

# 北京師範大學



## 多元线性回归

部 院 系: 统计学院  
专 业: 经济与金融统计  
学 号: 202322011053  
学 生 姓 名: 欧宇翔  
指 导 教 师: 金蛟

2023 年 11 月

# 多元线性回归

## 摘 要

多元线性回归是研究一个变量与另一些变量间线性关系的统计分析方法。当误差项满足古典假定时，此时多元线性回归具有优良的统计性质，模型具有可解释性。但在实际情况中，误差项是否满足古典假定还需要进一步验证，否则模型的预测泛化能力无法评估甚至失效。本文主要利用 python 中 Sklearn 自带的糖尿病数据集进行回归分析。同时通过对误差项进一步探索，判断误差是否服从正态分布，是否存在异方差等其他问题，并提出相应的解决办法。在本文的最后，还提出一种新的 Xgboost 回归算法，同时与多元线性回归进行对比，分析不同算法的适用场景。

**关键词：**异方差，自相关，多重共线性，Xgboost

## **Multiple linear regression**

### **ABSTRACT**

Multiple linear regression is a statistical analysis method to study the linear relationship between one variable and some other variables. When the error term meets the classical assumption, then multiple linear regression has excellent statistical properties and the model is interpretable. However, in practice, whether the error term satisfies the classical assumption needs to be further verified, otherwise the predictive generalization ability of the model cannot be assessed or even invalidated. In this paper, we mainly utilize the diabetes dataset that comes with Sklearn in python for regression analysis. Meanwhile, by further exploring the error term, we determine whether the error obeys normal distribution, whether there are other problems such as heteroskedasticity, and propose corresponding solutions.

**KEY WORDS:** Heteroscedasticity, Autocorrelation, Multicollinearity

# 目 录

多元线性回归 .....	1
摘 要 .....	1
Multiple linear regression.....	2
<b>1 多元线性回归概述.....</b>	<b>3</b>
1.1 最小二乘法 .....	3
1.2 模型评估 .....	3
<b>2 数据预处理.....</b>	<b>5</b>
2.1 糖尿病数据集 .....	5
2.2 特征工程 .....	5
2.3 数据可视化 .....	6
<b>3 多重共线性.....</b>	<b>10</b>
3.1 多重共线性诊断 .....	10
3.1.1 相关系数图 .....	10
3.1.2 方差膨胀因子 .....	10
3.2 多重共线性处理 .....	11
3.2.1 主成分分析 .....	11
3.2.2 岭回归 .....	13
3.2.3 LASSO .....	15
3.2.4 弹性网 .....	15
3.2.5 逐步回归 .....	16
<b>4 异方差 .....</b>	<b>17</b>
4.1 正态性检验 .....	17
4.2 异方差诊断 .....	18
4.2.1 残差散点图 .....	18
4.2.2 Breusch-Pagan 异方差检验与 White 检验 .....	19
4.2.3 Goldfeld-Quandt 异方差检验 .....	20
4.3 异方差处理 .....	20
4.3.1 加权最小二乘回归 .....	20

4.3.2 异方差稳健标准误 .....	21
4.3.3 Box-cox 变换 .....	22
<b>5 自相关 .....</b>	<b>23</b>
5.1 自相关诊断 .....	23
5.1.1 自相关函数图（ACF）和偏自相函数图（PACF） .....	23
5.1.2 Durbin-Watson 检验 .....	23
5.2 自相关处理 .....	24
5.2.1 差分处理 .....	24
5.2.2 移动平均法 .....	24
5.2.3 Box-cox 变换 .....	24
<b>6 Xgboost .....</b>	<b>25</b>
6.1 目标函数 .....	25
6.1.1 模型表达 .....	25
6.1.2 损失函数 .....	25
6.2 确定树的结构 .....	26
6.2.1 精确贪婪算法 .....	26
6.2.2 近似算法 .....	27
6.3 算法实现与对比 .....	27
<b>参考文献 .....</b>	<b>29</b>

# 1 多元线性回归概述

多元线性回归是一种常用的统计分析方法，它通过建立多个自变量和一个因变量之间的线性关系模型，来探究多个自变量对一个因变量的影响。相较于一元线性回归，多元线性回归可以考虑多个自变量对因变量的影响，具有更广泛的应用场景。

在多元线性回归中，需要选择合适的自变量，并且需要对自变量与因变量之间的关系进行建模。通常情况下，使用最小二乘法来拟合模型，即通过使预测值与实际值之间误差的平方和达到最小来确定模型参数。多元线性回归可以用于预测和解释因变量的值，同时考虑多个自变量的影响。

这个方法在许多领域都有应用，例如经济学、金融学、社会科学、医学研究等。

## 1.1 最小二乘法

最小二乘法(least squares method)是一种数学优化技术，常用于拟合函数和估计参数。在统计学和机器学习中经常会用到最小二乘法。

最小二乘法的基本思想是通过最小化观测数据的实际值与拟合值（模型预测值）之间的残差平方和来寻找最优解。对于给定的数据集，最小二乘法可以帮助找到一条曲线或者函数，使得这条曲线或函数与数据点的误差平方和最小。

在回归分析中，最小二乘法被广泛用于拟合线性回归模型，通过最小化实际观测值与线性模型预测值之间的残差平方和来确定回归系数。除了线性回归，最小二乘法也可以应用于非线性模型的拟合。

## 1.2 模型评估

在多元线性回归中，可以使用多种指标来评估模型的性能和拟合程度。以下是一些常用的模型评估指标：

1. 均方误差 (Mean Squared Error, MSE): 衡量模型预测结果与真实数值之间差异的一种常用指标。均方误差的计算方法是将每个预测值与相应的真实数值之差的平方求和，然后除以样本数量，得到均方误差。数学公式表示为：

$$MSE = \frac{\sum (y_i - \hat{y}_i)^2}{n} \quad (1.1)$$

其中， $y_i$ 表示第  $i$  个样本的真实数值， $\hat{y}_i$  表示对应的模型预测值， $n$  表示样本数量。

**MSE** 的值越小，表示模型的预测结果与真实数值之间的差异越小，因此通常用作衡量模型预测准确性的指标。在训练模型或比较不同模型表现时，可以使用均方误差来评估它们的预测性能。

需要注意的是，均方误差在某些情况下可能存在一些局限性，例如对异常值敏感。因此，在实际应用中，有时候会结合其他指标一起考虑，以全面评估模型的性能。

2.均方根误差（Root Mean Squared Error, RMSE）：是均方误差的平方根，它具有与原始数据的相同单位，更易于解释。

3.决定系数（Coefficient of Determination,  $R^2$ ）：表示模型对因变量变异性的解释程度，取值范围为 0 到 1。 $R^2$ 越接近 1，表示模型对数据的解释能力越好。

4. 调整决定系数（Adjusted R-squared）：是对决定系数（ $R^2$ ）的一种修正，考虑了自变量的数量和样本量对模型拟合的影响。调整决定系数可以更准确地评估多元线性回归模型的拟合程度，避免了仅仅因为增加自变量而导致模型拟合度提高的假象。调整决定系数的计算公式如下：

$$\text{Adjusted } R - \text{squared} = 1 - \frac{(1 - R^2) * (n - 1)}{(n - k - 1)} \quad (1.2)$$

其中， $R^2$ 是原始的决定系数， $n$  是样本量， $k$  是自变量的数量。

调整决定系数的取值范围与决定系数相同，即 0 到 1 之间。较高的调整决定系数表示模型能够更好地解释因变量的变异性，并且在考虑了自变量的数量和样本量后依然保持较高的拟合度。

需要注意的是，当增加自变量时，决定系数通常会增加，但这并不意味着模型的预测能力提高。因此，调整决定系数是一个更合适的指标来评估模型的拟合程度，特别是在比较具有不同自变量数量的模型时。

5.F 统计量（F-statistic）：用于检验线性回归模型整体的显著性。较大的 F 统计量值表示模型的拟合优于随机模型。

6.t 统计量（t-statistic）和 p 值（p-value）：用于评估每个自变量的显著性。t 统计量表示自变量对因变量的影响是否显著，p 值表示该统计量的显著性水平。

7.残差分析：通过分析模型的残差（预测值与真实值之差），可以检查模型是否满足线性回归假设的前提条件，并找出可能的异常值或离群点。

在评估多元线性回归模型时，一般会综合考虑以上指标，以全面了解模型的性能和适用性。同时，还可以使用交叉验证<sup>[1]</sup>、留一法等技术来进一步评估模型的泛化能力和稳定性。

## 2 数据预处理

### 2.1 糖尿病数据集

本文主要利用 python 中 Sklearn 包自带的糖尿病数据集进行回归分析和回归诊断。该组数据集共有 442 个样本，10 个特征，最后一列即为预测目标。其中前 10 个自变量已经经过数据中心化处理，并且通过标准差与样本量平方根乘积进行缩放。

	age	sex	bmi	bp	s1	s2	s3	s4	s5	s6	target
0	0.038076	0.050680	0.061696	0.021872	-0.044223	-0.034821	-0.043401	-0.002592	0.019907	-0.017646	151.0
1	-0.001882	-0.044642	-0.051474	-0.026328	-0.008449	-0.019163	0.074412	-0.039493	-0.068332	-0.092204	75.0
2	0.085299	0.050680	0.044451	-0.005670	-0.045599	-0.034194	-0.032356	-0.002592	0.002861	-0.025930	141.0
3	-0.089063	-0.044642	-0.011595	-0.036656	0.012191	0.024991	-0.036038	0.034309	0.022688	-0.009362	206.0
4	0.005383	-0.044642	-0.036385	0.021872	0.003935	0.015596	0.008142	-0.002592	-0.031988	-0.046641	135.0
...	...	...	...	...	...	...	...	...	...	...	...
437	0.041708	0.050680	0.019662	0.059744	-0.005697	-0.002566	-0.028674	-0.002592	0.031193	0.007207	178.0
438	-0.005515	0.050680	-0.015906	-0.067642	0.049341	0.079165	-0.028674	0.034309	-0.018114	0.044485	104.0
439	0.041708	0.050680	-0.015906	0.017293	-0.037344	-0.013840	-0.024993	-0.011080	-0.046883	0.015491	132.0
440	-0.045472	-0.044642	0.039062	0.001215	0.016318	0.015283	-0.028674	0.026560	0.044529	-0.025930	220.0
441	-0.045472	-0.044642	-0.073030	-0.081413	0.083740	0.027809	0.173816	-0.039493	-0.004222	0.003064	57.0

442 rows × 11 columns

图 2.1 标准化后的糖尿病数据集

### 2.2 特征工程

图 2.2 表明样本所有特征均不存在缺失值，且在 Sklearn 糖尿病数据集中性别标签已经从分类变量转换成数值标签。

除了本例的数值标签转换，其他常用的分类标签转换还包括 One-Hot 编码<sup>[2]</sup>等。



```

<class 'pandas.core.frame.DataFrame'>
RangeIndex: 442 entries, 0 to 441
Data columns (total 10 columns):
#   Column  Non-Null Count  Dtype
---  -
0   age     442 non-null     float64
1   sex     442 non-null     float64
2   bmi     442 non-null     float64
3   bp      442 non-null     float64
4   s1      442 non-null     float64
5   s2      442 non-null     float64
6   s3      442 non-null     float64
7   s4      442 non-null     float64
8   s5      442 non-null     float64
9   s6      442 non-null     float64
dtypes: float64(10)
memory usage: 34.7 KB
None

```

图 2.2 探查数据类型和缺失值

图 2.3 为对该组数据集进行初步的描述统计。在实际情况中通常有助于判断数据类型以及数据大小，是否存在离群值等。本例中数据已经进行标准化处理，故此处仅作展示。

	count	mean	std	min	25%	50%	75%	max
age	442.0	-2.511817e-19	0.047619	-0.107226	-0.037299	0.005383	0.038076	0.110727
sex	442.0	1.230790e-17	0.047619	-0.044642	-0.044642	-0.044642	0.050680	0.050680
bmi	442.0	-2.245564e-16	0.047619	-0.090275	-0.034229	-0.007284	0.031248	0.170555
bp	442.0	-4.797570e-17	0.047619	-0.112399	-0.036656	-0.005670	0.035644	0.132044
s1	442.0	-1.381499e-17	0.047619	-0.126781	-0.034248	-0.004321	0.028358	0.153914
s2	442.0	3.918434e-17	0.047619	-0.115613	-0.030358	-0.003819	0.029844	0.198788
s3	442.0	-5.777179e-18	0.047619	-0.102307	-0.035117	-0.006584	0.029312	0.181179
s4	442.0	-9.042540e-18	0.047619	-0.076395	-0.039493	-0.002592	0.034309	0.185234
s5	442.0	9.293722e-17	0.047619	-0.126097	-0.033246	-0.001947	0.032432	0.133597
s6	442.0	1.130318e-17	0.047619	-0.137767	-0.033179	-0.001078	0.027917	0.135612

图 2.3 自变量描述统计

## 2.3 数据可视化

在数据预处理阶段，利用图表对数据信息进行展示时，效率通常要比表格更加直观，更易于读者理解内容。下面利用 Python 中 matplotlib 和 seaborn 绘图包对数据集进行可视化。

从相关系数图中可以发现 s2 与 s1 变量之间的皮尔逊相关系数达到了 0.9，此外 s3 和 s4 的相关系数绝对值也达到了 0.7，说明上述两组变量之间极有可能存在线性相关关系，若直接将所有特征引入模型中，极有可能产生多重共线性。因此需要对自变量进行相应处理，以消除多重共线性带来的严重后果。

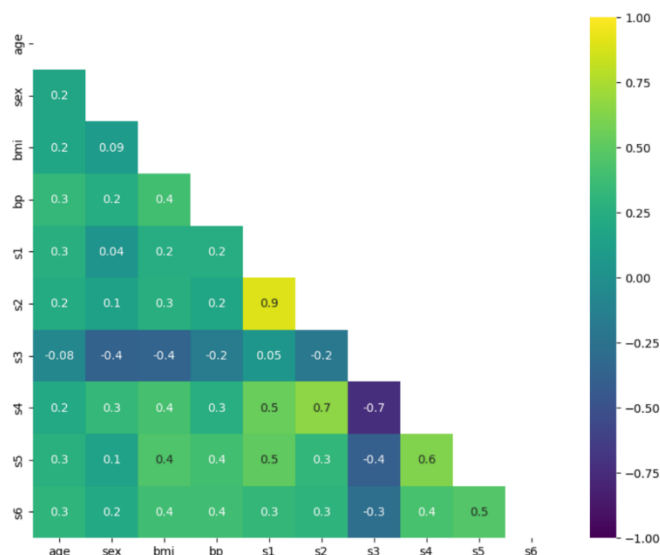


图 2.4 自变量相关系数图

不同特征的分布直方图与箱线图如下。显然部分变量存在离群点，同时通过直方图观察可知，sex 变量为分类变量，其余变量大体呈现对称分布，因此暂不考虑对数据进行相应变换。

对自变量的变换主要包括对数，幂变换等。主要目的是为了让数据集呈现对称分布，提升模型的训练效果与模型预测能力。

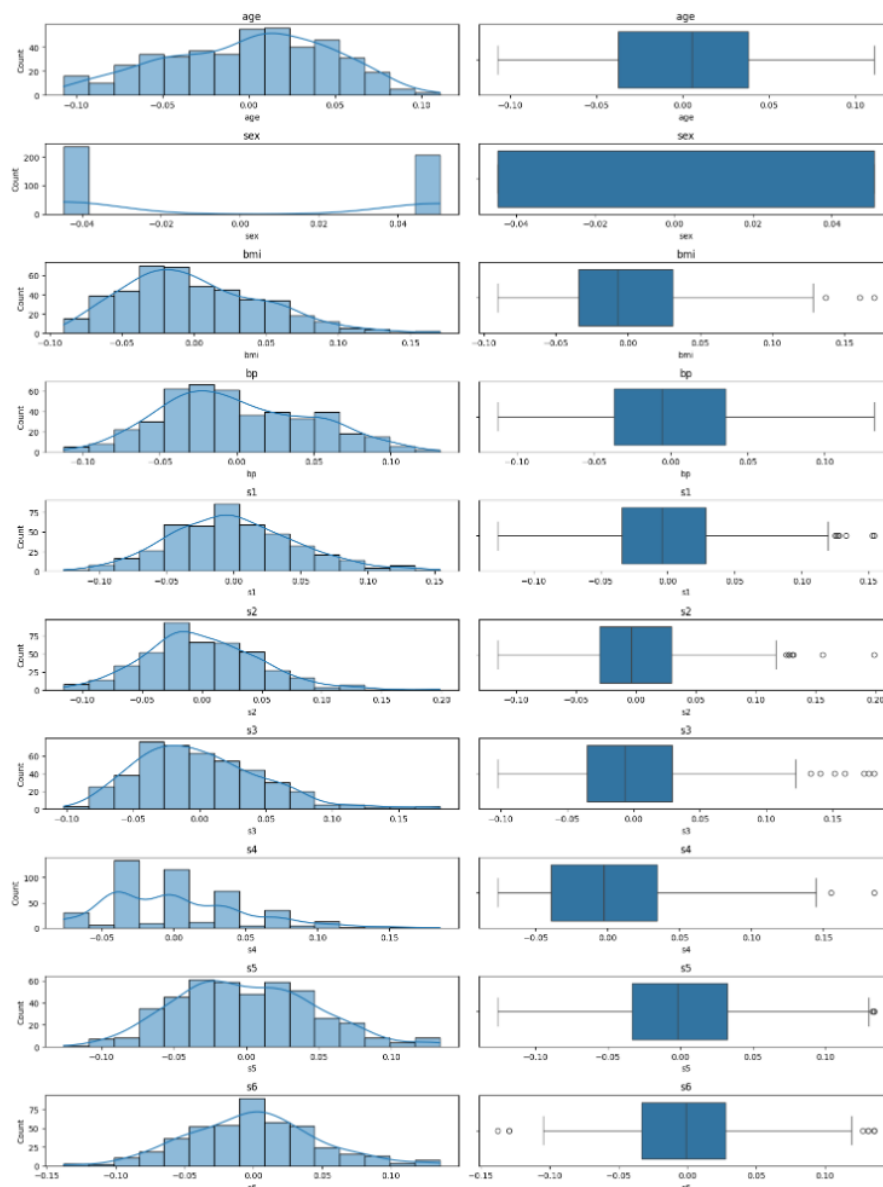


图 2.5 自变量数据分布图

图 2.6 展示了不同特征与预测变量之间的散点图，通过散点图可以直观感受不同变量与预测变量之间的相关程度。在糖尿病数据集中，bmi, s5 与目标变量呈现出较强的正相关性，s3 与目标变量则呈现出较强的负相关。而其他部分特征也与因变量呈现一定程度的相关关系，因此可以初步认为可以利用线性回归拟合该数据。

Scatter Plot of All Features versus target

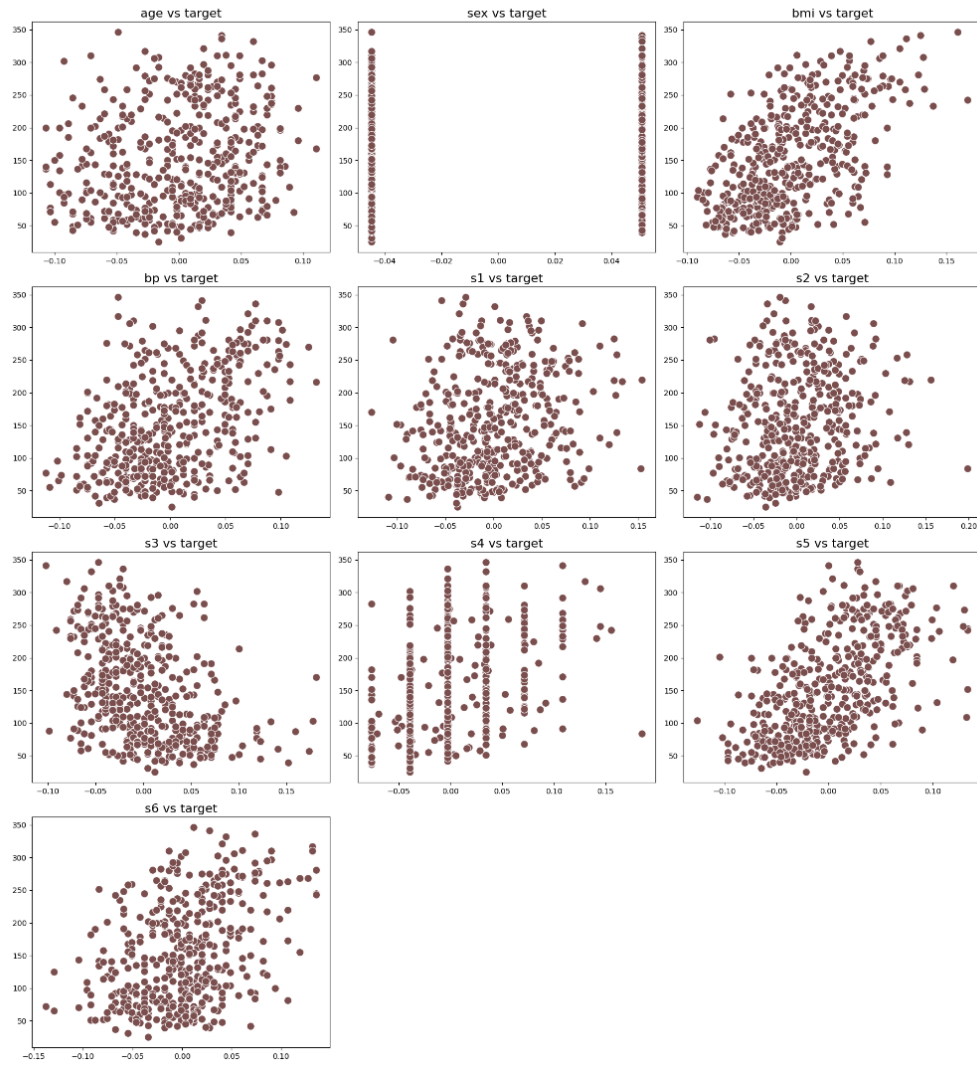


图 2.6 自变量与预测目标散点图

## 3 多重共线性

### 3.1 多重共线性诊断

#### 3.1.1 相关系数图

在数据可视化时已经提到部分变量之间存在高度相关性，因此在构建模型时考虑删除其中一个变量，或者利用其他方法。此外，该相关系数图新加入因变量与各变量之间的相关系数程度。经过验证，与可视化中初步得出的结论一致。

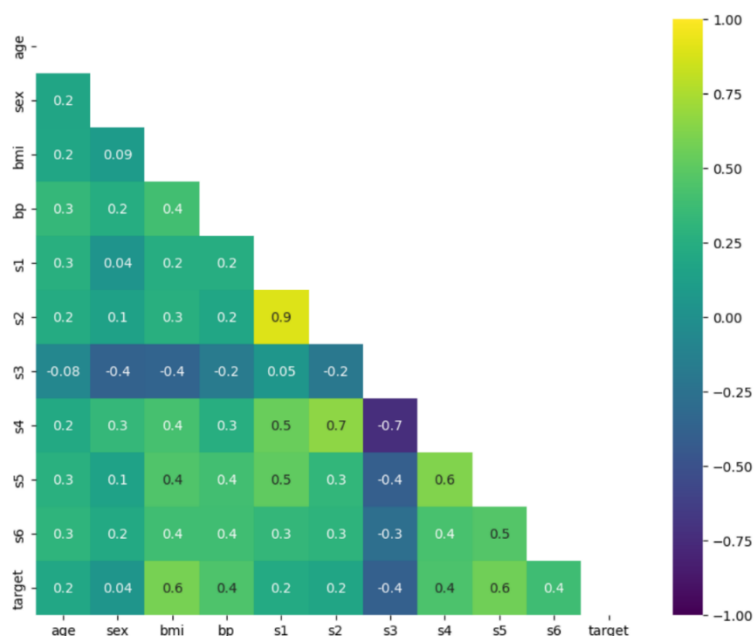


图 3.1 自变量与因变量相关系数图

#### 3.1.2 方差膨胀因子

判断是否存在多重共线性的另一个办法则是计算方差膨胀因子，方差膨胀因子（Variance Inflation Factor, VIF）是用于检测多元线性回归模型中自变量之间是否存在多重共线性的指标。它衡量了每个自变量与其他自变量之间的相关性程度。

方差膨胀因子的计算公式如下：

$$VIF = \frac{1}{(1 - R^2_i)} \quad (3.1)$$

其中,  $R^2_i$  是将第  $i$  个自变量作为因变量, 使用其余自变量进行回归得到的决定系数。

方差膨胀因子的取值范围为大于等于 1。一般来说, 如果某个自变量的方差膨胀因子超过 10, 就表示存在较高的多重共线性问题; 如果方差膨胀因子超过 5, 则也可能存在多重共线性问题。较高的方差膨胀因子表明该自变量与其他自变量高度相关, 可能会导致估计结果不稳定或解释力度受到影响。

通常认为 VIF 大于 10 则认为模型存在严重的多重共线性。观察下列结果可知 s2, s3, s4, s6 与与其他变量存在严重的多重共线性。因此不能将所有变量纳入模型进行回归, 而应对多重共线性进行相应处理

选择变量时并不能同时不考虑上述四个变量, 在删除变量时应根据实际情况进行分析。有可能仅仅在删除其中一个变量之后, 模型不存在多重共线性。

	variable	VIF
0	age	1.000000
1	sex	1.217307
2	bmi	1.278071
3	bp	1.509437
4	s1	1.459428
5	s2	59.202510
6	s3	39.193370
7	s4	15.402156
8	s5	8.890986
9	s6	10.075967

图 3.2 方差膨胀因子计算结果

## 3.2 多重共线性处理

### 3.2.1 主成分分析

主成分分析 (Principal Component Analysis, PCA)<sup>[3]</sup> 是一种常用的多元数据降维方法, 通过将原始变量转化为若干个线性无关的主成分, 实现对数据的降维和简化。其基本思想是在保留数据主要信息的同时, 尽可能减少数据的冗余和噪声。

PCA 的步骤如下:

1. 数据标准化: 对原始数据进行标准化处理, 使得各个变量具有相同的比例尺和权重。
2. 构建协方差矩阵: 计算标准化后的数据的协方差矩阵, 反映各个变量之间的相关性

程度。

3. 计算特征值和特征向量：对协方差矩阵进行特征值分解，得到特征值和对应的特征向量。

4. 选择主成分：按照特征值从大到小的顺序，选择前  $k$  个特征向量作为主成分，其中  $k$  是需要降维到的维数。

5. 计算主成分得分：将原始数据投影到所选的主成分上，得到新的主成分得分矩阵。

通过 PCA 可以实现数据的降维和简化，减少数据的冗余和噪声，同时保留数据的主要信息。

在本例中，可以发现前五个主成分累计方差比例超过 80%，故令  $k=5$ ，将原始自变量维数从 10 维降至 5 维，新的 5 个特征向量之间线性无关，从而消除多重共线性。

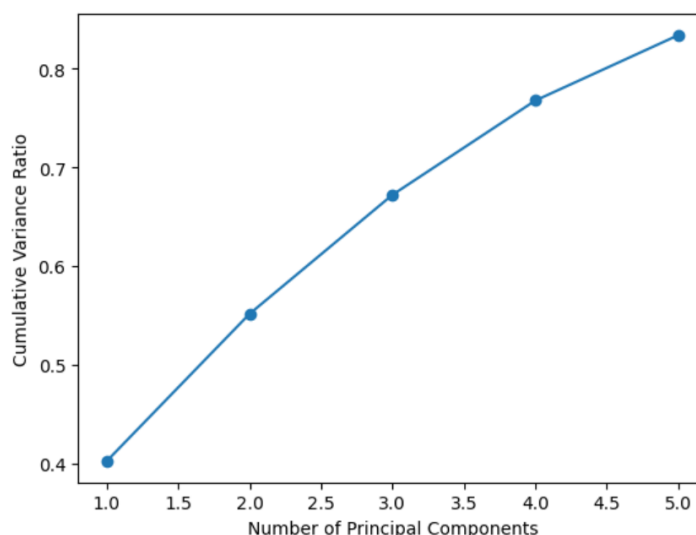


图 3.3 主成分方差累计比例

图 3.4 为利用主成分分析进行降维后，对新的变量进行回归拟合的结果，从图中可以看到第五个主成分线性关系并不显著，因此在后续进一步分析中可以考虑剔除。下列结果利用 Python 中的 statsmodels 库实现

OLS Regression Results						
Dep. Variable:	y	R-squared:	0.500			
Model:	OLS	Adj. R-squared:	0.495			
Method:	Least Squares	F-statistic:	87.33			
Date:	Wed, 22 Nov 2023	Prob (F-statistic):	1.75e-63			
Time:	13:50:55	Log-Likelihood:	-2393.8			
No. Observations:	442	AIC:	4800.			
Df Residuals:	436	BIC:	4824.			
Df Model:	5					
Covariance Type:	nonrobust					
	coef	std err	t	P> t	[0.025	0.975]
const	152.1335	2.607	58.361	0.000	147.010	157.257
x1	448.1948	27.319	16.406	0.000	394.501	501.889
x2	-256.7489	44.862	-5.723	0.000	-344.922	-168.576
x3	238.2413	49.905	4.774	0.000	140.157	336.326
x4	593.1035	56.066	10.579	0.000	482.910	703.297
x5	-16.6825	67.348	-0.248	0.804	-149.049	115.684
Omnibus:	4.306	Durbin-Watson:	1.949			
Prob(Omnibus):	0.116	Jarque-Bera (JB):	3.142			
Skew:	0.045	Prob(JB):	0.208			
Kurtosis:	2.597	Cond. No.	25.8			

图 3.4 主成分分析后回归结果

### 3.2.2 岭回归

岭回归（Ridge Regression）是一种常用的线性回归方法，它可以解决因变量与自变量之间存在多重共线性（Multicollinearity）时，最小二乘法无法求解的问题。在多重共线性的情况下，最小二乘法的系数估计会非常不稳定，甚至会出现有误差方向相反的情况，岭回归通过对系数加入 L2 正则化项来降低模型的过拟合程度，提高模型的泛化能力。

岭回归的目标函数为：

$$\min \|Xw - y\|^2 + \alpha \|w\|^2 \quad (3.2)$$

其中， $X$ 为自变量矩阵， $y$ 为目标变量， $w$ 为回归系数， $\alpha$ 为正则化系数。当 $\alpha=0$ 时，岭回归退化为最小二乘法。当 $\alpha$ 越大时，岭回归对系数的惩罚越大，模型的复杂度越低。

岭迹图（Ridge Trace Plot）是一种用于选择岭回归正则化系数的可视化方法。岭迹图可以显示在不同正则化系数下模型系数的变化情况，从而选择最合适的正则化系数。

岭迹图的横轴是正则化系数 $\alpha$ ，纵轴是模型系数的值。在岭迹图中，每个曲线表示一个特征的系数随着正则化系数的变化而变化的情况。当 $\alpha = 0$ 时，所有系数都等于最小二乘法的系数；当 $\alpha$ 越来越大时，系数逐渐趋向于 0。

对于该组糖尿病数据集，绘制的岭迹图如下：



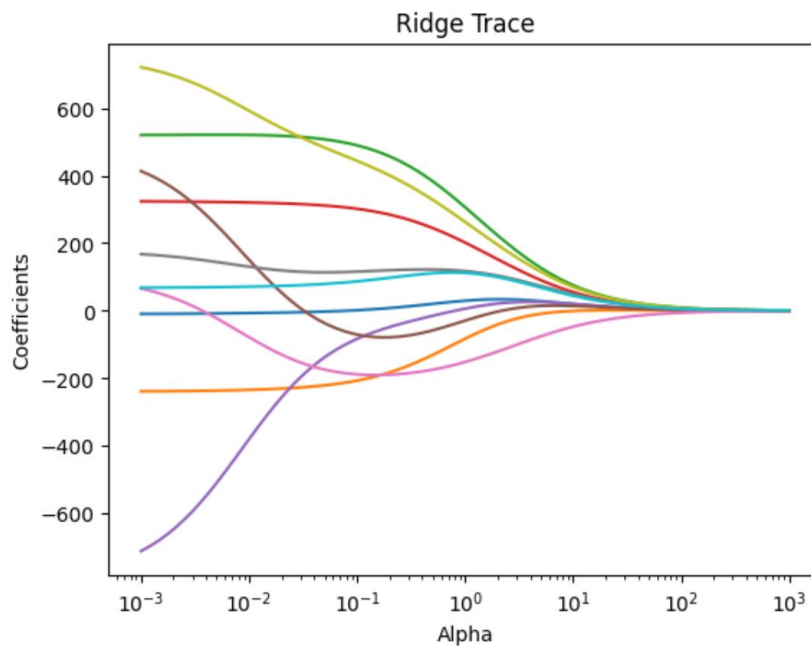


图 3.5 岭迹图

在 Python 中，还可以使用网格搜索法来搜寻最优参数，在本例中，利用网格搜索可以确定最优  $\alpha = 0.0004$ ，而后利用交叉验证计算平均 MSE，其中折数  $k=5$ ，并计算调整后的  $R^2$ ，部分代码及输出结果如下

```
ridge = Ridge(alpha=0.00041320124001153346)

# 拟合模型
ridge.fit(X, y)
y_pred = ridge.predict(X)

# 计算R方值
r_squared = r2_score(y, y_pred)

# 使用交叉验证计算MSE
mse = -cross_val_score(ridge, X, y, cv=5,
                        scoring='neg_mean_squared_error').mean()

# 使用交叉验证计算MAE
mae = -cross_val_score(ridge, X, y, cv=5,
                        scoring='neg_mean_absolute_error').mean()

# 计算调整R方
adjusted_r_squared = 1 - (1 - r_squared) * (X.shape[0] - 1) / (X.shape[0] - X.shape[1] - 1)
print("Mean Squared Error (MSE):", mse)
print("Mean Absolute Error (MAE):", mae)
print("adjusted_r_squared:", adjusted_r_squared)

Mean Squared Error (MSE): 3017.450348761696
Mean Absolute Error (MAE): 44.38738463079251
adjusted_r_squared: 0.5024211954276052
```

图 3.6 岭回归部分代码展示

### 3.2.3 LASSO

LASSO (Least Absolute Shrinkage and Selection Operator) [4]是另一种常用的线性回归方法,它可以通过加入 L1 正则化项来实现特征选择以及降低模型复杂度。与岭回归不同,LASSO 倾向于使得一些特征的系数变为 0,从而实现特征的稀疏性,即可以剔除对模型预测影响较小的特征。

LASSO 的目标函数为:

$$\min \|Xw - y\|^2 + \alpha \|w\|_1 \quad (3.3)$$

其中,  $X$  为自变量矩阵,  $y$  为目标变量,  $w$  为回归系数,  $\alpha$  为正则化系数。与岭回归类似,当  $\alpha = 0$  时, LASSO 退化为最小二乘法。而当  $\alpha$  越大时, LASSO 倾向于使得一些系数变为 0。

代码大致与上述岭回归相同,仅需要调用 Lasso 包即可,结果如下

```
# 输出最佳的alpha值
print("最佳的alpha值: ", grid_search.best_params_['alpha'])

lasso = Lasso(alpha=grid_search.best_params_['alpha'])

# 拟合模型
lasso.fit(X, y)
y_pred = lasso.predict(X)

# 计算R方值
r_squared = r2_score(y, y_pred)

# 使用交叉验证计算MSE
mse = -cross_val_score(lasso, X, y, cv=5,
                        scoring='neg_mean_squared_error').mean()

# 使用交叉验证计算MAE
mae = -cross_val_score(lasso, X, y, cv=5,
                        scoring='neg_mean_absolute_error').mean()

# 计算调整R方
adjusted_r_squared = 1 - (1 - r_squared) * (X.shape[0] - 1) / (X.shape[0] - X.shape[1] - 1)
print("Mean Squared Error (MSE):", mse)
print("Mean Absolute Error (MAE):", mae)
print("adjusted_r_squared:", adjusted_r_squared)

最佳的alpha值: 0.049770235643321115
Mean Squared Error (MSE): 3001.2987319526583
Mean Absolute Error (MAE): 44.5187028487956
adjusted_r_squared: 0.5008128982867004
```

图 3.7 LASSO 部分代码展示

### 3.2.4 弹性网

弹性网 (Elastic Net) [5]是一种结合了 L1 和 L2 正则化的线性回归方法,它可以克服 LASSO 在高度相关特征存在时的一些限制。弹性网通过引入混合项来平衡 L1 和 L2 正则化的影响,既能够实现特征选择,又能够保留高度相关特征,并降低模型复杂度。

弹性网的目标函数为:

$$\min \|Xw - y\|^2 + \alpha \rho \|w\|_1 + \alpha \frac{1-\rho}{2} \|w\|_2^2$$

(3.4)

其中,  $X$  为自变量矩阵,  $y$  为目标变量,  $w$  为回归系数,  $\alpha$  为正则化系数,  $\rho$  为弹性网系数。当  $\rho = 0$  时, 弹性网退化为岭回归; 当  $\rho = 1$  时, 弹性网退化为 LASSO。

```
# 输出最佳的参数组合
print("最佳的参数组合: ", grid_search.best_params_)

elastic = ElasticNet(alpha=grid_search.best_params_['alpha'], l1_ratio=grid_search.best_params_['l1_ratio'])

# 拟合模型
elastic.fit(X, y)
y_pred = elastic.predict(X)

# 计算R方值
r_squared = r2_score(y, y_pred)

# 计算调整R方
adjusted_r_squared = 1 - (1 - r_squared) * (X.shape[0] - 1) / (X.shape[0] - X.shape[1] - 1)

print("adjusted_r_squared:", adjusted_r_squared)
```

图 3.8 弹性网部分代码展示

在本例中利用网格搜索计算最优参数  $\alpha$  和  $\rho$ , 计算求出  $\rho = 1$ , 此时利用弹性网计算结果与 LASSO 计算结果相同。

### 3.2.5 逐步回归

逐步回归分析方法的基本思路是自动从大量可供选择的变量中选取最重要的变量, 建立回归分析的预测或者解释模型。其基本思想是: 将自变量逐个引入, 引入的条件是其偏回归平方和经检验后是显著的。同时, 每引入一个新的自变量后, 要对旧的自变量逐个检验, 剔除偏回归平方和不显著的自变量。这样一直边引入边剔除, 直到既无新变量引入也无旧变量删除为止。它的实质是建立“最优”的多元线性回归方程。

依据上述思想, 可利用逐步回归筛选并剔除引起多重共线性的变量, 其具体步骤如下: 先用被解释变量对每一个所考虑的解釋变量做简单回归, 然后以对被解释变量贡献最大的解释变量所对应的回归方程为基础, 再逐步引入其余解释变量。经过逐步回归, 使得最后保留在模型中的解释变量既是重要的, 又没有严重多重共线性。

需要注意的是, 经过逐步回归后的模型并不总是最优模型, 或者模型并不总是具有较高的可解释性, 因此应谨慎考虑

在本例中, 利用逐步子集进行变量筛选, 最终结果仅剩有 **bmi**, **bp**, **s5** 三个变量, 而调整后的  $R^2$  降低为 46.8%, 同时交叉验证后的 MSE 也有所提高。说明使用逐步回归时, 还应慎重考虑变量筛选准则等其他因素, 并结合实际情况具体分析


## 4 异方差

### 4.1 正态性检验

在传统的线性回归分析中，通常假设误差项（噪声）服从正态分布。这个假设是基于最小二乘法的推导过程，其中假设了残差（观测值与模型预测值之间的差异）是独立同分布的正态随机变量。

正态分布假设的重要性在于它使得最小二乘估计具有一些优良的性质，例如估计量的无偏性、最小方差性等。此外，正态分布假设还为假设检验和置信区间的构建提供了理论基础。

在糖尿病数据集中，选择删除 s1 变量来消除多重共线性，经过验证，剩余变量的方差膨胀因子均小于 10，因此认为模型不存在多重共线性。



	variable	VIF
0	age	1.000000
1	sex	1.216892
2	bmi	1.275049
3	bp	1.502320
4	s2	1.457413
5	s3	2.926535
6	s4	3.736890
7	s5	7.818670
8	s6	2.172865

图 4.1 删除变量后的方差膨胀因子计算结果

利用剩余变量对因变量进行回归，并进行残差正态性分析，从残差直方图可以看到误差大致服从正态分布，从 Q-Q 图中可以看到分位数点落在 45 度倾斜直线上，因此可以初步判断误差项服从正态分布。

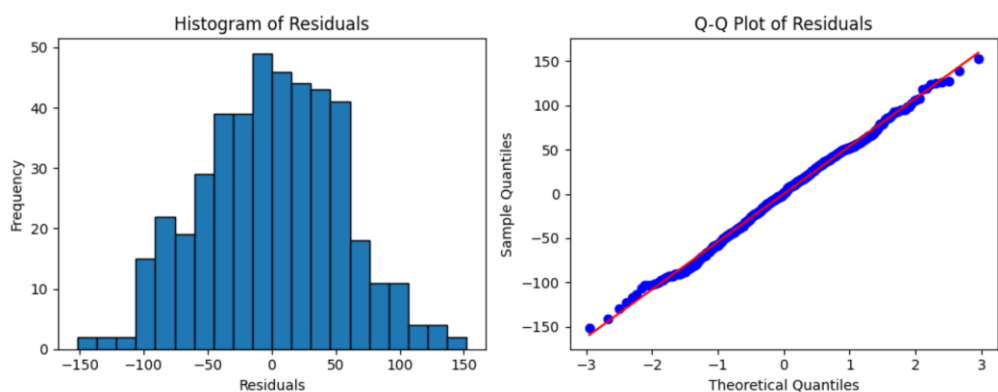


图 4.2 残差项的直方图和 Q-Q 图

夏皮罗-威尔克 (Shapiro-Wilk) 检验<sup>[6]</sup>和 D'Agostino and Pearson 检验都是用于检验数据是否符合正态分布的统计检验方法。前者基于样本数据计算出的统计量与正态分布的理论值进行比较,从而评估数据是否具有正态分布特征。后者基于样本数据的偏度 (skewness) 和峰度 (kurtosis) 两个统计量来评估数据的正态性。

两者的原假设 ( $H_0$ ) 是数据来自一个具有正态分布的总体。如果 p-value (显著性水平) 小于设定的显著性水平 (通常为 0.05), 则拒绝原假设, 认为数据不服从正态分布。如果 p-value 大于显著性水平, 则接受原假设, 认为数据可能服从正态分布。

对该残差项进行上述两种检验, 结果均表明数据可能服从正态分布, 因此可以认为误差项服从正态分布

---

```
Shapiro-Wilk Test:
Test Statistic: 0.9964734315872192
p-value: 0.4430351257324219
数据符合正态分布

D'Agostino and Pearson Test:
Test Statistic: 1.85292532719826
p-value: 0.39595185096231794
数据符合正态分布
```

图 4.3 正态性检验结果

## 4.2 异方差诊断

### 4.2.1 残差散点图

观察误差项是否存在异方差, 可以利用残差散点图进行直观分析, 若数据不存在异方

差，则标准化后的残差应均匀落在-3-3 之间，对该数据集进行残差分析，观察残差与自变量之间不存在显著的线性关系，初步认为模型不存在异方差。

Scatter Plot of All Features versus  $res^2$

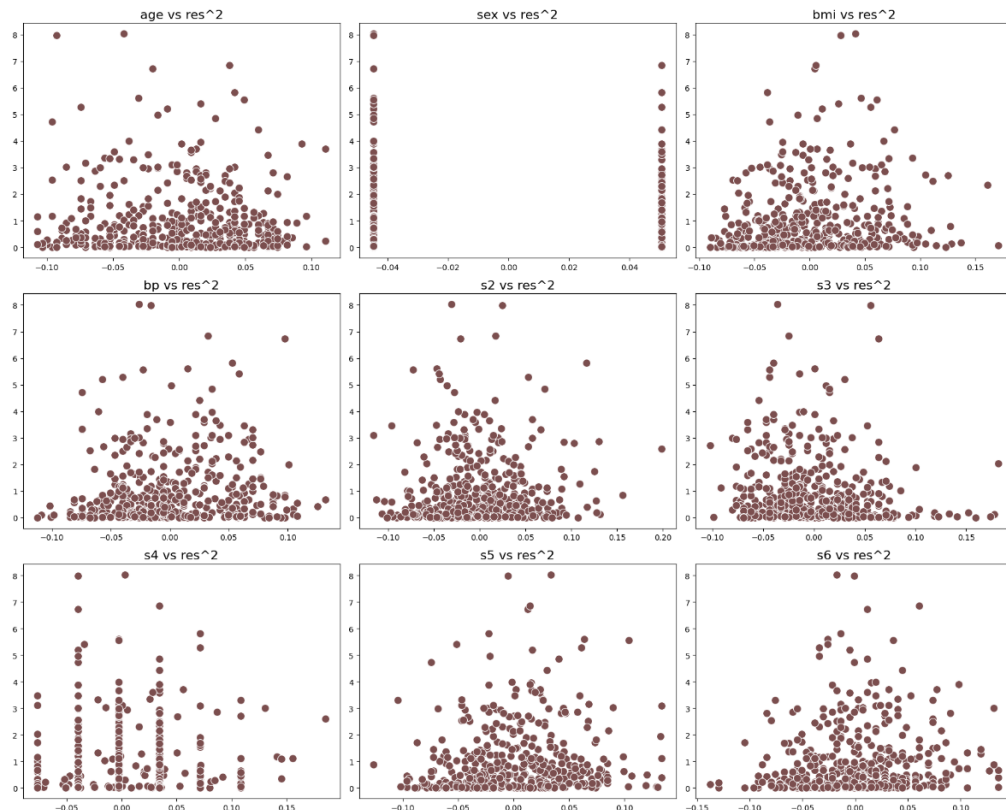


图 4.4 残差项与各变量的散点图

### 4.2.2 Breusch-Pagan 异方差检验与 White 检验

Breusch-Pagan 异方差检验<sup>[7]</sup>的核心思想是通过检验辅助回归模型的回归系数是否显著来判断数据是否存在异方差。如果解释变量对残差的平方有显著影响，即辅助回归模型的回归系数显著非零，那么可以认为数据存在异方差。否则，可以认为数据不存在异方差。

White 检验与 Breusch-Pagan 检验类似，都是基于回归分析中的残差进行的。

White 检验的核心思想是通过构建一个辅助回归模型，将解释变量和解释变量的平方作为因变量进行回归分析。

检验的原假设 ( $H_0$ ) 是：辅助回归模型的所有回归系数都等于零，即解释变量和解释变量的平方对残差的平方没有显著影响。如果 p-value 小于设定的显著性水平(通常为 0.05)，则拒绝原假设，认为数据存在异方差。

### 4.2.3 Goldfeld-Quandt 异方差检验

Goldfeld-Quandt 异方差检验（Goldfeld-Quandt test）是一种常用的异方差检验方法，用于检验线性回归模型中的残差是否存在异方差问题，并提供了一种对异方差问题进行处理的方法。

Goldfeld-Quandt 异方差检验的基本原理是通过将数据集按照某个解释变量的值排序，并将数据分为两个子样本，然后比较这两个子样本的残差方差是否存在显著差异。

Goldfeld-Quandt 异方差检验的优点是简单易用，适用于小样本情况。然而，它也有一些限制，例如对数据分布的假设较强，只能检验特定解释变量导致的异方差，不能检验全局异方差。在实际应用中，需要根据具体情况和研究目的选择合适的异方差检验方法。

对上述数据集进行三种异方差检验，结果显示利用 GQ 检验时，模型通过异方差检验，但利用 BP 检验和 White 检验时，模型出现异方差，说明残差与解释变量存在某种线性关系，与通过残差散点图得到的初步结论相反。

表 4.1 异方差检验结果

	Statistic	P-Value	F-Value	P-Value
GQ 检验			0.96737	0.5950
BP 检验	25.44222	0.002519	2.93171	0.002188
White 检验	81.36513	0.007373	1.65168	0.004293

## 4.3 异方差处理

### 4.3.1 加权最小二乘回归

加权最小二乘回归（Weighted Least Squares Regression, WLS）是一种用于处理异方差（heteroscedasticity）的回归分析方法。在普通最小二乘回归（OLS）中，假设残差的方差是恒定的，但在实际情况中，残差的方差可能是不同的，这时就需要使用加权最小二乘回归来更好地处理数据。

在加权最小二乘回归中，通过为每个样本赋予一个权重来对残差进行加权，其中权重通常被选择为方差的倒数或者方差的函数。通过对残差进行加权，可以更好地应对异方差的情况，提高了估计系数的效率和准确性。

在本数据集中，使用残差的平方的倒数作为权重，并进行异方差检验，结果显著性水平平均大于 0.05，说明利用加权最小二乘回归后，模型的异方差得以消除。



表 4.2 WLS 后异方差检验结果

	Statistic	P-Value	F-Value	P-Value
GQ 检验			0.96737	0.5950
BP 检验	11.53244	0.10653	0.59765	0.095412
White 检验	57.25415	0.08536	0.87451	0.075125

### 4.3.2 异方差稳健标准误

异方差稳健标准误可以通过调整标准误来更准确地估计回归系数的置信区间和显著性。异方差稳健标准误的计算方法是基于广义最小二乘法（GLS）或加权最小二乘法（WLS）。它通过使用异方差的一致估计量来计算标准误，从而适应了异方差的情况。

异方差稳健标准误考虑了残差的异方差性质，能够更准确地估计回归系数的置信区间和显著性。通过使用异方差稳健标准误，可以避免在存在异方差时产生的无效或误导性的统计推断。这对于确保回归结果的可靠性和准确性非常重要。

需要注意的是，异方差稳健标准误可以在大样本和小样本情况下使用，但在样本较小的情况下，可能会导致标准误估计偏向过高。另外，异方差稳健标准误只解决了标准误的问题，对于回归系数估计的一致性并没有影响。因此，在使用异方差稳健标准误时，仍需谨慎解释和推断回归结果。

本例中使用 HC3 异方差稳健标准误进行修正，观察得知加入异方差稳健标准误后，仅改变了回归系数的标准误。

OLS Regression Results

Dep. Variable:	target	R-squared:	0.514			
Model:	OLS	Adj. R-squared:	0.504			
Method:	Least Squares	F-statistic:	50.71			
Date:	Wed, 22 Nov 2023	Prob (F-statistic):	3.06e-62			
Time:	16:06:43	Log-Likelihood:	-2387.8			
No. Observations:	442	AIC:	4796.			
Df Residuals:	432	BIC:	4837.			
Df Model:	9					
Covariance Type:	nonrobust					
	coef	std err	t	P> t	[0.025	0.975]
const	152.1335	2.584	58.883	0.000	147.055	157.212
age	-7.9150	59.920	-0.132	0.895	-125.685	109.855
sex	-234.1555	61.335	-3.818	0.000	-354.707	-113.604
bmi	528.5315	66.577	7.939	0.000	397.676	659.387
bp	319.7634	65.574	4.876	0.000	190.879	448.648
s2	-143.2818	92.922	-1.542	0.124	-325.918	39.354
s3	-250.5971	105.002	-2.387	0.017	-456.976	-44.219
s4	70.4496	151.883	0.464	0.643	-228.072	368.971
s5	461.8393	80.068	5.768	0.000	304.468	619.211
s6	69.1270	66.179	1.045	0.297	-60.946	199.200

OLS Regression Results

Dep. Variable:	target	R-squared:	0.514			
Model:	OLS	Adj. R-squared:	0.504			
Method:	Least Squares	F-statistic:	70.36			
Date:	Wed, 22 Nov 2023	Prob (F-statistic):	4.68e-79			
Time:	16:09:03	Log-Likelihood:	-2387.8			
No. Observations:	442	AIC:	4796.			
Df Residuals:	432	BIC:	4837.			
Df Model:	9					
Covariance Type:	HC3					
	coef	std err	z	P> z	[0.025	0.975]
const	152.1335	2.612	58.239	0.000	147.014	157.253
age	-7.9150	58.297	-0.136	0.892	-122.175	106.345
sex	-234.1555	60.064	-3.898	0.000	-351.878	-116.433
bmi	528.5315	69.902	7.561	0.000	391.526	665.537
bp	319.7634	66.314	4.822	0.000	189.791	449.736
s2	-143.2818	92.746	-1.545	0.122	-325.061	38.498
s3	-250.5971	92.490	-2.709	0.007	-431.874	-69.320
s4	70.4496	152.447	0.462	0.644	-228.341	369.240
s5	461.8393	88.396	5.225	0.000	288.585	635.093
s6	69.1270	64.228	1.076	0.282	-56.757	195.011

图 4.5 使用异方差稳健标准误前后对比



### 4.3.3 Box-cox 变换

Box-Cox 变换通过对数据应用一个参数化的幂函数来实现。给定一个非负数  $\lambda$ , Box-Cox 变换可以定义为以下公式:

$$y(\lambda) = \begin{cases} \frac{x^\lambda - 1}{\lambda} & \text{if } \lambda \neq 0, \\ \log(x) & \text{if } \lambda = 0 \end{cases} \quad (4.1)$$

其中,  $x$  表示原始数据,  $y(\lambda)$  表示经过 Box-Cox 变换后的数据。

Box-Cox 变换的目标是使得经过变换后的数据尽可能接近正态分布并满足方差齐性的假设。通过选择最佳的  $\lambda$  值, 可以达到这个目标。通常情况下, 可以使用最大似然估计或其他统计方法来确定最佳的  $\lambda$  值。

Box-Cox 变换在实际应用中广泛用于统计建模、回归分析、时间序列分析等领域, 特别是当数据不满足正态分布和方差齐性的要求时, 可以通过 Box-Cox 变换来改善数据的性质。

本例中利用 `scipy` 库下的 `stat.boxcox` 自动筛选出最优  $\lambda$  值, 然后对因变量进行变换, 随后拟合多元线性回归模型, 并对异方差结果进行检验, 结果表明变换后的模型通过异方差检验, 证明 Box-Cox 具有可行性。

表 4.3 Box-Cox 变换后异方差检验结果

	Statistic	P-Value	F-Value	P-Value
GQ 检验			0.955300	0.629951
BP 检验	13.9519	0.12404	1.56452362	0.12347
White 检验	64.1580	0.140156	1.243073	0.129145

## 5 自相关

### 5.1 自相关诊断

#### 5.1.1 自相关函数图（ACF）和偏自相函数图（PACF）

自相关函数（ACF）是用来衡量时间序列数据中各个时刻数据之间相关性的一种统计方法。在多元线性回归中，也可以使用 ACF 来检验残差项之间是否存在自相关性。

偏自相关函数（PACF）是一种用于衡量时间序列数据中各个时刻数据之间相关性的统计方法，它展示了一个时间序列数据在剔除了其他阶数影响后，特定滞后阶数上的纯自相关关系。

引入经过删除变量而未经过异方差处理的数据集，拟合回归模型后绘制 ACF 和 PACF，观察结果可知模型可能并不存在自相关性，主要原因有可能是该组数据不是时间序列数据。

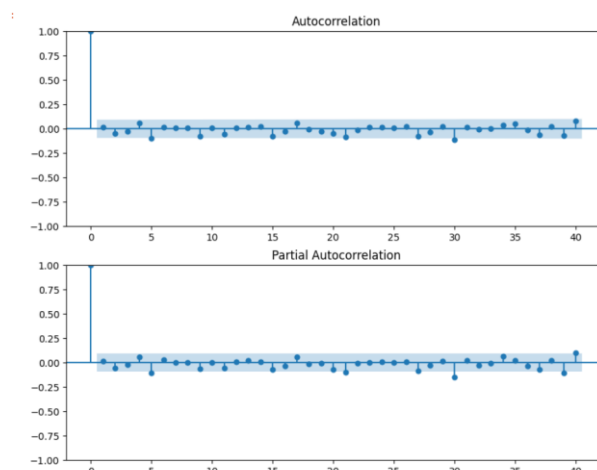


图 5.1 ACF 和 PACF

#### 5.1.2 Durbin-Watson 检验

Durbin-Watson 检验是一种用于检验线性回归模型残差序列中是否存在一阶自相关性的统计方法。它基于残差序列的自相关系数来判断是否存在自相关关系。

Durbin-Watson 检验的统计量取值范围为 0 到 4 之间，其值可以提供有关自相关性的信息：

如果统计量接近于 0，则表示存在正自相关性（正相关）。

如果统计量接近于 2，则表示不存在自相关性（无相关）。

如果统计量接近于 4，则表示存在负自相关性（负相关）。

一般来说，当 Durbin-Watson 统计量的值接近于 0 或 4 时，可以认为存在自相关性；当统计量的值接近于 2 时，可以认为不存在自相关性。通常情况下，如果统计量小于 1.5 或大于 2.5，则可以认为存在自相关性。

利用 5.1.1 的残差作 Durbin-Watson 检验，计算结果为 1.970，故可认为残差项不存在自相关性。

## 5.2 自相关处理

### 5.2.1 差分处理

差分处理是一种常见的处理自相关的方法。它可以通过计算相邻两个观测值之间的差异来消除自相关。一阶差分是指对原始序列进行一次差分操作，即  $y_t - y_{t-1}$ 。二阶差分是指对原始序列进行两次差分操作，即  $(y_t - y_{t-1}) - (y_{t-1} - y_{t-2})$ 。根据实际情况选择不同级别的差分可以消除自相关。

### 5.2.2 移动平均法

移动平均法是一种常见的平滑方法，也可以用于处理自相关。它通过计算一定窗口内的平均值来减少噪声和波动，从而消除自相关。常见的移动平均方法包括简单移动平均、加权移动平均等。

### 5.2.3 Box-cox 变换

Box-Cox 变换将数据转换后，不仅可以消除数据的异方差性，还能够消除数据的自相关性，相关公式见 4.3.3 章。使用 Box-Cox 方法需要注意变换后数据的可解释性。

## 6 Xgboost

尽管多元线性回归的理论较为完备，可解释性程度也较高。但在现实生活中，所要研究的自变量与因变量之间并非总是满足线性关系。即便是经过一系列数据处理变换使得模型满足古典假定，但在很多情况下模型的预测精度并非得到有效提升，模型的预测能力也相对较弱。在本章中介绍了一种新的回归算法 Xgboost<sup>[8]</sup>。Boosting 是一种集成学习方法，旨在通过将多个弱分类器组合成一个强分类器来提高模型性能。该算法的核心思想是通过反复训练弱分类器并加权调整样本权重，来逐步提升整个模型的性能。Boosting 算法的优点在于，它可以有效地减少模型的偏差，提高模型的泛化能力和准确度。

### 6.1 目标函数

#### 6.1.1 模型表达

Xgboost 模型由多棵树组成，每棵树都是一个回归树或分类树。对于回归问题，每个叶节点对应一个实数值。在训练过程中会不断添加新的树，并将新的树的输出与前面的树相加，得到最终预测值。该过程可以表示为下式

$$\hat{y}_i = \sum_{k=1}^K f_k(x_i), f_k(x_i) \in F \quad (6.1)$$

其中， $\hat{y}_i$ 表示样本 $i$ 的预测值， $K$ 表示树的数量， $f_k(x_i)$ 表示第 $k$ 棵树的输出， $F$ 表示所有可能的树的集合。

#### 6.1.2 损失函数

Xgboost 算法中，需要优化的目标函数如下：

$$\operatorname{argmin} \sum_{i=1}^N L(y_i, \hat{y}_i^{(t)}) + \sum_{j=1}^t \Omega(f_j) \quad (6.2)$$

其中第一个项 $L(y_i, \hat{y}_i^{(t)})$ 表示损失函数，通常为均方误差，第二个项则表示正则化，此处表示模型的复杂度，正则项包括了树的复杂度和叶节点的输出值的平方和。具体表达式如下：

$$\Omega(f_j) = \gamma T + \frac{1}{2} \lambda \sum_{i=1}^T \omega_j^2 \quad (6.3)$$

其中 $T$ 为第 $j$ 棵树的叶子节点个数， $\omega_j^2$ 为叶子节点值的平方， $\gamma$ 和 $\lambda$ 为正则化超参数。

Xgboost 采用了前向分布算法，即拟合出当前决策树后再拟合下一棵决策树，因此 $\hat{y}_i^{(t)}$

可表示为：

$$\hat{y}_i^{(t)} = \hat{y}_i^{(t-1)} + \omega_{q(x_i)} \quad (6.4)$$

当拟合第  $t$  棵回归树时，前面  $t-1$  棵回归树已经确定，在目标函数中看作为常量，故此目标函数转变为求第  $t$  棵回归树的参数，因此目标函数可以改写成如下。

$$\operatorname{argmin} \left\{ \gamma T + \sum_{i=1}^T \left[ \sum_{i \in I_j} L(y_i, \hat{y}_i^{(t-1)} + \omega_j) + \omega_j^2 \right] \right\} \quad (6.5)$$

在上式中， $L$  的函数无法确定，故使用泰勒二阶展开，为了省略篇幅，此处不作细致推导，最后公式为：

$$\operatorname{argmin} \left\{ \gamma T + \sum_{i=1}^T \left[ \omega_j \sum_{i \in I_j} g_i + \frac{1}{2} \omega_j^2 (\lambda + \sum_{i \in I_j} h_i) \right] \right\} \quad (6.6)$$

上述目标函数为近似值，其中  $\sum_{i \in I_j} g_i$  可以用  $G_j$  表示，代表某个叶子节点的所有样本一阶梯度累和， $\sum_{i \in I_j} h_i$  可以用  $H_j$  表示，代表某个叶子节点的所有样本二阶梯度累和。据此可以求出每个节点的权重值，例如第一个节点的权重值  $\omega_j$  计算公式为：

$$\omega_j G_j + \frac{1}{2} \omega_j^2 (\lambda + H_j) \quad (6.7)$$

上述公式为  $\omega_j$  的一元二次函数，经过推导后最终目标函数为：

$$\operatorname{argmin} \left\{ \gamma T - \frac{1}{2} \sum_{i=1}^T \left[ \frac{G_j^2}{\lambda + H_j} \right] \right\} \quad (6.8)$$

## 6.2 确定树的结构

### 6.2.1 精确贪婪算法

由 6.1 可以计算出每棵树的目标函数值，但并未给出如何确定树的结构，若采用穷举法将所有可能性罗列出来，将会导致运算时间和计算成本极大增加。在 Xgboost 中，使用贪心算法来确定树的结构，即每次只考虑添加一个新的叶节点来拟合当前模型的负梯度。构建树的步骤大致如下：

初始化：初始时，只有一个根节点，表示整个训练集。

计算负梯度：对于每个样本，计算其关于当前模型的负梯度。负梯度表示了当前模型对样本的残差，即样本真实值与当前模型预测值之间的差异。

构建树：从根节点开始递归地进行树的构建。对于每个节点，考虑在该节点的基础上增加一个新的节点，并计算节点的输出值。为了选择最佳的叶节点位置和输出值，需要进行如下的步骤：

1.特征选择：对于每个节点，在当前节点的基础上考虑所有的特征，并计算每个特征的增益（gain）。

2.分裂点选择：选择一个合适的特征和阈值，将当前节点的样本划分为左右子节点。划分的目标是最大化增益。

3.节点输出值：在确定分裂点后，计算新的叶节点的输出值。输出值通常通过最小化损失函数来确定。

### 6.2.2 近似算法

当模型的特征个数较多时，采用精确贪心算法时会导致计算复杂度大大提高，因为此时需要把所有特征考虑在内计算增益。为了简化计算，我们可以采用近似算法来逼近最优值。近似算法中对于特征的选取一共有两种方法：

1.按树随机抽样：指的是在根节点分裂之前就随机筛选出一部分特征，而后在该树的分裂过程中仅考虑这些特征，当该回归树分裂完成后。在下一棵新的回归树又重新筛选一部分特征。

2.按层随机抽样：指的是从根节点开始划分时，每一次都从所有特征中随机筛选一部分特征，并计算被筛选的特征的增益。

为了提高计算效率，不仅可以只考虑特征变量的筛选，还可以考虑分裂点的选取。分裂点的选取也可以采用上述两种方法。在 Xgboost 中，另外一种常用的分桶方法为加权分位法。权重即为该样本在该特征上的二阶导  $h_i$ 。加权分位法是一种高效计算带权重数据集分位数的方法，可以用于选择最佳的分裂点，从而提高 Xgboost 模型的训练速度和性能。

## 6.3 算法实现与对比

为了方便进行对比，本次实验将数据按 7: 3 划分为训练集和测试集，模型在训练集上进行拟合，并且计算在训练集和测试集上的均方误差。

首先使用网格搜索确定部分最优参数组合，然后直接在原始数据上使用 Xgboost 算法。同时使用第三章各类方法来进行对比，观察结果可知，使用主成分分析降维后进行回归的训练集和测试集差距较大，一方面可能是偶然误差，另一方面可能是模型欠拟合，因此该列数据不具有参考性。显然，Xgboost 在训练集和测试集上的 MSE 均达到最小，并且模型还可以进行后续调优。

此处还验证了多重共线性对 Xgboost 的影响，在删除高度相关的变量后模型的预测能力并未得到有效提升，说明 Xgboost 模型泛化能力较强，鲁棒性较好。

在本次实验中仅通过控制 min\_child\_weight 大小来预防过拟合，它含义是每个叶子节点上至少需要包含  $n$  个样本。此外，模型还可以引入 L1, L2 正则项来控制过拟合程度，具

体最优参数组合可以通过网格搜索法，随机搜索法等进行筛选，从而对模型进行进一步调优。

对于提高模型的精度和预测能力，除了对超参数进行调优之外，还可以使用神经网络。

表 6.1 不同方法回归结果

	PCA	Ridge	LASSO	cut_X	Xgboost
Train_MSE	3056.90	2924.15	2926.27	2958.81	2727.41
Test_MSE	2773.00	2821.17	2818.16	2827.41	2778.37

## 参考文献

- [1]张剑飞;王真;崔文升;杜晓昕. 基于交叉验证和神经网络融合的医学数据分类[J]. 齐齐哈尔大学学报(自然科学版), 2019, 35(04):1-5.
- [2]刘辉玲;陶洁;邱磊. 基于 Python 的 One-hot 编码的实现[J]. 武汉船舶职业技术学院学报, 2021, 20(03):136-139.
- [3]李竞时;匡晓迪;李琼;何恩业;张聿柏;袁承仪;张延琳. 基于主成分分析和 LSTM 神经网络的海温预报模型[J]. 海洋预报, 2023, 40(02):1-10.
- [4]储嘉诚;唐炎林. 高维线性模型中的纠偏 LASSO 综述[J]. 应用概率统计, 2023, 39(03):455-474.
- [5]蒋仕旗;戴家佳. 一种改进弹性网估计及其在 Logistic 回归上的应用[J]. 应用数学学报, 2023, 46(05):721-743.
- [6]张纪泉. 总体分布的正态性检验——介绍夏皮罗-威尔克的 W 检验法[J]. 中国纤检, 1982, (05):34-40. DOI:10.14162/j.cnki.11-4772/t.1982.05.009
- [7]朱金蝶. 回归模型中异方差检验方法研究[D]. 山西大学, 2019. DOI:10.27284/d.cnki.gsxiu.2019.000670
- [8]赵世雄;韩斌;张紫妍. 基于 CNN-XGBoost 的恶意 URL 检测[J]. 软件导刊, 2023, 22(05):150-157.