Titanic MLOps Pipeline Report

1. Architecture & Toolchain

The pipeline uses a robust MLOps stack:

- Git: Version control for code.

- DVC: Data versioning and pipeline orchestration.

- MLflow: Experiment tracking and model registry.

- GitHub Actions: CI/CD automation.

- Scikit-Learn: Machine learning modeling.

2. Dataset Versioning Strategy

Three versions were created:

- V1: Raw Titanic data (handled nulls/encoding in code).

- V2: Cleaned data (Age/Fare scaled, missing values handled).

- V3: Feature engineered data (Family size, Title extraction).

3. Pipeline Stages

The DVC pipeline (dvc.yaml) automates:

1. clean_data -> 2. feature_engineering -> 3. training -> 4. registration.

4. Model Comparison Results

Logistic Regression: F1 ~0.74

Random Forest: F1 ~0.80

Note: Random Forest outperformed Logistic Regression in this experiment.

5. Advanced Feature: Automatic Model Selection

A custom script (register_model.py) automatically identifies the run with the highest F1-score and registers it in the MLflow Model Registry, promoting it to 'Production'.

6. Conclusion

This project demonstrates a production-grade MLOps setup, ensuring reproducibility and automated governance for machine learning models.