**AI-Powered IT Support Deployment Guide**

**1. Prerequisites**

Ensure you have the following installed:

* Python 3.8+
* Flask
* Scikit-learn
* Joblib
* ServiceNow Developer Instance (or access to an existing instance)
* REST API permissions in ServiceNow

**2. Setting Up the AI Service**

**Step 1: Install Dependencies**

Run the following command to install necessary Python libraries:

pip install flask scikit-learn joblib pandas

**Step 2: Train the AI Model**

Run the training script to create the model and vectorizer files:

python train\_ai\_model.py

This will generate ticket\_classifier\_pipeline.pkl and tfidf\_vectorizer.pkl.

**Step 3: Start the Flask API**

Run the following command to start the AI service:

python ai\_ticket\_resolution.py

The service will be available at http://localhost:5000/predictTicket.

**3. Configuring ServiceNow**

**Step 1: Create a REST API Integration**

1. Navigate to **System Web Services > REST Message** in ServiceNow.
2. Click **New** and configure:
   * Name: AI\_Ticket\_Service
   * Endpoint: http://your-server-ip:5000/predictTicket
   * HTTP Method: POST
   * Content-Type: application/json

**Step 2: Implement the Business Rule**

1. Navigate to **System Definition > Business Rules**.
2. Click **New** and configure:
   * Name: AI Incident Resolver
   * Table: Incident
   * When: After Insert
   * Script: Copy & paste the servicenow\_integration.js script.

**Step 3: Add Custom Fields in Incident Table**

* u\_ai\_resolved (Boolean) – To track if AI auto-resolved the incident.
* u\_predicted\_category (String) – Stores AI-predicted category.

**4. Testing the Integration**

**Step 1: Submit a Test Incident**

1. Create a new incident in ServiceNow with a short description like:

Cannot access email, forgot password.

1. The AI should classify it as **Password Reset** and auto-resolve it.

**Step 2: Check Logs and Debugging**

* View AI Service logs in the terminal.
* Check ServiceNow **System Logs > Script Logs** for API call details.

**5. Deployment in Production**

**Option 1: Deploy on a Local Server**

Run the Flask app in the background:

nohup python ai\_ticket\_resolution.py &

**Option 2: Deploy on a Cloud Server (Recommended)**

1. Choose a cloud provider (AWS, Azure, GCP, etc.).
2. Set up a virtual machine and install Python.
3. Deploy using a service like **Gunicorn**:
4. pip install gunicorn

gunicorn -w 4 -b 0.0.0.0:5000 ai\_ticket\_resolution:app

1. Configure firewall rules to allow requests to port 5000.

**Option 3: Deploy with Docker**

1. Create a Dockerfile:
2. FROM python:3.8
3. WORKDIR /app
4. COPY . /app
5. RUN pip install flask scikit-learn joblib pandas

CMD ["python", "ai\_ticket\_resolution.py"]

1. Build and run:
2. docker build -t ai\_ticket\_service .

docker run -p 5000:5000 ai\_ticket\_service

**6. Monitoring & Maintenance**

* Monitor logs in ServiceNow and Flask.
* Regularly retrain the AI model with new tickets.
* Update ServiceNow scripts for optimizations.

**7. Future Enhancements**

* Improve NLP accuracy with fine-tuned models.
* Integrate AI with ServiceNow’s Virtual Agent.
* Enable AI to suggest knowledge base articles for unresolved tickets.