Федеральное государственное бюджетное образовательное учреждение высшего образования

«Сибирский государственный университет телекоммуникаций и информатики»

(СибГУТИ)

Кафедра прикладной математики и кибернетики

Отчет

по лабораторной работе №3

Выполнил:

студент гр. ИП-712

Алексеев С.В.

Проверил:

Ассистент кафедры  
Морозова К.И.

Новосибирск, 2020 г.

# Задание

Разработка программы, реализующей применение метода линейной регрессии, а именно модели LASSO к заданному набору данных.

# Процесс выполнения работы

1. Данные были разбиты на обучающий и тестовый наборы 10 раз с соотношением 7:3
2. На обучающем наборе данных была обучена модель LASSO.
3. На тестовом наборе данных было протестирована обученная модель.

* Точность на 10 разбиениях для всего вина составила:  
  84.87, 85.28, 85.28, 85.74, 85.59, 84.31, 84.46, 85.54, 85.08, 83.08. Таким образом средняя точность составила 84.92.
* Точность на 10 разбиениях для белого вина составила:  
  83.4, 84.22, 84.76, 84.35, 83.47, 84.35, 83.95, 84.76, 83.74, 84.69. Таким образом средняя точность составила 84.17.
* Точность на 10 разбиениях для красного вина составила:  
  90.21, 89.58, 90.42, 89.17, 89.58, 89.38, 87.92, 89.17, 88.33, 89.58. Таким образом средняя точность составила 89.33.

# Вывод

В лабораторной работе был реализован метод линейной регрессии модели LASSO. Для данного набора данных метод показывает хорошие результаты. Для красного вина данная модель отрабатывает с чуть лучшими результатами, чем для белого.

# Листинг

|  |
| --- |
| **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**)** |