Министерство цифрового развития, связи и массовых коммуникаций Российской Федерации

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

**Отчёт по лабораторной работе №3**

**Метод линейной регрессии**

Выполнилa:

студент гр. ИП-813

Захарова К.Ю.

Проверилa:

Морозова К.И.

Новосибирск 2021

Оглавление

[Задание 3](#_Toc84969499)

[Результаты работы программы 5](#_Toc84969500)

[Код программы 6](#_Toc84969501)

# 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. Задания выполняются согласно варианту.

**Вариант задания:** 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:

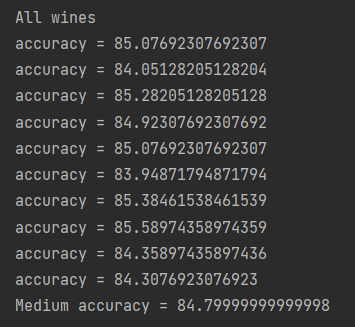
1) Данные разнотипные, поэтому необходимо все столбцы привести к одному типу. Все данные должны быть вещественными числами. В данных есть пропуски, а это означает, что при считывании они будут записаны как NaN (либо произойдёт ошибка).

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.

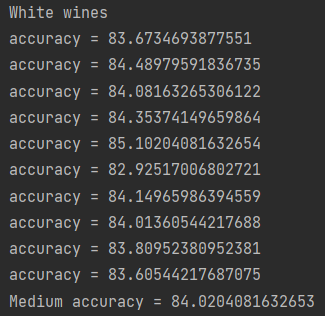
3) Данные в столбцах имеют разную размерность. Поэтому необходимо их нормализовать. Можно воспользоваться, например, методом preprocessing.normalize().



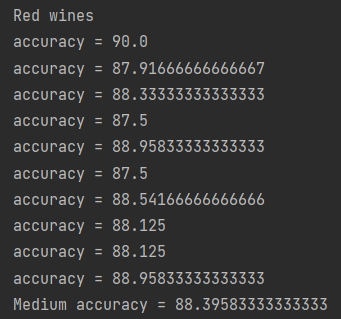
# Результаты работы программы



Независимо от цвета среднее значение: 85%.



У белых вин среднее значение: 84%.



У красных вин среднее значение: 88%.

# Код программы

import csv  
  
import numpy as np  
import sklearn as sk  
from sklearn import linear\_model  
from sklearn import preprocessing  
from sklearn.impute import SimpleImputer  
  
  
def calculate\_accuracy(dataset):  
 X = dataset[:, 1:-1]  
 y = dataset[:, -1]  
 X\_train, X\_test, y\_train, y\_test = sk.model\_selection.train\_test\_split(X, y, test\_size=0.3)  
 clf = linear\_model.LassoCV

clf.fit(X\_train, y\_train)   
 predicted = clf.predict(X\_test)   
 success = 0  
 for i in range(len(X\_test)):  
 if abs(y\_test[i] - predicted[i]) < 1:  
 success += 1  
 return success / len(X\_test) \* 100  
  
  
dataset = np.genfromtxt('winequalityN.csv', delimiter=',', skip\_header=True)  
  
with open('winequalityN.csv') as datafile:  
 next(datafile)  
 datareader = csv.reader(datafile, delimiter=',')  
 first\_col = []  
 whites = 0  
 for row in datareader:  
 first\_col.append(row[0])  
 if row[0] == 'white':  
 whites += 1  
  
  
le = preprocessing.LabelEncoder()  
first\_col = np.array([le.fit\_transform(first\_col)]).T   
dataset = np.hstack((first\_col, dataset))  
imp = SimpleImputer()  
imp.fit(dataset)  
dataset = imp.transform(dataset)  
for i in range(len(dataset[0]) - 1):  
 dataset[..., i] = preprocessing.normalize([dataset[..., i  
print('All wines')  
total = 0  
for \_ in range(10):  
 acc = calculate\_accuracy(dataset)  
 print('accuracy =', acc)  
 total += acc  
print('Medium accuracy =', total / 10)  
  
print('\nWhite wines')  
total = 0  
for \_ in range(10):  
 acc = calculate\_accuracy(dataset[:whites])  
 print('accuracy =', acc)  
 total += acc  
print('Medium accuracy =', total / 10)  
  
print('\nRed wines')  
total = 0  
for \_ in range(10):  
 acc = calculate\_accuracy(dataset[whites:])  
 print('accuracy =', acc)  
 total += acc  
print('Medium accuracy =', total / 10)