Assignment: Invoice Data Extraction Using Machine Learning

Objective: Develop a Python application to extract key information from invoices using machine learning. The project involves training a model, optimizing it for deployment, and running it on a client desktop. The solution should handle various invoice formats in English, Dutch, and French without hardcoded labels, understanding the context to accurately extract information.

Instructions

Part 1: Model Training

1. **Environment Setup**:
   - Install Python and necessary libraries.
   - Set up a virtual environment (optional but recommended).
2. **Data Collection**:
   - Collect a diverse dataset of invoices in PDF format from the internet. Ensure the dataset includes invoices in English, Dutch, and French.
   - Use OCR to convert PDFs to text.
3. **Data Preprocessing**:
   - Clean and preprocess the extracted text.
   - Annotate the data to identify key information (e.g., sender, receiver, VAT number, amounts) without relying on hardcoded labels.
4. **Model Training**:
   - Use a pre-trained model and fine-tune it on your annotated dataset.
   - Ensure the model understands context to extract information from various formats and languages.
   - Evaluate the model's performance and adjust parameters as necessary.

Part 2: Model Optimization

1. **Convert to ONNX**:
   - Export the trained model to ONNX format.
2. **Optimize the Model**:
   - Use techniques like quantization to reduce model size and improve performance.

Part 3: Model Deployment

1. **Set Up Client Environment**:
   - Ensure the client machine has the necessary libraries installed.
2. **Load and Run the Model**:
   - Write a script to load and run the optimized model on the client desktop.

Deliverables

1. **Code Repository**:
   - All scripts for data preprocessing, model training, optimization, and deployment.

   - A README file with detailed instructions on how to run the code.
2. **Documentation**:
   - A report documenting your approach, model architecture, training process, evaluation metrics, optimization techniques, deployment steps, and performance on the client desktop.

Evaluation Criteria

- **Data Handling**: Ability to collect, preprocess, and annotate data, including handling multiple languages.
- **Model Training**: Effectiveness in training and fine-tuning the model to handle diverse formats.
- **Optimization**: Success in optimizing the model for deployment.
- **Deployment**: Ability to set up the client environment and run the model.
- **Documentation**: Clarity and completeness of the documentation and code repository.