Classificação das flores IRIS - Projeto final do curso python para data science do DATA ICMC

Matheus Henrique Batista dos Santos

[1]

1. Tratamento dos dados

Inicialmente obtive uma visão geral do dataframe para entender as colunas que foram carregadas. A tabela 1 mostra o head do dataset.

SepalLengthCm	SepalWidthCm	PetalLengthCm	PetalWidthCm	Species
5.1	3.5	1.4	0.2	Iris-setosa
4.9	3.0	1.4	0.2	Iris-setosa
4.7	3.2	1.3	0.2	Iris-setosa
4.6	3.1	1.5	0.2	Iris-setosa
5.0	3.6	1.4	0.2	Iris-setosa

Table 1. Tabela de dados das flores Iris

Verifiquei também valores ausentes demonstrados na tabela 2.

Atributo	Valores Ausentes
SepalLengthCm	0
SepalWidthCm	0
PetalLengthCm	0
PetalWidthCm	0
Species	0

Table 2. Valores ausentes nas colunas

Depois disso verifiquei as linhas duplicadas e encontrei 3 linhas, pensei em dropa-las mas resolvi testar os modelos com e sem as linhas para ver diferentes combinações. Nos resultados isso pode ser visto.

Para avaliar se era possível dropar alguma coluna do dataframe, realizei algumas análises dos dados. Inicialmente utilizei o pair plot do seaborn, na figura 1, para verificar como estão a distribuição dos valores em cada atributo e como está a dispersão, o objetivo foi verificar se eles estão bem definidos e agrupados para cada classe ou se há uma intersecção forte entre as classes.

É possível notar que a classe Iris-setosa está distribuída em intervalos de valor bem definidos para todos os parâmetros e os modelos não devem ter dificuldade de identifica-la.

Já para as classes Iris-versicolor e Iris-virginica é possível ver algumas intersecções principalmente nas distribuições SepalLengthCm e SepalWidthCm. Nessas classes o modelo pode ter uma dificuldade maior para classificar corretamente.

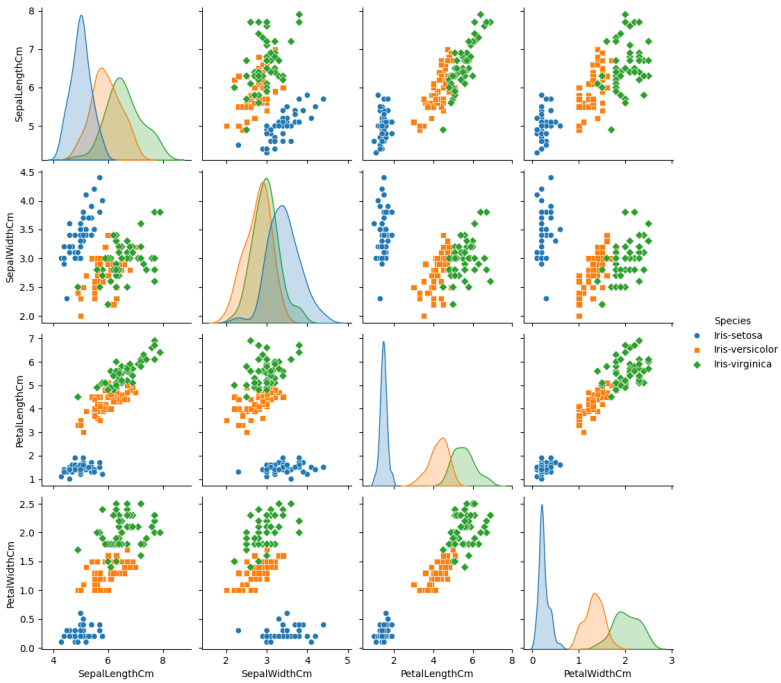

Figure 1. Visão geral dos dados

Nas curvas de distribuição é possível ver que a setosa fica bem definida para PetalWidthCm e PetalLengthCM, já para os outros atributos a distribuição se sobrepõe um pouco. Para SepalWidthCm vemos que versicolor e virginica se sobrepõe muito.

A correlação entre PetalLenthCm e PetalWidthCm, na figura 2 indica que essas duas categorias podem ensinar a mesma informação para o modelo, pois ambas estão muito correlacionadas, resolvi testar os modelos dropando e não dropando uma dessas colunas. A coluna SepalLengthCm também possui correlação alta e também entrou para os testes.

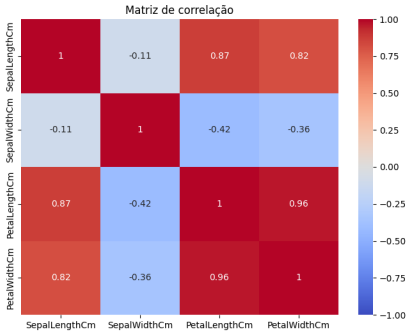

Figure 2. Matriz de correlação

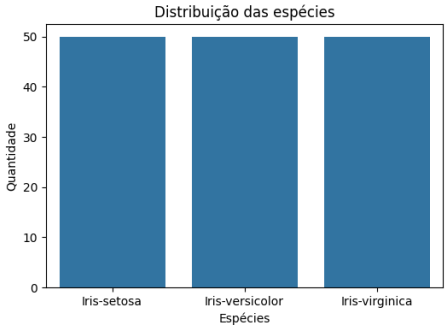

Figure 3. Distribuição das classes

Os dados estão bem distribuídos, como mostra figura 3, entre as classes então o modelo não irá aprender muito sobre uma determinada classe e pouco sobre outra, não é necessário under nem overfitting.

2. Treinamento

Realizado o tratamento, foi possível pensar no treinamento dos modelos.

Os dados foram separados com 70% para treino e 30% para teste. Os modelos treinados foram Random forest, SVM, Logistic regression e KNeighbors. Todos são da biblioteca sklearn.

Tentei variar bastante o pré-processamento dos dados até para entender como essa etapa impacta no treinamento dos modelos. Dropei as linhas duplicadas e as colunas que possuem uma grande correlação em diferentes cenários. Os cenários treinados foram:

- sem dropar nada.
- dropando somente as linhas duplicadas.
- dropando somente a coluna PetalWidthCm.
- dropando somente a coluna PetalLengthCm.
- dropando somente a coluna SepalLenghtCm.
- dropando linhas duplicadas com coluna PetalWidthCm.
- dropando linhas duplicadas com coluna PetalLengthCm.
- dropando linhas duplicadas com coluna SepalLenghtCm.

3. Teste

Realizados os treinamentos nos diversos cenários, os testes foram realizados avaliando quatro métricas: acurácia, precisão, recall e F1-score. Os resultados podem ser vistos nas tabelas 3, 4, 5, 6, 7, 7, 8, 9 e 10.

Modelo	Acurácia	Precisão	Recall	F1-score
Random Forest	0.9778	0.9792	0.9778	0.9777
SVC	0.9778	0.9792	0.9778	0.9777
Regressão Logística	0.9778	0.9792	0.9778	0.9777
Árvore de Decisão	0.9556	0.9608	0.9556	0.9551
KNeighbors	0.9778	0.9792	0.9778	0.9777

Table 3. Cenário 1 - sem dropar nada

Modelo	Acurácia	Precisão	Recall	F1-score
Random Forest	0.9556	0.9615	0.9556	0.9558
SVC	0.9556	0.9615	0.9556	0.9558
Regressão Logística	0.9556	0.9615	0.9556	0.9558
Árvore de Decisão	0.9556	0.9615	0.9556	0.9558
KNeighbors	0.9556	0.9615	0.9556	0.9558

Table 4. Cenário 2 - dropando somente as linhas duplicadas.

No cenário 1, sem remoção de dados, todo os modelos apresentaram um bom desempenho, acima de 0.97 em todas as métricas. No cenário 2, com remoção das linhas duplicadas, as métricas caem um pouco de forma uniforme e ficam próximas de 0.95. Então sobre a sensibilidade aos dados duplicados, a remoção de duplicatas reduz ligeiramente o desempenho, indicando que duplicatas podem trazer informações úteis para a classificação.

Modelo	Acurácia	Precisão	Recall	F1-score
Random Forest	0.9111	0.9111	0.9111	0.9111
SVC	0.9333	0.9343	0.9333	0.9331
Regressão Logística	0.9333	0.9343	0.9333	0.9331
Árvore de Decisão	0.9111	0.9149	0.9111	0.9102
KNeighbors	0.9333	0.9343	0.9333	0.9331

Table 5. Cenário 3 - dropando somente a coluna PetalWidthCm.

Modelo	Acurácia	Precisão	Recall	F1-score
Random Forest	0.9333	0.9343	0.9333	0.9331
SVC	0.9778	0.9794	0.9778	0.9778
Regressão Logística	0.9778	0.9794	0.9778	0.9778
Árvore de Decisão	0.8889	0.8970	0.8889	0.8867
KNeighbors	0.9556	0.9556	0.9556	0.9556

Table 6. Cenário 4 - dropando somente a coluna PetalLengthCm.

Sobre a importância das variáveis, PetalWidthCm e PetalLenghtCm impactam signifcamente os modelos baseados em árvore, ramdom foreste e árvore de decisão, os ourtos modelos são mais estáveis a ess aremoção. Já SepalLenghtCm impactou menos todos os modelos, ficaram próximos de 0.95 com destaque para a regressão logística em que as métricas ficaram próximas de 0.97, mesmo valor do cenário 1.

Nos três últimos cenários, a remoção de linhas duplicadas e de uma coluna específica reduziu o desempenho geral dos modelos em comparação com o cenário completo (Cenário 1). Contudo, o Cenário 8 mostrou o maior equilíbrio entre os modelos, com todos atingindo métricas semelhantes, indicando que SepalLengthCm tem um impacto menor em comparação com PetalWidthCm e PetalLengthCm.

O melhor cenário foi o cenário 1, sem remoção de colunas, onde a maioria dos modelos apresentou o desempenho mais alto e consistente. PetalWidthCm e PetalLengthCm são mais relevantes para os modelos, especialmente para os modelos de árvore e baseados em proximidade. A remoção de duplicatas reduz ligeiramente o desempenho. SVC e regressão logística mostram-se mais resilientes à perda de variáveis, enquanto random forest e árvore de decisão são mais sensíveis.

O pior cenário foi o cenário 3, dropando a coluna PetalWidthCm, em que houve uma redução significativa das métricas, indicando que indicando que PetalWidthCm é uma coluna essencial para a qualidade dos modelos.

Para entender como foram os acertos do pior e do melhor modelo, foi feita a matriz de confusão para cada modelo em ambos os cenários, apresentados nas figuras **??** e **??**. A árvore de decisão, na matriz de confusão, errou apenas 2 predições e todos os outros modelos erraram apenas 1 predição no melhor cenário. No pior cenário todos os modelos erraram 3 predições.

Modelo	Acurácia	Precisão	Recall	F1-score
Random Forest	0.9556	0.9608	0.9556	0.9551
SVC	0.9556	0.9556	0.9556	0.9556
Regressão Logística	0.9778	0.9792	0.9778	0.9777
Árvore de Decisão	0.9556	0.9608	0.9556	0.9551
KNeighbors	0.9556	0.9608	0.9556	0.9551

Table 7. Cenário 5 - dropando somente a coluna SepalLenghtCm

Modelo	Acurácia	Precisão	Recall	F1-score
Random Forest	0.8889	0.9011	0.8889	0.8896
SVC	0.9333	0.9458	0.9333	0.9338
Regressão Logística	0.9556	0.9615	0.9556	0.9558
Árvore de Decisão	0.9111	0.9320	0.9111	0.9115
KNeighbors	0.9111	0.9320	0.9111	0.9115

Table 8. Cenário 6 - dropando linhas duplicadas com coluna PetalWidthCm

Modelo	Acurácia	Precisão	Recall	F1-score
Random Forest	0.9556	0.9615	0.9556	0.9558
SVC	0.9556	0.9615	0.9556	0.9558
Regressão Logística	0.9333	0.9352	0.9333	0.9336
Árvore de Decisão	0.9111	0.9111	0.9111	0.9111
KNeighbors	0.9111	0.9172	0.9111	0.9117

Table 9. Cenário 7 - dropando linhas duplicadas com coluna PetalLengthCm

Modelo	Acurácia	Precisão	Recall	F1-score
Random Forest	0.9556	0.9615	0.9556	0.9558
SVC	0.9556	0.9615	0.9556	0.9558
Regressão Logística	0.9556	0.9615	0.9556	0.9558
Árvore de Decisão	0.9556	0.9615	0.9556	0.9558
KNeighbors	0.9556	0.9615	0.9556	0.9558

Table 10. Cenário 8 - dropando linhas duplicadas com coluna SepalLenghtCm

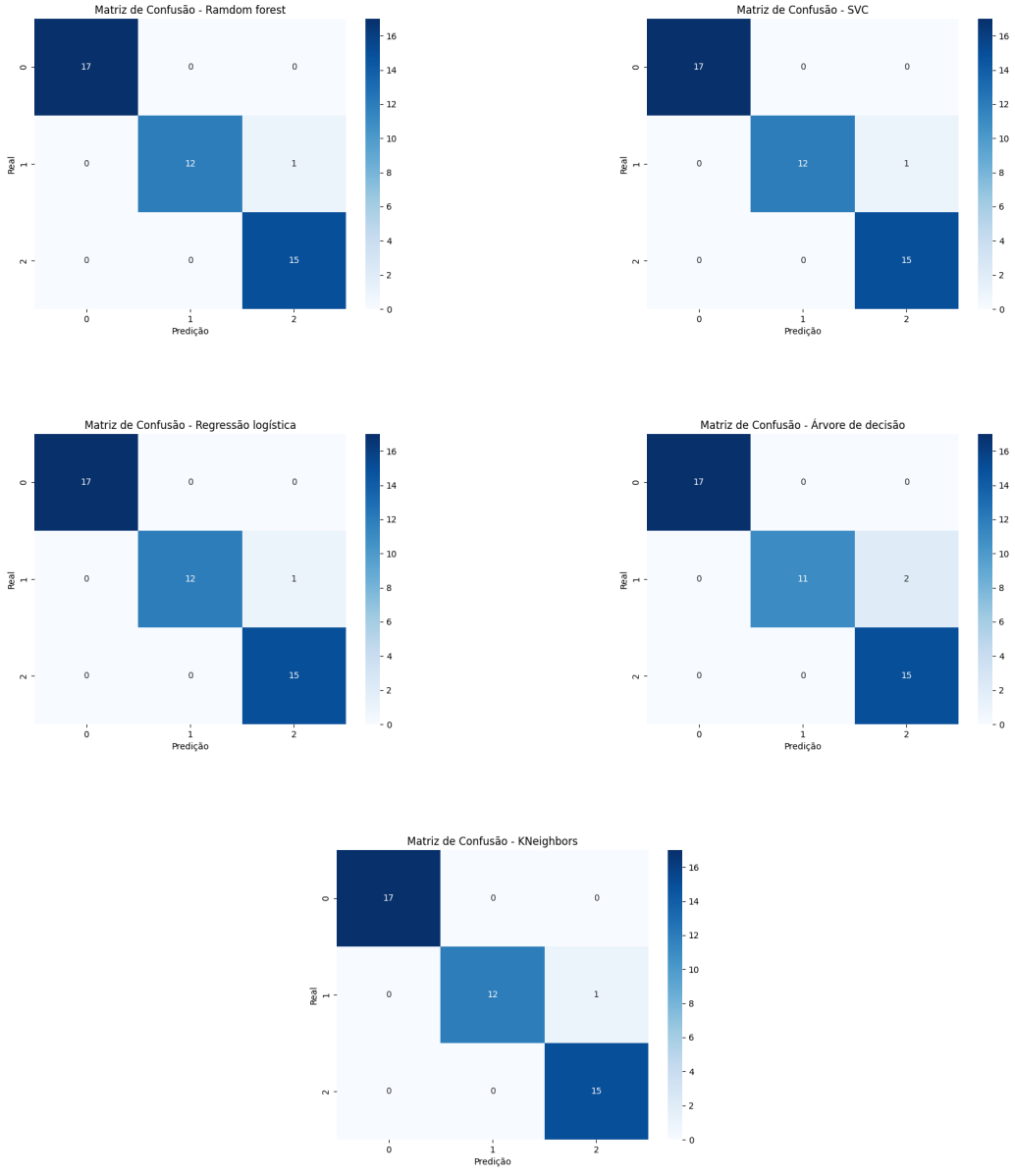

Figure 4. Melhor cenário - sem dropar nada

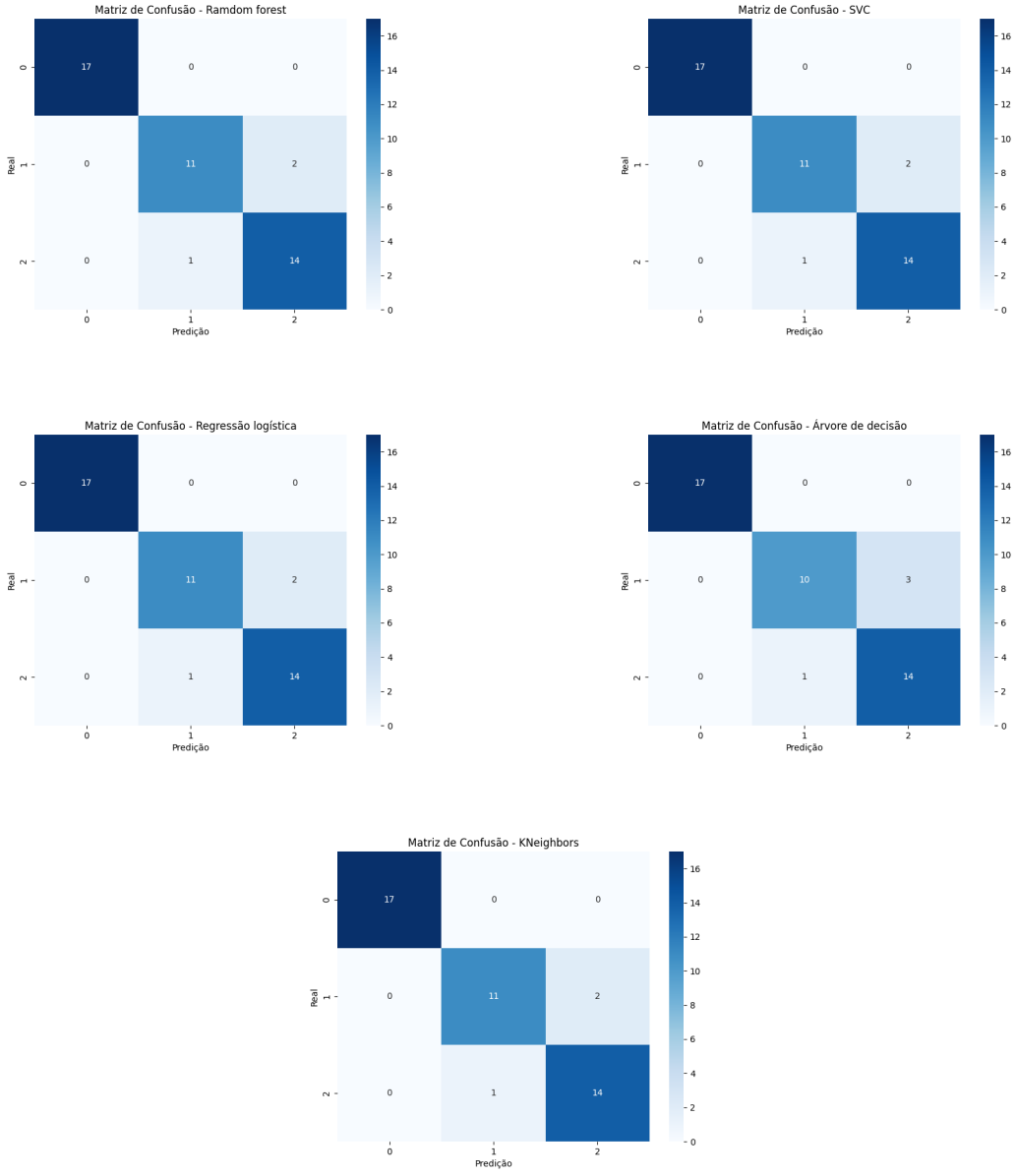

Figure 5. Pior cenário - dropando a coluna PetalWidhtCm