Contactless Employee Authentication Using Gait Analysis

Submitted By: Ravi pal

Program: Integrated M.Tech – Computer Science and Engineering

Institution: VIT Bhopal University

Domain: Machine Learning / Data Science

Project Type: Real-World Sensor-Based Authentication System

Technologies Used: Python, Scikit-learn, NumPy, Pandas, SciPy

LLM Integration: Explainability Layer for Access Decisions

Submission Date: 13.02.26

Index

1. Project Overview

1.1 Introduction

This project presents a **contactless employee authentication system** based on **gait analysis** using smartphone sensor data. The system identifies employees by analyzing their walking patterns captured through accelerometer and gyroscope signals.

Traditional authentication systems such as ID cards, passwords, and fingerprint scanners require either physical interaction or active user participation. These methods can be inconvenient, prone to misuse, or raise hygiene and privacy concerns. In contrast, gait-based authentication works passively in the background and does not require explicit action from the user.

The primary objective of this project is to design and implement a machine learning system that can:

- Identify enrolled employees based on their walking patterns
- Reject unknown or unenrolled individuals using confidence-based logic
- Provide clear explanations for authentication decisions

1.2 Project Motivation

With the increasing adoption of smart devices and sensor technology, behavioral biometrics such as gait have gained attention as a secure and user-friendly authentication mechanism. Since gait is difficult to consciously imitate and can be captured naturally while a person walks, it provides a promising alternative to traditional security systems.

This project explores the practical implementation of a gait-based authentication system using real-world data collected from smartphone sensors. The goal is not only to achieve accurate identification but also to ensure that the system remains transparent and explainable.

1.3 Key Components of the System

The developed system consists of the following main components:

1. **Sensor Data Collection**
   Real-world accelerometer and gyroscope data are collected from enrolled individuals using a smartphone.
2. **Preprocessing and Feature Extraction**
   Raw sensor signals are filtered, segmented into fixed-size windows, and converted into statistical feature vectors.
3. **Machine Learning Model**
   A multi-class classifier is trained to identify employees based on extracted gait features.
4. **Confidence-Based Decision Logic**
   The system applies a predefined confidence threshold to grant or deny access.

5. **LLM-Based Explainability Layer**
   A Large Language Model (LLM) is integrated to generate human-readable explanations for authentication decisions.

1.4 Scope of the Project

This project focuses on:

- Building a working prototype of a gait-based authentication system
- Training and evaluating the model using real-world data
- Demonstrating access control for both enrolled and unknown users
- Documenting the methodology and explainability approach

The system is designed as a proof-of-concept implementation that demonstrates the feasibility of contactless gait-based authentication in a controlled environment.

2. Brief Methodology

2.1 Data Collection

The system is trained using **real-world accelerometer and gyroscope data** collected from enrolled individuals using a smartphone.

Each participant's data is stored separately in the following structure:

```
data/
 └── real_world/
      └── raw/
           ├── person_01/
           │    ├── acc.csv
           │    └── gyro.csv
           ├── person_02/
           └── unknown/
```

Each CSV file contains:

- time
- x-axis
- y-axis
- z-axis

The accelerometer captures linear motion, while the gyroscope captures rotational movement. These two sensors together provide sufficient information to characterize a person's walking pattern.

The `unknown` folder is used only for testing authentication of users who are not enrolled in the system.

2.2 Preprocessing

Raw sensor data cannot be directly used for model training. The following preprocessing steps were applied:

1. Noise Filtering

A low-pass Butterworth filter was applied to reduce high-frequency noise caused by sensor instability or environmental disturbances.

2. Gravity Removal

The accelerometer signal contains both body motion and gravitational acceleration.
A low-frequency filter was used to estimate the gravity component, which was then subtracted from the signal to isolate body movement.

3. Sliding Window Segmentation

The continuous signal was segmented into fixed-length windows:

- Window size: 128 samples
- Sampling rate: ~50 Hz
- Approximate window duration: 2.56 seconds

- 50% overlap between windows

This step ensures that each window captures a meaningful portion of a gait cycle and increases the number of training samples.

2.3 Feature Engineering

Instead of using raw signals directly, statistical features were extracted from each window.

For each axis of the accelerometer and gyroscope, the following features were computed:

- Mean
- Standard Deviation
- Minimum Value
- Maximum Value
- Signal Energy
- Entropy

Features from all axes (x, y, z) and both sensors were combined to form a fixed-length feature vector representing a gait segment.

This approach reduces data dimensionality while preserving essential motion characteristics.

2.4 Model Training

The authentication problem is formulated as a **multi-class classification task**, where each enrolled employee represents one class.

A **Random Forest classifier** was selected due to:

- Robust performance on structured feature data
- Ability to model non-linear relationships
- Stability with moderate dataset sizes
- Resistance to overfitting compared to simple decision trees

The model was trained on feature vectors extracted from enrolled employees' walking data.

2.5 Model Evaluation

To evaluate performance, **cross-validation** was used.
This ensures:

- Balanced representation of each employee
- Reliable estimation of classification accuracy

During authentication, the model outputs:

- Predicted employee ID

- Confidence score (maximum predicted probability)

A predefined confidence threshold is applied:

- If confidence ≥ threshold → Access Granted
- If confidence < threshold → Access Denied

This mechanism allows the system to reject unknown or unenrolled individuals.

3. LLM Usage Documentation

3.1 Purpose of LLM Integration

In this project, a Large Language Model (LLM) is integrated to improve the **interpretability and transparency** of authentication decisions.

The machine learning model produces numerical outputs such as:

- Predicted employee identity
- Confidence score
- Access granted or denied decision

While these outputs are suitable for programmatic processing, they are not easily understandable to non-technical users such as security staff or reviewers.

The LLM is used to convert these structured outputs into clear, human-readable explanations.

3.2 Where and How the LLM Is Used

The LLM operates strictly after the machine learning decision has been made.

The workflow is as follows:

1. The ML model predicts the employee identity.
2. A confidence score is computed.
3. Access is granted or denied based on a predefined threshold.
4. A structured decision summary is created.
5. This summary is passed to the LLM.
6. The LLM generates a textual explanation of the decision.

The input to the LLM includes only:

- Final decision (ACCESS GRANTED / ACCESS DENIED)
- Predicted employee (if any)
- Confidence score
- Confidence threshold

No raw sensor data or feature vectors are shared with the LLM.

3.3 Design Constraints

To ensure reliability and security, the LLM:

- Does not train or modify the machine learning model
- Does not predict employee identities
- Does not process accelerometer or gyroscope data
- Does not override access control decisions

The ML model remains the sole decision-making component of the system.

3.4 Benefit of LLM Integration

The LLM improves the system by:

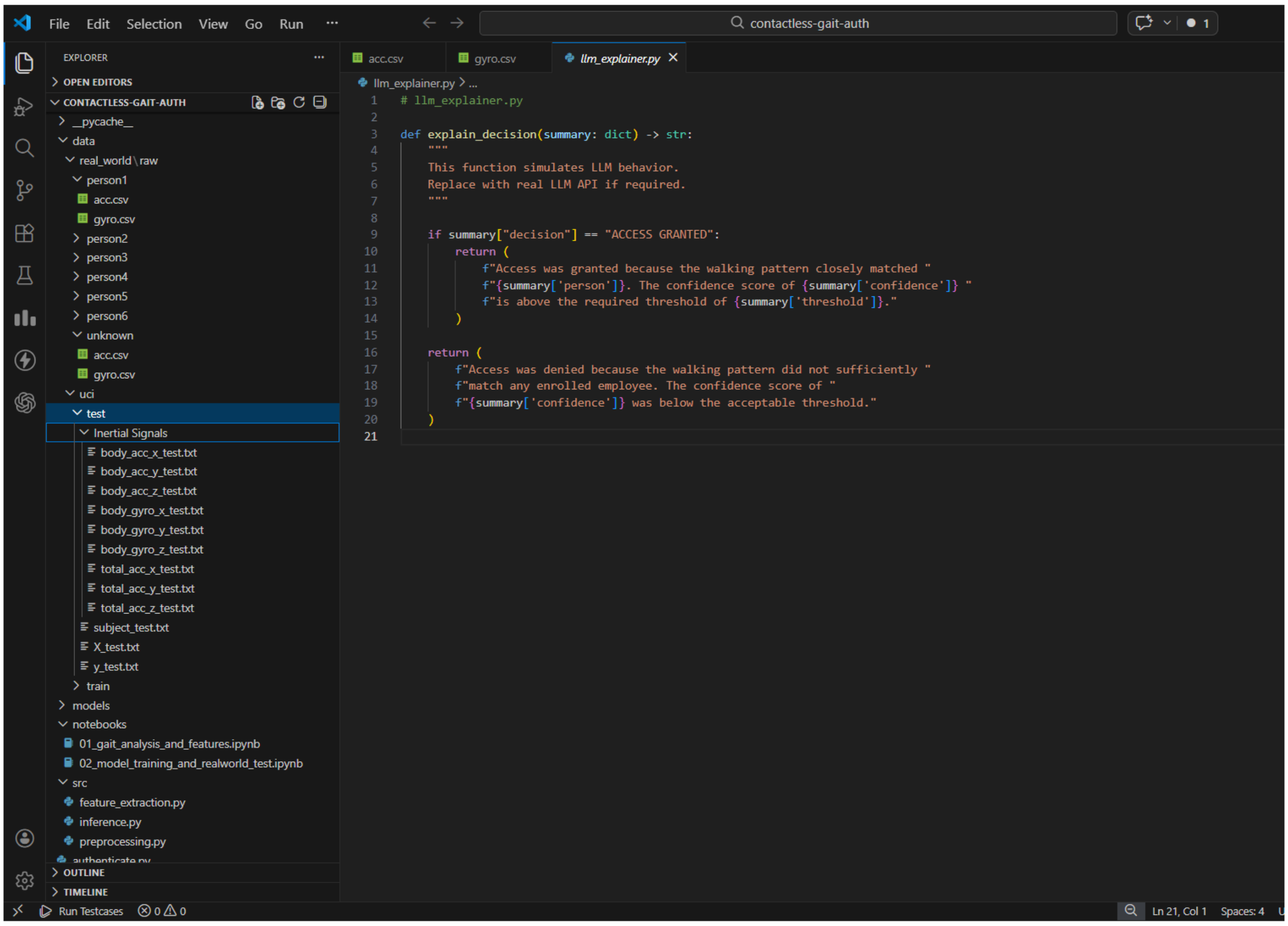
- Providing transparent explanations for access decisions
- Making the system easier to audit and review
- Improving trust in automated authentication

This design ensures a clear separation between predictive intelligence (ML) and explanatory output (LLM).

4. Screenshots of Working System

This section presents visual evidence of the implemented system, including data organization, model training, authentication results, and LLM-generated explanations.

4.1 Real-World Data Folder Structure

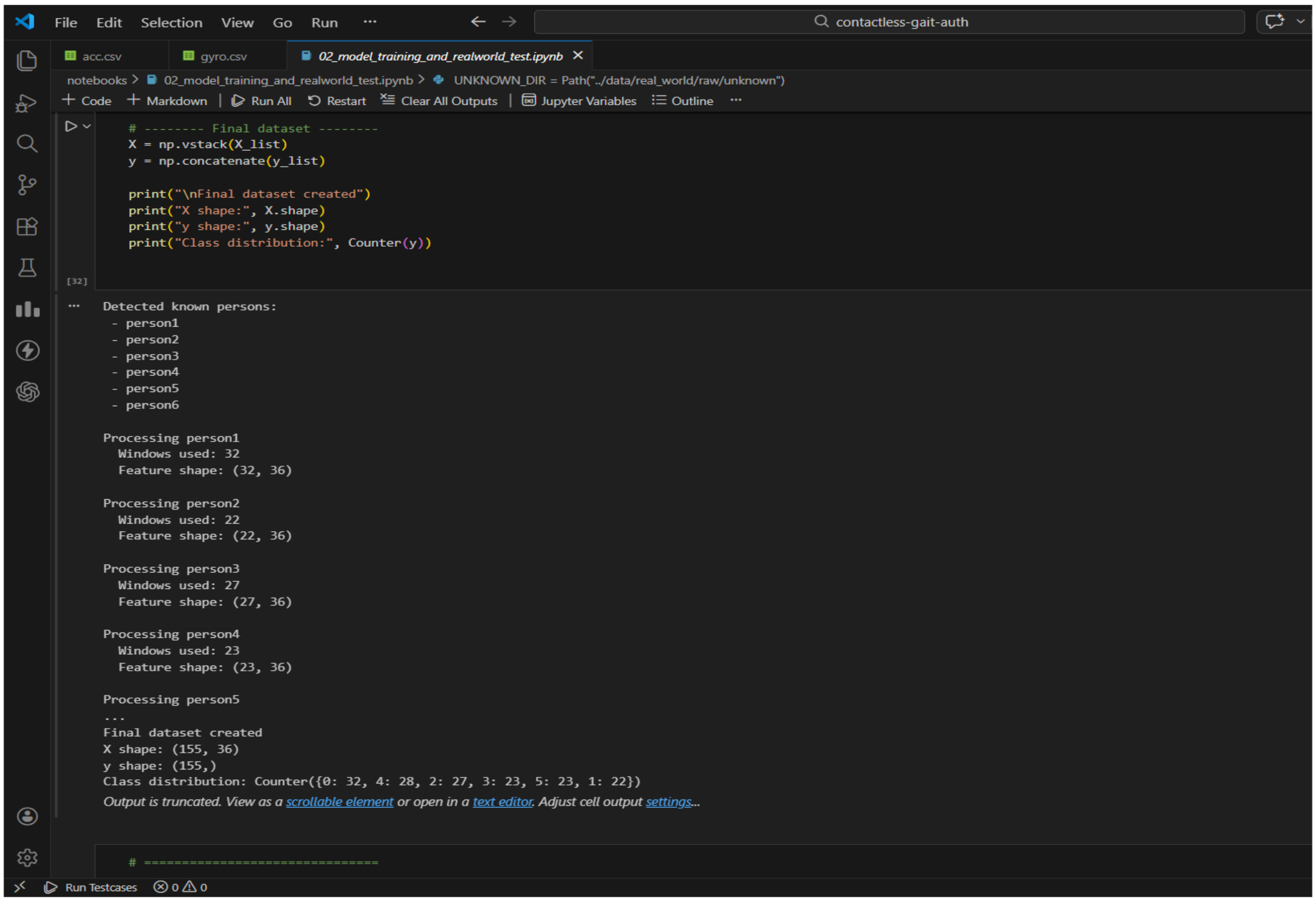


```
CONTACTLESS-GAIT-AUTH
  > __pycache__
  v data
    v real_world\raw
      v person1
          acc.csv
          gyro.csv
      > person2
      > person3
      > person4
      > person5
      > person6
      v unknown
          acc.csv
          gyro.csv
    v uci
      v test
        v Inertial Signals
            body_acc_x_test.txt
            body_acc_y_test.txt
            body_acc_z_test.txt
            body_gyro_x_test.txt
            body_gyro_y_test.txt
            body_gyro_z_test.txt
            total_acc_x_test.txt
            total_acc_y_test.txt
            total_acc_z_test.txt
          subject_test.txt
          X_test.txt
          y_test.txt
      > train
  > models
  v notebooks
      01_gait_analysis_and_features.ipynb
      02_model_training_and_realworld_test.ipynb
  v src
      feature_extraction.py
      inference.py
      preprocessing.py
```

```python
# llm_explainer.py

def explain_decision(summary: dict) -> str:
    """
    This function simulates LLM behavior.
    Replace with real LLM API if required.
    """

    if summary["decision"] == "ACCESS GRANTED":
        return (
            f"Access was granted because the walking pattern closely matched "
            f"{summary['person']}. The confidence score of {summary['confidence']} "
            f"is above the required threshold of {summary['threshold']}."
        )

    return (
        f"Access was denied because the walking pattern did not sufficiently "
        f"match any enrolled employee. The confidence score of "
        f"{summary['confidence']} was below the acceptable threshold."
    )
```

Figure 1: Real-world dataset folder structure used for employee enrollment and unknown user testing.

4.2 Feature Extraction / Preprocessing Output


File Edit Selection View Go Run ···
contactless-gait-auth
acc.csv
gyro.csv
02_model_training_and_realworld_test.ipynb
notebooks > 02_model_training_and_realworld_test.ipynb > UNKNOWN_DIR = Path("../data/real_world/raw/unknown")
+ Code + Markdown | Run All Restart Clear All Outputs | Jupyter Variables Outline ···
Run Testcases ⓧ 0 ⚠ 0

```
# -------- Final dataset --------
X = np.vstack(X_list)
y = np.concatenate(y_list)

print("\nFinal dataset created")
print("X shape:", X.shape)
print("y shape:", y.shape)
print("Class distribution:", Counter(y))
```

[32]

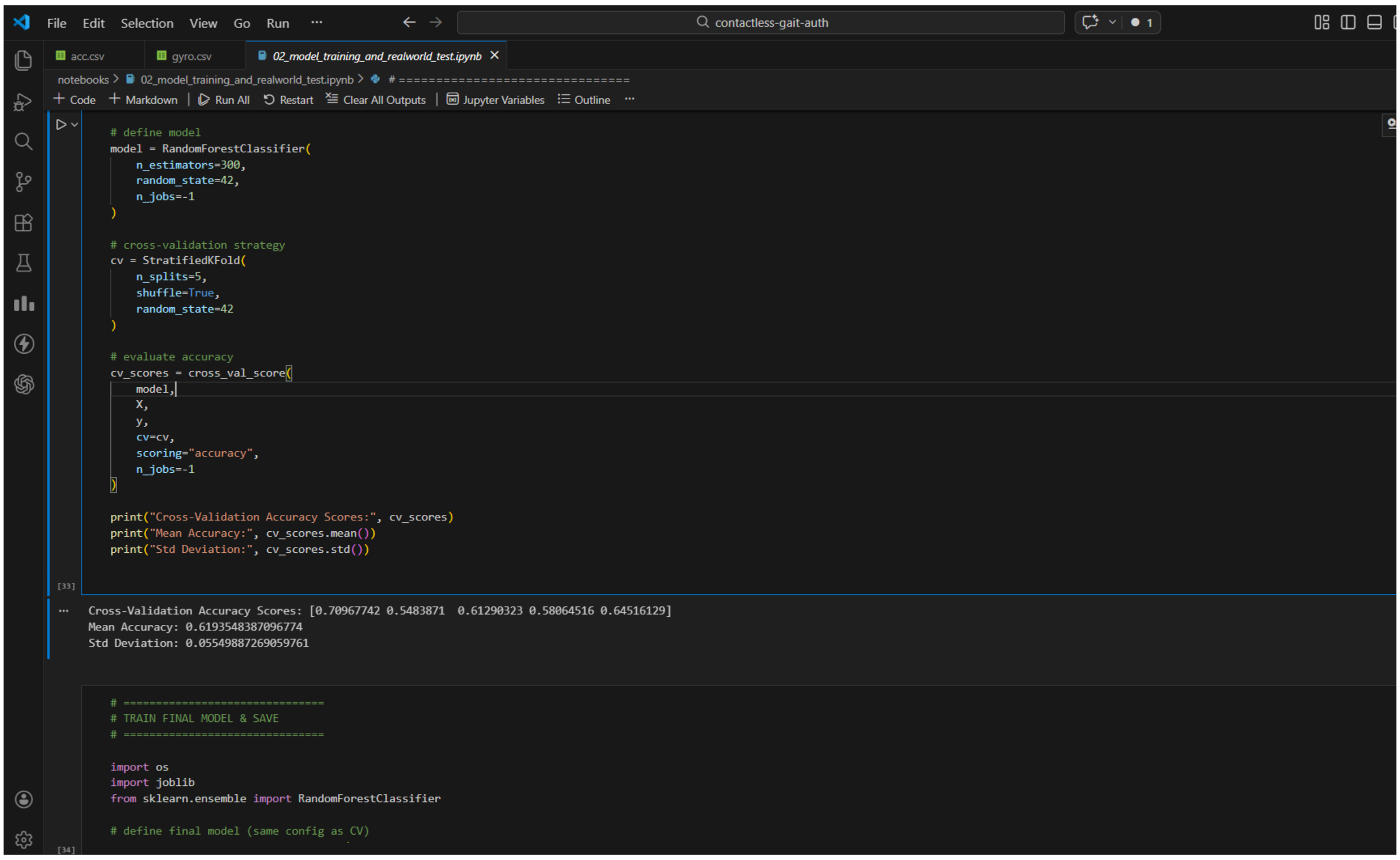
```
Detected known persons:
 - person1
 - person2
 - person3
 - person4
 - person5
 - person6

Processing person1
  Windows used: 32
  Feature shape: (32, 36)

Processing person2
  Windows used: 22
  Feature shape: (22, 36)

Processing person3
  Windows used: 27
  Feature shape: (27, 36)

Processing person4
  Windows used: 23
  Feature shape: (23, 36)

Processing person5
 ...
Final dataset created
X shape: (155, 36)
y shape: (155,)
Class distribution: Counter({0: 32, 4: 28, 2: 27, 3: 23, 5: 23, 1: 22})
```

Output is truncated. View as a scrollable element or open in a text editor. Adjust cell output settings...

```
# ==============================
```

Figure 2: Example of preprocessing and feature extraction applied to raw sensor data.

4.3 Model Training and Evaluation Output


File Edit Selection View Go Run ···
contactless-gait-auth
● 1
acc.csv
gyro.csv
02_model_training_and_realworld_test.ipynb
notebooks > 02_model_training_and_realworld_test.ipynb > # ==============================
+ Code + Markdown | Run All Restart Clear All Outputs | Jupyter Variables Outline ···

```
# define model
model = RandomForestClassifier(
    n_estimators=300,
    random_state=42,
    n_jobs=-1
)

# cross-validation strategy
cv = StratifiedKFold(
    n_splits=5,
    shuffle=True,
    random_state=42
)

# evaluate accuracy
cv_scores = cross_val_score(
    model,
    X,
    y,
    cv=cv,
    scoring="accuracy",
    n_jobs=-1
)

print("Cross-Validation Accuracy Scores:", cv_scores)
print("Mean Accuracy:", cv_scores.mean())
print("Std Deviation:", cv_scores.std())
```

[33]

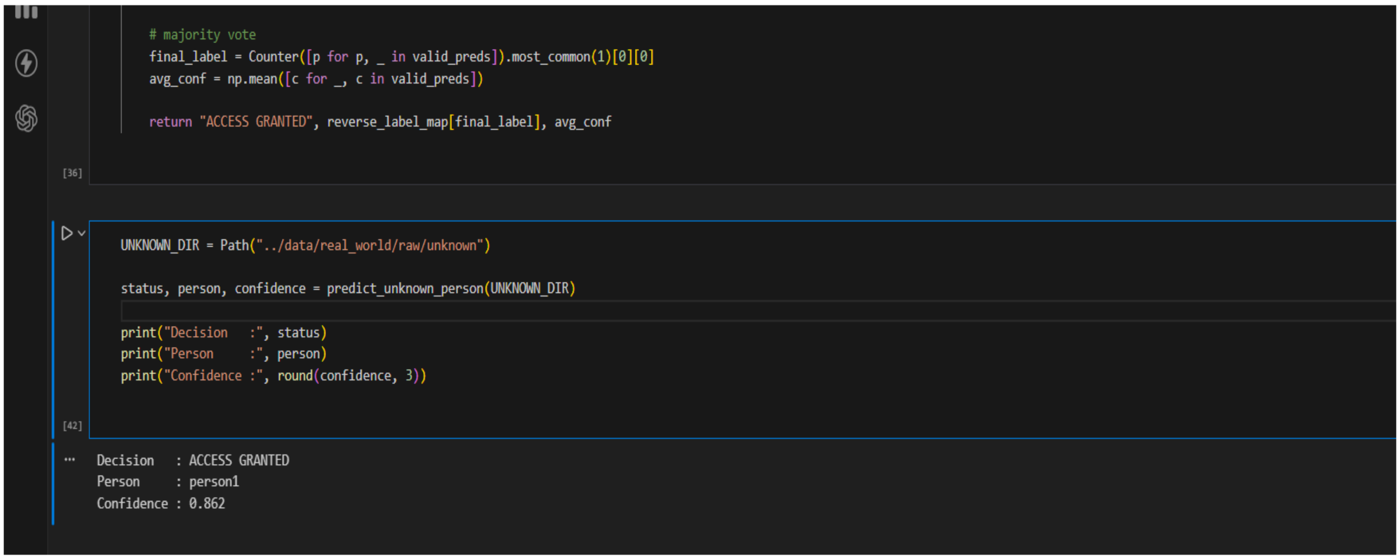
```
Cross-Validation Accuracy Scores: [0.70967742 0.5483871  0.61290323 0.58064516 0.64516129]
Mean Accuracy: 0.6193548387096774
Std Deviation: 0.05549887269059761
```

```
# ==============================
# TRAIN FINAL MODEL & SAVE
# ==============================

import os
import joblib
from sklearn.ensemble import RandomForestClassifier

# define final model (same config as CV)
```

[34]

Figure 3: Model training and cross-validation performance output.

4.4 Authentication Result (Known User)



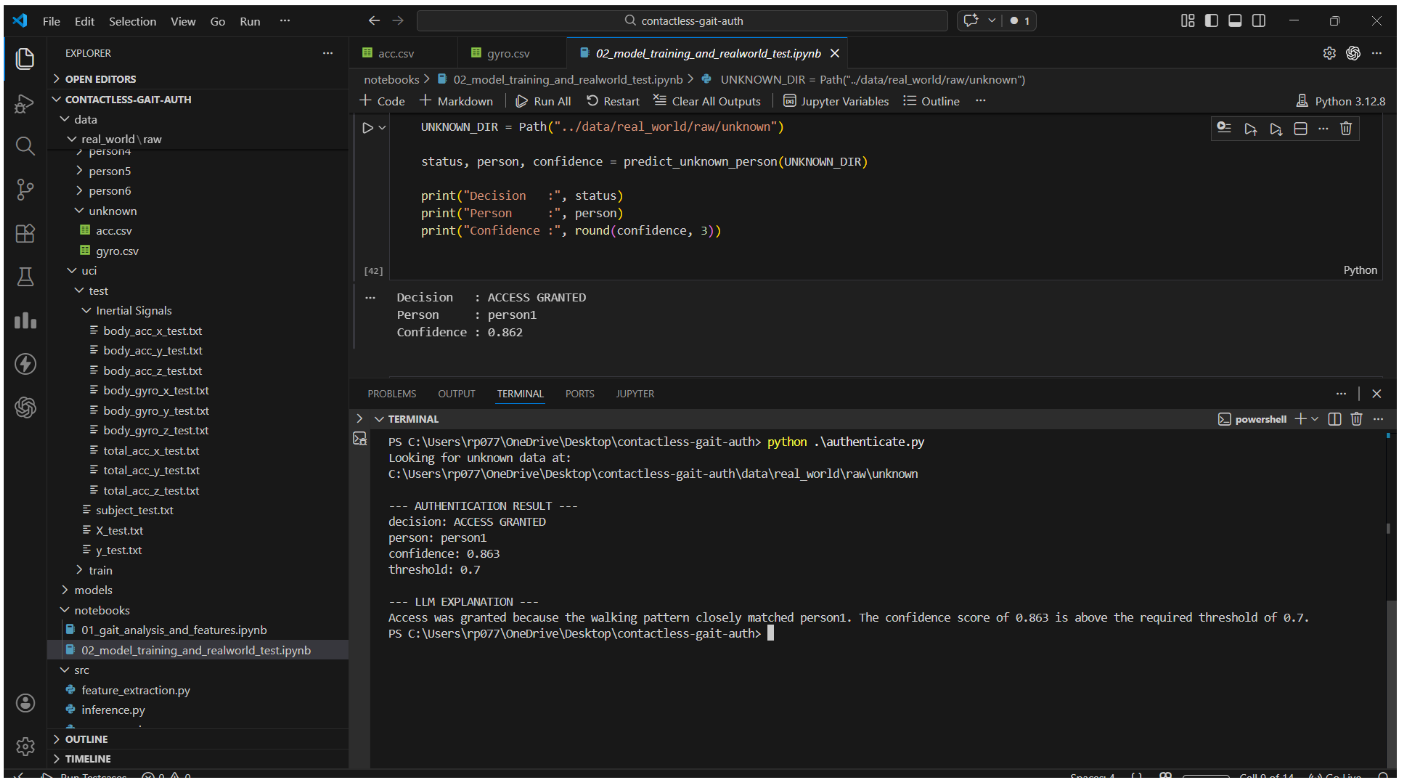
```
# majority vote
final_label = Counter([p for p, _ in valid_preds]).most_common(1)[0][0]
avg_conf = np.mean([c for _, c in valid_preds])

return "ACCESS GRANTED", reverse_label_map[final_label], avg_conf
```

```
UNKNOWN_DIR = Path("../data/real_world/raw/unknown")

status, person, confidence = predict_unknown_person(UNKNOWN_DIR)

print("Decision   :", status)
print("Person     :", person)
print("Confidence :", round(confidence, 3))
```

```
Decision   : ACCESS GRANTED
Person     : person1
Confidence : 0.862
```

Figure 4: Authentication result for an enrolled employee.

4.5 Authentication Result (Unknown User) + LLM Explanation



```
UNKNOWN_DIR = Path("../data/real_world/raw/unknown")

status, person, confidence = predict_unknown_person(UNKNOWN_DIR)

print("Decision   :", status)
print("Person     :", person)
print("Confidence :", round(confidence, 3))
```

```
Decision   : ACCESS GRANTED
Person     : person1
Confidence : 0.862
```

```
PS C:\Users\rp077\OneDrive\Desktop\contactless-gait-auth> python .\authenticate.py
Looking for unknown data at:
C:\Users\rp077\OneDrive\Desktop\contactless-gait-auth\data\real_world\raw\unknown

--- AUTHENTICATION RESULT ---
decision: ACCESS GRANTED
person: person1
confidence: 0.863
threshold: 0.7

--- LLM EXPLANATION ---
Access was granted because the walking pattern closely matched person1. The confidence score of 0.863 is above the required threshold of 0.7.
PS C:\Users\rp077\OneDrive\Desktop\contactless-gait-auth>
```

Figure 5: Authentication result for an unknown user with LLM-generated explanation.