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

Кафедра прикладной математики и кибернетики

Отчёт
по лабораторной работе № 3 «Метод линейной регрессии»

Выполнил:
студент группы ИП-013
Копытина Татьяна
Алексеевна

Проверил работу:
Дементьева Кристина Игоревна

Новосибирск 2023 г.

Оглавление

Задание:

Целью данной лабораторной работы является разработка программы, реализующей применение метода линейной регрессии к заданному набору данных.

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) Классы упорядочены и не сбалансированы (например, нормальных вин гораздо больше, чем отличных или плохих). В предоставленных данных есть пропуски и неточности.

В моем варианте нужно было:
Использовать модель LASSO

Задание: 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.

Особенности работы с данными:

1) Данные разнотипные, поэтому необходимо все столбцы привести к одному типу. Все данные должны быть вещественными числами. В данных есть пропуски, а это означает, что при считывании они будут записаны как NaN (либо произойдёт ошибка).

2) Результат работы модели будет тоже вещественным числом. Поэтому для оценки качества работы модели, необходимо использовать не прямое сравнение, а учитывать разницу между настоящим значением и смоделированным.

3) Данные в столбцах имеют разную размерность. Поэтому необходимо их нормализовать. Можно воспользоваться, например, методом preprocessing.normalize().

Код программы:

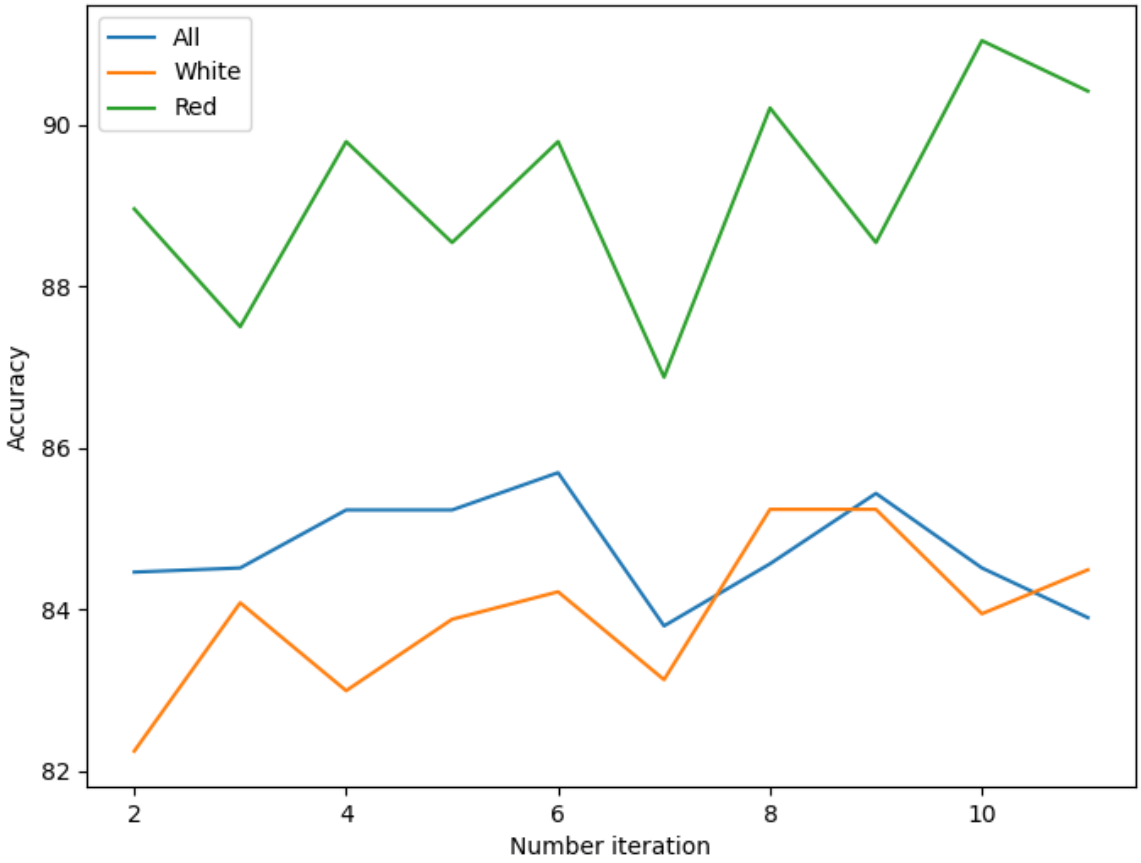
```
import pandas as pd
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LassoCV


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]



    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 = LassoCV(cv=5, 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]
    print(f'Белые вина:')
    result = 0
    for _ in range(10):
        x_train, x_test, y_train, y_test = train_test_split(x1, y1, test_size=0.3)
        model = LassoCV(cv=5, 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')


    x2 = clf[white_wine:, 0:12]
    y2 = clf[white_wine:, 12]
    print(f'Красные вина:')
    result = 0
    for _ in range(10):
        x_train, x_test, y_train, y_test =
train_test_split(x2, y2, test_size=0.3)
        model = LassoCV(cv=5,
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')
```

Результат работы программы:

```
RED:
ACCURACY: 88.96%
ACCURACY: 87.5%
ACCURACY: 89.79%
ACCURACY: 88.54%
ACCURACY: 89.79%
ACCURACY: 86.88%
ACCURACY: 90.21%
ACCURACY: 88.54%
ACCURACY: 91.04%
ACCURACY: 90.42%
AVERAGE ACCURACY: 89.17%
```

```
ALL:
ACCURACY: 84.46%
ACCURACY: 84.51%
ACCURACY: 85.23%
ACCURACY: 85.23%
ACCURACY: 85.69%
ACCURACY: 83.79%
ACCURACY: 84.56%
ACCURACY: 85.44%
ACCURACY: 84.51%
ACCURACY: 83.9%
AVERAGE ACCURACY: 84.73%

WHITE:
ACCURACY: 82.24%
ACCURACY: 84.08%
ACCURACY: 82.99%
ACCURACY: 83.88%
ACCURACY: 84.22%
ACCURACY: 83.13%
ACCURACY: 85.24%
ACCURACY: 85.24%
ACCURACY: 83.95%
ACCURACY: 84.49%
AVERAGE ACCURACY: 83.95%
```


All
White
Red
90
88
86
84
82
Accuracy
2
4
6
8
10
Number iteration

Выводы:

В ходе работы я изучила и применила на практике метод линейной регресии, используя библиотеку для машинного обучения sklearn.