# 岭回归其实就是线性回归的改进

|  |
| --- |
|  |

# 1.带有L2正则化的线性回归-岭回归

## 1>API模块

|  |
| --- |
|  |
|  |
|  |

### 岭回归很常用,比梯度下降还要好.里面还要标准化功能,用正规api,我们就不需要做标准化了

## 2>观察正则化程度的变化,对结果的影响

|  |
| --- |
|  |

## 3>boston房价预测,使用岭回归,先用默认参数

|  |
| --- |
| """   波士顿房价预测,使用岭回归来实现   sklearn1.2以上版本没有load\_boston方法,需要使用fetch\_openml  """  from sklearn.datasets import fetch\_openml  from sklearn.model\_selection import train\_test\_split  from sklearn.preprocessing import StandardScaler  from sklearn.linear\_model import Ridge  from sklearn.metrics import mean\_squared\_error  # 1.获取数据集  data\_x, data\_y = fetch\_openml(name="boston", version=1, as\_frame=True,                                return\_X\_y=True, parser="pandas")  # 2 划分数据集  x\_train,x\_test,y\_train,y\_test = train\_test\_split(data\_x,data\_y,random\_state=22)  # 3.特征工程,标准化  transfer = StandardScaler()  x\_train = transfer.fit\_transform(x\_train)  x\_test  = transfer.transform(x\_test)  print(x\_train)  # 4.预估器流程  estimator = Ridge()  estimator.fit(x\_train,y\_train)  # 5. 得出模型  print("岭回归-权重系数为：\n", estimator.coef\_)  print("岭回归-偏置为：\n", estimator.intercept\_)  # 模型评估  y\_predict = estimator.predict(x\_test)  print("岭回归-预测房价：", y\_predict)  error = mean\_squared\_error(y\_test,y\_predict) # 第一个参数是真实值,第二个参数是预测值  print("岭回归-均方误差：\n", error) |

### 效果

|  |
| --- |
|  |

### 可以调整运行参数,但是效果有限

|  |
| --- |
| """   波士顿房价预测,使用岭回归来实现   sklearn1.2以上版本没有load\_boston方法,需要使用fetch\_openml  """  from sklearn.datasets import fetch\_openml  from sklearn.model\_selection import train\_test\_split  from sklearn.preprocessing import StandardScaler  from sklearn.linear\_model import Ridge  from sklearn.metrics import mean\_squared\_error  # 1.获取数据集  data\_x, data\_y = fetch\_openml(name="boston", version=1, as\_frame=True,                                return\_X\_y=True, parser="pandas")  # 2 划分数据集  x\_train,x\_test,y\_train,y\_test = train\_test\_split(data\_x,data\_y,random\_state=22)  # 3.特征工程,标准化  transfer = StandardScaler()  x\_train = transfer.fit\_transform(x\_train)  x\_test  = transfer.transform(x\_test)  print(x\_train)  # 4.预估器流程  estimator = Ridge(alpha=0.2,max\_iter=9000)  estimator.fit(x\_train,y\_train)  # 5. 得出模型  print("岭回归-权重系数为：\n", estimator.coef\_)  print("岭回归-偏置为：\n", estimator.intercept\_)  # 模型评估  y\_predict = estimator.predict(x\_test)  print("岭回归-预测房价：", y\_predict)  error = mean\_squared\_error(y\_test,y\_predict) # 第一个参数是真实值,第二个参数是预测值  print("岭回归-均方误差：\n", error) |
|  |