Iniciado em	sexta, 27 mar 2020, 01:37
Estado	Finalizada
Concluída em	sexta, 27 mar 2020, 01:41
Tempo empregado	3 minutos 50 segundos

Questão **1**

Completo

Vale 2,00 ponto(s).

Considere a base de dados 'vertebralcolumn-3C'. Calcule a precisão na classificação usando SVM com C=10. Use o código abaixo. Arrendonde o valor para uma casa decimal.

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
from sklearn.preprocessing import StandardScaler
from sklearn.model_selection import train_test_split

np.random.seed(42) # define the seed (important to reproduce the results)

data = pd.read_csv('data/vertebralcolumn-3C.csv', header=(0))
data = data.dropna(axis='rows') #remove NaN
data = data.to_numpy()
nrow,ncol = data.shape
y = data[:,-1]
X = data[:,0:ncol-1]

scaler = StandardScaler().fit(X)
X = scaler.transform(X)

p = 0.2 # fraction of elements in the test set
x_train, x_test, y_train, y_test = train_test_split(X, y, test_size = p, random_state = 42)
```

Escolha uma:

- a. 0.5
- b. 1.0
- c. 0.8
- d. 0.1
- e. 0.2

Questão **2**

Completo

Vale 2,00 ponto(s).

Para os dados gerados com o código abaixo, qual classificar oferece a melhor precisão (precision_score)? Arredonde para uma casa decimal.

```
from sklearn import datasets
import matplotlib.pyplot as plt
from sklearn.preprocessing import StandardScaler
from sklearn.model_selection import train_test_split
import numpy as np

np.random.seed(42) # define the seed (important to reproduce the results)

plt.figure(figsize=(6,4))

n_samples = 1000
data  = noisy_circles = datasets.make_circles(n_samples=n_samples, factor=.5, noise=.2, random_state = 42)
X = data[0]
y = data[1]
plt.scatter(X[:,0], X[:,1], c=y, cmap='viridis', s=50, alpha=0.7)
plt.show(True)

scaler = StandardScaler().fit(X)
X = scaler.transform(X)

p = 0.2 # fraction of elements in the test set
x_train, x_test, y_train, y_test = train_test_split(X, y, test_size = p, random_state = 42)
```

Escolha uma:

- a. Todos oferecem a mesma acurácia (considere apenas uma casa decimal)
- b. Gaussian Naive Bayes
- c. SVM com C = 10
- d. Floresta aleatória com n=100 árvores
- e. Árvore de decisão usando o critério entropia

Questão **3**

Completo

Vale 2,00 ponto(s).

Considere a base de dados da iris. Usando o algoritmo random forest, qual é o atributo mais importante para a classificação? Considere o código abaixo para ler e preparar os dados.

```
import random
random.seed(42) # define the seed (important to reproduce the results)
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt

data = pd.read_csv('data/iris.csv', header=(0))

# remove NaN
data = data.dropna(axis='rows') #
# armazena o nome das classes
classes = np.array(pd.unique(data[data.columns[-1]]), dtype=str)  #name of the clases
features_names = data.columns

data = data.to_numpy()
nrow,ncol = data.shape
y = data[:,-1]
X = data[:,0:ncol-1]

from sklearn.preprocessing import StandardScaler
scaler = StandardScaler().fit(X)
X = scaler.transform(X)

from sklearn.model_selection import train_test_split
p = 0.2 # fracao de elementos no conjunto de teste
x_train, x_test, y_train, y_test = train_test_split(X, y, test_size = p, random_state = 42)
```

Escolha uma:

- a. petal_length
- b. sepal_width
- c. petal_width
- d. todas tem a mesma importância
- e. sepal_length

Questão **4**

Completo

Vale 2,00 ponto(s).

Considere a base de dados da Vehicle. Usando o algoritmo random forest, qual é o atributo mais importante para a classificação? Considere o código abaixo para ler e preparar os dados.

```
import random
random.seed(42) # define the seed (important to reproduce the results)
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt

data = pd.read_csv('data/Vehicle.csv', header=(0))

# remove NaN
data = data.dropna(axis='rows') #
# armazena o nome das classes
classes = np.array(pd.unique(data[data.columns[-1]]), dtype=str)  #name of the clases
features_names = data.columns

data = data.to_numpy()
nrow,ncol = data.shape
y = data[:,-1]
X = data[:,0:ncol-1]

from sklearn.preprocessing import StandardScaler
scaler = StandardScaler().fit(X)
X = scaler.transform(X)

from sklearn.model_selection import train_test_split
p = 0.2 # fracao de elementos no conjunto de teste
x_train, x_test, y_train, y_test = train_test_split(X, y, test_size = p, random_state = 42)
```

Escolha uma:

- a. D.Circ
- b. Comp
- c. Scat.Ra
- d. Max.L.Ra
- e. Max.L.Rect

Questão **5**

Completo

Vale 2,00 ponto(s).

Considere a base BreastCancer. Qual o valor da área sob a curva Roc para o método SVM com C = 10? Considere o código abaixo para ler e preparar os dados. Arredonde para uma casa decimal.

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
from sklearn.preprocessing import StandardScaler
from sklearn.model_selection import train_test_split

np.random.seed(42) # define the seed (important to reproduce the results)

data = pd.read_csv('data/BreastCancer.csv', header=(0))
data = data.dropna(axis='rows') #remove NaN
data = data.to_numpy()
nrow,ncol = data.shape
y = data[:,-1]
X = data[:,0:ncol-1]

scaler = StandardScaler().fit(X)
X = scaler.transform(X)

p = 0.2 # fraction of elements in the test set
x_train, x_test, y_train, y_test = train_test_split(X, y, test_size = p, random_state = 42)
```

Escolha uma:

- a. 0.5
- b. 0.2
- c. 1.0
- d. 0.7
- e. 0.1

◄ Material de Base - zip

Seguir para...

Avaliação Final da Disciplina - pdf ►