CloudSEK

ML Assignment

Problem Statement - 1

Design and implement a continuous training and deployment pipeline for a transformer-based text classification model. The system should support retraining when the model becomes stale or performance degrades. Model staleness can be defined by any metric you choose (e.g., accuracy drop, data drift).

Deploy the model locally or using any cloud infrastructure of your choice (AWS, GCP, Azure). The solution should accommodate both live and batch inference.

Deliverables

1. **Design Document**: A document outlining the architecture for the continuous training and deployment pipeline. The document should:
    - Explain how you will monitor model staleness.
    - Describe the infrastructure for deployment (local or cloud).
    - Detail the inference methods (live and batch) and cost optimizations.

2. **Source Code**: Python code that implements the pipeline, including:
    - Data preprocessing
    - Training and retraining logic.
    - Monitoring setup for performance degradation and retraining triggers / can create gmail alerts or anything suitable. You can use MLflow.
    - Inference scripts for both batch and live requests.
3. **Deployment Instructions**: Steps for deploying the model locally, or in a cloud environment of your choice.

Constraints

- Use a transformer model for text classification (e.g., BERT, GPT, etc.).
- You can use any text classification dataset available, i.e sentiment classification, fraud classification etc.
- Model must support live predictions and batch predictions.
- The model should be retrained upon detecting performance staleness or degradation.
- Cost optimizations should be considered for the training and inference pipelines.

Problem Statement - 2

Diffusion Model for Feature Extraction and Reconstruction Error Detection.

You are tasked with designing and implementing a machine learning pipeline that leverages a diffusion model for identifying reconstruction errors. The extracted features will then be processed using a CNN+LSTM architecture for further analysis.

Objective

The goal is to utilize a diffusion model to capture reconstruction errors in the data, which will serve as a method of feature extraction. These features will then be used by a CNN+LSTM model to perform predictions or classifications based on a downstream task (e.g., anomaly detection, video binary classification).

Deliverables

1. **Design Document**:

    - A detailed design document that explains the architecture and flow of the pipeline.
    - Justify the use of a diffusion model for reconstruction error detection and explain how the CNN+LSTM model leverages the extracted features for classification or regression.
    - Explain the process, assumptions, and configurations.
    - Steps to replicate the training process and how to use the trained models for inference.
    - Mention any research papers you have used and benchmarks they have used.

2. **Source Code**:
    - Implementation of the diffusion model for reconstruction error detection.
    - CNN+LSTM model code for processing the features extracted from the diffusion model.
    - End-to-end training pipeline that includes preprocessing, feature extraction, and prediction.
3. **Evaluation Metrics**:
    - Documentation explaining how you will evaluate the architecture of the pipeline.
4. **Optional**:
    - Instructions for deploying the model locally or in a cloud environment for inference.

Constraint :

1. No need to train the model, create the architecture using a single video file.
2. Take the final prediction of CNN+LSTM to be a binary output, i.e 1,0.

Timeline : You have to submit the assignment in 5 days.

Submission : Share your zip file over the google drive with access to **lasya.ippagunta@cloudsek.com** and **apurv.singh@cloudsek.com.**

Keep the name of file as following :

1. Problem_1_{your_name}.zip
2. Problem_2_{your_name}.zip