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: http://archive.ics.uci.edu/ml/datasets/Statlog+%28Heart%29
Предметная область: медицина
Задача: определить, присутствует ли сердечная болезнь или нет
Количество записей: 270
Количество атрибутов: 13
Атрибуты:
-- 1. age
-- 2. sex
-- 3. chest pain type (4 values)
-- 4. resting blood pressure
-- 5. serum cholestoral in mg/dl
-- 6. fasting blood sugar > 120 mg/dl
-- 7. resting electrocardiographic results (values 0,1,2)
-- 8. maximum heart rate achieved
-- 9. exercise induced angina
-- 10. oldpeak = ST depression induced by exercise relative to rest
-- 11. the slope of the peak exercise ST segment
-- 12. number of major vessels (0-3) colored by flourosopy
-- 13. thal: 3 = normal; 6 = fixed defect; 7 = reversable defect
Классы:
-- 14. Absence (1) or presence (2) of heart disease

3. Ход работы

```
from sklearn.naive_bayes import GaussianNB
from sklearn.discriminant_analysis import LinearDiscriminantAnalysis as LDA
import numpy as np
from sklearn.model_selection import cross_val_score

# чтение данных
dataset = np.loadtxt(open("heart.dat","r"), delimiter=",", skiprows=0)
X = dataset[:, 0 : -1] # атрибуты
y =  (dataset[:, -1]).astype(np.int64, copy=False) # классы
kFold=cross_validation.KFold(n=len(X),n_folds=10, random_state=7)

#metric Accuracy for GaussianNB and LDA
```

```
gnb = GaussianNB()
scores = cross_val_score(gnb, X, y, cv=kFold, scoring='accuracy')
print("Accuracy for GaussianNB: %0.3f (%0.3f)" % (scores.mean(), scores.std() ))

lda=LDA()
scores = cross_val_score(lda, X, y, cv=kFold, scoring='accuracy')
print("Accuracy for LDA: %0.3f (%0.3f)" % (scores.mean(), scores.std() ))

from sklearn.preprocessing import label_binarize
#from classes[1,2] invert to classes[0,1]
for i in range(len(y)):
    y[i]=y[i]-1



#metric Logarithmic Loss for GaussianNB and LDA
scores = cross_validation.cross_val_score(gnb, X, y, cv=kFold, scoring='neg_log_loss')
print("log_loss for GaussianNB: %0.3f (%0.3f)" % (scores.mean(), scores.std() ))

scores = cross_validation.cross_val_score(lda, X, y, cv=kFold, scoring='neg_log_loss')
print("log_loss for LDA: %0.3f (%0.3f)" % (scores.mean(), scores.std() ))

#metric ROC_auc for GaussianNB and LDA

scores = cross_validation.cross_val_score(gnb, X, y, cv=kFold, scoring='roc_auc')
print("auc for GaussianNB: %0.3f (%0.3f)" % (scores.mean(), scores.std() ))

scores = cross_validation.cross_val_score(lda, X, y, cv=kFold, scoring='roc_auc')
print("auc for LDA: %0.3f (%0.3f)" % (scores.mean(), scores.std() ))

#metric confusion matrix for GaussianNB and LDA
from sklearn.metrics import confusion_matrix
X_train, X_test, Y_train, Y_test = cross_validation.train_test_split(X, y, test_size=0.3, random_state=7)


gnb.fit(X_train, Y_train)
gnb_predicted = gnb.predict(X_test)
gnb_matrix = confusion_matrix(Y_test, gnb_predicted)
print("confusion matrix for 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("confusion matrix for LDA")
print(lda_matrix)
```

```
#metric classification report for GaussianNB and LDA
from sklearn.metrics import classification_report
gnb_report=classification_report(Y_test,gnb_predicted)
print('classification report for GaussianNB')
print(gnb_report)


lda_report=classification_report(Y_test,lda_predicted)
print('classification report for LDA')
print(lda_report)
```

Результаты:

```
Accuracy for GaussianNB: 0.841 (0.037)
Accuracy for LDA: 0.833 (0.034)

log_loss for GaussianNB: -0.605 (0.203)
log_loss for LDA: -0.408 (0.110)

auc for GaussianNB: 0.907 (0.043)
auc for LDA: 0.910 (0.043)
confusion matrix for GaussianNB
[[39 8]
[ 6 28]]
confusion matrix for LDA
[[40 7]
[ 7 27]]

classification report for GaussianNB

          precision    recall  f1-score   support


        0      0.87      0.83      0.85       47
        1      0.78      0.82      0.80       34

avg / total      0.83      0.83      0.83       81

classification report for LDA

          precision    recall  f1-score   support


        0      0.85      0.85      0.85       47
        1      0.79      0.79      0.79       34

avg / total      0.83      0.83      0.83       81
```

Вывод: Точность для двух алгоритмов очень высокая и погрешность маленькая. Точность по matrix confusion равна 0.82,а полнота равна 0.87, следовательно мало ложных срабатываний и мало ложный пропусков. Область под кривой ошибок auc показала высокие результаты, что говорит о качественности классификатора.