Robert Pavlik

D601 – Deployment

QBN1 Task 1: Business Case Analysis

A. Objectives of an MLOps Deployment Architecture

Implementing a Machine Learning Operations (MLOps) deployment architecture at Kronkers aims to address critical challenges in the current machine learning (ML) workflow. The primary objectives include:

1. **Standardization and Automation**:
   Establishing a structured and automated ML pipeline to improve model reproducibility, scalability, and operational efficiency across departments.

2. **Model Deployment and Accessibility**:
   Enabling seamless deployment of ML models in a production environment, ensuring that business teams can access and utilize models effectively.

3. **Integration Across Multiple Programming Languages:**
   Developing a platform that supports models built in Python, R, and Julia, ensuring a unified environment for all data analysts and scientists.

4. **Centralized Model and Data Management:**
   Creating a repository for ML models, datasets, and results that facilitates version control, monitoring, and governance.

5. **Real-time and Batch Processing Capabilities:**
   Supporting both real-time and scheduled model executions to improve decision-making and operational efficiency.

6. **Security and Intellectual Property Protection:**
   Implementing security protocols to safeguard proprietary ML models, data assets, and insights from competitors.

7. **Scalability and Maintainability:**
   Establishing a system that can grow with the company and adapt to evolving business needs, reducing technical debt and maintenance costs.

B. Constraints to Implementing an MLOps Solution

Despite the potential benefits of MLOps, several constraints must be considered

1. **Leadership Skepticism:**
   Some senior executives question the necessity of MLOps, which may impact approval and funding.

2. **Budgetary Limitations:**
   While there is an allocated budget for model maintenance, tracking, and quality control, funding constraints may limit the scope of the initial deployment.

3. **Technical Complexity:**
   Integrating multiple ML models developed in different languages with different parameters adds complexity to system design and implementation.

4. **Current Infrastructure Limitations:**
   The existing storage solution (OneDrive) lacks advanced version control, automated deployment, and real-time processing capabilities.

5. **Workforce and Expertise:**
   With only a few experienced programmers and data scientists, Kronkers lacks a dedicated MLOps team to oversee deployment and maintenance.

6. **Data Governance and Compliance:**
   Ensuring compliance with data security and regulatory requirements while maintaining accessibility for multiple departments.

C. Functional and Non-Functional Requirements for the Proposed MLOps Solution

Functional Requirements

1. **Model Training and Deployment:**
   The system must support the training, retaining, and deployment of ML models in multiple programming languages.

2. **Version Control:**
   A robust versioning system for both code and data assets to enable reproducibility and rollback capabilities.

3. **Automated Model Monitoring and Logging:**
   Real-time tracking of model performance, drift detection, and logging mechanisms to ensure accuracy and reliability.

4. **Scalable Infrastructure:**
   Cloud-based or hybrid solutions allow flexibility in resource allocation for various workloads.

5. **User Access Control:**
   Role-based access control (RBAC) to restrict and grant permissions as needed across different teams.

6. **Interoperability:**
   The system should allow integration with existing tools, including Python, R, Julia, and popular cloud platforms.

Non-Functional Requirements

1. **Scalability:**
   The architecture should support increasing workloads as the company grows.

2. **Security and Compliance:**
   Data encryption, authentication mechanisms, and adherence to industry regulations.

3. **Performance and Efficiency:**
   The solution should optimize compute and storage resources to ensure cost-effectiveness.

4. **Reliability and Availability:**
   High system uptime and failover mechanisms to prevent disruptions.

5. **Usability:**
   A user-friendly interface for non-technical stakeholders to access and interpret model results.