Lab 02 - Aprendizagem de Maquina

Helena Regina Salomé D'Espindula

2024-08-31

Lab 02 - Aprendizagem de Maquina

Codigo base:

```
#!/usr/bin/python
# -*- encoding: iso-8859-1 -*-

import sys
import numpy as np
import time

from sklearn import linear_model
from sklearn.neighbors import KNeighborsClassifier
from sklearn.discriminant_analysis import LinearDiscriminantAnalysis

from sklearn.metrics import confusion_matrix
from sklearn.datasets import load_svmlight_file

from sklearn.naive_bayes import BernoulliNB
from sklearn.naive_bayes import GaussianNB
from sklearn.naive_bayes import MultinomialNB
from sklearn.linear_model import Perceptron



def main(X_train, y_train, X_test, y_test, history):

    # cria o classificador - ESCOLHER UM
    #clf = KNeighborsClassifier(n_neighbors=3, metric='euclidean')
    #clf = linear_model.LogisticRegression()
    #clf = GaussianNB()
    #clf = LinearDiscriminantAnalysis()

    X_train_dense = X_train.toarray()
    clf.fit(X_train_dense, y_train)

    X_test_dense = X_test.toarray()
    y_pred = clf.predict(X_test_dense)

    # mostra o resultado do classificador na base de teste
    history.append(clf.score(X_test_dense, y_test))
```

```
    # cria a matriz de confusao
    #cm = confusion_matrix(y_test, y_pred)
    #print cm

    #print y_predProb

if __name__ == "__main__":
    #if len(sys.argv) != 3:
    #   sys.exit("Use: lab2.py <dataTR> <dataTS>")
    arq_test = "test.txt"
    arq_train = "train.txt"

    # loads data
    print ("Loading data...")
    X_train, y_train = load_svmlight_file(arq_test)
    X_test, y_test = load_svmlight_file(arq_train)
    size = X_train.shape
    batchsize = 1000
    ini = 0
    end = batchsize

    history = []

    while(end <= size[0] ):
        xt = X_train[0:end]
        yt = y_train[0:end]
        print ("Training size... ", end)
        start_time = time.time()
        main(xt, yt, X_test, y_test, history)
        print("--- %s seconds ---" % (time.time() - start_time))
        end = end + batchsize

    print (history)
```

Regressão Logistica

Utilizando a linha *clf = linear_model.LogisticRegression()*

```
Loading data...
Training size...  1000
--- 0.17670392990112305 seconds ---
Training size...  2000
--- 0.16291046142578125 seconds ---

[...]

Training size...  58000
--- 7.998165369033813 5 seconds ---
[0.7393, 0.8624, 0.8943, 0.9000, 0.9129, 0.9197,
0.9242, 0.9283, 0.9280, 0.9314, 0.9335, 0.9362,
0.9352, 0.9368, 0.9372, 0.9370, 0.9388, 0.9395,
```

```
0.9404, 0.9392, 0.9392, 0.9395, 0.9415, 0.9415,
0.9418, 0.9428, 0.9428, 0.9432, 0.9431, 0.9424,
0.9441, 0.9440, 0.9451, 0.9449, 0.9458, 0.9455,
0.9459, 0.9447, 0.9457, 0.9458, 0.9455, 0.9452,
0.9460, 0.9461, 0.9459, 0.9464, 0.9462, 0.9466,
0.9466, 0.9465, 0.9471, 0.9466, 0.9467, 0.9463,
0.9467, 0.9469, 0.9465, 0.9474]
```

K-nearest-neighbor (KNN)

Utilizando a linha *clf = KNeighborsClassifier(n_neighbors=3, metric='euclidean')*

```
Loading data...
Training size...  1000
--- 2.380507230758667 seconds ---
Training size...  2000
--- 2.8666324615478516 seconds ---

[...]

Training size...  58000
--- 32.63629508018494 seconds ---
[0.7532, 0.8995, 0.9194, 0.9252, 0.9354, 0.9333,
0.9363, 0.9369, 0.9367, 0.9368, 0.9394, 0.94,
0.9403, 0.9425, 0.9431, 0.9445, 0.945, 0.9459,
0.9458, 0.945, 0.9451, 0.9452, 0.9478, 0.9496,
0.9506, 0.9511, 0.9517, 0.9516, 0.9518, 0.9513,
0.9516, 0.9516, 0.9531, 0.9527, 0.9536, 0.9533,
0.9531, 0.9532, 0.9533, 0.9541, 0.9544, 0.9538,
0.9538, 0.954, 0.9541, 0.9547, 0.9544, 0.9542,
0.9552, 0.9557, 0.956, 0.956, 0.9555, 0.9558,
0.9553, 0.9556, 0.9554, 0.9557]
```

Gaussiana NB

Utilizando a linha *clf = GaussianNB()*

```
Loading data...
Training size...  1000
--- 0.31212711334228516 seconds ---
Training size...  2000
--- 0.309798002243042 seconds ---

[...]

Training size...  58000
--- 0.5339694023132324 seconds ---
[0.6558, 0.8556, 0.8691, 0.8796, 0.8971, 0.8941,
0.8958, 0.8948, 0.8958, 0.9001, 0.8988, 0.9001,
0.8984, 0.8993, 0.8995, 0.8977, 0.9017, 0.9004,
0.9018, 0.9031, 0.9078, 0.9074, 0.9097, 0.9088,
```

```
0.9097, 0.9109, 0.9109, 0.9112, 0.9112, 0.9112,
0.9109, 0.9111, 0.9115, 0.911, 0.9118, 0.9115,
0.9105, 0.9106, 0.9109, 0.9113, 0.9106, 0.9108,
0.9109, 0.9117, 0.9119, 0.9122, 0.9125, 0.9111,
0.9115, 0.9112, 0.9121, 0.9118, 0.912, 0.9124,
0.912, 0.9118, 0.9119, 0.9121]
```

Linear discriminant analysis (LDA)

Utilizando a linha *clf = LinearDiscriminantAnalysis()*

```
Loading data...
Training size...  1000
--- 0.08165097236633301 seconds ---
Training size...  2000
--- 0.09105205535888672 seconds ---

[...]

Training size...  58000
--- 2.976746082305908 seconds ---
[0.7908, 0.9055, 0.9169, 0.9182, 0.9275, 0.9295,
0.9305, 0.9325, 0.9319, 0.9316, 0.933, 0.9338,
0.9336, 0.9343, 0.9334, 0.9335, 0.933, 0.9339,
0.9343, 0.9341, 0.9338, 0.9334, 0.9334, 0.9336,
0.9344, 0.9349, 0.9355, 0.9349, 0.9346, 0.9345,
0.9348, 0.935, 0.9351, 0.9352, 0.9349, 0.9355,
0.9356, 0.9358, 0.936, 0.9357, 0.9357, 0.9358,
0.9351, 0.9353, 0.9349, 0.935, 0.9348, 0.9352,
0.9353, 0.9358, 0.9358, 0.9361, 0.9354, 0.9359,
0.9355, 0.9359, 0.9357, 0.9354]
```

Analise comparativa

```
batelada <- 1:58
lr_dados <- c(
  0.7393, 0.8624, 0.8943, 0.9000, 0.9129, 0.9197,
  0.9242, 0.9283, 0.9280, 0.9314, 0.9335, 0.9362,
  0.9352, 0.9368, 0.9372, 0.9370, 0.9388, 0.9395,
  0.9404, 0.9392, 0.9392, 0.9395, 0.9415, 0.9415,
  0.9418, 0.9428, 0.9428, 0.9432, 0.9431, 0.9424,
  0.9441, 0.9440, 0.9451, 0.9449, 0.9458, 0.9455,
  0.9459, 0.9447, 0.9457, 0.9458, 0.9455, 0.9452,
  0.9460, 0.9461, 0.9459, 0.9464, 0.9462, 0.9466,
  0.9466, 0.9465, 0.9471, 0.9466, 0.9467, 0.9463,
  0.9467, 0.9469, 0.9465, 0.9474
)
knn_dados <- c(
  0.7532, 0.8995, 0.9194, 0.9252, 0.9354, 0.9333,
  0.9363, 0.9369, 0.9367, 0.9368, 0.9394, 0.94,
```

```
  0.9403, 0.9425, 0.9431, 0.9445, 0.945, 0.9459,
  0.9458, 0.945, 0.9451, 0.9452, 0.9478, 0.9496,
  0.9506, 0.9511, 0.9517, 0.9516, 0.9518, 0.9513,
  0.9516, 0.9516, 0.9531, 0.9527, 0.9536, 0.9533,
  0.9531, 0.9532, 0.9533, 0.9541, 0.9544, 0.9538,
  0.9538, 0.954, 0.9541, 0.9547, 0.9544, 0.9542,
  0.9552, 0.9557, 0.956, 0.956, 0.9555, 0.9558,
  0.9553, 0.9556, 0.9554, 0.9557
)
gnb_dados <- c(
  0.6558, 0.8556, 0.8691, 0.8796, 0.8971, 0.8941,
  0.8958, 0.8948, 0.8958, 0.9001, 0.8988, 0.9001,
  0.8984, 0.8993, 0.8995, 0.8977, 0.9017, 0.9004,
  0.9018, 0.9031, 0.9078, 0.9074, 0.9097, 0.9088,
  0.9097, 0.9109, 0.9109, 0.9112, 0.9112, 0.9112,
  0.9109, 0.9111, 0.9115, 0.911, 0.9118, 0.9115,
  0.9105, 0.9106, 0.9109, 0.9113, 0.9106, 0.9108,
  0.9109, 0.9117, 0.9119, 0.9122, 0.9125, 0.9111,
  0.9115, 0.9112, 0.9121, 0.9118, 0.912, 0.9124,
  0.912, 0.9118, 0.9119, 0.9121
)
lda_dados <- c(
  0.7908, 0.9055, 0.9169, 0.9182, 0.9275, 0.9295,
  0.9305, 0.9325, 0.9319, 0.9316, 0.933, 0.9338,
  0.9336, 0.9343, 0.9334, 0.9335, 0.933, 0.9339,
  0.9343, 0.9341, 0.9338, 0.9334, 0.9334, 0.9336,
  0.9344, 0.9349, 0.9355, 0.9349, 0.9346, 0.9345,
  0.9348, 0.935, 0.9351, 0.9352, 0.9349, 0.9355,
  0.9356, 0.9358, 0.936, 0.9357, 0.9357, 0.9358,
  0.9351, 0.9353, 0.9349, 0.935, 0.9348, 0.9352,
  0.9353, 0.9358, 0.9358, 0.9361, 0.9354, 0.9359,
  0.9355, 0.9359, 0.9357, 0.9354
)
todos_dados <- data.frame("batelada" = batelada, "score" = c(lr_dados, knn_dados, gnb_dados, lda_dados)

head(todos_dados)
```

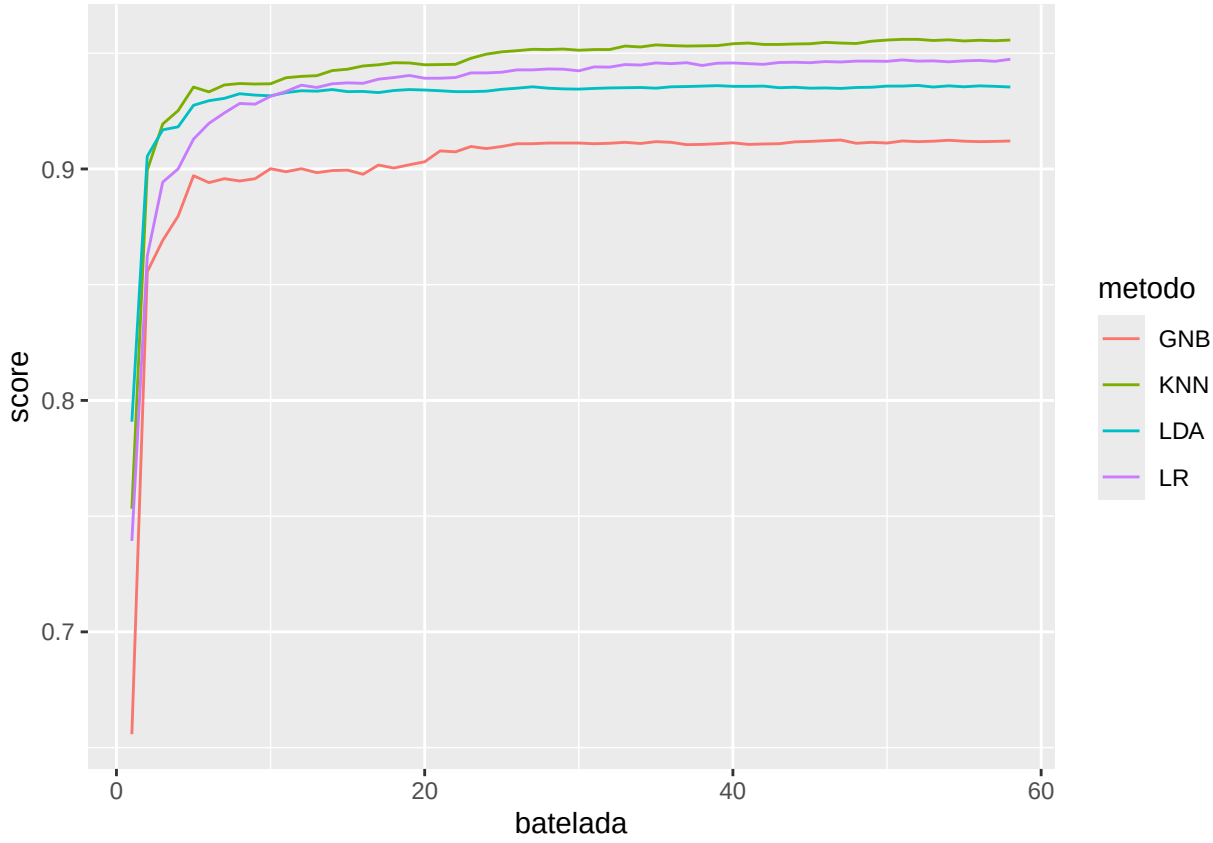
```
##   batelada  score metodo
## 1        1 0.7393     LR
## 2        2 0.8624     LR
## 3        3 0.8943     LR
## 4        4 0.9000     LR
## 5        5 0.9129     LR
## 6        6 0.9197     LR
```

```
require(ggplot2)
```

```
## Carregando pacotes exigidos: ggplot2
```

```
grafico <- ggplot(todos_dados, aes(x=batelada, y=score, color=metodo)) +
    geom_line(aes(group=metodo))

grafico
```




```
ggsave("test.png", grafico)
```

```
## Saving 6.5 x 4.5 in image
```