Project: Coding Fingerprint Classifier

Names: Mayan Sequeira and Mohammed Bilal

SRN: PES2UG23CS332 and PES2UG23CS344

1. Project Statement

This project aims to predict a competitive programmer's skill rank and country by analyzing their source code. The idea is that every programmer has a unique "coding fingerprint" — a combination of habits, patterns, and style choices that reflect their background, experience, and training.

By studying these fingerprints, the system can learn to recognize patterns that correlate with skill levels and regional trends. Such a model could be applied in educational tools, technical recruitment, and global programming analytics, helping to understand how coding styles differ across the world.

2. High-Level Architecture

The project follows a structured machine learning pipeline consisting of data collection, feature extraction, and model training:

Data Collection:

A resumable Python script gathers and filters data for over 14,000 competitive programmers from the Codeforces API, including their profiles and submitted source codes.

Feature Engineering:

Each source code file is analyzed to extract style-based numerical features such as:

Keyword frequency (e.g., use of loops, conditionals), Average code length and number of functions, Indentation and spacing patterns, Variable naming styles

These features represent a programmer's unique coding behavior.

Model Training:

A Random Forest Classifier is trained on the extracted features to predict both rank and country.The model uses the parameter class_weight='balanced' to handle the imbalanced dataset, ensuring that less common classes (e.g., rare countries or high-rated programmers) are not ignored during training.

3. Result

The project successfully showed that a programmer's "coding fingerprint" can be used to predict their skill rank and country of origin with good accuracy. The Random Forest Classifier effectively captured stylistic patterns such as code structure, indentation, and keyword preferences. The class_weight='balanced' parameter was essential for dealing with the uneven data distribution, leading to more stable and fair predictions. Overall, the system achieved promising results, confirming that source code contains meaningful signals that reflect both the skill level and the cultural or educational background of a programmer. This experiment opens up possibilities for future research in automated skill assessment and developer profiling using coding patterns.