rows × 5 columns

*Figura 14: Comportamento de espécies quanto as medições de comprimento e largura.*

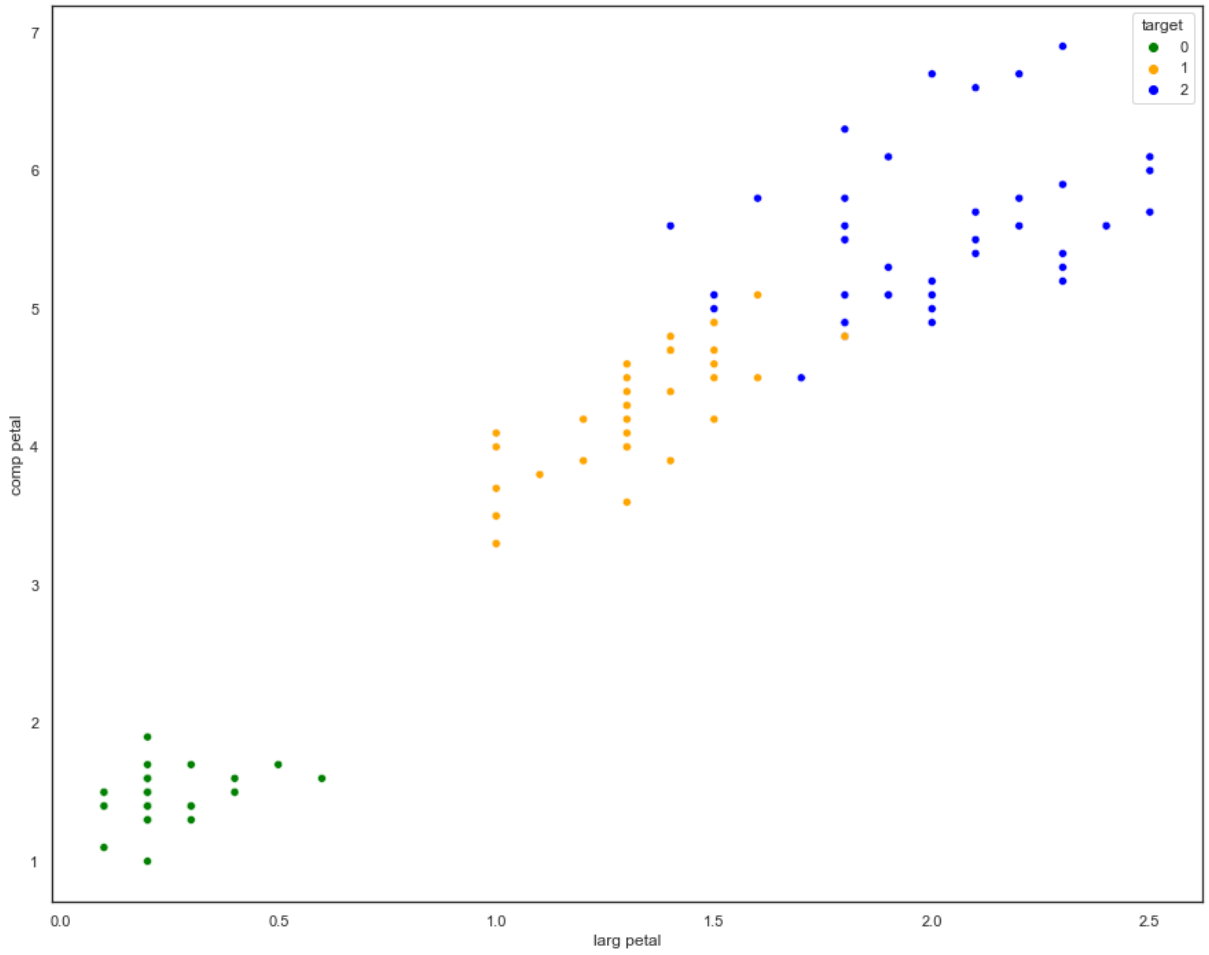


*Figura 15: algoritmo para separação de targets.*

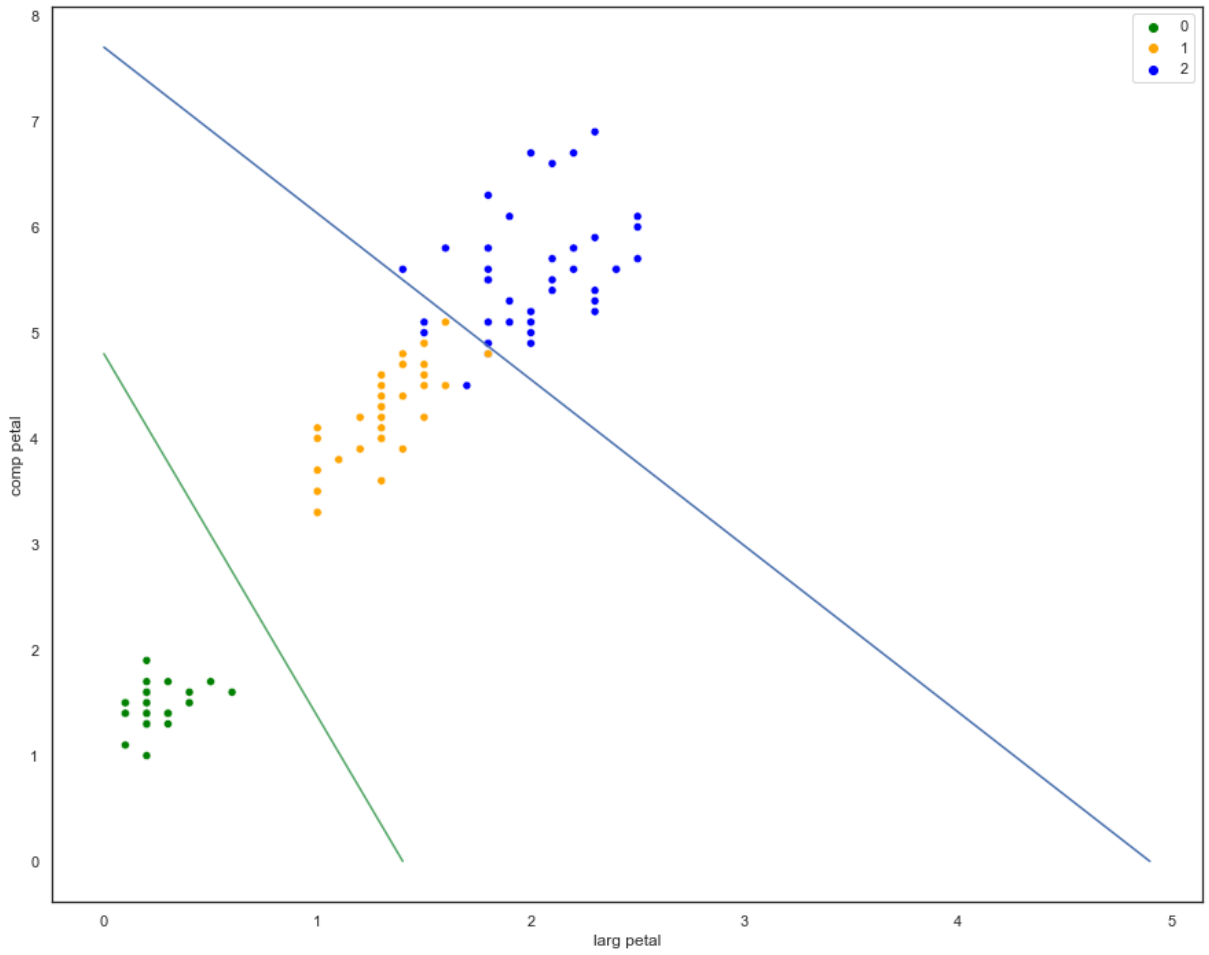
```
In [22]: #separar os targets
         treino[(treino['target']== 2) & ((treino['larg petal'] == 1.7) | (treino['larg petal']==1.5))]
```

Out[22]:

| | comp sepal | larg sepal | comp petal | larg petal | target |
|---|---|---|---|---|---|
| **4** | 6.0 | 2.2 | 5.0 | 1.5 | 2 |
| **81** | 4.9 | 2.5 | 4.5 | 1.7 | 2 |
| **107** | 6.3 | 2.8 | 5.1 | 1.5 | 2 |

```
In [23]: #traçando duas retas
         sns.scatterplot(x='larg petal',y='comp petal',data=treino, hue='target', palette=['green', 'orange','blue'])
         x_verde = [0,1.4]
         y_verde = [4.8,0]
         sns.lineplot(x=x_verde, y=y_verde,color='g')
         x_azul = [0,4.9]
         y_azul = [7.7,0]
         sns.lineplot(x=x_azul, y=y_azul, color='b')
         plt.show()
```

*Figura 16: Delimitação de espécies: dados de treino.*




### 3.2.2. Dados de teste.

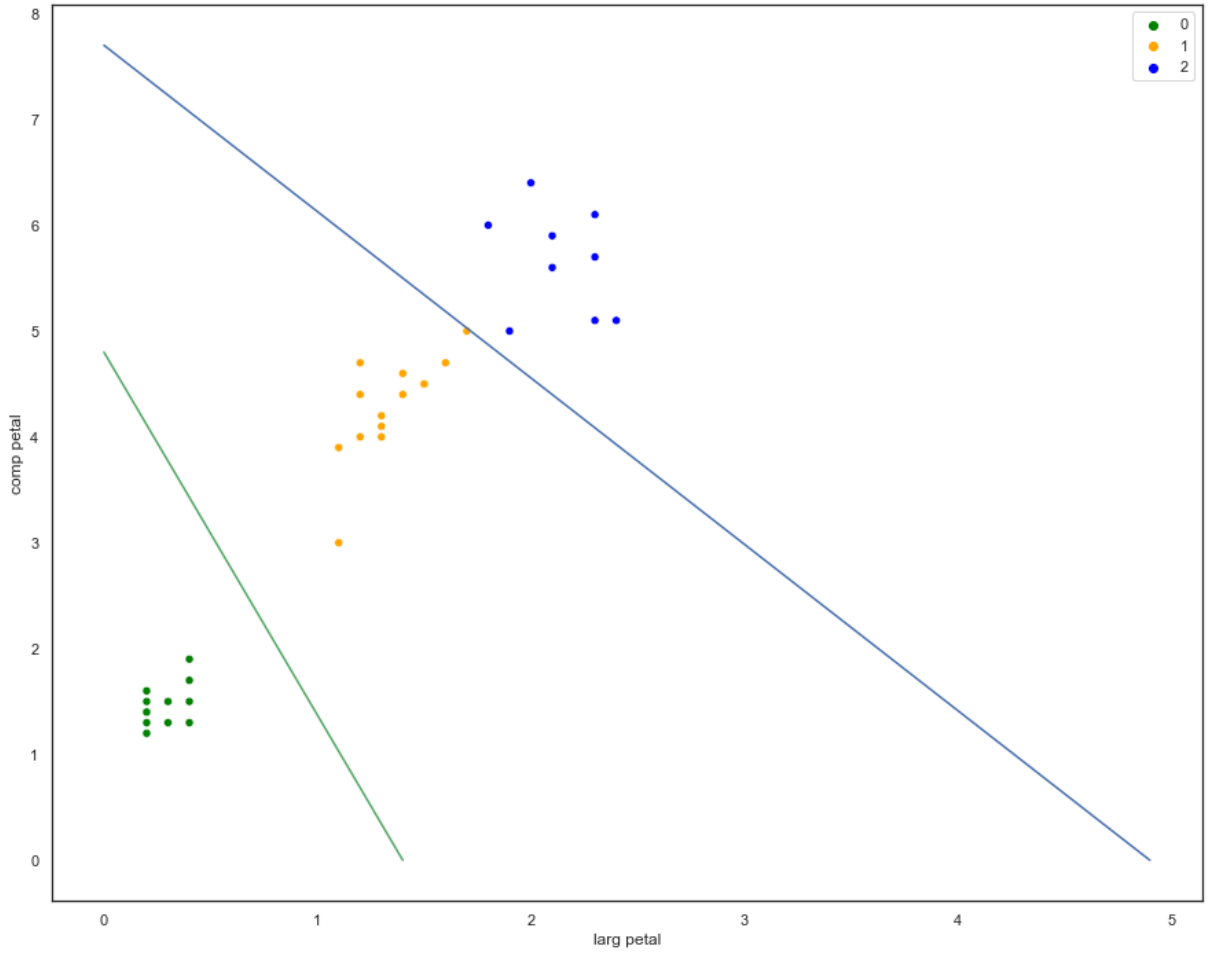
Realizado a criação do modelo de classificação para dados de treino, preparou-se os mesmos processos para os dados de teste, criando uma base de treino (Figura 17), com os dados tratados e utilizando-os na geração de um outro gráfico que trace as duas retas que delimitam as espécies (Figura 18 ), sendo observado que todos os dados de setosa e virginica foram

classificados corretamente, e possivelmente um dado da versicolor apresente algum erro pois se encontra numa posição muito próxima da reta traçada entre versicolor e virginica (azul).

*Figura 17: Criação de uma base de dados de teste.*

```
In [24]: #tratamento de dados de teste
         teste = pd.DataFrame(x_teste)
         teste.columns = ['comp sepal', 'larg sepal', 'comp petal', 'larg petal']
         teste['target'] = y_teste
```

*Figura 18: Delimitação de espécies: dados de teste.*




### 3.3. Execução das predições.

Ao utilizar o RandomForest, para determinar as predições para o modelo, onde se aplicou uma coluna para os dados com x_treino e y_treino, e ao lado solicitou-se uma coluna com a predição de classificação para os dados de teste (Figura 19), e partir disso foi gerado o resultado dessa comparação (Figura 20), no qual apresentou apenas um resultado conflitante entre o target e a predição para os dados de teste, sendo esta, a medida 22 (Figura 21).

*Figura 19: RandomForest para predição.*

```
In [27]: #Usando Random Forest
         from sklearn.ensemble import RandomForestClassifier

In [28]: modelo = RandomForestClassifier()

In [29]: modelo.fit(x_treino,y_treino)
Out[29]: RandomForestClassifier()

In [30]: predicao = modelo.predict(x_teste)

In [31]: predicao
Out[31]: array([0, 1, 1, 0, 2, 1, 2, 0, 0, 2, 1, 0, 2, 1, 1, 0, 1, 1, 0, 0, 1, 1,
                2, 0, 2, 1, 0, 0, 1, 2, 1, 2, 1, 2, 2, 0, 1, 0])

In [32]: teste['predicao'] = predicao
         teste['check'] = teste['target'] == teste['predicao']
         display(teste)
```

*Figura 20: Check para valores de predição e target.*

| | comp sepal | larg sepal | comp petal | larg petal | target | predicao | check |
|---|---|---|---|---|---|---|---|
| **0** | 5.8 | 4.0 | 1.2 | 0.2 | 0 | 0 | True |
| **1** | 5.1 | 2.5 | 3.0 | 1.1 | 1 | 1 | True |
| **2** | 6.6 | 3.0 | 4.4 | 1.4 | 1 | 1 | True |
| **3** | 5.4 | 3.9 | 1.3 | 0.4 | 0 | 0 | True |
| **4** | 7.9 | 3.8 | 6.4 | 2.0 | 2 | 2 | True |
| **5** | 6.3 | 3.3 | 4.7 | 1.6 | 1 | 1 | True |
| **6** | 6.9 | 3.1 | 5.1 | 2.3 | 2 | 2 | True |
| **7** | 5.1 | 3.8 | 1.9 | 0.4 | 0 | 0 | True |
| **8** | 4.7 | 3.2 | 1.6 | 0.2 | 0 | 0 | True |
| **9** | 6.9 | 3.2 | 5.7 | 2.3 | 2 | 2 | True |
| **10** | 5.6 | 2.7 | 4.2 | 1.3 | 1 | 1 | True |
| **11** | 5.4 | 3.9 | 1.7 | 0.4 | 0 | 0 | True |
| **12** | 7.1 | 3.0 | 5.9 | 2.1 | 2 | 2 | True |
| **13** | 6.4 | 3.2 | 4.5 | 1.5 | 1 | 1 | True |
| **14** | 6.0 | 2.9 | 4.5 | 1.5 | 1 | 1 | True |
| **15** | 4.4 | 3.2 | 1.3 | 0.2 | 0 | 0 | True |
| **16** | 5.8 | 2.6 | 4.0 | 1.2 | 1 | 1 | True |
| **17** | 5.6 | 3.0 | 4.5 | 1.5 | 1 | 1 | True |
| **18** | 5.4 | 3.4 | 1.5 | 0.4 | 0 | 0 | True |
| **19** | 5.0 | 3.2 | 1.2 | 0.2 | 0 | 0 | True |
| **20** | 5.5 | 2.6 | 4.4 | 1.2 | 1 | 1 | True |
| **21** | 5.4 | 3.0 | 4.5 | 1.5 | 1 | 1 | True |
| **22** | 6.7 | 3.0 | 5.0 | 1.7 | 1 | 2 | False |

*Figura 21: Check inválido para predição e target.*

```
In [33]: teste[teste.check == False]
```

Out[33]:

| | comp sepal | larg sepal | comp petal | larg petal | target | predicao | check |
|---|---|---|---|---|---|---|---|
| **22** | 6.7 | 3.0 | 5.0 | 1.7 | 1 | 2 | False |

### 3.4.Avaliação de precisão.

Por último, foi realizada a avaliação da acurácia do modelo para as predições de classificações de setosa, versicolor e virginica (Figura 22), em que se analisou uma precisão de 100% para a classificação de setosa e versicolor, e 90% para virginica, como também, uma acurácia de 97% quanto as predições realizadas.

*Figura 22: Acurácia do modelo.*

| | precision | recall | f1-score | support |
|---|---|---|---|---|
| setosa | 1.00 | 1.00 | 1.00 | 13 |
| versicolor | 1.00 | 0.94 | 0.97 | 16 |
| virginica | 0.90 | 1.00 | 0.95 | 9 |
| accuracy | | | 0.97 | 38 |
| macro avg | 0.97 | 0.98 | 0.97 | 38 |
| weighted avg | 0.98 | 0.97 | 0.97 | 38 |

## 4. Conclusões

O modelo realizado apresentou dados de correlações muito fortes entre medidas iguais, além destas, observou-se expressiva correlação >0,95 para os casos entre largura e comprimento de pétala, onde é visto uma facilidade de separação entre as espécies. Também foi analisado casos de correlações fortes, porém inversas, como as correlações entre comprimento de sépala e largura de sépala, que definem bem a inversão entre ambas, quando uma cresce e a outra decresce na mesma proporção, assim como a inexistência de correlações nulas para essa base de dados. As predições foram corretas a partir da validação de precisão, cujas possuem 100% de precisão para classificação de espécies do tipo setosa e versicolor, e de 90% para virginica, e uma acurácia de 97% em relação ao modelo, sendo valores válidos quanto às expectativas do experimento do modelo, levado em consideração a impossibilidade de um modelo de aprendizagem que possua uma porcentagem de erros inexistente.