TAI Frameworks - Use Case Infrastructure

Peter Ainsworth, Nicholas Clark, Daniel Weldon, James Sweet, Charles Vardeman, and Paul Brenner
The University of Notre Dame and Crane Naval Surface Warfare Center

Challenge and Opportunity

Trust is a critical aspect of developing and deploying machine learning (ML) systems, particularly in high-stakes environments such as Naval and other Department of Defense (DoD) applications. This project aims to identify and develop both a "Conceptual Framework" and a set of tools that enable best practices for building Trusted AI Pipelines.

Frameworks Toolkit

Software Development Tools

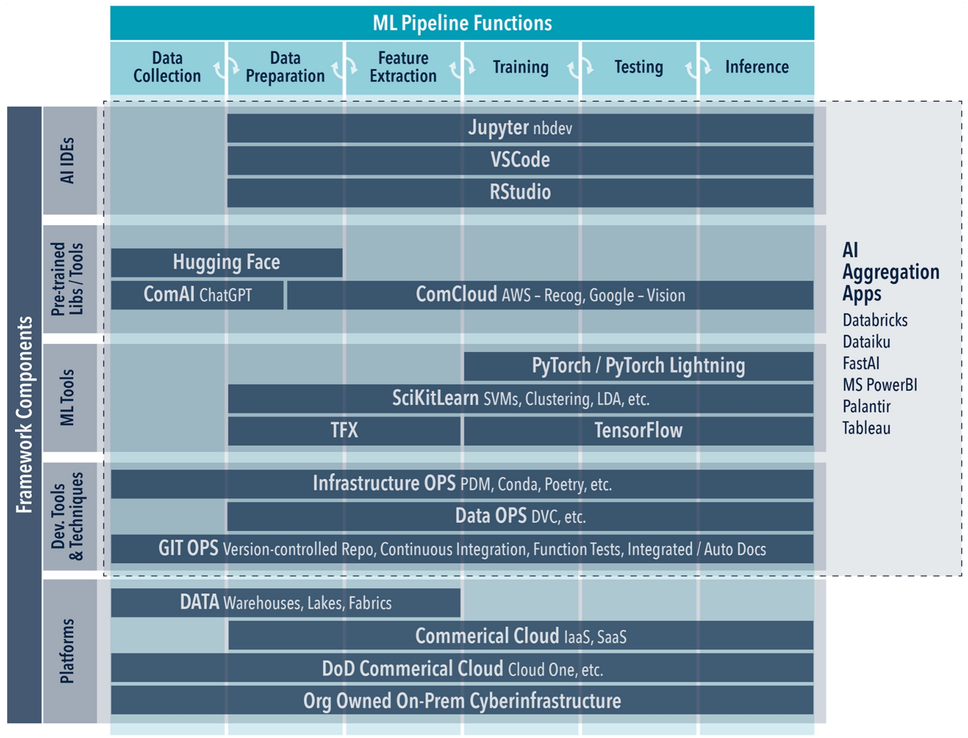
The frameworks' methodology emphasizes the establishment of trust using traditional software development tools, including **GitOps, Virtual Environments, Jupyter,** and **VSCode.** These tools set the foundation for any trusted AI pipeline.

Adding Trust Specific Tools

To emphasize explainability, accountability, and robustness in pipelines, **Docker, PDM, DVC,** and **nbdev** were added to the frameworks toolkit.

1. *Docker* - provides containerization for ML applications, enabling standardized environments that ensure safety, consistency, and deployability
2. *PDM* - streamlines Python dependency management, with accurate package versioning and dependency resolution
3. *Data Version Control* - facilitates data versioning, enhancing data integrity and reproducibility throughout the ML pipeline
4. *nbdev* - simplifies development and collaboration with literate programming in Jupyter notebooks, fostering explainability and transparency in ML models

Frameworks Component Mapping



Securing the AI Pipeline

Example Project: Use NLP on FAA incident report data to predict incident causes, incorporating all stages of the TAI pipeline from data collection to model inference.

Frameworks Approach:

- Manage all dependencies and versions with PDM including PyTorch and scikit-learn
- Load and track raw data with DVC from a common mount point
- Split pipeline into Jupyter notebooks by stage
- Convert notebooks into pipeline documentation using nbdev and publicly host with GitHub pages
- Track inputs and outputs for each stage using DVC and GitOps
- Build docker image with appropriate dependencies for reproducible distribution

This process ensures 1) **reproducibility** through extensive dependency management and packaging, 2) **accountability** through stage-by-stage data tracking and pipeline management, and 3) **explainability** through auto-generated documentation and a traceable file structure.

Managing Experimentation

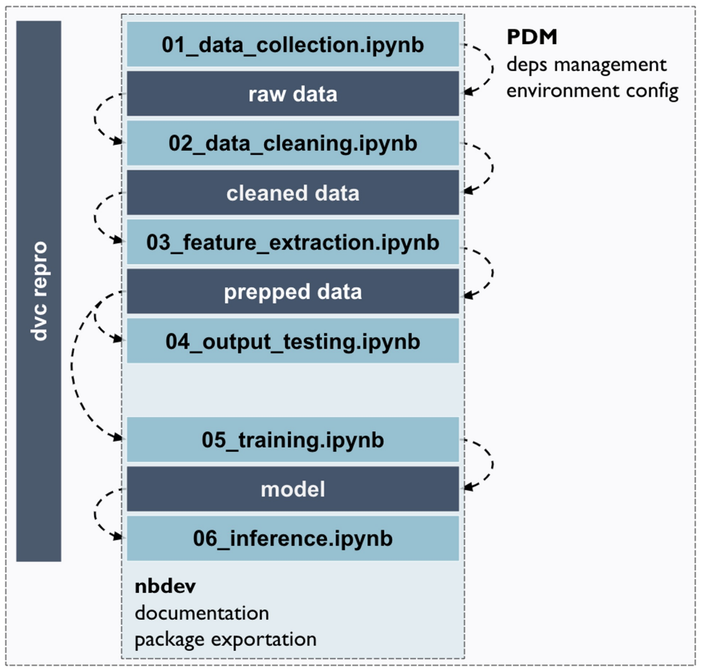
Example Project: Utilizing innovative salience metrics to expand the quality and scope of explainability in computer vision systems.

Frameworks Approach:

- Build off pipeline foundational tools
- Parameterize notebooks using DVC and track output images
- Integrate DVC experiments with VSCode for quick **reproducibility**

This process enhances 1) **reproducibility** through experimentation management and 2) **accountability** through input and output tracking.

Pipeline Structure



Acknowledgements

We extend our gratitude to Dr. Chris Sweet, Priscilla Moreira, Lindsey Michie, Chandler Cunningham, Dr. Kristina Davis, Dr. Adam Shull of NSWC Crane, Notre Dame's CVRL, and Purdue University's SCALE for their contributions and support throughout this project.