**沈思妤-10235501458-数据科学导论第六次作业**

**题目：**

请完成以下任务：

1. 数据获取：数据集已下载至bike.csv文件中，请使用pandas库读取该文件。

提示：原始数据集共10,000条数据。

2.数据预处理：id属性对构建回归预测模型没有帮助，请剔除掉该列。

3.数据预处理 1：我们暂不考虑不同城市对单车租用的影响，请筛选出上海市的所有数据，然后剔除city列。

提示：目前剩余4,998条数据。

4. 数据预处理 川：为简化数据，请将hour列中原来6点-18点统一为1：19点-次日5点统一为0。

5. 数据预处理 I：y列为单车租用数量，是我们的预测目标 （标签），请将该列提取出来，并转换为一个numpy列向量，将原先的y列剔除。

6. 数据预处理 V：请将DataFrame对象转换为Numpy数组，方便后续操作。

7.数据集划分：请按照训练集与测试集8:2的比例将原始数据集划分。

8. 数据预处理 V：请分别对训练集数据、训练集标签、测试集数据和测试集标签进行归一化。

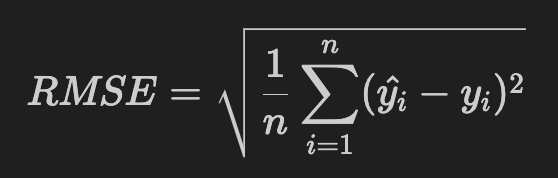
9. 模型构建：请先构建一个线性回归模型（多元一次函数），然后利用训练集训练模型。

10. 模型测试：利用测试集对训练好的模型进行评估。

」提示：使用predict(data\_array)方法输入测试集，该函数返回值为模型预测值。

11.模型评估：请使用均方根误差 (RMSE）作为评估指标，并输出RMSE值。

均方根误差(Root of Mean Squared Error)，公式为



可以直接通过对MSE求平方根获得。

**做题过程：**

1. 配置环境：

下载anaconda，在终端输入

python3 -m venv /Users/shen/Desktop/sjkxdl/venv

source /Users/shen/Desktop/sjkxdl/venv/bin/activate

pip install pandas numpy scikit-learn -i https://pypi.tuna.tsinghua.edu.cn/simple

1. 编写代码：

# 1. 数据获取

import pandas as pd

# 读取数据集

data = pd.read\_csv("bike.csv")

print(f"数据集大小: {data.shape}")

# 2. 数据预处理：移除 id 列

data = data.drop(columns=["id"])

print(f"移除id列后数据集大小: {data.shape}")

# 3. 数据预处理 1：筛选上海市数据并移除 city 列

data = data[data['city'] == 0] # 假设0表示上海市

data = data.drop(columns=["city"])

print(f"筛选上海市数据并移除city列后数据集大小: {data.shape}")

# 4. 数据预处理 川：hour 列简化

data['hour'] = data['hour'].apply(lambda x: 1 if 6 <= x <= 18 else 0)

# 5. 数据预处理 I：提取 y 列并剔除完成

y = data['y'].values

data = data.drop(columns=['y'])

# 6. 数据预处理 V：转换为 Numpy 数组

X = data.values

print("转换为Numpy数组完成")

# 7. 数据集划分：划分训练集与测试集

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

X\_train, X\_test, y\_train, y\_test = train\_test\_split(X, y, test\_size=0.2, random\_state=42)

print(f"训练集大小: {X\_train.shape}, 测试集大小: {X\_test.shape}")

# 8. 数据预处理 V：归一化

from sklearn.preprocessing import MinMaxScaler

scaler\_X = MinMaxScaler()

scaler\_y = MinMaxScaler()

X\_train = scaler\_X.fit\_transform(X\_train)

X\_test = scaler\_X.transform(X\_test)

y\_train = scaler\_y.fit\_transform(y\_train.reshape(-1, 1)).flatten()

y\_test = scaler\_y.transform(y\_test.reshape(-1, 1)).flatten()

# 9. 模型构建：线性回归模型

from sklearn.linear\_model import LinearRegression

model = LinearRegression()

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

# 10. 模型测试：利用测试集预测

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

# 11. 模型评估：计算 RMSE

from sklearn.metrics import mean\_squared\_error

import numpy as np

mse = mean\_squared\_error(y\_test, y\_pred)

rmse = np.sqrt(mse)

print(f"均方根误差 (RMSE): {rmse}")

1. 运行代码：

获得结果：

数据集大小: (10000, 9)

移除id列后数据集大小: (10000, 8)

筛选上海市数据并移除city列后数据集大小: (5002, 7)

转换为Numpy数组完成

训练集大小: (4001, 6), 测试集大小: (1001, 6)

均方根误差 (RMSE): 0.16278694923220455