

# Real-Time Prediction with Apache Kafka and Neural Networks

## Overview

In this assignment, you will build a real-time data processing system using Apache Kafka. Your system will consist of a **producer** that streams input data (any data you can take like stock price etc.) and a **consumer** that applies a neural network (NN) model to predict an output value (say next day price based on input price) from the incoming data. This project will help you understand distributed messaging systems, streaming data processing, and integrating machine learning into real-time applications.

**Note :** Can take any dataset as input.

## Objectives

- **Kafka Fundamentals:** Gain hands-on experience with Apache Kafka, including setting up topics, producers, and consumers.
- **Producer-Consumer Architecture:** Understand how to design and implement a system where the producer sends data and the consumer processes it asynchronously.
- **Neural Network Integration:** Develop and integrate a neural network using a framework such as TensorFlow or PyTorch to predict values based on streaming data.

## Tasks and Requirements

### 1. Environment Setup

- **Kafka Cluster:** Install Apache Kafka (locally or using Docker) and create necessary topics (e.g., input-data for the producer's messages and predictions for output).
- **Programming Language:** Use Python along with libraries such as kafka-python (or Confluent's Kafka client) for interacting with Kafka.
- **Neural Network Framework:** Choose a machine learning framework (TensorFlow, Keras, or PyTorch) to build your prediction model.

### 2. Producer Implementation

- **Data Generation:** Implement a producer that simulates or streams input data. This data can be a simple numerical feature set (e.g., a list of sensor readings or simulated values or live stock price or anything you feel convenient) that the neural network will use for prediction (next days value etc.).
- **Message Formatting:** Ensure that each Kafka message includes:
  - o A unique identifier.
  - o The input features (in JSON or another serializable format).
  - o A timestamp.

### 3. Consumer Implementation with Neural Network Prediction

- **Data Ingestion:** Build a consumer that listens to the input-data topic.
- **Data Preprocessing:** Include steps to parse and preprocess the incoming data so that it fits the neural network's input format.
- **Neural Network Model:**
  - **Training Phase:** (Either offline or online) Train a neural network model on historical or simulated data. The task can be a regression or classification problem based on the input features.
  - **Integration:** Load the trained model in the consumer. For every incoming message, the consumer should use the model to predict an output value.
- **Output Handling:** The predicted value (along with the original input and a prediction timestamp) should be published to a separate Kafka topic, such as predictions.

#### 4. Testing and Evaluation

- **Performance Metrics:** Choose appropriate metrics (e.g., Mean Squared Error for regression or accuracy for classification) to evaluate the predictions.
- **Real-Time Monitoring:** Implement logging or a simple dashboard to monitor:
  - The number of messages processed.
  - Prediction latencies.
  - Performance metrics over time.

### Deliverables

- **Source Code:** Provide all source code files for the producer and consumer.
  - **Report:** A report detailing the design, implementation, and evaluation of your system.
- 

This assignment is designed to give you practical experience with integrating stream processing and machine learning. It encourages you to think about the complexities of real-time data systems while applying neural network models to solve predictive tasks.