1. Overview

The **deployment phase** involves making a trained machine learning model accessible in a real-world or production environment where it can generate predictions on new data. Key steps include model serialization, selecting a platform for hosting, API integration, implementing security measures, and establishing monitoring and logging mechanisms. The goal is to ensure the model performs efficiently and securely while delivering accurate results in real-time.

2. Model Serialization

Model serialization is the process of converting the trained model into a format that can be saved and loaded for future use. Common formats include:

- **Pickle** for Python-based models.

3. Model Serving

The serialized model is **served** by loading it into a suitable environment for making predictions.

- The model deployed via streamlit

4. Security Considerations

Security measures include:

- **Authentication**: Ensuring that only authorized users can access the model (e.g., API keys, OAuth tokens).

5. Monitoring and Logging

The deployed model's **performance is monitored** continuously to track metrics like:

- **Prediction accuracy**: Comparing predicted outcomes against true labels (if available).
- **Latency**: Time taken for the model to generate predictions.
- **Throughput**: Number of requests handled over time. **Alerting mechanisms** can be implemented to notify if performance degrades or there are system issues (e.g., using tools like Prometheus and Grafana). Logs are collected to track errors, request history, and performance over time for analysis and troubleshooting.
- **Feedback:** The feedback from the users will help to get to know the model performance and how to improve it.