**Invoice Information Extraction Using Machine Learning**

**1. Introduction**

This project involves developing a Python application to extract key information from invoices, including sender, receiver, VAT number, amounts, etc., in multiple languages and formats. The application leverages machine learning techniques and OCR for accurate information extraction.

**2. Approach**

The approach can be divided into several key stages:

1. **Data Collection and Preprocessing**
2. **Model Training**
3. **Model Optimization**
4. **Model Deployment**

**Part 1: Model Training**

**1. Environment Setup**

* **Install Python and necessary libraries:**
  + Python 3.8 or later.
  + Libraries: pandas, numpy, scikit-learn, torch (for PyTorch), transformers (for pre-trained models), pytesseract (for OCR), pdf2image (to convert PDFs to images), nltk (for text preprocessing).
* **Set up a virtual environment:**

python -m venv invoice\_env

source invoice\_env/bin/activate # On Windows use `invoice\_env\Scripts\activate`

**2. Data Collection**

* **Collect a diverse dataset of invoices:**
  + Download or scan invoices in English, Dutch, and French.
  + Store them in a structured format, e.g., data/invoices/.
* **Convert PDFs to text using OCR**

**3. Data Preprocessing**

* **Clean and preprocess the extracted text:**
  + Normalize text (convert to lowercase, remove special characters).
  + Tokenize and split into sentences.
* **Annotate the data:**
  + Identify key information (e.g., sender, receiver, VAT number, amounts) using regex and other text processing techniques.

**4. Model Training**

* **Use a pre-trained model and fine-tune it:**
  + Use a pre-trained language model like BERT or GPT for Named Entity Recognition (NER).
  + Fine-tune the model on your annotated dataset.

**Evaluate the model's performance:**

* Split data into training and validation sets.
* Use metrics like precision, recall, and F1 score.

**Part 2: Model Optimization**

**1. Convert to ONNX**

* Export the trained model to ONNX format:

import torch

import onnx

dummy\_input = torch.zeros(1, 512, dtype=torch.int64)

torch.onnx.export(model, dummy\_input, "model.onnx")

**2. Optimize the Model**

* **Use quantization**

from onnxruntime.quantization import quantize\_dynamic, QuantType

quantize\_dynamic("model.onnx", "model\_quantized.onnx", weight\_type=QuantType.QUInt8)

**Part 3: Model Deployment**

**1. Set Up Client Environment**

* **Ensure the client machine has the necessary libraries:**
  + Install ONNX runtime: pip install onnxruntime.

**2. Load and Run the Model**

* **Write a script to load and run the optimized model**