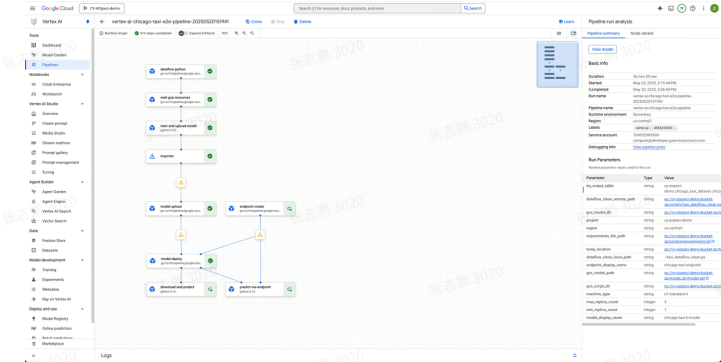
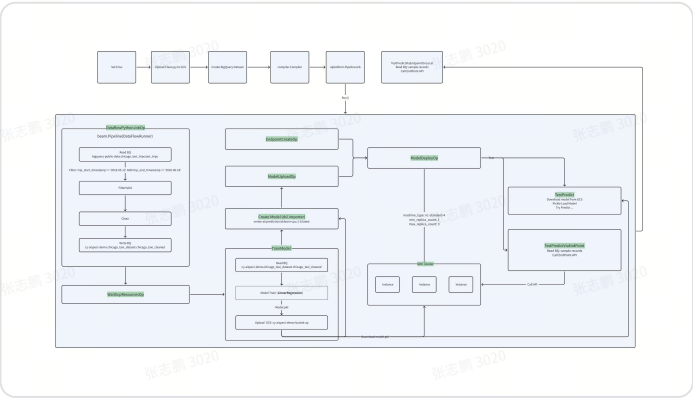


# 芝加哥出租车预测Demo - Vertex AI + Dataflow + BigQuery + GCS

代码仓库地址：  
<https://github.com/ChuanYang-AI/Demo4/tree/main>

## 整体流程图



## 项目配置信息

配置项	示例值
Project Name	CY-AISpeci-demo
Project ID	cy-aispeci-demo
BigQuery 表	cy-aispeci-demo.chicago_taxi_dataset.chicago_taxi_cleaned
GCS Bucket	cy-aispeci-demo-bucket-zp
GCS 脚本目录	gs://cy-aispeci-demo-bucket-zp/scripts/
GCS 模型目录	gs://cy-aispeci-demo-bucket-zp/model_dir
GCS Pipeline 根目录	gs://cy-aispeci-demo-bucket-zp/pipeline_root
Vertex AI Endpoint 名称	chicago-taxi-endpoint

## 1. 业务目标与机器学习解决方案

### 业务目标

本项目基于芝加哥出租车行程数据，构建端到端自动化机器学习流水线，预测每次行程的总费用（`trip_total`），以助力出租车公司优化定价、运营和服务。

### 机器学习用例

- 预测单次出租车行程费用，辅助业务决策
- 支持自动化数据清洗、特征工程、模型训练、部署和在线推理

### 解决方案合规性说明

- 全流程采用 Google Cloud Vertex AI/Kubeflow Pipeline，符合 GCP 机器学习最佳实践
- 数据预处理采用 Dataflow（Apache Beam），数据存储于 BigQuery，训练与推理均在 Vertex AI 上完成
- 代码、数据、模型、API 全流程可追溯、可复现、可交付

### 交付证明

The screenshot displays the Google Cloud Vertex AI console interface. The left sidebar shows the navigation menu with options like Overview, Create prompt, Media Studio, Stream realtime, Prompt gallery, Prompt management, Tuning, Agent Builder, Agent Garden, Agent Engine, Vertex AI Search, Vector Search, Data, Feature Store, Datasets, Model development, Training, Experiments, Metadata, Ray on Vertex AI, Deploy and use, Model Registry, Online prediction, Batch predictions, Monitoring, and Marketplace. The main content area is titled 'chicago-taxi-ir-model' and shows the 'Deploy to endpoint' button. Below this, a table lists the deployed model with columns: Name, ID, Status, Models, Deployment resource pool, Region, Monitoring, Most recent monitoring job, Most recent alerts, Last updated, API, Labels, and Encryption. The table shows one entry: 'chicago-taxi-ir-model' with ID '53056236471681024', Status 'Active', and Region 'us-central1'. Below the table, the 'Test your model' section shows a JSON request and its corresponding response. The JSON request is: 

```
{  "instances": [    [ 63, 0.01, 1, 28, 28 ],    [ 66, 0.01, 1, 35, 35 ],    [ 74, 0.01, 1, 0, 0 ],    [ 79, 0.01, 1, 0, 0 ],    [ 135, 0.01, 1, 28, 28 ]  ]  }
```

 The response is: 

```
{  "predictions": [    3.44693098835026,    4.833934562375286,    1.88275981948935,    1.91159344884658,    3.46216955252788  ],  "deployModelId": "847821289379841848",  "model": {    "project": "projects/784352985598/locations/us-central1/models/8998854",    "modelDisplayName": "chicago-taxi-ir-model",    "modelId": "chicago-taxi-ir-model"  },  "modelId": "chicago-taxi-ir-model"}
```

## 2. 数据探索

## 数据来源与合规性

- 原始数据集：`bigquery-public-data.chicago_taxi_trips.taxi_trips`（公开数据，合规可用）
- 处理后数据存储于 BigQuery（如：`cy-aispeci-demo.chicago_taxi_dataset.chicago_taxi_cleaned`）

本项目处理的数据时间范围为 2018-05-12 至 2018-06-18。

## 探索方式与工具

- 使用 Bigquery 数据加载并使用 `pandas` 进行初步分析
- 通过 Dataflow 脚本对数据进行清洗和特征生成
- 主要探索内容包括：行程时长、里程、费用、支付方式、社区区域分布等

## 探索影响的决策

- 过滤异常值（如极端时长、费用、里程）
- 仅保留主要支付方式（信用卡、现金）
- 生成时间相关特征（工作日/周末、白天/夜间）

## 代码片段与证明

见 `taxi_dataflow_clean.py` 的 `filter_and_clean` 和 `clean_row` 函数。

## 3. 特征工程

### 特征处理内容

- 过滤无效/异常数据（如时长过短/过长、费用异常等）
- 生成新特征：
  - `trip_hours`：行程时长（小时）
  - `trip_speed`：平均速度
  - `payment_type`：支付方式编码（0=信用卡，1=现金）
  - `dayofweek`：是否工作日
  - `hour`：是否白天

### 特征选择理由

- 选用与费用强相关的特征（时长、里程、支付方式、区域等）

- 时间特征有助于捕捉高峰/低谷时段的价格变化

## 代码片段与证明

见 `taxi_dataflow_clean.py` 的 `clean_row` 函数。

## 4. 数据预处理与数据管道

### 数据预处理流程

- 使用 Dataflow (Apache Beam) 从 BigQuery 读取原始数据
- 进行数据清洗、特征工程处理
- 结果写回 BigQuery 新表，供后续训练使用

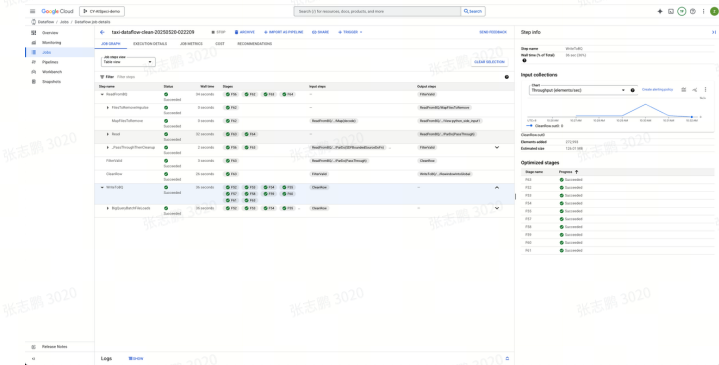
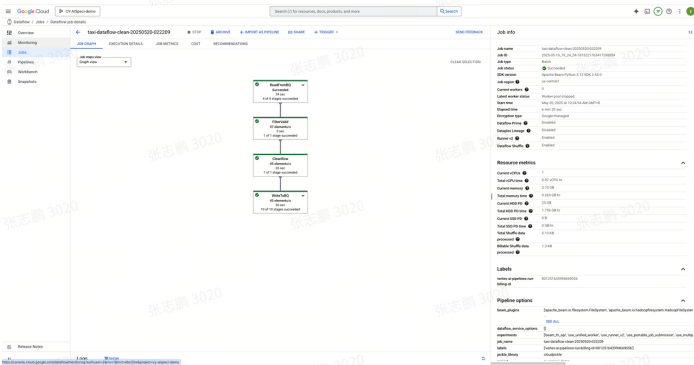
### 可调用 API 说明

- 数据清洗脚本 `taxi_dataflow_clean.py` 可通过 Dataflow Runner 以参数化方式运行
- 在 Vertex AI Pipeline 中通过 `DataflowPythonJobOp` 创建并运行 Dataflow 清洗任务流。

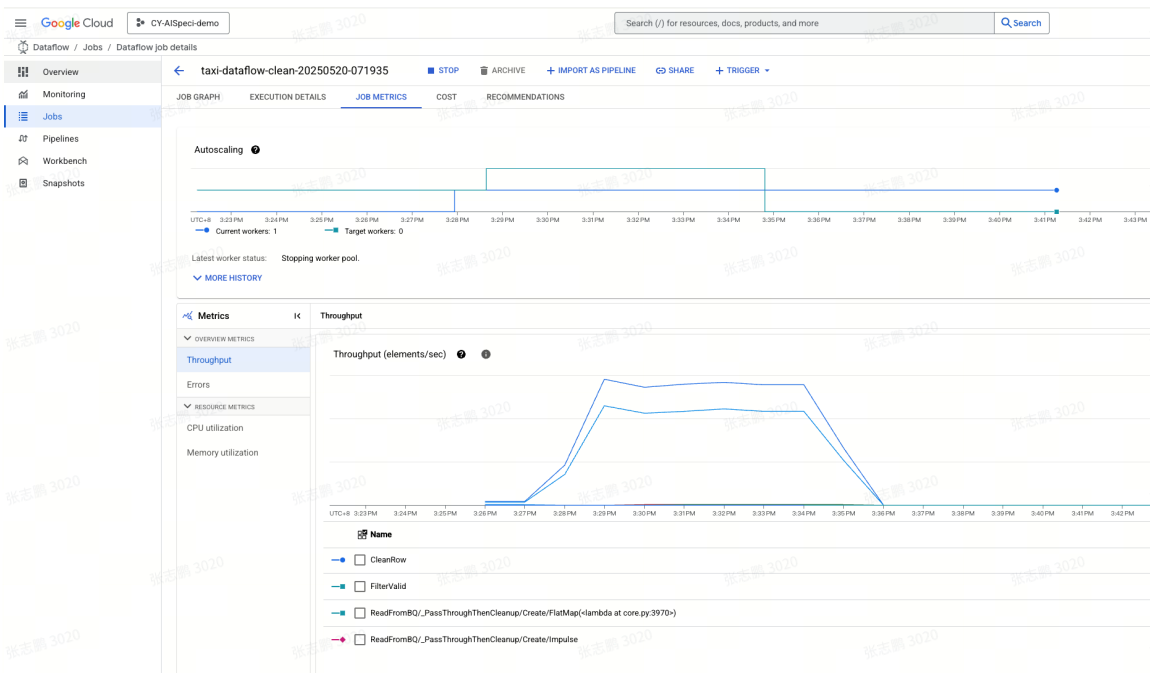
### 合规性与交付证明

- 代码片段见 `taxi_vertex_pipeline.py` pipeline 定义

### Dataflow 流程



### 性能监控



## Bigquery

Bigquery 查询清洗后的数据前 100 条记录, 以及总数.

trip_id	pickup_datetime	dropoff_datetime	pickup_longitude	pickup_latitude	dropoff_longitude	dropoff_latitude	fare_amount	tip_amount	total_amount
1	2015-01-01 00:00:00	2015-01-01 00:05:00	-87.6302	40.7590	-87.6295	40.7590	10.00	1.00	11.00
2	2015-01-01 00:05:00	2015-01-01 00:10:00	-87.6295	40.7590	-87.6295	40.7590	10.00	1.00	11.00
3	2015-01-01 00:10:00	2015-01-01 00:15:00	-87.6295	40.7590	-87.6295	40.7590	10.00	1.00	11.00
4	2015-01-01 00:15:00	2015-01-01 00:20:00	-87.6295	40.7590	-87.6295	40.7590	10.00	1.00	11.00
5	2015-01-01 00:20:00	2015-01-01 00:25:00	-87.6295	40.7590	-87.6295	40.7590	10.00	1.00	11.00

trip_id	pickup_datetime	dropoff_datetime	pickup_longitude	pickup_latitude	dropoff_longitude	dropoff_latitude	fare_amount	tip_amount	total_amount
1	2015-01-01 00:00:00	2015-01-01 00:05:00	-87.6302	40.7590	-87.6295	40.7590	10.00	1.00	11.00
2	2015-01-01 00:05:00	2015-01-01 00:10:00	-87.6295	40.7590	-87.6295	40.7590	10.00	1.00	11.00
3	2015-01-01 00:10:00	2015-01-01 00:15:00	-87.6295	40.7590	-87.6295	40.7590	10.00	1.00	11.00
4	2015-01-01 00:15:00	2015-01-01 00:20:00	-87.6295	40.7590	-87.6295	40.7590	10.00	1.00	11.00
5	2015-01-01 00:20:00	2015-01-01 00:25:00	-87.6295	40.7590	-87.6295	40.7590	10.00	1.00	11.00

## Google Cloud Storage

模型文件

Object	Size	Type	Created	Storage class	Last modified	Public access	Retention	Object metadata	Access control
models	-	Folder	May 20, 2025, 12:02:00 AM	Standard	May 20, 2025, 12:02:00 AM	Not public	-	Storage class	Storage class
models/model.pkl	1.0 MB	File	May 20, 2025, 12:02:00 AM	Standard	May 20, 2025, 12:02:00 AM	Not public	-	Storage class	Storage class
models/model.json	1.0 KB	File	May 20, 2025, 12:02:00 AM	Standard	May 20, 2025, 12:02:00 AM	Not public	-	Storage class	Storage class

数据清洗代码

Object	Size	Type	Created	Storage class	Last modified	Public access	Retention	Object metadata	Access control
code	-	Folder	May 20, 2025, 12:02:00 AM	Standard	May 20, 2025, 12:02:00 AM	Not public	-	Storage class	Storage class
code/clean.py	1.0 KB	File	May 20, 2025, 12:02:00 AM	Standard	May 20, 2025, 12:02:00 AM	Not public	-	Storage class	Storage class
code/read.py	1.0 KB	File	May 20, 2025, 12:02:00 AM	Standard	May 20, 2025, 12:02:00 AM	Not public	-	Storage class	Storage class

## Bug 修复

Apache beam 有个严重 bug, 创建 `DataflowPythonOP` 任务, 并执行 `beam pipeline` 时因获取不到 JobID 导致失败.

Issue: <https://github.com/apache/beam/issues/35013> PR:

<https://github.com/apache/beam/pull/34952>

## 5. 机器学习模型设计与选择

### 模型选择

- 采用 scikit-learn 的线性回归模型 (LinearRegression)
- 选择理由: 回归问题、特征线性相关性强、易于解释, 便于上线和后续扩展

### 模型选择标准

- 简单有效, 支持 Vertex AI 线上部署, 便于客户理解和二次开发

### 代码片段与证明

见 `taxi_vertex_pipeline.py` 的 `train_and_upload_model` 组件。

## 6. 模型训练与开发

### 训练流程

- 从 BigQuery 读取清洗后的数据
- 划分训练集和验证集 (80/20)
- 训练线性回归模型
- 评估模型性能 (可扩展为 RMSE、MAE 等指标)

### Google Cloud 最佳实践

- 使用 Vertex AI Pipeline 自动化训练与部署, 支持分布式、可监控、可追溯

### 合规性与交付证明

- 训练日志、模型文件、训练参数、pipeline 运行截图
- 代码片段见 `train_and_upload_model` 组件

>	i	2025-05-20 15:44:53.974	workerpool0-0	训练数据 shape: (1823278, 13)
>	i	2025-05-20 15:44:53.974	workerpool0-0	训练数据前5行:
>	i	2025-05-20 15:44:53.974	workerpool0-0	taxi_id ... company
>	i	2025-05-20 15:44:53.974	workerpool0-0	0 cb9492c5c6fd621dc2732b3bb6ade16360b9102b3e7ab8... Blue Diamond
>	i	2025-05-20 15:44:53.974	workerpool0-0	1 85c3369c8899bd8ed86ed34cb9476e05387680743cb9f7... Chicago Carriage Cab Corp
>	i	2025-05-20 15:44:53.974	workerpool0-0	2 f8fa70b6dff94cdd06986f0e4bf6020252284cd5e0b88f... Sun Taxi
>	i	2025-05-20 15:44:53.974	workerpool0-0	3 434fca602baa9fe4190d20b8804ad17ea43f64f8bf16cb... Chicago Carriage Cab Corp
>	i	2025-05-20 15:44:53.974	workerpool0-0	4 f5337a97915bf0266c525517caa320f251c924c8ece954... Taxi Affiliation Service Yellow
>	i	2025-05-20 15:44:53.974	workerpool0-0	{'levelname': 'INFO', 'message': ''}
>	i	2025-05-20 15:44:53.974	workerpool0-0	[5 rows x 13 columns]
>	i	2025-05-20 15:44:53.974	workerpool0-0	X columns: Index(['trip_seconds', 'trip_miles', 'payment_type', 'pickup_community_area',
>	i	2025-05-20 15:44:53.974	workerpool0-0	'dropoff_community_area'],
>	i	2025-05-20 15:44:53.974	workerpool0-0	dtype='object')
>	i	2025-05-20 15:44:53.974	workerpool0-0	X shape: (1823278, 5)
>	i	2025-05-20 15:44:53.974	workerpool0-0	y shape: (1823278,)
>	i	2025-05-20 15:44:53.974	workerpool0-0	y head: 0 8.0
>	i	2025-05-20 15:44:53.974	workerpool0-0	1 10.5
>	i	2025-05-20 15:44:53.974	workerpool0-0	2 7.5
>	i	2025-05-20 15:44:53.974	workerpool0-0	3 3.5
>	i	2025-05-20 15:44:53.974	workerpool0-0	4 4.0
>	i	2025-05-20 15:44:53.974	workerpool0-0	Name: trip_total, dtype: float64
>	i	2025-05-20 15:44:53.974	workerpool0-0	模型系数: [ 0.00576676 1.81439868 -3.48683641 0.06514632 0.01623556]
>	i	2025-05-20 15:44:53.974	workerpool0-0	模型截距: 4.273657288597514
>	i	2025-05-20 15:44:53.974	workerpool0-0	验证集 R2-score: 0.9254, 验证集 RMSE: 4.3862
>	i	2025-05-20 15:44:53.974	workerpool0-0	训练集 R2-score: 0.9114, 训练集 RMSE: 4.8220
>	i	2025-05-20 15:44:53.974	workerpool0-0	本地模型文件大小: 663
>	i	2025-05-20 15:44:53.974	workerpool0-0	准备上传的本地模型路径: /tmp/model.pkl
>	i	2025-05-20 15:44:53.974	workerpool0-0	目标 GCS 路径: gs://cy-aispeci-demo-bucket-zp/model_dir/model.pkl
>	i	2025-05-20 15:44:53.974	workerpool0-0	Uploaded /tmp/model.pkl to gs://cy-aispeci-demo-bucket-zp/model_dir/model.pkl
>	i	2025-05-20 15:44:53.974	workerpool0-0	[KFP Executor 2025-05-20 07:44:53,935 INFO]: Wrote executor output file to /gcs/cy-aispeci-demo-bucket-zp/pipeline

## 7. 模型评估

### 评估方式

- 在独立验证集上评估模型预测效果
- 可扩展为在独立测试集上评估

### 合规性与交付证明

- 见 `train_and_upload_model` 组件中的 `train_test_split` 和模型评估部分

### 模型验证与效果

本项目流水线自动训练的线性回归模型在芝加哥出租车数据集上的表现如下（部分训练日志）：

- 模型系数: [ 0.00576676 1.81439868 -3.48683641 0.06514632 0.01623556]
- 模型截距: 4.273657288597514
- 验证集 R2-score: 0.9254, 验证集 RMSE: 4.3862
- 训练集 R2-score: 0.9114, 训练集 RMSE: 4.8220

- 模型系数**：分别对应特征 `trip_seconds` , `trip_miles` , `payment_type` , `pickup_community_area` , `dropoff_community_area` , 反映每个特征对总费用的影响
- 模型截距**：所有特征为0时的基础预测值。



- **R2-score**: 训练集和验证集均在 0.91~0.92，说明模型拟合效果良好，无明显过拟合或欠拟合。
- **RMSE**: 训练集和验证集的均方根误差接近，模型稳定。

整体来看，模型在验证集和训练集上都表现良好，具备较强的泛化能力，可用于后续的线上推理和业务分析。

## 8. 部署与可调用性

### 部署方式

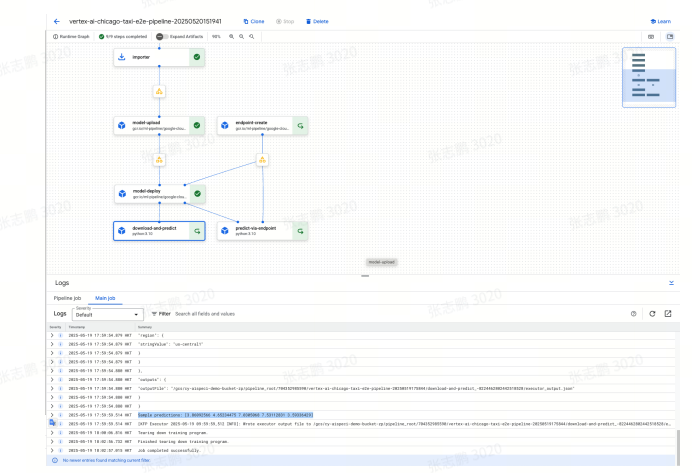
- 训练好的模型自动上传至 GCS
- 通过 Vertex AI ModelUploadOp 上传并部署为 Endpoint
- 自动创建 Endpoint 并完成流量切换
- 相关 GCP 项目、BigQuery、GCS 路径等配置信息详见前文"项目配置信息"表。

### API 调用演示

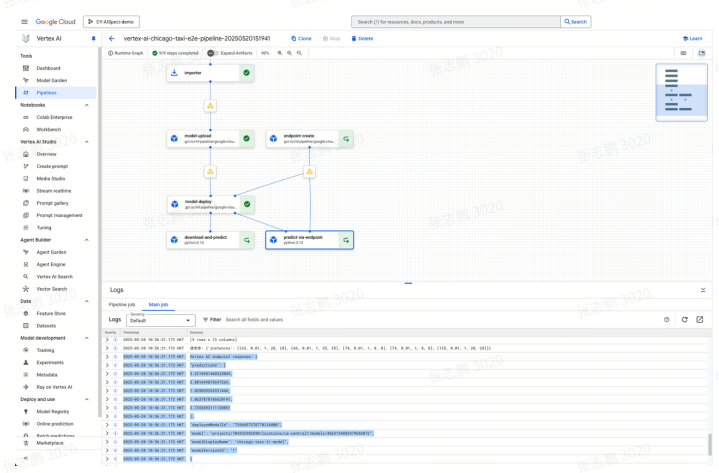
- 支持通过 HTTP API 调用 Vertex AI Endpoint 进行实时推理
- 提供本地和云端两种推理方式

### 合规性与交付证明

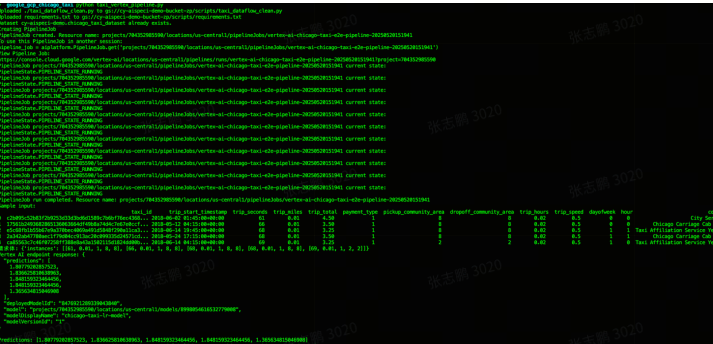
- Endpoint 部署截图、API 调用日志、推理结果



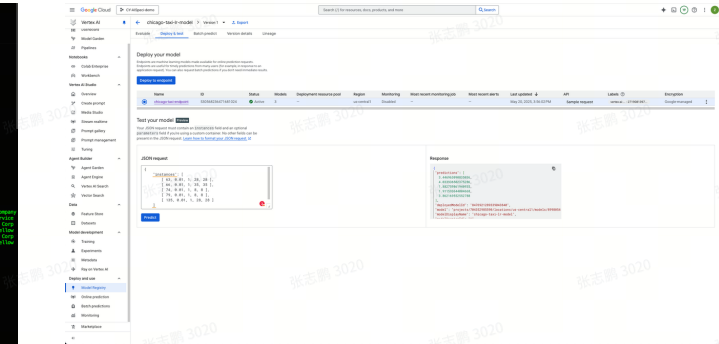
The diagram illustrates a Vertex AI Pipeline workflow. It starts with an 'Ingestion' node, followed by 'model upload' and 'model deploy' nodes. The pipeline concludes with a 'predict on endpoint' node. The logs section below the diagram shows the execution details of the pipeline job.



This screenshot shows the Vertex AI Endpoint configuration page. It displays the endpoint name, location, and the associated model. The logs section at the bottom provides a detailed view of the endpoint's activity, including the request and response data.



The terminal window displays the command used to invoke the endpoint via the curl command. The output is a JSON array containing the model's predictions for the provided input data.



This screenshot shows the Vertex AI Model Registry page. It provides an overview of the model's lifecycle, including its training, deployment, and monitoring status. The 'Deployed' tab is selected, showing the model's current deployment details.



- 代码片段见 `predict_via_endpoint` 组件和 `predict_via_endpoint_local` 函数

## 9. 可编辑性与可扩展性

- 所有代码和参数均可通过环境变量或参数传递，便于客户自定义和二次开发
- 代码结构清晰，便于修改和扩展

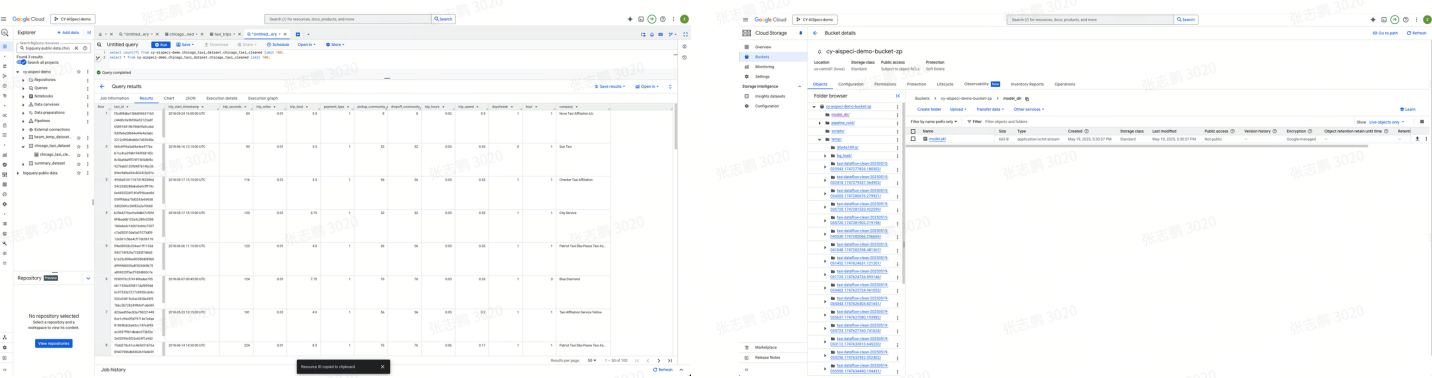
## 交付证明

```
bash
export GOOGLE_CLOUD_PROJECT=cy-aispeci-demo
export PROJECT_ID=cy-aispeci-demo
export REGION=us-central1
export BUCKET_NAME=cy-aispeci-demo-bucket-zp...# 你的 GCS bucket 名称
export TEMP_LOCATION=gs://$BUCKET_NAME/temp
export DATAFLOW_CLEAN_LOCAL_PATH=./taxi_dataflow_clean.py
export GCS_SCRIPT_DIR=gs://$BUCKET_NAME/scripts/
export BQ_OUTPUT_TABLE=cy-aispeci-demo.chicago_taxi_dataset.chicago_taxi_cleaned
export GCS_MODEL_DIR=gs://$BUCKET_NAME/model_dir
export GCS_MODEL_PATH=gs://$BUCKET_NAME/model_dir/model.pkl
export PIPELINE_ROOT=gs://$BUCKET_NAME/pipeline_root
export MACHINE_TYPE=n1-standard-4
export MIN_REPLICA_COUNT=1
export MAX_REPLICA_COUNT=3
export ENDPOINT_DISPLAY_NAME=chicago-taxi-endpoint
```

## 10. 数据与模型存储位置（Google Cloud）

- 数据存储、模型存储、Pipeline 根目录等配置信息详见前文"项目配置信息"表。

## 交付证明



## 11. 代码仓库与运行说明

- 代码仓库包含所有源码、说明文档和运行脚本
  - 详细运行说明及环境变量配置见 `README.md`
  - 支持一键本地运行和云端自动化部署
- 

## 12. 代码来源声明

- 本项目所有代码均为原创开发，部分依赖开源库（如 scikit-learn、apache-beam），已严格遵循相关开源协议
- 如有第三方代码，均已注明来源并合规使用

## 交付证明

- 代码原创声明、开源协议遵循说明
- 

## 13. 安全与隐私

- 所有数据均存储于 Google Cloud，权限可控
- 支持数据去标识化、分桶等隐私保护措施
- 仅使用公开数据集，无敏感信息泄露风险

## 合规性与交付证明

- 数据权限配置截图、去标识化处理说明
- 

## 14. 参考与扩展

- 详细参数、环境变量、GCS/BQ 配置等见 `README.md`
  - 可根据实际业务需求扩展特征、模型和自动化流程
-