# BMI , GENDER PREDICTION USING PATTERN RECOGNITION AND MACHINE LEARNING

**CS22B1100 R.SHRINIDHI**

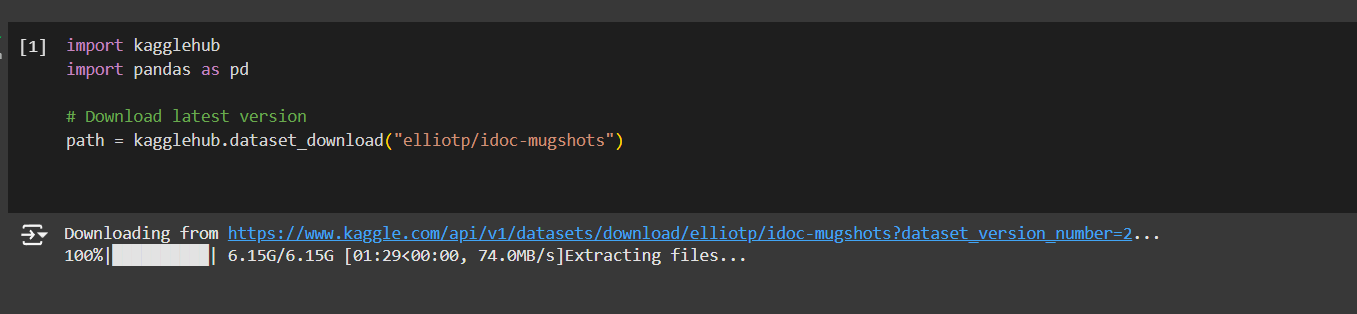
## Objective

The project aims to process and analyze mugshot images from the IDOC mugshots dataset to:  
1. Extract image features using Opencv   
2. Predict Body Mass Index (BMI) and gender from the features using Random forest regressor and classifier  
3. Evaluate the model's performance using metrics like Mean Squared Error (MSE), R² score, and accuracy.

## Implementation details

### Dataset Handling

* Dataset Source: Downloaded using `kagglehub` from the dataset ID `elliotp/idoc-mugshots`.
* Base Directory: `/root/.cache/kagglehub/datasets/elliotp/idoc-mugshots/versions/2`
* File Structure:  
   - Labels File: `labels\_utf8.csv`.  
   - Images: Stored in `front/front` and `side/side` directories.



### Data Preprocessing

* The labels file contains metadata like prisoner ID, gender, height, weight, hair/eye color, and reason for imprisonment.
* Columns like `Height` and `Weight` are parsed for numerical computation
* Height Conversion:
* Converts height from 'ft. in.' format to meters.
* Uses a conversion factor: 1 foot = 0.3048 meters, 1 inch = 0.0254 meters.
* Weight Conversion:
* Converts weight from 'lbs.' to kilograms
* Uses a conversion factor: 1 pound = 0.453592 kg

### Image Feature Extraction

* Convert height and weight
* Read images
* Resize images to a fixed size: **Image Resizing: cv2.resize()**
* Convert images to grayscale: **Convert to Grayscale: cv2.cvtColor()**
* Extract mean and standard deviation of pixel intensities(front and side separately) **Mean and Standard Deviation: np.mean() and np.std()**
* **Mean intensity: Indicates overall brightness, which could correlate with factors like skin tone or lighting consistency.**
* **Standard deviation: Reflects texture and contrast, capturing details like body contours and muscle/fat distribution.**
* Combine all features
* Append to the list
* Convert features list to DataFrame
* Add features to the labels DataFrame

**4.Using the extract features functions and calculating bmi**

* **X**: A list to store extracted features from images (front and side views).
* **y\_bmi**: A list to store the BMI values calculated from height and weight.
* **y\_gender**: A list to store gender labels (1 for Male, 0 for Female).
* Extracting features from paired images (front and side).
* Calculating BMI.
* Encoding gender.
* This processed data is used for regression (BMI prediction) and classification (gender prediction).

### 5.Prediction Tasks:

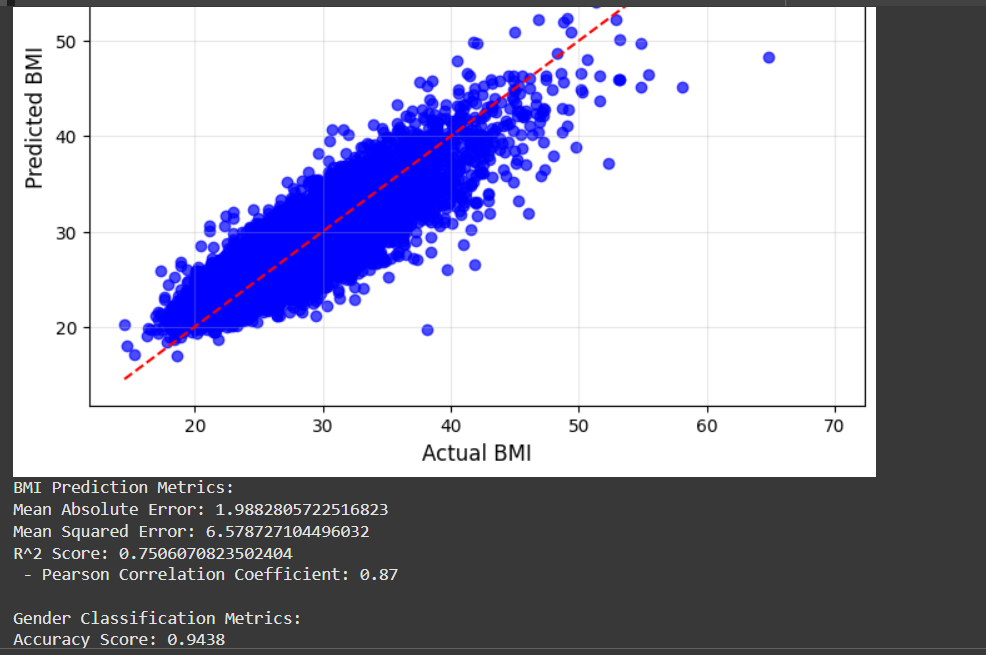
* Converts the lists X, y\_bmi, and y\_gender into NumPy arrays to allow numerical operations and compatibility with machine learning models.
* **train\_test\_split()**: Divides the dataset into training and testing sets
* **Outputs**:
  + - X\_train, X\_test: Features for training and testing.
    - y\_bmi\_train, y\_bmi\_test: BMI targets for training and testing.
    - y\_gender\_train, y\_gender\_test: Gender targets for training and testing.
* **RandomForestRegressor()**: A regression model that uses an ensemble of decision trees to predict BMI. Trains the model using the training features (X\_train) and BMI labels (y\_bmi\_train).
* **RandomForestClassifier()**: A classification model that uses an ensemble of decision trees to predict gender. Trains the model using the training features (X\_train) and gender labels (y\_gender\_train).
* **predict()**: Generates predictions for the test dataset (X\_test).
* **y\_bmi\_pred**: Predicted BMI values (continuous outputs from the regression model).
* **y\_gender\_pred**: Predicted gender labels (binary outputs from the classification model: 0 or 1).

## 6.Model Metrics

* **BMI Metrics**:
  + **Mean Absolute Error (MAE)**: Displays the average error in predictions
  + **Mean Squared Error (MSE)**: Shows the average squared difference between the predicted and actual values, indicating the model’s precision.
  + **R² (Coefficient of Determination)**: Reflects the proportion of the variance in the target that is predictable from the model.
  + **Pearson Correlation Coefficient**: Provides a measure of the linear relationship between the actual and predicted BMI values.
* **Gender Metrics**:
  + **Accuracy**: Provides a straightforward metric of the model's classification accuracy for gender prediction.
* **Classify BMI Function**:

Categorizes the predicted BMI into one of the classes based on predefined BMI thresholds:

* + - **Underweight**: BMI < 18.5
    - **Normal**: 18.5 ≤ BMI < 25
    - **Overweight**: 25 ≤ BMI < 30
    - **Obese**: BMI ≥ 30



**7.Prediction for user given image:**

* rf\_regressor: Trained Random Forest Regressor model for BMI prediction.
* rf\_classifier: Trained Random Forest Classifier model for gender classification.
* Both are passed as parameters for the predict bmi,gender function
* Get image paths from user
* Load the images
* Check if images are loaded correctly
* Preprocess images: resize to a fixed size and convert to grayscale
* Extract features: mean and standard deviation of pixel intensities
* Combine features into a single array
* Predict BMI and gender
* Return the predictions



**8.Plot distribution of Offences**

* Initialize an empty set all\_keywords to store unique keywords.
* Split each offense string by comma, clean each keyword, and add to the set.
* Return the sorted list of unique keywords.
* Split the offense string by comma and clean each keyword.
* For each unique keyword, check if it's present in the offense list.
* Create a dictionary with each keyword marked as 1 if present, otherwise 0.
* Transform the 'Offense' column into a DataFrame with offense categories.
* Determine the number of plots (num\_plots) and rows (num\_rows) based on the number of unique keywords.
* Create subplots using plt.subplots() with appropriate size.
* For each chunk of offenses, compute the sum of occurrences across inmates using df\_offenses[chunk].sum(axis=0).
* Plot a bar plot for each chunk using sns.barplot(), setting titles and axis labels.
* Adjust layout with plt.tight\_layout() for better display.

