**不同学习率对房价预测的影响**

**源代码：**

X=raw\_data.drop(['id','date','price','sqft\_lot15','sqft\_living15','long','lat','zipcode','yr\_built','sqft\_basement'],axis=1)

y=raw\_data['price']

X\_train,X\_test,y\_train,y\_test=train\_test\_split(X,y,test\_size=0.3,random\_state=1026)

sc.fit(X\_train)

X\_train=sc.transform(X\_train)

X\_test=sc.transform(X\_test)

# 拼接训练样本的X\_b

X\_b = np.hstack([np.ones((len(X\_train),1)),X\_train])

# 初始化theta

initial\_theta = np.random.random(size=(X\_b.shape[1]))

# 学习率为0.1

theta,cost\_value = BGD(X\_b,y\_train,initial\_theta,eta=0.1,iters=1e5)

# 拼接测试样本的X\_b

X\_b\_test = np.hstack([np.ones((len(X\_test),1)),X\_test])

# 在测试集上预测

y\_predict = X\_b\_test.dot(theta)

from sklearn.metrics import mean\_squared\_error

r2 = 1 - mean\_squared\_error(y\_test,y\_predict) / np.var(y\_test)

print("在测试集上的R^2为：",r2)

# 拼接训练样本的X\_b

X\_b = np.hstack([np.ones((len(X\_train),1)),X\_train])

# 初始化theta

initial\_theta = np.random.random(size=(X\_b.shape[1]))

# 学习率为0.01

theta,cost\_value = BGD(X\_b,y\_train,initial\_theta,eta=0.01,iters=1e5)

# 拼接测试样本的X\_b

X\_b\_test = np.hstack([np.ones((len(X\_test),1)),X\_test])

# 在测试集上预测

y\_predict = X\_b\_test.dot(theta)

from sklearn.metrics import mean\_squared\_error

r2 = 1 - mean\_squared\_error(y\_test,y\_predict) / np.var(y\_test)

print("在测试集上的R^2为：",r2)

**运行（测试）过程及结果：**



