Student Profile Authenticity Scoring Microservice

A microservice that evaluates student bios and headlines for authenticity and originality. Designed to detect fake, AI-generated, or exaggerated content using NLP and machine learning models.

# Features

- Detects fabricated or padded profile content

- Scores profiles based on authenticity

- Flags overused/buzzword-heavy language

- Works fully offline

- REST API built with FastAPI

- Dockerized for easy deployment

# Project Structure

profile\_scorer/  
├── app/  
│ ├── api/  
│ ├── core/  
│ ├── models/  
│ ├── utils/  
│ └── main.py  
├── data/  
├── tests/  
├── config.json  
├── Dockerfile  
├── requirements.txt  
└── README.md

# Configuration (config.json)

{  
 "score\_threshold": 0.5,  
 "regex\_blacklist": ["highly motivated", "passionate", "team player"],  
 "model\_fallback": true  
}

# Example API Usage

POST /check-profile

{  
 "user\_id": "stu\_9317",  
 "profile\_data": {  
 "headline": "Highly motivated engineering student",  
 "bio": "A passionate learner aiming to revolutionize tech."  
 }  
}

Response:

{  
 "user\_id": "stu\_9317",  
 "authenticity\_score": 0.38,  
 "verdict": "likely\_fabricated",  
 "reason": "Overused language, low uniqueness",  
 "flagged\_fields": ["headline", "bio"]  
}

# Model Information

- Features Used:

* - TF-IDF n-grams
* - Readability score (textstat)
* - Buzzword count

- Model Type: Logistic Regression / Random Forest

- Performance:

* - Accuracy: ~87%
* - Precision: 0.84
* - Recall: 0.88
* - ROC-AUC: 0.91

# Running the Microservice

1. Build Docker Image:

docker build -t profile-scorer .

2. Run the Container:

docker run -p 8000:8000 profile-scorer

3. Access API Docs:

Swagger UI: http://localhost:8000/docs

# API Endpoints

GET /health - Health check

GET /version - Returns API version

POST /check-profile - Scores a profile's authenticity

# Testing

Run all tests:

pytest tests/

Includes unit tests, model inference validation, and integration tests.

# Packaging & Delivery

To share:

Upload code and model files to GitHub or export Docker image:

docker save -o profile\_scorer\_image.tar profile-scorer

# Tools & Libraries

API Development: FastAPI, Uvicorn

NLP & ML: Scikit-learn, NLTK, Textstat

Feature Engineering: TF-IDF, Regex, Readability Score

Testing: Pytest, HTTPX

Deployment: Docker

Documentation: Swagger (via FastAPI)

# Author

Built by Sakshi as part of the TruePersona project.