«КАЗАНСКИЙ (ПРИВОЛЖСКИЙ) ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ»

Лабораторная работа №1 по теме: «Логистическая регрессия»

Выполнил

Студент 1 курса

группы 09-115(3)

Зиновьев Е. А.

Преподаватель:

Шустова Е.П

Казань 2022

Задание:

1. Построение и применение однофакторной логистической регрессионной модели

Найдите реальные данные для задачи однофакторной бинарной классификации и выполните для них следующее:

- Найдите коэффициенты логистической регрессионной модели.

- постройте логистическую кривую и облако точек.

- выполните прогноз переменной отклика для нескольких новых объектов и нескольких объектов обучающей выборки.

- проиллюстрируйте на графике результаты прогнозирования.

- сделайте выводы о качестве логистической регрессионной модели.

- результаты оформите в виде Word-файла и прикрепите его здесь (весь Python-код поместите в конец файла как приложение).

1. Построение и применение множественной логистической регрессионной модели

Найдите реальные данные для задачи множественной бинарной классификации и выполните для них следующее:

- найдите коэффициенты логистической регрессионной модели.

- выполните прогноз переменной отклика для нескольких новых объектов и нескольких объектов обучающей выборки.

- сделайте выводы о качестве логистической регрессионной модели.

- результаты оформите в виде Word-файла и прикрепите его здесь (весь Python-код поместите в конец файла как приложение).

Выполнение работы:

1. В качестве датасета был выбран:

<https://github.com/harpreetSinghGuller/Data-Science-R/blob/master/SAHeart.csv>

Рассмотрим данные, касающиеся ишемической болезни сердца (ИБС) в Южной Африке. Цель состоит в том, чтобы использовать различные переменные, такие как употребление табака, семейный анамнез, уровень холестерина ЛПНП, употребление алкоголя, ожирение, тип поведения для выяснения наличия ишемической болезни сердца у человека.

Параметры:

sbp: систолическое артериальное давление

tobacco: совокупный табак (кг)

ldl: холестерин липопротеидов низкой плотности

adiposity: ожирение

famhist: семейная история болезни сердца (Присутствует, отсутствует)

typea: оценка поведения типа А, которое характеризуется интенсивным образом жизни (быстро есть, ходить, работать и тд, более подробно про влияние этого типа поведения на ишемическую болезнь сердца можно прочитать в книге Фридмана:

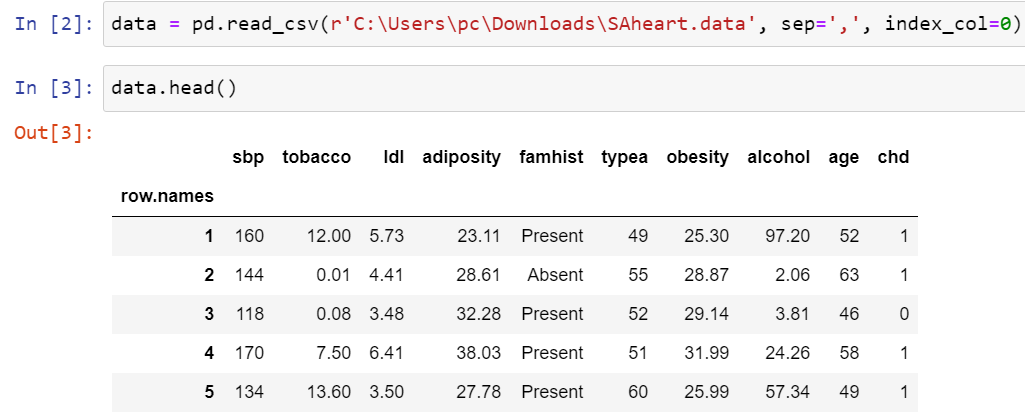
<https://www.amazon.com/Type-Behavior-Heart-Meyer-Friedman/dp/0394480112>

obesity: процент жира

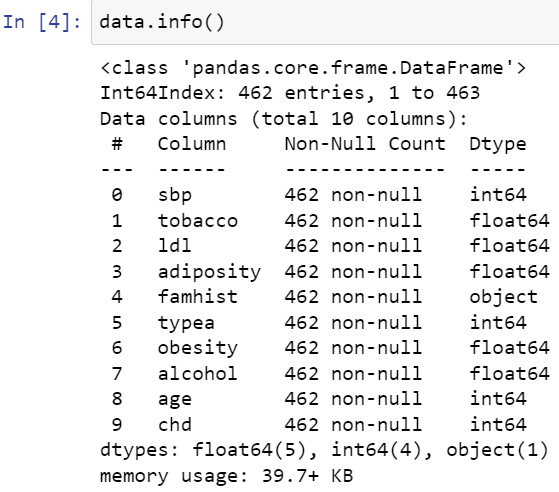
alcohol: текущее потребление алкоголя

age: возраст начала ишемической болезни

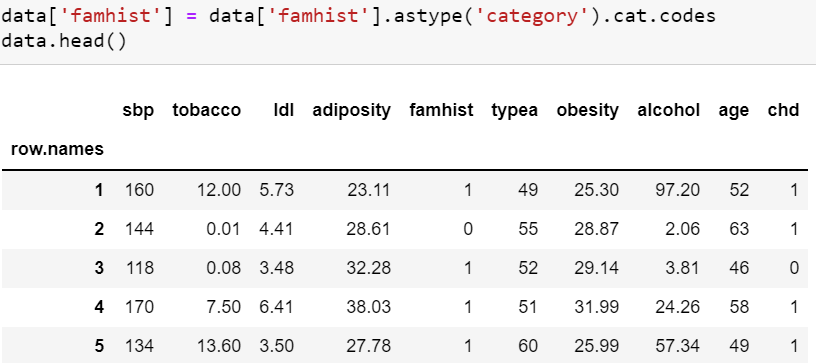
chd: ишемическая болезнь сердца (1, 0)



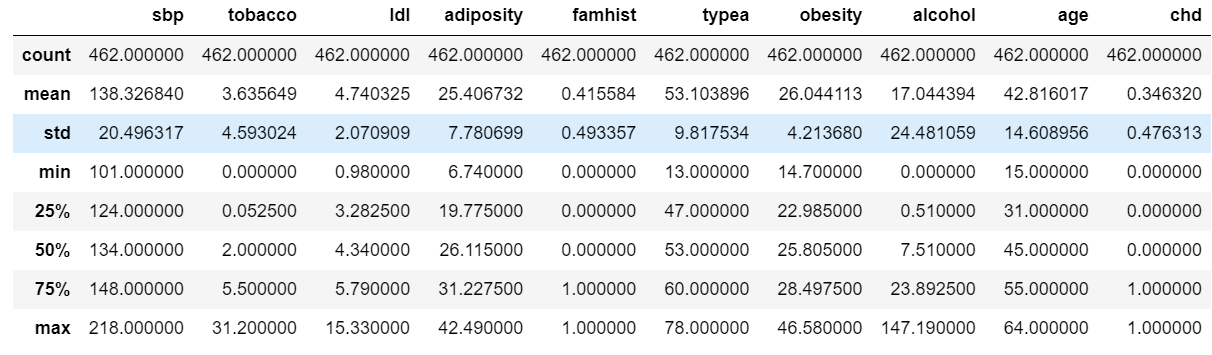
Как видим, здесь присутствует всего 2 категорийные переменные, одна из которых наша таргетная – наличие ишемической болезни сердца (chd), представленная целочисленным типом, вторая объектным:



Некоторые алгоритмы машинного обучения могут либо плохо работать с объектным представлением переменной, либо и вовсе не работать, поэтому преобразуем вторую категорийную переменную с помощью cat. codes кодирования, где теперь 1 значит наличие ишемической болезни у родни, а 0 - отсутствие:



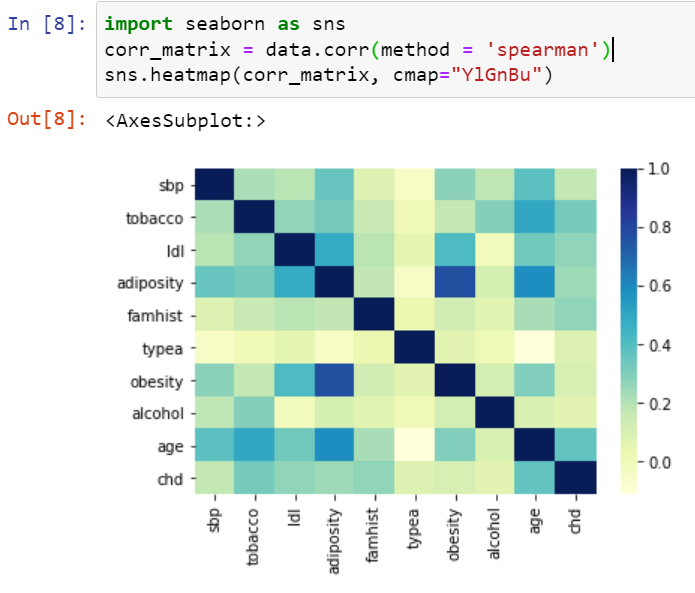
Можем посмотреть на основные описательные характеристики непрерывных переменных:



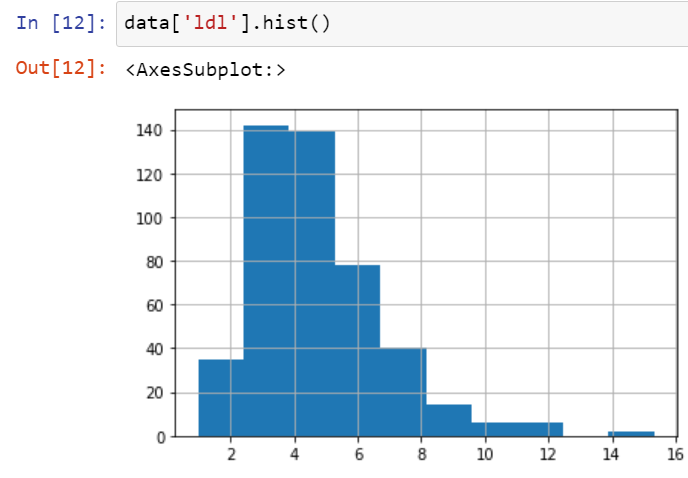
Как видим, выборка состоит из 462 человек, со средним давлением 138, средним возрастом начала ишемической болезни сердца 42 года, хотя было зарегистрировано и в 15 лет даже и тд.

Предобработка данных завершена, переходим к построению логистической регрессии.

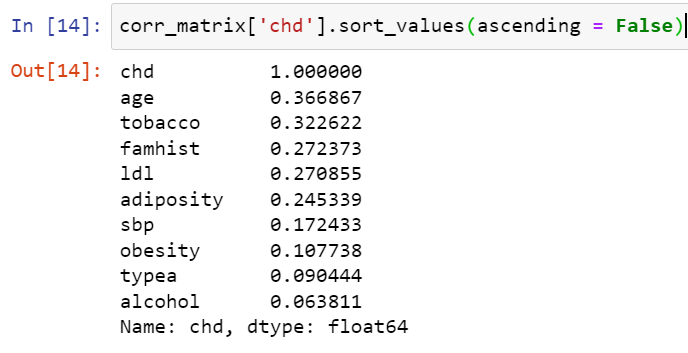
1. Для однофакторной регрессии выберем независимый параметр по тепловой карте у библиотеки seaborn:



Используем метод рассчета корреляций Спирмена, так как выборки не совсем нормально

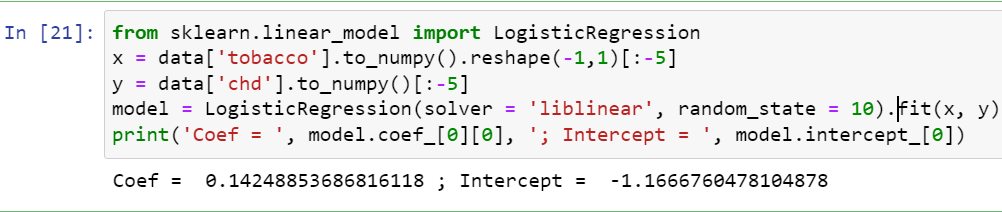
распределены, например, распределение давления имеет явную ассиметрию и крутую форму:

По хитмэпу видим очевидную связь ожирения с процентом жира, например. По коэффициентам корреляции таргетной переменной chd надо определить, какую переменную будем использовать:



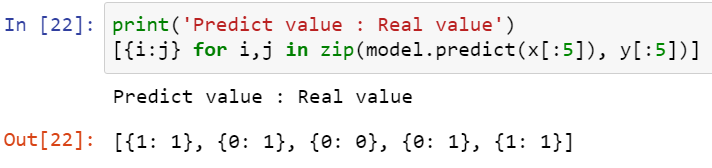
Наиболее сильная связь от переменной возраста, но в качестве инсайдерской информации она мало полезна, поэтому рассмотрим влияние табака на заболевание сердца.

Используем библиотеку sklearn, преобразуем сначала независимую и зависимую переменные к нампаевскому аррэю для подачи на вход модели импортированного модуля LogisticRegression, используем метод решения liblinear, так как он лучше всего работает всего с мелкими датасетами. Для вывода коэффициента при переменной tobacco используем атрибут coef\_ и свободного коэффициента intercept\_ (пересечение с осью у), но предварительно уберем 5 последних значений для дальнейшего использования их при тестировании, лучше бы, конечно, больше, но выборка и так очень мелкая:

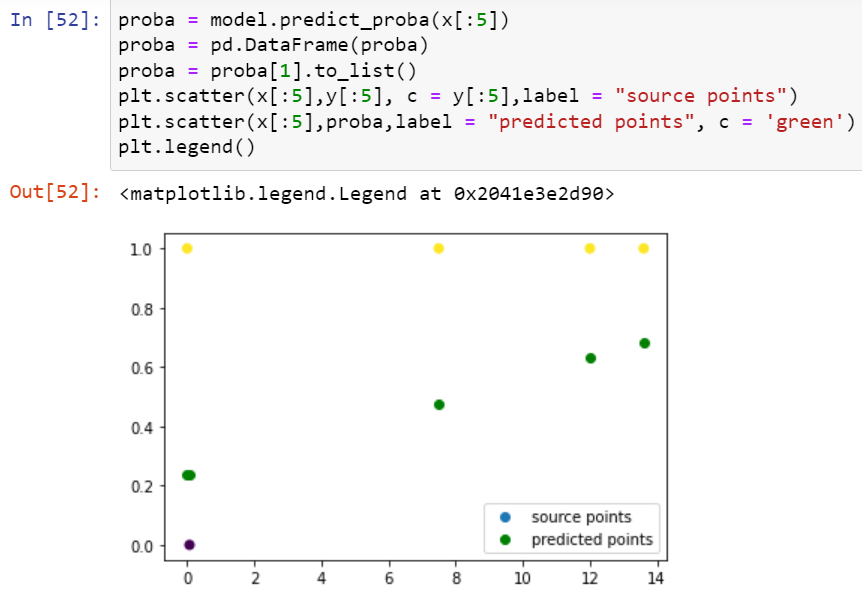


Это значит, что при увеличении количества употребляемого табака, например, в 2 раза, вероятность ишемической болезни сердца увеличивается на 0.28.

Посмотрим сначала на прогноз для строк, которые видела модель:

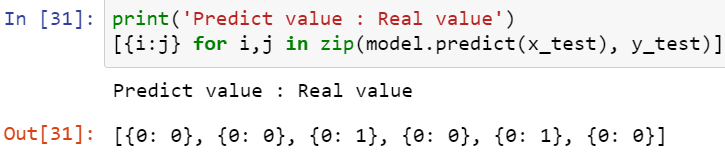


Было угадано 3 из 5 значений. Посмотрим на графике, какие вероятности были поставлены каждой точке, для этого используем атрибут predict\_proba и оставим только 1 столбец вероятности отнесения к одному классу, построим скаттерплот с помощью библиотеки matplotlib:

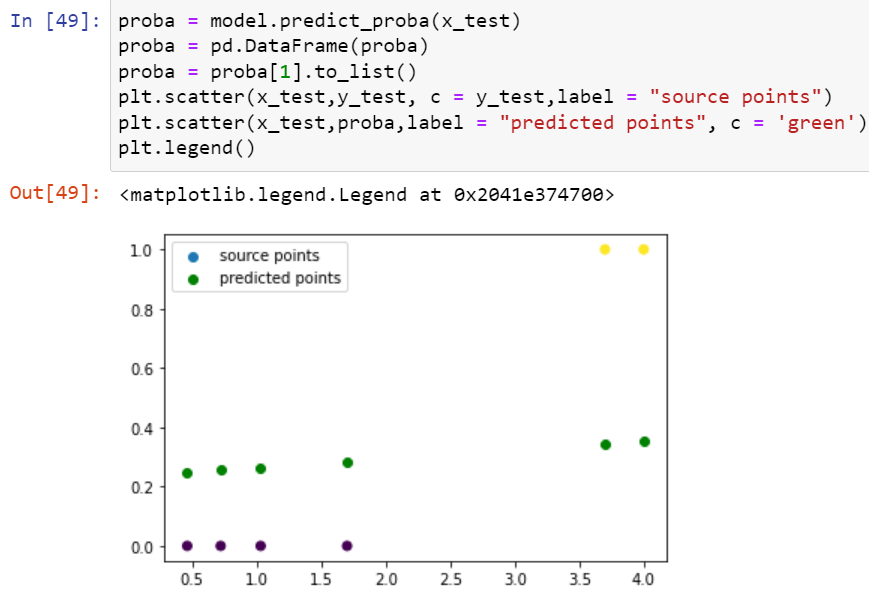


Как видим, при нулевом потреблении табака модель присвоила к 0 классу обе точки, точка в центре практически равновероятно, но все таки была ошибочно отнесена к 0 классу и 2 точки были правильно отнесены к 1 классу.

Теперь на неизвестных моделью строках, которые были ранее извлечены из модели для тестирования:

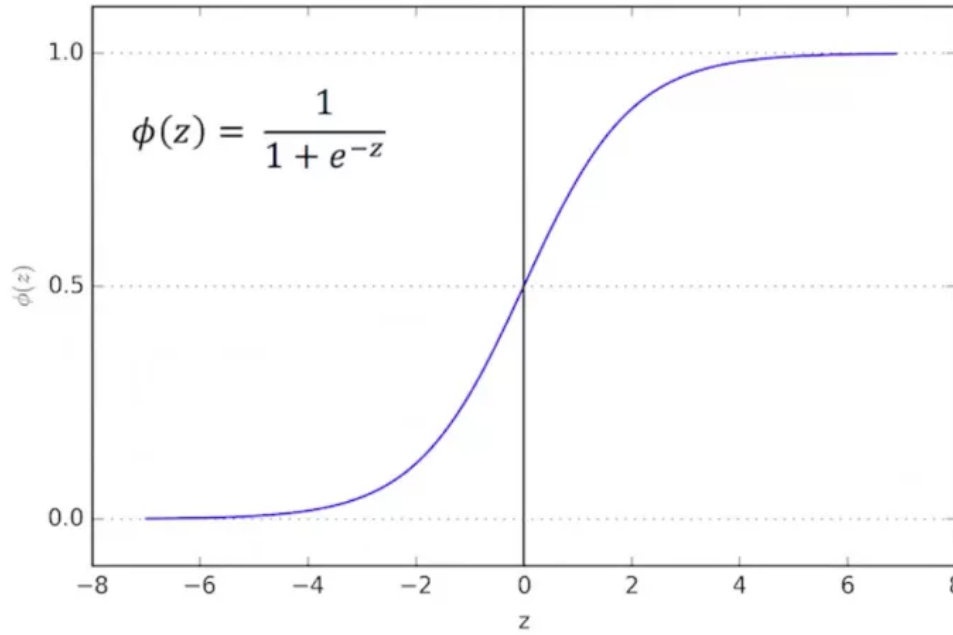


Также угадано 3 из 5 значений. Посмотрим на графике, какие вероятности были поставлены каждой точке:

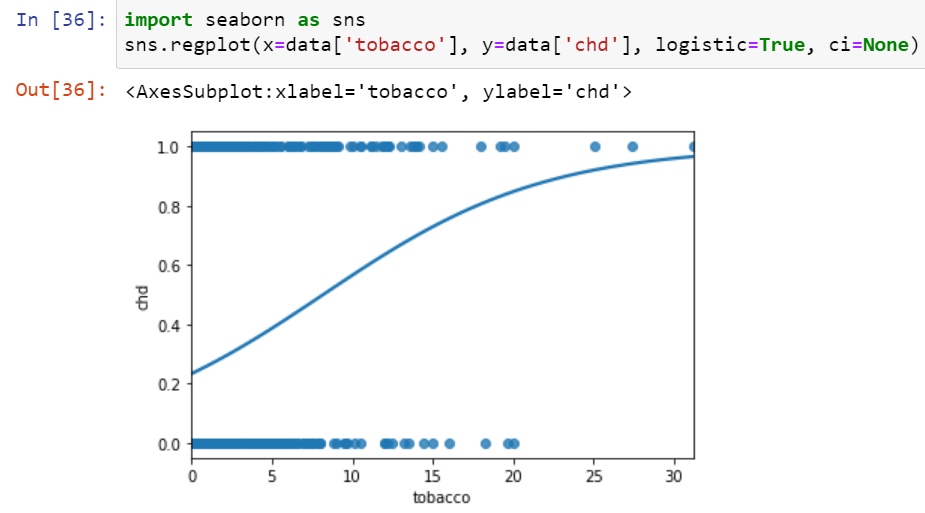


Видим, что 2 правые точки, которые должны были бы отнести к 1 классу, чуть не дотянули до 50%, поэтому также остались в 0 классе.

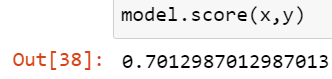
Построим логистическую кривую и облако точек. Логистическая кривая имеет вид сигмоиды и ее функция выглядит следующим образом:



Реализовать это можно с помощью библиотеки seaborn:

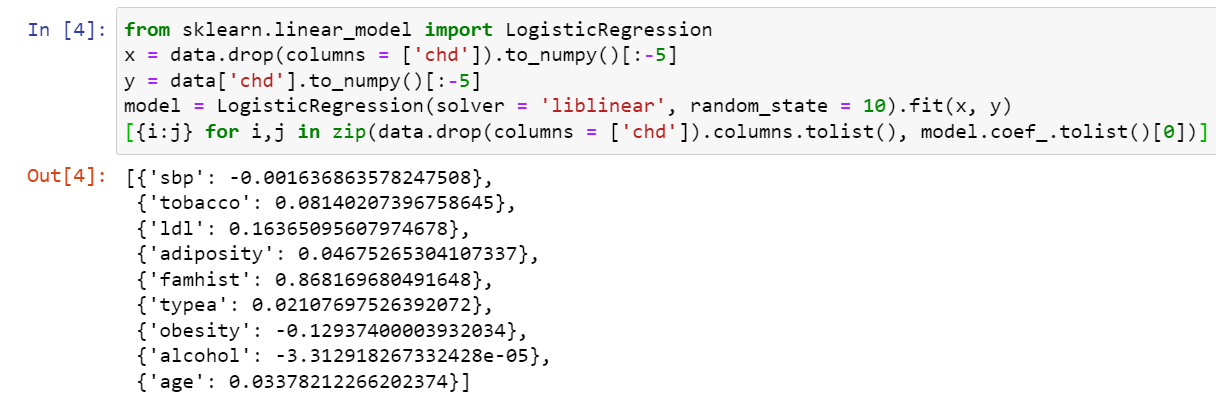


Кривая не очень хорошо ложится на исходные точки, так как идет сильный скос налево, то есть в сторону меньшего кол-ва потребляемого табака. Проверим точность построенной модели на исходных данных:



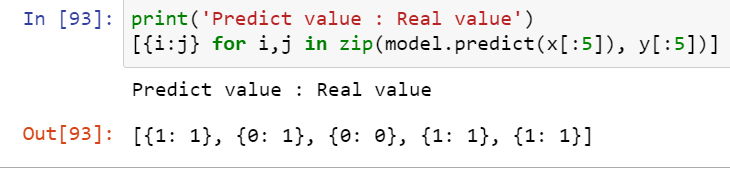
То есть 7 из 10 случаев на исходных данных выявляется правильно. Можем повысить точность с помощью учета нескольких переменных, а не только влияние табака, а для этого используем множественную логистическую регрессию.

1. Используем библиотеку sklearn, преобразуем сначала независимые и зависимую переменные к нампаевскому аррэю для подачи на вход модели импортированного модуля LogisticRegression, используем метод решения liblinear, так как он лучше всего работает всего с мелкими датасетами. Выведем результат полученных коэффициентов виде словаря для наглядности:

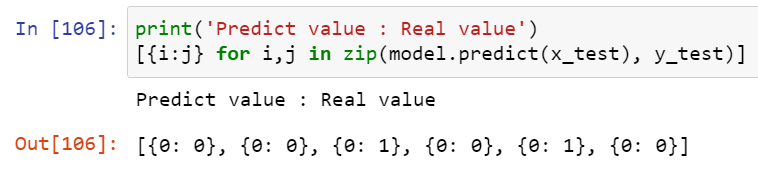


Как видим, наиболее сильные коэффициенты у параметров famhist (0.86) и ldl (0.16), то есть наличие болезни у родни и уровень холестерина сильнее всего показывают наличие ишемической болезни сердца. Можно сказать, что практически не влияют алкоголь (близко к нулю) и давление (чуть меньше нуля). А коэффициент при влияинии табака упал вдвое по сравнению с однофакторной моделью.

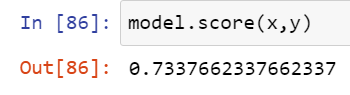
Посмотрим сначала на прогноз для строк, которые видела модель:



То есть было угадано 4 из 5 значений. Теперь на неизвестных моделью строках, которые были ранее извлечены из модели для тестирования:



На неизвестных моделью данных точность упала, теперь 3 из 5 угадано. Проверим точность построенной модели на исходных данных:



То есть модель в среднем угадывает 3 из 4 значения на данных, на которых она обучалась, что довольно сносный результат для такой небольшой выборки данных, хотя и довольно мелкий прирост по сравнению с учетом только одной переменной влияния табака (всего на 3%).

Код в питоне:

import matplotlib.pyplot as plt

import numpy as np

import pandas as pd

data = pd.read\_csv(r'C:\Users\pc\Downloads\SAheart.data', sep=',', index\_col=0)

data.head()

data.info()

data['famhist'] = data['famhist'].astype('category').cat.codes

data.head()

data.describe()

import seaborn as sns

corr\_matrix = data.corr(method = 'spearman')

sns.heatmap(corr\_matrix, cmap="YlGnBu")

data['ldl'].hist()

corr\_matrix['chd'].sort\_values(ascending = False)

x = data['tobacco'].to\_numpy().reshape(-1,1)[:-5]

y = data['chd'].to\_numpy()[:-5]

x\_test = data['tobacco'].to\_numpy().reshape(-1,1)[-5:]

y\_test = data['chd'].to\_numpy().reshape(-1,1)[-5:]

from sklearn.linear\_model import LogisticRegression

model = LogisticRegression(solver = 'liblinear', random\_state = 10).fit(x, y)

print('Coef = ', model.coef\_[0][0], '; Intercept = ', model.intercept\_[0])

print('Predict value : Real value')

[{i:j} for i,j in zip(model.predict(x[:5]), y[:5])]

print('Predict value : Real value')

[{i:j} for i,j in zip(model.predict(x\_test), y\_test)]

proba = model.predict\_proba(x[:5])

proba = pd.DataFrame(proba)

proba = proba[1].to\_list()

plt.scatter(x[:5],y[:5], c = y[:5],label = "source points")

plt.scatter(x[:5],proba,label = "predicted points", c = 'green')

plt.legend()

proba = model.predict\_proba(x\_test)

proba = pd.DataFrame(proba)

proba = proba[1].to\_list()

plt.scatter(x\_test[,y\_test, c = y\_test,label = "source points")

plt.scatter(x\_test,proba,label = "predicted points", c = 'green')

plt.legend()

import seaborn as sns

sns.regplot(x=data['tobacco'], y=data['chd'], logistic=True, ci=None)

model.score(x,y)

x = data.drop(columns = ['chd']).to\_numpy()[:-5]

y = data['chd'].to\_numpy()[:-5]

model = LogisticRegression(solver = 'liblinear', random\_state = 10).fit(x, y)

[{i:j} for i,j in zip(data.drop(columns = ['chd']).columns.tolist(), model.coef\_.tolist()[0])]

print('Predict value : Real value')

[{i:j} for i,j in zip(model.predict(x[:5]), y[:5])]

print('Predict value : Real value')

[{i:j} for i,j in zip(model.predict(x\_test), y\_test)]

model.score(x,y)