1. Постановка задачи

Для двух любых методов классификации из предыдущих работ и своего набора данных посчитать следующие метрики качества:

a. Точность классификации (Classification Accuracy)
b. Логарифм функции правдоподобия (Logarithmic Loss)
c. Область под кривой ошибок (Area Under ROC Curve)
d. Матрица неточностей (Confusion Matrix)
e. Отчет классификации (Classification Report)

2. Исходные данные

- Датасет: https://archive.ics.uci.edu/ml/datasets/Wine
- Предметная область: Состав вина разного географического происхождения
- Задача: определить, в какой из 3 областей произведено вино
- Количество записей: 178
- Количество атрибутов: 13
- Атрибуты:
  1. Алкоголь
  2. Малиновая кислота
  3. Зола
  4. Алкалиния золы
  5. Магний
  6. Всего фенолов
  7. Флаваноиды
  8. Нефлаваноидные фенолы
  9. Проантоцианы
  10. Интенсивность цвета
  11. Оттенок
  12. OD280 / OD315 разведенных вин
  13. пролин

2. Ход работы

```
import numpy as np
from sklearn.naive_bayes import GaussianNB
from sklearn.discriminant_analysis import LinearDiscriminantAnalysis as LDA
from sklearn.model_selection import cross_val_score
from sklearn import cross_validation
from sklearn.preprocessing import label_binarize
from sklearn.metrics import confusion_matrix
from sklearn.metrics import classification_report

dataset = np.loadtxt(open("data.csv","r"), delimiter=",", skiprows=0)
X = dataset[:,1:]
y =  (dataset[:,0]).astype(np.int64, copy=False)
kFold=cross_validation.KFold(n=len(X),n_folds=10, random_state=7, shuffle=True)

#Accuracy
print("Accuracy of methods:")
lda=LDA()
result = cross_val_score(lda, X, y, cv=kFold, scoring='accuracy')
#print(" - LDA: %0.5f (%0.5f)" % (result.mean(), result.std() ))
```

```
print(" LDA:")
print (" - mean: %0.5f" % result.mean())
print (" - standart deviation: %0.5f" % result.std())
gnb = GaussianNB()
result = cross_val_score(gnb, X, y, cv=kFold, scoring='accuracy')
print(" Gaussian:")
print (" - mean: %0.5f" % result.mean())
print (" - standart deviation: %0.5f" % result.std())

#converter class values 1/2 -> 0/1
for i in range(len(y)):
    y[i]=y[i]-1

#Logarithmic Loss
print("Logarithmic Loss Results:")
result = cross_validation.cross_val_score(lda, X, y, cv=kFold,
scoring='neg_log_loss')
print(" LDA:")
print (" - mean: %0.5f" % result.mean())
print (" - standart deviation: %0.5f" % result.std())
result = cross_validation.cross_val_score(gnb, X, y, cv=kFold,
scoring='neg_log_loss')
print(" Gaussian:")
print (" - mean: %0.5f" % result.mean())
print (" - standart deviation: %0.5f" % result.std())

#Area Under ROC Curve
print("Area Under ROC Curve Results: ")
result = cross_validation.cross_val_score(lda, X, y, cv=kFold, scoring='roc_auc')
print(" LDA:")
print (" - mean: %0.5f" % result.mean())
print (" - standart deviation: %0.5f" % result.std())
result = cross_validation.cross_val_score(gnb, X, y, cv=kFold, scoring='roc_auc')
print(" Gaussian: %0.5f (%0.5f)" % (result.mean(), result.std() ))
print (" - mean: %0.5f" % result.mean())
print (" - standart deviation: %0.5f" % result.std())

#Confusion Matrix
X_train, X_test, Y_train, Y_test = cross_validation.train_test_split(X, y,
test_size=0.3, random_state=7)
print("Confusion Matrixes:")
gnb.fit(X_train, Y_train)
gnb_predicted = gnb.predict(X_test)
gnb_matrix = confusion_matrix(Y_test, gnb_predicted)
print(" - GaussianNB:")
print(gnb_matrix)
lda.fit(X_train,Y_train)
lda_predicted=lda.predict(X_test)
lda_matrix=confusion_matrix(Y_test,lda_predicted)
print(" - LDA:")
print(lda_matrix)

#Classification Report
print("Classification Reports:")
lda_r=classification_report(Y_test,lda_predicted)
print(' - LDA:')
print(lda_r)
gaus_r=classification_report(Y_test,gnb_predicted)
print(" - GaussianNB:")
print(gaus_r)
```

Результаты:

```
Accuracy of methods:
 LDA:
 - mean: 1.00000
 - standart deviation: 0.00000
 Gaussian:
 - mean: 0.98462
 - standart deviation: 0.03077
Logarithmic Loss Results:
 LDA:
 - mean: -0.01022
 - standart deviation: 0.01030
 Gaussian:
 - mean: -0.08955
 - standart deviation: 0.16711
Area Under ROC Curve Results:
 LDA:
 - mean: 1.00000
 - standart deviation: 0.00000
 Gaussian: 1.00000 (0.00000)
 - mean: 1.00000
 - standart deviation: 0.00000
Confusion Matrixes:
 - GaussianNB:
 [13  0  0]
 [ 0 24  0]
 [ 0  0 17]
 - LDA:
 [13  0  0]
 [ 0 23  1]
 [ 0  0 17]
Classification Reports:
 - LDA:
              precision    recall  f1-score   support

          1       1.00      1.00      1.00        13
          2       1.00      0.96      0.98        24
          3       0.94      1.00      0.97        17

avg / total       0.98      0.98      0.98        54

 - GaussianNB:
              precision    recall  f1-score   support

          1       1.00      1.00      1.00        13
          2       1.00      1.00      1.00        24
          3       1.00      1.00      1.00        17

avg / total       1.00      1.00      1.00        54
```

Согласно данным, полученным в ходе лабораторной работы, можно сделать вывод о высокой точности результатов, предоставляемых методами, и имеющих малую погрешность. Основываясь на матрице ошибок получено значение точности (precision), равное 0.98, а также полноты (recall), также равное 0.98. Эти значения подтверждают высокое качество получаемых результатов.