Aprendizagem de Máquina: Atividade 04 − Avaliação de Classificadores

Carlos Emmanuel Pereira Alves
Curso de Bacharelado em Ciência da Computação
Universidade Federal do Agreste de Pernambuco (UFAPE)
Garanhuns, Brasil
carlos.emmanuel.236@gmail.com

1) Qual é o melhor classificador?
   Para saber qual vai ser o melhor classificador, primeiro precisamos olhar para o problema que queremos resolver. Quais os tipos de dados que ele utiliza, se queremos um treinamento mais rápido, um tempo de classificação menor, temos que olhar para todos os aspectos e analisá-los corretamente. O melhor classificador para um problema pode não ser o melhor para outro.

2) Qual a diferença entre avaliação qualitativa e quantitativa de classificadores? Cite exemplos de critérios avaliação qualitativa de classificadores. Cite exemplos de critérios avaliação quantitativa de classificadores.
   Avaliação Qualitativa: nela o classificador deve ter características como interpretabilidade, uma pessoa deve ser capaz de entendê-la. Alguns critérios são o tipo de dado trabalhado, se ele consegue aprender novos dados sem ter que ser treinado novamente.
   Avaliação Quantitativa: são métricas feitas a partir dos dados. Como exemplos temos a taxa de erro, o tempo de treinamento, o tempo de classificação, são dados mais numéricos.

3) Realize um Holdout com 1-NN (distância Euclidiana), utilizando 70% dos dados para treinamento e o restante (30%) para teste na base de dados Wine archive.ics.uci.edu/ml/datasets/Wine. Você precisa mostrar como calculou cada métrica, não pode utilizar biblioteca que já calcula a métrica diretamente mas pode utilizar biblioteca para o 1-NN e para dividir os dados entre treino e teste.
    a) Calcule a matriz de confusão.
    b) Calcule o Recall por classe.
    c) Calcule a Taxa de acerto do classificador.
    d) Calcule a Precisão por classe.
    e) Calcule a Medida-F por classe.
    f) Calcule a Taxa de FP por classe.

```python
import pandas as pd
import numpy as np
from sklearn.model_selection import train_test_split
from sklearn.neighbors import KNeighborsClassifier

def confusionMatrix(y_test, predict):
  classes = np.unique(y_test).astype(str)
  data = {outer_key: {inner_key: 0 for inner_key in classes} for outer_key in classes}

  cf = pd.DataFrame(data, index=classes)

  for i in range(len(y_test)):
    cf.loc[str(y_test.iloc[i]), str(predict[i])] += 1

  return cf

def accuracy(y_test, predict):
  cf = confusionMatrix(y_test, predict)
  total = 0
  correct = 0

  for row, index in cf.iterrows():
    for column, value in index.items():
        total += value
        if row == column:
          correct += value

  return correct / total

def recall_fscore_precision(y_test, predict):
  cf = confusionMatrix(y_test, predict)
  recall = {}
  precision = {}
  fscore = {}

  for row, index in cf.iterrows():
    recall_total = 0
    recall_correct = 0

    for column, value in index.items():
      recall_total += value
      if row == column:
          recall_correct = value
    recall[row] = recall_correct / recall_total

  for row, index in cf.iterrows():
    precision_correct = cf[row][row]
    precision_total = cf[row].sum()
    precision[row] = precision_correct / precision_total

  for row, index in cf.iterrows():
    fscore[row] = (2 * precision[row] * recall[row]) / (precision[row] + recall[row])

  return recall, fscore, precision
```

```python
cols = ['class', 'alcohol', 'malic_acid', 'ash', 'alcalinity_of_ash', 'magnesium', 'total_phenols', 'flavanoids',
        'nonflavanoid_phenols', 'proanthocyanins', 'color_intensity', 'hue', 'OD280/OD315_of_diluted_wines', 'proline']

data = pd.read_csv('wine.data', header=None, names=cols)

X_train, X_test, y_train, y_test = train_test_split(data[cols[1:]], data['class'], test_size=0.3, random_state=42)

knn = KNeighborsClassifier(n_neighbors=1)
knn.fit(X_train, y_train)
predict = knn.predict(X_test)

cf = confusionMatrix(y_test, predict)

acc = accuracy(y_test, predict)

recall, fscore, precision = recall_fscore_precision(y_test, predict)

print('Matriz de Confusão:\n', cf)
print('\nPrecisão:', precision)
print('\nRecall:', recall)
print('\nMedida-F:', fscore)
print('\nTaxa de acerto:', acc)
print('\nTaxa de Falsos Positivos (FP):')
for i in range(3):
  total = cf.iloc[i].sum()
  fp = total - cf.iloc[i, i]
  fp_rate = fp / total
  print(f"Classe {i+1}: {fp_rate:.4f}")
```

```
Matriz de Confusão:
     1   2   3
1  17   0   2
2   3  16   2
3   1   3  10

Precisão: {'1': 0.8095238095238095, '2': 0.8421052631578947, '3': 0.7142857142857143}

Recall: {'1': 0.8947368421052632, '2': 0.7619047619047619, '3': 0.7142857142857143}

Medida-F: {'1': 0.8500000000000001, '2': 0.8, '3': 0.7142857142857143}

Taxa de acerto: 0.7962962962962963

Taxa de Falsos Positivos (FP):
Classe 1: 0.1053
Classe 2: 0.2381
Classe 3: 0.2857
```

4) Compare com os resultados da questão anterior com resultados das métricas computados a partir de uma biblioteca que já calcula diretamente estas métricas.

a) Calcule a matriz de confusão.
b) Calcule o Recall por classe
c) Calcule a Taxa de acerto do classificador.
d) Calcule a Precisão por classe.
e) Calcule a Medida-F por classe.
f) Calcule a Taxa de FP por classe.

```python
import pandas as pd
import numpy as np
from sklearn.model_selection import train_test_split
from sklearn.neighbors import KNeighborsClassifier
from sklearn.metrics import confusion_matrix
from sklearn.metrics import precision_recall_fscore_support
from sklearn.metrics import accuracy_score

cols = ['class', 'alcohol', 'malic_acid', 'ash', 'alcalinity_of_ash', 'magnesium', 'total_phenols', 'flavanoids',
        'nonflavanoid_phenols', 'proanthocyanins', 'color_intensity', 'hue', 'OD280/OD315_of_diluted_wines', 'proline']

data = pd.read_csv('wine.data', header=None, names=cols)

X_train, X_test, y_train, y_test = train_test_split(data[cols[1:]], data['class'], test_size=0.3, random_state=42)

knn = KNeighborsClassifier(n_neighbors=1)
knn.fit(X_train, y_train)
predict = knn.predict(X_test)

confusionMatrix = confusion_matrix(y_test, predict)

precision, recall, fscore, support = precision_recall_fscore_support(y_test, predict)

accuracy = accuracy_score(y_test, predict)

print('Matriz de confusão:\n', confusionMatrix)
print('\nPrecisão:', precision)
print('\nRecall:', recall)
print('\nMedida-F:', fscore)
print('\nTaxa de acerto:', accuracy)

print('\nTaxa de Falsos Positivos (FP):')
for i in range(3):
  total = confusionMatrix[i].sum()
  fp = total - confusionMatrix[i, i]
  fp_rate = fp / total
  print(f"Classe {i+1}: {fp_rate:.4f}")
```

```
Matriz de confusão:
 [[17  0  2]
 [ 3 16  2]
 [ 1  3 10]]

Precisão: [0.80952381 0.84210526 0.71428571]

Recall: [0.89473684 0.76190476 0.71428571]

Medida-F: [0.85       0.8        0.71428571]

Taxa de acerto: 0.7962962962962963

Taxa de Falsos Positivos (FP):
Classe 1: 0.1053
Classe 2: 0.2381
Classe 3: 0.2857
```

5) Para cada um dos problemas abaixo, responda:
    a) Entre Precisão e Recall, indique qual métrica de avaliação você acha mais adequada para cada um dos seguintes problemas. A métrica deve ser calculada para a classe positiva.
    b) Quando sua resposta foi Precisão indique o que significaria maximizar o Recall, quando sua resposta foi Recall indique o significaria maximizar a Precisão.

        i) Login por impressão digital (fingerprint) em um dispositivo móvel. Nesta aplicação é aconselhável que o usuário consiga logar o maior número possível de vezes mesmo correndo o risco de um pequena probabilidade de outra pessoa consiga se passar por ele. Classe positiva: a impressão digital é do usuário. Classe negativa: a impressão digital não é do usuário.
        Recall. Porque aqui ele quer que o usuário possa logar com o máximo de tentativas, mesmo que isso signifique errar um pouquinho. Maximizar a precisão significa que sempre que a impressão digital fosse do usuário teríamos certeza que ela é do usuário, mas, às vezes, a impressão digital do usuário poderia não funcionar, sendo atribuída a classe errada.

        ii) Classicar o email como SPAM (Sending and Posting Advertisement in Mass) mas evitando que mensagens importantes sejam enviadas para a lixeira. Isto pode ter como consequência que alguns SPAMs vão chegar à caixa de entrada. Classe positiva: é SPAM. Classe negativa: não é SPAM.
        Precisão. Maximizar o recall significa que e-mails que não são spam podem acabar indo parar na lixeira, porque não vamos ter total certeza que a classe está correta.

        iii) Classificar uma fruta como saudável ou doente. Escolher o máximo de frutas saudáveis mesmo que alguma fruta doente apareça no meio daquelas selecionadas. Classe positiva: saudável. Classe negativa: doente.

Recall. Maximizar a precisão significa que frutas saudáveis podem acabar indo parar no lixo, pois só vamos classificar como saudável quando tivermos total certeza que ela é saudável, então algumas saudáveis podem acabar sendo jogadas fora.

iv) Detecção de faces, dizer que uma imagem é uma face humana somente quando tiver muita certeza de que é uma face. Deve-se evitar dizer que uma imagem é uma face quando não é de fato. Classe positiva: é uma face. Classe negativa: não é uma face.
Precisão. Maximizar o recall significa que poderíamos ter erros na hora da classificação e deixar passar imagens que não são de faces humanas, o que não é admitido neste problema.

6) A Figura 2 mostra três curvas ROC. Quais justificativas você teria para:
   a) Escolher o classificador B em detrimento do classificador A.
   Olhando para a área ROC o classificador B tem uma área muito maior que a do A, a taxa de verdadeiro positivo do B é muito melhor que a do A, e também a taxa de falso positivo do B é menor.

   b) Escolher o classificador B em detrimento do classificador C.
   O classificador B tem uma área ROC maior, e ele tem uma taxa de verdadeiro positivo melhor na grande maioria do tempo, o que torna ele o melhor classificador aqui.

   c) Escolher o classificador C em detrimento do classificador B.
   O classificador C é mais específico, pra escolher ele temos que querer uma taxa de falso positivo bem pequena, onde naquele intervalinho ele vai ser melhor, porque vai ter também uma taxa de verdadeiro positivo maior, mas a área ROC ainda vai continuar sendo menor que a do classificador B.