Contactless Employee Authentication Using Gait Analysis

A Real-World Sensor-Based Machine Learning System
with Explainable AI

Presented by: Ravi pal
Institute: VIT Bhopal University
Domain: Machine Learning & Data Science

PROBLEM STATEMENT

- Traditional employee authentication systems used in offices and organizations face several practical limitations:
- Access cards and ID badges can be lost, stolen, or shared
- Password-based systems are inconvenient and insecure
- Contact-based biometrics such as fingerprints require physical interaction
- Face recognition systems raise privacy and surveillance concerns
- Most existing methods require active effort from the user
- Objective: To design a system that can authenticate employees automatically using their natural walking behavior, without requiring any explicit action.

SOLUTION: GAIT ANALYSIS:

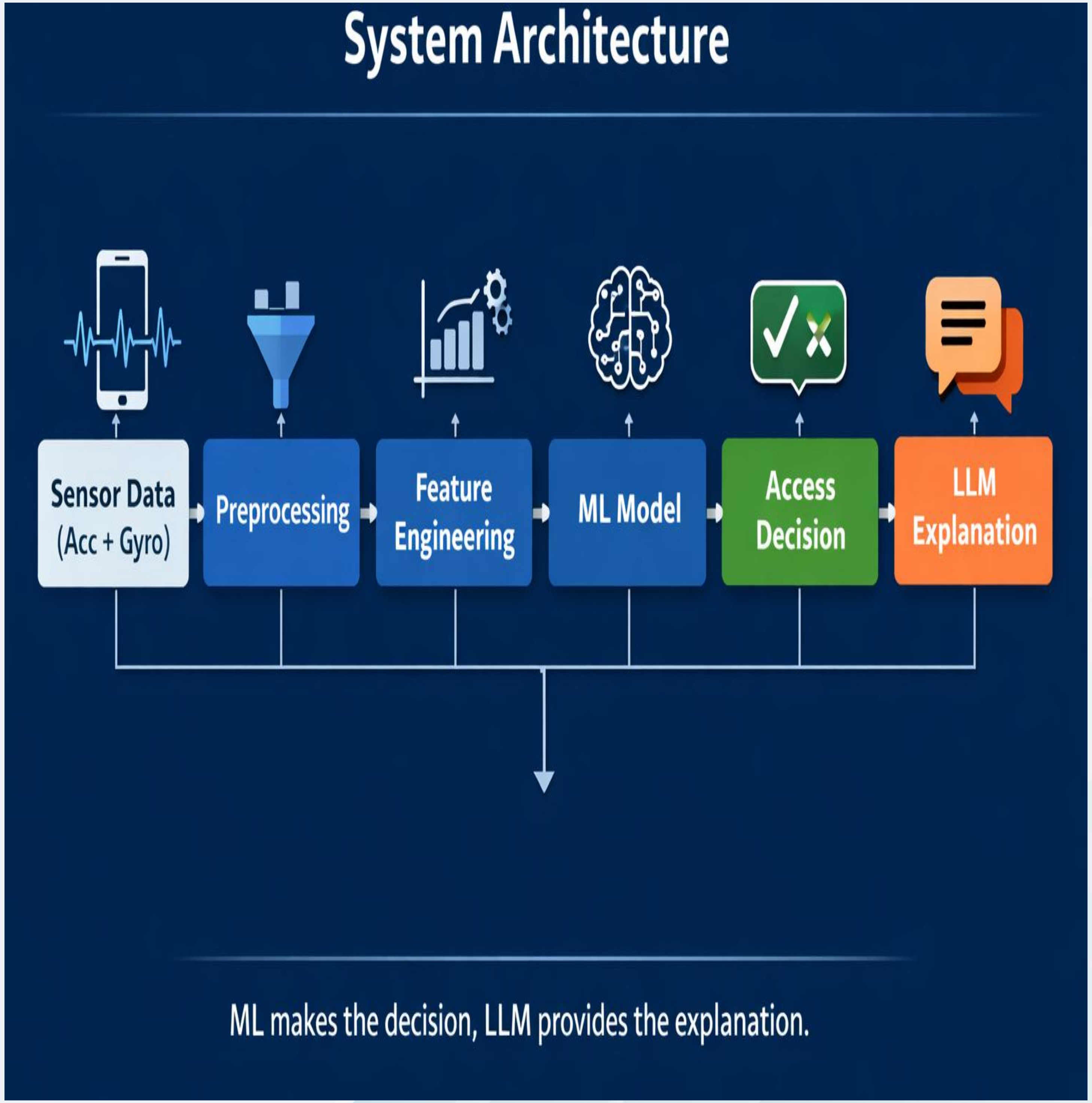
- Gait refers to the way a person walks. It is a **behavioral biometric** that is naturally produced during daily movement.
- Gait analysis is suitable for employee authentication because:
- Every individual has a **unique walking pattern**
- Gait is **difficult to consciously imitate**
- It can be captured **passively** while the user walks
- No physical contact or user interaction is required
- Smartphone sensors can record gait data **without extra hardware**
- Unlike passwords or cards, gait-based authentication works **in the background**, making it both secure and user-friendly.

SYSTEM ARCHITECTURE

- The proposed system follows a structured pipeline that converts raw sensor data into an access control decision.

Overall Flow

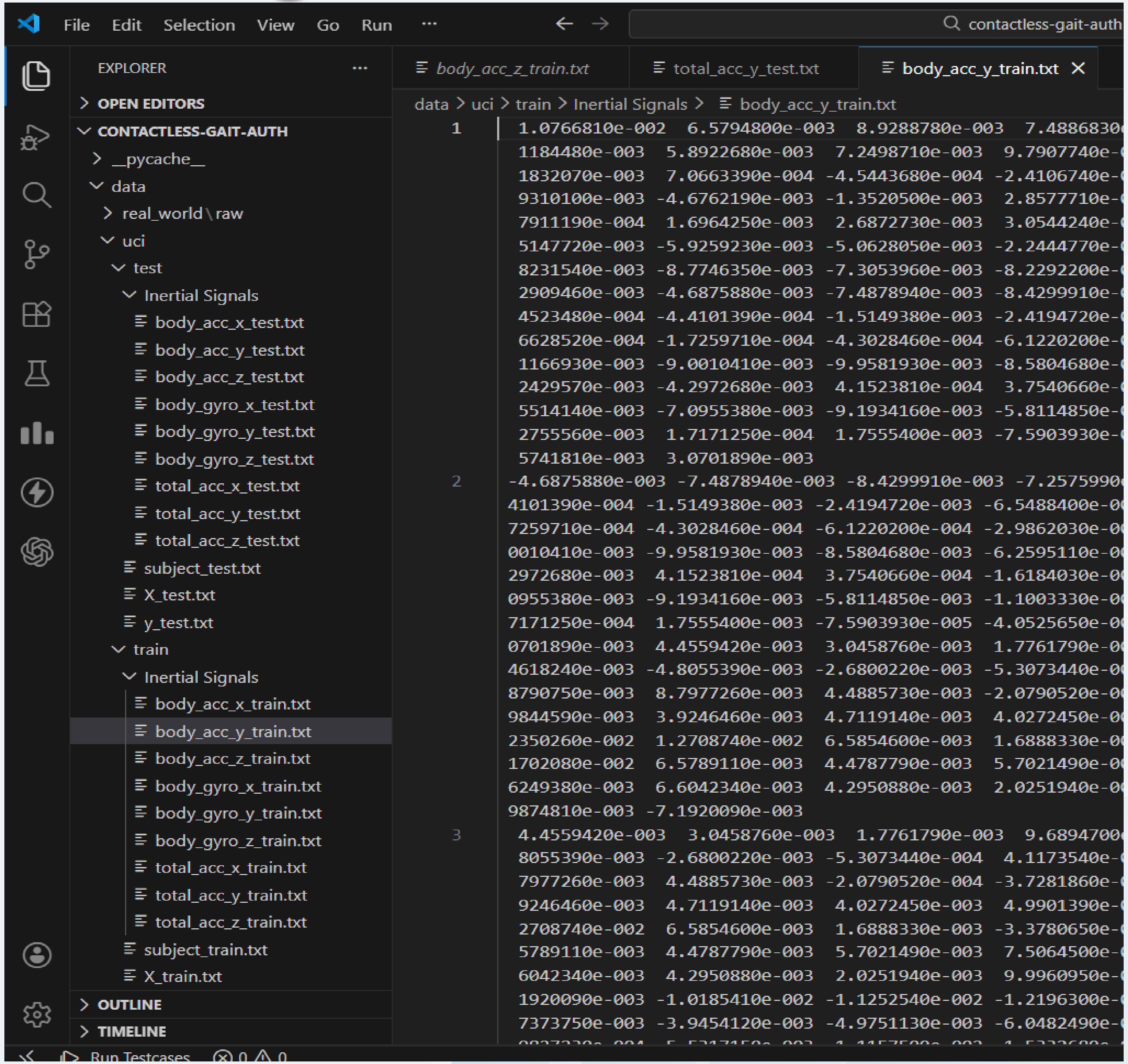
- Smartphone collects **accelerometer and gyroscope data**
- Raw sensor signals are **preprocessed** to remove noise and gravity
- Signals are segmented into windows and **features are extracted**
- A machine learning model **identifies the employee**
- A confidence-based rule decides **access grant or denial**
- An LLM generates a **human-readable explanation** of the decision



Data Collection & Preprocessing

1. Dataset-Based Training (UCI HAR)

1. Data collected from **30 subjects**
2. Sensors used:
   - **Accelerometer (3-axis)**
   - **Gyroscope (3-axis)**
3. Sampling frequency: **50 Hz**
4. Activities used for gait:
   - Walking
   - Walking Upstairs
   - Walking Downstairs
5. Preprocessed using:
   - Noise filtering
   - Gravity separation (Butterworth filter)
   - Sliding window segmentation
     (2.56 sec, 128 samples)



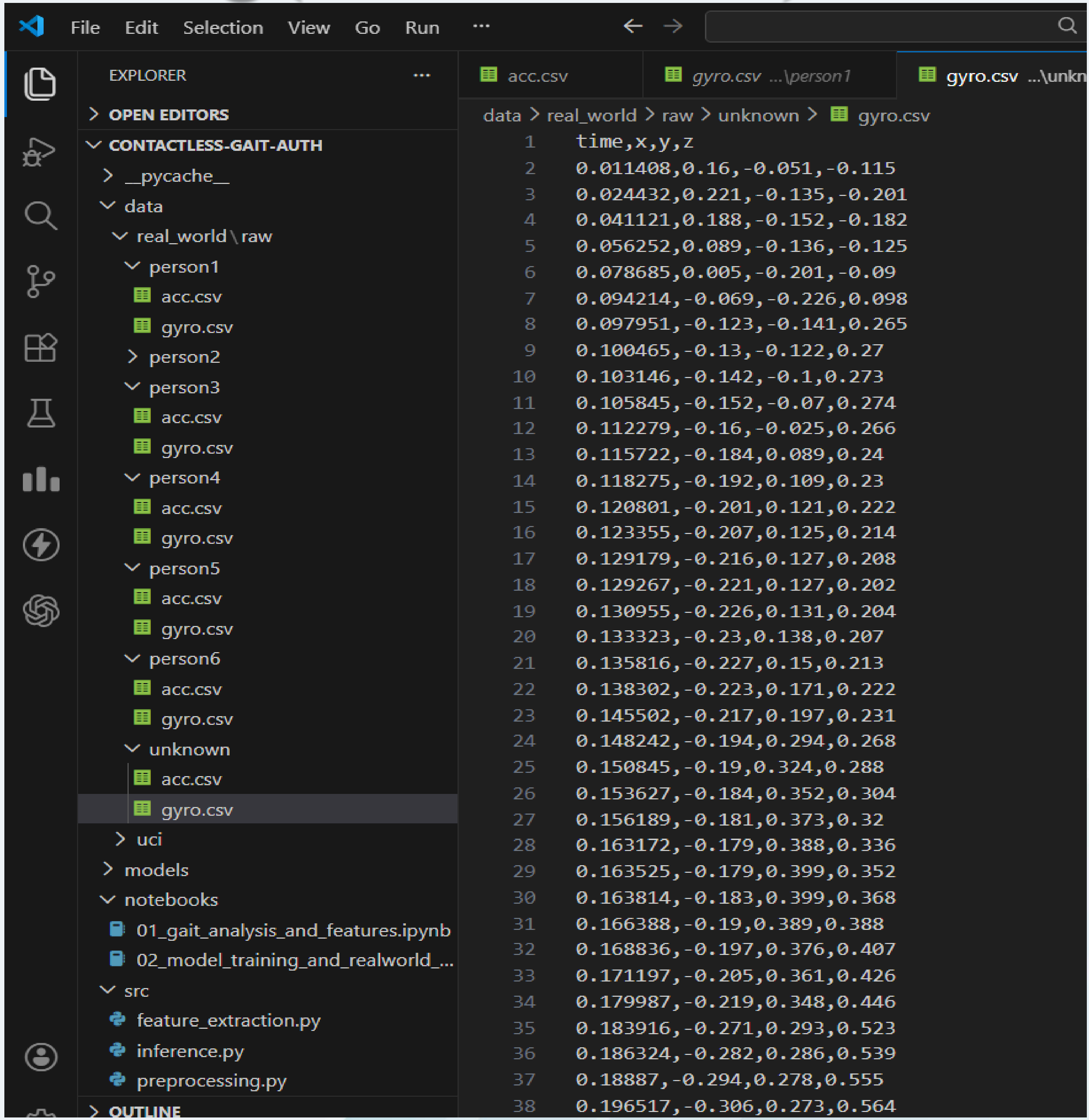
Data Collection & Preprocessing(Continue…)

2. Real-World Data Collection

- Collected using **Physics Toolbox Sensor Suite (Android app)**
- Data collected from **5+ real users**
- Raw data format:
  - time, x, y, z
- Sensors:
  - Accelerometer
  - Gyroscope
- No preprocessing applied at collection time

3. Preprocessing Pipeline (Applied by Us)

- Resampling to match **50 Hz**
- Noise filtering (low-pass filter)
- Gravity removal
- Windowing to fixed-size segments
- Feature extraction to match training format



MODEL TRAINING, EVALUATION & RESULTS

1. Problem Formulation

- Multi-class classification problem
- Each employee is treated as **one class**
- Goal: **Identify the employee from gait patterns**

2. Model Selection

- **Random Forest Classifier**
- Chosen because:
  - Handles non-linear gait patterns
  - Robust to noise in real-world sensor data
  - Performs well with limited samples
  - No strict feature scaling required

3. Training Strategy

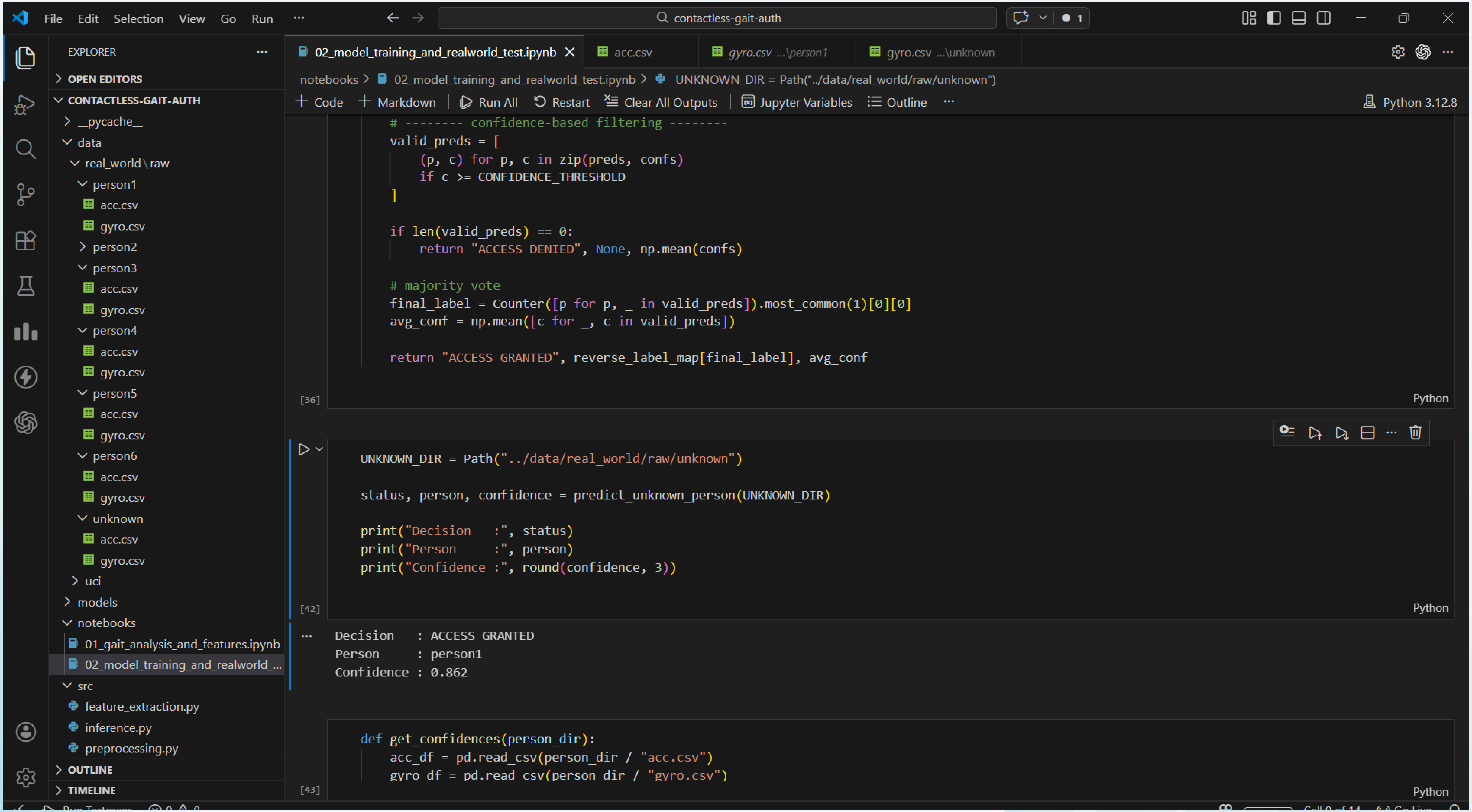
- Trained on **walking-related activities only**
  - Walking
  - Walking Upstairs
  - Walking Downstairs
- Feature vectors extracted from sliding windows
- Cross-validation used for performance estimation

4. Evaluation Metrics

- Cross-validation accuracy
- Confidence scores per prediction
- Threshold-based decision for access control

Results (Short & Honest)

- Model shows **high confidence** for enrolled employees
- **Low confidence** for unknown users
- Clear separation between known and unknown gait patterns


File Edit Selection View Go Run ···
contactless-gait-auth
02_model_training_and_realworld_test.ipynb
acc.csv
gyro.csv ...\person1
gyro.csv ...\unknown
EXPLORER
OPEN EDITORS
CONTACTLESS-GAIT-AUTH
pycache
data
real_world \ raw
person1
acc.csv
gyro.csv
person2
person3
acc.csv
gyro.csv
person4
acc.csv
gyro.csv
person5
acc.csv
gyro.csv
person6
acc.csv
gyro.csv
unknown
acc.csv
gyro.csv
uci
models
notebooks
01_gait_analysis_and_features.ipynb
02_model_training_and_realworld_...
src
feature_extraction.py
inference.py
preprocessing.py
OUTLINE
TIMELINE
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
Python 3.12.8
# -------- confidence-based filtering --------
valid_preds = [
(p, c) for p, c in zip(preds, confs)
if c >= CONFIDENCE_THRESHOLD
]
if len(valid_preds) == 0:
return "ACCESS DENIED", None, np.mean(confs)
# 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
[36]
Python
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))
[42]
Python
Decision   : ACCESS GRANTED
Person   : person1
Confidence : 0.862
def get_confidences(person_dir):
acc_df = pd.read_csv(person_dir / "acc.csv")
gyro_df = pd.read_csv(person_dir / "gyro.csv")
[43]
Python

LLM-BASED EXPLAINABILITY & DECISION TRANSPARENCY

- **Why Explainability Is Required**
  - The machine learning model produces **numerical outputs** such as probabilities and confidence scores
  - These outputs are **not easily understandable** by security staff or non-technical users
  - For access control systems, it is important to clearly explain **why access was granted or denied**
- **How the LLM Is Used**
  - After the ML model predicts the employee and confidence score:
    - A short decision summary is created
    - This summary is passed to the LLM
  - The LLM converts the decision into a **simple, human-readable explanation**
- **Key Design Principle**
  - The **ML model makes the decision**
  - The **LLM only explains the decision**