

Inteligência Artificial

Trabalho de Aprendizado de Máquina

Trabalho de Aprendizado de Máquina

Usando o exemplo que está em:

<https://paulovasconcellos.com.br/como-criar-seu-primeiro-aplicativo-de-machine-learning-7b6af291ba11>

Resolver um problema de classificação ou clusterização usando uma ferramenta de Aprendizado de Máquina (pode ser o WEKA, o Scikit-Learn, o Watson, ou outra).

Usar o modelo disponibilizado para descrever o problema, a ferramenta, a base de dados e os resultados obtidos.

Nomes dos componentes do Grupo:

Carolina Carvalho Simões

Rafael dos Santos de Oliveira Lima

Thalles Cotta Fontainha

Professora:

1. Descrição do problema.

Paulo Vasconcellos, autor do link base deste trabalho, em seu artigo utiliza o algoritmo *K-Nearest Neighbor*, obtendo uma acurácia de 97% entre dois conjuntos: treino e teste. Nosso grupo escolheu abordar o mesmo problema, porém utilizando o SVM e analisando mais 3 métricas (F1, recall e precision).

2. Justificativa e descrição da ferramenta escolhida.

A **scikit-learn** é uma biblioteca de aprendizado de máquina de código aberto para a linguagem de programação Python. Ela inclui vários algoritmos de classificação, regressão e agrupamento incluindo máquinas de vetores de suporte, florestas aleatórias, gradient boosting, k-means e DBSCAN, e é projetada para interagir com as bibliotecas Python numéricas e científicas NumPy e SciPy.

Usamos o **SVM** que é um modelo bastante popular e versátil capaz de realizar classificações, regressões e detecção de outliers. Esse modelo funciona muito bem para classes linearmente separáveis, que é o caso do dataset escolhido.

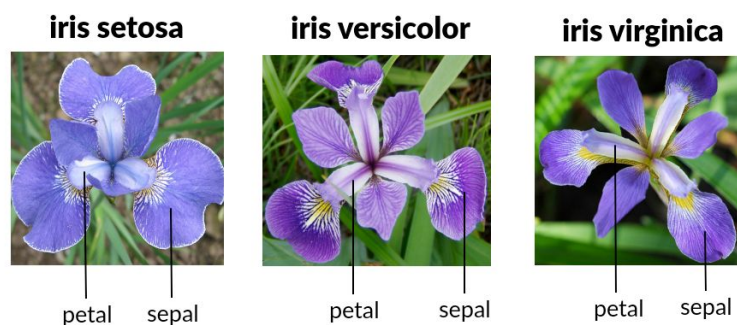
3. Descrição do dataset escolhido com o endereço de onde foi obtido.

O **conjunto de dados flor Iris** é um conjunto de dados multivariados introduzido pelo estatístico e biólogo britânico Ronald Fisher em seu artigo no ano de 1936.

O conjunto de dados consiste em 50 amostras de cada uma das três espécies de *Iris* (*Iris setosa*, *Iris virginica* e *Iris versicolor*).

Quatro variáveis foram medidas em cada amostra: o comprimento e a largura das sépalas e pétalas, em centímetros.

Com base na combinação dessas quatro características, Fisher desenvolveu um modelo discriminante linear para distinguir as espécies umas das outras.



Endereço: <https://archive.ics.uci.edu/ml/datasets/iris>

4. Apresentação dos resultados.

```
# Carregando as Libs

from sklearn import datasets

import pandas as pd

import seaborn as sns

# Carregando dataset e transformando ele em DataFrame

iris = datasets.load_iris()

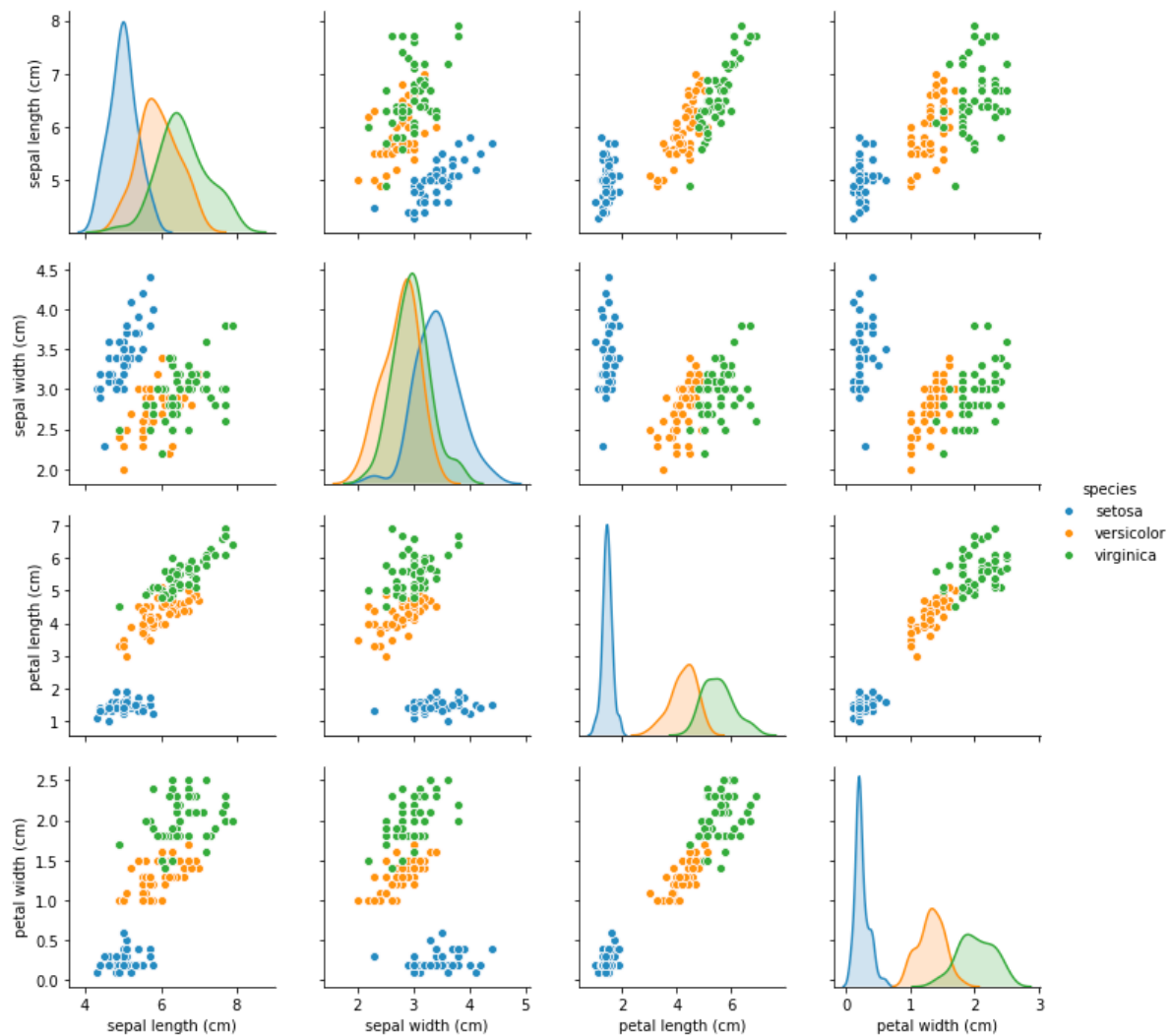
iris_df = pd.DataFrame(data=iris.data, columns=iris.feature_names)

iris_df['label'] = iris.target

iris_df['species'] = pd.Categorical.from_codes(iris.target,
iris.target_names)

# Plotando o gráfico em pares

sns.pairplot(iris_df[['sepal length (cm)', 'sepal width (cm)', 'petal
length (cm)', 'petal width (cm)', 'species']], hue='species')
```



```
# Usando o SVM para separar um conjunto de dados linearmente separáveis
(sem fç kernel)

from sklearn import svm

from sklearn.model_selection import train_test_split

X,y = load_iris(return_X_y=True)

X_train, X_test, y_train, y_test = train_test_split(X, y,
test_size=0.30, random_state=13)

# Parametro de regularizacao SVM
```

```

C = 1.0

clf = svm.SVC(kernel='linear', C=C)

# Treinando modelo em cima dos dados de treinamento
clf.fit(X_train, y_train)

# Previsao (valores de teste)
clf.predict(X_test)

```

```

array([1, 1, 0, 2, 2, 0, 2, 2, 0, 1, 2, 2, 1, 0, 2, 0, 2, 2, 1, 0, 1, 0,
       0, 2, 1, 2, 0, 2, 1, 2, 2, 0, 0, 2, 2, 1, 0, 1, 1, 0, 0, 2, 1, 2,
       2])

```

```

# Acuracia

clf.score(X_test, y_test)

```

```

0.9777777777777777

```

```

from sklearn.metrics import classification_report

# Mostrando as metricas de erro

print(classification_report(y_test, y_pred,
target_names=iris.target_names))

```

5. Discussão dos resultados (Diferenças dos modelos):

```

KNeighborsClassifier(algorithm='auto', leaf_size=30,
metric='minkowski',

metric_params=None, n_jobs=1, n_neighbors=1, p=2,

```

```

weights='uniform')

# Algoritmo do SVM
SVC(C=1.0, cache_size=200, class_weight=None, coef0=0.0,
    decision_function_shape='ovr', degree=3, gamma='auto_deprecated',
    kernel='linear', max_iter=-1, probability=False, random_state=None,
    shrinking=True, tol=0.001, verbose=False)

```

	precision	recall	f1-score	support
setosa	1.00	1.00	1.00	17
versicolor	1.00	0.93	0.96	14
virginica	0.93	1.00	0.97	14
accuracy			0.98	45
macro avg	0.98	0.98	0.98	45
weighted avg	0.98	0.98	0.98	45

6. Easter egg:

O que a assistente virtual disse para o desenvolvedor em um momento de intimidade?

Você me deixa nas nuvens.

