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

Регрессионный анализ

Выполнил:

студент гр. ИП-111

Кузьменок Д.В.

Проверил:

Старший преподаватель кафедры ПМиК

Дементьева К.И.

Новосибирск, 2024 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.

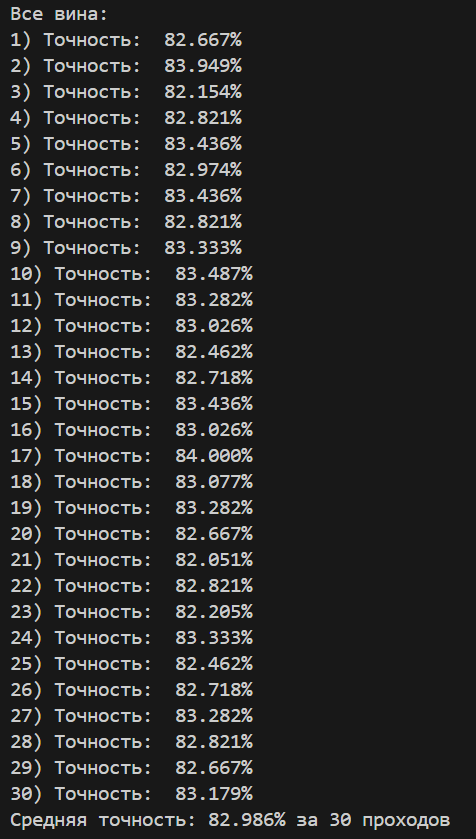
Вариант задания: использовать классическую модель LinearRegression.

Имя файла: winequalityN.csv.

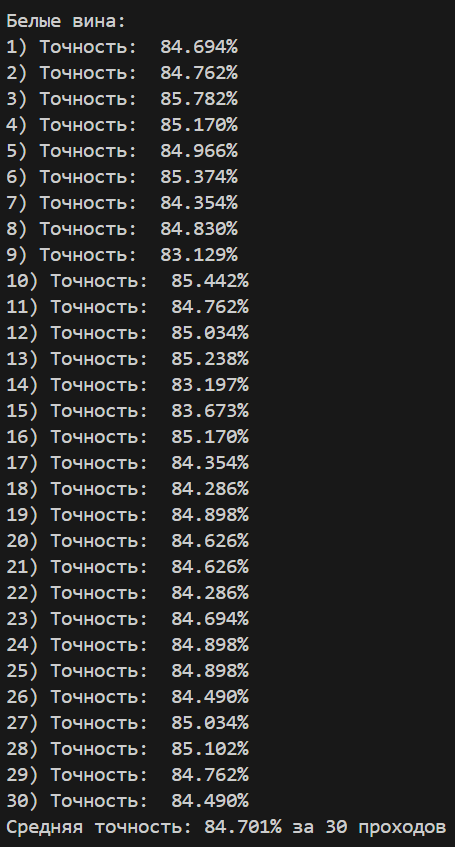
**Результаты**

Для проверки полученных результатов я запускаю 30 раз подсчет точности для вычисления средней точности угадывания класса.

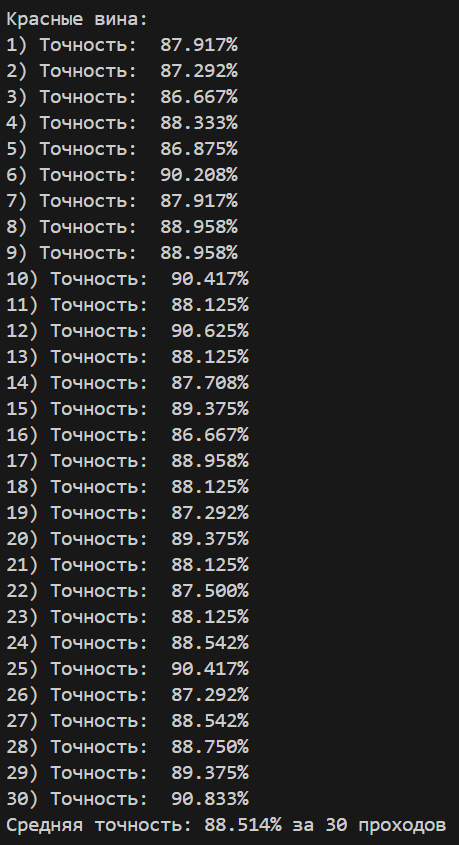
В случае всех вин:



В случае белых вин:



В случае красных вин:



**Код программы**

import pandas as pd

from sklearn import preprocessing

from sklearn.linear\_model import LinearRegression

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

def preprocess\_data(data):

    data\_X = preprocessing.normalize(data[:, :-1])

    data\_Y = data[:, -1]

    train\_x, test\_x, train\_y, test\_y = train\_test\_split(data\_X, data\_Y, test\_size=0.3, stratify=data\_Y)

    return train\_x, test\_x, train\_y, test\_y

def train\_linear\_regression(train\_x, train\_y):

    linear\_regression = LinearRegression()

    linear\_regression.fit(train\_x, train\_y)

    return linear\_regression

def evaluate\_model(model, test\_x, test\_y):

    predicted = model.predict(test\_x)

    success = 0

    for i in range(len(test\_x)):

        if abs(test\_y[i] - predicted[i]) < 1:

            success += 1

    return success / len(test\_x) \* 100

def run\_evaluation(data, n\_iterations):

    total\_accuracy = 0

    for i in range(n\_iterations):

        train\_x, test\_x, train\_y, test\_y = preprocess\_data(data)

        model = train\_linear\_regression(train\_x, train\_y)

        accuracy = evaluate\_model(model, test\_x, test\_y)

        print(f"{i + 1}) Точность:  {accuracy :.3f}%")

        total\_accuracy += accuracy

    return total\_accuracy / n\_iterations

def main():

    file\_name = "winequalityN.csv"

    n\_iterations = 30

    data = pd.read\_csv(file\_name, header=0).fillna(0)

    data.loc[data.type == 'white', 'type'] = 0

    data.loc[data.type == 'red', 'type'] = 1

    data = data.to\_numpy()

    print(f'\nВсе вина:')

    average\_accuracy = run\_evaluation(data, n\_iterations)

    print(f'Средняя точность: {average\_accuracy:.3f}% за {n\_iterations} проходов\n\n')

    print(f'Белые вина:')

    white\_data = data[data[:, 0] == 0]

    average\_accuracy = run\_evaluation(white\_data, n\_iterations)

    print(f'Средняя точность: {average\_accuracy:.3f}% за {n\_iterations} проходов\n\n')

    print(f'Красные вина:')

    red\_data = data[data[:, 0] == 1]

    average\_accuracy = run\_evaluation(red\_data, n\_iterations)

    print(f'Средняя точность: {average\_accuracy:.3f}% за {n\_iterations} проходов')

if \_\_name\_\_ == "\_\_main\_\_":

    main()