**实验（实习）报告**

1. 实验目的

熟悉利用线性回归对样本点数据进行拟合的方法

1. 实验任务

基于随机生成的数据，进行线性回归实验，实现对数据的拟合

1. 实验步骤
2. 实验结果
3. 完整代码

import numpy as np

import pandas as pd

from sklearn.cluster import KMeans

from sklearn.preprocessing import MinMaxScaler, LabelEncoder

from sklearn.linear\_model import LinearRegression, Ridge, SGDRegressor

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

from sklearn.model\_selection import StratifiedKFold

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

from sklearn.metrics import roc\_auc\_score

from sklearn import metrics

from sklearn.model\_selection import cross\_val\_predict

import matplotlib.pyplot as plt

from sklearn.metrics import roc\_auc\_score

from sklearn.model\_selection import StratifiedKFold

x = 10\*np.random.rand(50, 1)  # 随机生成一个0-2之间的，大小为(500，1)的向量

y = 5+3\*x+np.random.randn(50, 1)

X\_train, X\_test, y\_train, y\_test = train\_test\_split(

    x, y, test\_size=0.25, random\_state=2023)

linreg = LinearRegression()

linreg.fit(X\_train, y\_train)

# 查看需要的模型系数结果

print("截距：", linreg.intercept\_)

print("各系数:", linreg.coef\_)

# 模型拟合测试集

y\_pred = linreg.predict(X\_test)

# 用scikit-learn计算均方差MSE

print("均方误差:", metrics.mean\_squared\_error(y\_test, y\_pred))

# 用scikit-learn计算均方根差RMSE

print("均方根误差:", np.sqrt(metrics.mean\_squared\_error(y\_test, y\_pred)))

# 得到了MSE或者RMSE，如果我们用其他方法得到了不同的系数，需要选择模型时，就用MSE较小时对应的参数。

# 通过交叉验证来持续优化模型,这里采用10折交叉验证,即cv=10

predicted = cross\_val\_predict(linreg, x, y, cv=10)

# 用scikit-learn计算MSE

print("交叉验证的均方根误差:", metrics.mean\_squared\_error(y, predicted))

# 用scikit-learn计算RMSE

#print("交叉验证的均方根误差:", np.sqrt(metrics.mean\_squared\_error(y, predicted)))

# 画图观察结果  真实值和预测值的变化关系

fig, ax = plt.subplots()

ax.scatter(y, predicted)

ax.plot([y.min(), y.max()], [y.min(), y.max()], 'k--', lw=4)

ax.set\_xlabel('Measured')

ax.set\_ylabel('Predicted')

plt.show()

1. 结果截图

截距： [5.73556021]

各系数: [[2.90745485]]

均方误差: 1.2643584605779272

均方根误差: 1.1244369526914024

交叉验证的均方根误差: 0.6851332425925533

