MLOps

Annie (Yu-Chuan) Chiu
Elena (Wan-Chun) Liao
Youshi Zhang
Yu Hsin Wang

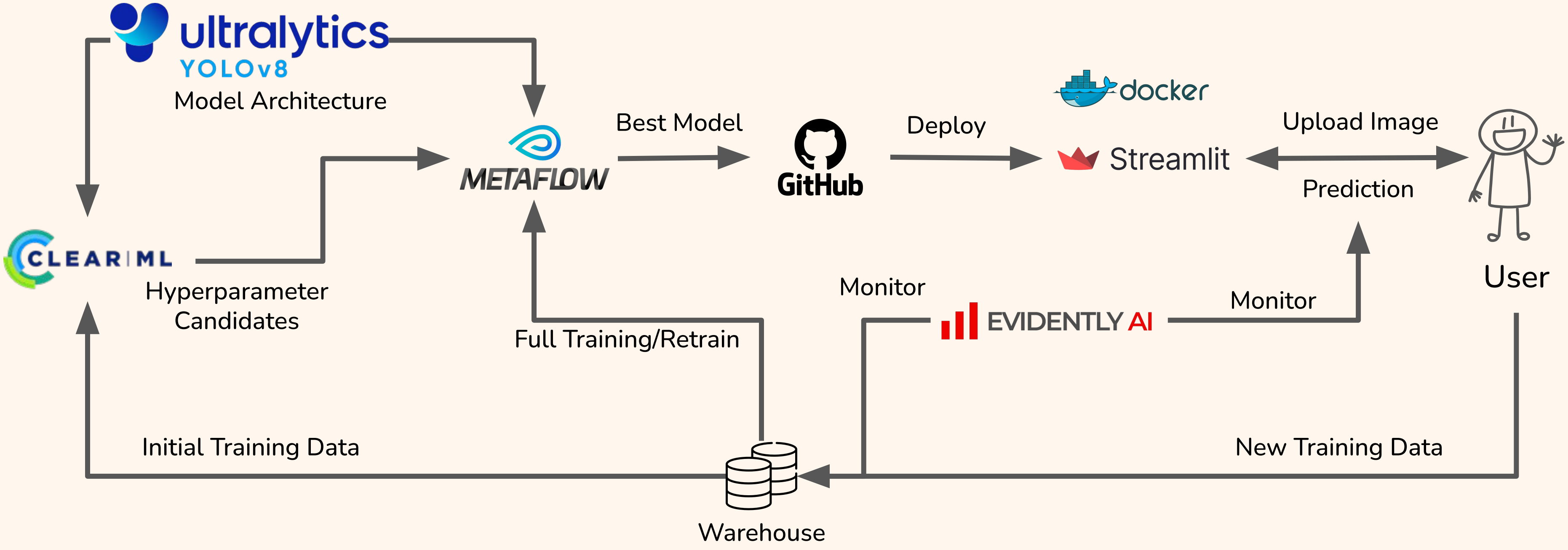
POC Agenda

1. Experiment and Artifact Tracking - Yu Hsin
2. Data Versioning & Data Quality - Elena
3. ML Pipeline Orchestration - Youshi
4. Model Deployment and App - Annie
5. Model Monitoring - Yu Hsin
6. CI/ CD - Elena
7. Architecture Structure

Overview - Food Image Detection System

- Goal: Detect food item from user uploaded image
- Dataset: ~3GB of image and annotation json files
- Model family: Yolo, an object detection computer vision model
- Other considerations
    - Start-up: prefer open source and free tier
    - Data scientists background

Architecture Structure



Development

Deployment

Interface

Experiment and Artifact Tracking POC

ClearML vs Comet

Yu Hsin & Annie

Presentor: Yu Hsin

ClearML vs Comet

	ClearML	Comet
Costs	**Open-source** and completely free to use 👍	Offers a free tier with limited features, but users need to **pay for advanced features** such as collaboration tools and data storage
Infrastructure	Supports cloud-based infrastructure such as AWS and GCP but **Limited support for Kubernetes**	Supports cloud-based infrastructure such as AWS and GCP and has a **Kubernetes integration** for deployment
Skills needed	User-friendly interface and comprehensive documentation	User-friendly interface and free community Slack support, but **requires additional setup and configuration** for certain use cases
Stability	Open source tool, **transparent development roadmap**	Commercial tool, may be **susceptible to changes in pricing**

Final Decision

ClearML

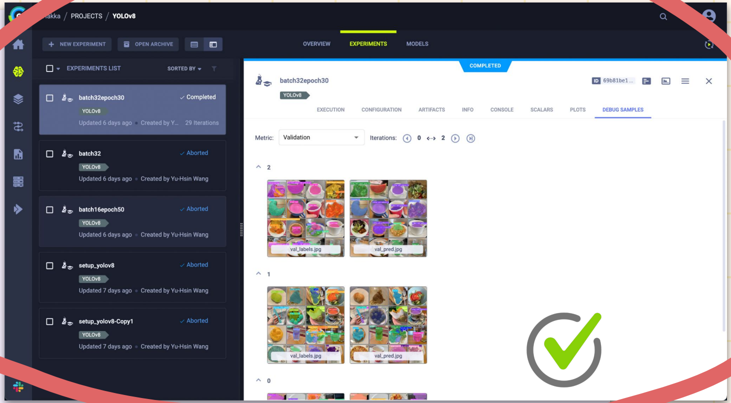
Based on our requirements, ClearML may be a better option for us for following reasons:

1. **YOLOv8 integrates seamlessly with ClearML**

   Install ClearML → init →Done 😎

2. **Collaboration**

   With ClearML, up to 3 users can use a workspace for free, while Comet's community plan does not include collaboration features

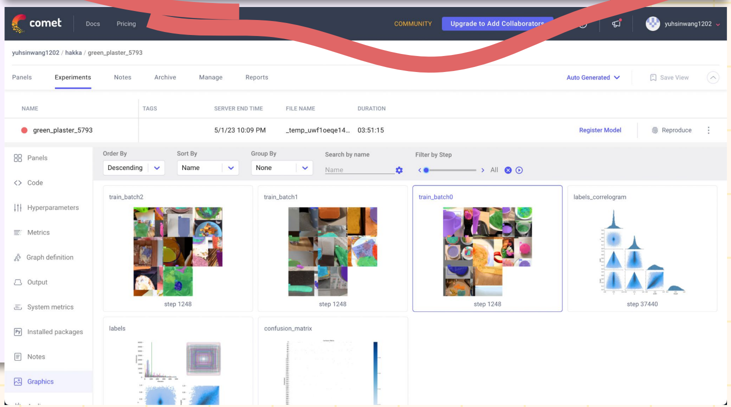


akka / PROJECTS / YOLOv8

- batch32epoch30 ✓ Completed · YOLOv8 · Updated 6 days ago · Created by Y... 29 Iterations
- batch32 ✓ Aborted · YOLOv8 · Updated 6 days ago · Created by Yu-Hsin Wang
- batch16epoch50 ✓ Aborted · YOLOv8 · Updated 6 days ago · Created by Yu-Hsin Wang
- setup_yolov8 ✓ Aborted · YOLOv8 · Updated 7 days ago · Created by Yu-Hsin Wang
- setup_yolov8-Copy1 ✓ Aborted · YOLOv8 · Updated 7 days ago · Created by Yu-Hsin Wang

batch32epoch30 · YOLOv8 · ID 69b81be1...

Metric: Validation · Iterations: 0 ↔ 2



yuhsinwang1202 / hakka / green_plaster_5793

NAME	TAGS	SERVER END TIME	FILE NAME	DURATION
green_plaster_5793		5/1/23 10:09 PM	_temp_uwf1oeqe14...	03:51:15

Order By: Descending · Sort By: Name · Group By: None · Search by name: Name · Filter by Step: All

Data Quality and Versioning POC

DVC/DagsHub vs Git LFS

Elena & Youshi

Presentor: Elena

DVC vs Git LFS

	DVC/DagsHub	Git LFS
Learning Curve	Requires additional learning Need to maintain both GitHub and DagsHub	Readily integrate with Git/GitHub
Storage and Cost	Free Tier: 10 GB Well integrated with S3	Free Tier: 2GB Can setup external S3
Pipeline Versioning	Allow pipeline versioning 👍	Doesn’t allow
File Structure	Can manage at directory level 👍	Can manage at directory level 👍

Final Decision

DagsHub/DVC

1. Our training data is **more than 2GB** but unlikely to go beyond 10GB
2. Some **preprocessing** is required to transform json files to yolo format

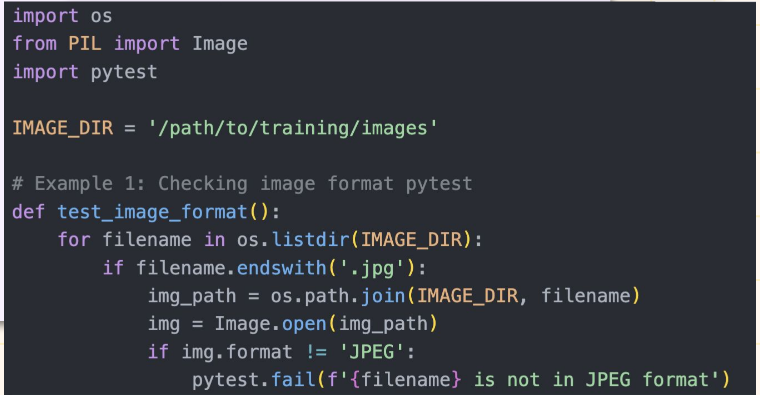
   Need to version control the pipeline as well
3. As an image classifier project, we may make use of DagsHub's image studio feature

Data Quality

Customized Pytest

1. Our training data contains: **image + json file -> yolo format**
2. While **Deepchecks** can examine image data, **json and yolo file is not supported** by any popular data quality tools
3. So we go with **customized pytest** script that checks
   a. Image format
   b. Image size
   c. Pixel variance
   d. Annotation format

Eg. Pytest to check Image format



```python
import os
from PIL import Image
import pytest

IMAGE_DIR = '/path/to/training/images'

# Example 1: Checking image format pytest
def test_image_format():
    for filename in os.listdir(IMAGE_DIR):
        if filename.endswith('.jpg'):
            img_path = os.path.join(IMAGE_DIR, filename)
            img = Image.open(img_path)
            if img.format != 'JPEG':
                pytest.fail(f'{filename} is not in JPEG format')
```

ML Pipeline Orchestration POC

Metaflow vs Airflow

Yu Hsin & Youshi

Presentor: Youshi

Metaflow vs Airflow

	Metaflow	Airflow
Costs	Open-source and completely free to use 👍	Open-source and completely free to use
Infrastructure	Seamless integration with cloud platform	Deployment is possible but requires more works
Learning Curve	Python-based and artifact persistence may not always work. Requires additional learning	Better support for bash operation and working with reading and saving operations
Branching	Supports dynamic branching	Needs work around for Dynamic branching

Final Decision

Metaflow

- Initially: Airflow
  a. Yolo model training more like ETL
  b. CLI command + auto save/load artifacts to/from hard drive
- In the future: move to **Metaflow**
  a. Overall, better supported for **ML specific tasks** especially if we consider **other model families**
  b. Easier and more flexible for **cloud deployment**
  c. Easier for **dynamic branching**; helpful when we have more computing power

Model Deployment and App POC

Streamlit vs FAST API

Elena & Annie

Presentor: Annie

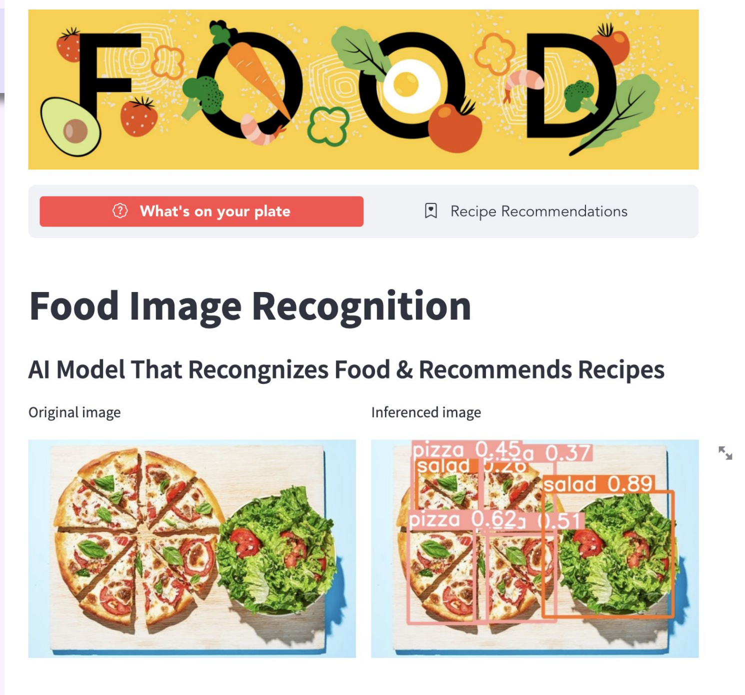
Streamlit vs FAST API

	Streamlit	FAST API
Service Positioning	👍 • Python based web framework that focuses on turning scripts into web apps with interactive UI	• Python based web framework that is designed for building high-performance, fully compatible API quickly
Development Complexity	👍 • Intuitive and easy to learn • Optimized for data science project/ visualization • Continuously releasing advanced features	• Steeper learning curve, such as familiarity with Starlette framework • Required additional frontend skills for developing pleasant looking interface
Scalability	• Primarily for local development and deployment, not for large-scale applications • Streamlit Community Cloud - 1GB per app • Deploy using Docker + Kubernetes	• Built on top on a lightweight Starlette ASGI (Asynchronous Server Gateway Interface) framework • Can handle a large number of request
Security	• No built-in security features • Streamlit Community Cloud ensures product and network security	👍 • Built-in security features, such as automatic request validation and authentication

Final Decision

Streamlit

1. Interactive frontend presentation that allows user to visualize detected images and customize parameter settings
   → Streamlit offers many pre-built widgets
2. Limited time, skills, and resources as a startup
   → Streamlit allows developers to work in a fast, interactive loop, enabling us to build prototypes in a timely fashion and test our product in the market
3. Flexibility to deploy on other cloud services such as AWS and Azure using Docker/ k8s, and to develop in combination with FAST API to enhance scalability and performance



Food Image Recognition

AI Model That Recongnizes Food & Recommends Recipes

Model Monitoring POC

Evidently AI vs Alibi Detect

Yu Hsin & Annie

Presentor: Yu Hsin

Evidently AI vs Alibi Detect

	Evidently AI	Alibi Detect
Costs	Free and open source. Paid plans available for enterprise use	Free and open source
Services	Offers model drift detection, fairness assessment, and feature importance analysis, and a wide range of metrics to monitor and analyze model performance, including accuracy, F1-score	In addition to drift detection, Alibi Detect offers a wide range of outlier detection algorithms, adversarial detection, and root cause analysis
Skills needed	Provides a user-friendly dashboard with a variety of metrics to track model performance; easier for beginners to use	Offers more customization options such as different detection methods, therefore requires more technical expertise to set up and use
Stability	Actively maintained with frequent updates	Actively maintained with frequent updates, but with limited documentation and community support compared to Evidently AI

Final Decision

Evidently AI

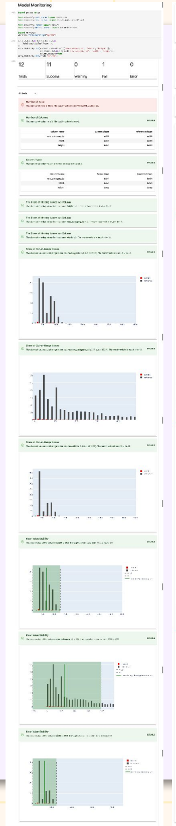
Based on our requirements, **Evidently AI may be a better option** for us for following reasons:

1. **User-Friendly**

   Evidently AI is designed to be easy to use with a minimal need for dashboard maintenance, while Alibi Detect requires more development and maintenance work for dashboards

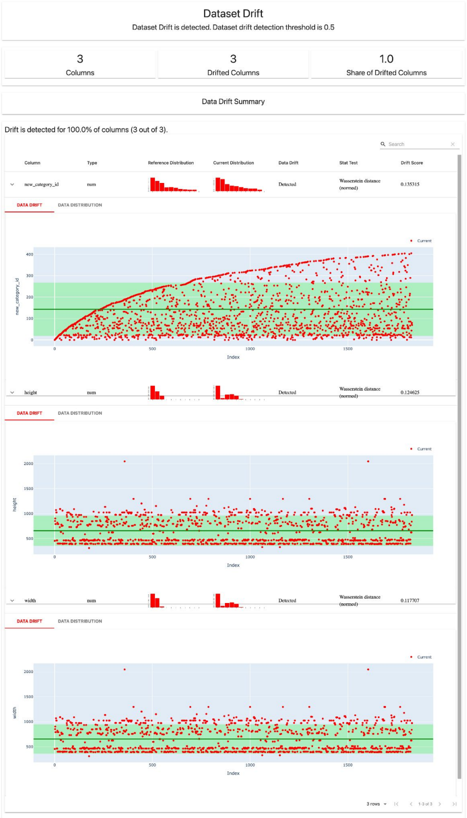
2. **Easy-to-Read and Interpretable Visualizations**

   Evidently AI provides interactive reports in Jupyter notebook and allows users to export them to an HTML file to facilitates model monitoring



Model Monitoring

12	11	0	1	0
Tests	Success	Warning	Fail	Error



Dataset Drift

Dataset Drift is detected. Dataset drift detection threshold is 0.5

3	3	1.0
Columns	Drifted Columns	Share of Drifted Columns

Data Drift Summary

Drift is detected for 100.0% of columns (3 out of 3).

Column	Type	Reference Distribution	Current Distribution	Data Drift	Stat Test	Drift Score
new_category_id	num			Detected	Wasserstein distance (normed)	0.135315
height	num			Detected	Wasserstein distance (normed)	0.124625
width	num			Detected	Wasserstein distance (normed)	0.117707

3 rows 1-3 of 3

CI / CD POC

GitHub Action vs Jenkins

Youshi & Elena

Presentor: Elena

GitHub Action vs Jenkins

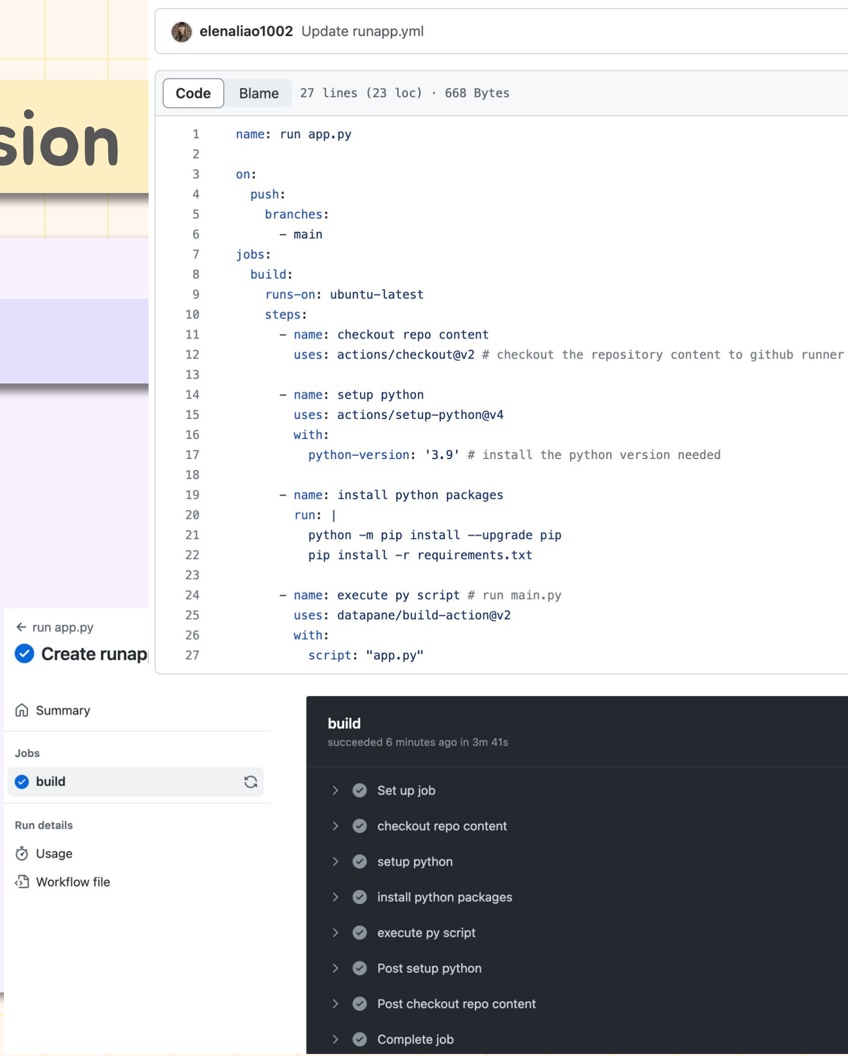
	GitHub Action	Jenkins
Scalability	new and has not yet been tested at the same scale as Jenkins, but it is designed to be scalable and should be able to handle most workflows	scale to support large, complex projects with multiple teams and workflows
security	managed by **secrete setting page**	**requires more manual security configuration** and maintenance
Skills needed	**Easy to set up**, no need for external servers	More complicated setup requiring external servers (plugins/Docker container, setting source code in Jenkins, webhooks in GitHub)
pipeline	Built-in pipeline visualization	Need to see in the execute shell in Jenkins / other setting

Final Decision

GitHub Action

We choose GitHub Actions. Since it is a newer, cloud-based solution that is often favored for its ease of setup and integration with GitHub. It’s easy to set up, no need for external servers.

In comparison, setting up Jenkins requires the installation of plugins, configuring the source code in Jenkins, and setting up webhooks on GitHub.



elenaliao1002 Update runapp.yml

Code Blame 27 lines (23 loc) · 668 Bytes

```yaml
name: run app.py

on:
  push:
    branches:
      - main
jobs:
  build:
    runs-on: ubuntu-latest
    steps:
      - name: checkout repo content
        uses: actions/checkout@v2 # checkout the repository content to github runner

      - name: setup python
        uses: actions/setup-python@v4
        with:
          python-version: '3.9' # install the python version needed

      - name: install python packages
        run: |
          python -m pip install --upgrade pip
          pip install -r requirements.txt

      - name: execute py script # run main.py
        uses: datapane/build-action@v2
        with:
          script: "app.py"
```

← run app.py

Create runap

Summary

Jobs

build

Run details

Usage

Workflow file

build

succeeded 6 minutes ago in 3m 41s

- Set up job
- checkout repo content
- setup python
- install python packages
- execute py script
- Post setup python
- Post checkout repo content
- Complete job

Linting and Code Style

PEP8 & Pylint