

Федеральное агентство связи

Федеральное государственное бюджетное образовательное учреждение
высшего образования «Сибирский государственный университет
телекоммуникаций и информатики»
(СибГУТИ)

Лабораторная работа №3

Выполнил: студент IV курса ИВТ,

гр. ИП-813

Бурдуковский И.А.

Проверила:

ассистент кафедры ПМиК

Морозова К.И.

Новосибирск, 2021 г.

Цель

Набор данных содержит в себе информацию о вариантах португальского вина "Винью Верде". Входные переменные представляют собой 13 столбцов со значениями, полученными на основе физико-химических тестов, а именно:

0 – цвет вина (“red” / ”white”)

1 - фиксированная кислотность

2 - летучая кислотность

3 - лимонная кислота

4 - остаточный сахар

5 - хлориды

6 - свободный диоксид серы

7 - общий диоксид серы

8 - плотность

9 - pH

10 - сульфаты

11 - спирт

Выходная переменная (на основе сенсорных данных):

12 - качество (оценка от 0 до 10, целое число)

Данные необходимо рассматривать как три набора. Данные для красного вина, данные для белого, общие данные вне зависимости от цвета. Необходимо построить модель для каждого из наборов, обучить её и сравнить полученные при помощи модели результаты с известными. Для обучения использовать 70% выборки, для тестирования 30%. Разбивать необходимо случайным образом, а, следовательно, для корректности тестирования качества модели, эксперимент необходимо провести не менее 10 раз и вычислить среднее значение качества регрессии.

Результат работы

Для выполнения данного задания была разработана программа с использованием библиотеки `sklearn`. Были считаны данные, обнулены все ячейки данных в которых есть неопределенные значения.

Было сделано 3 разделения данных в которые вошли сначала только белые, потом только красные и уже затем все вина.

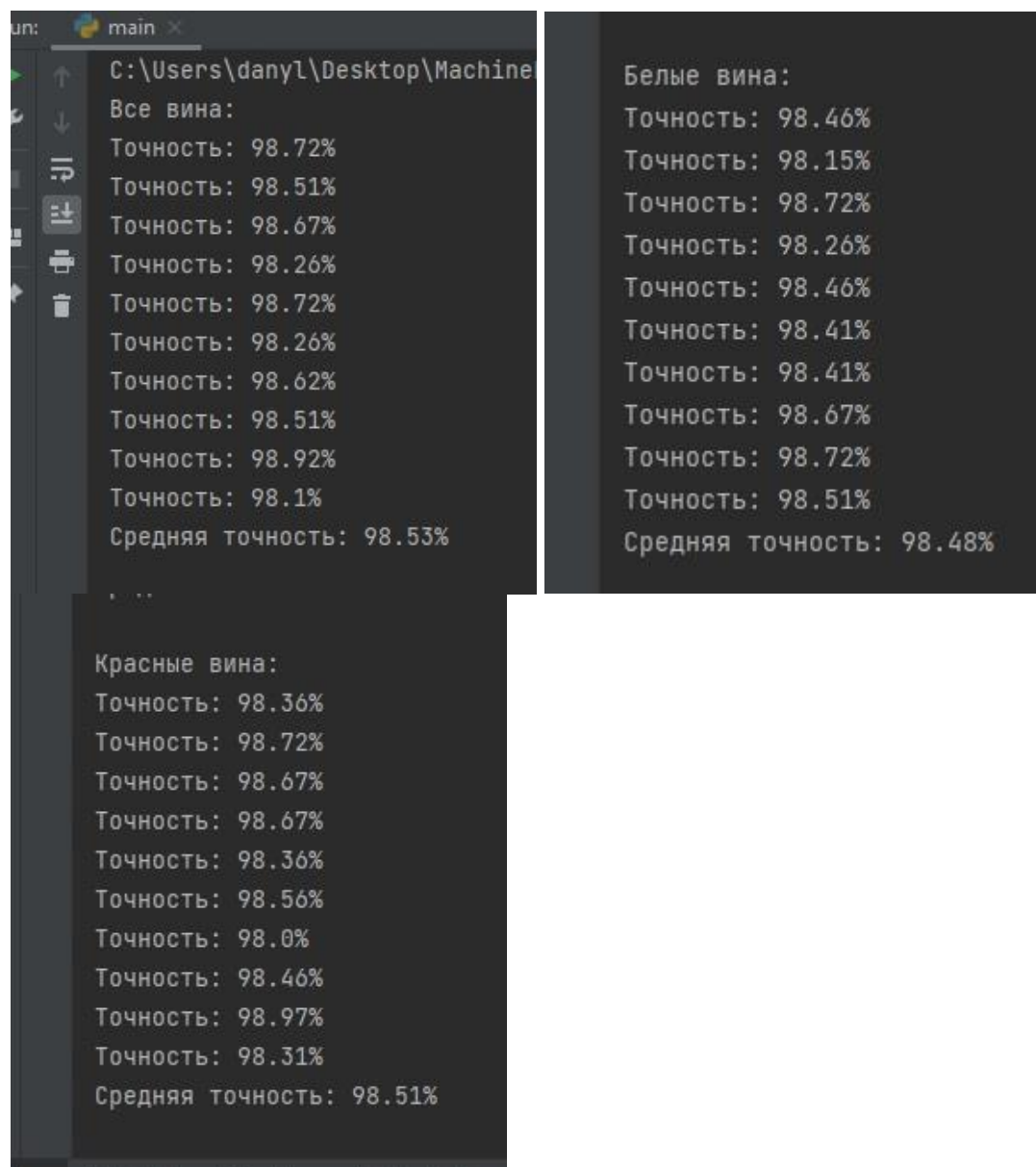
Далее в программе 3 раза был запущен цикл с разными разделениями, в котором идет разбиение на обучающую и тестовую выборки по 0.7 и 0.3 от общего количества данных соответственно. Инициализируется модель линейной регрессии LARS, модель обучается с помощью выборки, идет

предсказание по x_{est} и далее подсчитывается точность предсказания. Наша модель Lars с использованием Кросс-Валидации для автоматического определения коэффициента α .

За 10 прогонов точность модели для всех вин не опускалась ниже 83%, средняя за 10 – 84,81%;

Для белых вин не опускалась ниже 80% и средняя за 10 – 83.88%;

Для красных вин не опускалась ниже 85% и средняя за 10 – 88.96%.



```
main x
C:\Users\danyl\Desktop\Machine
Все вина:
Точность: 98.72%
Точность: 98.51%
Точность: 98.67%
Точность: 98.26%
Точность: 98.72%
Точность: 98.26%
Точность: 98.62%
Точность: 98.51%
Точность: 98.92%
Точность: 98.1%
Средняя точность: 98.53%

Белые вина:
Точность: 98.46%
Точность: 98.15%
Точность: 98.72%
Точность: 98.26%
Точность: 98.46%
Точность: 98.41%
Точность: 98.41%
Точность: 98.67%
Точность: 98.72%
Точность: 98.51%
Средняя точность: 98.48%

Красные вина:
Точность: 98.36%
Точность: 98.72%
Точность: 98.67%
Точность: 98.67%
Точность: 98.36%
Точность: 98.56%
Точность: 98.0%
Точность: 98.46%
Точность: 98.97%
Точность: 98.31%
Средняя точность: 98.51%
```

Рис.1 – Демонстрация точности работы программы.

Листинг программы

```
import pandas as pd
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LarsCV
from sklearn.preprocessing import normalize
from sklearn.pipeline import make_pipeline
from sklearn.preprocessing import StandardScaler

if __name__ == '__main__':
    white_wine = 0
    clf = pd.read_csv('winequalityN.csv',
header=0).fillna(0).values
    for i in clf:
        if i[0] == 'white':
            i[0] = 0
            white_wine += 1
        else:
            i[0] = 1
    x = clf[:, 0:12]
    y = clf[:, 12]

    for i in range(len(x[0])):
        x[:, i] = normalize([x[:, i]])

    print(f'Все вина:')
    result = 0
    for _ in range(10):
        x_train, x_test, y_train, y_test = train_test_split(x, y,
test_size=0.3)
        # model = make_pipeline(StandardScaler(with_mean=False),
LarsCV(normalize=False))
        model = LarsCV(normalize=False)
        model.fit(x_train, y_train)
        predict = model.predict(x_test)

        success = 0
        for i in range(len(x_test)):
            if abs(y_test[i] - predict[i]) < 1:
                success += 1
        print(f'Точность: {success / len(x_test) * 100:.4}%')
        result += success / len(x_test) * 100

    print(f'Средняя точность: {result / 10:.4}%\n')

    x1 = clf[0:white_wine, 0:12]
    y1 = clf[0:white_wine, 12]
    for i in range(len (x1[0])):
        x1[:, i] = normalize([x1[:, i]])
    print(f'Белые вина:')
    result = 0
    for _ in range(10):
```

```

        x_train, x_test, y_train, y_test = train_test_split(x, y,
test_size=0.3)
        # model = make_pipeline(StandardScaler(with_mean=False),
LarsCV())
        model = LarsCV(normalize=False)
        model.fit(x_train, y_train)
        prediction = model.predict(x_test)
        success = 0
        for i in range(len(x_test)):
            if abs(y_test[i] - prediction[i]) < 1:
                success += 1
        print(f'Точность: {success / len(x_test) * 100:.4}%')
        result += success / len(x_test) * 100

print(f'Средняя точность: {result / 10:.4}%\n')

x2 = clf[white_wine:, 0:12]
y2 = clf[white_wine:, 12]
for i in range(len(x2[0])):
    x2[:, i] = normalize([x2[:, i]])
print(f'Красные вина:')
result = 0
for _ in range(10):
    x_train, x_test, y_train, y_test = train_test_split(x, y,
test_size=0.3)
    # model = make_pipeline(StandardScaler(with_mean=False),
LarsCV())
    model = LarsCV(normalize=False)
    model.fit(x_train, y_train)
    prediction = model.predict(x_test)
    success = 0
    for i in range(len(x_test)):
        if abs(y_test[i] - prediction[i]) < 1:
            success += 1
    print(f'Точность: {success / len(x_test) * 100:.4}%')
    result += success / len(x_test) * 100

print(f'Средняя точность: {result / 10:.4}%\n')

```