**Lab 12**

**Deployment of model real time data streaming**

Implementing a **simple ML model** (e.g., a real-time sentiment analyzer) with **Kubernetes deployment** and **live visualization**. Use python-based tools.

**Prerequisites**

* Python 3.8+
* Docker + Kubernetes (Minikube for local testing)
* Kafka (or Redpanda for lightweight setup)
* Basic knowledge of Python ML (scikit-learn)

Install Tools:

|  |
| --- |
| # Python libraries  pip install pandas scikit-learn kafka-python flask plotly dash  # For Kubernetes  brew install kubectl helm minikube # macOS  minikube start # Start local K8s cluster  # Kafka (using Redpanda for simplicity)  docker run -d -p 9092:9092 --name redpanda docker.redpanda.com/vectorized/redpanda |

Train a **sentiment analysis model** on any Kaggle or public dataset: **recommended** Twitter API

python train\_model.py

**Set Up Real-Time Data Stream**

**Kafka Producer (**producer.py**)**

Sends fake tweets for sentiment analysis.

**Kafka Consumer + Predictor (**consumer.py**)**

Consumes tweets and runs predictions.

**Deploy on Kubernetes**

**Dockerize the Predictor**

|  |
| --- |
| # Dockerfile  FROM python:3.8  COPY . /app  WORKDIR /app  RUN pip install kafka-python scikit-learn  CMD ["python", "consumer.py"] |

Build and push:

|  |
| --- |
| docker build -t sentiment-predictor .  minikube image load sentiment-predictor # For local K8s |

**Kubernetes Deployment (**deployment.yaml**)**

|  |
| --- |
| apiVersion: apps/v1  kind: Deployment  metadata:  name: sentiment-predictor  spec:  replicas: 2  selector:  matchLabels:  app: predictor  template:  metadata:  labels:  app: predictor  spec:  containers:  - name: predictor  image: sentiment-predictor  env:  - name: KAFKA\_BROKER  value: "redpanda:9092" # Service name for Kafka  ---  apiVersion: v1  kind: Service  metadata:  name: redpanda  spec:  ports:  - port: 9092  selector:  app: redpanda |

Apply:

kubectl apply -f deployment.yaml

**Real-Time Visualization**

**Dash App (**visualization.py**) for kafka-consumer**

**Access at localhost.**

**Full Pipeline Workflow**

1. **Producer** sends tweets → Kafka topic tweets.
2. **Consumer** (K8s pod) reads tweets, predicts sentiment.
3. **Dash** app visualizes live predictions.

**Add Prometheus to monitor prediction latency**