Федеральное государственное бюджетное образовательное учреждение высшего образования “Сибирский государственный университет телекоммуникаций и информатики”

Лабораторная работа по теме “ Метод линейной регрессии”

Выполнили:  
Студенты группы ИП-712  
Гольцев Дмитрий Сергеевич

Ладышкин Алексей Витальевич

Работу проверил:

ассистент кафедры ПМиК  
Морозова К.И.

Новосибирск 2020 г.

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.

Вывод

Результаты по всем трем наборам предоставлены на скриншотах, где первые 3 строчки - это Альфа, а оставшиеся - точность модели.

Листинг

import numpy as np  
import pandas as pd  
from sklearn.metrics import mean\_squared\_error, mean\_absolute\_error  
from sklearn.preprocessing import normalize  
from sklearn.impute import SimpleImputer  
from sklearn.model\_selection import cross\_val\_score, train\_test\_split  
from sklearn.model\_selection import RepeatedKFold  
from sklearn.linear\_model import Lars, LarsCV  
  
  
dataset = [pd.read](https://vk.com/away.php?to=http%3A%2F%2Fpd.read&cc_key=" \t "_blank)\_csv('winequalityN.csv')  
dataset.head()  
  
X, y = dataset.loc[:, 'fixed acidity':'alcohol'], dataset['quality']  
X = X.replace('', np.nan)  
y = y.replace('', np.nan)  
  
X\_white, y\_white = dataset.loc[dataset['type'] == 'white', 'fixed acidity':'alcohol'], dataset.loc[dataset['type'] == 'white', 'quality']  
X\_white = X\_white.replace('', np.nan)  
y\_white = y\_white.replace('', np.nan)  
  
X\_red, y\_red = dataset.loc[dataset['type'] == 'red', 'fixed acidity':'alcohol'], dataset.loc[dataset['type'] == 'red', 'quality']  
X\_red = X\_red.replace('', np.nan)  
y\_red = y\_red.replace('', np.nan)  
  
imputer = SimpleImputer()  
X = [imputer.fit](https://vk.com/away.php?to=http%3A%2F%2Fimputer.fit&cc_key=" \t "_blank)\_transform(X)  
X\_white = [imputer.fit](https://vk.com/away.php?to=http%3A%2F%2Fimputer.fit&cc_key=" \t "_blank)\_transform(X\_white)  
X\_red = [imputer.fit](https://vk.com/away.php?to=http%3A%2F%2Fimputer.fit&cc_key=" \t "_blank)\_transform(X\_red)  
X = normalize(X)  
X\_white = normalize(X\_white)  
X\_red = normalize(X\_red)  
  
X\_train, X\_test, y\_train, y\_test = train\_test\_split(X, y, test\_size=0.3)  
X\_white\_train, X\_white\_test, y\_white\_train, y\_white\_test = train\_test\_split(X\_white, y\_white, test\_size=0.3)  
X\_red\_train, X\_red\_test, y\_red\_train, y\_red\_test = train\_test\_split(X\_red, y\_red, test\_size=0.3)  
  
X\_train = [imputer.fit](https://vk.com/away.php?to=http%3A%2F%2Fimputer.fit&cc_key=" \t "_blank)\_transform(X\_train)  
X\_white\_train = [imputer.fit](https://vk.com/away.php?to=http%3A%2F%2Fimputer.fit&cc_key=" \t "_blank)\_transform(X\_white\_train)  
X\_red\_train = [imputer.fit](https://vk.com/away.php?to=http%3A%2F%2Fimputer.fit&cc_key=" \t "_blank)\_transform(X\_red\_train)  
X\_test = [imputer.fit](https://vk.com/away.php?to=http%3A%2F%2Fimputer.fit&cc_key=" \t "_blank)\_transform(X\_test)  
X\_white\_test = [imputer.fit](https://vk.com/away.php?to=http%3A%2F%2Fimputer.fit&cc_key=" \t "_blank)\_transform(X\_white\_test)  
X\_red\_test = [imputer.fit](https://vk.com/away.php?to=http%3A%2F%2Fimputer.fit&cc_key=" \t "_blank)\_transform(X\_red\_test)  
  
cv = RepeatedKFold(n\_splits=10, n\_repeats=5, random\_state=1)  
model = LarsCV(cv=cv, n\_jobs=-1)  
model\_white = LarsCV(cv=cv, n\_jobs=-1)  
model\_red = LarsCV(cv=cv, n\_jobs=-1)  
# model = Lars()  
# model\_white = Lars()  
# model\_red = Lars()  
[model.fit](https://vk.com/away.php?to=http%3A%2F%2Fmodel.fit&cc_key=)(X\_train, y\_train)  
[model\_white.fit](https://vk.com/away.php?to=http%3A%2F%2Fmodel_white.fit&cc_key=)(X\_white\_train, y\_white\_train)  
[model\_red.fit](https://vk.com/away.php?to=http%3A%2F%2Fmodel_red.fit&cc_key=)(X\_red\_train, y\_red\_train)  
y\_pred = model.predict(X\_test)  
y\_white\_pred = model\_white.predict(X\_white\_test)  
y\_red\_pred = model\_red.predict(X\_red\_test)  
print('alpha: %f' % model.alpha\_)  
print('alpha white: %f' % model\_white.alpha\_)  
print('alpha red: %f' % model\_red.alpha\_)  
print()  
  
# for i in range(len(y\_pred)):  
# print('Predicted: %.3f' % y\_pred[i])  
# print()  
# for i in range(len(y\_white\_pred)):  
# print('Predicted white: %.3f' % y\_white\_pred[i])  
# print()  
# for i in range(len(y\_red\_pred)):  
# print('Predicted red: %.3f' % y\_red\_pred[i])  
  
scores = cross\_val\_score(model, X\_train, y\_train, scoring='r2', cv=cv, n\_jobs=-1)  
scores\_white = cross\_val\_score(model\_white, X\_white\_train, y\_white\_train, scoring='r2', cv=cv, n\_jobs=-1)  
scores\_red = cross\_val\_score(model\_red, X\_red\_train, y\_red\_train, scoring='r2', cv=cv, n\_jobs=-1)  
scores = np.absolute(scores)  
scores\_white = np.absolute(scores\_white)  
scores\_red = np.absolute(scores\_red)  
  
# print('All data: %.3f' % (1 - mean\_squared\_error(y\_test, y\_pred)))  
# print('White wine: %.3f' % (1 - mean\_squared\_error(y\_white\_test, y\_white\_pred)))  
# print('Red wine: %.3f' % (1 - mean\_squared\_error(y\_red\_test, y\_red\_pred)))  
print('All wine: train = %.3f; std = %.3f | test = %.3f' % (np.mean(scores), np.std(scores), model.score(X\_test, y\_test)))  
print('White wine: train = %.3f; std = %.3f | test = %.3f' % (np.mean(scores\_white), np.std(scores\_white), model\_white.score(X\_white\_test, y\_white\_test)))  
print('Red wine: train = %.3f; std = %.3f | test = %.3f' % (np.mean(scores\_red), np.std(scores\_red), model\_red.score(X\_red\_test, y\_red\_test)))

Скриншоты





