Федеральное государственное бюджетное образовательное учреждение высшего образования

«Сибирский государственный университет

телекоммуникаций и информатики»

Кафедра ПМиК

Методы машинного обучения

Лабораторная работа №3

Линейная регрессия

Выполнил: студент 4 курса

Ф. ИВТ, группа: ИП-711

Мартасов И. О.

Проверил: доцент кафедры ПМиК

Ракитский Антон Андреевич

Новосибирск, 2020

**Содержание**

1. Задание.
2. Текст программы.
3. Результаты работы

**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.

Использовать модель LASSO

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.

В качестве результата выполненной лабораторной работы должна быть

разработанная программа, решающая поставленную задачу и отчёт с

содержанием текста программы, краткими комментариями и результатами

работы программы

**Текст программы**

import csv

import pandas

import numpy

from sklearn.linear\_model import Lasso

from sklearn.linear\_model import LassoCV

from sklearn.model\_selection import train\_test\_split

from sklearn.preprocessing import normalize

data = {'red': [], 'white': []}

y = {'red': [], 'white': []}

with open ('winequalityN.csv') as f:

r = csv.reader(f)

features = next(r)

for row in r:

row = [x if x != '' else 0 for x in row]

data[row[0]].append([float(x) for x in row[1:-1]])

y[row[0]].append(int(row[-1]))

optimal = 0.0

optimal\_alpha = 0.0

for i in range(0, 10):

x\_train\_red, x\_test\_red, y\_train\_red, y\_test\_red = train\_test\_split(data['red'],

y['red'], test\_size = 0.3)

lr = LassoCV(normalize=True).fit(x\_train\_red, y\_train\_red)

alpha = lr.alpha\_

predicted\_red = lr.predict(x\_test\_red)

test\_len\_red = len(x\_test\_red)

correct = 0

for i in range(test\_len\_red):

if(abs(y\_test\_red[i] - predicted\_red[i]) < 1):

correct += 1

print("RED:" + str((correct / test\_len\_red) \* 100))

print("ALPHA:" + str(alpha) + "\n")

if(optimal < (correct / test\_len\_red) \* 100):

optimal = (correct / test\_len\_red) \* 100

optimal\_alpha = alpha

print("OPTIMAL RED:" + str(optimal))

print("OPTIMAL ALPHA:" + str(optimal\_alpha) + "\n")

optimal = 0.0

optimal\_alpha = 0.0

for i in range(0, 10):

x\_train\_white, x\_test\_white, y\_train\_white, y\_test\_white = train\_test\_split(data['white'],

y['white'], test\_size = 0.3)

lw = LassoCV(normalize=True).fit(x\_train\_white, y\_train\_white)

alpha = lw.alpha\_

predicted\_white = lr.predict(x\_test\_white)

test\_len\_white = len(x\_test\_white)

correct = 0

for i in range(test\_len\_white):

if(abs(y\_test\_white[i] - predicted\_white[i]) < 1):

correct += 1

print("WHITE:" + str((correct / test\_len\_white) \* 100))

print("ALPHA:" + str(alpha) + "\n")

if(optimal < (correct / test\_len\_white) \* 100):

optimal = (correct / test\_len\_white) \* 100

optimal\_alpha = alpha

print("OPTIMAL WHITE:" + str(optimal))

print("OPTIMAL ALPHA:" + str(optimal\_alpha) + "\n")

optimal = 0.0

optimal\_alpha = 0.0

for i in range(0, 10):

x\_train\_both, x\_test\_both, y\_train\_both, y\_test\_both = train\_test\_split(data['red'] + data['white'],

y['red'] + y['white'], test\_size = 0.3)

lb = LassoCV(normalize=True).fit(x\_train\_both, y\_train\_both)

alpha = lb.alpha\_

predicted\_both = lr.predict(x\_test\_both)

test\_len\_both = len(x\_test\_both)

correct = 0

for i in range(test\_len\_both):

if(abs((y\_test\_both[i]) - predicted\_both[i]) < 1):

correct += 1

print("BOTH:" + str((correct / test\_len\_both) \* 100))

print("ALPHA:" + str(alpha) + "\n")

if(optimal < (correct / test\_len\_both) \* 100):

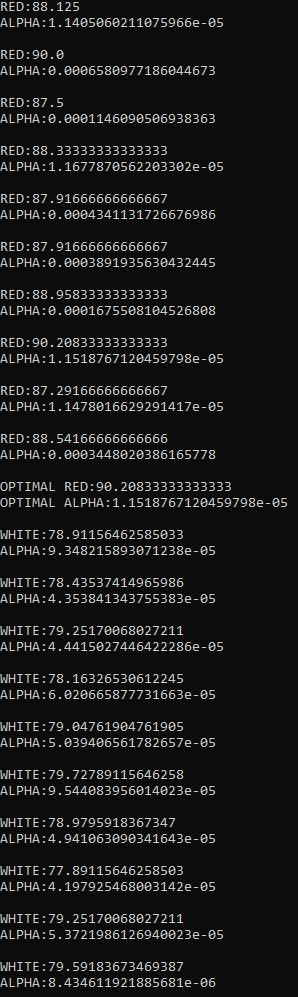
optimal = (correct / test\_len\_both) \* 100

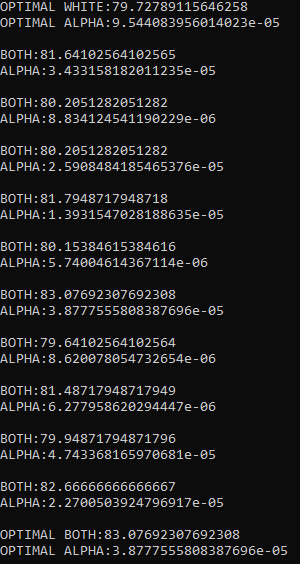
optimal\_alpha = alpha

print("OPTIMAL BOTH:" + str(optimal))

print("OPTIMAL ALPHA:" + str(optimal\_alpha) + "\n")

**Результаты работы**

****

****

**Оптимальный процент регресcии:**

**Красное вино: 90.2 %**

**Белое вино: 79.7 %**

**Оба вида вина: 83 %**