项目总览

项目名： mlops-bikeshare

一句话目标： 利用公开 API（GBFS + Meteostat）构建一个企业级 **MLOps 最小闭环**：数据采集→特征工程→训练与模型注册（SageMaker/MLflow）→CI/CD 自动部署→在线推理→监控与告警→业务可视化。

业务问题： 预测未来 30 分钟某站点是否会 **缺车（stockout bikes）** 或 **缺桩（stockout docks）**，为运维调度提供优先级告警（可扩展为再平衡优化建议）。

核心 KPI：

- 模型：PR-AUC ≥ 0.70，F1（缺车类）≥ 0.55；
- 系统：端点 P95 时延 ≤ 200ms；推理批次成功率 ≥ 99%；
- 数据：入湖数据延迟 ≤ 3 分钟，数据质量检查通过率 ≥ 99%；
- 监控：数据/模型漂移告警可用，MTTA ≤ 10 分钟。

---

1. 数据

1.1 公开数据源（免密钥）

- **GBFS（Global Bikeshare Feed Specification）**
- 入口（NYC）：`https://gbfs.citibikenyc.com/gbfs/gbfs.json`（可切换城市）
- 关键文件：
  - `station_information.json`（站点静态信息：经纬度、容量、区域）
  - `station_status.json`（站点动态状态：可用车/桩、更新时间）
- 采样频率：1–5 分钟
- **Meteostat 天气**
- 历史/当前/预报，字段：温度、降水、风速、天气类型等

1.2 数据契约（Data Contract）

- 强制字段（status）：`station_id, num_bikes_available, num_docks_available, last_reported`
- 强制字段（info）：`station_id, name, capacity, lat, lon`
- 时间戳统一为 UTC；
- 缺失处理：缺失率 > 5% 或数值越界直接拒绝入湖；
- 监控：字段级 schema 校验（Pandera / Great Expectations）。

1.3 标签构造

- 时间点 `t`，以 `t+30min` 的 `num_bikes_available`（或 `num_docks_available`）≤ 阈值（默认 2）生成二分类标签：是否“缺车/缺桩”；

- 同时保留回归目标（剩余车/桩数）以便多任务/备选建模。

1.4 特征示例

- 时序滚动：过去 15/30/60 分钟净借出量、均值/方差、趋势；
- 站点静态：容量、区域、经纬度（网格/聚类编码）；
- 邻域聚合：邻站可用车/桩之和、加权距离；
- 时间特征：小时、工作日/周末、节假日；
- 天气：温度、降水、风速、天气类别（One-hot / Target encoding）。

---

2. 技术栈与整体架构

AWS：S3、Glue/Athena、Lambda、EventBridge、SageMaker（Training/Endpoint/Model Registry/Model Monitor）、CloudWatch、ECR、IAM、(可选) QuickSight / Managed Grafana。

MLOps：SageMaker Pipelines 或 MLflow（两种注册与部署路径二选一）；GitHub Actions（CI/CD）、Terraform（IaC）。

可视化：Streamlit（业务面板） + CloudWatch/Grafana（系统面板）。

2.1 架构数据流（ASCII）

```
[GBFS API]   [Meteostat]
     \          /
     [Lambda Ingest (5m)] -- Pandera/GE --> [S3 Raw] -- Glue/Athena --\
                                                |                      [Batch
Feature Job] -> [S3 Features]

|                                  |

|                                  v
                                          [SageMaker Training] -- MLflow/SM
Experiments
                                              |                    \
                             (A) SM Model Registry | (B) MLflow Registry \
                                              |                       \
                                         [SageMaker Endpoint (staging -> prod)]
                                              ^             \
[Lambda Inference (5-10m)] -- featurize_online --|              [Model Monitor]
   |                                             \
   v                                              [S3 Inference + Monitoring]
-- Athena
[Streamlit Dashboard] <---------------------- [CloudWatch/Grafana/Alarms]
```

---

3. 代码库结构

```
mlops-bikeshare/
├─ infra/terraform/         # S3/Glue/ECR/SM/Lambda/EventBridge/IAM/CloudWatch
├─ pipelines/
│  ├─ training_pipeline.py  # SageMaker Pipelines 编排
│  └─ register_and_deploy.py
├─ src/
│  ├─ ingest/
│  │  ├─ gbfs_ingest.py
│  │  └─ weather_ingest.py
│  ├─ features/
│  │  ├─ build_features.py
│  │  └─ schema.py
│  ├─ training/
│  │  ├─ train.py           # XGBoost/LightGBM + mlflow.autolog
│  │  └─ eval.py            # 阈值选择/模型卡输出
│  ├─ inference/
│  │  ├─ handler.py         # 端点容器/推理入口
│  │  └─ featurize_online.py
│  └─ monitoring/
│     ├─ build_baseline.py  # Model Monitor 基线
│     └─ quality_job.py
├─ app/dashboard.py         # Streamlit
├─ tests/                   # pytest
├─ .github/workflows/       # CI/CD
├─ requirements.txt
├─ model_card.md
└─ README.md
```

---

4. 实施路线图（总工期 \~ 6–7 天，单人）

时间按"净工作时长"估算；可并行则标注 P（Parallel）。

里程碑	名称	目标	预估	关键产出
M0	基础设施	最小化 AWS 资源可用	6–8h	S3/Glue/Role/Lambda/SM/ECR/Alarms
M1	采集与入湖	周期性抓取 & 校验 & 入湖	5–6h	Raw 分区数据 + GE/Pandera 报告
M2	特征与训练	可复现实验 + 首版模型	8–10h	Features 数据集、MLflow 记录、模型卡
M3	注册与部署	Staging 端点上线	4–6h	SM Endpoint(staging)、CI/CD（stg）

里程碑	名称	目标	预估	关键产出
M4	监控闭环	数据/模型/系统监控就绪	6–8h	Model Monitor、告警、质量回写
M5	可视化	业务仪表盘	4–6h	Streamlit 地图/Top-N/健康度
M6	Prod 准入	指标达标后切 Prod	3–4h	流量切换、Runbook 完成

---

5. 详细分解步骤（任务、时间、验收、文档）

Step 0｜准备与约束（2–3h）

任务

- 创建 GitHub 仓库、分支保护规则；
- 准备 AWS 账户与最小权限角色；
- GitHub OIDC 到 AWS（免长秘钥，Terraform 模块）；
- 初始化 Python 环境与依赖锁定（`pip-tools` 或 `poetry`）。

完成指标

- `aws sts get-caller-identity` 正常；GitHub Actions 能 Assume Role；`pip-compile` 固定依赖成功。

文档

- `docs/security_compliance.md`：账号与权限；
- `README.md`：快速开始。

Step 1｜基础设施 IaC（6–8h）

任务

- Terraform：S3（raw/curated/features/inference/monitoring）、Glue DB/CT、IAM 角色、ECR；
- EventBridge + Lambda（空函数占位）；
- SageMaker 基础 Execution Role；
- CloudWatch 指标/初始告警（Lambda 错误、SM Endpoint 5xx）。

完成指标

- `terraform apply` 无漂移；资源创建完整；
- CloudWatch 中可见自定义指标命名空间；
- S3 已按分区命名规范创建路径（含清单）。

文档

- `docs/architecture.md`：组件与数据流；
- `docs/ops_sla.md`：初始 SLO/告警阈值。

Step 2｜数据采集与入湖（5–6h）

任务

- `ingest/gbfs_ingest.py`：拉取 `station_status` & `station_information`；
- `ingest/weather_ingest.py`：Meteostat（与站点经纬度匹配最近站或城市中心）；
- Pandera/Great Expectations：字段类型/取值范围/缺失率校验；
- Lambda 定时（5 分钟）+ 重试/幂等；
- 数据落地 S3：`raw/city=.../dt=YYYY-MM-DD-HH-mm/`；Glue 分区。

完成指标

- 原始数据每 5 分钟入湖；校验失败的批次被拒绝且写入错误日志；
- Athena 能查询近 2 小时数据，延迟 ≤ 3 分钟。

文档

- `docs/data_contract.md`：字段、约束、错误处理；
- `docs/monitoring_runbook.md`：采集失败排障流程。

Step 3｜离线特征与标签（4–5h）

任务

- `features/build_features.py`：
- 关联 status+info+weather；
- 构造滚动窗口统计、邻域聚合、时间/节假日特征；
- 生成 `t+30` 标签（缺车/缺桩），保留回归目标；
- 产出至 `features/` 分区；
- `features/schema.py`：特征级 schema 校验；
- 产出特征重要性 EDA 报告（简版）。

完成指标

- 单次构建 7–14 天样本；特征缺失率 < 1%；
- `features/` 可被训练脚本直接消费；
- EDA 报告包含分布、相关性、数据泄漏检查（时间泄漏）。

文档

- `docs/feature_store.md`：特征定义、窗口、线上/离线对齐策略；
- `docs/training_eval.md`（起稿）。

Step 4｜训练、评估与追踪（8–10h）

任务

- `training/train.py`：XGBoost/LightGBM 二分类；时间顺序划分（时间外推验证）；
- `training/eval.py`：阈值选择（最大化 Fβ 或 PR-AUC）；
- 开启 MLflow/SageMaker Experiments Autolog；
- 输出模型工件、指标、混淆矩阵、特征重要性；
- 生成 `model_card.md`（自动填充关键指标与假设）。

完成指标

- 首版模型 PR-AUC ≥ 0.70；过拟合检查合格（训练-验证差异 < 0.1）；
- MLflow/SM Experiments 可回溯实验；
- `model_card.md` 完整（场景、假设、风险、公平性）。

文档

- `model_card.md`、`docs/training_eval.md`（完善）。

Step 5A｜（路径A）SageMaker Model Registry & Pipelines（4–6h）

任务

- `pipelines/training_pipeline.py` 编排：特征→训练→评估→条件步→注册→部署到 **staging**；
- 注册模型（版本化、标签、审批状态）；
- 端点部署：实例选择（`ml.m5.large` 起步）；
- 生成 `register_and_deploy.py`（CLI 推广）。

完成指标

- 触发 Pipeline 成功，`staging` 端点可用；
- Registry 中出现新版本，带审阅备注；
- 端点 P95 时延 ≤ 200ms，错误率 < 1%。

文档

- `docs/cicd.md`：Pipeline、环境定义（dev/stg/prod）。

或

Step 5B｜（路径B）MLflow Registry + mlflow.sagemaker.deploy（4–6h）

任务

- MLflow 注册模型 & 阶段标签（Staging/Production）；
- `mlflow.sagemaker.deploy()` 发布到 SageMaker Endpoint；
- 编写 Promote 脚本（按 model version 或 stage 切换）。

完成指标

- MLflow UI 可见最新版本；
- Staging 端点调用成功，性能达标。

文档

- 同 Step 5A 的 `docs/cicd.md`，注明差异点。

Step 6｜CI/CD（3–4h）

任务

- GitHub Actions：
- `ci.yml`：lint（ruff/black）、`pytest`、安全扫描；
- `cd_staging.yml`：构建推送 ECR 镜像（如需）→ 运行 Terraform → 触发 Pipeline → 部署 `staging`；
- `promote_prod.yml`：手动/门禁触发，切换到 `prod`；
- 缓存与并行优化，失败回滚策略。

完成指标

- Push 到 `main` 自动部署到 `staging`；
- 通过 `workflow_dispatch` 能推广到 `prod`；
- 所有工作流在 PR 上必须通过（分支保护）。

文档

- `docs/cicd.md`：触发条件、Secrets、回滚流程。

Step 7｜在线推理与闭环回写（4–5h）

任务

- `inference/featurize_online.py`：按站点实时拼接在线特征（最新 status + 天气 + 静态信息缓存）；

- `inference/handler.py`：批量调用端点，写 `inference/` 分区；
- 延迟 30 分钟构造真实标签 `actuals`，与预测 join 写 `monitoring/quality/`；
- Athena 视图用于回溯评估。

完成指标

- 每 5–10 分钟推理一次，批次成功率 ≥ 99%；
- 质量回写完成且可被查询；
- 端点吞吐与时延达标。

文档

- `docs/monitoring_runbook.md`：闭环数据字典、回填与重算策略。

Step 8｜监控与告警（6–8h）

任务

- `monitoring/build_baseline.py`：基线统计；
- **SageMaker Model Monitor**：数据质量/数据漂移/模型质量作业（小时级）；
- CloudWatch 自定义指标：PR-AUC\@24h、F1\@24h、阈值触发率、端点 P50/P95；
- 告警：漂移阈值、数据延迟、失败率、时延阈值；SNS（邮件/Slack）。

完成指标

- 漂移/质量/系统告警均能触发并路由；
- 监控仪表盘展示近 24–72 小时趋势；
- MTTA ≤ 10 分钟（演练）。

文档

- `docs/monitoring_runbook.md`：告警解释、排障步骤、旁路策略；
- `docs/ops_sla.md`：SLO 与错误预算。

Step 9｜业务仪表盘（4–6h）

任务

- Streamlit 页面：
- **城市地图**（pydeck/folium）：按 `max(P缺车,P缺桩)` 热度着色，站点侧栏信息 + 2 小时预测曲线；
- **Top-N 风险站点**：筛选（区域/半径），显示建议调度量（可选 OR 模块）；
- **模型健康**：PR-AUC/F1/阈值、KS/PSI、Feature Drift；
- **系统健康**：端点时延、错误率、批次成功率；
- **数据新鲜度**：各分区延迟、失败批次。
- 读取 Athena/CloudWatch 指标。

完成指标

- 仪表盘加载 < 3s；
- 所有页面指标无空洞；
- 交互（筛选/选择站点）稳定。

文档

- `docs/architecture.md` 增补可视化与数据源关系。

Step 10｜Prod 准入与移交（3–4h）

任务

- 准入门槛：连续 24–48 小时线上质量指标达标；
- `promote_prod.yml` 触发流量切换，分配加权流量（如需 A/B）；
- 完成移交流程：Runbook 演示、权限最小化审计、成本估算。

完成指标

- `prod` 端点稳定运行 24h；
- 预估月成本与预算在可控范围；
- 所有文档在 `docs/` 下齐全并更新至最新。

文档

- `docs/ops_sla.md` 最终版；
- `README.md` 最终版；
- `CHANGELOG.md`（可选）。

---

6. 交付物清单（Deliverables）

- **代码仓库**（含 IaC、Pipeline、训练/推理/监控、仪表盘、测试）；
- **可运行的 Staging/Prod 端点**；
- **CI/CD 工作流**（CI、CD-Staging、Promote-Prod）；
- **模型卡** 与 **实验记录（MLflow/SM Experiments）**；
- **监控与告警体系**（Model Monitor + CloudWatch + SNS）；
- **Streamlit 仪表盘**；
- **Athena 视图/查询** 用于回溯分析；
- **完整文档集**（见下）。

---

7. 文档结构（Documents）

```
docs/
├─ README.md                    # 15分钟跑通指南（环境、命令、常见问题）
├─ architecture.md              # 架构图、数据流、组件职责、账号边界
├─ data_contract.md             # 数据字段、校验规则、错误处理、延迟门限
├─ feature_store.md             # 特征定义、窗口、线上/离线一致性
├─ training_eval.md             # 训练方案、评估指标、阈值选择、实验记录指引
├─ model_card.md                # 模型卡（自动填充 + 人工补充）
├─ cicd.md                      # CI/CD、环境、门禁、回滚策略、Secrets
├─ monitoring_runbook.md        # 监控项、告警阈值、排障与演练、旁路策略
├─ security_compliance.md       # 权限最小化、密钥管理（OIDC/Secrets Manager）
├─ ops_sla.md                   # SLO/错误预算/可用性目标/容量规划
├─ optimization_extension.md    # （可选）再平衡优化模块（MILP/OR-Tools）
└─ cost_estimate.md             # （可选）成本估算与优化建议
```

---

8. 完成标准与度量（汇总）

数据层

- ✅5 分钟采集一次，成功率 ≥ 99%；
- ✅GE/Pandera 每批校验通过率 ≥ 99%，失败有告警；
- ✅入湖延迟 ≤ 3 分钟。

建模层

- ✅PR-AUC ≥ 0.70，F1（少数类）≥ 0.55；
- ✅时间外推验证通过，过拟合差异 < 0.1；
- ✅模型卡完整并签字归档。

服务层

- ✅Staging → Prod 可受控推广；
- ✅端点 P95 ≤ 200ms，错误率 < 1%；
- ✅推理批次成功率 ≥ 99%。

监控层

- ✅Model Monitor 按小时运行，漂移阈值与告警生效；
- ✅质量回写与 24h 指标可查询；
- ✅MTTA ≤ 10 分钟（演练）。

可视化

- ✅地图/Top-N/健康度/数据新鲜度 4 页完整；
- ✅页面加载 < 3s；交互稳定。

工程化

- ✅全仓库 `pytest` 通过；覆盖率 ≥ 70%；
- ✅CI 在 PR 必过，CD 自动发至 `staging`，手动/门禁发 `prod`；
- ✅Terraform 无漂移，最小权限通过审计。

---

9. 可选扩展：再平衡优化模块（与你优化背景对齐）

- **目标**：在运力、时间窗、路线约束下，最小化缺车/缺桩风险 + 调度成本；
- **方法**：MILP（Gurobi/OR-Tools），每 30–60 分钟滚动求解；
- **接口**：从预测结果生成需求缺口；输出车次、提放量、线路；
- **部署**：Lambda Batch 或 SageMaker Batch Transform；
- **可视化**：仪表盘叠加推荐路线与收益对比。

---

10. 快速开始（命令速查）

```
# 一次性：IaC
cd infra/terraform && terraform init && terraform apply -var="project=mlops-
bikeshare" -var="region=us-east-1"

# 构建特征与本地训练（首版）
python src/features/build_features.py --city nyc --horizon 30
python src/training/train.py --city nyc --algo xgboost

# MLflow 注册 + 部署（路径B 示例）
export MLFLOW_TRACKING_URI="sqlite:///mlflow.db"
mlflow models register -m runs:/<run_id>/model -n bikeshare_risk
python pipelines/register_and_deploy.py --model-name bikeshare_risk --stage
Staging

# Model Monitor 基线
python src/monitoring/build_baseline.py --baseline-s3 s3://.../features/dt=...

# 仪表盘
streamlit run app/dashboard.py
```

---

11. 风险与缓解

- **API 稳定性**：多城市备用源 + 失败重试 + 熔断后回放；
- **时间泄漏**：严格基于可用时点构造特征与标签；
- **类不平衡**：分层抽样/权重/阈值调优 + PR 曲线度量；
- **成本超支**：实例选型及闲时自动扩缩容；
- **权限与合规**：OIDC、最小权限、日志脱敏。

---

12. 成本与容量（粗估）

- Dev/Staging：SageMaker `ml.m5.large` 实例（按需）+ Lambda + S3/Glue/Athena（低）；
- Prod：端点按小时计费，QPS 低时可用 `serverless inference` / `multi-model` 节省成本；
- 预算与上限在 `cost_estimate.md` 提供计算表与优化建议。

附录 A：报警阈值建议

- 数据延迟 > 5 分钟（Critical），> 3 分钟（Warning）；
- 端点 P95 > 300ms（Critical），> 200ms（Warning）；
- 漂移 PSI > 0.2（Warning），> 0.3（Critical）；
- 推理失败率 > 1%（Critical）。

附录 B：测试清单（部分）

- `tests/test_schema.py`：字段与范围；
- `tests/test_features.py`：窗口/对齐/无时间泄漏；
- `tests/test_train.py`：训练收敛、AUC 下限；
- `tests/test_inference.py`：端点契约与性能；
- `tests/test_monitoring.py`：基线与异常样本。

本说明书可直接作为项目 `README + docs/` 的蓝本。根据你即将定居法国，可将 `CITY=paris` 切换至 Vélib GBFS；其余流程保持不变。