

Import Libraries

```
In [1]: import os
import joblib
import numpy as np
import face_recognition
import ipywidgets as widgets
from IPython.display import display
```

Load the Trained BMI Model

```
In [2]: rf_model = 'rf_bmi_model_tuned'
model = joblib.load(rf_model)
```

Code Explanation

This model is trained on Illinois Prisoner dataset, which includes a full-frontal face image of prisoners alongside their weight and height, which makes it possible to calculate their BMI.

Define Helper Functions

```
In [3]: def face_into_encoding(path):
image = face_recognition.load_image_file(path)
face_encoding = face_recognition.face_encodings(image)
if not face_encoding:
    print("skipped" + path)
    return np.zeros(128).tolist()
return face_encoding[0].tolist()

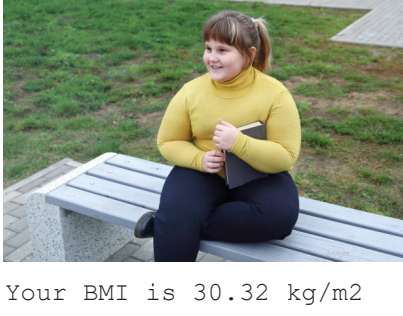
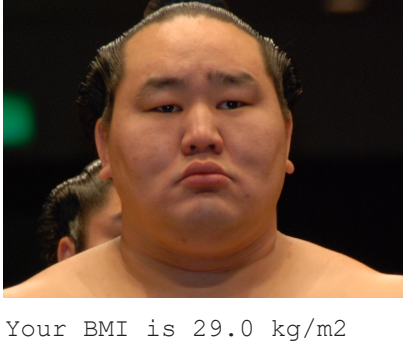
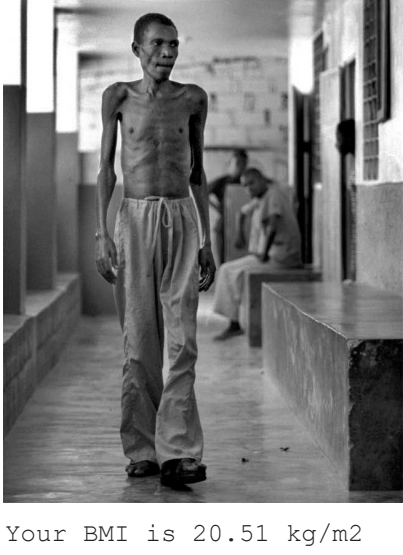
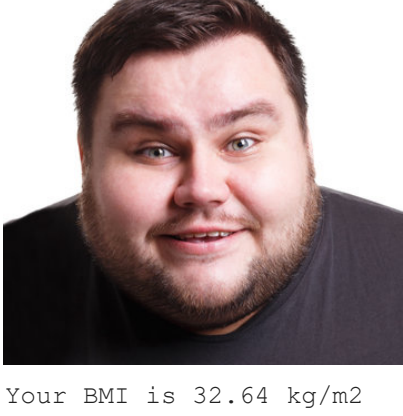
In [4]: def predict_bmi(X_images, model):
X = np.expand_dims(np.array(face_into_encoding(X_images)), axis=0)
log_value = model.predict(X)
bmi = np.exp(log_value)
return bmi.tolist()
```

Code Explanation

The face_into_encoding function is responsible for capturing the face inside the image, similar to extract_face function defined in the facial recognition test case, except that this one also turn the face image into face encodings in a form of numpy array. The predict_bmi function is using the face_into_encoding function to get the face encoding of the input image, before passing it to the model's predict function. The output of the predict function is a logarithm, so we want to get the exponential of the value to get the BMI by using numpy's exp function.

Test Case (Tested on Obese and Skinny People)

```
In [5]: images = os.listdir('bmitrial')
for img in images:
    img_location = 'bmitrial/' + img
    img = open(img_location, 'rb')
    test_img = img.read()
    preds = predict_bmi(img_location, model)
    display(widgets.Image(value=test_img, format='jpg', width=200, height=200))
    print("Your BMI is " + str(round(preds[0], 2)) + " kg/m2")
```



Code Explanation

Here, we loop through every images inside bmitrial folder and use the predict_bmi function to get their BMI.

```
In [ ]: 
```