Lista de exercícios - PCS3438

Gustavo Freitas de Sá Oliveira

05/12/2022

1 Introdução

Os programas desenvolvidos para a resolução dos problemas descritos foram desenvolvidos em Python. Foram utilizadas as bibliotecas: pandas (leitura e tratamento de dados), numpy (operações matemáticas) e sklearn (modelos de inteligência artificial e métodos de validação).

2 Questão 1

Considerando os dados presentes no arquivo class01.csv, treine o algoritmo Naive Bayes Gaussiano utilizando a metodologia de validação cruzada hold-out (utilize para treino as 350 primeiras linhas e para validação as demais). Qual o valor da acurácia a base de treino? Qual o valor da acurácia na base de validação?

```
1  import pandas as pd
2  import numpy as np
3
4  # Obtendo dados
5  df = pd.read_csv("data/class01.csv")
6  x = df.drop('target', axis = 1).to_numpy()
7  y = df['target'].to_numpy()
8
9  # Validação hold-out
10 from sklearn.model_selection import train_test_split
11 x_train, x_test, y_train, y_test = train_test_split(x, y,
       train_size = 350, shuffle = False)
12
13 # Modelo Naive Bayes Gaussiano
14 from sklearn.naive_bayes import GaussianNB
15 nb = GaussianNB()
16
17 nb.fit(x_train, y_train)
```

```
18
19 acuracia_treino = nb.score(x_train, y_train)
20 acuracia_validacao = nb.score(x_test, y_test)
21
22 print('Acurácia na base de treino:', acuracia_treino)
23 print('Acurácia na base de validação:', acuracia_validacao)
```

Saída observada:

```
1     Acurácia na base de treino: 0.76
2     Acurácia na base de validação: 0.6276923076923077
```

Nesse exemplo, percebe-se que a acurácia na base se treino foi maior do na base de validação. Esse fenômeno pode estar relacionado com o *overfitting*, refletindo uma "memorização" dos dados pelo modelo.

3 Questão 2

Considerando os dados presentes no arquivo class02.csv, treine o algoritmo 10-Nearest Neighbors (KNN com $k = 10$ e distância Euclidiana), utilizando a metodologia de validação cruzada k-fold com 5 folds. Considere que a primeira pasta de validação seja formada pelas primeiras 20% linhas do arquivo, que a segunda pasta de validação seja formada pelas 20% linhas seguintes, e assim por diante, até atingir a última pasta, formada pelas 20% linhas finais da base. Qual a acurácia média para a base de validação?

```
1  import pandas as pd
2  import numpy as np
3
4  # Obtendo dados
5  df = pd.read_csv("data/class02.csv")
6  x = df.drop('target', axis = 1).to_numpy()
7  y = df['target'].to_numpy()
8
9  # Validação K-fold
10 from sklearn.model_selection import KFold
11 kf = KFold(5)
12
13 # Modelo KNN
14 from sklearn.neighbors import KNeighborsClassifier
15 knn = KNeighborsClassifier(10)
16
17 acuracia_treino, acuracia_validacao = 0, 0
18 for train, test in kf.split(x):
19     x_train, x_test = x[train], x[test]
20     y_train, y_test = y[train], y[test]
```

```
21
22      knn.fit(x_train, y_train)
23
24      acuracia_treino += knn.score(x_train, y_train)
25      acuracia_validacao += knn.score(x_test, y_test)
26
27  acuracia_treino /= kf.get_n_splits(x)
28  acuracia_validacao /= kf.get_n_splits(x)
29
30  print('Acurácia na base de treino:', acuracia_treino)
31  print('Acurácia na base de validação:', acuracia_validacao)
```

Saída observada:

```
1       Acurácia na base de treino: 0.8661666666666668
2       Acurácia na base de validação: 0.8386666666666667
```

Assim como no exemplo anterior, percebe-se uma maior acurácia na base de treino. Isso pode estar também relacionado com o fenômeno anteriormente descrito.

4 Questão 3

Considerando os dados presentes no arquivo reg01.csv, obtenha um modelo de regressão linear com regularização L1 (LASSO com $\alpha = 1$) utilizando a metodologia Leave-One-out. Qual o valor médio do Root Mean Squared Error (RMSE) para a base de treino e para a base de validação?

```
1   import pandas as pd
2   import numpy as np
3
4   # Obtendo dados
5   df = pd.read_csv("data/reg01.csv")
6   x = df.drop('target', axis = 1).to_numpy()
7   y = df['target'].to_numpy()
8
9   # Validação Leave-One-Out
10  from sklearn.model_selection import LeaveOneOut
11  loo = LeaveOneOut()
12
13  # Modelo LASSO
14  from sklearn.linear_model import Lasso
15  l = Lasso(1)
16
17  rmse_treino, rmse_validacao = 0, 0
18  from sklearn.metrics import mean_squared_error
19  for train, test in loo.split(x):
```

```
    x_train, x_test = x[train], x[test]
    y_train, y_test = y[train], y[test]

    l.fit(x_train, y_train)

    rmse_treino += np.sqrt(mean_squared_error(y_train, l.predict
        (x_train)))
    rmse_validacao += np.sqrt(mean_squared_error(y_test, l.
        predict(x_test)))

rmse_treino /= loo.get_n_splits(x)
rmse_validacao /= loo.get_n_splits(x)

print('RMSE na base de treino:', rmse_treino)
print('RMSE na base de validação:', rmse_validacao)
```

Saída observada:

```
RMSE na base de treino: 19.220259837710355
RMSE na base de validação: 15.465218791702428
```

O *RMSE* foi maior na base de validação que na base de treino que na base de validação. A escolha do método de validação (Leave-One-Out) pode ter influenciado nesse fenômeno, uma vez que, nesse método, as validações acontecem para cada linha da base de dados.

5 Questão 4

Considerando os dados do arquivo reg02.csv, treine árvores de regressão, sem realizar podas, utilizando a metodologia de validação cruzada k-fold com $k = 5$. Qual o valor do Mean Absolute Error (MAE) para a base de treino? Qual o valor médio do MAE para a base de validação?

```
import pandas as pd
import numpy as np

# Obtendo dados
df = pd.read_csv("data/reg02.csv")
x = df.drop('target', axis = 1).to_numpy()
y = df['target'].to_numpy()

# Validação K-fold
from sklearn.model_selection import KFold
kf = KFold(5)

# Modelo árvore de regressão
from sklearn.tree import DecisionTreeRegressor
```

```
15 tree = DecisionTreeRegressor(criterion = 'absolute_error')
16
17 mae_treino, mae_validacao = 0, 0
18 from sklearn.metrics import mean_absolute_error
19 for train, test in kf.split(x):
20     x_train, x_test = x[train], x[test]
21     y_train, y_test = y[train], y[test]
22
23     tree.fit(x_train, y_train)
24
25     mae_treino += mean_absolute_error(y_train, tree.predict(
           x_train))
26     mae_validacao += mean_absolute_error(y_test, tree.predict(
           x_test))
27
28 mae_treino /= kf.get_n_splits(x)
29 mae_validacao /= kf.get_n_splits(x)
30
31 print('MAE na base de treino:', mae_treino)
32 print('MAE na base de validação:', mae_validacao)
```

Saída observada:

```
1     MAE na base de treino: 0.0
2     MAE na base de validação: 43.22051929803169
```

Nesse exemplo, o *MAE* foi nulo na base de treino. Esse resultado é esperado em virtude do tipo de modelo escolhido (árvore de regressão). O *MAE* na base de validação, entretanto, foi diferente de zero.