**MLOps Group 24**

**Group Members**

* Rajneesh Jha (2022ac05706)
* Sri Jayant (2022ac05045)
* Abhilesh Subhashrao Jawanjal (2022ac05604)
* Jeny Jasmine (2022ac05467)
* Mulla Akib Javed (2022ac05631)

#### **Description of the Work Done**

1. **GitHub Setup and CI/CD Pipeline Configuration(M1)**:
   * Created a GitHub organization and project repository with multiple branches for collaborative development.
   * Implemented a CI/CD pipeline using GitHub Actions, triggered on push and pull request events. The pipeline includes steps for code checkout, Python setup, dependency management, code linting, unit testing, Docker image building, and deployment to Docker Hub

GitHub Repository : [Link](https://github.com/Bits-Pilani-Wilp/MLOps-Assignment1.git)

1. **Experiment and Version Control (M2)**
   * We are using MLFlow to track the parameters and metrics and same can be visible in MLFlow UPA
   * We have done the dvc initialization, added the data to it and tried to maintain the version of the same.
   * Code Link :- [Link](https://github.com/Bits-Pilani-Wilp/MLOps-Assignment1/blob/main/model_development/Predicting%20Credit%20Eligibility%20v2(ML%20Flow%20integrated).ipynb)
   * ML Flow Run Link :- [Link](https://github.com/Bits-Pilani-Wilp/MLOps-Assignment1/blob/main/mlruns.zip)
   * DVC Link :- [Link](https://github.com/Bits-Pilani-Wilp/MLOps-Assignment1/tree/main/data)
   * Report Link :- [Link](https://github.com/Bits-Pilani-Wilp/MLOps-Assignment1/blob/main/ExperimentTracking_Data%20Versioning.docx)
2. **ML Model Development and Packaging(M3)**:
   * Developed a supervised binary classification model for banks to approve or reject credit card applications based on historical data. [Link](https://github.com/Bits-Pilani-Wilp/MLOps-Assignment1/blob/main/model_development/Predicting%20Credit%20Eligibility%20v1.ipynb)
   * Conducted data analysis, feature engineering, and feature selection to enhance model performance. Techniques used include extracting new features, label encoding, and various feature selection methods. [Link](https://github.com/Bits-Pilani-Wilp/MLOps-Assignment1/blob/main/model_development/Predicting%20Credit%20Eligibility%20v1.ipynb)
   * Built and tuned models (XGBoost, RandomForest) with hyperparameter optimization using Optuna. Stored model artifacts with Pickle and Joblib for future inferencing. [Link](https://github.com/Bits-Pilani-Wilp/MLOps-Assignment1/blob/main/model_development/Predicting%20Credit%20Eligibility%20v1.ipynb) [Link](https://github.com/Bits-Pilani-Wilp/MLOps-Assignment1/tree/main/model_artifacts)
   * Create a Flask application to expose and API for model inferencing [Link](https://github.com/Bits-Pilani-Wilp/MLOps-Assignment1/blob/main/app.py)
   * Packaged the model using Docker and created a Docker image for deployment. [Link](https://github.com/Bits-Pilani-Wilp/MLOps-Assignment1/blob/main/dockerfile)
   * M3 Detailed Report : [Link](https://github.com/Bits-Pilani-Wilp/MLOps-Assignment1/blob/main/model_development_and_packaging.docx)

4. **Deployment on GCP (M4)**

URL :<https://my-ml-app-jbw5x6tq6a-uc.a.run.app>

* + gcloud auth configure-docker
  + docker pull srijayantsingh/my-ml-app:latest
  + docker tag srijayantsingh/my-ml-app:latest gcr.io/mlops-assignment-1-431618/my-ml-app:latest
  + docker push gcr.io/mlops-assignment-1-431618/my-ml-app:latest
  + gcloud run deploy my-ml-app --image gcr.io/mlops-assignment-1-431618/my-ml-app:latest --platform managed --region us-central1 --allow-unauthenticated

#### **Justification of the Choices Made**

1. **CI/CD Pipeline**:
   * Automating testing and deployment ensures code is always deployable, reducing integration risks and speeding up development cycles
2. **MLflow for Experiment Tracking**:
   * Provides a robust framework for tracking experiments, crucial for reproducibility and collaboration in ML projects
3. **DVC for Version Control**:
   * Efficiently manages datasets and model versions, allowing easy tracking of changes and reversion to previous versions if necessary
4. **Optuna for Hyperparameter Optimization**:
   * Offers flexible and efficient hyperparameter tuning using Bayesian Optimization methods, significantly improving model performance. Integrates easily with popular ML libraries
5. **Flask for API Deployment**:
   * A lightweight and flexible framework ideal for deploying ML models as RESTful APIs, enabling easy integration with other services and applications.
6. **Docker for Containerization**:
   * Ensures consistent application performance across different environments, simplifying deployment and improving scalability
7. **Kubernetes for Orchestration**:

* Automates deployment, scaling, and management of containerized applications, ensuring high availability and efficient resource utilization​(Summary Document (1))​.

1. **Google Kubernetes Engine (GKE)**:

* Provides a managed Kubernetes service for deploying and managing containerized applications in GCP, enhancing efficiency and scalability

These choices collectively enhance the efficiency, reproducibility, and scalability of the ML workflow, aligning with best practices in MLOps.