**Skill Trend Detector API - Complete Setup & Documentation**

**Overview**

A REST API that analyses job descriptions to detect technical skills and classify them as either "emerging" (new/trendy) or "established" (common/traditional) based on market frequency patterns.

**Setup Instructions**

**Prerequisites**

* Python 3.7 or higher
* pip package manager
* A dataset CSV file with job descriptions

**Step 1: Environment Setup**

# Create project directory

mkdir skill-trend-detector

cd skill-trend-detector

# Create virtual environment (recommended)

python -m venv venv

# Activate virtual environment

# Windows:

venv\Scripts\activate

# macOS/Linux:

source venv/bin/activate

**Step 2: Install Dependencies**

# Install required packages

Install: pip install -r requirements.txt

**Step 3: Project Structure**

skill-trend-detector/

├── main.py # FastAPI application

├── Data\_analysis.ipynb # Analysis of Dataset

├── trend\_finder\_model.py # Core ML model

├── dataset.csv # Your job descriptions dataset

├── requirements.txt # Dependencies

└── README.md # This documentation

**Step 4: Dataset Preparation**

Your dataset.csv should have at least these columns:

job\_description\_text,other\_columns,

"Experience with Python and TensorFlow required.",

"Looking for Java developer with AWS skills.",

**Step 5: Run the API**

# Method 1: Direct run

python main.py

# Method 2: Using uvicorn

uvicorn main:app --reload --host 0.0.0.0 --port 8000

The API will be available at: http://localhost:8000/docs

**API Documentation**

**Base URL**

<http://localhost:8000>

**Endpoints**

**1. Root Endpoint**

**GET** /

Returns basic API information.

**Response:**

{

"message": "Skill Trend Detector API",

"version": "1.0.0",

"docs": "/docs"

}

**2. Health Check**

**GET** /health

Check API status and get dataset statistics.

**Response:**

{

"status": "healthy",

"total\_skills": 150

}

**3. Skill Detection (Main Endpoint)**

**POST** /detect\_skills

Analyzes job description and returns detected skills with trend classifications.

**Request Body**

{

"job\_description": "string (required)"

}

**Example Request**

{

"job\_description": "Experience with Python, TensorFlow required. Must have AWS cloud knowledge."

}

**Response Format**

{

"detected\_skills": [

{

"skill": "string",

"category": "emerging|established",

"trend\_score": "float (0-1)"

}

]

}

**Example Response**

{

"detected\_skills": [

{

"skill": "python",

"category": "established",

"trend\_score": 0.76

},

{

"skill": "tensorflow",

"category": "established",

"trend\_score": 0.39

},

{

"skill": "aws",

"category": "established",

"trend\_score": 0.45

}

]

}

**Error Responses**

* **400 Bad Request**: Invalid or empty job description

{

"detail": "Job description is empty."

}

* **500 Internal Server Error**: Processing error

{

"detail": "Error processing request: [error details]"

}

**Output**

A screenshot of a computer

AI-generated content may be incorrect.

**Testing the API**

**Using Interactive Docs (Recommended)**

1. Start the API: python main.py
2. Open browser: http://localhost:8000/docs
3. Click on /detect\_skills → "Try it out"
4. Enter your job description and click "Execute"

**How It Works**

**Skill Detection Process**

1. **Text Preprocessing**: Cleans and normalizes input text
2. **Keyword Matching**: Searches for predefined technical skills
3. **Frequency Analysis**: Compares against dataset statistics
4. **Trend Classification**: Categorizes based on popularity thresholds

**Classification Logic**

* **Trend Score**: skill\_frequency / most\_common\_skill\_frequency
* **Emerging**: Skills with trend\_score < 0.05 (less than 5% popularity)
* **Established**: Skills with trend\_score ≥ 0.05 (5%+ popularity)

**Local Development**

python main.py

# API runs on http://localhost:8000