

Министерство науки и высшего образования Российской Федерации
Муромский институт
федерального государственного бюджетного образовательного учреждения
высшего образования
**«Владимирский государственный университет
имени Александра Григорьевича и Николая Григорьевича Столетовых»
(МИВлГУ)**

Факультет _____ ИТ _____

Кафедра _____ ИС _____

ПРАКТИЧЕСКАЯ РАБОТА №7

по _____ Специальным главам математики _____

Тема _____ Логистическая регрессия. Бинарная классификация с
_____ помощью сигмоиды

Руководитель

Щаников С. А.
(фамилия, инициалы)

(подпись) _____ (дата)

Студент _____ ИСм-121
(группа)

Минеев Р. Р.
(фамилия, инициалы)

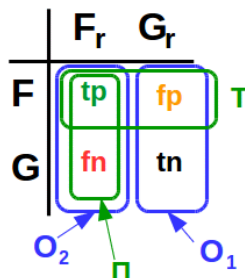
(подпись) _____ (дата)

Муром 2021

Практическая работа №7.

Тема: Логистическая регрессия. Бинарная классификация с помощью сигмоиды.

Задание на работу: реализовать бинарный классификатор и посчитать для неё полноту и точность, ошибки первого и второго рода, ROC AUC кривые.



Формулы для точности и полноты, соответственно:

$$Precision = \frac{TP}{TP + FP}$$

$$Recall = \frac{TP}{TP + FN}$$

Ошибка первого рода (O_1) — это вероятность того, что модель "определит" данные неправильно, при условии, что они правильные.

Ошибка второго рода (O_2) — это вероятность того, что модель "определит" данные как правильные, при условии, что они неверны.

$$O_1 = \frac{P(F_r)}{P(G_r)} \cdot \Pi \cdot \left(\frac{1}{T} - 1 \right)$$

$$O_2 = 1 - \Pi$$

```
from tensorflow import nn
from tensorflow.keras.models import Sequential
from tensorflow.keras.layers import Dense, Flatten
import numpy as np
from sklearn.metrics import roc_auc_score
from sklearn.metrics import roc_curve, auc
from matplotlib import pyplot as plt
```

					МИВУ 09.04.02-07.001			
Изм	Лист	№ докум.	Подп.	Дата	Практическая работа №7 Логистическая регрессия. Бинарная классификация с помощью сигмоиды	Литера	Лист	Листов
Студент	Минеев Р. Р.			08.01.		У	2	3
Руков.	Щаников С.А.					МИ ВлГУ ИСм-121		
Конс								
Н.контр.								
Утв.								

```

NUM_EPOCHS = 250

(trainX,      trainY),      (testX,      testY)      =      load_data('pr7_dataset.csv'),
load_data('pr7_dataset_test.csv')

model = Sequential([ Flatten(input_dim=3), Dense(10, activation=nn.sigmoid), Dense(2,
activation=nn.sigmoid)])

model.compile(optimizer="SGD", loss='sparse_categorical_crossentropy', metrics=['accuracy'])
model.fit(trainX, trainY, epochs=NUM_EPOCHS, batch_size=10, validation_data=(testX, testY))
print("Evaluate on test data")
'''

http://bazhenov.me/blog/2012/07/21/classification-performance-evaluation.html
'''

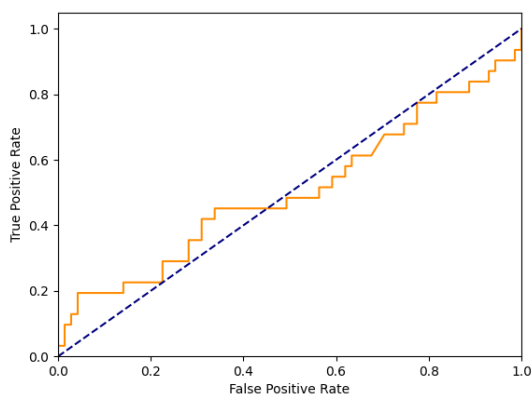
predictions = model.predict(testX)
# сохраняем вероятности только для исхода 1
probs = predictions[:, 1]
TP, FP, FN, TN = 0,0,0,0
for i in range(len(testY)):
    true = testY[i]
    pred = np.argmax(predictions[i])
    if true == pred:
        if true == 0: TP += 1
        else: TN += 1
    else:
        if true == 1: FP += 1
        else: FN += 1
recall = TP / (TP + FN)
precision = TP / (TP + FP)

O1 = ((testY==0).sum() / len(testY)) / ((testY==1).sum() / len(testY)) * recall *
(1/precision - 1)
O2 = 1 - recall

lr_auc = roc_auc_score(testY, probs)
print('LogisticRegression: ROC AUC=%.3f' % (lr_auc))
fpr, tpr, threshold = roc_curve(testY, probs)
roc_auc = auc(fpr, tpr)
print(f'recall: {recall}, precision: {precision}, O1: {O1}, O2: {O2}')
'''

LogisticRegression: ROC AUC=0.436
recall: 1.0, precision: 0.696078431372549, O1: 1.0000000000000002, O2: 0.0
'''

```



Вывод: в данной практической работе были получены навыки расчёта статистических характеристик созданного бинарного классификатора.

					МИВУ 09.04.02-07.001	Лист
						3
Изм	Лист	№ докум.	Подп.	Дата		