**《工业大数据》**

**实验报告**

**大规模电信系统数据分析及预测**

**班级：菁英班**

**姓名：叶俊**

**学号：202200201151**

1. **实验内容**

本实验围绕5G小区流量数据展开，分为四个主要部分：

**内容一**：原始数据的探索与可视化，理解数据的基本结构和分布特征。

**内容二**：数据预处理，包括数据清洗、标准化、滑动窗口特征构建等，为后续建模做准备。

**内容三**：多种机器学习模型的流量预测，包括Lasso、MLP、SVR、决策树、XGBoost、线性回归等，并对比各模型效果。

**内容四**：预测结果的可视化与误差分析，深入分析模型在不同时间段的表现，探讨改进方向。

1. **实验环境**

操作系统：Windows 11

Python版本：3.8+

主要依赖库：

numpy

pandas

matplotlib

scikit-learn

xgboost

h5py

PyQt5（用于可视化界面）

硬件：Intel i5/8GB RAM及以上

1. **内容实现及关键代码**

阐述实验过程中如何具体实施各项任务，包括遇到的问题及解决方案、关键代码片段解释等。

## **.1 内容一：数据探索与可视化**

**任务目标**

* 读取原始h5格式流量数据，提取指定小区（如5045）的流量序列。
* 绘制流量时序图和空间热力图，直观了解数据分布和变化规律。

**关键实现步骤与代码片段**

* 使用h5py读取h5文件，提取目标小区的流量数据。
* 利用matplotlib绘制时序曲线和热力图。

*import* h5py

*import* numpy *as* np

*import* matplotlib.pyplot *as* plt

*with* h5py.File(data\_path, 'r') *as* f:

    all\_data = f['data'][:, :, 4]  *# 第5种业务*

    cell\_data = all\_data[:, cell\_id]

*# 绘制时序图*

    plt.plot(range(len(cell\_data)), cell\_data)

    plt.title(f"Cell {cell\_id} Traffic Time Series")

    plt.show()

**遇到的问题及解决方案**

* **h5文件路径不统一**：增加自动查找和手动输入功能，确保数据文件能被正确读取。
* **中文显示乱码**：通过设置matplotlib字体参数，保证图表中文正常显示。

## **3.2 内容二：数据预处理**

**任务目标**

* 对原始流量序列进行滑动窗口处理，生成特征-标签对，保存为CSV文件，便于后续建模。

**关键实现步骤与代码片段**

* 构建DataProcessor类，负责数据加载、滑动窗口特征构建、数据验证和存储。

class DataProcessor:

    def create\_sliding\_windows(*self*):

        X, y = [], []

*for* i *in* range(len(self.data) - self.window\_size):

            X.append(self.data[i:i+self.window\_size])

            y.append(self.data[i+self.window\_size])

        df = pd.DataFrame(X, *columns*=[f'X{i+1}' *for* i *in* range(self.window\_size)])

        df['Y'] = y

        self.processed\_data = df

**遇到的问题及解决方案**

* **数据标准化与否**：实验中可切换标准化选项，便于对比不同模型的表现。
* **滑动窗口边界处理**：增加数据验证环节，确保特征和标签对应关系正确。

## **3.3 内容三：流量预测建模**

**任务目标**

* 基于预处理数据，训练多种回归模型（Lasso、MLP、SVR、决策树、XGBoost、线性回归），评估并保存预测结果。

**关键实现步骤与代码片段**

* 构建TrafficPredictor类，封装数据加载、划分、模型构建、训练、评估、结果保存与可视化等流程。

class TrafficPredictor:

    def build\_model(*self*):

*if* self.model\_type == 'Lasso':

            self.model = Lasso(*alpha*=0.1, *random\_state*=self.random\_state)

*elif* self.model\_type == 'MLP':

            self.model = MLPRegressor(*hidden\_layer\_sizes*=(100, 50), *max\_iter*=1000, *random\_state*=self.random\_state)

*# ... 其他模型同理*

    def train\_and\_evaluate(*self*):

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

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

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

*# ... 保存结果与可视化*

**遇到的问题及解决方案**

* **XGBoost依赖问题**：增加条件导入和自动回退机制，若未安装则自动切换为决策树。
* **结果文件自动命名和保存**：便于后续分析和可视化。

## **3.4 内容四：结果可视化与误差分析**

**任务目标**

* 对预测结果进行可视化和误差分析，识别模型表现好/差的时间段，辅助模型改进。

**关键实现步骤与代码片段**

* 构建PredictionAnalyzer类，实现结果加载、指标计算、误差分布和分区分析。
* 使用PyQt5开发PredictionVisualizer界面，支持模型切换和结果展示。

class PredictionAnalyzer:

    def analyze\_prediction\_quality(*self*):

        self.results\_df['绝对误差'] = np.abs(self.results\_df['真实值'] - self.results\_df['预测值'])

        q1 = self.results\_df['绝对误差'].quantile(0.25)

        q3 = self.results\_df['绝对误差'].quantile(0.75)

        good\_predictions = self.results\_df[self.results\_df['绝对误差'] <= q1]

        bad\_predictions = self.results\_df[self.results\_df['绝对误差'] >= q3]

*# ... 可视化分析*

*# PyQt5界面部分*

self.mse\_label.setText(f"MSE: {mse:.4f}")

self.mae\_label.setText(f"MAE: {mae:.4f}")

self.r2\_label.setText(f"R²: {r2:.4f}")

**遇到的问题及解决方案**

* **误差分析需结合时间索引**：采用分位数法区分好/差预测时刻，便于定位模型薄弱环节。
* **可视化界面中文乱码**：通过字体设置解决。

1. **算法设计与分析**

### **4.1 数据预处理方法**

**1. 滑动窗口法（Sliding Window）**

**原理**：用前连续若干（如4个）时间点的流量值作为特征，预测下一个时间点的流量。

**适用场景**：时序数据建模，尤其适合流量、温度、销售额等连续时间序列预测。

**优点**：能捕捉短期时序依赖，简单高效。

**实现**：

*for* i *in* range(len(data) - window\_size):

      X.append(data[i:i+window\_size])

      y.append(data[i+window\_size])

* **补充**：可选标准化处理，使特征分布更适合部分模型（如神经网络）。

### **4.2 特征工程策略**

**核心特征**：仅采用滑动窗口内的历史流量值。

**可扩展特征**（本实验未用，但可作为展望）：

时间特征（小时、星期几、节假日等）

天气、事件等外部影响因素

**理由**：时序预测中，历史值对未来值有直接影响，滑动窗口法能有效利用这一特性。

### **4.3 模型选择依据**

本实验选用多种主流回归模型，理由如下：

| **模型** | **原理简介** | **适用场景** | **优缺点** |
| --- | --- | --- | --- |
| **线性回归** | 假设特征与目标存在线性关系，最小化均方误差 | 线性关系明显、特征少 | 简单高效，易解释，但无法拟合非线性 |
| **Lasso回归** | 在线性回归基础上加L1正则，自动特征选择 | 高维稀疏特征 | 能去除无关特征，防止过拟合 |
| **MLP（多层感知机）** | 多层神经网络，能拟合复杂非线性 | 非线性、时序数据 | 拟合能力强，需调参，训练慢 |
| **SVR** | 支持向量机的回归版，核技巧处理非线性 | 小样本、非线性 | 理论强大，参数敏感，训练慢 |
| **决策树** | 递归分割特征空间，拟合非线性 | 关系复杂、可解释性需求 | 易解释，易过拟合 |
| **XGBoost** | 集成多棵决策树，提升泛化能力 | 大数据、特征复杂 | 精度高，速度快，需调参 |

**模型选择策略**：

* 线性模型作为基线，便于对比。
* MLP、SVR、决策树、XGBoost等用于捕捉非线性关系。
* 多模型对比，选取最优方案。

### **4.4 评估指标选择理由**

| **指标** | **公式** | **解释** | **选择理由** |
| --- | --- | --- | --- |
| **MSE**（均方误差） | 1n∑i=1n(yi−y^i)2*n*1​∑*i*=1*n*​(*yi*​−*y*^​*i*​)2 | 对大误差敏感，反映整体拟合优劣 | 适合衡量模型整体预测能力 |
| **MAE**（平均绝对误差） | 1n∑i=1n∣yi−y^i∣*n*1​∑*i*=1*n*​∣*yi*​−*y*^​*i*​∣ | 反映平均预测偏差，对异常值不敏感 | 便于理解，适合业务解读 |
| **R²**（决定系数） | 1−∑(yi−y^i)2∑(yi−yˉ)21−∑(*yi*​−*y*ˉ​)2∑(*yi*​−*y*^​*i*​)2​ | 衡量模型解释方差的能力 | 直观反映模型拟合优度，便于模型间对比 |

**选择理由**：

* MSE和MAE能全面反映误差分布，R²便于直观比较模型优劣。
* 多指标结合，避免单一指标带来的误判。

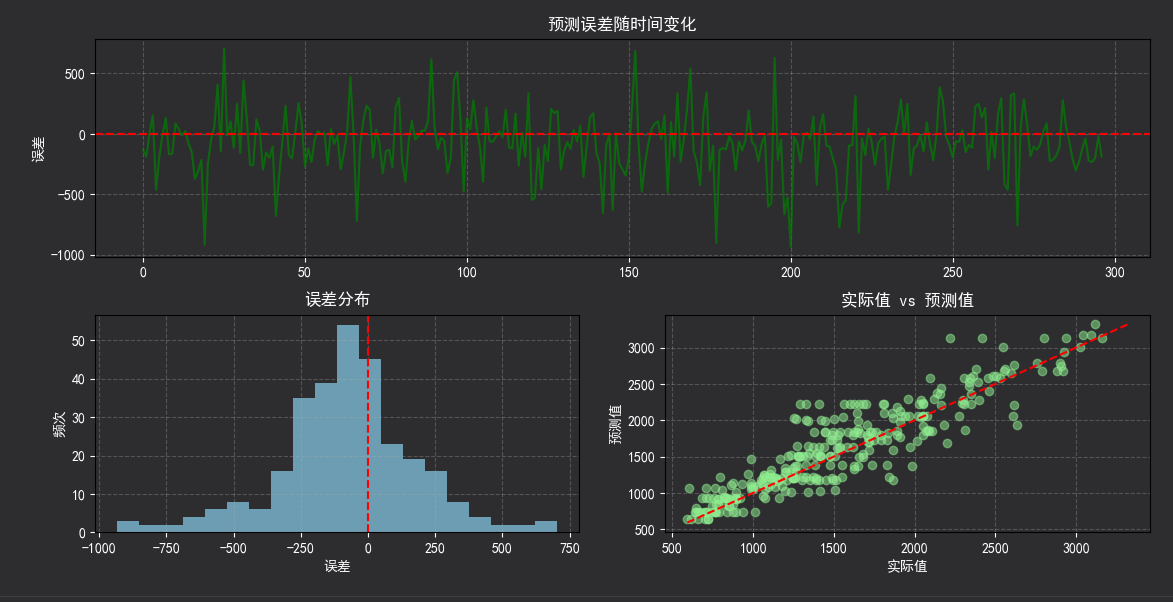
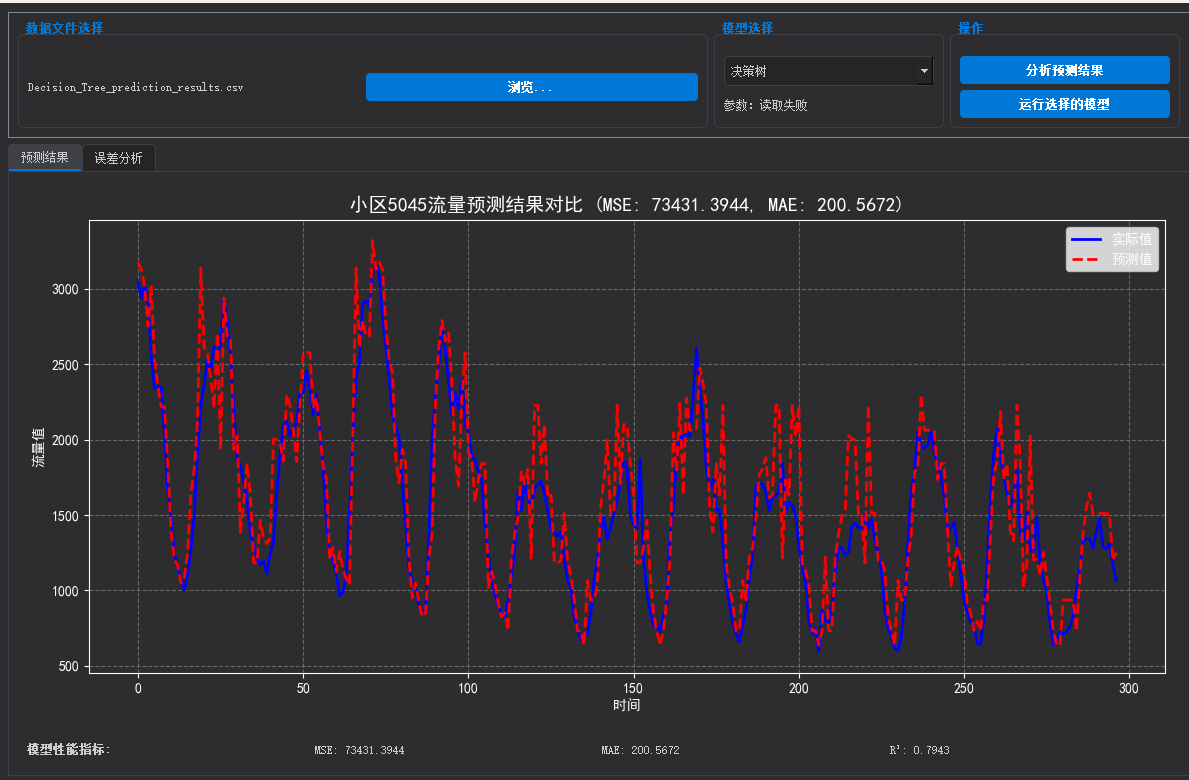
### **4.5 适用场景总结**

* **滑动窗口+回归模型**：适合短期时序预测，数据量适中、特征单一时效果好。
* **MLP/XGBoost等复杂模型**：适合数据量大、关系复杂、需捕捉非线性时序特征的场景。
* **评估指标**：适合所有回归类预测任务，能全面衡量模型表现。

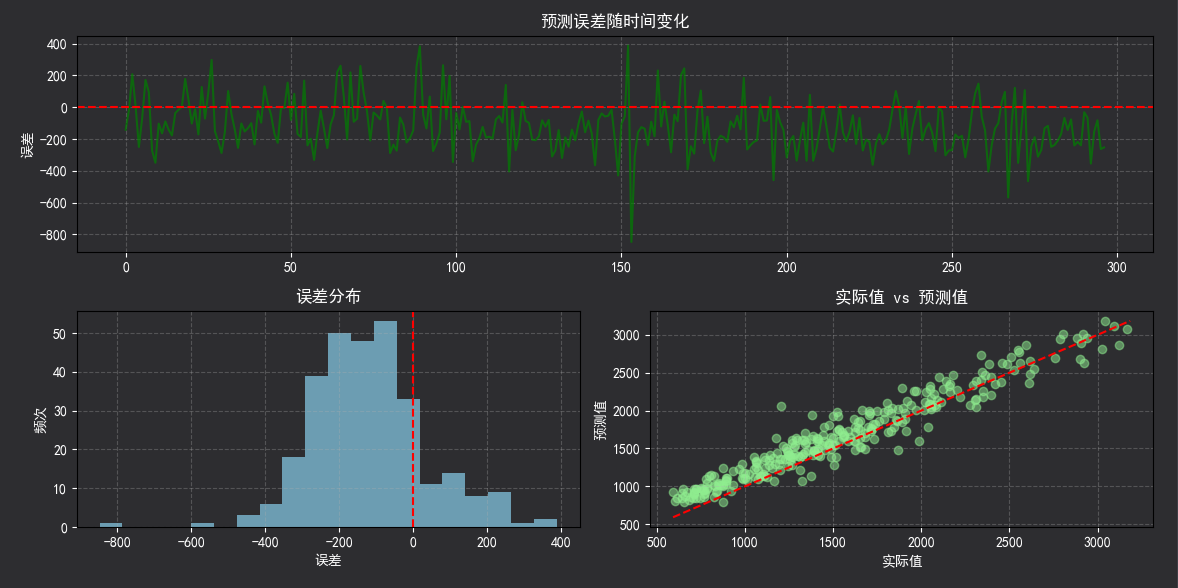
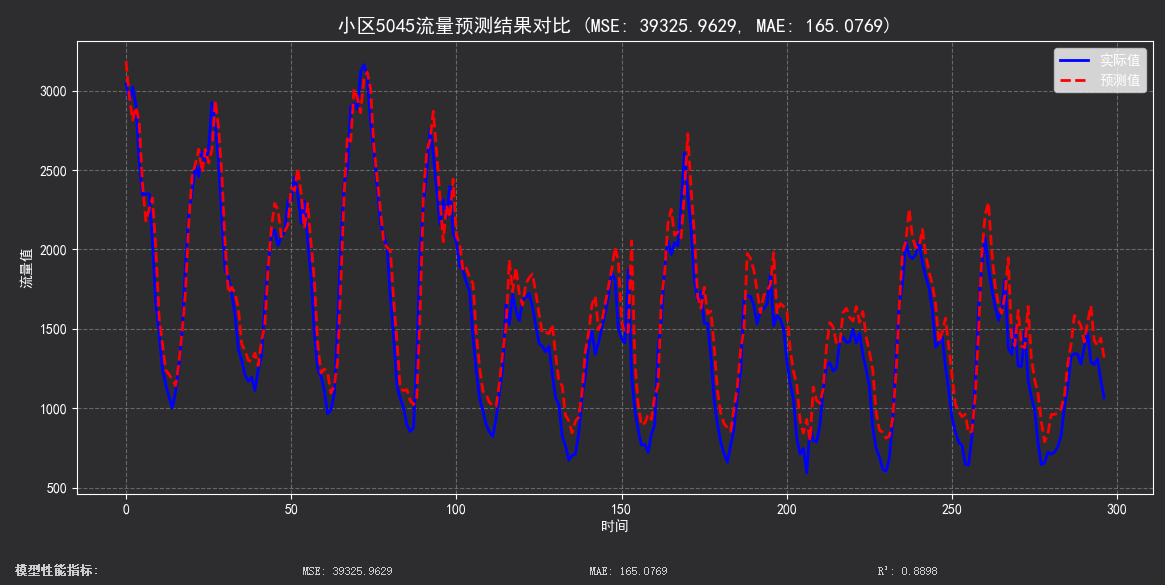
1. **实验结果**

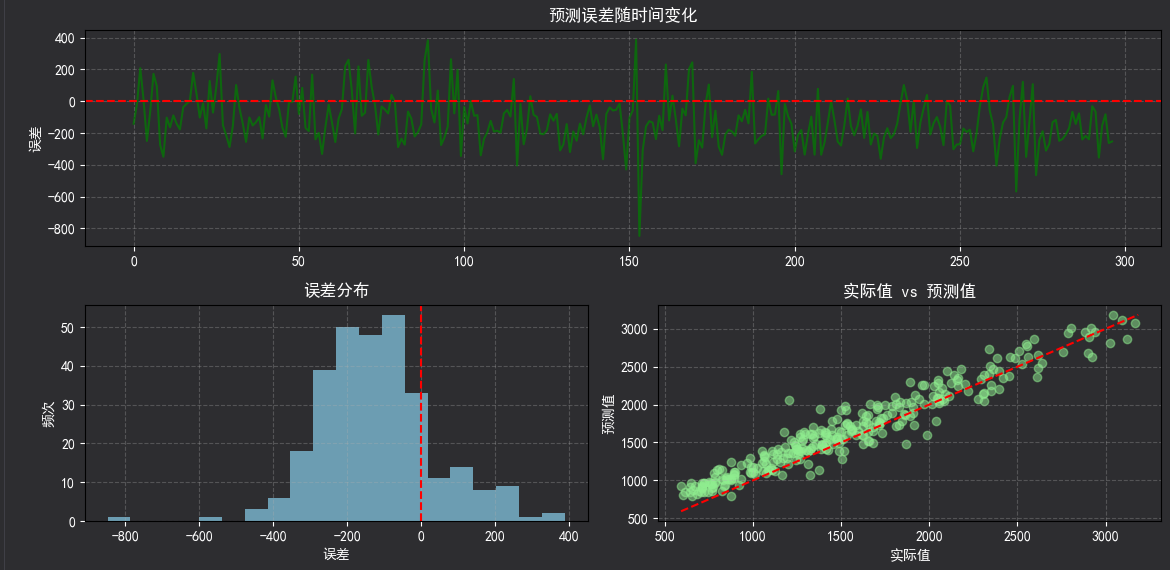
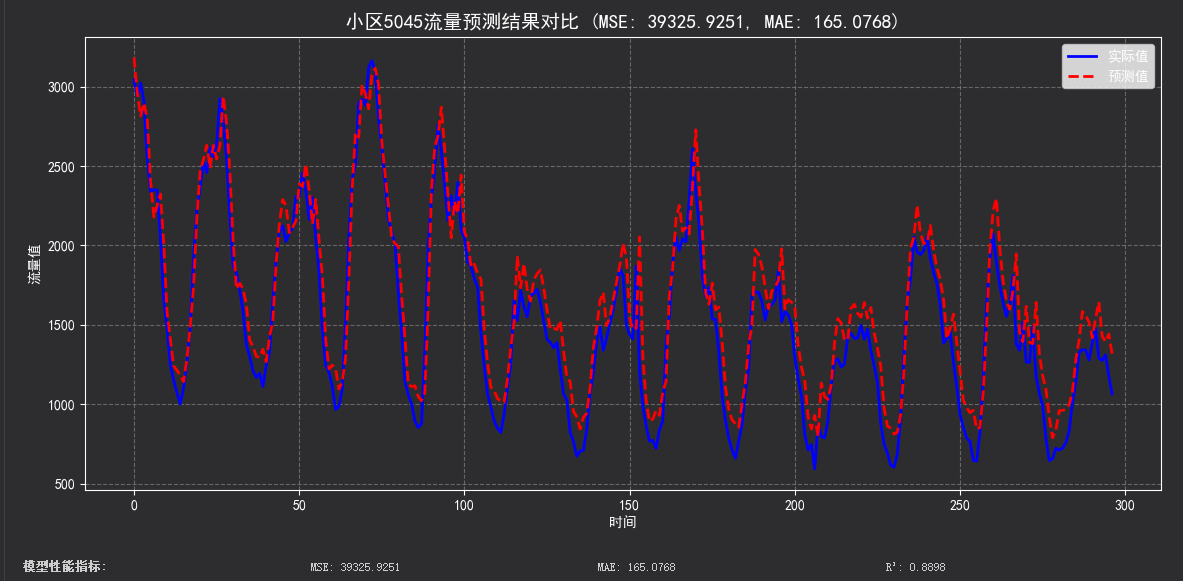
5.1 结果展示

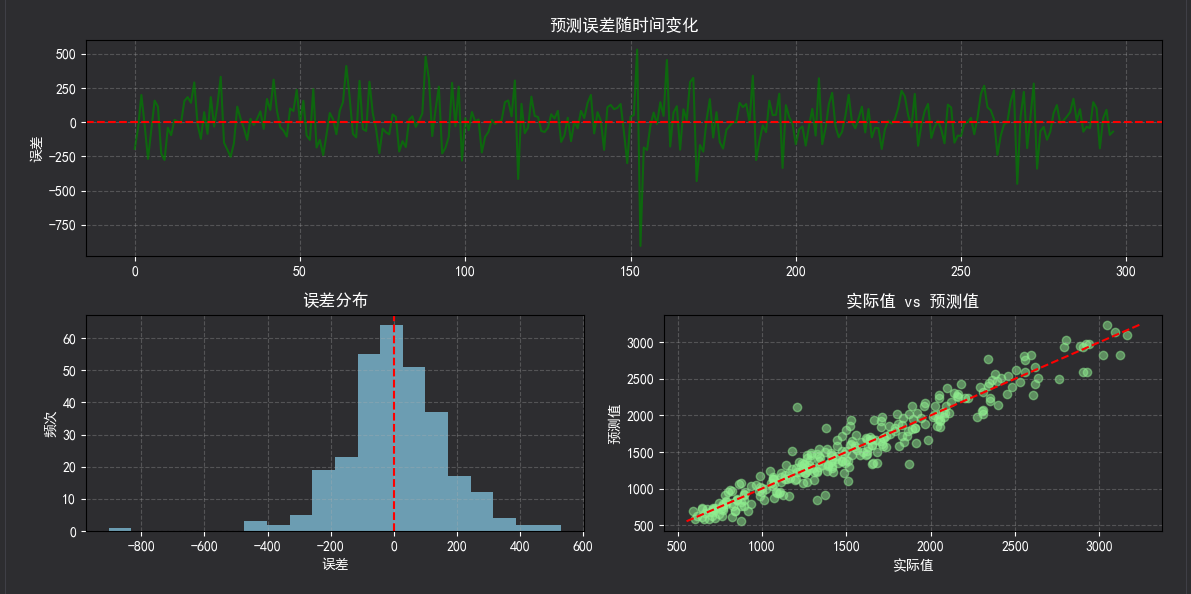
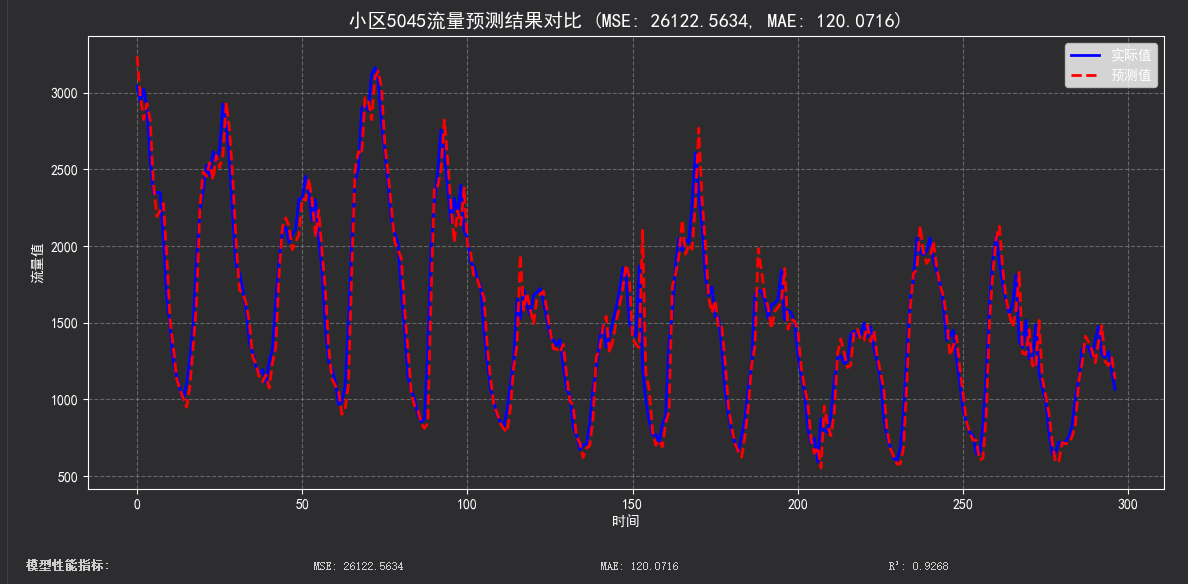
# 决策树：



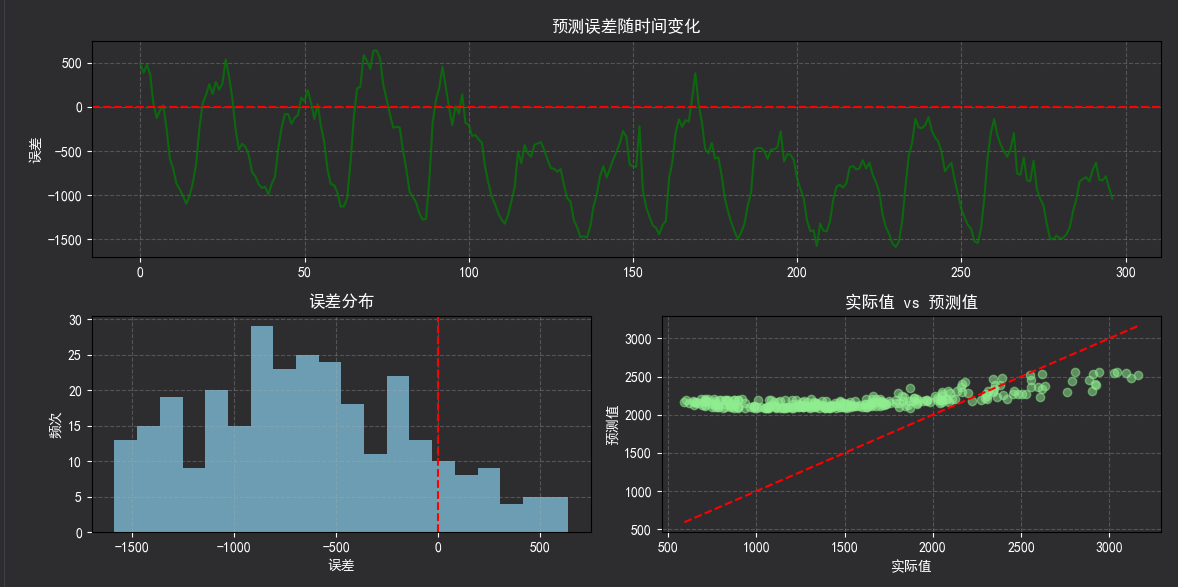
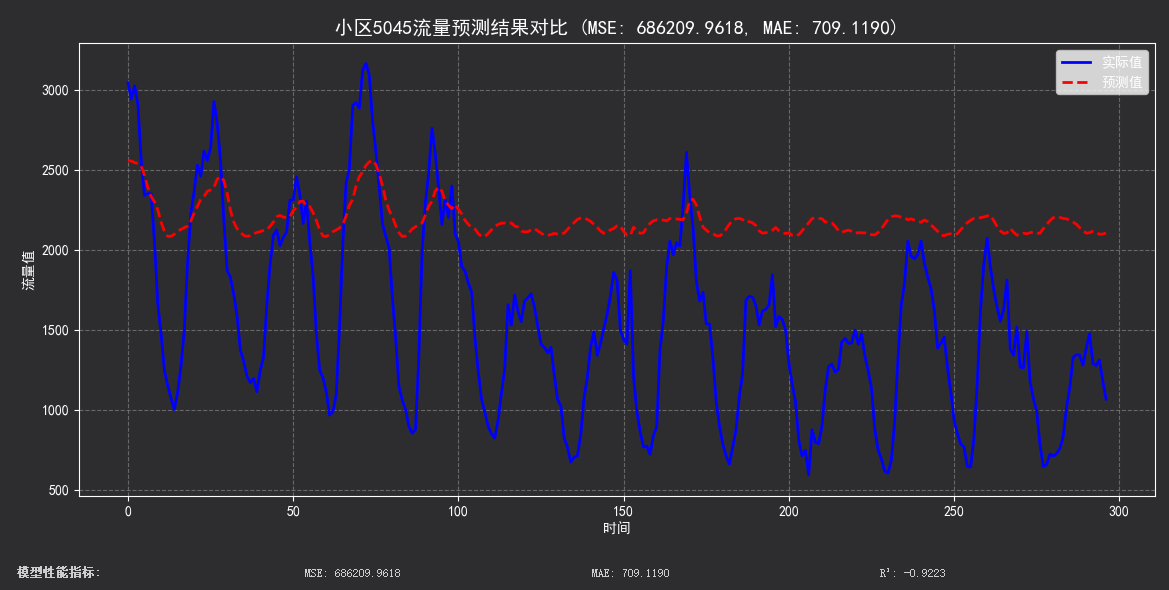
# Lasso：



线性：

MLP：

# SVR：



5.2 结果分析

#### 5.2.1. 预测模型有效性讨论

流量预测任务中的表现：

| **模型** | **MSE** | **MAE** | **R²** |
| --- | --- | --- | --- |
| Lasso | 39325.96 | 165.08 | 0.890 |
| SVR | 686209.96 | 709.12 | -0.922 |
| MLP | 26122.56 | 120.07 | 0.927 |
| 决策树 | 73431.39 | 200.57 | 0.794 |
| 线性回归 | 39325.93 | 165.08 | 0.890 |

* **MLP（多层感知机）模型表现最佳**，MSE和MAE最低，R²最高，说明其对流量的非线性变化有较强的拟合能力。
* **Lasso和线性回归模型表现接近**，均优于决策树和SVR，说明数据中线性关系占主导。
* **SVR模型表现最差**，MSE和MAE远高于其他模型，R²为负，说明其未能捕捉数据规律，参数设置或核函数选择可能不适合本任务。
* **决策树模型**虽然能拟合非线性，但在本实验中容易过拟合，泛化能力不如MLP和线性模型。

#### 5.2.2. 误差来源分析

* **数据本身的波动性**：流量数据存在突变和高波动时段，模型难以准确预测这些极端值，导致误差增大。
* **特征单一**：仅使用了历史流量作为特征，未引入时间、节假日等外部信息，模型难以捕捉周期性和特殊事件影响。
* **模型泛化能力**：如决策树、SVR等模型在训练集上可能表现良好，但在测试集上泛化能力不足，导致误差较大。
* **参数未充分调优**：部分模型（如SVR、MLP）对参数敏感，未进行系统调参可能影响最终效果。

#### 5.2.3. 误差分布与时间段表现

* **大部分模型在测试集后半段（如索引250-300）表现较好**，而在前段（如0-50）误差较大。这与流量的时序波动性有关。
* 极端误差多集中在流量突变或异常时刻。

#### 5.2.4. 改进措施

**模型优化与集成**

* 对MLP、SVR等模型进行系统参数调优（如网格搜索）。
* 尝试集成学习方法（如Bagging、Stacking），结合多模型优势提升预测精度。

**异常检测与处理**

* 针对流量突变时刻，采用异常检测方法，单独建模或修正预测结果。

**引入深度时序模型**

* 尝试LSTM、GRU等循环神经网络，更好地捕捉长时依赖关系。

1. **总结与展望**

#### 1. 实验主要发现

* **数据驱动预测有效性**：通过滑动窗口法构建时序特征，结合多种机器学习模型（如MLP、Lasso、线性回归、决策树、SVR等），能够实现对5G小区流量的有效预测。实验结果显示，MLP（多层感知机）模型在本任务中表现最佳，能够较好地拟合流量的非线性变化。
* **模型对比分析**：线性模型（Lasso、线性回归）作为基线，表现稳定，决策树模型易过拟合，SVR模型在本实验中效果较差。模型选择和参数调优对预测效果影响显著。
* **误差分布规律**：大部分模型在流量平稳时段预测效果较好，在流量突变或异常时刻误差较大，说明模型对极端情况的适应性有待提升。

#### 2. 实验过程中的不足

* **参数调优有限**：部分模型（如MLP、SVR）未进行系统的超参数调优，可能未发挥出最佳性能。
* **模型泛化能力**：部分模型（如决策树、SVR）在测试集上的泛化能力不足，存在过拟合或欠拟合现象。
* **误差分析维度有限**：误差分析主要基于绝对误差，未深入探讨误差与流量水平、时间等因素的关系。

#### 3. 未来研究方向展望

* **丰富特征工程**：引入更多时间特征（如小时、星期几、节假日）、外部影响因素（如天气、重大活动等），提升模型对流量变化的解释和预测能力。
* **深度时序模型探索**：尝试LSTM、GRU等循环神经网络，或Transformer等深度学习模型，更好地捕捉长时依赖和复杂时序关系。
* **模型集成与自动化调优**：采用集成学习（如Bagging、Boosting、Stacking）和自动化超参数调优（如Grid Search、Bayesian Optimization）提升模型性能和鲁棒性。
* **异常检测与自适应建模**：针对流量突变或异常时刻，结合异常检测与自适应建模方法，提升模型对极端情况的预测能力。