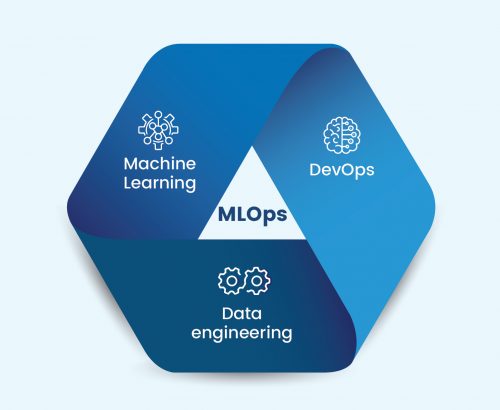
**End-to-End Report**

**MLOps for Predictive Maintenance**

Final Projec



|  |  |
| --- | --- |
| Group Members | |
| Zahid Imran | 20i-0469 |
| Ali Tajir | 20i-0512 |
| Muhammad Shehzad | 20i-17 |

**MLOps for Predictive Maintenance: End-to-End Report**

December 10, 2023

1. **Introduction**

Predictive maintenance is crucial in industrial settings to optimize operational efficiency and reduce downtime. This report outlines the successful completion of an end-to-end MLOps pipeline for predictive maintenance, incorporating industry-standard tools such as GitHub, DVC, MLflow, Docker, Flask, and Python libraries like Scikit-learn and XGBoost.

1. **Repository Structure**

The project repository, located at [https://github.com/mlopsmystics/project.git](https://github.com/mlopsmystics/project.git" \t "https://chat.openai.com/c/_new), is organized into branches corresponding to individual group members' roll numbers.The repository structure is designed to facilitate collaboration and organization among team members (i200469, i201756, and i200512) responsible for different components of the project. The repository features branches named after each team member's roll number, ensuring a clear separation of responsibilities. The branches are structured as follows:

20i0469: Dedicated to the team member with roll number i200469.

20i1756: Dedicated to the team member with roll number i201756.

20i0512: Dedicated to the team member with roll number i200512.

Each branch serves as an isolated workspace for the respective team member to work on their assigned tasks.

Main Branch - Coordination and Documentation

The main branch acts as a coordination hub, bringing together the work of individual members. It includes a file titled "tasks.txt," outlining the specific tasks assigned to each member. This file serves as a roadmap for evaluation and coordination.

20i0469: Responsible for DVC implementation.

20i1756: Responsible for model training.

20i0512: Responsible for model deployment.

1. **Data Acquisition**

A data pipeline was established to collect simulated sensor data from industrial machines. DVC was utilized for version controlling the data, and the data was stored on Google Drive. A Python script to generate dummy data periodically simulates the gathering of live stream data.

1. **Data Pre-processing**

Data pre-processing involved normalizing or scaling features as necessary. The dataset was then split into training and validation sets, ensuring a robust foundation for model training.

1. **Model Training**

A suitable machine learning algorithm for time-series data, XGBoost, was chosen. The predictive maintenance model was implemented, trained, and hyperparameters tuned for optimal performance. MLflow was used to track model parameters and metrics, and the best model was registered for the production stage. Testing on live data was automated using a provided script.

1. **Deployment**

The trained model was packaged into a Docker container, facilitating easy accessibility and reproducibility. The Docker image was pushed to Docker Hub, and a Flask application was created to serve predictions using the deployed model.

1. **Concept Drift Monitoring**

To ensure the model's ongoing performance, a concept drift monitoring system was implemented. Relevant drift metrics were defined and tracked, and automated retraining triggers based on observed drift were established.

1. **Documentation and Presentation**

Each stage of the MLOps pipeline was thoroughly documented, outlining the tools and methodologies used. A comprehensive presentation summarizing the project was prepared, highlighting challenges and solutions encountered during the implementation of MLOps concepts.

1. **Tools and Technologies**

The project leveraged the following tools and technologies:

Version Control: GitHub

Data Tracking: DVC

Model Tracking: MLflow

Containerization: Docker

Web Framework: Flask

Programming Languages: Python

Machine Learning Libraries: Scikit-learn, XGBoost, MLflow

Data Processing: Pandas, NumPy

1. **Evaluation Criteria**

The project's evaluation focused on the implementation of MLOps concepts rather than model accuracy. Each team member's effort in executing the assigned tasks was assessed, and post-deadline demos allowed for an in-depth explanation of individual contributions.

In conclusion, the successful completion of this end-to-end MLOps pipeline for predictive maintenance showcases the effective integration of tools and methodologies to enhance operational efficiency and reliability in industrial settings. The emphasis on documentation ensures reproducibility, and the presentation provides a comprehensive overview of the project's achievements and challenges.