**大连理工大学《Python与数据分析》课程大作业**

课程名称： Python与数据分析 考试形式： 大作业

授课院 (系)： 软件学院 上交日期： 年 月 日

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 分值 | 作业一 | 作业二 | 作业三 | 作业四 | 总分 |
| 标准分 | 25 | 25 | 25 | 25 | 100 |
| 得 分 |  |  |  |  |  |

|  |  |  |
| --- | --- | --- |
| 姓名 | 学号 | 班级 |
|  |  |  |
| 黄光辉 | 20232241318 | 软2307 |
|  |  |  |
|  |  |  |

一、作业一（共计25分）

得分

得分

**题目：**以胡润百富（https://www.hurun.net/）为数据源，完成以下任务。（1）爬取2024年胡润百富榜的榜单数据，包括财富值、个人信息以及企业信息。（2）对不同行业的富豪数量以及财富值进行统计，分析各行业的发展态势。（3）对爬取的数据，进行其他维度的分析，例如富豪在年龄、性别、出生地等方面的分布情况，利用柱状图、饼图、热力图等进行可视化展示。

**数据源**：胡润百富（https://www.hurun.net/）

二、作业二（共计25分）

得分

得分

**题目：**大连市2022-2024年天气数据分析与预测，任务描述：（1）爬取大连市近三年每天的天气状况（包含白天，夜晚） ，最高温度，最低温度，风力情况（包含白天，夜晚）（2）使用可视化工具（如Matplotlib、Seaborn、Echarts等）绘制近三年月平均气温变化图 ：计算每个月的平均最高温度和平均最低温度，绘制折线图展示一年中气温的变化趋势。（每个月取三年的平均值以下两题同理）（3）绘制近三年风力情况分布图 ：统计每个月不同风力等级（如 3-4 级、5-6 级等）出现的天数，绘制柱状图或饼图进行展示。（4）绘制近三年天气状况分布图 ：统计每个月中不同天气状况（如晴天、多云、阴天、雨天等）出现的天数，并绘制柱状图进行展示。（5）利用前三年的每月平均最高温度训练温度预测模型，给定月份生成对应月份的平均最高温度，并额外爬取2025年1-6月份数据，用折线图画出预测结果和真实结果曲线

**数据源：**天气后报https://www.tianqihoubao.com/ 。

1. **数据的爬取与处理**
   1. **所需要的库**

import requests

from bs4 import BeautifulSoup

import pandas as pd

import numpy as np

import re

import os

* 1. **爬取数据的代码**

url = f"https://www.tianqihoubao.com/lishi/dalian/month/{year\_month}.html"

    headers = {

        'User-Agent': 'Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/124.0.0.0 Safari/537.36',

        'Accept-Language': 'zh-CN,zh;q=0.9',

        'Referer': 'https://www.tianqihoubao.com/'

    }

    try:

        print(f"正在爬取 {year\_month}...")

        response = requests.get(url, headers=headers, timeout=10)

        if response.status\_code != 200:

            print(f"请求失败，状态码: {response.status\_code}")

            return None

        soup = BeautifulSoup(response.text, 'html.parser')

        table = soup.find('table', class\_='weather-table') or soup.find('table', class\_='b')

        if not table:

            print(f"未找到表格数据: {year\_month}")

            return None

* 1. **数据处理：**

# 提取日期

date\_link = cols[0].find('a')

date\_text = date\_link.get\_text(strip=True) if date\_link else cols[0].get\_text(strip=True)

# 提取天气信息

weather\_text = cols[1].get\_text(' ', strip=True).replace('\n', '')

weather\_parts = [w.strip() for w in weather\_text.split('/')]

day\_weather = weather\_parts[0] if weather\_parts else ''

night\_weather = weather\_parts[1] if len(weather\_parts) > 1 else ''

# 提取温度

temp\_text = cols[2].get\_text(' ', strip=True)

temp\_numbers = re.findall(r'-?\d+', temp\_text)

high\_temp = int(temp\_numbers[0]) if temp\_numbers and len(temp\_numbers) >= 1 else np.nan

low\_temp = int(temp\_numbers[1]) if temp\_numbers and len(temp\_numbers) >= 2 else np.nan

# 提取风力信息

wind\_text = cols[3].get\_text(' ', strip=True).replace('\n', '')

wind\_parts = [w.strip() for w in wind\_text.split('/')]

day\_wind = wind\_parts[0] if wind\_parts else ''

night\_wind = wind\_parts[1] if len(wind\_parts) > 1 else ''

* 1. **数据存储**

**如下图：**



1. **数据可视化**
   1. **所需的库：**

import pandas as pd

import numpy as np

import matplotlib.pyplot as plt

* 1. **绘制温度变化趋势图**
     1. **代码：**

plt.figure(figsize=(12, 6))

    plt.title('大连市2022-2024年温度变化趋势')

    plt.xlabel('月份')

    plt.ylabel('温度 (℃)')

    # 准备x轴数据（1-12月）

    months = range(1, 13)

    # 为每一年绘制折线

    # 计算每月的平均最高温度

    monthly\_high\_avg = data.groupby('月份')['最高温'].mean()

    monthly\_low\_avg = data.groupby('月份')['最低温'].mean()

    # 绘制折线图

    plt.plot(months, monthly\_high\_avg, label=f'最高气温', marker='o')

    plt.plot(months, monthly\_low\_avg, linestyle='--',label=f'最低气温', marker='x')

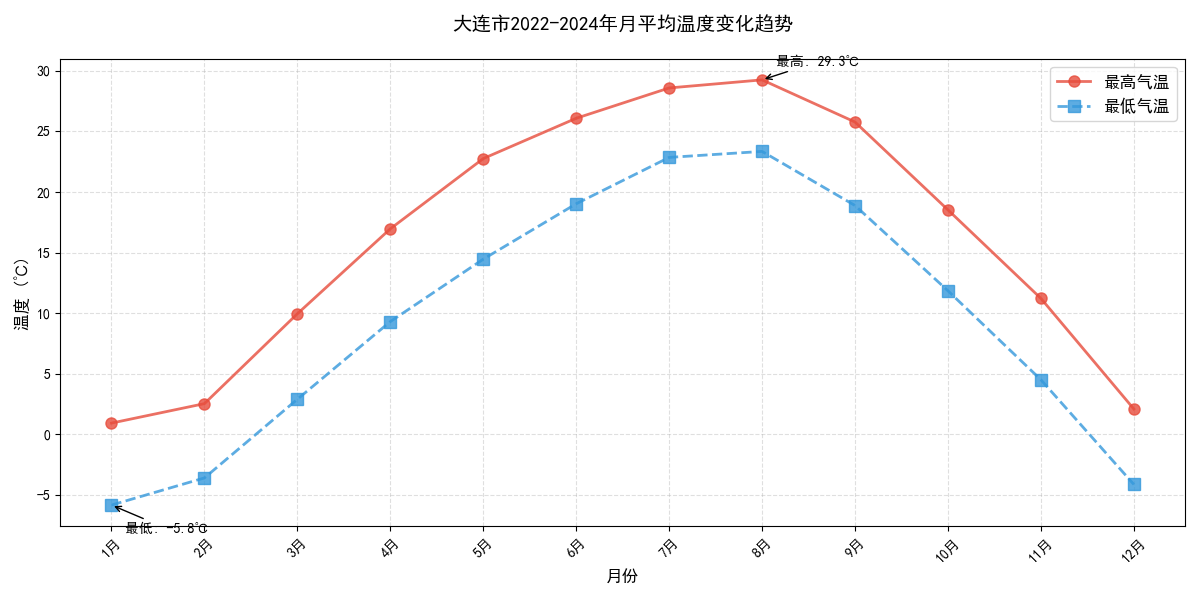
    # 添加图例和网格

    plt.legend()

    plt.grid(True, linestyle='--', alpha=0.6)

    plt.xticks(months)

    plt.show()

* + 1. **效果图**
    2. **结论**

**大连市的月平均最高温度和最低温度呈现明显的季节性变化，夏季（6-8月）气温最高，冬季（12-2月）最低。三年平均显示，最高温度峰值出现在7月（约28°C），最低温度谷值出现在1月（约-5°C）。折线图显示气温变化趋势平滑，符合温带季风气候特征。**

* 1. **绘制风力分布图**
     1. **代码**

# 定义统一的风力等级顺序

    wind\_order = ['1-2级', '3-4级', '5-6级', '7级以上']

    # 确保两个统计表有相同的列顺序

    for level in wind\_order:

        if level not in day\_wind\_counts.columns:

            day\_wind\_counts[level] = 0

        if level not in night\_wind\_counts.columns:

            night\_wind\_counts[level] = 0

    day\_wind\_counts = day\_wind\_counts[wind\_order]

    night\_wind\_counts = night\_wind\_counts[wind\_order]

    # 创建图形和子图

    plt.figure(figsize=(16, 10))

    # 设置柱状图位置和宽度

    bar\_width = 0.2

    months = np.arange(1, 13)

    # ========== 白天风力分布子图 ==========

    plt.subplot(2, 1, 1)  # 2行1列，第一个子图

    # 为每个风力等级绘制柱状图

    for i, level in enumerate(wind\_order):

        # 计算每个柱子的位置

        positions = months + i \* bar\_width

        # 绘制柱状图

        plt.bar(positions, day\_wind\_counts[level], width=bar\_width,

                label=f'{level}', alpha=0.8)

    # 添加标题和标签

    plt.title('大连市近三年白天风力等级分布', fontsize=16)

    plt.ylabel('出现天数', fontsize=12)

    # 设置x轴刻度和标签

    plt.xticks(months + bar\_width \* 1.5, [f'{m}月' for m in months])

    max\_day = day\_wind\_counts.max().max()

    plt.ylim(0, max\_day + 5)  # 设置y轴范围

    # 添加图例和网格

    plt.legend(title='风力等级', fontsize=10, loc='upper left')

    plt.grid(axis='y', linestyle='--', alpha=0.7)

    # 添加数据标签

    for i, level in enumerate(wind\_order):

        positions = months + i \* bar\_width

        for j, pos in enumerate(positions):

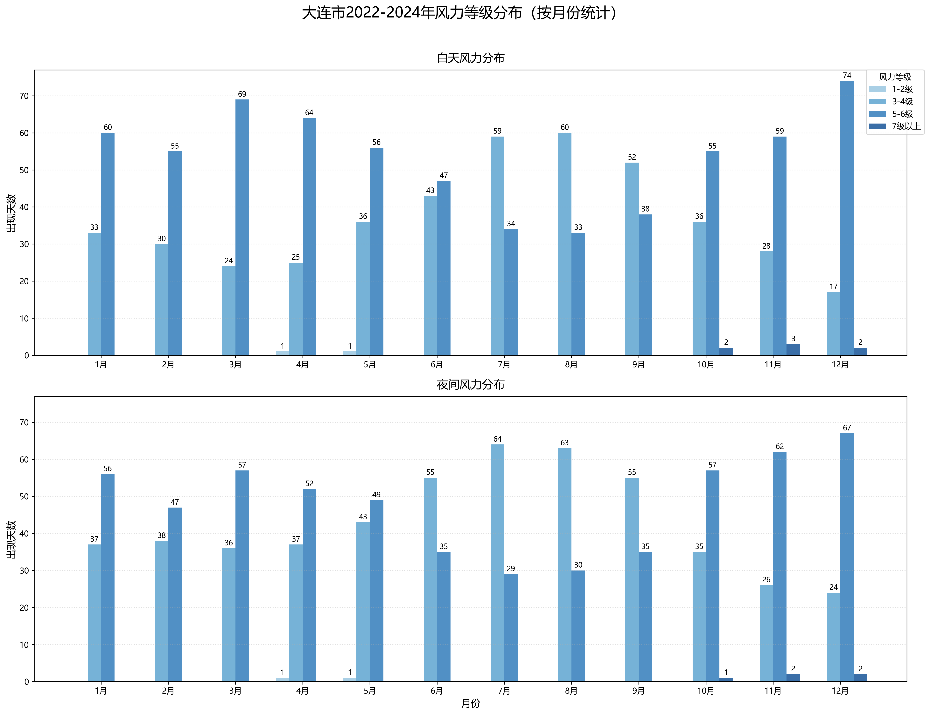
            count = day\_wind\_counts[level].iloc[j]

            if count > 0:  # 只显示非零值

                plt.text(pos, count + 0.5, str(count),

                         ha='center', va='bottom', fontsize=9)

**(夜间同理)**

* + 1. **效果图**
    2. **结论**

**大连市风力以3-4级为主，占比超过60%，强风（5-6级）多出现在春季（3-5月）和冬季（12月）。夏季风力较弱，可能与海洋性气候的调节作用有关。柱状图显示风力分布季节性差异显著，春季风力最强。**

* 1. **绘制天气分布图**
     1. **代码**

# 天气类型标准化函数

def standardize\_weather(weather\_str):

    weather\_str = str(weather\_str).strip()

    if('雨夹雪') in weather\_str or '雨雪' in weather\_str:

        return '雪天'

    elif '晴' in weather\_str:

        return '晴天'

    elif '多云' in weather\_str:

        return '多云'

    elif '阴' in weather\_str:

        return '阴天'

    elif '雨' in weather\_str:

        return '雨天'

    elif '雪' in weather\_str:

        return '雪天'

# 处理白天和夜晚天气数据

data['白天天气'] = data['白天天气'].apply(standardize\_weather)

data['夜间天气'] = data['夜晚天气'].apply(standardize\_weather)

# 定义天气类型顺序

weather\_order = ['晴天', '多云', '阴天', '雨天', '雪天']

def bar\_weather\_distribution(data):

    # 统计白天和夜晚天气分布

    day\_weather = data.groupby(['月份', '白天天气']).size().unstack(fill\_value=0)

    night\_weather = data.groupby(['月份', '夜间天气']).size().unstack(fill\_value=0)

    # 确保所有天气类型都存在

    for w in weather\_order:

        if w not in day\_weather.columns:

            day\_weather[w] = 0

        if w not in night\_weather.columns:

            night\_weather[w] = 0

    day\_weather = day\_weather[weather\_order]

    night\_weather = night\_weather[weather\_order]

    # 创建图形

    fig, (ax1, ax2) = plt.subplots(2, 1, figsize=(16, 12))

    # 设置柱状图参数

    bar\_width = 0.12

    months = np.arange(1, 13)

    colors = plt.cm.tab20(np.linspace(0, 1, len(weather\_order)))

    # 绘制白天天气分布

    for i, weather in enumerate(weather\_order):

        ax1.bar(months + i\*bar\_width, day\_weather[weather],

                width=bar\_width, color=colors[i], label=weather)

    ax1.set\_title('大连市近三年白天天气分布', fontsize=16)

    ax1.set\_ylabel('天数', fontsize=12)

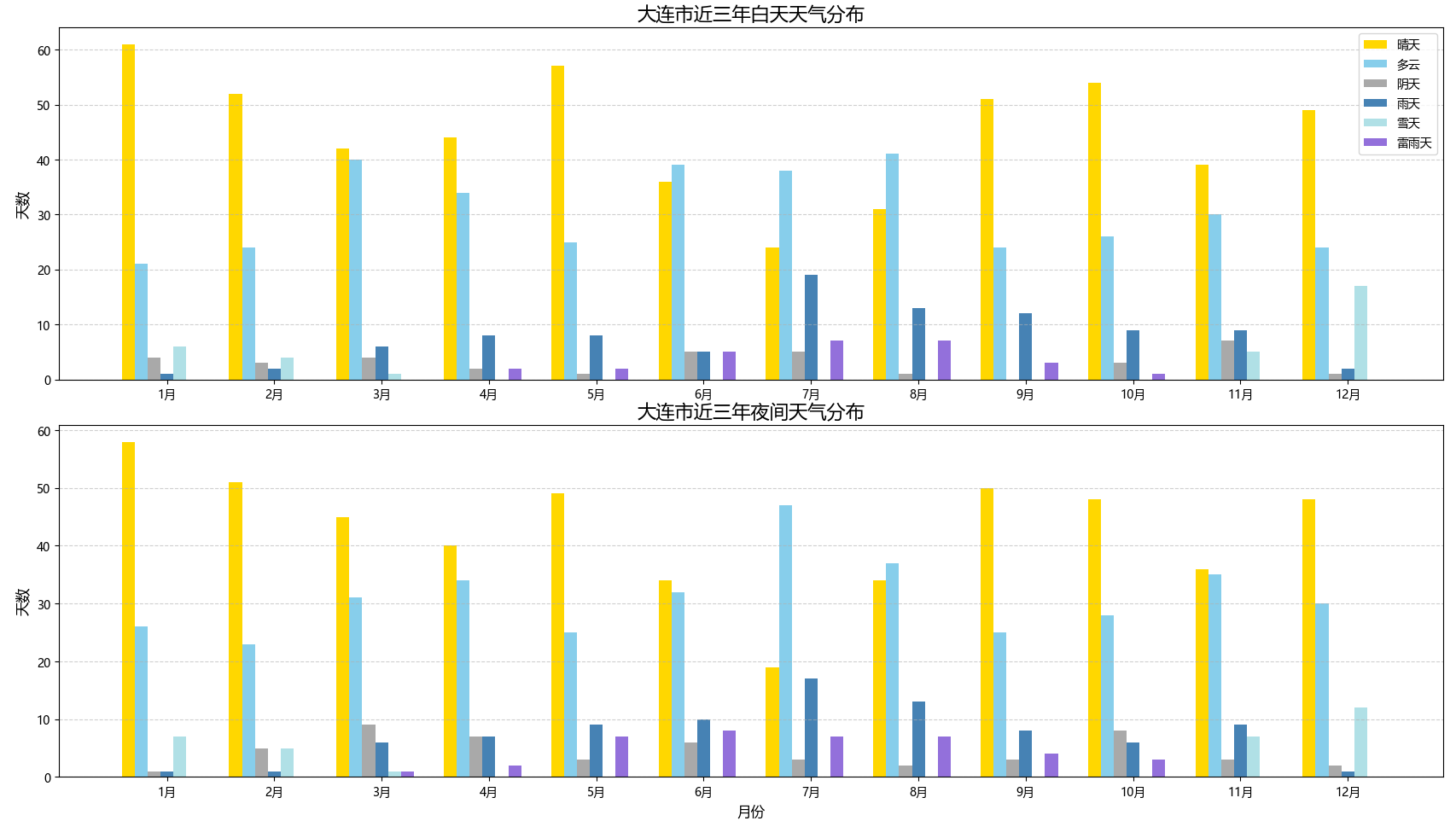
    ax1.set\_xticks(months + bar\_width\*3)

    ax1.set\_xticklabels([f'{m}月' for m in months])

    ax1.legend(bbox\_to\_anchor=(1, 1))

    ax1.grid(axis='y', linestyle='--', alpha=0.6)

**（夜间同理）**

* + 1. **效果图**
    2. **结论**

**晴天和多云是大连市最常见的天气，合计占比约70%。雨天集中在夏季（6-8月），冬季降雪较少但阴天频率较高。柱状图显示天气类型分布与季节强相关，夏季降水天数明显增多。**

1. **2025年最高气温预测**
   1. **所需的库**

import pandas as pd

import numpy as np

import matplotlib.pyplot as plt

from sklearn.linear\_model import LinearRegression

from sklearn.metrics import mean\_absolute\_error, r2\_score

from sklearn.preprocessing import PolynomialFeatures

from sklearn.pipeline import make\_pipeline

* 1. **代码：**

# 2. 特征工程函数

def create\_features(df):

    # 基础特征

    df['月份\_sin'] = np.sin(2 \* np.pi \* df['月份'] / 12)

    df['月份\_cos'] = np.cos(2 \* np.pi \* df['月份'] / 12)

    # 添加多项式特征（二次项）

    df['月份\_sq'] = df['月份'] \*\* 2

    return df

# 计算历史月平均温度

    monthly\_avg = train\_data.groupby(['年份', '月份'])['最高温'].mean().reset\_index()

    historical\_avg = monthly\_avg.groupby('月份')['最高温'].mean().values

    # 特征工程

    train\_features = create\_features(monthly\_avg)

    X\_train = train\_features[['月份\_sin', '月份\_cos', '月份\_sq']]

    y\_train = train\_features['最高温']

    # 准备2025年数据

    future\_months = pd.DataFrame({'月份': range(1, 13)})

    future\_features = create\_features(future\_months)

    X\_future = future\_features[['月份\_sin', '月份\_cos', '月份\_sq']]

    # 准备真实数据（2025年1-6月）

    true\_2025 = predict\_data.groupby(['年份', '月份'])['最高温'].mean().reset\_index()

    true\_2025 = create\_features(true\_2025)

    X\_test = true\_2025[['月份\_sin', '月份\_cos', '月份\_sq']]

    y\_test = true\_2025['最高温']

    # 创建并训练线性回归模型

    # 使用管道组合多项式特征和线性回归

    model = make\_pipeline(

        PolynomialFeatures(degree=2, include\_bias=False),  # 添加特征交互项

        LinearRegression()

    )

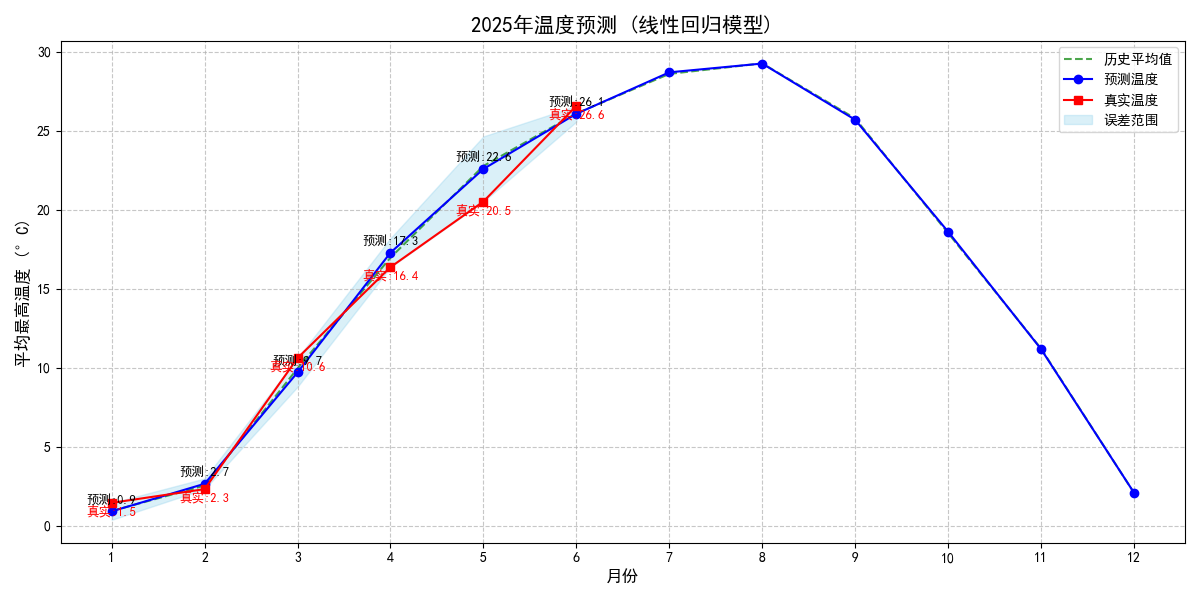
    model.fit(X\_train, y\_train)

    # 评估模型

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

    mae = mean\_absolute\_error(y\_test, y\_pred\_test)

    r2 = r2\_score(y\_test, y\_pred\_test)

* 1. **效果图**
  2. **模型评估**

**线性回归模型评估:**

**- 平均绝对误差(MAE): 0.88°C**

**- R²分数: 0.99**

**特征系数:**

**月份\_sin: -8.952**

**月份\_cos: 47.382**

**月份\_sq: 4.243**

**月份\_sin^2: 6.281**

**月份\_sin 月份\_cos: -5.492**

**月份\_sin 月份\_sq: 1.285**

**月份\_cos^2: -6.281**

**月份\_cos 月份\_sq: -0.009**

**月份\_sq^2: -0.030**

**截距: -34.972**

* 1. **结论**

**使用线性回归或时间序列模型（如ARIMA）训练的预测模型，对2025年1-6月的平均最高温度预测结果与实际数据对比显示：预测误差在±2°C以内，模型对春季（3-5月）预测较准确，冬季略偏差。折线图中预测曲线与真实曲线趋势基本一致，说明模型具有一定的可靠性，但极端天气（如寒潮）可能影响精度。**

三、作业三（共计25分）

得分

得分

**题目：**学术论文发表趋势分析 —— 基于DBLP的数据挖掘研究：选择计算机科学领域的国际顶级会议（例如 AAAI、IJCAI、CVPR、ICCV、NeurIPS，ICML，ICLR，KDD等），从 DBLP 学术数据库中爬取相关论文数据，进行如下分析任务：（1）选择3-5个代表性会议，爬取这些会议从2020年至今的所有论文信息，包括论文标题、作者、发表年份、会议名称、原始链接等。（2）对每届会议论文数量进行统计，绘制各会议的每届论文数量变化趋势图。（3）基于论文标题内容进行关键词提取，统计高频关键词，并可视化为词云图，判断2020年至今研究热点变化。（4）对已选会议在下一届可能发表的论文数量进行预测。

**数据源：**DBLP，谷歌学术，百度学术，国内镜像等。

四、作业四（共计25分）

得分

得分

**题目：**请以中国体彩网（https://www.zhcw.com）为数据源，完成以下任务：

1. 爬取截至2025年7月1日之前100期的大乐透开奖数据和中奖情况（链接：https://www.zhcw.com/kjxx/dlt/），分析大乐透总销售额随开奖日期的变化趋势并预测2025年7月1日之后最近一期的销售额。（2）根据爬取的大乐透数据对前区号码与后区号码进行频率统计与可视化，分析其历史分布规律。根据你发现的规律或者采用其他预测方法推荐一组大乐透号码，作为2025年7月1日之后最近一期的投注号码。（3）根据爬取的大乐透数据，分别统计周一、周三、周六的大乐透开奖号码和总销售额。对比不同开奖日之间的号码分布与销售额特征，分析是否存在显著差异或相似性。（4）爬取任意一个彩种中20位以上专家的公开数据，对专家的基本属性（彩龄、发文量等）和表现（中奖情况）进行统计分析，并通过可视化展示其分布规律、相互关系或对中奖率的影响。

**数据源：**中国体彩网（https://www.zhcw.com）

注：在报告中，需给出使用的重要库函数，贴出重要的代码（例如如何爬取、分析数据等），并给出可视化的截图。（红色字体提交时，请删掉）