```
Caique de Souza Vilhegas
RA: 823130458
E-mail: caiquevilhegas.0458@aluno.saojudas.br
Turma: SIN1AN-MCB3
Curso: Ciência da computação
```

Caique de Souza Vilhegas RA: 823130458 E-mail: caiquevilhegas.0458@aluno.saojudas.br Turma: SIN1AN-MCB3 Curso: Ciência da computação

```
clima=['Ensolarado', 'Ensolarado', 'Nublado', 'Chuvoso', 'Chuvoso', 'Chuvoso', 'Nublado','Ensolarado', 'Ensolarado', 'Chuvoso', 'Ensolara
temp=['Quente', 'Quente', 'Quente', 'Suave', 'Legal', 'Legal', 'Legal', 'Suave','Legal', 'Suave', 'Suave', 'Suave', 'Quente', 'Suave']
brincar=['Não','Não','Sim','Sim','Sim','Não','Sim','Não','Sim','Sim','Sim','Sim','Sim','Não']
from sklearn import preprocessing
le = preprocessing.LabelEncoder()
clima_encoded = le.fit_transform(clima)
temp_encoded = le.fit_transform(temp)
print(clima_encoded)
print(temp_encoded)
alvo = le.fit_transform(brincar)
print(alvo)
carac=list(zip(clima_encoded,temp_encoded))
carac
```

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

```
from sklearn.neighbors import KNeighborsClassifier
modelo = KNeighborsClassifier(n_neighbors=3)
modelo.fit(carac,alvo)
predito = modelo.predict([[0,2]]) # 0:Nublado, 2:Suave
print(predito)
```

```
[1]
```

```
import numpy as np
import pandas as pd
from matplotlib import pyplot as plt
import seaborn as sns
sns.set()
from sklearn.datasets import load_breast_cancer
from sklearn.metrics import confusion_matrix
from sklearn.neighbors import KNeighborsClassifier
from sklearn.model_selection import train_test_split


cancer_mama = load_breast_cancer()
X = pd.DataFrame(cancer_mama.data, columns=cancer_mama.feature_names)
X = X[['mean area', 'mean compactness']]
X
y = pd.Categorical.from_codes(cancer_mama.target, cancer_mama.target_names)
y = pd.get_dummies(y, drop_first=True)
y
```

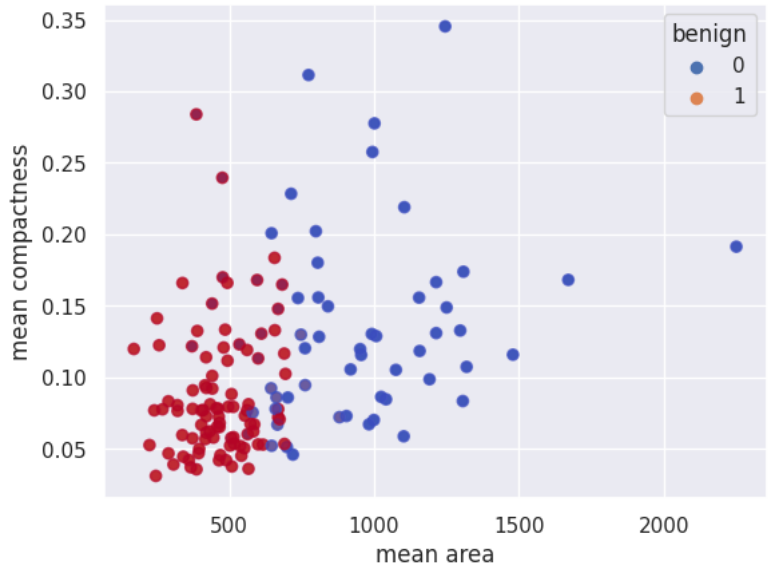
	benign
0	0
1	0
2	0
3	0

```
X_train, X_test, y_train, y_test = train_test_split(X, y, random_state=1)
knn = KNeighborsClassifier(n_neighbors=5, metric='euclidean')
knn.fit(X_train, y_train)
y_pred = knn.predict(X_test)
y_pred
```

```
/usr/local/lib/python3.10/dist-packages/sklearn/neighbors/_classification.py:215: DataConversionWarning: A column-vector y was pass
  return self._fit(X, y)
array([1, 1, 1, 0, 0, 0, 0, 0, 1, 0, 1, 1, 0, 1, 1, 1, 1, 1, 0, 1, 1,
       0, 1, 1, 1, 1, 0, 0, 0, 0, 1, 0, 1, 1, 1, 0, 0, 1, 1, 1, 1, 1, 1,
       1, 0, 1, 1, 0, 0, 0, 0, 1, 1, 1, 1, 1, 0, 0, 1, 1, 1, 1, 1, 1, 1,
       1, 0, 1, 1, 1, 1, 1, 0, 1, 0, 0, 1, 1, 0, 1, 0, 1, 0, 1, 1, 1, 1,
       0, 1, 1, 1, 1, 1, 0, 1, 1, 1, 1, 1, 1, 1, 1, 1, 0, 1, 1, 1, 0, 0,
       1, 1, 0, 1, 0, 0, 1, 1, 1, 1, 1, 0, 0, 1, 1, 0, 1, 0, 0, 0, 1, 0,
       1, 0, 1, 0, 0, 0, 1, 1, 0, 0, 1], dtype=uint8)
```

```
sns.scatterplot(
x='mean area',
y='mean compactness',
hue='benign',
data=X_test.join(y_test, how='outer')
)
plt.scatter(
X_test['mean area'],
X_test['mean compactness'],
c=y_pred,
cmap='coolwarm',
alpha=0.7
)
confusion_matrix(y_test, y_pred)
```

```
array([[42, 13],
       [ 9, 79]])
```



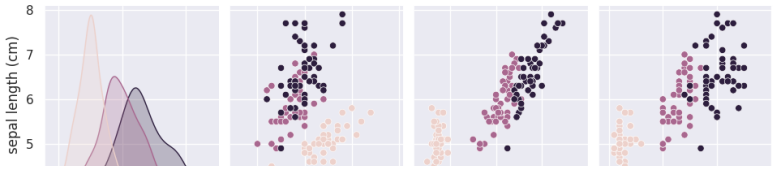
```
from sklearn.datasets import load_iris
from sklearn.model_selection import train_test_split
from sklearn.neighbors import KNeighborsClassifier
iris = load_iris()
X = iris.data
y = iris.target
(X_train, X_test, y_train, y_test) = train_test_split(X,y)
modelo = KNeighborsClassifier()
modelo.fit(X_train,y_train)
precisao = str(round(modelo.score(X_test,y_test) * 100, 2))+"%"
print("A acurácia do modelo k-NN foi",precisao)
y_pred = modelo.predict(X_test)
from sklearn.metrics import classification_report
print(classification_report(y_test, y_pred))
```

```
A acurácia do modelo k-NN foi 89.47%
              precision    recall  f1-score   support

           0       1.00      1.00      1.00        13
           1       0.92      0.80      0.86        15
           2       0.75      0.90      0.82        10

    accuracy                           0.89        38
   macro avg       0.89      0.90      0.89        38
weighted avg       0.90      0.89      0.90        38
```

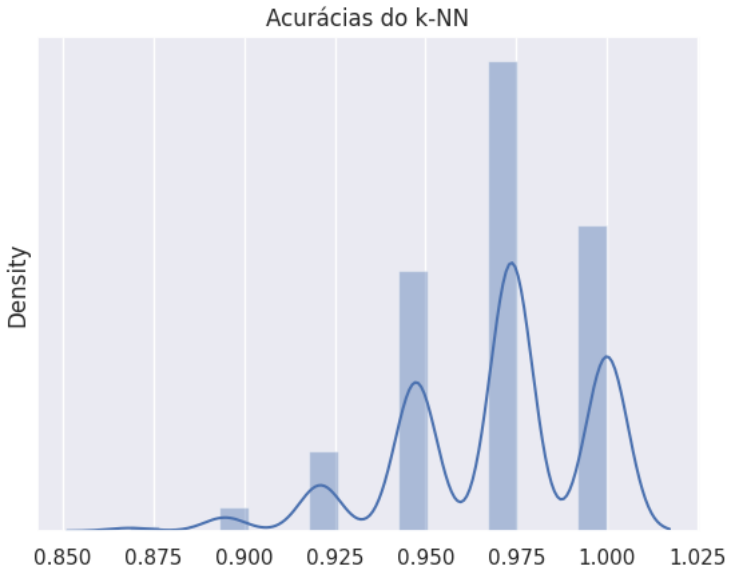
```
from sklearn.datasets import load_iris
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
iris = load_iris()
df = pd.DataFrame(iris.data,columns=iris.feature_names)
df['Species'] = iris.target
sns.pairplot(df, hue='Species', vars=iris.feature_names)
plt.show()
```




```
from sklearn.datasets import load_iris
from sklearn.model_selection import train_test_split
from sklearn.neighbors import KNeighborsClassifier
from sklearn.preprocessing import Normalizer
import numpy as np
import warnings
warnings.filterwarnings("ignore")
```

```
iris = load_iris()
X = iris.data
y = iris.target
scaler = Normalizer()
scaler.fit(X)
X = scaler.transform(X)
scores = []
for i in range(2000):
  X_train, X_test, y_train, y_test = train_test_split(X,y)
  model = KNeighborsClassifier()
  model.fit(X_train,y_train)
  precisao = model.score(X_test,y_test)
  scores.append(precisao)
print("Média: {:.2f}%".format(np.mean(scores)*100))
print("Desvio padrão: {:.2f}%".format(np.std(scores)*100))
import matplotlib.pyplot as plt
import seaborn as sns
sns.distplot(scores)
plt.yticks([])
plt.title("Acurácias do k-NN")
plt.show()
```

```
Média: 96.89%
Desvio padrão: 2.63%
```



Produtos pagos do Colab - Cancelar contratos

✓ 6s conclusão: 13:00