MLOps Proof-of-Concept: Building a Machine Learning Pipeline for Hakka

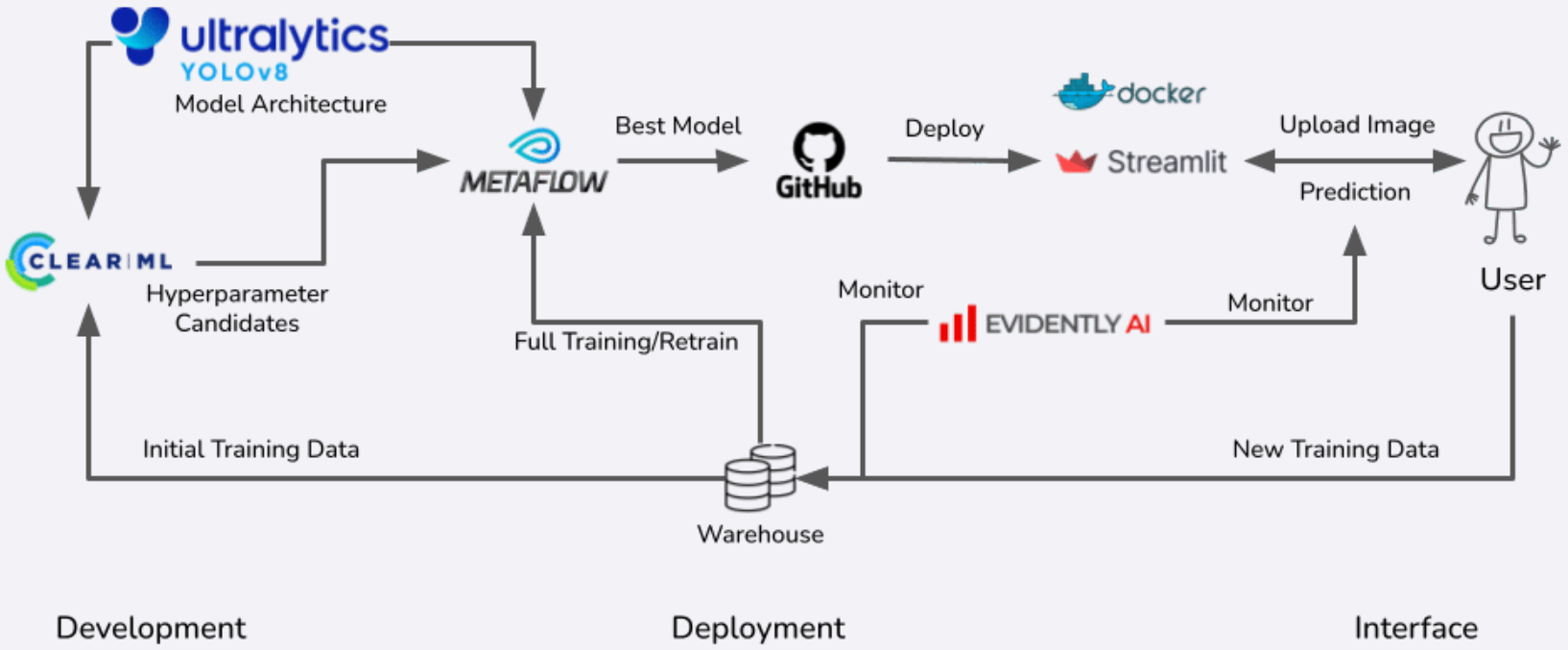
Discover the power of MLOps in the context of Hakka - an innovative food recognition and recipe recommendation app. In this presentation, we'll explore the latest tools and techniques for building an effective machine learning pipeline, and the key components and best practices that enable Hakka to deliver accurate food recognition and personalized recipe recommendations.

Y **by Yuhsin Wang**

Overview - Food Image Detection System

- **Goal:** Detect food items in user-uploaded images
- **Dataset:** ~3GB of images and annotation JSON files
- **Model family:** YOLO, a computer vision model for object detection
- **Other considerations:**
  - Prefer open-source and free tools
  - Team has a background in data science

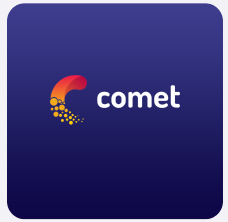
Structure



Experiment Tracking

ClearML

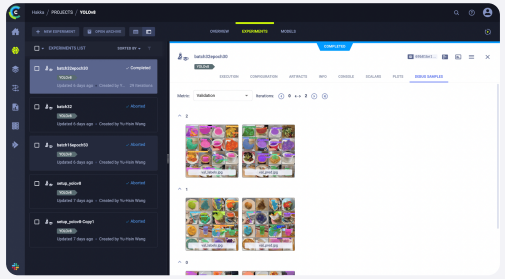
- **Open-source** and completely free to use 👍
- Supports cloud-based infrastructure such as AWS and GCP but **has limited support for Kubernetes**
- User-friendly interface and comprehensive documentation 👍
- Transparent development roadmap **as it is an open-source tool**

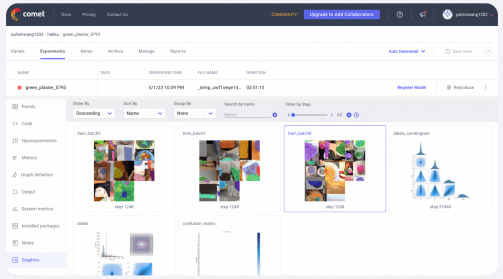


Comet

- Offers a free tier with limited features, but users need to **pay for advanced features** such as collaboration tools and data storage
- Supports cloud-based infrastructure such as AWS and GCP and has a **Kubernetes integration** for deployment 👍
- User-friendly interface and free community Slack support, but **requires additional setup and configuration** for certain use cases
- Comes with a commercial license and may be **susceptible to changes in pricing**

Final Decision: ClearML





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

- **YOLOv8 integrates seamlessly with ClearML**

Install ClearML → init → Done 😎

- **Collaboration**

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

Data Versioning

DVC/DagsHub

- **Requires additional learning** - Need to maintain both GitHub and DagsHub
- **Free Tier: 10 GB** 👍
- Well integrated with S3 👍
- Allows pipeline versioning 👍
- Can manage at directory level 👍

Git LFS

- **Readily integrates with Git/GitHub** 👍
- **Free Tier: 2GB**
- Can set up external S3
- Doesn't allow pipeline versioning
- Can manage at directory level 👍

Final Decision: DagsHub/DVC

- Our training data is **more than 2GB**, but unlikely to go beyond 10GB
- Some **preprocessing** is required to transform JSON files to YOLO format. Need to version control the pipeline as well.
- As an image classifier project, we may make use of DagsHub's Image Studio feature.

Data Quality: Customized Pytest

- Our training data contains: **image + JSON file -> YOLO format**
- While **Deepchecks** can examine image data, **JSON and YOLO files are not supported** by any popular data quality tools
- So we go with a **customized Pytest** script that checks:
  - Image format
  - Image size
  - Pixel variance
  - Annotation 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')
```

Machine Learning Pipeline Orchestration

1 Metaflow

- **Open-source** and completely free to use 👍
- Seamless integration with cloud platforms 👍
- **Python-based** but artifact persistence may not always work. Requires additional learning
- Supports dynamic branching 👍

2 Airflow

- **Open-source** and completely free to use 👍
- Deployment is possible but requires more work
- Better support for Bash operations and working with reading and saving operations 👍
- Needs workarounds 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**

Easier for **dynamic branching**; helpful when we have more computing power

Model Deployment and App

Streamlit

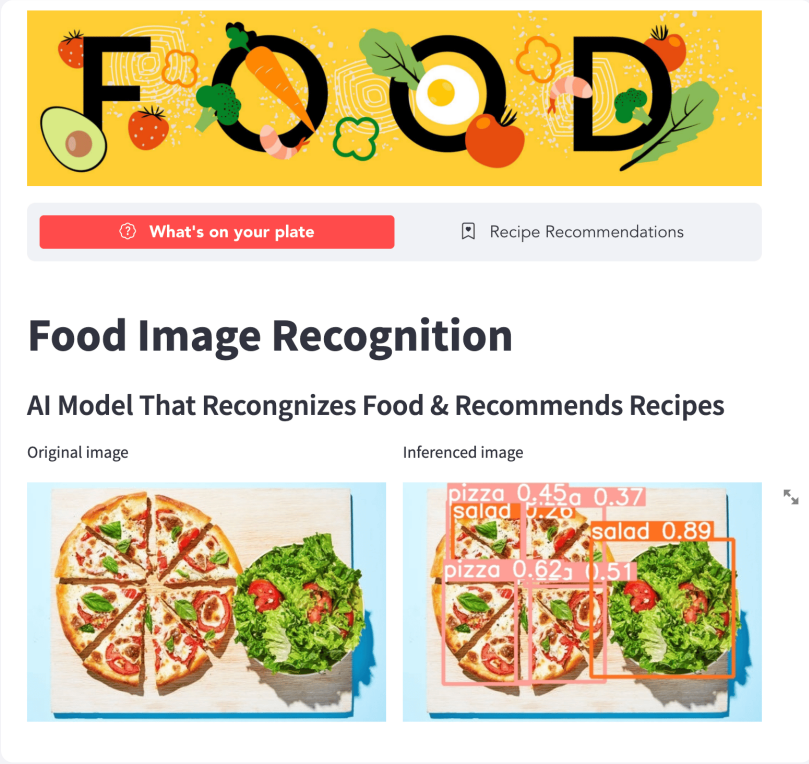
- Python-based web framework that focuses on turning scripts into web apps with an interactive UI 👍
- Intuitive and easy to learn 👍
- Optimized for data science projects/visualization 👍
- Continuously releasing advanced features 👍
- Primarily for local development and deployment, not for large-scale applications
- Streamlit Community Cloud - 1GB per app
- Deploy using Docker + Kubernetes
- No built-in security features
- Streamlit Community Cloud ensures product and network security

FastAPI

- Python-based web framework that is designed for building high-performance, fully compatible APIs quickly
- Steeper learning curve, such as familiarity with Starlette framework
- Requires additional frontend skills for developing a pleasant-looking interface
- Built on top of a lightweight Starlette ASGI (Asynchronous Server Gateway Interface) framework
- Can handle a large number of requests
- Built-in security features, such as automatic request validation and authentication 👍

Final Decision: Streamlit

- Interactive frontend presentation that allows users to visualize detected images and customize parameter settings
  - Streamlit offers many pre-built widgets 👍
- 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 👍
- Flexibility to deploy on other cloud services such as AWS and Azure using Docker/k8s, and to develop in combination with FastAPI to enhance scalability and performance



Model Monitoring

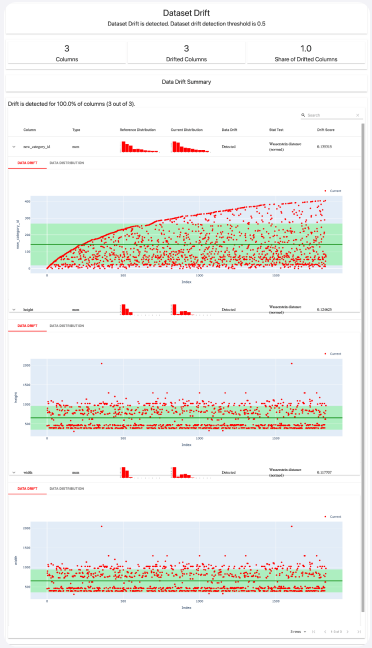
Evidently AI

- **Free and open source.** Paid plans available for enterprise use 👍
- Offers **model drift detection, fairness assessment, and feature importance analysis**, and a wide range of **metrics** to monitor and analyze model performance, including accuracy and F1-score.
- Provides a **user-friendly dashboard** with a variety of metrics to track model performance, making it **easier for beginners to use** 👍
- **Actively maintained** with frequent updates 👍

Alibi Detect

- **Free and open source** 👍
- In addition to **drift detection**, Alibi Detect offers a wide range of **outlier detection** algorithms, **adversarial detection**, and **root cause analysis** 👍
- Offers more customization options such as different detection methods, therefore **requires more technical expertise** to set up and use.
- 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 the following reasons:

- **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.
- **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 facilitate model monitoring.

CI/CD Pipeline

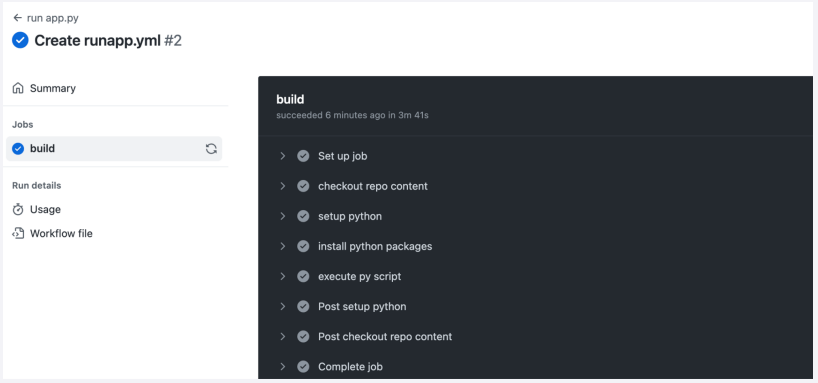
Jenkins

- Scales to support large, complex projects with multiple teams and workflows 👍
- **Requires manual security configuration and ongoing maintenance**
- Involves a more complex setup, including the use of external servers (plugins/Docker container), configuring source code in Jenkins, and setting up webhooks in GitHub
- Execution details can be viewed in the Jenkins "execute shell" or other relevant settings.

GitHub Actions

- Although relatively new and not tested at the same scale as Jenkins, GitHub Actions is designed to be scalable and capable of handling most workflows
- Secrets are managed through a dedicated setting page, ensuring **secure storage and access** 👍
- **Setting up GitHub Actions is straightforward** and does not require external servers 👍
- GitHub Actions provides a built-in pipeline visualization feature, making it easier to understand and track the workflow process 👍

Final Decision: GitHub Action



We chose GitHub Actions due to its **newer cloud-based solution**, which is often favored for its ease of setup and seamless integration with GitHub. With GitHub Actions, there is no need for external servers, making the setup process straightforward.

In comparison, setting up Jenkins requires additional steps such as installing plugins, configuring source code within Jenkins, and setting up webhooks on GitHub.

Streamline Machine Learning Operations

Effective MLOps is critical for unlocking the full potential of models and driving impactful results. By implementing an experiment tracking system, ensuring data quality and versioning, monitoring models, and deploying them using a CI/CD pipeline, we can streamline our machine learning operations and create value.