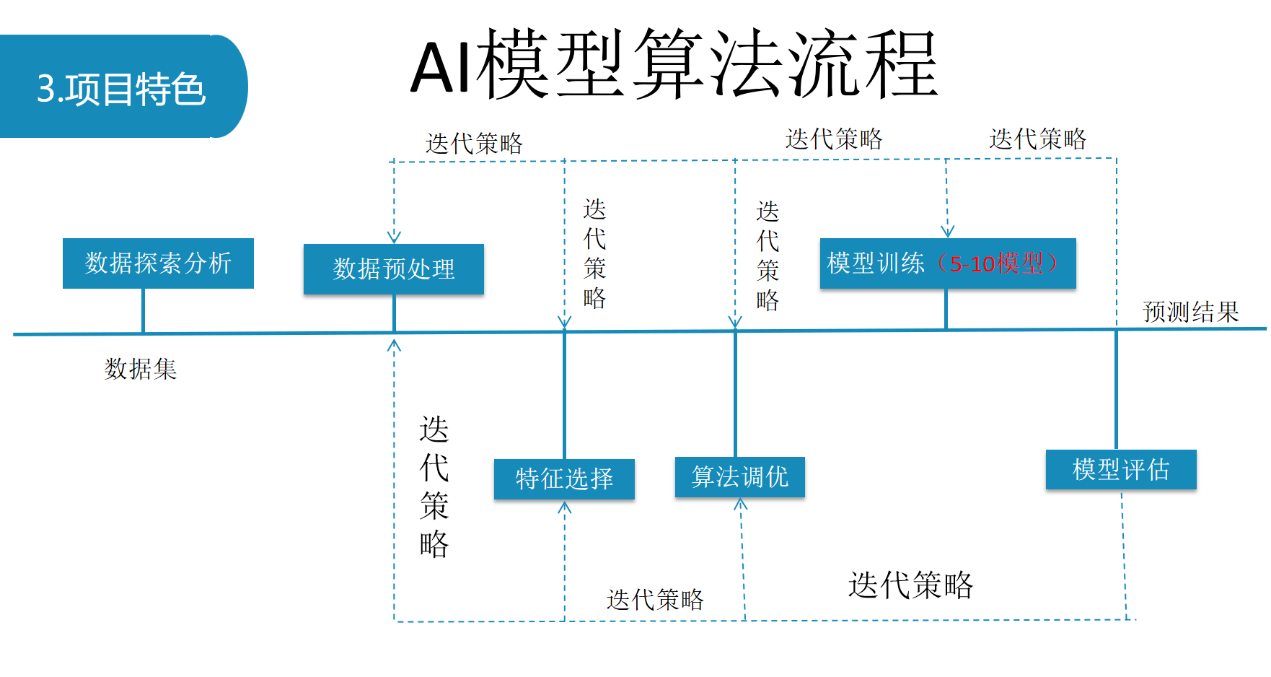
大数据模型推理的交通预测数据分析与智能决策系统

我是项目的数据算法工程师， 接下来我将从算法的设计到算法的实现来介绍我的工作

首先是算法的设计部分，它相当于系统的“大脑”，负责构建和优化预测模型，为后期模型预测奠定基础



如图展示是算法设计的流程图，整体逻辑可以分为数据阶段 → 模型构建与优化 →迭代策略 → 预测输出阶段，最终导出预测结果。

**首先**，数据探索阶段对原始数据进行全面分析和观察，掌握数据的整体情况。

，在数据预处理阶段，对数据进行梳理与清洗，特征提取，数据标准化保证数据质量

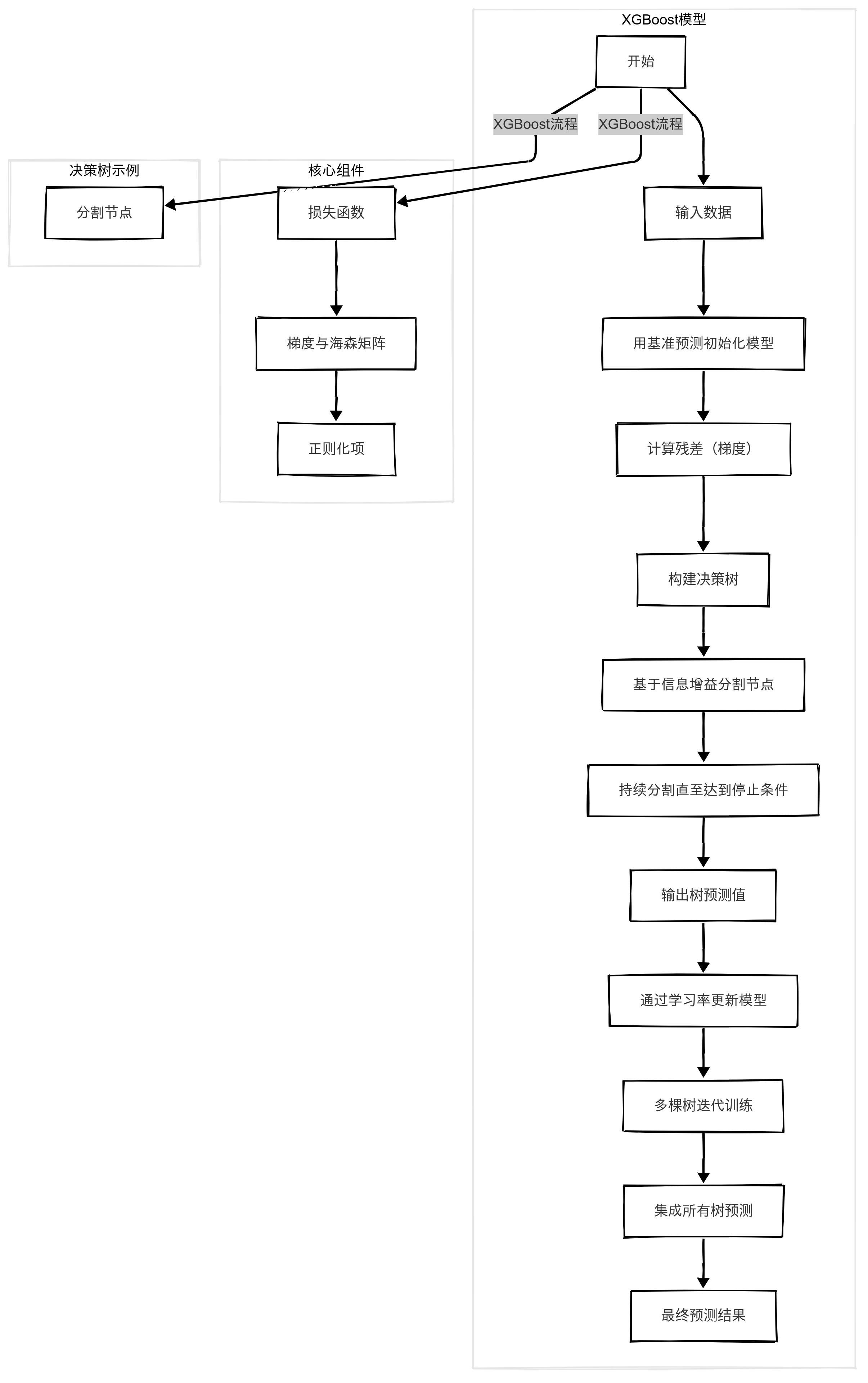
**接下来**，迭代策略贯穿整个流程，它涵盖了涵盖特征选择、算法调优等方面。在特征选择方面：从原始特征中选择有用特征剔除无关或冗余特征，以提高模型性能及训练效率。

在算法调优方面：我们采用网格搜索自动遍历所有参数组合（如3×4×3=36种组合）进行超参数迭代优化，其核心思想是​​通过穷举所有可能的超参数组合，结合交叉验证评估性能，从而找到最优参数配置​​。**并**每次用80%数据训练，20%交叉验证，避免偶然性

超参调优就像给AI模型"定制考试复习计划"——通过系统调整模型的学习规则（如每天刷题量、思考深度），形成不断反馈，不断调整的闭环，通过混淆矩阵和学习曲线分析模型的过拟合和欠拟合情况，指导模型后续优化。

整个流程并非简单的线性顺序，每一个环节的反馈都有会引导返回上游环节进行重新调整，目的在于不断逼近最优解。整体思路体现了从数据出发，以策略选择和模型训练为核心，通过持续优化实现目标导向的过程。

**然后**模型的选择和训练，在模型上我们考虑了XGBoost，梯度提升，随机森林，神经网络，SVM，决策树，逻辑回归这几种常见的机器学习模型



XGBoost是一种基于梯度提升的优化模型，每个模型都试图修正前一个模型的错误，就像是一个超级团队，它在梯度提升的基础上做了很多改进。它不仅考虑每个成员的错误，还会考虑如何让整个团队的判断更稳定、更准确

|  |  |  |
| --- | --- | --- |
| 模型类型 | **代表算法** | 技术特性 |
| 集成学习 | 随机森林/XGBoost | 抗过拟合、并行计算 |
| 深度学习 | 神经网络 | 复杂模式捕捉 |
| 传统模型 | SVM/逻辑回归 | 线性可分场景 |

候选模型池：

接下来，我们对多种模型进行了性能评估，以下是模型的性能对比

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **模型** | **准确率** | **精确率** | **召回率** | **F1分数** |
| 随机森林 | 99.0% | 99.0% | 99.0% | 99.0% |
| XGBoost | 99.9% | 99.9% | 99.9% | 99.9% |
| 神经网络 | 91.0% | 91.0% | 91.0% | 90.8% |
| SVM | 91.7% | 91.7% | 91.7% | 91.5% |
| 梯度提升 | 99.9% | 99.9% | 99.9% | 99.9% |
| 逻辑回归 | 85.0% | 85.0% | 85.0% | 85.0% |
| 决策树 | 90.0% | 90.0% | 90.0% | 90.0% |

如表中模型性能评估表格中，展示了这几种模型在准确率、精确率、召回率以及 F1 分数这四个关键指标上的表现，其中：准确率和精确率是两个不同的概念，准确率是“猜对多少”，精确率是“说对多少”

其中XGBoost、梯度提升和随机森林和SVM表现优秀,它们在性能、效率和适用性上表现最佳，能够满足系统对高精度和实时性的需求。

## 二：系统人工智能算法的实现（4.6）

如图4- 5所示，我们的系统人工智能算法详细设计通过模块化的方式实现了从数据读取到模型预测的完整流程，详细描述了在系统中的具体应用，包括数据处理、模型训练、在线预测和结果展示。

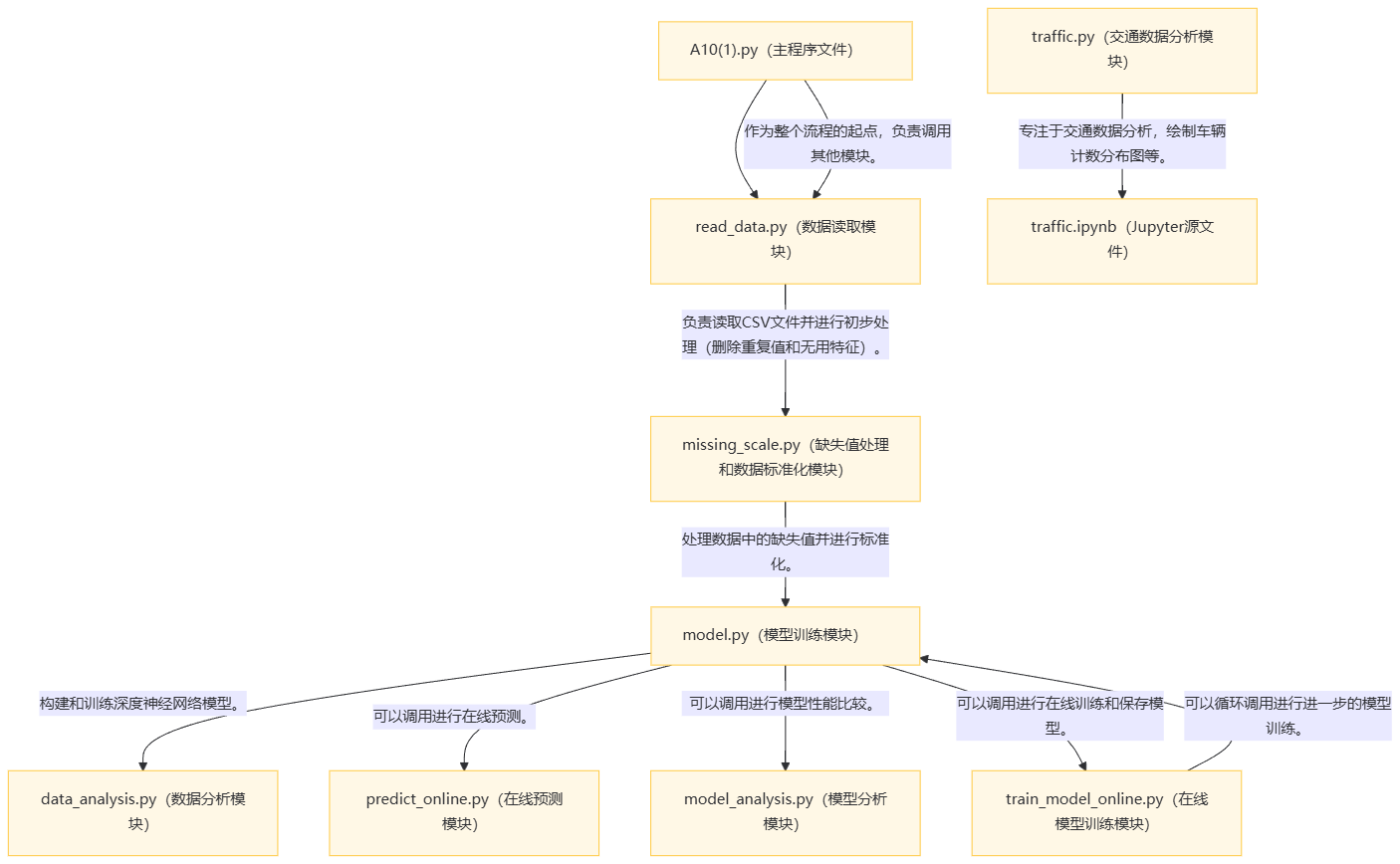


图4- 5 人工智能算法详细图

### 接下来我将重点介绍模型训练，模型分析，交通数据分析这几个模块de代码实现

首先是模型训练模块

### 1.model.py（模型训练模块）：

用如图4- 10所示，即为**模型训练代码。**

图4- 10 **模型训练代码**

通过定义train\_model函数，实现了神经网络模型的构建、训练、评估和保存的完整流程。首先，当没有提供验证集时，函数会自动将训练数据按8:2划分；接着，计算类别权重以处理类别不平衡问题,训练结束后，使用验证集评估模型性能。整个模块覆盖了从数据划分、建模、训练、评估到模型持久化的关键步骤，是深度学习建模流程中的核心部分。

### 2.model\_analysis.py（模型分析模块）：

 是进行模型性能比较调用的模块，还可以与 train\_model\_online.py 协同进行在线训练和保存模型。如图4- 12所示，即为**模型分析代码。**



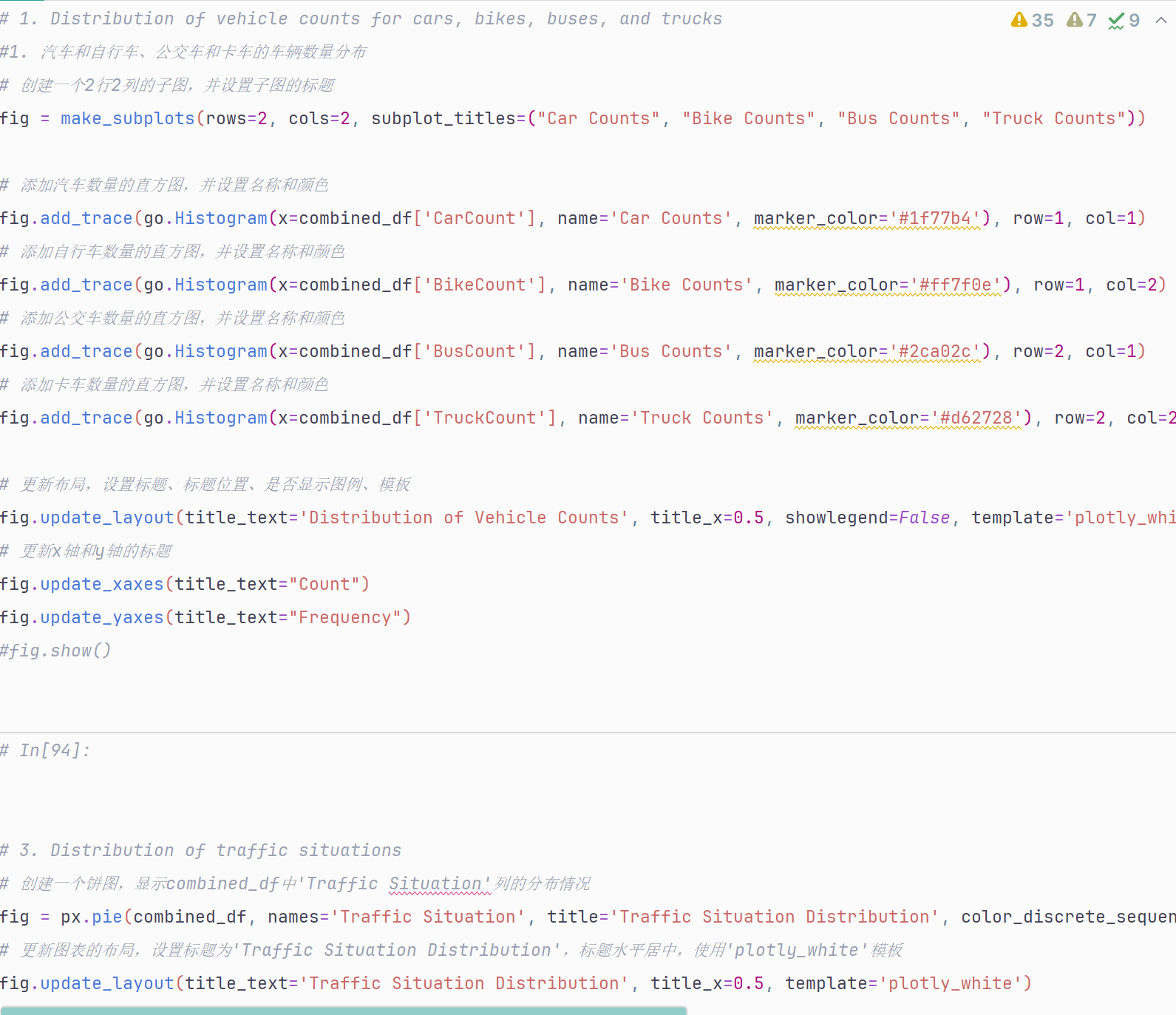
该主程序文件实现了从数据加载到多模型自动化训练评估的全流程：首先通过命令行参数接收训练/验证集路径，执行数据清洗、缺失值填充、标准化预处理；随后并行模型训练，决策树和随机森林三类模型；通过计算加权F1值、准确率等指标生成分类报告表格，并自动绘制模型对比柱状图，最终整合所有评估结果输出结构实现端到端的建模分析报告生成。如图4- 13所示，即为报告生成代码。

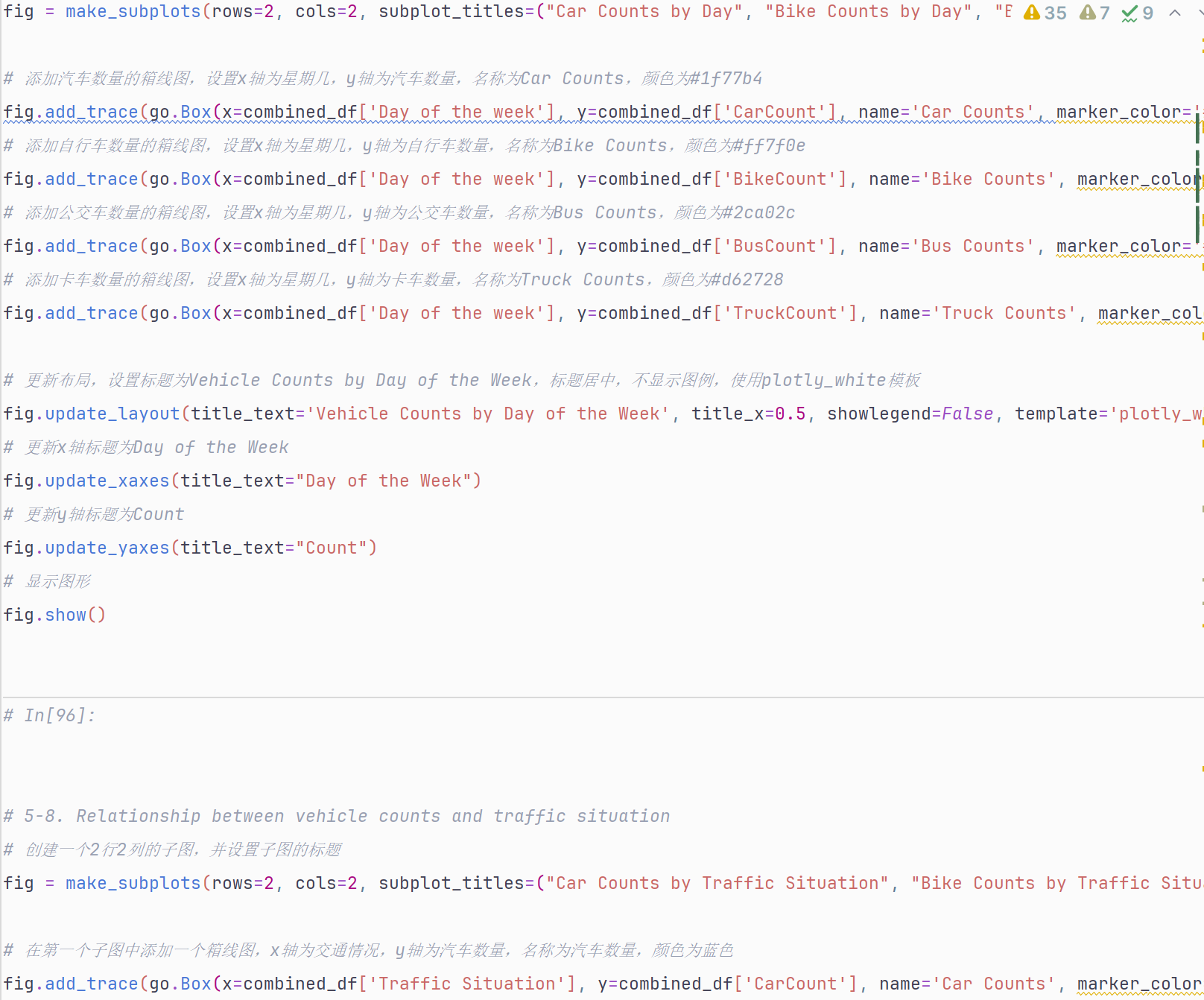
表4- 13多种模型

### 3.traffic.py（交通数据分析模块）：

专注于交通数据分析，绘制车辆计数分布图等，可以与 traffic.ipynb（Jupyter源文件）配合使用。如图4- 17所示，即为交通数据分析代码。







在数据分析中，我们可以借助 matplotlib 和 seaborn 这两个强大的库来进行数据可视化。它们就像是神奇的画笔，能把枯燥的数据变成直观的图形，让我们一目了然。

可以绘制直方图、箱线图、饼图等多种图形，来直观呈现数据的分布情况。“绘画工具”。。

通过这些直观的图形，我们能快速把握数据的分布特征，就像在黑暗中找到了明灯，轻松发现数据里的异常值和潜在问题

以上是本项目的数据算法核心部分详解