ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ
ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ
«СИБИРСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ
ТЕЛЕКОММУНИКАЦИЙ И ИНФОРМАТИКИ»

ЛАБОРАТОРНАЯ РАБОТА № 3
Регрессивный анализ

Выполнил:
Студент группы: ИП-715
Винтер А.В.

Проверил: ассистент кафедры ПМиК
Морозова К.И.

Новосибирск – 2020

Оглавление

Текст задания.

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.

Вариант задания:
2) 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().

$$x_{norm} = \frac{x - x_{min}}{x_{max} - x_{min}} \in [0,1]$$

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.

Описание основных функций

Регрессионный анализ — метод моделирования измеряемых данных и исследования их свойств. Данные состоят из пар значений зависимой переменной (переменной отклика) и независимой переменной (объясняющей переменной).

Регрессия — зависимость математического ожидания (например, среднего значения) случайной величины от одной или нескольких других случайных величин (свободных переменных)

Разности между фактическими значениями зависимой переменной и восстановленными называются регрессионными остатками (residuals). В литературе используются также синонимы: невязки и ошибки.

pandas - это Python библиотека для анализа и обработки данных. Она действительно быстрая и позволяет вам легко исследовать данные.

В Python sklearn — это пакет, который содержит все необходимые пакеты для реализации алгоритма машинного обучения.

Lasso (Least absolute shrinkage and selection operator) - метод оценивания коэффициентов линейной регрессионной модели

Метод заключается во введении ограничения на норму вектора коэффициентов модели, что приводит к обращению в 0 некоторых коэффициентов модели. Метод приводит к повышению устойчивости модели в случае большого числа обусловленности матрицы признаков X, позволяет получить интерпретируемые модели - отбираются признаки, оказывающие наибольшее влияние на вектор ответов.

функция:

```
train_test_split(x, y, test_size=0.3, random_state=randint(0, 10000))
```

test_size : float, int или None, необязательно (по умолчанию=None), если float, должно быть между 0.0 и 1.0 и представлять долю набора данных, включаемого в тестовое разделение

random_state: Управляет перетасовкой, применяемой к данным перед применением разделения

функция dt.fit(x_train, y_train):

Экземпляр оценки dt (для классификатора) сначала устанавливается в модель; то есть он должен учиться у модели. Это делается путем

передачи нашего тренировочного набора в метод fit. Для обучающего набора мы будем использовать все изображения из нашего набора данных, за исключением последнего изображения, которое мы оставим для нашего прогнозирования. Мы выбираем обучающий набор с синтаксисом Python (x_train), который создает новый массив, содержащий все, кроме последнего элемента (y_train)

Результат работы программы

```
Общие данные не зависимо от цвета вина:
Точность: 85.23%
Точность: 84.72%
Точность: 85.03%
Точность: 84.62%
Точность: 84.05%
Точность: 85.18%
Точность: 86.46%
Точность: 85.64%
Точность: 84.41%
Точность: 85.69%
Среднее значение качества регрессии для 10 запусков: 85.1%


Для белого вина:
Точность: 85.17%
Точность: 84.29%
Точность: 84.76%
Точность: 84.63%
Точность: 85.17%
Точность: 83.95%
Точность: 84.15%
Точность: 84.9%
Точность: 85.85%
Точность: 84.22%
Среднее значение качества регрессии для 10 запусков: 84.71%
```

```
Для красного вина:
Точность: 88.12%
Точность: 86.88%
Точность: 87.08%
Точность: 85.62%
Точность: 89.79%
Точность: 89.17%
Точность: 86.67%
Точность: 85.83%
Точность: 90.0%
Точность: 90.21%
Среднее значение качества регрессии для 10 запусков: 87.94%
```

Код программы

```
from random import randint


import pandas as pd
from sklearn.linear_model import LassoCV
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import normalize


def main() -> int:
  size_white = 0

  data = pd.read_csv('winequalityN.csv', header=0).fillna(0).values
  for i in data:
    if i[0] == 'white':
      i[0] = 0
      size_white += 1
    else:
      i[0] = 1
  x = data[..., 0:12]
  y = data[..., 12]
  for i in range(len(x[0])):
    x[..., i] = normalize([x[..., i]])

  runs = 10
  print("\033[32m")
  print(f'Общие данные не зависимо от цвета вина:')

  result = 0
  for _ in range(runs):
    x_train, x_test, y_train, y_test = train_test_split(x, y, test_size=0.3, random
_state=randint(0, 10000))
    model = LassoCV()
    model.fit(x_train, y_train)
    prediction = model.predict(x_test)

    success = 0
    for i in range(len(x_test)):
      if abs(y_test[i] - prediction[i]) < 1:
        success += 1
    print(f'Точность: {success / len(x_test) * 100:.4}%')
    result += success / len(x_test) * 100

  print(f'Среднее значение качества регрессии для {runs} запусков: {result / runs:.
4}%\n')
  x_1 = data[0:size_white, 0:12]
  y_1 = data[0:size_white, 12]
  for i in range(len(x_1[0])):
    x_1[..., i] = normalize([x_1[..., i]])
  print("\033[0m")
```

```
  print("\033[33m")
  print(f'Для белого вина:')
  result = 0
  for _ in range(runs):
    x_train, x_test, y_train, y_test = train_test_split(x_1, y_1, test_size=0.3, ra
ndom_state=randint(0, 10000))
    model = LassoCV()
    model.fit(x_train, y_train)
    prediction = model.predict(x_test)

    success = 0
    for i in range(len(x_test)):
      if abs(y_test[i] - prediction[i]) < 1:
        success += 1
    print(f'Точность: {success / len(x_test) * 100:.4}%')
    result += success / len(x_test) * 100

  print(f'Среднее значение качества регрессии для {runs} запусков: {result / runs:.
4}%\n')

  x_2 = data[size_white:, 0:12]
  y_2 = data[size_white:, 12]
  for i in range(len(x_2[0])):
    x_2[..., i] = normalize([x_2[..., i]])
  print("\033[0m")

  print("\033[31m")
  print(f'Для красного вина:');

  result = 0
  for _ in range(runs):
    x_train, x_test, y_train, y_test = train_test_split(x_2, y_2, test_size=0.3, ra
ndom_state=randint(0, 10000))
    model = LassoCV()
    model.fit(x_train, y_train)
    prediction = model.predict(x_test)

    success = 0
    for i in range(len(x_test)):
      if abs(y_test[i] - prediction[i]) < 1:
        success += 1
    print(f'Точность: {success / len(x_test) * 100:.4}%')
    result += success / len(x_test) * 100

  print(f'Среднее значение качества регрессии для {runs} запусков: {result / runs:.
4}%\n')
  print("\033[0m")

if __name__ == '__main__':
  exit(main())
```