ASSIGNMENT-2

Name: Dhruv Kasana

Course -B.Tech. C.S.E.

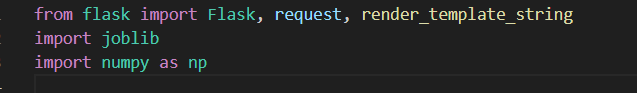


Batch: 11

Sap.id.: 500120632

Enrolment no.: R2142230320

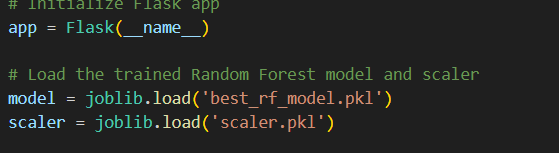
Instructor: Prof. C.M. Sherma



Flask: A micro web framework to create a web application.

**joblib**: Used for loading the pre-trained model and scaler.

**NumPy**: To handle numerical data, especially arrays.



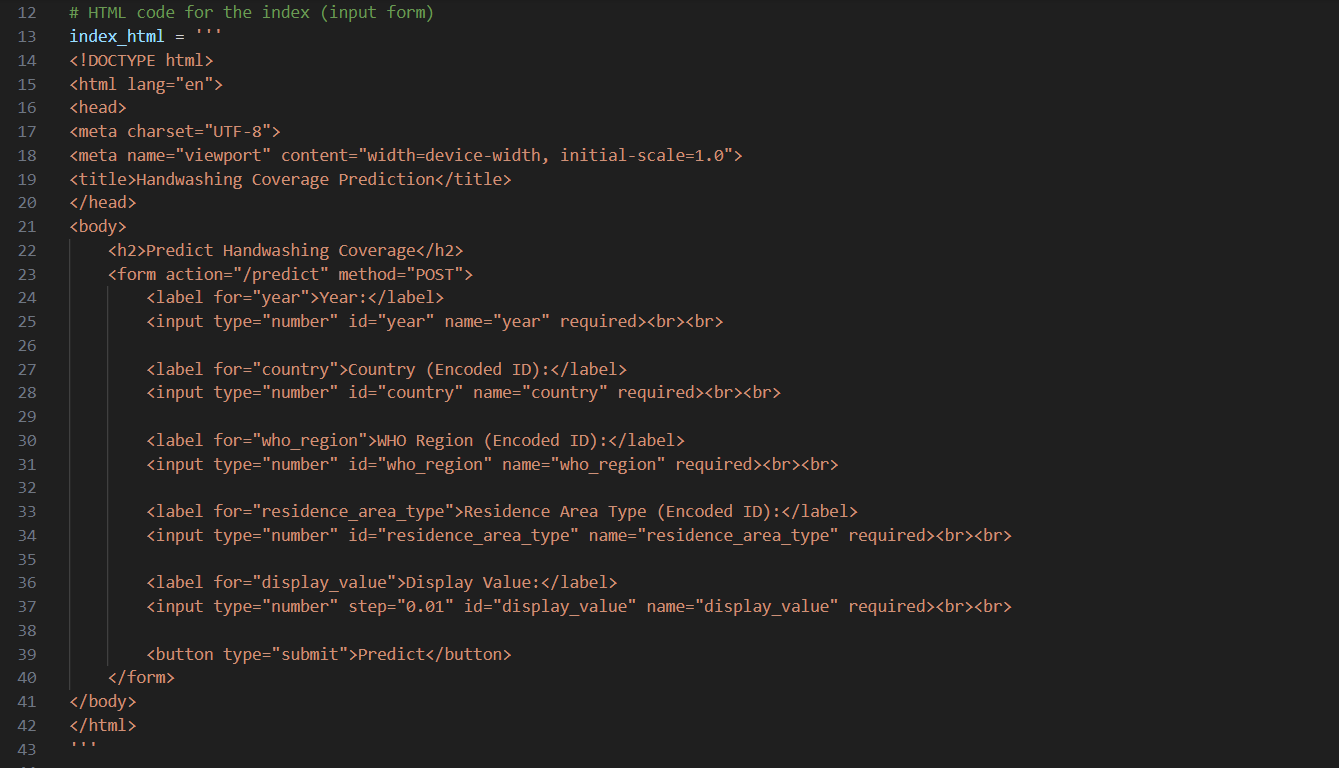
**2. Initialize Flask App**

This initializes the Flask application.

**3. Load the Pre-trained Model and Scaler**

**best\_rf\_model.pkl**: The trained Random Forest model saved using joblib.

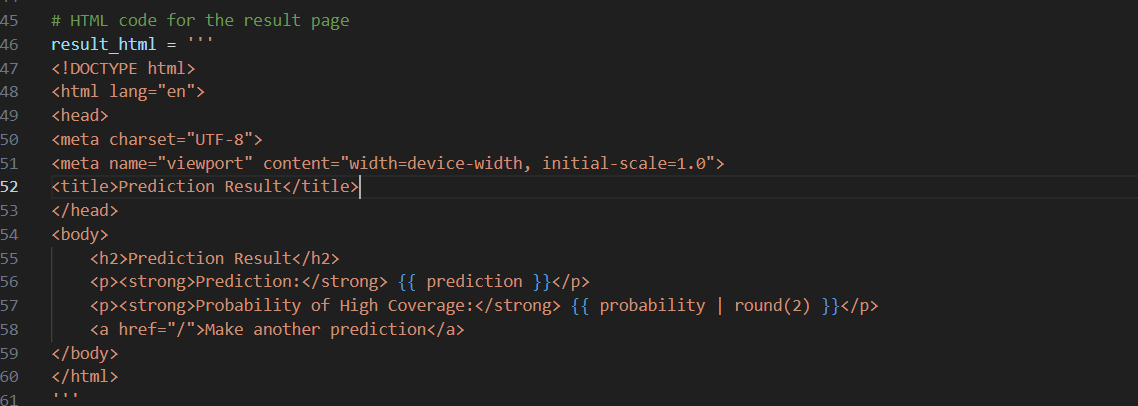
* **scaler.pkl**: The Standard Scaler used to standardize the features during training.



This is a simple HTML form where users can input:

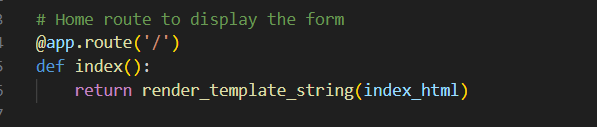
* **Year**: The year of data (e.g., 2020).
* **Country (Encoded ID)**: A numeric ID representing the country (label-encoded).
* **WHO Region (Encoded ID)**: Encoded ID for the WHO region.
* **Residence Area Type (Encoded ID)**: Encoded ID (e.g., 0 for Urban, 1 for Rural).
* **Display Value**: The numeric value representing handwashing coverage.

The form uses the POST method to send input data to the predict route.



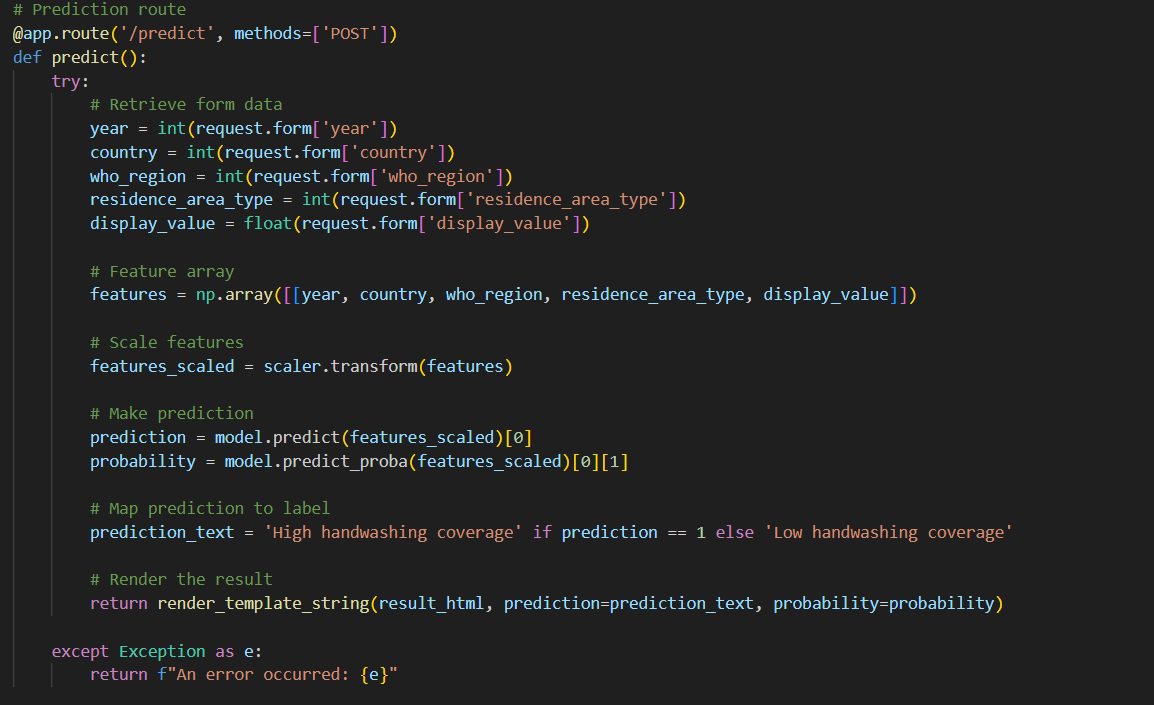
This HTML template displays the prediction result:

* **Prediction**: Whether the model predicts high or low handwashing coverage.
* **Probability**: The probability score for high handwashing coverage (rounded to 2 decimal places).
* A link to go back to the input form.



This route displays the input form when a user navigates to the home page (/).

**render\_template\_string**: Directly renders the HTML form defined in index\_html



**Form Data Retrieval**: Extracts the input values from the form using request. Form.

**Feature Array**: Creates a 2D NumPy array with the input values to be used for prediction.

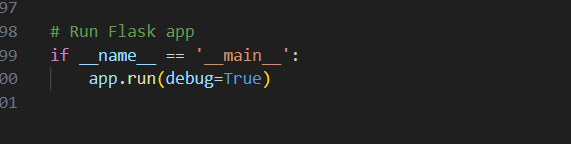
**Scaling**: Standardizes the input features using the loaded scaler.

**Prediction**:

* **model.predict(features\_scaled)**: Predicts whether the handwashing coverage is high or low (binary: 1 or 0).
* **model.predict\_proba(features\_scaled)**: Returns the probabilities for each class. [0][1] is the probability of class 1 (high coverage).

**Mapping Prediction**: Converts the binary prediction into a text label.

**Rendering Result**: Displays the prediction result using result\_html.



Starts the Flask server in debug mode, allowing live updates and error logging.

SUMMARY:

* **Form Input**: The user enters features like year, country ID, WHO region ID, area type, and display value.
* **Model Prediction**: The features are scaled and fed into the trained Random Forest model for prediction.
* **Output**: The web app displays whether handwashing coverage is predicted to be high or low and shows the probability of high coverage.

This simple web interface allows users to input data and get real-time predictions using the trained model. It's a good approach for quickly testing a machine learning model with different inputs without the need for a complex front-end setup.